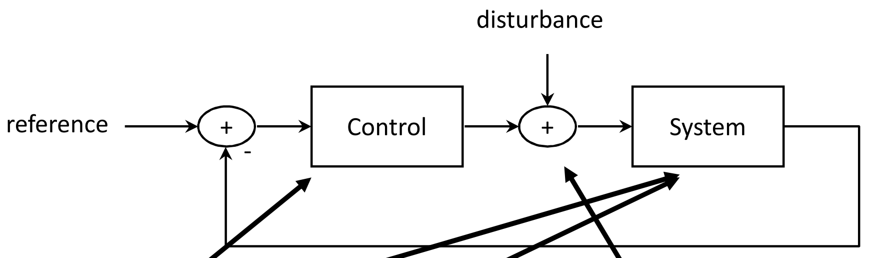
Example: Cruise Controller

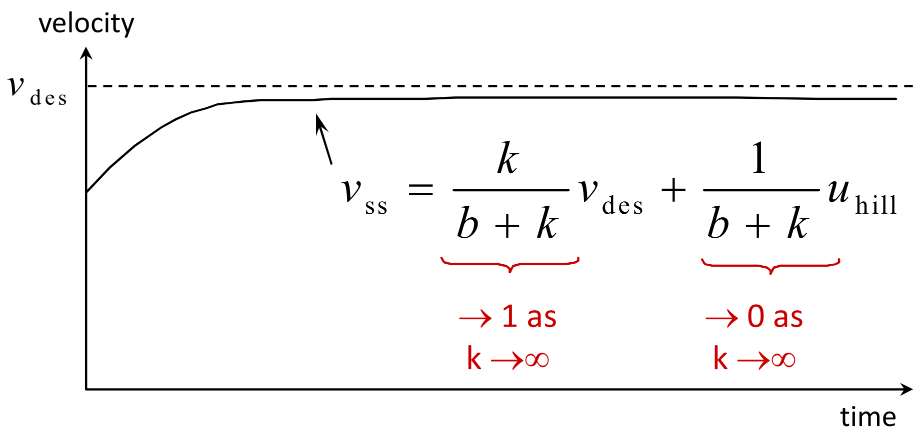
Dr. Mitch Pryor

Previously…



$$m\dot{v} = -bv + f_{\text{hill}} + f_{\text{engine}}$$

$$f_{\text{engine}} = k(v_{\text{desired}} - v)$$

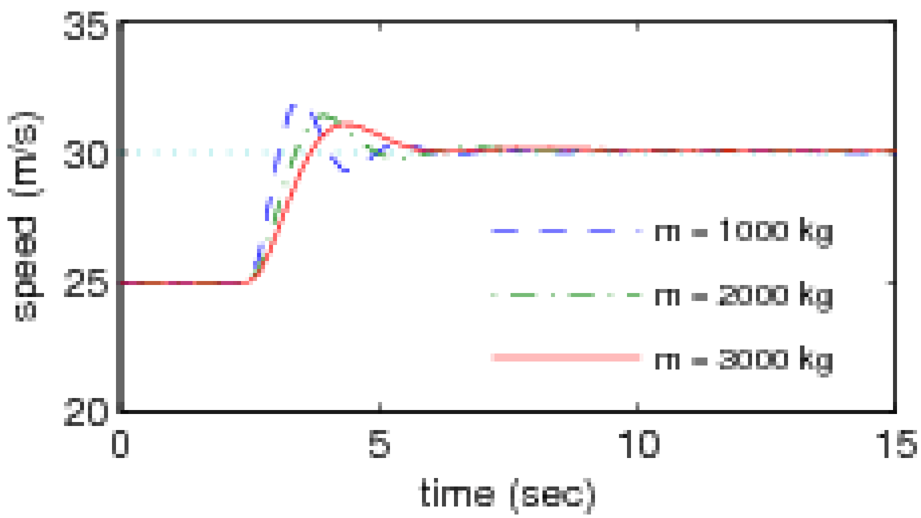


$$v_{ss} = \underbrace{\frac{k}{b+k}}_{\to 1 \text{ as } k \to \infty} v_{\text{des}} + \underbrace{\frac{1}{b+k}}_{\to 0 \text{ as } k \to \infty} u_{\text{hill}}$$

- We discussed some issues with this simple proportional controller
  - The model is overly simplistic
  - The model is incomplete
  - The controller could not eliminated the steady state error

Lesson Objectives

- Let's design and build a controller for a more complicated system
- Introduce a PID controller and its key elements
- Use this example to hone MATLAB capabilities.

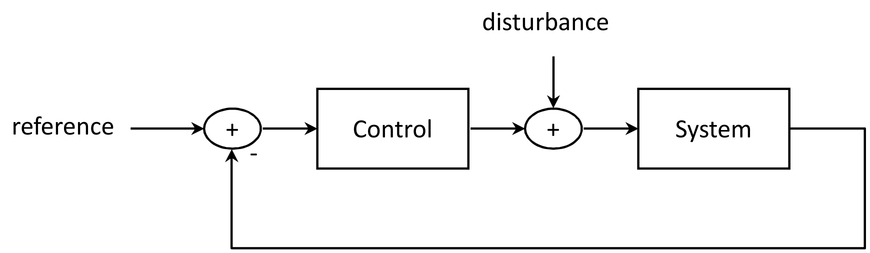


In this example, a controller gets the car up to approx. 67 mph (from approx. 55 mph)

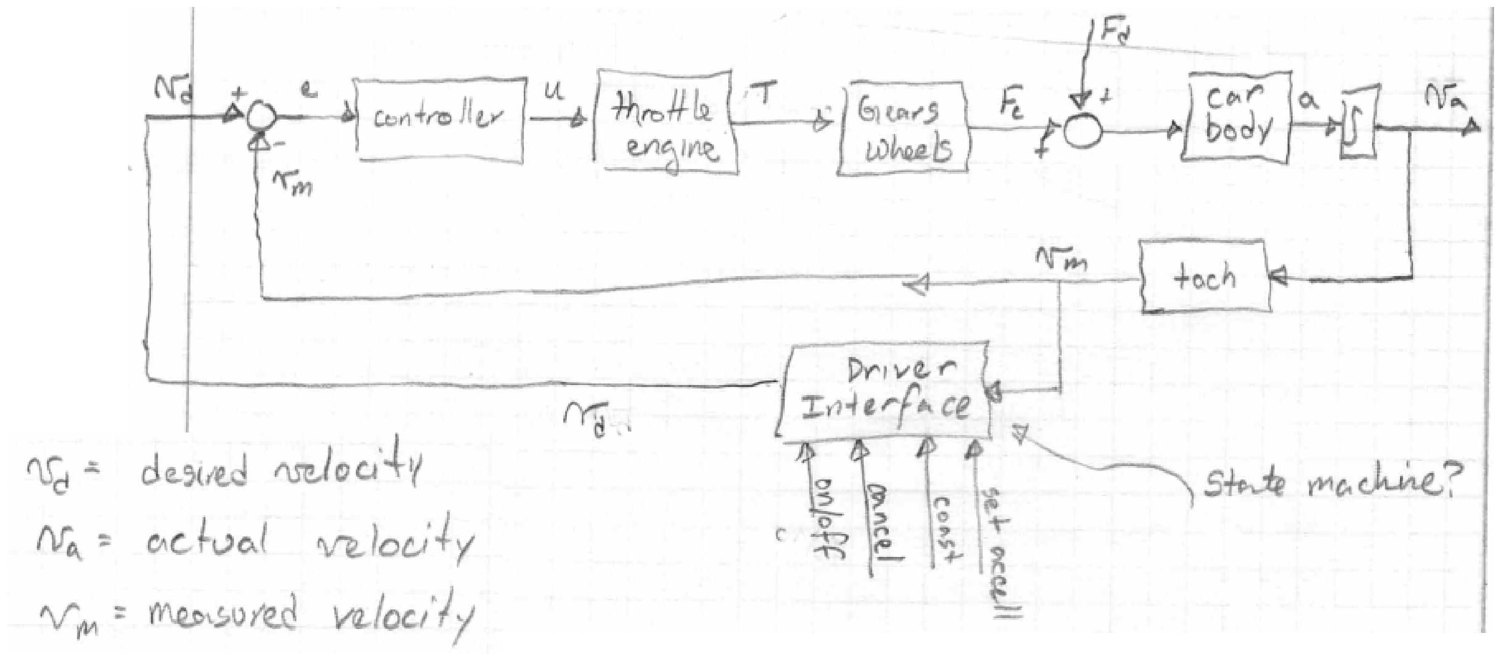
- There is no steady state error.
- Rise time is about 5 seconds.
- The controller works even if the mass varies by a factor of 3.

A more "realistic" system…

From this…



To this…



Dynamic Model

The initial equation is quite simple…

$$m\frac{dv}{dt} = F - F_d$$

m – Total mass including passengers

v = speed of the car

F = force generated by the contact of the wheels with the road

F_d = *total disturbance force due to gravity, friction and aerodynamic drag*

Let's first determine the force *F* generated by the car on the road.

$$0 \le u \le 1$$

u = control signal proportional to fuel injection rate

if *u*=0, then no fuel is producing torque

if *u=1*, then maximum fuel injections (pedal to the metal!!!)

Dynamic Model

And torque is proportional to the injection rate, which is given by the equation.

$$T_e = uT_m \left(1 - \beta \left(\frac{\omega}{\omega_m} - 1 \right)^2 \right)$$

T_e – Torque produced by the engine

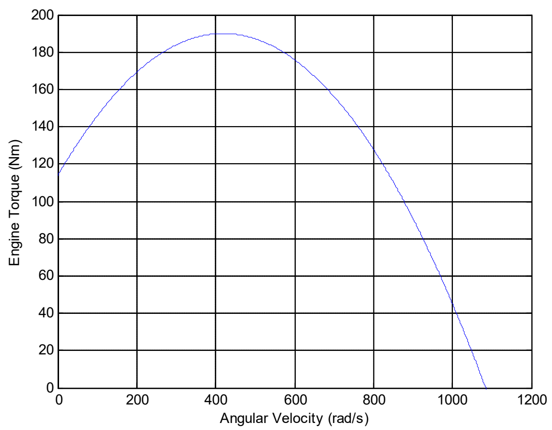
T_m = Maximum torque obtained at the maximum engine speed ω_m

ω_m = Maximum engine speed

ω = current engine speed

β = efficiency coefficient (typically 0.4)

An engine that produces a peak torque of 190 Nm at a speed of 420 rad/sec, has the following torque-speed curve.




```
clear all

Tm = 190; wm = 420; b = 0.4; u=1;
w = [0:1:1200];

for i=1:length(w)
    Te(i) = u*Tm*(1-b*((w(i)/wm)-1)^2);
end

plot( w, Te )
axis( [ 0 1200 0 200 ] )
grid on
xlabel('Angular Velocity (rad/s)')
ylabel('Engine Torque (Nm)')
```

Dynamic Model

To find the force this torque provide, we convert the angular speed to velocity provided via the transmission and wheels

$$\omega = \frac{n}{r} v = \alpha_n v$$

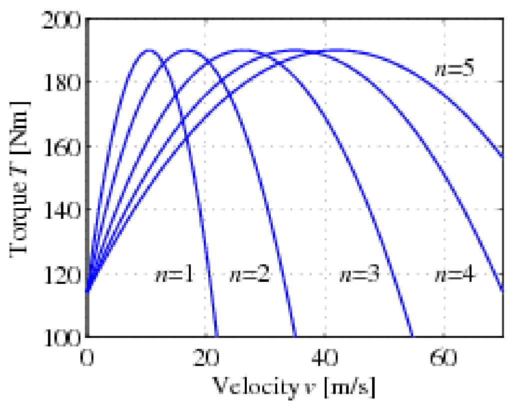
n = gear ratio

r = wheel radius

α_n = mechanical gain for a combination n of wheel radius and gear ratio

v = car velocity

So with some simple modification to code above we have a torque-velocity curve. Where we recognize we need multiple gear ratios (say a 5 speed car) to use the car at close to peak torque over the desired velocity range.



$$F = \frac{n}{r} u T_e(\alpha_n, v)$$

Dynamic Model, cont'd

So we found *F* so we now need to find F_d.

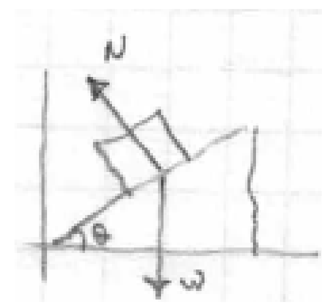
$$m\dot{v} = F - F_d$$
$$= F - (F_g + F_r + F_a)$$

F_g = Force due to gravity (due to hills)
F_r = Force from rolling friction
F_a = Force from aerodynamic drag

We can estimate each of these with the following equations.

$$F_g = mg\sin(\theta)$$
$$F_r = mgC_r\,\mathrm{sgn}(v)$$
$$F_a = \frac{1}{2}\rho C_d A v^2$$

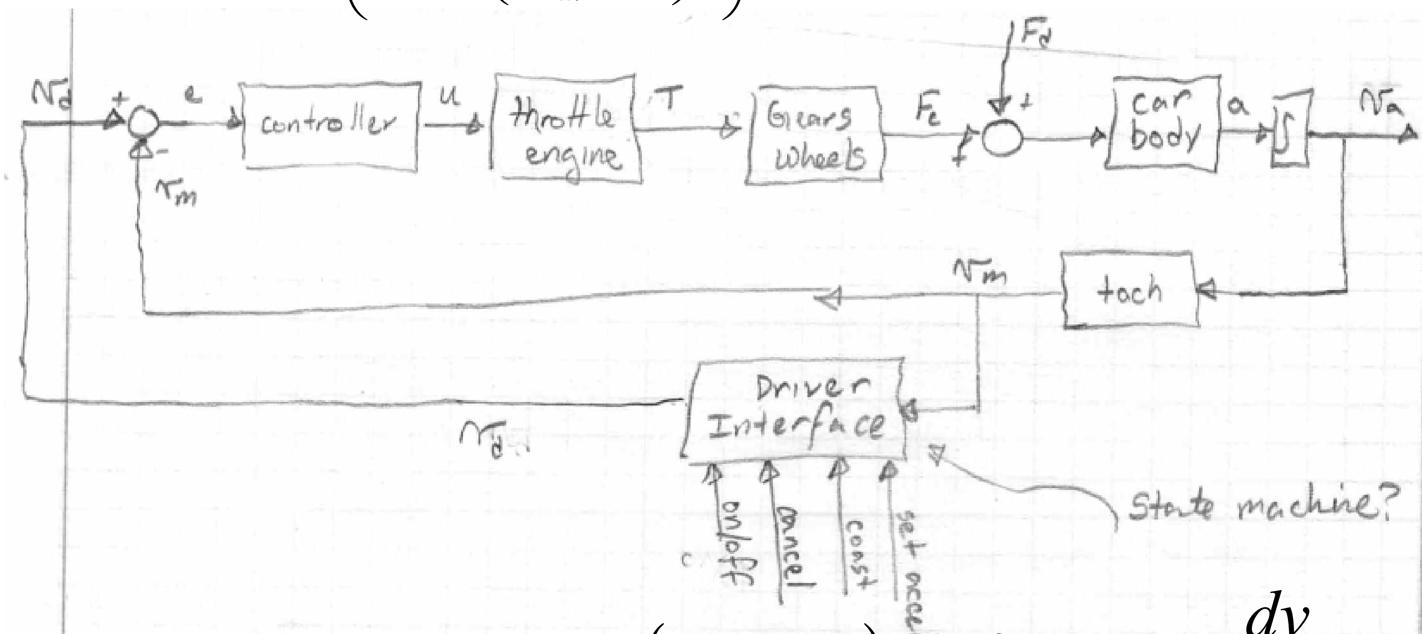
g = gravity constant
C_r = Coefficient of rolling friction
C_d = *Drag coefficient*
ρ = air density
A = front cross-sectional area of car



Controller Design

Plugging it all in, our system model is one highly nonlinear 1st order ODE.

$$m\frac{dv}{dt} = \alpha_n u T_m\left(1-\beta\left(\frac{\alpha_n v}{\omega_m}-1\right)^2\right)-(mg\sin(\theta)+mgC_r\operatorname{sgn}(v)+\frac{1}{2}\rho \mathrm{A} v^2)$$



Objective: Pick *u such that* $e=\left(v_d-v_m\right)\rightarrow 0$ and $\dfrac{dv}{dt}=0 @ v_d$

Controller Design

<u>Objective</u>: Pick *u such that* $e=\left(v_d - v_m\right)\rightarrow 0$ and $\dfrac{dv}{dt}=0 @ v_d$

<u>Possible solutions</u>: Bang-bang control.

$$u=\begin{cases} u_{\max}=1 & v<v_d \\ u_{\min}=0 & v>v_d \end{cases}$$

Recipe for carsick passengers.

$$u=\begin{cases} u_{\max}=1 & v<v_d-\Delta v_{tolerance} \\ u_{\min}=0 & v>v_d+\Delta v_{tolerance} \end{cases}$$

Slightly less sick passengers, but now we have a car that is continuously passed and being passed on the highway.

Controller Design

Objective: Pick *u such that* $e = \left(v_d - v_m\right) \to 0$ and $\frac{dv}{dt} = 0 @ v_d$

Possible solutions: Proportional Control

$$u(t) = \begin{cases} u_{\max} & if \quad e > e_{\max} \\ k_p e & if \quad e_{\min} < e < e_{\max} \\ u_{\min} & if \quad e < e_{\min} \end{cases}$$

- Better than bang-bang control
- (e_{min},e_{max}) define the *proportional band*
- But if some input is required to maintain a zero error, then this controller is insufficient since the control input is 0 when the error is 0.

Dynamic Model, cont'd

<u>Objective</u>: Pick *u such that* $e = \left(v_d - v_m\right) \to 0$ and $\dfrac{dv}{dt} = 0 @ v_d$

<u>Possible solutions</u>: Proportional plus feedforward control

$$u(t) = \begin{cases} u_{\max} & if \quad e > e_{\max} \\ u_{ff} + k_p e & if \quad e_{\min} < e < e_{\max} \\ u_{\min} & if \quad e < e_{\min} \end{cases}$$

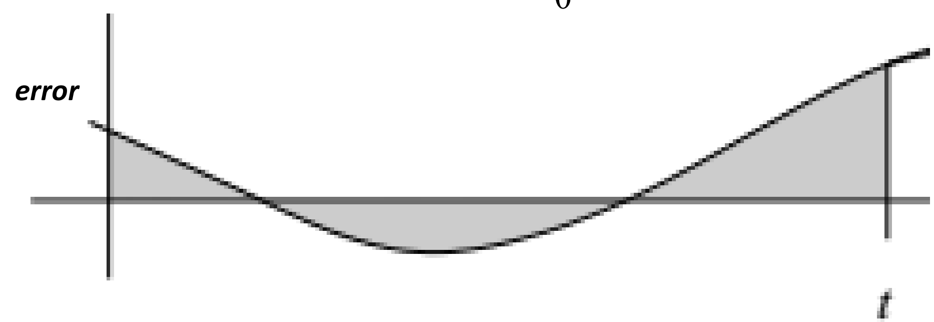
- Provides a baseline *feed forward* input if input is required to maintain zero error
- Feedforward input is basically an "open loop" addition to the control
  - It will not be robust to disturbances or changes in system parameters
  - For example, a change in passenger weight or wind conditions

Controller Design

<u>Objective</u>: Pick *u such that* $e = \left(v_d - v_m\right) \to 0$ and $\frac{dv}{dt} = 0 @ v_d$

<u>Possible solutions</u>: Proportional plus Integral (PI) Control

$$u(t) = k_P e(t) + k_I \int_0^t e(\tau)dt$$



- Eliminates steady state error by reducing the total “historical” error to zero.
- However, if not implemented properly, there may not be a “steady state” due to oscillation.

Controller Design

Objective: Pick *u such that* $e = \left(v_d - v_m\right) \rightarrow 0$ and $\frac{dv}{dt} = 0 @ v_d$

Possible solutions: Proportional plus Derivative (PD) Control

$$e(t + T_d) \approx e(t) + T_d \frac{de(t)}{dt}$$

which predicts the error T_d time units ahead.

$$u(t) = k\left(e(t) + T_d \frac{de(t)}{dt} \right) = k_P e(t) + k_D \frac{de(t)}{dt}$$

- May reduce oscillations in the controlled output
- Utilizes the fundamental 1st order nature of our state-space models
- Doesn't fix the steady state error by itself
- Can be problematic if measured output signal is noisy (may need a filter)

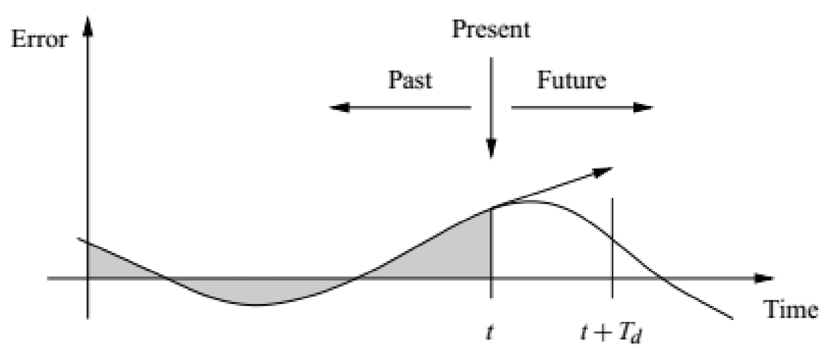
Controller Design

<u>Objective</u>: Pick *u such that* $e = \left(v_d - v_m\right) \to 0$ and $\dfrac{dv}{dt} = 0 @ v_d$

<u>Possible solutions</u>: Proportional Integral Derivative (PID) Controller

$$u(t) = k_P e(t) + \mathrm{k}_I \int_0^t e(\tau)dt + k_D \frac{de(t)}{dt}$$

- A controller that eliminates steady-state error
- Address the “past, present, and future” error in the controller
- More than 95% of all industrial control problems are solved using a PID Controller



Controller Summary

$$u(t) = K_p e(t) + K_I \int_0^t e(\tau)\,d\tau + K_d \frac{de(t)}{dt} = K\left(e(t) + \frac{1}{T_i}\int_0^t e(\tau)\,d\tau + T_d \frac{de}{dt} \right)$$

- This "textbook" version of the controller **can**
  - Minimize the error
  - Eliminate the steady state error
  - Work for most applications
- The controller is physically intuitive.
- Sufficient design and testing can be done with a reasonably good model.
- In our case, does the controller work
  - over a range reasonable car weights?
  - on a range reasonable hills?
  - Over a frequency of wind conditions?
  - In a way that does not distress the driver or passengers?

- This "textbook" version of the controller **cannot** alone account for
  - Noise filtering
  - High frequency roll-off
  - Wind-up in the presence of controller saturation
  - Set-point weighting
- Can be difficult to design/tune (i.e. gain selection)
- Can be difficult to implement using a computer that reasonably accounts for real world situations