University of Waterloo

Faculty of Engineering  
Department of Electrical and Computer Engineering

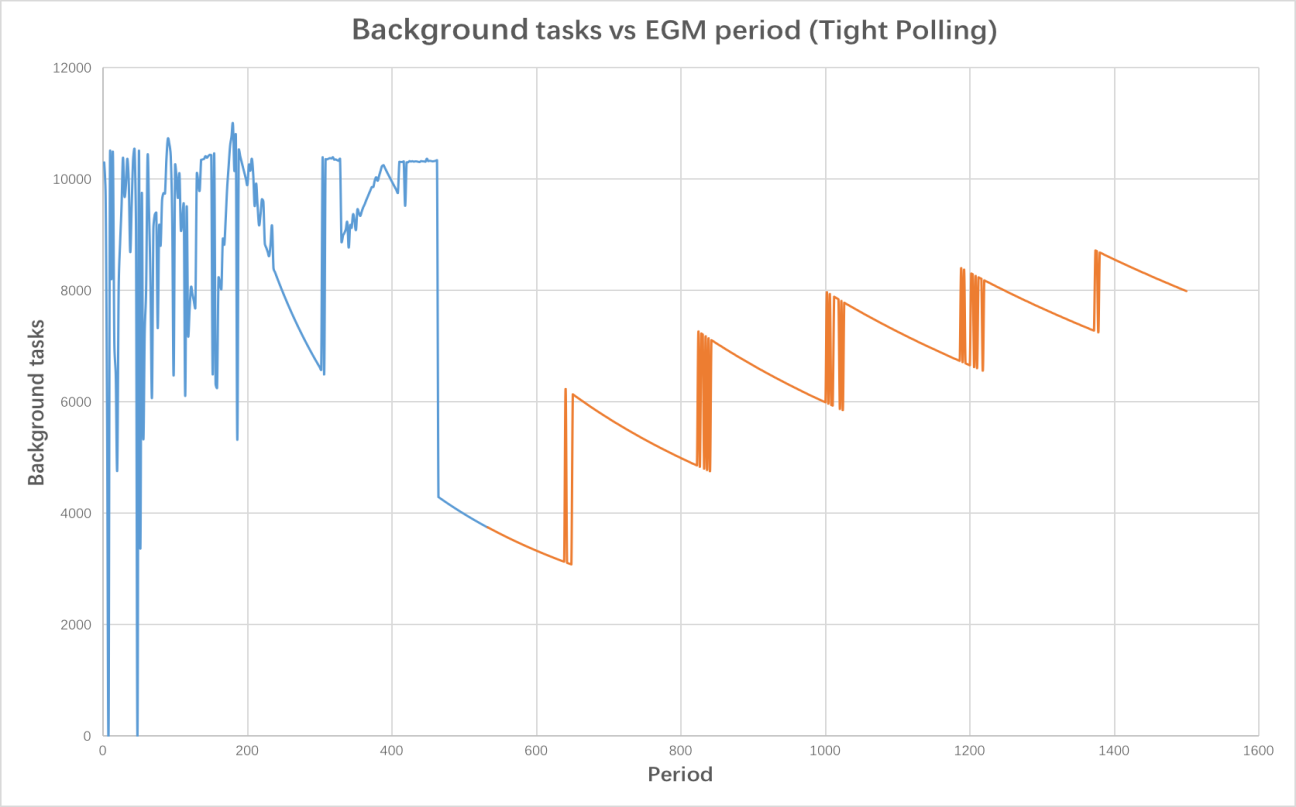
Interrupts vs. Tight Polling

University of Waterloo

200 University Ave W,

Prepared by  
Joseph Balawejder and Jinming Zhang  
20620064 and 20613667  
jbalawej@uwaterloo.ca and Jm3zhang@edu.uwaterloo.ca  
2B Computer Engineering  
20 October 2017

1. Tight Polling Background Tasks vs EGM Period



Non-zero missed pulses

Zero missed pulse

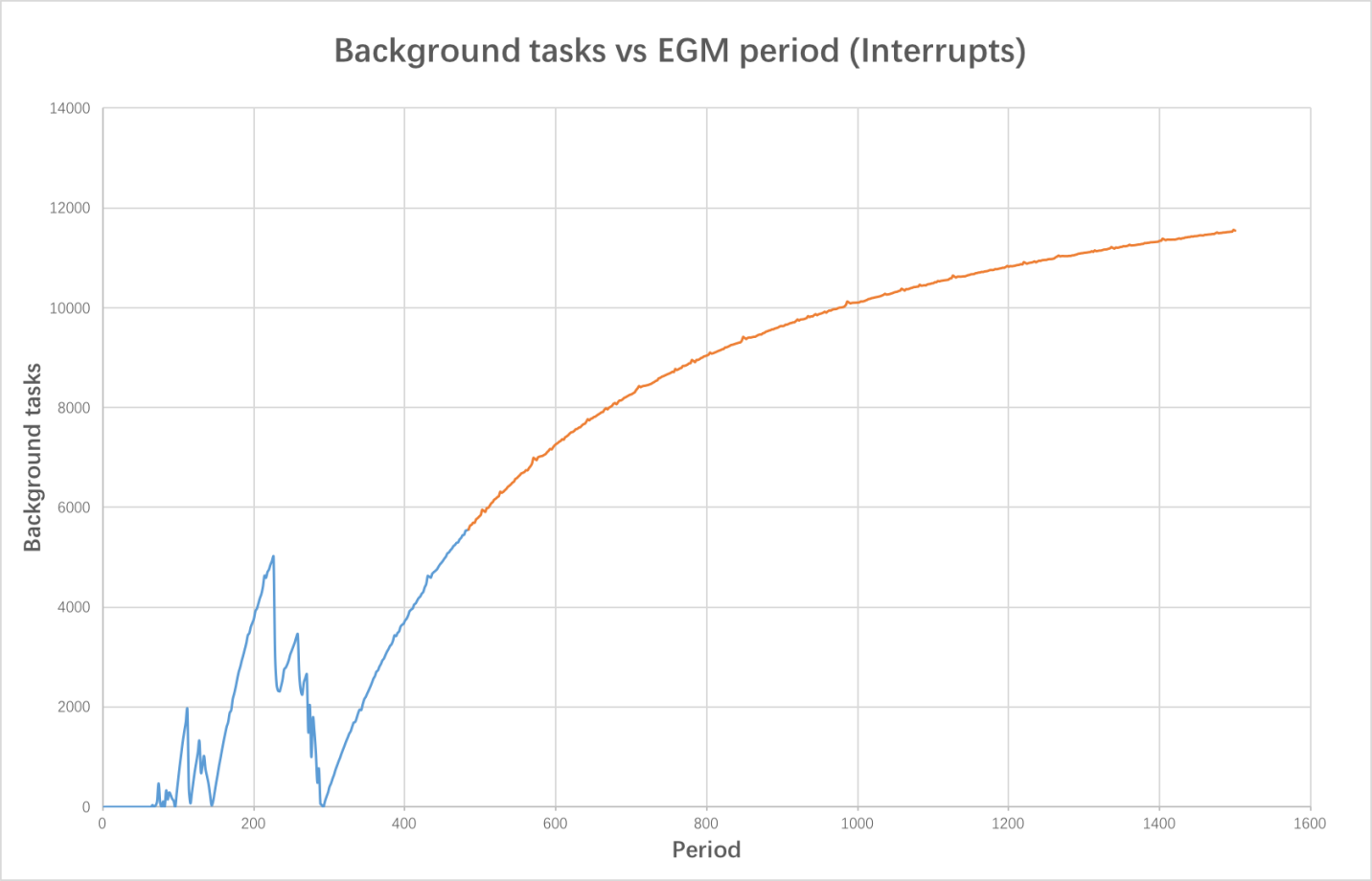
The root cause for missing pulses is because the stimulus is too fast for a response to be sent in time thus responses will be lost when looking for the next rising edge of the stimulus.

The root cause for overall trend line is that the total amount of background tasks are increasing over time. Since the frequency gets larger, the total number of cycles decreases thus the size of jumps decreases as the slope flattens out.

The root cause for discontinuities is because the number of tasks that can be executed corresponds directly to the initial characterization. There is enough time for the stimulus in to complete an extra background task before tight polling resumes. Thus an additional task is completed each cycle but the period is about the same therefore there is a jump discontinuity.

The root cause for negative sloped curve between discontinuities is because the period is increasing but the characterization does not change. As a result of the period increasing, the total number of background tasks that can be executed decreases with a negative slope.

1. Interrupt Background Tasks vs EGM Period



Zero missed pulse

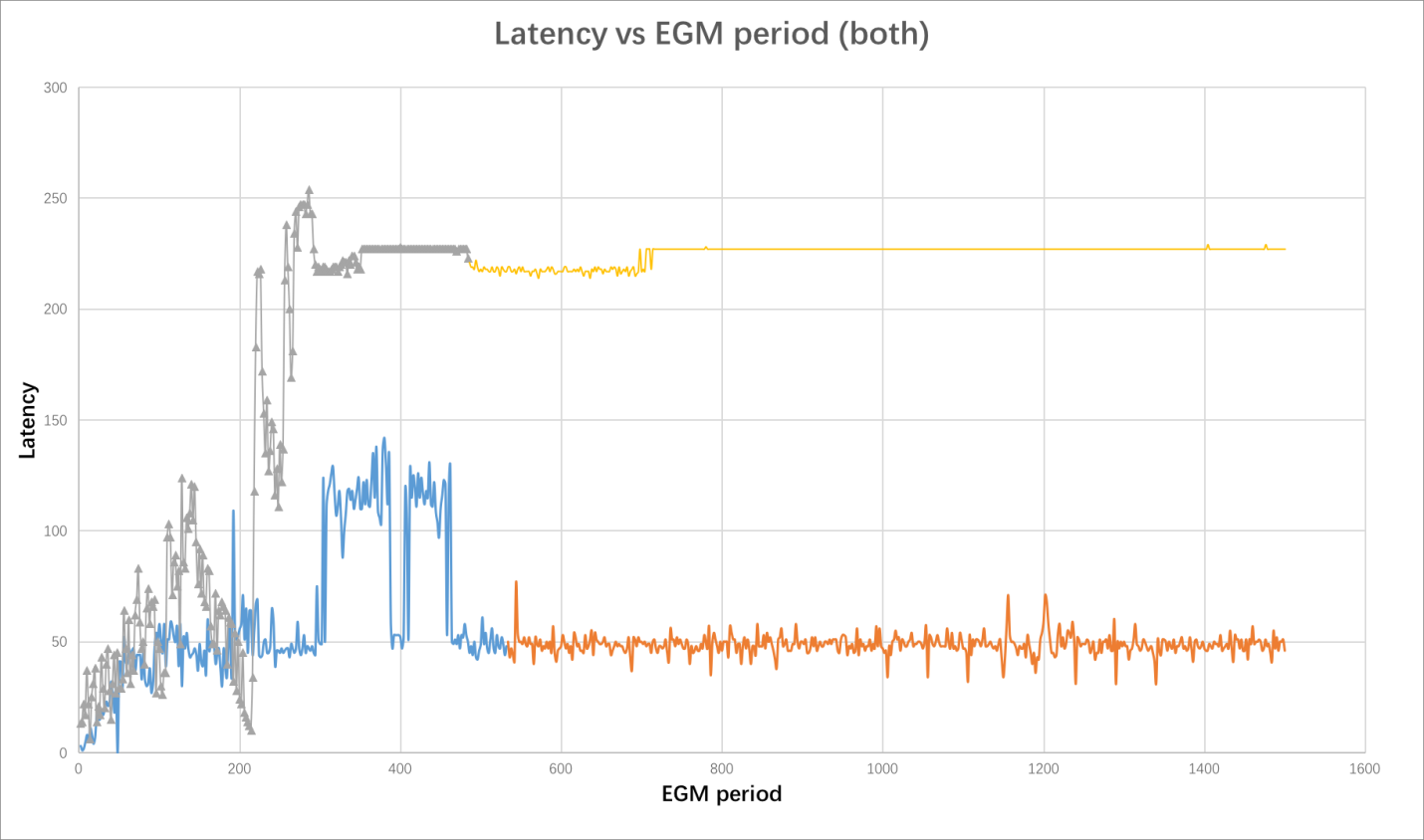
Non-zero missed pulses

The root cause for missing pulses at the start of the experiment the period is too small and the EGM sends interrupts too fast so the response cannot be generated in time before the stimulus pulse ends.

The root cause for overall trend line looks like a logarithmic curve because as the interval between stimulus gets larger most of the time is spent on processing tasks rather than the ISR. It levels off at the end because the EGM period gets really big, resulting a change in the number of interrupts serviced will not change the number of background tasks that much.

Interrupts have a much greater background task efficiency compared to tight polling because the system does not need to constantly check the stimulus and estimate the number of background tasks per cycle. Background tasks are focused on and then interrupted in the interrupt case.

1. Latency data for both modes in a single plot (Latency vs EGM period 2-1500 clock cycles)



Differences between period where latency is stable

Differences between stable latency

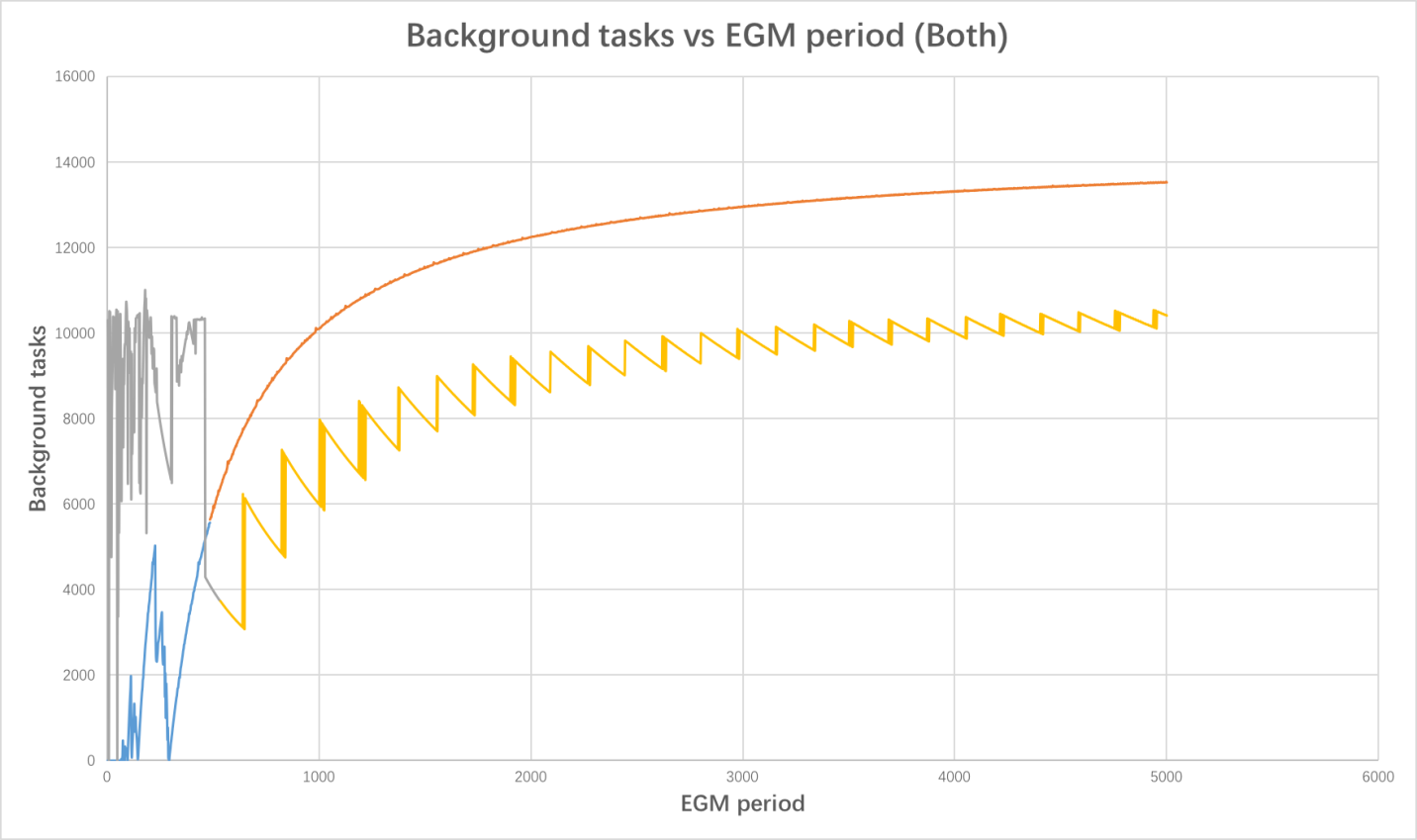
Tight polling(in blue and orange) from the graph has a much smaller response latency compared to interrupts (in grey and yellow). This makes sense because in tight polling it is continually checking thus the latency or the amount of error for the response is very small. Interrupts is much larger because it is not always checking.

Another point is that the latency for tight polling is generally low and independent of the frequency of the EGM pulses. Because of the characterization run, the system will always be polling right before a pulse arrives. There is a little bit of a higher latency for EGM periods in between around 300 and 500 this may be at the cause of missed cycles.

For interrupts when the EGM period is very short less than around 300 the latency is quite small compared to when the EGM period is greater than around 400. This may be because interrupts are arriving very frequently. Thus, before the system can unwind the stack and restore it to a state where the background task can execute, another interrupt has arrived.. This is in line with the number of background tasks preformed because while the EGM period is small, the number of background tasks is 0 or a small amount. Therefore the low latency could be due to the fact there are extremely frequent interrupts resulting in the system being able to send response signals very quickly.

Another difference is that when the interrupt and tight polling trend lines level off interrupt is a stable straight line whereas tight polling is an unstable straight line. This is because the ISR is expected to be served at a constant time with regard to the stimulus pulse. Whereas, tight polling does not run constantly with regard to the stimulus pulse therefore this introduces instability.

1. Background tasks data for both modes in a single plot (Background tasks vs EGM period 2-5000 clock cycles)



Tight polling performs more tasks

Interrupts perform more tasks

From the graph we can see that for EGM periods greater than 500, using interrupts allows the system to consistently execute more background tasks than tight polling. The main difference is characterization, a task must either go to completion or never even start. This is because the system needs to be polling when the next pulse arrives and the CPU cannot do background work. On the other hand, interrupts, can have a background task be halfway through executing, get interrupted, and then continue. Thus, when the EGM period is fairly large, using interrupts allows for more background tasks to be completed because semi completed tasks can occur.

When the EGM period is less than 500, the amount background tasks performed with tight polling is much greater compared to interrupts. This is because servicing an interrupt takes a long time. Handling an ISR takes much longer than tight polling, this can be concluded from the response latency of the previous graph. The CPU is completely used when handling interrupts, it never has the chance to resume to execute a background task. Thus, the system cannot execute a lot of background tasks (sometimes none). On the other hand for tight polling as long as the characterization determines that a single background task can be executed in the cycle, there will be a significant amount of background tasks executed. This is because the frequency of the pulses is so high. Although for both methods high frequencies will miss EGM pulses. However, using tight polling will still allow for background task execution while interrupts will not.

Therefore, interrupts are effective for large EGM frequencies and tight polling is effective for small EGM frequencies, with respect to the number of background tasks able to be executed.

Tight polling and intersect lines intersect at about 500 for the EGM period.