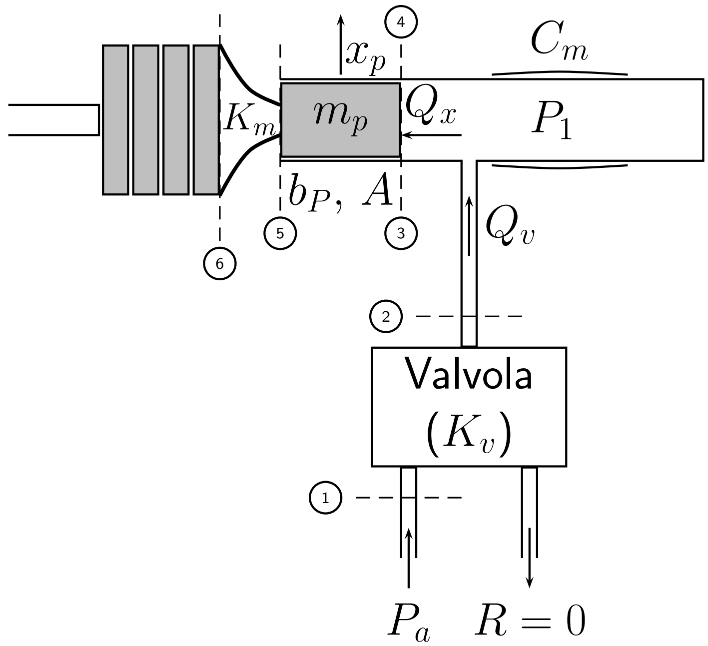
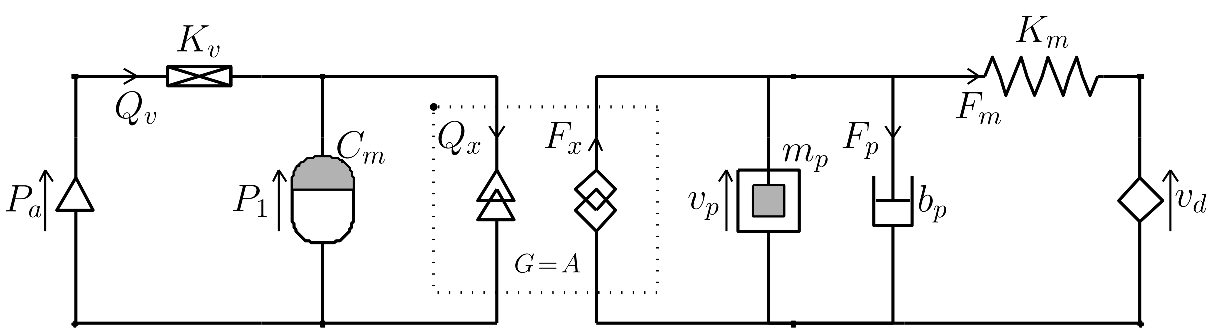
Hydraulic clutch system: POG modeling and state space equations

P_a	Supply pressure
Q_a	Volume flow rate
K_v	Valve proportional constant
C_m	Hydraulic capacity of the cylinder
P_1	Pressure within the cylinder
A	Section of the piston
x_p	Position of the piston
v_p	Velocity of the piston
m_p	Mass of the piston
b_p	Friction coefficient of the piston
K_m	Stiffness of the spring
F_m	Spring force against the piston
v_d	Wear velocity of the clutch discs



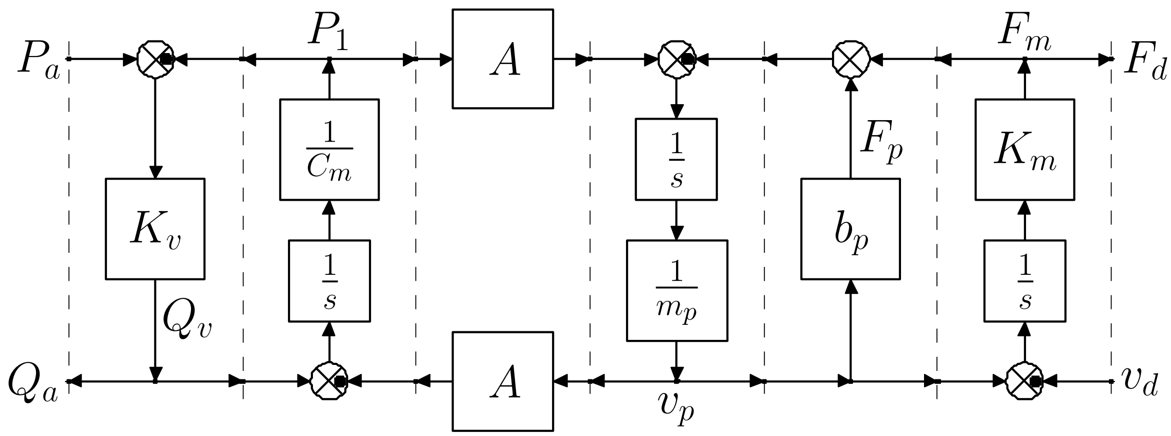
- The system can be graphically represented as follows:



- This scheme is obtained in POG Modeler using the following commands:

```
**, Gr, Si, Sn, Si, As, Si, POG, Si, EQN, Si, SLX, Si
iP, 1, a, En, P_a=1000, Fn, Q_a, An, -90
iG, 1, 2, Kn, K_v, Fn, Q_v
--, a, b
iC, 2, b, Kn, C_m, En, P_1
CB,[2;3],[b;c], Kn,F2=A*E1, En,[P_1;v_p], Fn,[Q_x; F_x], Sh,[0.7;-0.8]
mM, 3, c, Kn, m_p, En, v_p, Sh, -0.1
mB, 3, c, Kn, b_p, Fn, F_p, Sh, 0.4
mK, 3, 4, Kn, K_m, Fn, F_m, Ln, 1.4, Tr, 0.3
--, c, d, Ln, 1.4
mV, 4, d, En, -v_d, Fn, F_d, Pin,1
```

- The corresponding POG block scheme is:

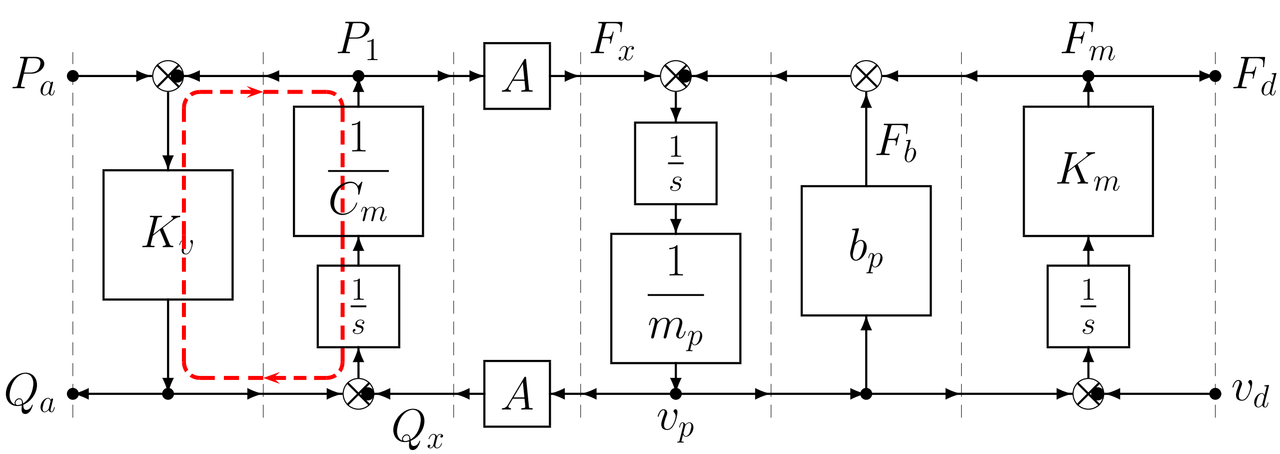


- The POG state space equations $\mathbf{L}\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}$ and $\mathbf{y} = \mathbf{C}\mathbf{x} + \mathbf{D}\mathbf{u}$ are:

$$\underbrace{\begin{bmatrix} C_m & 0 & 0 \\ 0 & m_p & 0 \\ 0 & 0 & \frac{1}{K_m} \end{bmatrix}}_{\mathbf{L}} \dot{\mathbf{x}} = \underbrace{\begin{bmatrix} -K_v & -A & 0 \\ A & -b_p & -1 \\ 0 & 1 & 0 \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} P_1 \\ v_p \\ F_m \end{bmatrix}}_{\mathbf{x}} + \underbrace{\begin{bmatrix} K_v & 0 \\ 0 & 0 \\ 0 & -1 \end{bmatrix}}_{\mathbf{B}} \mathbf{u}$$

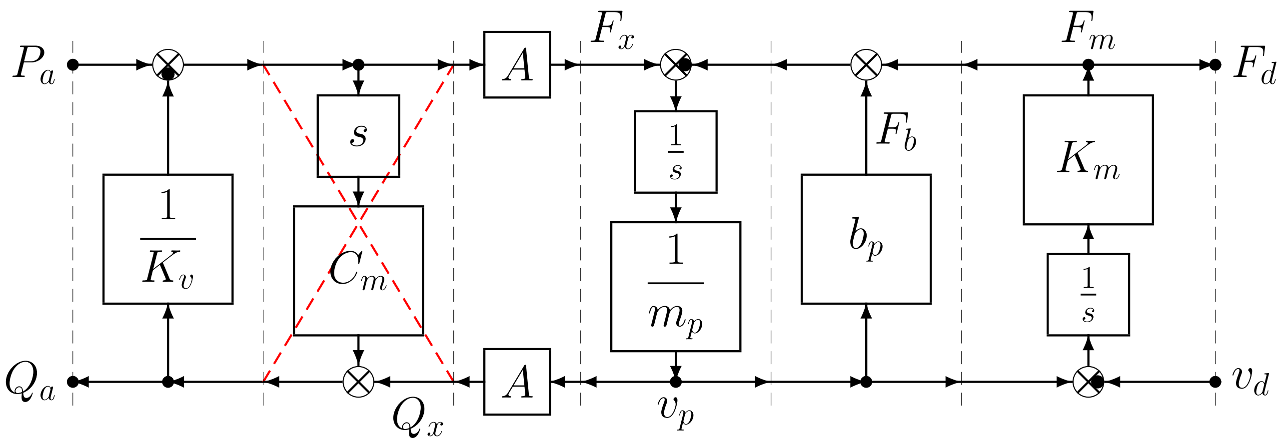
$$\underbrace{\begin{bmatrix} Q_a \\ F_d \end{bmatrix}}_{\mathbf{y}} = \underbrace{\begin{bmatrix} -K_v & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{\mathbf{C}} \mathbf{x} + \underbrace{\begin{bmatrix} K_v & 0 \\ 0 & 0 \end{bmatrix}}_{\mathbf{D}} \underbrace{\begin{bmatrix} P_a \\ v_d \end{bmatrix}}_{\mathbf{u}}$$

- Usually the hydraulic capacitance C_m is very small. The reduced model when $C_m = 0$ can be obtained graphically inverting the following path:

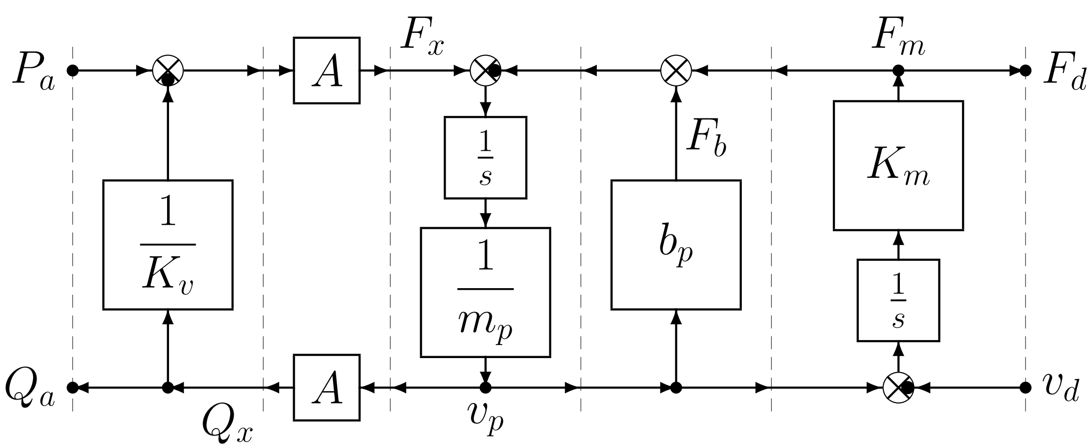


- Note: the hydraulic capacitor C_m is the only dynamic element present along the considered path.

- In the new block scheme the hydraulic capacitor $C_m = 0$ can be neglected:



- The reduced POG block scheme when $C_m = 0$ is:



- The state space equations of the reduced system are:

$$\underbrace{\begin{bmatrix} m_p & 0 \\ 0 & \frac{1}{K_m} \end{bmatrix}}_{\overline{\mathbf{L}}} \dot{\overline{\mathbf{x}}} = \underbrace{\begin{bmatrix} -b_p - \frac{A^2}{K_v} & -1 \\ 1 & 0 \end{bmatrix}}_{\overline{\mathbf{A}}} \underbrace{\begin{bmatrix} v_p \\ F_m \end{bmatrix}}_{\overline{\mathbf{x}}} + \underbrace{\begin{bmatrix} A & 0 \\ 0 & -1 \end{bmatrix}}_{\overline{\mathbf{B}}} \mathbf{u}$$

$$\underbrace{\begin{bmatrix} Q_a \\ F_d \end{bmatrix}}_{\mathbf{y}} = \underbrace{\begin{bmatrix} A & 0 \\ 0 & 1 \end{bmatrix}}_{\overline{\mathbf{C}}} \mathbf{x} + \underbrace{\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}}_{\overline{\mathbf{D}}} \underbrace{\begin{bmatrix} P_a \\ v_d \end{bmatrix}}_{\mathbf{u}}$$

- Note: when the graphical reduction is used, the state space equations of the reduces system are obtained "directly reading" the the POG block scheme of the reduced system.

- The same reduced state space equations could have been obtained mathematically as follows.
- Putting $C_m = 0$ in the state space equations of the original system, one obtains the following constraint:

$$-K_v P_1 - A v_p + K_v P_a = 0$$

- The state variable P_1 can be expressed as follows:

$$P_1 = -\frac{A}{K_v} v_p + P_a = \begin{bmatrix} -\frac{A}{K_v} & 0 \end{bmatrix} \overline{\mathbf{x}} + \begin{bmatrix} 1 & 0 \end{bmatrix} \mathbf{u}$$

where $\overline{\mathbf{x}}$ is the new state space vector and $\mathbf{u}$ is the input vector.

- In this case the old state space vector $\mathbf{x}$ can be expressed as follows:

$$\underbrace{\begin{bmatrix} P_1 \\ v_p \\ F_m \end{bmatrix}}_{\mathbf{x}} = \underbrace{\begin{bmatrix} -\frac{A}{K_v} & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}}_{\mathbf{T}} \underbrace{\begin{bmatrix} v_p \\ F_m \end{bmatrix}}_{\overline{\mathbf{x}}} + \underbrace{\begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}}_{\mathbf{T}_u} \underbrace{\begin{bmatrix} P_a \\ v_d \end{bmatrix}}_{\mathbf{u}} \quad \Leftrightarrow \quad \mathbf{x} = \mathbf{T}\overline{\mathbf{x}} + \mathbf{T}_u \mathbf{u}$$

- When $\mathbf{x} = \mathbf{T}\overline{\mathbf{x}} + \mathbf{T}_u \mathbf{u}$ the system cannot be transformed using a normal "congruent state-space transformation".
- A "congruent input state-space transformation" $\mathbf{x} = \mathbf{T}\overline{\mathbf{x}} + \mathbf{T}_u \mathbf{u}$ applied to POG system $\mathbf{S}$ generates a transformed and reduced POG system $\overline{\mathbf{S}}$:

$$\mathbf{S} = \begin{cases} \mathbf{L}\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \\ \mathbf{y} = \mathbf{C}\mathbf{x} + \mathbf{D}\mathbf{u} \end{cases} \quad \Rightarrow \quad \overline{\mathbf{S}} = \begin{cases} \overline{\mathbf{L}}\dot{\overline{\mathbf{x}}} = \overline{\mathbf{A}}\overline{\mathbf{x}} + \overline{\mathbf{B}}\mathbf{u} \\ \mathbf{y} = \overline{\mathbf{C}}\overline{\mathbf{x}} + \overline{\mathbf{D}}\mathbf{u} \end{cases}$$

where the new matrices $\overline{\mathbf{L}}$, $\overline{\mathbf{A}}$, $\overline{\mathbf{C}}$, $\overline{\mathbf{B}}$ and $\overline{\mathbf{D}}$ are defined as follows:

$$\overline{\mathbf{L}} = \mathbf{T}^T \mathbf{L} \mathbf{T}, \qquad \overline{\mathbf{A}} = \mathbf{T}^T \mathbf{A} \mathbf{T}, \qquad \overline{\mathbf{C}} = \mathbf{C}\mathbf{T},$$

and

$$\overline{\mathbf{B}} = \mathbf{T}^T(\mathbf{A}\mathbf{T}_u + \mathbf{B}), \qquad \overline{\mathbf{D}} = (\mathbf{C}\mathbf{T}_u + \mathbf{D}).$$

- When $\mathbf{T}_u = 0$ this new "congruent input state-space transformation" reduces to the previously defined "congruent state-space transformation".

- Using the command "EQN, Yes" one obtains an ascii file "*_EQN.txt" containing the Matlab state space equations of the original system:

```
 -- Matlab commands ----- (POG_Hydraulic_Clutch_SS_when_Cm_is_0.m) ---------
% State space equations:
% LM*dot_X = AM*X + BM*U
%        Y = CM*X + DM*U

% Symbolic Parameters of the POG block scheme:
syms  P_a Q_a K_v C_m P_1 A m_p v_p b_p K_m F_m v_d F_d s

LM = ...    % Energy matrix LM:
 [C_m,   0,     0 ;...
    0, m_p,     0 ;...
    0,   0, 1/K_m];

AM = ...    % Power matrix AM:
 [-K_v,   -A,  0 ;...
     A, -b_p, -1 ;...
     0,    1,  0];

BM = ...    % Input matrix BM:
 [K_v,  0 ;...
    0,  0 ;...
    0, -1];

CM = ...    % Output matrix CM:
 [-K_v, 0, 0 ;...
     0, 0, 1];

DM = ...    % Input-output matrix DM:
 [K_v, 0 ;...
    0, 0];

X  = ...    % State vector X:
 [P_1 ;...
  v_p ;...
  F_m];

U  = ...    % Input vector U:
 [P_a ;...
  v_d];

Y  = ...    % Output vector Y:
 [Q_a ;...
  F_d];
---------------------------------------------------------------------------
```

- The reduced system can be obtained using the following commands:

```
 -- Matlab commands ----- (POG_Hydraulic_Clutch_SS_when_Cm_is_0.m) ---------
%%%%%%%%%%  Matrices of the state space transformation  %%%%
 C_m = 0; LM=eval(LM);
 T  = [-A/K_v 0; 1 0; 0 1];
 Tu = [     1 0; 0 0; 0 0];
%%%%%%%%%%  Input state space congruent transformation  %%%%
 [Lt,At,Bt,Ct,Dt] = POG_Congruent_Transformation(LM,AM,BM,CM,DM,T,Tu);
---------------------------------------------------------------------------
```

- The function "POG_Congruent_Transformation" applies a POG congruent transformation to a given (LM, AM, BM, CM, DM) system:

```
-- Matlab commands ---- (POG_Congruent_Transformation.m) -------------------
%
%  [Lt,At,Bt,Ct,Dt] = POG_Congruent_Transformation(L,A,B,C,D,T,Tu)
%
% Applies a Congruent Transformation to a POG state space system:
%    L*dot_X = A*X + B*U       =>       Lt*dot_Xt = At*Xt + Bt*U
%          Y = C*X + D*U       =>               Y = Ct*Xt + Dt*U
% where Lt=T'*L*T, At=T'*A*T, Bt=T'*(A*Tu+B), Ct=C*T and Dt=C*Tu+D.
%
function [Lt,At,Bt,Ct,Dt] = POG_Congruent_Transformation(LM,AM,BM,CM,DM,T,Tu)
if nargin<7; Tu=zeros(size(LM,1),size(BM,2)); end
Lt = T.'*LM*T;
At = T.'*AM*T;
Bt = T.'*(AM*Tu+BM);
Ct = CM*T;
Dt = CM*Tu+DM;
return
---------------------------------------------------------------------------
```

- The transformed system computed using Matlab is the following:

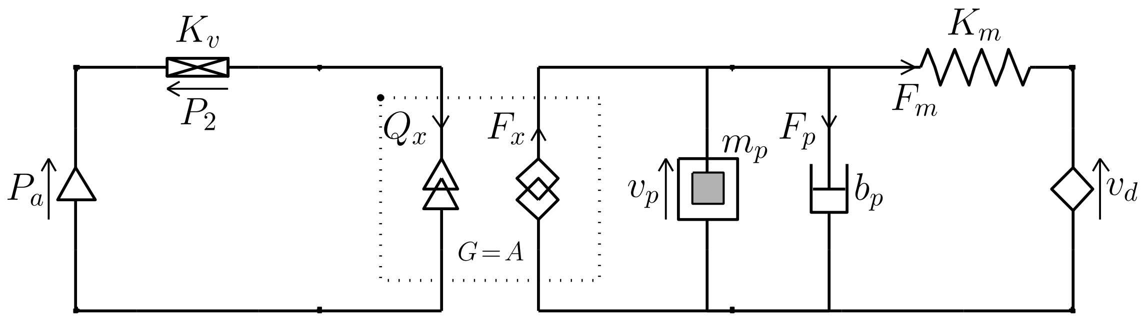
```
-- Matlab Output ----- (POG_Hydraulic_Clutch_SS_when_Cm_is_0.m) ---------
Lt =
[ m_p,      0]
[   0, 1/K_m]

At =
[ - A^2/K_v - b_p, -1]
[                1,  0]

Bt =
[ A,  0]
[ 0, -1]

Ct =
[ A, 0]
[ 0, 1]

Dt =
[ 0, 0]
[ 0, 0]
---------------------------------------------------------------------------
```

- This output has been obtained as follows:

```
-- Matlab commands ----- (POG_Hydraulic_Clutch_SS_when_Cm_is_0.m) ---------
disp('Lt = '); disp(Lt)
disp('At = '); disp(At)
disp('Bt = '); disp(Bt)
disp('Ct = '); disp(Ct)
disp('Dt = '); disp(Dt)
---------------------------------------------------------------------------
```

- When $C_m = 0$, the same reduced system could have been obtained defining a POG scheme without the hydraulic capacitor $C_m = 0$.

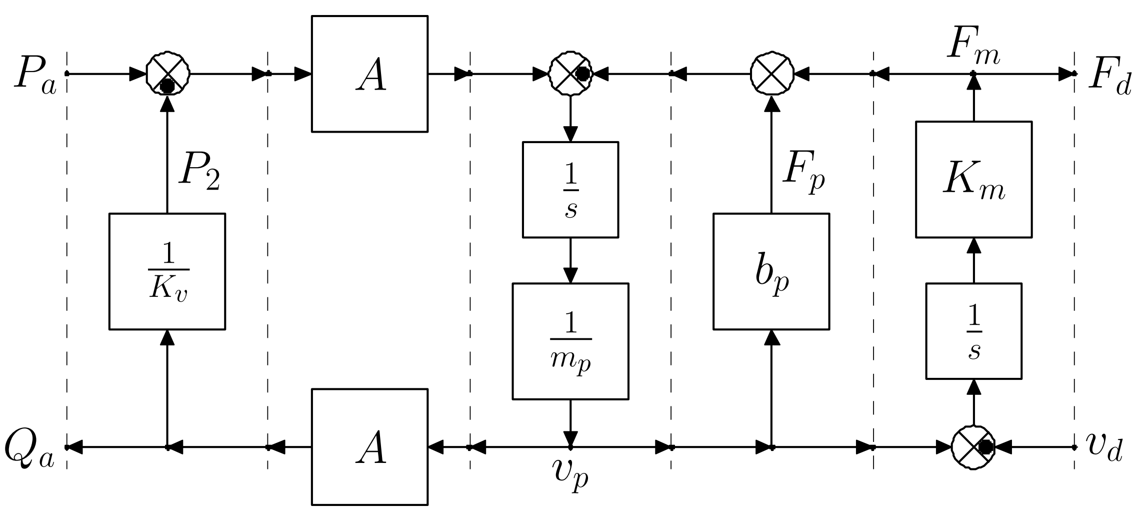
- The POG scheme:



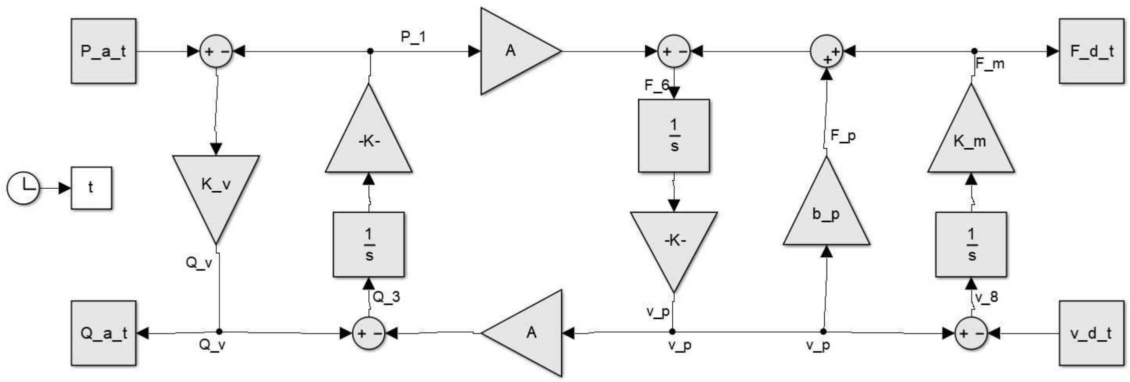
- The POG source code of the reduced system:

```
**, Gr, Si, Sn, No, As, Si, EQN, Si, POG, No, SLX, Si
iP, 1, a, En, P_a=1000, Fn, Q_a, An, -90
iG, 1, 2, Kn, K_v, Fn, Q_v
--, a, b
CB,[2;3],[b;c], Kn,F2=A*E1, En,[P_1;v_p], Fn,[Q_x; F_x], Sh,[0.5;-0.8]
mM, 3, c, Kn, m_p, En, v_p, Sh, -0.1
mB, 3, c, Kn, b_p, Fn, F_p, Sh, 0.4
mK, 3, 4, Kn, K_m, Fn, F_m, Ln, 1.4, Tr, 0.3
--, c, d, Ln, 1.4
mV, 4, d, En, -v_d, Fn, F_d, Pin,1
```

- The POG block scheme of the reduced system:



- Using the command "SLX, Si" the POG modeler generates the Simulink file "*_SLX.slx" and the Matlab file "*_SLX_m.m":

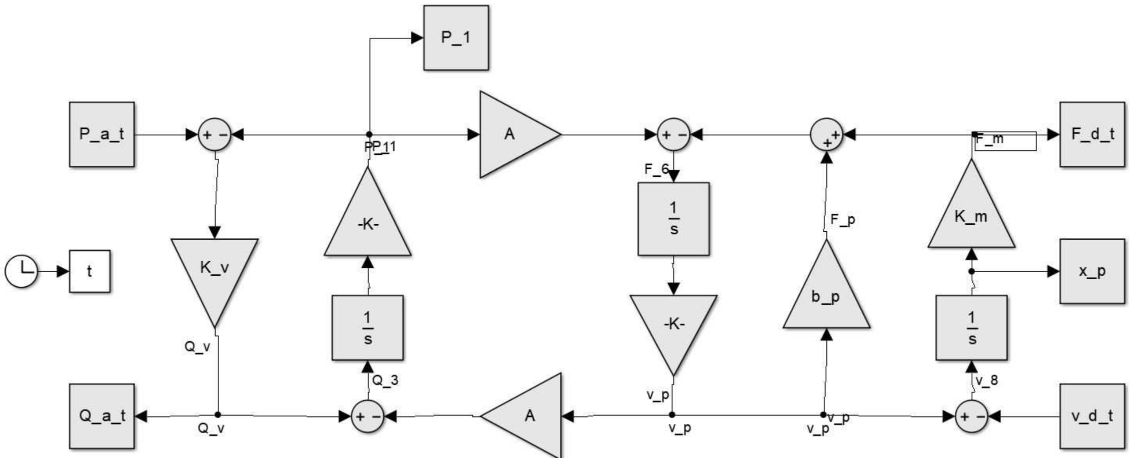



```
-- Matlab commands ----- (POG_Hydraulic_Clutch_SS_SLX_m.m) ----------------
clear all; close all
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%% SIMULATION OF SYSTEM: POG_Hydraulic_Clutch_SS
%%%%%%%%%%%%   SYSTEM PARAMETERS   %%%%%%%%%%%%%%%%%%%%%%%%%%
K_v=1;         % 2. Hyd. Conductance. Internal parameter.
C_m=1;         % 3. Hyd. Capacitance. Internal parameter.
A=1;           % 5. Parameter. Transformer/Gyrator.
m_p=1;         % 6. Mass. Internal parameter.
b_p=1;         % 7. Friction. Internal parameter.
K_m=1;         % 8. Stiffness. Internal parameter.
%%%%%%%%%%%%   INPUT VALUES   %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
P_a=1000;      % 1. Pressure. Input value.
v_d=0;         % 9. Velocity. Input value.
%%%%%%%%%%%%   INITIAL CONDITIONS   %%%%%%%%%%%%%%%%%%%%%%%%%
V_3_0=0;       % 3. Volume. Initial condition.
P_6_0=0;       % 6. Momentum. Initial condition.
x_8_0=0;       % 8. Displacement. Initial condition.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Tfin=10;  Nr_Ts_Points=2000;  Ts=Tfin/Nr_Ts_Points;
SLX_Name = 'POG_Hydraulic_Clutch_SS_SLX';
Out_Sim=sim(SLX_Name,'stoptime',num2str(Tfin),'SaveOutput','on','SaveTime','off');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Variables=Out_Sim.get;
Variables=setdiff(Variables,'t');
Nr_Var=length(Variables);
t=Out_Sim.get('t');
%%%%%% PLOT THE OUTPUP VARIABLES Y  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
S=[2,1]; Ns=S(1)*S(2); Nr=Ns;
for ii=1:Nr_Var; Unit(ii)={'1'}; Var(ii)={['Var\_' num2str(ii)]}; end
for ii=0:Ns:Nr_Var
    if ii+Ns>Nr_Var; Nr=Nr_Var-ii; end
    for jj=1:Nr
        figure(900+ii/Ns)
        y=Out_Sim.get(Variables{ii+jj});
        subplot(S(1),S(2),jj)
        plot(t,y/eval(['(' Unit{ii+jj} ')'])); grid on
        ylabel([strrep(Variables{ii+jj},'_','\_') ' [' Unit{ii+jj} ']'])
        title(Var{ii+jj})
        if jj>Nr-S(2); xlabel('Time [s]'); end
    end
end
---------------------------------------------------------------------------
```

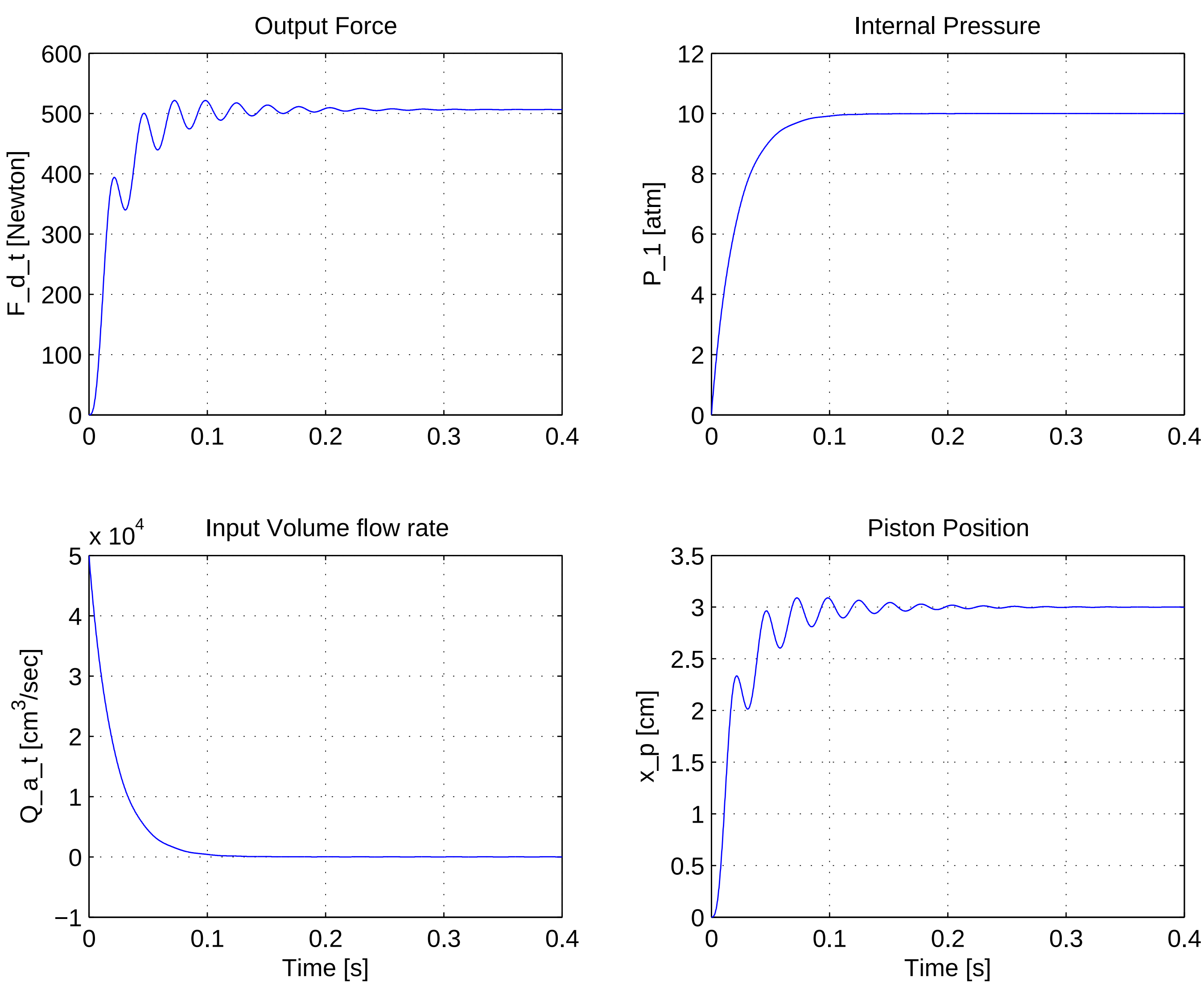
- The parameters of the system must be properly tuned (use SI Units):

```
 -- Matlab commands ----- (POG_Hydraulic_Clutch_NL_SLX_m.m) ----------------
clear all; close all
 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 %%% SI UNITS
 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 meters = 1; km=1000*meters; cm = 0.01*meters; mm = 0.001*meters; liters=0.001*meters^3;
 sec = 1; msec = 0.001*sec; microsec=0.001*msec; minutes = 60*sec; hours = 60*minutes;
 kg = 1; gr = 0.001*kg; Newton = 1; kN = 1000*Newton; Nm=Newton*meters;
 rad = 1; degrees =(pi/180)*rad; rounds=2*pi*rad; giri=2*pi;
 rpm = 2*pi/minutes*rad/sec; g_acc = 9.81*meters/sec^2; kgf = g_acc;
 Volt=1; mVolt=0.001*Volt; Amp=1; mAmp=0.001*Amp; Ohm=Volt/Amp; mOhm=0.001*Ohm;
 Watt=Volt*Amp; kWatt=1000*Watt; Weber=Volt*sec;
 Henry=Volt*sec/Amp; mHenry=0.001*Henry; Joule=1; kJoule=1000*Joule;
 K=1; Celsius=1;  Pa=1; atm=101325*Pa;  mol=1;
 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 %%% SIMULATION OF SYSTEM: POG_Hydraulic_Clutch_SS
 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 %%%%%%%%%%%%   INPUT VALUES  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 P_a_t=10*atm;      % 1. Pressure. Input value.
 v_d_t=0;           % 9. Velocity. Input value.
 %%%%%%%%%%%%   SYSTEM PARAMETERS  %%%%%%%%%%%%%%%%%%%%%%%%%%
 K_v=(10*cm)^3/(sec*(0.2*atm));   % 2. Hyd. Conductance. Internal parameter.
 C_m=1e-9;          % 3. Hyd. Capacitance. Internal parameter.
 A=5*cm^2;          % 5. Parameter. Transformer/Gyrator.
 m_p=300*gr;        % 6. Mass. Internal parameter.
 b_p=10;            % 7. Friction. Internal parameter.
 K_m=P_a_t*A/(3*cm);    % 8. Stiffness. Internal parameter.
 %%%%%%%%%%%%   INITIAL CONDITIONS  %%%%%%%%%%%%%%%%%%%%%%%%%
 P_1_0=0;           % 3. Perssure. Initial condition.
 v_p_0=0;           % 6. Velocity. Initial condition.
 x_p_0=0;           % 8. Displacement. Initial condition.
 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 Tfin=0.4; Nr_Ts_Points=2000;
 Ts=Tfin/Nr_Ts_Points;
 SLX_Name = 'POG_Hydraulic_Clutch_NL_SLX';
 Out_Sim=sim(SLX_Name,'stoptime',num2str(Tfin),'SaveOutput','on','SaveTime','off');
 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 Variables=Out_Sim.get;
 Variables=setdiff(Variables,'t');
 Nr_Var=length(Variables);
 t=Out_Sim.get('t');
 %%%%%% PLOT THE OUTPUP VARIABLES Y  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 S=[2,2]; Ns=S(1)*S(2); Nr=Ns;
 Unit={'Newton','atm','cm^3/sec','cm'};
 Var ={'Output Force','Internal Pressure','Input Volume flow rate','Piston Position'};
 for ii=0:Ns:Nr_Var
     if ii+Ns>Nr_Var; Nr=Nr_Var-ii; end
     for jj=1:Nr
         figure(900+ii/Ns)
         y=Out_Sim.get(Variables{ii+jj});
         subplot(S(1),S(2),jj)
         plot(t,y/eval(['(' Unit{ii+jj} ')'])); grid on
         ylabel([strrep(Variables{ii+jj},'_','\_') ' [' Unit{ii+jj} ']'])
         title(Var{ii+jj});
         if jj>Nr-S(2);  xlabel('Time [s]');  end
     end
 end
 ---------------------------------------------------------------------------
```

- The Simulink file "POG_Hydraulic_Clutch_NL_SLX.slx" has been modifies adding new "To Workspace" blocks:



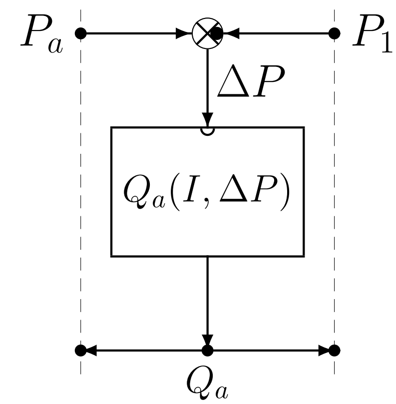
- With the new parameter one obtains the following simulation results:



- Files "*_SLX.slx" and "*_SLX_m.m" can be modified in order to insert in the model the same non linearities which are present in the real system.

- The hydraulic valve, for example is a non linear static element usually defined as follows:

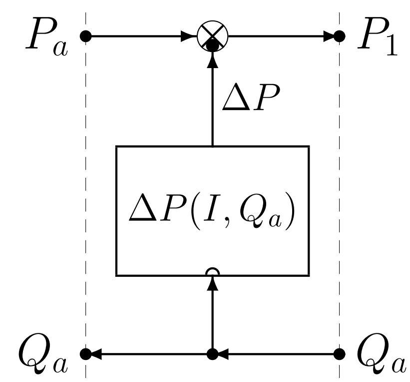
$$Q_a(I, \Delta P) = K_v(I)\,\mathrm{sign}(\Delta P)\sqrt{|\Delta P|}$$



where $\Delta P = P_a - P_1$ and $K_v(I)$ is a function of the current I.

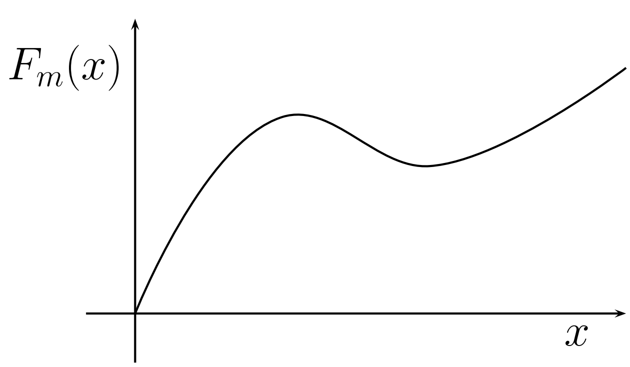
- The reduced model is characterized by the following nonlinearity:

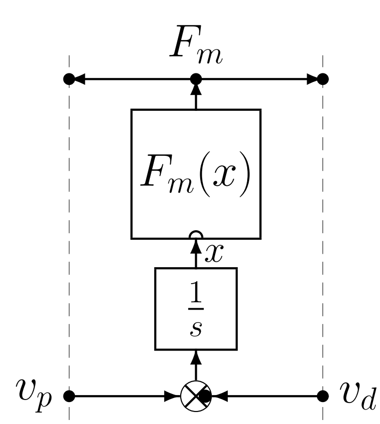
$$\Delta P(I, Q_a) = \left(\frac{Q_a}{K_v(I)}\right)^2$$



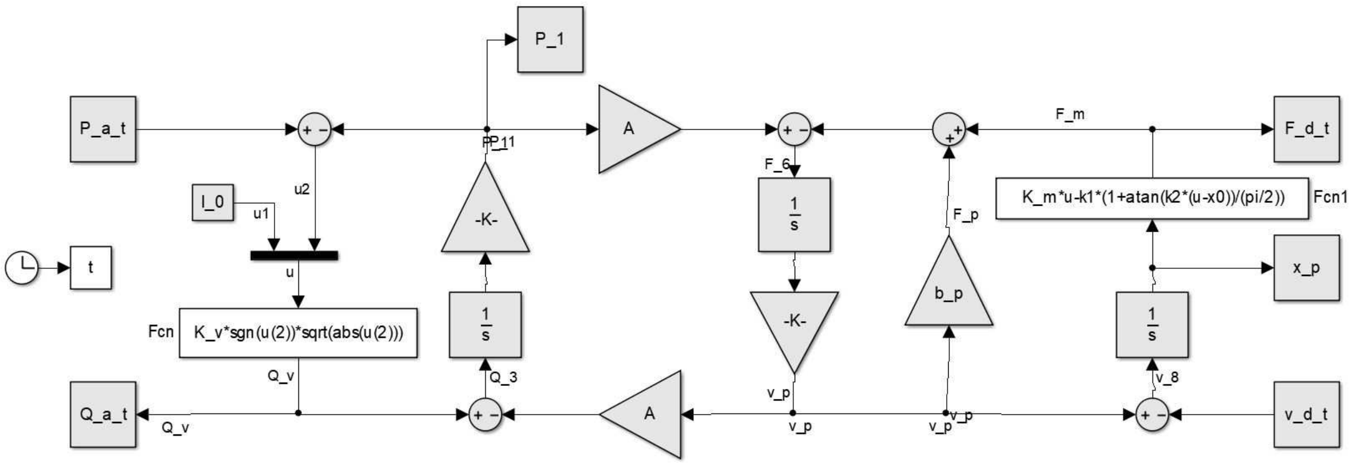
where Q_a is in the input volume flow rate.

- Also the stiffness K_m of the spring is usually nonlinear:

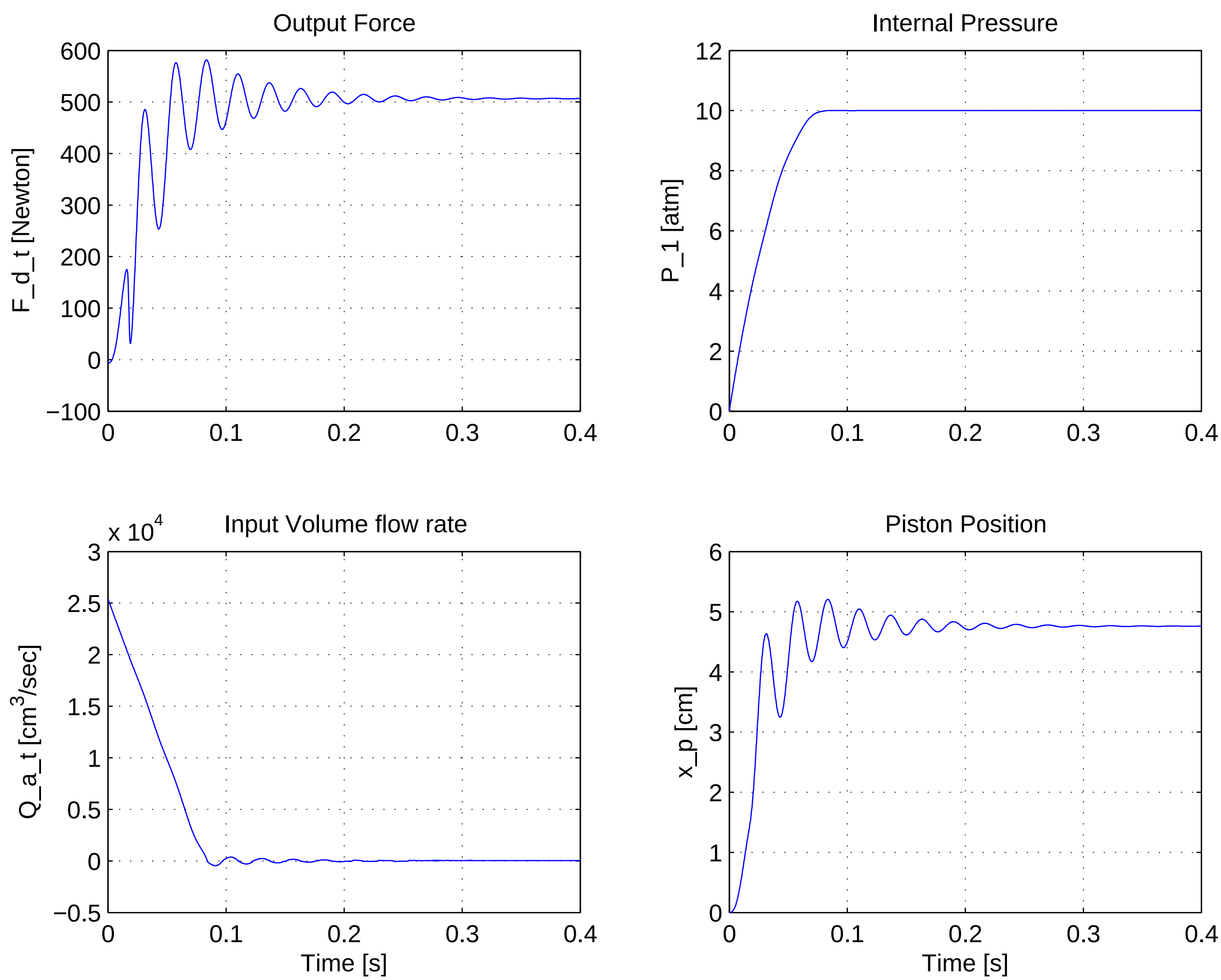




- The Simulink file “POG_Hydraulic_Clutch_NL_2_SLX.slx” has been modifies adding new “Fnc” blocks:



- With the new parameters one obtains the following simulation results:



- New parameters: `K_v=(80*cm)`^{3}`/(sec*(0.2*atm));` `I_0=0*Amp;` `x0=1.5*cm;` `k1=150;` `k2=1000.`