Proceedings of the 4th International Conference of Control, Dynamic Systems, and Robotics (CDSR'17)
Toronto, Canada – August 21 – 23, 2017
Paper No. 122
DOI: 10.11159/cdsr17.122

Determination of Cycle Time Constraints in Case of Link Failure in Closed Loop Control in Internet of Things

Arpit Ainchwar, Dan Necsulescu
University Of Ottawa
75 Laurier Ave E, Ottawa, Canada
aainc092@uottawa.ca; Dan.Necsulescu@uottawa.ca

Abstract **-** In today's era of the Internet of Things, it is crucial to study the real-time dependencies of the web, its failures and time delays. Today, smart grid, sensible homes, wise water networks, intelligent transportation, infrastructure systems that connect our world over fast developing. The shared vision of such systems is typically associated with one single conception internet of things (IoT), where through the employment of sensors, the entire physical infrastructure is firmly fastened with information and communication technologies; where intelligent observation and management is achieved via the usage of networked embedded devices. The performance of a real-time control depends not only on the reliability of the hardware and software used but also on the time delay in estimating the output, because of the effects of computing time delay on the control system performance. For a given fixed sampling interval, the delay and loss issues are the consequences of computing time delay. The delay problem occurs when the computing time delay is nonzero but smaller than the sampling interval, while the loss problem occurs when the computing time delay is greater than, or equal to, the sampling interval, i.e., loss of the control output. These two queries are analysed as a means of evaluating real-time control systems. First, a general analysis of the effects of computing time delay is presented along with necessary conditions for system stability. In this paper, we will focus on the experimental study of the closed loop control system in the Internet of Things to determine the cycle time constraints in case of link failure.

Keywords**:** Internet of Things (IoT), link failure, cycle time, time delay, closed loop control

1. Introduction

A wireless sensor network (WSN) is a composition of nodes where each node is fitted with sensors to measure some physical phenomena like pressure, light, temperature, etc. WSNs are considered as one of information collecting methods to build systems which will improve the efficiency of infrastructure. Compared with the wired solutions, WSNs emphasise on easier deployment and better versatility of devices. With the technological improvement of sensors, WSNs will become the key technology for IoT. Today's world is full of sensors in vehicles, in factories for controlling emissions, in homes, in smartphones, and even in the ground monitoring soil conditions in vineyards. Although it appears that sensors have been here for a while, research on WSNs began back in the 1980s, and it is only since 2001 that WSNs generated an enhanced interest from technical and research perspectives. This is due to the availability of economical, low powered miniature components like microcontrollers, processors, radios and sensors that were often integrated on a single chip (system on a chip(SoC)) [1].

The concept of the internet of things (IoT) was formed parallel to WSNs. The term Internet of Things was devised by Kevin Ashton in 1999 [2] and introduces to identifiable objects and their descriptions in an Internet-like structure. These "internet-like" structures can be composed of anything from industrial plants, machines, cars, specific parts of the larger system to animals, human beings and plants and even specific their body parts. While IoT does not imply a communication technology, wireless communication technologies will play a significant part, and WSNs has many applications in many industries. The small, sturdy, inexpensive and low powered WSN sensors will bring the IoT to even the smallest objects can be installed in any environment, at reasonable costs. Integration of these objects into IoT will be a significant part of further development of WSNs.

A WSN can be defined as a system of nodes that cooperatively sense and may regulate the conditions, enabling interaction between persons or computers and the surrounding environment [3]. In fact, the activity of sensing, processing, and communication with an inadequate amount of power, generates a cross-layer perspective typically requiring the joint consideration of distributed signal/data processing, medium access control, and communication protocols [4].

Through integrating existing WSN applications as part of the infrastructure system, possible new applications can be recognised and improved to meet future technology and new market trends. For instance, WSN technology utilizations for the smart grids, smart water, intelligent transportation systems, and smart home generate enormous amounts of data, and this data can serve many purposes.

By executing the sequence of instructions, a computer based controller performs the underlying control algorithms. Not only the MTBF (mean time between failures) of the controller hardware and software but also the delay in executing control algorithms affects the reliability of a digital control system, unlike wired analog control systems. The time of execution of the control algorithm is determined as the period from its trigger to the generation of the corresponding control command. This is referred as the extra time delay introduced to the feedback loop in a controlled system. For a given control algorithm the execution time or the computing delay is a random variable usually smaller than the corresponding sampling interval is because of the existence of the conditional branches and resource sharing delays.

A real-time digital control processor or controller computer can be considered as a three-stage channel: data acquisition from sensors, data processing to create control commands, and realization of the commands by the actuators. Every step will take the time to complete; this paper is concerned mainly with the data transmission processing, the most severe stage. The other two are much easier and more static [5]. More precisely the subject of the paper is to study the link failure effects for different time intervals in a wireless internet network and collect the data for the comparable study of various frequency cycles.

2. Time Delays and Data Losses

Network delays are usually short. However, packets can get dropped in a network system. For reliable connections, software is necessary checks for losses and for resending. If a resend is required, the overall delay is doubled; another round-trip time is computed for a resend request and response. For higher speed, stable data transfer protocols the impact can be even greater.

2.1. The Delay Components

The equation given describes the packet delay at a single node along its route from source to destination.

$$d_{nodal} = d_{proc} + d_{queue} + d_{trans} + d_{prop} \tag{1}$$

d_{proc} = nodal processing
d_{queue} = queueing delay
d_{trans} = transmission delay
d_{prop} = propagation delay

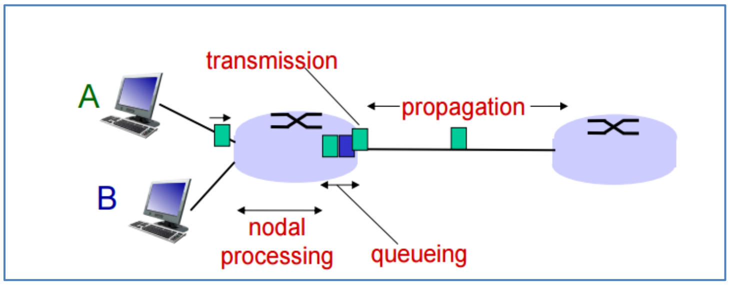

Fig. 1: Delay components.

2.2. Packet Losses

There are two principal causes of lost packets:

- Queue overflow
- Noise

The breakdown of delay equation makes it easy to analyze different delay components. Usually, network delays are small. However, lost packets in the network increase the impact of the delays. The computing time delay is considerably different from the usual system time delay and cannot be taken care of before establishing a system in use.

When the computing time delay is long versus the sampling interval (but small about the mission lifetime), it may severely affect control system performance. Depending on the magnitude of computing time delay about the sampling interval, its effects on the control system are categorized into either a delay or loss problem. To be more precise, let ξ and T_s, denote the computing time delay and the sampling interval, respectively. A delay problem results when $0 < \xi < T_s$, and the loss problem occurs when $\xi \geq T_s$. The former represents the undesirable effects caused by a nonzero computing time delay smaller than the deadline (that is, the start of the next sampling interval) of a control algorithm or task, while the latter represents the case of no update of control output for one or more sampling intervals.

The sampling rate must be chosen precisely not only satisfying the requirements of the Shannon's sampling theorem [6] but also accomplishing the expected performance. A good example to confirm this can be found in [7] where series of robot control experiments were conducted with different sampling rates. Time delay computation becomes more pronounced in the case of increasing the sample rate. However, these effects can not be neglected, particularly when the time constant of the plant is short and the order of the plant is high [8].

3. Experimental Setup

Fig. 2 shows an experimental setup to study the time delay and data losses in the given wireless network. The system one consists of (Arduino Uno + Wi-Fi Shield + LED) and the system two consists of (Arduino Uno + Wi-Fi Shield + Photocell). Both the systems are enclosed in the box to ensure the better readings. LED attached to the shield can be operated (blinked) continuously with regular time interval using the program run on the Arduino through IDE software. The photocell connected to the other shield will sense the blinking of the LED and sends the reading to the attached server, and it gets recorded subject to time constraints. The recorded data can be studied for the time delays and the link failures. The timing of each data recorded let us know the time delay in execution of the loop. If the time delay is greater than that of the sampling interval, there is a loss of the data, which we can find out from the experimentally recorded data.

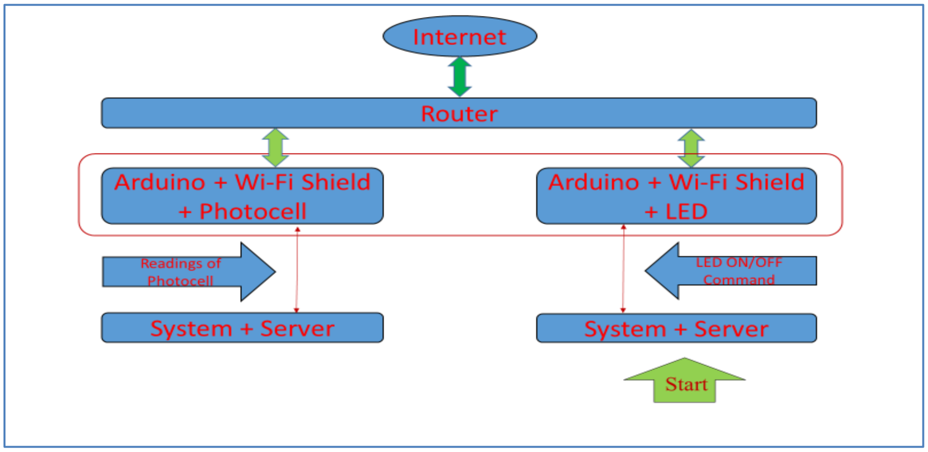

Fig. 2: Experimental setup of closed loop control system.

4. System Configuration

The experiment requires two sets of Arduino Uno^{TM} and Arduino Wi-Fi Shield 101^{TM} connected with each other in two separate systems. One system of (Arduino Uno + Arduino Wi-Fi Shield) is connected with the LED, while the other system contains the photocell.

4.1. Arduino Uno^{TM}

Arduino Uno is a board based on the microcontroller ATmega328P. It includes 14 digital input/output pins (of which six can be used as PWM outputs), and six pins are for analog inputs; it also has 16MHz quartz crystal, a power jack, a USB connection, a reset button and an ICSP header. It comprises everything needed to support the microcontroller. It needs to connect it with a computer USB cable, or we can power it with an AC-to-DC or battery. The UNO board is the reference model for the Arduino platform, the first in the series of USB Arduino Boards.

The Arduino board can be programmed with Arduino Software (IDE). The ATmega328 on this board comes preprogrammed with the bootloader which allowed you to upload the new code without any use of the external hardware programmer. It uses the original STK500 protocol. We can also program the microcontroller through the ICSP (In-Circuit Serial Programming) by bypassing the bootloader.

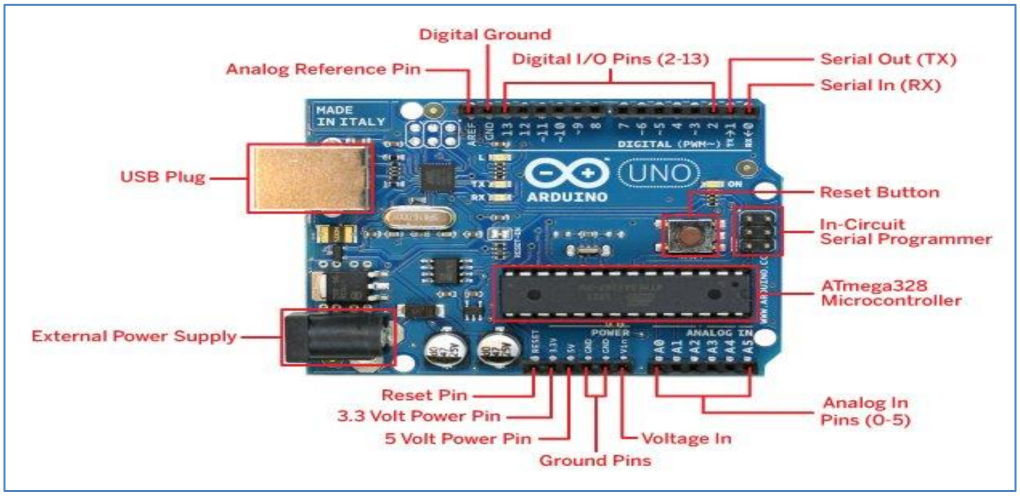

Fig. 3: Arduino Uno.

Arduino UNO does not use the FTDI USB-to-serial driver chip; instead, it features the Atmega 16U2 programmed as a USB-to-serial converter. This microcontroller board can be powered externally or via the USB connection. The power source is selected automatically.

4.2. Arduino Wi-Fi 101 Shield™

Wi-Fi 101 shield is a powerful IoT shield with crypto-authentication which allows you to wirelessly connect the Arduino, developed with ATMEL by using the IEEE 802.11 wireless specifications (Wi-Fi). Compliant with the IEEE 802.11 b/g/n standard the shield is based on the Atmel SmartConnect-WINC1500 module. The WINC1500 module is a network controller capable of TCP and UDP protocols. This shield is designed specifically for the IoT applications and features a hardware encryption/decryption security protocol implemented by the ATECC508A Crypto-Authentication chip which is an ultra-secure method to provide key agreement for encryption/decryption.

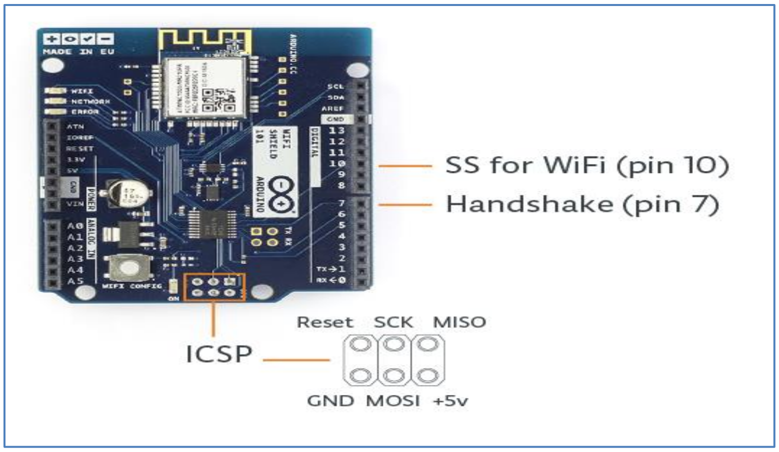

Fig. 4: Arduino Wi-Fi shield 101.

4.3. LED

Light-emitting diode (LED) is a light source with a two-lead semiconductor. It emits light when activated. It is a p-n junction diode. These are typically small of the size less than 1 mm^2. To shape its radiation pattern integrated optical patterns may be used.

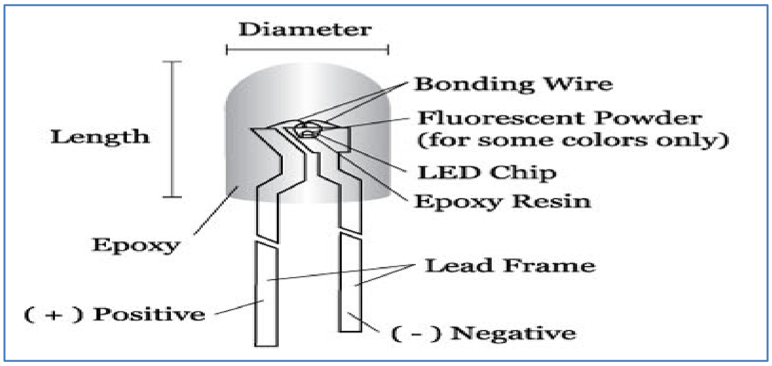

Fig. 5: LED.

4.4. Photoresistor

The photoresistor is made of a high resistance semiconductor. It is a light-controlled variable resistor that reacts to the intensity of the light; in other words, it exhibits photoconductivity. It is useful in light-sensitive detector circuits.

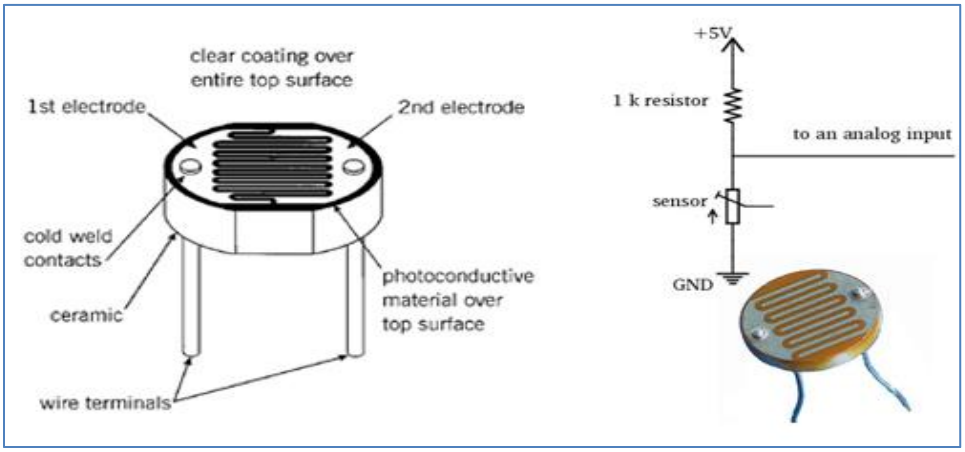

Fig. 6: Photoresistor.

5. Results

The increasing implementation of Internet of Things has made it crucial to thoroughly analyze the effects of time delay and data losses for the given wireless environment to study the link failures and find out cycle time in case of link failures. This computing time delay is varied from system time delay; it is a random delay resulting from the execution of control programs.

Consequences of computing time delay are categorized as the delay and loss problems which can be analyzed for cycle time to link failure. The experimental setup described in this article concentrates on the development of the practical approach to study the closed loop link failure problems and eventually find its randomness characteristics and cycle time to link failure for executing it for different time intervals. This study will help in determining the suitable time interval or frequency for performing the closed loop control accounting for data losses and link failures.

Fig. 7 shows the recorded data from examining the experiment. Time constraint is associated with each registered data which ultimately helps to find the delay and data losses which can be used in determining the suitable time interval for performing the closed loop control with reduced effects of failures.

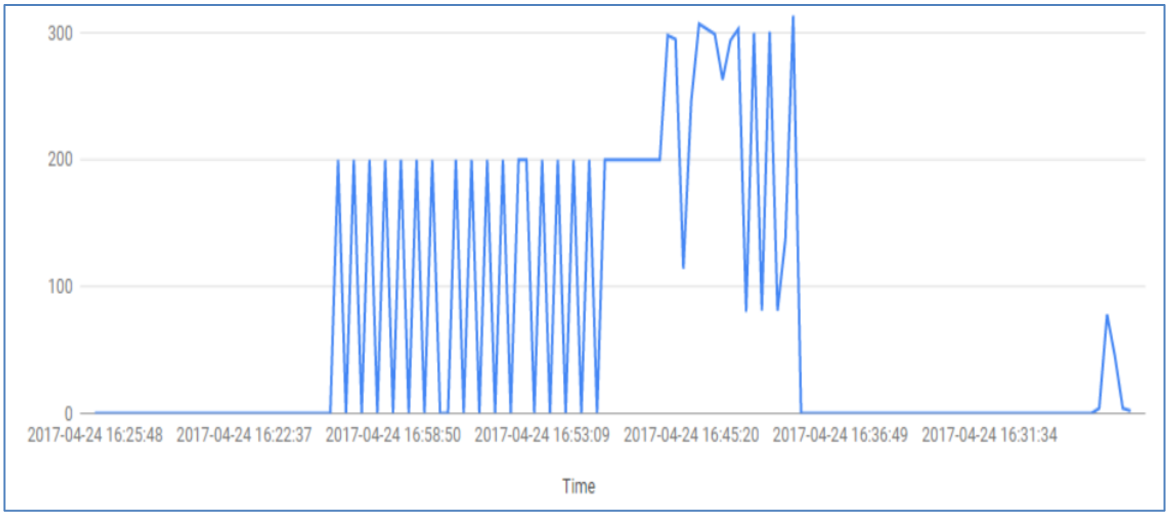

Fig. 7: Experimental results with time constraint.

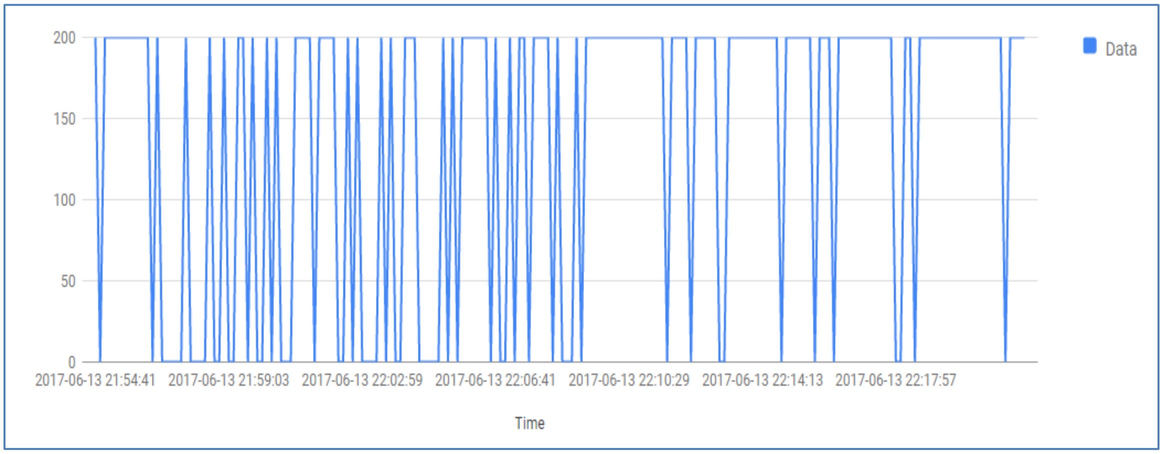

Fig. 8: Experimental graph results for 2 seconds' time interval.

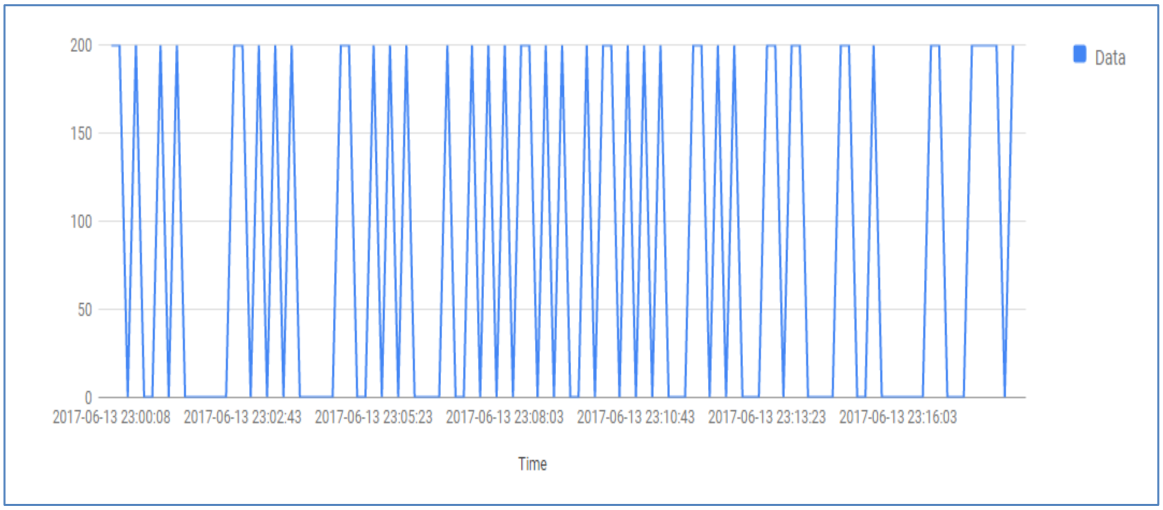

Fig. 9: Experimental graph results for 5 seconds' time interval.

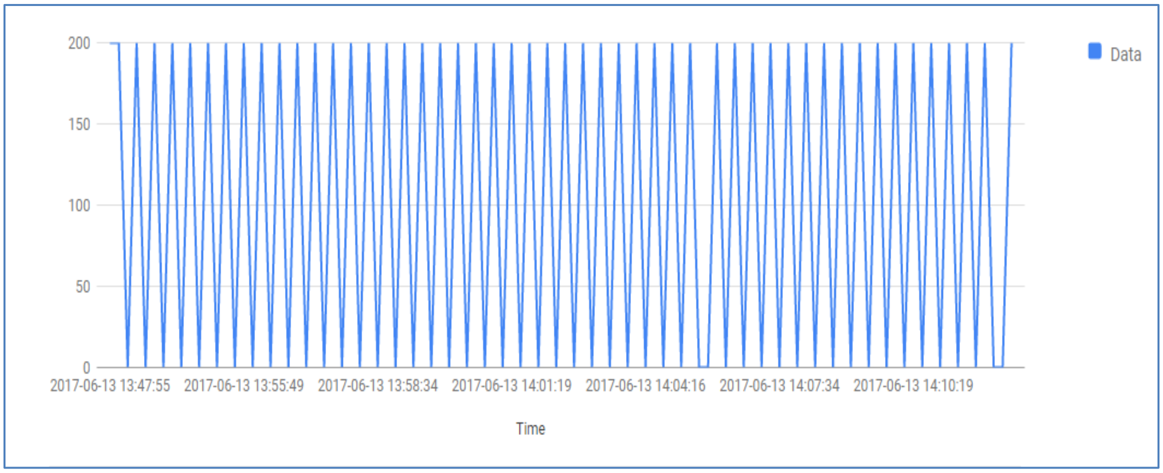

Fig. 10: Experimental graph results for 10 seconds' time interval.

In the graphs, the x-axis represents the time of the data recorded while the y-axis denotes the intensity of the LED light. The flat lines on the x-axis show the data losses as there is no data recorded at that particular time constraint.

Each experimentally recorded data in the Fig. 7 has its time constraint. The collected data for the sampling interval shows the randomness throughout the period. From the figure, we can observe that sometimes the time constraint of two continuous data is greater than the fixed sampling interval which shows the time delay, while for certain consecutive readings the time difference is in multiples of sampling intervals which shows us the loss of data.

Fig. 8, Fig. 9 and Fig. 10 shows the experimentally recorded data for the 2, 5 and 10 seconds' time interval of LED blinking respectively, each recorded data has its own time constraint. As we observe the figures thoroughly the data delay and data loss are more evident in the case of 2 seconds' time interval as compared to the other two graphs. As we compare the results of 5 seconds' and 10 seconds' time interval data delay and loss are more apparent in 5 seconds' graph than the graph represents the 10 seconds' time interval. This shows that as the time interval for closed loop control system increases there are lesser chances of data delay and data losses resulting in fewer failures of the system.

The experimental results show that, as the frequency of the closed loop control system increases, the more the chances of the failures because of the data delay and data loss as compared to the closed loop cycles of higher time intervals, i.e., lower frequency.

6. Conclusion and Future Work

The experimentally recorded data shows the randomness in the time delay and the data losses. The future work will help present the intended experimental study of a WSN for a velocity remote control of a mobile robot and a block diagram like Fig. 2, but for the robot velocity command in the form of a sinusoidal velocity command and robot velocity estimation subject to random cycle time. This study will help to operate remotely based sensor nodes and eventually record the sensor data accounting for the time delay and data loss problems.

References

[1] Y. Shu, P. Lanctot, and Fan Jianbin, "Internet of things: wireless sensor networks," *White Paper, International Electrotechnical Commission*, 2014.

[2] K. Ashton, "That 'Internet of Things' Thing. In the real world, things matter more than ideas," *RFID Journal*, vol. 22, 2009.

[3] A. Broring, et al, "New generation sensor web enablement," *Sensors*, vol. 11, 2011.

[4] Sensei, Integrating the physical with the digital world of the network of the future. [Online]. Available: http://www.sensei-project.eu/

[5] G. S. Kang and X. Cui, "Computing Time Delay and Its Effects on Real Time Control Systems," *IEEE transactions on control systems technology*, vol. 3, no. 2, 1995.

[6] T. Strohmer and J. Tanner, "Implementations of Shannon's sampling theorem, a time frequency approach," vol. 4, no. 1, pp. 1-17, 2005.

[7] P. K. Khosla, “Choosing Sampling Rates for Robot Control,” in *Proceedings of IEEE Int. Con. Robotics Automat.*, pp. 169-174, 1987.

[8] T. Mita, “Optimal digital feedback control systems counting computation time of control laws,” *IEEE Trans. AuromatContr.*, vol. AC-30, no. 6, pp. 542-547, 1985.