

# Importance of the Size of the Period of the Duty Cycle in `insertDelay.c` for use in DR Control Loop

Joseph Hall

October 18, 2015

## Abstract

This report shows the importance of the size of the duty cycle period relative to the size of the power measurement window used for calculating the power. If the period of the duty cycle is on the same order of magnitude as the power window, then the reported power has very large variance due to the fact that the computer will consume relatively large amounts of power during the pulse duration of the cycle. When the window over which we estimate the power is of similar length as the pulse duration, that power estimate will be very large compared to the power estimate during the idle duration.

## 1 Introduction

`insertDelays.c` is program in C written by Josiah which takes an input  $u[kh] \in (0, 1)$ , where  $k$  is the time index and  $h$  is the nominal time period between samples. The output of `insertDelays.c` are Linux signals, SIGSTOP and SIGCONT, which are sent to a specified process (including all its children and threads). The value of  $u[kh] \in (0, 1)$  can be thought of as a psuedo duty cycle for the process. The actual duty cycle is  $\frac{T_{work}}{T_{idle} + T_{work}}$ . In `insertDelays.c`,  $T_{work}$ , is fixed and can be specified as a commandline argument.  $u[kh]$  only changes  $T_{idle}$ . Several functions are possible.

$$T_{idle} = (1 - u[kh]) * X, \text{ where } X \text{ is some value} \quad (1)$$

In Josiah's original code, he used Eq. (1), where  $X$  was the time elapsed since that particular point in the code loop. This effectively subsummed the time required to update states `insertDelays.c` itself into  $T_{work}$  and therefore into  $T_{idle}$ . I have found that fixing  $X$  as a constant produces power output with less variance at the when the psuedo-duty  $\approx 50\%$ . I have yet to determine a clear rationale for this difference. A second alternative produces the true duty cycle

from  $u[kh]$  by Eq. (2):

$$\begin{aligned}
D &= u[kh] \\
&= \frac{T_{work}}{T_{work} + T_{idle}} \\
\Rightarrow T_{idle} &= \frac{T_{work} - T_{work}D}{D} \\
\Rightarrow T_{idle} &= T_{work} * (\frac{1}{u[kh]} - 1)
\end{aligned} \tag{2}$$

Best seems to be Eq. (??) with  $T_w = 1ms$ , because the  $argmax_{u[kh]}(T_{idle}) = 0.999sec$  and  $argmin_{u[kh]}(T_{idle}) = 1\mu sec$

## 2 Definitions

Definitions in this document are as follows:

Power consumed by the server at time,  $t$ , is reported as two different estimates *PKG* and *DAQ*:

- $PKG = \frac{E(t)}{T_w}$ , where  $E(t)$  is the value of the energy counter MSR (via the RAPL interface) since the last time it was read and  $T_w$  is the time elapsed since it was last read.
- *DAQ*

$$\alpha = \sqrt{\beta} \tag{3}$$

### 2.1 Subsection Heading Here

Write your subsection text here.

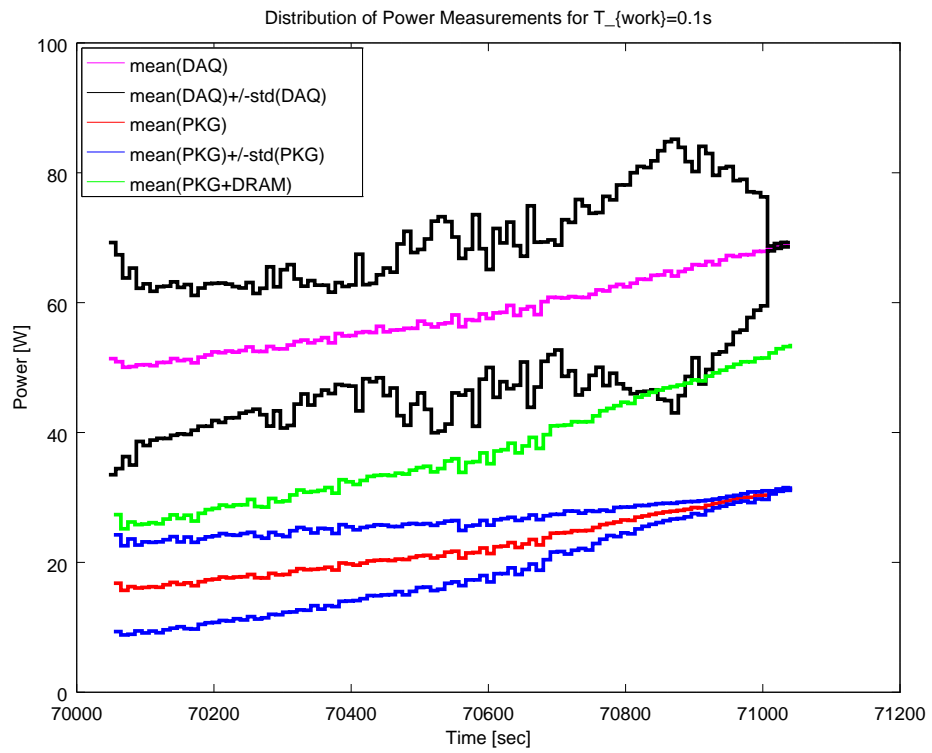


Figure 1: Simulation Results

### **3 Conclusion**

Write your conclusion here.