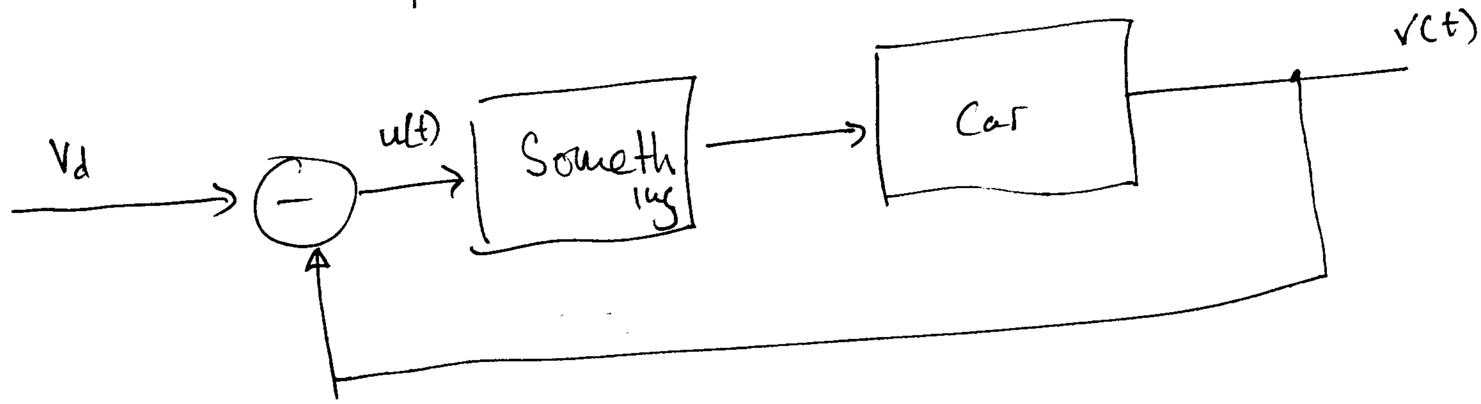
Very Simple control: ①

Our car has an accelerator input, bu[t]
we want fixed velocity

Q: Now what?

Idea: compare velocity to desired, use
to compute accel



Simplest case:

$$u(t) = V_d - V(t)$$

→ makes sense – ~~0 inpu~~ $V(t) = V_d(t)$
⇒ nothing

What happens?

$$\frac{dv(t)}{dt} = \alpha \left[V_d - V(t)\right]$$

so $\quad V(t) = A e^{-\alpha t} + B$

$V(0) = 0 \quad \Rightarrow \quad B = -A.$

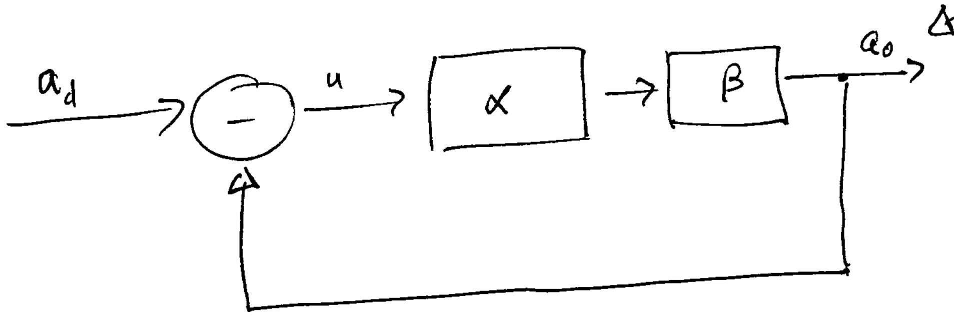
$\dot{V} + \alpha V = \alpha V_d \quad$ so $\quad B = V_d$

so $\quad V(t) = \cancel{\alpha} V_d \left(\cancel{e}\, 1 - e^{-\alpha t}\right)$

which is nice.
But Prop control doesn't always work so well.

Simple example: weird ~~veloct~~ acceleration on car (uncalibrated)

$$a_{out} = \beta\, a_{in}$$



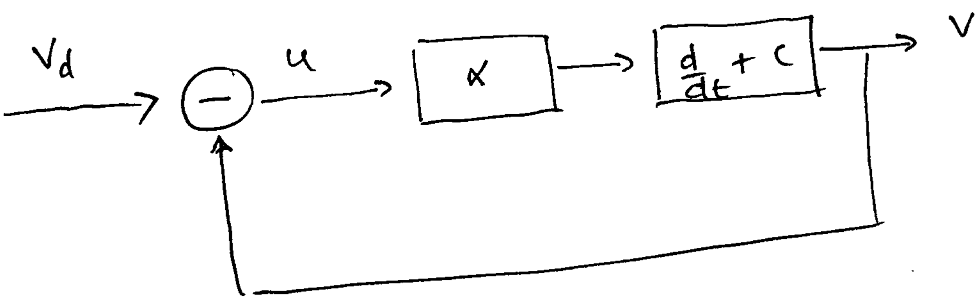
This is a NOT v

$$u = a_d - a_o$$

$$a_o = \alpha . \beta . u$$

$$\text{so} \quad a_o = \frac{\alpha \beta a_d}{1 + \alpha \beta}.$$

bias - you might fix with big α, but that causes other problems



$$u = (v_d - v)$$

$$\alpha \left[\frac{du}{dt} + c u \right] = v$$

$$\text{so} \quad \alpha c v_d = \alpha \dot{v} + (1 + \alpha c) v$$

$$\left[\frac{1 + \alpha c}{\alpha} \right]$$

$$\text{so} \quad v = A e^{-\left(\frac{\cancel{\alpha}}{\cancel{1+\alpha c}}\right) t} + B.$$

$$\text{and} \quad B = \frac{\alpha c v_d}{1 + \alpha c}, \quad A = -B \quad /$$

Notice we have two issues

- slow resp (time constant is $\frac{\alpha}{1+\alpha c}$ $\cancel{\frac{1+\alpha c}{\alpha}}$)

- wrong answer.

~~each coul~~ wrong answer could be addressed by making α big.

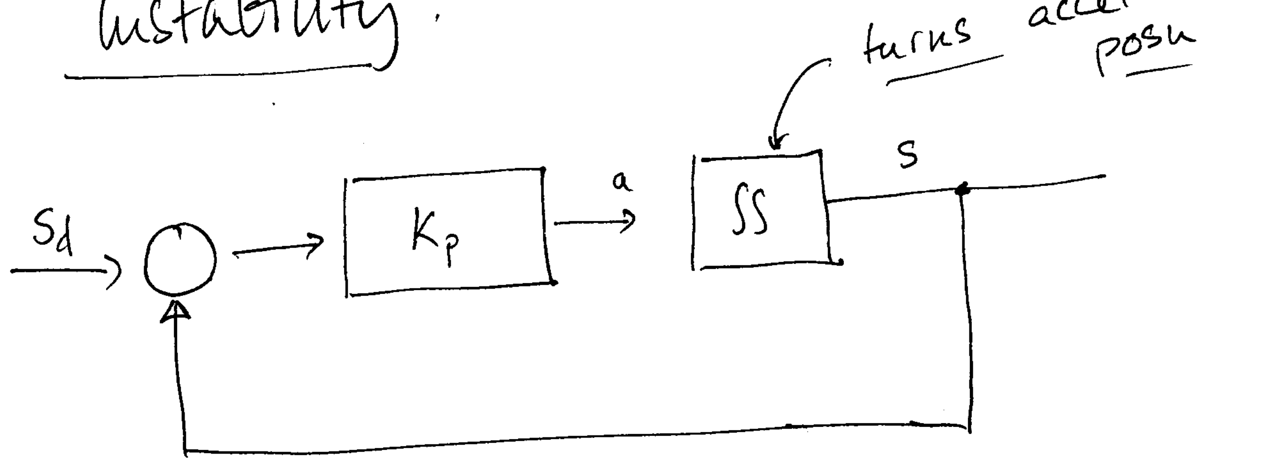
But what if $1+\alpha c < 0$?

(i.e. if $c<0$, big α might lead to serious trouble!).

INSTABILITY

. we tried to make it respond fast, but now it's unstable

Here is another example of <u>instability</u>:



$$\frac{d^2 s_o}{dt} = a = K_p(S_d - S_o).$$

so $$\frac{d^2 s_o}{dt} + K_p S_o = K_p S_d$$

Soln $A e^{zt} + Bt + C$

usual stuff reveals $B = 0$, $C = S_o$

$z^2 + K_p = 0$

consider ~~$|K_p|$~~ $K_p > 1$!

(too big a K_p is a problem)

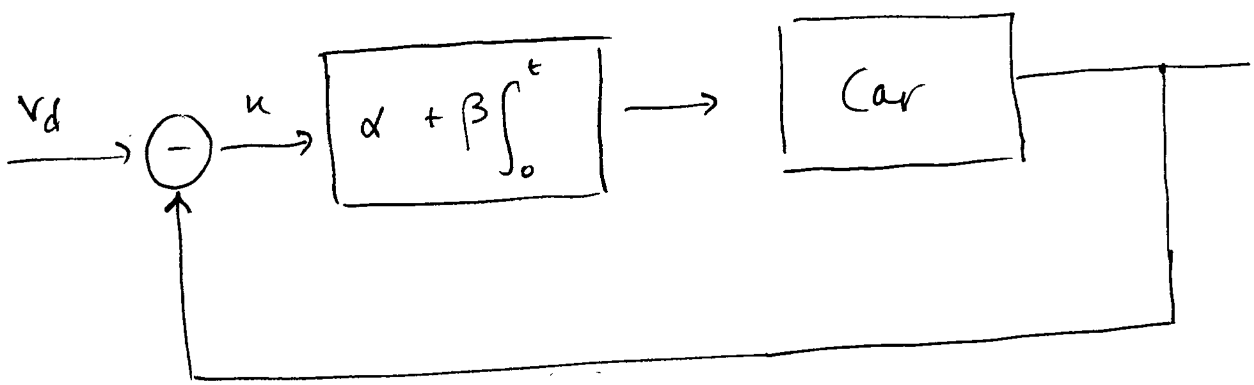
Strategy:

Feedback $\int_0^t [v(t) - v_d]\, dt$.

(Integral term)

as well.

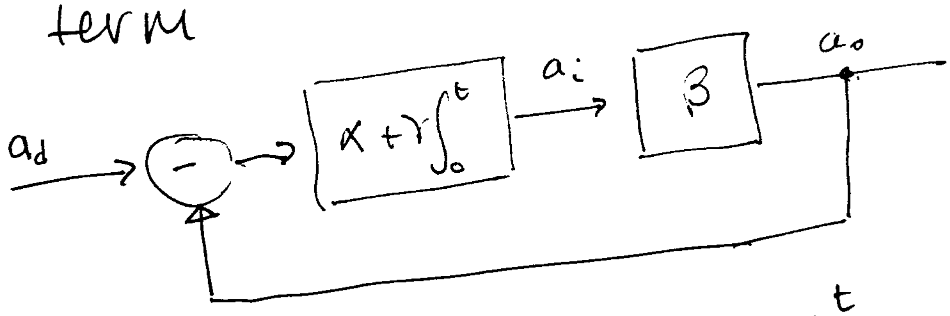
so



Why? because either $v(\tilde{t}) - v_d = 0$ for $\tilde{t} >$ some time, OR input grows to infinity

$\longrightarrow$ System is unstable OR has 0 steady state error.

Recall P. 3 example – now use $\int$ term



$$a_o = \beta a_i = \beta\left[\alpha(a_d - a_o) + \gamma\int_0^t (a_d - a_o)\,dt\right]$$

so
$$a_o + \frac{\beta\gamma}{1+\beta\alpha}\int_0^t (\cancel{a_d}\, a_o)\,dt = \frac{\alpha\beta\, a_d}{1+\alpha\beta} + \frac{\beta\gamma}{1+\alpha\beta}\int_0^t a_d\,dt$$

Differentiate

$$\frac{da_o}{dt} + \frac{\beta\gamma}{1+\beta\alpha}\,a_o = \frac{\beta\gamma}{1+\alpha\beta}\cdot a_d .$$

$$a_o = a_d\left[1 - e^{-\left[\frac{\beta\gamma}{1+\beta\alpha}\right]t}\right]$$

<u>Notice</u> :

- Steady state error has gone
- But you could have stability problems

want $\dfrac{\beta\gamma}{1+\beta\alpha} > 0$

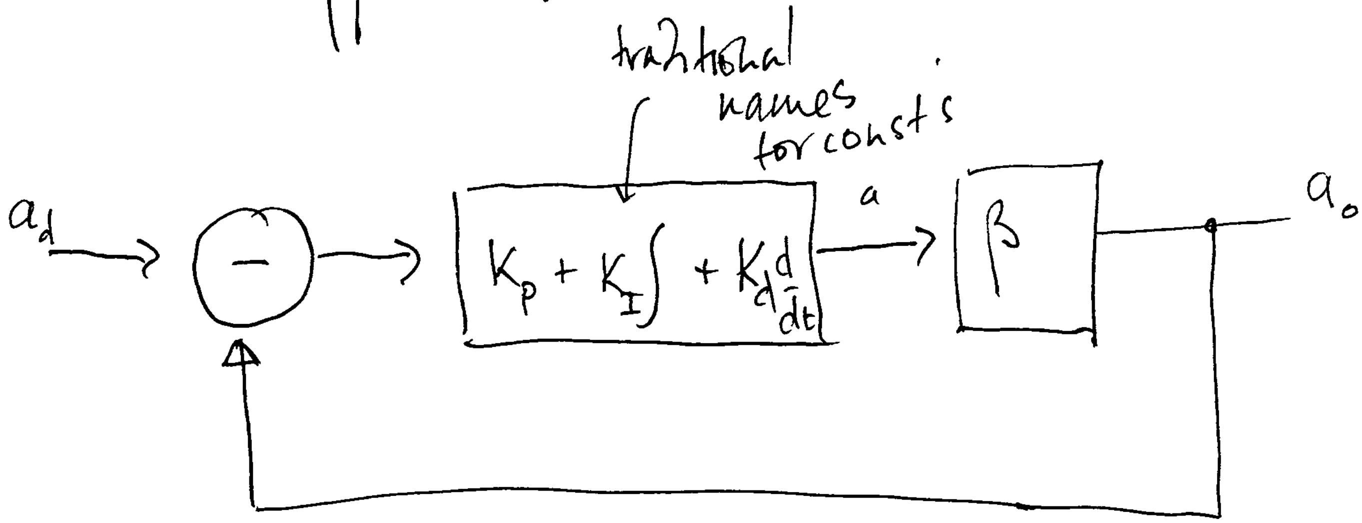
- system may respond slowly with small γ, big α.

- if I don't know system
  - already seen big α is dangerous.
  - big γ may be dangerous too. — input is large who knows what might happen?

Strategy:

- feed back derivative of error
- encourage faster resp.

What happens?



$$a_o = \beta a = \beta \left[K_p (a_d - a_o) + K_I \int_0^t (a_d - a_o)\, dt + K_D \frac{d}{dt}(a_d - a_o) \right]$$

So $$\left[1 + \beta K_p\right] a_o + \beta K_I \int_0^t a_o\, dt + \beta K_D \frac{da_o}{dt} = \beta K_p a_d + \beta K_I \int a_d\, dt + \beta K_D \frac{d}{dt} a_d$$

assume a_d is const, $t > 0$,

$0 \quad t < 0$

step input.

differentiate, get

$$a \frac{d^2 a_0}{dt} + b \frac{d a_0}{dt} + c a_0 = c a_d$$

$$\left[a = \beta K_D \; , \; b = 1 + \beta K_P \; , \; c = \beta K_I . \right]$$

Soln:

$$A e^{zt} + B e^{\bar{z}t} + Ct + D .$$

Check:

$$A\left[az^2 + bz + c\right] e^{zt} = 0$$

$$B\left[a\bar{z}^2 + b\bar{z} + c\right] e^{\bar{z}t} = 0$$

$$\cancel{bC +} \; cCt = 0 \Rightarrow C = 0$$

$$bC + cD = c a_d \Rightarrow D = a_d .$$

so $z, \bar{z}$ roots of

$$a x^2 + b x + c = 0 \; !$$

(usual thing!)

<u>Cases</u>:

- <u>roots are real.</u>
  - if either is non-negative, we're in trouble
    → there is a soln that grows w/ t.
  - try to avoid real roots unless v. negative
  - ~~try to a~~
- <u>roots are complex conj's:</u>
  - solns look like

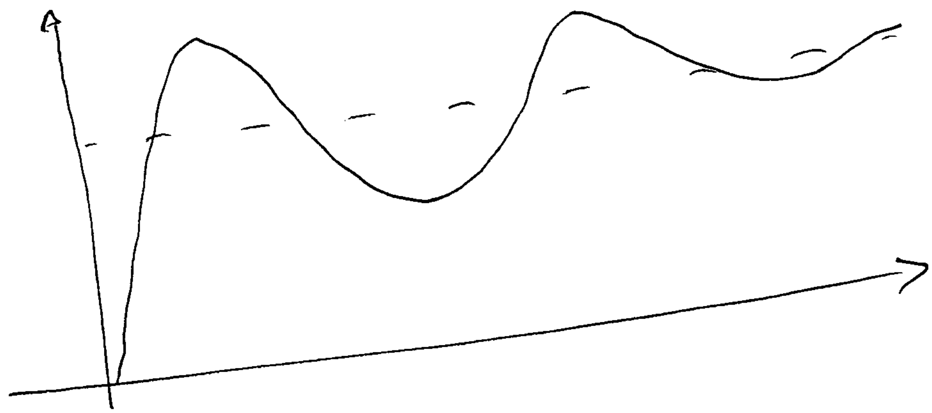
$$\cancel{e^{rt} + e}$$

$$e^{[R+iI]t} = e^{Rt} \cdot e^{iIt}$$

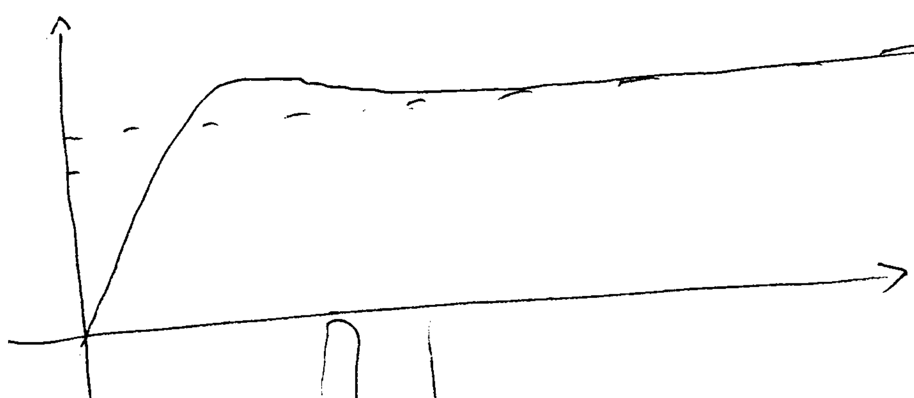
damping   oscillatio

  - sign of R is important
  - $R < 0$ — stable
    $R > 0$ — aargh! soln grows.

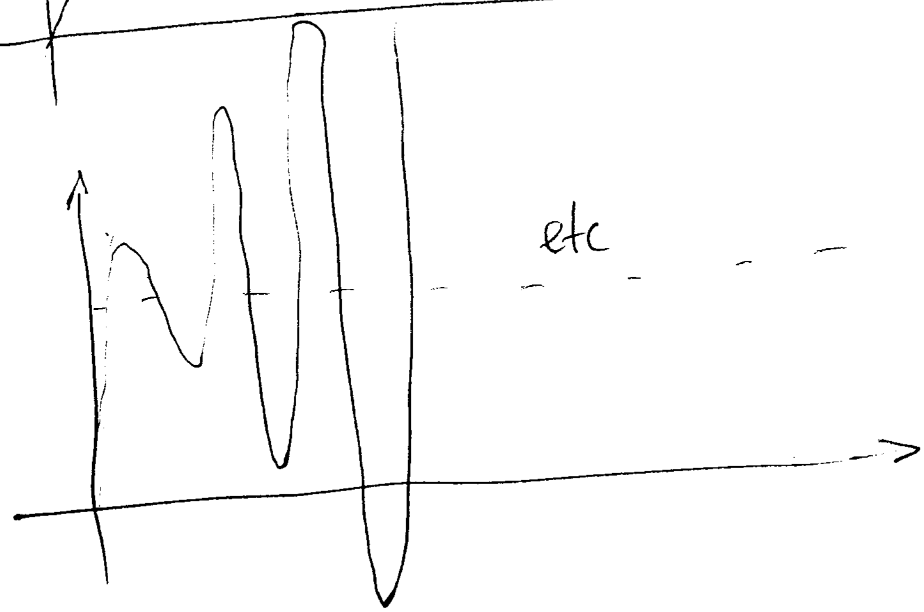
over ~~under~~damped



under ~~over~~damped



critically damped



unstable

Almost all of practical control:

- Build a PID controller
- adjust K_P, K_I, K_D till you're happy
- Rough guide
  - set K_I, K_D = 0
  - adjust K_P until it "just" oscillates
  - now $K_P = \frac{1}{2}$ that value
  - now increase K_I until offset is corrected reasonably fast
  - if too slow increase K_d

- Best way is to do this on Sim
- There are other strategies!