



Stefan Obermoser, BSc.

Development of a software for automated operations of a multifunctional tensile and pressure test facility

Master thesis for obtaining the degree Master of Science in the field of studies Mechanical Engineering & Business Economics - Production Science and Management

Realized at the Vehicle Safety Institute
Graz University of Technology

Chair: Univ.-Prof. Dipl.-Ing. Dr.techn. Hermann Steffan

Supervisor 1: Dipl.-Ing. Christian Ellersdorfer

Supervisor 2: Univ.-Prof. Dipl.-Ing. Dr.techn. Hermann Steffan

Graz, Mai 2013



EIDESSTATTLICHE ERKLÄRUNG

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommene Stellen als solche kenntlich gemacht habe.

STATUTORY DECLARATION

I declare that I have authored this thesis independently, that I have not used other than the declared sources / resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

Graz, am

.....

(Unterschrift)

Acknowledgement

Great thanks appertain to my supervisor Dipl. Ing. Christian Ellersdorfer for a lot of inspiring, idea generating and also funny talks to realize my thesis. Very high respect and extraordinary thanks are directed to Dipl. Ing. Helmut Wurm for spending hours in front of a computer to help me solve my coding problems.

I'd like to thank as well the whole staff of the Vehicle Safety Institute for the very good working atmosphere which influenced my way of working and my motivation to deliver good results pretty much.

Thank you also to Prof. Dr.techn. Dipl. Ing. Hermann Steffan for the possibility to write my master thesis at his institute.

To Prof. Dr.techn. Dipl. Ing. Josef W. Wohinz thanks are addressed because of his example he set during my time at the Institute of Industrial Engineering and Innovation Research and the Institute of Production Science and Management.

Acknowledgement to the whole "Mean Machine" because they are the best friends a person can ever have.

Not to forget to thank "Dr. med. univ. to-be" Marie-Christin Klöckl who enriches my life in so many ways.

The biggest thanks of all go to my family especially to my mom and dad who gave me the possibility to finish my studies and educated me to an independent individual. I'd like to thank them as well for the more friendship than parental relationship and for all the things which are not a matter of course for most of the people in the world. Big thanks as well to my older brother who was always a role model for me. His big efforts to change our world and society are outstanding but often not respected by the majority of people.

At the preliminary end of my students' career I'd like to quote, a in my point of view highly respectable person, sportsman, actor and politician, Arnold Schwarzenegger in

Terminator 2 – Judgement Day:

" Hasta La Vista Baby! "

Kurzfassung

Fahrzeugsicherheit gewinnt in der heutigen Gesellschaft immer weiter an Bedeutung. Die Verletzungs- und Tötungsrate bei Unfällen wurde seit Beginn der automobilen Geschichte kontinuierlich gesenkt. Entwicklung und Test neuer Materialien spielt dabei eine nicht unerhebliche Rolle. Für die Simulation von Materialverhalten mit Hilfe von numerischen Methoden werden unumgänglich Daten von empirischen Versuchen benötigt. Die Generierung und Aufzeichnung dieser Daten ist jedoch nicht immer eine triviale Angelegenheit. Hoch technisierte und ausgereifte Messeinrichtungen finden hier ihre Anwendung. Jedoch bestehen für viele der zu prüfende Materialien besondere Richtlinien beziehungsweise Anforderungen, wie man am Beispiel der Elektromobilität sehr gut erläutern kann:

Durch mannigfaltige ökologische und gesellschaftliche Gründe hat die Elektromobilität in den letzten Jahren einen regelrechten Boom erlebt. Bei der Entwicklung der verwendeten Batteriemodule spielt der Effekt des „Thermal Runaway“ eine signifikante Rolle. Laut (Barnett, et al., 2013) versteht man unter diesem Effekt eine sich selbst verstärkende Wärme produzierende Reaktion, welche zu einem sehr schnellen Temperaturanstieg führt. In Lithium-Ionen Zellen kann dies bis zum Brand, und oder zur Explosion dieser führen.

Eine Prüfeinrichtung die mit diesen Bedingungen umgehen kann ist weitestgehend am Markt noch nicht verfügbar oder erfüllt die Kriterien nur unzureichend. Daher wurde die Eigenentwicklung einer Testeinrichtung für Batteriesysteme und andere Materialien am Institut für Fahrzeugsicherheit an der Technischen Universität Graz bevorzugt.

Thema dieser Abschlussarbeit ist die mechanische Erweiterung der entwickelten Prüfumgebung - PRESTO 420 - für Zugversuche, da bis dato nur simple Druckversuche möglich waren. Des Weiteren soll eine Automatisierung dieser durch Entwicklung einer „easy to use“ Steuerungs- und Regelungssoftware erfolgen. Ziel der Entwicklungsarbeit ist es neben frei definierbaren Testsequenzen auch standardisierte Prüfabläufe mit hoher Reproduzierbarkeit durchführen zu können. Ein automatisches Erkennen von diversen Testabbruchgründen wie zum Beispiel maximale Kraft, maximaler Verfahrweg, Wartezeit und Kurzschluss sind dafür notwendig. Um eine hohe Reproduzierbarkeit erreichen zu können soll eine geeignete Geschwindigkeitsregelung entwickelt werden. Auch die Bedienerfreundlichkeit steht im Fokus der Bemühungen. Dem Benutzer soll eine intuitive Steuerung mit geringer Eingabefehlerwahrscheinlichkeit ermöglicht werden. Das bereits bestehende Messdatenerfassungssystem soll an neue Anforderungen angepasst und erweitert werden. Diese Maßnahmen sollen dem Institut für Fahrzeugsicherheit das Prüfen des Verhaltens von verschiedensten Materialien, wie auch Batteriezellen und –modulen, unter quasistatischen Zug- und Druckbelastungen ermöglichen.

Abstract

In the last years vehicle safety gained more and more importance in the society. The injury and death risk in accidents has dropped down continuously since the beginning of the automobile history. Development and testing of new materials plays a major role in this research field. To simulate the behaviour of materials with the help of numerical calculation methods it is indispensable to use empirical data gathered in experiments. The generation of this data is often not an easy task. Highly engineered and sophisticated test facilities find here its field of application. However, a lot of materials have defined guidelines or requirements for testing, which complicates this task. This fact is easy to explain with the example e-mobility:

Due to diversified ecological and social reasons e-mobility boomed in the last years. In the development of used battery modules, the effect of the “thermal runaway” plays a significant role. According to (Barnett, et al., 2013) it is a self-reinforcing heat generating process which results in a rapidly raising temperature. This causes an inevitably defect of the equipment. Lithium-Ion cells start to burn or even explode.

A test facility, which is able to handle these constraints, is as far as possible not available on the market or does not sufficiently fulfil the requirements. Therefore the Vehicles Safety Institute at the Graz University of Technology decided on a self-development of new test machinery for battery systems and other materials.

The task of this master thesis is to further develop the hardware of the test facility – PRESTO 420 – to enable tensile tests, by then just simple pressure tests were possible. Furthermore a full automation through programming of an “easy to use” control software has to be done. Goal of the development is to enable standardized tests with a high degree of reproducibility besides the possibility to freely define a test sequence. In favour of an automatic recognition of test abort reasons as for example maximum force, maximum displacement, waiting time and short-circuit is needed. To reach the high degree of reproducibility of a suitable velocity feedback control system should be developed. The operator convenience is also in the focus of the development work. The user should be able to intuitively control the test facility with a minimum of errors. The already existing data acquisition should be fitted to the new requirements and extended. These actions should allow the VSI to test the behaviour of different materials in the same ways as battery cells and -modules under quasi-static tensile or pressure conditions and to monitor the numerical data for simulation reasons.

Table of content

1	Introduction and goal of the thesis.....	1
2	Approach.....	3
2.1	Description of the different stages	3
3	Initial situation.....	5
3.1	Overview	5
3.2	Components of the test facility - PRESTO 420	5
3.3	Controlling & Measurement System - NI CompactRIO 9014.....	7
3.4	Software	9
4	Measurement and Control System	11
4.1	Requirements	11
4.2	Measurement System – CompactRIO 9024	14
4.3	Control System – CompactRIO 9014	18
5	Programming Environment – LabVIEW™ 2012	20
5.1	What is LabVIEW™	20
5.2	Programming levels in CompactRIO real-time systems.....	21
6	Software - PRESTO 420 AUT.....	23
6.1	Analysis of the existing software.....	26
6.2	Requirements - PRESTO 420 AUT.....	32
6.3	Program Structure - PRESTO 420 AUT	37
6.4	Manual Operations.....	42
6.5	Automated Operations	47
6.6	Preferences for Manual & Automatic Tests.....	52
7	Tensile Test.....	62
7.1	Hardware Modifications.....	62
7.2	Software Implementation	63

8	Velocity Control.....	64
8.1	Feedback Control System basics	64
8.2	Implemented Velocity Feedback Controller	71
9	Software Validation Tests	81
9.1	Crush Tube – Validation tests.....	81
9.2	Battery – Validation tests	87
10	Conclusion & Outlook.....	94
11	List of References	96
12	List of Figures.....	97
13	List of Tables.....	100
14	List of Charts.....	101
15	List of Equations	102
16	List of Abbreviations.....	103
	Appendix.....	i

1 Introduction and goal of the thesis

In the beginning of automobile history, safety did not play a big role. Since the number of vehicles and as a result the number of accidents raised, the need for a safer way of transportation grew. The automotive industry put big efforts into the design of safer cars. For that reason the need for new materials has developed as well. The test of these materials gained tremendously importance. Testing of materials is done to get knowledge of the material behaviour. With the in experiments gained data simulations can be fulfilled with numerical calculation methods. Previously performed simulations of components can reduce development time and costs very well.

As mentioned before testing is a major part in the development of new technology. This can be easily seen in a field which gained more and more importance in the last years due to worldwide effects like the climate change, the global warming and the limited resources of crude oil force. Big efforts to overcome future problems all over the world are the consequence. The result is the on-going development of CO₂ neutral cars, or at least CO₂ output reduced cars.

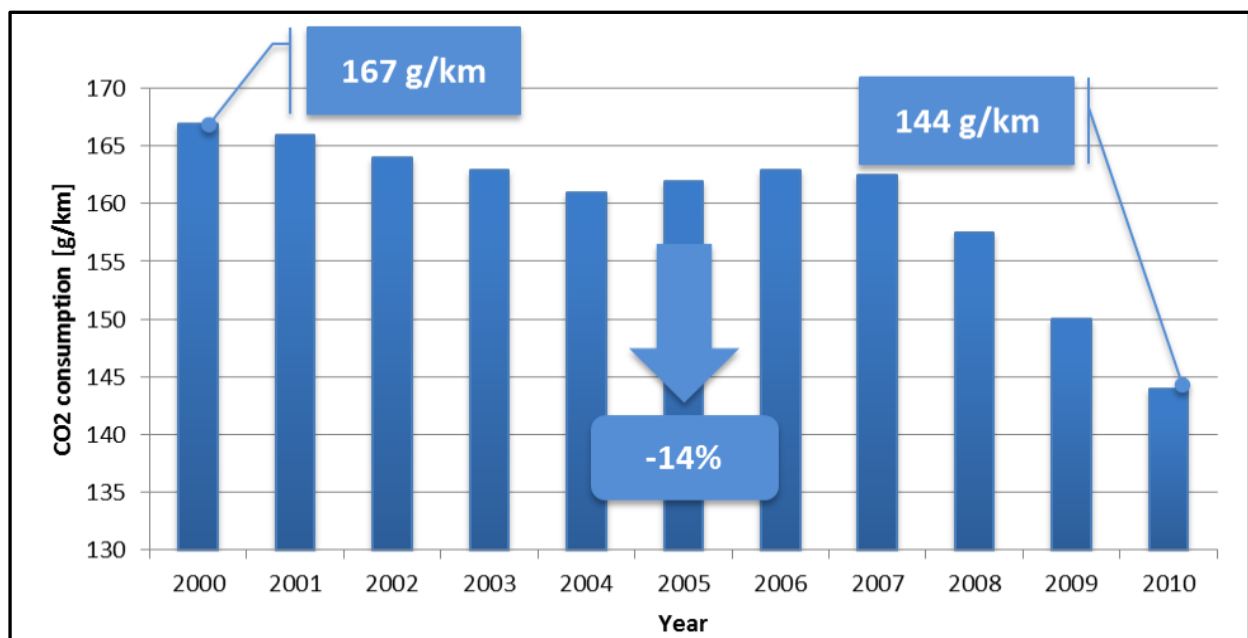


Chart 1: "Development CO₂-fleet consumption in Austria (Arbeitskreis der Automobilimporteure, 2012)"

This is why the electro mobility experiences some kind of revival. With the right energy storage systems it is a big step forward to design eco-friendly cars. At the moment the Lithium-Ion

technology is the most promising one. But it is not only a challenge for powertrain designing engineers also the area of vehicle safety has to face challenging new problems.

The outcome of this thesis provides the basis for the better understanding of the mechanical load of Lithium-Ion battery cells and other new materials. It should help to improve the predictability of a mechanical material failure with empirical data of tensile and pressure tests.

The goal is to develop a software for automated operations of a multifunctional tensile and pressure test facility called PRESTO 420 (**PRE**ss for Battery **SysT**ems and **O**thers **420**kN) based on the diploma thesis of Dipl.-Ing. Christoph Knauder.

The following tasks have to be fulfilled with this thesis:

- An easy to use software for automated and manual operations has to be programmed with minimum operating errors
- The control interface has to be locally independent of the test facility (industrial pc) for security reasons
- All communications have to be done via Ethernet
- Transition of the manual velocity control via setting voltage of a current regulating valve into an automated velocity control
- Tensile and pressure testing have to be possible with the test facility
- The PRESTO 420 has to be capable of performing all kinds of tests needed for battery testing (tensile phase, pressure phase, waiting phase, short circuit detection, maximum force detection)
- High reproducibility of test setups

2 Approach

In this chapter the way how to fulfil the goals of this thesis will be described. A detailed description of the different stages is included.

2.1 Description of the different stages

The in Figure 1: “Flowchart – approach master thesis” shown stages are describing the approach of this thesis.

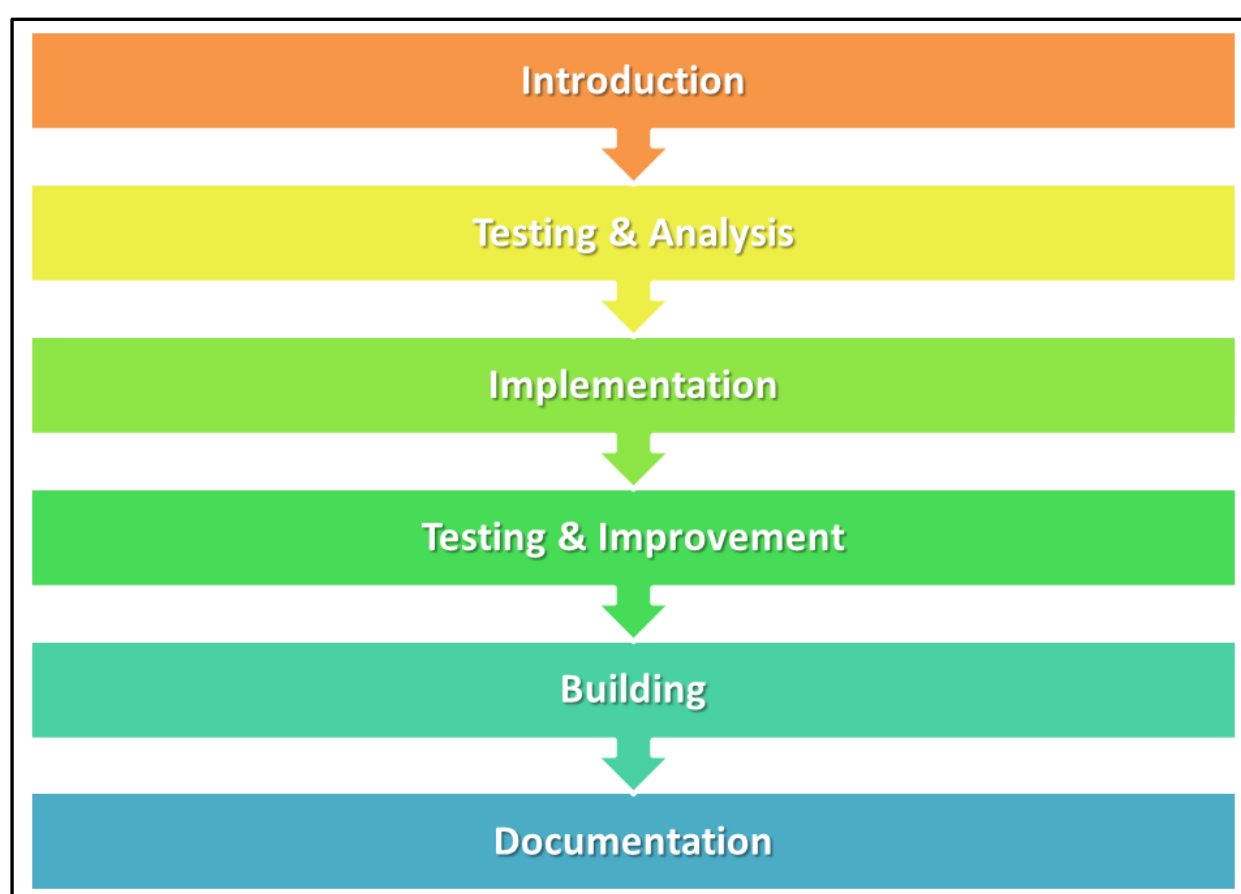


Figure 1: “Flowchart – approach master thesis”

2.1.1 Stage 1: Introduction

The goal of this phase is to gain the needed programming skills for following stages. To meet this goal a LabVIEW™ – Basics 1 lecture was taken.

2.1.2 Stage 2: Testing & Analysis

The analysis of the already existing basic control software and its problems is a major part of this stage. The improvement potential and weaknesses will be explored and requirements for the programming of the new software will be set.

The performing of first mechanical tests to gain knowledge of the requirements of the test environment is also an important part. With that a specification sheet is going to be derived.

2.1.3 Stage 3: Implementation

The implementation of the new software features “tensile test”, “velocity control” and “automated operations” as well as useful features which were discovered in Stage 2: Testing & Analysis” is the focus of this stage. All the new implemented features are going to be explained during this work.

2.1.4 Stage 4: Testing & Improvement

To validate the full functionality of the new software it has to be tested. These tests will be fulfilled with materials, where the mechanical properties are already known, to have the possibility to evaluate them. During these tests occurring software “bugs”¹ will as well be removed.

The software will be shown to the future operating staff to meet their expectations for good and easy to use control software.

2.1.5 Stage 5: Building

An enclosure for all the test facilities electronic components and a save enclosure for the operator pc have to be designed and to be built. The concept for the independent hardware user interface will be explained later in this work.

2.1.6 Stage 6: Documentation

To complete the master thesis everything will be documented.

¹ Bug – describes a malfunction in a software in general

3 Initial situation

In this stage the initial situation of this thesis will be explained. The status quo of the development of the test facility PRESTO 420 will be described. The initial situation of this thesis is the diploma thesis “Development of a testing environment to analyse the mechanical properties of lithium based batteries”, written by Dipl.-Ing. Christoph Knauder. This chapter is based on (Knauder, 2012).

3.1 Overview

The construction of the test facility PRESTO 420 was already existing and able to perform manual pressure tests. Because of that, a basic software was designed and a data acquisition was already programmed. The control of the feed motion worked manually via changing the voltage of the pressure current control valve. One NI Compact RIO 9014² controller was used for measuring and controlling means. This controller faced already its performance limits in terms of processing power. For this reason there was no more space for further expansion of the software.

3.2 Components of the test facility - PRESTO 420

As already mentioned the construction of the test facility was completed for simple pressure tests. The design was done according to a specification book developed by Dipl. Ing. Knauder and can be found in his diploma thesis. Here the main specification points will be repeated for a better understanding of the topic.

Plant-specific requirements:

- Mechanical solid realization to minimize deformations
- Displacement measurement at areas with the lowest deformations
- Force measurement directly at the test set-up
- Steady guided movement
- Enough force reserves for extended cases of application

² NI Compact RIO 9014 is an embedded real-time controller which performs controlling and data acquisition tasks

- Fire proof positioning of the measurement system and the drive elements
- Reproducibility of tests
- Flexibility of operability
- Transportability

Figure 2: “Components of the test facility – PRESTO 420 (Knauder, 2012)” shows the complete construction of the test facility with its components.

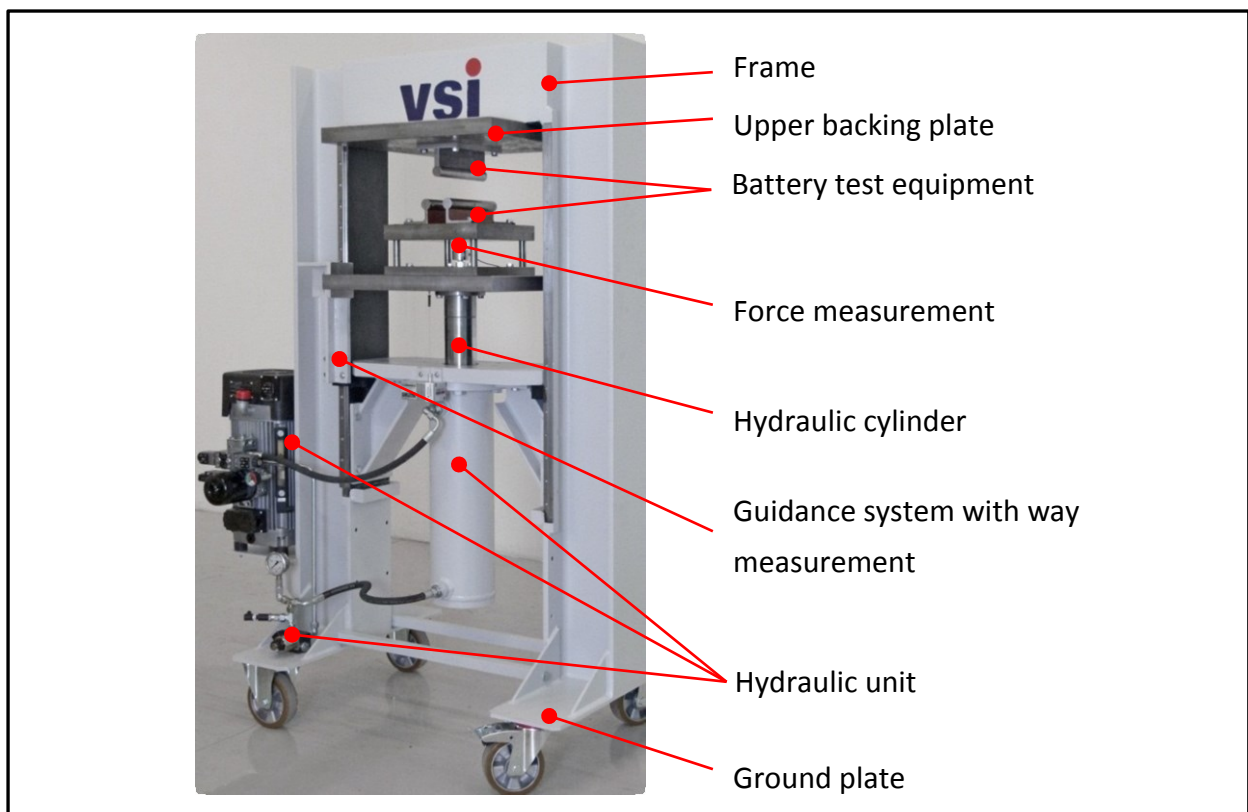


Figure 2: “Components of the test facility – PRESTO 420 (Knauder, 2012)”

3.3 Controlling & Measurement System - NI CompactRIO³ 9014

All the data acquisition and control tasks were performed by one CompactRIO controller with an included FPGA. The setup of this controller will be shown in Table 1: “Controlling system components (Knauder, 2012)” and in Figure 3: “Hardware components of the CompactRIO – controlling and measurement system (Knauder, 2012)”.

Components:	Name, Information:	Data:	Unit:
Real-time controller:	NI 9014		
	Clock rate:	400	MHz
	Working memory:	128	MB-DRAM
	Storage:	2	GB
Chassis:	NI 9114		
	Slots:	8	
FPGA:	Xilinx Virtex-5 LX50		
Input module:			
Digital module:	NI 9401	TTL-able (5V/TTL)	
	Channels:	8	
	Clock rate:	100	ns
Force measuring module:	NI 9237	Bridge module	
	Channels:	4	
	Resolution:	24 bit	
	Sample rate:	50	kS/s
Analogue input module:	NI 9206	Power input	
	Channels:	16	
	Resolution:	16 bit	
	Sample rate:	250	kS/s
Output module:			
Analogue output module:	NI 9263	Power output ($\pm 10V$)	
	Channel:	4	
	Sample rate:	100	kS/s
	Resolution:	16 bit	
Relay digital module:	NI9481	30VDC(2A), 60VDC(1A), 250VAC (2A)	
	Channels:	4	

Table 1: “Controlling system components (Knauder, 2012)”

³ CompactRIO – Compact Reconfigurable Input/Output

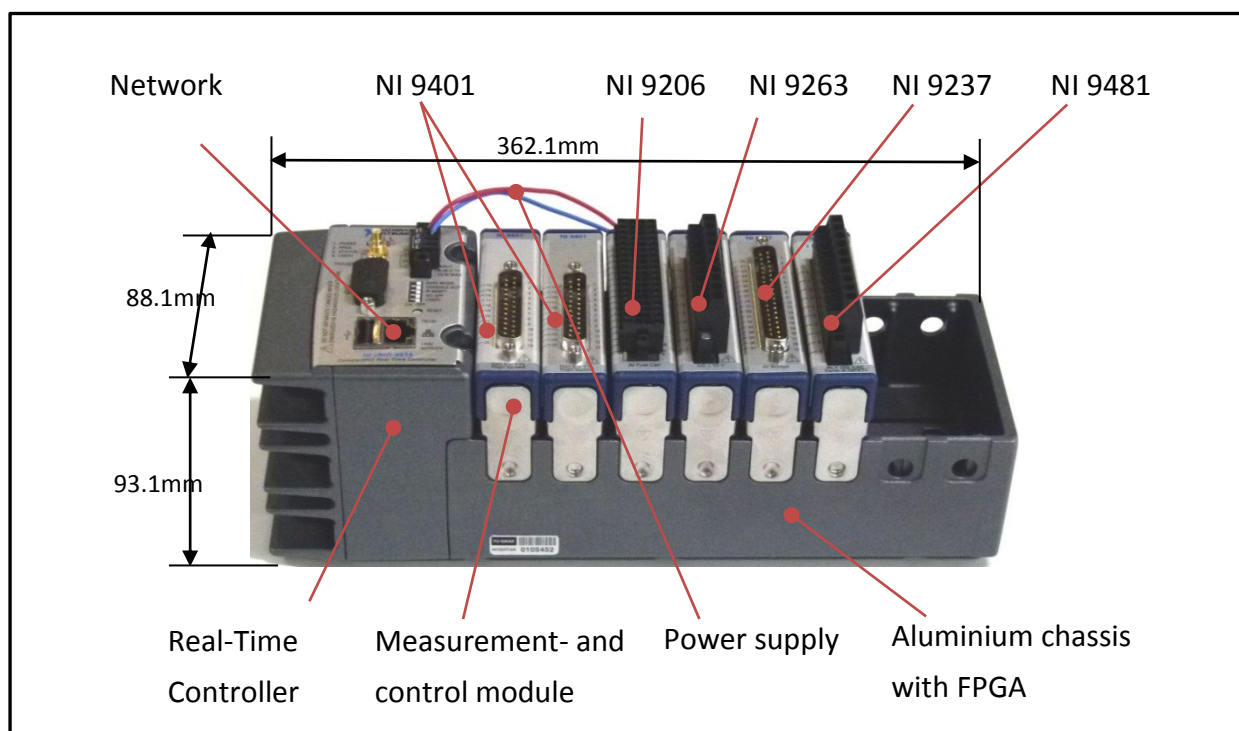


Figure 3: “Hardware components of the CompactRIO – controlling and measurement system (Knauder, 2012)”

The NI CompactRIO 9014 real-time controller is the heart of the system. It is an embedded controller with a clock rate of 400 MHz and 128 MB-DRAM working memory. It is connected via a frame (aluminium chassis) with the different modules. Every module has its own task. The NI 9401 module is gathering the data of the glass scale to measure the displacement with a clock rate of 100ns. The NI 9206 module is gathering the voltage data of the connected battery cell or module on 16 different channels with a resolution of 16 bit. The NI 9263 is controlling the current regulating valve for the piston movement and the current regulating valve for the pressure force. The NI 9237 module is collecting the force measurement data on four different channels with a resolution of 24 bit. The NI 9481 module is a digital relay module which activates the hydraulic system and the up/down movement of the piston. In the aluminium chassis there is also a built-in FPGA. The FPGA is a field programmable gate array which is the gateway between the real-time controller and the modules. Its task is to address the different modules with a defined programmed logic. It is also the fastest part of the system in terms of clock rate so some parts of the data processing also find here its application.

3.4 Software

Figure 4: “Principle of the controlling and measurement software (Knauder, 2012)” shows the basic structure of the control and data acquisition software.

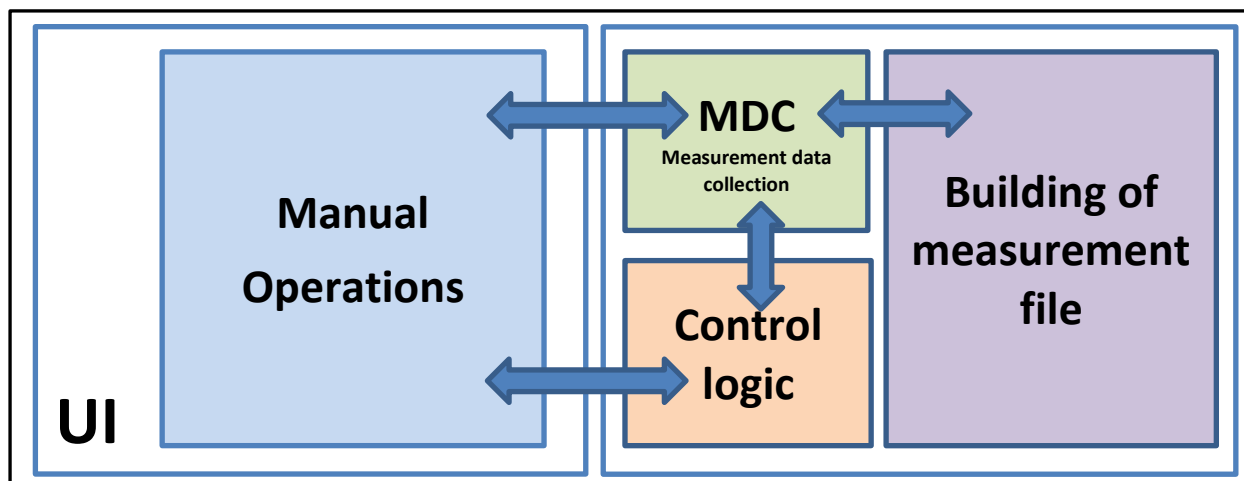


Figure 4: “Principle of the controlling and measurement software (Knauder, 2012)”

The communication with the operator is done in the UI⁴ running on a windows desktop system. This only exists for visualisation reasons. The measurement data collection and the control logic are located at the CompactRIO real-time system and at the FPGA. The communication between Windows Host and CompactRIO real-time system is realized via Ethernet. Since Ethernet is not a real-time bus, all program parts which have to be capable of real-time processing are based on the CompactRIO system. The maximum resolution of the data acquisition rate for the force measurement is 50kS/s, for the voltage metering 250kS/s. For more detailed information see (Knauder, 2012).

3.4.1 User Interface – Manual Operations

Figure 5: “User Interface – Manual Operations” shows the UI in the beginning of this thesis which made basic operations possible. Manual pressure tests and manual piston movement was possible at this development state. You can see that the UI is split up into two parts: on the left is the part where user interactions occurs (buttons, etc.) and on the right is the visualization part, where all the measuring data is shown to the user.

⁴ UI – User Interface

Before a test can be started, the user has to define a few things. For the data acquisition the measurement channels have to be selected, except of time and displacement measurement which is always performed. To enable the force measurement, the properties of the used load cell have to be defined. To enable the temperature measurement the parameters of the thermo element have to be defined. It is also needed to previously define the test speed, the test force and the name of the measurement file.

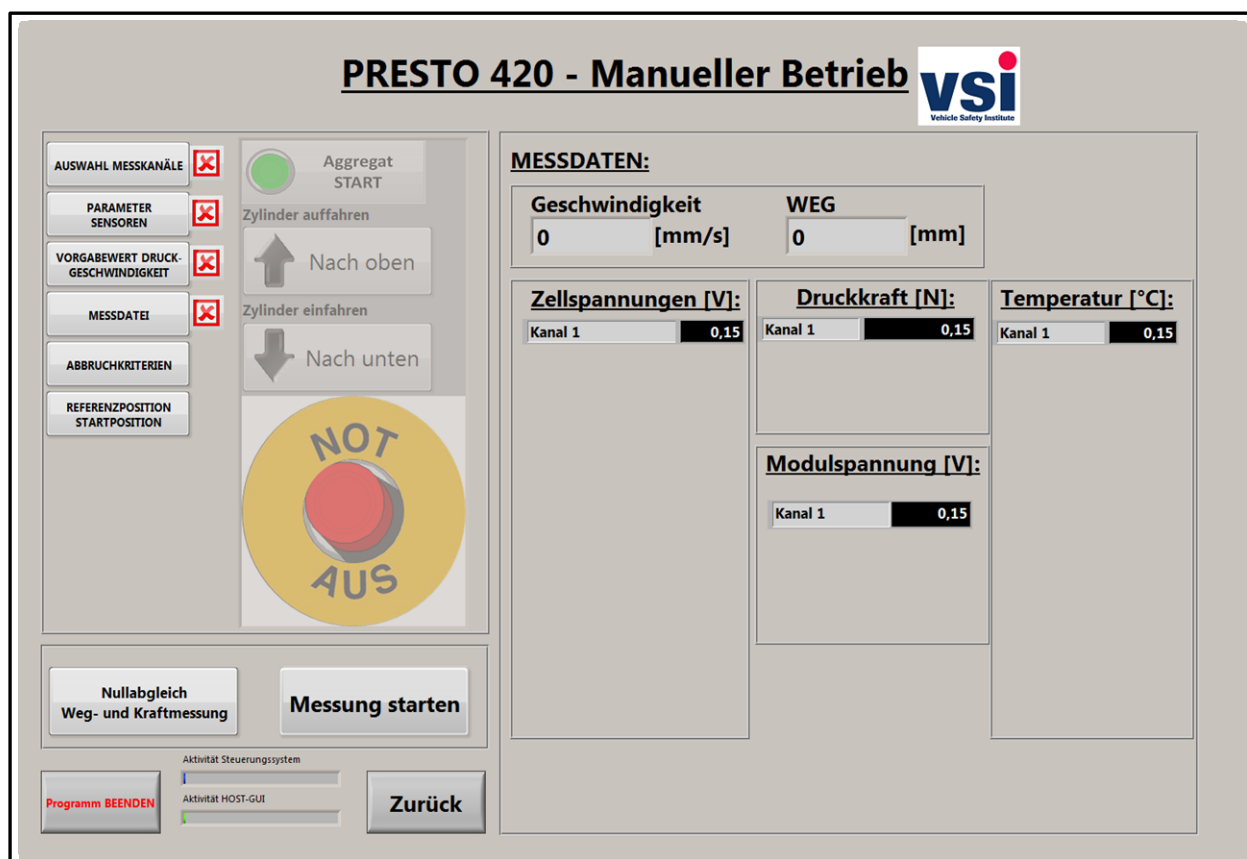


Figure 5: “User Interface – Manual Operations Knauder”

4 Measurement and Control System

In this chapter the used measurement and control system are described. The requirements for the systems are results of experiments, discussions with future machine operators and from (Knauder, 2012).

4.1 Requirements

To specify the needed IO modules and the needed calculation power of the CompactRIO systems, its requirements have to be defined before. Since the system, which is mentioned in chapter 0 was not powerful enough to process the additional data acquisition of a 16 channel thermo element module the software and the hardware were split up into two parts, the measuring system and the control system. The hardware had to be split up because of the needed space for additional modules. The new version of the system should also be capable of the calculation power needed for future features and new data acquisition modules.

In Figure 6: “Program structure PRESTO 420 AUT” you can find the structure of the developed test facility software. There the different program modules as well as the communication paths are shown.

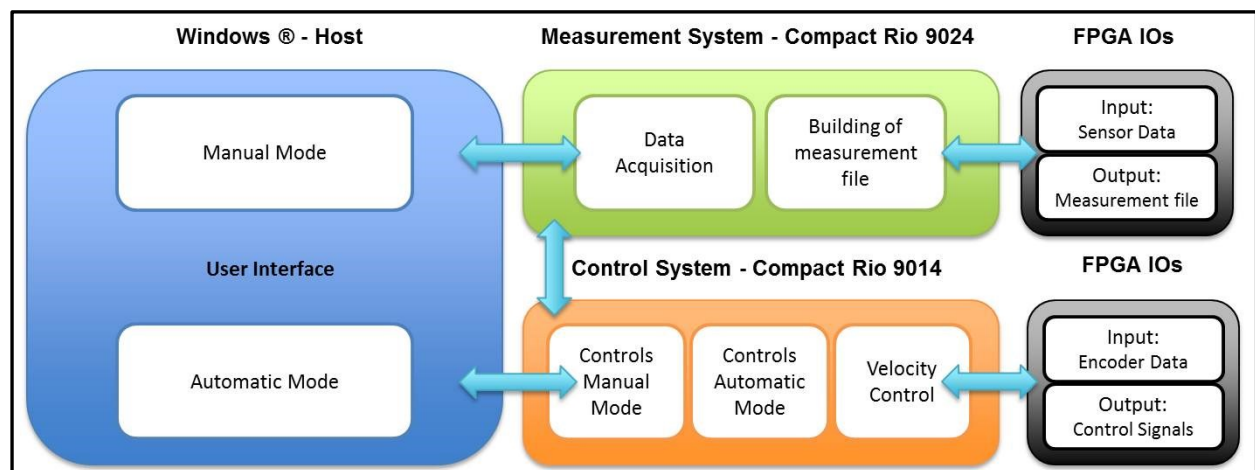


Figure 6: “Program structure PRESTO 420 AUT”

4.1.1 Measurement System

To be flexible with the location of the test facility, a communication via Ethernet was chosen. Since Ethernet is one of the most common bus-systems it is available almost everywhere. The

data acquisition has to be done in real-time. That means that one program run has only a limited time which is specified in front. The compact RIO system guarantees this real time capability. In our case we use a resolution of 10ms to collect enough data of the inputs and outputs. Enough storage space for the collected data is a crucial part. If the system runs out of storage space during a test it would be the worst case. To guarantee enough storage space, a compact RIO system with an adequate internal non-volatile SSD storage has to be used. The possibility to build in more IO modules is given by adding a new CompactRIO controller to the system. Since the “old” control system already faced its performance limits, because of the power needed for an additional data acquisition of a 16 channel thermo element module, a processor with enough cpu-power to provide future expansions has to be chosen. Because the data acquisition system gathers all the force and voltage data from the sensors, the detection of abort reasons⁵ is located there. Therefore it is needed to enable a communication between the measurement and the control system. You can find the structure of the software in Figure 6: “Program structure PRESTO 420 AUT”. In this picture the different communication paths are shown as well.

Requirements Measurement System:

- Ethernet port for communication reasons
- Data acquisition in real-time (way or time based)
- High resolution of the data acquisition
- Enough storage space for the data acquisition
- Additional space for future data acquisition
- Power reserves for expansions
- Communication with the control system (abort reasons)

4.1.2 Control System

For the control system the same thoughts were given to the communication bus as to the measurement system. So we have chosen Ethernet for the same reasons. Real-time is again a critical part of our system. The iteration time has to be guaranteed by the system. To get better

⁵Abort reason: specified limit (short-circuit, force, displacement, time) to either stop a test or move on to the next test stage

and comparable test results, the test setups have to be reproducible. To have enough space for future IOs it is possible to use a bigger frame for the CompactRIO controller, in comparison to the measurement system where already the biggest frame (8 slots) is used. Since the control tasks are not as CPU-intensive as the data acquisition tasks a weaker system could be used. To gather the abort reasons, a faster communication between the two controllers is needed as well.

Requirements Control System:

- Ethernet port for communication reasons
- Automated control of the test facility in real-time
- Facilitation of reproducible tests
- Additional space for IOs
- Power reserves for expansions
- Fast communication with the measurement system (abort reasons)

4.2 Measurement System – CompactRIO 9024

In Figure 7: “Measurement System – CompactRIO 9024” you can see the chosen controller for the measurement system with its different IO modules. In Table 2: “Measurement System components (National Instruments Corporation, 2011)” you can see the technical details of the different modules. Everything was chosen according to the requirements in chapter 4.1.1.

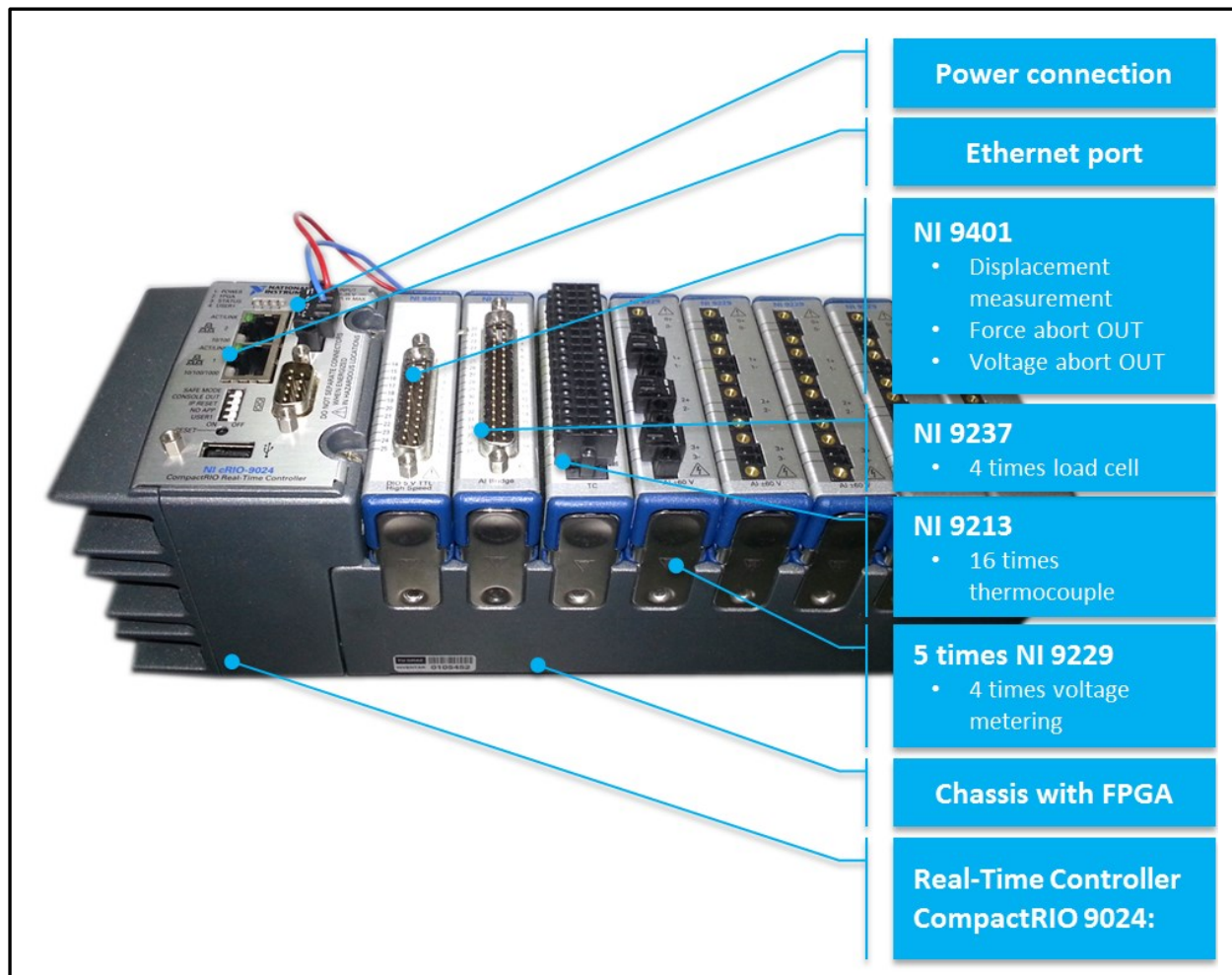


Figure 7: “Measurement System – CompactRIO 9024”

A controller with an 800 MHz processor and 512 MB DDR-RAM is used for the measurement system. So it should be guaranteed, that we will not meet the performance limits with future system expansions. To guarantee enough space, a compact RIO system with an internal non-volatile SSD storage of 4GB is used. There is no possibility to build in more I/O modules in this CompactRIO controller because the 8 slots frame is the maximum. It is only possible to add another controller to the system. It was discovered that with the chosen controller it is possible

to record 40 measurement channels at a maximum sample rate⁶ of 2,5kS/s which can be seen in Chart 2: “CPU-Workload as a function of the sample rate”. The result is a data writing rate of 437,08kB/s which can be seen in Chart 3: “Data-writing rate as a function of the sample rate”. When recording with this sample rate, it is possible to record data for 2,67h, which was calculated with the data writing rate and the size of the storage of the CompactRIO 9024. No possible test scenario should meet this time limit. To enable longer tests, it is still possible to reduce the sample rate to any number.

In Chart 2: “CPU-Workload as a function of the sample rate” the workload is represented as function of the sample rate of 40 recording measurement channels. The system had no other task to perform for as the data acquisition. The CPU-workload represents the workload of the measurement system NI CompactRIO 9024.

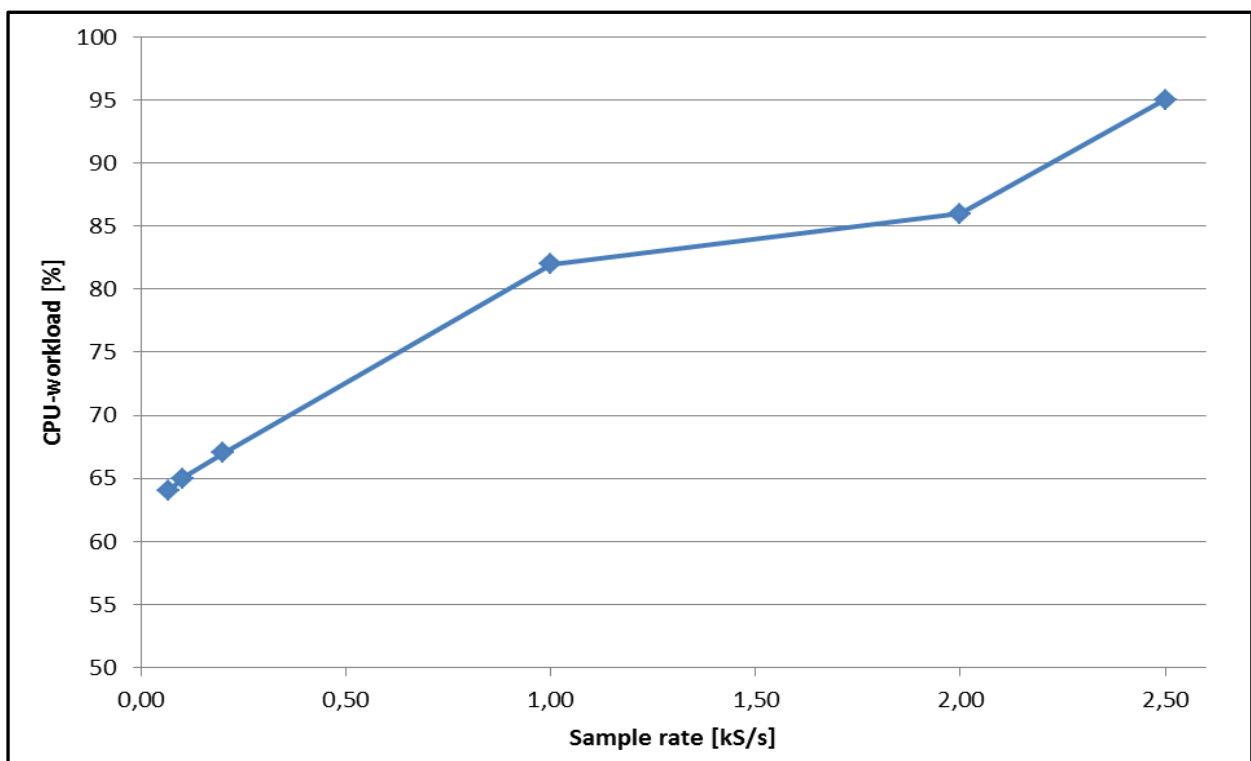


Chart 2: “CPU-Workload as a function of the sample rate”

⁶ Sample rate: the number of samples taken per specified time unit (normally seconds) which are taken from a continuous signal

In Chart 3: “Data-writing rate as a function of the sample rate” the data-writing rate is represented as function of the sample rate of 40 recording measurement channels. It is defined as the file size, of the measurement file, per time unit (seconds).

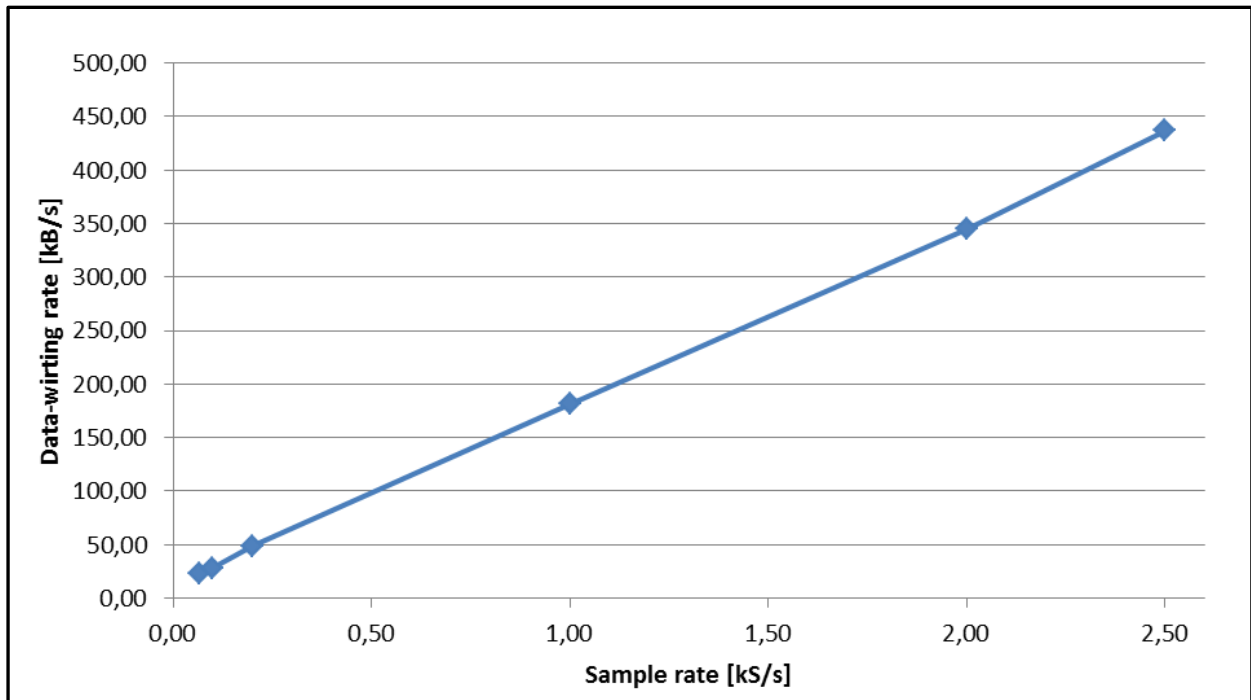


Chart 3: “Data-writing rate as a function of the sample rate”

To ensure a real-time communication between the two systems, four channels of the NI 9401 digital IO module are used as output to transmit abort reasons very quickly. The same module is also used for the displacement measurement. With the NI 9237 the possibility is given to connect four different load cells to our measurement system. To measure temperature the NI 9213 is able to connect 16 thermocouples with the different types J, K, T, E, N, B and R. The voltage metering is done via five NI 9229 modules which are able to meter voltages between +60V and -60V. It is also important to mention that the channels are electrically separated to not distort each other. So we have 20 voltage metering inputs which should be enough for all test cases.

In Table 2: “Measurement System components (National Instruments Corporation, 2011)” you can find a list of the technical specifications of the used hardware for the measurement system.

Components:	Name, Information:	Data:	Unit:
Real-time controller:	NI 9024		
	Clock rate:	800	MHz
	Working memory:	512	MB DDR2RAM
	Storage:	4	GB
Chassis:	NI 9114		
	Slots:	8	#
FPGA:	Xilinx Virtex-5 LX50		
Input module:			
Digital IO module:	NI 9401	TTL-able (5V/TTL)	
	Channels:	8	#
	Clock rate:	100	ns
Force metering module:	NI 9237	Bridge module	
	Channels:	4	#
	Resolution:	24	bit
	Sample rate:	50	kS/s
Thermocouple module:	NI 9213		
	Channels:	16	#
	Resolution:	24	bit
	Sample rate:	75	S/s
Voltage metering module:	5 times NI 9229		
	Channels:	4	#
	Resolution:	24	bit
	Sample rate:	50	kS/s

Table 2: “Measurement System components (National Instruments Corporation, 2011)”

4.3 Control System – CompactRIO 9014

In Figure 8: “Control System – CompactRIO 9014” you can see the chosen controller for the control system with its different IO modules. In Table 3: “Control System components (National Instruments Corporation, 2011)” you can see the technical details of the different modules. Everything was chosen according to the requirements in chapter 4.1.2.

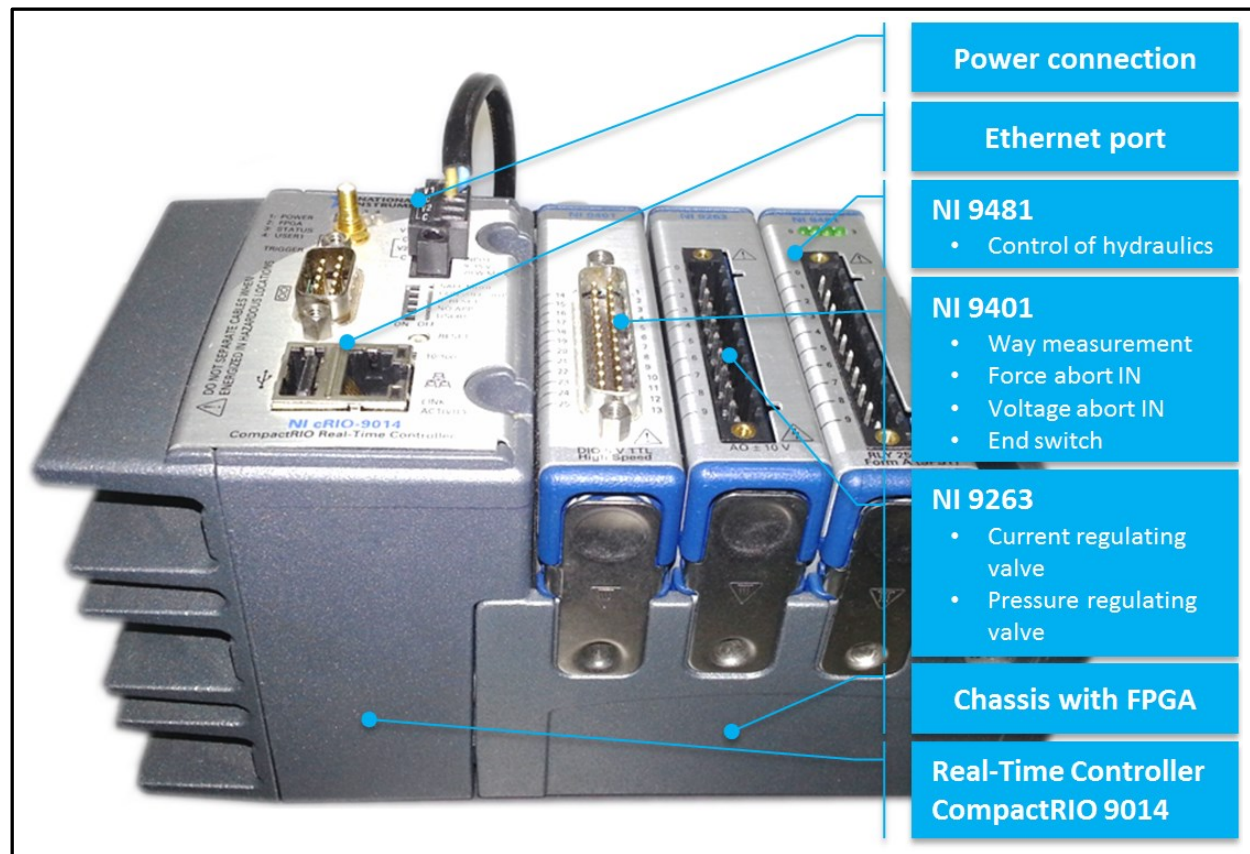


Figure 8: “Control System – CompactRIO 9014

Since the control tasks are not as CPU-intensive as the data acquisition tasks, and because of cost reasons, we have decided to take the old data acquisition controller for this new working field. The possibility to build in more IO modules is given by using a bigger frame (8 slots) for the CompactRIO system. To ensure a real-time communication between the two systems four channels of the NI 9401 digital IO module are used as input to collect the incoming abort reasons from the measurement system. The same module is also used for the way measurement. To control the hydraulic power unit we use the NI 9481 module. Speed regulation and pressure regulation are done via two current regulating valves and a pressure

regulating valve which are controlled by a NI 9263 module. You can find a scheme of the hydraulic system in Figure 36: “Hydraulic system scheme new – PRESTO 420”.

In Table 3: “Control System components (National Instruments Corporation, 2011)” you can find a list of the technical specifications of the used hardware for the control system.

Components:	Name, Information:	Data:	Unit:
Real-time controller:	NI 9014		
	Clock rate:	400	MHz
	Working memory:	128	MB-DRAM
	Storage:	2	GB
Chassis:	NI 9113		
	Slots:	4	
FPGA:	Xilinx Virtex-5 LX50		
<i>Input module:</i>			
Digital IO module:	NI 9401	TTL-able (5V/TTL)	
	Channels:	8	
	Clock rate:	100	ns
<i>Output module:</i>			
Analogue output module:	NI 9263	Power output ($\pm 10V$)	
	Channel:	4	
	Sample rate:	100	kS/s
	Resolution:	16 bit	
Relay digital module:	NI9481	30VDC(2A), 60VDC(1A), 250VAC (2A)	
	Channels:	4	

Table 3: “Control System components (National Instruments Corporation, 2011)”

5 Programming Environment – LabVIEW™ 2012

In this chapter a short introduction into the programming environment LabVIEW™ 2012 is given. The main differences between LabVIEW™ and text based programming languages are covered out and a short example is shown. This chapter is based on (National Instruments Corporation, 2013) and (National Instruments Corporation, 2011).

5.1 What is LabVIEW™

„Laboratory Virtual Instrumentation Engineering Workbench“ is a programming environment for various different hardware components. It is especially developed for measurement, control and feedback control systems. The programming takes place in a graphical based way considering the dataflow principle. This fact ensures faster and easier programming. It is also the main difference to text based programming languages like C++, C# and others. The “classical coding” is not needed anymore because the programming is done with a drag & drop of functions. The main tasks of LabVIEW™ are data acquisition and data processing. In Figure 9: “Difference between LabVIEW™ and C++” you can see an example for the differences between a text based and a graphical based programming language.

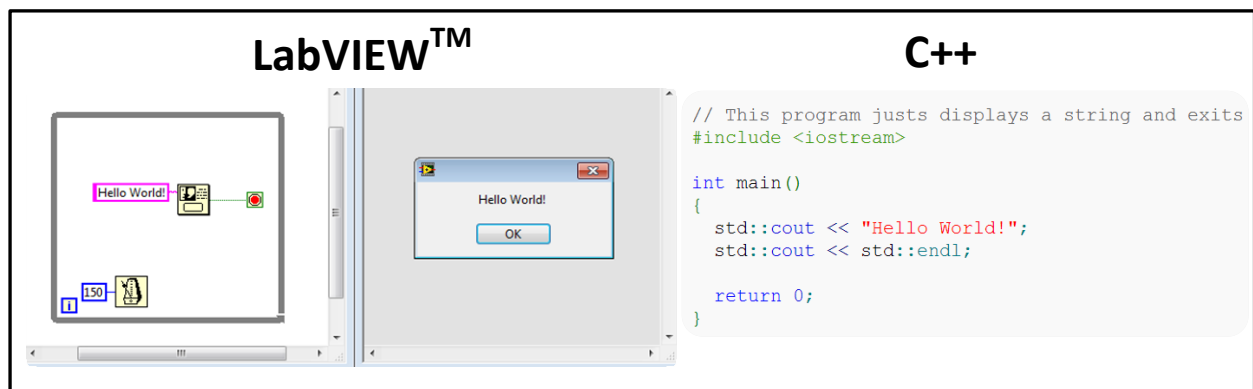


Figure 9: “Difference between LabVIEW™ and C++”

A program/subroutine is called VI or Virtual Instrument. It consists of three layers:

- the front panel
- the block diagram
- the connector panel

All the visualization and interaction with the user happens at the front panel, the programming happens at the block diagram. In the block diagram the programmer connects different nodes via wires to a source code. The connector panel is used to represent VIs in block diagrams of others, which means in calling VIs.

The up to date version is LabVIEW™ 2012 with which the software for the PRESTO 420 was written. LabVIEW™ is also capable of parallel program execution.

5.2 Programming levels in CompactRIO real-time systems

Figure 10: “Programming levels in CompactRIO real-time systems (National Instruments Corporation, 2011)” shows the different programming levels and the internal communication for CompactRIO controllers. The programming is split up into three levels:

- Host PC
- Real-Time Controller
- FPGA⁷

The communication between the host pc and the real-time controller is performed via Ethernet. The communication between the real-time controller and the FPGA is performed via an internal PCI bus.

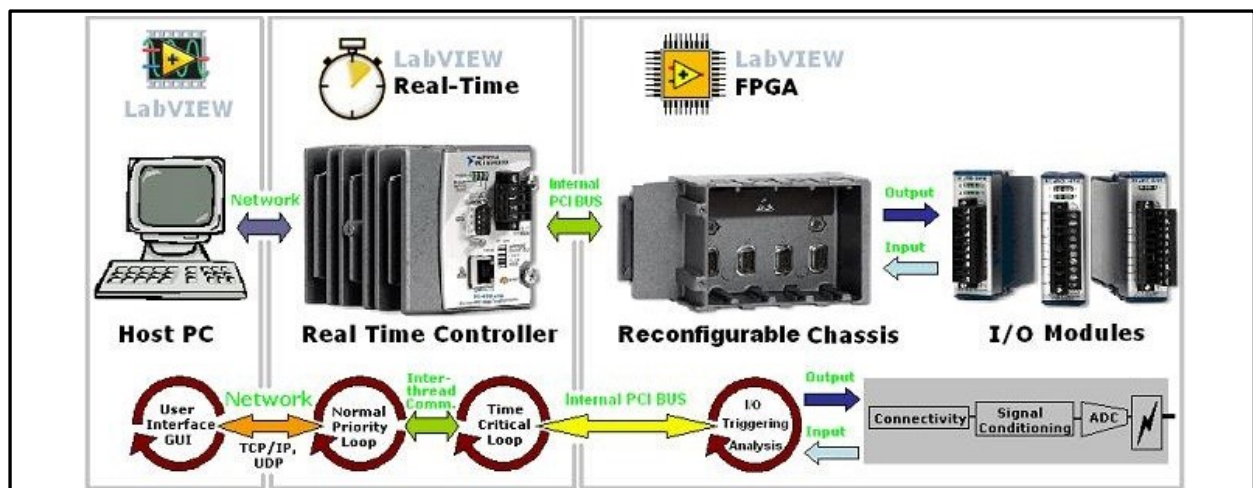


Figure 10: “Programming levels in CompactRIO real-time systems (National Instruments Corporation, 2011)”

⁷ Field-Programmable Gate Arrays

The separation into different levels gives you the possibility to make several independent program parts. For example the real-time controller is able to run without a connected Host PC. There is also the fact that the different programming levels have different calculation speeds.

5.2.1 Level 1: Host PC

The first level, the host pc, is the connection to the programmer/user per user interface (GUI⁸). There the FPGA and real-time controller are programmed via LabVIEW™. The communication to real-time controller happens over network, exactly TCP/IP⁹ & UDP¹⁰, through the exchange of network variables. Right after the programming the VIs are compiled and distributed to the addressed systems.

5.2.2 Level 2: Real-Time Controller

At the second level, the real-time controller, the whole control logic takes place. It is independent from the host pc that means a CompactRIO is a standalone system. Here you can separate between normal priority and time critical loops. One run of a time critical loop is finished after a pre-defined time. Normal priority loops are not real-time able. The communication with the FPGA happens via an internal PCI¹¹ bus.

5.2.3 Level 3: FPGA

The third level, the FPGA with its reconfigurable chassis, can be connected to various IO modules. The FPGA is the fastest of the three levels. It is possible to process data with a speed of 25ns. The FPGA is the place where the IO modules are addressed. It sets the value of the modules outputs and gets the data of the modules inputs. There also certain program algorithms, which have a need for a very high calculation speed, can be stored.

⁸ GUI: Graphical User Interface

⁹ TCP/IP: Transmission Control Protocol / Internet Protocol

¹⁰ UDP: User Datagram Protocol

¹¹ PCI: Peripheral Component Interconnect

6 Software - PRESTO 420 AUT

In this chapter the programming of the whole software for the PRESTO 420 will be described including the overworked manual operations and the completely new automated operations.

The requirements for such kind of software have to be explained as well. Flowcharts will be shown to give a better understanding of the source code logic.

Since the data acquisition part of the software was almost carried over from (Knauder, 2012) - it received only a few necessary changes - this will not be part of this thesis.

The workload of the development software is split up into four parts: analysis of the existing software, defining of requirements, overworking of the manual operations and programming of the new automated operations. This will also include the implementation of new features, a tensile test and a velocity feedback control as you can see in Figure 11: “Workload PRESTO 420 software development”.

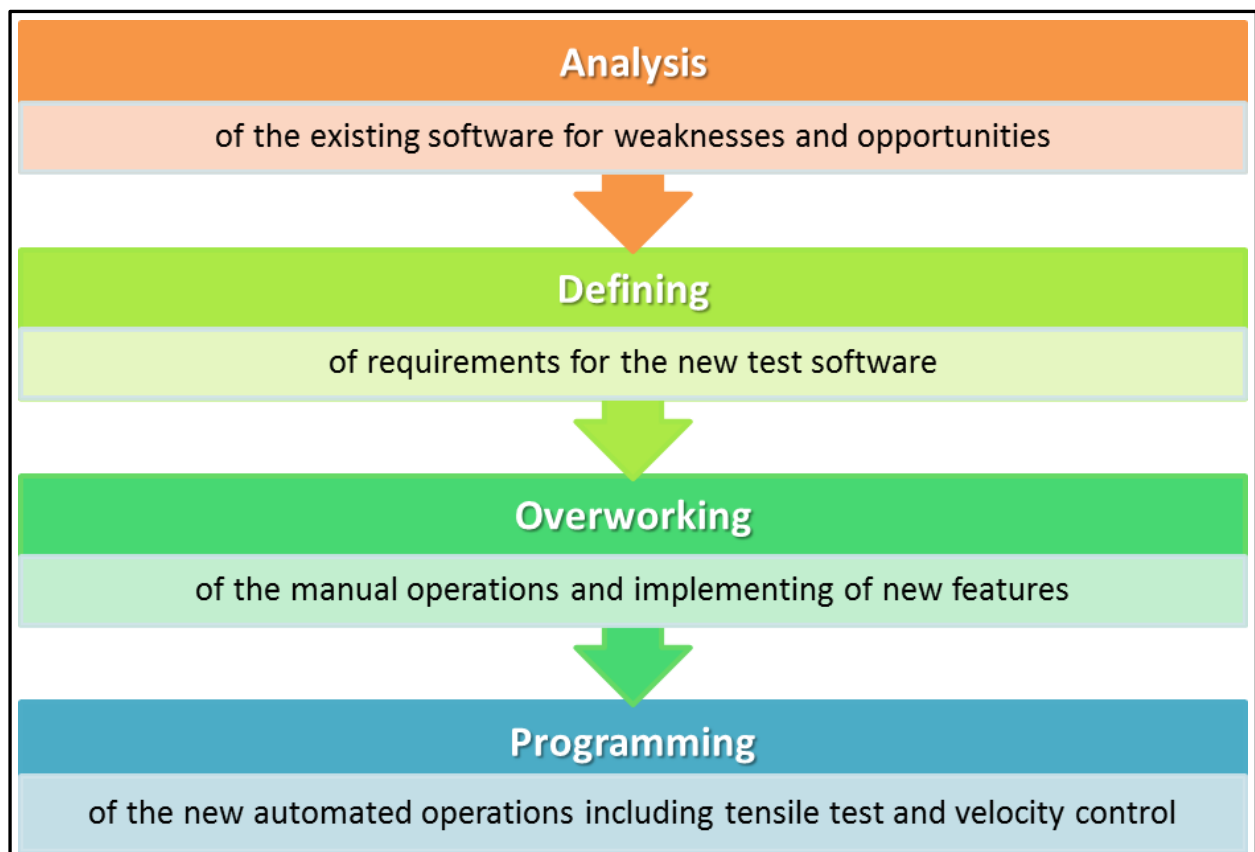


Figure 11: “Workload PRESTO 420 software development”

According to (Wörn, et al., 2005) the task of automation technology is to control and to monitor technical processes. Examples are digital cameras, washing machines, air conditions,

advanced driver assistance systems to prevent accidents and others. An important aspect of the process control is to control a facility with a high degree of autonomy. Thereby necessary function and security requirements should be fulfilled. An automated facility should replace humans in operative processes and should exculpate them from hard physical and psychical risky work. Lots of processes have no other possibility as to be controlled by a machine because of its high complexity or requirements in terms of accurateness or real-time ability. Humans took the job to monitor the process in a strategically way from a superior position.

To **control** means, according to a defined goal or to an algorithm, to influence or to change a process and its cycle via actuating and control elements.

To **monitor** means, to make data acquisition of the important states of the process via sensors and to present this data to the operator in an accurate way. Security algorithms have to interpret critical process states automatically to change the process in real-time or to stop the facility in an emergency case.

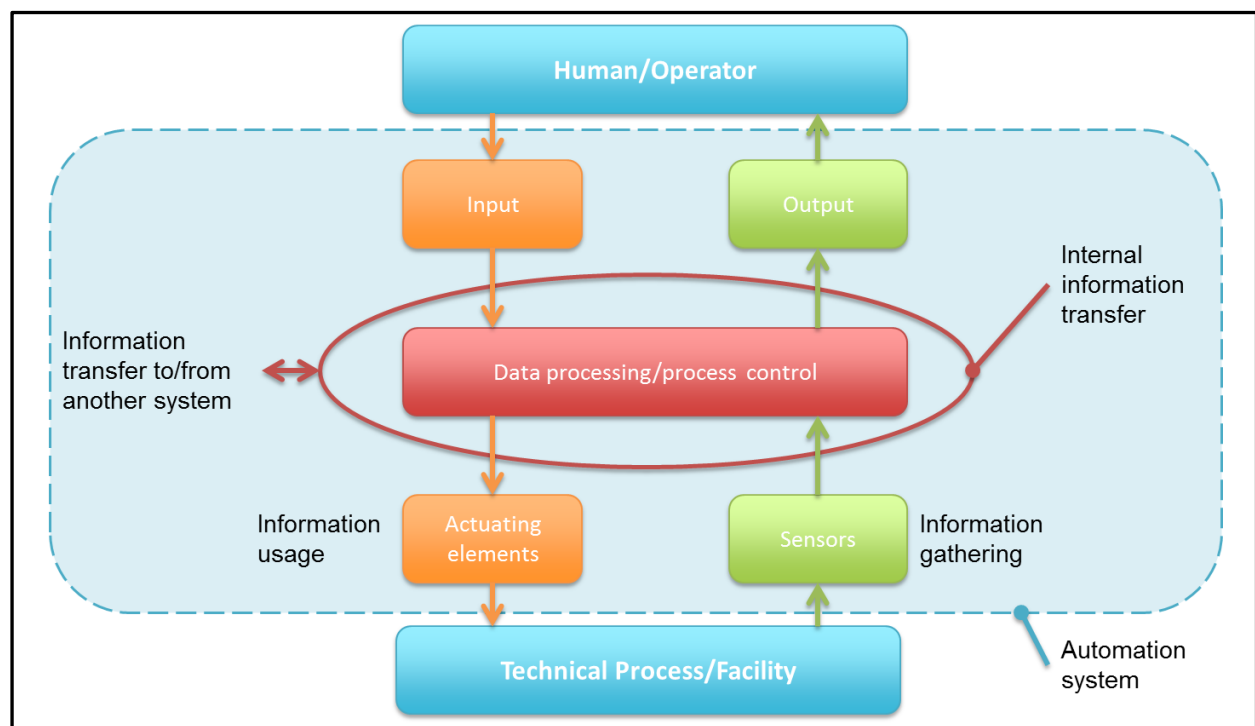


Figure 12: “Basic principle of automation (Wörn, et al., 2005)”

Figure 12: “Basic principle of automation (Wörn, et al., 2005)” shows the basic principle of automation. An automated system consist of a process which has to be automated (and the corresponding facility) and the automation system. The automation system has to gather the

information from sensors. It processes according to a strategy which is provided by the user. The strategy is realized via a program algorithm. Control commands influence via actuating elements like valves, pumps, engines and others the process directly.

In our system the strategy of a test can be influenced by the user with selecting the test sequence and its abort reasons (as mentioned in earlier chapters: short-circuit, force, time, displacement). The test facility fulfils the test exactly as defined. The scheme for the selected test stage (pressure test, tensile test, waiting stage, test start, test stop) used in the test sequence is previously defined in the program algorithm and cannot be changed during the usage of the PRESTO 420. For example in the stage “pressure test” the system activates via a digital relay module the upward movement of the piston. The desired test speed is gathered from the users input and controlled by the feedback controller. It sets the voltage of the current regulating valve for the pressure mode. The system also sets the pressure force according to the users input via regulating the voltage of another current regulating valve.

6.1 Analysis of the existing software

In this subchapter the points for improvements, which were found out, will be listed and explained. Interviews with operation staff and people who are involved in this project were done to get to know other opinions and new thoughts to finally develop a software with a high usability & reliability. The discovered improvements will be separated into the sections usability specific, test specific and software errors, as you can see in Figure 13: “Software specific improvement potential PRESTO 420”.

Usability specific	Test specific	Software errors
<ul style="list-style-type: none"> • no optical feedback when reaching an abort reason • no automated reset of zero adjustment for force and way measurement • no fast traverse in manual mode • no backward movement possible when facing the force abort reason in manual mode • only manual input of loadcell properties • no possibility to illustrate measurement data live in chart form • no automated operations 	<ul style="list-style-type: none"> • no speed control • no short-circuit abort reason detection • no tensile test possible • no communication observation • no obligate input of abort reasons • no exact reference and start position run 	<ul style="list-style-type: none"> • force abort reason only working for one load cell • no possibility to activate or deactivate laod cells • no GUI command handshake with control and measurement system

Figure 13: “Software specific improvement potential PRESTO 420”

In the next subchapters the different improvements, which were found out, are going to be explained in detail. To understand further explanations in a better way it should be referred to Figure 5: “User Interface – Manual Operations Knauder”.

6.1.1 Usability specific improvement potential

Since usability is a big deal in software design, a main focus of this thesis is to develop a software with a high degree of usability. Potential sources of input- or user errors should be brought to a minimum. These errors can result in an incorrect data output or machine defects. It is also a question of costs and time if tests have to be repeated because of user errors. To ensure a low failure rate several features have been built into the software, as listed below.

No optical feedback when reaching an abort reason

The user should immediately see why and that the machine stopped its movement. Here it should be mentioned that, since we have very low movement speeds in quasi static tests (<1mm/s), it is often hard to see if the machine has stopped. This can happen in various ways when programming with LabVIEW™. In manual mode it could be simply implemented via an LED for every abort reason which changes its colour to red when a reason is met. The test is not automatically stopped when facing an abort reason, only the movement is blocked. In automatic mode the test should be automatically stopped with an appearing popup which informs the operator why the machine has stopped and what has happened.

No automated reset of zero adjustment for force and way measurement

Since it is necessary to make a zero adjustment for the force and way measurement right before a test it should not be allowed to hit the start button before that. Zero adjustment means, that the way- and force measurement is set to 0 before starting a test. Another problem is when the zero adjustment button keeps being activated while the test has already been started. This results in incorrect data. So the implementation of a mechanism which automatically resets the button when the reset information went down through the programming levels to the FPGA would be a benefit. This could happen in the same way in automatic and manual mode.

No fast traverse in manual mode

In manual mode a fast position change of the piston is needed very often. To overcome the step of going into the options and change the voltage of the current regulating valve a button

for the fast traverse mode could be implemented. This would significantly improve the operation speed.

No backward movement possible when facing the force or displacement abort reason in manual mode

When the value of force or displacement was above the defined maximum no more movement was possible. This resulted in the fact that the machine was forced to a standstill and had to be rebooted. The force abort reason check is performed all the time for security reasons, not only while performing a test. This is done to protect the measurement instrumentation against damage which could result in unwanted costs. A mechanism, which allows the operator to move the piston in the other direction (pressure test: downward movement, tensile test: upward movement), would solve this problem.

Only manual input of load cell properties

Data input errors are one of the most common errors when operating a software. If you have to convert units to enter properties of load cells this is a big error source which could easily be prevented. You have to enter data for the supply voltage, the measurement range and the characteristic value. Especially the characteristic value often exists in many different units like mV/V, V/V or others. A database, which covers all at the Vehicle Safety Institute existing load cells, would decrease the possibility of the failure occurrence - there is still a rest risk because the specifications can be incorrectly inserted into the data base. You just have to choose the serial number of the used load cell, so the software will automatically load the defined properties.

No possibility to illustrate measurement data live in chart form

The possibility to show the measurement data in live chart form is just an additional feature, which gives the operator the opportunity to change the way of the data illustration. This cannot result in a failure. One benefit of this option is to recognize tendencies of the measurement charts in a very fast way. So malfunctions of the measurement equipment can be seen easier.

No automated operations

To have the possibility to make reproducible tests is a main focus of this thesis. An automatic test mode would help with this point. To setup a defined test sequence out of three independent modes (tensile test, pressure test, and waiting phase) with specific abort reasons would be the solution.

6.1.2 Test specific improvement potential

For a pressure and tensile test facility it is clear that “testing” is the core function. Repeatable test setups and constant movement speeds are crucial to get reproducible metering data. It should be possible to perform several exactly the same tests.

No speed control

As mentioned before, constant movement speeds are crucial. Only setting a constant voltage for the current regulating valve is not enough to get a constant feed rate. An automated control mechanism for the feed rate, which is feasible with LabVIEW™, has to be implemented. The programming of a feedback control strategy would highly influence the reproducibility of tests.

No short-circuit abort reason detection

When testing batteries short-circuit is a characteristic value. Crushing cells till short-circuit occurs is often claimed, see (Knauder, 2012). A mechanism, which automatically detects the dropdown of the voltage under a certain limit, is needed to stop a test.

No tensile test possible

For testing new material the behaviour under pressure is not the only interesting thing. The material properties under tension are as well a matter of interest. To realize tensile testing with the PRESTO 420 software implementation is not the only needed task, it is also necessary to make mechanical changes which are explained in a following chapter.

No communication observation

The communication between host-pc, control and measurement system is needed to guarantee a working system with a working data acquisition. To perform tests a constant communication is unavoidable. When the communication during a test breaks information should be sent to the operator and the test should be stopped automatically to avoid problems. Another need of a communication observation is given because no communication with the control system can result in unpredictable movement.

No obligate input of abort reasons

Without setting values for the different abort reasons it is not possible to check them. So an obligate input of abort reasons should be implemented to ensure its functionality.

No exact reference and start position run

For reproducibility reasons it is necessary that the test facility always uses the same reference position and fulfils an accurate movement to the start position. Therefore an end switch has to be built-in which defines the reference position exactly.

6.1.3 Software errors

To guarantee a reliable software, errors have to be eliminated as soon as they occur. Discovering such errors, which is not always easy, often requires other approaches to operate with a software. So user tests, also called beta tests, help finding out errors which can result in various problems. A few errors were already known because of the first version of the software. The new implemented features are also a source of new errors to occur which have to be tested.

Force abort reasons only working for one load cell

The force abort reason check was only implemented for one load cell. Since the possibility to attach four load cells to the device is given this is not enough. The check should work for all attached and activated load cells.

No possibility to activate or deactivate load cells

It should be possible to activate and deactivate load cells via software. Maybe in some time the case occurs that a load cell is connected but not needed for the force abort reason check. The easiest example is when only one load cell is used for a test, so the other inputs have to be deactivated.

No GUI command handshake with control and measurement system

To ensure that commands reached its goal a handshake with the target is unavoidable. As a handshake and algorithm is defined which waits for the answer of a system when a command is sent, if it was received, before the execution of the command happens. A Handshake should be implemented for all crucial commands like “start”, or “zero adjustment”.

6.2 Requirements - PRESTO 420 AUT

This subchapter describes the requirements for the software. Since most of the requirements were already carried out by (Knauder, 2012) only missing ones will be added. Only the control system is described in detail, the requirements of the measurement system will only be resumed.

6.2.1 Measurement System

In Table 4: “Requirements for the measurement system (Knauder, 2012)” all for the measurement system of the PRESTO 420 needed requirements are listed.

Requirement:	Remark:
Measurement System:	<ul style="list-style-type: none"> • Modular program structure • Easily extendible
Data acquisition:	<ul style="list-style-type: none"> • Exact relate ability of the measurement channels • Use of the real-time ability of the system and the FPGA for the data acquisition
Way-controlled	<ul style="list-style-type: none"> • Use of the resolution of the digital scale of length to trigger the data acquisition of all measurement channels
Time-controlled	<ul style="list-style-type: none"> • Use of the real-time ability of the system for time deterministic data acquisition

Table 4: “Requirements for the measurement system (Knauder, 2012)”

The measurement system itself has to be programmed in a modular way. This will keep the quality of the code on a high level and makes every new “program module” separately testable. To be easily extendible is also necessary because you never know which changes have to be made in the future. Every measurement channel has to be named in a unique way to be able to distinguish between them.

Way-controlled data acquisition

The way-controlled data acquisition mode is needed for all tests (pressure or tensile tests) where movement happens. In the way controlled data acquisition mode the resolution of the digital scale of lengths is used as trigger for the data acquisition of all selected measurement channels. That means at every position trigger the system collects the according data from the other sensors, like voltage, force and temperature.

Time-controlled data acquisition

The time-controlled data acquisition mode is needed for stand tests¹² and for waiting phases where no movement occurs. Since the way-controlled data acquisition is switched to time controlled when no movement occurs it is also possible to collect suitable data in such situations.

6.2.2 Control System

In Table 5: “Requirements for the control system extended” all for the PRESTO 420 control system needed requirements are listed.

Requirement:	Remark:
Control System:	<ul style="list-style-type: none"> • <i>Modular program structure</i> • <i>Easily extendible</i> • <i>High usability</i> • <i>Automated velocity control</i>
Abort reasons:	
Maximum force	<ul style="list-style-type: none"> • Input of the maximum pressure or tensile force • Measuring via load cells • Interaction between control and measurement system
Maximum displacement	<ul style="list-style-type: none"> • Input of the maximum movement distance • Measurement via digital scale of length
Waiting time	<ul style="list-style-type: none"> • Input of the waiting time • Measurement via time stamp of the cRIO
Short-circuit	<ul style="list-style-type: none"> • Dropdown of the selected voltage channels under 0,1V for more than 5 program cycles¹³
Manual Mode:	<ul style="list-style-type: none"> • Initiation • Principle tests • Adjustment work
Automatic Mode:	<ul style="list-style-type: none"> • High reproducibility of tests • Moving to a start or reference position • Test out of three test phases free configurable: tensile-, pressure-, waiting phase • Minimum user errors

Table 5: “Requirements for the control system extended”

¹² Tests where a defined force is brought on a test block for a specified time

¹³ Program cycle: one run through the whole software algorithm

In general the control system has to do all the control tasks where hardware is addressed. A modular program structure has to be the goal for the same reasons as in the measurement system. An easy expansion of the system has as well to be guaranteed for new hardware or other stuff. Since we have guidelines for standardized tests – one of those is described in the end of this subchapter - we would like to fulfil with the PRESTO 420 system certain functionalities have to be ensured. To have the possibility to reproduce these test setups an automated velocity control would be a big benefit. For the realization of such tests different abort reasons for different test stages are needed. In our software we have to use four of them:

1. Maximum Force
2. Maximum Displacement
3. Waiting Time
4. Short-Circuit

Abort reason – Maximum Force

The maximum force abort reason is needed to not destroy the used load cells, to detect possible collisions and to check the applied force for a defined maximum. For example, in battery testing the maximum of the force is 1000 times the module's mass according to (Doughty, et al., 2006). This abort reason is checked during all operations. The user has to type in the force limit which is the maximum force abort reasons and to select the load cells which should measure the necessary data. Here an interaction between the control and the measurement system is needed, because the measurement system gets the data from the load cells and the control system is performing all the movement and control operations. When a limit is reached the measurement system sends the Boolean value true over an NI 9401 output to the control system. This has to be done because of the very fast reaction of the software to an input from a module. This is much faster than sending such system critical values over Ethernet which is not capable of a real-time function. Through that hardware defects can be excluded.

Abort reason – Maximum Displacement

The maximum displacement abort reason is needed to define a fixed movement distance for tests. In battery testing a displacement of 15% and 50% of the modules height has to be

performed according to (Doughty, et al., 2006). This is only checked during a tensile or pressure test. It doesn't need an interaction between the measurement and control system because the control system also receives encoder data. This is done because of the same reason as in the abort reason maximum force, the time problem when sending data over Ethernet. The receiving of the correct encoder data is time critical in the control system because all the steering of the moving hardware is done with this data.

Abort reason – Waiting Time

To give the possibility to perform stand tests and a waiting phase during other tests a defined waiting time is needed. In battery testing a waiting time of 5 minutes is used for controlled crush tests according to (Doughty, et al., 2006). The checking of this abort reason is performed per comparing time stamps of the CompactRIO system which is done by the control system.

Abort reason – Short-Circuit

For battery testing it is often needed to perform a test till short-circuit occurs. That means that the abort reason is met when the voltage of a selected cell drops down under the limit of 0,1V for more than five program cycles. This is checked by the measurement system since the voltage measuring is done there.

Manual Mode

The requirements for the manual mode are quite simple. This mode exists to perform the initiation of the test facility (to see if the system and all its functions work properly). Principle tests are also a case of used for this mode. When a material is tested the first time here you can figure out the right settings for all the measurement equipment etc. Adjustment work should also be performed with this mode.

Automatic Mode

A major requirement for the automatic mode is to reach a high reproducibility of tests. This is crucial to perform standardized tests for many times. For these standardized tests the software has to have the possibility to define test sequences out of different stages. These stages are:

- Tensile phase

- Pressure phase
- Waiting phase

To have the possibility to perform such test sequences automatically we always have to guarantee the same start and reference position. Another requirement of the automatic mode is to have minimum user errors which were explained in a previous chapter.

Test procedure Controlled Crush test according to (Doughty, et al., 2006)

As previously mentioned a standardized test will be described in this part. This is done to give a better understanding of the requirements and features of our developed system.

As you can see in Figure 14: “Test procedure Controlled Crush test (Doughty, et al., 2006) this test occurs in two stages: “The first stage (displacement control) is a displacement of 15% of the module’s height, which is held for five minutes. The second stage (force control and displacement limit) is limited by either a 50% displacement of the module’s height or a force of 1000 times the module’s mass, whichever condition occurs first is held for five minutes.” (Doughty, et al., 2006)

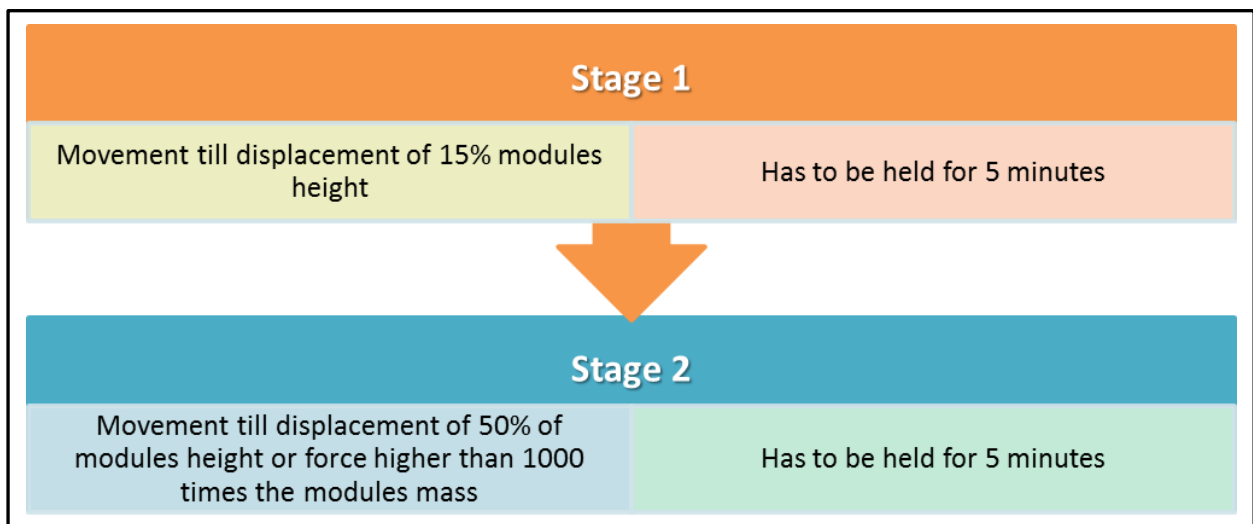


Figure 14: “Test procedure Controlled Crush test (Doughty, et al., 2006)

6.3 Program Structure - PRESTO 420 AUT

To meet all the prerequisites and requirements a suitable program structure has to be found. In Figure 15: “Program structure and changes PRESTO 420 AUT” you can see the chosen structure and what has been changed or stayed the same since (Knauder, 2012). As you can see the modular structure of the software was a main design criterion.

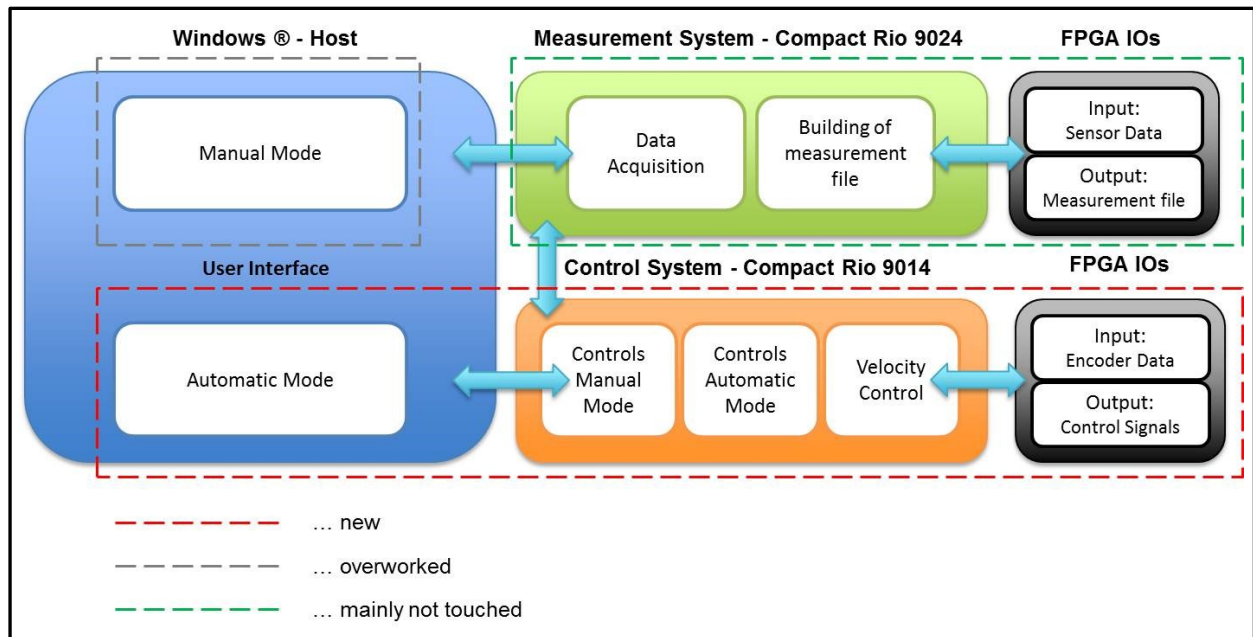


Figure 15: “Program structure and changes PRESTO 420 AUT”

The big benefit of this modular program structure is the changeability of single modules without changing the logic of another module. It is clear that some changes may affect other modules, for example when network variables or the data structures are changed. That means, if a variable is changed, which is used in more than one program part (network variable), it is clear that this affects all the program parts where the variable is used. But in general one module is change- and testable in its module borders.

The automatic mode GUI, the control and automated control module and its FPGA are completely new. The manual mode GUI was adapted to the new needs and extended with new features. The measurement system and its FPGA only received a few needed changes in its algorithms since in the beginning the data acquisition and the control & automated control modules were located on the same CompactRIO system. The split up of the measurement and control system on different CompactRIO controllers also solved the performance problem and now allows further expansions of the software.

6.3.1 Windows® Host - GUI¹⁴

The manual and automatic modes are implemented into the GUI. They have to communicate with network variables with the control and measurement system via Ethernet. Its task is to be the gateway between the operator and the PRESTO 420. Here the operator has to enter all the necessary inputs.

6.3.2 Measurement System Modules

The measurement system with its modules has to communicate with its FPGA and the control system. All measurement related tasks are up to this system. For more detailed information respective data acquisition please read (Knauder, 2012).

Data Acquisition Module

The task of the data acquisition module is to gather all the data from the sensors (voltage, force, temperature, and encoder) and hand it over to a real-time FIFO¹⁵ which communicates with the measurement file module as you can see in Figure 16: “Functions of the data acquisition (Knauder, 2012)”. The FIFO is needed because it is not possible to write the measurement file at the same speed as the data acquisition gathers data.

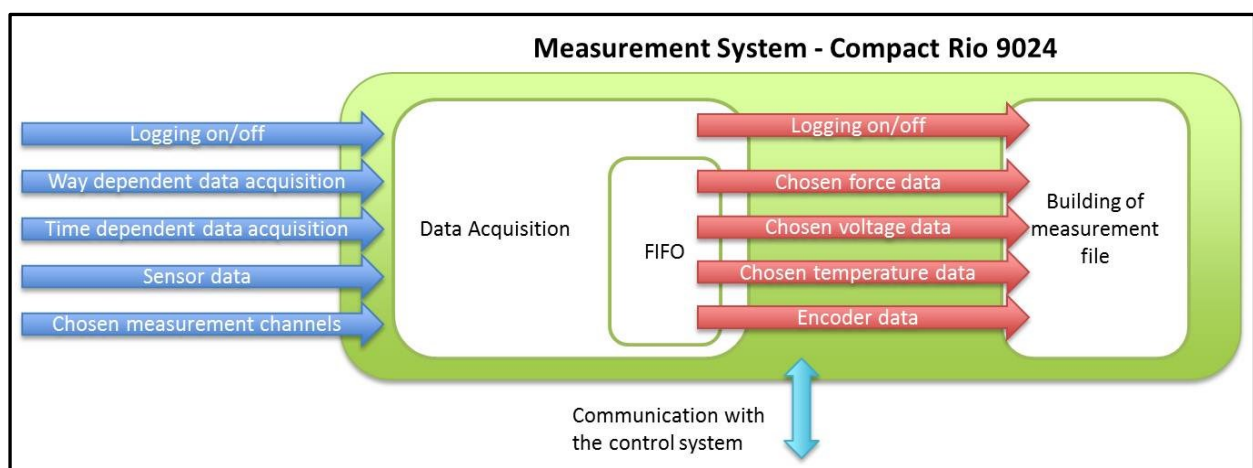


Figure 16: “Functions of the data acquisition (Knauder, 2012)”

¹⁴ Graphical User Interface

¹⁵ First in First out

Another task of the data acquisition module is the checking of the abort reasons for force and short-circuits.

Measurement File Module

As mentioned before the measurement file module receives data from a FIFO. With this data the measurement file is built and stored at the internal storage of the measurement system.

Measurement System FPGA

The FPGA is the gateway between the sensor modules and the data acquisition. Its task is to forward this data to the data acquisition as well as to trigger the way measurement. Another job of the FPGA is to perform parts of the signal processing. All of this is done because of the high performance of an FPGA.

6.3.3 Control System Modules

Since this was one of the main parts of this thesis the next chapters will be described more detailed.

The control system with its modules has to communicate with its FPGA and the measurement system. All control related tasks are up to this system. In Figure 17: “Functions of the control modules” you can see the functions of the control modules with its in- and outputs.

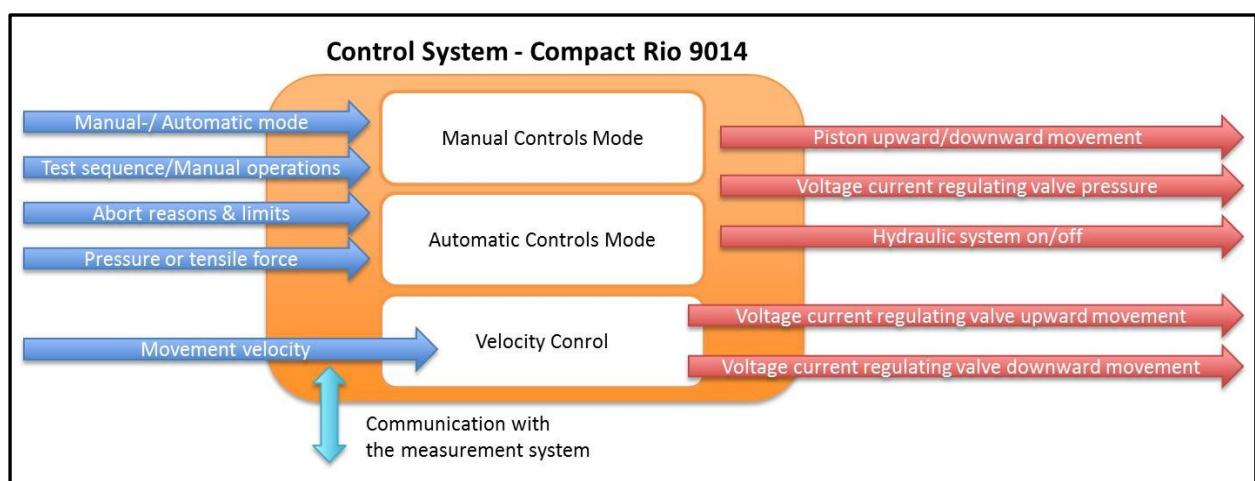


Figure 17: “Functions of the control modules”

Manual Controls Mode Module

When the manual mode is chosen via the GUI the task of the manual controls mode module has to translate the operator's commands to the PRESTO 420. It addresses the hydraulic system and the piston movement. In the manual mode the abort reason for distance is checked as well.

The output of the commands for hydraulic system on/off and piston up-/downward movement is done with the NI 9481 electromechanical relay module. The output of the commands for the voltage of the current regulating valve for the pressure force is done via the NI 9263 $\pm 10V$ analogue output module. In the manual mode these controls have to be performed by the user.

Automatic Controls Mode Module

When the operator has chosen the automatic mode, its task is to perform an automated test sequence with the user specified properties. The defined abort reasons for distance and time are checked here as well.

In automatic mode a reference run has to be performed in front of the test as well as the movement to a start position. The reason for the reference run is that the machine has to have a defined homing position from which all further movements emanate.

The output of the commands for hydraulic system on/off and piston up-/downward movement is done with the NI 9481 electromechanical relay module. The output of the commands for the voltage of the current regulating valve for the pressure force is done via the NI 9263 $\pm 10V$ analogue output module. In the automatic mode these controls are automatically performed by the software.

Velocity Control Module

The task of the velocity control module is to perform an automated control of the movement velocity. This is done via a PI-controller¹⁶ with pre-control which is explained in detail in chapter 8.2 “Implemented Velocity Feedback Controller”.

The output of the commands for the voltage of the current regulating valve for up/-downward movement is done via the NI 9263 $\pm 10V$ analogue output module. This is automatically performed to reach the defined velocity in automatic and manual mode.

Control System FPGA

The FPGA is the gateway between the NI modules and the Control Modules. Its task is to forward the commands to the output modules. Signal processing is also a part of its work as well as different algorithms which need its high processing speed.

¹⁶ Proportional Integral-controller

6.4 Manual Operations

In this chapter the implemented manual operations and its GUI according to chapter 6.1 “Analysis of the existing software” and to chapter 6.2 “Requirements – PRESTO 420 AUT” will be explained. The main purposes for the manual operations mode are initiation, principle tests and adjustment work as explained in an earlier chapter.

6.4.1 Manual Mode – GUI

In Figure 18: “GUI – Manual Mode” you can see the final GUI of the Software PRESTO 420 AUT. A goal was to separate the GUI into two fields, on the left an operation field and on the right a visualization field. This would guarantee a good overview of the user interface. Because of the limited display size it was not possible to stick to this rule in every point, so compromises had to be found.

The GUI is divided into 6 sections:

- Display of selected sensors
- Display of movement & communication
- Operations of the hydraulic system
- Test parameter definition
- Status LEDs
- Operations for testing

The section “display of selected sensors” shows the live measurement data. It is possible to see cell voltage, module voltage, force and temperature. The possibility to switch to a chart view mode is also given as you can see in Figure 19: “GUI – Manual Mode chart view”.

The section “display of movement & communication” shows the status of the communication of the measurement and control system, the movement velocity and two distances. One distance has its base at the homing position and one at the start position where the way measurement is zeroed.

The section “operations of the hydraulic system” consist of five buttons with different functions. The functions are hydraulic system start/stop, up-/downward movement, emergency stop and fast motion.

The section “test parameter definition” exists to define the parameters for an optional manual test and for optional movement to the reference- or a start position.

The section “Status LEDs” shows the status of the test facility: if a reference- or a start position was reached, if the zeroing was done, if a pressure- or tensile test is selected and if an abort reasons (force, distance, short-circuit) is met.

The section “Operations for testing” consists of a button for the zeroing of the force and way measurement and a button to start the testing.

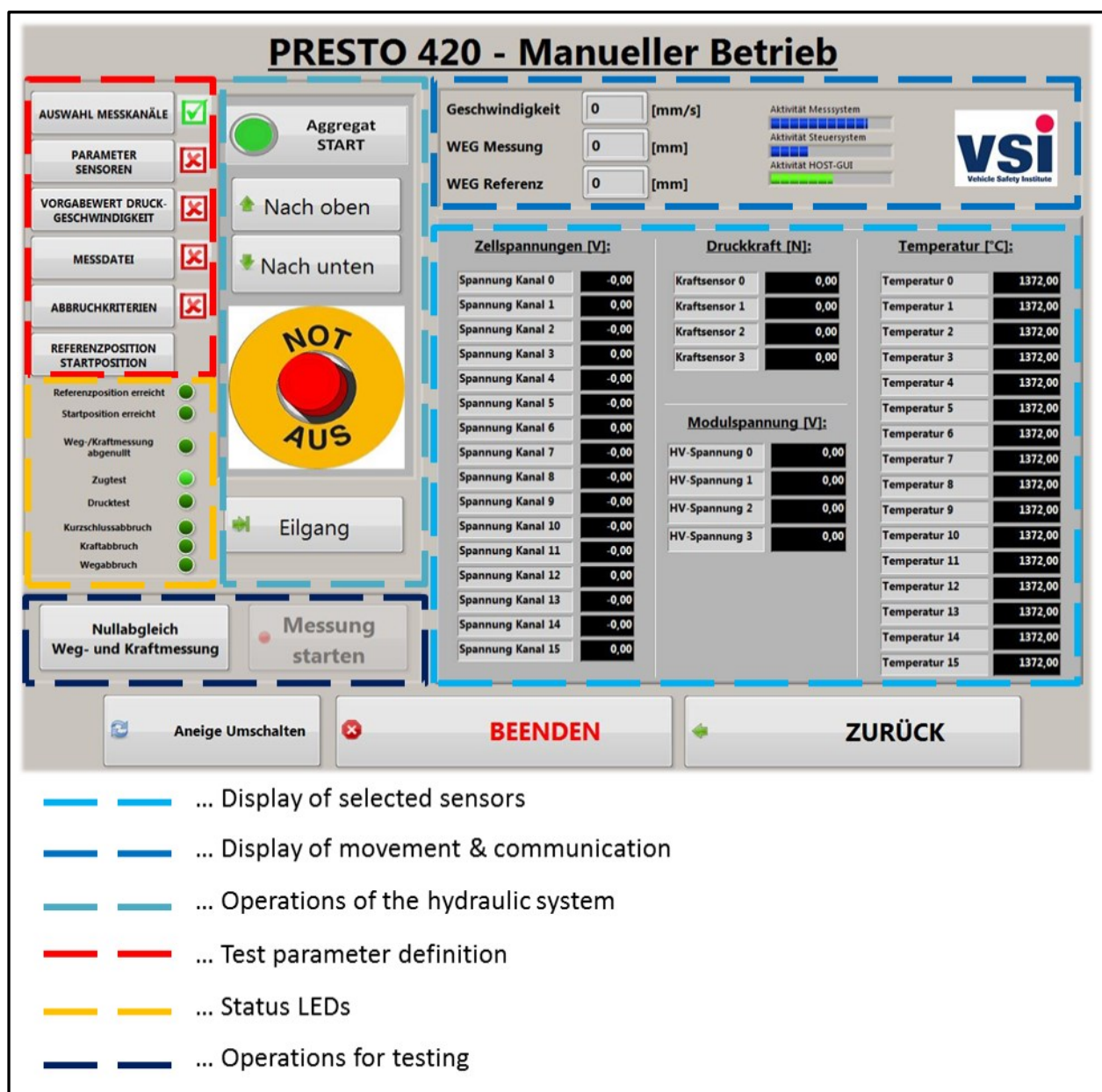


Figure 18: “GUI – Manual Mode”

In Figure 19: “GUI – Manual Mode chart view” you can see the previously mentioned chart view of the measurement data. This mode was implemented because of the explicit wishes of the future test facility operators.

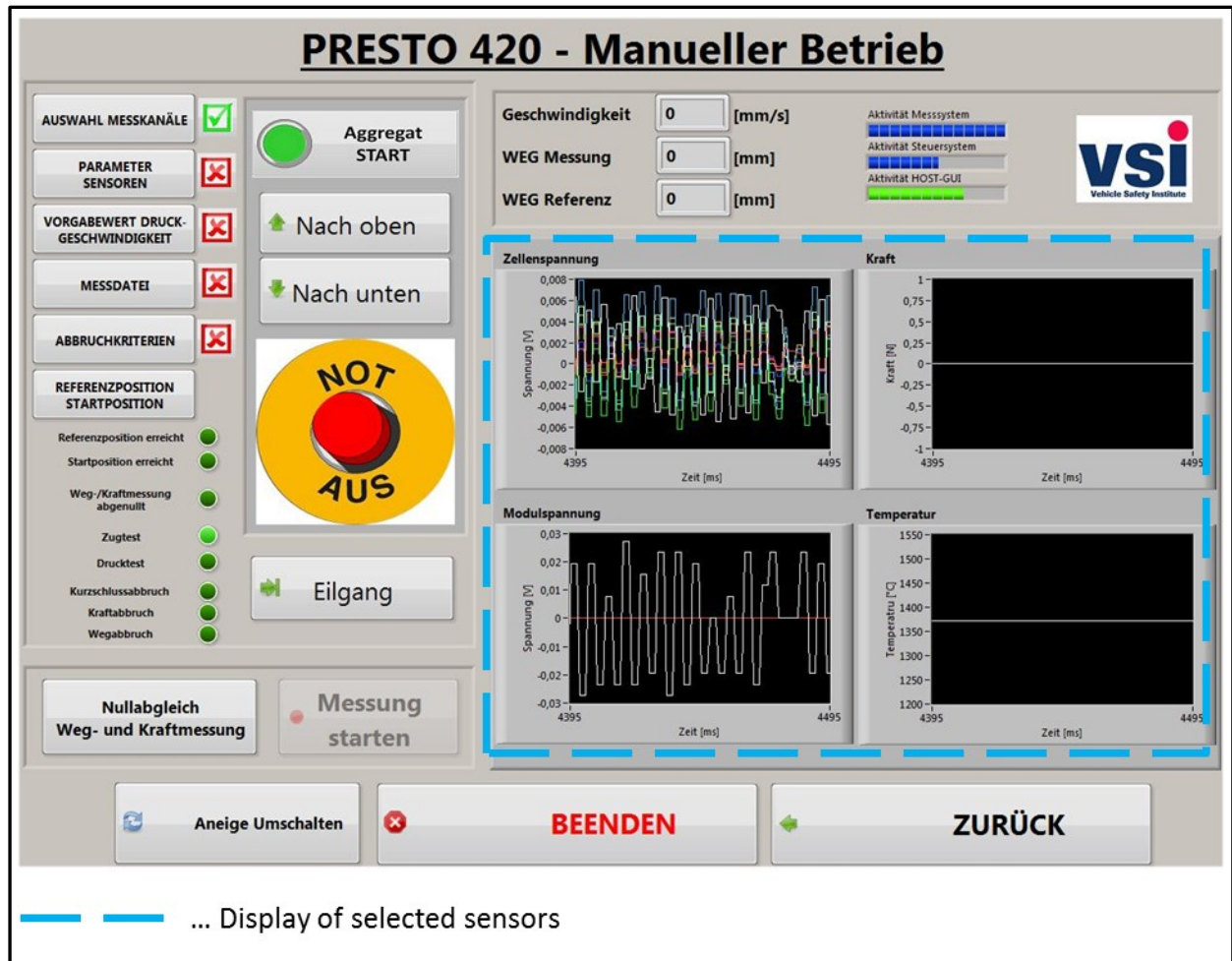


Figure 19: “GUI – Manual Mode chart view”

6.4.2 Process Manual Test

As mentioned before the main purposes of the manual operations mode are initiation, principle tests and adjustment work. So the purpose for testing is also given, you can see the process in Figure 20: “Process Manual Test”.

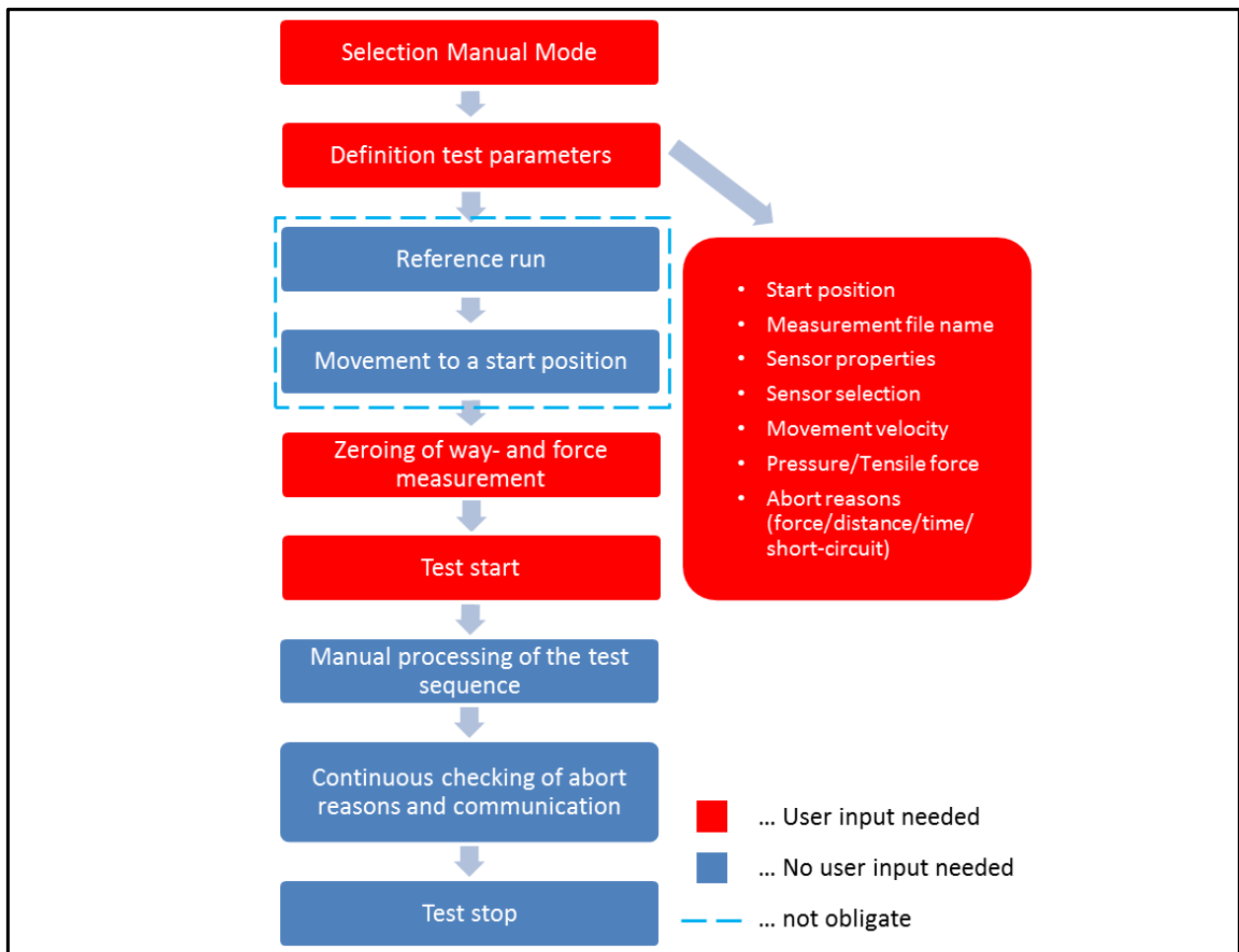


Figure 20: "Process Manual Test"

The first step is to select the manual mode at the start screen. It is only possible to select a mode, manual or automatic, when communication with the two systems is enabled. Afterwards the test parameters have to be defined in the section "test parameter definition". Detailed information about the parameter definition will be given in chapter 6.6 "Preferences for Manual & Automated Operations". It is possible to perform a reference run or an automated movement to a start position in manual mode as well, but it is not obligate. After the definition of all parameters the button for zeroing is enabled. To enable the test start button the way- and force measurement has to be set to zero. This was programmed to avoid failures. In earlier development stages users often have forgotten to push the zeroing button, so the whole test had to be repeated. When the start button was hit and the handshake¹⁷ with the control- and measurement system was performed the manual processing of the test

¹⁷ Control- and measurement system give a feedback when they received a command by the operator

sequence can occur. Meanwhile the system continuously checks the selected abort reasons and the communication. In manual mode the test is not automatically stopped when an abort reason is met. This is not done because while testing new materials (principle tests), sometimes you do not know the right value for an abort reasons, so you want to change it during the test when for example the distance was too short or the maximum force was too low. The test is only automatically stopped when the communication with one of the two systems is broken. The test can always be stopped in a manual way.

6.5 Automated Operations

In this chapter the implemented automated operations with its GUI according to chapter 6.1 “Analysis of the existing software” and to chapter 6.2 “Requirements – PRESTO 420 AUT” are going to be explained. The main purpose of the automated operations mode is to perform repeatable test sequences for more than one time.

6.5.1 Automatic Mode – GUI

Since the automated and the manual operations have a different field of application the GUI is designed in a different way. The automated operations only exist for test reasons. Therefore the section “test parameter definition” was separated from the automated operations GUI, as you can see in Figure 22: “GUI – Automatic Mode”. You can find the separated section “test parameter definition” in Figure 21: “GUI – Initialization Automatic Mode”.



Figure 21: “GUI – Initialization Automatic Mode”

Before you are able to move forward to the automated operations GUI you have to define all needed test parameters. Since the reference and start position run is obligate in the automated mode these parameters have to be defined in front. As you can remember from section 6.4.2 “Process Manual Test” this is not obligate in manual mode.

In automated mode we had the same goal: the separation the GUI into two fields, on the left an operation field and on the right a visualization field. In this case it was realized in a better way as you can see in Figure 22: “GUI – Automatic Mode”.

The GUI is divided into 4 sections:

- Display of selected sensors
- Display of movement & communication
- Status LEDs
- Operations

The section “display of selected sensors” shows the live measurement data. It is possible to see cell voltage, module voltage, force and temperature. The possibility to switch to a chart view mode is also given as you can see in Figure 23: “GUI – Automatic Mode chart view”.

The section “display of movement & communication” shows the status of the communication of the measurement and control system, the movement velocity and two distances. One distance has its base at the homing position and one at the start position where the way measurement is zeroed.

The section “Status LEDs” shows the status of the test facility, if a reference- or a start position was reached and if the zeroing was done. In automatic mode the abort reasons will be shown via pop-ups, not as in manual mode via LEDs. This is done because in automatic mode the test is stopped when an abort reason is reached.

The section “Operations” consists of a button for the zeroing of the force and way measurement, a button to start the testing and a button to switch the display mode.

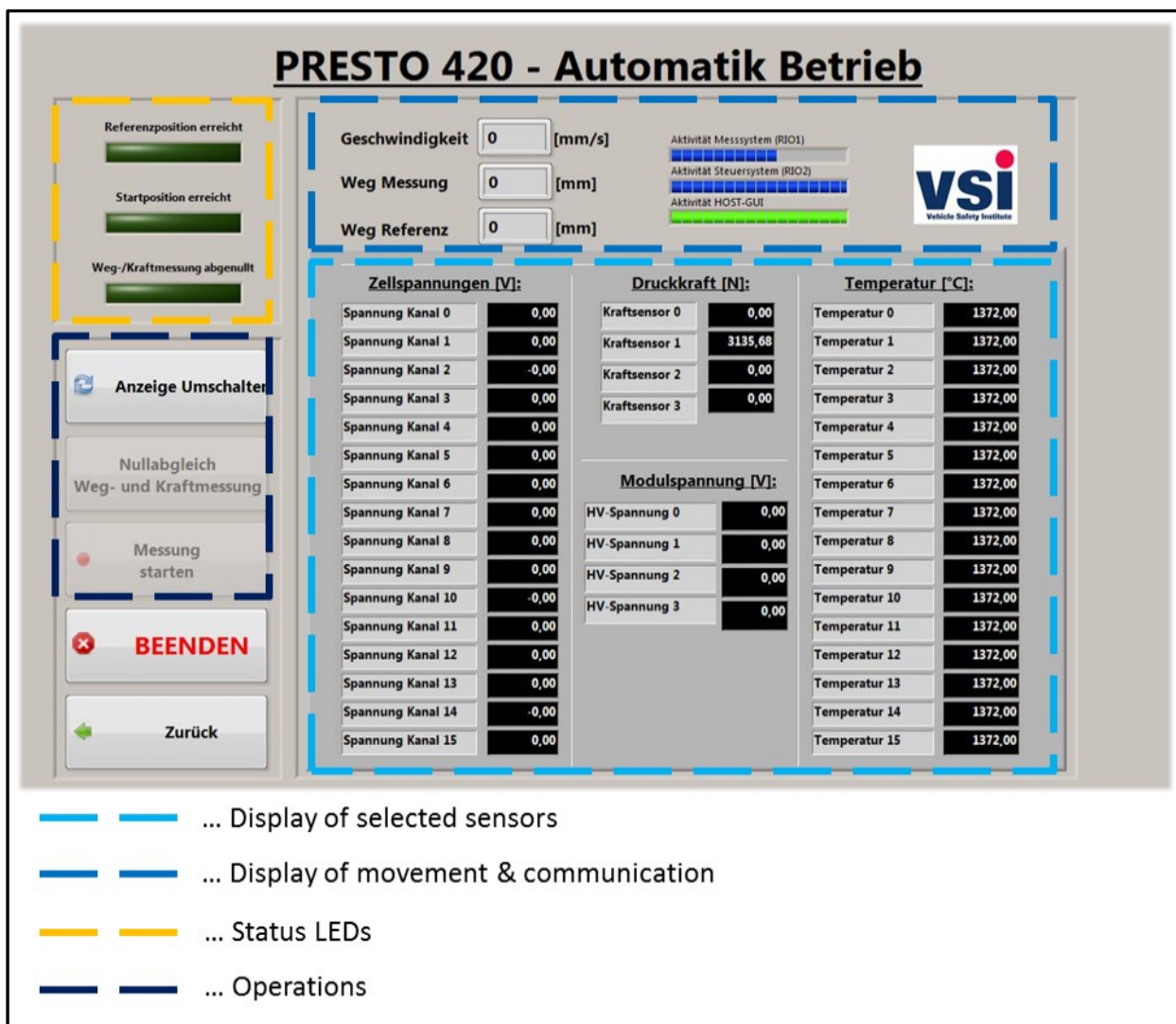


Figure 22: "GUI – Automatic Mode"

In Figure 23: "GUI – Automatic Mode chart view" you can see the chart view of the measurement data which was mentioned before. This mode was implemented for the same reason as in manual mode.

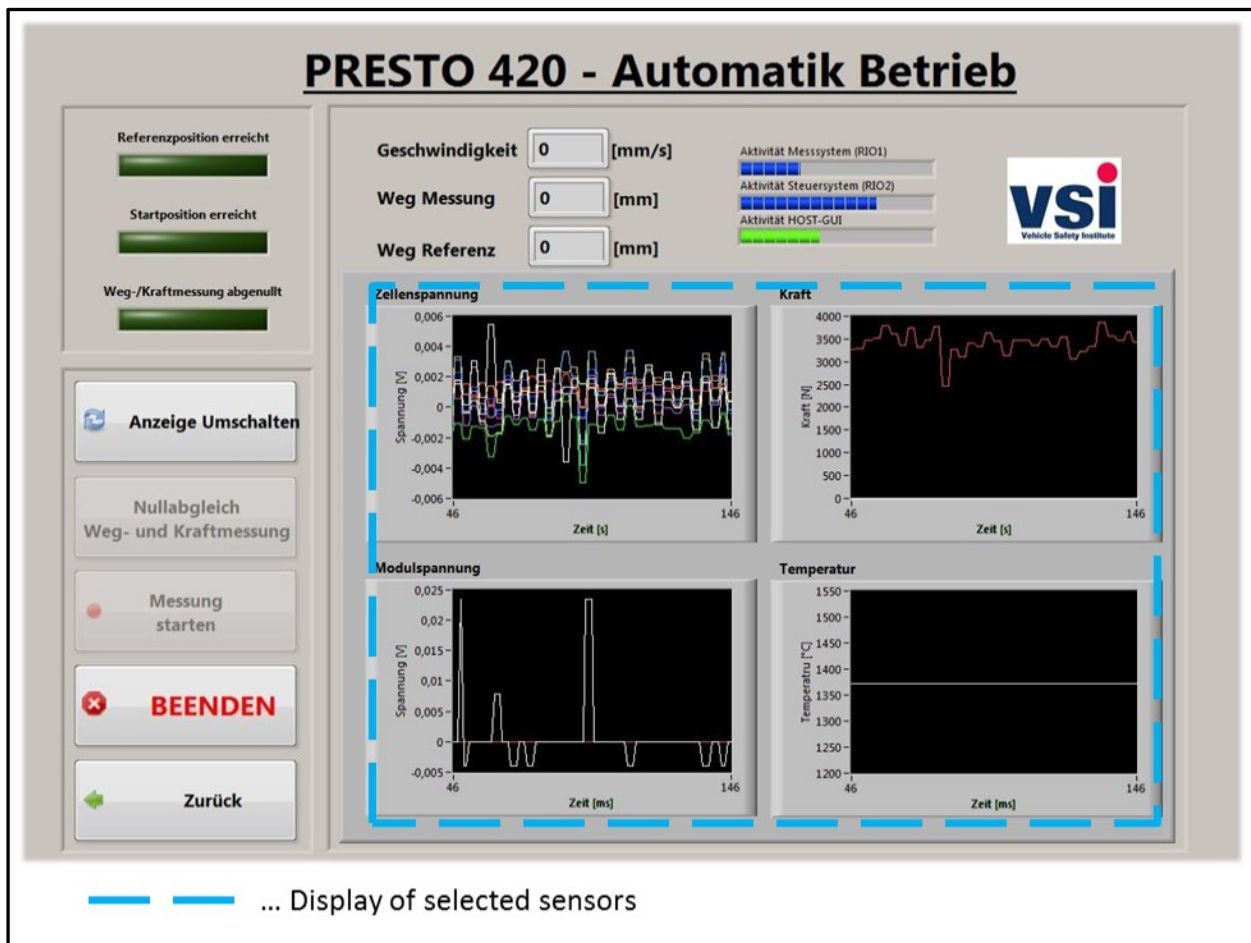


Figure 23: “GUI – Automatic Mode chart view”

6.5.2 Process Automatic Test

As mentioned before the main purpose of the automated operations mode is to perform repeatable test sequences for more than one time. You can see the process for performing a test in Figure 24: “Process Automatic Test”.

The steps till the selection of the usage mode are the same as in the process for a manual test. Afterwards the test parameters have to be defined according to Figure 21: “GUI – Initialization Automatic Mode”. Detailed information of the parameter definition will be given in chapter 6.6 “Preferences for Manual & Automatic Tests”. A reference run and an automated movement to a start position have to be performed in automatic mode before you are able to execute a test. After the definition of all parameters you are able to move forward to the automatic mode GUI. The next step is to perform a zeroing for the way and force measurement. To enable the test start button the way- and force measurement has to be set to zero. When the start button was

hit and the handshake with the control- and measurement system was performed the automated processing of the test sequence will occur. Meanwhile, the system continuously checks the selected abort reasons and the communication. In automatic mode the test is automatically stopped when an abort reason is met. A pop-up will appear either when the test has successfully been performed or when an abort reason has been met. In this pop-up the reason for its appearance is shown. You also have the possibility to automatically move back to the start position with one click.

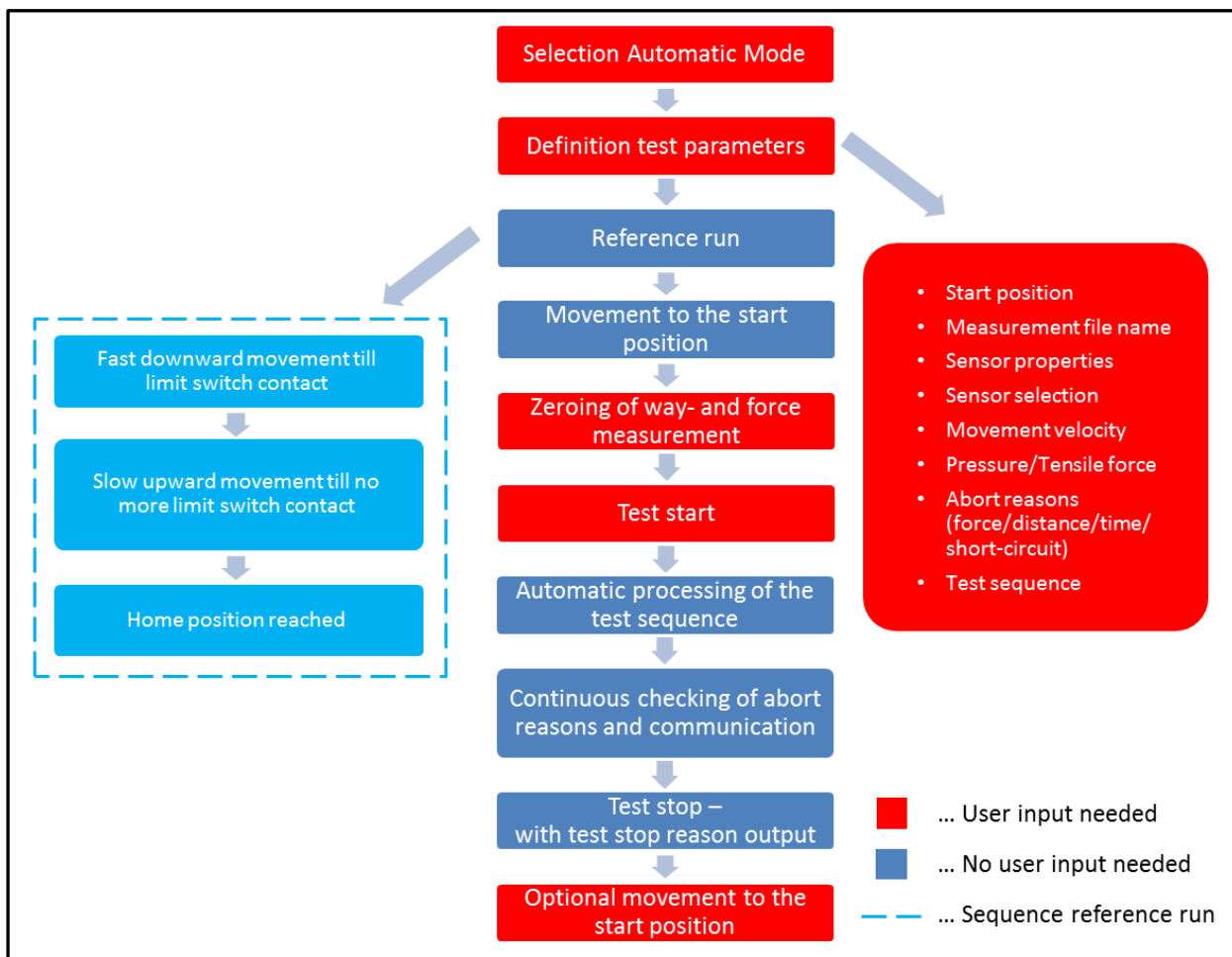


Figure 24: "Process Automatic Test"

6.6 Preferences for Manual & Automatic Tests

Without the specification of several parameters it is not possible to perform tests or even to perform a piston movement. In this section the necessary definitions for test parameters will be explained. As well as the GUIs for the input of these parameters will be shown in the following subchapters.

6.6.1 GUI – Selection Measurement Channels

To only measure the data you want to a selection for measurement channels has to be done. In Figure 25: “GUI – Selection Measurement Channels Manual & Automatic Mode” you can see all the selectable channels. It is also possible to define the sample rate of the measurement channels via input of the AI timing of the time-controlled data acquisition. A maximum sample rate of 2,5kS/s can be performed as you can see in Chart 2: “CPU-Workload as a function of the sample rate” and Chart 3: “Data-writing rate as a function of the sample rate”.

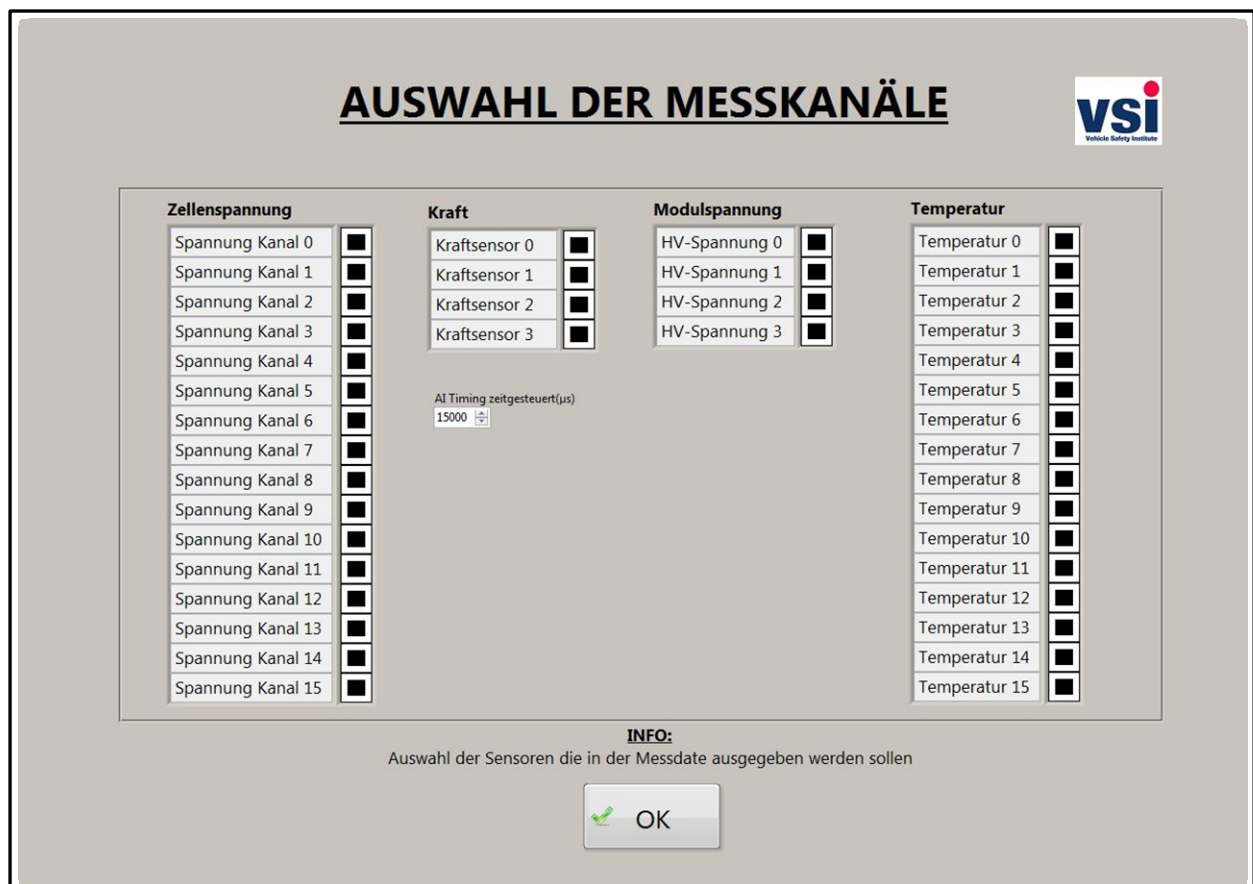


Figure 25: “GUI – Selection Measurement Channels Manual & Automatic Mode”

You can select the following measurement channels according to Table 2: “Measurement System components (National Instruments Corporation, 2011)”


- 16 low voltage ($\pm 10\text{V}$) channels
- 4 high voltage ($\pm 60\text{V}$) channels
- 4 force channels
- 16 temperature channels

Only the selected channels will be shown in the measurement channel display and will be written into the measurement file. Time and way measurement are always performed and written down into the measurement file.

6.6.2 GUI – Parameters Sensors

In Figure 26: “GUI – Parameters Sensors Manual & Automatic Mode” you can see the definition of the parameters for the force and the temperature measurement. The active load cells and their properties have to be defined as well as the type of the thermocouples (K, J, N, R, S, B, T and E).

PARAMETER SENSOREN:



Aktivität Kraftmessdosen

Kraftmessdose 0 AUS/EIN

Kraftmessdose 1 AUS/EIN

Kraftmessdose 2 AUS/EIN

Kraftmessdose 3 AUS/EIN

KMD 0	KMD 1	KMD 2	KMD 3
Seriennummer not specified	Seriennummer N41472	Seriennummer not specified	Seriennummer not specified
Modus Zug	Modus Druck	Modus Druck	Modus Druck
Messbereich [N] 5000	Messbereich [N] 0	Messbereich [N] 5000	Messbereich [N] 50000
Versorgung [V] 5	Versorgung [V] 0	Versorgung [V] 5	Versorgung [V] 5
Kennwert [V/V N] 0,0019997	Kennwert [V/V N] 0	Kennwert [V/V N] 0,0020002	Kennwert [V/V N] 0,0019998
Kraftmessung Funktion 0 Offset Abgleich	Kraftmessung Funktion 1 Offset Abgleich	Kraftmessung Funktion 2 Offset Abgleich	Kraftmessung Funktion 3 Offset Abgleich

ACHTUNG: Aktuell nur **Kraftsensor 1** angeschlossen
Neue Kraftmessdosen sind in Settings_Kraftmessdosen.txt einzutragen

Thermoelementmodul

Thermoelement Typ: K Temperatureinheit: Celsius

Figure 26: “GUI – Parameters Sensors Manual & Automatic Mode”

One big enhancement is the new database for the load cells which was found out as improvement in chapter 6.1 “Analysis of the existing software”. This can be seen in Figure 27: “GUI – Load Cell Database”

A file is stored at the system where all load cells of the Vehicle Safety Institute are written down with its parameters. The serial number of the load cell is used as unique name to select the right cell.

The necessary parameters for load cells are:

- The measurement mode (pressure or tensile)
- The measurement range (from 5000N to 500000N)
- The supply voltage (2,5V or 5V)
- The characteristic value

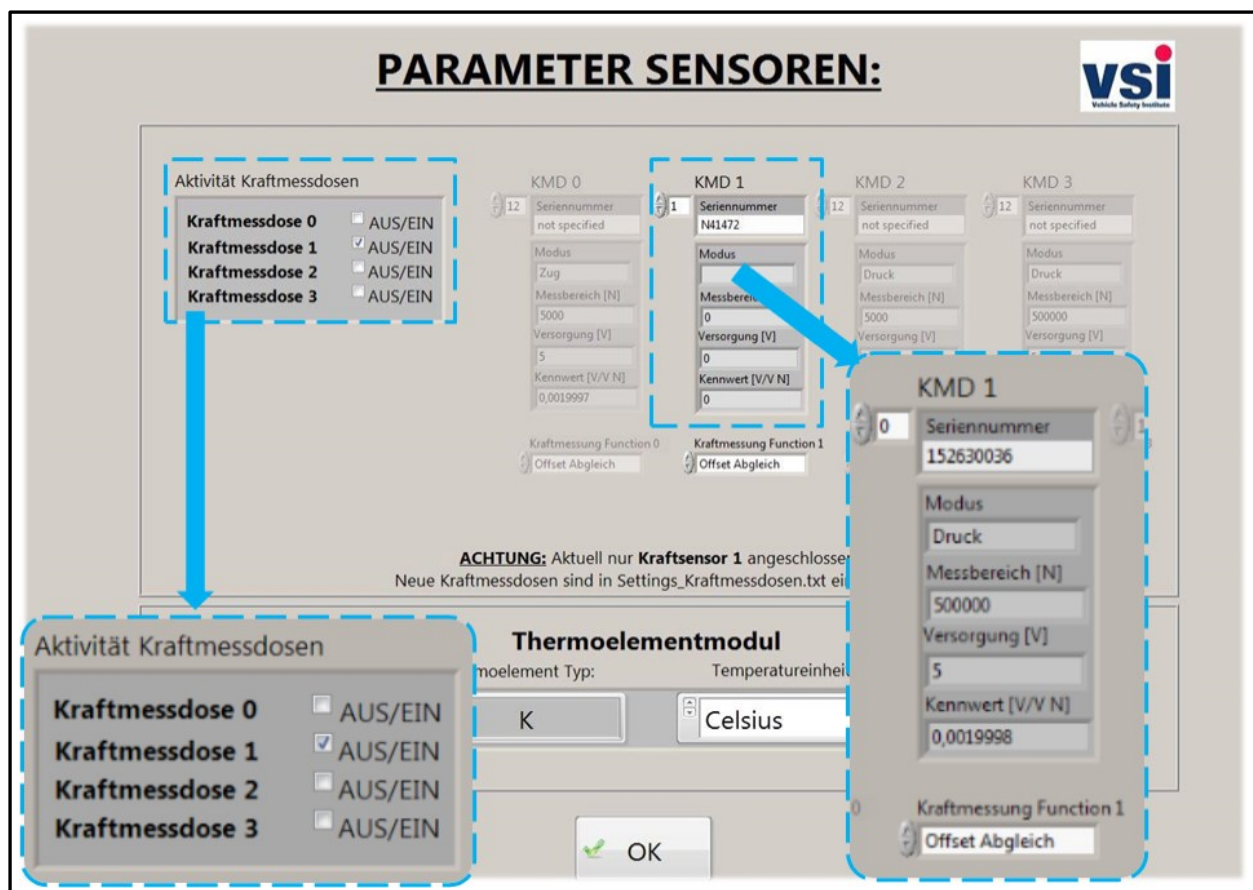


Figure 27: “GUI – Load Cell Database”

In Figure 27: “GUI – Load Cell Database” you can see the user interface for the definition of the sensor parameters. The marked areas are the activation of the different load cells and the

selection of the previously defined loads cells of the database. As you can see only load cell 2 is activated and therefore the others are greyed out.

6.6.3 GUI – Movement Velocity & Pressure Force

In this GUI the specification of the movement velocity and the voltage for the current regulating valve for the pressure and tensile force can be defined.

The machine is capable of all velocities between 0 mm/s and 10 mm/s. To voltage can be entered between 0V and 10 V. 10V equates to 420kN pressure or tensile force.

The velocity control is performed from a PI¹⁸ feedback controller with pre-control described in chapter 7 “Velocity Control”.




Figure 28: “GUI – Movement Velocity & Pressure Force Manual & Automatic Mode”

¹⁸ Proportional-Integral

6.6.4 GUI – Measurement File

To generate a measurement file certain parameters have to be defined according to Figure 29: “GUI – Measurement File Manual & Automatic Mode”:

- Name of the file
- Operator of the machine
- Name of the test

ERSTELLUNG DER MESSDATEI: 

Dateiname
20032013_Versuch1

Anlagenbediener Obermoser
Versuchsbezeichnung Test

INFO:
Keine Umlaute und Sonderzeichen verwenden!




Figure 29: “GUI – Measurement File Manual & Automatic Mode”

6.6.5 GUI – Abort Reasons Manual Mode

In manual mode the definition of the abort reasons is different from the automatic mode because there no test sequence has to be specified. In manual mode you can define an abort reason for the force, for the movement distance and if you want to do perform a short-circuit check. The implementation of this function was a result of the software analysis in chapter 6.1 “Analysis of the existing software”. For performing a short-circuit check it is needed to select the measurement channels you want to monitor. You have also to define the test mode if it is a tensile or pressure test.

The abort reason for the movement distance has to be entered in mm and the one for the maximum force in N. These values should only be defined in certain boarders, the distance

between 0mm and 400mm and the force between 0N and 420000N. If you type in higher values it equates to not perform an abort reason check. Only in automatic mode a check to see if the values are defined in the restricted domain is performed. It is not implemented in manual mode to test all possible cases without a restriction of the input parameters.

VORGABE ABBRUCHKRITERIEN:



<div style="border: 1px solid gray; padding: 5px; display: inline-block;"> Abbruchkriterium WEG <input style="width: 80%; text-align: center;" type="text" value="1E+10"/> [mm] </div>	<div style="border: 1px solid gray; padding: 5px; display: inline-block;"> Abbruchkriterium KRAFT <input style="width: 80%; text-align: center;" type="text" value="1E+10"/> [N] </div>
--	---

Versuchsart	Kurzschlussdetektion
<input type="radio"/> ZUG	<input type="radio"/> NEIN

INFO:
 Vorgabe der gewünschten Abbruchkriterien Maximal-Weg oder Maximal-Kraft und vorgebe ob ein Zug- oder Drucktest durchgeführt wird.

Eingabebereich Abbruchkriterium WEG: 0-400 [mm]
Eingabebereich Abbruchkriterium KRAFT: 0-420000 [N]

Spannungskanäle für Kurzschlussdetektion

Modulspannung 0	<input type="checkbox"/> AUS/EIN
Modulspannung 1	<input type="checkbox"/> AUS/EIN
Modulspannung 2	<input type="checkbox"/> AUS/EIN
Modulspannung 3	<input type="checkbox"/> AUS/EIN
Zellenspannung 0	<input type="checkbox"/> AUS/EIN
Zellenspannung 1	<input type="checkbox"/> AUS/EIN
Zellenspannung 2	<input type="checkbox"/> AUS/EIN
Zellenspannung 3	<input type="checkbox"/> AUS/EIN
Zellenspannung 4	<input type="checkbox"/> AUS/EIN
Zellenspannung 5	<input type="checkbox"/> AUS/EIN
Zellenspannung 6	<input type="checkbox"/> AUS/EIN
Zellenspannung 7	<input type="checkbox"/> AUS/EIN
Zellenspannung 8	<input type="checkbox"/> AUS/EIN
Zellenspannung 9	<input type="checkbox"/> AUS/EIN
Zellenspannung 10	<input type="checkbox"/> AUS/EIN
Zellenspannung 11	<input type="checkbox"/> AUS/EIN
Zellenspannung 12	<input type="checkbox"/> AUS/EIN
Zellenspannung 13	<input type="checkbox"/> AUS/EIN
Zellenspannung 14	<input type="checkbox"/> AUS/EIN
Zellenspannung 15	<input type="checkbox"/> AUS/EIN

Figure 30: "GUI – Abort Reasons Manual Mode"

6.6.6 GUI – Definition Test Sequence Automatic Mode

The definition of the test sequence is a major part of the automation of the PRESTO 420. As listed in chapter 6.2 “

Requirements - PRESTO 420 AUT” the performing of different test stages is needed. With the implemented automatic mode it is possible to define any sequence out of three stages with its own abort reasons:

- Tensile test – abort reason for movement distance [mm]
- Pressure test – abort reason for movement distance [mm]
- Waiting phase – abort reason for waiting time [s]

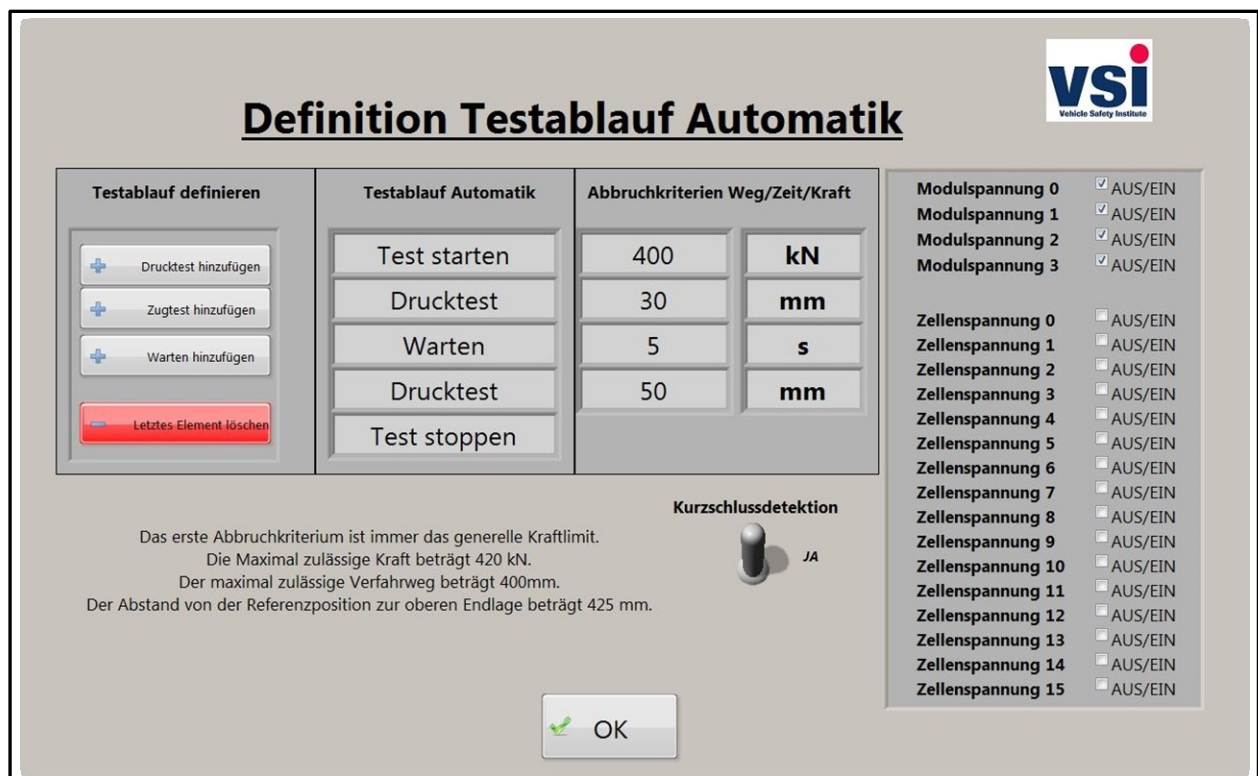


Figure 31: “GUI – Definition Test Sequence Automatic”

The definition of the test sequence can be seen in Figure 31: “GUI – Definition Test Sequence Automatic”. In this figure an example of a test sequence is given:

When you enter the GUI for the defining of the test sequence the first thing which has to be done is the input if the abort reason for the maximum force. This happens via a pop-up which can be seen in Figure 32: “POP - UP maximum force Input”. “Test start” is always performed because it is the initialization of a following test. The first phase is a pressure test with a movement distance of 30mm followed by a waiting phase of 5s. Then a second pressure test is performed with a displacement of 50mm. This is followed by the obligate “Test stop” phase which stops all the movement and turns off the hydraulic system. All entered abort reasons are

checked for its correctness. If a value is not correct you have to enter it again and again until it is correct.

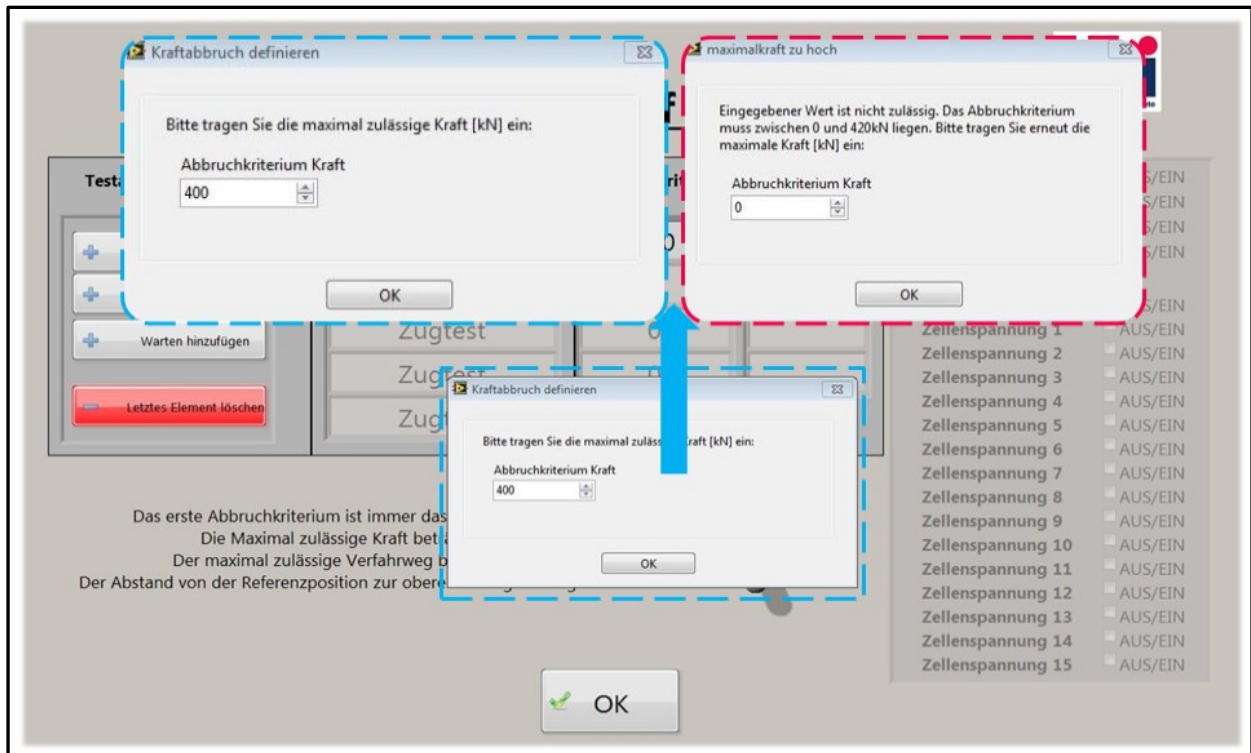


Figure 32: “POP - UP maximum force Input”

In Figure 32: “POP - UP maximum force Input” you can see the previously mentioned popup. Here the user made a wrong input. The value was higher than allowed (0kN till 420kN), so a second pop-up appeared which informs the user about his mistake. Now the user has to type in the value again until the input is correct. If an input was correct, the user can go on with the defining of the test sequence.

6.6.7 GUI – Reference Positions

The GUI for the reference positions is different in the manual and the automatic mode. The reason for that will be described in the following subchapters.

Manual Mode

In Figure 33: “GUI – Reference Positions Manual Mode” the GUI for the reference positions in manual mode can be seen. In manual mode neither a reference run nor a start position run is

needed to perform a test. Regardless the possibility to perform these runs is given in manual mode. For performing a reference run only a the button has to be hit, for performing a start position run the desired start position has to be entered. The position is the distance from the upper to the lower backing plate. The reference position is defined by the end switch.



Figure 33: "GUI – Reference Positions Manual Mode"

6.6.7.1 Automatic Mode

In automatic mode a reference run is obligate to have a base for further operations. The definition of the start position is the same as in manual mode.

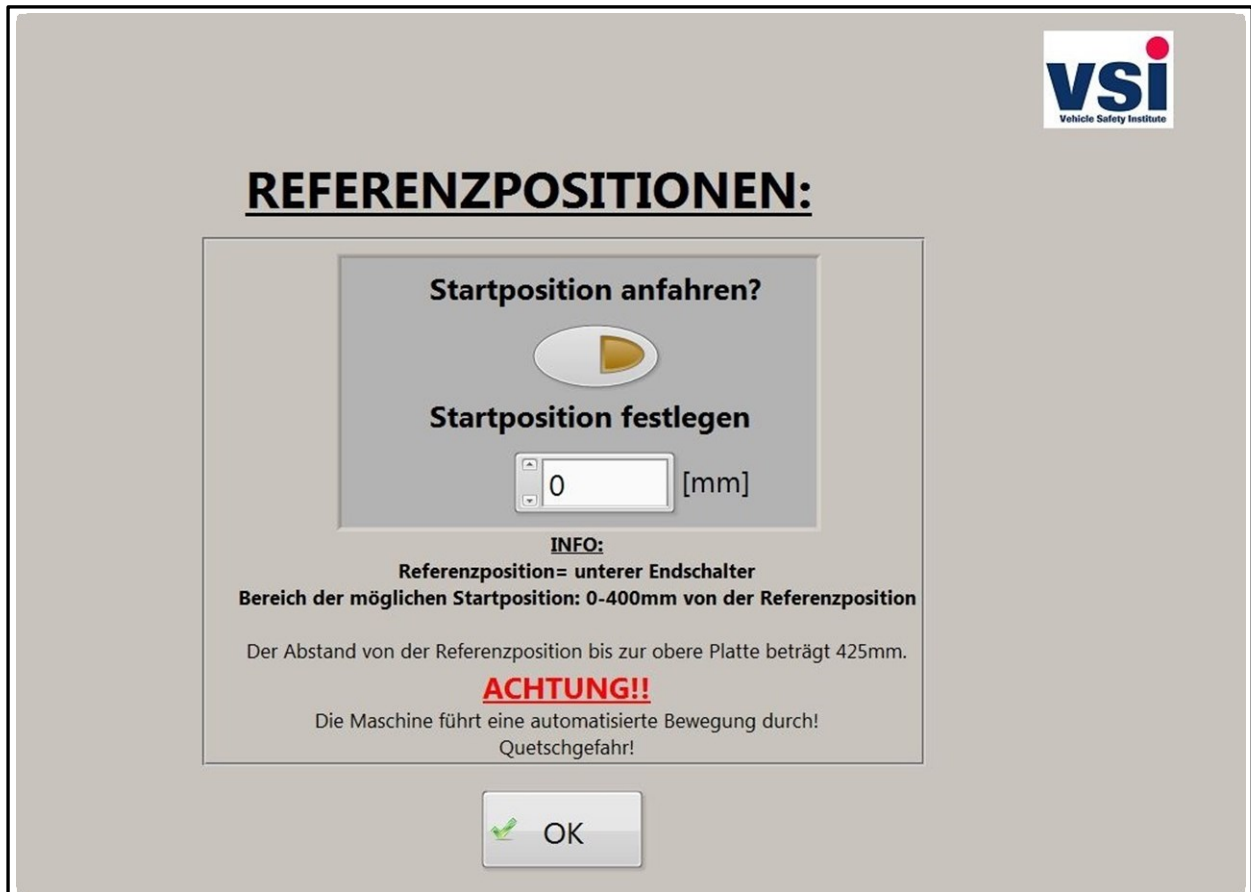


Figure 34: "GUI – Reference Positions Automatic Mode"

7 Tensile Test

One of the major goals of this thesis was to enable a tensile test mode for the PRESTO 420. To fulfil this task hardware modifications had to be done as well as the software implementation. This additional feature opens a whole new field of testing. For different materials tests of the behaviour under tensile stress is needed as well as under pressure stress. This is also a competitive advantage.

7.1 Hardware Modifications

As explained in chapter 3 “Initial situation” the PRESTO 420 was only capable of pressure tests. For that reason only one current regulating valve was built in to regulate the oil flow. This valve is regulating the flow in one direction only and it is open in the other direction. This is a so called throttle-check valve with choke. You can see the principle of the hydraulic system in Figure 35: “Hydraulic system scheme old – PRESTO 420”

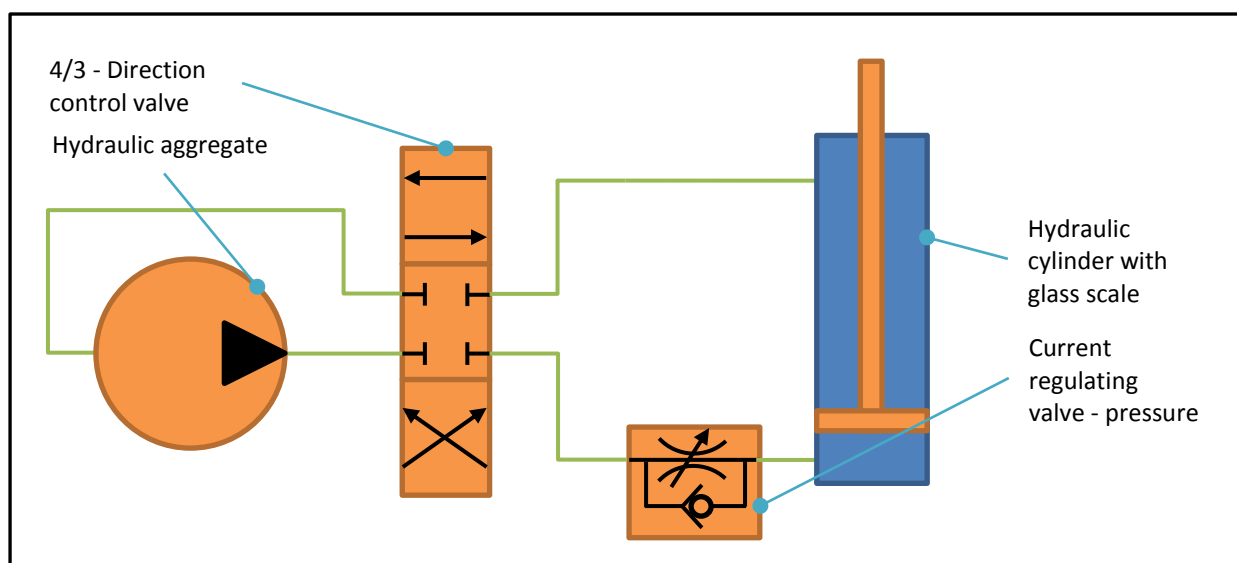


Figure 35: “Hydraulic system scheme old – PRESTO 420”

To enable a tensile mode a second current regulating valve had to be built in with additional force pipes. The new valve is regulating the oil flow in tensile direction. This system setup can be seen in Figure 36: “Hydraulic system scheme new – PRESTO 420”. The pressure of the system is generated by a hydraulic aggregate which allows a maximum pressure force of 420N. To activate an up- or downward movement an 4/3 – direction control valve is addressed. The

movement speed of the two way hydraulic cylinder is controlled via two current regulating valves. In comparison with the old system there was only one current regulating valve for the upward movement, so no control of the downward movement speed was possible.

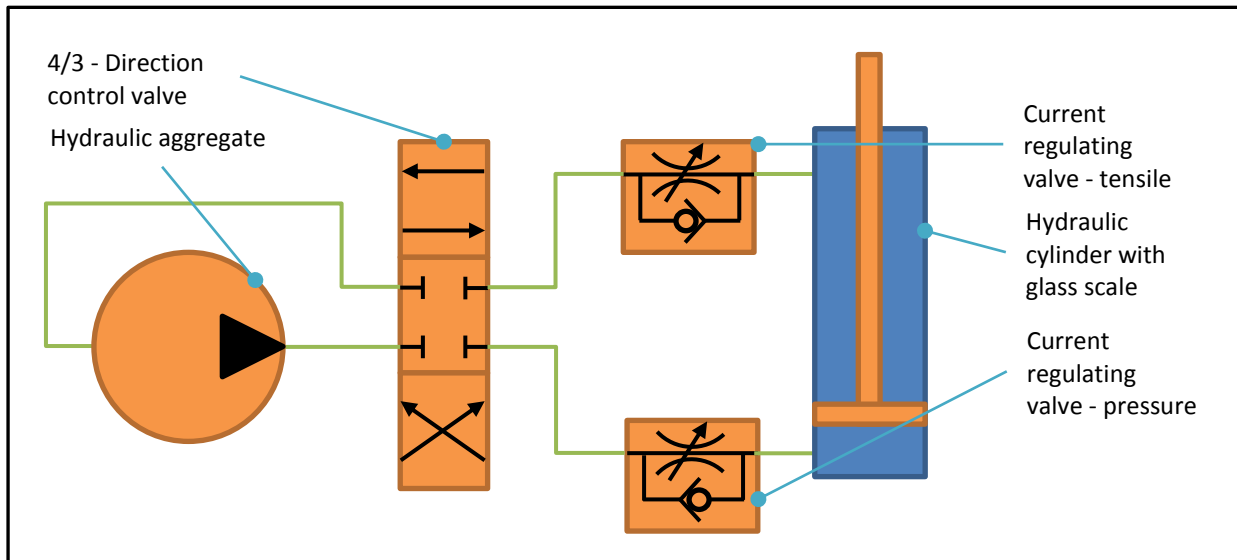


Figure 36: "Hydraulic system scheme new – PRESTO 420"

7.2 Software Implementation

In principle the implementation of the tensile mode is simple. Firstly the control of the valve had to be programmed at the FPGA. Then the implementation in the different modes had to be done:

- Only when a movement is performed in manual mode the selection of the pressure and tensile valve occurs automatically.
- In manual test mode the selection of pressure and tensile mode can be done via a dip-switch. This has to be done manually because of the abort reasons. They react in a different way on pressure and tensile mode.
- In automatic mode the selection of pressure and tensile mode is performed automatically depending on the defined test sequence. The user shall not be allowed to change the selection manually.

8 Velocity Control

According to (Lunze, 2013) feedback control has the task to influence a time dependent process from outside to follow a certain guideline. Velocity control is a classic example for a feedback control task and allows the PRESTO 420 to fulfil tests under constant conditions.

In this chapter the feedback control system of the developed test facility will be explained as well as the theoretical background.

8.1 Feedback Control System basics

This chapter is based on (Lunze, 2013), (Böge, 2013), (Czichos, et al., 2012) and (Grote, et al., 2012).

8.1.1 Scope of work of Feedback Control Systems

Feedback control has various areas of application for example the control of room temperature which is influenced by solar radiation, or the movement of a robotics gripper along the edge of a work piece.

In all this cases a measurement, which is influence able from outside, has to be selected to fulfil the specified goals. The selection occurs in dependence of how much of the goal is already fulfilled. Because of that the feedback control circle begins.

Feedback control deals with the control of dynamic systems. This definition refers to the two phrases “dynamic system” and “control” which should be explained.

A **dynamic system** exists when the most important parameters are changed with the time and therefore are able to be published as time dependent graphs. It is distinguished between input parameters, which influence the system internally, and output parameters, which describe the reaction of the system on the input parameters. For example when driving a car, the input parameters are the angularity of the gas-/brake pedal and the steering wheel, the output parameter is the velocity with its value and direction.

The changes in a dynamic system are called **dynamic process**. In an air condition the heating of the air in the room is the process. Because of that, a dynamic system can be seen as part of a

world where dynamic processes occur. For feedback control tasks the difference between system and process is negligible, they can be used as synonym for each other.

As **control** the goal oriented influencing of a dynamic system is designated. In terms of the scope of view there can be distinguished between more layers of control as you can see in Figure 37: "Structural design of a multilayer control (Lunze, 2013)". In the bottom level the input parameter u_i is calculated with the feedback controller i in dependence of the output parameter y_i . The controller and the controlled system are under constant interdependencies and are forming a controlling loop. This is called feedback control system. The higher layers are performing more long-term control tasks than the feedback control system. The optimization tries to identify the right parameter (command variable) w_i for the feedback control like development of its working points.

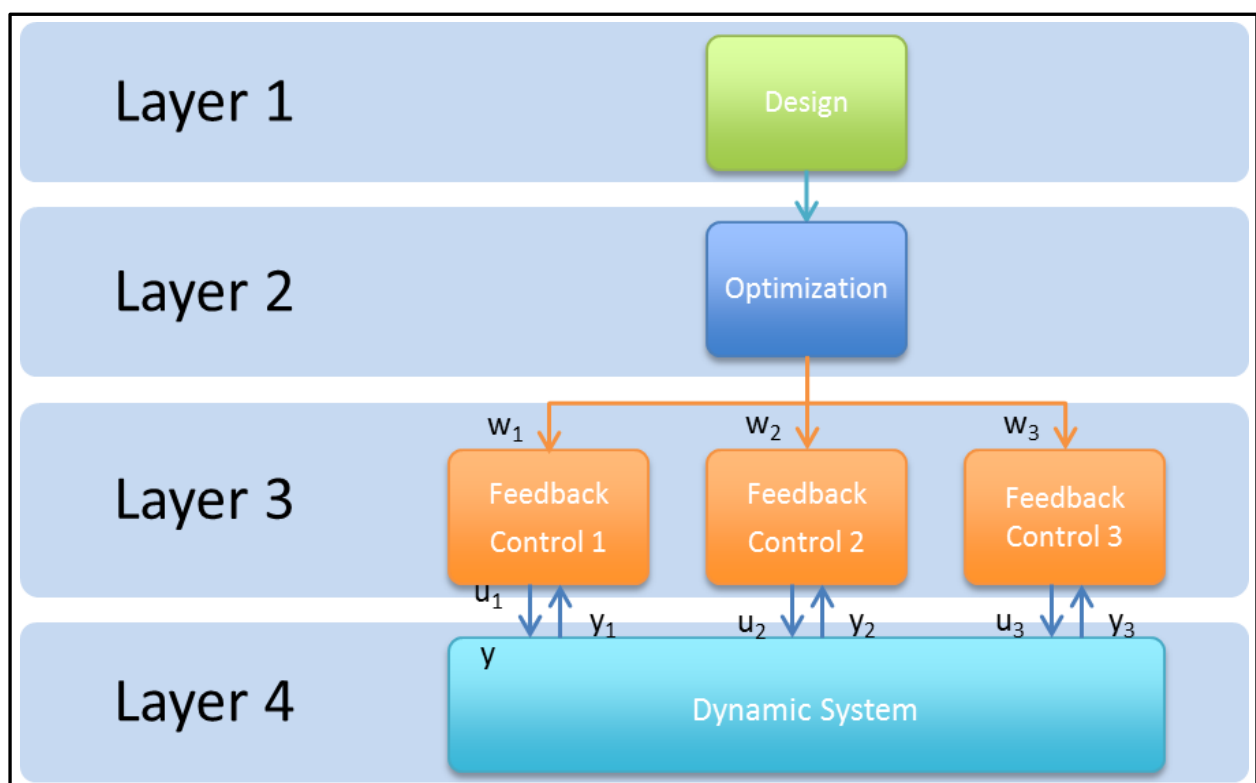


Figure 37: "Structural design of a multilayer control (Lunze, 2013)"

The previous explanations made clear that feedback control deals with the following scope of work:

- **Given** is a dynamic system (control path) with a parameter which is influenced from outside (input parameter or actuating variable) and a measurable parameter (output

parameter or control variable). As well the control goal is given which typical task is to hold the control variable on a certain value or to change the control variable in time in a way which is predetermined by the command variable

- **Sought** is a feedback control which is able to reach the goals under usage of the given values

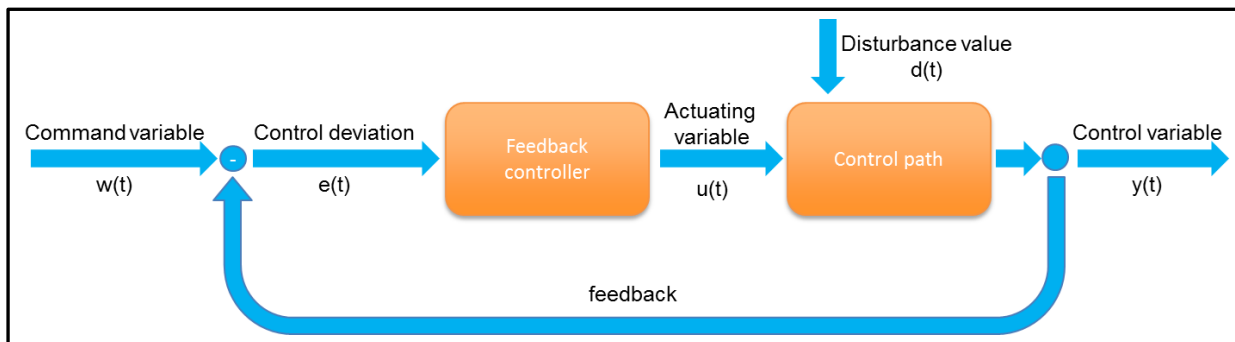


Figure 38: “Principle of a feedback control (Lunze, 2013)”

Figure 38: “Principle of a feedback control (Lunze, 2013)” explains in principle how a feedback control works. The control variable $y(t)$ depends on the actuating variable $u(t)$ and on the disturbance value $d(t)$ which is not influence able. The goal is to guide the control variable $y(t)$ to the same value as the command variable $w(t)$, in an ideal case $y(t) = w(t)$ for all points in time. So the feedback controller has to calculate the actuating variable $u(t)$ to compensates the disturbance value $d(t)$. As additional information to the given value $w(t)$ the feedback controller has the actual value $y(t)$. As difference

$$e(t) = w(t) - y(t)$$

Equation 1: “Control deviation $e(t)$ ”

of both measurements the feedback control is able to determine the control deviation. So it uses as well the actual value $y(t)$ for its calculation of the actuating variable.

8.1.2 Types of Feedback Controllers

According to (Böge, 2013) feedback controllers can be divided into three groups, continuous/discontinuous/quasi-continuous, as you can see in Figure 39: “Feedback controller types (Böge, 2013)”. In this thesis only the feedback controllers, which are relevant for our system, are going to be explained.

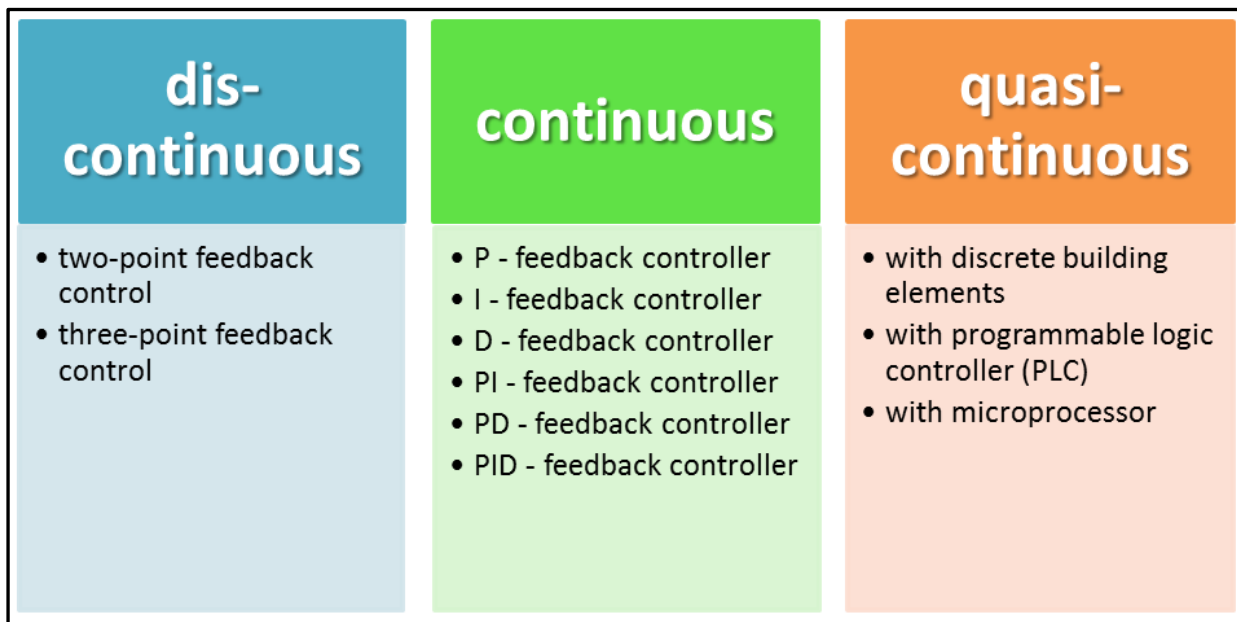


Figure 39: "Feedback controller types (Böge, 2013)"

Proportional (P) - Feedback Controller

The proportional feedback controller is in terms of maths the easiest way to realize a feedback control. The actuating variable $u(t)$ is proportional to the control deviation $e(t)$. Its feedback controller law is:

$$u(t) = k_p * e(t)$$

Equation 2: "Proportional feedback controller law (Böge, 2013)"

Its **transfer coefficient** is the proportional coefficient k_p and can be calculated with the rise of the P – feedback controllers characteristic curve which you can see in Chart 4: "Characteristic curve of a proportional feedback controller (Böge, 2013)"

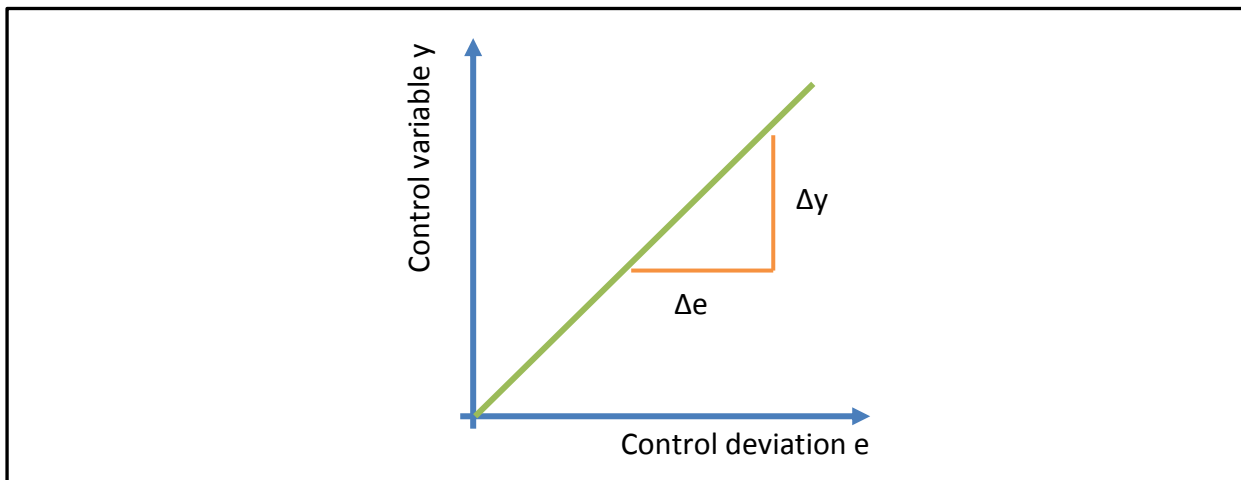


Chart 4: “Characteristic curve of a proportional feedback controller (Böge, 2013)”

The equation of k_p is

$$k_p = \frac{\Delta y}{\Delta e}$$

Equation 3: “Proportional coefficient (Böge, 2013)”

Its **step function response** can be easily shown because of its simple mathematical context. It is illustrated in Chart 5: “Step function & response of a proportional feedback controller (Böge, 2013)”

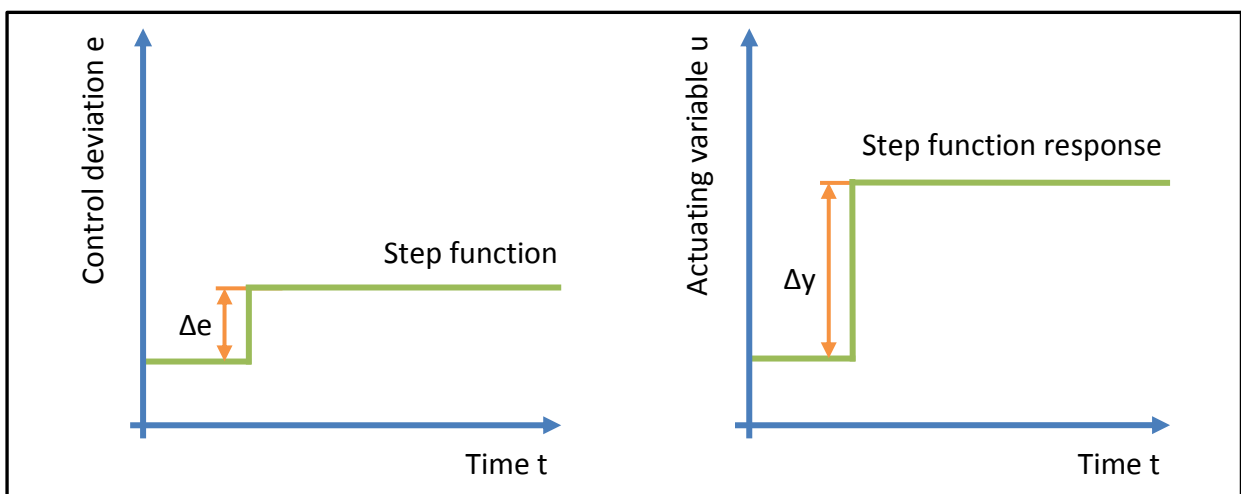


Chart 5: “Step function & response of a proportional feedback controller (Böge, 2013)”

Integral (I) - Feedback Controller

The integral feedback controller is in terms of math more complicated. The actuating variable $u(t)$ is proportional to the area which the control deviation $e(t)$ forms in a specified period of time. This area is represented as $\int e dt$. Its feedback controller law is:

$$u(t) = k_i * \int e dt$$

Equation 4: "Integral feedback controller law (Böge, 2013)"

For constant control deviations its equation can be reduced to:

$$u(t) = k_i * e * t$$

Equation 5: "Integral feedback controller law for constant control deviations (Böge, 2013)"

Its **transfer coefficient k** is $k_i * t$. k_i can be calculated as reciprocal of the integration time T_i . k grows for all $t \rightarrow \infty$ over all borders. K_i is defined as the integral action coefficient with its equation:

$$k_i = \frac{1}{T_i}$$

Equation 6: "Proportional coefficient (Grote, et al., 2012)"

Its **step function response** is illustrated in Chart 6: "Step function & response of an integral feedback controller (Böge, 2013)".

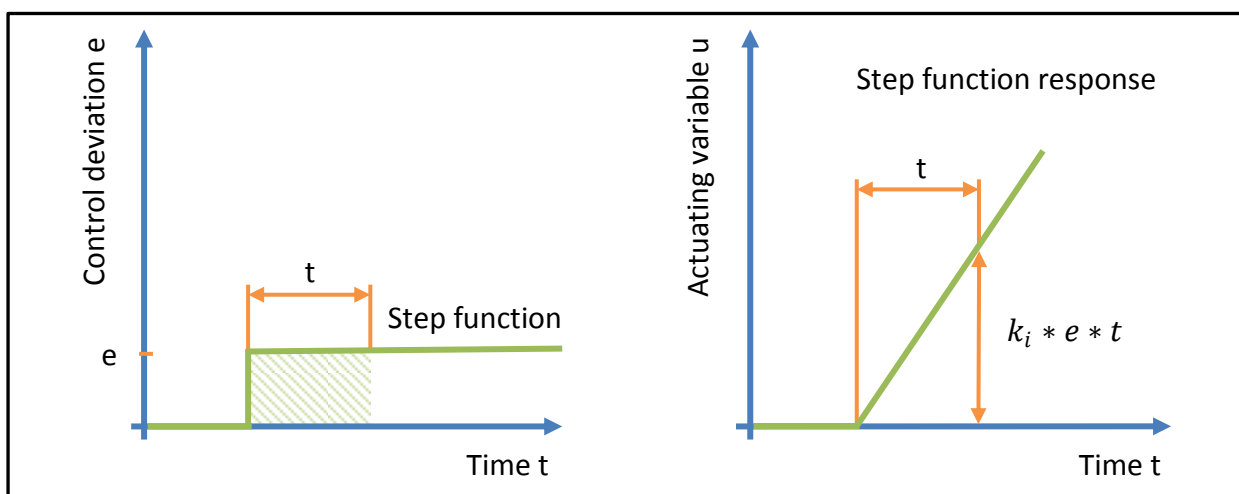


Chart 6: "Step function & response of an integral feedback controller (Böge, 2013)"

Proportional Integral (PI) - Feedback Controller

In principle the proportional integral – feedback controller combines the behaviour of an integral and a proportional controller which was explained before.

Its feedback controller law is:

$$u(t) = k_p * e(t) + k_i * \int e dt$$

Equation 7: “Proportional integral feedback controller law (Grote, et al., 2012)”

With introducing of the **reset time** T_N a new equation can be formed:

$$u(t) = k_p * \left[e(t) + \frac{1}{T_N} * \int e dt \right]$$

Equation 8: “Proportional integral feedback controller law with reset time (Grote, et al., 2012)”

The **reset time** is the second characteristic value of a PI feedback controller. It is defined as the time which is needed by the I-part to fulfil the same change of the actuating variable $u(t)$ as the P-part does immediately.

$$T_N = \frac{k_p}{k_i}$$

Equation 9: “Reset time proportional integral feedback controller”

Its **step function response** is illustrated in Chart 7: “Step function & response of a proportional integral feedback controller”.

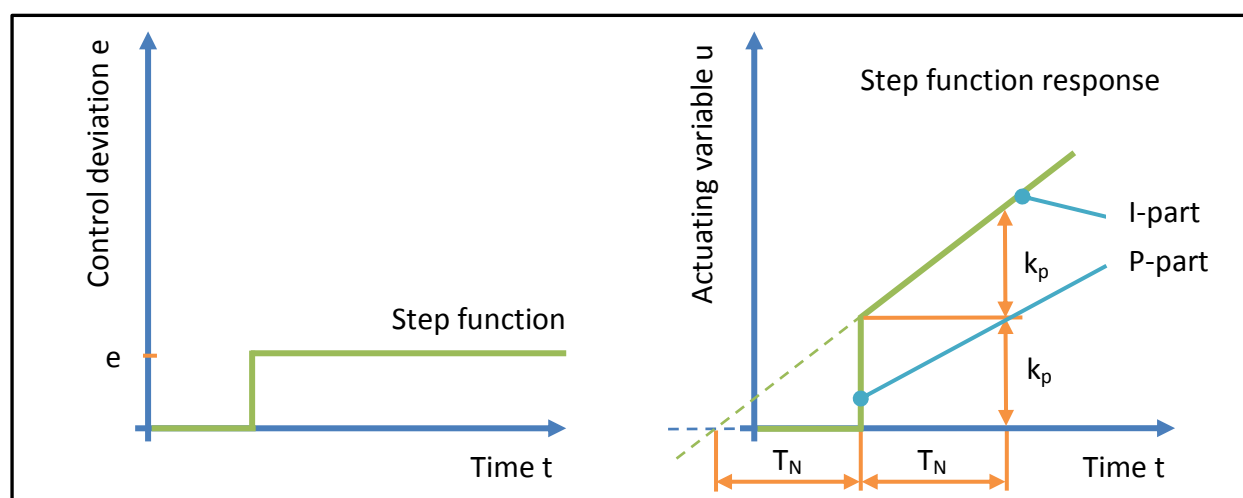


Chart 7: “Step function & response of a proportional integral feedback controller”

8.2 Implemented Velocity Feedback Controller

As already mentioned in previous chapters a velocity feedback control is crucial for the reproducibility of test setups.

8.2.1 Velocity measurement

Because no explicit speed measuring takes place, the velocity is calculated via feedback of the glass scale. The calculation of the movement velocity is done with the following equation:

$$v = \frac{\sum_{i=0}^9 \frac{s_{n+1} - s_n}{t_{n+1} - t_n}}{10}$$

Equation 10: "Movement velocity"

As you can figure out with Equation 10: "Movement velocity" the actual velocity is the average of the last 10 values. s_n is the value of the encoder position measured by the glass scale at the time t_n . With the difference $s_{n+1} - s_n$ we get as a result the movement distance in the time period $t_{n+1} - t_n$. To calculate the velocity the way has to be divided by the time: $v(t) = \frac{\Delta s}{\Delta t} = \frac{s_{n+1} - s_n}{t_{n+1} - t_n}$. Because of the fast response of the CompactRIO systems we calculate the average as shown in Equation 10: "Movement velocity". Since the calculation of the average has always a deviation to the actual speed a failure estimation was done which you can find in Table 6: "Failure estimation average velocity". The data for the chart was recorded during a pressure test of a crush tube.

Position [μm]	Time [ms]	Velocity [mm/s]	Desired Velocity [mm/s]	Average Velocity [mm/s]	Deviation Desired Velocity [%]	Deviation Average Velocity [%]
1098	615	1,07	1	1,13	6,67	-5,88
1114	630	1,13	1		13,33	0,00
1131	645	1,13	1		13,33	0,00
1148	660	1,13	1		13,33	0,00
1165	675	1,07	1		6,67	-5,88
1181	690	1,13	1		13,33	0,00

1198	705	1,13	1		13,33	0,00
1215	720	1,20	1		20,00	5,88
1233	735	1,13	1		13,33	0,00
1250	750	1,20	1		20,00	5,88
Mean deviation 1					13,33	
1268	765	1,13	1	1,11	13,33	1,80
1285	780	1,13	1		13,33	1,80
1302	795	1,13	1		13,33	1,80
1319	810	1,07	1		6,67	-4,19
1335	825	1,13	1		13,33	1,80
1352	840	1,20	1		20,00	7,78
1370	855	1,07	1		6,67	-4,19
1386	870	1,13	1		13,33	1,80
1403	885	1,07	1		6,67	-4,19
1419	900	1,07	1		6,67	-4,19
Mean deviation 2					11,33	
1435	915	1,20	1	1,02	20,00	17,65
1453	930	1,07	1		6,67	4,58
1469	945	1,07	1		6,67	4,58
1485	960	1,07	1		6,67	4,58
1501	975	0,93	1		-6,67	-8,50
1515	990	1,07	1		6,67	4,58
1531	1005	1,00	1		0,00	-1,96
1546	1020	0,80	1		-20,00	-21,57
1558	1035	0,93	1		-6,67	-8,50
Mean deviation 3					2	
Mean deviation					8,89	

Table 6: "Failure estimation average velocity"

With the measured position and time values two different mean deviations were calculated. The deviation to the desired velocity equates to the failure of the feedback controller. The deviation to the average velocity equates to the failure of the average velocity calculation.

8.2.2 Controlled hydraulic system

To regulate the movement speed we have to control the flow in our hydraulic system. This is done via controlling the voltage of two current regulating valves where only one flow direction is throttled. One valve controls the flow in pressure test mode and one in tensile test mode. The change of the movement direction is done via a 4/3-direction control valve with block middle position. The pressure of the system is gained by a hydraulic aggregate. You can see the principle of the described system in Figure 36: “Hydraulic system scheme new – PRESTO 420”

8.2.3 PI - velocity feedback control

To develop a feedback controller the first thing which has to be done is to define the variables. In our case the command variable $w(t)$ is represented by the given velocity, the actuating variable $u(t)$ is represented by the voltage of the current regulating valves, the control variable $y(t)$ is represented by the present velocity and the disturbance variable $d(t)$ is represented by the system inertia (inertia of the valves, etc.). The control deviation $e(t)$ is calculated with the difference $w(t)-y(t)$. After the definition of the variables a feedback controller has to be found which fits to the system.

The first implemented system controller was a PI-feedback controller, which can be seen in Figure 40: “PI-feedback velocity control– PRESTO 420”. A PI controller is used due to the fact that it is capable of eliminating a control deviation if a constant command variable is given. For our reasons this is necessary since the reaching of the desired test speed is crucial. Because of the high system inertia there was no use to implement a D-part in our feedback controller. The job of a D-part is to react on fast system changes which do not occur in our system. Since we were not happy with the results a new solution had to be found.

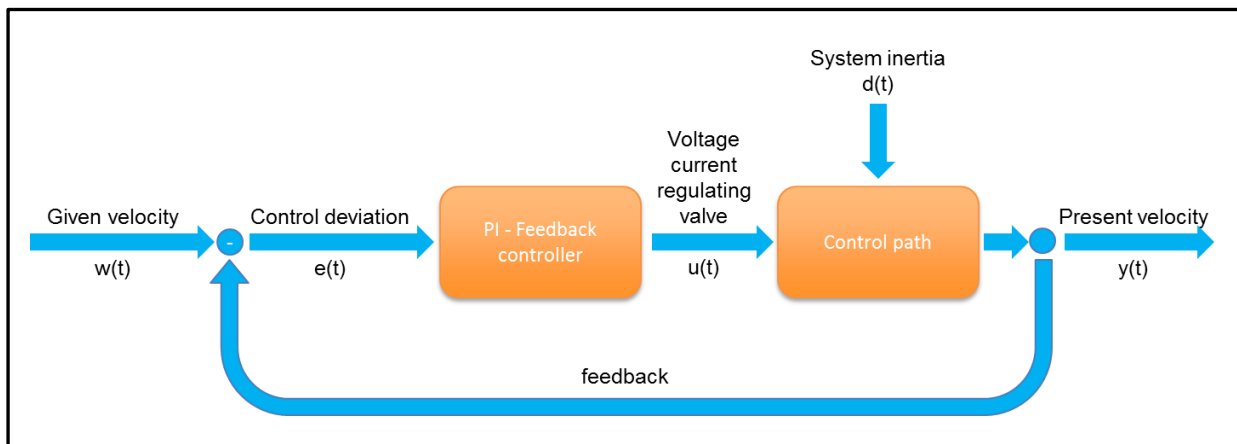


Figure 40: "PI-feedback velocity control– PRESTO 420"

The behaviour of the system was too slow, it can be seen in Chart 8: "System behaviour with PI-feedback control – valve open at the beginning" and Chart 9: "System behaviour with PI-feedback control – valve closed in the beginning". These charts are only valid for movement without a pressure or tensile test.

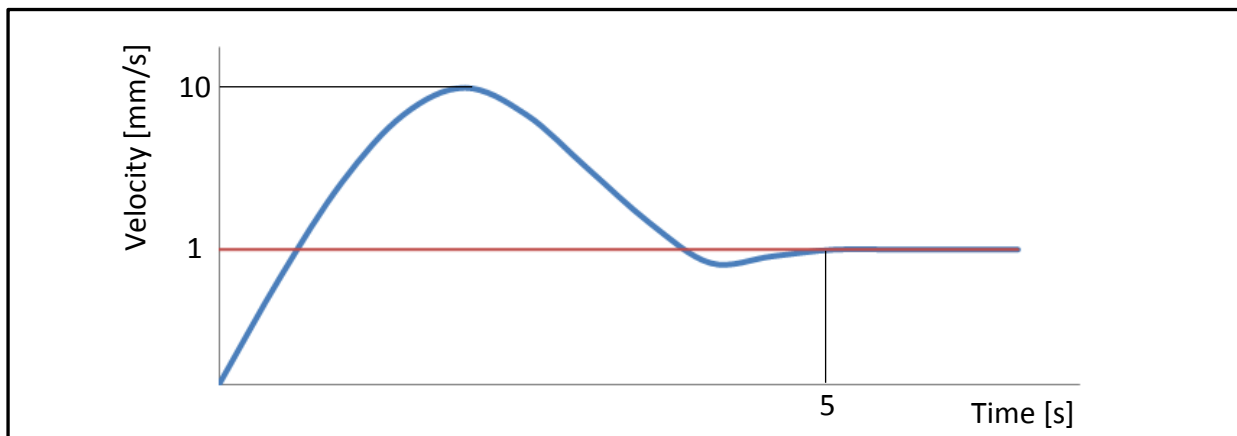


Chart 8: "System behaviour with PI-feedback control – valve open at the beginning"

In both charts the red line is the desired velocity and the blue line the actual velocity. If the valve is open at the beginning of a movement, the speed rises very quickly. Because of the system inertia it takes some time to close the valve. Here the problem is that a movement with low speeds for short distance is impossible. The start velocity is about 10 mm/s and it takes about five seconds to reduce it to 1mm/s. This system behaviour is unacceptable for our needs. Since the most tests are done with a speed of approximately 1mm/s the first idea was to close the valve at the beginning. The resulting system behaviour can be seen in Chart 9: "System behaviour with PI-feedback control – valve closed in the beginning".

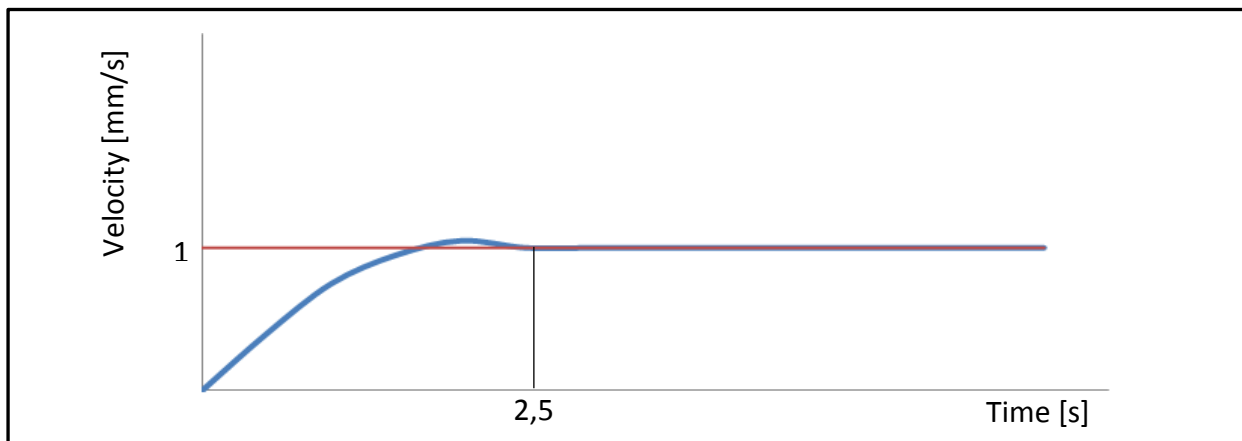


Chart 9: “System behaviour with PI-feedback control – valve closed in the beginning”

This behaviour was more suitable for our needs but still not perfect. The big benefit in comparison to the other one is that we start with a velocity of 0mm/s and are raising it to our designated velocity in about two and half seconds. So tests with low speeds and short movement distances are possible as well. The fact that it takes some time to reach higher velocities is not optimal, but in our case it is not a big deal because at the moment only quasi static test are performed with very low speeds. To further improve the behaviour of our system and to speed up the reaching of higher velocities a new feedback control concept was found.

8.2.4 PI - velocity feedback control with pre-control

To overcome the problems of a simple PI – Feedback controller, mentioned in the previous subchapter, a new feedback control concept was developed with the concept of a feedback control with pre-control. The theoretical background and the implementation are described in this subchapter.

Theoretical Background

The desirable property of feedback control in terms of reaction to errors or dynamic behaviour of the control path and the property of a control in an open-end chain to change the system in a very short period of time is the reason for the combination of those. You can see an illustration in Figure 41: “Feedback control combined with a pre-control (Lunze, 2013)”. The actuation variable is calculated with two parts, the part of the pre-control $u_v(t)$ and the part of the feedback controller $u_R(t)$. The task of the pre-controller is to reach the command variable as fast as possible, the task of the feedback controller is to equalize the disturbance value $d(t)$. This type of control circle is also called feedback controller with two degrees of freedom.

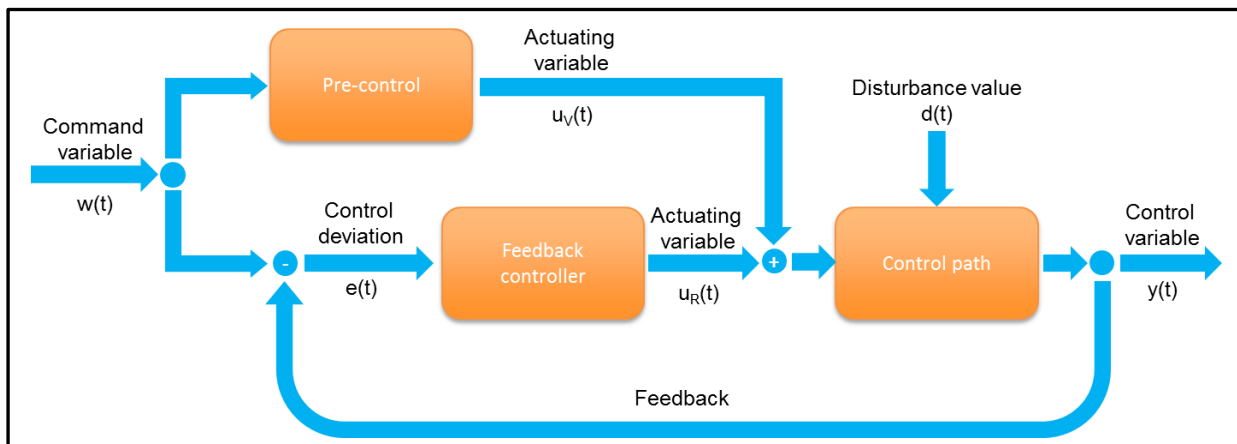


Figure 41: “Feedback control combined with a pre-control (Lunze, 2013)”

Because the parts of a system with two degrees of freedom are independent from each other they can be designed separately:

- **Stabilization and disturbance compensation:**

The feedback controller has to be selected to be stable and to stable an unstable control path. It has to have a good disturbance reaction.

- **Guidance reaction:**

The pre-control has to be designed according to a good guidance reaction.

To illustrate this topic in a better way an example will be discussed. Given is a filling level feedback control with pre-control which is shown in Figure 42: “Filling level feedback control with pre-control (Lunze, 2013)”.

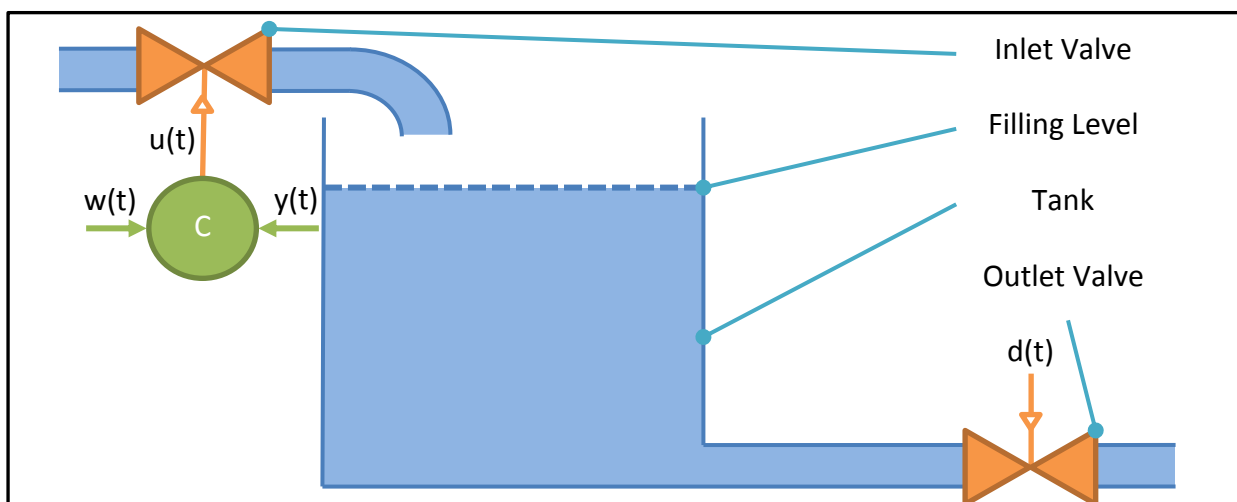


Figure 42: “Filling level feedback control with pre-control (Lunze, 2013)”

The goal of this example is to fill the empty tank to a level of 1m and keep it, this is the command variable $w(t)$. The extraction of water at the bottom acts as disturbance variable $d(t)$. The actuating variable $u(t)$ is influencing the volumetric flow rate to the tank. The actual filling level is the control variable $y(t)$. For the feedback control part a proportional control is used with the equation

$$u(t) = k_p * e(t)$$

Equation 11: "Feedback controller law – filling level (Lunze, 2013)"

You can see the behaviour of the different systems of the example in Figure 43: "Behaviour in an open-chain control (left) and in a feedback controller (right) (Lunze, 2013)" and Figure 44: "Behaviour of the combination of a feedback controller with a pre-controller (Lunze, 2013)"

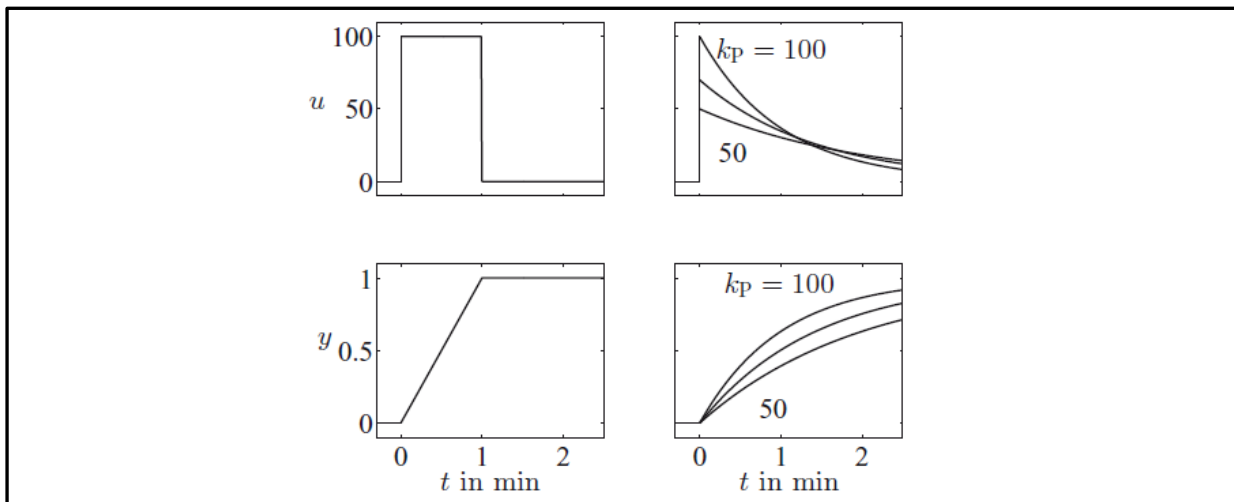


Figure 43: "Behaviour in an open-chain control (left) and in a feedback controller (right) (Lunze, 2013)"

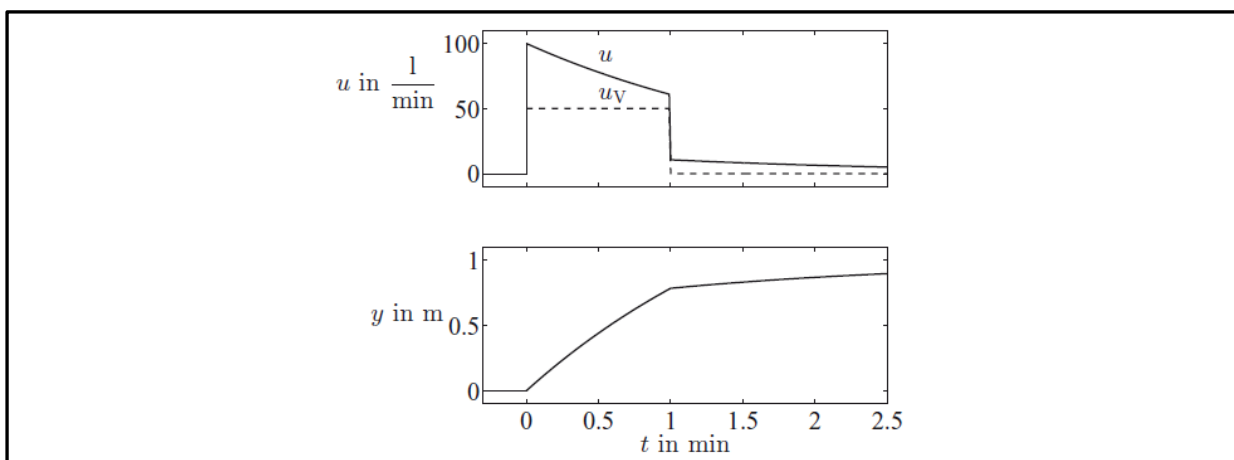


Figure 44: "Behaviour of the combination of a feedback controller with a pre-controller (Lunze, 2013)"

It is obvious that a pre-control can only be used when you know the exact surroundings of the system. If the tank would not be empty in the beginning or the volume would not be the same as in the calculations the pre-control with open-end chain could not be used. In comparison a feedback control would react on the changed surroundings without any problems.

Implemented feedback control with pre-control

With the previously explained theoretical background a new system was defined. The developed system can be seen in Figure 45: “Implemented PI-feedback velocity control with pre-control – PRESTO 420”.

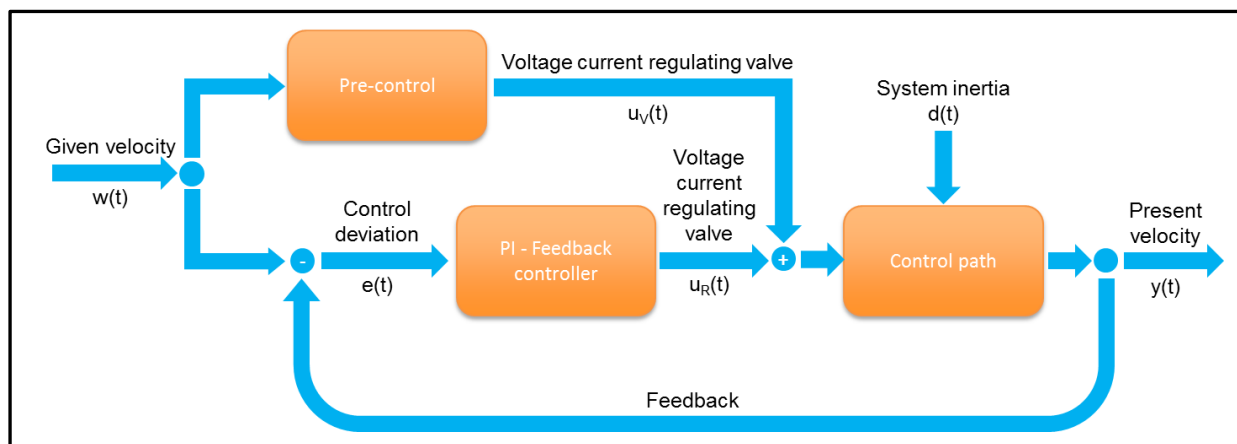


Figure 45: “Implemented PI-feedback velocity control with pre-control – PRESTO 420”

To reach higher movement velocities in shorter time the concept of a pre-control is used. Here the benefit of a control is used to open the valve from the beginning to a certain level. The PI-feedback controllers’ task is to accurately adjust the velocity and the task of the control is to roughly adjust the velocity. This is only possible because we know the system behaviour to pre adjust the speed. In Chart 10: “Velocity – Valve Voltage curve – pressure direction” and in Chart 11: “Velocity – Valve Voltage curve – tensile direction” you can see the behaviour of the movement speed in pressure and tensile mode. The speed is depending on the voltage of the current regulating valve of the pressure force and the voltage of the current regulating valve for the movement direction. With this chart the setup of the pre-control was done. A in depth knowledge of this behaviour is critical for the functionality of the system.

The data for Chart 10 and Chart 11 was gathered with a test run of the PRESTO 420. The settings for the force current regulating valve and for the velocity current regulating valve (tensile and pressure direction) were manually controller, the velocity was monitored. On the

x-axis the voltage for the velocity current regulating valve and on the y-axis the velocity is shown.

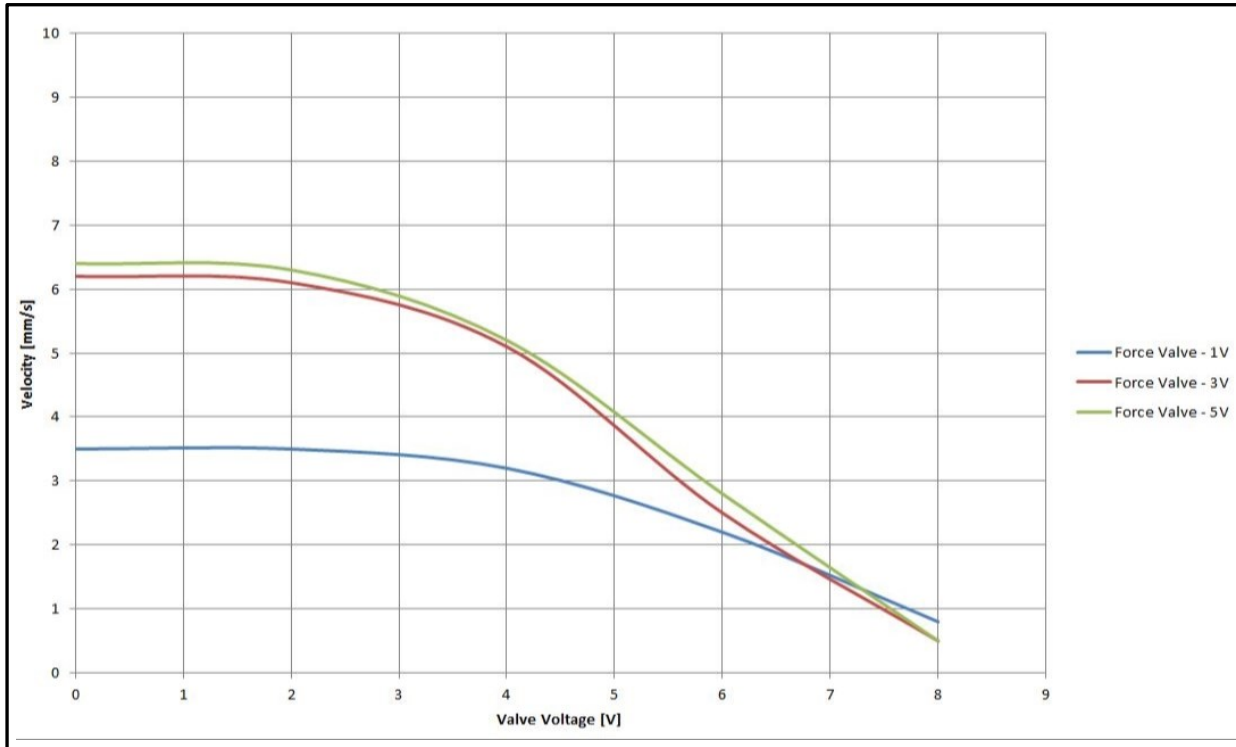


Chart 10: "Velocity – Valve Voltage curve – pressure direction"

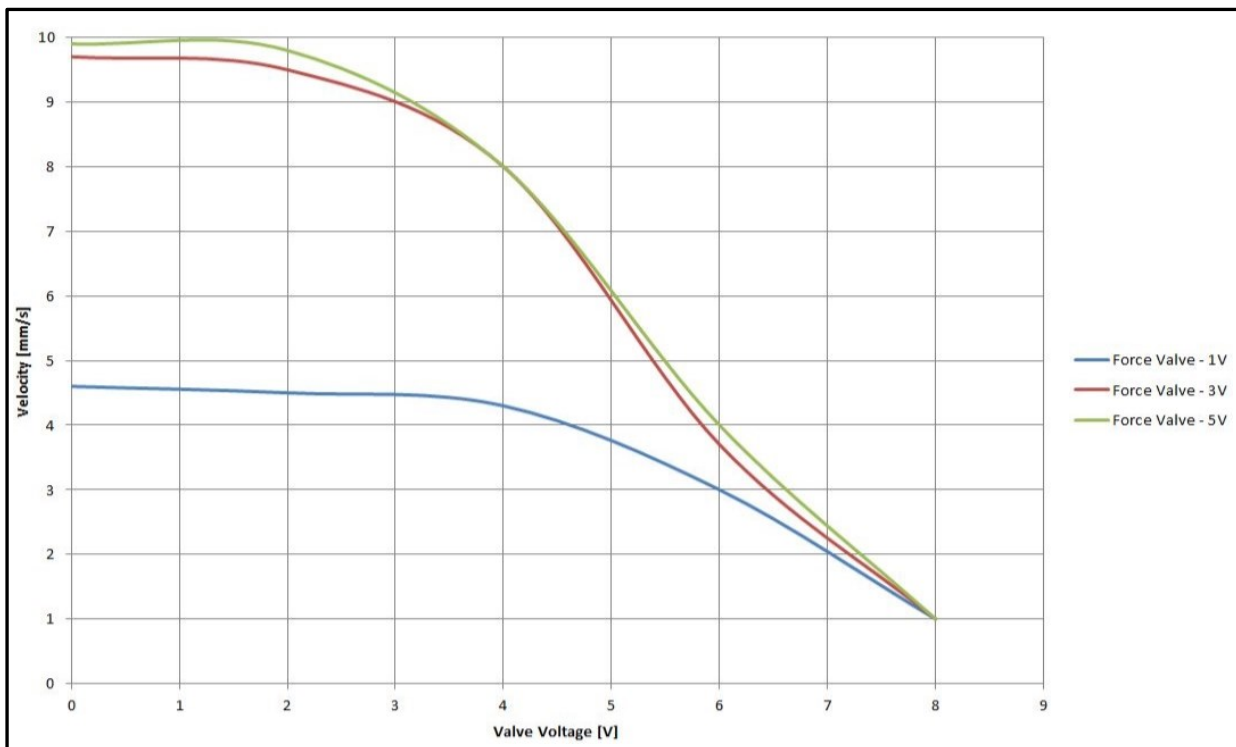


Chart 11: "Velocity – Valve Voltage curve – tensile direction"

The result of these two charts is the realization that a pre-control has to be developed which changes its parameters when changing designated speeds as well as when changing the test force and the test mode (tensile/pressure). With a higher force the speed raises much more quickly, this fact has to be considered. The result of the implemented PI – feedback control with pre-control can be seen in Chart 12: “System behaviour with pre-control – valve closed in the beginning”. This chart is only valid for movement without a pressure or tensile test.

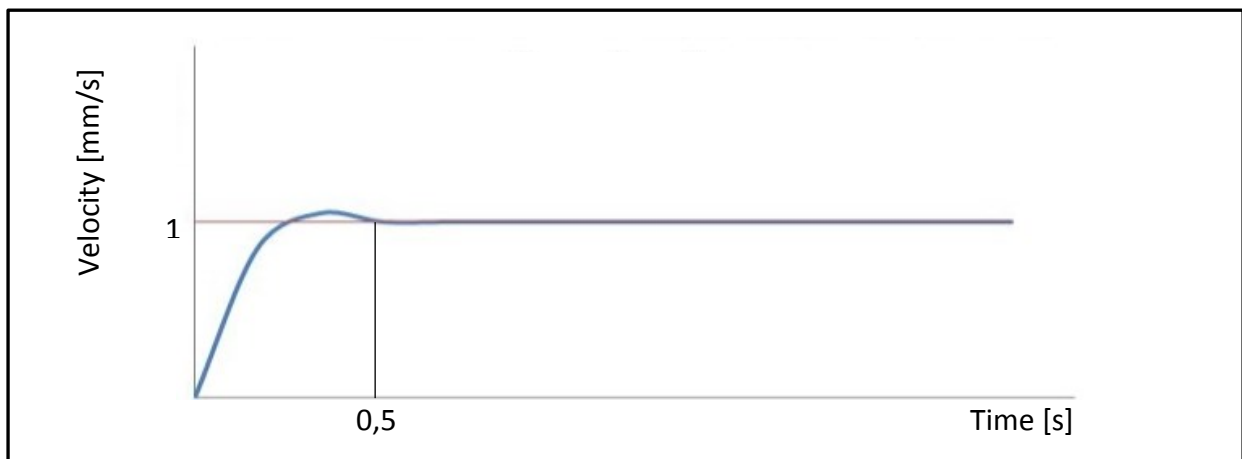


Chart 12: “System behaviour with pre-control – valve closed in the beginning”

The desired velocity is reached faster than with a PI – feedback control which was the intention of the system. The big benefit of this system is that it does not matter if you want to perform a test with low or high speeds because it reacts immediately. As previously mentioned the system with a PI-feedback control where in the beginning the valve was open it took about five seconds to reduce the speed from 10mm/s to 1mm/s. When the valve was closed it took about two and a half seconds to raise the speed from 0mm/s to 1mm/s. Now the feedback control has only to fine adjust the speed, because of that it only takes about 0,5s to reach the desired speed. The time to reach the speed is only depending on the adjustment error of the pre-control.

9 Software Validation Tests

To show the full functionality of the PRESTO 420 test facility and its software validation tests will be performed.

To demonstrate that the system is performing its measurement task correctly a material, which properties are well known, will be used for the first test under quasi static conditions. The reason for the second test is to show its functionality under real conditions, so we will test a battery cell under quasi static conditions.

Both tests will be done according to the guidelines of a FreedomCAR controlled crush test:

“The first stage (displacement control) is a displacement of 15% of the module’s height, which is held for five minutes. The second stage (force control and displacement limit) is limited by either a 50% displacement of the module’s height or a force of 1000 times the module’s mass; whichever condition occurs first is held for five minutes.” (Doughty, et al., 2006)

The test charts, which are shown in this chapter, were recorded with a test speed of 1mm/s. In all tests the force measurement was performed with calibrated uniaxial load cells (max load 500kN).

9.1 Crush Tube – Validation tests

The specimen is a pre-crushed crush tube with the dimensions:

- Length: 200mm
- Diameter: 45mm
- Wall thickness: 1mm

The test specifications are:

- Controlled Crush test according to FreedomCAR regulations
- Test velocity 1mm/s
- Stage 1: 15% displacement of part height = 30mm
hold position for 300s
- Stage 2: 50% displacement of part height = 100mm
hold position for 300s

9.1.1 Crush tube test example

As you can see in Figure 46: “PRESTO 420 – Validation Test 1 – Crush Tube” and in the following chart that the first validation test was successfully performed. The picture of “Stage 1” was taken right after the 15% displacement and the picture of “Stage 2” was taken right after the 50% displacement.



Figure 46: “PRESTO 420 – Validation Test 1 – Crush Tube”

In Chart 13: “PRESTO 420 – Validation Test 1 – Force – Time Chart” & Chart 14: “PRESTO 420 – Validation Test 1 – Force – Distance Chart” you can see the data which was measured during the tests. The behaviour of the crush tube was the same as expected, no conspicuousness. You can see the good folding behaviour of the crush tubes in both charts. In Chart 12: “PRESTO 420 – Validation Test 1 – Force – Time Chart” you can figure out that in the whole waiting stage a small downward movement of the piston is recognizable. This effect reduces the force from 25kN to 15kN, so definitely a plastic deformation happened. The force would not decrease if there was an elastic springback. The downward movement is so too small to see without a measurement chart. In Chart 13: “PRESTO 420 – Validation Test 1 – Force – Distance Chart” the area of the waiting phases is marked to see that no upward movement occurred. Here the dropdown of the force can be seen as well.

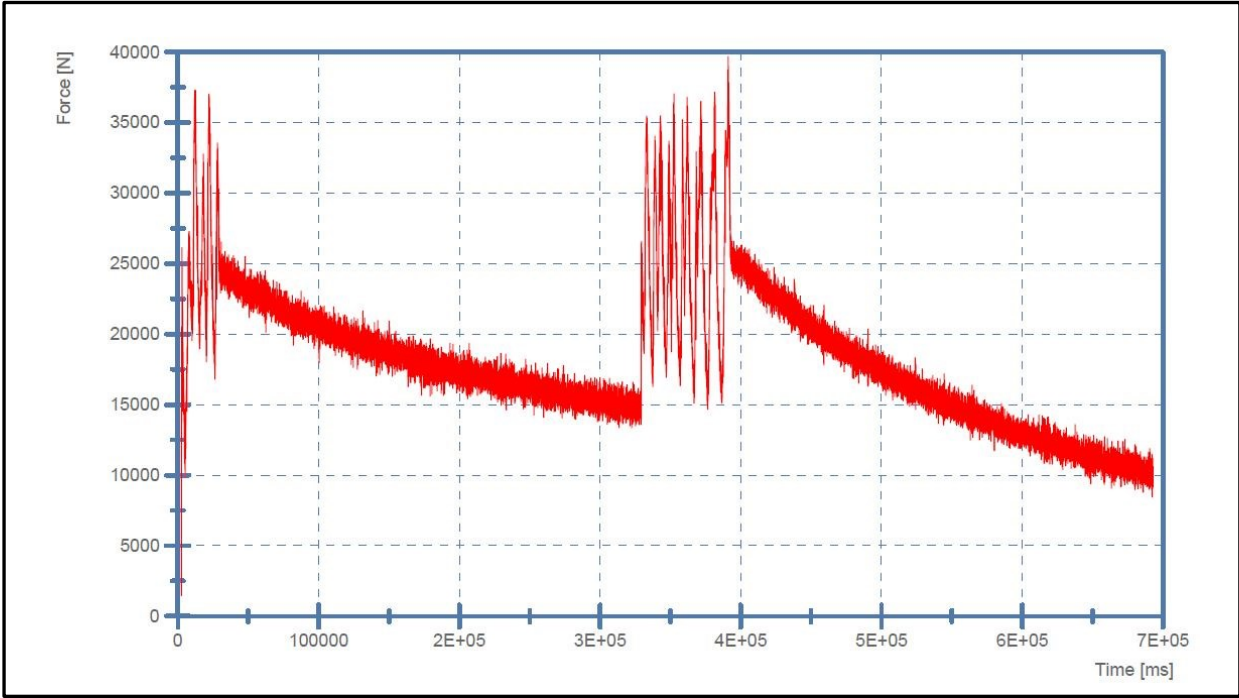


Chart 13: "PRESTO 420 – Validation Test 1 – Force – Time Chart"

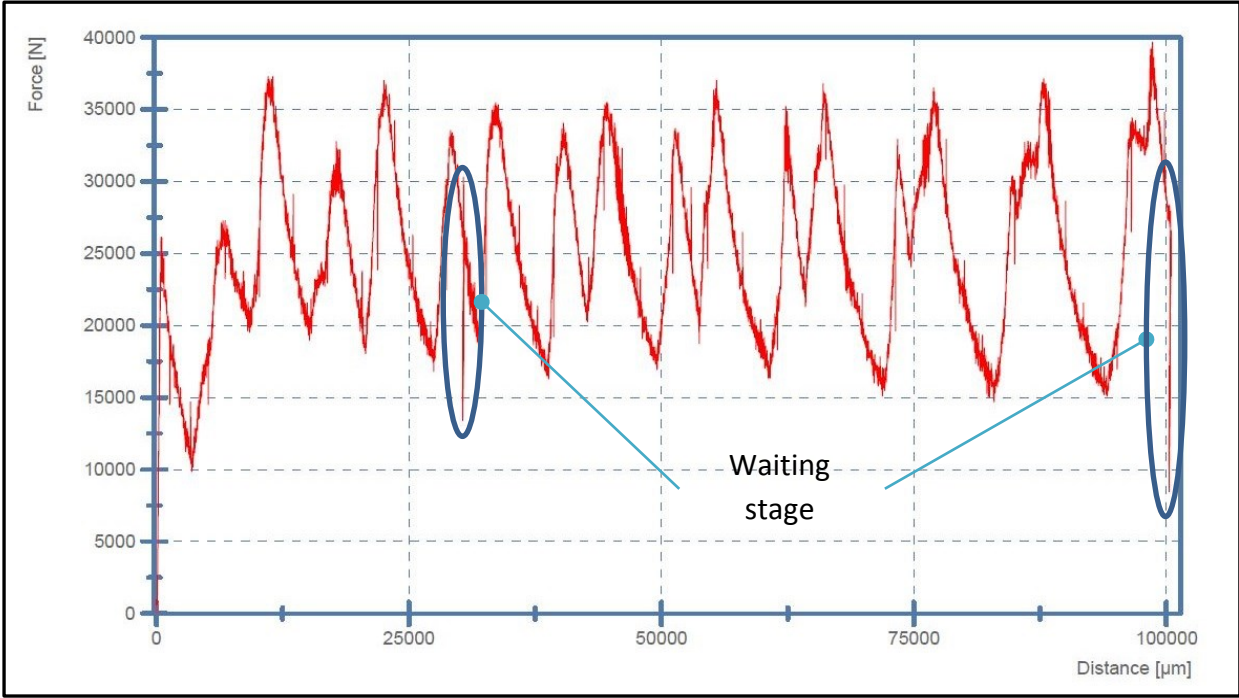


Chart 14: "PRESTO 420 – Validation Test 1 – Force – Distance Chart"

9.1.2 Comparison

In Chart 15: “PRESTO 420 – Validation Test Crush Tube Comparison – Force – Time Chart” & Chart 16: “PRESTO 420 – Validation Test Crush Tube Comparison – Force – Distance Chart” you can see the recorded measurement data of two crush tube validation tests. With the measurement charts you can easily figure out that with the new software a very high reproducibility was reached. The displacement and the waiting time of the two tests were exactly the same.

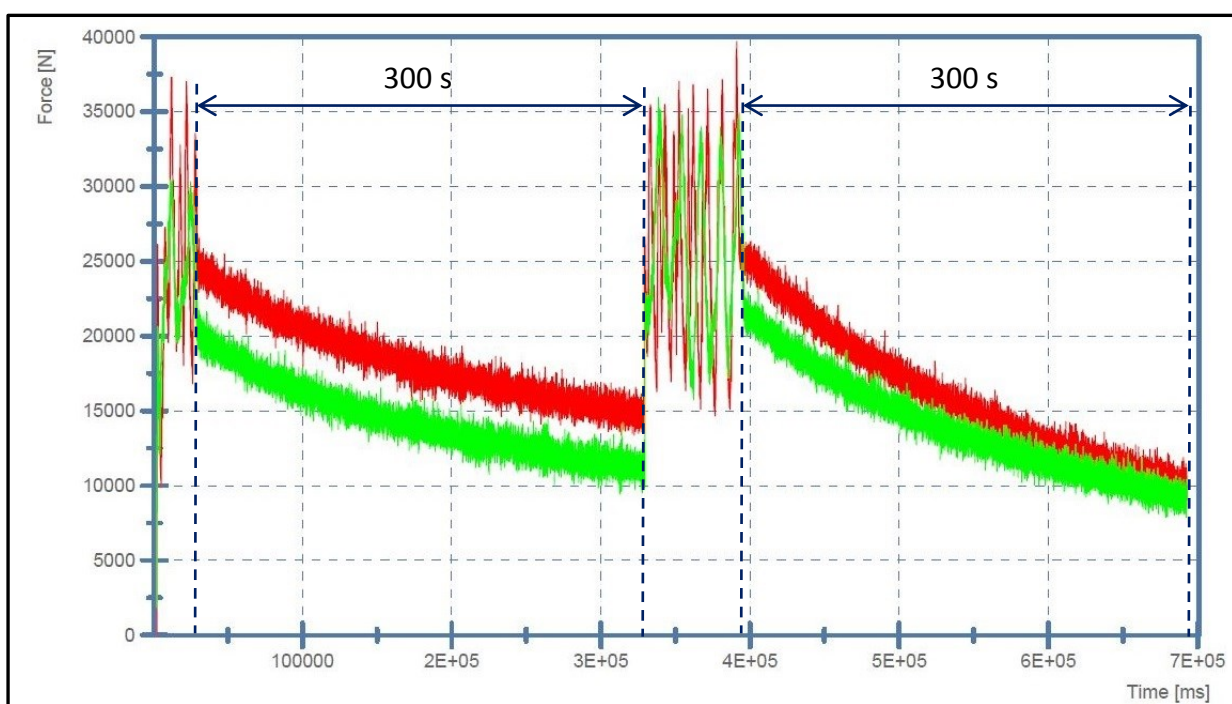


Chart 15: “PRESTO 420 – Validation Test Crush Tube Comparison – Force – Time Chart”

To see the time dependent behaviour of the crush tubes a force time chart was recorded. The data shows the high reproducibility of the waiting stages, they were marked in the chart. Both stages had a 300s waiting time.

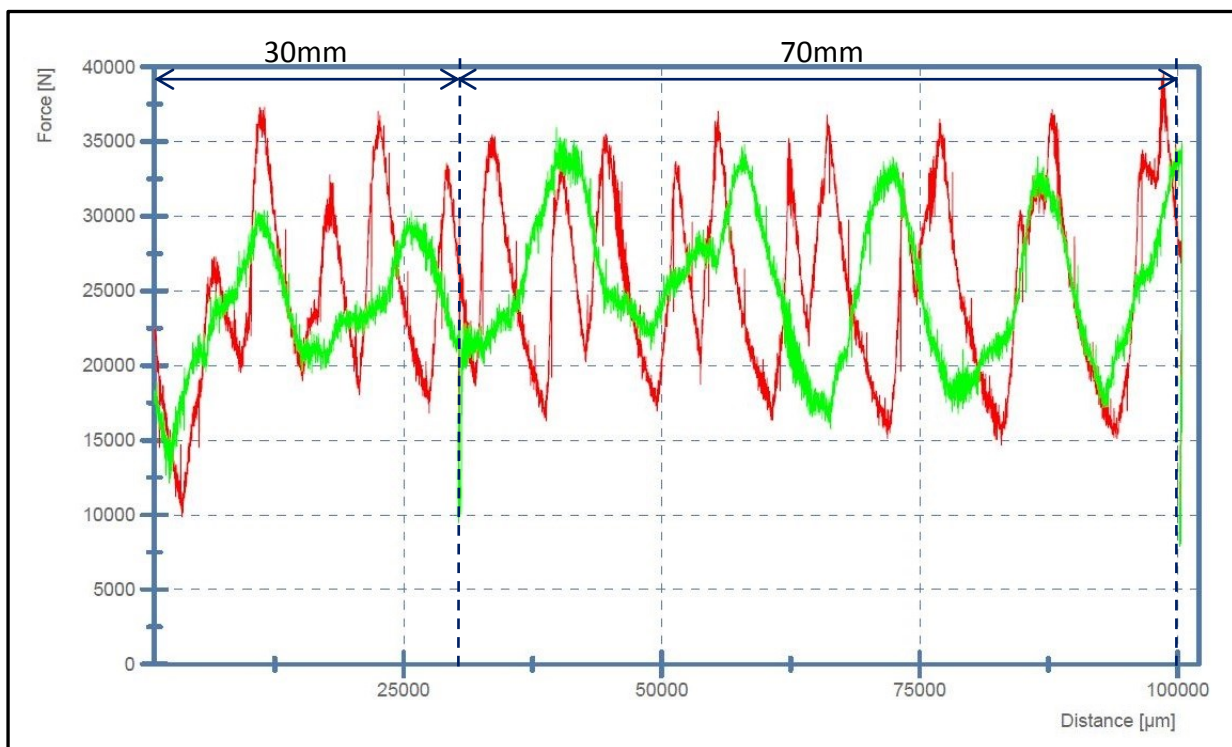


Chart 16: "PRESTO 420 – Validation Test Crush Tube Comparison – Force – Distance Chart"

To see the displacement dependent behaviour a force - distance chart was recorded. This data shows the high reproducibility of the pressure stages, they were marked in the measurement chart. You can see the first stage with 30mm displacement and the second one with 70mm displacement. These facts can also be seen in Chart 17: "PRESTO 420 – Validation Test Crush Tube Comparison – Distance – Time Chart".

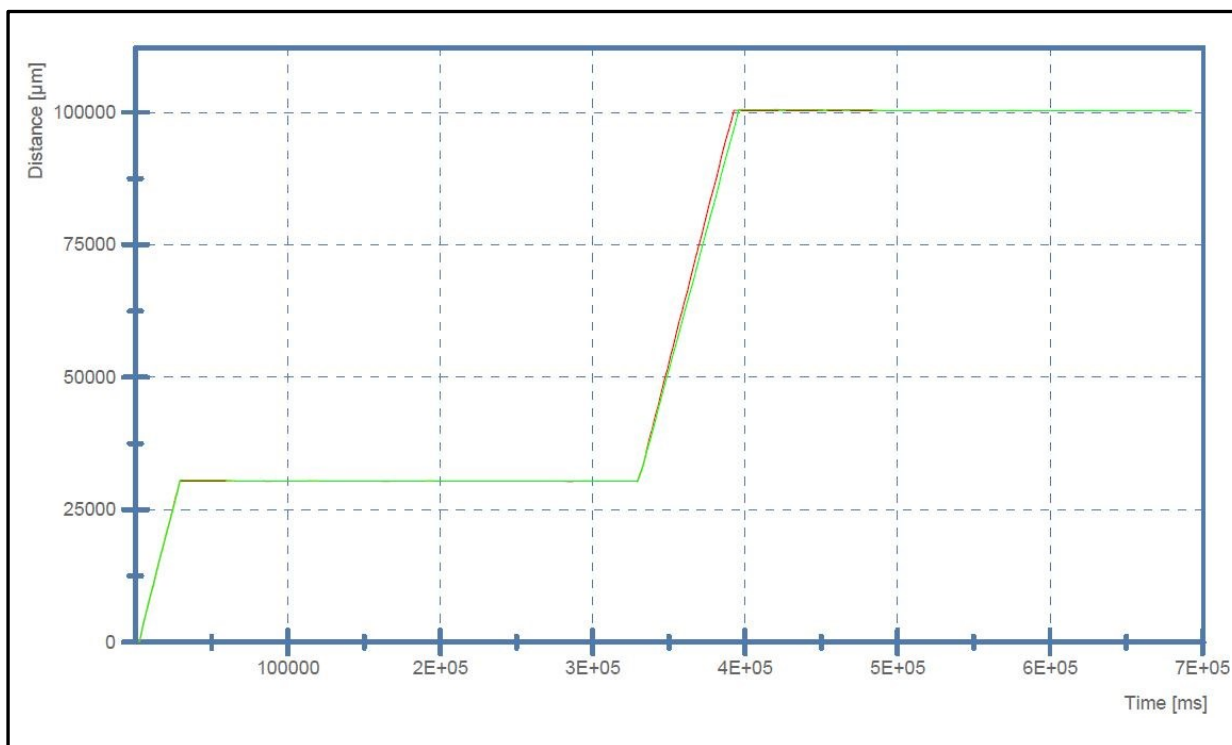


Chart 17: “PRESTO 420 – Validation Test Crush Tube Comparison – Distance – Time Chart”

Here the high reproducibility can be seen as well. The gradient of a distance-time chart equates to the velocity at this specific point. Since the gradient is almost constant it seems that the feedback control fulfils its task in a good way.

To evaluate the feedback control an average velocity can be calculated:

$$v = \frac{s}{t - (t_{w1} + t_{w2})} = \frac{100000}{692865 - 600000} = 1,076 \text{ mm/s}$$

Equation 12: “Calculated average velocity – Validation test 1”

$$v = \frac{s}{t - (t_{w1} + t_{w2})} = \frac{100000}{696015 - 600000} = 1,044 \text{ mm/s}$$

Equation 13: “Calculated average velocity – Validation test 2”

In the first validation test the average velocity was about 7,6% and in the second test the average velocity was about 4,4% higher than the goal velocity. For a quasi-static test this should be accurate enough, but there is still potential for improvements.

With this validation tests the principle functionality of performing a test with the PRESTO 420 was proofed.

9.2 Battery – Validation tests

The specimen is an uncharged Swing™ 4400 Battery cell with the dimensions:

- Length: 64,8mm
- Width: 37,1mm
- More details see appendix

The test specifications are:

- Controlled Crush test according to FreedomCAR regulations
- Test velocity 1mm/s
- Stage 1: 15% displacement of part height = 9,72mm
hold position for 300s
- Stage 2: 50% displacement of part height = 32,4mm
hold position for 300s

9.2.1 Battery test example

As you can see in Figure 47: “PRESTO 420 – Validation Test 7 – Battery” and in the following chart the seventh validation test was also successfully performed. The “Pre-Test” picture was taken right before the 15% displacement and the picture “Post-Test” was taken right after the 50% displacement.



Figure 47: “PRESTO 420 – Validation Test 7 – Battery”

In Chart 18: “PRESTO 420 – Validation Test 7 – Force – Time Chart” & Chart 19: “PRESTO 420 – Validation Test 7 – Force – Distance Chart” you can see the data which was measured during the tests. The behaviour of the battery cell was the same as expected, no conspicuousness. The material failure (the burst of the battery) happened approximately at 41% displacement, you can see it in the marked area of the force – distance chart. In Chart 16: “PRESTO 420 – Validation Test 7 – Force – Time Chart” you can figure out that in the whole waiting stage a small downward movement of the piston is recognizable. This effect reduces the force from approximately 8kN to 4kN, so definitely a plastic deformation happened for the same reasons as in the crush tube tests. The downward movement is so small, so it cannot be seen without a measurement chart. In Chart 17: “PRESTO 420 – Validation Test 7 – Force – Distance Chart” the area of the waiting phases is marked to see that no upward movement has happened. Here the dropdown of the force can be recognized as well.

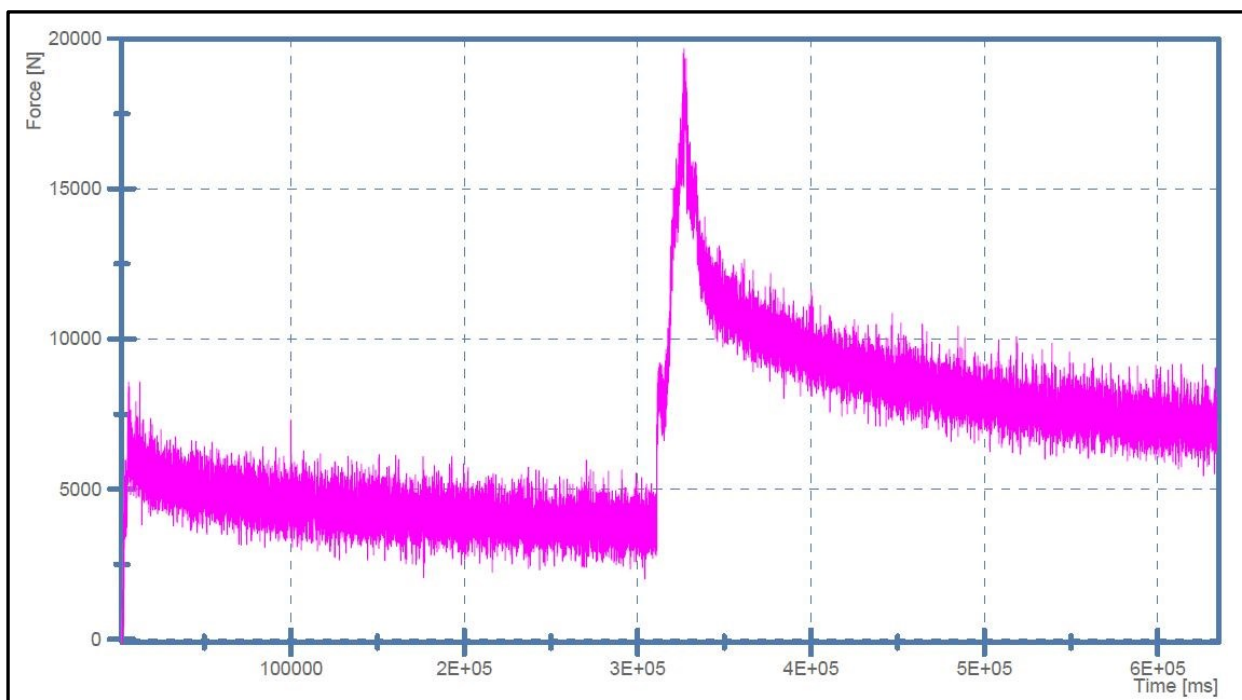


Chart 18: “PRESTO 420 – Validation Test 7 – Force – Time Chart”

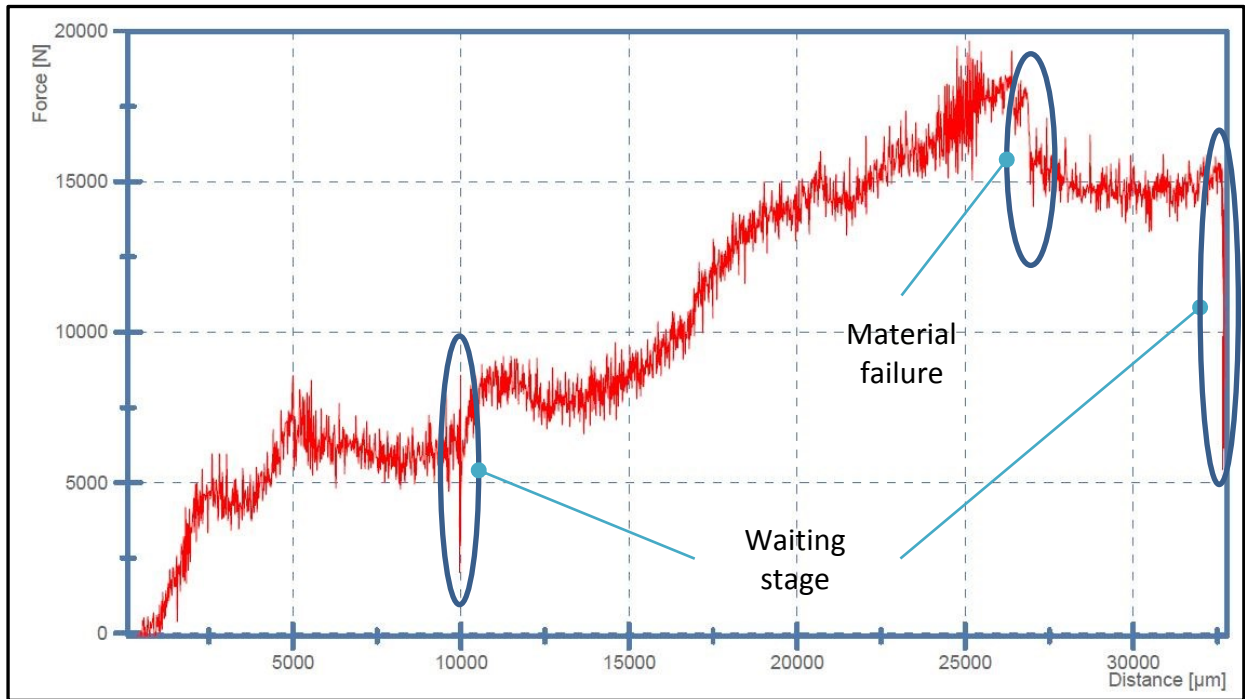


Chart 19: "PRESTO 420 – Validation Test 7 – Force – Distance Chart"

9.2.2 Comparison

In the following charts you can see the results of the battery validation tests in two different charts:

- Force – Time
- Force – Distance

You can see in both charts the high reproducibility in terms of reaching the abort limits of the different stages in the same way.

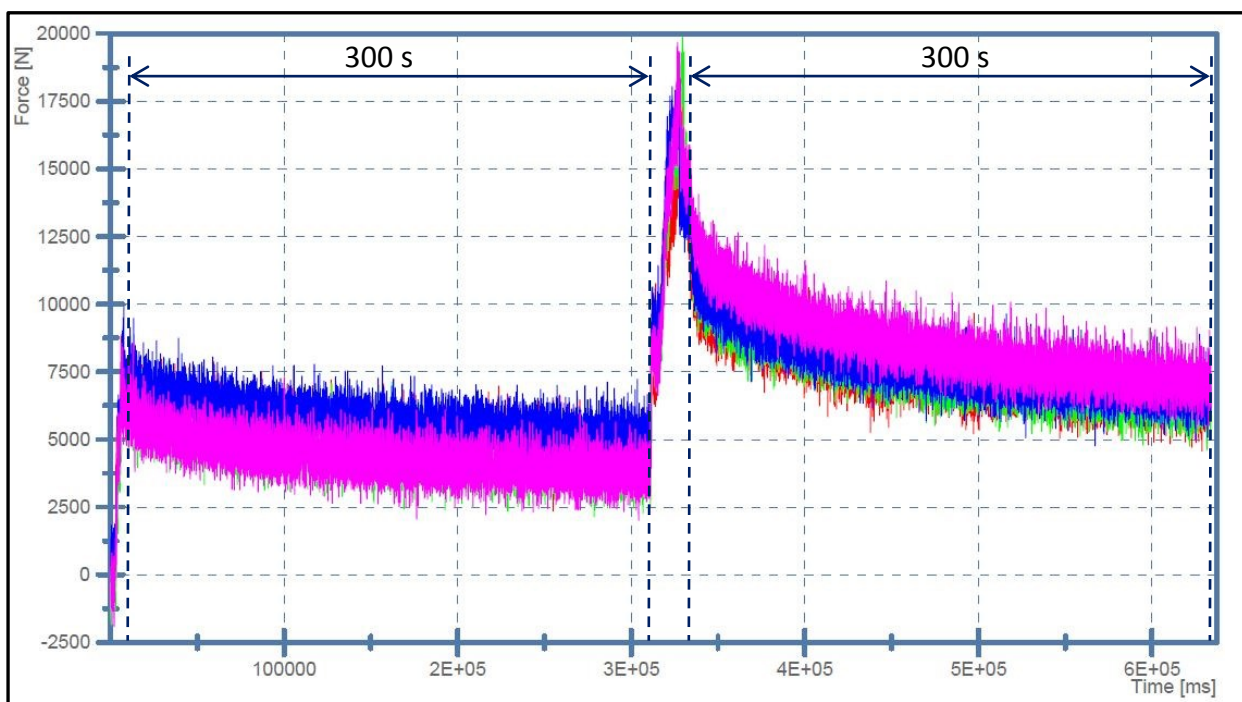


Chart 20: "PRESTO 420 – Validation Test Battery Comparison – Force – Time Chart"

To illustrate the high reproducibility in terms of waiting stages in a better way the two waiting stages were marked in Chart 20: "PRESTO 420 – Validation Test Battery Comparison – Force – Time Chart". Both stages had a waiting time of 300s.

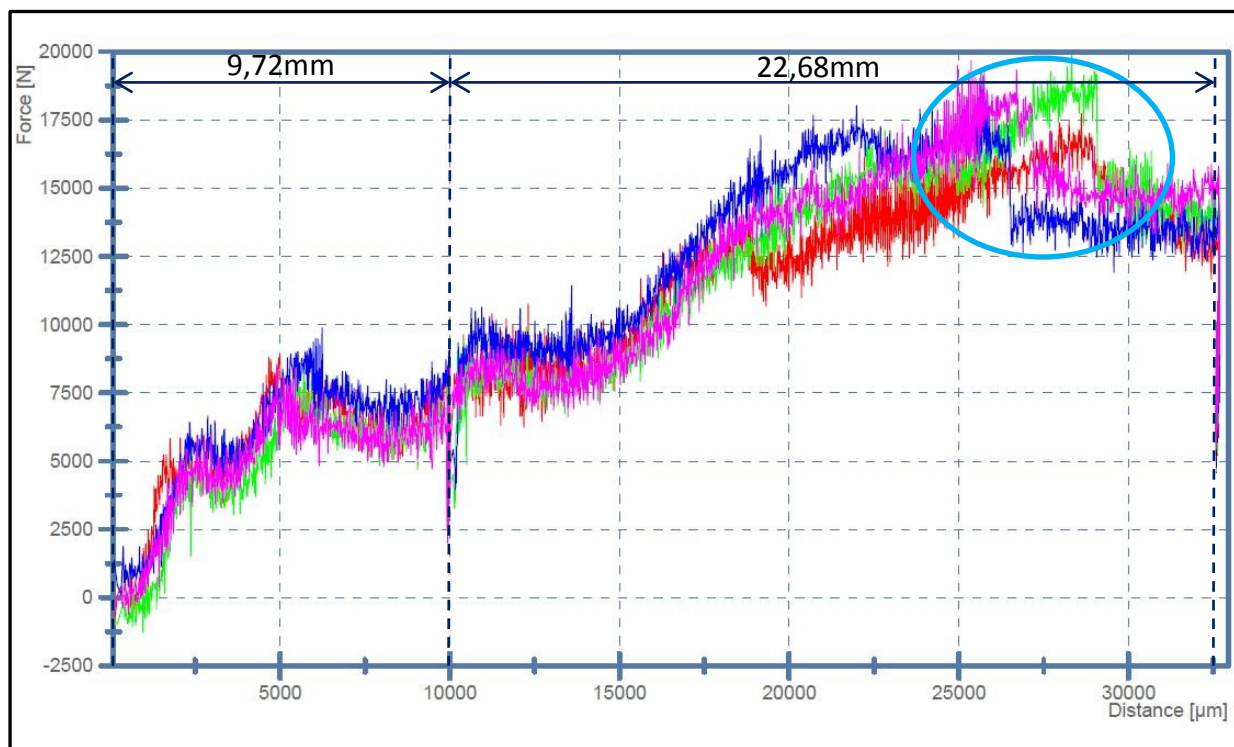


Chart 21: "PRESTO 420 – Validation Test Battery Comparison – Force – Distance Chart"

In Chart 21: "PRESTO 420 – Validation Test Battery Comparison – Force – Distance Chart" you can see the sudden material failure in the marked area. The PRESTO 420 had no problems with the data acquisition while the force drop-down happened.

The high reproducibility in terms of displacement can also be seen in Chart 22: "PRESTO 420 – Validation Test Battery Comparison – Distance – Time Chart".

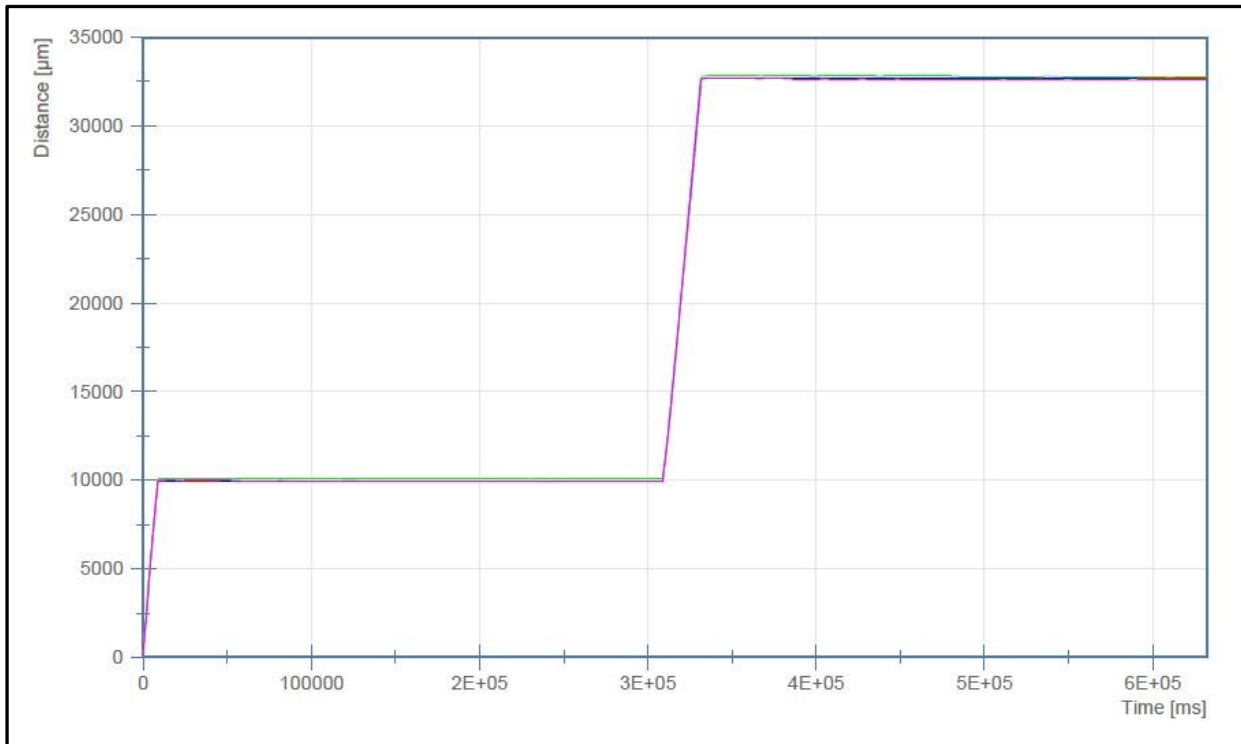


Chart 22: "PRESTO 420 – Validation Test Battery Comparison – Distance – Time Chart"

The gradient of a distance-time chart equates to the velocity at this specific point. Since the gradient is almost constant it seems that the feedback control fulfils its task in a good way.

To evaluate the feedback control an average velocity can be calculated:

$$v = \frac{s}{t - (t_{w1} + t_{w2})} = \frac{32400}{631455 - 600000} = 1,030 \text{ mm/s}$$

Equation 14: "Calculated average velocity – Validation test 4"

$$v = \frac{s}{t - (t_{w1} + t_{w2})} = \frac{32400}{631980 - 600000} = 1,013 \text{ mm/s}$$

Equation 15: "Calculated average velocity – Validation test 5"

$$v = \frac{s}{t - (t_{w1} + t_{w2})} = \frac{32400}{631860 - 600000} = 1,017 \text{ mm/s}$$

Equation 16: "Calculated average velocity – Validation test 6"

$$v = \frac{s}{t - (t_{w1} + t_{w2})} = \frac{32400}{631680 - 600000} = 1,023 \text{ mm/s}$$

Equation 17: "Calculated average velocity – Validation test 7"

In validation test 4 the average velocity was 3% higher than the desired value, in validation test 5 it was 1,3%, in validation test 6 it was 1,7% and in validation test 7 it was 2,3%. So the velocity was in average 2,1% higher than the goal velocity which is a pretty good value. The smaller deviation in comparison to the crush tube validation test results from the lower pressure force. Therefore the feedback control should still be improved for tests with higher forces.

10 Conclusion & Outlook

This thesis describes the development of an “easy to use” automation software for a multifunctional tensile and pressure test facility for batteries and other materials as well as the development of a feedback control for its movement velocity. Goal of the PRESTO 420 test facility is to generate empirical data of the material behaviour under quasi static tensile and pressure stress to generate a simulation model of the material on computer systems.

The programming of the software and the development of the feedback control was fulfilled to complete satisfaction for the Vehicle Safety Institute. An automatic test mode was implemented as well as a manual mode. The detection of several different abort criteria is as well possible: displacement, force, time and short-circuits. An automatic test can be freely designed out of three stages: waiting stage, pressure stage, tensile stage. These stages can be combined in every way to one test sequence. The possibility to fulfil tests according to the guidelines of FreedomCAR is as well given.

The performed validation tests have shown that the new software works properly. A high reproducibility is given through the automatic mode and its automated detection of abort reasons. The feedback control provides constant test velocities which is as well crucial for quasi static test. The validation tests with battery cells have proven that the requirements for testing of electrical storage media are as well given.

Since the PRESTO 420 is at the moment unique on the market a significant value generation in terms of market competitiveness of the Vehicle Safety Institute at the Graz University of Technology could be realized with this thesis. The knowhow generation in terms of testing electrical storage media has been increased as well, which is a big benefit because of the e-mobility boom and the boom of other related industries.

To further develop the PRESTO 420 test facility different actions can happen. For testing energy storage systems a test enclosure can be built. This would enable the possibility to test charged cells as well. An analysis of the emitting gases would also be a big benefit. The improvement of

the software should be a goal in the future. There is still potential to enable new test cases or implement new abort reasons. To gain more and more usability and further reduce the human failure potential is desirable. An embedded light source and video system could be a benefit as well. Shadows of the specimen make it really difficult to record a test video. This could be solved with a LED light source which could be built into the PRESTO420.

11 List of References

- Arbeitskreis der Automobilimporteure. 2012.** Fakten statt Vorurteile. *www.wko.at*. [Online] 2012. [Cited: 11 March 2013.]
http://wko.at/fahrzeuge/main_frame/statistik/wf_automobil//Folder%20Autoland%20%C3%96sterreich.pdf.
- Barnett, Brian, et al. 2013.** *Batteries for Sustainability*. New York : Springer New York, 2013. 978-1-4614-5790-9.
- Böge, Alfred. 2013.** *Handbuch Maschinenbau - Grundlagen und Anwendungen der Maschinenbau-Technik*. s.l. : Springer Fachmedien Wiesbaden, 2013. 978-3-8348-2479.
- Boston-Power, Inc. 2013.** [Online] 2013. [Cited: 16 May 2013.] <http://www.boston-power.com/products/swing-4400>.
- Czichos, Horst and Hennecke, Manfred. 2012.** *HÜTTE - Das Ingenieurwissen*. s.l. : Springer Berlin Heidelberg, 2012. 978-3-642-22849-0.
- Doughty, Daniel H. and Crafts, Chris C. 2006.** *FreedomCAR Electrical Energy Storage System Abuse Test Manual for Electric and Hybrid Electric Vehicle Applications*. California : s.n., 2006.
- Grote, Karl-Heinrich and Feldhusen, Jörg. 2012.** *Dubbel - Taschenbuch für den Maschinenbau*. s.l. : Springer Berlin Heidelberg, 2012. 978-3-642-17306-6.
- Knauder. 2012.** *Entwicklung einer Prüfumgebung zur Untersuchung der mechanischen Eigenschaften von lithiumbasierenden Batterien*. Graz : Graz University of Technology, 2012.
- Lunze, Jan. 2013.** *Regelungstechnik 1*. s.l. : Springer Berlin Heidelberg, 2013. 0937-7433.
- National Instruments Corporation. 2011.** *Design, Entwicklung und Programmierung von FPGAs*. s.l. : National Instruments Corporation, 2011.
- . **2011.** *LabVIEW-Grundlagen 1*. 2011.
- . **2013.** *www.ni.com*. [Online] 2013. [Cited: 13 March 2013.] www.ni.com.
- Parker Hannifin GmbH. 2012.** *EO-Rohre für Fittinge und Flansche*. 2012.
- Wörn, Heinz and Brinkschulte, Uwe. 2005.** *Echtzeitsysteme*. Karlsruhe : Springer-Verlag Berlin Heidelberg, 2005. 978-3-540-27416-2.

12 List of Figures

Figure 1: “Flowchart – approach master thesis”	3
Figure 2: “Components of the test facility – PRESTO 420 (Knauder, 2012)”.....	6
Figure 3: “Hardware components of the CompactRIO – controlling and measurement system (Knauder, 2012)”	8
Figure 4: “Principle of the controlling and measurement software (Knauder, 2012)”	9
Figure 5: “User Interface – Manual Operations Knauder”	10
Figure 6: “Program structure PRESTO 420 AUT”	11
Figure 7: “Measurement System – CompactRIO 9024”	14
Figure 8: “Control System – CompactRIO 9014	18
Figure 9: “Difference between LabVIEW™ and C++”	20
Figure 10: “Programming levels in CompactRIO real-time systems (National Instruments Corporation, 2011)”	21
Figure 11: “Workload PRESTO 420 software development”	23
Figure 12: “Basic principle of automation (Wörn, et al., 2005)”	24
Figure 13: “Software specific improvement potential PRESTO 420”	26
Figure 14: “Test procedure Controlled Crush test (Doughty, et al., 2006).....	36
Figure 15: “Program structure and changes PRESTO 420 AUT”	37
Figure 16: “Functions of the data acquisition (Knauder, 2012)”	38
Figure 17: “Functions of the control modules”	39
Figure 18: “GUI – Manual Mode”	43
Figure 19: “GUI – Manual Mode chart view”	44
Figure 20: “Process Manual Test”	45
Figure 21: “GUI – Initialization Automatic Mode”	47
Figure 22: “GUI – Automatic Mode”	49

Figure 23: “GUI – Automatic Mode chart view”	50
Figure 24: “Process Automatic Test”	51
Figure 25: “GUI – Selection Measurement Channels Manual & Automatic Mode”	52
Figure 26: “GUI – Parameters Sensors Manual & Automatic Mode”	53
Figure 27: “GUI – Load Cell Database”	54
Figure 28: “GUI – Movement Velocity & Pressure Force Manual & Automatic Mode”	55
Figure 29: “GUI – Measurement File Manual & Automatic Mode”	56
Figure 30: “GUI – Abort Reasons Manual Mode”	57
Figure 31: “GUI – Definition Test Sequence Automatic”	58
Figure 32: “POP - UP maximum force Input”	59
Figure 33: “GUI – Reference Positions Manual Mode”	60
Figure 34: “GUI – Reference Positions Automatic Mode”	61
Figure 35: “Hydraulic system scheme old – PRESTO 420”	62
Figure 36: “Hydraulic system scheme new – PRESTO 420”	63
Figure 37: “Structural design of a multilayer control (Lunze, 2013)”	65
Figure 38: “Principle of a feedback control (Lunze, 2013)”	66
Figure 39: “Feedback controller types (Böge, 2013)”	67
Figure 40: “PI-feedback velocity control– PRESTO 420”	74
Figure 41: “Feedback control combined with a pre-control (Lunze, 2013)”	76
Figure 42: “Filling level feedback control with pre-control (Lunze, 2013)”	76
Figure 43: “Behaviour in an open-chain control (left) and in a feedback controller (right) (Lunze, 2013)”	77
Figure 44: “Behaviour of the combination of a feedback controller with a pre-controller (Lunze, 2013)”	77
Figure 45: “Implemented PI-feedback velocity control with pre-control – PRESTO 420”	78

Figure 46: “PRESTO 420 – Validation Test 1 – Crush Tube” 82

Figure 47: “PRESTO 420 – Validation Test 7 – Battery” 87

Figure 48: “Boston Power Swing™ 4400 Data Sheet ½ (Boston-Power, Inc., 2013)” i

Figure 49: “Boston Power Swing™ 4400 Data Sheet 2/2 (Boston-Power, Inc., 2013)” ii

13 List of Tables

Table 1: “Controlling system components (Knauder, 2012)”	7
Table 2: “Measurement System components (National Instruments Corporation, 2011)”	17
Table 3: “Control System components (National Instruments Corporation, 2011)”	19
Table 4: “Requirements for the measurement system (Knauder, 2012)”	32
Table 5: “Requirements for the control system extended”	33
Table 6: “Failure estimation average velocity”	72

14 List of Charts

Chart 1: “Development CO2-fleet consumption in Austria (Arbeitskreis der Automobilimporteure, 2012)”	1
Chart 2: “CPU-Workload as a function of the sample rate”	15
Chart 3: “Data-writing rate as a function of the sample rate”	16
Chart 4: “Characteristic curve of a proportional feedback controller (Böge, 2013)”	68
Chart 5: “Step function & response of a proportional feedback controller (Böge, 2013)”	68
Chart 6: “Step function & response of an integral feedback controller (Böge, 2013)”	69
Chart 7: “Step function & response of a proportional integral feedback controller”	70
Chart 8: “System behaviour with PI-feedback control – valve open at the beginning”	74
Chart 9: “System behaviour with PI-feedback control – valve closed in the beginning”	75
Chart 10: “Velocity – Valve Voltage curve – pressure direction”	79
Chart 11: “Velocity – Valve Voltage curve – tensile direction”	79
Chart 12: “System behaviour with pre-control – valve closed in the beginning”	80
Chart 13: “PRESTO 420 – Validation Test 1 – Force – Time Chart”	83
Chart 14: “PRESTO 420 – Validation Test 1 – Force – Distance Chart”	83
Chart 15: “PRESTO 420 – Validation Test Crush Tube Comparison – Force – Time Chart”	84
Chart 16: “PRESTO 420 – Validation Test Crush Tube Comparison – Force – Distance Chart”	85
Chart 17: “PRESTO 420 – Validation Test Crush Tube Comparison – Distance – Time Chart”	86
Chart 18: “PRESTO 420 – Validation Test 7 – Force – Time Chart”	88
Chart 19: “PRESTO 420 – Validation Test 7 – Force – Distance Chart”	89
Chart 20: “PRESTO 420 – Validation Test Battery Comparison – Force – Time Chart”	90
Chart 21: “PRESTO 420 – Validation Test Battery Comparison – Force – Distance Chart”	91
Chart 22: “PRESTO 420 – Validation Test Battery Comparison – Distance – Time Chart”	92

15 List of Equations

Equation 1: "Control deviation $e(t)$ "	66
Equation 2: "Proportional feedback controller law (Böge, 2013)"	67
Equation 3: "Proportional coefficient (Böge, 2013)"	68
Equation 4: "Integral feedback controller law (Böge, 2013)"	69
Equation 5: "Integral feedback controller law for constant control deviations (Böge, 2013)"	69
Equation 6: "Proportional coefficient (Grote, et al., 2012)"	69
Equation 7: "Proportional integral feedback controller law (Grote, et al., 2012)"	70
Equation 8: "Proportional integral feedback controller law with reset time (Grote, et al., 2012)"	70
Equation 9: "Reset time proportional integral feedback controller"	70
Equation 10: "Movement velocity"	71
Equation 11: "Feedback controller law – filling level (Lunze, 2013)"	77
Equation 12: "Calculated average velocity – Validation test 1"	86
Equation 13: "Calculated average velocity – Validation test 2"	86
Equation 14: "Calculated average velocity – Validation test 4"	92
Equation 15: "Calculated average velocity – Validation test 5"	92
Equation 16: "Calculated average velocity – Validation test 6"	92
Equation 17: "Calculated average velocity – Validation test 7"	92

16 List of Abbreviations

ns	Nanoseconds
s	Seconds
min	Minutes
h	Hours
μm	Micrometre
mm	Millimetre
m	Metre
mm/s	Millimetres per second
N	Newton
kN	Kilonewton
bar	Bar
V	Volts
VAC	Volts alternating current
VDV	Volts direct current
A	Ampere
kB	Kilobyte
MB	Megabyte
GB	Gigabyte
kB/s	Kilobyte per second
S/s	Samples per second
kS/s	Kilo samples per second
MHz	Megahertz
Bit	Bit

Appendix

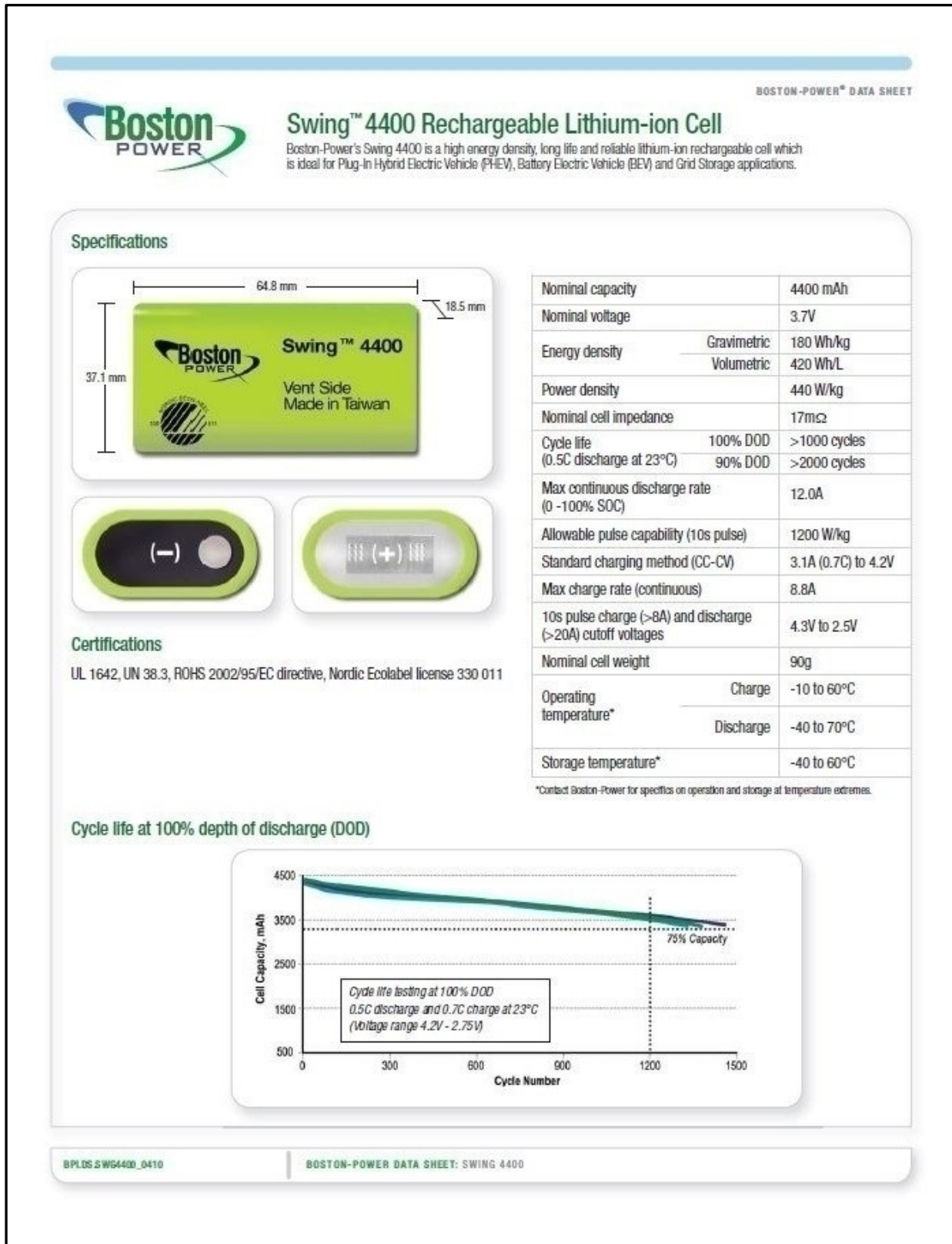


Figure 48: “Boston Power Swing™ 4400 Data Sheet ½ (Boston-Power, Inc., 2013)”

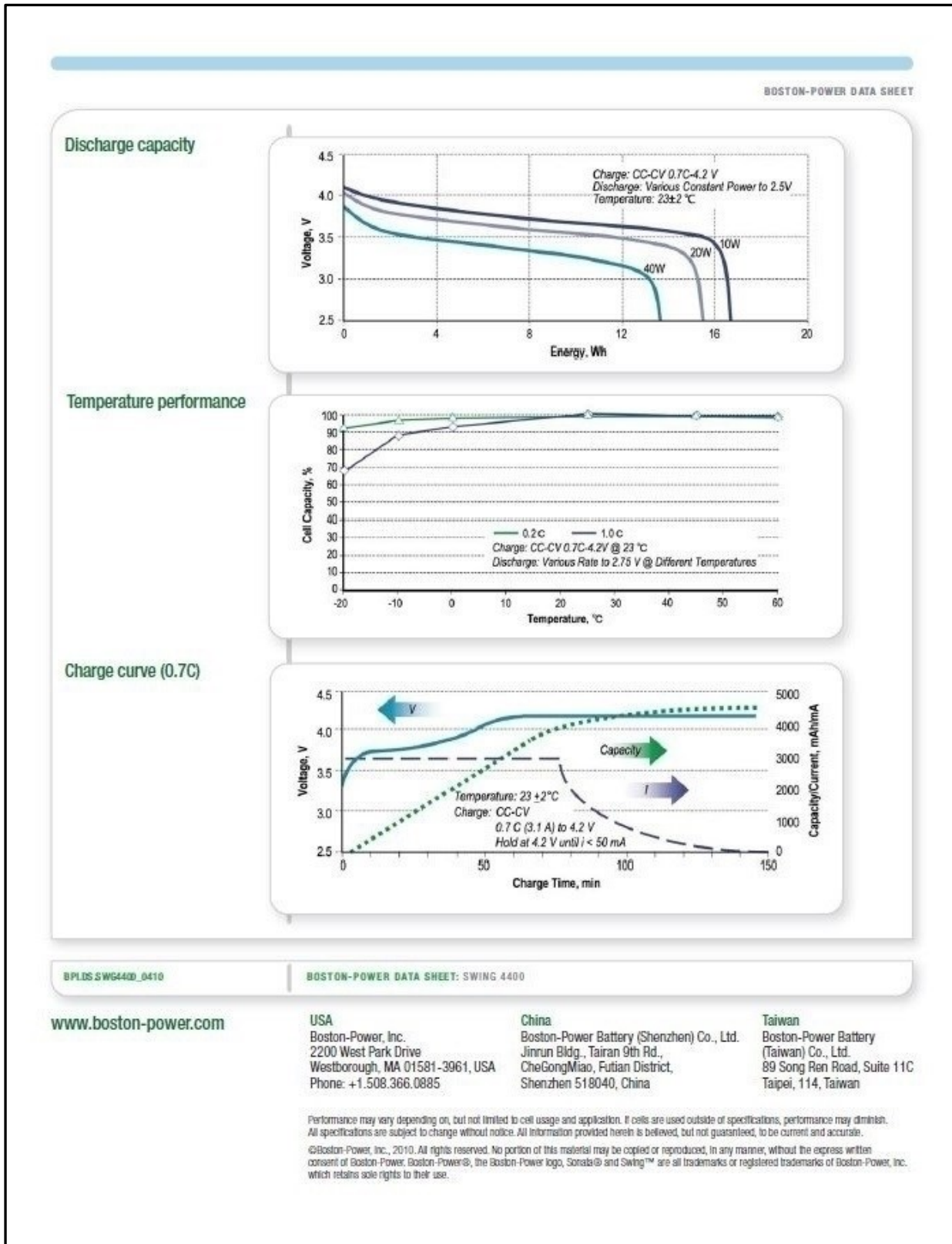


Figure 49: “Boston Power Swing™ 4400 Data Sheet 2/2 (Boston-Power, Inc., 2013)”