Optimising Quantisation Noise in Energy Measurement

William B. Langdon, Justyna Petke, and Bobby R. Bruce

CREST, Department of Computer Science,
University College London Gower Street, London WC1E 6BT, UK

Abstract. We give a model of parallel distributed genetic improvement. With modern low cost power monitors; high speed Ethernet LAN latency and network jitter have little effect. The model calculates a minimum usable mutation effect based on the analogue to digital converter (ADC)'s resolution and shows the optimal test duration is inversely proportional to smallest impact we wish to detect. Using the example of a 1KHz 12 bit 0.4095 Amp ADC optimising software energy consumption we find: it will be difficult to detect mutations which an average effect less than 58 microA, and typically experiments should last well under a second.

Keywords: theory, genetic improvement, genetic programming, software engineering, SBSE, parallel EC, distributed power monitoring

1 Introduction

Evolutionary computing (EC) can be incorporated into product development either by inventing new designs or optimising existing ones. In both it is fundamentally important to be able to decide if a design is fit or not. The widespread adoption of fully functional mobile computers in the form of smartphones has thrust optimising software energy usage, and so battery life, into the limelight.

In many cases the quality of designs is calculated using simulators before manufacture. However, it is necessary that the simulation be detailed enough so that it can tell automatically a better design from an already good design. In the case of simple electronics such high quality simulator may exist. However

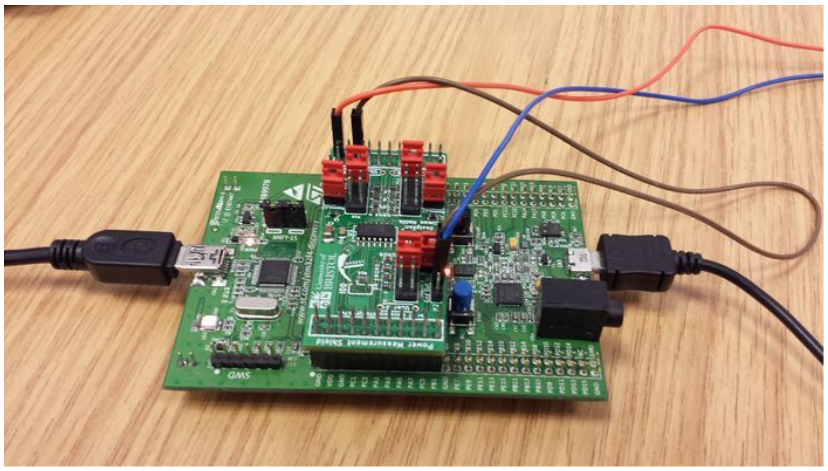


Fig. 1. MAGEEC Power Measurement Board ⓒ http://mageec.org/

even in the case of single chip devices, such simulators run several orders of magnitude slower that the software running on the chip and good simulators for the whole of a portable device may not be feasible. So for feasibility, cost, credibility and speed there is increasing interest in optimising portable electronic devices by using real devices and real power monitors (Figure 1) to measure their true energy consumption and use it as part of the EC fitness function [Bobby R. Bruce, 2015]. With the advent of genetic improvement (GI) [W.B. Langdon, 2015] it is increasingly common to view software as mutable and apply EC directly to it [David R. White *et al.*, 2008; Bobby R. Bruce, 2015; Eric Schulte *et al.*, 2014a]. There is great interest in using real measurements. Although our immediate use case is genetic improvement and the evolution of better software, here we are concerned with the practical limits of using real-world measuring devices in EC.

The next section presents a mathematical model of the accuracy of a single measuring device directly connected to single test device. Since fitness testing is usually the bottleneck in EC, it is common to consider running fitness tests in parallel. Section 3 expands the model of discretised measurement to a high speed Ethernet local area network based distributed system of dozens of computer hardware under test. Since Ethernet is a stochastic protocol, network delays are necessarily variable. Section 4 calculates that the best tests will be surprisingly short, under one second. This is in keeping with our view that often too much care is taken to get an accurate fitness value, where it is only necessary to be able to tell a good mutant from a less good one. Section 5 discusses the results in Section 4, ways to avoid EC degenerating into random search, three alternatives to LAN messages and concludes. To save space some of the intermediate mathematical steps and some of the discussion have be omitted. (The full text can be found in our technical report of the same name RN/16/01.)

2 Directly Connected Monitor

Figure 2 shows a system to automatically measure physical components of an EC fitness function. The "physical system" will be subject to mutations taken from the current population and the system will attempt to quantify the mutation's effect. Our model applies generally to EC using physical measurement. It could deal with not just the power consumed by the CPU but also by other activities particularly the screen [Ding Li *et al.*, 2015], radio links and GPS.

In the case of genetic improvement, the mutation is applied to the software running on the physical devices (e.g. a smartphone) and the ADC (analog to digital converter) will measure its power consumption. Since phones operate at about 5 volts little signal conditioning other than a fixed resistor is needed.

The simple model we present is potentially suitable for the very high frequency response that modern oscilloscopes are capable of. Since such oscilloscopes cost many thousands of pounds we will concentrate on automated power monitors costing a few tens of pounds each (such as the one in Figure 1). Notice that although they cannot measure very high frequency (short duration) effects, they can still accurately measure average power consumption. Even if there is

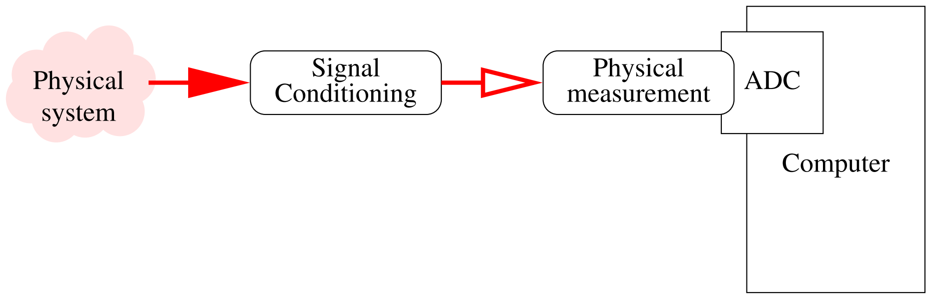


Fig. 2. Typical modern measuring and monitoring systems interface to the real world (physical system) via an analogue signal conditioning unit, a measuring device (e.g. a thermocouple) and an Analogue to Digital Converter (ADC). Although we consider optimising energy consumption, our mathematical framework can be generally applied. The conditioned signal is converted into an analogue electrical signal, which converted into a digital signal by the ADC, which is then read periodically at a fixed rate by the computer.

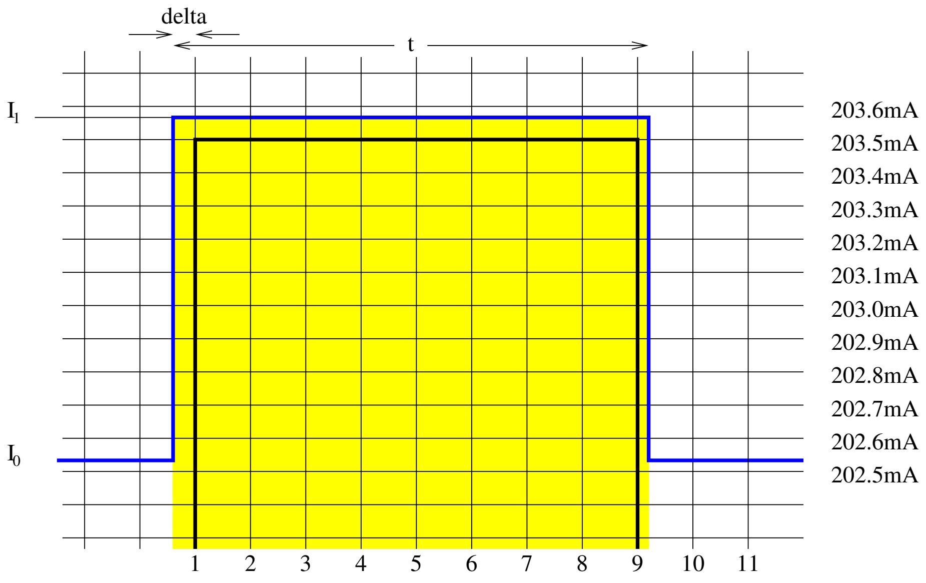


Fig. 3. Energy used is given by area of yellow rectangle times supply voltage (5 volts) $E = 5I_1t = 5 \times 203.6\text{mA} \times 8.6\text{mS} = 8.753\text{mJ}$. Current resolution $a = 0.1\text{mA}$ (12 bit ADC full scale 0.4095Amp). Sampling frequency $f = 1\text{KHz}$. Quantised energy $= 5 \times 203.5\text{mA} \times 8\text{mS} = 8.14\text{mJ}$. Noise $= 8.753 - 8.14 = 0.6134$. Relative noise $= 0.6134/8.753 \approx 7\%$.

significant amounts of power at high frequency, it does not disappear when measured at lower frequencies and (assuming there are no serious aliasing effects) it simply contributes to the low frequency average.

The simple model presented in Figure 3 assumes running the test causes the power consumption to rise but that the energy monitoring is quantised both into discrete time samples and that measurements of power consumption are also discrete. It assumes the power monitor is not synchronised to the start of the test software but that the start and end of the test are known. The actual energy used by the test is proportional to the area of the yellow rectangle in Figure 3 but the reported (discretised) energy is proportional to the number of unit rectangles inside the rectangle bounded by the thick black lines and the x axis. Next we will mathematically model the difference between the two.

– Supply voltage (assumed known and constant) V Volts.
– Sampling frequency = f, e.g. 1000 Hz.
– Current resolution = a, e.g. 0.1mA, thus a 12 bit Analogue to Digital Converter (ADC) will have a maximum reading of 0.4095 Amperes.
– Unloaded current draw I_0 Amps.
– Actual load I_1 Amps.
– The actual energy used is VI_1t Joules.
– δ is the time in seconds between the load being applied and first the sample.
– Assuming x is positive, the integer part of x is $\lfloor x \rfloor = x - \mathrm{frac}(x)$

The measured energy is $\frac{Va}{f} \left\lfloor \frac{I_1}{a} \right\rfloor \lfloor (t-\delta)f \rfloor$ so the discretization noise is

$$\begin{aligned}
&= VI_1t - \frac{Va}{f}\left\lfloor \frac{I_1}{a} \right\rfloor \lfloor (t-\delta)f \rfloor \\
&= VI_1t - \frac{Va}{f}\left(\frac{I_1}{a} - \mathrm{frac}\!\left(\frac{I_1}{a}\right)\right)\big((t-\delta)f - \mathrm{frac}((t-\delta)f)\big) \\
&= Vat\ \mathrm{frac}\!\left(\frac{I_1}{a}\right) + VI_1\delta - Va\delta\ \mathrm{frac}\!\left(\frac{I_1}{a}\right) \\
&\quad + \frac{V}{f}I_1\mathrm{frac}((t-\delta)f) - \frac{Va}{f}\mathrm{frac}\!\left(\frac{I_1}{a}\right)\mathrm{frac}((t-\delta)f) \qquad (1)
\end{aligned}$$

Since the start of running the software is unrelated to the exact point in time measurements are taken, δ will be uniformly scattered in the range [0 to $1/f$] and so the expected value of δ is $1/2f$ (Figure 3). Since I_1 is much bigger than a, it is reasonable to assume the fractional part of I_1/a, i.e. $\mathrm{frac}(I_1/a)$, is uniformly distributed across the interval [0-1]. (With a uniform distribution in [0-1], the expected value of $\mathrm{frac}(\cdot)$ is 1/2 and the standard deviation is $\sqrt{1/12} = 0.288675$). So the expected noise (Eq. 1) becomes:

$$\frac{Vat}{2} + VI_1\frac{1}{2f} - Va\ \frac{1}{2} \times \frac{1}{2f} + \frac{V}{f}I_1\frac{1}{2} - \frac{Va}{f}\frac{1}{2} \times \frac{1}{2} = \frac{1}{2}Vat + V\frac{I_1}{f} - \frac{1}{2}V\frac{a}{f}$$

$$\text{Fractional noise} = \frac{\text{noise}}{\text{true energy}} = \frac{\frac{1}{2}Vat + V\frac{I_1}{f} - \frac{1}{2}V\frac{a}{f}}{VI_1t} = \frac{1}{2}\frac{a}{I_1} + \frac{1}{ft} - \frac{1}{2}\frac{a}{fI_1t} \qquad (2)$$

We can approximate the fractional noise by dropping the last term in Equation 2. We can express Eq. 2 in terms of the current measurement resolution and number of samples $N = ft$. Each ADC raw value is $I_1/a = k$. (For a twelve bit resolution analogue to digital converter and I_1 near the middle of the range $k \approx 2048$.) So Eq. 2 becomes fractional noise $\approx 1/4096 + 1/N$. That is, with a coarse sampling the noise is dominated by the number of samples N but if we can either increase the sampling rate or run the experiment for longer, the $1/N$ term becomes less important and the noise tends to a limit given by the resolution of the ADC. Further, once the number of samples, N, exceeds the resolution of the ADC there is only marginal reduction in noise from increasing the number of samples. Using our 12 bit 1KHz example ADC, there is only marginal gain in increasing the number of measurements above 4096. That is, greatly increasing the measurement time, t, above $4096/f \approx 4$ seconds, gives little further improvement. See also end of Section 4.

3 Distributed Power Measurement

In the previous section we assume that the onset of the load and when its finished are known exactly. In the case of distributed power monitoring, two commands are sent via a local area network (LAN). The first is to start the recording of energy consumption and the second to stop the recording. Initially we shall concentrate upon the variation introduced by the LAN and then include the energy measurement noise given by Equation 1.

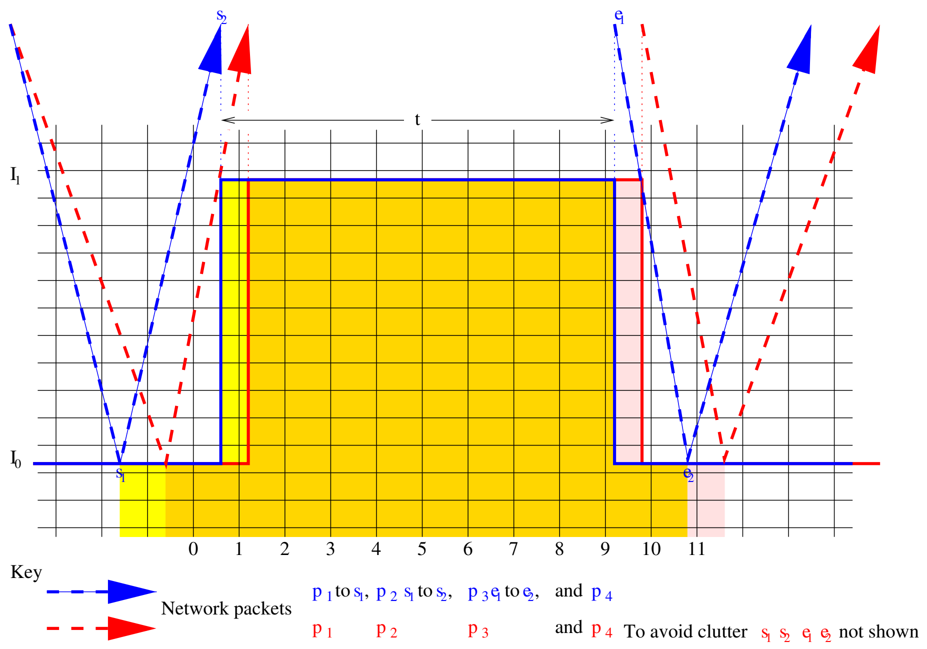
Measuring energy is initiated when the start message packet (p_1) reaches the monitoring computer at time s_1. (The LAN packets are shown by dotted arrows in Figure 4.) When the acknowledgement packet (p_2) reaches the test computer (s_2), it starts the experiment, raising the current from rest (I_0) to I_1. t seconds later (e_1) the experiment finishes: the load drops back to I_0 and the test computer sends a message packet (p_3) stopping the measurement (e_2). In Figure 4 the experiment is done twice but different results are obtained since although the test computer starts at the same time and the experiment takes t seconds in both cases, the network delays are different.

The measured energy is $V\left(I_0(s_2 - s_1) + (I_1 - I_0)t\right)$. Where $(s_2 - s_1)$ is the observed duration. This is longer than t because of the transit times of the two network packets p_2 and p_3. (Figure 5 gives transit times for two LAN packets, there and back.) Now $(s_2 - s_1) = p_2 + t + p_3$ so measured energy $=$ $V\left(I_0(p_2 + t + p_3) + (I_1 - I_0)t\right) = V\left(I_0(p_2 + p_3) + I_1 t\right)$

We will assume that the transit times for the LAN packets are on average the same and that variations are independent. Thus the variance in the energy measurement due to network work variations (i.e. V, I_1 and t are assumed fixed):

$$V^2 I_0^2 \left(\mathrm{var}\,(p_2) + \mathrm{var}\,(p_3)\right) = 2V^2 I_0^2\ \mathrm{var}\,(p) \tag{3}$$

Since we assume that p_2 are p_3 are equally distributed and independent we drop their subscripts are refer to them both as p. So $\mathrm{var}\,(p)$ is the variance of LAN packet transit times ($\mathrm{SD}\,(p) = \sqrt{\mathrm{var}\,(p)}$). The fractional variation in the energy



s2
e1
t
I1
I0
s1
e2
0 1 2 3 4 5 6 7 8 9 10 11
Key
Network packets
p 1 to s1, p 2 s1 to s2, p 3 e1 to e2, and p 4
p 1 p 2 p 3 and p 4
To avoid clutter s1 s2 e1 e2 not shown

Fig. 4. Measuring energy is initiated when the start message (left arrow) reaches the monitoring computer s_1. When the acknowledgement reaches the test computer s_2, it starts the experiment, raising the current from rest (I_0) to I_1. t seconds later the experiment finishes: the load drops back to I_0 and the test computer sends a message e_1 ending the measurement e_2. The experiment is done twice (blue and red) but different results are obtained since the network delays are different. As in Figure 3, energy used is given by area of under current curves times supply voltage (5 volts). Left (blue) = 12.60mJ, right (red) = 12.40mJ, relative difference = 0.2/12.60 ≈ 1.6%.

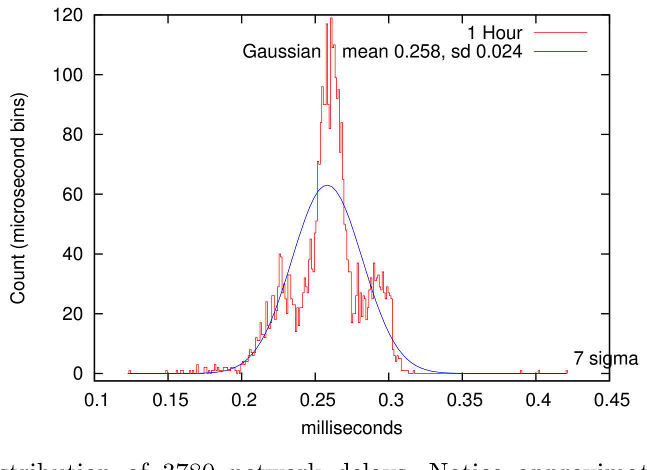


Fig. 5. Distribution of 3780 network delays. Notice approximate match of Normal distribution and also long tail of much longer delays.

measurement is

$$= \frac{\sqrt{2}\ V I_0 \mathrm{SD}\,(p)}{V\,(2I_0 p + I_1 t)} = \frac{\sqrt{2}\ \mathrm{SD}\,(p)}{(2p + tI_1/I_0)}$$

Figure 5 suggests the mean of the two packet transit time ($2p$) is typically 0.258mS and $\sqrt{2}\ SD(p)$ is 24 microseconds.

The variation in the discretization noise (given by Equation 1) is due to variation in the duration t and size I_1 of the load. Treating these as independent gives the variance in the discretization noise. (Remember the variance of the product of two independent variables x and y (of means X and Y) is $\mathrm{var}\,(xy) = X^2\mathrm{var}\,(y) + Y^2\mathrm{var}\,(x) + \mathrm{var}\,(x)\mathrm{var}\,(y)$ [Leo A. Goodman, 1960, Eq. 2].) Remember (Eq. 1) discretization noise/V

$$= at\,\mathrm{frac}\!\left(\frac{I_1}{a}\right) + I_1\!\left(\delta + \frac{1}{f}\mathrm{frac}\!\left((t-\delta)f\right)\right) - a\delta\ \mathrm{frac}\!\left(\frac{I_1}{a}\right) - \frac{a}{f}\mathrm{frac}\!\left(\frac{I_1}{a}\right)\mathrm{frac}\!\left((t-\delta)f\right)$$

We now calculate the variance of discretization noise/V one term at a time. Note the variance of the uniform distribution of the range [0-1] is 1/12. Starting with the first (depends on t) and last terms

$$\mathrm{var}\left(at\ \mathrm{frac}\!\left(\frac{I_1}{a}\right)\right) = a^2\mathrm{var}\,(t)/3 + a^2t^2/12 \qquad (4)$$

$$\mathrm{var}\left(-\frac{a}{f}\mathrm{frac}\!\left(\frac{I_1}{a}\right)\mathrm{frac}\!\left((t-\delta)f\right)\right) = 7/144\ \frac{a^2}{f^2} \qquad (5)$$

Now the middle terms (which depend on both I_1 and δ).

$$I_1\left(\delta + \frac{1}{f}\mathrm{frac}\!\left((t-\delta)f\right)\right) - a\delta\ \mathrm{frac}\!\left(\frac{I_1}{a}\right) = \delta\left(I_1 - a\ \mathrm{frac}\!\left(\frac{I_1}{a}\right)\right) + \frac{I_1}{f}\mathrm{frac}\!\left((t-\delta)f\right)$$

Taking the variance of the first part (assuming that δ and I_1 are independent)

$$\mathrm{var}\left(\delta\left(I_1 - a\ \mathrm{frac}\!\left(\frac{I_1}{a}\right)\right)\right) = \mathrm{var}\,(\delta)\,(I_1 - a/2)^2 + \delta^2\left(\mathrm{var}\,(I_1) + a^2/12\right) + \mathrm{var}\,(\delta)\left(\mathrm{var}\,(I_1) + a^2/12\right)$$

and of the second part $\quad \mathrm{var}\left(\frac{I_1}{f}\mathrm{frac}\!\left((t-\delta)f\right)\right) = \frac{\mathrm{var}\,(I_1)}{f^2}/3 + \frac{I_1^2}{f^2}/12 \qquad (6)$

Combining formulae 4–6 gives var (discretization noise/V) as:

$$= a^2\mathrm{var}\,(t)/3 + a^2t^2/12 + \mathrm{var}\,(\delta)\,(I_1 - a/2)^2 + \delta^2\left(\mathrm{var}\,(I_1) + a^2/12\right) + \mathrm{var}\,(\delta)\left(\mathrm{var}\,(I_1) + a^2/12\right) + \frac{\mathrm{var}\,(I_1)}{f^2}/3 + \frac{I_1^2}{f^2}/12 + 7/144\ \frac{a^2}{f^2}$$

$$= \frac{a^2\mathrm{var}\,(t)}{3} + \mathrm{var}\,(\delta)\left(\left(I_1 - \frac{a}{2}\right)^2 + \frac{a^2}{12}\right) + \delta^2\mathrm{var}\,(I_1) + \mathrm{var}\,(\delta)\mathrm{var}\,(I_1) + \frac{\mathrm{var}\,(I_1)}{3f^2} + \frac{a^2t^2}{12} + \frac{\delta^2a^2}{12} + \frac{I_1^2}{12f^2} + \frac{7a^2}{144f^2}$$

Referring to the end of Section 2 we have $t = N/f$ and $I_1 = ka$. Since the load and measurement computers are not synchronised $\delta = 1/2f$ and $\mathrm{var}\,(\delta) = f^2/12$ (Figure 3). So var (discretization noise$/V$) becomes

$$
\begin{aligned}
&= \frac{a^2\mathrm{var}\,(t)}{3} + \frac{a^2}{12f^2}\left(\left(k-\frac{1}{2}\right)^2 + \frac{1}{12}\right) + \frac{a^2}{4f^2}\mathrm{var}\,(k) \\
&\quad + \frac{a^2}{12f^2}\mathrm{var}\,(k) + a^2\frac{\mathrm{var}\,(k)}{3f^2} + \frac{a^2N^2}{12f^2} + \frac{a^2}{48f^2} + \frac{k^2a^2}{12f^2} + \frac{7a^2}{144f^2} \\
&= \frac{a^2}{3}\mathrm{var}\,(t) + \frac{2a^2}{3f^2}\mathrm{var}\,(k) + \frac{a^2}{144f^2}\left(12N^2 + 24k^2 - 12k + 14\right)
\end{aligned}
$$

Assuming a 12 bit ADC and I_1 approx. half full scale var (discretization noise)

$$
\begin{aligned}
&= \frac{V^2a^2}{3}\mathrm{var}\,(t) + \frac{2V^2}{3f^2}\mathrm{var}\,(I_1) + \frac{V^2a^2t^2}{12} + \frac{V^2a^2}{144f^2}\left(24k^2 - 12k + 14\right) \qquad (7) \\
&\approx \frac{V^2a^2}{3}\mathrm{var}\,(t) + \frac{2V^2}{3f^2}\mathrm{var}\,(I_1) + \frac{V^2a^2t^2}{12} + \frac{698880\ V^2a^2}{f^2}
\end{aligned}
$$

We will assume t is long compared to both the sampling frequency f and the network variation. This allows us to assume that the variance in the energy reported is given by the sum of the variance due to network variation (Equation 3) and that due noise in the measuring system (Equation 7).

$$
= 2V^2I_0^2\mathrm{var}\,(p) + \frac{V^2a^2}{3}\mathrm{var}\,(t) + \frac{2V^2}{3f^2}\mathrm{var}\,(I_1) + \frac{V^2a^2t^2}{12} + \frac{V^2a^2}{144f^2}\left(24k^2 - 12k + 14\right)
$$

Assuming both t and I_1 are fixed so variance of energy measurement is

$$
= 2V^2I_0^2\mathrm{var}\,(p) + \frac{V^2a^2t^2}{12} + \frac{V^2a^2}{144f^2}\left(24k^2 - 12k + 14\right) \qquad (8)
$$

4 Maximising Beneficial Mutation Detection Rate

Suppose we run the original version of the software to be improved and record its use of energy. We then mutate the software. Suppose the mutation is beneficial, in that it reduces the energy consumed by Δ. (Here we assume the power consumption is spread uniformly across the time the software runs. Notice we are assuming the mutation changes the power consumption but the runtime t is not changed. See Section 5). If Δ^2 is large compared to the measurement variance (Equation 8) then we can reasonably expect to measure that the mutation has been beneficial. If the difference is small, we may want to repeat the measurement to increase Δ. However, this would proportionately reduce the rate that we can test mutations. Equation 8 means we can ask: Is

$$
\Delta^2 \text{ much bigger than } 2V^2I_0^2\mathrm{var}\,(p) + \frac{V^2a^2t^2}{12} + \frac{V^2a^2}{144f^2}\left(24k^2 - 12k + 14\right)\ ? \qquad (9)
$$

Let $\mathit{\Delta} I = \mathit{\Delta}/Vt$ be the beneficial effect of the mutation expressed in terms of energy divided by the length of the testing period. Notice that increasing the mutation testing time also increases the variance in the energy measurement. We divide by the supply voltage V so that $\mathit{\Delta} I$ can be expressed as the average reduction in current. Using $\mathit{\Delta}^2 = (\mathit{\Delta} I)^2 V^2 t^2$ in Question 9 and then dividing through by V^2 means Question 9 is the same comparison as: Is (the signal)

$$t^2(\mathit{\Delta} I)^2 \text{ much bigger than } \frac{a^2t^2}{12} + 2I_0^2\mathrm{var}\,(p) + \frac{a^2}{144f^2}\left(24k^2 - 12k + 14\right)$$

Notice the last two terms do not depend on t and so for $\mathit{\Delta} I > a\sqrt{1/12}$ we can make the energy signal bigger than its variability by increasing t. However, we cannot effectively detect beneficial mutations with a proportionate effect less than $\mathit{\Delta} I = a\sqrt{1/12} \approx 0.3\ a$. If we require the signal to be at least twice the variability (4 times the variance) we can calculate the minimum time required.

$$t^2(\mathit{\Delta} I)^2 = \frac{a^2t^2}{3} + 8I_0^2\mathrm{var}\,(p) + \frac{a^2}{36f^2}\left(24k^2 - 12k + 14\right)$$

$$t = \sqrt{\frac{8I_0^2\mathrm{var}\,(p) + \frac{a^2}{36f^2}\left(24k^2 - 12k + 14\right)}{(\mathit{\Delta} I)^2 - a^2/3}}$$

Let $\mathit{\Delta} k = \mathit{\Delta} I/a$, assume $I_0 \approx I_1 = ka$

$$t \approx \sqrt{\frac{24k^2\mathrm{var}\,(p) + \frac{1}{12f^2}\left(24k^2 - 12k + 14\right)}{3(\mathit{\Delta} k)^2 - 1}} \tag{10}$$

Alternatively we can express this minimum time (Eq. 10) as a minimum number of number of samples using $N = ft$ (N was defined at the end of Section 2).

$$N \approx \sqrt{\frac{k^2\left(24f^2\mathrm{var}\,(p) + 2\right) - k + 14/12}{3(\mathit{\Delta} k)^2 - 1}}$$

Again assuming a 1KHz 12 bit ADC and noting that Figure 5 suggests $\sqrt{2}\,SD(p)$ is 24 microseconds. i.e. $\mathrm{var}\,(p) = 2.86\ 10^{-10}\mathrm{s}^2$. So $f^2\mathrm{var}\,(p) = 2.86\ 10^{-4}$.

$$N \approx k\sqrt{\frac{2}{3(\mathit{\Delta} k)^2 - 1}} \tag{11}$$

$\mathit{\Delta} k$ is the mutation's impact on energy consumption, assumed constant over time, expressed as a current in units of the analogue to digital converter's resolution. If the average impact of the mutation is large compared to the resolution of the ADC, then $\mathit{\Delta} k \gg 0.58$. Therefore for our 1KHz 12 bit ADC and mutations with a reasonably large impact the measurement need only last $1.7/\mathit{\Delta} k$ seconds.

5 Discussion and Conclusions

Experimental work suggests that the impact of software mutations is very non-uniform, with many mutations having no effect or being detrimental and only a small number being beneficial [W.B. Langdon and J. Petke, 2015; Schulte *et* *al.*, 2014b]. Hence setting the experimental parameters to allow rapid detection of large impact mutations risks not detecting many small impact mutations. Where large mutations are rare this risks the EC degenerating into random search. Indeed if the impact of mutations is too small to be reliably detected (i.e. $\Delta I < 0.58a$) then we cannot expect miracles from EC.

We have modelled the energy consumption of software mutations by assuming their impact is spread uniformly throughout each test run. This is unlikely to be true and more sophisticated models might look at how the impact of mutations is distributed. However, for a mutation to be detected its effect will still need to be large compared to the ADC sensitivity. This suggests our present lower bound ($\Delta I = 0.58a$) might be improved at the cost of assuming more about software mutants, however, it appears that a critical lower bound will still exist.

If the test program is run repeatedly in order to integrate the mutation's effect, we would expect repeated patterns in the power monitor's signal. There are very sensitive algorithms which can reliably measure periodic differences even in the presence of sizeable noise.

Alternatively, it might be possible to use signal processing to recognise the onset and termination of the measurement period. Or, several low end test beds (e.g. the Raspberry Pie) have output pins which could be used to start and stop energy measurement. Finally, both the computer under test and the computer running the energy monitors have sophisticated clocks, which can be synchronised and thus absolute time (rather than explicit message passing) might be used to keep track of the start and end of energy consumption experiments.

1. It will be difficult to detect mutations which have on average an effect less than $\sqrt{(1/3)}\ a$ (a is the ADC's resolution) on the current consumed. For our example 12 bit 0.4095Amp ADC this sets a lower limit of 58μA.
2. On the other hand if the effect is much bigger than 58μA, there is little to be gained by running measurement for longer than a second. Equation 11 suggests the ideal duration falls in proportion to the smallest effect size we wish our evolutionary system to detect.

References

Bobby R. Bruce, 2015. Energy optimisation via genetic improvement a SBSE technique for a new era in software development. *GECCO GI-2015 workshop*, pp819–820.

Leo A. Goodman, 1960. On the exact variance of products. *Journal of the American Statistical Association*, 55(292):708–713, Dec 1960.

W.B. Langdon and J. Petke, 2015. Software is not fragile. In *CS-DC'15*, Proceedings in Complexity, Springer, Forthcoming.

W.B. Langdon, 2015. Genetically improved software. In *Handbook of Genetic Programming Applications*, chapter 8, pages 181–220. Springer.

Ding Li *et al.*, 2015. Optimizing display energy consumption for hybrid Android apps. In *DeMobile 2015*, pages 35–36, Bergamo, Italy, 31 August. ACM. Invited Talk.

Eric Schulte *et al.*, 2014a. Post-compiler software optimization for reducing energy. In *ASPLOS'14*, pages 639–652, Salt Lake City, Utah, USA, 1-5 March. ACM.

Schulte *et al.*, 2014b. Software mutational robustness. *GP&EM*, 15(3):281–312, 2014.

David R. White *et al.*, 2008. Searching for resource-efficient programs: low-power pseudorandom number generators. In *GECCO '08* pages 1775–1782. ACM.