

# TP3 - Opérations sur les FT

Les pôles et les zéros sont des éléments nécessaires pour l'étude d'un système.

Pour obtenir les pôles d'une fonction de transfert on utilise la commande *pole*.

```
>> pole(G)
ans =
-10.0000
-4.0000
-0.1000
```

Pour obtenir les zéros d'une fonction de transfert on utilise la commande *zero*.

```
>> zero(G)
ans =
-2
-1
```

Pour calculer la fonction de transfert équivalente à deux fonctions de transfert en série on utilise la commande *series*.

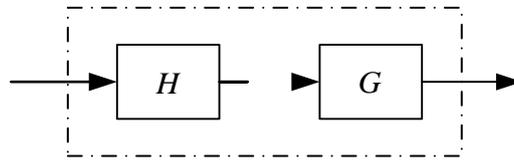


Figure 15 : fonctions de transfert en série

```
>> series(H,G)
ans =
  200 (s+1) (s+2) (s+0.25)
-----
(s+0.1) (s+4) (s+10) (s^2 + 1.5s + 3)
Continuous-time zero/pole/gain model.
```

Cette opération est équivalente à la commande

```
>>H*G
```

Pour calculer la fonction de transfert équivalente à deux fonctions de transfert en parallèle, on utilise la commande *parallel*

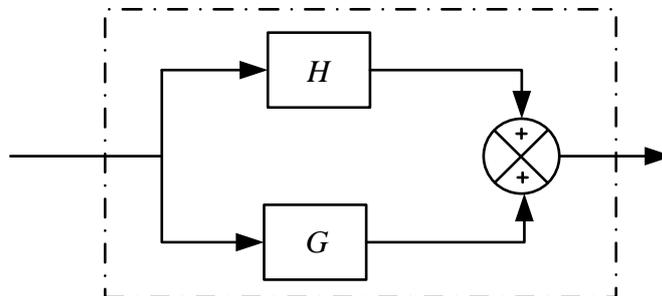


Figure 16 : fonction de transfert en parallèle

```
>> parallel(H,G)
ans =
  102 (s+1.515) (s+1.334) (s^2 + 1.844s + 2.92)
-----
(s+4) (s+10) (s+0.1) (s^2 + 1.5s + 3)
```

Continuous-time zero/pole/gain model.

Pour calculer la fonction de transfert en boucle fermée, on utilise la commande *feedback*.

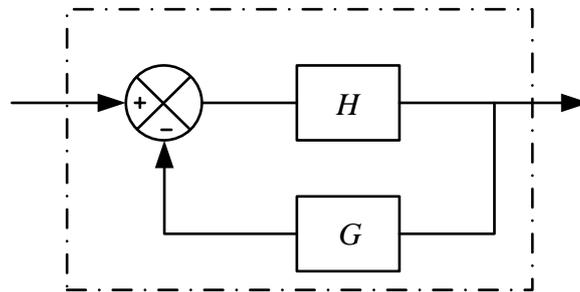


Figure 17 : fonction de transfert en boucle fermée

```
Trial>> feedback(H,G)
```

```
ans =
```

$$2 (s+0.25) (s+0.1) (s+4) (s+10)$$

-----  
 $(s+0.2104) (s^2 + 3.028s + 2.392) (s^2 + 12.36s + 222.5)$

Continuous-time zero/pole/gain model.

Pour inverser une fonction de transfert on peut utiliser la commande *inv*.

```
Trial>> inv(H)
```

```
ans =
```

$$2 s^2 + 3 s + 6$$

-----  
 $4 s + 1$

Continuous-time transfer function.

Exemple de fonction de transfert quelconque.

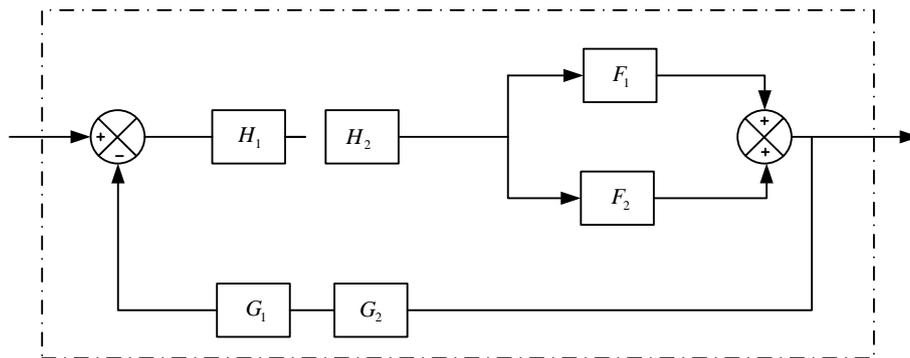


Figure 18 : fonction de transfert d'un système quelconque

On peut obtenir directement la fonction de transfert d'un tel système en tapant la commande

```
>>feedback((H1*H2)*parallel(F1,F2),G1*G2)
```

Il faudrait préalablement saisir les valeurs de chacune de ces fonctions.

Il est également très simple de construire la fonction de transfert d'un correcteur PID en utilisant la fonction *pid*.

```
>>pid(50,0.1,70)
```

*Les trois arguments représentent les coefficients  $K_p$ ,  $K_i$  et  $K_d$  du correcteur :  $pid(K_p, K_i, K_d)$*

```
ans =
```

$$K_p + K_i * \frac{1}{s} + K_d * s$$

with  $K_p = 50$ ,  $K_i = 0.1$ ,  $K_d = 70$

Continuous-time PID controller in parallel form.