CHALMERS
UNIVERSITY OF TECHNOLOGY

Troubleshooting and improving an existing robot used in verification test

Degree project in Mechatronics

Emil Skaug

DEPARTMENT OF ELECTRICAL ENGINEERING

CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2026
www.chalmers.se

Degree project 2026

Troubleshooting and improving an existing robot used in verification test

Emil Fredrik Skaug

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Electrical Engineering
Chalmers University of Technology
Gothenburg, Sweden 2026

Troubleshooting and improving an existing robot used in verification test
EMIL SKAUG

Supervisor: André Bezerra de Freitas Diniz, Electrical Engineering
Examiner: Erik Ström, Department of Electrical Engineering

Degree project 2026
Department of Electrical Engineering
Division of Communications, Antennas and Optical Networks
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 70 996 95 45

Cover: Picture of measuring robot.

Typeset in LaTeX
Gothenburg, Sweden 2026

Troubleshooting and improving an existing robot used in verification test
EMIL SKAUG
Department of Electrical Engineering
Chalmers University of Technology

Abstract

Reliable verification of radar measurements is essential for ensuring safe operation in industrial environments. At Emerson Rosemount Tank Radar AB, a wireless measuring robot is used inside pipes to perform factory acceptance tests of a radar transmitter. However, recurring stoppages and poor communication with the robot have delayed the verification process. This project investigates the root causes of these issues and the connection to the communication system. To investigate and evaluate potential improvements, a test script was developed to log registers and communication response times of the robot.

The results showed that relocating the Bluetooth transmitter improved communication stability, and several hypotheses regarding the unknown stoppages could be ruled out. In addition, the existing battery solution for the wireless robot is temporary, therefore three alternative concepts were developed and evaluated against technical and usability requirements. To ensure the robot's long-term operation, a spare parts analysis was conducted, revealing risks related to discontinued electronic components and wear parts.

The outcome of the project is a more stable Bluetooth connection and improved long-term maintainability of the measuring robot, helping to ensure its continued use in factory acceptance testing.

Acknowledgements

This degree project has been conducted as part of the Mechatronics Engineering programme at Chalmers University of Technology. The project was carried out at Emerson Rosemount Tank Radar AB, where the involved teams provided valuable technical expertise and support throughout the work. The collaboration with the company has offered great insight into industrial practices and to real engineering workflows.

I would like to express my sincere gratitude to my manager, Christer Möller at Emerson for giving me the opportunity to carry out this project, as well as my supervisor at Emerson, Hans Corneliusson and the team, for providing help throughout the project.

I would like to thank Chalmers University of Technology for preparing me for this project with three valuable years, and in particular I would like to thank my academic supervisor, André Bezerra de Freitas Diniz, for providing guidance and feedback throughout the work.

Emil Fredrik Skaug, Gothenburg, June 2026

List of Acronyms

Below is the list of acronyms that have been used throughout this thesis listed in alphabetical order:

BMS	Battery Monitoring System
BOM	Bill Of Material
CE	European Conformity
FAT	Factory Acceptance Test
GATT	Generic ATTribute
IC	Integrated Circuits
IP	Internet Protocol
LED	Light Emitting Diode
LNG	Liquefied Natural Gas
MR	Measuring Range
MTO	Made To Order
OPC	Open Platform Communication
PCM	Protection Circuit Module
RSSI	Received Signal Strength Indicator
RTT	Round Trip Time
RX	Receive
SCU	Supply and Communication Unit
SMA	Sub Miniature version A
SoC	State of Charge
SPP	Serial Port Protocol
TCP	Transmission Control Protocol
TGU	Tank Gauge Unit
TX	Transmit
UART	Universal Asynchronous Receiver/Transmitter
UI	User Interface
USB	Universal Serial Bus

Notation

Below is the nomenclature of Variables and Registers that have been used throughout this thesis.

Variables

t_1	Timestamp before TX
t_2	Timestamp between TX and RX
t_3	Timestamp after RX

Registers

ERR_STAT	Error and status register from MAC95 motor
U_SUPPLY	Voltage of battery measured by the robot

Contents

List of Figures

List of Tables

1 Introduction

This report presents a troubleshooting analysis and improvements to a robot used in a Factory Acceptance Test (FAT) at Emerson Rosemount Tank Radar AB's facility.

1.1 Background

Emerson Rosemount Tank Radar AB develops advanced radar systems for volume and level measurement in storage tanks for both land-based and marine-based environments [1]. In many installations, a vertical pipe is used for the radar signal. The radar transmitter is mounted on top of the tank, to measure the distance to the product surface inside the tank [2].

For certain marine applications, the radar transmitter Tank Gauge Unit (TGU) and the pipe can be factory-calibrated as a complete unit, with the verification witnessed by a third party prior to delivery and installation. This is not a general requirement when purchasing the TGU, as each unit is already tested during production, but it may be requested by the customer. The pipes are mounted horizontally in the FAT environment, and a measuring robot is used inside the pipe to simulate a reflective surface at different distances. This verification is done at Emerson's facility in Mölnlycke.

The FAT room consists of three measuring ranges. MR 1 and MR 2 are separate, but share identical setup, whereas MR 3 has a unique setup. Brief information about the measuring ranges are described in Table 1.1.

Content information	**MR 1**	**MR 2**	**MR 3**
Pipe lengths	30–35 m	30–35 m	40–50 m
Pipes quantity	Total of 4	Total of 4	Total of 2
Robot No	Robot 1	Robot 2	Robot 3
Transmitter No	Transmitter 1	Transmitter 2	Transmitter 3
Transmitter placement	At computer	At computer	At laser
Pipes	Customer	Customer	Emerson's
Usage	Used during FAT	Used during FAT	Used for production
Usage freq	Occasionally	Occasionally	Frequently

Table 1.1: Measuring ranges information.

When no customer pipes are present in the facility, that space is left open, while waiting for incoming pipes. On the contrary, MR 3 is utilized where customer pipes are not required and is more frequently used.

Fig. 1.1 illustrates the current setup of the three measuring ranges containing information about placement and general overview of the modules for each MR.

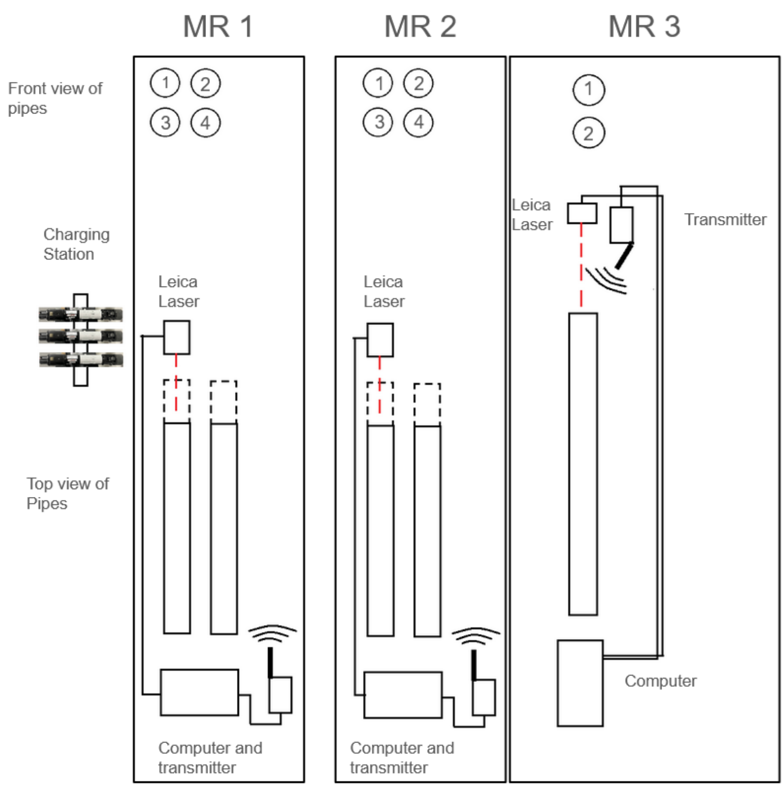


Figure 1.1: System overview of all the measuring ranges.

1.2 Objective

The objective of this thesis is to improve the existing measurement robot used in Emerson's FAT environment by troubleshooting the problems connected to communication and testing potential improvements. The work aims to analyze the current issues and hopefully rule out potential hypotheses connected to stoppages. The work also aims to provide alternative battery concepts and to assess the spare part availability. The overall purpose is to enhance the robot's operational reliability and reduce the risk of future stoppages.

1.3 Constraints

The thesis project will not include a totally new measurement robot, instead the work is limited to improving the existing concept. Emerson's internal software connected with the robot will not be troubleshot, but can be addressed. The thesis does not cover redesigning the full measuring range or implementing large scale hardware replacements beyond what is required to evaluate proposed improvements.

1.4 Research Questions

Based on the scope and objective of the thesis, the following questions have been formulated:

- How can the performance reliability of the Bluetooth communication be improved?
- Does the market offer any commercially available simple battery pack solutions?
- What battery concepts can be developed based on the requirements?
- What components inside the robot are critical?
- How can the long-term maintainability of the robot be ensured through spare part availability, and does the market offer similar standardized components?

1.5 Societal, Ethical and Ecological Aspects

As stated previously, Emerson manufactures multiple different radar transmitters, both guided waves, but also non-contacting radar. Some of the radar transmitters manufactured can later be installed on larger storage tanks filled with oil. This is directly connected to environmental and social safety. By using accurate level measurements, this reduces the risk of overfilling and potential hazards that could lead to accidents. An example of a tank overfill accident that is relevant to all aspects of this section is the accident in Puerto Rico in 2009, where inadequate level monitoring caused a large oil over spill [3]. This accident was not related to Emerson; however, such events illustrate why reliable level measuring systems and overfill prevention are essential, and why Emerson works to prevent overfill. With a robust level measuring system, personnel can work safely in a storage environment, supported by accurate and reliable measurements.

Since this thesis will improve a robot used in a FAT test, the customer will be more satisfied with the verification process and the performance of the level measurement. However, from an engineering perspective, it is also important that the improvements made throughout the thesis do not introduce new risks, such as fire hazards, electrical shocks, mechanical hazards, or such.

Furthermore, extending the lifetime of the robot instead of building a new unit aligns with today's standard of sustainability, that is prioritizing repair over replacement. However, since one of the improvements provides purchasing new batteries, it is important to consider the work culture associated with mineral extraction. Most of the minerals extracted originate from Africa, South America and Asia. The extraction contributes to water pollution, poor working conditions, and inadequate safety regulations [4].

2

Theory

This section will provide information about each module in the measuring range setup at a system and subsystem level. A high level system architecture of the setup is illustrated in Fig. 2.1.

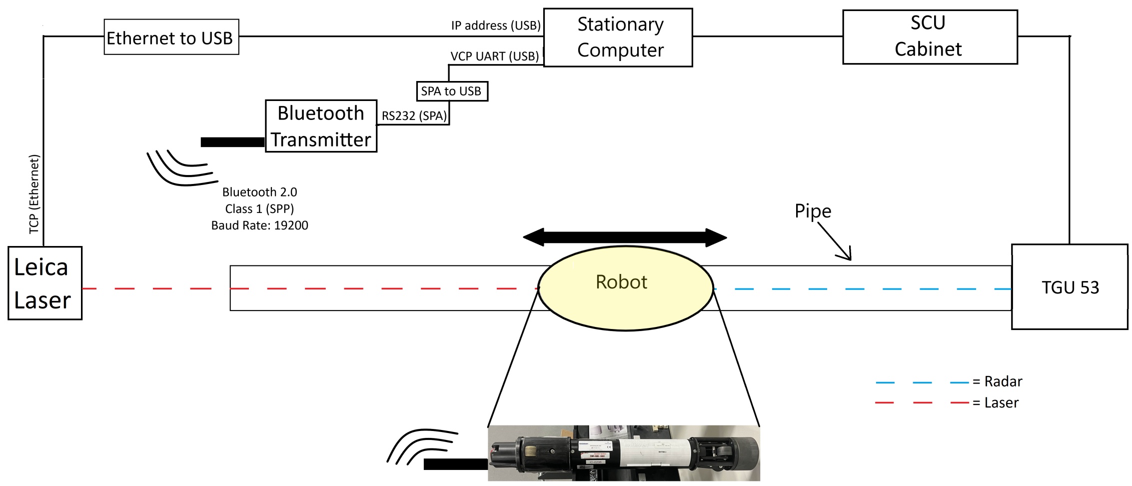


Figure 2.1: High level system architecture of the setup.

2.1 Leica Laser Tracker

The Leica laser tracker is a portable laser tracker used for high precision measurements in industries. The distance to a target is determined based on the transmitted and reflected laser off of a laser prism. The Leica laser communicates through Transmission Control Protocol (TCP)/Internet Protocol (IP)-address which is an internet protocol [5]. In the FAT setup, the connection between the computer and the Leica laser is established using an Ethernet-to-Universal Serial Bus (USB) adapter making it operate at a local internal IP address.

2.2 Stationary Computer

The stationary computer is the brain and command center of the system. To control the robot, the following three software are used:

- **MacTalk** - MacTalk is a paid software from JVL used to control MAC motors with an advanced User Interface (UI) [6].
- **MacRegIO** - MacregIO is a free software from JVL used to control the registers in the motor. It is mainly to control the register, with minimal UI [7].
- **dRange** - dRange is an Emerson developed program used during TGU testing.

2.3 Bluetooth Transmitter

The Bluetooth transmitter utilized in the FAT is a connectBlue. It utilizes Bluetooth which is a wireless communication standard used to exchange data between two devices. Common profiles are Serial Port Protocol (SPP) or Generic ATTribute (GATT). The connectBlue uses SPP with an older, Bluetooth 2.0 and class 1 for longer distances, instead of class 2 for shorter distances. The Light Emitting Diode (LED) on the connectBlue indicates blue for established and stable connection, purple for lost connection and green for actively looking for connection [8].

2.4 External System

One of the marine products system Emerson offers consists of:

- **TGU 53** - the TGU 53 is a radar gauge precision instrument that measures level on Liquefied Natural Gas (LNG) tanks by propagating a radar wave through a pipe. The deviation of the distance is ± 3mm [2]. In the FAT facility it is connected to the Supply and Communication Unit (SCU).
- **SCU Cabinet** - the SCU cabinet collects data from all the connected transmitters onboard an LNG tank, using server/client based communication interfaces, Open Platform Communication (OPC) [9]. In the FAT facility the SCU cabinet collects measured distance from TGU and provides the stationary computer with the distance over OPC.

2.5 Robot

The robot is a custom designed unit integrated with multiple sub components. It moves in and out of the pipe controlled through Bluetooth. A high level system architecture of the robot is illustrated in 2.2

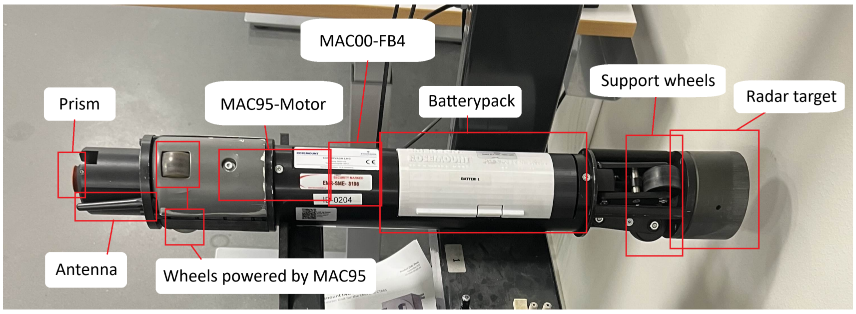


Figure 2.2: High level system architecture of the robot.

2.5.1 MAC95

The MAC95 is a 92 W integrated servomotor produced by JVL, designed for electronic motion control [10], it is used in the robot to drive the wheels forward. It consists of:

- **Servomotor** - a servomotor is a controlled electronic motor and is frequently utilized in industries to ensure controlled movement.
- **Encoder** - an encoder is a sensor that is integrated in a servomotor and detects angular displacement, also referred to as ticks used to achieve accurate motion control.
- **Communication Interface** - the servomotor can be controlled through Universal Asynchronous Receiver/Transmitter (UART) which is an industry standard serial interface.
- **Protocol** - the MAC95 motor includes register and can be addressed by the MacTalk protocol.

2.5.2 MAC00-FB4

The MAC95 is connected to a Bluetooth module MAC00-FB4 for controlling the robot wireless [11], the connection is illustrated in Fig. 2.3.

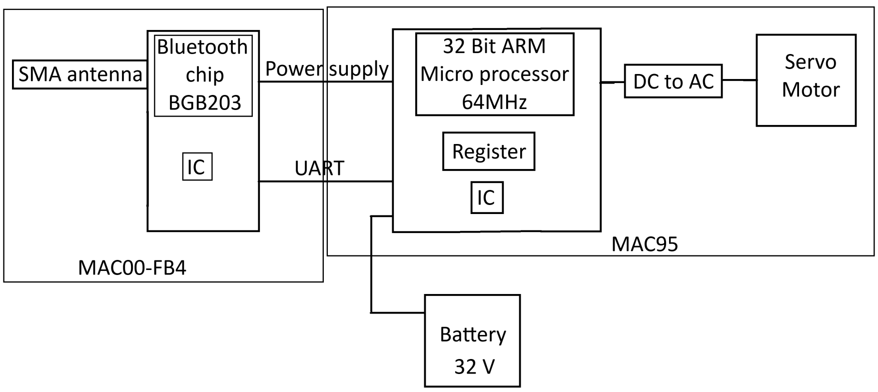


Figure 2.3: Electrical subsystem architecture of the robot.

The module consists of:

- **Bluetooth chip** - a Bluetooth chip is an Integrated Circuits (IC) that provides Bluetooth functionality. The chip integrated in the module is the BGB203.
- **Antenna** - an SMA-connected antenna used to receive signals from the transmitter.

2.5.3 Battery

The robot is powered by a battery pack consisting of multiple battery cells. A battery cell is a stored energy package. It stores electrical energy as chemical energy and it consists of an anode, a cathode and electrolyte. Battery cells usually come in different standard sizes example: (18650, 21700, 26650), and depending on the chemistry different aspects are to be considered [12]. A battery cell can consist of:

- **Lithium-ion battery** - a lithium-ion battery cell is a high density rechargeable battery cell. It usually consists of a graphite anode, a lithium metal oxide cathode and electrolyte [12].
- **Protection Circuit Module (PCM)** - for minimal protection during charging a PCM can be used. It provides basic safety functions more suitable for battery cells [13].
- **Battery Monitoring System (BMS)** – a BMS is an advanced protection and control system for battery packs. It includes cell balancing circuitry, supervised charging, and continuous monitoring of pack parameters, such as the State of Charge (SoC) [13]. Each cell is connected to the BMS through individual sense leads for precise measurement and management of cell voltages.

2.5.4 Other Components of the Robot

The robot consists of multiple other subcomponents including drive package which is illustrated in Fig. 2.4. This drive package is the mechanical distribution that turns the MAC95 motion to move the wheels, driving the robot.

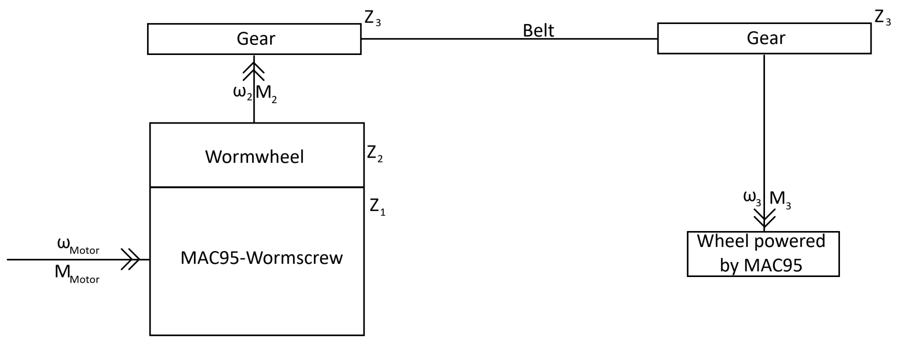


Figure 2.4: Mechanical subsystem architecture of the robot.

Some other large physical components of the robot are:

- **Radar target** - target for TGU transmitter to measure on inside the pipe.
- **Support wheels** - wheels that stabilize the robot, ensuring that it does not tilt.
- **Prism** - target prism for laser to measure where the robot is located inside the pipe.
- **Drive package** - the drive package consists of screws and belts.

3
Methods

This section includes the steps taken to develop a test script for communication testing and procedures used for data collection and evaluation for components inside the robot.

3.1 Troubleshooting Communication Performance

Since communication with the robot was unreliable and one of the primary issues of stoppages, the investigation focused on troubleshooting the communication by isolating and evaluating the communication subsystem. The objective was to provoke stoppages and identify communication performance. This included focusing on the link between the robot and stationary computer only, excluding the other modules.

3.1.1 Test Design

The communication performance was evaluated using a set of standard metrics. The Bluetooth chip BGB203 integrated in the MAC00-FB4 module supports Received Signal Strength Indicator (RSSI). However, the MAC00-FB4 module does not provide access to this feature. Therefore, the test was limited to the metrics presented in Table 3.1.

Information	**Description**
Latency	RTT
Intensity	Throughput
Stability over time	Performance consistency

Table 3.1: Communication metrics possible.

To evaluate the communication using the metrics described in Table 3.1, direct communication with the robot was required. This communication served two purposes, measuring the RTT and monitoring internal registers in the motor during operation. Since the existing programs did not support the desired features of logging register and RTT, a Python test script was developed. This test script communicated with the robot using MacTalk protocol. The Table 3.2 illustrates an example of a message transmitted and message received from reading register 5.

Type	Byte-sequence	Total Bytes
Transmitted	$\underbrace{50h\ 50h\ 50h}_{\text{Read}} - \underbrace{08h\ F7h}_{\text{Address}} - \underbrace{05h\ \text{F6h}}_{\text{RegNum}} - \underbrace{AAh\ \text{AAh}}_{\text{End}}$	9
Received	$\underbrace{52h\ 52h\ 52h}_{\text{write}} - \underbrace{08h\ F7h}_{\text{address}} - \underbrace{05h\ \text{FAh}}_{\text{regnum}} - \underbrace{02h\ \text{FDh}}_{\text{len}} - \underbrace{\text{58h A7h 02h FDh 00h FFh}}_{\text{data}} - \underbrace{\text{AAh AAh}}_{\text{end}}$	18

Table 3.2: Example of reading from register 5 with MacTalk protocol.

The MAC00-FB4 module operates at a baud rate of 19200 bits/s. With 1 byte being 8 bits, turning the total number of bytes from the transmitted message and the received message, described in Table 3.2 into bits, the minimum theoretical RTT for a read operation was calculated in Equation 3.1.

$$\frac{9 \text{ bytes} \cdot 8}{19200 \text{ bits/s}} + \frac{18 \text{ bytes} \cdot 8}{19200 \text{ bits/s}} = 10.8 \text{ ms} \tag{3.1}$$

The value calculated in Equation 3.1 was used as a reference when comparing to the measured RTT.

3.1.2 Development of the Logging Script

To monitor and communicate with the robot a python logging script was developed. This script transmitted a read message in hex and received the response in hex using python.

To transmit a byte-sequence using python, required no logic. On the contrary the received messages required logic due to fragmented incoming response. The fragmented Receive (RX) could be received as shown in Listing 3.1.

```
Transmit: b'PPP\xff\x00\x05\xfa\xaa\xaa'
b'R'
b'RR\xff'
b'\x00\x02\xfd'
b''
b'\x02\xfd\x01'
b'\xfe\x00\xff'
b''
b'\xaa'
b'\xaaP'
b''
b'PP\xff\x00'
b'\x02'
b'\xfd\xaa'
b'\xaa'
```

Listing 3.1: Python output of Transmit (TX) and RX from reading register 5

The logic to handle the incoming fragmented response from the RX was based on the MacTalk protocol. The RX function used the MacTalk protocol that defines a start byte sequence of 525252 and an end byte sequence of *AAAA*. These byte sequences were used to determine a complete RX message. A timeout of 3 seconds was also implemented to ensure that if the message was not received within this time, the request timed out and retried at most 3 times.

After the TX and RX functions were written, a set of registers were selected to monitor continuously, the selected registers are shown in Table 3.3 and were evaluated.

Reg No	**Reg Name**	**Data Type**	**Description**
20	FLWERR	Longint	Follow Error
24	FUNCERR	Longint	Function Error
32	ACC_EMERG	Word	Emergency Acceleration
35	ERR_STAT	Word	Error and Status
151	U_SUPPLY	Word	Feed Voltage

Table 3.3: Register Overview.

The `ERR_STAT` is an error status register, it will provide a value based on an active bit. The bit is connected to an error. Since JVL provides common errors connected to certain error bits, monitoring this register will show if the MAC95 provides any errors while running, page 416 in [10], to understand if the problem is the MAC95 or not.

The `U_SUPPLY` register provides a value of for example 602 or 598. The value is interpreted as a voltage using Equation 3.2 in dRange, making the value readable. The register was monitored to see if any sudden voltage dropped occurred during operation of the robot.

$$V = 0.05714 \cdot \texttt{U\_SUPPLY\_Value} - 1.97 \tag{3.2}$$

A log function was implemented and logged into a text file, the function logged:

- The transmitted TX message and the received RX message.
- Timestamps (t_1, t_2, t_3) to differentiate between TX time and RX time.
- The RTT that was calculated by $t_3 - t_1$.
- The amount of retries if a message was not completed within 3 seconds.

The structure and logic of the logging script is described Fig. 3.1.

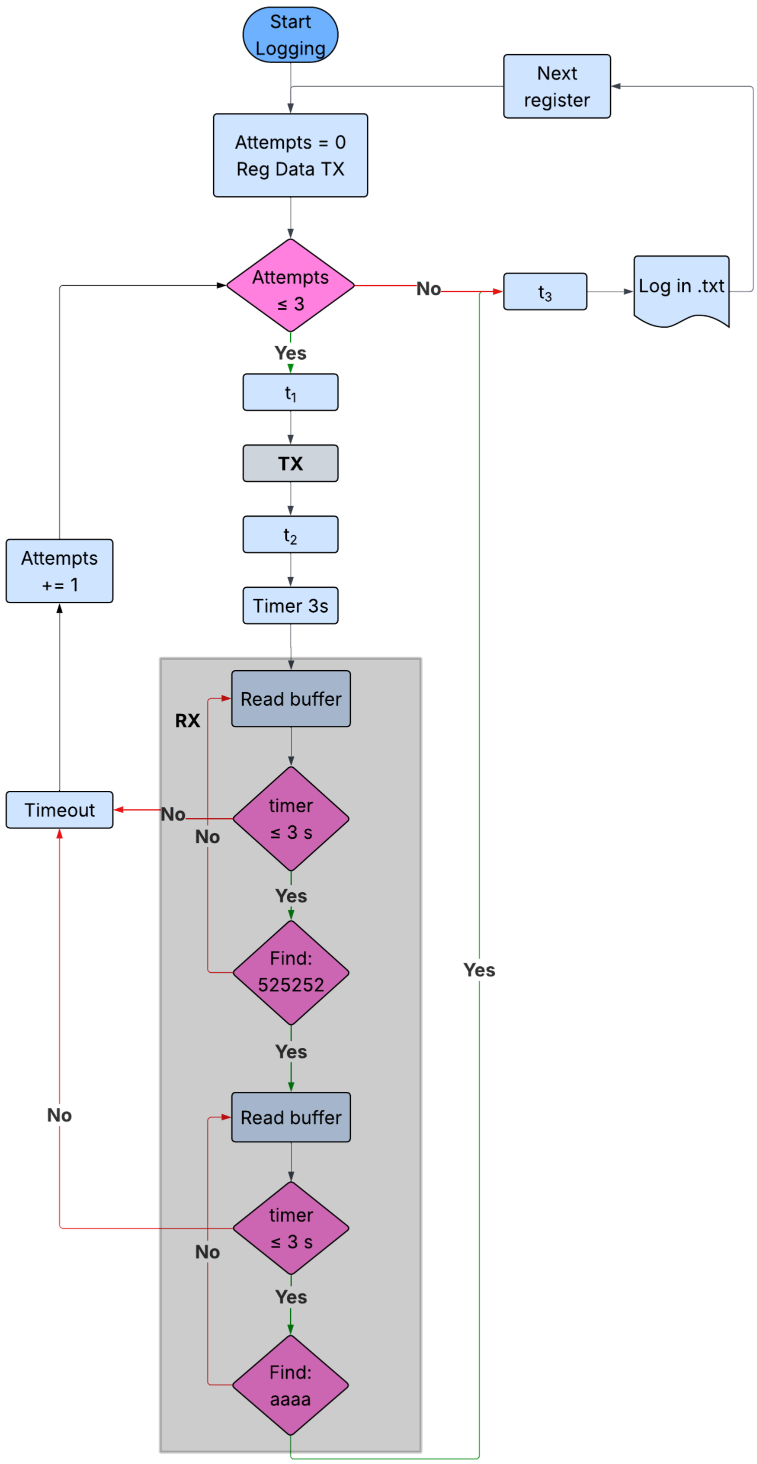


Figure 3.1: Code logic of the logging script.

3.1.3 Development of the Parsing Script

The logging script logged transmitted and received messages data in MacTalk format which is in hexadecimal format. To convert this data into a readable format, a parsing script was developed. The parsing code extracted the RX data and decoded the data depending on the registers data type. Different decoding methods were used depending on the datatype of the register, for example integer, long integer, or fixed8 format.

The process to decode the received message from `REG_MODE` (register 2), stored in integer format, into a readable format using the MacTalk protocol is illustrated in Table 3.4.

Type	Structure
Sent	$\underbrace{50h\ 50h\ 50h}_{\text{Read}} - \underbrace{08h\ F7h}_{\text{Address}} - \underbrace{05h\ F6h}_{\text{RegNum}} - \underbrace{AAh\ AAh}_{\text{End}}$
Received	$\underbrace{52\ 52\ 52}_{\text{write}} - \underbrace{00\ FF}_{\text{address}} - \underbrace{02\ FD}_{\text{regnum}} - \underbrace{04\ FB}_{\text{len}}$ $\underbrace{01\ FE\ 00\ FF\ 60\ 9F\ AE\ 51}_{\text{data}} - \underbrace{AA\ AA}_{\text{end}}$

Table 3.4: Example of reading from register 2.

The steps taken are illustrated in Table 3.5.

Function	Message
Start	52525200ff02fd04fb01fe00ff609fae51aaaa
Divide hex	52, 52, 52, 00, ff, 02, fd, 04, fb, 01, fe, 00, ff, 60, 9f, ae, 51, aa, aa,
Hex to dec	82, 82, 82, 0, 255, 2, 253, 4, 251, 1, 254, 0, 255, 96, 159, 174, 81, 170, 170
Extract data	1, 254, 0, 255, 96, 159, 174, 81
Remove mirrored	1, 0, 96, 174
Integer decode (little endian)	1, 0 $\rightarrow 1 + (0 \cdot 256) = \mathbf{1}$

Table 3.5: Process for READ_MODE_REG (integer decoding).

The robustness of the code was tested by comparing the decoded values using the parsing code and reading the values from MacRegIO. In addition a test was conducted to slowly unscrew the Sub Miniature version A (SMA) antenna on the robot as a simulation of poor connection, whilst keeping the received message intact.

The parsed values in addition to the logged values were transferred to Excel for better data processing.

The parsing process is illustrated in Fig. 3.2.

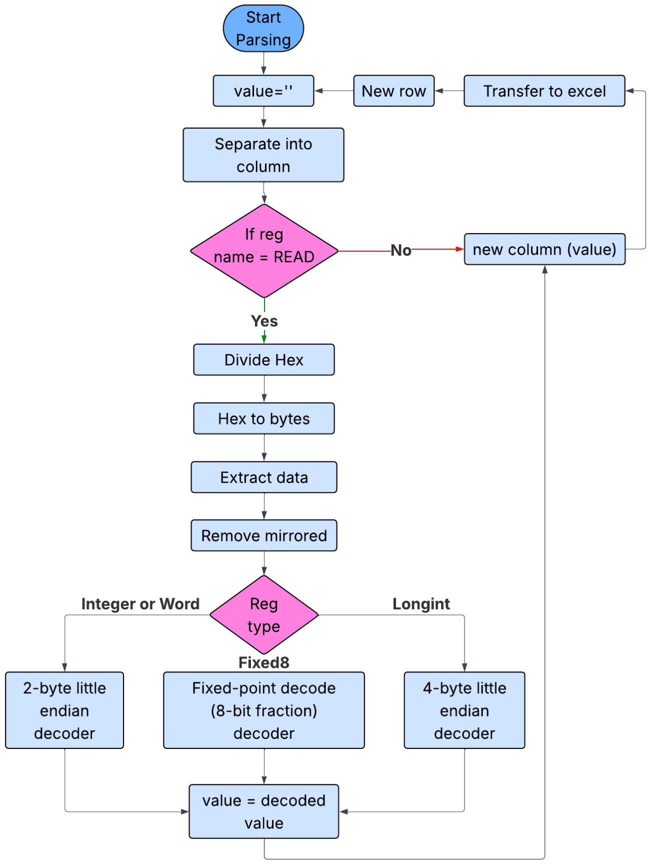


Figure 3.2: Code logic of the parsing script.

3.1.4 Extension of the Logging System: Sweep Test Implementation

After validating the logging script and the parsing script, additional code was implemented to the logging script, to write to the robot's register and was named sweep script. This was done to operate the robot during sweep testing whilst logging. To operate the robot, three registers were written to and are presented in the Table 3.6.

Reg	Name	Value(unitless)	Description
2	MODE_REG	1	Velocity mode
5	V_SOLL	1000	Moderate speed
6	A_SOLL	50	Moderate acceleration

Table 3.6: Register values used during sweep tests.

The description of the sweep test consisted of swapping between movement and stationary logging phases. The robot was commanded through writing to registers to move approximately 2 meters along the pipe, which was approximately 17 seconds at moderate speed, while logging, then stopped and remained stationary for 200 seconds while logging. This sequence was repeated a total of 13 times with an approximate distance of 26 meters. The robot was then commanded to reverse and repeat the process. The robot's position was monitored, but not logged using the Leica laser.

The total duration of the sweep script was approximately 2 hours. The logic for the sweep script is illustrated in Fig. 3.3.

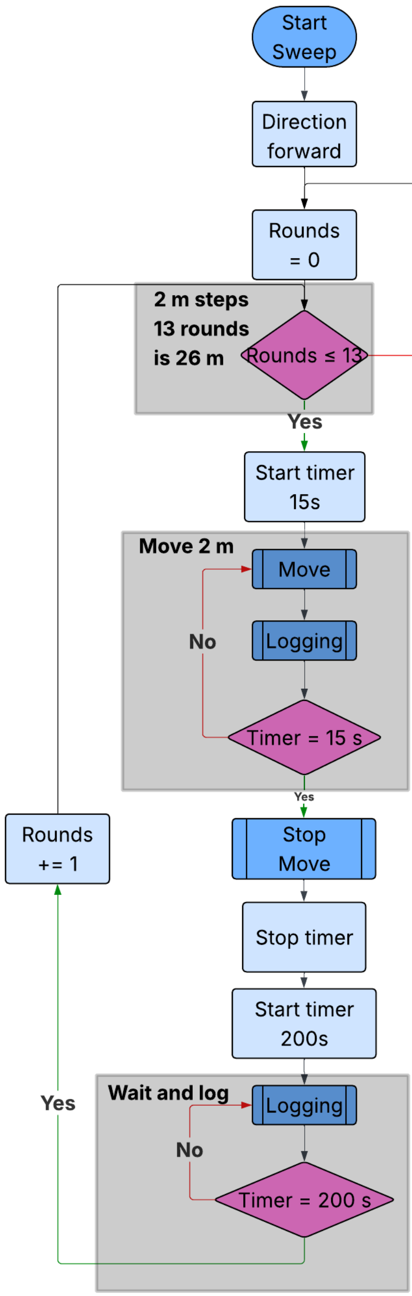


Figure 3.3: Code logic of the sweep function.

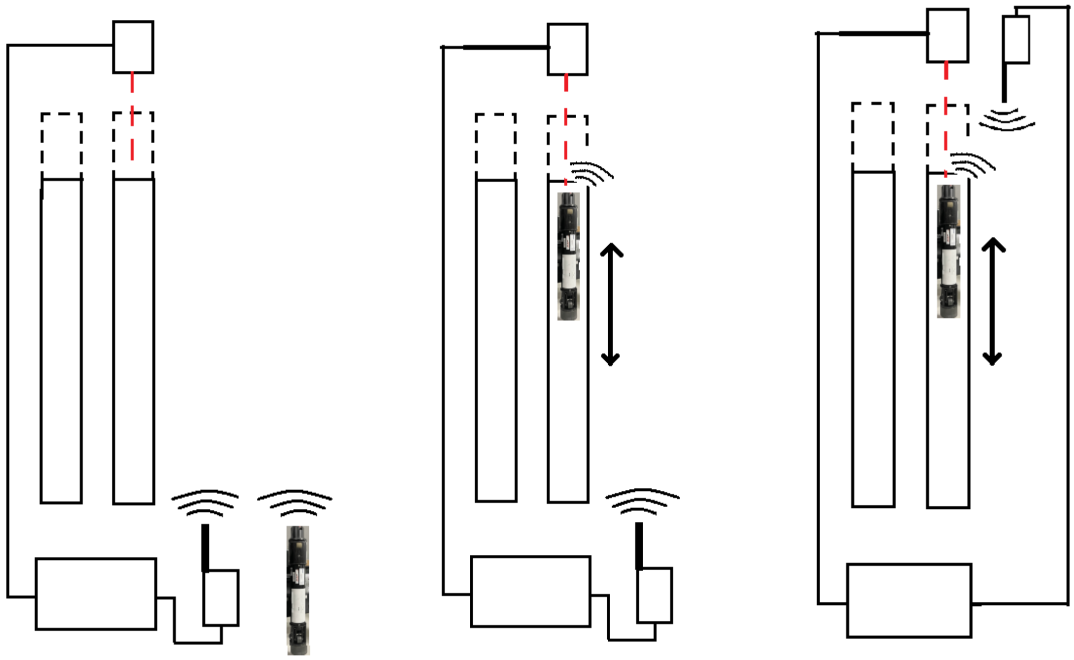
3.1.5 Defining Three Tests Configurations

The sweep test was performed on all measuring ranges to evaluate the RTT and registers values during operation. The sweep test was performed both inside the pipe and outside the pipe, besides the transmitter, to illustrate the difference in RTT and values in registers. The three tests performed are described in Table 3.7.

Test	Description
1	The robot was placed outside the pipe standing up beside the transmitter whilst sweep test was running, for best possible connection.
2	The robot was placed inside the pipe while using the standard FAT transmitter placement and running the sweep test with the robot inside the pipe. This was performed to measure the current connection for FAT setup.
3	The robot was placed inside the pipe and transmitter was relocated next to the laser tracker, whilst sweep test was running the robot inside the pipe. This was performed to measure if relocating the transmitter improved connection.

Table 3.7: Description of experimental test configurations.

The Fig. 3.4 illustrates the tests performed.



(a) Test 1 **(b)** Test 2 **(c)** Test 3

Figure 3.4: Experimental test configurations.

Since MR 3 already had relocated the Bluetooth transmitter, test 2 was not performed on that MR. The information of tests performed, pipe tested in, robot used, transmitter used are described in Table 3.8.

MR	Robot	Bluetooth Transmitter	Pipe No	Tests performed
1	1	1	4	1,2,3
2	2	2	4	1,2,3
3	3	3	2	1,3

Table 3.8: Test configurations performed in each MR.

3.1.6 Data Processing and Analysis

When the tests were performed the logged registers and RTT were evaluated. Statistical parameters were used to find maximum value, mean value, and standard deviation for RTT.

The logged registers were analyzed since some hypotheses have been thought to be connected to sudden power loss or internal errors, the primary focus was given to supply voltage `U_SUPPLY` and error/status registers `ERR_STAT` in order to understand potential correlation between movement, location and internal registers.

3.2 Battery Concepts Development

The battery pack required an updated solution and was critical. Since no complete electrical specification of the existing system was available, the concepts provided were based on the technical and usability requirements from previous battery packs and electrical components inside the robot.

The electrical requirements based on the robot are the MAC00-FB4 Bluetooth module and the MAC95 motor which requires a voltage of between 12V - 48V, and the safety fuse inside the robot of 7.5 A. The current batteries are 3.6 Ah, and documented from previous test done before the thesis illustrates that the robot approximately consumes 1 A when running, making the runtime currently at 3.6 hours.

Moreover, the battery holder dimensions can not be changed making the concepts provided, required to fit in the current holder. The interior of the battery holder can be slightly modified if needed.

Furthermore the battery concepts provided need to be CE-marked. And based on all those requirements, a list was created and labeled into technical requirements and is illustrated in Table 3.9.

No	Technical requirements
1	12V-48V
2	$\geq$ 3600mAh
3	$\geq$ 115Wh
4	Dimension to fit in current battery pack
5	BMS or PCM
6	European Conformity (CE)-marked

Table 3.9: Technical requirements of the new battery pack.

The technical requirement served as baseline and could not be changed. In addition to the technical requirements, further requirements were gathered to form an understanding on how the battery would be handled. These requirements were based on the operators experience with the previous battery packs and labeled as usability requirements, and are described in Table 3.10.

No	User requirements
1	Easy to charge
2	Easy to replace
3	Battery percentage indication

Table 3.10: User requirements of the new battery pack.

The Fig. 3.5 illustrates a simplification of the battery holder and its shape.

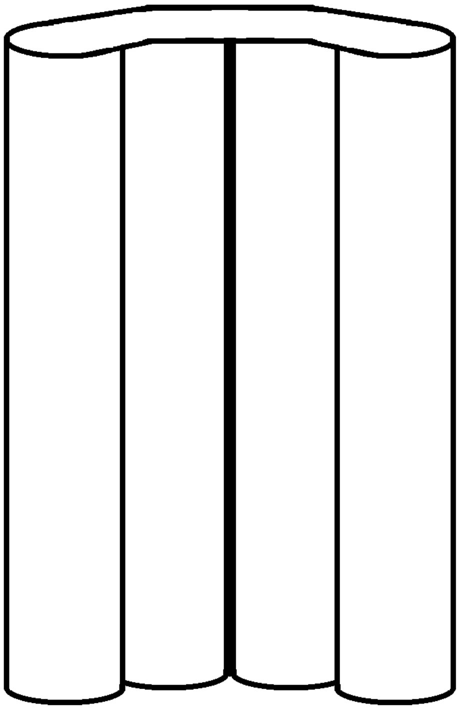


Figure 3.5: Simplified illustration of the battery holder (dimensionless).

In order to locate sufficient concepts based on the requirements listed in both Table 3.9 and Table 3.10, the battery development included researching the market, contacting potential vendors and evaluating commercially available pre-designed battery packs.

3.3 Spare Part Availability

The parts inside the robot are difficult to replace. Therefore, this section describes the methods used to identify potential replacement parts. The process consisted of three main steps.

The first step included researching the technical documents of the parts then creating a Bill Of Material (BOM) based on identifying the wear parts through a risk assessment, then filtering the BOM by availability to find the critical parts.

The second step included researching online to determine availability of the critical parts and evaluate if similar standardized parts exist.

The third step included evaluating the research, based on cost and other constraints.

4

Results

This section of the thesis includes the process, workflow and analyses made throughout the project.

4.1 Troubleshooting Communication

This section involves the test performed to ensure that the debug program worked as well as the measuring ranges sweep test involving the RTT and register values.

4.1.1 Validation of the Test and Parsing Scripts

To validate the test script under poor communication conditions, the SMA antenna was partially unscrewed to degrade the signal quality. This increased the communication noise and consequently the RTT. If a message was not received within 3000 ms, it was considered timed out, the RTT of the experiment is illustrated in Fig. 4.1.

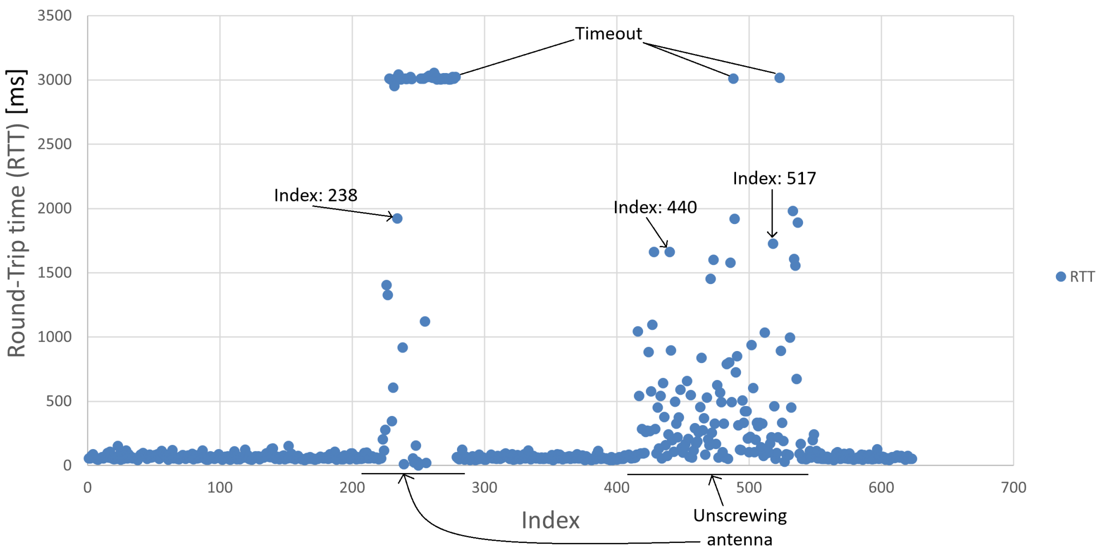


Figure 4.1: Unscrewing antenna on the robot.

The test performed in Fig. 4.1 resulted in 40 ms as minimum RTT, which is more than the calculated value of 10.8 ms from Equation 3.1, also validating that the full message was received. This is because the Equation 3.1 only accounts for the transportation time of the TX and RX messages, not the robot's internal processing time or other factors that could potentially delay the RTT.

In order to validate the parsing script, three different message indices from the antenna-unscrewing procedure shown in Fig. 4.1 were decoded and compared to the MacRegIO-decoded values. The indices chosen are visible in Fig. 4.1, and decoded in Table 4.1.

Name	**Index**	**Parse code decoded**	**MacRegIO decoded**
U_SUPPLY	238	602	602
ERR_STAT	440	0	0
ACC_EMERG	517	0	0

Table 4.1: Parse code decoded values vs MacRegIO decoded values.

The two tests validates that poor connection represents a high RTT without losing packages information. Therefore noise in form of poor connection is not an issue to the message reception in the test script. All the RTT are above 40 ms indicating that the calculated theoretical RTT value of 10.8 ms is below the measured RTT.

4.1.2 Test 1

The blue position illustrated is virtual since the robot was standing besides the transmitter and not moving in the pipe. The RTT result from test 1 using MR 1 setup is shown in Fig. 4.2.

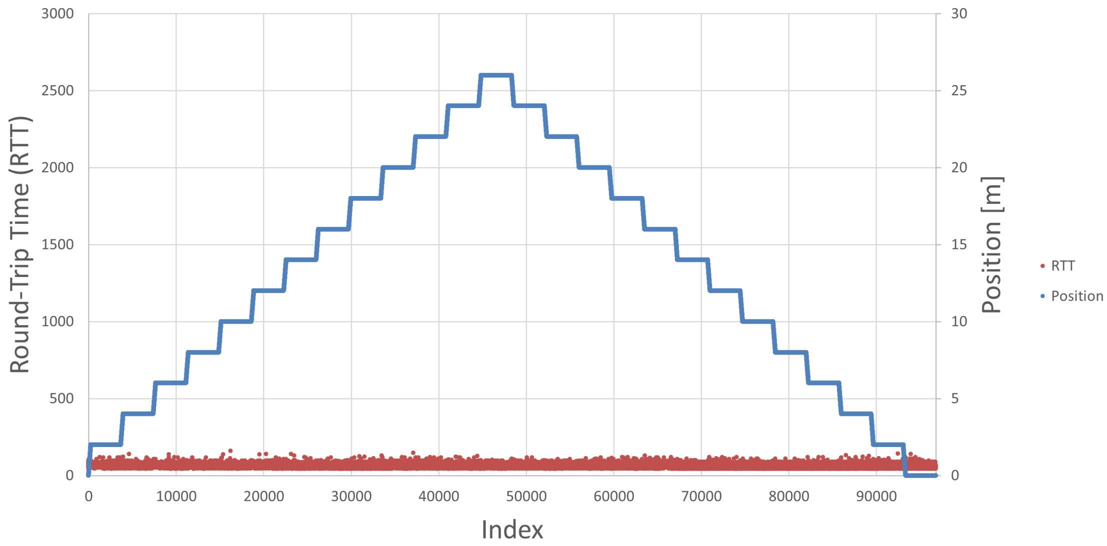


Figure 4.2: Graph of RTT and virtual position from test 1.

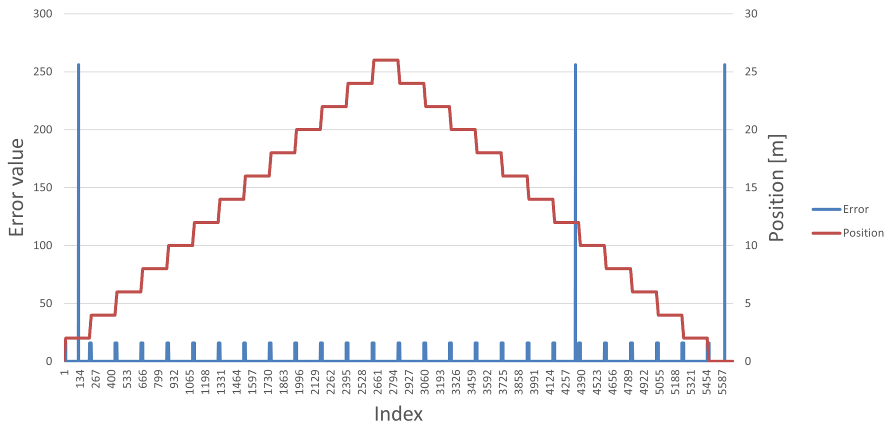
As illustrated in Fig. 4.2, the distribution of RTT is uniform, indicating a stable connection. Since the robot stopped and logged data for 200 s, and because the RTT remained uniform throughout this period, the robot was able to send approximately the same number of requests between each stop. Consequently, the resulting position increments appear evenly spaced.

When performing test 1 on MR 2 illustrated in appendix A.2.1 and MR 3 the RTT illustrated in appendix A.2.4, result was approximately the same which should be expected. The Table 4.2 summarizes the RTT including statistical calculation based from test 1 gathered from MR 1, MR 2 and MR 3.

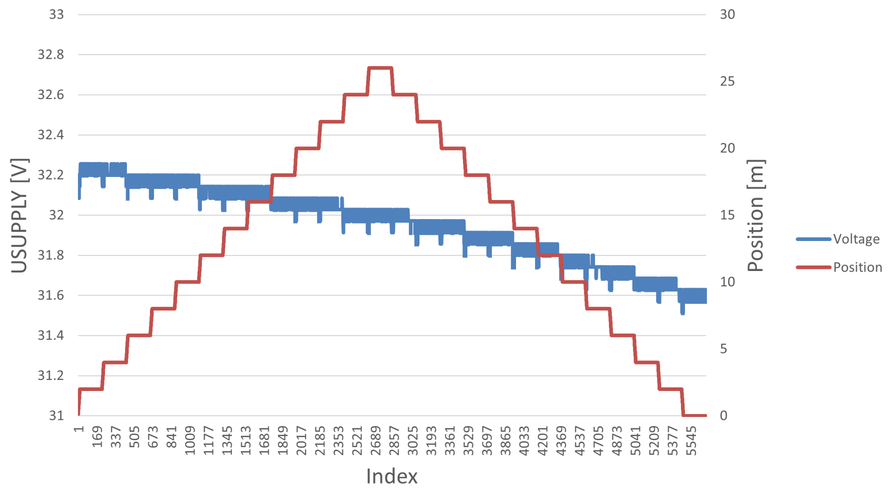
MR	Max RTT [ms]	STDEV [ms]	Mean value [ms]	Distribution
1	170	8	76	Uniform
2	160	8	72	Uniform
3	110	7	64	Uniform

Table 4.2: RTT values for each MR during test 1.

The registers value were reviewed for any error or sudden voltage drop. The register `U_SUPPLY` values was interpreted with Equation 3.2 into a voltage, and `ERR_STAT` graphs for MR 1 during test 1 are illustrated in Fig. 4.3.



(a) `ERR_STAT` value



(b) `U_SUPPLY` interpreted as voltage

Figure 4.3: `ERR_STAT` and `U_SUPPLY` values from test 1.

From Fig. 4.3 a) the value 16 is active whenever the robot is moving which is expected. From the test script the robot runs on velocity mode and value 16 (bit 4) indicates `AtVelocity` is active according to MAC manual page 93 in [10]. The peaks occur at the moment the robot begins to move indicating that the desired velocity is reached almost instantaneously, which is expected since the acceleration was set to be relatively high.

However, the three peaks at value 256 in Fig. 4.3 a) are not expected. The value indicates that bit 8 was active, which according to the MAC manual is a `FRAME_ERR_TX`. This error occurs when the message transmitted to the robot was not received correctly page 416 in [10].

From Fig. 4.3 b) the result from the voltage was expected, since current peaks occur whenever the MAC95 motor accelerates or decelerates, explained in the MAC manual. The result also indicates that no large voltage drop occurred.

The other error registers (FLWERR, FUNCERR and ACC_EMERG) were 0 during the entire test, indicating that no common internal motor error occurred with the robot during the test.

When performing test 1 on MR 2 and MR 3 the register values were similar to Fig. 4.3, with the only exception ERR_STAT register bit 8 was always 0 for MR 2 and MR 3. The register values for test 1 on MR 2 can be reviewed in A.2.1, and for test 1 on MR 3 can be reviewed in A.2.4. The Table 4.3 summarizes the register values from test 1 on MR 1, MR 2 and MR 3.

MR	ERR_STAT register	U_SUPPLY register [V]	Other error registers
1	0/16/256	32.3 → 31.5	0
2	0/16	32.3 → 31.9	0
3	0/16	31.9 → 31.1	0

Table 4.3: Register values for each MR during test 1

4.1.3 Test 2

After test 1 was performed indicating a stable connection, test 2 was performed and the RTT is illustrated in the Fig. 4.4 on MR 1.

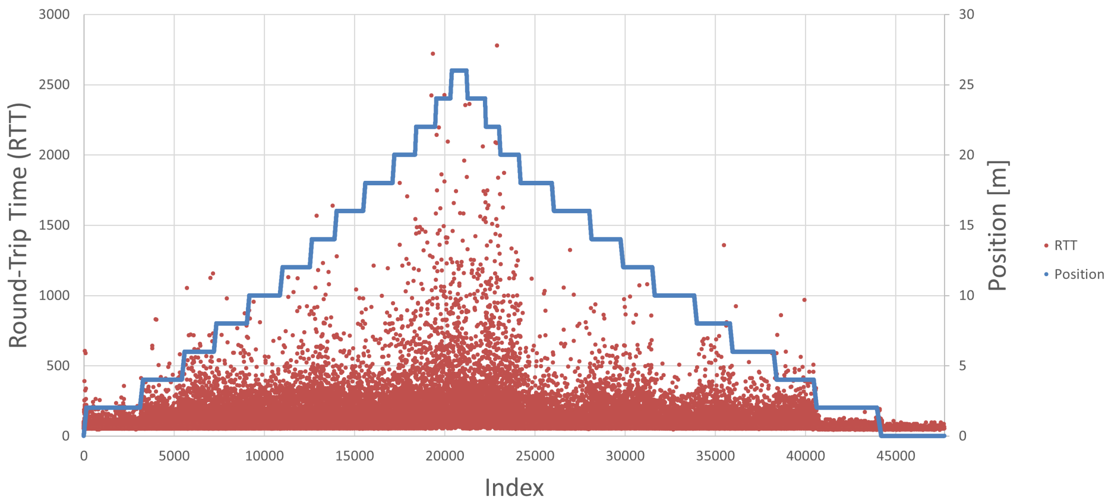


Figure 4.4: Graph of RTT and position from test 2.

The Fig. 4.4 illustrates the non-uniform distribution of RTT and how the RTT increases further in the pipe, with the existing FAT setup for MR 1. Since the robot stopped and logged for 200 s at each position, and because the RTT was non-uniform during these periods, the number of sent and received messages varied between the stops. As a result, the width of each position step becomes progressively shorter further in the pipe, since the step width directly correlates with how many cycles could be completed while the robot was stationary at that location.

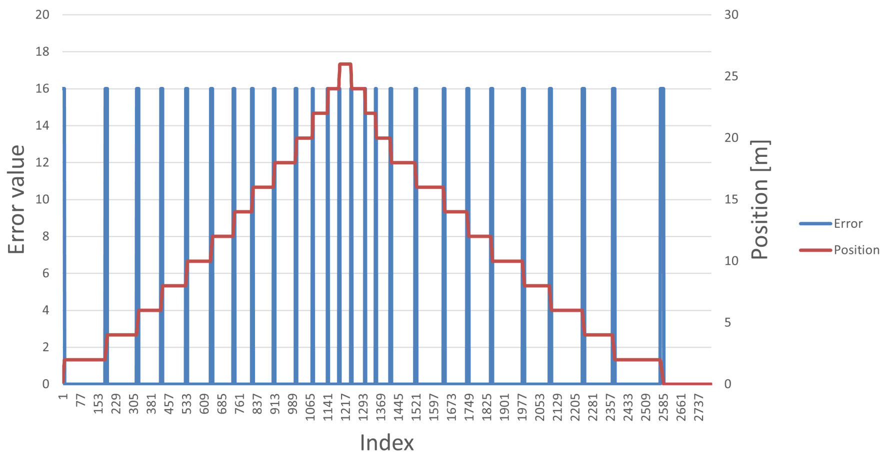
When performing test 2 on MR 2 the RTT illustrated in appendix, the result was similar to Fig. 4.4. The Table 4.4 summarizes test 2 RTT gathered from MR 1 and MR 2.

When performing test 2 on MR 2 the RTT which is illustrated in A.2.2, illustrates a similar graph, which was expected since the two measuring ranges share similar setup. The Table 4.4 summarizes test 2 RTT gathered from MR 1 and MR 2.

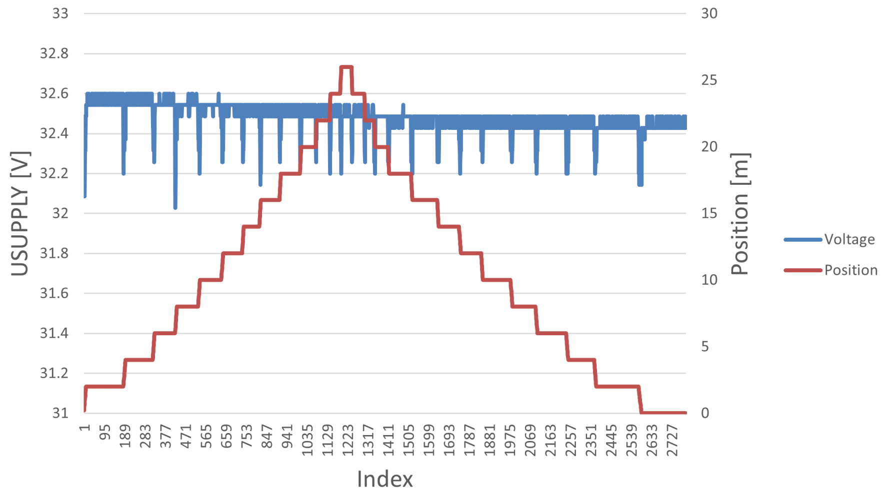
MR	Max RTT [ms]	STDEV [ms]	Mean value [ms]	Distribution
1	2780	133	370	non-Uniform
2	3001(timeout)	148	328	non-Uniform

Table 4.4: RTT values for each MR during test 2.

The register `U_SUPPLY` values was interpreted with Equation 3.2 into voltage, and the register `ERR_STAT` for MR 1 during test 2 is illustrated in 4.5.



(a) `ERR_STAT` value



(b) `U_SUPPLY` interpreted as voltage

Figure 4.5: `ERR_STAT` and `U_SUPPLY` values from test 2.

The error value in Fig. 4.5 a) indicates that `AtVelocity` was active whenever the robot moved, which was expected.

The voltage peaks are somewhat larger in Fig. 4.5 b) then during test 1 in Fig. 4.3 b) which indicates that the inner surface area of the pipe was a resistance to the robot, which was expected.

The other error registers (`FLWERR`, `FUNCERR` and `ACC_EMERG`) were 0 during the entire test, indicating that no common internal motor error occurred with the robot during the test.

The register values were similar during test 2 on MR 2, which expected since MR 1 and MR 2 share similar setup. The Table 4.5 summarizes the register values from test 2 for MR 1 and MR 2.

MR	ERR_STAT register	U_SUPPLY register [V]	Other error registers
1	0/16	32.6 → 32.0	0
2	0/16	32.7 → 31.7	0

Table 4.5: Register values for each MR during test 2

4.1.4 Test 3

After confirming longer RTT from test 2, test 3 was performed and the transmitter was relocated besides the laser. The result from MR 1 RTT is illustrated in Fig. 4.6.

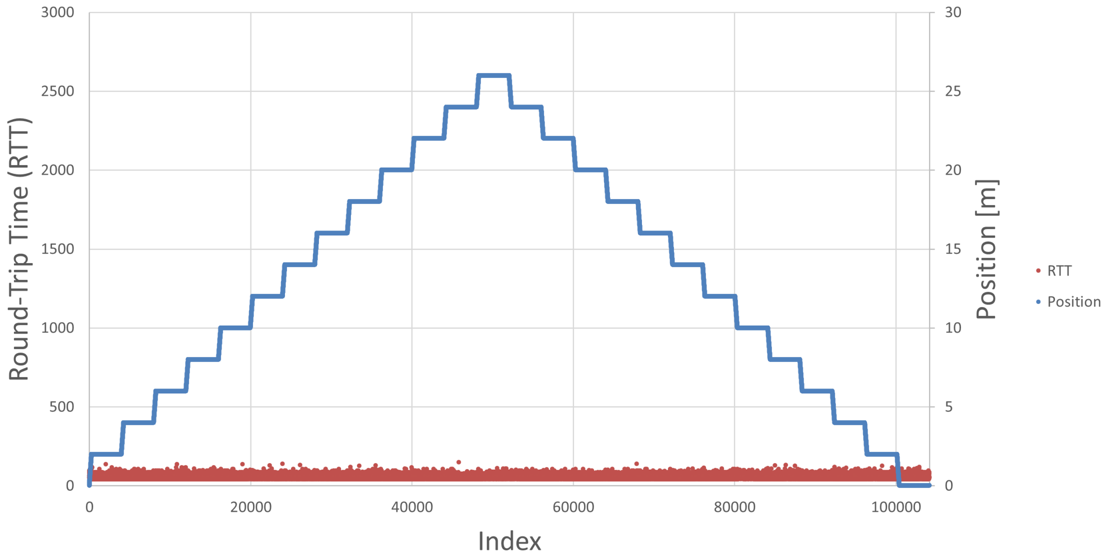


Figure 4.6: Graph of RTT and position from test 3.

As illustrated in Fig. 4.6, the distribution of RTT is uniform, indicating a stable connection. This was expected since moving the transmitter closer to the end of the pipe was one of the hypothesis to improve the connection.

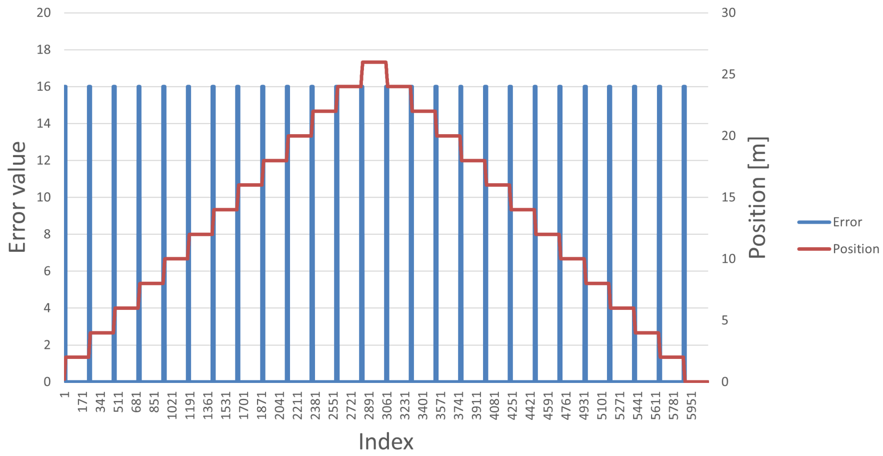
The RTT from performing test 3 on MR 2 is illustrated in A.2.3 and MR 3 is illustrated in A.2.5 and the results were similar to MR 1. This could be expected since the transmitter was relocated to the end of the pipe which is the standard location of the transmitter for MR 3. Table 4.6 summarizes the RTT of test 3 performed on MR 1, MR 2 and MR 3.

MR	Max RTT [ms]	STDEV [ms]	Mean value [ms]	Distribution
1	180	8	86.3	Uniform
2	182	8	91.1	Uniform
3	154	8	88.4	Uniform

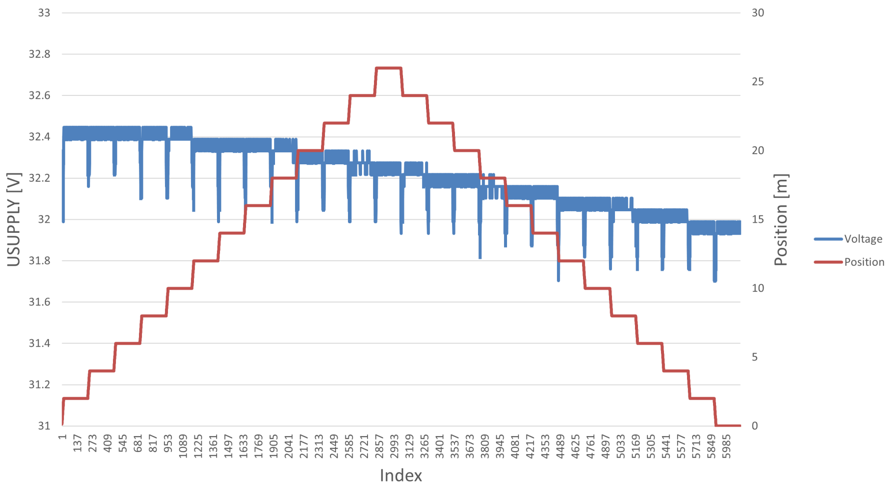
Table 4.6: RTT values for each MR during test 3.

The register `U_SUPPLY` values was interpreted with Equation 3.2 into voltage, and

ERR_STAT graph for MR 1 during test 3 is illustrated in 4.7.



(a) ERR_STAT value



(b) U_SUPPLY interpreted as voltage

Figure 4.7: ERR_STAT and U_SUPPLY values from test 3.

Reviewing the ERR_STAT in Fig. 4.7 a) illustrated a value 16 whenever the robot is moving, which is expected. The U_SUPPLY drop in Fig. 4.7 b) illustrates an expected lowered voltage over time with peaks when the robot accelerates and decelerates and could be expected. The register values after performing test 3 on MR 2 is illustrated in A.2.3 and MR 3 is illustrated in A.2.5. The results are similar to MR 1, which was expected. The Table 4.7 summarizes the register values from test 3 for MR 1, MR 2 and MR 3.

MR	**ERR_STAT register**	**U_SUPPLY register [V]**	**Other error registers**
1	0/16	32.5 → 31.7	0
2	0/16	32.5 → 31.9	0
3	0/16	32.5 → 31.7	0

Table 4.7: Register values for each MR during test 3

4.1.5 Observations of Occurred Stoppages and Bugs

The first time communication stopped was during test 2 with MR 1 when running the test script, steps taken are described:

1. Stopped sweep test for safety reasons.
2. MacTalk and MacRegIO were tested, but no connection was established.
3. Robot was located inside the pipe, no joint was found, approximately 18 m in.
4. MacTalk and MacRegIO tested again, still no communication.
5. Transmitter LED status was not checked.
6. Logs and register values were inspected before the stop, nothing unusual was found.
7. RTT values were similar to previous MR 2 test results.
8. Communication was suddenly restored after approximately 15 minutes without any manual action.
9. Error registers (`ERR_STAT`, `FLWERR`, `FUNCERR` and `ACC_EMERG`) were 0 from MacRegIO.
10. All values were normal, the root cause is most likely a communication issue.

The second time the robot stopped, was during FAT when using dRange. The same steps were taken however other observations were made. Steps and observations made:

1. Error message, stopped.
2. MacTalk and MacRegIO were started to test communication, but no connection was established.
3. Robot was located inside the pipe, no joint was found, approximately 25 m in, a high-pitched mechanical noise was observed from the robot.
4. Transmitter LED status was purple.
5. TGU was removed to unblock the pipe end. Communication was restored after approximately 15 minutes.
6. Error registers (`ERR_STAT`, `FLWERR`, `FUNCERR` and `ACC_EMERG`) 0, but mode register was 2 indicating position mode rather than velocity mode from MacRegIO.
7. Writing 0 then 2 again to register 2 in MacRegIO made it move.
8. Issue was communication-related and most likely due to a mode register value.

The software dRange also provided multiple errors related to laser values, register mode and voltage, when FAT responsible performed FAT utilizing MR 1 and MR 2.

4.2 Battery Concept Evaluation

This section includes evaluating the concepts provided for a potential battery pack.

4.2.1 Concept 1

Concept 1 consists of a custom battery pack based on lithium-ion cells connected through spot welding and managed by an external BMS. An illustration of the concept is shown in Fig. 4.8, where the:

- The numbers illustrate the individual cells in the battery pack.
- The yellow lines or dots indicate where the cells should be spot welded.
- The blue arrows show the connection of the cells to achieve a series connection.
- The thin black splines and lines from each yellow point represent the sense leads used to monitor the voltage of each individual cell with the BMS.
- The + and **-** are the charging ports.

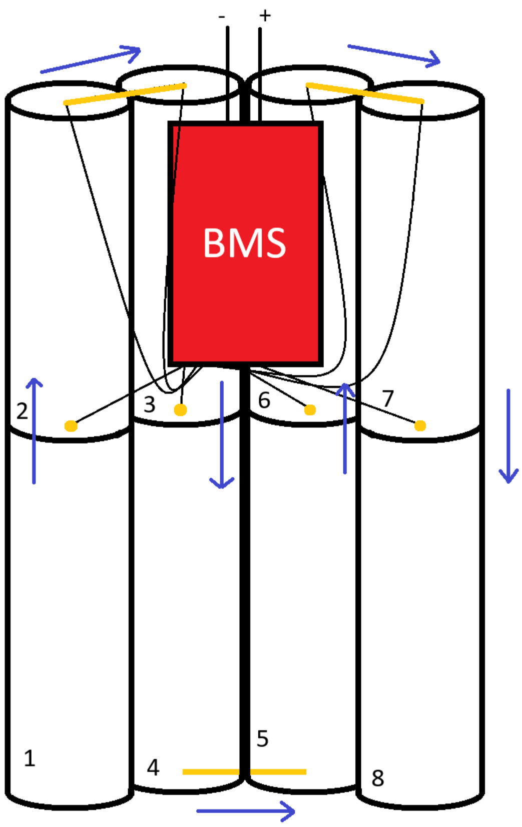


Figure 4.8: Illustration of concept 1.

This configuration can be achieved by commercially available unprotected cells, such as the **Tenpower INR21700-50ME** [14] and can be assembled at a battery facility in Kungsbacka. The external BMS could be a **DALY K Series Smart BMS** [15].

Based on this the resulting technical specification of the pack is described in Table 4.8.

Parameter	**Current Battery Pack**	**Concept 1 Battery Pack**
Cell format	18650	21700
Capacity	~3.6 Ah	~5.0 Ah
Cells in series	8	8
Nominal voltage	28.8 V	28.8 V
Estimated runtime	~3.6 hours	~5 hours
Protection	Individually protected	External BMS
Monitoring capability	None	Advanced (SoC, temp, power)
Dimensions	Fits	Fits with minor interior CAD changes
CE-marked	No	Yes

Table 4.8: Comparison of the technical specification for the current battery pack and concept 1.

The concept fulfills the technical requirements but it only fulfills some usability requirements which is described in Table 4.9.

Parameter	**Current Battery Pack**	**Concept 1 Battery Pack**
Easy to Charge	No	Yes
Easy to replace	Yes	No
Battery indication	No	Yes

Table 4.9: Comparison of the usability requirements for the current battery pack and concept 1.

4.2.2 Concept 2

Concept 2 is instead based on commercially available battery modules to form a battery pack. The Fig. 4.9 illustrates 4 cylindrical packs of **HCC21700W-2S1P** from Honcell [16].

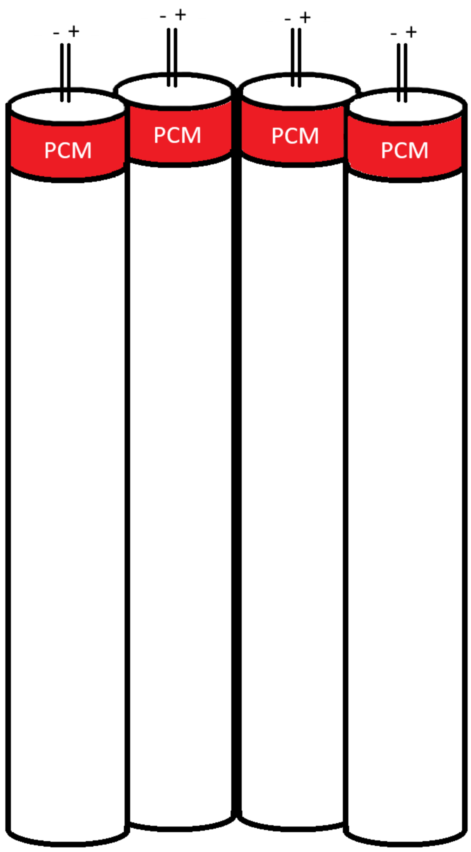


Figure 4.9: Illustration of concept 2.

The packs can be connected in series through JST connectors, but it is not recommended according through an email from Honcell. The packs each have a PCM which makes the charging individual per pack therefore requiring 4 chargers. Despite this the technical specification of the pack can be illustrated in Table 4.10

Parameter	**Current Battery Pack**	**Concept 2 Battery Pack**
Cell format	18650	21700
Capacity	∼3.6 Ah	∼5.0 Ah
Cells in series	8	8
Nominal voltage	28.8 V	29.2 V
Estimated runtime	∼3.6 hours	∼5 hours
Protection	Individually protected cells	Individually PCM protected packs
Monitoring capability	None	None
Dimensions	Fits	Fits with minor interior CAD changes
CE-marked	No	Yes

Table 4.10: Comparison of the technical specification for the current battery pack and concept 2.

The concept 2 fulfills the technical requirements but it only fulfills some usability requirements which is described in Table 4.11.

Parameter	**Current Battery Pack**	**Concept 2 Battery Pack**
Easy to Charge	No	Mediocre
Easy to replace	Yes	Yes
Battery indication	No	No

Table 4.11: Comparison of the usability requirements for the current battery pack and concept 2.

4.2.3 Concept 3

Concept 3 is theoretical and based on the existing battery pack but improved using larger protected lithium-ion cells, such as **5000mAh 10A 21700 Button Top Battery** [17], and adapters such as **21700 Battery Holder for PCB Mounting** [18], instead of spring loading the battery pack.

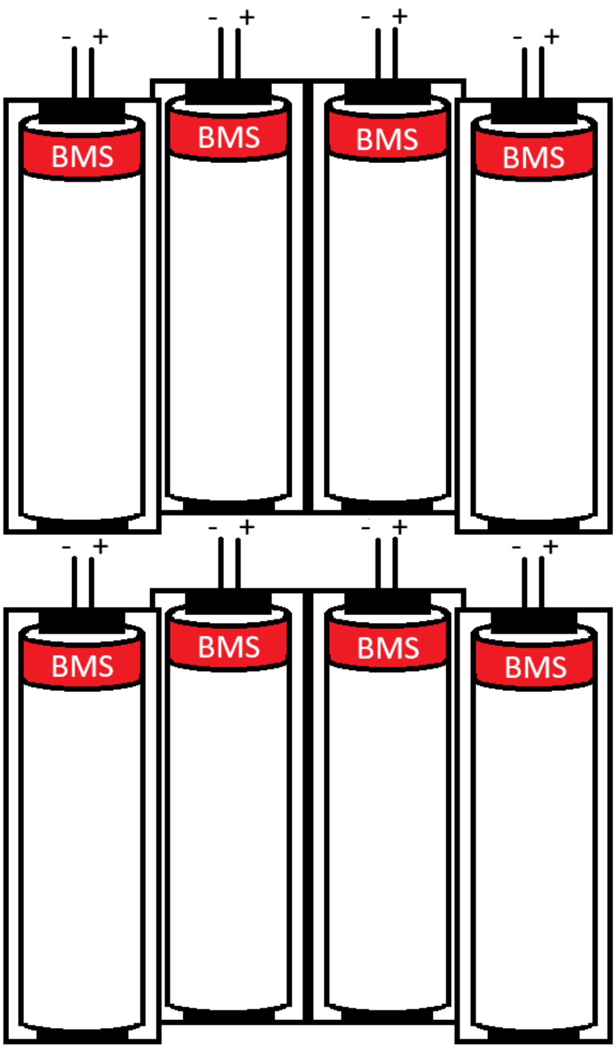


Figure 4.10: Illustration of concept 3.

Based on this concept, it still faces the same problems the current battery pack has, however it ensures tighter connection. The technical specification is described in Table 4.12.

Parameter	**Current Battery Pack**	**Concept 3 Battery Pack**
Cell format	18650	21700
Capacity	~3.6 Ah	~5.0 Ah
Cells in series	8	8
Nominal voltage	28.8 V	28.8 V
Estimated runtime	~3.6 hours	~5 hours
Protection	Individually protected cells	Individually protected cells
Monitoring capability	None	None
Dimensions	Fits	Unknown
CE-marked	No	Yes

Table 4.12: Comparison of the technical specification for the current battery pack and concept 3.

The concept 3 fulfills the technical requirements but it only fulfills some usability requirements which is described in Table 4.13.

Parameter	**Current Battery Pack**	**Concept 3 Battery Pack**
Easy to Charge	No	No
Easy to replace	Yes	Yes
Battery indication	No	No

Table 4.13: Comparison of the usability requirements for the current battery pack and concept 3.

4.3 Spare Part

To evaluate the long term maintainability of the current robot, a spare part analysis was conducted. The purpose of this analysis was to identify critical components and evaluate potential replacement parts of the robot. This section will include a risk analysis that is used in the BOM list.

4.3.1 Identifying Critical Parts

Based on the approximately 72 parts that the robot consists of, a risk analysis was conducted to map the critical parts. To identify the critical components all the parts of the robot was given a quick evaluation based on previous failure history. Since the robot consists of 72 parts, 63 parts were quickly grouped into a low risk. These components include the structural components. The other 9 parts were placed into medium or high risk based on previous failure history and include the drive package and electrical parts. The result of the risk analysis is demonstrated in Fig. 4.11.

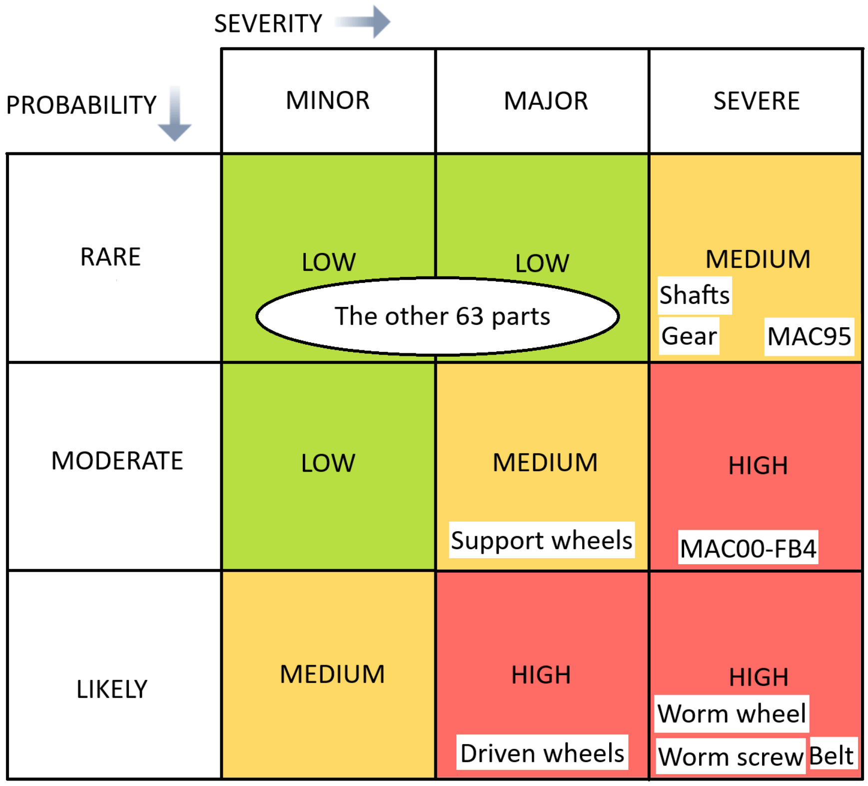


Figure 4.11: Risk graph of the robot's parts.

Based on the risk provided per part in Fig. 4.11, the medium and high risks parts were further evaluated in a BOM. This was based on if the part is standardized, if Emerson currently has a vendor connected, and if the part is Made To Order (MTO) or can be purchased of the shelf. This is shown in Table 4.14.

Part	Vendor connected	Part	Risk	Availability status
Worm wheel	No	Custom	High	Unknown
Worm screw	Yes	Custom	High	MTO
Shafts	No	Custom	Medium	Unknown
Gear	No	Custom	Medium	Unknown
Belt	Yes	Standard	High	Available
MAC95	Yes	Standard	Medium	Available
MAC00-FB4	Yes	Standard	High	Discontinued
Driven wheels	No	Custom	High	Unknown
Support wheels	No	Custom	Medium	Unknown

Table 4.14: BOM list of the medium and high risk wear parts.

The identified BOM of the parts with higher risks were further investigated to understand if the parts had a vendor connected or the product was discontinued. This provided the critical parts and can be illustrated in Table 4.15.

Part	Vendor connected	Part	Risk	Availability status
Worm wheel	No	Custom	High	Unknown
Shafts	No	Custom	Medium	Unknown
Gear	No	Custom	Medium	Unknown
MAC00-FB4	Yes	Standard	High	Discontinued
Driven wheels	No	Custom	High	Unknown
Support wheels	No	Custom	Medium	Unknown

Table 4.15: BOM list of the Critical wear parts.

If any of the listed parts in Table 4.15 were to fail, the robot would stop and repair would delay.

4.3.2 Research of Critical Parts

In order to find replacement parts of the listed parts in Table 4.15 research online and an evaluation was performed. Based on the results of the research no company offered a standardized, easy to purchase solution of the parts in Table 4.15.

Since Emerson has a connected vendor to the worm screw in Table 4.14, that vendor could potentially produce the customized mechanical parts in Table 4.15.

The MAC00-FB4 is discontinued in JVL assortment, but not obsolete yet, making the part orderable at an expensive price. When the part is obsolete however JVL offers no replacement making the component critical. The market offers small micro controllers that can serve as a custom interface solution.

5 Conclusion

This section summarizes the main findings based on the result gathered throughout the work, and evaluates the result.

5.1 Discussion of Communication Results

Emerson employees had previously identified communication instability as a recurring issue during operation. The conducted tests illustrated how relocating the Bluetooth transmitter improved the connection. In addition with developing the test script, it was discovered that many of the issues causing stoppages were most likely related to the software dRange or bad state of the robot's move register, in addition to the Bluetooth transmitter placement. To improve communication in the future, relocating the transmitter to the end of the pipe by purchasing two longer cables will hopefully suffice, and if the robot's register enters a bad state, this should be resolved in the software dRange. Future work could include troubleshooting the communication interfaces from the computer to the larger modules, to hopefully resolve Leica laser error experienced in dRange and other communication errors.

5.2 Discussion of Battery Concepts and Spare Parts

Prior to the project Emerson had considered investing in a new robot due to recurring issues with both communication and software, however the results from the project illustrated that the current robot is sufficient if the transmitter is relocated to the end of the pipe. Therefore investing in the current robots and finding spare parts was considered more important.

Based on the concepts provided and developed it was determined that concept 1 from subsection 4.2.1 would be the most suitable approach since it fulfills the technical and operational requirements. The provided vendor is also located in Gothenburg, simplifying the technical support. Even though the commercially available standardized battery pack solutions exists, no solution could meet the technical and user requirements.

Furthermore, the spare part availability demonstrates that several components within the robot are critical for long term maintainability, in addition to this, it was also discovered that the critical parts are custom made, making the replacement into standardized components difficult. This concludes that the critical parts need to be custom made at a workshop. Moreover, the discontinued MAC00-FB4 Bluetooth module is essential for wireless communication, and by ordering and storing a number of it in stock, the robot will be able to continue operating for many years.

Bibliography

[1] Emerson Electric Co. (2026). *Measurement Instrumentation*. Accessed: 8 Feb, 2026. Available at: https://www.emerson.com/en-us/automation/measurement-instrumentation

[2] Emerson Electric Co. (2024). *Rosemount TGU 53 Tank Radar Gauge*. Accessed: 8 Feb, 2026. Available at: https://www.emerson.com/en/measurement-instrumentation/products/rosemount-tgu-53-tank-radar-gauge

[3] Özay, M. E. (2022). *CAST Application: A Case Study of CAPECO Multiple Tank Explosion*. SDU Journal of Engineering Sciences & Design. Accessed: 12 Feb, 2026. Available at: https://dergipark.org.tr/en/download/article-file/1763534

[4] Amnesty International. (2021). *Hållbara litiumjonbatterier*. Accessed: 12 Feb, 2026. Available at: https://www.amnesty.se/vara-rattighetsfragor/foretagsansvar/hallbara-litiumjonbatterier/

[5] Leica Geosystems AG (2016). *Leica AT930/AT960 User Manual Version 1.2*. Accessed: 20 Feb, 2026. Available at: https://supportcenter.nexus.hexagon.com/service/s/article/Leica-Absolute-Tracker-AT930-AT960-User-Manual-v1-2-0en-1528374732805?language=en_US

[6] JVL A/S. (2024). *MacTalk – Integrated Servo Motors Software*. Accessed: 23 Feb, 2026. Available at: https://www.jvl.dk/523/software-mac-motor

[7] JVL A/S. (2024). *MACREGIO – MacTalk Protocol Expert Tool for Programmers*. Accessed: 23 Feb, 2026. Available at: https://www.jvl.dk/526/tool-for-programmers

[8] Embedded.com. (2013). *connectBlue launches dual mode Bluetooth Smart Ready serial port adapter*. Accessed: 23 Feb, 2026. Available at: https://www.embedded.com/connectblue-launches-dual-mode-bluetooth-smart-ready-serial-port-adapter/

[9] Emerson Electric Co. (2026). *Rosemount Supply and Communications Unit*. Accessed: 8 Feb, 2026. Available at: https://www.emerson.com/en/measurement-instrumentation/products/rosemount-supply-and-communications-unit

[10] JVL A/S. (2024). *Integrated Servo Motors MAC050–MAC4500: User Manual*. Accessed: 23 Feb, 2026. Available at: https://www.jvl.dk/files/pdf-1/user%20manuals/lb0047gb.pdf

[11] JVL A/S. (2006). *MAC00-FB4 Bluetooth Expansion Module: Technical Data Sheet*. Accessed: 8 Feb, 2026. Available at: https://www.jvl.dk/files/_2011clean/pdf/ld068gb.pdf

[12] Honcell Energy Co., Ltd. (2025). *Lithium-ion Battery Cell Structure and Working Principle*. Accessed: 16 Mar, 2026. Available at: https://www.honcell.com/news/industry/59

[13] Dan-Tech Energy GmbH. (2024). *PCM vs. BMS: Understanding the Key Differences and Which One Your Battery Needs*. Accessed: 16 Mar, 2026. Available at: https://www.danenergy.com/blog-posts/pcm-vs-bms

[14] NKON. (2024). *Tenpower INR21700-50ME 5000mAh – 15A*. Accessed: 16 Mar, 2026. Available at: https://www.nkon.nl/en/tenpower-inr21700-50me-5000mah-15a.html

[15] Daly BMS. (2024). *K Series Standard BMS 3S–24S*. Accessed: 16 Mar, 2026. Available at: https://www.dalybms.com/daly-smart-bms-electric-bicycle-scooter-lithium-lifepo4-lfp-battery-4-24s-30500a-nmclfplto-7s-8s-200a-500a-smart-programmable-bms-product/

[16] Honcell Energy. (2024). *HCC21700W-2S1P Cylindrical Battery Pack*. Accessed: 16 Mar, 2026. Available at: https://www.honcell.com/lithium-battery/packs/hcc-cylindrical-battery-packs/2768

[17] LiionWholesale. (2024). *Protected 5000mAh 15A 21700 Button Top Battery*. Accessed: 16 Mar, 2026. Available at: https://liionwholesale.com/products/protected-5000mah-15a-21700-button-top-battery-wholesale-discount?srsltid=AfmBOoqfpPGlcve2RS8fwx9fk8_4OngVyHm5mL8VtWqVuBePkVvVIQDG&variant=39848920744005

[18] Electrokit Sweden AB. (2026). *Battery holder 1x 21700 PCB pin*. Accessed: 16 Mar, 2026. Available at: https://www.electrokit.com/en/battery-holder-1x-21700-pcb-pin

A

Appendix 1

A.1 Code

```
import serial
import time
from datetime import datetime
from typing import Dict, Any
import keyboard

filename = "log" + datetime.now().strftime("%H-%M-%S.%f")[:-3] + ".txt"
ser = serial.Serial(port='COM4', baudrate=19200, parity=serial.PARITY_NONE)

read_register: Dict[str, Dict[str, Any]] = {
    "READ_MODE_REG":   {"reg_num": 2,   "type": "integer", "data": bytes([0x50,0x50,0x50,0xFF,0x00,0x02,0xFD,0xAA,0xAA])},
    "READ_P_SOLL":     {"reg_num": 3,   "type": "longint", "data": bytes([0x50,0x50,0x50,0xFF,0x00,0x03,0xFC,0xAA,0xAA])},
    "READ_V_SOLL":     {"reg_num": 5,   "type": "integer", "data": bytes([0x50,0x50,0x50,0xFF,0x00,0x05,0xFA,0xAA,0xAA])},
    "READ_A_SOLL":     {"reg_num": 6,   "type": "integer", "data": bytes([0x50,0x50,0x50,0xFF,0x00,0x06,0xF9,0xAA,0xAA])},
    "READ_T_SOLL":     {"reg_num": 7,   "type": "integer", "data": bytes([0x50,0x50,0x50,0xFF,0x00,0x07,0xF8,0xAA,0xAA])},
    "READ_P_SIM":      {"reg_num": 8,   "type": "longint", "data": bytes([0x50,0x50,0x50,0xFF,0x00,0x08,0xF7,0xAA,0xAA])},
    "READ_P_IST":      {"reg_num": 10,  "type": "longint", "data": bytes([0x50,0x50,0x50,0xFF,0x00,0x0A,0xF5,0xAA,0xAA])},
    "READ_V_IST":      {"reg_num": 12,  "type": "integer", "data": bytes([0x50,0x50,0x50,0xFF,0x00,0x0C,0xF3,0xAA,0xAA])},
    "READ_KVOUT":      {"reg_num": 13,  "type": "fixed8",  "data": bytes([0x50,0x50,0x50,0xFF,0x00,0x0D,0xF2,0xAA,0xAA])},
    "READ_12T":        {"reg_num": 16,  "type": "word",    "data": bytes([0x50,0x50,0x50,0xFF,0x00,0x10,0xEF,0xAA,0xAA])},
    "READ_UIT":        {"reg_num": 18,  "type": "word",    "data": bytes([0x50,0x50,0x50,0xFF,0x00,0x12,0xED,0xAA,0xAA])},
    "READ_FLWERR":     {"reg_num": 20,  "type": "longint", "data": bytes([0x50,0x50,0x50,0xFF,0x00,0x14,0xEB,0xAA,0xAA])},
    "READ_FNCERR":     {"reg_num": 24,  "type": "longint", "data": bytes([0x50,0x50,0x50,0xFF,0x00,0x18,0xE7,0xAA,0xAA])},
    "READ_ACCEMERG":   {"reg_num": 32,  "type": "word",    "data": bytes([0x50,0x50,0x50,0xFF,0x00,0x20,0xDF,0xAA,0xAA])},
```

```
    "READ_ERROR_STAT": {"reg_num": 35,  "type": "word",    "data":
       bytes([0x50,0x50,0x50,0xFF,0x00,0x23,0xDC,0xAA,0xAA])},
    "READ_CNTRL_BITS": {"reg_num": 36,  "type": "integer", "data":
       bytes([0x50,0x50,0x50,0xFF,0x00,0x24,0xDB,0xAA,0xAA])},
    "READ_U_SUPPLY":   {"reg_num": 151, "type": "word",    "data":
       bytes([0x50,0x50,0x50,0xFF,0x00,0x97,0x68,0xAA,0xAA])},
}

write_register_vel_for: Dict[str, Dict[str, Any]] = {
    "WRITE_A_SOLL_50": {"reg_num": 6,"type": "integer","data":
       bytes([0x52,0x52,0x52,0xFF,0x00,0x06,0xF9,0x02,0xFD,0x32,0
       xCD,0x00,0xFF,0xAA,0xAA,0x50,0x50,0x50,0xFF,0x00,0x06,0xF9,0
       xAA,0xAA])},
    "WRITE_KVOUT_1": {"reg_num": 13, "type": "fixed8","data": bytes
       ([0x52, 0x52, 0x52, 0xFF, 0x00, 0x0D, 0xF2, 0x02, 0xFD, 0x00
       , 0xFF, 0x01, 0xFE, 0xAA, 0xAA,0x50, 0x50, 0x50, 0xFF, 0x00,
        0x0D, 0xF2, 0xAA, 0xAA])},
    "WRITE_V_SOLL_1000": {"reg_num": 5, "type": "integer","data":
       bytes([0x52, 0x52, 0x52, 0xFF, 0x00, 0x05, 0xFA, 0x02, 0xFD,
        0x18, 0xE7, 0xFC, 0x03, 0xAA, 0xAA, 0x50, 0x50, 0x50, 0xFF,
        0x00, 0x05, 0xFA, 0xAA, 0xAA])},
    "WRITE_MODE_REG_1": {"reg_num": 2,"type": "integer","data":
       bytes([0x52, 0x52, 0x52, 0xFF, 0x00, 0x02, 0xFD, 0x02, 0xFD,
        0x01, 0xFE, 0x00, 0xFF, 0xAA, 0xAA, 0x50, 0x50, 0x50, 0xFF,
        0x00, 0x02, 0xFD, 0xAA, 0xAA])}
}

write_register_vel_back: Dict[str, Dict[str, Any]] = {
    "WRITE_A_SOLL_50": {"reg_num": 6,"type": "integer","data":
       bytes([0x52,0x52,0x52,0xFF,0x00,0x06,0xF9,0x02,0xFD,0x32,0
       xCD,0x00,0xFF,0xAA,0xAA,0x50,0x50,0x50,0xFF,0x00,0x06,0xF9,0
       xAA,0xAA])},
    "WRITE_KVOUT_1": {"reg_num": 13, "type": "fixed8","data": bytes
       ([0x52, 0x52, 0x52, 0xFF, 0x00, 0x0D, 0xF2, 0x02, 0xFD, 0x00
       , 0xFF, 0x01, 0xFE, 0xAA, 0xAA,0x50, 0x50, 0x50, 0xFF, 0x00,
        0x0D, 0xF2, 0xAA, 0xAA])},
    "WRITE_V_SOLL_n1000": {"reg_num": 5, "type": "integer","data":
       bytes([0x52, 0x52, 0x52, 0xFF, 0x00, 0x05, 0xFA, 0x02, 0xFD,
        0xE8, 0x17, 0x03, 0xFC, 0xAA, 0xAA, 0x50, 0x50, 0x50, 0xFF,
        0x00, 0x05, 0xFA, 0xAA, 0xAA])},
    "WRITE_MODE_REG_1": {"reg_num": 2,"type": "integer","data":
       bytes([0x52, 0x52, 0x52, 0xFF, 0x00, 0x02, 0xFD, 0x02, 0xFD,
        0x01, 0xFE, 0x00, 0xFF, 0xAA, 0xAA, 0x50, 0x50, 0x50, 0xFF,
        0x00, 0x02, 0xFD, 0xAA, 0xAA])}
}


write_register_pos_stop: Dict[str, Dict[str, Any]] = {
    "WRITE_V_SOLL_0": {"reg_num": 5, "type": "integer","data":
       bytes([0x52, 0x52, 0x52, 0xFF, 0x00, 0x05, 0xFA, 0x02, 0xFD,
        0x00, 0xFF, 0x00, 0xFF, 0xAA, 0xAA, 0x50, 0x50, 0x50, 0xFF,
        0x00, 0x05, 0xFA, 0xAA, 0xAA])},
    "WRITE_MODE_REG_0": {"reg_num": 2,"type": "integer","data":
       bytes([0x52,0x52,0x52,0xFF,0x00,0x02,0xFD,0x02,0xFD,0x00,0
       xFF,0x00,0xFF,0xAA,0xAA,0x50,0x50,0x50,0xFF,0x00,0x02,0xFD,0
       xAA,0xAA])},
```

```
    "WRITE_P_SIM_0": {"reg_num": 8,"type": "longint","data": bytes([0x52, 0x52, 0x52, 0xFF, 0x00, 0x08, 0xF7, 0x04, 0xFB, 0x00, 0xFF, 0x00, 0xFF, 0x00, 0xFF, 0x00, 0xFF, 0xAA, 0xAA, 0x50, 0x50, 0x50, 0xFF, 0x00, 0x08, 0xF7, 0xAA, 0xAA])},
    "WRITE_P_IST_0": {"reg_num": 10,"type": "longint","data": bytes([0x52, 0x52, 0x52, 0xFF, 0x00, 0x0A, 0xF5, 0x04, 0xFB, 0x00, 0xFF, 0x00, 0xFF,0x00, 0xFF, 0x00, 0xFF, 0xAA, 0xAA, 0x50, 0x50, 0x50, 0xFF, 0x00, 0x0A, 0xF5, 0xAA, 0xAA])},    # P_SOLL (reg 3)
    "WRITE_P_SOLL_0": {"reg_num": 3, "type": "longint","data": bytes([0x52, 0x52, 0x52, 0xFF, 0x00, 0x03, 0xFC, 0x04, 0xFB, 0x00, 0xFF, 0x00, 0xFF,0x00, 0xFF, 0x00, 0xFF, 0xAA, 0xAA, 0x50, 0x50, 0x50, 0xFF, 0x00, 0x03, 0xFC, 0xAA, 0xAA])},
}

def tx(ser, package):
    ser.write(package)
    #print("Data transmitted:", package)

def rx(ser, timeout=3):
    rx_buffer = b''
    start_time = time.time()
    while True:
        rx_buffer += ser.read_all()
        start_idx = rx_buffer.find(b'\x52\x52\x52')
        if start_idx != -1:
            end_idx = rx_buffer.find(b'\xAA\xAA', start_idx)
            if end_idx != -1:
                message = rx_buffer[start_idx:end_idx+2]
                rx_buffer = rx_buffer[end_idx+2:]  # Remove buffer
                #print("Message:", message)
                return message
        # Timeout check
        if time.time() - start_time > timeout:
            #print("Message was not received complete within", timeout, "seconds")
            return b''  # Empty answer

def save_log(name, time1, transmit, time2, respond, time3, error, deltatx, deltarx):
    with open(filename, "a") as f:
        f.write(
            f"{name}||;;{time1}||;;{transmit}||;;{transmit.hex()}||;;"
            f"{time2}||;;{respond}||;;{respond.hex()}||;;"
            f"{time3}||;;{error}||;;{deltatx:.2f}||;;{deltarx:.2f}\n"
        )

def send_reg_info(ser, regis):
    for reg_name, reg_info in regis.items():
        attempts = 0
        respond = b''
        while attempts < 3:
            t1 = time.time()
            time1 = datetime.now().strftime("%H:%M:%S.%f")[:-3]
```

```
            tx(ser, reg_info['data'])
            t2 = time.time()
            time2 = datetime.now().strftime("%H:%M:%S.%f")[:-3]
            respond = rx(ser)
            t3 = time.time()
            time3 = datetime.now().strftime("%H:%M:%S.%f")[:-3]
            deltatx = (t2 - t1)*1000
            deltarx = (t3 - t2)*1000
            error = "timeout" if respond == b'' else ""
            save_log(reg_name, time1, reg_info['data'], time2,
               respond, time3, error, deltatx, deltarx)
            attempts += 1
            if respond != b'':
                break
            else:
                print(f"No answer, trying {attempts}/3")

def sweep(ser):
    rounds = 13  # 13 trips of 2m approx 26m in total
    run_time = 17 # 17 s is approx 2m
    stop_time = 200 # Stop length in s
    directions = [write_register_vel_for, write_register_vel_back]
    for direction in directions:
        for _ in range(rounds):
            send_reg_info(ser, direction)
            start_time = time.time()
            while time.time() - start_time < run_time:
                if keyboard.is_pressed("space"):
                    send_reg_info(ser, write_register_pos_stop)
                if keyboard.is_pressed("q"):
                    send_reg_info(ser, write_register_pos_stop)
                    exit(1)
                send_reg_info(ser, read_register)
            send_reg_info(ser, write_register_pos_stop)
            start_time = time.time()
            while time.time() - start_time < stop_time:
                if keyboard.is_pressed("q"):
                    send_reg_info(ser, write_register_pos_stop)
                    exit(1)
                send_reg_info(ser, read_register)

sweep(ser)
ser.close()
print("\n Saved in: ", filename)
```

Listing A.1: Python code for Sweep

```python
import pandas as pd
from openpyxl import load_workbook

REG_TYPES = {
    "MODE_REG":   "integer",
    "P_SOLL":     "longint",
    "V_SOLL":     "integer",
    "A_SOLL":     "integer",
    "T_SOLL":     "integer",
    "P_SIM":      "longint",
    "P_IST":      "longint",
    "V_IST":      "integer",
    "KVOUT":      "fixed8",
    "12T":        "word",
    "UIT":        "word",
    "FLWERR":     "longint",
    "FNCERR":     "longint",
    "ACCEMERG":   "word",
    "ERROR_STAT": "word",
    "CNTRL_BITS": "integer",
    "U_SUPPLY":   "word",
}

def hex_to_bytes(hex_str):
    if not hex_str:
        return []
    hex_str = hex_str.replace(" ", "")
    return [int(hex_str[i:i+2], 16) for i in range(0, len(hex_str), 2)]

def decode_value_bytes(data_bytes, reg_type):
    if reg_type in ["integer", "word"]:
        if len(data_bytes) < 2:
            return None
        return data_bytes[0] | (data_bytes[1] << 8)
    elif reg_type == "longint":
        if len(data_bytes) < 4:
            return None
        val = data_bytes[0] | (data_bytes[1]<<8) | (data_bytes[2]<<16) | (data_bytes[3]<<24)
        if val >= 2**31:
            val -= 2**32
        return val
    elif reg_type == "fixed8":
        if len(data_bytes) < 2:
            return None
        return data_bytes[0] + data_bytes[1]/256
    return None

def decode(reg_name, rx_hex):
    if not isinstance(reg_name, str) or not reg_name.startswith("READ_"):
        return None
    if not isinstance(rx_hex, str) or not rx_hex:
        return None
```

```
    reg = reg_name[5:]
    reg_type = REG_TYPES.get(reg, "unknown")
    rx_bytes = hex_to_bytes(rx_hex)
    if len(rx_bytes) < 11:
        return None
    data_bytes = rx_bytes[9:-2]
    if len(data_bytes) <= 0:
        return None
    data_bytes = data_bytes[0::2]
    return decode_value_bytes(data_bytes, reg_type)

txt_file = "RAWFILE.txt"

df = pd.read_csv(
    txt_file,
    sep=r"\|\|\;\;",
    header=None,
    engine="python"
)

df.columns = [
    "Reg Name", "Time1", "Transmit", "Transmit Hex",
    "Time2", "Respond", "Respond Hex",
    "Time3", "Error", "Delta TX", "Delta RX"
]
df["Value"] = None
for index, row in df.iterrows():
    try:
        value = decode(row["Reg Name"], row["Respond Hex"])
        df.at[index, "Value"] = value
    except Exception as e:
        print(f"Fel vid rad {index} ({row['Reg Name']}): {e}")
        df.at[index, "Value"] = None

file_name = "read_log_decoded.xlsx"
df.to_excel(file_name, index=False)
wb = load_workbook(file_name)
ws = wb.active
ws.freeze_panes = "A2"

for col in ws.columns:
    max_length = 0
    col_letter = col[0].column_letter
    for cell in col:
        if cell.value:
            max_length = max(max_length, len(str(cell.value)))
    ws.column_dimensions[col_letter].width = max_length + 2

wb.save(file_name)
```

Listing A.2: Python code for parsing

A.2 Graphs

This section includes graphs of tests performed on MR 2 and MR 3

A.2.1 MR 2 Test 1

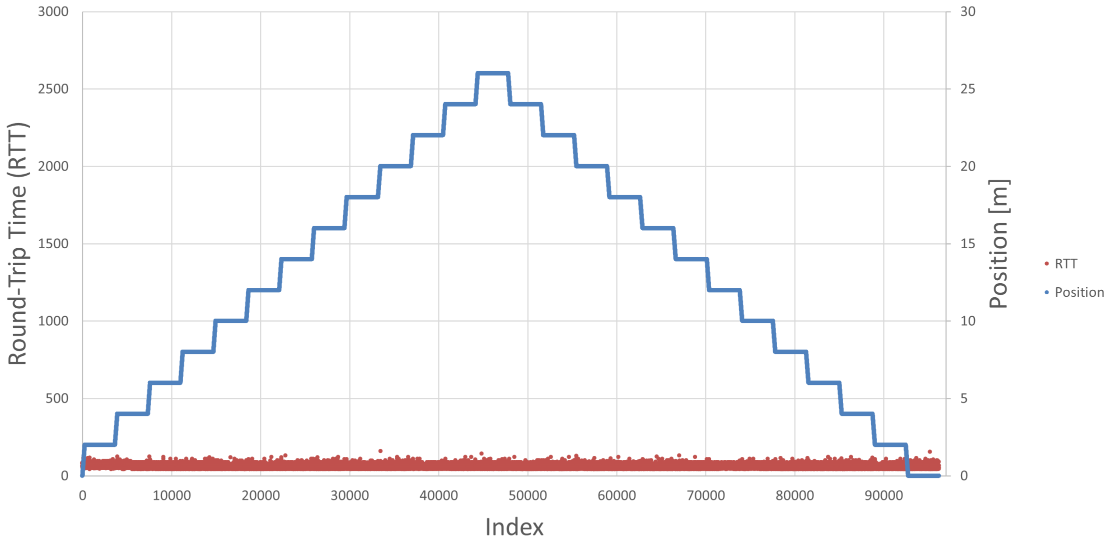
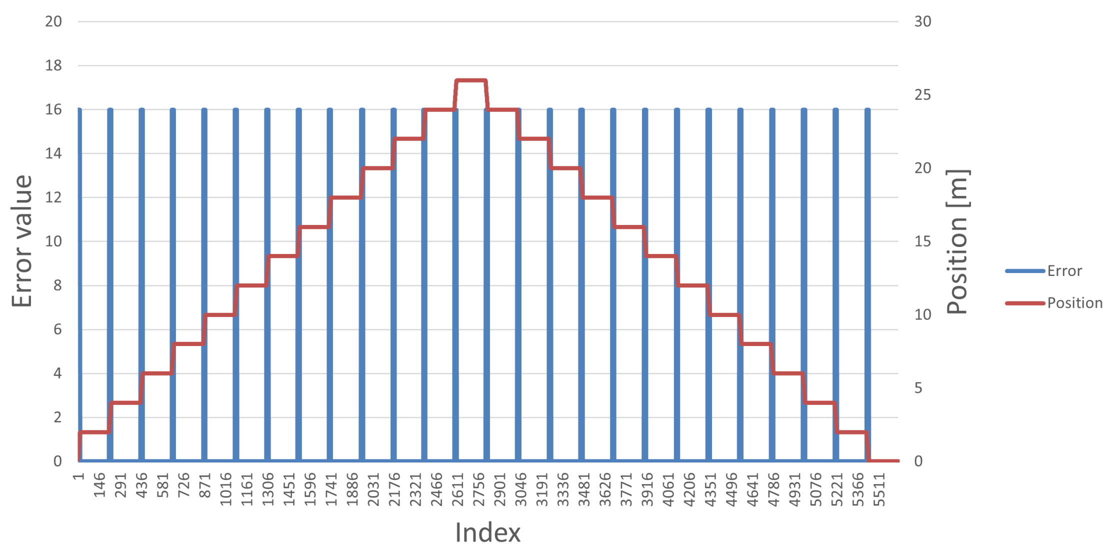
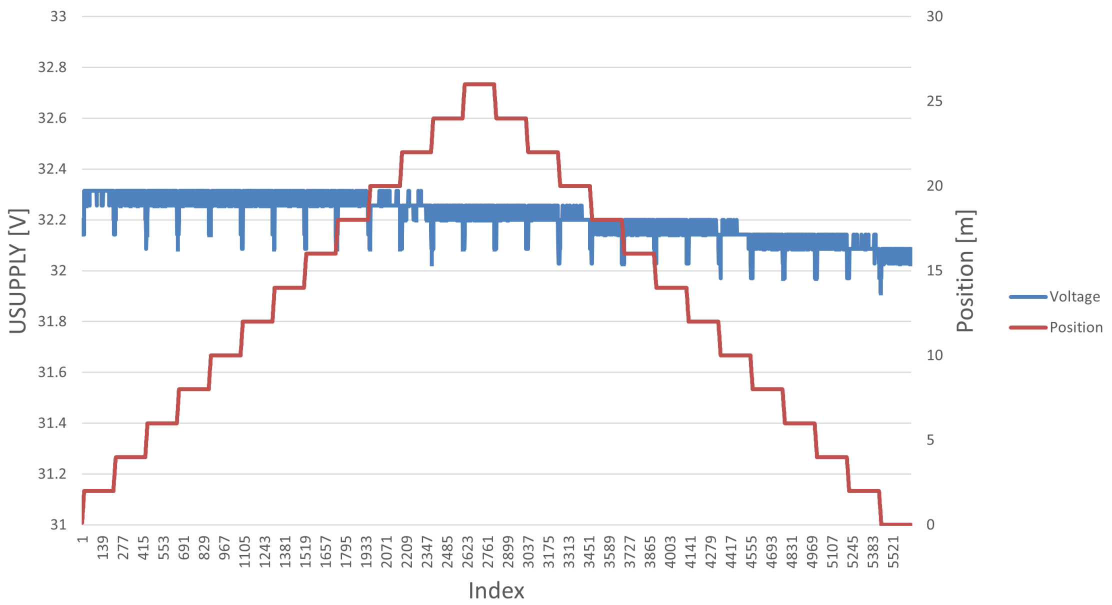


Figure A.1: Graph of Robot 2 test 1 illustrating RTT and virtual position.



(a) `ERR_STAT` value.



(b) `U_SUPPLY` interpreted as voltage.

Figure A.2: `ERR_STAT` and `U_SUPPLY` values from test 1.

A.2.2 MR 2 Test 2

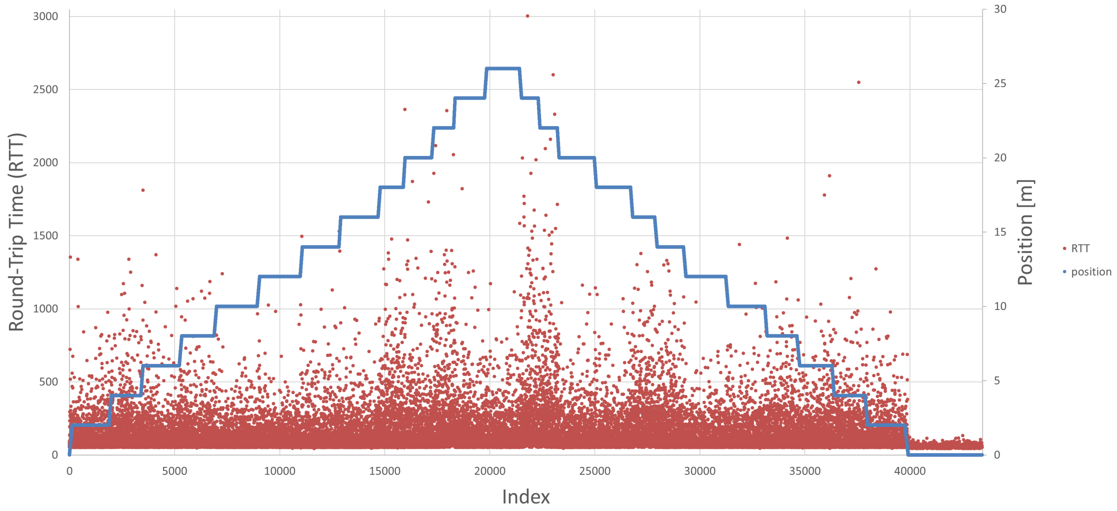
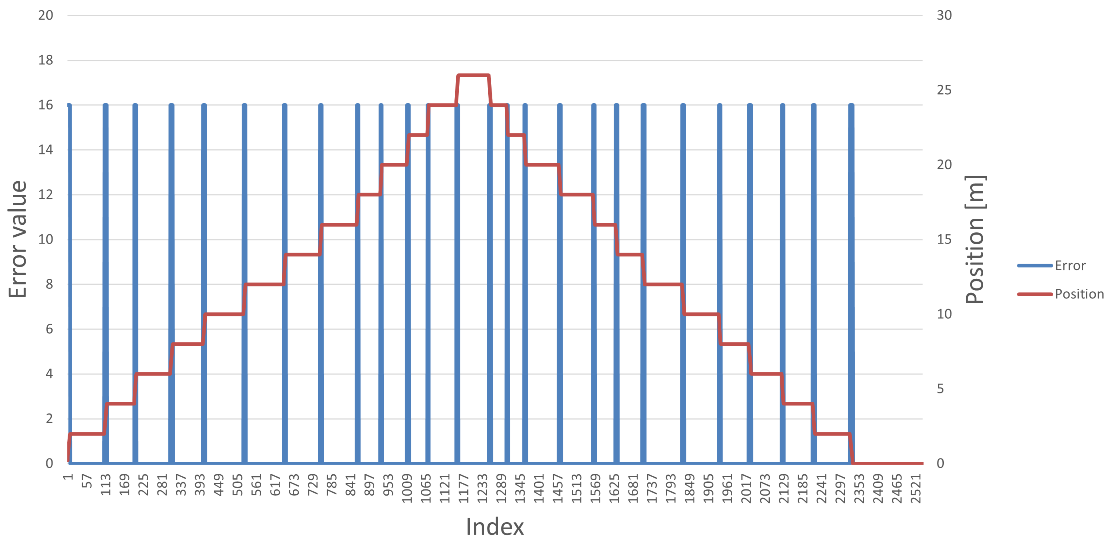
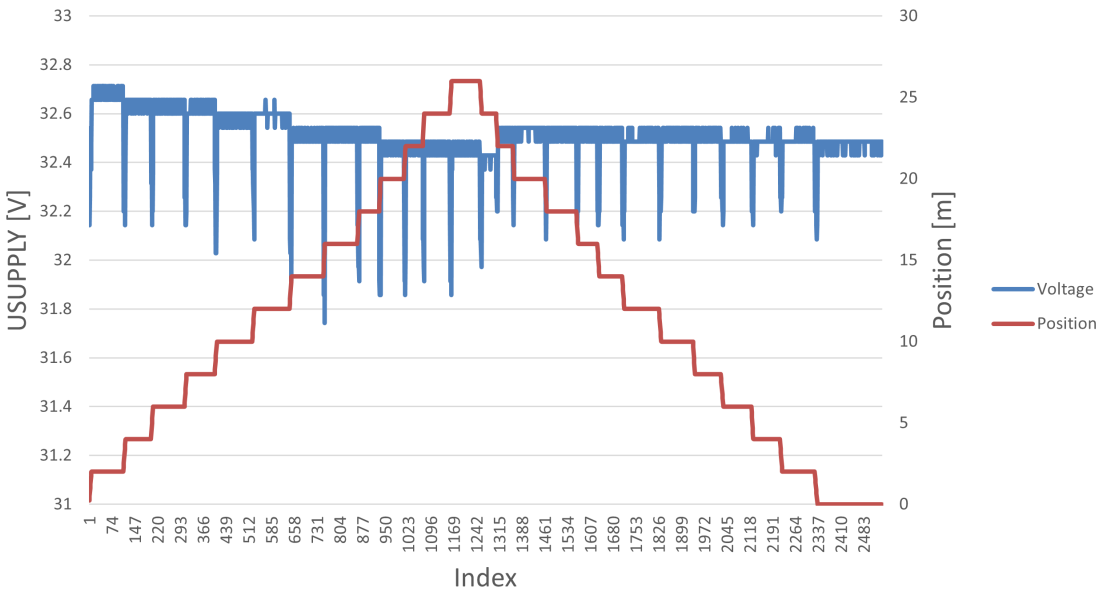


Figure A.3: Graph of Robot 2 test 2 illustrating RTT and position.



(a) `ERR_STAT` value.



(b) `U_SUPPLY` interpreted as voltage.

Figure A.4: `ERR_STAT` and `U_SUPPLY` values from test 2.

A.2.3 MR 2 Test 3

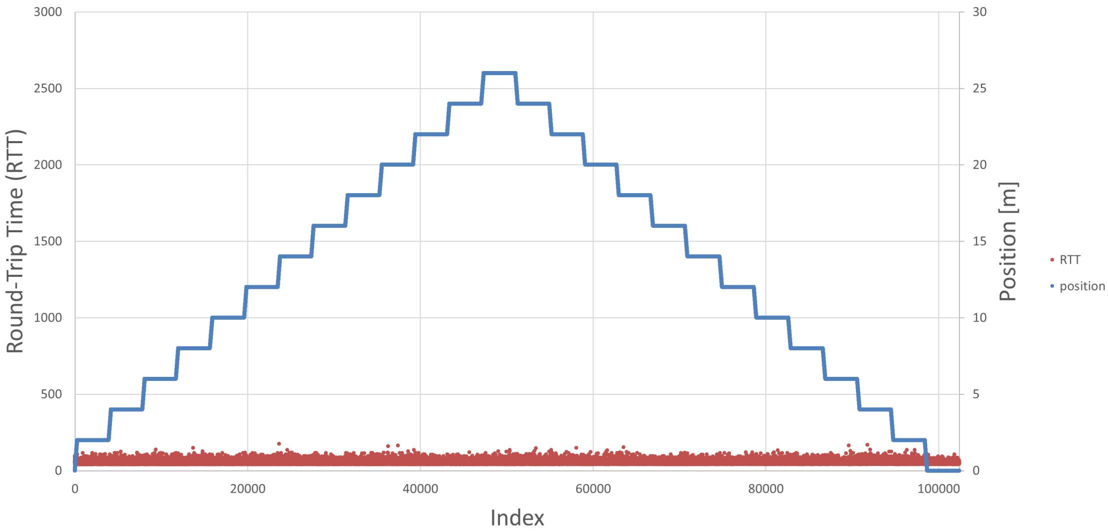
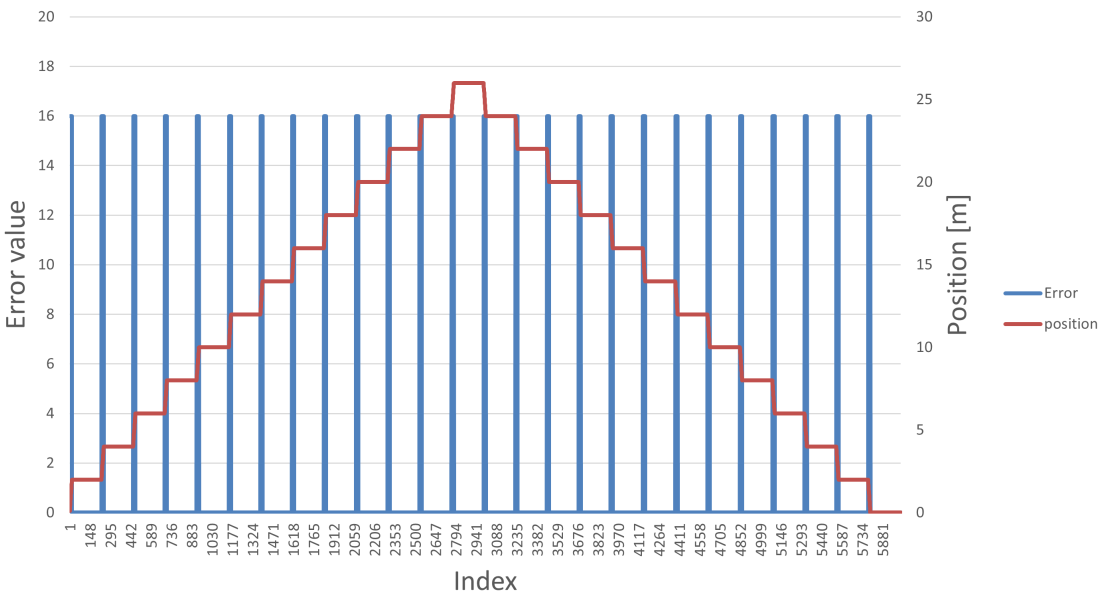
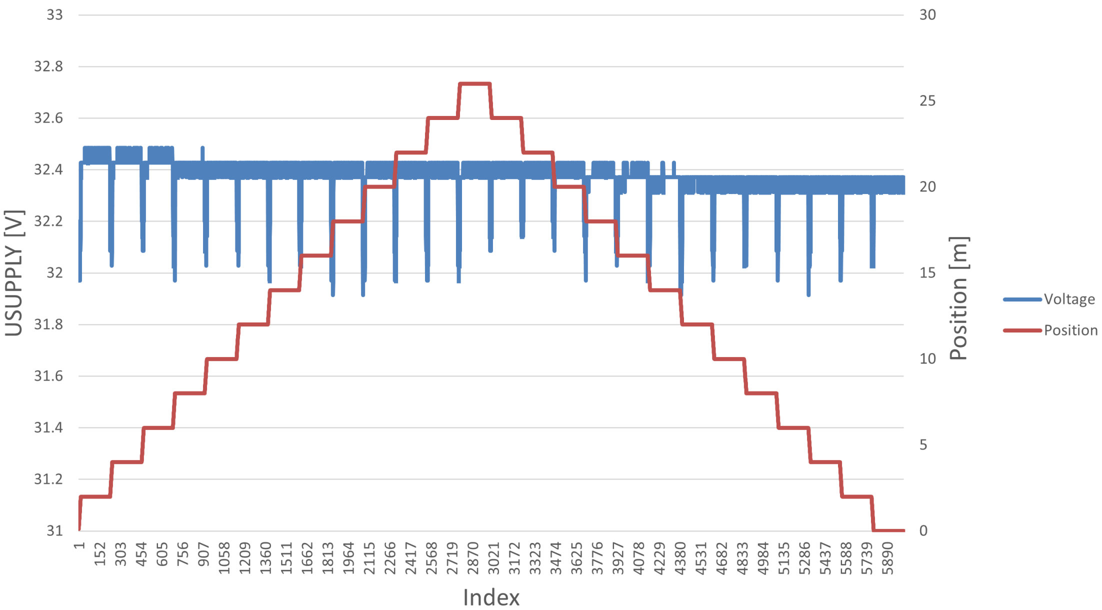


Figure A.5: Graph of Robot 2 test 3 illustrating RTT and position.



(a) `ERR_STAT` value.



(b) `U_SUPPLY` interpreted as voltage.

Figure A.6: `ERR_STAT` and `U_SUPPLY` values from test 3.

A.2.4 MR 3 Test 1

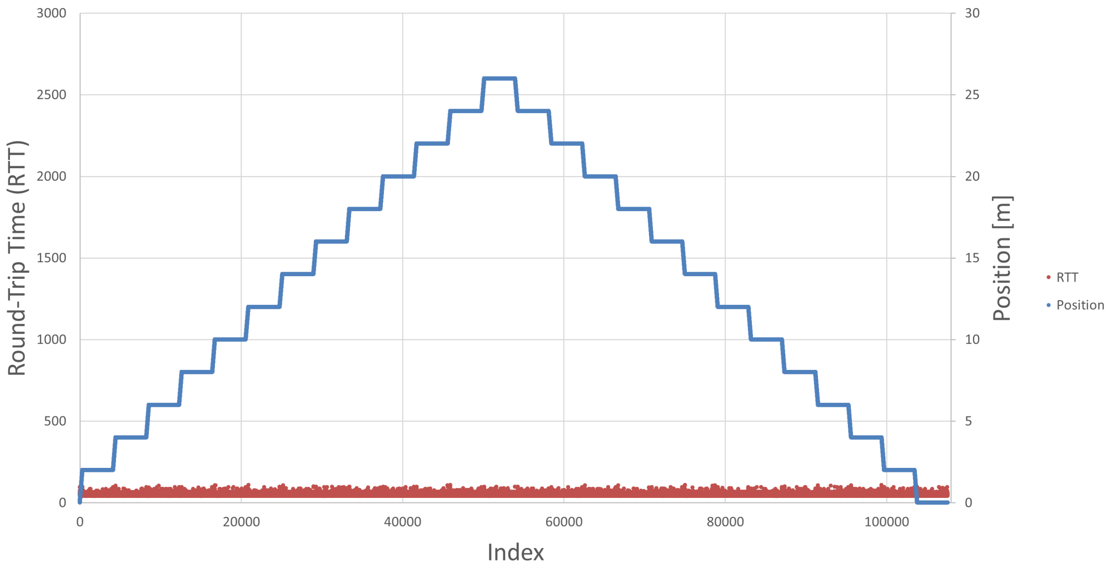
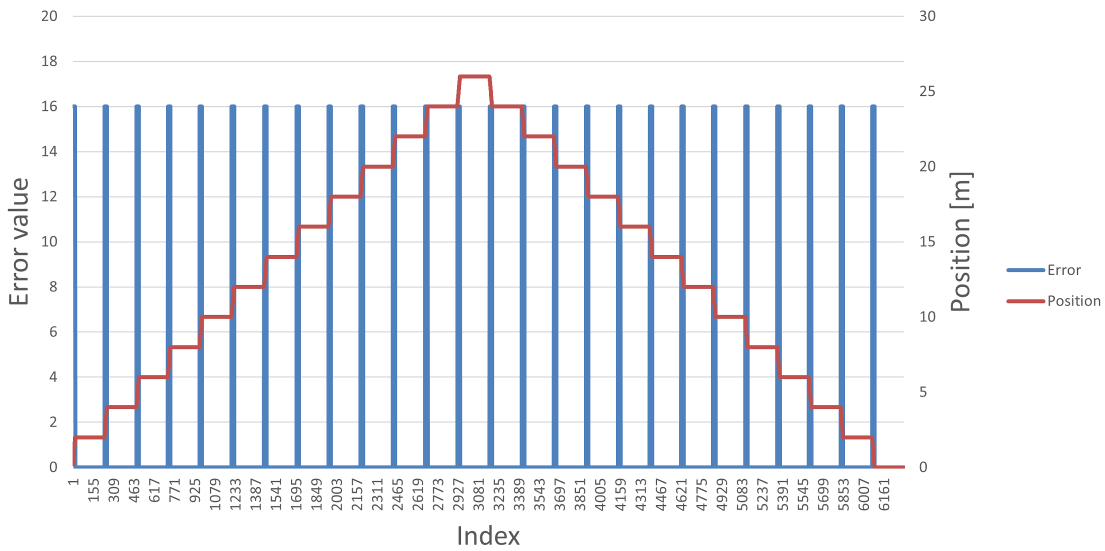
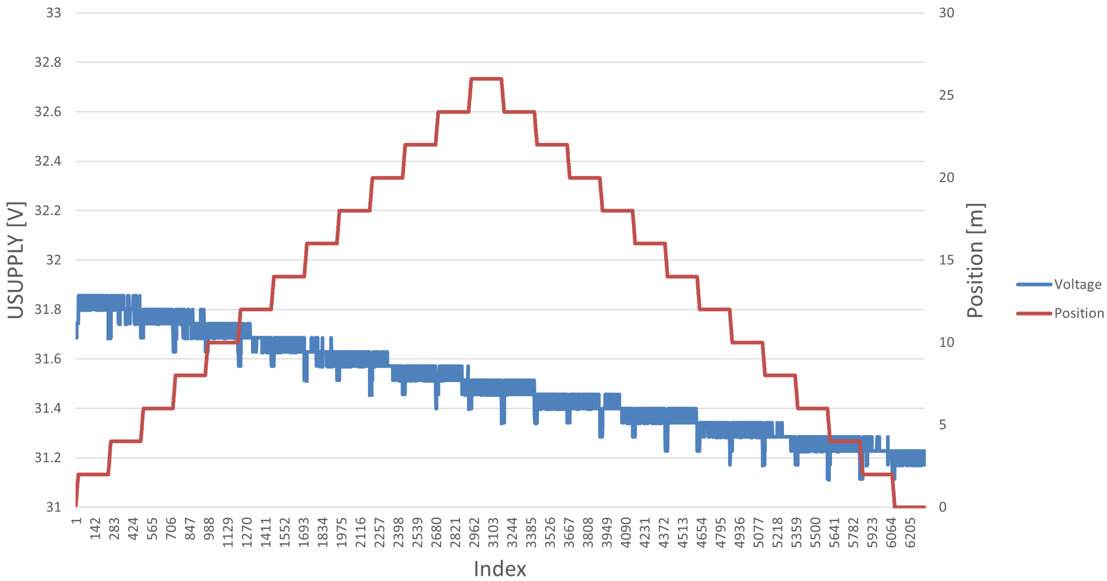


Figure A.7: Graph of Robot 3 test 1 illustrating RTT and virtual position.



(a) `ERR_STAT` value.



(b) `U_SUPPLY` interpreted as voltage.

Figure A.8: `ERR_STAT` and `U_SUPPLY` values from test 1.

A.2.5 MR 3 Test 3

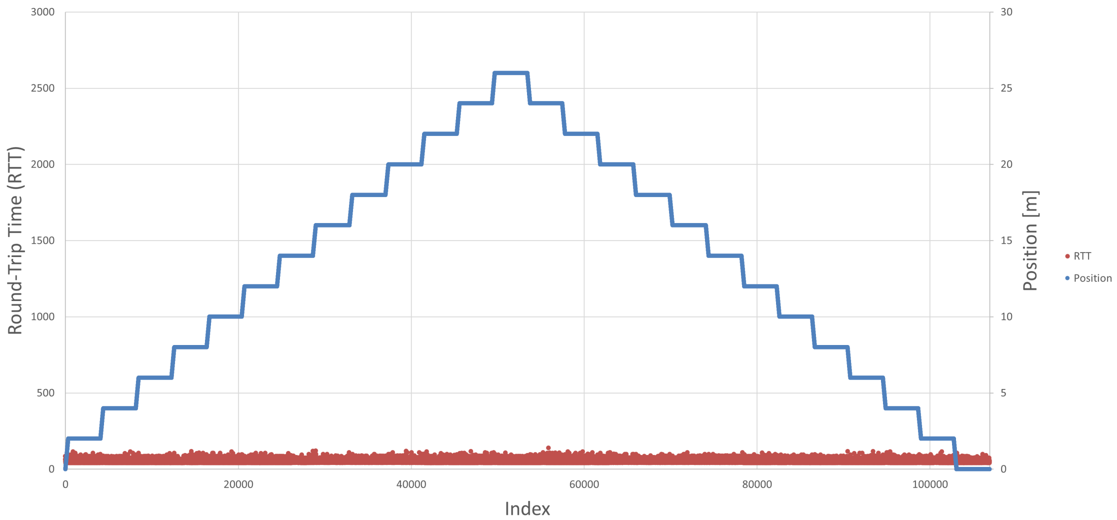
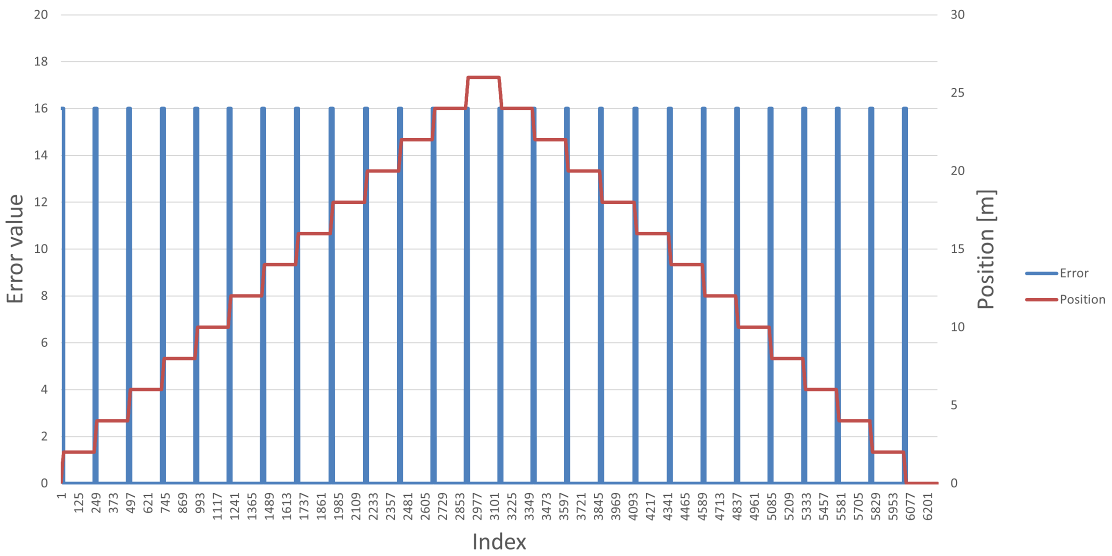
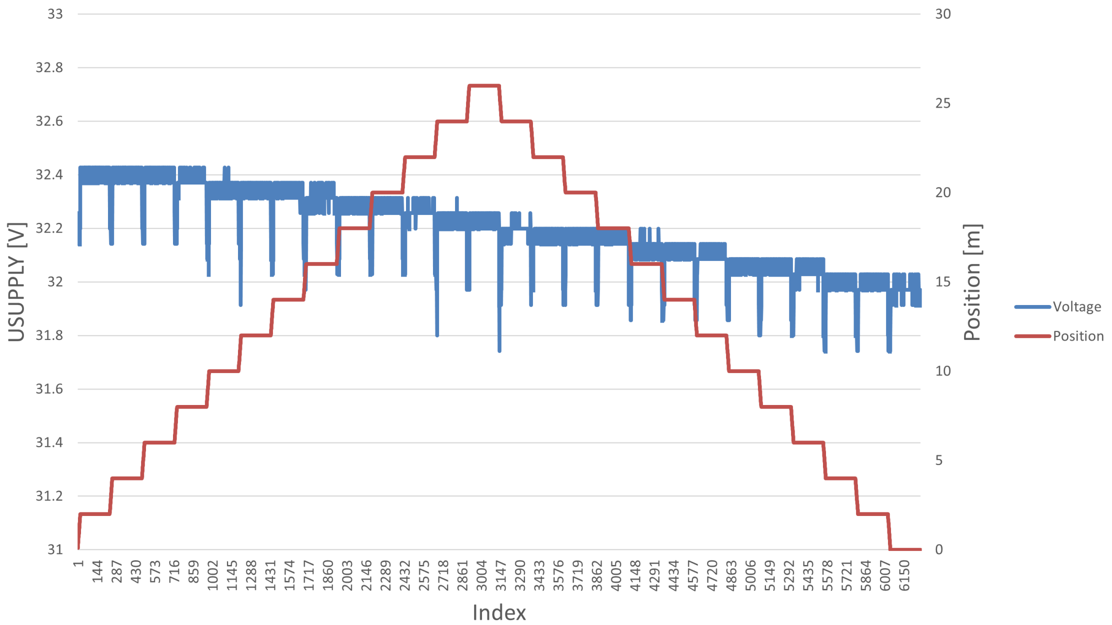


Figure A.9: Graph of Robot 3 test 3 illustrating RTT and position.



(a) `ERR_STAT` value.



(b) `U_SUPPLY` interpreted as voltage.

Figure A.10: `ERR_STAT` and `U_SUPPLY` values from test 3.

DEPARTMENT OF ELECTRICAL ENGINEERING
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se

CHALMERS
UNIVERSITY OF TECHNOLOGY