



Position calculation algorithm

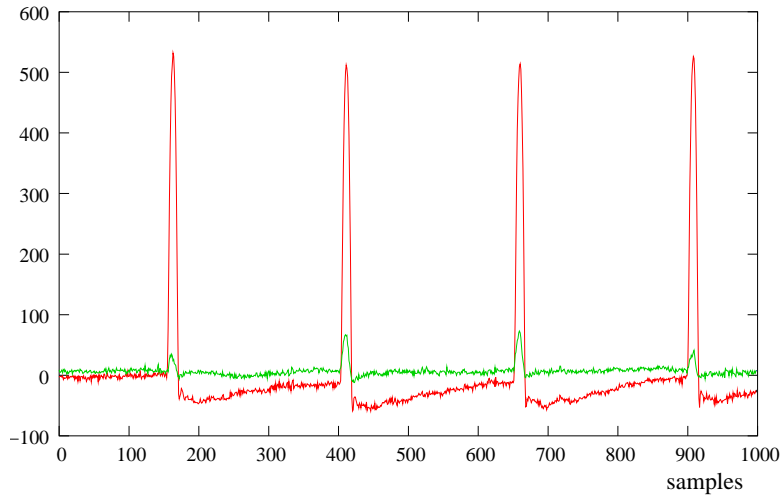
Getting the position of the centre of charge of a bunch

$$x = S_x \frac{\Delta_x}{\Sigma} + E_x$$

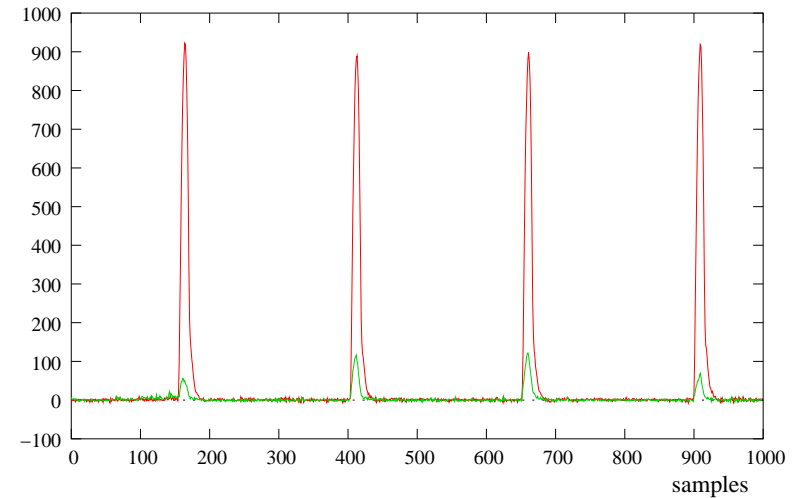
- Δ_x and Σ are the per-bunch integrals of the difference and sum signals respectively
- S_x is a proportionality constant that depends on the PU geometry and the values of Σ and Δ channel gains
- E_x is an error term that is the sum of mechanical and electrical alignment errors



Principle of base line restitution



Raw signal

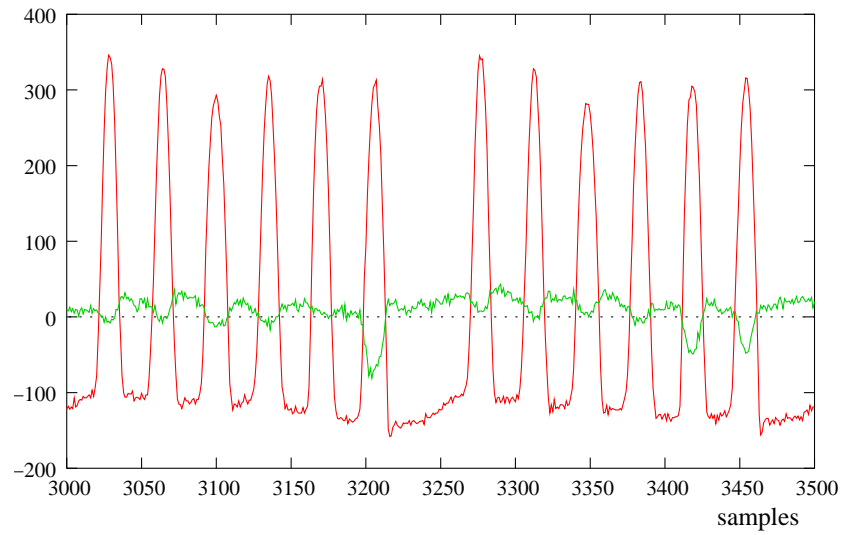


With base line restored

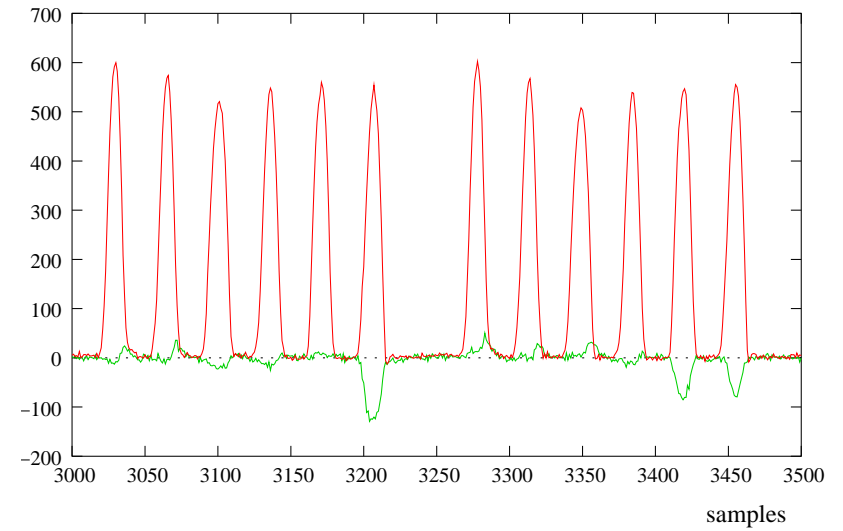
- Fullwave rectify and low-pass filter Σ to get an estimate of the baseline
- Then add that to the original Σ
- Similar for Δ , but still use Σ to get the sign of the correction



Principle of base line restitution



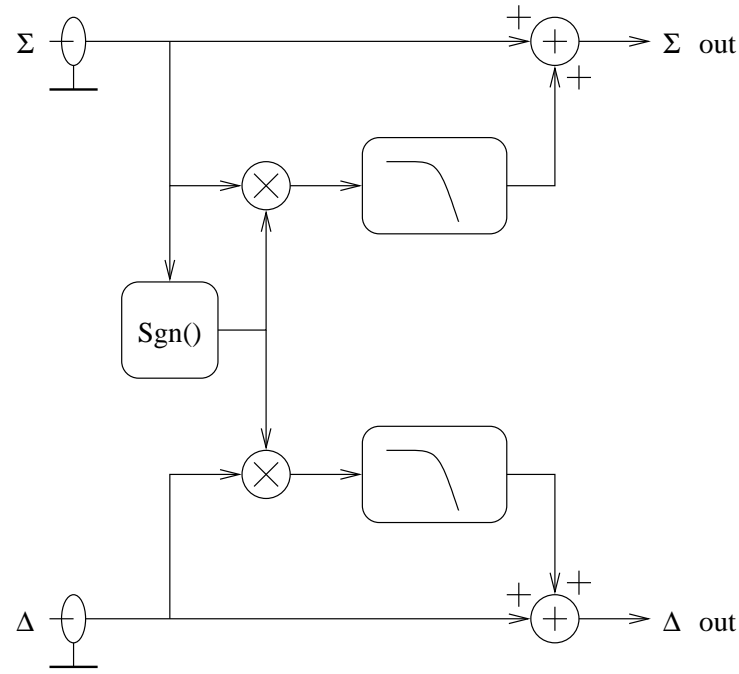
LHC beam without BLR



LHC beam with BLR



Principle of base line restitution



Block diagram of base line restorer

$$B_{\Sigma, n} = aB_{\Sigma, n-1} + (1-a)|\Sigma_n|$$

$$\Sigma_n = \Sigma_{raw} + B_{\Sigma, n}$$

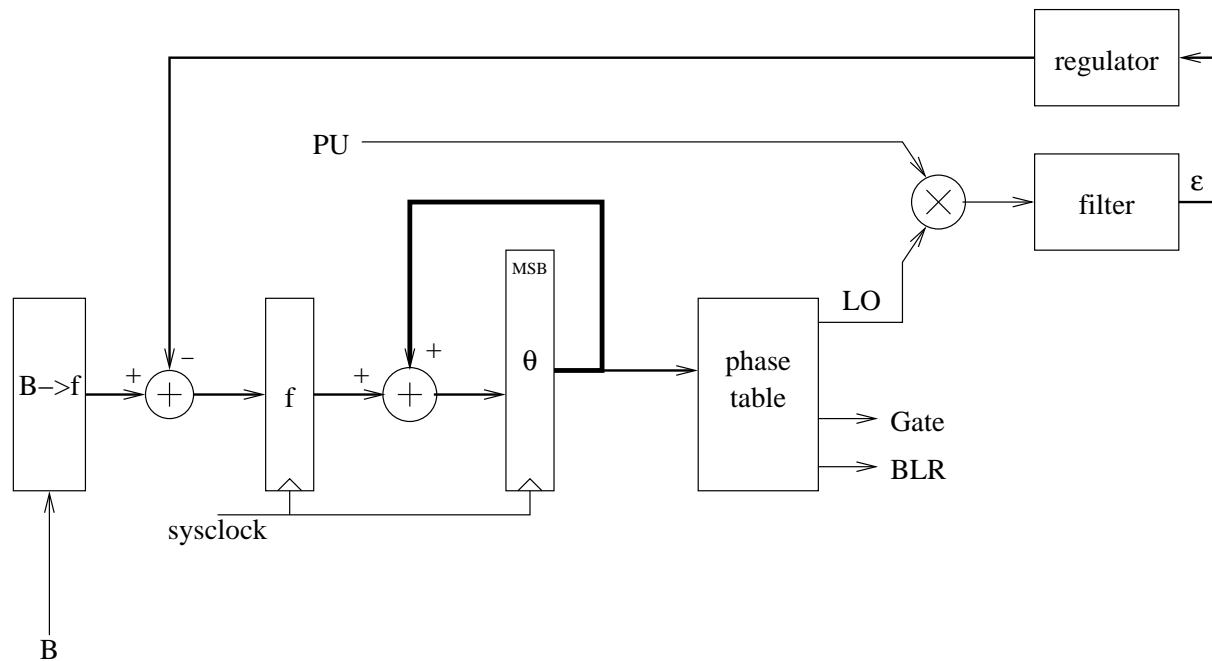
$$B_{\Delta, n} = aB_{\Delta, n-1} + (1-a)\text{sgn}(\Sigma)\Delta_{raw}$$

$$\Delta_n = \Delta_{raw} + B_{\Delta, n}$$



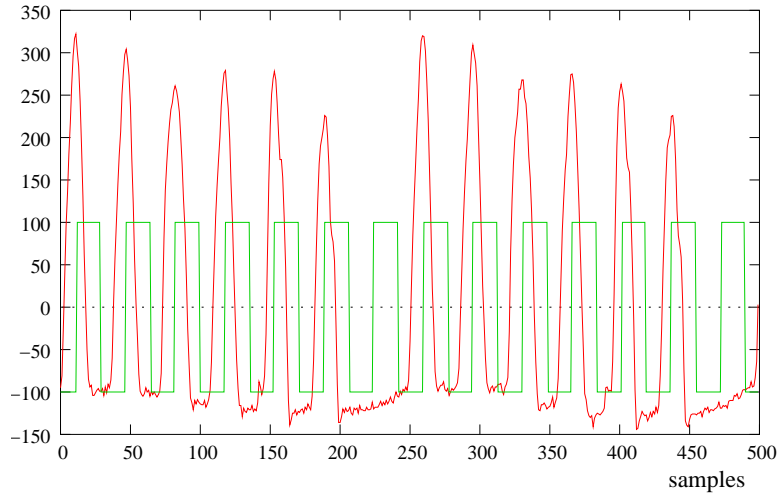
Creating a reference frequency

- Numerical phase locked loop
- DDS running at F_{rev}
- Lookup table generates LO and Gate

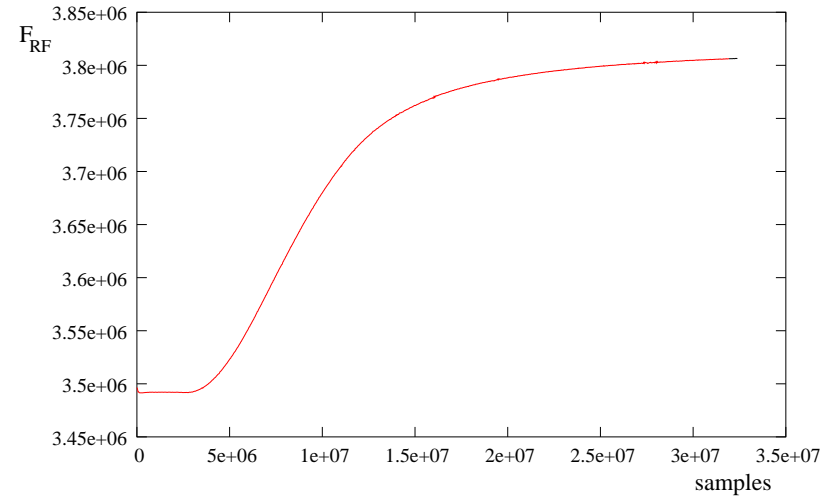




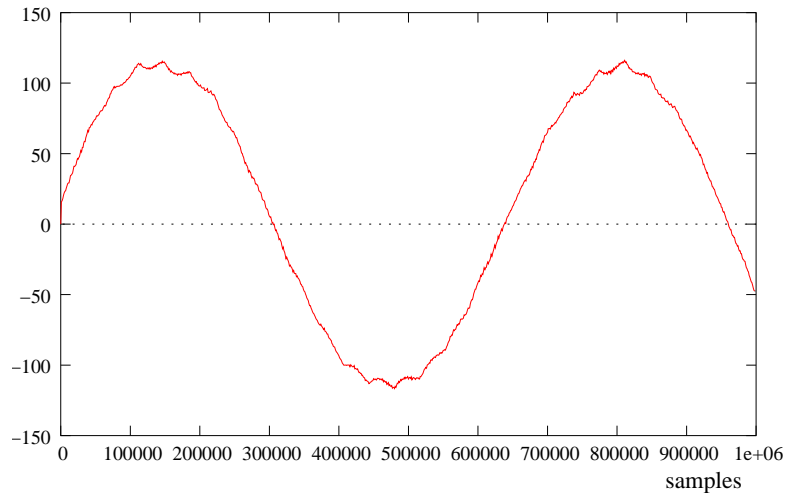
Creating a reference frequency



PU signal and reconstructed RF at h=7



Evolution of RF frequency during acceleration

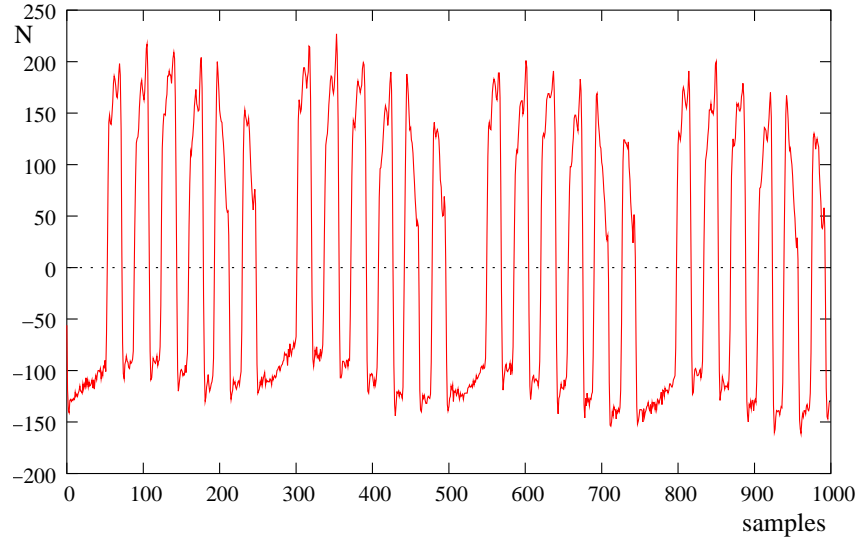


Phase error (ϵ) vs. PU and RF phase difference

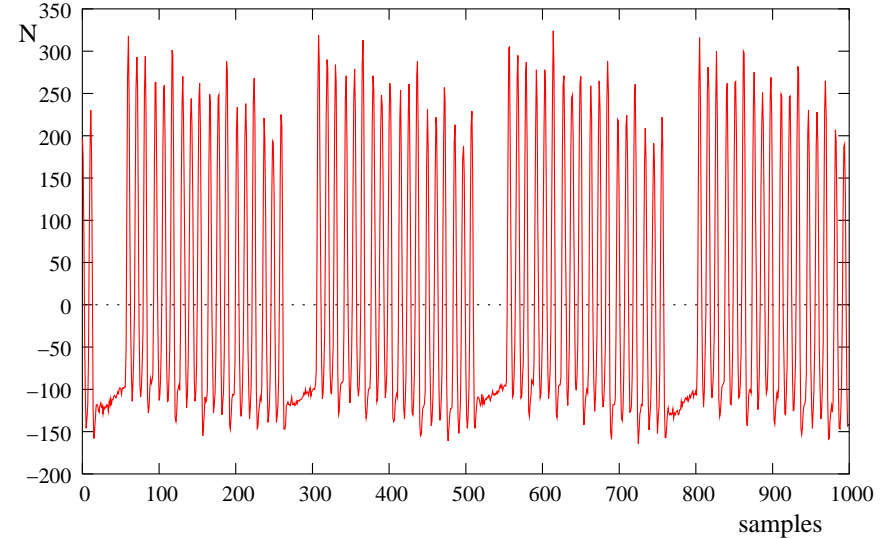
There is a trade off between settling time and accuracy:

- Too slow and it won't follow acceleration
- Too fast and the reconstructed RF will be noisy

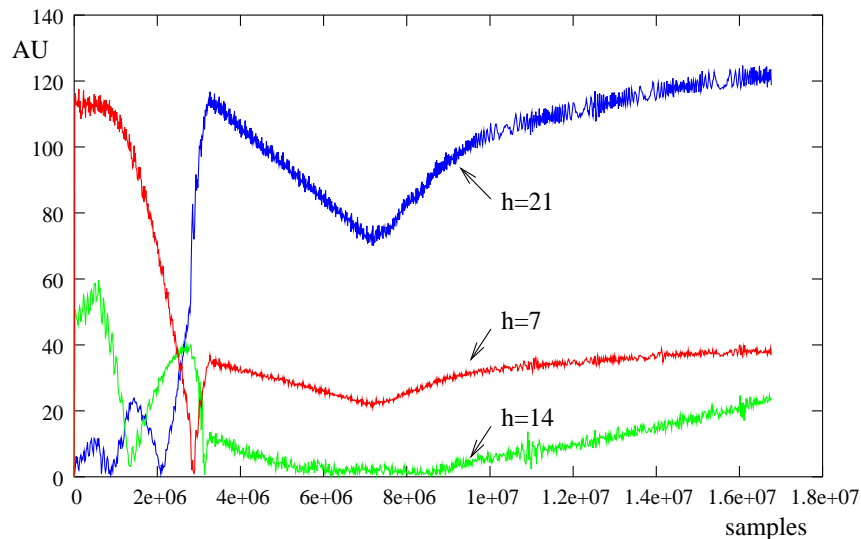
(Past experience indicates that 20-100 μ s is about right)



LHC beam at h=7



LHC beam at h=21



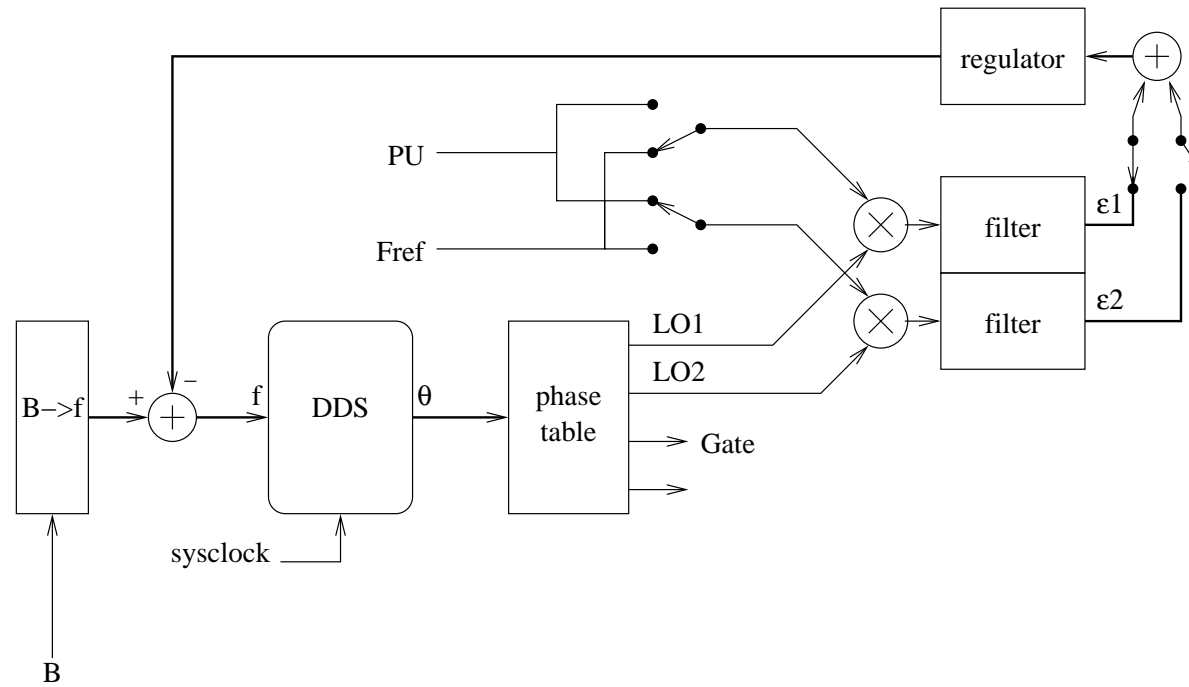
Evolution of magnitude of harmonics on LHC

RF gymnastics in PS have special requirements:

- Choose signal from several possible sources
- Produce several LO harmonic numbers
- Produce appropriate gate timings
- Switch from one to another dynamically
- **WITHOUT LOSING LOCK!**



Dealing with RF gymnastics



Switches are controlled by machine timing system



Concluding remarks

The algorithm described has been implemented in C-programs, has been tested with real PU data and works quite well.

BUT:

- The algorithm has been expressed in floating point
- The FPGA is much better at handling integers

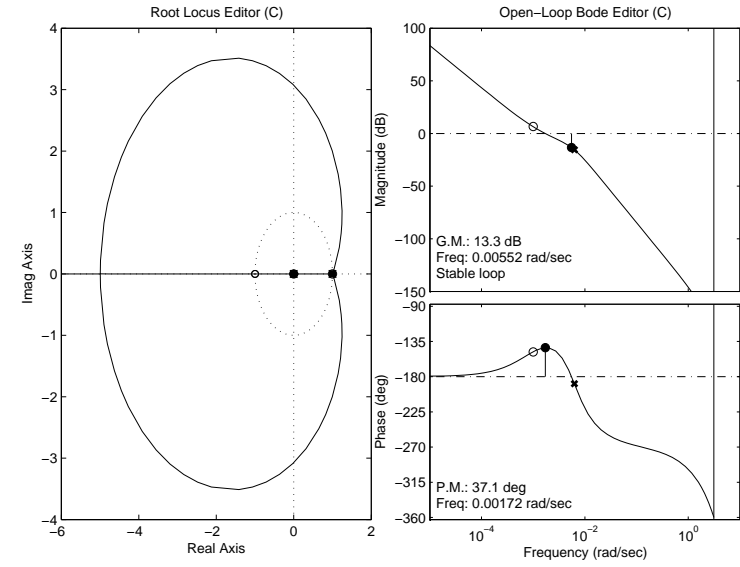
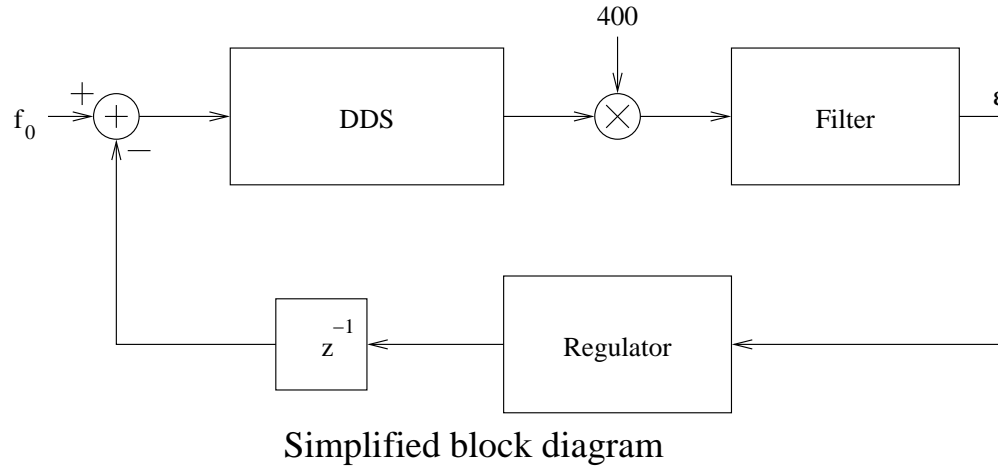
- It assumes only one clock delay per iteration
- Multiplications require several clock cycles

- Some filter parameter values are awkward

—▶ There is work to be done to make the algorithm fit the FPGA



PLL response analysis



Root locus and Bode plots

DDS

$$H_{dds} = \frac{h}{2^{32}} \frac{z^{-1}}{1-z^{-1}}$$

Mixer

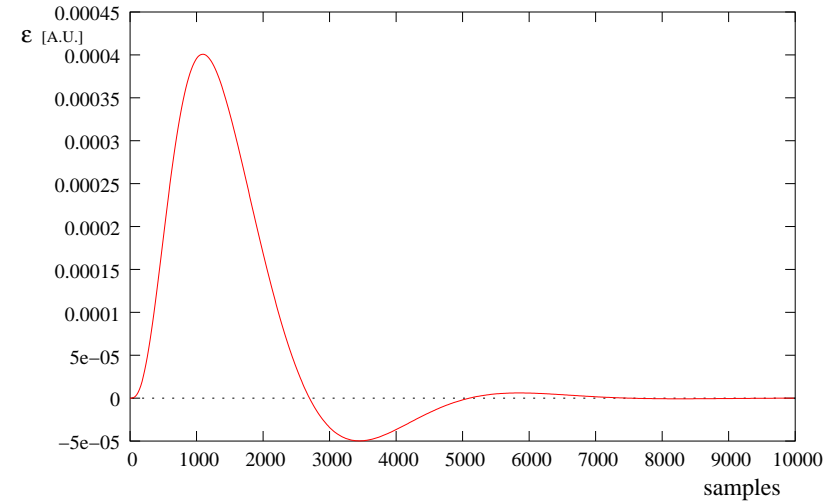
$$H_m = 400$$

Low-pass filter

$$H_F = 9.8 \cdot 10^{-6} \frac{1+2z^{-1}+z^{-2}}{1-1.9911z^{-1}+0.9911z^{-2}}$$

Regulator

$$H_R = K_R \cdot z^{-1} \cdot \frac{1-0.999z^{-1}}{(1-z^{-1})}$$



Step (in ϵ) response