Proceedings of the 9th World Congress on Electrical Engineering and Computer Systems and Sciences (EECSS'23)
Brunel University, London, United Kingdom - August 03-05, 2023
Paper No. CIST 143
DOI: 10.11159/cist23.143

Real-time Interfacing of a Single-Rod Electrohydrostatically Actuated Excavator Machine

Hossam Elwehishy[1,3], Saeid Parvanesekam[1], Witold Kinsner[2], Nariman Sepehri[1]
[1] Department of Mechanical Engineering, Price Faculty of Engineering, University of Manitoba
Winnipeg, MB R3T 5V6, Canada
[2] Department of Electrical and Computer Engineering, Price Faculty of Engineering, University of Manitoba
Winnipeg, MB R3T 5V6, Canada
[3]Department of Mechanical Engineering, Faculty of Engineering, Kafrelsheikh University,
Kafrelsheikh, Egypt
elwehish@myumanitoba.ca; parvanes@myumanitoba.ca; witold.kinsner@umanitoba.ca; nariman.sepehri@umanitoba.ca

Abstract **-** This paper describes the results of a study that centres on designing, prototyping, and evaluating of a low-cost and robust interface system using a board computer (Raspberry Pi) to perform data acquisition from an experimental excavator-type machine equipped with a hydrostatic powered circuit. Determining the excavator's status in real time is essential for its efficient and safe operation. To accomplish this objective, it is necessary to obtain real-time data that are both accurate and precise for a comprehensive representation of the machine condition. To ensure the reliability of the data transmission process, a communication system must also be implemented with the capability to receive, process, and transmit the required data to the server without losses or distortion. Such an interface system capable of high-quality data acquisition and transmission in hard real-time has been implemented. It obtains the data from sensors, processes the data, and sends the data to the server in real time through software that interacts with a remote computer, allowing the operator to monitor the excavator's condition through a graph displayed on a personal computer in real-time.

Keywords**:** Hard real-time interfacing; data acquisition; excavator; interface system; hydrostatic powered circuit; Raspberry Pi.

1. Introduction

An excavator is a powerful and versatile piece of heavy-duty equipment that is widely utilized in construction for a variety of tasks such as digging, ground levelling, transporting, and dumping loads, and providing straight traction. Excavators are powered by hydraulic systems, known for their high-power density, stiffness, and compactness [1]. Such systems can be divided into two types: (i) valve-based and (ii) hydrostatic-based. The valve-based systems use valves to control the flow of fluid, which can result in power losses [2]. Hydrostatic-based systems, on the other hand, use a direct connection to the pump, improving power delivery [3]. In these systems, the actuation force is created based on the fluid pressure differential between the two chambers of the cylinder. The chamber pressures are built upon the flows in and out, as well as the speed of the actuator. Therefore, the pressure signals of chambers play important roles in the system performance [4]. The absence of throttling losses (due to the direct control of the hydraulic cylinder by the pump) is the main advantage of the electrohydrostatic system [5].

Using an electrohydrostatic excavator has several challenges that must be considered. During the operation of the actuator, there are often variations in the system parameters. For example, changes in the demand for fluid can cause fluctuations in the output pressure of the hydraulic supply pump, particularly when multiple actuators are connected to a common supply. Additionally, alterations in the fluid temperature or contamination of the oil can affect the effective bulk modulus of the hydraulic fluid. The actuator viscous damping can also change at different locations in the stroke [6]. Systems may also experience actuator internal leakage from worn piston seals. The internal leakage affects the actuation performance as the flow moves between two chambers of the cylinder [4]. These systems can also be highly nonlinear.

In addition to the above challenges, the utilization of remotely controlled excavators is of paramount importance in scenarios where the environment poses a significant hazard, such as nuclear accidents or earthquakes, where it is not safe for human workers to be present. All these challenges highlight the need for a real-time interfacing system that can enable accurate observation of the excavator's condition.

In order to acquire data from any source, a data acquisition system (DAQ) is required. Although DAQ boards come in a variety of models and capabilities, most commercially available DAQs have at least one of the following two major features: they sketch an online graph of the signal, and they store data samples for post-experiment analysis [7]. The DAQ in this paper is an interface system responsible for data acquisition, processing, and transmission to a server, where the data can be monitored by the operator. The safe operation of the excavator and reliable and precise data acquisition are critical factors in this process. The interface system should also be able to transfer the data without errors and distortion as fast as possible. Furthermore, it should be able to provide a comprehensive representation of the excavator's conditions to enable the operator to make informed decisions and take necessary actions.

The next section of this paper provides an overview of the excavator system, followed by the required sensors that have been installed on it (Sec. 3), the robust interface system (Sec. 4), the data transfer process (Sec. 5), edge computing techniques utilized to ensure the valuable data represent the operation of the excavator (Sec. 6), and the experimental results obtained from the implementation of the interface system. We conclude the paper with remarks based on our findings and discuss potential areas for future research.

2. Excavator System

The excavator utilized in this research is built on a John Deere 48 backhoe, as depicted in Fig. 1. There is an actuator which is connected to the arm of the backhoe Fig. 2(a). This actuator generates the necessary force for the excavator to handle the load mass. The tip of the excavator, located at the end of the perpendicular link one, is loaded with weights to simulate real-world situations.

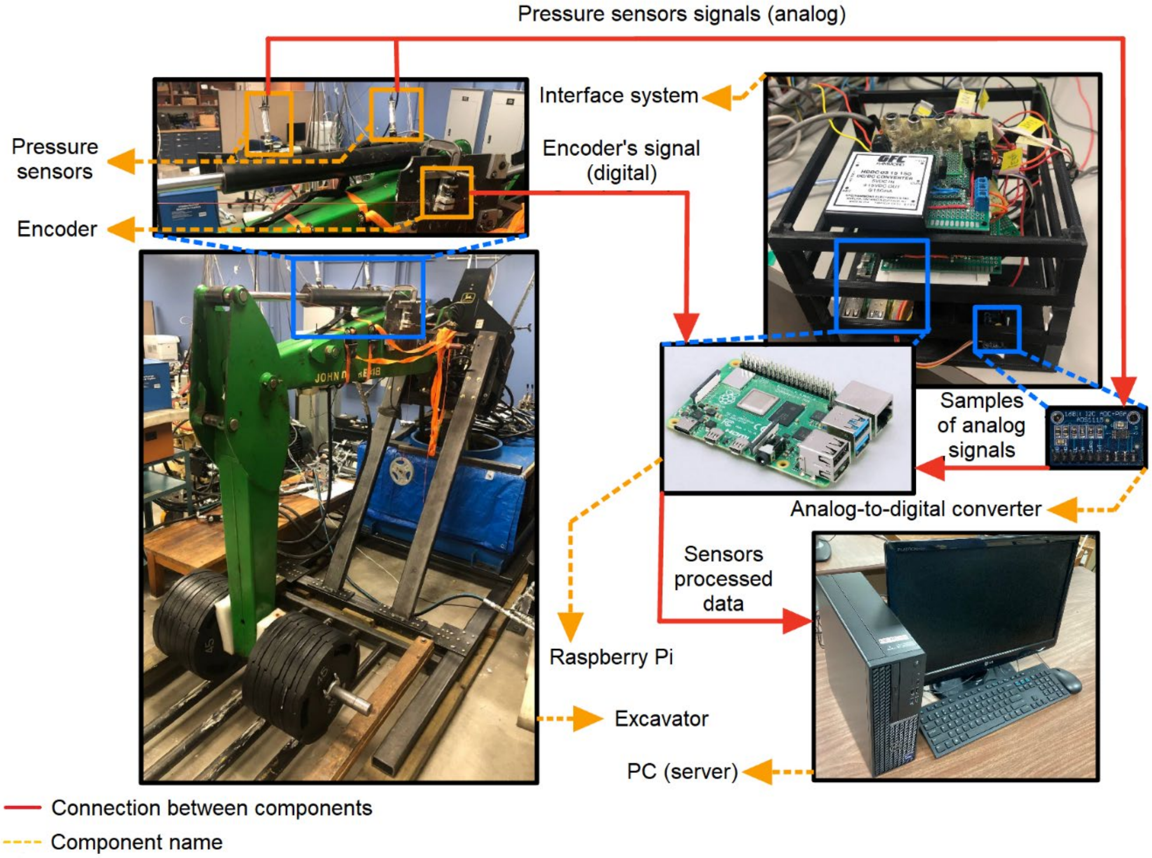


Fig. 1: Schematic of the connections between the excavator, sensors, interface system and the PC.

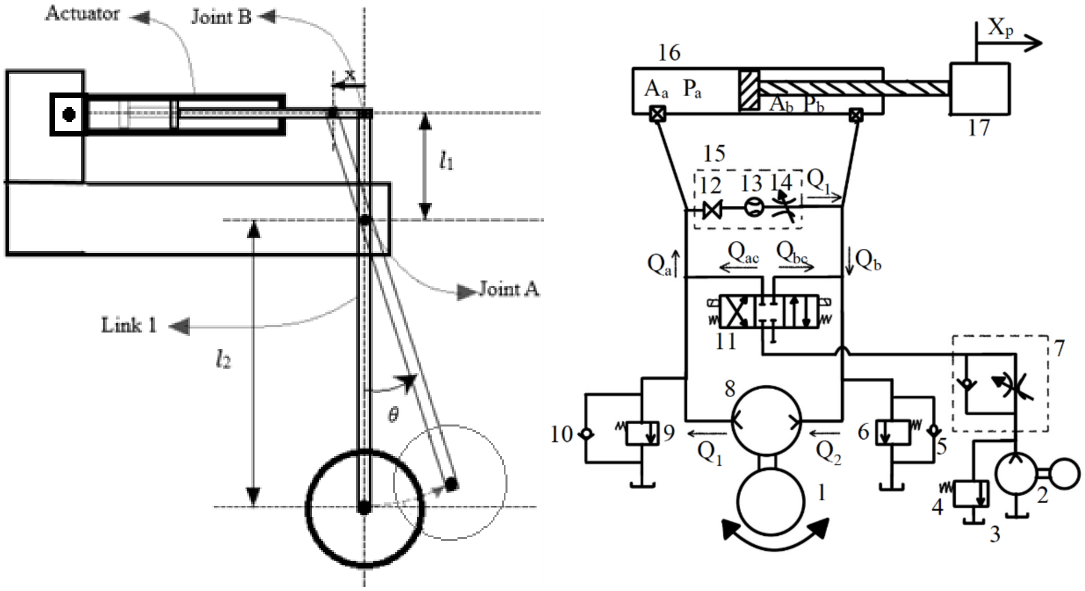


Fig. 2: (a) Schematic of excavator arm; b) Diagram of electrohydrostatic actuator circuit.

Table 1: Figure 2. Description of components in Fig. 2(b).

Part Number	Part Name	Part Number	Part Name
1	Servomotor	10	Check valve
2	Auxiliary pump	11	Three-position four-way directional valve
3	Tank	12	Manual ball valve
4	Relief valve	13	Flowmeter
5	Check valve	14	Orifice
6	Relief valve	15	Artificial Leakage
7	One-directional flow control valve	16	Actuator
8	Bidirectional pump	17	Mass of Piston and rod
9	Relief valve		

The electrohydrostatic actuator produces the force necessary for the excavator to operate. The system comprises of an electrohydrostatic motor, a hydraulic pump, a directional control valve, and a hydraulic cylinder. The electric motor drives the hydraulic pump, which generates hydraulic pressure. This pressure is controlled by the directional control valve that directs the flow of hydraulic fluid to the actuator illustrated in Fig. 2(b). The hydraulic pump is connected directly to the hydraulic cylinder. The actuator converts the hydraulic pressure into linear force, which is used to move the excavator's link one. The schematic diagram of the circuit for the electrohydrostatic actuator is depicted in Fig. 2(b).

3. Sensors

Accurate monitoring of the actuator's condition is one of the requirements. To monitor the actuator's condition effectively, it is necessary to capture and measure the pressures present at both ends of the actuator. This information can then be used to determine the condition of the system. So, two pressure sensors are utilized to capture and measure the pressures at the actuator's two ends. These sensors (identified as K17M0211F23000# and K13M0211F23000 in Fig. 1) are manufactured by ASHCROFT. The sensors have a 5-millisecond response time. Since the output signals of these sensors are analog, should be sampled at a proper sampling frequency and a proper number of bits per sample.

An incremental encoder is used on the excavator to determine the excavator's tip position. The amount of prismatic movement of the actuator rod captures by the encoder, and the excavator tip position is determined through the position of the actuator rod. The encoder integrated into the excavator is a Bourns® model EN – Rotary Optical Encoder, depicted in Fig. 1. This encoder generates a digital signal and has a resolution of 256 cycles per revolution and a square signal wave. The encoder output has a resolution of 0.03mm for the actuator output the rise and fall time is 200ns (nanoseconds).

4. Interface System

The required acquisition of the sensors data has been designed, and its schematic diagrams is shown in Fig. 1. The data captured by the sensors should be denoised in the system and then be transferred to a server. The interface system includes all these responsibilities. The task of transferring the signals generated by sensors to a server can be broken down into several stages. Firstly, the system receives the signals from the sensors. After this stage, the Raspberry Pi processes the received signals to generate the corresponding data. After all the processed data are transmitted to a computer located in the laboratory, which acts as a server for the data. This allows for real-time monitoring of the system's status.

The successful completion of the tasks performed by the interface system requires the integration of two key components, a Raspberry Pi, and an analog-to-digital converter (ADC). These components are integral to the functioning of the system and work in tandem to provide a safe and reliable operation for the excavator. The specific model of the Raspberry Pi used in this system is the Raspberry Pi 4 model b V.2018, and it is paired with the ADC through the I2C protocol. The inter-integrated circuit (I2C) protocol allows multiple "peripheral" digital integrated circuits (chips) to communicate with one or more "controller" chips. The ADC that has been set up in the interface system is Texas Instruments' ADS1115 having 8 channels combined with a 16-bit resolution, operating at a sampling rate of 8 to 860 samples per second (sps). This combination of components is critical in ensuring the accuracy and precision of the data transmitted to the server, thereby contributing to the overall efficiency and effectiveness of the excavator's operation.

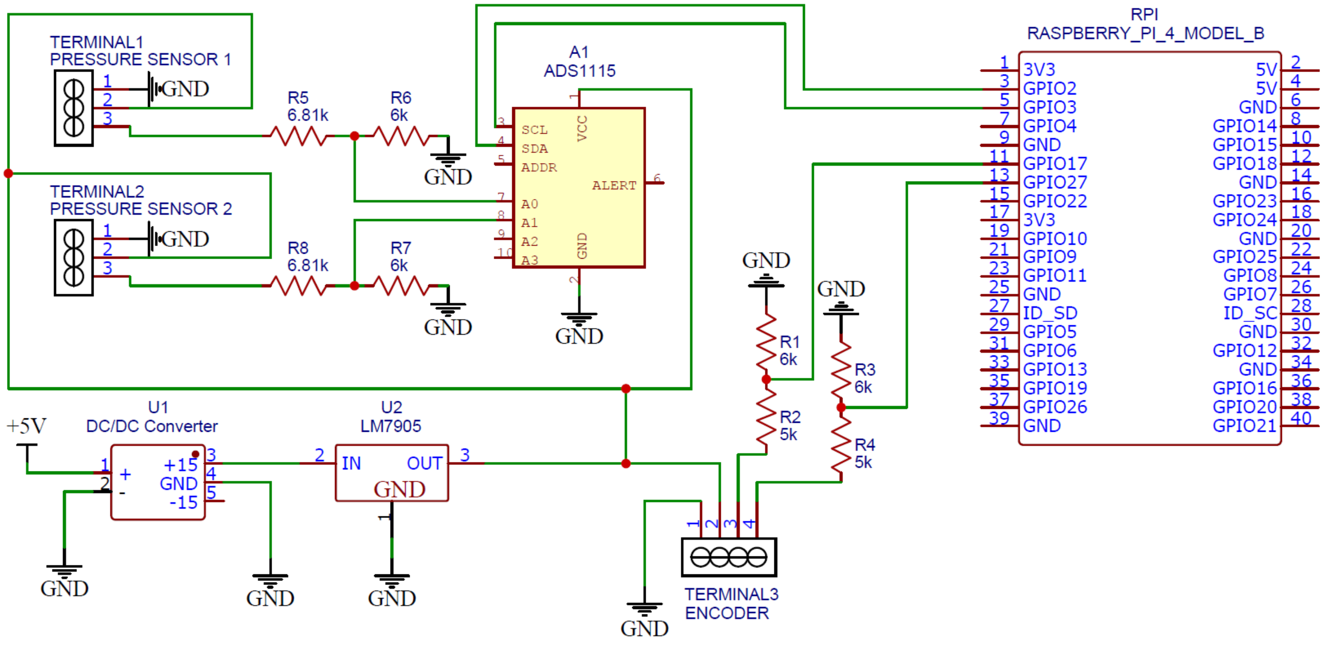


Fig. 3: Schematic circuit of the developed interface system.

5. Data Transfer

The transfer of data in the current paper involves receiving the signals from the sensors and their subsequent transfer to to a personal computer (PC). This presents a formidable challenge, especially in terms of ensuring a thorough and real-time time transfer of sensor data to the PC. To achieve this, there are two connections that must function optimally, (i) connection connection between sensors and the interface system, and (ii) connection between the interface system and the PC. Wire connection is used between the sensors and the interface system, and the connection between the interface system and the PC has been established via Ethernet LAN (local area networks). These connections have been chosen based on their reliability, speed, and ease of use.

The data received by the interface system consist of two analog signals and one digital signal. The analog signals are received and sampled by the ADC and then transferred to the Raspberry Pi. There are the following two factors important in determining the sampling rate: (i) the data should present the real-time condition of the excavator, and (ii) complying with requirements to reconstruct the signals. Since the cutoff frequency of the analog signals in this system is 10 Hz, the Nyquist sampling theorem requires the proper sampling rate to prevent aliasing is more than 20 sps.

Due to the reason that the pressure sensors have a 5 millisecond response time, the time step between two samples should be less than 5 milliseconds to show the excavator condition in real-time. This means that the actual sampling rate of the system should be more than 200 sps. Based on this real-time requirement, the Nyquist requirement is satisfied. Thus, because of reduction the requirement, 220 sps is chosen as the sampling rate.

To determine the number of bits required in each sample, the worst case for the waveform is considered. The following equation for the aperture time determines the bits required [9]:

$$T_{a\,min} \leq \frac{FS}{2^n \left(\frac{dv(t)}{dt}\right)_{max}} \tag{1}$$

Based on the above expression, the worst case is defined by the maximum slope of the waveform. In the equation $T_{\text{a min}}$ is worst-case aperture time, FS is the signal full-scale magnitude range (full scale), $\left(\frac{dv(t)}{dt}\right)_{max}$ is maximum slope of the signal and n is the number of bits required. The available n=16 satisfies this aperture-time requirement.

To ensure seamless communication between interface system and the PC, the client-server protocol is utilized which transfers data through Ethernet LAN. In a client-server model, a connection between the server (personal computer) and the client (interface system) establishes. The server listens for incoming client requests on a specific network port, and when a request is received, a new socket is created to handle data transmission between the two systems. The PC will receive data through this connection. The connection established using the client-server is written in the C language, resulting in fast, reliable, and efficient communication between the interface system and the personal computer. The data transfer process is seamless, enabling the personal computer to receive the real-time information from the interface system. This is of utmost importance as it allows for a transfer speed of 4 milliseconds per sample, which is faster than the 5 milliseconds response time of the pressure sensors that serves as the bottleneck of the system.

6. Edge Computing

The data generated by the interface system must match the need for real-time monitoring of the excavator's operation with the need for efficient data transmission. This, in turn, requires a balance between sample size and computational complexity. Edge computing provides a solution to this problem by enabling data processing to occur closer to the source. It can optimize the use of available communication bandwidth and improve the speed of data transfer. This is achieved by embedded processing capabilities to the interface system. The embedded processes enable the interface to conduct tasks such as noise removal and data size reduction locally and in real time.

In this paper, we use denoising rather than filtering. Denoising is the key to signal processing technology, which is the process of removing noise to the maximum extent of noise to restore the original signals, without affecting the signal appreciably. [10]. Filtering also removes the noise, but it affects the signal to a much greater extent. When a signal is captured by sensors, it often has noise due to many reasons like thermal noise, electrical noise, vibration noise and mechanical shock. Also, while transferring the signals other noises may occur. For instance, the connection between the Raspberry Pi and the A/D converter is based on the I2C protocol. This protocol may consist of noises due to electromagnetic interference (EMI) and crosstalk. Crosstalk occurs when signals from one device interfere with the signals from another device. These noises need to be removed before further processes. Through edge computing, this task can be done locally and in real-time, which guarantees not only the quality and accuracy of data, but also easing the data transfer.

The denoising method embedded in the interface system is the fuzzy thresholding method. In fuzzy thresholding, the threshold value is not fixed but is determined using fuzzy logic. Fuzzy logic is a mathematical system that can handle situations where there is no clear distinction between two classes and deals with uncertainty and imprecision. In fuzzy thresholding, the goal is to divide the signal into two classes - the background and the foreground - based on a threshold value. The background represents the parts of the signal that are not of interest, while the foreground represents the parts of the signal that contain the relevant information. In fuzzy thresholding, the threshold value is determined based on the degree of membership of each data point in the foreground or the background classes. This degree of membership is determined using a membership function that assigns a degree of membership to each data point based on its similarity to the foreground or background. Once the degree of membership of each data point is determined, a threshold value is set based on the degree of membership of the data points. Data points with a degree of membership above the threshold are classified as foreground data points, while data points with a degree of membership below the threshold are classified as background data points [11].

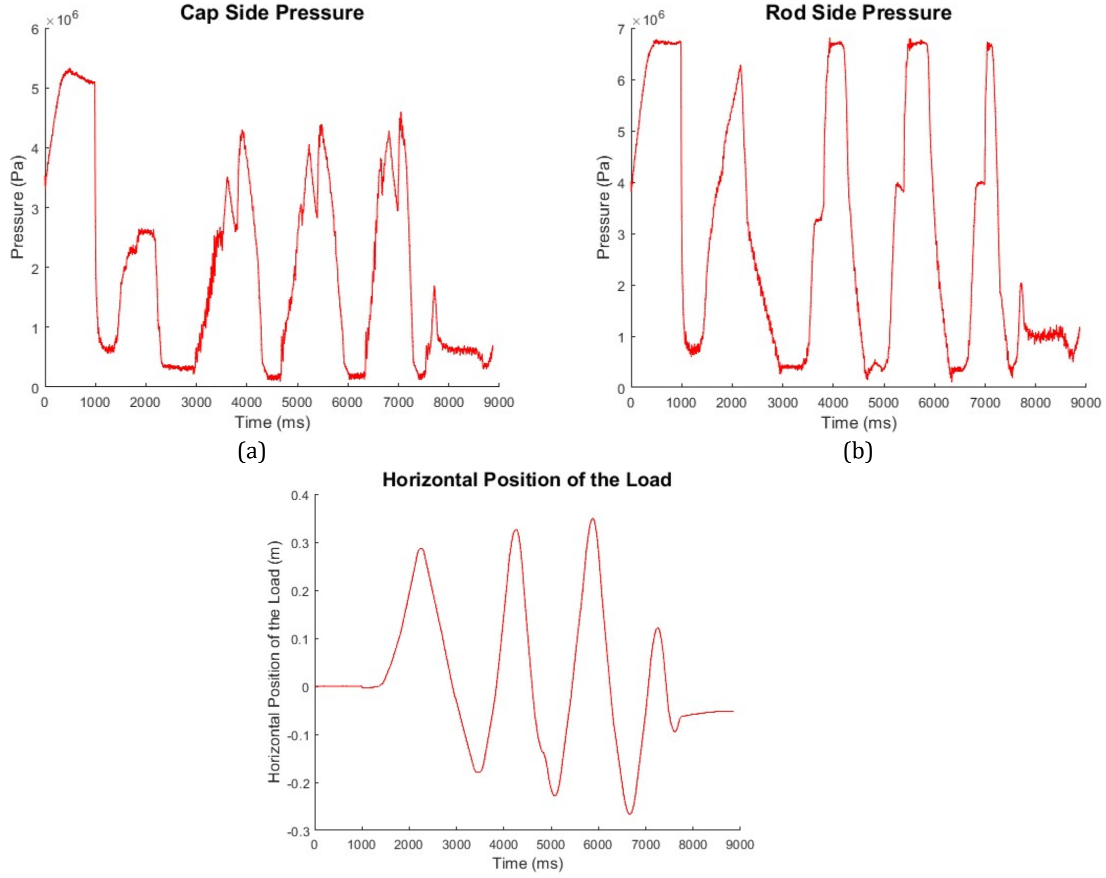
Fuzzy thresholding can be used for denoising by reducing the effects of noise in a signal and enhancing the relevant information. By using fuzzy logic to set the threshold value, fuzzy thresholding can handle situations where the signal has complex and varying characteristics, resulting in a clearer and more accurate signal segmentation which has been used in the interface system.

7. Results

The results are considered with respect to two features: (i) is the delay of the system satisfying the real-time requirements or not, and (ii) is the data plotting for better representation of the data satisfactory or not. In real-time operations, the system delay is a crucial factor that affects the reliability of the data. The length of time it takes for data to be processed and displayed has a direct impact on the accuracy of the results. There are multiple stages involved in the process, from the moment the sensors capture the signals to the time when the data is received by the PC. During these stages, the data undergo processing, leading to unavoidable delays. The sum of all these delays constitutes the total delay of the system. This total delay should be less than 5 milliseconds which is the system's real-time bottleneck.

The initial stage of transferring data from sensors to the interface system occurs nearly instantaneously and incurs minimal delay. The subsequent step of processing the data is also swift. The next stage involves transferring the data from the interface system to the PC. This stage also introduces a negligible delay in the data transfer process. The entire system operates within a delay of much less than 5 milliseconds, which is the response time of the pressure sensors and is considered a **hard real-time requirement**. This ensures that the data remain up-to-date during the entire transfer process, and the excavator condition can be observed in hard real-time.

The final step is plotting the data through the PC. This introduces one second step time to the system which means the PC updates the plots each 1 second. This step has been added for the operator's convenience. Displaying only numbers on the screen would not be very practical for operators. Although, transfer of the data to the PC happens in real-time and the operator has access to the data in real-time. For monitoring the performance of the system, the real-time plotting in MATLAB is useful. It can be replaced by other approaches. Figure 4 shows the plots which can be viewed by the operator.



(a)

(b)

(c)

Fig. 4: (a) Cap side pressure sensor plot in the PC. (b) Rod side pressure sensor plot in the PC. (c) Horizontal position of the load in the PC.

8. Conclusions

The interface system presented in this paper has played a key role in the successful acquisition, processing, and transmission of high-quality data to a server. Its ability to transfer data from the sensors in less than 5 milliseconds is a significant achievement, as this response time represents the timing bottleneck of the entire system in hard real-time applications. The precision and accuracy of the data displayed as plots on a remote computer are also noteworthy and can be attributed to the effective data processing within the interface system. As demonstrated, the interface system provides valuable information for operators, indicating the system condition, also sends hard real-time data to the PC. Overall, the interface system has proven to be a reliable and effective component of the data acquisition and transmission process. There are several potential areas for improvement in the interface system. A potential improvement includes a replacement of the current wired Ethernet LAN connection between the interface system and the PC with a wireless connection. This change could extend the device's range and improve the system's ability to send data to remote servers.

References

[1] Navid Niksefat, Nariman Sepehri, and Qiong Wu, *"Design and experimental evaluation of a QFT contact task controller for electro-hydraulic actuators,"* International Journal of Robust and Nonlinear Control: IFAC-Affiliated Journal, vol. 17, no. 2-3, pp. 225-250, 2007.

[2] Gustavo K. Costa and Nariman Sepehri, "*Hydrostatic transmissions and actuators: operation, modelling and applications."* John Wiley & Sons, 2015.

[3] Guangan Ren, Gustavo K. Costa, and Nariman Sepehri, "Position control of an electro-hydrostatic asymmetric actuator operating in all quadrants," *Mechatronics*, vol. 67, p. 102344, 2020.

[4] Ali Maddahi, Witold Kinsner, and Nariman Sepehri, "Internal leakage detection in electrohydrostatic actuators using multiscale analysis of experimental data," I*EEE Transactions on Instrumentation and Measurement*, vol. 65, no. 12, pp. 2734-2747, 2016.

[5] Gustavo Koury Costa and Nariman Sepehri, "Four-quadrant analysis and system design for single-rod hydrostatic actuators," J*ournal of Dynamic Systems, Measurement, and Control*, vol. 141, no. 2, 021011, 2019., vol. 3, pp. 771-775.

[6] M. Karpenko and Nariman Sepehri, "On quantitative feedback design for robust position control of hydraulic actuators," *Control Eng. Practice*, vol. 18, pp. 289-299, 2010.

[7] Alireza Akhoondi Asadi, Shahriar Bagheri, Ahmed Imam, Ehsan Jalayeri, Witold Kinsner, and Nariman Sepehri, "A data acquisition system based on Raspberry Pi: Design, construction and evaluation," In *Proc. 2016 IEEE 7th Annual Information Technology, Electronics and Mobile Communication Conference*, Vancouver, BC, Canada, pp. 1-5, 2016.

[8] Jonathan Valvano, *Embedded Microcomputer Systems: Real Time Interfacing*. Pacific Grove, CA, USA: Brooks/Cole, 2000.

[9] Witold Kinsner, Microcontroller, Microprocessor and Microcomputer Interfacing for Real-Time Systems. Winnipeg, MB: OCO Research, Sep 2020, 973 pages. {ISBN: 978-0-9939347-5-9, eBook}[10] T. F. Sanam and C. Shahnaz, "A semisoft thresholding method based on Teager energy operation on wavelet packet coefficients for enhancing noisy speech," *EURASIP Journal on Audio, Speech, and Music Processing,* vol. 2013, article 25, 2013.

[11] Amirreza Mirbeygi Moghaddam, Fault Detection and Fault-tolerant Control of Single-rod Electrohydrostatic Actuated System" MSc Thesis, Winnipeg, Manitoba: University of Manitoba MSpace, pp. 20-31, 2021, 135 pages.
https://mspace.lib.umanitoba.ca/xmlui/bitstream/handle/1993/35816/Thesis-Amirreza-Final.pdf?sequence=4&isAllowed=y

[12] K1 ASHCROFT, K1 Pressure Transmitter. Datasheet, Sep 3, 2020. Retrieved from: https://www.ashcroft.com/wp-content/uploads/2020/09/datasheet-k1-transmitter.pdf

[13] Bourns, Rotary Optical Encoder. Datasheet, Jun 8, 2011. Retrieved from: https://www.bourns.com/docs/default-document-library/enc1j.pdf?sfvrsn=2c0f9df1_0

[14] Texas Instruments, ADS111x Ultra-Small, Low-Power, I 2C-Compatible, 860-SPS, 16-Bit ADCs
with Internal Reference, Oscillator, and Programmable Comparator, ADS1115. Datasheet. Retrieved from: https://www.ti.com/lit/gpn/ADS1115

[15] Raspberry Pi, Raspberry Pi4 Documentation. Retrieved from: https://www.raspberrypi.org/documentation/hardware/raspberrypi/