APPENDIX C
Matlab File for Building the PPF Controller

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% PPF_val9.m  Last edited 12/15/99

% This file contains all values necessary for running
% the dSPACE model for the Positive Position Feedback (PPF) Controller.

% This m-file must be executed before building the
% dSPACE model!

Clear all;

f1=60;              %frequency of 1st PPF-filter in Hertz
d1=0.02;            %damping ratio of 1st PPF-filter
f2=130;             %frequency of 2nd PPF-filter in Hertz
d2=0.02;            %damping ratio of 2nd PPF-filter
f3=160;             %frequency of 3rd PPF-filter in Hertz
d3=0.02;            %damping ratio of 3rd PPF-filter
f4=170;             %frequency of 4th PPF-filter in Hertz
d4=0.02;            %damping ratio of 4th PPF-filter
f5 = 280;
d5 = 0.02;
f6=290;             %frequency of 5th PPF-filter in Hertz
d6=0.02;            %damping ratio of 5th PPF-filter

f_lp=10;            %frequency of low pass filter in Hertz

ts=0.0001;          %sample time for discrete system

% State Space Matrices for the PPF filter
A1=[0 1; -(2*pi*f1)^2 -4*pi*f1*d1];
B1=[0; 1];
C1=[1 0];
D1=0;
PPFF1=ss(A1,B1,C1,D1)
A2=[0 1; -(2*pi*f2)^2 -4*pi*f2*d2];
B2=[0; 1];
C2=[1 0];
D2=0;
PPFF2=ss(A2,B2,C2,D2)
A3=[0 1; -(2*pi*f3)^2 -4*pi*f3*d3];
B3=[0; 1];
C3=[1 0];
D3=0;
PPFF3=ss(A3,B3,C3,D3)
A4=[0 1; -(2*pi*f4)^2 -4*pi*f4*d4];
B4=[0; 1];
C4=[1 0];
D4=0;
PPFF4=ss(A4,B4,C4,D4)
A5=[0 1; -(2*pi*f5)^2 -4*pi*f5*d5];
B5=[0; 1];
C5=[1 0];
D5=0;
PPFF5=ss(A5,B5,C5,D5)
A6=[0 1; -(2*pi*f6)^2 -4*pi*f6*d6];
```

```
B6=[0; 1];
C6=[1 0];
D6=0;
PPFF6=ss(A6,B6,C6,D6)


% Driscretization
PPFFd1=c2d(PPFF1,ts);
PPFFd2=c2d(PPFF2,ts);
PPFFd3=c2d(PPFF3,ts);
PPFFd4=c2d(PPFF4,ts);
PPFFd5=c2d(PPFF5,ts);
PPFFd6=c2d(PPFF6,ts);


% Extracting data (matrices, numerator, denominator) from the systems
[Ad1,Bd1,Cd1,Dd1]=ssdata(PPFFd1)
[Ad2,Bd2,Cd2,Dd2]=ssdata(PPFFd2)
[Ad3,Bd3,Cd3,Dd3]=ssdata(PPFFd3)
[Ad4,Bd4,Cd4,Dd4]=ssdata(PPFFd4)
[Ad5,Bd5,Cd5,Dd5]=ssdata(PPFFd5)
[Ad6,Bd6,Cd6,Dd6]=ssdata(PPFFd6)


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Low pass filter 2nd order
num_lp=[1];
den_lp=[1/(2*pi*f_lp)^2 1/(2*pi*f_lp) 1];
LP_FILTER=tf(num_lp,den_lp);

% Driscretization
LP_FILTERd=c2d(LP_FILTER,ts);

% Extracting data (matrices, numerator, denominator) from the systems
[dnum_lp, dden_lp]=tfdata(LP_FILTERd,'v')
```