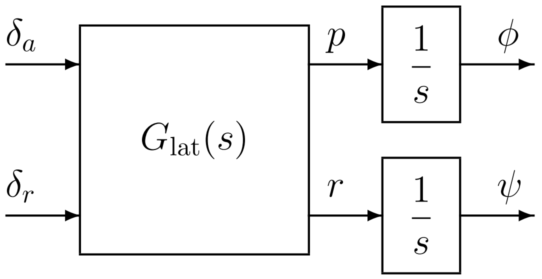
Lecture # 12

Aircraft Lateral Autopilots

- Multi-loop closure
- Heading Control: linear
- Heading Control: nonlinear

Lateral Autopilots

- We can stabilize/modify the lateral dynamics using a variety of different feedback architectures.



- Look for good sensor/actuator pairings to achieve desired behavior.
- **Example:** Yaw damper
  - Can improve the damping on the Dutch-roll mode by adding a feedback on r to the rudder: $\delta_r^c = k_r(r_c - r)$
  - Servo dynamics $H_r = \frac{3.33}{s+3.33}$ added to rudder $\delta_r^a = H_r \delta_r^c$
  - System:

$$G_{\delta_r^c r} = -\frac{1.618s^3 + 0.7761s^2 + 0.03007s + 0.1883}{s^5 + 3.967s^4 + 3.06s^3 + 3.642s^2 + 1.71s + 0.01223}$$

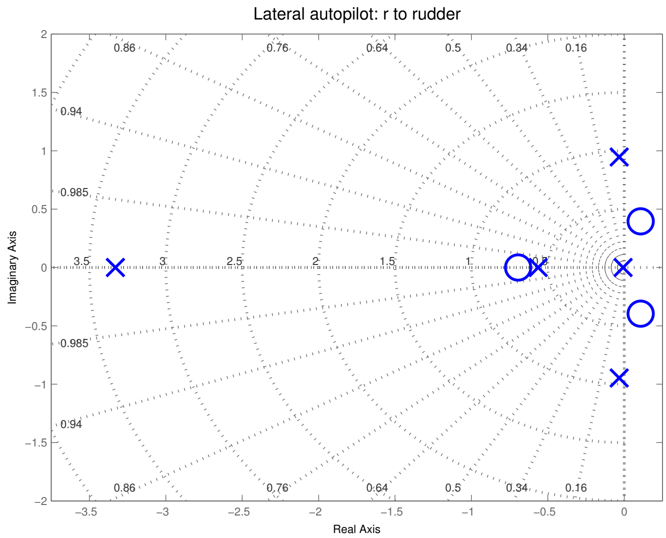


Figure 2: Lateral pole-zero map $G_{\delta_r^c r}$

- Note that the gain of the plant is negative ($K_{plant} < 0$), so if $k_r < 0$, then $K = K_{plant} k_r > 0$, so must draw a 180° locus (neg feedback)

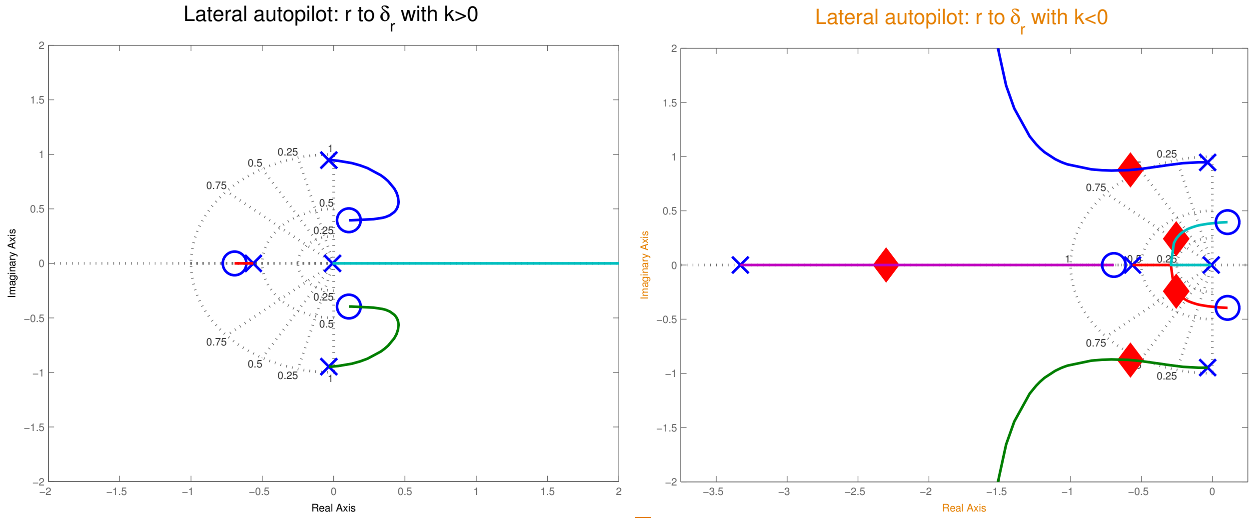


Figure 3: Lateral pole-zero map. Definitely need $k_r < 0$

- Root locus with $k_r < 0$ looks pretty good as we have authority over the four poles.
  - $k_r = -1.6$ results in a large increase in the Dutch-roll damping and spiral/roll modes have combined into a damped oscillation.

- Yaw damper looks great, but this implementation has a **problem**.
  - There are various flight modes that require a steady yaw rate ($r_{ss} \neq 0$). For example, steady turning flight.
  - Our current yaw damper would not allow this to happen – it would create the rudder inputs necessary to cancel out the motion **!!**
  - **Exact opposite of what we want to have happen**, which is to damp out any oscillations about the steady turn.

Yaw Damper: Part 2

- Can avoid this problem to some extent by filtering the feedback signal.
  - Feedback only a high pass version of the r signal.
  - High pass cuts out the low frequency content in the signal $\Rightarrow$ steady state value of r would not be fed back to the controller.
- New yaw damper: $\delta_r^c = k_r(r_c - H_w(s)r)$ where $H_w(s) = \frac{\tau s}{\tau s+1}$ is the "washout" filter.

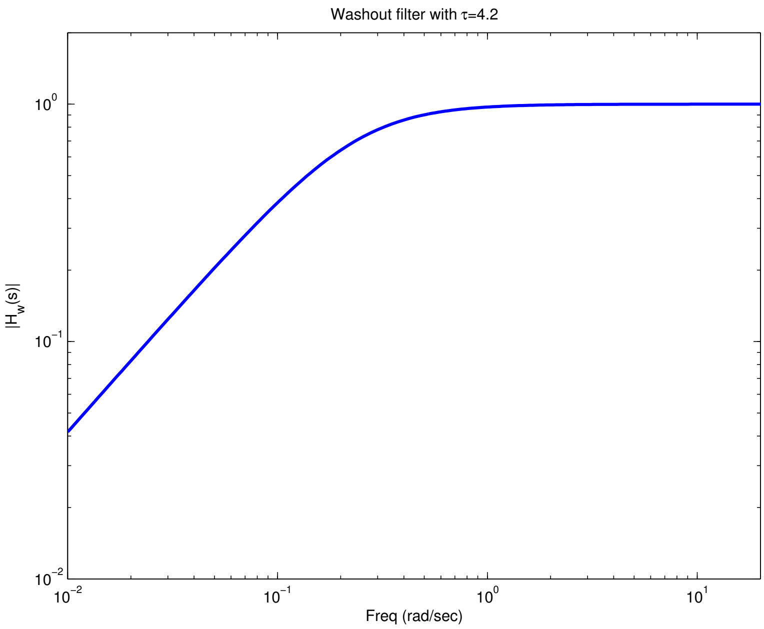
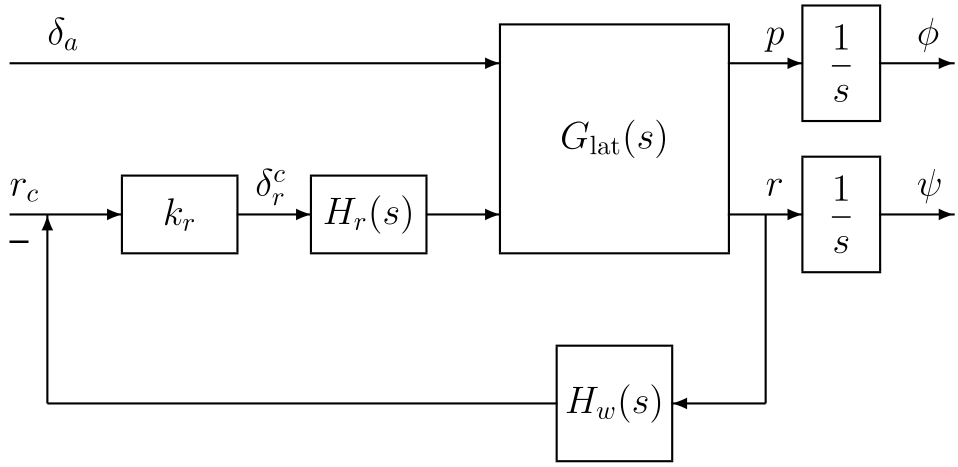


Figure 4: Washout filter with $\tau = 4.2$

- New control picture



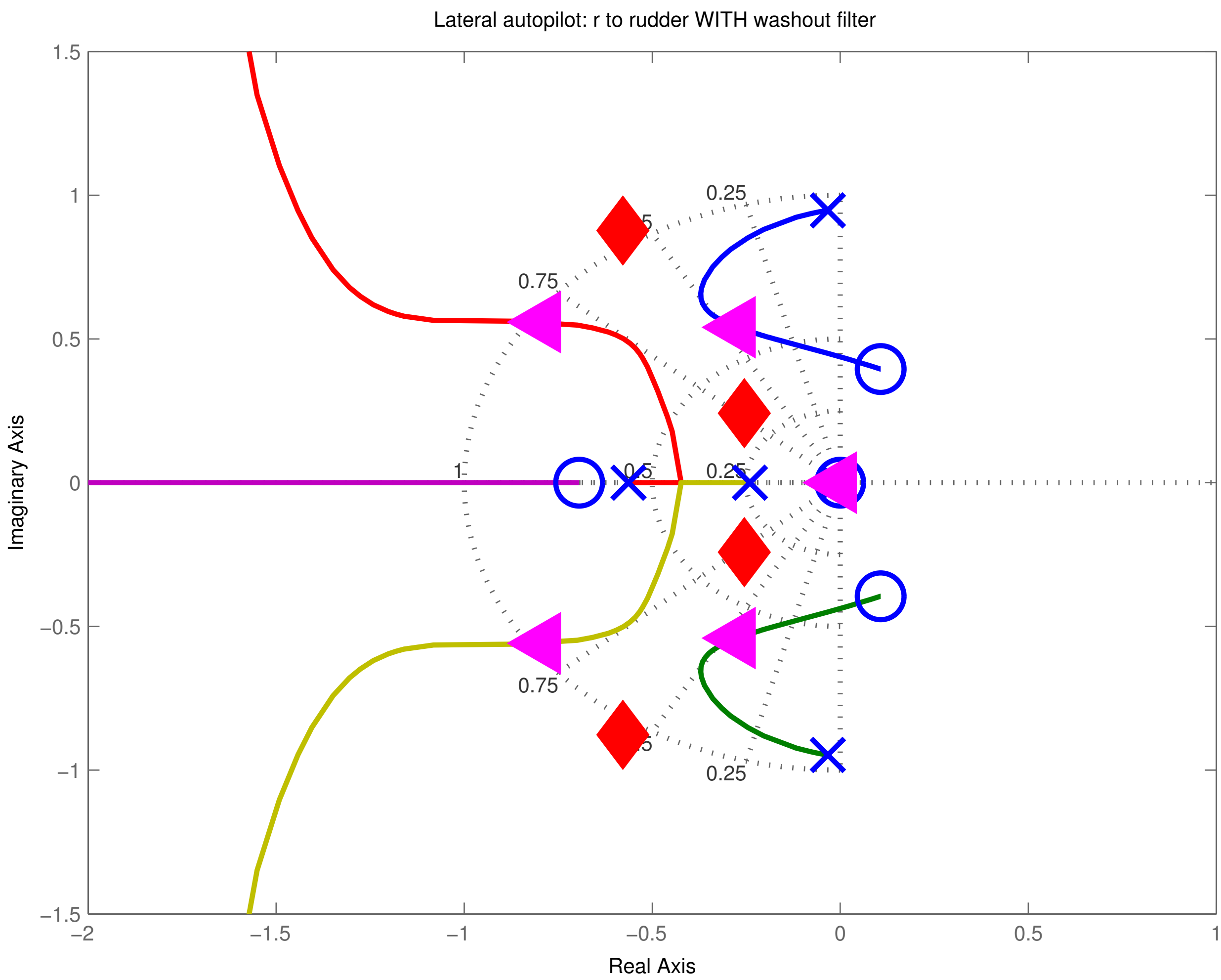


Figure 5: Root Locus with the washout filter included.

- Zero in $H_w(s)$ traps a pole near the origin, but it is slow enough that it can be controlled by the pilot.
- Obviously has changed the closed loop pole locations (♦ ⇒ ◀), but $k_r = -1.6$ still seems to give a well damped response.

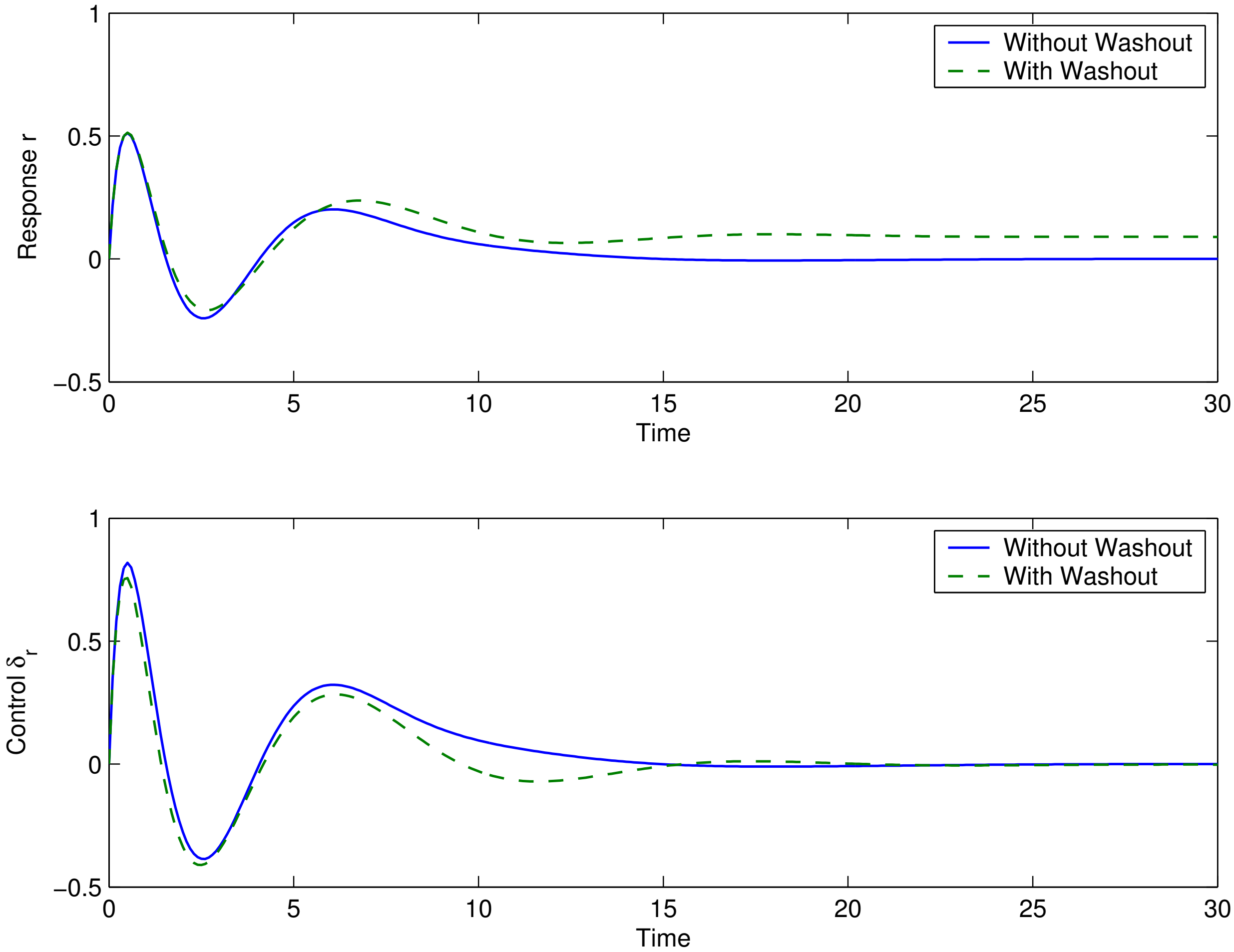
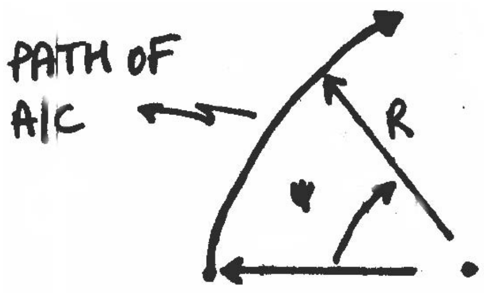


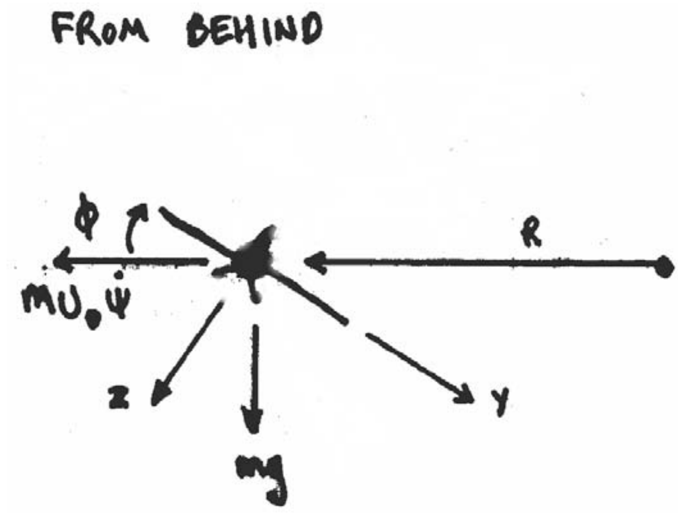
Figure 6: Impulse response of closed loop system with and without the Washout filter ($\tau = 4.2$). Commanded $r_c = 0$, and both have $(\delta_r)_{ss} = 0$, but without the filter, $r_{ss} = 0$, whereas with it, $r_{ss} \neq 0$.

- For direct comparison with and without the filter, applied impulse as r_c to both closed-loop systems and then calculated r and δ_r.
- Bottom plot shows that control signal quickly converges to zero in both cases, i.e., no more control effort is being applied to correct the motion.
- But only the one with the washout filter produces a zero control input even though the there is a steady turn $\Rightarrow$ the controller will not try to fight a commanded steady turn.

Heading Autopilot Design

- So now have the yaw damper added correctly and want to control the heading ψ.
  - Need to bank the aircraft to accomplish this.
  - Result is a "coordinated turn" with angular rate $\dot{\psi}$





- Aircraft banked to angle ϕ so that vector sum of mg and $mU_0\dot{\psi}$ is along the body z-axis
  - Summing in the body y-axis direction gives $mu_0\dot{\psi}\cos\phi = mg\sin\phi$

$$\tan\phi = \frac{U_0\dot{\psi}}{g}$$

- Since typically $\phi \ll 1$, we have

$$\phi \approx \frac{U_0\dot{\psi}}{g}$$

gives the desired bank angle for a specified turn rate.

- Problem now is that $\dot{\psi}$ tends to be a noisy signal to base out bank angle on, so we generate a smoother signal by filtering it.
  - Assume that the desired heading is known ψ_d and we want ψ to follow ψ_d relatively slowly
  - Choose dynamics $\tau_1\dot{\psi} + \psi = \psi_d$

$$\Rightarrow \frac{\psi}{\psi_d} = \frac{1}{\tau_1 s + 1}$$

    with τ_1=15-20sec depending on the vehicle and the goals.
  - A low pass filter that eliminates the higher frequency noise.

- Filtered heading angle satisfies

$$\dot{\psi} = \frac{1}{\tau_1}(\psi_d - \psi)$$

  which we can use to create the desired bank angle:

$$\phi_d = \frac{U_0}{g}\dot{\psi} = \frac{U_0}{\tau_1 g}(\psi_d - \psi)$$

Roll Control

- Given this desired bank angle, we need a **roll controller** to ensure that the vehicle tracks it accurately.
  – Aileron is best actuator to use: $\delta_a = k_\phi(\phi_d - \phi) - k_p p$
- To design k_ϕ and k_p, can just use the approximation of the roll mode

$$\left.\begin{array}{l} I'_{xx}\dot{p} = L_p p + L_{\delta_a}\delta_a \\ \dot{\phi} = p \end{array}\right\} I'_{xx}\ddot{\phi} - L_p\dot{\phi} = L_{\delta_a}\delta_a$$

which gives

$$\frac{\phi}{\delta_a} = \frac{L_{\delta_a}}{s(I'_{xx}s - L_p)}$$

- For the design, add the aileron servo dynamics

$$H_a(s) = \frac{1}{0.15s + 1}, \quad \delta_a^a = H_a(s)\delta_a^c$$

- PD controller $\delta_a^c = -k_\phi(s\gamma + 1) + k_\phi\phi_d$, adds zero at $s = -1/\gamma$
  – Pick $\gamma = 2/3$

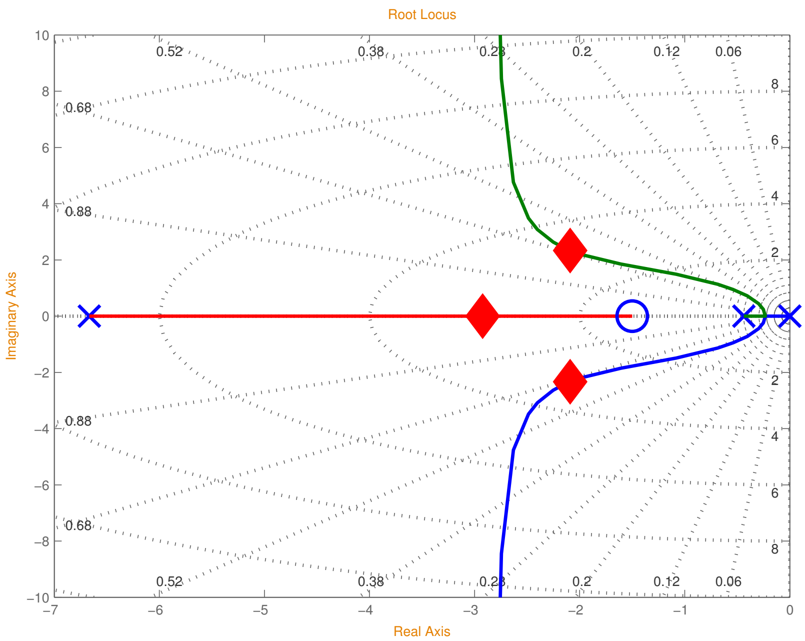


Figure 7: Root Locus for roll loop – closed Loop poles for $K_p = -20$, $K_\phi = -30$

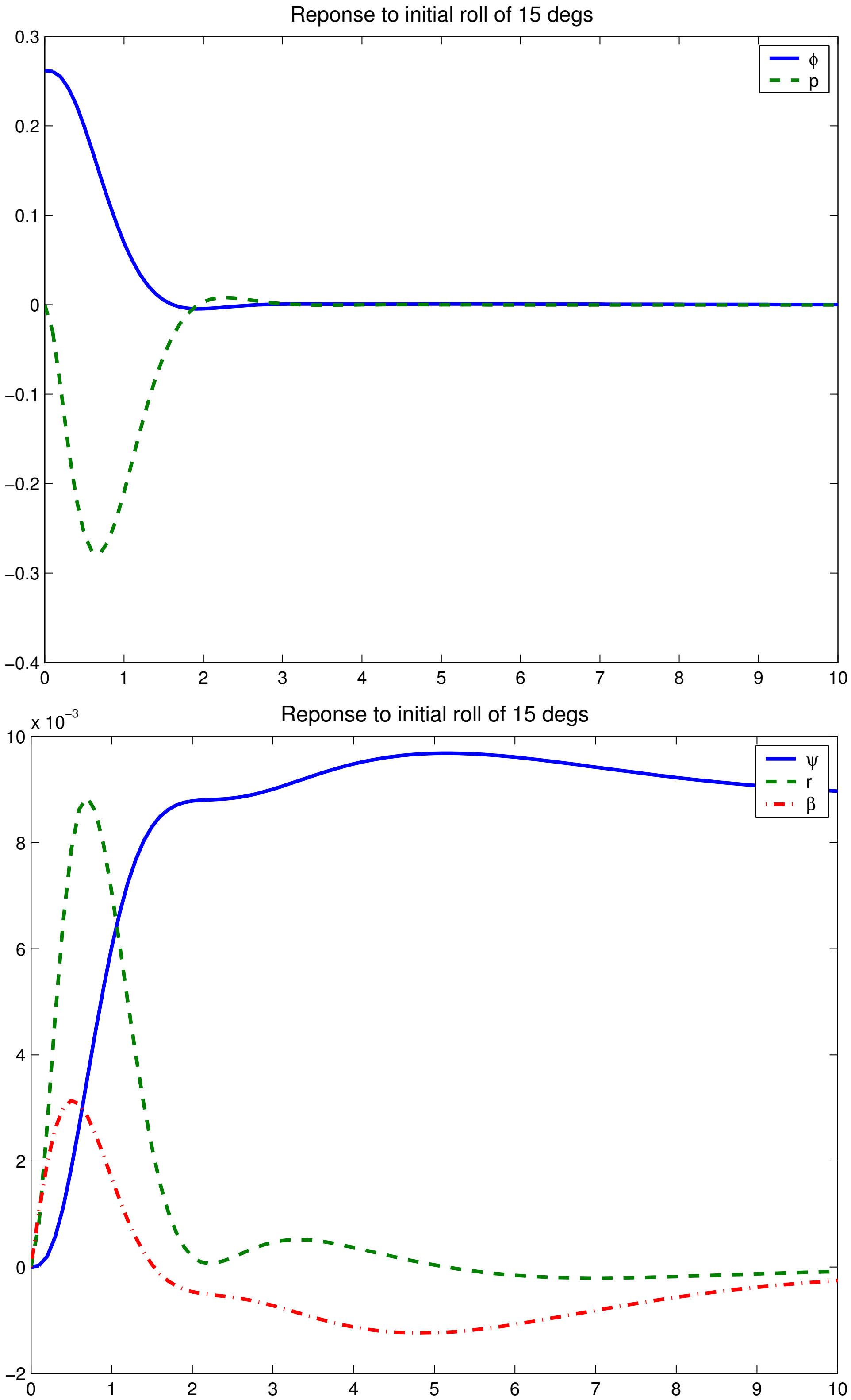


Figure 8: Roll response to an initial roll offset

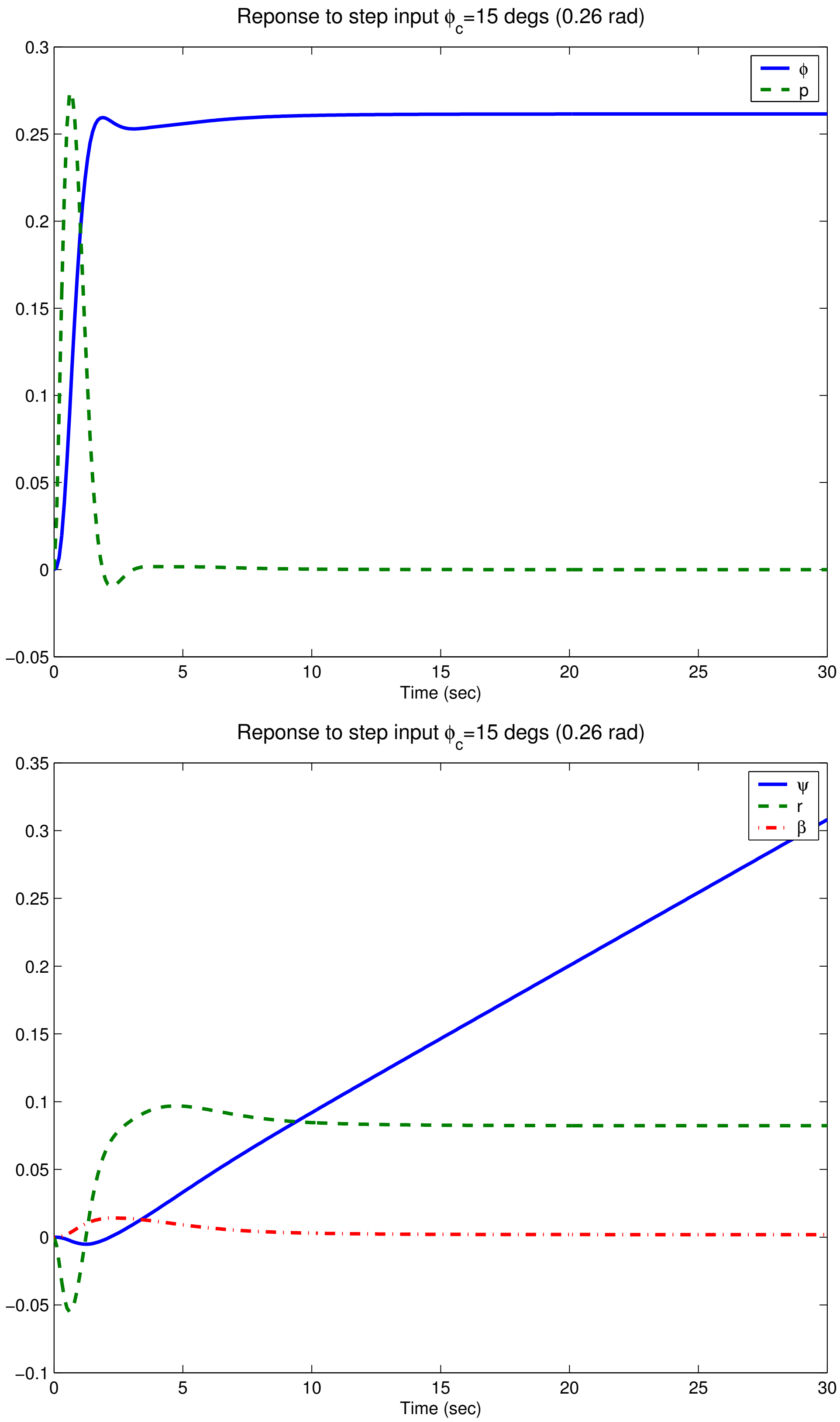
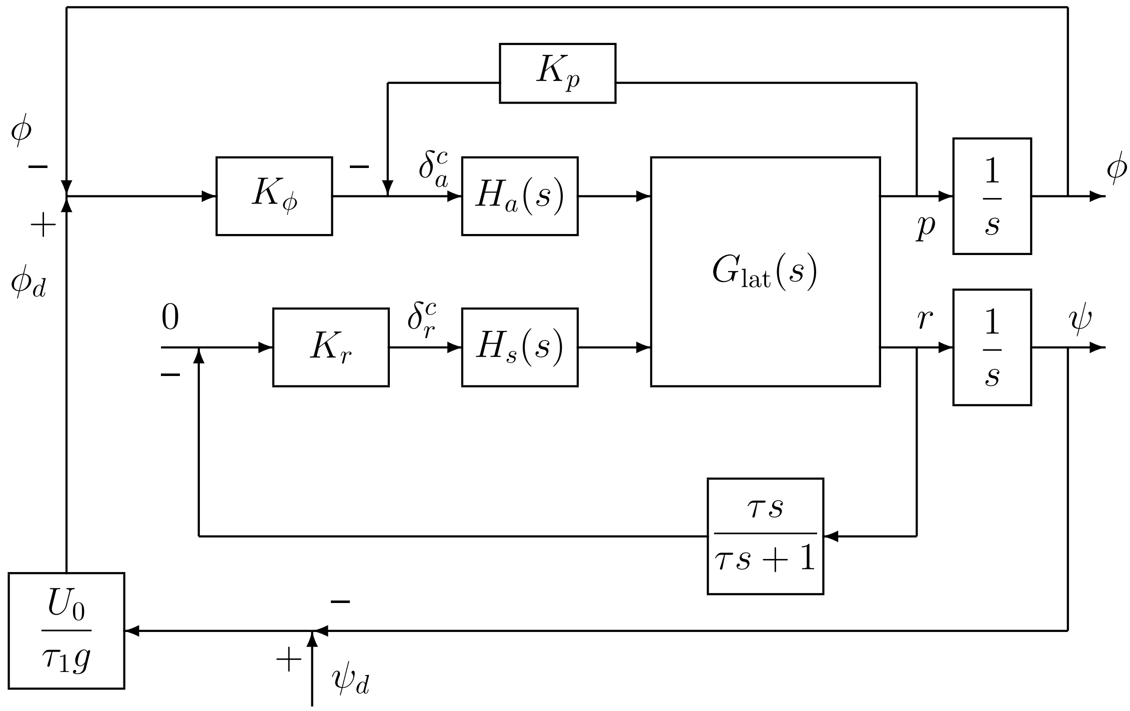


Figure 9: Roll response to a step command for ϕ_c. Note the adverse yaw effects.

Heading Autopilot

- Putting the pieces together we get the following autopilot controller



- Last step: analyze effect of closing the $\psi \rightarrow \phi_d$ loop

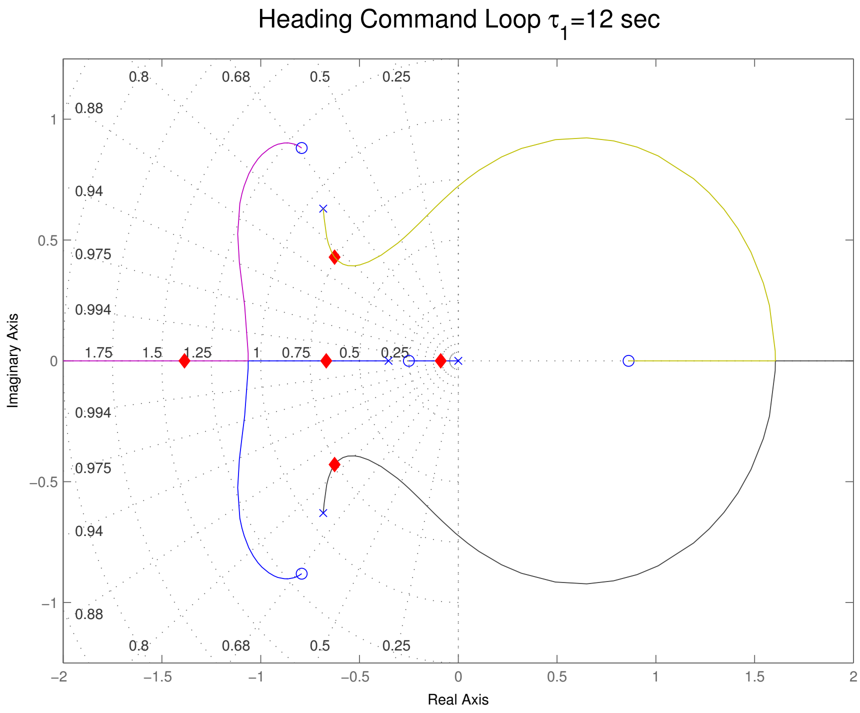


Figure 10: Heading loop root locus. Closed loop poles for gain $U_0/(g\tau_1)$. 8th order system, but RL fairly well behaved.

- Measure ϕ with a vertical gyro and ψ with a directional gyro
- Add a limiter to ϕ_d or else we can get some very large bank angles

- Design variables are K_ϕ, K_p, τ_1, τ, and K_r
- Multi-loop closure must be done carefully
  - Must choose the loop gains carefully so that each one is slower than the one "inside"

    $\Rightarrow$ can lead to slow overall response

  - Analysis on fully coupled system might show that the controllers designed on subsystem models interact with other modes (poles)

    $\Rightarrow$ several iterations might be required

- Now need a way to specify ψ_d

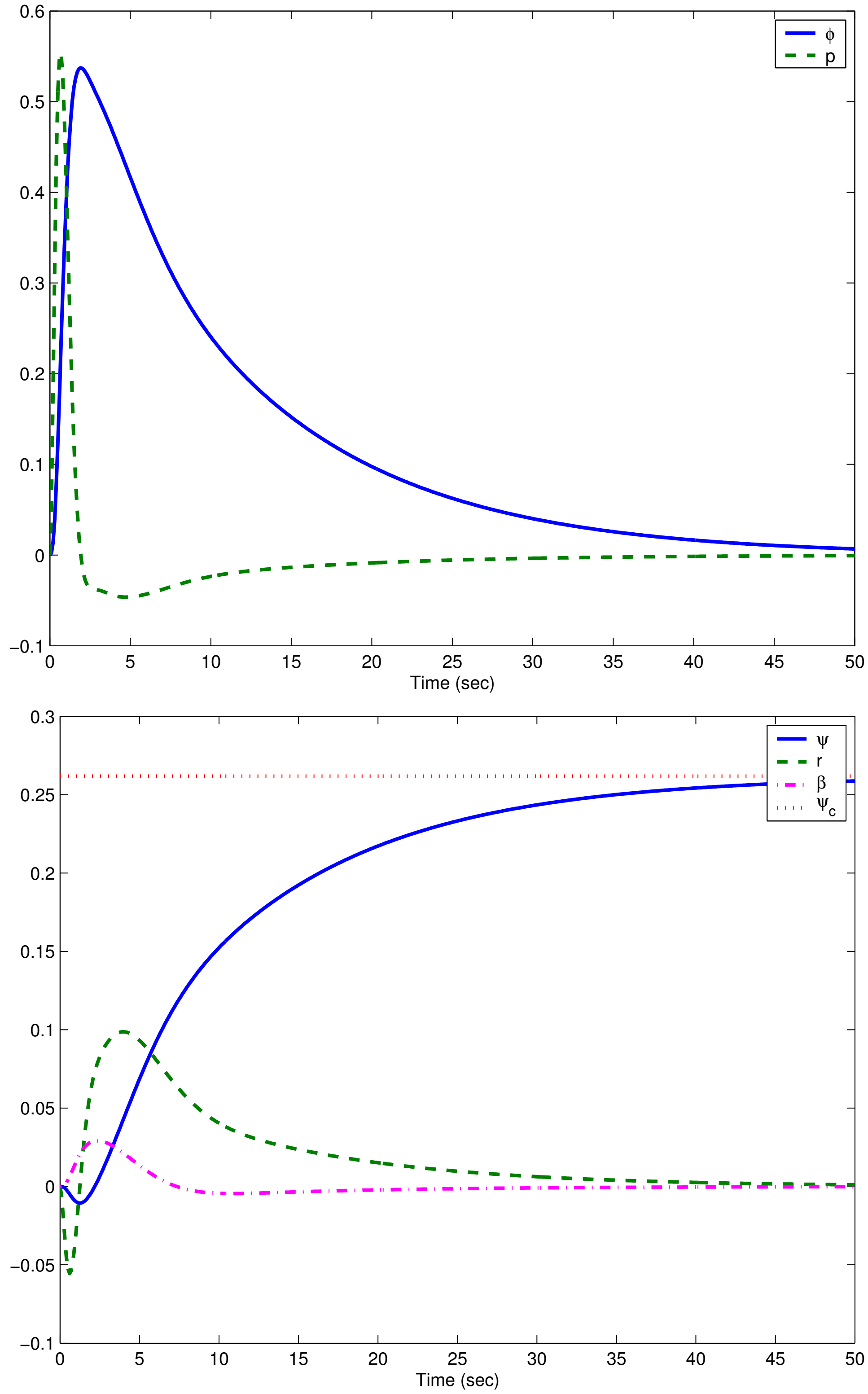


Figure 11: Response to 15 deg step in ψ_c. Note the bank angle is approximately 30 degs, which is about the maximum allowed. Decreasing τ_1 tends to increase $\phi_{\max}$.

Ground Track Control

- Consider scenario shown - use this to determine desired heading ψ_d

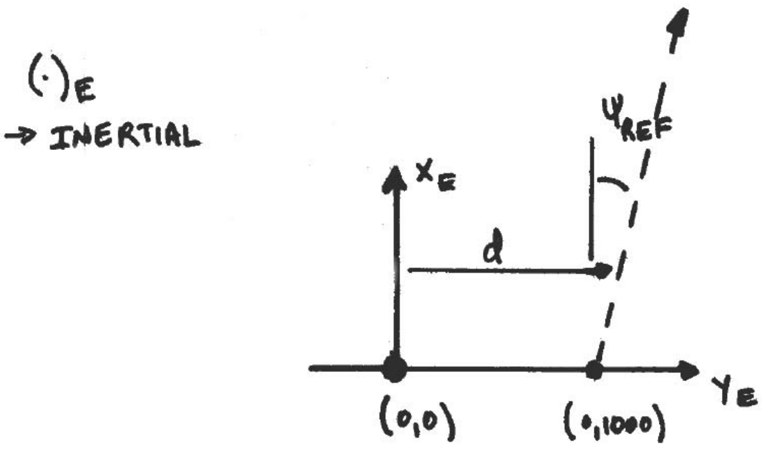


Figure 12: Heading definitions

  – Are initially at (0,0), moving along X_e ($\psi_0 = 0$)

  – Want to be at (0,1000) moving along dashed line angled at ψ_{ref}

- Separation distance $d = Y_{ref} - Y_{a/c}$ ($d_0 = 1000$)

  – Desired inertial y position - y position of aircraft

$$\dot{d} = U_0 \sin(\psi_{ref} - \psi) \approx U_0(\psi_{ref} - \psi)$$

- Want to smoothly decrease d to zero, use a filter so that

$$\begin{aligned} \dot{d} &= -\frac{1}{\tau_d} d \qquad \tau_d = 30 \text{ sec} \\ \Rightarrow -\frac{1}{\tau_d} d &= U_0(\psi_{ref} - \psi) \\ \Longrightarrow \psi &= \psi_{ref} + \frac{1}{\tau_d U_0} d = \psi_{ref} + \frac{1}{\tau_d U_0}(Y_{ref} - Y_{a/c}) \end{aligned}$$

which includes a feedback on the aircraft inertial Y position.
$\Rightarrow$ Use this as the input ψ_d.

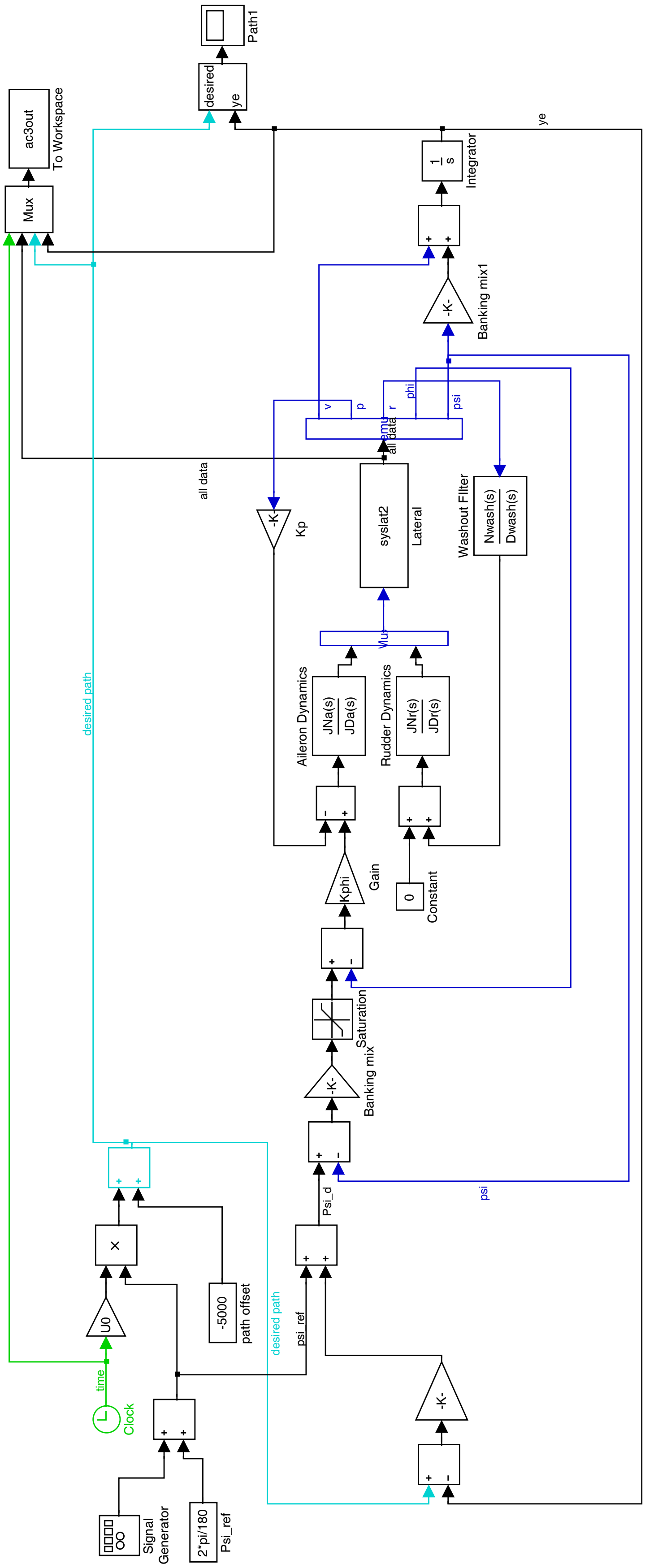


Figure 13: Simulink implementation – run ac3.m first

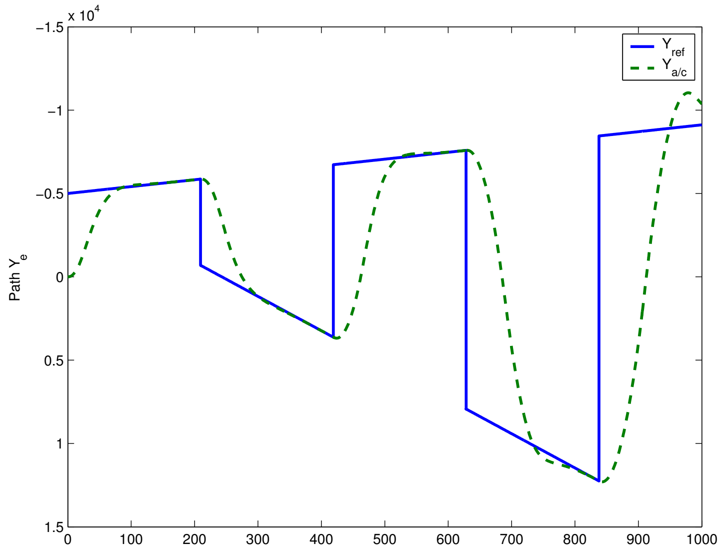


Figure 14: Path following for ground track controller

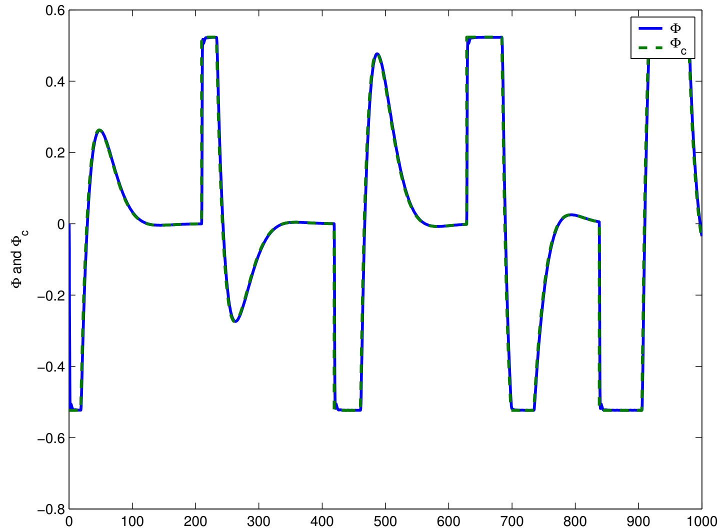


Figure 15: Roll command following for ground track controller

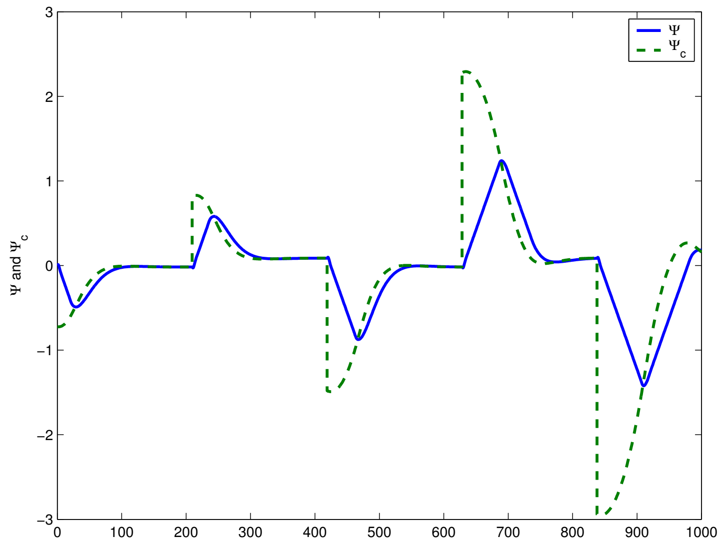


Figure 16: Heading following for ground track controller

Alternative Strategy

- Sanghyuk Park developed an alternative tracking algorithm that he presented this past summer[1]
- Guidance logic selects a reference point on the desired trajectory and uses it to generate a lateral acceleration command.
  - *Selection of Reference Point* – Reference point is on the desired path at a distance (L_1) forward of the vehicle – Figure 17.
  - *Lateral Acceleration Command* – determined by $a_{s_{cmd}} = 2\frac{V^2}{L_1}\sin\eta$

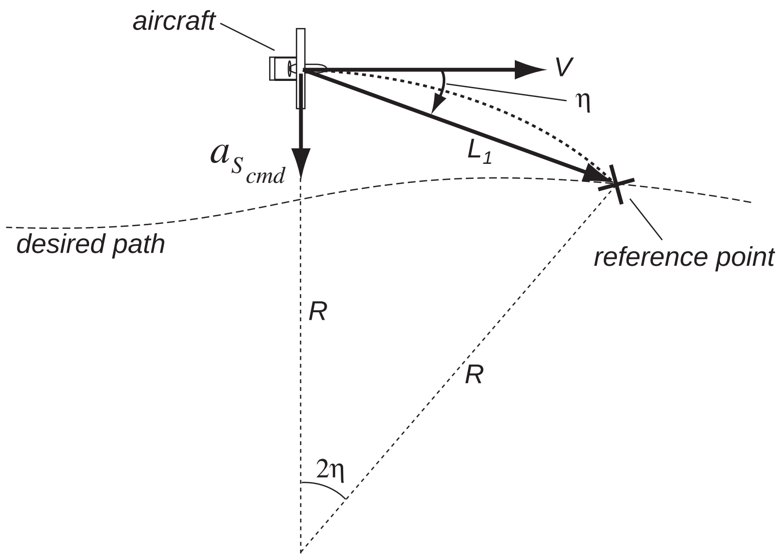


Figure 17: Diagram for Guidance Logic

- Direction of acceleration depends on sign of angle between the L_1 line segment and the vehicle velocity vector.
  - If selected reference point to the right of the vehicle velocity vector, then the vehicle will be commanded to accelerate to the right (see Figure 17) ⇒ vehicle will tend to align its velocity direction with the direction of the L_1 line segment.

[1] Sanghyuk Park, John Deyst, and Jonathan How, "A New Nonlinear Guidance Logic for Trajectory Tracking," AIAA GNC 2004.

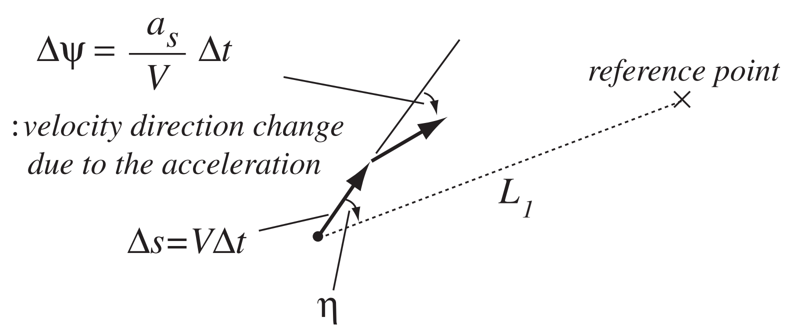


Figure 18: One Time Step

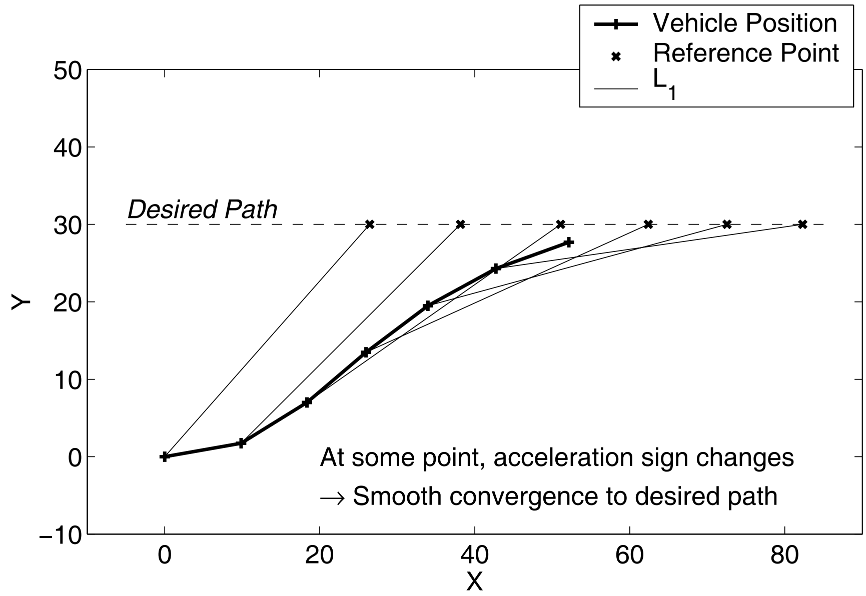


Figure 19: Step by Step (Δt=1, V=10, and L_1=40)

- Figure 18 shows evolution of the guidance logic in one time step and Figure 19 shows the trajectory of the vehicle over several time steps.
  - Vehicle initially starts from a location far away from the desired path, and eventually converges to it.
  - If vehicle far from the desired path, then the guidance logic rotates the vehicle so that its velocity direction approaches the desired path at a large angle.
  - If vehicle close to the desired path, then the guidance logic rotates the vehicle so its velocity direction approaches the desired path at a small angle.

- Figure 20 defines the notation used for a linearization.

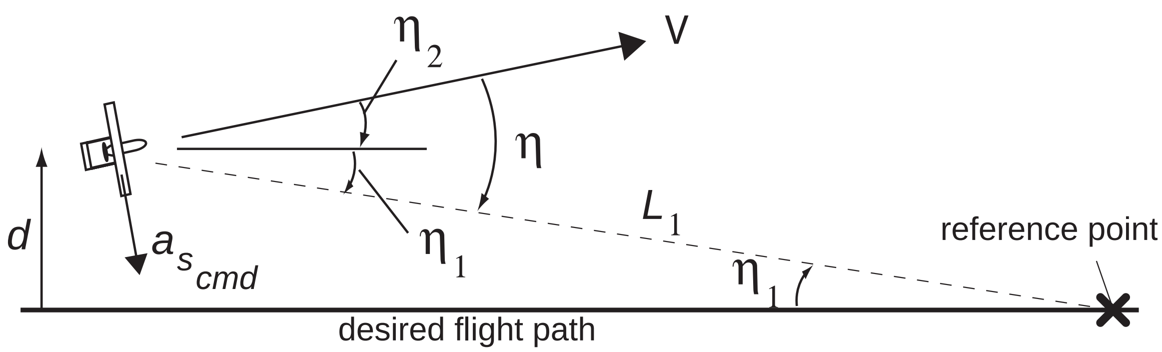


Figure 20: Linear Model for Straight-line Following Case

  – L_1 is the distance from the vehicle to the reference point.
  – d is the cross-track error
  – V is the vehicle nominal speed.

- Assuming η is small $\sin\eta \approx \eta = \eta_1 + \eta_2$ and

$$\eta_1 \approx \frac{d}{L_1}, \quad \eta_2 \approx \frac{\dot{d}}{V}$$

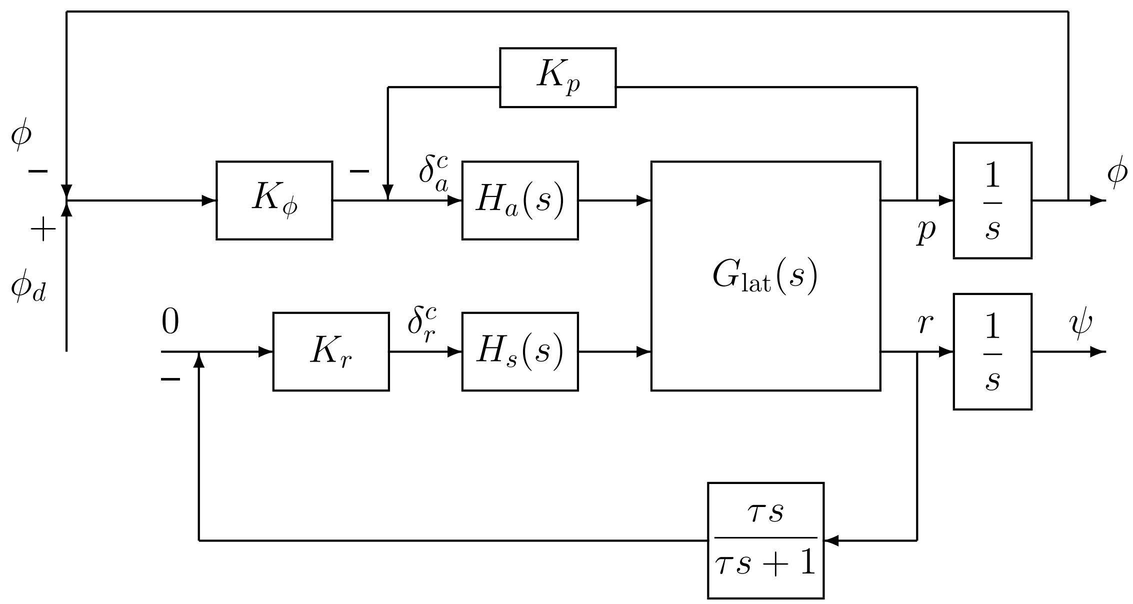
- Combining the above gives

$$a_{s_{cmd}} = 2\frac{V^2}{L_1}\sin\eta \approx 2\frac{V}{L_1}\left(\dot{d} + \frac{V}{L_1}d\right) \tag{1}$$

  – Linearization yields a PD controller for the cross-track error.
  – Ratio of V and separation distance L_1 is an important factor in determining the proportional and derivative controller gains.
  – **Key points:** NL form works significantly better than a PD and is much more tolerant to winds disturbances.

Implementation

- Can implement this command by assuming that vehicle maintains sufficient lift to balance weight, even though banked at angle ϕ.
  - Requires that the vehicle speed up and/or change to a larger α. Lift increment
  $$\Delta C_L = \frac{L - mg}{QS} \equiv (n-1)\frac{W}{QS}$$
  where $n \equiv L/W$ is the *load factor*.
  - Assume that $L\cos\phi = W$, then $L\sin\phi = ma_s$, so that
  $$\tan\phi = \frac{a_s}{g}$$
- We can use this to develop ϕ_d that we apply to the roll controller.



- Some simulations shown
  - Works well – hardest part is determining where to place the reference point, which is L_1 ahead along the path.
  - Recall that L_1 acts like a gain in the controller – making it too small can drive the aircraft unstable.

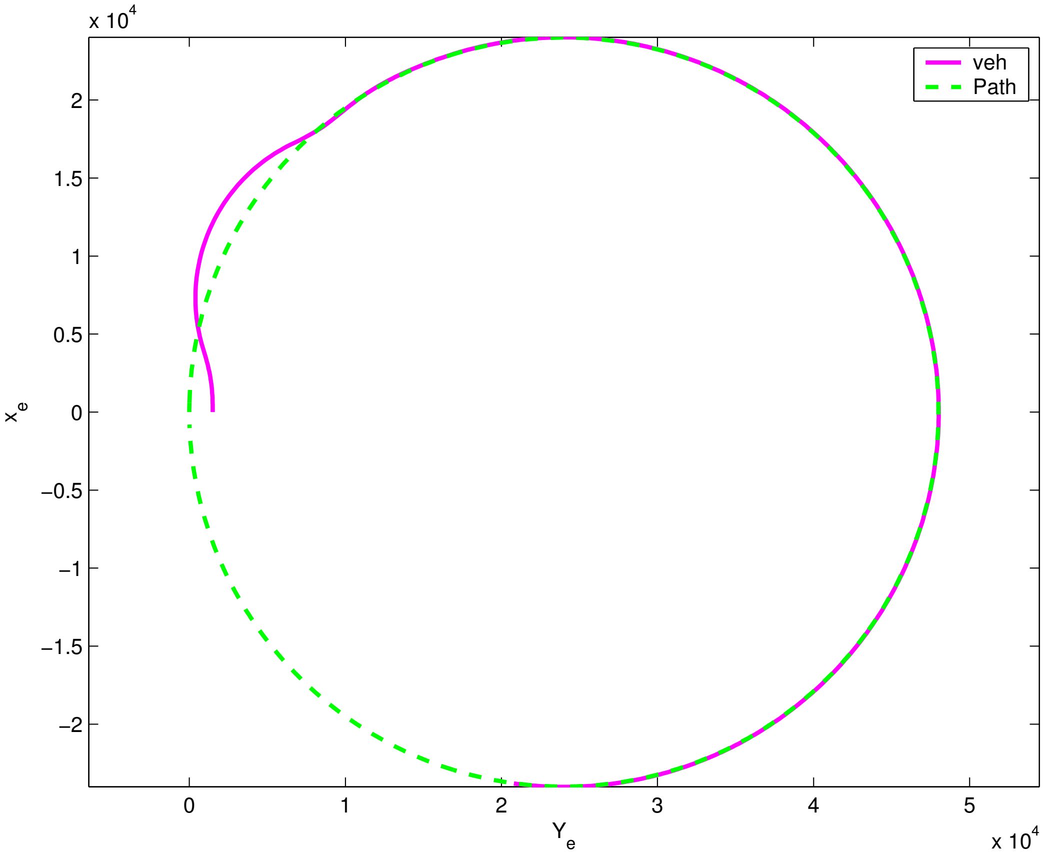


Figure 21: Simulation #1: path. Turn radius $R \approx V^2/(g \tan \phi)$

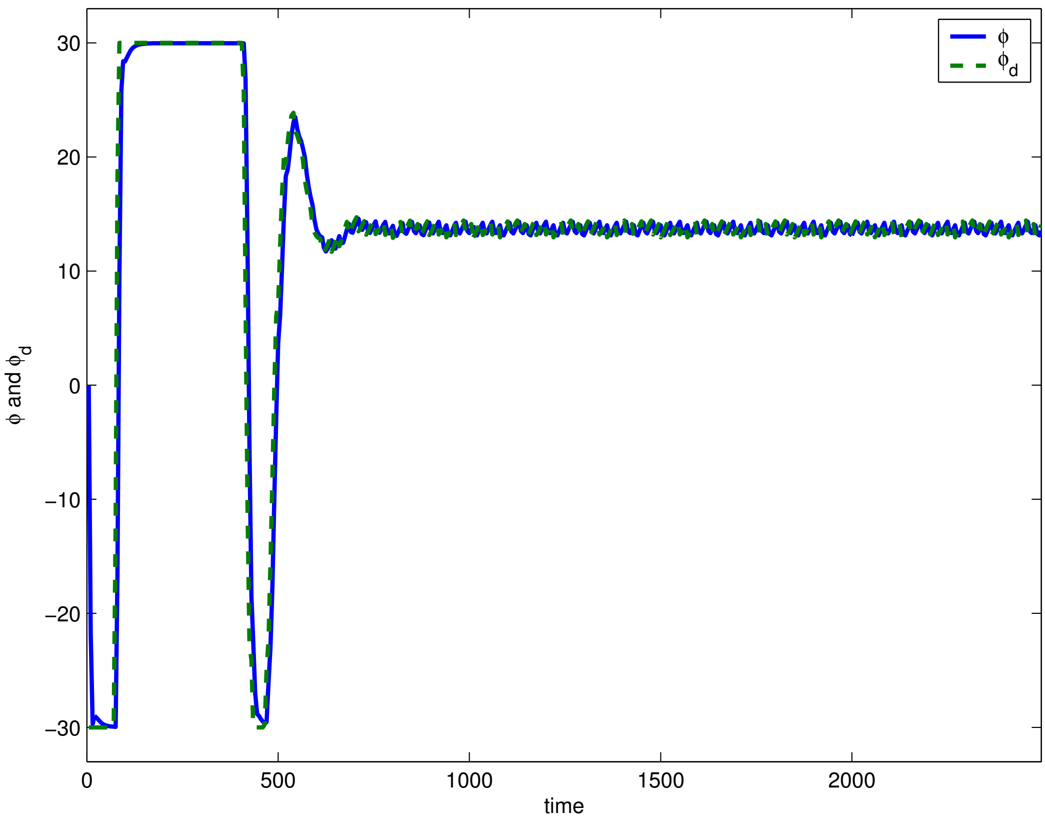


Figure 22: simulation #1: bank angle. Limited to 30 degs here.

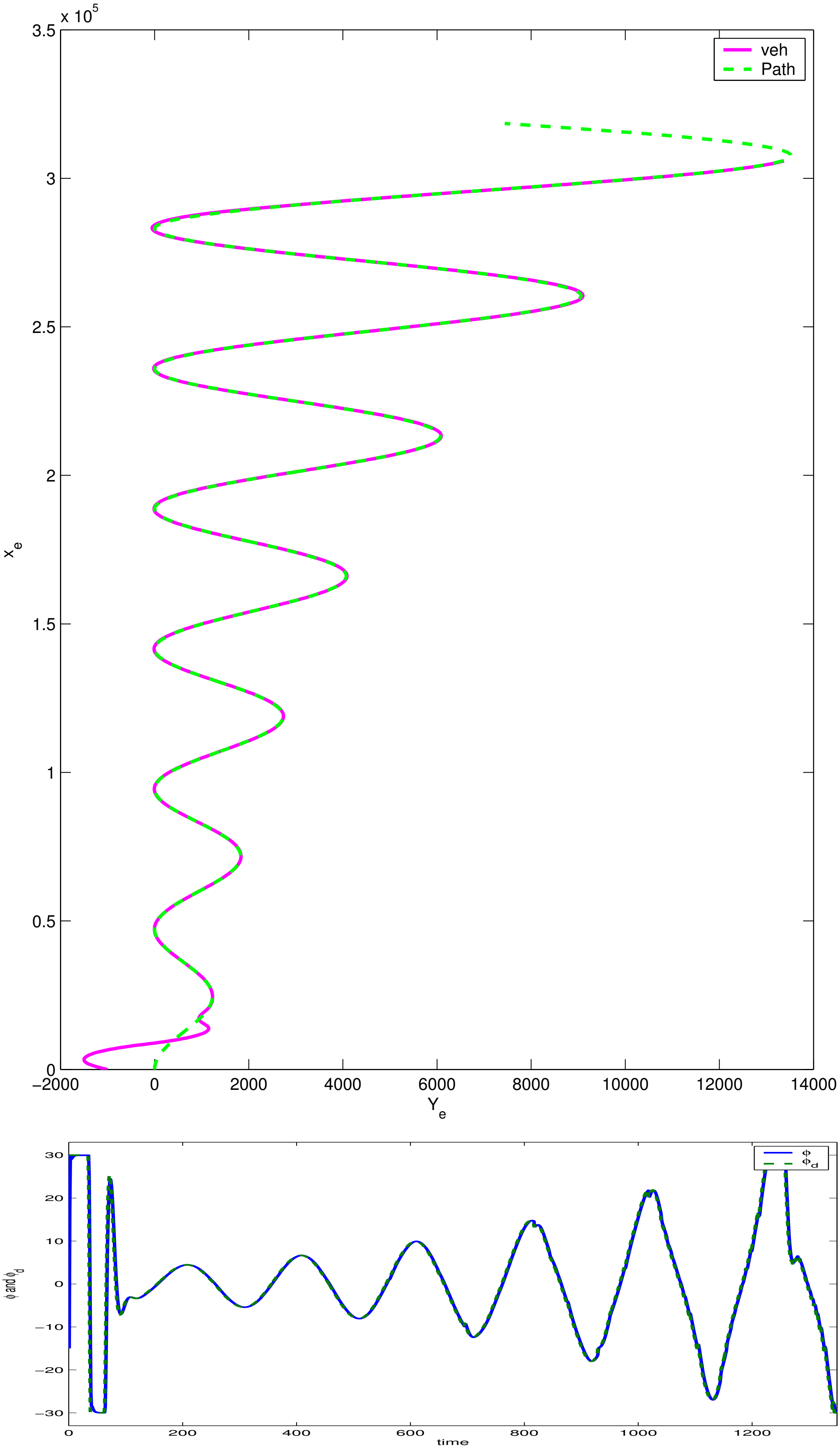


Figure 23: simulation #2

Flight Test

- Guidance algorithm implemented and tested with two UAVs [Parent Child Unmanned Air Vehicle (PCUAV) project by Prof. Deyst]
  - Required lateral acceleration achieved using bank angle control
  - Nominal flight velocity of about 25 m/s, the choice of L_1=150 m results in the associated crossover frequency at 0.4 rad/s.

- Figure 24 shows the flight data for the Mini vehicle using the guidance logic in the lateral dynamics.
  - Plot shows the 2-dimensional trajectory of the Mini vehicle (–) with a commanded desired trajectory (- -).
  - Small numbers along the trajectory are the flight times recorded in the onboard avionics.
  - Lateral displacement between the vehicle and the desired path within ± 2 meters for the 75% of its flight time and within ± 3 meters for 96% of the flight time

.

- A similar flight test was performed for the OHS Parent (see Figure 25)
  - After the transient period, the trajectory of the vehicle followed the commanded path within ± 2 meters for the 78% of its flight time and within ± 3 meters for 97% of the flight time.

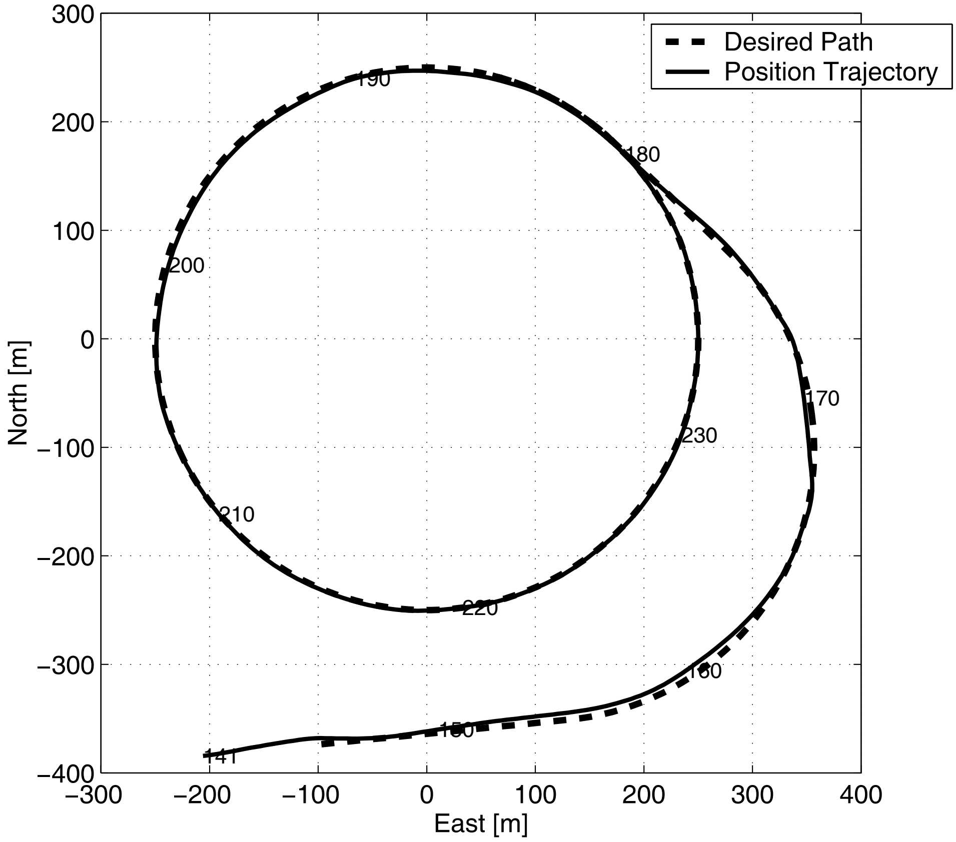


Figure 24: Flight Data of MINI - Trajectory Following

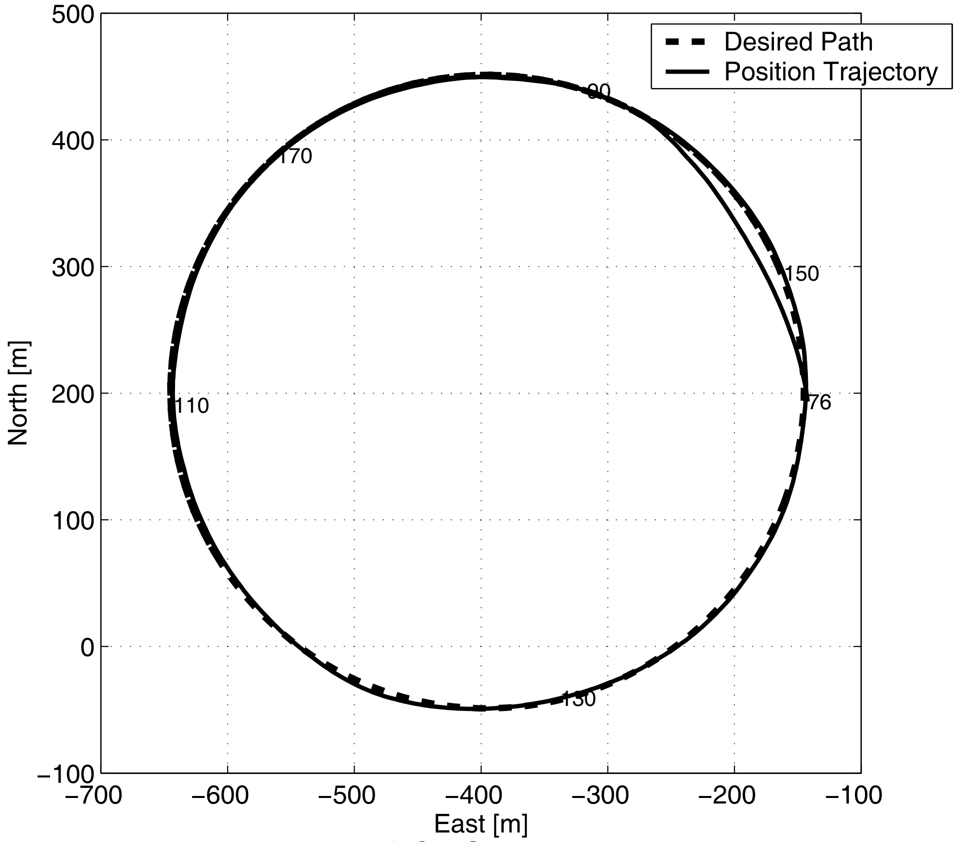


Figure 25: Flight Data of OHS Parent - Trajectory Following

- Then demonstrated rendezvous from any arbitrary initial positions to a configuration of tight formation flight.
  - Figures 26 show the positions of the Parent and the Mini in the north-east 2-D map every 10 seconds.
  - OHS Parent vehicle follows the circular flight path, with no knowledge of the Mini vehicle's location.
  - Mini vehicle schedules its flight path and performs formation flight by receiving position information from the OHS Parent.

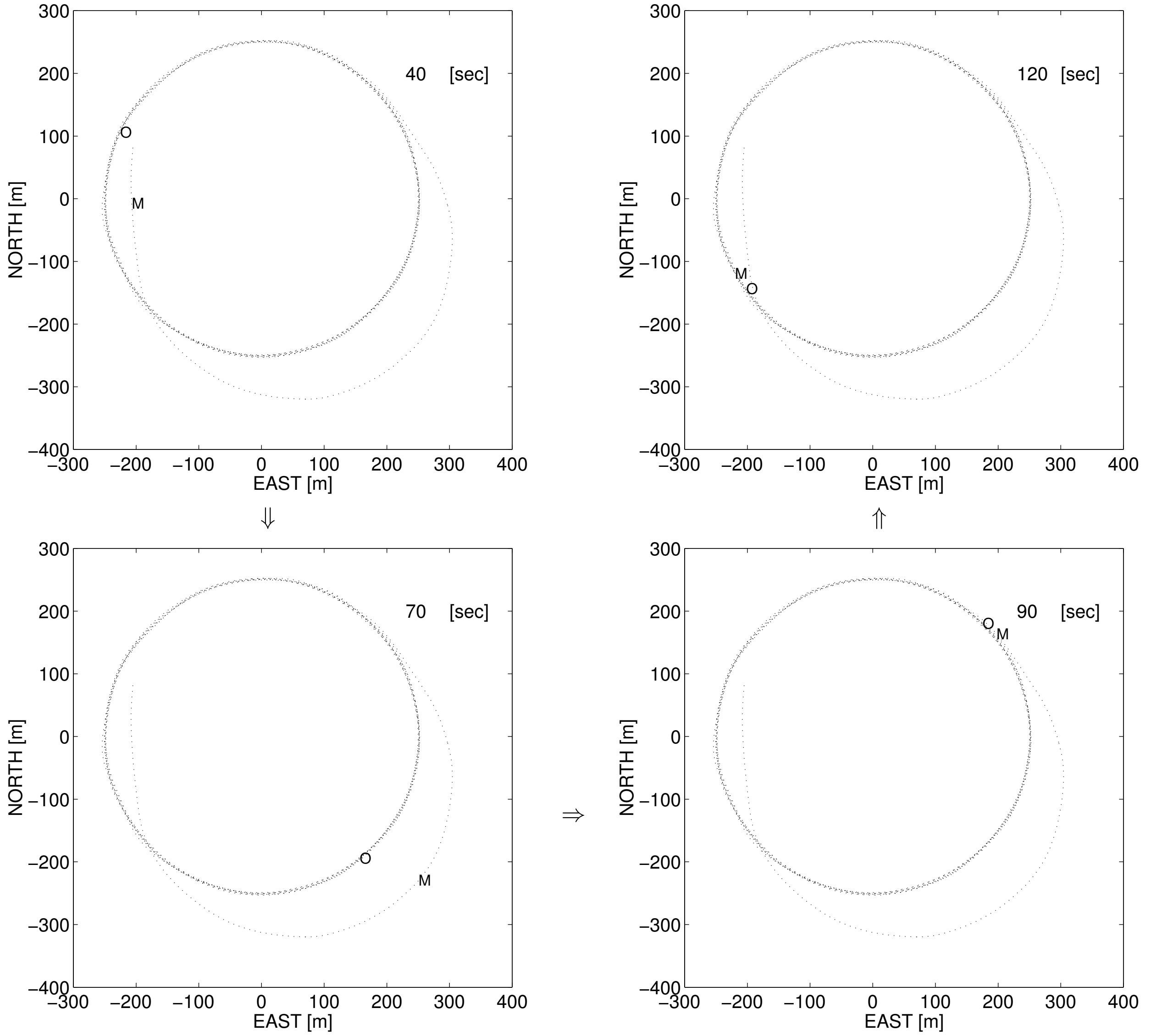


Figure 26: Flight Data - Rendezvous Trajectories of OHS and Mini (O:OHS, M:Mini)

```
% newr.m
% Analyze tracking algorithm by Park et al
% AIAA GNC 2004
%
% Assumes that ac3.m has been run to generate syscl
% Jonathan How
% MIT 16.333 Fall 2004
%
%
close all
dt=1; % time step for the simulation
U0=235.9;
path=[];

jcase=1;
% 2 path cases considered
if jcase==1
    t=[0:5*dt:2500]';
    omp=.0025;
    path=24000*[sin(omp*t) (1-cos(omp*t))];
    xe=0;ye=1500;
    X=[.1 0 0 0*pi/180 0*pi/180 0 0 0]';
else
    t=[0:dt:1350]';
    path(:,1)=U0*t;
    omp=.005;
    path(:,2)=500*(-cos(2*pi*omp*t)+1).*exp(.002*t);
    xe=0;ye=-1000;
    X=[.1 0 0 -15*pi/180 -15*pi/180 0 0 0]';
end

% Discretize the dynamics with time step dt
% system has the inner yaw and roll loops closed
[A,B,C,D]=ssdata(syscl);
syscld=c2d(ss(A,B,C,D),dt);
[Ad,Bd,Cd,Dd]=ssdata(syscld);
Bd=Bd(:,1);Dd=Dd(:,1); % only need first input

% bank angle limit
philim=30;
%
%inputs are phi_d and 0
%state x=[v p r phi Psi xx xx xx]
L1=2000; % look ahead distance
store=[];

% find the point on the path L1 ahead
ii=find((xe-path(:,1)).^2+(ye-path(:,2)).^2 < L1^2);
iii=max(ii);
%
%
%
kk=1;
while (~isempty(iii)) & (kk< length(t)-1)
    kk=kk+1;
    aim_point=path(iii,:);

    xedot=U0*cos(X(5))-X(1)*cos(X(4))*sin(X(5));
    yedot=U0*sin(X(5))+X(1)*cos(X(4))*cos(X(5));

    v1=[xedot yedot]';
    v2=[aim_point(1)-xe aim_point(2)-ye]';
    v1=v1/norm(v1);
    v2=v2/norm(v2);
    [v1 v2];
    temp=cross([v1;0],[v2;0]);
    eta=acos(v1'*v2)*sign(temp(3));
    phi_d=atan(2*U0^2/L1*sin(eta)/9.81);
    phi_d=max(min(phi_d,philim*pi/180),-philim*pi/180);

    store=[store;[t(kk) X' xe ye phi_d v2']];
    % propagate forward a step
    X=Ad*X+Bd*phi_d;
    xe=xe+xedot*dt;
    ye=ye+yedot*dt;

    ii=find((xe-path(:,1)).^2+(ye-path(:,2)).^2 < L1^2);
    iii=max(ii);
end

figure(1);clf
plot(store(:,11),store(:,10),'m');
hold on;plot(path(:,2),path(:,1),'g');
legend('veh','Path');xlabel('Y_e');
ylabel('x_e');setlines(2);hold off
if jcase==1
    axis('square');axis('equal')
    print -depsc park_1; jpdf('park_1')
else
    orient tall
    print -depsc park_1a; jpdf('park_1a')
end

figure(2);clf
plot(store(:,1),store(:,[5 12])*180/pi);
axis([0 t(kk) -philim*1.1 philim*1.1])
xlabel('time');ylabel('\phi and \phi_d');
legend('\phi','\phi_d');setlines(2)
if jcase==1
    print -depsc park_2; jpdf('park_2')
else
    print -depsc park_2a; jpdf('park_2a')
end
return
```