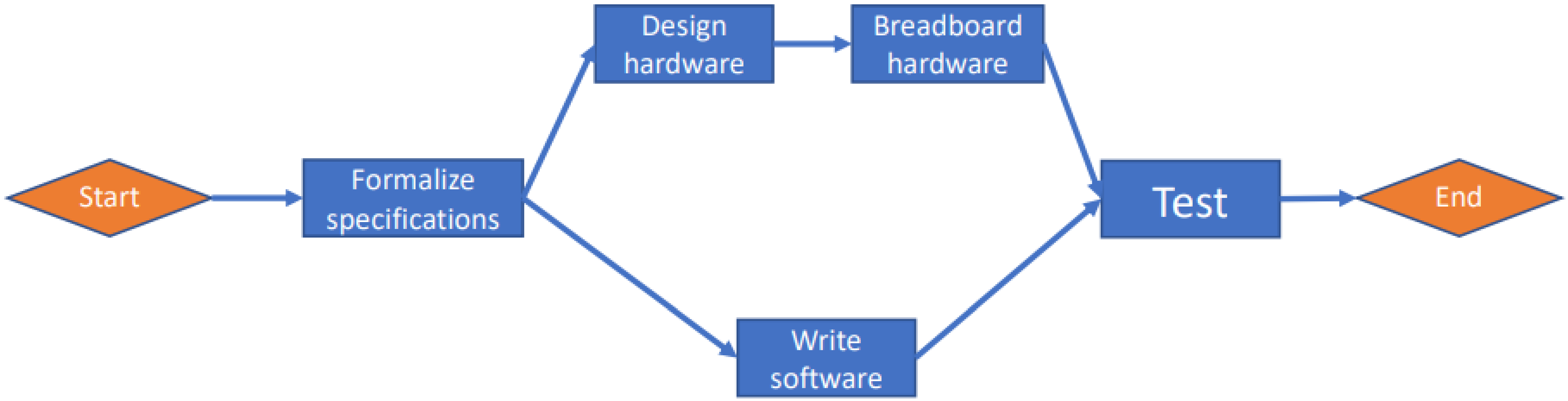
Performance Evaluation and Applications 2022 / 2023

Project Type C

Embedded system development cycle

Goal of the project

→ Find the best number of projects N that the company should work on at the same time to produce the best tradeoff between throughput and project completion time.



Characteristics of the model

→ Each department works independently of the others

→ As soon as one department finishes working on its project it starts the next one, if available

→ Branches work in parallel

Fitting

→ By means of MatLab and having at disposal samples of the durations of the corresponding phases I was able to define the characteristics of the various departments, exploiting the method of moments and fitting the traces according to several distributions, such as Uniform, Exponential, Hyper-Exponential, Hypo-Exponential and Erlang.

Then, looking at the coefficient of variation and at the curves in the graph I chose the most appropriate rate for each of the traces

Fitting - Formalize specifications

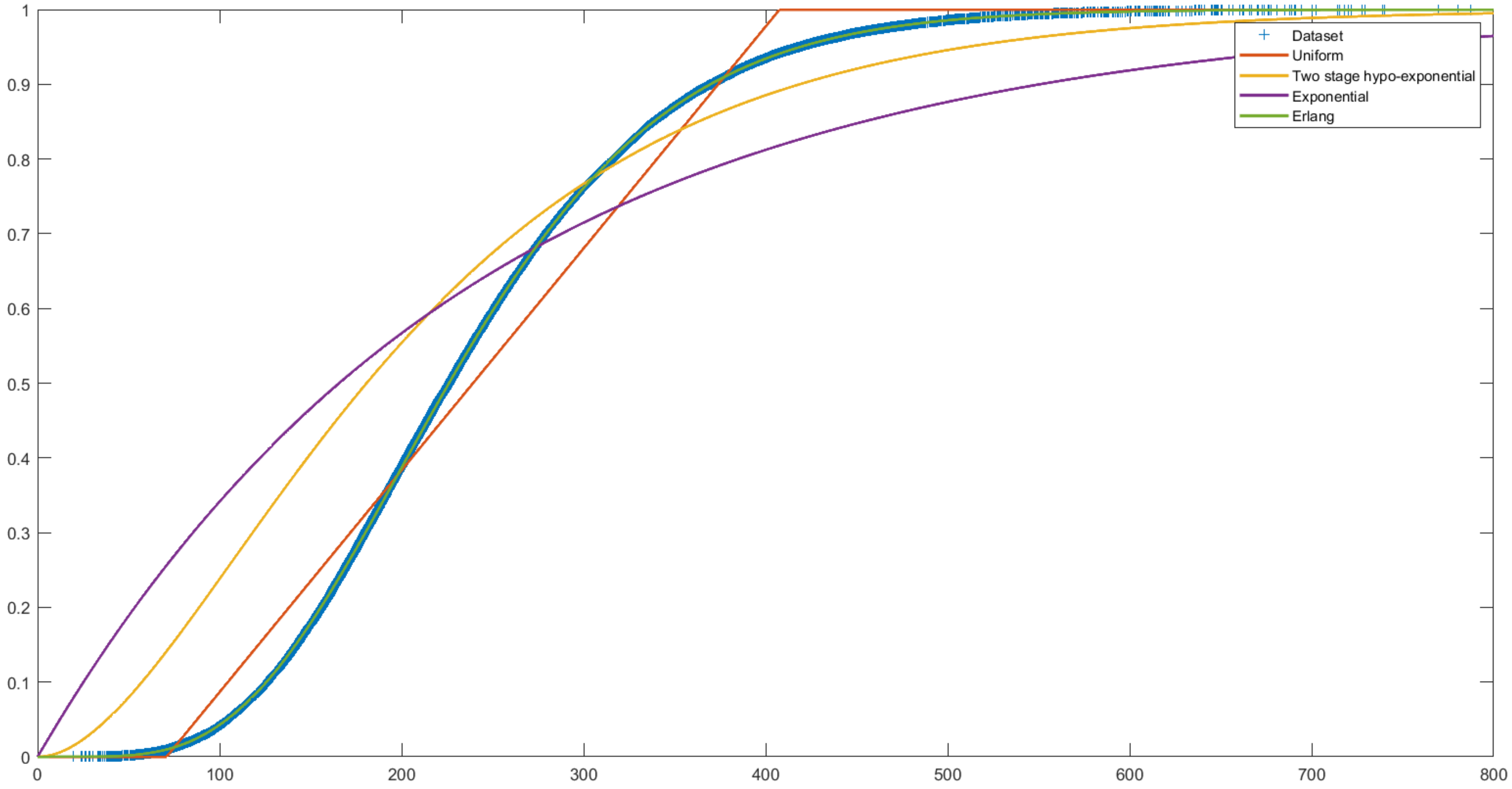
Method of moments is used to determine the best parameters producing samples with characteristics similar to the one measured in the real trace

Moments:

1^{st} moment: 239.169 [h]
2^{nd} moment: 66654.8 [h^2]
3^{rd} moment: 2.11947 E^7 [h^3]
CV: 0.40652

Distribution:

Erlang
k = 6
λ = 0.0250869 [h^{-1}]



Fitting - Design hardware

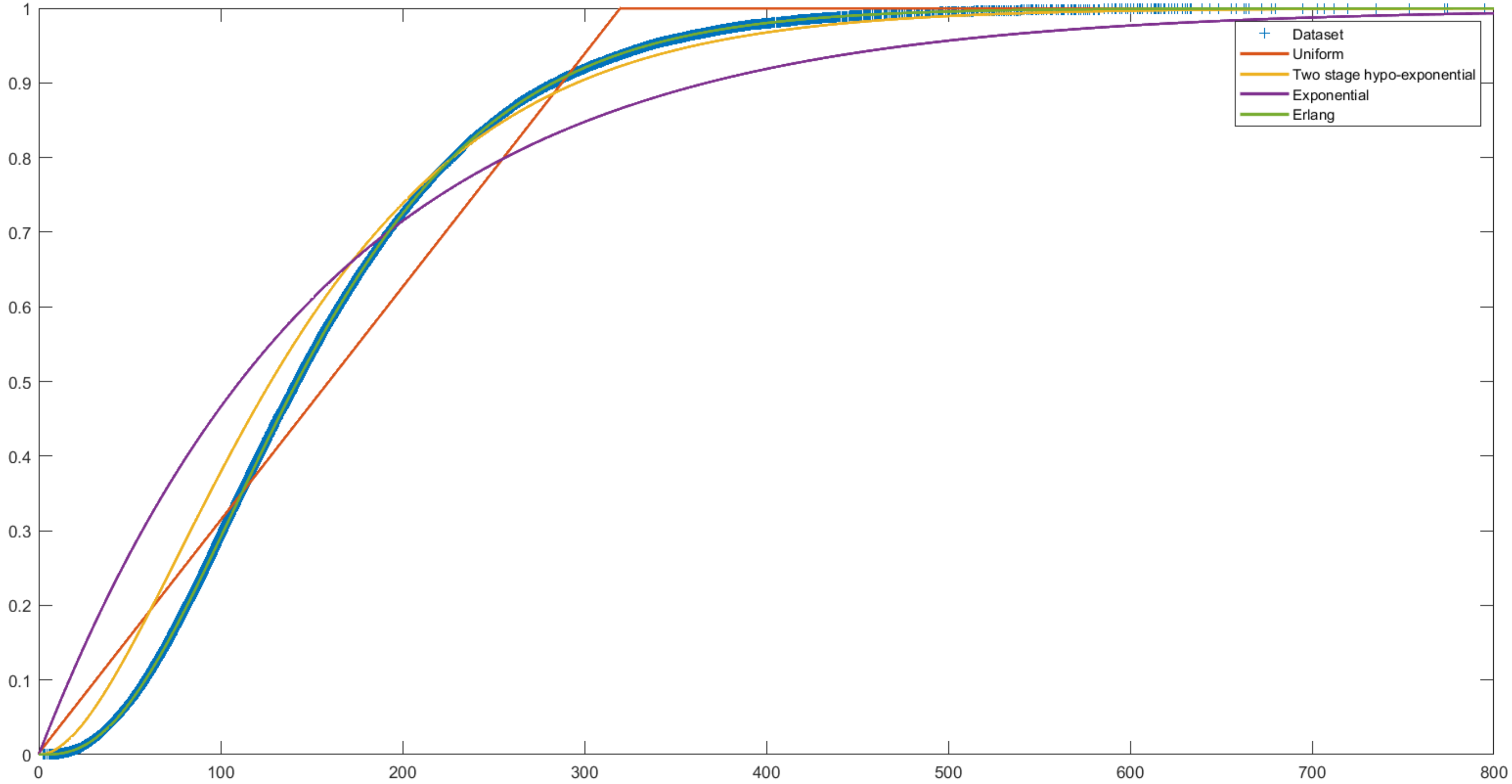
Method of moments is used to determine the best parameters producing samples with characteristics similar to the one measured in the real trace

Moments:

1st moment: 159.448 [h]
2nd moment: 33973.4 [h^2]
3rd moment: 9.06199 E^6 [h^3]
CV: 0.579903

Distribution:

Erlang
k = 3
λ = 0.0188149 [h^{-1}]



Fitting - Breadboard hardware

Method of moments is used to determine the best parameters producing samples with characteristics similar to the one measured in the real trace

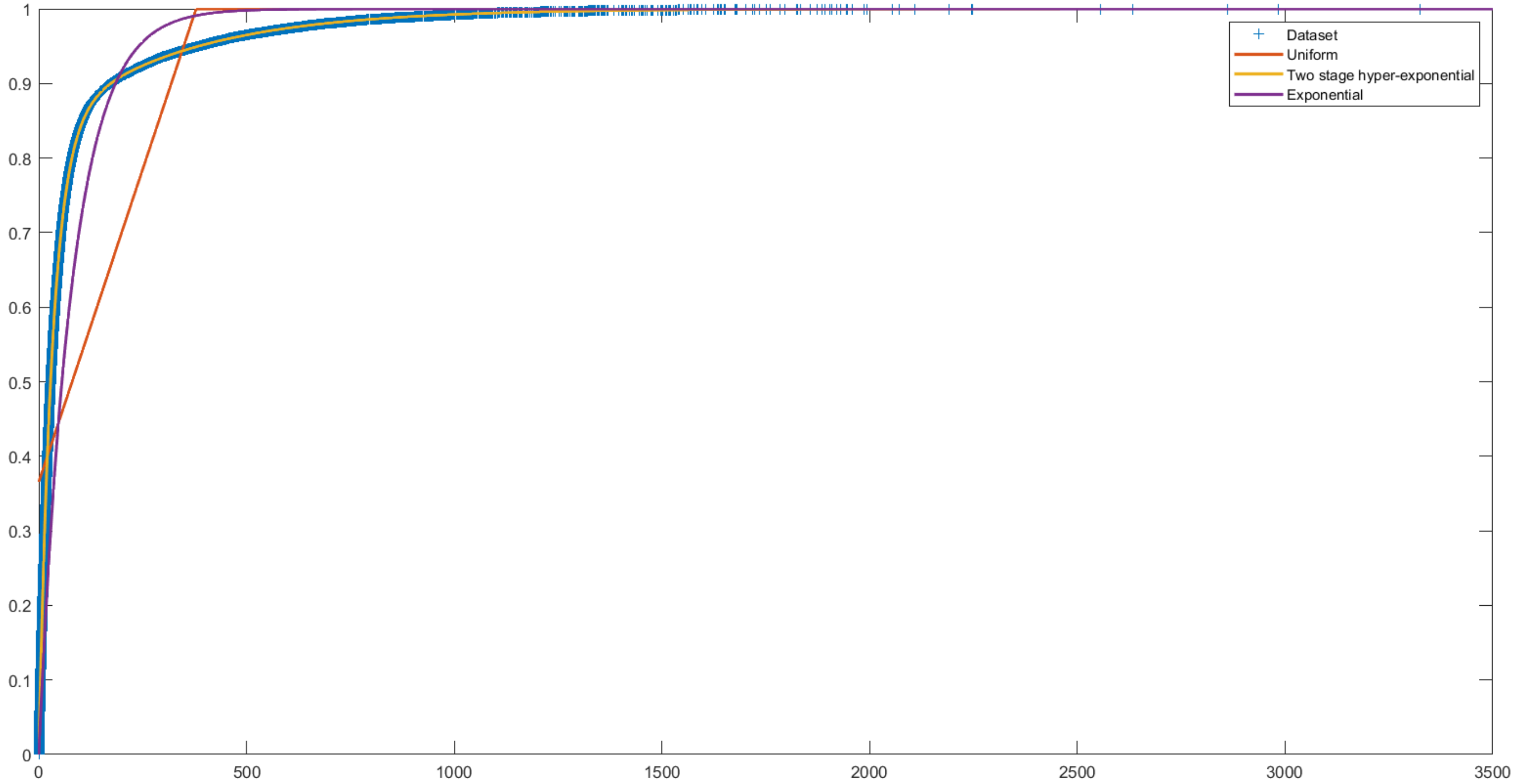
Moments:

1^{st} moment: 80.245 [h]
2^{nd} moment: 36102.6 [h^2]
3^{rd} moment: 3.35095 E^7 [h^3]
CV: 2.14631

Distribution:

2 stages hyper-exponential

$p = 0.163853$
$\lambda_1 = 0.00308955$ [h^{-1}]
$\lambda_2 = 0.0307289$ [h^{-1}]



Fitting - Write software

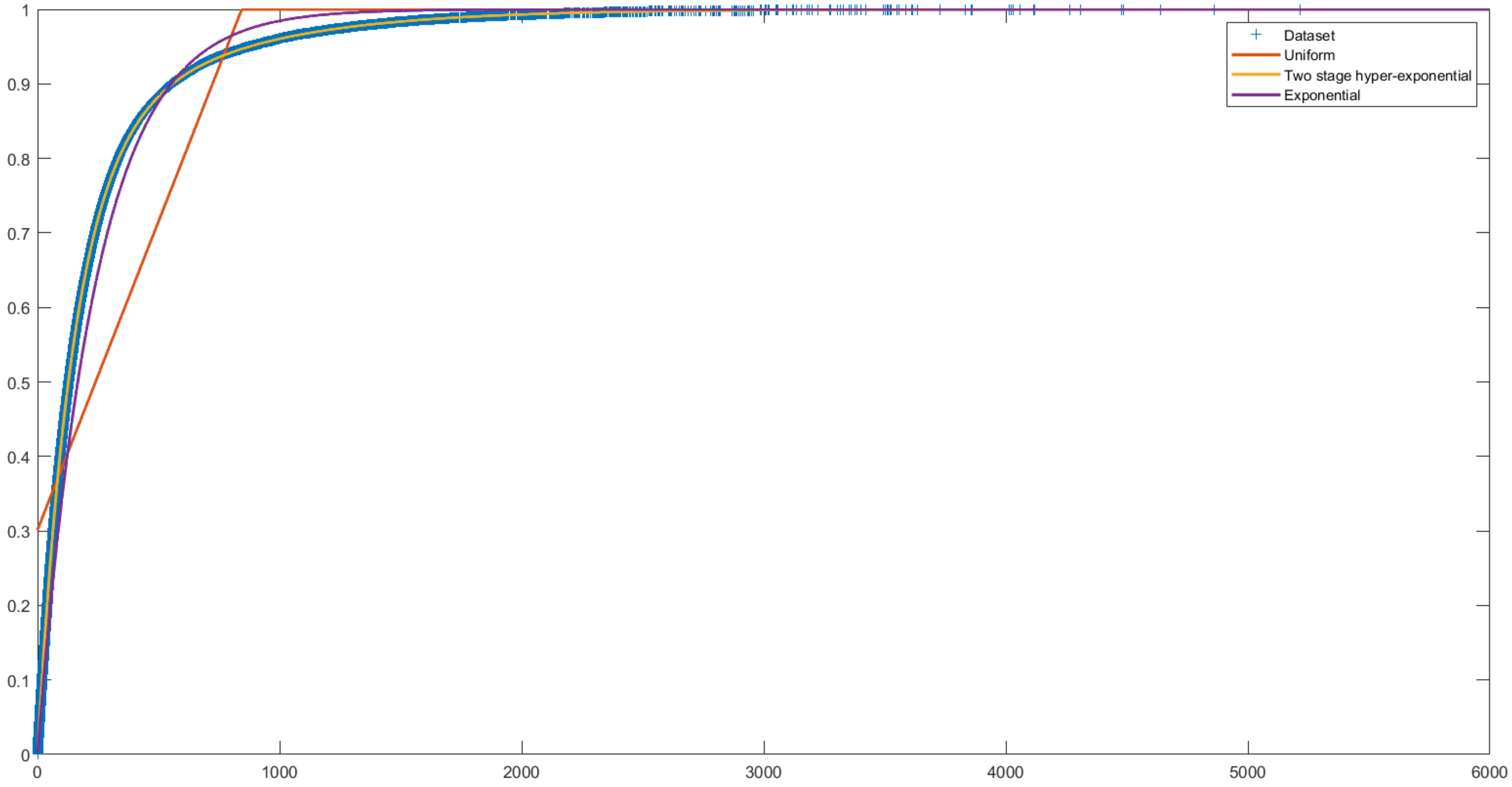
Method of moments is used to determine the best parameters producing samples with characteristics similar to the one measured in the real trace

Moments:

1st moment: 239.762 [h]
2nd moment: 179038 [h^2]
3rd moment: 2.67714 E^8 [h^3]
CV: 1.45413

Distribution:

2 stages hyper-exponential
p = 0.218902
λ_1 = 0.00173056 [h^{-1}]
λ_2 = 0.00689595 [h^{-1}]



Fitting - Test

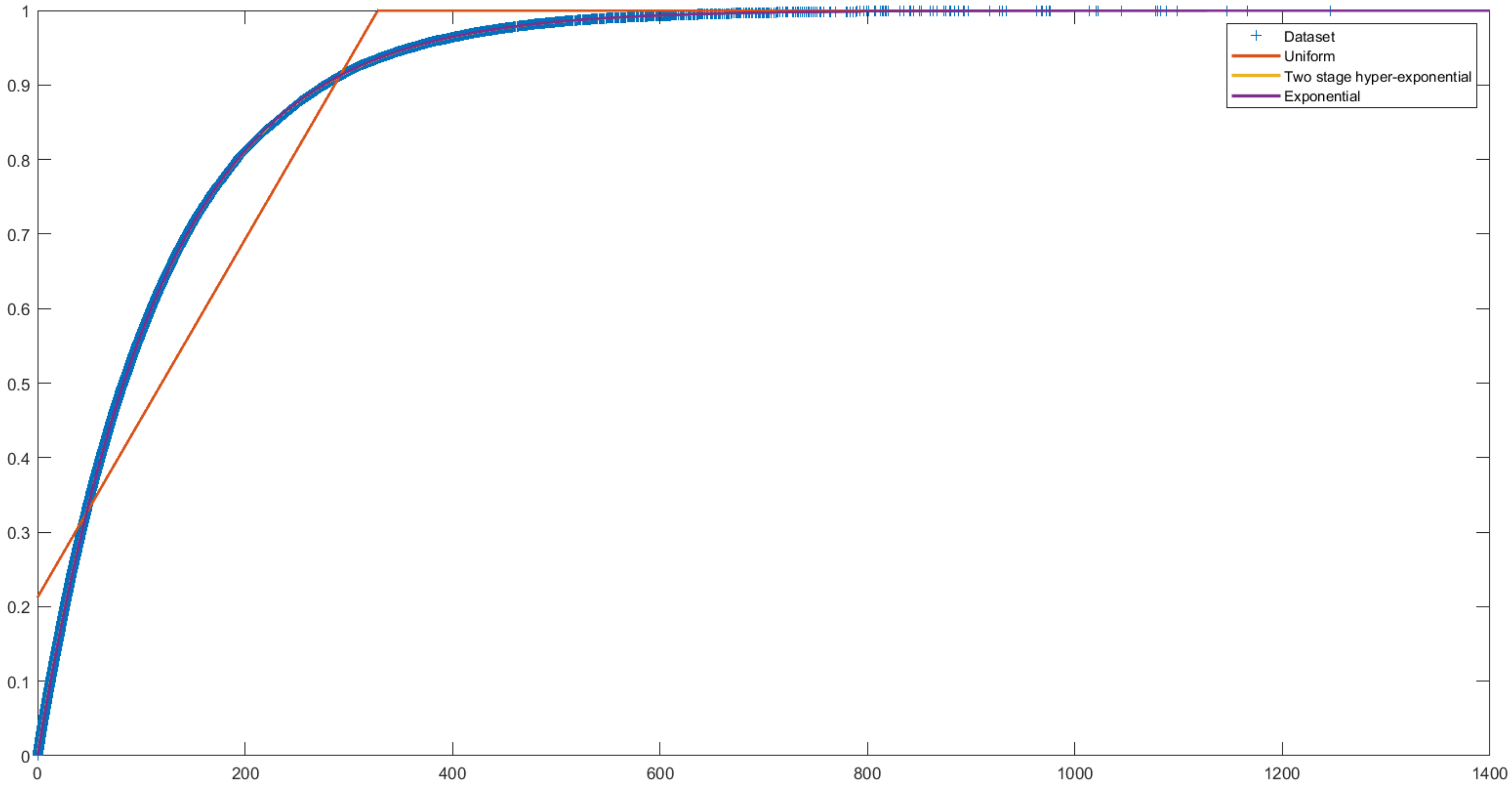
Method of moments is used to determine the best parameters producing samples with characteristics similar to the one measured in the real trace

Moments:

1st moment: 119.681 [h]
2nd moment: 28781.4 [h^2]
3rd moment: 1.03515 E^7 [h^3]
CV: 1.00467

Distribution:

Exponential
λ = 0.00835552 [h^{-1}]



Model definition

→ Using JSIMgraph of JMT, I implemented the case under examination as a closed model defining the service time distribution of the stations according to the parameters found in the step before

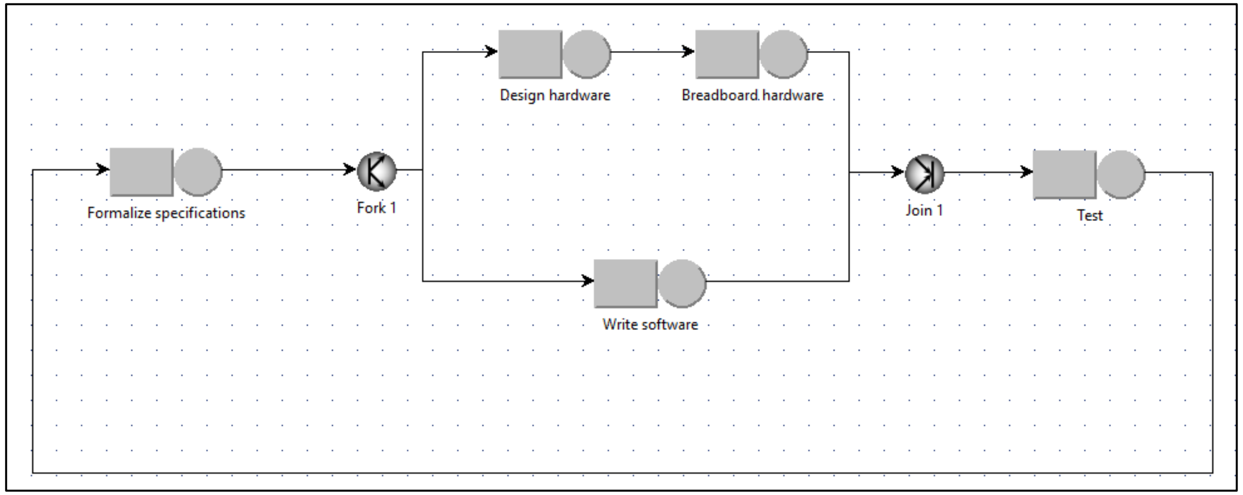
I modeled the departments as queue stations and for the queue capacity I decided to set them as infinite and then to use a FCFS policy and non-preemptive scheduling

The class of jobs in the system has as reference station the *Formalize Specification* one and starts the simulation with an initial population of 2 jobs thus to work on them at the same time

To make the branches work in parallel and then to start the last phase only when the two of them have finished, I made use of Fork and Join components, with one task generated for each input job

In order to obtain the simulation results I performed what-if analysis on the number of projects in the system, up to a population of 50 jobs

Model definition



- → Closed model (always the same number of jobs in the system)
- → Single class (all jobs are equal)
- → Queue stations with ∞ capacity
- → FCFS policy
- → Non-preemptive scheduling
- → Initial population $N_i = 2$

Performance indices analyzed

→ System Power: optimal operational point of a system, it’s the point corresponding to the maximum System Throughput with the minimum System Response Time; useful to find the best tradeoff

→ System Throughput: rate at which jobs depart from the system

→ System Response Time: average time a job spends in the system in order to receive service from the various stations it visits

→ Utilization: percentage of time a station is used, evaluated over all the simulation run

→ Queue Time: average time spent by the jobs waiting in a station queue

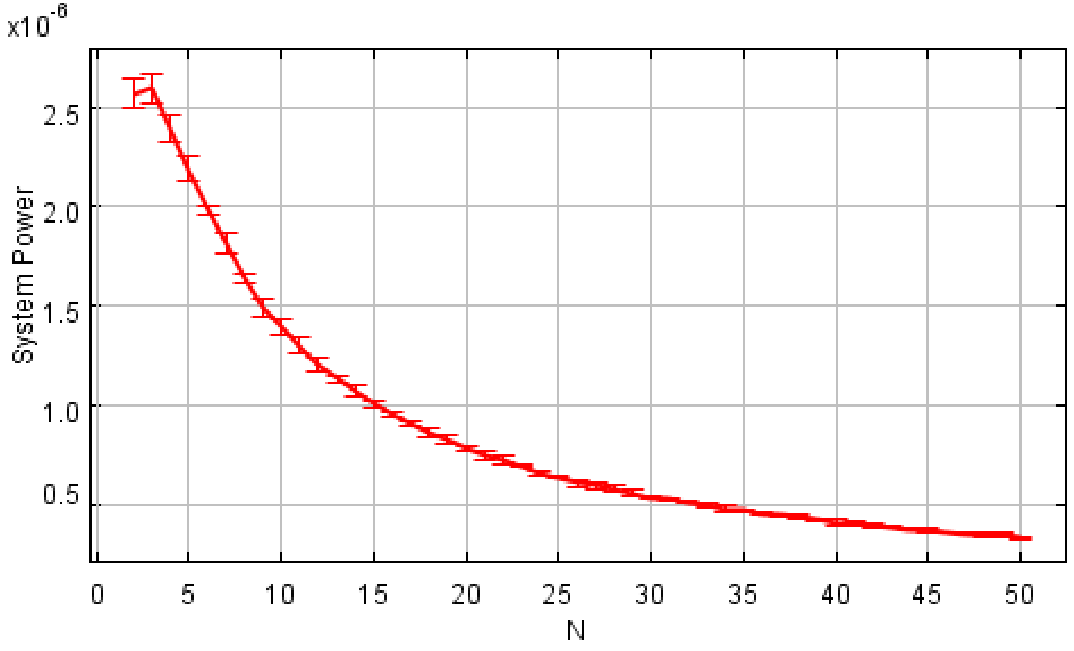
System Power

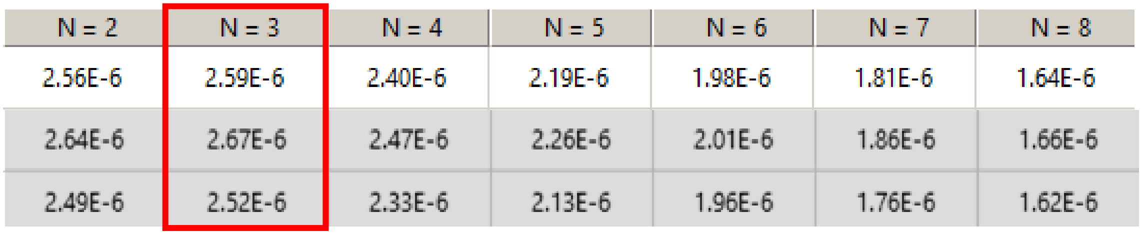
Results were obtained performing what-if analysis on the number of jobs in the system up to N = 50

→ The best tradeoff between throughput and project completion time is achieved when N = 3

→ Simulation results are given with confidence interval 0.99 and max relative error 0.03

→ Results are expressed in [j/h^2]





N = 2	N = 3	N = 4	N = 5	N = 6	N = 7	N = 8
2.56E-6	2.59E-6	2.40E-6	2.19E-6	1.98E-6	1.81E-6	1.64E-6
2.64E-6	2.67E-6	2.47E-6	2.26E-6	2.01E-6	1.86E-6	1.66E-6
2.49E-6	2.52E-6	2.33E-6	2.13E-6	1.96E-6	1.76E-6	1.62E-6

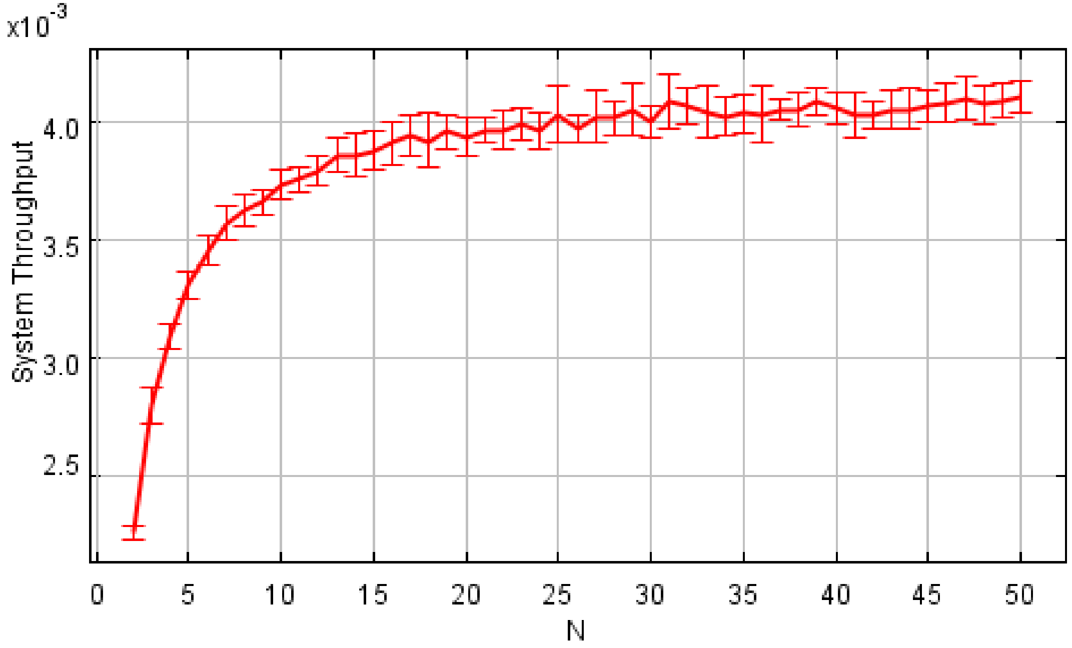
System Throughput

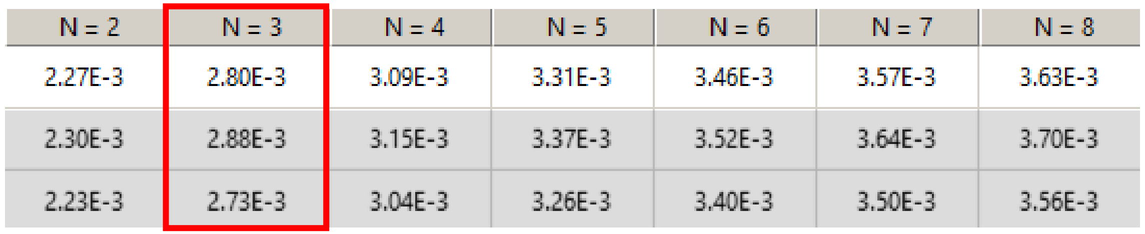
Results were obtained performing what-if analysis on the number of jobs in the system up to N = 50

- → The best tradeoff between throughput and project completion time is achieved when N = 3

- → Simulation results are given with confidence interval 0.99 and max relative error 0.03

- → Results are expressed in [j/h]



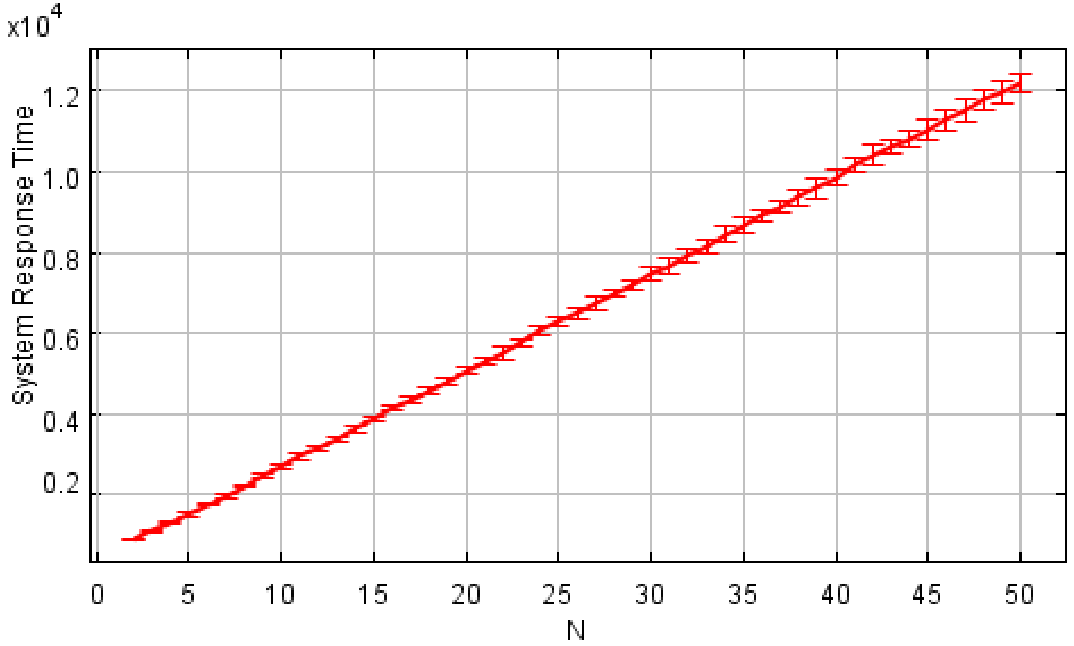


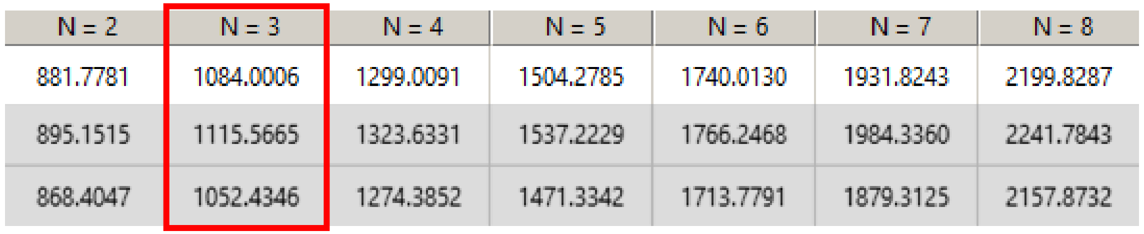
N = 2	N = 3	N = 4	N = 5	N = 6	N = 7	N = 8
2.27E-3	2.80E-3	3.09E-3	3.31E-3	3.46E-3	3.57E-3	3.63E-3
2.30E-3	2.88E-3	3.15E-3	3.37E-3	3.52E-3	3.64E-3	3.70E-3
2.23E-3	2.73E-3	3.04E-3	3.26E-3	3.40E-3	3.50E-3	3.56E-3

System Response Time

Results were obtained performing what-if analysis on the number of jobs in the system up to N = 50

- → The best tradeoff between throughput and project completion time is achieved when N = 3
- → Simulation results are given with confidence interval 0.99 and max relative error 0.03
- → Results are expressed in [h]



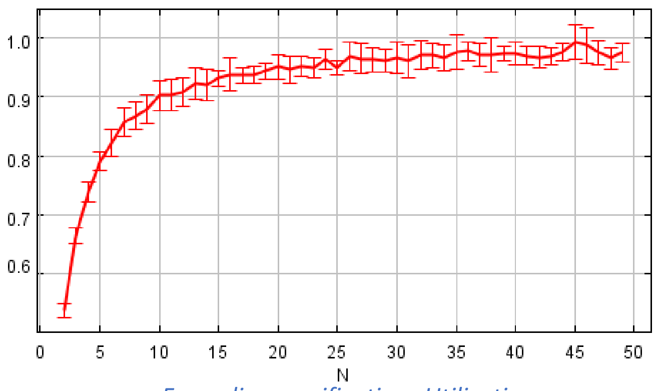


N = 2	N = 3	N = 4	N = 5	N = 6	N = 7	N = 8
881.7781	1084.0006	1299.0091	1504.2785	1740.0130	1931.8243	2199.8287
895.1515	1115.5665	1323.6331	1537.2229	1766.2468	1984.3360	2241.7843
868.4047	1052.4346	1274.3852	1471.3342	1713.7791	1879.3125	2157.8732

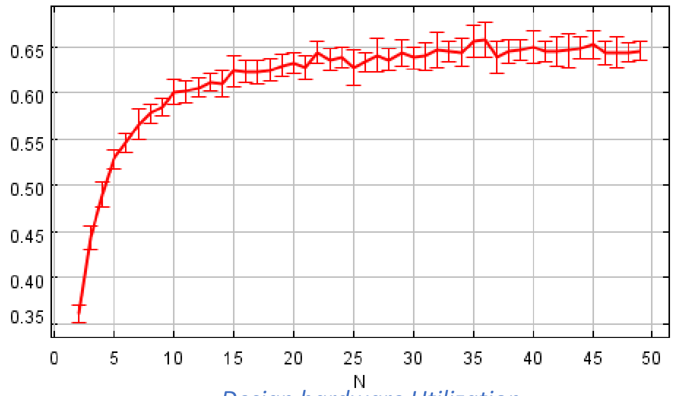
Utilization & Queue Time

Other interesting indices are Utilization and Queue Time of each station, varying the number of jobs in the system

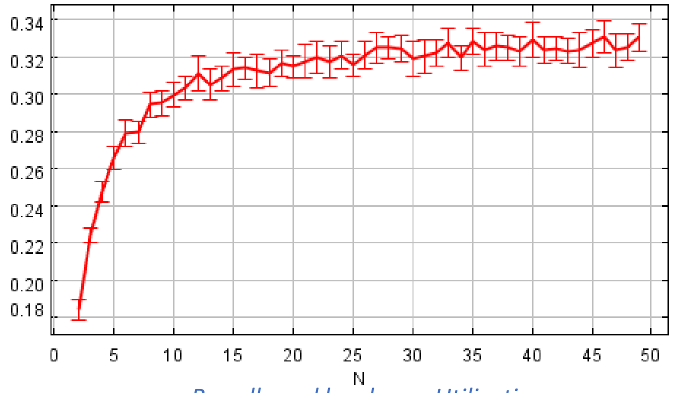
Confidence interval 0.99 - Max relative error 0.05



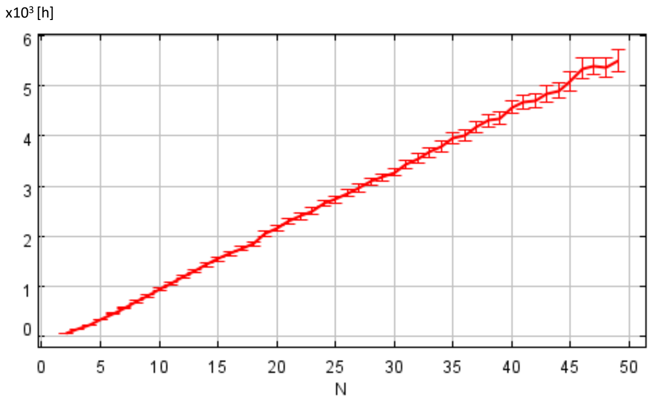
Formalize specifications Utilization



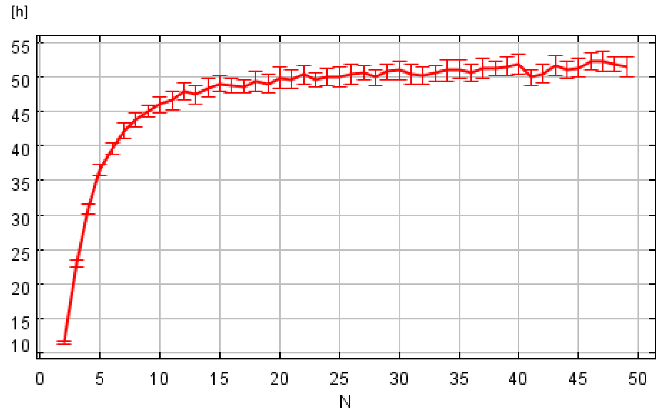
Design hardware Utilization



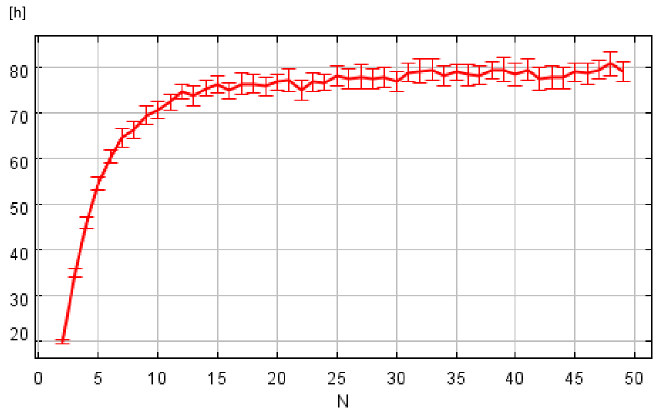
Breadboard hardware Utilization



Formalize specifications Queue Time



Design hardware Queue Time

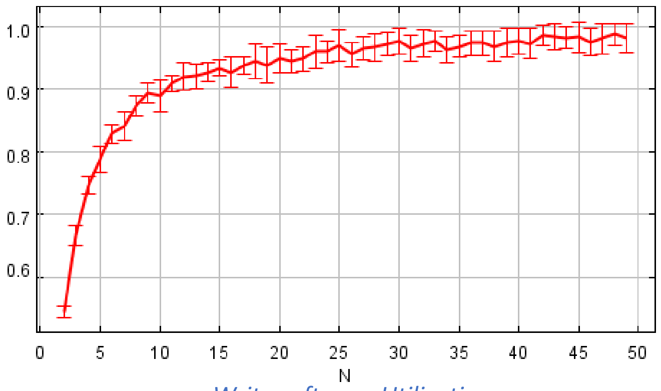


Breadboard hardware Queue Time

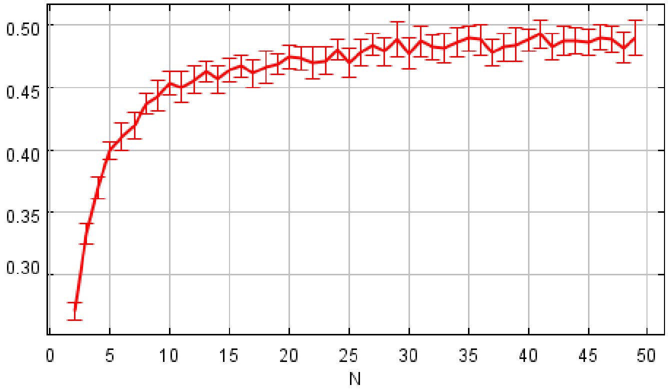
Utilization & Queue Time

Other interesting indices are Utilization and Queue Time of each station, varying the number of jobs in the system

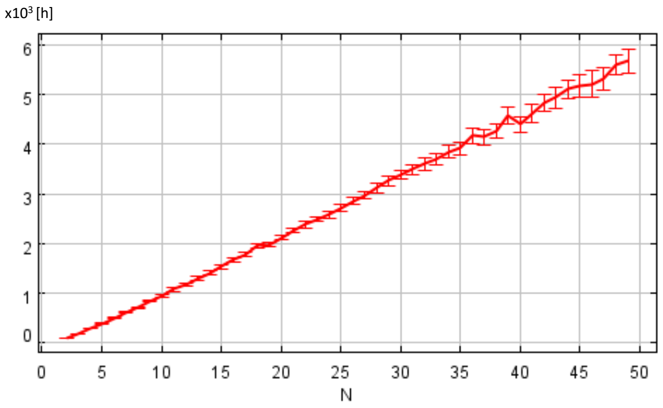
Confidence interval 0.99 - Max relative error 0.05



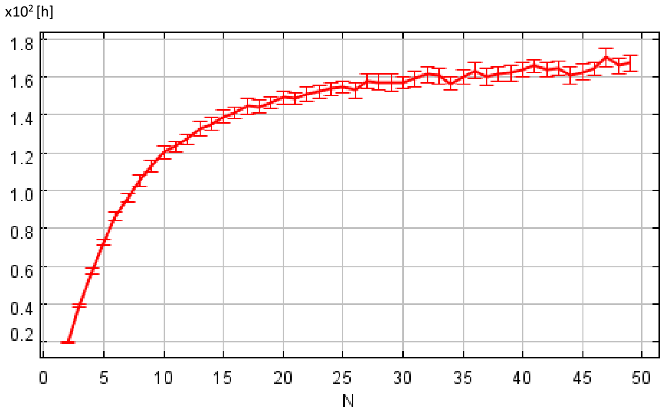
Write software Utilization



Test Utilization



Write software Queue Time



Test Queue Time

Conclusions

→ To understand which is the number of projects that allows to have the best tradeoff between system throughput and system response time, I focused on the System Power index, but I also observed other indices I thought they could be interesting like System Throughput and System Response Time, Utilization and Queue Time for each station

After having analyzed the behaviour of the system, I found out that the best tradeoff was given when N = 3 mainly. Looking just at the Throughput N = 4 is good too, but the Response Time keeps increasing leading to a worse Power, while with N = 2 the Response Time is low but also the Throughput is not so high, even though it has a noteworthy Power

Whereas, Queue Time follows a different trend depending on the inspected station, quite linear for *Formalize specification* and *Write software* departments in particular. These two stations turn out to be the bottlenecks of the model, since they have the highest Utilization indices. Another observation is that both Utilizations and System Throughput saturate when N is about 10 projects