Appendix

Simulation program in Matlab (M-file for s-function):

```
function [sys,x0]=block(t,x,u,flag)
% for this block state varaibles are x1, x2 and T
% control vraiable is pressure on the brake pedal
    Mv=1000;    % vehicle Mass (unit: kg)
    Rw=0.31;    % radius of the wheel (unit: m)
    Jw=0.65;    % wheel inertia (unit: Kg m^2)
    Nv=2287;    % normal tire force (unit: N)
    Nw=4;       % wheel number
    b1N=Nv*Nw/(Mv*Rw);
    % reorganize the coefficient based on equations in (3.10)
    b2N=Rw*Nv/Jw;
    % reorganize the coefficient based on equations in (3.10)
    b3=1/Jw;
    % reorganize the coefficient based on equations in (3.10)
    % k1-k5 are coefficient for electromagnetic
        brake static model
    k1=1.0507;
    k2=0.0666;
    k3=-0.00000070876036;
    k4=-5.45875;
    k5=-0.00093;
    if flag==1,
    % when flag=1, function should return state derivatives, xDot
```

```
Lamda=(x(2)-x(1))/x(1);      % wheel slip
Lamdap=-0.175;
% peak slip value, varies based on road condition
Mup=-0.5;
% peak adhension coefficient, varies based on road condition
LamdaD=-0.12;                % desired slip value
Mu=-0.3*Mup*Lamda/(Lamda^2+Lamdap^2);
% adhesion coefficient - slip curve characteristic function
f1=0.595*(Rw/Mv)*x(1)*x(1);
% reorganize the coefficient based on (3.10)
f2=0;
% reorganize the coefficient based on (3.10)
sys(1)=-f1+b1N*Mu;
% state equation for x1
sys(2)=-f2-b2N*Mu-b3*x(3);
% state equatuion for x2
Kb=0.25*k1*x(1)./(1+(k2*x(1).^2+k3*x(1).^4)./(1+k4*x(1)+k5*x(1).^3)).^2;
% brake gain coefficient, brake static characteristics is taken into
account  here
tau=0.15;
% rise time characteristic
sys(3)=(Kb*u(1)-x(3))/tau;
% state variable for brake torque
elseif  flag==0,
% when flag = 0, s-function returns sizes of parameters and initial
      conditions
x2init=60.8437;       % initial value for x2
x1init=69.0346;       % initial value for x1
Lamdainit=(x2init-x1init)/x1init;     % initial value for wheel slip
sys=[3;0;5;1;0;0];
```

```
% function returns information about :
% number of continuous states is 3
% they are for x1, x2 and braking torque
% number of discrete states is 0
% number of inputs is 1, which  is pedal pressure
% number of outputs is 5
Tinit=0;        % initial braking torque value
x0=[x1init;x2init;Tinit];        % initial conditions for the state vector
elseif flag==3,
% when flag = 3, function returns the output vector
Lamda_dot=(((1+Lamda)*f1-f2)
(b2N+(1+Lamda)*b1N)*Mu+b3*x(3))/x(1);
% intermediate result for calculating the controlled variable
uu=((1+Lamda)*f1-f2)-(b2N+(1+Lamda)*b1N)*Mu+b3*x(3);
% intermediate result for calculating the controlled variable

Mu_dot=((Lamda*Lamda+Lamdap*Lamdap)*(0.3*Mup*Lamda_dot)
- (-0.3*Mup*Lamda)*2*Lamda*Lamda_dot)
/(Lamda*Lamda+Lamdap*Lamda        p)^2;
% intermediate result for calculating the controlled variable
F=-uu*(-
f1+b1N*Mu)/(x(1)*x(1))+(Lamda_dot*f1+(1+Lamda)*f1_dot-
b2N*Mu_dot-b1N*Mu*Lamda_dot-(1+Lamda)*b1N*Mu_dot+b3*(0-
x(3))/tau)/x(1);
% intermediate result for calculating the controlled variable
bu=b3*Kb/tau/x(1);
% intermediate result for calculating the controlled variable
LamdaE=-(Lamda-LamdaD);
% difference between desired wheel slip and real wheel slip
% next part of the program, we try to get the controlled Brake
pressure
```

```
bmax=bu*1.1;
% assume that the maximum estimation error of the bu (control
% gain) is 10 percent less than the calculated value
bmin=bu*0.9;
% assume that the maximum estimation error of the bu (control
% gain) is 10 percent more than the calculated value
b_hat=sqrt(bmax*bmin);
% reasonable estimate of the dynamic control gain can be the
% geometric mean of the upper bound and lower bound
f_hat=F*0.9;
% assume that estimate of f in dynamics is 90% of clculated value
Gamma=250;
% bandwidth of the system
alpha=sqrt(bmax/bmin);
% gain margin of the design
p_hat=-f_hat-Gamma*Lamda_dot;
% approximation of control law, see (4.10)
k=alpha*(abs(F)*0.1+10)+(alpha-1)*abs(p_hat)+8000;
% gain of the design,  see (4.16).
% gain must be big enough to assure the sliding condition
phi=0.005;
% boundary layer thickness for chatter reduction
if abs(LamdaE)<phi
    i_control=LamdaE/phi;
    % inside the boundary layer,
    % interpolate using (error of Lamda)/thickness
else
    i_control=sign(LamdaE);
    % outside the boundary, use sign function
end
```

```
BT=(1/b_hat)*(p_hat-k*ipush);
% implement the control law, see (4.15)
sys(1)=Lamda;
% output wheel slip for recording and monitoring
sys(2)=BT;
% output controlled pedal pressure which would be feedback as
% the input for the next time step
sys(3)=x(1);
% output x1 for recording and monitoring
sys(4)=x(2);
% output x2 for recording and monotoring
sys(5)=x(3);
% output braking torque for recording and monitoring
else
    sys=[];
end
```