

تشریح برنامه و نحوه اجرای آن

با سلام

در فولدر ارسالی برای شما ۵ mfile و ۲ فایل سیمپولینک به شرح زیر وجود دارد.

۱- فایل سیمپولینک MPC_BOOST_CONVERTER_CONSTRAINED.mdl سیستم کنترل مبدل

بوست را در حالت محدودیت دار پیاده سازی می کند.

۲- فایل MPC_CONTROLLER_CONSTRAINED.m کنترل کننده پیش بین را در حالت محدودیت

دار پیاده سازی می کند که توسط سیمپولینک ۱ فراخوانی خواهد شد.

۳- فایل BOOST_CONVERTER_MODEL.m مدل ریاضی مبدل بوست را شامل می شود که به صورت

یک فایل در سیمپولینک فراخوانی می شود.

۴- فایل MPC_OBJ.m تابع هزینه را برای حالت محدودیت دار پیاده سازی می کند.

۵- فایل State_Constrained.m قیدهای حالت را برای کنترل کننده پیش بین محدودیت دار پیاده سازی

می کند که توسط تابع کمینه ساز فراخوانی خواهد شد.

۶- فایل سیمپولینک MPC_BOOST_CONVERTER_Unconstrained.mdl کنترل پیش بین مبدل

بوست را در حالت بدون قید پیاده سازی می کند.

۷- فایل MPC_CONTROLLER_Unconstrained.m کنترل کننده پیش بین بدون قید را پیاده سازی

می کند.

توجه شود بنده تمام شبیه سازی ها را در نرم افزار متلب ۲۰۱۱ نسخه b انجام داده ام. شما جهت مشاهده نتایج

کافی است فایل های سیمپولینک را اجرا نمایید و نیاز به اجرای هیچکدام از mfile ها نخواهید داشت. این فایل ها

به طور خودکار و بر حسب نیاز توسط سیمپولینک فراخوانی خواهند شد. مقاله شما هیچ شبیه سازی انجام نداده

و تمام کارهای آن تجربی بوده است. بنابراین تفاوت های که ممکن است در شکل ها وجود داشته باشد ناشی از

ساده سازی های مدل سازی و غیرتجربی بودن نتایج است. به هر حال سیستم کنترل به خوبی عمل خواهد کرد.

تشریح برنامه و نحوه اجرای آن

کنترل کننده پیش بین در حالت بدون محدودیت:

مبدل بوست با معادلات حالت

$$\dot{x}(t) = (u(t)A_{c1} + (1-u(t))A_{c2})x(t) + (u(t)B_{c1} + (1-u(t))B_{c2})v(t)$$

$$A_{c1} = \begin{bmatrix} -\frac{R_{no}}{L} & 0 \\ 0 & -\frac{1}{RC} \end{bmatrix} \quad A_{c2} = \begin{bmatrix} 0 & \frac{-1}{L} \\ \frac{1}{C} & \frac{-1}{RC} \end{bmatrix}$$

$$B_{c1} = \begin{bmatrix} \frac{1}{L} & 0 \\ 0 & 0 \end{bmatrix} \quad B_{c2} = \begin{bmatrix} \frac{1}{L} & \frac{-1}{L} \\ 0 & 0 \end{bmatrix}$$

با تعریف ماتریس های

$$B_c = (B_{c1} - B_{c2})v$$

$$G_c = A_{c1} - A_{c2}$$

معادلات گسسته سیستم به صورت زیر حاصل خواهد شد.

$$x(k+1) = A_2x(k) + (B + Gx(k))u(k) + B_2v(k)$$

$$A_2 = hA_{c2} + I, \quad B = hB_c, \quad G = hG_c, \quad B_2 = hB_{c2}$$

مطابق آنچه در مقاله ذکر شده است کنترل کننده پیش بین به شرح زیر است.

ابتدا مقدار r_I توسط کنترل کننده PID محاسبه خواهد شد. سپس مقدار $x_1^0 = r_I$ و مقدار x_2^0 نیز از رابطه

$$x_2^0 = \frac{-\alpha_1 \pm \sqrt{\alpha_1^2 - 4\alpha_0\alpha_2}}{2\alpha_2}$$

محاسبه خواهد شد. از بین این دو مقدار یکی انتخاب می شود که u^0 متناظر با آن که از رابطه زیر محاسبه

می شود شدنی باشد.

تشریح برنامه و نحوه اجرای آن

$$u^0 = \frac{(1-a_{21})r_I - a_{22}x_2^0 - b_{21}v_g - b_{22}v_D}{b_1 + g_1r_I + g_2x_2^0}$$

همچنین در روابط بالا

$$\alpha_0 = -(a_{23}r_I + b_{23}v_g + b_{24}v_d)(b_1 + g_1r_I) - ((1-a_{21})r_I - b_{21}v_g - b_{22}v_d)(b_2 + g_3r_I)$$

$$\alpha_1 = (b_1 + g_1r_I)(1-a_{24}) - (a_{23}r_I + b_{23}v_g + b_{24}v_d)g_2 + a_{22}(b_2 + g_3r_I)$$

$$-((1-a_{21})r_I - b_{21}v_g - b_{22}v_d)g_4$$

$$\alpha_2 = (1-a_{24})g_2 + a_{22}g_4$$

از روابط بالا بردار $x^0 = [x_1^0 \quad x_2^0]^T$ محاسبه خواهد شد. سپس ماتریس T از رابطه

$$T = B + Gx^0$$

محاسبه شده و بردار e_x نیز از رابطه زیر محاسبه خواهد شد.

$$e_x = x(k) - x^0$$

با انتخاب ماتریس P_c و ρ به صورت

$$P_c = \begin{bmatrix} 0.0016 & 0 \\ 0 & 0.001 \end{bmatrix}$$

$$\rho = 0.01$$

سیگنال کنترلی از رابطه

$$u_{uc}^* = \frac{-c_1(e_x) + 2\rho u^0}{2(c_2(e_x) + \rho)}$$

$$c_1(e_x) = 2(T + Ge_x)^T P_c (A_2 e_x - Tu^0)$$

$$c_2(e_x) = (T + Ge_x)^T P_c (T + Ge_x)$$

محاسبه خواهد شد. این روابط در فایل MPC_CONTROLLER_Unconstrained.m پیاده‌سازی شده‌اند.

تشریح برنامه و نحوه اجرای آن

```
function Do=MPC_CONTROLLER_Unconstrained(In)
```

این تابع ورودی ها را از سیمپولینک دریافت کرده و سیگنال کنترلی را محاسبه می کند.

```
%-----  
%-----  
rI=In(1);  
iL=In(2);  
vc=In(3);  
R=In(4);
```

تفکیک ورودی ها

```
x=[iL;vc];  
C=1880e-6;  
L=3e-3;  
Ron=0.08;  
vg=67;  
vd=0.67;  
h=1e-4;  
Ac1=[-Ron/L,0;0,-1/R/C];  
Ac2=[0,-1/L;1/C,-1/R/C];  
Bc1=[1/L,0;0,0];  
Bc2=[1/L,-1/L;0,0];  
v=[vg;vd];  
Bc=(Bc1-Bc2)*v;  
Gc=Ac1-Ac2;  
A2=h*Ac2+eye(2);  
B=h*Bc;  
G=h*Gc;  
B2=h*Bc2;
```

در خطوط بالا ماتریس های سیستم تعریف شده اند

```
%-----  
%-----  
x1_0=rI;  
a2_1=A2(1,1);  
a2_2=A2(1,2);  
a2_3=A2(2,1);  
a2_4=A2(2,2);  
b1=B(1,1);  
b2=B(2,1);  
b2_1=B2(1,1);  
b2_2=B2(1,2);  
b2_3=B2(2,1);  
b2_4=B2(2,2);  
g1=G(1,1);  
g2=G(1,2);  
g3=G(2,1);  
g4=G(2,2);
```

در این روابط درایه های ماتریس ها تفکیک شده اند

```
W1=g2*(1-a2_4)+g4*a2_2;  
W2=(b1+g1*rI)*(1-a2_4)+g2*(-a2_3*rI-b2_3*vg-b2_4*vd)+(b2+g3*rI)*a2_2-g4*((1-a2_1)*rI-b2_1*vg-b2_2*vd);  
W3=(b1+g1*rI)*(-a2_3*rI-b2_3*vg-b2_4*vd)-(b2+g3*rI)*((1-a2_1)*rI-b2_1*vg-b2_2*vd);  
Q=roots([W1 W2 W3]);
```

در این روابط مقدار x_2^0 تعیین شده است.

```
if (abs(Q(1)-vc)>abs(Q(2)-vc))
```

تشریح برنامه و نحوه اجرای آن

```

u0=((1-a2_1)*rI-a2_2*Q(2)-b2_1*vg-b2_2*vd)/(b1+g1*rI+Q(2)*g2);
x2_0=Q(2);
else
u0=((1-a2_1)*rI-a2_2*Q(1)-b2_1*vg-b2_2*vd)/(b1+g1*rI+Q(1)*g2);
x2_0=Q(1);
end

```

در این خطوط u^0 شدنی انتخاب شده است

```

rho=0.01;
ex=x-[x1_0;x2_0];
Pc=[0.0016,0;0,0.001];
T=B+G*[x1_0;x2_0];
c1_ex=2*(T+G*ex) .* Pc*(A2*ex-T*u0);
c2_ex=(T+G*ex) .* Pc*(T+G*ex);
u_star=(-c1_ex+2*rho*u0)/2/(c2_ex+rho);
Do=u_star;

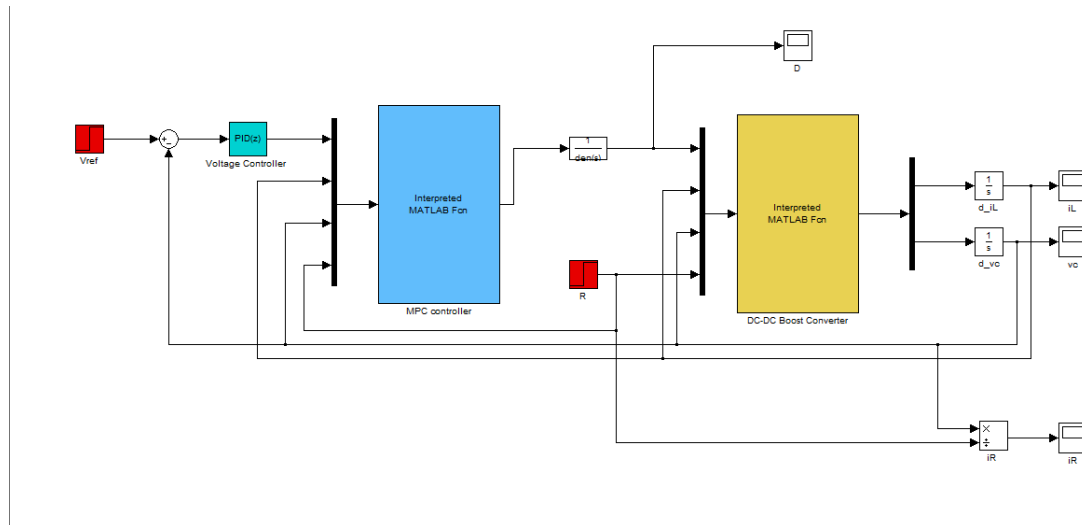
```

در این روابط مطابق مقاله سیگنال کنترلی تعیین شده است.

حال فایل سیمولینک MPC_BOOST_CONVERTER_Unconstrained.mdl را تشریح می کنیم. این

فایل کنترل کننده ذکر شده در روابط بالا را بر روی مدل سیستم پیاده سازی می کند. شکل زیر این فایل را

نشان می دهد.



در این شکل بلوک آبی رنگ تابع MPC_CONTROLLER_Unconstrained.m را فراخوانی می کند. همچنین

بلوک زرد رنگ تابع زیر را که مدل سیستم است فراخوانی می کند.

```

function out=BOOST_CONVERTER_MODEL(In)
u=In(1);
iL=In(2);
vc=In(3);
R=In(4);
C=1880e-6;

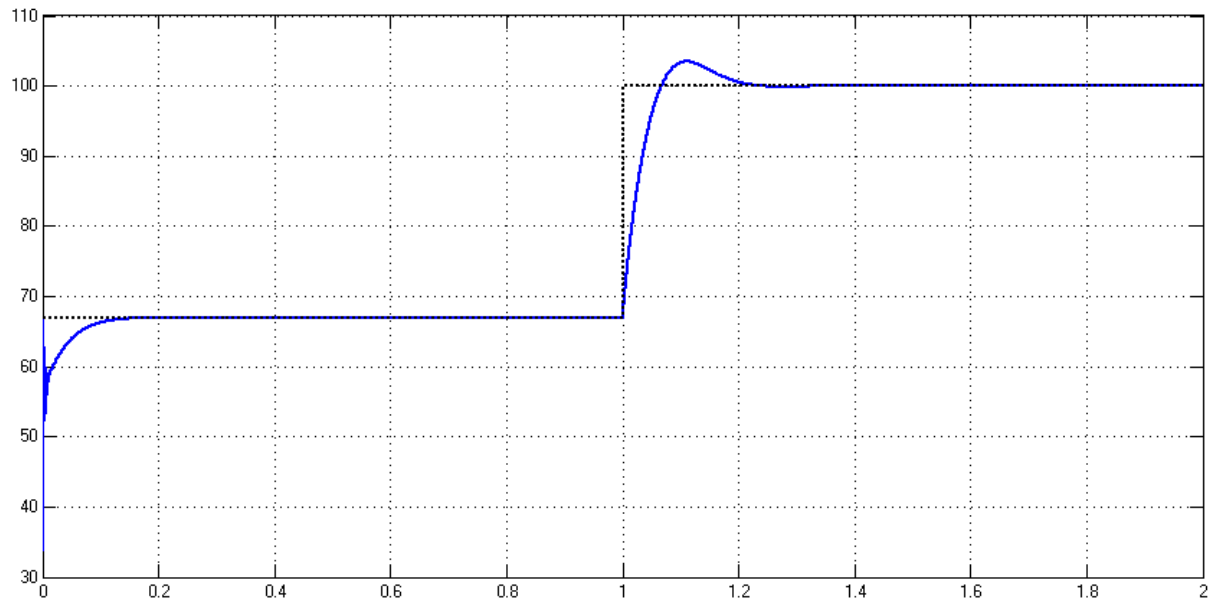
```

تشریح برنامه و نحوه اجرای آن

```
L=3e-3;  
Ron=0.08;  
vg=67;  
vd=0.67;  
Ac1=[-Ron/L,0;0,-1/R/C];  
Ac2=[0,-1/L;1/C,-1/R/C];  
Bc1=[1/L,0;0,0];  
Bc2=[1/L,-1/L;0,0];  
v=[vg;vd];  
Bc=(Bc1-Bc2)*v;  
Gc=Ac1-Ac2;  
x=[iL;vc];  
d_x=Ac2*x+(Bc+Gc*x)*u+Bc2*v;  
out=[d_x(1);d_x(2)];
```

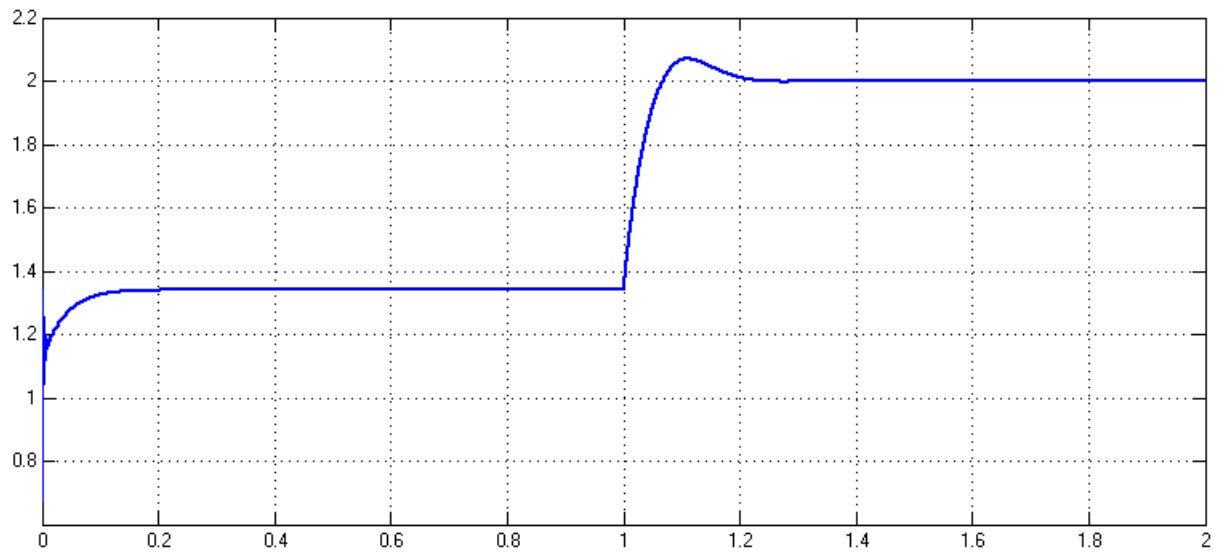
در تابع بالا مقادیر مشتقات محاسبه شده اند و عمل انتگرال گیری بیرون از بلوک انجام می شود.

مقدار بار در این شبیه سازی ۵۰ اهم در نظر گرفته شده است که البته به دلخواه می توان آن را تغییر داد. در صورتی که فرض کنیم ۱ ثانیه ولتاژ مرجع ۶۷ ولت بوده و سپس به ۱۰۰ ولت تغییر می کند نتایج شبیه سازی به صورت زیر حاصل می شود.

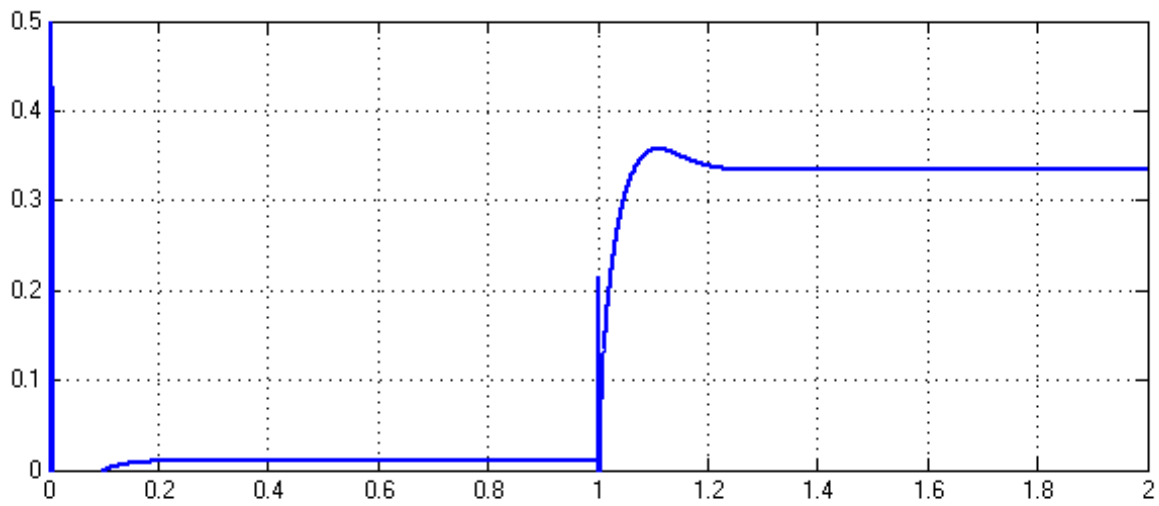


ولتاژ خروجی و مقدار مرجع

تشریح برنامه و نحوه اجرای آن



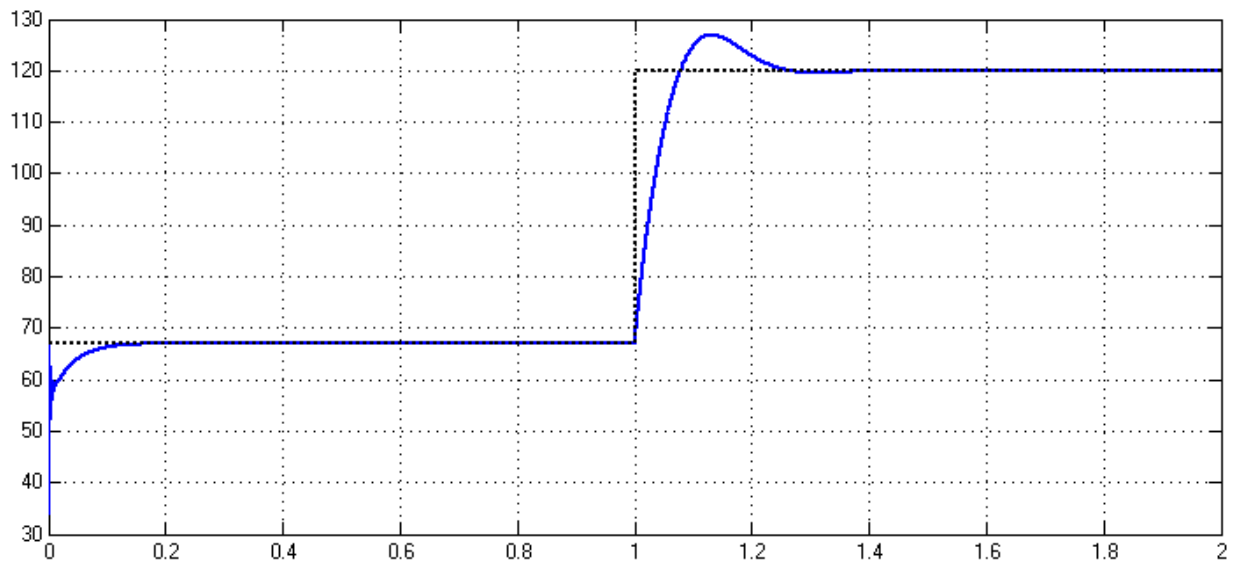
جریان بار



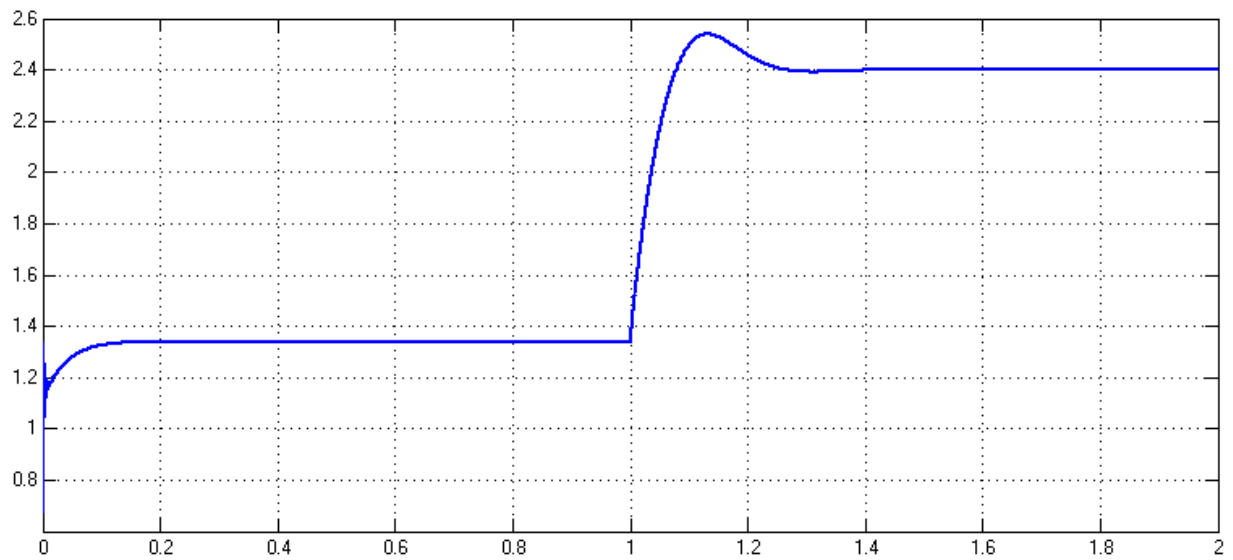
سیگنال کنترلی

در صورتی که ولتاژ مرجع از ۶۷ به ۱۲۰ ولت تغییر کند نتایج شبیه سازی به صورت زیر حاصل می شود.

تشریح برنامه و نحوه اجرای آن



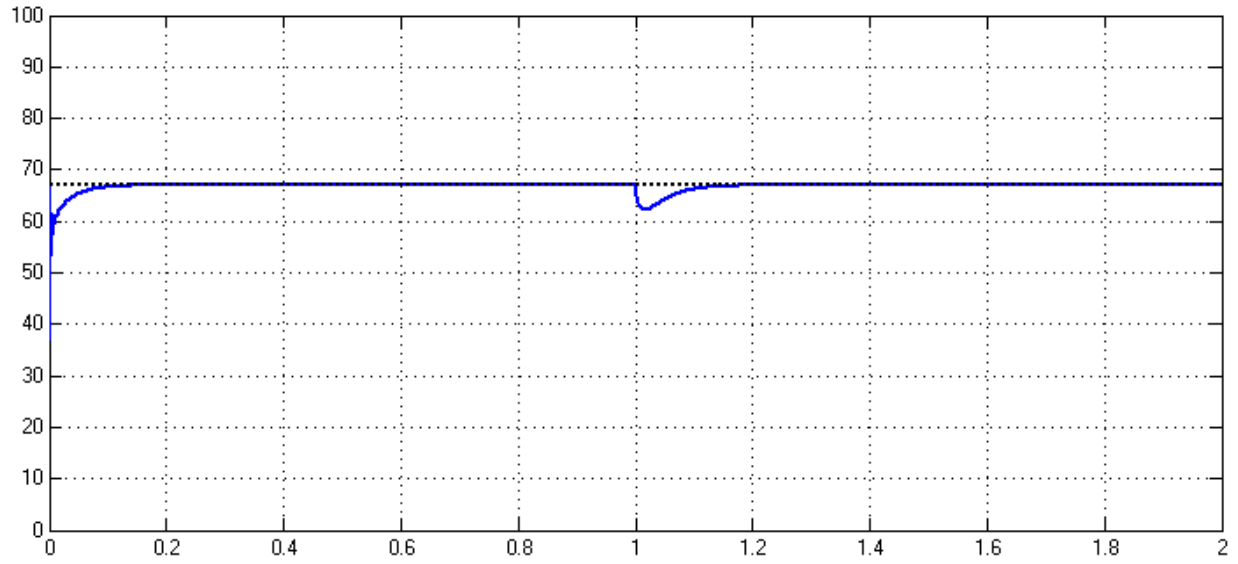
ولتاژ خروجی



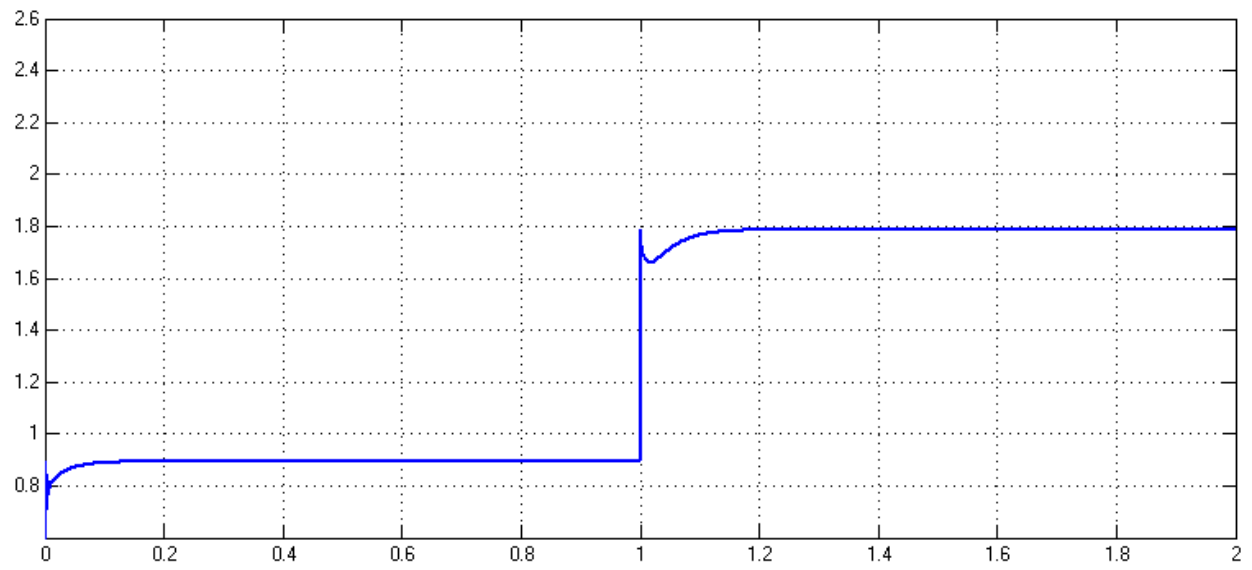
جریان بار

در صورتی که ولتاژ مرجع ۶۷ بوده و مقاومت بار از ۷۵ به ۳۷،۵ تغییر کند نتایج شبیه سازی به صورت شکل های زیر خواهد شد.

تشریح برنامه و نحوه اجرای آن



ولتاژ خروجی



جریان بار

تشریح برنامه و نحوه اجرای آن

کنترل کننده پیش بین در حالت با محدودیت:

در این حالت سیگنال کنترلی باید با در نظر گرفتن محدودیت های زیر محاسبه شود

$$D_{\min} \leq D \leq D_{\max}$$
$$0 \leq i_L(k) \leq 5$$
$$0 \leq v_c(k) \leq 150$$

محاسبه شود. برای این منظور از توابع زیر کمک می گیریم.

```
function J=MPC_OBJ(ud)
```

تابع هزینه برای کنترل کننده پیش بین با محدودیت

```
global x rI R
C=1880e-6;
L=3e-3;
Ron=0.08;
vg=67;
vd=0.67;
h=1e-4;
Ac1=[-Ron/L,0;0,-1/R/C];
Ac2=[0,-1/L;1/C,-1/R/C];
Bc1=[1/L,0;0,0];
Bc2=[1/L,-1/L;0,0];

v=[vg;vd];
Bc=(Bc1-Bc2)*v;
Gc=Ac1-Ac2;
A2=h*Ac2+eye(2);
B=h*Bc;
G=h*Gc;
B2=h*Bc2;
x_next=A2*x+(B+G*x)*ud+B2*v;
y_next=x_next(1);
J=0.5*(rI-y_next)^2;
```

تعریف ماتریس های سیستم

سیگنال کنترلی به گونه ای محاسبه می شود که جریان مبدل برابر r_I شود.

```
function [c,ceq] = State_Constrained(ud)
```

این تابع قید روی حالت های سیستم را برای مسئله بهینه سازی تعریف می کند.

```
global x R
C=1880e-6;
L=3e-3;
Ron=0.08;
vg=67;
vd=0.67;
h=1e-4;
Ac1=[-Ron/L,0;0,-1/R/C];
Ac2=[0,-1/L;1/C,-1/R/C];
Bc1=[1/L,0;0,0];
Bc2=[1/L,-1/L;0,0];
```

تشریح برنامه و نحوه اجرای آن

```
v=[vg;vd];
Bc=(Bc1-Bc2)*v;
Gc=Ac1-Ac2;
A2=h*Ac2+eye(2);
B=h*Bc;
G=h*Gc;
B2=h*Bc2;
x_next=A2*x+(B+G*x)*ud+B2*v;
c=x_next-[5;150];
ceq=[];
```

```
function D=MPC_CONTROLLER_Constrained(In)
```

این تابع کنترل کننده پیش بین با محدودیت را پیاده سازی می کند

```
global x rI R
```

```
rI=In(1);
```

```
iL=In(2);
```

```
vc=In(3);
```

```
R=In(4);
```

```
x=[iL;vc];
```

```
options = optimset('algorithm','interior-point');
```

```
D=fmincon(@MPC_OBJ,0,[1;-1],[1;0],[[],[],[],[]],@State_Constrained,options);
```

این تابع برای حل مسئله بهینه سازی محدودیت دار و جهت محاسبه سیگنال کنترل مورد استفاده قرار می گیرد.

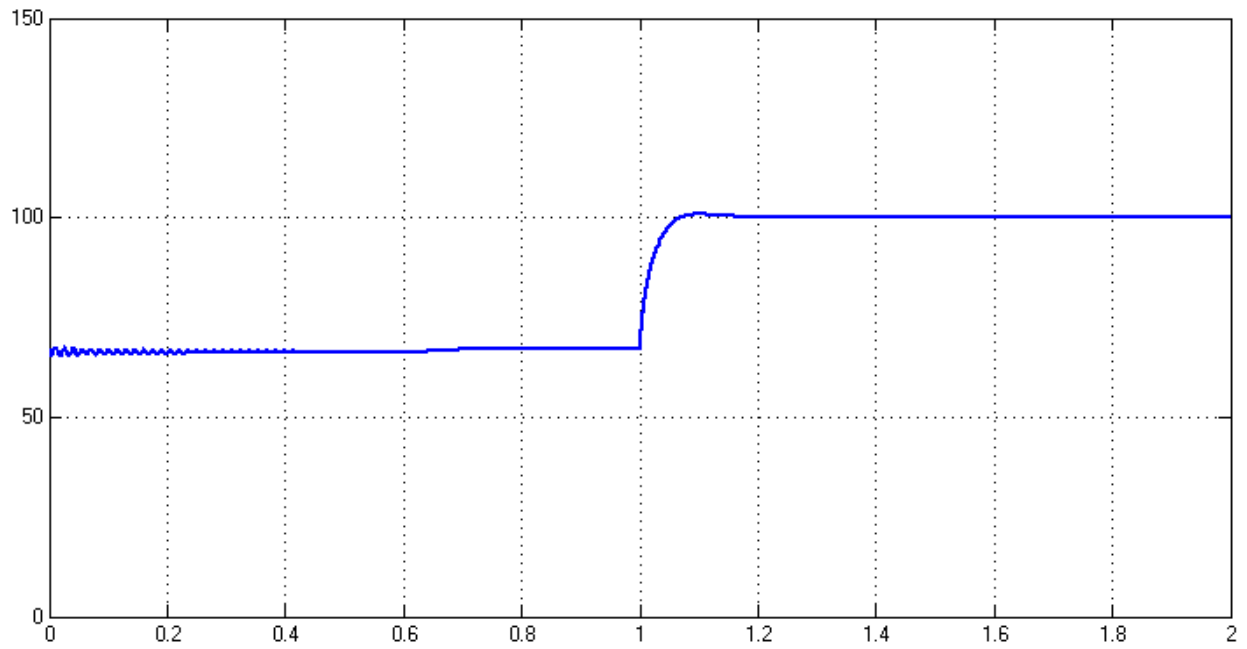
این فایل ها توسط فایل سیمولینک MPC_BOOST_CONVERTE_Constrained.mdl فراخوانی خواهند

شد.

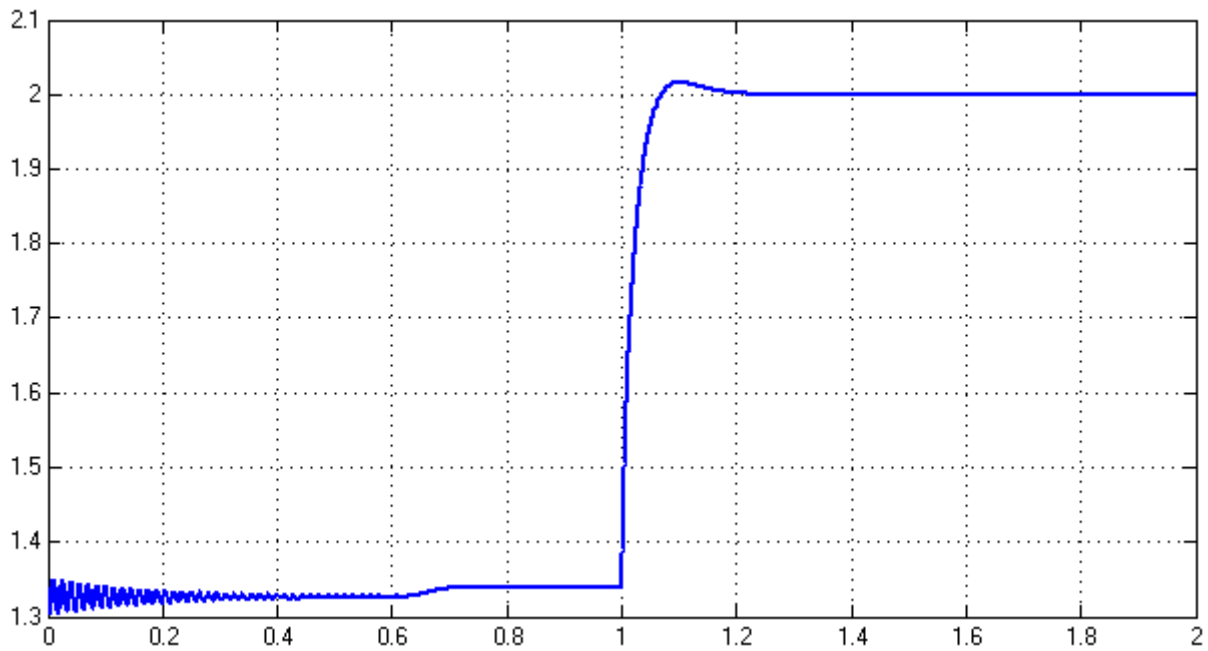
در صورتی که ولتاژ مرجع ابتدا ۶۷ ولت و سپس به مقدار ۱۰۰ ولت تغییر یابد نتایج شبیه سازی به صورت زیر

حاصل خواهد شد.

تشریح برنامه و نحوه اجرای آن



ولتاژ خروجی برای حالت با محدودیت



جریان خروجی برای حالت با محدودیت