Photocopy and Use Authorization

In presenting this thesis in partial fulfillment of the requirements for an advanced degree at Idaho State University, I agree that the Library shall make it freely available for inspection. I further state that permission for extensive copying of my thesis for scholarly purposes may be granted by the Dean of the Graduate School, Dean of my academic division, or by the University Librarian. It is understood that any copying or publication of this thesis for financial gain shall not be allowed without my written permission.

Signature: _____________________________

Date: __________________________________

Development of Control System Implementation Platform in Advanced Manufacturing

by

Golam Gause Jaman

A thesis

submitted in partial fulfillment

of the requirements for the degree of

Master of Science in the Department of Mechanical Engineering

Idaho State University

December, 2023

Committee Approval

The members of the committee appointed to examine the thesis of **Golam Gause Jaman** find it satisfactory and recommend that it be accepted.

Dr. Marco P. Schoen

Dr. Anish Sebastian

Dr. Nancy Devine

Acknowledgement

I would like to express my heartfelt gratitude to my advisor Professor Marco Schoen, for keeping patience with me while constantly providing me with guidance and insightful feedback. His support is critical in shaping my research work and inspiring me to pursue this study. I would like to thank the Department of Mechanical Engineering and all the faculty members who are always been there to support my study and address my questions and concerns during my research. I am thankful to all the members of the Measurement and Control Engineering Research Center.

Table of Contents

List of Figures

List of Tables

Development of Control System Implementation Platform in Advanced Manufacturing

Thesis Abstract - Idaho State University (2023)

This research establishes a control platform for continuous rapid joule heating thermal control in advanced manufacturing applications. The developed prototype utilizes a pair of graphite rollers, integrating modular actuators, precise sensors, and multifaceted components interconnected through either serial RS-485 or analog signal chains to a central National Instruments CompactRIO. Together, these elements enable effective system orchestration and thermal regulation within the range of several hundred degrees Celsius. Closed-loop PID controllers configured using empirical system identification models demonstrate robust setpoint tracking and faster rise times relative to fuzzy logic controllers in simulation case studies. The tuned PID implementation shows less than 5 percent overshoot and 5 percent steady-state error when tracking 200 degrees Celsius setpoint commands in the actual experimental system. The controller continuously modulates power input levels based on the disparity between temperature sensor data from analog inputs and predetermined targets. Insights into the intricate transient relationships are derived from first-principles grey-box system modeling, guiding controller synthesis using MATLAB®tools. The front-end interface acquires rich feedback on process critical parameters like flow rates, current distribution, voltages, device states, and most critically, graphite roller surface temperatures while the back-end seamlessly coordinates automated sequential control actions through RS-485/Modbus communication. In conclusion, the integrated solution showcases an instrumentation and control approach to transition specialized rapid joule heating setups towards continuous process compatibility. This enhances flexibility for small-scale advanced manufacturing systems over conventional start-stop, batch-based electric field-assisted platforms. The research thus provides a stepping stone towards unlocking efficiency and product customizability improvements in advanced manufacturing through inventive thermal control techniques.

Keywords: Control System, Advanced Manufacturing, Electric Field Assisted Sintering, Instrumentation, System Identification.

1 Introduction

Advanced Manufacturing encompasses a range of innovative processes and technologies designed to improve efficiency and performance in the production of goods. One of the significant subfields within this domain is additive manufacturing, which has revolutionized the way objects are designed and built, allowing for greater customization and complexity. Sintering, a process integral to this field, involves the application of heat and pressure to powdered material to form solid parts. Traditional sintering relies on heat propagation from the outer surface inward, facilitating boundary diffusion and curing to attain specific material characteristics.

A notable advancement in this process is Electric Field Assisted Sintering (EFAS). EFAS differs from conventional sintering by employing joule heating, which deposits energy at the molecular level, akin to spark plasma sintering. This technique has been praised for its efficiency and the quality of the resulting material.

The work presented in the thesis contributes to innovation within the EFAS domain by introducing a continuous process utilizing a pair of rapid joule heated graphite rollers. This design contrasts the traditional batch process limited by the EFAS machine's operational mechanism and the constraints imposed by the machine's geometry. The key challenge in this work is the temperature control of the rollers, critical to the successful implementation of continuous EFAS.

The research methodology involves several stages, starting with the instrumentation of the continuous EFAS prototype. This stage includes collecting temperature data from the rotating graphite rollers and adjusting crucial parameters such as the voltage supplied to the rollers and the rotation per minute (RPM) of the shaft motor. Additional feedback mechanisms are incorporated to mitigate thermal and electrical hazards.

Subsequent to the instrumentation phase, a central command unit is established using hardware components from National Instruments (NI). This unit is responsible for orchestrating the interconnection between various system components. The interconnection steps are meticulously designed using LabVIEW™, which serves as the platform for controller implementation.

The final stages of the project involve running experiments to gather data, developing a model using MATLAB®, and designing a controller, also in MATLAB®. The most complex aspect of the study is the development of a communication sequence for proper system interconnection, along with the modeling and controller design. The project's novelty lies in its approach to making EFAS a continuous process, a significant leap from conventional batch processing. This advancement could potentially pave the way for more efficient and versatile manufacturing processes in the field. The project and the study are conducted using resources and facilities provided by the Idaho National Laboratory (INL).

2 Literature Review

The field of advanced manufacturing has experienced significant advancements through the development of Electric Field Assisted Sintering techniques, particularly Spark Plasma Sintering (SPS). This section aims to encapsulate the evolution of Electric Field Assisted Sintering, with a specific focus on the development and nuances of SPS. The review section chronicles the historical progression of the technology, highlights key technological milestones, and delves into the current state of the art. Additionally, it provides insights into potential future trends and unexplored areas within this domain. Through comprehensive analysis and synthesis of existing literature, this section serves as a detailed resource for researchers and practitioners in the field of advanced manufacturing.

Electric Field Assisted Sintering represents a significant segment in the realm of advanced manufacturing techniques. This method, which involves the application of an electric field to enhance the sintering process of materials, has evolved over several decades. Its roots can be traced back to early efforts in improving sintering efficiency and material properties [1]. Spark Plasma Sintering, a subset of Electric Field Assisted Sintering, emerged as a breakthrough technique offering enhanced capabilities over conventional sintering methods. Characterized by its rapid sintering process and potential for creating denser and stronger materials, SPS has garnered considerable attention in materials science and engineering [2].

Electric Field Assisted Sintering (EFAS) technology has its roots in the mid-20th century when researchers began exploring ways to enhance sintering processes. Early iterations involved the application of an electric field to powdered materials to facilitate densification at lower temperatures and reduced sintering times compared to conventional methods. These initial endeavors laid the foundation for what would become a diverse and complex field within materials science [3]. As the technology evolved, various methods falling under the umbrella of EFAS were developed. These included techniques such as Hot Pressing (HP), Hot Isostatic Pressing (HIP), and Field Assisted Sintering Technology (FAST), each offering unique advantages and applications. These methods shared a common principle: the use of an electric field or current to enhance the sintering process, but differed in their specific mechanisms and efficiencies [4]. The advent of Spark Plasma Sintering (SPS) marked a significant milestone in the EFAS domain. This technique, introduced in the 1990s, was a notable departure from traditional sintering methods. SPS utilizes a pulsed direct current that passes through the sample, generating a localized plasma and high heating rates. This leads to rapid sintering, reduced grain growth, and enhanced material properties [2]. The technological milestones in SPS are characterized by improvements in equipment design, control systems, and understanding of the underlying mechanisms. Key developments included the refinement of pulse patterns for current delivery, advancements in die materials to withstand higher temperatures, and enhanced control systems for precise temperature and pressure regulation. These improvements allowed for the expansion of SPS applications to a wider range of materials, including ceramics, metals, and composites [5]. SPS distinguishes itself from other EFAS methods in several ways. The rapid heating and cooling rates, achievable only through the pulsed electric current, reduce sintering time significantly. Furthermore, the localized plasma generation contributes to the effective removal of surface oxides and contaminants, leading to higher-density materials. These unique capabilities have positioned SPS as a preferred method for many advanced material synthesis applications [1].

The current state of Spark Plasma Sintering (SPS) is characterized by significant technological advancements and a broadening range of applications. Recent developments have seen the introduction of more sophisticated control systems, enabling precise manipulation of sintering parameters. These advancements have enhanced the ability to tailor material properties to specific applications, resulting in superior performance characteristics [6]. One of the most notable aspects of contemporary SPS technology is its application to a diverse array of materials, including advanced ceramics, high-entropy alloys, and nanocomposites. This versatility has opened new avenues in fields such as aerospace engineering, biomedical implants, and electronic devices. The ability of SPS to maintain the purity and nano-features of materials has been particularly beneficial in the development of high-performance and specialized materials [7]. SPS is increasingly being integrated with other advanced manufacturing technologies, such as additive manufacturing and 3D printing. This integration allows for the creation of complex geometries with enhanced material properties, showcasing the potential of SPS as a complementary technology in modern manufacturing processes [8]. Looking forward, Spark Plasma Sintering is poised for further exploration and development. Potential areas for research include the optimization of sintering parameters for novel materials, exploration of the mechanisms at play during the SPS process, and the development of hybrid techniques combining SPS with other manufacturing processes. These areas hold promise for unlocking new material properties and broadening the application spectrum of SPS [9]. Despite its advancements, SPS faces challenges such as high equipment costs and a limited understanding of certain underlying physical phenomena. Addressing these challenges through research and development could lead to more cost-effective and versatile SPS systems. Additionally, the exploration of environmentally friendly and energy-efficient SPS processes presents a significant opportunity for sustainable manufacturing practices [10]. As advanced manufacturing continues to evolve, SPS is expected to play a pivotal role in the production of high-performance materials. Its ability to rapidly process materials with enhanced properties positions SPS as a key technology in the future of manufacturing, particularly in sectors demanding material excellence [11].

In the current landscape, SPS stands out for its rapid sintering process, ability to maintain material purity, and applicability to a wide range of materials. The integration of SPS with other advanced manufacturing technologies, such as additive manufacturing, has further expanded its application spectrum, demonstrating its versatility and potential in producing complex geometries and high-performance materials. Looking to the future, SPS presents several promising avenues for research and development. While challenges such as equipment costs and incomplete understanding of underlying mechanisms persist, these also represent opportunities for innovation and advancement. The exploration of energy-efficient and environmentally friendly SPS processes is particularly noteworthy, aligning with the global shift towards sustainable manufacturing practices.

The significance of Spark Plasma Sintering in the realm of advanced manufacturing cannot be overstated. As a technology capable of processing a diverse range of materials with enhanced properties, SPS is pivotal in the production of high-performance components for industries such as aerospace, biomedical, and electronics. Its role in future manufacturing is expected to be of crucial importance, especially in sectors where material excellence and efficiency are paramount.

Spark Plasma Sintering emerges not just as a novel manufacturing technique but as a symbol of the ongoing evolution in material science and engineering. Its continued development and integration into various manufacturing processes underscore its potential to revolutionize the production of advanced materials. As the field progresses, SPS will continue to be a focal point of research and application, driving innovation in advanced manufacturing.

3 System Components

In this study, a systematic approach is employed for the development of an integrated system, focusing on the selection of components based on defined objectives and constraints. This is essential to ensure the achievement of the target performance while adhering to safety standards. The central aspect of this research is the expedited Joule heating of a pair of graphite rollers. Owing to the symmetrical material properties and configuration of these rollers, a uniform operational environment is established.

For the purpose of this thesis, the system configuration is elucidated by considering a singular roller to eliminate repetitive descriptions. The specified roller features a geometric configuration with a diameter of 5 inches, a width of 3 inches, and a thickness of 1/4 inches, as depicted in Figure 3.1. Electrical energy is transferred to the roller from the power supply through its contact with curved brushes. The rotation of the roller is facilitated by a brushless DC motor operating at 0.02 rpm.

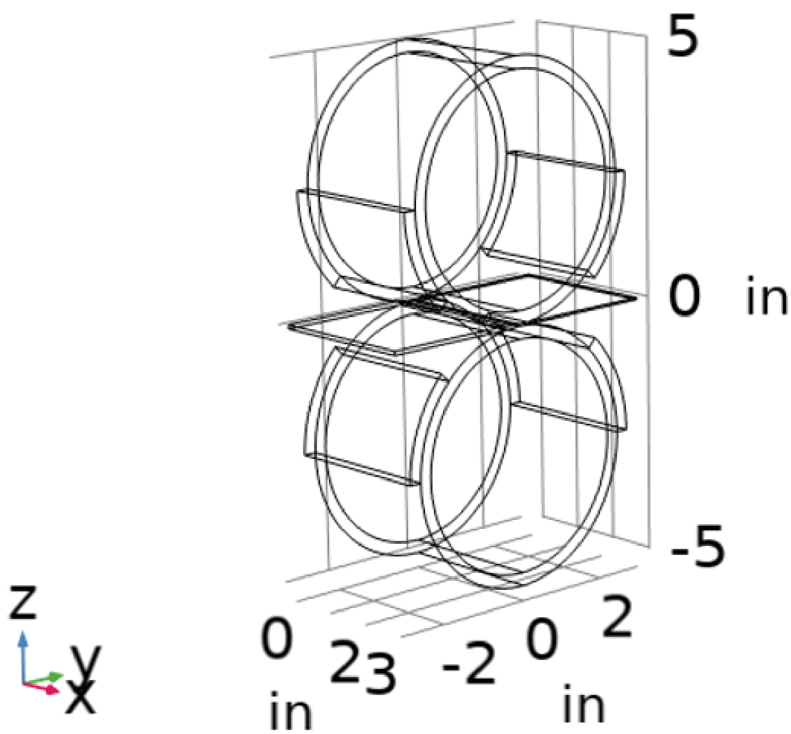


Figure 3.1: The geometric configuration of the roller.

A crucial aspect of the system is the heat management mechanism. This is achieved by driving water axially through the center of the roller at a constant rate using a water pump, effectively serving as a heat sink. Additionally, an air-blowing setup is implemented to induce forced convection around the roller, thereby enhancing the heat dissipation process.

Under the assumption of constant impedance for the roller and negligible power loss, the Joule heating mechanism is modeled as a heat transfer equation incorporating both heat sources and sinks, as formulated in Equation 3.1. Figure 3.2 and 3.3 presents a schematic of the overall system, illustrating the integrated components and their interplay in the heating and cooling processes.

The 3D heat transfer equation with a Joule heating source and Newton's law of cooling is given by:

$$\rho c_p \frac{\partial T}{\partial t} = k\nabla^2 T + Q - hA(T - T_{\text{ambient}}) \tag{3.1}$$

where T is the temperature, ρ is the density, c_p is the specific heat capacity, k is the thermal conductivity, Q is the internal heat generation (Joule heating), h is the heat transfer coefficient, A is the surface area, and T_{ambient} is the ambient temperature.

Temperature regulation and monitoring are crucial aspects of the system's functionality. To this end, the system is equipped with an array of thermocouples and multiple pyrometers. These instruments play a pivotal role in maintaining the desired thermal state and implementing essential safety measures. Additionally, a flowmeter is integrated into the system to monitor the flow of coolant, specifically water, thereby ensuring optimal thermal management.

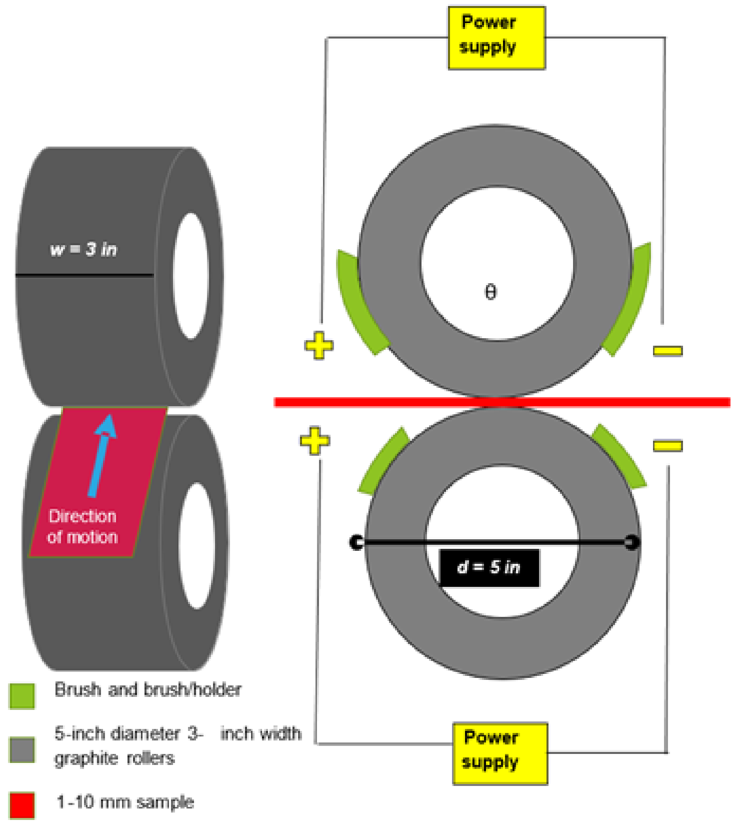


Figure 3.2: The overall system configuration.

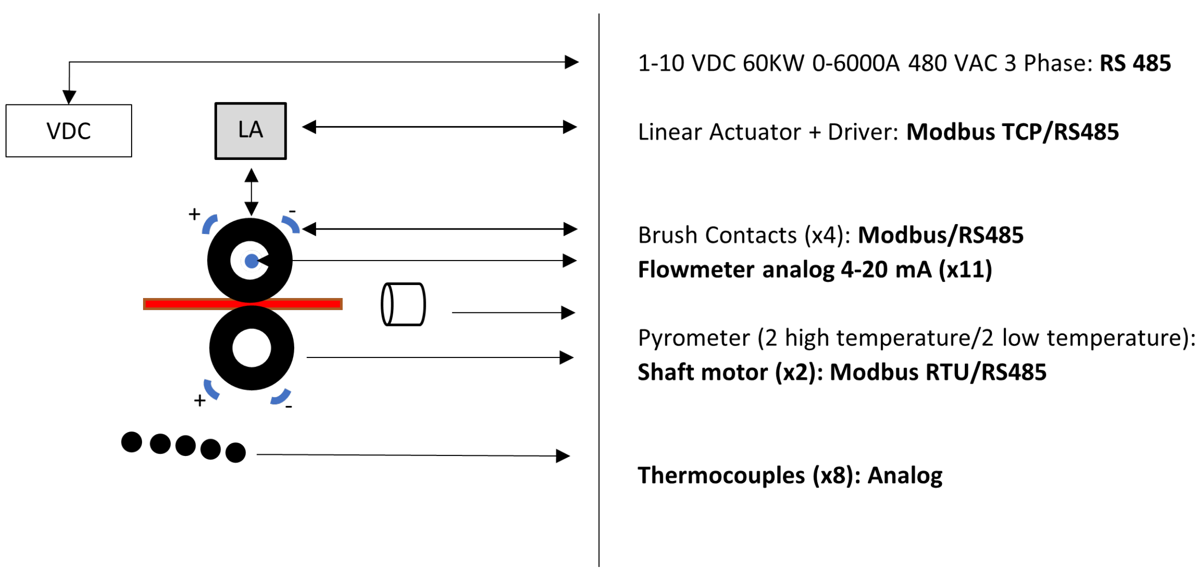


Figure 3.3: The system schematic with integrated components.

The system's actuating and feedback mechanisms are interconnected with a central control station. This integration is achieved through the use of compact and robust hardware, specifically the NI cRIO-9045. This hardware features a 1.3 GHz Dual-Core processor and a 70T FPGA, encompassed within an 8-slot, RT, Non-XT framework. This setup facilitates efficient data processing and system responsiveness.

Communication within the system is streamlined through the RS 485 module of the cRIO, utilizing both MODBUS RTU and MODBUS ASCII protocols. These protocols enable effective command transmission to the shaft motor and the power supply. The intricacies of these communication setups, along with a detailed exploration of each component's capacity and role within the system, are elucidated in the subsequent subsections of this document. The Figure 3.4 shows a cRIO-9045 with preinstalled modules.

Figure 3.4: cRIO-9045 with preinstalled modules.

3.1 cRIO-9045 NI Modules

The NI 9205, NI 9203, and NI 9871 modules collectively offer a versatile array of functions, from high-density analog input and precise current measurement to robust serial communications. Their integration capabilities with CompactRIO and CompactDAQ systems underscore their applicability in a wide range of industrial scenarios, fulfilling various roles in data acquisition and control system architectures.

3.1.1 NI 9205

The NI 9205 is a high-density analog input module designed for multipurpose data acquisition. This module is notable for its high channel count and wide voltage range. It features 32 single-ended or 16 differential analog input channels, with a ±10 V input range, making it suitable for a broad spectrum of industrial sensors and signals. The NI 9205 operates with a 16-bit resolution and a maximum sample rate of 250 kS/s, ensuring precise data acquisition for applications requiring high-speed measurements. Its compatibility with CompactRIO and CompactDAQ systems enhances its utility in diverse measurement scenarios. The Figure 3.5 shows an NI 9205 module.

Figure 3.5: NI 9205 module

3.1.2 NI 9203

The NI 9203 is a current input module, designed to accurately measure current signals. It provides 8 channels with a measurement range of 0 to 20 mA, which is a common range for industrial sensors and transmitters. The module's 16-bit resolution and 200 kS/s maximum sample rate make it well-suited for monitoring current-driven sensors in real-time. The NI 9203's integration into CompactRIO and CompactDAQ systems allows for flexible deployment in various industrial environments, especially in applications involving process monitoring and control. The Figure 3.6

shows an NI 9203 module.

Figure 3.6: NI 9203 module

3.1.3 NI 9871

The NI 9871 is a serial module designed for communication and data exchange. This module features 4 RS232 serial ports, supporting baud rates up to 1,000,000 baud. The NI 9871's capabilities extend to applications requiring robust serial communications, such as interfacing with barcode scanners, printers, and other RS232 devices. Its seamless integration with CompactRIO systems enables it to serve as a bridge for legacy equipment in modern control systems, thereby ensuring the compatibility of various devices within an industrial setup. In this work, the NI 9871 modules are used for RS 485 serial communication in simplex mode or two-wire mode for communicating with the power supply using MODBUS ASCII and the shaft motor driver using MODBUS RTU. The Figure 3.7 shows an NI 9871 module.

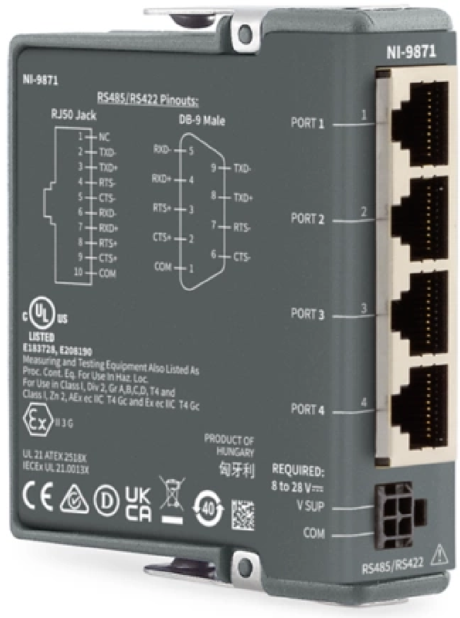


Figure 3.7: NI 9871 module

3.2 Magna Power 10-6000 TSD

The Magna Power TSD 10-6000 DC power supply, part of the TS Series, is designed to offer a broad range of voltage and current capabilities while maintaining a high power density within its rack-mount packaging. This series spans voltages from as low as 5 Vdc up to a substantial 6000 Vdc and can handle current levels ranging from 1.2 Adc to a significant 8000 Adc.

The TSD10-6000 stands out for its capacity to offer solutions for specific low voltage but high current demands, presenting an option for such requirements. This power supply series is equipped with a standard isolated 37-pin external I/O, RS232 remote interface software, IVI drivers, and LabVIEW™drivers, enhancing its adaptability and ease of integration into a variety of programming environments. The inclusion of these features makes the TSD10-6000 a highly flexible and efficient choice for a wide range of applications requiring precise power control and distribution. In the present work the power supply with a RS 485 - RS 232 converter is used to communicate with the NI 9871 over MODBUS ASCII protocol. Figure 3.8 shows a TSD 10-6000 DC Power Supply module.

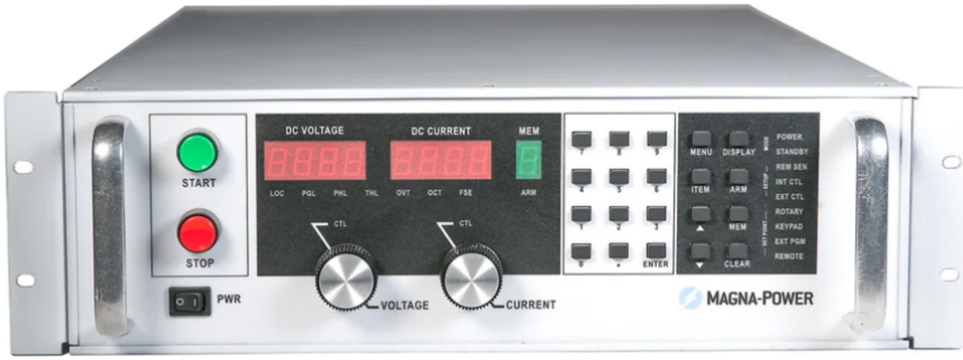


Figure 3.8: TSD 10-6000 DC Power Supply module

3.3 Fluke Endurance Pyrometers

The Fluke Endurance series pyrometers are advanced infrared based noncontact temperature measurement systems designed for high accuracy and reliability in industrial settings. These devices are equipped with an adjustable focus, through-the-lens sighting capability, and a parallax-free optical axis, making them precise in measuring the heat energy emitted from objects. The energy is then converted into an electrical signal, which the pyrometer measures. Each model in the Endurance series functions as an integrated temperature measurement subsystem, comprised of optical elements, spectral filters, detectors, and digital electronics, all housed within a sealed unit. They are built to operate continuously, with a 100 percent duty cycle, in challenging industrial environments.

The series includes monochrome models (one-color) designed for standard temperature measurement applications. The 1-color mode excels in situations where the target area is free from any sighting obstructions, either solid or gaseous, and where the target completely fills the measurement spot. Additionally, the series offers ratio models (two-color) for more specific temperature measurement applications. These ratio pyrometers determine the object's temperature by analyzing the ratio of two separate and overlapping infrared bands. The two-color mode is particularly effective for measuring the temperature of targets partially obscured by other objects, openings, screens, or viewing windows that reduce energy, as well as by dirt, smoke, or steam in the atmosphere. These pyrometers can measure and determine the object temperature in both one-color and two-color modes, with two infrared detectors active in each case, offering flexibility and adaptability for a wide range of measurement scenarios. Figure 3.9 shows a Fluke Endurance Pyrometer.

Figure 3.9: Fluke Endurance Pyrometer

3.4 SV4614/5614 Temperature and Flowmeter

The SV4614/5614 is a flowmeter designed to measure the temperature and flow rate of fluids. This device is adept at providing continuous monitoring of fluid flow in various applications, ranging from industrial processes to laboratory settings. Its dual functionality allows it not only to measure the flow rate of the fluid passing through it but also to gauge its temperature. This makes the SV4614/5614 particularly valuable in scenarios where both the flow and thermal characteristics of the fluid are critical to the process or system's operation. The SV4614/5614 is designed to integrate seamlessly into various control systems, making it a versatile choice for those needing comprehensive fluid monitoring solutions. Figure 3.10 shows an SV4614/5614 Temperature and Flowmeter module.

Figure 3.10: SV4614/5614 Temperature and Flowmeter module

3.5 Thermocouple

A K-type thermocouple is a widely utilized sensor for temperature measurement, known for its reliability and extensive temperature range capability. Constructed from nickel-chromium (NiCr) and nickel-alumel (NiAl) alloys, with NiCr as the positive leg and NiAl as the negative, these thermocouples are adept at handling temperatures ranging from approximately -200°C to +1350°C (-328°F to +2462°F). Their hallmark lies in their accuracy and stability, especially at high temperatures, making them a reliable choice for continuous and long-term applications. Commonly employed in high-temperature industrial processes such as heat treating, kilns, gas turbine exhaust, and diesel engines, K type thermocouples generate a small voltage at their junction when heated, which is proportional to the temperature. This voltage is then measured and translated into a temperature reading. Their cost-effectiveness, coupled with a broad operational temperature range and durability, contributes significantly to their widespread u se. Furthermore, their compatibility with various measurement instruments, including digital thermometers, data loggers, and process controllers, facilitates easy integration into a variety of systems, thereby enhancing their applicability across numerous industrial and scientific s ettings. Figure 3.11 shows a Type-K t hermocouple. In this work, additional treatment is considered to calibrate the thermocouple using the cold junction temperature.

Figure 3.11: Type-K thermocouple

3.6 BLMR6200SKM

The BLMR6200SKM motor, part of the BLV Series R Type, is a brushless motor suitable for a variety of industrial and automation applications such as robotics and precision motion control. These brushless motors are characterized by smoother operation, extended lifespan, and reduced maintenance needs compared to traditional brushed motors. The specific technical details, including power rating, speed, torque, and other critical specifications, are detailed in the operating manual, specifically referenced as HP-5142.

Regarding the BLVD-KRD motor drive, while specific details are not available, it is typically a motor drive designed to be compatible with brushless motors like the BLMR6200SKM. Such drives usually feature capabilities for speed and torque control and are equipped with communication interfaces for system integration, often including protocols like RS-485 or MODBUS. The drivers also incorporate various protection features to ensure safe operation, guarding against overcurrent, overvoltage, and thermal overload. Figure 3.12 shows the shaft motor and driver interconnection to the host system.

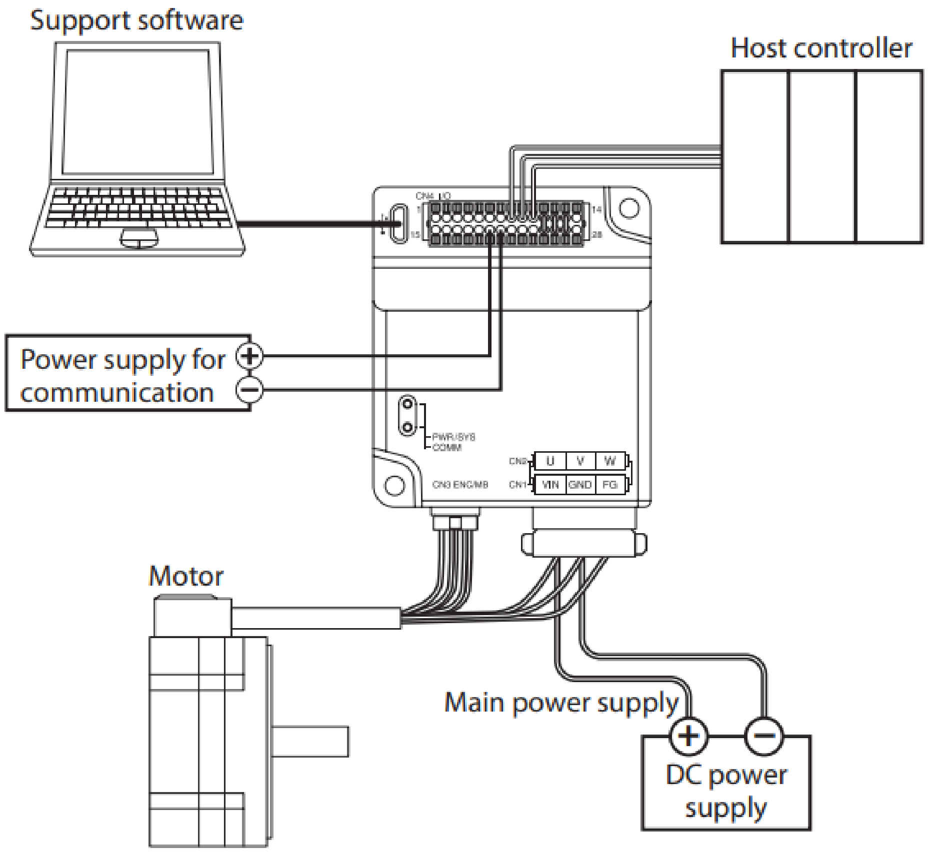


Figure 3.12: Shaft motor and driver interconnection to the host system.

3.7 System Components Interconnection

The actuating modules such as power supply and the shaft motors are connected to NI 9871 for communication over RS 485 using MODBUS ASCII and MODBUS RTU protocol respectively. The temperature feedback from the pyrometers or thermocouples and flowrates/temperature of the coolant are sent in terms of analog signals to the NI 9203 or NI 9205 module based on current/voltage readings. Figure 3.13 shows the interconnection of the system components without the host station. The Table 3.1 summarizes the overall components utilized in developing the instrumentation of the system.

Table 3.1: Components and Their Specifications

Components	Power Supply	Communication Mode	Response Time
TSD10-6000/480+HS+LXI+WC	380 VAc 3-Phase	RS 232/Analog/RS-485	
SV4614/5614	18-30 V DC	Analog	1s
Fluke Endurance	20-48 VDC 12 W	RS 485 / Analog	2ms - 20ms
BLMR6200SKM/BLVD-KRD	24-48 VDC	RS 485 / Modbus RTU	
K type - Thermocouple			

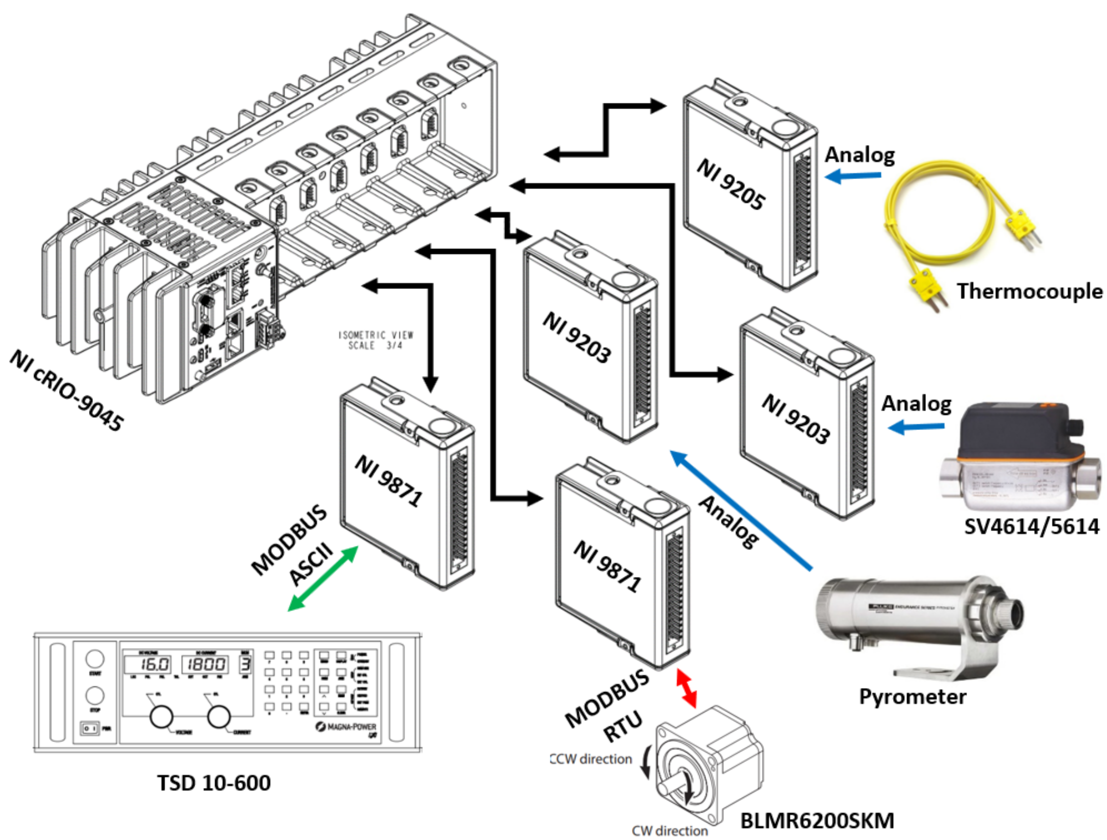


Figure 3.13: Interconnection of the system components.

4 Methodology

The study revolves around the investigation of a specialized system designed to facilitate effective Joule heating, with specific applications in advanced manufacturing, including Electric Field Assisted Sintering (EFAS). The primary aim of this research is to establish a robust temperature regulation process within a substantial range of several hundred degrees Celsius, utilizing the components constituting the system, while concurrently addressing critical safety considerations.

The system comprises several key components, each playing a vital role in its overall functionality. These components include:

- The core of the system involves an electrical energy input mechanism that provides the necessary voltage to induce Joule heating within the setup.
- To control the rotational dynamics and thermal distribution within the system, a shaft motor is employed, contributing the operational efficiency.
- Incorporated for real-time feedback, temperature sensors are strategically placed within the system to monitor and help regulate temperature levels throughout the operation.
- To prevent thermal hazards and ensure efficient energy dissipation, the system features an integrated cooling mechanisms. Notably, both water cooling and air blowing systems are incorporated, although they operate without active control intervention.
- Integral to the system, graphite rollers are employed due to their thermal and electrical conductivity properties. The application of voltage across the system initiates current flow through these rollers, assuming low impedance within other system components. The Joule heating phenomenon manifests as a result of several contributing factors, including current density, material conductivity, roller geometry, and voltage levels.

The primary challenge lies in rapidly elevating the temperature of the rollers by several hundred degrees Celsius while maintaining precise temperature regulation at a predetermined setpoint.

This task is achieved through the integrated cooling mechanisms, which facilitate continuous heat dissipation from the rollers during system operation. Figure 4.1 illustrates the block diagram of the control system, highlighting the architecture. The power supply unit features built-in protection mechanisms with over-voltage and over-current trip settings. Additionally, a Central Command System (CCS) serves as the central regulatory hub for supply control, enforcing stringent safety limits, and ensuring communication between system components. The CCS underwent rigorous testing to assess communication reliability, response time, accuracy, and overall impact on system performance.

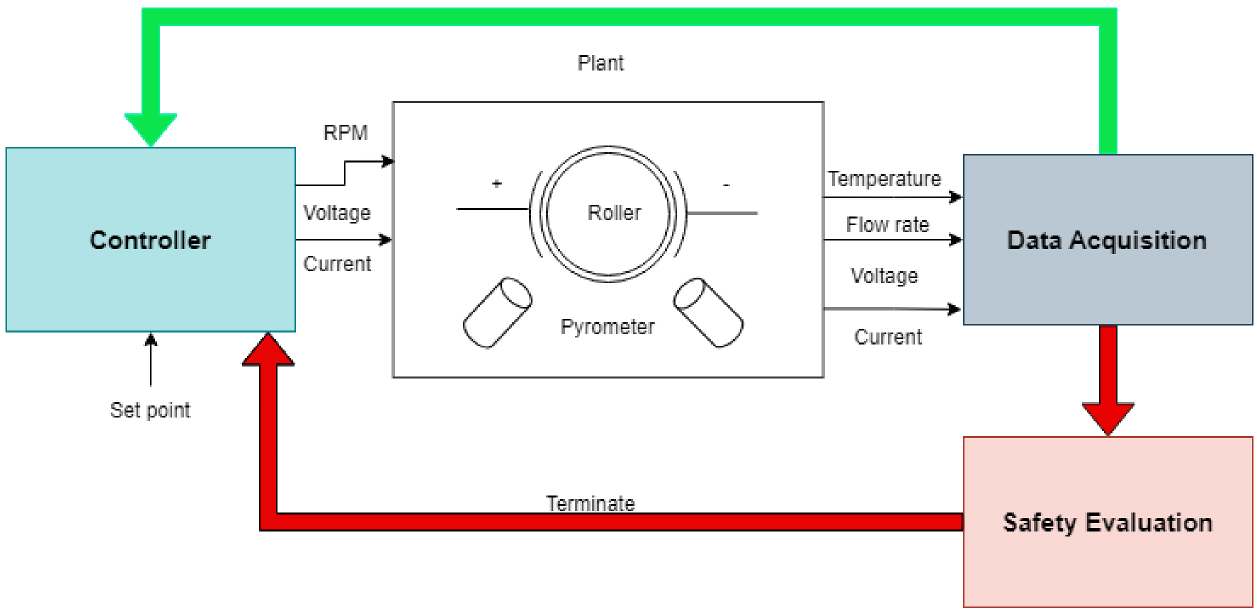


Figure 4.1: Block diagram of the control system.

In the pursuit of safety and system efficacy, temperature emerges as a critical state variable, closely monitored throughout the operation. Furthermore, the flow rate of the water cooling system is observed to mitigate potential thermal hazards and facilitate optimal heat dissipation. Sensors are systematically calibrated to ensure precise data acquisition. The control strategy employed in this study relies on the data collected during experiments involving variable voltage inputs and the subsequent recording of temperature responses over time. The system is treated as a Single-Input Single-Output (SISO) process, enabling System Identification (SID) and transfer function model estimation. A Proportional-Integral-Derivative (PID) controller is fine-tuned based on the derived model, optimizing the system's ability to regulate temperature effectively. Furthermore, a fuzzy logic gain scheduler is implemented to enhance control system performance. The controller's functionality is simulated and tested for overall performance, ensuring its efficacy in real-world applications. Figure 4.2 provides an overview of the controller's implementation process.

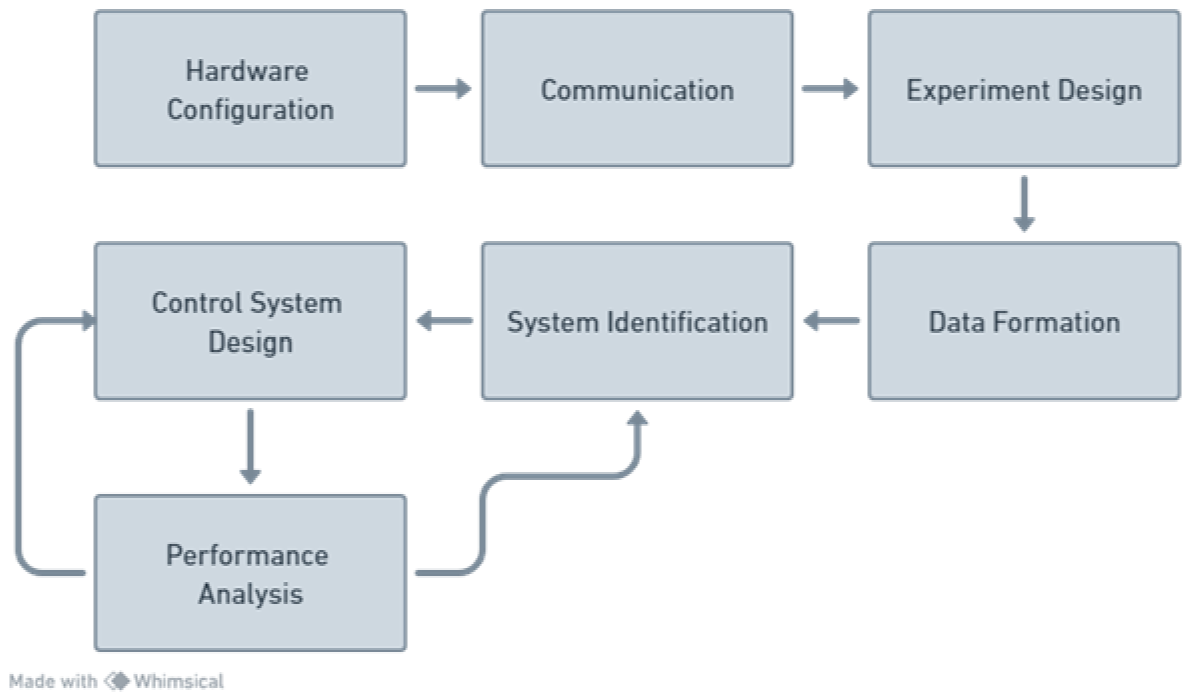


Figure 4.2: The control system implementation process.

5 Communication

5.1 Graphical Code and Virtual Instrument

Laboratory Virtual Instrument Engineering Workbench (LabVIEW™) is a graphical programming platform used by engineers and scientists to develop sophisticated measurement, test, and control systems. In LabVIEW™, programs are called virtual instruments (VIs) because their appearance and operation imitate physical instruments.

VIs consist of two main components: a front panel which serves as the user interface, and a block diagram containing the graphical source code. The front panel can be designed with controls and indicators like knobs, buttons, and graphs. In contrast to text-based languages, the block diagram coding is done by connecting different graphical nodes.

Several key programming concepts shape the development of solutions in LabVIEW™. Execution depends on structure rather than sequence, with multiple nodes able to execute simultaneously enabling inherent parallelism. Iterative structures are facilitated through while loops and for loops. SubVIs act as modular routines similar to functions. Case structures provide conditional branching like if/else statements. Data storage management utilizes arrays and clusters.

Functionality for data acquisition, instrument control, measurement analysis, and other domains is enabled through LabVIEW™APIs and toolkits. Deployment targets for applications in LabVIEW™range from the development computer to dedicated hardware like PXI systems or microprocessors.

LabVIEW™is known for quick development cycles, easy hardware integration, and built-in visualization. Drawbacks compared to text-based languages include poorer code reuse and increased difficulty with debugging. In summary, LabVIEW™provides a unique graphical approach to accelerate test and measurement application development but has a learning curve to master efficient coding practices.

5.2 First-in-first-out (FIFO)

First-in-first-out (FIFO) is a common buffer data structure utilized in Laboratory Virtual Instrument Engineering Workbench (LabVIEW™) programming for regulated data transfer between execution loops and processes.

FIFOs temporarily store data in the order it arrives, emitting the oldest, first-stored data first. They act as a constrained queue, blocking new data when full. Their first-in-first out behavior enables seamless data handoff, critical for coordinated parallel execution. FIFO access matches data production to consumption rates, preventing data loss when these processes proceed at different speeds. As synchronization mechanisms, FIFOs also facilitate safe data exchange avoiding complications like race conditions or stale reads.

In a typical implementation, a FIFO buffer resides in a producer loop, collecting data until read asynchronously via indexing by the consumer process. The reading loop can run at an independent rate given sufficient FIFO capacity. This decouples the production and consumption rates. Common applications taking advantage of FIFOs include communication between program modules or hardware interfaces, streaming substantial volumes of sensor measurements, and data handoff between real-time and user interface environments.

Utilizing FIFOs appropriately requires balancing production/consumption ratios and setting adequate buffer capacity to avoid blocking write operations. Due to their prevalence in measurement and control systems, LabVIEW™includes FIFO constructs as built-in data-sharing enablers between parallel execution processes. When coordinated data throughput is needed, FIFO implementation provides an efficient synchronization technique across independent loops.

5.3 FPGA VI

The NI cRIO-9045 control system leverages NI 9871 for RS 485 communication, NI 9203, and NI 9205 for analog sensor readings. The installed cRIO modules are configured for FPGA-based LabVIEW™programming. Hence, all analog sensors and RS 485 communications are defined in FPGA virtual instruments (VI). Analog signals undergo analog-to-digital conversion (ADC) via appropriate blocks, with readings transformed to match desired temperature and flow rate units. Figure 5.1 shows a portion of the ADC signal distribution defined under FPGA mode.

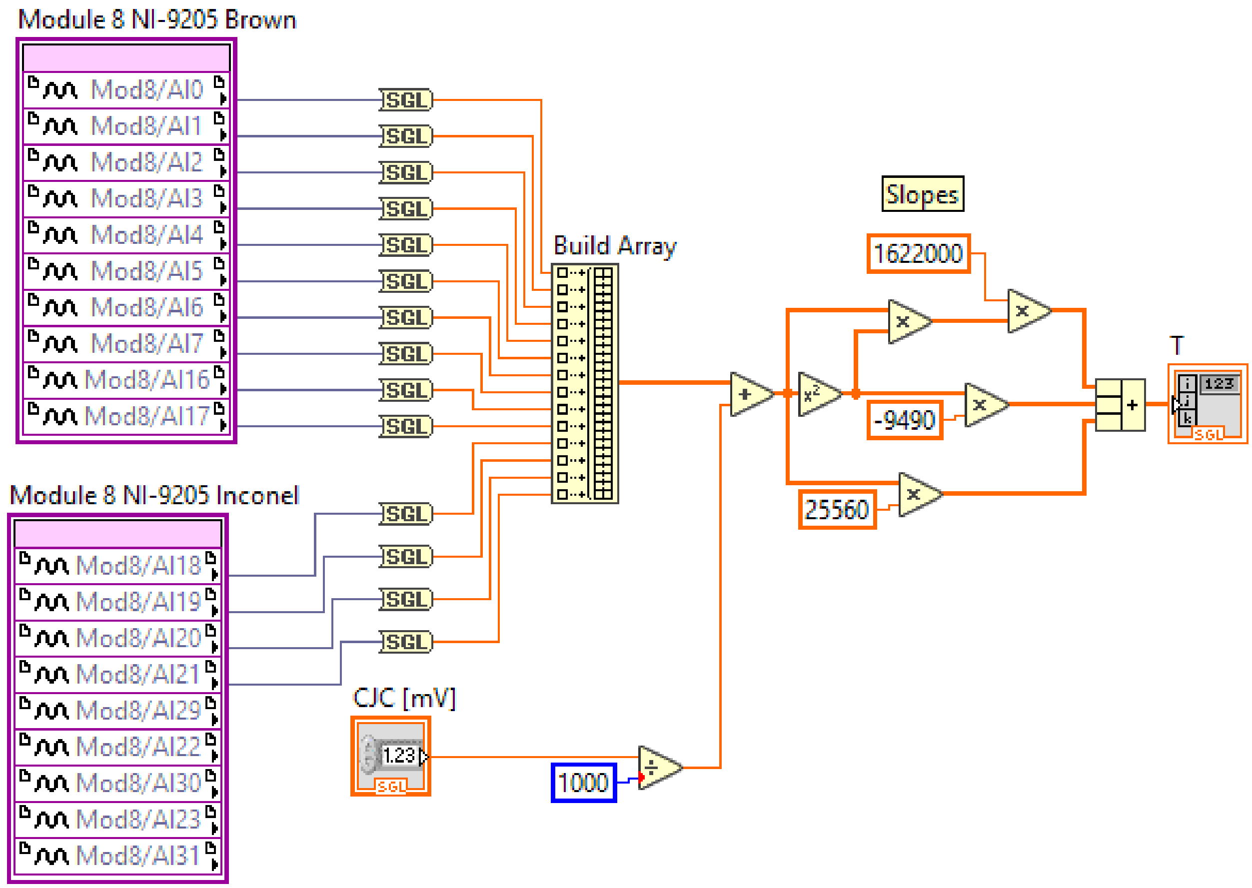


Figure 5.1: A portion of the ADC signal distribution.

FPGA VI constructs utilize a restricted VI library subset compared to the non-FPGA, host VI mode. This results from the 40MHz clock cycle limit constrained computational capacity, and ability to generate standalone control commands. Thus, RS 485 communication leverages basic blocks available in FPGA mode, configured for a two-wire duplex needing only RX and TX channels. Data read blocks get bytes from the physical port into a buffer, while data write blocks take the next buffer byte to the port. Read and write data are respectively processed and generated in the host VI. Stacks of identical routines address each NI 9871 port separately. Figure 5.2 shows a portion of the FPGA VI for the two-wire duplex RS 485 routine.

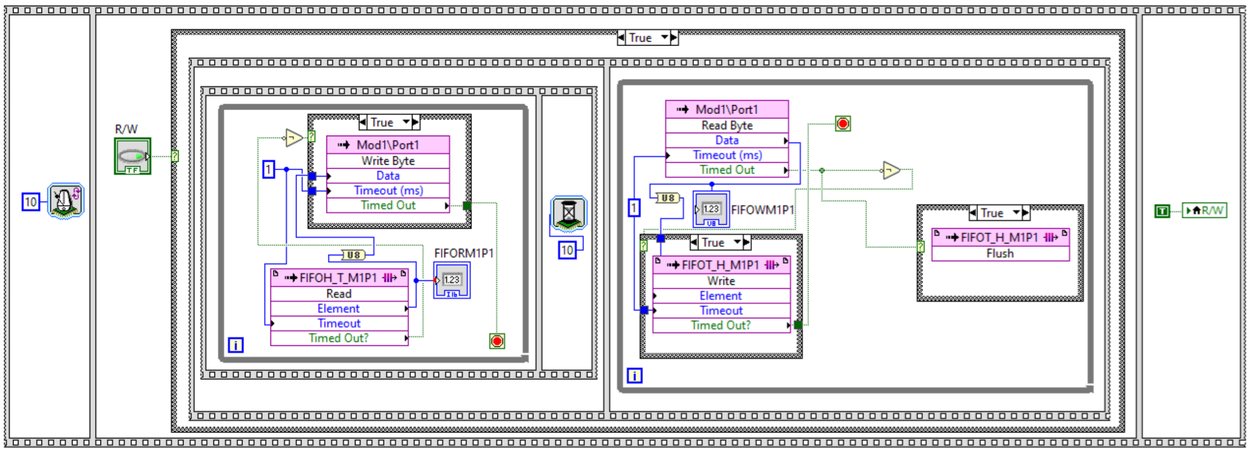


Figure 5.2: A portion of the FPGA VI for the two-wire duplex RS 485 routine.

5.4 HOST VI

The host VI contains sequences similar to the FPGA VI with inverted roles. First-in-first-out (FIFO) buffers exchange bytes between the host and FPGA blocks. The host VI draws from the extensive LabVIEW™library. Bytes created from user/controller instructions are sent to the FPGA through FIFOs to reach physical ports. Equally, FIFOs read incoming bytes from the FPGA to assess feedback conditions. This cascades across all components slated for modulation. Additionally, flags and states dictate safe sequences. Figure 5.3 shows a snippet of the HOST VI.

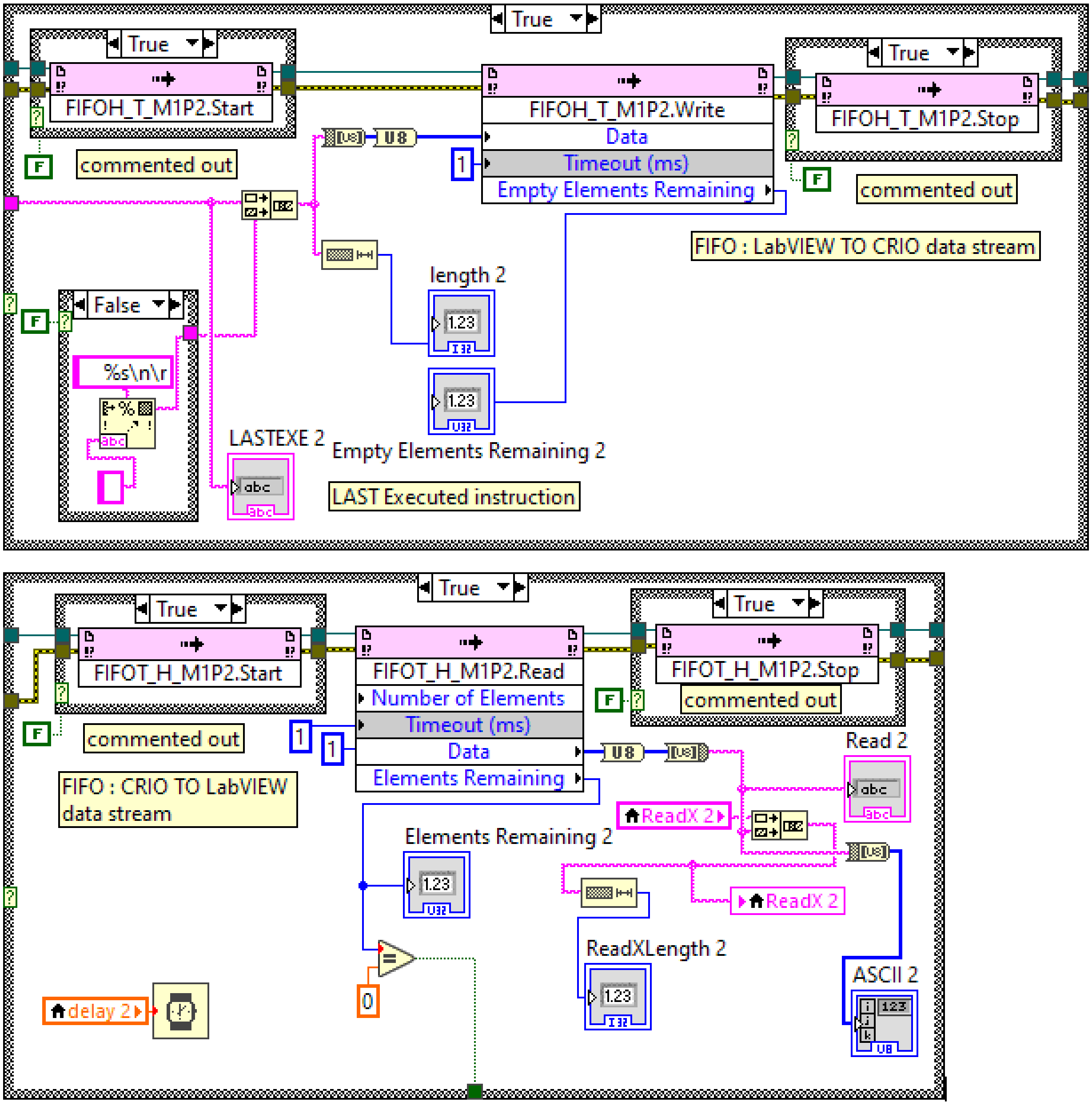


Figure 5.3: A snippet of the HOST VI.

Instructions for the power supply module leverage MODBUS ASCII using directly readable string commands. However, the shaft motor uses MODBUS RTU, needing framed data with mode, register addresses, payload, and CRC 16. Figure 5.4 shows the sample RTU framing. The host VI has dedicated CRC 16 routines, with the computation steps depicted in Figure 5.5.

Slave address	Function code	Data	Error check
8 bits	8 bits	Nx8 bits	16 bits

Figure 5.4: MODBUS RTU frame

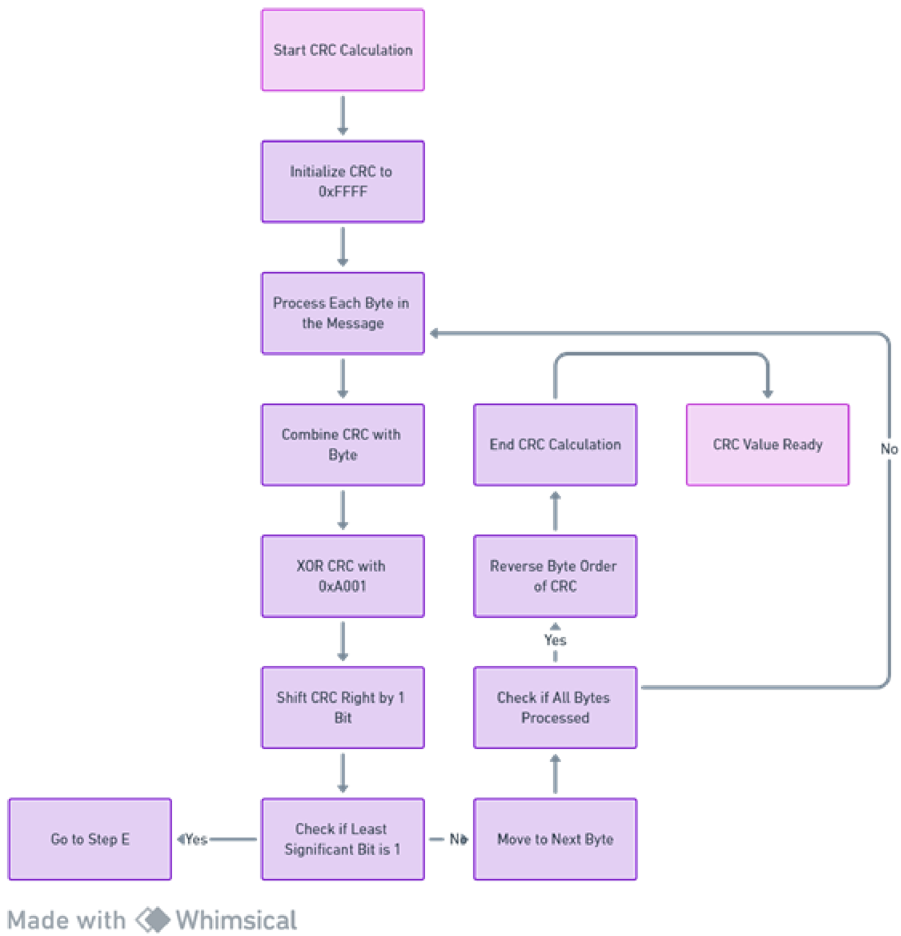


Figure 5.5: CRC 16 Computation Steps

The user interface enables manual command sequences for proper operation. The power supply and motor steps all wait for feedback, enabling redundant safety. Manual test commands assist in debugging during operation. A single switch swaps manual mode for autonomous control. Figure 5.6 shows the front panel segment. The instructions are formatted with the autonomous control placeholder. The controller is discussed in the later section.

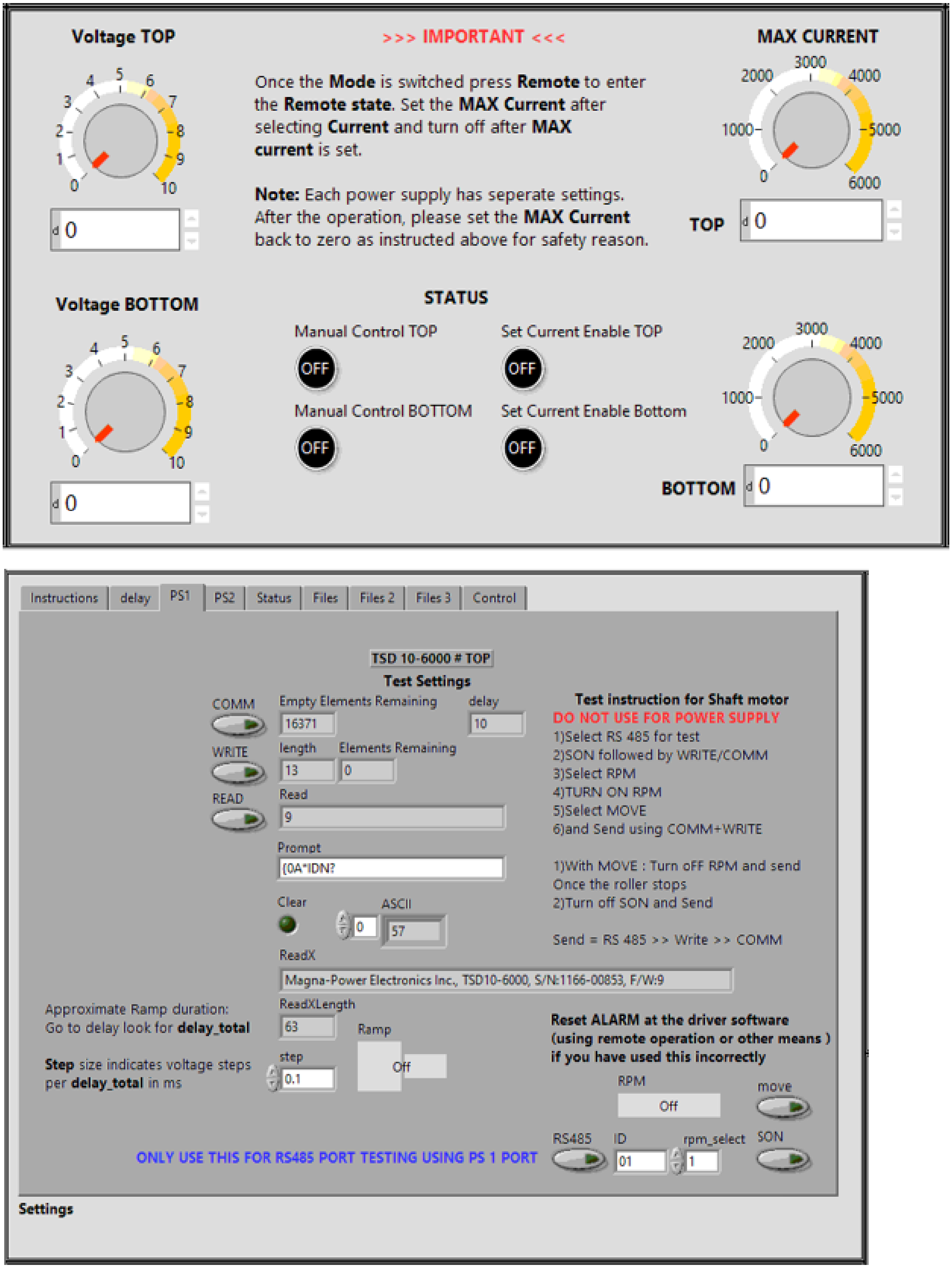


Figure 5.6: A section of the front panel.

6 Control System Implementation

This section details the experimental design and system modeling process for estimating the transfer function of a graphite roller system heated using electrical power. The system modeling is followed by controller design and simulation. Finally, the designed controller is implemented to test the performance. The objective of this study is to develop a single-input, single-output (SISO) transfer function that accurately represents the thermal dynamics of the roller. The experiment is designed to collect data under controlled conditions, and the data is subsequently used to identify the system's behavior.

6.1 Experimental Setup

The experiment involves a graphite roller that is rotated at a constant speed of 0.02 revolutions per minute (rpm). This controlled rotation ensures consistent conditions during data collection. Electrical power is applied to the roller through graphite brush contacts. The input to the system is the voltage applied, which ranges from 0 to 5 volts DC. Safety precautions limit the voltage range to ensure safe operation. The power supply used for this purpose is the Magna Power 10-6000 TSD, with a maximum current capacity of 6000 amps. However, during the experiment, the current is limited to a maximum of 1000 amps. The output of interest is the temperature of the roller at a specific point. Temperature is measured in real-time using a pyrometer, providing accurate and instantaneous temperature readings. Data is collected at a sampling rate of 1 second. Experiments are conducted with varying input voltages, and the corresponding temperature responses are recorded. The data is exported as CSV files and later formatted into the IDDATA format in MATLAB®for further analysis. Figure 6.1 shows experimental data plots for the system identification.

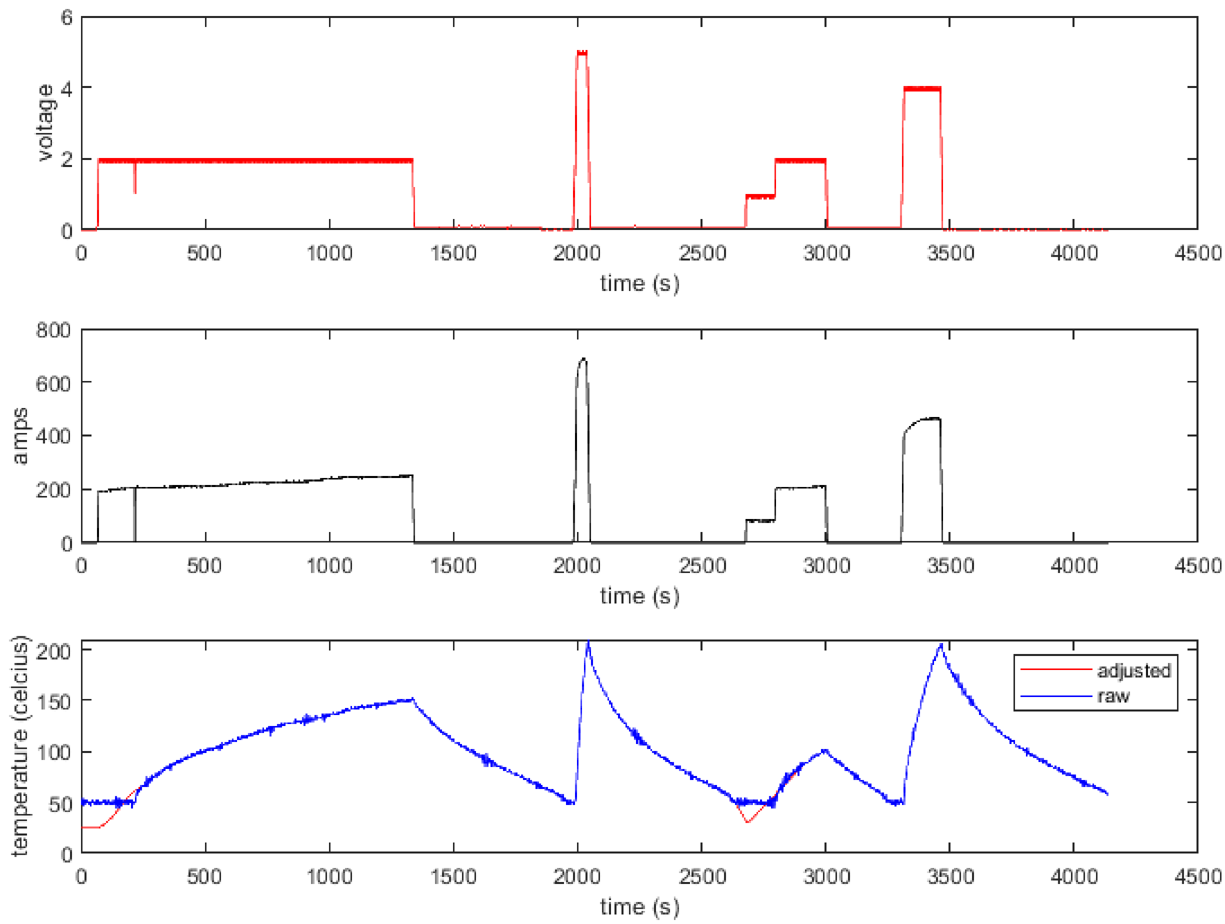


Figure 6.1: Experimental data plots

6.2 System Identification Process

The system identification process is a crucial step in developing a transfer function that accurately represents the thermal behavior of the roller system. MATLAB®'s System Identification Toolbox is employed for this purpose. Before system identification, the collected data is preprocessed. This includes interpolating and extrapolating the data for the range of temperature not captured by the pyrometer. In addition, cleaning and formatting the data is done to ensure its suitability for analysis. The data is loaded into MATLAB®and converted into the IDDATA format. Transfer function models of various orders are estimated using the System Identification Toolbox. The goal is to find the model that best describes the relationship between the input voltage and the roller's temperature response. To select the best model, various performance metrics and validation techniques are employed. The identified models are assessed for goodness of fit and predictive accuracy. Model validation ensures that the selected transfer function accurately captures the system's behavior. The best-fitted transfer function estimation is shown in Equation 6.1. Equation 6.2 shows the discrete-time transfer function estimate with a sampling time of 1 seconds. Figure 6.2 shows the model fitting with a validation accuracy of 56.37%. The estimated model's step response, bode plot, pole-zero map, and Nyquist plots are shown in Figures 6.3,6.4, 6.5, and 6.6 respectively.

$$tfest = \frac{0.32s^3 + 0.0005s^2 + 0.0002s + 3.94 \times 10^{-7}}{s^4 + 0.009s^3 + 0.0006s^2 + 5.461 \times 10^{-6}s + 3.33 \times 10^{-9}} \tag{6.1}$$

$$tfest_{\text{discrete}} = \frac{0.3876z^{-1} - 1.162z^{-2} + 1.161z^{-3} - 0.3868z^{-4}}{1 - 3.988z^{-1} + 5.965z^{-2} - 3.966z^{-3} + 0.9889z^{-4}} \tag{6.2}$$

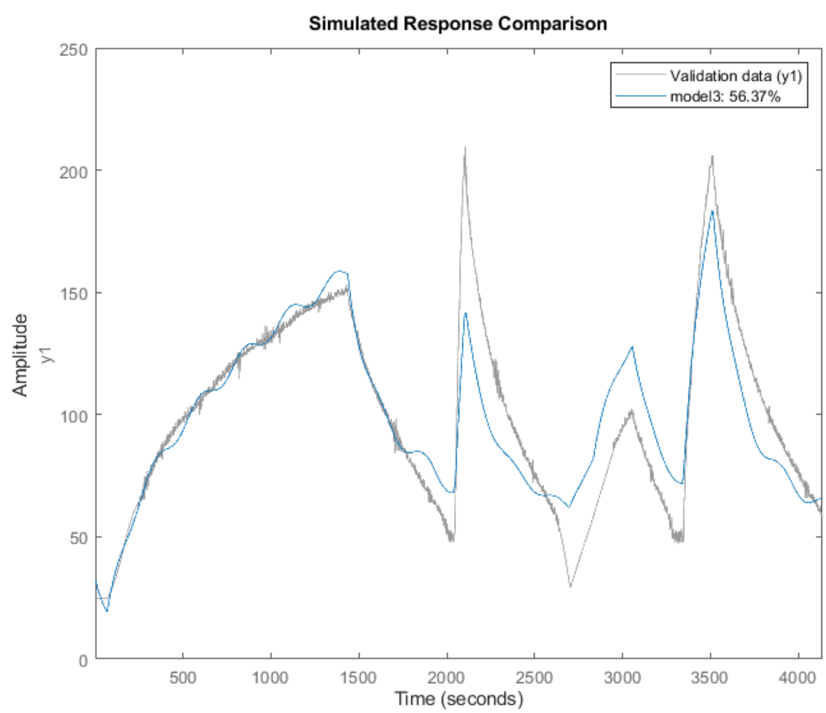


Figure 6.2: Model Fitting Accuracy

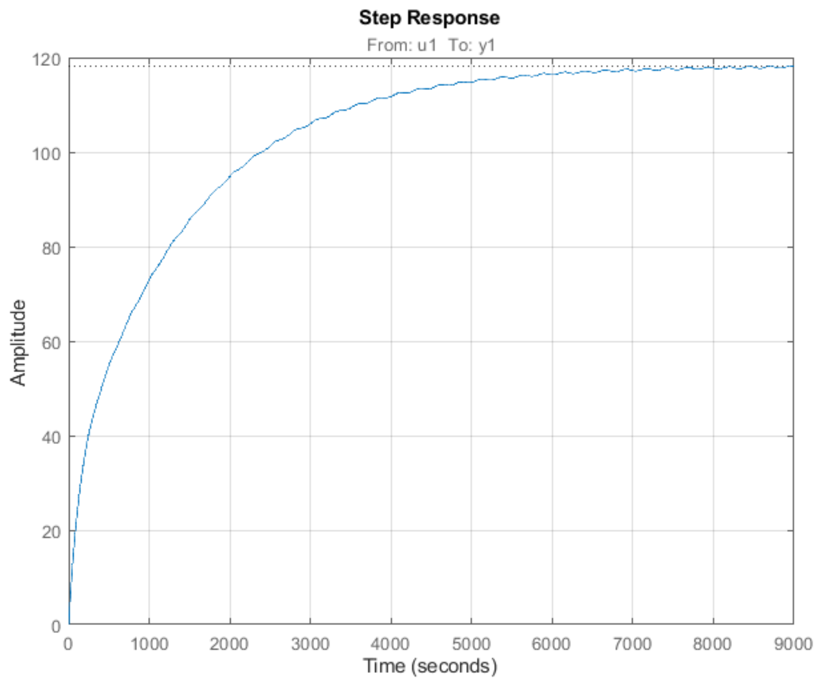


Figure 6.3: Step Response

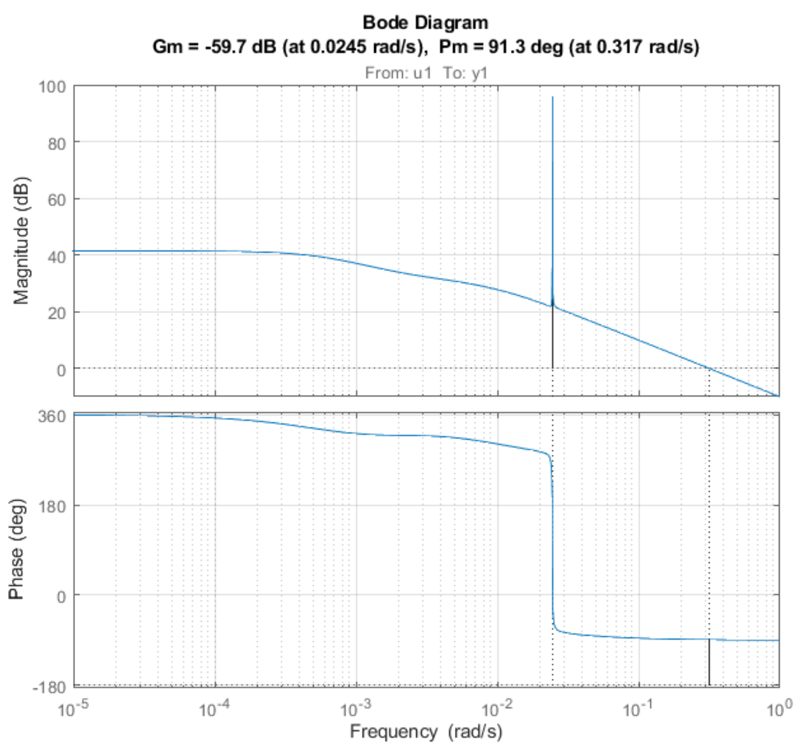


Figure 6.4: Bode Plot

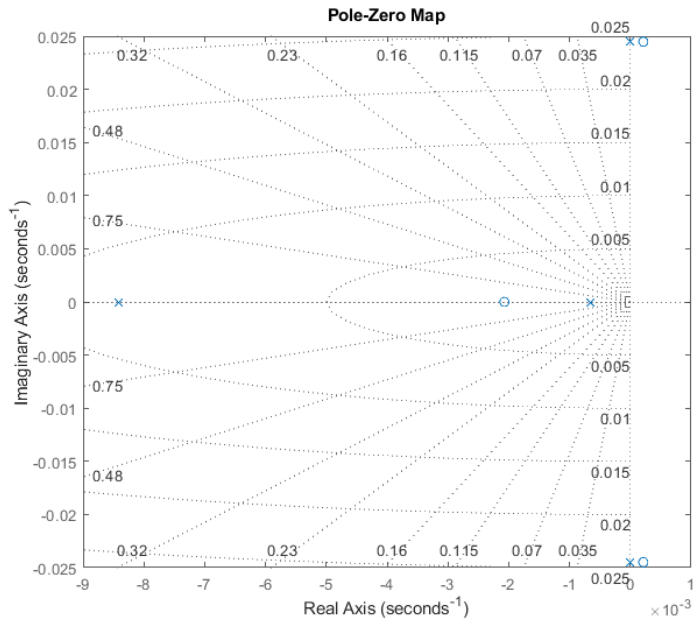


Figure 6.5: Pole-Zero Map

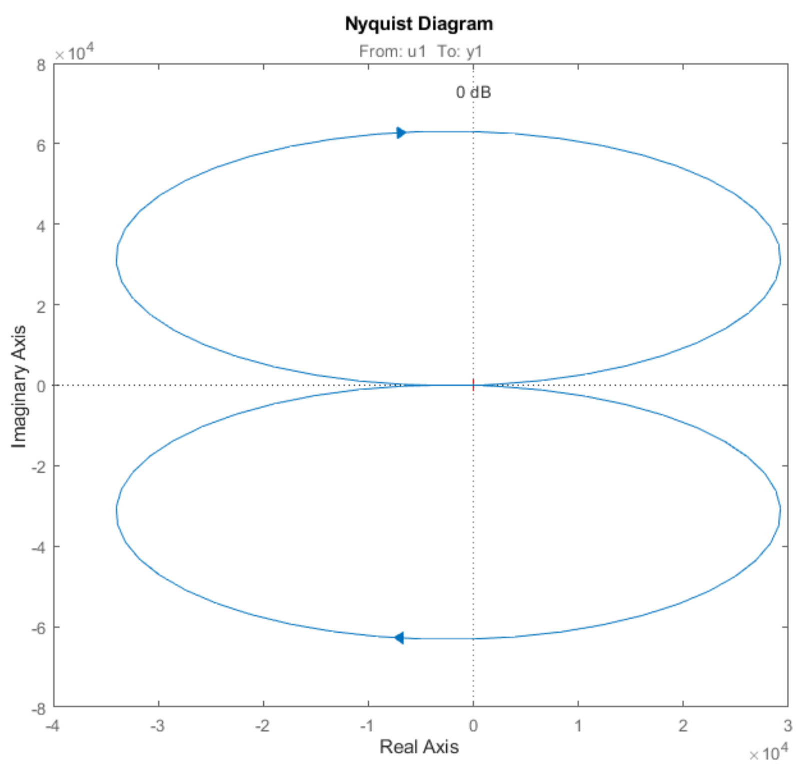


Figure 6.6: Nyquist Plot

6.3 Controller Design

6.3.1 PID Controller

The proportional–integral–derivative (PID) controller is one of the most common control loop strategies utilized in industrial process control and automation systems. The popularity of PID controllers results from their simple control structure, which enables effective closed-loop setpoint following across single-input, single-output systems in applications ranging from motor drives to temperature regulation.

A PID controller continuously calculates error values defined as the difference between a measured process variable and a desired setpoint. It attempts to minimize the error over time by adjusting the control input variable through three terms: proportional (P), integral (I), and derivative (D). The proportional term accounts for present error conditions by contributing a control signal proportionate to current deviations. The integral term integrates historical errors, steadily ramping up control input based on the accumulated offset to eliminate steady-state errors. Finally, the derivative term predicts future trajectory by differentiating error and improving settling time and stability.

Tuning PID gains appropriately helps optimize transient performance given rise times, overshoot, settling time, and other dynamic system requirements. Tuning techniques include manual trial-and-error adjustment guided by recording system response, as well as analytical methods like Ziegler-Nichols. Optimal PID tuning minimizes error and achieves steady-state setpoint tracking within tolerance constraints. In this work, MATLAB®'s PID tuning tool is used to find the initial controller parameters. Figure 6.7 shows the Simulink®schematic for the controller simulation.

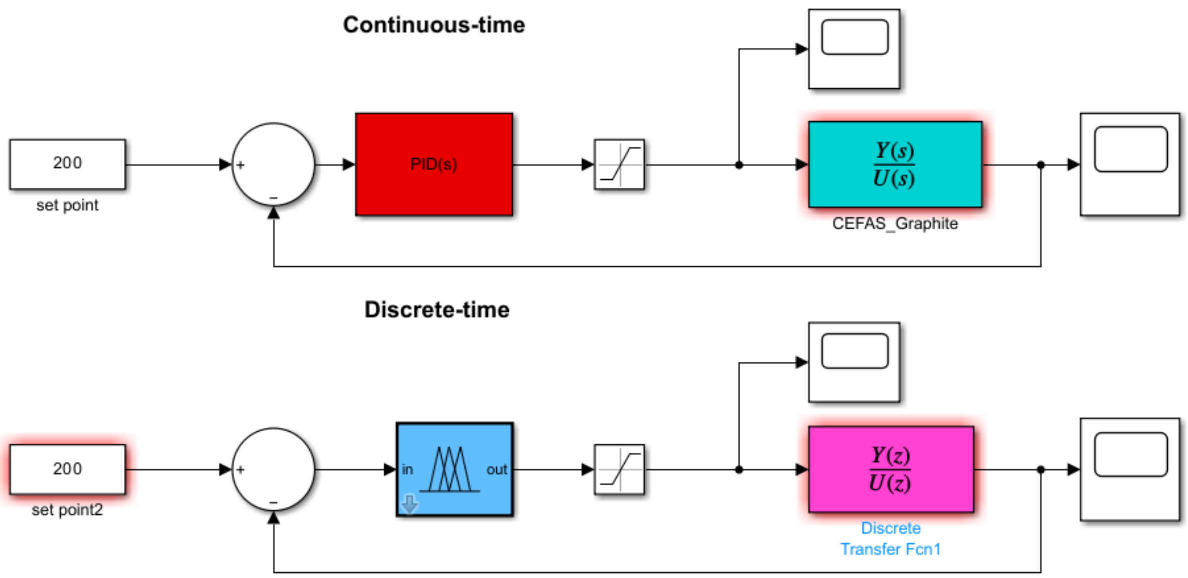


Figure 6.7: Simulink®schematic for the controller simulation.

6.3.2 Fuzzy Logic Controller

Fuzzy logic provides an alternative framework for capturing uncertain, non-linear real-world behaviors using linguistic variables and heuristic, qualitative rules. In contrast with classical logic that relies on binary true or false assignments, fuzzy logic enables partial truths between absolute affirmation and negation. This provides a smoothing construct similar to human cognition for decision-making under ambiguity.

A fuzzy logic system consists of fuzzifiers to map crisp measurements into fuzzy sets assigning membership grades, an inference engine housing IF-THEN rules that govern interactions, and defuzzifiers producing aggregated outputs. The Mamdani inference technique uses fuzzy inputs and outputs, synthesizing multiple rules via boolean operators into a single fuzzy output.

Fuzzy logic controllers designed using Mamdani inference play a key role in industrial process control applications requiring qualitative human insights. Membership functions translate sensor readings into degrees of truth for intuitive states like LOW, MEDIUM, or HIGH temperature. Expert rules subsequently qualitative goals to modify heating power based on current conditions. Defuzzification resolves aggregated fuzzy outputs from multiple rules into an applicable control signal.

This approach handles operator expertise and Provides customization reflecting the environment. Fuzzy logic delivering appropriate, stable control amid complex interactions or unmodeled effects. The fuzzy rulebase mediates measured states and desired criteria, approximating reasoning processes with linguistic semantics. Despite increased computational overhead, fuzzy logic controllers prove invaluable for dealing with imprecise, noisy data while enabling qualitative guidelines that enhance automation capability.

In summary, fuzzy logic controllers built on heuristic Mamdani inference provide an interpretable yet robust solution for industrial regulation problems where uncertainty limits traditional modeling approaches but human insights are available. The fuzzy framework linguistically processes data for increased automation flexibility.

In this study, the Mamdani Fuzzy Logic is developed to determine the control input to the target plant. The Fuzzy Logic controller is configured to take the error between the reference point and the current temperature state as input. Figure 6.7 shows the Simulink®schematic for the controller simulation with Fuzzy Logic-based control input. The fuzzy logic rules are shown in Figure 6.8 and the membership function utilized in the study is shown in Figure 6.9.

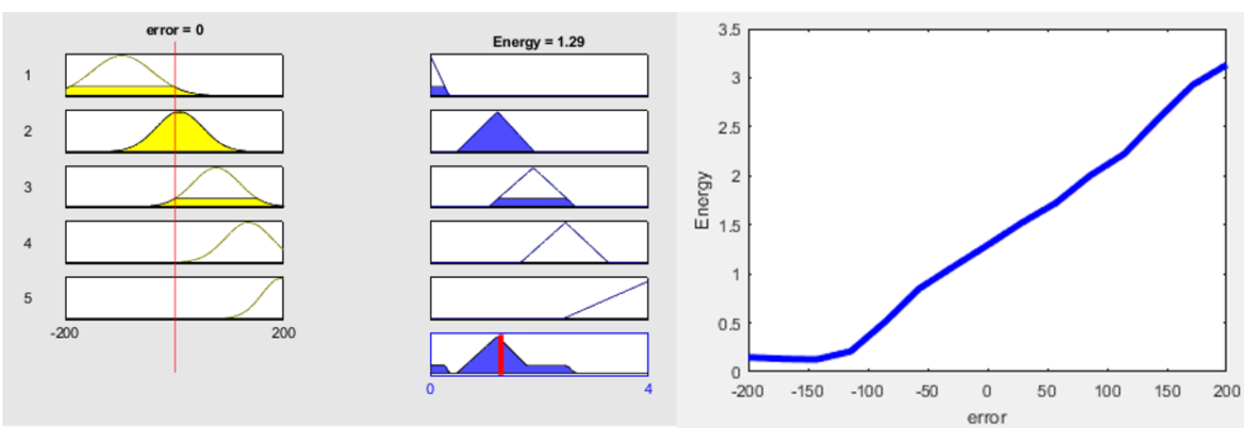


Figure 6.8: The Fuzzy Logic Rules

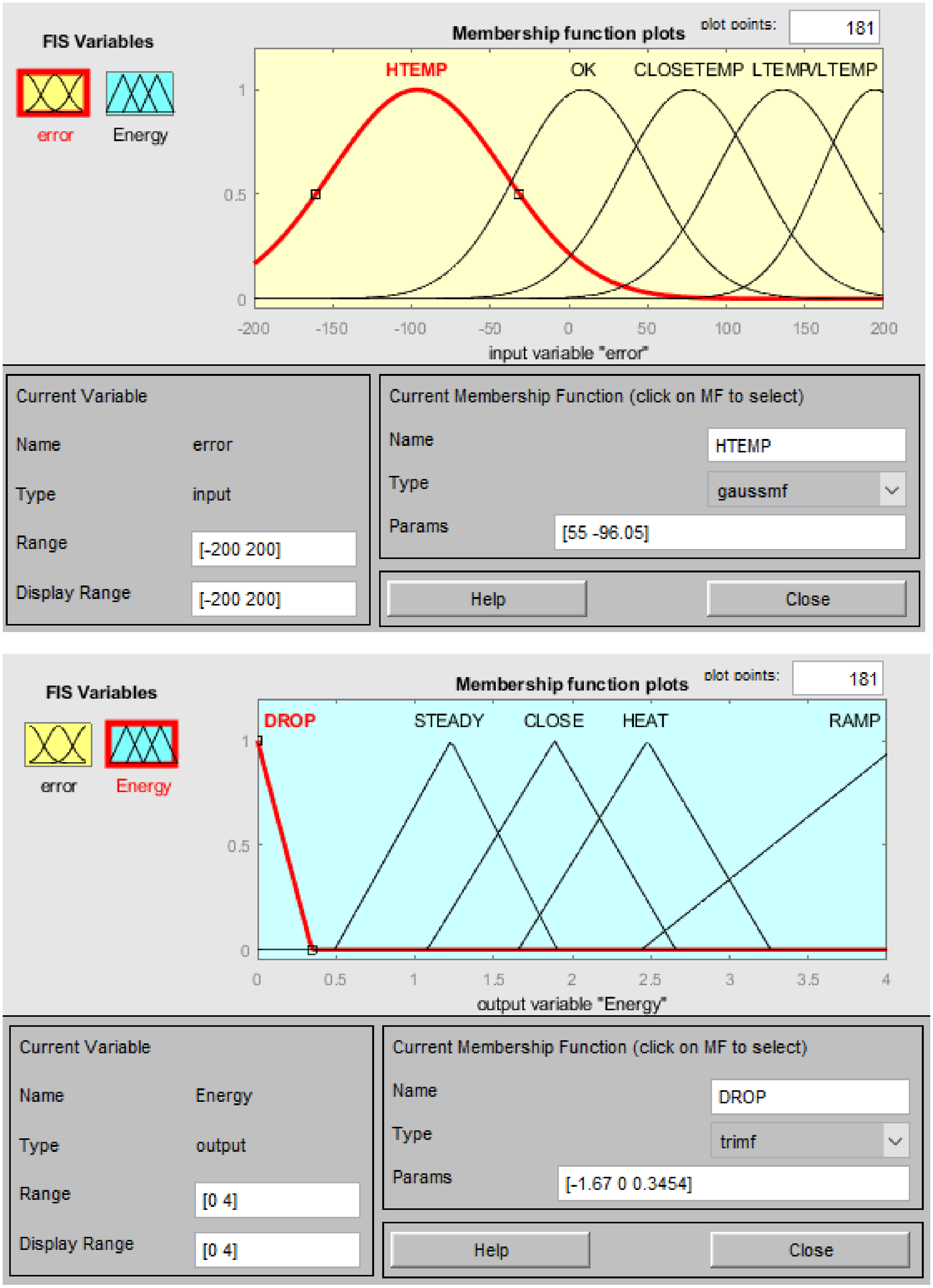


Figure 6.9: The fuzzy logic membership functions

6.4 Performance Analysis

The identified system model from the system identification process provides a characterization of process dynamics for controller tuning and evaluation. This mathematical approximation of the input-output relationship enables simulation to inform controller design prior to implementation.

Specifically, the model is leveraged to optimize proportional-integral-derivative (PID) controller parameters before deployment. A simulated 5000-second test applies a setpoint change from ambient to 200 degrees Celsius. The PID controller response demonstrates a settling time of approximately 2500 seconds with a 12.5 percent overshoot. This confirms acceptable but underdamped closed-loop behavior given the process requirements.

An alternative fuzzy logic controller is then simulated on the same system model, triggering an overdamped response without overshoot but with a longer 3000-second settling time. Additionally, while the PID controller showed no steady-state error, the fuzzy controller's response settled to a final value 7.5 percent below the setpoint. Comparatively, the PID controller demonstrates superior performance for the rapid heating application given its faster response and setpoint tracking.

Figure 6.10 overlays the simulation test results, highlighting the PID controller's faster rise time and lack of steady-state error relative to the fuzzy controller's slower and low but acceptable response. With increased confidence from the simulation results, the PID controller is coded into the LabVIEW host virtual instrument for autonomous temperature regulation.

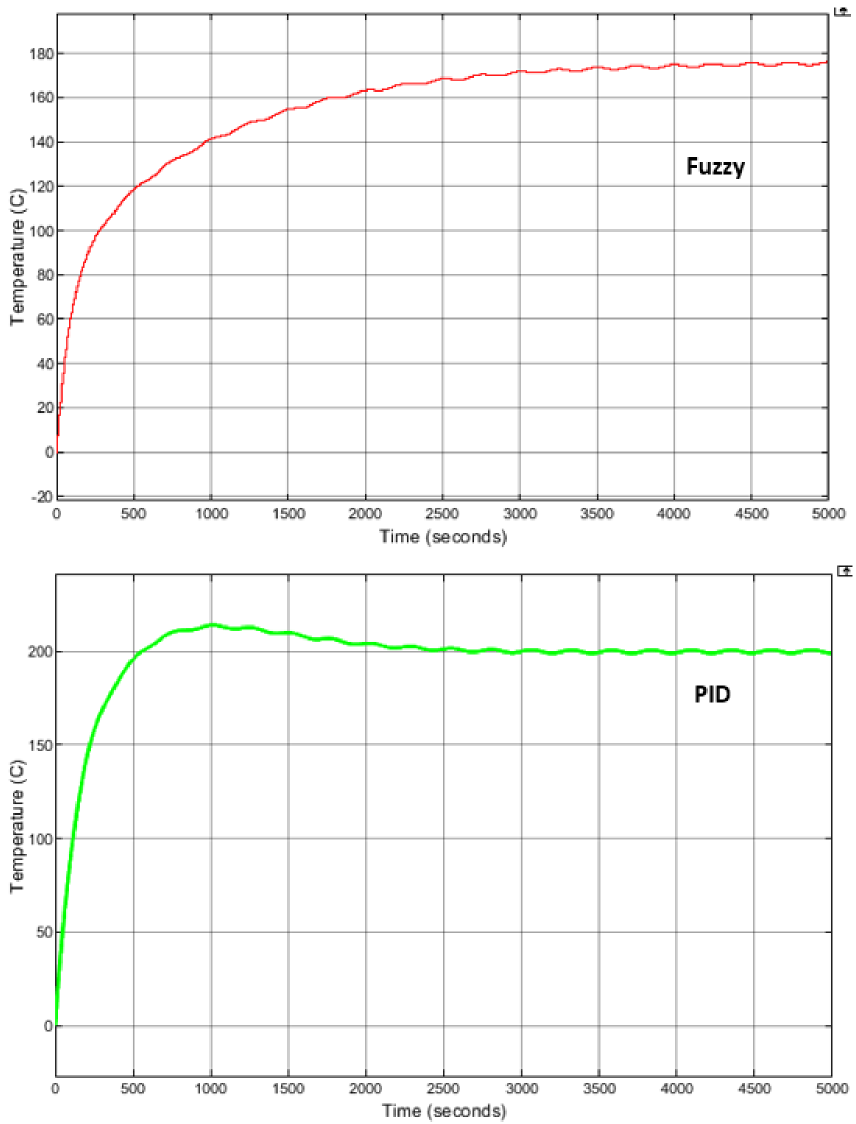


Figure 6.10: The simulation test results

The LabVIEW implementation seen in Figure 6.11 utilizes dedicated PID subvirtual instruments (subVIs) to carry out the control calculations and output updates at each time step. Experimental results indicate similar 2500-second settling behavior as estimated in the simulation. The corresponding underdamped response results in less than 5 percent overshoot and 5 percent steady-state error, aligning with the simulation. This confirms acceptable controller performance within the expected range given current system limitations. The Figure 6.12 shows the test results in the actual plant.

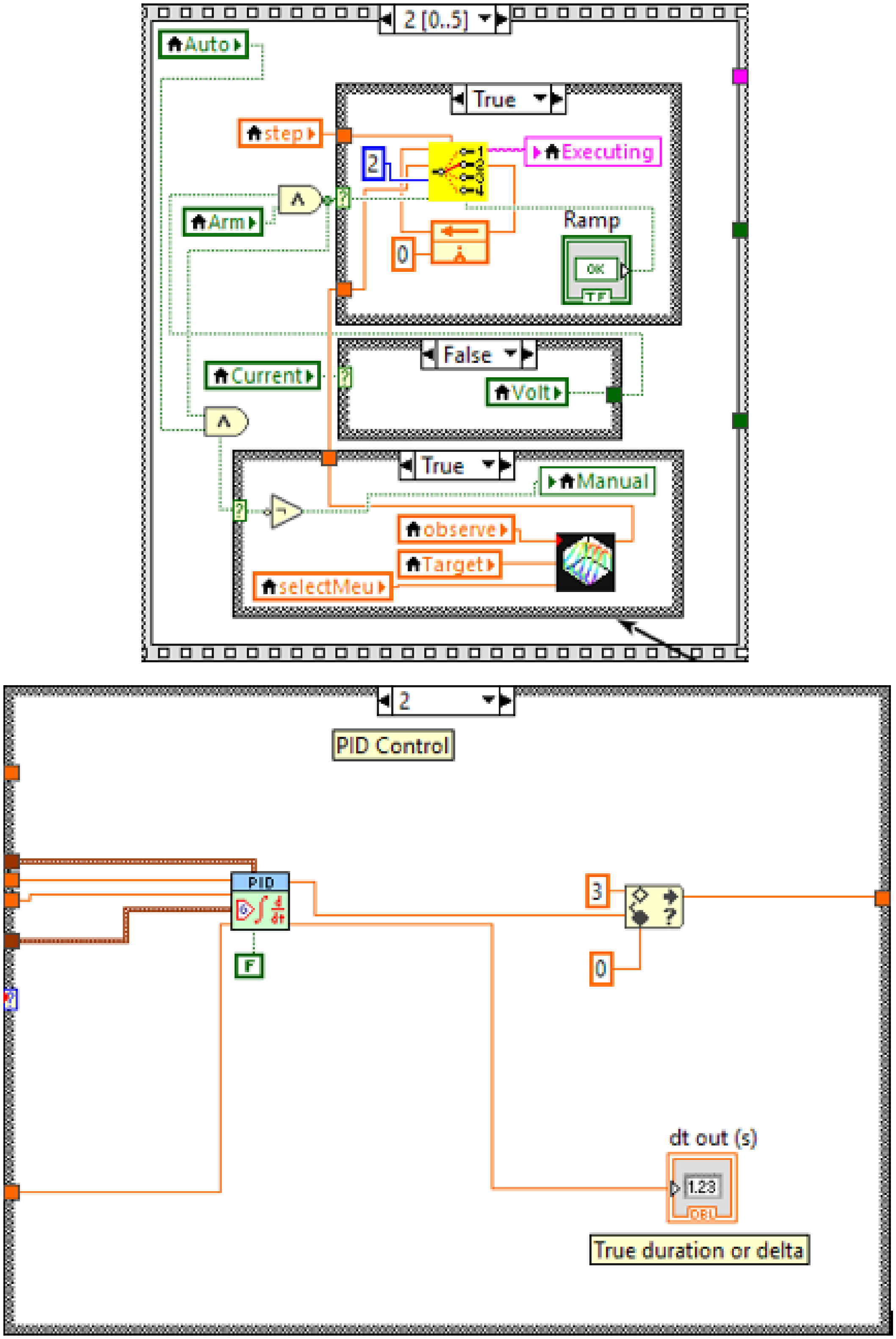


Figure 6.11: The LabVIEW controller implementation

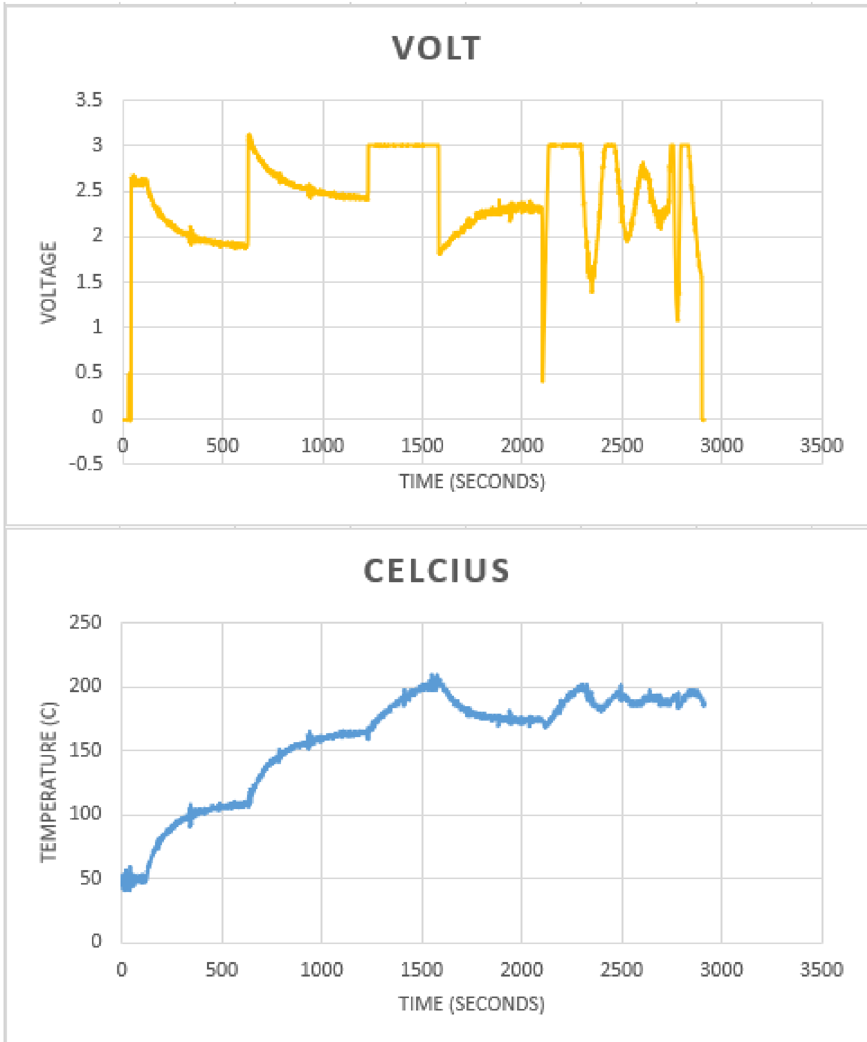


Figure 6.12: Control Implementation test result (actual)

7 Conclusion

This research makes valuable contributions towards advancing rapid joule heating processes for advanced manufacturing through the development of a continuous temperature control approach. The novelty lies in transitioning from conventional batch methods constrained by equipment geometries to leveraging a pair of rapidly joule heated graphite rollers. Effective thermal regulation within these rollers is integral to enabling continuous operation.

The project methodology facilitates this aim through systematic instrumentation of a prototype featuring integrated cooling mechanisms and extensive feedback. Rigorous experiments collect input/output data to model the system's thermal dynamics. The derived transfer function accurately captures transient behavior, later leveraging simulations to validate controller performance prior to deployment.

Notably, a PID control strategy demonstrates superior setpoint tracking and faster response compared to a fuzzy logic alternative when evaluated in simulation. The PID is subsequently implemented within a LabVIEW-based central command unit responsible for system orchestration. Experiments confirm close alignment with the simulated behavior, achieving steady-state regulation within a several hundred degrees Celsius span through closed-loop control.

In conclusion, this work puts forth a platform to make rapid joule heating processes continuous for advanced manufacturing, enhancing efficiency and flexibility. The integration of multifaceted system components enables precise thermal management. Developing first-principles models and tailored PID controllers facilitate the delicate temperature regulation needs. Future work should focus on incorporating in-line monitoring to adjust controller tuning during operation. Extending the instrumentation and control framework from the current benchtop setup to industrial-scale systems can help realize the tangible benefits of continuous operation.

References

[1] E. A. Olevsky, C. Garcia-Cardona, and W. L. Bradbury. “Electric Field Assisted Sintering: Theories and Applications”. In: *Advanced Engineering Materials* 22.1 (2020), p. 1901258. DOI: 10.1002/adem.201901258.

[2] M. Tokita. “Trends in advanced SPS spark plasma sintering systems and technology”. In: *Journal of the Society of Powder Technology, Japan* 30.11 (1993), pp. 790–804. DOI: 10.4164/sptj.30.790.

[3] D. Wang et al. “Electric Field-Assisted Sintering in Materials Processing: Principles, Methods, and Applications”. In: *Advanced Functional Materials* 26.33 (2016), pp. 6013–6024. DOI: 10.1002/adfm.201601292.

[4] Y. Zhang et al. “Progress in the Development and Application of Electric Field Assisted Sintering Technologies”. In: *Journal of Materials Engineering and Performance* 27.9 (2018), pp. 4638–4653. DOI: 10.1007/s11665-018-3582-2.

[5] Z. A. Munir, U. Anselmi-Tamburini, and M. Ohyanagi. “The effect of electric field and pressure on the synthesis and consolidation of materials: A review of the spark plasma sintering method”. In: *Journal of Materials Science* 41.3 (2006), pp. 763–777. DOI: 10.1007/s10853-005-5918-7.

[6] G. Liu and S. Grasso. “Advances in Spark Plasma Sintering: Control Systems and Applications”. In: *Journal of Materials Science & Technology* 35.2 (2019), pp. 242–256. DOI: 10.1016/j.jmst.2018.09.039.

[7] W. Chen, Y. Li, and C. Xu. “Application of Spark Plasma Sintering in Advanced Material Production: A Review”. In: *Materials Science and Engineering: A* 772 (2020), p. 138814. DOI: 10.1016/j.msea.2019.138814.

[8] L. Zhao, H. Zhang, and X. Zhou. “Integrating Spark Plasma Sintering with Additive Manufacturing: Opportunities and Challenges”. In: *Advanced Manufacturing Processes* 36.3 (2021), pp. 455–467. DOI: 10.1002/amp2.10055.

[9] E. A. Olevsky, A. L. Maximenko, and L. Froyen. “Future Directions in Spark Plasma Sintering Research”. In: *Advanced Engineering Materials* 25.1 (2023), p. 2200054. DOI: 10.1002/adem.202200054.

[10] J. Smith, A. Patel, and R. Kumar. “Challenges and Opportunities in Spark Plasma Sintering: A Review”. In: *Journal of Materials Processing Technology* 288 (2022), p. 116918. DOI: 10.1016/j.jmatprotec.2021.116918.

[11] K. M. Gupta and W. L. Wong. “Spark Plasma Sintering in the Era of Industry 4.0: A Manufacturing Perspective”. In: *Smart Manufacturing* 3 (2021), pp. 1–13. DOI: 10.1007/s41871-021-00091-6.

Appendix

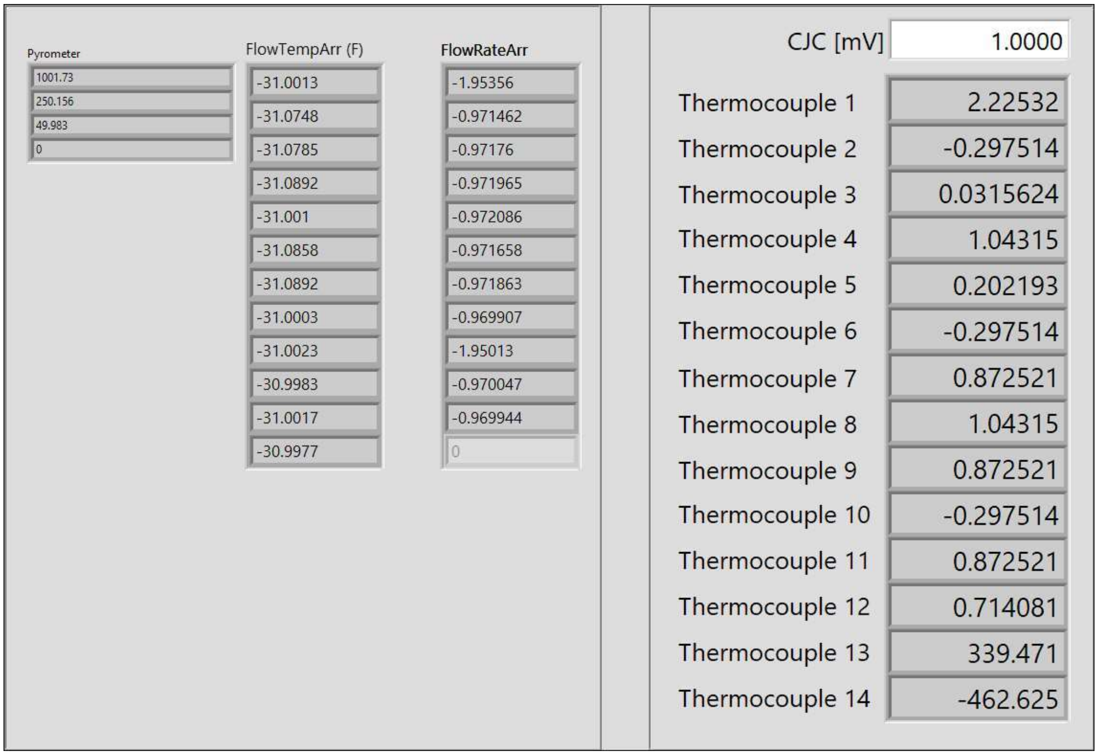
A Reproducible Resources

All the development work is available in the GitHub page link given below:
https://github.com/jamagola/EFAS

B Graphical Source Code and MATLAB®scripts

The Graphical Source Code and MATLAB®scripts used in the development are attached on the following pages.

20230621_ZMG_FPGAanalogReadIns.vi



CJC [mV]

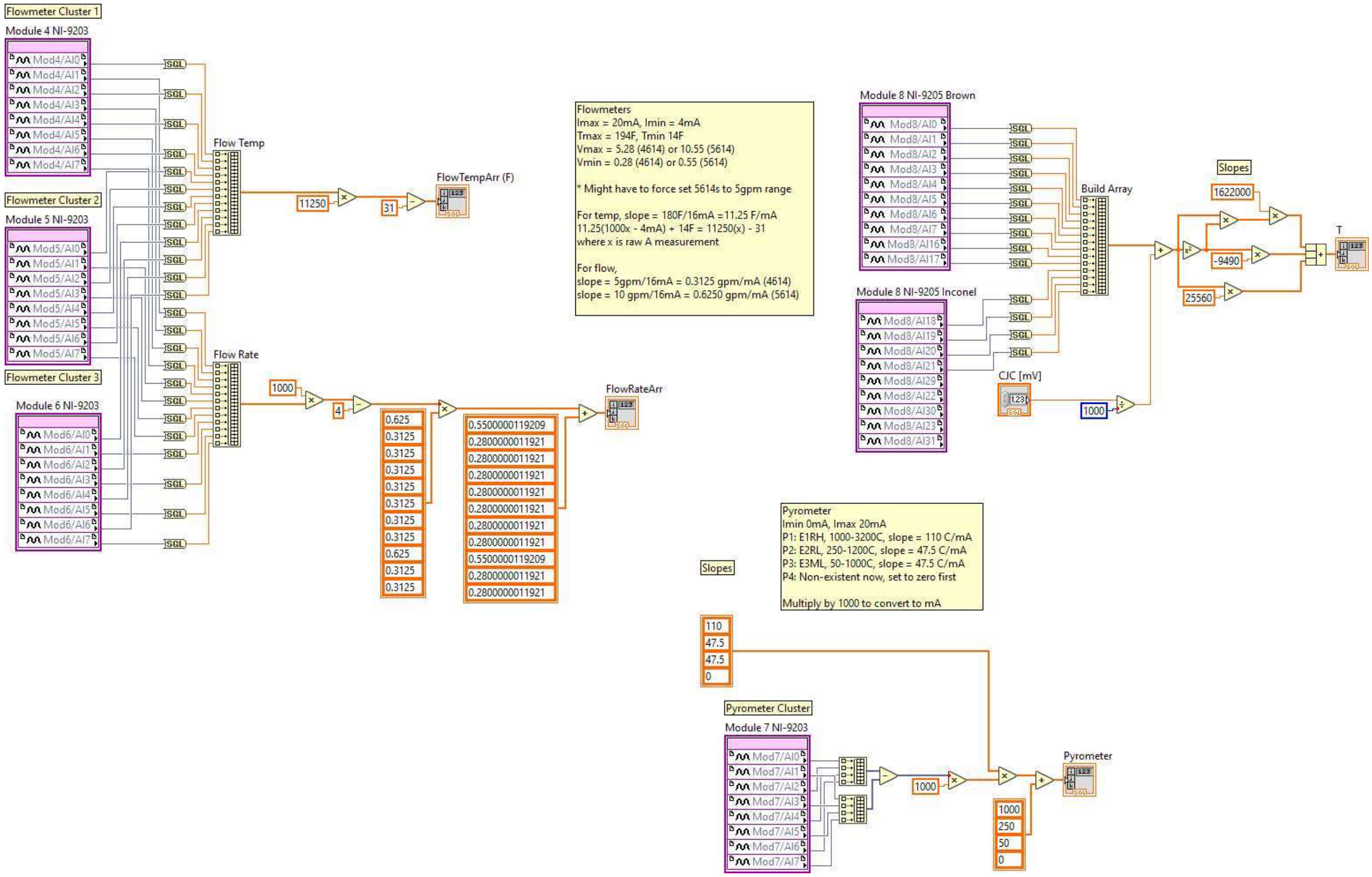
T

FlowRateArr[FlowRateArr]

FlowTempArr (F)

Pyrometer

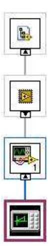
[]



"20230621_ZMG_FPGAanalogReadIns.vi History"

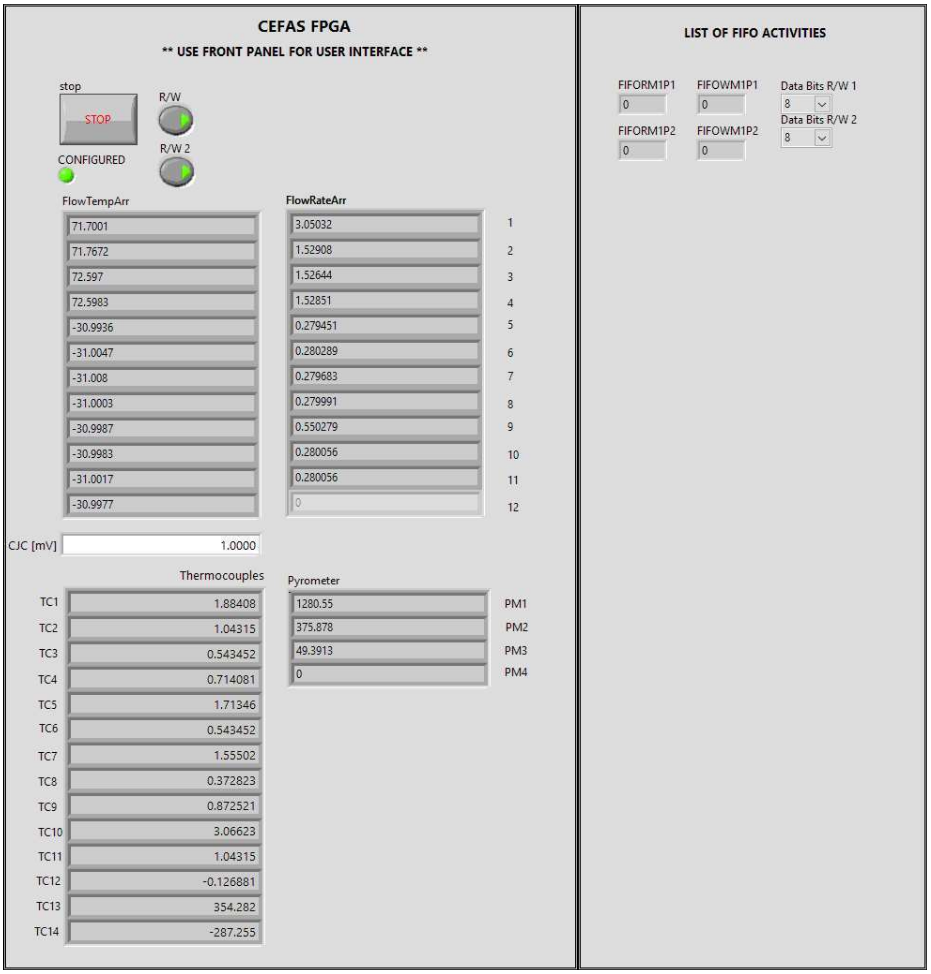
Current Revision: 152

Position in Hierarchy



Iconified Cluster Constants

CEFAS_FPGA.vi



stop
R/W
R/W 2
CJC [mV]
FIFORM1P1
CONFIGURED
FIFOWM1P1
Data Bits R/W 1
Data Bits R/W 2
FIFORM1P2
FIFOWM1P2
FlowTempArr

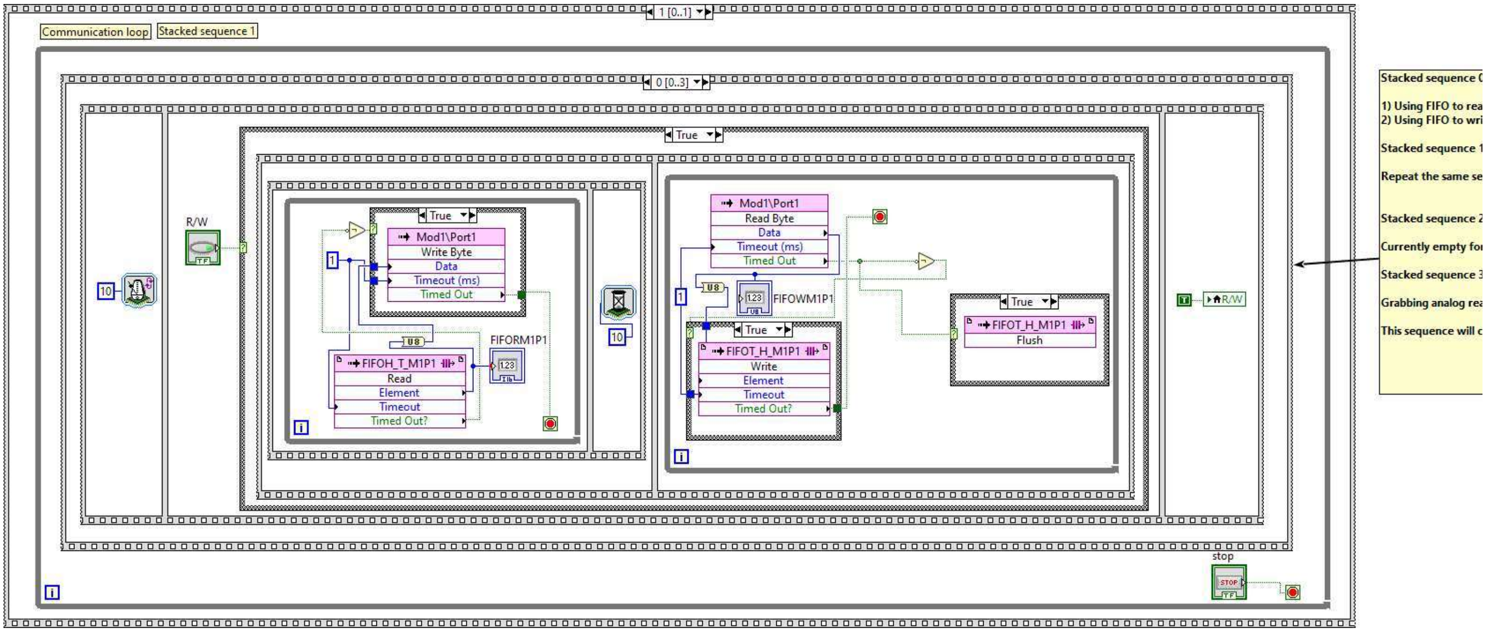
FlowRateArr[FlowRateArr]

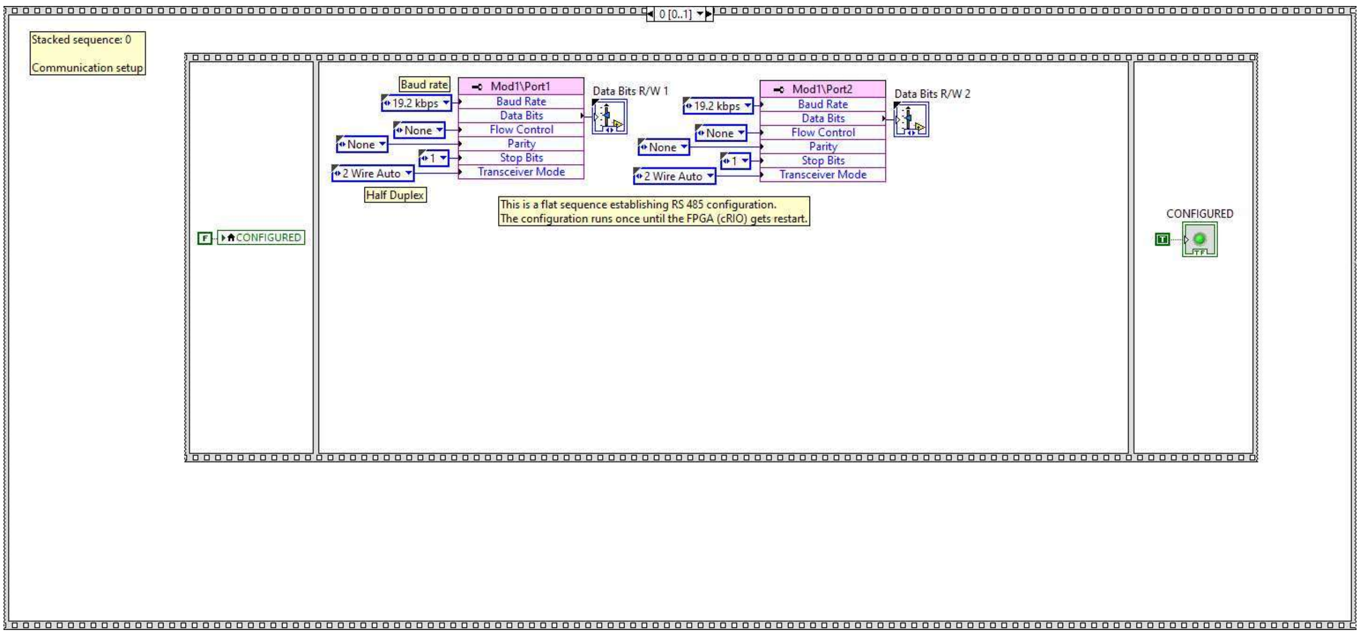
Thermocouples[T]

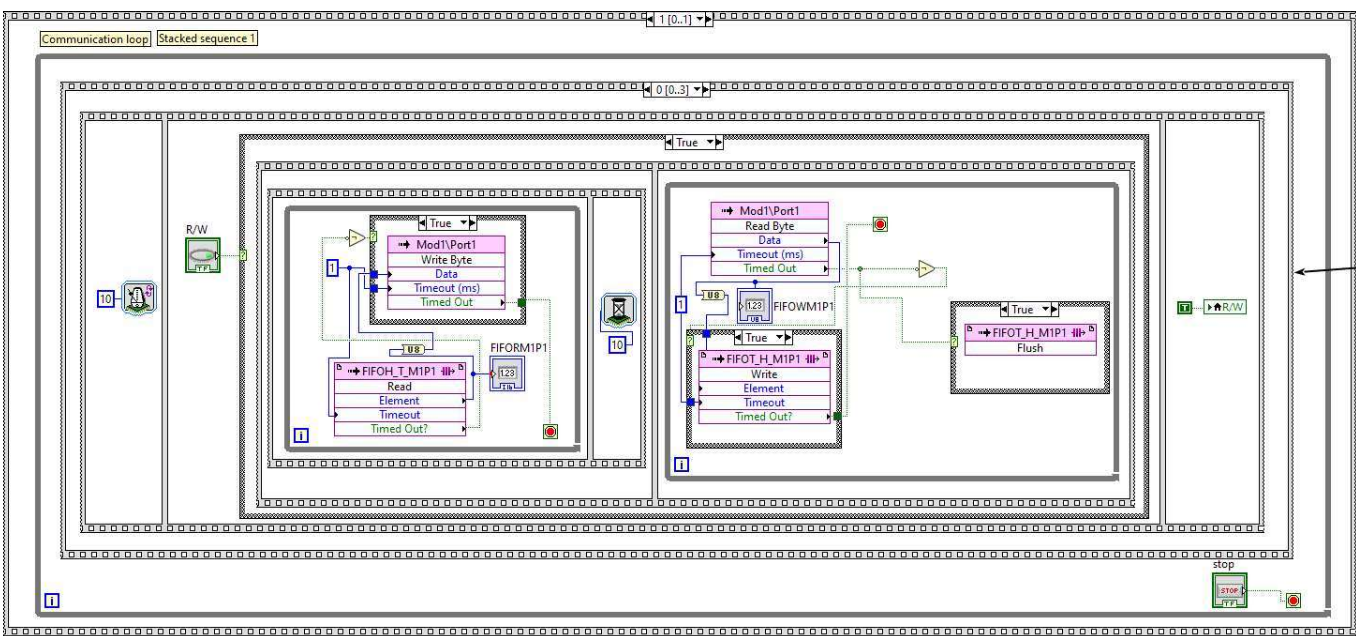
[]

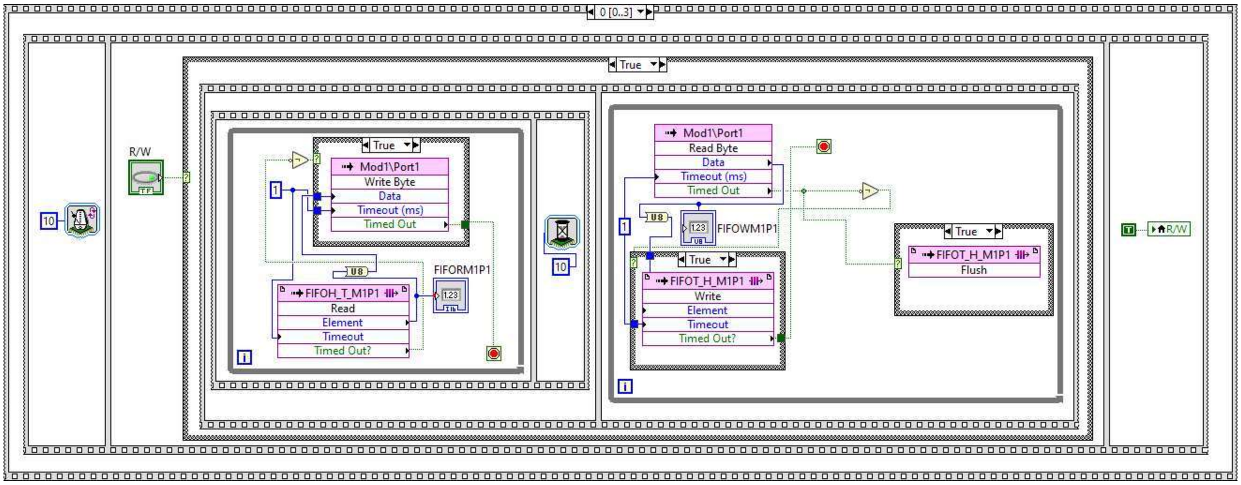
Pyrometer

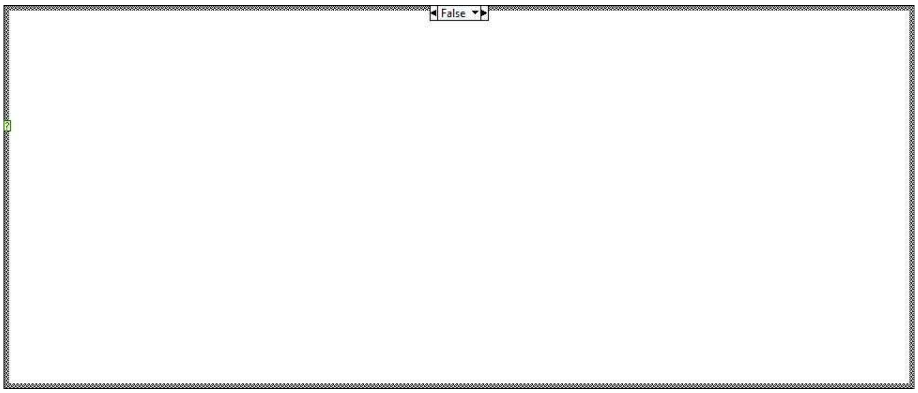
[]

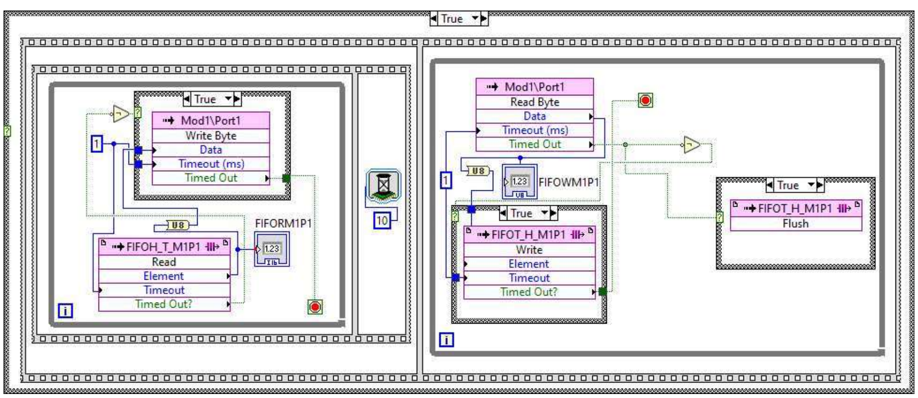

1 [0..1]
Communication loop
Stacked sequence 1
0 [0..3]
True
R/W
10
1
Mod1\Port1
Write Byte
Data
Timeout (ms)
Timed Out
U8
FIFOH_T_M1P1
Read
Element
Timeout
Timed Out?
FIFORM1P1
Mod1\Port1
Read Byte
Data
Timeout (ms)
Timed Out
FIFOWM1P1
FIFOT_H_M1P1
Write
Element
Timeout
Timed Out?
FIFOT_H_M1P1
Flush
R/W
stop
Stacked sequence 0
1) Using FIFO to rea
2) Using FIFO to wri
Stacked sequence 1
Repeat the same se
Stacked sequence 2
Currently empty fo
Stacked sequence 3
Grabbing analog rea
This sequence will c

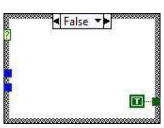

0 [0..1]
Stacked sequence: 0
Communication setup
CONFIGURED
Baud rate
19.2 kbps
None
None
1
2 Wire Auto
Half Duplex
Mod1\Port1
Baud Rate
Data Bits
Flow Control
Parity
Stop Bits
Transceiver Mode
Data Bits R/W 1
19.2 kbps
None
None
1
2 Wire Auto
Mod1\Port2
Baud Rate
Data Bits
Flow Control
Parity
Stop Bits
Transceiver Mode
Data Bits R/W 2
This is a flat sequence establishing RS 485 configuration.
The configuration runs once until the FPGA (cRIO) gets restart.
CONFIGURED

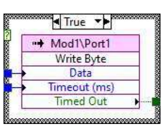

1 [0..1]
Communication loop
Stacked sequence 1
0 [0..3]
True
R/W
10
1
Mod1\Port1
Write Byte
Data
Timeout (ms)
Timed Out
U8
FIFOH_T_M1P1
Read
Element
Timeout
Timed Out?
FIFORM1P1
Mod1\Port1
Read Byte
Data
Timeout (ms)
Timed Out
FIFOWM1P1
FIFOT_H_M1P1
Write
Element
Timeout
Timed Out?
FIFOT_H_M1P1
Flush
R/W
stop

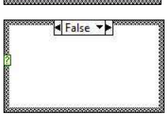

0 [0..3]
10
R/W
True
True
Mod1\Port1
Write Byte
Data
Timeout (ms)
Timed Out
1
U8
FIFOH_T_M1P1
Read
Element
Timeout
Timed Out?
FIFORM1P1
10
Mod1\Port1
Read Byte
Data
Timeout (ms)
Timed Out
1
U8
FIFOWM1P1
True
FIFOT_H_M1P1
Write
Element
Timeout
Timed Out?
True
FIFOT_H_M1P1
Flush
R/W

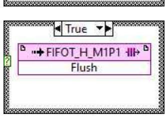

False

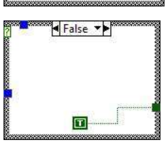

True
True
Mod1\Port1
Write Byte
Data
Timeout (ms)
Timed Out
1
U8
FIFOH_T_M1P1
Read
Element
Timeout
Timed Out?
FIFORM1P1
10
Mod1\Port1
Read Byte
Data
Timeout (ms)
Timed Out
1
U8
FIFOWM1P1
True
FIFOT_H_M1P1
Write
Element
Timeout
Timed Out?
True
FIFOT_H_M1P1
Flush

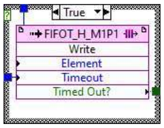

False

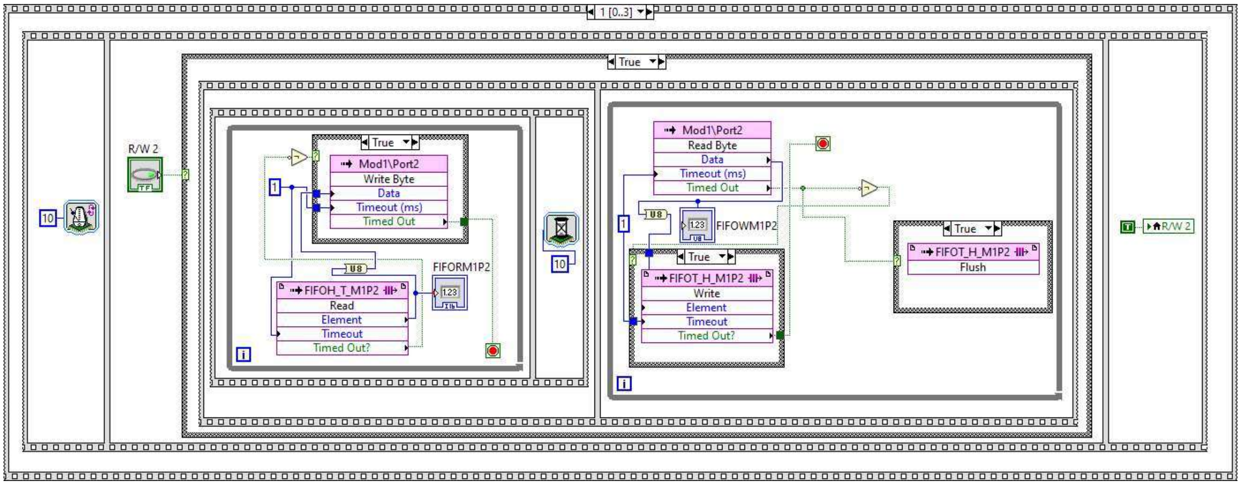

True
Mod1\Port1
Write Byte
Data
Timeout (ms)
Timed Out

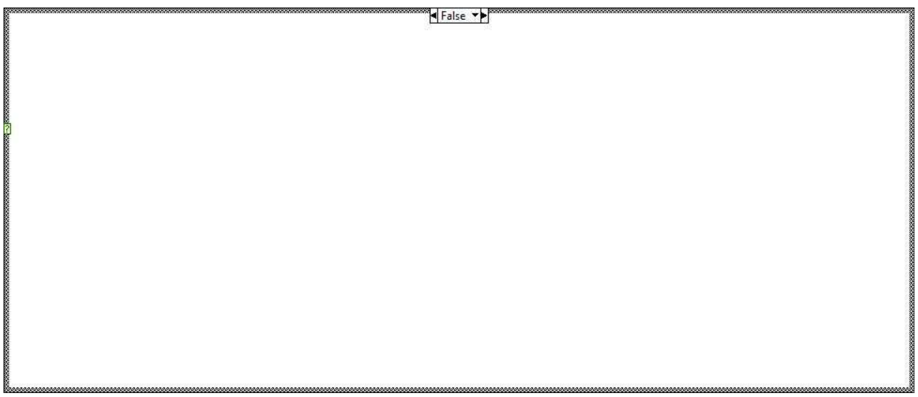

False

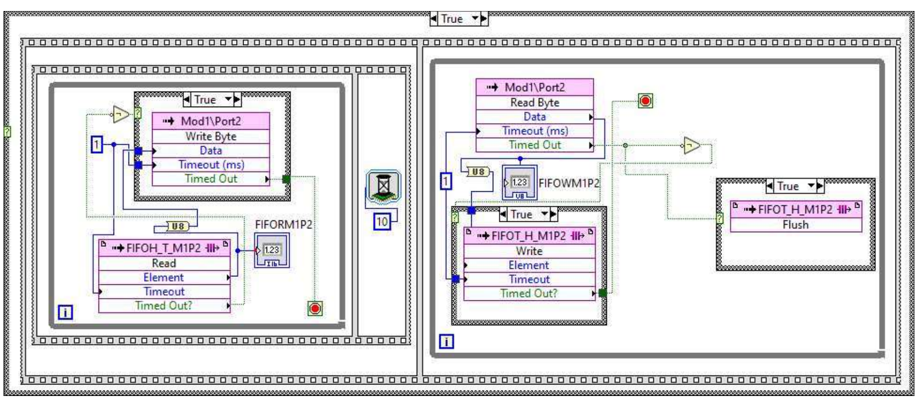

True
FIFOT_H_M1P1
Flush

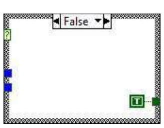

False

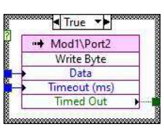

True
FIFOT_H_M1P1
Write
Element
Timeout
Timed Out?

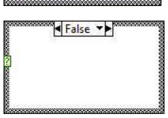

1 [0..3]
10
R/W 2
True
True
Mod1\Port2
Write Byte
Data
Timeout (ms)
Timed Out
1
U8
FIFORM1P2
FIFOH_T_M1P2
Read
Element
Timeout
Timed Out?
10
Mod1\Port2
Read Byte
Data
Timeout (ms)
Timed Out
1
U8
FIFOWM1P2
True
FIFOT_H_M1P2
Write
Element
Timeout
Timed Out?
True
FIFOT_H_M1P2
Flush
R/W 2

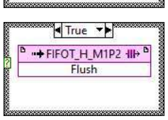

False

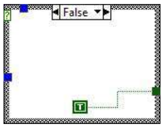

True
True
Mod1\Port2
Write Byte
Data
Timeout (ms)
Timed Out
1
U8
FIFORM1P2
FIFOH_T_M1P2
Read
Element
Timeout
Timed Out?
10
Mod1\Port2
Read Byte
Data
Timeout (ms)
Timed Out
1
U8
FIFOWM1P2
True
FIFOT_H_M1P2
Write
Element
Timeout
Timed Out?
True
FIFOT_H_M1P2
Flush

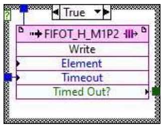

False

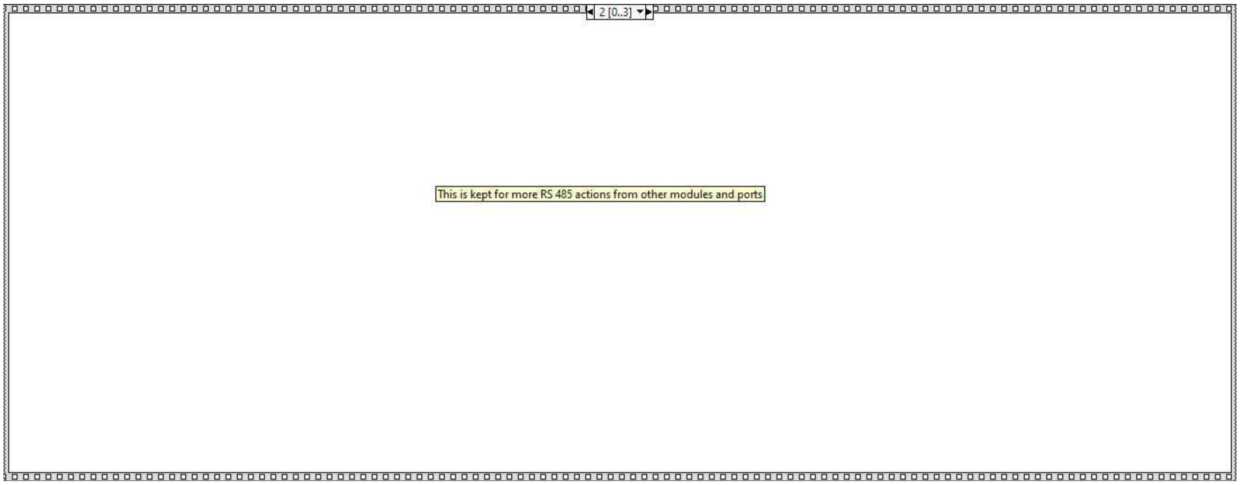

True
Mod1\Port2
Write Byte
Data
Timeout (ms)
Timed Out

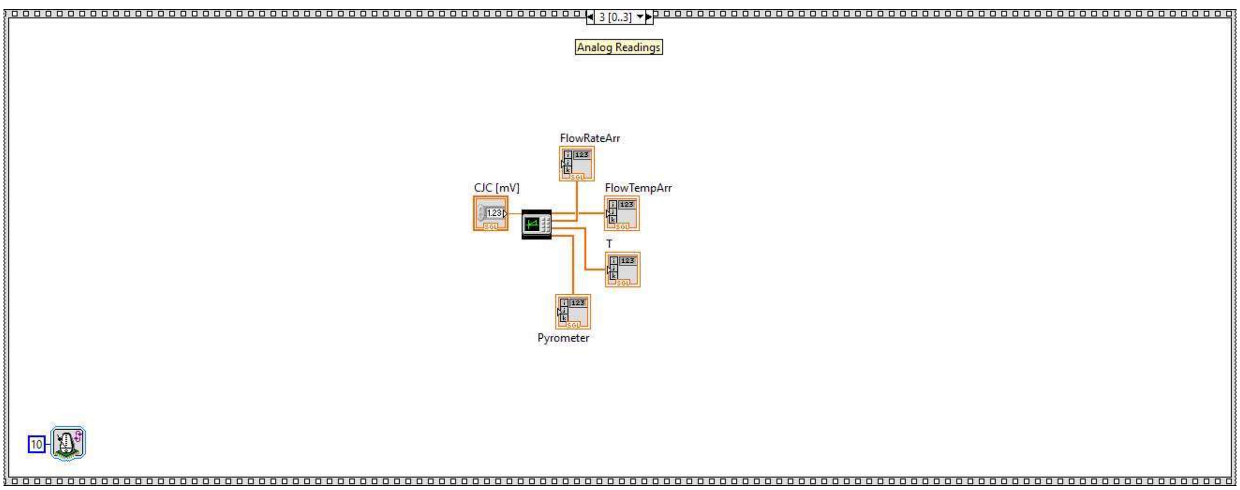

False


True
FIFOT_H_M1P2
Flush









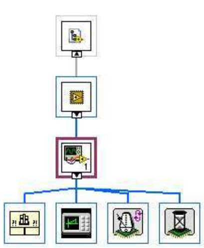
 **20230621_ZMG_FPGAanalogReadIns.vi**

C:\Users\inladmin\Desktop\cRIO_test_VI\EXPLORE\analog\20230621_ZMG_FPGAanalogReadIns.vi

"CEFAS_FPGA.vi History"

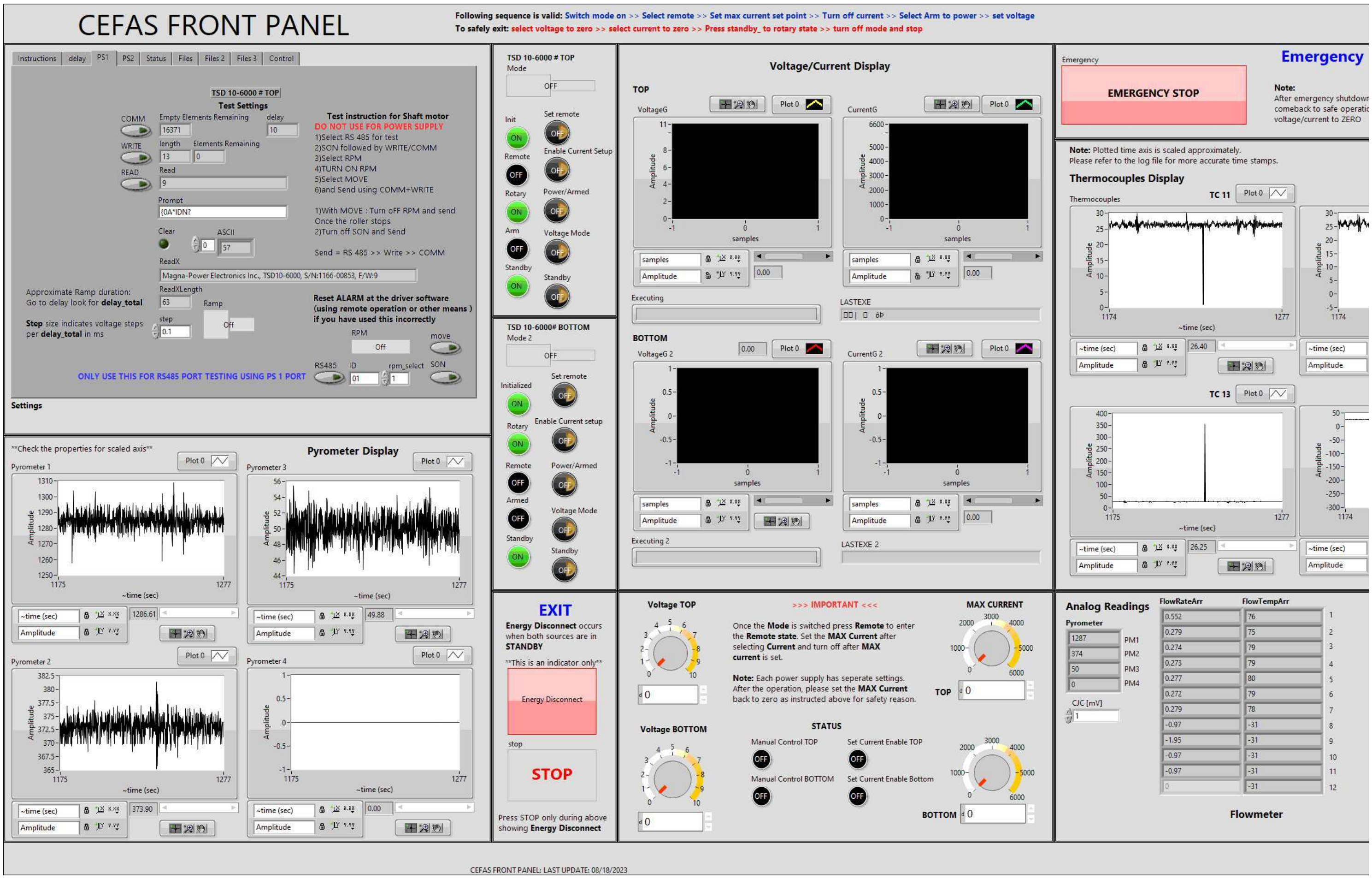
Current Revision: 42

Position in Hierarchy



Iconified Cluster Constants

CEFAS_HOST.vi



- COMM
- WRITE
- READ
- delay_total
- COMM 2
- READ 2
- WRITE 2
- Mode
- Prompt
- stop
- Set remote[Remote_]
- Power/Armed[Arm_]
- Voltage Mode[Volt]
- Standby[Standby_]
- step
- Voltage_[Voltage_]
- Mode 2
- Prompt 2
- Set remote[Remote_ 2]
- Power/Armed[Arm_ 2]
- Voltage Mode[Volt 2]
- Standby[Standby_ 2]
- step 2
- Voltage_[Voltage_ 2]
- CJC [mV]
- file path TC
- access TC
- operation TC
- Current_
- Enable Current Setup[Current]
- Enable Current setup [Current 2]
- Current_ 2
- file path Pyro
- access Pyro
- operation Pyro
- file path FlowMeter Temp
- access FlowMeter Temp
- operation FlowMeter Temp
- file path FlowMeter Rate
- access FlowMeter Rate
- operation FlowMeter Rate
- file path ms
- access ms
- operation ms
- file path volt
- access volt
- operation volt
- file path current
- access current
- operation current
- delay REC
- Controller[Auto]
- Controller 2[Auto 2]
- file path complete
- access complete
- operation complete
- selectMeu
- Target

selectMeu 2
Target 2
Settings
Ramp
Ramp 2
Emergency
RS485
ID
SON
rpm_select
RPM
move
Empty Elements Remaining
length
Elements Remaining
Read
Iteration
ReadX
ReadXLength
Clear
delay
ASCII

ASCII 2

ReadXLength 2
Read 2
Elements Remaining 2
Clear 2
ReadX 2
length 2
Empty Elements Remaining 2
delay 2
Rotary
Standby
Init
Remote
Executing
Arm
VoltageG
Rotary[Rotary 2]
Standby[Standby 2]
Remote[Remote 2]
Executing 2
Armed[Arm 2]
VoltageG 2
Initialized[Init 2]
Cutoff[Cutoff]
Pyrometer

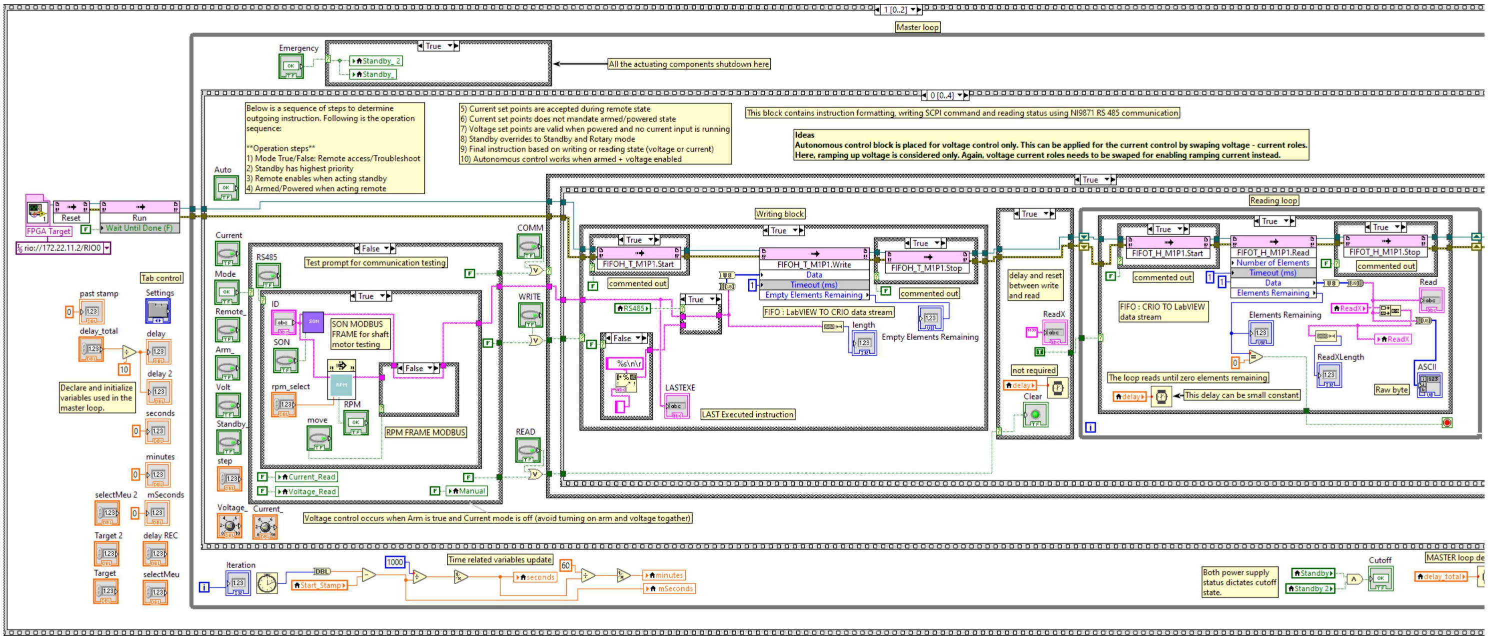
[]

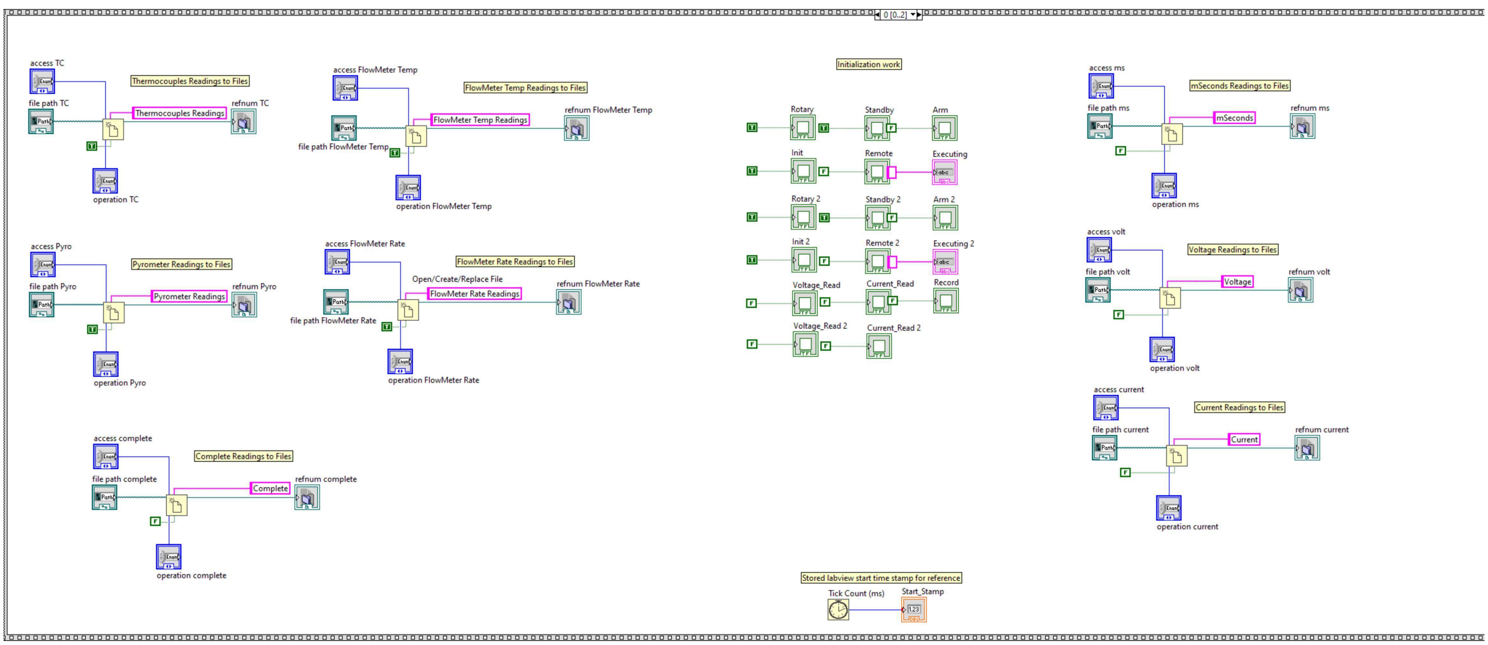
FlowTempArr

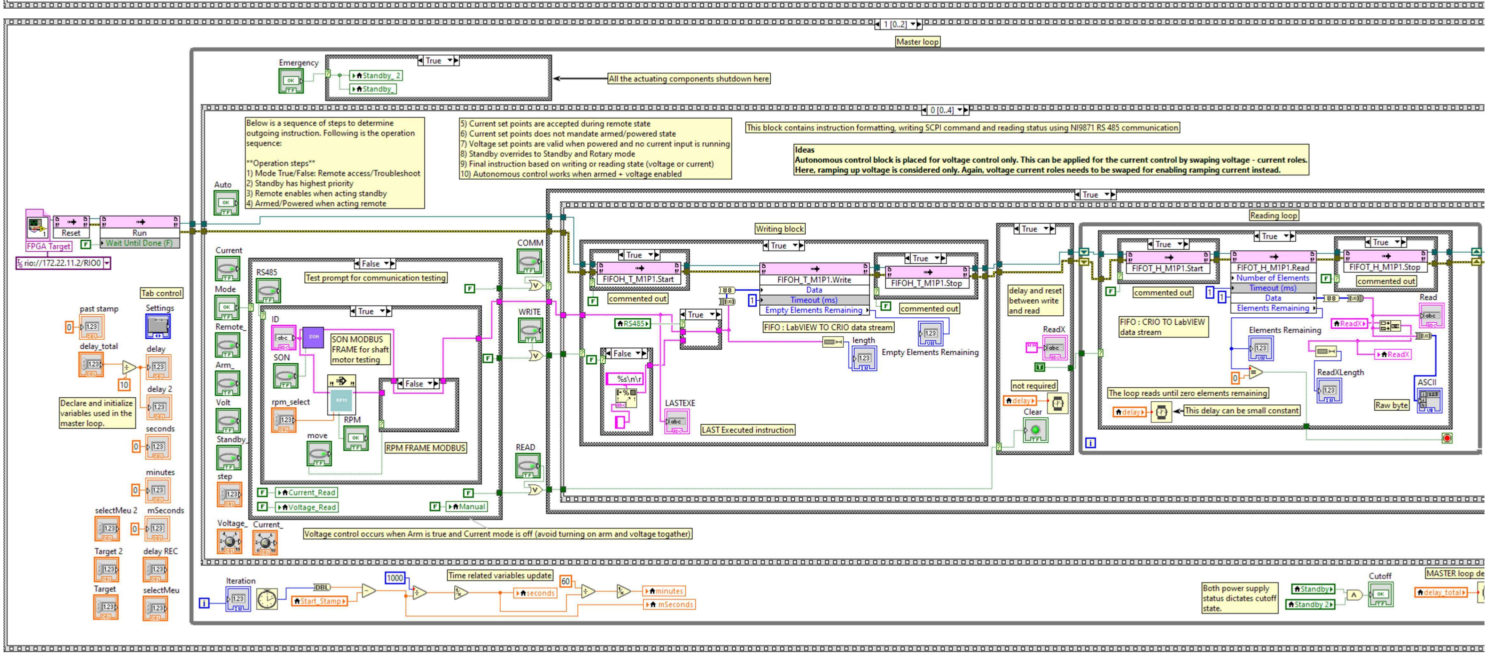
FlowRateArr

Thermocouples

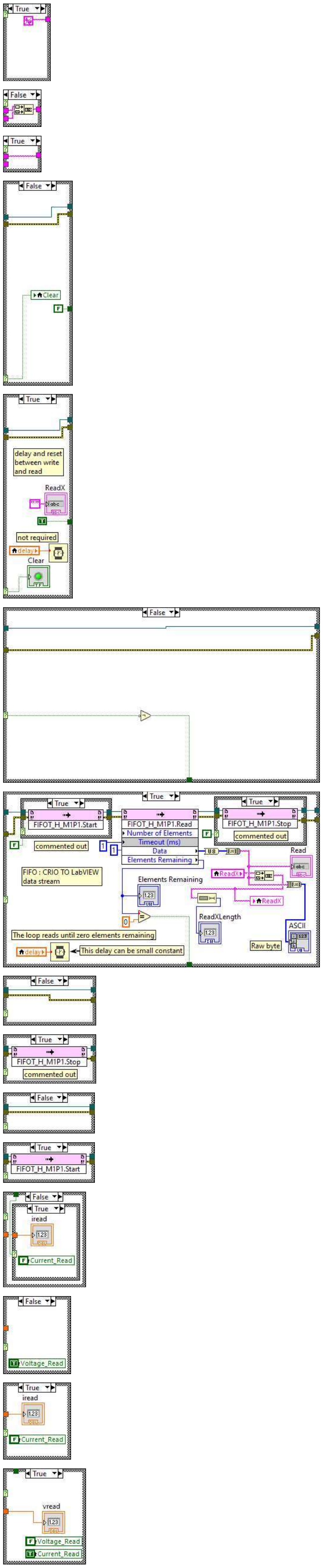
refnum TC
seconds
minutes
closing file TC
CurrentG
CurrentG 2
refnum Pyro
closing file Pyro
refnum FlowMeter Temp
closing file Flowmeter Temp
refnum FlowMeter Rate
closing file Flowmeter Rate
Voltage_Read
Current_Read
Voltage_Read 2
Current_Read 2
Record
vread
iread
iread 2
Voltage 2[vread 2]
mSeconds
refnum ms
refnum volt
refnum current
closing file ms
closing file volt
closing file current
past stamp
LASTEXE
LASTEXE 2
refnum complete
closing file complete
Start_Stamp
observe
Waveform Chart[Pyrometer 1]
observe 2
Waveform Chart[Pyrometer 2]
Waveform Chart[Pyrometer 3]
Waveform Chart[Pyrometer 4]
Waveform Chart[Thermocouples]
Manual Control TOP[Manual]
Manual Control BOTTOM[Manual 2]
Set Current Enable TOP[setCurr]
Set Current Enable Bottom[setCurr 2]
Waveform Chart[Thermocouples 2]
Waveform Chart[Thermocouples 3]
Waveform Chart[Thermocouples 4]

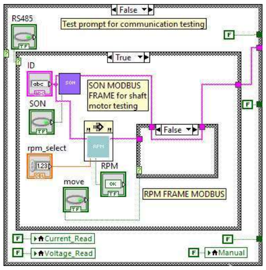

Master loop
All the actuating components shutdown here
Below is a sequence of steps to determine outgoing instruction. Following is the operation sequence:
Operation steps
1) Mode True/False: Remote access/Troubleshoot
2) Standby has highest priority
3) Remote enables when acting standby
4) Armed/Powered when acting remote
5) Current set points are accepted during remote state
6) Current set points does not mandate armed/powered state
7) Voltage set points are valid when powered and no current input is running
8) Standby overrides to Standby and Rotary mode
9) Final instruction based on writing or reading state (voltage or current)
10) Autonomous control works when armed + voltage enabled
This block contains instruction formatting, writing SCPI command and reading status using NI9871 RS 485 communication
Ideas
Autonomous control block is placed for voltage control only. This can be applied for the current control by swaping voltage - current roles.
Here, ramping up voltage is considered only. Again, voltage current roles needs to be swaped for enabling ramping current instead.
Tab control
Declare and initialize variables used in the master loop.
Test prompt for communication testing
SON MODBUS FRAME for shaft motor testing
RPM FRAME MODBUS
Writing block
FIFO : LabVIEW TO CRIO data stream
LAST Executed instruction
delay and reset between write and read
not required
Reading loop
FIFO : CRIO TO LabVIEW data stream
The loop reads until zero elements remaining
This delay can be small constant
Voltage control occurs when Arm is true and Current mode is off (avoid turning on arm and voltage togather)
Time related variables update
Both power supply status dictates cutoff state.
MASTER loop de

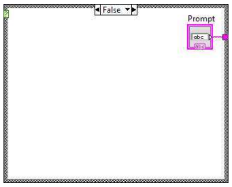

Thermocouples Readings to Files
Thermocouples Readings
FlowMeter Temp Readings to Files
FlowMeter Temp Readings
Initialization work
mSeconds Readings to Files
mSeconds
Pyrometer Readings to Files
Pyrometer Readings
FlowMeter Rate Readings to Files
FlowMeter Rate Readings
Open/Create/Replace File
Voltage Readings to Files
Voltage
Current Readings to Files
Current
Complete Readings to Files
Complete
Stored labview start time stamp for reference
Tick Count (ms)
Start_Stamp

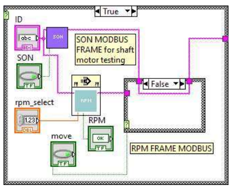

Master loop
All the actuating components shutdown here
This block contains instruction formatting, writing SCPI command and reading status using NI9871 RS 485 communication
Writing block
Reading loop
Time related variables update
Both power supply status dictates cutoff state.

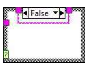
Below is a sequence of steps to determine outgoing instruction. Following is the operation sequence:
Operation steps
1) Mode True/False: Remote access/Troubleshoot
2) Standby has highest priority
3) Remote enables when acting standby
4) Armed/Powered when acting remote
5) Current set points are accepted during remote state
6) Current set points does not mandate armed/powered state
7) Voltage set points are valid when powered and no current input is running
8) Standby overrides to Standby and Rotary mode
9) Final instruction based on writing or reading state (voltage or current)
10) Autonomous control works when armed + voltage enabled
This block contains instruction formatting, writing SCPI command and reading status using NI9871 RS 485 communication
Ideas
Autonomous control block is placed for voltage control only. This can be applied for the current control by swaping voltage - current roles.
Here, ramping up voltage is considered only. Again, voltage current roles needs to be swaped for enabling ramping current instead.
Test prompt for communication testing
SON MODBUS FRAME for shaft motor testing
RPM FRAME MODBUS
Writing block
FIFOH_T_M1P1.Start
FIFOH_T_M1P1.Write
FIFOH_T_M1P1.Stop
commented out
FIFO : LabVIEW TO CRIO data stream
LAST Executed instruction
delay and reset between write and read
not required
Reading loop
FIFOT_H_M1P1.Start
FIFOT_H_M1P1.Read
FIFOT_H_M1P1.Stop
FIFO : CRIO TO LabVIEW data stream
The loop reads until zero elements remaining
This delay can be small constant
Test logic OFF
update reading status
Numerical read conversion
Voltage control occurs when Arm is true and Current mode is off (avoid turning on arm and voltage togather)

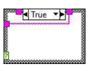

True
False
True
False
Clear
True
delay and reset between write and read
ReadX
not required
delay
Clear
False
True
True
FIFOT_H_M1P1.Start
commented out
FIFOT_H_M1P1.Read
Number of Elements
Timeout (ms)
Data
Elements Remaining
True
FIFOT_H_M1P1.Stop
commented out
Read
ReadX
FIFO : CRIO TO LabVIEW data stream
Elements Remaining
ReadX
ReadXLength
ASCII
The loop reads until zero elements remaining
delay
This delay can be small constant
Raw byte
False
True
FIFOT_H_M1P1.Stop
commented out
False
True
FIFOT_H_M1P1.Start
False
True
iread
Current_Read
False
Voltage_Read
True
iread
Current_Read
True
vread
Voltage_Read
Current_Read

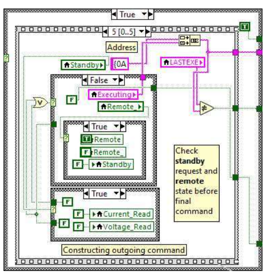

False
RS485
Test prompt for communication testing
True
ID
SON
SON MODBUS FRAME for shaft motor testing
rpm_select
RPM
move
False
RPM FRAME MODBUS
Current_Read
Voltage_Read
Manual

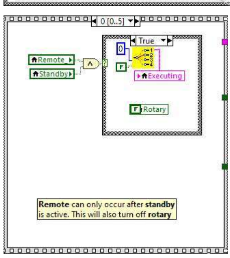

False
Prompt

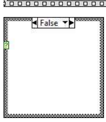

True
ID
SON
SON MODBUS FRAME for shaft motor testing
rpm_select
RPM
move
False
RPM FRAME MODBUS

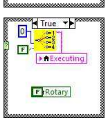

False

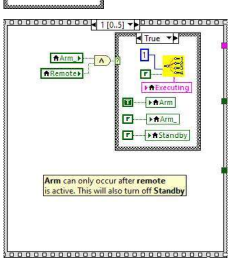

True

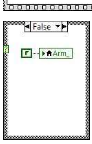

True
5 [0..5]
Address
Standby
0A
LASTEXE
False
Executing
Remote_
True
Remote
Remote_
Standby
Check standby request and remote state before final command
True
Current_Read
Voltage_Read
Constructing outgoing command

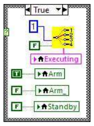

0 [0..5]
Remote_
Standby
True
0
Executing
Rotary
Remote can only occur after standby is active. This will also turn off rotary

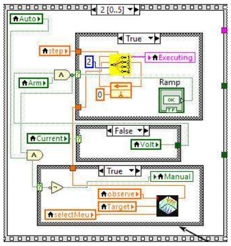

False

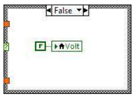

True
0
Executing
Rotary

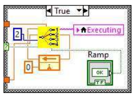

1 [0..5]
Arm_
Remote
True
1
Executing
Arm
Arm_
Standby
Arm can only occur after remote is active. This will also turn off Standby

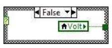

False
Arm_

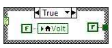


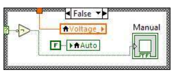


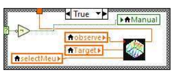


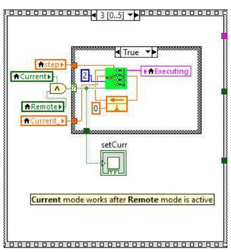


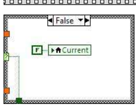


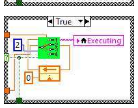


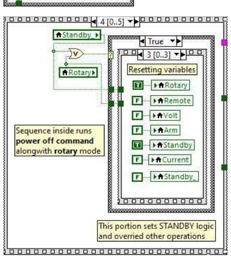


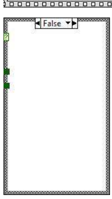


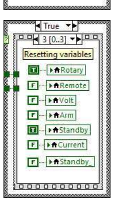

Current mode works after Remote mode is active

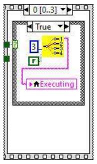


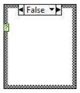


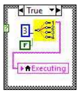

Resetting variables
Sequence inside runs power off command alongwith rotary mode
This portion sets STANDBY logic and overried other operations

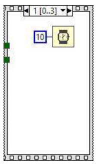


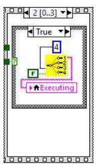

Resetting variables

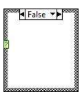

0 [0..3]
True
3
F
Executing

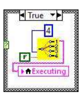

False

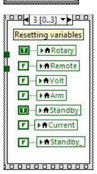

True
3
F
Executing

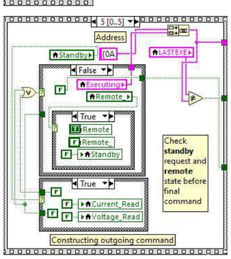

1 [0..3]
10

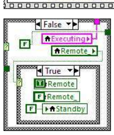

2 [0..3]
True
4
F
Executing

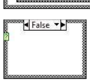

False

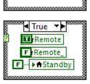

True
4
F
Executing

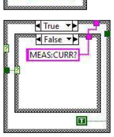

3 [0..3]
Resetting variables
Rotary
F
Remote
F
Volt
F
Arm
Standby
F
Current
F
Standby_

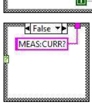

5 [0..5]
Address
Standby
0A
LASTEXE
False
F
Executing
Remote_
True
Remote
F
Remote_
F
Standby
Check
standby
request and
remote
state before
final
command
True
F
F
Current_Read
F
Voltage_Read
Constructing outgoing command

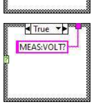

False
F
Executing
Remote_
True
Remote
F
Remote_
F
Standby

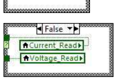

False

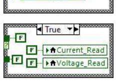

True
Remote
F
Remote_
F
Standby

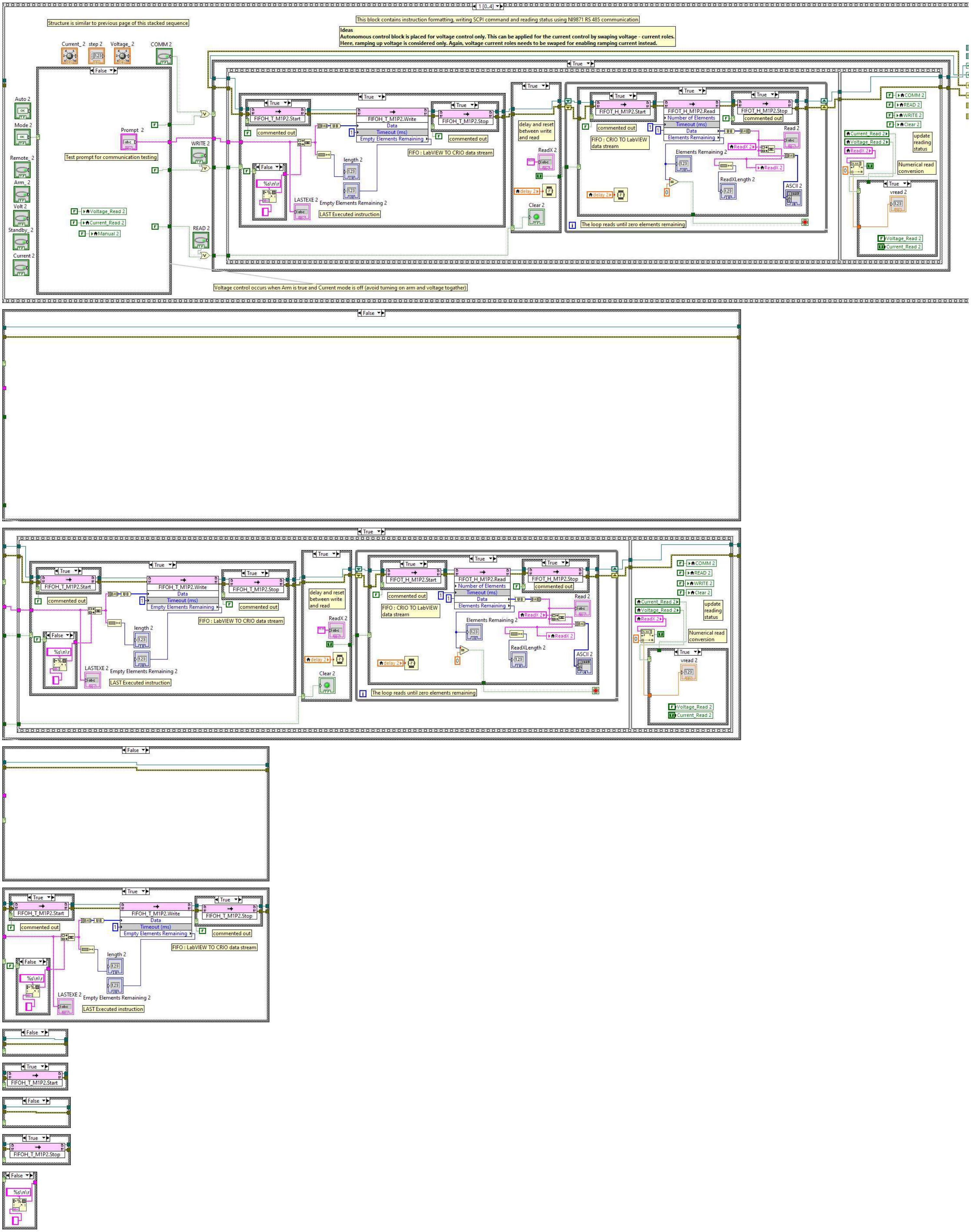

True
False
MEAS:CURR?

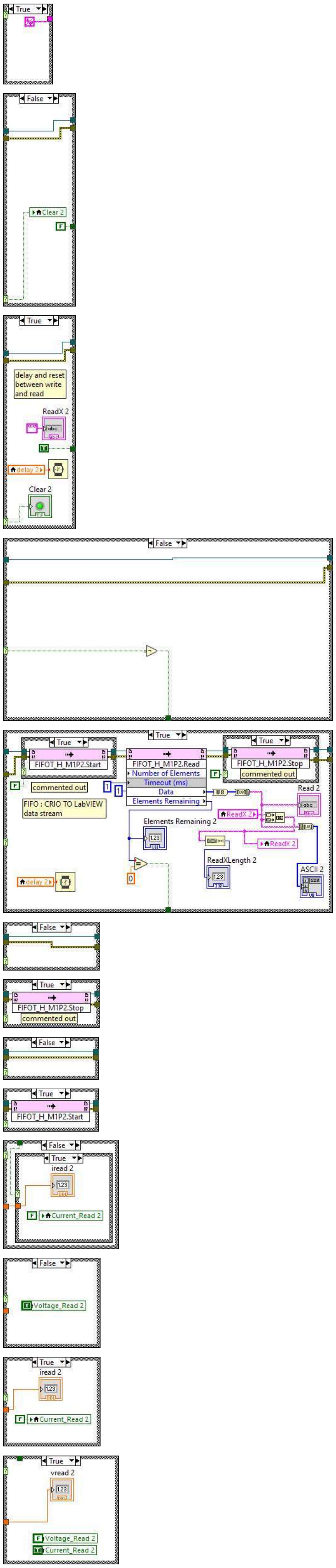

False
MEAS:CURR?

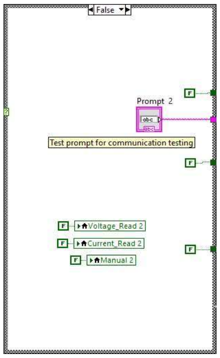

True
MEAS:VOLT?

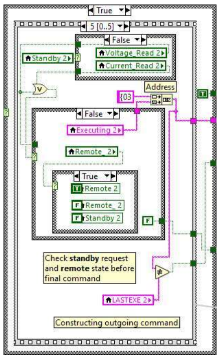

False
Current_Read
Voltage_Read

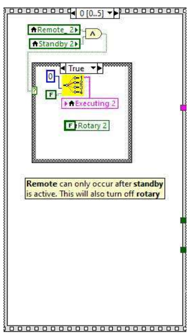

True
F
F
Current_Read
F
F
Voltage_Read

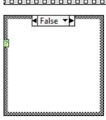

Structure is similar to previous page of this stacked sequence
This block contains instruction formatting, writing SCPI command and reading status using NI9871 RS 485 communication
Ideas
Autonomous control block is placed for voltage control only. This can be applied for the current control by swaping voltage - current roles.
Here, ramping up voltage is considered only. Again, voltage current roles needs to be swaped for enabling ramping current instead.
Current_2
step 2
Voltage_2
COMM 2
Auto 2
Mode 2
Remote_2
Arm_2
Volt 2
Standby_2
Current 2
Prompt 2
Test prompt for communication testing
WRITE 2
READ 2
Voltage_Read 2
Current_Read 2
Manual 2
FIFOH_T_M1P2.Start
FIFOH_T_M1P2.Write
FIFOH_T_M1P2.Stop
Data
Timeout (ms)
Empty Elements Remaining
commented out
FIFO : LabVIEW TO CRIO data stream
length 2
Empty Elements Remaining 2
LASTEXE 2
LAST Executed instruction
%s\n\r
delay and reset between write and read
ReadX 2
delay 2
Clear 2
FIFOT_H_M1P2.Start
FIFOT_H_M1P2.Read
FIFOT_H_M1P2.Stop
Number of Elements
Elements Remaining
FIFO : CRIO TO LabVIEW data stream
Elements Remaining 2
Read 2
ReadXLength 2
ASCII 2
The loop reads until zero elements remaining
COMM 2
Clear 2
update reading status
Numerical read conversion
vread 2
Voltage control occurs when Arm is true and Current mode is off (avoid turning on arm and voltage togather)
True
False
1 [0..4]

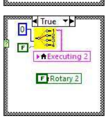

True
False
Clear 2
delay and reset between write and read
ReadX 2
delay 2
FIFOT_H_M1P2.Start
FIFOT_H_M1P2.Read
Number of Elements
Timeout (ms)
Data
Elements Remaining
FIFOT_H_M1P2.Stop
commented out
FIFO : CRIO TO LabVIEW data stream
Read 2
Elements Remaining 2
ReadXLength 2
ASCII 2
iread 2
Current_Read 2
Voltage_Read 2
vread 2

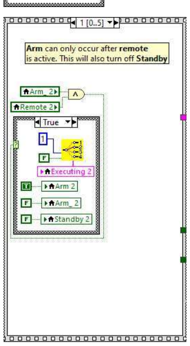

False
Prompt 2
Test prompt for communication testing
Voltage_Read 2
Current_Read 2
Manual 2

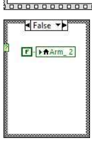

True
5 [0..5]
Standby 2
False
Voltage_Read 2
Current_Read 2
Address
03
False
Executing 2
Remote_ 2
True
Remote 2
Remote_ 2
Standby 2
Check standby request and remote state before final command
LASTEXE 2
Constructing outgoing command

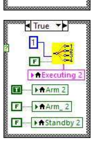

0 [0..5]
Remote_ 2
Standby 2
True
0
Executing 2
Rotary 2
Remote can only occur after standby is active. This will also turn off rotary

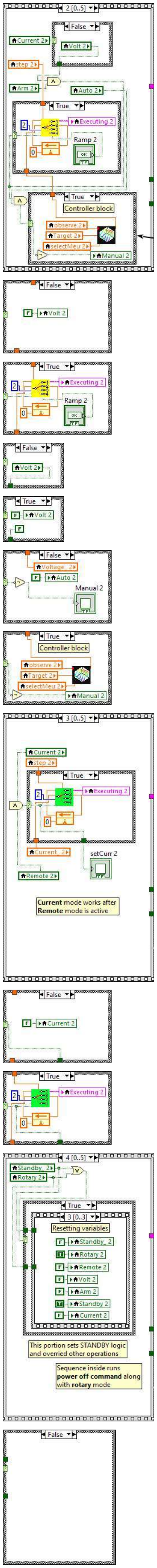

False


True
0
Executing 2
Rotary 2

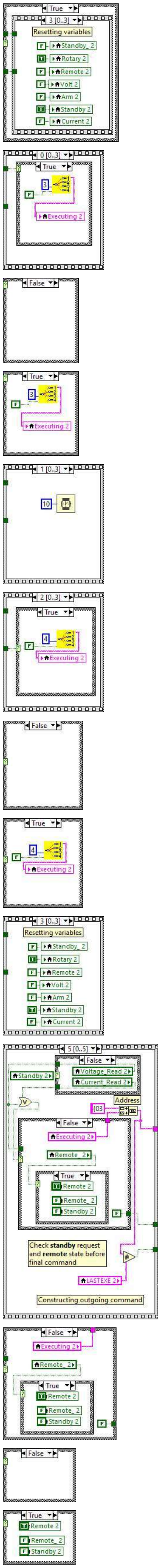

1 [0..5]
Arm can only occur after remote is active. This will also turn off Standby
Arm_ 2
Remote 2
True
1
Executing 2
Arm 2
Arm_ 2
Standby 2

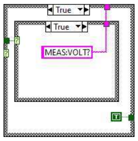

False
Arm_ 2

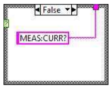

True
1
Executing 2
Arm 2
Arm_ 2
Standby 2

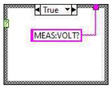


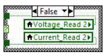
2 [0..5]
Current 2
step 2
Arm 2
Auto 2
Executing 2
Ramp 2
Controller block
observe 2
Target 2
selectMeu 2
Manual 2
Volt 2
Voltage_2
3 [0..5]
Current_2
setCurr 2
Remote 2
Current mode works after Remote mode is active
4 [0..5]
Standby_2
Rotary 2
3 [0..3]
Resetting variables
Standby 2
This portion sets STANDBY logic and overrided other operations
Sequence inside runs power off command along with rotary mode

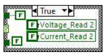

True
3 [0..3]
Resetting variables
Standby_ 2
Rotary 2
Remote 2
Volt 2
Arm 2
Standby 2
Current 2
0 [0..3]
True
Executing 2
False
True
Executing 2
1 [0..3]
10
2 [0..3]
True
4
Executing 2
False
True
4
Executing 2
3 [0..3]
Resetting variables
Standby_ 2
Rotary 2
Remote 2
Volt 2
Arm 2
Standby 2
Current 2
5 [0..5]
False
Standby 2
Voltage_Read 2
Current_Read 2
Address
False
Executing 2
Remote_ 2
True
Remote 2
Remote_ 2
Standby 2
Check standby request and remote state before final command
LASTEXE 2
Constructing outgoing command
False
Executing 2
Remote_ 2
True
Remote 2
Remote_ 2
Standby 2
False
True
Remote 2
Remote_ 2
Standby 2

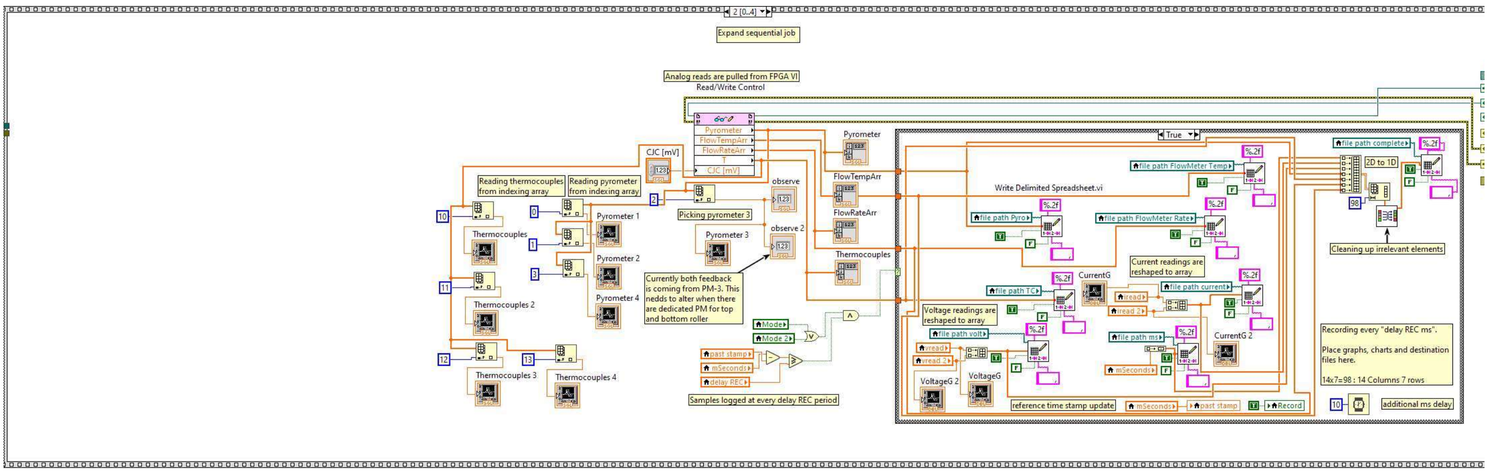

True
True
MEAS:VOLT?

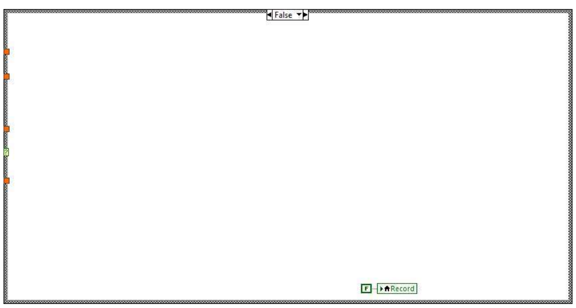

False
MEAS:CURR?

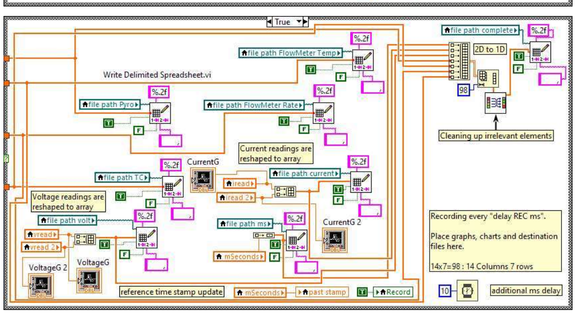

True
MEAS:VOLT?


False
Voltage_Read 2
Current_Read 2


True
Voltage_Read 2
Current_Read 2


2 [0..4]
Expand sequential job
Analog reads are pulled from FPGA VI
Read/Write Control
Pyrometer
FlowTempArr
FlowRateArr
CJC [mV]
Reading thermocouples from indexing array
Reading pyrometer from indexing array
Picking pyrometer 3
observe
observe 2
Thermocouples
Thermocouples 2
Thermocouples 3
Thermocouples 4
Pyrometer 1
Pyrometer 2
Pyrometer 3
Pyrometer 4
Currently both feedback is coming from PM-3. This nedds to alter when there are dedicated PM for top and bottom roller
Mode
Mode 2
past stamp
mSeconds
delay REC
Samples logged at every delay REC period
True
Write Delimited Spreadsheet.vi
file path Pyro
file path FlowMeter Temp
file path FlowMeter Rate
file path TC
file path current
file path volt
file path ms
file path complete
2D to 1D
Cleaning up irrelevant elements
Current readings are reshaped to array
Voltage readings are reshaped to array
CurrentG
CurrentG 2
VoltageG
VoltageG 2
read
read 2
reference time stamp update
Record
Recording every "delay REC ms".
Place graphs, charts and destination files here.
14x7=98 : 14 Columns 7 rows
additional ms delay


False
Record


True
Write Delimited Spreadsheet.vi
file path FlowMeter Temp
file path Pyro
file path FlowMeter Rate
file path TC
file path current
file path volt
file path ms
file path complete
2D to 1D
Cleaning up irrelevant elements
Current readings are reshaped to array
Voltage readings are reshaped to array
CurrentG
CurrentG 2
VoltageG
VoltageG 2
read
read 2
mSeconds
past stamp
reference time stamp update
Record
Recording every "delay REC ms".
Place graphs, charts and destination files here.
14x7=98 : 14 Columns 7 rows
additional ms delay

SON.vi

C:\Users\inladmin\Desktop\cRIO_test_VI\EXPLORE\SON.vi

commands.vi

C:\Users\inladmin\Desktop\cRIO_test_VI\EXPLORE\analog\commands.vi

commands2.vi

C:\Users\inladmin\Desktop\cRIO_test_VI\EXPLORE\analog\commands2.vi

controller0.vi

C:\Users\inladmin\Desktop\cRIO_test_VI\EXPLORE\analog\controller0.vi

Write Delimited Spreadsheet.vi

C:\Program Files\National Instruments\LabVIEW 2023\vi.lib\Utility\file.llb\Write Delimited Spreadsheet.vi

trimArr.vi

C:\Users\inladmin\Desktop\cRIO_test_VI\EXPLORE\analog\trimArr.vi

Write Delimited Spreadsheet (DBL).vi

C:\Program Files\National Instruments\LabVIEW 2023\vi.lib\Utility\file.llb\Write Delimited Spreadsheet (DBL).vi

 **Error Cluster From Error Code.vi**

C:\Program Files\National Instruments\LabVIEW 2023\vi.lib\Utility\error.llb\Error Cluster From Error Code.vi

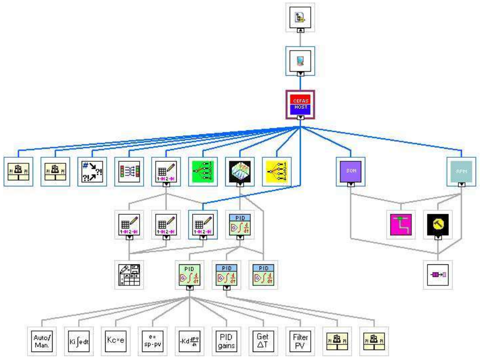
 **rpm.vi**

C:\Users\inladmin\Desktop\cRIO_test_VI\EXPLORE\rpm.vi

"CEFAS_HOST.vi History"

Current Revision: 129

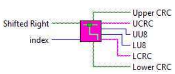
Position in Hierarchy

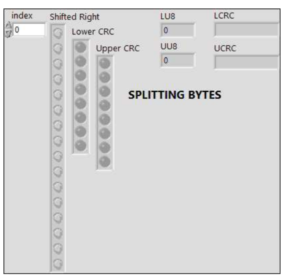


Iconified Cluster Constants

---

u8Split.vi





Shifted Right

index

Lower CRC

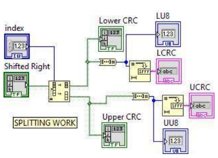
Upper CRC

LCRC

UCRC

LU8

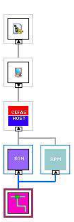
UU8



"u8Split.vi History"

Current Revision: 7

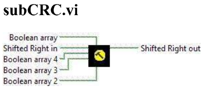
Position in Hierarchy

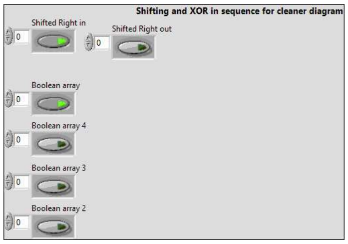


Iconified Cluster Constants

---

subCRC.vi





Boolean array

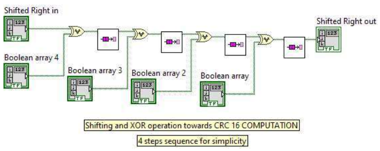
Boolean array 2

Boolean array 3

Shifted Right in

Boolean array 4

Shifted Right out



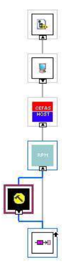
shiftR.vi

C:\Users\inladmin\Desktop\cRIO_test_VI\EXPLORE\shiftR.vi

"subCRC.vi History"

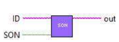
Current Revision: 7

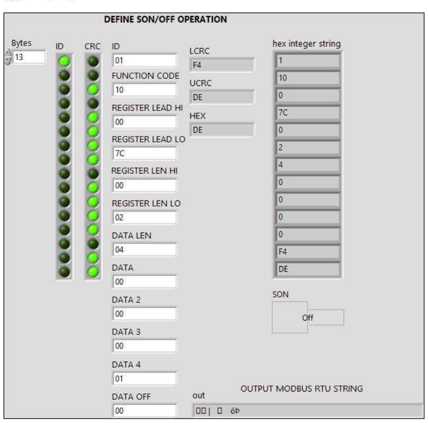
Position in Hierarchy



Iconified Cluster Constants

SON.vi





ID
FUNCTION CODE
REGISTER LEAD HI
REGISTER LEAD LO
REGISTER LEN HI
REGISTER LEN LO
DATA LEN
DATA
DATA 2
DATA 3
DATA 4
Bytes
SON
DATA OFF
CRC

ID

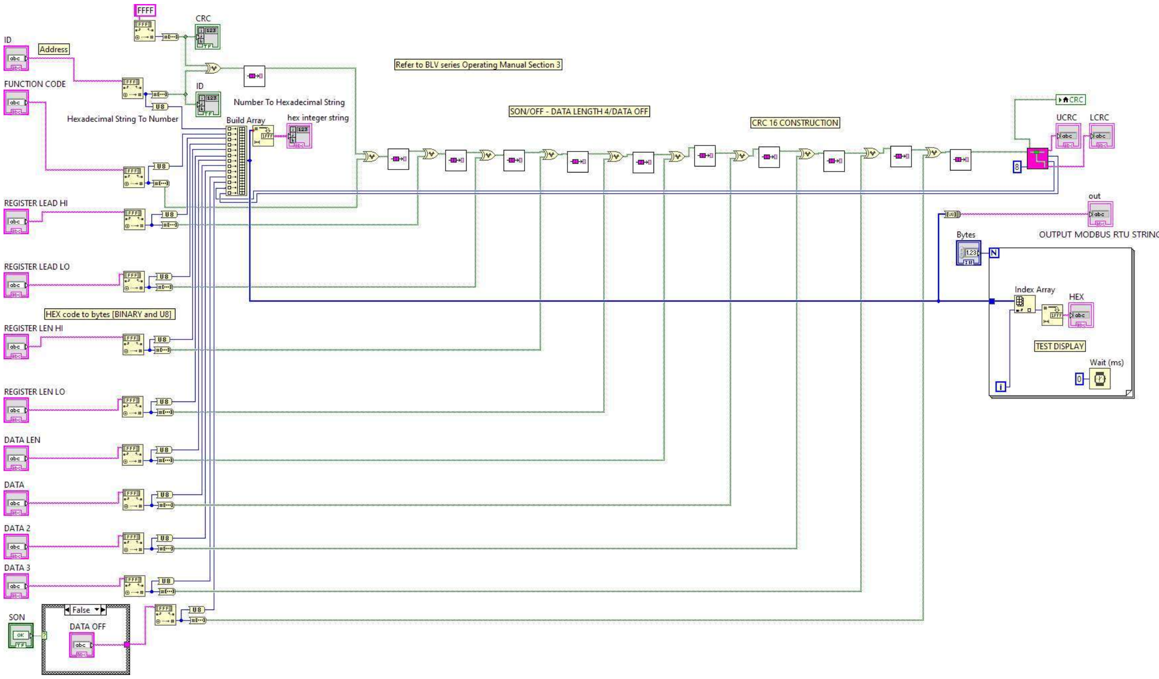
hex integer string

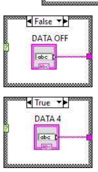
LCRC

UCRC

HEX

out





u8Split.vi

C:\Users\inladmin\Desktop\cRIO_test_VI\EXPLORE\u8Split.vi

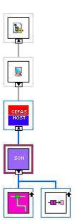
shiftR.vi

C:\Users\inladmin\Desktop\cRIO_test_VI\EXPLORE\shiftR.vi

"SON.vi History"

Current Revision: 23

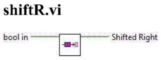
Position in Hierarchy

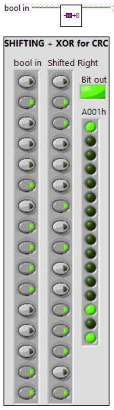


Iconified Cluster Constants

---

shiftR.vi



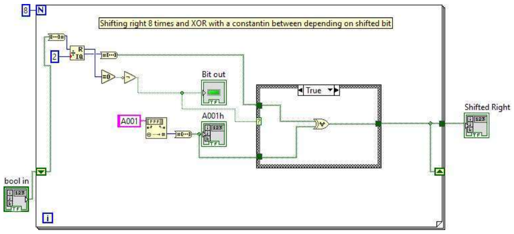


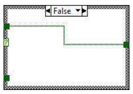
bool in

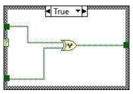
Bit out

Shifted Right

A001h



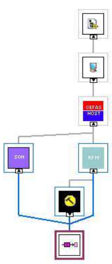




"shiftR.vi History"

Current Revision: 9

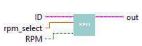
Position in Hierarchy

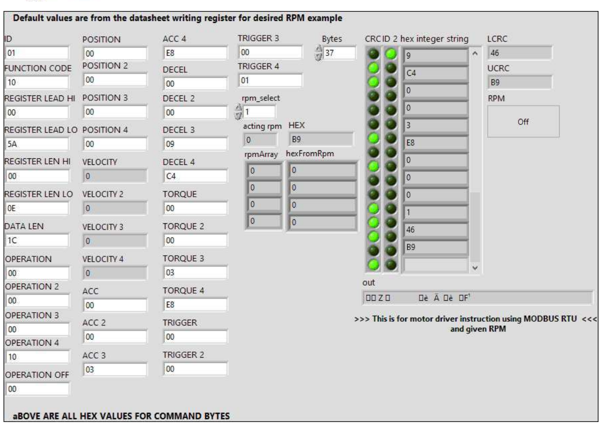


Iconified Cluster Constants

---

rpm.vi





ID
FUNCTION CODE
REGISTER LEAD HI
REGISTER LEAD LO
REGISTER LEN HI
REGISTER LEN LO
DATA LEN
OPERATION
OPERATION 2
OPERATION 3
Bytes
RPM
OPERATION OFF
OPERATION 4
POSITION
POSITION 2
POSITION 3
POSITION 4
ACC
DECEL
TORQUE
TRIGGER
ACC 2
ACC 3
ACC 4
DECEL 2
DECEL 3
DECEL 4
TORQUE 2
TORQUE 3
TORQUE 4
TRIGGER 2
TRIGGER 3

TRIGGER 4
rpm_select
CRC

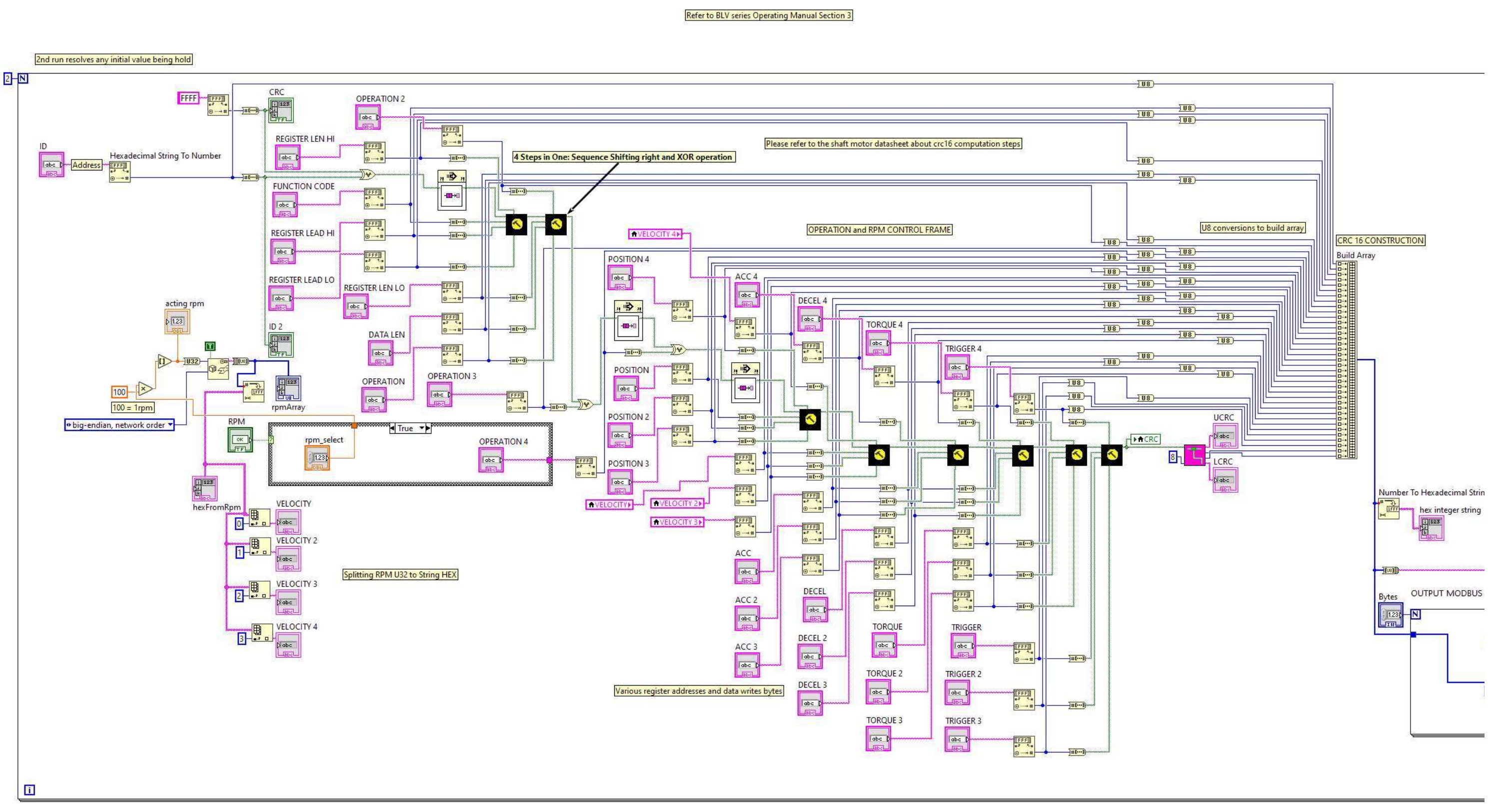
ID 2

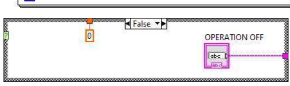
hex integer string

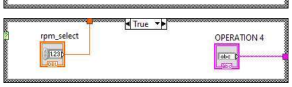
LCRC
UCRC
HEX
out
acting rpm
rpmArray

hexFromRpm

VELOCITY
VELOCITY 2
VELOCITY 3
VELOCITY 4







u8Split.vi

C:\Users\inladmin\Desktop\cRIO_test_VI\EXPLORE\u8Split.vi

subCRC.vi

C:\Users\inladmin\Desktop\cRIO_test_VI\EXPLORE\subCRC.vi

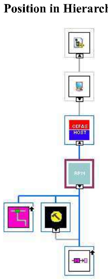
shiftR.vi

C:\Users\inladmin\Desktop\cRIO_test_VI\EXPLORE\shiftR.vi

"rpm.vi History"

Current Revision: 33

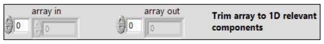
Position in Hierarchy



Iconified Cluster Constants

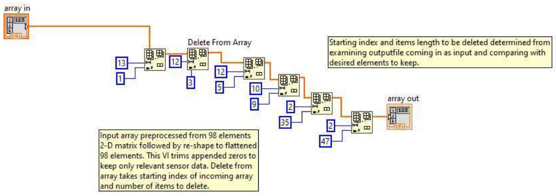
---

trimArr.vi



array in

array out



"trimArr.vi History"

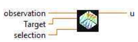
Current Revision: 6

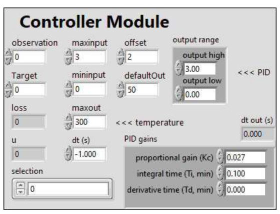
Position in Hierarchy

Iconified Cluster Constants

---

controller0.vi





observation
Target
maxinput
mininput
maxout
offset
defaultOut
selection
output range

output range specifies the range to which to coerce the control output. The default range is -100 to 100, which corresponds to values specified in terms of percentage of full scale.

output high

output high specifies the maximum value of the controller output. The default is 100.

output low

output low specifies the minimum value of the controller output. The default is -100.

dt (s)

dt (s) specifies the loop-cycle time, or interval in seconds, at which this VI is called. If **dt (s)** is less than or equal to zero, this VI calculates the time since it was last called using an internal timer with 1 ms resolution. If **dt (s)** must be less than 1 ms, specify the value explicitly. The default is -1.

PID gains

PID gains specifies the proportional gain, integral time, and derivative time parameters of the controller.

proportional gain (Kc)

proportional gain (Kc) specifies the proportional gain of the controller. The default is 1.

integral time (Ti, min)

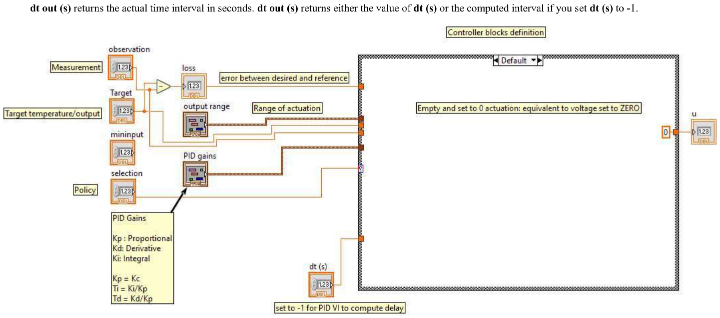
integral time (Ti, min) specifies the integral time in minutes. The default is 0.01.

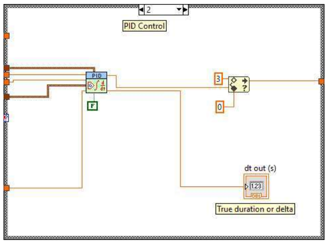
derivative time (Td, min)

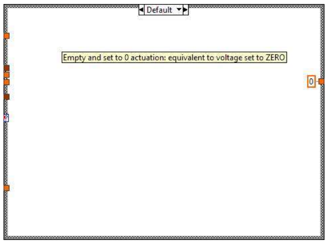
derivative time (Td, min) specifies the derivative time in minutes. The default is 0.

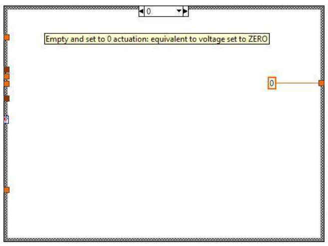
loss
u
dt out (s)

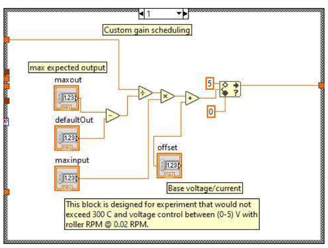
dt out (s) returns the actual time interval in seconds. **dt out (s)** returns either the value of **dt (s)** or the computed interval if you set **dt (s)** to -1.











NI_PID_pid.lvlib:PID.vi

C:\Program Files\National Instruments\LabVIEW 2023\vi.lib\addons\control\pid\pid.llb\PID.vi

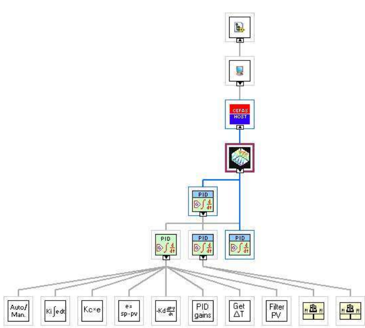
NI_PID_pid.lvlib:PID (DBL).vi

C:\Program Files\National Instruments\LabVIEW 2023\vi.lib\addons\control\pid\pid.llb\PID (DBL).vi

"controller0.vi History"

Current Revision: 8

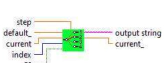
Position in Hierarchy



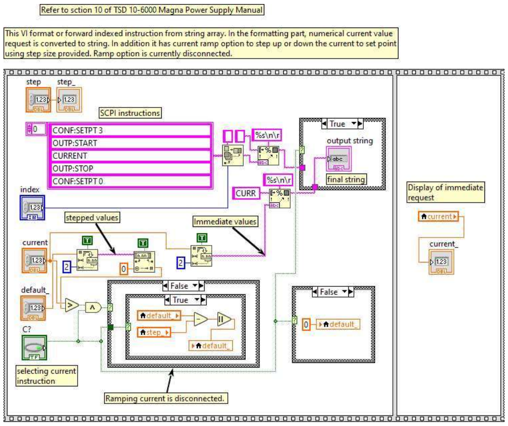
Iconified Cluster Constants

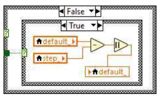
---

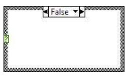
commands2.vi

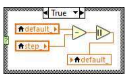


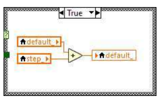
step
C?
default_
current
index
output string
step_
current_

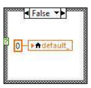


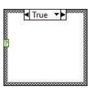


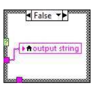


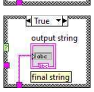








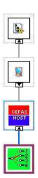




"commands2.vi History"

Current Revision: 8

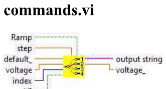
Position in Hierarchy

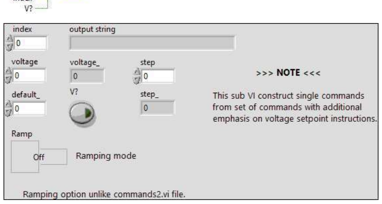


Iconified Cluster Constants

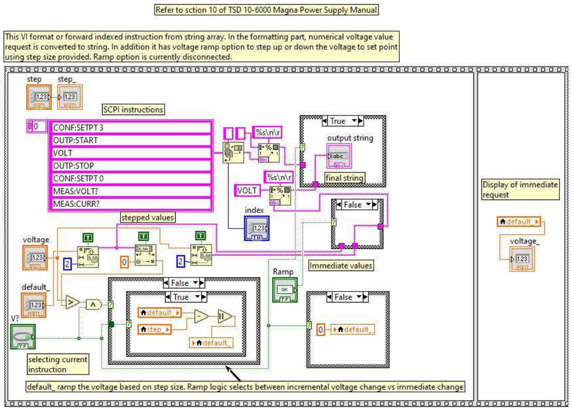
---

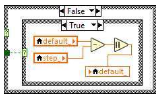
commands.vi

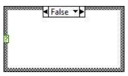


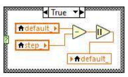


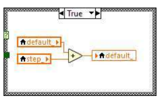
index
voltage
default_
V?
step
Ramp
output string
voltage_
step_

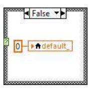


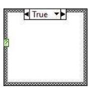


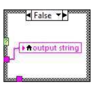


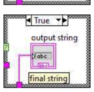


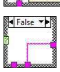


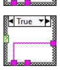












"commands.vi History"

Current Revision: 21

Position in Hierarchy

Iconified Cluster Constants

Table of Contents

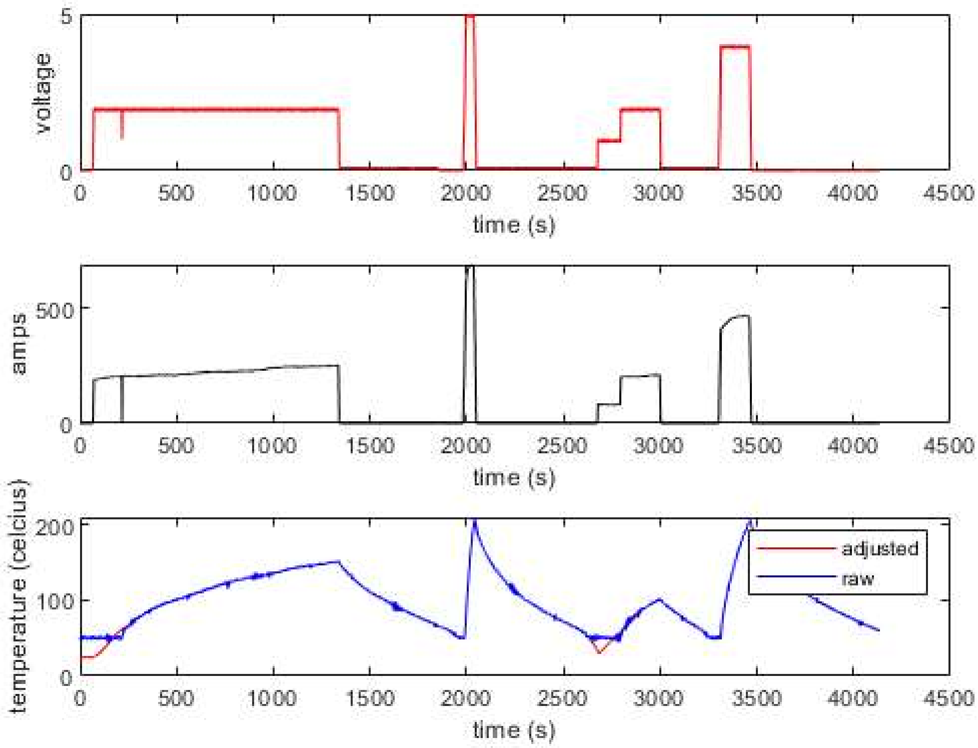
CEFAS SID : Golam Gause Jaman

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%             CEFAS SID             %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clc; clear all; close all;
```

Load Data

```
matrix0=readmatrix('sidCEFAS.xlsx');
time=matrix0(1:3369,2);
volt=matrix0(1:3369,3);
amps=matrix0(1:3369,4);
temp=matrix0(1:3369,5);
temp_=temp;
x=linspace(-1,1,140);
temp_(61:200)=(((tanh(x)+0.8)/2)*50)+25;
temp_(1:60)=25; %room temp.
p=polyfit(time(2100:2150),temp(2100:2150),2);
f=polyval(p,time(2150:2200));
temp_(2150:2200)=f;

q=polyfit(time(2300:2450),temp(2300:2450),1);
g=polyval(q,time(2201:2400));
temp_(2201:2400)=g;

figure(1)
subplot(3,1,1);
plot(time,volt,'r');
xlabel('time (s)');
ylabel('voltage');

subplot(3,1,2);
plot(time,amps,'k');
xlabel('time (s)');
ylabel('amps');

subplot(3,1,3);
plot(time,temp_,'r',time,temp,'b');
xlabel('time (s)');
ylabel('temperature (celcius)');
```

```
legend('adjusted','raw');

Ts=mean(time(2:end,1)-time(1:end-1,1));
gamma=4;
```



SID Construction

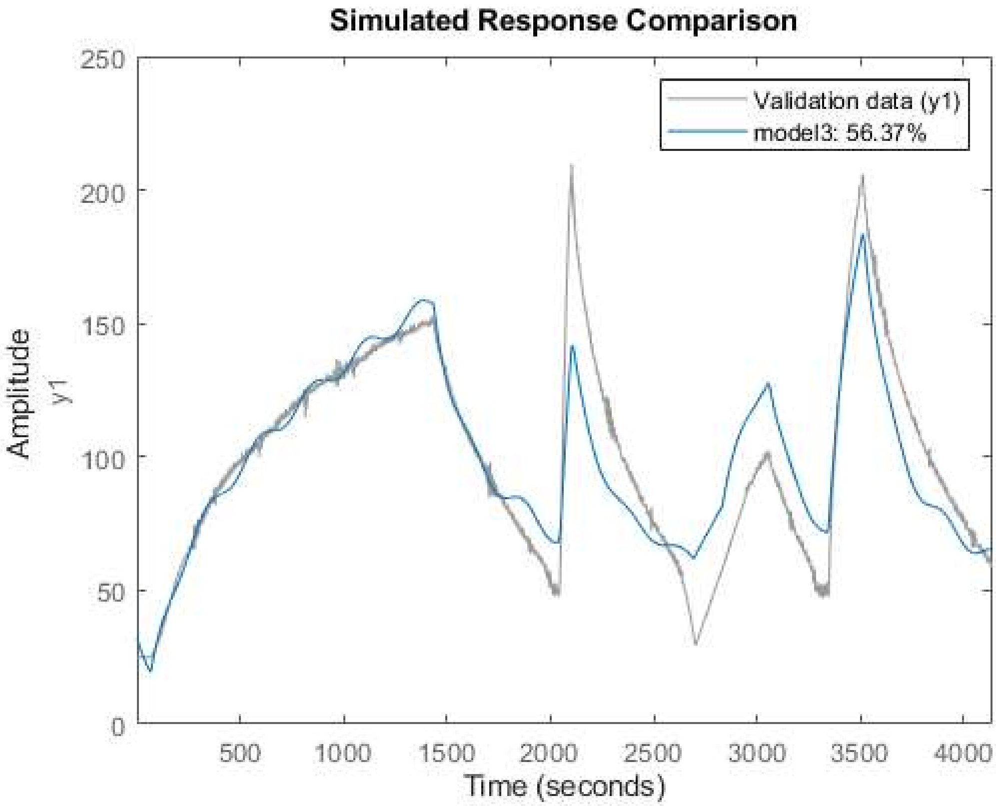
```
u_=volt;
y_=temp_;
data_=iddata(y_,u_,Ts);

nx=[1:10];
sys_d=n4sid(data_, nx)
sys=d2c(sys_d)
[num,den]=ss2tf(sys.A, sys.B, sys.C, sys.D);
model0=tf(num,den)

poles_=length(eig(model0));
model1=tfest(data_, poles_)
model2=c2d(model0,Ts)

% Best model (validation fit%) appeared : transfer function (4 pole, 3 zero)
% and NARX

model3=tfest(data_, 4)
figure(2)
```

```
%compare(data_,sys_d)
compare(data_,model3)
model4=c2d(model3,Ts)
```

```
sys_d =
  Discrete-time identified state-space model:
    x(t+Ts) = A x(t) + B u(t) + K e(t)
       y(t) = C x(t) + D u(t) + e(t)

  A =
             x1
   x1    0.9963

  B =
                u1
   x1    8.566e-05

  C =
           x1
   y1    4364

  D =
         u1
   y1     0

  K =
                y1
   x1    6.846e-05

Sample time: 1.2279 seconds

Parameterization:
   FREE form (all coefficients in A, B, C free).
   Feedthrough: none
   Disturbance component: estimate
   Number of free coefficients: 4
   Use "idssdata", "getpvec", "getcov" for parameters and their
uncertainties.

Status:
Estimated using N4SID on time domain data "data_".
Fit to estimation data: 95.6% (prediction focus)
FPE: 2.908, MSE: 2.901


sys =
  Continuous-time identified state-space model:
      dx/dt = A x(t) + B u(t) + K e(t)
       y(t) = C x(t) + D u(t) + e(t)

  A =
                x1
   x1    -0.002991
```

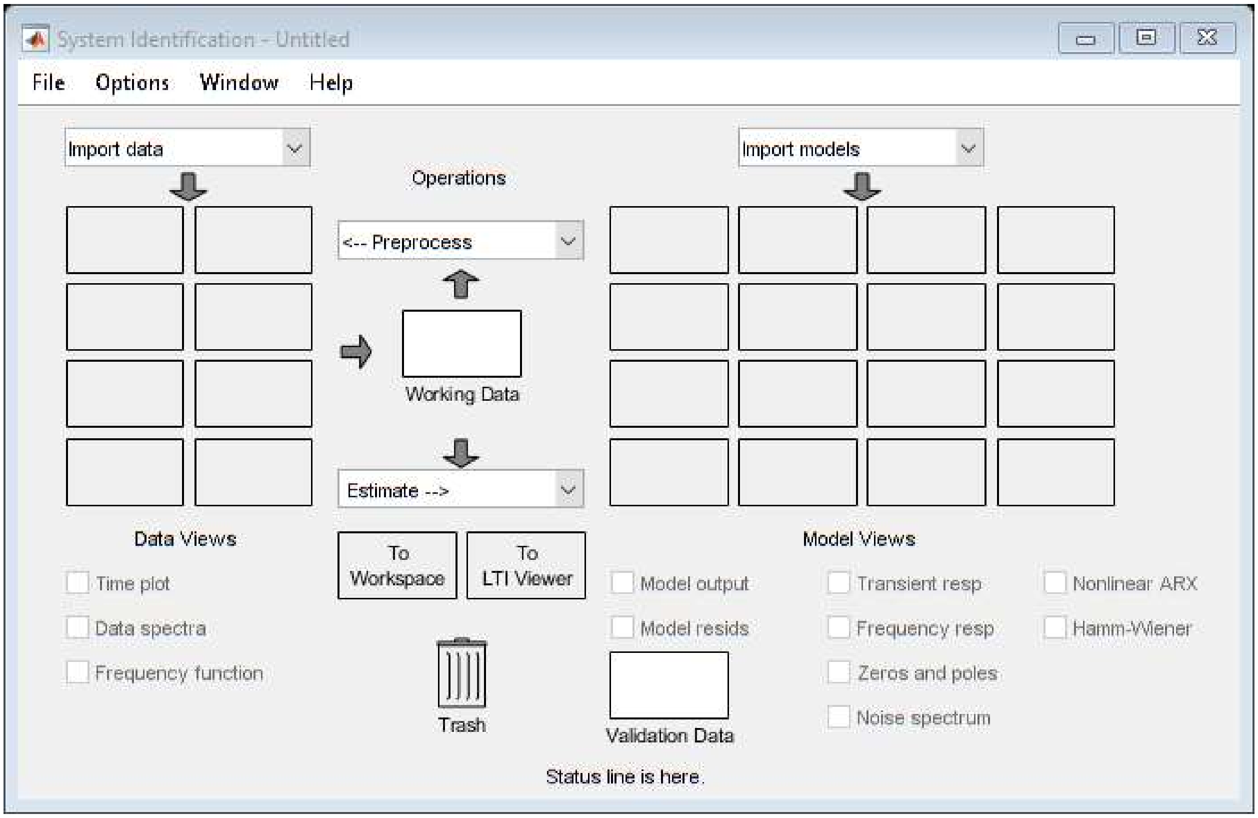
```
  B =
              u1
   x1  6.989e-05

  C =
         x1
   y1  4364

  D =
       u1
   y1   0

  K =
              y1
   x1  5.586e-05

Parameterization:
   FREE form (all coefficients in A, B, C free).
   Feedthrough: none
   Disturbance component: estimate
   Number of free coefficients: 4
   Use "idssdata", "getpvec", "getcov" for parameters and their
uncertainties.

Status:
Created by direct construction or transformation. Not estimated.


model0 =

     0.305
  ------------
  s + 0.002991

Continuous-time transfer function.

model1 =
  From input "u1" to output "y1":
     0.1312
  ------------
  s + 0.001294

Continuous-time identified transfer function.

Parameterization:
   Number of poles: 1   Number of zeros: 0
   Number of free coefficients: 2
   Use "tfdata", "getpvec", "getcov" for parameters and their uncertainties.

Status:
Estimated using TFEST on time domain data "data_".
Fit to estimation data: 39.67%
FPE: 545.3, MSE: 544.4
```

```
model2 =

    0.3738
  ----------
  z - 0.9963

Sample time: 1.2279 seconds
Discrete-time transfer function.

model3 =
  From input "u1" to output "y1":
      0.3171 s^3 + 0.0005216 s^2 + 0.0001898 s + 3.939e-07
  -------------------------------------------------------------
  s^4 + 0.009076 s^3 + 0.0006072 s^2 + 5.461e-06 s + 3.331e-09

Continuous-time identified transfer function.

Parameterization:
   Number of poles: 4   Number of zeros: 3
   Number of free coefficients: 8
   Use "tfdata", "getpvec", "getcov" for parameters and their uncertainties.

Status:
Estimated using TFEST on time domain data "data_".
Fit to estimation data: 56.37%
FPE: 286.7, MSE: 284.7


model4 =
  From input "u1" to output "y1":
   0.3876 z^-1 - 1.162 z^-2 + 1.161 z^-3 - 0.3868 z^-4
  -------------------------------------------------------
  1 - 3.988 z^-1 + 5.965 z^-2 - 3.966 z^-3 + 0.9889 z^-4

Sample time: 1.2279 seconds
Discrete-time identified transfer function.

Parameterization:
   Number of poles: 4   Number of zeros: 4
   Number of free coefficients: 8
   Use "tfdata", "getpvec", "getcov" for parameters and their uncertainties.

Status:
Created by direct construction or transformation. Not estimated.
```



Apply SID GUI to find best model and apply in simulink defining the PID/LQG/RL controller.

```
systemIdentification
%nntart
```

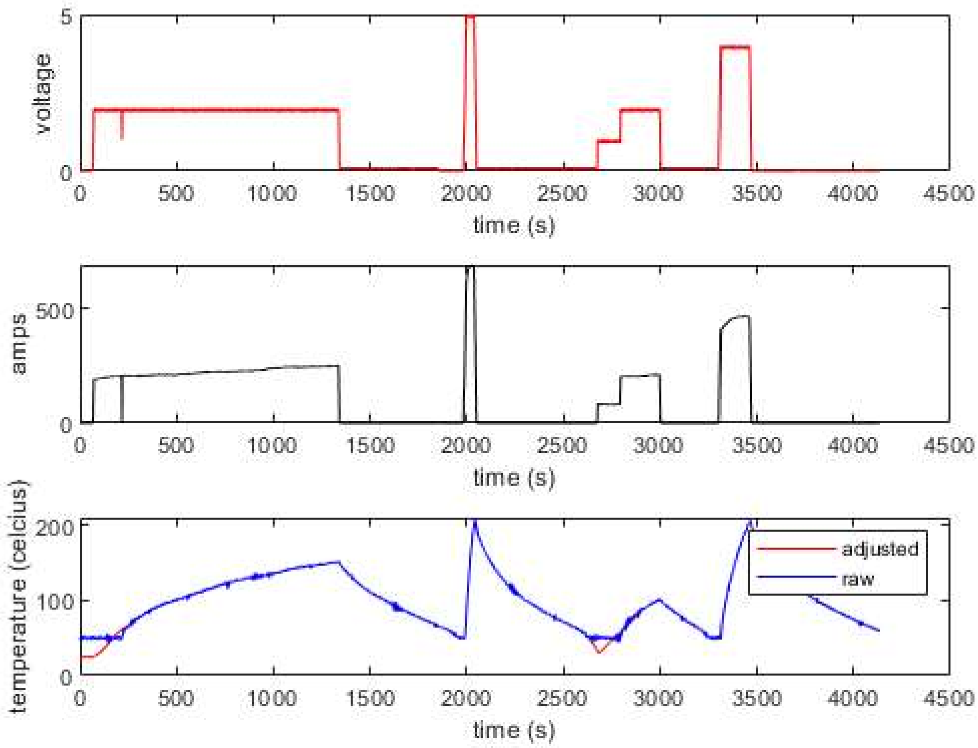


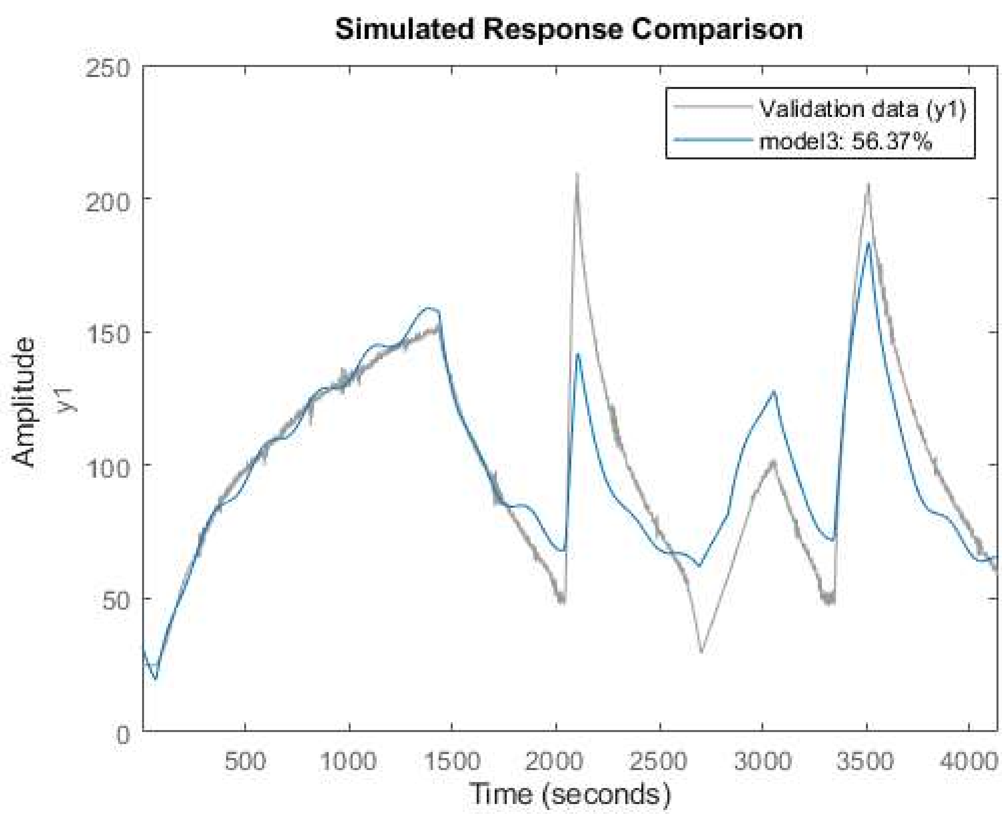
Evaluate continuous plant

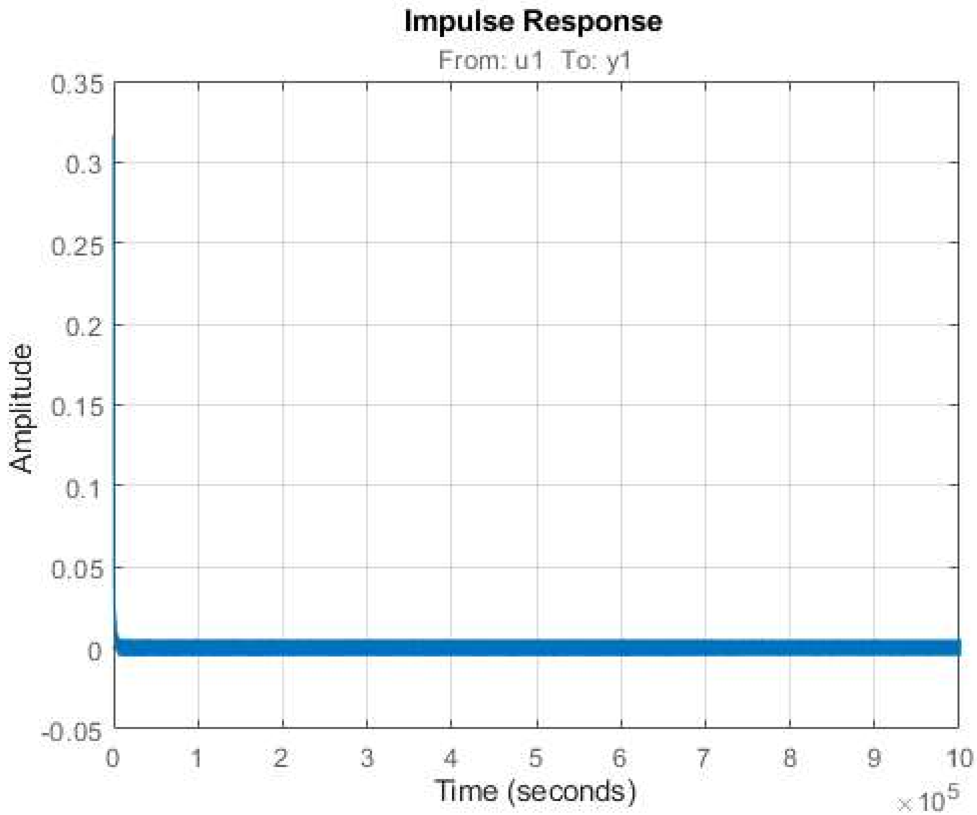
```
EIG=eig(model3)
% Impulse response
figure(3);
impulse(model3);
grid on;
% Step response
figure(4);
step(model4);
grid on;
% Margin (Bode)
figure(5);
margin(model3);
grid on;
% PZ-map
figure(6);
pzmap(model3);
grid on;
% Nyquist plot
figure(7);
nyquist(model3);
grid on;


Fs=1/Ts;
L=length(time);
```

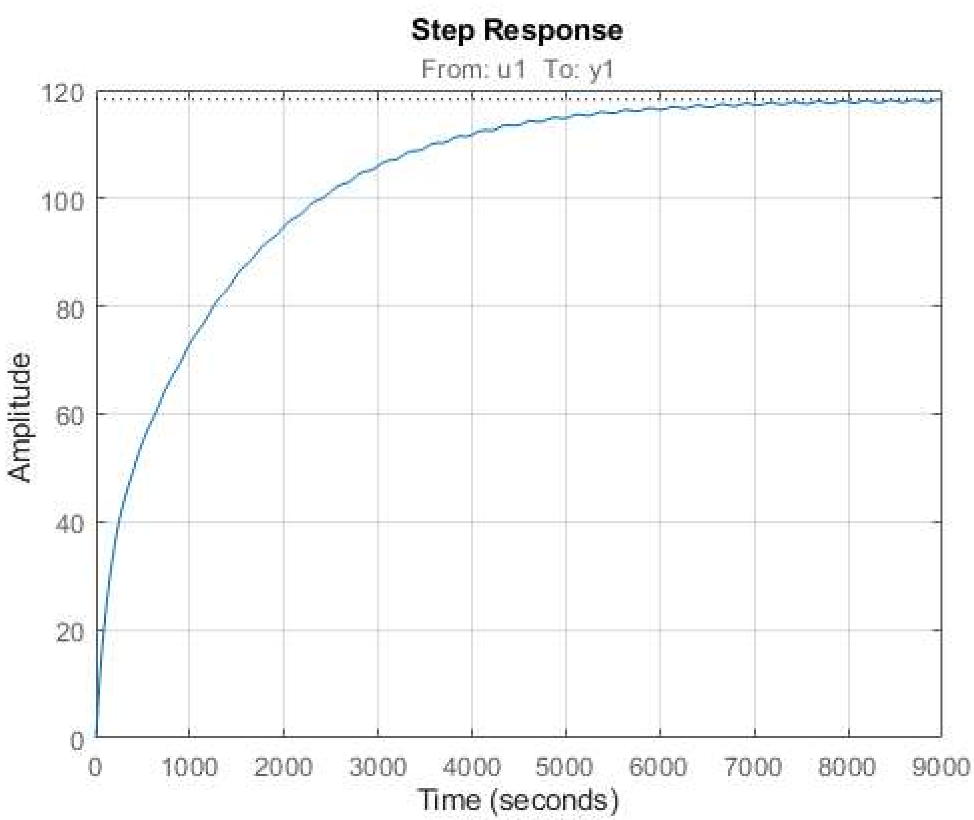
```
Freq=(Fs/L)*(0:L-1);

Y_=fft(y_);
figure(8)
plot(Freq(500:end-500), abs(Y_(500:end-500)), 'k', 'linewidth', 1);
xlabel('Hz');
ylabel('|fft(y)|');
title('FFT of y')
grid on;

U_=fft(u_);
figure(9)
plot(Freq(500:end-500), abs(U_(500:end-500)), 'b', 'linewidth', 1);
xlabel('Hz');
ylabel('|fft(u)|');
title('FFT of u')
grid on;

sysN0=nlarx(data_,[10,10,1]);
figure(10)
compare(data_,sysN0)

sysN1=arx(data_,[10,10,1]);
figure(11)
compare(data_,sysN1)

sysN2=oe(data_,[10,10,1]);
figure(12)
compare(data_,sysN2)
```

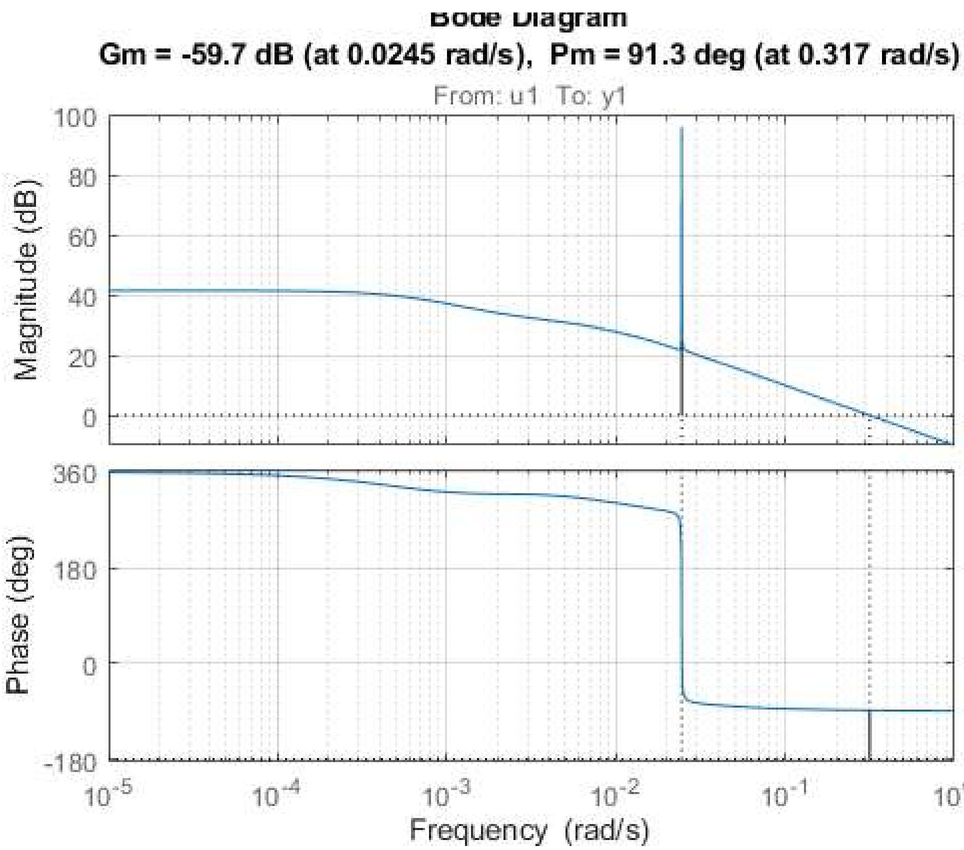
```
EIG =

  -0.0000 + 0.0245i
  -0.0000 - 0.0245i
  -0.0084 + 0.0000i
  -0.0007 + 0.0000i
```

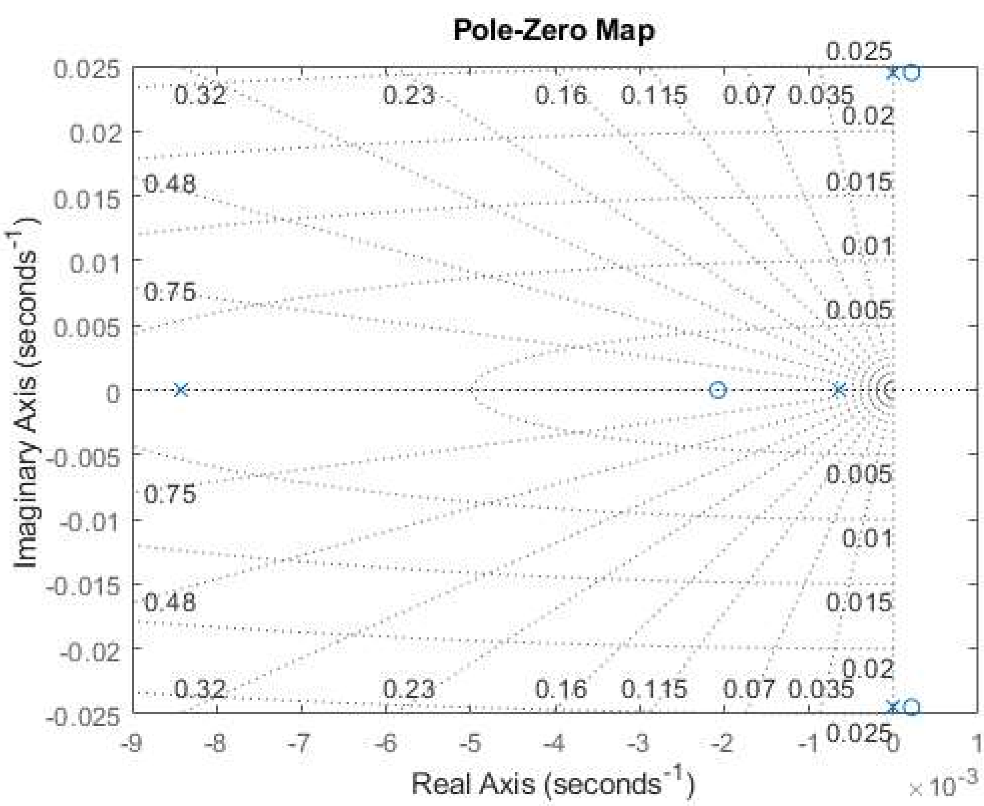

voltage
amps
temperature (celcius)
time (s)
adjusted
raw

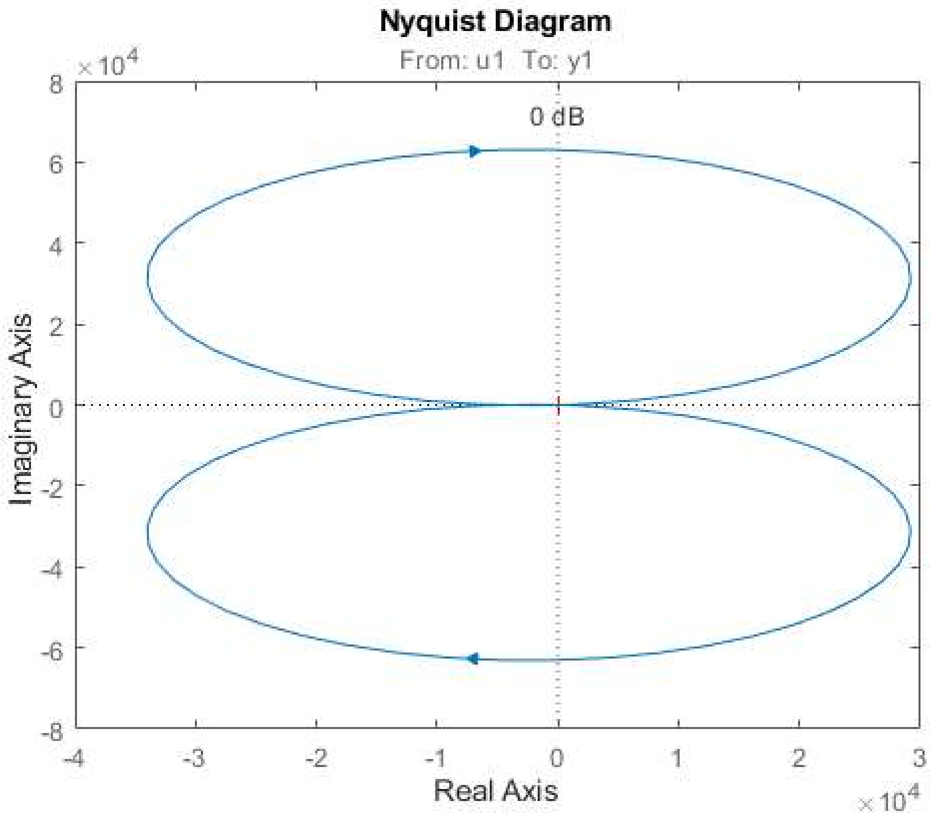

Simulated Response Comparison
Validation data (y1)
model3: 56.37%
Amplitude
y1
Time (seconds)

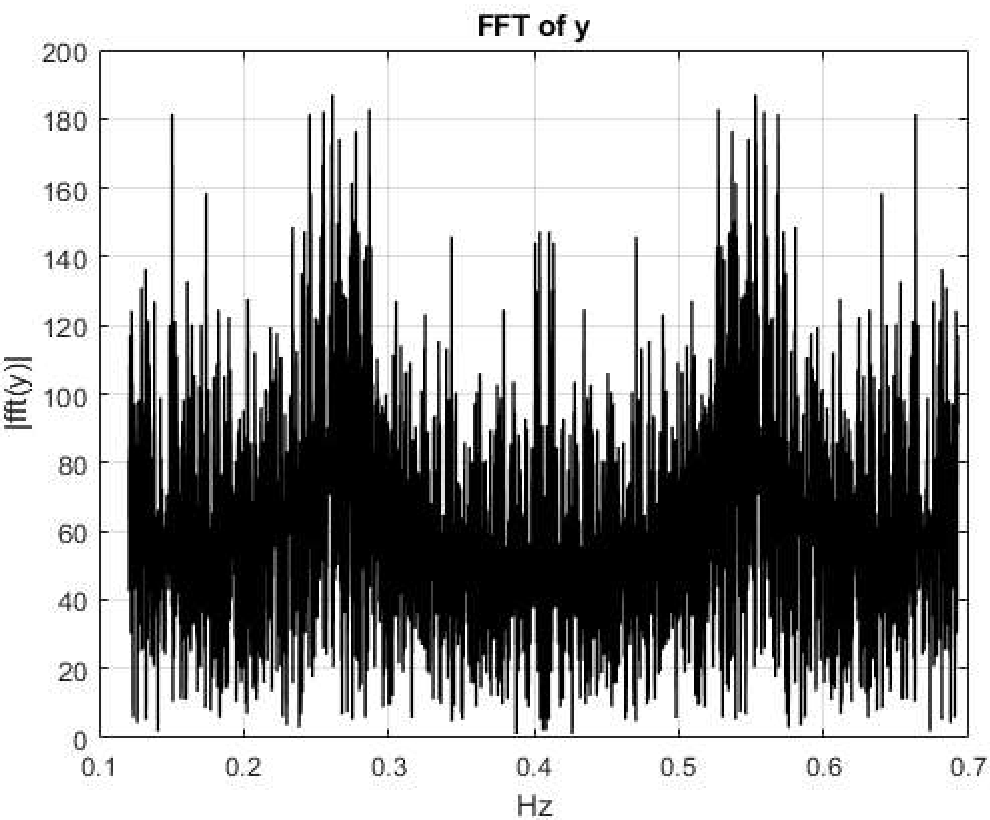

Impulse Response
From: u1 To: y1
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
-0.05
Amplitude
0 1 2 3 4 5 6 7 8 9 10
Time (seconds)
×10^5

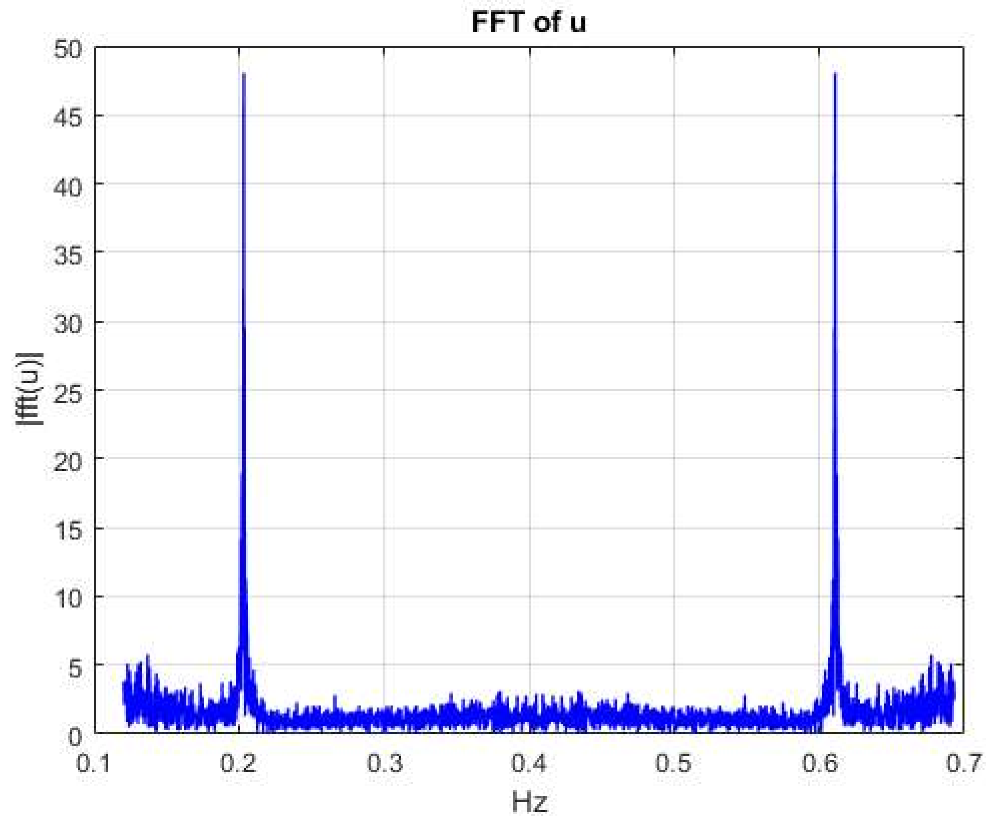

Step Response
From: u1 To: y1
120
100
80
60
40
20
0
Amplitude
0 1000 2000 3000 4000 5000 6000 7000 8000 9000
Time (seconds)

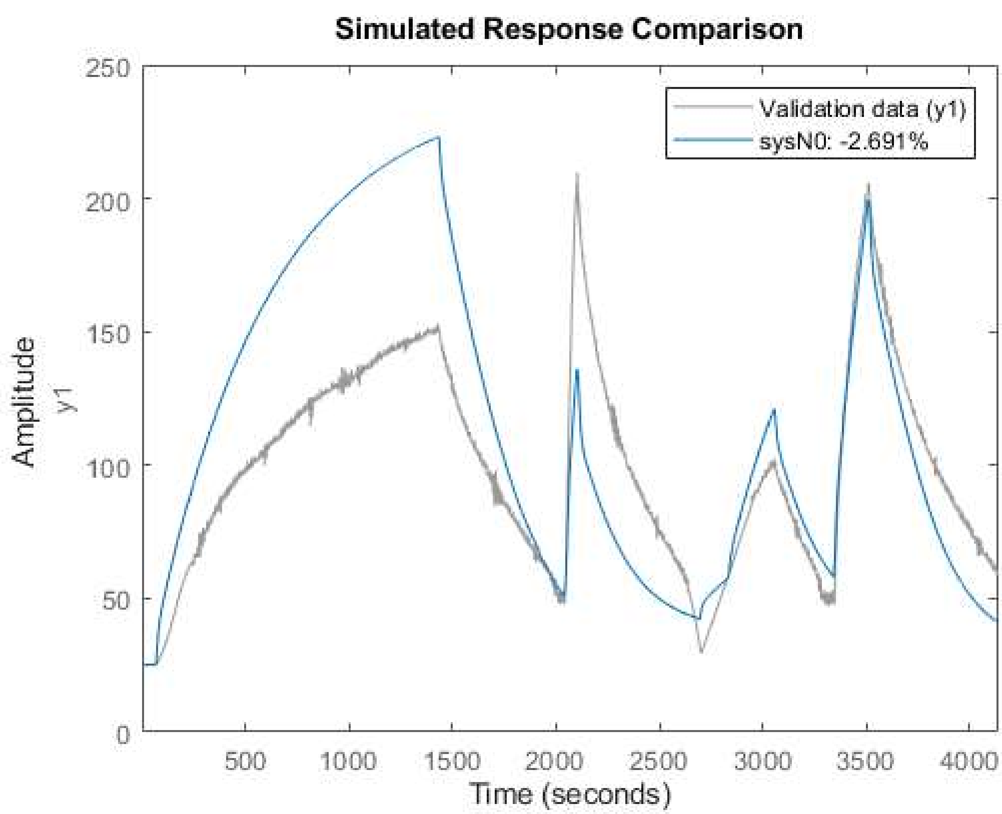

Bode Diagram
Gm = -59.7 dB (at 0.0245 rad/s), Pm = 91.3 deg (at 0.317 rad/s)
From: u1 To: y1
Magnitude (dB)
Phase (deg)
Frequency (rad/s)

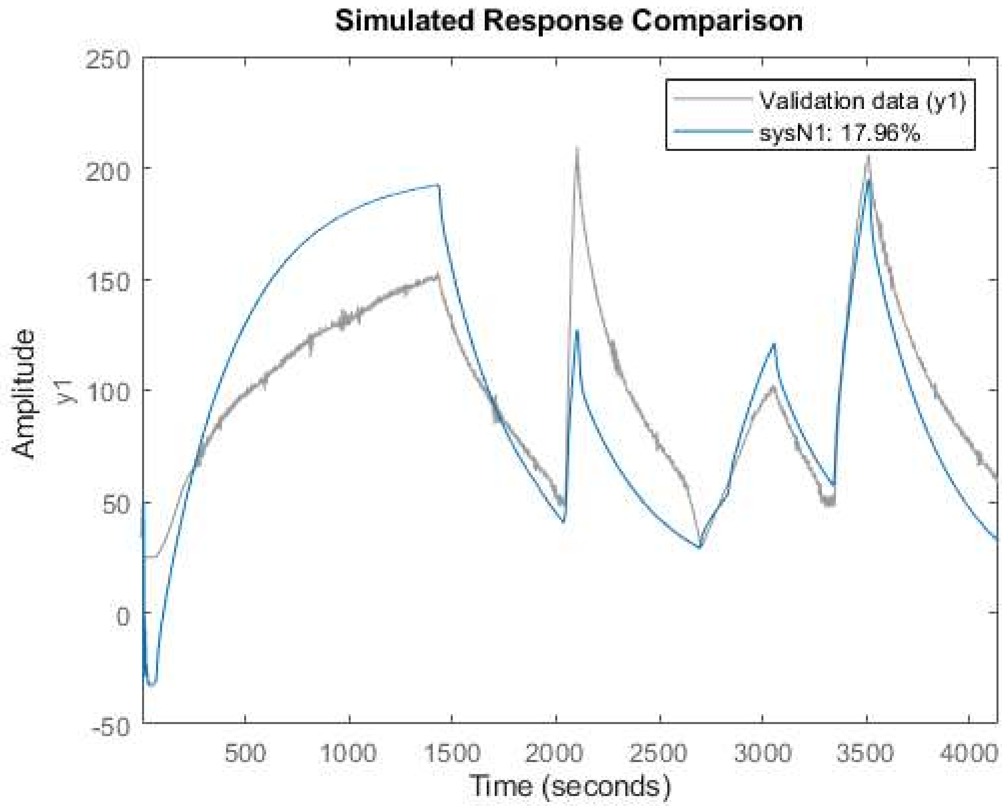

Pole-Zero Map
Imaginary Axis (seconds^{-1})
Real Axis (seconds^{-1})
$\times 10^{-3}$

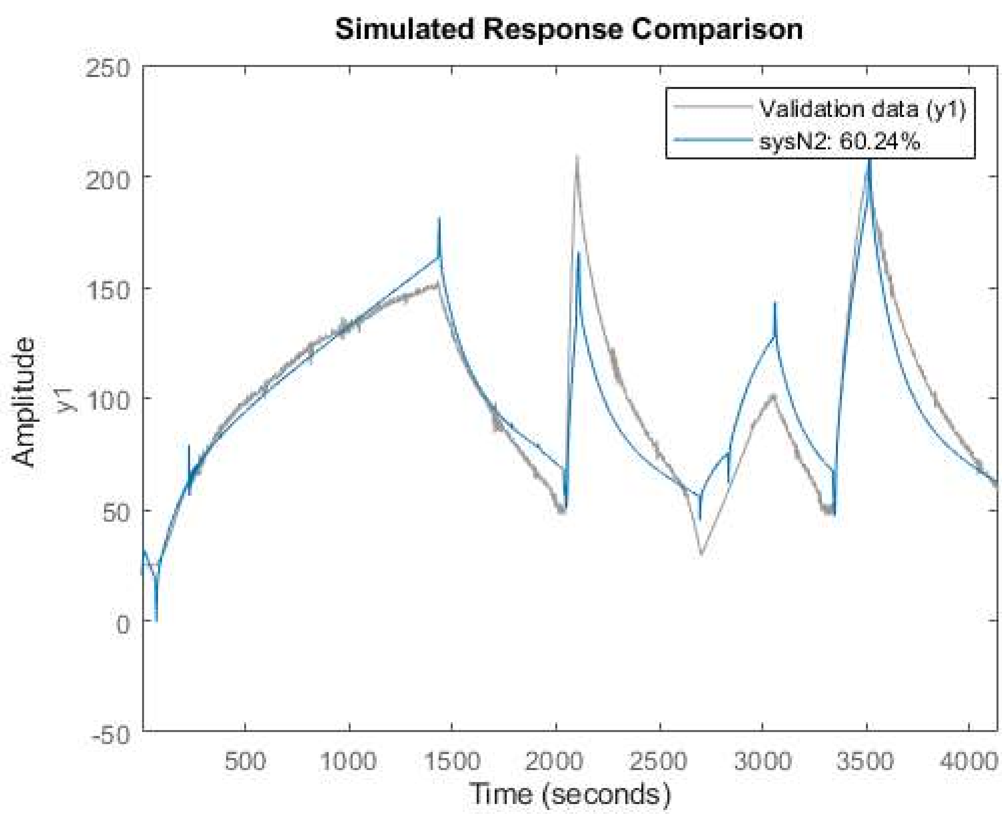

Nyquist Diagram
From: u1 To: y1
×10^4
0 dB
Imaginary Axis
Real Axis
×10^4
8
6
4
2
0
-2
-4
-6
-8
-4
-3
-2
-1
0
1
2
3


FFT of y
|fft(y)|
Hz
200
180
160
140
120
100
80
60
40
20
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7


FFT of u
|fft(u)|
Hz
0
5
10
15
20
25
30
35
40
45
50
0.1
0.2
0.3
0.4
0.5
0.6
0.7


Simulated Response Comparison
Validation data (y1)
sysN0: -2.691%
Amplitude
y1
Time (seconds)
0
50
100
150
200
250
500
1000
1500
2000
2500
3000
3500
4000


Simulated Response Comparison
Validation data (y1)
sysN1: 17.96%
Amplitude
y1
Time (seconds)
250
200
150
100
50
0
-50
500
1000
1500
2000
2500
3000
3500
4000


Simulated Response Comparison
Validation data (y1)
sysN2: 60.24%
Amplitude
y1
Time (seconds)
250
200
150
100
50
0
-50
500
1000
1500
2000
2500
3000
3500
4000

Save best model

```
save('idpoly','sysN2');

% State Space representation of best TF models

model3_ss=ss(model3)
model4_ss=ss(model4)

control=rank(ctrb(model4_ss.A, model4_ss.B))
observe=rank(obsv(model4_ss.A, model4_ss.C))

test_ss=ss(model2)


% LQR/LQG work, assuming A,B,C,D creates observability and controllability
%Q=1*[1 0 0 0; 0 1 0 0; 0 0 1 0; 0 0 0 1]; % state x state
%R=[5]; % 1 x 1 : (u x u)
%[K,S,P] = lqr(model4_ss,Q,R)
[K,S,P] = lqr(test_ss,0.1,0.1)

% A=model4_ss.A;
% B=model4_ss.B;
% C=model4_ss.C;
% D=model4_ss.D;

A=test_ss.A;
B=test_ss.B;
C=test_ss.C;
D=test_ss.D;

%result=inv([A,B;C,D])*[0;0;0;0;1]
result=inv([A,B;C,D])*[0;1]

Nx=result(1)
Nu=result(2)
Nbar=Nu+K*Nx

% Nx=result(1:4)
% Nu=result(5)
% Nbar=Nu+K*Nx
sysLQR=ss((A-B*K),B,C,D,Ts)
```

```
model3_ss =

  A = 
                x1          x2          x3          x4
   x1  -0.009076    -0.01943    -0.01118   -0.003493
   x2    0.03125           0           0           0
   x3          0     0.01562           0           0
   x4          0           0    0.001953           0
```

```
  B =
       u1
   x1   1
   x2   0
   x3   0
   x4   0

  C =
            x1       x2       x3       x4
   y1   0.3171  0.01669   0.3888    0.413

  D =
       u1
   y1   0

Continuous-time state-space model.

model4_ss =

  A =
            x1       x2       x3       x4
   x1    3.988   -2.982   0.9915  -0.4945
   x2        2        0        0        0
   x3        0        2        0        0
   x4        0        0      0.5        0

  B =
       u1
   x1   1
   x2   0
   x3   0
   x4   0

  C =
            x1       x2       x3       x4
   y1   0.3876  -0.5808   0.2902  -0.1934

  D =
       u1
   y1   0

Sample time: 1.2279 seconds
Discrete-time state-space model.

control =

     4


observe =

     4
```

```
test_ss =

  A =
           x1
   x1  0.9963

  B =
        u1
   x1  0.5

  C =
           x1
   y1  0.7476

  D =
       u1
   y1   0

Sample time: 1.2279 seconds
Discrete-time state-space model.

K =

    0.7745


S =

    0.2543


P =

    0.6091


result =

    1.3376
   -2.6653


Nx =

    1.3376


Nu =

   -2.6653


Nbar =
```

```
  -1.6293


sysLQR =

  A =
             x1
   x1  0.6091

  B =
          u1
   x1  0.5

  C =
             x1
   y1  0.7476

  D =
         u1
   y1   0

Sample time: 1.2279 seconds
Discrete-time state-space model.
```

Published with MATLAB® R2023b

Sintering

Sintering is the process of compacting and forming a solid mass of material by heat or pressure without melting it to the point of liquefaction.

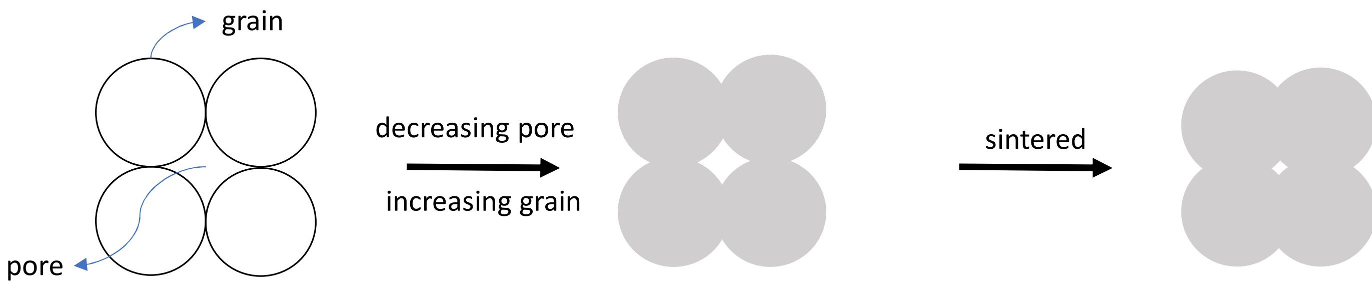


Figure 1: Pore and grain transition during sintering

Purpose

- ❑ Improved strength - The compacted material acts more like a continuous solid
- ❑ Densification - reduces porosity in the powder compact, resulting in higher density and fewer structural defects.
- ❑ New composites - can be created that combine beneficial aspects of different materials.

Traditional Sintering

Attributes

- ❑ High temperature (50%-70% of the melting temperature)
- ❑ Longer duration in the furnace
- ❑ Temperature propagates from the outer surface to the inside

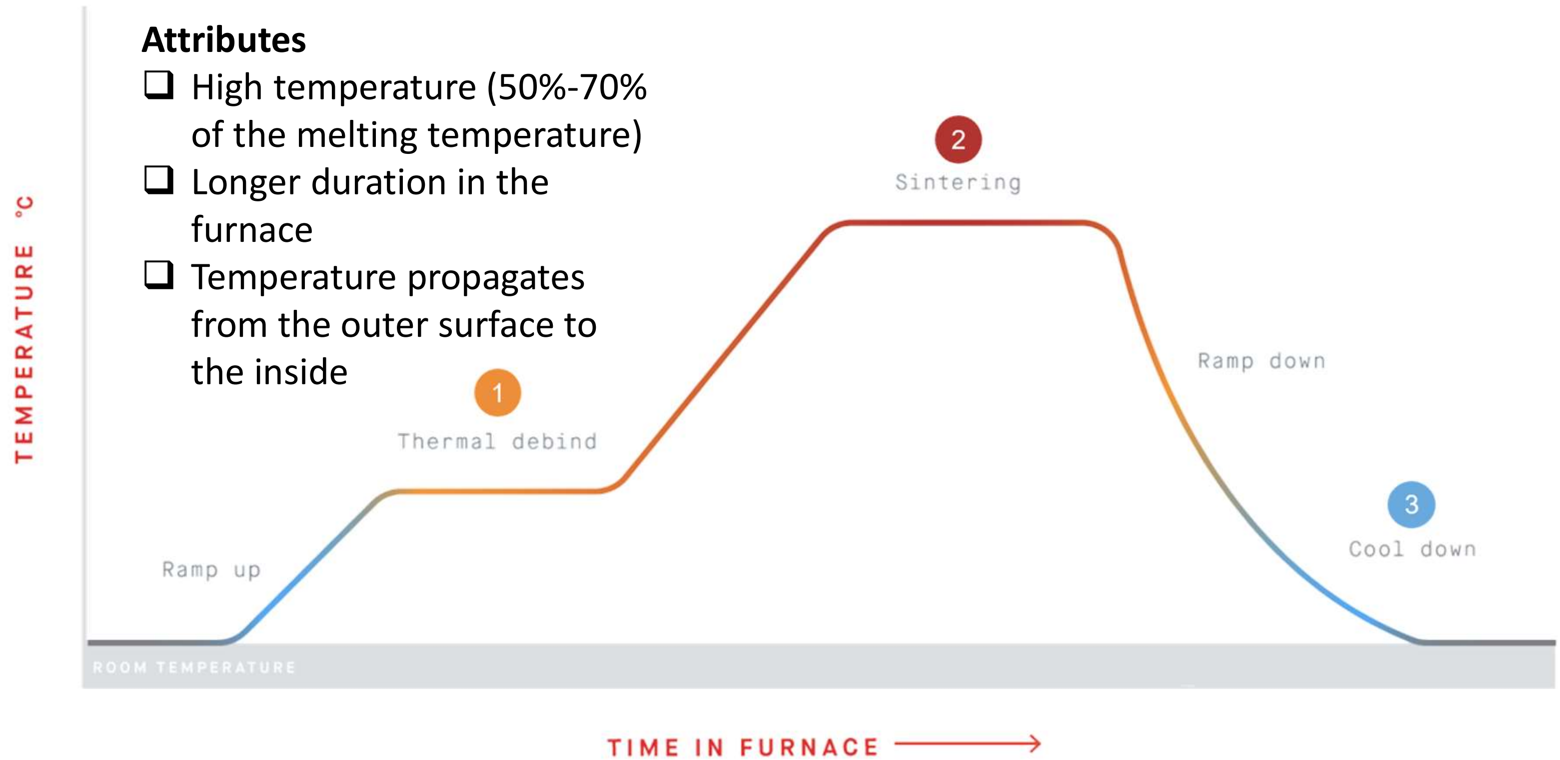


Figure 2: Temperature profile during sintering

Current State of Sintering

Attributes

- ❑ Lower sintering temperature
- ❑ Uses electric pulses, promoting spark plasma sintering
- ❑ Energy deposited at the molecular level
- ❑ Small duration completing the sintering cycle
- ❑ Saves energy

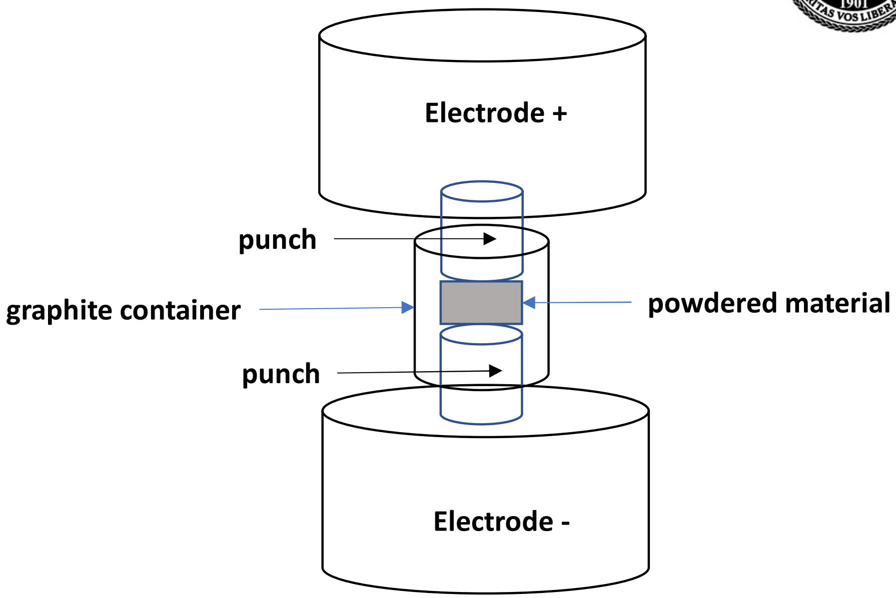


Figure 3: An Electric Field Assisted Sintering setup

Idaho National Laboratory

Manufacturing Methods Cost/Energy Comparison

	Energy use	Cost for 5,000 parts
Hot pressing	1.83 kW/g	$5.7M
Spark Plasma Sintering	0.1 kW/g	$520K
Savings via Spark Plasma Sintering	1.82 kW/g	$5.2M