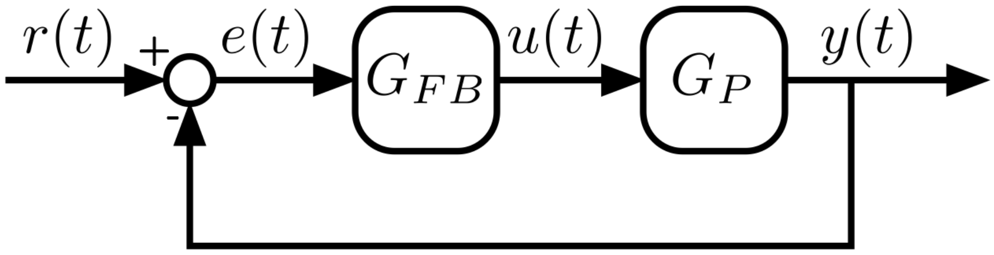
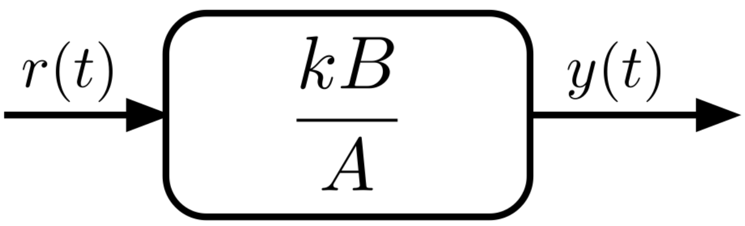
Tracking Control



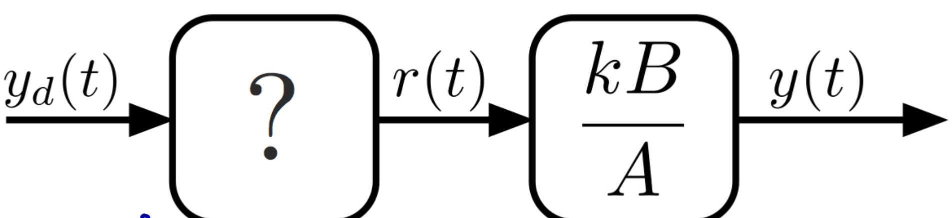
Q: What should $r(t)$ be such that $y(t) = y_d(t)$?

Let's simplify the block diagram to investigate

or generically

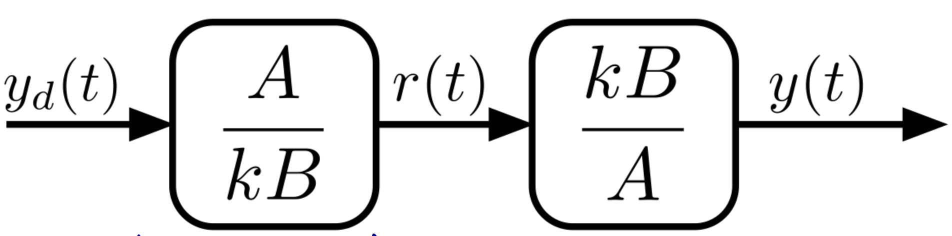


Q: So, what should $r(t)$ be?



What should this be?

Let's try the inverse of the system.



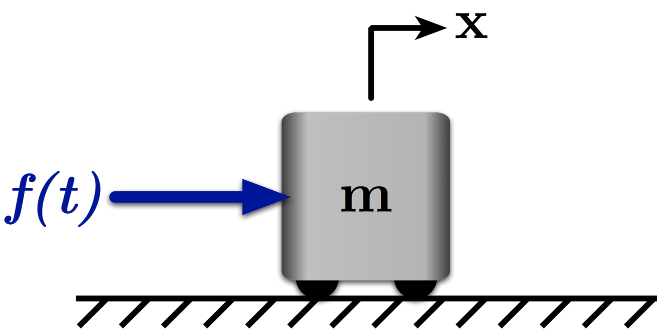
Plant Inversion

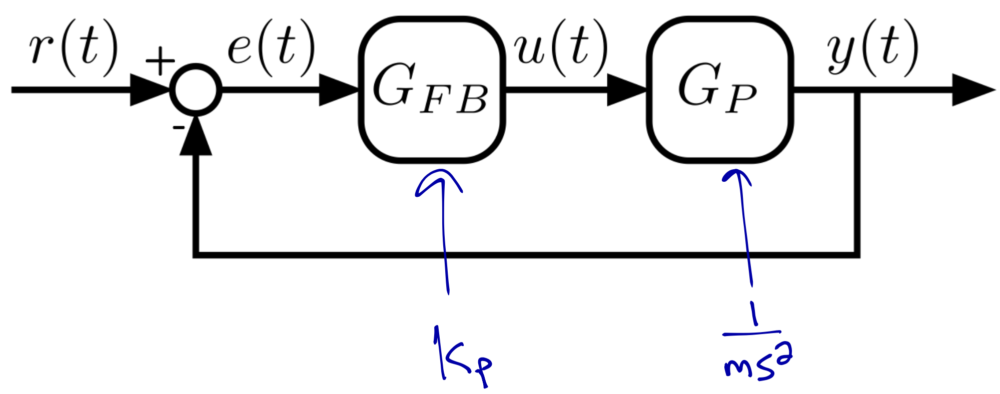
$$r(t) = \frac{A}{kB} y_d(t)$$

$$\underbrace{y(t) = \frac{kB}{A} r(t) = \frac{\cancel{kB}}{\cancel{A}}\left[\frac{\cancel{A}}{\cancel{kB}} y_d(t)\right]}_{y(t) = y_d(t)}$$

Plant Inversion Example

system is just a mass under prop. control.





In simplified form:

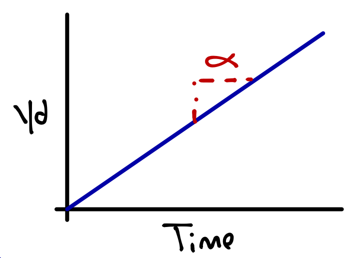
$$r(t) \rightarrow \boxed{\frac{K_p}{ms^2+K_p}} \rightarrow y(t)$$

define $\omega_n = \sqrt{\frac{K_p}{m}}$

$$r(t) \rightarrow \boxed{\frac{\omega_n^2}{s^2+\omega_n^2}} \rightarrow y(t)$$

So, for this system $KB = \omega_n^2$ and $A = s^2+\omega_n^2$

Let's try to track a ramp: $\underbrace{y_d(t) = \alpha t}_{\text{In the Laplace domain}}$



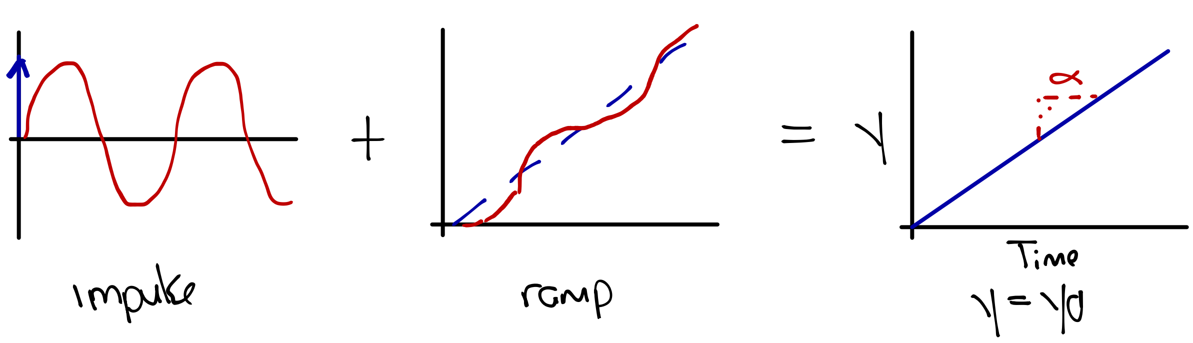
$$Y_d(s) = \frac{\alpha}{s^2}$$

$$r(t) = \frac{A}{KB}\, y_d(t) \quad r(s) = \frac{A}{KB}\, Y_d(s) = \left(\frac{s^2+\omega_n^2}{\omega_n^2}\right)\left(\frac{\alpha}{s^2}\right)$$

$$= \alpha\left(\frac{s^2+\omega_n^2}{s^2\omega_n^2}\right) = \underbrace{\frac{\alpha}{\omega_n^2}\left(1 + \frac{\omega_n^2}{s^2}\right)}$$

converting back to time domain via inverse Laplace Trans.

$$r(t) = \frac{\alpha}{\omega_n^2}\Big(\underbrace{\delta(t)}_{\text{impulse}} + \underbrace{\omega_n^2 t}_{\text{ramp}}\Big)$$



Plant Inversion Example (cont.)

Q: What's the problem with this?

Look at the force this r(t) generates:

$$y(s) = \frac{f(s)}{ms^2} \longrightarrow f(s) = ms^2 y(s)$$

$$= m\cancel{s^2}\left(\frac{\alpha}{\cancel{s^2}}\right)$$

$f(s) = m\alpha \longrightarrow$ convert to time domain $\rightarrow$ $f(t) = m\alpha\underbrace{\delta(t)}$

This is a problem!

We cannot generate an ∞ force

Q: What other problems are there?

We've assumed our model exactly matches the "real" system...

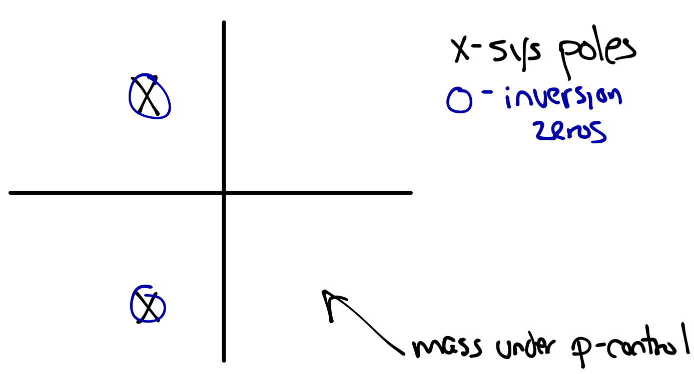
But, it never exactly matches. So, we don't get the nice cancellation

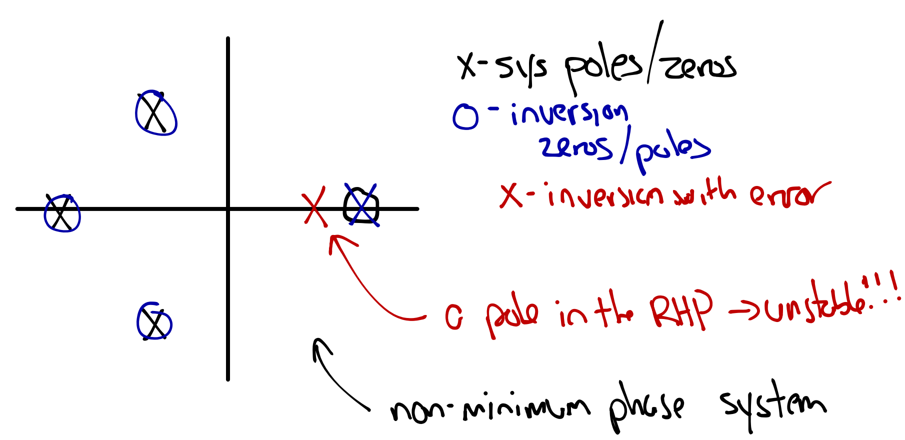
Limitations of Plant Inversion:

- Desired states must be achievable
- Model must be *very* good
- Closed loop transfer function must be in the $\underbrace{\text{*right*}}$ form

What is the right form? (or wrong)

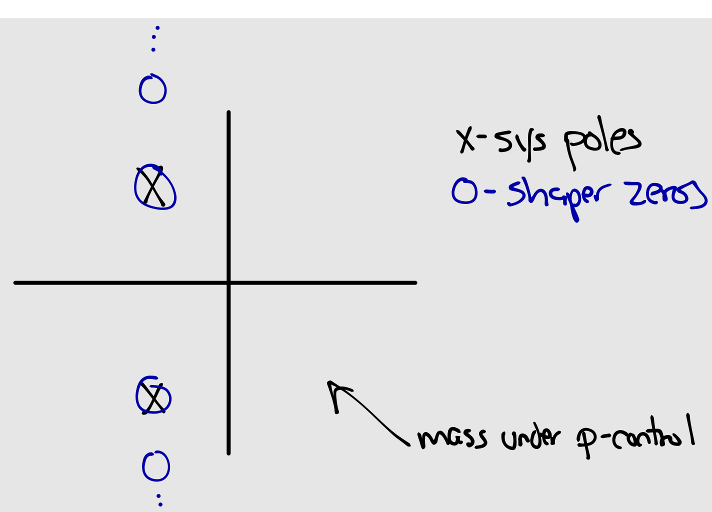
To answer, let's look at this method in the s-plane





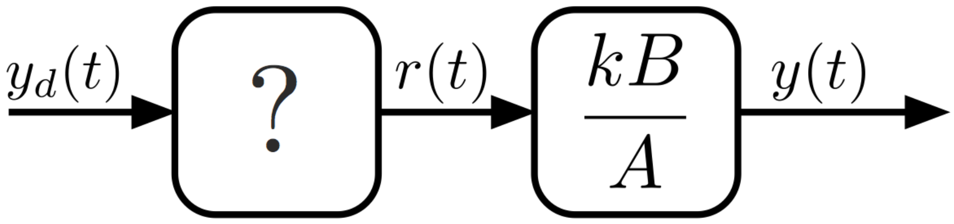
Aside:

Input shaping places a column of zeros over flexible poles.



Zero Phase Error Tracking Controller (ZEPTC)

The problem with inversion is RHP zeros, so just don't invert those



split B into parts

B_A - acceptable to invert

B_u - unacceptable for inversion

$\frac{A}{B_A B_u(1)}$ ← $B_u(1)$ is a constant, needed to scale properly

So,

$$y = \left[\frac{A}{B_A B_u(1)}\right]\left[\frac{kB}{A}\right] y_d$$

$$= \left[\frac{\cancel{A}}{\cancel{B_A} B_u(1)}\right]\left[\frac{\cancel{B_A} B_u}{\cancel{A}}\right] y_d$$

$$y = \frac{B_u}{B_u(1)} y_d$$

you can think of ZPETC as "partial" model inversion

Note: There are a few more steps for a "real" ZPETC. Below is the final form,

It is usually shown in the digital domain.

$$r(k) = \frac{A(z^{-1})\, B_u(z^{-1})}{B_A(z^{-1})\,[B_u(1)]^2}\, y_d(k+d+s)$$

d = time delay inherit to plant

s = # of unacceptable zeros

See slides for example.

Repetitive Learning Control

Repeated Trajectories are common in:

- manufacturing
- machining
- robotics
- scanning
- automated cranes (the few that are existence)

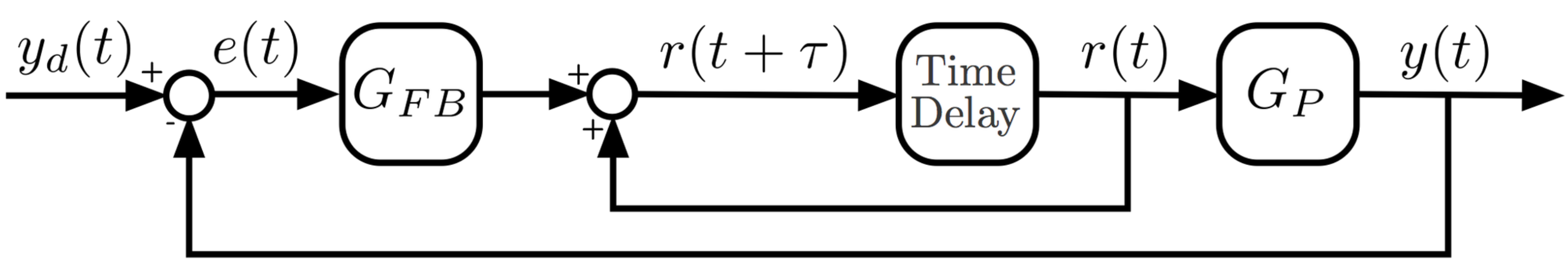
Objective: "Learn" a good control input while tracking a repeated (periodic) trajectory

Q: What is a repeated trajectory, mathematically?

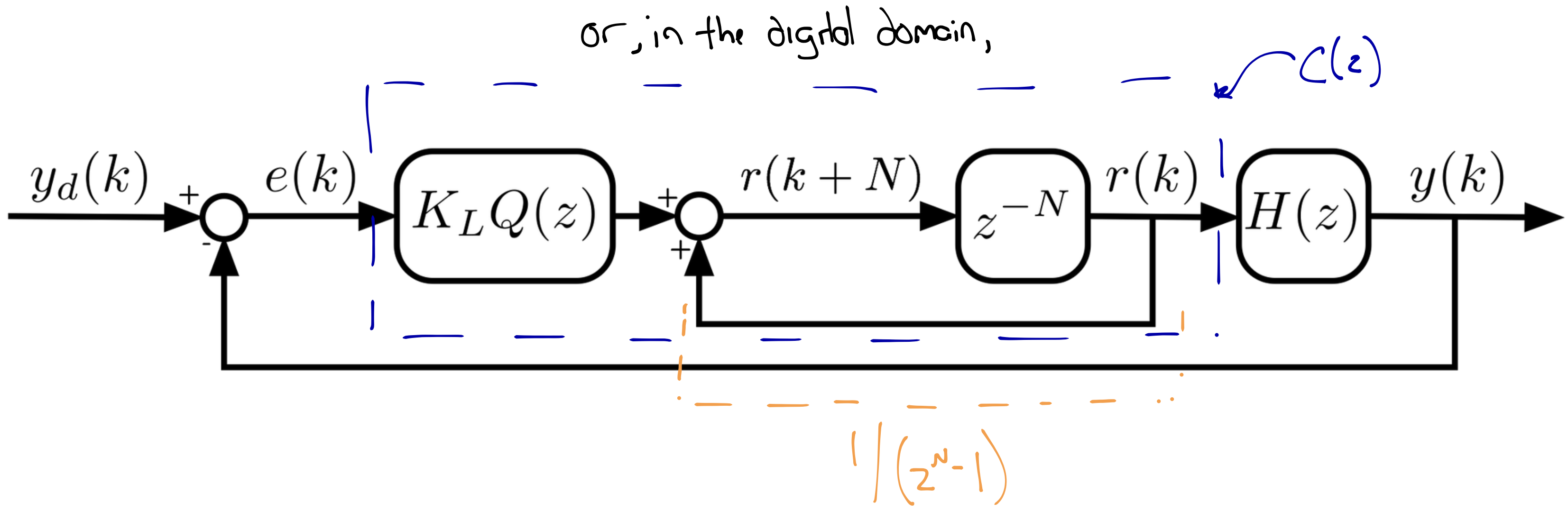
$$y_d(t+\tau) = y_d(t) \quad \text{or} \quad y_d(k) = y_d(k+N)$$

τ ← periodic of traj.

N ← # of timesteps in period of traj.



or, in the digital domain,



$$C(z) = \frac{K_L Q(z)}{z^N - 1}$$

$K_L > 0$ – learning gain

N – period of repetition (in # of samples)

$Q(z)$ – "feedback" controller

Repetitive Learning Control (cont)

Let's look at the reference input r(k):

$$\underbrace{r(k+N)}_{\text{input next cycle}} = \underbrace{r(k)}_{\text{input this cycle}} + \underbrace{K_L Q(z)\left[y_d(k) - y(k)\right]}_{\text{compensation for error during this cycle}}$$

Q: What's the effect of increasing K_L?

- more emphasis on error
- faster "learning"
- BUT more sensitive to noise