



Modbus TCP/IP connection of power analysers for real-time monitoring of a demonstration microgrid: A Simulink approach

ANTHONY PELEMAN

Junho de 2016

Modbus TCP/IP connection of power analysers for real-time monitoring of a demonstration microgrid: A Simulink approach

Master degree in Electrical Engineering – Power Systems

Anthony Peleman

6-6-2016

Abstract

The demand for energy is growing more than ever and energy consumption is higher than the natural resources are able to supply. Therefore it becomes necessary that companies start reducing their energy consumption. Governments are creating more stringent environmental regulations, which raise energy costs for polluting companies and makes energy-saving measures more attractive.

Companies are constantly searching for methods to lower their energy costs. Hereby energy monitoring can offer a solution, as companies can examine their energy consumption at any time and every consumer can be watched separately.

By linking the measurement devices via a Modbus TCP/IP connection it is possible to make numerous measurements in various places. The software program proposed in this work does not depend on a specific brand of power analyser, making it an interesting and very flexible tool.

The Matlab/Simulink software provides a perfect program environment, giving a transparent and clear overview of the measured values. The software is easy to handle, even for beginners and makes it possible to make changes really fast.

In several case studies the program was tested on different types of consumers. This way the accuracy and the reliability of the program could be concluded.

Contents

Abstract	1
Contents	2
List of figures	4
Abbreviations	5
1. Introduction.....	6
1.1 Motivation	6
1.2 Energy monitoring.....	9
1.3 Matlab / Simulink	10
1.4 Thesis organization.....	10
2. Measurement devices and network.....	11
2.1 Power analyser	11
2.1.1 Specifications.....	11
2.1.2 Installation.....	12
2.2 MODBUS.....	13
2.2.1 MODBUS Application Protocol	13
2.2.2 MODBUS TCP/IP	19
3. Monitoring an electrical grid	22
3.1 Electrical grid	22
3.1.1 Location	22
3.1.2 Details of the grid	22
3.2 Monitoring an electrical grid	23
3.2.1 Overall Model	23
3.2.2 Power Analyser.....	25
3.2.3 Switchable network configuration	27
3.2.4 Real time simulation platform.....	28
3.2.5 Other busses.....	30
3.2.6 Cable L2A-L2B	32
3.3 Matlab code for the MODBUS TCP/IP connection	33
3.3.1 Communication between Matlab and the power analyser	33
3.3.2 Convert value from float to decimal.	35
3.3.3 Closing port and communication between Matlab and Simulink.....	35
4. Case studies.....	37

4.1	Ideal fixed-step time.....	37
4.2	Resistant load	39
4.3	Capacitive load	41
4.4	Inductive load	41
4.5	Load Cable	44
4.6	All loads implemented.....	45
5.	Conclusion	48
5.1	Word of appreciation	48
5.2	Main conclusion and future perspectives	48
	Bibliography.....	49
	Appendix.....	51

List of figures

FIGURE 1.1 2013 SHARES OF WORLD ELECTRICITY CONSUMPTION. (IEA, 2015)	6
FIGURE 1.2 MATLAB LOGO (MATHWORKS, 2016)	10
FIGURE 2.1 JANITZA UMG 96RM EL (JANITZA, 2016).....	11
FIGURE 2.2 CONNECTION DIAGRAM VOLTAGE MEASUREMENT (JANITZA, OPERATING INSTRUCTIONS AND TECHNICAL DATA, 2016) 12	
FIGURE 2.3 CONNECTION DIAGRAM CURRENT MEASUREMENT (JANITZA, OPERATING INSTRUCTIONS AND TECHNICAL DATA, 2016) . 12	
FIGURE 2.4 MODBUS (MODBUS, 2012).....	13
FIGURE 2.5 MODBUS PROTOCOL DESCRIPTION (MODBUS, 2012)	14
FIGURE 2.6 MODBUS PROTOCOL WITHOUT ERROR (MODBUS, 2012)	14
FIGURE 2.7 MODBUS PROTOCOL WITH ERROR (MODBUS 2012).....	14
FIGURE 2.8 IMPLEMENT FUNCTION CODE (MODBUS, 2012)	15
FIGURE 2.9 IMPLEMENT FUNCTION CODE (MODBUS 2012)	15
FIGURE 2.10 MODBUS PUBLIC FUNCTION CODES (MODBUS, 2012)	16
FIGURE 2.11 MODBUS FUNCTION CODE CATEGORIES (MODBUS, 2012)	17
FIGURE 2.12 EXAMPLE TO READ HOLDING REGISTERS 108-110 (MODBUS, 2012)	17
FIGURE 2.13 FUNCTION FOR READING THE HOLDING REGISTER (MODBUS, 2012)	18
FIGURE 2.14 TYPES OF MESSAGES IN TCP/IP NETWORKS (MODBUS, 2006).....	19
FIGURE 2.15 MODBUS APPLICATION PROTOCOL HEADER (MODBUS, 2006)	19
FIGURE 2.16 MODBUS APPLICATION DATA UNIT (MODBUS, 2006)	20
FIGURE 3.1 ELECTRICAL GRID OF THE GECAD LABORATORY	22
FIGURE 3.2 SIMULINK MODEL FIRST VIEW	24
FIGURE 3.3 SIMULINK MODEL POWER ANALYSER.....	25
FIGURE 3.4 SIMULINK MODEL POWER ANALYSER SUBSYSTEM.....	26
FIGURE 3.5 TCP/IP RECEIVE BLOCK (MATHWORKS, 2016)	26
FIGURE 3.6 INTERPRETED MATLAB FUNCTION BLOCK.....	26
FIGURE 3.7 SIMULINK MODEL SWITCHABLE NETWORK CONFIGURATION	27
FIGURE 3.8 SIMULINK MODEL BUS B1.....	28
FIGURE 3.9 SIMULINK MODEL BUS B1 SUBNETWORK	28
FIGURE 3.10 SIMULINK MODEL BUS B1 CALCULATIONS	29
FIGURE 3.11 CALCULATION CURRENTS	30
FIGURE 3.12 CALCULATION POWERS	30
FIGURE 3.13 SIMULINK MODEL BUS B4 SUBNETWORK	31
FIGURE 3.14 SIMULINK MODEL CABLE L2A-L2B	32
FIGURE 3.15 SIMULINK MODEL CABLE L2A-L2B CALCULATIONS	32
FIGURE 3.16 CONNECTION SIMULINK / MATLAB / POWER ANALYSER	33
FIGURE 3.17 MODBUS ADDRESS LIST FOR JANITZA UMG 96RM-EL (MATHWORKS, 2016).....	35
FIGURE 3.18 FLOW CHART OF MATLAB CODE THE MODBUS TCP/IP CONNECTION	36
FIGURE 4.1 SCREENSHOT OF WIRESHARK CAPTURING 1 REQUEST AND ANSWER OF METER POWER SUPPLY.....	37
FIGURE 4.2 SIMULATION OF A RESISTANT LOAD OF 1kW	40
FIGURE 4.3 SIMULATION OF A CAPACITIVE LOAD OF 4KVA	42
FIGURE 4.4 SIMULATION OF AN INDUCTIVE LOAD OF 0.37kW $\cos(\phi)=0.69$	43
FIGURE 4.5 SIMULATION OF THE LOAD OF CABLE L1A-L1B	44
FIGURE 4.6 SIMULATION OF ALL LOADS Bus B1, B2 AND B4.....	46
FIGURE 4.7 SIMULATION OF ALL LOADS TOTAL GRID CONSUMPTION	47

Abbreviations

- A = ampere
- ADU = Application Data Unit
- ANSI = American National Standards Institute
- bit = binary digit
- Dec = decimal
- EC = European Commission
- EIA = Energy Information Administration
- EU = European Union
- FPF = fundamental power factor
- GECAD = Research Group on Intelligent Engineering and Computing for Advanced Innovation
- GHG = Greenhouse Gas
- Hex = hexadecimal
- IEA = International Energy Agency
- IEC = International Electro-technical Commission
- IP = internet protocol
- IPCC = Intergovernmental Panel on Climate Change
- kWh = kilowatt hour
- kWh-class = effective energy class
- MBAP = MODBUS Application Protocol Header
- P = real power, active power
- PDU = Protocol Data Unit
- PLC = Programmable Logic Controller
- Q = reactive power
- RMS-value = Root Mean Square value
- S = apparent power
- s = second(s)
- TCP = Transmission Control Protocol
- V = voltage
- VA = volt-ampere
- VAR = volt-ampere reactive
- W = watt

1. Introduction

1.1 Motivation

The demand for energy is growing more than ever and energy consumption is higher than the natural resources are able to supply. The greenhouse gas (GHG) emissions, like energy related CO₂ emission is estimated by the Energy Information Administration to increase from 31.2 billion metric tons in 2010 to 36.4 billion metric tons in 2020 and 45.5 billion metric tons in 2040 (EIA,2013). The human population is growing and countries are developing, this increases the demand for energy. At the current rate, the Intergovernmental Panel on Climate Change expects the energy demand to increase by 65% from 2004 to 2030 (IPCC, 2007). They also state that global warming is the most important environmental problem of our time and reducing energy consumption seems a solution of handling the emission problem. The EU has set itself a target of 20% energy savings by 2020 compared to the projected use of energy in 2020, equivalent to turning off 400 power stations. For 2030 the EU countries agreed on an energy efficiency target of 27% or greater, a little bit under the proposed 30% of the European Commission (EC, 2014).

The report of the International Energy Agency puts forward that the industry uses 42.3% of the world electricity consumption (IEA, 2015).

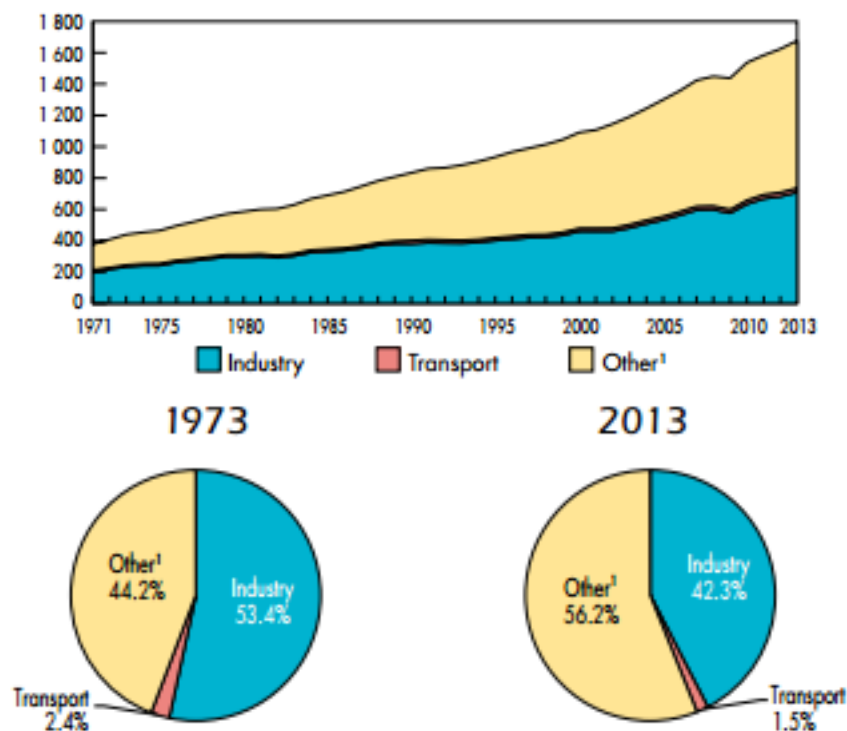


Figure 1.1 2013 Shares of world electricity consumption. (IEA, 2015)

Figure 1.1 shows that the industry is not the biggest consumer anymore, as other sectors have largely increased their consumption. Looking only at the EU, the EC made the same conclusions based on their annual data provided by each member state in their annual Eurostat. In the service sector the electricity consumption doubled and in the residential sector it increased by around 41%. The industry on the other hand was more influenced by the economic cycle. From 1994 until 2008 the energy consumption yearly decreased by 2.1% and in 2009 with 13.6%. In 2010 and 2011 there was a little growth of respectively 6.6% and 0.7%, but in 2012 and 2013 the consumption decreased again with respectively 3.0% and 0.7% (Eurostat, 2015).

Energy saving is clearly not only a must for companies, but a necessity for all energy consumers. Governments are creating more stringent environmental regulations, which makes energy-consumption more expensive and energy-saving more attractive.

One of the numerous ways to save energy is to install a monitoring-system.

In bigger scale grids, like the power-grid of a country, all users have a power analyser that sends the data to a monitor. The analysers are called smart meters and the developed system got the name Smart grid. The aim is to develop an energy network that can automatically monitor energy flows and adjust to changes in energy supply and demand accordingly. Throughout the day, energy prices change according to the demand and the production of energy at that exact moment. In this way, consumers will adapt their energy usage to different energy prices and save money by consuming more energy in low price periods. The smart grid also allows the grid operators to better plan the production of energy and balance their networks. The EC already published a report about the deployment of smart meters in 2014. It is expected that almost 72% of European consumers will have a smart electricity meter by 2020. The installation cost is between €200 and €250 and the EU is planning a placement of 200 million meters by 2020, which represents an investment of €45 billion. The EC expects that Portugal will have around 6.500 million electricity smart meter systems and Belgium around 5.975 million. This investment may save energy up to 3%, offering potential for load shifting and overall benefits in the order of €309 per meter point (EC, 2014).

In smaller scale grids, like companies, energy monitoring can be done by implementing analysers in the company's power-grid. This is an easy way to get an overview of the energy consumption of all company sections. The company is able to balance out their network and push their production to the very maximum if needed. In case of errors in the network, the measured values will indicate the cause of the failure. Energy monitoring can also help to better integrate renewable energy. If the company possesses wind turbines or solar panels, anemometers and solar analysing tools can be

added to the system. This way, the consumption of energy can be adjusted to the company's energy production, resulting in 'greener' and cheaper electricity.

The market for energy monitoring system has been booming for several years. Doing research about this topic will help to improve the current systems and gives great opportunities for the future.

This particular research focusses on creating a monitoring system for small scale grids that does not depend on any specific brand. The measurement devices are connected via a Modbus TCP/IP connection. This communication protocol is openly published and royalty-free, making it perfectly qualified for creating an open system. The software tool used is the Matlab/Simulink. This environment is able to create a transparent and comprehensible overview of the measured values. The software is easy to handle, even for beginners and makes it possible to make changes really fast.

1.2 Energy monitoring

Energy monitoring is a technique that was first implemented in the United Kingdom in 1980. It is a way of controlling energy-usage, it permits companies to get informed about the energy-consumption of their power grid in real-time, but may also predict how much energy they will consume under different circumstances. It allows making adaptations to increase the productivity and reduce energy waste.

How it works

First of all, it is necessary to group consumers, for example office, compressors, heating, lighting... . An electrical scheme of the company's power grid is indispensable for doing this. Second, secondary analysers must be determined, like temperature analysers for heating installation or sunlight-measurement devices for lighting installations. When all these variables are set, the necessary measurement devices can be installed at the right places and energy monitoring can start. The data retrieved by the measurement devices has to be captured and saved by the software.

The data will be plotted in a graph to define a base-line for the energy consumption. The consumption can also be used in a scatter plot against other variables like temperature or light-intensity. The best fit line is an image of the general energy consumption. The interception of this line with the y-axis gives an idea of the minimum energy consumption. If this is very high, there could be faults in the power grid causing unnecessary energy consumption. The slope gives an idea of the efficiency and the scatter gives an idea of variability.

The difference between measured values and expected values can now be monitored. Possible variations can be identified and the company is now able to set new goals for the future. This all has to be repeated at specific time points, resulting in adjustments that will lead to a lower energy-consumption as well as a lower energy-cost.

Following data retrieval from the measurement devices, the cycle starts all over again. This way, the company will keep improving and the newest technologies will be implemented.

1.3 Matlab / Simulink



Figure 1.2 Matlab logo
(MathWorks, 2016)

Matlab is an easy-to-use environment for numerical calculation and simulation. It is also usable for programming as it has its own programming language, called 'M-code'. The software is developed by THE MATHWORKS in 1984 and is commonly used in companies as well as academic institutions. Several extensions in the Matlab-software, called 'Toolboxes', make it possible to calculate and simulate problems without the need of writing Matlab code. This makes the software very easy to use, fast and inviting.

As mentioned previously, Matlab has several extensions, one of those 'Toolboxes' is called Simulink. In Simulink the user can create block diagrams for the simulation of divers problems. It is most of all used as a computer model for dynamical systems, but can also be used for complex math problems. The biggest advantage of Simulink is the user friendliness, it gives the opportunity to construct complicated models in a clean and transparent way and makes it possible for outsiders to quickly understand and adjust the program.

The used version for this thesis is Matlab R2015b.

1.4 Thesis organization

This thesis is organized in 5 chapters. This chapter is the introduction. The second chapter provides the information about the power analysers and the used communication protocol for connecting the power analysers with the software. Chapter three gives an overall look at the electrical grid, explains the components of the Simulink model and the Matlab code. In chapter four case studies where done to check the reliability of the program. Chapter five concludes the work and gives a word of appreciation to the people involved in this work.

2. Measurement devices and network

2.1 Power analyser



Figure 2.1 Janitza UMG 96RM EL (Janitza, 2016)

The power analysers used for this thesis are of the type JANITZA UMG 96RM EL. The analysers are suitable for measuring, recording and monitoring electrical parameters in low and medium voltage networks, 1 and 3-phase systems and with a neutral conductor. The device has a compact construction, high data security and a large measured data memory.

2.1.1 Specifications

Measurement accuracy

It has a reliable measurement accuracy of 0.2% (V) and effective energy class (kWh-class) of 0.5. The kWh-classes are defined and used in IEC and ANSI standards. kWh-class of 0.5% means that the power analyser has an accuracy better than $\pm 0.5\%$ of the nominal reading (Janitza, 2016). The max current is 5 A effective.

Communication options

Numerous interfaces and protocols, guaranteeing an easy system connection over Ethernet, Profinet, Modbus, M-Bus, ... (Janitza, 2016). In this project the Ethernet (TCP/IP) communication is used.

Harmonics analyser

It can measure up to the 40th harmonic and provides information about the power quality, grid disturbances and possible “network polluters”.

(Janitza, Power Analyser UMG 96 RM-EL, 2016)

2.1.2 Installation

The UMG 96RM-EL can be used for voltage measurement in TN, TT and IT systems. Voltage measurement in the UMG 96RM-EL is designed for the 300 V overvoltage category CATIII (4 kV rated pulse voltage). Figure 2.2 shows the principle circuit diagram of how the measurement device is connected for the voltage measurements.

The current measurement is done with use of 50/5 A current transformers (Figure 2.3).

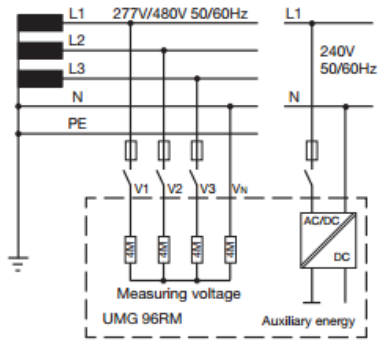


Figure 2.2 Connection diagram voltage measurement (Janitza, Operating instructions and technical data, 2016)

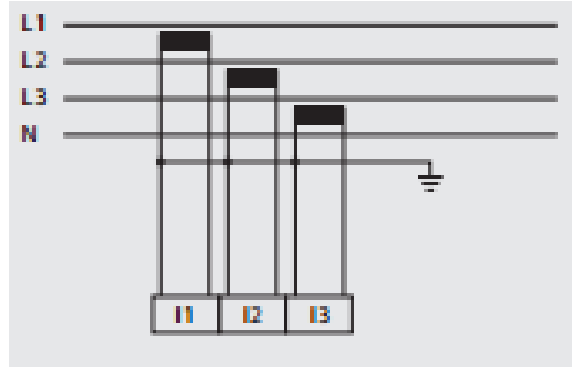


Figure 2.3 Connection diagram current measurement (Janitza, Operating instructions and technical data, 2016)

2.2 MODBUS

In 1979 the company Modicon developed a serial communication protocol for their programmable logic controllers (PLCs). From that moment this protocol was expected as the unofficial standard because of its simplicity and robustness. It offers communication between client /server on various networks or many different buses, it even is despite its old age still the standard today.

Nowadays MODBUS is still growing and is able to work with Ethernet and TCP/IP. The protocol is very easy to implement as it uses the same application layer for all the different physical layers, as shown in Figure 2.4.

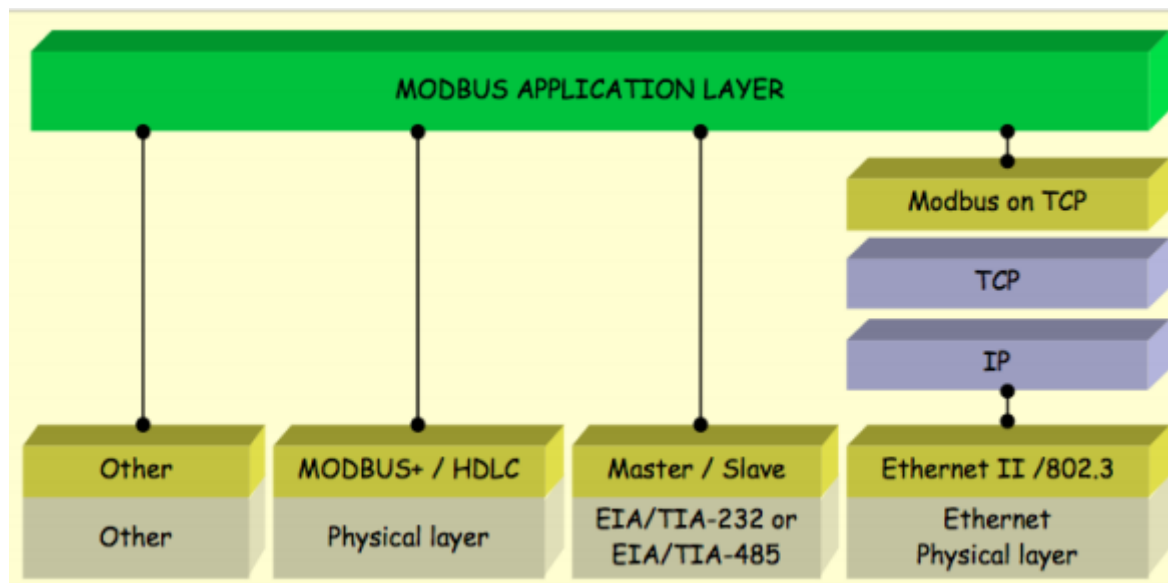


Figure 2.4 MODBUS (Modbus, 2012)

2.2.1 MODBUS Application Protocol

In the MODBUS protocol, the client is also known as the Master and the server as the slave. The client tells every server that he is allowed to connect if he has data to send. This principle is called the polling principle. The number of quarrels is negligible as the client is the only device in a MODBUS network to initialize a connection.

2.2.1.1 Protocol description

Figure 2.5 shows the frame used for MODBUS, existing of an Application Data Unit (ADU) and a Protocol Data Unit (PDU). An ADU contains an address, the PDU and an error check. The PDU is a frame containing function codes and the actual data that the client needs from the server.

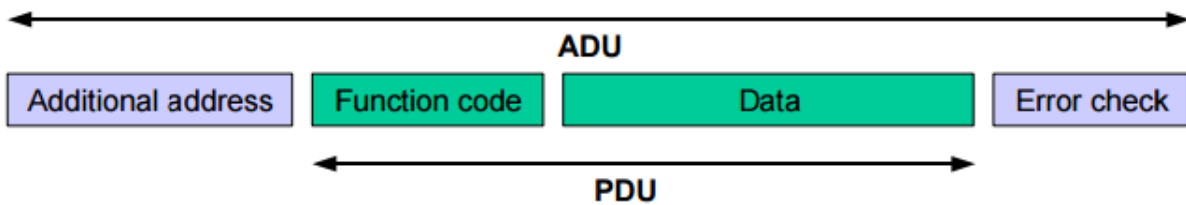


Figure 2.5 MODBUS Protocol description (Modbus, 2012)

When a client requests to connect with a server, the function code field consists of 1 byte and tells the server what kind of action it must perform. A value of “0” is not valid. Sometimes sub-function codes are added to define multiple actions. The data field contains, in this situation, additional information needed for the action that the server must execute. A value of “0” is valid for the data field, in this case the function code alone specifies the action and addition information is not required.

The server will send the requested data to the client and the server simply echoes the original function code (Figure 2.6). This only when there are no errors. When an errors does occur, the PDU will contain an exception code which matches the error (Figure 2.7).

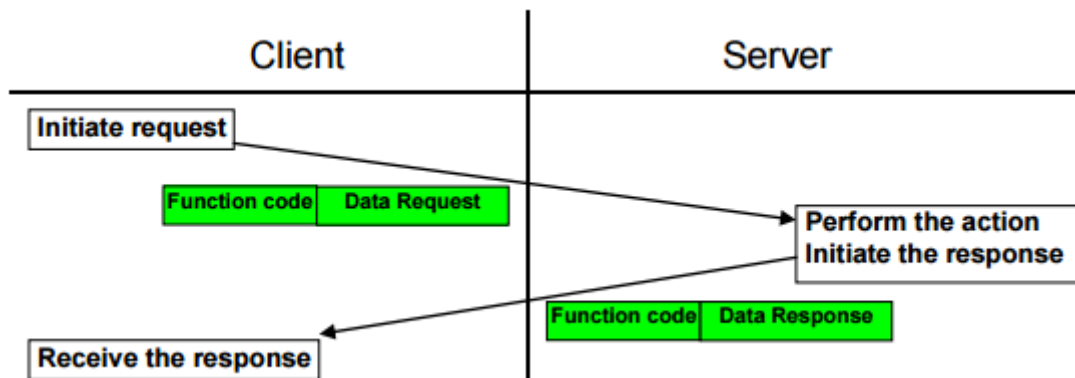


Figure 2.6 MODBUS Protocol without error (Modbus, 2012)

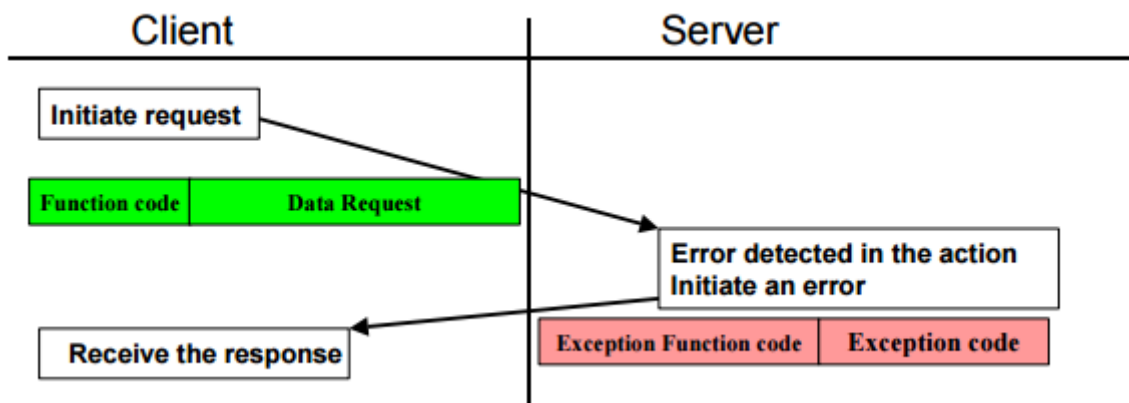


Figure 2.7 MODBUS Protocol with error (Modbus 2012)

2.2.1.2 MODBUS Data model

MODBUS uses a 'big-Endian' representation for addresses and data items. Meaning that the most significant byte is sent first. When a value of 0x7654 has to be send, 0x76 will be send first followed by the less significant bytes 0x54.

The model is based on a series of tables that have distinguishing characteristics. The four primary are:

Object Type	Access	Size
Coil	Read-Write	1-bit
Discrete Input	Read-Only	1-bit
Input Register	Read-Only	16-bits
Holding Register	Read-Write	16-bits

The user can only read data from the inputs but can read and write data to the holding register and the coil. The holding register is used to store numerical data (e.g.: numbers, words) and the coil is used for a single output (e.g.: open or close a relay).

The user needs to implement different function codes for all four blocks. For example the function code (0x02) is needed to write data to the discrete input and code (0x05, 0x0F) could be used to write data to the coils. (Figure 2.8)

It is possible to consider all 4 tables as overlaying one another, if this is the best way of interpreting the target machine in question. In this case the user can implement the function code (0x04) to get the statuses for the whole register or he can ask just for one bit in the register using the function code (0x02) for the read discrete input. (Figure 2.9)

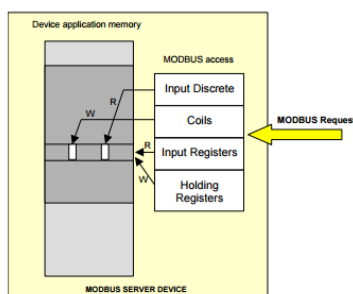


Figure 2.8 Implement function code (Modbus, 2012)

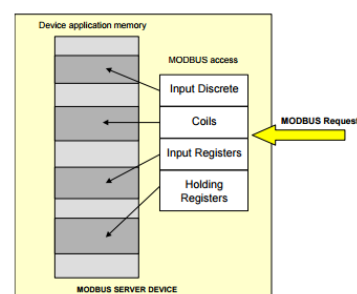


Figure 2.9 Implement function code (Modbus 2012)

2.2.1.3 Function codes

The MODBUS function codes are divided into 3 categories (Figure 2.11): public function codes, user-defined function codes and reserved function codes. The code requires 8 bits (1 byte) of memory in

the PDU. Decimal codes in the range 1 ... 255 are valid, but those in the range of 128 ... 255 are reserved for exception responses.

The public function codes are validated by the MODBUS.org community and therefore guaranteed to be unique. The codes are well defined and publicly documented. (Figure 2.10)

				Function Codes		(hex)	Section
				code	Sub code		
Data Access	Bit access	Physical Discrete Inputs	Read Discrete Inputs	02		02	6.2
		Internal Bits Or Physical coils	Read Coils	01		01	6.1
			Write Single Coil	05		05	6.5
	Write Multiple Coils		15		0F	6.11	
	16 bits access	Physical Input Registers	Read Input Register	04		04	6.4
			Read Holding Registers	03		03	6.3
		Internal Registers Or Physical Output Registers	Write Single Register	06		06	6.6
			Write Multiple Registers	16		10	6.12
			Read/Write Multiple Registers	23		17	6.17
			Mask Write Register	22		16	6.16
		Read FIFO queue	24		18	6.18	
		File record access	Read File record	20		14	6.14
	Write File record		21		15	6.15	
	Diagnostics	Read Exception status		07		07	6.7
		Diagnostic		08	00-18,20	08	6.8
Get Com event counter		11		0B	6.9		
Get Com Event Log		12		0C	6.10		
Report Server ID		17		11	6.13		
Read device Identification		43	14	2B	6.21		
Other	Encapsulated Interface Transport		43	13,14	2B	6.19	
	CANopen General Reference		43	13	2B	6.20	

Figure 2.10 MODBUS Public Function Codes (Modbus, 2012)

The User defined function codes can be applied by anyone and are not supported by the specifications. There is no guarantee that this code is unique and therefore it could already be implemented by another user. Two ranges are assigned to this code, in particular range 65 to 72 and 100 to 110. (Figure 2.11)

The reserved function codes are used by companies for legacy products and are not available for public use.

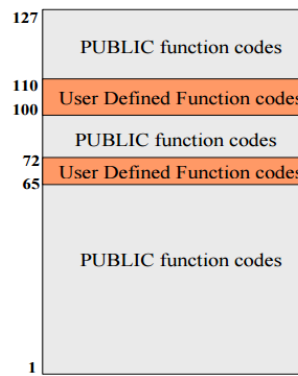


Figure 2.11 Modbus function code categories (Modbus, 2012)

As an example of how a MODBUS PDU message is assembled, the reading of holding register from a remote device will be fully explained. The request message to the server is assembled as followed:

the first byte is the function code, in this case 03 (hexadecimal), the following 2 bytes are the starting address and the last 2 bytes are the quantity of holding registers. The starting address starts at “0”, therefor numbers 1 to 16 are addressed as 0 to 15.

The response’s first byte consists of a function byte equally to the one in the request. The second byte gives information about the byte count, which specifies the quantity of complete bytes of data. The remaining bytes are the register value which contains N x 2 bytes, with N defined as the quantity of bytes that the byte count register specifies.

Example: Request to read holding registers 108-110 (Figure 2.12)

Request		Response	
Field Name	(Hex)	Field Name	(Hex)
Function	03	Function	03
Starting Address Hi	00	Byte Count	06
Starting Address Lo	6B	Register value Hi (108)	02
No. of Registers Hi	00	Register value Lo (108)	2B
No. of Registers Lo	03	Register value Hi (109)	00
		Register value Lo (109)	00
		Register value Hi (110)	00
		Register value Lo (110)	64

Figure 2.12 Example to read holding registers 108-110 (Modbus, 2012)

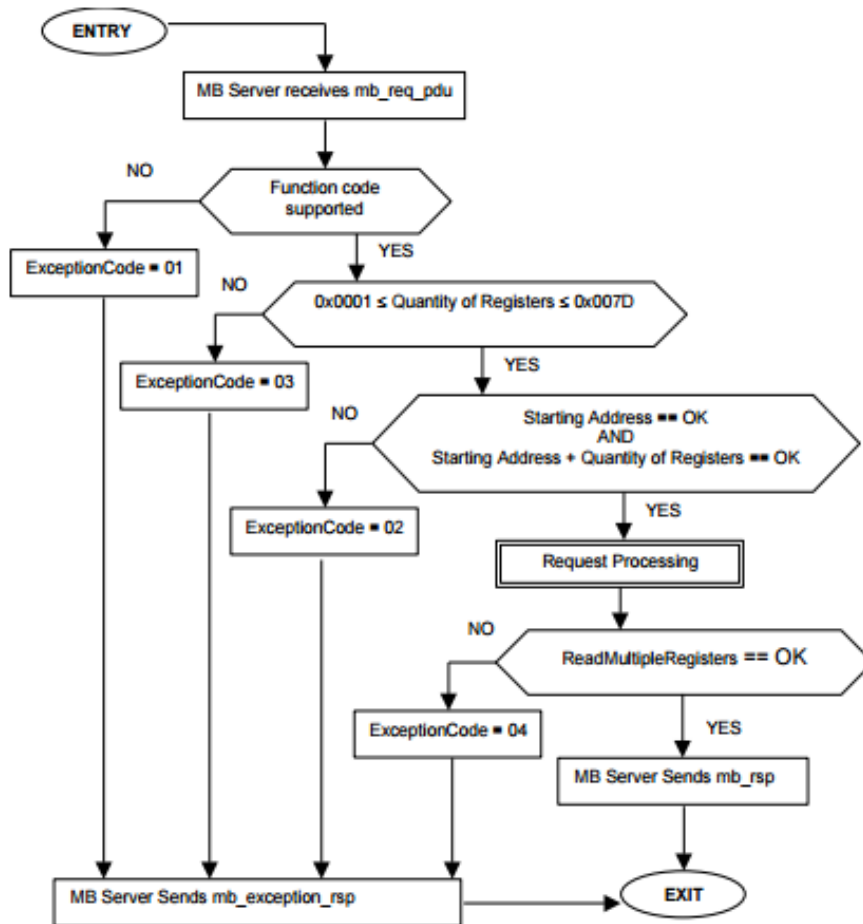


Figure 2.13 Function for reading the holding register (Modbus, 2012)

Figure 2.13 shows a read holding register function. The server receives a request from a MODBUS client and checks if the server supports the send function code. If the code is not valid, an exception code is send. If it's valid, 3 more checks are done. If all are good, a response message is send to the client.

2.2.2 MODBUS TCP/IP

Modbus TCP/IP or Modbus TCP is a Modbus variant used for communications over TCP/IP networks, connecting over port 502. The client/ server model is based on 4 types of messages (Figure 2.14).

1. MODBUS request: the message sent on the network by the client to initiate a transaction.
2. MODBUS Indication: The request message received on the server side.
3. MODBUS Response: The response message sent by the server.
4. MODBUS Confirmation: The response Message received on the client side.

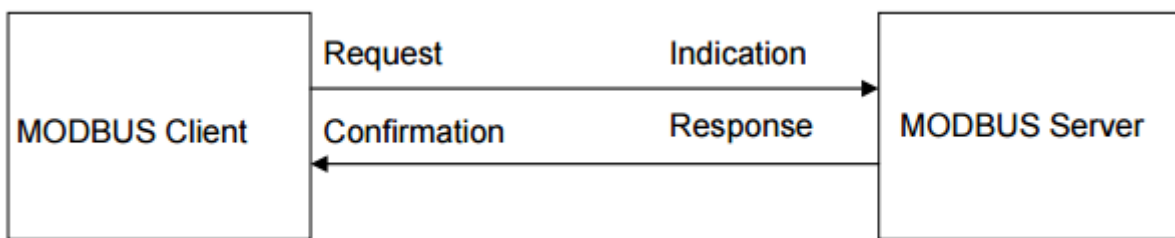


Figure 2.14 Types of messages in TCP/IP networks (Modbus, 2006)

2.2.2.1 MODBUS application protocol header

The ADU for a MODBUS TCP/IP network is different than the ADU used with MODBUS over serial line. The address and the error check are replaced by the MODBUS application protocol header (MBAP header)(Figure 2.15)(Figure 2.16).

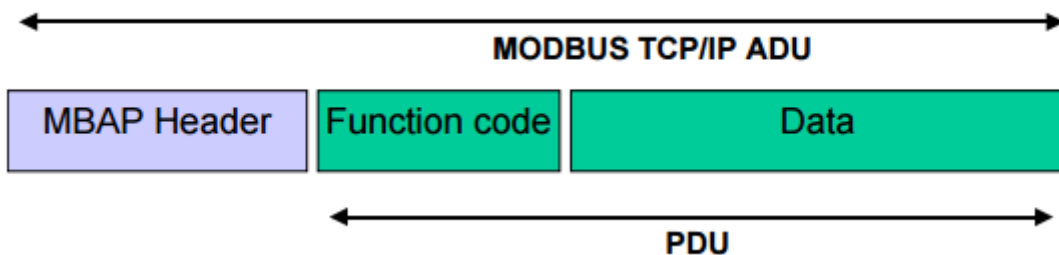


Figure 2.15 MODBUS application protocol header (Modbus, 2006)

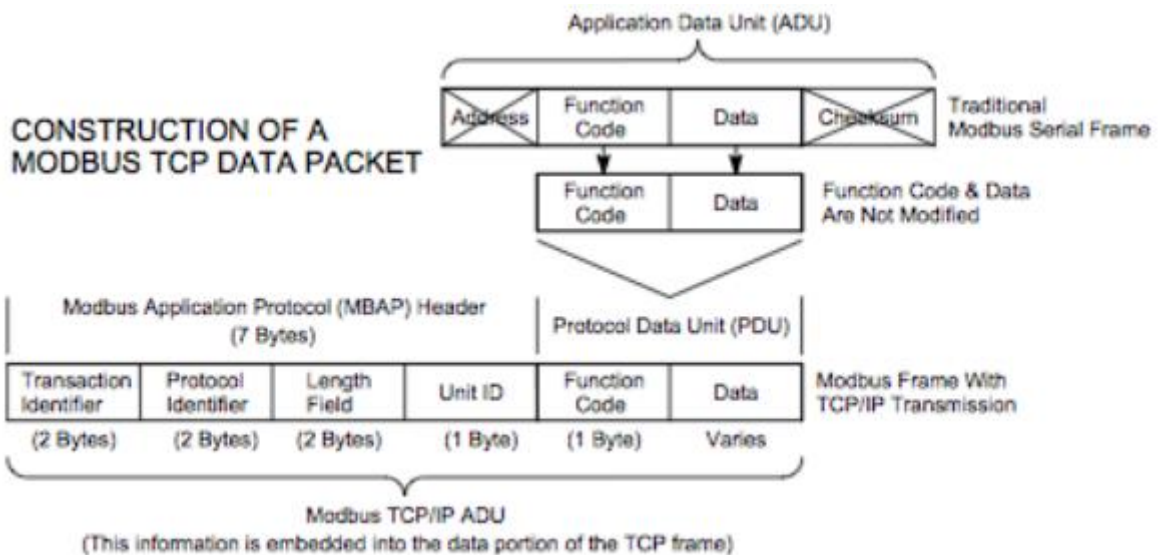


Figure 2.16 MODBUS application data unit (Modbus, 2006)

The Transaction Identifier determines a MODBUS request or response. It is initialized by the client and recopied by the server. The identifier must be unique, as it is used to associate the future response with the request.

The Protocol Identifier is used for intra-system multiplexing (Transmit a number of separate signals simultaneously over a single channel or line). This is always "0" as there is always only 1 signal transmitted. This field is also recopied by the server from the received request.

The Length defines the number of following bytes, including the Unit Identifier and the PDU. The field is initialized by the server for the response.

The Unit Identifier carries the MODBUS slave address of the remote device if the device is part of a MODBUS serial line sub-network. With TCP/IP the server is addressed with an IP address, therefore the Unit Identifier is useless and the value must be set to 0xFF. The field is set by the MODBUS client in the request and must be returned with the same value in the response by the server.

Example: MODBUS request ADU encoding for reading the register #5 in a remote server.

MODBUS Request ADU encoding :

	Description	Size (bytes)	Example
MBAP Header	Transaction Identifier Hi	1	0x15
	Transaction Identifier Lo	1	0x01
	Protocol Identifier	2	0x0000
	Length	2	0x0006
	Unit Identifier	1	0xFF
MODBUS request	Function Code (*)	1	0x03
	Starting Address	2	0x0004
	Quantity of Registers	2	0x0001

Transaction Identifier is randomly chosen as 0x1501, the only condition is that the client as well as the server uses the same value for the request and the response. The Protocol Identifier is "0" as always in TCP/IP communication. The Length is here "6" this because there are 6 bytes needed for the following fields. 1byte Unit Identifier, 1 byte Function Code, 2 bytes Starting Address and 2 bytes Quantity of Registers. The Unit Identifier is always "0xFF" for TCP/IP. The Function Code is here 0x03 as it is a request to read the holding registers (see Figure 2.10). The Starting Address is "0x004" because the intension is to read register "5". As the count of registers starts with "0", meaning that register 1-5 becomes address 0-4. The quantity of Registers is "1" as there is only 1 register requested.

3. Monitoring an electrical grid

3.1 Electrical grid

3.1.1 Location

The electrical grid used to test this project is situated in one of the laboratories of the GECAD research group.

GECAD, Research Group on Intelligent Engineering and Computing for Advanced Innovation and Development, is a research unit settled in the Institute of Engineering - Polytechnic of Porto (ISEP/IPP) having as mission the promotion and development of scientific research in the Knowledge and Decision Sciences domains, having Information Technologies as support. It involves 2 main areas: Intelligent Systems and Power Energy Systems. GECAD is known worldwide in its areas of research, leading some research domains. (GECAD,2016)

3.1.2 Details of the grid

Figure 3.1 shows the used electrical grid from the laboratory in GECAD. The grid contains four 3P+N busses and one 3F+N connection with the public grid. Nine power analysers are installed all of them are of the type JANITZA UMG 96RM-EL. 3 loads are installed that have the same specifications as the most common cables used in industrial power grids.

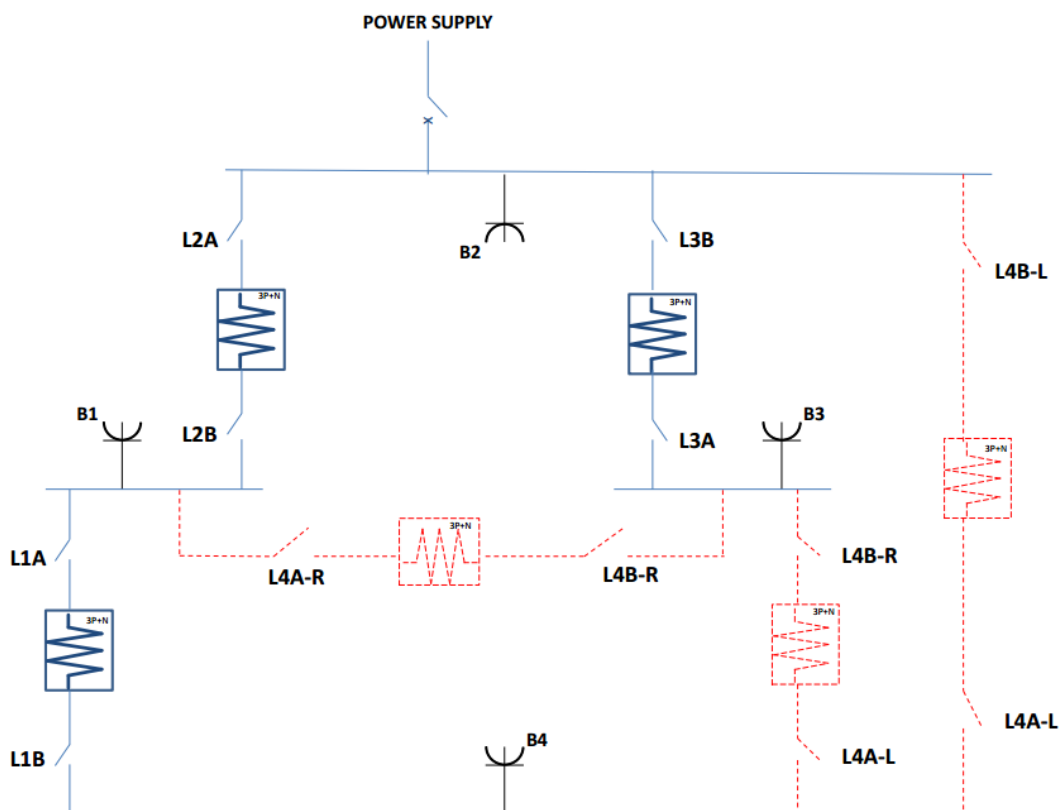


Figure 3.1 Electrical grid of the GECAD laboratory

3.2 Monitoring an electrical grid

In this chapter the entire created solution is presented in an overview and the parts are explained in detail. As it is not easy to give a full explanation of a Simulink model, nether trying to understand it, without using it. Dividing the program in a lot of small chapters seemed to be the best approach. The explanation starts with analysing the overall model and all its components in chapter 3.2.1. These components have underlying layers of sub-models called subsystems. The chapters 3.2.2, 3.2.3, 3.2.4, 3.2.5 and 3.2.6 each explain such a subsystem. These subsystems contain some sub sub-models, therefor these chapters contain smaller chapters.

3.2.1 Overall Model

The Simulink model used for monitoring the electrical grid is constructed out of several subnetworks. Figure 3.2 shows the first view of the model when opened. The model is divided in 4 big groups of subsystems. The “Meter” blocks that request and receive the demanded values from the power analysers. The collar of this blocks is red. The orange “Bus” blocks contain the interfaces and mathematical calculations needed for measuring all what’s interesting on electrical domain for the loads connected to the busses in real time. The “Cable” blocks, the green blocks, do the same as the “Bus” blocks, but for the cables instead of the busses. The last group, contains the “Switchable network configuration” blocks. These blocks make it possible to easily adapt to changes to the power grid, keeping the configuration of the Simulink model the same as that of the power grid. The blocks are visible in the upper right corner of Figure 3.2 and are recognizable as switches. Implementing changes In the next chapters the subnetworks will be explained separately.

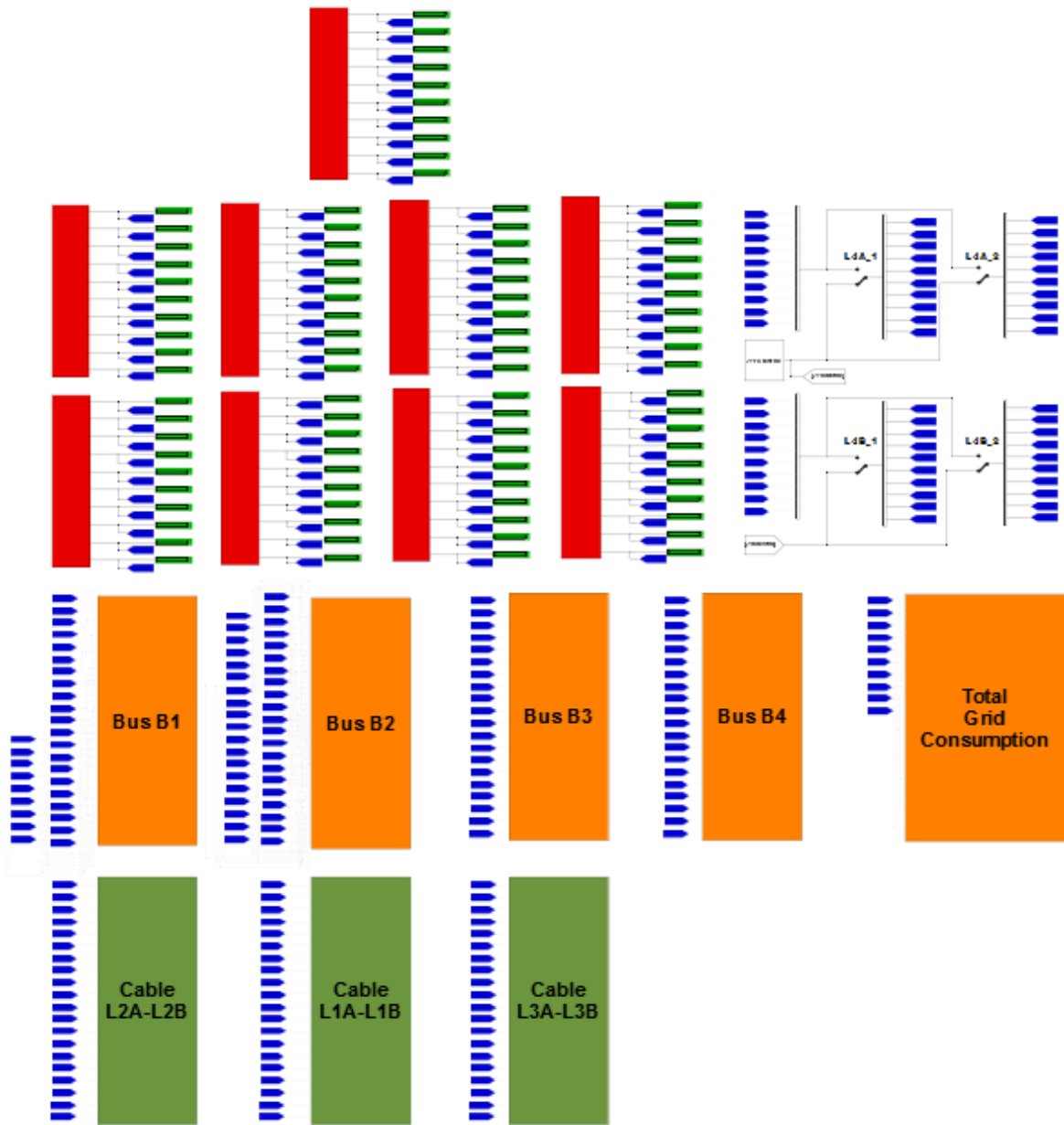


Figure 3.2 Simulink model first view

3.2.2 Power Analyser

Each power analyser has his own subnetwork which contains a TCP/IP receive block. The values are then send via Goto-blocks to the corresponding Bus-subnetwork for further calculations. The Figure 3.3 contains the subnetwork, shown in red, the blue Goto-block and a light-green display for making the received corresponding value visible.

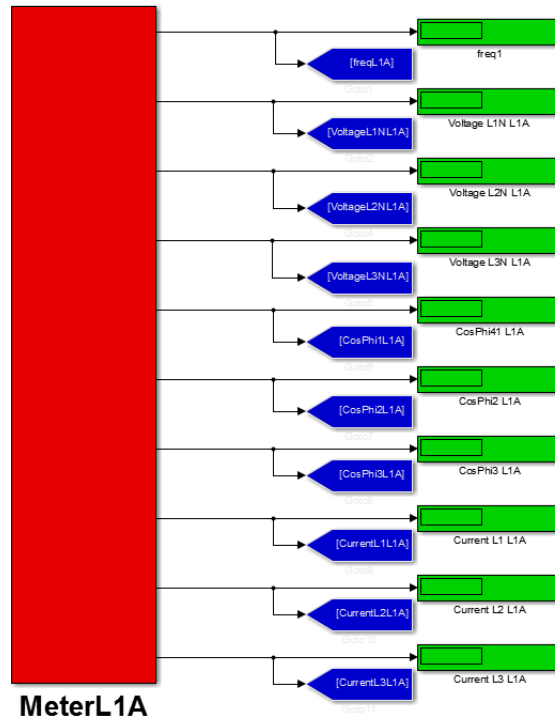


Figure 3.3 Simulink model power analyser

3.2.2.1 Power analyser subnetwork

With a double-click on the red block, the underlying subnetwork is made visible as shown in Figure 3.4. This network contains a 'Constant Value block', 10 'Outport blocks' to send the value from the subsystem to the parent, a 'manual switch block', a 'matrix block' and an 'Interpreted Matlab Function block'. The 'matrix block' and the 'manual switch block' are used when the TCP/IP connection fails with the meter. The program can still run, because the user can define the values manually in the matrix. The consequences are that busses relying on this meter can't be measured in real time, but it is the only way that the other busses still can. The last block ('Matlab Function Block')(Figure 3.5) is the most critical of the entire system. It contains the Matlab-code needed for the Modbus TCP/IP communication with the power analyser.

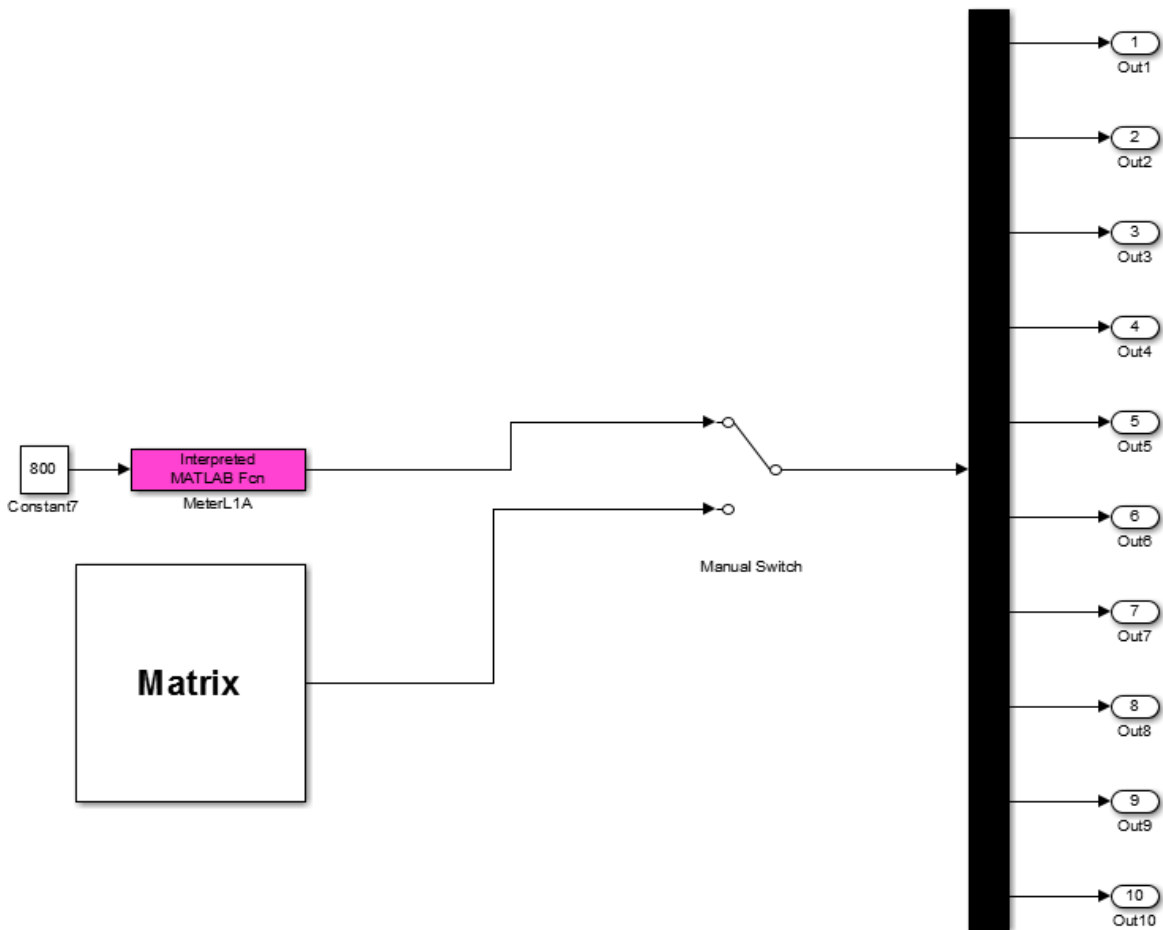


Figure 3.4 Simulink model power analyser subsystem

Why not use a ‘TCP/IP Receive block’ (Figure 3.6) instead of writing a total new code? The code in this block has been developed specially for this project, because a normal TCP/IP Receive block does not support every brand of power analyser. The block can only function if there is a support block implemented. These blocks are downloadable from the brand’s website, but there is no support block for a JANITZA power analyser. The Matlab code used in “Interpreted Matlab Function block” is fully explained in chapter “3.3 Matlab code for the MODBUS TCP/IP connection”.

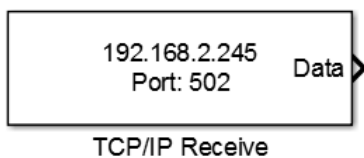


Figure 3.5 TCP/IP Receive block (MathWorks, 2016)

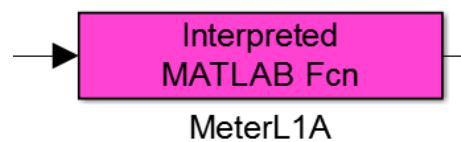


Figure 3.6 Interpreted Matlab Function block

3.2.3 Switchable network configuration

The power analysers L4A and L4B are implemented after switches. These switches make it possible to change the configuration of the network if necessary. See “3.1.2 Details of the grid / Figure 3.1”. For example measurement device L4A can measure the current of the network configuration if the switch is implemented as L4A_1, L4A_2 or open. In each of the three configurations the power analyser will measure different currents. Therefore the Simulink model must run in the same configuration as the power grid. So if meter L4A is implemented as L4A_1 in the power grid, the Simulink block “manual switch L4A_1” must transfer the values of “meter L4A”. The “Manual switch L4A_2” must transfer the “Zero-Matrix”, because this network is open and therefore all the measured values are “0”. Figure 3.7 shows a part of the Simulink model where these configuration can be made.

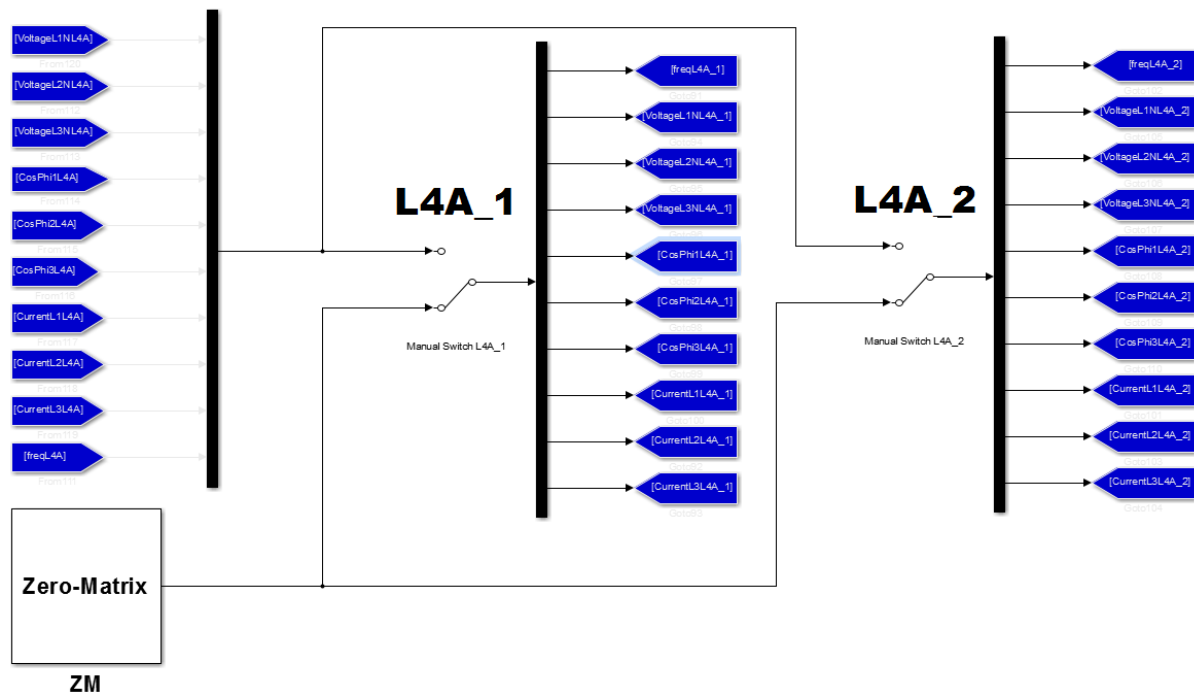


Figure 3.7 Simulink model switchable network configuration

3.2.4 Real time simulation platform

The most interesting places to measure are the busses. Every bus has the same layout and the same construction, explaining one random bus should be sufficient to understand all. For the visibility the subnetworks are all named after the busses. For example bus B1 (Figure 3.8) has as imputes the measured values of power analysers “Meter L1A”, “Meter L2B” and “Manual Switch L4A_2”. This because the bus is situated between power analyser L1A, L2B and L4A configuration 2 (L4A_2), see “3.1.2 Details of the grid / Figure 3.1”.

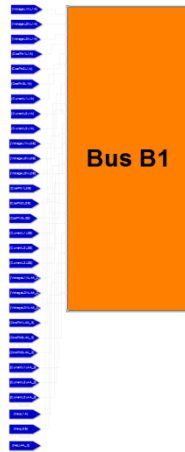


Figure 3.8 Simulink model bus B1

3.2.4.1 Bus subnetwork

By double clicking on the subnetwork B1 Figure 3.9 becomes visible. The subnetwork contains 22 displays that visualize all the values interesting for measuring a power grid, one scope pictures the currents, one the powers and one scope (the dark-green one) gives an overview of the three previous ones.

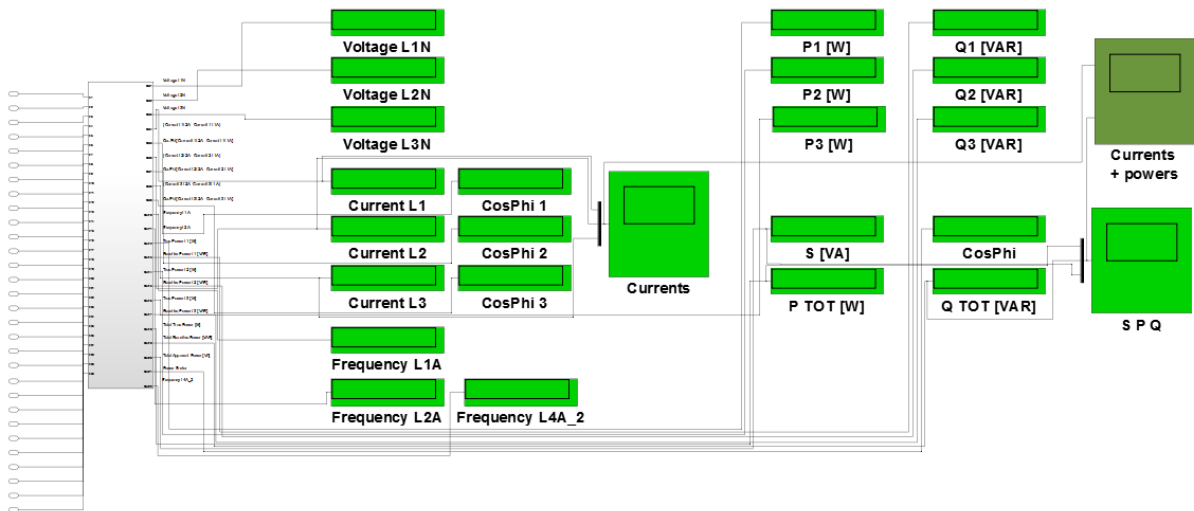


Figure 3.9 Simulink model bus B1 subnetwork

The big advantage of using Simulink is that the user can simply adjust the model and measure all the things he wants. For example: By implementing an extra scope and connecting it with a current, a graph of the current and time can be created. This graph is visible during the simulation and can provide important information.

3.2.4.2 Mathematical calculations subnetwork

In this subnetwork there is another subnetwork hiding the mathematical calculations. As the user doesn't need to see this, hiding them improves the visibility of the Simulink model. Figure 3.10 visualises the subnetwork.

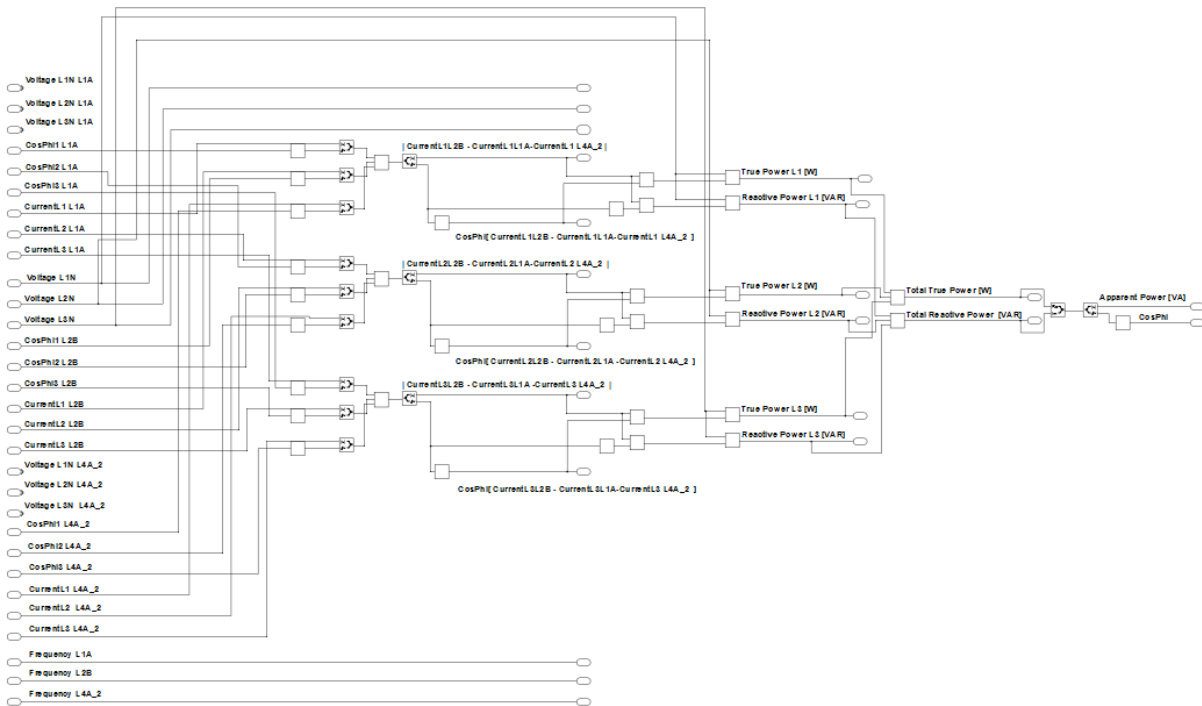


Figure 3.10 Simulink model bus B1 calculations

In the middle three identical calculation are made, in particular the calculations for the current in each Phase. See Figure 3.11 for the calculation of the current in the first phase (L1) of bus B1, the calculations for phase 2 and 3 or similar. The current flowing through bus B1 is the difference of the current in power analyser L2B and power analysers L1A, L4A_2. This can be easily verified by watching “3.1.2 Details of the grid / Figure 3.1”. The calculations have to be performed with complex numbers as the currents are complex.

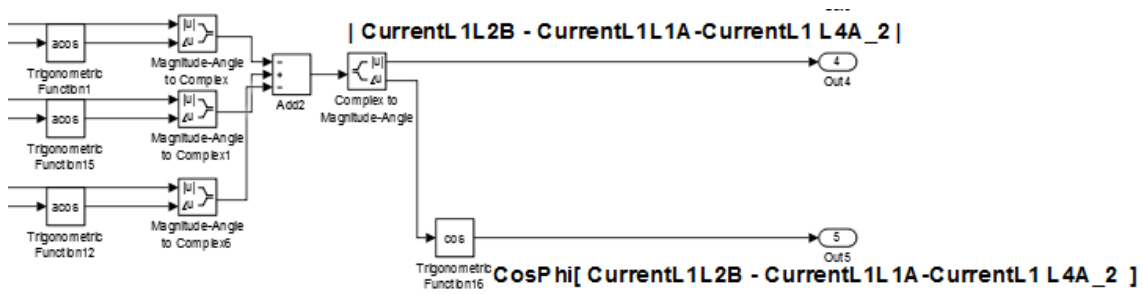


Figure 3.11 Calculation currents

The next step is to calculate the true power and reactive power in every line, The total true and reactive power, the total apparent power and the total $\cos(\varphi)$. (Figure 3.12)

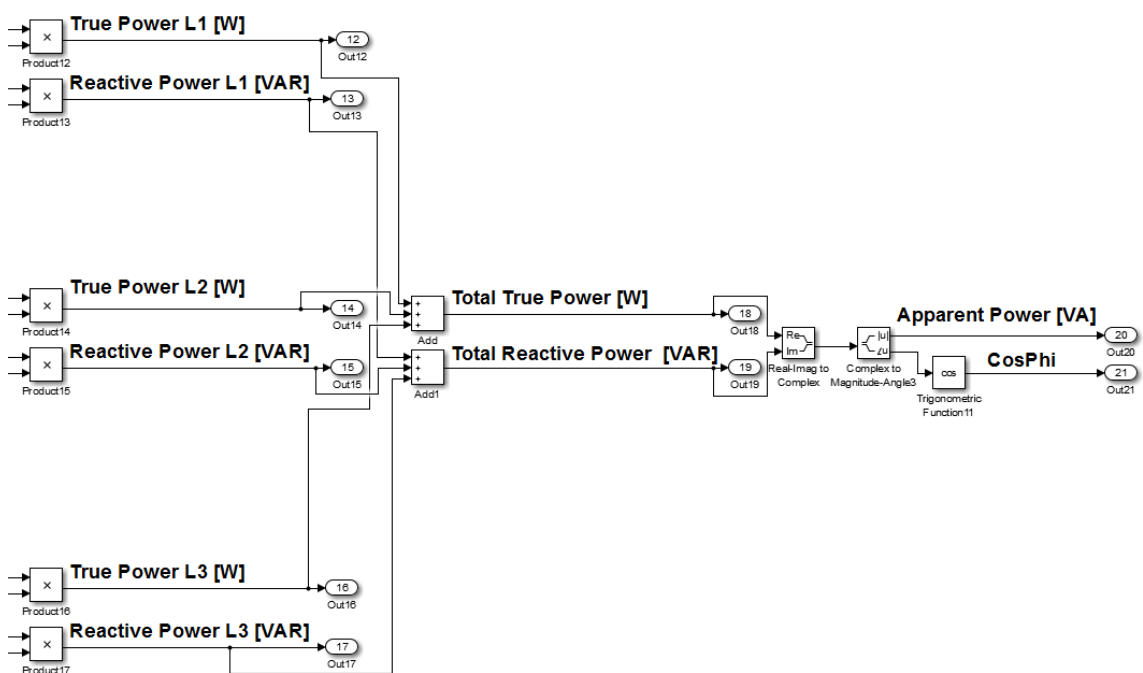


Figure 3.12 Calculation powers

3.2.5 Other busses

It is more than logic that all the busses have different configurations of the blocks. The displaying of the measured values are all identical to Figure 3.9, but the “Mathematical calculations subnetwork” differs. The “Total Grid Consumption” block uses the measured values of “Meter Power Supply” to calculate the powers and currents for the total power grid.

Mathematical calculations subnetwork for bus B2

“3.1.2 Details of the grid” clarifies that the current flowing through bus B2 is the current coming from the power supply reduced with the current flowing through meter L2A, L3B and L4B_1.

$$I_{B2} = I_{PS} - I_{L2A} - I_{L3B} - I_{L4B_1}$$

Mathematical calculations subnetwork for bus B3

The current flowing through bus B3 is the difference of the current in meter L3A and meter L4B_2.

$$I_{B3} = I_{L3A} - I_{L4B_2}$$

Mathematical calculations subnetwork for bus B4

Bus B4 has a current that depends on the configuration of the network. In normal circumstances it is directly connected to meter L1B, but if necessary the power can also be delivered from the lines connected to meter L4A_1. As there is no meter after this bus, the voltage drops can't be measured here.

$$I_{B4} = I_{L1B} \text{ or } = I_{L4A_1}$$

Important for the measurements in this bus is that the connected line is always specified in the Simulink model. Therefore a switch block is used, giving the model displayed in Figure 3.13.

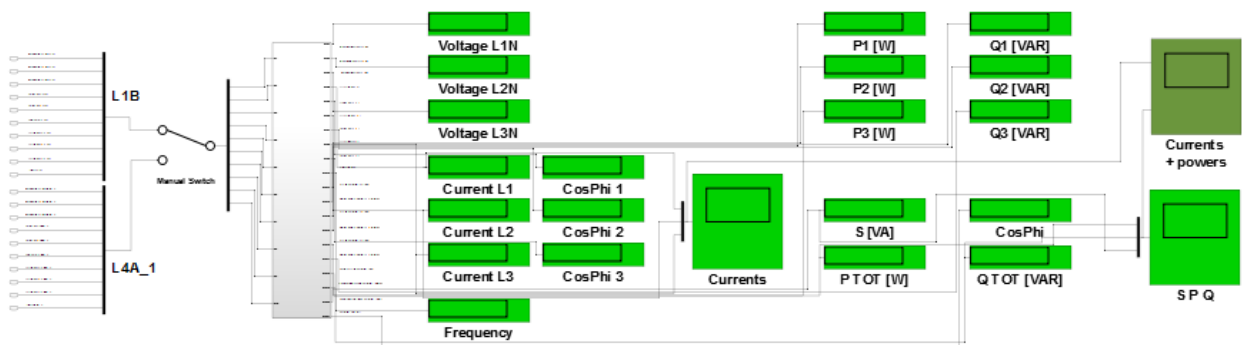


Figure 3.13 Simulink model bus B4 subnetwork

3.2.6 Cable L2A-L2B

The cables used for the energy transportation also consume energy and therefore cause a voltage drop. How longer the line, how bigger its load is and how bigger the voltage drop is. Measuring this can help understanding why the energy consumption is how it is. There are three cables in the network each surrounded by two power analysers. The first one is the cable between meter L2A and L2B, the second between meter L1A and L1B and the third between meter L3B and L3A. The displaying of the measured values are identical to those of the busses. Figure 3.14 shows the “Cable” block for the cable between meter L2A and L2B. The other two cables look identical.

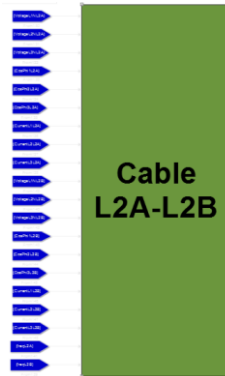


Figure 3.14 Simulink model Cable L2A-L2B

The Voltage drops over the cable can be calculated by decreasing the voltages measured by the two meters. Therefore an extra part has to be implemented in the calculation sub network. This part is presented in figure 3.15.

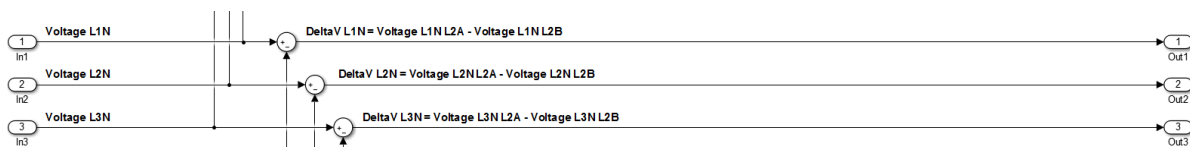


Figure 3.15 Simulink model Cable L2A-L2B calculations

3.3 Matlab code for the MODBUS TCP/IP connection

In chapter “3.2.3 Power analyser subnetwork” the Simulink block “Interpreted Matlab Function block” has been introduced. This block is used to connect the Simulink model with the Matlab file used for the MODBUS TCP/IP connection. The Simulink block transfers a value of “800” to the Matlab file, the Matlab file runs his code and afterwards an array of 10 values is send to the Simulink model. The array contains values that where measured by the power analysers and transferred to the user using a MODBUS TCP/IP connection. (Figure 3.16)

The complete Matlab code can be found in the appendix on the last pages of this work. The flow chart of the Matlab code is visualised in Figure 3.18.

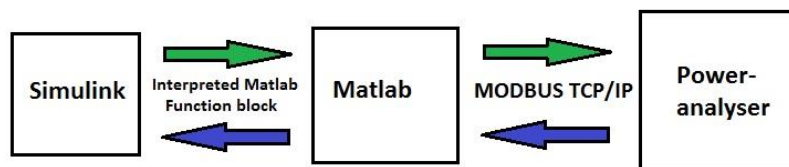


Figure 3.16 Connection Simulink / Matlab / Power analyser

3.3.1 Communication between Matlab and the power analyser

Configure the port (line 2-8):

The IP address of the analyser where the data is retrieved from is in this case meter L1A. By default, the TCP/IP object has an Inputbuffersize of 512, which means it can read 512 bytes at a time. The object doesn't have to be this big, but leaving it on the default value is easier and doesn't cause any consequences. MODBUS TCP/IP always uses port 502 and a 'big-Endian' representation for addresses and data items.

Open the port (line 9-16):

An error occurs when the program fails to connect with the power analyser.

Send message (line 17-35):

If the connection is a success, a message can be send:

Transaction ID: This can be any arbitrary number. Typically the value is incremented by 1 every time a message is send. The power analyser will mirror this number in its response to specify which request is answered, see chapter “2.2.1 MODBUS application protocol header”.

Protocol ID: This should be all zeros to indicate Modbus TCP.

- Length:** The Length is here “6” this because there are 6 bytes needed for the following fields.
- Unit ID:** The Unit Identifier is always “0xFF” for TCP/IP in theoretical models. But this didn’t work in the lab of GECAD. By testing the discovery of other working values was done. with the value “0”, the connection was easily made. A reason for the malfunctioning with a unitID of “0xFF” (Hex) or “255” (Dec), could be that this value is reserved for communication with a particular device in the subnetwork.
- Function Code:** This is a number that corresponds to a specific function code that instructs the device what to do. In this case the goal is to read the holding registers of the power analyser and therefor the value is 3 (see chapter 2.2.1.3 Function codes).
- Add:** The address of the starting register is set by the value that the Simulink model transferred at the beginning of the run. For this project it is always “800” as it’s the first register needed. Information about the registers of the Janitza UMG 96RM-EL power analyser is available in the product datasheets. Figure 3.17 shows a fragment of the datasheet with all the information needed for assigning the registers. (Janitza electronics GmbH, Power Analyser UMG 96RM Modbus-address list and Formulary).
- Val:** Here all the registers needed for the simulation are chosen. The last register needed is “864: Current I3 L3”. This means there are 65 registers needed. Every register is 2 bytes so the reading starts at register 800 and contains 130 bytes of information.

The message is now composed and the program can write it to the device (line 37).

Response (line 38-40):

The program will first wait for the response using a while-loop. When the response is available the while-loop is ended and the program continues running. The received data is than saved as a variable “res”.

Retrieve the demanded values from the response (line 42-51):

10 values have to be retrieved from the meters to have all the information requested for the simulation. Consequently the frequency (registers 800-801), the RMS-values (Root Mean Square value) of the line-neutral voltages (registers 808-813), the $\cos(\varphi)$ or fundamental power factor (registers 820-825) and the RMS-values of the currents of the three phases (registers 860-865). The demanded values are retrieved from the response by searching the corresponding bytes. The value of the first register will start at byte 10 and every value takes 4 bytes. Meaning that the demanded values are found as followed.

Modbus address list

Measured values, type float

Address	Format	RD/WR	Unit	Note	Index
800	FLOAT	RD	Hz	Measured frequency	
802	FLOAT	RD	-	Voltage, zero sequence	
804	FLOAT	RD	-	Voltage, negative sequence	
806	FLOAT	RD	-	Voltage, positive sequence	
808	FLOAT	RD	V	Voltage U1 L1-N	[0]
810	FLOAT	RD	V	Voltage U2 L2-N	[1]
812	FLOAT	RD	V	Voltage U3 L3-N	[2]
814	FLOAT	RD	V	Voltage U1 L1-L2	[0]
816	FLOAT	RD	V	Voltage U2 L2-L3	[1]
818	FLOAT	RD	V	Voltage U3 L3-L1	[2]
820	FLOAT	RD	-	Fund. power factor, CosPhi; ULN, IL1	[0]
822	FLOAT	RD	-	Fund. power factor, CosPhi; ULN, IL2	[1]
824	FLOAT	RD	-	Fund. power factor, CosPhi; ULN, IL3	[2]
826	FLOAT	RD	-	Sum; CosPhisum3=P0sum3/Ssum3	[3]
828	FLOAT	RD	-	Power factor; UL1N, IL1	[0]
830	FLOAT	RD	-	Power factor; UL2N, IL2	[1]
832	FLOAT	RD	-	Power factor; UL3N, IL3	[2]
834	FLOAT	RD	-	Sum; Power factor sum3=Psum3/Ssum3	[3]
836	FLOAT	RD	%	THD, U L1N, bezogen auf U0 L1	[0]
838	FLOAT	RD	%	THD, U L2N, bezogen auf U0 L2	[1]
840	FLOAT	RD	%	THD, U L3N, bezogen auf U0 L3	[2]
842	FLOAT	RD	%	THD, U L1L2, bezogen auf U0 L1L2	[0]
844	FLOAT	RD	%	THD, U L2L3, bezogen auf U0 L2L3	[1]
846	FLOAT	RD	%	THD, U L3L1, bezogen auf U0 L3L1	[2]
848	FLOAT	RD	V	Voltage, real part U1 L1N	[0]
850	FLOAT	RD	V	Voltage, real part U2 L2N	[1]
852	FLOAT	RD	V	Voltage, real part U3 L3N	[2]
854	FLOAT	RD	V	Voltage, imaginary part U L1N	[0]
856	FLOAT	RD	V	Voltage, imaginary part U L2N	[1]
858	FLOAT	RD	V	Voltage, imaginary part U L3N	[2]
860	FLOAT	RD	A	Current I1 L1	[0]
862	FLOAT	RD	A	Current I2 L2	[1]
864	FLOAT	RD	A	Current I3 L3	[2]

Figure 3.17 MODBUS address list for Janitza UMG 96RM-EL (Janitza, Power Analyser UMG 96 RM-EL , 2016)

3.3.2 Convert value from float to decimal.

The values in the response are all in data type float, this means they have to be converted to integer for making them readable for all users and to make mathematical calculations with them. Luckily the internet is full of examples for writing this particular Matlab code and simply copy one of them would be sufficient. The code used in this work (line 54-78): Roger Stafford, Decoding Floating point (IEEE-754), http://nl.mathworks.com/matlabcentral/newsreader/view_thread/132795, 20 Sep 2006.

For converting all the other values the same code is used, with different variables. (line 79-272)

3.3.3 Closing port and communication between Matlab and Simulink

The port is closed to end the MODBUS TCP/IP connection and last but not least, all the generated values are send to the Simulink model for further implementation. (line 274-278)

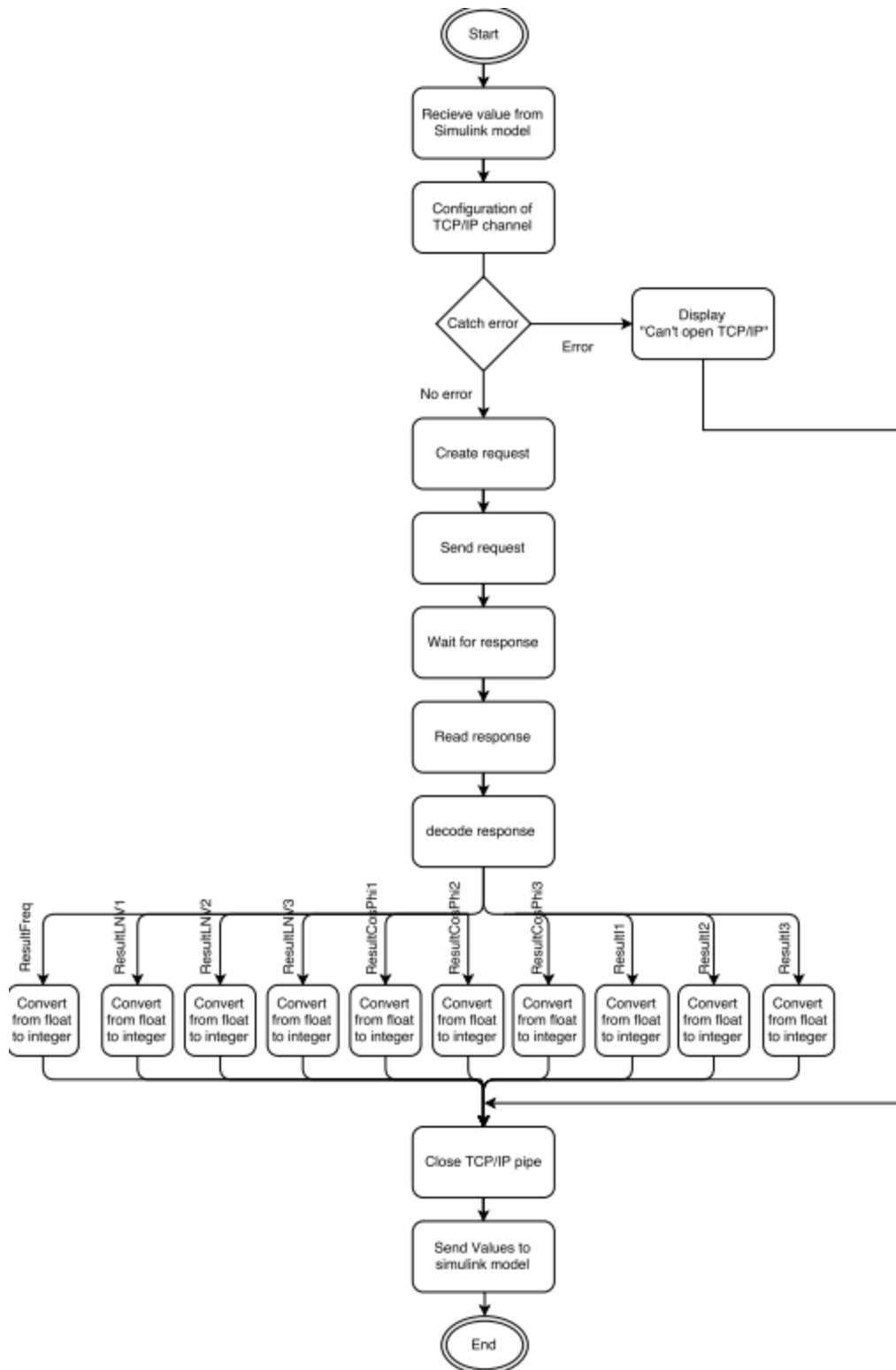


Figure 3.18 Flow chart of Matlab code the MODBUS TCP/IP connection

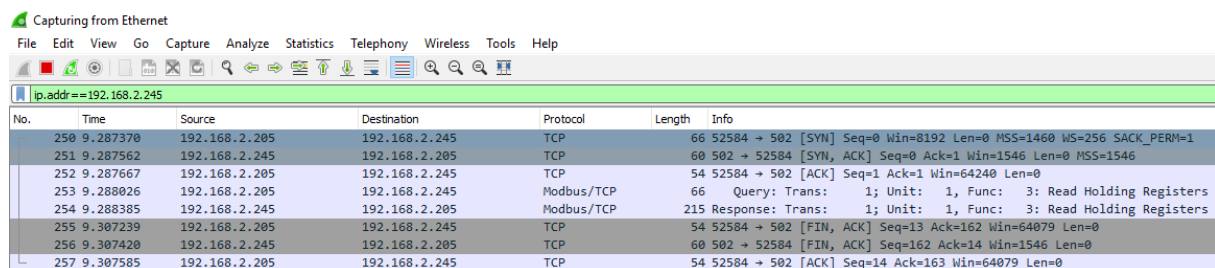
4. Case studies

By testing the program for different kinds of loads, the reliability and accuracy of the program can be rated. Starting with small simple loads and building up to big divers ones make it possible to figure out where the program has difficulties.

4.1 Ideal fixed-step time

The fixed-step time (fundamental sample time) is a critical part of a simulation in Simulink. It defines the period of time in which the program will rerun itself. For this particular program, it means the period of time in which new values where requested and received from the power analysers. If the time is chosen to big the program will measure more inaccurate and if the time is chosen to small the program will be slower and more vulnerable to crash. As mentioned before, the Simulink model has a very fast connection with all the 9 analysers and therefor creates a fast and reliable real time simulation. Multiple tests on the program made clear that only the network slows the program down. Big traffic in the network will cause a slow simulation and will even lead to a failure in the TCP/IP connection.

Testing this connection can be done very easily by using a network protocol analyser. The one used in this case was WireShark version 2.0.2 because it is free and easy to use. Figure 4.1 contains a screenshot while WireShark captures one request and one answer from a pc to the meter “power supply” with IP-address 192.168.2.245 and the total transaction took about 20 ms (from 9.287370 s to 9.307585 s).



No.	Time	Source	Destination	Protocol	Length	Info
250	9.287370	192.168.2.205	192.168.2.245	TCP	66	52584 → 502 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
251	9.287562	192.168.2.245	192.168.2.205	TCP	60	502 → 52584 [SYN, ACK] Seq=0 Ack=1 Win=1546 Len=0 MSS=1546
252	9.287667	192.168.2.205	192.168.2.245	TCP	54	52584 → 502 [ACK] Seq=1 Ack=1 Win=64240 Len=0
253	9.288026	192.168.2.205	192.168.2.245	Modbus/TCP	66	Query: Trans: 1; Unit: 1, Func: 3: Read Holding Registers
254	9.288385	192.168.2.245	192.168.2.205	Modbus/TCP	215	Response: Trans: 1; Unit: 1, Func: 3: Read Holding Registers
255	9.307239	192.168.2.205	192.168.2.245	TCP	54	52584 → 502 [FIN, ACK] Seq=13 Ack=162 Win=64079 Len=0
256	9.307420	192.168.2.245	192.168.2.205	TCP	60	502 → 52584 [FIN, ACK] Seq=162 Ack=14 Win=1546 Len=0
257	9.307585	192.168.2.205	192.168.2.245	TCP	54	52584 → 502 [ACK] Seq=14 Ack=163 Win=64079 Len=0

Figure 4.1 Screenshot of Wireshark capturing 1 request and answer of Meter Power Supply

To determine the ideal fixed-step size, a test was done with a value of 10 ms. The captured data from Wireshark made clear that every new request was done after more than 10ms, meaning that the ideal fixed-step size will be bigger. To determine the time, an average time between request was calculated (Table 4.1). The Average time between requests for 30 requests was 0.178492345 s, meaning that a fixed-step size of 0.15 s will be ideal.

The simulation step time of the Simulink model can be put to “inf” (infinite) this way the simulation can run as long as needed. To stop the simulation, a simple push on the “stop” button is enough.

Request time [s]	Time between requests [s]	Average time between requests [s]
10,796439	0,155343	0,178492345
10,951782	0,225093	
11,176875	0,11002	
11,286895	0,225034	
11,511929	0,105324	
11,617253	0,23242	
11,849673	0,138163	
11,987836	0,222337	
12,210173	0,145131	
12,355304	0,22129	
12,576594	0,155024	
12,731618	0,270629	
13,002247	0,185074	
13,187321	0,209816	
13,397137	0,142352	
13,539489	0,22035	
13,759839	0,165076	
13,924915	0,217123	
14,142038	0,140534	
14,282572	0,23361	
14,516182	0,123099	
14,639281	0,220859	
14,86014	0,150983	
15,011123	0,211204	
15,222327	0,126798	
15,349125	0,179227	
15,528352	0,129716	
15,658068	0,182175	
15,840243	0,132474	
15,972717		

Table 4.1 Measurements and calculations for finding the ideal fixed-step time

4.2 Resistant load

The best first test is the measurement of one resistant load installed on one bus. This because there will be no shifting between the currents and voltages, no reactive power consumption and no harmonic distortion. If the program would fail here, the error(s) will be easy to find.

For the implementation of the resistant load bus B1 was used. The results of the test are visible in Figure 4.2.

The total simulation time was 100 seconds (s), meaning that the scale of the horizontal axe for this simulation is 1/18 s. The chosen fixed-step time (fundamental sample time) was 0.15 s, meaning that every 150 milliseconds new values were requested and received from the power analysers. The load was implemented in the grid after 40 s. The graph of the measured currents shows that there is already some energy consumption after 20 s, before the implementation of the load. The explanation for this is that the installation of the resistant load contains fans to keep the temperature at a normal level. The fans start working as soon as the installation gets connected with the grid. Another observation is that the fans only seem to work on one line, more specific line 2. The measured current after the implementation of the load is 2 ampere (A) for line 2 and 1.3 A for lines 1 and 3. When the consumption of the fans is not taken in account, the current is 1.3 A and the same for each line. The measured active power (P) is 1149 watt (W) this is very close to the theoretical value (1kW) of the load. The Fundamental power factor ($\cos(\varphi)$) and the reactive power (Q) is almost negligible and is caused by the fans.

Overall can be concluded that the measured powers and currents are trustable for this kind of loads.

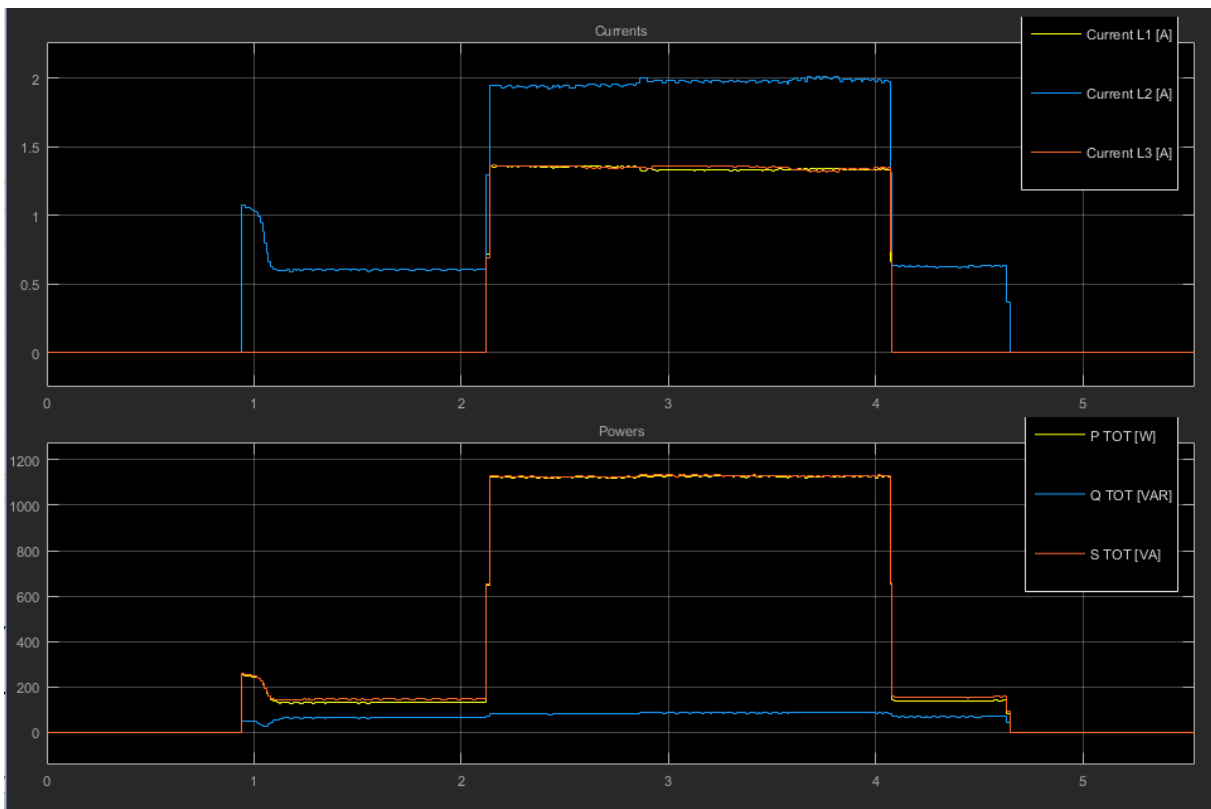
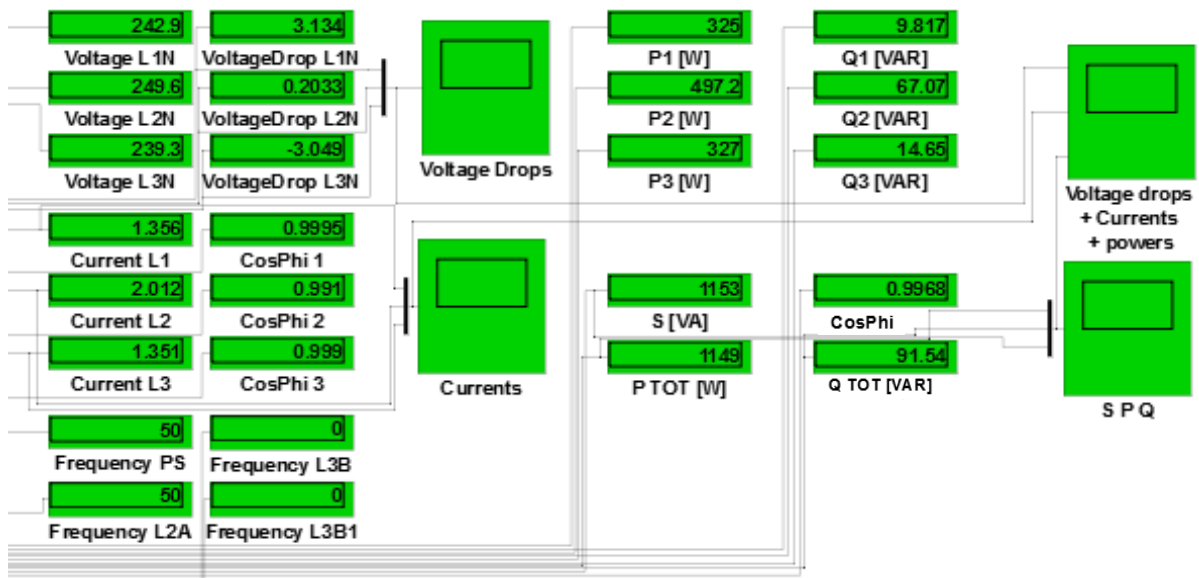


Figure 4.2 Simulation of a resistant load of 1kW

4.3 Capacitive load

The second load to implement is a capacitive load. In this case, the real power will be negligible and the reactive power will almost entirely define the apparent power. Because of this, the $\cos(\varphi)$ of the currents will be close to zero. (Figure 4.3)

The measurement was done in 100 seconds, meaning that the horizontal axes have a scale of 1/18 seconds. The fixed-step size was 0.15 s. Based on the retrieved data during the simulation (see Figure 4.3) can be concluded that the apparent power is almost entirely defined by the reactive power, respectively 7504 VAR and 7520 VA. The $\cos(\varphi)$ is close to zero and the real power is negligible in comparison with the reactive power. The apparent power is not 4kVA as could be expected, this because the load wasn't fully used. Some peaks can be spotted around 18 s and 36 s, the peak at 18 s was caused by connecting the capacitive load installation to the power grid and the peak at 36 s has the implementation of the capacitive component as cause.

The measured values match perfectly with the theoretically expected values and seem trustable.

4.4 Inductive load

The most common inductive load is an induction motor, as it contains coils for inducing magnetic fields. To do so, large electrical conductors are required, which means there is a lot of copper and making the engine not only an inductive but also a resistant load. (Figure 4.4)

The results of the simulation are presented in Figure 4.4 with a horizontal axis scale of 1/15. Fixed-step size 0.15 s. After 20 s the engine was connected with the power grid. The $\cos(\varphi)$ is 0.16 and the apparent power is defined by the reactive power. The copper losses cause an active power consumption of 101 W. All possible values for an inductive load.

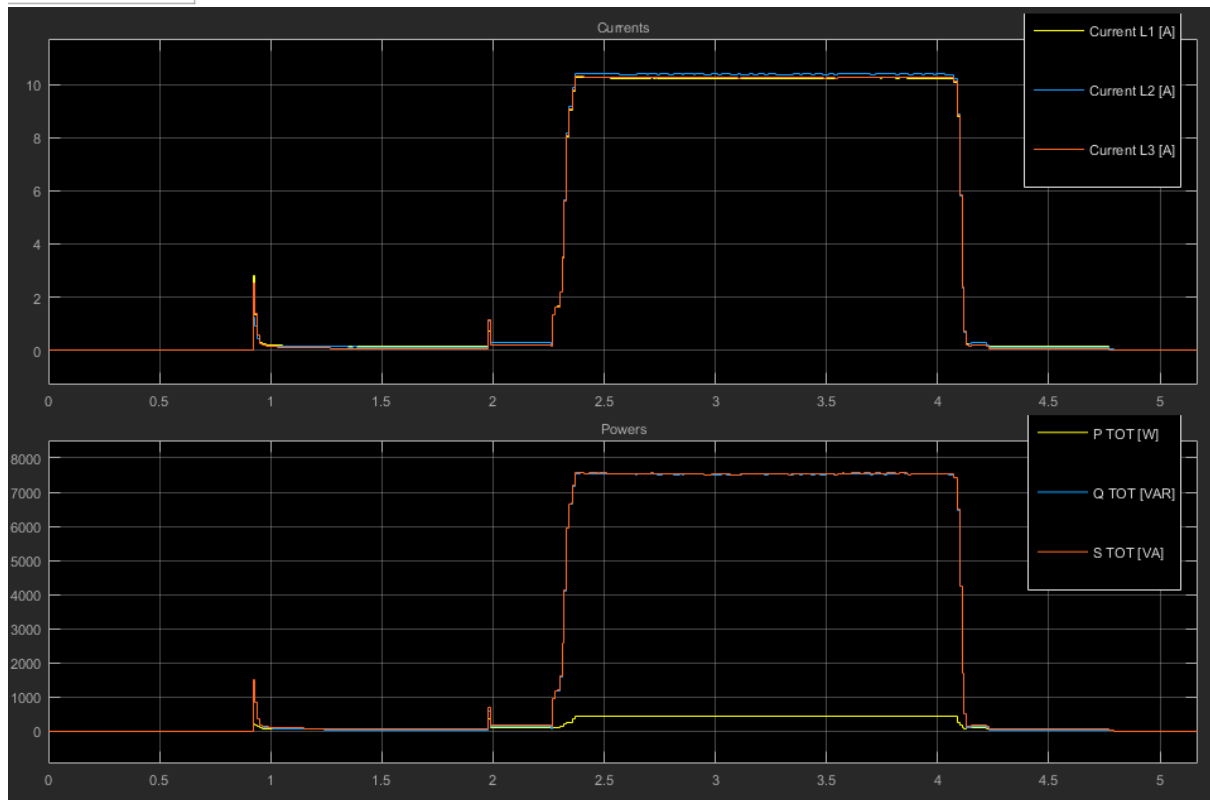
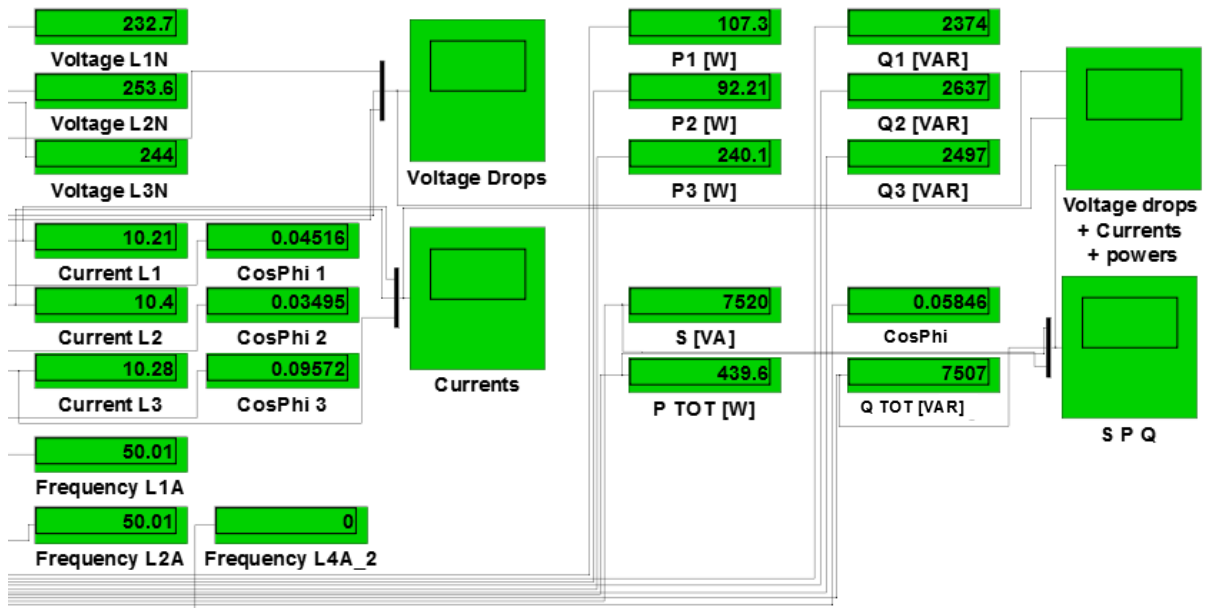


Figure 4.3 Simulation of a capacitive load of 4kVA

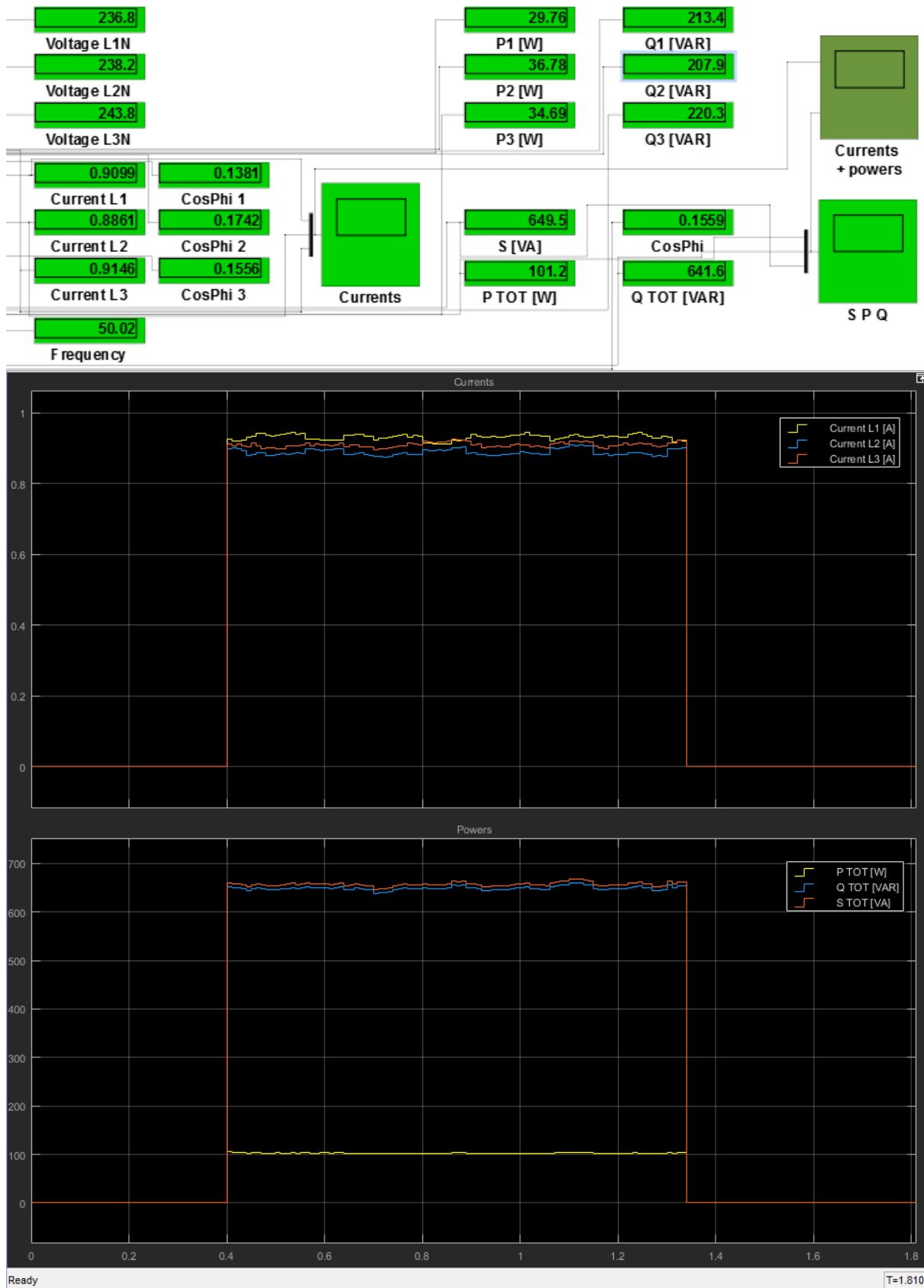


Figure 4.4 Simulation of an inductive load of 0.37kW $\cos(\varphi)=0.69$

4.5 Load Cable

As the cable between meter L1A and L1B is the only one who has only one load underlying, it is easy to check if the measurement seems correct. This was done when bus B4 had the previously mentioned resistive load (see chapter '4.2 Resistive load') implemented. The measured values are plotted in the graphs shown in Figure 4.5. The cable has an average active power of around 2 W, a reactive power of around 2 VAR (CAP) and an apparent power of around 3 VA. The values of the consumed powers are typical for a long cable, concluding that the program measures values close to the reality.

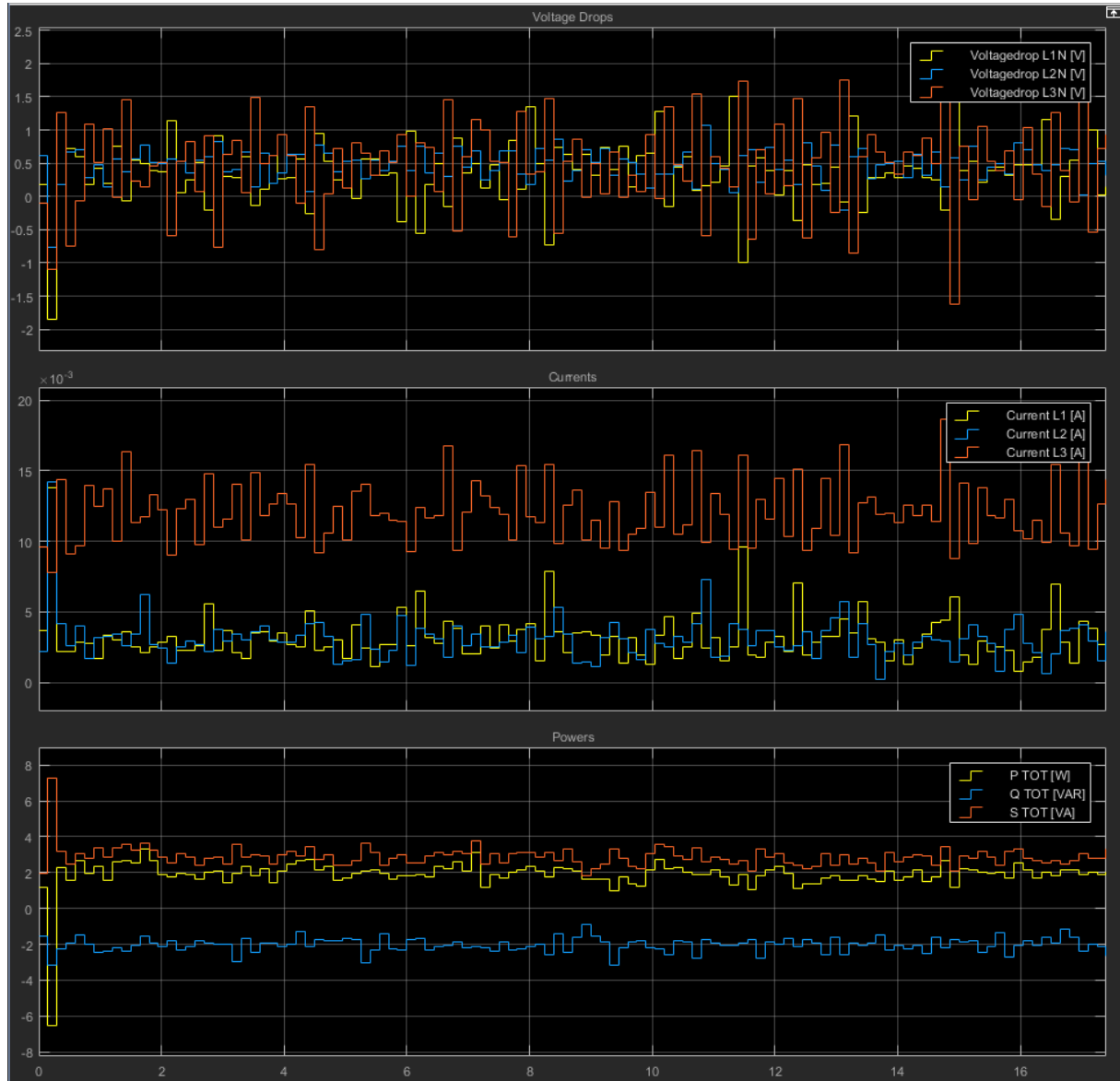


Figure 4.5 Simulation of the load of cable L1A-L1B

4.6 All loads implemented

A test of a fully functional grid is made by implementing all the used loads in different busses. The resistive load in bus B1 (left graphs), the capacitive load in bus B2 (middle graphs) and the inductive load in bus B4 (right graphs). Figure 4.5 gives the resulting graphs of the different busses. The load in bus B2 was on when the simulation started, the resistive load was then implemented and the Inductive load was implemented as last. It is clearly visible that the measured powers and currents are the same as when the loads were measured separately. Except for the capacitive load where, because of the implementation of the inductive load, the reactive power changes to 6000 VA.

Figure 4.6 shows the consumption of all the loads plus the 2 cables. As the load of the cables is negligible to that of the loads in the busses, the measured values depend only on the loads. The resistive load (Bus B2) has a total power of 1149 W and 92 VAR (values from previous measurement in chapter '4.2 Resistive load'), the capacitive load (Bus B1) has a total power of 440 W and 6000 VA (CAP) (see left graphs on Figure 4.5) and the inductive load 101 W and 641 VAR (IND). This should give a total of $\sqrt{(1149 + 440 + 101)^2 + (92 + 6000 + 641)^2} = 6941$ VA, 1690 W and 6733 VAR and that is very near to what the middle graph of Figure 4.7 shows.

Given this, the program can be described as trustable and accurate for the real time simulation of a power grid.

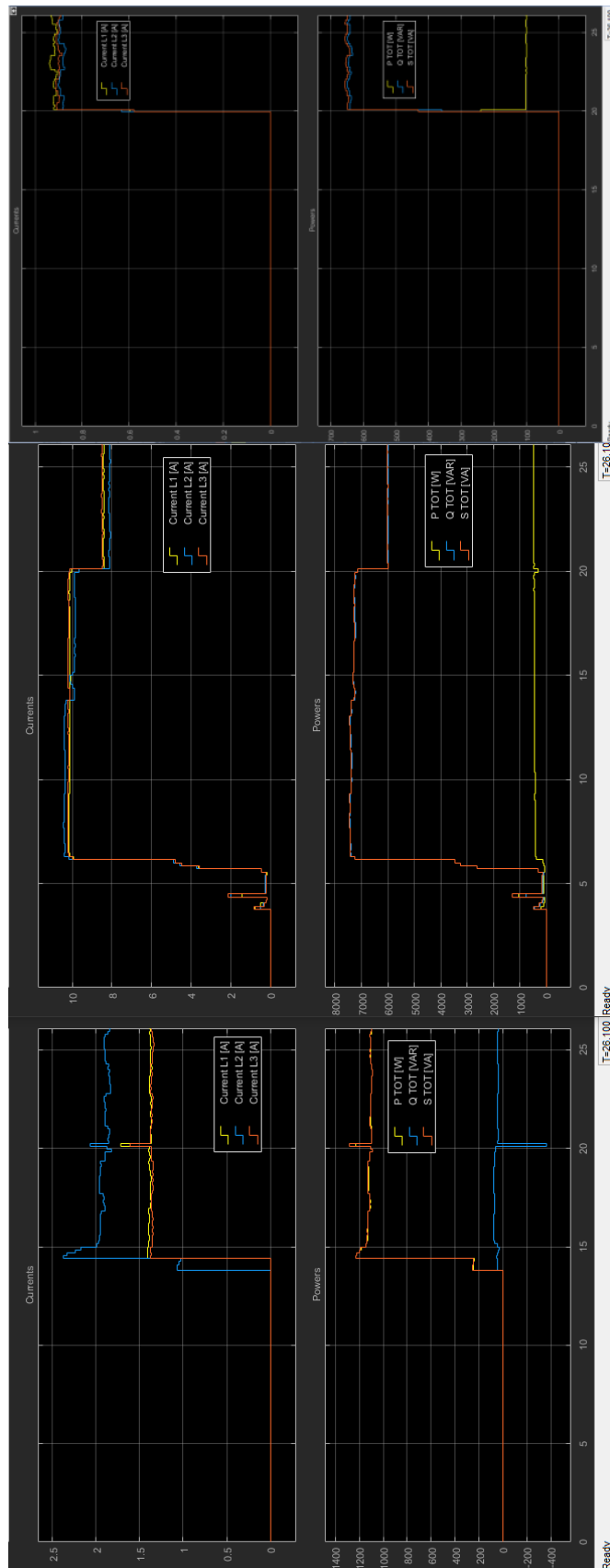


Figure 4.6 Simulation of all loads Bus B1, B2 and B4

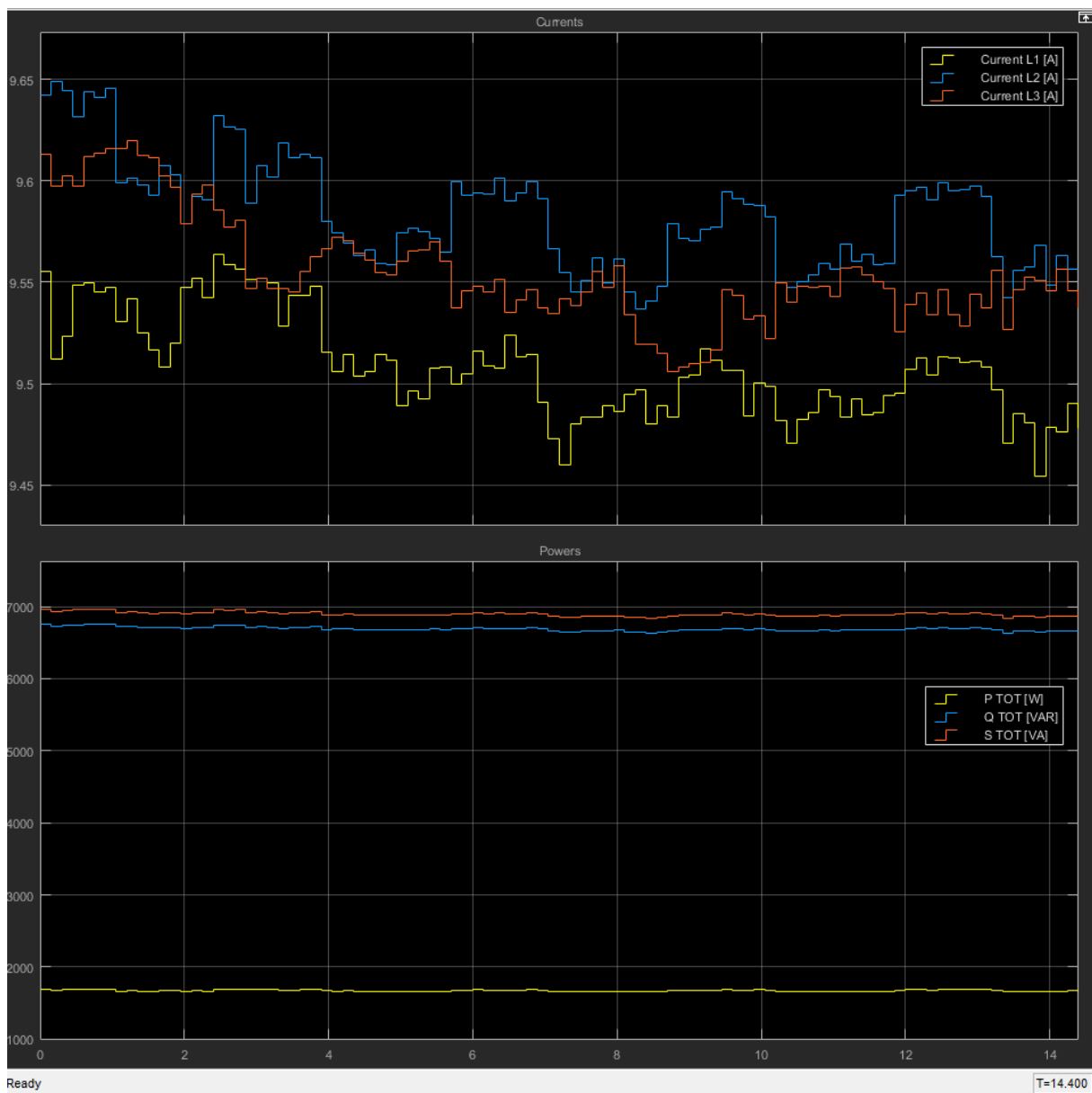
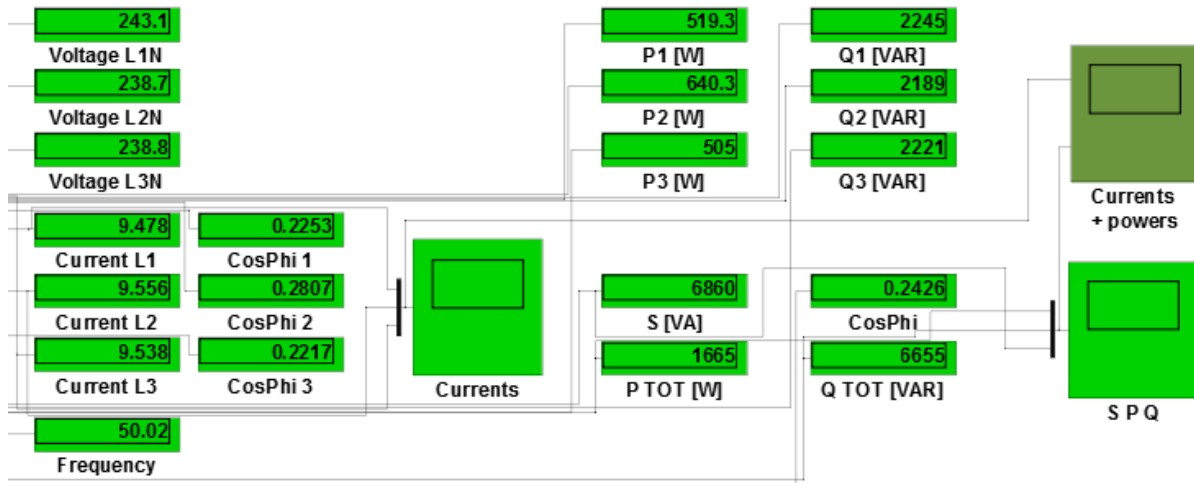


Figure 4.7 Simulation of all loads total grid consumption

5. Conclusion

5.1 Word of appreciation

Words of appreciation are in place for the people who have helped to accomplish this work. First of all there is prof. Zita Vale for advising where possible. The second person is Pedro Faria who helped on the field, was always contactable and put the research back on the rails when that was necessary. The third person was Omid Abrishambaf who was always one room away and never bothered taking time to solve problems. The people of the GECAD research centre gave me a warm welcome and a friendly environment to do my work. My experiences in Porto would not be as good as it was without them.

5.2 Main conclusion and future perspectives

This thesis work focused on developing a real-time simulation tool for monitoring an electrical grid. The monitoring was done by JANITZA UMG 96RM EL power analysers, the connection was done over Ethernet using MODBUS TCP/IP as communication protocol and the programmable environment was Matlab/Simulink. As it is a simulation tool that does not depend on a specific brand of power analysers, the program can be used for all types of power analysers.

In a world where the demand for energy is growing and energy consumption is higher than the natural resources are able to supply, all help is welcome to find solutions and this work is one. Via the energy monitoring system developed in this work, the users are able to check their energy consumption and adept where needed.

This work can serve as a foundation for companies to develop a same system in their environment. It is also much more user friendly than the software program Janitza developed. Furthermore, extra studies can use the same techniques in different programmable environments. GECAD can use this work for measurements in their laboratory and can further develop it in the scope of future energy monitoring systems.

Bibliography

1. EIA, International Energy Outlook 2013, [www.eia.gov/forecasts/ieo/pdf/0484\(2013\).pdf](http://www.eia.gov/forecasts/ieo/pdf/0484(2013).pdf), 2013
2. IPCC, Climate change 2007: the physical science basis. Contribution of Working Group III to the Fourth Assessment Report of the Intergovernmental Panel on Climate Change, Paris, 2007
3. IEA, 2015 Key world energy statistics, https://www.iea.org/publications/freepublications/publication/KeyWorld_Statistics_2015.pdf, France, November 2015
4. EC, Energy Efficiency: Saving energy saving money, <http://ec.europa.eu/energy/en/topics/energy-efficiency>
5. EC, Eurostat Statistics Explained, Electricity and heat statistics, http://ec.europa.eu/eurostat/statistics-explained/index.php/Electricity_and_heat_statistics, May 2015
6. EC, Energy, Smart grids and meters, <https://ec.europa.eu/energy/en/topics/markets-and-consumers/smart-grids-and-meters>, 6 October 2016
7. EC, COMMISSION STAFF WORKING DOCUMENT Cost-benefit analyses & state of play of smart metering deployment in the EU-27 Accompanying the document Report from the Commission Benchmarking smart metering deployment in the EU-27 with a focus on electricity, <http://eur-lex.europa.eu/legal-content/EN/NOT/?uri=SWD:2014:189:FIN>, 17 June 2014
8. N.N. , Energy monitoring and targeting, https://en.wikipedia.org/wiki/Energy_monitoring_and_targeting
9. MathWorks, Creating the Matlab logo, http://nl.mathworks.com/examples/matlab/mw/matlab_featured-ex16591756-creating-the-matlab-logo?s_tid=srchtitle, 2016
10. JANITZA UMG 96RM -P/-M/-CBM/-EL, <http://www.janitza.com/umg-96rm-p-m-cbm-el-features.html>
11. Janitza, Power Analyser UMG 96 RM-EL, Operating instructions and technical data, 2016
12. MathWorks, TCP/IP Receive, <http://nl.mathworks.com/help/instrument/tcpipreceive.html>, 2016
13. Joakim Wargh, Timestamping over MODBUS TCP from Siemens S7, https://publications.theseus.fi/bitstream/handle/10024/60951/Wargh_Joakim.pdf?sequence=1, 2013
14. Modbus, <https://en.wikipedia.org/wiki/Modbus>
15. George Thomas, Introduction to Modbus Serial and Modbus TCP, the EXTENSION, September-October 2008
16. Modbus, MODBUS APPLICATION PROTOCOL SPECIFICATION V1.1b3, http://modbus.org/docs/Modbus_Application_Protocol_V1_1b3.pdf, 26 April 2012
17. MODBUS MESSAGING ON TCP/IP IMPLEMENTATION GUIDE V1.0b, http://modbus.org/docs/Modbus_Messaging_Implementation_Guide_V1_0b.pdf, 24 October 2006
18. GECAD, Brief Description, <http://www.gecad.isep.ipp.pt/GECAD/Pages/Presentation/Home.aspx>

19. Modbus TCP and MATLAB, <http://stackoverflow.com/questions/21416239/modbus-tcp-and-matlab>
 20. Roger Stafford, Decoding Floating point (IEEE-754),
http://nl.mathworks.com/matlabcentral/newsreader/view_thread/132795, 20 Sep 2006
- WireShark, About Wireshark, <https://www.wireshark.org>, 2016

Appendix

This appendix contains:

- The complete Simulink model
- The complete Matlab Code of the 'Matlab Function block' for the meter L1A
 - The other 8 meters have the same code, with exception of the IP-address.

Note: The appendix is not publicly published, for looking into this please contact ISEP or GECAD.