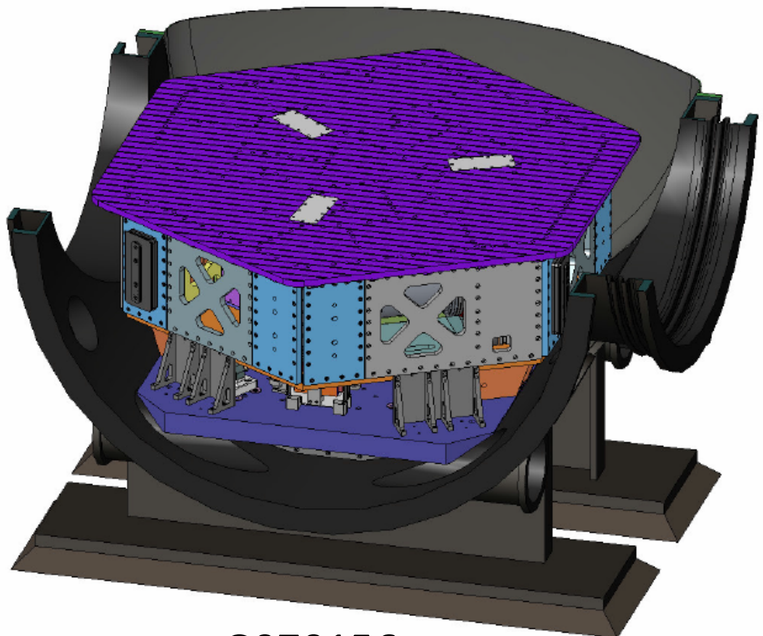
Lecture 2

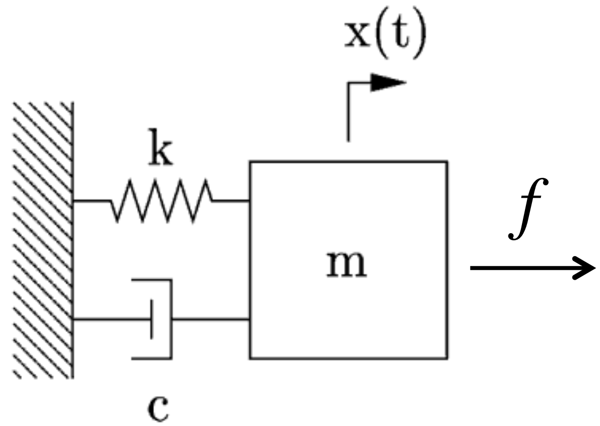
Basic control design

- Part 1: Feedforward
- Part 2: Feedback
- Part 3: Sensor blending

Example system – HAM ISI

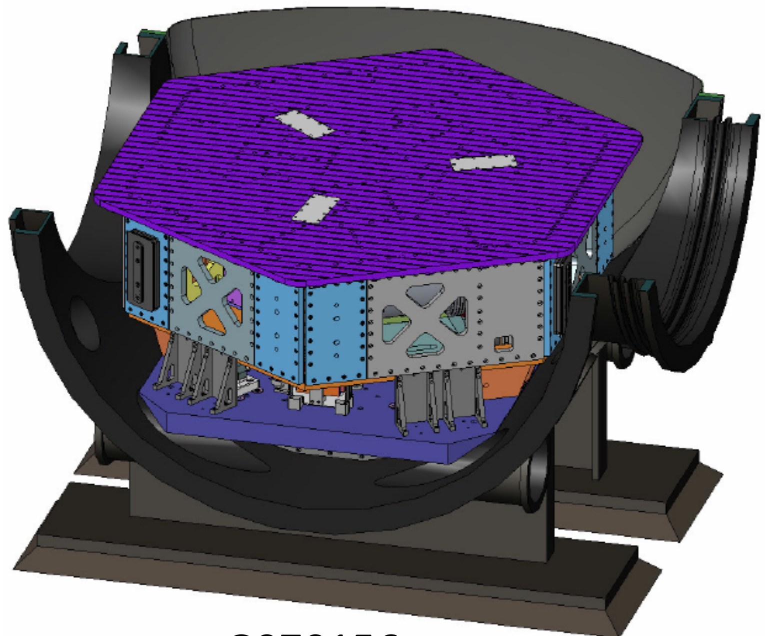


G070156

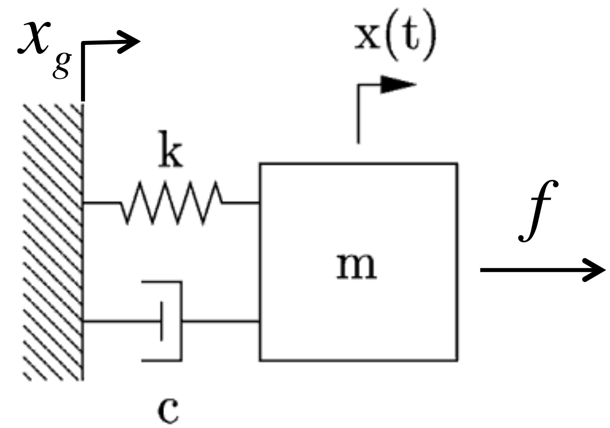


$$m\ddot{x} + c\dot{x} + kx = f$$

Example system – HAM ISI



G070156

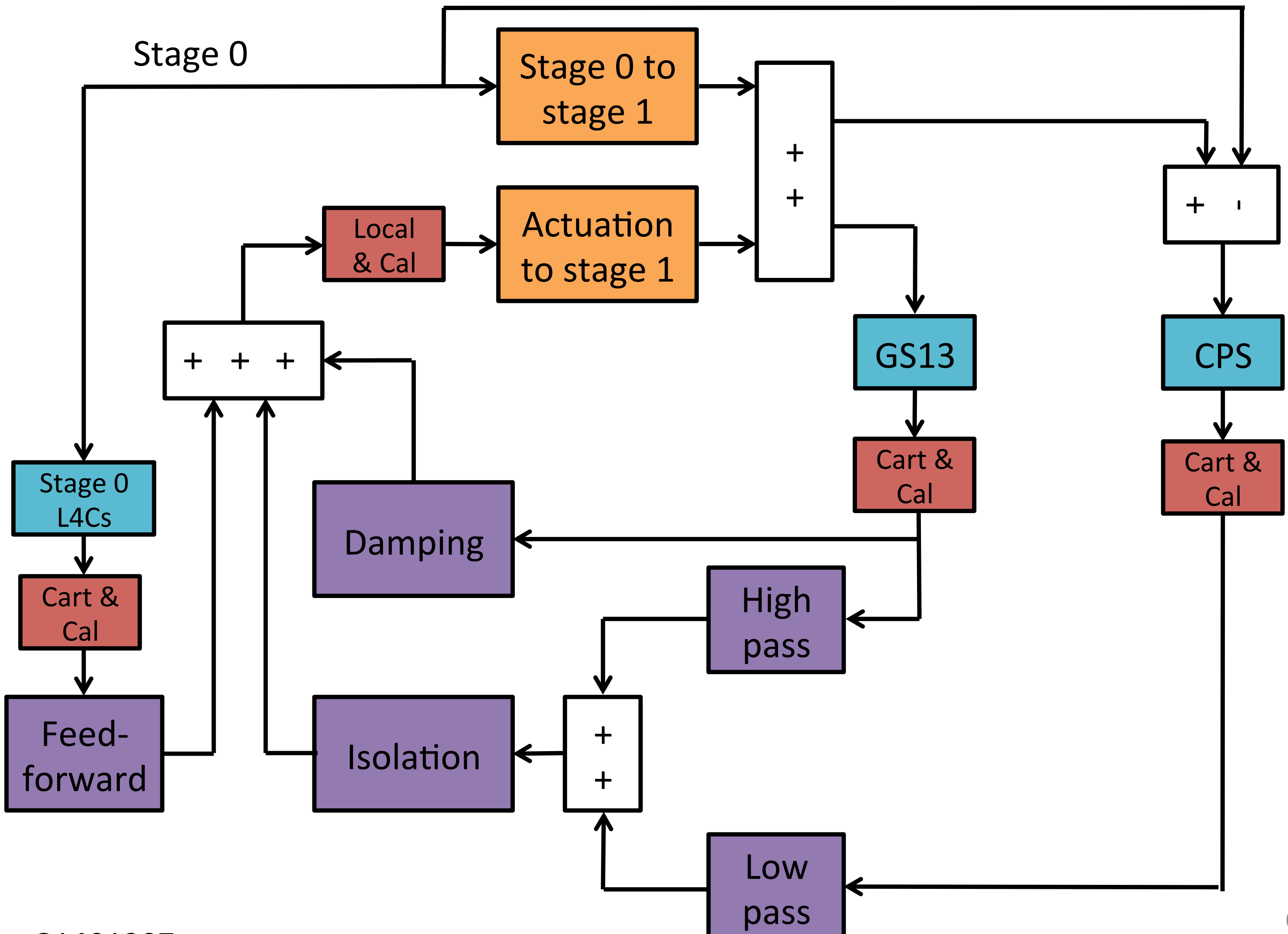


$$m\ddot{x} + c\dot{x} + kx = c\dot{x}_g + kx_g + f$$

Goals:

- Use f to reduce the influence of ground displacement, x_g, on the ISI
- Don't amplify the ISI motion with sensor noise

Stage 1 of 1-stage of in-vacuum isolation table (HAM-ISI)

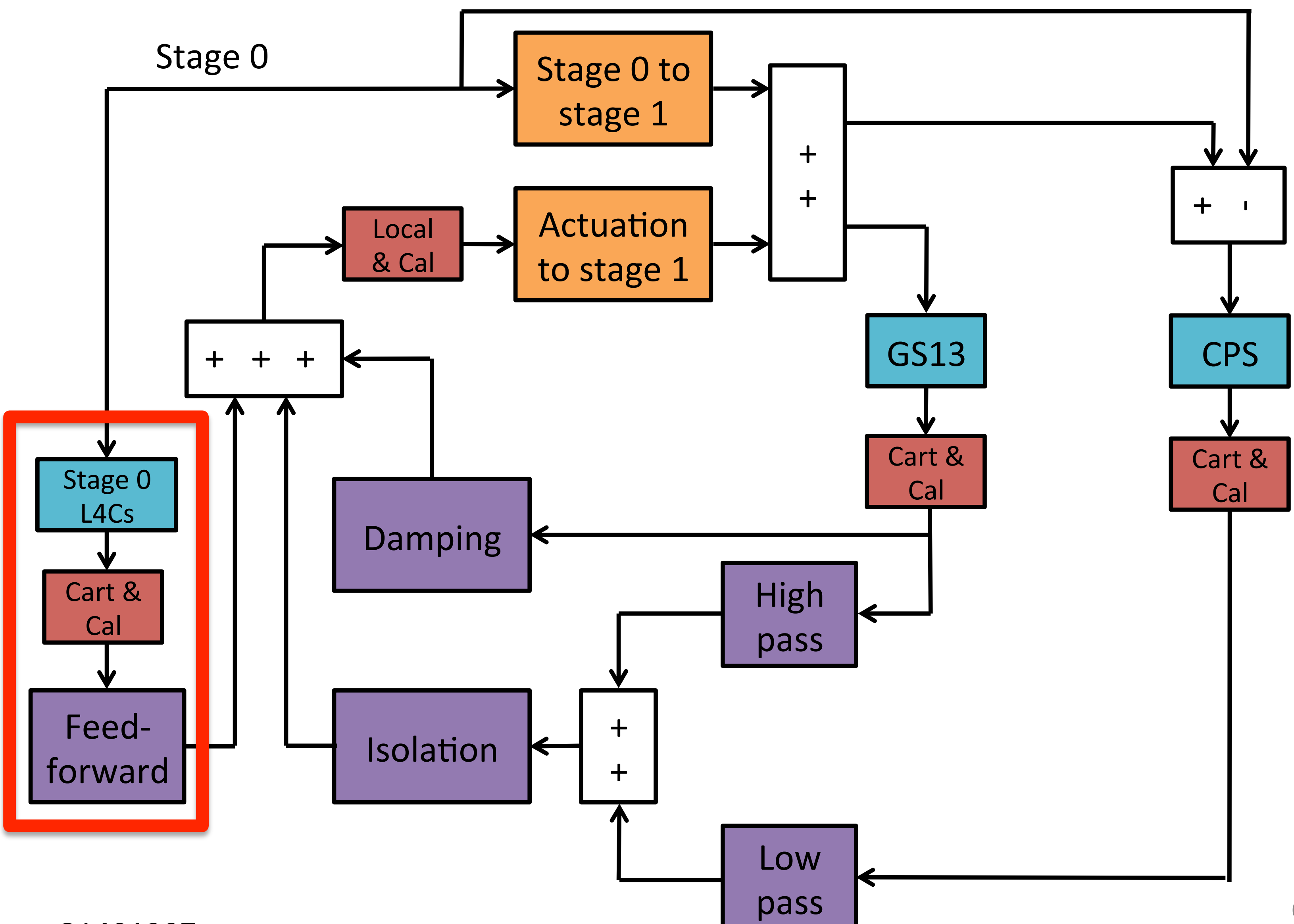


Reference G1401207

Lecture 2 – Part 1

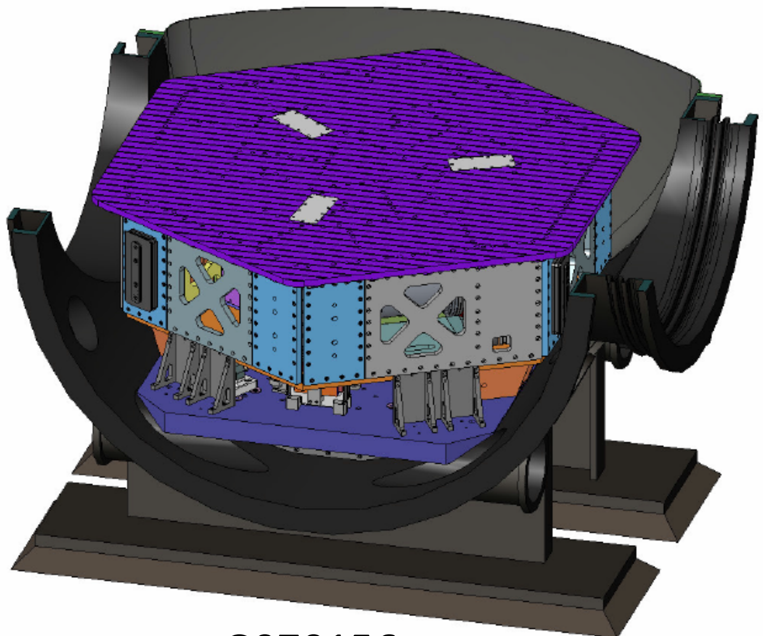
Feedforward

Stage 1 of 1-stage of in-vacuum isolation table (HAM-ISI)

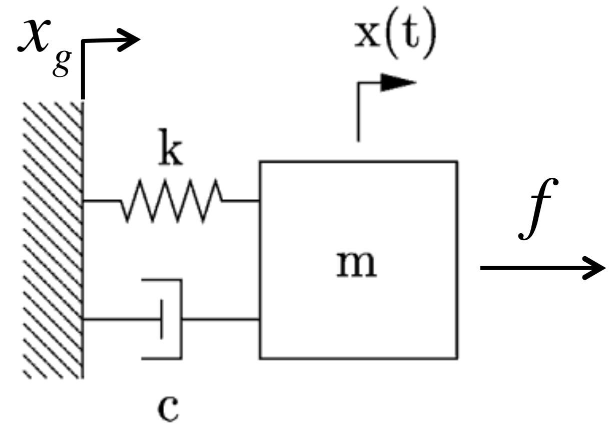


Reference G1401207

Feedfoward Control



G070156



$$m\ddot{x} + c\dot{x} + kx = c\dot{x}_g + kx_g + f$$

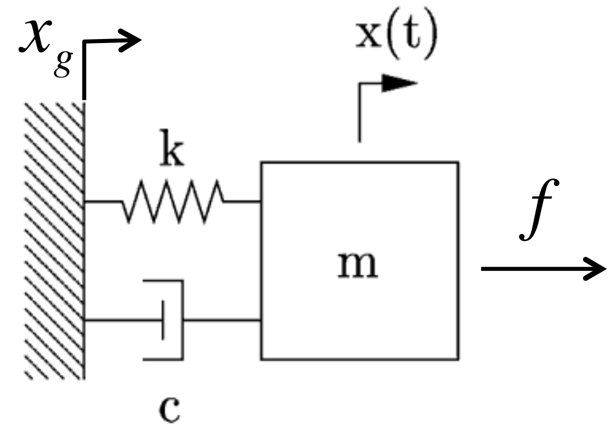
Ideal feedforward controller → $$f = -c\dot{x}_g - kx_g$$

- An inertial sensor on the ground measures x_g
- The actuator applies the correcting force f before the ISI responds
  - it's like the ground never even moved
- Performance is limited by how well the controller is tuned, and how much coherence there is between the ground sensor and the ISI sensor.
- The feedforward controller does not depend on the feedback design

Feedfoward Control

In practice, the feedforward control is achieved with the following 4 steps:

1. Measure the TF between the ground and the ISI
2. Measure the TF between the actuator and the ISI
3. Calculate the ratio of step 1 to step 2
4. Fit a filter to this TF ratio. This is the feedforward control filter.

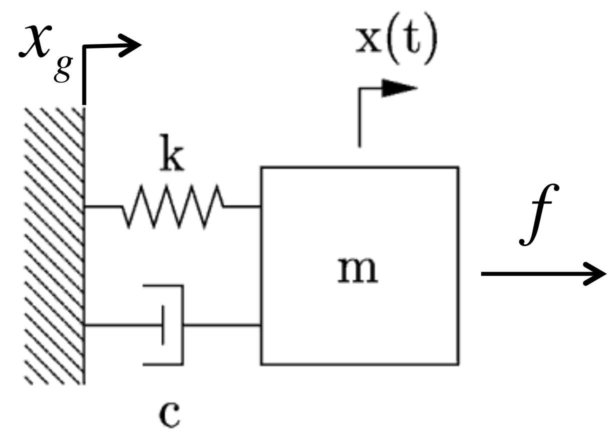


1)
$$\frac{x}{x_g} = \frac{cs + k}{ms^2 + cs + k}$$

Feedfoward Control

In practice, the feedforward control is achieved with the following 4 steps:

1. Measure the TF between the ground and the ISI
2. Measure the TF between the actuator and the ISI
3. Calculate the ratio of step 1 to step 2
4. Fit a filter to this TF ratio. This is the feedforward control filter.



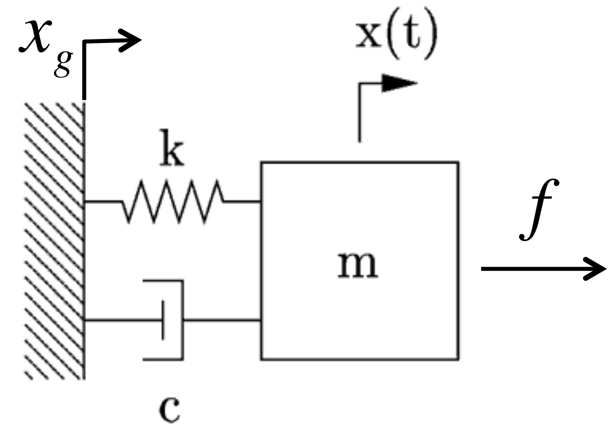
1) $$\frac{x}{x_g} = \frac{cs + k}{ms^2 + cs + k}$$

2) $$\frac{x}{f} = \frac{1}{ms^2 + cs + k}$$

Feedfoward Control

In practice, the feedforward control is achieved with the following 4 steps:

1. Measure the TF between the ground and the ISI
2. Measure the TF between the actuator and the ISI
3. Calculate the ratio of step 1 to step 2
4. Fit a filter to this TF ratio. This is the feedforward control filter.



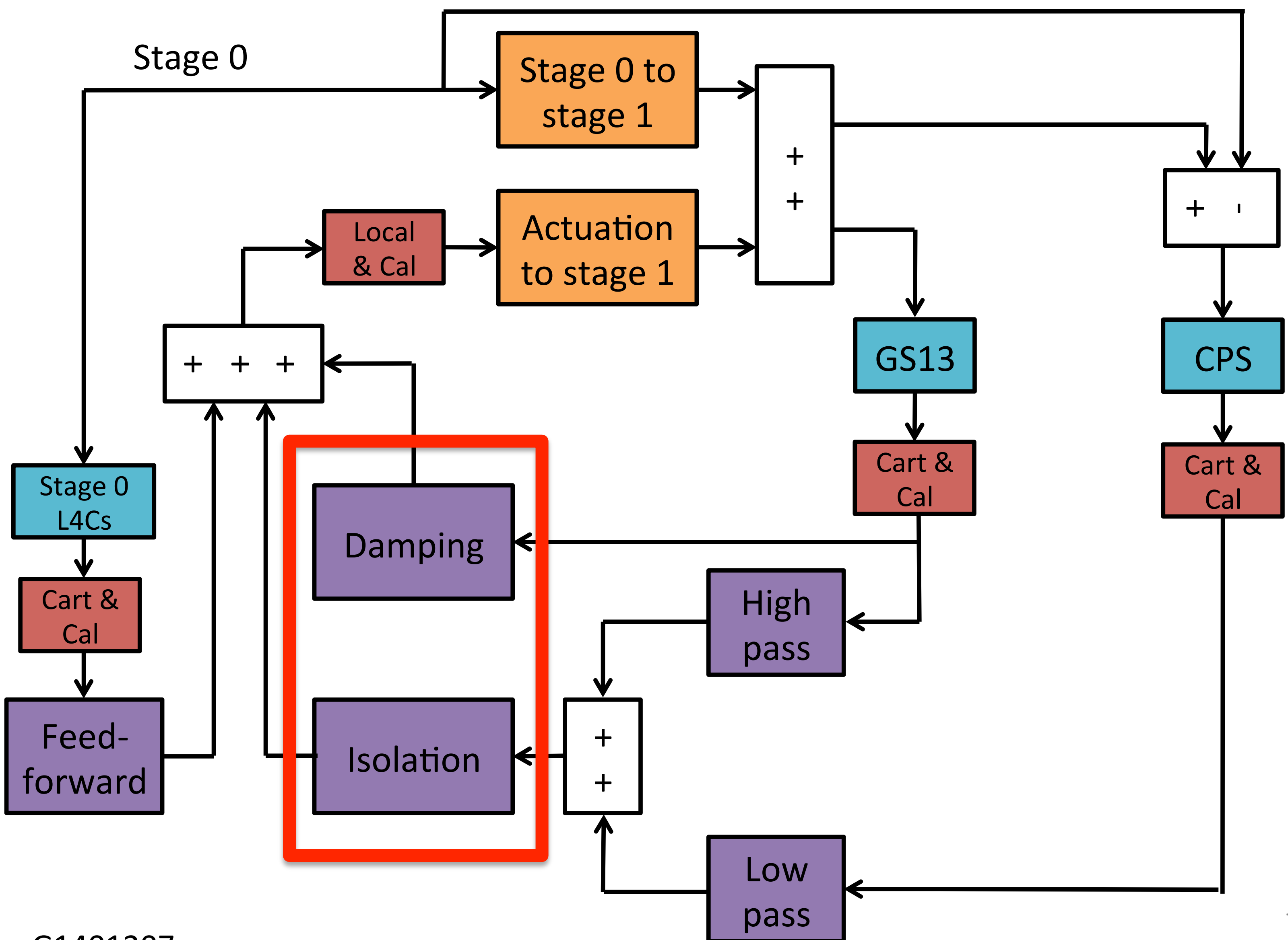
1) $$\frac{x}{x_g} = \frac{cs+k}{ms^2+cs+k}$$ $\div$ 2) $$\frac{x}{f} = \frac{1}{ms^2+cs+k}$$

$=$ 3) $$\frac{f}{x_g} = cs+k$$

Lecture 2 – Part 2

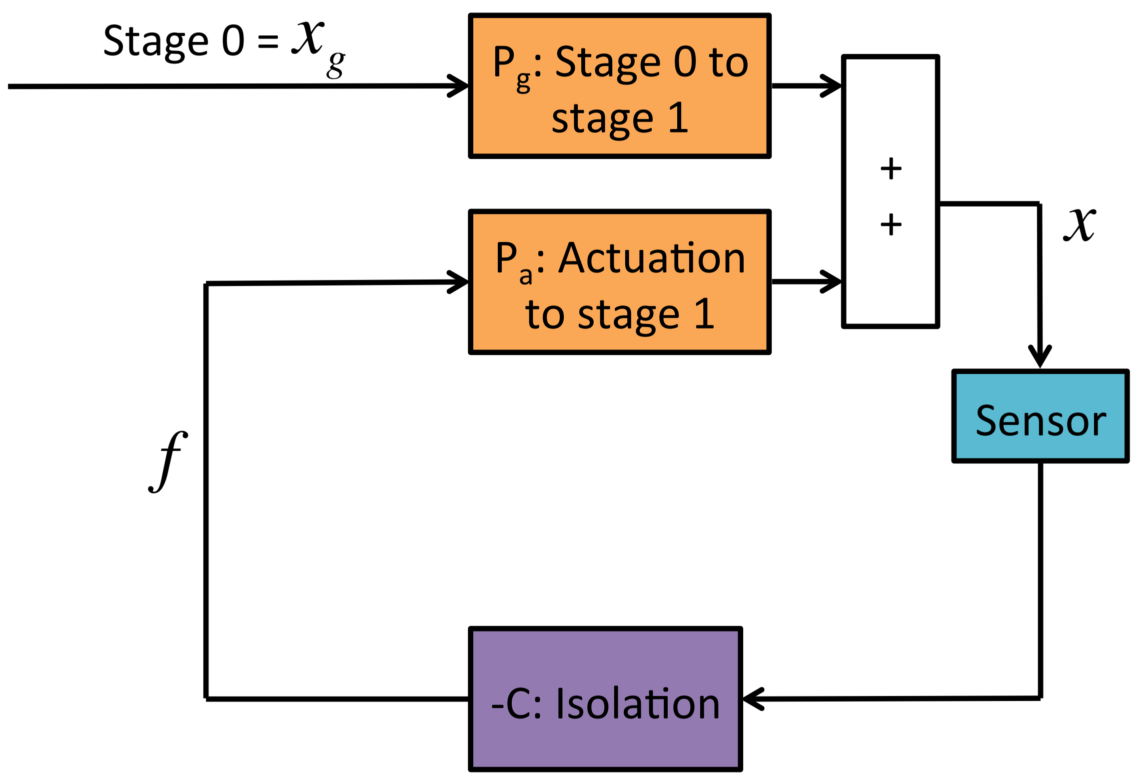
Feedback

Stage 1 of 1-stage of in-vacuum isolation table (HAM-ISI)

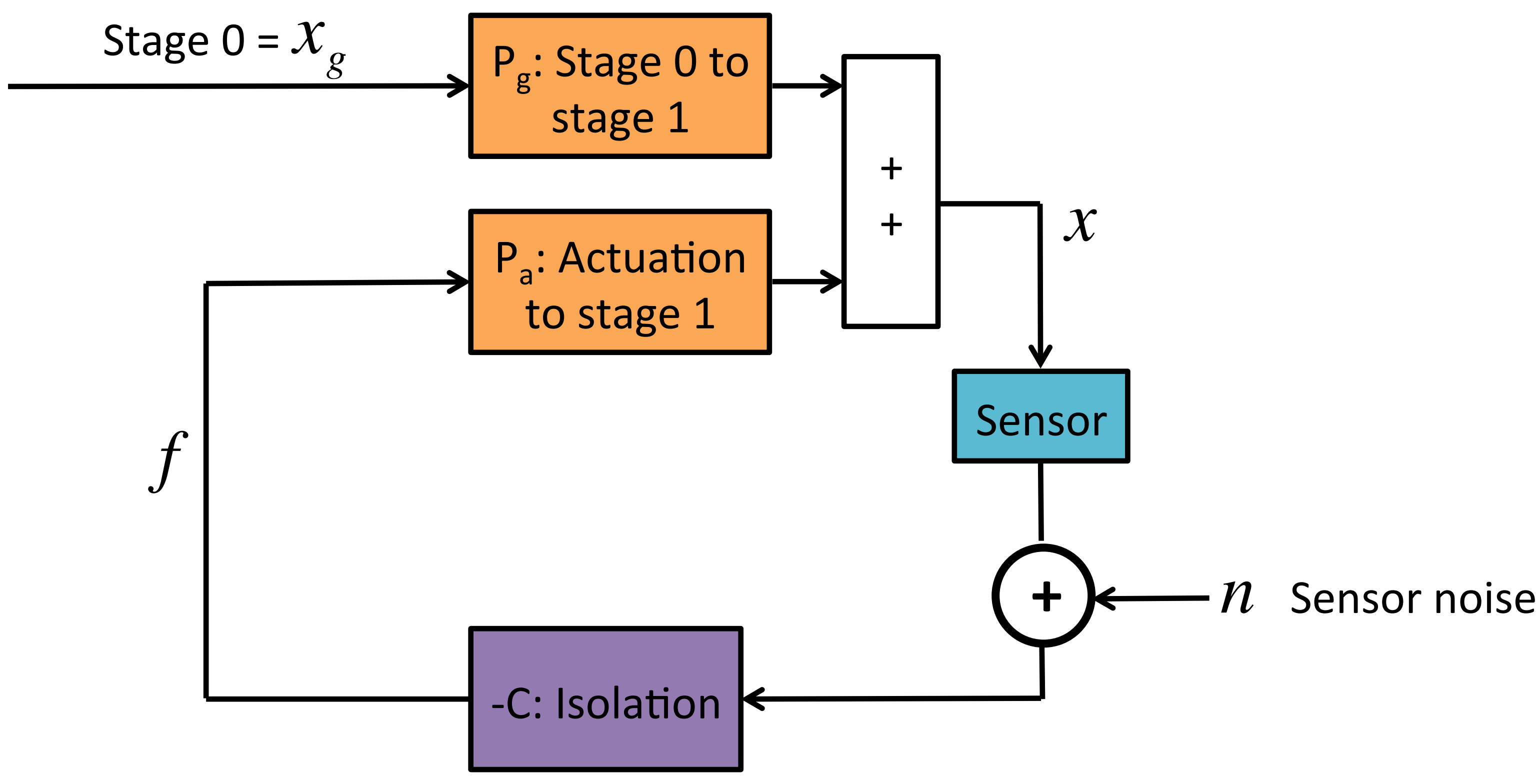


Reference G1401207

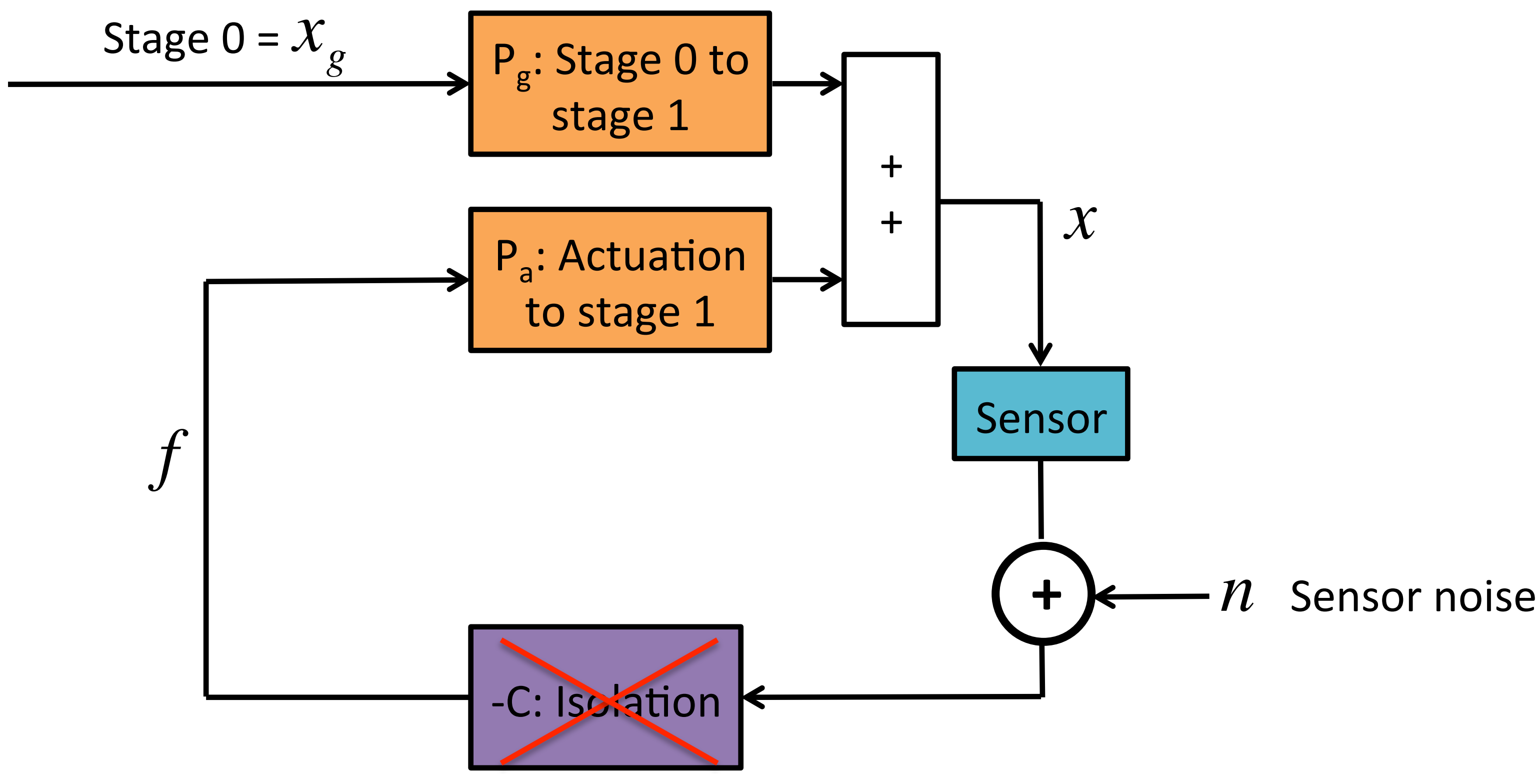
Stage 1 of 1-stage of in-vacuum isolation table (HAM-ISI)



Stage 1 of 1-stage of in-vacuum isolation table (HAM-ISI)



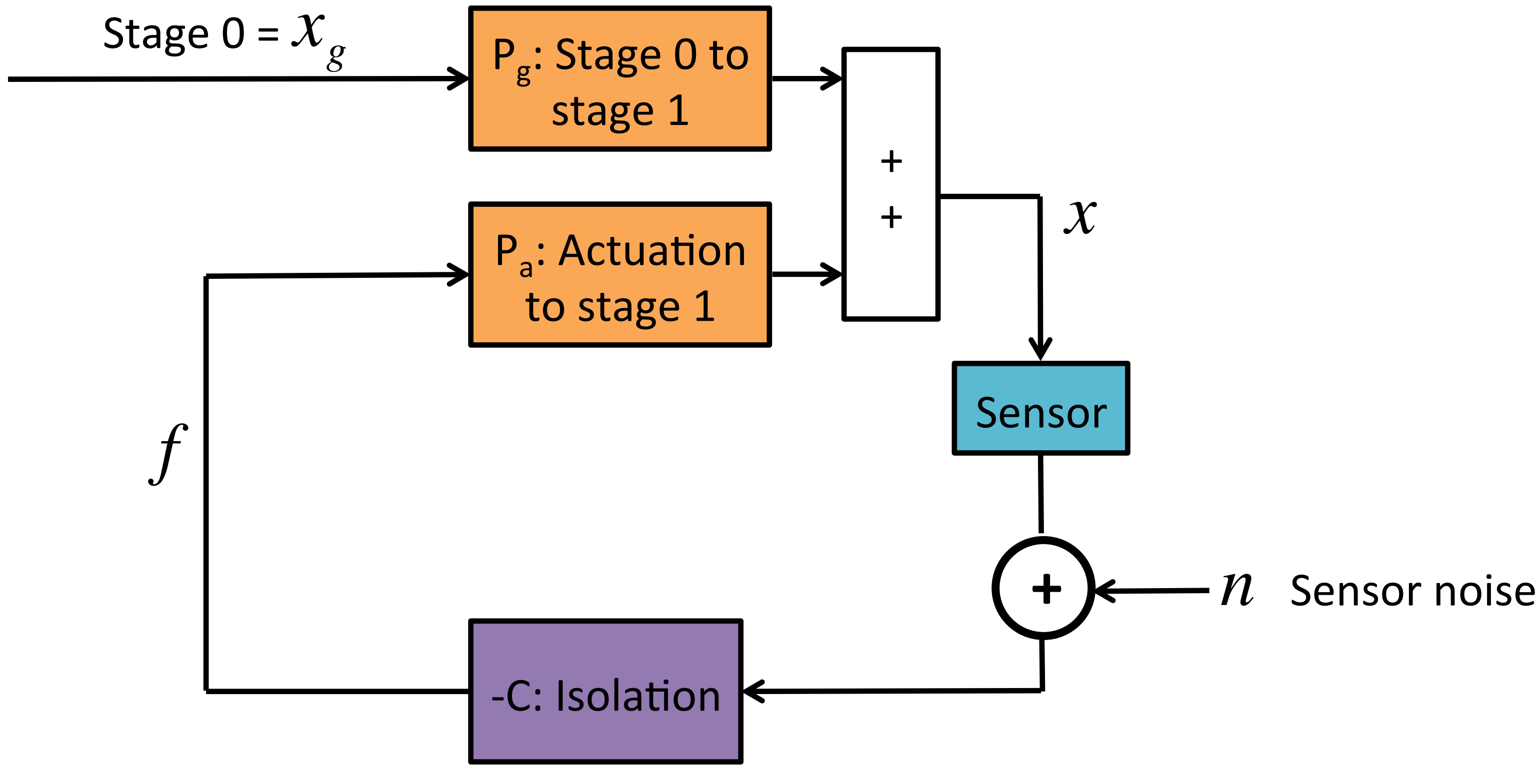
Stage 1 of 1-stage of in-vacuum isolation table (HAM-ISI)



- Uncontrolled TF from ground to stage 1

  $x = P_g x_g$

Stage 1 of 1-stage of in-vacuum isolation table (HAM-ISI)

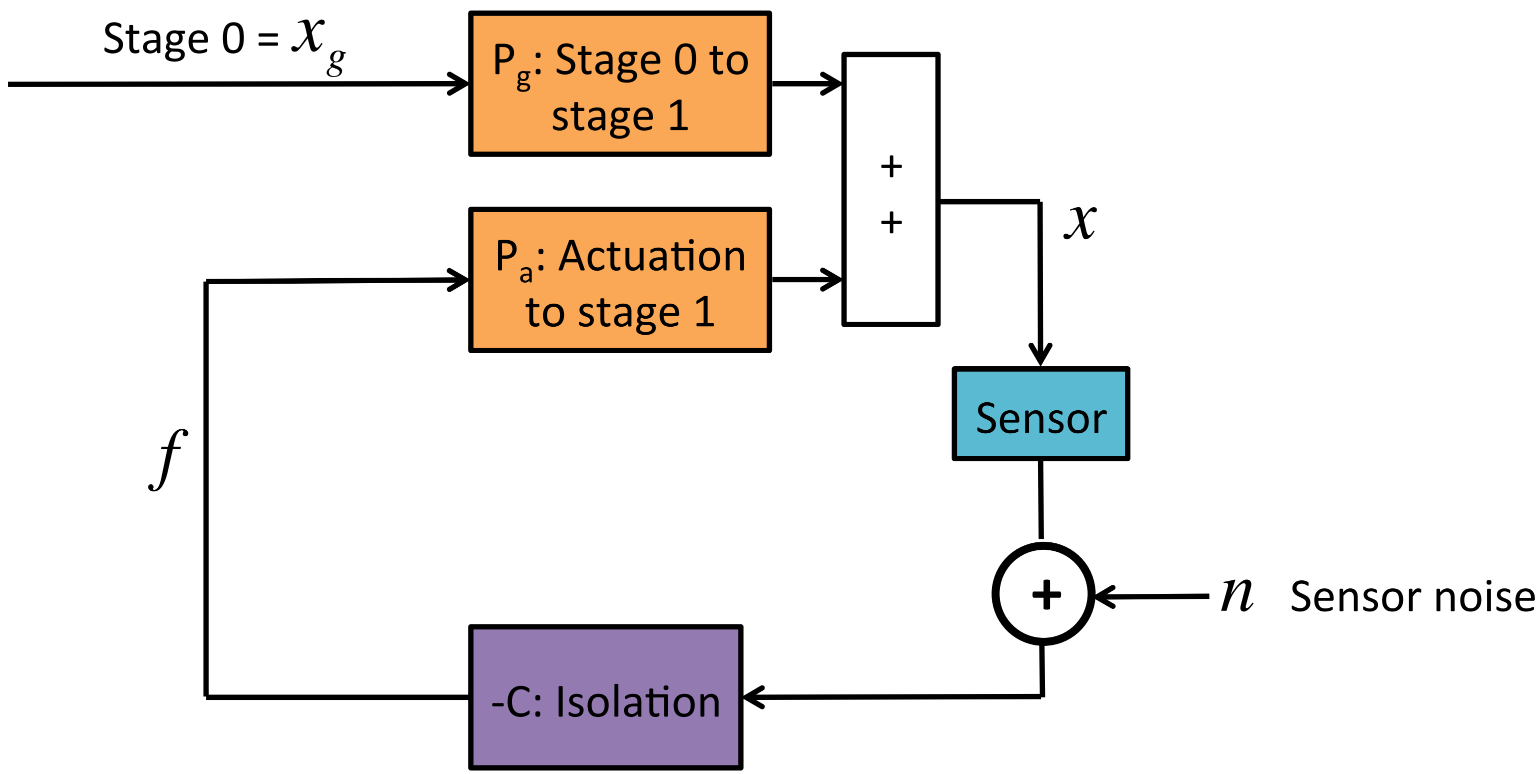


Close loop TF from ground to stage 1

$$x = P_g x_g - P_a C x$$

(Ignoring the sensor response for now)

Stage 1 of 1-stage of in-vacuum isolation table (HAM-ISI)



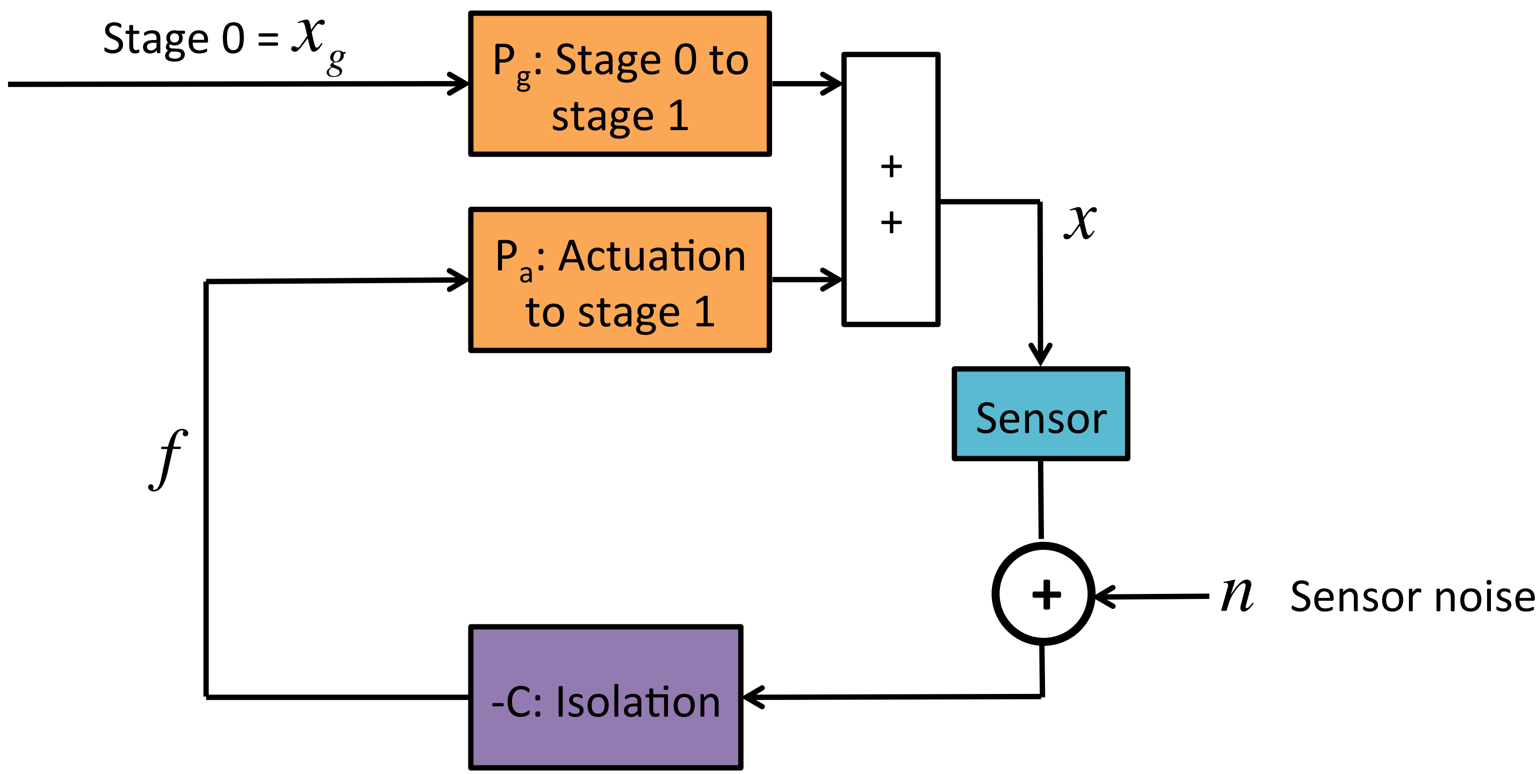
Close loop TF from ground to stage 1

$$x = P_g x_g - P_a C x$$

(Ignoring the sensor response for now)

$$\boxed{x = \frac{P_g}{1 + P_a C} x_g}$$

Stage 1 of 1-stage of in-vacuum isolation table (HAM-ISI)



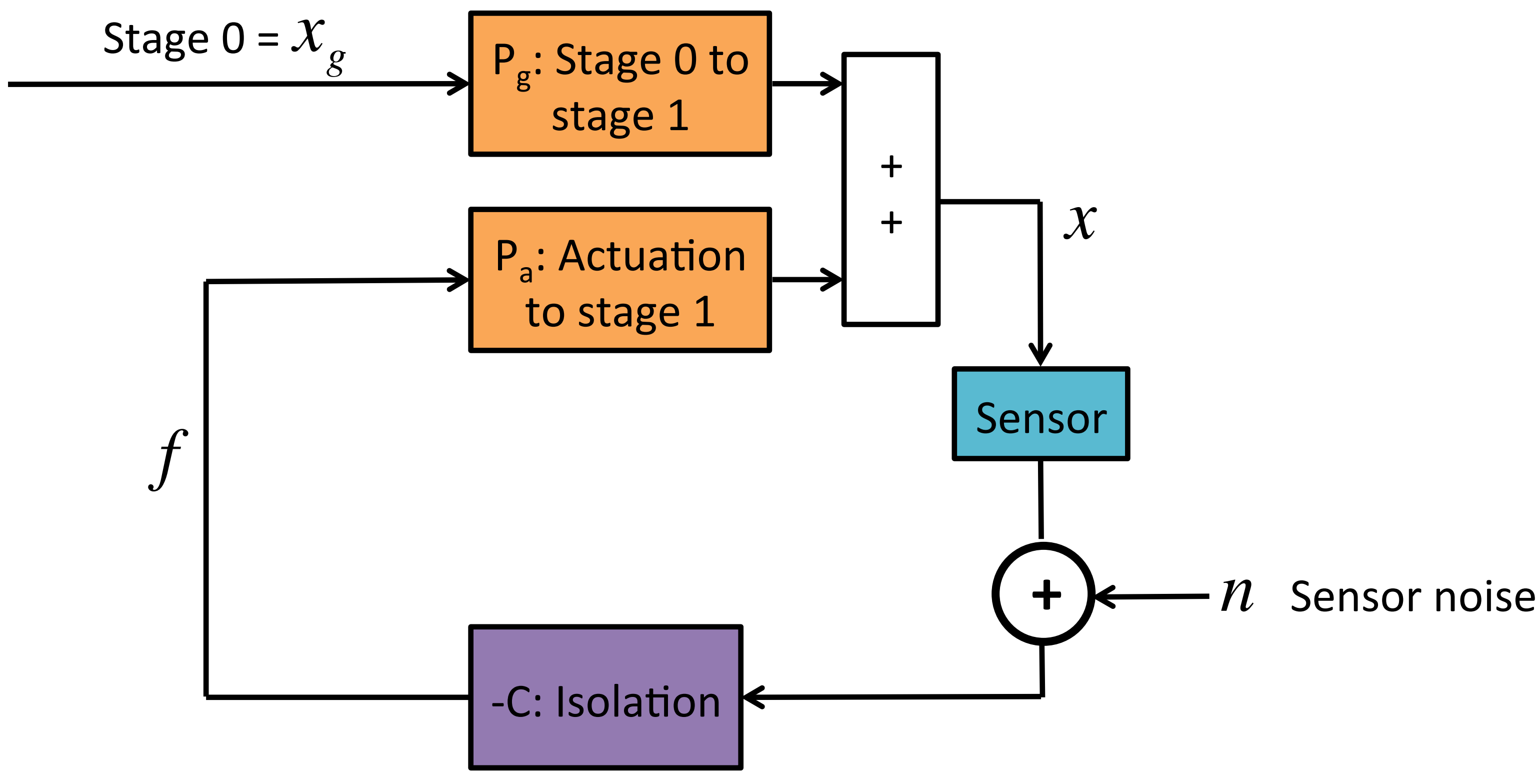
Close loop TF from sensor noise to stage 1

$$x = -P_a C(n + x)$$

(Ignoring the sensor response for now)

$$\boxed{x = \frac{-P_a C}{1 + P_a C} n}$$

Stage 1 of 1-stage of in-vacuum isolation table (HAM-ISI)



Close loop TF from sensor noise to stage 1

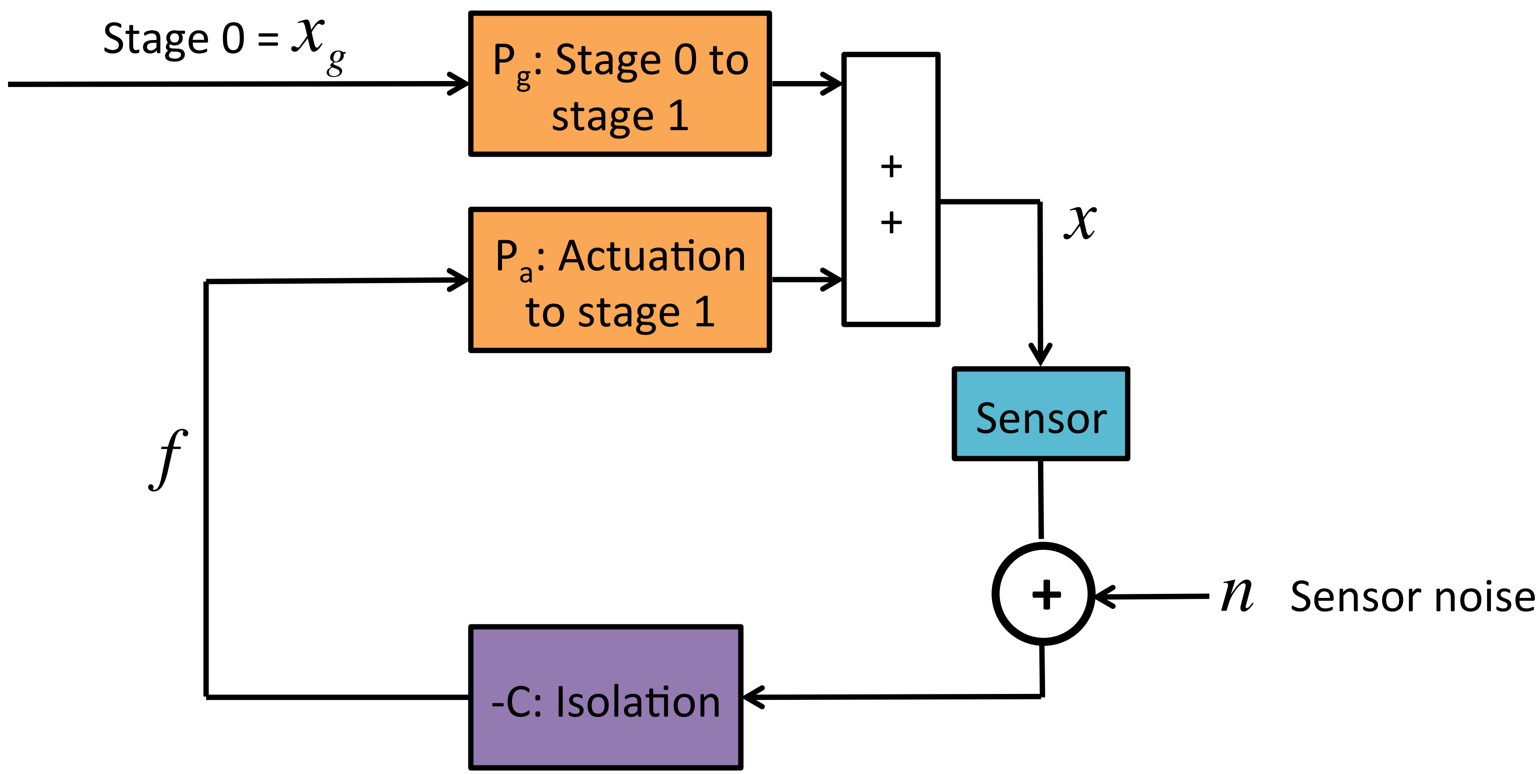
$$x = -P_a C(n + x)$$

(Ignoring the sensor response for now)

$$x = \frac{-P_a C}{1 + P_a C} n$$

All closed loop TFs in this loop will have the same denominator

Stage 1 of 1-stage of in-vacuum isolation table (HAM-ISI)



Close loop TF from sensor noise to stage 1

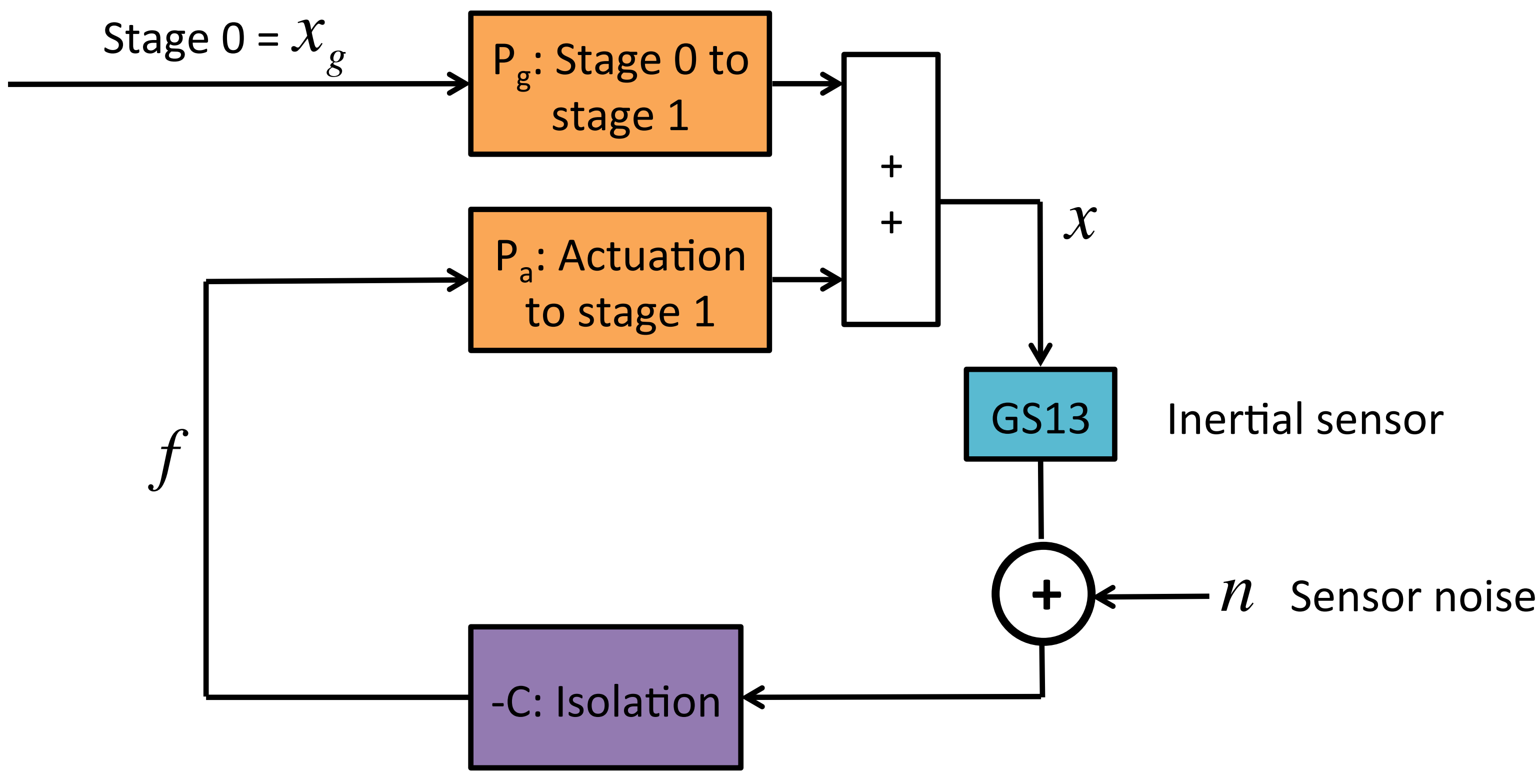
$$x = -P_a C (n + x)$$

(Ignoring the sensor response for now)

$$x = \frac{-P_a C}{1 + P_a C} n$$

Loop gain TF: important for studying stability

Stage 1 of 1-stage of in-vacuum isolation table (HAM-ISI)



Close loop TF from sensor noise to stage 1

$$x = -P_a C(n + x)$$ (Ignoring the sensor response for now)

$$x = \frac{-P_a C}{1 + P_a C} n$$

Numerator: boxes between input and output

Loop gain TF: important for studying stability

Closed Loop TFs

$$x = \frac{P_g}{1 + P_a C} x_g$$

Seismic noise transmission

$$x = \frac{-P_a C}{1 + P_a C} n$$

Sensor noise transmission

Closed Loop TFs

$$x = \frac{P_g}{1 + P_a C} x_g$$

Seismic noise transmission

$$x = \frac{-P_a C}{1 + P_a C} n$$

Sensor noise transmission

- When the loop gain is > 1, seismic noise is reduced, but the system tends to follow the sensor noise

Closed Loop TFs

$$x = \frac{P_g}{1 + P_a C} x_g$$

Seismic noise transmission

$$x = \frac{-P_a C}{1 + P_a C} n$$

Sensor noise transmission

- When the loop gain is > 1, seismic noise is reduced, but the system tends to follow the sensor noise
- If the loop gain -> -1, the system goes unstable

Closed Loop TFs

$$x = \frac{P_g}{1 + P_a C} x_g$$

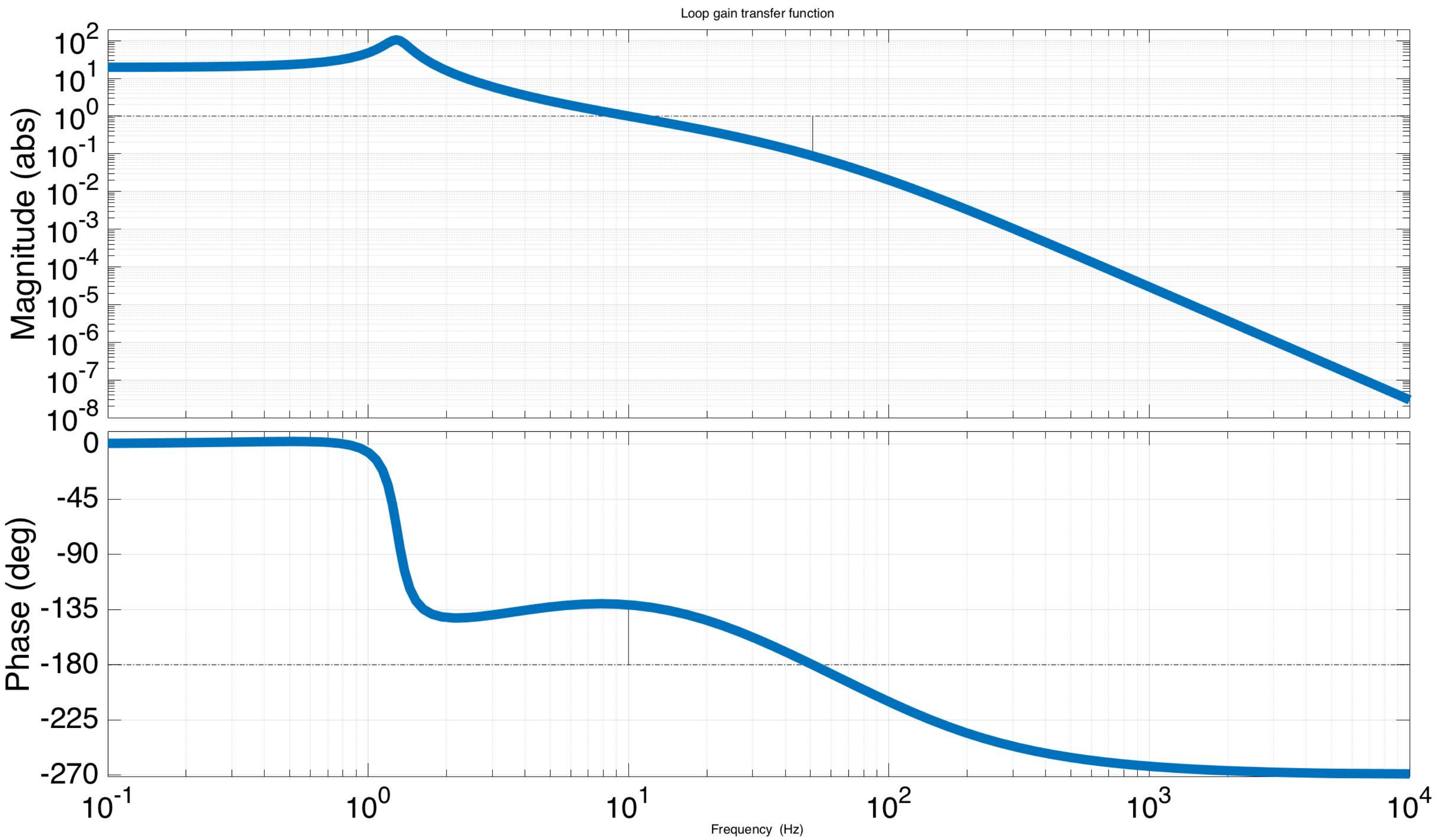
Seismic noise transmission

$$x = \frac{-P_a C}{1 + P_a C} n$$

Sensor noise transmission

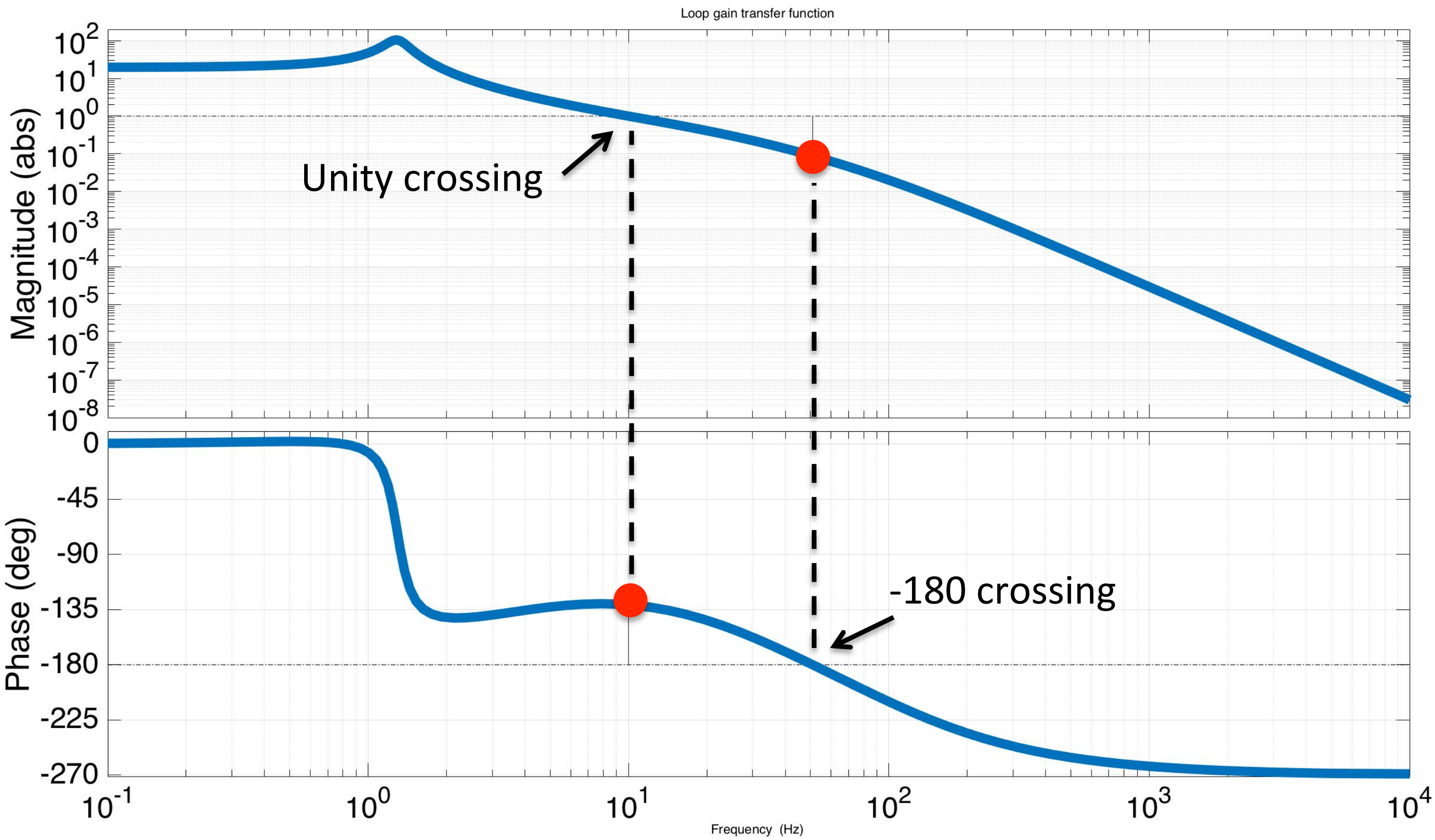
- When the loop gain is > 1, seismic noise is reduced, but the system tends to follow the sensor noise
- If the loop gain -> -1, the system goes unstable
- To study stability, just look at the loop gain

Ex. Loop Gain TF: $P_a C$



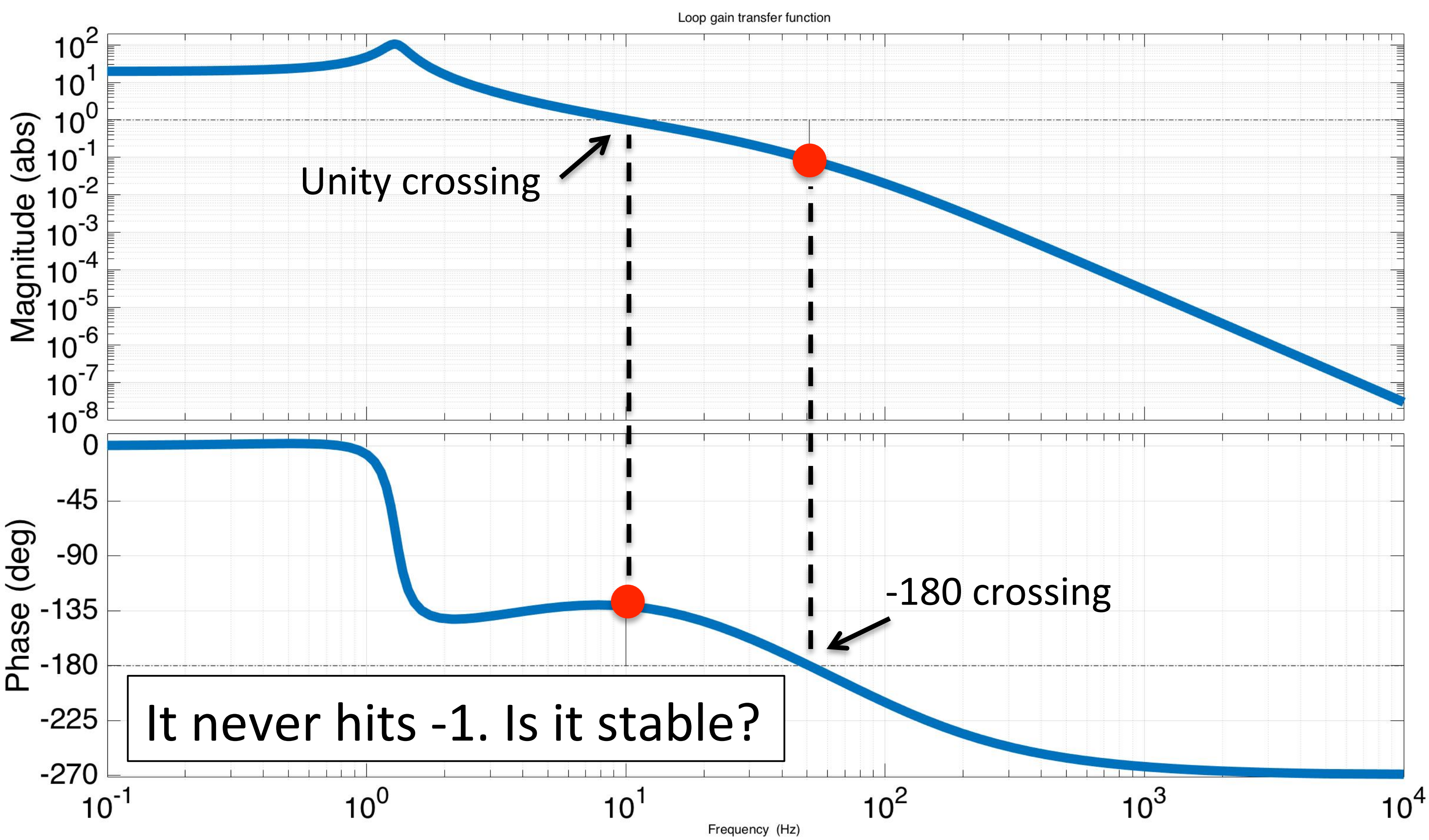
Matlab for control filter: `C = zpk(-2*pi*3.33,-2*pi*[30;100],1.4e+10)`

Ex. Loop Gain TF: $P_a C$



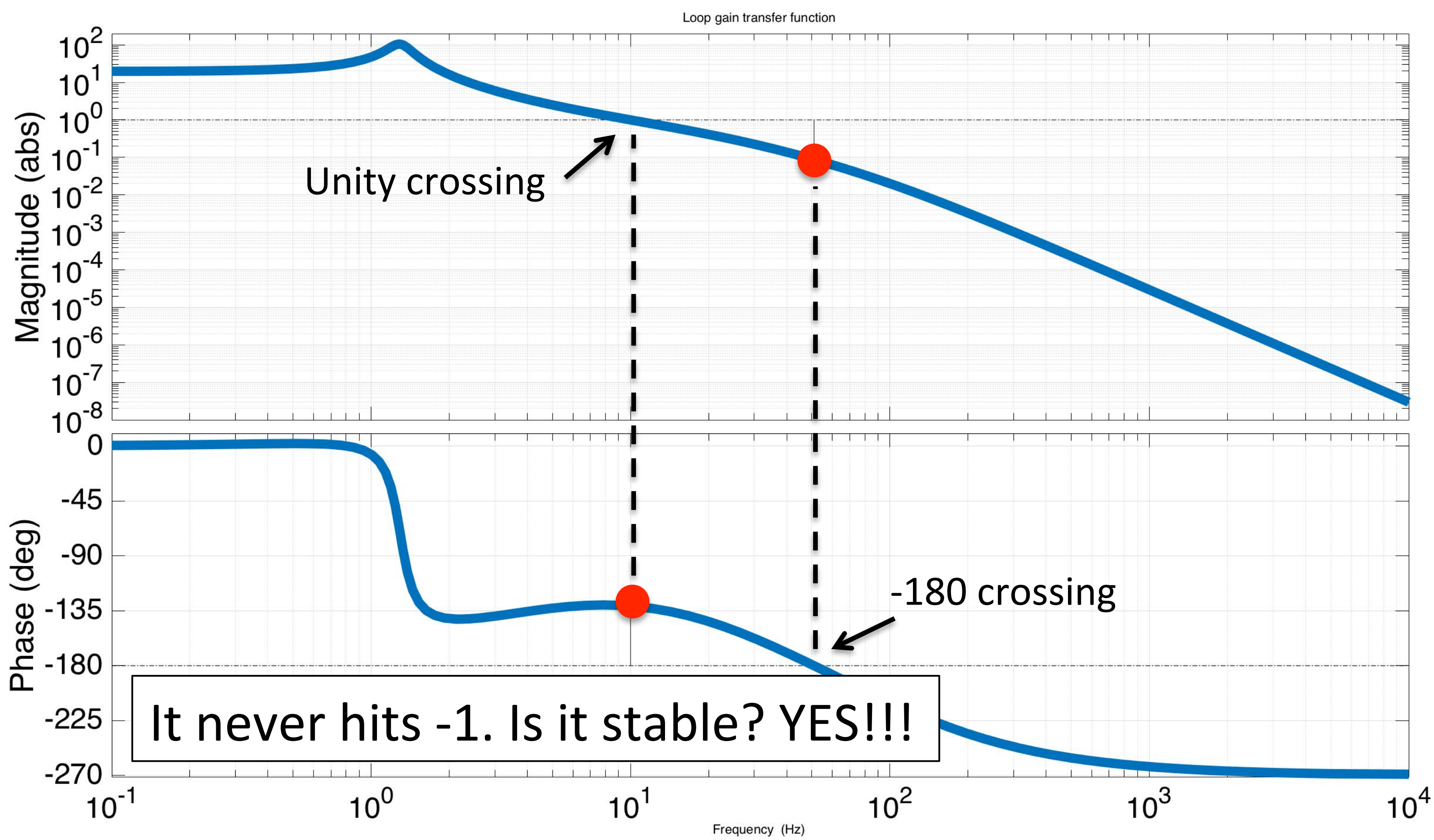
Matlab for control filter: `C = zpk(-2*pi*3.33,-2*pi*[30;100],1.4e+10)`

Ex. Loop Gain TF: $P_a C$



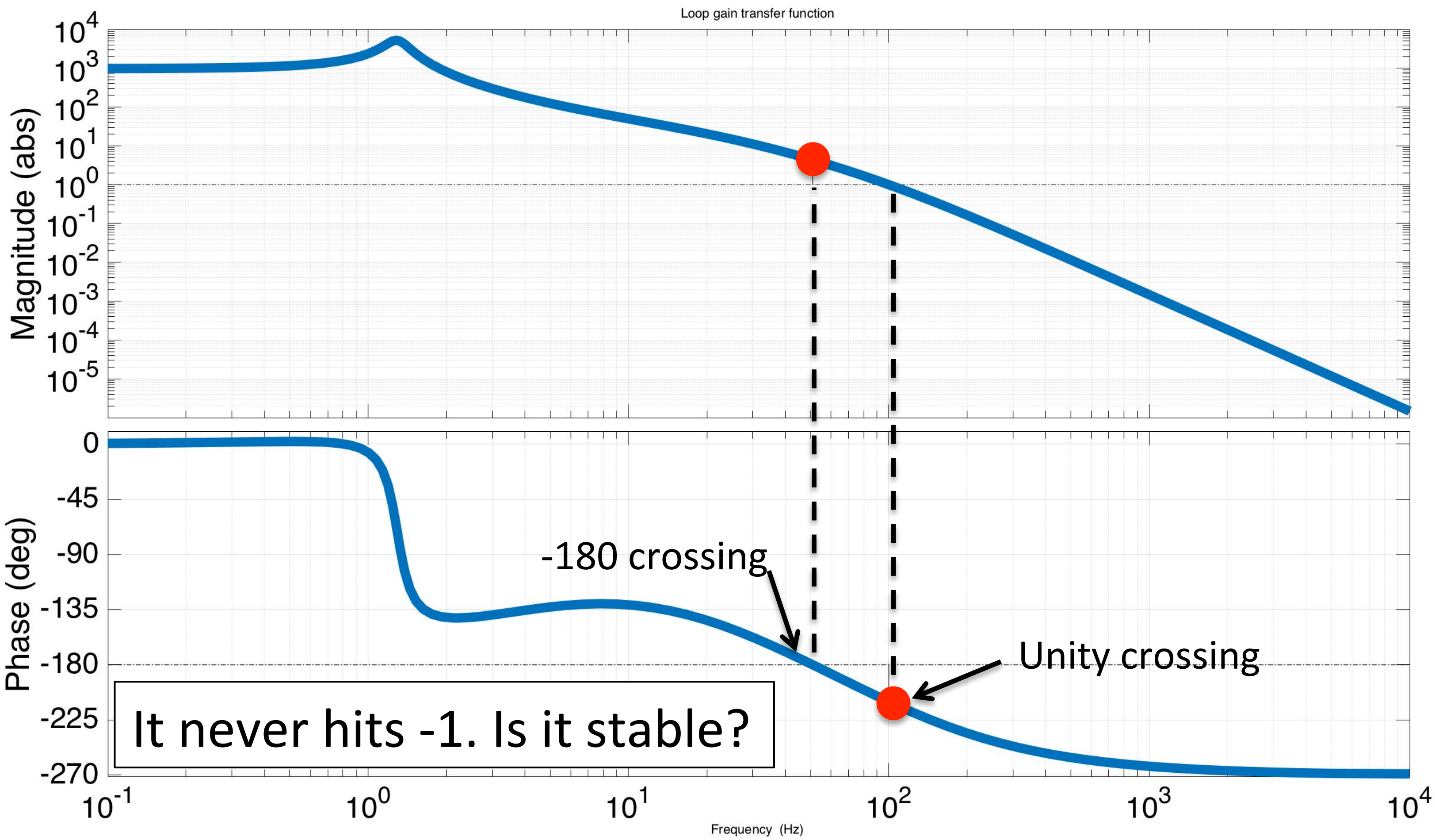
Matlab for control filter: `C = zpk(-2*pi*3.33,-2*pi*[30;100],1.4e+10)`

Ex. Loop Gain TF: $P_a C$



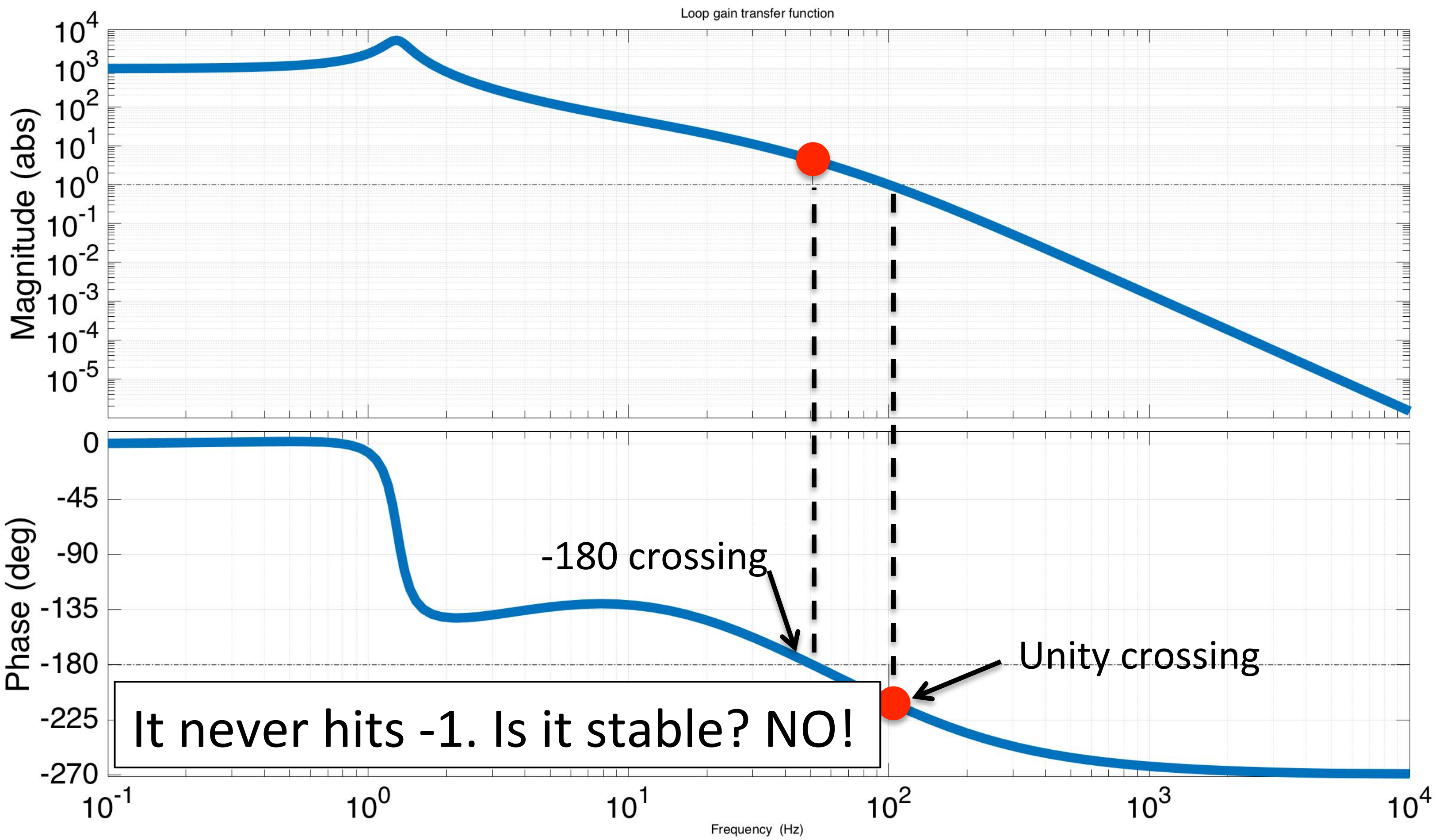
Matlab for control filter: `C = zpk(-2*pi*3.33,-2*pi*[30;100],1.4e+10)`

Ex. Loop Gain TF: $50 * P_a C$



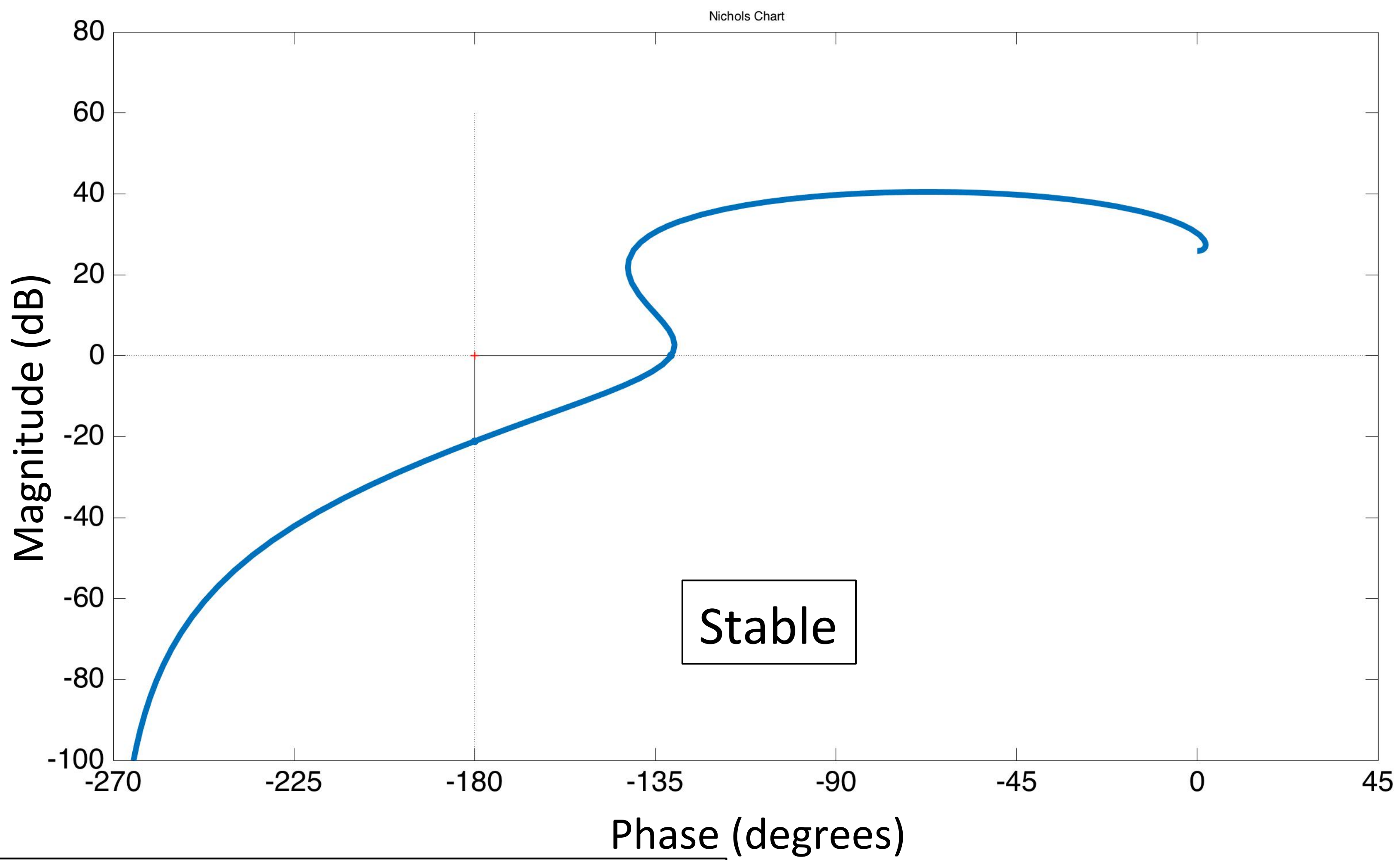
Matlab for control filter: `C = zpk(-2*pi*3.33,-2*pi*[30;100],7.0e+11)`

Ex. Loop Gain TF: $50 * P_a C$



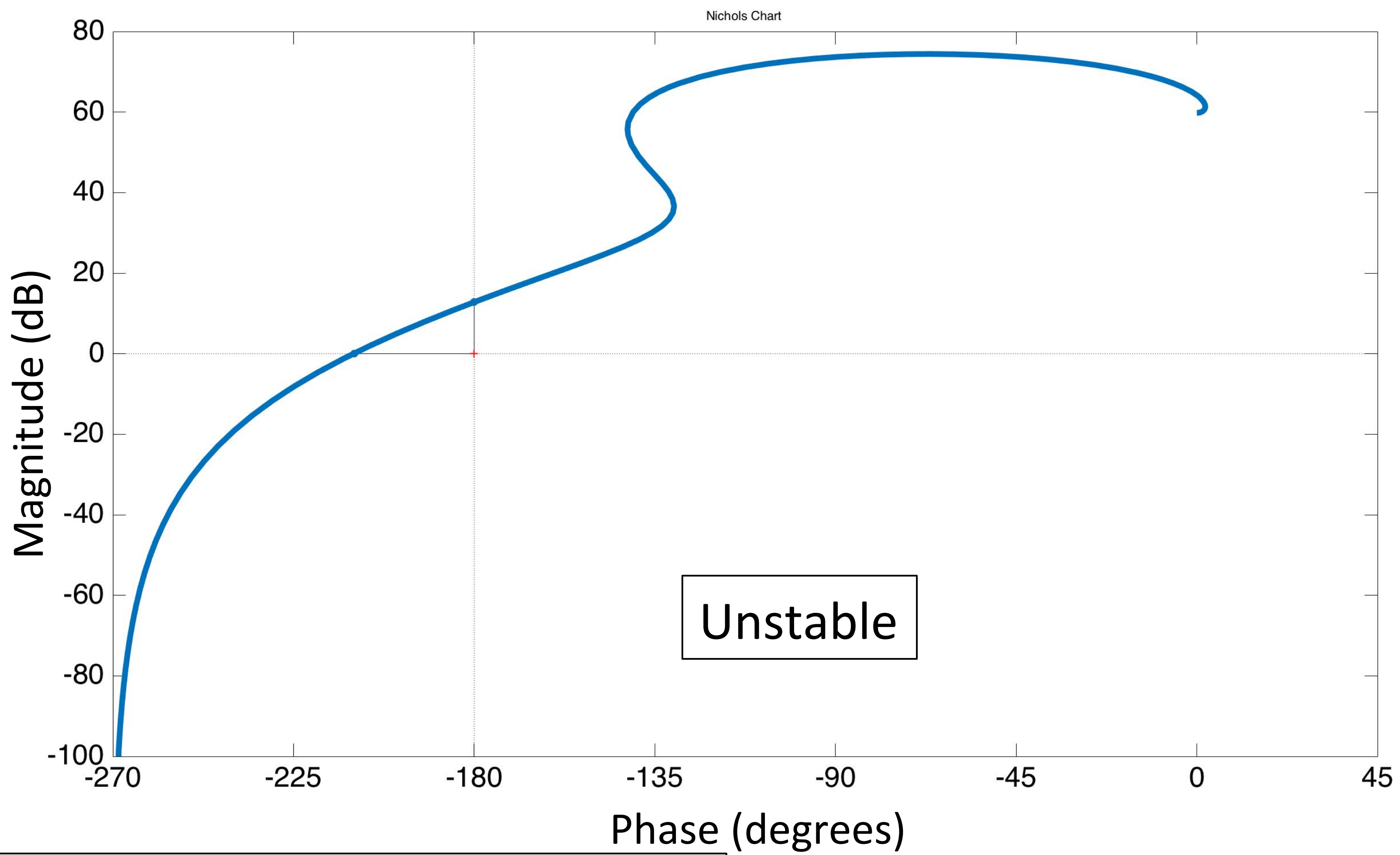
Matlab for control filter: `C = zpk(-2*pi*3.33,-2*pi*[30;100],7.0e+11)`

Loop Gain Nichols Plot: $P_a C$



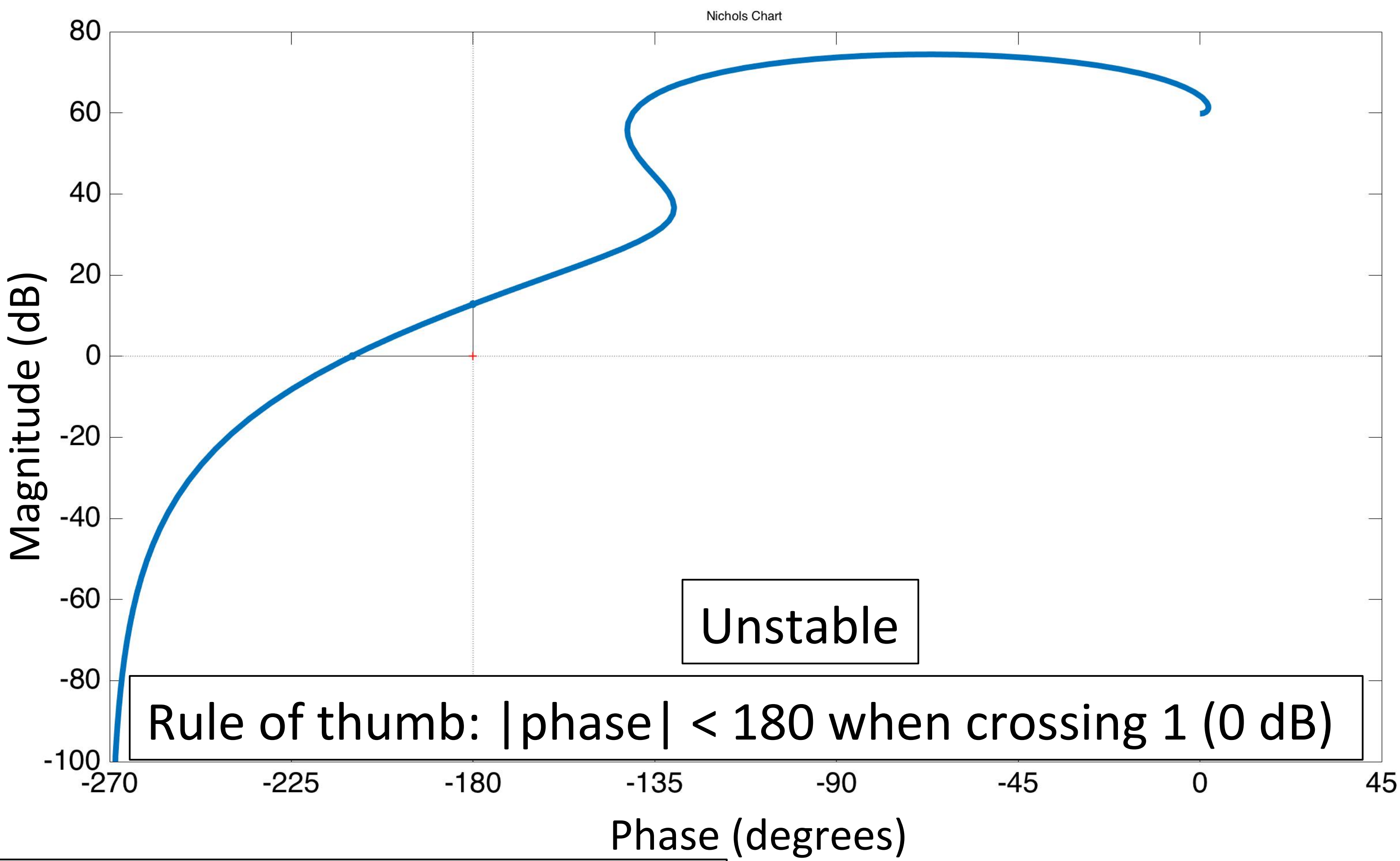
Matlab code: `nichols(Pa*C),grid off`

LIGO Loop Gain Nichols Plot: 50*$P_a C$



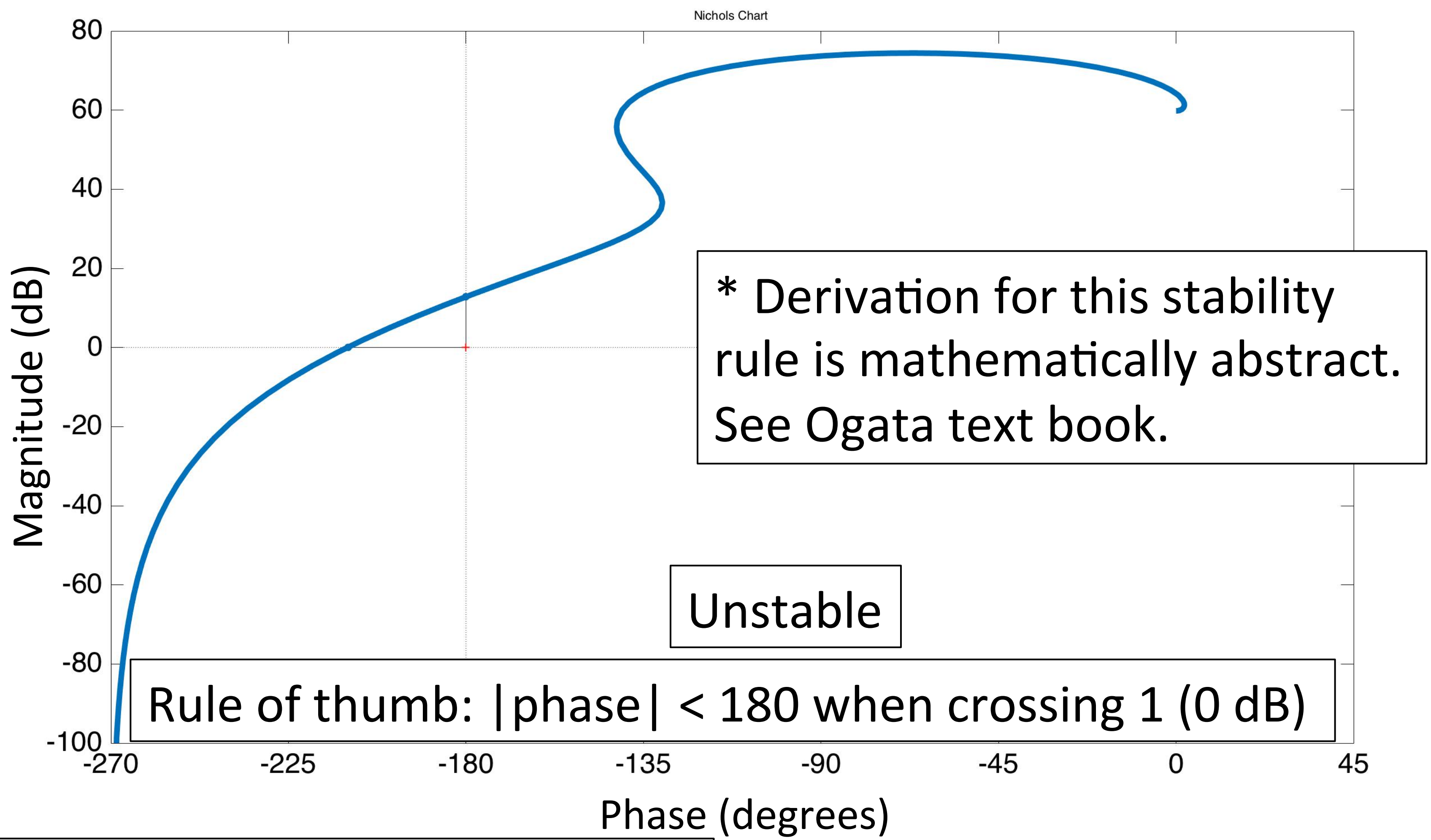
Matlab code: `nichols(50*Pa*C),grid off`

Loop Gain Nichols Plot: 50*$P_a C$



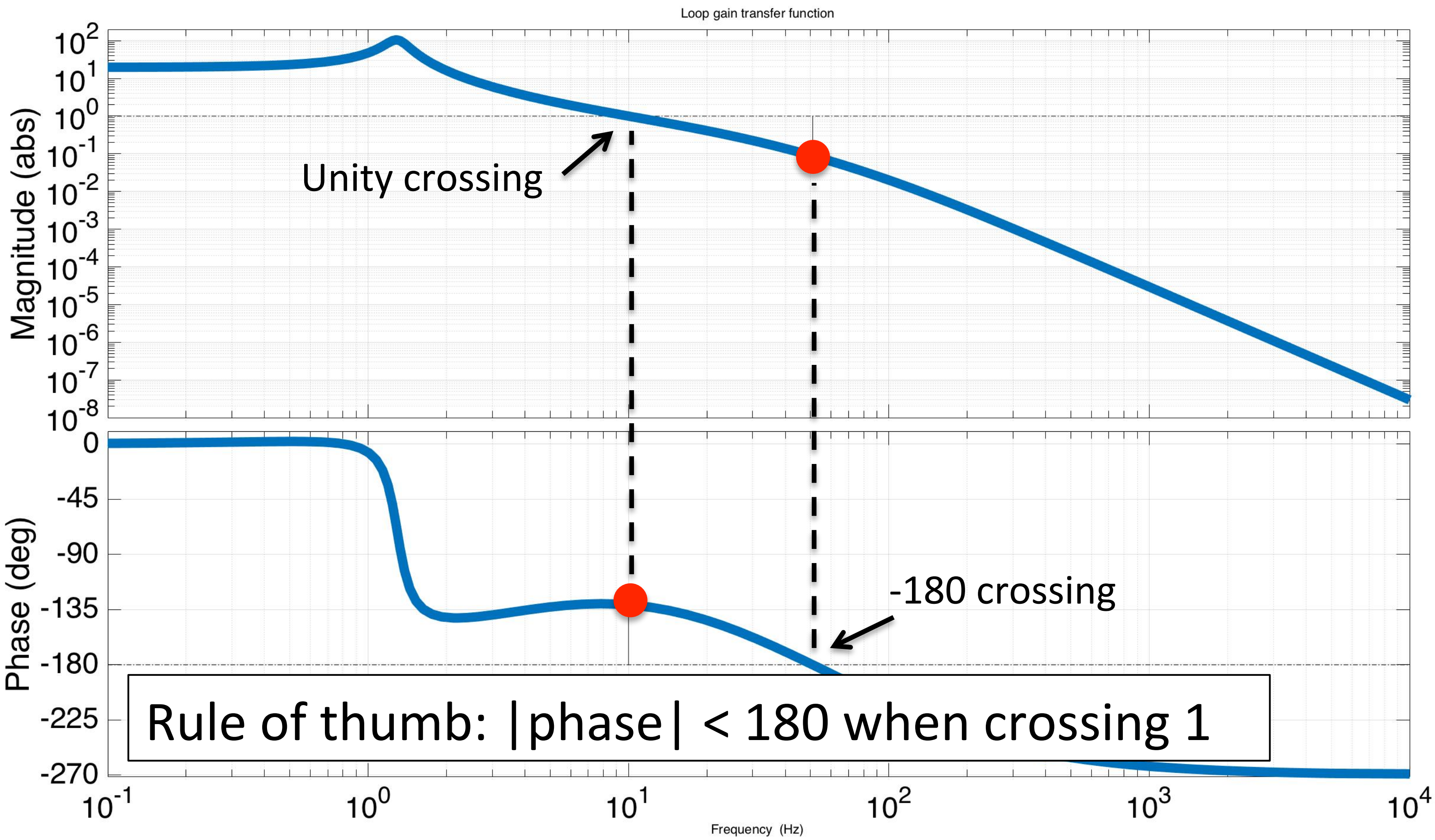
Matlab code: `nichols(50*Pa*C),grid off`

Loop Gain Nichols Plot: 50*$P_a C$



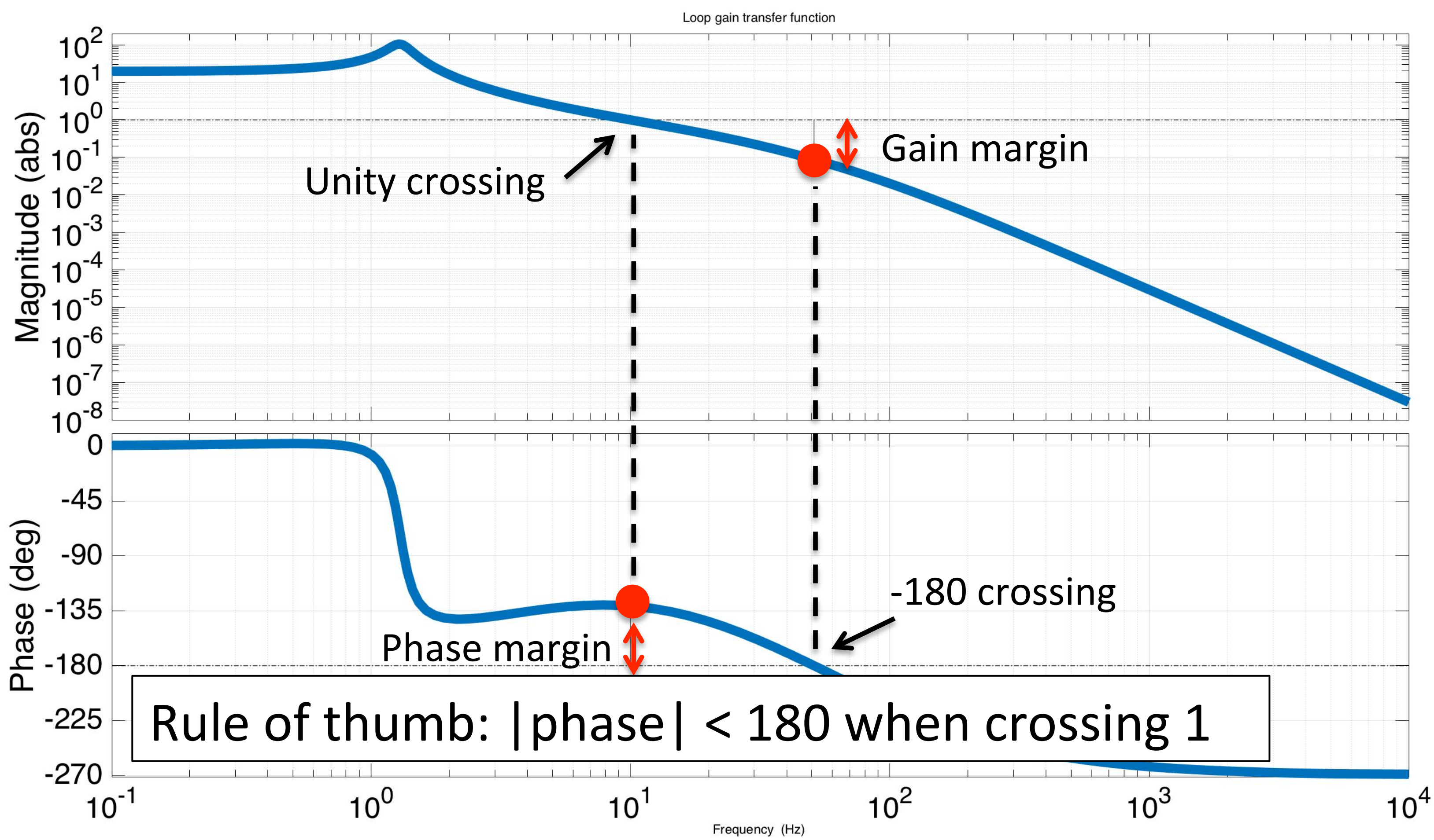
Matlab code: `nichols(Pa*C),grid off`

Loop Gain TF: $P_a C$



Matlab for control filter: `C = zpk(-2*pi*3.33,-2*pi*[30;100],1.4e+10)`

Loop Gain TF: $P_a C$



Rule of thumb: |phase| < 180 when crossing 1

Matlab for control filter: `C = zpk(-2*pi*3.33,-2*pi*[30;100],1.4e+10)`

How to do control design

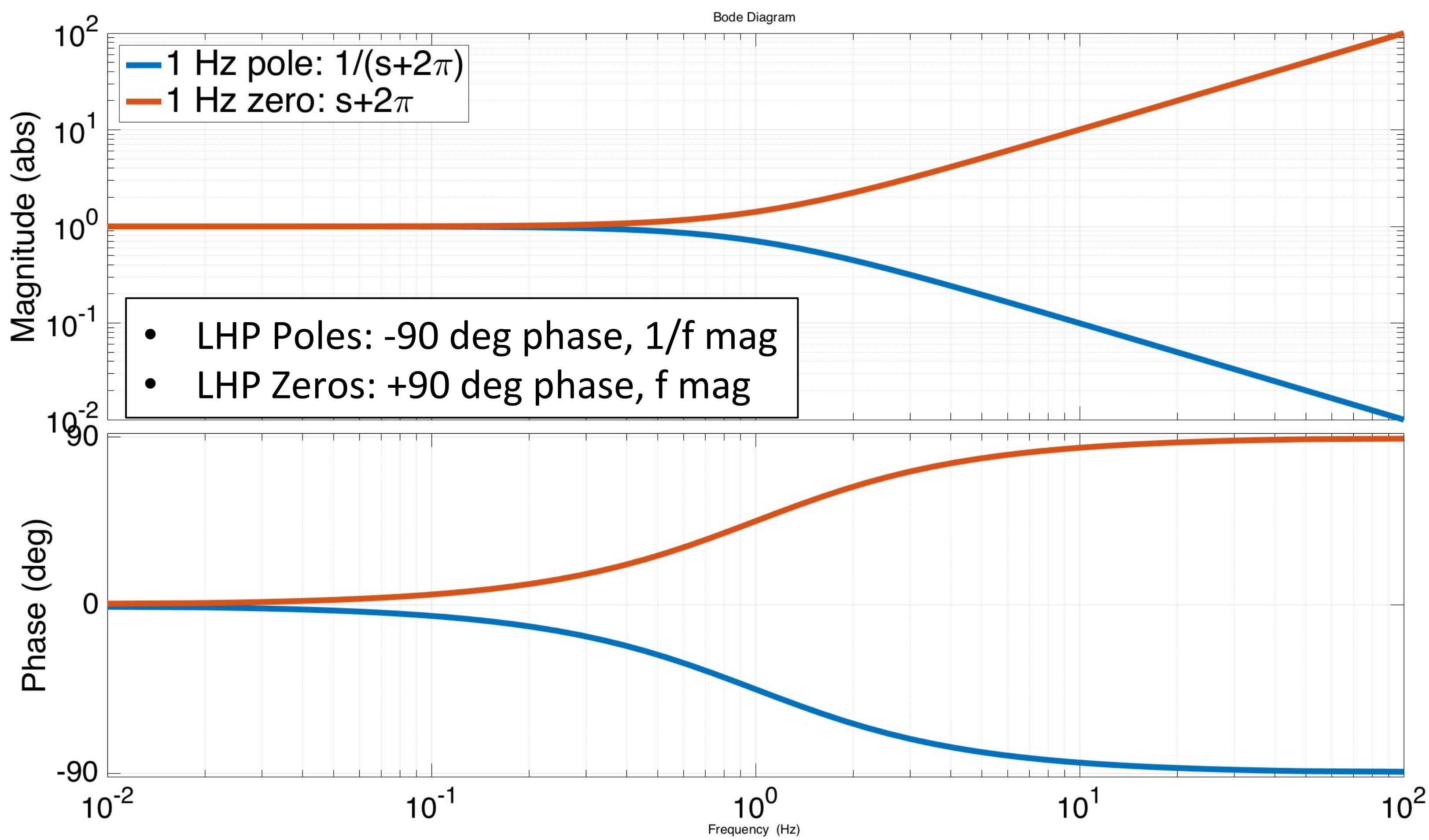
Multiple methods, but the most common is called '**loop shaping**'

$$C = \frac{\prod_{j=1}^{m} \left(s + z_j \right)}{\prod_{k=1}^{n} \left(s + p_k \right)}$$

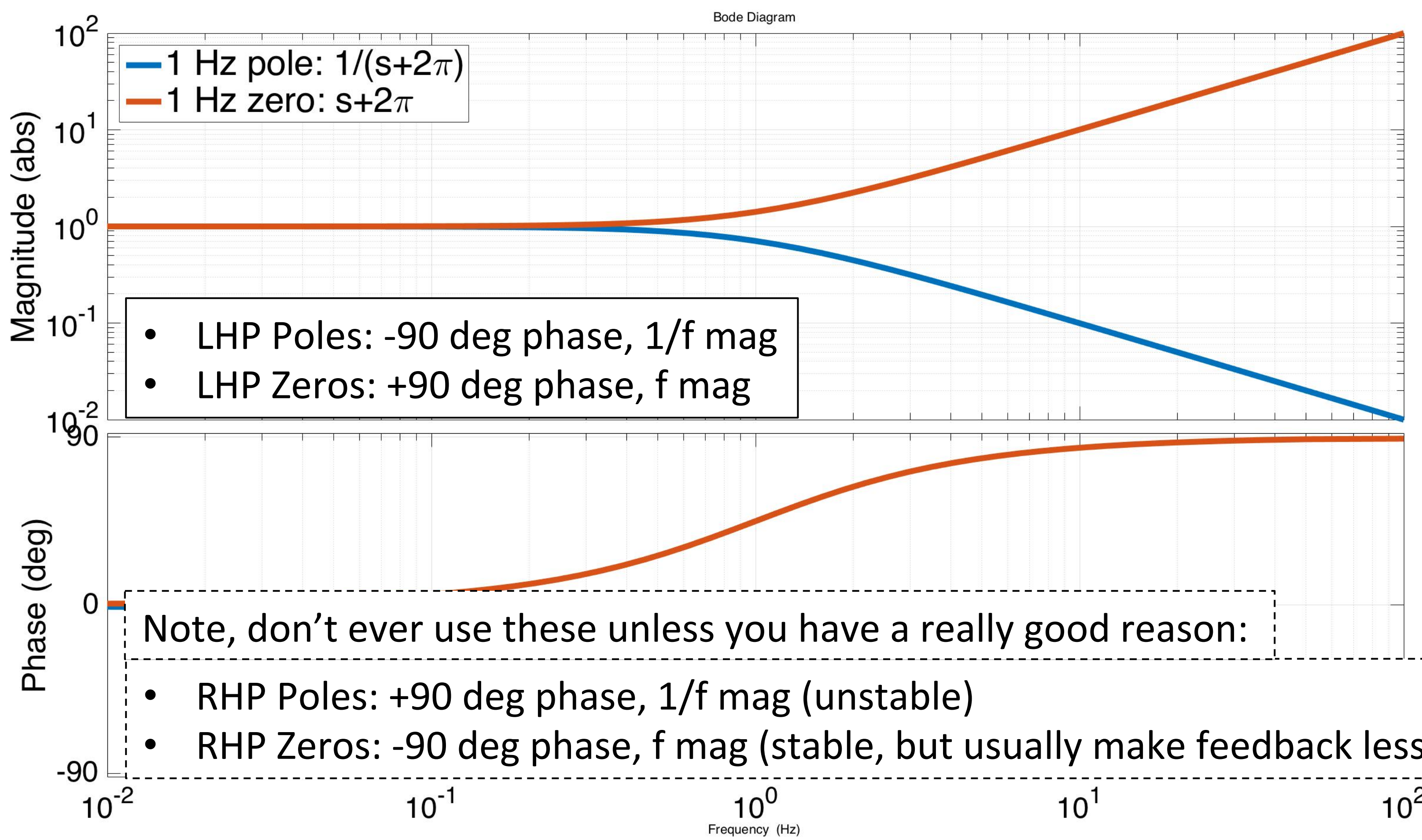
- Place poles and zeros until the loop gain is 'shaped' the way you like it

- Causal filters require at least as many poles as zeros: n ≥ m. Non-causal filters respond with infinite magnitude and positive phase at infinite frequency, which they can only do if they have access to future data.

Control design: poles and zeros



LIGO Control design: poles and zeros

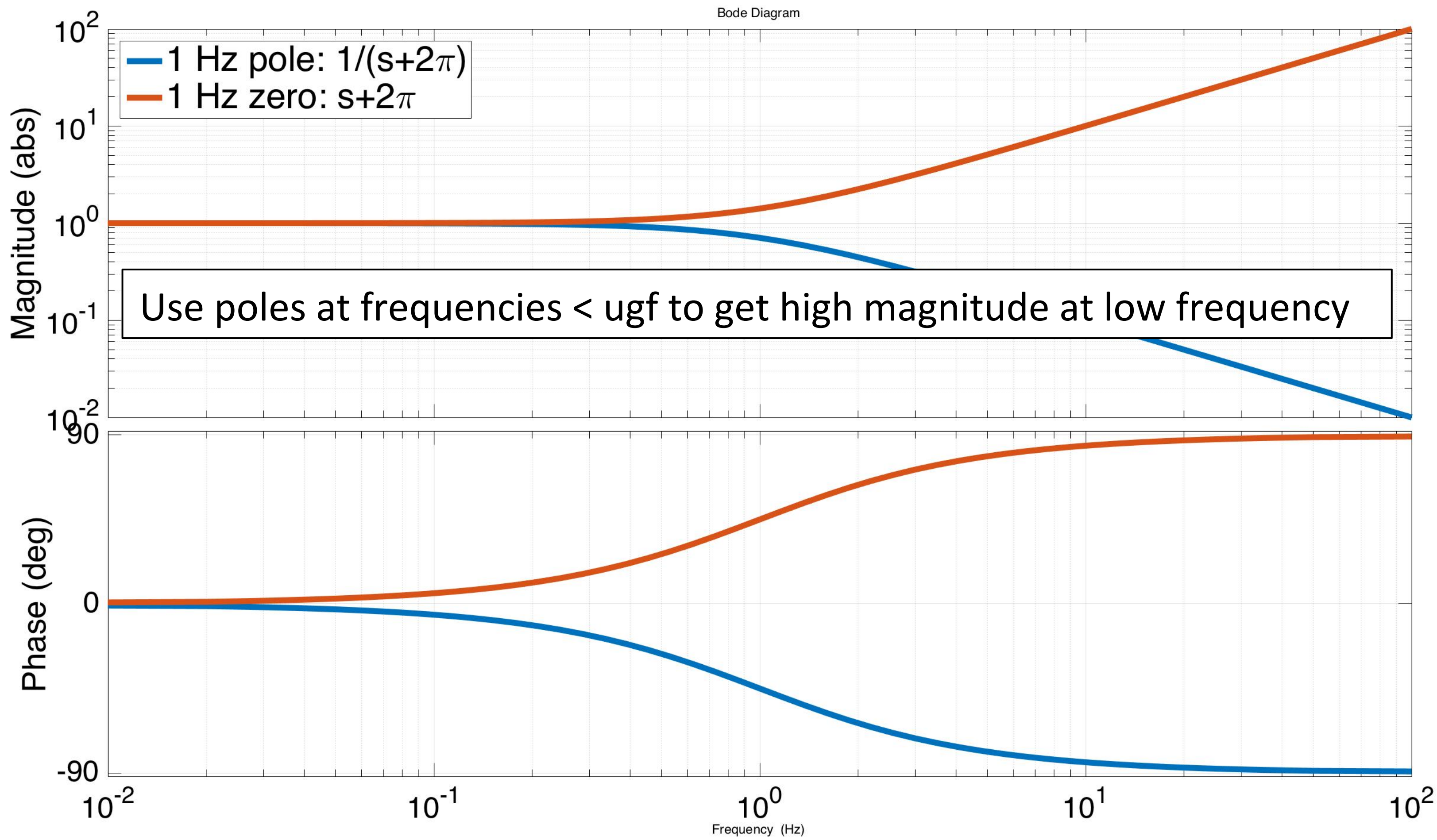


- LHP Poles: -90 deg phase, 1/f mag
- LHP Zeros: +90 deg phase, f mag

Note, don't ever use these unless you have a really good reason:

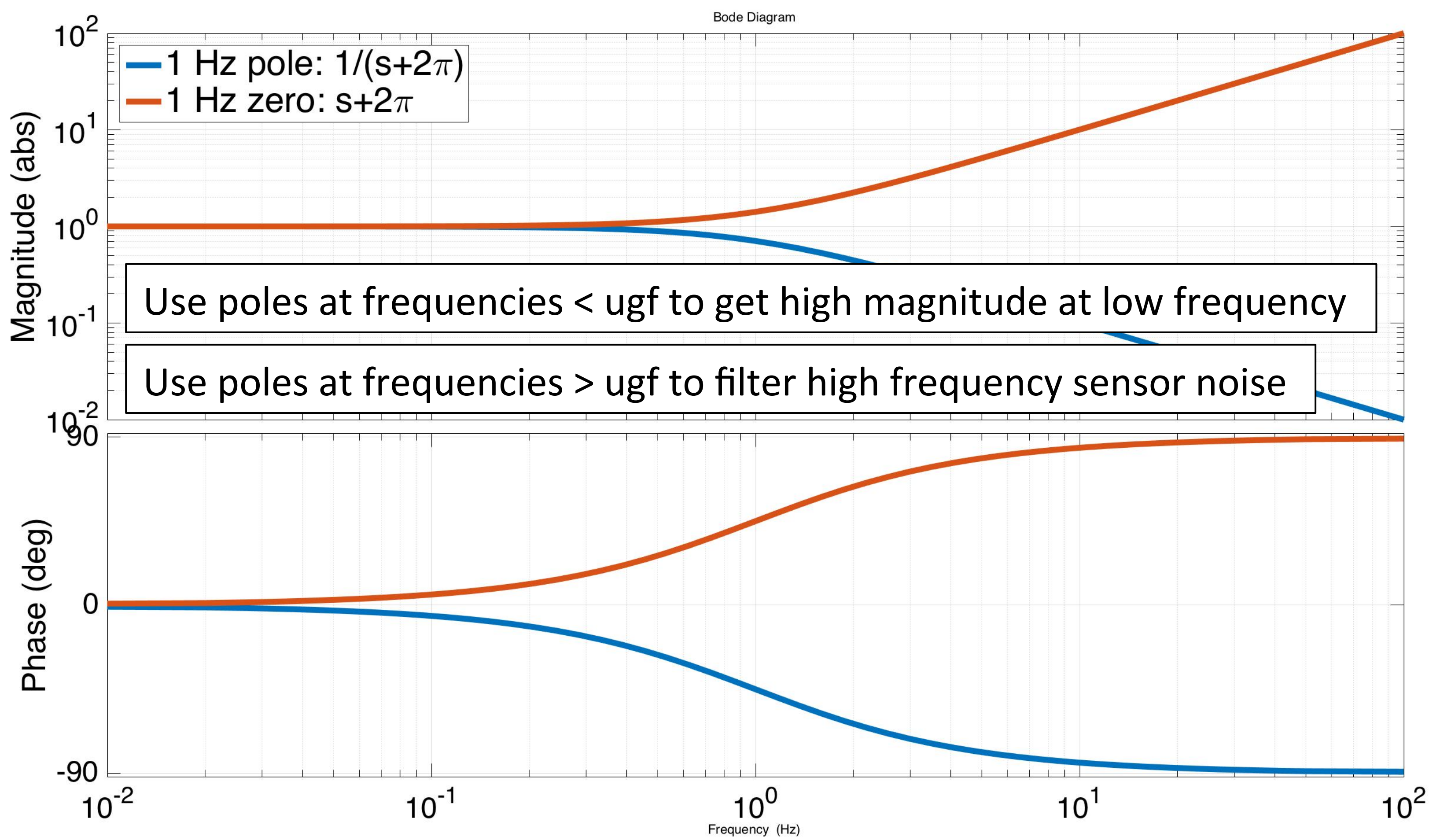
- RHP Poles: +90 deg phase, 1/f mag (unstable)
- RHP Zeros: -90 deg phase, f mag (stable, but usually make feedback less stable)

Control design: poles and zeros

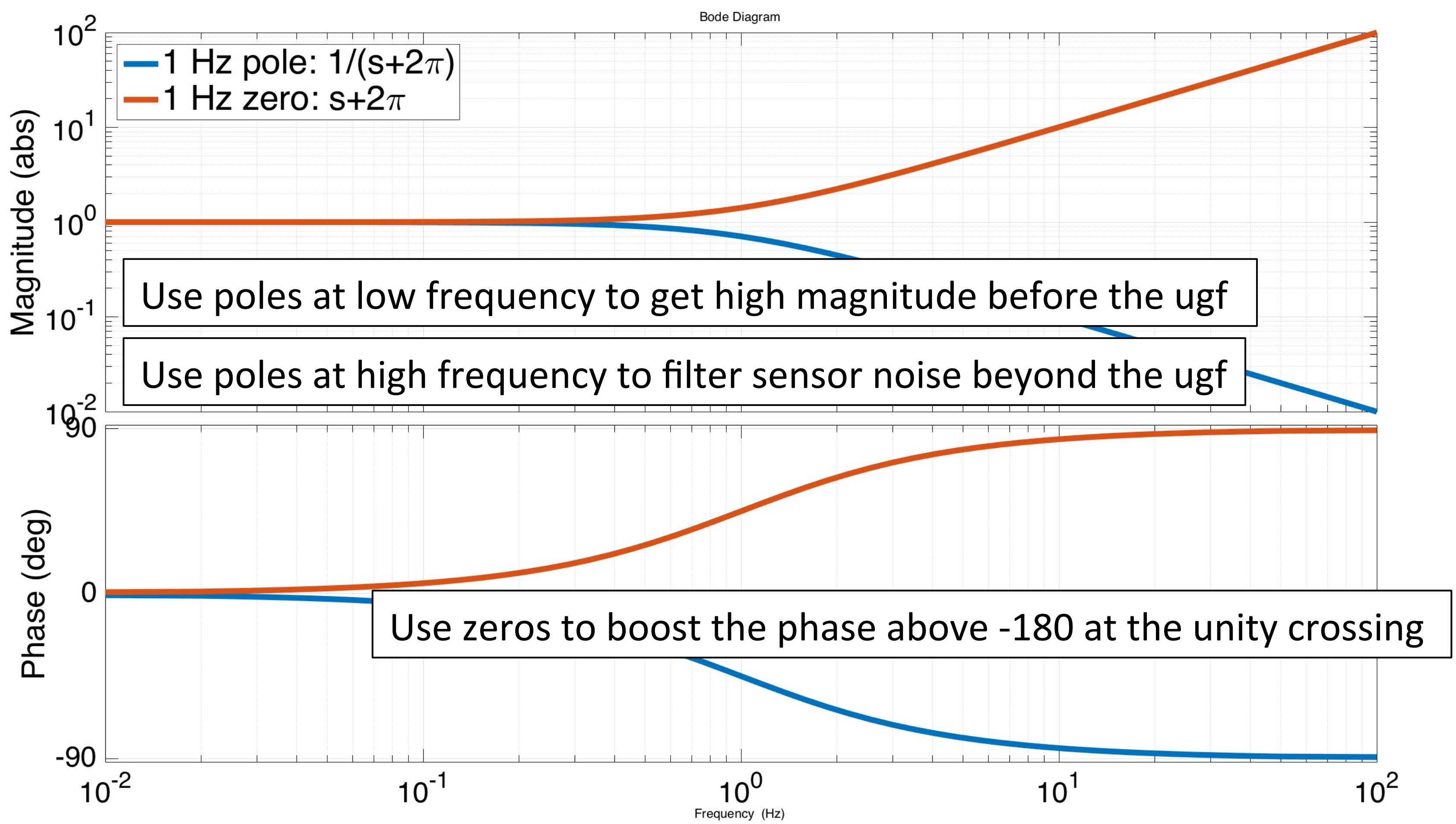


Use poles at frequencies < ugf to get high magnitude at low frequency

Control design: poles and zeros

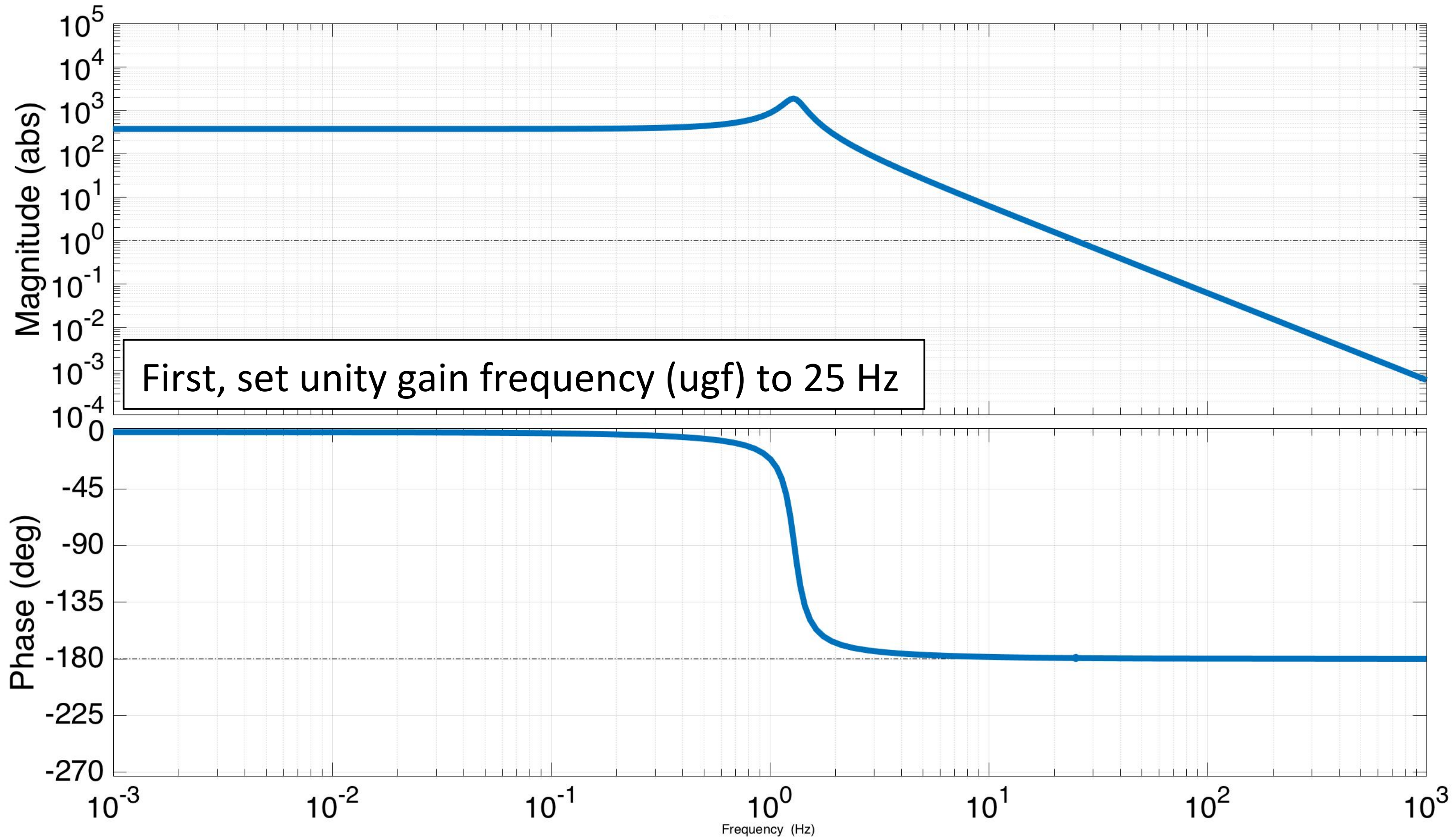


LIGO Control design: poles and zeros



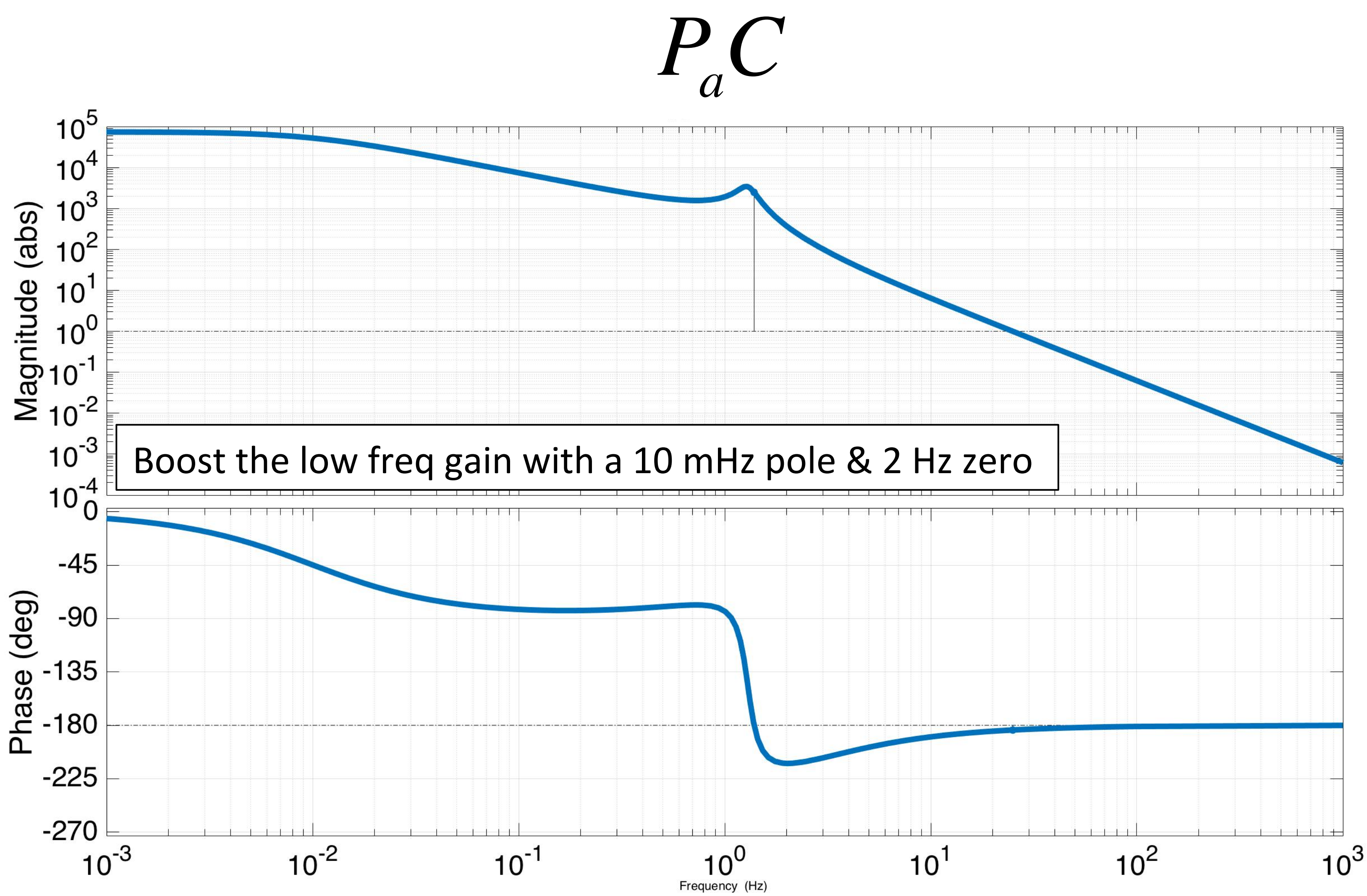
LIGO Control design: loop shaping

$$P_a C$$



Matlab for control filter: `C = zpk([],[],4.7e7)`

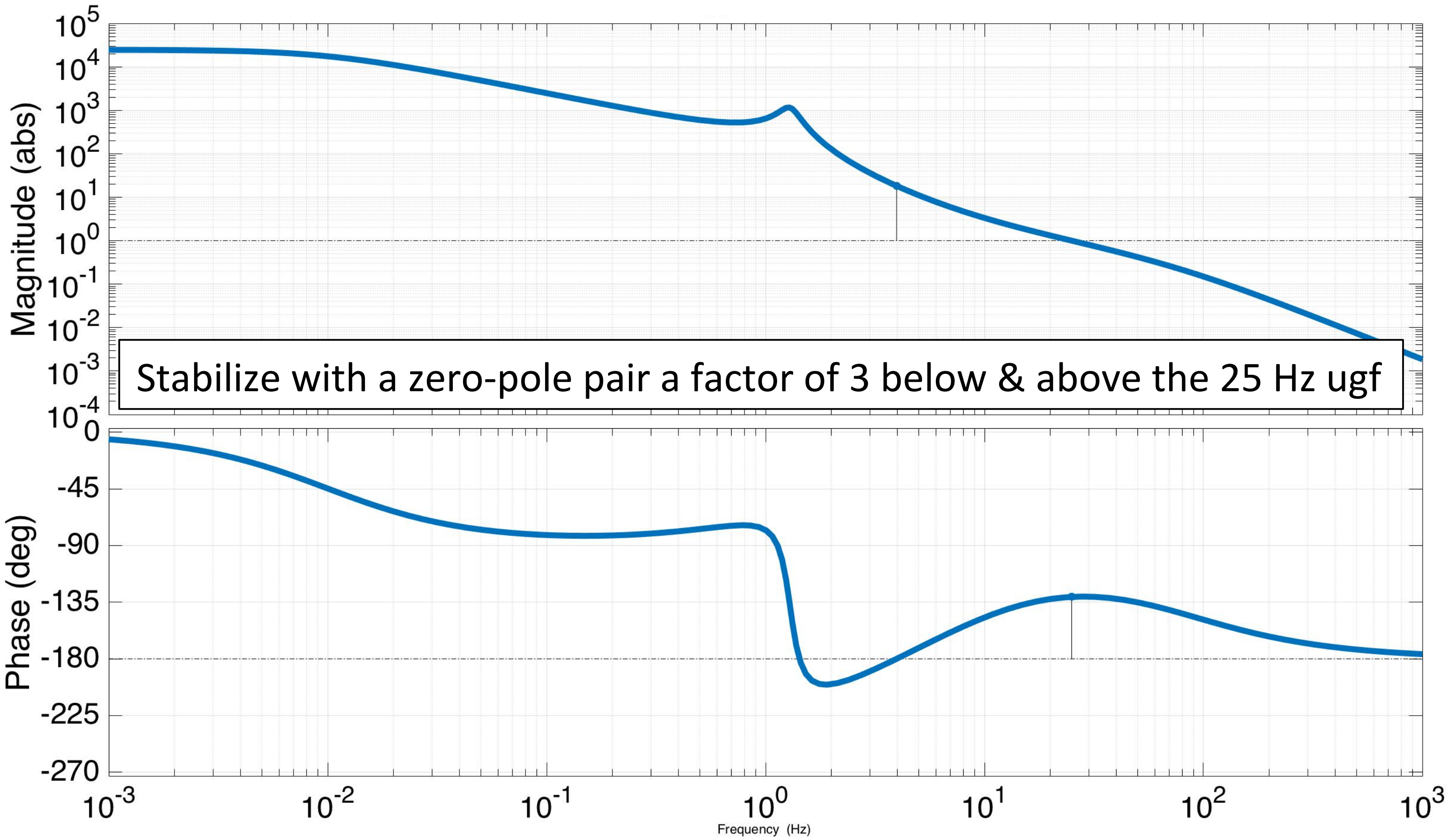
Control design: loop shaping



Matlab for control filter: `C = zpk(-2*pi*[2],-2*pi*[0.01],4.7e7)`

Control design: loop shaping

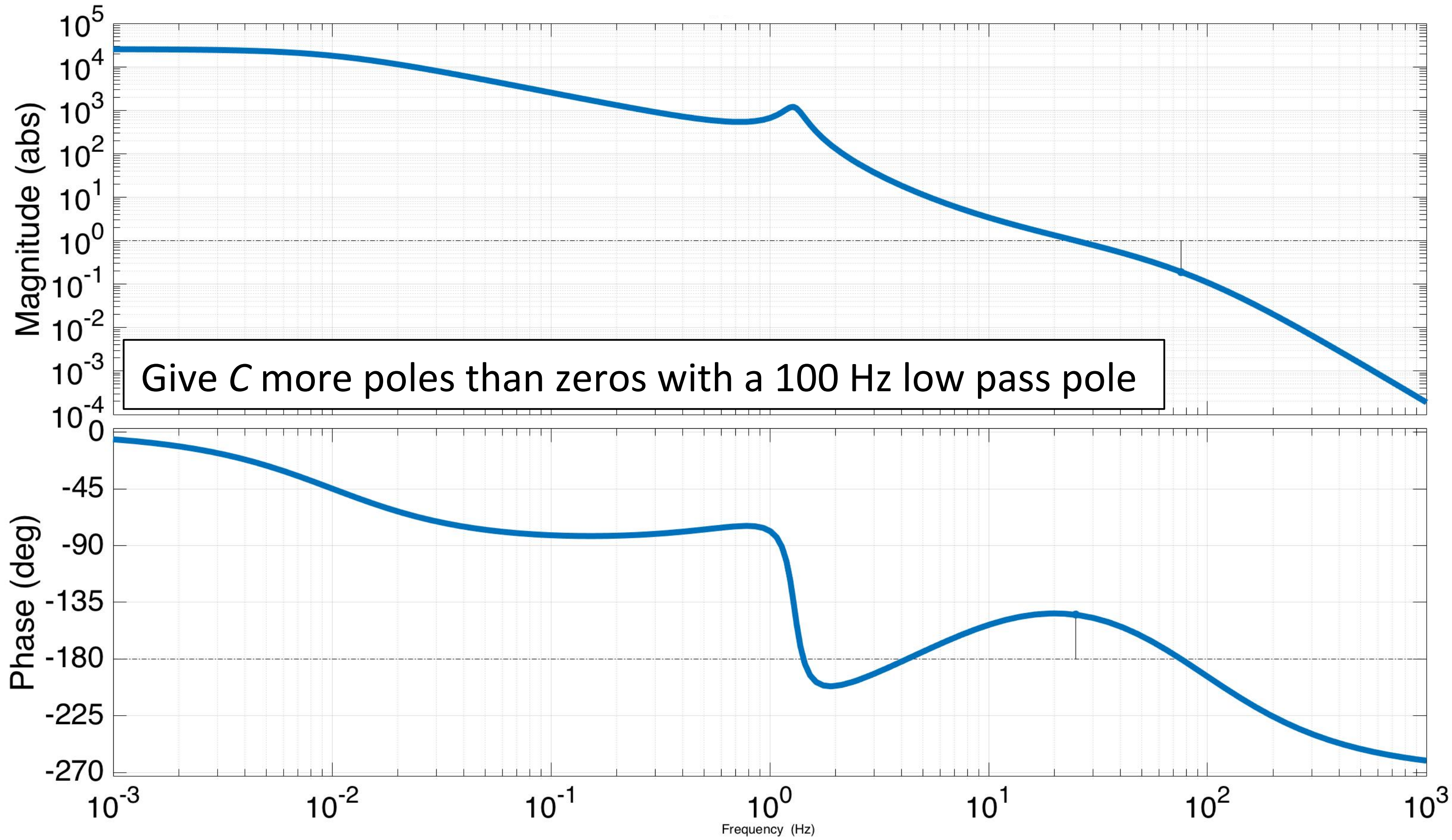
$P_a C$



Matlab for control filter: `C = zpk(-2*pi*[2,25/3],-2*pi*[0.01,25*3],1.4e8)`

Control design: loop shaping

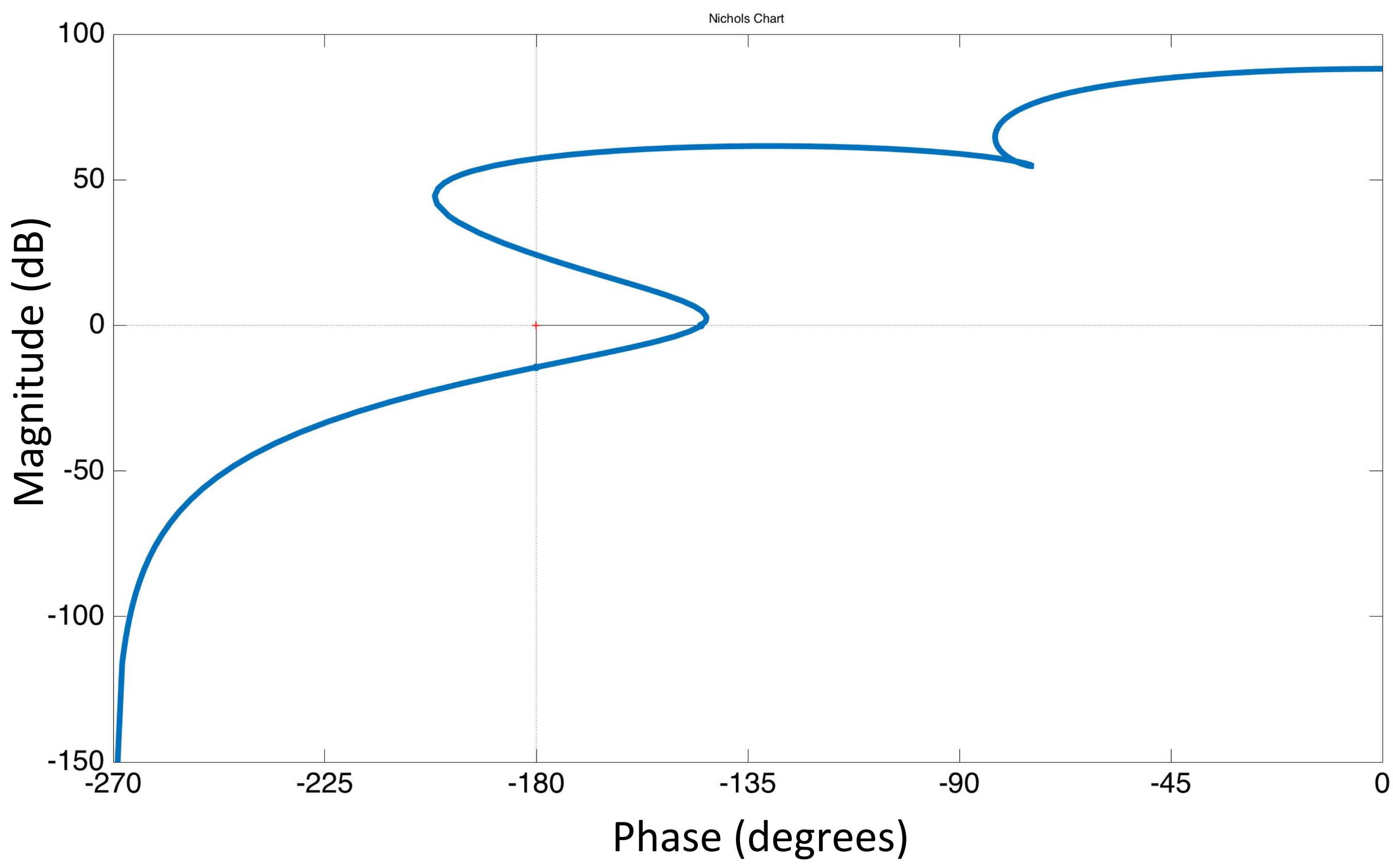
$$P_a C$$



Matlab for control filter: `C = zpk(-2*pi*[2,25/3],-2*pi*[0.01,25*3,100],9.1e10)`

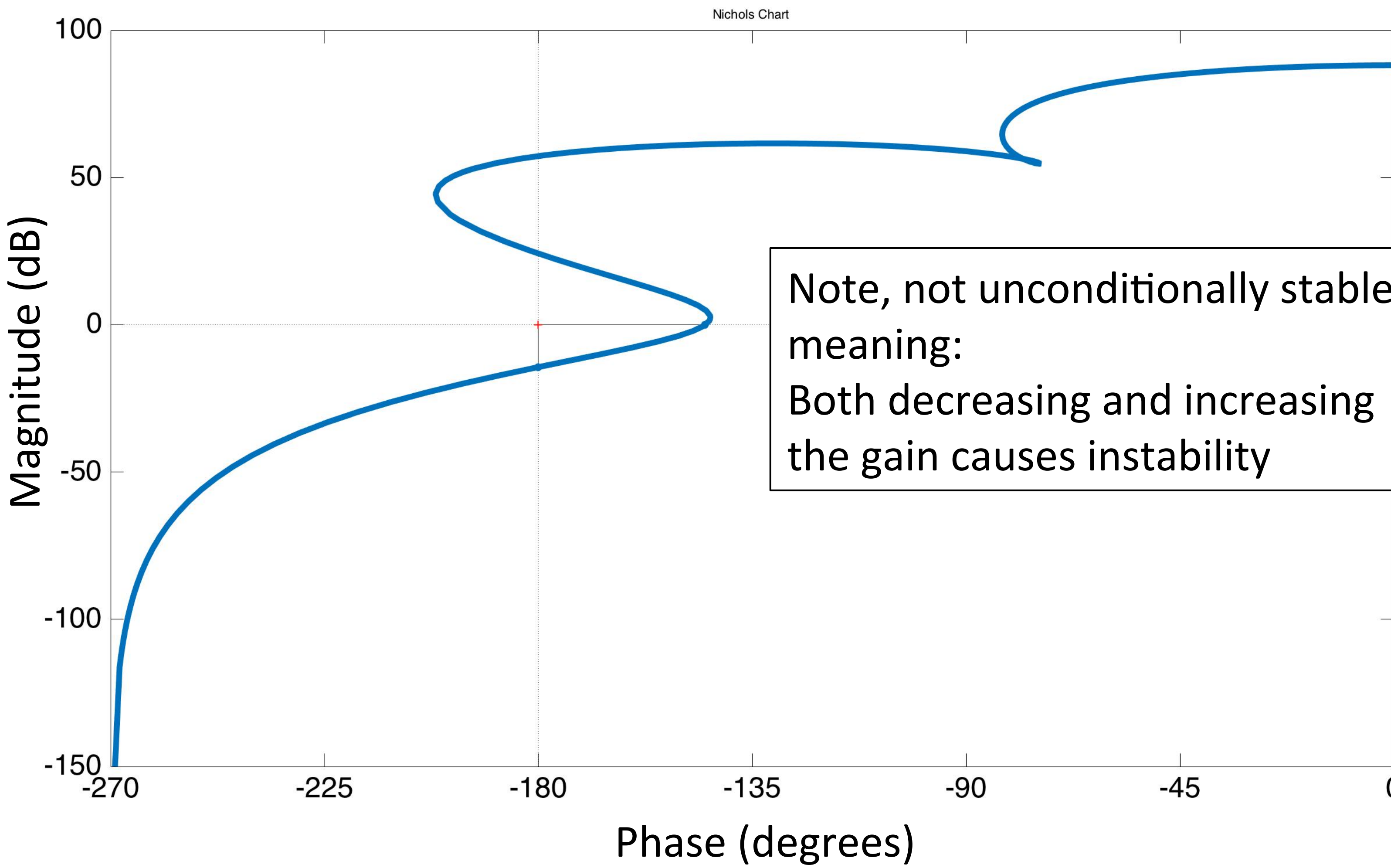
Control design: loop shaping

$P_a C$ Nichols Plot



Control design: loop shaping

$P_a C$ Nichols Plot

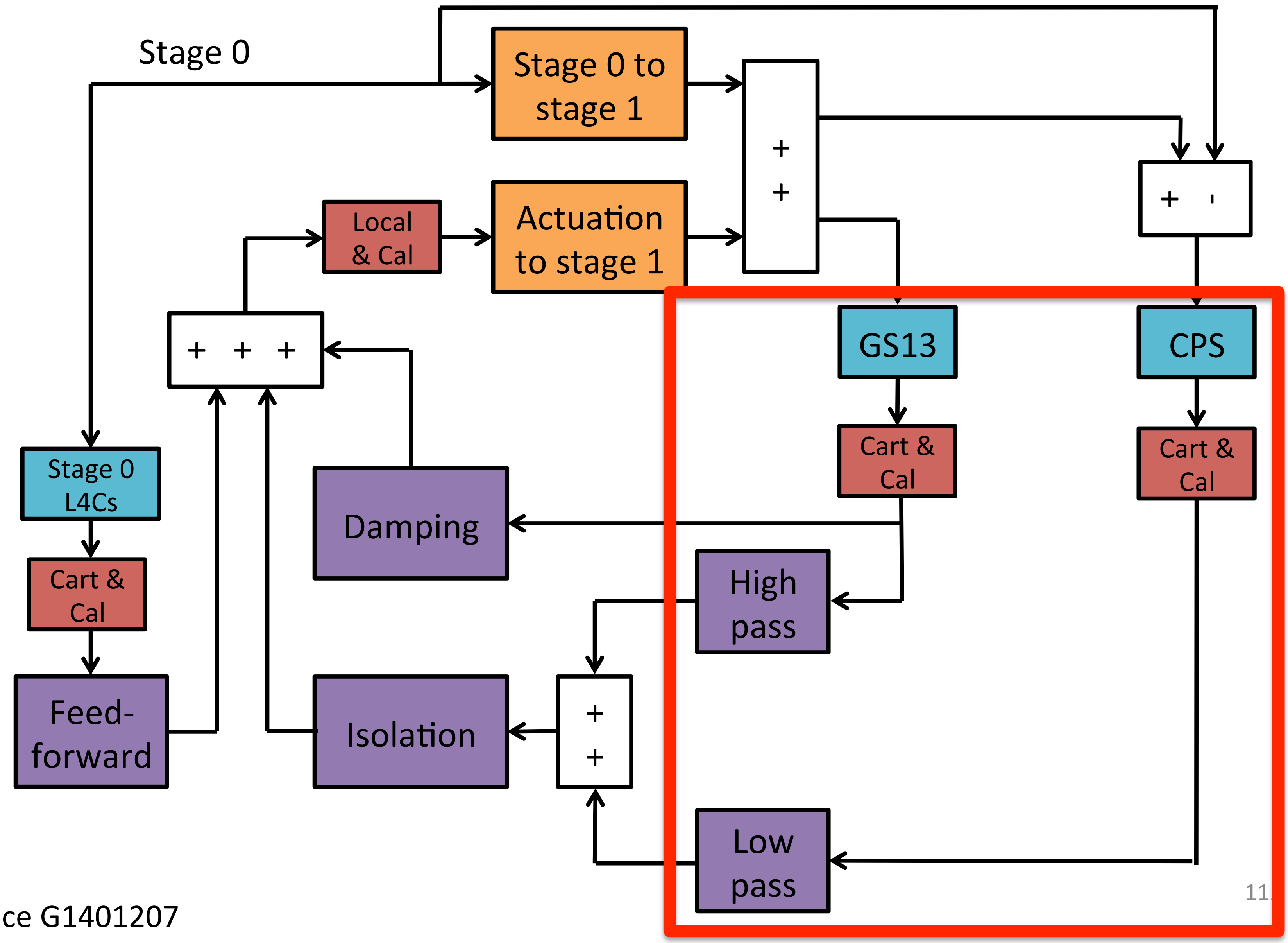


Note, not unconditionally stable, meaning:
Both decreasing and increasing the gain causes instability

Lecture 2 – Part 3

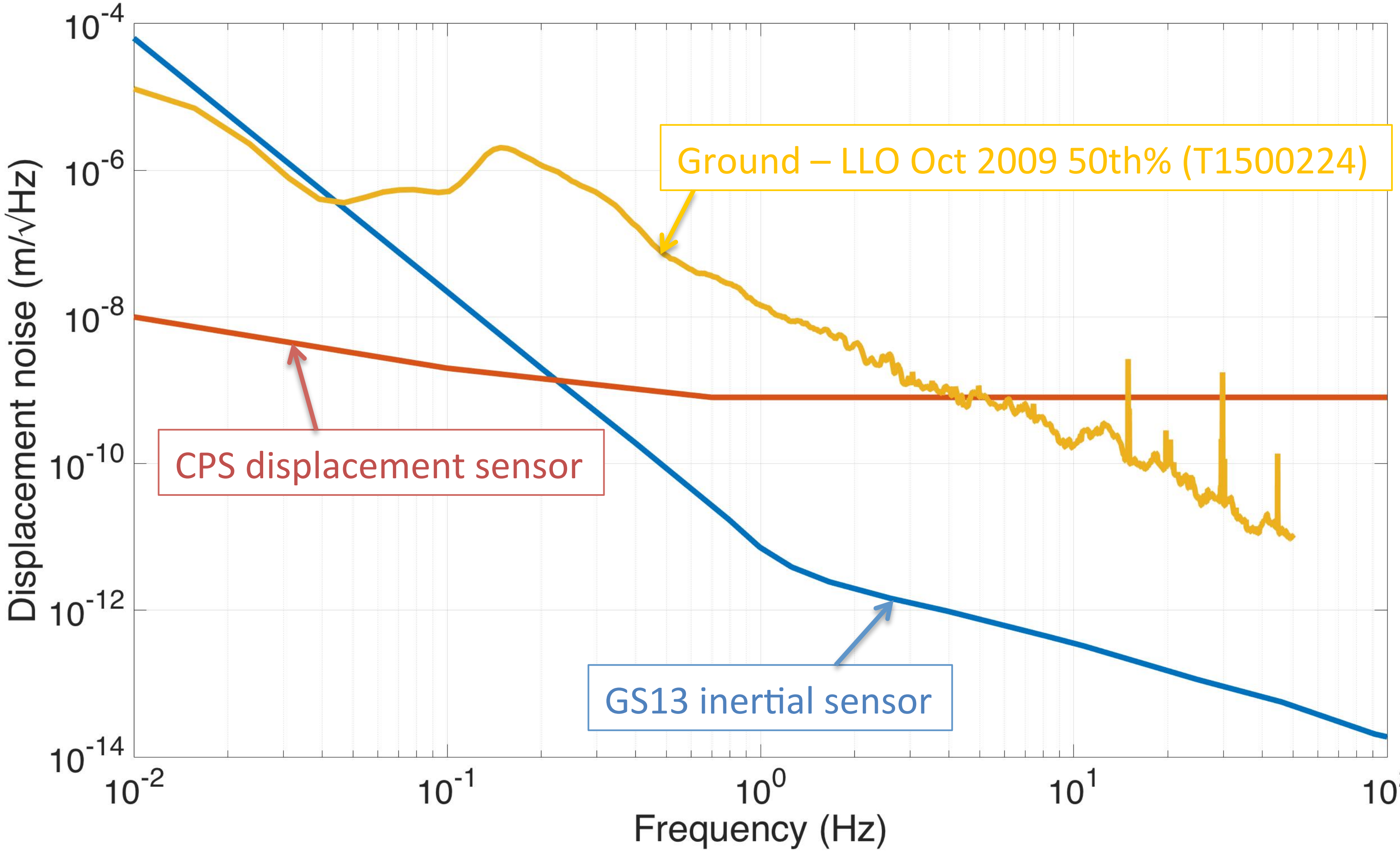
Sensor Blending

Stage 1 of 1-stage of in-vacuum isolation table (HAM-ISI)

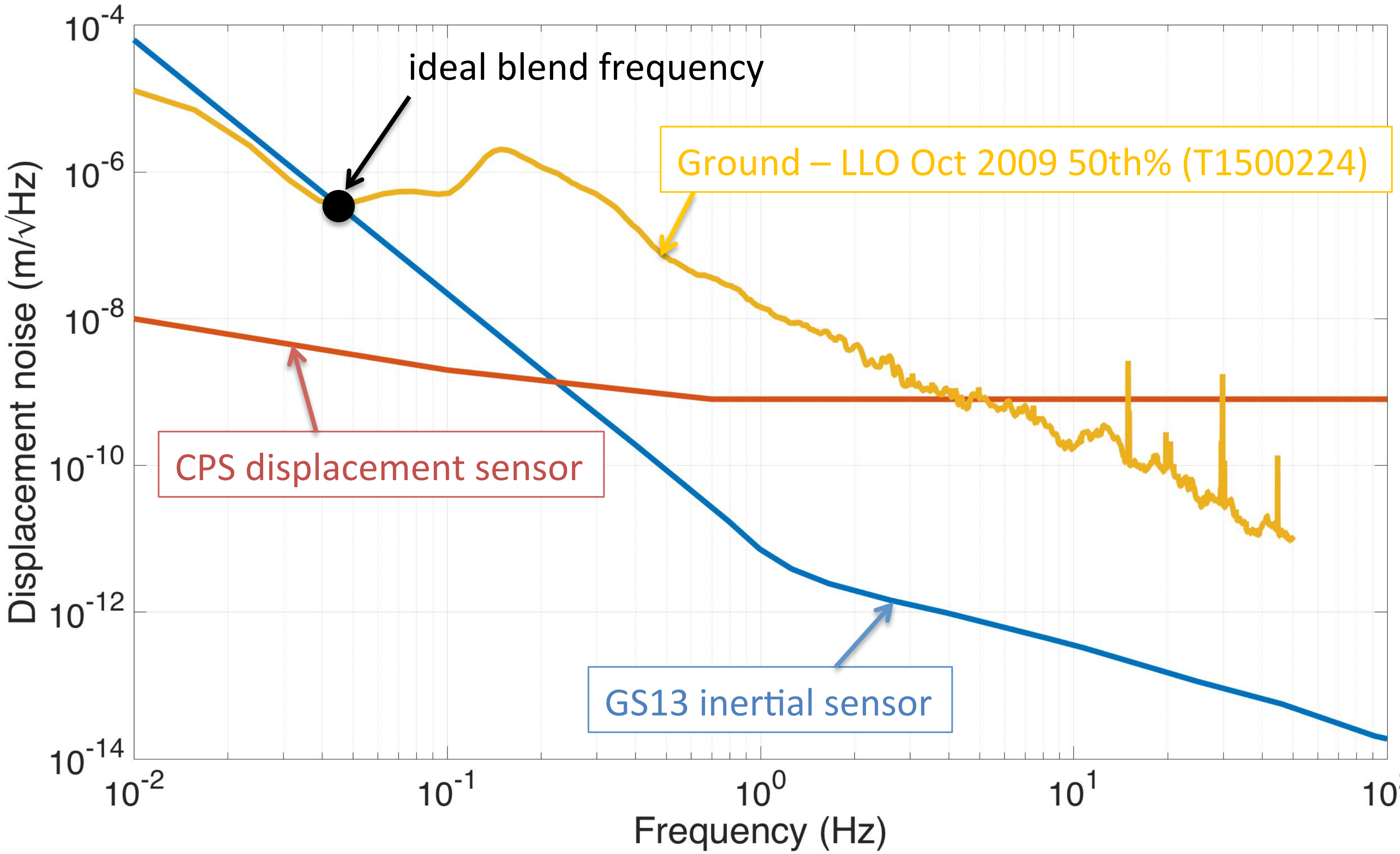


Reference G1401207

HAM ISI Sensor Noises



HAM ISI Sensor Noises



Blend filter design

- Low pass the inertial sensor, high pass the displacement sensor
- Low pass + high pass = 1

Blend filter design

- Low pass the inertial sensor, high pass the displacement sensor
- Low pass + high pass = 1

Simple approach

B_{LP} Make some low pass filter

Blend filter design

- Low pass the inertial sensor, high pass the displacement sensor
- Low pass + high pass = 1

Simple approach

B_{LP} Make some low pass filter

Then, the high pass is simply

$$B_{HP} = 1 - B_{LP}$$

Blend filter design

- Low pass the inertial sensor, high pass the displacement sensor
- Low pass + high pass = 1

Simple approach

B_{LP} Make some low pass filter

Then, the high pass is simply

$$B_{HP} = 1 - B_{LP}$$

This works, but hard to tune both simultaneously.

Blend filter design

- Low pass the inertial sensor, high pass the displacement sensor
- Low pass + high pass = 1

Simple approach

B_{LP} Make some low pass filter

Then, the high pass is simply

$$B_{HP} = 1 - B_{LP}$$

This works, but hard to tune both simultaneously.
Try this instead:

$$B_{LP} = \frac{B_{LP\_prototype}}{B_{LP\_prototype} + B_{HP\_prototype}} \qquad B_{HP} = \frac{B_{HP\_prototype}}{B_{LP\_prototype} + B_{HP\_prototype}}$$

Blend filter design

But be careful!

$$B_{LP} = \frac{B_{LP\_prototype}}{B_{LP\_prototype} + B_{HP\_prototype}}$$

Blend filter design

But be careful!

$$B_{LP} = \frac{B_{LP\_prototype}}{B_{LP\_prototype} + B_{HP\_prototype}}$$

$$B_{LP} = \frac{1}{B_{LP\_prototype}} \frac{B_{LP\_prototype}}{1 + B_{HP\_prototype} / B_{LP\_prototype}}$$

This looks like a closed loop TF,
where the 'loop gain' is the ratio of the prototype filters.

Blend filter design

But be careful!

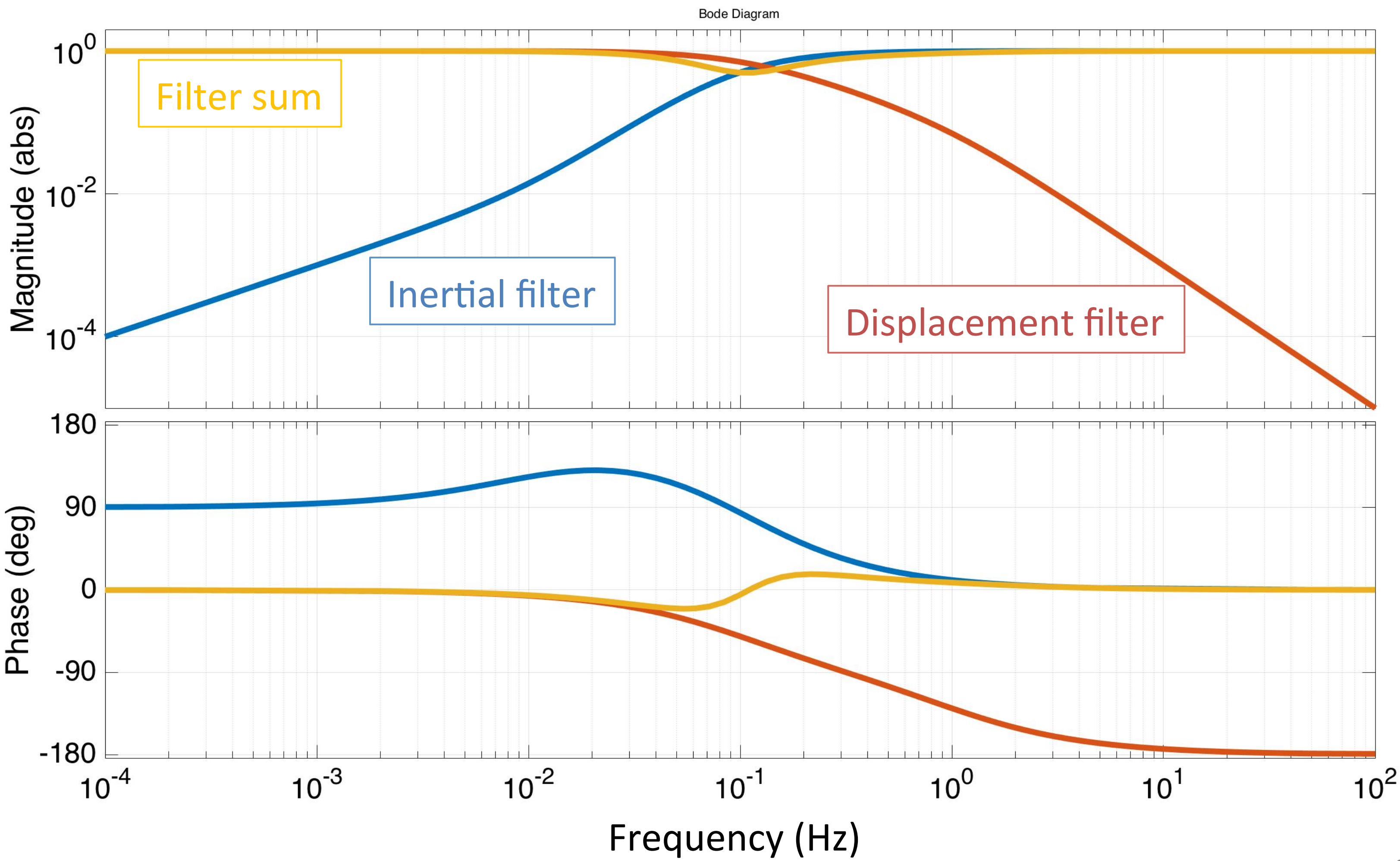
$$B_{LP} = \frac{B_{LP\_prototype}}{B_{LP\_prototype} + B_{HP\_prototype}}$$

$$B_{LP} = \frac{1}{B_{LP\_prototype}} \frac{B_{LP\_prototype}}{1 + B_{HP\_prototype} / B_{LP\_prototype}}$$

This looks like a closed loop TF,
where the ‘loop gain’ is the ratio of the prototype filters.

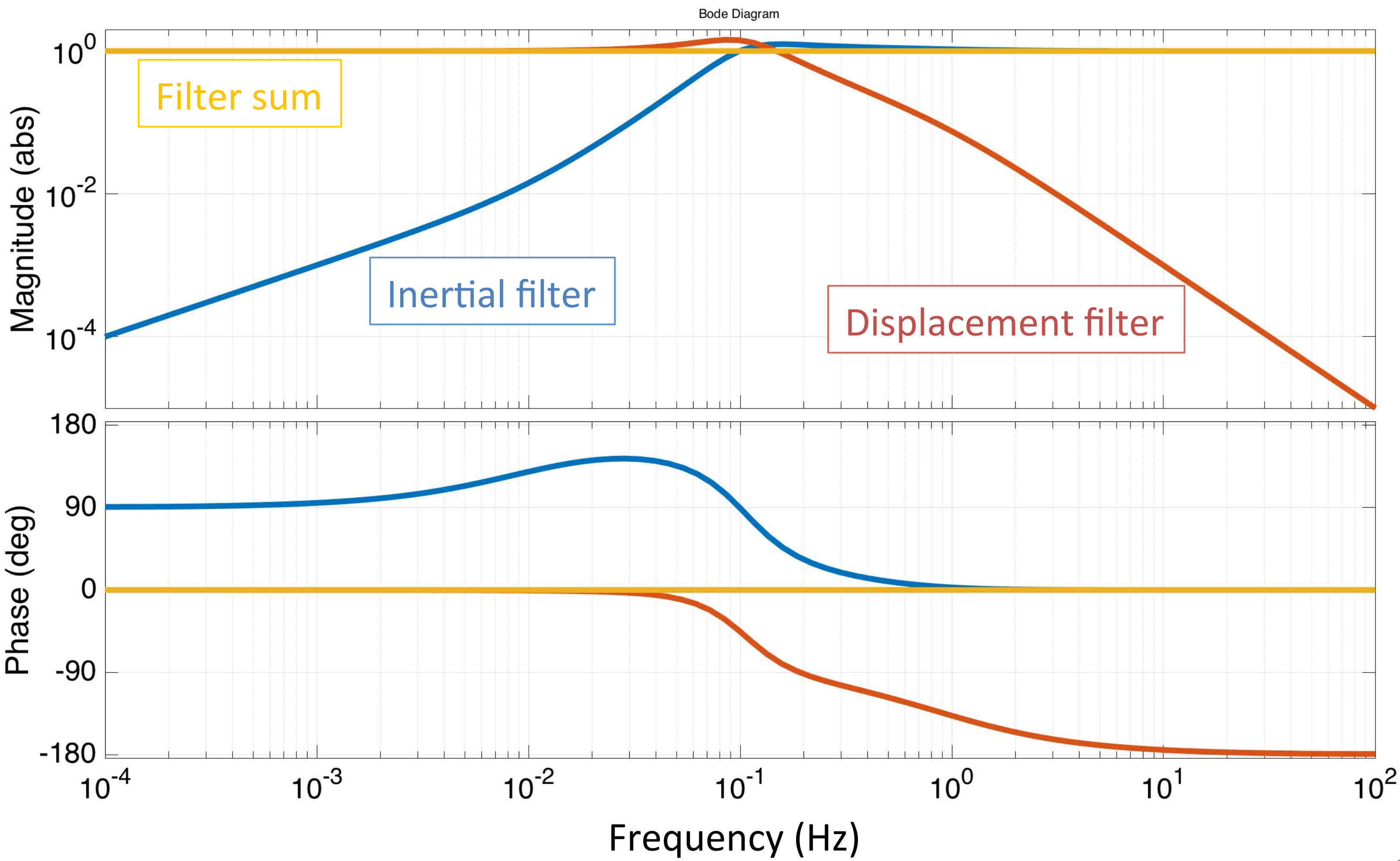
- We must watch out for stability.
- In practice, just keep the filters within 180 degrees of each other when their magnitudes cross.
- With this approach we’ve traded some stability for more design parameters

Prototype Blends



See Matlab code Blend_Example_Lecture2.m

Implemented Blends

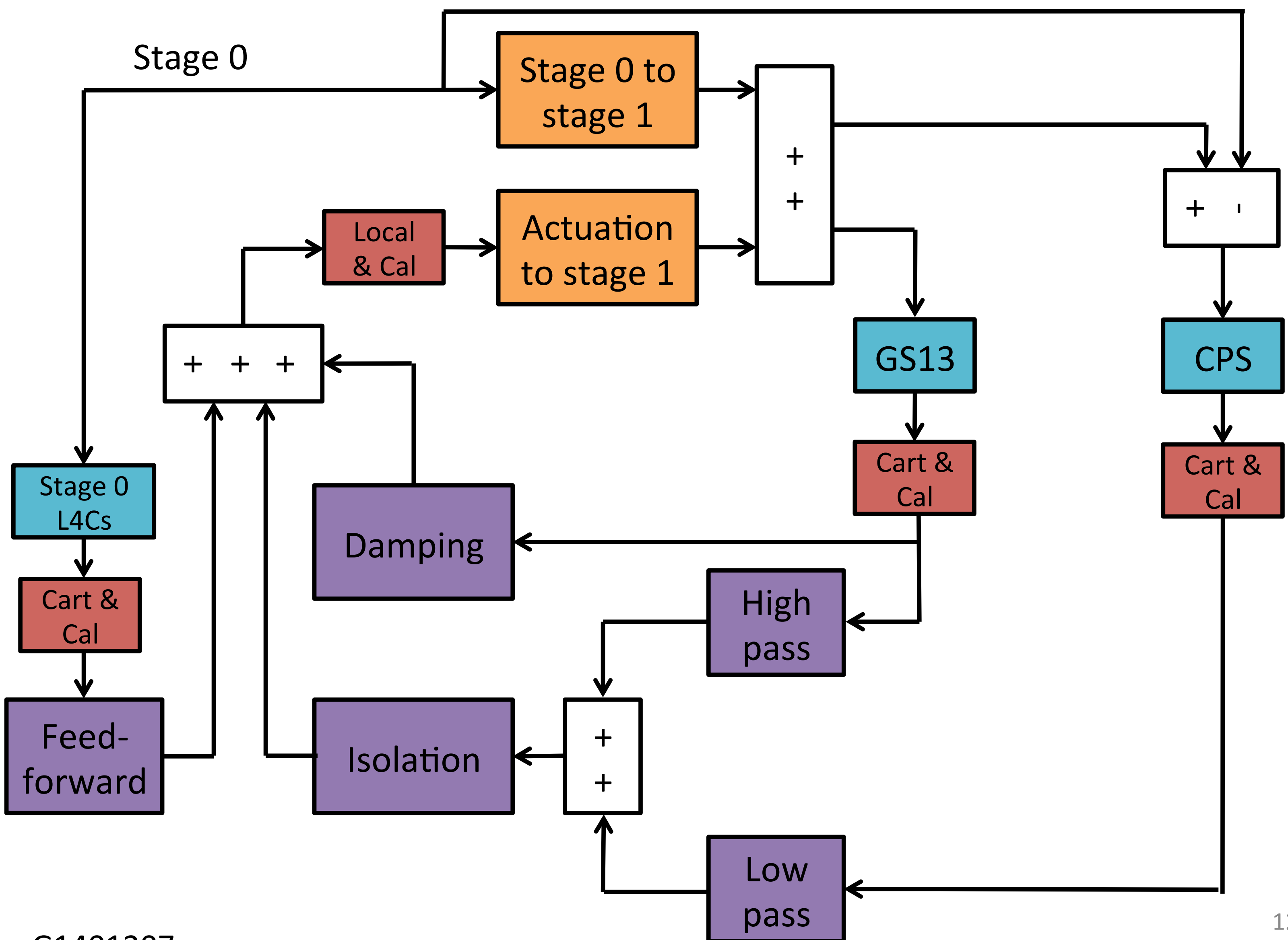


See Matlab code Blend_Example_Lecture2.m

Lecture 2 Summary

- Seismic feedforward control depends only on the system's connection to the ground.
- For feedback stability the abs(phase) < 180 when the magnitude drops below 1
- Sensor blending uses the displace sensor at low frequencies, the inertial sensor at high frequencies. Stability rules apply.

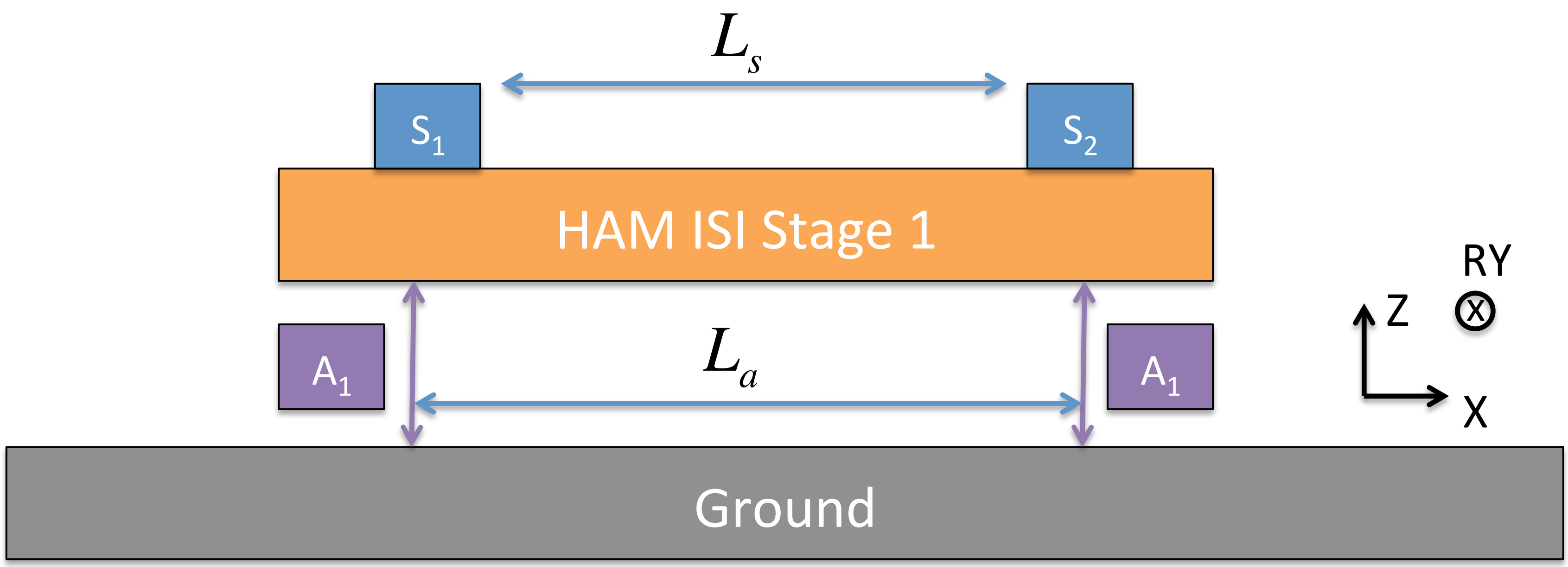
Stage 1 of 1-stage of in-vacuum isolation table (HAM-ISI)



Reference G1401207

Lecture 2 – Backups

Matrix transformations



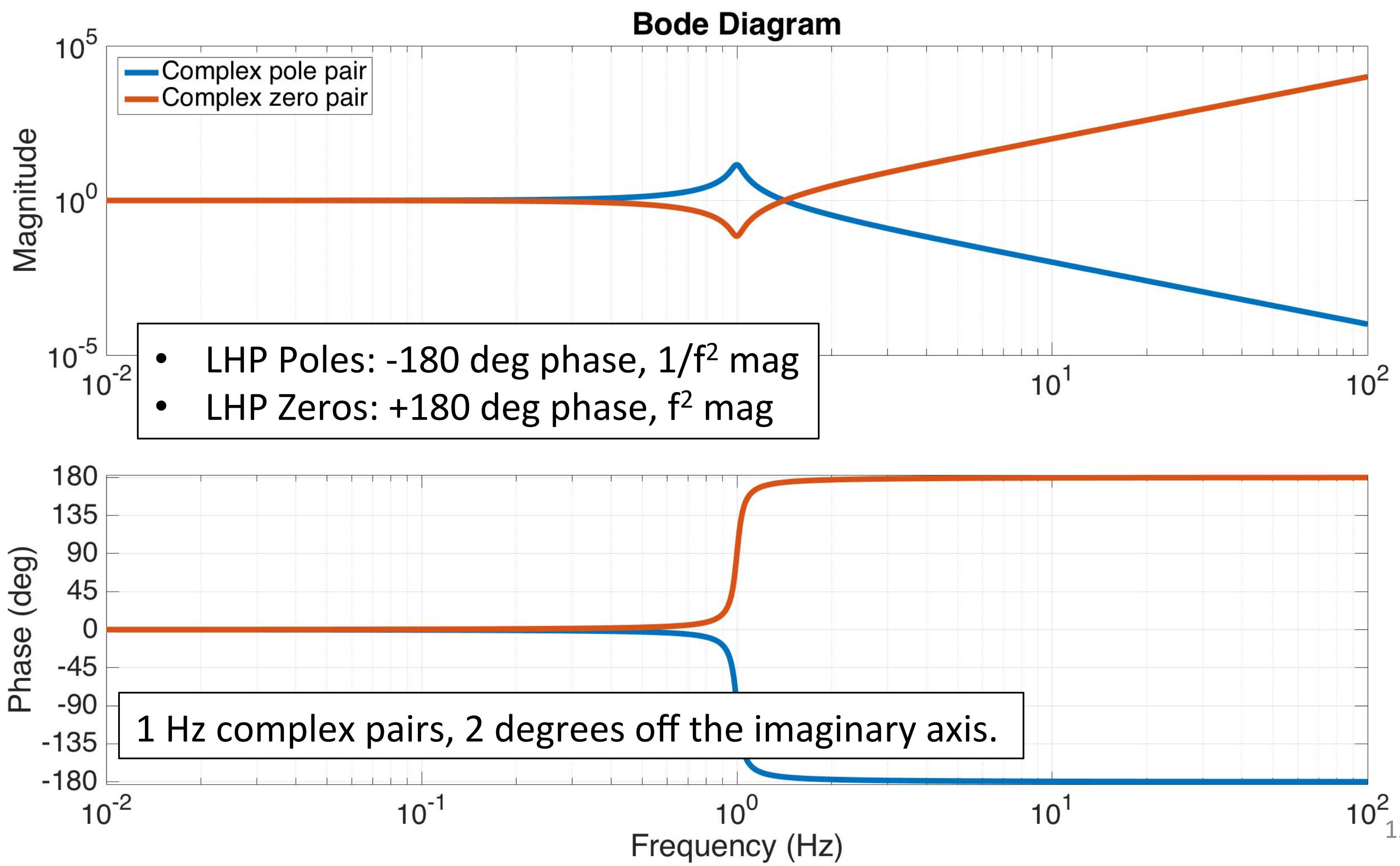
Sensing matrix

$$\begin{bmatrix} Z_s \\ RY_s \end{bmatrix} = \begin{bmatrix} 0.5 & 0.5 \\ 1/L_s & -1/L_s \end{bmatrix} \begin{bmatrix} S_1 \\ S_2 \end{bmatrix}$$

Actuation matrix

$$\begin{bmatrix} A_1 \\ A_2 \end{bmatrix} = \begin{bmatrix} 0.5 & 1/L_a \\ 0.5 & -1/L_a \end{bmatrix} \begin{bmatrix} Z_a \\ RY_a \end{bmatrix}$$

LIGO Complex pairs of poles and zeros

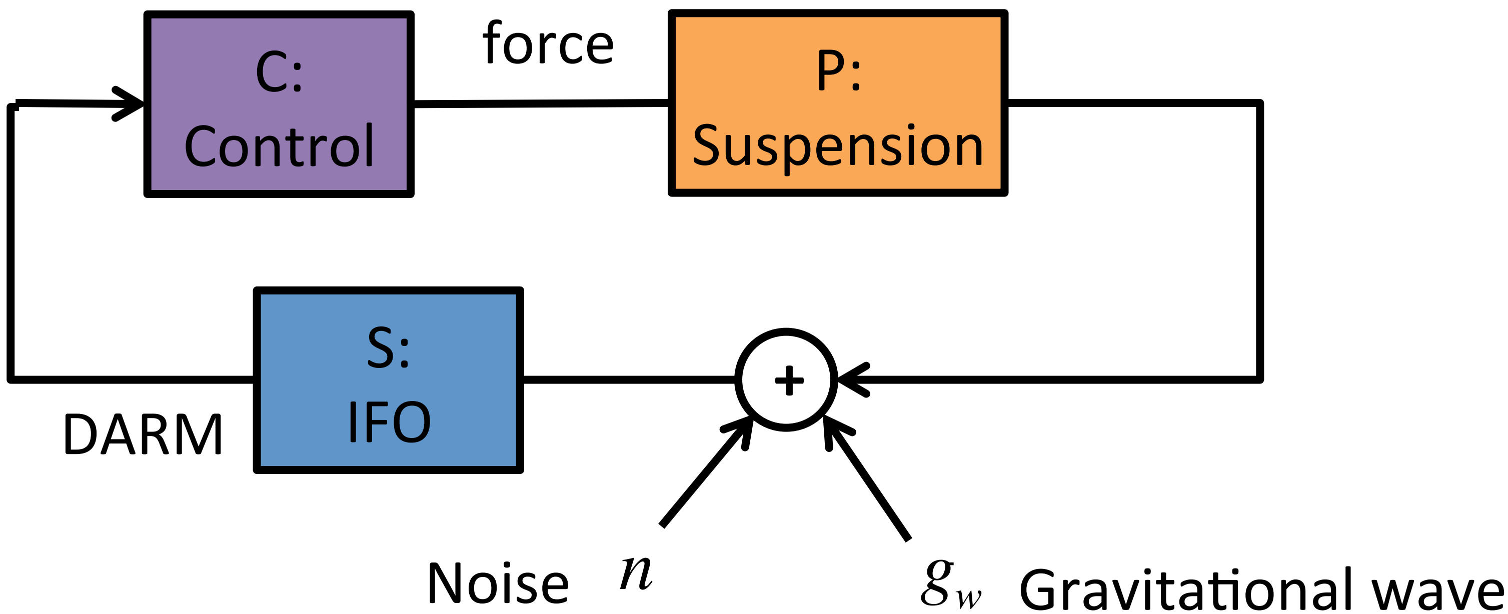


- LHP Poles: -180 deg phase, $1/f^2$ mag
- LHP Zeros: +180 deg phase, f^2 mag

1 Hz complex pairs, 2 degrees off the imaginary axis.

Extracting the GW signal

More detail in G1600412

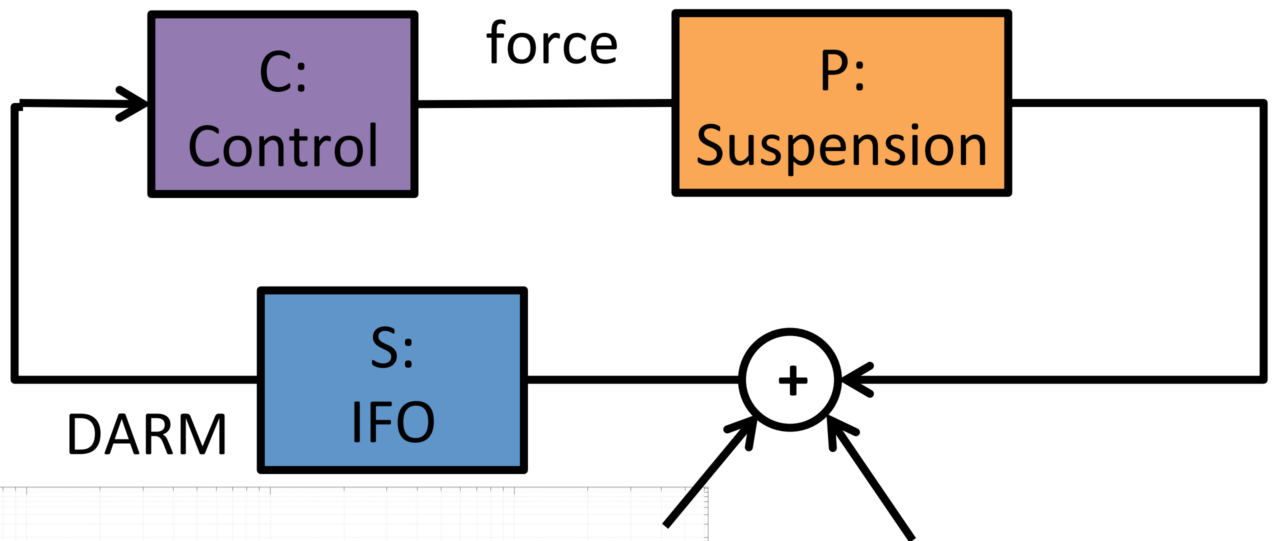


$$DARM = \frac{S}{1+CPS}(n+g_w) \qquad force = \frac{CS}{1+CPS}(n+g_w)$$

The GW exists in both signals. For large gains, DARM -> 0, while the force does not. However, the signal to noise is the same, so both are equally good.

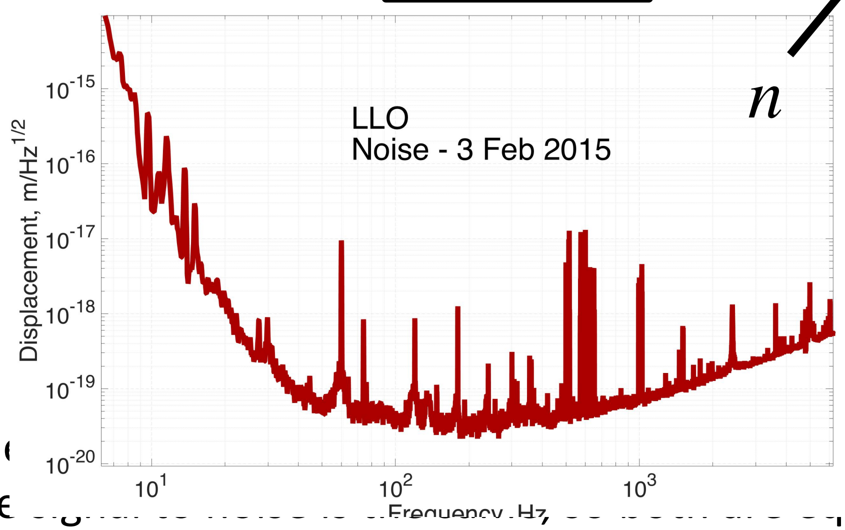
Extracting the GW signal

More detail in G1600412



g_w Gravitational wave

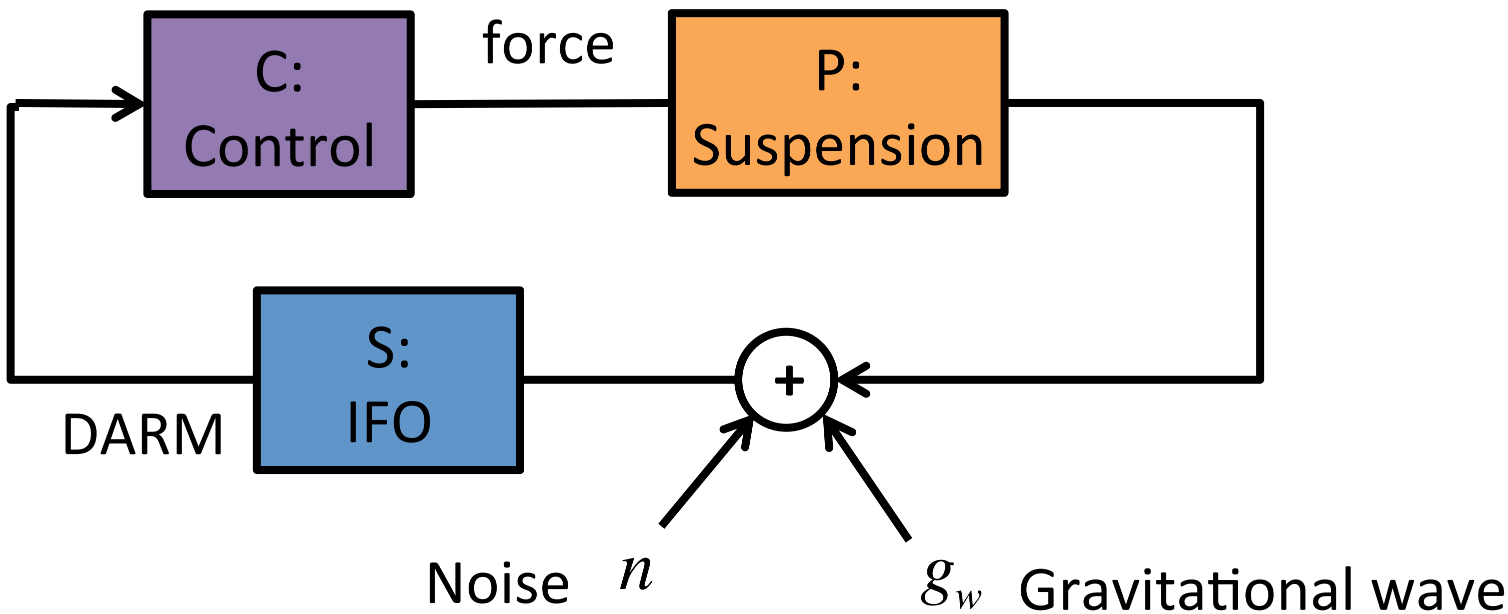
$$\ldots rce = \frac{CS}{1+CPS}\left(n+g_w\right)$$



Th… …RM -> 0, while the force does not. However, the … / good.

Extracting the GW signal

More detail in G1600412



$$DARM = \frac{S}{1+CPS}\left(n + g_w\right)$$

$$g_w \approx DARM \frac{1+\hat{C}\hat{P}\hat{S}}{\hat{S}}$$

If noise is small enough!

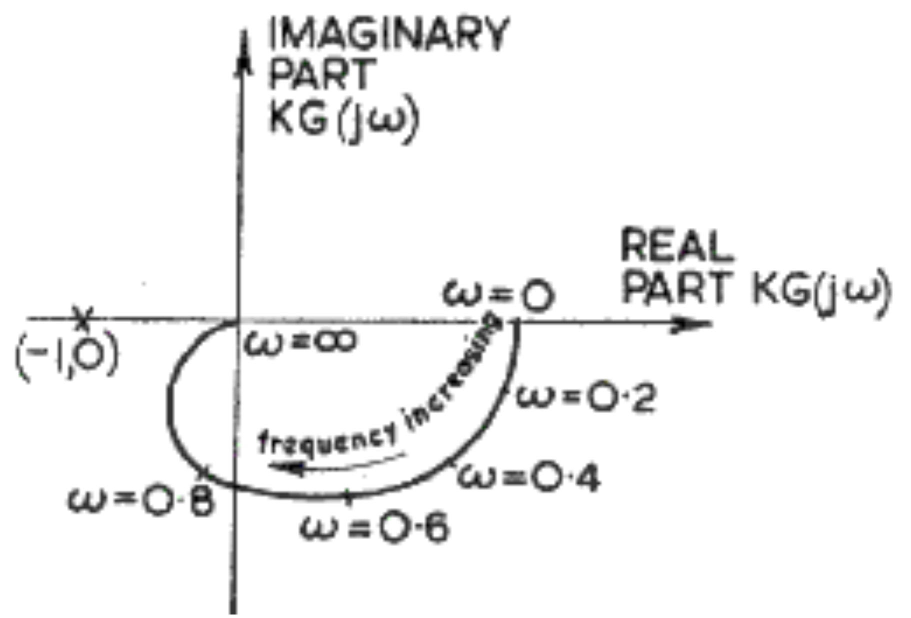
Hat indicates a system model

Nyquist plot

- These plots are traditionally shown over Nichols plots,

but are harder to look at since they can't be put in logspace.

- Stability is achieved by not circling the -1 point



- If a plant has unstable poles, then the rules change. See Ogata text.

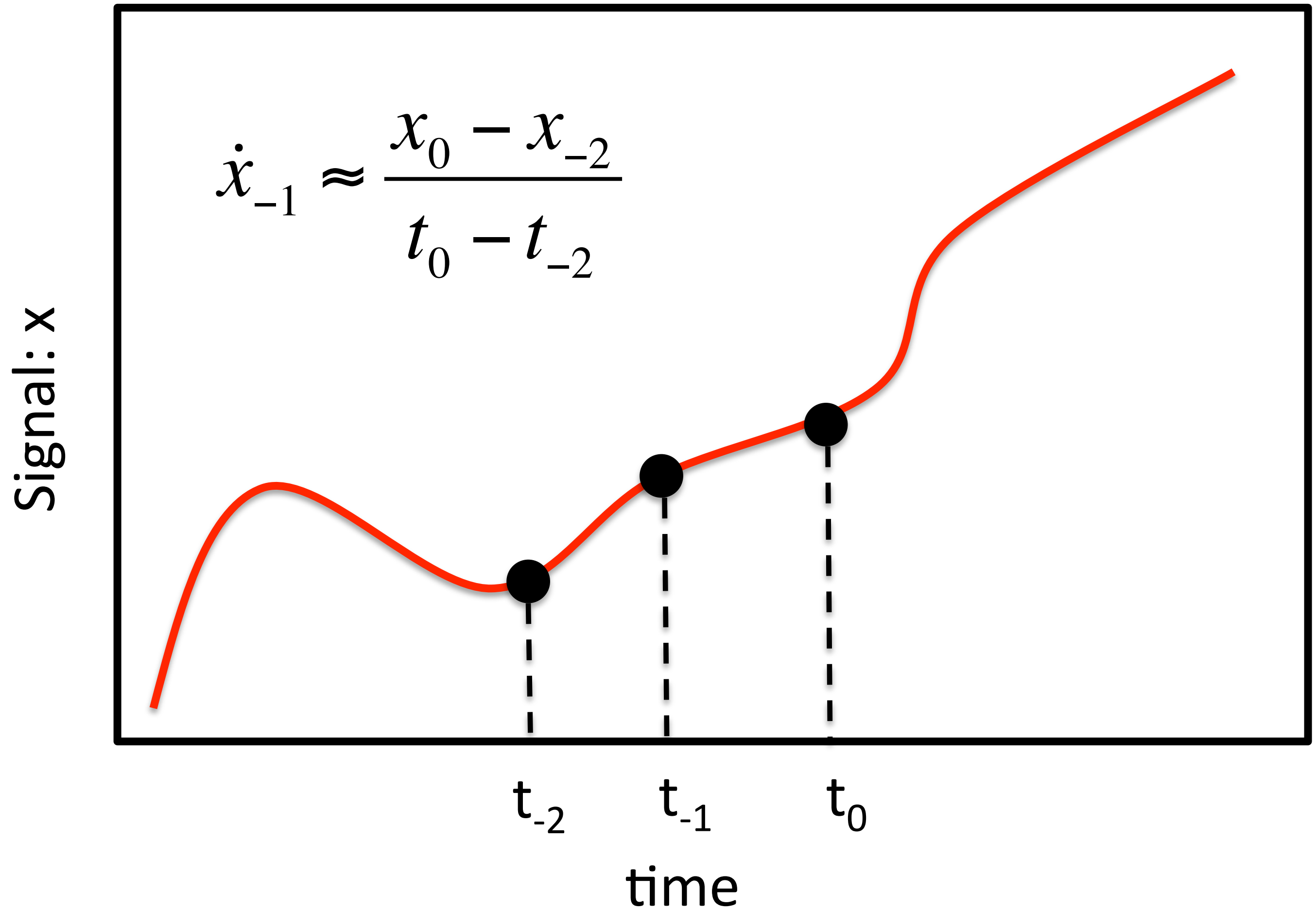
Closed Loop TFs

$$x = \boxed{\frac{1}{1 + P_a C}} P_g x_g$$

S

S is called the 'sensitivity' TF and is common to all closed loop TFs in a loop

Another explanation for causality



A zero-pole pair (where the pole is at higher freq) is like a derivative approximation, where the pole determines the effective sampling time. Deleting the pole is the same as setting it to infinite frequency, which makes the time step = 0, which means we'd effectively be seeing the future by knowing the slope instantaneously.