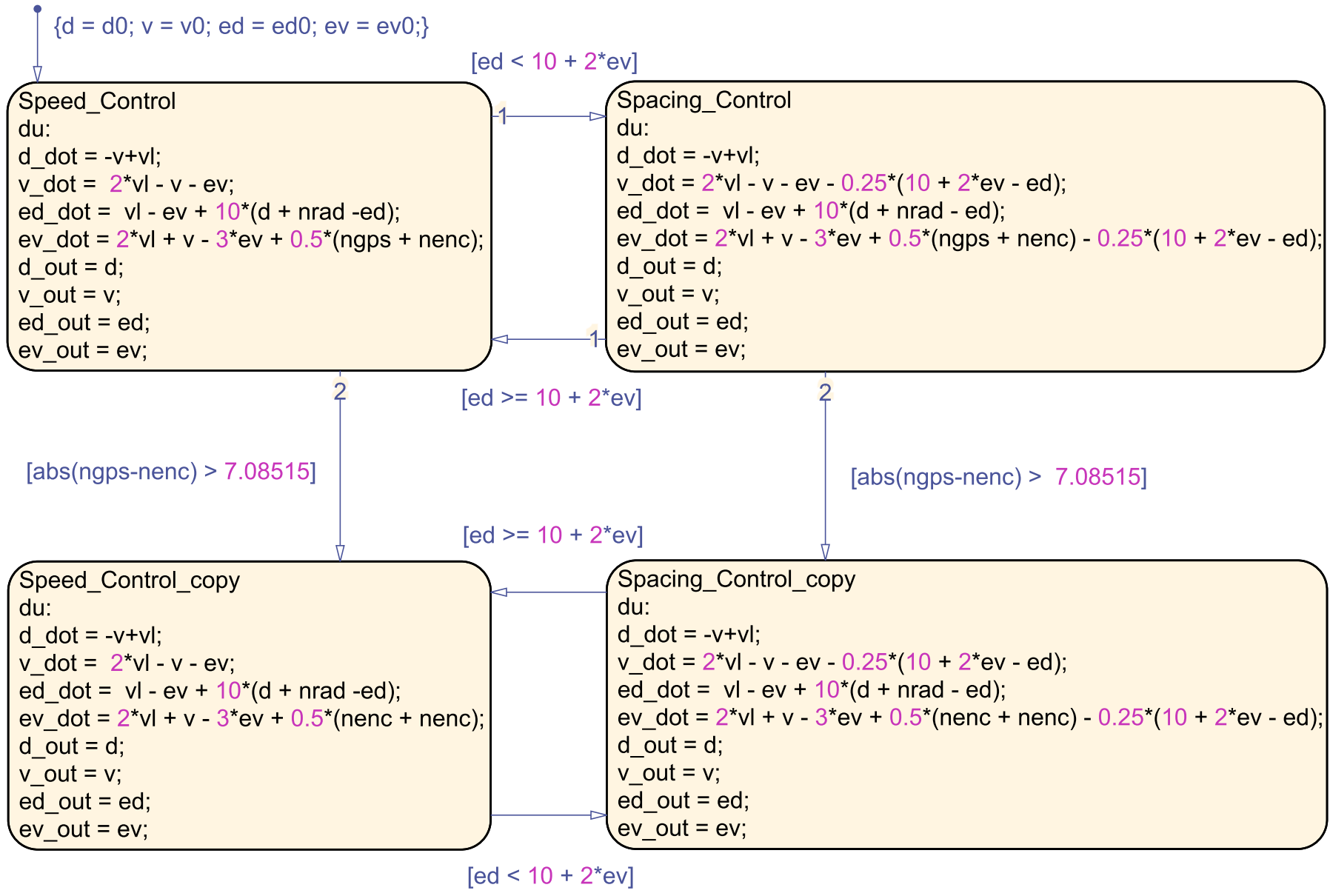
{d = d0; v = v0; ed = ed0; ev = ev0;}
[ed < 10 + 2*ev]
Speed_Control
du:
d_dot = -v+vl;
v_dot = 2*vl - v - ev;
ed_dot = vl - ev + 10*(d + nrad -ed);
ev_dot = 2*vl + v - 3*ev + 0.5*(ngps + nenc);
d_out = d;
v_out = v;
ed_out = ed;
ev_out = ev;
1
Spacing_Control
du:
d_dot = -v+vl;
v_dot = 2*vl - v - ev - 0.25*(10 + 2*ev - ed);
ed_dot = vl - ev + 10*(d + nrad - ed);
ev_dot = 2*vl + v - 3*ev + 0.5*(ngps + nenc) - 0.25*(10 + 2*ev - ed);
d_out = d;
v_out = v;
ed_out = ed;
ev_out = ev;
1
2
[ed >= 10 + 2*ev]
2
[abs(ngps-nenc) > 7.08515]
[abs(ngps-nenc) > 7.08515]
[ed >= 10 + 2*ev]
Speed_Control_copy
du:
d_dot = -v+vl;
v_dot = 2*vl - v - ev;
ed_dot = vl - ev + 10*(d + nrad -ed);
ev_dot = 2*vl + v - 3*ev + 0.5*(nenc + nenc);
d_out = d;
v_out = v;
ed_out = ed;
ev_out = ev;
Spacing_Control_copy
du:
d_dot = -v+vl;
v_dot = 2*vl - v - ev - 0.25*(10 + 2*ev - ed);
ed_dot = vl - ev + 10*(d + nrad - ed);
ev_dot = 2*vl + v - 3*ev + 0.5*(nenc + nenc) - 0.25*(10 + 2*ev - ed);
d_out = d;
v_out = v;
ed_out = ed;
ev_out = ev;
[ed < 10 + 2*ev]