

---

## Table of Contents

3.2.1 Filtrage profil de production PV .....	1
3.2.2 Potentiel d'hybridation en puissance .....	3
3.2.3 Puissance fournie .....	4
3.2.4 Energie à stocker .....	5
3.3 Energie cinétique .....	5
3.4 Modélisation et commande vectorielle du moteur .....	7
3.5 Contrôle-commande du moteur .....	11
3.5.1 Définition des spécifications .....	11
3.5.2 Correcteurs .....	11

### 3.2.1 Filtrage profil de production PV

```
clear all;
close all;

%load data
load("Centrale250kW_16_sept.mat") ;
Ppv = Grandeur_lapalud_16sept( :,4) ;

%filtre
Fe = 0.0069 ; % fréquence d'échantillonnage
M = 477 ;
B=fir1(M,2*Fe,"low",hamming(M+1)) ;
PV_filtre=filtfilt(B,1,Ppv);

t=(1:1:1440);

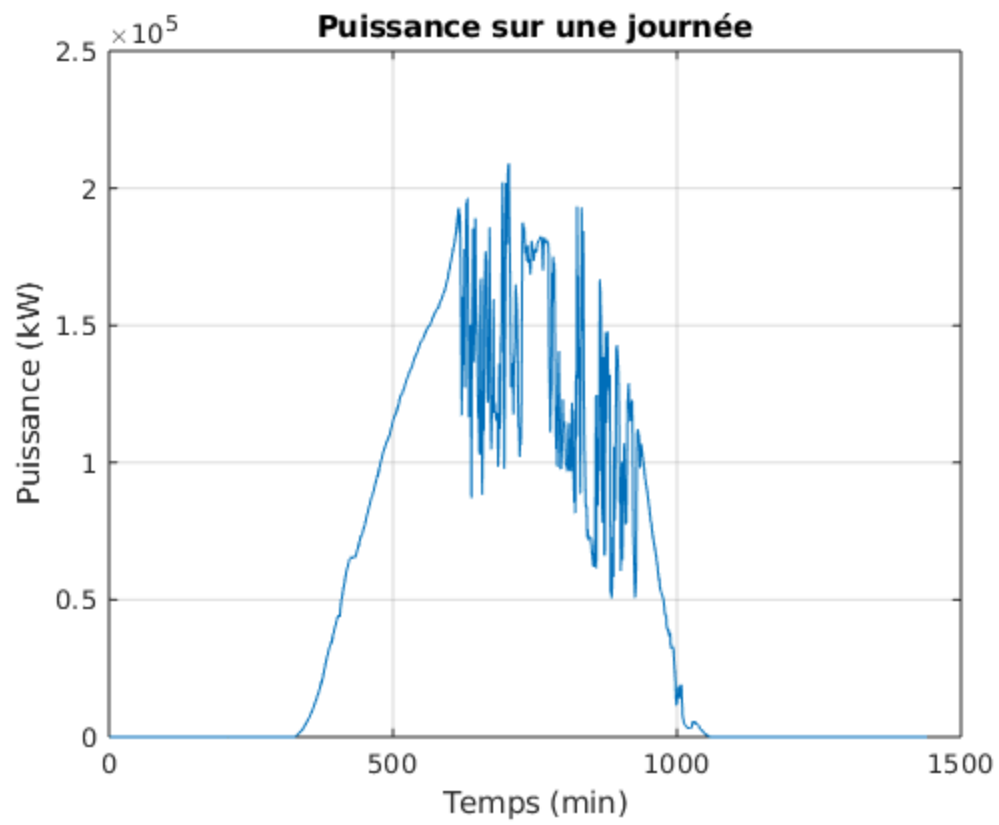
%plot
figure("Name","Ppv");
plot(t,Ppv);
title("Puissance sur une journée");
xlabel("Temps (min)");
ylabel("Puissance (kW)");
grid on;

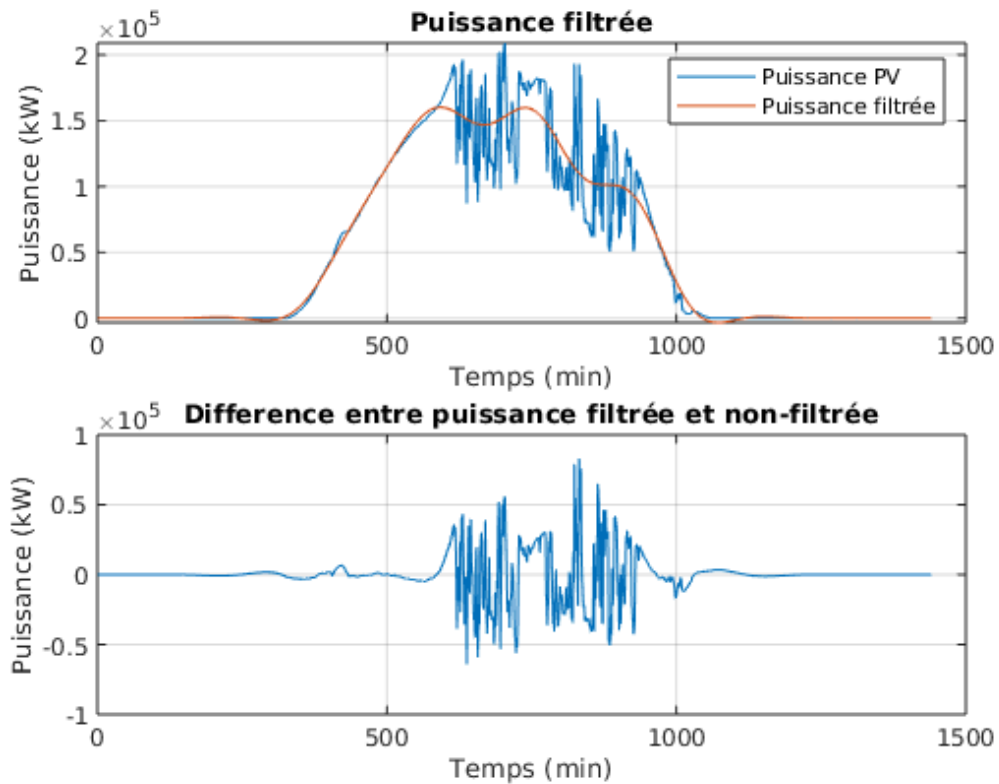
figure("Name","Analyse photovoltaïque");
subplot(2,1,1);
plot(t,Ppv,t,PV_filtre);
title("Puissance filtrée");
xlabel("Temps (min)");
ylabel("Puissance (kW)");
grid on;
legend("Puissance PV","Puissance filtrée");

subplot(2,1,2);
plot(t,Ppv-PV_filtre);
title("Difference entre puissance filtrée et non-filtrée");
xlabel("Temps (min)");
```

---

```
ylabel("Puissance (kW)");  
grid on;
```





### 3.2.2 Potentiel d'hybridation en puissance

```
%Calcul PHP
Pmoy = mean(Ppv);
Pmax = max(Ppv);
PHP = 1 - (Pmoy/Pmax);

%ecart type, dist norm (indicateur)
ecart_type = std(Ppv);
yEc = normpdf(t,Pmoy,ecart_type);

%puissance moyenne sur la journée
tDay = t(round(0.25*1440):round(0.75*1440));
ppvDay = Ppv(round(0.25*1440):round(0.75*1440));
PmoyDay = mean(ppvDay)
PmaxDay = max(ppvDay);
PHP_day = 1 - (PmoyDay/PmaxDay)

PmoyDay =

    1.0013e+05

PHP_day =
```

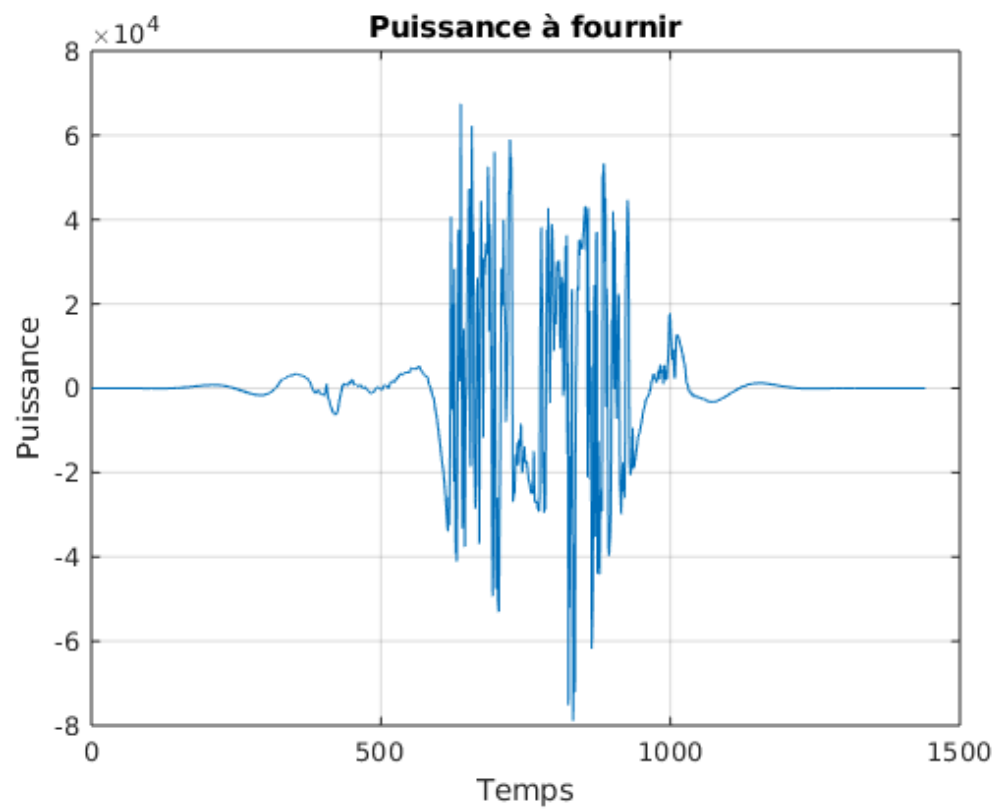
---

0.5208

### 3.2.3 Puissance fournie

```
%Puissance à fournir
Psse = PV_filtre-Ppv;
Pvi = zeros(1440,1);
for i=1:1:1440
    if Psse(i) <= 0
        Pvi(i) = 0.95*Psse(i);
    end
    if Psse(i) > 0
        Pvi(i) = Psse(i)/0.95;
    end
end

figure("Name","Puissance à fournir");
plot(t,Pvi);
title("Puissance à fournir");
xlabel("Temps");
ylabel("Puissance");
grid on;
```



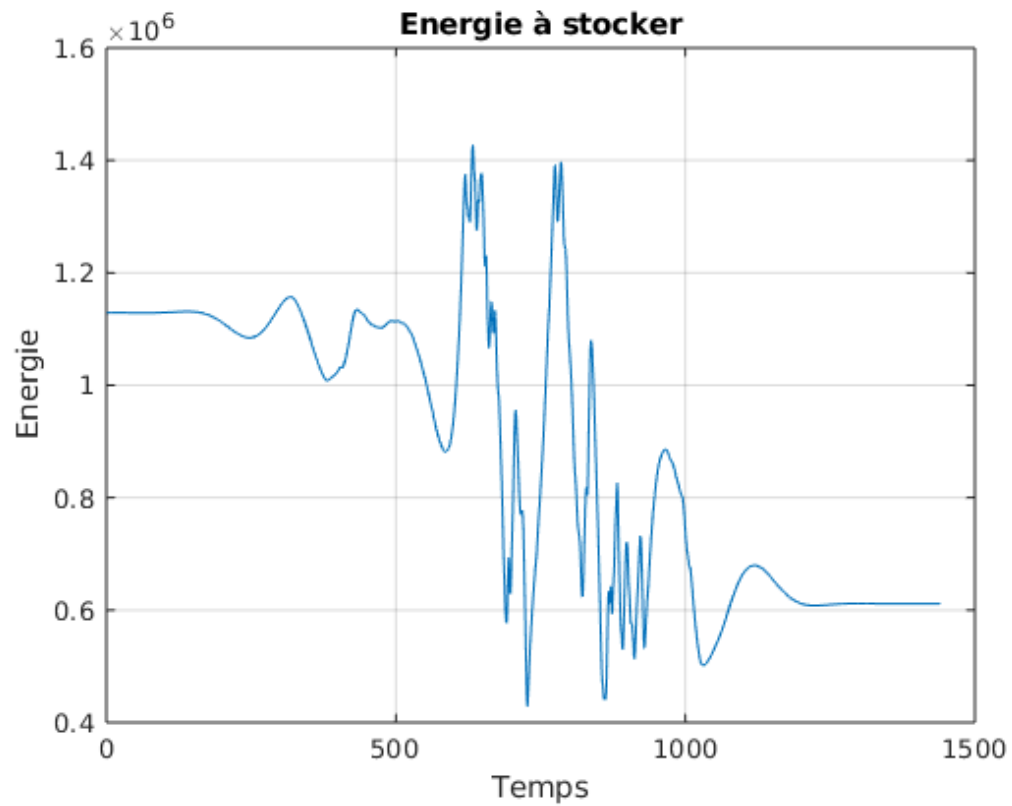
---

## 3.2.4 Energie à stocker

```
%Energie stockée dans le volant
Evi = -cumtrapz(Pvi);

%en utilisant l'énergie utile
Eutil = max(Evi) - min(Evi); %Watt*min
Evi = Evi - min(Evi)+0.3*(Eutil/0.7); %Watt*min

%plot
figure("Name","Energie à stocker");
plot(t,Evi);
title("Energie à stocker");
xlabel("Temps");
ylabel("Energie");
grid on;
```



## 3.3 Energie cinétique

```
DoDmax = 0.7;
OmegaMin = 2760; %tour/min
Capacity = Eutil/DoDmax;
Emin=0.3*Capacity;
OmegaMax = OmegaMin/sqrt(0.3); %tour/min
```

---

```

OmegaMaxconvertie=OmegaMax*2*pi/60; %rad/s
OmegaMinconvertie=OmegaMin*2*pi/60; %rad/s
Eutilconvertie = Eutil*60; %Joule

ResElas = 1800*10^6; %Pa
MasseVol = 7800; % kg/m^3

Jrad = 2*Eutilconvertie/(OmegaMaxconvertie*OmegaMaxconvertie-
OmegaMinconvertie*OmegaMinconvertie)
J = 2*Eutilconvertie/(OmegaMax*OmegaMax-OmegaMin*OmegaMin);

%rayon du volant
R=1.5;

%Masse du volant
Masse = 2*J/(R^2)
%longueur
L=Masse/(MasseVol*pi*R^2)

%R1 = sqrt(ResElas/(MasseVol*OmegaMaxconvertie));
GammaMax=[];
Pmax=[];
for i=1:1:13001*(2*pi/60)
    GammaMax(i)=240;
    Pmax(i)= 70000;
end
Omega1=(1:1:13001*(2*pi/60));
Omega2=(1:1:13001*(2*pi/60));
Gamma = Pmax(1:13001*(2*pi/60)) ./ Omega2;

% plot des limites physiques
figure("Name","Points de fonctionnement");
plot(Omega1(1:292),GammaMax(1:292),Omega2(292:1331),Gamma(292:1331),Omega1(1:292),
GammaMax(1:292),Omega2(292:1331),-Gamma(292:1331));
title("Points de fonctionnement");
xlabel("Vitesse");
ylabel("Couple");
hold on

%Calcul des points de fonctionnement
OmegaInstantane = sqrt(2*Evi*60/Jrad);
GammaUtilPtf = Pvi./OmegaInstantane;
plot(OmegaInstantane,GammaUtilPtf,"+");
grid on;

Jrad =

    615.1715

Masse =

    5.9965

```

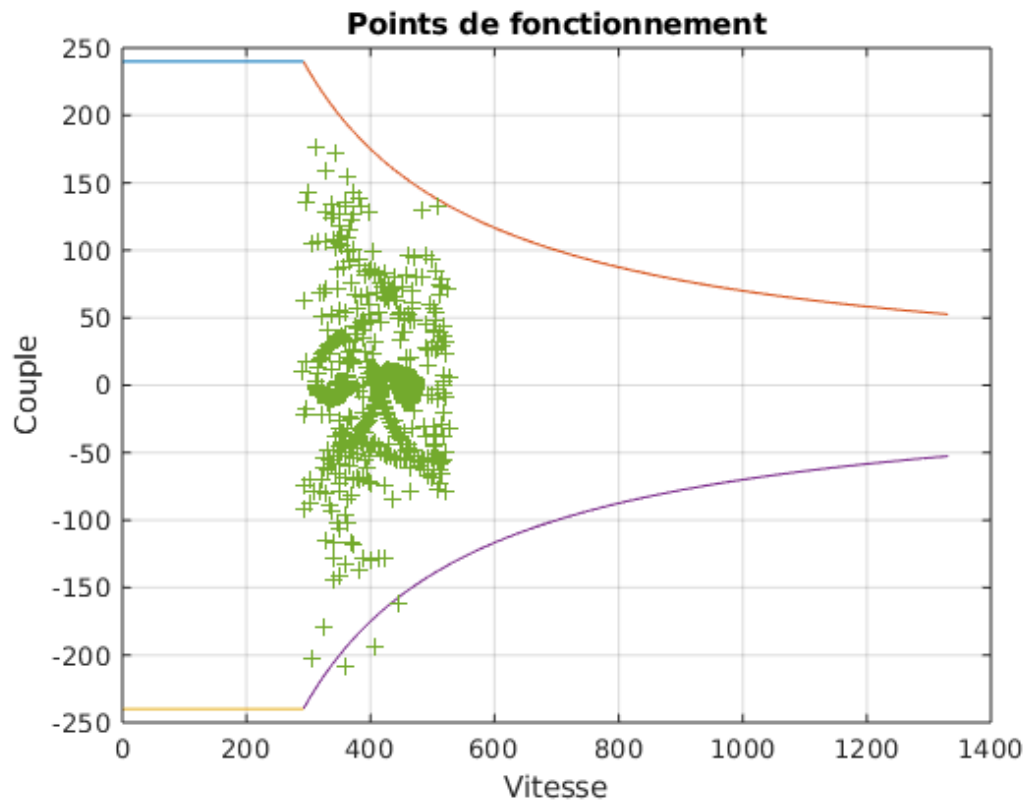
---

---

$L =$

$1.0876e-04$

*Warning: Integer operands are required for colon operator when used as index.*



## 3.4 Modélisation et commande vectorielle du moteur

```
% Evolution de la vitesse du moteur
figure;
plot((1:1440),OmegaInstantane);
title("Evolution de la vitesse du moteur");
xlabel("Temps (min)");
ylabel("Vitesse (rad/s)");
grid on;

% Evolution du couple moteur
figure;
plot((1:1440),GammaUtilPtf);
title("Evolution du couple moteur ramené sur l'arbre moteur");
xlabel("Temps (min)");
```

---

```

ylabel("Couple (N.m)");
grid on;

% Caractéristiques mécaniques du moteur
p=4;
Rs=0.0952; %Ohm
Phifsd=1.17; %rad/s
Ld=2.59*0.001; %H
Lq=2.62*0.001; %H

% calcul vitesse du moteur
OmegaInstantane = sqrt(2*Evi*60/Jrad);

% calcul courant statorique
isd = zeros(1440,1);
isq = GammaUtilPtf/(p*Phifsd);
Is_eff = abs(isq/sqrt(3));
Is = [ isd isq ];

% calcul pulsation
omega_r=zeros(1,1440);

% calcul pulsation statorique
omega_s = p*OmegaInstantane;

% tension simple statorique
Vsd =omega_s.*isq*Ld;
Vsqr = Rs*isq +omega_s*Phifsd;
Vs = [Vsd Vsqr];

figure("Name","Vérification du moteur");
subplot(4,1,1);
plot(t,OmegaInstantane);
title("Vitesse du moteur");
xlabel("Temps");
ylabel("Vitesse");
grid on;

subplot(4,1,2);
plot(t,Vsd,t,Vsqr);
title("Tension Vsd et Vsqr");
xlabel("Temps");
ylabel("Tension");
legend("Vsd","Vsqr");
grid on;

subplot(4,1,3);
plot(t,Is_eff);
title("Courant effectif");
xlabel("Temps");
ylabel("Courant");
grid on;

subplot(4,1,4);

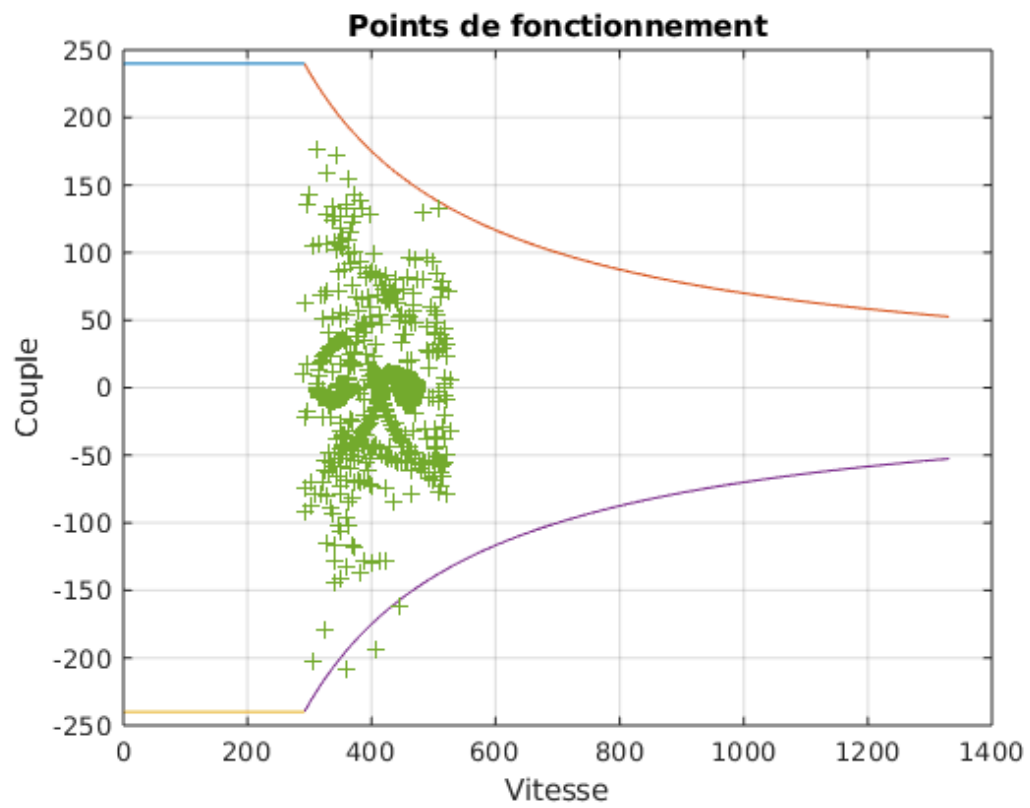
```

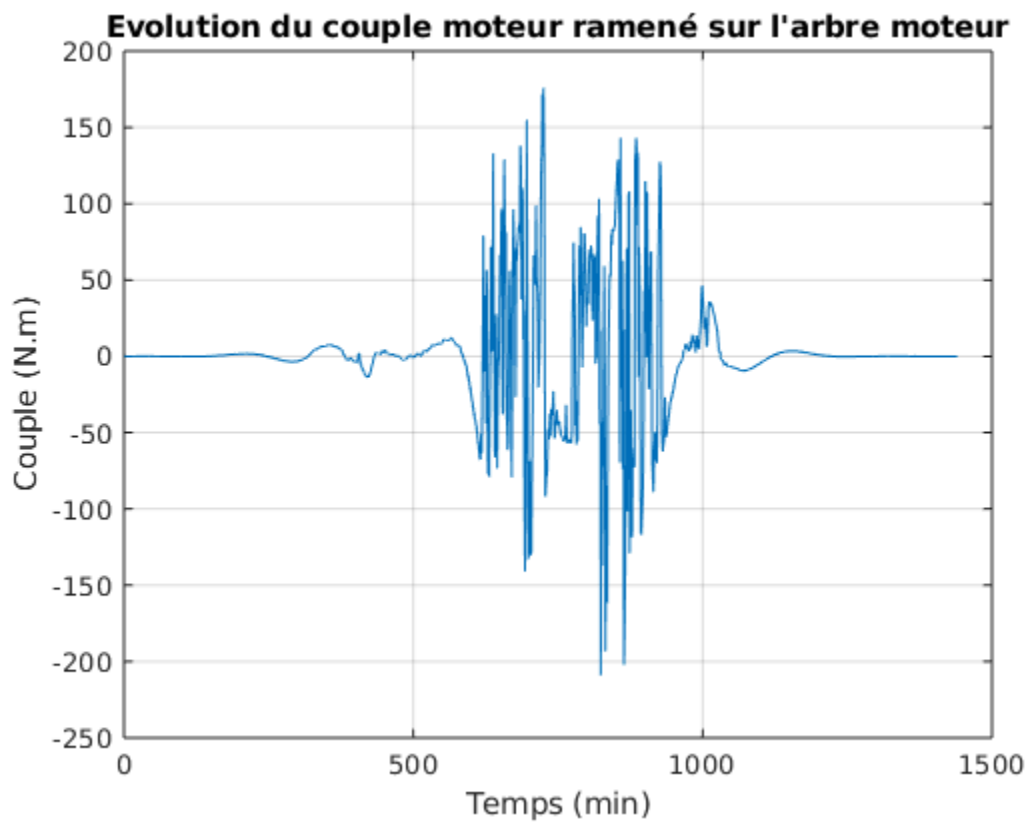
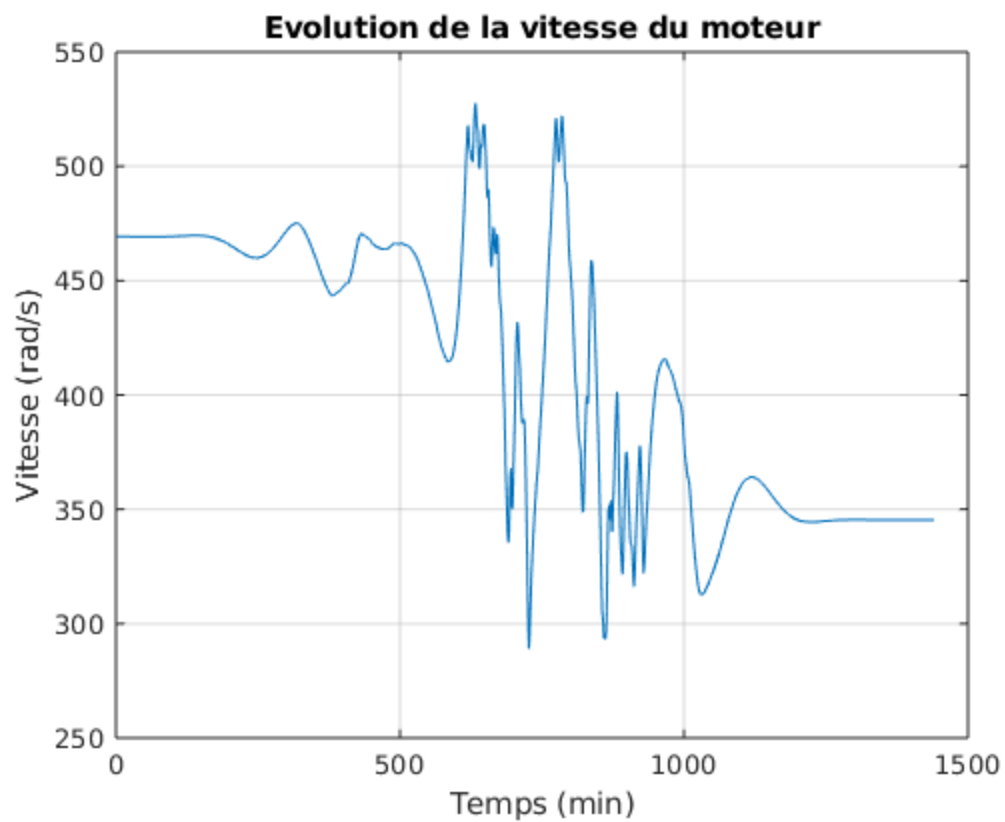
---

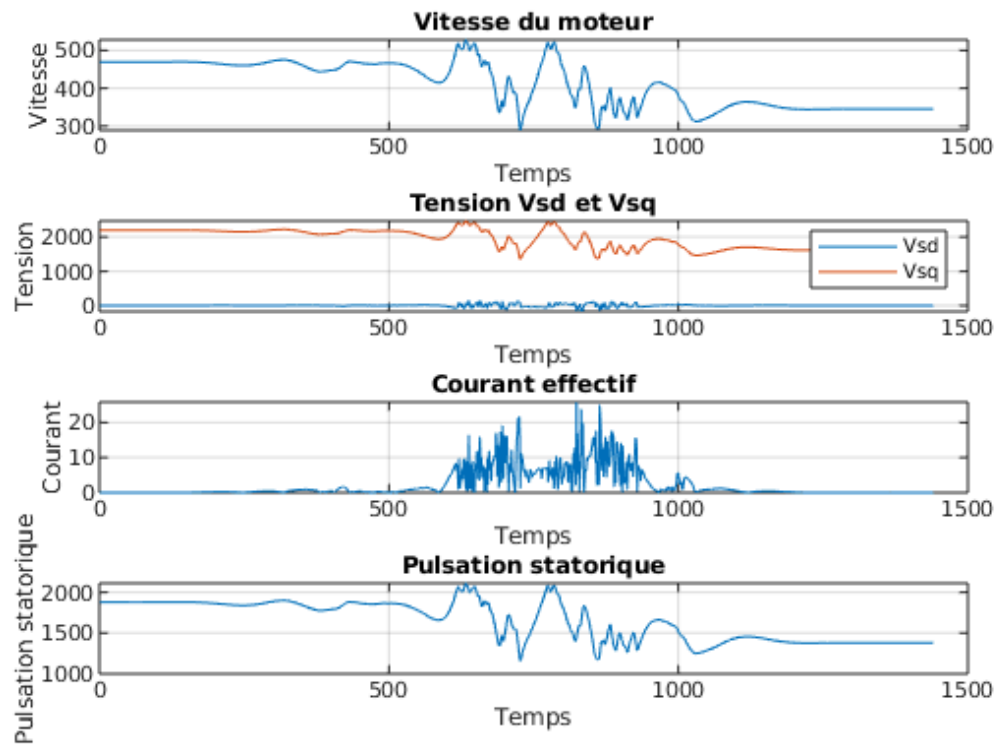


---

```
plot(t,omega_s);  
title("Pulsation statorique");  
xlabel("Temps");  
ylabel("Pulsation statorique");  
grid on;
```







## 3.5 Contrôle-commande du moteur

### 3.5.1 Définition des spécifications

```
ksi = 1.0;
% Temps de réponse entre trmax et trmin
tr_max = 3*Ld/Rs;
Tmli = 1/10000;
tr_min = 15*Tmli;
tr = (tr_max-tr_min)/2;
omega_n = 3/tr;
```

### 3.5.2 Correcteurs

```
% boucle intérieure
% Calcul des gains des PI des boucles intérieures
kid = Ld*omega_n*omega_n;
kpd = (2*ksi*kid)/omega_n - Rs;

kiq = Lq*omega_n*omega_n;
kpq = (2*ksi*kiq)/omega_n - Rs;

omega_fonct = p*OmegaMaxconvertie;
```

---

```
% boucle exterieure
% moins rapide que boucle intérieure, tresp_ext = 50*tres_int
omega_n_ext = omega_n / 50;

ki = Jrad*omega_n_ext*omega_n_ext;
kp = (2*ksi*ki)/omega_n_ext;
```

*Published with MATLAB® R2019b*