SAND2015-1211C

Controlling qubit drift by recycling error correction syndromes

Robin Blume-Kohout

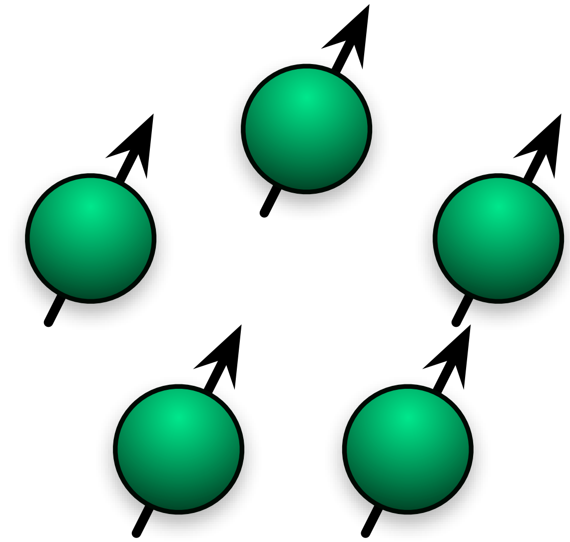
(Sandia National Labs)

U.S. DEPARTMENT OF ENERGY

Sandia National Laboratories

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

The Situation:



You have some qubits. You'd like to do something with them. Like run a quantum circuit. Maybe error correction.

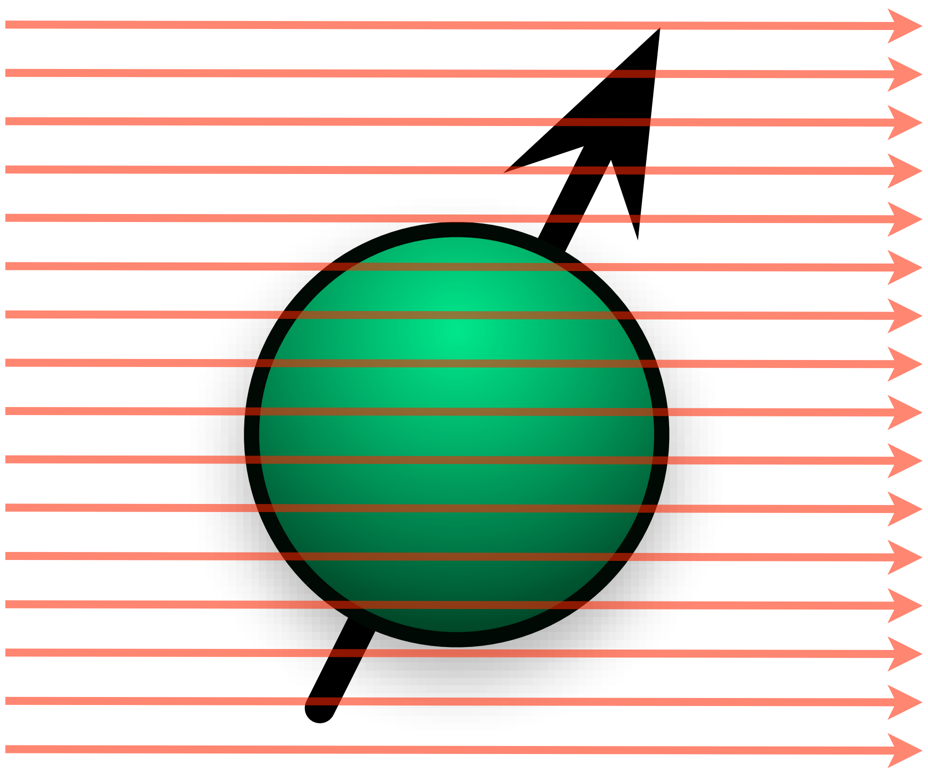
Problem 1: Your unitary gates aren't stable! Control parameters **drift** over time. Error rates rise. This is bad.

Problem 2: Dynamical decoupling (DCG) only works if your gates are *already* pretty good (need decent π pulses).

Solution: You're already doing error correction, right? Here's how to compensate drift using QEC syndrome data.

Hamiltonians drift ☹

$\mathbf{H} = B(t)\sigma_x$



H causes errors. It is unknown *and* changes in time.

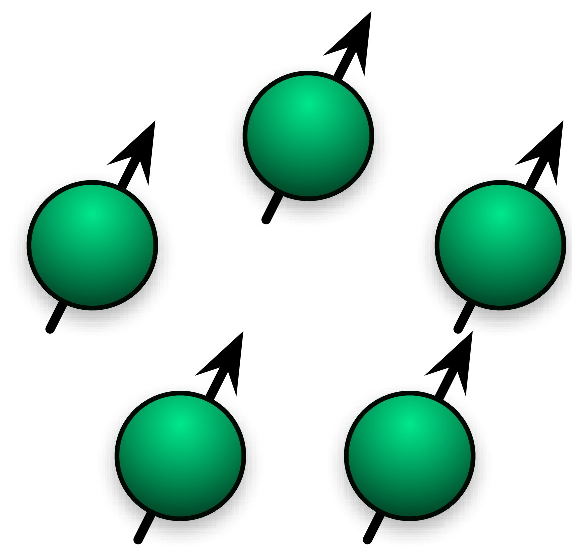
DD/DCG (dynamical decoupling) can help -- but only within limits.

If we *knew* $B(t)$, we could just compensate it away!

KNOWLEDGE IS POWER

The Basic Idea

I'll assume your qubits are engaged in error correction.



Every *cycle*, you measure stabilizers.
Result = *syndrome*.
Syndromes => what error happened.

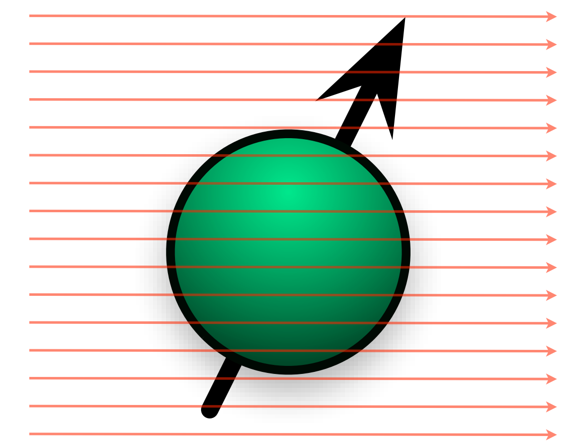
By monitoring this stream of syndromes,
we can infer which qubits are going bad... and fix them.

→ *time* →

Disclaimer: all results in this talk are THEORY.

A simple example

1. <u>One</u> physical qubit in a [1,0,1] code.



2. Hamiltonian fluctuates as $H = B(t)\sigma_x$
   where B(t) = random walk.

$$\frac{\mathrm{d}}{\mathrm{d}t}\left\langle B^2 \right\rangle = \eta$$

3. We measure σ_z every *cycle* (Δt)
   and check for *errors*.

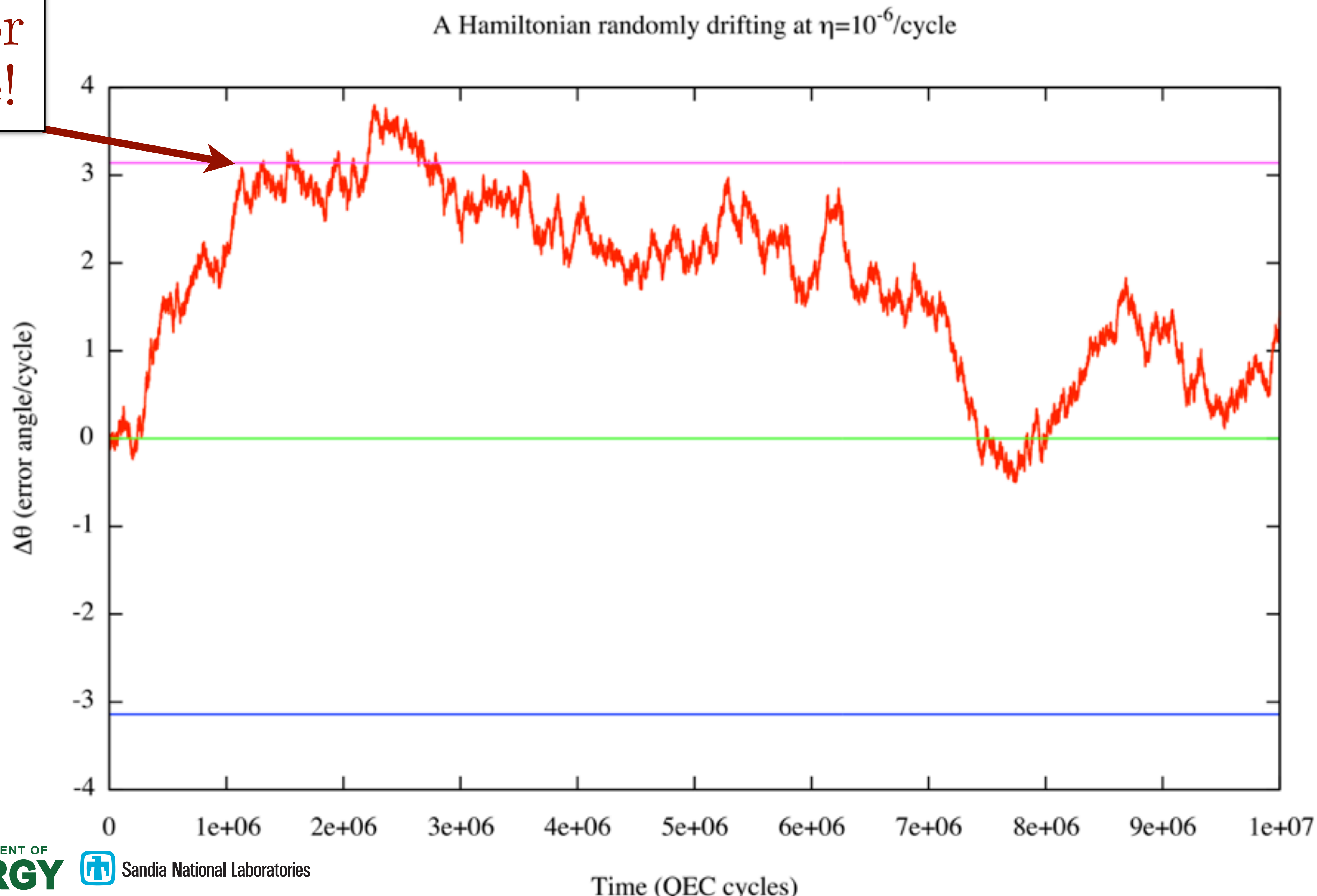
4. The probability of an error in a cycle is:

$$\begin{aligned} P_{err} &= \sin^2(\Delta\theta) \\ \Delta\theta &= B(t) \cdot \Delta t \end{aligned}$$

6. For now, assume errors are *only* caused by H.

Uncompensated B(t) drift

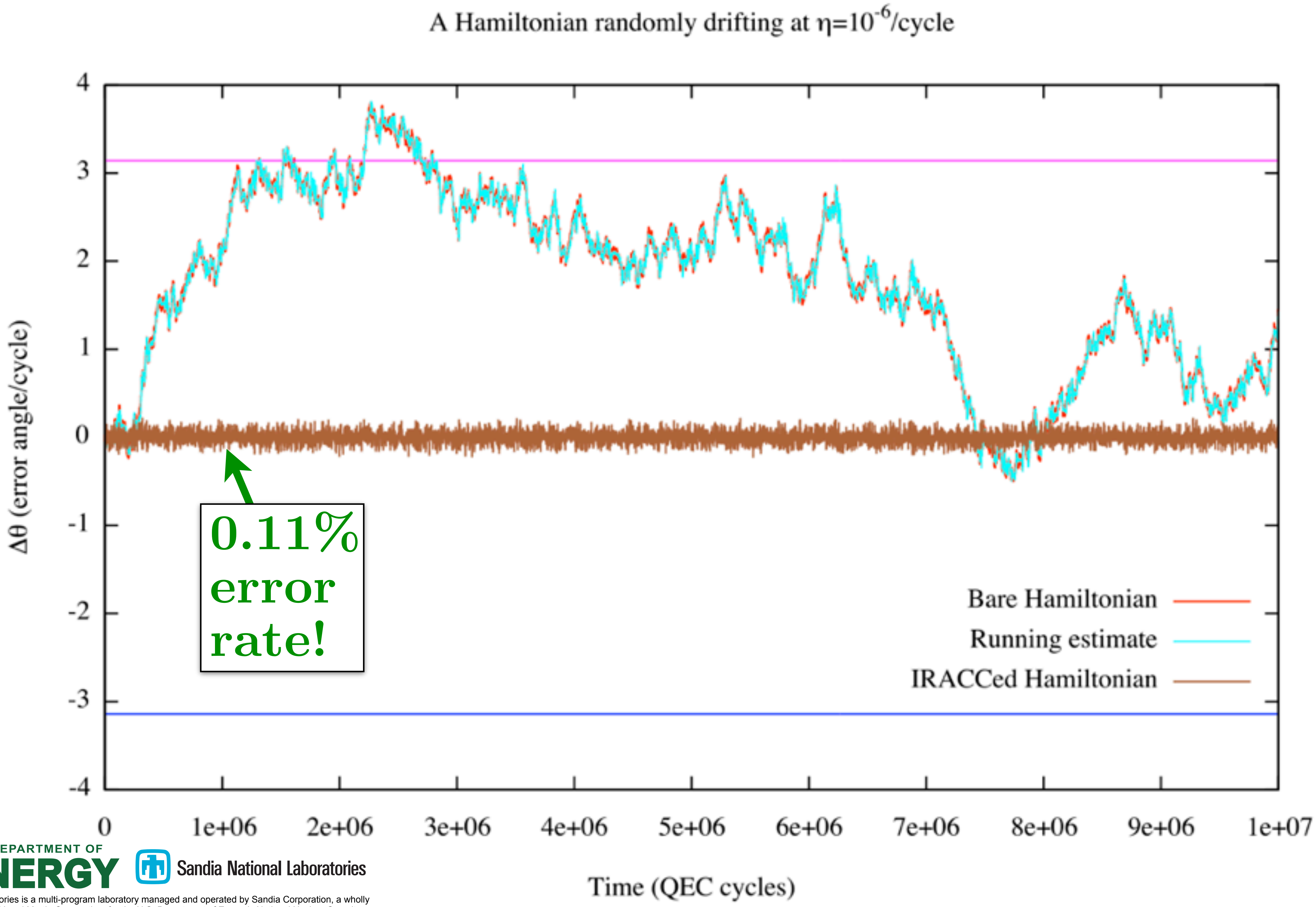
100% error rate!



U.S. DEPARTMENT OF ENERGY

Sandia National Laboratories

What's left after compensation



The Algorithm

1. Count cycles (T) until 1 error occurs.

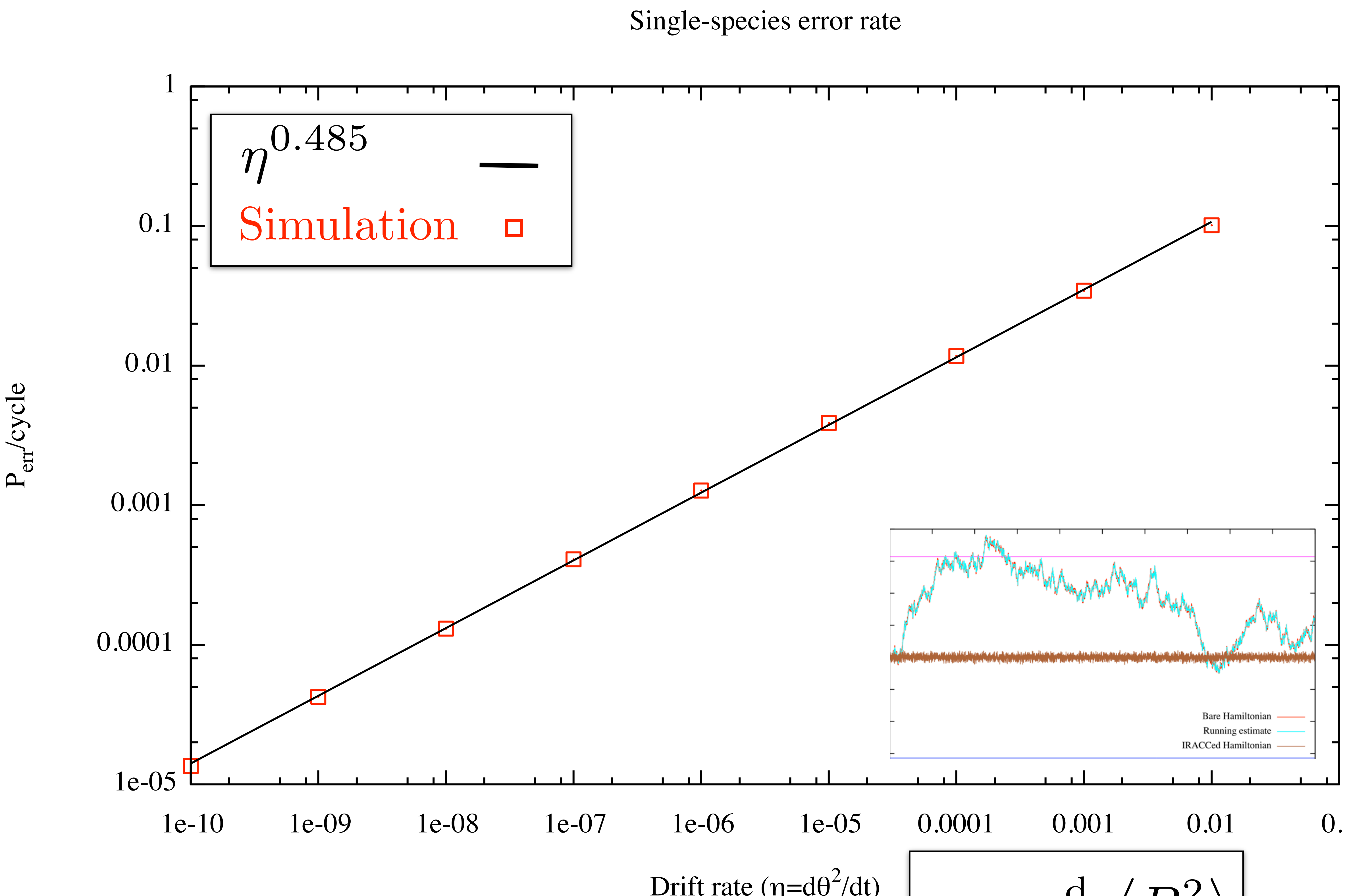
2. Guess $p_{\mathrm{err}} = \theta^2 = P/T \Longrightarrow \theta = \sqrt{P/T}$.

3. Compensate by adding $\Delta H = \pm\theta/\Delta T$.

4. Next time, choose the opposite sign. GOTO 1.

Want to see how well this works?

Stabilized error rate vs η



$$\eta \equiv \frac{\mathrm{d}}{\mathrm{d}t} \left\langle B^2 \right\rangle$$

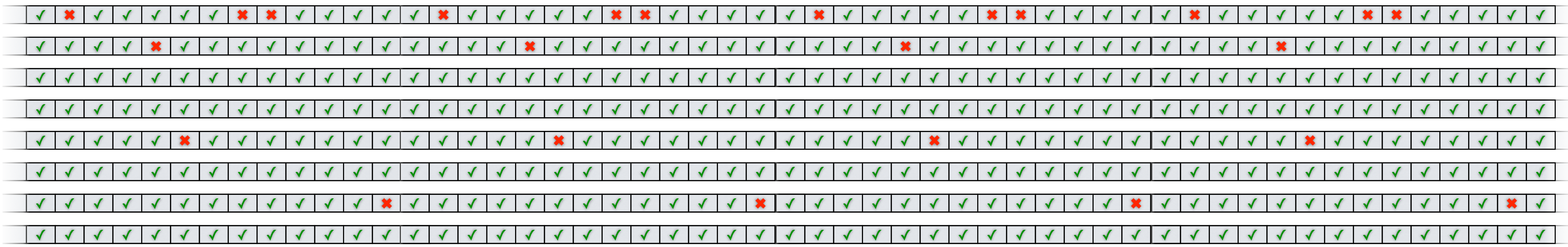
Okay. Monitoring syndrome measurements can stabilize drift. **But does it help QEC?**

$X \otimes Z \otimes Z \otimes X \otimes I,$
$I \otimes X \otimes Z \otimes Z \otimes X,$
$X \otimes I \otimes X \otimes Z \otimes Z,$
$Z \otimes X \otimes I \otimes X \otimes Z.$

Let's apply this to a real code. Like [5,1,3].

1. Now we measure 4 Pauli stabilizers, & get one of 16 outcomes.
2. "0" means no error. {1..15} indicate X/Y/Z errors on qubits 1-5.
3. So now we have to monitor 15 syndrome streams in parallel...

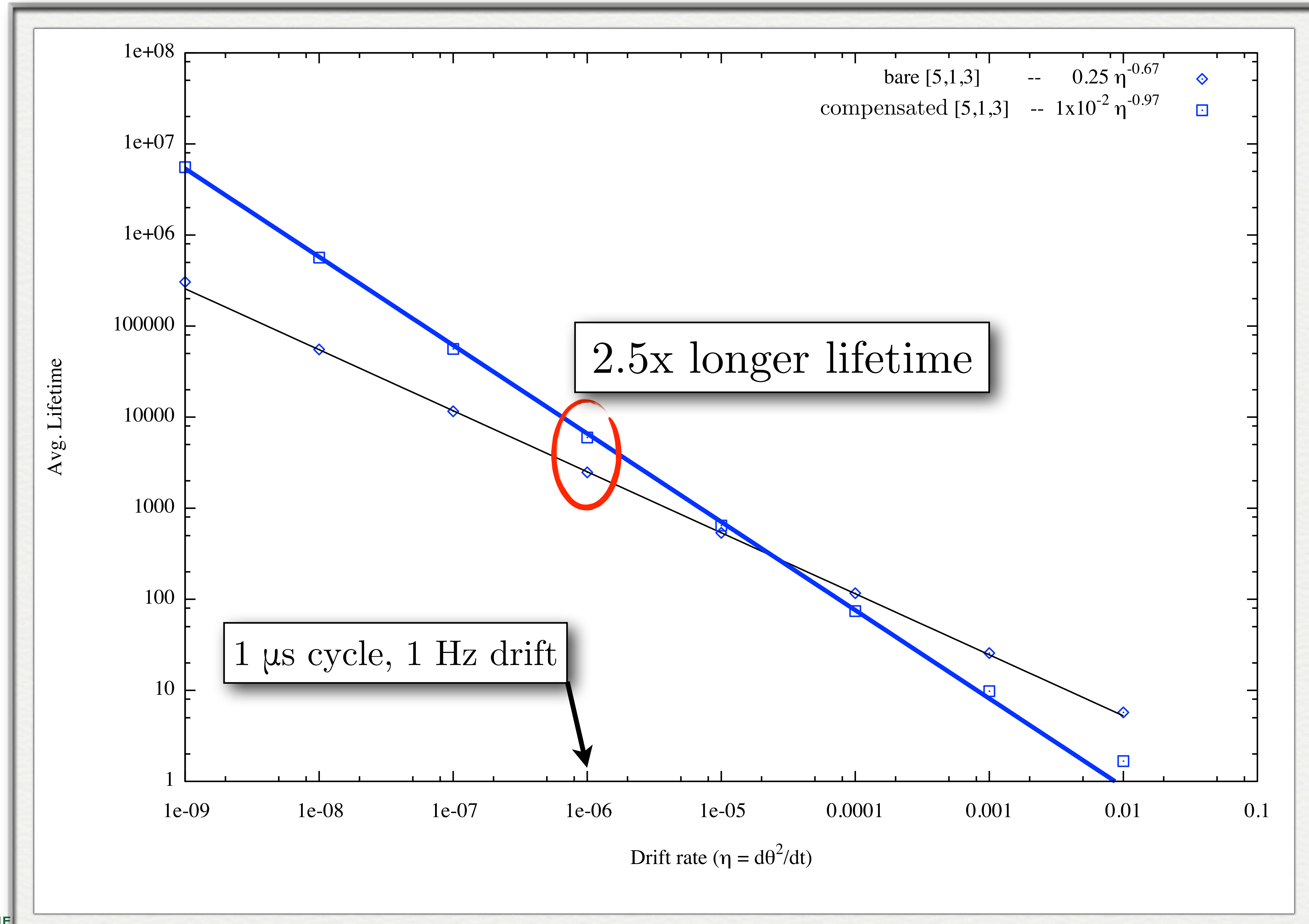
   ... and adjust 15 terms in the Hamiltonian.



Two errors in the same cycle => **DEATH**.

Natural metric: ***expected logical qubit lifetime.***

Drift control improves logical qubit lifetime *if* drift rate is low enough

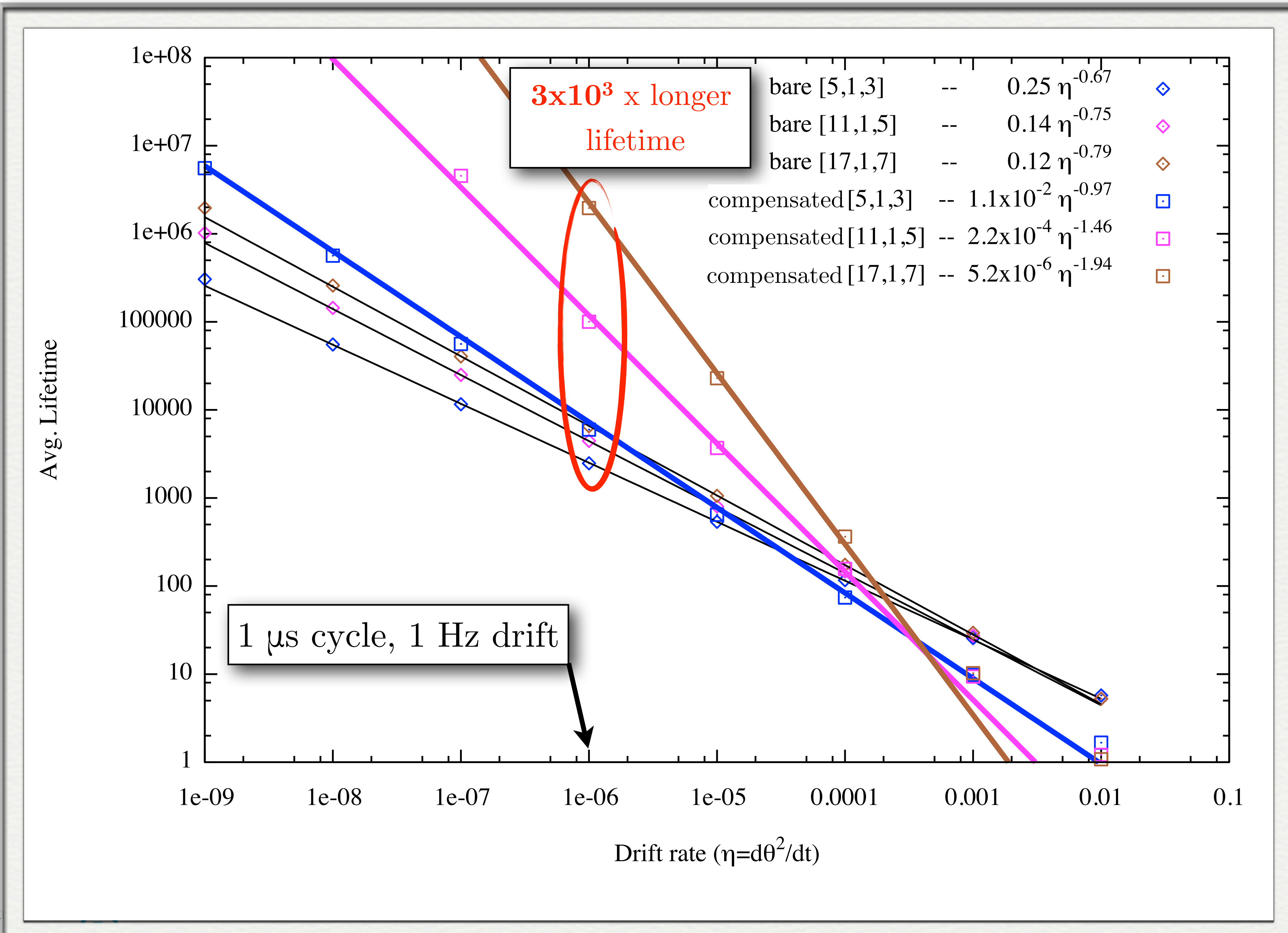


How about a bigger code?

The [5,1,3] code can tolerate 1 error.

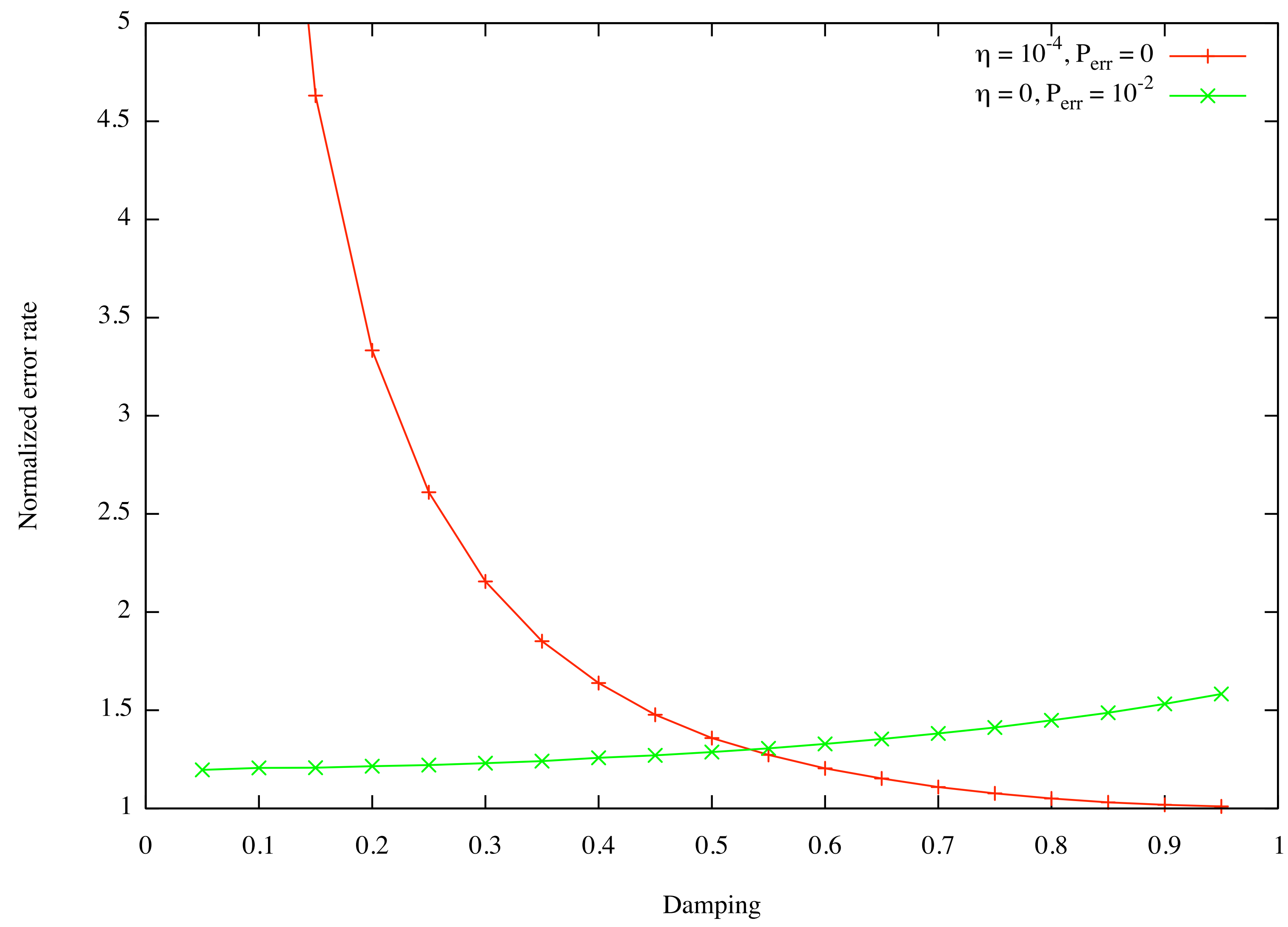
A [11,1,5] code can tolerate 2 errors.
A [17,1,7] code can tolerate 3 errors.

Lifetimes for [11,1,5] and [17,1,7] codes



What if there's <u>incoherent</u> noise?

- Stochastic noise causes errors even when $\boldsymbol{H}=0$.
  ==> We can't tell <u>these</u> errors from ones caused by $\boldsymbol{H}$!
- **Easy fix:**
  "Damp" each adjustment by a factor ≈0.55.
- Stabilized error rate is within 30% of optimal.



FAQ

Q: Wouldn't a more sophisticated algorithm do better?

A: Our protocol gets $p_{\mathrm{err}} = \eta^{0.485}$. I can prove a lower bound of $\eta^{0.5}$.

Q: Is it fault tolerant?

A: Yes, if you ignore two consecutive errors (=failed measurement).

Q: Does it work for $1/f$ noise?

A: Sure. But $1/f$ noise is a lot whiter than Brownian drift, so there's a lot less that *anything* can do about it.

Q: What about degenerate codes?

A: Good question! We already deal with +/- degeneracy in $\boldsymbol{H}$, but truly degenerate codes are harder. It *does* work (cycle through possible errors), but this needs more research.