16.333: Lecture # 13

Aircraft Longitudinal Autopilots

Altitude Hold and Landing

Altitude Controller

- In linearized form, we know from 1–5 that the change of altitude h can be written as the flight path angle times the velocity, so that

$$\dot{h} \approx U_0 \ \sin\gamma = U_0 \ (\theta - \alpha) = U_0 \ \theta - U_0 \left(\frac{w}{U_0}\right) = U_0 \ \theta - w$$

  – For fixed U_0, $\dot{h}$ determined by variables in short period model

- Use short period model augmented with θ state

$$x = \begin{bmatrix} w \\ q \\ \theta \end{bmatrix} \quad \Rightarrow \begin{cases} \dot{x} = \tilde{A}_{sp} x + \tilde{B}_{sp}\delta_e \\ \dot{h} = \begin{bmatrix} -1 & 0 & U_0 \end{bmatrix} x \end{cases}$$

where

$$\tilde{A}_{sp} = \begin{bmatrix} A_{sp} & 0 \\ \begin{bmatrix} 0 & 1 & 0 \end{bmatrix} \end{bmatrix}, \ \tilde{B}_{sp} = \begin{bmatrix} B_{sp} \\ 0 \end{bmatrix}$$

- In transfer function form, we get

$$\frac{h}{\delta_e} = \frac{K(s+4)(s-3.6)}{s^2(s^2 + 2\zeta_{sp}\omega_{sp}s + \omega_{sp}^2)}$$

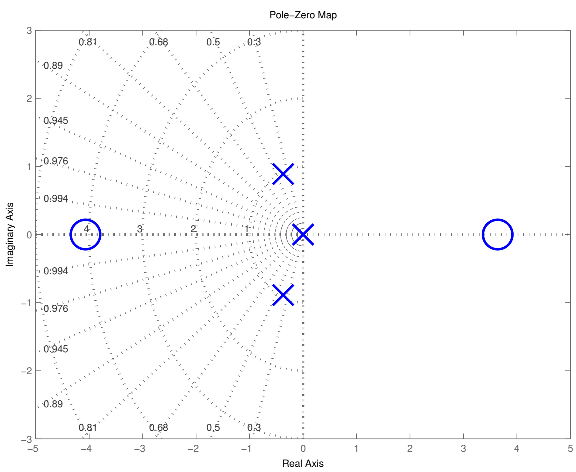


Figure 1: Altitude root locus #1

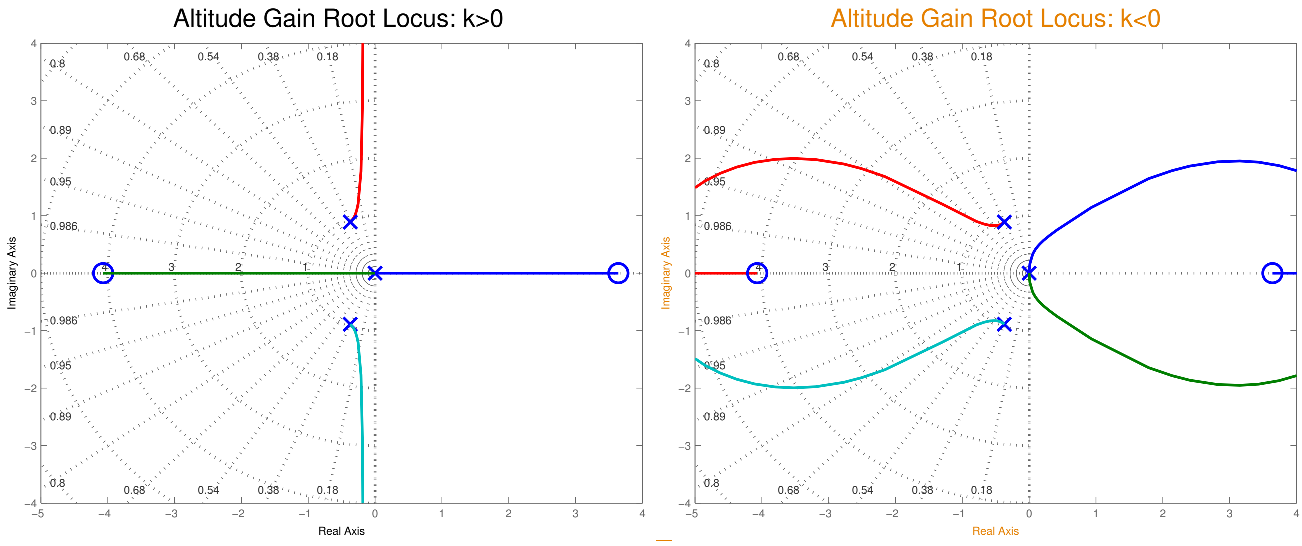


Figure 2: Altitude root locus #2

- Root locus versus h feedback clearly **NOT** going to work!

- Would be better off designing an inner loop first. Start with short period model augmented with the θ state

$$\delta_e = -k_w w - k_q q - k_\theta \theta + \delta_e^c = -\left[\begin{array}{ccc} k_w & k_q & k_\theta \end{array}\right] x + \delta_e^c = -K_{IL} x + \delta_e^c$$

  – Target pole locations $s = -1.8 \pm 2.4\mathbf{i}$, $s = -0.25$

  – Gains: $K_{IL} = \left[\begin{array}{ccc} -0.0017 & -2.6791 & -6.5498 \end{array}\right]$

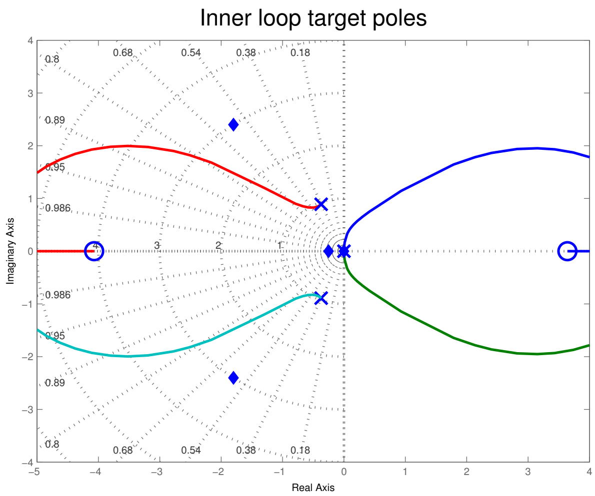


Figure 3: Inner loop target pole locations – won't get there with only a gain.

- Giving the closed-loop dynamics

$$\begin{aligned}
\dot{x} &= \tilde{A}_{sp}x + \tilde{B}_{sp}\left(-K_{IL}x + \delta_e^c\right) \\
&= (\tilde{A}_{sp} - \tilde{B}_{sp}K_{IL})x + \tilde{B}_{sp}\delta_e^c \\
\dot{h} &= \begin{bmatrix} -1 & 0 & U_0 \end{bmatrix} x
\end{aligned}$$

- In transfer function form

$$\frac{h}{\delta_e^c} = \frac{\tilde{K}(s+4)(s-3.6)}{s(s+0.25)(s^2+3.6s+9)}$$

with ζ_d and ω_d being the result of the inner loop control.

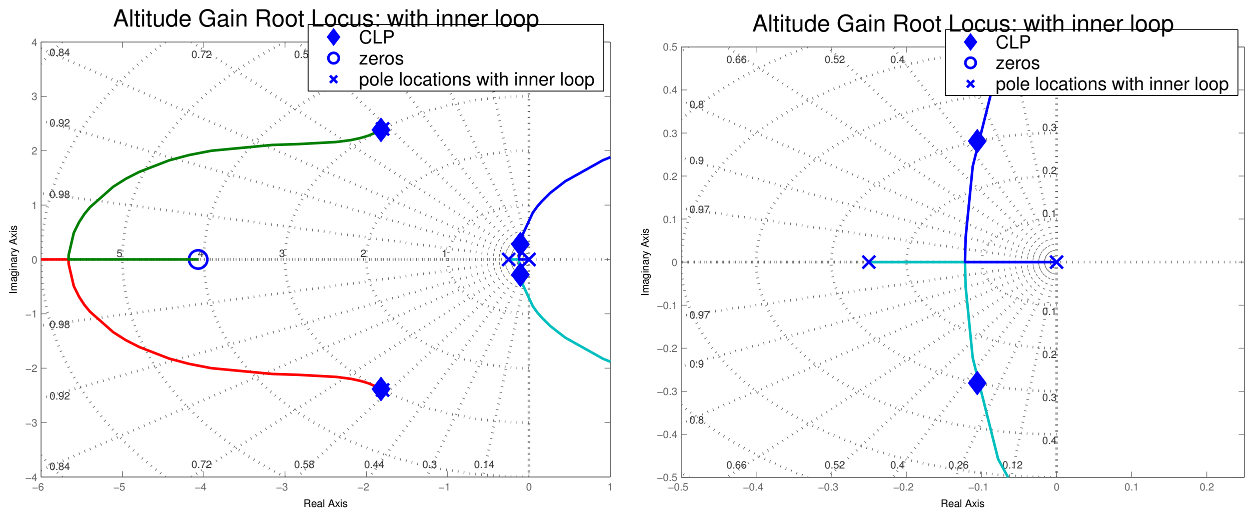


Figure 4: Root loci versus altitude gain $K_h < 0$ with inner loop added (zoomed on right). Much better than without inner loop, but gain must be small ($K_h \approx -0.01$).

- Final step then is to select the feedback gain on the altitude K_h and implement (h_c is the commanded altitude.)

$$\delta_e^c = -K_h(h_c - h)$$

  - Design inner loop to damp the short period poles and move one of the poles near the origin.
  - Then select K_h to move the 2 poles near the origin.

- Poles near origin dominate response $s = -0.1056 \pm 0.2811\mathbf{i}$
  $\Rightarrow \omega_n = 0.3$, $\zeta = 0.35$
  - Rules of thumb for 2nd order response:

10-90% rise time	$t_r = \dfrac{1 + 1.1\zeta + 1.4\zeta^2}{\omega_n}$
Settling time (5%)	$t_s = \dfrac{3}{\zeta\omega_n}$
Time to peak amplitude	$t_p = \dfrac{\pi}{\omega_n\sqrt{1-\zeta^2}}$
Peak overshoot	$M_p = e^{-\zeta\omega_n t_p}$

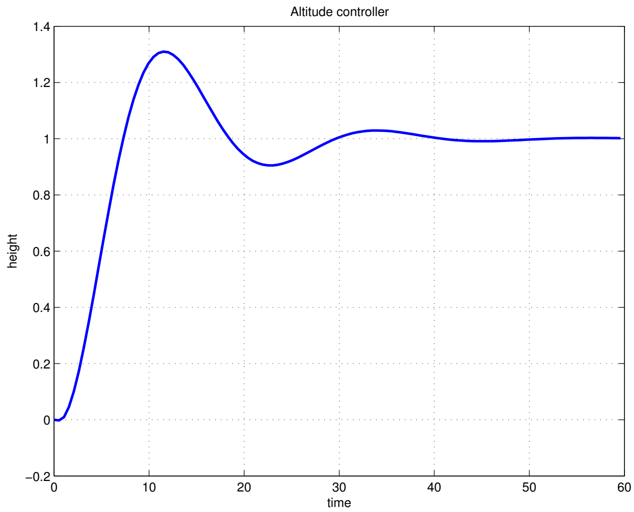


Figure 5: Time response for the altitude controller.

- Predictions: $t_r = 5.2$sec, $t_s = 28.4$sec, $t_p = 11.2$sec, $M_p = 0.3$

- Now with h feedback, things are a little bit better, but not much.
  - Real design is complicated by the location of the zeros
  - Any actuator lag is going to hinder the performance also.

  $\Rightarrow$ Typically must tweak inner loop design for altitude controller to work well.

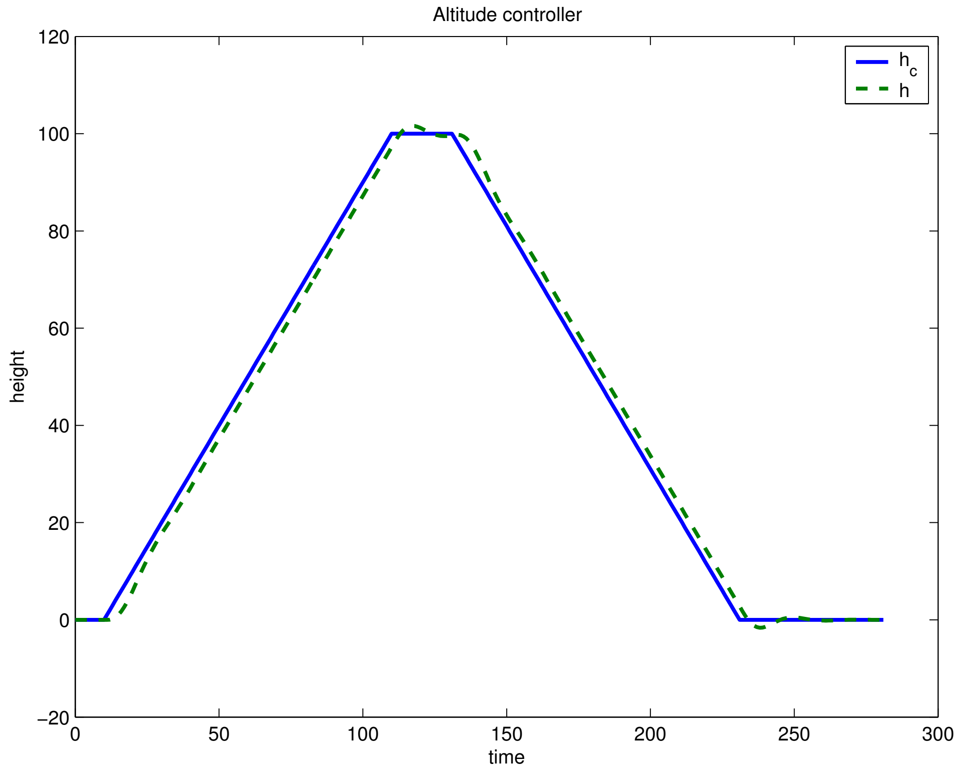


Figure 6: Altitude controller time response for more complicated input.

- The gain K_h must be kept very small since the low frequency poles are heading into the RHP.

$\Rightarrow$ Must take a different control approach to design a controller for the full dynamics.

  – Must also use the throttle to keep tighter control of the speed.

  – Example in Etkin and Reid, page 277. Simulink and Matlab code for this is on the Web page.

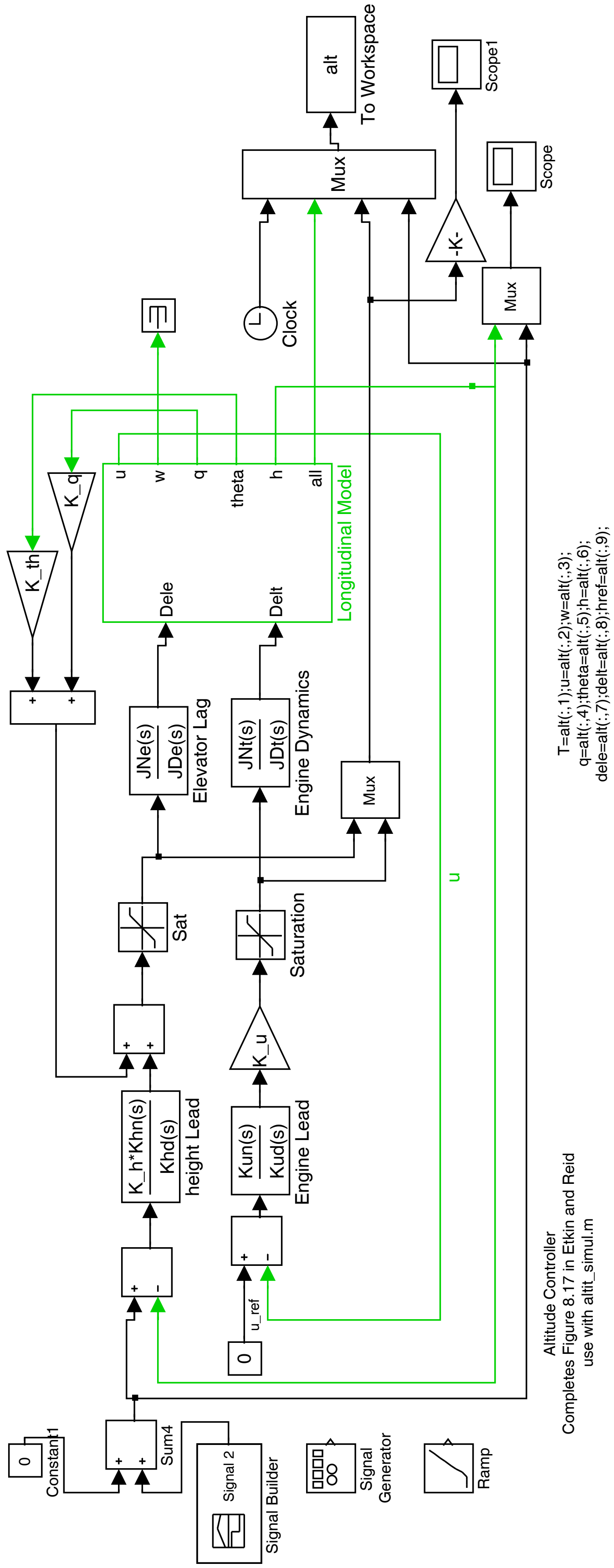


Figure 7: Simulink block diagram for implementing the more advanced version of the altitude hold controller. Follows and extends the example in Etkin and Reid, page 277.

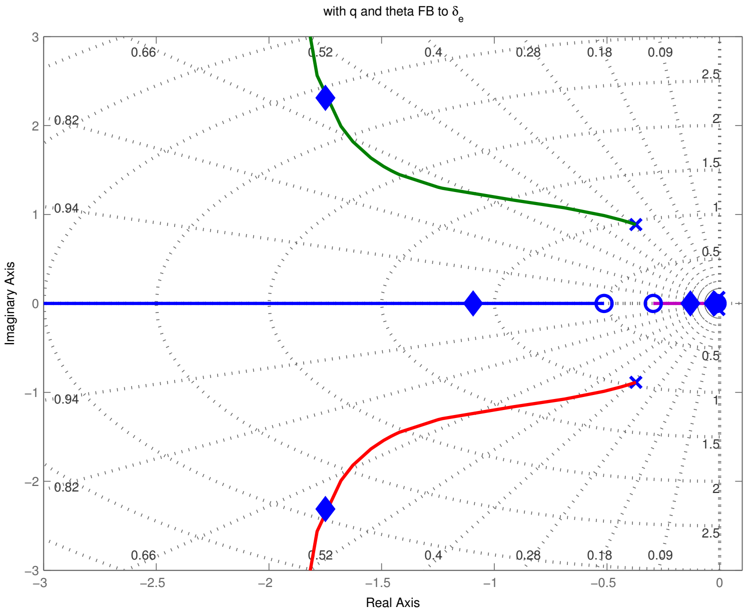


Figure 8: Root loci associated for simulation (q, θ to δ_e).

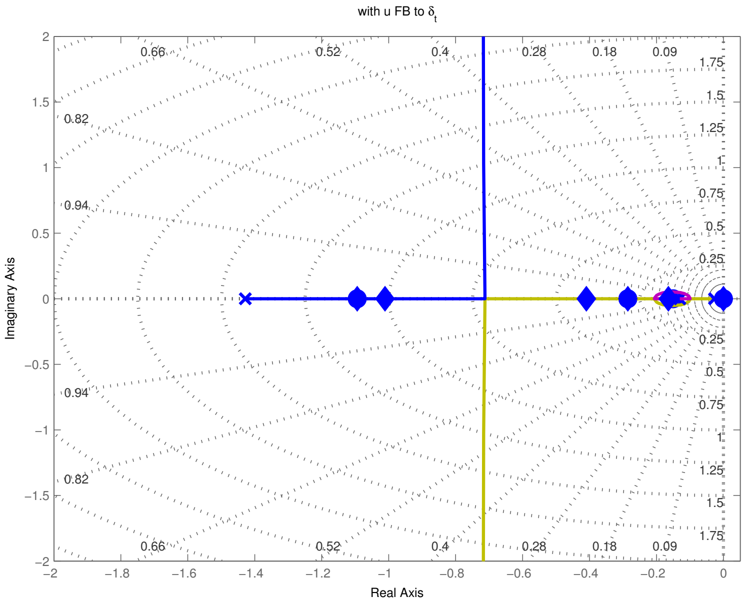


Figure 9: Root loci associated with simulation (u to δ_t). There is more authority in the linear control analysis, but this saturates the nonlinear simulation.

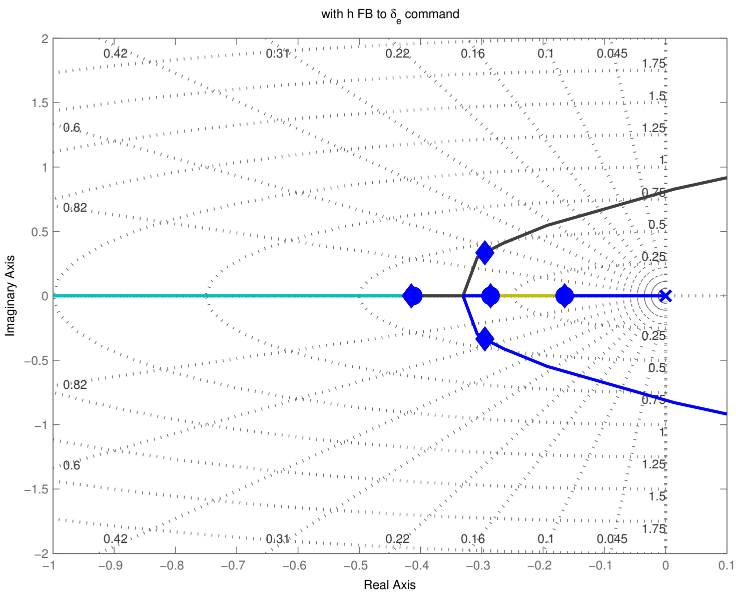


Figure 10: Root loci associated with simulation (h to δ_e). Lead controller

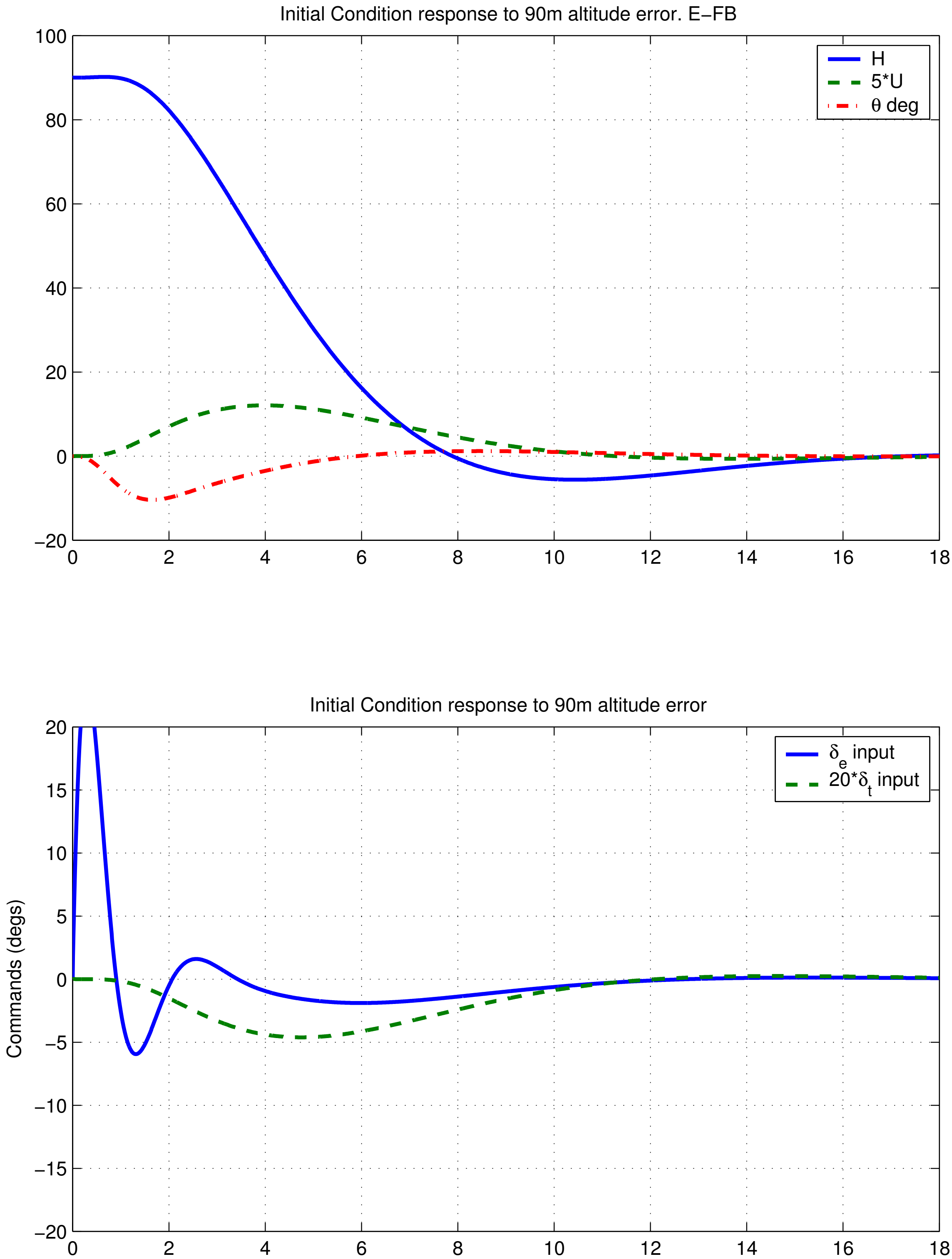


Figure 11: Initial condition response using elevator controller. Note the initial elevator effort and that the throttle is scaled back as the speed picks up.

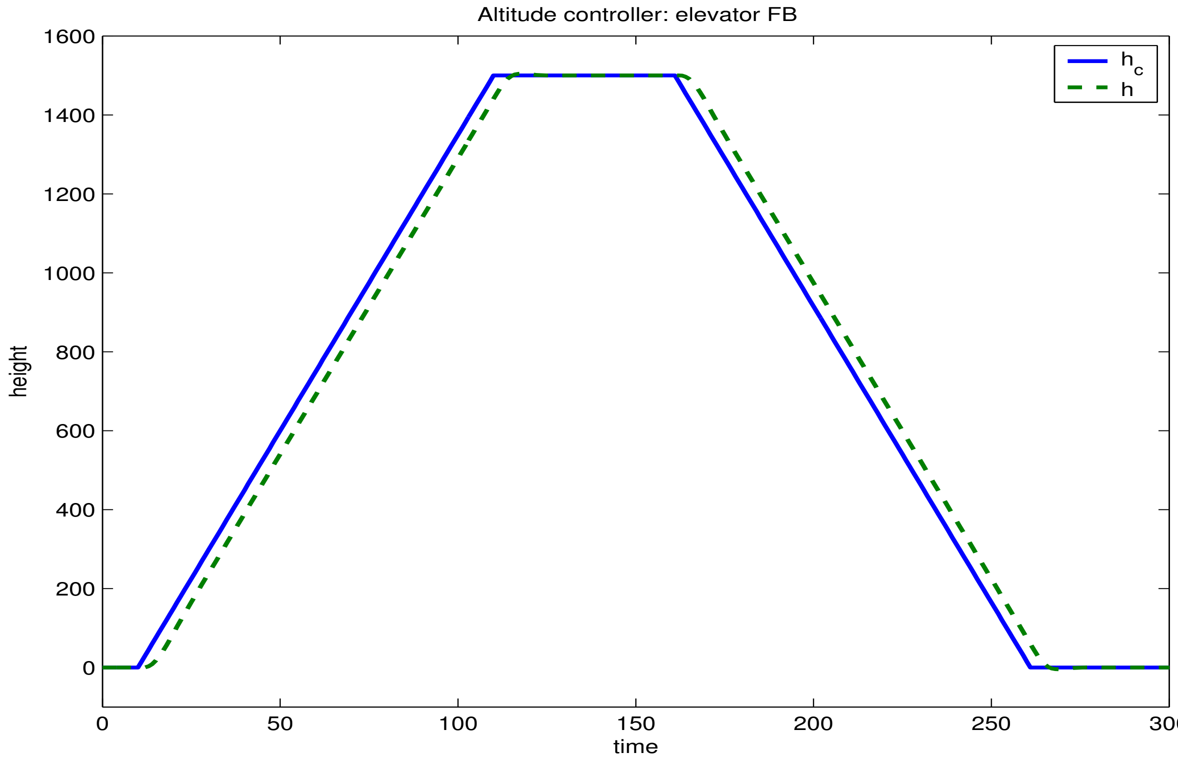


Figure 12: Command following ramps up and down. Offset, but smooth transitions.

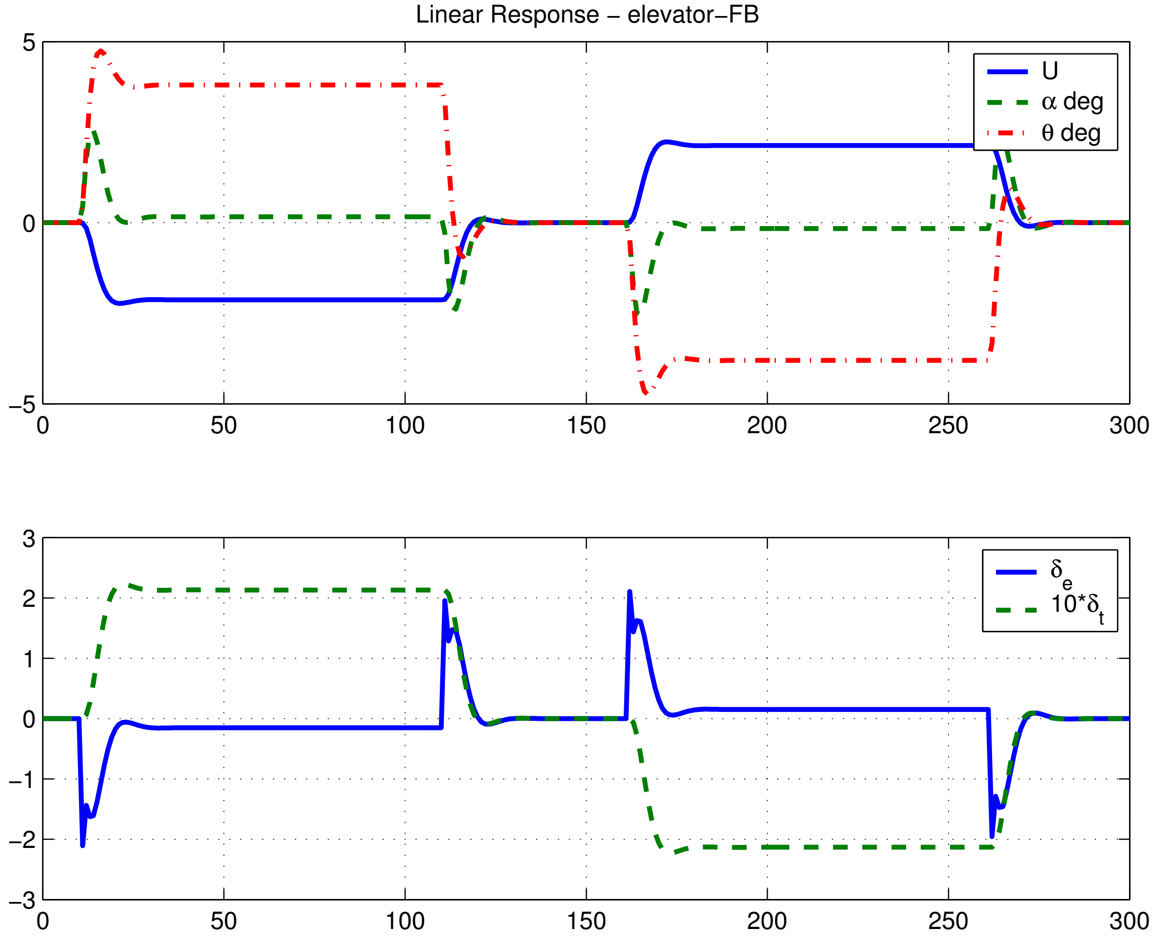


Figure 13: Elevator controller – Overall response and inputs

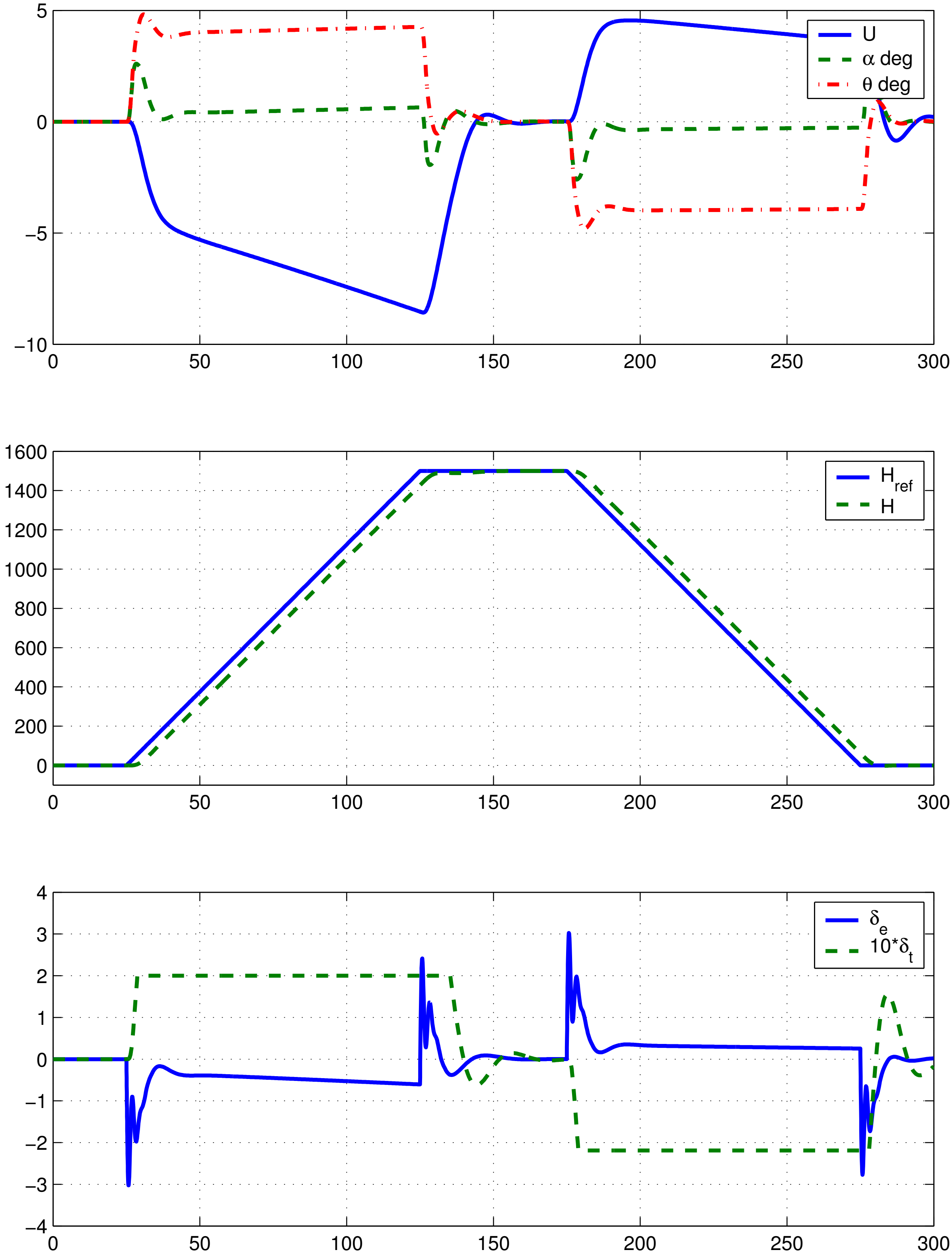


Figure 14: Nonlinear simulation - Note: throttle at the limits, speed not well maintained, but path similar to linear simulation.

- Recall: could have used throttle to control γ directly (see 6–9)
  - Alternative altitude control strategy
  - However, engine lag dynamics are much slower.

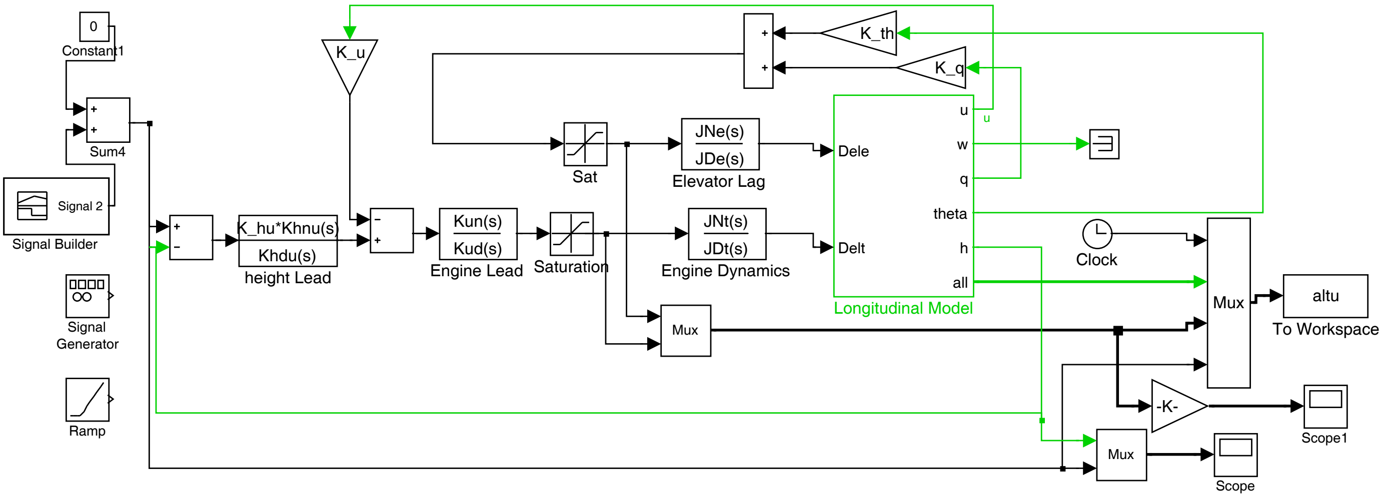


Figure 15: Altitude Controller using FB to the thrust.

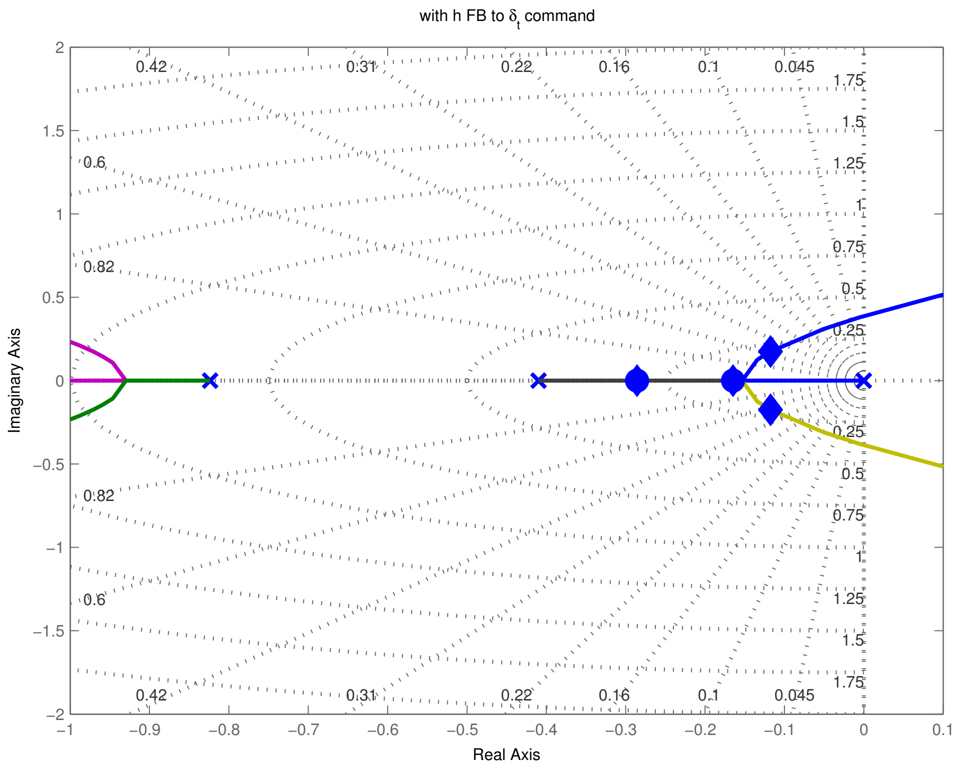
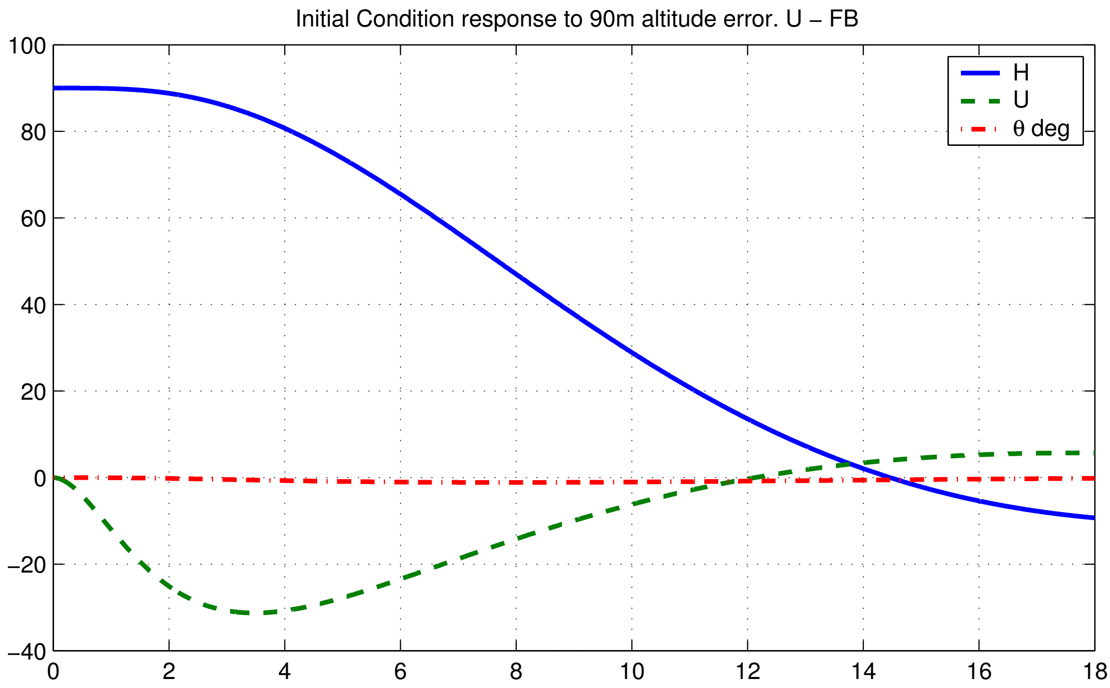


Figure 16: Root loci associated with simulation (h to δ_t). Lead controller



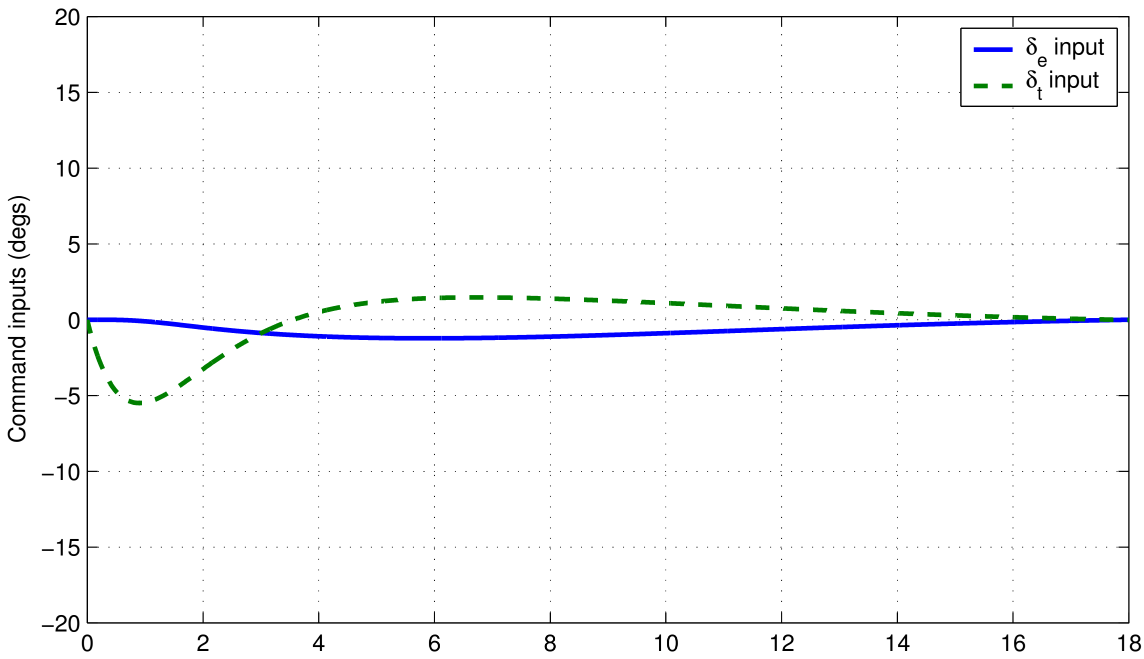


Figure 17: Initial condition response (linear sim) using thrust controller. Slows down to descend. Much slower response, but throttle commands still quite large (beyond saturation of ± 0.2)

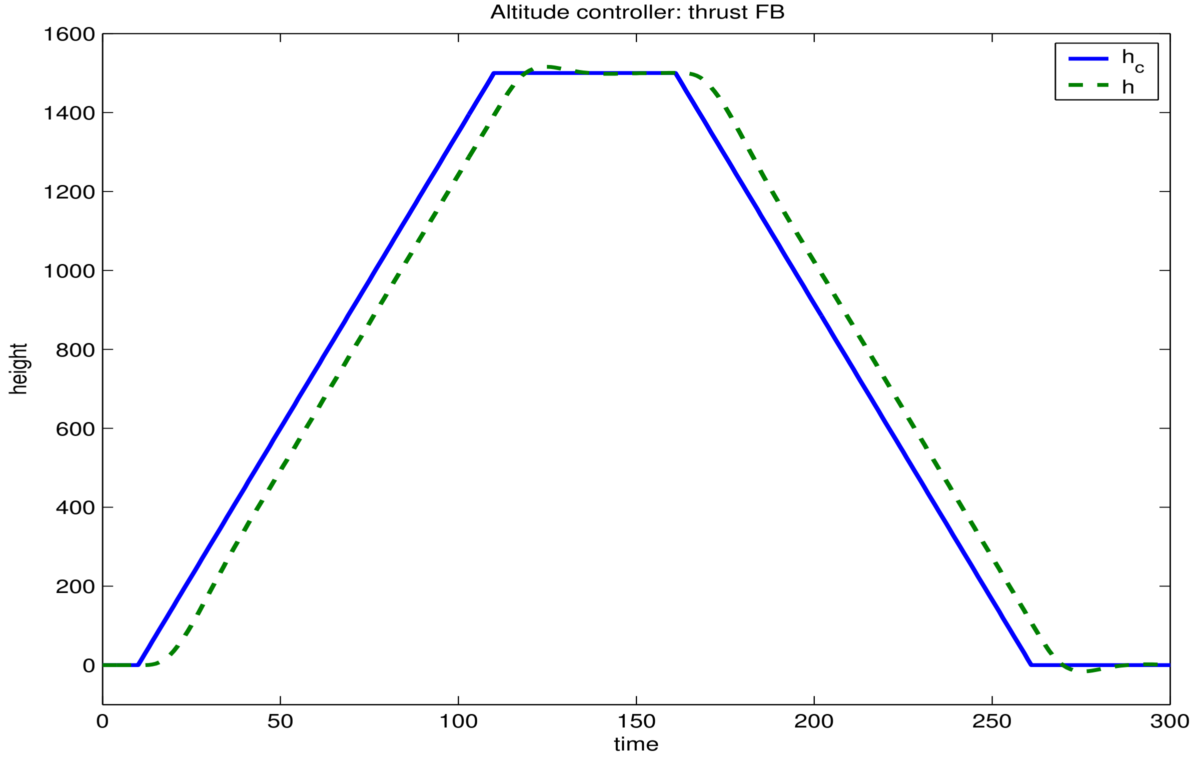


Figure 18: Command following ramps up and down. Offset, but smooth transitions.

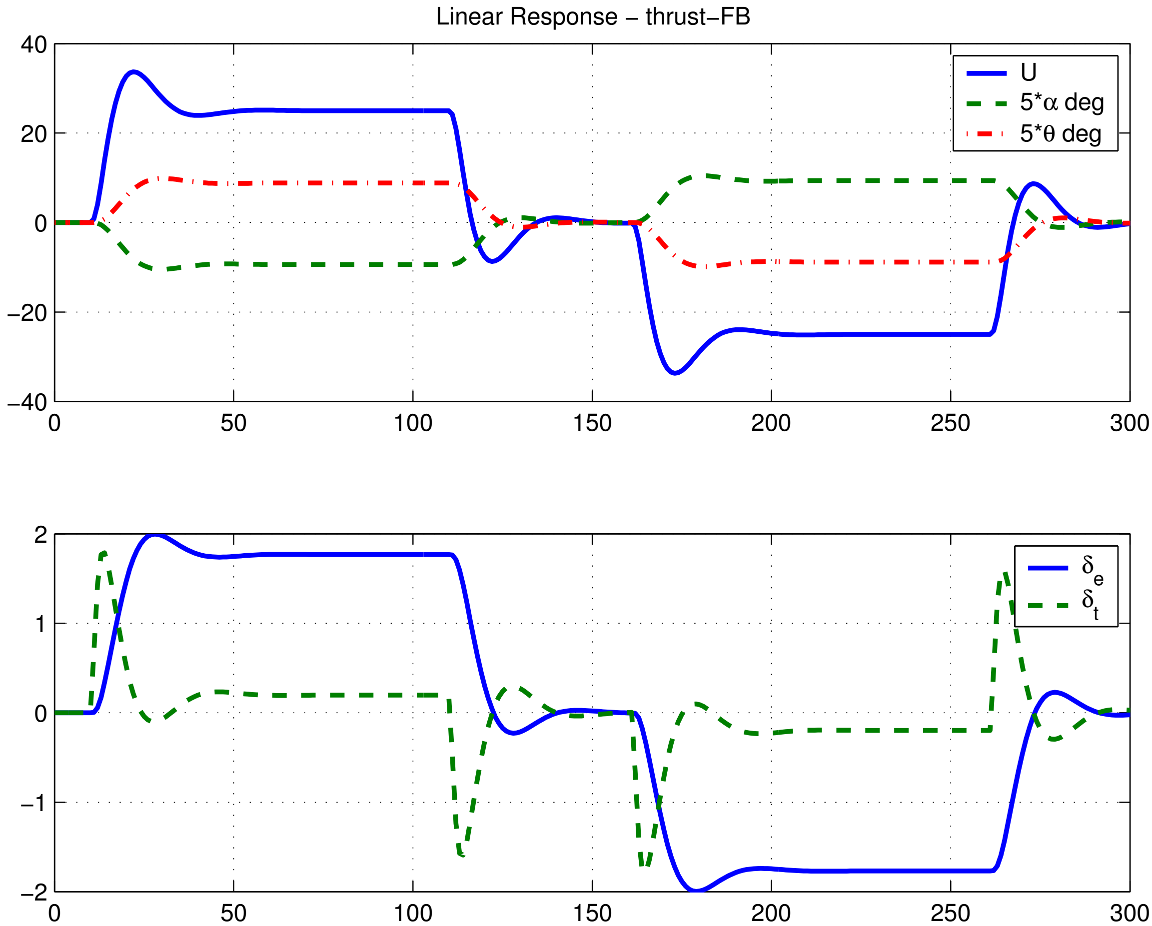


Figure 19: Thrust controller – Overall response and inputs. Linear response looks good, but commands are very large.

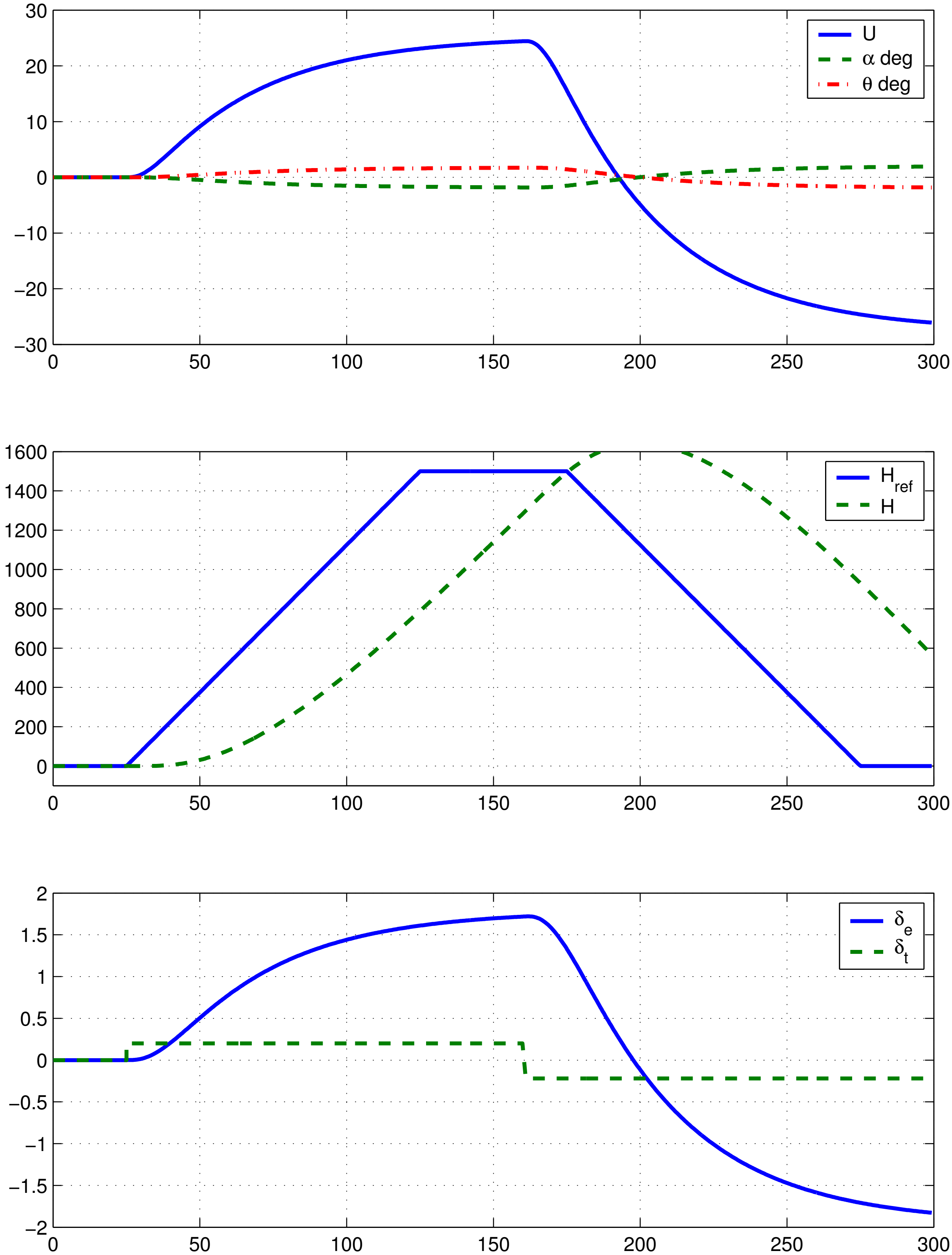


Figure 20: Nonlinear simulation - throttle saturations limit path following performance.

```
%
% altitude hold controller
%
clear all
prt=1;
for ii=1:9
  figure(ii);clf;
  set(gcf,'DefaultLineLineWidth',2);
  set(gcf,'DefaultlineMarkerSize',10)
end

Xu=-1.982e3;Xw=4.025e3;
Zu=-2.595e4;Zw=-9.030e4;Zq=-4.524e5;Zwd=1.909e3;
Mu=1.593e4;Mw=-1.563e5;Mq=-1.521e7;Mwd=-1.702e4;

g=9.81;theta0=0;S=511;cbar=8.324;
U0=235.9;Iyy=.449e8;m=2.83176e6/g;cbar=8.324;rho=0.3045;
Xdp=.3*m*g;Zdp=0;Mdp=0;
Xde=-3.818e-6*(1/2*rho*U0^2*S);Zde=-0.3648*(1/2*rho*U0^2*S);
Mde=-1.444*(1/2*rho*U0^2*S*cbar);;
%
%
% use the full longitudinal model
%
% x=[u w q theta h];
% u=[de;dt];
sen_u=1;sen_w=2;sen_q=3;sen_t=4;sen_h=5;sen_de=6;sen_dt=7;
act_e=1;act_t=2;
%
% dot h = U0 (theta - alpha) = U0 theta - w
%
A=[Xu/m Xw/m 0 -g*cos(theta0) 0;[Zu Zw Zq+m*U0 -m*g*sin(theta0)]/(m-Zwd) 0;
[Mu+Zu*Mwd/(m-Zwd) Mw+Zw*Mwd/(m-Zwd) Mq+(Zq+m*U0)*Mwd/(m-Zwd) ...
     -m*g*sin(theta0)*Mwd/(m-Zwd) 0]/Iyy;
[ 0 0 1 0 0];[0 -1 0 U0 0]];
B=[Xde/m Xdp/m;Zde/(m-Zwd) Zdp/(m-Zwd);(Mde+Zde*Mwd/(m-Zwd))/Iyy ...
  (Mdp+Zdp*Mwd/(m-Zwd))/Iyy;0 0;0 0];
C=[eye(5);zeros(2,5)];
D=[zeros(5,2);[eye(2)]]; %last 2 outputs are the controls
%
% add actuator dynamics
%
% elevator
%tau_e=.1;%sec
tau_e=.25;%sec
JNe=1;JDe=[tau_e 1];
syse=tf(JNe,JDe);
%
% thrust
%
tau_t=3.5;%sec
JNt=1;JDt=[tau_t 1];
syst=tf(JNt,JDt);
sysd=append(syse,syst);
syslong=ss(A,B,C,D);
syslong2=series(sysd,syslong);
[A,B,C,D]=ssdata(syslong2);
na=size(A);
%
% q and theta loops
%
% inner loop on theta/q
% u=kq*q+Kth*theta + de_c
K_th=1;K_q=1.95*K_th;
if 1
   figure(1);clf;%orient tall
   sgrid([.5 .707]',[1]);
   rlocus(tf(-[1.95 1],1)*tf(syslong2(sen_t,act_e)))
   r_th=rlocus(tf(-[1.95 1],1)*tf(syslong2(sen_t,act_e)),K_th)
   hold on
   plot(r_th+eps*j,'bd','MarkerFace','b');axis([-3,1,-3,3]);hold off
   axis([-3 .1 -3 3])
```

```
   title('with q and theta FB to \delta_e');
   if prt
      print -depsc alt_sim1.eps
      jpdf('alt_sim1')
   end
end
% close inner loop
syscl=feedback(syslong2,[K_q K_th],act_e,[sen_q sen_t],1)
[eig(syscl) [nan;r_th;nan]]
[Acl,Bcl,Ccl,Dcl]=ssdata(syscl);
%
% engine loops
% mostly the phugoid ==> design on top of the short period
%
% system with q/th loop feedback
figure(9);clf
% interested in vel output and engine input
sys_u=syscl(sen_u,act_t);
Kun=[1/0.2857 1]; % comp zero on plant pole
Kud=[1/(0.2857*5) 1];
[Au,Bu,Cu,Du]=tf2ss(Kun,Kud);
%K_u=.075;
%K_u=.35;
K_u=.1;
rlocus(sys_u*tf(Kun,Kud));
rr_u=rlocus(sys_u*tf(Kun,Kud),K_u);
hold on;plot(rr_u+eps*j,'bd','MarkerFace','b');hold off;

if 1
   na=size(Acl,1);
   Aclt=[Acl Bcl(:,act_t)*Cu;zeros(1,na)  Au];
   Bclt=[[Bcl(:,act_e);0] [Bcl(:,act_t)*Du;Bu]];
   Cclt=[Ccl zeros(size(Ccl,1),1)];
   Dclt=[zeros(size(Ccl,1),2)];
   figure(2);clf;axis([-.5 .05,-.4,.4]);sgrid([.5 .707]',[.05]);hold on;
   rlocus(Aclt,Bclt(:,act_t),Cclt(sen_u,:),Dclt(sen_u,act_t));
   axis([-2 .05,-2,2])
   r_u=rlocus(Aclt,Bclt(:,act_t),Cclt(sen_u,:),Dclt(sen_u,act_t),K_u)';
   [[r_th;NaN;NaN;NaN] r_u]
   hold on;plot(r_u+eps*j,'bd','MarkerFace','b'),hold off
   title('with u FB to \delta_t')
   if prt
      print -depsc alt_sim2.eps
     jpdf('alt_sim2')
   end
end
Acl2=Aclt-Bclt(:,act_t)*K_u*Cclt(sen_u,:);
Bcl2=Bclt;
Ccl2=Cclt;
Dcl2=Dclt;
sysclt=ss(Acl2,Bcl2,Ccl2,Dcl2);
[eig(sysclt) rr_u]
%
% now close loop on altitude
%
% de_c = kh*(h_c-h)
if 1
    % design lead by canceling troubling plant pole
    % zero located p*8
   tt=eig(sysclt);[ee,ii]=min(abs(tt+.165));
   Khn=[1/abs(tt(ii)) 1];Khd=[1/(8*abs(tt(ii))) 1];
   K_h=-1*.00116;
   Gc_eng=tf(Khn,Khd);
   Loopt=series(append(Gc_eng,tf(1,1)),sysclt);
   figure(3);clf;
   sgrid([.5 .707]',[.1:.1:1]);
   hold on;
   rlocus(sign(K_h)*Loopt(sen_h,act_e));
   axis([-1 .1, -2 2])
   r_h=rlocus(Loopt(sen_h,act_e),K_h)
   hold on;plot(r_h+eps*j,'bd','MarkerFace','b'),hold off
   title('with h FB to \delta_e command')
```

```
    if prt
        print -depsc alt_sim22.eps
       jpdf('alt_sim22')
    end
end
syscl3=feedback(series(append(tf(K_h,1),tf(1,1)),Loopt),[1],act_e,sen_h,-1);
[r_h eig(syscl3)]
%
% try using the thrust act to control altitude
%
if 1
%   K_hu=.154;
    K_hu=0.025;
    tt=eig(sysclt);[ee,ii]=min(abs(tt+.165));
    Khnu=[1/abs(tt(ii)) 1];Khdu=[1/(5*abs(tt(ii))) 1];
    Gc_engu=tf(Khnu,Khdu);
    Loopt=series(append(tf(1,1),Gc_engu),sysclt);
    figure(7);clf;axis([-.6 .1,-.5,.5]);sgrid([.5 .707]',[.05]);hold on;
    rlocus(sign(K_hu)*Loopt(sen_h,act_t));
    axis([-1 .1,-2,2])
    r_hu=rlocus(Loopt(sen_h,act_t),K_hu)
    hold on;plot(r_hu+eps*j,'bd','MarkerFace','b'),hold off
    title('with h FB to \delta_t command')
    if prt
        print -depsc alt_sim22a.eps
       jpdf('alt_sim22a')
    end
end
syscl4=feedback(series(append(tf(1,1),tf(K_hu,1)),Loopt),[1],act_t,sen_h,-1);
[r_hu eig(syscl4)]

t=[0:.01:18]';
[ystep,t]=initial(syscl3,[0 0 0 0 90 0 0 0 0],t);
figure(4);clf;orient tall;
subplot(211)
ll=1:length(t);
U=ystep(:,sen_u);W=ystep(:,sen_w);q=ystep(:,sen_q);
TH=ystep(:,sen_t);H=ystep(:,sen_h);
DELe=ystep(:,sen_de);DELt=ystep(:,sen_dt);
plot(t(ll),H(ll),t(ll),5*U(ll),t(ll),TH(ll)*180/pi);setlines(2)
title('Initial Condition response to 90m altitude error. E-FB')
legend('H','5*U','\theta deg');grid
subplot(212)
plot(t(ll),DELe(ll)*180/pi,t(ll),20*DELt(ll));
setlines(2)
legend('\delta_e input','20*\delta_t input');grid
axis([0 18 -20 20])
ylabel('Commands (degs)')
title('Initial Condition response to 90m altitude error')
if prt
    print -depsc alt_sim3.eps
    jpdf('alt_sim3')
end
%
t=[0:.01:18]';
[ystep,t]=initial(syscl4,[0 0 0 0 90 0 0 0 0],t);
figure(14);clf;orient tall;
subplot(211)
ll=1:length(t);
U=ystep(:,sen_u);W=ystep(:,sen_w);q=ystep(:,sen_q);
TH=ystep(:,sen_t);H=ystep(:,sen_h);
DELe=ystep(:,sen_de);DELt=ystep(:,sen_dt);
plot(t(ll),H(ll),t(ll),U(ll),t(ll),TH(ll)*180/pi);setlines(2)
title('Initial Condition response to 90m altitude error. U - FB')
legend('H','U','\theta deg');grid
subplot(212)
plot(t(ll),DELe(ll)*180/pi,t(ll),DELt(ll));
setlines(2)
legend('\delta_e input','\delta_t input');grid
axis([0 18 -20 20])
ylabel('Command inputs (degs)')
if prt
```

```
   print -depsc alt_sim3a.eps
   jpdf('alt_sim3a')
end

figure(5);clf
%Path to follow
Hmax=1500;
Hc=[zeros(10,1);Hmax/100*[0:1:100]';Hmax*ones(50,1);Hmax/100*[100:-1:0]';zeros(50,1)];
T=0:length(Hc)-1;
[Yh,T]=lsim(syscl3(:,1),Hc,T);
U=Yh(:,sen_u);W=Yh(:,sen_w);q=Yh(:,sen_q);TH=Yh(:,sen_t);H=Yh(:,sen_h);
DELe=Yh(:,sen_de);DELt=Yh(:,sen_dt);
plot(T,[Hc],T,H,'--');setlines(2)
axis([0 300 -100 1600])
legend('h_c','h')
title('Altitude controller: elevator FB')
ylabel('height')
xlabel('time')
if prt
   print -depsc alt_sim4.eps
   jpdf('alt_sim4')
end

figure(6);clf;
subplot(211)
ll=1:length(T);
plot(T(ll),U(ll),T(ll),W(ll)/U0*180/pi,T(ll),TH(ll)*180/pi);setlines(2)
title('Linear Response - elevator-FB')
legend('U','\alpha deg','\theta deg');grid
axis([0 300 -5 5])
subplot(212)
plot(T(ll),DELe(ll)*180/pi,T(ll),10*DELt(ll));
setlines(2)
axis([0 300 -3 3])
legend('\delta_e','10*\delta_t');grid
%axis([0 250 -4 4])
if prt
   print -depsc alt_sim5.eps
   jpdf('alt_sim5')
end

figure(15);clf
%Path to follow
Hmax=1500;
Hc=[zeros(10,1);Hmax/100*[0:1:100]';Hmax*ones(50,1);Hmax/100*[100:-1:0]';zeros(50,1)];
T=0:length(Hc)-1;
[Yh,T]=lsim(syscl4(:,2),Hc,T);
U=Yh(:,sen_u);W=Yh(:,sen_w);q=Yh(:,sen_q);TH=Yh(:,sen_t);H=Yh(:,sen_h);
DELe=Yh(:,sen_de);DELt=Yh(:,sen_dt);
plot(T,[Hc],T,H,'--');setlines(2)
axis([0 300 -100 1600])
legend('h_c','h')
title('Altitude controller: thrust FB')
ylabel('height')
xlabel('time')
if prt
   print -depsc alt_sim4a.eps
   jpdf('alt_sim4a')
end

figure(16);clf;
subplot(211)
ll=1:length(T);
plot(T(ll),U(ll),T(ll),5*W(ll)/U0*180/pi,T(ll),5*TH(ll)*180/pi);setlines(2)
title('Linear Response - thrust-FB')
legend('U','5*\alpha deg','5*\theta deg');grid
axis([0 300 -40 40])
subplot(212)
plot(T(ll),DELe(ll)*180/pi,T(ll),DELt(ll));
setlines(2)
legend('\delta_e','\delta_t');grid
axis([0 300 -2 2])
```

```
%axis([0 250 -4 4])
if prt
   print -depsc alt_sim5a.eps
   jpdf('alt_sim5a')
end

return

sT=alt(:,1);u=alt(:,2);w=alt(:,3);
q=alt(:,4);theta=alt(:,5);h=alt(:,6);
dele=alt(:,7);delt=alt(:,8);href=alt(:,9);

figure(17);clf;orient tall
ll=1:length(sT)-2;
subplot(311)
title('Simulink hard path following: elevator FB')
plot(sT(ll),u(ll),sT(ll),w(ll)/U0*180/pi,sT(ll),theta(ll)*180/pi);setlines(2)
legend('U','\alpha deg','\theta deg');grid
subplot(312)
plot(sT(ll),[href(ll) h(ll)])
setlines(2)
legend('H_{ref}','H');grid
axis([0 300 -100 1600])
subplot(313)
plot(sT(ll),dele(ll)*180/pi,sT(ll),10*delt(ll));
setlines(2)
legend('\delta_e','10*\delta_t');grid
%axis([0 60 -1000 600])
if prt
   print -depsc alt_sim6.eps
   jpdf('alt_sim6')
end

sTu=altu(:,1);uu=altu(:,2);wu=altu(:,3);
qu=altu(:,4);thetau=altu(:,5);hu=altu(:,6);
deleu=altu(:,7);deltu=altu(:,8);hrefu=altu(:,9);

figure(18);clf;orient tall
ll=1:length(sTu)-2;
subplot(311)
title('Simulink hard path following: thruster FB')
plot(sTu(ll),uu(ll),sTu(ll),wu(ll)/U0*180/pi,sTu(ll),thetau(ll)*180/pi);setlines(2)
legend('U','\alpha deg','\theta deg');grid
subplot(312)
plot(sTu(ll),[hrefu(ll) hu(ll)])
setlines(2)
legend('H_{ref}','H');grid
axis([0 300 -100 1600])
subplot(313)
plot(sTu(ll),deleu(ll)*180/pi,sTu(ll),deltu(ll));
setlines(2)
legend('\delta_e','\delta_t');grid
%axis([0 60 -1000 600])
if prt
   print -depsc alt_sim6a.eps
   jpdf('alt_sim6a')
end
```

Summary

- Multi-loop closure complicated
  - Requires careful consideration of which sensors and actuators to use together.
  - Feedback gain selection often unclear until full controller is done - balance between performance and control authority (saturation).
  - Design of successive loops can interact.

- Could design it by hand, identify good closed-loop pole locations, and then use state feedback controller to design a fully integrated controller.

- Nonlinear simulations important to at least capture impact of the saturations.

Autolanding Controller

- A related, but even more complicated scenario is the problem of landing an aircraft onto a runway
  - Consider the longitudinal component only, but steering to get aligned with the runway is challenging too.

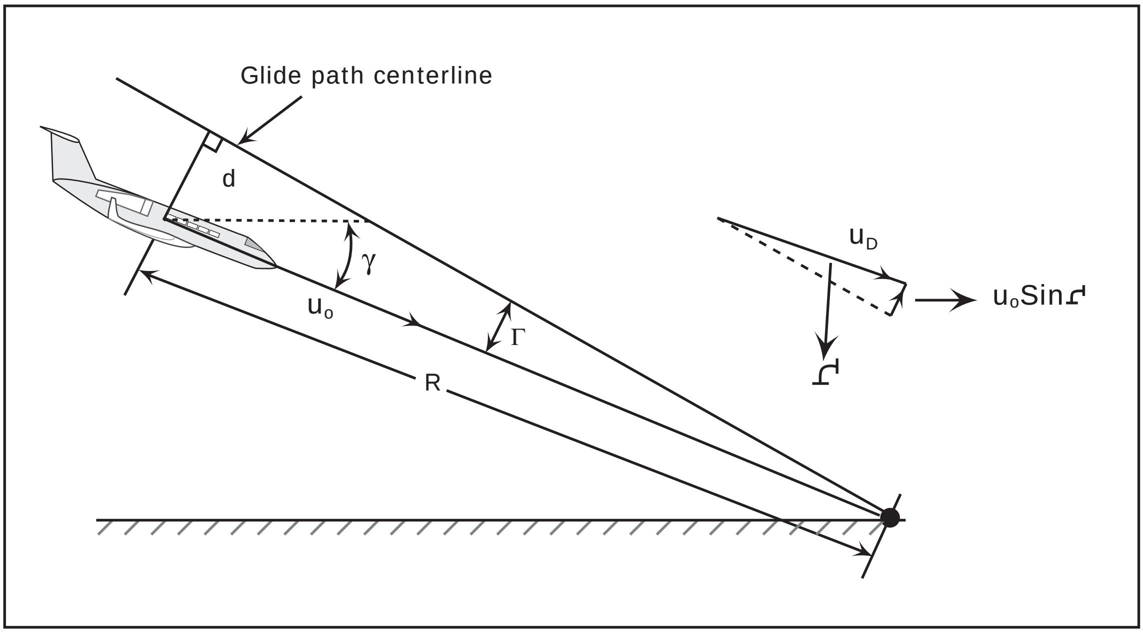


Figure 21: Glide slope basic terminology. Runway is to the right, and the glide slope intersects the ground there. Aircraft is distance R from the runway.

- Aircraft below intended flight path (i.e. offset $d > 0$)
- Desired glide slope $\gamma_r < 0$, and actual glide slope given by $\gamma < 0$
- Difference between desired and actual glide slopes:

$$\Gamma = \gamma - \gamma_r$$

  In this configuration, $\gamma_r \approx -3^\circ$ and $\gamma \approx -2^\circ$, so $\Gamma \approx 1^\circ > 0$.

- Deviation d grows as a function of Γ – projection of U_0 onto a normal of the desired glide slope

$$\dot{d} = -U_0 \sin \Gamma = -U_0 \sin(\gamma - \gamma_r) \approx U_0(\gamma_r - \gamma)$$

  - $\Gamma > 0$ reduces d

- Along the glide path, we must control the pitch attitude and speed
  - Large number of loops must be closed.
  - Similar to altitude controller, but now concerned about how well we track the *ramp slope down* $\Rightarrow$ will also use feedback of separation distance d.

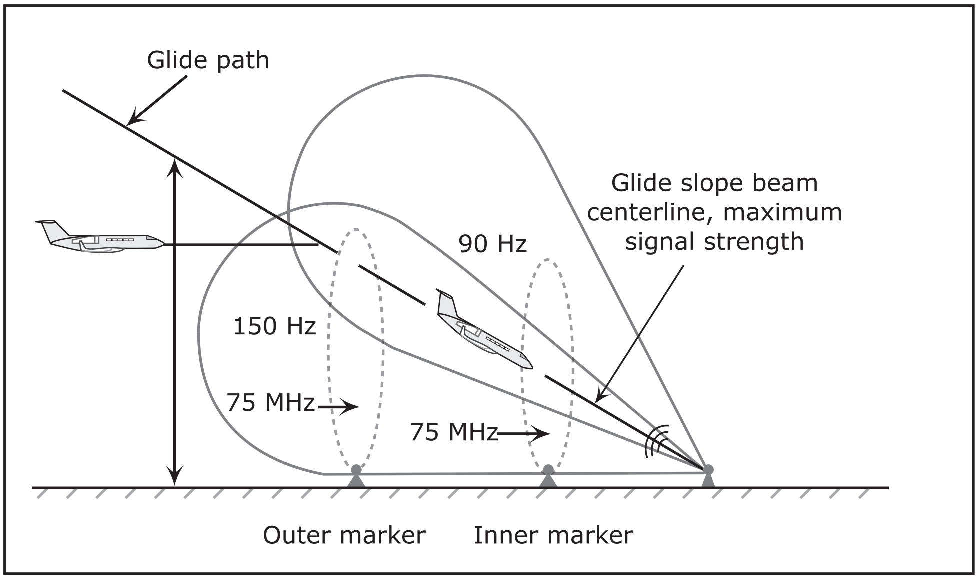


Figure 22: Coded signals from two stations at 90 & 150 MHz indicate whether you are above or below the desired glide slope. The 75 MHz signals (outer/inner markers) give a 4 mile and 3500 ft warning.

- Can measure the angle deviation from the glide slope
  - Crudely using lights.
  - More precisely using an ILS (instrument landing system).
- If d not directly measurable, the can write that

$$\sin\Gamma = \frac{d}{R} \;\Rightarrow\; \Gamma \approx \frac{d}{R}$$

  and do feedback on Γ, which is measurable from ILS.
  - Controller gains must then change with range R. Complex, but feasible (*gain scheduled control*)
- With GPS can measure the vehicle X, Y, Z position to (2-5m)

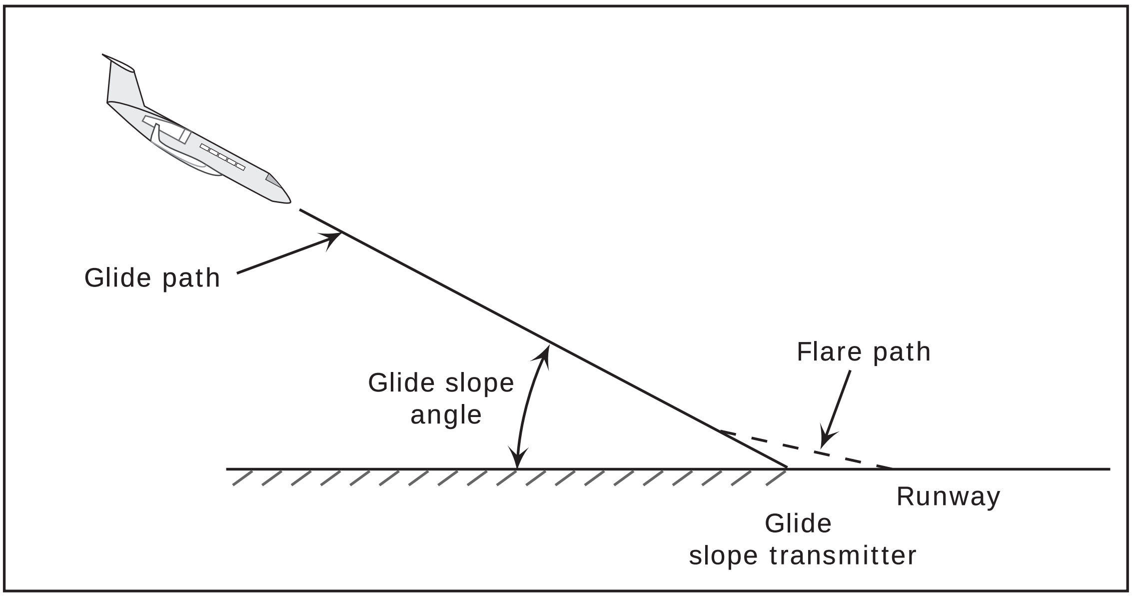


Figure 23: Flare path.

- Other complications in this control problem:
  - We cannot just fly into the ground along this glide slope. The vertical velocity (sink rate) is too high (10 ft/sec).
    $\Rightarrow$ Need to *flare* to change the sink rate to a level that is consistent with the capabilities of the landing gear (2 ft/sec).
  - Aircraft configuration changes when we deploy the flaps and slats, so we need to change our model of the dynamics.

- **Flare Control:** At some decision height ($\approx$70 ft) we will change the desired trajectory away from *glide slope following* to direct *control of the altitude*.
  - Want the height $h(t)$ to follow a smooth path to the ground

$$h_{\text{ref}}(t) = h_0 \; e^{-t/\tau} \; , \; \tau = 3 - 10 \text{ sec}$$

    which implies that $\dot{h}_{\text{ref}} = -h_{\text{ref}}/\tau$.
  - Need to find the decision height h_0.

- We start the flare when the height above the runway matches the value we would get from the flare calculation using the current descent rate $\dot{h}$, i.e. when

$$h(t) = -\tau \dot{h}(t) = \tau U_0 \sin(-\gamma_r) > 0$$

  – Defines the decision height h_0

  – Ensures a smooth transition at the start of the flare.

- Landing the aircraft requires a complex combination of δ_e and δ_t to coordinate the speed/pitch/height control.

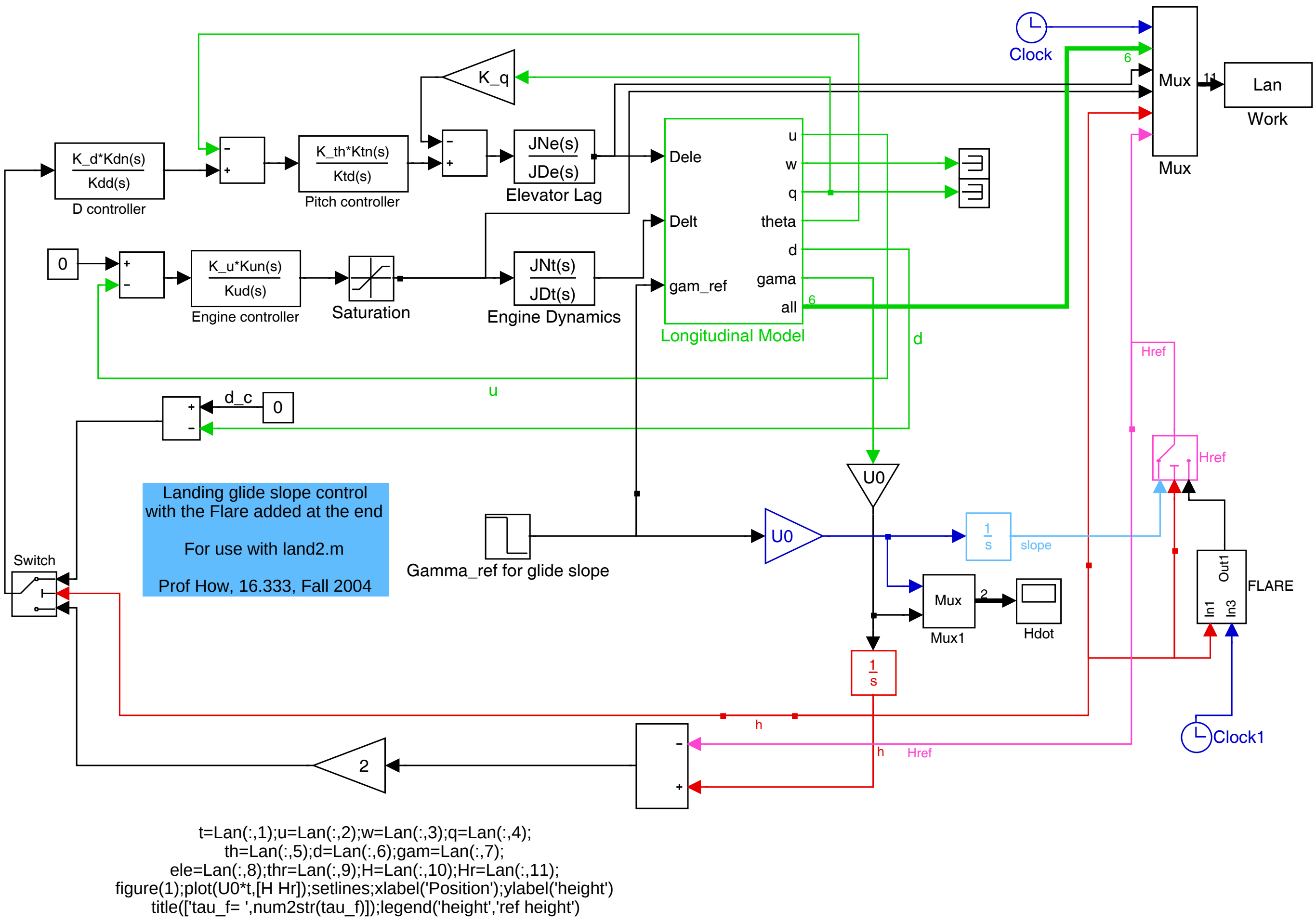


Figure 24: Simulink block diagram for implementing the autoland controller. Follows and extends the example in Etkin and Reid, page 277.

- Simulation based on the longitudinal model with elevator and throttle lags. (See also Stevens and Lewis, page 300)
  - Tight q and θ loops
  - Engine loop designed using Phugoid model, tries to maintain constant speed.

- Simulation explicitly switches the reference commands to the controller when we reach the decision height.
  - Could change the controller gains, but current code does not.
  - Above decision height, feedback is on d with the commands used to generate reference pitch commands.

- At the decision height, the control switches in two ways:
  - Flare program initiated $\Rightarrow$ generates h_{ref}.
  - Switch from d feedback to feedback on altitude error $h - h_{\text{ref}}$.

- With $\tau_f = 8$, start flare at about 77 ft (60 sec). Need tighter control to lower τ_f further.

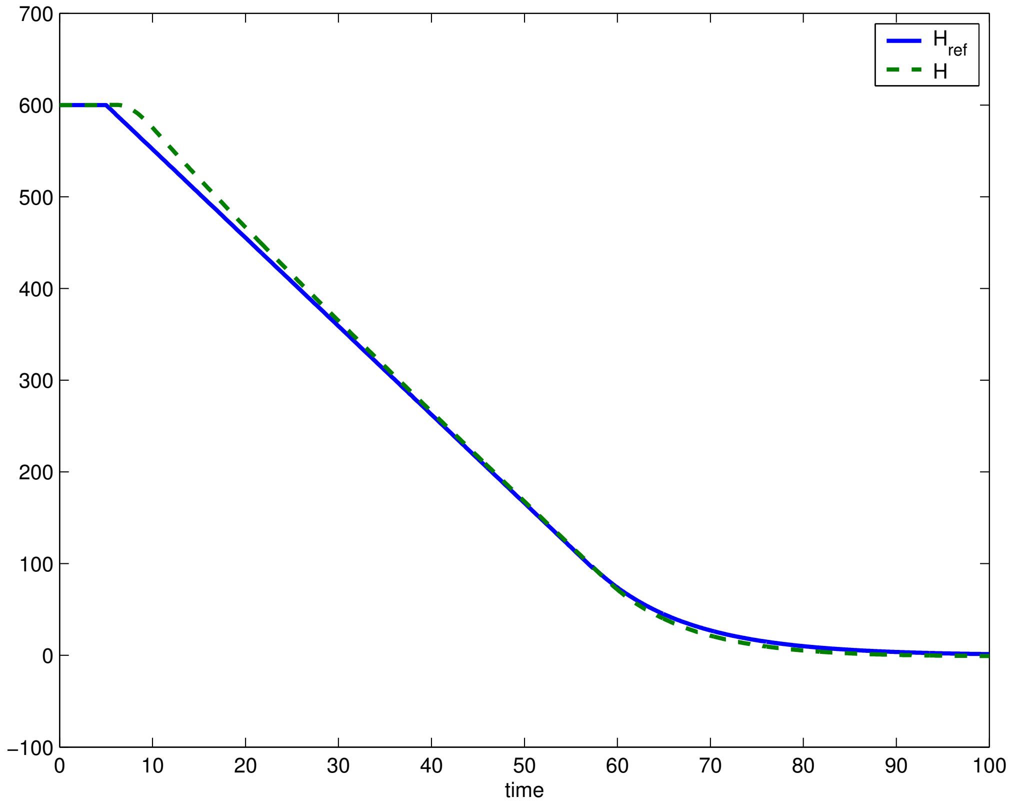


Figure 25: Typical results from the simulink block diagram. Slight initial error, but then the aircraft locks onto the glideslope. Follows flare ok, but not great.

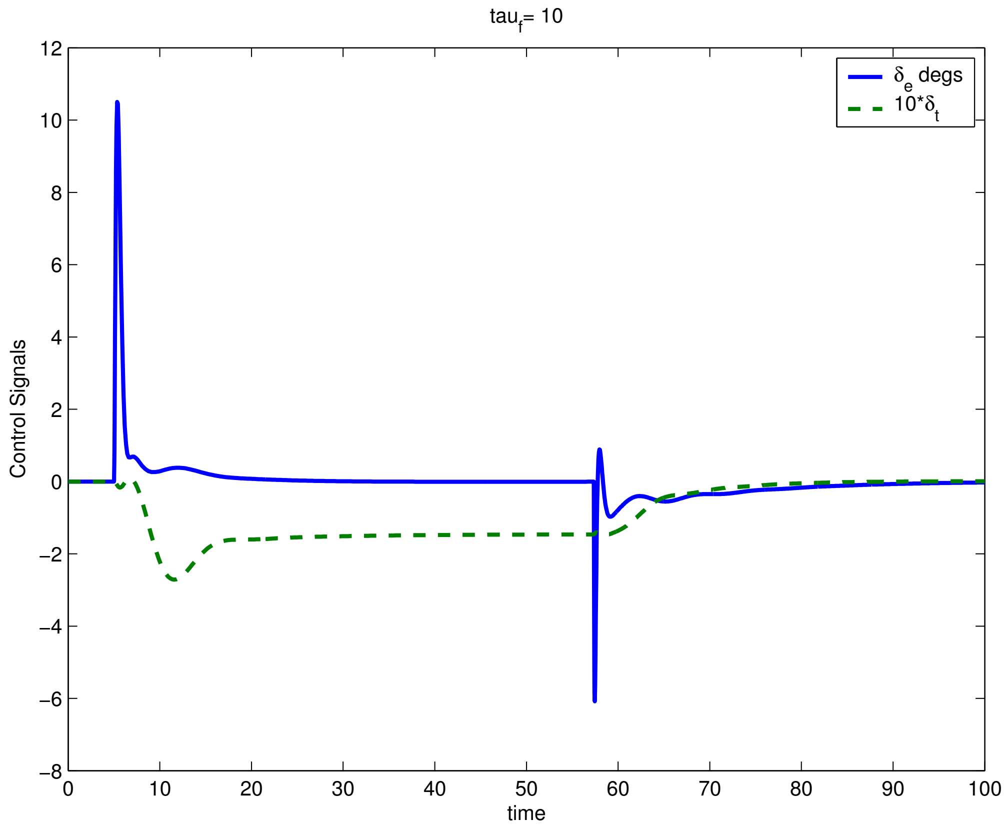


Figure 26: Control signals. Positive elevator pitches aircraft over to initiate descent, and then elevator up (negative) at 60sec pitches nose up as part of the flare.

Landing Code

```
%
% landing Case I (etkin and reid)
% for use within landing3.mdl
%
% from reid
clear all
prt=1;
for ii=1:9
  figure(ii);clf;
  set(gcf,'DefaultLineLineWidth',2);
  set(gcf,'DefaultlineMarkerSize',10)
end
% case 1, page 371 reid
Xub=-3.661e2;Xwb=2.137e3;Xqb=0;Xwdb=0;
Zub=-3.538e3;Zwb=-8.969e3;Zqb=-1.090e5;Zwdb=5.851e2;
Mub=3.779e3;Mwb=-5.717e4;Mqb=-1.153e7;Mwdb=-7.946e3;
Xdeb=1.68e4;Zdeb=-1.125e5;Mdeb=-1.221e7;

g=32.2;U0=221;Iyy=3.23e7;theta0=0;S=5500;m=5.640e5/g;cbar=27.31;

alphae=-8.5*pi/180;
CA=cos(alphae);SA=sin(alphae);
%
% convert to stability axes - page 356
%
Xu=Xub*CA^2+Zwb*SA^2+(Xwb+Zub)*SA*CA;
Xw=Xwb*CA^2-Zub*SA^2-(Xub-Zwb)*SA*CA;
Zu=Zub*CA^2-Xwb*SA^2-(Xub-Zwb)*SA*CA;
Zw=Zwb*CA^2+Xub*SA^2-(Xwb+Zub)*SA*CA;
Mu=Mub*CA+Mwb*SA;
Mw=Mwb*CA-Mub*SA;
Xq=Xqb*CA+Zqb*SA;
Zq=Zqb*CA-Xqb*SA;
Mq=Mqb;
Xwd=Xwdb*CA^2+Zwdb*SA*CA;
Zwd=Zwdb*CA^2-Xwdb*SA*CA;
Mwd=Mwdb*CA;
Xde=Xdeb*CA+Zdeb*SA;
Zde=Zdeb*CA-Xdeb*SA;
Mde=Mdeb;
Xdp=.3*m*g;Zdp=0;Mdp=0;

%x=[u w q th d]
%
% full longitudinal model
%
AL=[Xu/m Xw/m Xq/m -g*cos(theta0) 0;[Zu Zw Zq+m*U0 -m*g*sin(theta0)]/(m-Zwd) 0;
   [Mu+Zu*Mwd/(m-Zwd) Mw+Zw*Mwd/(m-Zwd) Mq+(Zq+m*U0)*Mwd/(m-Zwd) ...
         -m*g*sin(theta0)*Mwd/(m-Zwd)]/Iyy 0;
   [ 0 0 1 0 0];[0 1 0 -U0 0]]; %changed
BL=[Xde/m Xdp/m 0;Zde/(m-Zwd) Zdp/(m-Zwd) 0;(Mde+Zde*Mwd/(m-Zwd))/Iyy ...
      (Mdp+Zdp*Mwd/(m-Zwd))/Iyy 0;0 0 0;0 0 U0]; %changed
if 1
   [V,ev]=eig(AL);ev=diag(ev);D=V;
        %
        % Short-period Approx
        %
    Asp=[Zw/m U0;
                        [Mw+Zw*Mwd/m Mq+U0*Mwd]/Iyy];
        Bsp=[Zde/m;(Mde+Zde/m*Mwd)/Iyy];
                [nsp,dsp]=ss2tf(Asp,Bsp,eye(2),zeros(2,1));
             [Vsp,evsp]=eig(Asp);evsp=diag(evsp);
        Ap=[Xu/m -g;-Zu/(m*U0) 0];
        Bp=[(Xde-(Xw/Mw)*Mde)/m Xdp/m-Xw/Mw*Mdp/m;(-Zde+(Zw/Mw)*Mde)/m/U0 (-Zdp+(Zw/Mw)*Mdp)/m/U0];
                [Vp,evp]=eig(Ap);evp=diag(evp);
end
% x=[u w q theta d];
```

```
% u=[de;dt];
sen_u=1;sen_w=2;sen_q=3;sen_t=4;sen_d=5;sen_g=6;
act_e=1;act_t=2;act_gr=3;

CL=[eye(5);0 -1/U0 0 1 0]; %adds a gamma output
DL=zeros(6,3);
%
% y=[u w q th d gamma]
%
% CONTROL
% add actuator dynamics
%
% elevator
tau_e=.1;%sec
JNe=1;JDe=[tau_e 1];
syse=tf(JNe,JDe);
%
% thrust
%
tau_t=3.5;%sec
JNt=1;JDt=[tau_t 1];
syst=tf(JNt,JDt);
sysd=append(syse,syst,1);
syslong=ss(AL,BL,CL,DL);
syslong2=series(sysd,syslong);
[A,B,C,D]=ssdata(syslong2);
na=size(A);
%
% q and theta loops
% using full model
% inner loop on theta/q
% u=kq*q+Kth*theta + de_c
%
% higher BW than probably needed
% numerator from zero placement
% and den from pole at 0 and one at 10
%
K_th=-6;K_q=-1.5;
if 1
   Ktn=[1 1.4 1];Ktd=conv([1/5 1],[1 0]);
   [Ath,Bth,Cth,Dth]=tf2ss(Ktn,Ktd);
   na=size(A,1);nt=size(Ath,1);
   At=[A-B(:,act_e)*K_q*C(sen_q,:) B(:,act_e)*Cth;zeros(nt,na)  Ath];
   Bt=[[B(:,act_e)*Dth;Bth] [B(:,[act_t act_gr]);zeros(nt,2)]];
   Ct=[C zeros(size(C,1),nt)];
   Dt=[zeros(size(C,1),size(B,2))];

   figure(1);clf;sgrid([.5 .707]',[1]);hold on;
   rlocus(At,-Bt(:,act_e),Ct(sen_t,:),Dt(sen_t,act_e));
   r_th=rlocus(At,Bt(:,act_e),Ct(sen_t,:),Dt(sen_t,act_e),K_th)';
   plot(r_th+eps*j,'bd','MarkerFace','b');axis([-1.5,.1,-.1,1.2]);hold off
   title('with q and theta FB to dele')
   if prt
      print -depsc land2_1.eps
      jpdf('land2_1')
   end
end
Acl=At-Bt(:,act_e)*K_th*Ct(sen_t,:);
Bcl=[Bt(:,act_e) Bt(:,[act_t act_gr])]; % act_e now a ref in for theta
Ccl=Ct;
Dcl=[Dt(:,act_e) Dt(:,[act_t act_gr])];

if 0
figure(3);clf
K_th=-6;K_q=-1.5;
Ktn=[1 1.4 1];Ktd=conv([1/5 1],[1 0]);
rlocus([K_q K_th*ss(tf(Ktn,Ktd))]/norm(K_th)*syslong2([sen_q sen_t],act_e));axis([-2 1 -1 1])    ;
rr_qq=rlocus([K_q K_th*ss(tf(Ktn,Ktd))]*syslong2([sen_q sen_t],act_e),1);
hold on;
plot(rr_qq+eps*sqrt(-1),'gd')
plot(eig(Acl)+eps*sqrt(-1),'bs')
hold off
```

```
figure(4);clf
Ktn=[1];Ktd=[1];
Ktn=[1 2*0.7*2 4];Ktd=conv([1/5 1],[1 0]);
K_th=-6;K_q=-3;
rlocus([K_q K_th*ss(tf(Ktn,Ktd))]/norm(K_th)*syslong2([sen_q sen_t],act_e));axis([-2 1 -1 1]*2)    ;
rr_qq=rlocus([K_q K_th*ss(tf(Ktn,Ktd))]*syslong2([sen_q sen_t],act_e),1);
hold on;
plot(rr_qq+eps*sqrt(-1),'gd')
plot(eig(Acl)+eps*sqrt(-1),'bs')
hold off

end


%
%
% engine loops
% mostly the phugoid ==> design on top of the short period
%
syst=tf(JNt,JDt);
Cp=[1 0]; % pulls u out of phugiod model
syslongp=ss(Ap,Bp(:,act_t),Cp,0);
syslongp2=series(syst,syslongp);
[Apt,Bpt,Cpt,Dpt]=ssdata(syslongp2);
%
% model will have a zero at the origin
% so cancel that with a compensator pole
% cancel pole in eng dynamics
% put a zero at 0.1 to pull in the phugoid
%
K_u=.005;
if 1
   Kun=conv([1/.2857 1],[1/.1 1]);Kud=conv([1 0],[1/1 1]);
%   Kun=1;Kud=1;
   [Au,Bu,Cu,Du]=tf2ss(Kun,Kud);
   na=size(Apt,1);nu=size(Au,1);
   Aclt=[Apt Bpt*Cu;zeros(nu,na)  Au];
   Bclt=[[Bpt*Du;Bu]];
   Cclt=[Cpt zeros(size(Cpt,1),nu)];
   Dclt=[zeros(size(Cclt,1),size(Bclt,2))];
   figure(3);clf;axis([-.5 .05,-.05,.6]);sgrid([.5 .707]',[.05]);hold on;
   rlocus(Aclt,Bclt,Cclt,Dclt);
   axis([-.5 .05,-.05,.6])
   r_u=rlocus(Aclt,Bclt,Cclt,Dclt,K_u)'
   hold on;plot(r_u+eps*j,'rd'),hold off
   title('with u FB to delt')
end

% standard so far, except revised to account for changes in the dynamics
% Now close the loop on d

%close loop on q
syscl=feedback(syslong2,diag([K_q]),[act_e],[sen_q]);
%close loop on theta and engine
syst=tf(K_th*Ktn,Ktd);
sysu=tf(K_u*Kun,Kud);
sysc=append(syst,sysu,1);
syslong3=series(sysc,syscl);
syscl2=feedback(syslong3,diag([1 1]),[act_e act_t],[sen_t sen_u]);
[Acl2,Bcl2,Ccl2,Dcl2]=ssdata(syscl2);
%
% pole at origin for tracking perf
% lead at 1 and 2
% zero at 0.05 to catch low pole and pull in the ones at DC
%
if 1
        %Kdn=conv([1/.11 1],[1/.05 1]);Kdd=conv([1/.1 1],[1 0]);   K_d=.0001;
        Kdn=conv([1/1 1],[1/.05 1]);Kdd=conv([1/2 1],[1 0]);   K_d=-.0001;
        [Ad,Bd,Cd,Dd]=tf2ss(Kdn,Kdd);
   na=size(Acl2,1);nd=size(Ad,1);
```

```
    Acl2t=[Acl2 Bcl2(:,act_e)*Cd;zeros(nd,na)  Ad];
    Bcl2t=[[Bcl2(:,act_e)*Dd;Bd] [Bcl2(:,[act_t act_gr]);zeros(nd,2)]];
    Ccl2t=[Ccl2 zeros(size(Ccl2,1),nd)];
    Dcl2t=[zeros(size(Ccl2t,1),size(Bcl2t,2))];
    figure(4);clf;axis([-1 .05,-.05,1]);sgrid([.5 .707]',[.05]);hold on;
    rlocus(Acl2t,sign(K_d)*Bcl2t(:,act_e),Ccl2t(sen_d,:),sign(K_d)*Dcl2t(sen_d,act_e),[0:.000005:.0005])
    %,[0:.0000005:.0001]);
    axis([-4 .05,-.05,2])
    r_d=rlocus(Acl2t,Bcl2t(:,act_e),Ccl2t(sen_d,:),Dcl2t(sen_d,act_e),K_d)'
    hold on;plot(r_d+eps*j,'rd'),hold off
    title('with d FB to dele')
end
Acl3=Acl2t-Bcl2t(:,act_e)*Ccl2t(sen_d,:)*K_d;
Bcl3=[Bcl2t(:,act_e)*K_d Bcl2t(:,[act_t act_gr])];
Ccl3=Ccl2t;
Dcl3=Dcl2t;
%
% add h for simulations
% dot h = U0 (theta - alpha) = U0 theta - w
na=size(Acl3,1);
Acl3t=[Acl3 zeros(na,1);[0 -1 0 U0 zeros(1,na-4+1)]];
syscl3=ss(Acl3t,[Bcl3;[0 0 0]],[[Ccl3 zeros(size(Ccl3,1),1)];...
      [zeros(1,na) 1]],[Dcl3;[0 0 0]]);

t=[0:1:200]';
D0=0;%just above the glide slope;
H0=600;
% U=[d_c U_c Gam_r]
Gam_r=-2.5*pi/180;
INP=[zeros(size(t,1),2) Gam_r*[zeros(10,1);ones(size(t,1)-10,1)]];
% x=[u w q theta d f1 f2 f3 f4 h];
X0=[ zeros(1,na) H0];
[ystep,t]=lsim(syscl3,INP,t,X0);
figure(5)
% y=[u w q th d gamma h]
U=ystep(:,sen_u);W=ystep(:,sen_w);
q=ystep(:,sen_q);TH=ystep(:,sen_t);d=ystep(:,sen_d);
Gam=ystep(:,sen_g);H=ystep(:,sen_g+1);
plot(t,[H0+d H],[t(11);max(t)],[H0;H0+U0*(max(t)-t(11))*tan(Gam_r)]);

figure(6)
plot(t,[Gam]*180/pi)


return
f=logspace(-3,2,400);
g=freqresp(Acl3,Bcl3(:,3),Ccl3,Dcl3,1,2*pi*f*j);
loglog(f,abs(g(:,sen_d)))

% phugoid approx from reid
a1=U0*(-Mw*Xdp/m);
a0=0;
AA=-U0*Mw;
BB=g*Mu+U0/m*(Xu*Mw-Mu*Xw);
CC=g/m*(Zu*Mw-Mu*Zw);
rlocus(conv([a1 a0],JNt),conv([AA BB CC],JDt))


return
t=Lan(:,1);u=Lan(:,2);
w=Lan(:,3);q=Lan(:,4);
th=Lan(:,5);d=Lan(:,6);
gam=Lan(:,7);ele=Lan(:,8);
thr=Lan(:,9);H=Lan(:,10);
Hr=Lan(:,11);

figure(6)
plot(t,[Hr H]);setlines(2)
xlabel('time');
legend('H_{ref}','H')
print -depsc land3_1.eps
jpdf('land3_1')
```

```
figure(1);plot(t,ele*180/pi,t,10*thr);setlines;xlabel('time');
ylabel('Control Signals')
title(['tau_f= ',num2str(tau_f)]);legend('\delta_e degs','10*\delta_t')
setlines(2)
print -depsc land3_2.eps
jpdf('land3_2')

return
if 1
   Kdn=conv([1/.11 1],[1/.05 1]);Kdd=conv([1/.1 1],[1 0]);[Ad,Bd,Cd,Dd]=tf2ss(Kdn,Kdd);
   K_d=.00005;
   na=size(Acl2,1);nd=size(Ad,1);
   Acl2t=[Acl2 Bcl2(:,act_e)*Cd;zeros(nd,na)  Ad];
   Bcl2t=[[Bcl2(:,act_e)*Dd;Bd] [Bcl2(:,[act_t act_gr]);zeros(nd,2)]];
   Ccl2t=[Ccl2 zeros(size(Ccl2,1),nd)];
   Dcl2t=[zeros(size(Ccl2t,1),size(Bcl2t,2))];

   figure(4);clf;axis([-1 .05,-.05,1]);sgrid([.5 .707]',[.05]);hold on;
   rlocus(Acl2t,Bcl2t(:,act_e),Ccl2t(sen_d,:),Dcl2t(sen_d,act_e),[0:.000005:.0005])
   %,[0:.0000005:.0001]);
   axis([-1 .05,-.05,1])
   r_d=rlocus(Acl2t,Bcl2t(:,act_e),Ccl2t(sen_d,:),Dcl2t(sen_d,act_e),K_d)'
   hold on;plot(r_d+eps*j,'rd'),hold off
   title('with d FB to dele')
end
```