```
function phaseout = DSContinuous(input)
s              = input.phase(1).state;
u              = input.phase(1).control;
p              = input.phase(1).parameter;

z              = s(:,3);
v              = s(:,4);
gamma          = s(:,5);
psi            = s(:,6);

CL             = u(:,1);
mu             = u(:,2);

beta           = p(:,1);

rho            = input.auxdata.rho;
S              = input.auxdata.S;
CD0            = input.auxdata.CD0;
K              = input.auxdata.K;
g              = input.auxdata.g;
m              = input.auxdata.m;
W0             = input.auxdata.W0;

% % % w_h            = beta.*z + W0;
% % % DWxDt          = beta.*v.*sin(gamma);

A              = input.auxdata.A;
w_h            = beta.*(A.*z + (1 - A)./213.*z.^2) + W0;
DWxDt          = beta.*v.*sin(gamma).*(A + (1 - A)./213.*2.*z);


xdot           = v.*cos(gamma).*sin(psi) + w_h;
ydot           = v.*cos(gamma).*cos(psi);
zdot           = v.*sin(gamma);
vdot           = -(rho*S)/(2*m)*(CD0+K*CL.^2).*v.^2 - g*sin(gamma) - DWxDt.*sin(psi).*cos↙
(gamma);
gammadot       = (rho*S)/(2*m)*CL.*v.*cos(mu) - g*cos(gamma)./v + DWxDt.*sin(psi).*sin↙
(gamma)./v;
psidot         = ((rho*S)/(2*m)*CL.*v.*sin(mu)- DWxDt.*cos(psi)./v)./cos(gamma);

ngconstant     = (0.5*rho*S/m/g);
ng             = ngconstant.*CL.*v.^2;

phaseout.dynamics  = [xdot, ydot, zdot, vdot, gammadot, psidot];
phaseout.path = ng;
```