EE3L11

Motion Reference Unit Testing Platform

Hardware

Bachelor Thesis

in partial fulfillment of the requirements for the degree of
Bachelor of Science in Electrical Engineering

at Delft University of Technology
Faculty of Electrical Engineering, Mathematics, and Computer Science

by
Group B:
Matti Dreef, 4456122
Remon van Duijnen, 4489683
July 23, 2018

Supervisors:
dr. ir. Rob Remis - TU Delft
dr. ir. Frank Nieuwenhuizen - Ampelmann Operations B.V.

Delft
University of
Technology

Challenge the future

Abstract

Commissioned by Ampelmann Operations B.V., a testing system for Motion Reference Units has been developed. This testing system provides a platform to compare different motion reference units to be able to verify their capabilities. The design of this system includes validation of a mechanical setup, as well as the development of an electrical controller to drive the mechanical setup.

Preface

This thesis is written as part of the graduation project of the bachelor Electrical Engineering at Delft University of Technology. For this project, a cooperation with Ampelmann Operations B.V. (“Ampelmann”) was set up to provide the project to be worked on, as well as the necessary assistance and supervision.

Ampelmann is a company developing offshore access systems for people and cargo. Due to sea motion, transfer from ships to offshore structures and back is usually a difficult and sometimes dangerous task. Ampelmann has developed a motion-compensating platform to make this process easier and safer.

The project is divided into three parts: hardware, software and MRU assessment. A separate thesis is written for each of these parts. This thesis will focus on the hardware components, which encompasses the test rail, control system, and hardware interface.

We would like to thank Frank Nieuwenhuizen for being our project coordinator at Ampelmann, and Rob Remis as coordinator from TU Delft. We would also like to thank Alexander Verweij, Suzanne Weller, and Niels van der Geld who provided guidance for the three parts of the project. Further thanks go to the other Ampelmann employees who kindly answered our questions. Finally, we would like to thank our families in supporting us through this final stage of our bachelor program.

Contents

1 Introduction

To compensate for the motion of a ship, first the motion needs to be measured. This is done by Motion Reference Units (MRUs). MRUs measure six Degrees of Freedom (6DoF): 3 translational (surge, heave and sway) and 3 rotational (yaw, pitch and roll). These motions are shown in Figure 1.

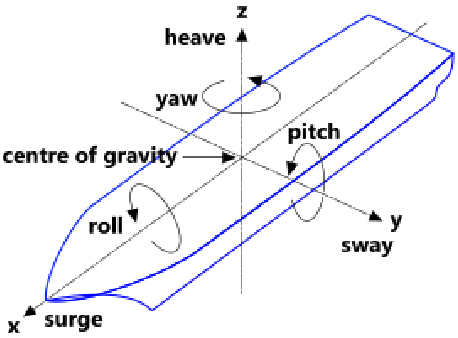


Figure 1: The six degrees of freedom of a ship on sea. [1]

MRU output values are used by the Ampelmann system, which compensates the ship's motion to keep a transfer platform stable with regard to the offshore structure. An example of such a system is shown in Figure 2.

Figure 2: Ampelmann system

It is useful for Ampelmann to test the MRU in conditions similar to the operating conditions of the Ampelmann system. Based on Ampelmann's requirements and the test results, the best suited MRU can be selected. The aim of this project is to quantify the performance of different MRUs. This will be done using a test setup on which an MRU can be moved over a 1DoF rail as shown in Figure 3. This photo is also included in Appendix D which additionally includes labels for the relevant components.

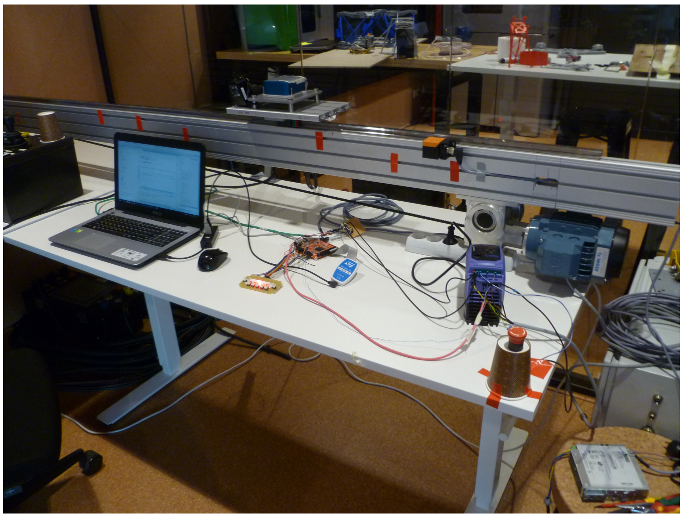


Figure 3: The 1DoF rail and the surrounding hardware.

Figure 4 shows the test system overview. The MRU is mounted on a cart which is moved on the rail using an electric motor. By controlling the movement of the motor, test scenarios, including realistic sea motions, can be simulated. The MRU output data and the position data of the MRU on the rail are collected to be post-processed and analyzed at a later stage.

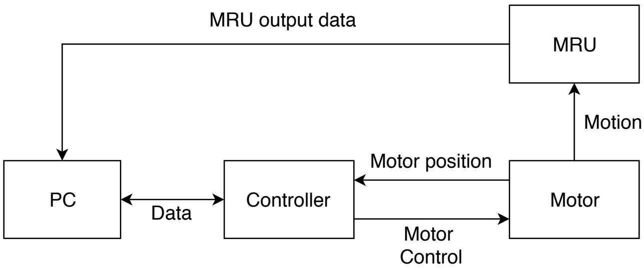


Figure 4: Test system overview

1.1 Hardware

This thesis describes the development of the hardware component of the project. The hardware can further be split into two parts: a mechanical aspect, concerning the test rail itself, and an electrical aspect, which encompasses the control system and its connectivity.

A mechanical setup including a rail was already available. However, an analysis can still be made, exploring other possibilities and comparing these with the existing setup. Furthermore, designing a new mechanical system is out of the scope of this thesis.

For this system, a control unit is needed which connects and controls all components. This control unit will require hardware and software tools to be developed. This thesis is primarily concerned with the hardware, while also some low-level software support has been provided.

1.2 Problem definition

The main purpose of this project is to develop a properly working linear motion test system in order to evaluate different types of MRUs in different wave motion conditions. Nowadays, Ampelmann is using a specific, very expensive MRU, which certainly meets their system requirements. However, at this moment, Ampelmann does not have a system to test the accuracy of other MRU types, which are less expensive and may have great potential of being used in future systems. This is certainly desired in order to make proper design choices in the future without needlessly spending resources. The test system is finished when Ampelmann is able to mount a sensor on the rail, press a start button, and the system will perform all work necessary to give the final test results. These test result should at least be given in a data file and preferably in a visual interface. The main challenge in this project is the time pressure: a useful and safe system should be designed and implemented within a time frame of circa 10 weeks.

The final product will be able verify the specifications of different MRUs so that Ampelmann knows whether an specific brand and type of MRU is up to par with their requirements, thus helping to create a safe system without needlessly spending resources.

1.3 Thesis outline

This thesis describes the project from setting up the requirements up to the end of testing and validating the implementation. The program of requirements is thoroughly discussed in Chapter 2. Chapter 3 describes the designing process in detail. This entails the possible design options as well as the design option chosen after carefully weighing the options. Once the design choices are made, the actual implementation is done. The chosen components and detailed implementation is described in Chapter 4. Finally, the system will have to be tested and validated to ensure it behaves as intended and meets the requirements, which is described in Chapter 5. Once all of the results are examined and the conclusions have been stated in Chapter 6, limitations and potential improvements will be discussed in Chapter 7.

2 Program of requirements

The part of the test system this thesis is concerned with is the hardware, which provides the interface between the software and the physical world. The software should be able to make the MRU move according to a predetermined motion. The hardware should open up the possibilities for software to translate calculation into movement, and provide the proper connections, logic and integrated circuits to fulfill the systems requirements.

In this section, the requirements to which the hardware must comply are given, coming from both Ampelmann and the other subgroups. Additionally, a set of preliminary specifications is given which acts as a guide during the project and allows the subgroups to work somewhat independently.

2.1 Requirements

Listed below are the main requirements, which come directly from Ampelmann and should be followed as closely as possible. These specify exactly what Ampelmann wants the system to be capable of.

R.1 The translational degrees of the MRU must be tested.

R.2 The motion of the platform must be controllable by a host PC which specifies the movement exactly. This specification must be followed within 1 mm by the hardware platform.

R.3 The system should be able to run for at least 8 hours continuously, with 24 hours being preferred. Long tests may consist of repeated shorter periods.

R.4 The most ‘extreme’ motion the platform needs to be able to make is a sinusoidial waveform with a frequency of 0.5 Hz and a peak-to-peak amplitude of 1 m.

R.5 The platform with MRU will have a maximum weight of 5 kg.

R.6 The hardware must be easily usable and expandable for future work by Ampelmann employees.

R.7 The system must be safe for bystanders under all circumstances, and must not cause damage to any components or attached MRUs.

Next, the two other subgroups in this project have requirements as well. From the software subgroup [2]:

R.8 The controller must be able to control the motor speed and read out the encoder and proximity sensor. The motor speed must be controllable from zero to the maximum speed necessary in both directions, and the encoder readout must keep up at this maximum speed.

R.9 The controller must be programmable, preferably in C.

R.10 Documentation and example code for the controller should be available.

And finally, from the MRU assessment subgroup [3]:

R.11 A ground truth of the MRU motion must be provided, accurate to within 1 mm.

R.12 The MRU motion data must be collected on a PC, together with the time the data is recorded.

2.2 Preliminary specifications

The preliminary specifications include current boundary conditions and preferred design choices, but may be influenced by the design and implementation of the system. The following specifications have been set up in the first few weeks of the project to act as an initial guide and allow the three subgroups to work somewhat independently.

First, a mechanical setup is already available, which features the rail, cart, and gearing, a motor [4] and motor drive [5], a rotary encoder [6], and an induction sensor [7]. The cart slides over the rail and is pulled by a belt which is guided by two bearings at both ends. One bearing is powered by the motor through a gearbox, to which also the rotary encoder is attached. The motor is controlled and powered by the motor drive.

P.1 A 1DoF rail will be used, with each translational degree being tested separately. This will require the rail to be oriented vertically to test heave motion.

P.2 The rail has a total length of 230 cm, of which 186 cm is usable due to the positioning of bearings. With 15 cm safety margins at both ends, this finally leaves 156 cm of safe testing space.

P.3 The gearbox has a ratio of 10:1, providing lower speed but higher torque to the cart.

P.4 The motor has a nominal rating of 250 W at 1410 rpm, providing 1.71 Nm of torque.

P.5 The motor drive powers the motor, and is itself controlled by the controller to be developed in this thesis.

P.6 The rotary encoder keeps track of the cart position. It has 10000 positions per rotation, and is attached to the rail side of the gearbox.

P.7 The induction sensor marks a reference point for the cart, located at one end of the rail.

Next, some further specifications result from wishes by Ampelmann and the other subgroups, or from practical limitations.

P.8 A microcontroller is used, which is programmable in C.

P.9 A motor control loop running at 500 Hz will be used to achieve the required accuracy.

P.10 The MRU, ground truth, and control loop setpoint data will be sampled at 50 Hz.

P.11 The use of UDP/Ethernet communication is preferred.

P.12 A simple hardware interface is present which allows controlling basic functionality.

P.13 Three-phase power is not used, as it is not available.

P.14 An emergency stop button is present which will quickly and reliably bring the system into a safe state.

3 Design

3.1 Mechanical Design

The existing mechanical setup is to be used as much as possible, and the components have been chosen by Ampelmann based on their needs. However, it is still necessary to validate that they meet the exact requirements given for this project and are compatible with the electrical design.

3.1.1 Motor

An ABB motor (3GVA072001-CSC) [4] was available at Ampelmann. To validate whether this motor will be sufficient, its operating conditions must be calculated from the requirements. To calculate the required values, first the following variables are defined:

A	Sinusoidal motion amplitude
f	Sinusoidal motion frequency
m	Mass of cart
r	Radius of the belt-driving wheel
η	Gear ratio of motor to driving wheel
θ	$= 2\pi f t$, motor angle

Table 3.1: Variable definitions

For a sinusoidal motion, the position x, velocity v, and acceleration a of the cart can be expressed as:

$$\begin{aligned} x &= A\sin\theta, \\ v &= 2\pi f A\cos\theta, \\ a &= -4\pi^2 f^2 A\sin\theta. \end{aligned}$$

And the instantaneous mechanical power of the cart can be calculated with

$$P = F\cdot v = m\cdot a\cdot v = -4\pi^3 f^3 A^2 m\sin 2\theta,$$

resulting in $P_{max} = 4\pi^3 f^3 A^2 m$. Note that this maximum power has to be delivered to the cart when the power is positive, and absorbed back into the motor when the power is negative. Per requirements **R.4** and **R.5**, the cart with a weight of $m = 5\,\mathrm{kg}$ should be able to make a movement with $f = 0.5\,\mathrm{Hz}$ and $A = 0.5\,\mathrm{m}$. Substituting these values gives

$$P_{max} = 19\,\mathrm{W}.$$

This power is calculated assuming the cart moves horizontally, without influence of gravity. When gravity is taken into account, an additional term $P_{grav} = m\cdot g\cdot v$ has to be added. Using the above expression for v and necessary substitutions, the maximum additional power with $g = 9.81\,\mathrm{m/s}$ is

$$P_{grav,max} = 2\pi f A g m = 77\,\mathrm{W}.$$

Thus, the total maximum power required with a vertical rail is

$$P_{vert,max} = P_{max} + P_{grav,max} = 96\,\mathrm{W}.$$

This is the minimum required power, as it does not take friction into account. Additionally, the motor itself has winding losses, which together with friction result in a net power consumption, converting this power into heat. Thus, when the cart accelerates electrical power is provided and turned into mechanical power and heat, and during deceleration the mechanical power is converted back into electrical power and more heat. Even when moving with a constant speed power is required to counter the friction and winding losses. The actual power consumption can be measured once the motor is in place.

Next, the maximum rotational speed required must not exceed the allowable speed of the motor. This can be calculated as

$$\omega_{rpm,max} = 60 \cdot \frac{fA}{r} \cdot \eta,$$

which gives 2143 rpm using the previously given values and taking $r = 7\,\mathrm{cm}$ as measured in the current setup, well below the maximum 6000 rpm of the motor.

As the driving frequency and motor speed change, the torque available at the output changes according to Figure 3 in [4]. Nominally, the motor delivers 250 W at 1410 rpm when driven by a 50 Hz input. At the maximum speed required the input will be at $2143/1410 \times 50 = 76\,\mathrm{Hz}$, at which point the recommended torque has reduced to 58% of the nominal torque of 1.71 Nm. Thus, the power available at this speed is

$$\frac{2143}{60}\,[\mathrm{Hz}] \;\times 2\pi\,[\mathrm{rad}] \;\times 1.71 \cdot 58\%\,[\mathrm{Nm}] \;= 223\,\mathrm{W}.$$

Assuming the available power stays between 223 and 250 W, in horizontal orientation 203 to 230 W of headroom is available over the required 20 W. In vertical orientation, 127 to 156 W remains. Some of this power will be required to counter friction losses in the gearbox and cart.

Finally, the motor torque required can be calculated. Starting from the maximum force on the cart:

$$F_{max} = m \cdot a_{max} = 4\pi^2 f^2 Am = 25\,\mathrm{N}.$$

Multiplying this by the radius of the driving gear and the gearbox ratio of 0.1 gives a motor torque requirement of

$$\tau_{max} = F_{max} \cdot r \cdot \eta_{gearbox} = 0.17\,\mathrm{Nm},$$

which is well under the nominal motor torque of 1.71 Nm.

3.1.2 Motor drive

The motor has to be driven by a high-voltage driver capable of handling the required power. Either an existing drive or own design can be used. While designing and building an own design is a challenge in itself, it is considered out of the scope of this project. Furthermore, a motor drive is already available in the current setup: an Invertek Optidrive E2 series motor drive [5]. The drive has a variety of control modes, including an analog input mode and a serial connection mode. It also offers plenty of safety functionality, as will be discussed in Section 3.3. The drive does not need three-phase power input and is capable of delivering 750 W of continuous power, way over the nominal 250 W of the motor. Thus, a lower-rated motor drive could have been used, but as the current one is already available it is deemed satisfactory for this purpose. Furthermore, if a motor upgrade is necessary in the future, the current drive might be still usable.

The motor drive has a multitude of parameters, allowing it to be configured for different motors and applications. Parameters include motor ratings, the method by which the motor drive is to be controlled, and self-contained parameters which allow the drive to function without external controller. Additionally, the drive is agnostic about whether the motor is connected in a star or delta topology; it can easily be configured to work with either.

Controlling the motor drive can be done using several methods, but should fit requirement **R.8**. Besides allowing configuration using an on-board interface, the motor speed can be set using analog control voltages, as well as with the Modbus RTU serial interface [8]. Both can originate from an external controller, which will be designed and implemented later in this thesis.

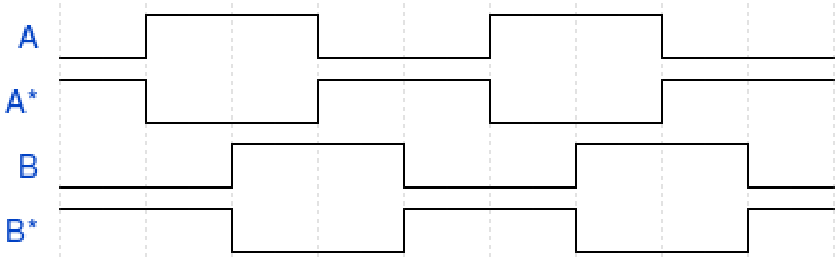


Figure 5: Differential quadrature signal

The Modbus interface allows reading and writing settings other than motor speed, thus could be used to for example display the motor power in real-time to the user. With these possibilities in mind, the Modbus interface is the preferred method of controlling the motor drive, but the analog method will fulfill the requirements as well. Besides controlling motor speed, the controller should also be able to simply turn the motor on and off, and it should be notified of drive faults such as overcurrent or power loss.

3.1.3 Rotary encoder

Ground truth motion data needs to be available per requirement **R.11**, as well as some sort of feedback for the motor control loop. This loop compares the input to the motor with what is actually happening, making it possible to correct for unpredictable errors caused by e.g. friction or not fully linear motor speed. Both can be provided by a rotary encoder which keeps track of the amount of rotation of its axis. A Tamagawa H48 series rotary encoder [6] is available for use, and is connected to the belt-driving gear *after* the gearbox. This rotary encoder has a differential quadrature output with 10000 signal transitions per rotation.

Figure 5 shows the signals that make up the encoder output. Signals A and B are 90 degrees out of phase with respect of each other, thus in quadrature. Which signal is leading determines the direction in which the encoder is rotating, and the amount of pulses corresponds to the amount of rotation. To provide a method of countering interference on the signal, both signals are provided as a differential pair by including the inverted signals A* and B*. Interference affects both signals, thus by subtracting A* from A the interference cancels while the signal is amplified.

Instead of converting the differential signals to single-ended signals for further processing, the positive signals A and B can be used for simplicity. This removes the need to design the required circuitry and allows prototyping to start sooner. If the resulting encoder data appears to be affected by interference, a differential amplifier can be implemented to use the available signals and counter the interference effects. As reported in Section 5.2.2, this was indeed necessary, and the differential amplifier implementation is discussed in Section 4.2.2.

The belt is driven by a gear with a measured radius of approximately 7 cm. The linear position of the cart can thus be measured in steps of

$$\frac{2\pi \times 0.07}{10^4} = 44\,\mu\mathrm{m},$$

well below the required 1 mm accuracy. It is thus expected that the largest measurement error will originate from the mechanical system, especially the belt.

Furthermore, the usable part of the rail is covered by a total of $1.9/44\mu = 43.2$ thousand encoder positions. This brings up a requirement for the software responsible for reading and handling this data: a 16-bit integer has a range from 0 to 65535 unsigned or -32768 to 32767 signed, thus using a signed 16-bit integer with the zero position at one end of the rail will result in an overflow.

The maximum required cart speed is at the zero-crossing of a 0.5 Hz, 0.5 m amplitude sine wave, where the velocity is $2\pi \times 0.5 \times 0.5 = 1.57\,\mathrm{m/s}$. At this speed, both the rotary encoder and its decoder on the controller side will have to keep up at $1.57/44\mu = 35.7$ thousand positions per second.

3.1.4 Drivetrain

Given that the encoder provides a high resolution of the cart position, other causes of error are to be investigated. The gearbox the motor is attached to has a small dead zone when switching directions, in which the gears move freely from each other for a short while. While this reduces the accuracy of the driving motion, this error does not influence the MRU-encoder comparison as the encoder is located *after* the gearbox. Thus, of main interest is that which is located *between* the encoder and the MRU cart: the belt.

The belt is driven by a gear on which it hooks with sturdy teeth, providing a non-slipping driving force. However, there will always be a length of belt between the driving gear and the cart, and this piece of belt can stretch. Unfortunately, no accurate measurement has been made, but pulling on the belt with a significant force in excess of 15 kgf resulted in an extension of approximately 5 mm over a 150 cm portion of the belt. The maximum force exerted on the cart can be calculated using Section 3.1.1 to be 25 N. Assuming the belt stretches linearly, this results in a maximum deviation of

$$\frac{25}{15 \cdot 9.81} \times 5 = 0.85\,\mathrm{mm},$$

which in combination with the encoder provides an accuracy of 1 mm as required by **R.11**.

3.2 Controller

To provide a platform for the software, some sort of controller is required. This controller will have to meet several strict requirements, as well as provide some general additional functionality.

3.2.1 Requirements

Requirements specific to the controller are listed here, referring back to the corresponding original requirements.

- Following **R.12**, data has to be logged and transferred to a PC. This is required to analyze the data afterwards, and may also be useful for debugging.
- Following **R.8**, the controller should run at a sufficiently high frequency. This is especially important for the motor control loop which will have a real-time priority at 500 Hz, while at the same time the controller has to send and receive data at 50 Hz.
- As an UDP/Ethernet connection is preferred (**P.11**). Dedicated hardware for this is necessary to handle the physical interface and medium access control layers [9].
- The available rotary encoder has an incremental output, but an absolute position is needed for both **R.8** and **R.11**. As the position can change at a frequency of up to 35.7 kHz, a software readout might not be practical, thus dedicated hardware might be necessary.

3.2.2 Implementations

Possible implementations of the controller could be a Micro Controller Unit (MCU), or a Field Programmable Gate Array (FPGA). Each of these implementations come with their respective advantages and disadvantages [10] [11].

MCU	**FPGA**
• Less expensive	• More expensive
• Easily programmable	• Higher performance
• More on-board I/O functionality	• High flexibility

A final possibility is a combination of MCU and FPGA. This can be fully integrated, such as found in the Xilinx Zynq SoCs [12], or using separate packages. FPGAs were ruled out because of their complex programming and debugging methods, and the extremely high performance is most likely overkill for the system. Additionally, as the controller should preferably be programmable in C per **R.9**, a microcontroller is deemed the most suitable option, in particular one from the STM32 series [13]. These microcontrollers have plenty of on-board functionality and are easily programmable, with a plethora of examples available on the Internet.

Choosing a specific microcontroller is more difficult, as differences between the many options can be small. To decrease the chance of finding out that the microcontroller chosen here is insufficient later in the implementation process, an over-proportioned option is desirable. Additionally, for some microcontrollers development boards are available which feature many included peripherals. For these development boards the peripherals and availability of documentation and examples determines which is most suitable for this project.

In terms of peripherals, the hardware should be able to connect all parts of the system with their respective in- and outputs. Most important is the connection to a host PC which is responsible for generating wave data and collecting and processing the results. Three options for connecting the system to a PC are: Universal Serial Bus (USB), Ethernet, and a serial interface.

USB is a very widely known and used technology. The USB protocol can be used for a large amount of functionality, including programming of MCUs and FPGAs.

Ethernet is another widely known protocol. Integrated circuits (ICs) for Ethernet communication are available and multiple connections can be made easily if desired.

Serial can be used as a last resort. While both USB and Ethernet are also serial communication busses, there is a scala of other serial interfaces, such as RS-232 and RS-422. These interfaces can be used through any arbitrary connector.

Since an Ethernet connection is desired by Ampelmann, it should at least be possible to explore this option, thus on-board Ethernet functionality should be present on the development board. Many STM32 microcontrollers feature an Ethernet controller with a (Reduced) Media-Independent Interface (RMII/MII) [14] to an external device handling the physical interface (PHY) [15]. This PHY should be present on the development board or be easily attachable using expansion headers.

Two development boards have been selected based on the established criteria: the ST Nucleo-F767ZI [16] and the Olimex STM32-E407 [17], sporting the STM32F767ZI and STM32F407ZGT6 microcontrollers, respectively. The microcontrollers run at 216 MHz and 168 MHz respectively, providing the performance necessary to run the motor control loop and additional functionality (**P.9**). Both boards had already been used by group members, are well-documented, and have a multitude of examples available (**R.10**). Both microcontrollers have a built-in quadrature decoder (**P.6**), QSPI and other serial interfaces, and Ethernet controller which is effectuated by PHYs on the boards (**P.11**).

3.2.3 Hardware control interface

While a User Interface (UI) could be implemented purely in software on a PC, some direct hardware interfacing might also be desirable. To this end, simple buttons and switches in combination with for example a small display could create a very clear and intuitive interface, where some of the basic functions or parameters of the system can be viewed and changed.

As there is also a UI available on the PC itself, a small screen such as an LCD is not deemed useful enough to implement. A set of buttons and LEDs will however be implemented as this is required by the software group, serving as basic input and output for the program running on the microcontroller.

3.3 Safety

3.3.1 Homing

One possible safety hazard is the cart getting too close to the edge of the rail, possibly colliding into the bearings on either side of the rail. As the system must not be damaged following requirement **R.7**, this should be prevented from happening. To this end, the position of the cart can be checked against safety margins at these boundaries. As the rotary encoder used is incremental and not absolute [18], it is necessary to provide a reference position the system at start-up. This is done by homing; the cart is slowly moved to one edge of the rail, where a inductance-based proximity sensor is located (**P.7**). The IM5097 [7] triggers once it senses the metal cart. The system responds to this signal by stopping the cart, and subsequently using the current position on the rail as its reference point.

As both the rotary encoder and MRU give relative positions, their data can be compared directly. Thus, the homing function is not required to know the exact encoder position; it is only needed to implement the safety margins. These margins need not be checked to millimeter accuracy, thus the exact position at which the proximity sensor triggers is not of great concern. The sensor has been positioned and tested to reliably trigger every time the cart is over it.

3.3.2 Safety margins

Now that the absolute position of the cart is known, safety checks can be done on the position while the cart is moving. A forbidden safety zone can be defined on each side of the rail. Once the cart enters one of these regions due to any internal or external malfunctioning, the system will perform a safe stop. Additionally, when during development the internal checking is not yet implemented, these margins can be used as a guideline for when to manually press the emergency stop button. The size of these regions depends on the speed the cart is travelling at, as the braking distance will increase with the cart's speed, and the cart will have traveled further in the average 200 ms human reaction time [19].

The wave motion the cart follows will never exceed these boundaries, as this is checked by the waveform generator implemented by the MRU assessment group [3]. Thus, the cart can only cross these bounds when control is somehow lost, of which two cases are distinguished. The first possibility is complete loss of control, in which the motor starts spinning at its maximum configured speed. This is expected to happen only in the initial development stages, and the cart will in this stage always start from the center of the rail. As discussed in Section 3.1.1, the maximum speed of the cart is 1.57 m/s, at which it will cover half of the rail (93 cm) in $0.93/1.57 = 0.60\,\mathrm{s}$, well over the 200 ms reaction time.

A second possibility is subtle loss of control or drift, due to for example an incorrectly tuned motor control loop. In this case, the cart should already be moving slowly, again providing more time to press the emergency stop. Safety margins of 15 cm on both sides have been settled upon. As can be seen in Figure 6, this provides over 0.25 seconds of reaction time for a cart which is about to collide with a bearing at one end of the rail, and this may be clearly seen even before the safety bound is crossed. With these margins, a safe usable space of 156 cm remains, satisfying requirement **R.4** of a 1 m peak-to-peak wave while still leaving extra space for even larger motions.

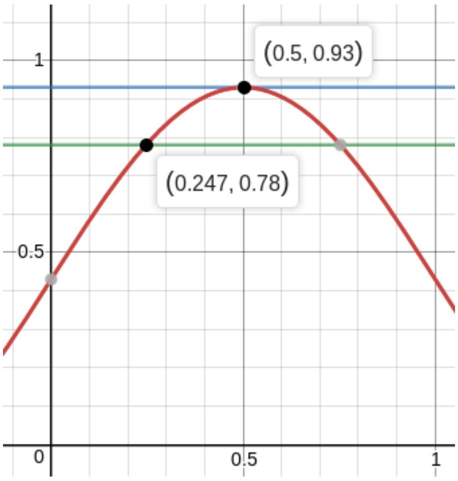


Figure 6: Motion of an out-of-control cart performing a 0.5 m, 0.5 Hz sine wave motion which touches the bearing at one end of the rail.
Red = cart, blue = bearing position, green = 15 cm margin. Y-axis is in meters, X-axis in seconds.

3.3.3 Motor ratings

As the movement the cart is going to make is entirely predetermined, it is possible to calculate the maximum power necessary to drive the cart, as illustrated in Section 3.1.1. If the cart is for example blocked by an obstacle, more power will be necessary. If this unexpected increase in power consumption occurs, the system may be in an unsafe state and should stop immediately. Furthermore, the motor voltage and current ratings should also never be exceeded, which will be handled by the motor drive configured with the parameters in Appendix C.

3.3.4 Loose cables

A common mistake is to forget to connect a cable when setting up the system [20]. This could potentially lead to hazardous situations. Any signal that does not guarantee a safe state once the potential is floating should be brought to a fixed pre-determined potential once the connection is not made properly. This can be done by using pull-up and pull-down resistors [21] on analog signals, and using time-outs on digital signals. When using Modbus control, the motor drive can be configured to perform a ‘Fast Stop’ when no communication has happened over a predetermined amount of time.

3.3.5 General safety design

In addition to these safety measures some self-explanatory safety features should be implemented as well. A prominent emergency button should be present and functioning at all times, bringing the system into a safe state once pressed. The system should also not automatically restart when the button is closed again. A visual and auditory warning that the system is running can also be implemented.

4 Implementation

With the hardware requirements and design specifications known, the system can be implemented. During implementation, it is expected that requirements and specifications can change due to new insights or unforeseen difficulties.

The first implementation considered is a fully custom one, with specialized components and a self-designed PCB. The second implementation is is based around an existing development board, which only slightly limits the choice of controllers and removes the need to design, produce and assemble a custom PCB.

4.1 TAU - Test & Assessment Unit

The fully custom design has been dubbed the Test & Assessment Unit, or TAU for short. This implementation will be designed completely from scratch, which includes picking components and designing a PCB. The primary advantage of this solution is its complete integration and specialization for the final system. All required components can be placed on a single board, without having to waste space on unused components. Care must however be taken to leave some possibilities for future expansion, would this be desired. However, designing a complete and working PCB with all the required components can be a challenge, especially within the given time frame. Additionally, it is not a very sustainable and fool-proof solution: if it breaks, a new board and components have to be ordered and assembled, which costs both time and money.

A start has been made selecting components and connecting them in a schematic design. Available schematics of development boards, as well as datasheets of specific components have been used as a guide. However, due to unforeseen difficulties and time constraints, the TAU approach was eventually abandoned.

4.1.1 Ethernet

Between a microcontroller and an Ethernet cable, an Ethernet controller is necessary to convert the microcontroller's MII/RMII signal to the physical layer that is sent over the cable to the PC [15] [14]. There are off-the-shelf devices that are capable of this conversion, such as the LAN8710A-EZC [22]. This transceiver is capable of running 100 Mbit/s Ethernet following the IEEE 802.3 protocol [9].

4.1.2 Storage memory

Following requirement **R.3**, the system should be able to run a test for a continuous 24 hours. Additional on-board storage memory is necessary if data will not be streamed live over the UDP connection. A test of this length will consist of $24 \times 3600 \times 50 = 4,320,000$ data points at the sampling frequency of 50 Hz (**P.10**). Estimating the data size at 6 bytes for now, this means at least $4,320,000 \times 6 \times 8 = 207\,\mathrm{Mb}$ of storage memory is necessary to pre-load the entire waveform onto the TAU before starting the test. A flash memory chip with a capacity of 256 Mb can be implemented, using QSPI to increase bandwidth [23]. It should be noted that long tests may consist of a shorter, repeating waveform, reducing the storage requirement.

4.1.3 Power

To power the components present on the PCB, power conversion is necessary. The required voltage levels are listed in Table 4.1. A 24 V AC/DC power adapter can be used as a primary voltage source, and can be used for the communication to the motor drive. It is also possible to use the 24 V provided by the motor drive. Using voltage regulators such as the L7805ACD2T-TR [24] and UA78M33CKVURG3 [25], the 5 V and 3.3 V voltage levels can be created, respectively.

Ethernet controller	3.3 V
Microcontroller	3.3 V
Motor drive	24 V
LCD screen	5 V

Table 4.1: Components and their required voltage levels

4.1.4 Top-level

A top-level schematic of the TAU design has been made, and is given in Figure 7. This schematic shows the different functional blocks and how they are connected. How each block is implemented is not further discussed, as work on the TAU has not continued from this point.

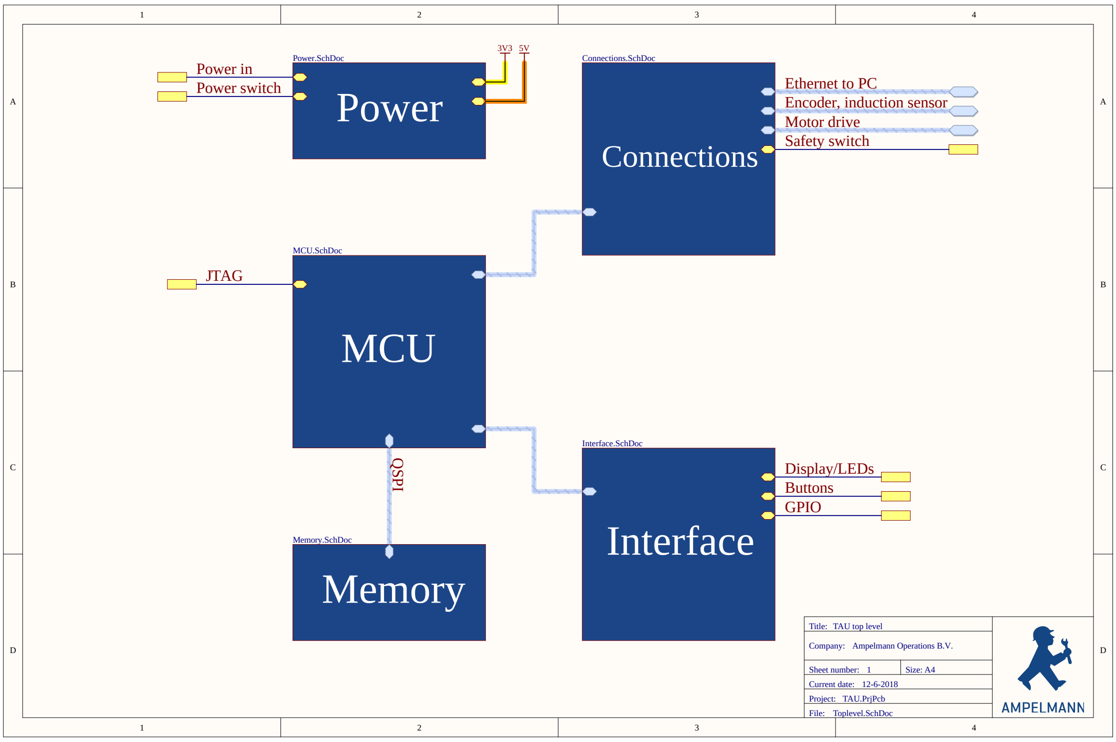


Figure 7: TAU top-level schematic

4.2 PI - Peripheral Interface

Instead of making a new design based on a development board, it is also possible to use a development board with the chosen microcontroller as part of the final system. This might require the use of several extra components, as the development board might not provide all required features. This design has been dubbed the Peripheral Interface, or PI for short.

This approach has the advantage of being easily maintainable and extendable, by replacing and adding components when needed. For most development boards, large amounts of documentation and example code are readily available, making this an even more attractive solution with eye on the Ampelmann requirement of being intuitively usable and easily expandable.

4.2.1 Development board

As stated in Section 3.2, the STM32F407ZGT6 meets the requirements and packs quite some useful peripherals. A development board based on this chip is available from OLIMEX. The OLIMEX STM32 E-407 [17]. This development board also has a built-in Ethernet - PHY transceiver based on the LAN8710 [22]. The peripheral components necessary to complete the system will be described in Section 4.2.2.

4.2.2 Peripheral components

With the PI, any additional components required must be added externally to the development board, which is done on the peripheral extension board. This board implements the hardware interfaces not available directly on the development board, which includes connections to the motor drive, rotary encoder, and inductive homing sensor.

The serial connection on the motor drive utilizes the Modbus RTU protocol, described in Section 4.4.1, over RS-485 [26]. A RS-485 differential line transceiver is used as interface between the PI and the Drive. The OLIMEX MOD RS485 ISO [27] is directly compatible with the OLIMEX development board used during this project. This extension board can be plugged into the UEXT header on the development board. Communication between the development board and the extension board is done over RS-232 [28].

As the digital inputs on the drive function on 24 V and need a minimum of 8 V to trigger as a logic ‘1’, it is not possible to directly connect the PI, with a maximum output voltage of 3.3 V, to the drive. Thus, a level shifter is required to increase the output voltage to 24 V. This is easily achieved using an n-MOSFET and pull-up resistor to a 24 V supply. When the MOSFET is on it will pull the output to ground, and when it is off the resistors pulls the output to 24 V.

However, the same effect can be achieved in many other ways. As a collection of VO617C optocouplers [29] was already available, this has been used to implement the level shifter. The schematic is given in Figure 8. Once the PI outputs a logic ‘1’, thus putting 3.3 V on the ‘Enable 3V3’ port, the LED will light up, allowing the phototransistor to conduct. This means 24 V will now be present at the ‘Enable 24V’ port. A pull-down resistor is present to keep the output signal in a defined state when the input is low. While an optocoupler is generally used to keep two circuits galvanically isolated, this is thus not the case here, however it is also not necessary since the PI and motor drive share a common ground already. Thus, while using a MOSFET could be considered a more elegant solution, the use of an optocoupler is justified by one already being available.

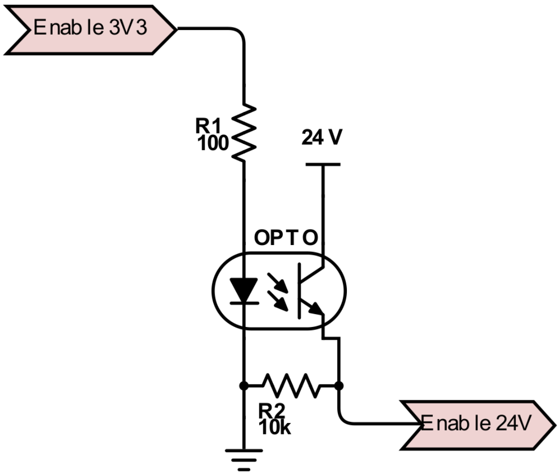


Figure 8: Step-up circuit

As stated in Section 3.2, the microprocessor on the E407 board has built-in quadrature encoder input.

This input, however, is not able to handle a four-line differential quadrature input. As described in Section 5.2.2, this differential input is required, thus a differential amplifier has been designed, as illustrated in Figure 9. This circuit translates the differential input to a single-ended output according to the truth table in Table 4.2. It does so by implementing both an inverting and non-inverting amplifier in the same circuit, effectively subtracting the 'In-' signal from the 'In+' signal as shown in Figure 10. This operation cancels any common-mode signal caused by interference, and passes the difference in the signals to the single-ended output. An LM358 op-amp [30] has been used as it integrates two op-amps in one package and is designed for single-supply operation. It is fed with a single supply voltage of 5 V, effectively clipping the output at this level to prevent the microcontroller from being exposed to higher voltages which could damage it.

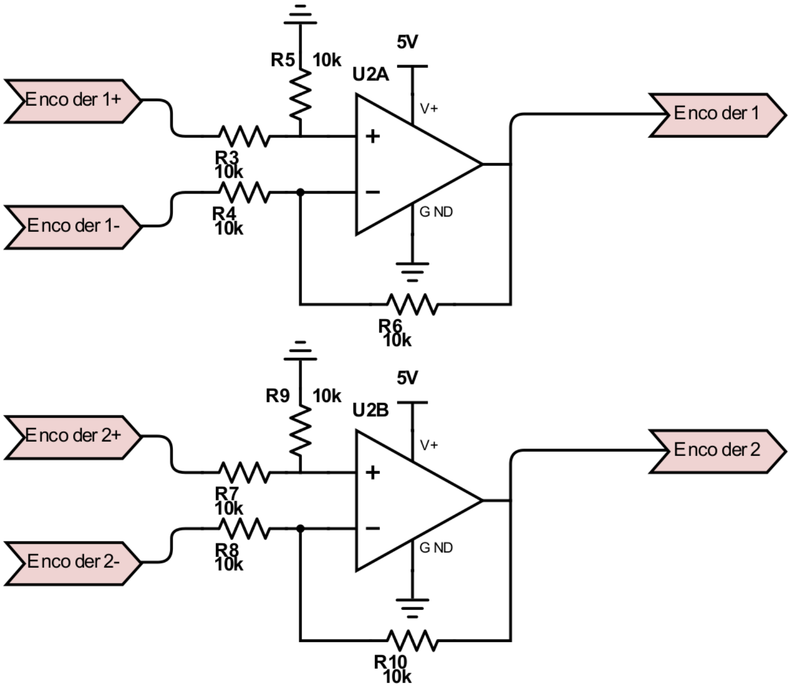


Figure 9: Differential amplifier circuit implementation.

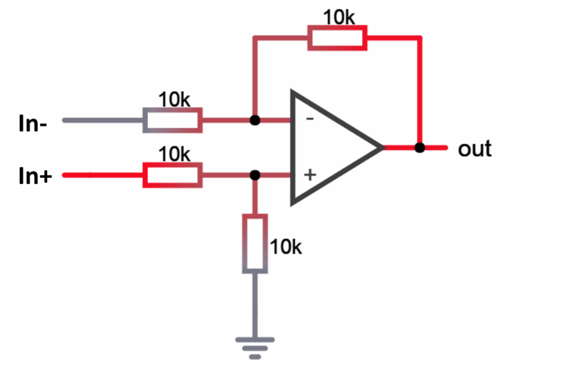


Figure 10: Working of the differential amplifier circuit, with 'In+' high and 'In-' low.

In-	In+	Out
0	0	X
0	1	1
1	0	0
1	1	X

Table 4.2: Truth table of the differential amplifier.

4.3 Hardware control interface

A simple hardware control interface has been implemented, featuring 7 momentary buttons and 5 LEDs. The buttons are normally-closed, with one end connected to a 3.3 V rail. On the microcontroller, the input pins will be configured with pull-down resistors, resulting in a low signal when the button is pressed, disconnecting the input from the 3.3 V rail. The LEDs are also connected to the 3.3 V rail through a 91 Ω current-limiting resistor. The other end is connected to an output of the microcontroller, which has thus be held low to turn the LED on.

4.4 Motor drive control

As discussed in Section 3.1.2, the motor drive can be controlled both using the serial Modbus RTU interface, or using analog and digital control voltages. Additionally, the motor drive has to be configured by setting its parameters to the correct values. A description of all motor drive parameters and the values they were set to are listed in Appendix C. This appendix includes a short motivation for each relevant parameter.

4.4.1 Modbus RTU control

The Modbus RTU protocol is a standard for industrial control systems. The specification uses RS-485 communication over a twisted-pair cable and defines several standard commands to be used [31] [8]. The motor drive implements two of these which allow holding registers to be read and written. The data to be sent and received for these functions is given in Tables 4.3 and 4.4.

Name	**Bytes**	**Value & description**
Master telegram		
Slave address	1	Can be chosen by motor drive settings; set to 18 for this system.
Function code	1	Specifies function; must be 03 for reading.
1st register address	2	The address of the first register in sequence to read.
No. of registers	2	The number of consecutive registers to read; kept at 1 to simplify communication.
CRC checksum	2	A cyclic redundancy check of the previous bytes; this must be calculated for each packet.
Slave response		
Slave address	1	These have the same value as what was send, to allow packet identification.
Starting address	1	
1st register value	2	The value of the first register in sequence.
2nd, 3rd, ... values	2	The values of subsequent registers. As only a single register will be read at a time, these values will not be present.
CRC checksum	2	A cyclic redundancy check of the previous 16-bit words, allowing packet integrity checking.

Table 4.3: Modbus data sequence to read a register.

Name	Bytes	Value & description
Master telegram		
Slave address	1	Can be chosen by motor drive settings; set to 18 for this system.
Function code	1	Specifies function; must be 06 for writing.
Register address	2	The address of the register to write.
Value	2	The value to write to the selected register.
CRC checksum	2	A cyclic redundancy check of the previous bytes; this must be calculated for each packet.
Slave response		
Slave address	1	These have the same value as what was send, to allow packet identification and check operation success.
Function code	1	
Register address	2	
Register value	2	
CRC checksum	2	A cyclic redundancy check of the previous 16-bit words, allowing packet integrity checking.

Table 4.4: Modbus data sequence to write a register

Both the read and write functions require sending 8 bytes and subsequently receiving 8 bytes. Modbus on the motor drive is half-duplex, which means only one side can send at a time, in contrast with full-duplex where both sides can send and receive simultaneously. Furthermore, it is configured to run at 115200 baud, with 8 data bits, 1 stop bit, and no parity. Together with the start bit, each byte thus takes 10 clock cycles to transmit. With a total of 16 bytes to be sent and received, this means the minimum time required for each function is $10 \times 16/115200 = 1.39\,\mathrm{ms}$, corresponding to 720 Hz. Taking into account uncertain delays and allowing further communication that might be necessary, the motor drive update frequency should be much lower. While only real-life tests can confirm the maximum possible frequency, an estimation of 200 Hz seems reasonable, which does not agree with the preliminary specification of 500 Hz from **P.9**.

4.4.2 Analog control

During implementation, setting up the Modbus communication did not proceed as expected. Using a simple oscilloscope it was confirmed that data was being set to the motor drive, but no reply was received. As no digital logic analyzer was available, it could not be checked exactly what data was being sent. Due to time constraints, the alternative method of controlling the motor drive with an analog signal has first been implemented. The microcontroller has an on-board Digital to Analog Converter (DAC) which can output any voltage from 0 to 3.3 V with a 12-bit resolution. By configuring the motor drive to using the analog input, this voltage range can be used to control the motor over the required speed range, both forwards and backwards. The motor drive expects a voltage range from 0 to 10 V, thus 3.3 V covers only 33% of the total range. This is first centered around zero by applying an offset of -16.5%, after which the maximum scaling of 500% is applied to give a final range of $\pm 82.5\%$ of the maximum motor speed. Thus, to reach the maximum required 2143 rpm, the maximum speed on the motor drive must be set to $2143/82.5\% \approx 2600\,\mathrm{rpm}$.

With this setting determined, the complete conversion factor from DAC to cart speed can be calculated. The on-board DAC has a resolution of 12 bits, giving a total of $2^{12} = 4096$ voltage levels. As the analog reference input is centered around zero on the drive, this leaves 2048 levels positive and negative (actually 2047 on the positive side, but this error is considered small enough to be negligible). From this value can be determined that the conversion factor from DAC input to motor speed is

$$\eta_{\mathrm{DAC,motor}} = \frac{2143}{2048} = 1.046\,\mathrm{rpm/lsb},$$

where a Least-Significant Bit (lsb) represents the smallest change possible on the DAC. This can be combined with $\eta_{motor,axle} = 1/10$ to take into account the gearbox between motor and axle, and

$$\eta_{\text{axle,cart}} = \frac{2\pi 0.07}{60} = 7.3 \times 10^{-3}\,\text{m/s/rpm}$$

which converts rpm to Hz and converts the rotation to linear motion. Combining all these conversion factors gives

$$\eta_{\text{DAC,cart}} = \eta_{\text{DAC,motor}} \times \eta_{\text{motor,axle}} \times \eta_{\text{axle,cart}} = 767 \times 10^{-6}\,\text{m/s/lsb}.$$

Multiplying this with the maximum DAC value of 2048 gives a cart speed of 1.57 m/s, which is indeed the maximum speed for a 0.5 Hz, 0.5 m amplitude sine wave.

As a side effect, using an analog signal allows for simpler code and a faster update frequency; instead of sending multiple bytes over a serial line, only the DAC output has to be set to a new value. By additionally using dedicated digital signals for turning the motor on and off and notifying the controller of drive fault states, all required functionality is implemented.

4.5 Safety

As discussed before, in all circumstances it should be possible to bring the system to a safe state and stop the cart from moving. This is achieved with an emergency safety switch connected to the motor drive, using T3 as input. Under normal operating conditions, this switch is normally-closed and allows the motor drive to power the motor as instructed by the PI controller. When the switch is pressed, it opens the connection and the drive responds by quickly bringing the motor to a halt using a ‘Fast Stop’. The switch remains open until released by rotating it, allowing it to return to its original position.

During testing, the presence of the safety switch was essential to run the first fully integrated tests, as the software was in some cases not yet able to reliably turn the motor drive on and off. Additionally, when the DAC is configured incorrectly, the analog reference input is left floating or remains at zero voltage, which is interpreted by the motor drive as full speed. In the few cases this happened, the safety switch prevented collisions which could have damaged the mechanical components of the system.

5 Testing & Validation

This chapter will discuss the tests performed with the system and its components, and the results of these tests. To be able to thoroughly test all components of the system, a test plan has been written that should be followed when testing the system or any part of it, which can be found in Appendix A. Any part of the testing that went smoothly will only be discussed very briefly, while the difficulties will be elaborated on.

5.1 Unit tests

5.1.1 Encoder

Testing of the encoder has been done by the software group, and is described in their thesis [2].

5.1.2 Proximity sensor

After connecting the proximity sensor, the sensor did not seem to switch its differential output when a metal object was held near it. It was found that the way the proximity sensor was mounted on the rail made it trigger on the rail itself, thus not being able to sense the cart. After some adjustments in the mounting position, the sensor worked as intended.

5.2 Partially integrated tests

5.2.1 Motor drive, motor

After attaching the motor to the motor drive, the same setpoints as given during the drive unit test can be sent to the drive again. At this point the verification was done by measuring the time it took the motor to perform a certain amount of turns, and calculating the rpm accordingly. At short time intervals this is not a very accurate test, but as the time intervals increase in length this can give a good estimation of the actual rotational velocity of the motor shaft. The reported measurements have been made with a duration of 30 to 60 seconds. This test is also only appropriate for low rotational velocities, and is only used to make sure the motor behaves as expected before hooking it up to the rail and encoder. The test results are given in Table 5.1. From the measurements we can conclude that the motors' rotational velocity is close to the requested setpoint.

The small difference between the setpoint rpm and the actual rpm is because of a number of reasons. Firstly, the signal from the DAC has a voltage between 0 V and 3.3 V, while the drive expects values from 0 V to 10 V. The drive internally shifts and scales this signal according to the settings in Appendix C, meaning any error will be scaled by the same value. It also appeared that the drive rpm increased in steps of 3 rpm, or sometimes even 4 rpm, rather than the approximately 1 rpm accuracy that should be provided by the DAC (4.4.2). These errors will be compensated by the motor control loop, which is implemented by the software group [2]. Despite the small error, it is considered safe to attach the motor to the rail, making it possible to do far more accurate rotational velocity measurements using the rotary encoder.

Requested RPM	DAC digital value	DAC analaog value [V]	Drive-shown RPM	RPM measured
-1500	614	0.495	-1532	
-1000	1092	0.880	-1028	
-500	1570	1.265	-523	
-200	1857	1.496	-222	
-100	1952	1.573	-120	-119.08
-50	2000	1.612	-69	-67.32
-20	2029	1.635	-41	-37.10
0	2048	1.650	-19	-10.37
20	2067	1.666	0	0.00
50	2096	1.689	31	28.09
100	2144	1.728	82	80.03
200	2239	1.804	180	
500	2526	2.036	485	
1000	3004	2.421	990	
1500	3482	2.806	1494	

Table 5.1: The requested rotational velocity in rpm against the actual measured velocity, and some additional values.

5.2.2 Motor drive, motor, encoder

During this test the motor was spinning with a sinusoidal motion, while simultaneously reading out the encoder value. This measurement can be seen in Figure 11, where it is visible that the peaks drift 10 cm over 18 periods. By marking the physical peak positions on the rail as the cart passes by, this drift was measured to be only 3 cm. The 3 cm drift can be explained by irregularities in the system, causing higher speeds in one direction. These irregularities are to be solved by implementing the motor control loop (**P.9**). Thus, there is still 7 cm unaccounted for.

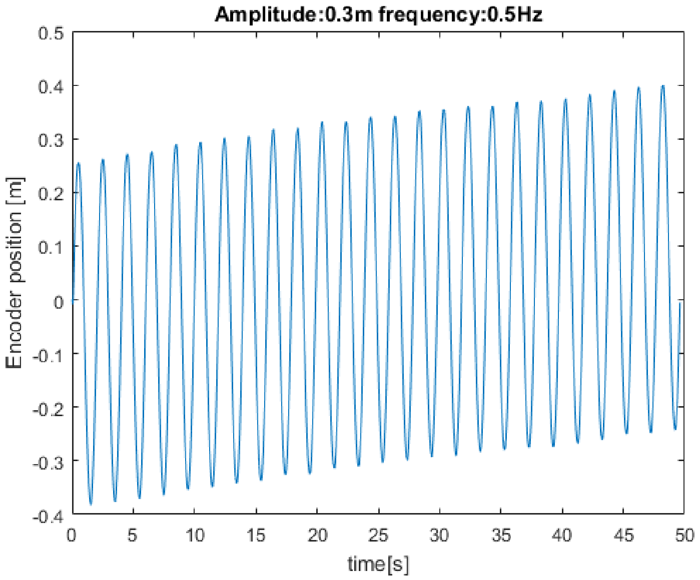


Figure 11: Encoder reading clearly showing drift.

When turning the shaft by hand, there was no measurable drift on the signal, verifying that the results from the encoder unit test still hold. The motor was detached from the rail at this point. Next, the encoder shaft was turned by hand with the motor spinning freely at the same time, again showing a very noticeable drift on the signal. Knowing this could not be a problem in the software (the exact same software was used every time), electromagnetic interference was suspected to affect the signal [32]. The cable connecting the drive to the motor (6.5 m long) and the cable connecting the encoder to the development board (4.0 m long) are both completely unshielded and were stacked on top of each other as only about 0.5 m is actually necessary at this time. To test this hypothesis, the motor cable was put in a metal box, creating a Faraday cage [33] to prevent further interference to other signals, while the encoder cable was placed as far away from any other cables as possible. Running the same test again resulted in far more pleasurable measurements, as can be seen in Figure 12.

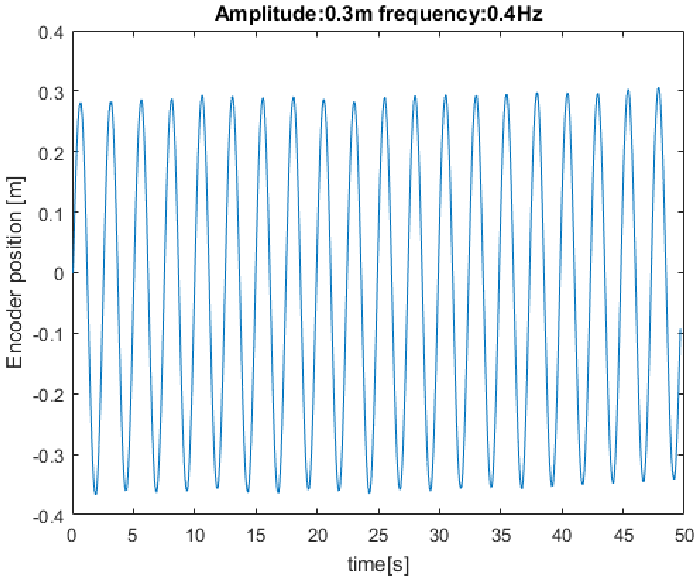


Figure 12: Encoder reading with cables shielded.

To further investigate the problem, the output signals of the encoder were measured with an oscilloscope. Figures 13 and 14 show encoder outputs A and A* at the end of the cable. On top of the encoder pulses with an amplitude of 3.7 V, interference is visible with peaks up to 1 V. These short pulses can, with large enough amplitude, trigger a transition in the encoder readout and thus cause drift. As the encoder readout works with the edges of the signal rather than averages over a certain period of time, the usual definition of Signal-to-Noise Ratio (SNR) as the ratio of average powers is of no use. However, the instantaneous SNR can be calculated at each peak, giving for the mentioned values an SNR of

$$20\log_{10}\left(\frac{3.7}{1}\right) = 11\,\mathrm{dBV}.$$

One way of increasing the SNR is by using the differential output of the rotary encoder, as already mentioned in Section 3.1.3 and as recommended by the MCU reference manual [34]. By using a differential amplifier which subtracts the negative signals A* and B* from the positive signals A and B, any common-mode interference is canceled while the difference including the signal is amplified. This was thus implemented as can be read in Section 4.2.2. The output of the differential amplifier can be seen in Figure 15, which shows almost no spurious voltage spikes. With interference peaks up to 0.2 V on the now 2.9 V signal, the SNR has improved to

$$20\log_{10}\left(\frac{2.9}{0.2}\right) = 23\,\mathrm{dBV}.$$

It should be noted that the signal dip just before each rising edge will not pose a problem. These are present on only one signal (A or B) at a time, rather than on both simultaneously as with interference. Running the original tests again, results similar to Figure 12 were achieved where the excessive drift was no longer present, further confirming that the interference problem is now solved.

Another possibility to increase the SNR is to amplify the signal at the transmitting side, and scale down the received signal together with the interference. This will scale the SNR linearly with the amplification factor. As 24 V power is available, this could be used, theoretically improving the SNR to

$$20 \log_{10} \left(\frac{24}{1} \right) = 28\,\mathrm{dBV}.$$

While potentially providing an even higher SNR, the main issue with this implementation is that it requires circuitry at the transmission side, introducing another point of failure. Additionally, it makes sense to use the already available differential signals.

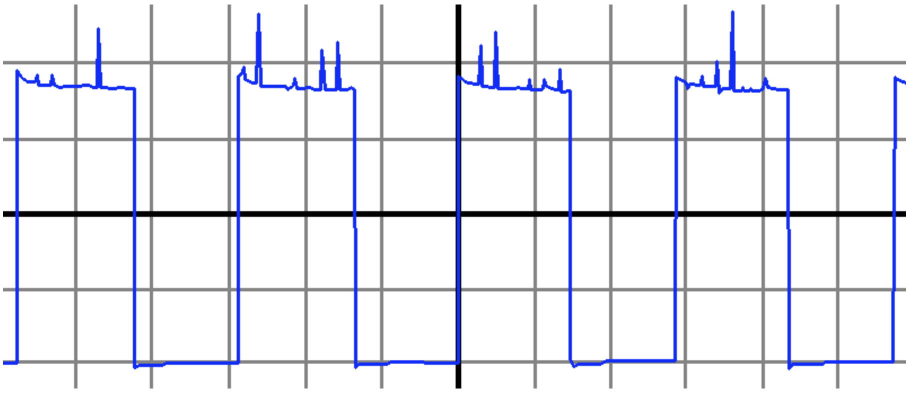


Figure 13: Encoder output A, with the motor spinning. Horizontal 200 µs/div, vertical 1 V/div.

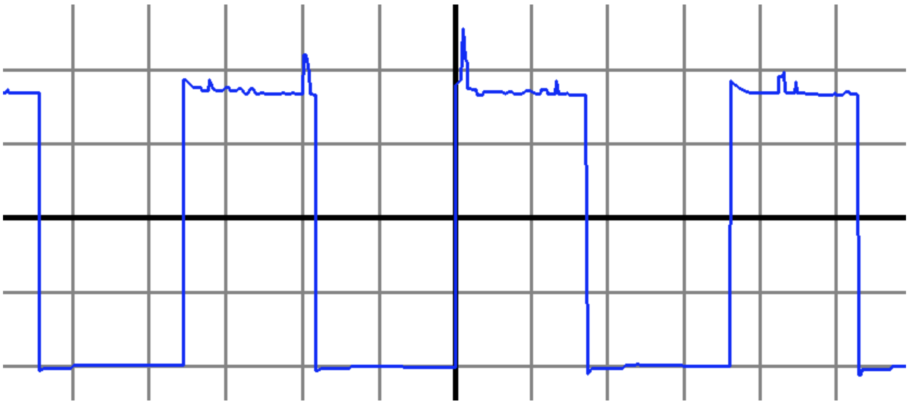


Figure 14: Encoder output A*, with the motor spinning. Horizontal 200 µs/div, vertical 1 V/div.

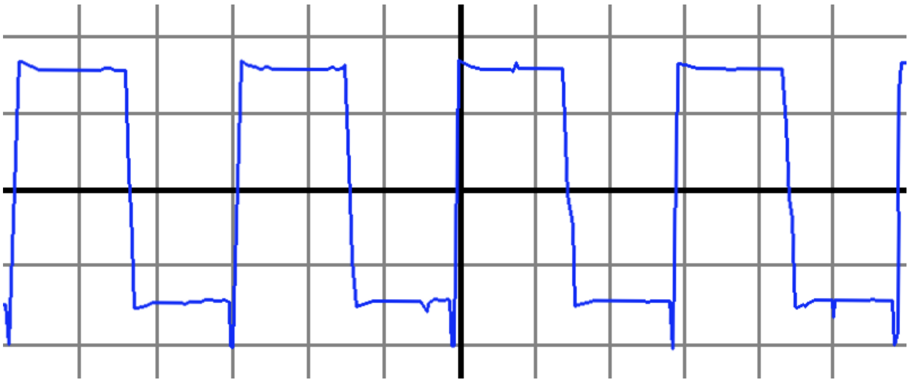


Figure 15: Differential amplifier output, with the motor spinning. Horizontal 200 µs/div, vertical 1 V/div.

5.2.3 Motor drive, motor, encoder, cart

At this point the cart is attached to the gear. The encoder measurements will from this point on be converted from rotational encoder positions to distance on the rail with a conversion factor of 44 µm per position, as calculated in Section 3.1.3. In the measurement seen in Figure 12 there seems to be some drift over time, as most clearly seen in the peaks. This does however correspond to the actual drift of the cart, and is most likely caused by irregularities in the system. This drift is expected, as the only driving signal is a hard-coded feed-forward sinusoid and no feedback control loop is implemented at this point. Any amplitude error will also be solved by implementing the control loop software [2].

6 Conclusion

A testing system was successfully delivered to Ampelmann, meeting most of the requirements. The mechanical setup was verified and deemed satisfactory for its purpose. The electrical control platform was designed, implemented and validated. Together with the peripheral components, the development board-based system is able to control the MRUs motion while simultaneously harvesting data from the rotary encoder, creating an base-line to which MRUs can be compared. This system now provides a platform for control systems and software to accurately describe the behaviour of an MRU, ultimately verifying whether a certain MRU is deemed satisfactory for its purpose in the Ampelmann systems.

Figure 16 gives an overview of the complete system, with the MRU assessment and software subgroup components included. This figure is the diagrammatic version the figure in Appendix D.

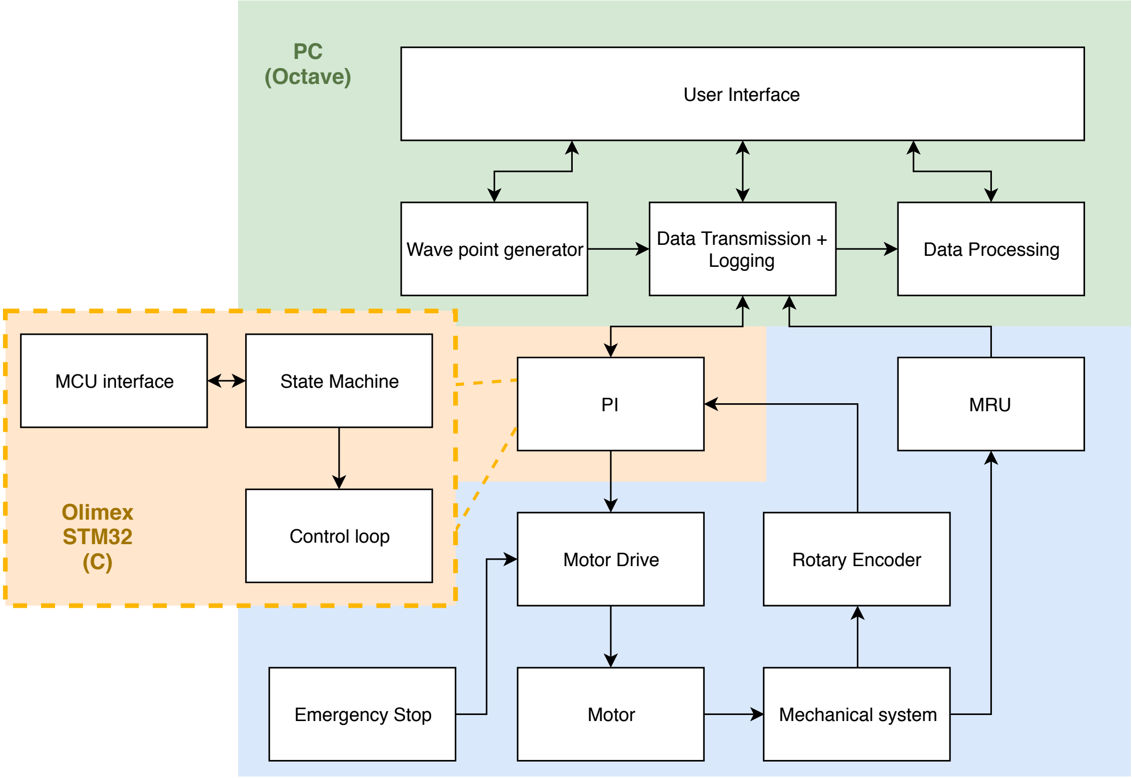


Figure 16: Complete system overview

6.1 Final specifications

The following list provides the final specifications of the system, and how they relate to the set requirements and preliminary specifications.

S.1 Corresponding to **R.1** and **P.1**, a 1DoF rail is used. However, it can not yet be placed vertically.

S.2 Corresponding to **R.2** and **R.3**, the motion of the cart is streamed live from a host PC, providing virtually unlimited customization in waveform shape and length.

S.3 Corresponding to **R.2**, **R.11**, **P.6**, **P.9**, a rotary encoder is available for use in the motor control loop to follow the motion to within 1 mm. Whether this actually is achieved depends on the motor control loop itself.

S.4 Corresponding to **R.4** and **P.2**, a safe usable space of 156 cm is available.

S.5 Corresponding to **R.4**, **R.5**, **P.4**, **P.5**, **P.13**, the motor and motor drive can provide enough power to the system while not requiring three-phase power.

S.6 Corresponding to **R.4**, **R.5**, **P.3**, **P.4**, the motor can provide the torque required to move the cart with the specified motion.

S.7 Corresponding to **R.6** and **P.12**, a simple hardware interface is available, the system has been documented, and screw-terminals or simple connectors are used where possible.

S.8 Corresponding to **R.7**, **P.7**, **P.14**, safety has always been taken into account, especially concerning the mechanical system with moving parts, and an emergency stop button has been implemented.

S.9 Corresponding to **R.8**, **R.9**, **R.10**, **P.8**, an Olimex E407 development board with microcontroller is the center of the electrical control system, complemented with some additional electronics to interface with peripheral components.

S.10 Corresponding to **R.11**, **P.6**, **P.10**, the rotary encoder provides the ground truth within 1 mm, sampled at 50 Hz.

S.11 Corresponding to **R.12**, the MRU is connected directly to a host PC for data acquisition.

S.12 Corresponding to **P.11**, the E407 board communicates with the host PC over UDP.

6.2 Future work

To fully fulfill requirement **R.1**, the rail should be able to operate in both a horizontal and vertical orientation. The vertical operation mode should still be implemented. This does however require some more thought as cutting off the motor drive using the safety switch may result in the cart with the MRU dropping towards the bottom of the rail. Some kind of braking system should be implemented in order for this to work. A possibility is using the motor drive to brake using DC injection braking [35]. Additionally, some sort of cushioning blocks at the ends of the rail may prevent damage to the components.

Furthermore, the hardware control interface proved useful during development, but many of its functions have been taken over by the host PC. Therefore, a simpler interface will suffice, mainly providing the emergency stop button and status LEDs.

6.3 Recommendations

At the start of the project, our initial idea of building the self-designed TAU was the main focus. Due to the anticipated complexity of designing and building a complete PCB, we wanted to quickly start with ordering components for prototyping. However, as later in the project the TAU was dropped in favour of the PI, much of this effort was lost. It would have been better to have started with careful consideration of the requirements and cost of the system before prototyping.

As this was a project with three groups, the definition of the interface requirements was very important. This as well proved harder than initially expected. It was often necessary to take a step back and reconsider what interfaces were currently defined.

Finally, working at a real company was a very different experience from doing projects at the university. While at the university the main goal is to do research and come with new ideas, at a company the focus is more on getting the job done with whatever means necessary, with the primary constraints being time and money.

7 Discussion

When choosing a development board to use as main component for the PI, there was no possibility to actually try a few different development boards. Especially for the software group it would have been a good thing to be able to test different development boards to weigh some pros and cons that are not easily reconstructable out of datasheets and schematic designs. There were eventually two different development boards that were tested. Even though the board that was chosen was thoroughly examined before selection, having more options might have proved useful in hindsight.

The mechanical system was already partially available at the start of this project. While this saved the groups a lot of time on the mechanical aspect of the project, it also comes with a drawback. The mechanical setup was largely predefined, leaving less options for the group to create an optimal system for the provided use-cases. Were this system not to be as pre-defined, more options could have been explored, such as a system that supports more than one DoF.

Time was a clear constraint in this project. While it is certainly not impossible to finish this project in the pre-determined 10 weeks, it does not leave much room for iterating. If more time was available, thus making it possible to have at least 1 iteration on design, there would be less drawbacks to continuing the entirely self-designed TAU.

When measuring the encoder drift, it was found that the actual drift per sinusoidal period of the cart during a test was different if the cart moved at different speeds. This observations should have been caught earlier on during the project. This could very well have to do with the drive being able to match the sinusoidal movement better if a higher speed if asked, but this should be investigated. Also when other components were tested for the first time should the results have been documented more thoroughly, instead of quickly moving on to the next part.

A proper next step for this system is expansion in terms of DoF. The current setup is able to measure 1 DoF, but excludes the rotational degrees, roll, pitch & yaw. Expanding the system to be able to measure these rotational degrees would be recommended in order to better assess the different MRUs. Working up to a 6 DoF system will prove challenging, as a lot of complicated mechanical and electrical hardware and controls will be necessary. A two or three DoF system, however, could be a proper next goal. A rotational degree of freedom should be the next to investigate, as a smart implementation of a 2 DoF system could potentially measure all 6 DoFs. This could be done by repositioning the rail and the MRU.

Another topic of discussion is the ethical aspect behind this project and the Ampelmann systems. Ampelmann clearly and rightfully so cares a lot about the safety of people using their systems. Instead of using just the datasheets of MRUs, the company thrives for testing every part in their system as thoroughly as possible. A pitfall for Ampelmann could be technological enthusiasm. Working with systems like this really makes engineers want to push the boundaries of what is possible. Care should be taken that employees stay focussed on their main goal, which is letting people cross the ocean just as easily as crossing the street.

Sustainability is also a part of the ethical correctness at Ampelmann. Moving cargo and people over an Ampelmann system is much more sustainable than transport using for example a helicopter.

It is certain that Ampelmann puts thought into ethical topics.

References

[1] CalQlata, *Vessel 6-Degrees of Movement*. [Online]. Available: http://www.calqlata.com/productpages/00059-help.html

[2] D. Benders and B. Burgers, "Bachelor graduation project, thesis," *Pre-release*, June 2018.

[3] C. van der Horst and M. van der Marel, "Bachelor graduation project, thesis," *Pre-release*, June 2018.

[4] *Low Voltage Aluminium Motors, Range M3000*, ABB.

[5] *Optidrive ODE-2 User Guide*, Invertek Drives, 2014, Revision 3.30.

[6] *H48 2500 C/T rotary encoder datasheet*, Tamagawa Seiko Company.

[7] *IM5097 inductive sensor datasheet*, ifm electronic gmbh, 2003.

[8] *Modbus Application Protocol Specification V1.1b3*, Modbus.org, April 2012.

[9] (2016, March) 802.3-2015 - IEEE Standard for Ethernet. IEEE. [Online]. Available: https://ieeexplore.ieee.org/document/7433918/

[10] K. Wang and H. Shen, "A New Design of Multiaxial Efficient Motion Control Card Based on MCU and FPGA," in *2010 International Conference on Measuring Technology and Mechatronics Automation*, March 2010.

[11] P. Bishop. (2009) A tradeoff between microcontroller, DSP, FPGA and ASIC technologies. EE Times. [Online]. Available: https://www.eetimes.com/document.asp?doc_id=1275272

[12] Xilinx, "Zynq-7000 Product Overview," https://www.xilinx.com/support/documentation/data_sheets/ds190-Zynq-7000-Overview.pdf, June 2017, Accessed: 2018-04-24.

[13] STMicroelectronics. STM32 32-bit ARM Cortex MCUs. Accessed: 2018-04-24. [Online]. Available: http://www.st.com/en/microcontrollers/stm32-32-bit-arm-cortex-mcus.html

[14] *CoreRMII v2.0 Handbook*, Microsemi, 2016.

[15] *Ethernet PHY Transceiver Reference Guide (Rev. A)*, Texas Instruments, 2016.

[16] *STM32 Nucleo-144 boards User Manual*, 7th ed., STMicroelectronics, December 2017, UM1974.

[17] *STM32-E407 User's Manual*, OLIMEX, March 2018, .

[18] Liu et al. (2017, November) Absolute Rotary Encoder Patent Application Publication.

[19] R. J. Kosinski, *A Literature Review on Reaction Time*, September 2013, Clemson University.

[20] Mouser.com. Don't leave your pins floating. . [Online]. Available: https://www.mouser.com/blog/dont-leave-your-pins-floating

[21] Electronics Tutorials. Pull-up Resistor and Pull-down Resistor Explained. . [Online]. Available: https://www.electronics-tutorials.ws/logic/pull-up-resistor.html

[22] *LAN8710A/LAN8710Ai Small Footprint MII/RMII 10/100 Ethernet Transceiver with HP Auto-MDIX and flexPWR® Technology Datasheet*, SMSC, August 2012.

[23] *Quad-SPI (QSPI) interface on STM32 microcontrollers*, STMicroelectronics, April 2016, .

[24] *L78 Positive voltage regulator ICs Datasheet - production data*, STMicroelectronics, November 2016, .

[25] *uA78Mxx Positive-Voltage Regulators*, Texas Instruments, January 2015.

[26] *RS485 Communications Interface: Technical Manual*, Parker SSD Drives, 2007.

[27] *MOD-RS485-ISO User's Manual*, OLIMEX, April 2017, Accessed: 2018-05-21.

[28] *Fundamentals of RS-232 Serial Communications*, DALLAS semiconductor, March 1998, .

[29] *VO617C Datasheet*, Vishay, August 2013.

[30] *LM358 Datasheet*, Texas Instruments, January 2017. [Online]. Available: http://www.ti.com/lit/ds/

symlink/lm358.pdf

[31] *MODBUS over Serial Line: Specification & Implementation Guide V1.02*, Modbus.org, December 2006.

[32] M. Kaur, S. Kakar, and D. Mandal, "Electromagnetic interference," in *2011 3rd International Conference on Electronics Computer Technology*, 2011, . [Online]. Available: https://ieeexplore.ieee.org/document/5941844/

[33] *The Faraday Cage: What Is It? - Technical Note*, Gamry instruments, December 2010.

[34] STMicroelectronics. STM32F407 Reference Manual. Accessed: 2018-07-18. [Online]. Available: http://www.st.com/content/ccc/resource/technical/document/reference_manual/3d/6d/5a/66/b4/99/40/d4/DM00031020.pdf/files/DM00031020.pdf/jcr:content/translations/en.DM00031020.pdf

[35] C. Somers, C. Moussa, and K. Al-Haddad, "Emergency dc injection braking system," in *Proceedings of the IEEE International Conference on Industrial Technology*, vol. 2015, 06 2015, pp. 726–730.

Appendices

A Integration test plan

Under any circumstances are the people performing any part of this test plan urged to use common sense to keep a safe environment. No test should be executed by a single person. A minimum of two people is required to do the testing, a three person group is recommended however.

If at any time during testing the system does not act as expected, it is advised to abort the test and only continue testing after some thorough thought has been put into the observations.

Document all observations thoroughly during testing.

A.1 Unit tests

In addition to the components described in the tests, the the following components are required:

- OLIMEX E407 development board
- Peripheral extension board
- OLIMEX MOD RS-485 ISO peripheral board
- ST-Link programmer
- A PC with CubeMX and Eclipse installed to upload software to the development board

For all tests, the peripheral extension board is to be connected to the E407 development board according to Table A.1.

Extension board	**E407**
Black	Ground (pin 2 and 20 on any outside header)
Red	5 V supply (pin 19 on any outside header)
Yellow (extended with white)	Motor drive enable (configured as G15, pin 18 on PG header)
Blue/white-striped	Single-ended quadrature encoder signal (configured as E9 and E11, pins 12 and 14 on PE header)

Table A.1: Peripheral extension board connections

A.1.1 Encoder

This test covers the readout of the incremental encoder (Tamagawa H48 series incremental encoder).

- Connect the encoder to the peripheral extension board by matching all colored wires.
- Upload code capable of reading encoder data to the E407 development board.
- Reset the development board using the reset button and read the encoder value on the PC.
- Turn the encoder shaft and watch the rotational angle and distance value on the PC. Make sure these values roughly correspond to what is happening, and that the encoder position ends up at the same value when the shaft is rotated back to its original position.

Note: Do not forget to take the gear ratio in mind when processing the encoder data to read out in meters.

A.1.2 Proximity sensor

This test covers the readout of the proximity sensor (IFM electronic IM5097).

- Connect the proximity sensor to the peripheral extension board according to Table A.2

- Hold the rail cart close to the sensor and watch if the proximity flag triggers. This can be seen as the amber LED on the sensor will light up if it detects an object.
- Check if both of the outputs produce the expected differential signal and change value when the sensor triggers.

A.1.3 Motor drive

This test covers the communication to the motor drive (Invertek Optidrive E2) as well as the safety switch and communication loss behavior of the drive. Any test involving the drive may only be executed after the settings of the drive have been checked for correctness. The correct drive setting can be found in Appendix C.

- Connect the DAC output of the E407 (D10) to the analog input (T6) of the drive.
- Connect the level-shifted enable signal from the peripheral extension board to the drive enable input (T2). The extension board needs 24 V provided by the motor drive on T1.
- Connect the emergency safety button to the drive. One end needs to be connected to T1, the other to either T3 (analog control with P-15 set to 12) or T2 (Modbus control with P-15 set to 0).
- Test whether the drive stays in the 'STOP' state while the enable pin is set to a logic '0' (voltage between 0 and 4 V) or left floating, and exits the 'STOP' state when enable is set to a logic '1' (8 to 30 V).
- Check whether the rotational velocity setpoints match the expected setpoints according to the DAC output. This can be done by reading the drives' rpm setpoint on the drive itself.
- Check whether the drives' setpoint becomes 0 rpm and stops turning as soon as the safety button is pressed.

A.2 Partially integrated tests

Before executing any of the partially integrated tests, all of the unit tests should have been executed successfully and the results documented. Connecting the components can be done as described in the unit test for that specific component.

A.2.1 Motor drive, motor

This test covers the communication to the drive, as well as the use of the motor. Before executing this test, make sure the drive settings are correct and the motor drive stand alone test has been successfully executed.

- Connect the motor (ABB motors 3GVA072001) to the drive
- Send a setpoint for 100 rpm to the drive, check if the motor speed matches the setpoint.
- Press the safety switch, check if the motor stops according to the dec. ramp in setting P-04.
- Release safety switch, check that motor does NOT start spinning again.
- Reset system, check if motor starts spinning again after sending the 100 rpm setpoint.
- break communication, check that motor stops according to dec. ramp.
- restart communication, check that motor does NOT start spinning again.
- reset the system, check if motor starts spinning again after sending the 100 rpm setpoint.
- gradually increase speed setpoints (100 rpm increments), up to the max desired rpm. (2100), checking the safety switch and communication loss faults every 500 rpm.

A.2.2 Motor drive, motor, encoder

This test covers the communication to the drive, as well as the use of the encoder in combination with the motor. Attach the motor to rail but do NOT attach the cart to the gear!

- Connect the motor (ABB motors 3GVA072001) to the drive.
- Send a setpoint for 100 rpm to the drive, check if the motor speed matches the setpoint. Check if the encoder readout matches the expected value.
- gradually increase speed setpoints (100 rpm increments), up to the max desired rpm. (2100), checking the safety switch and communication loss faults every 500 rpm. Constantly check whether the encoder output matches the expected value.

A.2.3 Motor drive, motor, encoder, cart

This test covers the communication to the drive as well as the use of the motor, encoder and cart on the rails.

- Upload the Motor drive communication + encoder code to the E407 dev. board.
- Connect the motor (ABB motors 3GVA072001) to the drive
- Send a setpoint for 100 rpm to the drive, once the motor (and cart) start to move, press the safety switch to verify that the cart stops as expected.
- Release the emergency button, the cart should NOT start to move again.
- Reset the system. Send the 100 rpm setpoint again and break the communication to the drive once the cart moves. Check if the cart comes to a stop as expected.
- Check if cart moves at the expected speed by measuring the time it takes for the cart to move a predetermined distance on the rail.
- Move the cart close to the safety bounds of the rail (implemented in software). Set the speed to 100 rpm again so that the cart moves towards the safety bound, and check if it stops at the expected position. (25 cm from the rail edge). Repeat the test for the other side of the rail.
- Increase speed of motor until max rpm needed (2100rpm) constantly checking safety switch, bounds and communication errors. Use higher speed settings to check whether the dynamic bounds (software functionality) function as intended.

A.2.4 Motor drive, motor, encoder, rail, proximity sensor

This test covers the communication to the drive as well as the use of the motor, encoder, proximity sensor and cart on the rails.

- Upload the Motor drive communication + encoder code to the E407 dev. board.
- Connect the motor (ABB motors 3GVA072001) to the drive
- Place cart close to induction sensor
- Send a setpoint for 100 rpm to the drive, So that the cart starts to move towards the induction sensor.
- Check if the sensor triggers and the cart stops. Write down the distance between the cart and the rail edge
- Repeat test another two times.
- Increase speed to a speed that seems reasonable for the homing functionality and repeat test three times.

A.3 Fully integrated test

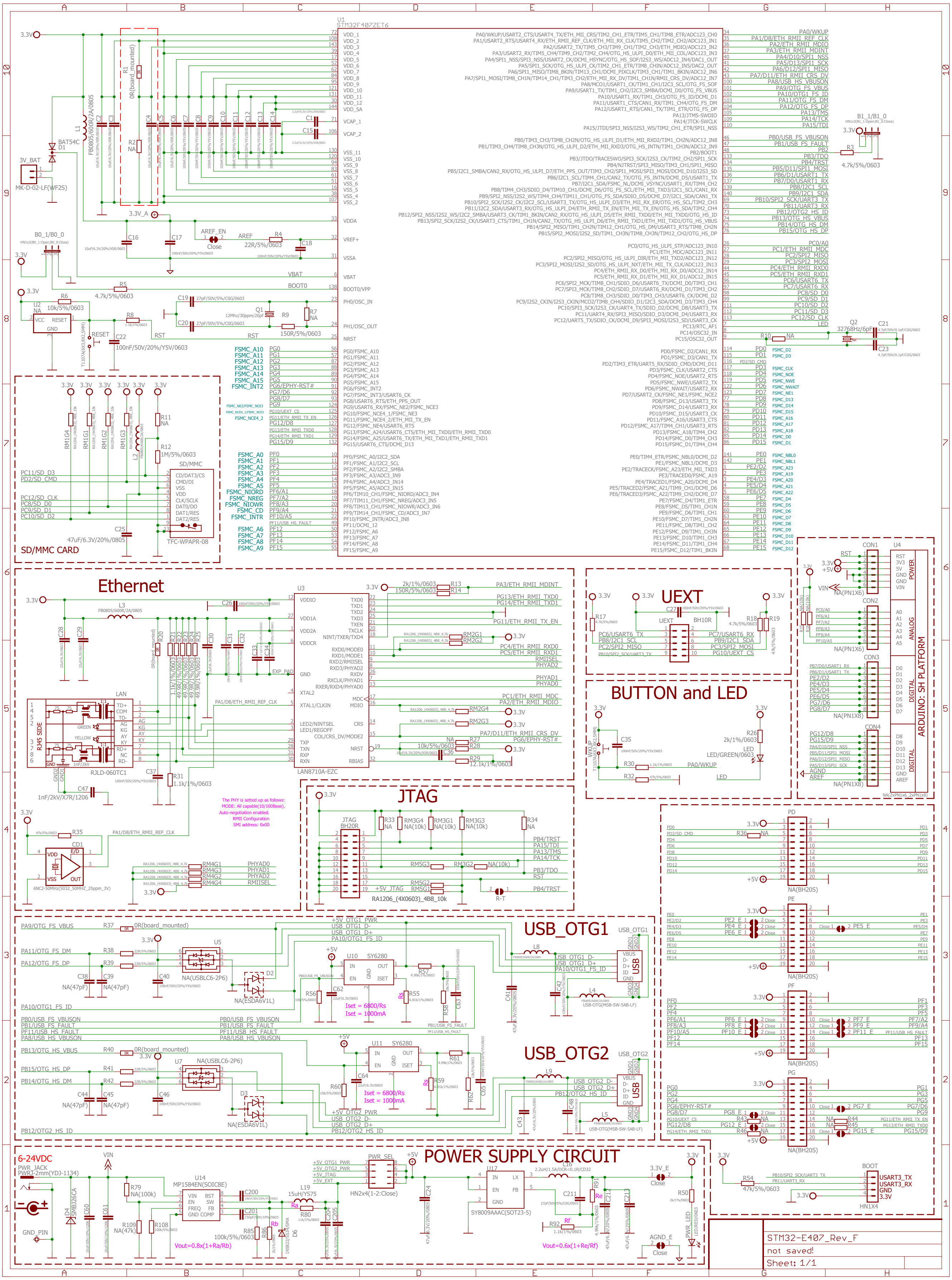
Before executing any of the fully integrated tests, all of the Unit tests and partially integrated tests should be successfully executed and documented. From this point on the entire system code will be flashed on the dev. board.

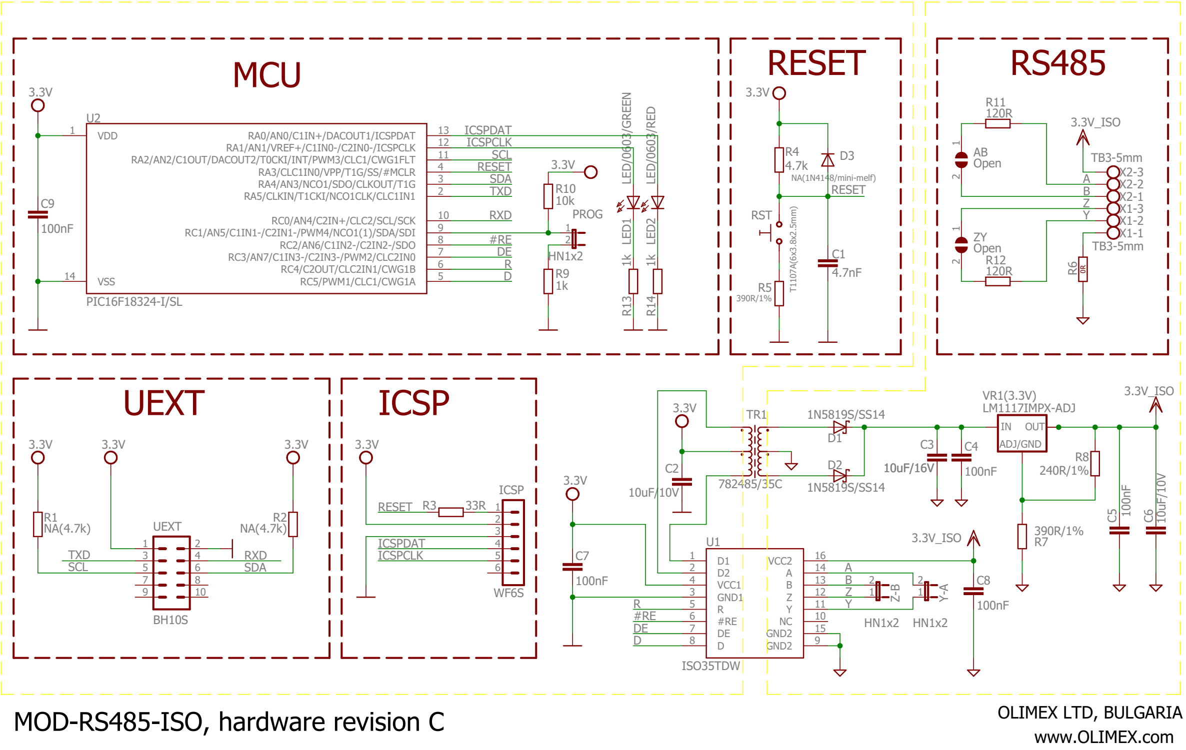
As integration tests begin, all of the hardware has already been tested and verified. The fully integrated test is mostly verifying the state machine written by the software group, while still implicitly verifying the hardware. Some hardware problems might still come up at this stage, but are certainly not expected.

Proximity sensor wire	**Connection**
Blue	Ground (any ground pin on the extension board or E407)
Brown	24 V supply (from the extension board or external power supply)
White	Out + (to white on extension board)
Black	Out - (to black on extension board)

Table A.2: Inductive proximity sensor connections

B OLIMEX E407 AND MOD-RS485 ISO SCHEMATICS




MCU
RESET
RS485
UEXT
ICSP
3.3V
U2
VDD
VSS
PIC16F18324-I/SL
RA0/AN0/C1IN+/DACOUT1/ICSPDAT
RA1/AN1/VREF+/C1IN0-/C2IN0-/ICSPCLK
RA2/AN2/C1OUT/DACOUT2/T0CKI/INT/PWM3/CLC1/CWG1FLT
RA3/CLC1IN0/VPP/T1G/SS/#MCLR
RA4/AN3/NCO1/SDO/CLKOUT/T1G
RA5/CLKIN/T1CKI/NCO1CLK/CLC1IN1
RC0/AN4/C2IN+/CLC2/SCL/SCK
RC1/AN5/C1IN1-/C2IN1-/PWM4/NCO1(1)/SDA/SDI
RC2/AN6/C1IN2-/C2IN2-/SDO
RC3/AN7/C1IN3-/C2IN3-/PWM2/CLC2IN0
RC4/C2OUT/CLC2IN1/CWG1B
RC5/PWM1/CLC1/CWG1A
ICSPDAT
ICSPCLK
SCL
RESET
SDA
TXD
RXD
#RE
DE
R
D
C9
100nF
R10
10k
PROG
HN1x2
R9
1k
LED/0603/GREEN
LED/0603/RED
LED1
LED2
R13
R14
1k
R4
4.7k
D3
NA(1N4148/mini-melf)
RST
T1107A(6x3.8x2.5mm)
R5
390R/1%
C1
4.7nF
R11
120R
3.3V_ISO
AB
Open
ZY
Open
R12
120R
TB3-5mm
A
B
Z
Y
X2-3
X2-2
X2-1
X1-3
X1-2
X1-1
R6
0R
R1
NA(4.7k)
R2
NA(4.7k)
UEXT
BH10S
TXD
SCL
RXD
SDA
RESET
R3
33R
ICSP
WF6S
ICSPDAT
ICSPCLK
C7
100nF
C2
10uF/10V
TR1
782485/35C
1N5819S/SS14
D1
D2
C3
10uF/16V
C4
100nF
VR1(3.3V)
LM1117IMPX-ADJ
IN
OUT
ADJ/GND
R8
240R/1%
390R/1%
R7
C5
100nF
C6
10uF/10V
U1
ISO35TDW
D1
D2
VCC1
GND1
R
#RE
DE
D
VCC2
A
B
Z
Y
NC
GND2
GND2
Z-B
Y-A
HN1x2
C8
100nF
MOD-RS485-ISO, hardware revision C
OLIMEX LTD, BULGARIA
www.OLIMEX.com

P-XX	Function	Value	Comment
01	Maximum Frequency / Speed Limit	2600 [rpm]	Never reached with analog control due to scaling limits.
02	Minimum Frequency / Speed Limit	0 [rpm]	
03	Acceleration Ramp Time	0.01 [s]	Short time for fast response.
04	Deceleration Ramp Time	0.01 [s]	Short time for fast response.
05	Stopping Mode	2	Ramp to stop.
06	Energy Optimiser	0	No optimisation.
07	Motor Rated Voltage	230 [V]	Motor nameplate rating in delta configuration.
08	Motor Rated Current	1.4 [A]	Motor nameplate rating in delta configuration.
09	Motor Rated Frequency	50 [Hz]	Motor nameplate rating.
10	Motor Rated Speed	1410	Motor nameplate rating.
11	Voltage Boost	5 [%]	Slightly boost torque at low speeds, to overcome static friction.
12	Primary Command Source	0	Terminal control. Set to 3 for Modbus control.
13	Trip Log History	N/A	
14	Extended Menu Access code	101	Default access code for extended parameters
15	Digital Input Function Select	12	Terminal control for normal stop, fast stop, and analog speed reference. Set to 0 for Modbus control.
16	Analog Input 1 Signal Format	b-10-10	Allow both forward and backward rotation with analog voltage control.
17	Maximum Effective Switching Frequency	8 [kHz]	
18	Output Relay Function Select	1	Output drive healthy, relay on when no faults are present.
19	Relay Threshold Level	100 [%]	
20	Preset Frequency / Speed 1	N/A	
21	Preset Frequency / Speed 2	N/A	
22	Preset Frequency / Speed 3	N/A	
23	Preset Frequency / Speed 4	N/A	
24	2nd Decel Ramp Time (Fast Stop)	0.01 [s]	Should be a fast as possible for safety.
25	Analog Output Function Select	8	Output frequency, can be fed back to controller (but not necessary).
26	Skip frequency hysteresis band	0.0	Default value, feature not used in this application
27	Skip Frequency	0.0	Default value, feature not used in this application
28	V/F Characteristic Adjustment Voltage	0	Default value, feature not used in this application
29	V/F Characteristic Adjustment Frequency	0.0	Default value, feature not used in this application
30	Terminal Mode Restart function	auto-0	
31	Keypad / Modbus Mode Restart Function	2	minimum speed, terminal enable
32	DC Injection Time On Stop	25 [s]	
33	DC Injection Time On Start	0	
34	Brake Chopper Enable	N/A	Brake chopper not present in motor drive
35	Analog Input 1 Scaling	500 [%]	Maximum value, for large control range.
36	Modbus RTU Serial Communications Configuration		Only required when Modbus is used. Address: 18, baud rate: 115200, timeout: 50 [ms]
37	Access Code Definition	101	Default access code
38	Parameter Access Lock	0	
39	Analog Input 1 Offset	16.5 [%]	Half of 33% to bring center analog reference around 0.
40	Display Speed Scaling Factor	0.000	Default value, feature not used in this application
41	PI Controller Proportional Gain	1.0	Default value, feature not used in this application
42	PI Controller Integral Time	1.0	Default value, feature not used in this application
43	PI Controller Operating Mode	0	Default value, feature not used in this application
44	PI Reference (Setpoint) Source Select	0	Default value, feature not used in this application
45	PI Digital Setpoint	0.0	Default value, feature not used in this application
46	PI Feedback Source Select	0	Default value, feature not used in this application
47	Analog Input 2 Signal Format	N/A	
48	Standby Mode Timer	0.0 [s]	No standby
49	PI Control Wake Up Error Level	0.0	
50	Thermal Overload Value Retention	0	

D Prototype setup

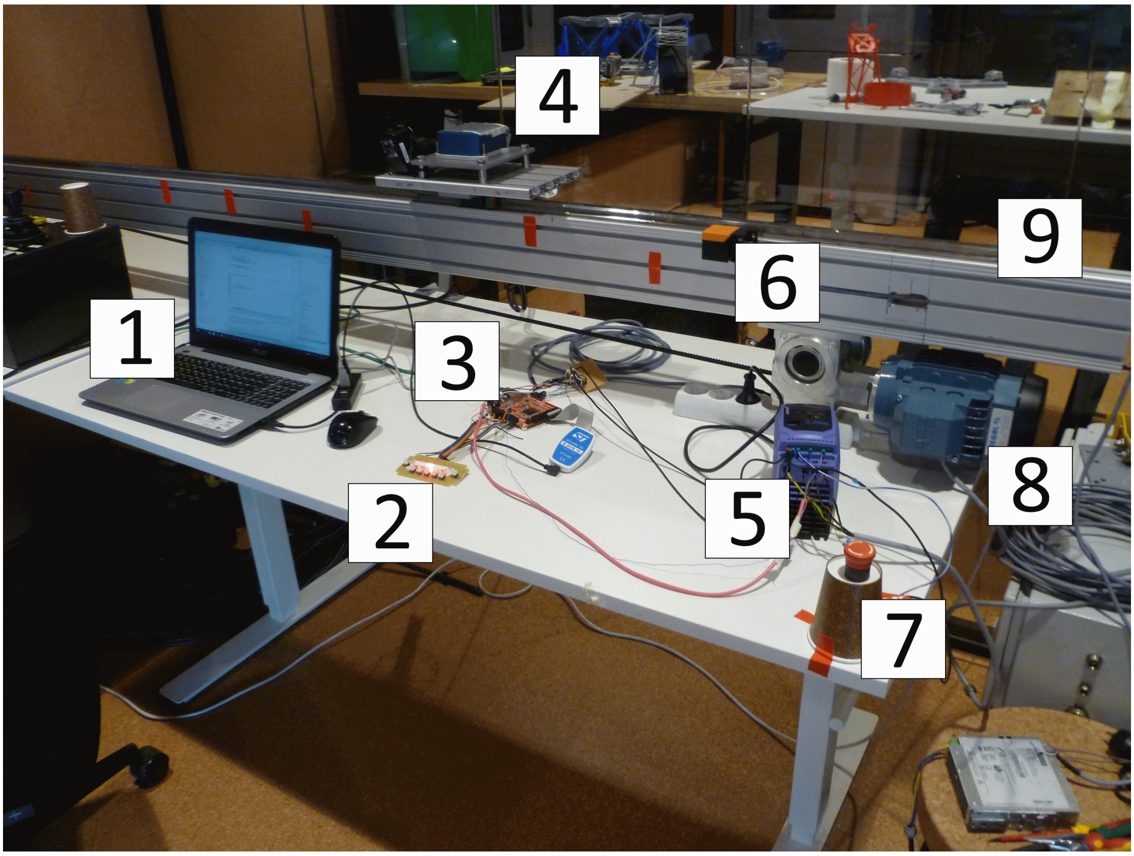


Figure 17: The complete system prototype setup

1	PC
2	Hardware interface
3	Development board & peripherals (PI)
4	Cart with MRU-under-test
5	Motor drive
6	Inductive proximity sensor
7	Emergency stop button
8	Motor
9	Rail
	Encoder located on the rear side (not visible here)

Table D.1: Components of the setup