

Development and implementation of a control system for the primary target of the \bar{P} ANDA hypernuclei experiment

von

Christian Tiefenthaler

Bachelorarbeit in Physik
vorgelegt dem Fachbereich Physik, Mathematik und Informatik (FB 08)
der Johannes Gutenberg-Universität Mainz
am 05. April 2018

1. Gutachter: Prof. Dr. Josef Pochodzalla
2. Gutachter: Jun.-Prof. Dr. Florian Hug

Ich versichere, dass ich die Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt sowie Zitate kenntlich gemacht habe.

Mainz, den 05.04.2018

Christian Tiefenthaler
SPECF
Helmholtz-Institut Mainz
Staudingerweg 18
Johannes Gutenberg-Universität D-55099 Mainz
ctiefent@kph.uni-mainz.de

German Abstract

Zur Produktion von $\Lambda\Lambda$ -Hyperkernen im Rahmen des $\bar{\text{P}}\text{ANDA}$ Hyperkern-Experiments, wird ein zweistufiges Targetsystem eingesetzt. Das primäre Target besteht dabei aus einer dünnen Kohlenstofffaser. Das sekundäre Target, welches das primäre Target umgibt, besteht aus Absorberlagen und Siliziumstreifendetektoren.

Um zu verhindern, dass es zu langwierigen Reparaturarbeiten kommt, wenn ein Kohlenstofffilamenttarget reißt, werden mehrere solcher Targets an ausfahrbaren Halterungen auf einem Trägertisch montiert. Dieser entfernt das defekte Target, fährt dann ein ausgewähltes neues Target parallel zum Antiprotonenstrahl in eine Position, von der aus es senkrecht in den Strahl des $\bar{\text{P}}\text{ANDA}$ -Experiments eingebracht werden kann. Dieser Austausch geschieht mit Hilfe eines Kontroll- und Steuersystems. Eine weitere Aufgabe dieses Systems ist es, das Kohlefasertarget so im Strahl zu positionieren, dass die Reaktionsrate zwischen Target und Strahl auf eine optimale Reaktionsrate für die Produktion und den Nachweis der $\Lambda\Lambda$ -Hyperkerne ausgerichtet ist.

Unter Einsatz der im Rahmen dieser Arbeit entwickelten Steuerung werden parallel zum Strahl zwei Positionen, die exemplarisch für die Positionen zweier verwendeter Targethalter stehen, so genau wie möglich angefahren, um dann den Tausch des beschädigten Targets und das Einbringen des neuen Targets in den Strahl zu simulieren. Durch die Optimierung der Regelparameter wird die Genauigkeit der angefahrenen Positionen maximiert um dann durch genaue Messungen der Bewegungsabläufe festzustellen, inwieweit die Steueralgorithmen und der Versuchsaufbau die Anforderungen des $\bar{\text{P}}\text{ANDA}$ Hyperkern-Experiments erfüllen.

Dabei stellt sich heraus, dass die Target-Positionen mit einer Genauigkeit von $0,5\ \mu\text{m}$ angefahren werden können, was weit unter den Anforderungen von $100\ \mu\text{m}$ liegt. Auch konnte der Targetwechsel über 610 mal simuliert werden, was dem Austausch von 1220 Targets gleich kommt und weit über dem beabsichtigten maximalen Tausch von 5 Targets liegt.

Contents

German Abstract	iii
1. Preamble	1
2. Introduction	2
2.1. Hypernuclei	2
2.2. The \bar{P} ANDA hypernuclei experiment	2
2.3. The primary target	4
3. Setup of the primary target control system	6
3.1. Mechanical components	8
3.1.1. The PiezoWave motor and the corresponding PMWD10 driver	8
3.1.2. The HR1 ceramic motor and the corresponding AB1A driver box	9
3.1.3. The Mercury 1000V encoder	10
3.2. Control system	10
3.2.1. Software components and EPICS	11
3.2.1.1. IOCs, databases and records	11
3.2.1.2. The sequencer and the state notation language	12
3.2.1.3. The device support	13
3.2.2. Hardware components	13
3.2.3. The PID controller	14
4. Design of the primary target control	15
4.1. The EPICS database files	15
4.2. The state machines	16
5. Performance Test	19
5.1. First preliminary measurement	19
5.2. Optimizing the PID controller parameter setting	22
5.3. Long term driving test	23
5.3.1. Position accuracy	24
5.3.2. Mechanical stability over time	27
6. Summary and Outlook	30
A. Tables	37

1. Preamble

This bachelor thesis documents the development and improvement of a control system for the primary target of the $\overline{\text{P}}\text{ANDA}$ ¹ hypernuclei experiment as a part of the future research facility FAIR². The purpose of the experiment is the production of double Λ hypernuclei by a system of two targets. The primary target is used to produce Ξ^- particles. They decelerate and are captured in the secondary target that surrounds the primary target to produce the hypernuclei. To position the primary target correctly in the beam, it has to be led precisely through a 2mm gap in the secondary target's absorber material, and so a control system with two functions is required: The first is to keep the interaction rate between the primary target and the antiproton beam at a constant and detectable level of 4 MHz. The second function is to provide a possibility to exchange the target in case of a malfunction or a damaged target material. Both of these functions have to be reliable as well as precise, because the setup will be placed in the center of the $\overline{\text{P}}\text{ANDA}$ detector, where a quick maintenance access is not possible. To be sure the control system is working correctly, security features have to be implemented. This document is structured in the following manner: In chapter 2 the theoretical basics for the experiments will be presented; special attention will be paid to hypernuclei and their benefit to the $\overline{\text{P}}\text{ANDA}$ experiment. A description of the experimental setup follows in chapter 3, including the hardware parts and the software architecture. Chapter 4 contains the main part, in which the development of the control system's source code is documented. Chapter 5 includes the performance tests of the control system, both hardware and software, and their analyses are described. The conclusion in chapter 6 summarizes the acquired results and their value for the $\overline{\text{P}}\text{ANDA}$ experiment is estimated.

¹AntiProton ANnihilation at DArmstadt

²Facility for Antiproton and Ion Research in Darmstadt. It is a future extension to the already existing GSI Helmholtzzentrum für Schwerionenforschung.

2. Introduction

The spectroscopic study of strange nuclear systems is an important pillar of the $\overline{\text{P}}\text{ANDA}$ experiment at FAIR. The strangeness nuclear physics program aims at the high resolution gamma-spectroscopy of hyperatoms and hypernuclei. The purpose of the experiment is to increase the understanding of baryon-baryon interactions and as a consequence the solution of the so called "hyperon puzzle" in neutron stars [PAN16].

2.1. Hypernuclei

Stable nuclei consist of protons and neutrons. Both are built from up- and down-quarks and as fermions they have a spin of $\frac{1}{2}$. By replacing an up- or a down-quark of a nucleon by a strange-quark, another degree of freedom is added to the system in the form of a new quantum number: the strangeness. This can be realized by "implanting" a hyperon³ into the nucleus, which is then called a hypernucleus. The relevant hyperons in this experiment are Ξ^- and Λ . Their properties are shown in table 2.1.

Baryon	Spin Parity	Isospin	Strange- ness	Mass [GeV/c ²]	Mean Life- time [s]
Λ (uds)	1/2 ⁺	0	-1	1.116	$2.632 \cdot 10^{-10}$
Ξ^- (dss)	1/2 ⁺	1/2	-2	1.322	$1.639 \cdot 10^{-10}$

Table 2.1.: Properties of Λ and Ξ^- baryon (see [Ber+12]).

2.2. The $\overline{\text{P}}\text{ANDA}$ hypernuclei experiment

The $\overline{\text{P}}\text{ANDA}$ experiment as a part of FAIR examines the collision of antiprotons with a fixed target at beam momenta between 1.5 and 9 GeV/c. The antiprotons are delivered by the HESR⁴ with a maximum load of 10^{10} particles [PAN09b].

To produce double Λ hypernuclei, a system of two targets is built. The primary target, a carbon filament, is brought into the antiproton beam in order to collide antiprotons and carbon [Sau18]. The annihilation process of the antiprotons with the nucleons in

³A hyperon is a baryon (particle compounded of three quarks) including at least one strange quark and no heavy quark (bottom or charm quark).

⁴High Energy Storage Ring

2. Introduction

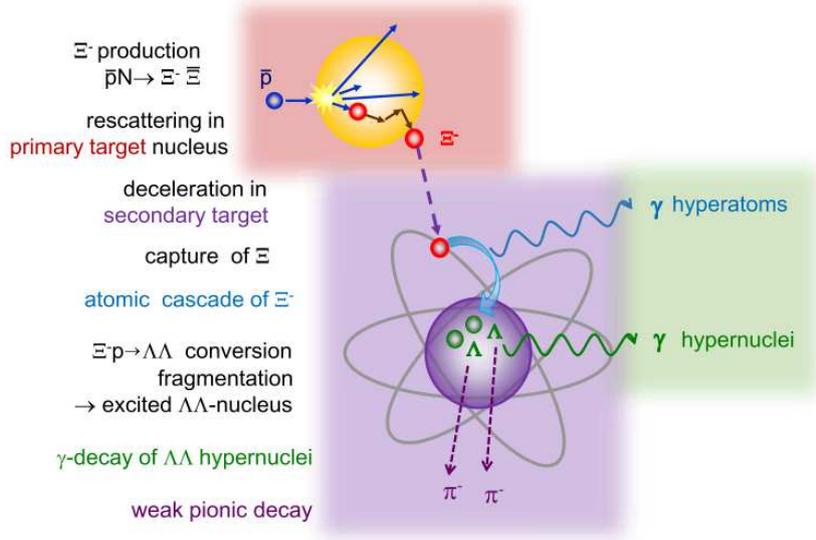
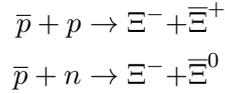


Figure 2.1.: Scheme of the production and decay cascade of the double Λ hypernuclei [PAN16]. The upper reaction $\bar{p}N \rightarrow \Xi^- \bar{\Xi}$ highlighted in red is the relevant process happening in the primary target [GHM16].

the target produces a slow Ξ^- and a $\bar{\Xi}$ particle as shown in the following reactions:



After the Ξ^- have been slowed down by scattering inside the primary target nuclei, they enter the secondary target, where they are slowed down further by alternating layers of silicon strip detectors and absorber material. If the Ξ^- particle gets stopped, it may be caught by an atom of the absorber material, cascade through the electron shell by emitting x-ray photons and react in the nucleus in the following way:



The created hypernuclear system is usually in an excited state, which it can leave by fragmentation [GHM16]. If both Λ particles are located in the same fragment, this fragment can de-excite into a particle-stable ground state by emitting one or more photons and finally decay weakly. The measurement of the weak decay products is sufficient to determine a specific hypernucleus. Charged pions from mesonic decays can be detected by the silicon detector. The whole process is summarized in figure 2.1.

Figure 2.2 presents the arrangement of the two targets in the hypernuclear setup. The secondary target, a layer structure of absorber material and silicon strip detectors, surrounds the primary target in its position inside the antiproton beam. The secondary

2. Introduction

target has to be placed as near as possible to the primary target in order to interact with the Ξ^- before they decay. [Sch17]

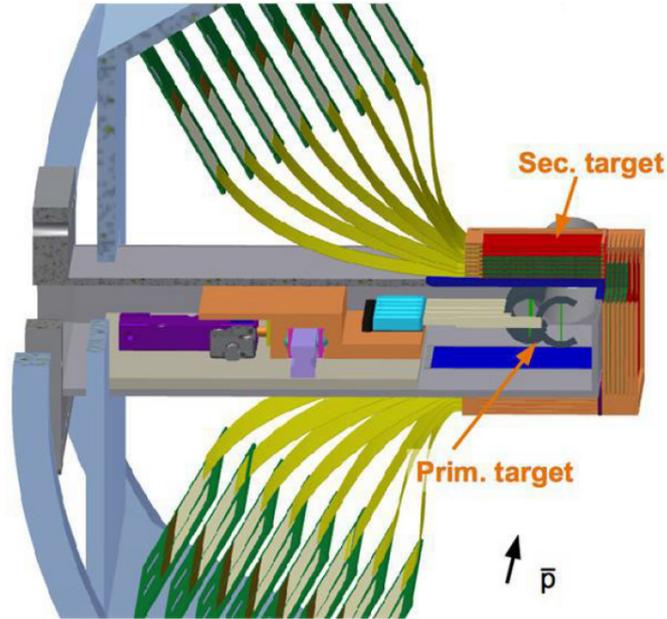


Figure 2.2.: CAD sketch of primary and secondary target in \bar{P} ANDA ([Roj16]), view in beam direction.

2.3. The primary target

The target chamber for the primary target is mounted to the HESR beam pipe in order to position the target inside the beam. A CAD sketch of the target chamber can be seen in figure 2.3. To ensure a relatively constant reaction rate between the beam and the target, it is necessary to be able to move the target perpendicular to the beam direction, as the luminosity of the beam decreases due to the antiproton - nucleon reactions. Once the target chamber is installed inside the \bar{P} ANDA spectrometer, one is unable to change or repair the target without dismounting the whole setup again. To prevent a downtime in case of a damaged or broken target, a stack of targets is mounted on a stage, so that a simple and rapid replacement can be guaranteed. The target carriage will be separated from the antiproton beam by a wall of absorber material with one gap of 2 mm width in it, just big enough for one target to get through into the beam. A safe passage through the gap without damaging the target demands a high precision of the control system, therefore it will be in the focus of the control system's development (chapter 4) and the test data analysis (chapter 5).

2. Introduction

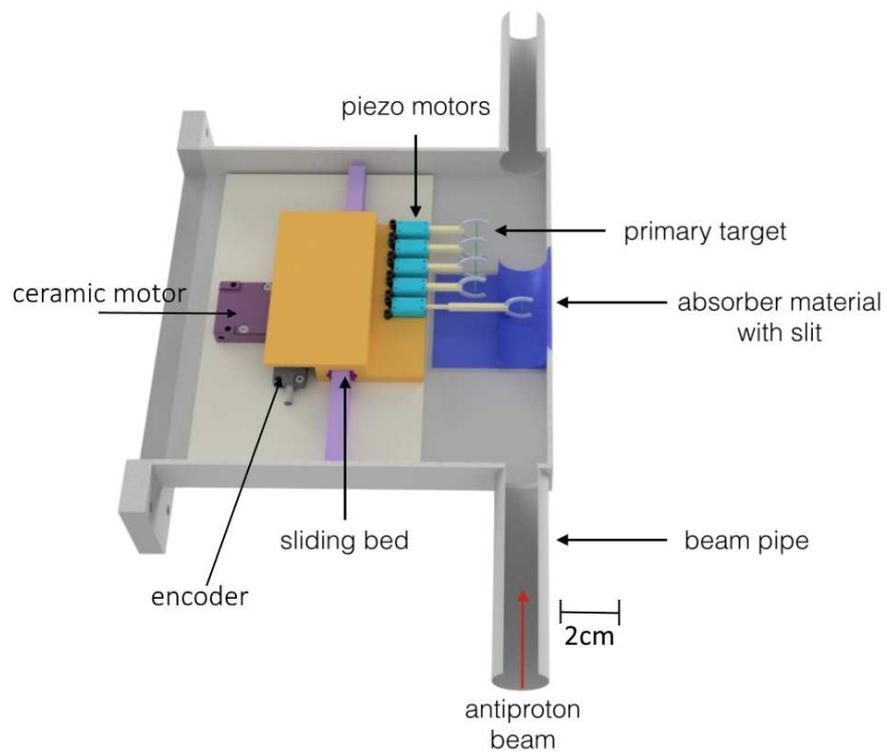


Figure 2.3.: CAD sketch of the primary target control system inside the target chamber (modified from [Ble]), nearly top view.

3. Setup of the primary target control system

The operating place for the target system as described above presents special challenges to the material and the electronics: The pressure inside the target chamber amounts to 10^{-9} mbar, so the used materials may not degas too much, otherwise the vacuum is impaired. Due to the control system's proximity to the target and the antiproton beam, the hardware components have to be radiation hard. Also, hardware parts have to work reliably in a magnetic field up to 2.0 T.

Additionally, the setup has to be small enough to fit into the target chamber: The height of the system is limited by the 2 cm diameter of the beam pipe, the radial length is limited to 15 cm by the subsequent detectors of the $\bar{\text{P}}\text{ANDA}$ spectrometer.

Under these conditions, the control system has to work very precisely: To pass the target through the gap in the beam pipe (cf. chapter 2.3), the position precision parallel to the beam direction has to be better than $100\ \mu\text{m}$ ⁵.

The following control system was built to fulfill these requirements. It consists of two branches. One is built from a BeagleBone Black, with 4 GPIO⁶ pins linked to two PiezoMotor[®] PiezoWave[®] Linear 0.1N PMWD10 driver boards with one PiezoMotor[®] PiezoWave[®] Linear 0.1N WL0104A-08A connected each. They are used for radial movements of the primary target perpendicular to the beam direction (below, this direction will be called x direction).

The other branch contains a Galil DMC 4113 motion controller board, connected to a Nanomotion Ltd. AB1A Driver/Amplifier. It is wired to the Nanomotion Ltd. HR1-1-U Motor. The Galil DMC 4113 motion controller board is also connected to the MicroE Systems[®] Mercury[™]1000V encoder. These components are used for motions parallel to the antiproton beam (in the following this direction will be called the z direction).

Both hardware parts, the Galil board and the BeagleBone Black are controlled by EPICS. The whole setup and its wiring is schematically summarized in figure 3.1.

⁵Estimating the required accuracy: The target frame with a thickness of 1 mm has to pass through a 2 mm gap. With a safety buffer, an accuracy of 0.1 mm is intended.

⁶General Purpose Input/Output

3. Setup of the primary target control system

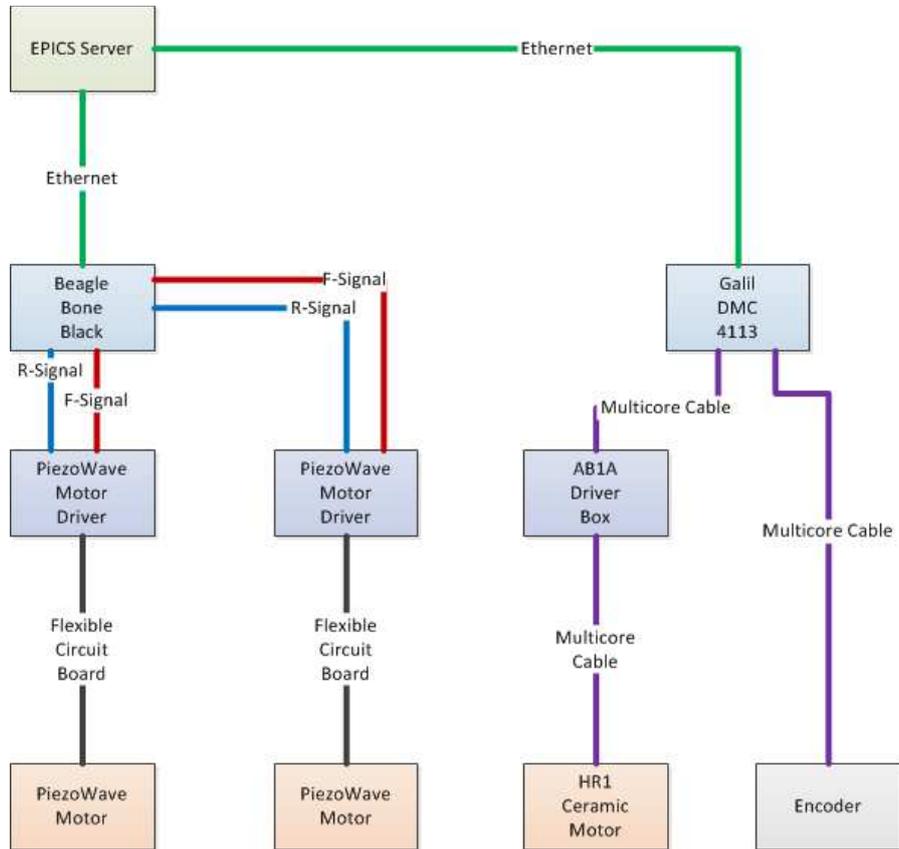


Figure 3.1.: Block diagram of the control system. The system separates into two branches, one for the x motions and one for the z motions. Both branches are connected to each other per ethernet cables via the EPICS server. The x motions branch contains the BeagleBone Black, that transmits driving direction signals (red "F" signals, blue "R" signals) to the PiezoWave motor drivers, that forward the information to the PiezoWave motors. On the other branch, the Galil controller board sends its driving information via a multicore cable to the AB1A driver box, that redirects the information to the HR1 motor, while the Mercury 1000V encoder sends position information to the Galil board, that adjusts the driving signal for the HR1 motor. To keep the scheme clear, the power supplies are excluded.

3. Setup of the primary target control system

3.1. Mechanical components

The mechanical part of this experimental setup consists of two different segments, one for motions in x direction and one for movements in z direction. The basic design of the z direction segment is based on [Roj16]. The setup includes an aluminium base plate with

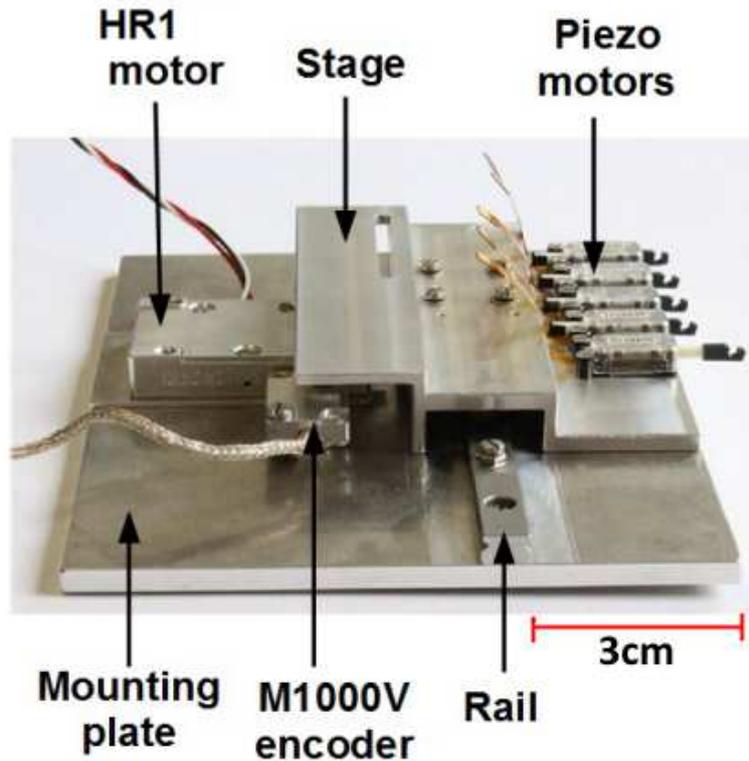


Figure 3.2.: The mechanical parts of the setup (from [Roj16]). On the mounting plate, the HR1 motor, the encoder and the rail are screwed on. The stage with the Piezo motors is mounted to a sledge, that sits on the rail. For the sake of clarity the drivers of the PiezoWave motors have been left out of the picture.

the HR1 ceramic motor for the movement in z direction, a rail and a carriage screwed on. On top of the skid, an aluminium stage is affixed. It contains the mounting points of the PiezoWave motors, responsible for motions in x direction. The glass scale of the encoder is glued on to determine the encoder position. The complete setup can be seen in figure 3.2

3.1.1. The PiezoWave motor and the corresponding PMWD10 driver

Motions of the target perpendicular to the beam direction are realized by the PiezoWave piezo motors. Each motor is connected to a PMWD10 driver board by a flexible circuit board, while the driver boards themselves are connected to the GPIO pins of the BeagleBone Black and a power supply (12V/100mA). At the end of each PiezoWave motor,

3. Setup of the primary target control system

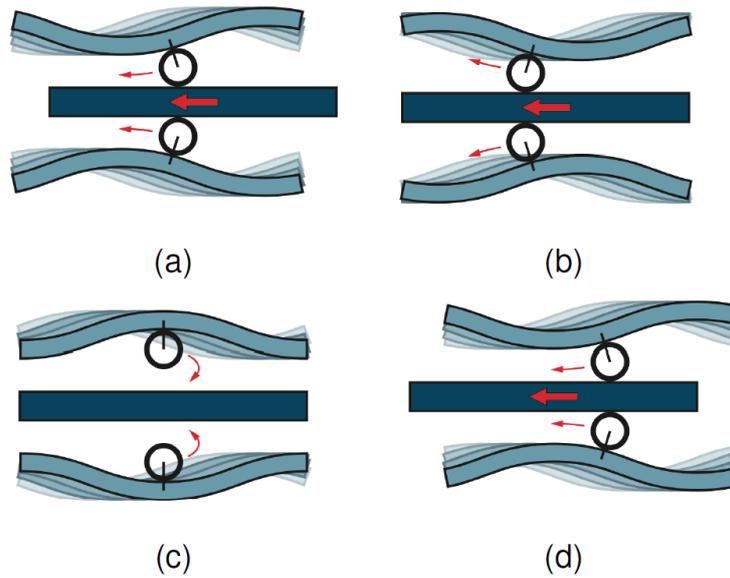


Figure 3.3.: Functional principle of the PiezoWave motor [Rau16a]. By applying a voltage to the piezoelectrical bars (light blue), they are deformed into a sine shape that pushes the driving rod (dark blue).

a frame containing the target can be attached. The PiezoWave motor is a small piezo motor that is designed to move a driving rod in axial direction. The functional principle is related to the HR1 motor (see 3.1.2). Two piezoelectrical bars are deformed into a sine shape by applying a voltage that was modulated by the driver board. By letting the sine wave move along the bar, the motion can be used to push a driving rod in the direction of the moving wave (see figure 3.3).

The PiezoWave motor gets its commands from the PWMD10 Driver, which is connected to the GPIOs of a BeagleBone Black (see [Rau16b]). The driver gets the signals to drive forward or backward from the GPIOs (3.3V) and needs a 12V power supply to convert the GPIO signals to an 8V/92kHz signal for each of the piezoelectrical parts (see [Pieb])⁷.

3.1.2. The HR1 ceramic motor and the corresponding AB1A driver box

Movements of the stage are realized by the ceramic motor HR1. It is bolt-on onto the base plate and pressed to the running surface. By applying voltage to the HR1, the motor pushes the mounting plate in direction of the rail.

The HR1 is an ultrasonic ceramic motor. Its functional principle is based on the inverted piezoelectric effect, where electrical fields, applied to a piezoelectrical material, are converted into a mechanical motion. This is realized by moving elementary dipoles

⁷A more detailed description of the PiezoWave motors can be found in [Rau16a].

3. Setup of the primary target control system

in the crystal by the electric force caused by the electric field.

In the HR1 motor, a ceramic rod is deformed by this effect in transversal and longitudinal direction, building up standing waves on the rod that lead to an elliptical motion of the tip. This tip is pressed onto a ceramic driving plate. At excitation frequencies much higher than the eigenfrequency of the setup, the driving plate is smoothly pushed forward by the tip.

The HR1 motor gets its high voltage of 250 to 290Vrms from its AB1A driver box that converts the analog signal from the Galil control board ($<10\text{V}$) into a high voltage signal. Motor and driver box are connected via a D-SUB 9 pin connector.

The complete setup for z movements is shown in figure 3.2.

3.1.3. The Mercury 1000V encoder

By moving the stage, the connected glass scale moves over the encoder, that reads out its actual position. In this way the analog linear encoder ascertains the z position of the stage in relation to a zero point, set at the moment the encoder is started. A microscope

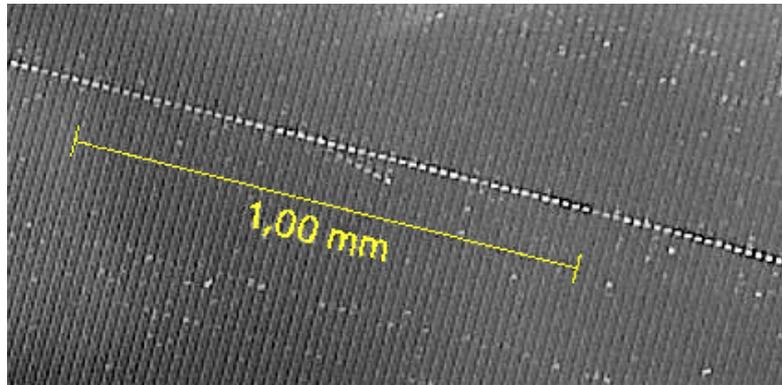


Figure 3.4.: Microscope picture of the encoder's glass scale at a magnitude of 200. One can see the periodically arranged increments that are scanned by the encoder to determine their relative position.

picture of the glass scale can be seen in figure 3.4. The increments are very narrow to create a high resolution of the position. By counting the increments, one can estimate the resolution to $20\ \mu\text{m}/\text{increment}$. By using the encoder's interpolating electronics, the resolution increases up to $0.078\ \mu\text{m}/\text{increment}$ (see [Mic]).

3.2. Control system

The control system can be divided into two sections: software and components. Both are explained below.

3. Setup of the primary target control system

3.2.1. Software components and EPICS

One of the requirements of the control system was the usage of EPICS⁸. EPICS is a software environment used to build and operate distributed control systems and will be used by the HESR and the PANDA experiment.

3.2.1.1. IOCs, databases and records

EPICS consists of at least one IOC⁹. Multiple IOCs are able to communicate with each other via EPICS' Channel Access network protocol. Each IOC can provide data to the network and read data provided from other IOCs. In the setup of the primary target control, one IOC runs on the BeagleBone Black to control the PiezoWave motors, the other one runs on a PC, connected to the Galil board to control the HR1 motor and read the encoder position.

Each IOC contains at least one database and the device support. The database is made of at least one record with multiple fields. The combination of a record and its value is called PV¹⁰. Records are either used for input and output (get or put data from respectively to the hardware or other records, by addressing to the other records PV¹¹), or for algorithms and calculations. The following example (3.1) shows the record PANDAHYP:BBB2:PSWITCHP of type bi, which means binary input. The record is built of 5 fields with specific purposes. For example, the field SCAN scans for a new value every 0.1 seconds, the field DTYP is set to devgpio¹² to allow the record to scan and manipulate the status of the BeagleBone GPIOs.

```
1 record( bi, "PANDAHYP:BBB2:PSWITCHP" ) {
2   field( DTYP, "devgpio" )
3   field( INP, "@P8_26 H " )
4   field( ZNAM, "off" )
5   field( ONAM, "on" )
6   field( SCAN, ".1 second" )
7 }
```

Code snippet 3.1: Example of a record

By using pattern files, one can replace wildcards in records with predefined values, to shorten the database and keep it clear.

⁸Experimental Physics and Industrial Control System. The EPICS version for BeagleBone Black from <https://panda-service.gsi.de/repo/> was installed.

⁹Input-Output-Controllers

¹⁰Process Variable

¹¹PV names have to be unique in the network, in order to assure this distinctive addressing

¹²From <https://github.com/ffeldbauer/epics-devgpio>, written by Florian Feldbauer.

3. Setup of the primary target control system

```
1 { pattern{DIRECTION ,NUMBER ,PIN}
2   {F,1,P9_13}
3   {R,1,P9_14}
4   {F,2,P9_15}
5   {R,2,P9_16}
6 }
```

Code snippet 3.2: Example of a pattern

The code snippet 3.2 contains the substitutions for `DIRECTION`, `NUMBER` and `PIN`, that are used in the record in code snippet 3.3. This way, one record is used to set four different GPIOs.

```
1 record( bo, "PANDAHYP:BBB2:PM${DIRECTION}${NUMBER}" ) {
2   field( DTYP, "devgpio" )
3   field( OUT, "@${PIN} H " )
4   field( ZNAM, "off" )
5   field( ONAM, "on" )
6 }
```

Code snippet 3.3: Example of a record with wildcards for pattern substitution

3.2.1.2. The sequencer and the state notation language

The sequencer is used to run SNL¹³ programs, which is used to program finite state machines. In EPICS they are called State Sets (ss). All defined state sets in the IOC run simultaneously.

The state machines consist of different states that are all built in the same way:

- At first, it is possible to initialize the local variables.
- At second, the state has an optional `ENTRY` sequence, that contains code that is processed when entering the state.
- Afterwards, as many `WHEN` sequences as needed are attached. Every `WHEN` part begins with an expression that has to be fulfilled to enter an optional sequence of code to be processed, before jumping to another state within this state set.
- Every state ends with an optional `EXIT` part, where code is placed that has to be executed when leaving the state.

Altogether, these syntax components can be seen in code snippet 3.4.

¹³State Notation Language. From <http://www-csr.bessy.de/control/SoftDist/sequencer/>

3. Setup of the primary target control system

```
1 /* ===== variables ===== */
2   int zbewstatus; /* motion status of the z motor */
3   assign zbewstatus to "galil01:A:SPMG"; /* (0:"Stop", 3:"Go")
   */
4 /* ===== state machines ===== */
5 ss testmachine {
6   state drivez {
7     entry {
8       /* starting z motion */
9       zbewstatus = 3;
10      pvPut(zbewstatus);
11    }
12    when (delay(0.1)) {
13      /* stopping z motion */
14      zbewstatus = 0;
15      pvPut(zbewstatus);
16    } state wait
17  }
18  state wait {
19    when (delay(1.0)) {
20    } state drivez
21  }
22 }
```

Code snippet 3.4: Example of a state machine code

3.2.1.3. The device support

The device support connects and controls hardware devices, that are used for input and output (like gauges, motors, etc). Generally, the device support (comparable to a driver) has to be programmed. For this thesis, the driver for the PiezoWave motors were already written in the context of [Rau16a] and implemented. For ceramic motors, Galil provides a driver that had to be adjusted to the HR1 motor.

3.2.2. Hardware components

The HR1 ceramic motor is controlled by the Galil DMC 4113 control board, as the BeagleBone Black controls the PiezoWave motors.

BeagleBone Black

The BeagleBone Black is a small single board computer that can be used for prototyping and development. It comes with a 1 GHz CPU and 512 MB RAM. It also has two headers with 46 pins each, including several GPIO pins. In this thesis, EPICS (see 3.2.1) on a Linux OS is used to control the GPIO pins.

3. Setup of the primary target control system

Galil DMC 4113

The Galil DMC 4113 is a motion controller board for one motion axis. It is equipped with ethernet, a RS232 port and an USB port. Additionally, the controller board can handle encoders with a frequency up to 15 MHz. By updating the firmware, it is possible to achieve compatibility with the HR1 ceramic motor. The z motor can be controlled by uploading self-written scripts to the Galil board in a simple scripting language. However, it is also possible to control the DMC 4113 via EPICS.

3.2.3. The PID controller

The accuracy of the Galil DMC 4113 motion controller is based on the settings of its internal PID controller. It compares the actual motor position to the estimated position and minimizes the difference between both values [Lun10].

PID controllers are made of three elements: a proportional part, an integral and a differential part. Explained in a mathematical way, the controller's transfer function $u(t)$ is calculated as the sum of the three controller terms $P(t)$, $I(t)$ and $D(t)$:

$u(t) = K_p \cdot e(t) + K_i \cdot \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt}$. The error function $e(t) = s - p(t)$ is defined, with s as the estimated setpoint and $p(t)$ the process signal.

The proportional term $P(t)$

The transfer function of the proportional part is defined as $P(t) = K_p \cdot e(t)$. A high value of K_p means a strong reaction of $P(t)$, which contains the risk of an overshoot.

The integral term $I(t)$

The integral's transfer function is defined as $I(t) = K_i \cdot \int_0^t e(\tau) d\tau$. The integral part lets the controller minimize $e(t)$ faster than the single proportional part, but increases the risk of an overshoot.

The differential term $D(t)$

The transfer function of the differential part is defined as $D(t) = K_d \frac{de(t)}{dt}$. By differentiating the error function, the velocity of the change of the deviation of the process signal from the setpoint is calculated. The faster the error function changes, the higher $D(t)$ gets. This flattens $u(t)$ as the setpoint is reached, thus the overshooting of an PI regulator is compensated. By adding the factor K_d , the influence of $D(t)$ to the process signal can be regulated.

By varying the controller parameters, the accuracy of the z motor changes. This influence can be seen in chapter 5.1.

4. Design of the primary target control

The purpose of the control system is to allow the user to change the position of the target stack between predefined encoder positions to change the targets. At each of these predefined positions it is possible to move the target on the related x motor through a gap in the beam pipe wall into the particle beam. All other x motors must not move their driving rods when not in position. The z motor is not allowed to run, if one of the x motors has not fully retracted its target. These security steps are configured in the state machines. But first of all, it is necessary to declare and define the process variables and the assigned pattern in the EPICS database files.

4.1. The EPICS database files

Each connected EPICS IOC provides the defined PVs to all other IOCs in the network. Therefore it is required to gather them up in a database file, where the PVs get their properties. A sample database code of the records for the motor's driving directions `PANDAHYP:BBB2:M${NUMBER}${DIRECTION}` and their extender positions `PANDAHYP:BBB2:M${NUMBER}POS` is shown in code snippet 4.1.

```
1 record( bo, "PANDAHYP:BBB2:M${NUMBER}${DIRECTION}" ) {
2     field( DTYP, "devgpio" )
3     field( OUT, "@${PIN} H " )
4     field( ZNAM, "off" )
5     field( ONAM, "on" )
6 }
7
8 record( longout, "PANDAHYP:BBB2:M${NUMBER}POS" ) {
9     field( VAL, "" )
10 }
```

Code snippet 4.1: Example for a record

Herein, the pattern of code snippet 3.2 was used. It is also possible to deposit a startup script in the Galil IOC, written in the Galil scripting language, to be executed at the start of EPICS. Herein the basic properties of the HR1 motor are set, like PID controller parameters, offset voltages and the encoder mode. The initialization of the ceramic motor is also executed here. The startup script is shown in 4.2.

4. Design of the primary target control

```
1 'Galil Motion Control epics@galilmc.com http://www.galilmc.com/  
2 #epics  
3 'Insert code to run on GalilStartController() here...  
4 'include any motor initialization, etc.  
5 SH 'servo here!  
6 WT 100 'delay for 100ms  
7 CE 2 'initializing the encoder  
8 ZP 8 'setting the motors positive offset voltage  
9 ZN 8.7 'setting the motors negative offset voltage  
10 DP*=0  
11 KP 500 'setting the PID control parameters  
12 KD 1000  
13 KI 25  
14 EN
```

Code snippet 4.2: Galil startup script

4.2. The state machines

The SNL code is structured in the following way:

At first, the variables are defined and declared. An overview of all used variables and their meaning to the code is shown in table A.1. One has to differentiate between variables only used in the SNL code and variables that are assigned with PVs. The latter are needed to let the SNL communicate with the EPICS interface.

Below, the state machine code is described in detail¹⁴.

The state machine drive

The task of the state machine `drive` is to realize all motions in x and z directions, including error management and position determination. It bases on 8 different states. The characteristic points that are used to describe the z motions determined by the state machine, are labelled in the following manner:

- A: Start point of the z motion
- C: First resting point
- E: Second resting point
- G: End point of the z motion

The missing letters B, D, F and H are used to label the motions inbetween two points. A schematic view of the whole state machine sequence can be seen in figure 4.1.

- On every start of the control system, the states `init` and `initz` are passed through. The variables are initialized, x motors are retracted and the z motor is moved to

¹⁴The entire SNL code can be seen on the CD appended to this thesis.

4. Design of the primary target control

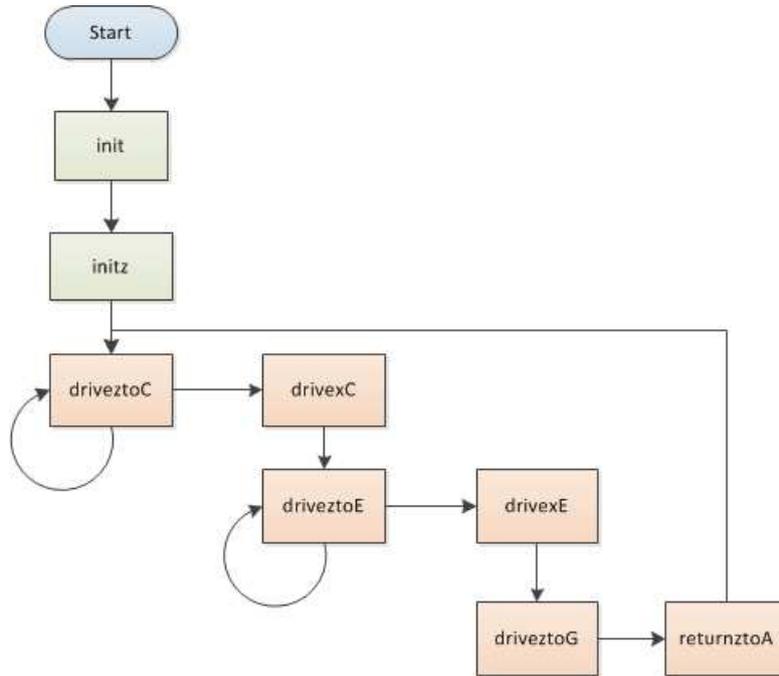


Figure 4.1.: Schematic view of the state machine `drive`. After initializing the variables and moving all the motors to their starting points in the states `init` and `initz`, the z motor drives to its first resting point (`driveztoC`), where the first x motor simulates a changing in the first target's position (`drivexC`). Following this, the z motor drives to its second resting point (`driveztoE`), where the second x motor moves its target (`drivexE`). Subsequently, the z motor drives towards its end point in state `driveztoG`, before the target stack is returned to its starting point (`returnztoA`).

its starting point by sending it to $z = -10000$ encoder units¹⁵. The state machine stays in the state `initz` until the z motor pushes the switch on the starting point, sending a signal to the BeagleBone Black or, due to mechanical issues (see 5.3.2), waits for 10 seconds of timeout. Now, the actual position is saved as the position offset value to define a zero point for the following movements and the state machine switches to state `driveztoC`.

- In the state `driveztoC`, the z motor is sent to its first resting point at $z = 500$ encoder units, where the first x motor is allowed to work. As long as the exact position of the first resting point is not reached, the state machine retains in `driveztoC`¹⁶. When the intended position is reached, the state machine switches to the state `drivexC`.

¹⁵The thrust of the z motor depends on the distance it is about to drive. In order not to get the motor stopped by the mechanical problems, the motor is sent to positions far away and stopped by hitting the switches on the start and the end point.

¹⁶Due to the high position accuracy determined by the PID controller parameters, it is not necessary to add a position tolerance bandwidth to the resting points positions (see 5.2).

4. Design of the primary target control

- The motions of the first x motor are determined by the state **drivexC**. The leg of the x motor is extended in one step and driven back in by 25 small steps to mimic the later on required precision of the target positioning. After the x motion ended, the state machine transits to the state **driveztoE**.
- In the state **driveztoE**, the z motor is sent to the second resting point at $z = 1000$ encoder units. Here the second x motor is able to work. Analogous to state **driveztoC**, the state machine retains in **drivetoeE** as long as the exact position of the first resting point is not reached. When the intended position is reached, the state machine switches to the state **drivexE**.
- State **drivexE** manages the motions of the second x motor. The leg of the x motor is moved out of its position in one step and driven back in by 25 small steps. After that, the state machine may transit to state **driveztoG**.
- State **driveztoG** is used to push the target stack in z direction towards the motions end point by sending it to $z = +3000$ encoder units. By pushing the switch on the end point (or alternatively keeping stuck for 10 seconds) the state machine switches into state **returnztoA**.
- The state **returnztoA** is used to drive all x motor legs back in again, so that the z motor is then allowed to drive back to the start point by sending it to $z = -10000$ encoder units. The state machine keeps in this state until it hits the switch on the start point and transits to state **driveztoC**.

The state machine mylogfile

The job of this state machine is to create and write a logfile with the parameters of the motions. These data are used for the analysis of the precision and stability of the movement.

5. Performance Test

5.1. First preliminary measurement

In a first driving test, the HR1 motor was connected to the provided driver box and a small joystick. With this setup, the behavior of the combination of the motor and the mounting plate during motion was tested. It came out that the motion was not constant over the full length of the rail.

After connecting the HR1 motor, the encoder and the driver box with the Galil DMC4113 board and some calibration work, it was detected that the full possible driving distance was about 1200 encoder units (approx. 6 mm), measured from the far left end stop. To improve the setup, a special firmware to control ceramic motors was written and uploaded on DMC4113.

In a first measurement, the accuracy of the driven distance of the motor was ascertained. Therefore, the Galil DMC4113 board was equipped with a small software program to drive the motor automatically for different distances with increasing length¹⁷. The code is structured in the following manner:

1. The variables are initialized, the hardware parameters are set (see 5.1) and written to the logfile. The motor is driven to the starting point.
2. The driving distance is set to 100 encoder points and the driving sequence starts:
 - a) The motor drives for the set driving distance, stops and returns.
 - b) The driving distance is increased by 100 encoder points.
 - c) The following parameters are written to the logfile:
 - number of the run
 - theoretical start point
 - deviation from the theoretical start point (real start point)
 - theoretical end point
 - deviation from the theoretical end point (real end point)
 - d) This process is repeated until the driving distance equals the maximal possible driving distance.

¹⁷The program code can be found on the CD appended to this thesis.

5. Performance Test

3. In case the motor is not moving for 500ms, it has to be assumed, that a mechanical error has happened during the motion process and the error handling routine is started:
 - a) The motor speed, the acceleration and the deceleration are set to their maximum values.
 - b) The motor jogs as fast as possible ten times for 500 encoder points to both directions.
 - c) The motor speed, the acceleration and the deceleration are set back to their normal values.
 - d) The program returns to the driving sequence.

Hardware Parameters		
Parameter name	Parameter value	Comment
SP	5000	Motor speed
AC	256000	Acceleration
DC	256000	Deceleration
CE	2	Encoder polarity
ZP	5	Positive motor offset voltage
ZN	-6.5	Negative motor offset voltage
KP	500	Proportional parameter for the PID controller
KD	1000	Differential parameter for the PID controller
KI	25	Integral parameter for the PID controller
TW	500	Time period of inactivity in ms, until the controller enters the error handling routine

Table 5.1.: Hardware parameters of the Galil board, set as default for the measurement in chapter 5.1

Each run contains about 250 motions, starting at position -100 to a maximum position of -1100 encoder units. After each run, the control parameters of the PID controller of the DMC4113, the bias voltages of the ceramic motor and the motion parameters (speed, acceleration and deceleration) were varied.

Table 5.2 displays the results. Herein d is the mean difference between the target distance and the real driven distance, Δd is its standard deviation, both measured in encoder units. SP, AC and DC are the values for driving speed, acceleration and deceleration of the motor. ZP and ZN are the bias voltages of the motor for motion in positive and negative directions. To keep the motion steady, a maximum motion distance of 1100 encoder units was set. At higher distances, the mechanical issues like canting of the

5. Performance Test

sledge and irregular velocities became noticeable(see section 5.3.2).

Table 5.2 shows, how the position error become smaller by adjustment of the parame-

Run	$d \pm \Delta d$ [encoder units]	SP	AC	DC	KP	KD	KI	ZP [V]	ZN [V]
1	-0.5619 ± 0.7998	5000	25600	25600	100	4000	40	5	-7.500
2	-0.7285 ± 0.7811	5000	25600	25600	1023	4095	255	5	-7.500
3	-0.7857 ± 0.7100	5000	25600	25600	500	2000	120	5	-7.500
4	-0.8500 ± 0.8773	5000	25600	25600	250	1000	60	5	-7.500
5	-0.6619 ± 0.8036	5000	25600	25600	125	500	30	5	-7.500
6	-0.1229 ± 7.7910	5000	25600	25600	50	100	20	5	-7.500
7	-1.5142 ± 1.8043	50000	25600	25600	500	1000	100	5	-6.499
10	-1.5398 ± 1.4564	50000	256000	256000	500	1000	100	5	-6.499
11	-1.5132 ± 1.5831	5000	256000	256000	500	1000	100	5	-6.499
12	-0.2055 ± 0.4052	5000	256000	256000	500	1000	25	5	-6.499
13	-0.1658 ± 0.3728	5000	256000	256000	500	1000	25	5	-6.499

Table 5.2.: Accuracy of the motor motion, depending on the variation of the used parameters

ters. The usage of the PID control causes the negative values of d as it tries to prevent the motor to drive further than the intended distance. Run 13 was chosen as the most accurate.

To calculate the conversion factor between a metric scale and the encoder unit scale, the motor was driven to distances from 1000 encoder units down to 600 encoder units in steps of 100 encoder units. For each of these points the driven length was measured with a calliper. The results of these measurements are shown in table 5.3. Herein D is the driven distance in encoder values, its error ΔD is taken from run 13 from table 5.2. L represents the measured length of the distance in cm. Its error ΔL was assumed to be 0.02 cm. With these values, the conversion factor F between encoder units and metric scale and its error ΔF can be calculated¹⁸. The unit of F is $\mu\text{m}/\text{encoder unit}$. The

Distance D [encoder units]	Length L [cm]	Conversion factor F [$\mu\text{m}/\text{encoder unit}$]
1000 ± 0.1658	0.52 ± 0.02	5.20 ± 0.20
900 ± 0.1658	0.46 ± 0.02	5.11 ± 0.22
800 ± 0.1658	0.42 ± 0.02	5.25 ± 0.25
700 ± 0.1658	0.37 ± 0.02	5.28 ± 0.28
600 ± 0.1658	0.33 ± 0.02	5.50 ± 0.33

Table 5.3.: Calculating the conversion factor between the encoder scale and a metric scale

arithmetic mean value \hat{F} of all values of F is calculated as:

$$\hat{F} = (5.27 \pm 0.26) \mu\text{m}/\text{encoderunit}$$

¹⁸Unless otherwise stated, uncertainties are calculated via gaussian error propagation.

5.2. Optimizing the PID controller parameter setting

After the first driving test, EPICS was included and therefore a \bar{P} ANDA-viable slow control system was built. In a second driving test, the Galil board was controlled via EPICS. A state machine code was installed to drive the z motor from the start point for a distance of

$z = 500$ encoder units. Using the the PID controller parameters from section 5.1, it was not possible to drive closer than 2 encoder units to the intended position. By further varying the PID controller parameters, the z motor was able to drive correctly to $z = 500$ encoder units. The best results were achieved with the combination $KP = 1000$, $KI = 255$, $KD = 25$. Table 5.4 summarizes the results for all tested PID parameters. A plot of these results is shown in figure 5.1: Besides the deviation of the end point, the shape of the dynamics shows a strong dependence on the parameter settings. Generally one can see that a small integral part leads to long driving times. On the other hand, a small differential part smoothens the motion, but lets the carriage decelerate earlier the bigger it gets. The proportional part mainly influences the mean velocity of the carriage.

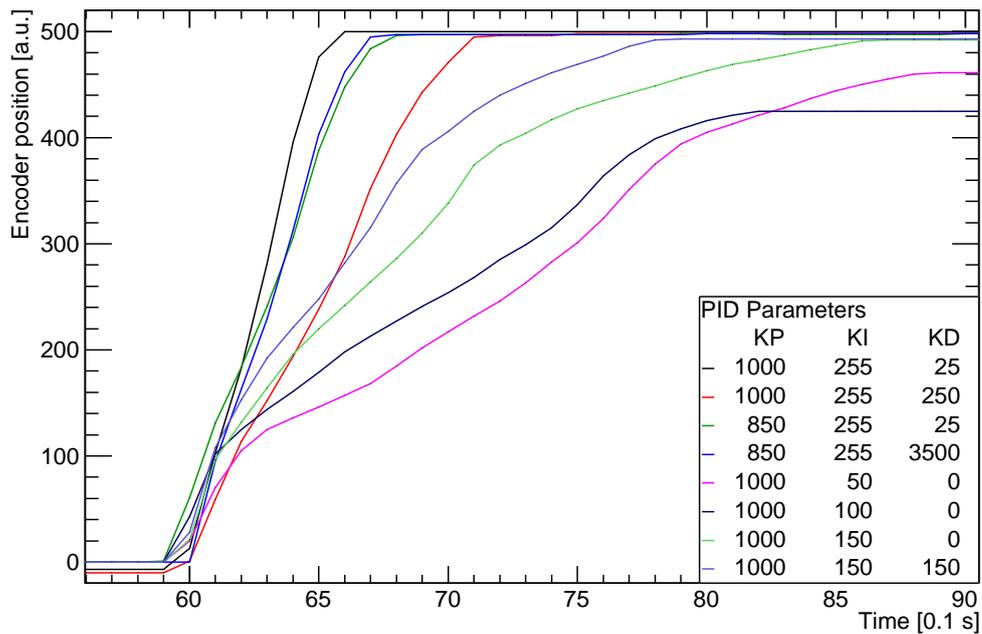


Figure 5.1.: Comparison of the carriage's dynamics for eight different sets of PID control parameters. The highest position accuracy was achieved with the parameter combination $KP = 1000$, $KI = 255$, $KD = 25$.

5. Performance Test

KP	KI	KD	Nearest approach [encoder units]
1000	50	0	-39
1000	100	0	-3
1000	150	0	-8
1000	150	150	-7
1000	255	25	0
1000	255	250	-2
850	255	25	-2
850	255	3500	-2

Table 5.4.: Changing of the position accuracy depending on the PID controller parameters. For 8 sets of parameters the z motor was driven from its starting point for 500 encoder units. Depending on the parameters the point of nearest approach to $z = 500$ encoder units changed. The combination $KP = 1000$, $KI = 255$, $KD = 25$ provides the best result in hitting the exact estimated position. Also the duration for driving the distance depends on the combination of parameters.

5.3. Long term driving test

After the optimization of the regulation parameters, a long term driving test with EPICS has been performed, where the Galil DMC 4113 board and the BeagleBone Black were controlled by an SNL code on the EPICS server. The driving sequence described in section 5.3.1 was repeated for 612 times, before the motion stopped due to mechanical problems (in section 5.3.2 these problems will be displayed in detail). On two predefined positions, $z = 500$ encoder units and $z = 1000$ encoder units away from the start point, the z motion was stopped and the x motors were driven out and back in again. After the x motions, the z motor continued its movement, until the positive end stop was reached. That made a total driving distance of (13.65 ± 0.05) mm. Finally, the z motor returned to the negative end stop. Also, every 0.1 seconds, a logfile entry was written, containing the following information:

- the entry number
- the estimated position of the z motor
- the real position of the x motor, provided by the encoder
- the number of the x motor that is extended or retracted
- the status of the two pressure switches

Assuming the logfile entries were written with a constant time shift of 0.1 seconds, the entry number was used as a time signal.

5.3.1. Position accuracy

In figure 5.2, a path-time diagram of one typical driving sequence is shown. It is separated into 8 sections¹⁹.

- A: The z motor rests at its starting point at $z = 0$ encoder units.
- B: The z motor drives to its first resting point at $z = 500$ encoder units.
- C: The z motor stops its motion and the first x motor is allowed to move.
- D: After the motion of the first x motor is completed, the z motor drives to the second resting point at $z = 1000$ encoder units.
- E: At the second resting point, the z motor stops its motion again. Now the second x motor is allowed to move.
- F: After the motion of the second x motor is completed, the z motor is moving towards the end of the rail at $z = 2613$ encoder units.
- G: The z motor reaches its end point and hits the switch at the end of the rail.
- H: The z motor returns to its starting point at $z = 0$ encoder units.

Adding up the position values of all 612 runs reveal a high position precision for the relevant resting points C and E, the start point A and the stop points G (see figure 5.3, the numerical values are summarized in table 5.5). The achieved precision of the two

point	estim. position [encoder units]	real position [encoder units]
A	0.000	0.569 ± 1.071
C	500.000	500.000 ± 0.012
E	1000.000	1000.000 ± 0.080
G	2613.000	2609.000 ± 3.233

Table 5.5.: Estimated and measured positions of the points A, C, E and G.

most important positions, C and E, is

$$C = (0.000 \pm 0.012) \text{ encoder units}$$

$$E = (0.060 \pm 0.080) \text{ encoder units}$$

The values of E are the mean and standard deviation values of its histogram in figure 5.3, provided by Root²⁰, as for the calculation of the values of C its histogram shows a uniform distribution in only a single bin with a mean of 0, a bin width of 1 and a height

¹⁹For reasons of clarity, the measured position values are normalized to 0.

²⁰root.cern.ch

5. Performance Test

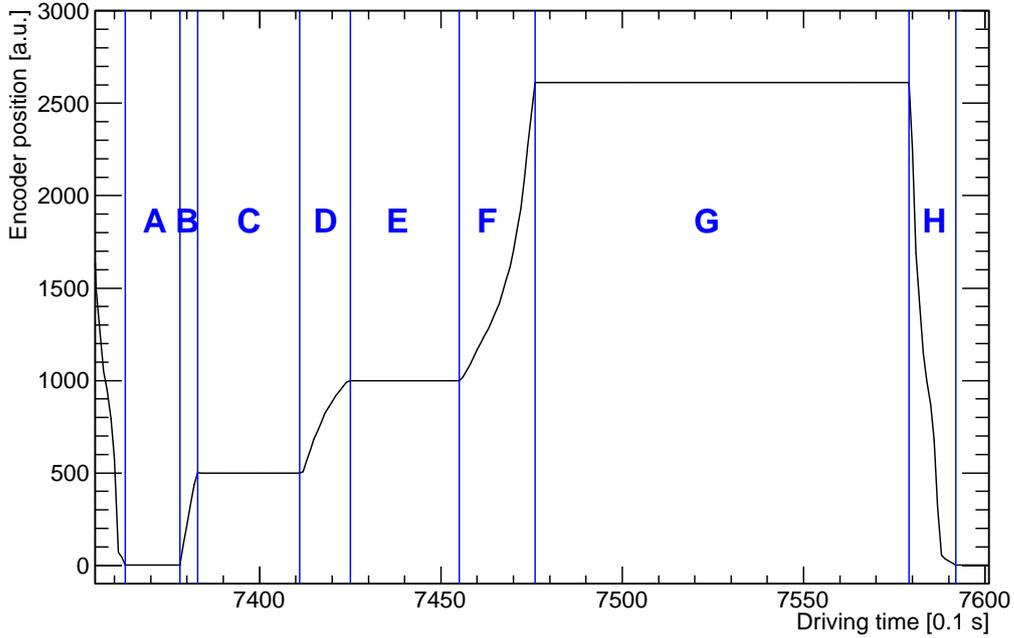


Figure 5.2.: Position distribution during one typical driving sequence of the z motor. The motor starts on a low position level (A), drives to its first resting point (B), stops there for the first x motor to move (C), drives further to the second resting point (D), stays there for the second x motor to move (E), drives towards the end point (F), stays there (G) and returns to its starting point (H).

of $N=612$ runs. The standard deviation is calculated via $\sqrt{(\text{bin width})^2/12} \cdot \sqrt{1/(N)}$. With table 5.5 the mean distance $A \rightarrow G$ can be calculated: $D_{\text{EPICS}} = 2608.431 \pm 4.304$ encoderunits. Measured with a calliper, the same distance is (13.65 ± 0.05) mm in a metric scale, so the conversion factor is $F_{\text{EPICS}} = (5.23 \pm 0.02)$ $\mu\text{m}/\text{encoder unit}$. This is in line with $\hat{F} = (5.27 \pm 0.26)$ $\mu\text{m}/\text{encoder unit}$ from section 5.1. The conversion factor allows one to determine the position accuracy of the resting points.

$$C = (0.000 \pm 0.063)\mu\text{m}$$

$$E = (0.314 \pm 0.418)\mu\text{m}$$

One can compare the precision $\Delta z < 0.5\mu\text{m}$ calculated here to the error of the encoder. It amounts to $\pm 3\mu\text{m}$ [Mic]. Over 612 runs, it lowers to $\pm 3\mu\text{m}/\sqrt{612} \approx \pm 0.121\mu\text{m}$. So Δz and the encoder error are in the same order; This result confirms Δz as a realistic error value that easily satisfies the accuracy requirements of a precision of $100\mu\text{m}$.

5. Performance Test

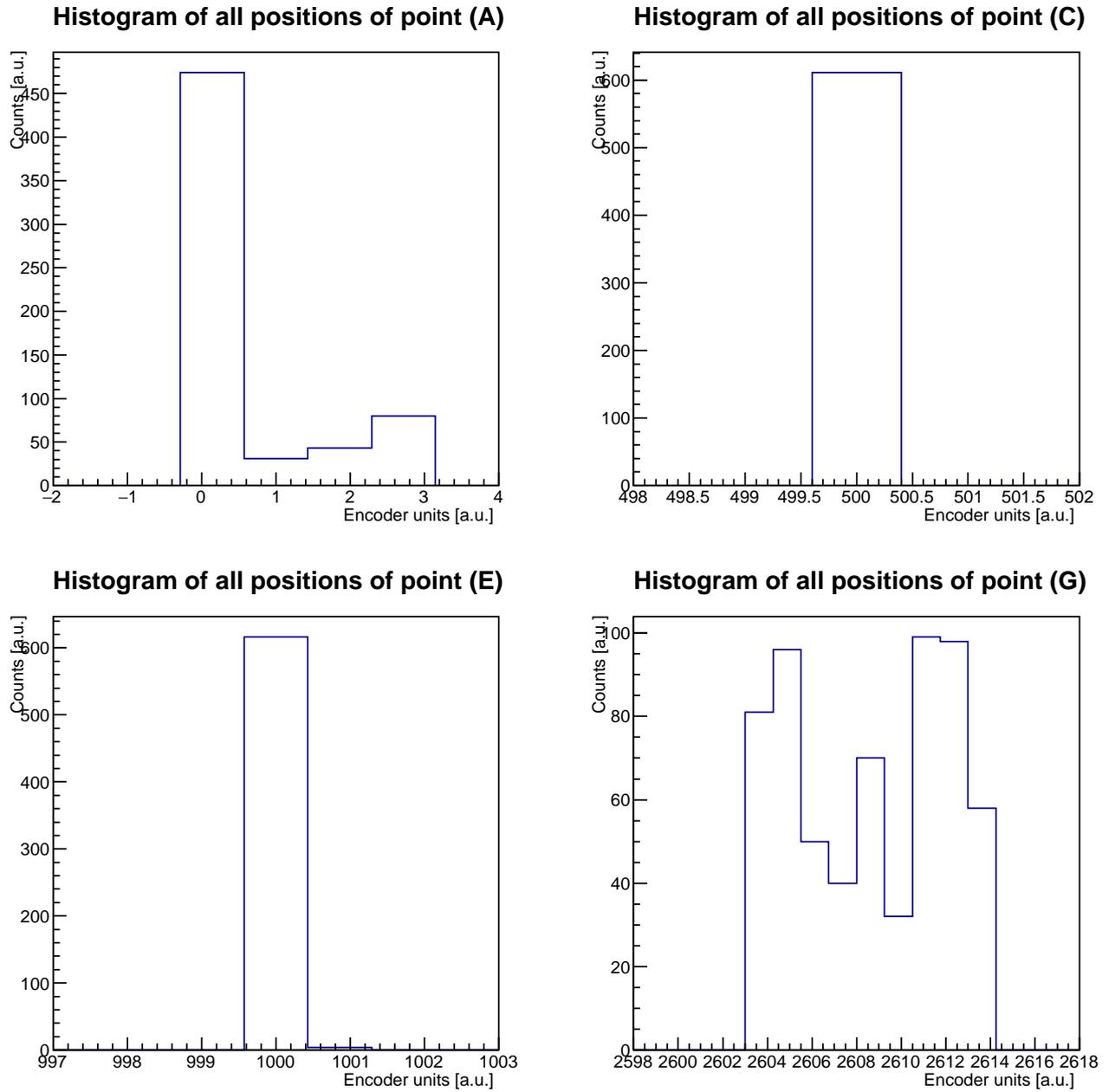


Figure 5.3.: Histograms of all positions of the points A, C, E and G. The form of the peak in histogram G is a result of the mechanical issues during the driving test, but has no effect on the precision of the other points.

5.3.2. Mechanical stability over time

After 612 runs, the setup stopped moving due to mechanical weakness. As section 5.3.1 showed, the position's accuracy did not change, so the derivation (the velocity) was checked for irregularities. Analogous to figure 5.2, the velocity during the motion (sections B, D, F and H) of a typical driving sequence can be plotted. The results are shown in figure 5.4. The motion is separated into the same 8 sections as in 5.2.

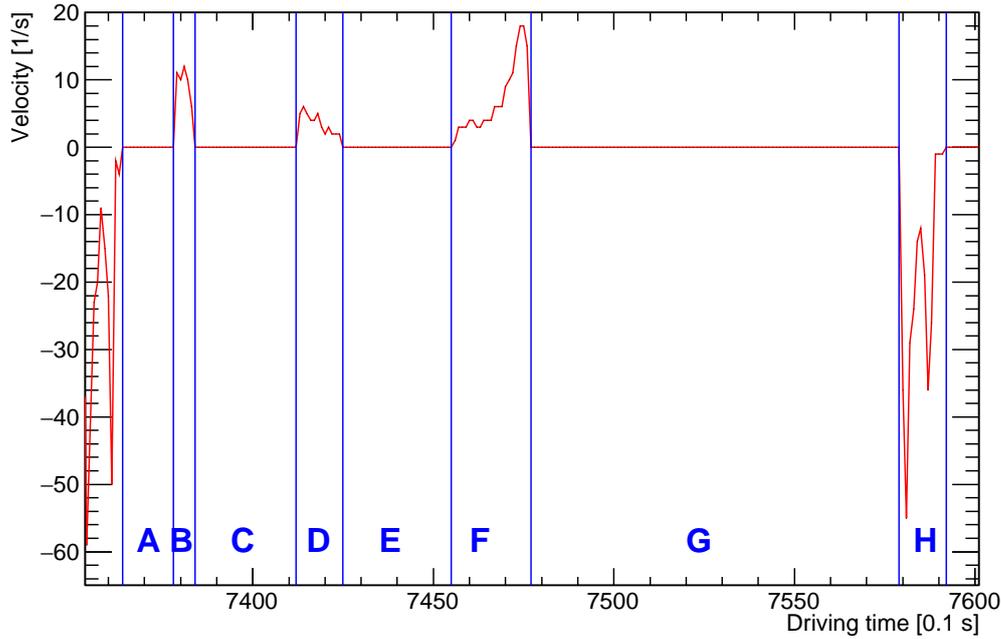


Figure 5.4.: Velocity distribution during one typical driving sequence. The motor starts on a low position level (A) ($v_A = 0$), drives to its first resting point (B) ($v_B > 0$), stops there for the first x motor to move (C) ($v_C = 0$), drives further to the second resting point (D) ($v_D > 0$), stays there for the second x motor to move (E) ($v_E = 0$), drives towards the end point (F) ($v_F > 0$), stays there (G) and returns to its starting point (H) ($v_H < 0$).

- A: The z motor rests ($v_A = 0$) at its starting point at $z = 0$ encoder units.
- B: The z motor drives ($v_B > 0$) to its first resting point at $z = 500$ encoder units.
- C: The z motor stops ($v_C = 0$) its motion and the first x motor is allowed to move.
- D: After the motion of the first x motor is completed, the z motor drives to the second resting point ($v_D > 0$) at $z = 1000$ encoder units.
- E: At the second resting point, the z motor stops its motion again ($v_E = 0$). Now the second x motor is allowed to move.

5. Performance Test

- F: After the motion of the second x motor is completed, the z motor is moving ($v_F > 0$) towards the end of the rail at $z = 2613$ encoder units.
- G: The z motor reaches its end point and hits the switch at the end of the rail ($v_G = 0$).
- H: The z motor returns to its starting point at $z = 0$ encoder units ($v_H < 0$).

By comparing the sections B and D, the shapes and the durations of v_B and v_D are noticeable different, even though their distances are the same (both 500 encoder units²¹). This is based on mechanical issues in the experimental setup. The position of the pressure's working point on the carriage varies over time and leads to a cant between the rail and the carriage that may also slow down the motion. Additionally, the force transmission from the z motor onto the carriage decreases during the driving test's duration: every motion causes vibrations, proceeding from the motor. Bit by bit, this loses the screws that keep the motor pressed onto its ceramic running surface, thus the transmission of force onto the carriage decreases and the motion slows down. A graphical representation of this phenomenon's effect on all the z motions can be seen in figure 5.5: The motion's duration increases over time. Another effect of the decreasing

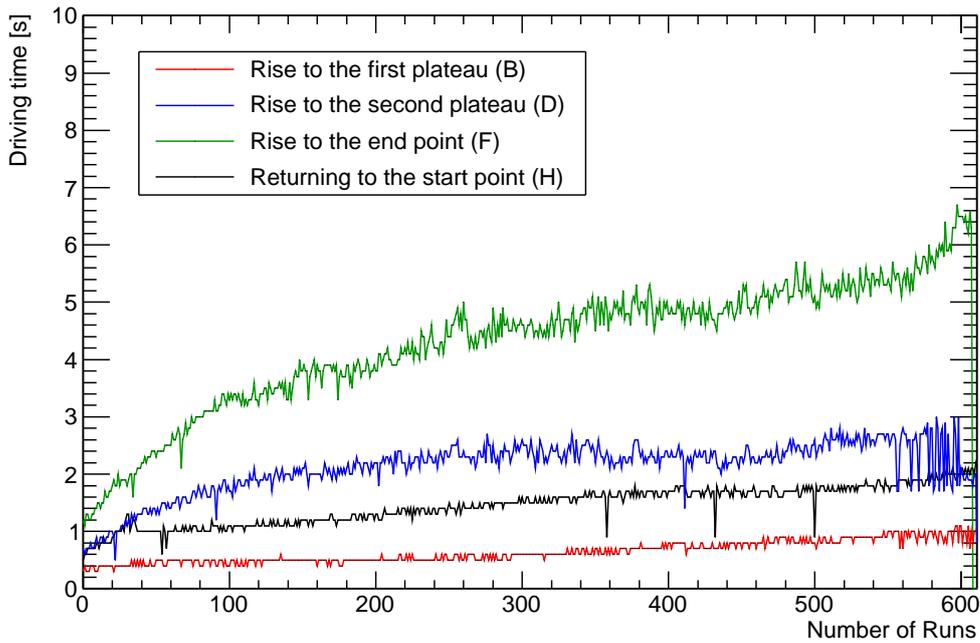


Figure 5.5.: Variations in the duration of each motion over all runs. The motion's duration gets longer due to mechanical issues in the experimental setup.

force transmission is the distribution of the end point position (G). The less pressure

²¹The thrust of the motor depends on the distance to drive.

5. Performance Test

the z motor applies to the sledge, the less sufficient is the z motor thrust to push the carriage towards the end point switch and the earlier the motion stops. This can be seen in figure 5.6, where the time dependency of the end point position is plotted. To prevent a complete motion breakdown, the state machine's state `driveztoG` passes over to state `returnztoB` either when the switch is pressed or after 10 seconds without any changing of position. Even though the experimental setup's mechanical parts fail after

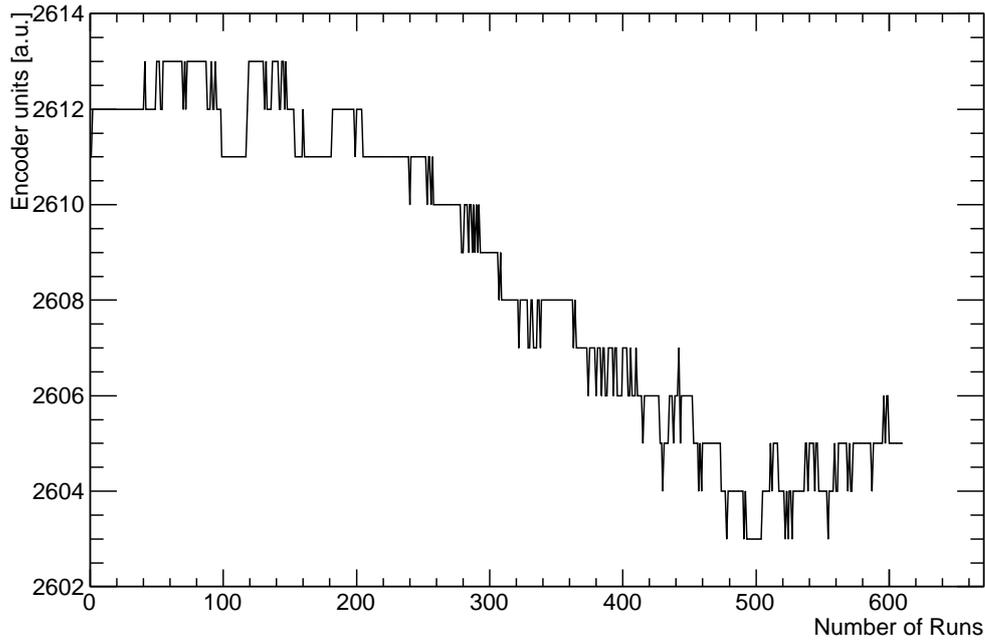


Figure 5.6.: Decreasing of the motion end points by time. Due to mechanical issues in the experimental setup, the end point moves towards the starting point.

612 runs, it achieves the requirements of reliability: 612 completed runs are equivalent to 1224 target substitutions, respectively 244 changes of the whole rack's targets, for it is planned to switch every target only once.

6. Summary and Outlook

The development and implementation of a control system for the primary target for the $\overline{\text{P}}\text{ANDA}$ hypernuclei experiment was realized as follows:

First, the motions in z direction were controlled by using a joystick. With this setup, the behavior of the combination of the z motor and other mechanical components during motion was tested. It showed up that the velocity varied, depending on the z position, due to mechanical problems with the rail. After that, the z motor was controlled by the Galil board with scripts. To improve the setup, a special firmware to control ceramic motors was written and uploaded on the Galil controller board. In a second driving test, the Galil board was controlled with EPICS and a state machine code. By optimizing the PID controller parameters, the accuracy was increased further. In a long time test, both x and z motions were automatized by a state machine to simulate the control process during the hypernuclei experiment.

After analyzing the data in chapter 5, it turned out that the primary target control system achieves the requirements in precision and reliability of the $\overline{\text{P}}\text{ANDA}$ hypernuclear experiment. The carriage can be positioned with a precision of $\Delta z < 0.5 \mu\text{m}$, which is much more exact than the necessary $100 \mu\text{m}$ that are needed to drive the target safely through the gap in the wall of absorber material into the antiproton beam.

The reliability of the experimental setup also achieves the requirements. The setup fulfilled over 1220 simulated target substitutions in succession. The rack can carry a load of 5 targets, so the simulation contained the substitution of 244 complete target stacks.

The adjustment of the PID controller parameters and the different settings in the state machine code could compensate small mechanical issues and keep the experiment running. Future modifications of this setup could be the insertion of a second carriage underneath the carriage to keep the pressure from the motor perpendicular to the rail constant. This might prevent the carriage from tilting. In addition this effect could be minimized by using a rail with a higher stiffness.

On the other hand, the plastic parts of the carriage are not applicable to be used in a vacuum environment, so the whole rail system would have to be modified for a potential usage as a part of the $\overline{\text{P}}\text{ANDA}$ hypernuclear experiment. Also plastic parts of the x motors make them unsuited for vacuum. In addition, the encoder is not radiation-hard. However, with an alternative position control system, the setup is usable.

Although the mechanical setup has to be revised, the control system, as it is based on EPICS, can be transferred and adapted to other hardware components.

Bibliography

- [Arg] ARGONNE NATIONAL LABORATORY.
Experimental Physics and Industrial Control System.
URL: <http://www.aps.anl.gov/epics/>.
- [Ber+12] BERINGER, J. et al.
“Review of Particle Physics”.
In: *Phys. Rev. D* 86 (1 July 2012), p. 010001.
DOI: 10.1103/PhysRevD.86.010001.
URL: <https://link.aps.org/doi/10.1103/PhysRevD.86.010001>.
- [Ber06] BERGER, CHRISTOPH.
Elementarteilchenphysik: Von den Grundlagen zu den modernen Experimenten.
2., aktualisierte und überarbeitete Aufl.
Springer-Lehrbuch.
Berlin and New York: Springer, 2006.
URL: <http://search.ebscohost.com/login.aspx?direct=true&scope=site&db=nlebk&db=nlabk&AN=163425>.
- [Ble] BLESER, SEBASTIAN.
“The target system for the hypernuclear experiment at PANDA”.
Dissertation. Mainz: Johannes Gutenberg-Universität.
In preparation.
- [Ble10] BLESER, SEBASTIAN.
“Eigenschaften von Silizium-Streifendetektoren in direktem Kontakt mit Absorbermaterial für das sekundäre Target bei PANDA”.
Diplomarbeit. Mainz: Johannes Gutenberg-Universität, 2010.
- [Böl17] BÖLTING, MICHAEL.
“Design, Realisierung und Test der Steuersoftware für das primäre Target des PANDA-Hyperkernexperiments”.
Diplomarbeit. Mainz: Johannes Gutenberg-Universität, 2017.
- [GHM16] GAL, A., HUNGERFORD, E. V., and MILLENER, D. J.
“Strangeness in nuclear physics”.
In: *Reviews of Modern Physics* 88.3 (2016), p. 1.
DOI: [10.1103/RevModPhys.88.035004](https://doi.org/10.1103/RevModPhys.88.035004).
- [Kas06] KASEMIR, KAY.
EPICS Channel Access and Client Tools.
2006.

Bibliography

- URL: <http://www.aps.anl.gov/epics/docs/USPAS2014/1-Monday/CA-Clients.pdf>.
- [KC16] KASEMIR, KAY and CARCASSI, GABRIELE.
Control System Studio Guide for installers and maintainers of CS-Studio.
Documentation.
2016.
- [Kno10] KNOLL, GLENN F.
Radiation Detection and Measurement.
4., Auflage.
New York, NY: Wiley, J, 2010.
- [Leo94] LEO, WILLIAM R.
Techniques for nuclear and particle physics experiments: A how-to approach.
Berlin: Springer, 1994.
- [LF15] LUPTON, WILLIAM and FRANKSEN, BENJAMIN.
State Notation Language and Sequencer Users' Guide: Release 2.2.3.
2015.
URL: <http://www-csr.bessy.de/control/SoftDist/sequencer/Manual.pdf>.
- [Lun10] LUNZE, JAN.
Regelungstechnik.
8., neu bearb. Aufl.
Springer-Lehrbuch.
Berlin: Springer, 2010.
- [Mic] MICROE SYSTEMS.
Mercury 1000V: Vacuum Rated Analog Encoder.
- [PAN05] PANDA COLLABORATION.
Technical Progress Report for: PANDA: (AntiProton Annihilations at Darmstadt): Report.
2005.
URL: https://panda.gsi.de/oldwww/archive/public/panda_tpr.pdf
(visited on 06/02/2016).
- [PAN09a] PANDA COLLABORATION.
Physics Performance Report for: PANDA: (AntiProton Annihilations at Darmstadt): Report.
2009.
URL: https://panda.gsi.de/oldwww/archive/public/panda_pb.pdf
(visited on 06/02/2016).
- [PAN09b] PANDA COLLABORATION.
Physics Performance Report for PANDA: Strong Interaction Studies with Antiprotons.
2009.

Bibliography

- URL: <https://arxiv.org/pdf/0903.3905.pdf>.
- [PAN16] PANDA COLLABORATION.
Study of doubly strange systems using stored antiprotons.
2016.
URL: <http://dx.doi.org/10.1016/j.nuclphysa.2016.05.014>.
- [Par14a] (PARTICLE DATA GROUP), K.A. OLIVE ET AL.
LAMBDA BARYONS.
2014.
URL: <http://pdg.lbl.gov/2014/tables/rpp2014-tab-baryons-Lambda.pdf> (visited on 03/27/2018).
- [Par14b] (PARTICLE DATA GROUP), K.A. OLIVE ET AL.
XI BARYONS.
2014.
URL: <http://pdg.lbl.gov/2014/tables/rpp2014-tab-baryons-Xi.pdf> (visited on 03/27/2018).
- [PH] POGORZELSKI, DIPL.-ING KAMIL and HILLENBRAND, DR.-ING. FRANZ.
Inkrementalgeber - Funktionsprinzip und grundsätzliche Auswertemöglichkeiten ihrer Signale.
URL: http://www.imc-berlin.de/fileadmin/Public/Downloads/Whitepapers/WP_Inkrementalgeber_I.pdf.
- [Piea] PIEZOMOTOR.
Piezo Wave Demokit Manual.
Ed. by PIEZOMOTOR.
URL: http://sei.ckcest.cn/product_img/390002/15653/3446885/Document/p1197919411412.pdf.
- [Pieb] PIEZOMOTOR.
PiezoWave Linear 0.1N Manual.
Ed. by PIEZOMOTOR.
URL: <http://www.nrc.com.tw/CAT/PiezoMotor/WL0104A.pdf>.
- [Pov+06] POVH, BOGDAN et al.
Teilchen und Kerne: Eine Einführung in die physikalischen Konzepte.
7. Auflage.
Springer-Lehrbuch.
Berlin, Heidelberg: Springer-Verlag Berlin Heidelberg, 2006.
DOI: 10.1007/3-540-36686-5.
URL: <http://dx.doi.org/10.1007/3-540-36686-5>.
- [Rau16a] RAUSCH, NICOLAS.
“Slowcontrol for the primary target of the PANDA hypernuclear experiment”.
Bachelorarbeit. Mainz: Johannes Gutenberg-Universität, 2016.
- [Rau16b] RAUSCH, NICOLAS.

Bibliography

Slowcontrol for the primary target of the PANDA hypernuclear experiment.
Vortrag.

Mainz, 2016.

- [Roj16] ROJO, MARTA MARTINEZ.
“Design, construction and test of the primary target system for PANDA”.
Master Thesis. Mainz: Johannes Gutenberg-Universität, 2016.
- [Sau18] SAUER, BIRTE.
“Carbon fiber as primary target at the hypernuclear experiment of PANDA”.
Bachelorarbeit. Mainz: Johannes Gutenberg-Universität, 2018.
- [Sch17] SCHUPP, FALK.
“Entwicklung einer Methode zur strahlenharten Positionierung für das primäre
Hyperkerntarget an PANDA”.
Diplomarbeit. Mainz: Johannes Gutenberg-Universität, 2017.
- [Ste] STEINEN, MARCELL.
“The germanium detector array for the hypernuclear experiment at PANDA”.
Dissertation. Mainz: Johannes Gutenberg-Universität.
In preparation.
- [Ste11] STEINEN, MARCELL.
“Untersuchung von Germanium-Detektoren für das PANDA-Experiment”.
Diplomarbeit. Mainz: Johannes Gutenberg-Universität, 2011.

List of Figures

2.1. Scheme of the production and decay cascade of the double λ hypernuclei .	3
2.2. Primary and secondary target	4
2.3. CAD sketch of the target chamber	5
3.1. Block diagram of the setup	7
3.2. The mechanical parts of the setup	8
3.3. Functional principle of the PiezoWave motor	9
3.4. Microscope picture of the encoders' glass scale	10
4.1. Schematic view of the state machine drive	17
5.1. Comparison of the carriage's dynamics for several PID control parameters	22
5.2. Position distribution during one typical driving sequence	25
5.3. Histograms of all plateau positions	26
5.4. Velocity distribution during one typical driving sequence	27
5.5. Variations in the duration of each motion over all runs	28
5.6. Decreasing of the motion end points by time	29

List of Tables

2.1. Properties of Λ and Ξ^- baryons	2
5.1. Hardware parameters of the Galil board	20
5.2. Accuracy of the motor motion, depending on the variation of the used parameters	21
5.3. Calculating the conversion factor between the encoder scale and a metric scale	21
5.4. Optimizing the PID controller parameters	23
5.5. Estimated and measured positions	24
A.1. Variables of the state machine drive	37

A. Tables

Variable Name	Variable type	assigned PV	Comments
Variables of the encoder			
encoderoffset	Integer	PANDAHYP:BBB2:ENCOFFSET	Offset value of the encoder position
encoderposition	Integer	galil01:A:REP	actual encoder position
Variables of the z motor			
zmoveto	Integer	galil01:A:VAL	Estimated z position
Variables of the x motors			
xmotor[n]out	Integer*	PANDAHYP:BBB2:PMF[n]	set to 1 as forward driving command for the nth x motor
xmotor[n]in	Integer*	PANDAHYP:BBB2:PMB[n]	set to 1 as backward driving command for the nth x motor
Variables of the pressure switches			
pswitch	Integer	PANDAHYP:BBB2:PSWITCHP	Status of the end point switch (0 when pressed)
nswitch	Integer	PANDAHYP:BBB2:PSWITCHN	Status of the end point switch (1 when pressed)
miscellaneous variables			
z	Integer	–	counter for the x motor's steps
fehler	Integer	–	difference between destination and the actual encoder position
soll	Integer	–	encoder position to go to
i	Long	–	Index of the logfile entry
xout	Integer	–	Count of x motors extended
xin	Integer	–	Count of x motors in resting position
*) 1-dimensional array of integers with n entries			

Table A.1.: Variables of the state machine **drive**

Acknowledgement

So. Geschafft.

Mein Dank gilt Prof. Dr. Josef Pochodzalla, dass ich diese Arbeit in seiner Forschungsgruppe schreiben durfte und für seine Ratschläge und Hilfe.

Ein dickes Dankeschön an Marcell Steinen, Sebastian Bleser, Michael Bölting und Falk Schupp für ihre schier unendliche Geduld und Hilfsbereitschaft.

Und an meine Frau, die mir den Rücken freihält und mich immer motiviert, weiterzumachen, an meine Freunde und Familie und denjenigen, der koffeinhaltige Getränke erfunden hat.