Fast(er) Difficulty Adjustment for Secure Sidechains

Mark Friedenbach
<mark@friedenbach.org>

work in collaboration with
an anonymous contributor

The problem

- Bitcoin protocol adjusts difficulty every 2016 blocks, which nominally happens every 2 weeks.
- Bitcoin hash rate is relatively stable. A “slow” adjustment algorithm works fine, so far.
- But what about low probability events, or attacks not previously seen on mainnet? We must plan for the future!

The problem

- Alternative networks experience rapid hashrate fluctuation: 10x flux on short (<1 day!) timescales is not unheard of, when there exists hardware capable of being re-purposed for other networks (e.g. CPUs, GPUs, sha256 and scrypt chains).

- Merged mined sidechains or extension blocks are expected to experience large shocks in hashrate in current ecosystem as mining pools add or remove support, or during active attacks.

- Bitcoin may experience a large shock event in some unlikely, but possible scenarios e.g. of Internet access to China being shut off (war?). While unlikely, this would be a situation where Bitcoin is needed most...

The problem

What would happen if the hashrate experienced a 10x reduction?

- Transaction throughput reduced to 10% of normal: 0.3tps.
- Up to 4½ *months* until relief from the first difficulty adjustment downward, almost 6 months until normalcy.
- Reduction of throughput and lack of response leads to drop in price, dropping price leads to dropping profitability for remaining miners, and a viscous “hash crash” cycle of further hashrate reductions ensues.

This exact scenario has been the common experience of many alternative networks. It *could* happen to Bitcoin too.

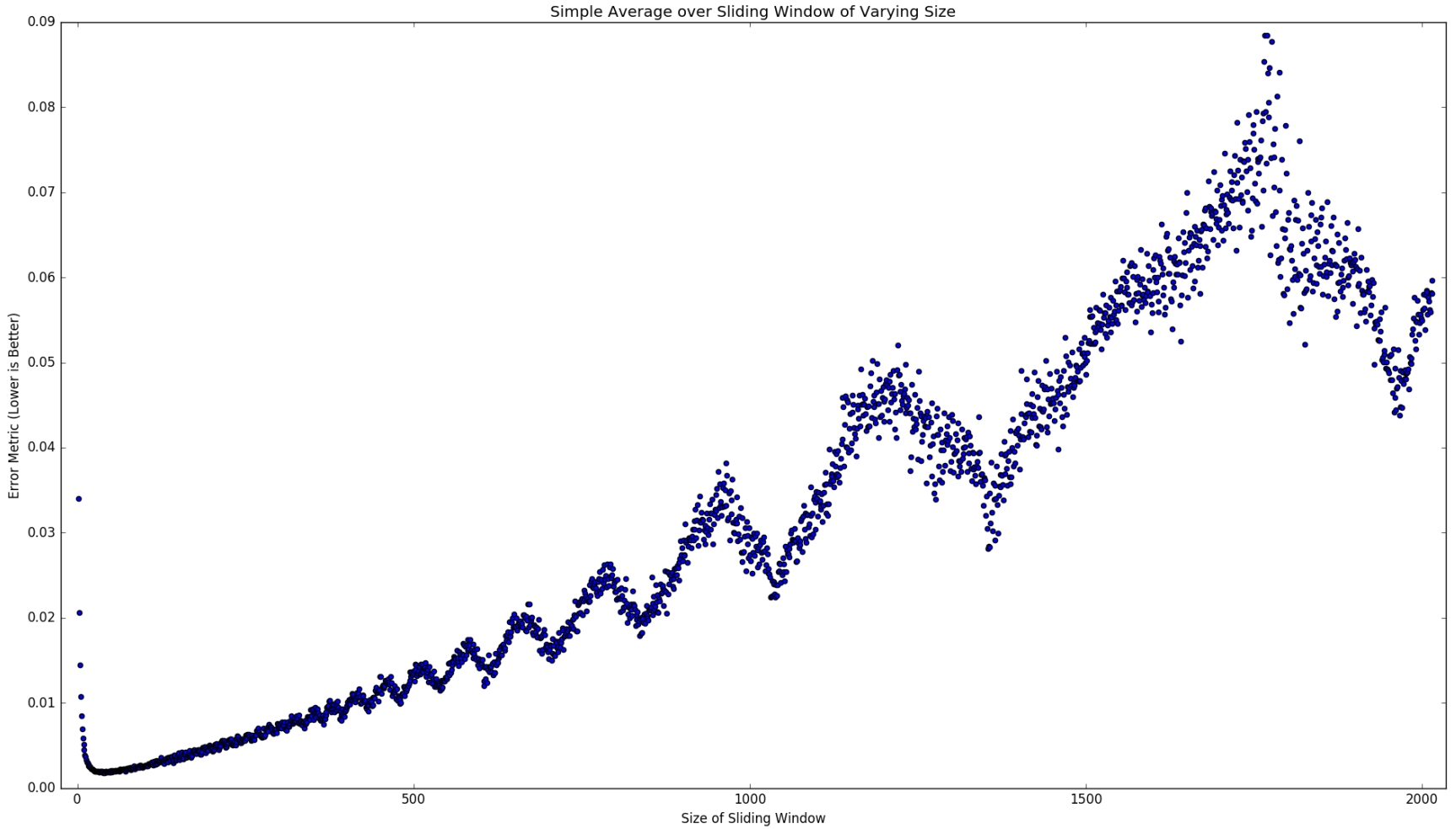
The solutions

- Adjust more frequently.
  - Why not adjust every block? Or at least every N, for N << 2016.
    Early adjustment has compounding effects, allowing quick response to a shock.

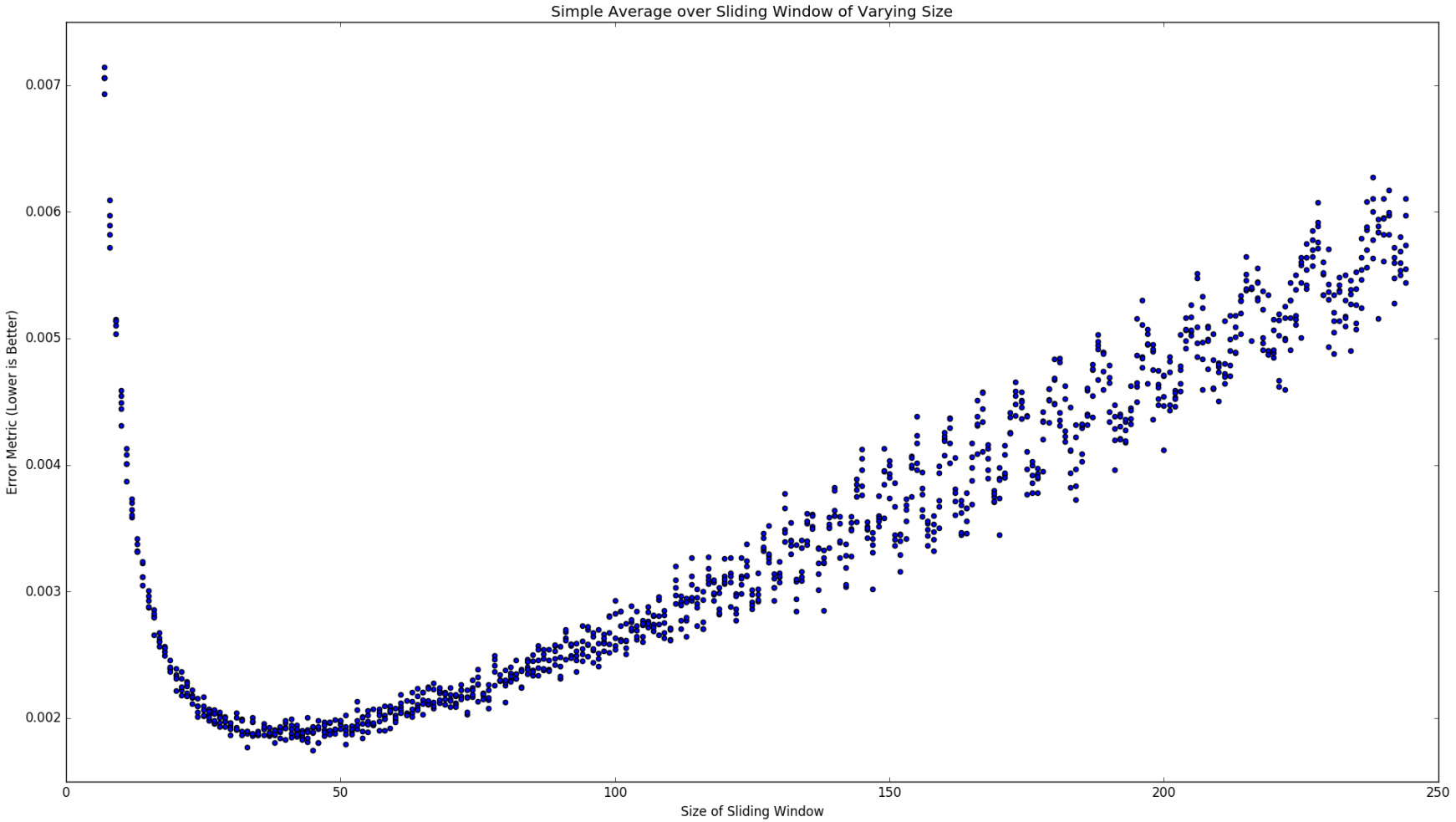
- Decouple sliding window from adjustment frequency.
  - Size of sliding window affects how quickly the controller responds to a hashrate shock.
    Note that how frequently the adjustment is performed, and how large of a sliding window is used are mostly-independent knobs to tweak, even if in Bitcoin they are almost the same (2016 and 2015, respectfully).

- Change the control algorithm.
  - Bitcoin uses an extremely simple, naïve control algorithm.
    Control theory literature is full of better alternatives...

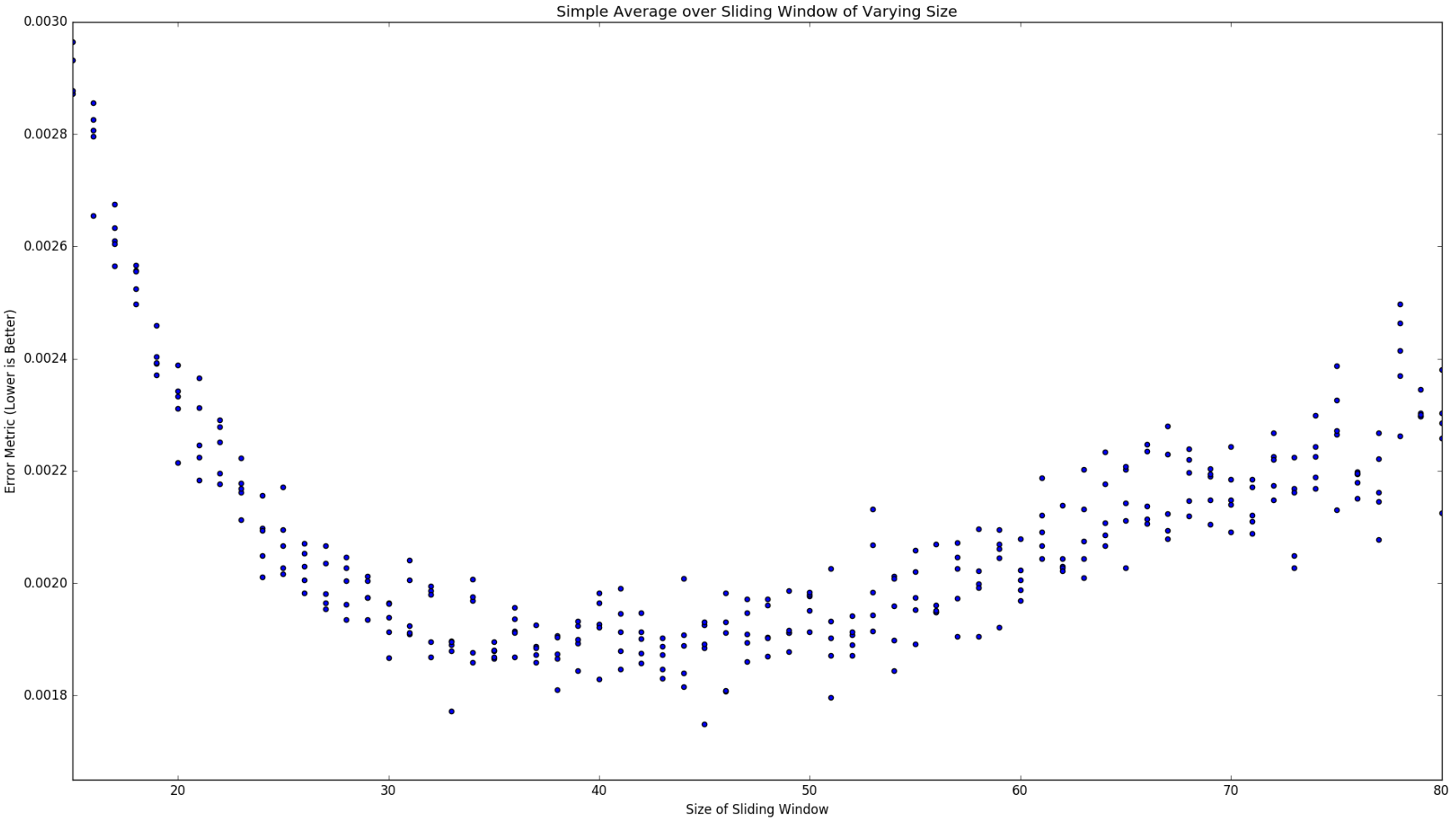
Adjust more frequently?



Adjust more frequently?



Adjust more frequently?



Adversarial aside…

Bitcoin has an incentive problem encouraging miners to lie about timestamps of blocks before and just after a difficulty adjustment.

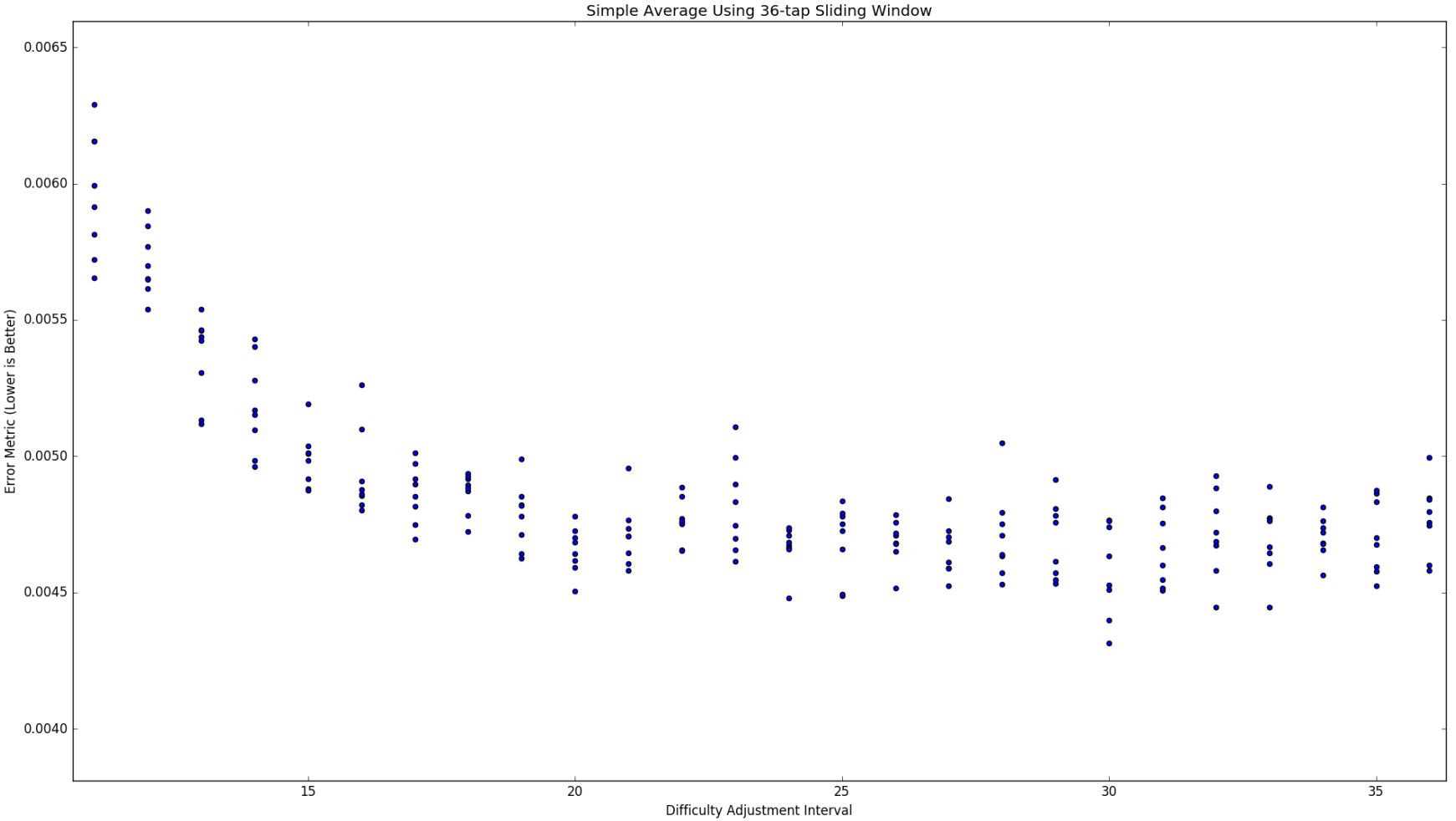
Doing so results in a easier (~1%) difficulty than 10 minute blocks with actual hashrate would require, resulting in higher revenue per unit time.

Shortening the adjustment interval makes this *worse*.

However this instance of the problem is mostly solved by making sure that the sliding window is equal to or larger than the adjustment interval, which is not the case for bitcoin.

We will revisit this issue shortly...

Decouple frequency and interval



The problem, revisited

There is a variable, “hashrate” which we sample once per block.

The samples are subject to significant random noise.

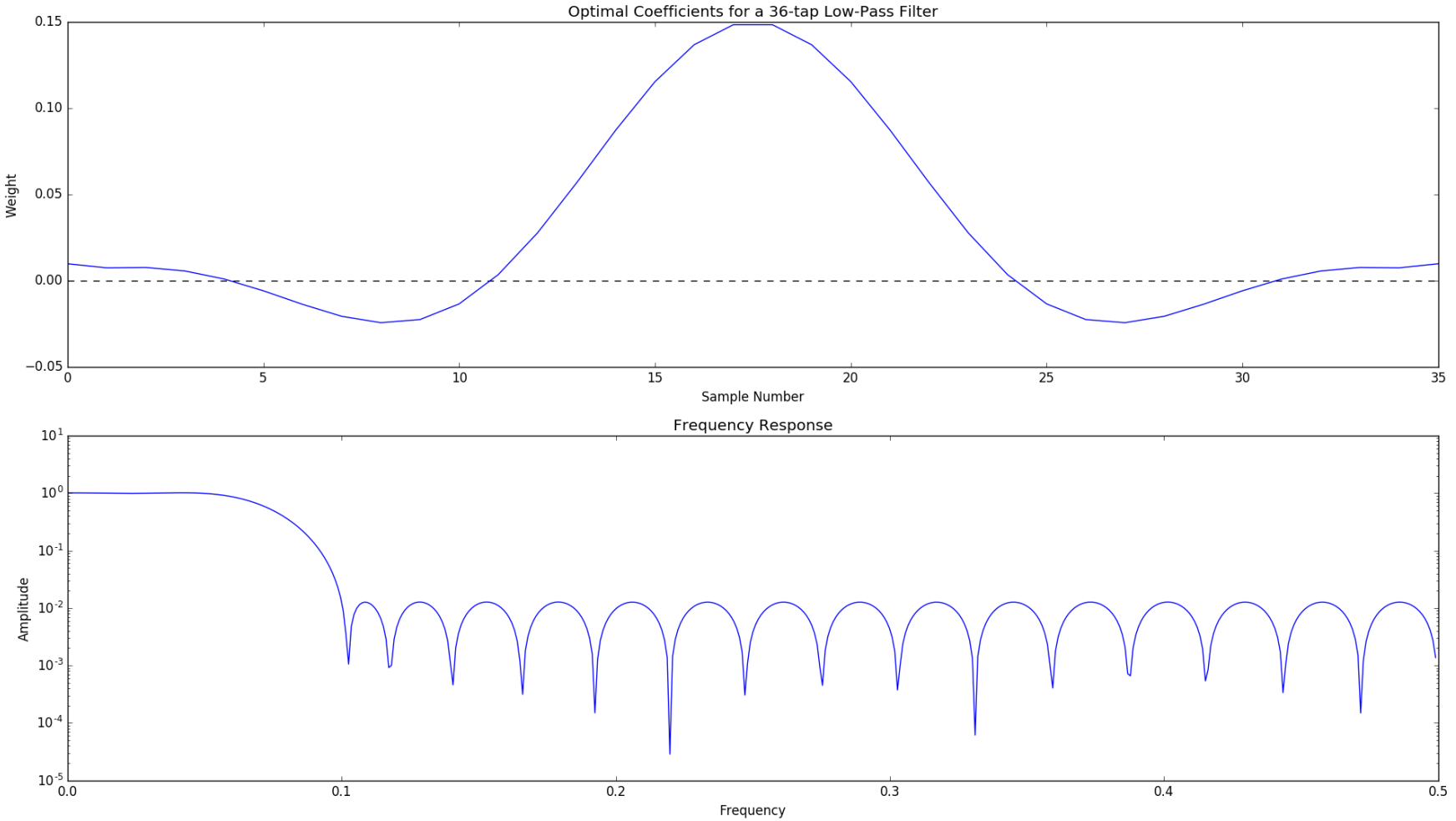
We wish to ignore the high-frequency noise to isolate and respond to the long-term trend: by how much is the variable being sampled generally going up or down.

This is generally known as a *low-pass filter*.

Thankfully there are methods for generating provably optimal filter designs… e.g. the Parks–McClellan filter design algorithm:

```
scipy.signal.remez(36, [0, 0.05, 0.1, 0.5], [1, 0])
```

Low-pass filter design



Low-pass filter, continued

Adds two more tweakable parameters: *gain* and *limiter*. Optimal selections for both based on the shown 36-tap linear filter, validated by simulation:

Gain: increase or decrease is modulated by this factor.

```
gain = 0.125
```

Limiter: maximum increase or decrease allowed per adjustment.

```
limiter = 1.375
```

Questions

mark@friedenbach.org