JOURNAL OF SCIENTIFIC, TECHNOLOGY AND ENGINEERING RESEARCH
Bilim, Teknoloji ve Mühendislik Araştırmaları Dergisi, (2021) - 2(2): 22-30
ISSN : 2717-8404 *https://dergipark.org.tr/en/pub/jster*

RESEARCH ARTICLE

A Reliability Assessment of an Industrial Communication Protocol on a Windows OS Embedded PC for an Oil Rig Control Application

*Ongun Yücesan, [1]Altan Özkil, and [2]Mehmet Efe Özbek

*Atilim University, Modes, Ankara, Turkey, ongunyucesan@gmail.com, Orcid.0000-0003-2263-6803,
[1]Atilim University, School of Civil Aviation, Ankara, Turkey, altan.ozkil@atilim.edu.tr, Orcid.0000-0001-8136-6087,
[2]Atılım University, School of Nat. & App. Sci., EEE, Ankara, Turkey, efe.ozbek@atilim.edu.tr, Orcid.0000-0001-5216-7062

HIGHLIGHTS

- *Investigates Reliability vs Load in Window Computer*
- *Processor consumption been investigated in literature but "Reliability vs. Load" is rarely investigated.*
- *Measures reliability of Data History Polls and monitoring*
- *Such a reliability vs. load is observed on physical mockup, findings are hard to predict and close to actual scenarios.*

Keywords:

- Plug and Produce
- OPC UA
- Industry 4.0
- M2M Communications.
- MTBF

GRAPHICAL ABSTRACT

Reliability is an important criterion for the applications, especially when there are humans involved in the activity. There are a limited number of works addressing the reliability concerns of a digital industrial plant scenario with machine to machine (M2M) communications. Results on a windows embedded platform are rare as well. In this paper, we investigate the suitability of such devices for presenting the data of a digital oil rig to the humans within the scope of the TÜBİTAK KAMAG Project. There can be two types of reads. A simple one regarding current values of observed variables. Also, a period of their occurrences can be recalled, referred to as a Historic read. An algorithm can consider both techniques. Our observations indicate: the simple reads are more resilient than the historical ones that represent past data of a readable variable.

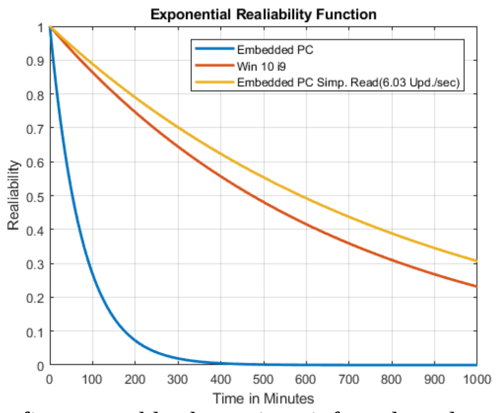

Figure A. The name figure or table about given info and results

Article Info:

Received : 14.07.2021
Accepted : 08.08.2021
Published : 21.12.2021

DOI:
10.53525/jster.971534

***Correspondence:**
Dr. Ongun Yücesan
ongunyucesan@gmail.com

Aim of Article : *Article illustrates a Mean Time Between Failures (MTBF) based method can be used to identify more reliable choice of design element.*

Theory and Methodology: *The techniques employed include development of an industrial communication client using a Prosys Open Platform Communications Unified Access (OPC UA). MTBF based method used to eliminate failures. To choose among the suitable techniques of accessing data, experiments were conducted on a physical connection between the server and a desktop computer.*

Findings and Results: *Polling past history of a data parameter is less reliable than monitoring it.*

Conclusion : *The algorithms used during development of the data monitoring activity need to keep the past of parameters at the server side not allowing the depletion of precious CPU resources and taking advantage of more reliable less burdensome simple data transfers.*

RESEARCH ARTICLE

A Reliability Assessment of an Industrial Communication Protocol on a Windows OS Embedded PC for an Oil Rig Control Application

[*]Ongun Yücesan, [1]Altan Özkil, and [2]Mehmet Efe Özbek

[*]Atilim University, Modes, Ankara, Turkey, ongunyucesan@gmail.com, Orcid.0000-0003-2263-6803,
[1]Atilim University, School of Civil Aviation, Ankara, Turkey, altan.ozkil@atilim.edu.tr, Orcid.0000-0001-8136-6087,
[2]Atılım University, School of Nat. & App. Sci., EEE, Ankara, Turkey, efe.ozbek@atilim.edu.tr, Orcid.0000-0001-5216-7062

Citation:
Yücesan, O., Özkil, A., Özbek, M.E., (2021). *A Reliability Assessment of an Industrial Communication Protocol on a Windows OS Embedded PC for an Oil Rig Control Application*, Journal of Scientific Technology and Engineering Research, 2(2): 22-30.
DOI: 10.53525/jster.971534

HIGHLIGHTS

- *Investigates Reliability vs Load in Windows Computer*
- *Processor consumption been investigated in literature but "Reliability vs. Load" is rarely investigated*
- *Measures reliability of Data History Polls and monitoring*
- *Such a reliability vs. load is observed on physical mockup, findings are hard to predict and close to actual scenarios*

Article Info

Received : 14.07.2021
Accepted : 08.08.2021
Published : 21.12.2021

DOI:
10.53525/jster.971534

***Corresponding Author:**
Dr. Ongun Yücesan
ongunyucesan@gmail.com

ABSTRACT

Reliability is an important criterion for the applications, especially when there are humans involved in the activity. There are a limited number of works addressing the reliability concerns of a digital industrial plant scenario with machine to machine (M2M) communications. Results on a windows embedded platform are rare as well. Extension to existing literature includes reliability measurements with respect to processor load increase. In this paper, we investigate the suitability of such devices for presenting the data of a digital oil rig to humans. There can be two types of reads. A simple one regarding current values of observed variables. Also, a period of their occurrences can be recalled, referred to as a Historic read. For the review made by Geological teams for residues of ore, an algorithm can consider both techniques. Our observations indicate: the simple reads are more resilient than the historical ones that represent a data past of a readable variable.

Keywords: *Plug and Produce, OPC UA, MTBF*

I. INTRODUCTION

In Industrial Automation, Internet of Things (IoT), and Machine to Machine (M2M) applications, special purpose machines need to collaborate and communicate with each other. The success of these applications relies on Embedded or Industrial PCs.

Among the previous studies about performance bottleneck identifications, Burger et. al. [1] indicates the CPU resources deplete before memory or other resources. This work is extended by considering inner server load increases rather than load coming from network with employment of Confidence intervals. Investigation of the reliability comparison with respect to processor load can be seen as an extension to this work. For this purpose, Mean Time between Failure

(MTBF) is employed. This is inversely proportional to the failure rate which is a fundamental parameter in reliability calculations. Cavalieri (et.al) [2] studies the end-to-end delay performance of OPC UA client-server applications, where Transport Control Protocol (TCP) and Simple Object Access Protocol (SOAP/HTTP) are interchangeably used with and without the employment of encryption and security certificates. One of their main conclusions is that security comes at a cost of higher delay. Eckhardt et al. [3] studies the round-trip time of OPC UA Server and client architecture running on Xilinx Embedded PC boards with IEEE Time Sensitive Network (TSN) capabilities. Their results show that milliseconds RTTs can be achieved, so that delay contributions of the sender, network and receiver are comparable with the same order of magnitude. Cenedese et. al. [4], Morato et.al.[24], [22] compares the RTT performances of open source OPC UA servers and clients, where open62541 is identified as more efficient compared to C++ and Phyton based implementations of the same free OpcUa software. Later work includes more varieties, including a result with Prosys Java OPC UA implementation. Kim et al. [5] indicates that demanding more data by decreasing query interval below a certain value does not further improve the amount of queried and read data per second. Unix, Linux and Windows seem to be equally open to the Denial-of-Service attacks (DoS) which have been studied by Neu et. al. [6] for OPC UA. Garcia et. al. [7] work on incorporating OPC UA to IEC 61499 which is a standard promoting inter-operability for distributed industrial automation systems. Their work is concentrated on Oil and Gas industries indicating the need for reliable communication in a harsh environment of this type. [23] Yucesan et. al. includes some generic testing methodologies incorporating MTBF.

Elfeham et. al. [8], Zhang et. al. [9], Hoffmann et. al. [10], Grüner et. al. [11] are among those who make fundamental extensions to OPC UA, Horrmann et. al. [12], Tu et. al. [13], Oksanen et. al. [14], Jo et. al. [15], Kim et. al. [16], Latif et. al. [17], Garcia et. al. [18], Wen et. al. [19] studied field integration of OPC UA, Cho et. al. [20] are among studies where the performance and power consumption are investigated, Kim et. al. [16] investigates backward compatibility of OPC UA.

These works open the way forward for OPC UA. They all make the OPC UA more understandable and usable. This study involves a continuous block of data transfers rather than a single value, which is called Historic Data Access (HDA). Apart from the above studies, we develop a reliability model of HDA on COTS Windows OPC UA platform. Comparisons to a single read and a better hardware platform for the software type HDA experiments are conducted based on reliability as well. The results obtained allows for design of more reliable systems. TÜBİTAK KAMAG project is a project about construction and development of control mechanisms for a Native Oil Rig. Obtained results will be useful in development of monitoring and Human Machine Interface (HMI) software.

The document follows with Section II, description of methodologies followed. In Section III Results are presented. Section IV includes the reliability model estimation. Section V Conclusion includes discussion and final remarks.

II. Methodology

The control application in the form of PLC codes is implemented offline using an automation software development kit (SDK) and loaded into SDK Embedded PC run-time environment. The PLC implementation is performed using limited license *MATLAB Simulink PLC Target* and *Structured Text* script for additional load generation. A real-time test-bed consisting of a COTS Windows OPC UA Server connected through Ethernet to a desktop computer hosting an OPC UA client is utilized for the experiments. For safe and successful operation of the rig, the hardware and software employed for the control application should not have a failure during the operation.

The number of variables which are monitored by the control application is an important parameter determining the amount of the CPU power consumption. By varying the number of variables active on the OPC UA Server, its effects on CPU utilization, Memory utilization and Reliability are studied. The number of successful read/write experiments to the first failure from the beginning of an experiment is recorded for analysis. The mean of these recorded observations represents MTBF, which is a fundamental parameter for reliability predictions. In addition to reading/writing a single value of a parameter, there is a capability that allows one to read the historical values of any parameter. Single reads/writes are observed to be more resilient than historical reads indicating that the reliability bottleneck is the historical read activity. OPC UA client SDK is modified to provide a Human Machine Interface for reading variables from the OPC UA Server. The number of variables observed from the client side is kept constant as two (2) while the number of variables on the OPC UA Server is varied from 12 to 152 for a controlled experimentation.

A. Test Bed

The Test-bed includes one client and one server connected through 100 Mbps Local Area Network of the university. Here, the server is relatively low-capacity Intel 1.3 GHz Single Celeron Processor, 2GB Ram, *Windows 7 Embedded Real Time Operating system*, passive cooled "Embedded PC". A COTS Windows based OPC UA server runs on this platform and communicates through the Ethernet ports. In test-bed, there is also a "Desktop computer" hosting the modified Prosys OPC UA client software. The desktop computer has Intel quad core i7, 3.4 GHz clock rated processor with 16GB RAM. This computer also keeps the Programmable Logic Controller (PLC) code which is loaded to Embedded PC run-time on the server. The PLC code defines the variables to be observed. To increase the testing coverage, a limited licensed version of a MATLAB Simulink target is employed to convert Simulink blocks to PLC code.
The Software infrastructure as depicted in Fig. 1 consists of COTS lightweight components for Embedded PC while all processing intensive user interface for coding and/or Simulink is performed on Win 10 Desktop. When you need to run the developed code you can transfer it to the embedded platform saving from the processing power.

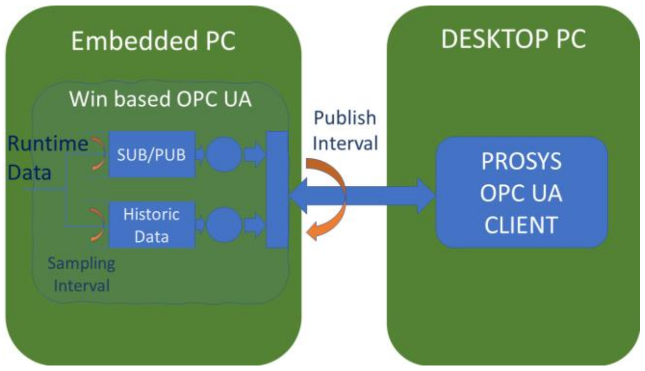

Figure. 1. The Experimental Software Setup.

Once a variable is subscribed, it is read at a regular interval called *publish interval* which is determined by the client every time subscription is made. It is also possible to monitor the working flow of the PLC code variables from Desktop while they are running on Embedded PC. This brings an extra load on the Network, Memory, and CPU, from which we observe their effects in our experiments. This capability can be turned on and off with respective referring names as "logged in" and "logged out" activity. After the operation is started, the PLC run-time exchanges information with the COTS OPC UA server on Embedded PC which can be connected from the OPC UA client. By increasing the number of "integer variables" from the set {12, 52, 102, 152}, defined in the PLC code, the CPU and Memory are overloaded for experimentation. To test the effect of Embedded PC Server hardware on reliability, we utilize Intel i9 platform @2.9 GHz with 32 GB RAM with 64 bit Windows 10 Home OS while Prosys Client code runs on the same Desktop.

B. Sample Size

During the initial phases of the study, the limitations and resources, represented in the sense of a metric that can be employed both for guiding the programming effort and purchase of the equipment were not that much present. A windows tool with research results is also rare in the literature. Our aim was to identify worst reasonable performing number of variables as a metric representing the load on Embedded PC. This knowledge would be a hypothetical threshold for number of variables, which should not be surpassed or if had to be surpassed hardware upgrades or algorithmic improvements should be made. A coarse increment of counters is handy for such knowledge's understanding. For this aim, Double Integers (DINT) counters of Structured Text coding is employed. These are 32-bit registers. The lowest value they can represent is -2147483648and the highest is 2147483647. As per computer CPU hardware, an update in value of a counter means first data is taken to the register then value updated on CPU, later re-sent back to memory location. Therefore, counters continuously are upgraded, which we hope will fully represent worst case, in the sense of heaviest data update scenarios. Our counters are mostly changing between -15000 to 25000. They are incremented with different values from 1 to 11 and sometimes 100 as well. This way, simultaneous logic operations are prevented presenting a more average case scenario.

At first, we started from small steps trying to see the limitations. Activities were conducted in "logged-off" manner. Considering the Antivirus burden of the CPU no scan activity is allowed during experimentation, which is started after upgrades and scan activities were completed.

In the case of expeditions, gradually starting from 12 variables, which we increased observing the effects till 152 variables. CPU utilization wasn't too different however at 152 variables Embedded PC started to demonstrate certain lags and delays where we stopped further increasing our number of counters or variables.

C. Sample Characteristics

During Read Count observations, we observed that the experiments conducted by repeating the read experiments repetitions were a little low at first. However, during this phase we notice that the CPU on a passively cooled Embedded PC is affected by the room temperature in read counts sense. Oil rigs do have air conditioned, a high volume of cooled air flowing cabinets. We try to represent this by employing two fans. One is sucking hot air coming from the Embedded PC and the other is pouring fresh air of room temperature of roughly 22 to 24 degrees Celsius into the PC to the best of our abilities.

The CPU utilizations are irrelevant to the heat of CPU, however all Read counts and reliability observations are made under cooled environment. The simple read count is obtained during a non-cooled experiment yet still representative of its worst case, since if cooled would perform to a longer duration. It's *publish interval* is 10 ms, slightly above the single cycle completion time of a loop with two (2) variables on our Embedded PC, which we observed to be roughly around 5 to 7 ms. Therefore, it is representing its worst-case scenario with at best a marginal difference. By observations of worst-case scenarios, it is possible to find a more reliable method based on calculation. Scaling the results to certain usage profiles, it can highlight some expected reliabilities.
In an Oil rig Control software monitoring scenario, under conditions with depleted CPU resources, the control software run can be interrupted causing interruption to crane controls. This can cause dangerous accelerations for crew working under heavy equipment. Whenever HMI for monitoring conditions have failed, it may cause certain parameters to go high without alerts. These may cause fires, explosions, catastrophic failures. Also, precious ore can be missed.

III. RESULTS

To model an oil rig environment one server on an Embedded PC and a client modified to needs on a Desktop PC was considered. One can read data from the server using two different methods: subscription based and historic data access readings. The latter is, as its name indicates, the history of the subscribed values.

A. Central Processing Unit (CPU) Utilization for Historic Data Access Reads

History reads/queries are used for monitoring the progress of an activity in an Oil Rig environment. For the conduct of the experiments, the history data is kept in Embedded PC Hard-disk with 4 samples/second and 10000 queue size. 7.5 minutes window of data is queried by Prosys Client. The number of variables observed in OPC UA Server is changed from 12 to 152. As the number of variables approaches 150, our tiny 1.4 GHz Celeron presents effects, like mouse pointer not moving smoothly and fluently. Averages and maximum utilization are taken over a 120 second interval while historic access queries are conducted with a 10 second period. This experiment is repeated five (5) times for statistical accuracy. All data is presented with 90\% confidence intervals marking the uncertainty on them. The x-axis on figures presents the number of variables included in experimentation representing the load, while y-axis is the representative of the CPU utilization that is occurring with that load. In Figure 2, we observe average CPU utilization, where "logged in activity" resides slightly higher over "logged out activity". However, as the number of variables increases, the CPU utilization figures for logged out and logged in cases approach each other.

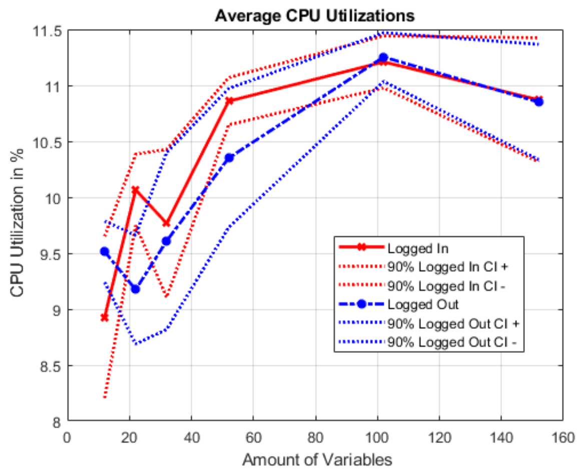

Figure 2. Average Utilization, both logged in and out.

The load created by historic data access is low considering the Ram capacity of the Embedded PC. The utilization figures reflect the load increases, and all of the average utilization measurements are within the 3% margin. The associated 90% confidence intervals are tight as depicted in the figure. The oscillations are similar for both logged in and out activities. These also are indicative of the lack of wide register resources of the Celeron CPU. There is a chance the tasks were not served on time but had to wait or were dropped in Random Access Memory (RAM) in a random manner.

The results were of the average utilization. As an outline of other observations regarding the hardware resources: The maximum CPU utilization is always within 4% margin and reaches to 100% levels when the number of the variables is around 50. This situation is visible in Figure 3. However, "Logged in" maximum utilization does not reach 100% level. These can be due to overwhelming job demands on a single core CPU with no waiting queue. The Confidence intervals in the figure can give clue for how much extra CPU demand exists. Even though, it is not possible to have more than 100% utilization, the overshoots indicate variabilities. In load levels, where CPU is more utilized, success rate of the tasks at hand should be high. It is because of the lesser variance in these regions. The levels, where CPU is not well utilized, are accompanied with high variances. It is reasonable to assume there are more off times with respect to rejected tasks in these loads. Therefore, overshooting confidence interval boundaries indicate higher chance for interruption to services.

The levels for memory, both logged in and out activity, are consistent with findings in literature in the sense that they are relatively flat. Logged in case requires slightly less memory, this may be due to the job demands returning to the RAM, to be accessed back in a random time. In cases where CPU utilization approaches 100%, there can be interruptions to other functions on the server.

B. *Effect of Number of Variables on Read Counts*

In this section, we report the history read counts, which are defined as the number of successful read attempts from the beginning of an experiment.

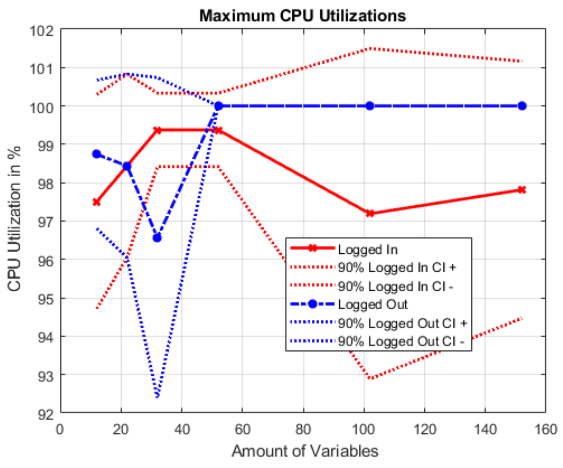

Figure 3. Average Utilization, both logged in and out.

Figure 4, shows the history read counts by varying the number of variables when the query period is set to 15 seconds. This selection of query period gives a chance for the client to present data on time, yet it is still among the lowest observed duration of reads from the previous section, namely worst-case. The mean of history read counts increases as the number of variables changes from 12 to 152. The COTS OPC UA server employed includes two program pieces to achieve its functions. One is called Direct Access (DA) Server and other is called Unified Access (UA) Server. Both are needed to achieve operation. DA server for OPC framework uses TCP for data transmission. UA server is the new server architecture that allows for various transmission control protocols. Even if there is no increase in network demand, internal communication between these two program pieces should be using TCP for communicating. TCP is known to have better latency performance under heavier load because of protocol overheads. Increasing trend can be because of such a situation. The standard deviation σ and confidence interval boundaries keep widening indicating the highest uncertainty at 152 variables. This result is intuitive due to the modest embedded computing resources.

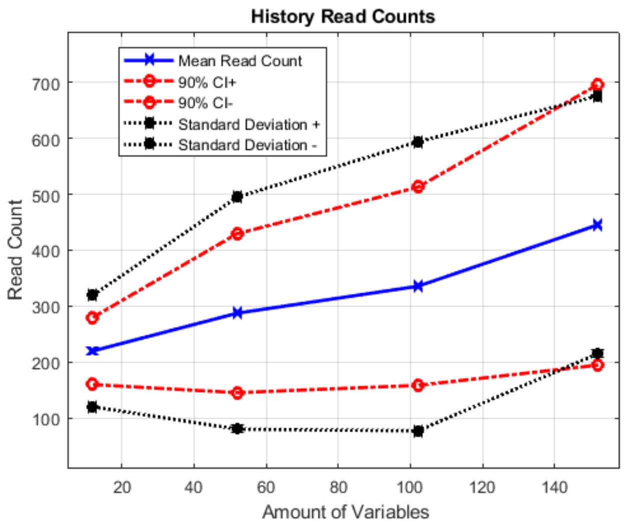

Figure 4 Effect of Increasing variables count on reliability

We carry out another experiment by moving the OPC UA server to a stronger hardware while keeping the Prosys client on the same desktop. The new server hardware is a 64bit Win10 OS working on i9 Intel platform. We observe that the stronger hardware provides 2732 successful history reads until its termination while Embedded PC provides lower than 500 read counts for all experiments.

C. Reading and Writing Data with Client/Server Subscription/Publishing

Up to this point, we have reported the results of Historic Data Access read. In the subscription-based method, a variable in the server side needs to be "subscribed" by the client for monitoring its updated values. For experiment in this section Publish Interval is set to 10 ms. The number of data read and write counts is reported as 1.163 Million and 764K, respectively while the PLC Runtime/OPC UA server is running on the Embedded PC. With respect to the observations made for the historic read counts, 1.163 Million is significantly higher. In another set of observations, we have observed 210675, 326690, 381121 where the experiment was terminated at the end of a duration. 6.03 updates are made every second during this set of experiments. The publish interval was again selected as 0.01 corresponding to 10ms. The average obtained from the second set is 306162 single value updates.

IV. Reliability Model Estimation

The observed read counts from the experiments are translated into time by multiplying with the query period Δ, where Δ is 15 seconds. This choice of the query period is due to its closeness to the worst case read count achieved in time perspective. It also allows avoiding unnecessary processing delays in the client side. By using all the values obtained from Figure 4, the sample mean of history read counts is calculated as μ = 304 Reads. This is translated into time as follows $\mu_t = \mu \times \Delta = 304 \times 15$ seconds = 4560 seconds = 76 Minutes. In this study, we employ the exponential reliability function. Using the expectation of $E[X_{\text{Exponential}}\ \_] = 1/\lambda = \mu_t$ as defined in the book by Trivedi [21], the exponential reliability function can be expressed as in equation (1).

$$R(t_{\text{Exponential}}) = e^{-\lambda . t} \qquad (1)$$

The same calculation for historic read count on OPC UA Server on *Win 10 i9 PC* yields the result of $\mu_{t-\text{win10}} = \mu \times \Delta = 2732 \times 15$ seconds = 40980 seconds = 683 Minutes in order to serve as a basis for a comparison.

We translate the second set of simple read counts into time by calculations based on updates per interval, this value is according to the observations made during, second set of values collection $\mu_{t-\text{sub}} = \mu \times \Delta$ = 306162 / 6.03 upd./sec. = 50773.13 seconds = 846.22 Minutes.

The estimated exponential reliability functions obtained from the above calculated values are shown in Figure 5. Highest reliability of history read count values come from strong i9 platform as depicted with a yellow line. Observation from subscription reads with orange lines are residing in the highest range. Reliability of the historic read counts is the lowest curve drawn with blue lines.

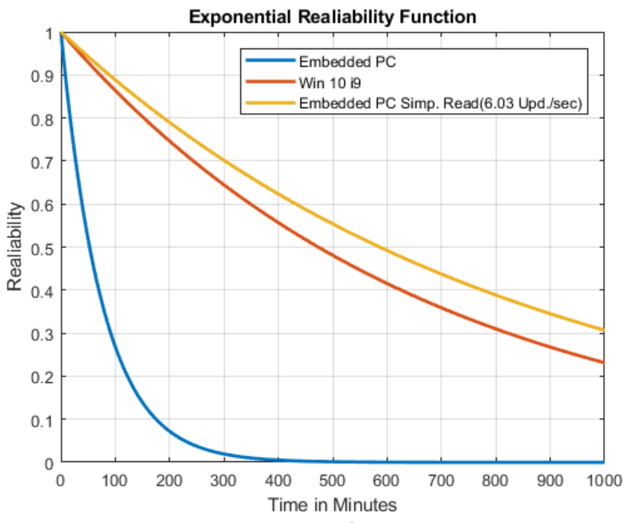

Figure 5. Reliability Estimate of the Embedded PC with 15 second Query Period

We can see that our model using historic read counts is a lower bound for the system reliability compared to subscription reliability model. From the figure, we can also observe that our software platform is capable of achieving higher reliability levels on stronger hardware compared to those observed in a limited resourced Embedded PC indicating correct functioning given broad resources.

V. CONCLUSION

In this paper, an Embedded PC, as an OPC UA communication server is considered. There are three scenarios. These consist of the historic access; the subscription reads and History Access over a stronger Intel i9 platform. The MTBF figures are employed in the classical exponential reliability models. The results indicate that the HDA activity has a lower chance of success. The same software on powerful i9 hardware yields better reliability.

Over the TUBİTAK KAMAG project, the results been reported to implementation team. The purchase of a stronger hardware was addressed as the writing date of the article. Reports to the development team indicating chances for overload with HDA access been provided to be considered during still ongoing effort. The outcomes are indicative of benefits of making physical experiments rather than using available predictions. The oscillations of CPU utilizations are characteristic with Celeron Processors, whereas increasing trend in MTBF vs load figure may be due to TCP achieving better with increasing load. Widening confidence intervals are useful in representing increasing uncertainties.

As a future work, further investigation of statistical distributions underlying the failure behavior can be considered.

CONFLICTS OF INTEREST

They reported that there was no conflict of interest between the authors and their respective institutions.

RESEARCH AND PUBLICATION ETHICS

In the studies carried out within the scope of this article, the rules of research and publication ethics were followed.

ACKNOWLEDGMENT

This research is supported by TUBITAK (KAMAG) under the grant no: 115G007. Research is conducted for the needs of Turkish Petroleum Anonymous Partnership (TPAO) and Turkish Petroleum International Company (TPIC).

REFERENCES

[1] A. Burger, H. Koziolek, J. Ruckert, M. Platenius-Mohr, and G. Stomberg, "Bottleneck identification and performance modeling of opc ua communication models," in Proceedings of the 2019 ACM/SPEC International Conference on Performance Engineering, ser. ICPE '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 231–242.
[2] S. Cavalieri and F. Chiacchio, "Analysis of opc ua performances," Computer Standards and Interfaces, vol. 36, no. 1, pp. 165 – 177, 2013.
[3] A. Eckhardt and S. Muller, "Analysis of the round-trip time of opc ua and tsn based peer-to-peer communication," in 2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), Sep. 2019, pp. 161–167.
[4] A. Cenedese, M. Frodella, F. Tramarin, and S. Vitturi, "Comparative assessment of different opc ua open–source stacks for embedded systems," in 2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), Sep. 2019, pp. 1127–1134.
[5] W. Kim and M. Sung, "Opc-ua communication framework for plc-based industrial iot applications: Poster abstract," in Proceedings of the Second International Conference on Internet-of-Things Design and Implementation, ser. IoTDI '17. New York, NY, USA: Association for Computing Machinery, 2017, p. 327–328.
[6] C. V.Neu, I. Schiering, A. Zorzo. Simulating and Detecting Attacks of Untrusted Clients in OPC UA Networks. CECC 2019: Proceedings of the Third Central European Cybersecurity Conference. 2019, p.1-6.
[7] M. V. Garcia, E. Irisarri, F. Perez, E. Estévez, and M. Marcos, "An opencpps automation architecture based on iec-61499 over opc-ua for flexible manufacturing in oilgas industry", 20th IFAC World Congress. IFAC-PapersOnLine, vol. 50, no. 1, pp. 1231 – 1238, 2017
[8] H. Elfaham, F. Palm, S. Gruner, and U. Epple, "Full integration of matlab/simulink with control application development using opc unified architecture," in 2016 IEEE 14th International Conference on Industrial Informatics (INDIN), July 2016, pp. 371–376.
[9] L. Zhang, "Specification and design of cyber physical systems based on system of systems engineering approach," in 2018 17th International Symposium on Distributed Computing and Applications for Business Engineering and Science (DCABES), Oct 2018, pp. 300–303.
[10] M. Hoffmann, C. Buscher, T. Meisen, and S. Jeschke, "Continuous integration of field level production data into top-level information systems using the opc interface standard," Procedia CIRP, vol. 41, pp. 496 – 501, 2016,
[11] S. Gruner, J. Pfrommer, and F. Palm, "Restful industrial communication with opc ua," IEEE Transactions on Industrial Informatics, vol. 12, no. 5, pp. 1832–1841, Oct 2016.
[12] R. Hormann, S. Nikelski, S. Dukanovic, and E. Fischer, "Parsing and extracting features from opc unified architecture in industrial environments," in Proceedings of

the 2nd International Symposium on Computer Science and Intelligent Control, ser. ISCSIC '18. New York, NY, USA: Association for Computing Machinery, 2018.

[13] N. T. T. Tu, N. D. Cuong, V. V. Tan, and H. Q. Thang, "Research and development of opc client-server architectures for manufacturing and process automation," in Proceedings of the 2010 Symposium on Information and Communication Technology, ser. SoICT '10. New York, NY, USA: Association for Computing Machinery, 2010, p. 163–170.

[14] T. Oksanen, P. Piirainen, and I. Seilonen, "Remote access of iso 11783 process data by using opc unified architecture technology," Comput. Electron. Agric., vol. 117, no. C, p. 141–148, Sep. 2015.

[15] G. Jo, S.-H. Jang, and J. Jeong, "Design and implementation of cpps and edge computing architecture based on opc ua server," Procedia Computer Science, vol. 155, pp. 97 – 104, 2019.

[16] J. Kim, G. Jo, and J. Jeong, "A novel cpps architecture integrated with centralized opc ua server for 5g-based smart manufacturing," Procedia Computer Science, vol. 155, pp. 113 – 120, 2019.

[17] H. Latif, G. Shao, and B. Starly, "Integrating a dynamic simulator and advanced process control using the opc-ua standard," Procedia Manufacturing, vol. 34, pp. 813 – 819, 2019.

[18] M. V. Garcia, E. Irisarri, F. Perez, E. Estévez, D. Orive, and M. Marcos, "Plant floor communications integration using a low cost cpps architecture," in 2016 IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA), Sep. 2016, pp. 1–4.

[19] X. Wen, H. Chen, B. Wen, J. Liu, Y. Li, and N. Xi, "Conceptual framework of smart factory based on opc ua and lstm encoder-decoder," in 2018 IEEE 1st International Conference on Micro/Nano Sensors for AI, Healthcare, and Robotics (NSENS), Dec 2018, pp. 44–48.

[20] H. Cho and J. Jeong, Performance Evaluation of Industrial OPC UA Gateway with Energy Cost-Saving: Third International Conference, SmartCom 2018, Tokyo, Japan, December 10–12, 2018, Proceedings, 12 2018, pp. 55–66.

[21] K. S. Trivedi, Probability and Statistics with Reliability, Queuing and Computer Science Applications, 2nd ed. GBR: John Wiley and Sons Ltd., 2001.

[22] A. Morato, S. Vitturi, F. Tramarin and A. Cenedese, "Assessment of Different OPC UA Implementations for Industrial IoT-Based Measurement Applications," in IEEE Transactions on Instrumentation and Measurement, vol. 70, pp. 1-11, 2021

[23] O. Yucesan, A. Ozkil, "Time Complexity Comparison of Stopping at First Failure and Completely Running the Test." J Electron Test vol. 36, pp. 409–417, 2020.

[24] A. Morato, S. Vitturi, F. Tramarin and A. Cenedese, "Assessment of Different OPC UA Industrial IoT solutions for Distributed Measurement Applications," 2020 IEEE International Instrumentation and Measurement Technology Conference (I2MTC), Dubrovnik, Croatia, 2020, pp. 1-6.