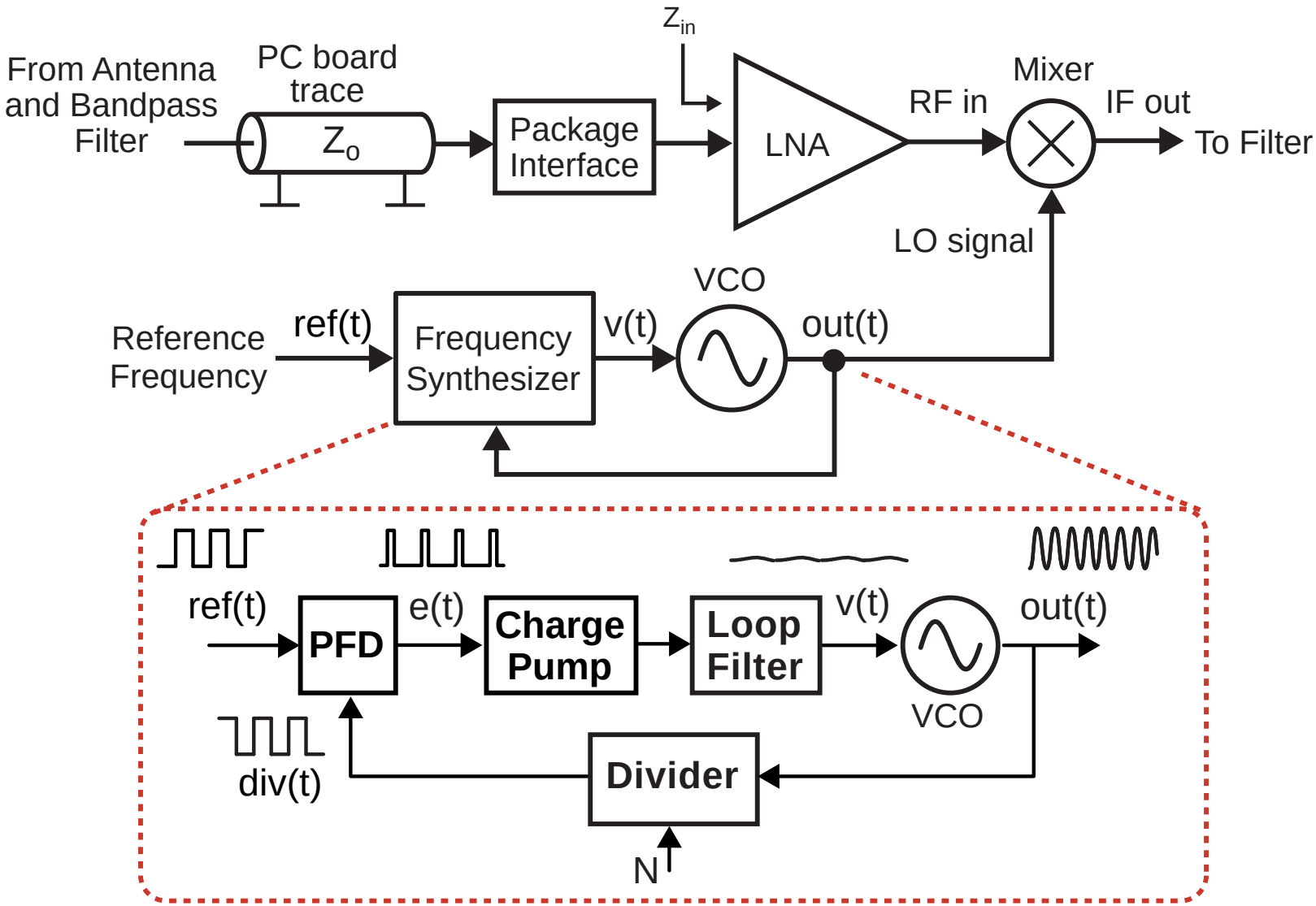
High Speed Communication Circuits and Systems
Lecture 15
Integer-N Frequency Synthesizers

Michael H. Perrott

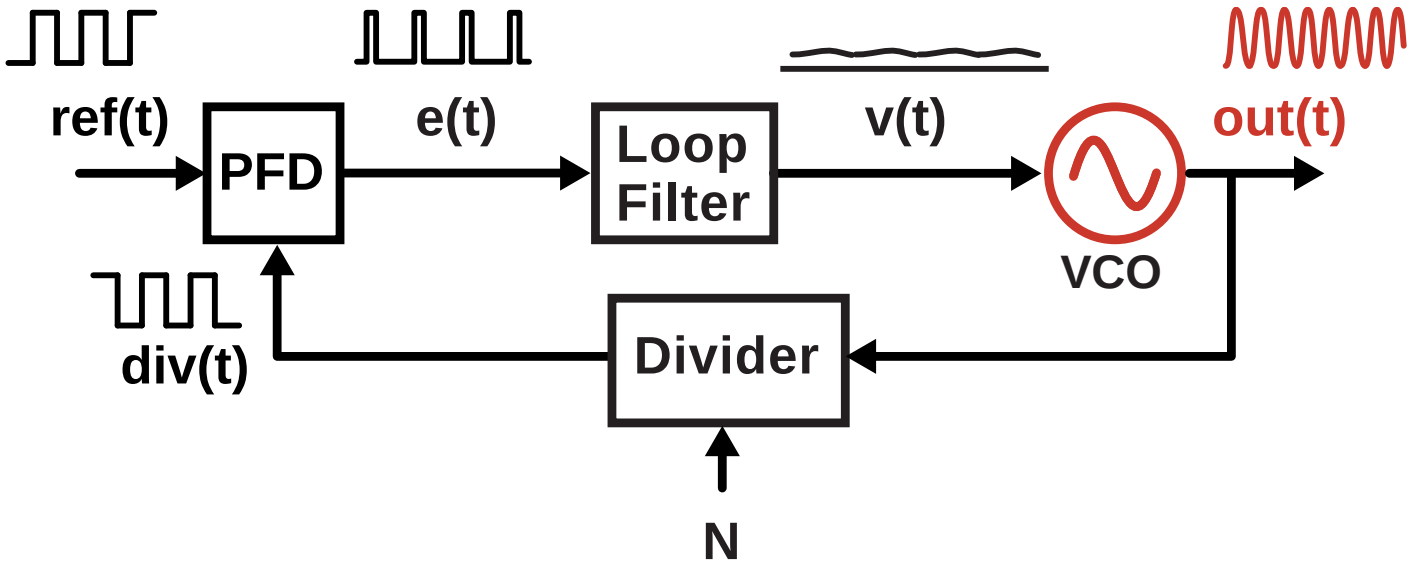
March 31, 2004

Integer-N Frequency Synthesizers in Wireless Systems



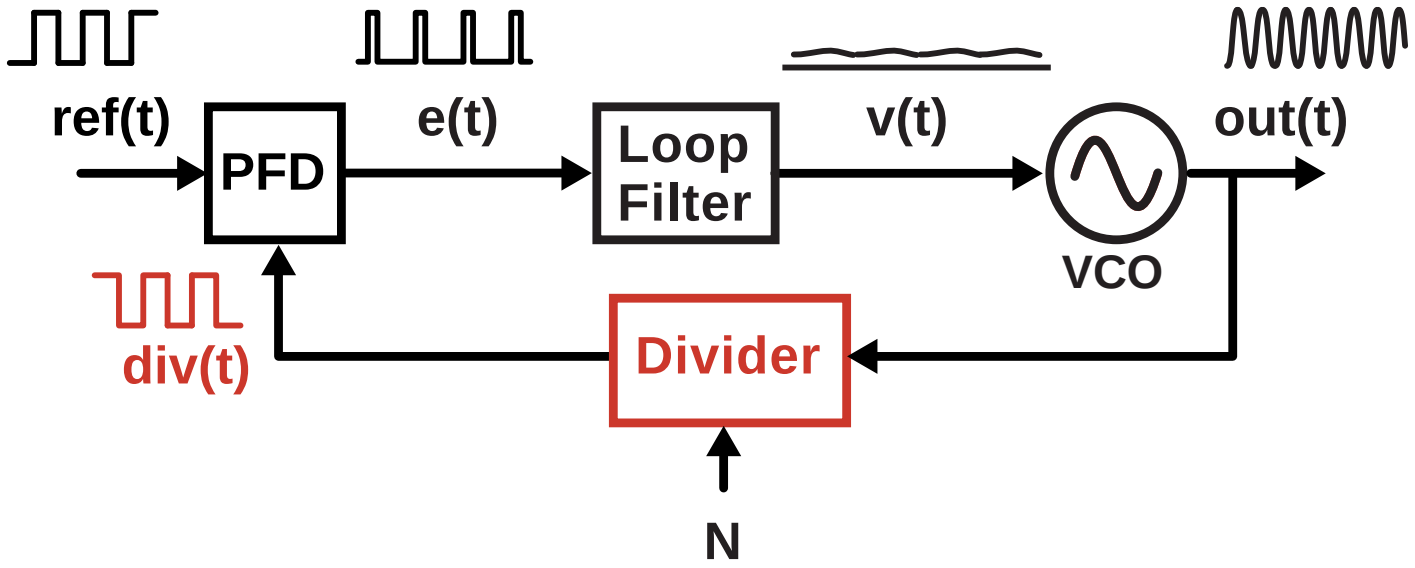
- **Design Issues: low noise, fast settling time, low power**

Overview of Integer-N Frequency Synthesizer



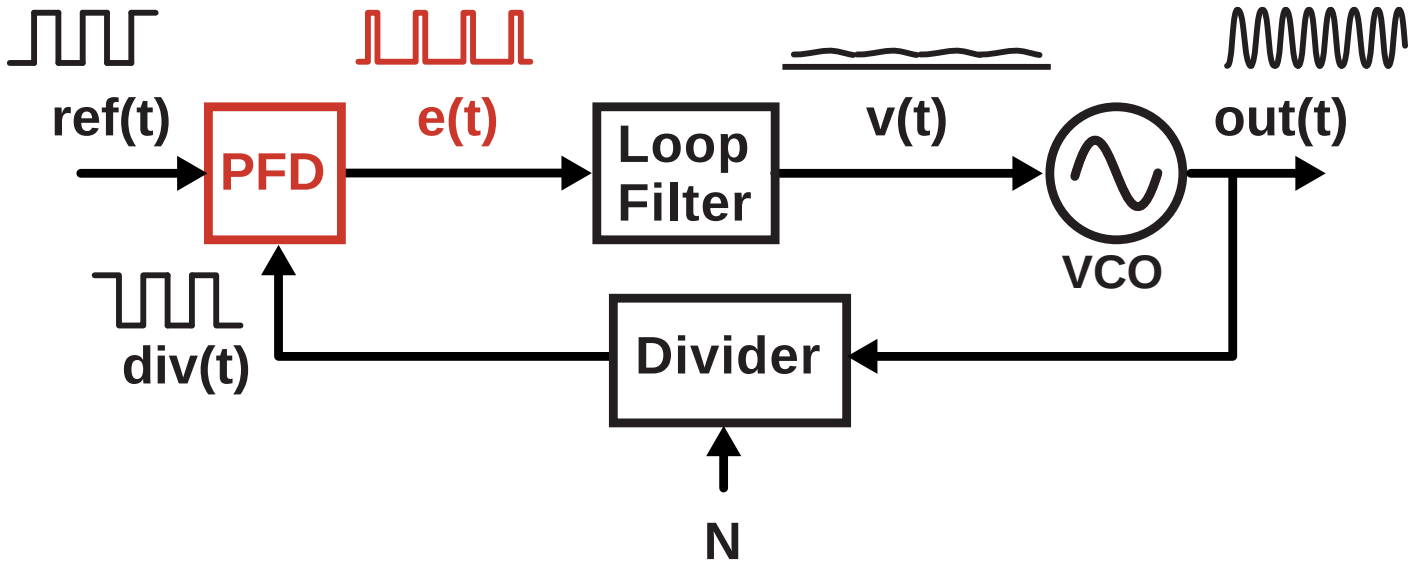
- VCO ⇨ produces high frequency sine wave

Overview of Integer-N Frequency Synthesizer



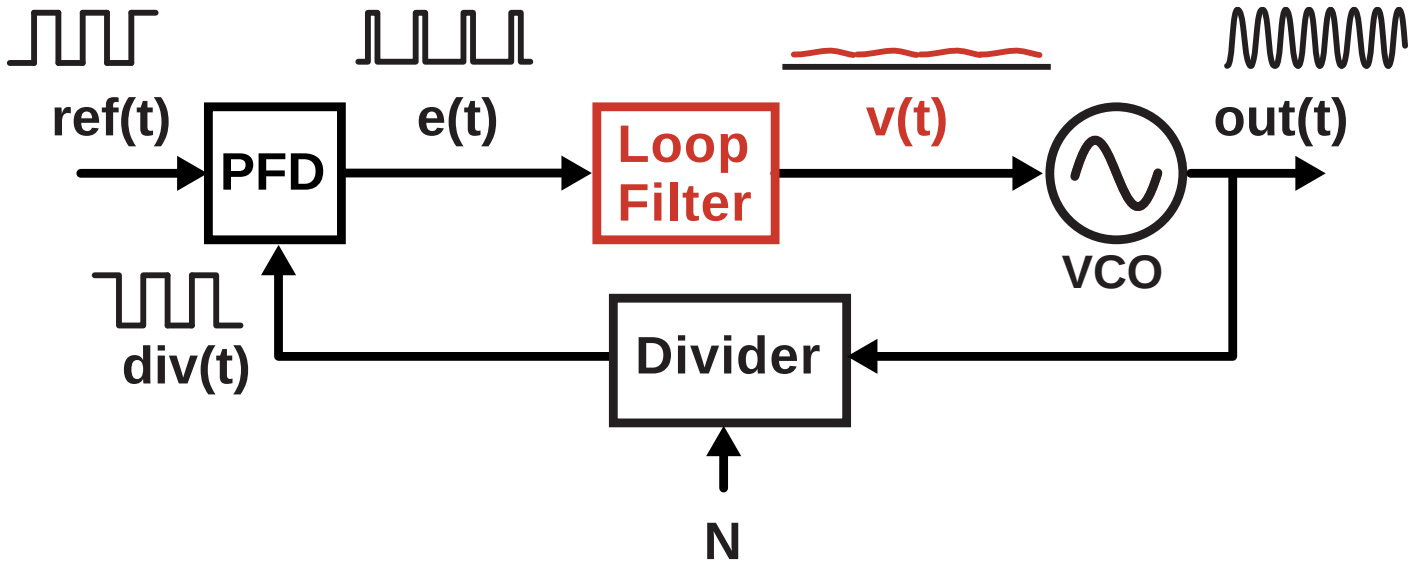
- VCO ⇨ produces high frequency sine wave
- Divider ⇨ divides down VCO frequency

Overview of Integer-N Frequency Synthesizer



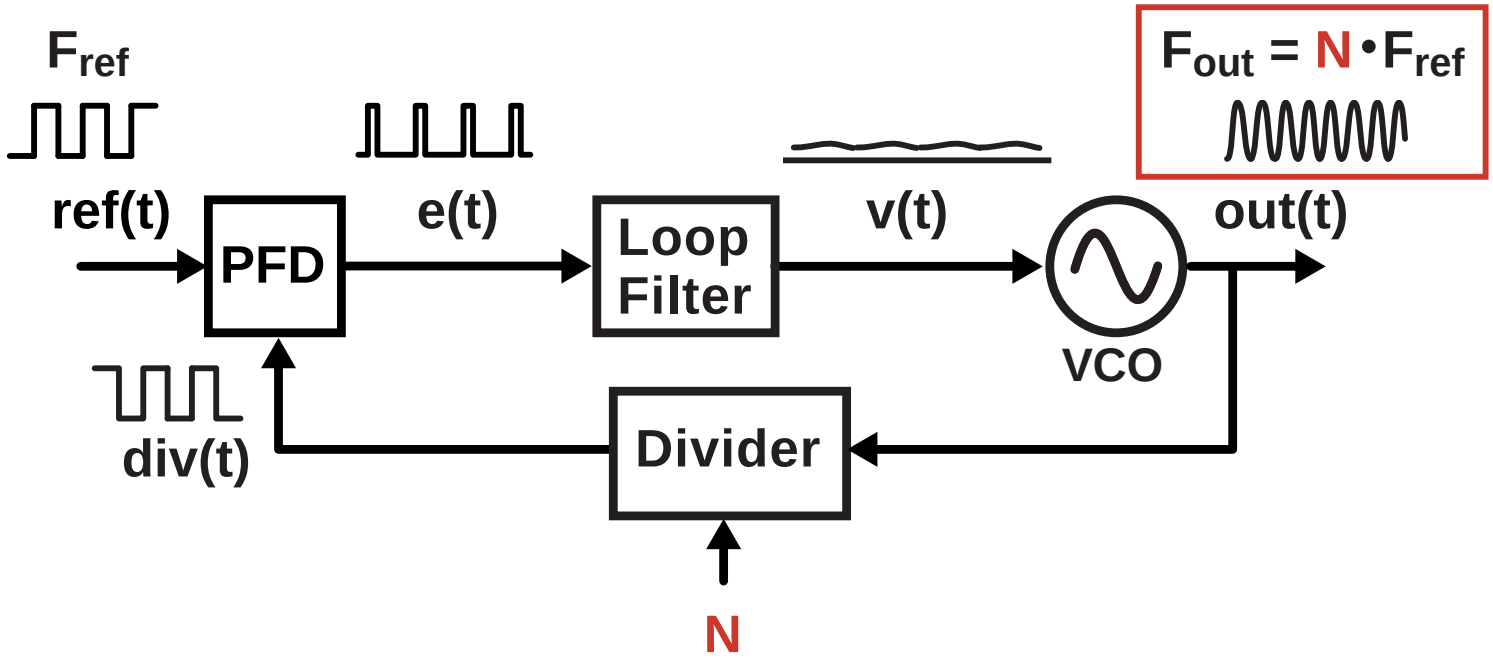
- VCO ⇨ produces high frequency sine wave
- Divider ⇨ divides down VCO frequency
- PFD ⇨ compares phase of ref and div

Overview of Integer-N Frequency Synthesizer



- VCO ⇨ produces high frequency sine wave
- Divider ⇨ divides down VCO frequency
- PFD ⇨ compares phase of ref and div
- Loop filter ⇨ smooths phase error signal

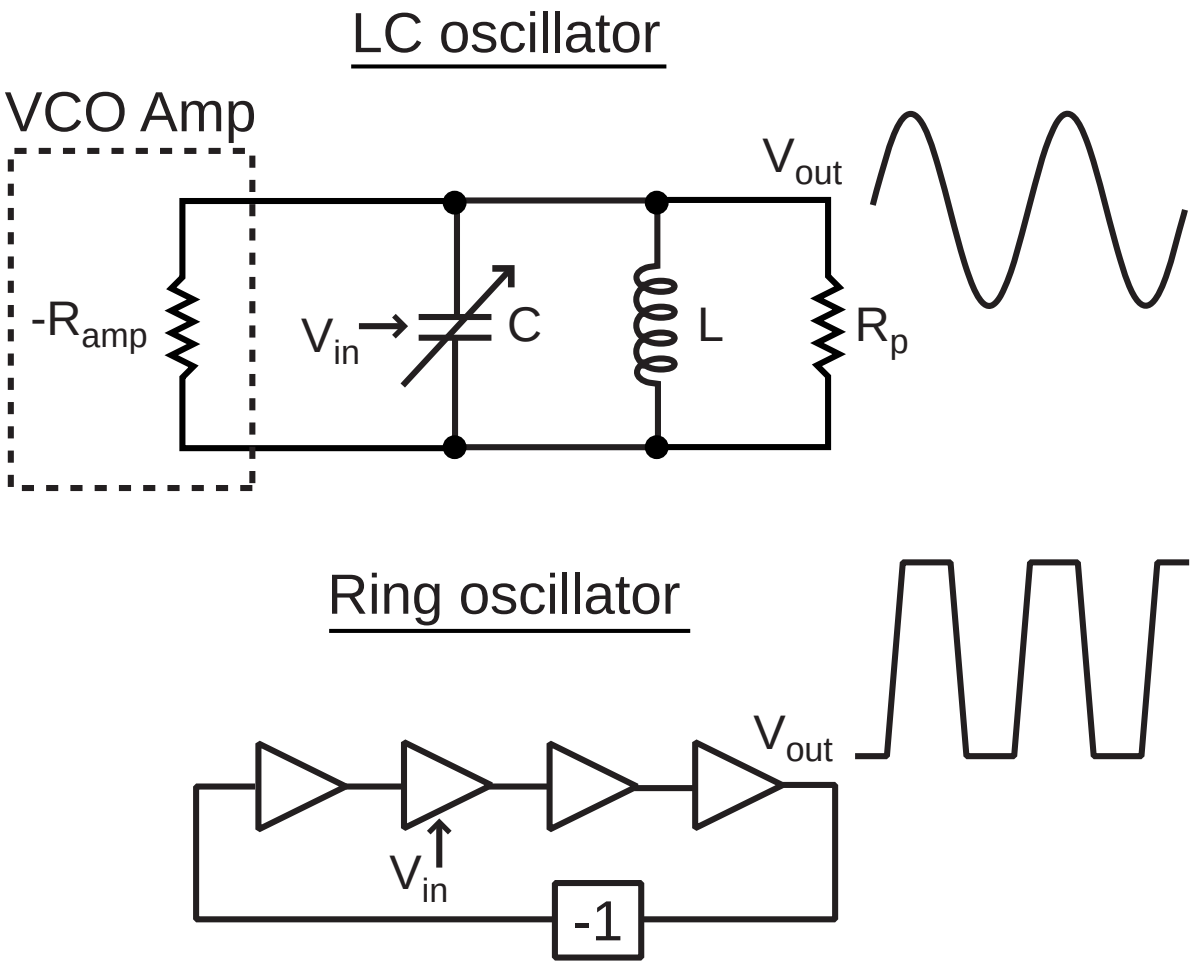
Overview of Integer-N Frequency Synthesizer



- VCO ⇨ produces high frequency sine wave
- Divider ⇨ divides down VCO frequency
- PFD ⇨ compares phase of ref and div
- Loop filter ⇨ smooths phase error signal

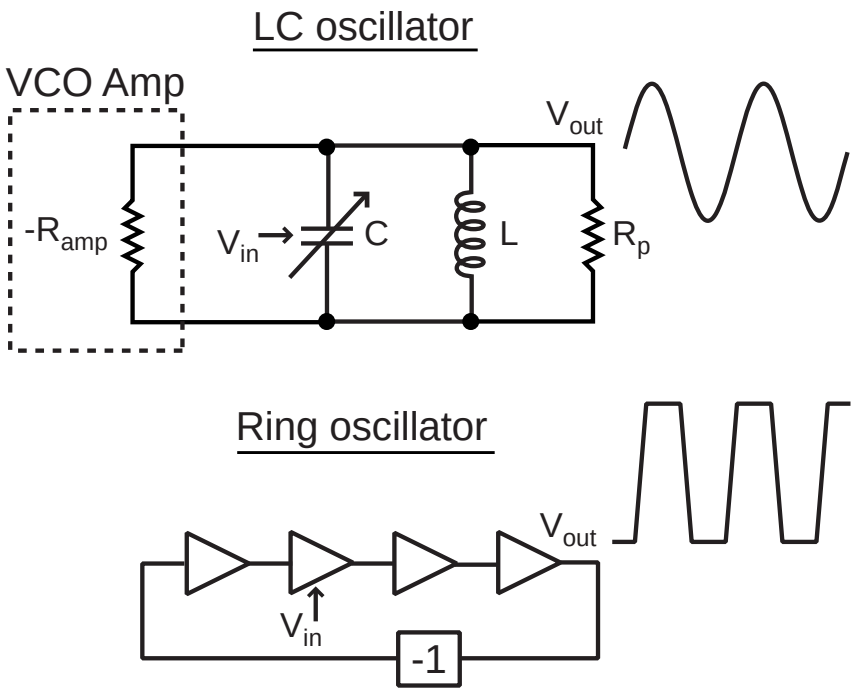
VCO frequency locks to ref. frequency multiplied by N

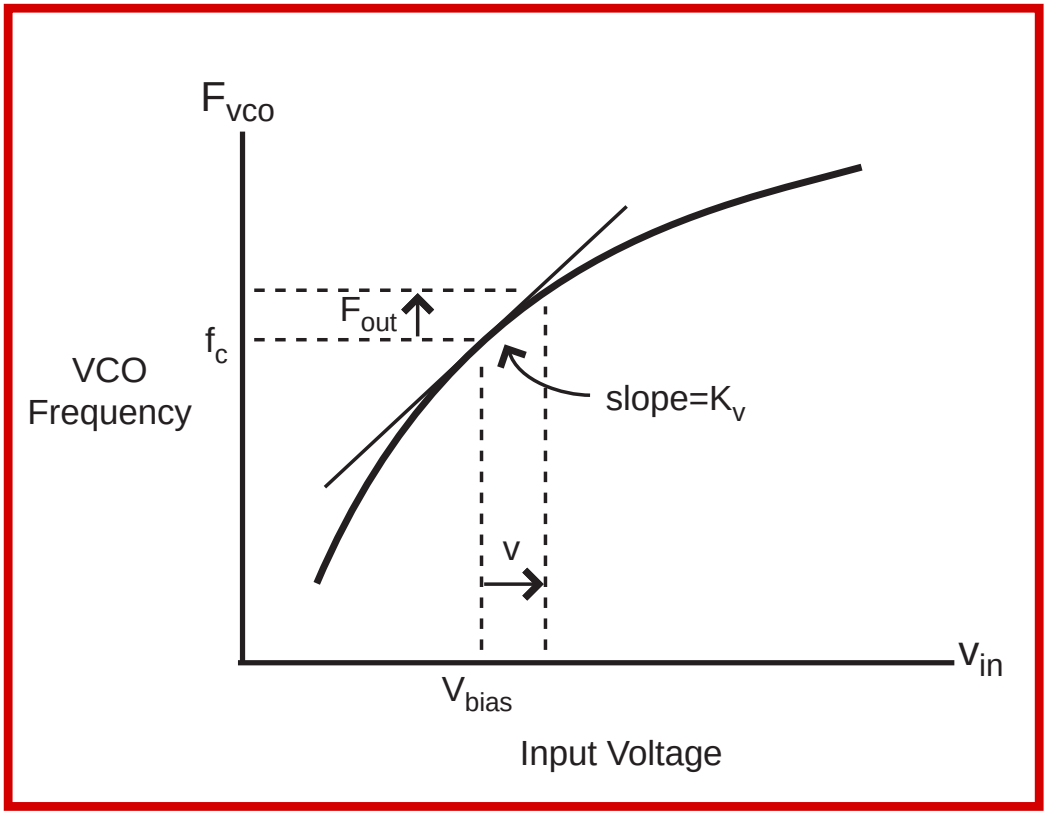
Popular VCO Structures



- **LC Oscillator: low phase noise, large area**
- **Ring Oscillator: easy to integrate, higher phase noise**

Model for Voltage to Frequency Mapping of VCO





$$F_{out}(t) = K_v v(t)$$

Model for Voltage to Phase Mapping of VCO

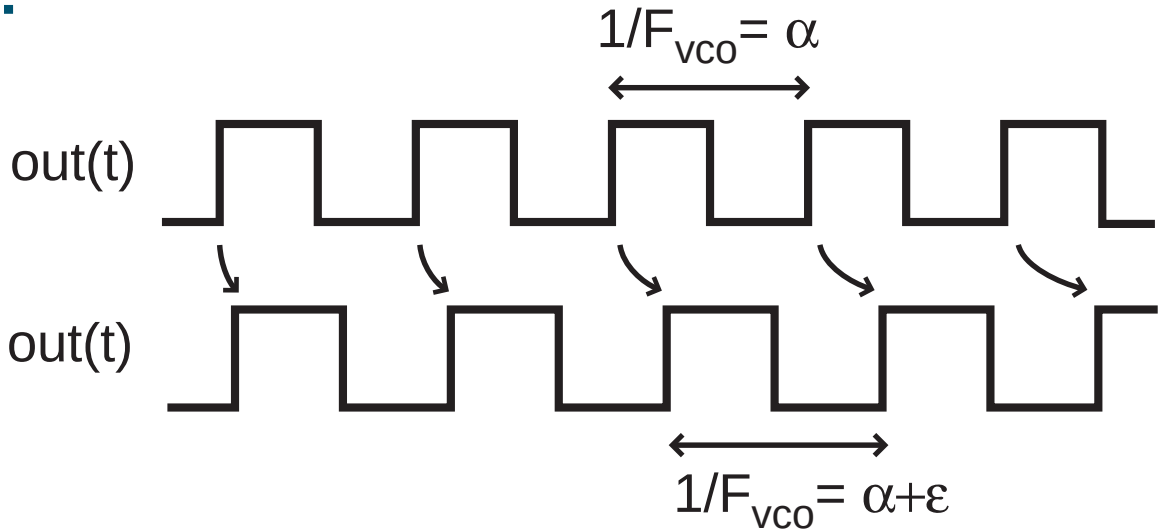
- **Time-domain frequency relationship (from previous slide)**

$$F_{out}(t) = K_v v(t)$$

- **Time-domain phase relationship**

$$\Phi_{out}(t) = \int_{-\infty}^{t} 2\pi F_{out}(\tau) d\tau = \int_{-\infty}^{t} 2\pi K_v v(\tau) d\tau$$

- **Intuition of integral relationship between frequency and phase:**

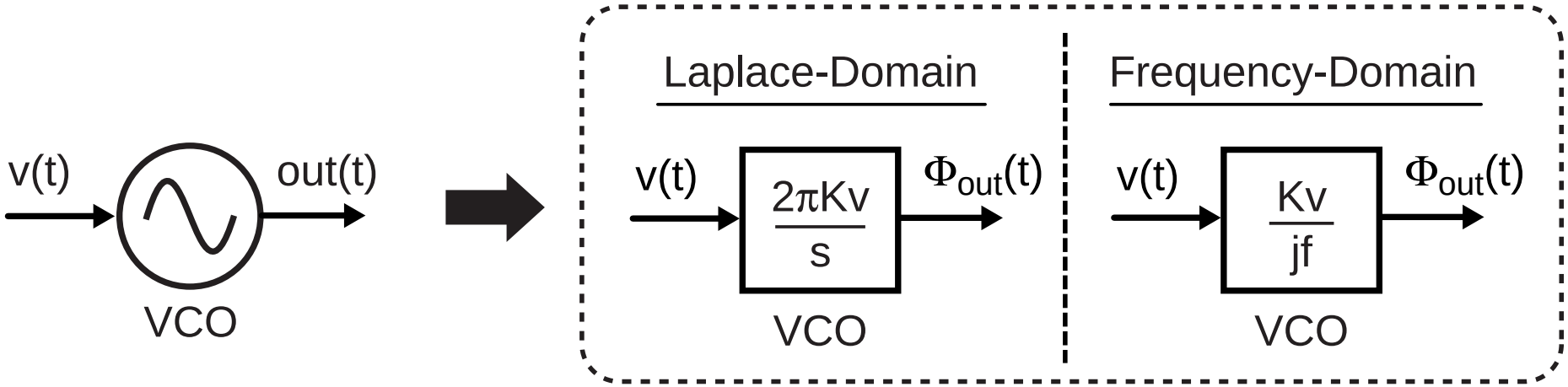


Frequency-Domain Model for VCO

- **Time-domain relationship (from previous slide)**

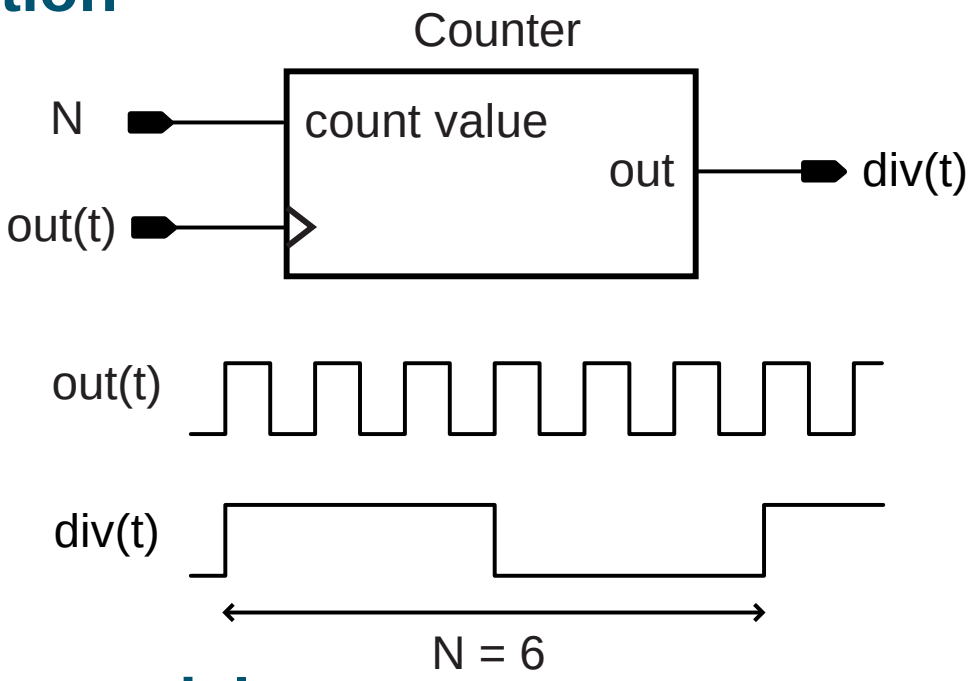
$$\Phi_{out}(t) = \int_{-\infty}^{t} 2\pi K_v v(\tau) d\tau$$

- **Corresponding frequency-domain model**



Divider

- **Implementation**



- **Time-domain model**
  - **Frequency:**

$$F_{div}(t) = \frac{1}{N} F_{out}(t).$$

  - **Phase:**

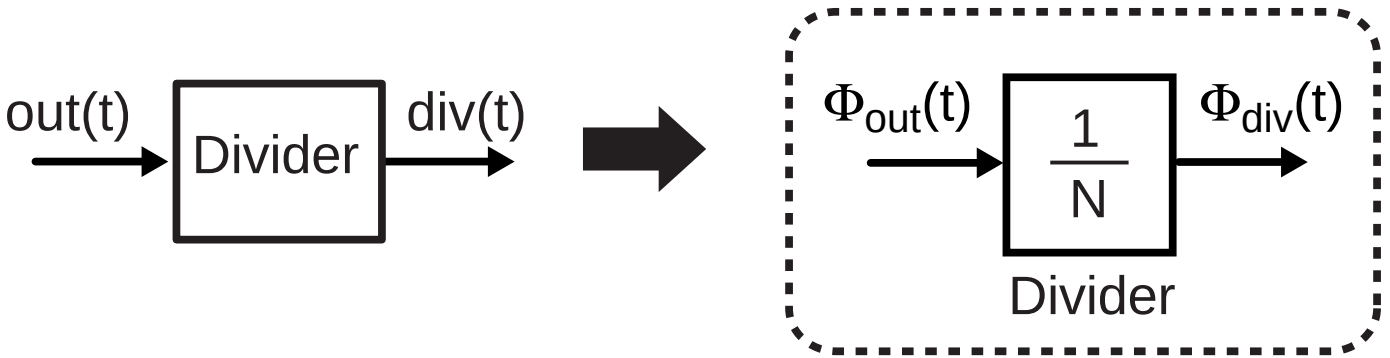
$$\Phi_{div}(t) = \int_{-\infty}^{t} 2\pi \frac{1}{N} F_{out}(\tau) d\tau = \frac{1}{N} \Phi_{out}(t)$$

Frequency-Domain Model of Divider

- **Time-domain relationship between VCO phase and divider output phase (from previous slide)**

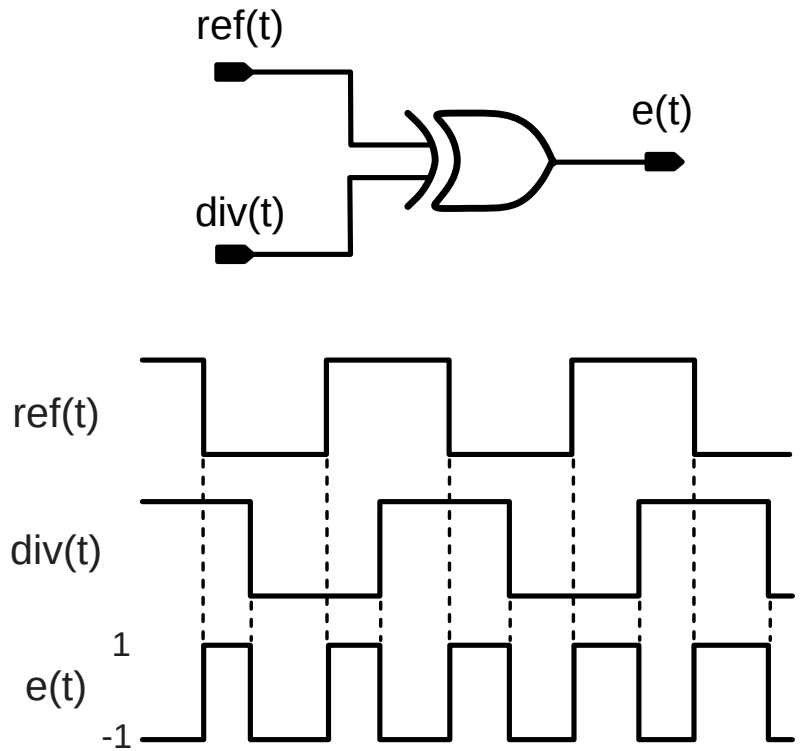
$$\Phi_{div}(t) = \frac{1}{N}\Phi_{out}(t)$$

- **Corresponding frequency-domain model (same as Laplace-domain)**



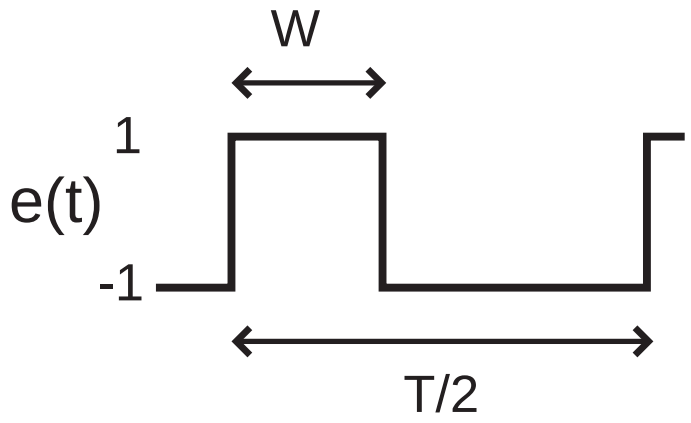
Phase Detector (PD)

- XOR structure
  - Average value of error pulses corresponds to phase error
  - Loop filter extracts the average value and feeds to VCO



Modeling of XOR Phase Detector

- **Average value of pulses is extracted by loop filter**
  - **Look at detector output over one cycle:**



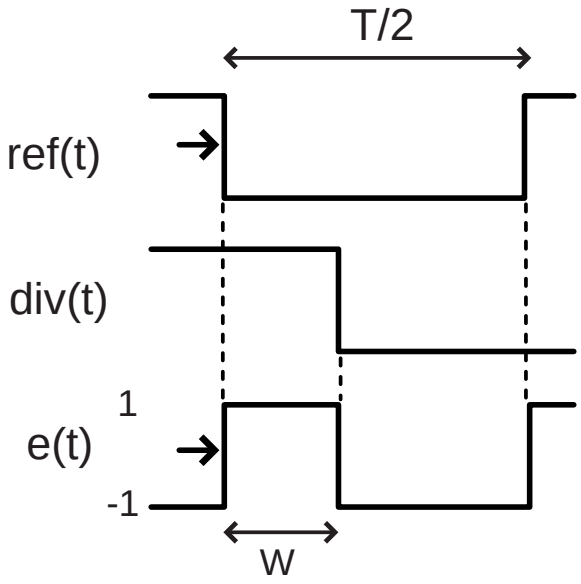
- **Equation:**

$$avg\{e(t)\} = -1 + 2\frac{W}{T/2}$$

Relate Pulse Width to Phase Error

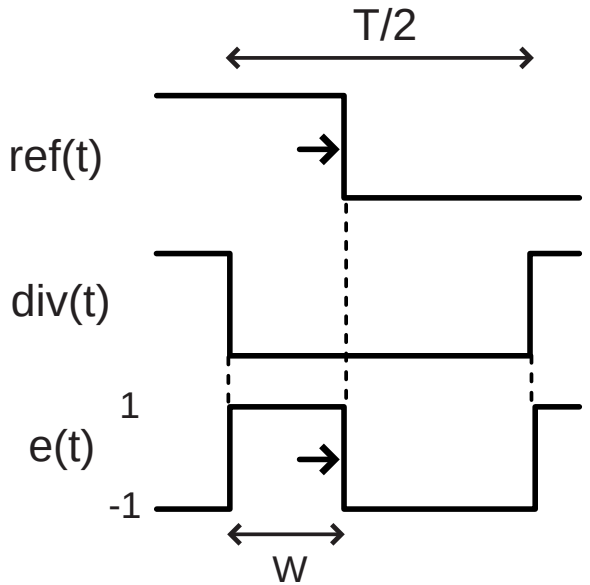
- **Two cases:**

$$-\pi < \Phi_{ref} - \Phi_{div} < 0$$



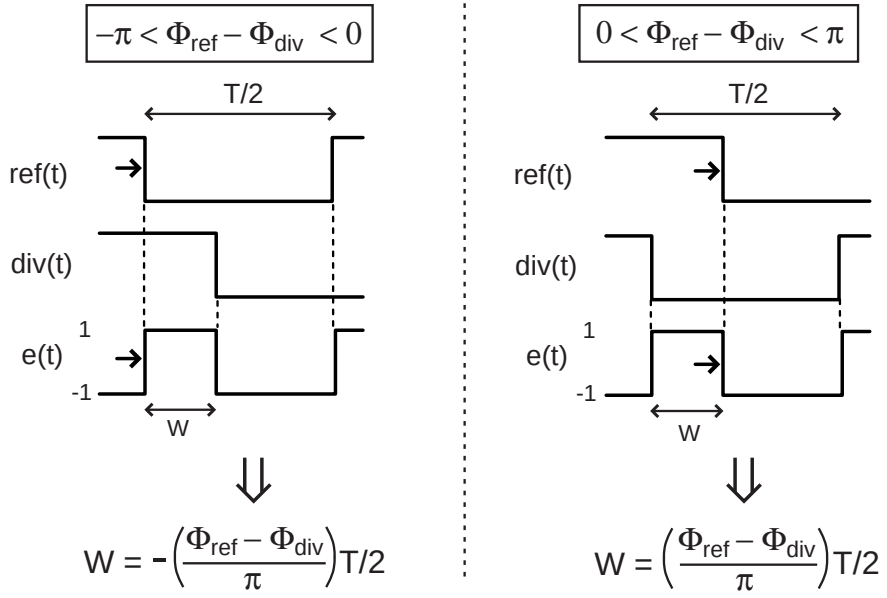
$$W = -\left(\frac{\Phi_{ref} - \Phi_{div}}{\pi}\right) T/2$$

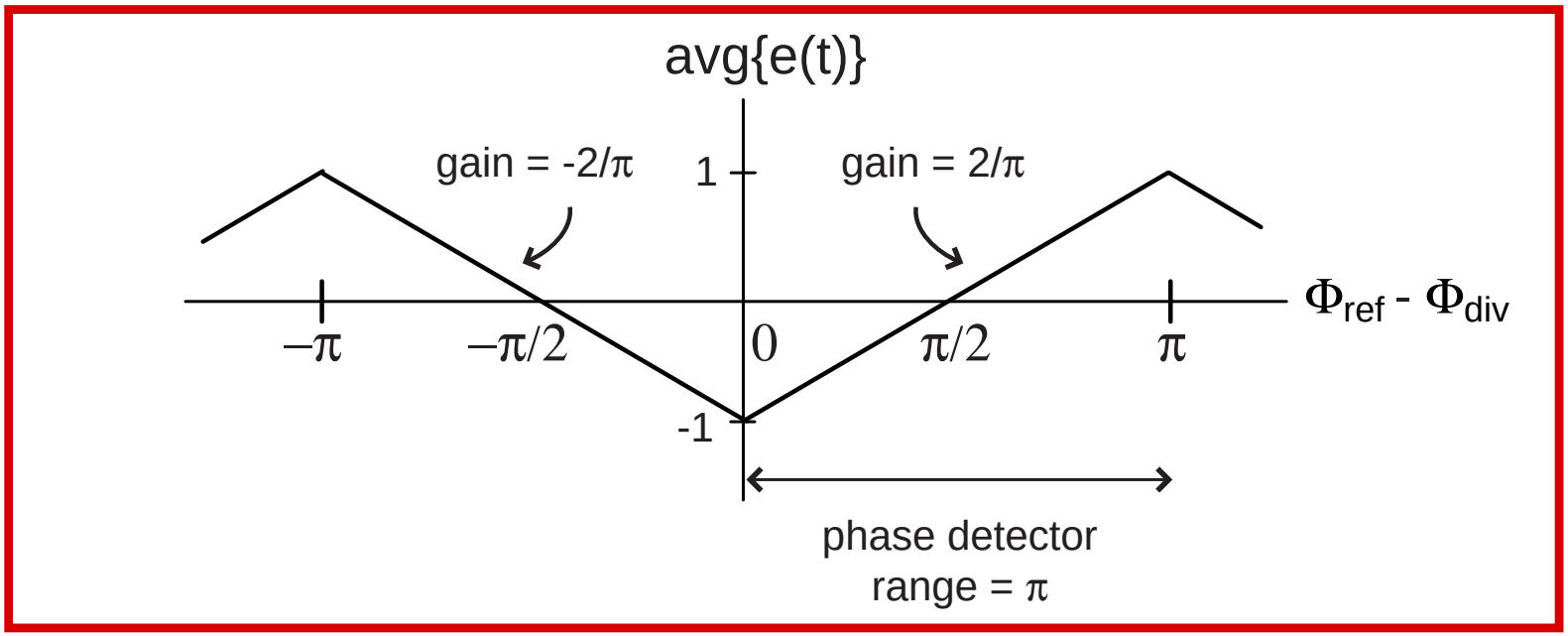
$$0 < \Phi_{ref} - \Phi_{div} < \pi$$



$$W = \left(\frac{\Phi_{ref} - \Phi_{div}}{\pi}\right) T/2$$

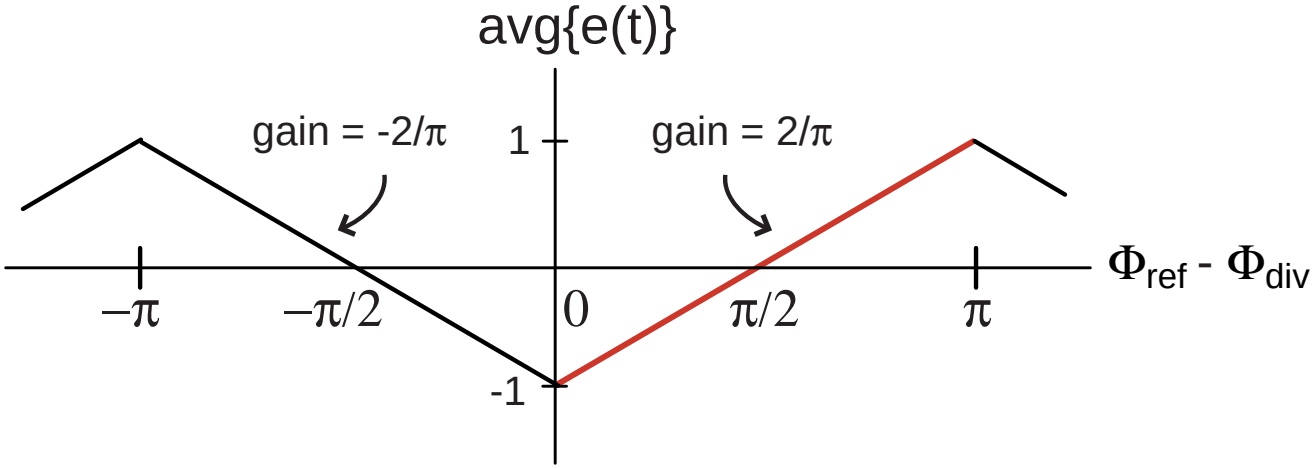
Overall XOR Phase Detector Characteristic



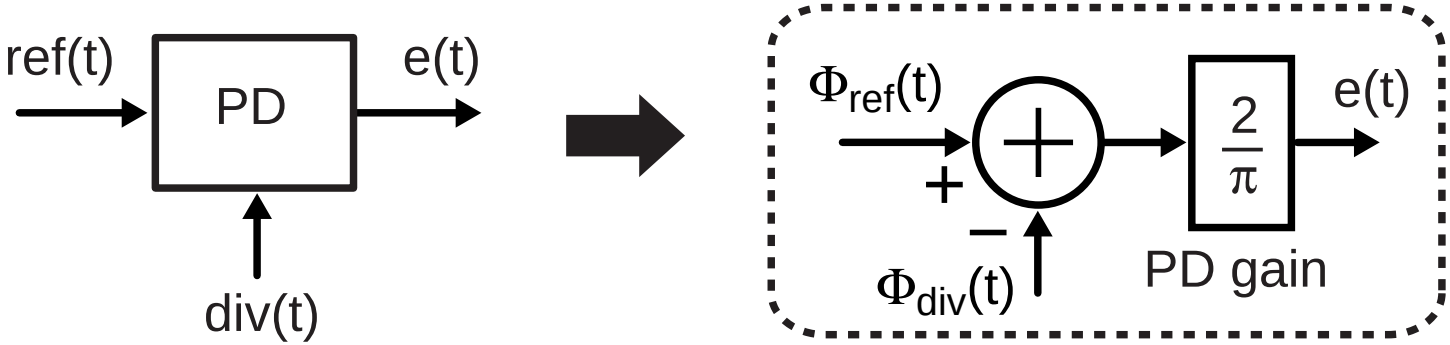


Frequency-Domain Model of XOR Phase Detector

- **Assume phase difference confined within 0 to π radians**
  - **Phase detector characteristic looks like a constant gain element**

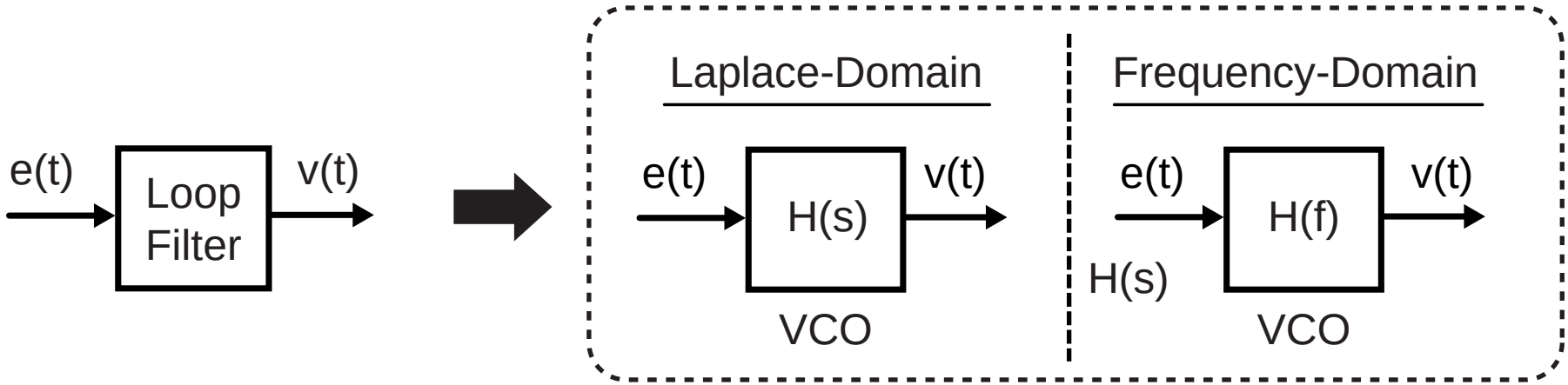


- **Corresponding frequency-domain model**

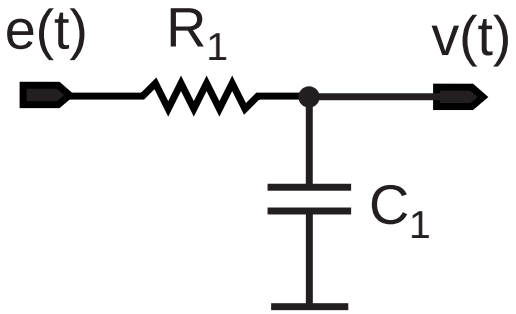


Loop Filter

- Consists of a lowpass filter to extract average of phase detector error pulses
- Frequency-domain model



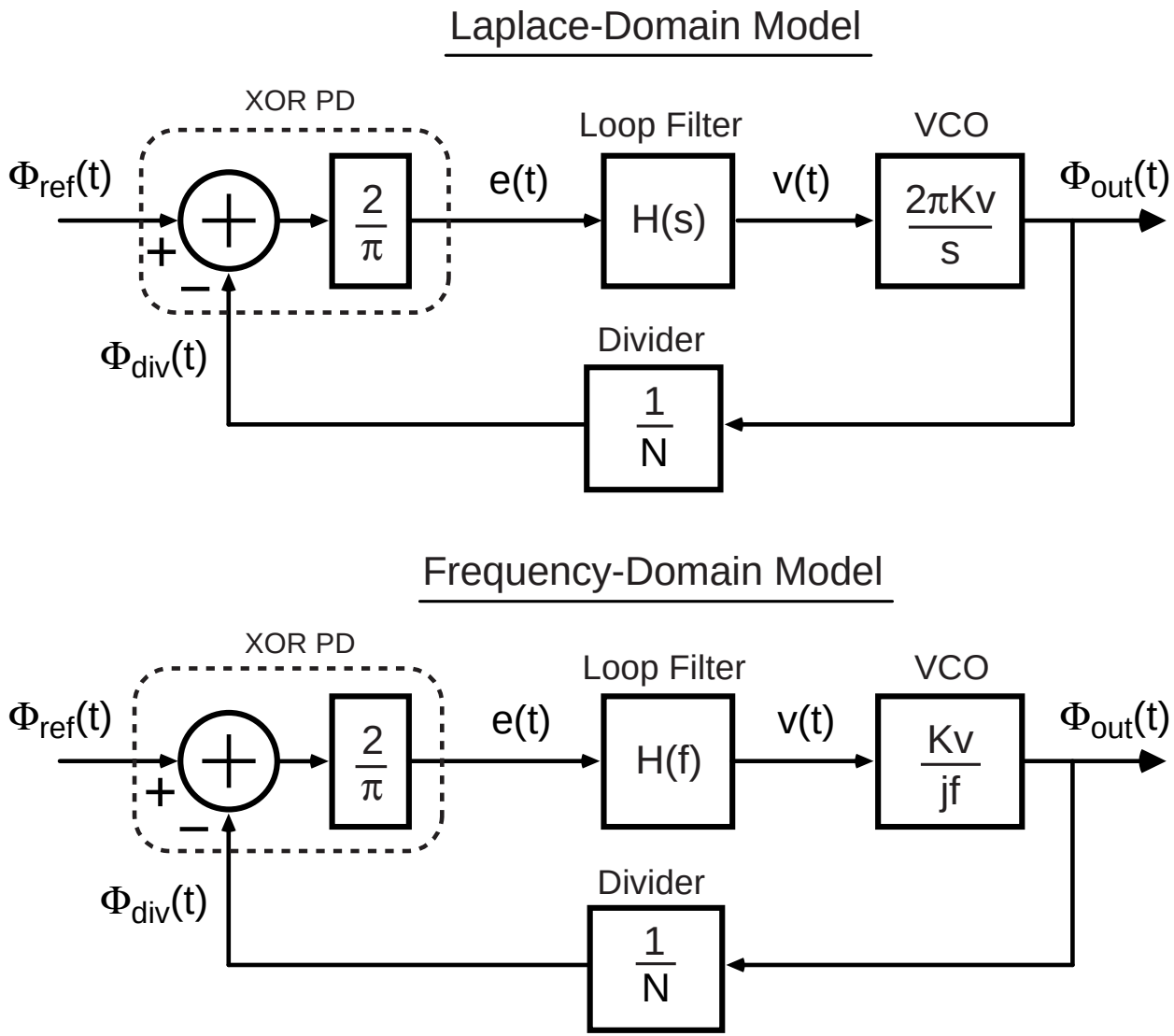
- First order example



$$\Rightarrow \; H(s) = \frac{1}{1 + sR_1C_1}$$

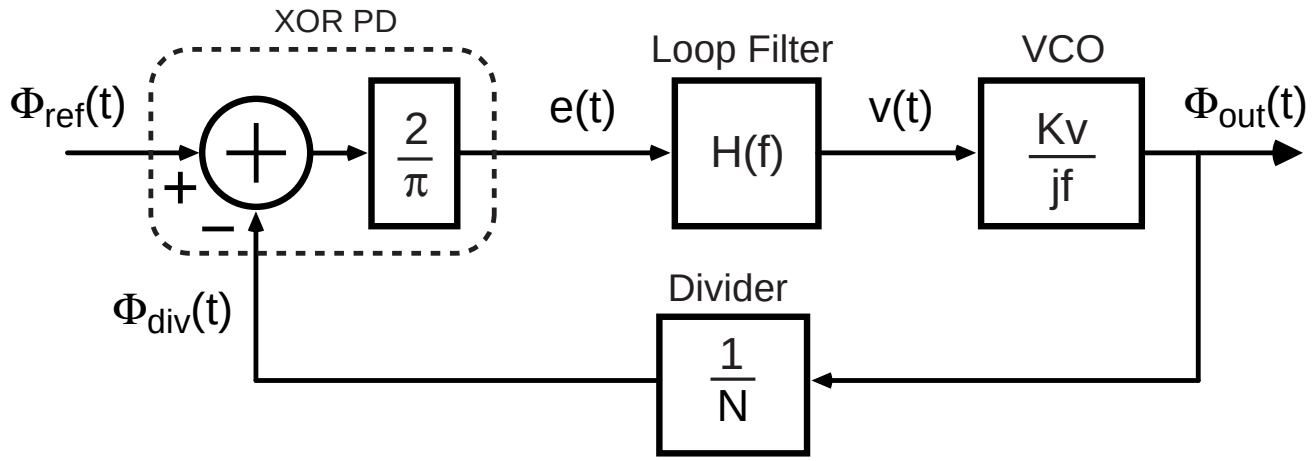
Overall Linearized PLL Frequency-Domain Model

- **Combine models of individual components**



Open Loop versus Closed Loop Response

- **Frequency-domain model**



- **Define A(f) as open loop response**

$$A(f) = \frac{2}{\pi} H(f) \left(\frac{K_v}{jf} \right) \frac{1}{N}$$

- **Define G(f) as a parameterizing function (related to closed loop response)**

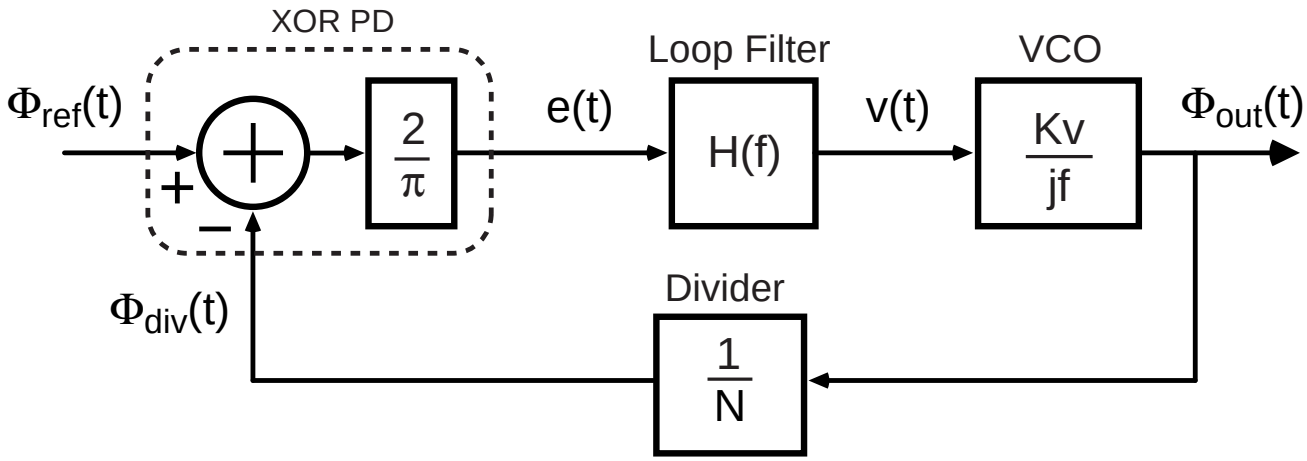
$$G(f) = \frac{A(f)}{1 + A(f)}$$

Classical PLL Transfer Function Design Approach

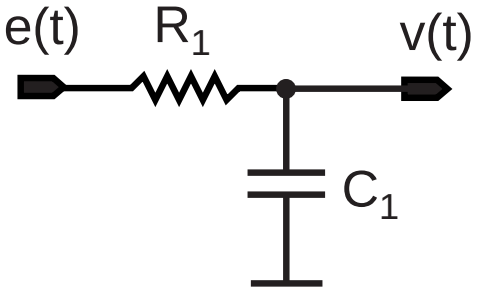
1. Choose an appropriate topology for $H(f)$
   - Usually chosen from a small set of possibilities
2. Choose pole/zero values for $H(f)$ as appropriate for the required filtering of the phase detector output
   - Constraint: set pole/zero locations higher than desired PLL bandwidth to allow stable dynamics to be possible
3. Adjust the open-loop gain to achieve the required bandwidth while maintaining stability
   - Plot gain and phase bode plots of $A(f)$
   - Use phase (or gain) margin criterion to infer stability

Example: First Order Loop Filter

- **Overall PLL block diagram**

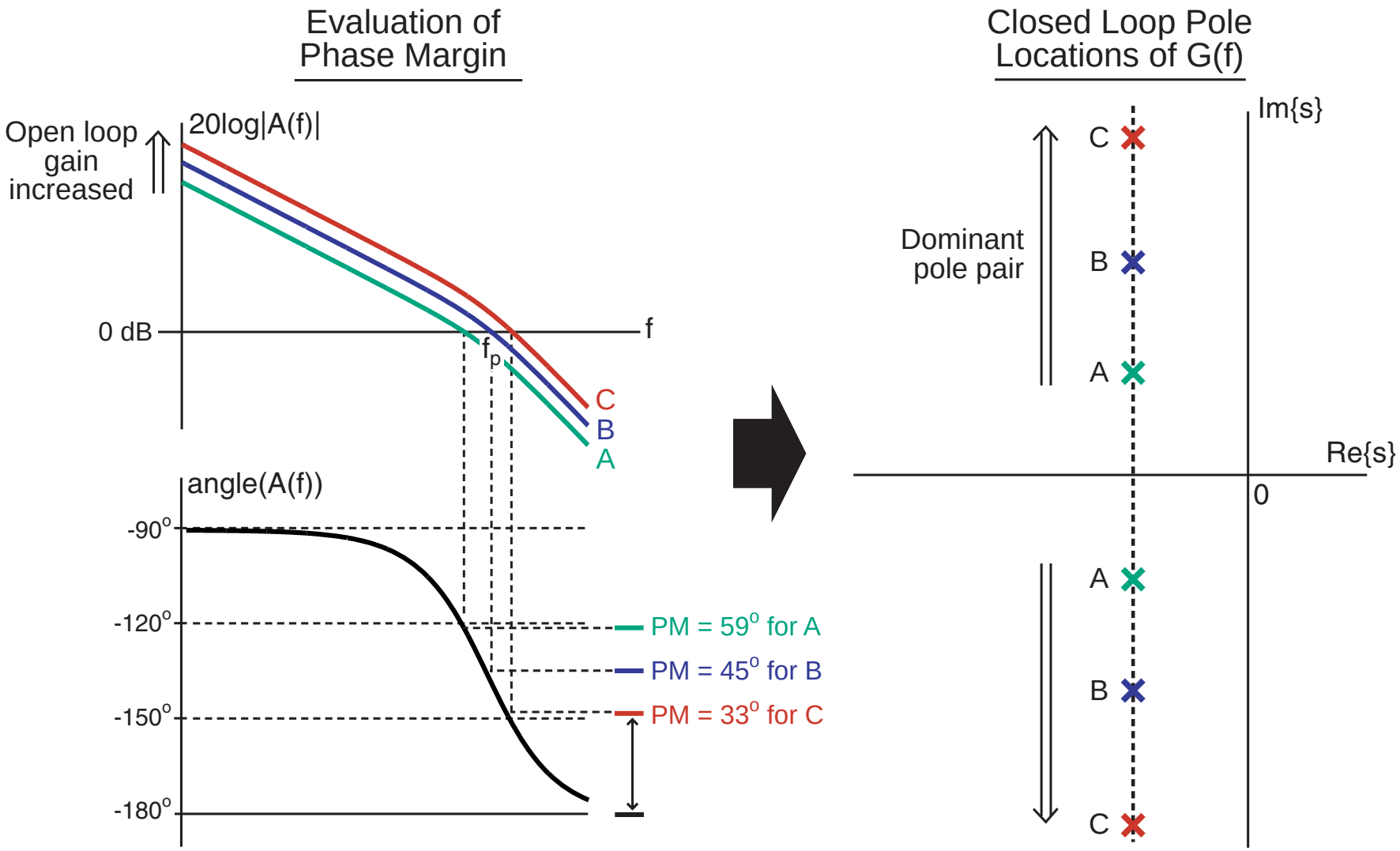


- **Loop filter**



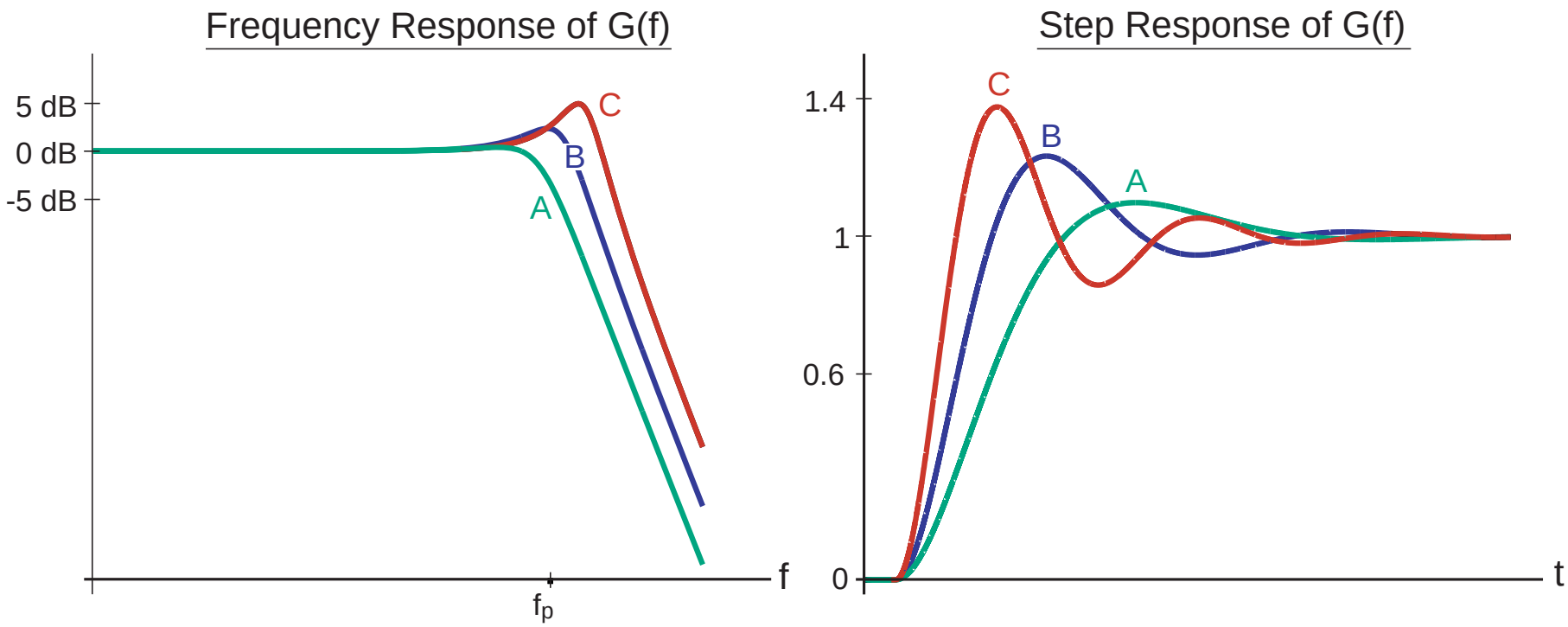
$$\Rightarrow \; H(f) = \frac{1}{1 + jf/f_p}$$

Closed Loop Poles Versus Open Loop Gain



- **Higher open loop gain leads to an increase in Q of closed loop poles**

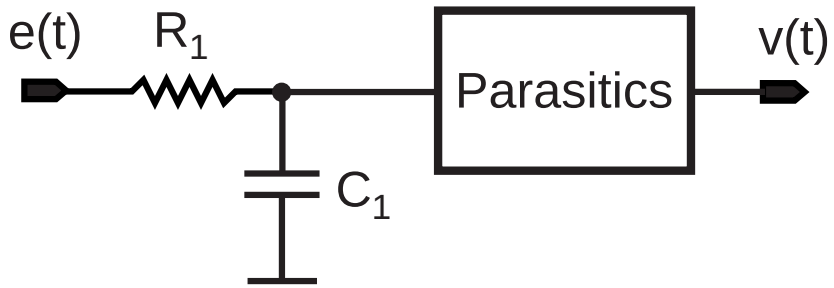
Corresponding Closed Loop Response



- **Increase in open loop gain leads to**
  - **Peaking in closed loop frequency response**
  - **Ringing in closed loop step response**

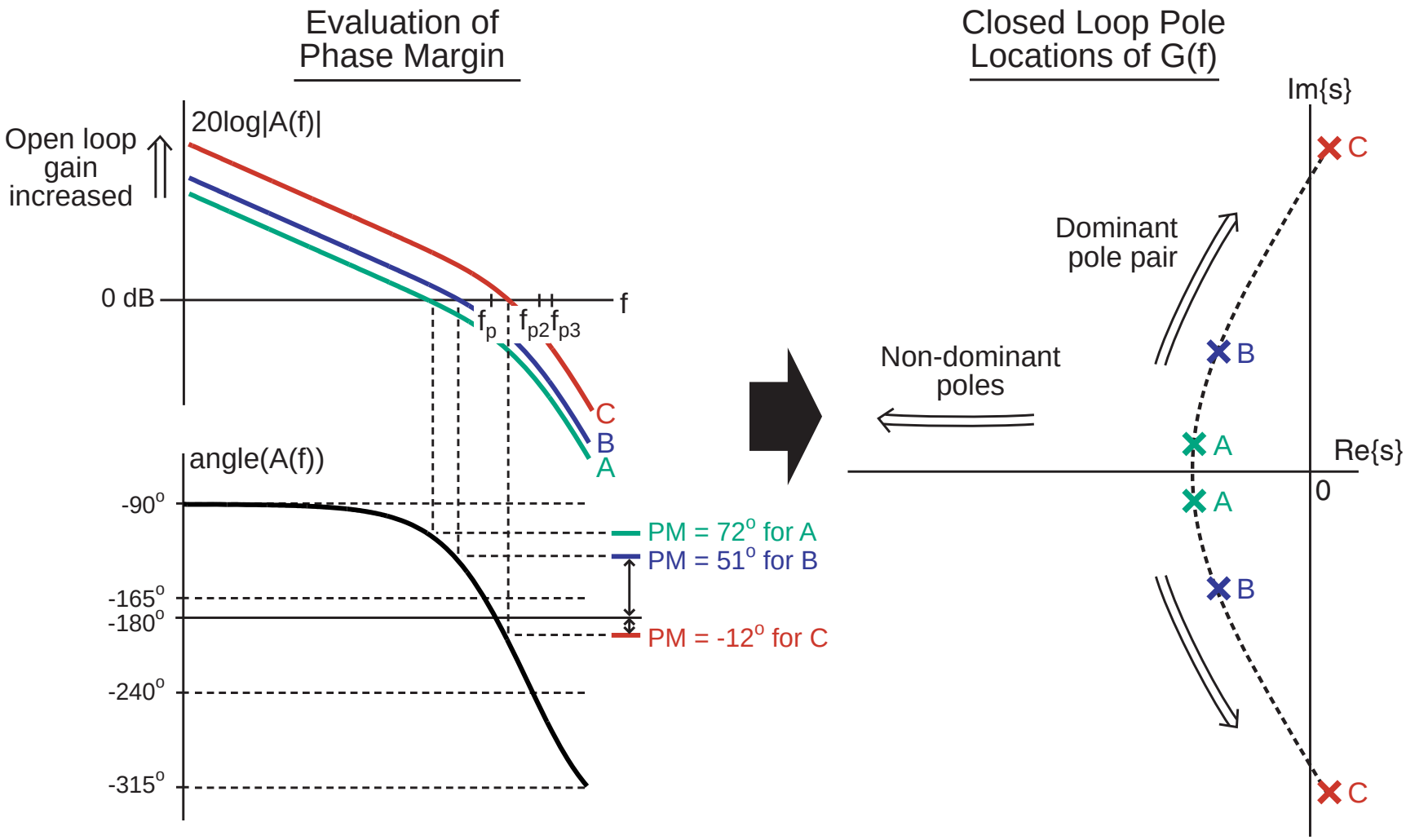
The Impact of Parasitic Poles

- Loop filter and VCO may have additional parasitic poles and zeros due to their circuit implementation
- We can model such parasitics by including them in the loop filter transfer function
- Example: add two parasitic poles to first order filter

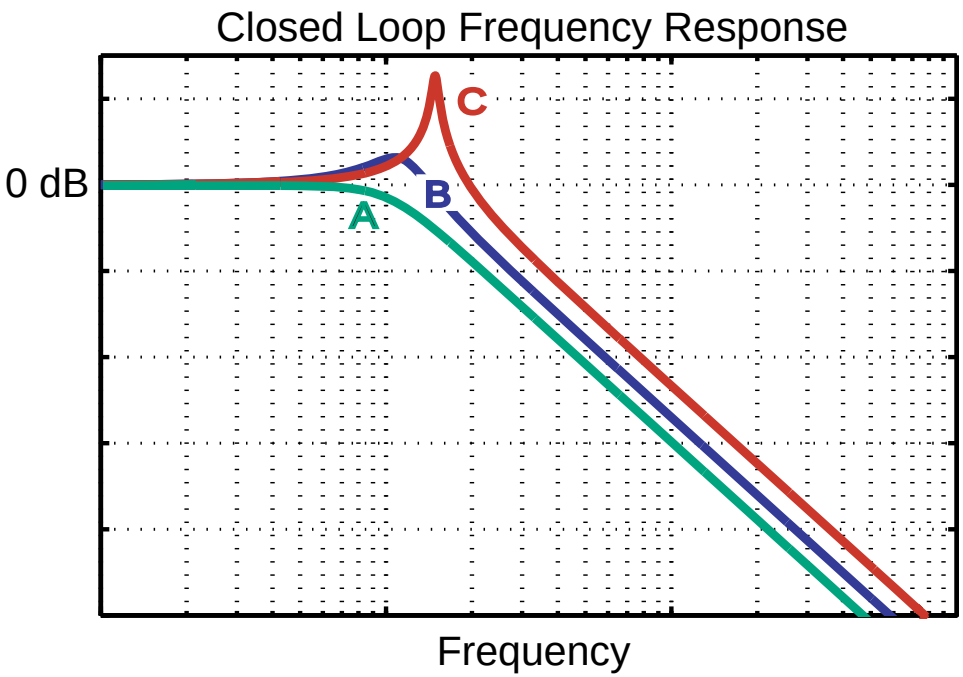


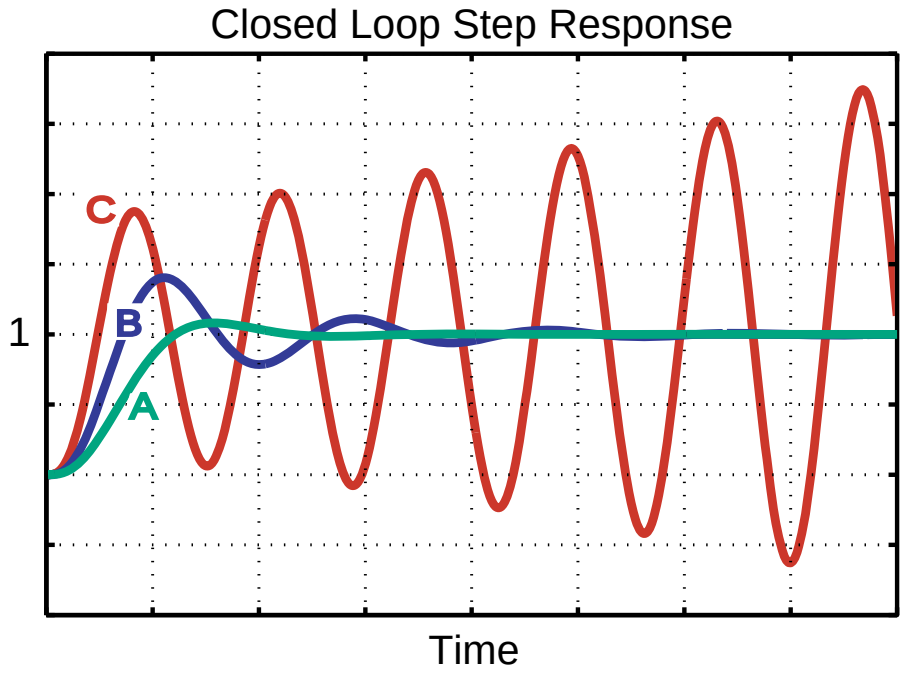
$$\Rightarrow \quad H(f) = \left(\frac{1}{1+jf/f_1}\right)\left(\frac{1}{1+jf/f_2}\right)\left(\frac{1}{1+jf/f_3}\right)$$

Closed Loop Poles Versus Open Loop Gain



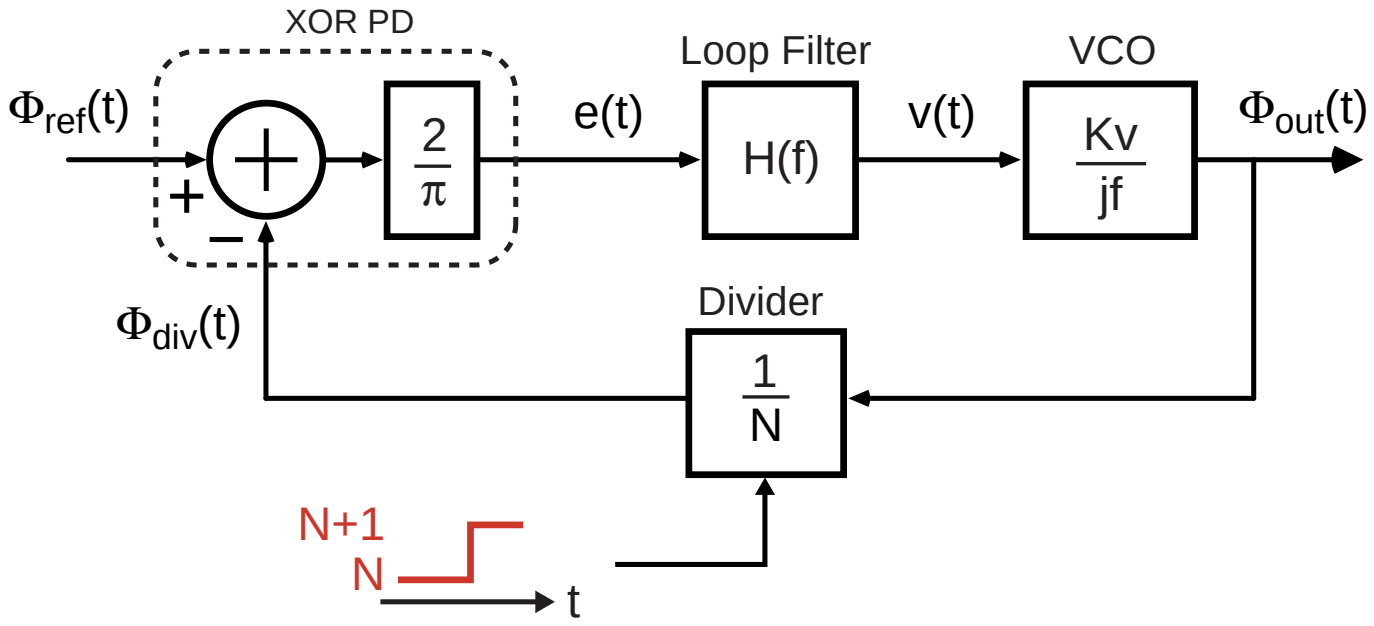
Corresponding Closed Loop Response





- **Increase in open loop gain now eventually leads to instability**
  - **Large peaking in closed loop frequency response**
  - **Increasing amplitude in closed loop step response**

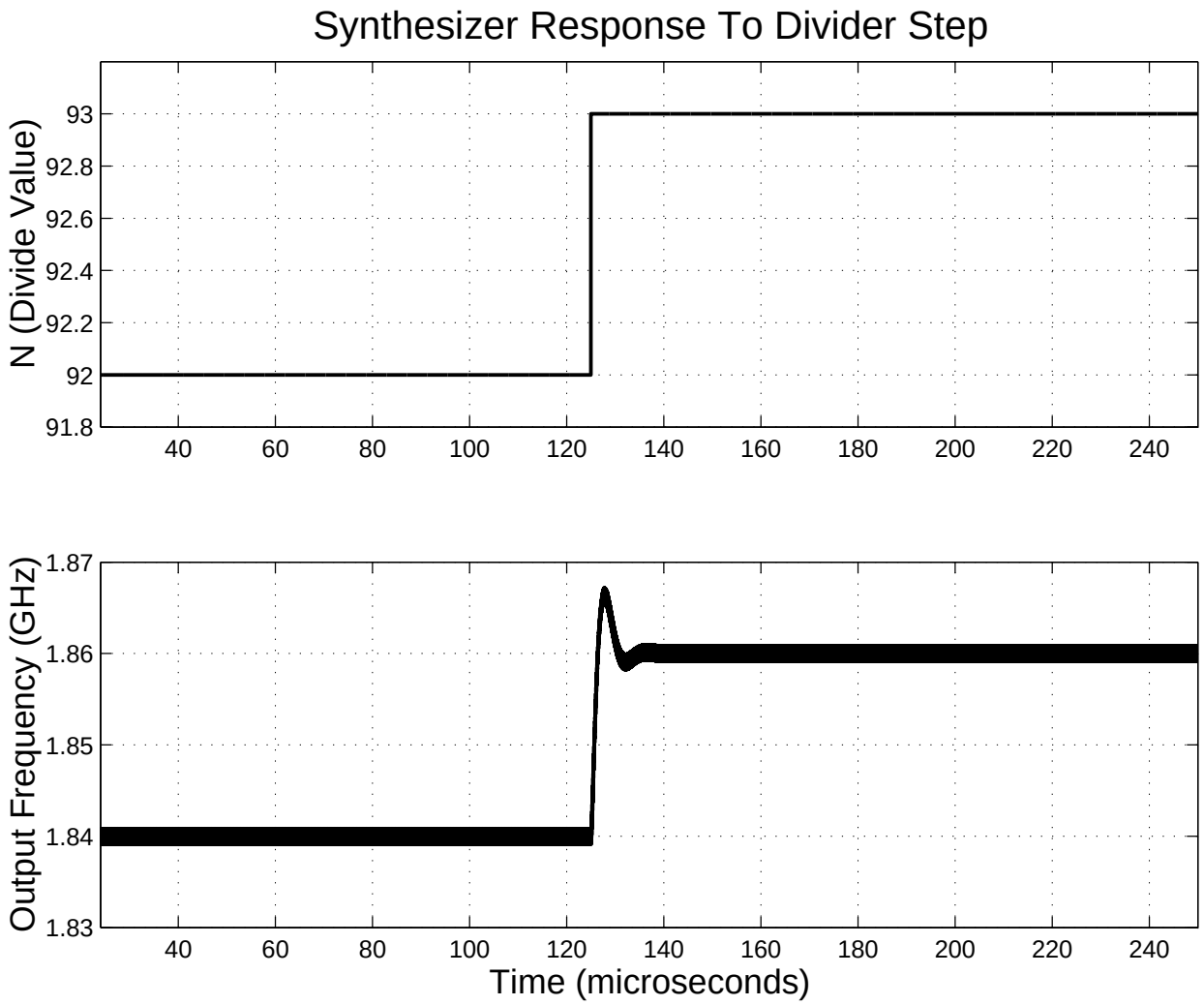
Response of PLL to Divide Value Changes



- **Change in output frequency achieved by changing the divide value**
- **Classical approach provides no direct model of impact of divide value variations**
  - **Treat divide value variation as a perturbation to a linear system**
    - PLL responds according to its closed loop response

Response of an Actual PLL to Divide Value Change

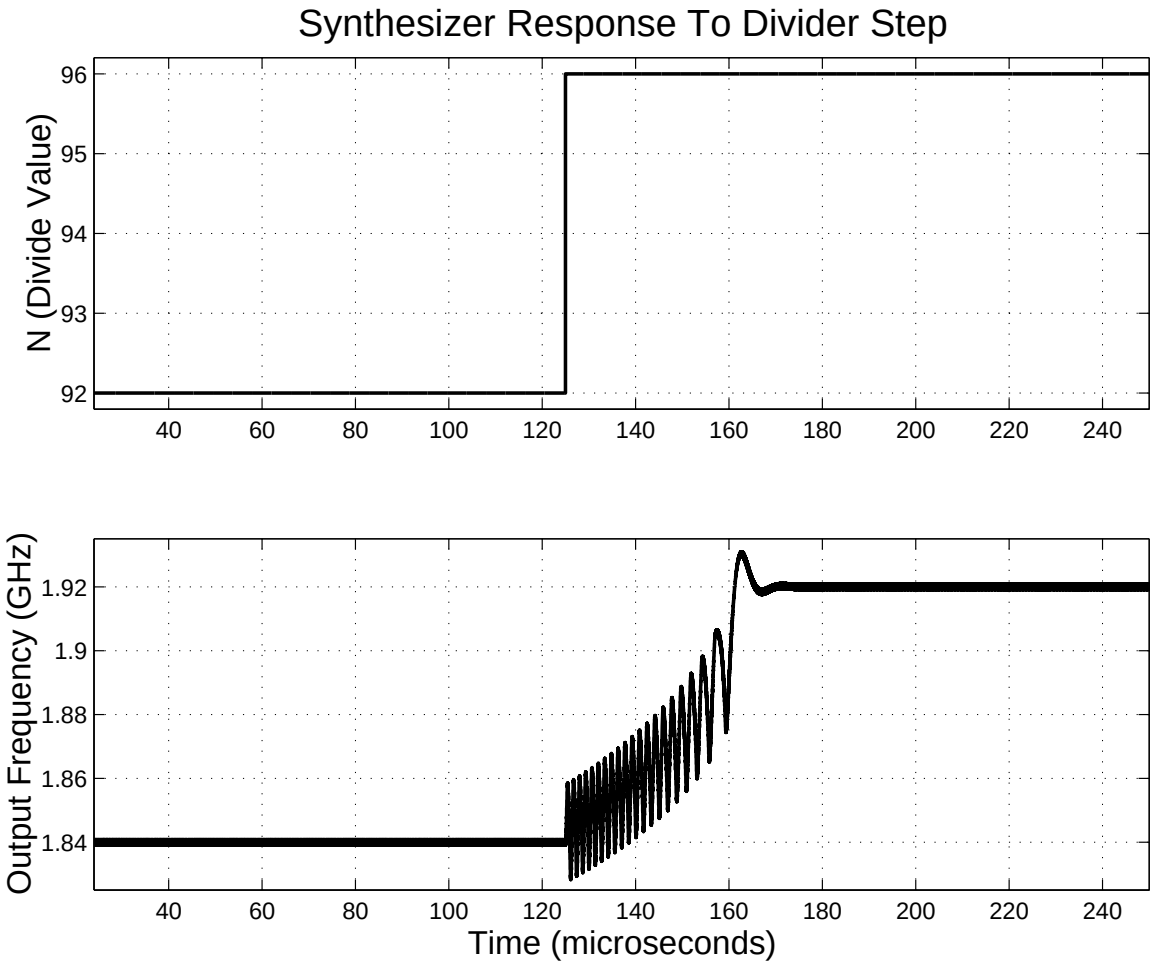
- **Example: Change divide value by one**



- **PLL responds according to closed loop response!**

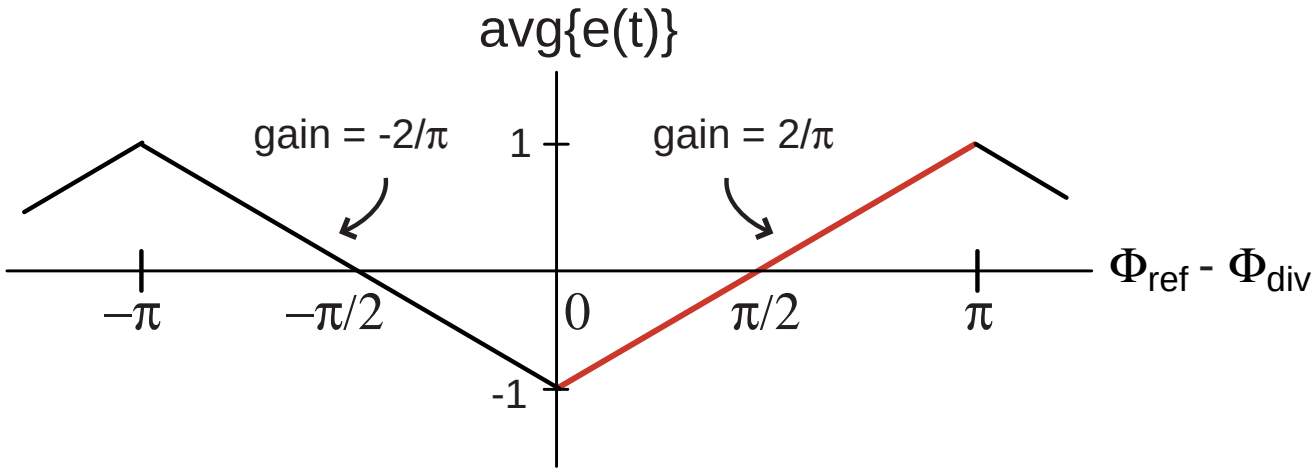
What Happens with Large Divide Value Variations?

- PLL temporarily loses frequency lock (cycle slipping occurs)



  - Why does this happen?

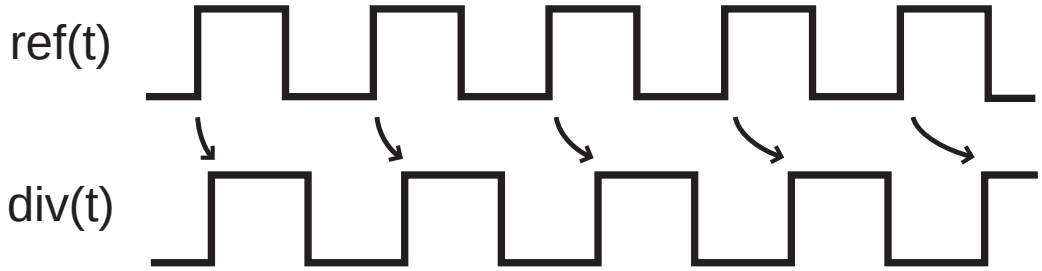
Recall Phase Detector Characteristic



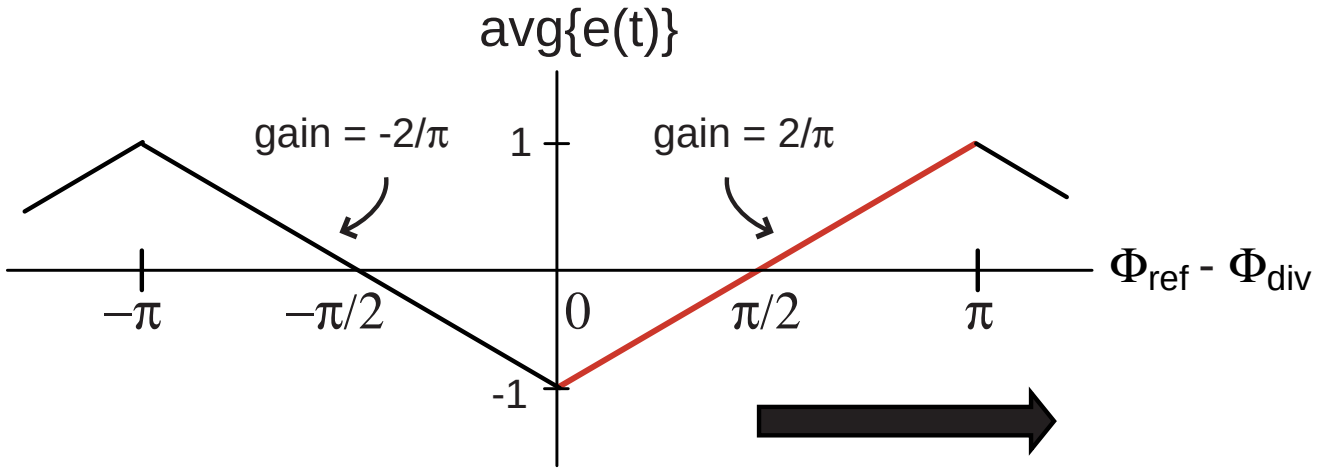
- To simplify modeling, we assumed that we always operated in a confined phase range (0 to π)
  - Led to a simple PD model
- Large perturbations knock us out of that confined phase range
  - PD behavior varies depending on the phase range it happens to be in

Cycle Slipping

- Consider the case where there is a frequency offset between divider output and reference
  - We know that phase difference will accumulate

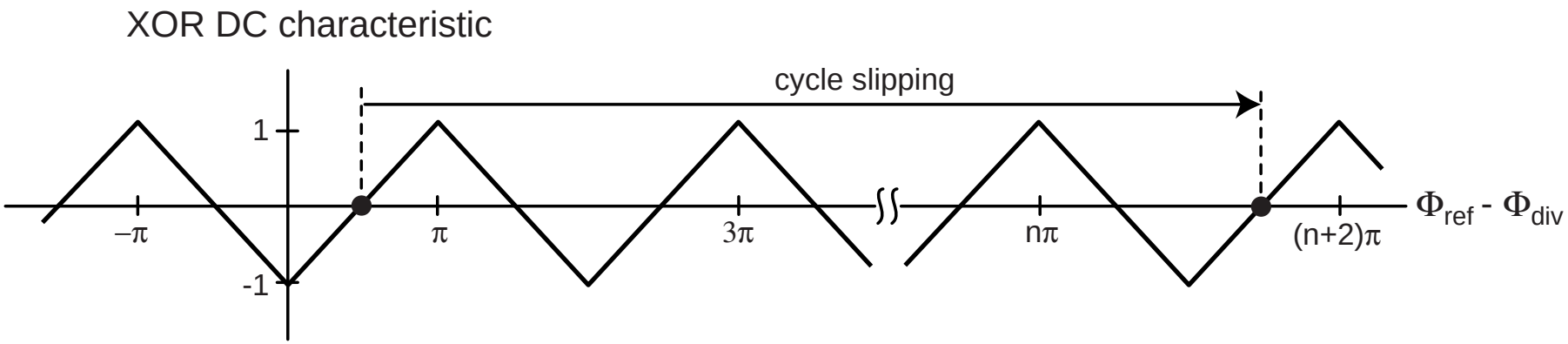


- Resulting ramp in phase causes PD characteristic to be swept across its different regions (cycle slipping)



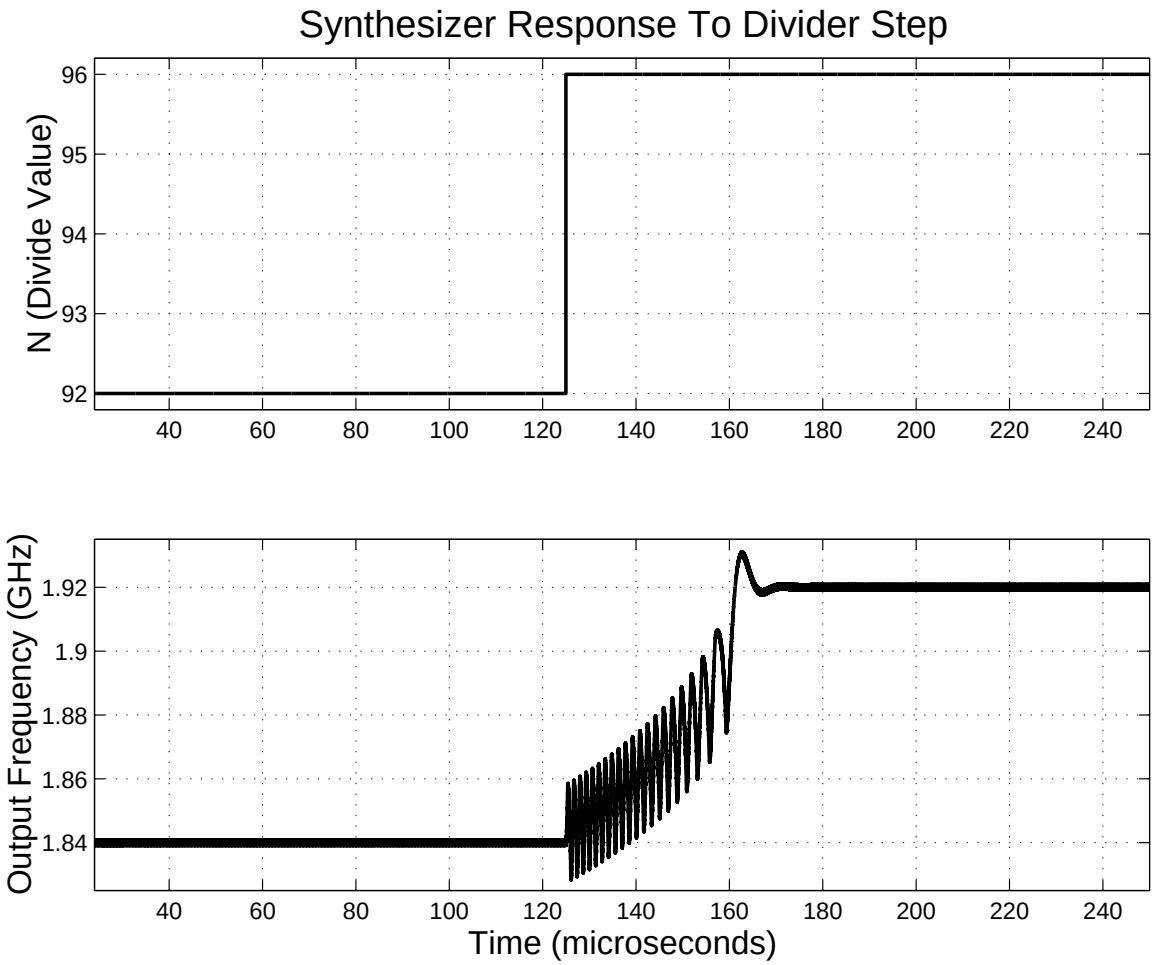
Impact of Cycle Slipping

- **Loop filter averages out phase detector output**
- **Severe cycle slipping causes phase detector to alternate between regions very quickly**
  - **Average value of XOR characteristic can be close to zero**
  - **PLL frequency oscillates according to cycle slipping**
  - **In severe cases, PLL will not re-lock**
    - PLL has finite frequency lock-in range!



Back to PLL Response Shown Previously

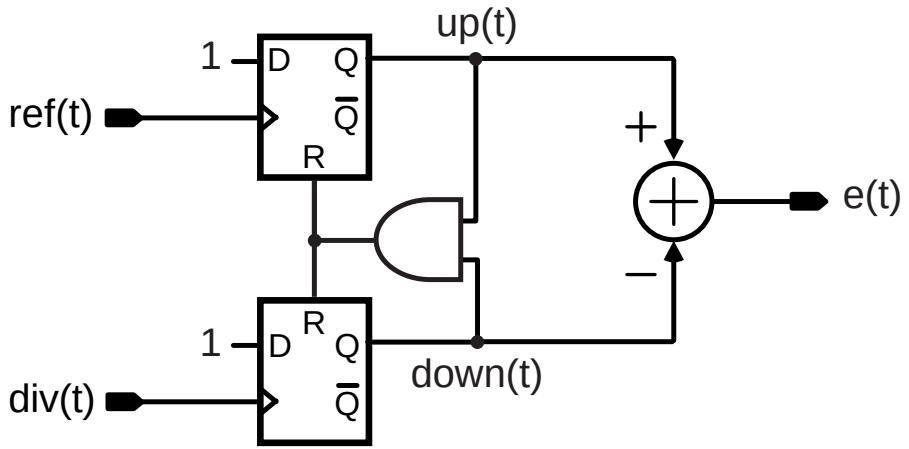
- PLL output frequency indeed oscillates
  - Eventually locks when frequency difference is small enough

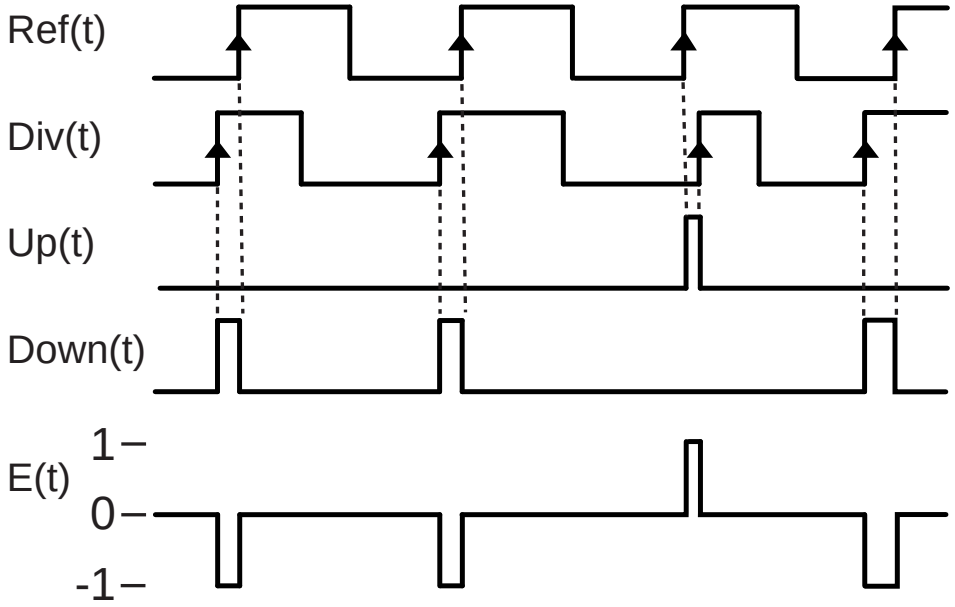


  - How do we extend the frequency lock-in range?

Phase Frequency Detectors (PFD)

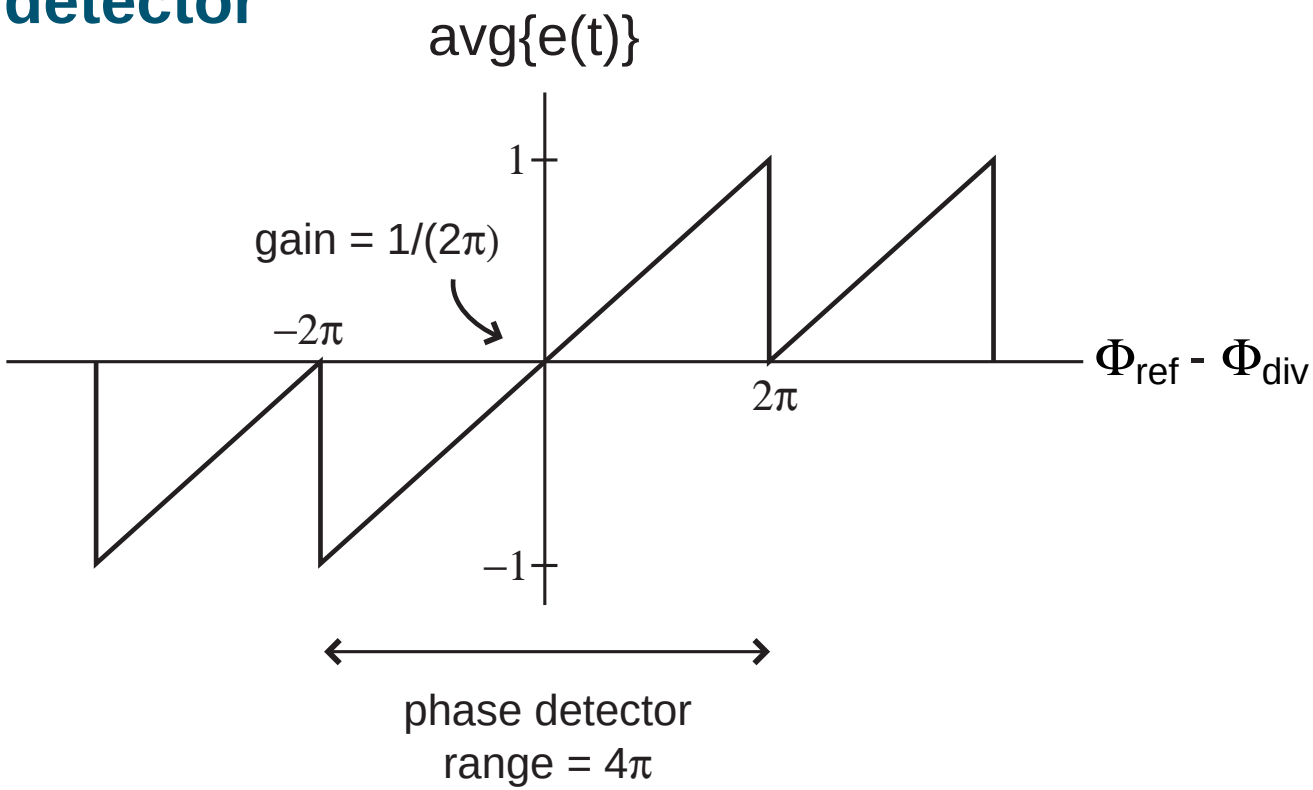
- **Example: Tristate PFD**





Tristate PFD Characteristic

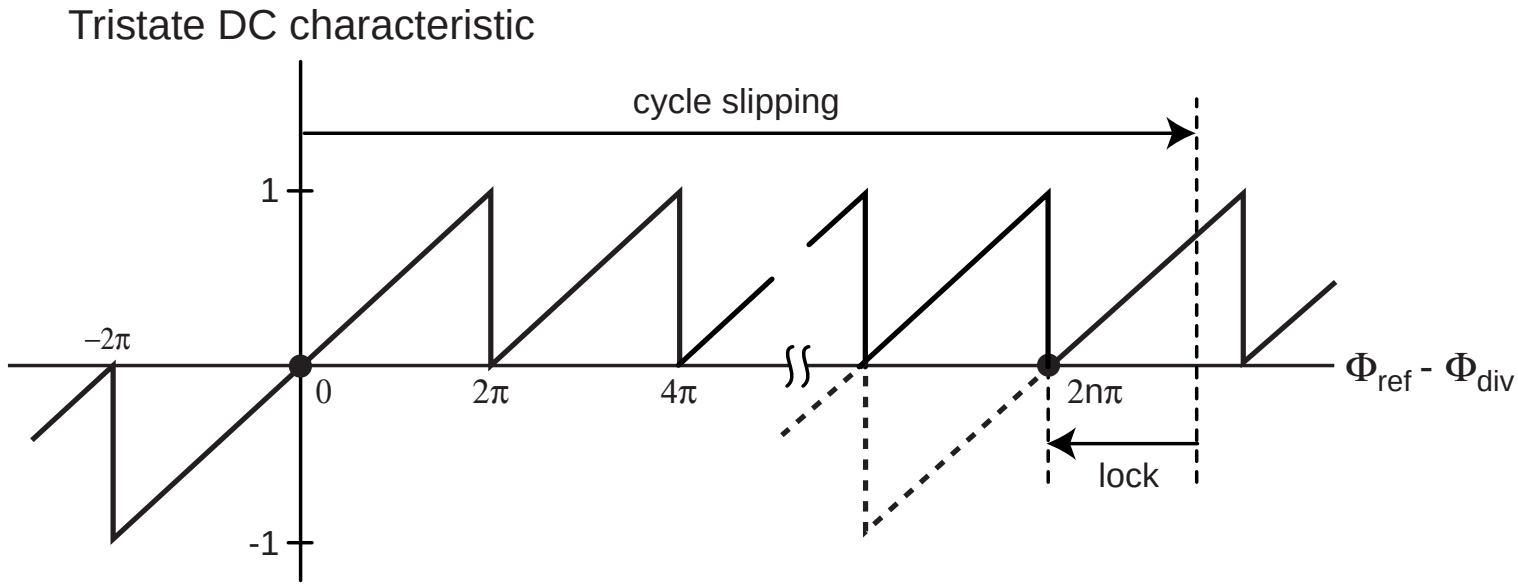
- **Calculate using similar approach as used for XOR phase detector**



- **Note that phase error characteristic is asymmetric about zero phase**
  - **Key attribute for enabling frequency detection**

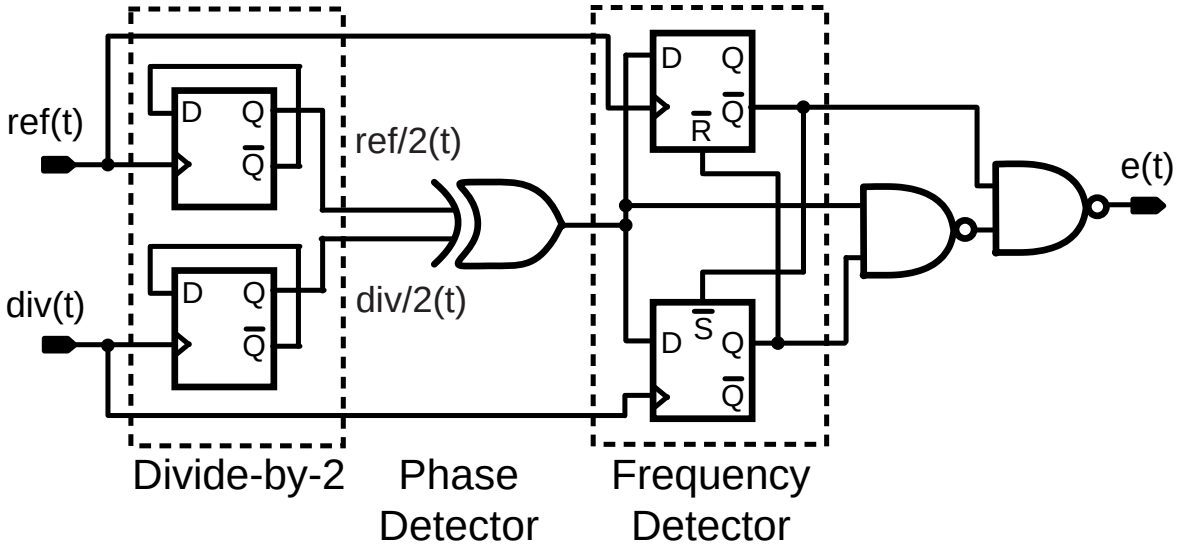
PFD Enables PLL to Always Regain Frequency Lock

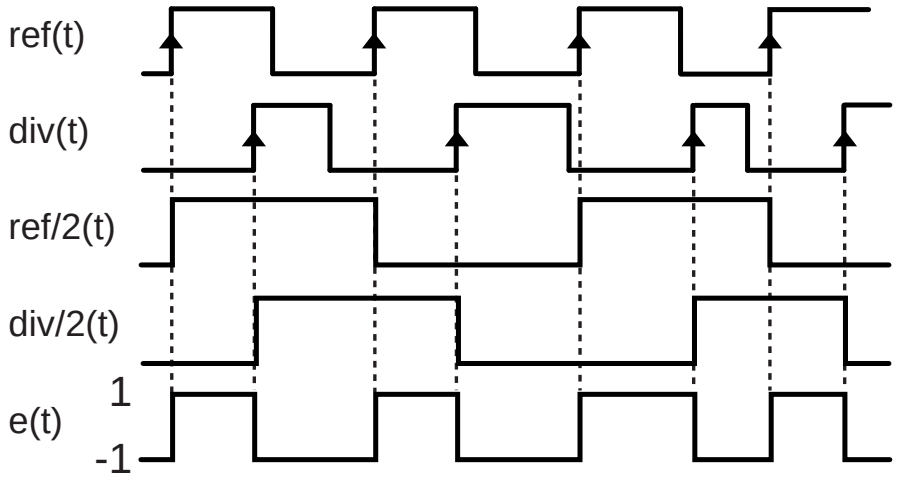
- Asymmetric phase error characteristic allows positive frequency differences to be distinguished from negative frequency differences
  - Average value is now positive or negative according to sign of frequency offset
  - PLL will always relock



Another PFD Structure

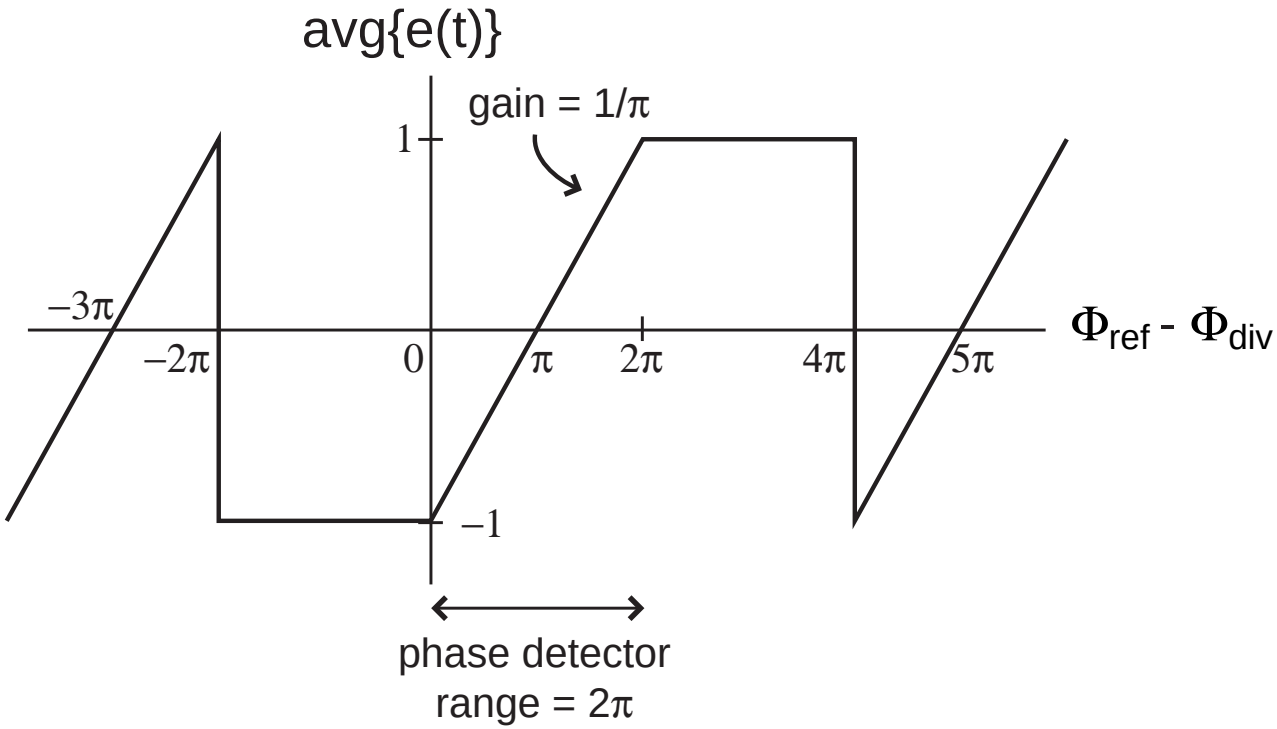
- **XOR-based PFD**





XOR-based PFD Characteristic

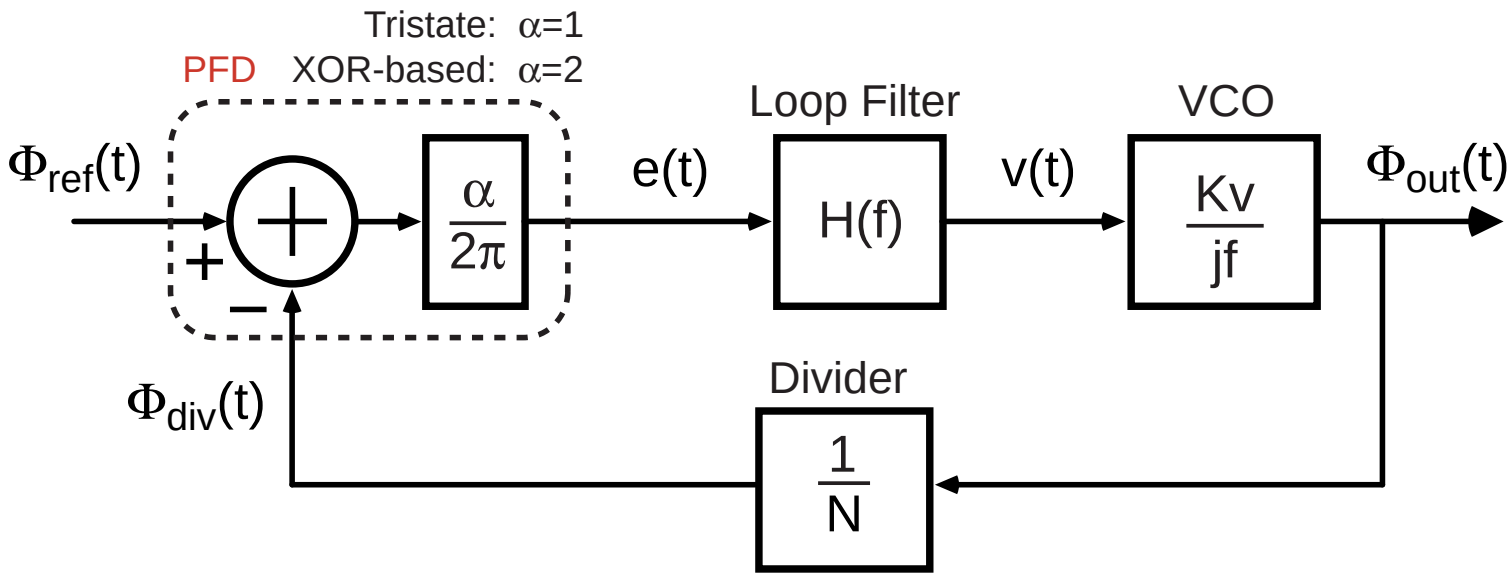
- **Calculate using similar approach as used for XOR phase detector**



- **Phase errror characteristic asymmetric about zero phase**
  - **Average value of phase error is positive or negative during cycle slipping depending on sign of frequency error**

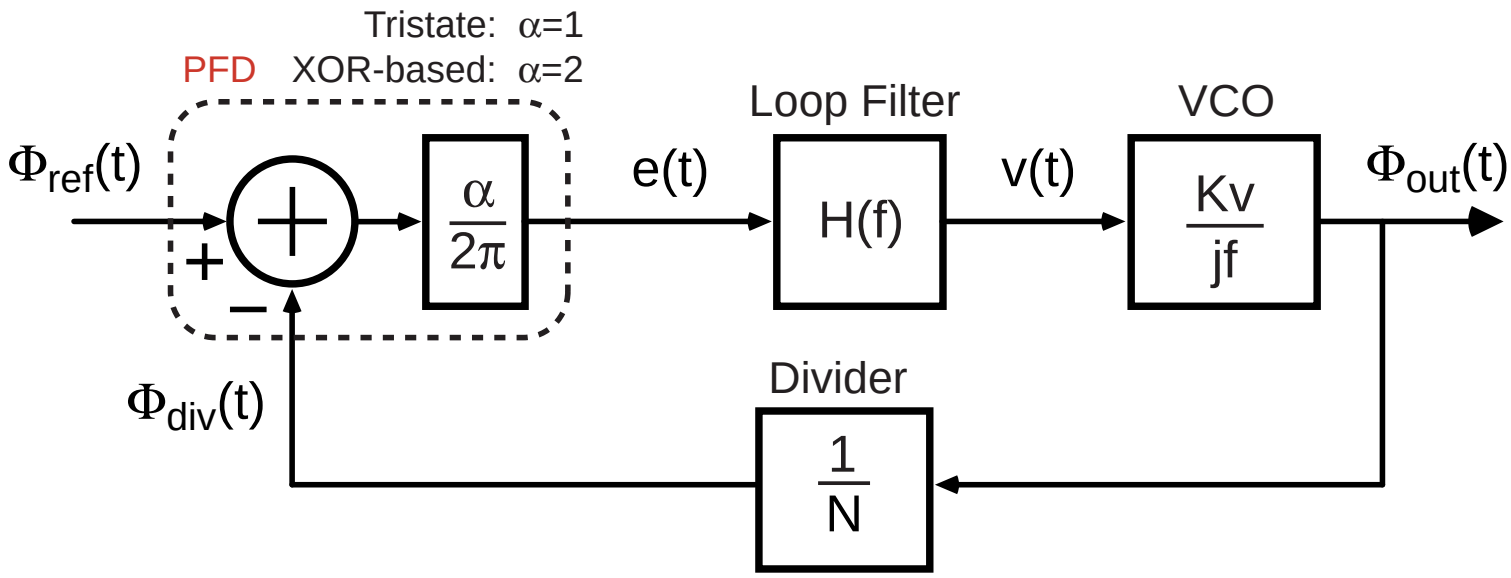
Linearized PLL Model With PFD Structures

- Assume that when PLL in lock, phase variations are within the linear range of PFD
  - Simulate impact of cycle slipping if desired (do not include its effect in model)
- Same frequency-domain PLL model as before, but PFD gain depends on topology used



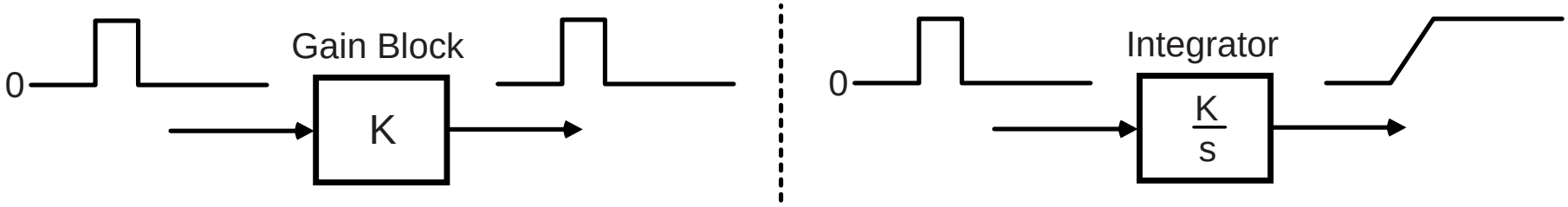
Type I versus Type II PLL Implementations

- **Type I: one integrator in PLL open loop transfer function**
  - **VCO adds on integrator**
  - **Loop filter, H(f), has no integrators**
- **Type II: two integrators in PLL open loop transfer function**
  - **Loop filter, H(f), has one integrator**

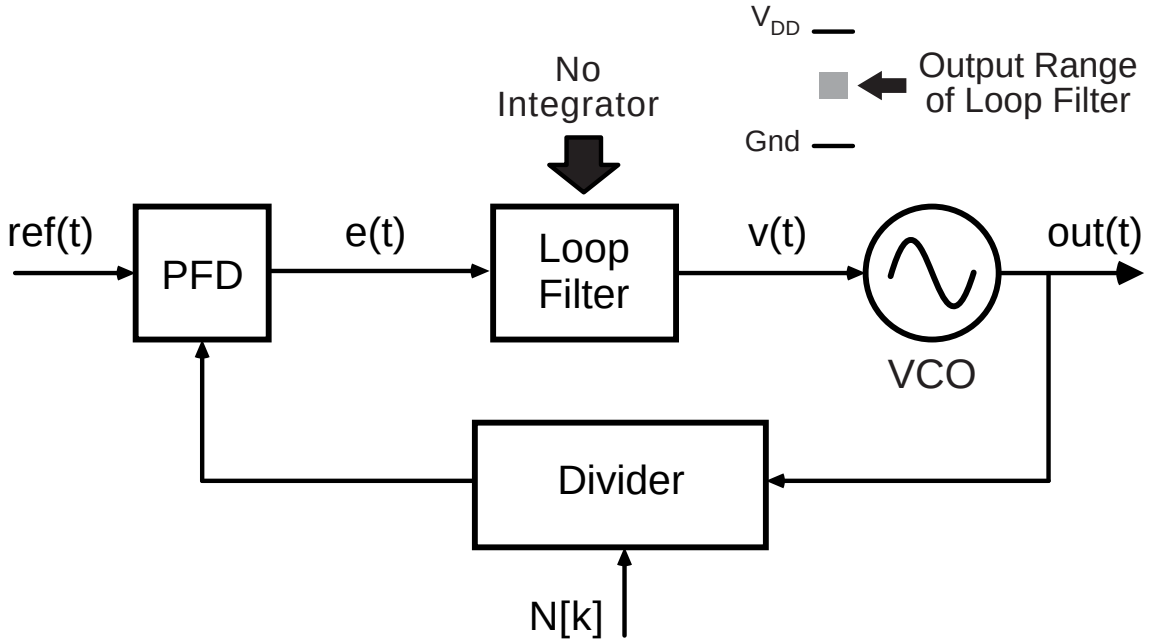


VCO Input Range Issue for Type I PLL Implementations

- DC output range of gain block versus integrator

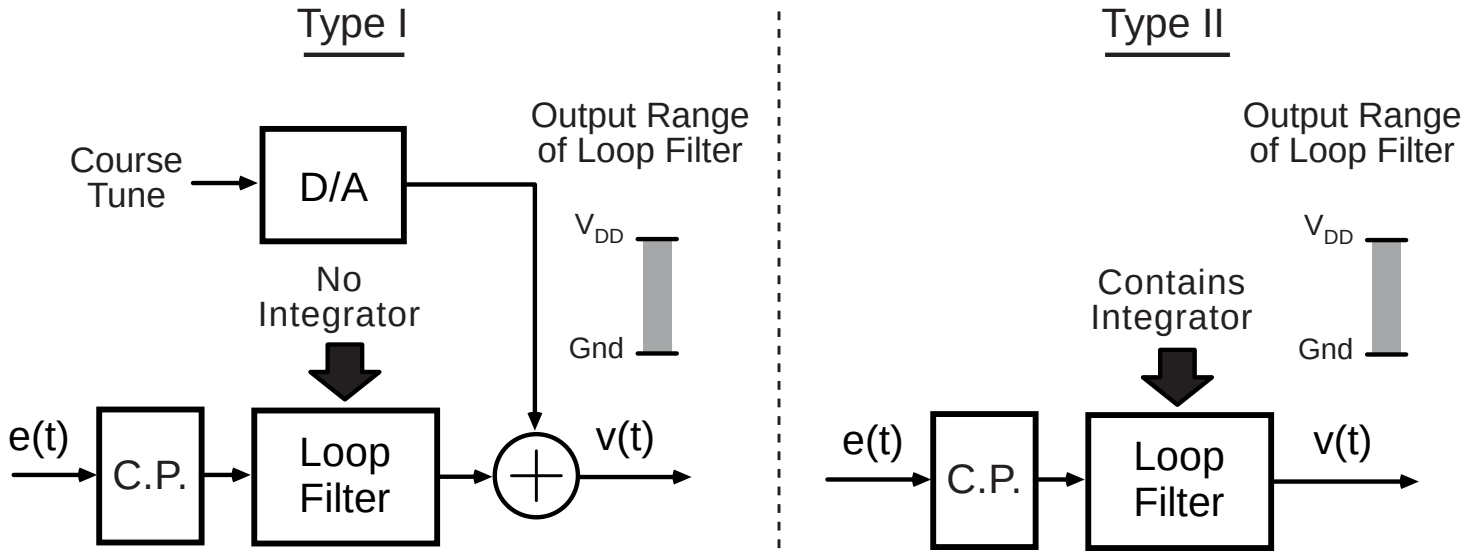


- Issue: DC gain of loop filter often small and PFD output range is limited
  - Loop filter output fails to cover full input range of VCO



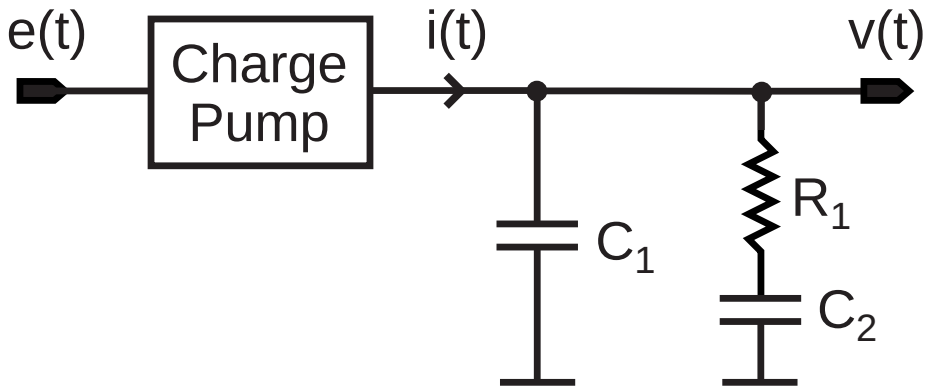
Options for Achieving Full Range Span of VCO

- **Type I**
  - **Add a D/A converter to provide coarse tuning**
    - Adds power and complexity
    - Steady-state phase error inconsistently set
- **Type II**
  - **Integrator automatically provides DC level shifting**
    - Low power and simple implementation
    - Steady-state phase error always set to zero



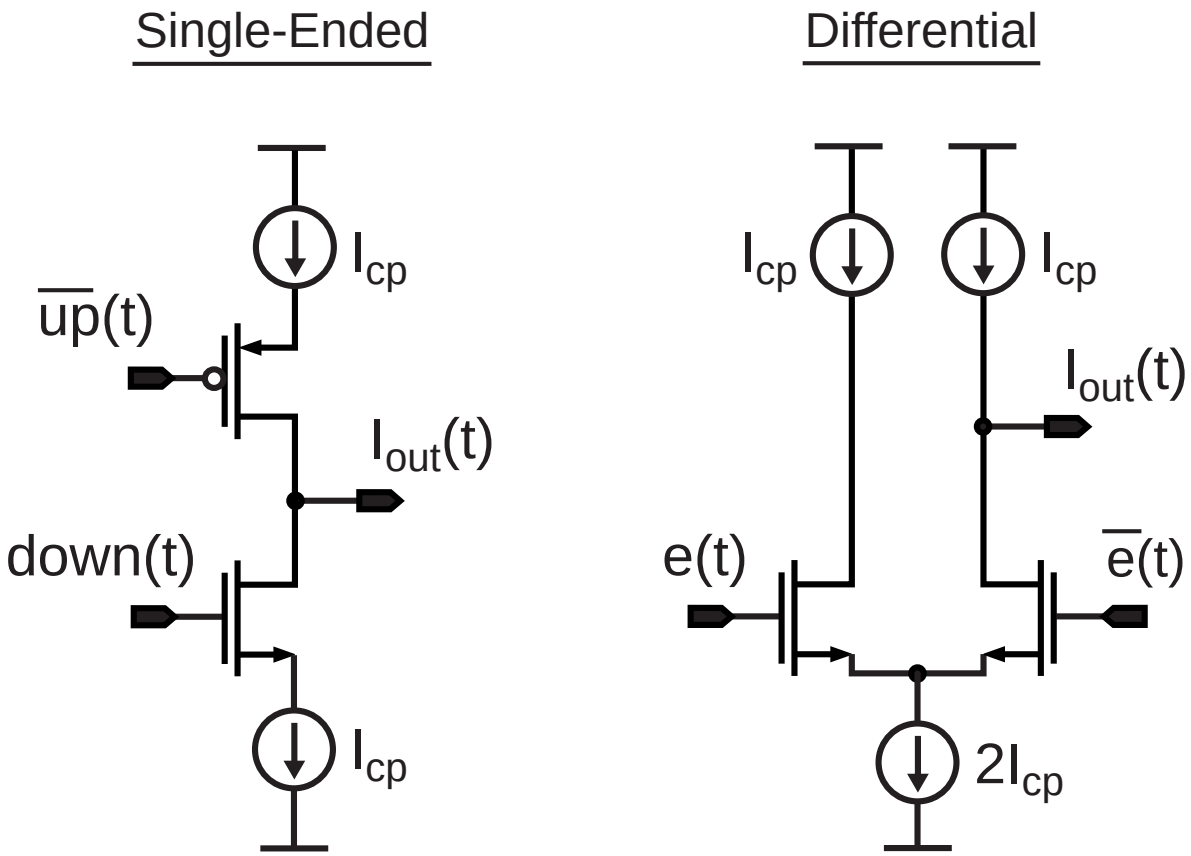
A Common Loop Filter for Type II PLL Implementation

- Use a charge pump to create the integrator
  - Current onto a capacitor forms integrator
  - Add extra pole/zero using resistor and capacitor
- Gain of loop filter can be adjusted according to the value of the charge pump current
- Example: lead/lag network



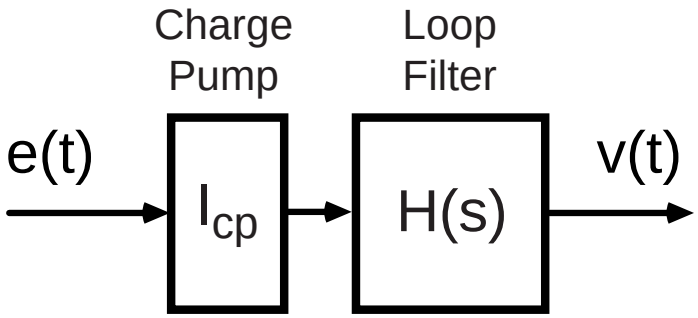
Charge Pump Implementations

- **Switch currents in and out:**



Modeling of Loop Filter/Charge Pump

- Charge pump is gain element
- Loop filter forms transfer function



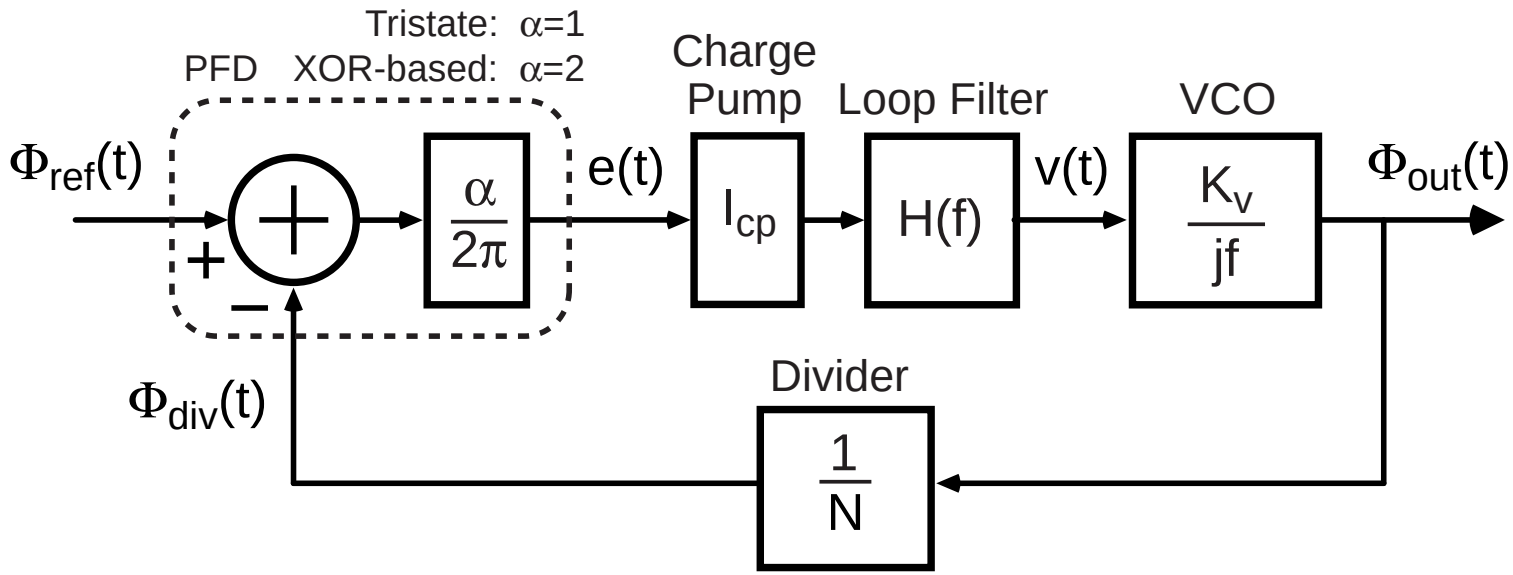
- Example: lead/lag network from previous slide

$$H(f) = \left(\frac{1}{sC_{sum}}\right)\frac{1 + jf/f_z}{1 + jf/f_p}$$

$$C_{sum} = C_1 + C_2, \quad f_z = \frac{1}{2\pi R_1 C_2}, \quad f_p = \frac{C_1 + C_2}{2\pi R_1 C_1 C_2}$$

PLL Design with Lead/Lag Filter

- **Overall PLL block diagram**

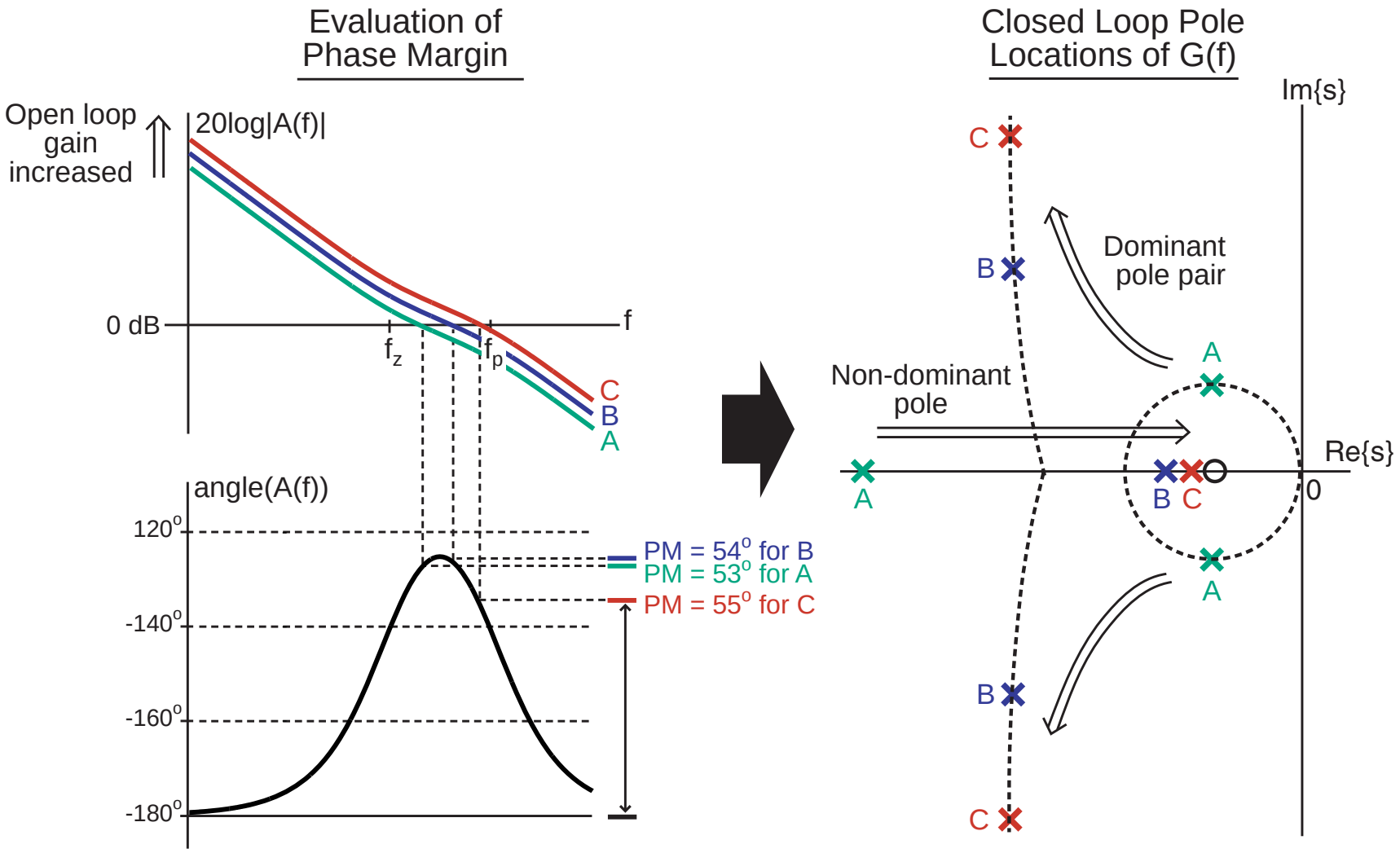


- **Loop filter**

$$H(f) = \left(\frac{1}{sC_{sum}}\right)\frac{1 + jf/f_z}{1 + jf/f_p}$$

- **Set open loop gain to achieve adequate phase margin**
  - **Set f_z lower than and f_p higher than desired PLL bandwidth**

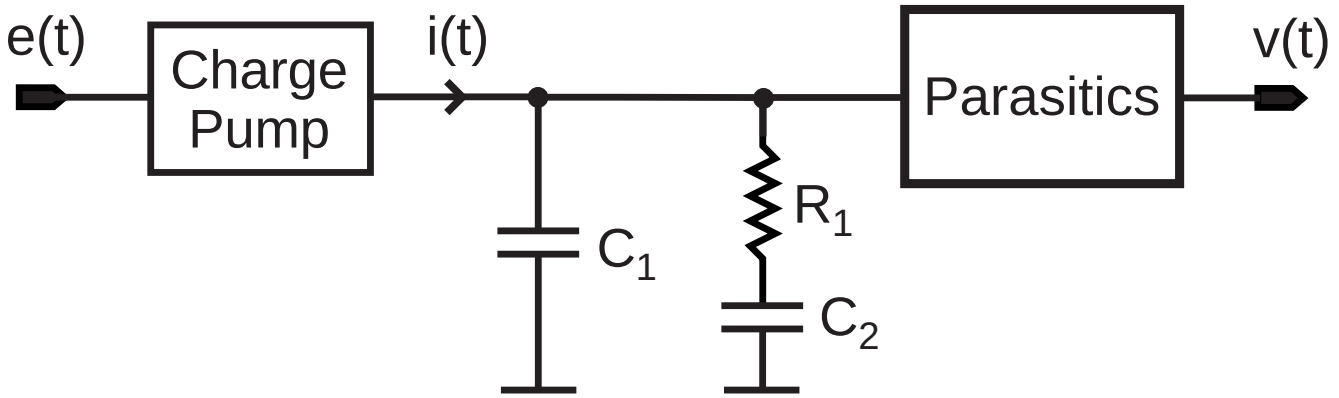
Closed Loop Poles Versus Open Loop Gain



- **Open loop gain cannot be too low or too high if reasonable phase margin is desired**

Impact of Parasitics When Lead/Lag Filter Used

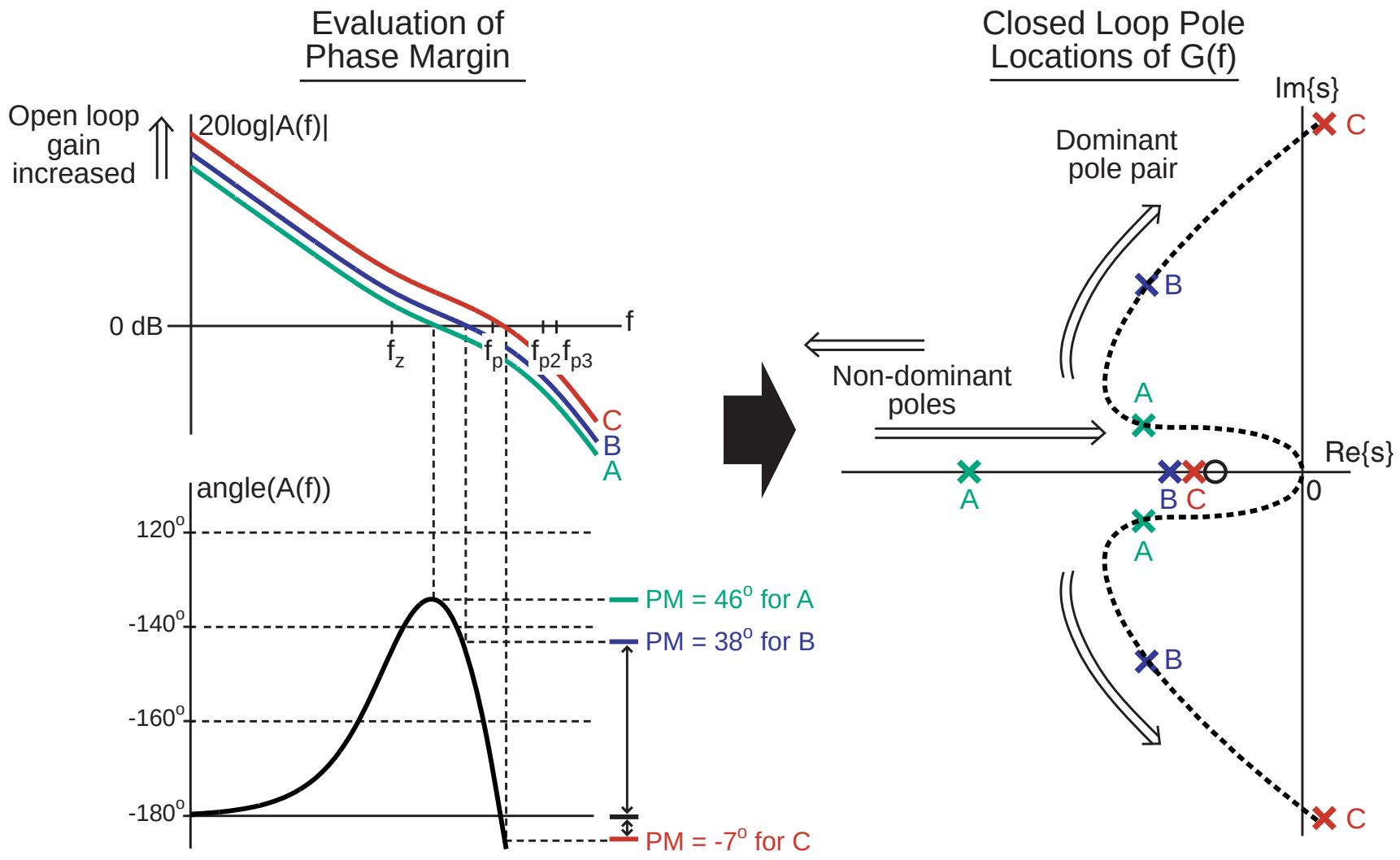
- We can again model impact of parasitics by including them in loop filter transfer function



- Example: include two parasitic poles with the lead/lag transfer function

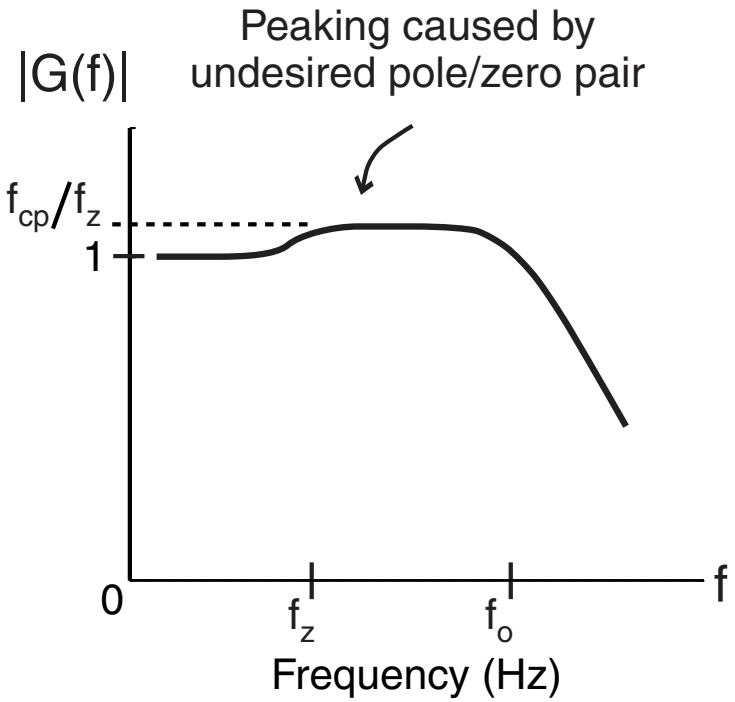
$$H(f) = \left(\frac{1}{sC_{sum}}\right)\frac{1+jf/f_z}{1+jf/f_p}\left(\frac{1}{1+jf/f_{p2}}\right)\left(\frac{1}{1+jf/f_{p3}}\right)$$

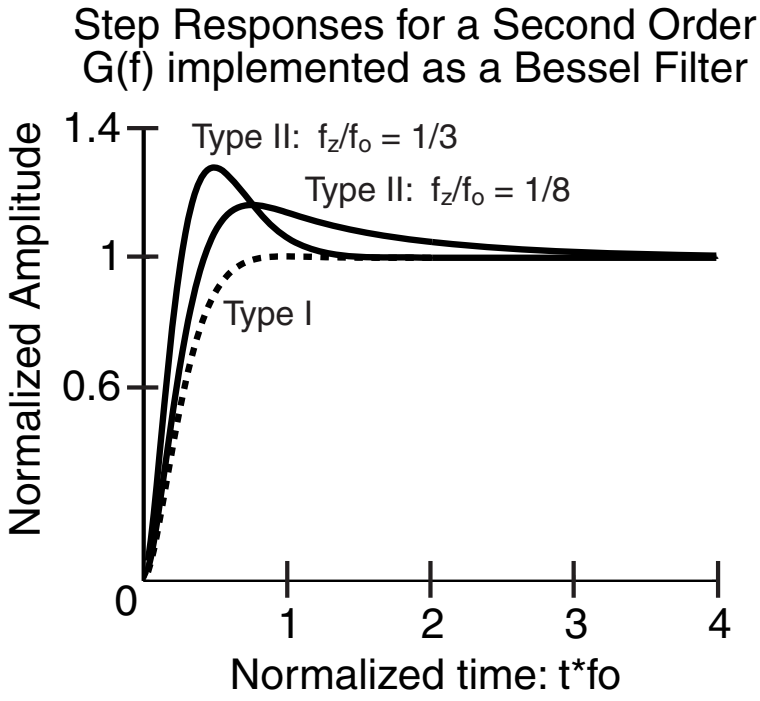
Closed Loop Poles Versus Open Loop Gain



- **Closed loop response becomes unstable if open loop gain is too high**

Negative Issues For Type II PLL Implementations





- **Parasitic pole/zero pair causes**
  - **Peaking in the closed loop frequency response**
    - A big issue for CDR systems, but not too bad for wireless
  - **Extended settling time due to parasitic "tail" response**
    - Bad for wireless systems demanding fast settling time