UCL DEPARTMENT OF
COMPUTER SCIENCE

Research Note

RN/16/01

Optimising Quantisation Noise in Energy Measurement

27 Jan 2016

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

Abstract

A simple model of distributed Genetic Improvement running in parallel across a local area network in which start/stop commands are sent to measuring devices calculates a minimum usable software mutation effect based on the analogue to digital convert (ADC)'s resolution. With modern low cost power monitors, the high speed Ethernet LAN's jitter and delays appear to have little effect. Where the software to be improved permits it, optimal test duration is inversely proportionate to minimum mutation effect size, typically well under a second.

Keywords

theory, genetic improvement, genetic algorithms, genetic programming, software engineering, SBSE, parallel evolutionary computing, artificial intelligence, distributed power monitoring, local area network, Ethernet, jitter

1 Introduction

The widespread adoption of fully functional mobile computers in the form of smartphones, has thrust software energy usage and its effect on battery life into the limelight. Evolutionary computing [Goldberg, 1989] can be incorporated into product development. Usually this is done by representing many candidate product designs as "chromosomes" within the EC's population. There are artificial rules for making changes to designs (mutations) and for combining designs (crossover). New designs are created from the better members of the population and the worse are discarded and the Darwinian [Darwin, 1859] process of selection and breeding from the fitter continues through a sequence of generations. EC has been highly successful at inventing new designs [Koza *et al.*, 2003]. Of fundamental importance is deciding if a design is fit or not.

In many cases the quality of designs is calculated directly from each mutated design by using simulators before the design is manufactured. In the case of EC, it is primarily 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 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 power monitors to measure their true energy consumption and use it as part of the EC fitness function [Bruce, 2015]. For this to be viable, it must be possible to speedily mutate the design, load it into the device and test it. Previously this has meant using reconfigurable hardware (i.e. "Evolvable hardware") often based on Xilinx FPGAs [Thompson, 1996]. However with the advent of Genetic Improvement (GI) [Langdon, 2015] it is increasingly common to view software as mutable and apply EC directly to it. In the case of reducing energy consumption this has been via simulation [White *et al.*, 2008; Bruce, 2015] or using linear models of energy consumption [Schulte *et al.*, 2014]. However there is great interest in using real measurements. Although our immediate use case is GI 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 [Stender, 1993]. 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 and even this can be subject to a fair degree of noise as long as the noise is unbiased [Langdon, 2011]. Section 5 discusses the results in Section 4 and ways to avoid EC degenerating into random search. Section 6 considers three alternatives to using LAN messages to synchronise the software being tested with the distributed measuring device. Section 7 concludes.

2 Directly Connected Monitor

Figure 1 shows a schematic of a general computerised measuring system that might be used to automatically measure physical components of an EC fitness function. In EC the "Physical system" will be subject to mutations taken from the current population and the system in Figure 1 will attempt to quantify the mutation's effect. Our model applies generally to EC using physical measurement.

In the case of Genetic Improvement, the mutation is applied to the software running on the physical devices (e.g. a smart phone) and the ADC will measure its power consumption. Since phones operate at about 5 volts little signal conditioning other than a fixed resistor is needed to convert the analogue signal (the phone's power draw) into a voltage suitable for the ADC.

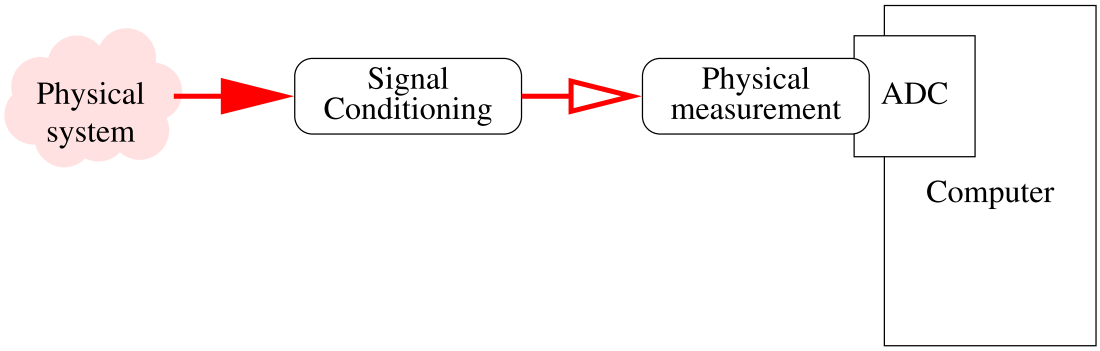

Figure 1: Typical modern measuring and monitoring systems interface to the real world (Physical system) via an analogue signal conditioning unit, a measuring device and an Analogue to Digital Converter (ADC). Signal conditioning might amplify the analogue input and/or filter the signal. E.g. in a central heating thermostat, it might insulate a thermistor from the air in the room to prevent the heater responding immediately to short term fluctuations, e.g. caused by someone opening a door to exit the building and then closing it behind themselves. The conditioned signal is converted into an analogue electrical system by a physical measuring device (e.g. a thermocouple) which is then converted into a digital signal by the ADC which is then read periodically at a fixed rate by the computer.

The simple model we present is potentially suitable for the very high frequency response that modern oscilloscopes are capable of. It is possible that oscilloscopes operating above 100 MHz might measure the power consumption of individual software method calls. Whilst such fine temporal resolution seems possible, given the complexity of smart-phones one would need to be very careful to ensure power consumed while the method was executing is directly and only due to the method itself and not due to incidental effects or the cross coupling between different software running on the phone e.g. via shared caches. Similarly the power consumed by the CPU might be confounded by other activities particularly the screen, radio links and GPS. Since such oscilloscopes cost many thousands of pounds we will concentrate on automated power monitors costing a few tens of pounds each. Notice that although they cannot measure very high frequency (short duration) effects, they can still accurately measure average power consumption. Even if there is 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 2 assumes running the test causes the power consumption to rise but that the energy monitoring is quantised both into discrete time samples and 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 2 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.

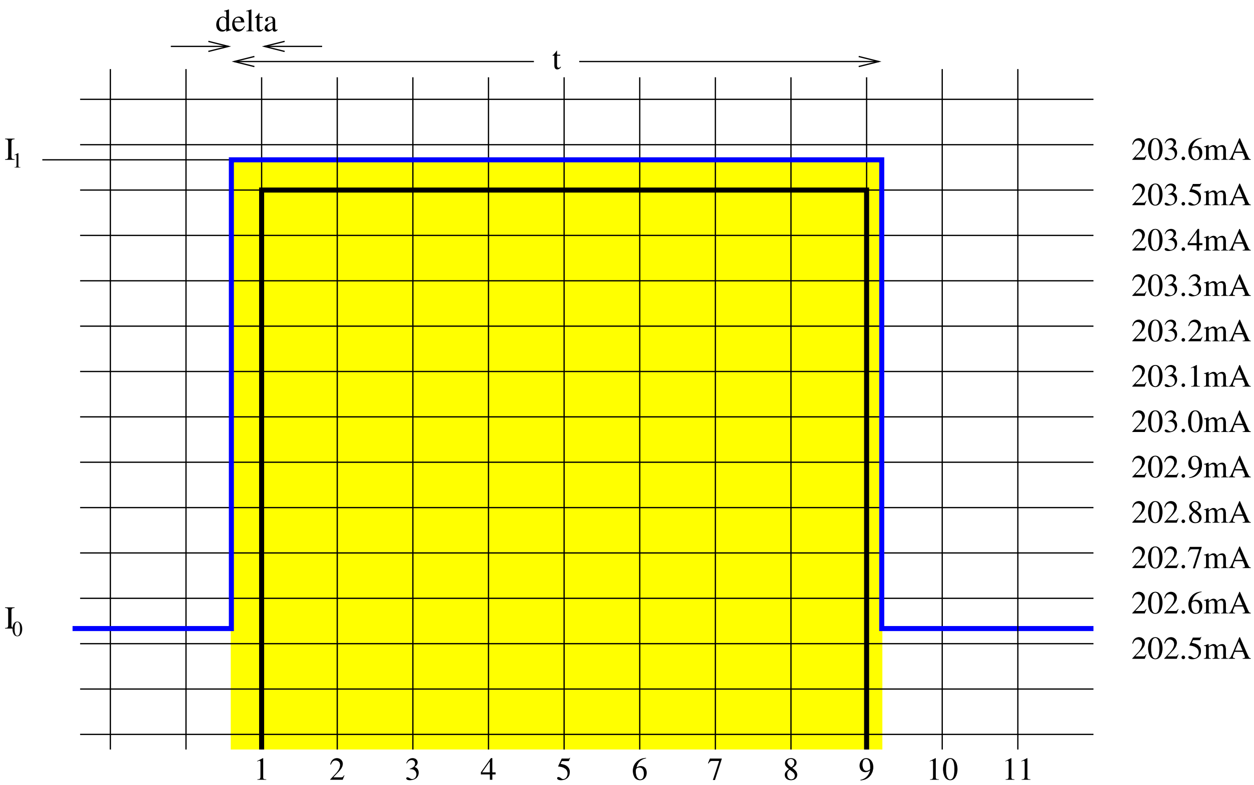

Figure 2: Energy used is given by area of yellow rectangle times supply voltage (5 volts) $E = 5I_1t = 5\ 203.567\mathrm{mA}\ 8.6\mathrm{mS} = 8.75337\mathrm{mJ}$. Current resolution $a = 0.1\mathrm{mA}$ (12 bit ADC full scale 0.4095Amp). Sampling frequency $f = 1$KHz. Quantised energy $= 5\ 203.5\mathrm{mA}\ 8\mathrm{mS} = 8.14\mathrm{mJ}$. Noise $= 8.75337 - 8.14 = 0.613367$. Relative noise $= 0.613367/8.75337 \approx 7\%$.

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 time in seconds between the load being applied and first sample being taken.

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

Assuming x is positive, the integer part of x is $\lfloor x\rfloor = x - \mathrm{frac}\,(x)$

Discretization noise

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

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 [0-1/f] and so the expected value of δ is $0.5/f$ (Figure 2). 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 0.5 and the standard deviation is $\sqrt{1/12} = 0.288675$). So the expected noise (Eq. 1) becomes:

$$\begin{aligned}
&= 0.5Vat + VI_1 0.5/f - Va\,0.5\,0.5/f + \frac{V}{f}I_1 0.5 - \frac{Va}{f}0.5\,0.5 \\
&= 0.5Vat + 0.5V\frac{I_1}{f} - 0.25\frac{Va}{f} + 0.5V\frac{I_1}{f} - 0.25\frac{Va}{f} \\
&= 0.5Vat + V\frac{I_1}{f} - 0.5V\frac{a}{f}
\end{aligned}$$

Fractional noise

$$\begin{aligned}
\frac{\text{noise}}{\text{true energy}} &= \frac{0.5Vat + V\frac{I_1}{f} - 0.5V\frac{a}{f}}{VI_1t} \\
&= 0.5\frac{at}{I_1t} + \frac{I_1}{fI_1t} - 0.5\frac{a}{fI_1t} \\
&= 0.5\frac{a}{I_1} + \frac{1}{ft} - 0.5\frac{a}{fI_1t}
\end{aligned}$$

We can express the fractional noise 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 $k \approx 2000$.)

$$\begin{aligned}
&= 1/2k + 1/N - 1/2kN \\
&\approx 1/2k + 1/N \\
&= 1/4096 + 1/N
\end{aligned}$$

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 page 11.

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 3.) 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 3 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 4 gives transit times for two LAN packets, there and back.)

$(s_2 - s_1) = p_2 + t + p_3$

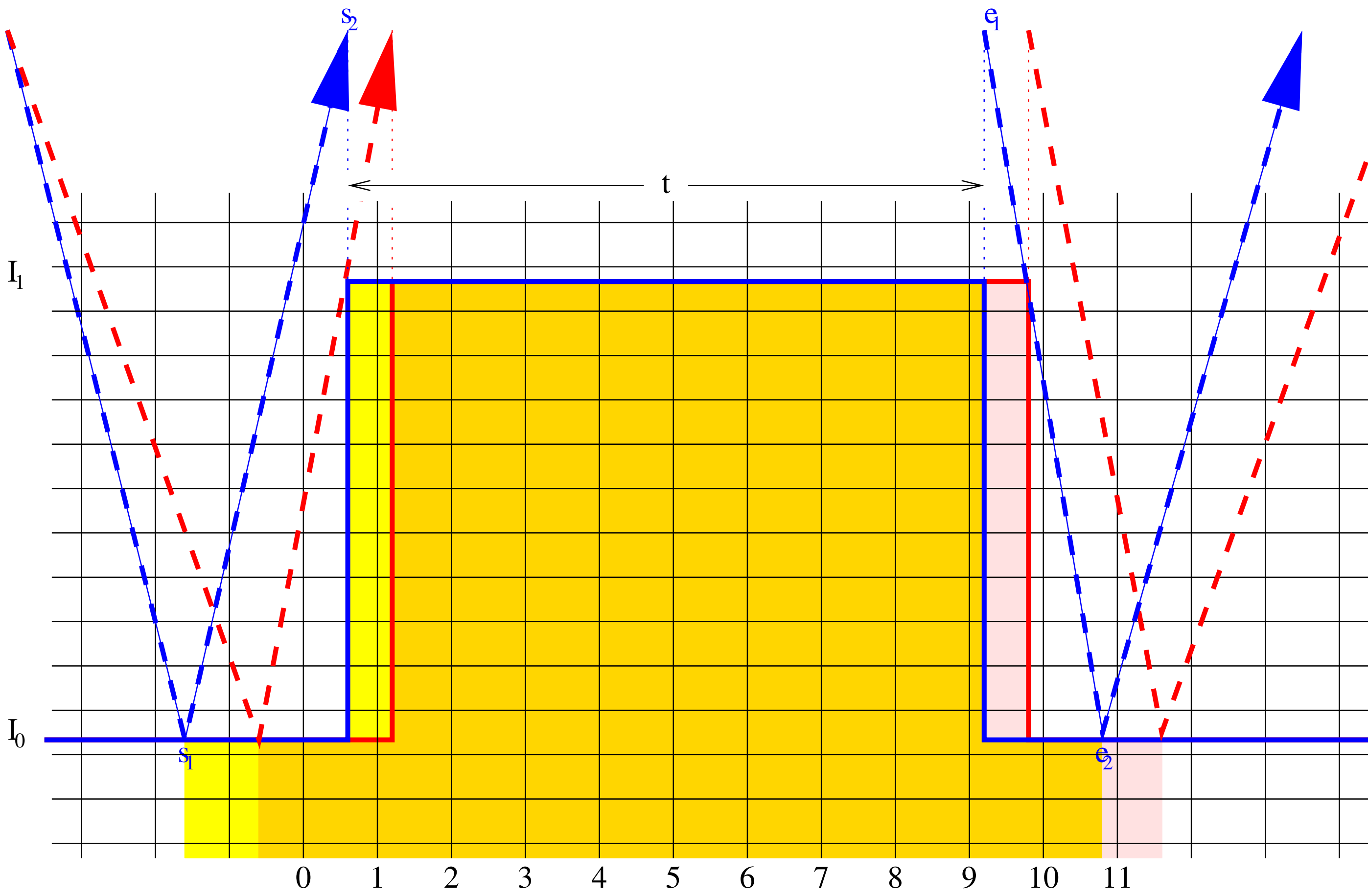

Figure 3: Measuring energy is initiated when the start message (left arrow) reaches the monitoring computer s_1. When the acknowledgement reaches the test computer s_2, it starts the experiment, raising the current from rest (I_0) to I_1. t seconds later the experiment finishes: the load drops back to I_0 and the test computer sends a message e_1 ending the measurement e_2. The experiment is done twice but different results are obtained since the network delays are different. As in Figure 2, energy used is given by area of under current curves times supply voltage (5 volts). Left (blue) $5\ 202.533\text{mA}\ 12.4\text{mS}{+}5\ (203.567{-}202.533)\text{mA}\ 8.6\text{mS} = 12.6015\text{mJ}$. Right (red) $5\ 202.533\text{mA}\ 12.2\text{mS} + 5\ (203.567 - 202.533)\text{mA}\ 8.6\text{mS} = 12.399\text{mJ}$. Difference $= 12.6015 - 12.399 = 0.2025\text{mJ}$, relative difference $= 0.2025/12.6015 \approx 1.6\%$.

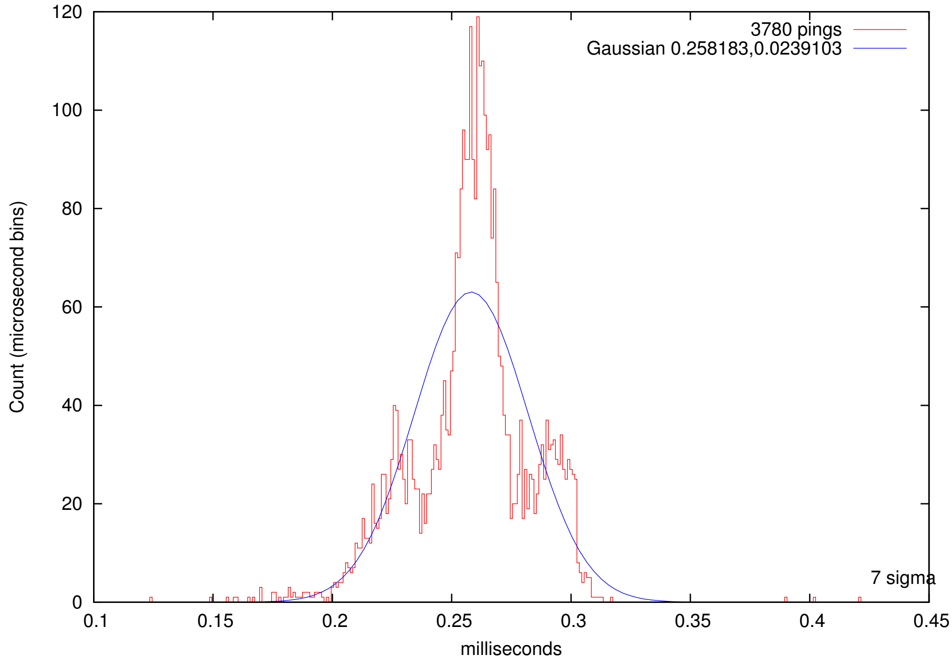

Figure 4: Distribution of network delays over 1 hour. Notice approximate match of Normal distribution and also long tail of much longer delays.

Measured energy is

$$\begin{aligned} &= V\Big(I_0(p_2+t+p_3)+(I_1-I_0)t\Big) \\ &= V\Big(I_0p_2+I_0t+I_0p_3+I_1t-I_0t\Big) \\ &= V\Big(I_0p_2+I_0p_3+I_1t\Big) \\ &= V\Big(I_0(p_2+p_3)+I_1t\Big) \end{aligned}$$

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) is:

$$\begin{aligned} &= V^2I_0^2\Big(\mathrm{var}\,(p_2)+\mathrm{var}\,(p_3)\Big) \\ &= 2V^2I_0^2\,\mathrm{var}\,(p) \qquad (2) \end{aligned}$$

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 measurement is

$$= \frac{\sqrt{2}\,VI_0\mathrm{SD}\,(p)}{V\,(2I_0p+I_1t)} = \frac{\sqrt{2}\,\mathrm{SD}\,(p)}{(2p+tI_1/I_0)}$$

Figure 4 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 page 4) 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

$\mathrm{var}\,(xy) = X^2\mathrm{var}\,(y) + Y^2\mathrm{var}\,(x) + \mathrm{var}\,(x)\mathrm{var}\,(y)$ [Goodman, 1960, Eq. 2].)
Remember (Eq. 1) discretization noise/V

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

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

$$\begin{aligned}
\mathrm{var}\left(at\ \mathrm{frac}\left(\frac{I_1}{a}\right)\right) &= a^2\mathrm{var}\,(t)/4 + a^2t^2/12 + a^2\mathrm{var}\,(t)/12 \\
&= a^2\mathrm{var}\,(t)/3 + a^2t^2/12 && (3)\\
\mathrm{var}\left(-\frac{a}{f}\mathrm{frac}\left(\frac{I_1}{a}\right)\mathrm{frac}\,((t-\delta)f)\right) &= \frac{a^2}{f^2}\left(\frac{1}{4}\times\frac{1}{12} + \frac{1}{4}\times\frac{1}{12} + \frac{1}{12}\times\frac{1}{12}\right) \\
&= 7/144\,\frac{a^2}{f^2} && (4)
\end{aligned}$$

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

$$\begin{aligned}
& I_1\left(\delta + \frac{1}{f}\mathrm{frac}\,((t-\delta)f)\right) - a\delta\ \mathrm{frac}\left(\frac{I_1}{a}\right) \\
=\; & I_1\delta + \frac{I_1}{f}\mathrm{frac}\,((t-\delta)f) - \delta a\ \mathrm{frac}\left(\frac{I_1}{a}\right) \\
=\; & \delta\left(I_1 - a\ \mathrm{frac}\left(\frac{I_1}{a}\right)\right) + \frac{I_1}{f}\mathrm{frac}\,((t-\delta)f)
\end{aligned}$$

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

$$\begin{aligned}
& \mathrm{var}\left(\delta\left(I_1 - a\ \mathrm{frac}\left(\frac{I_1}{a}\right)\right)\right) \\
=\; & \mathrm{var}\,(\delta)\,(I_1 - a/2)^2 + \delta^2\left(\mathrm{var}\,(I_1) + a^2/12\right) + \mathrm{var}\,(\delta)\left(\mathrm{var}\,(I_1) + a^2/12\right) && (5)
\end{aligned}$$

and of the second part

$$\begin{aligned}
\mathrm{var}\left(\frac{I_1}{f}\mathrm{frac}\,((t-\delta)f)\right) &= \frac{\mathrm{var}\,(I_1)}{f^2}/4 + \frac{I_1^2}{f^2}/12 + \frac{\mathrm{var}\,(I_1)}{f^2}/12 \\
&= \frac{\mathrm{var}\,(I_1)}{f^2}/3 + \frac{I_1^2}{f^2}/12 && (6)
\end{aligned}$$

Combining formulae 3–6 gives $\mathrm{var}\,(\text{discretization noise}/V)$ as:

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

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

Referring back to page 5, we have $t = N/f$ and $I_1 = ka$. Since the load and measurement computers are not synchronised $\delta = 1/2f$ and $\mathrm{var}\,(\delta) = 1/12f^2$ (Figure 2).

So Equation 7 becomes $\mathrm{var}\,(\text{discretization noise}/V)$

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

Assuming a 12 bit ADC and I_1 approximately half full scale.

$$\begin{aligned}
\mathrm{var}\,(\text{discretization noise}) &= \frac{V^2 a^2}{3}\mathrm{var}\,(t) + \frac{2V^2}{3f^2}\mathrm{var}\,(I_1) + \frac{V^2 a^2 t^2}{12} + \frac{V^2 a^2}{144f^2}\left(24k^2 - 12k + 14\right) \quad (8) \\
&\approx \frac{V^2 a^2}{3}\mathrm{var}\,(t) + \frac{2V^2}{3f^2}\mathrm{var}\,(I_1) + \frac{V^2 a^2 t^2}{12} + \frac{698880\,V^2 a^2}{f^2}
\end{aligned}$$

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 give by the sum of the variance due to network variation (Equation 2) and that due noise in the measuring system (Equation 8).

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

Assuming both t and I_1 are fixed

$$\mathrm{var}\,(\text{energy measurement}) = 2V^2 I_0^2 \mathrm{var}\,(p) + \frac{V^2 a^2 t^2}{12} + \frac{V^2 a^2}{144f^2}\left(24k^2 - 12k + 14\right) \qquad (9)$$

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.) If Δ^2 is large compared to the measurement variance (Equation 9) 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 9 means we can ask:

$$\text{Is } \Delta^2 \quad \text{much bigger than} \quad 2V^2 I_0^2 \mathrm{var}\,(p) + \frac{V^2 a^2 t^2}{12} + \frac{V^2 a^2}{144 f^2}\left(24k^2 - 12k + 14\right) \qquad (10)$$

Let $\Delta I = \Delta / Vt$ 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 ΔI can be expressed as the average reduction in current. Using $\Delta^2 = (\Delta I)^2 V^2 t^2$ in Question 10 and then dividing through by V^2 means Question 10 is the same comparison as:

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

Notice the last two terms do not depend on t and so for $\Delta I > a\sqrt{1/12}$ 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 $\Delta I = a\sqrt{1/12} \approx 0.3\ a$. If we require the signal to be at least twice the variability (4 times the variance) we can calculate the minimum time required.

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

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

$$\begin{aligned}
t &\approx \sqrt{\frac{8a^2k^2 \mathrm{var}\,(p) + \frac{a^2}{36f^2}\left(24k^2 - 12k + 14\right)}{a^2(\Delta k)^2 - a^2/3}} \\
&= \sqrt{\frac{24k^2 \mathrm{var}\,(p) + \frac{1}{12f^2}\left(24k^2 - 12k + 14\right)}{3(\Delta k)^2 - 1}} \qquad (11)
\end{aligned}$$

Alternatively we can express this minimum time (Eq. 11) as a minimum number of number of samples using $N = ft$ (page 5).

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

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

$$
N \approx k\sqrt{\frac{2}{3(\Delta k)^2 - 1}} \tag{12}
$$

Δk 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. (See definition of Δk on previous page.) If the average impact of the mutation is large compared to the resolution of the ADC, then $\Delta k \gg 0.58$. Therefore for our 1KHz 12 bit ADC and mutations with a reasonably large impact the measurement need only last $1.7/\Delta k$ seconds.

5 Discussion

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 [Langdon and Petke, 2015]. It also appears that the impact of beneficial software mutations is very non-uniform. 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. However it remains open to further research as to whether EC can find worthwhile energy improvements when only guided by mutations which individually have only a small impact.

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 expense of making more assumptions 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. However it appears these are not used by the existing power monitor.

6 Alternatives to Network Synchronisation Messages

Although network variability does not appear to be critical, here we record some ideas for avoiding it. The computers to be measured are linked to the computers hosting the power monitors via approx 4 meters of UTP cable and a high performance network switch. The network switch is needed to allow simultaneous measurement of multiple load computers by each measuring computer. The network is essentially idle except for management probe packets and the start-stop commands and so should not strain the network switch at all. Instead network delays are assumed to be primarily due to the complexities of passing packets through the complex time sharing operating system (Linux) and the application software at both ends.

6.1 Synchronisation via the Power Signal

It might be possible to use signal processing on the power load signal itself to recognise the onset and termination of the measurement period. Thus avoiding using the network for energy measurement at all. If the pure signal was not always sufficiently clear, one could imagine adding known disturbances before and after the signal. For example, three increases of a set amount in the load a known time before onset and three decreases a known time after the load to be measured has stopped.

6.2 Synchronisation via Additional Trigger Hardware

In the case of oscilloscopes, it is common to use an external trigger. Several low end test beds (e.g. the raspberry pie) have easily accessible output pins which could provide a trigger synchronised to a software event relevant to measuring each mutants fitness.

6.3 Synchronisation via Absolute Clocks

Both the computer under test and the computer running the energy monitors have sophisticated clocks. [Ridoux and Veitch, 2010, Fig. 5] suggest even Unix's ntpd clock utility can on average keep computer time globally synchronised to within 40μ. (Figure 4 suggests the typical variation between two Ethernet messages is 24μ.) Since 40μ refers to synchronisation across the World, it would seem reasonable that it should be possible to do at least as well across a LAN where each computer is within 4 meters and thus maintain much better local consistency during the course of an experiment. ([Ridoux and Veitch, 2010, Fig. 5] suggests clock variation does not follow a Gaussian distribution. However, they do not mention a problem with long tails of large variations.) Hence it may be possible avoid noise introduced by variation in network delays by accurately recording the start and end of each energy loading experiment and ensuring the computer clocks are locally synchronised.

7 Conclusions

The results at the bottom of page 11 suggest:

- It will be difficult to detect mutations which have on average an effect less than $\sqrt{(1/3)}a$ on the current consumed. For our example 12 bit ADC this sets a lower limit of 57μA.
- On the other hand if the effect is much bigger than 57μA, there is little to be gained by running measurement for longer than a second. Equation 12 suggests the ideal duration falls in proportion to the smallest effect size we wish our GA to detect.

References

[Bruce, 2015] Bobby R. Bruce. Energy optimisation via genetic improvement A SBSE technique for a new era in software development. In William B. Langdon et al., editors, *Genetic Improvement 2015 Workshop*, pages 819–820, Madrid, 11-15 July 2015. ACM.

[Darwin, 1859] Charles Darwin. *The Origin of Species*. John Murray, penguin classics, 1985 edition, 1859.

[Goldberg, 1989] David E. Goldberg. *Genetic Algorithms in Search Optimization and Machine Learning*. Addison-Wesley, 1989.

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

[Koza *et al.*, 2003] John R. Koza, Martin A. Keane, and Matthew J. Streeter. Evolving inventions. *Scientific American*, 288(2):52–59, February 2003.

[Langdon and Petke, 2015] William B. Langdon and Justyna Petke. Software is not fragile. In Paul Bourgine and Pierre Collet, editors, *Complex Systems Digital Campus E-conference, CS-DC'15*, Proceedings in Complexity, page Paper ID: 356. Springer, September 30-October 1 2015. Invited talk, Forthcoming.

[Langdon, 2011] William B. Langdon. Generalisation in genetic programming. In Natalio Krasnogor et al., editors, *GECCO '11: Proceedings of the 13th annual conference companion on Genetic and evolutionary computation*, page 205, Dublin, Ireland, 12-16 July 2011. ACM.

[Langdon, 2015] William B. Langdon. Genetically improved software. In Amir H. Gandomi et al., editors, *Handbook of Genetic Programming Applications*, chapter 8, pages 181–220. Springer, 2015.

[Ridoux and Veitch, 2010] Julien Ridoux and Darryl Veitch. Principles of robust timing over the internet. *Communications of the ACM*, 53(5):54–61, May 2010.

[Schulte *et al.*, 2014] Eric Schulte, Jonathan Dorn, Stephen Harding, Stephanie Forrest, and Westley Weimer. Post-compiler software optimization for reducing energy. In *Proceedings of the 19th International Conference on Architectural Support for Programming Languages and Operating Systems, ASPLOS'14*, pages 639–652, Salt Lake City, Utah, USA, 1-5 March 2014. ACM.

[Stender, 1993] Joachim Stender, editor. *Parallel Genetic Algorithms: Theory and Applications*. IOS press, 1993.

[Thompson, 1996] Adrian Thompson. Silicon evolution. In John R. Koza et al., editors, *Genetic Programming 1996: Proceedings of the First Annual Conference*, pages 444–452, Stanford University, CA, USA, 28–31 July 1996. MIT Press.

[White *et al.*, 2008] David R. White, John Clark, Jeremy Jacob, and Simon M. Poulding. Searching for resource-efficient programs: low-power pseudorandom number generators. In Maarten Keijzer et al., editors, *GECCO '08: Proceedings of the 10th annual conference on Genetic and evolutionary computation*, pages 1775–1782, Atlanta, GA, USA, 12-16 July 2008. ACM.