

Topic #23

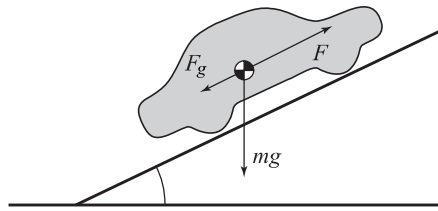
16.30/31 Feedback Control Systems

Analysis of Nonlinear Systems

- Anti-windup
- Notes developed in part from 16.30 Estimation and Control of Aerospace Systems, Lecture 21: Lyapunov Stability Theory by Prof. Frazzoli

Nonlinear System Analysis

- Example: Car cruise control¹



- Equation of motion in the direction parallel to the road surface:

$$m \frac{dv}{dt} = F_{\text{eng}} + F_{\text{aero}} + F_{\text{frict}} + F_g.$$

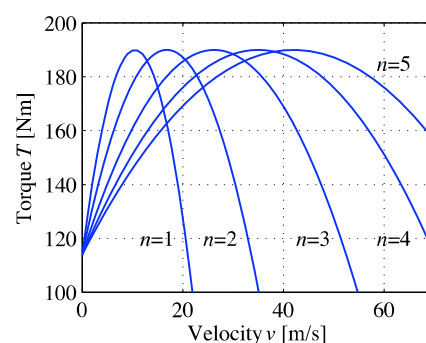
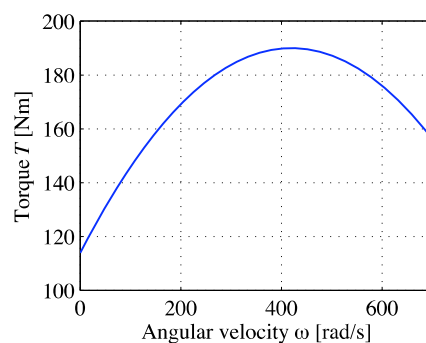
where

$$F_{\text{aero}} = -\frac{1}{2} \rho C_d A v \cdot |v|,$$

$$F_g = -mg \sin(\theta),$$

$$F_{\text{frict}} = -mg C_r \cos(\theta) \text{sgn}(v).$$

- Engine model



- Engine torque (at full throttle): $T_\omega = T_m \left(1 - \beta \left(\frac{\omega}{\omega_m} - 1 \right)^2 \right)$,
where $\omega = \frac{n}{r} v = \alpha_n v$, n is gear ratio, and r wheel radius.
- The engine driving force can hence be written as

$$F_{\text{eng}} = \alpha_n T(\alpha_n v) u, \quad 0 \leq u \leq 1.$$

¹The example is taken from Åström and Murray: Feedback Systems, 2008

Jacobian Linearization

- Any (feasible) speed corresponds to an equilibrium point.
- Choose a reference speed $v_{ref} > 0$, and solve for $dv/dt = 0$ with respect to u , assuming a horizontal road ($\theta = 0$).

$$0 = \alpha_n T(\alpha_n \bar{v}) \bar{u} - \frac{1}{2} \rho C_d A \bar{v}^2 - mg C_r$$

i.e.,

$$\bar{u} = \frac{\frac{1}{2} \rho C_d A \bar{v}^2 + mg C_r}{\alpha_n T(\alpha_n \bar{v})}.$$

- Linearized system ($\xi = v - \bar{v}$, $\eta = u - \bar{u}$):

$$\frac{d}{dt} \xi = \underbrace{\frac{1}{m} \left(\alpha_n \frac{\partial T(\alpha_n v)}{\partial v} \bigg|_{\bar{v}} \bar{u} - \rho C_d A \bar{v} \right)}_{A_{dyn}} \xi + \underbrace{\frac{1}{m} \alpha_n T(\alpha_n \bar{v})}_{B_{dyn}} \eta$$

- Example: numerical values

- Let us use the following numerical values (all units in SI):

$$T_m = 190, \beta = 0.4, \omega_m = 420, \alpha_5 = 10, C_r = 0.01,$$

$$m = 1500, g = 9.81, \rho = 1.2, C_d A = 0.79.$$

- For $\bar{v} = 25$ (90 km/h, or 55 mph), we get $\bar{u} = 0.2497$.
- The linearization yields:

$$A_{dyn} = -0.0134, \quad B_{dyn} = 1.1837$$

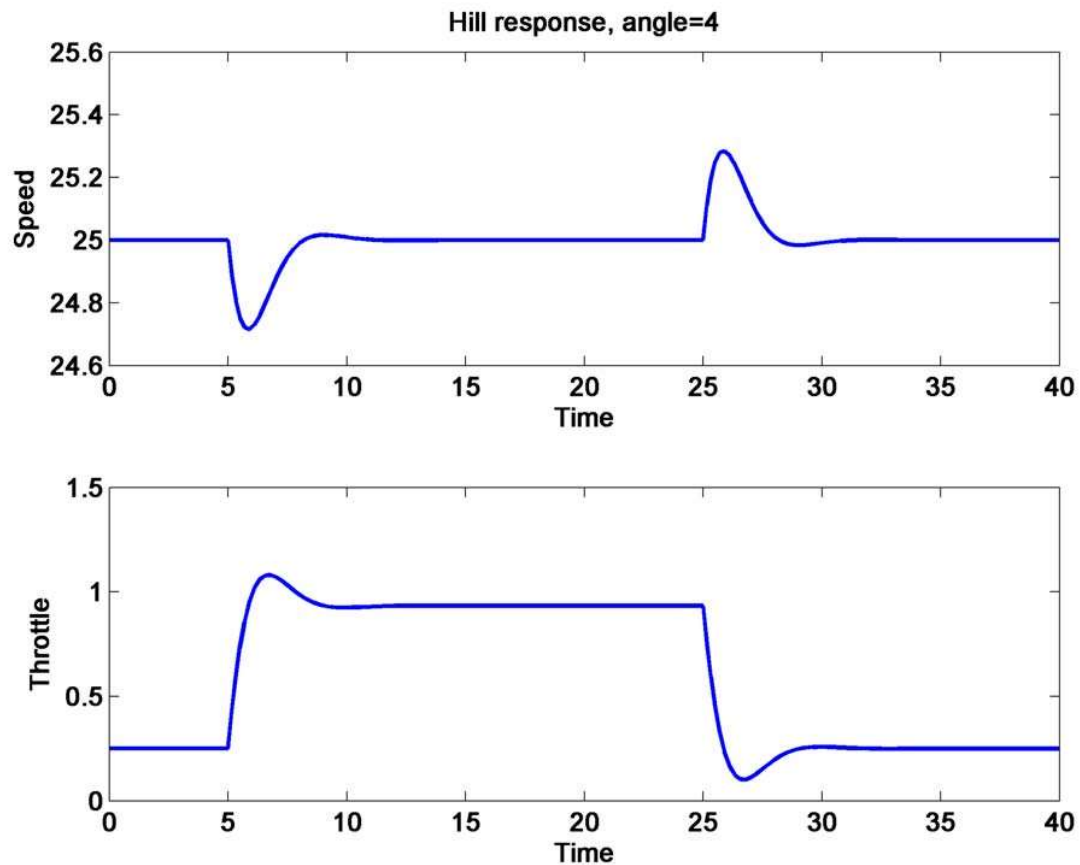
$$\Rightarrow G(s) = \frac{1.1837}{s + 0.0134}$$

Cruise control design

- A proportional controller would stabilize the closed-loop system.
- Assume we want to maintain the commanded speed (cruise control): we need to add an integrator.
- A PI controller will work, e.g.,

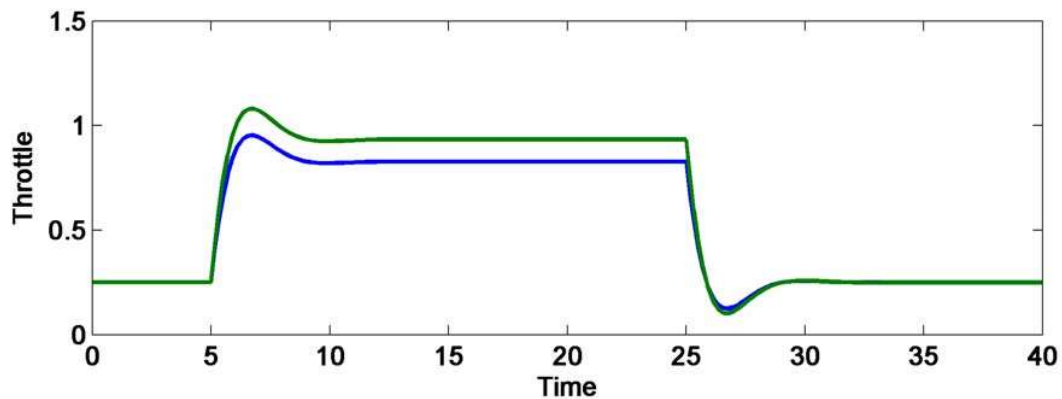
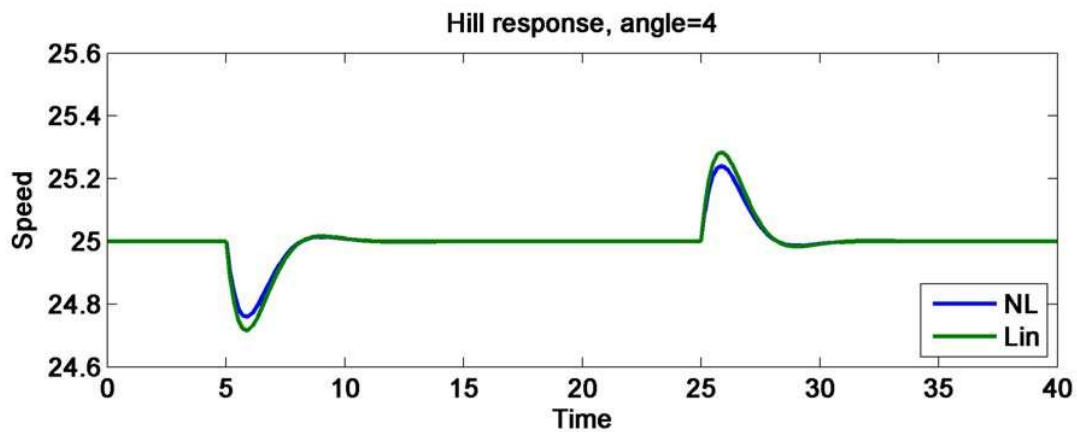
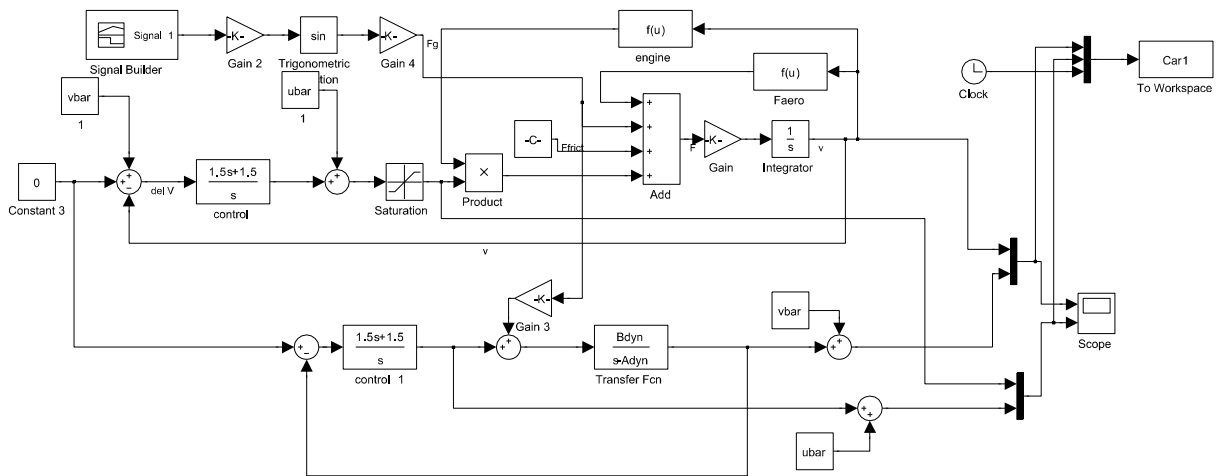
$$C(s) = 1.5 \frac{s+1}{s}$$

- Linear control/model response to hill at specified angle between 5 and 25 sec



Nonlinear Simulation

- Check with BOTH linear AND nonlinear simulation



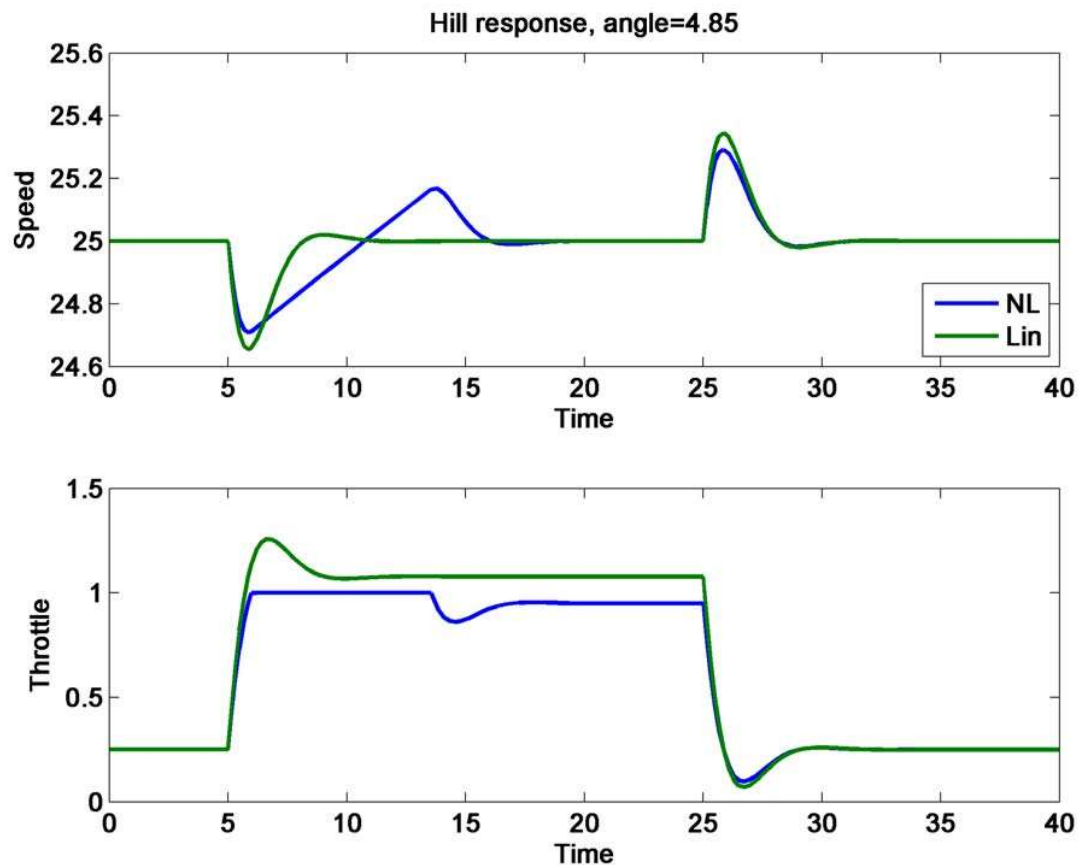
Review

- (Jacobian) linearization:
 - Find the desired equilibrium condition (state and control).
 - Linearize the non-linear model around the equilibrium.

- Control design:
 - Design a linear compensator for the linear model.
 - If the linear system is closed-loop stable, so will be the nonlinear system—in a neighborhood of the equilibrium.
 - Check in a (nonlinear) simulation the robustness of your design with respect to “typical” deviations.

Effects of the saturation

- What if the slope is a little steeper (say 4.85 degrees)?



- What is wrong?
- Systems experiencing **Integrator wind-up**
 - Once the input saturates, the integral of the error keeps increasing.
 - When the error decreases, the large integral value prevents the controller from resuming “normal operations” quickly (the integral error must decrease first!) – so the response is delayed
- **Idea:** once the input saturates, stop integrating the error (can't do much about it anyway!)

Anti-windup Logic

- One option is the following logic for the integral gain:

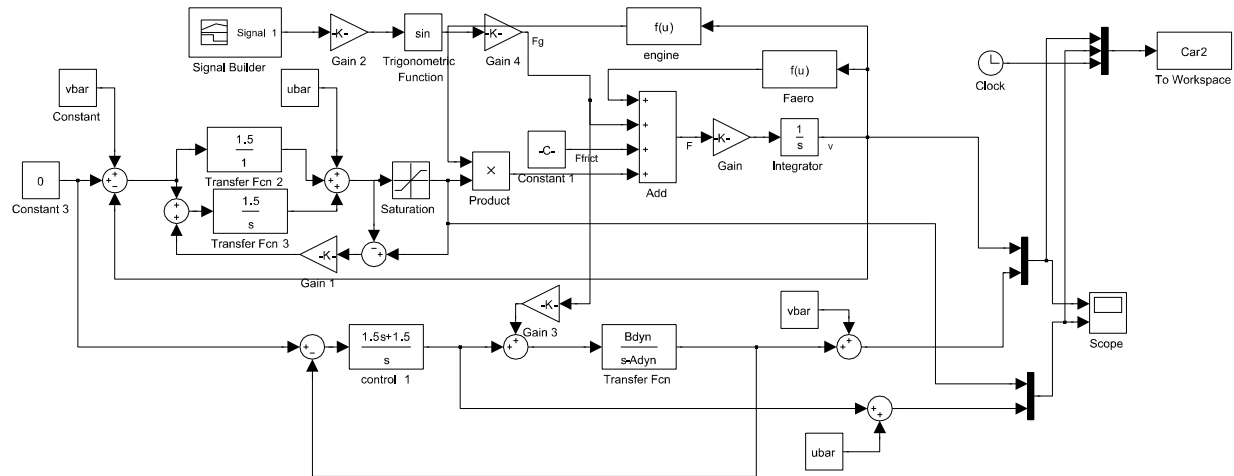
$$K'_I = \begin{cases} K_I & \text{if the input does not saturate;} \\ 0 & \text{if the input saturates} \end{cases}$$

- Another option is the following:
 - Compare the actual input and the commanded input.
 - If they are the same, the saturation is not in effect.
 - Otherwise, reduce the integral error by a constant times the difference.

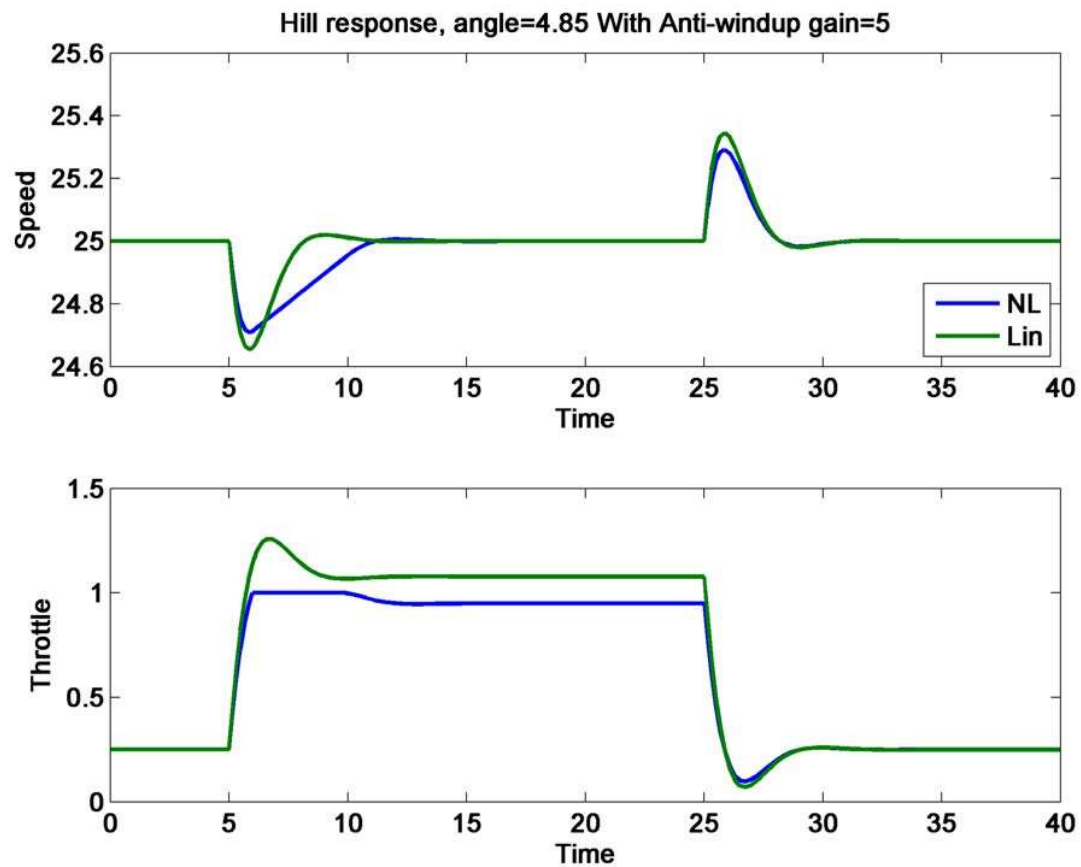
$$u_{int} = \int K_i (e + K_{aw}(\text{sat}(u) - u)) dt$$

- With this choice, under saturation, the actuator/commanded difference is feedback to the integrator so that $e_{act} = \text{sat}(u) - u$ tends to zero
 - Implies that the controller output is kept close to saturation limit.
 - The controller output will then change as soon as the error changes sign, thus avoiding the integral windup.
 - If there is no saturation, the anti-windup scheme has no effect.

Anti-windup Scheme



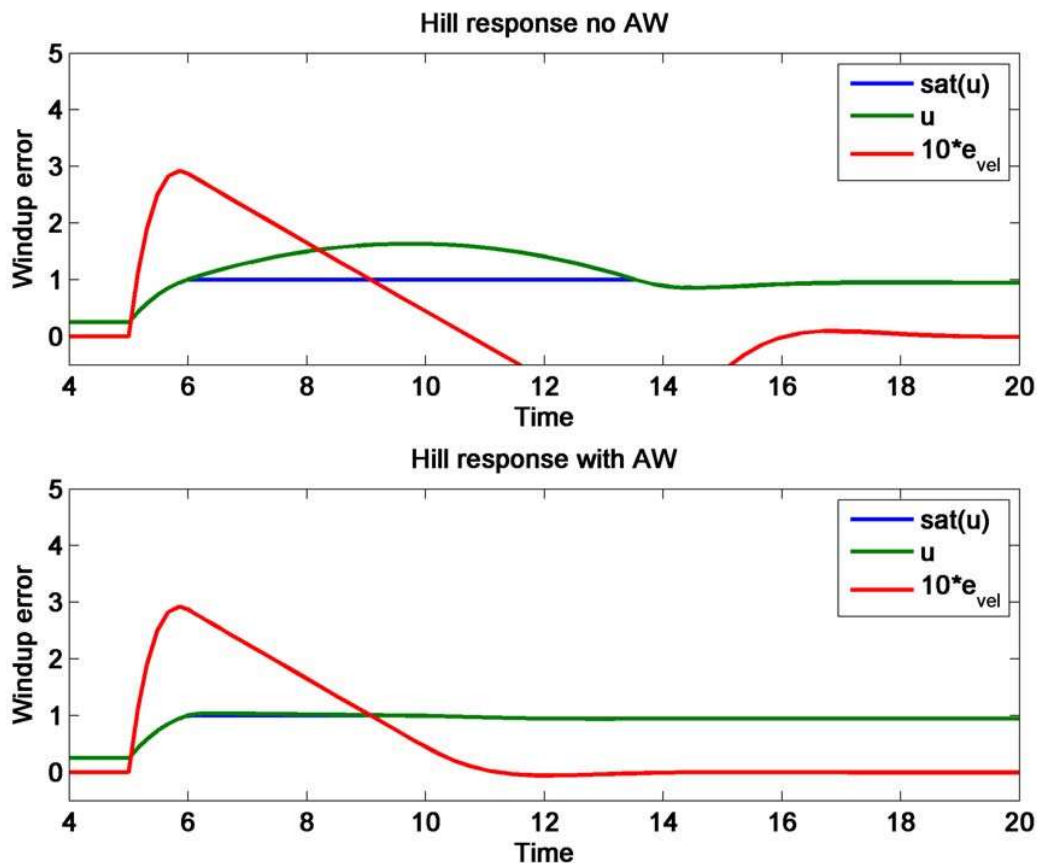
- Response to a 4.85 degree slope



- Anti-windup compensator avoids the velocity overshoot by preventing the error buildup in the integral term

Anti-windup Summary

- Anti-wind up schemes guarantee the **stability of the compensator** when the (original) feedback loop is effectively opened by the saturation.
- Prevent divergence of the integral error when the control cannot keep up with the reference.
- Maintain the integral errors “small.”



Code: Car Setup

```

1  set(0, 'DefaultAxesFontSize', 12, 'DefaultAxesFontWeight','demi')
2  set(0, 'DefaultTextFontSize', 12, 'DefaultTextFontWeight','demi')
3  set(0, 'DefaultAxesFontName', 'arial')
4  set(0, 'DefaultAxesFontSize', 12)
5  set(0, 'DefaultTextFontName', 'arial')
6
7  clear all
8  %close all
9  % Parameters for defining the system dynamics
10 theta=0;
11 gear=5;
12 alpha = [40, 25, 16, 12, 10];      % gear ratios
13 Tm = 190;                          % engine torque constant, Nm
14 wm = 420;                          % peak torque rate, rad/sec
15 beta = 0.4;                        % torque coefficient
16 Cr = 0.01;                         % coefficient of rolling friction
17 rho = 1.2;                         % density of air, kg/m^3
18 A = 2.4;                           % car area, m^2
19 Cd = 0.79/A;                       % drag coefficient
20 g = 9.8;                           % gravitational constant
21 m=1500;                             % mass
22 v=25;
23
24 % Compute the torque produced by the engine, Tm
25 omega = alpha(gear) * v;           % engine speed
26 torque = Tm * ( 1 - beta * (omega/wm - 1)^2 );
27 F = alpha(gear) * torque;
28
29 % Compute the external forces on the vehicle
30 Fr = m * g * Cr;                   % Rolling friction
31 Fa = 0.5 * rho * Cd * A * v^2;     % Aerodynamic drag
32 Fg = m * g * sin(theta);           % Road slope force
33 Fd = Fr + Fa + Fg;                 % total deceleration
34
35 ubar=(Fa+Fr)/(F)
36 vbar=v;
37
38 dTdv=Tm*(-2*beta*(alpha(gear)*vbar/wm-1)*(alpha(gear)/wm))
39 Adyn=(alpha(gear)*dTdv*ubar-rho*Cd*A*vbar)/m;
40 Bdyn=F/m;
41
42 Hillangle=4; % non-saturating angle
43 sim('cruise_control')
44 figure(4);
45 subplot(211);
46 plot(Carl(:,5),Carl(:,2]), 'LineWidth',2);xlabel('Time');ylabel('Speed')
47 title(['Hill response, angle=',num2str(Hillangle)])
48 subplot(212);
49 plot(Carl(:,5),Carl(:,4]), 'LineWidth',2);xlabel('Time');ylabel('Throttle')
50
51 figure(1);
52 subplot(211);
53 plot(Carl(:,5),Carl(:,1:2]), 'LineWidth',2);xlabel('Time');ylabel('Speed')
54 legend('NL','Lin','Location','SouthEast');
55 title(['Hill response, angle=',num2str(Hillangle)])
56 subplot(212);
57 plot(Carl(:,5),Carl(:,3:4]), 'LineWidth',2);xlabel('Time');ylabel('Throttle')
58
59 Antiwindup_gain=0; % us esame code with and without AWup
60 Hillangle=4.85;
61 sim('cruise_control_awup')
62 figure(2);
63 subplot(211);
64 plot(Car2(:,5),Car2(:,1:2]), 'LineWidth',2);xlabel('Time');ylabel('Speed')
65 legend('NL','Lin','Location','SouthEast');
66 title(['Hill response, angle=',num2str(Hillangle)])
67 subplot(212);
68 plot(Car2(:,5),Car2(:,3:4]), 'LineWidth',2);xlabel('Time');ylabel('Throttle')
69
70 Car2.no_AW=Car2;
71
72 Antiwindup_gain=5;
73 sim('cruise_control_awup')
74 figure(3);

```

```

75 subplot(211);
76 plot(Car2(:,5),Car2(:,[1 2]),'LineWidth',2);xlabel('Time');ylabel('Speed')
77 legend('NL','Lin','Location','SouthEast');
78 title(['Hill response, angle=',num2str(Hillangle),' With Anti-windup gain=',num2str(Antiwindup_gain)])
79 subplot(212);
80 plot(Car2(:,5),Car2(:,[3 4]),'LineWidth',2);xlabel('Time');ylabel('Throttle')
81
82 figure(5);
83 subplot(211);
84 plot(Car2_noAW(:,5),Car2_noAW(:,[6 7]),Car2_noAW(:,5),10*Car2_noAW(:,[8]),'LineWidth',2);xlabel('Time');ylabel('Speed')
85 legend('sat(u)','u','10*e-{\vel}','Location','NorthEast');
86 axis([4 20 -0.5 5])
87 title(['Hill response no AW'])
88 subplot(212);
89 plot(Car2(:,5),Car2(:,[6 7]),Car2(:,5),10*Car2(:,[8]),'LineWidth',2);xlabel('Time');ylabel('Windup error')
90 legend('sat(u)','u','10*e-{\vel}','Location','NorthEast');
91 title(['Hill response with AW'])
92 axis([4 20 -0.5 5])
93
94 print -dpng -r300 -f1 AW1.png
95 print -dpng -r300 -f2 AW2.png
96 print -dpng -r300 -f3 AW3.png
97 print -dpng -r300 -f4 AW4.png
98 print -dpng -r300 -f5 AW5.png

```

MIT OpenCourseWare
<http://ocw.mit.edu>

16.30 / 16.31 Feedback Control Systems
Fall 2010

For information about citing these materials or our Terms of Use, visit: <http://ocw.mit.edu/terms>.