Politecnico di Torino

# Energy management for IoT application

# Report laboratory session 1

Author: Grottesi Lorenzo

# INTRODUCTION

Dynamic power management is a technique developed for the improvement of the power consumption of a generic system. The main goal of DPM is to reduce the switching activity, and consequentially the power consumption. Given a generic IoT system is possible to generate a performance – power plain that will describe the system power curve. The basic idea is that when high performance is required, the device enters in an active power consumption state. When, instead, low performance is required the system is carried into a low power state. The average power consumption will be improved. In order to change the power state of the system, a power state machine (PSM) must be improved inside the system itself. The general DPM structure of the whole system is summarized in the figure 1. A monitor will take the workload as input and the power state is set accordingly to policies implemented.

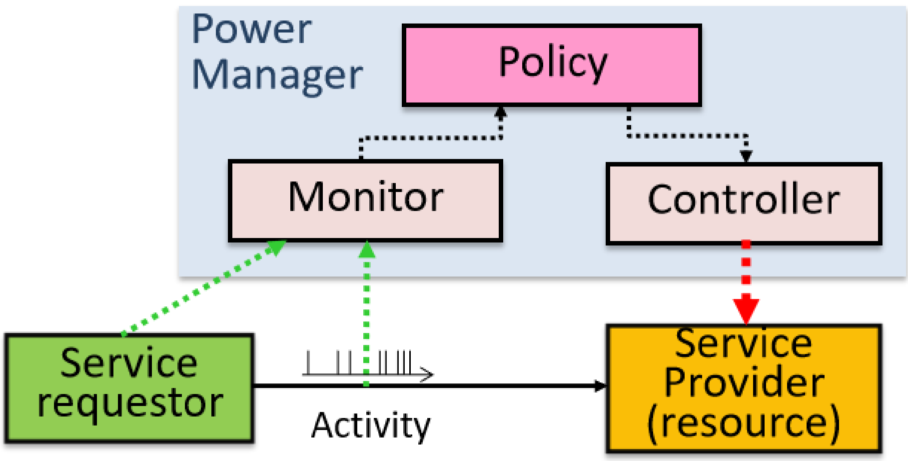



**Figure 1 - DPM structure**

The goal of the laboratory is to perform simulations of a DPM system implementing a PSM. In particular, some workloads will be generated using different probability distributions and some policies will written in order to reduce the power consumption. The PSM implements three states ACTIVE, IDLE, SLEEP with different energy consumptions.

The goal is generating the workloads and change the code simulation in order to implement two policies:

- Timeout based policy
- History based policy

## Workload generation

In order to simulate the two policies some workloads must be generated. In particular the workloads have been generated using active times uniformly distributed between 1 and 500 us. The idle periods, instead has been generated using several distributions:

- Uniform distribution 1 to 100
- Uniform distribution 1 to 400
- Gaussian distribution (mean: 100 , var: 20)
- Exponential (mean: 50)
- Trimodal distribution

All the samples of the distributions are independent among themselves due to the independence of the generation. The plot of the frequency of the outcomes is reported in figure 2.

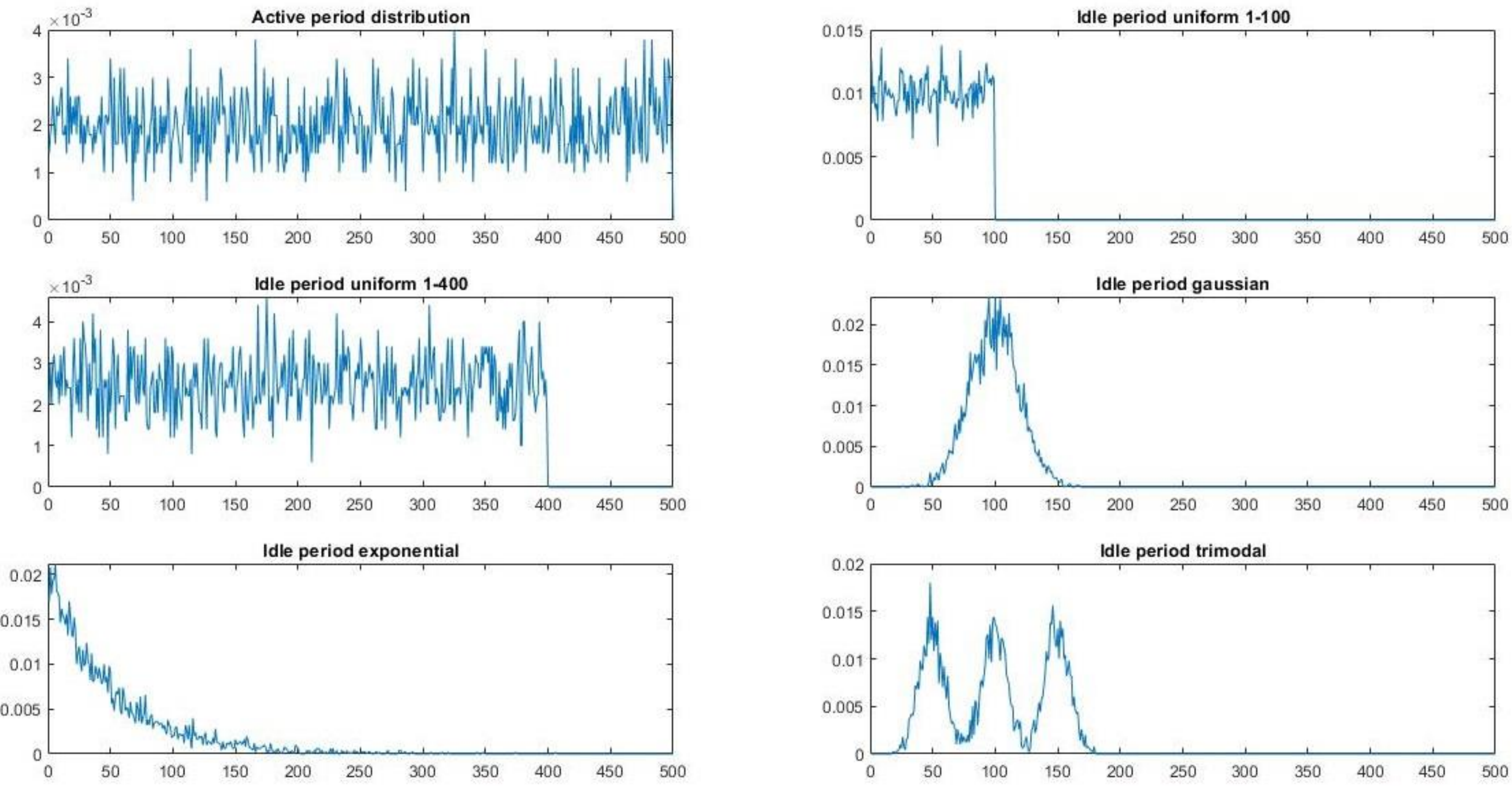



**Figure 2 - Distributions**

All the profiles as been generated using Matlab functions. The trimodal workload has been generated sweeping the three gaussian till the number of samples is reached.

After the generation, the algorithm prints the workload in a simulator readable format. The file containing the intervals is read and the distribution re-extracted in order to check the correctness of the workload. The file performing the generation is “workloadGenerator.m”.

# TIMEOUT POLICIES

Once the workloads have been generated, the simulation can start. In particular two kind of timeout policies has been simulated, one considering only the idle state and another considering both idle and sleep states. In the first case given an input timeout, the system passes from the active state to idle state. In the second, once the system is in an active state it can decide to pass into a sleep state. To compare these two simulation policies, an automatic bash script has been written. The “timeout_script.sh” sweeps the timeout values from 5 to 400 us by a step of 5 us. The bash script executes both simulations considering idle and idle-sleep states. The result for both cases is reported in a csv file containing the value of the timeout and the relative energy saving. The csv file contains then the statistics of the single profiles. A dedicated Matlab script has been developed to read and visualize the results. The name of the file is “plotStatistics.m”.

In the following sections, the distributions profile will be discussed one by one.

## Uniform distribution 1 to 100

This distribution profile considers a high utilization from the active time point of view. Indeed, the workload enters in idle state just for few microseconds meanwhile stay in active for many hundred seconds. Knowing the distributions is possible to compute the expected values as:

$$E(T\ active) = \frac{500 + 1}{2} = 250{,}5\ us$$

$$E(T\ idle) = \frac{100 + 1}{2} = 50{,}5\ us$$

In this distribution, the bash program has been modified in order to sweep the timeouts from 5 to 100 using step of 5 us.

The figure 3 represents the relation between timeout value and the energy saving considering a workload using this distribution profile. As possible to imagine the timeout policy increases the amount of power instead of reducing it. This is reasonable because the idle periods are, on average, too small with respect the transition time and the relative energy. This situation led to a loss of power when changing the state from active to idle. Is also possible to notice that energy saved is zero when the timeout reaches the 100, this is given by the fact that the transition is never taken, due to high value of threshold timeout. This last point corresponds also to the maximum energy saved of the function. Knowing the distribution is also easy to understand that rising the sweep maximum bound (e.g. from 100 to 400 us, as the other distributions) the function will remain stable to zero.

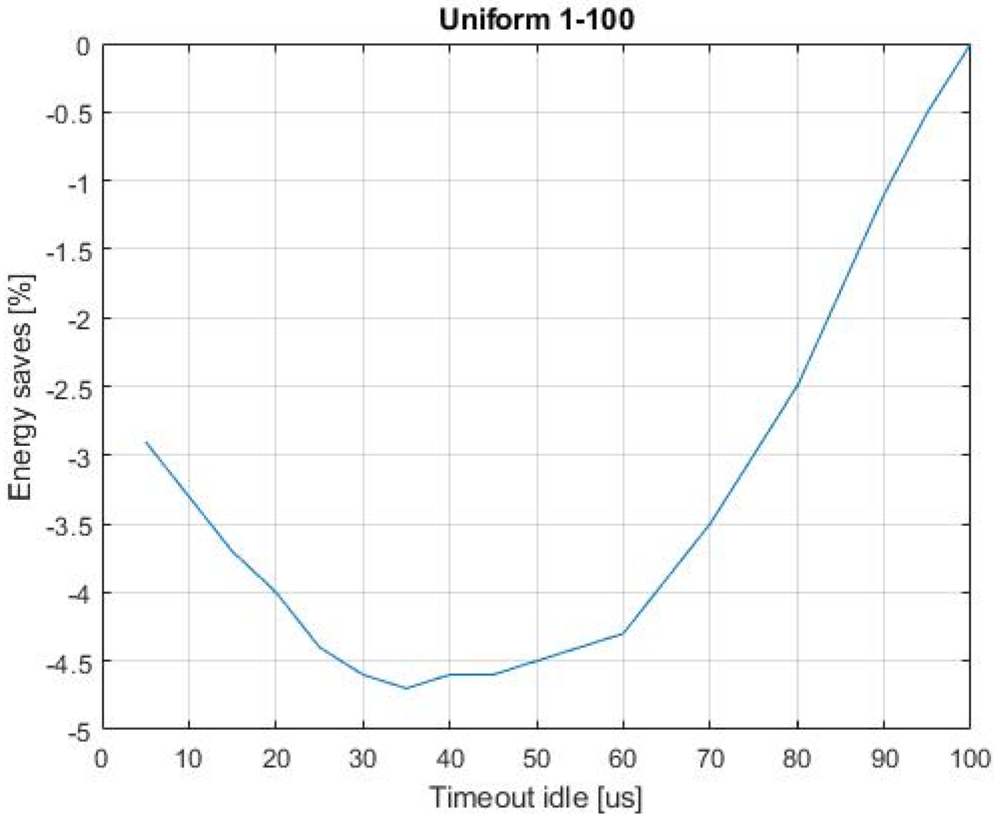



**Figure 3 - Energy saving w.r. idle timeout for uniform 1-100 distribution**

With the same profile, the plot considering also the possibility to go into sleep state has been considered and reported in figure 4. As possible to see there is not any advantage to enter into another state. Furthermore, in the proximity of the origin the energy saving is much more negative than before. The reasons are the same as before.

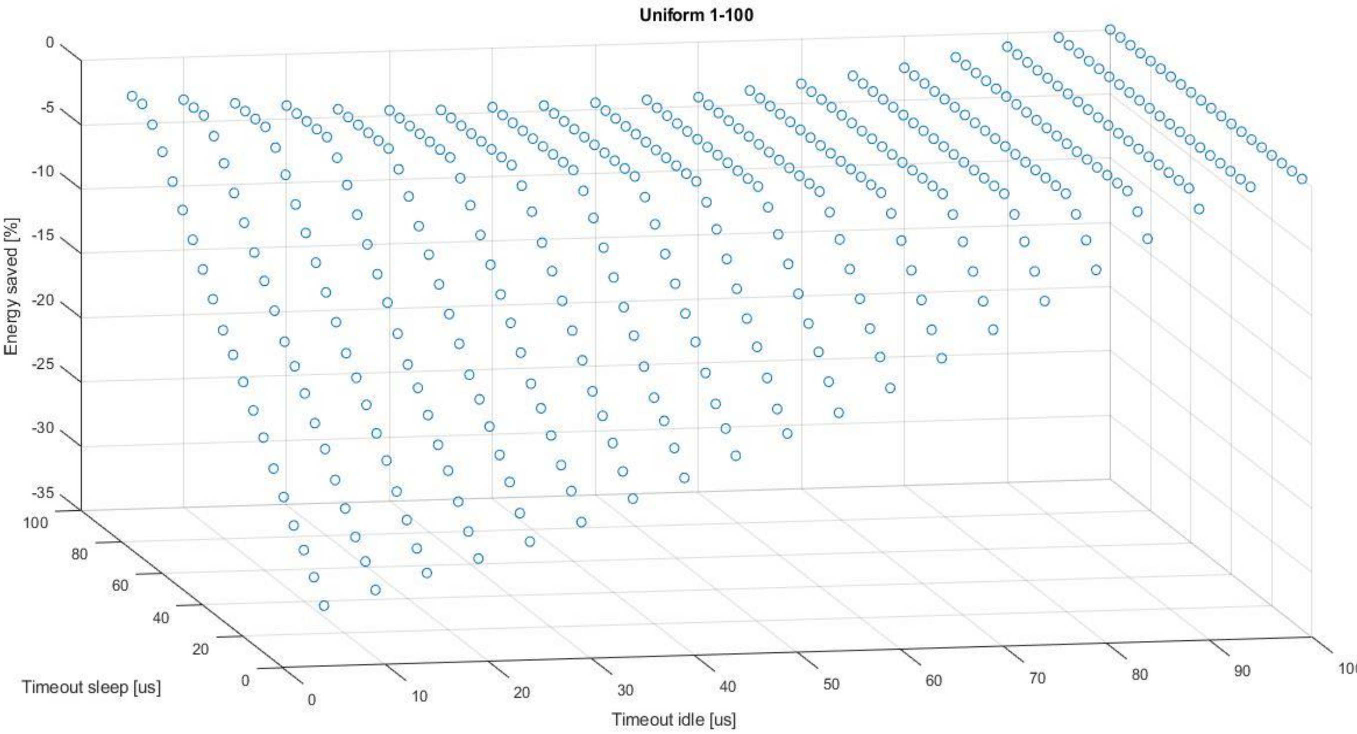



**Figure 4 - Energy saving w.r. idle and sleep state**

## Uniform distribution 1 to 400

In this case, the profile considers a wider range of idle times. This time

$$E(T\ idle) = \frac{400 + 1}{2} = 200{,}5\ us$$

That is closer to the expected value of the active time distribution, this means that the profile refers to a low utilization of the workload. As can see in figure 5 this time the timeout policy related to the idle time is much convenient. Indeed, near the origin there is the maximum energy saving that is around the 27 %. Arising the timeout, value the energy saved decreases until a negative effect between 300 and 400 us is produced. This is given by the fact that this range of timeouts make the transition closer to the end time of the idle period. So, would be possible to take the transition from active to idle but the idle time is not enough large to save energy. in this case When timeout reaches 400 the energy saved is zero.

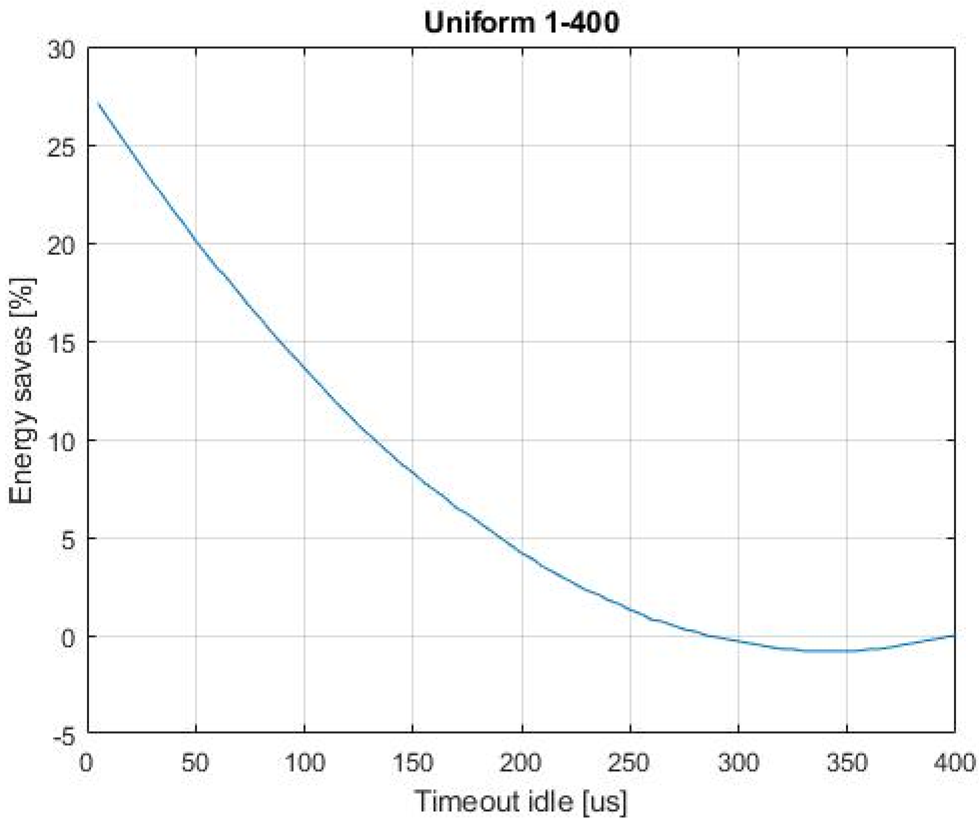



**Figure 5 - Energy saving w.r. idle uniform 1-400**

Considering also the presence of sleep state, the energy saving is reported in the following plot.

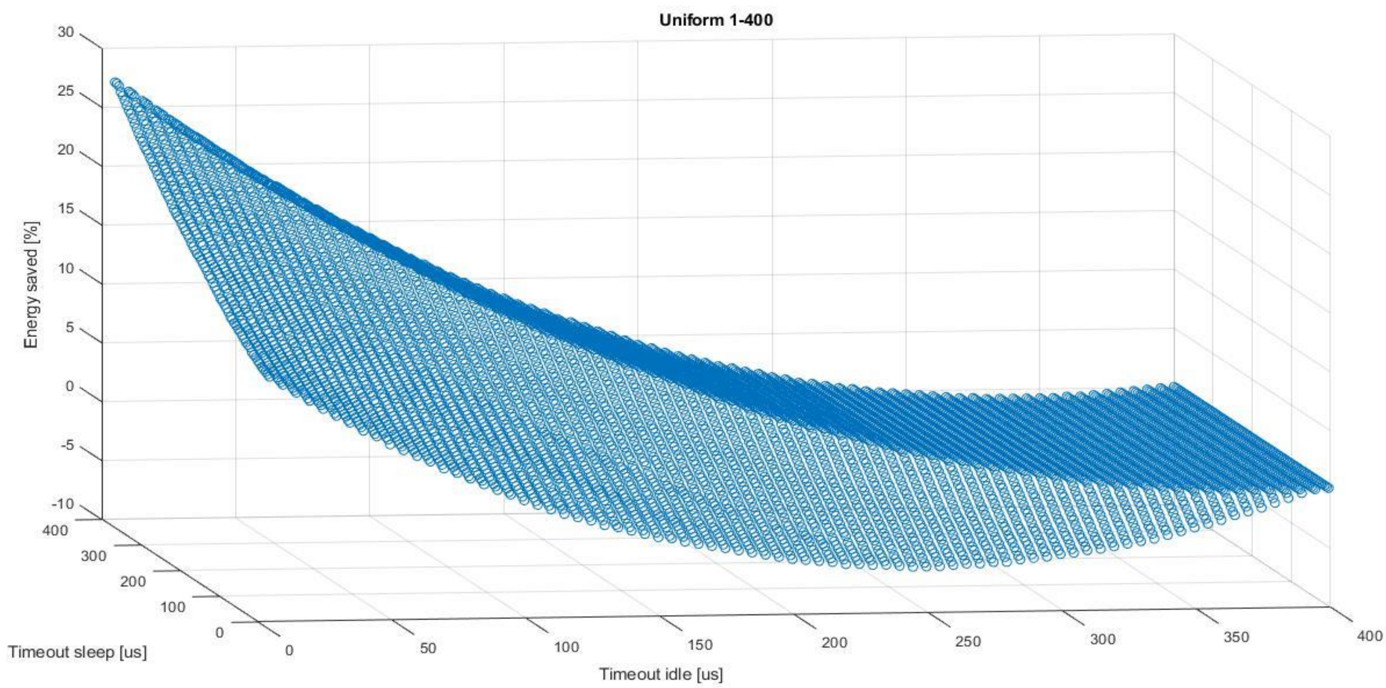



**Figure 6 - Energy saving w.r. sleep and idle uniform 1-400**

From the graph is possible to see that there is an energy saving near origin, but the maximum value of saving is when timeout goes around 400 us that bring the situation back to the previous case. Therefore, this case there is convenience to use both state but the best choice will be using only idle state.

## Exponential distribution

The exponential distribution has been generated with a mean around 50 us. This profile represents a high utilization workload. From the figure 7 and 8, is possible to see that like the uniform 1-100 any timeout policies actually don't save energy for this profile. Comparing results with the uniform (high utilization) is possible to notice that the energy savings are less negative in this case. This is probably given by the fact that this distribution is able to generate some high duration idle periods, although with very low probability and frequency.

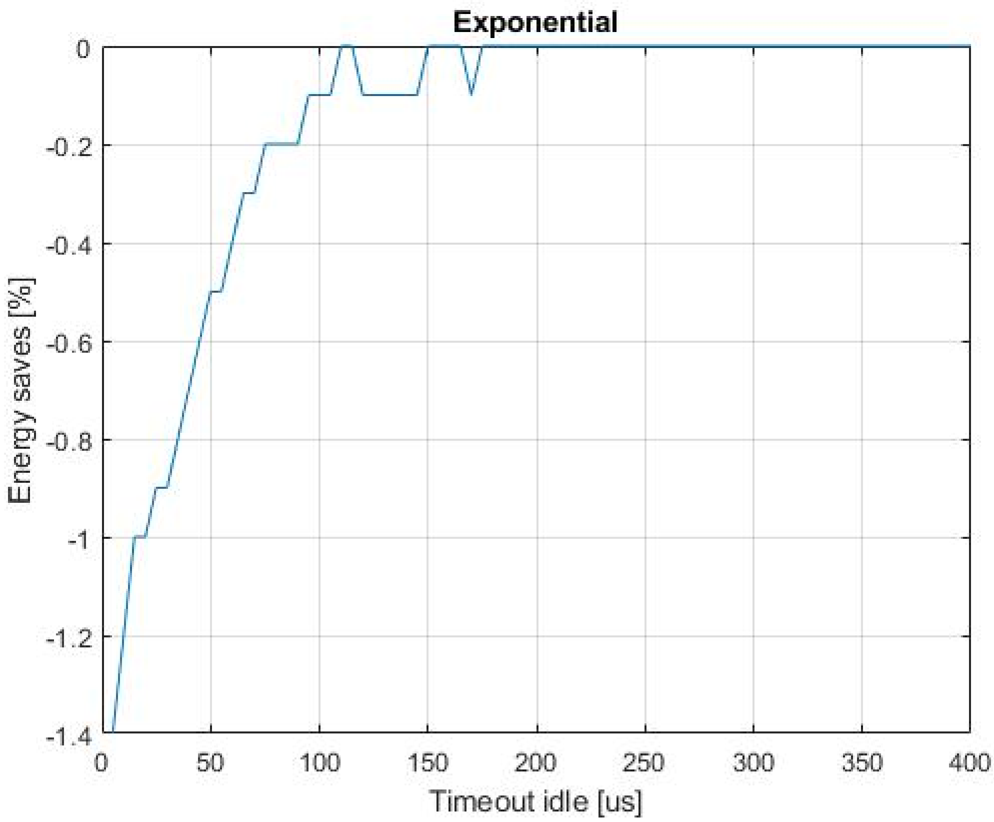



**Figure 7 - Energy saving w.r idle exponential**

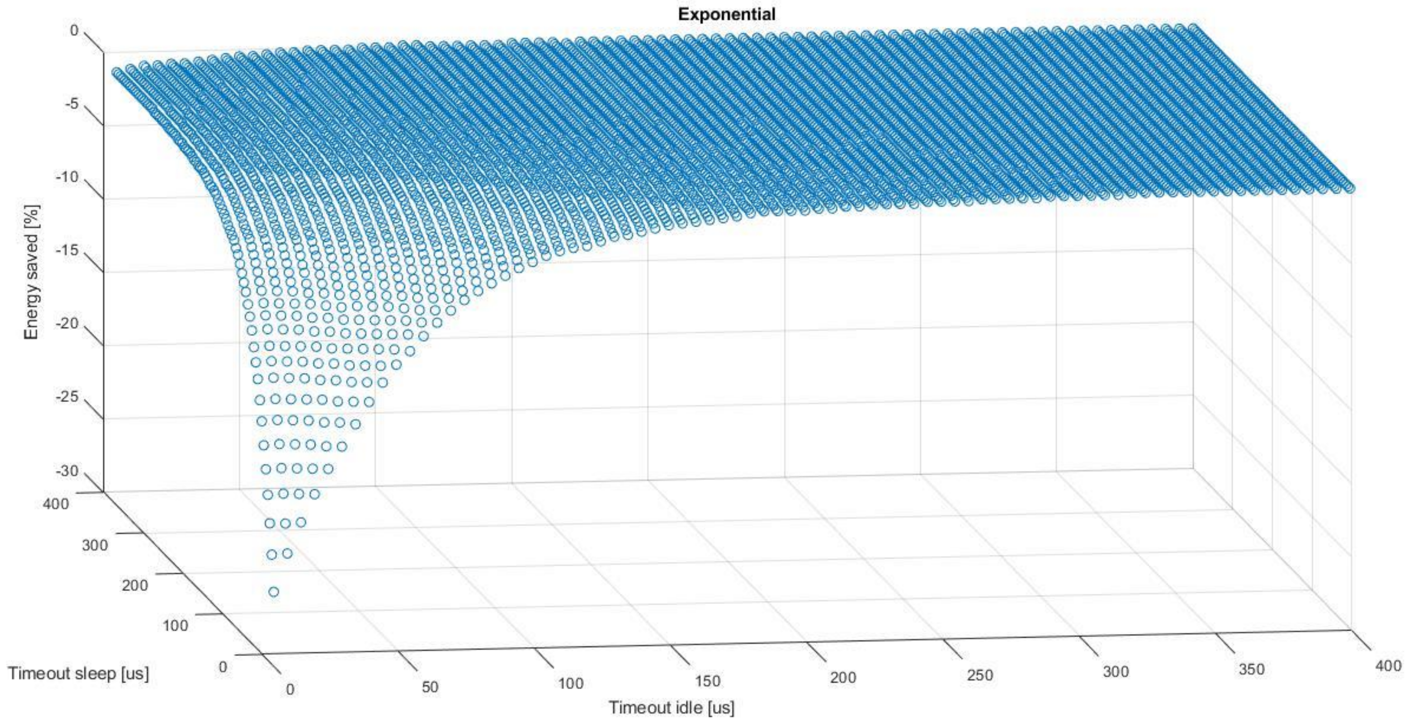



**Figure 8 - Energy saving w.r. idle and sleep exponential**

## Gaussian distribution

In this case the workload idle times are generated from a Gaussian with mean 100 us and variance 20 us. Figure 9 represents the relation between energy saving and timeout for idle state. For low threshold value the energy saving is maximum, around 10%. Increasing the timeout value, the energy saving decreases. The energy saving reaches the zero around the timeout value: 40 us. Then the energy saving reaches a minimum and start increasing till the asymptotic zero power saving. Looking at the curve could have sense to find some relations among the regions of the function and the parameters of the Gaussian. Indeed, possible to compute the state in which the energy saving is positive.

$$40 = 100 - x * 20 => x = \frac{100 - 40}{20} = 3$$

Could be reasonable to generalize the given formula for a generic Gaussian mean μ, variance σ would be possible to say:

$$Energy\ saving > 0, when$$

$$Timeout < \mu - 3 * \sigma$$

This equation cannot be proven by just one Gaussian statistic, but could be useful to have an estimator for the timeout for this specific profile. Notice that is reasonable to have a timeout value “enough” far from the mean value. Is also possible to see that the elbow of the curve in near the

80 us that is the mean minus the variance. Is reasonable have this result because having a timeout very close to the average leads to loss of power.

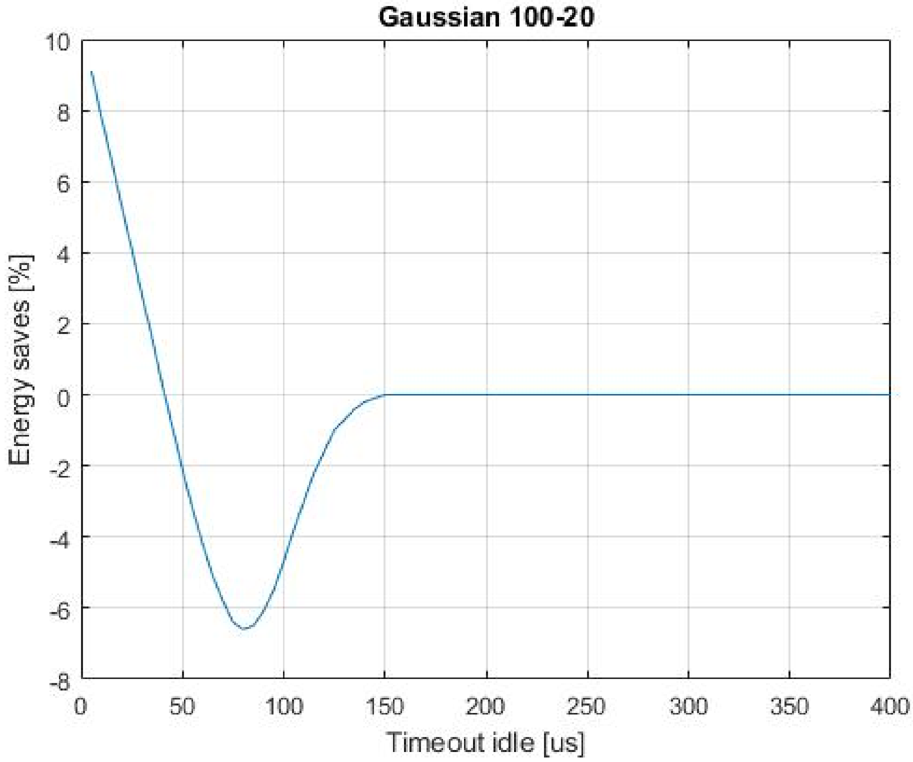



**Figure 9 - Energy saving w.r. idle Gaussian**

Figure 10 represents the same distribution considering the sleep state. Is possible to see that for the given distribution there are not significantly improvement on energy saving when reaching the

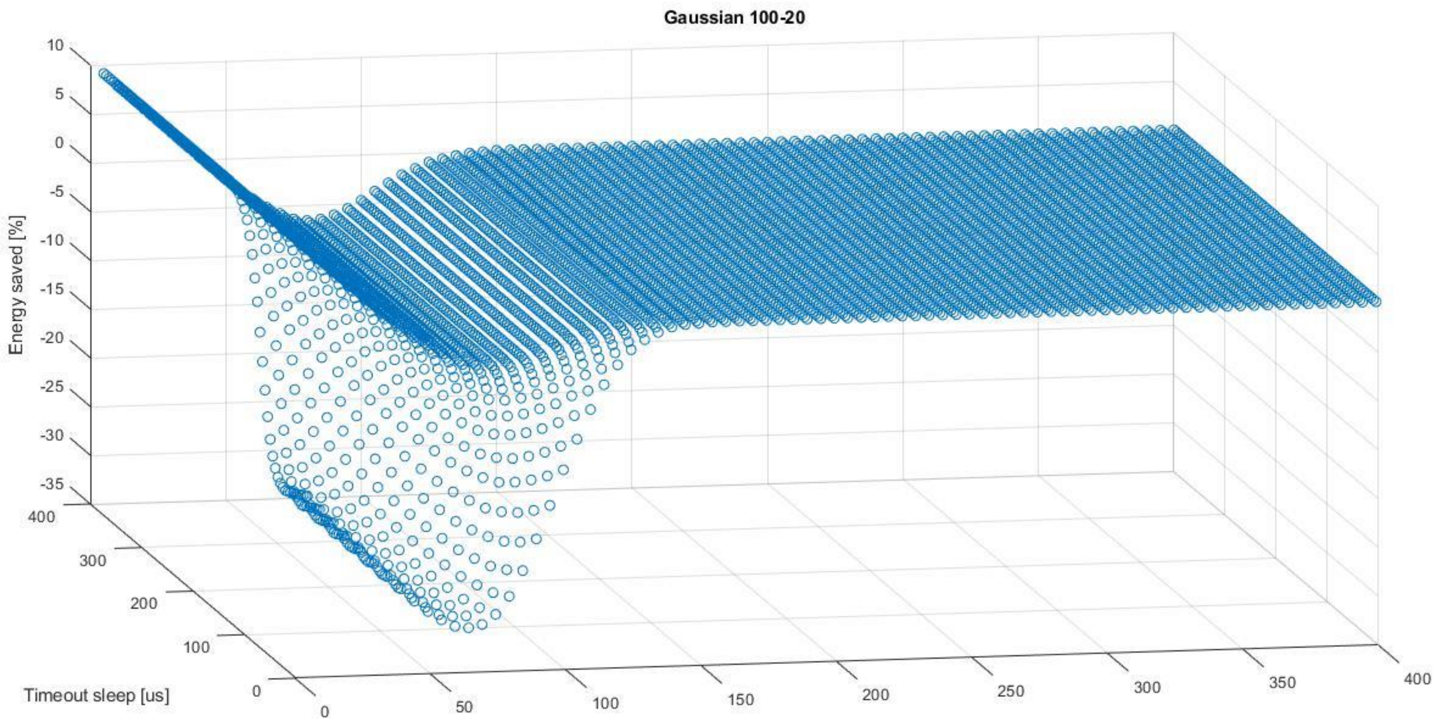



**Figure 10 - Energy saving w.r.idle and sleep Gaussian**

maximum values. However, there are many improved wastes of energy when reaching the minimum of the function.

## Trimodal

The trimodal energy saving follows the energy saving of a Gaussian. Indeed, in figure 11 curve there are three local minima that are related to each distribution mean (50, 100, and 150).

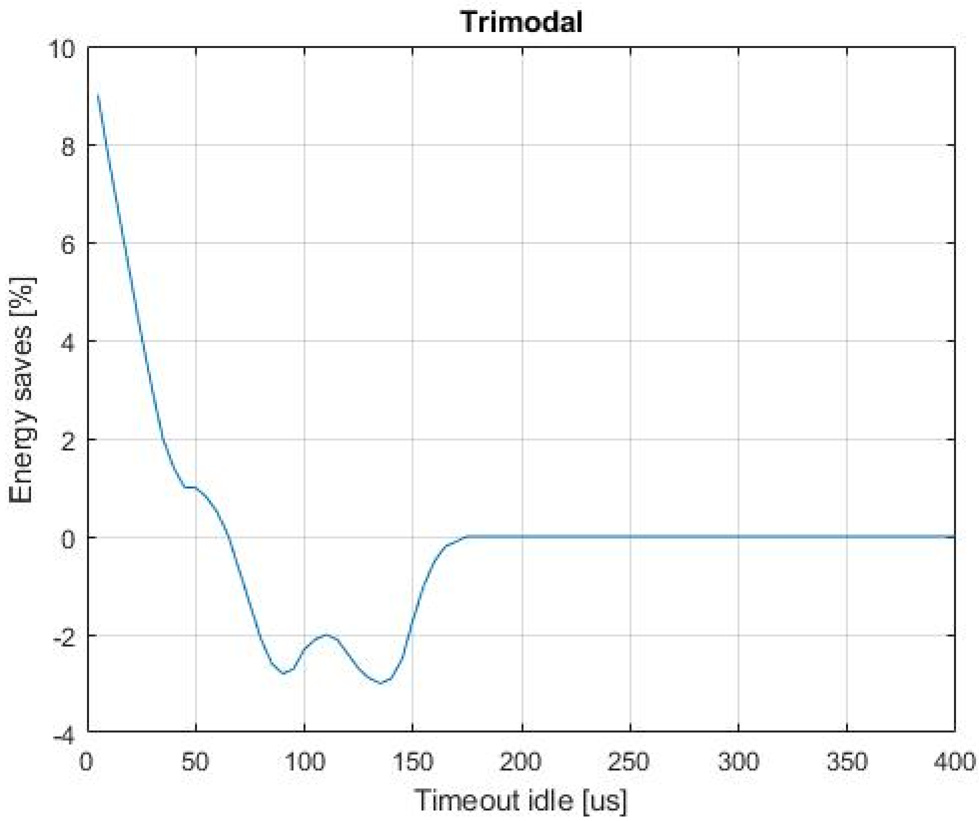



**Figure 11 - Energy saving w.r. idle trimodal**

Figure 12 shows the timeout policy considering also the sleep state. This graph follows the shape of the idle case but is possible to notice that in the local minima the energy wasting is much more.

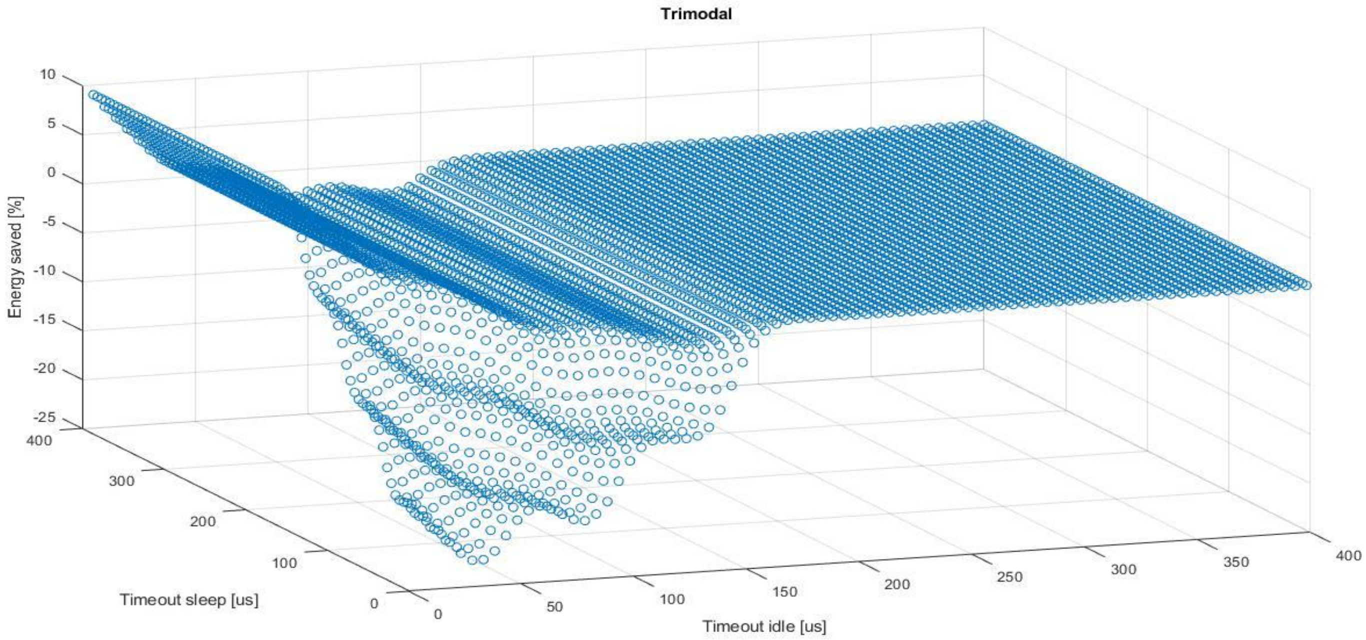



**Figure 12 - Energy saving w.r.t. idle and sleep trimodal**

# Additional test

Looking at the result is possible to notice that among the generated profiles, anyone have a real improvement of energy saving when considering both idle and sleep state. The conjecture was that the given distribution does not generate idle time enough large to make the use of sleep state more convenient than idle. An additional profile has been generated using a Gaussian distribution mean 400 and variance 25.

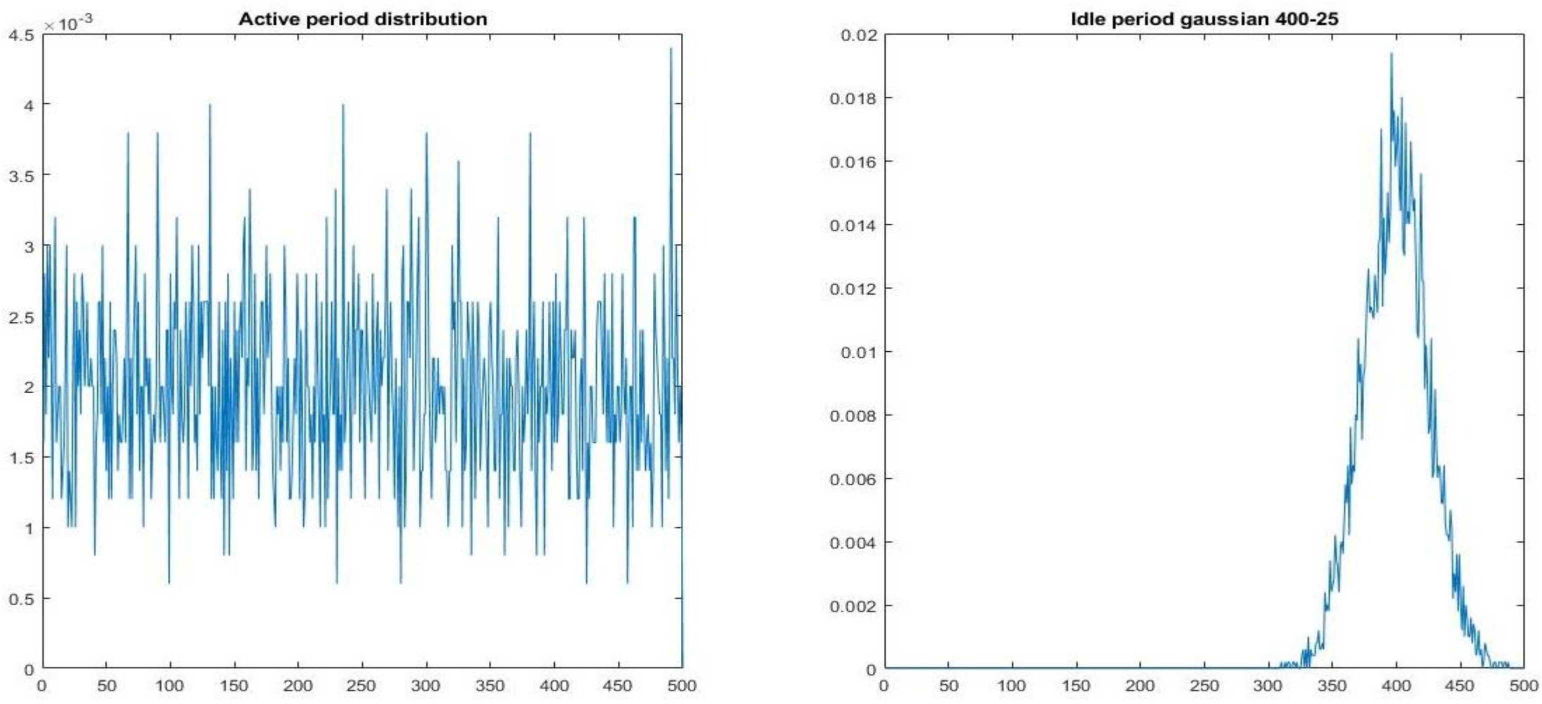



**Figure 13 - Gaussian low utilization profile**

Running the script using this distribution, the result is reported in figure 14. The energy saved in the first point near origin is 37,4% instead the value without sleep is around 45%.

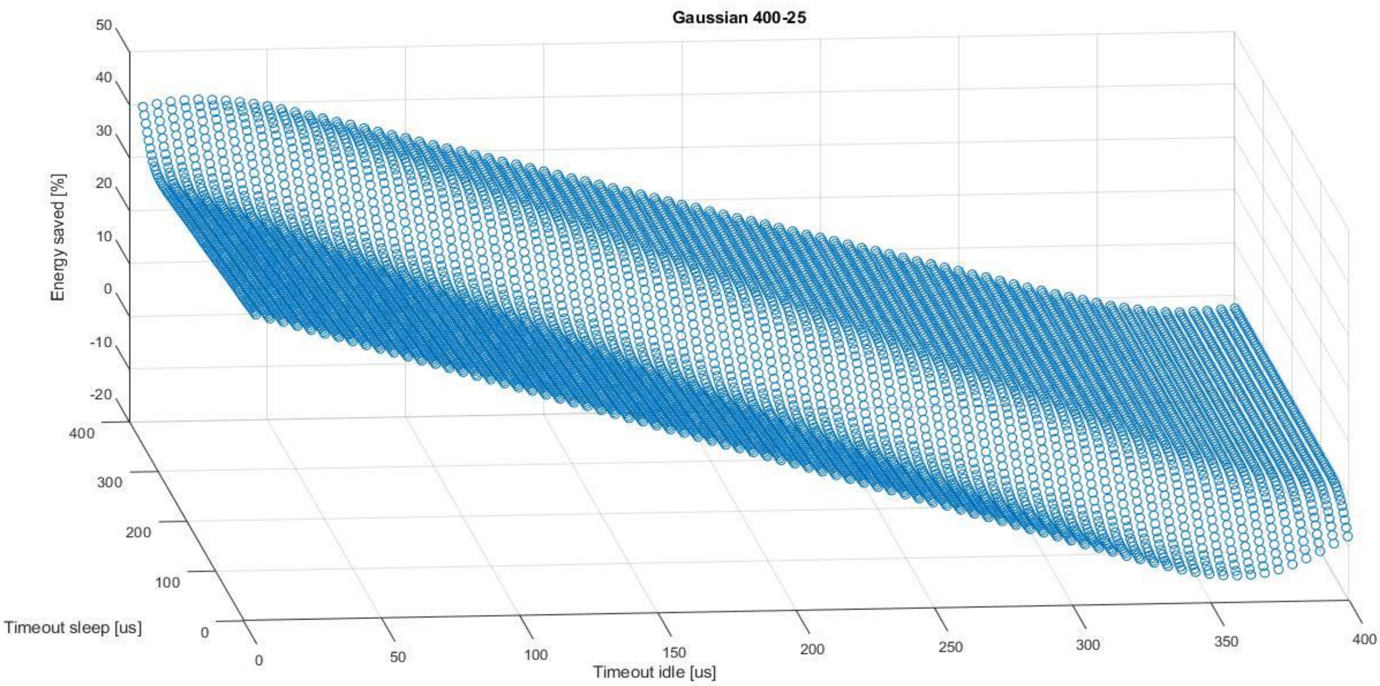



**Figure 14 - Gaussian low utilization energy saving**

So even using a much more convenient workload distribution the sleep state is less convenient than the idle. This could be given by the fact that the transition energy of from idle to sleep (passing through active) is higher than active energy. Figure 15 reports the same workload profile of figure 13 but using a more convenient PSM. The result is that the maximum value of energy saved is the closer to the origin, 47.7 %. Meanwhile the max saving considering only idle is 45.5%.

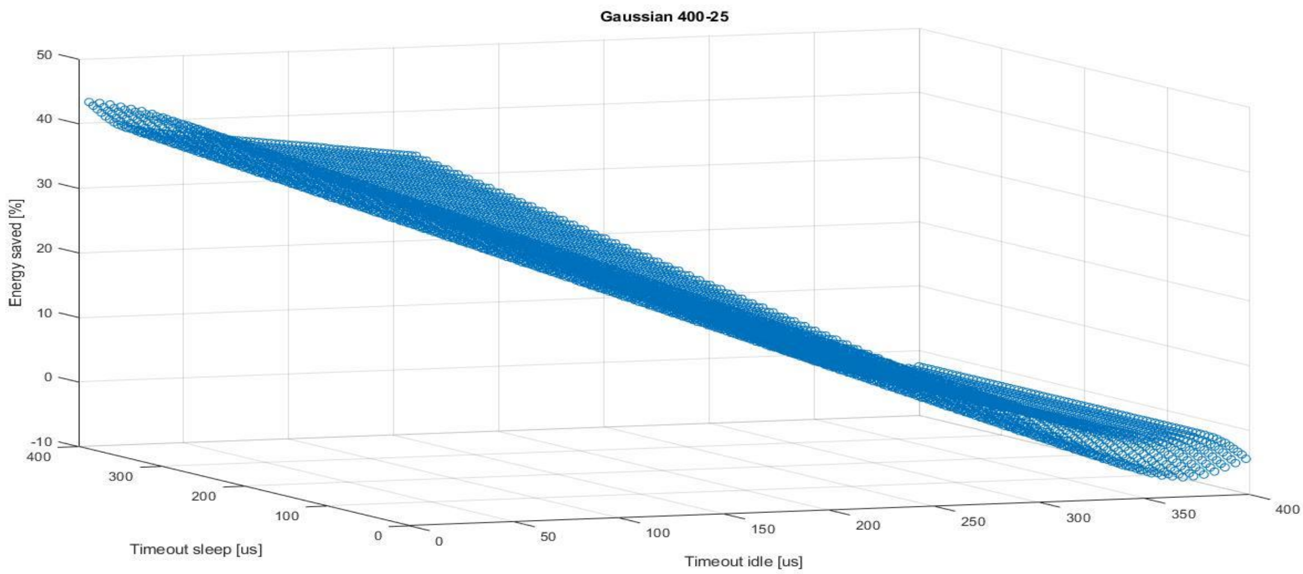



**Figure 15 - Gaussian low utilization and eco PSM**

## Custom workloads

For the two given custom workload the same analysis has been performed. Figure 16 reports the results for the custom workload 1. In particular is possible to notice that the idle curve seems the gaussian one. The idle-sleep plain there is a fold in the surface between sleep time 150 – 250. The maximum energy saving is 33.3 %.

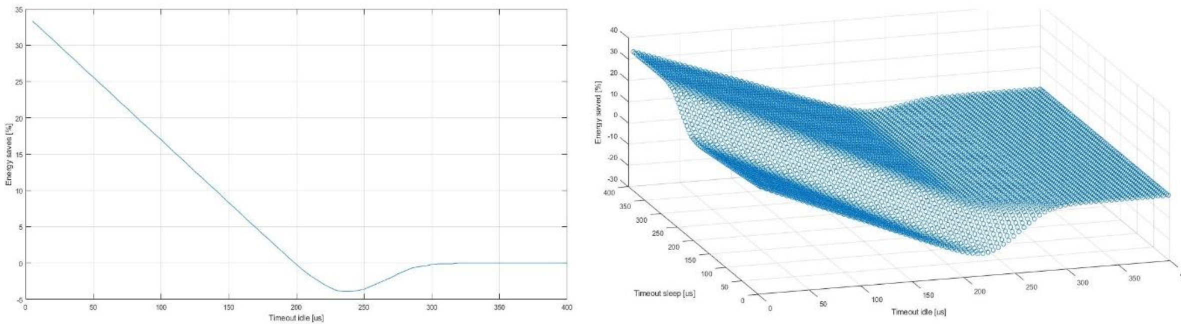


**Figure 16 - Custom workload 1**

The custom workload 2 is reported in figure below. Is possible to see that in this case for idle there is less energy saving, around 26%. Meanwhile the idle-sleep surface seems more regular than the previous.

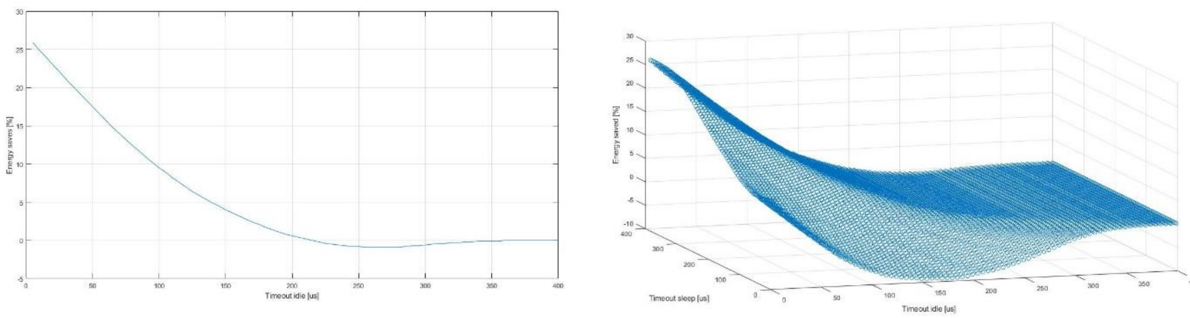


**Figure 17 - Custom workload 2**

## Timing and energy overhead

Beyond the energy saved should be considered also the time and energy wasted when transitioning from a state to another. In the particular case of non-extended idle timeout, the time and energy wasting are exactly the same numbers, this is due to the PSM. For this reason, will be carried a single description for both overheads. The figure below reports all the energy overhead (same for time overhead) changing the timeout value. Is important to notice that exponential and uniform 1-100 distribution have the lowest curves although has been shown that have the worst energy saving. This is given by the fact that the energy saving is not strongly characterized by the energy overhead rather than the amount of time spend in idle state. Is also interesting to notice that these curves tend toward zero by the increasing of timeout. The minimum value of timeout corresponds to the maximum of all curves, this is due to the fact that low timeout led to a high number of transitions.

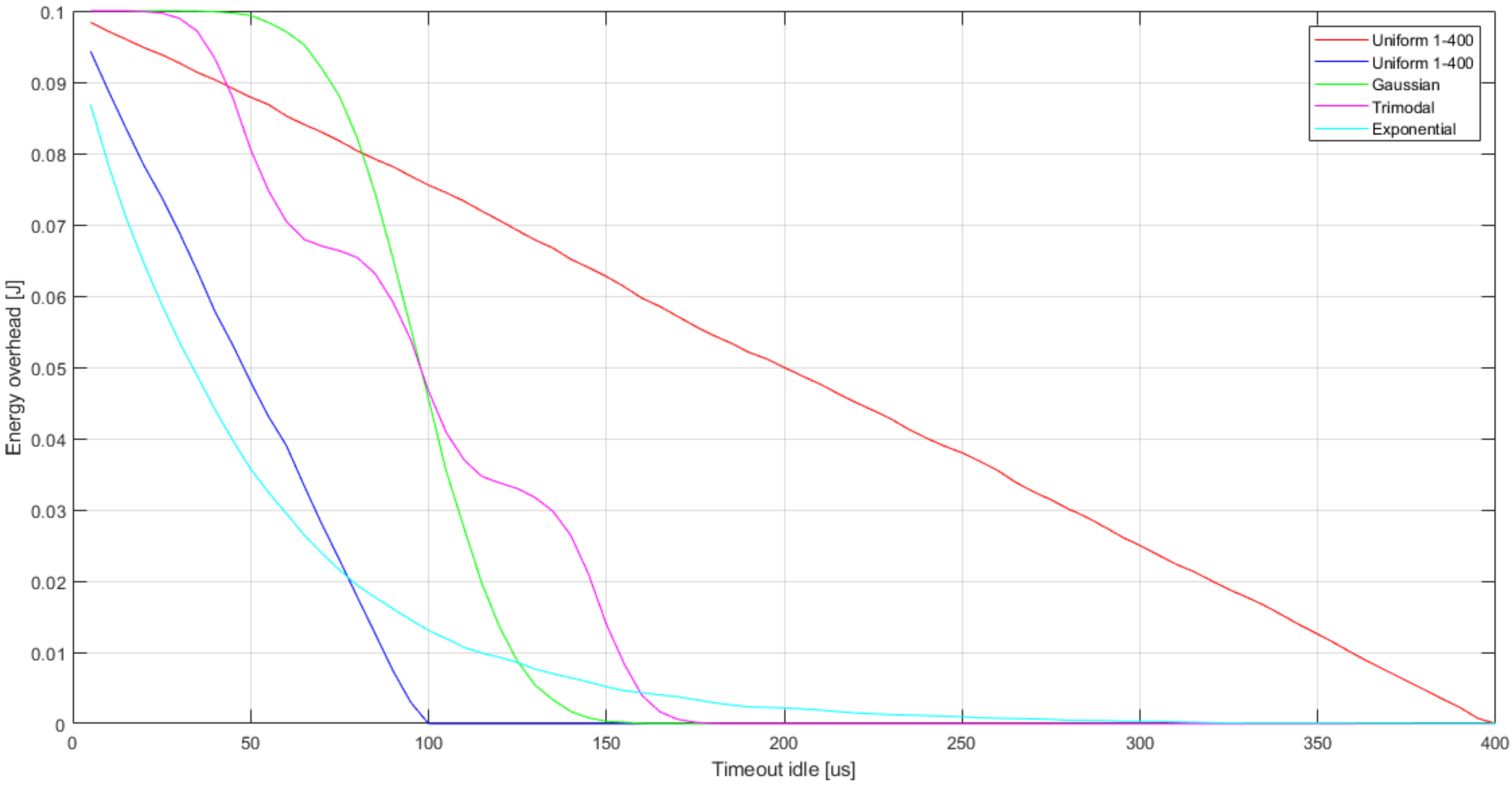



**Figure 18 - Energy overhead for the profiles shown**

When extend to sleep state the maximum of time and energy overhead will be higher due to the addition of sleep overheads. Figure 19 reports the energy wasted of the two uniform distributions. Is possible to see that for low values of sleep timeout the shape tends to add and overhead to the previous 0.1 J, while the sleep tends to grows the shapes reconverges to the one in figure 18. Same situation happened for the time overhead, but in a sharper manner due to the fact that sleep state has huge amount of transition time. Same situation happens for the other profiles.

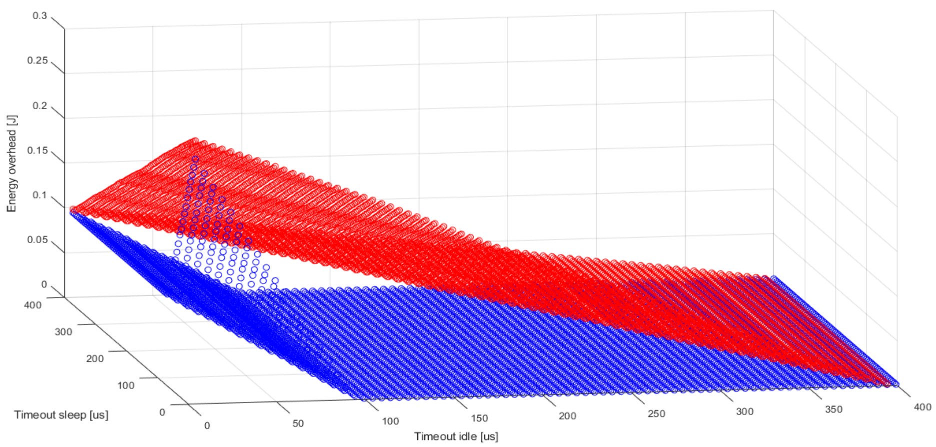



**Figure 19 – Extended timeout energy overhead, uniform distributions**

# HISTORY POLICY

This policy considers the past of the workload in order to have a prediction on the future idle values. Respect to the timeout policy here there is the advantage that the decision is taken immediately on the start time of the idle period. In order to predict the value of the incoming idle period the simulator implements an algorithm of regression. Could be possible to choose among several regression models. For this lab has been used the polynomial model:

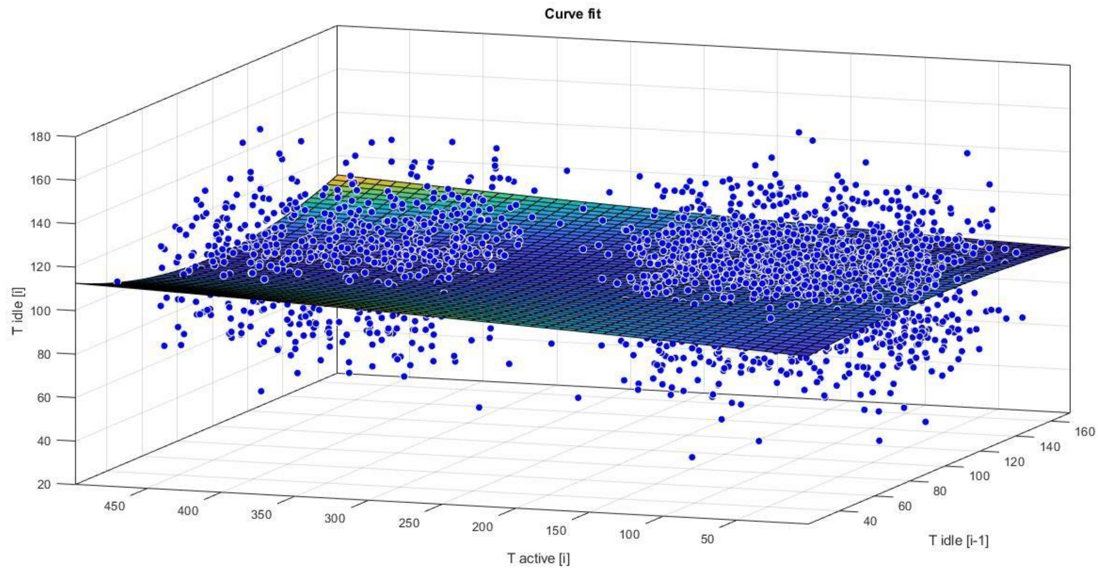
Eq.1 $f(x, y) = p0 + p1 \cdot x + p2 \cdot y + p3 \cdot x \cdot y + p4 \cdot x * y^2 + p5 \cdot x^2 \cdot y + p6 \cdot x^2 + p7 \cdot y^2$

Where x represents the past value of idle period (T idle [i-1]) and y represents the last active period (T active[i]). The result will approximate the value of the following idle period (T idle[i]).

Obviously, implement this kind of algorithm inside an IoT device will increase significantly the predictor computation time. This polynomial was taken only has starting point, indeed is able to correctly fit separated clouds of point in the tri-dimensional space. Once the distribution is generated, the 50% of the point will be used to train the coefficient of the polynomial. The coefficients going toward zero will be dropped and consequentially the model will be re–built. This way is easy to find both coefficient and model that fit the distribution. If for example a given distribution is independent from Tactive, all the coefficients multiplying this variable will tend toward zero. Dropping these coefficients will lead to a new model that present only the remaining variable Tidle.

Considering the distribution described in the first chapter, will be evident that are not suitable for this kind of application. Indeed, due to the independency among the samples, all the coefficients will be cleared and the remaining model will be a horizontal plain inside the sample cloud. The only distribution suitable for this policy is the trimodal that has been generated taking one sample for each Gaussian until the required samples are generated, this creates slightly correlation among the T idle values. To test this policy other workloads has been generated with a relevant correlation.

In order to implement the given formulation, the simulator has been improved to accept eight (p0 to p7) parameters as alpha. In case the model drops some coefficients, they will be inserted as zero in the simulator inputs. In order to train fit the coefficient “HistoryWorkload.m” script has been written.




Coefficients =

```
94.9846    0.1224    0.0666   -0.0014    0.0000    0.0000   -0.0007   -0.0000
```

**Figure 20 - Gaussian independent curve fitting. As reported many coefficients tends toward zero and the model can be approximated with a horizontal plain, saving computational time.**

## Independent distributions

Using the correlation coefficients has been proved that all distributions described in the introduction are not correlated, except for the trimodal one, described in a dedicated section. This means that a generic T idle[i] is independent from both T active [i] and T idle [i-1]. The model used for all these distributions is:

$$\text{Eq.2 } f(x, y) = p0 + p1 \cdot x + p2 \cdot y$$

The report of the all distribution is then reported in the following table:

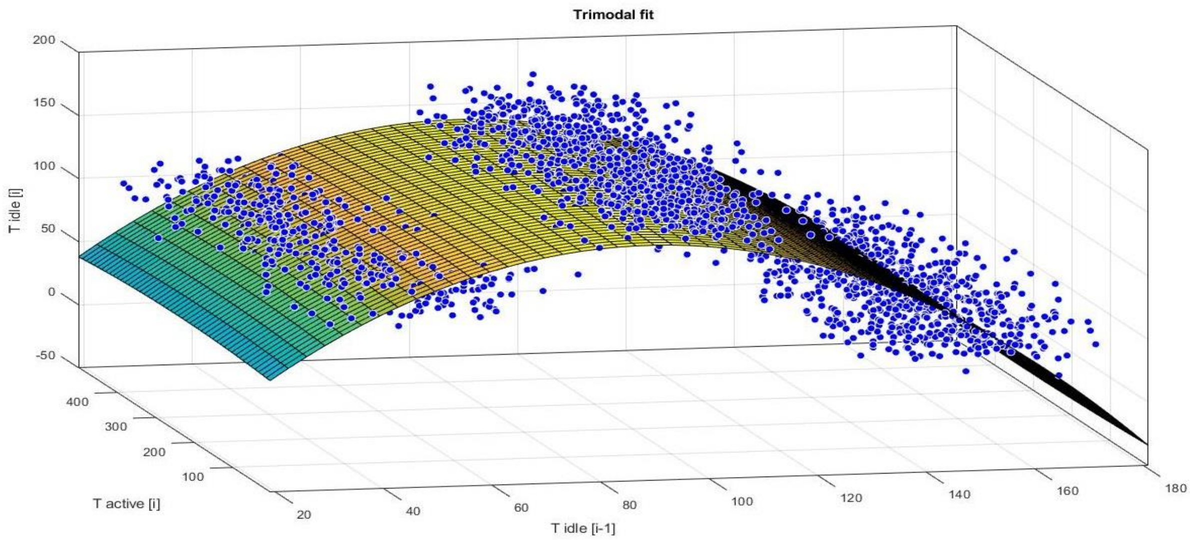
**Table 1 - History policy for independent distributions**

| Distribution | Energy saved | Coefficients (p0, p1, p2) |
|---|---|---|
| Uniform 1-100 | 0% | 49.4687 -0.0103 0.0034 |
| Uniform 1-400 | 34,20% | 195.8942 0.0151 0.0116 |
| Gaussian | 10,70% | 98.1137 0.0109 0.0041 |
| Exponential | 0% | 49.3037 0.0020 0.0007 |

The energy saving is convenient only for the gaussian and the unform 1-400, the others does not enter either in the idle or sleep state. This is due to the fact that considering the two separated break-even time (computed internally by the simulator) there is not convenience in transition.

## Trimodal distribution

Has said the trimodal distribution creates a correlation among the idle times, so can be interesting from a history point of view, try to fit this distribution. Using the profile explained in the Introduction chapter and the eq.1 model, the cloud points are fitted (figure 19).



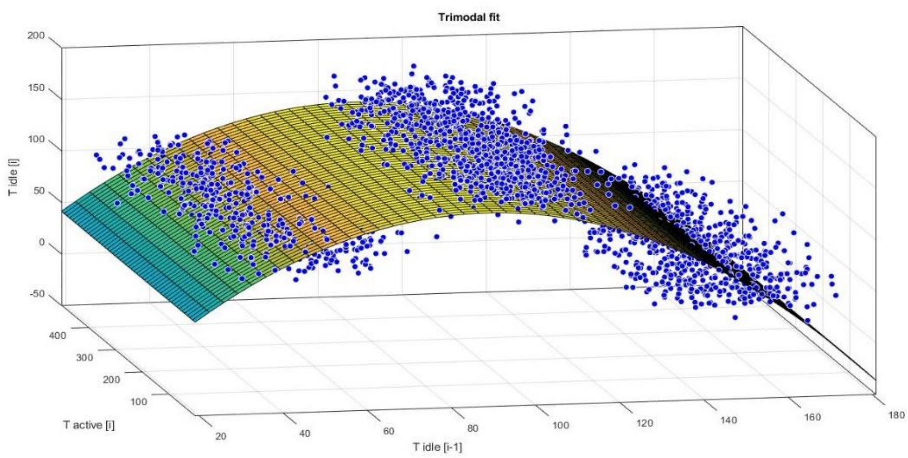

```
coefficents =

  -22.1658    3.5709    0.0382   -0.0002    0.0000   -0.0000   -0.0201   -0.0001
```

**Figure 21 - Trimodal fit**

As highlighted by the figure the distribution is fitted but many coefficients tend toward zero. So, coefficients *p3, p4, p5, p7* are dropped. The resulting model is:

$$\text{Eq.3 } f(x, y) = p0 + p1 \cdot x + p2 \cdot y + p3 \cdot x^2$$

Fitting again the train dataset using the new model (Eq.3) we have the following result.




```
coefficients =

  -19.4591    3.5721    0.0006   -0.0203
```

**Figure 22 - Optimized trimodal fit**

As possible to see, the result still fits correctly the distribution but the computation time is improved using less coefficients. Using these parameters, the simulator will save 10, 9% of energy.

## Periodical workload

In order to create a correlated workload, the idea was to use a practical use case. So, workload is generated considering a first profile that have low active periods and long idles. This profile is repeated until the total time reaches a given value T that wakes another profile characterized by high utilization and low idle period. This last profile appears once at period T. This can be view as a sensor workload that periodically stores values and once at T time it elaborates all the samples.

First characterization done was considering a period T equal 800 us, thus having low global utilization. The distributions are in figure 21.

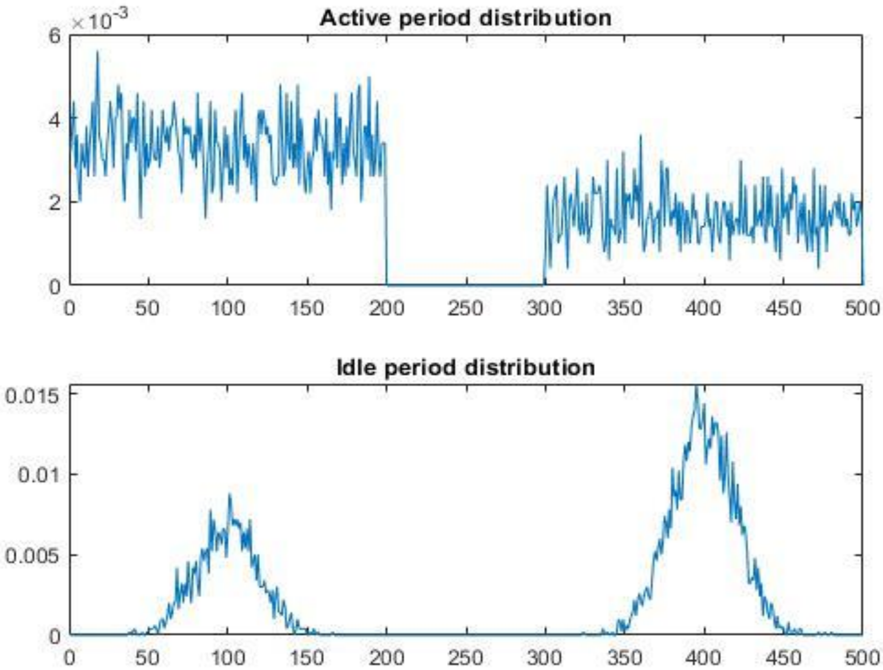
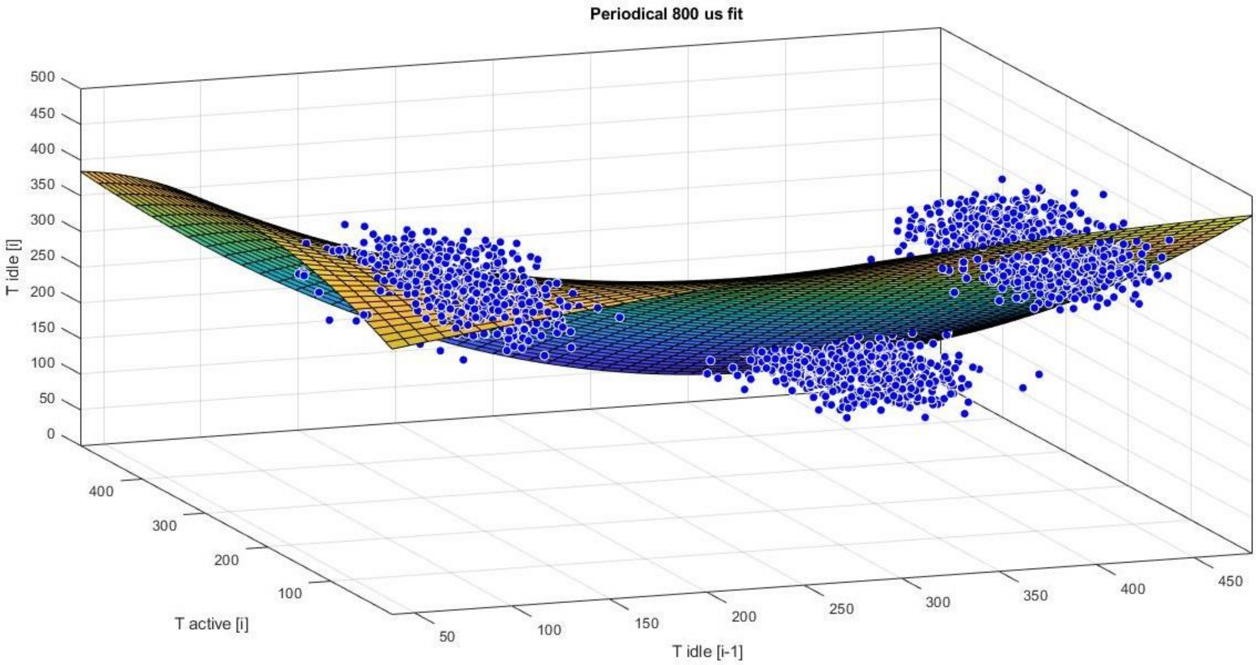



**Figure 23 - Periodical profile**

The approach used is the same explained in the introduction. Figure 22 reports the full model coefficients.

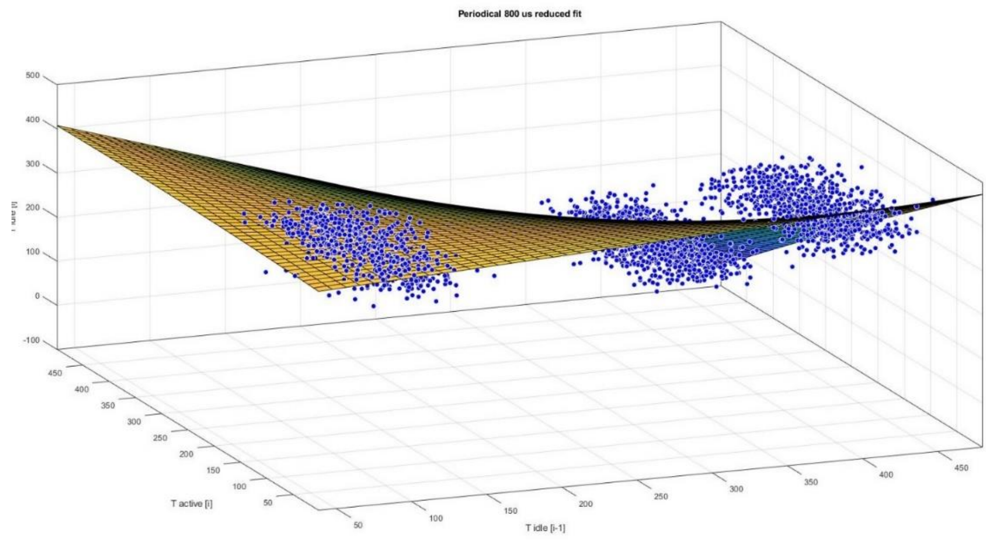



coefficients =

356.7551    0.4141    0.7657    -0.0068    0.0000    0.0000    -0.0004    -0.0011

**Figure 24 - Period 800 us fit**

Dropping the coefficients *p4, p5, p6* the model result is reported in figure 23.




coefficients =

390.0292    0.1749    0.1600    -0.0024    -0.0001

**Figure 25 - Reduced 800 us periodical fit**

Using these values, the energy saved is 42.4 %.

## Coefficient analysis

Until now the approach followed has been to start with eq.1 model and drop the small coefficients in order to optimize the computing overhead, this approach seems reasonable in order to select the coefficient to drop. Another approach can be to start from the complete polynomial and drop the coefficients in order. The result for the two correlated distributions (Trimodal and periodical) will be discussed below.

The table below reports the values of energy saving, time and energy overhead for the trimodal profile.

**Table 2 - Coefficients sweep for trimodal**

| | Model order | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | p0 | p0 p1 | p0 p1 p2 | p0 - p3 | p0 - p4 | p0 - p5 | p0 - p6 | p0 - p7 |
| Energy saving | 10,70% | 10,70% | 10,70% | 10,70% | 10,70% | 5,80% | 11,00% | 10,90% |
| Time overhead | 0,049s | 0,049s | 0,049s | 0,049s | 0,049s | 0,021J | 0,039s | 0,04s |
| Energy overhead | 0,049J | 0,049J | 0,049J | 0,049J | 0,049J | 0,021J | 0,039J | 0,04J |

As is possible to see the energy saving is quite flat, except for 6 coefficient case. This result is could depend by the fact that the Tbe threshold for the sleep is too high with respect the 3 gaussian averages, so all the predictions are compared with the idle ones. Due to the fact that idle Tbe is 54us the prediction of first models is almost taken. So, the situation reminds to a timeout-like solution. The increasing of performance can be seen on the last two columns where the models reaches a higher complexity. For this model must also be kept in consideration that the prefile presents only the correlation among Tidle[i-1] and Tidle but there is not with Tactive.

For the periodical profile (period T = 800 us) the same test has been performed and the results are reported on table below.

**Table 3 - Coefficient sweep for periodical**

| | Model order | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | p0 | p0 p1 | p0 p1 p2 | p0 - p3 | p0 - p4 | p0 - p5 | p0 - p6 | p0 - p7 |
| Energy saving | 40% | 41,20% | 42,30% | 42,10% | 42,20% | 39,50% | 33,60% | 39,90% |
| Time overhead | 0,62s | 0,43s | 0,43s | 0,43s | 0,43s | 0,32s | 0,28s | 0,32 s |
| Energy overhead | 0,09J | 0,083J | 0,082J | 0,081J | 0,081J | 0,058J | 0,050J | 0,058J |

In this case there is a huge variability of the energy saving with respect the number of coefficients used. The best results are the 3-coefficient and 4-coefficent models. This profile generates idle times that can used both sleep and idle considering the Tbe values. Has possible to notice the energy saving grows from single coefficient to 3 coefficients width. This is reasonable due to dependence from both Tidle and Tactive. Rising the number of coefficients, the energy saving stops growing, this means that the remaining coefficients lead to more mispredictions. This was also visible on the figure 24, where is highlighted that just first three components are the most important. For what concern the time and energy overhead is possible to notice that the more energy saving grows the more the overheads become bigger. This depends of the number of transitions from the PSM. In case of single constant p0 the overheads are the hugest, this depends by the fact that the coefficients are equal to the average of the training data Tidle, that is greater than Tbe sleep. So, all the transitions are always taken.

## Window analysis

From the two tables is possible to study also the dependency of the energy saving with respect the window size. The table resumes the energy results for the different profiles.

**Table 4 - Window energy saving**

| | p0 | p0 + p1*Tidle[i-1] | p0 + p1*Tidle[i-1] + p2*Tactive[i] |
|---|---|---|---|
| Uniform 1-100 | 0% | 0% | 0% |
| Uniform 1-400 | 34,20% | 34,20% | 34,20% |
| Gaussian | 10,70% | 10,70% | 10,70% |
| Exponential | 0% | 0% | 0% |
| Trimodal | 10,70% | 10,70% | 10,70% |
| Periodical | 40% | 41,20% | 42,30% |

As possible to notice there is no significant improvement for all profiles except for the periodical one that has a dependence on the window.

## Policy comparison

Comparing the history policy result with the timeout ones, same consideration can be done. As possible to see from table 1 the independent distributions reach the maximum value of timeout energy saving using the history policy. Is important to underline that the history regression of independent profiles tends to be the average of the profile itself. Using the break-even time, the simulator decides immediately the next state. For what concern the correlated distribution the history solution is much more interesting, indeed for the trimodal there is an improvement of energy saving around 0,02% (considering the energy result for timeout equal 0). For the periodical there is an improvement of 0,01%, considering the energy result for timeout equal 0. Must be kept in consideration that the history algorithm requires some computational time that is not considered in the current simulator.

## Time and energy overhead

The time and energy overhead for the timeout policy were increasing when rising the energy saving. The more energy saving the more transition were kept. For the history policy the scenario is the opposite, indeed if the quality of the predictor is good, the number of transitions is limited to the convenient ones. As possible to see from table 2 for the trimodal the best energy saving situation is also the one that presents the lower time and transition energy. Same append for the periodical on table 3 the first three columns points the growing of energy saving and the decreasing of overheads. The overhead time and energy values between predictive and timeout policies are not comparable, due to different length of the workloads. The predictive policy divide the generated workload in 50% test and 50% train, so the predictive workload will have the half of transitions with respect the timeout one.