Gisselquist Technology, LLC

Digital PLLs: A Tutorial in Slides

Daniel E. Gisselquist, Ph.D.

Topics

- Basic Theory
- Type One DPLLs (Phase tracking)
- DPLL Analysis
- Type Two DPLLs (Phase and Frequency tracking)

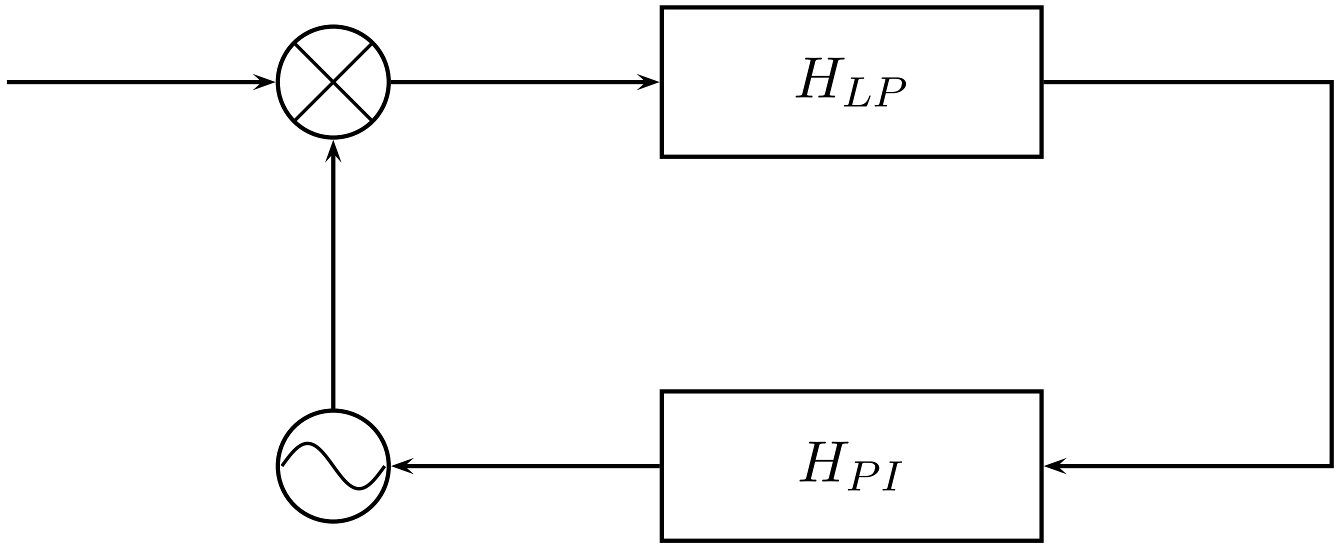
Basic Theory

PLL Theory

Purpose: Why this section?

- Loop Structure
- Small Sine approximation
- Loop Transforms
- Error Signals
- Lock Indication

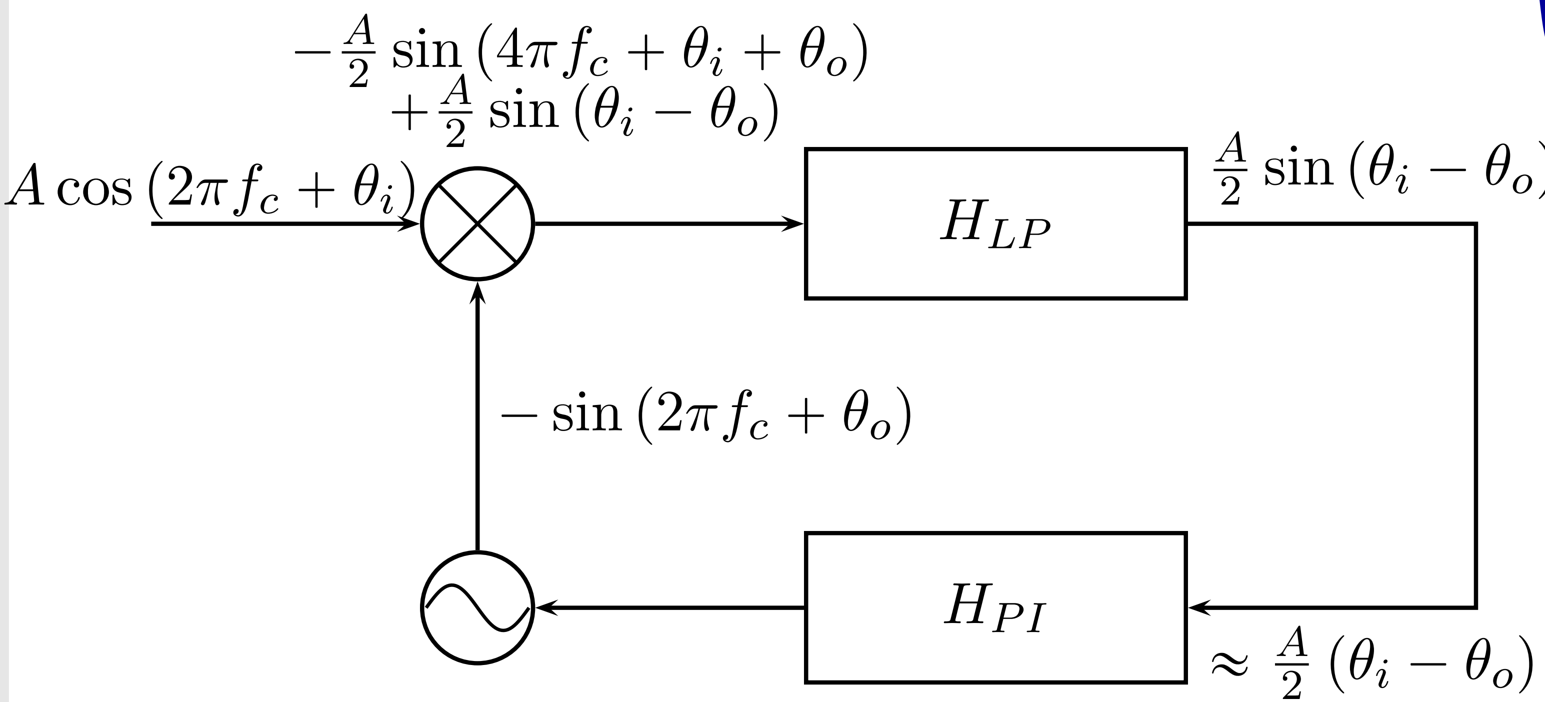
Block Diagram



This is the basic block diagram of any PLL.

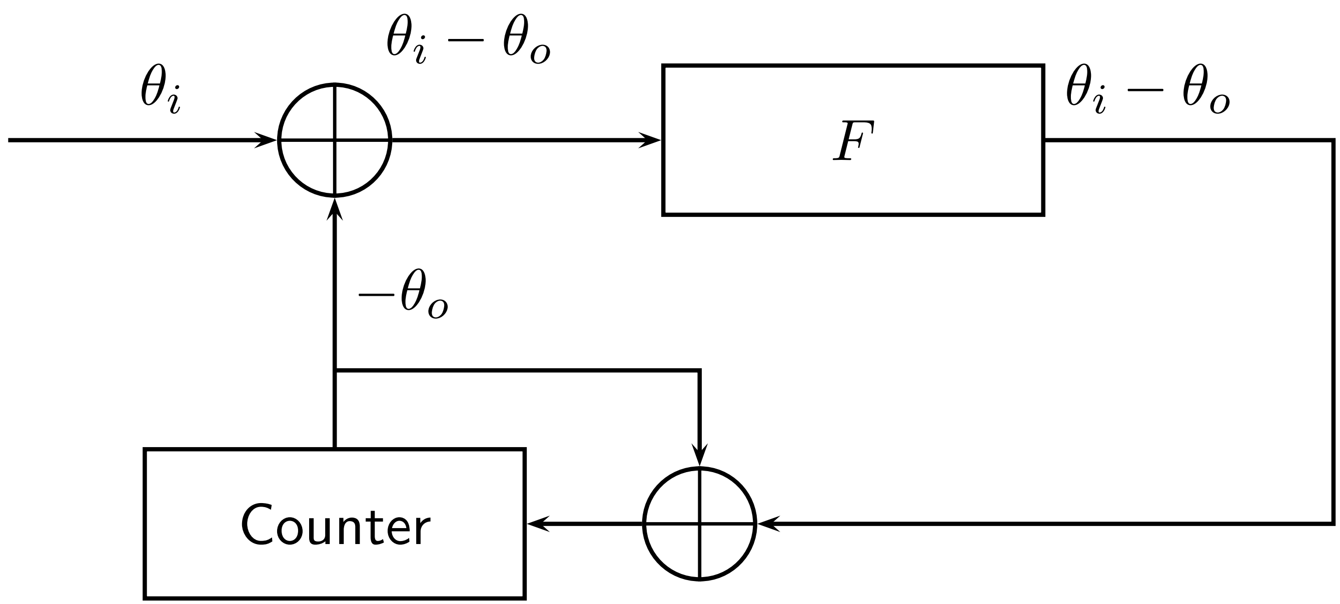
- There's a phase detector, $\bigotimes$
- A Loop filter, $F(z) = H_{LP}(z)\, H_{PI}(z)$, and
- A Numerically Controlled Oscillator (NCO).

Signal Structure



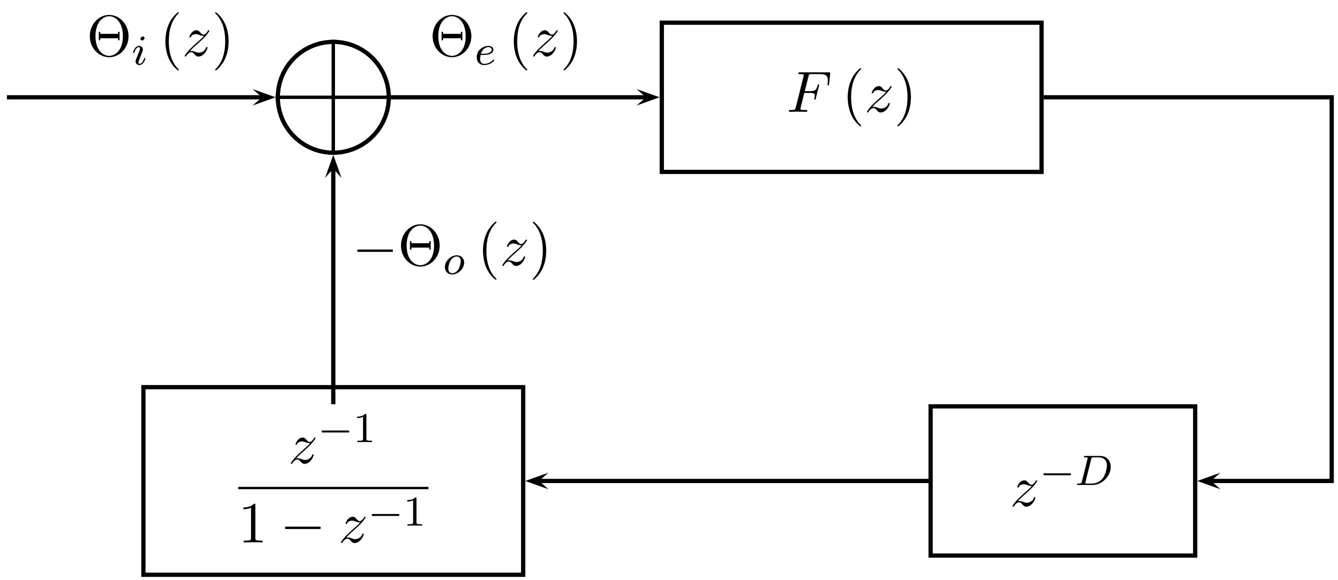
- A PLL is fundamentally a non-linear system
- Its performance is determined by the
  - Loop filter, $F(z) = H_{LP}(z)\, H_{PI}(z)$, and
  - The state of the NCO
- If $\frac{A}{2}\left(\theta_i - \theta_o\right) \approx 0$, then this system is approximately linear

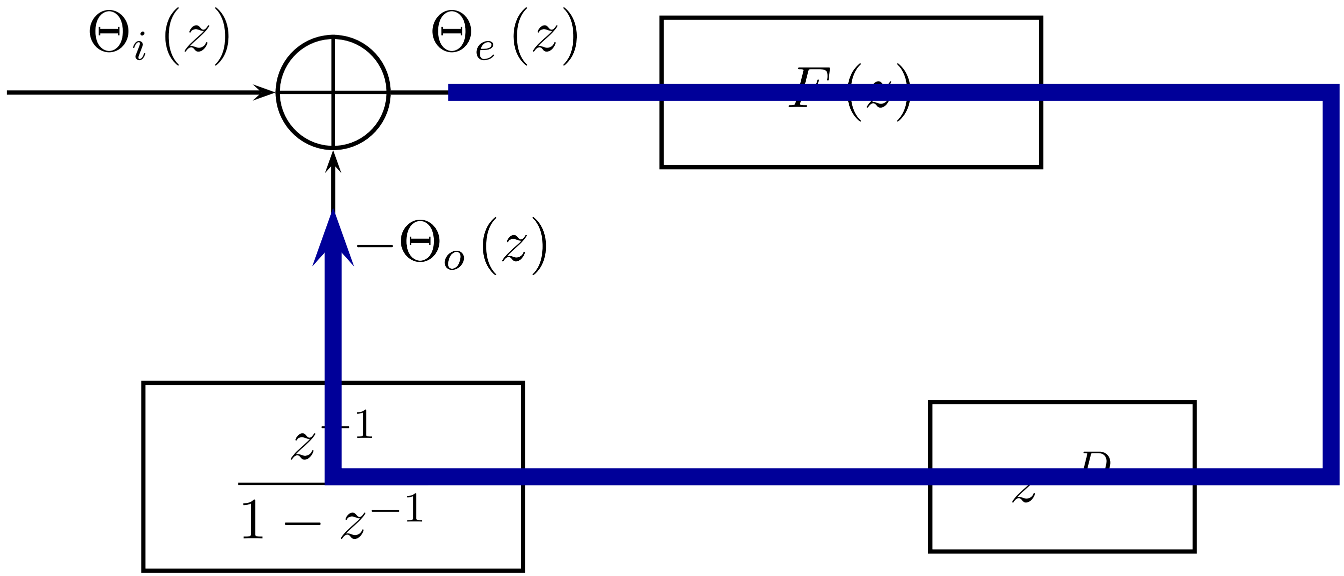
Linearization



- We'll make the small angle assumption throughout
- The entire loop may then be approximated as a linear system

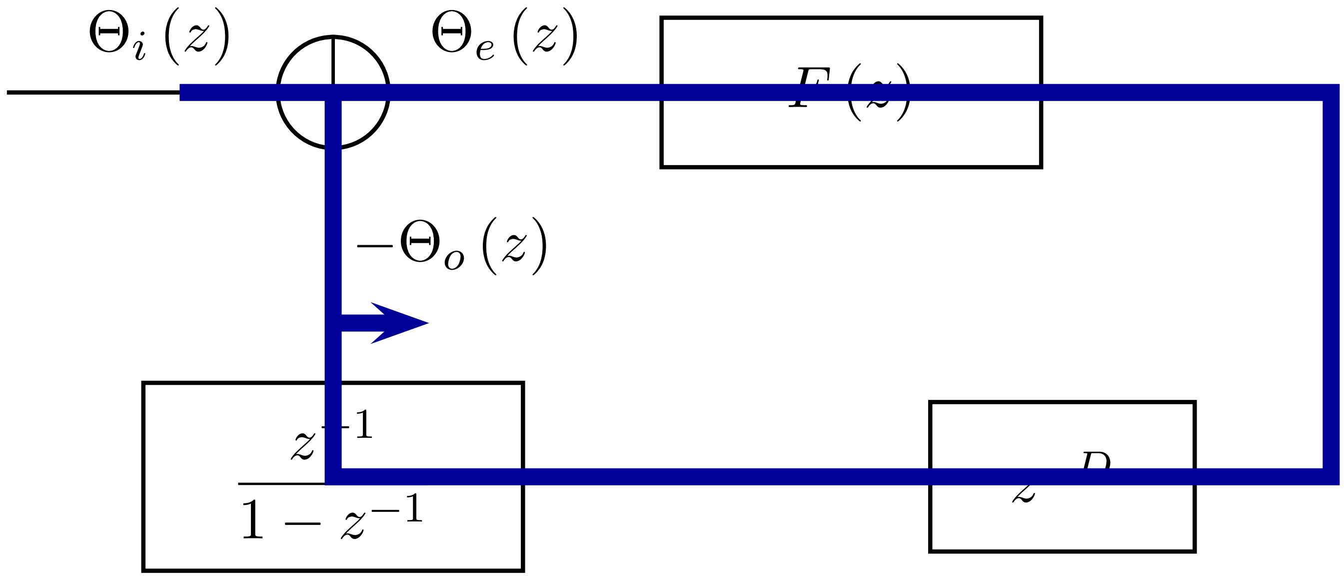
z–Transform





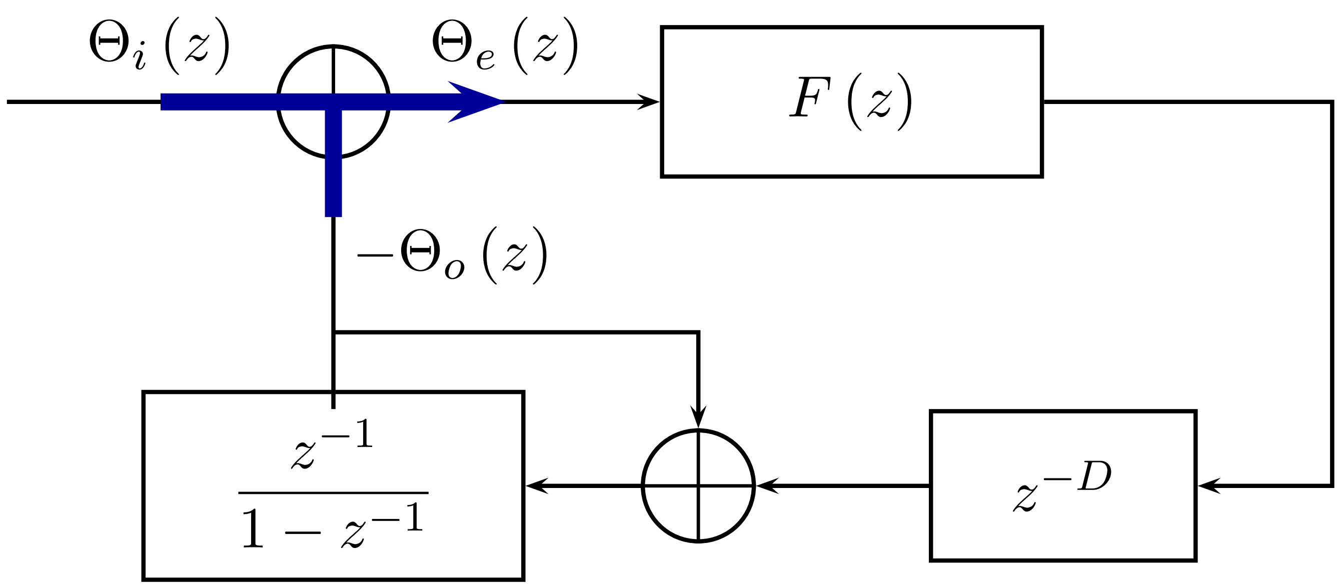
Open Loop Transfer Function

$$G(z) = \frac{\Theta_o(z)}{\Theta_e(z)} = \frac{\Theta_o(z)}{\Theta_i(z) - \Theta_o(z)} = \frac{z^{-1}}{1-z^{-1}} z^{-D} F(z)$$



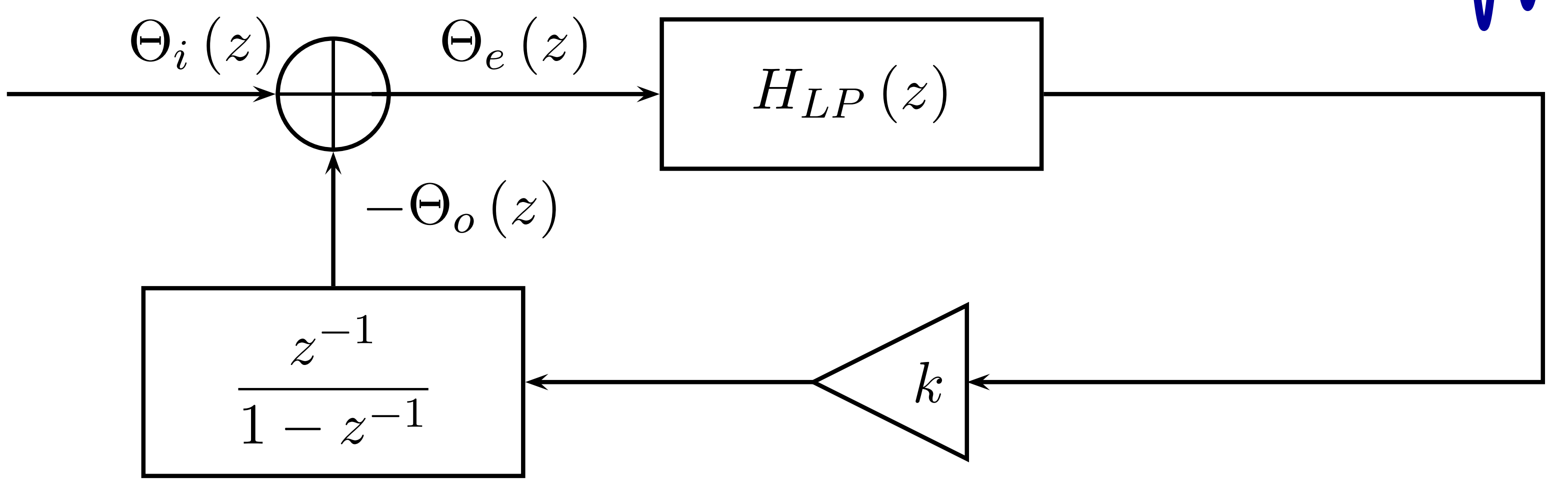
$$
\begin{aligned}
\Theta_o(z) &= G(z)\,\Theta_e(z) = G(z)\left(\Theta_i(z) - \Theta_o(z)\right) \\
\Theta_o(z)\,(1+G) &= G(z)\,\Theta_i(z) \\
H(z) &\triangleq \frac{\Theta_o(z)}{\Theta_i(z)} = \frac{G(z)}{1+G(z)}
\end{aligned}
$$

Error Function



$$E(z) \triangleq \frac{\Theta_e(z)}{\Theta_i(z)} = \frac{\Theta_i(z) - \Theta_o(z)}{\Theta_i(z)} = \frac{1}{1+G(z)}$$

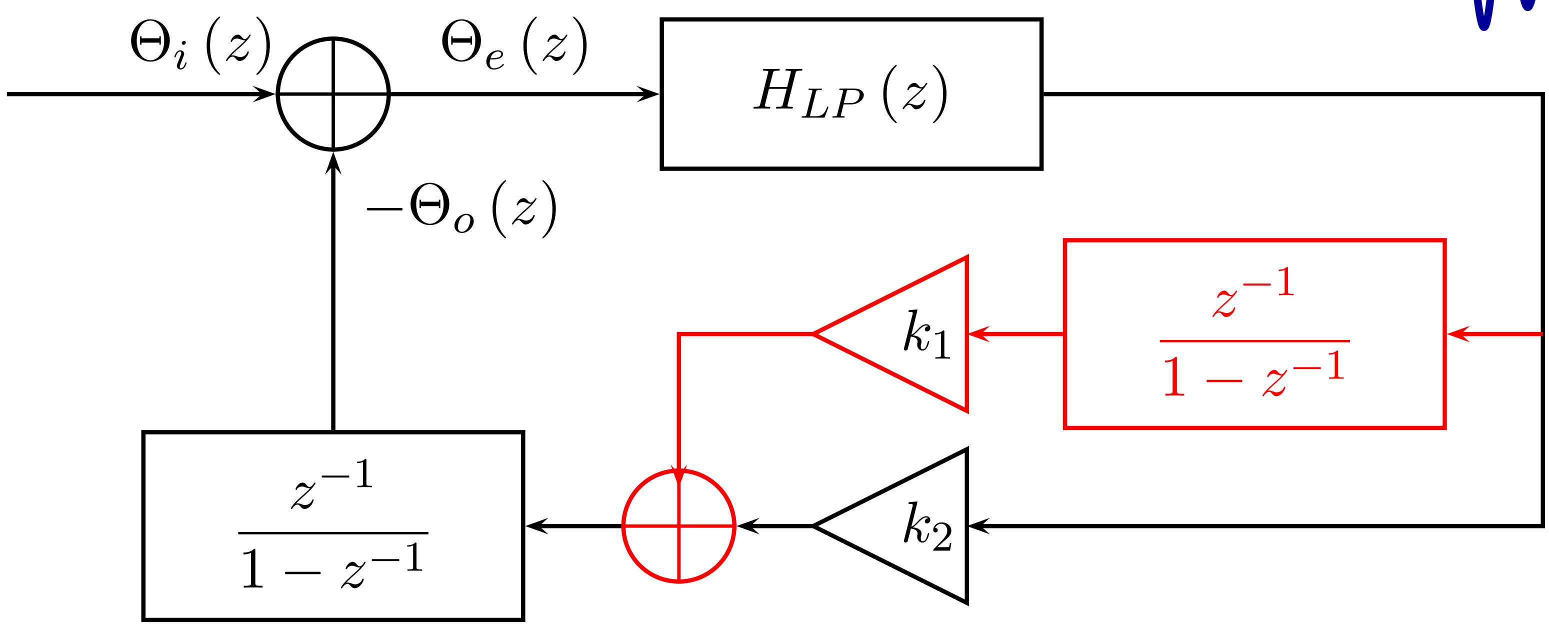
Type One DPLL



If you ignore the filter, $H_{LP}(z)$, this becomes:

```
always @(posedge i_clk)
        phase <= phase + (err * k);
```

Type Two DPLL



Keeping track of frequency requires a second integrator,

```
always @(posedge i_clk)
        step <= step + (err * k1);

always @(posedge i_clk)
        phase <= phase + (err * k2) + step;
```

Characteristic Equation

The equation, $1 + G(z) = 0$ is known as the *characteristic equation*.

Characteristic Polynomial

$G(z)$ can usually be expressed as a rational polynomial, $\frac{P(z)}{Q(z)}$. $H(z)$ is then $\frac{P(z)}{Q(z) + P(z)}$. The polynomial in the denominator, $Q(z) + P(z)$, is called the *characteristic polynomial*.

DPLL Order

The order of the polynomial in the denominator of $H(z)$, $Q(z) + P(z)$, is the *order* of the DPLL.

Error Signals

- Impulse Response

$$h[n] = \mathcal{Z}^{-1}\{H(z)\}$$

How long does the filter ring following a *single error?*

- Unit Step Response

$$\theta_s[n] = \mathcal{Z}^{-1}\left\{H(z)\frac{1}{1-z^{-1}}\right\}$$

How long does the filter take to track a change in *phase?*

- Frequency Step Response

$$\theta_f[n] = \mathcal{Z}^{-1}\left\{H(z)\frac{1}{(1-z^{-1})^2}\right\}$$

How well does this loop track a change in *frequency?*

We'll see more of these later . . .

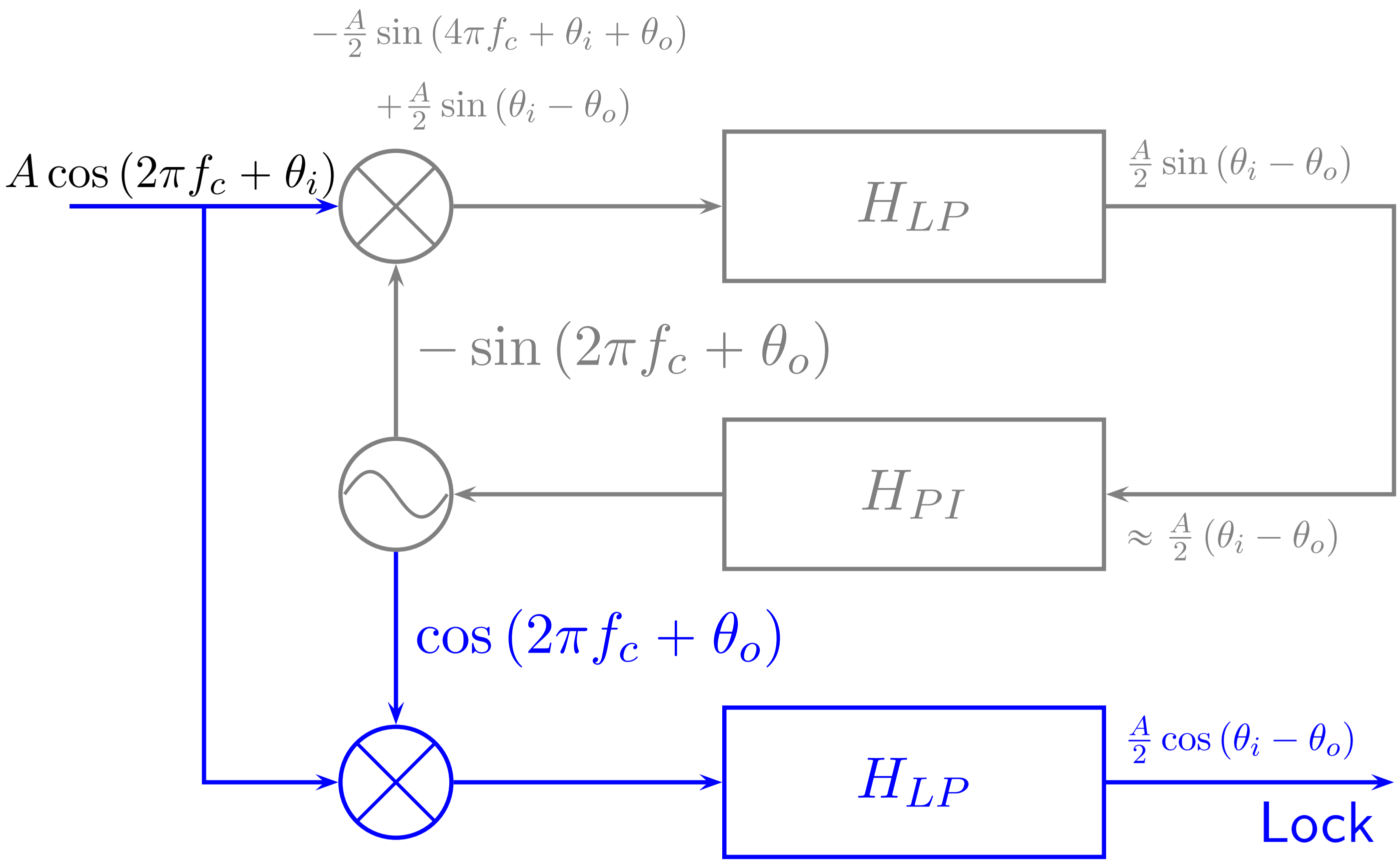
Noise Bandwidth

- In white noise, the PSD is constant, $S_n\left(e^{j2\pi f}\right) = S_n$
- Following the filter, the PSD becomes, $S_n\left|H\left(e^{j2\pi f}\right)\right|^2$
- Total output noise power is $S_n \int_0^1 \left|H\left(e^{j2\pi f}\right)\right|^2 df$

$$NBW = \int_0^1 \left|H\left(e^{j2\pi f}\right)\right|^2 df$$

We'll use the Noise Bandwidth (NBW) to compare PLL's later

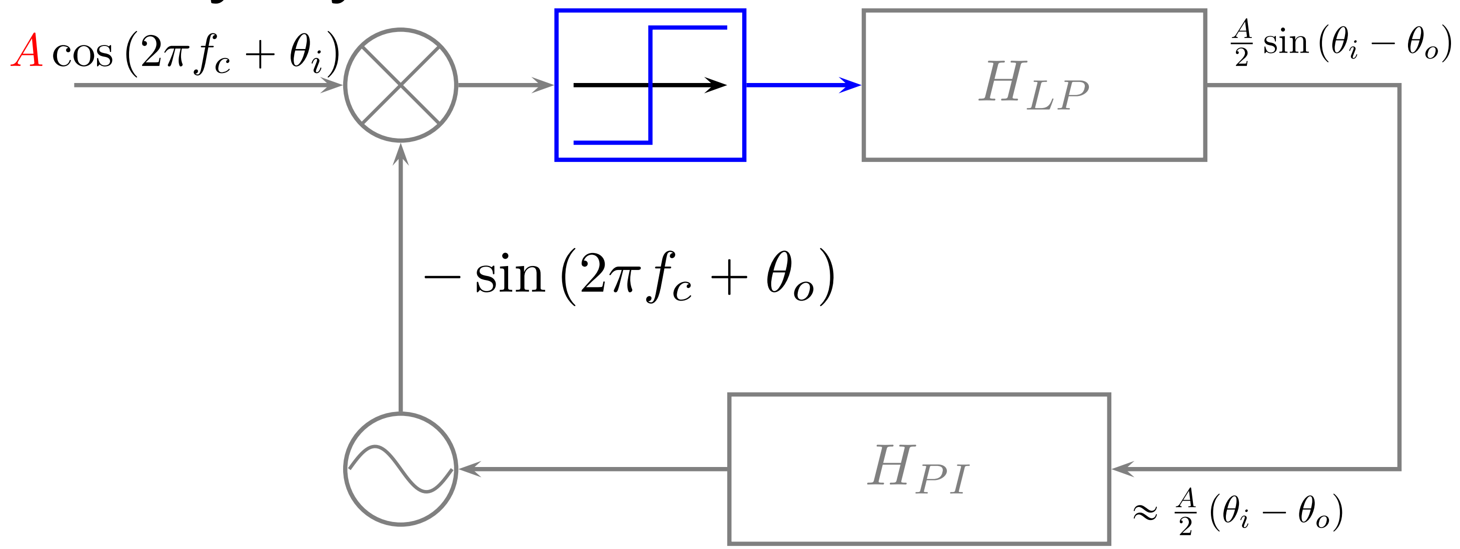
Lock Indication



If $\cos(\theta_i - \theta_o) \approx 1$, the PLL is locked

Gain Invariance

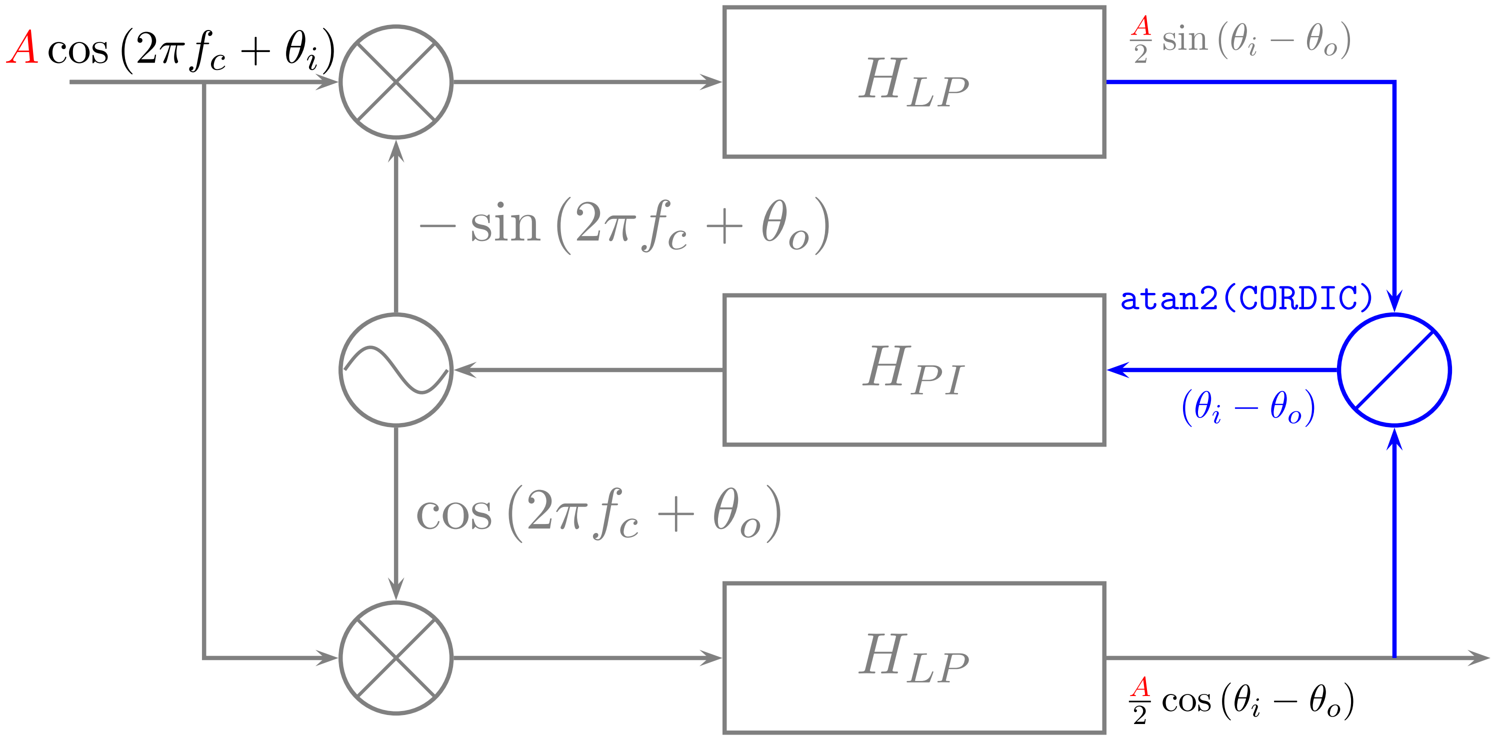
The easy way: Use a limiter



This works quite well, *if* the only input is the sinewave being tracked

Gain Invariance

Harder: Use a CORDIC to calculate `atan2`



- Beware, CORDICs can take many clock cycles
- What happens when no signal is present?

Type One DPLLs

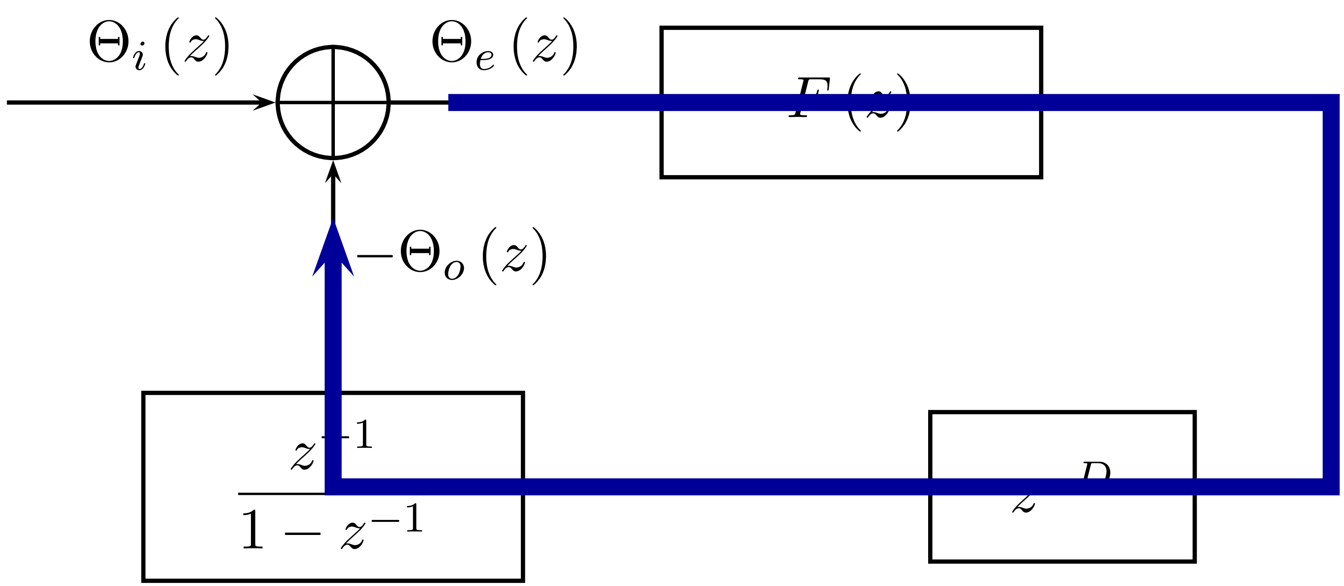
Overview

: **Type One DPLL:** Tracks changing phase, but not changing frequency

- No lowpass filter
- Filter design
  - FIR Filter
  - IIR Filtering
- Achieving critical damping

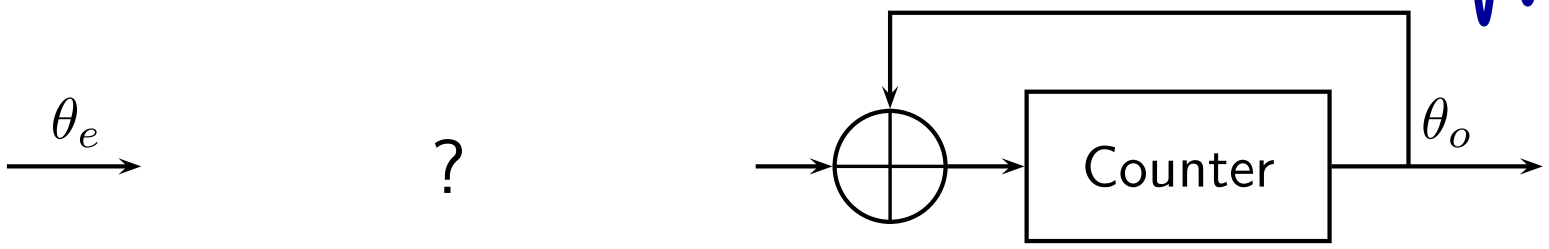
Filter Design

Let's unwrap our loop and discuss the open loop function alone



- $F(z)$ is supposed to be a lowpass filter
- What filter shall we use?

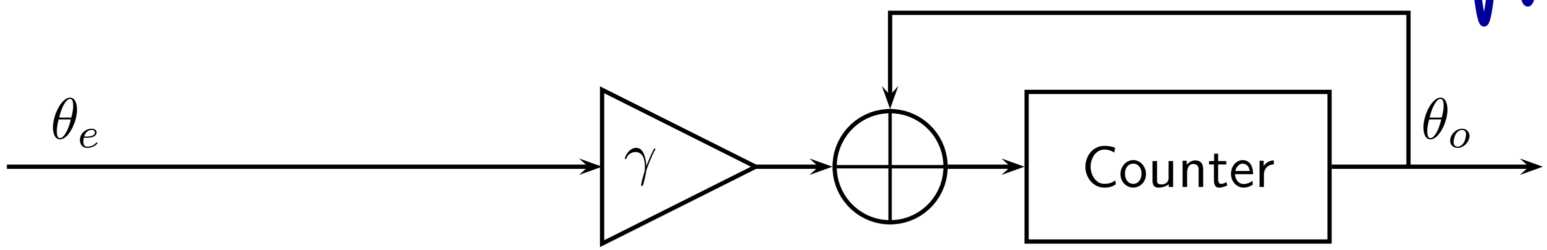
Filter Design



A good filter should be . . .

- Simple and easy to implement
- Robust across circumstances
- Variable/user selectable bandwidth
- With only one knob to tweak!

Scaled Error

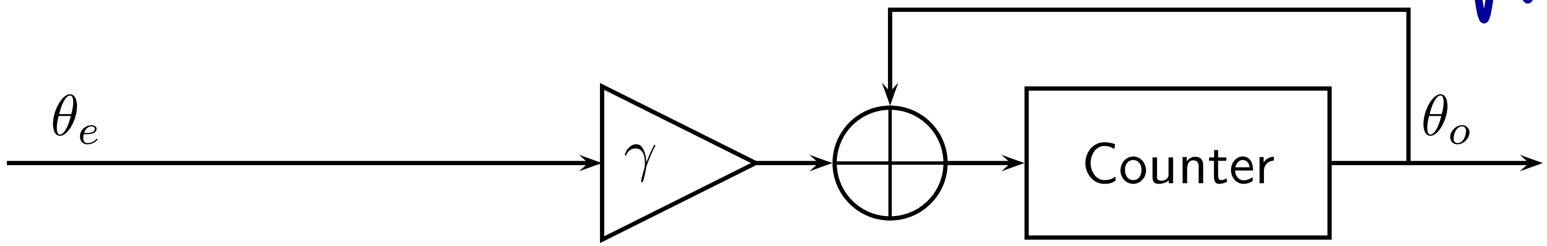


If we pick γ to be a power of two, then this becomes

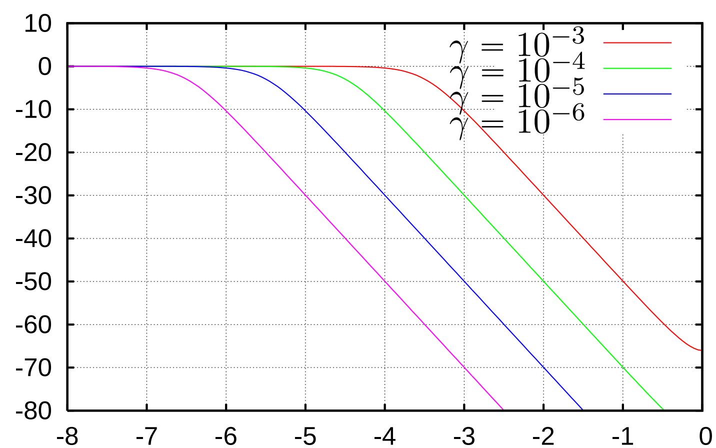
```
always @(posedge i_clk)
        phase <= phase + r_step
                + (error >> lggamma);
```

Easiest to implement, easiest to analyze

Scaled Error



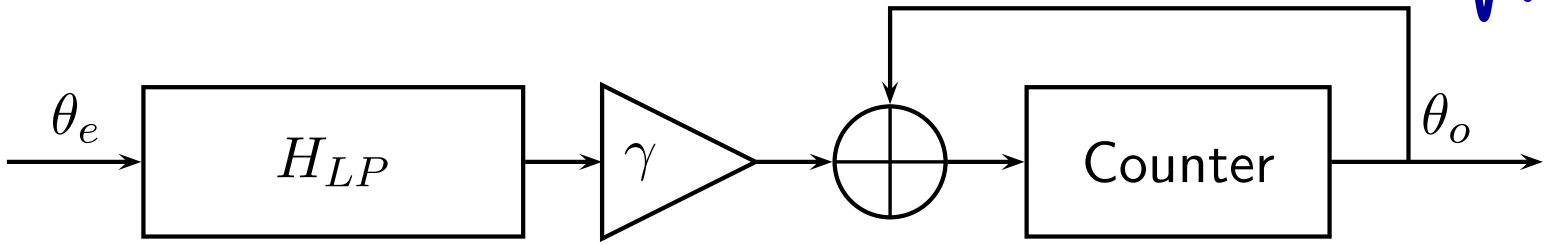
$$G(z) = \gamma \frac{z^{-1}}{1 - z^{-1}}, \; H(z) = \frac{G(z)}{1 + G(z)} = \frac{\gamma z^{-1}}{1 - (1-\gamma) z^{-1}}$$



$$10 \log_{10} \left| H\left(e^{j2\pi f}\right) \right|^2$$

But what about that low-pass filter? We said we needed one to get rid of the high frequency sine product.

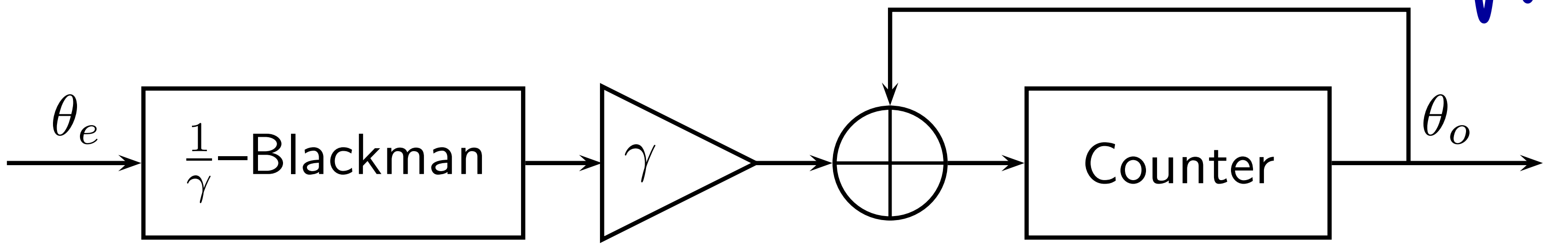
Filter Choices



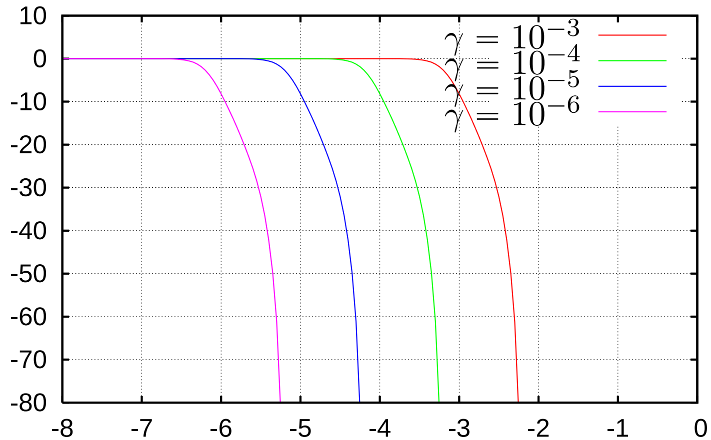
What lowpass filter shall we choose?

- FIR – Linear phase
- FIR – Non-linear phase
- IIR – *My favorite!*

Linear Phase FIR



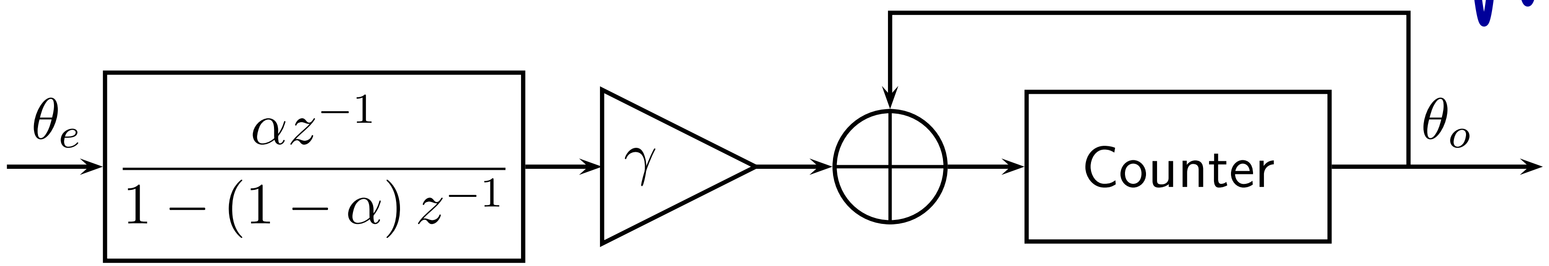
What if we used a blackman window of length $N \approx \frac{1}{\gamma}$?



$$10 \log_{10} \left| H\left(e^{j2\pi f}\right) \right|^2$$

Seems to perform well. It's just an expensive filter. It's also very difficult to implement–especially for small bandwidths.

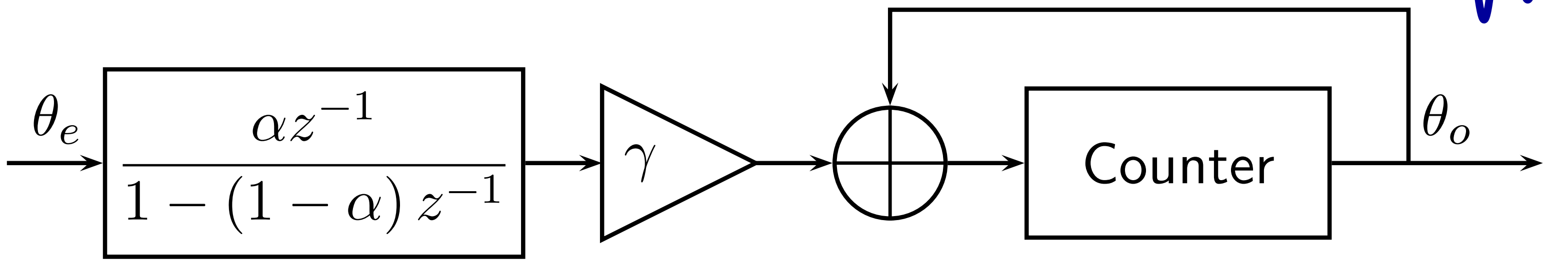
IIR Filter



Above, we use a simple, single pole IIR filter–also known as a recursive averager.

```
always @(posedge i_clk)
        filtered <= filtered
                + (error - filtered) >> lgalpha;

always @(posedge i_clk)
        phase <= phase + r_step
                + (filtered >> lggamma);
```

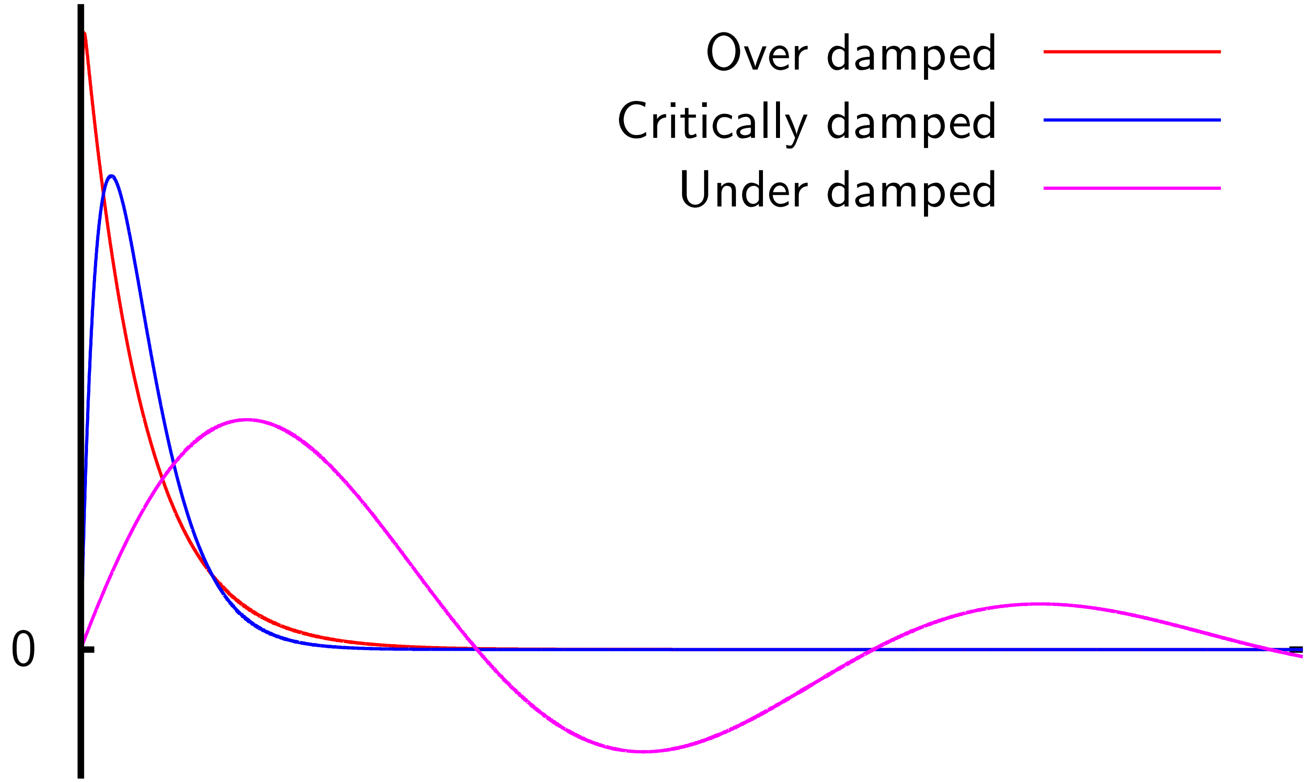
IIR Filter



$$H(z) = \frac{\alpha\gamma z^{-2}}{1-(1-\alpha)z^{-1}+(1-\alpha+\gamma\alpha)z^{-2}}$$

Problem: This filter leaves us with two knobs to tweak. I'd like a simpler filter that has only one knob to tweak. Can we collapse these two into one?

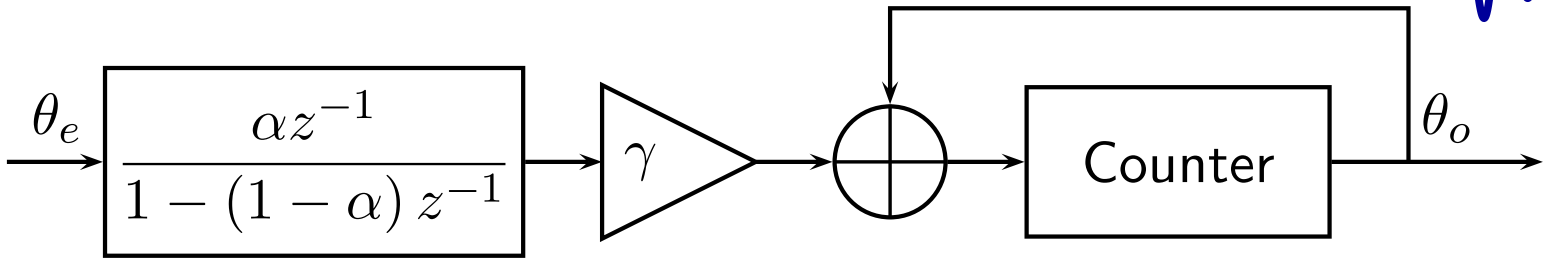
Damping



Critically damped systems

- No overshoot
- No ringing
- Converge faster than all others options

IIR Filter

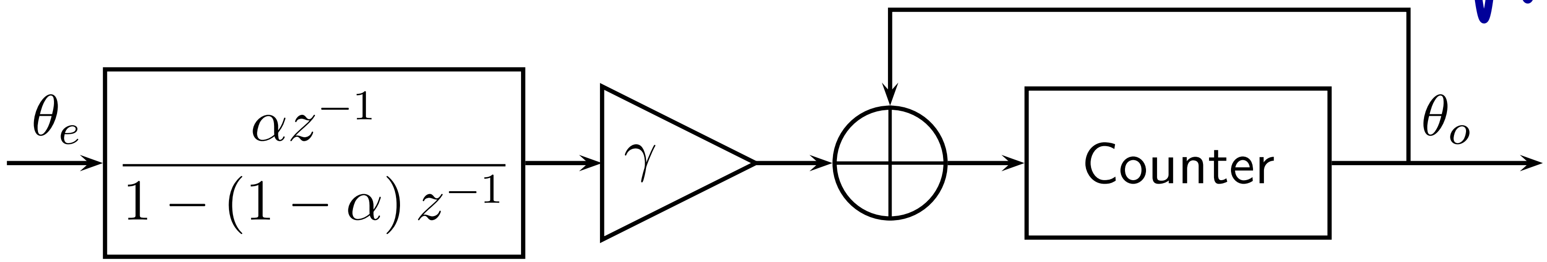


Solution: If $\alpha = 4\gamma$, the system will be critically damped

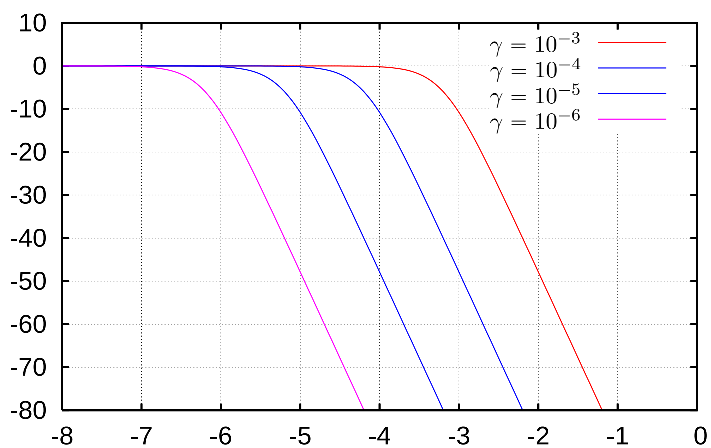
$$H(z) = \frac{4\gamma^2 z^{-2}}{\left[1-(1-2\gamma)\,z^{-1}\right]^2}$$

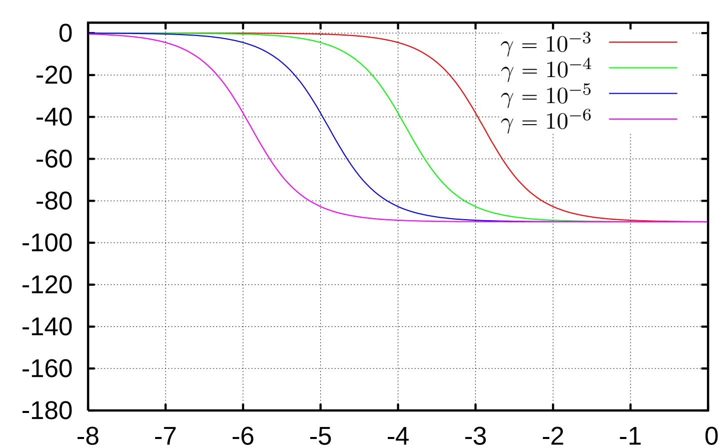
Bonus: We can still get away with shifts and adds alone

IIR Filter



$$H(z) = \frac{4\gamma^2 z^{-2}}{\left[1-(1-2\gamma)z^{-1}\right]^2}$$





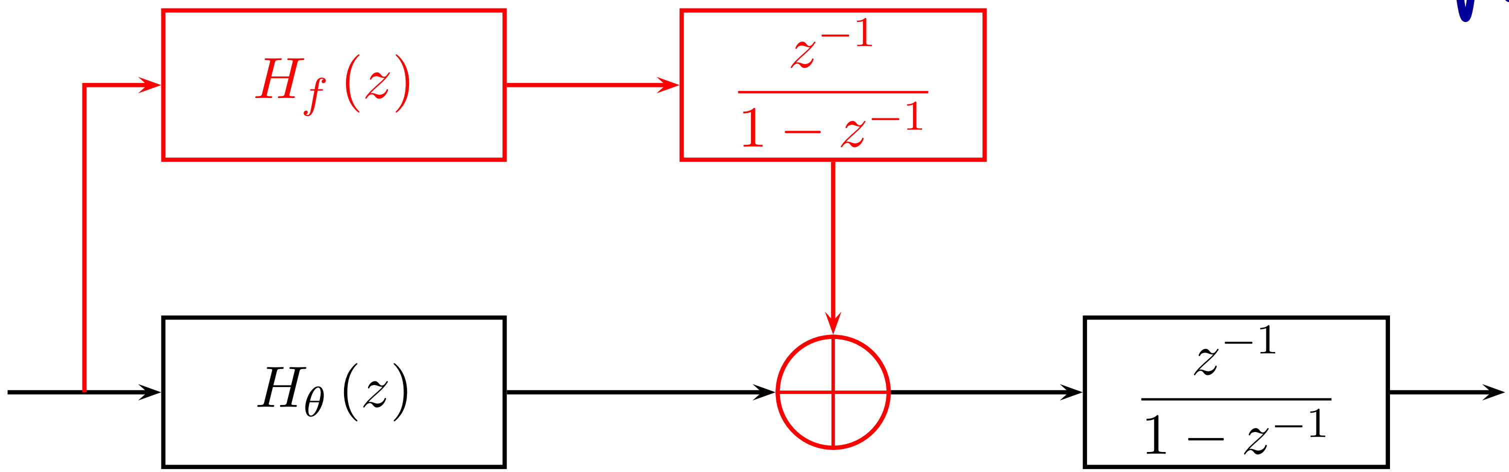
Type Two DPLLs

Overview

Type Two DPLL: Tracks frequency as well as phase, just not the frequency sweep rate

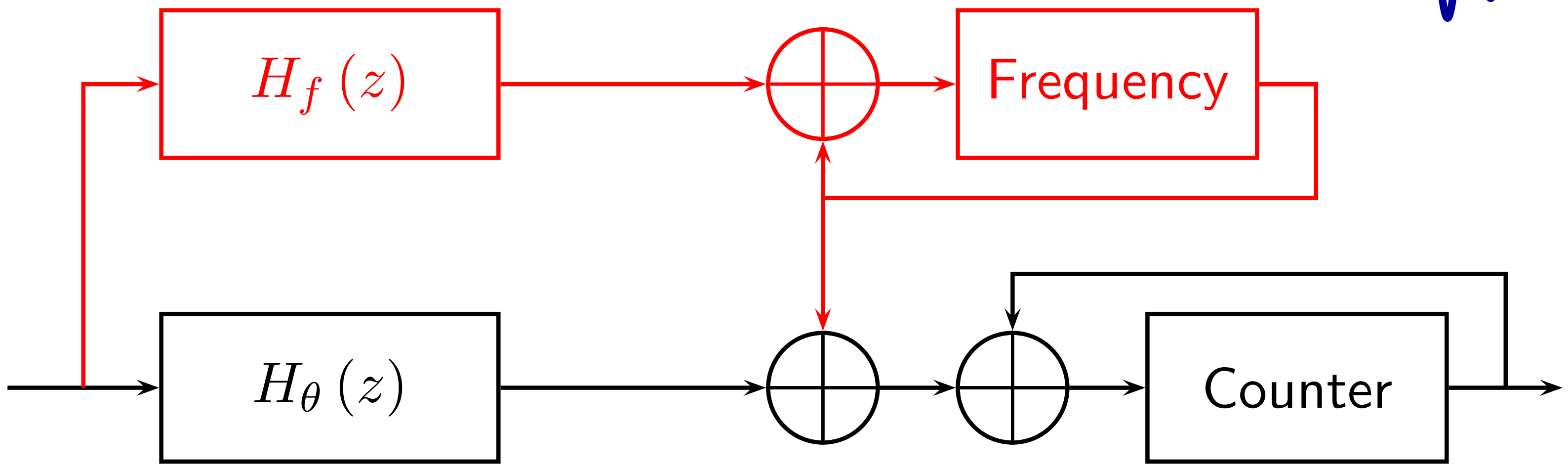
- Basic setup
- Filter design
- Loop performance

Loop Structure



- Differs from the type one DPLL by a frequency accumulator path, shown here in red.
- Unlike the type one DPLL, there are now two filters to specify
  - One to feed the phase tracking circuit, $H_\theta(z)$
  - And now a second one to feed the frequency tracking circuit, $H_f(z)$

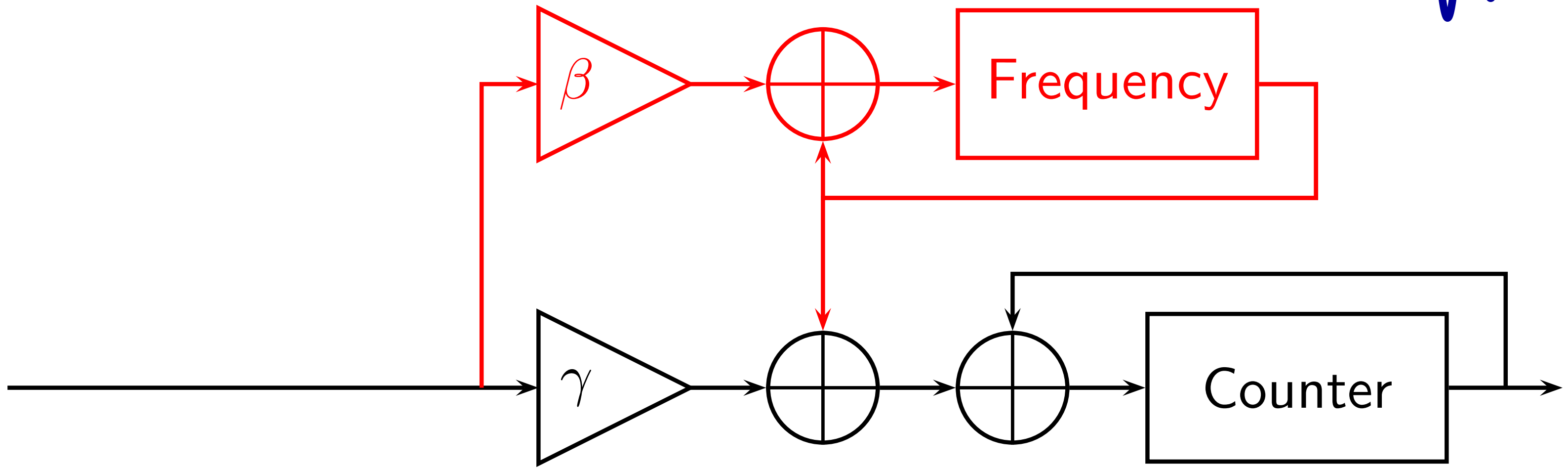
Loop Structure



- When implemented, the frequency accumulator is just another integrator

```
always @(posedge i_clk)
        frequency_step <= frequency_step
                + frequency_filter_output;
always @(posedge i_clk)
        phase <= phase + frequency_step
                + phase_filter_output;
```

Scaled Error



The easiest loop "filters" we might apply are just scale factors.

$$G(z) = \frac{z^{-1}}{1-z^{-1}}\left[\gamma + \beta\frac{z^{-1}}{1-z^{-1}}\right]$$

$$H(z) = \frac{\gamma z^{-1} + (\beta-\gamma)\, z^{-2}}{1-(2-\gamma)\, z^{-1} + (\beta-\gamma+1)\, z^{-2}}$$

Now, given γ, what value shall we choose for β?

Picking β

Let's do what we did the last time, and pick β so that $H(z)$ has two identical poles. These poles will be at,

$$z^{-1} = \frac{2-\gamma}{2} \pm \frac{1}{2}\sqrt{(2-\gamma)^2 - 4(\beta - \gamma + 1)}$$

In order for these poles to be identical, the determinant must be zero,

$$\begin{aligned} 0 &= (2-\gamma)^2 - 4(\beta - \gamma + 1) \\ &= 4 - 4\gamma + \gamma^2 - 4\beta + 4\gamma - 4 \\ \beta &= \frac{\gamma^2}{4} \end{aligned}$$

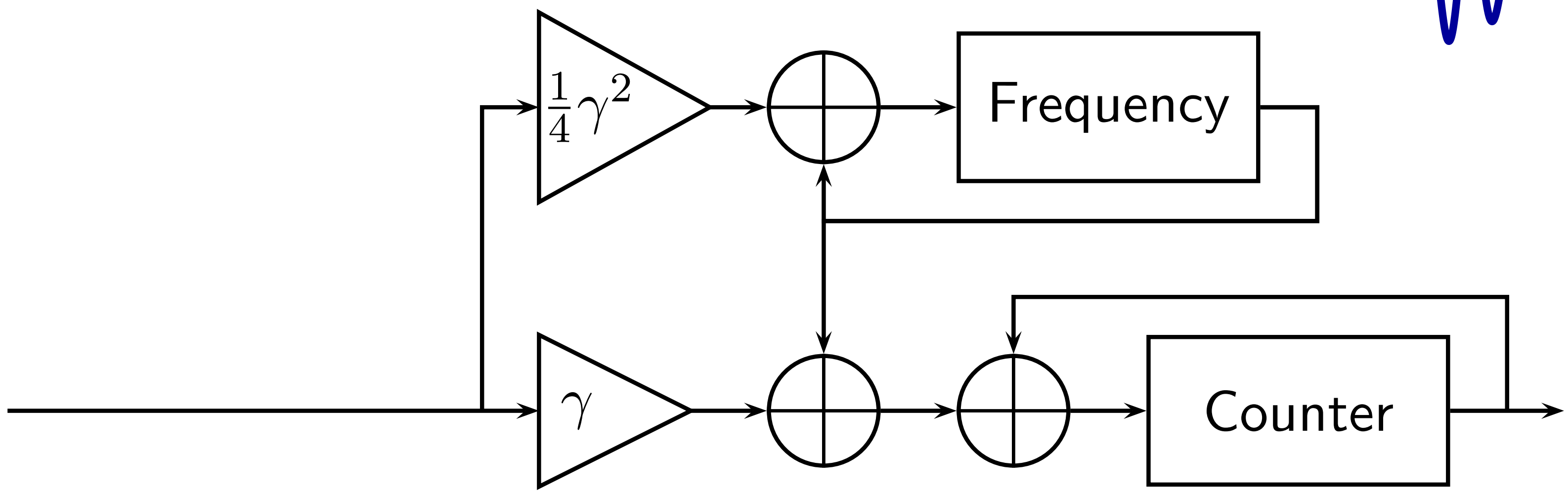
This will place two identical poles at $\left(1 - \frac{\gamma}{2}\right)$

Powers of two

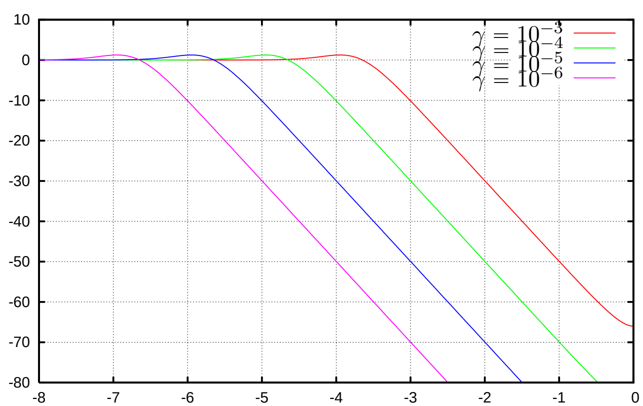
The neat thing about $\beta = \frac{\gamma^2}{2}$ is that both multiplications, by β and γ, can be implemented as pure shift

```
always @(*)
        log_beta = { gamma, 1'b0 } - 2

always @(posedge i_clk)
        frequency_step <= frequency_step
                + (err >> log_beta);

always @(posedge i_clk)
        phase <= phase + frequency_step
                + (err >> log_gamma);
```

Performance

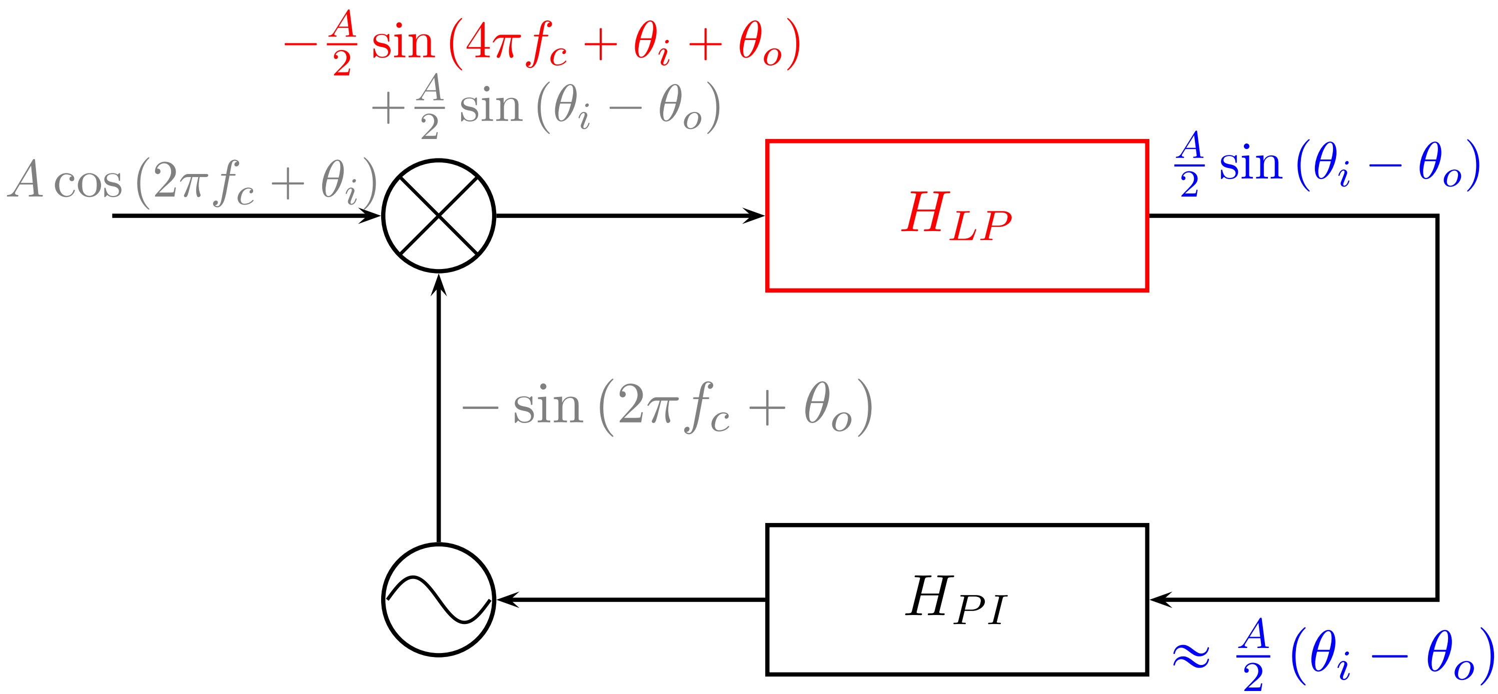


$$H\left(z\right) = \gamma z^{-1} \frac{1-\left(1-\frac{\gamma}{4}\right)z^{-1}}{\left[1-\frac{\gamma}{4}z^{-1}\right]^2}$$



As before, we have left out the lowpass filter.
Let's see what happens if/when we add one.

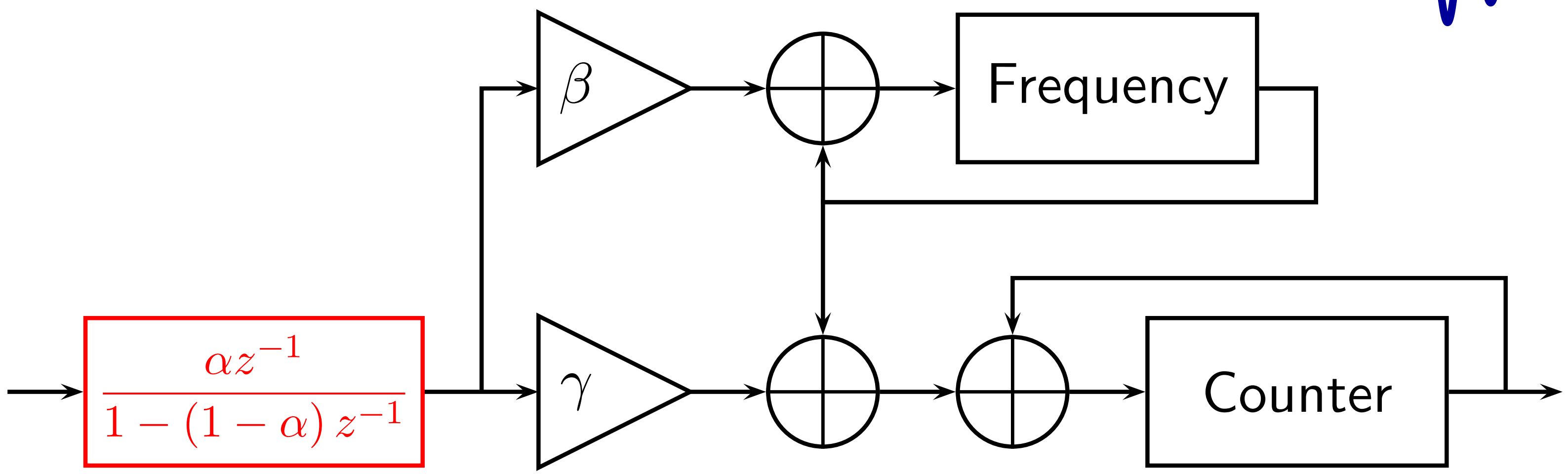
Structure Review



Remember: we made a linearity approximation

- This was valid when the input to our system contained a single component only
- Getting there required a lowpass filter (LPF) to remove the part of the signal found at twice our frequency of interest
- This LPF may also remove or limit other junk in the input

Loop Structure



How about using another single-pole lowpass filter?

$$G(z) = \frac{z^{-1}}{1-z^{-1}}\left[\gamma + \beta\frac{z^{-1}}{1-z^{-1}}\right]\frac{\alpha z^{-1}}{1-(1-\alpha)z^{-1}}$$

Unlike before, we now need to find values for α and β in terms of γ ...

Three poles

If we force $H(z)$ to have three identical poles in its denominator,

- We'll get a solution with only one knob to adjust
- It will converge faster than any other solution with the same gain
- It will have awesome out of band performance
- It will relate α and β to γ

Only, the algebra no longer fits on a slide very well

- It's still quite doable

Solution

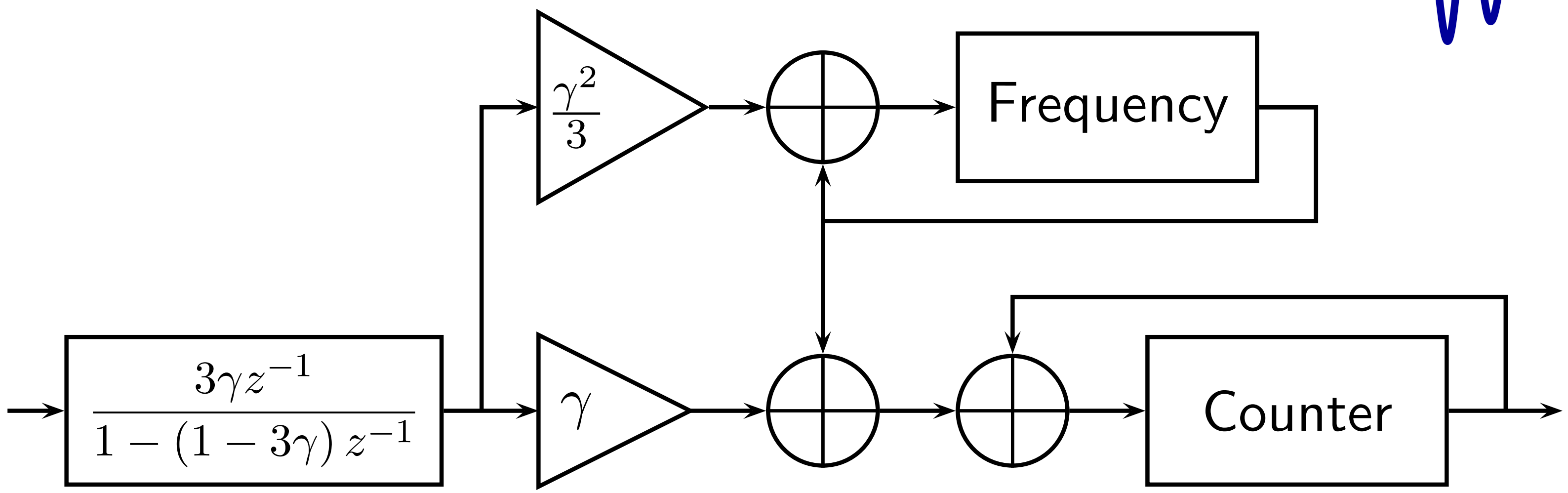
Here's our solution:

$$\begin{aligned}\alpha &= 3\gamma \\ \beta &= \frac{1}{3}\gamma^2\end{aligned}$$

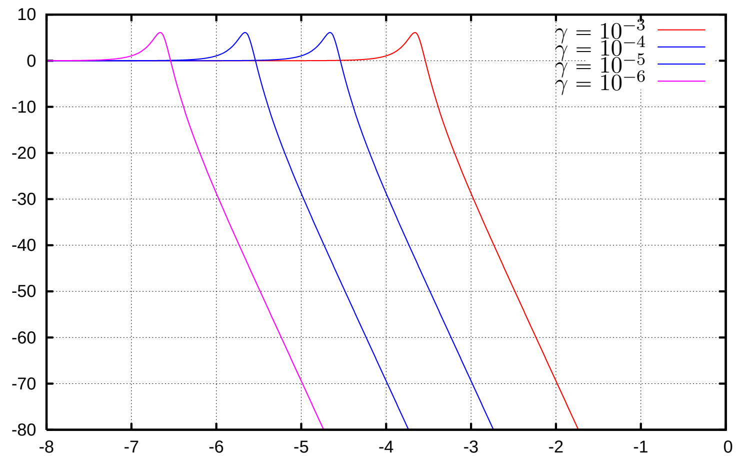
No bonus: these scale constants can no longer be applied with shifts and adds alone

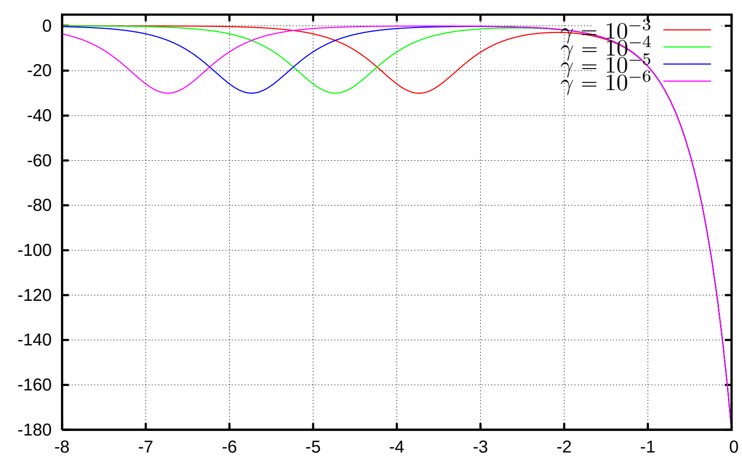
- I don't know of an easy way to implement this in logic
- Dividing by three can be approximated by a multiply and shift
- Multiplication by three can be replaced by a shift and add

Filtered

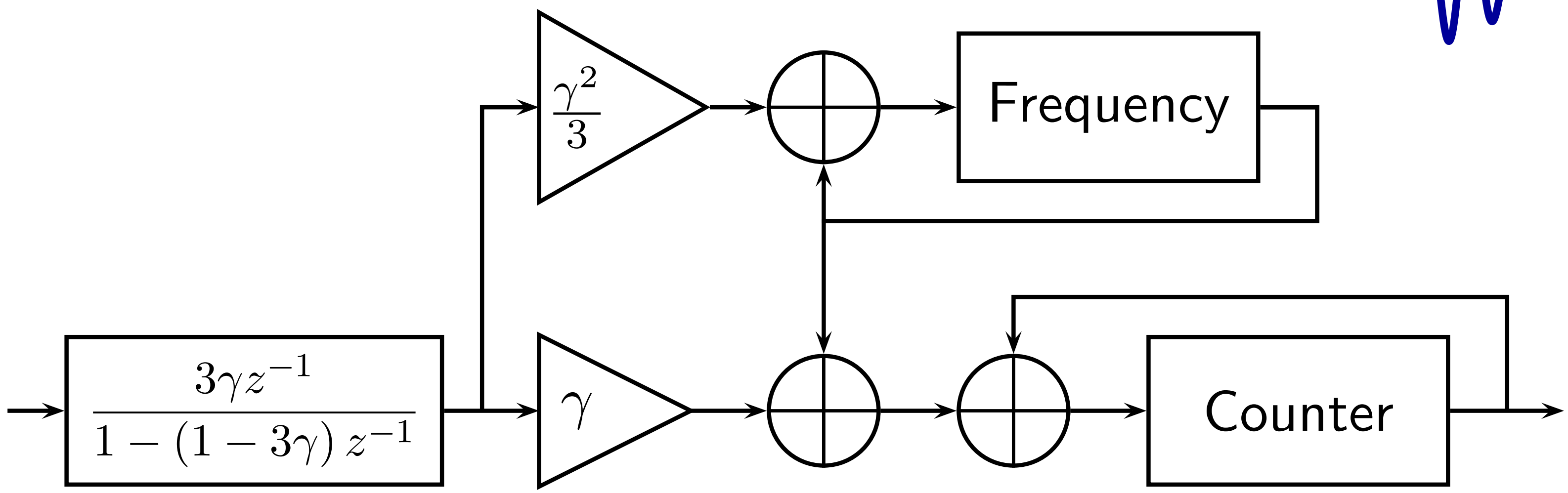


$$H(z) = 3\gamma^2 z^{-2}\frac{1-\left(1-\frac{\gamma}{3}\right)z^{-1}}{\left(1-\gamma z^{-1}\right)^3}$$

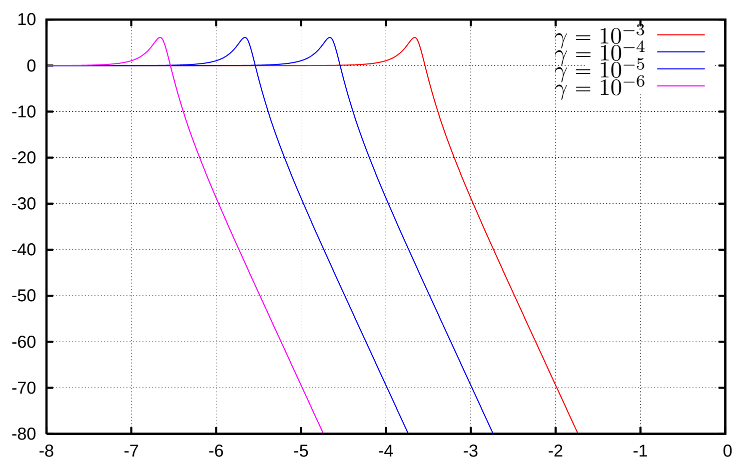




Filtered



$$H(z) = 3\gamma^2 z^{-2}\frac{1-\left(1-\frac{\gamma}{3}\right)z^{-1}}{\left(1-\gamma z^{-1}\right)^3}$$



- Awesome stop-band fall-off
- Harder to implement in logic (Not hard in S/W)
- In-band gain is no longer flat