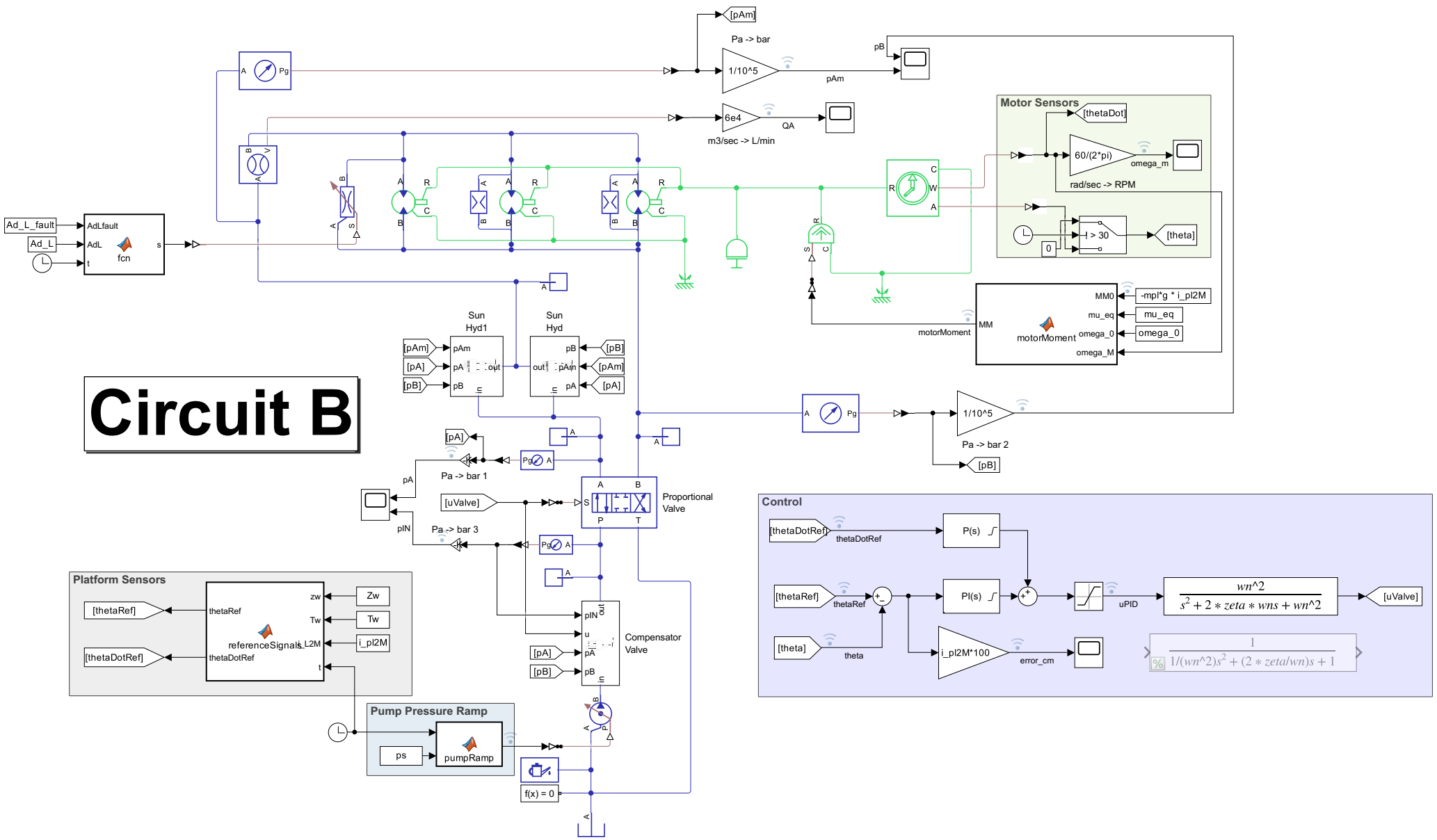
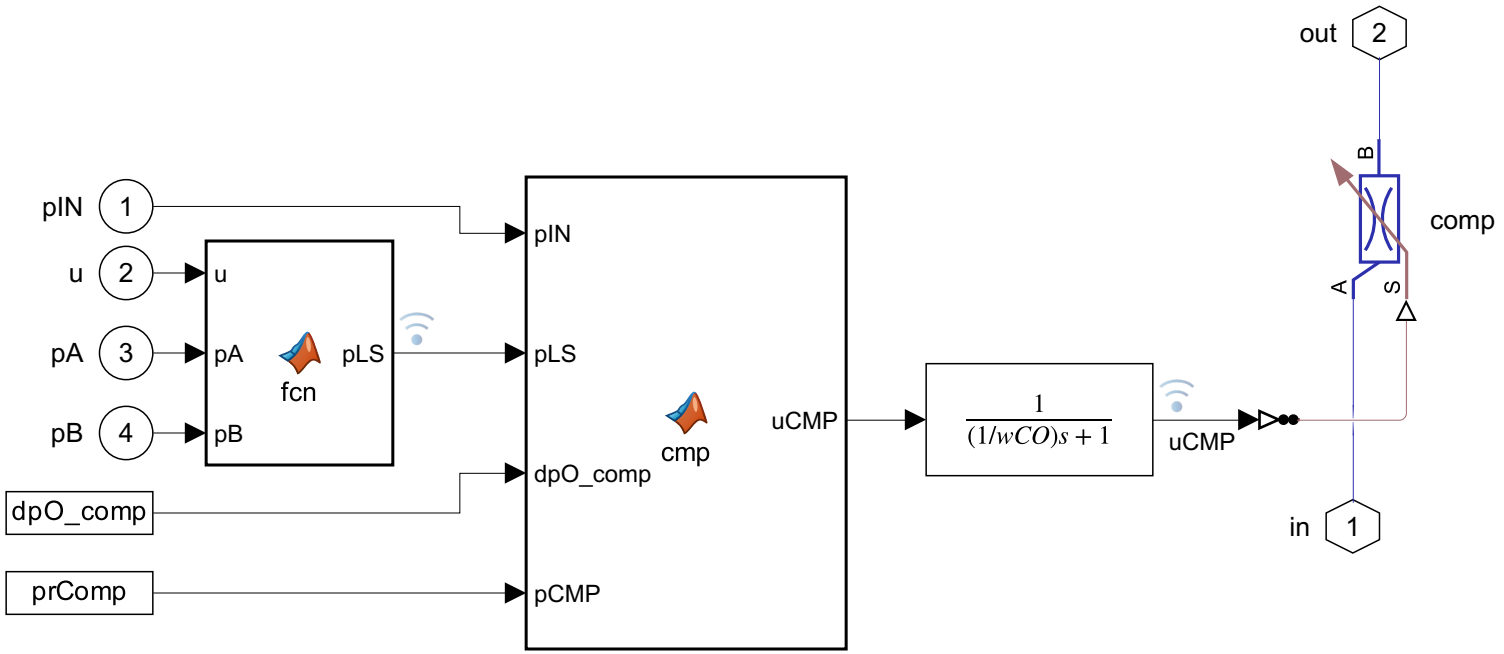

Circuit B
Ad_L_fault
Ad_L
AdLfault
AdL
fcn
[pAm]
Pa -> bar
1/10^5
pAm
pB
6e4
QA
m3/sec -> L/min
Motor Sensors
[thetaDot]
60/(2*pi)
omega_m
rad/sec -> RPM
[theta]
0
I > 30
motorMoment
MM
MM0
mu_eq
omega_0
omega_M
-mpl*g * i_pl2M
Sun Hyd1
Sun Hyd
[pA]
[pB]
Pa -> bar 1
Pa -> bar 2
Pa -> bar 3
[uValve]
Proportional Valve
pIN
Compensator Valve
Platform Sensors
[thetaRef]
[thetaDotRef]
referenceSignals
zw
Tw
i_pl2M
Zw
Tw
i_pl2M
Pump Pressure Ramp
ps
pumpRamp
f(x) = 0
Control
thetaDotRef
P(s)
PI(s)
theta
i_pl2M*100
error_cm
uPID
wn^2 / s^2 + 2 * zeta * wns + wn^2
1 / 1/(wn^2)s^2 + (2 * zeta/wn)s + 1


pIN
1
u
2
pA
3
pB
4
u
pA
pLS
fcn
pB
dpO_comp
prComp
pIN
pLS
dpO_comp
cmp
uCMP
pCMP
1
(1/wCO)s + 1
uCMP
out
2
B
A
S
comp
in
1

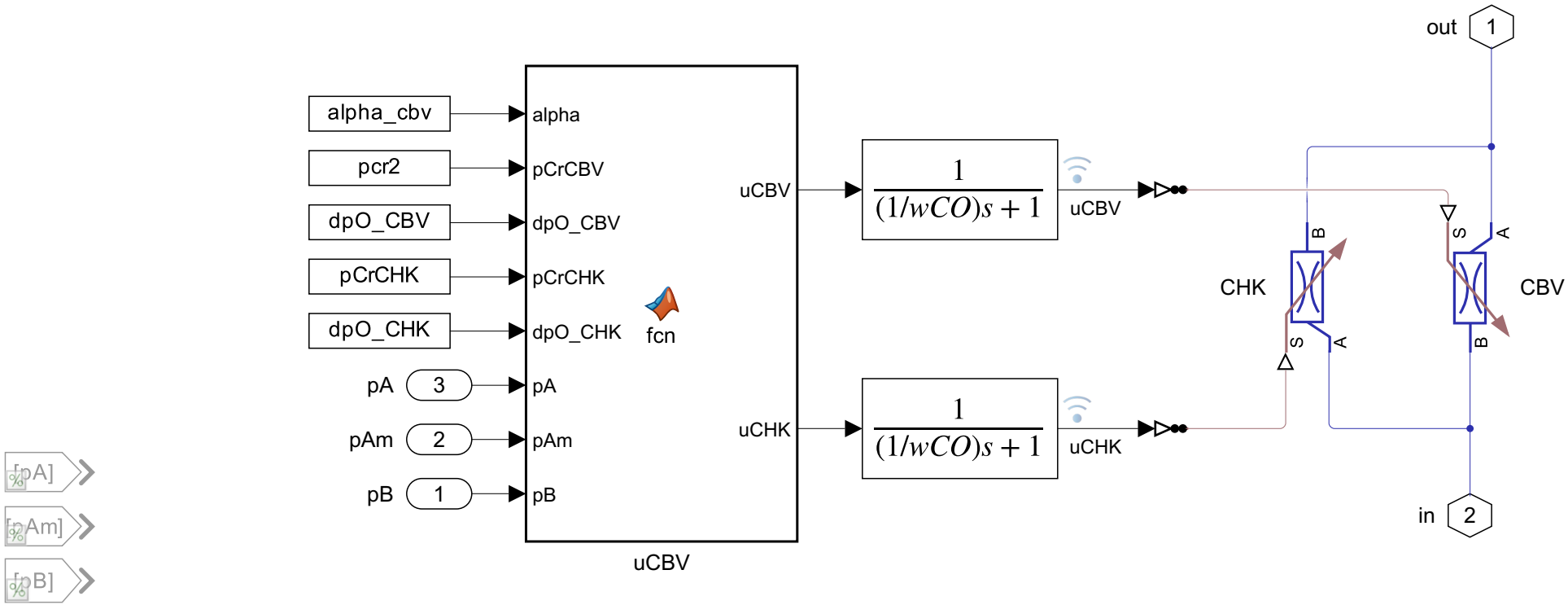
```
function uCMP = cmp(pIN, pLS, dpO_comp, pCMP)

uCMP = (pLS + pCMP - pIN)/dpO_comp;

% Saturation
if uCMP > 1
    uCMP = 1;
elseif uCMP < 0
    uCMP = 0;
end
```

```
function pLS = fcn(u, pA, pB)

if u > 0
    pLS = pA;
else
    pLS = pB;
end
```

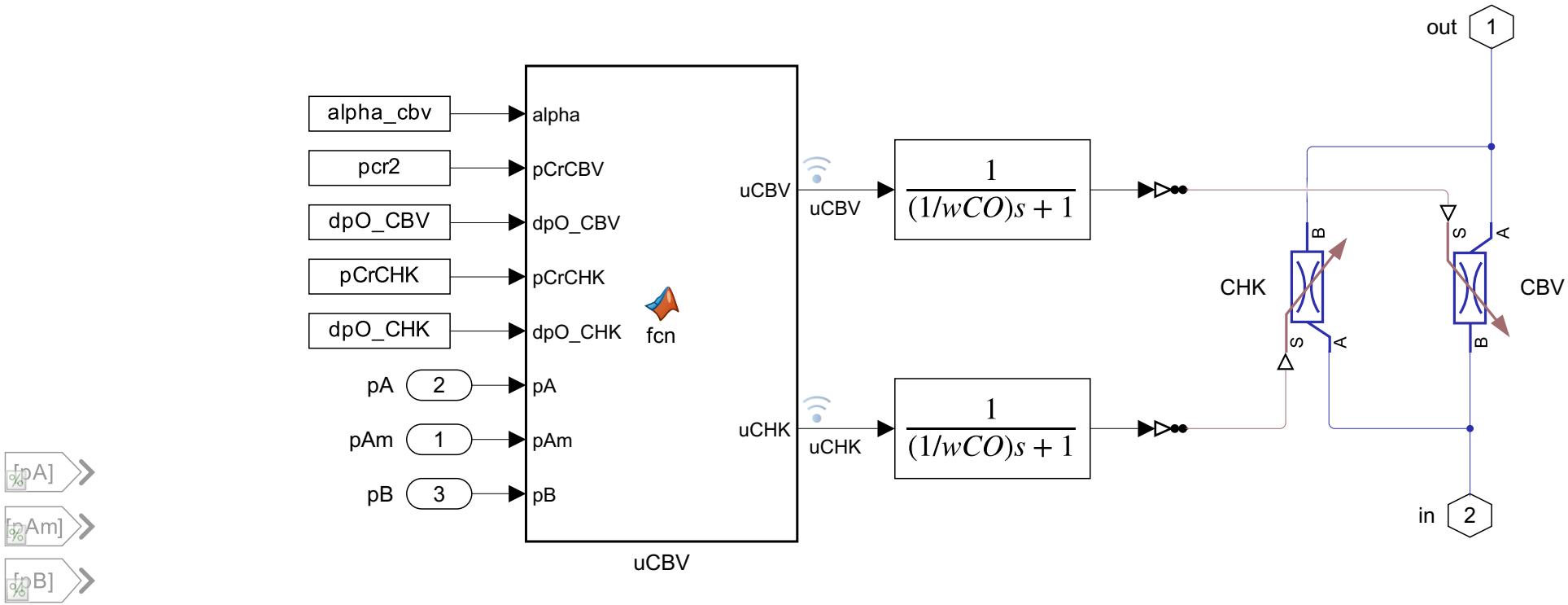
```
function s = fcn(AdLfault, AdL, t)

% if t<= 30
    s = AdL/AdLfault;
% else
%     s = 1;
% end
```

```
function [thetaRef,thetaDotRef] = referenceSignals(zw, Tw, i_L2M, t)

% Reference values for platform
zRef = zw * sin(2*pi/Tw * t); % [m]
zDotRef = 2*pi/Tw * zw * cos(2*pi/Tw * t); % [m/s]

% Reference values for motor
thetaRef = zRef/i_L2M; % [rad]
thetaDotRef = zDotRef/i_L2M; % [rad/sec]

% Envelope Functions
tEnv = 2*Tw; % [sec]
uEnv = t/tEnv; % [-]

if t < tEnv
    xEnv    = 3*uEnv^2    - 2*uEnv^3;
    xDotEnv = 6*uEnv/tEnv - 6*uEnv^2/tEnv;
else
    xEnv    = 1;
    xDotEnv = 0;
end

% Encorporating cubic polynomial into reference signals
thetaRef    = xEnv*thetaRef;
thetaDotRef = xEnv*thetaDotRef + xDotEnv*thetaRef;
```

```
function MM = motorMoment(MM0, mu_eq, omega_0, omega_M)

MM = MM0 * ( 1 + mu_eq * tanh(omega_M/omega_0) ); % [Nm]
```


alpha_cbv
pcr2
dpO_CBV
pCrCHK
dpO_CHK
pA
3
pAm
2
pB
1
alpha
pCrCBV
dpO_CBV
pCrCHK
dpO_CHK
pA
pAm
pB
fcn
uCBV
uCBV
uCHK
$\frac{1}{(1/wCO)s+1}$
uCBV
$\frac{1}{(1/wCO)s+1}$
uCHK
CHK
CBV
B
S
A
S
A
B
out
1
in
2
[pA]
[pAm]
[pB]

```
function [uCBV,uCHK] = fcn(alpha, pCrCBV, dpO_CBV, pCrCHK, dpO_CHK,pA, pAm, pB)

%%% CBV
uCBV = (alpha*pB + pAm - pCrCBV - (1 + alpha)*pA)/dpO_CBV;
    % Saturation
if uCBV > 1
    uCBV = 1;
elseif uCBV < 0
    uCBV = 0;
end

%%% Check Valve (CHK)
uCHK = (pA - pCrCHK - pAm)/dpO_CHK;
    % Saturation
if uCHK > 1
    uCHK = 1;
elseif uCHK < 0
    uCHK = 0;
end
```


alpha_cbv
pcr2
dpO_CBV
pCrCHK
dpO_CHK
pA
2
pAm
1
pB
3
alpha
pCrCBV
dpO_CBV
pCrCHK
dpO_CHK
pA
pAm
pB
fcn
uCBV
uCBV
uCBV
uCHK
uCHK
$\frac{1}{(1/wCO)s+1}$
$\frac{1}{(1/wCO)s+1}$
CHK
CBV
B
S
A
S
A
B
out
1
in
2
[pA]
[pAm]
[pB]

```
function [uCBV,uCHK] = fcn(alpha, pCrCBV, dpO_CBV, pCrCHK, dpO_CHK,pA, pAm, pB)

%%% CBV
uCBV = (alpha*pB + pAm - pCrCBV - (1 + alpha)*pA)/dpO_CBV;
    % Saturation
if uCBV > 1
    uCBV = 1;
elseif uCBV < 0
    uCBV = 0;
end

%%% Check Valve (CHK)
uCHK = (pA - pCrCHK - pAm)/dpO_CHK;
    % Saturation
if uCHK > 1
    uCHK = 1;
elseif uCHK < 0
    uCHK = 0;
end
```

```
function p = pumpRamp(t, pPump)

T = 0.2; % [sec]
tau = t/T; % [-]

if tau <= 1
    p = pPump*(3*tau^2 - 2*tau^3);
else
    p = pPump;
end

% p = 240e5;
```