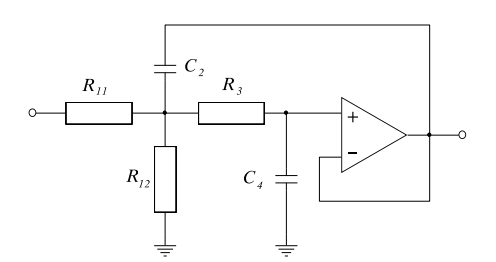
**Codes Matlab Filtres**

Nous allons ici nous focaliser sur les morceaux de code importants à récupérer dans chaque fonction matlab afin de réaliser notre code sous python. Pour ce faire, il est évident que nous négligerons tous les morceaux de code régissant l’interface graphique de matlab étant donné que nous travaillerons avec streamlit. Le tout consistant alors en un tri des morceaux de code à conserver.

* **LPLQ (Sallen-Key, Q<2)**

****

* + Code Matlab :

Nous prendrons en compte uniquement la fonction « Calculate\_Callback » qui prend en compte les calculs des composants du filtre.

Une image contenant texte

Description générée automatiquement

Initiation de la fonction.



Spécifique à Matlab

Une image contenant texte

Description générée automatiquement

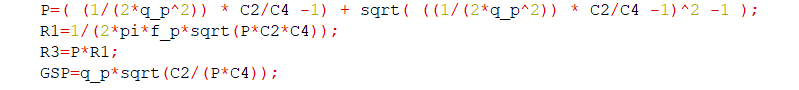
Récupération des données insérées par l’utilisateur.

Cfr Annexes :



Vérification de la condition sur C2.

Cfr Annexes :



Une image contenant texte

Description générée automatiquement Calcul des différents composants. (GSP ???)

Cfr Annexes :



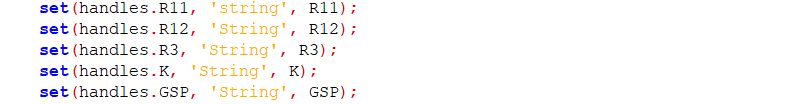
Vérification des conditions sur le K entré et application du cas où il n’est pas compris entre 0 et 1.

Cfr Annexes :



 Application du cas où K est compris entre 0 et 1.

Cfr Annexes :



Spécifique à Matlab.

Une image contenant texte

Description générée automatiquement

Spécifique à Matlab.

* + Code important pour python :

if (1/(2\*q\_p^2)) \* C2/C4 >=2;

fprintf('%f\n\n');

P=( (1/(2\*q\_p^2)) \* C2/C4 -1) + sqrt( ((1/(2\*q\_p^2)) \* C2/C4 -1)^2 -1 );

R1=1/(2\*pi\*f\_p\*sqrt(P\*C2\*C4));

R3=P\*R1;

GSP=q\_p\*sqrt(C2/(P\*C4));

if Kin==0 | 1-Kin <=0;

R11=R1;

R12='inf';

K=1;

else

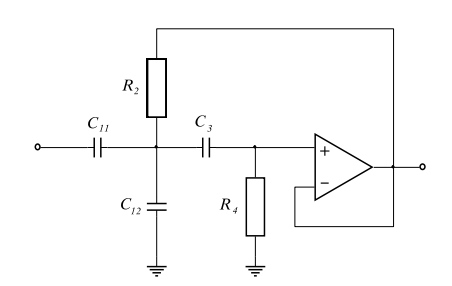
K=Kin;

R11=R1\*K;

R12=R1/(1-K);

end

* HPLQ (Sallen-Key, Q<2)



* + Code Matlab :

Une image contenant texte

Description générée automatiquement

Une image contenant texte

Description générée automatiquement

* + Code important pour python :

C1=(C11+C12);

P=((q\_p^2)\*(2+(C3/C1)+(C1/C3)));

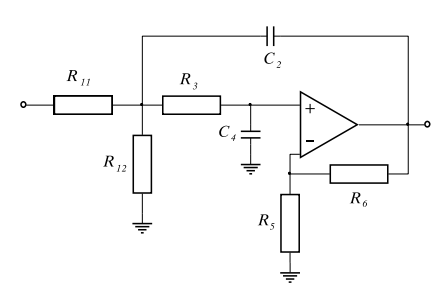
R2=(1/(2\*pi\*f\_p\*sqrt(P\*C1\*C3)));

R4=(P\*R2);

GSP=(q\_p\*sqrt(P\*C3/C1));

K=(C11/C1);

* LPMQ (Sallen-Key, Q<5)



* + Code Matlab :

Une image contenant texte

Description générée automatiquement

Une image contenant texte

Description générée automatiquement

* + Code important pour python :

P=(C2/C4)/(36\*q\_p^2)\*(sqrt(1+12\*q\_p^2\*(1+C4/C2))+1)^2;

R1=1/(2\*pi\*f\_p\*sqrt(P\*C2\*C4));

R3=P\*R1;

if R5==0

R5=1e4;

end;

R6=R5\*(C4/C2\*(1+P)-sqrt(P\*C4/C2)/q\_p);

K0=1+R6/R5;

GSP=q\_p\*K0^2\*sqrt(C2/(P\*C4));

if (Kin==0 | K0-Kin<=0)

R11=R1;

R12=1e90;

K=K0;

else

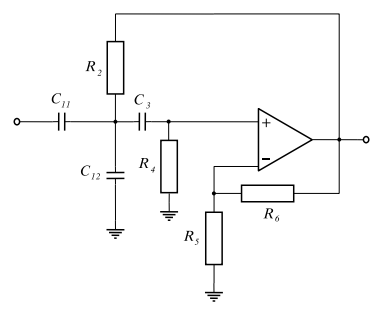
K=Kin;

R11=K0/K\*R1;

R12=K0/(K0-K)\*R1;

end;

* HPMQ (Sallen-Key, Q<5)



* + Code Matlab :

Une image contenant texte

Description générée automatiquement

Une image contenant texte

Description générée automatiquement

* + Code important pour python :

C1=C11+C12;

P=(C1/C3)/(4\*q\_p^2)\*(sqrt(1+12\*q\_p^2\*(1+C3/C1))-1)^2;

R2=1/(2\*pi\*f\_p\*sqrt(P\*C1\*C3));

R4=P\*R2;

if R5==0

R5=1e4;

end;

R6=R5\*(1/P\*(1+C1/C3)-sqrt(C1/(P\*C3))/q\_p);

K=C11/C1\*(1+R6/R5);

GSP=q\_p\*(1+R6/R5)^2\*sqrt((P\*C3)/C1);

* HPLQ (Sallen-Key, Q<2)
  + Code Matlab :
  + Code important pour python :
* HPLQ (Sallen-Key, Q<2)
  + Code Matlab :
  + Code important pour python :
* HPLQ (Sallen-Key, Q<2)
  + Code Matlab :
  + Code important pour python :