

# Feasibility Studies for Measuring the Astrophysical S-Factor of the Reaction $^{12}\text{C}(\alpha, \gamma)^{16}\text{O}$ via Electro-Disintegration at MAGIX

## Dissertation

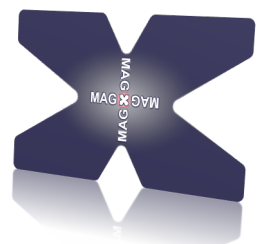
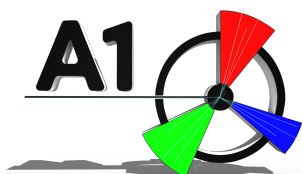
zur Erlangung des Grades  
„Doktor der Naturwissenschaften“

am Fachbereich Physik, Mathematik und Informatik  
der Johannes Gutenberg-Universität in Mainz

Stefan Lunkenheimer

INSTITUT FÜR KERNPHYSIK  
JOHANNES GUTENBERG-UNIVERSITÄT

MAINZ, 22. NOVEMBER 2021



**Autor**

Stefan Lunkenheimer  
MAGIX - Collaboration  
Institut für Kernphysik  
Johann-Joachim-Becher-Weg 45  
Johannes Gutenberg-Universität  
D-55128 Mainz  
stelunke@uni-mainz.de

**Tag der mündlichen Prüfung**

3. März 2022

## Abstract

The nucleosynthesis process  $^{12}\text{C}(\alpha, \gamma)^{16}\text{O}$  is of utmost importance for nuclear astrophysics. The S-factor of this reaction is a major parameter for determining the abundance ratio of oxygen to carbon in stellar nucleosynthesis and plays a crucial role in understanding the pre-supernova evolution. The current relative error of more than 17% in the S-factor for energies in the Gamow-peak does not allow a prediction of the abundance of higher mass elements produced in the pre-supernova evolution of stars. Since the cross section of the reaction  $^{12}\text{C}(\alpha, \gamma)^{16}\text{O}$  is low, a high-precision low-energy experiment with a high luminosity is required to measure the S-factor. One such experiment that aims at measuring this key and challenging quantity is MAGIX.

MAGIX is a versatile fixed-target experiment, which will be built at the new electron accelerator MESA in Mainz, Germany. The accelerator will deliver polarized and unpolarized electron beams to MAGIX with currents up to 1 mA at 105 MeV. Using its internal gas jet target, MAGIX will reach a luminosity of  $\mathcal{O}(10^{35} \text{ cm}^{-2} \text{ s}^{-1})$ . MAGIX will be equipped with two magnetic spectrometers, that allow to measure the kinematic parameters of the scattered electrons with a momentum resolution of  $\frac{\delta p}{p} < 10^{-4}$  and an angular resolution of  $\Delta\Theta < 0.05^\circ$ . To detect heavy charged particles, MAGIX will employ a silicon strip detector array. This will allow studying processes with very low cross sections at small momentum transfer within a rich physics program.

At MAGIX, a measurement is planned in which the S-factor will be determined by investigating the electro-disintegration reaction  $^{16}\text{O}(e, e'\alpha)^{12}\text{C}$ . Electrons will be scattered inelastically on oxygen and MAGIX will detect the scattered electrons and the produced  $\alpha$ -particles in coincidence. With this measurement, it will be possible to determine the S-factor of the reaction  $^{12}\text{C}(\alpha, \gamma)^{16}\text{O}$  as a function of the center of mass energy of the  $^{12}\text{C}$ - $\alpha$  system.

The simulations predict that MAGIX will be able to measure the S-factor for center of mass energies above 1 MeV with a precision competitive to the existing data. To improve these results, MAGIX will be operated in addition with a Zero Degree Tagger to detect the scattered electrons in the range  $\theta_e < 0.5^\circ$ . Given the promising simulation results, MAGIX expects to make important contributions to the measurement of the S-factor of the reaction  $^{12}\text{C}(\alpha, \gamma)^{16}\text{O}$  down to 0.5 MeV, exploring a new territory in nuclear astrophysics.

**Keywords:** MAGIX, MESA, Nuclear Astrophysics, Helium Burning, S-Factor,  $^{12}\text{C}(\text{a,g})^{16}\text{O}$ , Electro-Disintegration,  $^{16}\text{O}(e, e'\alpha)^{12}\text{C}$ , EPICS, Slow Control, Simulation





# Contents

<b>Abstract</b>	<b>i</b>
<b>Contents</b>	<b>iii</b>
<b>1 Introduction</b>	<b>1</b>
<b>2 The Reaction <math>^{12}\text{C}(\alpha, \gamma)^{16}\text{O}</math> in the Helium Burning Stage in Stars</b>	<b>5</b>
2.1 Overview of Stellar Nucleosynthesis . . . . .	5
2.1.1 Nuclear Reactions . . . . .	5
2.1.2 Cross Section of Stellar Nucleosynthesis . . . . .	7
2.1.3 Calculation of the Gamow Peak . . . . .	9
2.1.4 The Hydrogen Burning Stage . . . . .	12
2.1.5 The Helium Burning Stage . . . . .	16
2.2 The $\alpha$ -Capture Process of $^{12}\text{C}$ in Detail . . . . .	20
2.3 Discussion of Measurement Techniques . . . . .	24
2.3.1 Direct Measurement Techniques . . . . .	24
2.3.2 Indirect Measurement Techniques . . . . .	27
2.4 Results of Existing Measurements . . . . .	29
2.5 Production Factor . . . . .	34
<b>3 Electro-Disintegration Cross Section of <math>^{16}\text{O}</math></b>	<b>37</b>
3.1 Kinematics of the Electro-Disintegration Reaction . . . . .	37
3.2 Differential Cross Section of the Electro-Disintegration Reaction $^{16}\text{O}(e, e'\alpha)^{12}\text{C}$ . . . . .	38
3.3 Calculation of the Matrix Element . . . . .	40
3.4 Definition of Spherical Unit Vectors . . . . .	41
3.5 Calculation of Generalized Rosenbluth Factors . . . . .	42
3.6 Calculating the Generalized Mott Cross Section . . . . .	45
3.7 Combining Results to Differential Cross Section . . . . .	45
3.8 Transformation to Center of Mass Frame . . . . .	46
3.9 Approximation Using the Extreme Relativistic Limit . . . . .	48
3.10 Multipole Decomposition of the Response Functions . . . . .	50
3.11 Representing the Electric Matrix Elements $t_{EJ}$ with the S-Factor Components $S_{EJ}$ . . . . .	52
<b>4 Accelerator and Experimental Setup</b>	<b>53</b>
4.1 Mainz Energy-Recovering Superconducting Accelerator . . . . .	53
4.1.1 The P2-Experiment . . . . .	55
4.1.2 The DarkMESA Experiment . . . . .	56
4.2 Mainz Gas Injection Target Experiment . . . . .	56

4.2.1	Gas Jet Target . . . . .	58
4.2.2	STAR/PORT Spectrometer Setup . . . . .	60
4.2.3	Silicon Strip Recoil Detector Array . . . . .	63
4.2.4	Zero Degree Tagger – FORE . . . . .	65
4.3	S-Factor Measurement at MAGIX . . . . .	66
4.3.1	Discussion of Electro-Disintegration Experiment . . . . .	67
4.3.2	Discussion of the Planned Extension Stages . . . . .	68
<b>5</b>	<b>Development of the MAGIX Slow Control System</b>	<b>71</b>
5.1	Requirements for a Slow Control System . . . . .	71
5.2	Experimental Physics and Industrial Control System . . . . .	72
5.2.1	Input-Output-Controller . . . . .	73
5.2.2	CA Client . . . . .	76
5.2.3	The Channel Access Communication Protocol . . . . .	77
5.3	Community Modules . . . . .	78
5.3.1	The Asyn Module . . . . .	78
5.3.2	The StreamDevice Module . . . . .	79
5.3.3	The Autosave Module . . . . .	80
5.3.4	The PyEpics Library . . . . .	80
5.3.5	The dev-gpio Module . . . . .	81
5.4	The Graphical User Interface - Control System Studio . . . . .	81
5.5	MAGIX Slow Control System . . . . .	82
5.5.1	Installation Script . . . . .	82
5.5.2	Introduction to Environment Variables . . . . .	83
5.5.3	Execution via System Control Manager . . . . .	83
5.5.4	Defining UDEV Rules . . . . .	85
5.5.5	Implementation of the Bash/ZSH Completion . . . . .	85
5.5.6	Integration of an Archiver (MAGIX executable) . . . . .	86
5.5.7	Development of the MXFrontend (MAGIX module) . . . . .	87
5.5.8	Expansion by the dbLoadMultiRecords Function . . . . .	87
5.5.9	Development for Profibus Support (StreamDevice BUS API) . . . . .	90
5.5.10	MXSlowControl Tutorial . . . . .	90
5.5.11	Development of Device Support for MAGIX Built Devices . . . . .	90
5.6	Slow Control Development for the Silicon-Strip Detector Test Stand . . . . .	93
5.6.1	Readout of the MPT200 Device via StreamDevice ttyUSB . . . . .	93
5.6.2	Controlling the HMP4040 Device via StreamDevice TCP . . . . .	95
5.6.3	Driver Development to Control the SiDet board via RPi I <sup>2</sup> C . . . . .	96
<b>6</b>	<b>Simulation of the Electro-Disintegration Reaction <math>^{16}\text{O}(e, e'\alpha)^{12}\text{C}</math></b>	<b>103</b>
6.1	Overview of Physics Simulations . . . . .	103
6.1.1	Principle of Monte Carlo Integration . . . . .	104
6.1.2	General Idea of a Cross Section Simulation for a Scattering Experiment . . . . .	108
6.1.3	Initial State Particle Generator . . . . .	109
6.1.4	Final State Particle Generator . . . . .	110
6.1.5	Physics Model Calculations . . . . .	110
6.1.6	Acceptance Simulation . . . . .	111
6.1.7	Calculation of the Cross Section . . . . .	111
6.2	Implementation of the S-Factor Simulation . . . . .	112
6.2.1	Overview of the Initial State Generator . . . . .	112
6.2.2	Development of the Two-Body Final State Generator . . . . .	113
6.2.3	Development of the S-Factor Model Calculation . . . . .	115

---

6.2.4	Implementation of Acceptance Simulation . . . . .	116
6.2.5	Analysis: Calculation of Expected Count Rate . . . . .	118
6.2.6	Discussion of the Simulations Results . . . . .	122
<b>7</b>	<b>Summary and Outlook</b>	<b>127</b>
<b>A</b>	<b>Response Functions</b>	<b>131</b>
<b>B</b>	<b>Selected Data</b>	<b>135</b>
<b>C</b>	<b>EPICS Support Files</b>	<b>141</b>
<b>D</b>	<b>Supplemental Material – Simulation</b>	<b>157</b>
	<b>List of Figures</b>	<b>161</b>
	<b>List of Tables</b>	<b>163</b>
	<b>List of Codes</b>	<b>165</b>
	<b>Acronyms</b>	<b>167</b>
	<b>Index</b>	<b>169</b>
	<b>Bibliography</b>	<b>173</b>



---

---

# CHAPTER 1

---

## Introduction

### Studying the interior of stars in laboratory experiments

During a clear night, one can see with modern telescopes millions of galaxies in the sky, containing billions of stars. Some of these celestial objects are several billion years old before their light reaches us. Since ever, people have wondered what other solar systems and the planets to be found there might look like. In addition, the question also arises whether there is life on other worlds and how this could have evolved. To be able to provide answers to such questions, science depends on observations of the universe and, as far as possible, on experiments in terrestrial laboratories. This results in models, which can additionally be simulated. The results of such simulations can then be compared with the observations. All these endeavors improve our picture of the universe.

To understand the formation and evolution of solar systems, it is important to understand the formation of elements in stellar interiors. This is called stellar nucleosynthesis and describes the processes by which a star fuses hydrogen into more massive elements such as helium, carbon, nitrogen, oxygen, etc. during its lifetime. While our sun is still in the first stage, namely the hydrogen burning stage, we can observe other stars that are already in later stages, such as the helium or the carbon burning stage. To describe the evolution of a star, especially the different burning stages, we need to understand the nuclear processes and their reaction rates. Hereby, the cross section is an important quantity, which is related to the probability of a certain reaction.

In the hydrogen burning stage, protons are fused to helium via different proton-proton chains or carbon-nitrogen-oxygen cycles. The processes involved in this phase are well known and have been investigated to such an extent that the timing of this phase can be well determined. The situation is different with the subsequent phase, the helium burning stage. Here, two reactions are crucial: the triple- $\alpha$  process, in which a  $^{12}\text{C}$  is formed from three  $\alpha$ -particles via a reaction chain, and the fusion of carbon and helium to form oxygen  $^{12}\text{C}(\alpha, \gamma)^{16}\text{O}$ . These two processes occur in parallel in the helium burning stage, but are in no sense to be understood as a reaction chain. The second reaction, the  $\alpha$ -capture of carbon, proceeds much slower than the triple- $\alpha$  process, so that at the end of this burning stage only

a small fraction of the carbon has been converted to oxygen. The description of further processes and of the further evolution of a star depends sensitively on the C/O abundance ratio. To estimate this ratio, one must determine the cross section of the  $^{12}\text{C}(\alpha, \gamma)^{16}\text{O}$  reaction precisely. Therefore, the nucleosynthesis process  $^{12}\text{C}(\alpha, \gamma)^{16}\text{O}$  is of utmost importance for nuclear astrophysics. For energies available in the helium burning stage, the cross section of this reaction is low. For this reason, this reaction can be studied in the laboratory only with large efforts in experiments. An important parameter is the S-factor. This is the unknown part of the cross section when the well-known parameters such as the Coulomb penetration factor are removed. The S-factor of a fusion represents any effect related to the nuclear structure. Despite efforts over the past 50 years to determine the S-factor of this reaction, it has not yet been possible to determine it with sufficient accuracy. Current relative errors are in the range of 17 % to 28 %. In order to understand stellar evolution, and thus the further stages of burning and the production of more massive elements, theory requires that the S-factor must be determined in the low-energy range with an accuracy better than 10 %.

In order to measure the cross section and thus the S-factor in the low energy range, new low-energy experiments must be developed. These experiments must have a high luminosity in order to be able to measure reactions with a low cross section accurately. Only in this way it will be possible to further reduce the relative error of the S-factor of the reaction  $^{12}\text{C}(\alpha, \gamma)^{16}\text{O}$ .

One such experiment planned for the coming years will be MAGIX at MESA. MESA (Mainz Energy-Recovering Superconducting Accelerator) is an upcoming high-intensity electron accelerator and is being built at the Institute of Nuclear Physics of the Johannes Gutenberg University Mainz, Germany. MAGIX (Mainz Gas Injection Target Experiment) will be located in the energy recovering branch of MESA. The core element of MAGIX is the internal gas jet target. To measure the kinematic parameters of the scattered electrons, MAGIX will use two magnetic spectrometers, each equipped with a GEM-based TPC in the focal plane. To detect heavy charged particles, MAGIX will have a silicon strip detector array, located within the scattering chamber. In a later stage, MAGIX will be extended to include a beam-integrated tagging detector. MAGIX at MESA will allow studying processes with very low cross sections at small momentum transfer. One experiment planned is the determination of the cross section of the electro-disintegration reaction  $^{16}\text{O}(e, e'\alpha)^{12}\text{C}$ . The scattered electron and the recoiling  $\alpha$ -particle are detected in coincidence to determine the kinematic parameters of this inelastic reaction. Through theoretical calculations, the S-factor of the reaction  $^{12}\text{C}(\alpha, \gamma)^{16}\text{O}$  can be estimated from the cross section of the electro-disintegration reaction of oxygen.

The aim of this work was to perform a feasibility study using a simulation to answer the question whether MAGIX can significantly contribute to the determination of the S-factor in the low energy range by measuring the electro-disintegration reaction of oxygen.

## Outline

At the beginning, the nucleosynthesis reaction  $^{12}\text{C}(\alpha, \gamma)^{16}\text{O}$  is discussed in detail. For this purpose, the cross section and the S-factor of this reaction are introduced and described with respect to the different oxygen levels. Furthermore, different direct and indirect measurement methods are explained, which allow to determine

---

the S-factor or different contributions of the S-factor. Finally, the results of the previous measurements performed by other research groups are discussed and an overview of the impact of the results of these measurements on the production factor is given. This is a quantity that describes the ratio of isotopes produced during the pre-supernova evolution of massive stars.

While in Chapter 2 the direct reaction  $^{12}\text{C}(\alpha, \gamma)^{16}\text{O}$  was in the focus, in Chapter 3 the electro-disintegration reaction  $^{16}\text{O}(e, e'\alpha)^{12}\text{C}$  is discussed. The aim of this chapter is to describe the theoretical basis of this reaction and to derive the cross section. The theoretical assumptions and calculations of this chapter serve as a basis for a simulation of the electro-disintegration measurement at MAGIX.

Subsequently, the experimental environment is presented in Chapter 4. All components of MAGIX and MESA which are relevant for the electro-disintegration measurement are described here. In order not to go beyond the scope of this work, only the essential properties are described. At the end of this chapter, the planned measurements of the electro-disintegration of oxygen are outlined.

MAGIX uses a slow control system based on EPICS. In Chapter 5, an overview of the development of the MAGIX slow control system is given. Some parts of MAGIX already exist and are tested in different laboratories. At the end of this chapter, an example of setting up a test stand with EPICS is shown, namely the test stand for silicon strip detectors.

After the theoretical studies have been completed and the experimental setup has been described, a simulation is presented. Chapter 6 discusses the development and results of the simulation of the electro-disintegration reaction of oxygen at MAGIX at MESA. First, an overview of physics simulations is given. The principle of Monte Carlo integration and random numbers is also introduced. Then, the general idea of a cross section simulation for scattering experiments is presented. Based on the previous discussions, it is then described in detail how the simulation for the feasibility study is implemented in the MAGIX framework and how the data will be analyzed. At the end of this section, the results will be discussed.

The final chapter of this thesis summarizes the results of this work. It also gives a brief outlook on the S-factor measurement at MAGIX.





---

---

## CHAPTER 2

---

# The Reaction $^{12}\text{C}(\alpha, \gamma)^{16}\text{O}$ in the Helium Burning Stage in Stars

In this chapter we will take a closer look at the astrophysical S-factor of the reaction  $^{12}\text{C}(\alpha, \gamma)^{16}\text{O}$ . The chapter starts with a brief introduction to nuclear astrophysics and an overview of stellar evolution. Then the helium burning stage, especially the associated S-factor of the  $\alpha$ -capture process, will be discussed in detail. Finally, we will assess the results of previous measurements used to determine the S-factor and discuss why this is still challenging as of today.

### 2.1 Overview of Stellar Nucleosynthesis

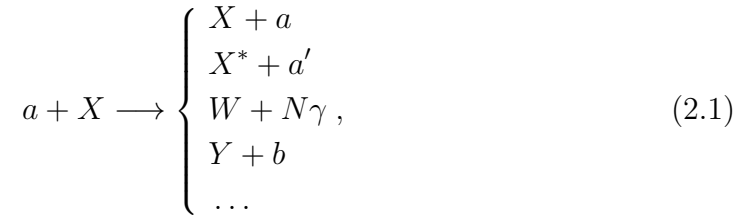
The Big Bang nucleosynthesis describes the production of the first elements in the early universe. After 3 minutes, the main part of the baryon mass consisted of hydrogen and helium, where the mass ratio of  $^4\text{He}$  to the total baryon mass was  $\sim 25\%$ . This corresponds to a ratio by number of atoms of  $^4\text{He}/^1\text{H} = 0.08 = 8\%$  [Zyl+20; ISG13]. Traces of deuterium, tritium and lithium have also been produced, but their abundance compared to hydrogen is several orders of magnitude lower. The ratio by number of atoms are  $\text{D}/^1\text{H} = \mathcal{O}(10^{-5})$ ,  $^3\text{He}/^1\text{H} = \mathcal{O}(10^{-5})$  and,  $^7\text{Li}/^1\text{H} = \mathcal{O}(10^{-10})$  [Zyl+20, Chapter 24]. Therefore, the primordial elements deuterium, tritium and lithium can be disregarded for the stellar evolution and nucleosynthesis [Won98].

After the short time period of the Big Bang nucleosynthesis, the universe expanded further and cooled down. However, it took another 100 million years before the first stars were formed [LB04]. The nuclear fusion between atoms within stars is described by the stellar nucleosynthesis. It is a highly predictive theory that today yields excellent agreement between calculations based upon it and the observed abundances of the elements.

#### 2.1.1 Nuclear Reactions

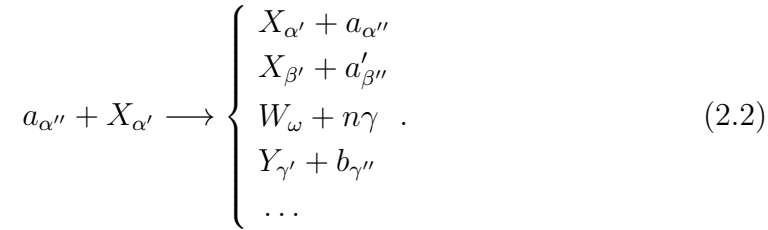
Nuclear reactions are processes in which a nuclear particle (nucleon or nuclei) comes into close contact with another nuclear particle. This may produce one or more nuclear particles that leave the interaction point. These produced particles

are usually different from the incident particle pair [BW91]. For a particle  $a$  that scatters at a particle  $X$  the whole set of nuclear reactions can be written as



where  $a, b, \dots$  denotes the outgoing nuclear particle,  $W$  and  $Y, \dots$  denotes the produced nucleus and  $N\gamma$  denotes the emitted photons. With the exception of the first reaction, these are inelastic nuclear reactions. The first reaction  $X(a, a)X$  of Eq. (2.1) represents the elastic scattering process, where in the center of mass system (cms) the incident particle  $a$  leaves the interaction with the same energy and the target particle  $X$  stays in its initial state. The second reaction  $X(a, a')X^*$  represents the inelastic scattering where the target particle  $X$  leaves the interaction in an excited state  $X^*$  and the outgoing particle  $a'$  leaves with lower energy than initial. The third reaction  $X(a, N\gamma)W$  represents the radiative capture process, where the two initial particles  $a$  and  $X$  stay together to form one final nucleus  $W$  while one or more photons  $\gamma$  are emitted. In the inelastic reactions, the strong and electromagnetic forces are usually involved in order to exchange energy and momentum. The weak nuclear force can also take place in such a reaction, for example to convert neutrons into protons via the  $\beta^-$  processes [BW91; Won98].

All inelastic reactions can be subdivided based on the quantum states of the final particles. The set  $(\alpha', \alpha'')$  denotes the states of the initial particles and the sets  $\{(\beta', \beta''), (\gamma', \gamma''), \dots\}$  the one of the final particles. For a fixed initial channel  $(\alpha', \alpha'')$  we can rewrite Eq. (2.1) as



Due to the conservation laws of energy, angular momentum, and parity, the outgoing quantum states of each set  $\{(\beta', \beta''), \dots\}$  are restricted. All possible sets with fixed quantum state are called reaction channels. Each reaction channel is abbreviated as  $\beta \equiv (\beta', \beta'')$ , for example [BW91].

Furthermore we have to distinguish between direct and compound reactions. In case of a direct reaction the two initial particles interact once with each other in a short time of the order of  $10^{-22}$  s. If the two initial particles „fuse“ together for a long time typically of the order of  $10^{-14}$  s the reaction is called compound reaction. In case of the compound reaction an intermediate state, a so called compound nucleus  $\mathcal{N}$ , is formed. Before any re-emission occurs, the energy of the incident particle is quickly shared through the strong force between all constituents. An important feature is that in a compound reaction the decay of the compound nucleus is independent of the formation of this nucleus. So, one compound nucleus

can have several final reaction channels [Won98; BW91]

$$\mathcal{N}_\alpha \longrightarrow \begin{cases} X_{\beta'} + a'_{\beta''} \\ W_\omega + n\gamma \\ Y_{\gamma'} + b_{\gamma''} \\ \dots \end{cases} . \quad (2.3)$$

### 2.1.2 Cross Section of Stellar Nucleosynthesis

The cross section is an important quantity to define the reaction rate of reactions. It is a measure of the probability with which a given reaction occurs for given initial and final particles. Assuming a scattering process in which an incident particle beam is scattered at a layer of independent scattering centers, the total cross section can be defined as [Pov+15; BW91]:

$$\sigma_{\text{tot}} := \frac{\text{number of reactions per unit time}}{\text{number of incident particles per unit time} \times \text{scattering centres per unit area}} .$$

The dimension of the cross section is area, which is usually given in  $\text{cm}^2$  or barn which is defined as

$$1 \text{ barn} = 1 \text{ b} = 10^{-24} \text{ cm}^2 .$$

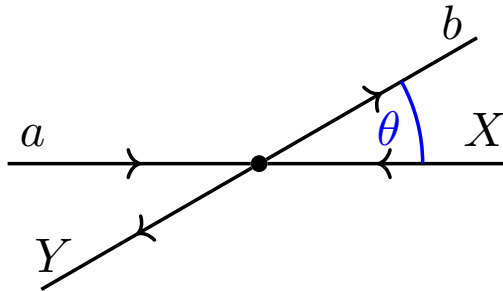
The total cross section  $\sigma_{\text{tot}}$  is a function that depends only on the cms energy  $E$  of a given reaction. The total cross section can be written as an integral of the differential cross section over the total solid angle [Fey04]

$$\sigma(E) = \int_{4\pi} \frac{d\sigma}{d\Omega} d\Omega .$$

The differential cross section describes the cross section for scattering into a small solid angle element  $d\Omega$ . It depends on the cms energy  $E$  and the cms scattering angle  $\theta$ , which is defined as the angle between the incident and final particles in the cms (see Fig. 2.1). In order to describe the dependence on the angle, one can use the multipole expansion to represent the differential cross section in terms of the Legendre polynomials  $P_k(\cos(\theta))$  as [Ros53; Fey04]

$$\frac{d\sigma}{d\Omega} = \sum_k a_k P_k(\cos(\theta)) , \quad (2.4)$$

where the factors  $a_k \equiv a_k(E)$  depend only on the cms energy  $E$ , with  $a_0 = \frac{\sigma}{4\pi}$ .



**Figure 2.1:** Schematic view of a scattering process in the cms. The scattering angle  $\theta$  is defined as the angle between the incident and final particles.

For an initial channel  $\alpha$ , the total cross section of a scattering process can be separated into the elastic cross section  $\sigma_{\text{el}}$  and the combined cross section of all inelastic processes  $\sigma_{\text{inel}}$ , also called the reaction cross section [Pov+15; BW91]:

$$\sigma_{\text{tot}}(\alpha) = \sigma_{\text{el}}(\alpha) + \sigma_{\text{inel}}(\alpha).$$

Moreover, the inelastic cross section can be divided into all possible reaction channels  $\beta$  according to (2.2) [Pov+15; BW91]:

$$\sigma_{\text{inel}}(\alpha) = \sum_{\beta \neq \alpha} \sigma(\alpha, \beta) + \sigma_{\text{cap}}(\alpha), \quad (2.5)$$

where  $\sigma(\alpha, \beta)$  is the inelastic scattering cross section with the initial channel  $\alpha$  and the resulting channel  $\beta$  and  $\sigma_{\text{cap}}$  is the cross section of the radiative capture process. Here, the cross section  $\sigma(\alpha, \beta)$  is defined as [BW91]

$$\sigma(\alpha, \beta) = \frac{\text{number of emissions into channel } \beta \text{ per unit time per scattering center}}{\text{number of incident particles in channel } \alpha \text{ per unit area per unit time}}.$$

In stellar nucleosynthesis studies, we are interested in the inelastic cross section for a given reaction with specific initial and final particles. Since the constituents interact through the strong force, we have to distinguish between two cases, the resonant and the nonresonant.

For the initial channel  $\alpha$  and the final channel  $\beta$  the cross section close to a resonance with the energy  $E_r$  can be written with the Breit-Wigner formula as [Bur+57; BW91; Pov+15]

$$\sigma_{(\alpha, \beta)}(E) = \pi \lambda^2 \frac{(2J+1)}{(2J_a+1)(2J_b+1)} \cdot \frac{\Gamma_\alpha \Gamma_\beta}{(E - E_r)^2 + \Gamma_{\text{tot}}^2/4}, \quad (2.6)$$

where  $\lambda \equiv \lambda(E)$  is the reduced de Broglie wavelength in the cms,  $J_a$  and  $J_b$  are the spin for the initial particles,  $J$  the spin for the final nucleus,  $\Gamma_\alpha$  and  $\Gamma_\beta$  the partial widths for the initial and final channels, and  $\Gamma_{\text{tot}}$  the total width. The total width is the sum of the partial widths of all possible final channels. This cross section is often written as

$$\sigma(E) = \pi \lambda^2 \omega \gamma \cdot \frac{\Gamma_{\text{tot}}}{(E - E_r)^2 + \Gamma_{\text{tot}}^2/4}, \quad (2.7)$$

with the resonance strength [deB+17]:

$$\omega \gamma = \frac{(2J+1)}{(2J_a+1)(2J_b+1)} \cdot \frac{\Gamma_\alpha \Gamma_\beta}{\Gamma_{\text{tot}}}.$$

If the energy falls between resonances the cross section is given as [Bur+57]

$$\sigma(E) = \pi \lambda^2 (2J+1) (\pi^2 \pm 8) \frac{\Gamma_\alpha \Gamma_\beta}{D^2}, \quad (2.8)$$

with the average level spacing  $D$ . If the two nearest resonances interfere constructive the factor  $(\pi^2 + 8)$  is used, otherwise if they interfere destructive the factor  $(\pi^2 - 8)$  is used.

We can now describe the cross section for stellar nucleosynthesis reactions. In case of stellar nucleosynthesis the temperature is around a few GK and hence the most probable energy is in the order of a few keV. It is therefore suitable to

use the non-relativistic approximation to calculate the kinetic energy  $E = \frac{1}{2}\mu v^2$ , with the reduced mass  $\mu$  and the relative velocity  $v$  of the incident particles. In combination with Eqs. (2.5) to (2.8) it follows that the cross section in this low energy region is proportional to  $\lambda^2$ , whereby [Fey04]

$$\lambda^2 \propto \frac{1}{v^2} \propto \frac{1}{E},$$

with the cms energy  $E$ .

Moreover, the probability for a nucleus with the proton number  $Z_1$  tunneling through the Coulomb barrier of a nucleus with proton number  $Z_2$  with relative velocity  $v \equiv v(E)$  is described by the Coulomb penetration factor

$$e^{-\frac{2\pi Z_1 Z_2 \alpha_{\text{em}} c}{v}}.$$

It is common to define the Sommerfeld number

$$\eta \equiv \eta(E) := \frac{Z_1 Z_2 \alpha_{\text{em}} c}{v}, \quad (2.9)$$

with the fine structure constant  $\alpha_{\text{em}}$  and the speed of light  $c$ . Since only charged nuclear particles are involved, the nucleosynthesis cross section contains the Coulomb penetration factor

$$e^{-2\pi\eta}.$$

In total we have

$$\sigma(E) = \frac{1}{E} e^{-2\pi\eta} \cdot S(E),$$

where the so called S-factor is defined as the contribution of the nucleosynthesis cross section without the known dependencies [Fey04]

$$S(E) := \sigma(E) \cdot E \cdot e^{2\pi\eta}. \quad (2.10)$$

It is usually given in the units keV b or MeV b. The specific S-factor of a compound reaction represents any effect pertaining to the nuclear structure [Won98].

### 2.1.3 Calculation of the Gamow Peak

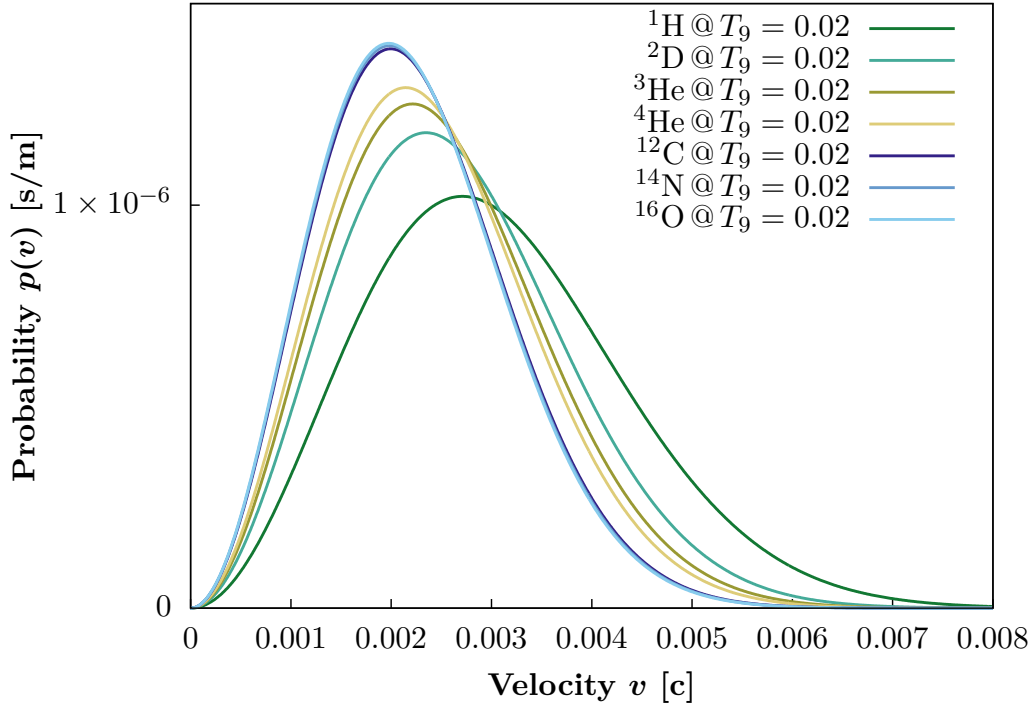
To describe the stellar nucleosynthesis it is crucial to know the energy range in which the reactions take place. On the one hand, the Maxwell-Boltzmann distribution describes the probability to find a particle in a certain energy state. For a gas at the temperature  $T$  the Maxwell-Boltzmann distribution  $p(v)$  as function of the relative velocity  $v$  between two arbitrary particles can be written as [Won98]

$$p(v) dv = 4\pi \left( \frac{\mu}{2\pi kT} \right)^{3/2} e^{-\frac{\mu v^2}{2kT}} v^2 dv, \quad (2.11)$$

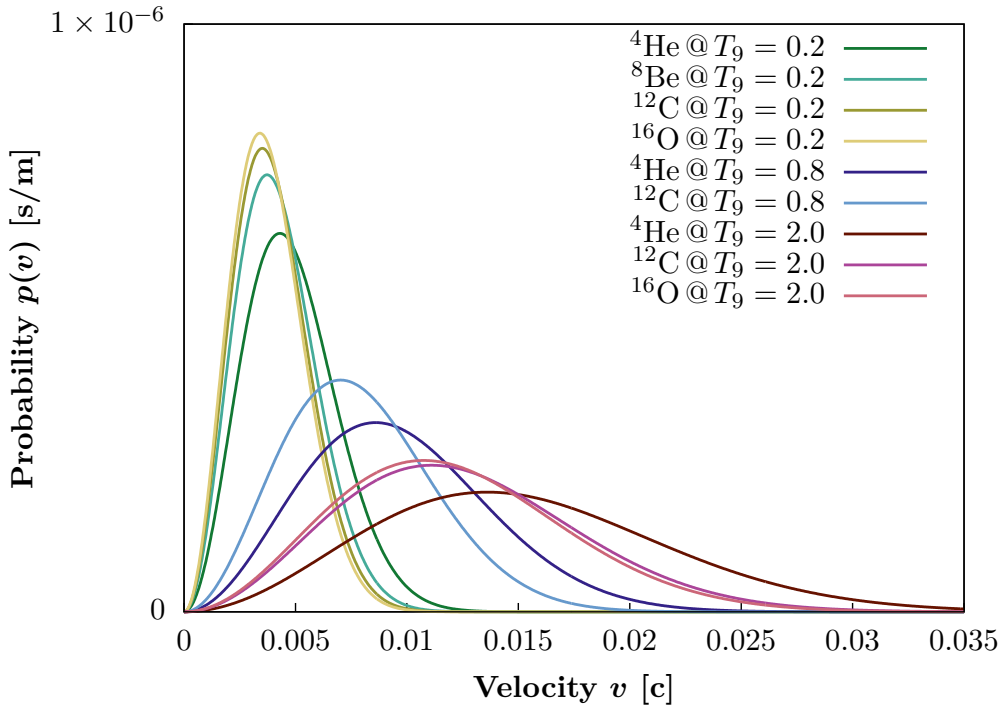
with the reduced mass  $\mu$  and the Boltzmann constant  $k$  (see Table B.8). The reduced mass is calculated with

$$\mu = \frac{M_1 M_2}{M_1 + M_2},$$

where  $M_i$  with  $i = 1, 2$  denotes the mass of the two initial particles. The maximum point of  $p(v)$  is given by  $\frac{1}{2}\mu v_{\text{max}}^2 = kT$ . In Fig. 2.2 a set of Maxwell-Boltzmann distributions for different elements for different burning stages in stars can be



(a) Maxwell-Boltzmann distribution for different elements involved in the hydrogen burning stage  $T_9 = 0.02$ . The velocity always refers to the relative velocity between the respective element and hydrogen.



(b) Maxwell-Boltzmann distribution for different elements involved in the helium burning stage  $T_9 = 0.2$ , the carbon burning stage  $T_9 = 0.8$  and the oxygen burning stage  $T_9 = 2.0$  [Won98]. The velocity always refers to the relative velocity between the respective element and  $^4\text{He}$ .

**Figure 2.2:** Maxwell-Boltzmann distribution for specific elements at the temperatures for specific burning stages in stars.  $T_9$  describes the temperature in units of GK. The maximum point of each distribution is at the relative velocity  $\hat{v} = \sqrt{\frac{2kT}{\mu}}$  for the specific reduced mass  $\mu$  (see Table B.5).

found. As mentioned above it is suitable to calculate the kinetic energy using the non-relativistic approximation  $E_{\max} = \frac{1}{2}\mu v_{\max}^2 = kT$ .

On the other hand, the probability for an element to tunnel through the Coulomb barrier of another element is given by the Coulomb penetration factor. For the energy range of a stellar fusion, we can express the Coulomb penetration factor  $c(E)$  as a function of the cms energy as

$$c(E) := \exp\left\{-2\pi Z_1 Z_2 \alpha_{\text{em}} c \sqrt{\frac{\mu}{2E}}\right\} = \exp\left\{-\frac{b}{\sqrt{E}}\right\}, \quad (2.12)$$

with  $b = \pi Z_1 Z_2 \alpha_{\text{em}} c \sqrt{2\mu}$ .

The mean reaction rate  $\lambda$  per unit volume per unit time in a gas mixture with the particles densities  $n_1$  and  $n_0$  is defined as the integral over the product of relative velocity, the cross section and the Maxwell-Boltzmann distribution [Won98; CHY92; Bur+57]:

$$\lambda := n_1 n_0 \langle \sigma v \rangle = n_1 n_0 \int_0^\infty \sigma(E) \cdot v \cdot p(v) dv. \quad (2.13)$$

The substitution of  $E = \frac{1}{2}\mu v^2$  with  $dE = \mu v dv$  yields

$$\begin{aligned} \lambda &= \frac{1}{\mu} n_1 n_0 \int_0^\infty \sigma(E) \cdot p(E) dE \\ &= \frac{4\pi}{\mu} n_1 n_0 \int_0^\infty \frac{S(E)}{E} e^{-\frac{b}{\sqrt{E}}} \left(\frac{\mu}{2\pi kT}\right)^{3/2} e^{-\frac{E}{kT}} \cdot \frac{2E}{\mu} dE \\ &= \sqrt{\frac{8}{\mu\pi}} \frac{1}{(kT)^{3/2}} n_1 n_0 \int_0^\infty S(E) \exp\left\{-\frac{b}{\sqrt{E}} - \frac{E}{kT}\right\} dE. \end{aligned} \quad (2.14)$$

The integrand can be interpreted as the reaction rate distribution, called Gamow peak. If we assume that there are no resonances involved in the region the reaction takes place, the nuclear cross section factor  $S(E)$  can be assumed to be essentially smooth. To calculate the maximum of the reaction rate distribution it is sufficient to calculate the maximum of the exponential function  $\exp(-f(x))$  with

$$f(E) := \frac{b}{\sqrt{E}} + \frac{E}{kT}.$$

The exponential function has a maximum if and only if the function  $f$  has a minimum. Therefore we calculate

$$\left. \frac{df}{dE} \right|_{E=E_G} = \frac{1}{kT} - \frac{b}{2E_G^{3/2}} \stackrel{!}{=} 0.$$

By solving this equation for the energy we get

$$E_G = \left(\frac{1}{2}bkT\right)^{2/3}. \quad (2.15)$$

In the following chapters we will refer to the energy  $E_G$  as Gamow peak. A schematic view of the Maxwell-Boltzmann distribution, the Coulomb penetration factor and the Gamow peak can be found in Fig. 2.3.

The full width of the Gamow peak is defined as the difference between the two energy values at which the Gamow function is equal to the maximum value divided

by the Euler's number  $e$ . To calculate the width of this distribution the Gamow peak is approximated by a normal distribution with the mean value  $E_G$ :

$$\exp\left\{-\frac{b}{\sqrt{E}} - \frac{E}{kT}\right\} \approx A_{\max} \cdot \exp\left\{-\frac{1}{2}\left(\frac{E - E_G}{\sigma}\right)^2\right\}, \quad (2.16)$$

where  $A_{\max}$  defines the maximum value and  $\sigma$  the standard deviation of the normal distribution. So, the width can be calculated as  $\Delta E_G = 2\sqrt{2} \cdot \sigma$ . To get the width  $\Delta E_G$  we must compare Eq. (2.16) and the first two derivatives at the point  $E = E_G$ . If we evaluate Eq. (2.16) for  $E = E_G$  we get

$$\begin{aligned} \exp\left\{-\frac{b}{\sqrt{E_G}} - \frac{E_G}{kT}\right\} &= \exp\left\{-\frac{2E_G^{3/2}}{kT\sqrt{E_G}} - \frac{E_G}{kT}\right\} = \exp\left\{-\frac{3E_G}{kT}\right\} \\ &\stackrel{!}{=} A_{\max} \cdot \exp\left\{-4\left(\frac{E_G - E_G}{\Delta E_G}\right)^2\right\} = A_{\max}. \end{aligned}$$

Hence, the maximum value is given as  $A_{\max} = \exp\left\{-\frac{3E_G}{kt}\right\}$ . In order to simplify the calculation of the derivatives, let us make use of the following formula

$$\frac{d^n}{dx^n} \exp\{g(x)\} = \left(\sum_{k=1}^n \frac{d^k g}{dx^k}\right) \cdot \exp\{g(x)\}, \quad (2.17)$$

for each differentiable function  $g(x)$ .

Taking a closer look at Eq. (2.16), it is immediately noticeable that at  $E = E_G$  the exponential functions result in the value 1. So, for all derivatives we only must equate the sum of the derivatives of the exponent function. We will now define

$$g(E) := 4\left(\frac{E - E_G}{\Delta E_G}\right)^2.$$

By definition it is clear, that  $\frac{df}{dE}\big|_{E=E_G} = 0$  and  $\frac{dg}{dE}\big|_{E=E_G} = 0$ , because both functions have a local minimum point at  $E = E_G$ .

Now, let's calculate the second derivatives of the exponent functions  $f$  and  $g$  according to the energy  $E$  evaluated at  $E_G$ :

$$\begin{aligned} \frac{d^2 f}{dE^2}\bigg|_{E=E_G} &= \frac{d^2}{dE^2} \left\{ \frac{b}{\sqrt{E}} + \frac{E}{kT} \right\}\bigg|_{E=E_G} = \left[ \frac{3}{4} b E^{-5/2} \right]\bigg|_{E=E_G} = \frac{3}{2} \frac{1}{kT E_G}, \\ \frac{d^2 g}{dE^2}\bigg|_{E=E_G} &= \frac{d^2}{dE^2} \left\{ 4 \left( \frac{E - E_G}{\Delta E_G} \right)^2 \right\}\bigg|_{E=E_G} = \frac{8}{\Delta E_G^2}. \end{aligned}$$

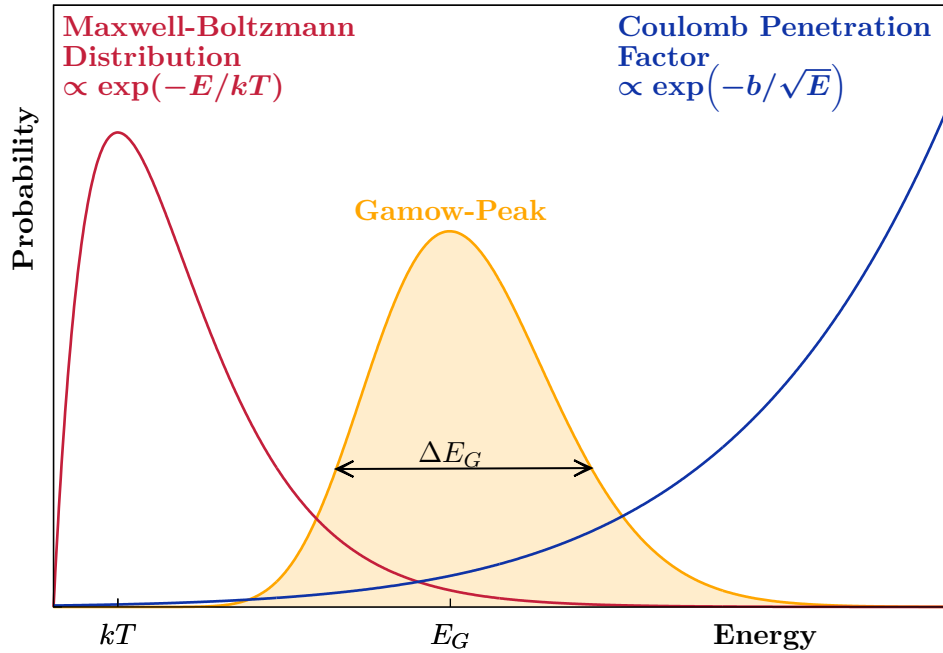
Finally, we can express the width of the Gamow peak as [Won98; Fey04]

$$\Delta E_G = 4\sqrt{\frac{E_G kT}{3}}. \quad (2.18)$$

### 2.1.4 The Hydrogen Burning Stage

This subsection will give a brief overview of the important reaction chains of the hydrogen burning stage by pointing out the most important points of the books *Introductory Nuclear Physics* [Won98, Chapter 10], *The Physics of Stars* [Phi99, Chapter 4], and *Evolution of Stars and Stellar Populations* [SC05, Chapter 5].





**Figure 2.3:** Sketch of the Gamow peak, Maxwell-Boltzmann distribution and Coulomb penetration factor. For better illustration, the Coulomb penetration factor as well as the Gamow peak are drawn several orders of magnitude larger.

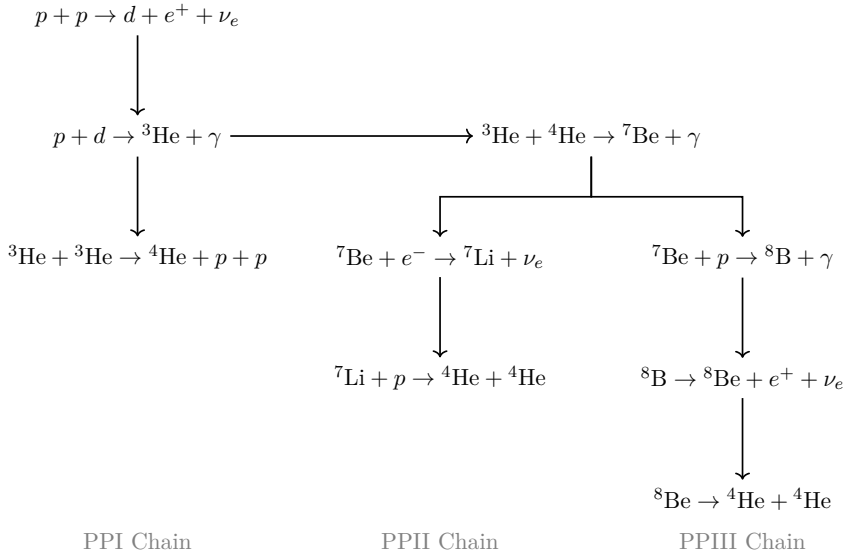
The first stage of stellar evolution is the hydrogen burning stage. In this stage the driving factor is hydrogen. The temperature raises up to  $T_9 = 0.02$ , where  $T_9$  describes the temperature in units of GK [Won98]. From the Maxwell-Boltzmann distribution Eq. (2.11) follows that the kinetic energy is  $kT \approx 2$  keV. Due to the high density and temperature within the core of a star a plasma is formed. Since the energy is significantly higher than the binding energy of the electrons, the plasma consists of protons and free electrons. The net result of this burning stage is the fusion of four protons into a  ${}^4\text{He}$  nucleus [Won98]. However, such a direct four-body reaction is extremely unlikely and the fusion process is separated into the Proton-Proton chains (PP chains) and the Carbon-Nitrogen-Oxygen cycle (CNO cycle). While the PP chain is involved in the fusion all the time, it is necessary for the CNO cycle that carbon, nitrogen or oxygen atoms are already present, as they act as catalysts [SC05].

### The Proton-Proton Chains

Let's now have a closer look at the Proton-Proton chains (PP chains). A schematic view can be found in Fig. 2.4. The first reaction of this chain is the production of deuteron through the collision of two protons



During the short timescale of this direct reaction, a proton is converted by the weak interaction into a neutron, a positron and an electron neutrino. Since a bound system with two protons is not possible and the weak interactions is involved, the probability of this reaction is very low even at stellar densities. In fact, the cross section of this reaction is so low that no experimental measurements are possible on earth [SC05; Phi99]. For a star with central density  $\rho = 10^5$  kg m $^{-3}$ , temperature  $T_9 = 0.015$  and hydrogen mass fraction  $X = 0.5$ , the reaction rate in the center of the star is  $\lambda = 5 \times 10^{13}$  s $^{-1}$  m $^{-1}$ . This means that a proton in



**Figure 2.4:** Schematic view of the Proton-Proton I-III chains. As a net result each of the three chains combines four protons to one  ${}^4\text{He}$ .

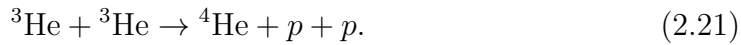
the center has to stay on average about  $9 \times 10^9$  years before it is converted to deuterium by the reaction (2.19). This illustrates how slow this process proceeds and that this reaction is the bottleneck in hydrogen burning and sets the time scale for this phase in a star. All further reactions in the PP chains proceeds much faster. [Phi99].

The second reaction in the PP chain is the fusion of a proton and a deuteron to  ${}^3\text{He}$



Since this is an electromagnetic reaction the cross section is in the order of  $\mathcal{O}(10^{18})$  greater than the cross section of the weak reaction Eq. (2.19). Thus, the deuteron is almost immediately converted to  ${}^3\text{He}$  by the reaction Eq. (2.20) [Phi99].

At this point multiple possible reactions can be processed to fuse  ${}^3\text{He}$  to higher mass elements. One possibility is that an  ${}^3\text{He}$  nucleus may tunnel through the Coulomb barrier of another one to form  ${}^4\text{He}$



The reaction chain of the three reactions Eqs. (2.19) to (2.21) is known as the PPI chain. On the left-hand side of this reaction chain a total of six protons are involved while on the right-hand side one  ${}^4\text{He}$ , two protons, two positrons, two electron neutrinos, and two  $\gamma$ -rays are produced [Won98]. Using the Einstein relation  $E = mc^2$  the energy released by this reaction chain can be calculated easily as 26.731 MeV [SC05].

The presence of  ${}^4\text{He}$  opens up another possible reaction for the fusion of  ${}^3\text{He}$



Due to the high electron density within the core of a star, the newly formed beryllium nuclei can be converted to lithium by the weak interaction



At the end of the PPII chain, the fusion of lithium and a proton results in two helium nuclei [Won98].



Even in the PPII chain, where  ${}^4\text{He}$  act as catalyst, a total of four protons are converted into one  ${}^4\text{He}$  nucleus [Phi99].

The PPIII chain is also connected to the production of  ${}^7\text{Be}$  from helium Eq. (2.22). But instead of an electron capture, the beryllium can fuse with a proton to form boron [Won98]



While  ${}^8\text{B}$  is unstable, it decays via  $e^+$ -emission into  ${}^8\text{Be}$



As  ${}^8\text{Be}$  isn't stable, it decays through  $\alpha$ -emission

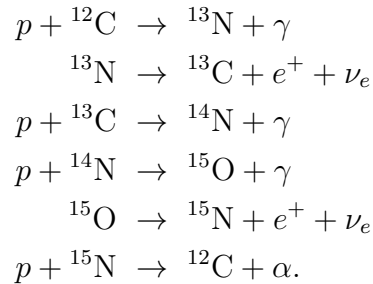


The PPIII chain also produces one  ${}^4\text{He}$  out of four protons in total.

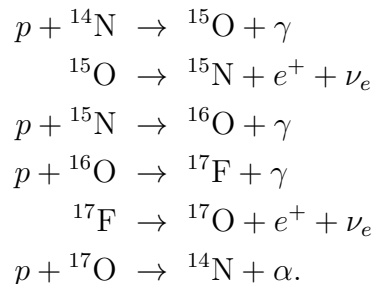
### The Carbon-Nitrogen-Oxygen Cycle

As discussed previously, a further possibility to convert protons into  ${}^4\text{He}$  is the Carbon-Nitrogen-Oxygen cycle (CNO cycle). The CNO cycle mainly consists of two independent cycles, the CN and the NO cycle. To start the reactions within these cycles the presence of the elements C, N, and O are essential as they act as catalysts [SC05]. Both cycles consists of three main reactions. The first reaction ( $p, \gamma$ ) is the fusion of a nucleus with a proton to another nucleus. The second reaction is the  $\beta^+$ -decay of a nucleus, and the third one is the  $\alpha$ -emission reaction.

The reactions involved in the CN cycle are [Won98; SC05]

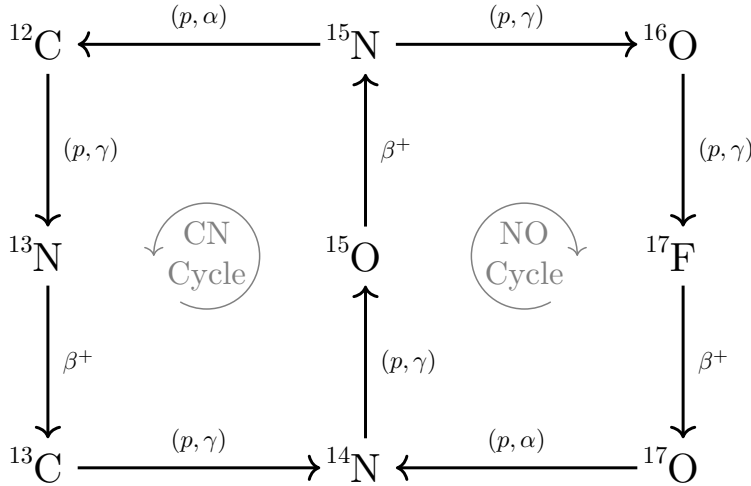


While carbon and nitrogen act as catalysts in the CN cycle, the catalysts in the NO cycle are nitrogen and oxygen. The reactions involved in the NO-cycle are [Won98; SC05]:



The result of both cycles are similar to the result of the PP-Chains: four protons are converted into one  $^4\text{He}$ , two positrons, two electron neutrinos, and three  $\gamma$ -particles. The two reactions are connected via the nucleus  $^{14}\text{N}$ ,  $^{15}\text{O}$  and  $^{15}\text{N}$ . Figure 2.5 shows a schematic view of these two cycles. There are other side cycles that we will not discuss in this work. A more detailed overview can be found in *Introductory Nuclear Physics* [Won98], for example.

Although the CNO cycle was predicted by Bethe and Weizsäcker already in the 1930s, the first experimental evidence was obtained in 2020 by the Boroxino experiment [Ago+20].



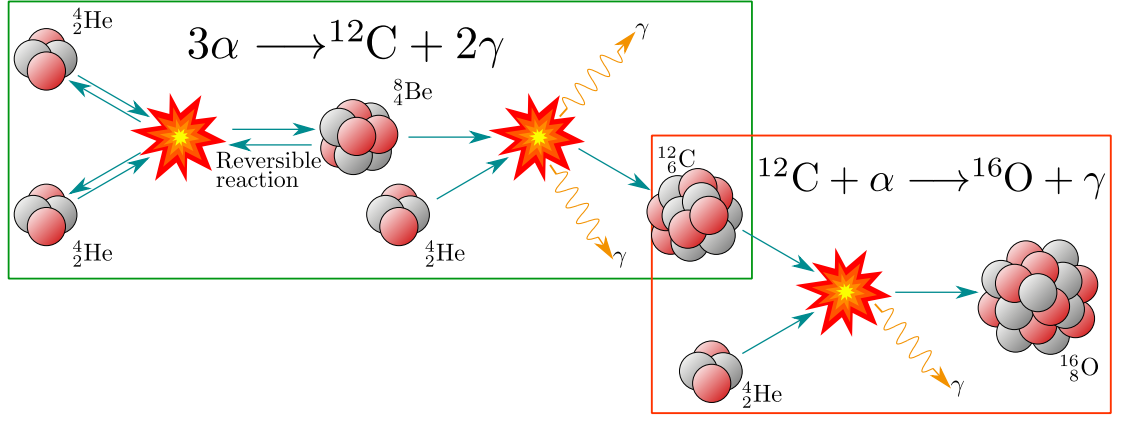
**Figure 2.5:** Schematic view of the CNO cycle which consists of the two independent cycles, the CN and the NO cycle. The elements Carbon, Nitrogen and Oxygen act as catalysts and their presence is essential for the two cycles. In total four protons are converted into one  $^4\text{He}$  nucleus, two positrons, two electron neutrinos, and three  $\gamma$ -particles. [Ago+20; SC05; Phi99; Won98]

### 2.1.5 The Helium Burning Stage

This subsection will give a short overview of the helium burning stage by summarizing the key messages of the books *Introductory Nuclear Physics* [Won98, Chapter 10-5], *The Physics of Stars* [Phi99, Chapter 4], and *Evolution of Stars and Stellar Populations* [SC05, Chapter 6].

When most of the hydrogen within the core of a star is fused to helium, the reaction rate of the PP chains and the CNO cycle decreases. Since the temperature is not sufficient to drive fusion reactions to produce higher mass elements, the hydrostatic equilibrium cannot be maintained, whereby gravity begins to compress the core of the star. This increases the central temperature up to  $T_9 = 0.2$ , at which the helium burning takes place [Sal52]. The remaining hydrogen continues the process of forming helium in the outer / cooler layers of the star, but with a lower reaction rate due to the lower density [Won98].

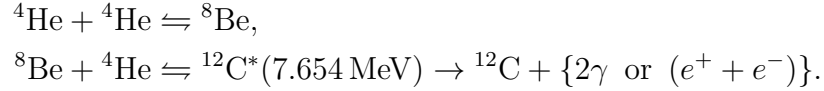
The helium burning stage consists of two main reactions. The first fundamental one is the fusion of three  $^4\text{He}$  nuclei to  $^{12}\text{C}$ . This is called the triple- $\alpha$  reaction. The second reaction is the production of higher mass nuclei via the  $\alpha$ -capture reaction. The triple- $\alpha$  process and the most important  $\alpha$ -capture reaction, the production of  $^{16}\text{O}$ , are shown schematically in Fig. 2.6.



**Figure 2.6:** Reaction scheme of the two most important reactions of the helium burning stage. The triple- $\alpha$  reaction is shown in the green rectangle, where two helium particles fuse to beryllium, which fuses to carbon through another  $\alpha$ -capture. The  $\alpha$ -capture reaction of  ${}^{12}\text{C}$  is shown in the red rectangle.

### Triple- $\alpha$ Reaction

In the triple- $\alpha$  reaction, three  ${}^4\text{He}$  nuclei fuse to one  ${}^{12}\text{C}$ . A simultaneous collision of three helium atoms is very unlikely, so the triple- $\alpha$  process usually takes place in two steps [FF14; Phi99; Sal52]:



The ground state of  ${}^8\text{Be}$  is 91.84 keV above the  $2\alpha$  threshold (see Fig. 2.7). The production of  ${}^8\text{Be}$  starts at a temperature at around  $T_9 = 0.12$ , because then the Gamow energy  $E_G = 92.3 \text{ keV}$  with a width of  $\Delta E_G = 72.1 \text{ keV}$  is sufficient to drive this fusion reaction (see also Table B.6) [FF14; Phi99; Sal52].

Since the ground state of  ${}^8\text{Be}$  is above the  $2\alpha$  threshold, it is unstable. It decays with a width of  $\Gamma = 5.57 \text{ eV}$  (see also Table B.1) back to  $2\alpha$ . However, the ground state of  ${}^8\text{Be}$  is a metastable state, which means that its lifetime is several orders of magnitude longer than that of other states which decay by the strong interaction ( $\sim 10^{-22} \text{ s}$ ) [BW91]. In fact, its half-life of

$$\tau_{1/2} = \frac{\hbar \ln(2)}{\Gamma} = 8.19 \times 10^{-17} \text{ s}$$

is a factor  $\sim 10^5$  larger than that of the nearest excited states (see Table B.1).

This leads to the fact that in the core of a star a state of dynamic equilibrium of  ${}^4\text{He}$  and  ${}^8\text{Be}$  is formed. The number density ratio can be calculated using the formula for the population of  ${}^8\text{Be}$  nuclei in a gas of  ${}^4\text{He}$  nuclei at temperature  $T$  [Phi99]

$$n_{s_{\text{Be}}} = \left[ \frac{2\pi\hbar M_{s_{\text{Be}}}}{M_\alpha kT} \right]^{\frac{3}{2}} \exp\left[ -\frac{(M_{s_{\text{Be}}} - 2M_\alpha)c^2}{kT} \right] \cdot (n_{4_{\text{He}}})^2,$$

with  $n_{4_{\text{He}}}$ ,  $n_{s_{\text{Be}}}$  the number density of  ${}^4\text{He}$ ,  ${}^8\text{Be}$  and  $M_\alpha$ ,  $M_{s_{\text{Be}}}$  the nuclear masses of  ${}^4\text{He}$ ,  ${}^8\text{Be}$ . For a central density of  $\rho = 10^8 \text{ kg m}^{-3}$ , the  ${}^4\text{He}$  number density is  $n_{4_{\text{He}}} = \frac{\rho}{M_\alpha} = 1.5 \times 10^{34} \text{ m}^{-3}$  for example. At  $T_9 = 0.12$  the  ${}^8\text{Be}$  number density is  $n_{s_{\text{Be}}} = 4.37 \times 10^{25} \text{ m}^{-3}$  giving a ratio of about  ${}^8\text{Be} : {}^4\text{He} = 1 : 3 \cdot 10^8$ . However, at  $T_9 = 0.2$  the  ${}^8\text{Be}$  number density increases to  $n_{s_{\text{Be}}} = 7.16 \times 10^{26} \text{ m}^{-3}$ , resulting in

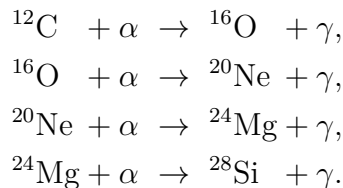
a ratio of about  $^8\text{Be} : ^4\text{He} = 1 : 2 \cdot 10^7$ . Although the  $^8\text{Be}$  number density is very low, its presence opens up the next stage of the triple- $\alpha$  process [Phi99].

The carbon production is strongly enhanced by the core structure of carbon. The excited state  $0^+$  at 7.654 MeV of carbon, known as the Hoyle state, lies 287 keV above the threshold of  $^8\text{Be} + \alpha$  and furthermore in the Gamow window (see Fig. 2.7). The value of the Gamow peak for  $T_9 = 0.2$  is  $E_G = 232$  keV with the width  $\Delta E_G = 146$  keV (see also Table B.6). This boosts the capture process by a factor of 10–100 million and as a result, most of the carbon in the core of a star is produced through the Hoyle state [FF14].

The Hoyle state seems to form a spatial structure, where three  $\alpha$ -cluster are arranged nearly linearly [Epe+11; Hjo11]. It has a finite lifetime with a half-life of  $\tau_{1/2} \approx 5 \times 10^{-17}$  s and decays in more than 99% of all cases into three  $\alpha$  (see also Table B.2). Only in one of 2403.85 cases the Hoyle state transits to the ground state of  $^{12}\text{C}$  via either two-gamma-decay ( $0^+ \rightarrow 2^+ \rightarrow 0^+$ ) or  $e^+e^-$ -pair production ( $0^+ \rightarrow 0^+$ ) [FF14]. The formation of  $^{12}\text{C}$  via the Hoyle state is crucial for nuclear astrophysics and the production of higher elements [FF14; Won98].

### $\alpha$ -Capture Reaction

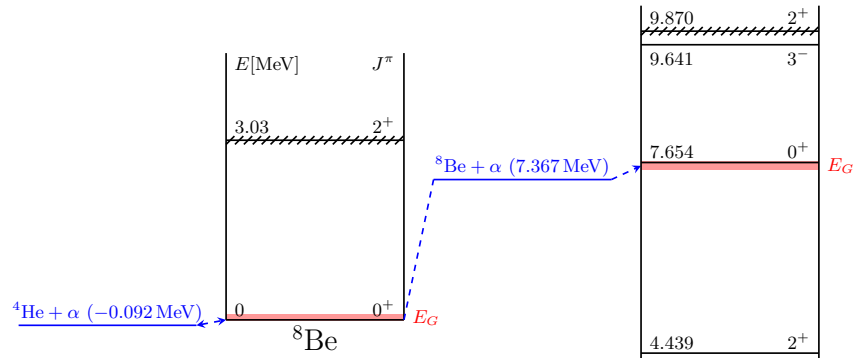
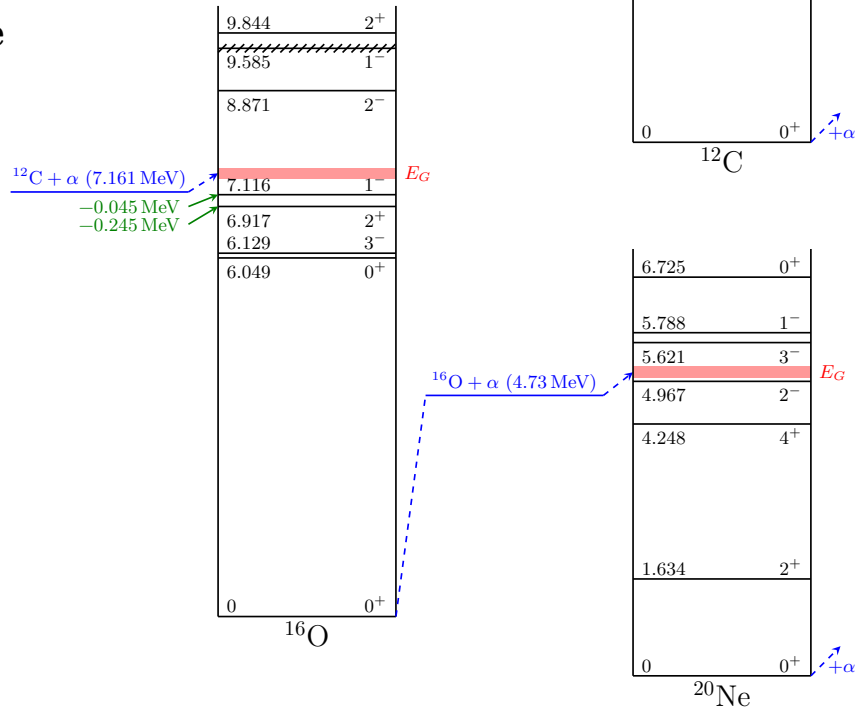
In the helium burning stage, higher mass elements are formed by the  $\alpha$ -capture reaction. The temperature for this reaction is around  $T_9 = 0.2$  [SC05; Won98]. The following  $\alpha$ -capture reactions take place in the helium burning stage [SC05]:



All of these reactions are compound reactions. Oxygen is mainly produced through the two subthreshold states  $1^-$  at 7.116 MeV and  $2^+$  at 6.917 MeV (see Fig. 2.7). The cross section for this reaction at the energy of the Gamow peak  $E_G \approx 300$  keV is in the order of  $10^{-17}$  barn. We will discuss this reaction in more detail in the next section.

With an increasing proton number  $Z$  the Coulomb penetration factor is decreasing exponentially. For example, at the cms energy  $E_{\text{cms}} = 300$  keV the Coulomb penetration factor for the reaction  $^{16}\text{O}(\alpha, \gamma)^{20}\text{Ne}$  is about  $\mathcal{O}(10^{-23})$ , which is a factor  $\sim 10^6$  smaller than for the reaction of  $^{12}\text{C}(\alpha, \gamma)^{16}\text{O}$  (see Table B.7). Furthermore, at  $T_9 = 0.2$  the Gamow window of the reaction  $^{16}\text{O}(\alpha, \gamma)^{20}\text{Ne}$  is close to the  $2^-$  state at 4.947 MeV of  $^{20}\text{Ne}$  (see Fig. 2.7). Since the initial particles of the reactions are in the ground state  $0^+$  this state is of unnatural parity. The term of (un)natural parity is further discussed in Section 2.2. As a consequence, the production of  $^{20}\text{Ne}$  is highly suppressed in the helium burning stage [Won98, p. 379]. Since the Coulomb penetration factor decreases exponentially with increasing charge number the production of magnesium and silicon is also highly suppressed in the helium burning stage (see also Table B.7).

Eventually, the triple- $\alpha$  process and the  $\alpha$ -capture of Carbon play an important role in the helium burning stage, while at  $T_9 = 0.2$  other fusion reactions can be neglected since they proceed much slower. Therefore a mixture of  $^{12}\text{C}$ ,  $^{16}\text{O}$  and only traces of  $^{20}\text{Ne}$  are formed [SC05]. One of the most important reactions in the stellar evolution is the reaction of  $^{12}\text{C}(\alpha, \gamma)^{16}\text{O}$ , since the production of elements

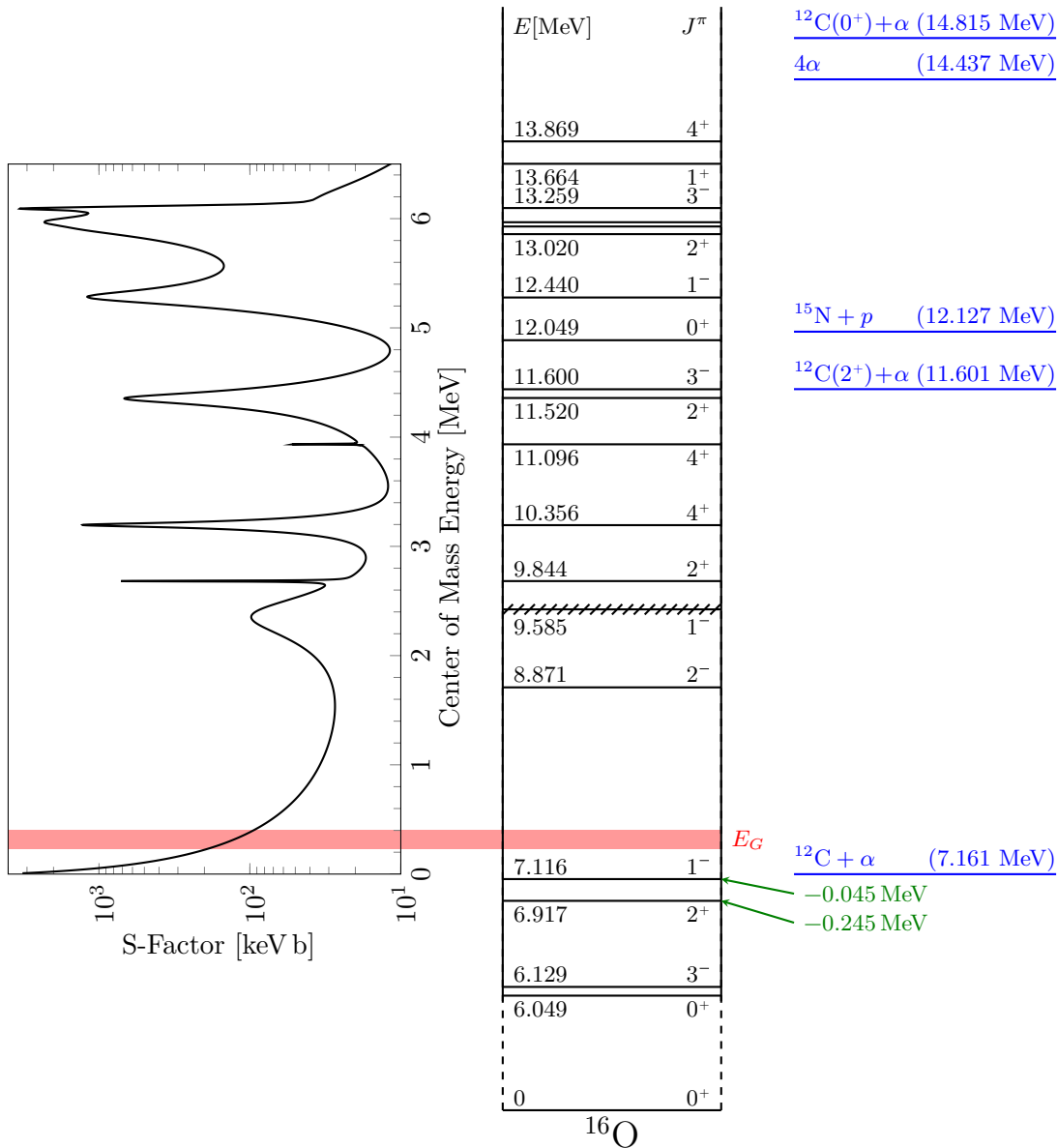
Triple- $\alpha$  $\alpha$ -Capture

**Figure 2.7:** Level scheme of the nuclei involved in the helium burning stage. The blue line shows the invariant mass of the initial particles in relation to the level scheme of the produced particle. The red bar shows the Gamow window of each reaction at  $T_9 = 0.2$ . In the triple- $\alpha$  reaction the carbon nucleus is produced through the  $0^+$  resonance at  $E = 7.654$  MeV. In the further  $\alpha$ -capture reactions the most important one is the  $\alpha$ -capture of carbon. The oxygen is formed via the two subthreshold resonances with  $E_{1^-}^{cms} = -45$  keV and  $E_{2^+}^{cms} = -245$  keV (green lines). In the helium burning stage with  $T_9 = 0.2$  the  $\alpha$ -capture reaction of oxygen can be neglected. On the one hand the Coulomb penetration factor is several orders of magnitude lower, and on the other hand the reaction is suppressed by the unnatural parity of the  $2^-$  state at  $E = 4.967$  MeV close to the Gamow window. All data can be found in the Tables B.1 to B.6.

in further burning stages is sensitive to the C/O abundance ratio. This is often referred as the holy grail in nuclear astrophysics. Many stellar models use the S-factor of this reaction as an input parameter. So, the study of the reaction  ${}^{12}\text{C}(\alpha, \gamma){}^{16}\text{O}$  plays a crucial role in understanding the evolution of stars.

## 2.2 The $\alpha$ -Capture Process of $^{12}\text{C}$ in Detail

In this section we will discuss the astrophysical reaction  $^{12}\text{C}(\alpha, \gamma)^{16}\text{O}$  in more detail. Hereby we will focus on the theoretical part, a discussion on the results of previous measurements can be found in Section 2.3. For an even more detailed description please refer to the papers [DB74; Bru01; BSS14; deB+17; FDM19]. All S-factor values used in this thesis were determined with the AZURE2 software [Ud15; Azu+10] using the configuration parameters given in the supplementary material from [deB+17]. These parameters represent the results of analysis of previous data from [deB+17]. AZURE2 can be used to calculate differential and angle-integrated cross sections for resonant nuclear reactions relevant to nuclear



**Figure 2.8:** Level diagram of the  $^{16}\text{O}$  nucleus. The corresponding S-factor for the reaction  $^{12}\text{C}(\alpha, \gamma)^{16}\text{O}$  is shown on the left, calculated with the AZURE2 software [Ud15] configured with the parameters of the supplemental material provided by Ref. [deB+17]. The S-factor strongly depends on the nuclear structure of  $^{16}\text{O}$ . Different thresholds are shown in blue on the right side. The two levels a few keV below the carbon- $\alpha$  threshold play an important role as they have the greatest impact in the area of the Gamow peak, shown in red.



astrophysics. The open-source code AZURE2 is based on an R-matrix code that can also be used for data fitting [Ud15; Azu+10].

The S-factor, Eq. (2.10), is characterized by the resonances, the non-resonant contribution and the interference between these components [deB+17]. The contribution strength of all these parts strongly depends on the nuclear structure of  $^{16}\text{O}$ . Figure 2.8 shows the level structure of oxygen and the corresponding S-factor of the reaction  $^{12}\text{C}(\alpha, \gamma)^{16}\text{O}$ . Since  $^{16}\text{O}$  is an even-even nucleus the ground state has quantum numbers  $J^\pi = 0^+$ , with total spin  $J$  and parity  $\pi$ . Below the threshold of  $^{12}\text{C} + \alpha$  at 7.161 MeV, four bound states are located at  $0^+$  (6.049 MeV),  $3^-$  (6.129 MeV),  $2^+$  (6.917 MeV), and  $1^-$  (7.116 MeV). The two states  $1^-$  and  $2^+$  are only a few keV below the threshold and contribute the most to the cross section in the area of the Gamow peak. Therefore they play a decisive role in this reaction [deB+17].

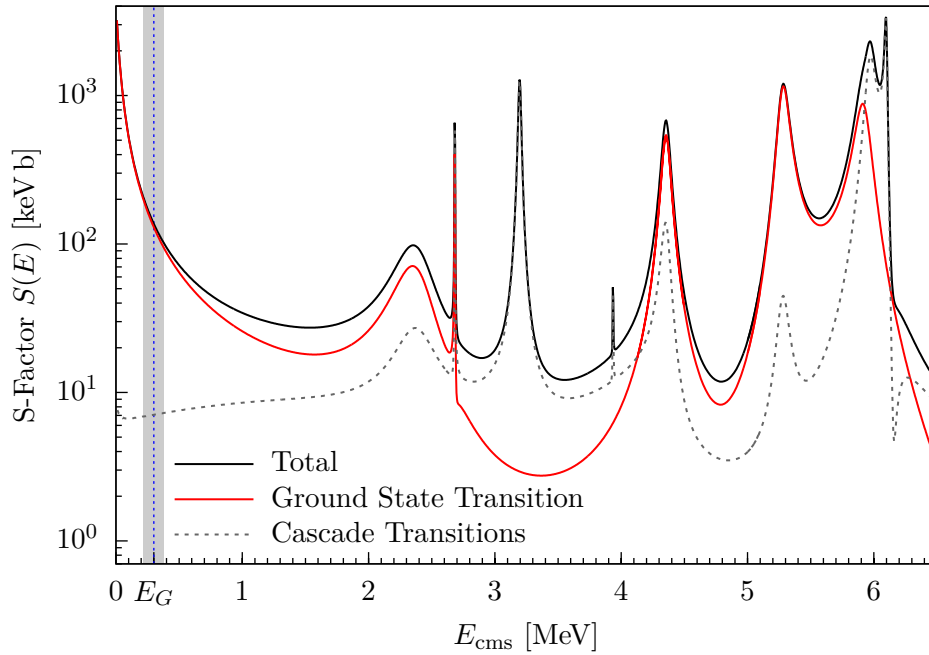
Oxygen is produced from the fusion of  $^{12}\text{C}$  and  $\alpha$ , which are also  $0^+$  nuclei. Due to the conservation of parity only states with a total spin of  $J = l$  and parity of  $\pi = (-1)^l$  can arise, where  $l$  denotes the relative orbital angular momentum. In addition, only electric transitions to the ground state can occur as described by the  $\gamma$ -ray decay selection rule. Furthermore, transitions from  $0^+$  to  $0^+$  states are prohibited [deB+17]. This can also be seen in Fig. 2.8, so for example the excited state  $0^+$  (12.049 MeV) does not contribute to the S-factor. At this point it should be mentioned that nearly 100% of all excited states decay to the ground state via  $\gamma$ -decay (see Table B.3). Exceptions are, for example the states  $4^+$  (10.356 MeV) and  $4^+$  (11.096 MeV), which can only decay into the states  $2^+$  (6.917 MeV),  $3^-$  (6.129 MeV). Any transition that passes through one or more intermediate states to the ground state is called a cascade transition. This explains that the largest contribution to the S-factor in the low-energy range near the Gamow peak is made by the ground state transition, see Fig. 2.9.

Since  $^{16}\text{O}$  is a mirror-symmetric nucleus, the E1-decay into the ground state is strictly isospin forbidden [BSS14]. The Coulomb interaction breaks the isospin symmetry, so that such a transition is possible, but the strength is strongly reduced [deB+17]. As a result, the contribution of the electric multipolar components E1 and E2 to the total S-factor for low energy levels is of similar importance, see Fig. 2.10. To summarize we can state that the oxygen is produced through the two subthreshold states  $1^-$  at 7.116 MeV and  $2^+$  at 6.917 MeV. The E1 and E2 contribution dominate the total S-factor in the area of the Gamow peak while the contribution of the cascade transitions is small compared to the ground state transition in this region.

Following the previous discussion, we can limit ourselves to the E1 and E2 multiplicities when calculating the differential cross section Eq. (2.4) for low energies. The differential cross section in the center of mass frame of carbon- $\alpha$  as a function of the cms energy  $E$  and photon angle  $\theta_\gamma$  is therefore given as [DB74; deB+17]

$$\begin{aligned} \left[ \frac{d\sigma}{d\Omega_\gamma^{\text{cms}}} \right]_{(\alpha, \gamma)}(E, \theta_\gamma) &= \frac{\sigma_{\text{E1}}}{4\pi} [Q_0 P_0(\cos \theta_\gamma) - Q_2 P_2(\cos \theta_\gamma)] \\ &+ \frac{\sigma_{\text{E2}}}{4\pi} \left[ Q_0 P_0(\cos \theta_\gamma) + \frac{5}{7} Q_2 P_2(\cos \theta_\gamma) - \frac{12}{7} Q_4 P_4(\cos \theta_\gamma) \right] \\ &+ \frac{6}{\sqrt{5}} \cos(\phi_{12}(E)) \frac{\sqrt{\sigma_{\text{E1}} \sigma_{\text{E2}}}}{4\pi} [Q_1 P_1(\cos \theta_\gamma) - Q_3 P_3(\cos \theta_\gamma)], \end{aligned} \quad (2.28)$$

where  $P_n(\cos \theta_\gamma)$  are the Legendre polynomials,  $Q_n$  the geometric attenuation



**Figure 2.9:** The total S-factor of the reaction  $^{12}\text{C}(\alpha, \gamma)^{16}\text{O}$  compared to the ground state transition and the sum of all cascade transitions. The curves are calculated with the AZURE2 software [Ud15] configured with the parameters of the supplemental material provided by Ref. [deB+17]. The ground state transition dominates the total S-factor near the Gamow peak, while the sum of all cascade transitions makes a small contribution in this region. The Gamow peak is indicated by the blue dashed line and the Gamow width by the gray bar on the left hand side.

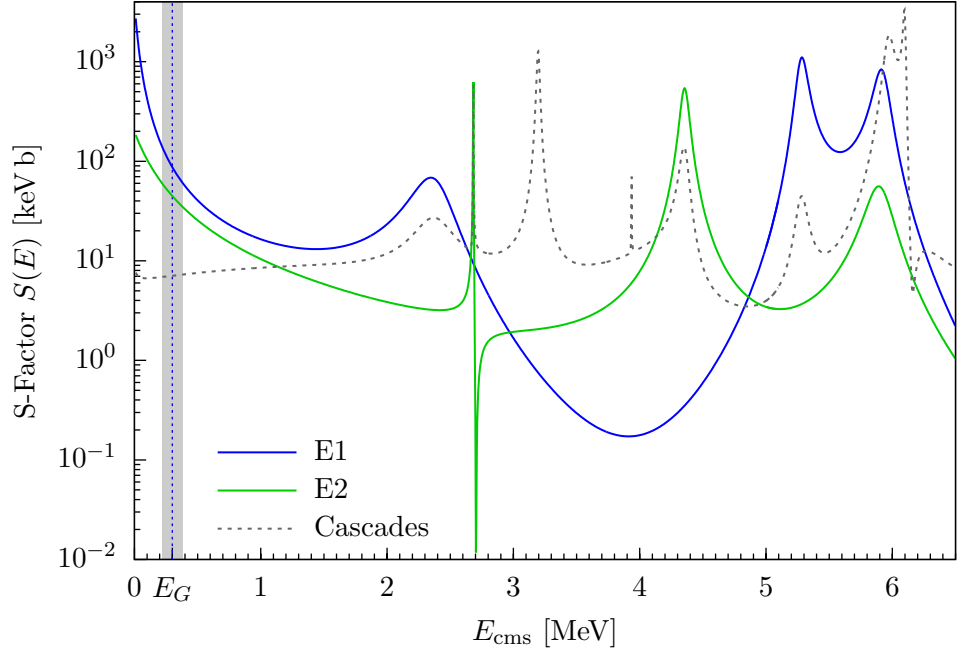
factors (see [Ros53]), and  $\phi_{12}$  the Coulomb phase shift difference between the E1 and E2 transition matrix elements. The phase shift difference can be written as [DB74; deB+17]

$$\cos(\phi_{12}) = \cos\left(\delta_{\alpha 1} - \delta_{\alpha 2} - \arctan\left(\frac{\eta}{2}\right)\right), \quad (2.29)$$

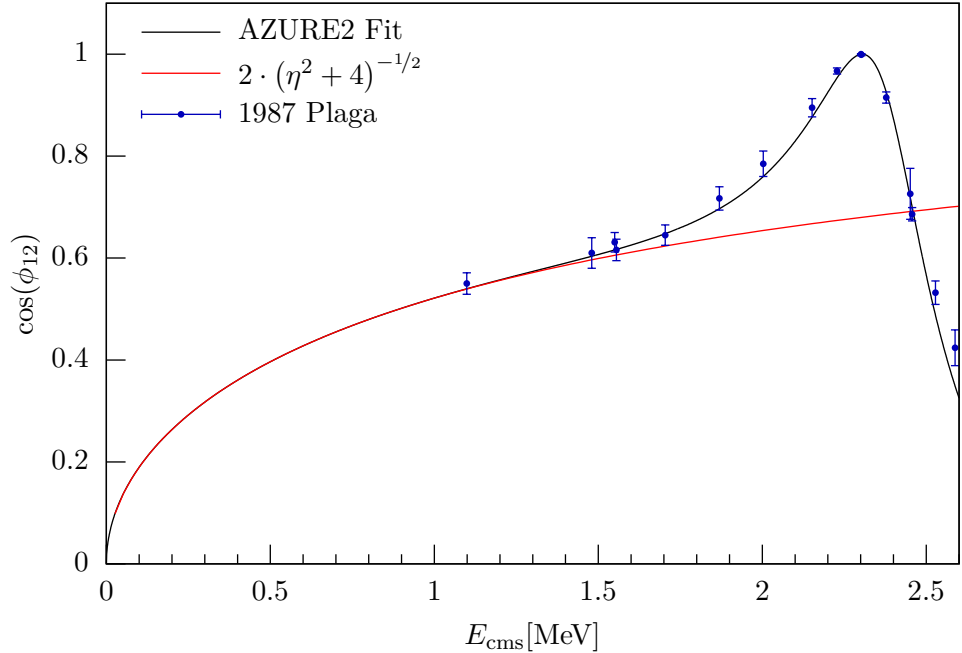
where  $\delta_{\alpha l/2}$  are the elastic scattering phase shifts for angular momentum  $l = 1$  and  $l = 2$ , respectively, and  $\eta$  the Sommerfeld number, see Eq. (2.9). For cms energies  $E_{\text{cms}} \leq 1.5$  MeV we can approximate the phase shift to [Bru01]

$$\cos(\phi_{12}) = \frac{2}{\sqrt{\eta^2 + 4}}. \quad (2.30)$$

The Coulomb phase shift and the approximated one can be seen in Fig. 2.11.



**Figure 2.10:** The  $E1$  and  $E2$  ground state contribution of the  $S$ -factor of the reaction  $^{12}\text{C}(\alpha, \gamma)^{16}\text{O}$  compared to the contribution of cascade transitions. The curves are calculated with the AZURE2 software [Ud15] configured with the parameters of the supplemental material provided by Ref. [deB+17]. The contributions of  $E1$  and  $E2$  to the  $S$ -factor near the Gamow peak are in the same order of magnitude. The Gamow peak is indicated by the dashed blue line and the Gamow width by the gray bar on the left hand side.



**Figure 2.11:** For  $cms$  energies  $E_{cms} \leq 1.5$  MeV the Coulomb phase shift  $\cos(\phi_{12})$  can be approximated using Eq. (2.30). The black line represents the results from the  $R$ -Matrix fit using the AZURE2 software [Ud15] configured with the results of Ref. [deB+17] provided as supplemental material. The red line represents the approximation of the Coulomb phase shift using Eq. (2.30). The blue dots represent the values of the Coulomb phase shift determined from the elastic  $\alpha$ -scattering  $^{12}\text{C}(\alpha, \alpha)^{12}\text{C}$  data from [Pla+87].

## 2.3 Discussion of Measurement Techniques

Experiments to determine the total S-factor of the reaction  $^{12}\text{C}(\alpha, \gamma)^{16}\text{O}$  have been carried out since 1955. The cross section at the temperature of the helium burning stage ( $E_G \approx 300$  keV) is around  $\mathcal{O}(10^{-17}$  b). So, measurements are really challenging. There is currently no experiment which has measured the total cross section below  $E_{\text{cms}} = 1$  MeV (see Fig. 2.14). For low energies the Coulomb penetration factor dominates the cross section and thus it decreases exponentially.

In the low energy range, the cross section falls by 6 orders of magnitude from  $\sigma \sim 4 \times 10^{-11}$  b at  $E_{\text{cms}} = 1$  MeV to  $\sigma \sim 2 \times 10^{-17}$  b at  $E_{\text{cms}} = 0.3$  MeV (see the black curve in Fig. 2.13). In order to determine the S-factor at the Gamow peak, it is necessary to extrapolate the results of the measurements at higher energies to lower energies. But the potential extrapolation errors must be taken into account. It is therefore essential to determine the S-factor as precisely as possible in order to reduce these errors. As described in Section 2.2, the cross section is influenced by the nuclear structure of  $^{16}\text{O}$ . In particular, the cross section contains resonances and interferences between these resonances. In order to extrapolate it to lower energies, a method taking all of these components into account must be used. These calculations require a high computational effort, as there are  $2^{n-1}$  possible interferences for  $n$  involved states. An appropriate method for such a calculation is the R-matrix method, which is explained in detail in [deB+17]. To be able to use this method, each multipolarity must be separated. Therefore it is important to measure the angular distribution of the emitted photons during the experiment. The E1 and E2 components of the ground state transition can then be determined using Eq. (2.28). As described in Section 2.2, these two multiplicities dominate the cross section for small energies. At higher energies, contributions from cascade transitions and further multiplicities also become relevant. At this point we also have to mention, that the determination of the E2 contribution in the range 1 to 3 MeV is challenging since the total cross section is dominated by the broad  $1^-(9.585$  MeV) resonance (see Figs. 2.8 and 2.10).

In this section we will discuss some experiments in more detail. At this point we do not provide a complete list of all previous experiments, a detailed description can be found in [deB+17]. In the following we will focus on some of the most important experiments. We can roughly divide the experiments into direct and indirect methods. In the direct method, the reaction  $^{12}\text{C}(\alpha, \gamma)^{16}\text{O}$  is studied. Whereas the indirect methods derives the cross section from the measurements of other nuclear reactions based on theoretical models.

### 2.3.1 Direct Measurement Techniques

The basic idea for the direct measurement of the reaction  $^{12}\text{C}(\alpha, \gamma)^{16}\text{O}$  is to carry out  $\alpha$ -carbon scattering experiments and to detect the recoil particles. The S-factor can be determined from this measurement data. There are two possible settings: the first uses an  $\alpha$ -beam that scatters on a solid carbon target, the second is employing the inverse kinematics where a  $^{12}\text{C}$  beam scatters on a  $^4\text{He}$  gas target. Both methods have their individual advantages and disadvantages, which we will discuss below. With direct measurements, it is almost impossible to measure in the Gamow energy range. The cross section must therefore be measured at higher energies and extrapolated to the Gamow peak [Fey04].

### $\alpha$ -Particle Scattering on Carbon

The first method to measure the S-factor of the reaction  $^{12}\text{C}(\alpha, \gamma)^{16}\text{O}$  is the  $\alpha$ -particle scattering experiment at a solid carbon target. Hereby the emitted photon will be detected to determine the angular distribution, in order to separate the E1 and E2 contributions. In most cases a NaI-detector array is used to detect the emitted photons. A major problem is contamination of the carbon target. Traces of  $^{13}\text{C}$  atoms in the target material generate a high neutron background through the reaction  $^{13}\text{C}(\alpha, n)^{16}\text{O}$ . It should be mentioned that the cross section of reaction  $^{12}\text{C}(\alpha, \gamma)^{16}\text{O}$  at the peak of the  $1^-$  (9.585 MeV) resonance is more than 6 orders of magnitude smaller than the cross section of the reaction  $^{13}\text{C}(\alpha, n)^{16}\text{O}$ . This is discussed in detail in [deB+17]. This large background made it difficult to determine the S-factor for a long time. Higher purity targets and better methods of extracting the background were developed that ultimately made the determination of cascade transitions possible.

The analysis of the data is challenging. As already mentioned, the E1 and E2 contributions are in the same order of magnitude. Equation (2.28) shows how the two components contribute to the differential cross section. Figure 2.12 shows the contribution factors of E1 and E2 in relation to this formula. It is immediately noticeable that the E1 component can be isolated at  $90^\circ$ , where it is maximal, while the E2 component vanishes. Thus, E1 can be determined by measuring the cross section at  $\theta_\gamma = 90^\circ$ . It then applies

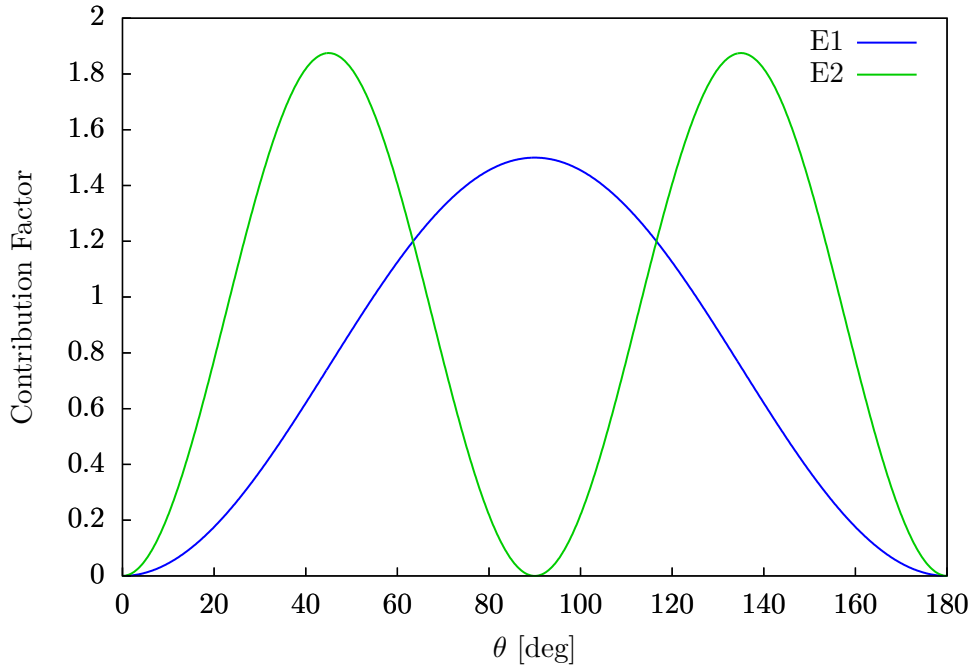
$$\sigma_{\text{E1}} = 4\pi \left(\frac{2}{3}\right) \left(\frac{d\sigma}{d\Omega}\right)_{\theta_\gamma=90^\circ}.$$

Unfortunately, E2 cannot be isolated. Therefore a different method must be used to determine E2. Intuitively, one would fit the data with Eq. (2.28). However, in practise it shows, that this is quite difficult, since the correlation between  $\sigma_{\text{E1}}$  and  $\sigma_{\text{E2}}$  is large due to the interference term  $\sqrt{\sigma_{\text{E1}}\sigma_{\text{E2}}}$ . It turns out that it is better to use a modified formula to fit the data [Kun+02; DB74]:

$$\begin{aligned} \left[\frac{d\sigma}{d\Omega_\gamma^{\text{cms}}}\right]_{(\alpha,\gamma)}(E, \theta_\gamma) = \frac{\sigma_{\text{E1}}}{4\pi} & \left\{ Q_0 P_0(\cos \theta_\gamma) - Q_2 P_2(\cos \theta_\gamma) \right. \\ & + \frac{\sigma_{\text{E2}}}{\sigma_{\text{E1}}} \left[ Q_0 P_0(\cos \theta_\gamma) + \frac{5}{7} Q_2 P_2(\cos \theta_\gamma) - \frac{12}{7} Q_4 P_4(\cos \theta_\gamma) \right] \\ & \left. + \frac{6}{\sqrt{5}} \cos(\phi_{12}(E)) \sqrt{\frac{\sigma_{\text{E2}}}{\sigma_{\text{E1}}}} \cdot [Q_1 P_1(\cos \theta_\gamma) - Q_3 P_3(\cos \theta_\gamma)] \right\}. \end{aligned} \quad (2.31)$$

In this formula, the parameters are given by the amplitude  $\sigma_{\text{E1}}$ , the ratio  $\frac{\sigma_{\text{E2}}}{\sigma_{\text{E1}}}$  and the phase difference  $\phi_{12}$ . There are two options to fit the data. The first option is to perform a three-parameter fit. Since the phase difference can also be determined via the elastic scattering  $^{12}\text{C}(\alpha, \alpha)^{12}\text{C}$ , this parameter can also be used as an input for the fit. The second option is to use a two-parameter fit including a fixed-phase. To do this, one determines  $\sigma_{\text{E1}}$  and the ratio  $\frac{\sigma_{\text{E2}}}{\sigma_{\text{E1}}}$  and can hence determine  $\sigma_{\text{E2}}$ . However, due to the error propagation, we expect a lower accuracy for E2 [deB+17; Ass+06].

There are several measurements planned to determine the cross section of the reaction  $^{12}\text{C}(\alpha, \gamma)^{16}\text{O}$ . Two of these experiments are for example the LUNA-MV experiment at the Gran Sasso laboratory and an experiment at the Felsenkeller



**Figure 2.12:** *Theoretical angular contribution of E1 and E2 for the ground state transition of  $^{12}\text{C}(\alpha, \gamma)^{16}\text{O}$ . The contribution is calculated in relation to Eq. (2.28), so that E1 represents  $P_0(\cos \theta) - P_2(\cos \theta)$  and E2 represents  $P_0(\cos \theta) + \frac{5}{7}P_2(\cos \theta) - \frac{12}{7}P_4(\cos \theta)$ . At  $90^\circ$ , the E1 contribution can be isolated from the data, since it is at its maximum there, while the E2 contribution vanishes. Since there is no angle at which E2 can be perfectly isolated, determining the E2 contribution requires more effort [deB+17].*

laboratory in Dresden [Gug14; Bem+18]. One challenge in the direct measurement is to detect the low-energy photons with sufficient resolution. This will be improved further through the continuous development of the detectors. Due to the low reaction rate near the Gamow peak, a major challenge is to separate the  $^{12}\text{C}(\alpha, \gamma)^{16}\text{O}$  signals from the much larger signals arising from the cosmic background. In this case, a location in the underground is the one solution to extremely reduce cosmic rays [Gug14; Bem+18]. For example, the LUNA experiment is commissioned 1400 m deep under the Gran Sasso mountain. The muon component of the cosmic background is therefore reduced by a factor of  $10^6$  [Gug14].

### Measurements in Inverse Kinematics

Another method to suppress the background of  $^{13}\text{C}$  is to measure the reaction in inverse kinematics. For this purpose, a high-intensity  $^{12}\text{C}$  beam scatters at an  $^4\text{He}$  gas target. With this method, the S-factor can be determined in two different ways. As in the  $\alpha$ -scattering case, the emitted photons can be detected in order to separate the E1 and E2 contributions. However, the expansion of the gas target and the forward kinematics must be taken into account for the analysis. This makes the analysis more difficult compared to the measurement of  $\alpha$ -scattering on carbon. The second way to determine the S-factor is to detect the recoiling oxygen. In the laboratory system it is emitted in a small window in the forward direction and must be detected with high efficiency. The challenge at this point is to separate the recoiling oxygen from the scattered carbon nuclei. For one  $^{16}\text{O}$  nucleus around  $10^{16}$  -  $10^{18}$   $^{12}\text{C}$  nuclei are detected. The advantage is that with the detection of the recoiling oxygen, the total S-factor can be determined, also includ-

ing the cascade transitions. Since it is impossible to perform this measurement for low cms energies, this technique cannot be used to determine the S-factor near the Gamow peak. An extrapolation without knowledge of the  $\gamma$  distribution is not possible. Even though this method only provides useful results at higher cms energies ( $E \gtrsim 1$  MeV), it still can be helpful. Other experiments, for example, can compare their results from the separations with those of the total S-factor before extrapolating them to the Gamow peak [deB+17; Fey04].

### 2.3.2 Indirect Measurement Techniques

There is a large number of indirect techniques to determine the S-factor of the reaction  $^{12}\text{C}(\alpha, \gamma)^{16}\text{O}$ . The basic idea of all indirect techniques is to measure related nuclear reactions with cross sections that are much larger than the one for  $^{12}\text{C}(\alpha, \gamma)^{16}\text{O}$ . Theoretical models are used to calculate the S-factor of the direct reaction afterwards. For example the  $\beta$ -delayed  $\alpha$ -emission of  $^{16}\text{N}$  can be used to extract the  $\alpha$  partial width of the E1 ground state transition [Fey04]. A full discussion of all these methods can be found in [deB+17]. We will only focus on two indirect methods in this section.

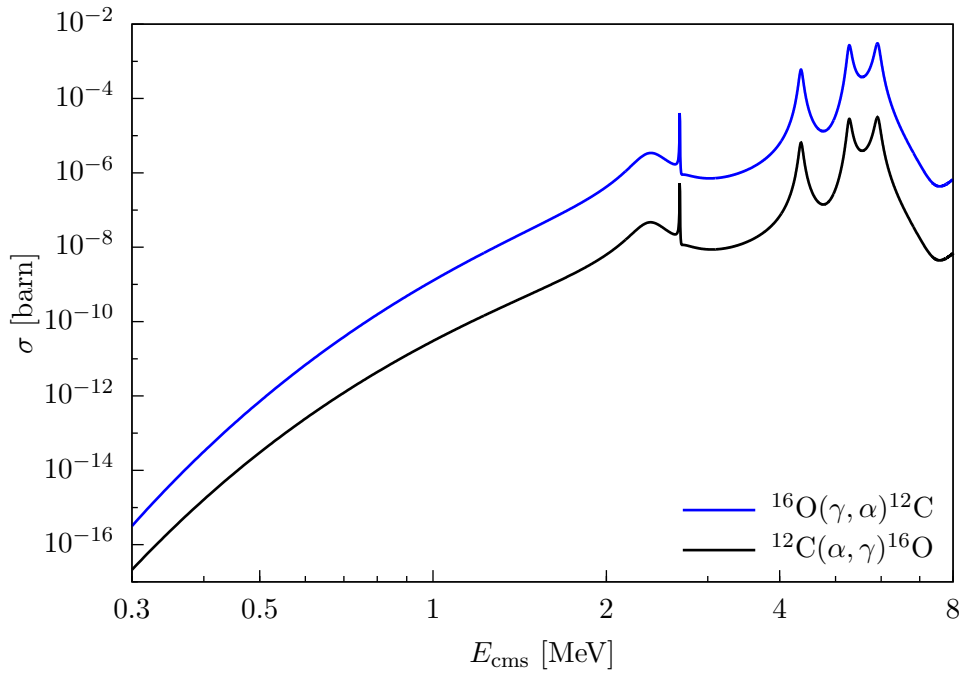
#### The Photo-Disintegration Measurement

The photo-disintegration reaction  $^{16}\text{O}(\gamma, \alpha)^{12}\text{C}$  is one of the most promising inverse techniques. The idea is to break up oxygen nuclei with a  $\gamma$ -ray beam. The recoiled  $\alpha$ -particles are then detected to determine the angular distribution. For a reaction  $A(a, b)B$ , the reciprocity theorem connects the cross sections of interchanged initial and final states. In the center of mass frame, the reciprocity theorem can be expressed as [May02, chapter 7.3]:

$$\frac{\sigma(a, b)}{\sigma(b, a)} = \frac{(2I_b + 1)(2I_B + 1)}{(2I_a + 1)(2I_A + 1)} \cdot \frac{|\vec{p}_b|^2}{|\vec{p}_a|^2}, \quad (2.32)$$

where  $I$  denotes the angular momentum for the particles involved and  $\vec{p}$  the three-momenta in the center of mass frame for the particles in the initial and final state. In the area of the Gamow peak the cross section of the time-reversed reaction  $^{16}\text{O}(\gamma, \alpha)^{12}\text{C}$  is by a factor of  $\sim 50$  higher than the cross section of the direct reaction  $^{12}\text{C}(\alpha, \gamma)^{16}\text{O}$  (see Fig. 2.13). This is an advantage of this method. There have been two different experimental approaches so far. One of them uses a bubble chamber as target, which also acts as a detector for the  $\alpha$ -particle at the same time [HFP19; HFP18; DiG+15; Uga+13]. Another experiment uses a  $\text{CO}_2$ -operated optical readout time projection chamber in which the outgoing  $\alpha$ -particles are detected. Corresponding experiments are currently under development and the first measurements have already been made [Gai12]. For this purpose, an intensive nearly monoenergetic  $\gamma$ -ray beam was used at the HI $\gamma$ S facility. This delivers around  $5 \times 10^7$   $\gamma$ /s to  $3 \times 10^9$   $\gamma$ /s for energies in the range of 1 MeV to 20 MeV [Zim+13; Wel+09].

Currently, the largest limiting factor of the photo-disintegration experiment is the energy spread of the photons, which is about 200 keV for the experiments mentioned above [deB+17]. Since the initial oxygen is in the ground state, this method can only be used to determine the cross section of the ground state transition. But this method can still be used to make important contributions to the determination of the total S-factor as the cross section in the area of the Gamow peak is dominated by the ground state transition of E1 and E2. The advantage of this method is, on the one hand, the higher effective cross section, which allows lower



**Figure 2.13:** The cross section of the time-reversed reaction  $^{16}\text{O}(\gamma, \alpha)^{12}\text{C}$  can be calculated using the reciprocity theorem Eq. (2.32). In the area of the Gamow peak the cross section of the time-reversed reaction is by a factor of  $\sim 50$  higher than the cross section of the direct reaction  $^{12}\text{C}(\alpha, \gamma)^{16}\text{O}$ .

energy regions to be measured with sufficient statistics. On the other hand, this setup allows a very precise measurement of the angular distribution of the outgoing  $\alpha$ -particles, which means that the E1 and E2 components can be separated very well. Experiments at the Extreme Light Infrastructure - Nuclear Physics facility (ELI-NP) are planned, which aim to achieve a higher  $\gamma$ -flux in addition to a better energy resolution [Gal+18; Mat+18; Bal17; Tes+16; Ton+05]. To sum it up, photo-disintegration measurement is a promising method to perform precise low energy measurements of the E1 and E2 components of the S-factor [deB+17].

### The Electro-Disintegration Measurement

The measurement of the electro-disintegration reaction  $^{16}\text{O}(e, e'\alpha)^{12}\text{C}$  is an extension of the photo-disintegration measurement  $^{16}\text{O}(\gamma, \alpha)^{12}\text{C}$  [FDM19]. Electrons are scattered inelastically on oxygen. The four-momentum of the virtual photons can be determined by measuring the four-momentum transfer in the scattering process. Therefore, the scattered electrons and the produced  $\alpha$ -particles have to be detected in coincidence. Finally, the S-factor as a function of the cms energy of the  $^{12}\text{C}$ - $\alpha$  system can be determined from the electro-disintegration cross section. The cross section in the center of mass frame can be written as [FDM19]:

$$\left[ \frac{d\sigma}{dE'd\Omega_e d\Omega_\alpha^{\text{cms}}} \right]_{(e, e'\alpha)} = \frac{M_\alpha M_{12\text{C}} p_\alpha^{\text{cms}} \sigma_{\text{Mott}}}{8\pi^3 W (\hbar c)^3} \cdot (\tilde{v}_L R_L + \tilde{v}_T R_T + \tilde{v}_{TL} R_{TL} + \tilde{v}_{TT} R_{TT}), \quad (2.33)$$

where  $W$  is the cms energy and  $p_\alpha^{\text{cms}}$  the three-momentum of the  $\alpha$ -particle in the center of mass frame. The functions  $\tilde{v}_K$  are factors that depend on the electron kinematics and  $R_K$  are the nuclear response functions: the longitudinal and transverse nuclear electromagnetic current components and the two interference



terms (longitudinal L and transversal T with respect to the direction of the virtual photon). Hereby the transverse nuclear electromagnetic current component  $R_T$  is connected to the differential cross section Eq. (2.28) of the direct reaction in the limit  $Q^2 \rightarrow 0$  via [FDM19]:

$$R_T = \frac{\hbar c \cdot p_\alpha^{\text{cms}} \cdot W}{2\alpha \cdot \omega^{\text{cms}} \cdot M_\alpha M_{12\text{C}}} \cdot 4\pi \left[ \frac{d\sigma}{d\Omega_\gamma^{\text{cms}}} \right]_{(\alpha,\gamma)}, \quad (2.34)$$

with the electron scattering energy transfer  $\omega$ . A full derivation of the cross section can be found in Chapter 3. For further information please refer to the paper by Frišćić, Donnelly, and Milner [FDM19].

There are some advantages and disadvantages compared to the photo-disintegration measurement. A major disadvantage is that the longitudinal and interference terms are currently unknown. These terms must also be determined experimentally and taken into account in the extrapolation for small energies. Another disadvantage is that contamination by  $^{17}\text{O}$ ,  $^{18}\text{O}$  or  $^{14}\text{N}$  of the gas target creates a background that has to be extracted using suitable filters. An advantage is that the background, induced by cosmic rays, is greatly reduced by the coincidence measurement of the electron and the recoiled  $\alpha$ -particle. Furthermore, the virtual photon flux is significantly higher and can be controlled by the scattering angle and beam energy. Finally, the energy resolution of the photon can be determined with high precision by detecting the scattered electron in a detector with high resolution. Nevertheless, this method is an excellent extension of the indirect methods that will also be carried out in the near future in the upcoming experiment MAGIX in Mainz [FDM19].

## 2.4 Results of Existing Measurements

In this section, we will briefly review selected previous results on the measurement of the S-factor of the  $^{12}\text{C}(\alpha, \gamma)^{16}\text{O}$ . All results can be found in Table 2.1. A detailed discussion of all the previous results can be found in [deB+17]. In Fig. 2.14 the measured data at low energies separated by the electric transitions E1 and E2 are presented. As discussed in the previous section, these contributions dominate the S-factor in the Gamow peak region, so we will focus on them in this section. When taking a closer look at the data, we immediately notice that no data exists in the energy range below 0.9 MeV. Thus, there is a large gap where the data must be extrapolated to the Gamow peak. The figure also includes an R-matrix fit and extrapolation performed with the AZURE2 software [Ud15] configured with the results of Ref. [deB+17] provided as supplemental material. The R-matrix fit correlates well with the previous experimental data. Even though the S-factor increases towards the Gamow peak, we must keep in mind that the cross section for small energies is dominated by the Coulomb penetration factor (see Section 2.2 and Fig. 2.13). The cross sections drops by more than 6 orders of magnitude in the range 1 MeV to 0.3 MeV. This makes the determination of the S-factor in the range below 1 MeV so challenging.

Several methods can be used to analyse the data and calculate the S-factor  $S(300 \text{ keV})$ . The R-matrix method (RM) is an excellent tool to analyze the data, taking into account the different states, the interference between them as well as the different transitions. However, this method is computationally very intensive. In many previous works, a specialized or simplified formalism was used, such as the three-level R-matrix method (TLRM) or the five-level R-matrix method

(FLRM), in which only the first three or five states were considered for the various electric ground state transitions [deB+17; Sch+12].

Before discussing the results of historical works, we must mention that the previous analyses indicate that the interference between the two states  $1^-(7.116 \text{ MeV})$  and  $1^-(9.585 \text{ MeV})$  is constructive. To rule out destructive interference between these two states with high confidence, data with high precision and accuracy below  $E_{\text{cms}} = 1.3 \text{ MeV}$  are required [Sch+12]. Most of the previous works assumed constructive interference, unless we mention otherwise.

Dyer and Barnes 1974 extrapolated the S-factor for the E1 ground state transition using two different methods. In addition to the TLRM, they used the hybrid R-matrix method (HRM), in which data from elastic scattering  $^{12}\text{C}(\alpha, \alpha)^{12}\text{C}$  was also included. The results are [DB74]

$$S_{\text{E1}}^{\text{TLRM}}(300 \text{ keV}) = 140_{-120}^{+140} \text{ keV b} \quad S_{\text{E1}}^{\text{HRM}}(300 \text{ keV}) = 80_{-40}^{+50} \text{ keV b.}$$

Also Redder et al. (1987) used the TLRM and the HRM to extrapolate the E1 ground state transition. They also determined the S-factor for the E2 ground state transition using the FLRM and the HRM. In addition, they calculated the contribution of the two cascade transitions via the states  $2^+(6.917 \text{ MeV})$  and  $1^-(7.116 \text{ MeV})$  using TLRM and HRM. The results are [Red+87]

$$\begin{aligned} S_{\text{E1}}^{\text{TLRM}}(300 \text{ keV}) &= 200_{-110}^{+270} \text{ keV b} & S_{\text{E1}}^{\text{HRM}}(300 \text{ keV}) &= 140_{-80}^{+120} \text{ keV b} \\ S_{\text{E2}}^{\text{FLRM}}(300 \text{ keV}) &= 96_{-30}^{+24} \text{ keV b} & S_{\text{E2}}^{\text{HRM}}(300 \text{ keV}) &= 80 \pm 25 \text{ keV b} \\ S_{6.92}^{\text{TLRM}}(300 \text{ keV}) &= 7 \pm 2 \text{ keV b} & S_{6.92}^{\text{HRM}}(300 \text{ keV}) &= 6 \pm 3 \text{ keV b} \\ S_{7.12}(300 \text{ keV}) &= 1.3 \pm 0.5 \text{ keV b} \end{aligned}$$

When comparing the results of Dyer and Barnes; Redder et al. for the different methods, the results for E2 are comparable, but the results differ significantly for the E1 transition. This model dependency is one problem in analyzing the data to determine the S-factor.

To improve the results Ouellet et al. (1996) performed a complete analysis of the data sets from the earlier work of [DB74; Red+87; Kre+88], including the phase shifts from elastic scattering experiments of the reaction  $^{12}\text{C}(\alpha, \alpha)^{12}\text{C}$  [Pla+87] and the  $\beta$ -delayed  $\alpha$  spectrum from the decay of  $^{16}\text{N}$  [Azu+94]. For fitting and extrapolating, they used the TLRM and an alternative method to the R-matrix, the K-matrix method (KM). The results for the E1 ground state transition are [Oue+96]

$$S_{\text{E1}}^{\text{TLRM}}(300 \text{ keV}) = 73.3 \pm 13.2 \text{ keV b} \quad S_{\text{E1}}^{\text{KM}}(300 \text{ keV}) = 84.8 \pm 15.7 \text{ keV b}$$

They combined all their results to calculate the E1 and E2 ground state transition and the total S-factor

$$\begin{aligned} S_{\text{E1}}(300 \text{ keV}) &= 79 \pm 16 \text{ keV b} & S_{\text{E2}}(300 \text{ keV}) &= 36 \pm 6 \text{ keV b} \\ S(300 \text{ keV}) &= 120 \pm 40 \text{ keV b.} \end{aligned}$$

In addition, they used both methods to calculate the E1 ground state transition for destructive interference between the two  $1^-(7.116 \text{ MeV})$  and  $1^-(9.585 \text{ MeV})$  states

$$S_{\text{E1}}(300 \text{ keV}) = 2.4 \pm 1 \text{ keV b.}$$

This clearly shows the effect of the potentially destructive interference on the cross section.

Roters et al. (1999) used the FLRM to calculate the S factor of the E1 ground state transition incorporating earlier data. The analysis results in [Rot+99]

$$S_{\text{E1}}^{\text{FLRM}}(300 \text{ keV}) = 79 \pm 21 \text{ keV b.}$$

This result is comparable with the KM and TLRM results of Ouellet et al. as well as the HRM results of Dyer and Barnes; Redder et al.

Gialanella et al. (2001) performed a full R-matrix analysis of their data, which included the previous data, the phase shifts from elastic scattering, and the  $\alpha$  spectrum data of the  $\beta$ -delayed  $^{16}\text{N}$  decay. For the first time they used a Monte Carlo (MC) simulation to calculate the systematic uncertainties of the differences between the different E1 data sets [Gia+01; deB+17]. Another result of theirs was that destructive interference between the two lowest E1 ground state transitions cannot be rigorously ruled out. The results are [Gia+01]

$$S_{\text{E1}}^{\text{constr.}}(300 \text{ keV}) = 82 \pm 16 \text{ keV b} \quad S_{\text{E1}}^{\text{destr.}}(300 \text{ keV}) = 2.4 \pm 1 \text{ keV b.}$$

Kunz et al. (2001) studied the angular distribution of the S-factor measurements in the low energy region 0.95 MeV to 2.8 MeV in detail. Using the TLRM for E1 and the FLRM for E2 in combination with the phase shifts from elastic scattering and the  $\alpha$  spectrum data of the  $\beta$ -delayed  $^{16}\text{N}$  decay they determined the E1 and E2 ground state transition, the sum of all cascade transitions and the total S-factor [Kun+01]

$$\begin{aligned} S_{\text{E1}}^{\text{TLRM}}(300 \text{ keV}) &= 76 \pm 20 \text{ keV b} & S_{\text{E2}}^{\text{FLRM}}(300 \text{ keV}) &= 85 \pm 30 \text{ keV b} \\ S_{\text{casc}}(300 \text{ keV}) &= 4 \pm 4 \text{ keV b} & S(300 \text{ keV}) &= 165 \pm 50 \text{ keV b.} \end{aligned}$$

The results of the data taken by Fey (2004) as part of his doctoral dissertation are published in [Ham+05]. They used the same technique as Kunz et al. to determine the E1 and E2 ground state transition and the total S-factor [Ham+05; Fey04]

$$\begin{aligned} S_{\text{E1}}^{\text{TLRM}}(300 \text{ keV}) &= 77 \pm 17 \text{ keV b} & S_{\text{E2}}^{\text{FLRM}}(300 \text{ keV}) &= 81 \pm 22 \text{ keV b} \\ S(300 \text{ keV}) &= 162 \pm 39 \text{ keV b.} \end{aligned}$$

The relative error of the total S-factor of the results of [Oue+96; Kun+01; Fey04; Ham+05] is in the range of 24 % to 33 %.

To improve these results, Schürmann et al. (2012) used a similar technique as Ouellet et al. to analyze a large set of historical data. They used a FLRM for the E1 and E2 ground state transition and a TLRM for the E3 ground state and the cascade transitions in agreement with the elastic scattering data and the  $\beta$ -delayed  $\alpha$  spectrum of  $^{16}\text{N}$ . In addition, they calculated the systemic uncertainties for the different data sets with the inclusion of a Monte Carlo simulation. They pointed out that a major problem are the significant discrepancies in the various absolute normalizations of the direct experiments. Their analysis results in [Sch+12]

$$S_{\text{tot}}(300 \text{ keV}) = 161 \pm 19_{(\text{stat})-2(\text{sys})}^{+8} \text{ keV b.}$$

Also, deBoer et al. (2017) analyzed a large set of historical data in consistency with the elastic scattering data and the  $\beta$ -delayed  $\alpha$  spectrum of  $^{16}\text{N}$ . For the fit, they used a global R-matrix fit using the AZURE2 software [Azu+10; Ud15]. A major innovation in their analysis is that they also calculated the uncertainties arising from different models describing the S-factor. Their results are [deB+17]

$$S_{\text{tot}}(300 \text{ keV}) = 140 \pm 21_{(\text{MC})-11(\text{model})}^{+18} \text{ keV b.}$$

The relative error of the total S-factor of [Sch+12; deB+17] results in a range from 17% to 28%. To understand stellar evolution the S-factor in the low energy region must be determined with a relative error of 10% or better. Although analysis and measurements have improved in recent years the present relative error is far from this desired uncertainty. The importance of the S-factor in predicting higher mass elements will be briefly discussed in the next section.

**Table 2.1:** *S-factor results at  $E_{\text{cms}} = 300$  keV of the previous works of [DB74; Red+87; Kre+88; Oue+96; Rot+99; Gia+01; Kun+01; Ham+05; Sch+12; deB+17]. The used models to fit and extrapolate the data are R-matrix method (RM), K-matrix method (KM), hybrid R-matrix method (HRM), three-level R-matrix method (TLRM), five-level R-matrix method (FLRM) and the Monte Carlo (MC) simulation. More details can be found in the text of Section 2.4. For a detailed description of all previous data please refer to [deB+17].*

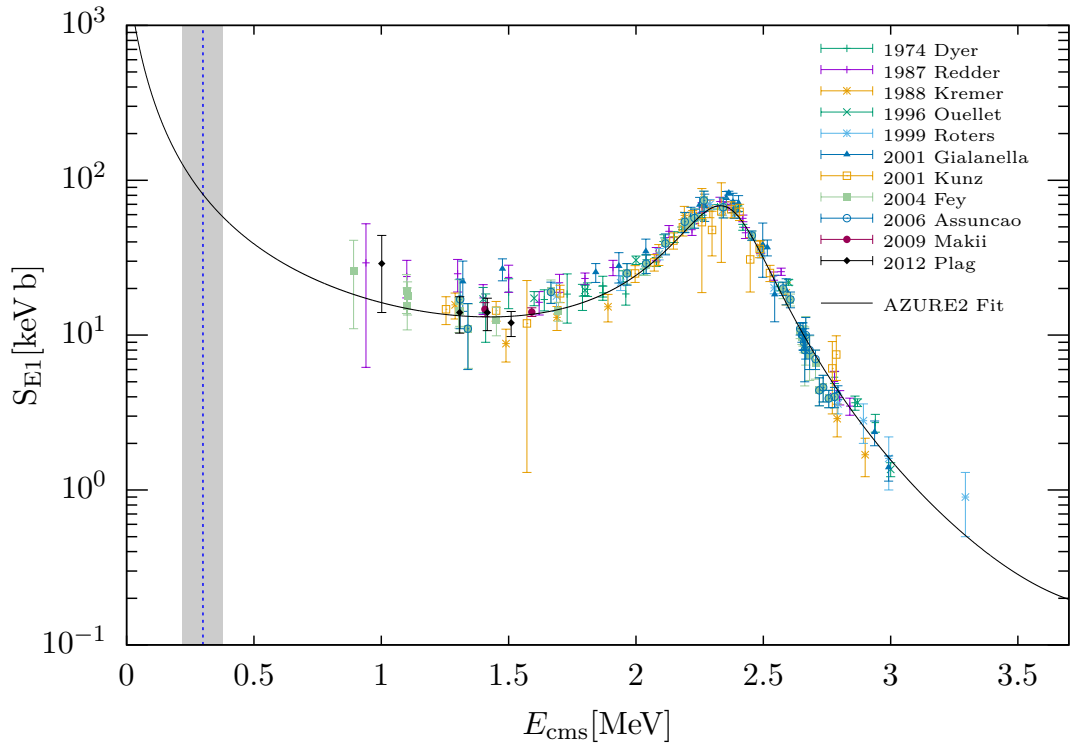
Reference	$S(300 \text{ keV})[\text{keV b}]$				Model
	E1	E2	Cascades	Total	
Dyer and Barnes (1974)	$140^{+140}_{-120}$				TLRM
	$80^{+50}_{-40}$				HRM
Redder et al. (1987)	$200^{+270}_{-110}$	$96^{+24}_{-30}$	$7 \pm 2^{\text{a}}, 1.3^{+0.5\text{b}}_{-1}$		TLRM; FLRM
	$140^{+120}_{-80}$	$80 \pm 25$	$6 \pm 3^{\text{a}}$		HRM
Kremer et al. (1988)	0 – 140				TLRM
Ouellet et al. (1996)	$73.3 \pm 13.2$				TLRM
	$84.8 \pm 15.7$				KM
	$79 \pm 16$	$36 \pm 6$		$120 \pm 40$	KM; TLRM
	$2.4 \pm 1^{\text{d}}$				KM; TLRM
Roters et al. (1999)	$79 \pm 21$				FLRM
Gialanella et al. (2001)	$82 \pm 16$				RM; MC
	$2.4 \pm 1^{\text{d}}$				RM; MC
Kunz et al. (2001)	$76 \pm 20$	$85 \pm 30$	$4 \pm 4^{\text{c}}$	$165 \pm 50$	TLRM; FLRM
Hammer et al. (2005)	$77 \pm 17$	$81 \pm 22$		$162 \pm 39$	TLRM; FLRM
Schürmann et al. (2012)	83.4	73.4	$4.4^{\text{c}}$	$161 \pm 19_{(\text{stat})}^{+8}_{-2(\text{sys})}$	FLRM; MC
	$7.9^{\text{d}}$				FLRM; MC
deBoer et al. (2017)	86.3	45.3	$7^{\text{c}}$	$140 \pm 21_{(\text{MC})}^{+18}_{-11(\text{model})}$	RM, MC

<sup>a</sup> 6.92 MeV transition

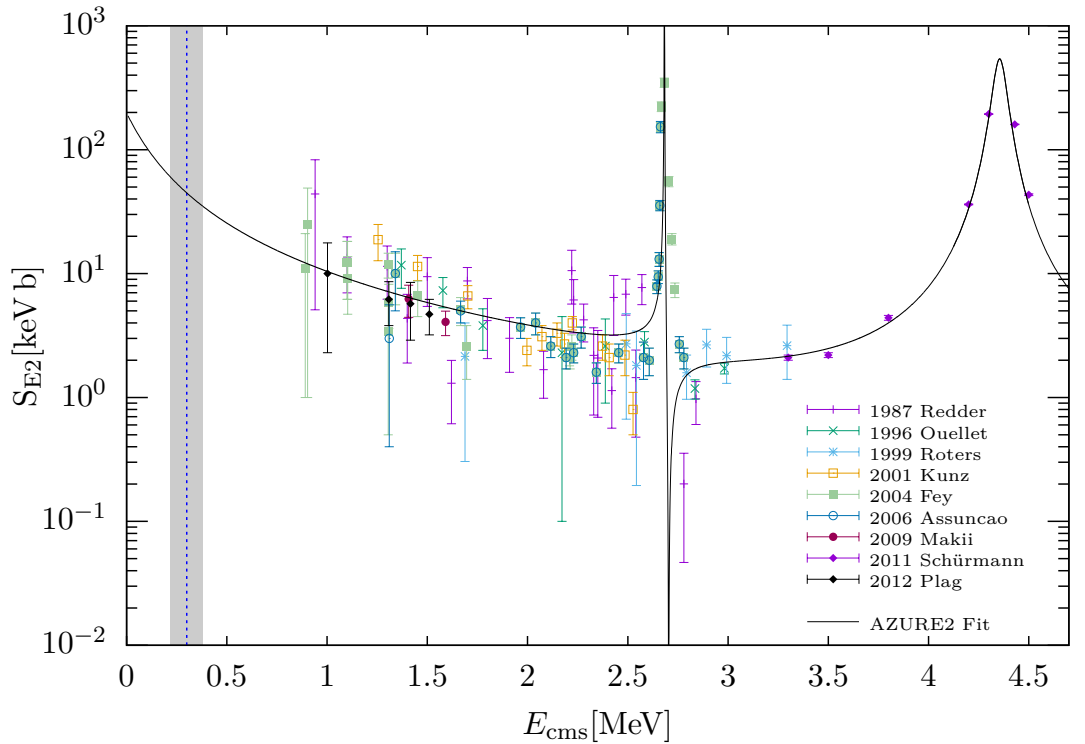
<sup>b</sup> 7.12 MeV transition

<sup>c</sup> all cascade transitions

<sup>d</sup> destructive interference



(a) E1 contribution data [DB74; Red+87; Kre+88; Oue+96; Rot+99; Gia+01; Kun+01; Fey04; Ass+06; Mak+09; Pla+12].



(b) E2 contribution data [Red+87; Oue+96; Rot+99; Kun+01; Fey04; Ass+06; Mak+09; Sch+11; Pla+12].

**Figure 2.14:** *E1* (a) and *E2* (b) contribution data fitted with the AZURE2 software [Azu+10; Ud15] configured with parameters of the supplemental material provided by Ref. [deB+17]. The gray bar represents the Gamow window together with the Gamow energy  $E_G$ , which is shown as a blue dotted line.

## 2.5 Production Factor

In this section, a brief summary of the production factor is given. The (isotopic) production factor is a quantity to describe the ratio of isotopes produced in pre-supernova evolution of massive stars with regards to nucleosynthesis. It is defined as [Koz14; WHW02]

$$P_{\text{iso}} := \frac{m_{\text{iso}}}{X_{\text{iso}}^{\odot} \cdot M_{\text{tot}}},$$

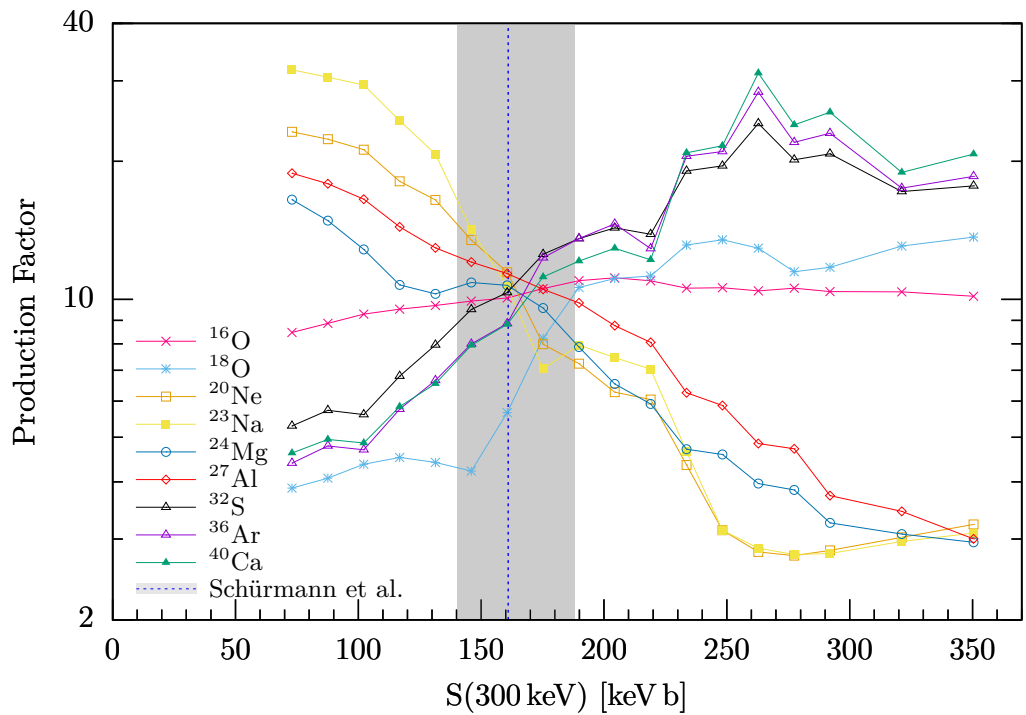
with the mass fraction in the ejecta of a given isotope  $m_{\text{iso}}$  in solar masses, the mass fraction of the isotope according to the solar metallicity  $X_{\text{iso}}^{\odot}$ , and the initial mass of the star  $M_{\text{tot}}$ .

The key uncertainty in all models describing the pre-supernova evolution is the S-factor of the reaction  $^{12}\text{C}(\alpha, \gamma)^{16}\text{O}$  [WH07; WW93]. Figure 2.15 shows the calculated production factor as a function of the S-factor value. These results were published in [WH07]. It is immediately visible that the production factor strongly depends on the value of the S-factor. In addition the results of Schürmann as well as deBoer are shown as gray bars in Figs. 2.15a and 2.15b. It is obvious that no precise prediction for the production factor can be made based on the present results. As discussed in the previous section, a relative uncertainty of better than 10% for the S-factor of the reaction  $^{12}\text{C}(\alpha, \gamma)^{16}\text{O}$  is required to obtain reasonable results for the pre-supernova evolution and the prediction of the mass fraction of stars after the helium burning stage. For more information on the production factor please refer to [WH07; WHW02; WW93; Koz14].

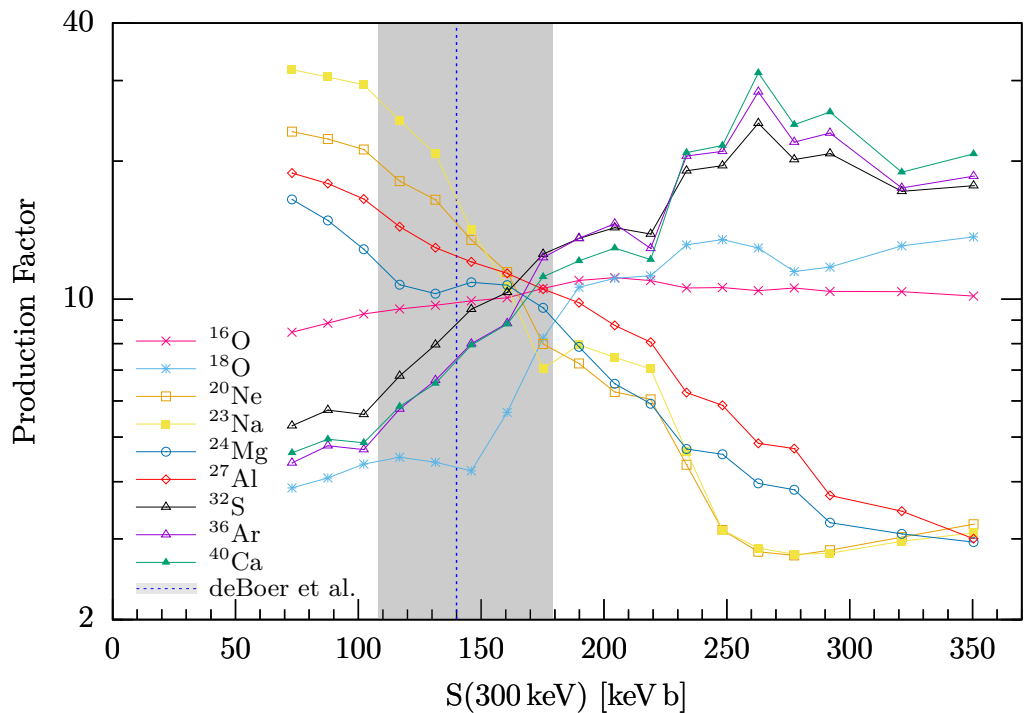
To improve the present results of the S-factor new measurements must be performed. These measurements must fulfill two important criteria. First, the errors in the range above 1 MeV of the S-factor for the E1 and E2 ground state transition must be substantially reduced. Furthermore, it is important to explore the range below 0.9 MeV. On the one hand, to rigorously exclude destructive interference and, on the other hand, to reduce the range over which extrapolation is necessary. Due to the strong decrease of the cross section from  $\sim 10^{-11}$  b at 1 MeV to  $\sim 10^{-17}$  b at 0.3 MeV, the experiment must not only have a high luminosity but also must be able to extract the background very accurately to determine the S-factor in the region below 0.9 MeV.

One of these upcoming experiments is MAGIX where processes with very low cross sections at small momentum transfer can be studied. One goal of the MAGIX physics program is the measurement of the electro-disintegration of oxygen to determine the S-factor of the reaction  $^{12}\text{C}(\alpha, \gamma)^{16}\text{O}$ .

In the following chapter, we will have a closer look at the theory of electro-disintegration. Subsequently, in Chapter 4, we will learn more about MAGIX and will discuss the different phases of the planned measurement. At the end of this thesis, we will discuss the results of the simulation of the electro-disintegration cross section.



(a) Isotopic production factor with the results of Schürmann et al. [Sch+12]



(b) Isotopic production factor with the results of deBoer et al. [deB+17].

**Figure 2.15:** The production factor of some key isotopes as function of the  $S$ -factor. The data is from [WH07]. Two different results of  $S$ -factor calculations are given in (a) and (b) to clarify the current situation with regards to the uncertainties. The dotted blue line is the calculated  $S$ -factor and the gray bar represents the error. Both results show that the current errors in the  $S$ -factor calculations do not allow any robust prediction of the abundance of higher elements that arise in star nucleosynthesis after the helium burning stage.





---

---

## CHAPTER 3

---

# Electro-Disintegration Cross Section of $^{16}\text{O}$

In this chapter, we will take a closer look at the theory of the electro-disintegration reaction  $^{16}\text{O}(e, e'\alpha)^{12}\text{C}$ . The emphasis is on the derivation of the cross section. The chapter concludes by showing how the cross section of the direct reaction  $^{12}\text{C}(\alpha, \gamma)^{16}\text{O}$  enters into the cross section of electro-disintegration.

In the electro-disintegration technique, electrons are scattered inelastically on oxygen. By detecting the scattered electrons and the recoiled  $\alpha$ -particles, the kinetic parameters of the reaction can be determined. In contrast to photo-disintegration, in which a  $\gamma$ -ray beam is used, a virtual photon is exchanged. In addition to the transverse component, the longitudinal components and possible interferences must also be considered in the electro-disintegration process.

We are using the natural system of units with  $\hbar = c = 1$ . The following derivation of the cross section is based on the procedure published in [FDM19; Don+17]. Nevertheless we have to keep in mind that the Extreme Relativistic Limit (ERL) is not applicable for small electron scattering angles. Therefore, it is crucial that formulas are used in its most general form, so that no approximations are done in our calculations.

### 3.1 Kinematics of the Electro-Disintegration Reaction

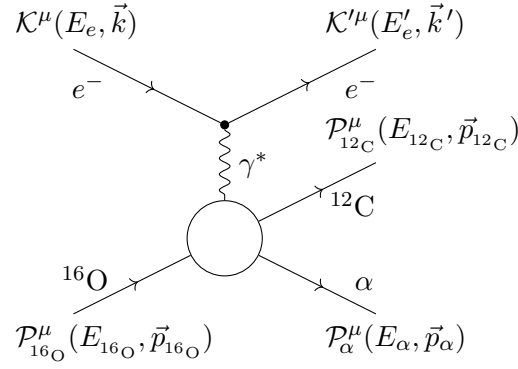
In the electro-disintegration reaction of  $^{16}\text{O}$  (see figure 3.1), we define the four-momentum of the virtual photon  $\gamma^*$  as

$$q^\mu := (\omega, \vec{q}) = \mathcal{K}^\mu - \mathcal{K}'^\mu = \mathcal{P}_\alpha^\mu + \mathcal{P}_{^{12}\text{C}}^\mu - \mathcal{P}_{^{16}\text{O}}^\mu, \quad (3.1)$$

where  $\omega$  is the energy and  $\vec{q}$  is the three-momentum transfer:

$$\omega = E_e - E'_e = E_\alpha + E_{^{12}\text{C}} - E_{^{16}\text{O}}, \quad (3.2)$$

$$\vec{q} = \vec{k} - \vec{k}' = \vec{p}_\alpha + \vec{p}_{^{12}\text{C}} - \vec{p}_{^{16}\text{O}}. \quad (3.3)$$



**Figure 3.1:** First order Feynman diagram for the electro-disintegration reaction  $^{16}\text{O}(e, e'\alpha)^{12}\text{C}$ .

By using the metric tensor  $g_{\mu\nu} = \text{diag}(1, -1, -1, -1)$  we obtain

$$\begin{aligned}
 q_\mu q^\mu &= (\mathcal{K}^\mu - \mathcal{K}'^\mu)^2 = (\mathcal{K}^\mu)^2 + (\mathcal{K}'^\mu)^2 - 2 \cdot g_{\mu\nu} \mathcal{K}^\nu \mathcal{K}'^\mu & (3.4) \\
 &= 2m_e^2 - 2E_e E'_e + 2\vec{k} \cdot \vec{k}' \\
 &= 2m_e^2 - 2E_e E'_e + 2|\vec{k}||\vec{k}'| \cos(\theta_e) \\
 &= 2m_e^2 - 2E_e E'_e (1 - \beta_e \beta'_e \cos(\theta_e)) \\
 &=: -Q^2,
 \end{aligned}$$

where

$$\beta_e = \frac{|\vec{k}|}{E_e} = \frac{|\vec{v}_e|}{c},$$

and  $\theta_e$  the scattering angle of the electron. For the absolute values of the three-momenta we will simply write  $q \equiv |\vec{q}|$ ,  $p_x \equiv |\vec{p}_x|$  or  $k \equiv |\vec{k}|$ .

### 3.2 Differential Cross Section of the Electro-Disintegration Reaction $^{16}\text{O}(e, e'\alpha)^{12}\text{C}$

Following the Appendix B in [BD64] and the equation (B1) for the differential cross section of spinless particles and photons, the cross section of the reaction  $^{16}\text{O}(e, e'\alpha)^{12}\text{C}$  in lab frame, where  $\mathcal{P}_{16\text{O}}^{\mu\text{lab}} = (M_{16\text{O}}, \vec{0})$ , can be written as:

$$\begin{aligned}
 d\sigma &= \frac{1}{\beta_e} \left( \frac{m_e}{E_e} \right) \left( \frac{M_{16\text{O}}}{E_{16\text{O}}} \right) \overline{\sum}_{f_i} |\mathcal{M}_{f_i}|^2 \cdot \frac{m_e}{E'_e} \frac{d^3\vec{k}'}{(2\pi)^3} \cdot \frac{M_\alpha}{E_\alpha} \frac{d^3\vec{p}_\alpha}{(2\pi)^3} \cdot \frac{M_{12\text{C}}}{E_{12\text{C}}} \frac{d^3\vec{p}_{12\text{C}}}{(2\pi)^3} & (3.5) \\
 &\quad \cdot (2\pi)^4 \delta^4 \left( \mathcal{K}^\mu + \mathcal{P}_{16\text{O}}^\mu - \mathcal{K}'^\mu - \mathcal{P}_\alpha^\mu - \mathcal{P}_{12\text{C}}^\mu \right),
 \end{aligned}$$

where  $\mathcal{M}$  is the Lorentz invariant amplitude and  $\overline{\sum}_{f_i}$  represents the sum over final and the average over initial spin states.

In an electro-disintegration experiment, the four-momentum of the scattered electron and the recoiled  $\alpha$  particle are detected, but not the residual  $^{12}\text{C}$ . This means we have to integrate the cross section over the momentum  $d^3\vec{p}_{12\text{C}}$ . With the conservation of the three-momentum

$$\int d^3\vec{p}_{12\text{C}} \delta^3 \left( \vec{k} + \vec{p}_{16\text{O}} - \vec{k}' - \vec{p}_\alpha - \vec{p}_{12\text{C}} \right) = 1,$$

we get

$$d\sigma = \frac{m_e^2 M_\alpha M_{12\text{C}}}{E_e} \frac{1}{\beta_e} \frac{1}{(2\pi)^5} \overline{\sum_{fi}} |\mathcal{M}_{fi}|^2 \cdot \frac{k'^2 dk' d\Omega'_e p_\alpha^2 dp_\alpha d\Omega_\alpha}{E'_e E_\alpha E_{12\text{C}}} \cdot \delta(E_e + E_{16\text{O}} - E'_e - E_\alpha - E_{12\text{C}}). \quad (3.6)$$

Furthermore, we integrate the energy-conserving  $\delta$  function over the recoil momenta  $dp_\alpha$ . The integration will be carried out following the procedure described in [GS64].

If we assume a function  $f(x)$  has any number of simple roots  $\{x_1, \dots, x_n\}$ , we can write

$$\delta(f(x)) = \sum_{i=1}^n \frac{\delta(x - x_i)}{|f'(x_i)|}.$$

In our case the function  $f(p)$  in the lab frame is:

$$\begin{aligned} f(p) &:= \omega + M_{16\text{O}} - \sqrt{|\vec{p}|^2 + M_\alpha} - \sqrt{|\vec{q} - \vec{p}|^2 + M_{12\text{C}}} \\ &= \omega + M_{16\text{O}} - \sqrt{p^2 + M_\alpha} - \sqrt{q^2 + p^2 - 2pq \cos \theta_\alpha + M_{12\text{C}}}, \end{aligned}$$

where  $\theta_\alpha$  is the angle between  $\vec{q}$  and  $\vec{p}_\alpha$ . According to the definition this function has one simple root at  $p \equiv p_\alpha$ :

$$\begin{aligned} f(p_\alpha) &= \omega + M_{16\text{O}} - \sqrt{p_\alpha^2 + M_\alpha} - \sqrt{p_{12\text{C}}^2 + M_{12\text{C}}} \\ &= E_e - E'_e + E_{16\text{O}} - E_\alpha - E_{12\text{C}}. \end{aligned}$$

The derivative of this function evaluated at  $p = p_\alpha$  is

$$\begin{aligned} f'(p) \Big|_{p=p_\alpha} &= \left| \frac{-p}{\sqrt{p^2 + M_\alpha}} + \frac{-p + q \cos \theta_\alpha}{\sqrt{|\vec{q} - \vec{p}|^2 + M_{12\text{C}}}} \right|_{p=p_\alpha} \\ &= \left| \frac{-p_\alpha}{E_\alpha} + \frac{-p_\alpha + q \cos \theta_\alpha}{E_{12\text{C}}} \right| \\ &= \frac{1}{E_\alpha E_{12\text{C}}} \left| E_\alpha q \cos \theta_\alpha - p_\alpha (E_\alpha + E_{12\text{C}}) \right| \\ &= \frac{1}{E_\alpha E_{12\text{C}}} \left| E_\alpha q \cos \theta_\alpha - p_\alpha (\omega + M_{16\text{O}}) \right|. \end{aligned}$$

The integration over  $dp_\alpha$  results in

$$\begin{aligned} &\int \frac{p_\alpha^2 dp_\alpha}{E_\alpha E_{12\text{C}}} \cdot \delta(E_e + E_{16\text{O}} - E'_e - E_\alpha - E_{12\text{C}}) \\ &= \int \frac{p^2 dp}{\left| E_\alpha |\vec{q}| \cos \theta_\alpha - p_\alpha (\omega + M_{16\text{O}}) \right|} \cdot \delta(p - p_\alpha) \\ &= \frac{p_\alpha^2}{\left| E_\alpha |\vec{q}| \cos \theta_\alpha - p_\alpha (\omega + M_{16\text{O}}) \right|} \\ &= \frac{p_\alpha}{M_{16\text{O}}} \cdot \left| 1 + \frac{\omega p_\alpha - E_\alpha |\vec{q}| \cos \theta_\alpha}{p_\alpha M_{16\text{O}}} \right|^{-1} \\ &= \frac{p_\alpha}{M_{16\text{O}}} \cdot f_{\text{rec}}^{-1}, \end{aligned}$$

with the hadronic recoil factor  $f_{\text{rec}}$ . Finally, the differential cross section can be expressed as

$$d\sigma = \frac{m_e^2 M_\alpha M_{12\text{C}}}{(2\pi)^5 M_{16\text{O}}} \frac{1}{\beta_e} \frac{k'^2}{E_e E'_e} \cdot p_\alpha f_{\text{rec}}^{-1} \overline{\sum_{fi}} |\mathcal{M}_{fi}|^2 dk' d\Omega'_e d\Omega_\alpha. \quad (3.7)$$

### 3.3 Calculation of the Matrix Element

Next, we will evaluate the Lorentz invariant matrix element  $\overline{\sum_{fi}} |\mathcal{M}_{fi}|^2$ , as described in [Don+17, Section 7.3]. In the natural system of units  $e^2 = 4\pi\alpha$ , where  $\alpha$  is the fine structure constant, the invariant matrix element can be expressed as [HM84; BD64]:

$$\overline{\sum_{fi}} |\mathcal{M}_{fi}|^2 = \left( \frac{4\pi\alpha}{Q^2} \right)^2 L_{\mu\nu} W_{fi}^{\mu\nu}, \quad (3.8)$$

where  $L_{\mu\nu}$  describes the leptonic and  $W^{\mu\nu}$  the hadronic tensor. With the electron current  $j^\mu$  and the hadronic current matrix element  $J_{fi}^\mu$  the leptonic and hadronic tensors yield [Don+17, eq. 7.10-11]

$$L_{\mu\nu} = \overline{\sum_{\text{leptons}}} j_\mu^* j_\nu, \quad (3.9)$$

$$W_{fi}^{\mu\nu} = \overline{\sum_{\text{hadrons}}} J_{fi}^{\mu*} J_{fi}^\nu. \quad (3.10)$$

Both tensors, the lepton and the hadronic, are conserving the electromagnetic current [HM84]

$$q^\mu L_{\mu\nu} = q^\nu L_{\mu\nu} = 0, \quad (3.11)$$

$$q_\mu W^{\mu\nu} = q_\nu W^{\mu\nu} = 0. \quad (3.12)$$

For unpolarized electron scattering the lepton tensor corresponds to [HM84]:

$$L_{\mu\nu}^{\text{unpol}} = \frac{1}{2m_e^2} \left( \mathcal{K}'_\mu \mathcal{K}_\nu + \mathcal{K}'_\nu \mathcal{K}_\mu - \left( \mathcal{K}'_\alpha \mathcal{K}^\alpha - m_e^2 \right) g_{\mu\nu} \right). \quad (3.13)$$

If we define

$$R_\mu = \frac{1}{2} \left( \mathcal{K}_\mu + \mathcal{K}'_\mu \right), \quad (3.14)$$

we get

$$\begin{aligned} q^\mu R_\mu &= \frac{1}{2} \left( \mathcal{K}_\mu + \mathcal{K}'_\mu \right) \left( \mathcal{K}^\mu - \mathcal{K}'^\mu \right) \\ &= \frac{1}{2} \left( \mathcal{K}^2 - \mathcal{K}'^2 \right) = \frac{1}{2} \left( m_e^2 - m_e^2 \right) \\ &= 0. \end{aligned}$$

We can then re-write the lepton tensor in terms of  $q_\mu$  and  $R_\mu$ . Remembering Eq. (3.4)

$$2\mathcal{K}'_\alpha \mathcal{K}^\alpha - 2m_e^2 = -\left( \mathcal{K}^2 - 2\mathcal{K}'_\alpha \mathcal{K}^\alpha + \mathcal{K}'^2 \right) = Q^2,$$

and considering

$$\begin{aligned} 4R_\mu R_\nu &= \left( \mathcal{K}_\mu + \mathcal{K}'_\mu \right) \left( \mathcal{K}_\nu + \mathcal{K}'_\nu \right) = \mathcal{K}_\mu \mathcal{K}_\nu + \mathcal{K}'_\mu \mathcal{K}_\nu + \mathcal{K}_\mu \mathcal{K}'_\nu + \mathcal{K}'_\mu \mathcal{K}'_\nu, \\ q_\mu q_\nu &= \left( \mathcal{K}_\mu - \mathcal{K}'_\mu \right) \left( \mathcal{K}_\nu - \mathcal{K}'_\nu \right) = \mathcal{K}_\mu \mathcal{K}_\nu - \mathcal{K}'_\mu \mathcal{K}_\nu - \mathcal{K}_\mu \mathcal{K}'_\nu + \mathcal{K}'_\mu \mathcal{K}'_\nu, \end{aligned}$$

we get

$$2R_\mu R_\nu - \frac{1}{2} q_\mu q_\nu = \mathcal{K}_\mu \mathcal{K}'_\nu + \mathcal{K}'_\mu \mathcal{K}_\nu.$$

Finally, we get

$$2m_e^2 L_{\mu\nu} = 2R_\mu R_\nu - \frac{1}{2}(q_\mu q_\nu + Q^2 g_{\mu\nu}). \quad (3.15)$$

In the next step we will focus on calculating:

$$\mathcal{R}_{fi} := 4m_e^2 \cdot L_{\mu\nu} W_{fi}^{\mu\nu}. \quad (3.16)$$

Using the current conservation[HM84]

$$q_\mu J_{fi}^\mu(\vec{q}) = 0,$$

we get

$$\begin{aligned} \mathcal{R}_{fi} &= (2R_\mu \cdot 2R_\nu - q_\mu q_\nu - Q^2 g_{\mu\nu}) \cdot J_{fi}^{\mu*} J_{fi}^\nu \\ &= 2R_\mu J_{fi}^{\mu*} \cdot 2R_\nu J_{fi}^\nu - q_\mu J_{fi}^{\mu*} q_\nu J_{fi}^\nu - Q^2 J_{fi}^{\mu*} g_{\mu\nu} J_{fi}^\nu \\ &= |2R \cdot J_{fi}|^2 - Q^2 |J_{fi}|^2. \end{aligned} \quad (3.17)$$

To calculate (3.17),  $J_{fi}^\mu(\vec{q})$  must be examined more closely. For a detailed approach see the following section.

### 3.4 Definition of Spherical Unit Vectors

The hadronic matrix elements are three-dimensional Fourier transforms of the general transition matrix elements of the electromagnetic current operator [Don+17]:

$$J_{fi}^\mu(\vec{q}) = \int dx e^{iqx} J_{fi}^\mu(x).$$

The component  $\mu = 0$  is the Fourier transformation of the transition charge density. The spatial components  $\mu = 1, 2, 3$  are the Fourier transformations of the transition three-vector current density [Don+17]. Therefore, we can state

$$J_{fi}^\mu(\vec{q}) := \left( \varrho(\vec{q}), \vec{J}_{fi}(\vec{q}) \right).$$

For a set of unit vectors of the three dimensional space  $e_x, e_y, e_z$  the eigenvectors for the operators  $S^2$  and  $S_z$  are [Edm57]

$$e_{\pm 1} = \mp \frac{1}{\sqrt{2}}(e_x \pm ie_y), \quad (3.18)$$

$$e_0 = e_z. \quad (3.19)$$

We obtain

$$\begin{aligned} S^2 e_m &= 2e_m, \\ S_z e_m &= m e_m. \end{aligned} \quad \text{for } m = 0, \pm 1$$

The set of  $e_m$  with  $m = 0, \pm 1$  is called *spherical unit vectors* and it defines an angular momentum system with *spin 1* [Edm57]. In addition the vectors have the following properties [Edm57]:

1. The complex conjugate can be written as

$$e_m^* = (-1)^m e_{-m} \quad \text{for } m = 0, \pm 1.$$

2. The vectors are orthogonal

$$e_m^* e_n = \delta_{mn} \quad \text{for all } m, n = 0, \pm 1.$$

3. A vector  $V$  can be expressed in spherical unit vectors as

$$V = \sum_m V_m e_m^*.$$

where the components are given as scalar product

$$V_m = e_m \cdot V.$$

The coordinate system we choose in this case is having its z-axis along the virtual photon direction  $\vec{q}$  and its y-axis normal to the scattering plane [Don+17]. In more detail we choose

$$e_z(\vec{q}) = \frac{\vec{q}}{|\vec{q}|}, \quad (3.20)$$

$$e_y(\vec{q}) = \frac{\vec{k} \times \vec{k}'}{|\vec{k} \times \vec{k}'|}, \quad (3.21)$$

$$e_x(\vec{q}) = e_y \times e_z. \quad (3.22)$$

In this specific coordinate system we get

$$\vec{J}_{fi}(\vec{q}) = \sum_m J_{fi}(\vec{q}; m) e_m^*(\vec{q}), \quad (3.23)$$

with

$$J_{fi}(\vec{q}; m) = e_m(\vec{q}) \cdot \vec{J}_{fi}(\vec{q}).$$

Furthermore the current conservation implies [Don+17]

$$\begin{aligned} 0 &= q_\mu \cdot J_{fi}^\mu(\vec{q}) = \omega \varrho_{fi}(\vec{q}) - \vec{q} \cdot \vec{J}_{fi}(\vec{q}) \\ &= \omega \varrho_{fi}(\vec{q}) - |\vec{q}| J_{fi}(\vec{q}; 0). \end{aligned}$$

Therefore, we can see a relationship between the longitudinal and charge matrix elements

$$J_{fi}(\vec{q}; 0) = \left( \frac{\omega}{q} \right) \varrho_{fi}(\vec{q}).$$

The representation of the electromagnetic current operator in spherical unit vectors allows to further calculate the right-hand side of Eq. (3.17).

### 3.5 Calculation of Generalized Rosenbluth Factors

In this section we will determine  $\mathcal{R}_{fi}$  by expressing it in terms of various combinations of bilinear products of the current matrix elements:

$$\begin{aligned} R_{fi}^L &:= |\varrho_{fi}(\vec{q})|^2, \\ R_{fi}^T &:= |J_{fi}(\vec{q}; +1)|^2 + |J_{fi}(\vec{q}; -1)|^2, \\ R_{fi}^{TL} &:= -2 \operatorname{Re}\{\varrho_{fi}^*(\vec{q}) (J_{fi}(\vec{q}; +1) - J_{fi}(\vec{q}; -1))\}, \\ R_{fi}^{TT} &:= 2 \operatorname{Re}\{J_{fi}^*(\vec{q}; +1) J_{fi}(\vec{q}; -1)\}, \end{aligned}$$

where the labels  $L$ ,  $T$ ,  $TL$ , and  $TT$  refer to the longitudinal, transverse, transverse-longitudinal interference and transverse-transverse interference responses. They are called nuclear response functions and contain all information of the nuclear structure [Don+17].

Firstly we will calculate the two terms of Eq. (3.17) separately

$$\begin{aligned} 2R_\mu J_{fi}^\mu &= 2R_0 J_{fi}^0 - 2\vec{R} \cdot \vec{J}_{fi} \\ &= (E_e + E_e') \rho(q) - 2\vec{R} \cdot \vec{J}_{fi}. \end{aligned} \quad (3.24)$$

For the scalar product we use (3.23) to get

$$\begin{aligned} 2\vec{R} \cdot \vec{J}_{fi} &= (\vec{k} + \vec{k}') \cdot \vec{J}_{fi}(q) \\ &= (\vec{k} + \vec{k}') \cdot \sum_m J_{fi}(\vec{q}; m) e_m^*(\vec{q}) \\ &= J_{fi}(\vec{q}; 0) (\vec{k} + \vec{k}') \cdot e_0^*(\vec{q}) + J_{fi}(\vec{q}; +1) (\vec{k} + \vec{k}') \cdot e_+^*(\vec{q}) \\ &\quad + J_{fi}(\vec{q}; -1) (\vec{k} + \vec{k}') \cdot e_-^*(\vec{q}). \end{aligned}$$

In a next step each of those scalar products will be evaluated separately

$$\begin{aligned} (\vec{k} + \vec{k}') \cdot e_0^*(\vec{q}) &= (\vec{k} + \vec{k}') \cdot e_z(\vec{q}) = (\vec{k} + \vec{k}') \cdot \frac{\vec{q}}{q} = \frac{(\vec{k} + \vec{k}') \cdot (\vec{k} - \vec{k}')}{q} \\ &= \frac{k^2 - k'^2}{q} = \frac{(E_e^2 - m_e^2) - (E_e'^2 - m_e^2)}{q} = \frac{E_e^2 - E_e'^2}{q} \\ &= \frac{\omega}{q} (E_e + E_e'). \end{aligned}$$

The coordinate system described in section 3.4 implies that the y-axis is perpendicular to the scattering plane =  $\text{span}\{\vec{k}, \vec{k}'\}$ . This means  $(\vec{k} + \vec{k}') \cdot e_y(\vec{q}) = 0$ , thus we get

$$\begin{aligned} (\vec{k} + \vec{k}') \cdot e_{\pm 1}^*(\vec{q}) &= \mp \frac{1}{\sqrt{2}} (\vec{k} + \vec{k}') \cdot (e_x(\vec{q}) - i e_y(\vec{q})) = \mp \frac{1}{\sqrt{2}} (\vec{k} + \vec{k}') \cdot e_x(\vec{q}) \\ &= \mp \frac{1}{\sqrt{2} \cdot q |\vec{k} \times \vec{k}'|} (\vec{k} + \vec{k}') \cdot (\vec{k} \times \vec{k}') \times \vec{q}. \end{aligned}$$

We use the relation  $(a \times b) \times c = (a \cdot c) b - (b \cdot c) a$  to evaluate

$$\begin{aligned} (\vec{k} + \vec{k}') \cdot (\vec{k} \times \vec{k}') \times \vec{q} &= (\vec{k} + \vec{k}') \cdot [k^2 \vec{k}' + k'^2 \vec{k} - (\vec{k} \cdot \vec{k}') (\vec{k} + \vec{k}')] \\ &= 2k^2 k'^2 - 2(\vec{k} \cdot \vec{k}')^2 = 2k^2 k'^2 (1 - \cos^2(\theta_e)) \\ &= 2k^2 k'^2 \sin^2(\theta_e). \end{aligned}$$

Additionally, considering the fact  $|\vec{k} \times \vec{k}'| = k k' \sin(\theta_e)$  we finally get

$$\begin{aligned} (\vec{k} + \vec{k}') \cdot e_{\pm 1}^*(\vec{q}) &= \mp \frac{1}{\sqrt{2} \cdot q k k' \sin(\theta_e)} \cdot 2k^2 k'^2 \sin^2(\theta_e) \\ &= \mp \frac{\sqrt{2}}{q} k k' \sin(\theta_e). \end{aligned}$$

When combining all these parts we get

$$\begin{aligned}
 2R_\mu J_{fi}^\mu(\vec{q}) &= (E_e + E'_e) \left( 1 - \left( \frac{\omega}{q} \right)^2 \right) \varrho_{fi}(\vec{q}) \\
 &\quad + \frac{\sqrt{2}}{q} k k' \sin(\theta_e) \left[ J_{fi}(\vec{q}; +1) - J_{fi}(\vec{q}; -1) \right] \\
 &= (E_e + E'_e) \frac{Q^2}{q^2} \varrho_{fi}(\vec{q}) + \frac{\sqrt{2}}{q} k k' \sin(\theta_e) \left[ J_{fi}(\vec{q}; +1) - J_{fi}(\vec{q}; -1) \right].
 \end{aligned}$$

To simplify the term we define

$$a := (E_e + E'_e) \frac{Q^2}{q^2}, \quad (3.25)$$

$$b := \frac{\sqrt{2}}{q} k k' \sin(\theta_e), \quad (3.26)$$

and we get

$$2R_\mu J_{fi}^\mu(\vec{q}) = a \cdot \varrho_{fi}(\vec{q}) + b \cdot \left[ J_{fi}(\vec{q}; +1) - J_{fi}(\vec{q}; -1) \right]. \quad (3.27)$$

This helps us to specify the first part of Eq. (3.17)

$$\begin{aligned}
 |2R_\mu J_{fi}^\mu(\vec{q})|^2 &= a^2 |\varrho_{fi}(\vec{q})|^2 + ab \varrho_{fi}^*(\vec{q}) \left[ J_{fi}(\vec{q}; +1) - J_{fi}(\vec{q}; -1) \right] \\
 &\quad + ab \varrho_{fi}(\vec{q}) \left[ J_{fi}^*(\vec{q}; +1) - J_{fi}^*(\vec{q}; -1) \right] \\
 &\quad + b^2 \left[ J_{fi}^*(\vec{q}; +1) - J_{fi}^*(\vec{q}; -1) \right] \left[ J_{fi}(\vec{q}; +1) - J_{fi}(\vec{q}; -1) \right] \\
 &= a^2 |\varrho_{fi}(\vec{q})|^2 + ab \operatorname{Re} \left\{ \varrho_{fi}^*(\vec{q}) \left[ J_{fi}(\vec{q}; +1) - J_{fi}(\vec{q}; -1) \right] \right\} \\
 &\quad + b^2 \left( |J_{fi}(\vec{q}; +1)|^2 + |J_{fi}(\vec{q}; -1)|^2 \right) \\
 &\quad - b^2 \cdot \operatorname{Re} \left\{ J_{fi}^*(\vec{q}; +1) J_{fi}(\vec{q}; -1) \right\} \\
 &= a^2 R_{fi}^L - ab R_{fi}^{TL} + b^2 R_{fi}^T - b^2 R_{fi}^{TT}. \quad (3.28)
 \end{aligned}$$

The second part of Eq. (3.17) becomes

$$\begin{aligned}
 |J_{fi}^\mu(\vec{q})|^2 &= J^0(\vec{q})^2 - \vec{J}^*(\vec{q}) \cdot \vec{J}(\vec{q}) = |\varrho_{fi}(\vec{q})|^2 - \sum_m |J_{fi}(\vec{q}; m)|^2 \\
 &= \left( 1 - \frac{\omega^2}{q^2} \right) |\varrho_{fi}(\vec{q})|^2 - \left( |J_{fi}(\vec{q}; +1)|^2 + |J_{fi}(\vec{q}; -1)|^2 \right) \\
 &= \frac{Q^2}{q^2} R_{fi}^L - R_{fi}^T. \quad (3.29)
 \end{aligned}$$

In summary we get

$$\begin{aligned}
 \mathcal{R}_{fi} &= \left( a^2 - Q^2 \frac{Q^2}{q^2} \right) R_{fi}^L + \left( b^2 + Q^2 \right) R_{fi}^T - ab R_{fi}^{TL} - b^2 R_{fi}^{TT} \\
 &=: v_0 v_L R_{fi}^L + v_0 v_T R_{fi}^T + v_0 v_{TL} R_{fi}^{TL} + v_0 v_{TT} R_{fi}^{TT}, \quad (3.30)
 \end{aligned}$$



with the *generalized Rosenbluth factors*

$$v_0 v_L = \left( (E_e + E'_e)^2 - q^2 \right) \left( \frac{Q^2}{q^2} \right)^2, \quad (3.31)$$

$$v_0 v_T = 2 \left( \frac{k k'}{q} \sin(\theta_e) \right)^2 + Q^2, \quad (3.32)$$

$$v_0 v_{TL} = -\sqrt{2} (E_e + E'_e) \left( \frac{Q^2}{q^2} \right) \frac{k k'}{q} \sin(\theta_e), \quad (3.33)$$

$$v_0 v_{TT} = -2 \left( \frac{k k'}{q} \sin(\theta_e) \right)^2. \quad (3.34)$$

By convention the common factor

$$v_0 = (E_e + E'_e)^2 - |\vec{q}|^2, \quad (3.35)$$

is separated. We will discuss its importance in the following section. We also define  $\rho := \frac{Q^2}{q^2} = 1 - \frac{\omega^2}{q^2}$  which implies  $0 \leq \rho \leq 1$  [Don+17, Section 7.1].

### 3.6 Calculating the Generalized Mott Cross Section

Before combining all results to describe the differential cross section it is necessary to determine the Mott cross section in its general form. As stated in [BD64, Section 7] the differential Coulomb scattering cross section can be written as

$$\frac{d\bar{\sigma}}{d\Omega'_e} = \frac{\alpha^2}{Q^4} (4E_e E'_e - 2\mathcal{K}^\mu \mathcal{K}'_\mu + 2m_e^2).$$

When using the relativistic energy-momentum relation  $m_e^2 = E_e^2 - k^2$  we get:

$$\begin{aligned} 4E_e E'_e - 2\mathcal{K}^\mu \mathcal{K}'_\mu + 2m_e^2 &= 2E_e E'_e + 2\vec{k} \cdot \vec{k}' + 2m_e^2 \\ &= E_e^2 - k^2 + E_e'^2 - k'^2 + 2E_e E'_e + 2\vec{k} \cdot \vec{k}' \\ &= (E_e + E'_e)^2 - (\vec{k} - \vec{k}')^2 = (E_e + E'_e)^2 - (\vec{q})^2 \\ &= v_0. \end{aligned}$$

And so the generalized Mott cross section is

$$\left( \frac{d\sigma}{d\Omega'_e} \right)_{\text{Mott}} = \frac{d\bar{\sigma}}{d\Omega'_e} \cdot \frac{E'_e}{E_e} = \frac{\alpha^2 v_0 E'_e}{Q^4 E_e}. \quad (3.36)$$

### 3.7 Combining Results to Differential Cross Section

As consequence of the previous derivation, we can write the invariant matrix element now as

$$\overline{\sum_{fi}} |\mathcal{M}_{fi}|^2 = \left( \frac{4\pi\alpha}{Q^2} \right)^2 \cdot \frac{v_0}{4m_e^2} \sum_K v_K R_K$$

with  $K = L, T, TL, TT$  and then the cross section becomes

$$\begin{aligned} d\sigma &= \frac{M_\alpha M_{12\text{C}} m_e^2}{(2\pi)^5 M_{16\text{O}}} \frac{1}{\beta_e} \cdot \frac{k'^2}{E_e E'_e} p_\alpha f_{\text{rec}}^{-1} \left( \frac{4\pi\alpha}{Q^2} \right)^2 \cdot \frac{v_0}{4m_e^2} \left( \sum_K v_K R_K \right) dk' d\Omega'_e d\Omega_\alpha \\ &= \frac{M_\alpha M_{12\text{C}}}{8\pi^3 M_{16\text{O}}} \cdot \frac{\beta_e'^2}{\beta_e} \cdot \frac{E'_e}{E_e} \frac{\alpha^2 v_0}{Q^4} \cdot p_\alpha f_{\text{rec}}^{-1} \left( \sum_K v_K R_K \right) dk' d\Omega'_e d\Omega_\alpha. \end{aligned}$$

Finally, we get the differential cross section for the electro-disintegration calculated in the lab frame

$$\frac{d\sigma}{dk'd\Omega'_e d\Omega_\alpha} = \frac{M_\alpha M_{12\text{C}}}{8\pi^3 M_{16\text{O}}} \cdot \frac{\beta_e'^2}{\beta_e} \cdot \left( \frac{d\sigma}{d\Omega'_e} \right)_{\text{Mott}} \cdot p_\alpha f_{\text{rec}}^{-1} \cdot \sum_K v_K R_K. \quad (3.37)$$

### 3.8 Transformation to Center of Mass Frame

Instead of calculating the differential cross section (3.37) in the lab frame of the experiment  $d\Omega_\alpha^{\text{lab}}$  it is useful to calculate the hadronic part in the center of mass system (cms) of the outgoing carbon- $\alpha$  pair. The goal is to express the term  $p_\alpha f_{\text{rec}}^{-1} \cdot \sum_K v_K R_K$  in the cms. The transition from the lab to the cms includes a Lorentz boost along the virtual photon three-momentum  $\vec{q}$ .

The invariant mass  $W$  of the outgoing carbon- $\alpha$  system is defined as

$$W^2 := |q^\mu + \mathcal{P}_{16\text{O}}^\mu|^2 = |\mathcal{P}_{12\text{C}}^\mu + \mathcal{P}_\alpha^\mu|^2.$$

In the lab frame the following applies

$$\mathcal{P}_{16\text{O}}^\mu + q^\mu = \begin{pmatrix} \omega + M_{16\text{O}} \\ \vec{q} \end{pmatrix},$$

and so  $W^2 = (\omega_{\text{lab}} + M_{16\text{O}})^2 - q_{\text{lab}}^2$ .

The right side of

$$\frac{p_\alpha^{\text{lab}}}{M_{16\text{O}}} \cdot f_{\text{rec}}^{-1} = \int \frac{p_\alpha^2 dp_\alpha}{E_\alpha E_{12\text{C}}} \cdot \delta(E_e + E_{16\text{O}} - E'_e - E_\alpha - E_{12\text{C}}),$$

needs to be evaluated in a specific frame and we will choose the cms:

$$\mathcal{P}_{16\text{O}}^\mu + q^\mu = \begin{pmatrix} W \\ \vec{0} \end{pmatrix} = \mathcal{P}_{12\text{C}}^\mu + \mathcal{P}_\alpha^\mu.$$

The energy and momentum conservation yields  $W = E_{16\text{O}} + \omega = E_{12\text{C}} + E_\alpha$ , and  $p_\alpha = p_{12\text{C}} =: p_\alpha^{\text{cms}}$ . So, defining

$$f(p) := W - \sqrt{p^2 + M_\alpha^2} - \sqrt{p^2 + M_{12\text{C}}^2},$$

we have

$$\left. \frac{\partial f}{\partial p} \right|_{p=p_\alpha^{\text{cms}}} = \left| \frac{p_\alpha^{\text{cms}}}{E_\alpha} + \frac{p_\alpha^{\text{cms}}}{E_{12\text{C}}} \right| = \frac{W p_\alpha^{\text{cms}}}{E_\alpha E_{12\text{C}}}.$$

The integration results in

$$\begin{aligned} \frac{p_\alpha^{\text{lab}}}{M_{16\text{O}}} \cdot f_{\text{rec}}^{-1} &= \int \frac{p_\alpha^2 dp_\alpha}{E_\alpha E_{12\text{C}}} \cdot \delta(E_e + E_{16\text{O}} - E'_e - E_\alpha - E_{12\text{C}}) \\ &= \int \frac{p^2 dp}{W p_\alpha^{\text{cms}}} \delta(p - p_\alpha^{\text{cms}}) = \frac{p_\alpha^{\text{cms}}}{W}. \end{aligned}$$

This means in particular

$$p_\alpha^{\text{lab}} f_{\text{rec}}^{-1} = \frac{M_{16\text{O}}}{W} p_\alpha^{\text{cms}}. \quad (3.38)$$

To calculate  $v_K$  and  $v_0$  in the cms we have to boost these scalar parameters along  $\vec{q}$ .

For a relative velocity  $\vec{v}_{\text{rel}} = |\vec{v}_{\text{rel}}| \cdot \hat{\mathbf{n}}$  the general Lorentz boost of a four-momentum  $P^\mu = (E, \vec{p})$  can be written as [DL05]

$$\begin{aligned} E^{\text{boost}} &= \gamma_{\text{rel}}(E - \beta_{\text{rel}}(\vec{p} \cdot \hat{\mathbf{n}})), \\ \vec{p}^{\text{boost}} &= \vec{p} + (\gamma_{\text{rel}} - 1)(\vec{p} \cdot \hat{\mathbf{n}}) \cdot \hat{\mathbf{n}} - \gamma_{\text{rel}}\beta_{\text{rel}}E \cdot \hat{\mathbf{n}}, \end{aligned}$$

with  $\beta_{\text{rel}} = \frac{v_{\text{rel}}}{c}$  and  $\gamma_{\text{rel}} = [1 - \beta_{\text{rel}}^2]^{-1/2}$ . If  $\vec{p} = \vec{p}_{\parallel} + \vec{p}_{\perp}$  with  $\vec{p}_{\perp} \cdot \vec{v} = 0$  and  $\vec{p} \cdot \vec{v} = \vec{p}_{\parallel} \cdot \vec{v}$  we have

$$\begin{aligned} \vec{p}_{\parallel}^{\text{boost}} &= \gamma_{\text{rel}}(\vec{p} \cdot \hat{\mathbf{n}} - \beta_{\text{rel}}E) \cdot \hat{\mathbf{n}}, \\ \vec{p}_{\perp}^{\text{boost}} &= \vec{p}_{\perp}. \end{aligned}$$

The Lorentz boost into the center of mass system means a boost along  $\vec{q}$ . The velocity vector is given as  $\vec{v}_{\text{rel}} = \frac{1}{W\gamma_{\text{rel}}}\vec{q}$ , so we have:

$$\begin{aligned} \beta_{\text{rel}} &= \frac{q}{\omega + M_{16\text{O}}}, \\ \gamma_{\text{rel}} &= \frac{\omega + M_{16\text{O}}}{W}, \\ \gamma_{\text{rel}} \cdot \beta_{\text{rel}} &= \frac{q}{W}. \end{aligned}$$

Boosting the three-momentum of the virtual photon we get

$$\begin{aligned} \vec{q}^{\text{cms}} &= \gamma_{\text{rel}}(q^{\text{lab}} - \beta_{\text{rel}}\omega^{\text{lab}}) \frac{\vec{q}^{\text{lab}}}{q^{\text{lab}}} = \left( \gamma_{\text{rel}} - \frac{\omega^{\text{lab}}}{W} \right) \vec{q}^{\text{lab}} \\ &= \frac{M_{16\text{O}}}{W} \vec{q}^{\text{lab}}. \end{aligned}$$

To boost the scalar parameter, let's also have a closer look at the energy of the incoming and outgoing electron

$$\begin{aligned} E_e^{\text{cms}} &= \gamma_{\text{rel}} \left( E_e^{\text{lab}} - \frac{\beta_{\text{rel}}}{q} \cdot \vec{k}^{\text{lab}} \cdot \vec{q}^{\text{lab}} \right) = \gamma_{\text{rel}} E_e^{\text{lab}} - \frac{\vec{k}^{\text{lab}} \cdot \vec{q}^{\text{lab}}}{W} \\ &= \gamma_{\text{rel}} E_e^{\text{lab}} - \frac{(k^{\text{lab}})^2 - \vec{k}^{\text{lab}} \cdot \vec{k}'^{\text{lab}}}{W}, \\ E_e'^{\text{cms}} &= \gamma_{\text{rel}} E_e'^{\text{lab}} - \frac{\vec{k}^{\text{lab}} \cdot \vec{k}'^{\text{lab}} - (k'^{\text{lab}})^2}{W}. \end{aligned}$$

Summing it up, we get

$$\begin{aligned} E_e^{\text{cms}} + E_e'^{\text{cms}} &= \gamma_{\text{rel}}(E_e^{\text{lab}} + E_e'^{\text{lab}}) - \frac{(k^{\text{lab}})^2 - (k'^{\text{lab}})^2}{W} = \left( \gamma_{\text{rel}} - \frac{\omega^{\text{lab}}}{W} \right) (E_e^{\text{lab}} + E_e'^{\text{lab}}) \\ &= \frac{M_{16\text{O}}}{W} (E_e^{\text{lab}} + E_e'^{\text{lab}}). \end{aligned}$$

This enables us to calculate

$$\begin{aligned} v_0^{\text{cms}} &= (E_e^{\text{cms}} + E_e'^{\text{cms}})^2 - (q^{\text{cms}})^2 = \left( \frac{M_{16\text{O}}}{W} \right)^2 (E_e^{\text{lab}} + E_e'^{\text{lab}})^2 - \left( \frac{M_{16\text{O}}}{W} \right)^2 (q^{\text{lab}})^2 \\ &= \left( \frac{M_{16\text{O}}}{W} \right)^2 v_0^{\text{lab}}. \end{aligned}$$

Additionally the substitution  $a$  Eq. (3.25) in the cms can be expressed as:

$$\begin{aligned} a^{\text{cms}} &= (E_e^{\text{cms}} + E_e'^{\text{cms}}) \frac{Q^2}{(q^{\text{cms}})^2} \\ &= \frac{W}{M_{^{16}\text{O}}} a^{\text{lab}}. \end{aligned}$$

The substitution  $b$  Eq. (3.26) can be written as:

$$b := (\vec{k} + \vec{k}') \cdot e_{-1}^*(\vec{q}) = -\frac{1}{\sqrt{2}} (\vec{k} + \vec{k}') \cdot e_x(\vec{q}).$$

Since  $e_x(\vec{q}) \perp \vec{q}$  we get

$$b = -\frac{1}{\sqrt{2}} (\vec{k} + \vec{k}')_{\perp} \cdot e_x(\vec{q}),$$

from which we directly obtain  $b^{\text{cms}} = b^{\text{lab}}$ , because the boost of the orthogonal part doesn't change. Inserting this in the formula to calculate  $v_K^{\text{cms}}$  we finally get

$$\begin{aligned} v_0^{\text{cms}} &= \frac{M_{^{16}\text{O}}^2}{W^2} v_0^{\text{lab}}, \\ v_L^{\text{cms}} &= \frac{W^4}{M_{^{16}\text{O}}^4} v_L^{\text{lab}}, & v_T^{\text{cms}} &= \frac{W^2}{M_{^{16}\text{O}}^2} v_T^{\text{lab}}, \\ v_{TL}^{\text{cms}} &= \frac{W^3}{M_{^{16}\text{O}}^3} v_{TL}^{\text{lab}}, & v_{TT}^{\text{cms}} &= \frac{W^2}{M_{^{16}\text{O}}^2} v_{TT}^{\text{lab}}. \end{aligned}$$

Defining  $\tilde{v}_0 := \frac{W^2}{M_{^{16}\text{O}}^2} v_0^{\text{cms}} = v_0^{\text{lab}}$  and  $\tilde{v}_K := \frac{M_{^{16}\text{O}}^2}{W^2} v_K^{\text{cms}}$ , we have  $\tilde{v}_0 \tilde{v}_K = v_0^{\text{cms}} v_K^{\text{cms}}$ .

This enables us to write the differential cross section of the electro-disintegration with the hadronic part calculated in the cms of the carbon- $\alpha$  system

$$\frac{d\sigma}{d\omega d\Omega'_e d\Omega_{\alpha}^{\text{cms}}} = \frac{M_{\alpha} M_{^{12}\text{C}}}{8\pi^3 W} \cdot \frac{\beta_e'^2}{\beta_e} \cdot p_{\alpha}^{\text{cms}} \sigma_{\text{Mott}} \cdot \sum_K \tilde{v}_K R_K^{\text{cms}}, \quad (3.39)$$

with

$$\sigma_{\text{Mott}} = \frac{\alpha^2 v_0^{\text{lab}}}{Q^4} \frac{E_e'^{\text{lab}}}{E_e^{\text{lab}}},$$

and

$$\begin{aligned} \tilde{v}_L &= \frac{W^2}{M_{^{16}\text{O}}^2} v_L^{\text{lab}}, & \tilde{v}_{TL} &= \frac{W}{M_{^{16}\text{O}}} v_{TL}^{\text{lab}}, \\ \tilde{v}_T &= v_T^{\text{lab}}, & \tilde{v}_{TT} &= v_{TT}^{\text{lab}}. \end{aligned}$$

### 3.9 Approximation Using the Extreme Relativistic Limit

It is also useful to determine the approximated formulas based on the ERL. This is applicable for high electron energies  $E_e \gg m_e$  with scattering angles  $\theta_e \geq 1^\circ$ . These can be used, for example, for simulations or analyses in this parameter range, since the calculations with these approximated formulas are significantly more time efficient.

By using the ERL, the four-momentum transfer  $Q^2$  Eq. (3.4) can be expressed as:

$$Q^2 = 2E_e E_e' (1 - \cos(\theta_e)).$$

Using the double-angle formula  $2 \sin^2(\theta) = 1 - \cos(2\theta)$  we get

$$Q^2 = 4E_e E'_e \sin^2\left(\frac{\theta_e}{2}\right). \quad (3.40)$$

At this point we also want to calculate the generalized Rosenbluth factors using the Extreme Relativistic Limit.

Using the ERL on Eq. (3.35) we get

$$\begin{aligned} v_0 &= (E_e + E'_e)^2 - q^2 = E_e^2 - k^2 + E_e'^2 - k'^2 + 2E_e E'_e + 2kk' \cos(\theta_e) \\ &= 2E_e E'_e (1 + \cos(\theta_e)) \\ &= 4E_e E'_e \cos^2\left(\frac{\theta_e}{2}\right). \end{aligned}$$

Per definition the longitudinal part can be expressed as

$$v_L = \left(\frac{Q^2}{q^2}\right)^2 = \rho^2.$$

We can use Eq. (3.40) to simplify

$$\begin{aligned} k^2 k'^2 \sin^2(\theta_e) &= 4k^2 k'^2 \sin^2\left(\frac{\theta_e}{2}\right) \cos^2\left(\frac{\theta_e}{2}\right) \\ &= \frac{1}{4} v_0 Q^2, \\ \frac{Q^2}{v_0} &= \tan^2\left(\frac{\theta_e}{2}\right). \end{aligned}$$

Therefore, we get

$$v_0 v_T = \frac{1}{2} v_0 \frac{Q^2}{q^2} + v_0 \tan^2\left(\frac{\theta_e}{2}\right),$$

and finally

$$v_T = \frac{1}{2} \rho + \tan^2\left(\frac{\theta_e}{2}\right).$$

For the transverse-longitudinal interference we have

$$\begin{aligned} v_{TL} &= -\frac{\sqrt{2}}{v_0} \cdot (E_e + E'_e) \rho \cdot \frac{kk'}{q} \sin(\theta_e) \\ &= -\frac{\sqrt{2} \rho (E_e + E'_e) E_e E'_e \cdot 2 \sin\left(\frac{\theta_e}{2}\right) \cos\left(\frac{\theta_e}{2}\right)}{q \cdot 4E_e E'_e \cos^2\left(\frac{\theta_e}{2}\right)} \\ &= -\frac{\rho (E_e + E'_e) \sin\left(\frac{\theta_e}{2}\right)}{\sqrt{2} q \cos\left(\frac{\theta_e}{2}\right)}. \end{aligned}$$

If we take a closer look at the denominator of the right fraction we can express

$$\begin{aligned} (E_e + E'_e)^2 \sin^2\left(\frac{\theta_e}{2}\right) &= ((E_e - E'_e)^2 + 4E_e E'_e) \sin^2\left(\frac{\theta_e}{2}\right) \\ &= \omega^2 \sin^2\left(\frac{\theta_e}{2}\right) + Q^2 = q^2 \sin^2\left(\frac{\theta_e}{2}\right) + Q^2 \left(1 - \sin^2\left(\frac{\theta_e}{2}\right)\right) \\ &= q^2 \sin^2\left(\frac{\theta_e}{2}\right) + Q^2 \cos^2\left(\frac{\theta_e}{2}\right). \end{aligned}$$

Finally, we get

$$v_{TL} = -\frac{\rho}{\sqrt{2}} \sqrt{\rho + \tan^2\left(\frac{\theta_e}{2}\right)}.$$

Similar to the calculations we performed for the transverse part, the transverse-transverse interference part results in

$$v_{TT} = -\frac{1}{2}\rho.$$

In summary, we achieve

$$v_0 = 4E_e E'_e \cos^2\left(\frac{\theta_e}{2}\right) \quad (3.41)$$

$$v_L = \rho^2, \quad (3.42)$$

$$v_T = \frac{1}{2}\rho + \tan^2\left(\frac{\theta_e}{2}\right), \quad (3.43)$$

$$v_{TL} = -\frac{\rho}{\sqrt{2}} \sqrt{\rho + \tan^2\left(\frac{\theta_e}{2}\right)}, \quad (3.44)$$

$$v_{TT} = -\frac{1}{2}\rho. \quad (3.45)$$

All of these results are in agreement with those of [Don+17].

The Mott cross section can be simplified with the ERL

$$\left(\frac{d\sigma}{d\Omega'_e}\right)_{\text{Mott}} = \frac{4\alpha^2 E_e'^2}{Q^4} \cos^2\left(\frac{\theta_e}{2}\right).$$

That is equivalent to the Mott cross section, that one can find for example in [Pov+15]. When using Eq. (3.40) we get

$$\left(\frac{d\sigma}{d\Omega'_e}\right)_{\text{Mott}} = \frac{\left(\alpha \cos\left(\frac{\theta_e}{2}\right)\right)^2}{\left(2E_e \sin^2\left(\frac{\theta_e}{2}\right)\right)^2},$$

which is equivalent to the Mott cross section, that one can find for example in [Don+17].

### 3.10 Multipole Decomposition of the Response Functions

To determine the nuclear response functions it is necessary to make some assumptions. First of all, we assume that the initial oxygen nucleus is in the ground state  $J = 0$ . Furthermore we assume that the recoil nuclei are also in the ground state. In addition, we will restrict the consideration to the electric and Coulomb multipoles up to the quadrupole contribution. Following the development in [RD89; FDM19] the response functions  $R^K$  in the cms can be written in terms of Legendre polynomials  $P_k(\cos\theta_\alpha)$  and associated Legendre polynomials  $P_k^n(\cos\theta_\alpha)$

$$R_L = P_0(\cos \theta_\alpha) (|t_{C0}|^2 + |t_{C1}|^2 + |t_{C2}|^2) \quad (3.46)$$

$$\begin{aligned} &+ P_1(\cos \theta_\alpha) \left( 2\sqrt{3}|t_{C0}||t_{C1}| \cos(\delta_{C1} - \delta_{C0}) + 4\sqrt{\frac{3}{5}}|t_{C1}||t_{C2}| \cos(\delta_{C2} - \delta_{C1}) \right) \\ &+ P_2(\cos \theta_\alpha) \left( 2|t_{C1}|^2 + \frac{10}{7}|t_{C2}|^2 + 2\sqrt{5}|t_{C0}||t_{C2}| \cos(\delta_{C2} - \delta_{C0}) \right) \\ &+ P_3(\cos \theta_\alpha) \left( 6\sqrt{\frac{3}{5}}|t_{C1}||t_{C2}| \cos(\delta_{C2} - \delta_{C1}) \right) + P_4(\cos \theta_\alpha) \left( \frac{18}{7}|t_{C2}|^2 \right), \end{aligned}$$

$$R_T = P_0(\cos \theta_\alpha) (|t_{E1}|^2 + |t_{E2}|^2) \quad (3.47)$$

$$\begin{aligned} &+ P_1(\cos \theta_\alpha) \left( \frac{6}{\sqrt{5}}|t_{E1}||t_{E2}| \cos(\delta_{E2} - \delta_{E1}) \right) \\ &+ P_2(\cos \theta_\alpha) \left( -|t_{E1}|^2 + \frac{5}{7}|t_{E2}|^2 \right) \\ &+ P_3(\cos \theta_\alpha) \left( -\frac{6}{\sqrt{5}}|t_{E1}||t_{E2}| \cos(\delta_{E2} - \delta_{E1}) \right) \\ &+ P_4(\cos \theta_\alpha) \left( -\frac{12}{7}|t_{E2}|^2 \right), \end{aligned}$$

$$R_{TL} = \left\{ P_1^1(\cos \theta_\alpha) \left( 2\sqrt{3}|t_{C0}||t_{E1}| \cos(\delta_{E1} - \delta_{C0}) \right) \right. \quad (3.48)$$

$$\begin{aligned} &\quad - 2\sqrt{\frac{3}{5}}|t_{C2}||t_{E1}| \cos(\delta_{C2} - \delta_{E1}) \\ &\quad \left. + \frac{6}{\sqrt{5}}|t_{C1}||t_{E2}| \cos(\delta_{C1} - \delta_{E2}) \right) \end{aligned}$$

$$\begin{aligned} &+ P_2^1(\cos \theta_\alpha) \left( 2|t_{C1}||t_{E1}| \cos(\delta_{C1} - \delta_{E1}) + 2\sqrt{\frac{5}{3}}|t_{C0}||t_{E2}| \cos(\delta_{E2} - \delta_{C0}) \right. \\ &\quad \left. + \frac{10}{7\sqrt{3}}|t_{C2}||t_{E2}| \cos(\delta_{C2} - \delta_{E2}) \right) \end{aligned}$$

$$+ P_3^1(\cos \theta_\alpha) \left( 2\sqrt{\frac{3}{5}}|t_{C2}||t_{E1}| \cos(\delta_{C2} - \delta_{E1}) + \frac{4}{\sqrt{5}}|t_{C1}||t_{E2}| \cos(\delta_{C1} - \delta_{E2}) \right)$$

$$\left. + P_4^1(\cos \theta_\alpha) \left( \frac{6\sqrt{3}}{7}|t_{C2}||t_{E2}| \cos(\delta_{C2} - \delta_{E2}) \right) \right\} \cdot \cos \phi_\alpha,$$

$$R_{TT} = -R_T \cos(2\phi_\alpha), \quad (3.49)$$

with  $t_{C,EJ}$  representing the reduced Coulomb and electric matrix elements and  $\delta_{C,EJ}$  the phases of each multi-pole operator. Both are functions of  $q$  and  $\omega$  [FDM19]. Using the *Rodrigues' Formula* the Legendre Polynomials can be represented as [AS74]

$$P_n(x) = \frac{1}{2^n n!} \frac{d^n (x^2 - 1)^n}{dx^n}, \quad (3.50)$$

$$P_n^m(x) = (-1)^m (1 - x^2)^{\frac{m}{2}} \frac{d^m P_n(x)}{dx^m}. \quad (3.51)$$

In Appendix A the nuclear response functions up to octupole contributions in terms of the Legendre and the associated Legendre polynomials can be found. The nuclear response functions are also given there in terms of sine and cosine.

### 3.11 Representing the Electric Matrix Elements $t_{EJ}$ with the S-Factor Components $S_{EJ}$

The differential cross section for the photo-disintegration can be established with similar calculations as done for the electro-disintegration cross section [FDM19]

$$\left[ \frac{d\sigma}{d\Omega_\alpha^{\text{cms}}} \right]_{(\gamma,\alpha)} = \left( \frac{M_\alpha M_{12\text{C}}}{4\pi W} \right) \frac{p_\alpha^{\text{cms}}}{\hbar c} \left( \frac{\alpha}{E_\gamma^{\text{cms}}} \right) R_T^{\text{cms}}(\gamma, \alpha), \quad (3.52)$$

with  $E_\gamma^{\text{cms}} = \omega^{\text{cms}} = |\vec{q}_{\text{cms}}|$  and the transverse nuclear response function  $R_T^{\text{cms}}(\gamma, \alpha)$ . In the lab frame, where  $\mathcal{P}_{16\text{O}}^{\mu\text{lab}} = (M_{16\text{O}}, \vec{0})$ , the invariant mass  $W$  can be expressed as

$$W = \sqrt{M_{16\text{O}}(M_{16\text{O}} + 2E_\gamma^{\text{lab}})}.$$

For the limit  $Q^2 \rightarrow 0$ , the transverse nuclear response function in electro-disintegration is the same as the one in photo-disintegration [FDM19]. This allows us to establish a connection between the electric matrix elements  $t_{EJ}$  and the S-factor components  $S_{EJ}$ . In the following the transverse nuclear response function in cms is simplified as  $R^T$ .

First, we use the reciprocity theorem Eq. (2.32) to represent the response function in terms of the S-factor. In the center of mass frame of the carbon- $\alpha$  system we have  $p_{16\text{O}}^{\text{cms}} = q^{\text{cms}} = E_\gamma^{\text{cms}}$ . As we have  $l = 0$  for all initial and final particles the reciprocity equation for the photo-disintegration reaction is

$$\frac{\sigma(\gamma, \alpha)}{\sigma(\alpha, \gamma)} = \frac{1}{2} \frac{|\vec{p}_\alpha^{\text{cms}}|^2}{|E_\gamma^{\text{cms}}|^2}.$$

So, the differential cross section of the  $\alpha$ -capture reaction can be written as

$$\left[ \frac{d\sigma}{d\Omega_\gamma^{\text{cms}}} \right]_{(\alpha,\gamma)} = \left( \frac{M_\alpha M_{12\text{C}}}{2\pi W} \right) \frac{E_\gamma^{\text{cms}}}{\hbar c} \left( \frac{\alpha}{p_\alpha^{\text{cms}}} \right) R_T. \quad (3.53)$$

Finally, the transverse nuclear response function in cms becomes

$$R_T = \frac{\hbar c \cdot p_\alpha^{\text{cms}} \cdot W}{2\alpha \cdot E_\gamma^{\text{cms}} \cdot M_\alpha M_{12\text{C}}} \cdot 4\pi \left[ \frac{d\sigma}{d\Omega_\gamma^{\text{cms}}} \right]_{(\alpha,\gamma)}. \quad (3.54)$$

By comparing the coefficients of equations Eq. (2.28) and Eq. (3.54), the electric matrix elements in the limit  $Q^2 \rightarrow 0$  can be expressed as functions of the S-factor components:

$$t_{E1} = \sqrt{\frac{\hbar c}{2\alpha} \cdot \frac{p_\alpha^{\text{cms}}}{E_\gamma^{\text{cms}}} \cdot \frac{W}{M_\alpha M_{12\text{C}}} \cdot \frac{S_{E1}(E_\alpha^{\text{cms}})e^{-2\pi\eta}}{E_\alpha^{\text{cms}}}}, \quad (3.55)$$

$$t_{E2} = \sqrt{\frac{\hbar c}{2\alpha} \cdot \frac{p_\alpha^{\text{cms}}}{E_\gamma^{\text{cms}}} \cdot \frac{W}{M_\alpha M_{12\text{C}}} \cdot \frac{S_{E2}(E_\alpha^{\text{cms}})e^{-2\pi\eta}}{E_\alpha^{\text{cms}}}}, \quad (3.56)$$

where the Sommerfeld number depends on the cms energy  $\eta \equiv \eta(E_\alpha^{\text{cms}})$ .



---

---

# CHAPTER 4

---

## Accelerator and Experimental Setup

In this chapter we will discuss the entire experimental setup for the planned measurement of the S-factor of the reaction  $^{12}\text{C}(\alpha, \gamma)^{16}\text{O}$ . First, we will get a brief overview of the electron accelerator – MESA. Then, we will get a broad overview of the experiment MAGIX located at MESA. For this purpose, we will take a look on all important components of MAGIX at MESA that are required for the S-factor measurement. At the end of the chapter the different phases of the planned measurements are summarized.

### 4.1 Mainz Energy-Recovering Superconducting Accelerator

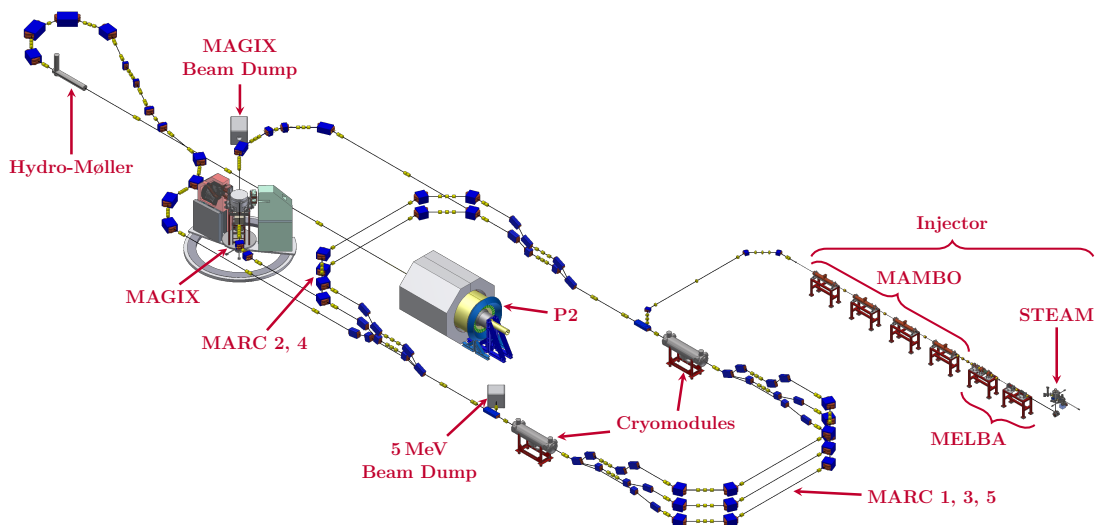


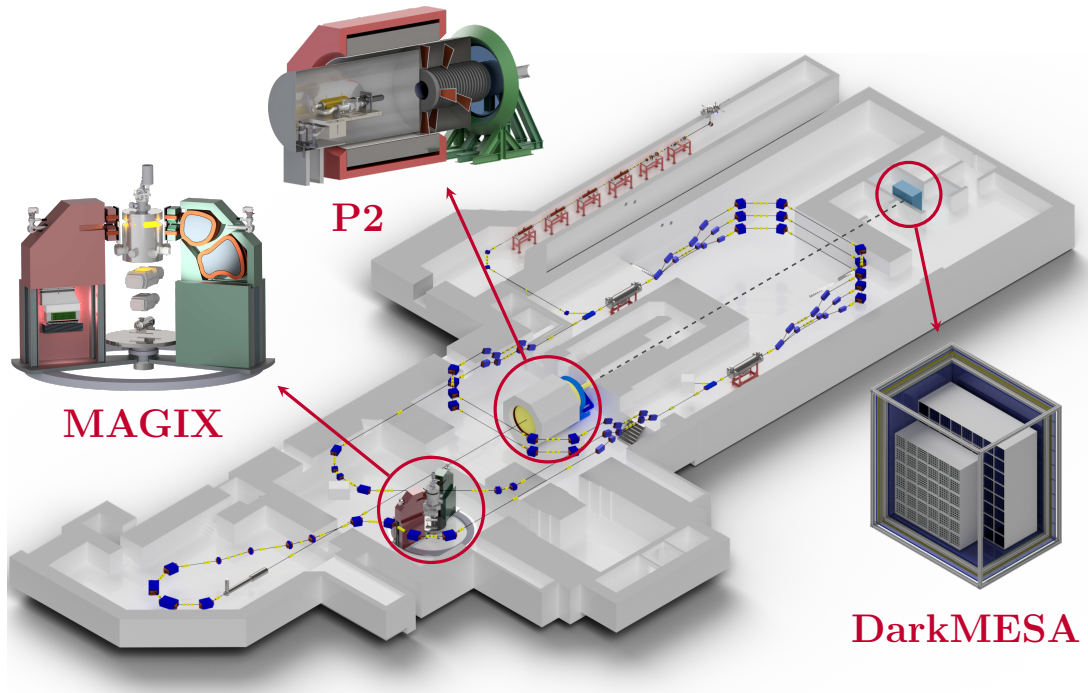
Figure 4.1: Overview of the structure of MESA [Sim21b; Bec+18].

The Mainz Energy-Recovering Superconducting Accelerator (MESA) (see Figs. 4.1 and 4.2) is an electron accelerator that is currently under construction at the Johannes Gutenberg University in Mainz, Germany. It will be a low-energy continuous wave recirculating electron linac for various particle and nuclear physics experiments [Hug+20; Hug+17]. Figure 4.1 shows the structure of MESA. The first part of MESA is the injector. The photo-source STEAM (Small Thermalized Electron Source At Mainz) supplies a polarized or an unpolarized electron beam with an energy of 100 keV and a beam current of up to 1 mA. The photo-source is followed by the low-energy beam transport system MELBA (MESA Low-energy Beam Apparatus). MELBA contains a spin manipulation system and a chopper-buncher section for longitudinal matching. This acts as a beam preparation for the normal-conducting booster linac MAMBO (Milliamper Booster) [Hug+20]. MAMBO accelerates the electrons to an energy of 5 MeV. Then, the electron beam is directed into the main linac. The main linac consists of two accelerator units, the cryomodules MEECs (MESA Enhanced ELBE Cryomodules). The electrons are boosted up to 25 MeV per cryomodule. Depending on the mode and maximum beam energy, the electrons are transported through the recirculation arcs 1-5, MARCs (MESA Arcs) 1-5. With this method, the electron beam can be boosted several times in the two MEECs [Sim21b; Led20; Hug+20; Hug+17]. MESA can be operated in two different operating modes.

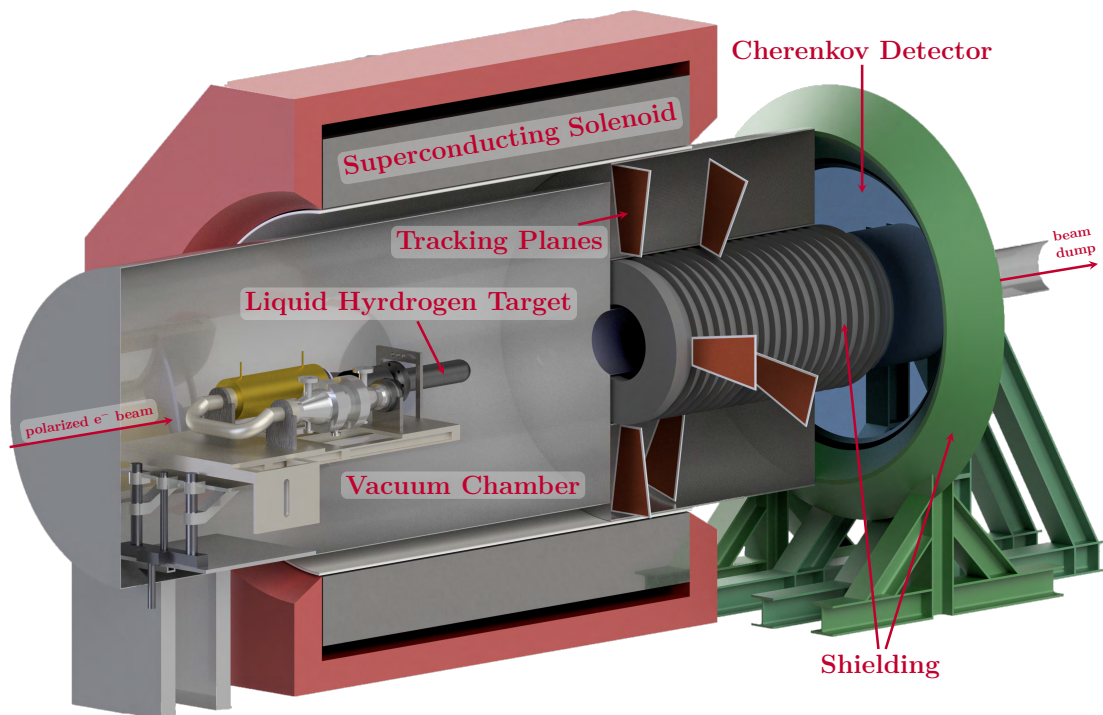
In the External Beam (EB) mode MESA provides a polarized beam with an energy of up to 155 MeV and a current of 150  $\mu\text{A}$  [Hug+20; Bec+18]. For this purpose, the beam is accelerated up to 3 circulations using MARCs 1-5. With each circulation the beam gains a total energy of up to 50 MeV. Once the beam has reached its final energy, it is extracted from the accelerator part and guided through the Hydro-Møller to the P2 experiment. The Hydro-Møller measures the polarization of the electron beam [Kem20]. After the P2 experiment, the beam is dumped in a dedicated beam dump [Sim21b; Hug+20; Bec+18].

In the Energy Recovering (ER) mode a polarized or an unpolarized electron beam with an energy of up to 105 MeV and a current of 1 mA is used for the internal target experiment – Mainz Gas Injection Target Experiment (MAGIX). For this purpose only MARC 1-3 are used and the beam will be extracted from the accelerator part after a maximum of 2 recirculations. Downstream of MAGIX the beam is guided back to the accelerator part with a phase shift of  $180^\circ$ . Due to this phase shift, the beam transfers its energy back to the cryomodules and is thereby decelerated. Up to 25 MeV can be returned to the MEEC each pass with this method. After a maximum of 4 recirculations, the remaining 5 MeV electron beam is stopped in the 5 MeV beam dump. This energy recovery allows the high electron current of 1 MeV. In the second expansion stage of MESA, 10 mA electron beam current should also be possible in ER mode [Sim21b; Led20; Hug+20]. Instead of using the energy recovery, the electron beam can also be dumped after MAGIX. For energies up to 30 MeV a current of up to 1 mA can be achieved. At higher energies, the beam dump can only be used if the beam current is reduced. This is necessary on the one hand because otherwise the energy demand would be too high without ER mode, and on the other hand the beam dump is not designed for such a high radiation exposure [Sim21b, section 11.2 & 13.1].

For a full description of MESA, the components and the two modes please refer to the PhD thesis [Sim21b]. For further information also refer to [Hug+20; Led20; Hug+17].



**Figure 4.2:** Overview of the MESA hall [Sim21a] and the experiments P2 [Sor17], DarkMESA [Chr21b] and MAGIX [Sch+21]. The dashed line intends to clarify the alignment of P2 and DarkMESA. P2 and DarkMESA carry out their measurements in EB-mode. In ER mode MAGIX will carry out its measurements.



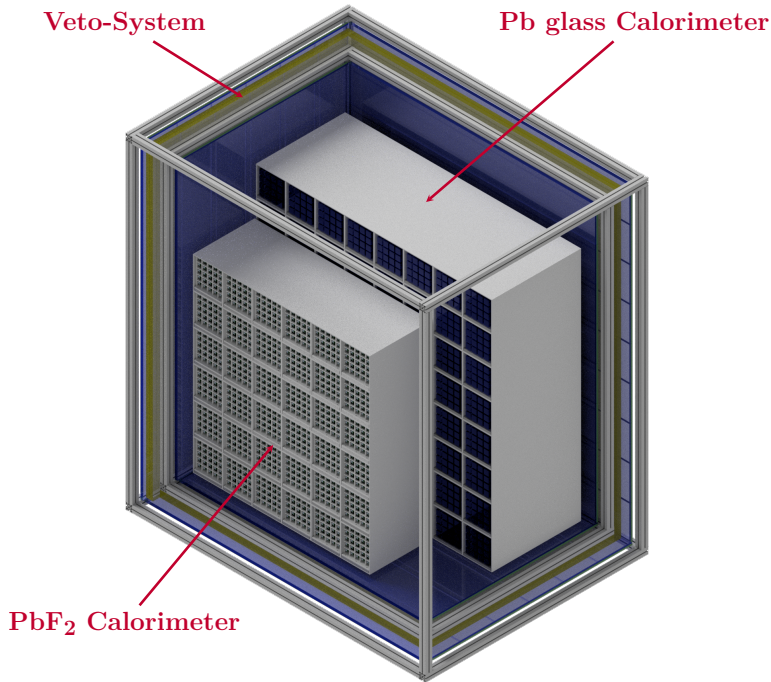
**Figure 4.3:** Overview of the P2 experiment [Bec+18; Sor17; Ber16]

#### 4.1.1 The P2-Experiment

The P2 experiment plans to measure the weak mixing angle  $\theta_W$ , which is also known as Weinberg angle. For this purpose, the parity-violating asymmetry of the elastic electron-proton scattering is determined. Figure 4.3 shows the layout of the

P2 experiment. The longitudinally polarized electron beam scatters on the liquid hydrogen  $H_2$  target. The scattered electrons are focused towards the Cherenkov detector by the P2 spectrometer magnet. To measure the scattering angle, the track of the electrons can be reconstructed by detecting them in the tracking planes. The shield located radially around the outgoing beam is used to remove the background created by bremsstrahlung or Møller scattering. After the P2 experiment the beam will be dumped. [Ber21; Kem20; Bec+18; Sor17; Ber16].

#### 4.1.2 The DarkMESA Experiment

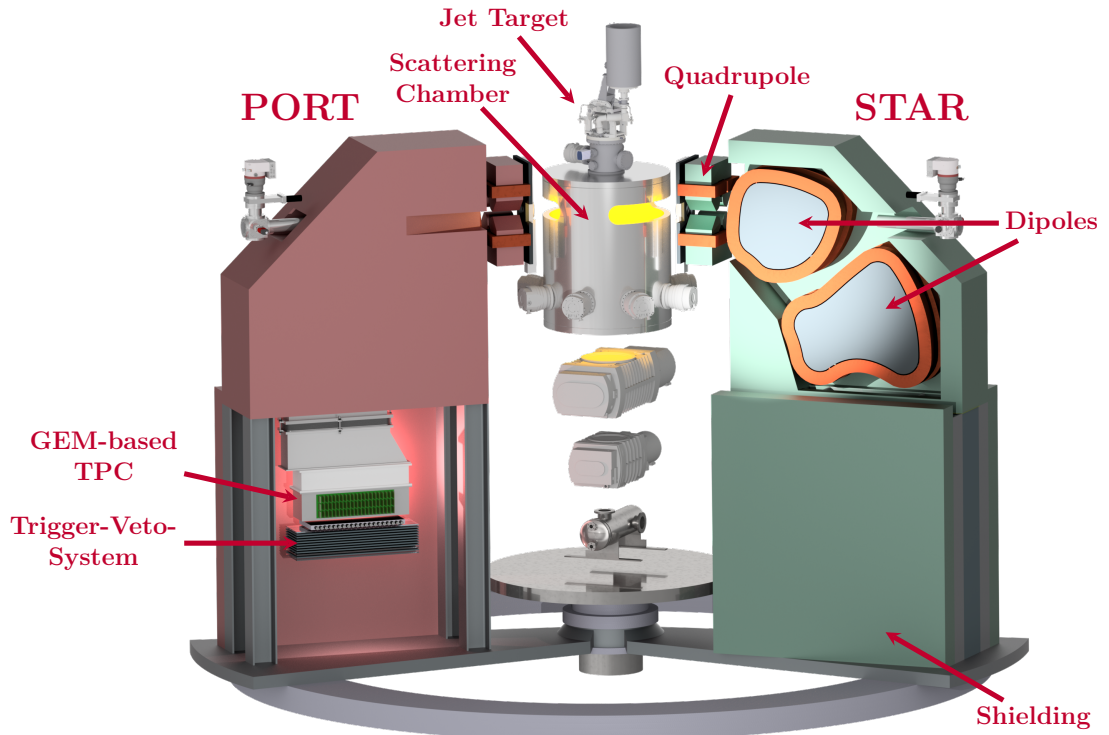


**Figure 4.4:** Overview of the DarkMESA experiment [Chr21b].

The parasitic dark sector experiment DarkMESA will use the high-power beam dump of the P2 experiment as target. DarkMESA (see Fig. 4.4) was developed to detect Light Dark Matter (LDM). In the simplest model, the LDM couples to a massive vector particle, the so-called dark photon. This can potentially be generated in the P2 beam dump by a process analogous to bremsstrahlung. The dark photon then decays into a pair of dark matter particles. In the DarkMESA calorimeter, a fraction of these dark matter particles can scatter on electrons which can then be detected. The experiment is placed outside of the accelerator hall well shielded from beam-related SM particles. To get rid of any kind of remaining background, a versatile veto system is built around the calorimeter. The beam time of the DarkMESA experiment is coupled to that of P2, which means that DarkMESA can take data while P2 is operated in EB mode. In addition, DarkMESA can carry out background measurements when MESA is switched off [Chr21b; Chr21a; Chr+20a; Chr+20b; Dor+19; Dor+18].

## 4.2 Mainz Gas Injection Target Experiment

The Mainz Gas Injection Target Experiment (MAGIX) is a versatile fixed-target experiment, which will be operated at the new electron accelerator MESA in Mainz (see Figs. 4.2 and 4.5). One of the core components of MAGIX is the internal gas jet target [Gri+18; Gri18; Aul21] which allows the production of a directional gas



**Figure 4.5:** Overview of the MAGIX experiment. The gas jet target will be mounted in the center of the scattering chamber. A pumping station will remove the injected gas. Two identical magnetic spectrometers STAR and PORT will be able to rotate around the interaction point. The spectrometers will be connected to the vacuum of the scattering chamber. They will focus the scattered electrons with a quadrupole-dipole-dipole configuration onto the focal plane located in the GEM-based TPC. At the entrance to the TPC, the scattered electrons will pass for the first time through a vacuum termination foil. After the TPC, a veto trigger system will be mounted for triggering, timing and particle identification purposes [Sch+21].

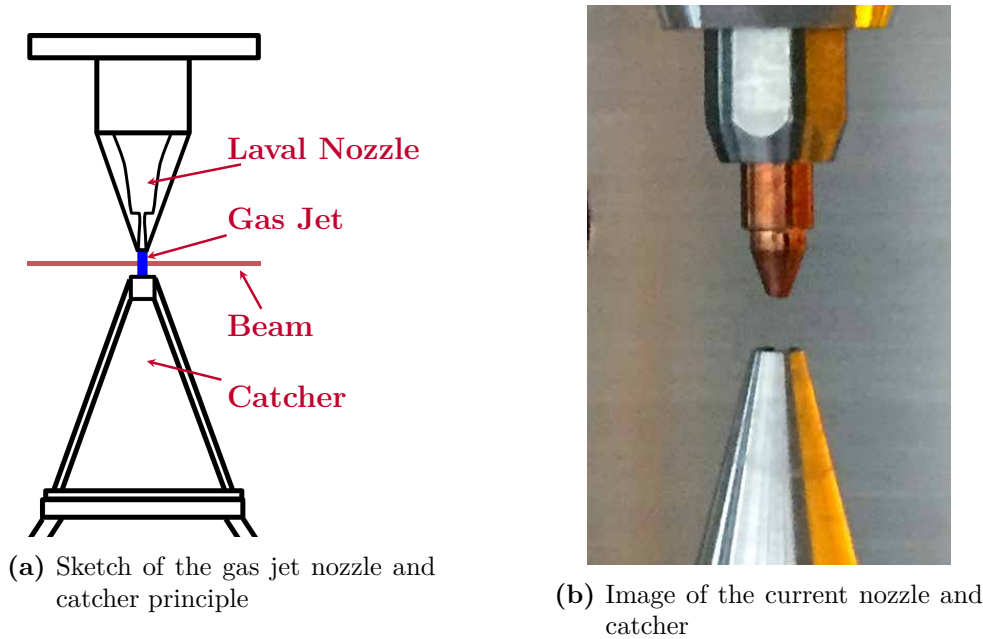
beam through the vacuum of the scattering chamber with an areal particle density of  $>10^{18}$  atoms/cm<sup>2</sup>. With the combination of the low-density target and the high beam current, MAGIX will reach a luminosity of  $\mathcal{O}(10^{35} \text{ cm}^{-2} \text{ s}^{-1})$ . To determine the electron scattering kinematics with a high momentum and angular resolution, MAGIX will use two magnetic spectrometers equipped with GEM-based TPCs in the focal planes (see Fig. 4.5). This will allow to study processes with very low cross sections at a small momentum transfer. The rich physics program of MAGIX ranges from low-energy nuclear physics up to new physics searches, like [Sch+21]

- Measurement of electromagnetic form factors, charge radii, e. g. proton radius;
- Search for particles of the dark sector [Mer+11];
- Reaction studies of astrophysical relevance, e. g.  $^{12}\text{C}(\alpha, \gamma)^{16}\text{O}$ ;
- Nuclear few-body physics experiments.

In the following we will get a brief overview of the different parts of MAGIX. We will focus on the components necessary for measuring the S-factor of  $^{12}\text{C}(\alpha, \gamma)^{16}\text{O}$ .



## 4.2.1 Gas Jet Target

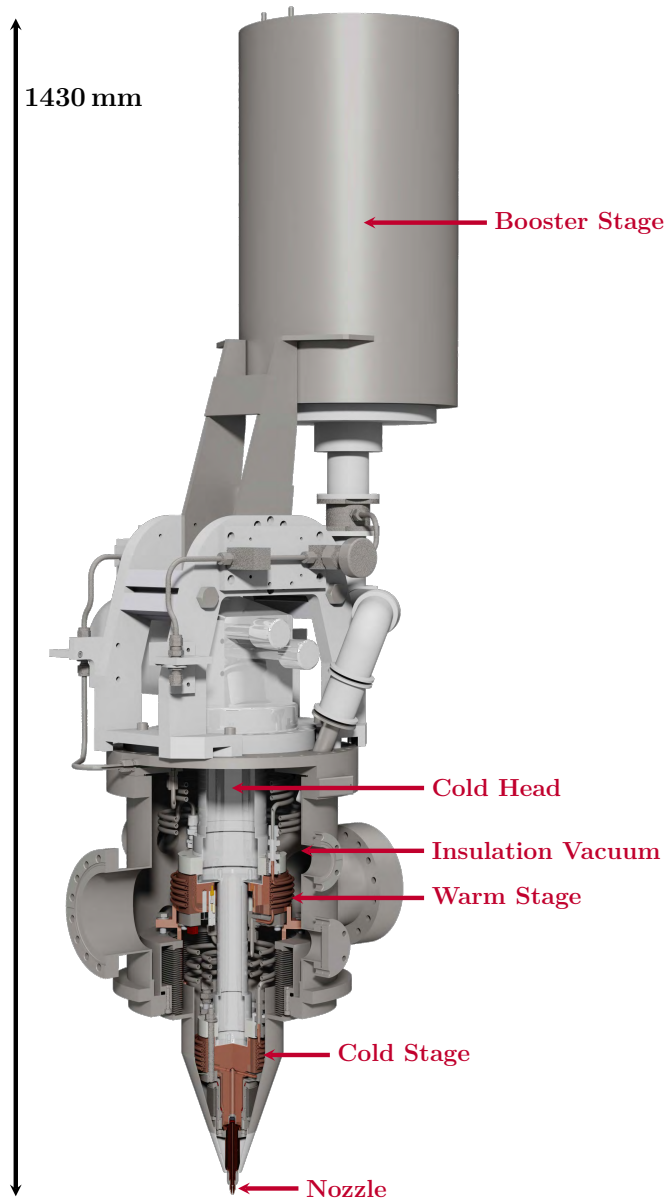


**Figure 4.6:** The left sketch (a) shows the direction of the gas jet and the interaction with the electron beam. The picture on the right (b) shows the real nozzle and catcher used in the current test stand [Aul21].

The basic idea of the MAGIX target is to scatter electrons on gas that passes through the vacuum chamber without breaking it. For this purpose a thin, directed gas jet is generated by a Laval nozzle. Due to the hydrodynamics of the Laval nozzle, the gas becomes a directed laminar flow in the scattering chamber. This laminar flow is stable for a few millimeters. The gas jet is then sucked off by a catcher placed closely behind the nozzle. The electron beam does not pass any window which is an advantage of such a system. The background is reduced significantly in comparison to other fixed target experiments where for instance target walls are inducing multiple scattering. Figure 4.6 shows a schematic sketch of the nozzle and catcher as well as the gas jet.

To ensure a constant areal particle density, the pressure and temperature in front of the nozzle and the volume flow must be controlled. In addition, the design of the Laval nozzle, especially the narrowest inner and the outlet diameter, play a decisive role. The calculations in [Gri+18] show that the stagnation conditions with the lowest temperature-pressure combination should be selected. This allows MAGIX to be carried out with a low volume flow.

The MAGIX Gas-Jet Target (see Fig. 4.7) has been developed by the University of Münster, Germany. It essentially consists of a cold head in which the gas is adiabatically cooled. The cold head has two cooling stages, the warm stage can cool down to 28 K and the cold stage can cool down to 8 K. The nozzle is mounted directly behind the cold stage. In order to protect the cooling stages against thermal influences, the cold head is surrounded by an insulation vacuum. For a large volume flow, the pre-mounted booster stage can pre-cool the gas down to 77 K. An areal particle density of  $>10^{18}$  atoms/cm<sup>2</sup> is planned for the MAGIX. For hydrogen this can be achieved by cooling the gas with a gas flow of 2400 L/h to 40 K. Two different nozzles with an inner diameter of 0.5 mm were also designed by the University of Münster, Germany. For hydrogen, an areal particle density of  $5 \times 10^{18}$  atoms/cm<sup>2</sup> is calculated for the nozzle with an outlet diameter of 1 mm

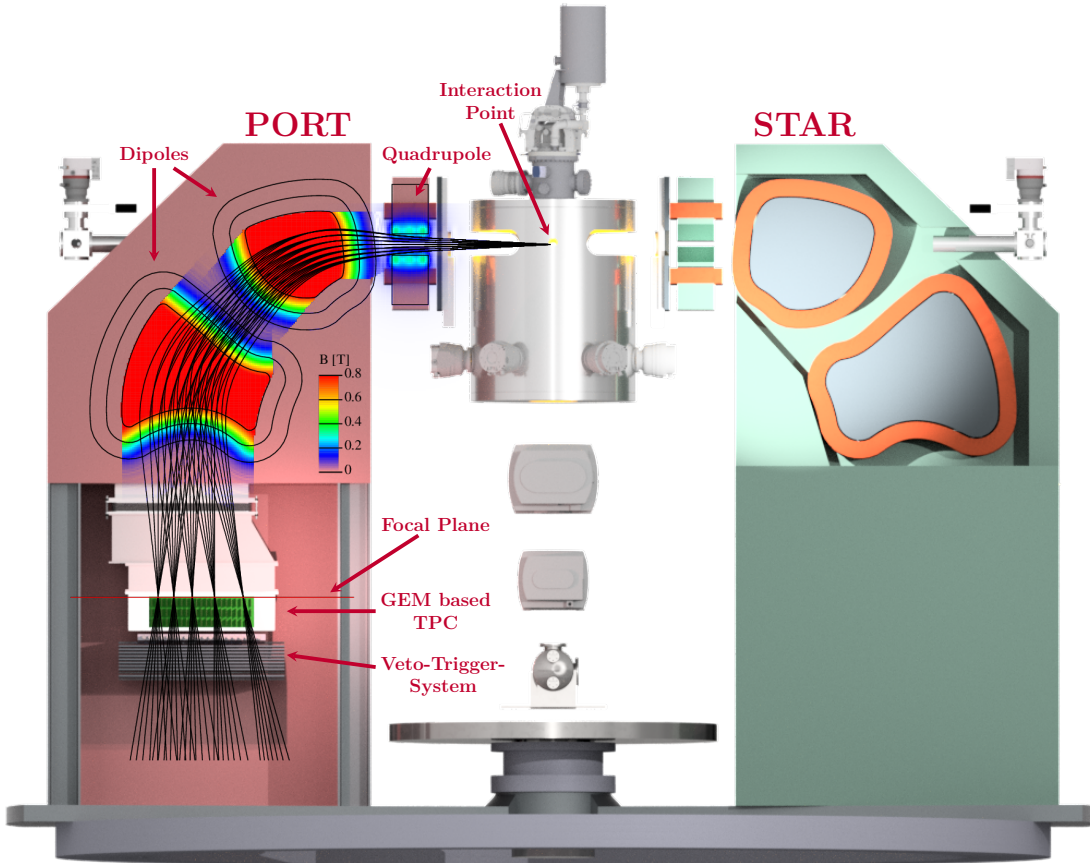


**Figure 4.7:** Overview of the MAGIX gas jet target. With the cold head in combination with the booster stage hydrogen at a gas flow of 2400 l/h can be cooled down to 40 K to achieve an areal particle density of  $>10^{18}$  atoms/cm<sup>2</sup>. In addition to hydrogen, helium, argon, oxygen, nitrogen, xenon and various other types of gas can be used. The nozzle can easily be exchanged to adapt to the respective gas. [Gri18; Gri+18; Aul21].

and an areal particle density of  $2.4 \times 10^{18}$  atoms/cm<sup>2</sup> with an outlet diameter of 2 mm [Gri+18]. In addition to hydrogen, helium, argon, oxygen, nitrogen, xenon and various other types of gas can be used. The nozzle can easily be exchanged to adapt to the respective gas [Sch+21; Gri+18; Gri18; Aul21].

At the A1 experiment at Mainz Microtron (MAMI), the MAGIX Gas-Jet Target has been commissioned in 2018 and 2019 to carry out first experiments. During these measurements it was also possible to calculate the areal particle density. With a hydrogen flow rate of 2400 l/h at a temperature of 40 K with the nozzle of 1 mm outlet diameter a maximum areal particle density of  $1.6 \times 10^{18}$  atoms/cm<sup>2</sup> was achieved [Sch+21]. Further information on these measurements can be found in [Sch+21].

## 4.2.2 STAR/PORT Spectrometer Setup



**Figure 4.8:** Overview of the MAGIX spectrometer. The spectrometers are connected to the vacuum of the scattering chamber. Each of the two identical magnet-spectrometers focuses the scattered electrons with a quadrupole-dipole-dipole configuration on the focal plane. The focal plane is located in entrance of the GEM-based TPC. After the TPC, a veto trigger system is mounted for triggering, timing and particle identification purposes [Sch+21].

To detect the scattered electrons or recoiled positrons, MAGIX uses two identical, high-resolution magnetic spectrometers, called PORT and STAR (see Fig. 4.8). The spectrometers are named after the nautical terms port and starboard and are colored with their respective colors, red and green. Because of the limited height of the hall and the relatively high position of the beamline, the spectrometers had to be oriented upside down. They are mounted on opposite sides of the beam line on a system that allows the spectrometers to be rotated around a common pivot point. The adjustable angle range per spectrometer relative to the beam line is  $15^\circ$  to  $165^\circ$ . The spectrometers are connected to the scattering chamber via a sliding seal so that they share the same vacuum. This means that the scattered electrons and recoiled particles enter the spectrometer without passing through any foil. By reversing the polarity of the magnets, the particle species (electron or positron) can be selected.

The quadrupole-dipole-dipole configuration of the spectrometer maps the absolute momentum, the scattering angles, the azimuthal angle and the vertex to the Cartesian coordinates  $x$  and  $y$  of the focal plane, and the corresponding entrance angles at the focal plane. Thus, it is possible to replace the direct measurement of these quantities with the potentially simpler position measurement in the focal plane. For this purpose a GEM-based TPC located in the focal plane is used. For

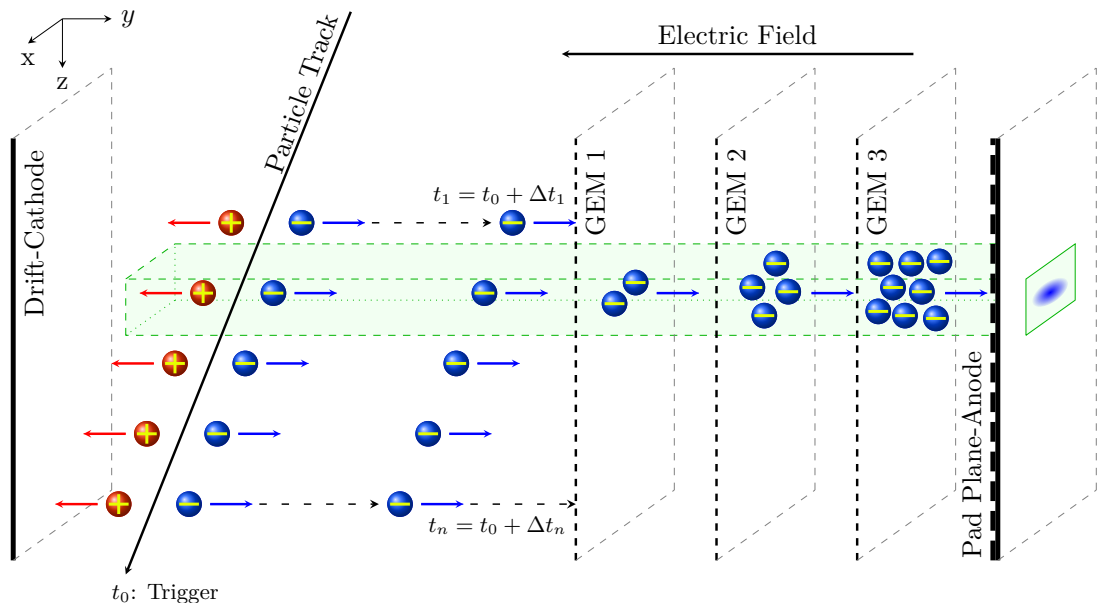


example, the absolute momentum is proportional to the x-coordinate in the focal plane to the first order.

At the entrance of the TPC is the first vacuum foil, which the particles must pass. With this setup, MAGIX will be able to determine the scattering parameters with high resolution. Due to the small material budget, effects such as multiple scattering will be reduced to a minimum. From the three-dimensional reconstructed track of the TPC in combination with the trigger, timing and particle information of the trigger veto system, the scattering parameters such as momentum, polar and azimuthal angles, and interaction point can be determined with high precision. The goal is to achieve a relative momentum resolution of  $\frac{\Delta p}{p} < 10^{-4}$  and an angular resolution of  $\Delta\Theta < 0.05^\circ$  [Sch+21; Sch21].

### Time Projection Chamber

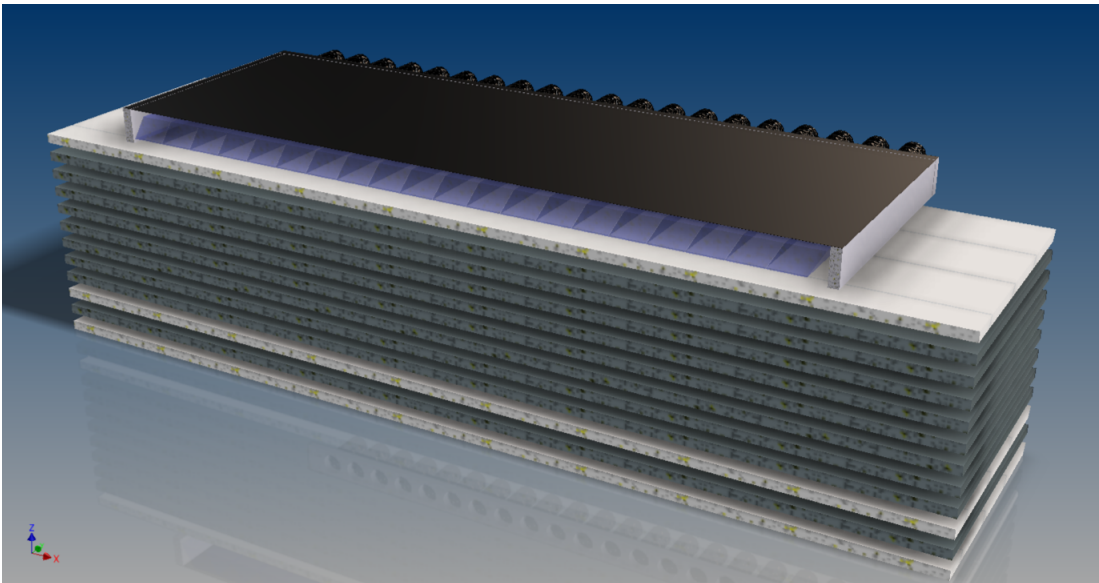
In the focal plane of each spectrometer, the electrons or positrons can be detected with a Time Projection Chamber (TPC). Figure 4.9 illustrates the working principle of a TPC. When a charged particle passes through the active gas volume of the TPC, the counting gas is ionized. The free electrons drift along the external electric field towards the amplification stage, where several Gas Electron Multipliers (GEMs) are located. The  $x$ - and  $z$ -coordinates of the electrons read out at the pad plane anode represent the projected coordinates of the particle track. Using the information of the drift time  $\Delta t_i = t_i - t_0$ , with  $t_i$  time of detection and  $t_0$  trigger time, the  $y$ -coordinate of the track can be calculated. Thus, a 3D track of the traversing particle can be reconstructed. To minimize the material budget, the MAGIX TPC is designed with a novel open field cage concept. For more information please refer to the doctoral thesis of P. Gülker [Gül21].



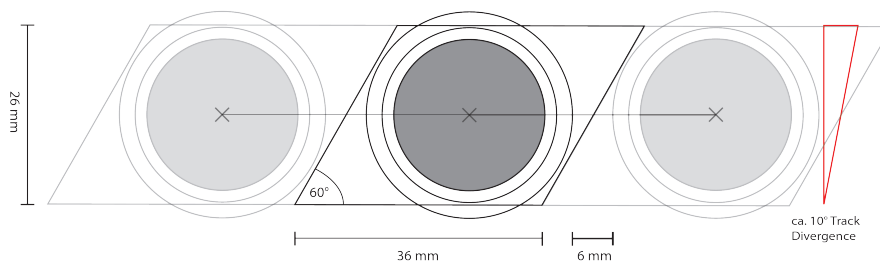
**Figure 4.9:** Schematic principle of a TPC. The green block illustrates the trajectory of an electron-ion pair. The ion drifts along the homogeneous electric field to the cathode while the electron drifts to the anode. Each electron generates a signal after the time  $t_i = t_0 + \Delta t_i$  with trigger time  $t_0$  and drift time  $\Delta t_i$ . The electrons are amplified in each GEM layer. The pad plane anode can be used to read the signal and determine the  $x$  and  $z$  coordinate. The drift time corresponds to the  $y$  coordinate.

## Trigger Veto System

The trigger veto system of each spectrometer (see Figs. 4.10 and 4.12a) consists of one layer of segmented plastic scintillator detectors (trigger layer) followed by several layers of additional scintillation detectors (veto layers) with lead absorbers in between. The trigger layer at the top of the system is used for fast time and energy loss measurements. Through its segmentation, it provides the basic hit and position measurement used to trigger the readout of the TPC. The cross section of the segments of the trigger layer is diamond-shaped at an angle of  $60^\circ$ . This results in an overlap in the electron direction. The maximum track divergence for the scattered electrons is  $10^\circ$ , therefore it is ensured that every particle in the trigger layer is detected (see Fig. 4.10) [Ste21a; Ste21b]



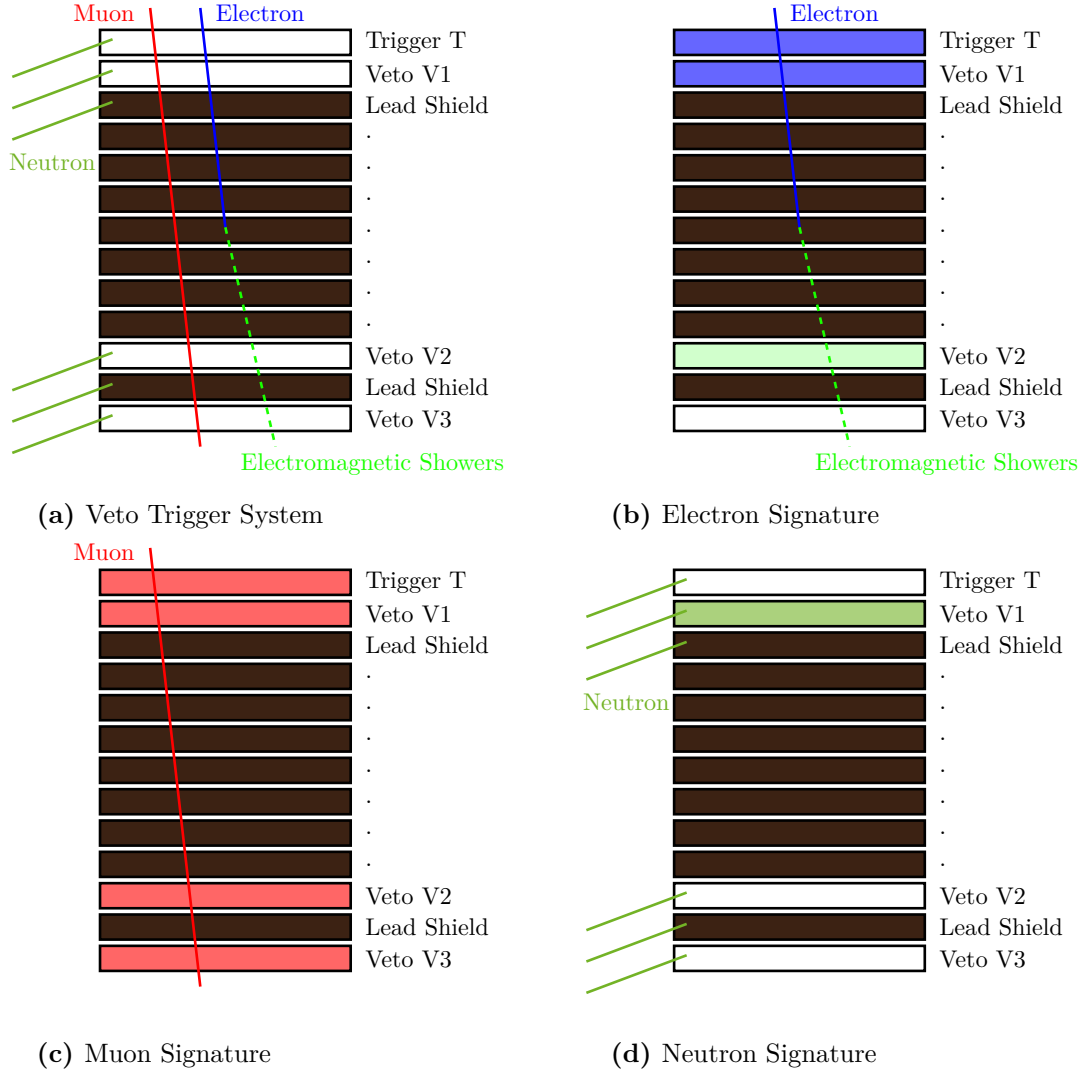
**Figure 4.10:** The current 3D concept of the trigger veto system. The cross sections of the segments of the trigger layer are diamond-shaped.



**Figure 4.11:** Schematic view of the cross section of the diamond-shaped segmented trigger layer.

Because of the special upside down design of the spectrometers, a more complex trigger veto system is required. The challenge is to separate the scattered electrons and cosmic muons as they point in the same direction. Beam-induced neutrons can also produce signals in the scintillating layers. All of these particles leave characteristic signatures in the trigger veto system (see Figs. 4.12b to 4.12d) that allow them to be distinguished. The electrons generate signals in the trigger layer T and the first veto layer V1. The electrons and electron-induced electromagnetic showers are stopped in the absorber which extremely reduces the probability of obtaining an electron-induced signal in veto layers 2 and 3. Unlike electrons, muons pass through the absorbers and thus generate a signal in the scintillator

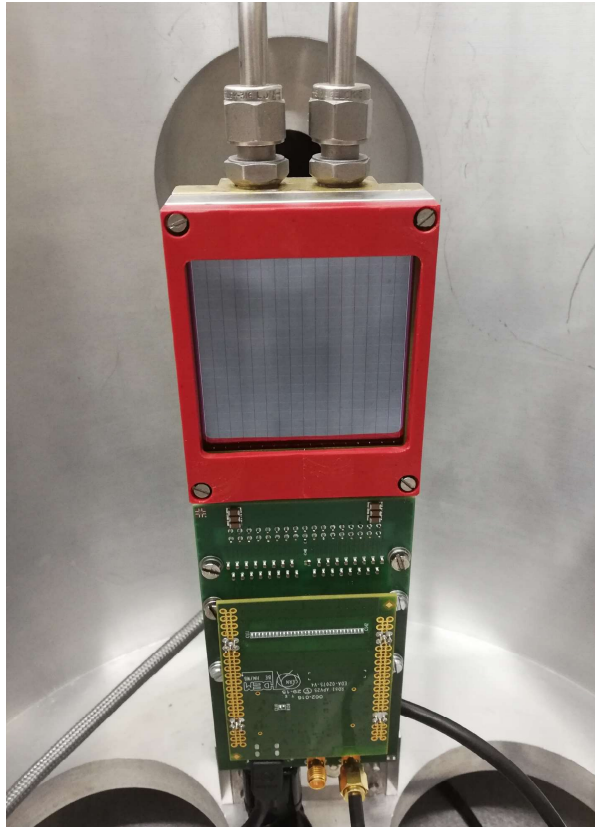
layers V2 and V3. Therefore, we can distinguish between electrons and muons using the Signal V2&V3 as a muon veto. The beam-induced neutrons do not have a specific direction, they usually produce a signal in only one scintillator layer. The probability that a neutron will produce a signal in both layers T and V1, or that a random coincidence will occur, is low [Ste21a; Ste21b].



**Figure 4.12:** Schematic diagram of the MAGIX trigger veto system [Ste21a]. Figure (a) shows the detector setup with all types of particles passing through the detector system. Figures (b)-(d) show the different detector signatures for the different types of particles expected at MAGIX [Ste21a].

### 4.2.3 Silicon Strip Recoil Detector Array

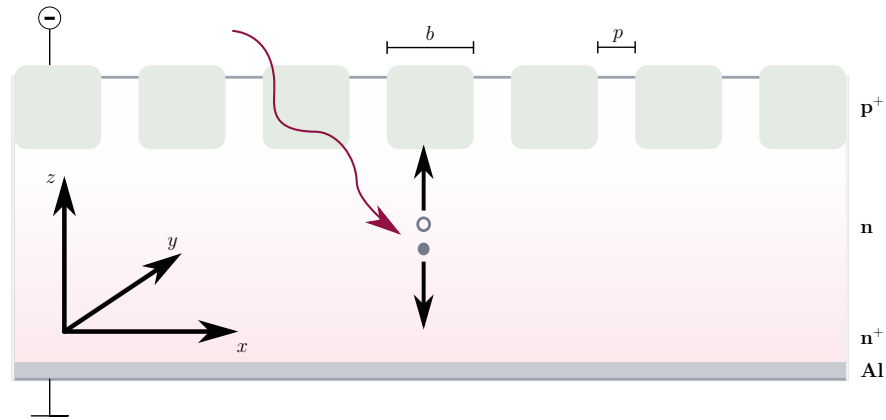
For some of the reactions that will be studied at MAGIX it is necessary to detect the recoil particles as well as the scattered electrons. In order to detect the various low energy heavy charged recoil particles, such as protons, deuterons or  $\alpha$ -particles, MAGIX needs a further detector system in addition to the spectrometers. For this purpose, a silicon strip detector array will be used. It consists of several silicon strip detectors mounted inside the scattering chamber at a distance of about 30 cm from the interaction point. Each individual silicon strip detector can be rotated independently around a common pivot point [Gei21a; Gei21b; Gei20].



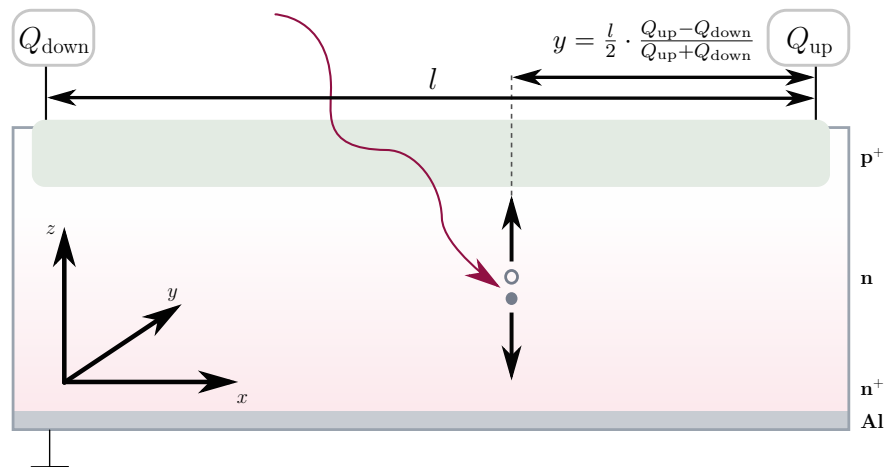
**Figure 4.13:** *Prototype of the silicon strip detector inside a test chamber. The cooling element is attached to the back of the X1 detector. The readout electronics is connected to the bottom of the detector. The APV25-S1 front-end chip is located at the front of the electronics [Gei20].*

The detector array is in development. Currently, the X1 silicon strip detector from *Micron Semiconductor Limited* is being tested. The active area of the detector is  $50 \times 50 \text{ mm}^2$  and consists of a  $995 \mu\text{m}$  thick silicon substrate. On the back side, the substrate is covered with an aluminum layer. 16 p-doped silicon stripes with a width of  $b = 3 \text{ mm}$  are embedded in the n-doped substrate. Those have no entrance window, so, the low-energy charged heavy particles do not lose energy before depositing their energy in the substrate. The X1 is mounted on a holder which contains a cooling element, actively cooled by circulating ethanol (see Fig. 4.13). Since the dark current of the silicon strip detector is extremely temperature dependent, the signal to noise ratio will be optimized by cooling the detector down to  $-20^\circ\text{C}$ . So, for example,  $\alpha$ -particles with a kinetic energy of 300 keV can be detected. The readout of the detectors is based on the APV25-S1 frontend chip [Gei21a; Gei20].

The silicon strip detector is a semiconductor detector. Its working principle is illustrated in Fig. 4.14. In addition to the classical pn-junction a further, more heavily n-doped layer, called  $n^+$ , is added to semiconductor detectors. This enlarges the space-charge region, which forms the sensitive area of the detector. The detector is operated via external voltage in the reverse direction. If a charged particle hits the space charge region of the detector, electron-hole pairs are produced. The holes drift along the external applied field to the stripes where the position will be detected representatively for the incoming particle. With the APV, the generated signal charge can be read out on both sides of each stripe. Due to the charge asymmetry, the  $y$ -position along the stripes can be determined. The



(a) Cross section of the silicon strip detector across the stripes. The 16 p-doped stripes have a width  $b$  and a distance  $p$  to each other. The spatial resolution in x-direction is  $\sigma_x = \frac{b}{\sqrt{12}}$ .



(b) Cross section of the silicon strip detector along the stripes. The produced signal charge is measured for each stripe at both ends. Due to the charge asymmetry, the position along the y-direction can be determined.

**Figure 4.14:** Schematic representation of the working principle of the silicon strip detector. When a charged particle hits the space charge region of the substrate, electron-hole pairs are produced. The electron drifts along the applied external field to the anode, whereas the hole drifts to the stripes. [Gei20].

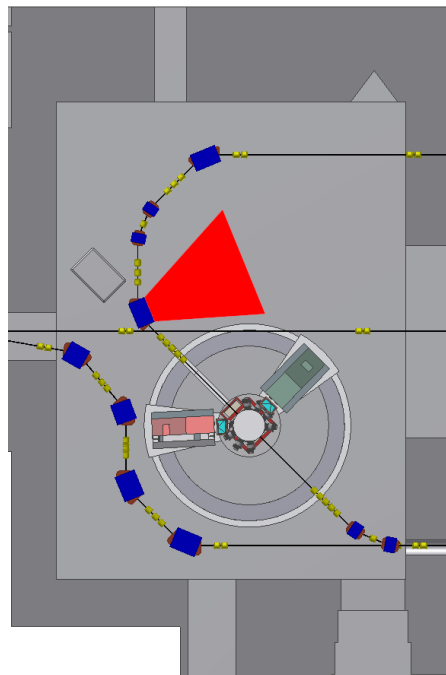
segmentation into stripes of equal width  $b$  results in a discrete resolution in the x-direction. The spatial resolution is  $\sigma_x = \frac{b}{\sqrt{12}}$ . To start data acquisition, the APV needs a trigger signal. Since most particles are stopped in the detector, no additional detector can be used to generate the required trigger signal. A novel frontend-board has been developed by the MAGIX group to obtain this timing information from the electrons that are generated simultaneously with the holes and drift to the anode. The board amplifies the electrons and passes them to the APV as a trigger signal [Gei20; Gei21a].

#### 4.2.4 Zero Degree Tagger – FORE

In a later phase of the MAGIX program, another electron detector, the beam-line integrated Zero Degree Tagger (ZDT), will be available. This detector is designed to measure electrons scattered at an angle of less than  $0.5^\circ$  in the forward direction. The planning of this detector has begun and in this section we will only get a basic

idea of this detector. The project name of this detector is FORE, like the nautical term for forward, analogous to the two spectrometers.

The idea is to use the first dipole magnet in the beam-line after MAGIX to focus the scattered electrons based on their energy on a plane. A detector system located in this area, not yet further defined, will then measure these electrons, in particular to determine their energy (see Fig. 4.15). For this purpose a special design of the dipole magnet is necessary. To ensure that all the requirements in the planning process are taken into account the MAGIX group is working closely with the MESA accelerator group. According to current plans, the magnet will have a deflection angle of  $45^\circ$  to guide the unscattered electrons back to the cryomodules [Led20; Sim21b]. Since the magnetic field cannot be adjusted as desired, the measuring range of this detector is restricted by a fixed reaction and beam energy. However, by reducing the beam energy, different energy ranges can be measured. In inelastic reactions, the energy of the scattered electrons is significantly lower than the beam energy. This makes it possible to separate the scattered electrons from the beam and detect them in the ZDT. For example, in the reaction  $^{16}\text{O}(e, e'\alpha)^{12}\text{C}$ , to split  $^{16}\text{O}$  into  $^{12}\text{C}$  and  $\alpha$ , the energy transfer  $\omega$  must be greater than 7.16 MeV (see Table B.5). The electrons involved in this reaction therefore lose at least 7.16 MeV and can thus be separated from the beam and measured in the ZDT.



**Figure 4.15:** *Schematic representation of the zero degree tagger for MAGIX. The first deflection magnet is used to focus the electrons onto a plane where a detector system will measure them. The red area is to represent a scattering plane to show the area of the ZDT.*

### 4.3 S-Factor Measurement at MAGIX

A general overview of the electro-disintegration experiment is given in Section 2.3.2. In this section, we will discuss the planned measurement of the electro-disintegration reaction  $^{16}\text{O}(e, e'\alpha)^{12}\text{C}$  at MAGIX. First, we will discuss the general setup

and show how the various challenges of this measurement will be solved. This particular experiment will be performed in three stages. At the end of this section, these different stages, their improvements, and their planned measurement ranges will be discussed.

### 4.3.1 Discussion of Electro-Disintegration Experiment

To determine the S-factor of the reaction  $^{12}\text{C}(\alpha, \gamma)^{16}\text{O}$ , MAGIX will measure the cross section of the electro-disintegration reaction  $^{16}\text{O}(e, e'\alpha)^{12}\text{C}$ . Therefore, electrons will scatter on an oxygen gas jet. The scattered electrons will either be measured at the MAGIX spectrometers or at the zero degree tagger in coincidence with the recoiled  $\alpha$ -particle. The  $\alpha$ -particle will be detected with the silicon strip detector array. The four-momentum of the virtual photons  $q^\mu$  and the center-of-mass energy  $W$  will be calculated by measuring the four-momentum of the scattered electron. With the measured count rate  $\dot{N}$  the cross section of the electro-disintegration reaction Eq. (3.39) can be determined using the formula [Pov+15]

$$\dot{N} = \sigma \cdot \mathcal{L},$$

where the quantity  $\mathcal{L}$  represents the luminosity. The luminosity describes the number of particle collisions per area per time. For a linear accelerator the luminosity can be calculated using the equation [Pov+15]

$$\mathcal{L} = \dot{N}_a \cdot \rho_b,$$

with the number of incoming beam particles per time  $\dot{N}_a$  and the areal particle density of the target  $\rho_b$ . For an electron accelerator we have  $\dot{N}_a = \frac{I}{e}$  with the beam current  $I$  and the elementary charge  $e$ . The areal particle density of the oxygen gas jet will be about  $2 \times 10^{18}$  atoms/cm<sup>2</sup>. So, with a beam current of 1 mA the luminosity is around  $1.25 \times 10^{34}$  cm<sup>-2</sup> s<sup>-1</sup>. Since both the jet current and the gas density are not constant, the luminosity is monitored in the final experiment.

One problem of the electro-disintegration technique is the contamination of the oxygen gas with  $^{17}\text{O}$ ,  $^{18}\text{O}$  and  $^{14}\text{N}$ . As discussed in [FDM19] the cross sections of the reaction  $^{17}\text{O}(e, e'\alpha)^{13}\text{C}$ ,  $^{18}\text{O}(e, e'\alpha)^{14}\text{C}$ , and  $^{14}\text{N}(e, e'p)^{13}\text{C}$  are several orders of magnitude larger than the cross section of the reaction  $^{16}\text{O}(e, e'\alpha)^{12}\text{C}$  close to the Gamow peak. However, Frišćić, Donnelly, and Milner also showed that by measuring the kinetic energy of the recoil particles and using the Time of Flight (ToF) information, the particles can be assigned to the corresponding reactions. Thus, an appropriate filter can be used in the analysis to eliminate the background created by this contamination. For this purpose, a minimum distance of 30 cm between the silicon strip detectors and the interaction point is recommended [FDM19].

One advantage in the MAGIX setup is, that neither the electrons nor the  $\alpha$ -particles pass any windows before they are detected by their individual detector. This allows to calculate the cms energy and the angular distribution very accurately. With the silicon strip detector array, we can measure the cross section for many different angles of the  $\alpha$ -particles. This allows us to determine the longitudinal, transverse, and interference terms and, in particular, the multipolarities E1 and E2 very accurately. The ground-state transition cross sections  $\sigma_{\text{E1}}$  and  $\sigma_{\text{E2}}$  are then analyzed using the R-matrix method and the S-factor at the Gamow peak is extrapolated. Due to its high luminosity, low background and high accuracy in determining the parameters, MAGIX is an excellent experiment for determining the S-factor close to the Gamow peak.



### 4.3.2 Discussion of the Planned Extension Stages

The experiment is performed in three successive phases. The phases are numbered, starting with number zero. In Phase 0, the MAGIX target is used at Mainz Microtron (MAMI) in Mainz, Germany. In this phase, the target system and the silicon strip detector array are tested. In addition, this phase will be used to verify the theory. Phase 1 will take place when MAGIX is commissioned at MESA. After some further testing, the S-factor measurement will be performed. Phase 2 will begin after MAGIX is expanded by the ZDT. At each stage, the aim is to improve the already achieved results. In the following, we will discuss these phases in detail.

#### Phase 0: MAGIX@MAMI

In Phase 0 we will use the MAGIX target adapted to the A1-experiment at the existing accelerator MAMI in Mainz, Germany. The electron beam will have an energy of  $E_0 = 195$  MeV and a current of  $I = 100$   $\mu$ A. The scattered electrons will be detected by one of the A1-spectrometers and the recoiled  $\alpha$ -particle will be detected by several silicon strip detectors. These will be mounted around the interaction point at a fixed angle and a distance of around 30 cm (see Fig. 4.16). The arrangement of the silicon strip detectors will be optimized to measure the E1 ground state transition, as discussed in Section 2.3.1. Since the scattering chamber at A1 is smaller than the final one at MAGIX, only a few silicon strip detectors can be installed in the chamber. Preparations for this phase have only just begun. According to current planning, the phase is to be carried out between 2022 and 2023.

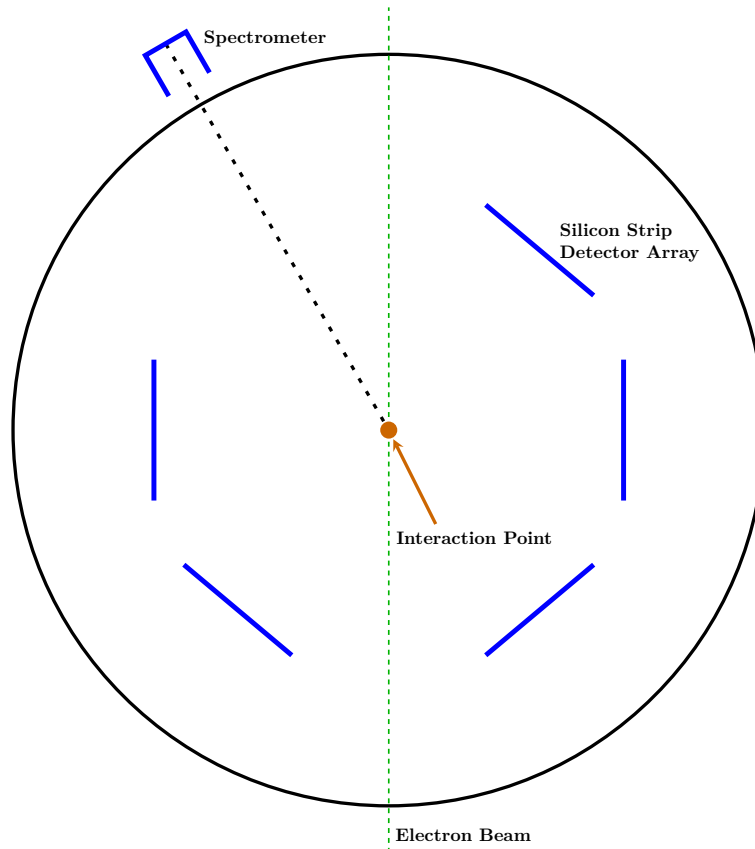
In this phase, MAGIX will determine the S-factor for cms energies in the well know region of 1.7 MeV to 2.7 MeV. The goal of this phase is to test the crucial parts of the setup as well as the reconstruction and analysis of the data. The results will be compared to the well-known existing data in this parameter range. Thus, we can test the theory of electro-disintegration and make a first contribution to the global S-factor data. In addition, we can determine the longitudinal as well as the two interference terms in this parameter range. This will allow us to improve the simulation and optimize the setup for the  $\alpha$ -particle detectors in the next phases.

#### Phase 1: MAGIX@MESA

The Phase 1 can begin after MAGIX is commissioned at MESA. The electron beam will have an energy of  $E_0 = 25 - 105$  MeV with a current of  $I = 1$  mA. We will detect the scattered electrons with the high resolution spectrometers and the  $\alpha$ -particles with the full size silicon strip detector array. The spectrometers will be placed at the lowest possible angle of around  $15^\circ$ . Each silicon strip detector can individually rotate around the interaction point. Thus, we can choose different angular arrangements to optimize the determination of the E1 and E2 ground state transition. An example of the experimental setup can be found in Fig. 4.16. With this setup, MAGIX will be able to perform the multipole expansion of the cross section of the reaction  $^{16}\text{O}(\gamma, \alpha)^{12}\text{C}$  to determine the S-factor for the E1 and E2 contributions. According to current planning, the Phase 1 is to be carried out between 2023 and 2026.

The goal of this phase is to determine the S-factor in the cms energy range above 0.9 MeV with high precision. For this purpose, we will determine the longitudinal, transverse, and both interference terms in the known range above 0.9 MeV with





**Figure 4.16:** Schematic view of the experimental setup for Phase 0 and Phase 1. In both phases the scattered electrons will be detected by a spectrometer while the  $\alpha$ -particles will be detected by the silicon strip detector array. The single silicon strip detectors will be mounted inside the scattering chamber around the interaction point.

high accuracy. In addition, we will improve the analysis software and also include a Monte Carlo simulation to extrapolate the S-factor to the Gamow peak and determine the errors. For data analysis, we will use the R-matrix method. This phase will also be used to calibrate the detectors and prepare for the Phase 2.

### Phase 2: MAGIX@MESA including Zero Degree Tagger

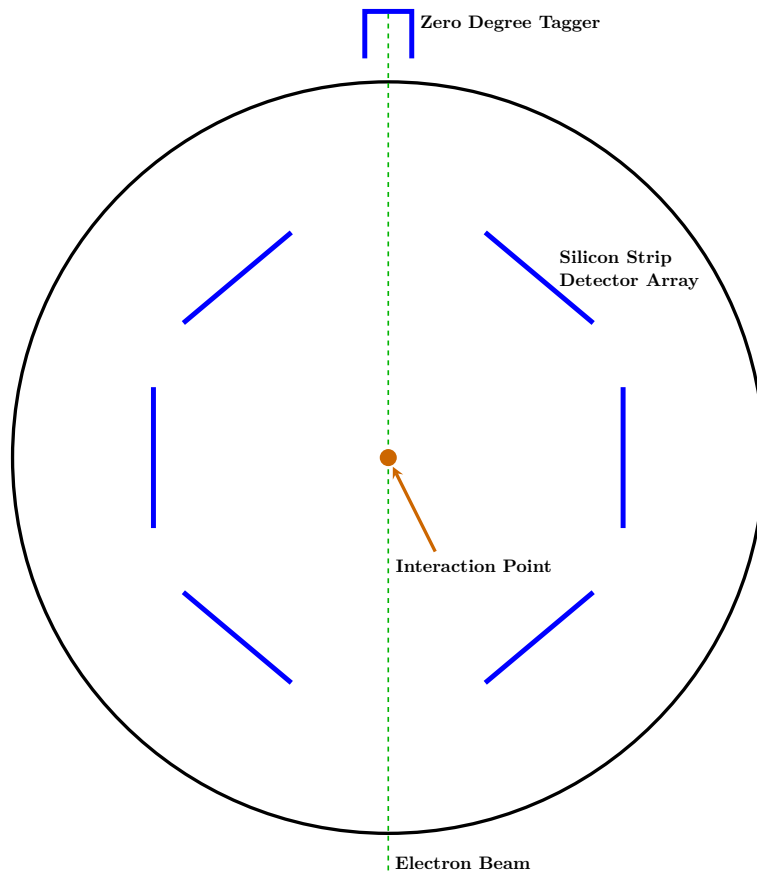
In Phase 2 we will detect the electrons with the ZDT instead of using the spectrometers. The electron beam will have an energy of  $E_0 = 25 - 105$  MeV with a current of  $I = 1$  mA. The cross section Eq. (3.39) is proportional to the Mott cross section, which can be written using Extreme Relativistic Limit (ERL) as

$$\sigma_{\text{Mott}} = \frac{\alpha^2 \cos^2\left(\frac{\theta}{2}\right)}{4E_e^2 \sin^4\left(\frac{\theta}{2}\right)}.$$

To increase the measured cross section there are two possible parameters we can change in the experimental setup. Decreasing the beam energy is limited to  $E_0 = 10$  MeV and at those energies the cross section will increase by a factor of about 100. If we use the beam integrated zero degree tagger to detect the scattered electron with scattering angles below  $0.5^\circ$  we can further increase the measured cross section with a factor of around  $10^6$ . But we have to keep in mind, that in this parameter range the ERL cannot be used. Therefore, the cross section and all included quantities need to be used in general form as derived in

Chapter 3. The associated analysis and also the simulation must be adapted accordingly. However, a major advantage at this point is that for such small angles of the scattered electrons, it is not necessary to measure the angle and we can integrate directly over the angular acceptance of the ZDT. According to current planning, the Phase 2 will be carried out between 2026 and 2032.

Since the electrons are scattered with an angle close to  $0^\circ$  the carbon- $\alpha$  system is boosted nearly in forward direction. This simplifies the experimental setup, since the silicon strip detectors can be arranged symmetrically on both sides of the beamline. Thus, also the determination of the multipolarities E1 and E2 is simplified. With this experimental setup MAGIX plans to determine the S-factor in the so far unknown cms energy area below 0.9 MeV. However, the cross section of the  $^{12}\text{C}(\alpha, \gamma)^{16}\text{O}$  decreases by one order of magnitude per 0.1 MeV step for cms energies below 1 MeV. Even though the cross section is roughly  $10^6$  larger than the cross section of Phase 1 it is still challenging to determine the S-factor at the Gamow peak. Another advantage is that the longitudinal and both interference terms are very small for small  $Q^2$ , so the transverse part can be extracted more easily. In order to determine the possible parameter range that MAGIX can access to determine the S-factor of the reaction  $^{12}\text{C}(\alpha, \gamma)^{16}\text{O}$ , a Monte Carlo simulation was performed. The results of this simulation for phases 1 and 2 are discussed in Chapter 6.



**Figure 4.17:** Schematic view of the experimental setup for Phase 2. The scattered electrons will be detected by the zero degree tagger while the  $\alpha$ -particles will be detected by the silicon strip detector array inside of the scattering chamber.

---

---

## CHAPTER 5

---

# Development of the MAGIX Slow Control System

As mentioned in the previous chapter, some of the MAGIX components exist and initial tests are being performed in different environments. For example, the current TPC and one silicon strip detector are being tested in their respective test stands. Since a common control software is needed for all these components, the development of a MAGIX Slow Control System (MXSlowControl) was necessary. One goal of this thesis was the development of a basic version of the MXSlowControl system which will be used for MAGIX at MESA. The development is in progress as new devices constantly will be added to the experimental setup. So far, most of the components used by MAGIX are controlled by MXSlowControl.

In this chapter we will discuss major parts of the MXSlowControl system, starting with the basic requirements for such a system. Then we will review the base software for MXSlowControl: EPICS. After a review of the community modules, an introduction to the development of MXSlowControl will follow, in particular to the modules created in the context of this thesis. At the end of the chapter, an example of the setup for a test stand for the silicon strip detector is given.

### 5.1 Requirements for a Slow Control System

Before looking at the requirements for the MAGIX slow control system it is necessary to understand the difference between a slow control software and a data acquisition for an experiment. Naively, you can separate both terms by looking at the data rate and the amount of data. The data acquisition rate of the detectors used by MAGIX is in the range of several kHz or MHz [Sch21]. This means that a high amount of data must be written to a hard disk in a short amount of time. In contrast, it is not necessary to read or write control parameters such as supplemental voltage, pressure, flow rate, etc. at such high rates. In a running measurement, most of these parameters remain more or less constant or are accessed only in specific time rates in the order of a few Hz. The associated communication effort and also the amount of data for slow control is therefore several orders of magnitude smaller than for data acquisition.

Based on the over 40 years of experience with the MAMI accelerator and experiments performed there, we are able to prepare a list of requirements for the MAGIX slow control. First of all, it is necessary to separate the system, and thus the software, into smaller parts. Each of these parts have to run independently and control its own devices. Communication between these decentralized parts and with a Graphical User Interface (GUI) is necessary. This ensures the stability of the system, as tasks can be distributed among several computers and the failure of one subsystem does not directly affect the entire system. In addition, it is desirable that the slow control systems of the accelerator and the experiments can communicate with each other. For example, the system of the experiments should be able to monitor and archive relevant accelerator data.

Furthermore, easy integration of new devices and replacement of damaged devices must be possible without large programming effort. To control all parameters one GUI for the entire control system is needed. However, instead of having only one access to the system, both, the slow control and the GUI, must be accessible from the whole experimental area.

The slow control system must allow the setting of various system states, especially the configuration of connected devices. There must also be a secure alarm system that allows automatic response to time-critical errors. In order to perform highly precise experiments, all parameters must also be properly archived so that the analysis can reproduce the system state of the available data at any time.

Although almost all devices are delivered with their own software, this is not used in the final experiment. Even with a small number of different devices, the control and readout of the devices becomes confusing. In addition, communication between the devices is usually not possible, which would pose a risk to the experiment. For example, if the pressure in the scattering chamber increases too much, the system must be able to close the corresponding valves automatically.

One possibility is to implement a custom-made slow control. This has the advantage that it can be perfectly tailored to the needs of the experiment. The disadvantage is that it is then the responsibility of the developer to ensure the long-term support. Alternatively, an already existing and well tested slow control system can be used. One of these is EPICS. As part of a bachelor thesis, a feasibility study was performed to investigate the advantages of an EPICS based slow control system for MAGIX [Emi17]. Based on the results of the investigation and the fact that both P2 and MESA will use an EPICS based slow control system, it was decided to also use EPICS as basis.

## 5.2 Experimental Physics and Industrial Control System

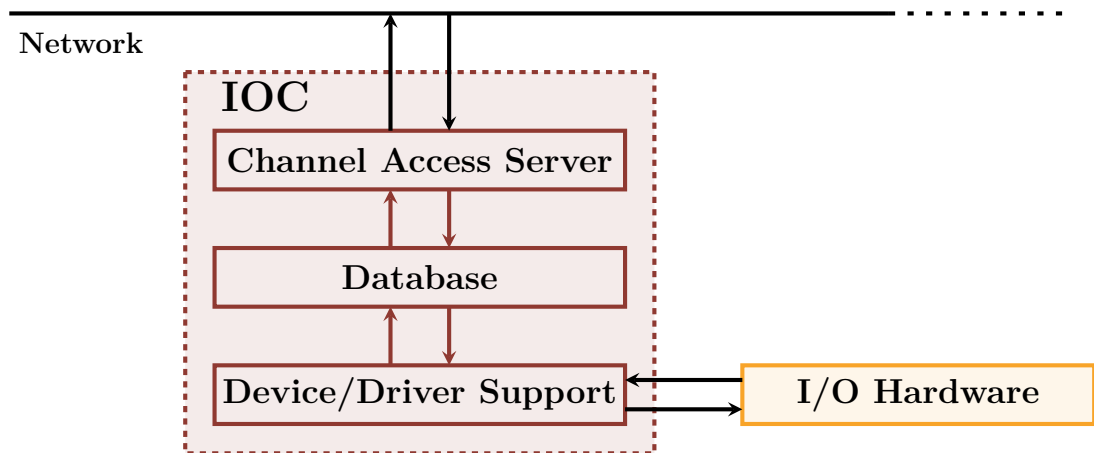
Experimental Physics and Industrial Control System (EPICS) is a decentralized, slow control system. This means that the servers and clients involved run independently of each other and communicate via a common network. It is a control system architecture supported by a huge international collaboration. It was developed by the collaboration of the Argonne National Laboratory, the Los Alamos National Laboratory, and other Departments of Energy Laboratories [EPICS21; APS04]. Due to the modular concept of EPICS, many groups worldwide work on different software toolkits and make them available to other users via the community. These can easily be installed to extend the base EPICS version. In addition to the software modules, many device drivers are also offered on the EPICS homepage, with which a device can directly be integrated into EPICS. Furthermore,

some manufacturers already provide their own EPICS drivers for their devices. This allows EPICS to be adapted very flexibly to different experiments.

In this section, we will discuss the principle of the EPICS-BASE system and how the various components work. We start with the Input-Output-Controller (IOC), which is the main program of EPICS. It handles all communication with the connected devices and provides all control parameters over the network. The core of the network communication is the communication protocol Channel Access (CA). Next, we discuss the CA client (CAC), which can write, read, and monitor data from the IOCs. At the end of this section, we will get a detailed explanation on how the channel access communication protocol works. For more information about EPICS please refer to [EPICS21; APS04].

### 5.2.1 Input-Output-Controller

The main process of EPICS is the Input-Output-Controller (IOC). Each IOC can be divided into three different major software components, the CA server (CAS), the database (DB) and the Device Support (see Fig. 5.1) [B3.14; APS04]. Processes run in the background to perform the individual tasks of the three components and enable the communication between them. After executing the IOC process the IOC shell opens. In the IOC shell the Database and Device Support needs to be configured before the CA server can be started.



**Figure 5.1:** Overview of the major software components of an IOC. The CA server provides process variables over the network. The database is a collection of EPICS records to collect, process and distribute data. The device or driver support is the interface between the database and the hardware. [APS04; B3.14].

#### Database

The database is a purely IOC-internal software module that, as the name suggests, contains the data objects. The data objects are called **records** in EPICS. Each record is of a specific type and can be identified by its unique name. Typical types are [B3.14]

- analog input (ai): used for reading double values from devices
- analog output (ao): used for writing double values to devices
- long input (longin): used for reading long values from devices
- long output (longout): used for writing long values to devices

- bool input (bi): used for reading bool values from devices
- bool output (bo): used for writing bool values to devices
- calculation record (calc): calculates a value from one or multiple other records
- calc out (calcout): calculates a value from one or multiple other records and writes this value to a device

Depending on the type, records have different properties, called fields. How a record type is structured and what tasks it can perform is specified in the database definition file (DBD). EPICS base contains a number of predefined types. These can be modified, or completely new types can be created. Typical record fields are [B3.14]

- NAME: Unique name of the record
- DESC: Contains a human readable description of the record
- TIME: Timestamp of the record
- VAL: Value of the record
- EGU: Engineering units of the value
- SCAN: Periodic time interval to automatically process the record
- INP: Input specification for Device Support
- OUT: Output specification for Device Support

Each record is an active object. The tasks provided are reading or writing data to other records or devices, checking the range of values, issuing a warning, and so on [APS04]. The scope of the tasks depends on the type of the record. For more information on record types and the assigned fields please refer to the *Record Reference Manual* at [B3.14].

The database is configured via different text files, with the format ending `.db`. It is common to create a separate db file for different devices. The db files can be loaded with different patterns. Thus, one db file is sufficient for identical devices. These can then simply be loaded with different parameters. There are multiple IOC commands to load a specific db file. In code 5.1 an example is given how a db file can be loaded with different patterns in the IOC shell. Further possibilities for loading db files will be discussed in Section 5.5.8. Code 5.2 shows an extract from the `devMPT200.db` file. Here a calc record named `$(P)$(NR)Pressure` is defined and some properties (fields) are set. The CALC field specifies how the raw value is converted to mbar. With the patterns of code 5.1 the calc record will be loaded two times with the unique names `MPT200:0:Pressure` and `MPT200:1:Pressure`.

**Code 5.1:** *IOC shell example. The db file "devMPT200.db" is loaded with different patterns using the IOC shell command `dbLoadRecords`.*

```
# Load database for two MPT200 with different configuration pattern  
dbLoadRecords("$MXSlowControl/db/devMPT200.db","P=MPT200,┘NR=0,┘BUS=ttyUSB0")  
dbLoadRecords("$MXSlowControl/db/devMPT200.db","P=MPT200,┘NR=1,┘BUS=ttyUSB1")
```

The database is responsible for collecting, processing and distributing the information for all loaded records. For this purpose, it distributes these tasks to different processes depending on the configuration. In addition, it serves as an interface

**Code 5.2:** *Example of a calc record. The input values are from the record  $\$(P):$(NR):GetPressureRaw$  in  $U\_Expo\_New$  format [MPT200]. The  $CALC$  field specifies the conversion from the raw value into mbar. The  $SCAN$  field with the value  $Passive$  defines, that the calculation is performed by an external trigger. In this case the record  $\$(P):$(NR):GetPressureRaw$  will trigger the calculation each time its value changes.*

```
record(calc, "$(P):$(NR):Pressure")
{
    field(DESC, "Pressure_in_mbar;_device_$(BUS)")
    field(INPA, "$(P):$(NR):GetPressureRaw")
    field(CALC, "FLOOR(A/100)*10^(A%100-23)")
    field(SCAN, "Passive")
    field(EGU, "mbar")
    field(PREC, 7)
    field(HOPR, 2e+3)
    field(LOPR, 1e-5)
}
```

and toolkit for creating an individually configured IOC that performs the required tasks. The database enables the other software modules to access all process information. To read and write data from devices, the database (records) makes the appropriate requests to device support [B3.14; APS04].

### The Device Support Interface

Device support, also called driver support, is the interface between the database and the input/output hardware. Any device that can be connected via USB, serial port, network, etc. can be addressed with the device support. To add new devices to EPICS, the driver for that device must be written in C or C++. This driver can be loaded into EPICS using the device support interface. Code 5.3 shows an example of such an interface. The C++ struct `devSIDET_ai` is exported to EPICS and can be accessed with the device support module. In this example two functions `init_recordSIDET` and `read_aiSIDET` have to be defined. The `init_recordSIDET` function is executed by each record associated with the SIDET device. This is performed during the initialization of the database prior to any other processes getting started. This allows loading and configuring the device driver. After all records are initialized, the database starts collecting data. Thereby the device support executes the function `read_aiSIDET` every time a record requests data from the device. In addition to the device driver, the `init_recordSIDET` and `read_aiSIDET` functions must also be written in C or C++ by the user. A major advantage of using EPICS is that the EPICS community provides some modules to simplify the integration of new devices. In addition, a large number of device supports for various devices are already available on the EPICS home page [EPICS21]. Those can easily be installed and used to control new devices. The communication structure for devices is often the same, thus a new device can be included by adjusting the device support appropriately [B3.14; EPICS21; APS04].

### CA Server

The CA server (CAS) is the interface between the database and the network. The record names are combined with its associated field names to form an individual identification string called Process Variable (PV). For example, with the configuration code 5.1 and the record code 5.2, the PVs `MPT200:0:Pressure`,

**Code 5.3:** *Example of a Device Support C interface. The C++ struct `devSIDET_ai` is exported, so that EPICS can access it. The function `init_recordSIDET` is executed from the database during the initialization of the record. To read a value from the device the function `read_aiSIDET` is executed from the Device Support [B3.14].*

```

1 #include "devSup.h"
2 #include "aiRecord.h"
3 #include "epicsExport.h"
4
5 extern long init_recordSIDET(struct aiRecord *pai);
6 extern long read_aiSIDET(struct aiRecord *pai);
7
8 struct {
9     long      number;
10    DEVSUPFUN report;
11    DEVSUPFUN init;
12    DEVSUPFUN init_record;
13    DEVSUPFUN get_ioint_info;
14    DEVSUPFUN read;
15    DEVSUPFUN special_linconv;
16 } devSIDET_ai = {
17     6,
18     NULL,
19     NULL,
20     init_recordSIDET,
21     NULL,
22     read_aiSIDET,
23     NULL
24 };
25
26 epicsExportAddress(dset, devSIDET_ai);

```

MPT200:0:Pressure.DESC, MPT200:0:Pressure.EGU, ... are created. The CA server provides access to all PVs via the network. However, this implies that a record name may not be assigned twice within the same subnet [B3.14].

## 5.2.2 CA Client

A CA client (CAC) is a program that accesses the process variables of a CA server. A client may connect to several CASs to write, read, or monitor data from PVs. Since it may be necessary for a record to request information from records loaded for processing in another IOC, each IOC may also contain a CA client. EPICS base is supplied with three different CAC programs [B3.14]:

**caput:** write a value to one PV. Example: `caput PV0 4.2`, sets the value of PV0 to 4.2.

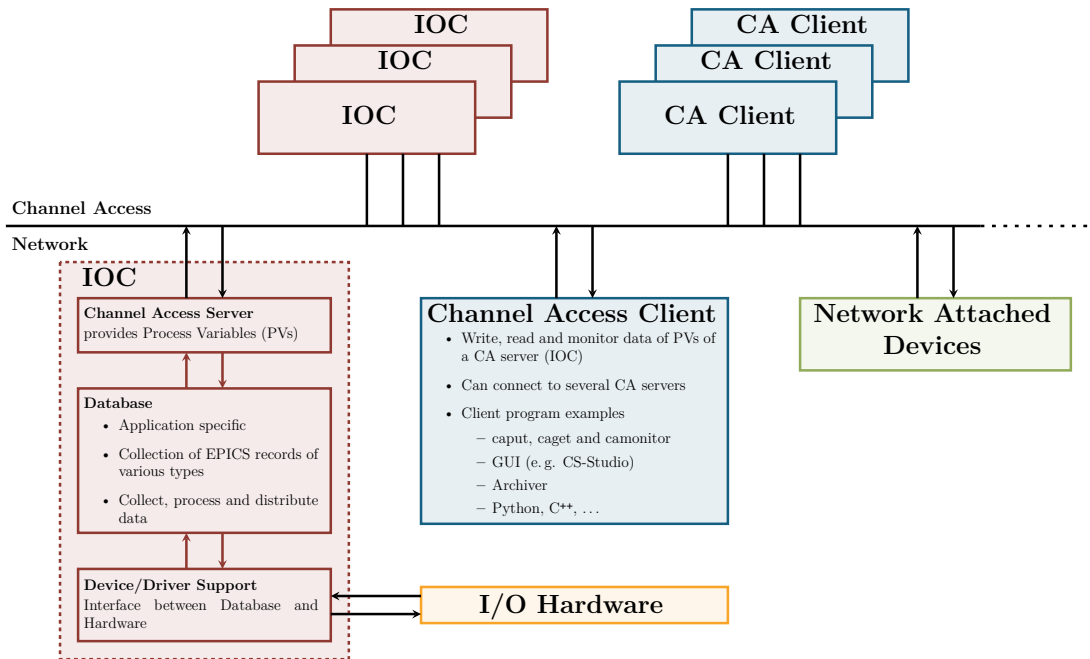
**caget:** read a value of one PV. Example: `caget PV0`, returns the value of PV0.

**camonitor:** monitors the values of one or more PVs. Whenever the value of one of the PVs changes, the new value is sent to the CA client `camonitor` together with a timestamp. Example `camonitor PV0 PV1 PV2`, monitors the value of PVs PV0, PV1, and PV2.

In addition, each GUI like CSS or archiver is a CA client application. EPICS also provides libraries to access the channel access and thus the PVs of the IOC via Python scripts or C++ programs. All these programs are then also considered as CA client processes. The application fields of a CA client can be designed very



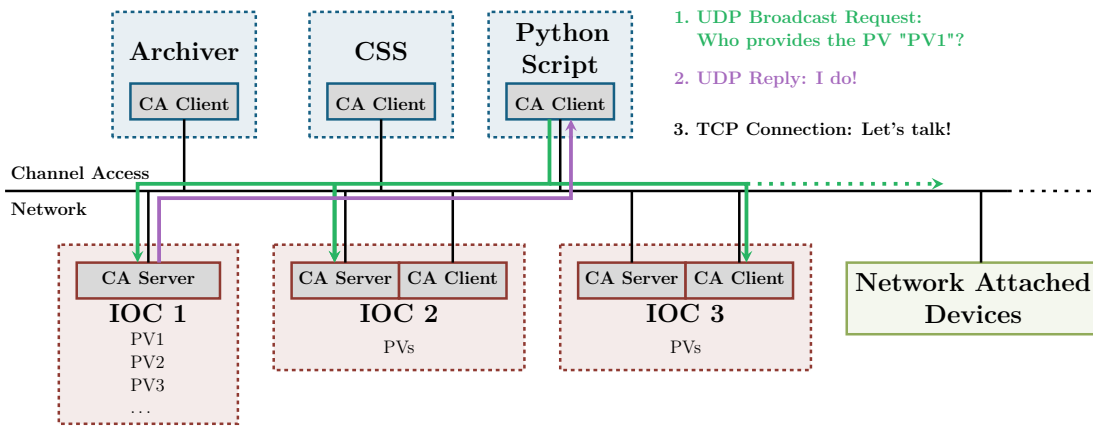
flexibly and adapted to the requirements. Figure 5.2 shows an overview of the task distribution of the IOCs and CA clients in an experimental setup.



**Figure 5.2:** Overview of the task distribution for the Input-Output-Controller (IOC) and the CA clients (CACs). An IOC always contains a CA server, but can also include CA client processes.

### 5.2.3 The Channel Access Communication Protocol

Channel Access (CA) is the communication protocol of EPICS. This enables the CA clients to communicate with the CA servers. Channel access is based on the two network protocols User Datagram Protocol (UDP) and Transmission Control Protocol (TCP). Figure 5.3 shows an overview of the CA's working principle. If a CA client application wants to access a process variable, the IOC providing this information must first be determined. The CA client sends a UDP broadcast request to the subnet asking which IOC contains a particular PV. The corresponding IOC responds via UDP. The client then opens a TCP connection to the IOC which is used for further communication. If only one value is requested or set, as in the case of `caget` or `caput`, the TCP connection is subsequently terminated. In the case of a monitoring request, as from `camonitor`, the connection is held. The IOC creates a callback function for each monitored PV, which forwards the value change of a PV directly to the client(s) [B3.14].



**Figure 5.3:** Overview of the channel access process. A client starts a search request for each PV, to find its server via UDP. After identifying the CA Server the following communication is done via TCP.

## 5.3 Community Modules

As mentioned earlier, there is a long list of modules provided by the EPICS collaboration and community to add further functionalities to the base version of EPICS. In this section, we will discuss some of these modules and highlight the benefits and, to some extent, the need for them. We will limit our discussion to the modules that are necessary for MXSlowControl.

### 5.3.1 The Asyn Module

Most devices can be controlled via low-level communication. This means that only a `socket` has to be configured. By using simple byte streams the control commands are written to the device via the `socket`. The devices process the commands and eventually send a feedback to the socket. This is where the `asynDriver` module comes into place. It provides a low-level driver interface that handles the communication with the devices. For this purpose IOC shell commands are implemented to configure the communication port of the devices. Code 5.4 shows an example of such a configuration. In addition, a set of `asyn` records is created to provide generic access to a device port. The `asynDriver` module handles all defined port accesses. For diagnostics, `asyn` provides a logging function that allows the output of error messages down to the entire byte stream, called `trace` [ASYN21]. Since communication is handled by `asynDriver`, this simplifies the implementation of new devices. Users only need to include the device-specific code in the device support and specify how the response is processed.

**Code 5.4:** Example for an `asynDriver` configuration of a serial port. The device `/dev/MPT200` is loaded with `asynDriver` and stored under the port name `L0`. Then the baud rate, data word length, parity and stop bit are configured for this port.

```
# Set up ASYN ports
# drvAsynSerialPortConfigure port ipInfo priority noAutoconnect noProcessEos
drvAsynSerialPortConfigure("L0", "/dev/MPT200", 0, 0, 0)
asynSetOption("L0", -1, "baud", "9600")
asynSetOption("L0", -1, "bits", "8")
asynSetOption("L0", -1, "parity", "no")
asynSetOption("L0", -1, "stop", "1")
```

### 5.3.2 The StreamDevice Module

A supplement for asynDriver is the StreamDevice module, which is a generic device support that implements a string stream communication interface. StreamDevice allows to include devices without recompiling the slow control system. For this purpose, StreamDevice is configured with so called protocol files that describe the communication with a device. The protocol files include one protocol for each function of the device, where the device-specific commands are defined in ASCII code. The string stream can be formatted and values can be passed using format converters similar to the C functions `printf` and `scanf`. An example of a protocol file and the corresponding record can be found in code 5.5. StreamDevice can be used for all devices that provide a simple string stream communication. To adapt StreamDevice to specific needs, it can be expanded with further format converters. If a device can not be addressed with asynDriver, StreamDevice can also be extended with further bus drivers via the StreamDevice BUS Application Programming Interface [SDev21; SDev2.8].

**Code 5.5:** *Example of a protocol and the corresponding record. The record file `devMPT200.db` is loaded into the IOC memory with the given pattern. As in code 5.4 defined `LO` is referred to the serial port of the device `MPT200`. The record field `DTYP` with the value `stream` defines that the record loads the StreamDevice driver support. The string of the field `INP` is passed to StreamDevice to configure the communication. This specifies that the command reading the pressure from the device connected to port `$PORT` is defined in the protocol `GetPressure` in the protocol file `devMPT200.proto`. The number `001` is passed as physical address to the protocol as an argument. The format of the control command `out` and the response `in` are defined in the protocol. The value is read in 6 digits decimal `U_Expo_New` format*

*IOC shell command*

```
dbLoadRecords("devMPT200.db","P=MPT200,R=,PORT=L0,ADDR=001")
```

*FILE: devMPT200.db*

```
...
record(ai, "$(P)$R)GetPressureRaw")
{
  field(DESC, "Get_Pressure_from_device_MPT200_as_U_EXPO_NEW")
  field(DTYP, "stream")
  field(INP, "@devMPT200.proto_GetPressure('${ADDR}')_${PORT}")
  field(SCAN, ".1_second")
}
...
```

*FILE: devMPT200.proto*

```
...
CMDREAD = "740"; # Pressure in [hPa], readable, size=6, u_expo_new "100023"

# Remote Control Command Format:
# Address, Action, Command, Size, Buffer, Checksum, Terminator
GetPressure {
  # Send Control Command
  out $1, "00", $CMDREAD, "02", "=", "%+<sum8>", "\r";
  # Receive Data Response
  in $1, "10", $CMDREAD, "06", "%6d", "%+<sum8>", "\r";
}
...
```

The advantage of StreamDevice in combination with asynDriver is that any device that can be controlled via a string stream can be integrated into EPICS without any programming effort. Since many devices support this, the integration of most devices is simplified by these two modules. Only the protocols for the configuration of StreamDevice and the asyn port have to be defined for each new device. However, this requires minimal programming skills, so even non-expert users can integrate and control devices in EPICS. With the inclusion of StreamDevice and asynDriver in EPICS, this point of the requirements is therefore also fulfilled.

### 5.3.3 The Autosave Module

Each time an IOC is restarted, the IOC state is reset to the default configuration defined in the db files. Even if the input records update their value by the data request itself, the output records remain in the default configuration. Regardless of the fact that the IOC does not write any data to the devices when loading the db files, no statements can be made about the set values of the data sets after the restart due to this circumstance. Unfortunately, in many cases it is also not possible to read back the value that was send to the device. This leads to a problem, because all data of an experiment must be available at any time. Naively, one way to avoid this problem is not to restart any IOC during the entire runtime of the experiment. However, this can not be avoided for a number of reasons. For example, devices may need to be replaced, which may require an IOC to be restarted. Also, IOCs or the operating system may crash due to an error and cause a restart.

The Autosave module provides a solution to this problem by adding a function to preserve the values of an IOC through a restart. Therefore, the autosave module stores the information of a set of PVs in files like an archiver. In addition, this module adds the functionality to configure the database before a process starts its tasks to the IOC. The combination of both functions allows an IOC to load the latest configuration into the database before resuming work. Autosave only saves and loads PVs that have been defined by the user. The PVs are specified in files that can be configured via patterns when loaded into the module, just like the db files [A5.7].

### 5.3.4 The PyEpics Library

In some cases it can be very useful to have small programs controlling or monitoring multiple PVs simultaneously. This can be a simple BASH script or up to a powerful C++ program. One programming language that has proven to be very capable for this task is python. Python is a dynamic scripting language. It allows object-oriented programming and can be extended by numerous libraries. Python is characterized by its flexible usability and high user-friendliness due to its clear syntax. Since the code is interpreted at runtime, it is not necessary to compile the programs in advance, as is the case with C or C++. This makes it possible to develop and run programs for various tasks very quickly and easily [Python3].

The module EPICS Channel Access for Python (pyEPICS) builds an interface between channel access and python. After loading the pyEPICS module to a python script the command-line tools `caget`, `caput` and `camonitor` can be used in python. In addition, the module extends python with an object-oriented interface by providing a PV class. The PV class exposes an EPICS process variable as a full-featured and easy-to-use python object. Among other aspects, these objects can be used to create callback functions that monitor values and perform a specific

action when needed [PyE3]. Due to the simple language of python in conjunction with CA, any user can create a variety of Python scripts to perform various tasks related to slow control. Such a script can be used, for example, to switch between different states of a system. For safety reasons, scripts can also be executed to monitor specific parameters and automatically set the system to a safe state when a critical threshold is reached.

### 5.3.5 The dev-gpio Module

The dev-gpio module can be used for IOCs running on a Raspberry Pi (RPi) or a BeagleBone to provide easy access to the general purpose input-output (GPIO) pins. An IOC shell command is used to load the lookup table for the GPIO names. After that, the voltage of a GPIO can be read or written by a bool record. Thereby the GPIO pin can be configured as active high or active low [GPIO]. An example can be found in code 5.6.

**Code 5.6:** *Example of a record with dev-gpio. The IOC shell command `GpioConstConfigure` loads the lookup table for the GPIO names of the Raspberry Pi 2 B. The bool-in record is connected to the dev-gpio module via the `devgpio` type. The input field is used to specify that the values of GPIO pin 4 are read back as active low. The `ZNAM` and `ONAM` fields define a human readable value associated with the zero and one values.*

```

IOC shell command
GpioConstConfigure( "RASPI_B+" )

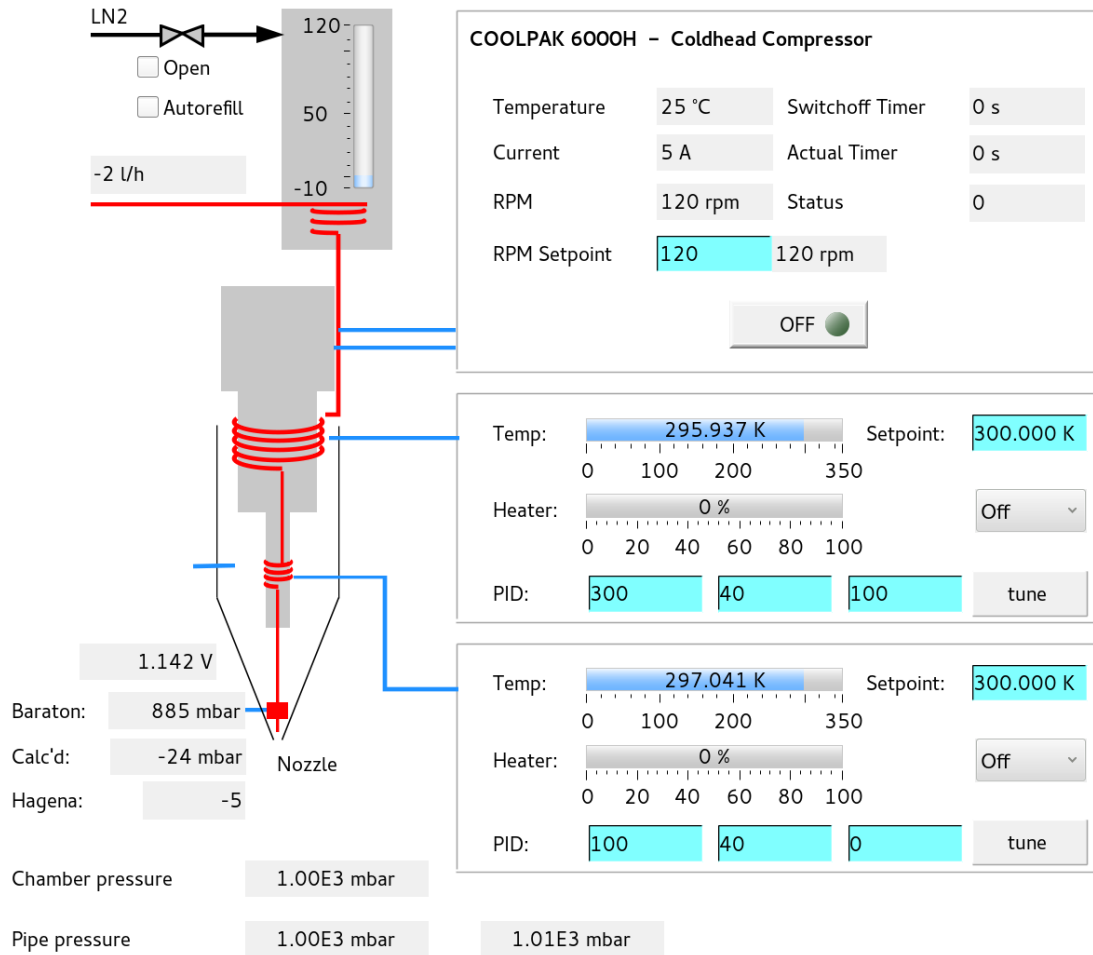
FILE: devValveControl.db

...
record(bi, "$(P):Valve15")
{
    field(DTYP, "devgpio")
    field(INP, "@GPIO4_L")
    field(SCAN, ".2_second")
    field(ZNAM, "closed")
    field(ONAM, "opened")
}
...

```

## 5.4 The Graphical User Interface - Control System Studio

There are several solutions for a Graphical User Interface (GUI) for EPICS. One of them is Control System Studio (CSS), which is a collection of various tools for controlling and monitoring data for EPICS based control systems. The main component of CSS is the Operator Interface (OPI). To create a control interface tools can be dragged and dropped onto an OPI. Each tool can easily be linked with a PV. The tools range from simple text fields to complex data browsers. CSS makes it easy to create a graphical user interface for new devices [CSS21]. Figure 5.4 shows an example of a CSS OPI. An additional feature is that scripts, such as python scripts, can be connected and executed within the OPI.



**Figure 5.4:** Example of a CSS OPI. This OPI is used to control the cold head and the booster stage. Different information like the mass flow or the pressure at different positions is also shown on this OPI.

## 5.5 MAGIX Slow Control System

The MAGIX Slow Control System (MXSlowControl) is developed based on the EPICS framework. EPICS provides a shell command called `makeBaseApp` to create a base application for this purpose. It creates a complete directory environment including an IOC application. The slow control system of the experiment can be built there, especially the db files and device supports. Like the community modules, the developed modules are standalone libraries and are installed in the EPICS base framework. They can be linked to MXSlowControl if required. In this section some of the most important developments of the MXSlowControl system of this work are presented.

### 5.5.1 Installation Script

Installing EPICS, the (community) modules and MXSlowControl on new operating systems is a complex task. In addition to creating the appropriate environment variables and integrating the libraries and applications into the system, many configuration files have to be adapted manually to link the individual parts together. Only when EPICS base and all modules are correctly linked and the corresponding variables are set up, MXSlowControl can access them. This makes the installation a time-consuming part and troubleshooting incorrect settings turns out to be difficult.

Since the installation steps are identical for debian-based operating systems, an installation script was implemented that automatically installs EPICS and the community modules. The script installs all required packages to the operating system before it starts installing EPICS. The BASH commands `sed`, `cat` and `echo` are used to customize the configuration files of EPICS and the modules. If the script is executed on a RaspberryPi, this is detected and the `dev-gpio` module will also be installed. The script is configurable. At the beginning of the script it can be selected what MAGIX modules have to be installed in addition. The script creates all definitions of the environment variables of EPICS and the modules in a file called `thisEPICS.sh`. This is installed in `/etc/profile.d/` so that the operating system sources it automatically at every reboot. After running the installation script a full working version of EPICS and the selected modules will be installed on the operating system. The installation script can be found in the appendix under code C.1.

Similar to the installation script for EPICS, an installation script for MXSlowControl has been implemented. This script builds the MXSlowControl IOC base application, links all necessary modules to it and copies all files, such as the db files, into the appropriate directories. Each MAGIX module is made with its own installation script and can be installed separately if needed. The installation script of MXSlowControl and one MAGIX module can be found in the appendix under code C.2 and C.3.

### 5.5.2 Introduction to Environment Variables

Environment variables on operating system level contain information that is available for all applications. They are used to adapt applications to the environment they are running in. Environment variables can be accessed through unique strings [Deb]. To run EPICS a set of environment variables have to be defined or modified. As mentioned in the section before, all the definitions are set in the file `thisEPICS.sh`. After sourcing this file all definitions will be loaded into the system. For example, to tell the system where to find the EPICS executable programs, such as `caget`, the command locations have to be specified in the environment variable `$PATH`. To load libraries, the system must know where to look for them. Therefore, the environment variable `$LD_LIBRARY_PATH` is used, where the library locations for EPICS, MXSlowControl and the modules have to be specified. Additional environment variables must also be defined to determine the module locations. In addition, the executables can be configured by defining the appropriate environment variables. For example, the list of destination addresses for CA clients can be specified with the EPICS environment variable `EPICS_CA_ADDR_LIST`. An example of `thisEPICS.sh` and the environment variables definition can be found in code 5.7.

### 5.5.3 Execution via System Control Manager

In most cases MXSlowControl runs permanently on the control machines. In addition, each machine must automatically resume its work after a restart. Therefore, the system must handle the execution of MXSlowControl itself. For this purpose, the Debian system manager `systemd` is used. This offers the possibility to initiate processes by system start in parallel. For this the generic system management operation `systemctl` is used [Deb]. To declare the start and end of the execution of a command running in the background, a `systemctl` configuration file must be created and loaded into the system. An example of such a file can be found in

Code 5.7: Example of a *thisEPICS.sh* file

```

#EPICS
export EPICS_ROOT=/opt/epics
export EPICS_BASE=${EPICS_ROOT}/base
export EPICS_HOST_ARCH=${EPICS_BASE}/startup/EpicsHostArch
export EPICS_BASE_BIN=${EPICS_BASE}/bin/${EPICS_HOST_ARCH}
export EPICS_BASE_LIB=${EPICS_BASE}/lib/${EPICS_HOST_ARCH}
if [ "" = "${LD_LIBRARY_PATH}" ]; then
    export LD_LIBRARY_PATH=${EPICS_BASE_LIB}
else
    export LD_LIBRARY_PATH=${EPICS_BASE_LIB}:${LD_LIBRARY_PATH}
fi
export PATH=${PATH}:${EPICS_BASE_BIN}

export HOST=$(hostname)

#Modules
export ASYN=${EPICS_ROOT}/modules/asyn
export STREAM=${EPICS_ROOT}/modules/stream
export AUTOSAVE=${EPICS_ROOT}/modules/autosave
export DEVGPIO=${EPICS_ROOT}/modules/devgpio

#PYEPICS
export PYEPICS_LIBCA=${EPICS_BASE_LIB}/libca.so

#MAGIX Modules
export FRONTEND=${EPICS_ROOT}/modules/frontend

#MXSlowControl
export MXSLOWCONTROL=${EPICS_ROOT}/MXSlowControl
export MXSLOWCONTROL_BIN=${MXSLOWCONTROL}/bin/${EPICS_HOST_ARCH}
export MXSLOWCONTROL_LIB=${MXSLOWCONTROL}/lib/${EPICS_HOST_ARCH}
export IOC=${MXSLOWCONTROL}/iocBoot/iocMXSlowControl
if [ "" = "${LD_LIBRARY_PATH}" ]; then
    export LD_LIBRARY_PATH=${MXSLOWCONTROL_LIB}
else
    export LD_LIBRARY_PATH=${MXSLOWCONTROL_LIB}:${LD_LIBRARY_PATH}
fi
export PATH=${PATH}:${MXSLOWCONTROL_BIN}

#Channel Access List
export EPICS_CA_ADDR_LIST="(<${EPICS_ROOT}/CA_ADDR_LIST.local)"

#caArchiver
export CA_ARCHIVER_LOG=${EPICS_ROOT}/log

```

code 5.8. After loading the `epics.service` file, `MXSlowControl` can be started with the command `systemctl start epics`. To initialize `MXSlowControl` at every system start, the service has to be enabled with `systemctl enable epics`.

One problem is, that the IOC shell can not be accessed if `MXSlowControl` is a pure background process. For several reasons like debugging, changing configurations or some testings this access is required. To solve this the `MXSlowControl` IOC Shell is running in a `tmux` session. The `tmux` program provides access to a number of windows from the command line. It is structured as a server-client system. When a `tmux` session is started, a server is automatically created to manage the session. Any window opened in a session can be accessed from any number of clients [Deb]. By combining `tmux` and `systemctl`, the IOC shell can be invoked from the terminal after `MXSlowControl` has been initialized. Another advantage is that this session can be accessed by multiple clients at the same time, so that the information in the IOC shell can be retrieved from different locations.



**Code 5.8:** *Example of the epics.service file*

```

[Unit]
Description=MAGIX EPICS IOC
After=network-online.target
After=epicsrepeater.service

[Service]
Type=forking
User=mxuser
Group=mxuser
TimeoutStartSec=5
ExecStart=/usr/bin/tmux new-session -ds EPICS -c
↳ /opt/epics/MXSlowControl/iocBoot/iocMXSlowControl \
    'source /etc/profile.d/epics.sh; \
    $EPICS_ROOT/MXSlowControl/iocBoot/iocMXSlowControl/st.cmd;'
ExecStop=/usr/bin/tmux send-keys -t EPICS Enter 'exit' Enter
RemainAfterExit=yes

[Install]
WantedBy=multi-user.target

```

### 5.5.4 Defining UDEV Rules

The automatic detection and initialization of the connected hardware is managed by the `udev` system. By default, new devices can only be accessed as **super user** [Deb]. This means that to control devices via `MXSlowControl`, the IOC must be run as **super user**. This is not recommended for several reasons such as security. There are several solutions for this problem. The solution used for `MXSlowControl` is to define `udev` rules to configure the hardware for its use at initialization.

After the hardware is detected, basic information is read back from the devices, called attributes. Based on the attributes `udev` configures the devices. Most devices have unique attributes and it is possible to distinguish between them. For example, to get the attributes for a particular device, the command `udevadm info -a -n /dev/$YOUR_DEVICE` can be used. An example of the result of this command can be found in code 5.9. To change the configuration for a specific device, a new `udev` rule must be defined. You can define, for example, the user or group that gets access to the device. Also, unique system links can be created to identify and load the devices by their individual names into the software. An example of such a rule can be found in code 5.10. Those rules must be defined in a file with the extension `.rules` in the folder `/etc/udev/rules.d/`. For each device, all rules defined in this directory are applied by `udev` to a device. When defining rules, it is important to note that rules can overwrite each other. For `MAGIX`, we have therefore created a global file that contains the rules for all devices. This file is also installed via the installation script (see Section 5.5.1 and code C.1). It is therefore important that the rules of new devices are added to this global file.

### 5.5.5 Implementation of the Bash/ZSH Completion

For the standard user it is sufficient to control EPICS via a GUI. However, experienced users and developers very often use the command line commands to access PVs. Especially, in driver development when there is no GUI yet, developers depend on it. As the names of PVs can be long and complex strings, entering them is often a pain. EPICS itself does not have a bash completion. In the con-

**Code 5.9:** *Example using the `udevadmin` command to read out the attributes of a connected device*

```
KERNELS=="ttyUSB0"
SUBSYSTEM=="tty"
ATTRS{interface}=="USB-RS485 Cable"
ATTRS{idProduct}=="6001"
ATTRS{idVendor}=="0403"
ATTRS{serial}=="FT03G5K6"
...
```

**Code 5.10:** *Example for a `udev` rule.*

```
SUBSYSTEM=="tty", ATTRS{idVendor}=="0403",
→ ATTRS{idProduct}=="6001", ATTRS{serial}=="FT03G5K6",
→ GROUP="mxuser", SYMLINK+="MPT200"
```

text of this thesis a bash completion was developed for the standard commands like `caget`. Here, in addition to bash, the completion was also developed for the shell ZSH, as most EPICS developers in the MAGIX group prefer it. However, since the principle is identical and only the language differs, we will not make a distinction between the two completions in this section.

The current version of the bash completion needs some preparation by the IOC. The IOC lists all loaded records using the IOC shell command `dbl`. This list can be stored in a file. Each startup script of the IOC uses this to create a file named `#{EPICS_ROOT}/dbl.txt` including the list of all records provided by this IOC. The bash completions uses this file as input to fill the array `COMPREPLY`. The completion system retrieves this array to perform the completions or to output the options. This allows the user to complete the PVs of the locally running IOC. The completion also uses a second file, `#{EPICS_ROOT}/dbl_global.txt`, as input. All PVs defined in this file can also be completed. The content of this global file is up to the user. In most cases it will contain the db list of the other IOCs in the subnet. These can be copied from the hosts using the `scp` command, for example, and added to the `dbl_global.txt` file. The scripts for the Bash and ZSH completion can be found in code C.4 and C.5

A new version for Bash completion is currently under development. Section 5.5.7 introduces the `MXFrontend` module. This allows to query the provided PVs of each IOC over the network. The new version of Bash completion will make use of this instead of querying the PV list from a file as before. How these two parts can work together without causing a delay due to different query times is currently being tested. Development will continue beyond the period of this work.

### 5.5.6 Integration of an Archiver (MAGIX executable)

EPICS does not have its own archiving program, but it does provide a powerful tool, `camonitor`. This tool is connected to a list of PVs and prints the changes of each PV in `stdout` with the timestamp, PV name and current value. If a PV connection can not be opened or is lost, it attempts to reconnect to the PV and prints a brief information. To archive this information, a simple solution is to redirect the output to a file using the command `camonitor $PV_LIST > $LOG_FILE`. This solution is very rudimentary and is not a definitive solution for an archiver. However, it is very useful when you need to log some data quickly.

Since EPICS is an open source, we could create an archiving program for MAGIX based on the `camonitor` program. The changes we made are described below.

First of all, we changed the argument handling. Instead of expecting a list of PVs, the `caArchiver` expects a file with a list of all PVs to be logged. For each PV in the list, a connection is opened and an event callback function is attached. The callback function handles the return for each change to a PV. This function is modified to output the information to a file instead of the `stdout`. For each new day, a new log file is opened with the corresponding timestamp. This prevents the log files from becoming too large. Finally, we have added the ability to configure the log directory via an environment variable called `CA_ARCHIVER_LOG`.

Despite the minor modifications, we managed to turn the `camonitor` tool into a very stable and powerful archiver `caArchiver`. This is started via `systemctl` in a `tmux` session like the IOC application. Only the file `$CA_ARCHIVER_LOG/log.txt` must be created and filed with a list of PVs.

### 5.5.7 Development of the MXFrontend (MAGIX module)

The basic idea of the MXFrontend module is to read internal information of the IOCs or to identify them via the network. For this purpose, the module extends the IOCs with the appropriate functionalities. The communication principle is based on that of Channel Access. MXFrontend opens an additional UDP socket for each IOC, which waits for an external request. If a corresponding client makes a request, the IOC opens a TCP connection to communicate with the client.

Although the module is still in development, it is already loaded and used by all MXSlowControl IOCs. So far, the module supports only a few different requests. One of them is the query of the status of an IOC (on or off). This function can also be used to query which IOCs are currently running via an UDP broadcast throughout the subnet. Another possibility is to poll the PV list of one or more IOCs. Each IOC then sends this list to the client via the network, which can process it. In addition, MXFrontend provides a command that queries all PVs in one or more subnets and stores them in the file `$EPICS_ROOT/db1_global.txt`. All parameters like the UDP or TCP port numbers can be controlled by environment variables. These are read by the system at startup and used for configuration.

The complete communication structure is already programmed and based on this, other queries or commands can be sent to the IOC. Two further functions are currently in preparation. The first one is the possibility to restart an IOC via the network. The second function is to pass the information that the IOC currently outputs as a `stdout` to a client. There are no limits for this module and so one could also think about controlling certain IOCs completely via the module MXFrontend. This means to send the configurations remotely to the IOC and start it. Development of this module will also continue beyond the time period of this work.

### 5.5.8 Expansion by the dbLoadMultiRecords Function

EPICS Base provides two functions for loading and configuring a db file. The basic idea is to create db files with variables as placeholders. These variables must be defined when loading a db file. The first IOC shell command to load and configure a DB file is `dbLoadRecords`. This command takes two arguments, first the db file to be loaded and second a pattern for the variable definitions. An example of `dbLoadRecords` can be found in code 5.1. If a database is to be loaded with different configurations, there are two possible solutions. First, the `dbLoadRecords` command can be executed multiple times to load the same

database file with different configurations. It should be noted that each loaded record is created with a unique name as a new object in the database. If a name is repeated, the second initialization is ignored [B3.14].

Additionally, the function `dbLoadTemplate` can be used. The command only needs a template file as parameter in which all configurations are defined. The configurations are stored as simple lists in the template file. The template function can load multiple db files at once with a large number of different configurations. An example of the `dbLoadTemplate` command can be found in code 5.11 [B3.14].

**Code 5.11:** *Example of the `dbLoadTemplate` command. The file `devADC1248_mxрпи-14.template` is loaded by the IOC shell. This contains several configuration patterns for different db files. One of these definitions is shown for the db file `devADC1248Digital.db`. The digital inputs for the different channels are connected to the GPIO pins of a RPi. The mapping between the channel number and the GPIO pin is defined in the template file. So, each channel is connected to the corresponding GPIO pin after loading and can be read out.*

*IOC shell command*

```
dbLoadTemplate("devADC1248_mxрпи-14.template")
```

*FILE: devADC1248\_mxрпи-14.template*

```
file devADC1248Digital.db {
    pattern{P,Ch,GPIO}
    {mxрпи-14,0,15}
    {mxрпи-14,1,17}
    {mxрпи-14,2,18}
    {mxрпи-14,3,27}
    {mxрпи-14,4,23}
    {mxрпи-14,5,22}
}
```

...

*FILE: devADC1248Digital.db*

```
record( bi, "$(P):ADC1248:Input:Digital:Ch$(Ch)")
{
    field(DTYP, "devgpio")
    field(INP, "@GPIO$(GPIO)_H")
    field(SCAN, ".2_second")
    field(ZNAM, "low")
    field(ONAM, "high")
}
```

...

Both options have a disadvantage. If a large number of records with one or more consecutive numberings must be loaded, you have to write all possible configurations into the IOC start script or template file manually. This may still be possible for smaller sequences, but is a time-consuming and error-prone task for larger sequences. For example, 1024 identical records with sequential numbers had to be loaded for a fiber detector. For this reason, another possibility for loading db files was integrated in the context of this dissertation. This IOC shell command is called `dbLoadMultiRecords`.

Like the `dbLoadRecords` command, the `dbLoadMultiRecords` command takes two arguments: the db file to load and a pattern for the variable definitions. The interpretation of the pattern has been changed to allow sequences. The pattern is interpreted and for each sequence the db file is loaded with the corresponding

pattern. For a sequence of numbers with the notation `parameter1=1..8`, the db file is loaded with the patterns

```
parameter1=1
parameter1=2
...
parameter1=8.
```

To load the db file with a sequence of strings or non-continuous numbers, the notation `parameter2={val1;val2;val3}` is used to generate the patterns:

```
parameter2=val1
parameter2=val2
parameter2=val3.
```

Multiple sequences can be used for one pattern. All possible combinations are used to configure and load the db file. For example, the pattern `P={device1;device2}`, `channel=1..8` is interpreted as:

```
P=device1, channel=1
P=device1, channel=2
...
P=device1, channel=8
P=device2, channel=1
P=device2, channel=2
...
P=device2, channel=8.
```

Two examples are given in code 5.12. The `dbLoadMultiRecords` command is an useful extension to the standard db load commands and is used for many MXSlowControl devices.

**Code 5.12:** *Example of the command `dbLoadMultiRecords`. With the first IOC shell command, `dbLoadMultiRecords`, the file `devNewStepSB.db` is loaded three times with the three different axis numbers 1,3 and 5. With the second IOC shell command the file `devNewStepSB.db` is loaded four times with the different channel numbers 1 to 4.*

#### *IOC shell command*

```
dbLoadMultiRecords("devNewStepSB.db","P=NewStep,□cID=0,□Axis={1;3;5},□PORT=xyzTable")
dbLoadMultiRecords("devHMP4040.db","P=HMP4040,□PORT=HMP4040,□CH=1..4")
```

#### *FILE: devNewStepSB.db*

```
record(longout, "$(P):Axis$(Axis):SetPosition")
{
    field(DESC, "Set□Position□of□Axis$(Axis)□Absolut")
    field(DTYP, "stream")
    field(OUT, "@devNewStepSB.proto_setPos($(cID),$(Axis))□$(PORT)")
    field(LOPR, "-60000")
    field(HOPR, "120000")
    field(EGU, "micro-steps")
}
```

#### *FILE: devHMP4040.db*

```
record(ai, "$(P):$(CH):getVoltage") {
    field(DTYP, "stream")
    field(EGU, "V")
    field(PREC, 3)
    field(INP, "@devHMP4040.proto_getDouble($(P),$(CH),MEAS:VOLT)□$(PORT)")
    field(SCAN, "1□second")
    field(PINI, "YES")
}
```

### 5.5.9 Development for Profibus Support (StreamDevice BUS API)

Some MAGIX devices to control the gas jet target, like the pumping station or the solenoid valve control can only be accessed using Process Field Bus (Profibus). Since there is no EPICS module that allows communication with Profibus, we had to develop a solution ourselves. Instead of writing our own device support, we decided to write a StreamDevice BUS interface. For this purpose StreamDevice provides an Application Programming Interface (API), especially a C++ interface called `StreamBusInterface`. To add a new BUS, it is necessary to create a new class as inheritance of `StreamBusInterface`. Then, the necessary methods have to be implemented, such as connect, write or read. This is the place where the communication via Profibus is specified. Finally, the new class must be registered as a new StreamDevice bus. Then, the new bus can be used in EPICS [SDev2.8]. The advantage of this solution is that the Streamdevice environment can be used. This means that the messages can be defined in the proto file and streamdevice takes care of the communication via Profibus. An example for the usage of the Profibus StreamDevice bus is given in code 5.13.

### 5.5.10 MXSlowControl Tutorial

For unexperienced EPICS developers, a tutorial was created in the internal group's mediawiki as part of this work. This tutorial is constantly updated to support users in getting started with EPICS and MXSlowControl easier and to provide additional professional tips for more complex tasks. The current version covers the topics of process variables, StreamDevice, and PyEpics. Since almost all devices can be controlled via StreamDevice, any user can integrate and use new devices in MXSlowControl after completing this tutorial. Each section of the tutorial covers the most important content and necessary information. Besides examples, tasks are defined. Each solution of the defined tasks leads the user to an important learning step. If a user gets stuck with a task, suggested solutions can be displayed. Other topics that are still under construction are the creation of OPIs with CSS, the creation of device supports, custom modules and more. The goal is to create an all encompassing guide that can be used in the final experiment to allow all experimenters to work with EPICS, quickly fix problems, or add new devices.

### 5.5.11 Development of Device Support for MAGIX Built Devices

MAGIX has also built some self-developed devices for the experiment, which must be controlled via EPICS. For these devices, you have to dive deep into driver development, since most components only understand a very rudimentary language. The electronics required for the application were developed in cooperation with the in-house electronics workshop. To be able to read information from built-in chips, the corresponding registers usually have to be set and read out. In this section, we will look at two of these devices and get a basic overview of how device support is developed for such a device.

#### ADC 1248

The device ADC1248 is a high frequency 24-Bits analog-to-digital converter (ADC) developed from a MAGIX group member [Bir21]. In total eight differential ADC inputs are provided. In addition, it provides six digital entries for signals of 3.3 V and one relay. All the electronics are placed on a board that can directly be

**Code 5.13:** Example for the Profibus StreamDevice bus. The db is loaded with two variables. The variable `PROFIBUS` contains the system link, the destination address, the user data, the source address and the baud rate. The baud rate is configured by an index. In this example the baud rate `baudrates[5]=1.5Mbps` is set. The record `calcout` contains the information to perform the communication. The `DTYP` field specifies that `StreamDevice` is used for communication. The `OUT` field specifies Profibus as `StreamDevice BUS`. The protocol `set` defines the structure of the message. Each valve corresponds to one bit. The message is a 4 byte long word to configure the entire valve setting. For each bit that has the value 1, the corresponding valve is activated. All valves whose corresponding bit has the value 0 are then deactivated.

#### IOC shell command

```
# PROFIBUS=Device:Destination Address:User Data:Source Address:Baudrate
dbLoadRecords("db/devVUVG.db", "P=VUVG, PROFIBUS=/dev/VUVG:3:840000:1:5")
```

#### FILE: devVUVG.db

```
record (bo, "$(P):Valve1")
{
    field(ZNAM, "off")
    field(ONAM, "on")
    field(FLNK, "$(P):AllValves")
}
...

record(calcout, "$(P):AllValves")
{
    field(DESC, "Set_all_Valves_at_once")
    field(DTYP, "stream")
    field(OUT, "@devVUVG.proto_set_Profibus_$(PROFIBUS)")
    field(INPA, "$(P):Valve1")
    field(INPB, "$(P):Valve2")
    field(INPC, "$(P):Valve3")
    field(INPD, "$(P):Valve4")
    field(INPE, "$(P):Valve5")
    field(INPF, "$(P):Valve6")
    field(INPG, "$(P):Valve7")
    field(INPH, "$(P):Valve8")
    field(INPI, "$(P):Valve9")
    field(INPJ, "$(P):Valve10")
    field(CALC, "A*1+B*2+C*4+D*8+E*16+F*32+G*64+H*128+I*256+J*512")
    field(SCAN, "Passive")
}
}
```

#### FILE: devVUVG.proto

```
set {
    out "%#0.4r";
    in "%*1r";
}
```

mounted on a RPi. For the communication the board is connected to the GPIO header of the RPi. To develop the control via EPICS was part of this thesis. The digital entries and the relay can be controlled using the `dev-gpio` module (see Section 5.3.5), so we will focus below on the development of the device support to control the ADC inputs.

Two ADS1248 chips [ADS] are used for the ADC. The registers of the chips are controlled through the Serial Peripheral Interface (SPI). To read the voltage for a channel, the chips must be configured. This is managed by the chip registers. Since



the communication is too complex for StreamDevice, we decided to build a device support for this part. For this purpose, an individual C++ class was developed for each register to manage the corresponding information. An example of such a class can be found in code C.6 and C.7. All register classes are combined into one class representing the ads1248 chip. the ads1248 class also handles the SPI communication for all registers. This allows the chip to be configured as desired and the voltage can be read back from the chip. Up to this point the development is independent of EPICS and the chip can be controlled by different programs using this driver. To read out the voltage with EPICS, a device support has been developed (see Section 5.2.1). An example of a data set for reading out the voltage can be found in code 5.14. The complete device support is packaged in a module that can be installed on an RPi if needed. After installation the ADC1248 board can be controlled.

**Code 5.14:** *Example of a record using the ADC1248 module. The getVoltage function of the ADC1248 module takes as paramter the channel to be read, the gain and the sampling rate. Since different voltage dividers can be placed on the ADC the voltage needs to be calculated using this information as well. The db file is loaded via template since each channel can be configured individually.*

*IOC shell command*

```
dbLoadTemplate("${TOP}/db/devADC1248_RPi0.template")
```

*FILE: devADC1248\_RPi0.template*

```
file ../../db/devADC1248Analog.db {
  pattern{P,Ch,Divider,Gain,SPS,OFFSET}
  {RPi0,0,5,1,20, 0}
  {RPi0,1,5,1,20, 0}
  {RPi0,2,5,1,20, 0}
  {RPi0,3,5,1,20, 0}
  {RPi0,4,5,1,20, 0}
  {RPi0,5,5,1,20, 0}
  {RPi0,6,5,1,20, 0}
  {RPi0,7,5,1,20, 0}
}
```

*FILE: devADC1248Analog.db*

```
record(longin, "${P):ADC1248:Input:Analog:Ch$(Ch):Raw")
{
  field(DESC, "Raw_value_of_the_voltage_for_Channel_$(Ch)")
  field(DTYP, "ADC1248")
  field(SCAN, "1_second")
  field(INP, "getVoltage_$(Ch)_$(Gain)_$(SPS)")
  field(FLNK, "${P):ADC1248:Input:Analog:Ch$(Ch):Voltage")
}

record(calc, "${P):ADC1248:Input:Analog:Ch$(Ch):Voltage")
{
  field{DESC, "Voltage_of_Channel_$(Ch)_with_voltage_divider_1/$(Divider)"}
  field(INPA, "${P):ADC1248:Input:Analog:Ch$(Ch):Raw")
  field(CALC, "${Divider}*2.048/_(_1<23_)*A_-_$(OFFSET)")
  field(SCAN, "Passive")
}
```

## MXSiDet

Another large device carrier module is the MXSiDet module. To control a silicon strip detector (see Section 4.2.3), a front-end board was developed by the



electronics workshop. On the board are several chips, which control e.g. the trigger threshold or the temperature. The whole communication is based on Inter-Integrated Circuit (I<sup>2</sup>C). Several front-end boards (silicon strip detectors) can be controlled by an I<sup>2</sup>C controller via an I<sup>2</sup>C-Multiplexer (I<sup>2</sup>C-MUX). The development of the entire slow control driver for this front-end board was part of this work. As in the case of the ADC1248, the entire communication is too complex for a StreamDevice. Therefore, a separate device support module was developed to control the whole device. A detailed description of this driver can be found in the Section 5.6.3. This is part of the next section which explains the design and development of the entire slow control for the silicon strip detector test bench.

## 5.6 Slow Control Development for the Silicon-Strip Detector Test Stand

As mentioned in Section 4.2.3 currently one silicon strip detector is being tested by the MAGIX group (see Section 4.2.3). The construction of a spatially resolving silicon strip detector system was the topic of a master thesis in the MAGIX group [Gei20]. In this thesis, the initial development of the detectors electronics and the construction of a test stand are described as well as a discussion of the results of the first measurements. In this section, we will focus on the slow control development for this test bench, which is a part of this thesis.

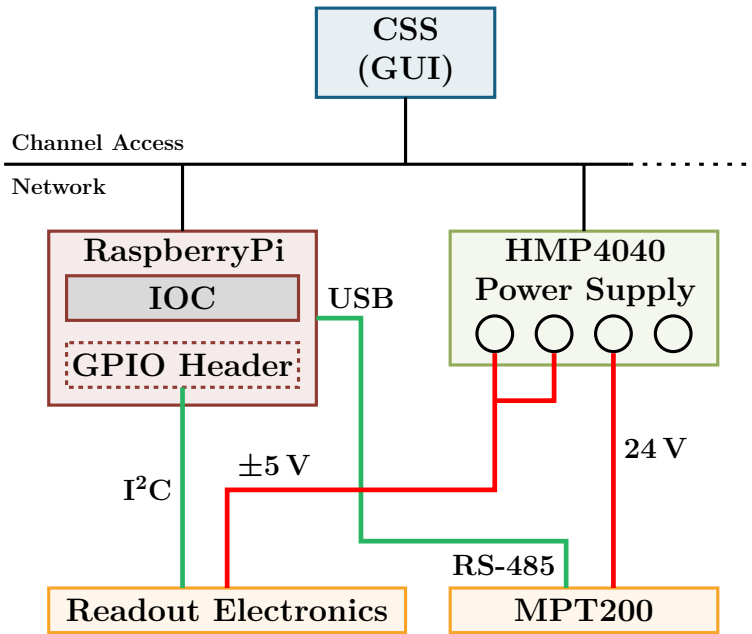
The test stand consists of a vacuum chamber equipped with several supply lines to enable operation of the detector. To achieve a vacuum of the order of  $10^{-5}$  mbar, a rotary vane pump and a turbomolecular pump are connected to the chamber. The detector assembly itself is mounted inside the chamber and consists of the X1 chip directly connected to the manufactured readout electronics and the APV. An MPT200 device from Pfeiffer Vacuum is used to control the chamber pressure. The low voltage supply to the detector and the vacuum transmitter is controlled by the HMP4040 device from Rohde & Schwarz.

This section is used as an example of how to integrate different types of devices into EPICS. First, we will see how to read a MPT200 connected via the serial port. Then, we will discuss how the network device HMP4040 can be controlled via EPICS. Finally, the frontend board of the silicon strip detector (SiDet) will be explained in more detail as well as the development of a corresponding device driver. The general structure from the point of view of the control software is shown in Fig. 5.5.

### 5.6.1 Readout of the MPT200 Device via StreamDevice ttyUSB

The MPT200 device is a pressure sensor that can be controlled via RS-485. To connect the MPT200 to a RPi a RS-485 to USB converter cable is used. For the communication with the RS-485 so called telegram frames are used. They consist only of ASCII code characters [MPT200]. This makes this device a fantastic candidate to be integrated into EPICS using the StreamDevice module.

First, the asynPort driver needs to be configured. Since the device is connected via serial port the IOC shell command `drvAsynSerialPortConfigure` is used for this purpose. The configuration for the serial port can be taken from the device documentation as [MPT200]:

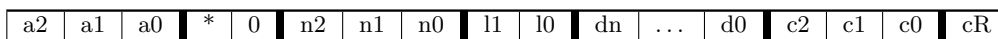


**Figure 5.5:** The test stand of the silicon strip detector is controlled by an IOC running on a RPi. The low voltage power supply HMP4040 is controlled via network. The pressure detector MPT200 for the test chamber is controlled via RS-484 and directly connected to the RPi via USB. The readout electronics is controlled via I<sup>2</sup>C. The support voltage is provided by the HMP4040.

- 9600 baud
- 8 data bits
- 1 stop bit
- no parity.

An example using asyn to specify this configuration is already given in code 5.4 using the udev rule code 5.10.

Telegram frames are used for the communication. The structure of this communication frame is [MPT200]



with

Data Byte	Description
a2 - a0	Address of MPT200
*	Action (0 for data request; 1 for control command or data response)
n2 - n0	Control Command
l1-l0	Data length of dn ... d0 as integer
dn - d0	Data in the corresponding data type
c2 - c0	checksum (sum of ASCII values of a2 to d0) modulo 256
cR	Terminator: Carriage return (ASCII 13).

The address of each MPT200 can be set directly with a switch at the device. The control parameters and the data types are given in the device documentation. This frame must be translated into a StreamDevice protocol. An example for the command read pressure P:740 is already given in code 5.5. The checksum is calculated using the StreamDevice converter %+<sum8>. Hereby, sum8 defines that

the sum of all characters are calculated modulo  $2^8$ . The character + is defined to represent the checksum as decimal ASCII [SDev2.8].

The pressure is returned as `U_Expo_New` type with a size of 6 characters. The `U_Expo_New` integer number `nnnnee` can be converted to a float in units of mbar using the formula [MPT200]

$$p = nnnn * 10^{ee-23} \text{ [mbar]}.$$

To convert the pressure in mbar a `calc` record is used (see code 5.15).

**Code 5.15:** *Example of a `calc` record converting `U_Expo_New` format to mbar*

```
record(calc, "$(P)$ (R)Pressure")
{
  field(DESC, "Calculate Pressure of device MPT200 in mbar")
  field(INPA, "$(P)$ (R)GetPressureRaw")
  field(CALC, "FLOOR(A/100)*10^(A%100-23)")
  field(SCAN, "Passive")
  field(EGU, "mbar")
  field(PREC, 7)
}
```

The further control commands can be translated into a protocol and linked to a record in the same way. The conversions are then also done via `calc` records.

### 5.6.2 Controlling the HMP4040 Device via StreamDevice TCP

The low voltage supply for the entire test stand is provided by the HMP4040 network device. Each of the four output channels can easily be controlled from the front panel. There is also a small display on the front panel that shows some information. Although front panel control may be sufficient for a test stand, the device is connected to the network via the built-in Ethernet port to allow remote control. After configuring the HMP4040 TCP/IP network protocol, the unit can be controlled over the network. All control commands consist of ASCII code characters [HMP]. Thus, the HMP4040 is also an excellent device that can be controlled by EPICS with StreamDevice.

First the `asynPort` driver must be configured. Since the HMP4040 is a network device, the IOC shell command `drvAsynIPPortConfigure` is used for this. To configure `asyn`, the required information can be obtained from the device documentation or set on the front panel of the device [HMP]. With the information

- Network Protocol: TCP
- IP address: 10.32.108.213
- Port Number: 5025 (default)

the `asynPort` can be configured as:

```
drvAsynIPPortConfigure("HMP4040", "10.32.108.213:5025 TCP", 0, 1, 0)
```

The structure for the control commands of the HMP4040 device is simple. First, the output channel must be selected, then the control command can be sent. For example, to set the output voltage for channel 1 to 25 V, the following two commands must be sent to the device [HMP]

```
INST OUT1
VOLT 25.0
```

To read a value, the control command is extended by the character ?. For example, to read the current voltage for channel 2, the following two commands must be sent [HMP]

```
INST OUT2
MEAS:CURR?
```

The response is only the value of the current in ASCII characters, e.g. 1,200, in units of A.

Several commands can be sent to the device at once. For this purpose the line feed character `\n` (ASCII 10) must be used as terminating character between the commands. The structure of the individual commands for this device is identical. They differ only in the corresponding data types that must be transmitted. Since a large number of commands must be integrated into EPICS, it makes sense to create a separate StreamDevice protocol for each of the different data types and to pass the commands as an argument through the records. Apart from reading the identifier as a string, the HMP4040 only uses the double, integer and boolean data types for transmission. This results in only seven different protocols having to be written for the HMP4040: `getID`, `getDouble`, `getInt`, `getBool`, `setDouble`, `setInt`, and `setBool`. An example of the records and protocols for setting the voltage and reading back the current can be found in code 5.16. In addition to the command, the channel number is also an argument to the protocols. Since the channel number is also a parameter of the database, each record has to be defined only once. With the IOC shell command `dbLoadMultiRecords` (see Section 5.5.8) all four channels will be loaded into the IOC.

### 5.6.3 Driver Development to Control the SiDet board via RPi I<sup>2</sup>C

This section explains how the driver for the readout electronics of the silicon strip detector array (see Section 4.2.3) was built. Figure 5.6 shows the readout electronics developed in cooperation with the electronics workshop for the SiDet. Since a space-saving design is required for all parts in the scattering chamber, the electronics are divided into two parts, the frontend-board and the HDMI distribution board. The frontend-board is mounted directly under the detector in the vacuum chamber (see Fig. 4.13). All the electronics to control the detector are located there. An overview of the integrated circuits (ICs) placed on this frontend-board is given in Table 5.1. The HDMI distribution board is located outside of the vacuum chamber. It provides the supply voltage of  $\pm 5$  V and an I<sup>2</sup>C interface to the frontend-board via an HDMI cable. Additionally, an external trigger signal can be input or the generated trigger signal and the analog signals of the frontend-board can be output via LEMO sockets. To control the electronics, the I<sup>2</sup>C master is connected to the HDMI contribution board [Gei20].

A RaspberryPi is used as I<sup>2</sup>C master. This can be connected directly to the HDMI board via the GPIO header. To control multiple boards with one master an I<sup>2</sup>C multiplexer based on the IC PCA9548AD can be used. This allows to control up to 8 silicon strip detectors with one I<sup>2</sup>C master [Gei20].

#### The SiDet library

The first step in creating a device driver is to create a C++ library that enables the communication with the device. The advantage of an EPICS independent library is, that the device can be controled with different programs. This allows flexible use of the device and simplifies the distribution of the driver to the community.

**Code 5.16:** Example of the db and proto file for the HMP4040 device. The db is loaded with `dbLoadMultiRecords` to initialize the four channels separately. Each record is associated with the protocol for the corresponding data type. The channel number and command are passed as arguments to the protocol. Each protocol selects the output channel and sets or requests the values for the passed commands. Since the set values can be read back, each setting protocol requests the current value from the HMP4040 device during initialization using the `@init` macro.

*IOC shell command*

```
dbLoadMultiRecords("devHMP4040.db","P=Silicon:LV,_PORT=HMP4040,_CH=1..4")
```

*FILE: devHMP4040.db*

```
record(ai, "$(P):$(CH):getCurrent") {
    field(DESC, "read_the_current_value_of_channel_$(CH)")
    field(DTYP, "stream")
    field(EGU, "A")
    field(PREC, 3)
    field(INP, "@devHMP4040.proto_getDouble($(P),$(CH),MEAS:CURR)_$(PORT)")
    field(SCAN, "1_second")
    field(PINI, "YES")
}

record(ao, "$(P):$(CH):setVoltage") {
    field(DESC, "set_voltage_value_of_channel_$(CH)")
    field(DTYP, "stream")
    field(EGU, "V")
    field(PREC, 3)
    field(OUT, "@devHMP4040.proto_setDouble($(P),$(CH),VOLT)_$(PORT)")
}
...
```

*FILE: devHMP4040.proto*

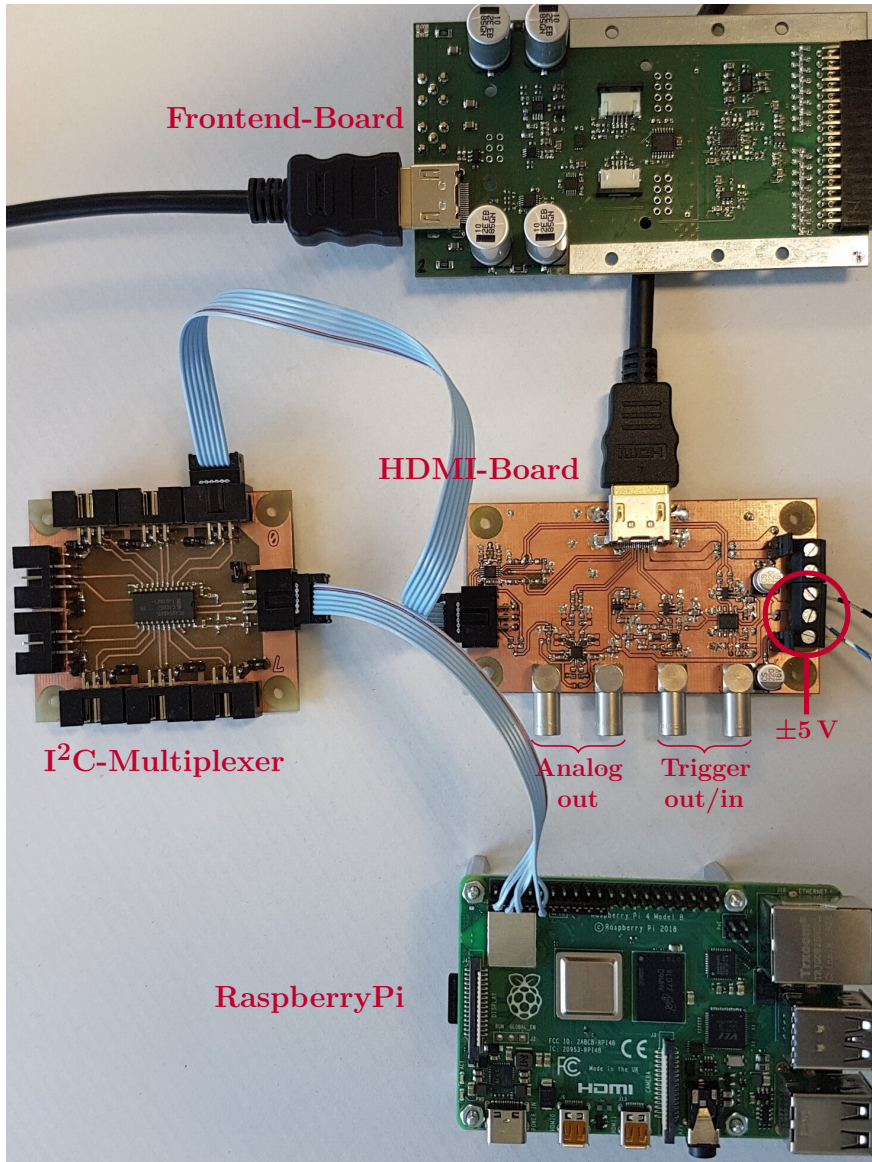
```
getDouble{ # $1=$(P); $2=${Channel}; $3=${Command}
    out "INST_OUT\ $2\n\ $3?\n";
    in "%f";
}

setDouble { # $1=$(P); $2=${Channel}; $3=${Command}
    out "INST_OUT\ $2\n\ $3_%f\n";
    in "%*f";
    @init {
        out "INST_OUT\ $2\n\ $3?\n";
        in "%f";
    }
}
...
```

Since all chips can be accessed with I<sup>2</sup>C one class was built to handle the entire communication via the I<sup>2</sup>C bus. To avoid problems caused by parallel read and write accesses, a singleton class was written for this purpose. The special feature of a singleton is that only one instance (one object) of the class can exist at the same time. The singleton class provides a static method to return the instance of this class. This allows other methods to access this object/class. The class for the I<sup>2</sup>C communication is called `RPiI2C`. It provides methods to open or close the I<sup>2</sup>C bus and to write or read data to/from the device. The source code of the class `RPiI2C` can be found in code C.8 and C.9.

**Table 5.1:** Overview of the integrated circuits (ICs) placed on the frontend-board of the readout electronics for the silicon strip detector [Gei20].

IC	Type	Description
AD5693RBR	16-bit DAC	Set threshold for the trigger signal [AD]
ADS122C04IP	24-bit ADC	Measure the voltage drop from two PT1000 temperature sensors [ADS2]
-	2xPT1000	Platinum Resistance Temperature Detectors; Connected to the cooling plate of the frontend-board and the detector
MCP9808T	Digital Temperature Sensor	Measurement of the local temperature on the frontend-board [MCP]

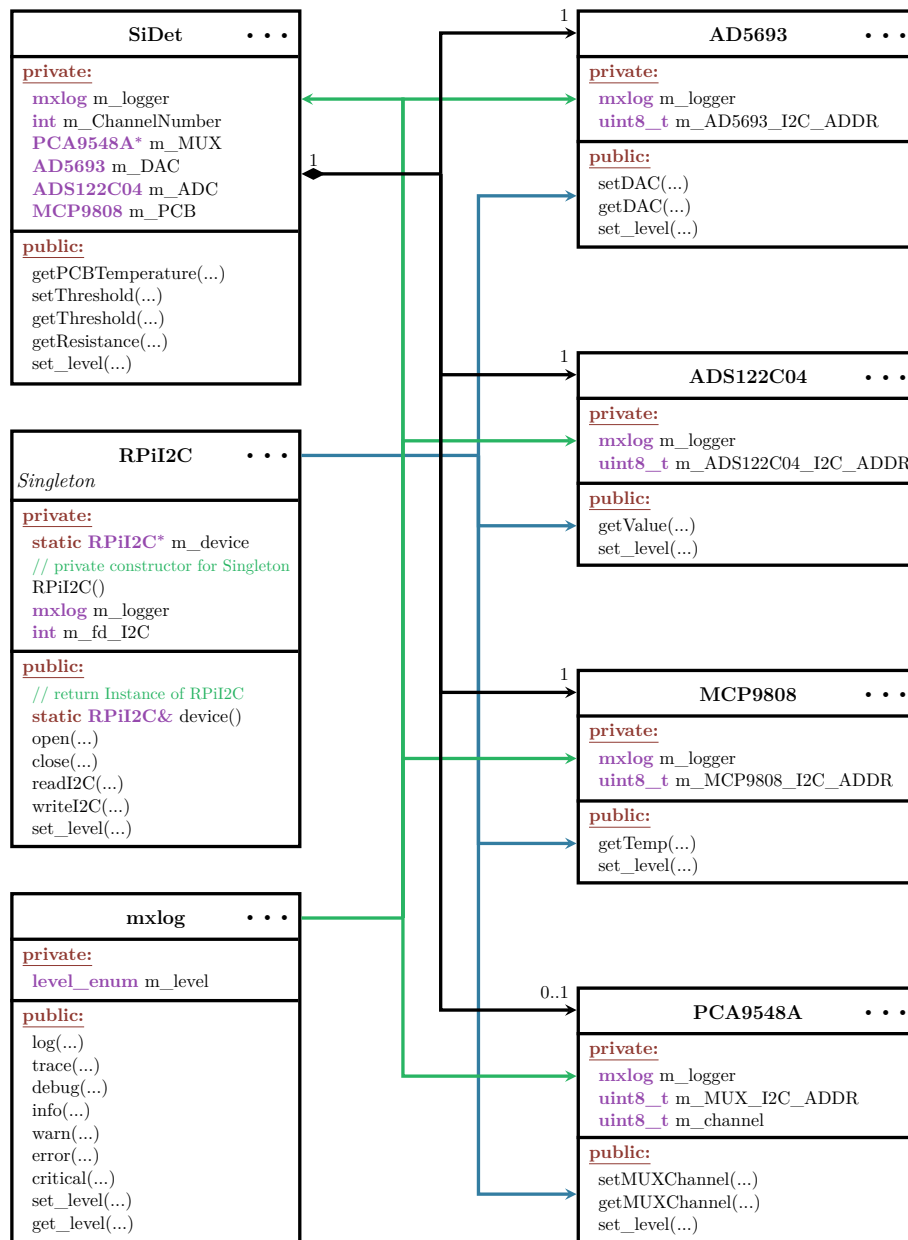
**Figure 5.6:** Overview of the SiDet readout electronics [Gei20]. A detailed description can be found in the text.

Similar to the ADC1248 (compare Section 5.5.11), a separate class has been written for each chip on the frontend-board to provide the complete communication. Each of these classes uses the `RPiI2C` class to write or read data. The structure of the classes is similar and differs only in the way the registers of the chips are set or read and which data can be supplied. To describe the frontend-board, a



class `SiDet` was created, which contains an object of the classes for each chip. The requests to read or write the information are then automatically assigned to the chips respectively the corresponding class objects and processed. In order to include a possibly integrated I<sup>2</sup>C multiplexer, a further class for controlling was added. This can be loaded if necessary also into the `SiDet` class. By creating an instance of the `SiDet` class the complete frontend-board can be controlled. A class diagram of the entire device driver can be found in Fig. 5.7.

In addition to the basic functions for controlling the device, another useful feature has been added, namely a logging system. This was implemented by a class `mxlog` based on the header-only library `spdlog` which was developed to control the entire logging. Each class of the device driver has an individually configured instance



**Figure 5.7:** Class diagram of the device driver for the silicon strip detector array. Each detector can be controlled using the `SiDet` class. For each chip an instance of the corresponding class is loaded. All communication is handled by the singleton class `RPiI2C`. For logging the class `mxlog` is used.

of `mxlog`. For each instance, a log level can be set that defines what information should be output. So, it is possible to output only error messages or more detailed messages up to trace level. Since each object can be configured separately, it is possible to set the logging specifically for different application areas, which e.g. greatly simplifies troubleshooting in case of problems. The log messages are displayed in the terminal with different color codes. Each message contains a time stamp, an entry about the class that issues the message, and the protocol level. By this the log has a search functionality. An example of the output is given in Fig. 5.8. The source code of the class `mxlog` can be found in code C.10 and C.11.

```
[2021-02-19 08:49:21.811] [RPiI2C] [error] Unable to write to slave: 0x40
[2021-02-19 08:49:21.811] [ADS122C04] [error] WriteReg to Device 0x40 failed.
[2021-02-19 08:49:21.811] [SiDet] [error] getResistance: Can't configure ADC!
```

(a) Output with log level set to error for all logger.

```
[2021-02-21 18:36:32.729] [ADS122C04] [debug] Register 0x02 = 0x83
[2021-02-21 18:36:32.730] [ADS122C04] [debug] getValue: rawValue = 0x1f741f
[2021-02-21 18:36:32.731] [ADS122C04] [debug] getValue: value = 0.24573123455047607
[2021-02-21 18:36:32.731] [SiDet] [debug] getResistance: Resistance0 = 958.3518147468567 Ohm
[2021-02-21 18:36:32.731] [SiDet] [debug] getPCBTemperature
[2021-02-21 18:36:32.731] [MCP9808] [debug] getTemp Register Bitset 0b00000101
[2021-02-21 18:36:32.732] [MCP9808] [debug] getTemp rawValue = 0x011b
[2021-02-21 18:36:32.732] [MCP9808] [debug] Temperature = 17.6875°C
[2021-02-21 18:36:32.732] [ADS122C04] [debug] writeRegister WriteReg Bitset 0b01000100
[2021-02-21 18:36:32.733] [ADS122C04] [debug] Register 0x01 = 0x02
```

(b) Output with log level set to debug for all logger.

```
[2021-09-29 10:50:29.112] [RPiI2C] [trace] 0x40 write 1: 0x28
[2021-09-29 10:50:29.113] [RPiI2C] [trace] 0x40 read 1: 0x03
[2021-09-29 10:50:29.213] [RPiI2C] [trace] 0x40 write 1: 0x28
[2021-09-29 10:50:29.214] [RPiI2C] [trace] 0x40 read 1: 0x83
[2021-09-29 10:50:29.214] [RPiI2C] [trace] 0x40 write 1: 0x10
[2021-09-29 10:50:29.215] [RPiI2C] [trace] 0x40 read 3: 0x23 0xc4 0x95
[2021-09-29 10:50:29.216] [RPiI2C] [trace] 0x4c write 1: 0x10
[2021-09-29 10:50:29.217] [RPiI2C] [trace] 0x4c read 2: 0x00 0x00
[2021-09-29 10:50:34.002] [RPiI2C] [trace] 0x40 write 2: 0x44 0x02
[2021-09-29 10:50:34.003] [RPiI2C] [trace] 0x40 write 2: 0x48 0x03
[2021-09-29 10:50:34.003] [RPiI2C] [trace] 0x40 write 2: 0x4c 0xa0
[2021-09-29 10:50:34.004] [RPiI2C] [trace] 0x40 write 2: 0x40 0x00
```

(c) Output with log level set to trace for logger I2C.

**Figure 5.8:** Example output of the `mxlog` class for the silicon strip detector device driver. The first column shows the timestamp. The second column represents the class printing the message and the third one the message level. Finally, the message is attached.

When creating a library, it is a good practice to write unit tests and create a documentation. For each class, a unit test was created based on the header-only library Catch2. Each unit test verifies the functionality and gives an error message if something goes wrong. For the documentation the doxygen system was used. This creates a documentation from the comments, which were entered in the header files of the individual classes in a special syntax. This not only makes it easier for users to access, but also helps developers find their way around the code.

## How to Build EPICS Modules and Link them to the Library

After installing the SiDet library, the module can be created to integrate the driver into EPICS. With the EPICS basic command `makeBaseApp.pl -t support` a directory structure for a new module can be created. Then the device support (see Section 5.2.1) must be programmed. As shown in the example code 5.3 EPICS provides an interface where for an input record only the two functions `init_record` and `read_ai` have to be designed. For an output data set, `write_ai` must be defined instead of the read function. In the initialization function, a new



instance of the `SiDet` class is created, if it does not already exist. The read or write functions have access to the information stored in the `INP` or `OUT` field. This can be used to specify the information that is controlled by the record. In the case of the silicon strip detector, simple commands are defined, such as `getThreshold`. If this command is defined in the `INP` field, the corresponding record will receive the threshold value as a result. Similar commands are defined for the other parameters. The full definition of these functions can be found in code C.12. To connect the `SiDet` module to EPICS, a database definition must be set. In it, the data type must be linked to a record. In addition, the device support structure associated with the data type must be defined. The two entries for the `MXSiDet` module are

```
device(ai,CONSTANT,devSIDET_ai,"SIDET")
device(ao,CONSTANT,devSIDET_ao,"SIDET")
```

Thus, the data sets can access the silicon strip detector data by configuring the `codewordDTYP` as `codewordSIDET`. An example of reading the PCB temperature and setting or reading the threshold is given in code 5.17.

**Code 5.17:** *Example of some of the SiDet records. The DTYP field specifies that the SiDet library should be used. The value in the INP and OUT fields specifies what information is associated with the record. For example, getTemperature specifies that the temperature of the PCB is to be read from the frontend-board and stored in the VAL field of the record.*

```
record(ai, "$(P):getPCBTemperature")
{
    field(DTYP,"SIDET")
    field(SCAN, "10_second")
    field(INP, "getTemperature")
    field(PREC, 3)
    field(EGU, "°C")
}

record(ai, "$(P):getThreshold")
{
    field(DTYP,"SIDET")
    field(SCAN, "5_second")
    field(INP, "getThreshold")
    field(PREC, 3)
    field(EGU, "V")
}

record(ao, "$(P):setThreshold")
{
    field(DTYP,"SIDET")
    field(OUT, "setThreshold")
    field(EGU, "V")
    field(PREC, 3)
    field(DRVL, 0)
    field(DRVH, 2.5)
}
```

In addition, a new IOC shell command called `MXSiDet_SetLogLevel` has been written to set the logging level of the `mxlog`. As the device support, EPICS provides an interface for this purpose as well. After the appropriate C++ structures have been defined, they must be registered in EPICS. This is also done via a `dbd` entry with

```
registrar(SetMXSiDetLogLevelRegistrar)
```

Now the new IOC shell function can be used. To set the log level for all devices, the command

```
MXSiDet_SetLogLevel("log_level")
```

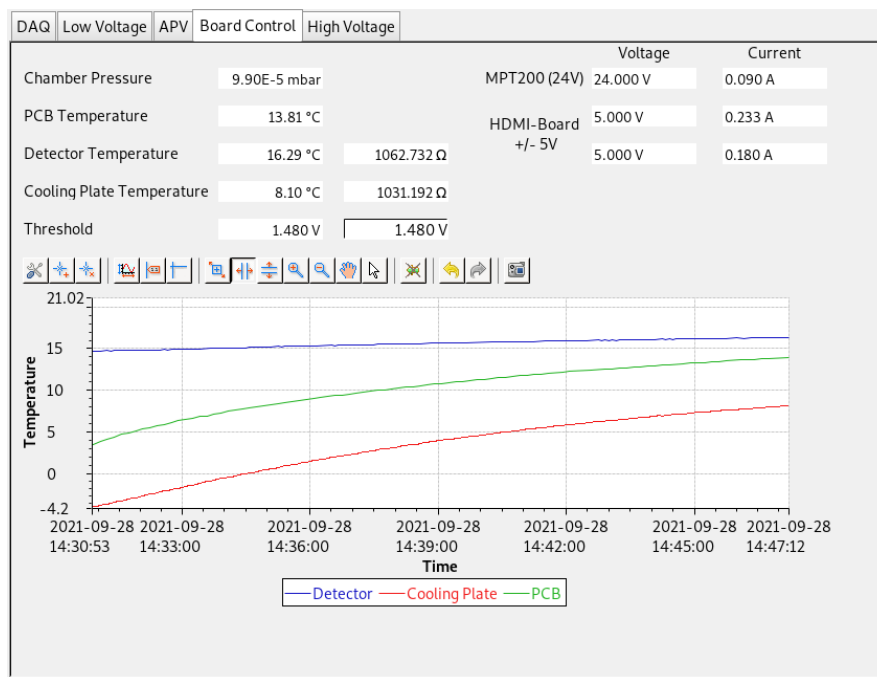
can be used. The strings trace, debug, info, warning, error, critical or off can be used as `log_level`. With the exception of off, each log level specifies the minimum level for the output. For example, if the level is set to info, warnings or errors will also be displayed. To set the logging level for only one class, a second parameter can be set

```
MXSiDet_SetLogLevel("log_level","device_name")
```

The strings SiDet, MUX, DAC, ADC, PCB or I2C can be used as device name. The full definition of this IOC shell command is shown in code C.13.

## Summary and Outlook

After the SiDet library is connected to EPICS via a module, the silicon strip detector test stand can be fully controlled. The structure of the slow control was created in such a way that further devices, which are later required for MAGIX, can easily be integrated. In order to be able to control the test stand in the lab, a corresponding GUI was created in CSS. An example of one OPI can be seen in Fig. 5.10.



**Figure 5.10:** CSS OPI for the silicon strip detector test stand. The Board Control display contains all the information associated with the SiDet frontend-board. The pressure in the chamber, the voltage and the current of the power supply are displayed in text fields. In addition to the PCB temperature, the temperature of the detector and the cooling plate for the frontend-board are displayed. For the two PT1000s, the resistances are also displayed. The threshold value of the DAC can be read or set with the OPI. To have an overview of the trend for all temperatures, a corresponding diagram is displayed on the bottom of the OPI.

---

---

## CHAPTER 6

---

# Simulation of the Electro-Disintegration Reaction $^{16}\text{O}(e, e'\alpha)^{12}\text{C}$

In this chapter, we will discuss the simulation of the cross section of the electro-disintegration reaction  $^{16}\text{O}(e, e'\alpha)^{12}\text{C}$ . The objective of the simulation performed as part of this thesis was to estimate the expected statistical errors of the electro-disintegration reaction of  $^{16}\text{O}$  for the experimental setup in Phase 1 and 2 of the planned measurements at MAGIX. Since no definite information on the efficiency of the detectors and the beam parameters of MESA is available at this time, background simulations could not be included in the simulation. In addition, the acceptances of the detectors were determined purely geometrically because of the efficiencies that still need to be determined.

In the first section, we will get a general overview of physics simulations. For this purpose, we will take a closer look at the Monte Carlo integration, a powerful tool used for simulations. In addition we will discuss the general idea of simulating the cross section of a scattering experiment. Then, in Section 6.2 the simulation written within this thesis for the reaction  $^{16}\text{O}(e, e'\alpha)^{12}\text{C}$  will be explained. All necessary parts of the simulation as well as the analysis will be discussed. Finally, the results will be presented and discussed.

### 6.1 Overview of Physics Simulations

Simulations have become powerful tools that can be used in a variety of ways. To list the multitude of possibilities is beyond the scope of this thesis. In general, one can say that simulations use the computing power of machines or even entire server systems to perform calculations based on models. The results of these simulations can or must then be processed and interpreted by the users. To understand the evolution of galaxies, for example one can simulate different scenarios using suitable models and by varying the input parameters. The results can then be compared with astrophysical observations. Another possible application is the simulation of an experiment. For example, if you want to estimate the expected

count rate in a detector for a scattering measurement, you can simulate this experiment. For this purpose, the cross section must be calculated for a given detector acceptance. To achieve this, you usually have to integrate the differential cross section. The desired count rates can then be determined from the results of the simulation. A very useful tool to perform such an integration within a simulation is the Monte Carlo integration.

### 6.1.1 Principle of Monte Carlo Integration

The Monte Carlo (MC) integration is a useful tool to calculate the value of a multidimensional integral. In the following, we will see that the error of the MC integration scales like  $1/\sqrt{N}$ , where  $N$  is the number of evaluations, regardless of the dimension. Even though the convergence of  $1/\sqrt{N}$  is quite slow, MC integration is one of the preferred methods for evaluating integrals with high dimensions [Wei00].

For a function  $f : \mathbb{R}^n \rightarrow \mathbb{R}$ , with  $n \in \mathbb{N}$ , the mean value in a finite volume  $V \subset \mathbb{R}^n$  is defined as [Mer21]

$$\langle f \rangle := \frac{1}{\text{vol}(V)} \int_V f(x) \, d^n x.$$

The mean value can be approximated using the formula [Mer21; Wei00]

$$\langle f \rangle \approx \frac{1}{N} \sum_{i=1}^N f(x_i),$$

with uniformly distributed  $x_i \in V \forall i$ . So, the integral can be approximated with [Mer21; Wei00]

$$I := \int_V f(x) \, d^n x \approx \frac{V}{N} \sum_{i=1}^N f(x_i),$$

with  $V \equiv \text{vol}(V)$ . This is called the Monte Carlo integral. The law of large numbers ensures that the MC integral converges to the true value of the integral [Wei00]

$$I = \lim_{N \rightarrow \infty} \frac{V}{N} \sum_{i=1}^N f(x_i).$$

The error of the MC integration can be described using the variance  $\sigma^2(f)$  of the function  $f$  [Mer21; Wei00]

$$\begin{aligned} \sigma^2(f) &:= \int_V (f(x) - I)^2 \, d^n x \\ &= \langle f^2 \rangle - \langle f \rangle^2. \end{aligned}$$

Therefore, we get [Mer21; Wei00]:

$$\int_V \left( \frac{V}{N} \sum_{i=1}^N f(x_i) - I \right)^2 \, d^n x = \frac{V^2}{N} \sigma^2(f).$$

This means that the average error of the MC integration is given as

$$\frac{V}{\sqrt{N}} \sigma(f).$$

As result the MC integration can be written as

$$\int_V f(x) \, d^n x = \frac{V}{N} \sum_{i=1}^N f(x_i) \pm \frac{V}{\sqrt{N}} \sqrt{\langle f^2 \rangle - \langle f \rangle^2}. \quad (6.1)$$

This shows that the error of the MC integration scales like  $1/\sqrt{N}$  and is independent of the dimension  $n$ . The exact value of the variance cannot be determined, but can be approximated using [Wei00]

$$\sigma^2(f) \approx \frac{1}{N} \sum_{i=1}^N (f(x_i))^2 - \langle f \rangle^2. \quad (6.2)$$

In practice the points  $x_i$  are calculated using random numbers. Since true random numbers do not exist on a computer the so called pseudo-random numbers are used. Pseudo-random numbers are not generated randomly, but are generated deterministically in the computer by a certain algorithm. Any sequence of pseudo-random numbers should appear randomly to someone who does not know the algorithm. Pseudo-random number generators can establish correlations between sequentially generated numbers. Balling points can be generated in this way. This problem must be considered when selecting a pseudo-random number generator. To avoid errors caused by the generator, it is useful to perform the same MC integration with different pseudo-random number generators [Wei00].

Furthermore, quasi-random numbers can be used. Quasi-random numbers are not random numbers, but are generated by a numerical algorithm and designed to be distributed as uniformly as possible to reduce the errors in the MC integration. Even though quasi-random numbers are designed to sample an  $n$ -dimensional space as uniformly as possible, they are not generally preferable to pseudo-random numbers. Only if the integrand is sufficiently smooth it makes sense to use the quasi-random numbers [Wei00]. For more information on pseudo-random and quasi-random numbers please refer also to [Knu97; Nie92].

Now, we will perform a simple example. Therefore, the area of a circle with radius  $r = 1$  will be calculated using a MC integration. The area is given as

$$A = \pi r^2 = \pi = 3.1415926 \dots$$

Using the function

$$f(x, y) = \begin{cases} 1 & \text{for } x^2 + y^2 \leq 1 \\ 0 & \text{otherwise,} \end{cases}$$

the area can also be calculated as

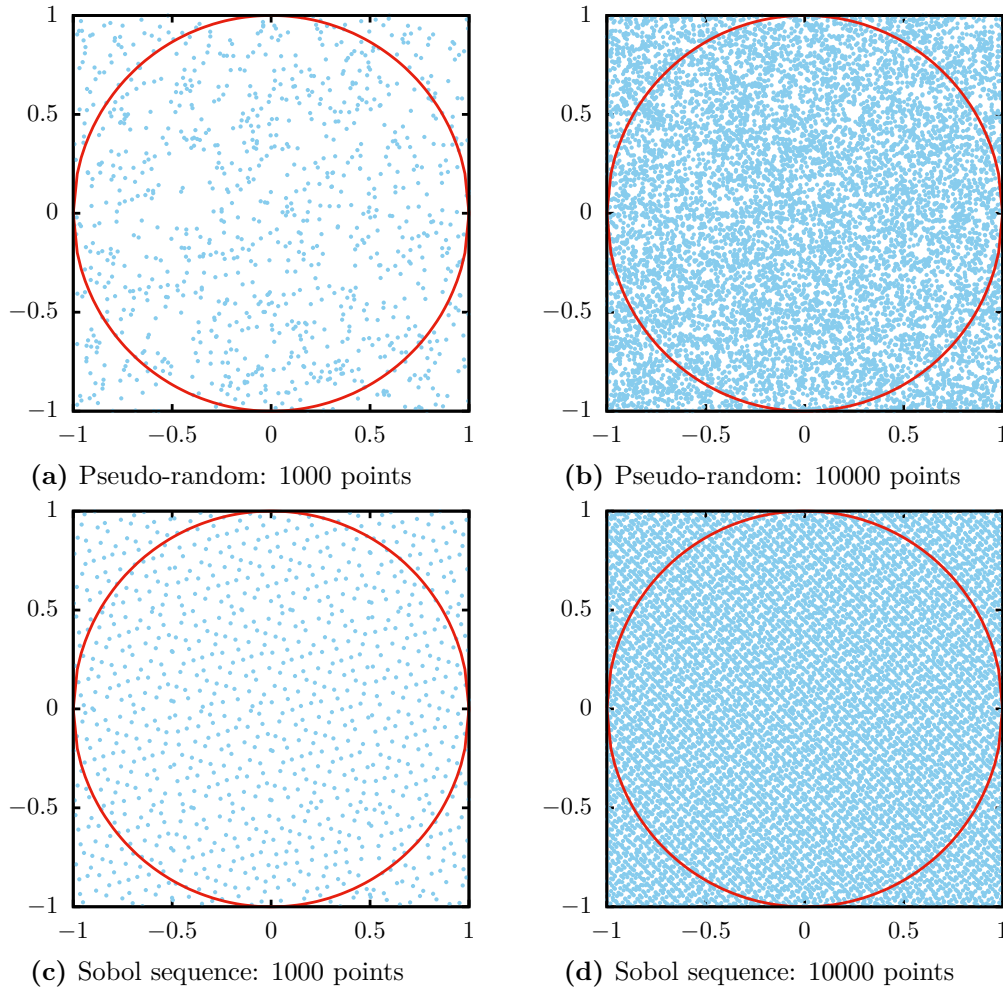
$$A = \int_{-1}^1 \int_{-1}^1 f(x, y) \, dx \, dy.$$

Using a MC integration the area can be approximated as

$$A \approx \frac{4}{N} \sum_{i=1}^N f(x_i, y_i),$$

with the random numbers  $x_i, y_i \in [-1, 1]$  and  $V = 2 \cdot 2 = 4$ . The integration was performed with different numbers of points  $N$ . Two different random number generators were used, one pseudo-random generator and one quasi-random generator using the Sobol sequence. Figure 6.1 illustrates the distribution of the random numbers for both generators. The results of the MC integration are given in Table 6.1. In this table also the average error is given, approximated using Eq. (6.2) to calculate the variance.

As we can see in this example, the error decreases as the number of generated points increases. In general, one way to improve the MC integration result is to



**Figure 6.1:** Illustration of the distribution of the random numbers using a MC integration to calculate  $\pi$ . Two different generators were used, a pseudo-random generator and a quasi-random generator using the Sobol sequence. This example shows how the sobol sequence is used to uniformly sample the space.

increase  $N$  as much as possible. However, since the error converges like  $1/\sqrt{N}$ , one quickly gets into orders of magnitude where computations take much more time without making any significant contribution to the improvement.

On the other hand, the variance can be reduced using different techniques. One of these techniques is the importance sampling. For simplicity, we assume  $n = 1$ . The idea is to reduce the variance by changing the integration variables [Wei00]:

$$\int f(d) dx = \int \frac{f(x)}{p(x)} p(x) dx = \int \frac{f(x)}{p(x)} dP(x),$$

with  $p(x) = \frac{dP(x)}{dx}$ . The mean value of the integral is then calculated as [Wei00]

$$E := \langle f \rangle \approx \frac{1}{N} \sum_{i=0}^N \frac{f(x_i)}{p(x_i)},$$

with  $x_i = P^{-1}(y_i)$  and  $y_i \in V' = P(V)$  the sampling point of the MC integration. Then, the error of the MC integration becomes [Wei00]

$$\frac{V'}{\sqrt{N}} \sigma\left(\frac{f}{p}\right).$$

**Table 6.1:** This table shows the results of the calculation of  $\pi$  using a MC integration. The first column indicates the number of grid points used for the calculation. The error is calculated using Eq. (6.2). The difference  $\Delta_{\text{rel}}$  represents the deviation in percent. Two random number generators were used, a pseudo-random generator and a quasi-random generator based on the Sobol sequence. In the case of the pseudo-random generator, it can be seen that the result for 100 sample points is closer to the true value than for 1000 or 10000 points. Since the range is covered randomly, such effects may well occur. In the case of the Sobol sequence, the results improve continuously as the number of events increases. This is to be expected in this case, since the range is covered as uniformly as possible. In both cases, the results improve as expected for large numbers.

N	Pseudo-Random			Sobol Sequence		
	$\pi_{\text{calc}}$	$\text{err}_{\text{calc}}$	$\Delta_{\text{rel}}[\%]$	$\pi_{\text{calc}}$	$\text{err}_{\text{calc}}$	$\Delta_{\text{rel}}[\%]$
100	3.160 00	0.245 24	0.585 92	3.000 00	0.264 58	4.507 03
1 000	3.192 00	0.076 23	1.604 52	3.112 00	0.079 47	0.941 96
10 000	3.166 00	0.024 45	0.776 91	3.137 20	0.024 82	0.139 82
100 000	3.136 72	0.004 65	0.155 10	3.141 04	0.004 62	0.017 59
1 000 000	3.142 34	0.001 71	0.023 66	3.141 67	0.001 71	0.002 40
10 000 000	3.141 63	0.000 28	0.001 25	3.141 59	0.000 59	0.000 19

The relevant quantity for the error is thus the function  $f(x)/p(x)$ . The variance can be greatly reduced if  $p(x)$  is chosen to be as close as possible in form to  $f(x)$ . The variance can be approximated as [Wei00]

$$\sigma^2\left(\frac{f}{p}\right) \approx \frac{1}{N} \sum_{i=0}^N \left(\frac{f(x_i)}{p(x_i)}\right)^2 - E^2.$$

The function  $p(x)$  must be chosen carefully because it can also increase the variance and thus the error. For example, if the function  $p(x)$  is zero at one point  $\hat{x}$  and  $f(x)$  is not, then the variance becomes large for each  $x_i \approx \hat{x}$ .

As an example of importance sampling, we now consider the integral of the function  $f(x) = \frac{1}{x^2}$  on the interval  $[\varepsilon, 10]$  with a fix  $\varepsilon \in (0, 1]$  small:

$$I = \int_{\varepsilon}^{10} \frac{1}{x^2} dx.$$

The function  $f(x)$  converges to zero for large  $x$ . For  $x \rightarrow 0$  it diverges to  $+\infty$ . The dominating part of this integral is the integral on the region  $[\varepsilon, 1]$ . Analytically the solution to this integral is

$$I = \int_{\varepsilon}^{10} \frac{1}{x^2} dx = \left[-\frac{1}{x}\right]_{\varepsilon}^{10} = \frac{1}{\varepsilon} - \frac{1}{10}.$$

For  $\varepsilon = 0.001$  this becomes  $I = 999.9$ . The variance becomes huge since

$$\langle f^2 \rangle = \frac{1}{V} \int_{\varepsilon}^{10} (f(x))^2 dx = \frac{1}{10 - \varepsilon} \left[-\frac{1}{3x^3}\right]_{\varepsilon}^{10} \approx \frac{1}{30\varepsilon^3}.$$

So, the variance is approximately

$$\sigma^2(f) = \langle f^2 \rangle - \langle f \rangle^2 \approx \frac{1}{30\varepsilon^3} - \frac{1}{100\varepsilon^2}.$$

This leads to a problem using a MC integration to solve this integral. Using a sample  $x_i$  on the interval  $[\varepsilon, 10]$  only one of ten parts will cover the dominating

area below 1 (see Fig. 6.2). In Table 6.2 the results of such an integration can be found. As we can see, the error given is very large, and even with 10 000 000 generated events, the result is incorrect by more than 1 %.

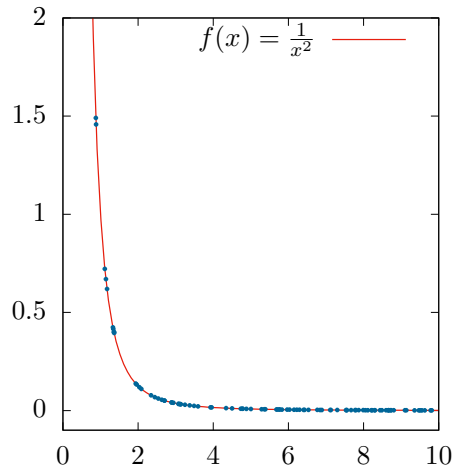
Now, we will use importance sampling to improve the variance and thus the MC integration. For this purpose, we choose  $p(x) = \frac{1}{x^2} = f(x)$ , with  $P(x) = -\frac{1}{x}$ . Then the integral becomes

$$I = \int_{0.1}^{1/\varepsilon} dP = \text{vol}([0.1, 1/\varepsilon]) = \frac{1}{\varepsilon} - \frac{1}{10},$$

and the variance becomes

$$\sigma^2\left(\frac{f}{p}\right) = \sigma^2(1) = 0.$$

This is a perfect case, since only one sampling point is sufficient now to calculate the integral to get the true value. This example is only intended to illustrate the functionality of importance sampling. In practice, the MC integration is used to solve integrals that cannot be solved analytically. Thus the choice  $p(x) = f(x)$  is not possible, because otherwise the integral could be solved without MC. In this case, a function  $p(x)$  should be chosen which describes the form of  $f(x)$  as close as possible.



**Figure 6.2:** The function  $f(x) = \frac{1}{x^2}$  diverges for small  $x$ . A sample in the interval  $[\varepsilon, 10]$  sufficiently covers the range above 1, but not the range below. This leads to a problem when calculating the integral with the MC integration.

### 6.1.2 General Idea of a Cross Section Simulation for a Scattering Experiment

In this section we will take a close look on the general idea how to simulate the cross section for scattering experiments using the MC integration. The goal of such a simulation is for example to estimate the count rate for a specific detector or to determine the expected statistical errors for a dedicated experiment. In all these cases an integration of the differential cross section is needed. However, we are not looking for the total cross section, but for the cross section in small areas, e.g., in a specific window of a detector. The cross section of a specific set  $\Omega_i \subset \Omega$  can be calculated with

$$\sigma_i := \int_{\Omega_i} \frac{d\sigma}{d\Omega} d\Omega.$$



**Table 6.2:** This table shows the results of calculating the integral of the function  $f(x) = \frac{1}{x^2}$  over the interval  $[\varepsilon, 10]$ . The true value is  $I = 999.9$  for  $\varepsilon = 0.001$ . Even for 10 million sample points, the result of the MC integration is deviates by more than 1%. The calculated error for MC is large because the variance is large. Therefore, to better calculate the integral, the variance must be reduced. One method for this is importance sampling.

$N$	$I$	err <sub>calc</sub>	$\Delta_{rel}$ [%]
1000	198.42	25 634.80	80.16
10000	257.07	43 503.59	74.29
100000	826.12	897 671.77	17.38
1000000	851.14	258 797.36	14.88
10000000	982.92	97 977.15	1.70

Since the geometrical description of a detector area can be difficult, this formula is modified in a more general form using the acceptance function  $A(\Omega)$ . This describes the efficiency, 0 for not detected and 1 for full acceptance:

$$\sigma_A := \int_{\Omega} \frac{d\sigma}{d\Omega} A(\Omega) d\Omega. \quad (6.3)$$

This formula allows us to calculate the cross section in any possible detector setup. Only the acceptance needs to be defined properly. To calculate the expected number of events  $N$  for a given detector setup within a time interval  $I$  we can use

$$N_A = \int_I \int_{\Omega} \frac{d\sigma}{d\Omega} A(\Omega) \mathcal{L} d\Omega dt,$$

where  $\mathcal{L}$  is the luminosity. For time independent reactions this becomes

$$N_A = \int_{\Omega} \frac{d\sigma}{d\Omega} A(\Omega) d\Omega \cdot \int_I \mathcal{L} dt = \sigma_A \cdot L.$$

The part  $L := \int_I \mathcal{L} dt$  is called integrated luminosity. For simplicity, we assume that the luminosity is constant, so we get  $L = \mathcal{L} \cdot I$ . To determine the count rate, the integral in Eq. (6.3) is estimated using a MC integration.

As explained in Section 6.1.1 the integral Eq. (6.3) can be approximated as

$$\sigma_A = \int_{\Omega} \frac{d\sigma}{d\Omega} A(\Omega) d\Omega \approx \frac{\text{vol}(\Omega)}{N} \sum_{i=1}^N \frac{d\sigma}{d\Omega}(\omega_i) \cdot A(\omega_i), \quad (6.4)$$

for a set  $\omega_i \in \Omega$  of randomly generated events. To perform such a MC integration different parts need to be defined. The first step is to create a random set of events  $\omega_i \in \Omega$  through an event generator. This can be divided into two parts, the initial state particle generator and the final state particle generator.

### 6.1.3 Initial State Particle Generator

Each reaction can be described using a Feynman diagram. The initial state particle generator will produce all incoming particles of such a Feynman diagram. Random numbers may be included at this point. For example, to randomly generate the starting point, direction or momentum of a particle. These values can be distributed differently, e.g. homogeneous, Gaussian, linear or Poisson. For example, in a scattering process the beam position may be Gaussian distributed.

For a fixed target of length  $d$  and homogeneous density, the target particle can then be generated along the path of the scattering particle in the range  $[-d/2, d/2]$ . How the initial particle generator looks like in detail depends on the experimental setup and the simulation itself.

#### 6.1.4 Final State Particle Generator

After the initial particles are generated, the final state generator generates the parameter of the outgoing particles of the Feynman diagram. They are generated according to the Feynman rules and in agreement with the conservation laws. The reaction plays a decisive role, since it determines possible values for the outgoing particles. The integral or the integration ranges determine which parameters must be generated randomly. The number of different random parameters corresponds to the number of degrees of freedom of the reaction. If other parameters should be generated randomly, a corresponding variable transformation must be applied to the integral. This can be useful for various reasons and must be adapted to the problem. It is important to mention that the random numbers must be independent. Therefore, such transformations must be performed carefully. At this point, one would then also include importance sampling. The weight  $w_i$  of the event is calculated as

$$w_i = |\det(J(\omega_i))| \cdot V,$$

with  $J$  the Jacobian matrix of the variable transformations and the volume element  $V$ . The volume element of the integral results from the parameter ranges which are used to generate the random numbers.

In the example of elastic scattering there are only two degrees of freedom, the scattering angle  $\theta$  as well as the rotation angle  $\phi$ . From there all other parameters of the outgoing particles can be calculated. Instead of generating  $\theta \in [0, \pi]$ , it is more efficient to generate the value  $\cos(\theta) \in [-1, 1]$  through the variable transformation  $\sin(\theta) d\theta = d\cos\theta$  on the integral. Furthermore, we have [Pov+15]

$$d\sigma \propto \frac{1}{\sin^4\left(\frac{\theta_e}{2}\right)}$$

for the cross section of elastic scattering. Thus we know that small scattering angles are preferred. According to importance sampling a transformation

$$p(\sin^2(x)) = \frac{1}{\sin^4(x)}$$

reduces the variance and improves the effectiveness of the MC integration.

To further increase the effectiveness of the MC integration, it is important to limit the range of randomly generated parameters in advance. This is to avoid generating values in ranges that are not relevant to the integration. For example, if it is known that a particular detector can only detect particles in a certain parameter range, the integration volume should be limited to this range to save expensive computation time.

#### 6.1.5 Physics Model Calculations

After the events have been created it is necessary to calculate the physical model. At this point the value  $\frac{d\sigma}{d\Omega}(\omega_i)$  is calculated by using the values generated in the event generator. If the differential cross section is known, it can easily be

calculated. However, in most cases this will not be the case. Therefore, it is necessary to use an adequate model to calculate the differential cross section. At this point, for example, a simplified model can be used. However, it is also possible to approximate the differential cross section from values obtained from fitting existing data. The user is free to develop a suitable model. It is only important that the value of the differential cross section can be calculated for the respective events. This part is probably going to change and evolve over the duration of an experiment in order to improve the MC integration result and thus the desired predictions. With the physics model calculation, the weight of the event is expanded as follows:

$$w_i = |\det(J(\omega_i))| \cdot V \cdot \frac{d\sigma}{d\Omega}(\omega_i).$$

### 6.1.6 Acceptance Simulation

In the final step to calculate the weight, the detectors become a part of the simulation. For each event, it must be calculated whether it can be detected by the detector system. If the event is not detected, the function  $A(\omega_i)$  takes the value 0. If an event is detected with absolute certainty, the value is 1. By characterizing the detectors, the efficiency of the detectors can be determined and the acceptance function can be adjusted accordingly. How the acceptance function is calculated in detail depends strongly on the detector system and its design. Simple models may be designed to geometrically calculate whether the detector area is traversed by the particle, and output the value 1 if it is. However, more complex models are also possible. For example, the track of a particle in a spectrometer could be followed to calculate the focal plane location. Then, the acceptance could be calculated, taking into account the efficiency of the focal plane detector. It is also possible to combine several detector systems into one system using logical AND or OR operations. This would correspond to a detector array or a coincidence for example. No matter how the acceptance is simulated, in the end the function  $A(\omega_i)$  results in a value in the interval  $[0, 1]$  which corresponds to the probability that an event is detected by the system. The acceptance also contribute to the weight of the event:

$$w_i = |\det(J(\omega_i))| \cdot V \cdot \frac{d\sigma}{d\Omega}(\omega_i) \cdot A(\omega_i).$$

### 6.1.7 Calculation of the Cross Section

After all events and weights are calculated they should be stored in a dedicated way. It is advisable to choose a data format that is designed for large amounts of data. For example, one can use the ROOT library developed by CERN to store the data in a tree structure [ROOT]. This can later be read and processed by the user's own analysis or by an analysis based on ROOT.

The total cross section can finally be calculated as

$$\sigma_A = \frac{1}{N} \sum_{i=1}^N w_i.$$

At this point, however, it is usually not the total cross section that is of interest, but rather how the cross section is distributed over small parameter ranges. Histograms can be created out of the events and the weights for example. Each bin of a histogram then corresponds to the integral over a small parameter range

and thus this represents the integrated differential cross section over this range. By multiplying the luminosity to each bin the expected count rates is given. The extent of the analysis depends heavily on what information a user would like to receive from the events and the associated weights.

## 6.2 Implementation of the S-Factor Simulation

In this section we will take a closer look at the simulation of the measurement of the electro-disintegration reaction  $^{16}\text{O}(e, e'\alpha)^{12}\text{C}$  at MAGIX. We will focus on the Phases 1 and 2 of the measurement.

For the simulation a C++ framework developed in the MAGIX group was used. Some parts like the initial state generator and an acceptance simulation of the spectrometer were already included in the framework. All other parts were developed and integrated into the system during this work, especially the final state generator and the model. In addition, I participated in the general development of this software. Besides the development of a physics library based on the one of ROOT, I also worked on the generators and the acceptance classes. Since the description of all this work would go beyond the scope of this thesis, we will limit ourselves here to the development of the simulation for the S-factor and only describe its most important points.

When creating simulations, a consistent coordinate and unit system must be specified. In Table B.9 the coordinate system of the target and the generator system used for the simulation is specified. In addition, the base units defined in Table B.10 are used in the simulation software.

### 6.2.1 Overview of the Initial State Generator

As mentioned before, some initial state generators already existed in the MAGIX framework. For the generation of the beam particle the ideal beam generator was used. As the name implies, the beam particles are ideally generated, i.e. no parameters are randomly assigned. The mass and the charge of the produced particle is loaded from an appropriate particle list using the MC particle numbering scheme, also known as particle identification (PID), [Zyl+20, section 44]. In a next step a starting point is associated to the particle. Since the beam particles are assumed as ideal and thus there is no deviation from the ideal beam, we have  $x = y = 0$ . Then, the beam energy and the direction are defined as a four-momentum. For example an electron (PID 11) with a beam energy of 105 MeV will have the mass  $m_e$ , charge 1, together with the starting point  $\vec{x}_0$  and a four-momentum  $\mathcal{K}^\mu$  as:

$$\vec{x}_0 = \begin{pmatrix} 0 \\ 0 \\ -\infty \end{pmatrix}, \quad \mathcal{K}^\mu = \begin{pmatrix} 105 \\ 0 \\ 0 \\ \sqrt{105^2 - m_e^2} \end{pmatrix}.$$

The  $-\infty$  indicates internally that the particle is generated in the source.

To generate the target particle a homogeneous cube generator was used. A uniform particle distribution in a 3-dimensional box with widths  $dx$   $dy$  and  $dz$  around the origin is assumed. In the case of the simulation for the S-factor, the widths  $dx=dy=0$  and  $dz=2$  are used. In addition, the generator assumes that the target particles are at rest. Similar to the beam generator, the mass and the charge of the particle are loaded using a MC particle numbering scheme. For example

an oxygen target particle (PID 1000080160) will have the mass  $M_{16\text{O}}$  with charge  $Z = 8$ , together with the starting point  $\vec{x}_0$  and a four-momentum  $\mathcal{P}_{16\text{O}}^\mu$  as:

$$\vec{x}_0 = \begin{pmatrix} 0 \\ 0 \\ z \end{pmatrix}, \quad \mathcal{P}_{16\text{O}}^\mu = \begin{pmatrix} M_{16\text{O}} \\ 0 \\ 0 \\ 0 \end{pmatrix},$$

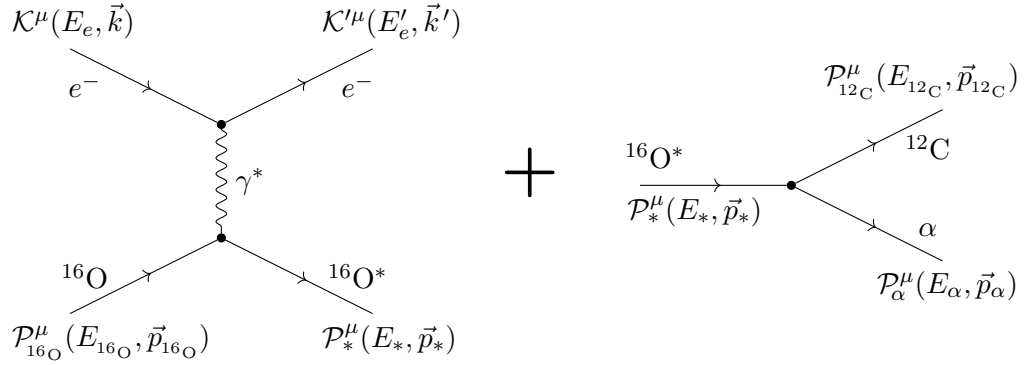
where  $z \in [-1, 1]$  is randomly generated.

### 6.2.2 Development of the Two-Body Final State Generator

The outgoing particles: the scattered electron, the recoiled  $\alpha$ -particle and the  $^{12}\text{C}$  particle have to be generated in the final state generator. The differential cross section of the electro-disintegration reaction  $^{16}\text{O}(e, e'\alpha)^{12}\text{C}$  has five degrees of freedom Eq. (3.39):

$$\frac{d\sigma}{d\omega d\Omega'_e d\Omega_\alpha^{\text{cms}}},$$

where  $\omega = E_e - E'_e$ ,  $d\Omega'_e = d\cos(\theta_e) d\phi_e$  and  $d\Omega_\alpha^{\text{cms}} = d\cos(\theta_\alpha^{\text{cms}}) d\phi_\alpha^{\text{cms}}$ . It is easy to see that  $d\omega = dE'_e$ . For the simulation we split the Feynman diagram Fig. 3.1 and thus the calculations into two parts, see Fig. 6.3.



**Figure 6.3:** The Feynman diagram of the electro-disintegration reaction of  $^{16}\text{O}$  (Fig. 3.1) can be separated into two parts. The left diagram represents the formation of the compound nucleus state  $^{16}\text{O}^*$ . The right one represents the decay into  $\alpha$  and  $^{12}\text{C}$ .

Instead of the two parameters  $E'_e$  and  $\cos(\theta_e)$ , the invariant parameters  $W$  and  $q^2$  are used for the MC integration. Thus a variable transformation  $\Phi$

$$\Phi : \begin{pmatrix} E'_e \\ \cos(\theta_e) \end{pmatrix} \mapsto \begin{pmatrix} W \\ q^2 \end{pmatrix} (E'_e, \cos(\theta_e)),$$

is necessary. In Section 3.8 we find

$$\begin{aligned} W &= \sqrt{2\omega M_{16\text{O}} + M_{16\text{O}}^2 + q^2} \\ &= \sqrt{2(E_e - E'_e)M_{16\text{O}} + M_{16\text{O}}^2 + q^2}, \end{aligned} \quad (6.5)$$

where the four-momentum transfer is given as Eq. (3.4)

$$\begin{aligned} q^2 &:= 2m_e^2 - 2E_e E'_e + 2kk' \cos \theta_e \\ &= 2m_e^2 - 2E_e E'_e + 2\sqrt{E_e^2 - m_e^2} \sqrt{E_e'^2 - m_e^2} \cos \theta_e. \end{aligned} \quad (6.6)$$

To determine the Jacobian matrix  $J_\Phi$  the partial derivatives of the transformation have to be calculated

$$\begin{aligned} \frac{\partial W}{\partial E'_e} &= -\frac{M_{^{16}\text{O}}}{W} & \frac{\partial W}{\partial \cos(\theta_e)} &= 0 \\ \frac{\partial q^2}{\partial E'_e} &= -2E_e + 2E'_e \cdot \frac{k}{k'} \cos(\theta_e) & \frac{\partial q^2}{\partial \cos(\theta_e)} &= 2kk'. \end{aligned}$$

This results in

$$\det(J_\Phi) = -\frac{2M_{^{16}\text{O}}kk'}{W}.$$

The weights for the events are then defined as

$$\begin{aligned} w &= |\det(J_\Phi^{-1})| \cdot \text{vol}(V) = \frac{1}{|\det(J_\Phi)|} \cdot \text{vol}(V) \\ &= \frac{W}{2M_{^{16}\text{O}}kk'} \cdot \text{vol}(V), \end{aligned}$$

with the integration volume

$$\text{vol}(V) = \int dW dq^2 d\phi_e d\Omega_\alpha^{\text{cms}}.$$

Since the differential cross section is proportional to the Mott cross section we have

$$\frac{d\sigma}{d\omega d\Omega'_e d\Omega_\alpha^{\text{cms}}} \propto \left( \frac{d\sigma}{d\Omega'_e} \right)_{\text{Mott}} \propto \frac{1}{q^4}.$$

So, events with low  $q^2$  are preferred. This requires a further variable transformation according to importance sampling. Therefore we choose

$$p(q^2) := \frac{1}{(q^2)^2}.$$

Then we get  $dq^2 = q^4 dP$  with  $P \in \{1/q^2\}$ . Finally, the weight is given as

$$w = q^4 \cdot \frac{W}{2M_{^{16}\text{O}}kk'} \cdot \text{vol}(V').$$

In the case of an connected volume  $V'$  the integration volume becomes

$$\begin{aligned} \text{vol}(V') &= \int dW dP d\phi_e d\Omega_\alpha^{\text{cms}} \\ &= \int_{W_{\min}}^{W_{\max}} dW \int_{P_{\min}}^{P_{\max}} dP \int_{\phi_{e,\min}}^{\phi_{e,\max}} d\phi_e \int_{\cos(\theta_{\alpha,\max}^{\text{cms}})}^{\cos(\theta_{\alpha,\min}^{\text{cms}})} d\cos(\theta_\alpha) \int_{\phi_{\alpha,\min}^{\text{cms}}}^{\phi_{\alpha,\max}^{\text{cms}}} d\phi_\alpha^{\text{cms}}. \end{aligned}$$

With

$$\nu := \int_{P_{\min}}^{P_{\max}} dP = \int_{q_{\min}^2}^{q_{\max}^2} p(q^2) dq^2 = \frac{1}{q_{\min}^2} - \frac{1}{q_{\max}^2},$$

we get

$$\text{vol}(V') = \nu \cdot (W_{\max} - W_{\min}) \cdot (\phi_{e,\max} - \phi_{e,\min}) \cdot (\cos(\theta_{\alpha,\min}^{\text{cms}}) - \cos(\theta_{\alpha,\max}^{\text{cms}})) \cdot (\phi_{\alpha,\max}^{\text{cms}} - \phi_{\alpha,\min}^{\text{cms}}).$$

For each event the parameters are generated as a random value. For example, if  $u_W$  is a random number in the range  $[0, 1]$  the invariant mass of an event is calculated as

$$W = W_{\min} + u_W \cdot (W_{\max} - W_{\min}).$$

In the case of four-momentum transfer the value is calculated as

$$q^2 = \frac{1}{\frac{1}{q_{\min}^2} - u_{q^2} \cdot \nu},$$

with a random number  $u_{q^2} \in [0, 1]$ .

With the set  $(W, q^2, \phi_e, \cos(\theta_\alpha^{\text{cms}}), \phi_\alpha^{\text{cms}})$  each four-momenta of the outgoing particles can be determined. In addition, a starting point is defined for each particle. The charge and the mass of each particle is loaded from a particle list using the PID. All this information is needed in the further steps of the simulation.

The starting point for all outgoing particles will be defined as the interaction point from the beam particle with the target particle. In this particular simulation this is equal to the starting point of the target particle  $x_i = x_0$ .

The four-momentum of the electron  $\mathcal{K}'^\mu$  is defined by  $(E', \cos(\theta_e), \phi_e)$ . The values  $E'$  and  $\cos(\theta_e)$  are calculated by using the formulas Eqs. (6.5) and (6.6).

In the cms the intermediate state  $\mathcal{P}_*^\mu(E_*, \vec{p}_*)$  is given as

$$\mathcal{P}_*^\mu = \begin{pmatrix} W \\ \vec{0} \end{pmatrix}.$$

To calculate the four-momentum of  $\alpha$  and  $^{12}\text{C}$  we can use the formulas of the two-body decay from [Zyl+20, chapter 48]:

$$\begin{aligned} E_\alpha^{\text{cms}} &= \frac{W^2 - M_{^{12}\text{C}}^2 + M_\alpha^2}{2W} \\ E_{^{12}\text{C}}^{\text{cms}} &= W - E_\alpha^{\text{cms}} \\ p_\alpha^{\text{cms}} = p_{^{12}\text{C}}^{\text{cms}} &= \frac{\sqrt{(W^2 - (M_{^{12}\text{C}} + M_\alpha)^2)(W^2 - (M_{^{12}\text{C}} - M_\alpha)^2)}}{2W}. \end{aligned}$$

In the cms we have  $\vec{p}_\alpha = -\vec{p}_{^{12}\text{C}}$  with a homogeneous angular distribution. In this particular simulation the range used for the cms particle generation is set to  $\cos(\theta_\alpha^{\text{cms}}) \in [-1, 1]$  and  $\phi_\alpha^{\text{cms}} \in (-\pi, \pi]$ . In the final step the four-momentum of the  $\alpha$ -particle and the  $^{12}\text{C}$  is boosted into the lab frame.

At the end of the final state generator the four-momenta  $\mathcal{K}'^\mu$ ,  $\mathcal{P}_\alpha^\mu$  and  $\mathcal{P}_{^{12}\text{C}}^\mu$  in the lab frame, the interaction point  $\vec{x}_i$  and the weight  $w$  is calculated for each event. This information is passed to the model calculation in combination with the PID of each particle.

### 6.2.3 Development of the S-Factor Model Calculation

The cross section Eq. (3.39) must be determined for each event in the model calculator. Some assumptions and simplifications are made to calculate the proportion

$$\sum_K \tilde{v}_K R_K^{\text{cms}}.$$

Since the four-momentum transfer  $Q^2$  will be small ( $\mathcal{O}(10^{-4} \text{ GeV}^2)$  in Phase 1;  $\mathcal{O}(10^{-9} \text{ GeV}^2)$  in Phase 2), the virtual photon is approximated as quasi-real. Therefore, the longitudinal and both interference parts are neglected in the current version of the simulation:

$$\tilde{v}_L = \tilde{v}_L = \tilde{v}_L = 0.$$

So, only the transverse part is calculated. The same assumption is made to determine the nuclear response function  $R_T^{\text{cms}}$ , so that Eqs. (3.54) to (3.56) are used to calculate the value of  $R_T^{\text{cms}}$ . At this point, the value of  $S_{E1}(E_\alpha^{\text{cms}})$  and  $S_{E2}(E_\alpha^{\text{cms}})$  are also taken into consideration. For this purpose, the AZURE2 software [Ud15] was used to extrapolate the S-factor of the ground state transition for E1 and E2 in the cms energy range  $W = 0.01 - 20$  MeV. Therefore, the AZURE2 software was configured with the parameters of the supplementary material provided by [deB+17]. The output of AZURE2 is stored in a file containing the cms energy and the value of  $S_{E1}$  or  $S_{E2}$ . In the extrapolation, different step widths in different energy ranges were chosen (see Table B.11). The produced files of the AZURE2 software are loaded into the model calculator. To calculate the S-factor value for a given  $E_\alpha^{\text{cms}}$ , the calculator searches for the two entries which cms energy is closest. The value of the S-factor is calculated by a linear approximation based on the two entries. Since the loaded list is sorted, the value can be determined quickly.

Finally, the weight is calculated as

$$w = q^4 \cdot \frac{W}{2M_{^{16}\text{O}}kk'} \cdot \text{vol}(V') \cdot \frac{d\sigma}{d\omega d\Omega'_e d\Omega_\alpha^{\text{cms}}},$$

with

$$\frac{d\sigma}{d\omega d\Omega'_e d\Omega_\alpha^{\text{cms}}} = \frac{M_\alpha M_{^{12}\text{C}}}{8\pi^3 W} \cdot \frac{\beta_e'^2}{\beta_e} \cdot p_\alpha^{\text{cms}} \sigma_{\text{Mott}} \cdot \tilde{v}_T R_T^{\text{cms}},$$

and

$$R_T^{\text{cms}} = \frac{\hbar c \cdot p_\alpha^{\text{cms}} \cdot W \cdot e^{-2\pi\eta}}{2\alpha \cdot E_\gamma^{\text{cms}} \cdot M_\alpha M_{^{12}\text{C}} \cdot E_\alpha^{\text{cms}}} \cdot \left\{ S_{E1} [P_0(\cos(\theta_\alpha^{\text{cms}})) - P_2(\cos(\theta_\alpha^{\text{cms}}))] \right. \\ \left. + S_{E2} \left[ P_0(\cos(\theta_\alpha^{\text{cms}})) + \frac{5}{7} P_2(\cos(\theta_\alpha^{\text{cms}})) - \frac{12}{7} P_4(\cos(\theta_\alpha^{\text{cms}})) \right] \right. \\ \left. + \frac{6}{\sqrt{5}} \cos(\phi_{12}(E)) \sqrt{S_{E1} S_{E2}} [P_1(\cos(\theta_\alpha^{\text{cms}})) - P_3(\cos(\theta_\alpha^{\text{cms}}))] \right\}.$$

The phase shift was approximated equal to Eq. (2.30) as

$$\cos(\phi_{12}(E)) = \frac{2}{\sqrt{\eta^2 + 4}}.$$

#### 6.2.4 Implementation of Acceptance Simulation

After generating an event and calculating the weight, the value of the acceptance function  $A$  must be determined. Therefore, the detector setup must be simulated. Each detector is implemented as an ideal version in the MAGIX framework as a separate C++ class. The acceptance function of each detector returns the value 1 if the particle is detected. Otherwise it returns the value 0.

In this section we will discuss how the different detector systems are integrated into the simulation. Each detector is individually designed to suit the specific requirements. After a review of the individual detectors for the electrons and the  $\alpha$ -particles, the linking of several detectors as a set or as coincidence is discussed.



### Spectrometer

The acceptance for the spectrometers was already implemented to the framework. The spectrometer acceptance is a 4-dimensional space with the parameters  $k, \theta, \phi, y$  in the spectrometer coordinate system. The spectrometer acceptance was implemented as a binary tree as discussed in [Mül14]. Each level of the tree represents a bisection of a coordinate and one moves to the next level after iterating over all coordinates alternately. Each evaluation represents a bisection of the available phase space in that coordinate, so that after iterating over all coordinates, the search restricts the phase space of each node to a hypercube of dimension  $n$  and volume  $\frac{1}{n^2}$  compared to the previous iteration. Any leaf that is completely inside or outside the acceptance space is not subdivided further. This results in different volumes for each leaf in the different iteration steps. The result of the iterations indicates whether the particle hits the focal plane. In this case the acceptance returns  $A = 1$ , otherwise  $A = 0$ .

### Zero Degree Tagger

The zero degree tagger is implemented as a purely geometric detector in the simulation software. Each particle generated in the final state generator is passed to the acceptance class of the zero degree tagger to test whether any of these particles will be detected. For each particle, the acceptance is calculated separately. First, the detector verifies the charge of the particle. This must be negative. The next step is to confirm that the particle's momentum is within the momentum acceptance range. It is assumed that any electron with a scattering angle below  $\theta_{\max}$  is focused by the first deflecting magnet onto the active area of the detector system. Thus, the last step is to prove that the scattering angle is less than  $\theta_{\max}$ . If one particle passes all three tests, the zero degree detector returns  $A = 1$ , otherwise  $A = 0$ . The accepted momentum range as well as the maximum angle acceptance can be configured in the initialization phase of the simulation.

### Silicon Strip Detector

A silicon strip detector array is used for the detection of the  $\alpha$ -particles. This array consists of a number of individual configured silicon strip detectors. Similar to the acceptance simulation for the zero degree tagger, the simulation of such a silicon strip detector is purely geometric. Each silicon strip detector is configured with a set of parameters that can be specified in a configuration file that is loaded into the simulation during the initialization phase (see code D.3 and D.4). There, the minimum accepted particle energy, the charge sensitivity, the rotation angle, the distance to the target center  $d_S$ , and the length  $l_S$  as well as the width  $w_S$  of the detector can be specified. In addition, each detector can be restricted to a list of particles. For this purpose, a list of PIDs is used to configure the detector.

For each particle calculated in the final state generator, the acceptance class of the silicon strip detector is used to verify whether it will be detected. If a PID list is set, it is first checked whether the particle's PID is listed there. If not, it will not be accepted by the detector. In the next step, the energy and the charge are tested. In other words, it is checked if the energy of the particle is above the minimum accepted energy value and if the charge of the particle is positive. In the last step, it is proved whether the particle hits the detector area. For this purpose, a straight line is formed with the interaction point as the starting point and a direction parallel to the three-momentum of the particle. The entire system is then transformed so that the detector is located in the  $x$ - $y$  plane with  $z = d_S$ .

Then, the  $x$  and  $y$  positions of the line are determined for  $z = d_s$ . In the case  $|x| < w_s/2$  and  $|y| < l_s/2$  the particle hits the detector. If one of the particles hits the detector, the acceptance class makes  $A = 1$ , otherwise  $A = 0$ .

### Acceptance Set and Coincidence

To simulate a set of detectors or the coincidence of two detector systems, additional acceptance functions are implemented into the simulation framework. The acceptance for a set of detectors is calculated as follows

$$A = 1 - \prod_i (1 - A_i).$$

Where  $A_i$  are the individual acceptances of the detectors. In the current version of the simulation, this gives  $A = 1$  when even one detector can accept one of the particles. This was used for the simulation of the silicon strip detector array.

The acceptance for a coincidence of several detectors, on the other hand, is calculated as

$$A = \prod_i A_i,$$

where  $A_i$  are the individual acceptances of the detectors involved. In the current version of the simulation, this gives  $A = 0$  when even one detector cannot accept any of the particles. This was used to simulate coincidence between the spectrometer (or zero degree tagger) and the silicon strip detector array.

### 6.2.5 Analysis: Calculation of Expected Count Rate

After an event is generated, the weight for this event is estimated, and the acceptance for the event is determined, the event is passed to the analysis. This processes the data depending on the information to be obtained from the simulated data.

In general, the result of the MC integration gives the total integrated cross section for a given acceptance  $A$ .

$$\sigma_A = \frac{1}{N} \sum_{i=1}^N w_i,$$

where  $w_i$  denotes the weight for each event, and  $N$  the number of total events. This is only one single value and offers rather less information. The basic idea is to split the integration space into small volumes where the differential cross section can be assumed to be constant. In the one-dimensional case such an integration can be written for example as

$$\sigma(E) = \int_E^{E+\Delta E} \frac{d\sigma}{dE}(E) dE \approx \frac{d\sigma}{dE}(E) \cdot \Delta E.$$

If a parameter is entered into a histogram together with the associated weight for each event, a weighted histogram is obtained. After all events have been written into the histogram, it must be scaled by the factor  $1/N$ . In this particular case, each bin represents the integrated differential cross section, integrated over the parameter range on which that bin is based. An example can be found in Fig. 6.4. There, the cms energy of the  $\alpha - ^{12}\text{C}$  system is entered in a one-dimensional histogram with the associated weight for each event. The left  $y$ -axis shows the value of each weighted bin and represents the cross section on a small energy width  $\Delta E = 0.069 \text{ MeV}$ . The right  $y$ -axis of this histogram represents the number of

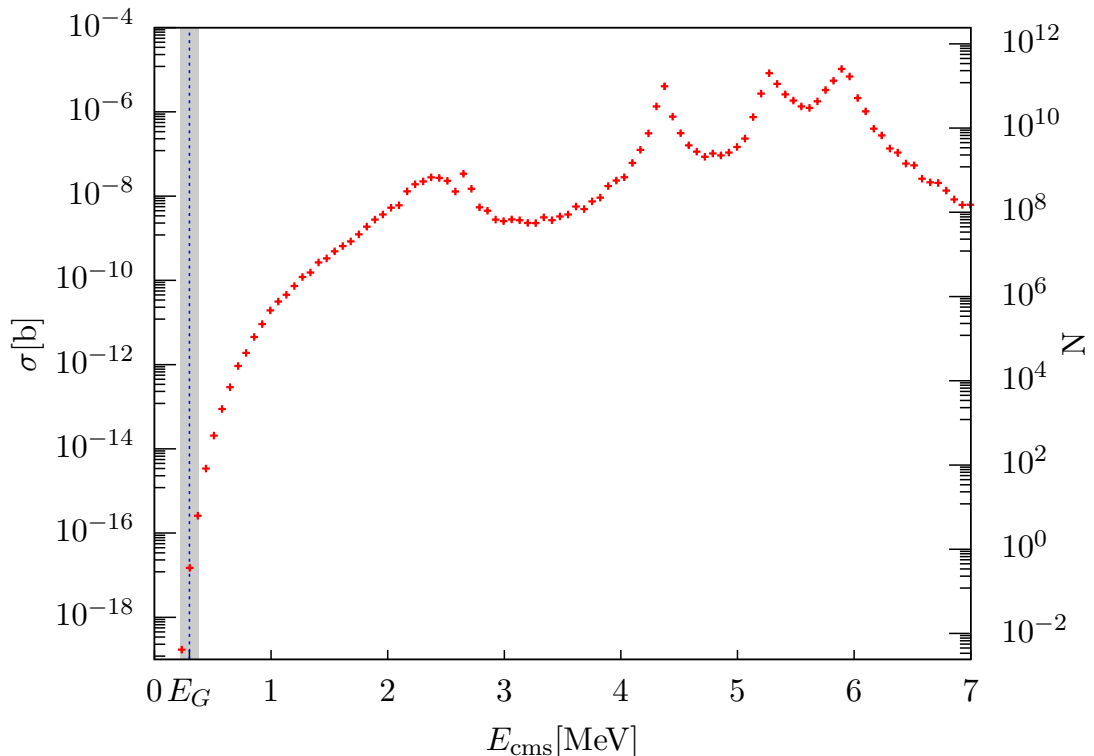
expected reactions per energy width  $\Delta E$ . The number of reactions  $N_r$  is calculated using

$$N_r = \sigma \cdot L,$$

with the integrated luminosity

$$L = \mathcal{L} \cdot T,$$

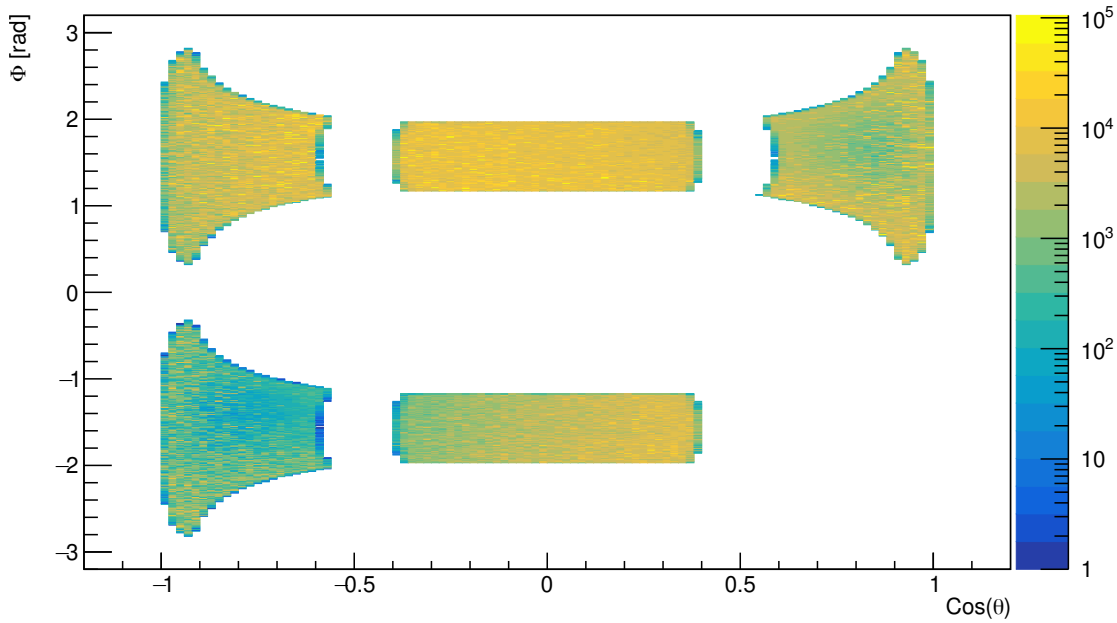
where a constant luminosity  $\mathcal{L} = 10^{34} \text{ cm}^{-2} \text{ s}^{-1}$  integrated over the time duration  $T = 4 \text{ weeks} = 2\,419\,200 \text{ s}$  was assumed.



**Figure 6.4:** Histogram representing the weighted cms energy  $E^{cms}$  for Phase 2. Each bin represents the integration over a small energy width of  $\Delta E = 0.069 \text{ MeV}$ . The left y-axis represents the cross section while the right y-axis represents the expected number of reactions.

Also multi-dimensional histograms can be produced. Figure 6.5 shows a two-dimensional histogram representing the hitmap for the silicon strip detector array of Phase 1. The histogram is scaled by the same integrated luminosity and represents the expected number of detected events in coincidence with the spectrometer.

Some of the histograms are produced through the simulation and are stored in a ROOT file so that they are available after the simulation [ROOT]. All histograms can be configured in the initialization phase of the simulation, for example to set the parameter range or the number of bins (see code D.2). The analysis is started directly with the simulation and processes the events as soon as they were generated. This has the advantage that the results are available directly after the simulation in the form of histograms and can be used for further processing. However, one disadvantage is that the raw data of the events and also the weights are not available after the simulation and cannot be regenerated from the histograms. This means if changes are made to the analysis, the entire simulation has to re-run to store the desired information. Depending on the simulation and the number of events generated, this can take a long time.



**Figure 6.5:** Hitmap for the silicon strip detector array. It shows the expected number of events detected on the surface of the different silicon strip detectors for one possible configuration in Phase 1.

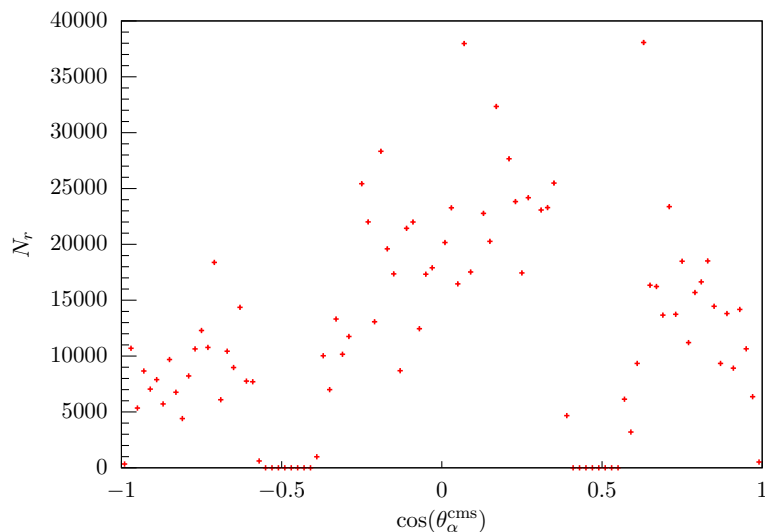
Therefore, all data generated in the simulation was additionally stored in a file. Thus it is available for later analyses. For this purpose, all data that can be obtained from the events are stored in a tree structure. Therefore, TTree from the ROOT package was used [ROOT]. A separate branch is created for each particle so that the information can be stored and retrieved individually. The structure for all particle branches is identical and contains the vertex, the mass, the four-momentum and the PID of the particle. In this particular case, the vertex corresponds to the interaction point and is therefore identical for all particles. In addition to the initial and final state particles, the intermediate state particles,  $\gamma^*$  and  $^{16}\text{O}^*$ , were also stored in the ROOT file. Additionally, to the particle information, a further branch containing the event-related data is created. Besides the calculated weight, the parameters randomly generated in the final state generator are also stored in this branch. The explicit specification of the two TTree structure used can be found in code D.1.

The produced ROOT TTree was used to perform more complex analysis on the data, such as calculating the expected statistical errors for the measurement of the E1 and E2 ground state distribution of the S-factor measurement. This was done in several steps.

In the first step the data is splitted into small cms energy ranges accordingly. For example the cms energy range  $[0.3 \text{ MeV}, 3 \text{ MeV}]$  was splitted in smaller ranges  $[E_i, E_{i+1})$  with  $E_0 = 0.3 \text{ MeV}$  and  $E_{i+1} = E_i + \Delta E$  for  $\Delta E = 0.05 \text{ MeV}$ . For each energy range a histogram was produced entering  $\cos(\theta_\alpha^{\text{cms}})$  with the corresponding weights. To get the distribution of expected detected events over the cms angle of the  $\alpha$ -particle, each histogram was scaled with the factor  $L/N$ , with  $N$  being the total number of generated events. The number of particles detected in a time intervall  $T$  with time independent particle flux and time independent detector efficiency is Poisson distributed [Lyo86]. Therefore, the error for each bin  $k$  was approximated by using the formula

$$\sqrt{N_k + 2}, \quad (6.7)$$

where  $N_k$  is the number of events of bin  $k$ . The  $+2$  in the square root is used to also include the empty bars. Figure 6.6 shows an example of a histogram including the approximated errors for Phase 2.



**Figure 6.6:** Number of expected events distributed over the cms angular of the  $\alpha$ -particle in the cms energy range  $E_{cms} = 1.3 - 1.4$  MeV. The error of each bin  $k$  is calculated using  $\sqrt{N_k + 2}$  where  $N_k$  is the number of events of bin  $k$ .

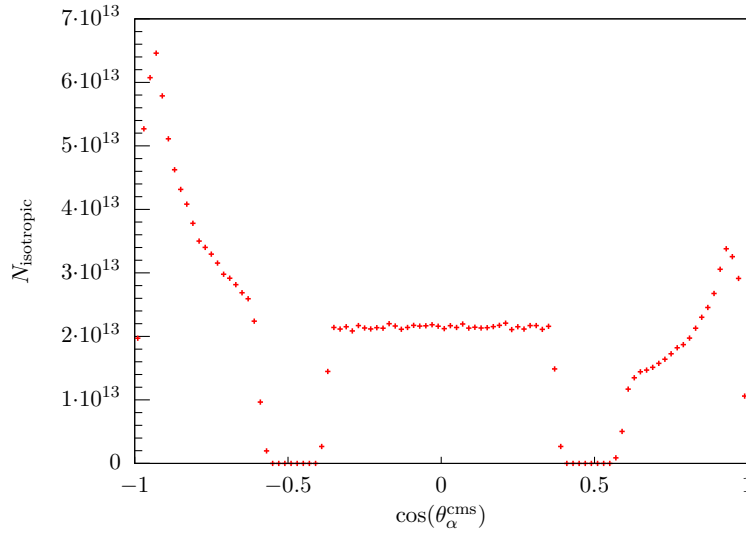
In addition one histogram representing the isotropic generated value  $\cos(\theta_\alpha^{cms})$  using weight  $w = 1$  was produced. For this histogram the data of the entire energy range was taken into account. This histogram is also scaled by  $L/N$  and the error for each bin was calculated using Eq. (6.7). Figure 6.6 shows an example of this histogram including the approximated errors. Even if the cms angle of the  $\alpha$ -particle is isotropically generated the result of this histogram looks different. This is due to the acceptance, as the projection of the silicon strip detectors on the surface of the unit sphere is not conformal. In Fig. 6.5 one can already see that for the assumed detector setup the acceptance in the range close to  $\pm 1$  for  $\cos(\theta_\alpha^{cms})$  is higher.

In the second step the statistical error of the MC integration must be eliminated. Therefore, each histogram is normalized by the isotropic histogram. Technically, this means that the value of each bin of a histogram is divided by the corresponding value of the bin of the isotropic histogram. The errors are calculated by simple error propagation. As a result, for each cms energy range, one obtains a histogram representing the distribution of the cross section  $\sigma$  for the cms angle of the  $\alpha$ -particle. An example of one of these produced histograms can be found in Fig. 6.8.

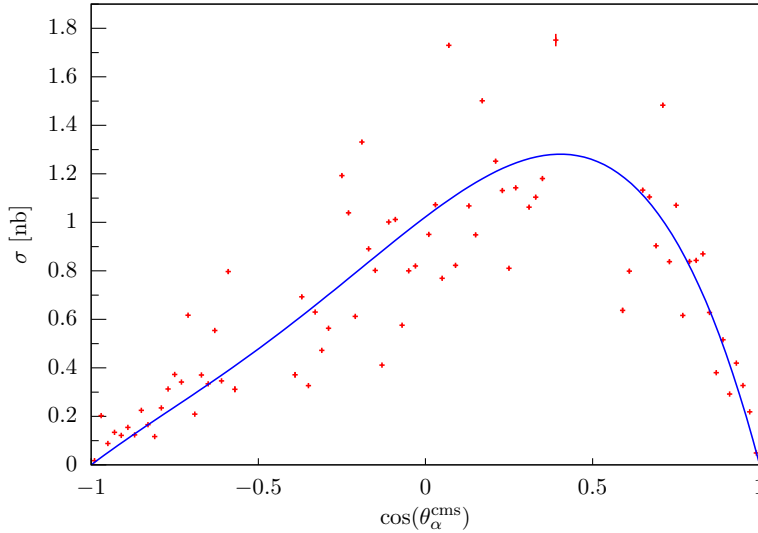
In the last step each of the produced normalized histograms were fitted using Eq. (2.31) with the fit parameters  $a := \sigma_{E1}$  and  $b := \sigma_{E2}/\sigma_{E1}$  as discussed in Section 2.3.1. The phase shift was calculated using Eq. (2.30). The result of such a 2-parameter fit can be found in Fig. 6.8. As result of each fit one gets the values  $\sigma_{E1}(E_i)$  and  $\sigma_{E2}(E_i)$  and the corresponding errors  $\Delta\sigma_{E1}(E_i)$  and  $\Delta\sigma_{E2}(E_i)$ . The error  $\Delta\sigma_{E1}(E_i)$  is equal to the error  $\Delta a$  of the fit parameter. The error for E2 was calculated using error propagation

$$\Delta\sigma_{E2}(E_i) = \sigma_{E1}(E_i) \cdot \Delta b + b \cdot \Delta\sigma_{E1}(E_i).$$

The errors correspond to the statistical errors that are expected for the measurement with an identical detector setup. Figure 6.9 shows the results of this analysis



**Figure 6.7:** The histogram represents the isotropically distributed cms angle of the  $\alpha$ -particle. The error of each bin  $k$  is calculated using  $\sqrt{N_k + 2}$  where  $N_k$  is the number of events of bin  $k$ .



**Figure 6.8:** The histogram represents the weighted cms angle of the  $\alpha$ -particle for the energy range  $E_{cms} = 1.3 - 1.4$  MeV corresponding to the cross section distribution. The error of the MC integration is eliminated as described in the text. The data is fitted using Eq. (2.31).

for the different Phases 1 and 2 for one specific detector setup. To cross-check the validity of the simulation results, an independent simulation was performed within the MAGIX group based on the theoretical assumptions and calculations described in this thesis. It confirmed the results, that will be discussed in the next section.

### 6.2.6 Discussion of the Simulations Results

In this section we will discuss the results of the simulations performed as part of this work. A summary of the parameters used for the simulations and the analysis of the electro-disintegration measurement at MAGIX for Phases 1 and 2 are given in Table 6.3. The results of the simulations can be found in Fig. 6.9.

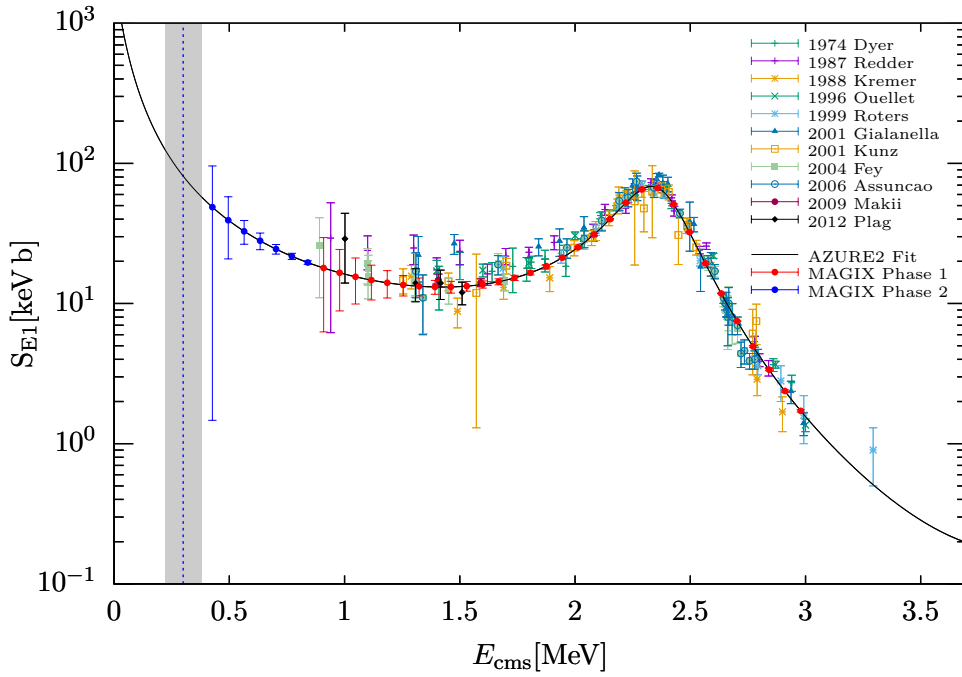
**Table 6.3:** Summary of the relevant simulation and analysis parameters.

Parameter	Value	
Electron Beam <sup>1,2</sup>	energy	105 MeV
	current	1 mA
Oxygen Target <sup>1,2</sup>	areal number density	$2 \times 10^{18}$ particles/cm <sup>2</sup>
	gas type	pure <sup>16</sup> O
Spectrometer <sup>1</sup>	central momentum	75 MeV
	central in-plane angle	13°
Zero Degree Tagger <sup>2</sup>	energy acceptance	$E'_e \geq 7$ MeV
	angular acceptance	$\theta_e \leq 0.5^\circ$
Silicon-Strip Detector Array <sup>1,2</sup>	detector count	5
	active detector area	$50 \times 50$ mm <sup>2</sup>
	distance	10 cm
	central in-plane angle	30°, ±90°, ±120°
Luminosity <sup>1,2</sup>	$1 \times 10^{34}$ cm <sup>-2</sup> s <sup>-1</sup>	
Time Period <sup>1,2</sup>	4 weeks	
Integrated Luminosity <sup>1,2</sup>	$2.4192 \times 10^{40}$ cm <sup>-2</sup> $= 2.4192 \times 10^7$ nb <sup>-1</sup>	
$E_\alpha^{\text{cms}}$ -Range <sup>1,2</sup>	0 – 25 MeV	

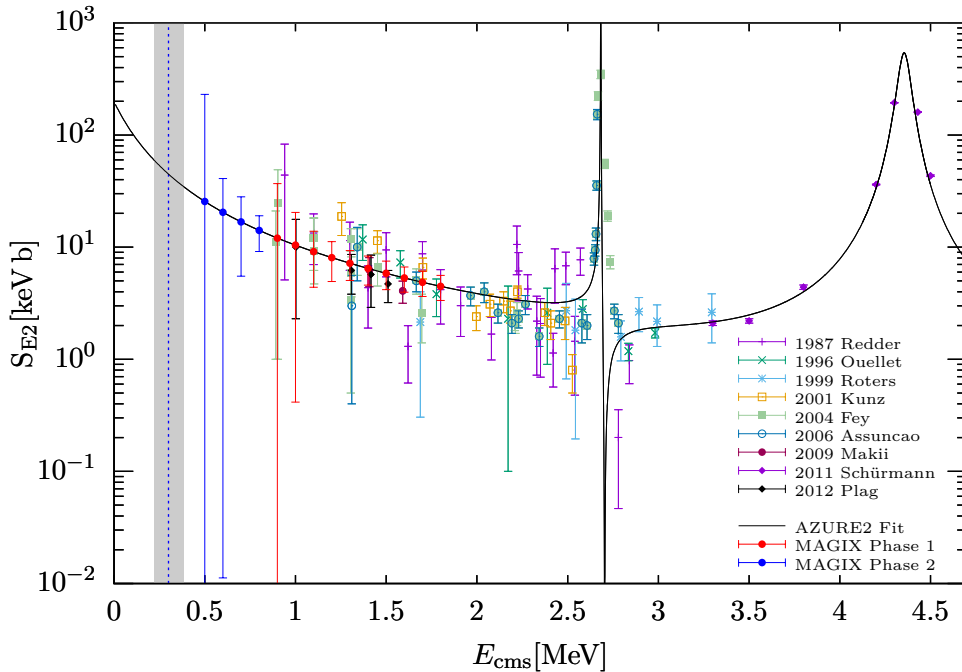
<sup>1</sup> MAGIX Phase 1<sup>2</sup> MAGIX Phase 2

In Figs. 6.9a and 6.9b, the data from previous measurements is given for the E1 and E2 ground state contributions. The E1 and E2 contribution are fitted using the AZURE2 software [Azu+10; Ud15] which was configured with the parameters from the supplementary material of Ref. [deB+17]. The results of the simulations are added to these figures to show the expected errors compared to the existing data. For this purpose, the S-factor values calculated with AZURE2 were added to the figures with the calculated errors from the simulations. In Fig. 6.9a the simulated results for the E1 contribution are shown. As it can be seen, the expected statistical errors for Phase 1 have a precision comparable to the existing data in the range above 0.9 MeV. A similar result is obtained for the E2 contribution Fig. 6.9b. As expected, the error for the E2 contribution is larger than for the E1 contribution due to error propagation. Moreover, it can be seen that for Phase 2, the expected errors in the region above 0.9 MeV decrease greatly and, in addition, it will be possible to study the region below. With this particular detector setup, the E1 contribution can be determined down to 0.5 MeV and the E2 contribution down to 0.7 MeV. For both components it will be possible to make contributions in the range below 0.9 MeV in Phase 2 and thus reducing the gap to the Gamow peak.

The results of these simulations show that MAGIX can make significant measurements of the S-factor. A reduction of the errors in E1 and E2 can be achieved by MAGIX in two ways. The first important point is that reducing the errors in the region above 1 MeV allows a more accurate extrapolation of the S-factor to the Gamow peak. This reduces the errors caused by the extrapolation. The second point is that the simulations show that MAGIX can measure the S-factor down to 0.5 MeV in Phase 2. This will reduce the gap to the Gamow peak and thus improve the extrapolation further.



(a) E1 contribution data [DB74; Red+87; Kre+88; Oue+96; Rot+99; Gia+01; Kun+01; Fey04; Ass+06; Mak+09; Pla+12].



(b) E2 contribution data [Red+87; Oue+96; Rot+99; Kun+01; Fey04; Ass+06; Mak+09; Sch+11; Pla+12].

**Figure 6.9:** E1 (a) and E2 (b) contribution data fitted with the AZURE2 software [Azu+10; Ud15] configured with parameters of the supplemental material provided by [deB+17]. MAGIX Phase 1/2 shows the projected errors for the Phases 1 and 2.

Although the longitudinal and the two interference terms were not included in this simulation, the simulation provides a good estimate of the expected statistical errors of the S-factor measurement at MAGIX, since the transversal term dominates in the low  $Q^2$  range, whereas the omitted terms only play a minor role, as discussed in Section 6.2.3. Compared to the photo-disintegration, the



electro-disintegration contains more nuclear response functions. This means that electro-disintegration has a potentially higher information content. This will allow a more accurate determination of the various contributions. In particular, the error in E1 and E2 can be reduced. For more details, please refer to [FDM19].

At this point it is important to note that the results given are the results of a simulation with one specific detector setup. The position of the silicon strip detectors have been chosen in favor of the measurement of E1 (see Fig. 2.12). In the experiment, several measurements will be performed with different detector positions optimized for either E1 or E2. This will allow the reduction of the errors of the E1 and E2 contribution best as possible

As mentioned at the beginning of the chapter, no simulation was performed to determine the background. This is due to the fact that no final information on the beam parameters of MESA or the efficiency of the detectors is currently available.

Compared to the direct measurements e.g. at LUNA-MV or Felsenkeller laboratory, the expected background from cosmic particles is low, due to the coincidence measurement of the scattered electrons and the recoiled  $\alpha$ -particles. This allows MAGIX to perform the measurement next to an accelerator and not several hundred meters under the ground. The detectors will be optimized for their respective particle species. The amount of background signals in the final experiment and how effectively they can be separated from the signals depends on the efficiency of the detectors and other experimental conditions. Both are currently unknown and must be determined experimentally.

One known background source are the signals arising from the contamination of  $^{17}\text{O}$ ,  $^{18}\text{O}$ , and  $^{15}\text{N}$  in the oxygen gas (see Section 4.3.1). As described in Ref. [FDM19], a ToF measurement can be used to separate the signals arising from the contamination. To ensure this, the silicon strip detectors are additionally optimized for high time resolution. Verifying whether it is possible to identify these background reactions and determining the efficiency of the silicon strip detectors will be part of the Phase 0 measurement at the A1 spectrometer facility at MAMI.

As calculated in the simulation we expect a very low event rate for cms energies near the Gamow peak which makes it quite challenging to gain high statistics. Nevertheless, MAGIX can still make a decisive contribution to the determination of the S-factor. For cms energies above 0.9 MeV, the expected counting rates are high, so that in Phase 2 the statistical errors in this range will be low. This allows to perform a more accurate extrapolation towards the Gamow peak and consequently to reduce the error in the S-factor determination. Each additional data point that can be taken in the sub-MeV scale improves the extrapolation.

Furthermore, the beam time period is a variable parameter. In addition the integrated luminosity can be increased by the planned upgrade of the beam current of MESA to 10 mA. This means that statistical errors can be reduced due to longer measuring times and a increased beam current.



---

---

# CHAPTER 7

---

## Summary and Outlook

The S-factor of the reaction  $^{12}\text{C}(\alpha, \gamma)^{16}\text{O}$  plays a decisive role in describing the pre-supernova evolution of stars. As outlined in Section 2.5 ‘*Production Factor*’, the S-factor is a crucial factor in the calculation of the isotope abundance ratio in a star. Despite its importance and several attempts to determine the S-factor at the Gamow peak  $E_G = 0.3$  MeV experimentally, the relative error of the S-factor is currently greater than 17 %. An overview and discussion of the already existing data can be found in Section 2.4 ‘*Results of Existing Measurements*’. However, to make reasonable predictions about pre-supernova evolution, a relative uncertainty of under 10 % is required. After considering all theoretical aspects as laid out in Chapter 2 ‘*The Reaction  $^{12}\text{C}(\alpha, \gamma)^{16}\text{O}$  in the Helium Burning Stage in Stars*’, it turned out that the problem in determining the S-factor is that the cross section of the reaction  $^{12}\text{C}(\alpha, \gamma)^{16}\text{O}$  near the Gamow peak is dominated by the Coulomb penetration factor. As a consequence, the cross section decreases exponentially from  $\sim 10^{-11}$  b at 1 MeV to  $\sim 10^{-17}$  b at 0.3 MeV. This makes an experimental determination of the S-factor at the Gamow peak so challenging. Thus, low-energy high-precision experiments are needed to measure the S-factor near the Gamow peak. In the coming years, several experiments are planned to measure the S-factor using different methods. In Section 2.3 ‘*Discussion of Measurement Techniques*’ an overview of different measurement techniques is given. Besides direct measurements, e.g. planned at LUNA and Felsenkeller in Dresden, indirect measurements via photo-disintegration, e.g. at the HI $\gamma$ S facility or at the ELI-NP facility, are planned.

Another indirect measurement technique is the electro-disintegration measurement of oxygen. For this purpose, electrons are scattered on oxygen and the scattered electrons and the recoiled  $\alpha$ -particles are detected in coincidence. According to the theoretical derivations made in Chapter 3 ‘*Electro-Disintegration Cross Section of  $^{16}\text{O}$* ’ the E1 and E2 ground state contribution of the S-factor of the reaction  $^{12}\text{C}(\alpha, \gamma)^{16}\text{O}$  can be determined from the electro-disintegration cross section of oxygen.

This technique is applied for the planned experiment MAGIX. MAGIX will be commissioned in the coming years at the high-intensity electron accelerator MESA in Mainz, Germany, which is currently under construction. In addition to measur-

ing the S-factor of the  $^{12}\text{C}(\alpha, \gamma)^{16}\text{O}$ , electromagnetic form factors or searching for dark sector particles, and reaction studies of astrophysical relevance are planned. The experimental setup of MAGIX at MESA for determining the S-factor is discussed in Chapter 4 ‘*Accelerator and Experimental Setup*’. This measurement will be conducted in three phases (Phase 0, Phase 1 and Phase 2).

To estimate the expected statistical errors of the Phase 1 and 2 of the planned electro-disintegration measurement at MAGIX a simulation was performed as part of this thesis. Since in Phase 2 the scattering angle of the electrons will be less than  $0.5^\circ$ , the cross section had to be derived without using the ERL. The derivations of the electro-disintegration cross section without using ERL is outlined in Chapter 3 ‘*Electro-Disintegration Cross Section of  $^{16}\text{O}$* ’. Since information about the efficiency of the detectors and the beam parameters is not yet available, no background simulations were included. Furthermore, due to this reason, the efficiency could not be included in the simulation of the acceptance of the detectors. They are therefore determined as purely geometrically.

Based on the results of this simulation discussed in Chapter 6 ‘*Simulation of the Electro-Disintegration Reaction  $^{16}\text{O}(e, e'\alpha)^{12}\text{C}$* ’, it can be stated that MAGIX will be able to determine the S-factor in the low-energy range near the Gamow peak of the helium burning stage  $E_G = 0.3 \text{ MeV}$ . Already in Phase 1, MAGIX will be able to determine the S-factor in the cms energy range  $E_\alpha^{\text{cms}} > 1 \text{ MeV}$  with relative errors competitive to previous experiments performed by other research groups. In Phase 2, which includes the zero degree tagger, MAGIX can then reduce the relative errors above 1 MeV and perform high-precision measurements even on a sub-MeV scale with a superior statistical significance.

The main conclusion from this thesis is that MAGIX can make an important contribution to the determination of the S-factor in the coming years. Based on these results, MAGIX will continue to plan and prepare measurements of the S-factor. The simulation was only one of several steps that will contribute to the success of the measurement of the S-factor for the reaction  $^{12}\text{C}(\alpha, \gamma)^{16}\text{O}$ .

Even though the simulation is promising, it has its limitations. Although background simulations could not be included, important conclusions can be drawn from the results of this simulation. The most important aspects were discussed in Section 6.2.6 ‘*Discussion of the Simulations Results*’. Despite the fact that Phase 1 will not begin until 2023 at the earliest, it was very important to perform this simulation at an early stage. The development of the detectors, especially the silicon strip detector, is closely related to the results of this work. During the development of the simulation, the analysis of the simulation was adjusted to obtain important parameters needed for the development of the silicon strip detector.

An essential next step is the experimental determination of the efficiency of the silicon strip detectors, especially for low cms energies. This will be part of Phase 0, in which the MAGIX gas jet target will be operated at the A1 facility at MAMI in Mainz using oxygen. A further important step in Phase 0 will be verifying the ToF method to separate the signals arising from the contamination of  $^{17}\text{O}$ ,  $^{18}\text{O}$ , and  $^{15}\text{N}$  in the oxygen gas. The information gained in Phase 0 will then in turn be used to improve the simulation in particular including background calculations and the efficiency of the silicon strip detector.

Some MAGIX components, such as the gas jet target or the silicon strip detector that will be used to detect the  $\alpha$ -particles, are already tested in different labora-

---

tories. Another important part of this thesis was the development of the MAGIX slow control system, which will later be used in the final experiment. As described in Chapter 5 '*Development of the MAGIX Slow Control System*' the slow control system was extended by those test stands. To adapt the slow control system based on the needs of MAGIX, the EPICS base software was extended by several MAGIX modules that were build as part of this thesis. Running the test stands with the MAGIX Slow Control system served as a stress test of the software. This showed how reliable an EPICS-based slow control system can work even under load. The structure of the EPICS-based MAGIX slow control was designed in such a way that all devices that will later be needed for MAGIX can easily be integrated. Special care was taken to ensure that the slow control can be used in exactly the same way for the final experiment. To support users in getting started with MXSlowControl and to provide professional tips for more complex tasks to all team members a tutorial was created as part of this thesis. Under consideration of the continuous development of MXSlowControl this tutorial needs to be updated constantly.

In summary, the simulation was a very important step to enable the measurement of the S-factor of the  $^{12}\text{C}(\alpha, \gamma)^{16}\text{O}$  at MAGIX. In combination with the development of the slow control for MAGIX as part of this thesis, MAGIX will be able to prepare and perform Phase 0 up to Phase 2 based on the theoretical work carried out in this thesis.



---



---

# APPENDIX A

---

## Response Functions

Following the derivation published in [RD89; FDM19] the nuclear response functions  $R^K$  in terms of the Legendre and the associated Legendre polynomials up to octupole contributions can be written as:

$$\begin{aligned}
R_L = & P_0(\cos \theta_\alpha) \left( |t_{C0}|^2 + |t_{C1}|^2 + |t_{C2}|^2 + |t_{C3}|^2 \right) \\
& + P_1(\cos \theta_\alpha) \left( 2\sqrt{3}|t_{C0}||t_{C1}| \cos(\delta_{C1} - \delta_{C0}) + 4\sqrt{\frac{3}{5}}|t_{C1}||t_{C2}| \cos(\delta_{C2} - \delta_{C1}) \right. \\
& \quad \left. + \frac{18}{\sqrt{35}}|t_{C2}||t_{C3}| \cos(\delta_{C3} - \delta_{C2}) \right) \\
& + P_2(\cos \theta_\alpha) \left( 2|t_{C1}|^2 + \frac{10}{7}|t_{C2}|^2 + 2\sqrt{5}|t_{C0}||t_{C2}| \cos(\delta_{C2} - \delta_{C0}) \right. \\
& \quad \left. + 6\sqrt{\frac{3}{7}}|t_{C1}||t_{C3}| \cos(\delta_{C3} - \delta_{C1}) + \frac{4}{3}|t_{C3}|^2 \right) \\
& + P_3(\cos \theta_\alpha) \left( 6\sqrt{\frac{3}{5}}|t_{C1}||t_{C2}| \cos(\delta_{C2} - \delta_{C1}) + \frac{8}{3}\sqrt{\frac{7}{5}}|t_{C2}||t_{C3}| \cos(\delta_{C3} - \delta_{C2}) \right. \\
& \quad \left. + 2\sqrt{7}|t_{C0}||t_{C3}| \cos(\delta_{C3} - \delta_{C0}) \right) \\
& + P_4(\cos \theta_\alpha) \left( \frac{18}{7}|t_{C2}|^2 + 8\sqrt{\frac{3}{7}}|t_{C1}||t_{C3}| \cos(\delta_{C3} - \delta_{C1}) + \frac{18}{11}|t_{C3}|^2 \right) \\
& + P_5(\cos \theta_\alpha) \left( \frac{20}{3}\sqrt{\frac{5}{7}}|t_{C2}||t_{C3}| \cos(\delta_{C3} - \delta_{C2}) \right) \\
& + P_6(\cos \theta_\alpha) \left( \frac{100}{33}|t_{C3}|^2 \right), \\
R_T = & P_0(\cos \theta_\alpha) \left( |t_{E1}|^2 + |t_{E2}|^2 + |t_{E3}|^2 \right) \\
& + P_1(\cos \theta_\alpha) \left( \frac{6}{\sqrt{5}}|t_{E1}||t_{E2}| \cos(\delta_{E2} - \delta_{E1}) + 12\sqrt{\frac{2}{35}}|t_{E2}||t_{E3}| \cos(\delta_{E3} - \delta_{E2}) \right) \\
& + P_2(\cos \theta_\alpha) \left( -|t_{E1}|^2 + \frac{5}{7}|t_{E2}|^2 + 6\sqrt{\frac{2}{7}}|t_{E1}||t_{E3}| \cos(\delta_{E3} - \delta_{E1}) + |t_{E3}|^2 \right) \\
& + P_3(\cos \theta_\alpha) \left( -\frac{6}{\sqrt{5}}|t_{E1}||t_{E2}| \cos(\delta_{E2} - \delta_{E1}) + \frac{2}{3}\sqrt{\frac{14}{5}}|t_{E2}||t_{E3}| \cos(\delta_{E3} - \delta_{E2}) \right)
\end{aligned}$$

$$\begin{aligned}
 & + P_4(\cos \theta_\alpha) \left( -\frac{12}{7} |t_{E2}|^2 - 6\sqrt{\frac{2}{7}} |t_{E1}| |t_{E3}| \cos(\delta_{E3} - \delta_{E1}) + \frac{3}{11} |t_{E3}|^2 \right) \\
 & + P_5(\cos \theta_\alpha) \left( -\frac{10}{3} \sqrt{\frac{10}{7}} |t_{E2}| |t_{E3}| \cos(\delta_{E3} - \delta_{E2}) \right) \\
 & + P_6(\cos \theta_\alpha) \left( -\frac{25}{11} |t_{E3}|^2 \right), \\
 R_{TL} = & \left\{ P_1^1(\cos \theta_\alpha) \left( 2\sqrt{3} |t_{C0}| |t_{E1}| \cos(\delta_{E1} - \delta_{C0}) - 2\sqrt{\frac{3}{5}} |t_{C2}| |t_{E1}| \cos(\delta_{C2} - \delta_{E1}) \right. \right. \\
 & + \frac{6}{\sqrt{5}} |t_{C1}| |t_{E2}| \cos(\delta_{C1} - \delta_{E2}) - 6\sqrt{\frac{3}{35}} |t_{C3}| |t_{E2}| \cos(\delta_{C3} - \delta_{E2}) \\
 & \left. \left. + 6\sqrt{\frac{6}{35}} |t_{C2}| |t_{E3}| \cos(\delta_{C2} - \delta_{E3}) \right) \right. \\
 & + P_2^1(\cos \theta_\alpha) \left( 2 |t_{C1}| |t_{E1}| \cos(\delta_{C1} - \delta_{E1}) + 2\sqrt{\frac{5}{3}} |t_{C0}| |t_{E2}| \cos(\delta_{E2} - \delta_{C0}) \right. \\
 & + \frac{10}{7\sqrt{3}} |t_{C2}| |t_{E2}| \cos(\delta_{C2} - \delta_{E2}) - 2\sqrt{\frac{3}{7}} |t_{C3}| |t_{E1}| \cos(\delta_{C3} - \delta_{E1}) \\
 & \left. + 4\sqrt{\frac{2}{7}} |t_{C1}| |t_{E3}| \cos(\delta_{C1} - \delta_{E3}) + \frac{2}{3} \sqrt{\frac{2}{3}} |t_{C3}| |t_{E3}| \cos(\delta_{C3} - \delta_{E3}) \right) \\
 & + P_3^1(\cos \theta_\alpha) \left( 2\sqrt{\frac{3}{5}} |t_{C2}| |t_{E1}| \cos(\delta_{C2} - \delta_{E1}) + \frac{4}{\sqrt{5}} |t_{C1}| |t_{E2}| \cos(\delta_{C1} - \delta_{E2}) \right. \\
 & + 2\sqrt{\frac{7}{6}} |t_{C0}| |t_{E3}| \cos(\delta_{E3} - \delta_{C0}) + \frac{2}{3} \sqrt{\frac{7}{15}} |t_{C3}| |t_{E2}| \cos(\delta_{C3} - \delta_{E2}) \\
 & \left. + 2\sqrt{\frac{7}{30}} |t_{C2}| |t_{E3}| \cos(\delta_{C2} - \delta_{E3}) \right) \\
 & + P_4^1(\cos \theta_\alpha) \left( \frac{6\sqrt{3}}{7} |t_{C2}| |t_{E2}| \cos(\delta_{C2} - \delta_{E2}) + 2\sqrt{\frac{3}{7}} |t_{C3}| |t_{E1}| \cos(\delta_{C3} - \delta_{E1}) \right. \\
 & \left. + \frac{6}{\sqrt{14}} |t_{C1}| |t_{E3}| \cos(\delta_{C1} - \delta_{E3}) + \frac{6}{11} \sqrt{\frac{3}{2}} |t_{C3}| |t_{E3}| \cos(\delta_{C3} - \delta_{E3}) \right) \\
 & + P_5^1(\cos \theta_\alpha) \left( \frac{40}{3} \frac{1}{\sqrt{105}} |t_{C3}| |t_{E2}| \cos(\delta_{C3} - \delta_{E2}) + 10\sqrt{\frac{2}{105}} |t_{C2}| |t_{E3}| \cos(\delta_{C2} - \delta_{E3}) \right) \\
 & \left. + P_6^1(\cos \theta_\alpha) \left( \frac{50}{33} \sqrt{\frac{2}{3}} |t_{C3}| |t_{E3}| \cos(\delta_{C3} - \delta_{E3}) \right) \right\} \cdot \cos(\phi_\alpha)
 \end{aligned}$$

$$R_{TT} = -R_T \cos(2\phi_\alpha).$$

For the simulation, we used the nuclear response function where the Legendre polynomials are expressed in terms of sine and cosine: [FDM19]

$$\begin{aligned}
 R_L = & |t_{C0}|^2 + 3|t_{C1}|^2 \cos^2(\theta_\alpha) + \frac{5}{16} |t_{C2}|^2 (1 + 3 \cos(2\theta_\alpha))^2 + \frac{7}{64} |t_{C3}|^2 (3 \cos(\theta_\alpha) + 5 \cos(3\theta_\alpha))^2 \\
 & + 2\sqrt{3} |t_{C0}| |t_{C1}| \cos(\theta_\alpha) \cos(\delta_{C1} - \delta_{C0}) + \frac{\sqrt{5}}{2} |t_{C0}| |t_{C2}| (1 + 3 \cos(2\theta_\alpha)) \cos(\delta_{C2} - \delta_{C0}) \\
 & + \frac{\sqrt{7}}{4} |t_{C0}| |t_{C3}| (3 \cos(\theta_\alpha) + 5 \cos(3\theta_\alpha)) \cos(\delta_{C3} - \delta_{C0}) \\
 & + \frac{\sqrt{15}}{4} |t_{C1}| |t_{C2}| (5 \cos(\theta_\alpha) + 3 \cos(3\theta_\alpha)) \cos(\delta_{C2} - \delta_{C1}) \\
 & + \frac{\sqrt{21}}{8} |t_{C1}| |t_{C3}| (3 + 8 \cos(2\theta_\alpha) + 5 \cos(4\theta_\alpha)) \cos(\delta_{C3} - \delta_{C1}) \\
 & + \frac{\sqrt{35}}{32} |t_{C2}| |t_{C3}| (30 \cos(\theta_\alpha) + 19 \cos(3\theta_\alpha) + 15 \cos(5\theta_\alpha)) \cos(\delta_{C3} - \delta_{C2}),
 \end{aligned}$$



$$\begin{aligned}
 R_T = & \frac{3}{2}|t_{E1}|^2 \sin^2(\theta_\alpha) + \frac{15}{8}|t_{E2}|^2 \sin^2(2\theta_\alpha) + \frac{21}{256}|t_{E3}|^2 (\sin(2\theta_\alpha) + 5 \sin(3\theta_\alpha))^2 \\
 & + \frac{3\sqrt{5}}{2}|t_{E1}||t_{E2}|(\sin(\theta_\alpha) \sin(2\theta_\alpha)) \cos(\delta_{E2} - \delta_{E1}) \\
 & + \frac{3}{2}\sqrt{\frac{5}{2}}|t_{E1}||t_{E3}|(\sin^2(2\theta_\alpha) - \sin^4(\theta_\alpha)) \cos(\delta_{E3} - \delta_{E1}) \\
 & + \frac{3}{4}\sqrt{\frac{35}{2}}|t_{E2}||t_{E3}| \sin(2\theta_\alpha)(\sin(\theta_\alpha) + \sin(3\theta_\alpha) - \sin^3(\theta_\alpha)) \cos(\delta_{E3} - \delta_{E2}) \\
 R_{TL} = & \left\{ \begin{aligned}
 & -2\sqrt{3}|t_{C0}||t_{E1}| \sin(\theta_\alpha) \cos(\delta_{E1} - \delta_{C0}) - \sqrt{15}|t_{C0}||t_{E2}| \sin(2\theta_\alpha) \cos(\delta_{E2} - \delta_{C0}) \\
 & - \frac{1}{4}\sqrt{\frac{21}{2}}|t_{C0}||t_{E3}|(\sin(\theta_\alpha) + 5 \sin(3\theta_\alpha)) \cos(\delta_{E3} - \delta_{C0}) \\
 & - 3|t_{C1}||t_{E1}| \sin(2\theta_\alpha) \cos(\delta_{E1} - \delta_{C1}) - \frac{3\sqrt{5}}{2}|t_{C1}||t_{E2}|(\sin(\theta_\alpha) + \sin(3\theta_\alpha)) \cos(\delta_{E2} - \delta_{C1}) \\
 & - \frac{3}{8}\sqrt{\frac{7}{2}}|t_{C1}||t_{E3}|(6 \sin(2\theta_\alpha) + 5 \sin(4\theta_\alpha)) \cos(\delta_{E3} - \delta_{C1}) \\
 & + \frac{\sqrt{15}}{4}|t_{C2}||t_{E1}|(\sin(\theta_\alpha) - 3 \sin(3\theta_\alpha)) \cos(\delta_{E1} - \delta_{C2}) \\
 & - \frac{5\sqrt{3}}{8}|t_{C2}||t_{E2}|(2 \sin(2\theta_\alpha) - 3 \sin(4\theta_\alpha)) \cos(\delta_{E2} - \delta_{C2}) \\
 & + \frac{1}{32}\sqrt{\frac{3}{70}}|t_{C2}||t_{E3}|(278 \sin(\theta_\alpha) - 455 \sin(3\theta_\alpha) - 525 \sin(5\theta_\alpha)) \cos(\delta_{E3} - \delta_{C2}) \\
 & + \frac{\sqrt{21}}{8}|t_{C3}||t_{E1}|(2 \sin(2\theta_\alpha) - 5 \sin(4\theta_\alpha)) \cos(\delta_{E1} - \delta_{C3}) \\
 & + \frac{\sqrt{105}}{16}|t_{C3}||t_{E2}|(2 \sin(\theta_\alpha) - 3 \sin(3\theta_\alpha) - 5 \sin(5\theta_\alpha)) \cos(\delta_{E2} - \delta_{C3}) \\
 & - \frac{7}{64}\sqrt{\frac{3}{2}}|t_{C3}||t_{E3}|(13 \sin(2\theta_\alpha) + 20 \sin(4\theta_\alpha) + 25 \sin(6\theta_\alpha)) \cos(\delta_{E3} - \delta_{C3}) \end{aligned} \right\} \cdot \cos(\phi_\alpha), \\
 R_{TT} = & -R_T \cos(2\phi_\alpha).
 \end{aligned}$$

Using the Rodrigues' formula Eq. (3.50) the Legendre polynomials up to  $n = 6$  are

$$\begin{aligned}
 P_0(x) &= 1, \\
 P_1(x) &= x, \\
 P_2(x) &= \frac{1}{2}(3x^2 - 1), \\
 P_3(x) &= \frac{1}{2}(5x^3 - 3x), \\
 P_4(x) &= \frac{1}{8}(35x^4 - 30x^2 + 3), \\
 P_5(x) &= \frac{1}{8}(63x^5 - 70x^3 + 15x), \\
 P_6(x) &= \frac{1}{16}(231x^6 - 315x^4 + 105x^2 - 5).
 \end{aligned}$$

Using Eq. (3.51) the associated Legendre polynomials with  $m = 1$  up to  $n = 6$  are

$$\begin{aligned}
 P_1^1(x) &= -\sqrt{1-x^2}, \\
 P_2^1(x) &= -3x\sqrt{1-x^2}, \\
 P_3^1(x) &= -\frac{3}{2}(5x^2 - 1)\sqrt{1-x^2},
 \end{aligned}$$

$$\begin{aligned}
 P_4^1(x) &= -\frac{5}{2}(7x^3 - 3x)\sqrt{1-x^2}, \\
 P_5^1(x) &= -\frac{15}{8}(21x^4 - 14x^2 + 1)\sqrt{1-x^2}, \\
 P_6^1(x) &= -\frac{21}{8}(33x^5 - 30x^3 + 5x)\sqrt{1-x^2}.
 \end{aligned}$$

For  $x = \cos(\theta)$  the Legendre polynomials are given as

$$\begin{aligned}
 P_0(\cos(\theta)) &= 1, \\
 P_1(\cos(\theta)) &= \cos(\theta), \\
 P_2(\cos(\theta)) &= \frac{1}{2}(3\cos^2(\theta) - 1), \\
 P_3(\cos(\theta)) &= \frac{1}{2}(5\cos^3(\theta) - 3\cos(\theta)), \\
 P_4(\cos(\theta)) &= \frac{1}{8}(35\cos^4(\theta) - 30\cos^2(\theta) + 3), \\
 P_5(\cos(\theta)) &= \frac{1}{8}(63\cos^5(\theta) - 70\cos^3(\theta) + 15\cos(\theta)), \\
 P_6(\cos(\theta)) &= \frac{1}{16}(231\cos^6(\theta) - 315\cos^4(\theta) + 105\cos^2(\theta) - 5).
 \end{aligned}$$

The associated Legendre polynomials for  $x = \cos(\theta)$  with  $0 \leq \theta \leq \pi$  in terms of sine and cosine can be written as

$$\begin{aligned}
 P_1^1(\cos(\theta)) &= -\sin(\theta), \\
 P_2^1(\cos(\theta)) &= -3\sin(\theta)\cos(\theta), \\
 P_3^1(\cos(\theta)) &= -\frac{3}{2}\sin(\theta)(5\cos^2(\theta) - 1), \\
 P_4^1(\cos(\theta)) &= -\frac{5}{2}\sin(\theta)\cos(\theta)(7\cos^2(\theta) - 3), \\
 P_5^1(\cos(\theta)) &= -\frac{15}{8}\sin(\theta)(21\cos^4(\theta) - 14\cos^2(\theta) + 1), \\
 P_6^1(\cos(\theta)) &= -\frac{21}{8}\sin(\theta)\cos(\theta)(33\cos^4(\theta) - 30\cos^2(\theta) + 5).
 \end{aligned}$$

---



---

# APPENDIX B

---

## Selected Data

In this chapter all nuclear data used for the calculations or the simulations is given. Tables B.1 to B.4 represents the list of relevant adopted levels provided by [NuDat]. In these tables, the errors are indicated separately by spaces. For example, 5.57 eV 25 corresponds to  $5.57(25) \text{ eV} \equiv 5.57 \pm 0.25 \text{ eV}$ .

**Table B.1:** *List of adopted levels for  $^8\text{Be}$  [NuDat].*

$E(\text{level})$ (keV)	$J^\pi$	$\Gamma(\text{level})$	$E(\gamma)$ (keV)	$M(\gamma)$	Final level $J^\pi(\text{keV})$
0	$0^+$	5.57 eV 25 $\alpha = 100$			
3030 10	$2^+$	1513 keV 15 $\alpha \approx 100$	3029	E2	$0^+(0)$
11.35E+3 15	$4^+$	$\approx 3.5 \text{ MeV}$ $\alpha \approx 100$	11350	E4	$0^+(0)$

**Table B.2:** *List of adopted levels for  $^{12}\text{C}$  [NuDat].*

$E(\text{level})$ (keV)	$J^\pi$	$\Gamma(\text{level})$	$E(\gamma)$ (keV)	$I(\gamma)$	$M(\gamma)$	Final level $J^\pi(\text{keV})$
0	$0^+$	STABLE				
4439.82 21	$2^+$	$10.8 \cdot 10^{-3} \text{ eV}$ 6 IT=100	4438.94	100	E2	$0^+(0)$
7654.07 19	$0^+$	9.3 eV 9 IT=4.16E-2 $\alpha \approx 100$	3213.79	100	E2	$2^+(4439.82)$
9641 5	$3^-$	46 keV 3 $\alpha \approx 100$ IT < 4.1E-5	9637	100	E3	$0^+(0)$
9870 60	$2^+$	850 keV 85 IT $\approx 7.1\text{E-}6$ $\alpha \approx 100$				

**Table B.3:** List of adopted levels for  $^{16}\text{O}$  [NuDat].

$E(\text{level})$ [keV]	$J^\pi$	$\Gamma(\text{level})$	$E(\gamma)$ [keV]	$I(\gamma)$	$M(\gamma)$	Final level $J^\pi$ [keV]
0	$0^+$	STABLE				
6049.4 10	$0^+$	67 ps 5	6048.21		[E0]	$0^+(0)$
6129.89 4	$3^-$	18.4 ps 5	6128.634	100	[E3]	$0^+(0)$
6917.1 6	$2^+$	4.70 fs 13	787.26	$\leq 0.008$	[E1]	$3^-(6129.89)$
			867.712	0.027 3	[E2]	$0^+(6049.4)$
			6915.56	100	[E2]	$0^+(0)$
7116.85 14	$1^-$	8.3 fs 5	986.93 15	0.07014	[E2]	$3^-(6129.89)$
			1067.5 10	$< 6\text{E-}4$	[E1]	$0^+(6049.4)$
			7115.15 14	100	[E1]	$0^+(0)$
8871.9 5	$2^-$	125 fs 11	1754.9 6	14.77	[M1+E2]	$1^-(7116.85)$
			1954.7 8	4.6 7	[E1]	$2^+(6917.1)$
			2741.5 5	100 21	[M1+E2]	$3^-(6129.89)$
			2822.2 12	0.15 5	[M2]	$0^+(6049.4)$
			8869.3 5	9.3 10	[M2]	$0^+(0)$
9585 11	$1^-$	420 keV 20 IT = $6.7\text{E-}6$ 10 $\alpha = 100$	2688 11	12 4	[E1]	$2^+(6917.1)$
			9582 11	100 16	[E1]	$0^+(0)$
9844.5 5	$2^+$	0.62 keV 10 IT = 0.0016 3 $\alpha = 100$	2927.1 8	34 7	[M1]	$2^+(6917.1)$
			3794.6 12	30 7	[E2]	$0^+(6049.4)$
			9841.2 5	100 7	[E2]	$0^+(0)$
10356 3	$4^+$	26 keV 3 IT = $2.4\text{E-}4$ 4 $\alpha = 100$	3439 3	100 10	[E2]	$2^+(6917.1)$
			4225 3	$< 1.6$	[E1]	$3^-(6129.89)$
			10352 3	$9\text{E-}5$ 3	[E4]	$0^+(0)$
10957 1	$0^-$	5.5 fs 35	3839.6 10	100	[M1]	$1^-(7116.85)$
11080 3	$3^+$	$< 12$ keV				
11096.7 16	$4^+$	0.28 keV 5 IT = 0.0020 6 $\alpha = 100$	4179.0 17	81 20	[E2]	$2^+(6917.1)$
			4966.0 16	100 42	[E1]	$3^-(6129.89)$
11260?	$(0^+)$	2500 keV $\alpha = 100$				
11520 4	$2^+$	71 keV 3 IT = $9.4\text{E-}5$ 3 $\alpha = 100$	4402 4	$\leq 0.9$	[E1]	$1^-(7116.85)$
			4602 4	4.4 11	[M1]	$2^+(6917.1)$
			5470 5	4.6 8	[E2]	$0^+(6049.4)$
11516 4			11516 4	100.0 13	[E2]	$0^+(0)$
11600 20	$3^-$	800 keV 100 $\alpha = 100$				
12049 2	$0^+$	1.5 keV 5 IT = ? $\alpha = 100$	12044.1 20		[E0]	$0^+(0)$
12440 2	$1^-$	91 keV 6 IT = 0.0132 24 p = 0.9 1 $\alpha = 99.1$ 1	6389.2 23	1.2 4	[E1]	$0^+(6049.4)$
			12434.8 20	100	[E1]	$0^+(0)$
12530 1	$2^-$	0.111 keV 10 IT = 3.2 3 p = 14 7 $\alpha = 83$ 3	3657.7 12	67 4	[M1]	$2^-(8871.9)$
			5412.1 10	24.5 14	[M1]	$1^-(7116.85)$
			5611.8 12	$< 2$	[E1]	$2^+(6917.1)$
			6398.7 10	100 4	[M1]	$3^-(6129.89)$
			12524.7 10	12.2 12	[M2]	$0^+(0)$
12796 4	$0^-$	40 keV 4 IT = 0.0062 8 p = 100	5678 4	100	[M1]	$1^-(7116.85)$
12948.6 4	$2^-$	1.34 keV 4 IT = 0.28 3 p = 78 4 $\alpha = 22$ 4	4096.1 7	84 4	[M1]	$2^-(8871.9)$
			5850.7 5	12 2	[M1]	$1^-(7116.85)$
			6837.1 4	100 4	[M1]	$3^-(6129.89)$
			12963.0 4	4.2 8	[M2]	$0^+(0)$
13020 10	$2^+$	150 keV 10 IT = ? p = ? $\alpha = ?$	13014 10	100		$0^+(0)$

13090 8	1 <sup>-</sup>	130 keV 5 IT = 0.026 4 p = 71 α = 29	5972 8 7039 8 13084 8	3.1 8 0.58 12 100	[M1] [E1] [E1]	1 <sup>-</sup> (7116.85) 0 <sup>+</sup> (6049.4) 0 <sup>+</sup> (0)
13129 10	3 <sup>-</sup>	110 keV 30 IT = ? p = 1 α = 99				
13259 2	3 <sup>-</sup>	21 keV 1 IT = ? p = ? α = ?				
13664 3	1 <sup>+</sup>	64 keV 3 p = 14 α = 86 IT < 0.0015				
13869 2	4 <sup>+</sup>	89 keV 2 IT = ? p = 0.6 α = 99.4				

**Table B.4:** List of adopted levels for <sup>20</sup>Ne [NuDat].

$E(\text{level})$ (keV)	$J^\pi$	$\Gamma(\text{level})$	$E(\gamma)$ (keV)	$I(\gamma)$	$M(\gamma)$	Final level $J^\pi(\text{keV})$
0	0 <sup>+</sup>	STABLE				
1633.674 15	2 <sup>+</sup>	0.73 ps 4	1633.602 15	100	[E2]	0 <sup>+</sup> (0)
4247.7 11	4 <sup>+</sup>	64 fs 6	2613.8 11	100	[E2]	2 <sup>+</sup> (1633.674)
4966.51 20	2 <sup>-</sup>	3.3 ps 4	3332.54 20 4965.85 20	99.4 2 0.6 2	[E1+M2+E3] [M2]	2 <sup>+</sup> (1633.674) 0 <sup>+</sup> (0)
5621.4 17	3 <sup>-</sup>	139 fs 35 IT = 7 3 α = 93 3	654.9 18 3987.3 17 5620.6 17	4.8 16 87.6 10 7.6 10	[M1] [E1] [E3]	2 <sup>-</sup> (4966.51) 2 <sup>+</sup> (1633.674) 0 <sup>+</sup> (0)
5787.7 26	1 <sup>-</sup>	0.028 keV 3 IT = 0.016 3 α = 100	4154 3 5787 3	82 5 18 5	[E1] [E1]	2 <sup>+</sup> (1633.674) 0 <sup>+</sup> (0)
6706 47						
6725 5	0 <sup>+</sup>	19.0 keV 9 IT = 1.7E-4 α = 100	5090 5 6724 5	100	[E2] [E0]	2 <sup>+</sup> (1633.674) 0 <sup>+</sup> (0)
7004.0 36	4 <sup>-</sup>	305 fs 62	1383 4 2037 4 2756 4 5369 4	25 11 63.5 0.5 2	[M1] [E2] [E1] [M2]	3 <sup>-</sup> (5621.4) 2 <sup>-</sup> (4966.51) 4 <sup>+</sup> (4247.7) 2 <sup>+</sup> (1633.674)

**Table B.5:** *The atomic masses are from [Wan+17]. The nuclear masses marked with \* are from [Hua+17]. The other nuclear masses are calculated with the formula  $M_N(A, Z) = M_A(A, Z) - Z \cdot m_e + B_e(Z)$ , with electron mass  $m_e$  and total electron binding energy  $B_e(Z)$ . For  $Z = 1$  the binding energy is  $B_e(Z) = 13.6 \text{ eV}$  and for  $Z \geq 2$  the approximated formula  $B_e(Z) = [14.4381 \cdot Z^{2.39} + 1.55468 \cdot 10^{-6} \cdot Z^{5.35}] \text{ eV}$  was used. This formula provides an rms error of 150 eV over the range up to  $Z = 106$  [LPT03, Appendix A] for the listed binding energies in [Hua+76]. In comparison to the listed binding energies we assumed an error for  $B_e$  of 2 eV for  $Z \leq 4$ , 20 eV for  $Z = 6$  and, 40 eV for  $Z \geq 7$ , to calculate the error of the nuclear masses.*

N	Z	A	Element	Atomic Mass $\mu u$	Nuclear Mass $\mu u$	Nuclear Mass MeV/ $c^2$
0	1	1	H	1 007 825.032 241(94)	1 007 276.466 93(9)*	938.272 081 38(8)
1	1	1	D	2 014 101.778 114(122)	2 013 553.213 704(247)	1875.612 929(11)
1	2	4	He	3 016 029.322 645(220)	3 014 932.244 071(307)	2808.391 583(17)
2	2	4	He	4 002 603.254 130(63)	4 001 506.179 127(60)*	3727.379 378(23)
4	4	8	Be	8 005 305.10(4)	8 003 111.206(40)	7454.850 833(59)
6	6	12	C	12 000 000	11 996 709.642(21)	11 174.864 196(71)
7	7	14	N	14 003 074.004 460(207)	13 999 235.567(21)	13 040.205 271(82)
8	8	16	O	15 994 914.619 598(173)	15 990 528.212(43)	14 895.082 612(100)
10	10	20	Ne	19 992 440.1762(17)	19 986 958.182(43)	18 617.733 531(121)
12	12	24	Mg	23 985 041.697(14)	23 978 464.621(45)	22 335.798 211(143)

**Table B.6:** *Calculated Gamow peaks for the reactions of the helium burning stage. Due to the high density and temperature within the core of a star a plasma is formed. The energy  $kT$  of around 10–20 keV is significantly higher than the binding energy of the electrons so that the plasma consists of ions and free electrons. Thus, in the calculation of the Gamow peak, nuclear masses are used instead of atomic masses. The Coulomb penetration factor for each reaction at the Gamow peak is calculated using Eq. (2.12).*

Reaction	T GK	$E_G$ keV	$\Delta E_G$ keV	Coulomb penetration factor
${}^4\text{He}(\alpha, \gamma){}^8\text{Be}$	0.12	94.292	72.113	$1.202 \times 10^{-8}$
${}^4\text{He}(\alpha, \gamma){}^8\text{Be}$	0.2	132.549	110.380	$2.088 \times 10^{-7}$
${}^8\text{Be}(\alpha, \gamma){}^{12}\text{C}$	0.2	231.585	145.901	$2.131 \times 10^{-12}$
${}^{12}\text{C}(\alpha, \gamma){}^{16}\text{O}$	0.2	315.595	170.320	$1.244 \times 10^{-16}$
${}^{16}\text{O}(\alpha, \gamma){}^{20}\text{Ne}$	0.2	390.626	189.488	$2.057 \times 10^{-20}$
${}^{20}\text{Ne}(\alpha, \gamma){}^{24}\text{Mg}$	0.2	459.495	205.514	$6.958 \times 10^{-24}$

**Table B.7:** *Calculated Coulomb penetration factors Eq. (2.12) of the reactions of the helium burning stage for a fixed cms energy  $E_{cms} = 300$  keV. Due to the high density and temperature within the core of a star a plasma is formed. The energy  $kT$  of around 10–20 keV is significantly higher than the binding energy of the electrons so that the plasma consists of ions and free electrons. Thus, in the calculations, nuclear masses are used instead of atomic masses.*

Reaction	$E_{cms}$ keV	Coulomb factor
${}^4\text{He}(\alpha, \gamma){}^8\text{Be}$	300	$3.628 \times 10^{-5}$
${}^8\text{Be}(\alpha, \gamma){}^{12}\text{C}$	300	$5.565 \times 10^{-11}$
${}^{12}\text{C}(\alpha, \gamma){}^{16}\text{O}$	300	$4.859 \times 10^{-17}$
${}^{16}\text{O}(\alpha, \gamma){}^{20}\text{Ne}$	300	$3.433 \times 10^{-23}$
${}^{20}\text{Ne}(\alpha, \gamma){}^{24}\text{Mg}$	300	$2.189 \times 10^{-29}$

**Table B.8:** *Physical constants used in this work. The quantities marked with \* are from [Hua+17], the others are from [Zyl+20].*

Quantity	Symbol (equation)	Value
speed of light in vacuum	$c$	$299\,792\,458\text{ m s}^{-1}$
reduced Planck constant	$\hbar$	$6.582\,119\,569 \times 10^{-22}\text{ MeV s}$
electron charge magnitude	$e$	$1.602\,176\,634 \times 10^{-19}\text{ C}$
Boltzmann constant	$k$	$1.380\,649 \times 10^{-23}\text{ J K}^{-1}$
conversion constant	$\hbar c$	$197.326\,980\,4\text{ MeV fm}$
atomic mass unit*	$u=(\text{mass } {}^{12}\text{C atom})/12$	$931.494\,095\,4(57)\text{ MeV}/c^2$
electron mass*	$m_e$	$548\,579.909\,070(16)\text{ nu}$ $=0.510\,998\,946\,1(31)\text{ MeV}/c^2$
proton mass*	$M_p$	$1\,007\,276\,466.93(9)\text{ nu}$ $=938.272\,081\,38(8)\text{ MeV}/c^2$
alpha particle mass*	$M_\alpha$	$4\,001\,506\,179.127(60)\text{ nu}$ $=3727.379\,378(23)\text{ MeV}/c^2$
fine-structure constant	$\alpha_{em} = e^2/4\pi\epsilon_0\hbar c$	$7.297\,352\,569\,3(11) \times 10^{-3}$ $=1/137.035\,999\,084(21)$

**Table B.9:** *Coordinate system for the target and generator system. The angles are defined analogously to the spherical coordinate system.*

Coordinate	Definition
$x$	up
$y$	right with the beam direction $y = z \times x$
$z$	beam direction
Origin	Target middle point $(0, 0, 0)$
$\theta$	Angle from the beamline within in the range $[0, \pi)$
$\phi$	Rotation angle in $x - y$ plane in range $[-\pi, \pi)$

**Table B.10:** *Base units used in the simulation software*

Quantity	Unit
Length	mm
Cross Section	$\mu\text{b} = 10^{-30} \text{cm}^2$
Angle	rad
Energy	MeV
Momentum	$\text{MeV } c^{-1}$
Mass	$\text{MeV } c^{-2}$
Charge	$e$
Beam Current	$\mu\text{A}$
Time	ns
Temperature	K

**Table B.11:** *To extrapolate the  $S$ -factor different step length are used for different energy ranges.*

$E_{\min}^{\text{cms}}[\text{MeV}]$	$E_{\max}^{\text{cms}}[\text{MeV}]$	$dE[\text{MeV}]$
0.01	6	0.001
5.5	14	0.005
12	20	0.01



---

---

# APPENDIX C

---

## EPICS Support Files

In this chapter the installation scripts of EPICS, MXSlowControl and the modules are presented. In addition different EPICS related C++ header and source files are displayed.

### Code C.1: Installation Script EPICS

File: *setup-epics-ioc-raspbian.sh*

```
1  #!/bin/bash
2  # author: Stefan Lunkenheimer
   ↳ (stelunke@uni-mainz.de)
3  # Basic file on: https://gist.github.com/
   ↳ inigoalonso/a77314206b516e94d4aa
4
5  epicsDir=/opt/epics
6  gitDir="$(cd "$(dirname "$0")/.." && pwd )"
7  epicsVersion=3.14.12.6
8  asynVersion=asyn4-31
9  streamVersion=StreamDevice-2
10 streamSubVersion=6
11 autosaveVersion=5-7-1
12 DOWNLOADS=${gitDir}/install/external
13
14 pyepics=true
15 frontend=true
16
17 export LC_ALL=en_US.UTF-8
18 export LC_CTYPE=en_US.UTF-8
19 export LANG=en_US.UTF-8
20
21
22 # Download the necessary source files
23 echo "Downloading the necessary source files"
24
25 cd ${DOWNLOADS}
26 #wget http://www.aps.anl.gov/epics/download/
   ↳ base/baseR${epicsVersion}.tar.gz ||
   ↳ exit
27 #wget http://www.aps.anl.gov/epics/download/
   ↳ modules/${asynVersion}.tar.gz ||
   ↳ exit
28 #wget http://epics.web.psi.ch/software/
   ↳ streamdevice/${streamVersion}.tgz ||
   ↳ exit
29 #wget https://github.com/epics-modules/
   ↳ autosave/archive/
   ↳ R${autosaveVersion}.tar.gz -O
   ↳ autosave-R${autosaveVersion}.tar.gz ||
   ↳ exit
30
31
32 # Preliminary installations!!!
33 sudo apt-get install libreadline7
   ↳ libreadline-dev
34
35 # -----
36 # ----- INSTALL EPICS BASE -----
37 # -----
38
39 # Create a place to work
40 echo "Creating a place to work: ${epicsDir}"
41 sudo mkdir -p ${epicsDir}
42 sudo chown ${USER}:${GROUP} ${epicsDir}
43
44 echo "Untar-ing EPICS base"
45 tar -zxf
   ↳ ${DOWNLOADS}/baseR${epicsVersion}.tar.gz
   ↳ -C ${epicsDir}/ || exit
46
47 echo "Creating links to epics base folder"
48 ln -s ${epicsDir}/base-${epicsVersion}
   ↳ ${epicsDir}/base
49
50 # Set the EPICS environment variables --
   ↳ Create thisEPICS.sh
51 echo "Setting up the EPICS enviroment
   ↳ variables"
52 cat << EOF > ${epicsDir}/thisEPICS.sh
53 #EPICS
54 export EPICS_ROOT=${epicsDir}
55 export EPICS_BASE=\${EPICS_ROOT}/base
56 export EPICS_HOST_ARCH=\${EPICS_BASE}/
   ↳ startup/EpicsHostArch
```

```

57 export EPICS_BASE_BIN=${EPICS_BASE}/bin/
58 ↪ \${EPICS_HOST_ARCH}
59 export EPICS_BASE_LIB=${EPICS_BASE}/lib/
60 ↪ \${EPICS_HOST_ARCH}
61 if [ "" = "\${LD_LIBRARY_PATH}" ]; then
62     export LD_LIBRARY_PATH=${EPICS_BASE_LIB}
63 else
64     export LD_LIBRARY_PATH=
65 ↪ \${EPICS_BASE_LIB}:\${LD_LIBRARY_PATH}
66 fi
67 export PATH=${PATH}:\${EPICS_BASE_BIN}
68
69 export HOST=$(hostname)
70 EOF
71
72 chmod +x ${epicsDir}/thisEPICS.sh
73
74 # Set environment Variables for the
75 ↪ installation
76 source ${epicsDir}/thisEPICS.sh
77
78 # Configure RELEASE
79 echo "EPICS_BASE=${EPICS_ROOT}/base" >
80 ↪ ${EPICS_ROOT}/RELEASE.local
81
82 # Compile EPICS
83 echo "Compiling EPICS"
84 cd ${EPICS_BASE}
85 make -j4 || exit
86
87 # ----- INSTALL MODULES -----
88
89 # ----- Install ASYN -----
90 echo "Installing ASYN library into EPICS"
91 mkdir -p ${EPICS_ROOT}/modules
92 echo "Untar-ing"
93 tar -zxvf ${DOWNLOADS}/${asynVersion}.tar.gz
94 ↪ -C ${EPICS_ROOT}/modules/ || exit
95 echo "Linking"
96 ln -s ${EPICS_ROOT}/modules/${asynVersion}
97 ↪ ${EPICS_ROOT}/modules/asyn
98
99 # MODIFY RELEASE
100 echo "Editing the configuration"
101 cd ${EPICS_ROOT}/modules/asyn
102 sed -ie '/IPAC/ s/^##/ #' configure/RELEASE
103 sed -ie '/SNCSEQ/ s/^##/ #' configure/RELEASE
104 sed -ie '/EPICS_BASE/ s/^##/ #'
105 ↪ configure/RELEASE
106 cat << EOF >> configure/RELEASE
107
108 # These allow developers to override the
109 ↪ RELEASE variable settings
110 # without having to modify the
111 ↪ configure/RELEASE file itself.
112 #-include \$(TOP)/../RELEASE.local
113 #-include \$(TOP)/configure/RELEASE.local
114 EPICS_BASE=${EPICS_BASE}
115 EOF
116
117 # Compile ASYN
118 echo "Compiling"
119 cd ${EPICS_ROOT}/modules/asyn
120 make -j4 || exit
121
122 # Modify thisEPICS.sh
123 echo -e "\n#Modules" >>
124 ↪ ${EPICS_ROOT}/thisEPICS.sh
125 echo "export
126 ↪ ASYN=${EPICS_ROOT}/modules/asyn" >>
127 ↪ ${EPICS_ROOT}/thisEPICS.sh
128 export ASYN=${EPICS_ROOT}/modules/asyn
129
130 # Modify RELEASE.local
131
132 echo "ASYN=${EPICS_ROOT}/modules/asyn" >>
133 ↪ ${EPICS_ROOT}/RELEASE.local
134 # -----
135 # ----- Install StreamDevice -----
136 echo "Installing StreamDevice library into
137 ↪ EPICS"
138 mkdir -p ${EPICS_ROOT}/modules/stream
139 cd ${EPICS_ROOT}/modules/stream
140 echo "Untar-ing"
141 tar -zxvf ${DOWNLOADS}/${streamVersion}.tgz
142 ↪ -C ${EPICS_ROOT}/modules/stream || exit
143
144 echo "Configuring"
145 echo | makeBaseApp.pl -t support
146 echo "ASYN=${EPICS_ROOT}/modules/asyn" >>
147 ↪ configure/RELEASE.local
148 echo "Compiling 1"
149 make -j4 || exit
150
151 cd ${streamVersion}-${streamSubVersion} ||
152 ↪ exit
153 #
154 # ----- Insert PROFIBUS BUS API to
155 ↪ StreamDevice -----
156 echo "Copy Profibus src-files"
157 cp ${gitDir}/install/src/* src/ || exit
158 echo "Add Profibus to config_stream"
159 cat << EOF >> src/CONFIG_STREAM
160
161 #PROFIBUS
162 BUSSES += Profibus
163 CPPFLAGS += -std=c++11
164 CXXFLAGS += -std=c++11
165 EOF
166 echo "Compiling 2"
167 make -j4 || exit
168
169 # Modify thisEPICS.sh
170 echo "export
171 ↪ STREAM=${EPICS_ROOT}/modules/stream" >>
172 ↪ ${EPICS_ROOT}/thisEPICS.sh
173 export STREAM=${EPICS_ROOT}/modules/stream
174
175 # Modify RELEASE.local
176 echo "STREAM=${EPICS_ROOT}/modules/stream" >>
177 ↪ ${EPICS_ROOT}/RELEASE.local
178 # -----
179 # ----- Install AUTOSAVE -----
180 echo "Installing Autosave Module into EPICS"
181 echo "Untar-ing"
182 tar -zxvf ${DOWNLOADS}/autosave-
183 ↪ R${autosaveVersion}.tar.gz -C
184 ↪ ${EPICS_ROOT}/modules/ || exit
185 echo "Linking"
186 ln -s ${EPICS_ROOT}/modules/autosave-
187 ↪ R${autosaveVersion}
188 ↪ ${EPICS_ROOT}/modules/autosave
189 cd ${EPICS_ROOT}/modules/autosave
190
191 # MODIFY RELEASE
192 echo "Editing the configuration"
193 sed -ie '/EPICS_BASE/ s/^##/ #'
194 ↪ configure/RELEASE
195 cat << EOF >> configure/RELEASE
196
197 # These allow developers to override the
198 ↪ RELEASE variable settings
199 # without having to modify the
200 ↪ configure/RELEASE file itself.
201 #-include \$(TOP)/../RELEASE.local
202 #-include \$(TOP)/configure/RELEASE.local
203 EPICS_BASE=${EPICS_ROOT}/base
204 EOF
205
206 echo "Compiling"

```

```

180 cd ${EPICS_ROOT}/modules/autosave
181 make -j4 || exit
182
183 # Modify thisEPICS.sh
184 echo "export
↳ AUTOSAVE=${EPICS_ROOT}/modules/autosave"
↳ >> ${EPICS_ROOT}/thisEPICS.sh
185 export
↳ AUTOSAVE=${EPICS_ROOT}/modules/autosave
186
187 # Modify RELEASE.local
188 echo
↳ "AUTOSAVE=${EPICS_ROOT}/modules/autosave"
↳ >> ${EPICS_ROOT}/RELEASE.local
189 # -----
190
191 # ----- Install dev-gpio -----
192 # Testing on RaspberryPi
193 if [ $EPICS_HOST_ARCH = "linux-arm" ]
194 then
195     gpioVersion=master
196     cd ${DOWNLOADS}
197     #wget -O dev-gpio_${gpioVersion}.zip
↳ https://github.com/ffeldbauer/epics-
↳ devgpio/archive/${gpioVersion}.zip ||
↳ exit
198 unzip ${DOWNLOADS}/dev-
↳ gpio_${gpioVersion}.zip -d
↳ ${EPICS_ROOT}/modules || exit
199 ln -s ${EPICS_ROOT}/modules/epics-
↳ devgpio-${gpioVersion}
↳ ${EPICS_ROOT}/modules/devgpio || exit
200 cd ${EPICS_ROOT}/modules/devgpio
201
202 # MODIFY RELEASE
203 echo "Editing the configuration"
204 sed -ie '/EPICS_BASE/ s/^#*#/'
↳ configure/RELEASE
205 echo "EPICS_BASE=${EPICS_BASE}" >>
↳ configure/RELEASE
206
207 make -j4 || exit
208
209 # Modify thisEPICS.sh
210 echo "export DEVPPIO=${EPICS_ROOT}/
↳ modules/devgpio" >>
↳ ${EPICS_ROOT}/thisEPICS.sh
211 export
↳ DEVPPIO=${EPICS_ROOT}/modules/devgpio
212
213 # Modify RELEASE.local
214 echo "DEVPPIO=${EPICS_ROOT}/modules/
↳ devgpio" >>
↳ ${EPICS_ROOT}/RELEASE.local
215 fi
216 # -----
217
218 # ----- Install pyEPICS -----
219 if [ $pyepics = true ] ;
220 then
221     sudo apt-get install python-pip ||
222     ( echo -e
↳ "\n\n-----"
↳ \nInstalling python-pip failed"
↳ && exit )
223     sudo pip install pyepics ||
224     ( echo -e
↳ "\n\n-----"
↳ \nInstalling pyepics failed" &&
↳ exit )
225
226 # Modify thisEPICS.sh
227 echo -e "\n#PYEPICS" >>
↳ ${EPICS_ROOT}/thisEPICS.sh
228 echo "export PYEPICS_LIBCA=
↳ \${EPICS_BASE_LIB}/libca.so" >>
↳ ${EPICS_ROOT}/thisEPICS.sh
229 fi
230 # -----
231
232 # ----- Frontend Module -----
233 if [ $frontend = true ] ;
234 then
235     ${gitDir}/modules/frontend/
↳ install_module_frontend.sh
236 fi
237 # -----
238
239 # ----- Install Environment -----
240 sudo install ${EPICS_ROOT}/thisEPICS.sh
↳ /etc/profile.d/epics.sh
241
242 echo "Done!"

```

## Code C.2: Installation Script MXSlowControl

File: install\_MXSlowControl.sh

```

1 #!/bin/bash
2 # author: Stefan Lunkenheimer
↳ (stelunke@uni-mainz.de)
3
4 # Remark: Check if the export variables are
↳ set correctly!
5 source /etc/profile.d/epics.sh
6
7 iocNAME="MXSlowControl"
8
9 gitDir="$(cd "$(dirname "$0")/.." && pwd )"
10
11 echo "Install $iocName"
12
13 cd ${EPICS_ROOT}
14 echo "Generate Base Application"
15 mkdir -p ${iocNAME} || exit
16 cd ${iocNAME}
17 $ASYN/bin/${EPICS_HOST_ARCH}/makeSupport.pl -t
↳ streamSCPI ${iocNAME} || exit
18 rm -rf configure
19 ${EPICS_BASE}/bin/${EPICS_HOST_ARCH}/
↳ makeBaseApp.pl -t ioc
↳ ${iocNAME}
20 echo | ${EPICS_BASE}/bin/${EPICS_HOST_ARCH}/
↳ makeBaseApp.pl -t ioc -i
↳ ${iocNAME}
21
22 cp ${gitDir}/MXSlowControl/CONFIG_SITE.local
↳ ${EPICS_ROOT}/ || exit
23
24 echo "Add modules to Base Application"
25 # Add streamDevice and Asyn to the ioc
26 sed -e "/${iocNAME}_DBD += xxx.dbd/ a
↳ ${iocNAME}_DBD += stream.dbd asyn.dbd
↳ drvAsynSerialPort.dbd drvAsynIPPort.dbd
↳ asSupport.dbd" \
27 -e "/${iocNAME}_LIBS += xxx/ a
↳ ${iocNAME}_LIBS += stream asyn
↳ autosave" \
28 -i ${iocNAME}App/src/Makefile
29
30 # Add dev-gpio to the ioc
31 if [ $EPICS_HOST_ARCH = "linux-arm" ]
32 then
33 sed -e "/${iocNAME}_DBD += xxx.dbd/ a
↳ ${iocNAME}_DBD += devgpio.dbd" \

```

```

34     -e "${iocNAME}_LIBS += xxx/ a
    ↪  ${iocNAME}_LIBS += devgpio" \
35     -i ${iocNAME}App/src/Makefile
36 fi
37
38 # Add Device Support components
39 echo "Add Device Support Drivers"
40 # Install necessary libraries
41 sudo apt-get install libusb-dev
42 cp -r ${gitDir}/MXSlowControl/App/src/*
    ↪  ${iocNAME}App/src
43
44 # Add USBADC and USBDAC to the ioc Makefile
45 sed -e "7i LIBRARY_IOC += dbLoadMultiRecords ]
    ↪  \ndbLoadMultiRecords_SRCS +=
    ↪  dbLoadMultiRecords.cpp ]
    ↪  \ndbLoadMultiRecords_LIBS +=
    ↪  \$(EPICS_BASE_IOC_LIBS)\n" \
46     -e "${iocNAME}_DBD += xxx.dbd/ a
    ↪  ${iocNAME}_DBD +=
    ↪  dbLoadMultiRecords.dbd" \
47     -e "${iocNAME}_LIBS += xxx/ a
    ↪  ${iocNAME}_LIBS +=
    ↪  dbLoadMultiRecords" \
48     -i ${iocNAME}App/src/Makefile
49
50 # Add USBADC and USBDAC to the ioc Makefile
51 sed -e "7i LIBRARY_IOC +=
    ↪  aoUSBADC\naoUSBADC_SRCS += devUSBADC.c
    ↪  USBADC.cpp\naoUSBADC_LIBS +=
    ↪  \$(EPICS_BASE_IOC_LIBS)\n" \
52     -e "7i LIBRARY_IOC +=
    ↪  aiUSBADC\naiUSBADC_SRCS +=
    ↪  devUSBADC.c USBADC.cpp\naiUSBADC_LIBS
    ↪  += \$(EPICS_BASE_IOC_LIBS)\n" \
53     -e "7i USR_CXXFLAGS += -std=c++11\n" \
54     -e "${iocNAME}_DBD += xxx.dbd/ a
    ↪  ${iocNAME}_DBD += aiUSBADC.dbd
    ↪  aoUSBADC.dbd" \
55     -e "${iocNAME}_LIBS += xxx/ a
    ↪  ${iocNAME}_LIBS += aiUSBADC
    ↪  aoUSBADC\n${iocNAME}_SYS_LIBS += usb"
    ↪  \
56     -i ${iocNAME}App/src/Makefile
57
58 # add caArchiver
59 sed -i "7i PROD_HOST +=
    ↪  caArchiver\ncaArchiver_SRCS +=
    ↪  caArchiver.c\ncaArchiver_LIBS +=
    ↪  \$(EPICS_BASE_HOST_LIBS)\n"
    ↪  ${iocNAME}App/src/Makefile
60 mkdir ${EPICS_ROOT}/log
61 touch ${EPICS_ROOT}/CA_ADDR_LIST.local
62 touch ${EPICS_ROOT}/log/log.txt
63
64 # module Frontend
65 if [ "$(grep FRONTEND
    ↪  ${EPICS_ROOT}/RELEASE.local)" != "" ]
66 then
67     sed -e "${iocNAME}_DBD += xxx.dbd/ a
    ↪  ${iocNAME}_DBD += frontend.dbd" \
68     -e "${iocNAME}_LIBS += xxx/ a
    ↪  ${iocNAME}_LIBS += frontend" \
69     -i ${iocNAME}App/src/Makefile
70 fi
71
72 # BUILD ALL
73 echo "Build Application"
74 make -j4 || exit
75
76 echo "Transfer Basic db and st.cmd"
77 rm -f db/*
78 cp -r ${gitDir}/MXSlowControl/db/* db/ ||
    ↪  exit
79 cp -r ${gitDir}/MXSlowControl/iocBoot/ ]
    ↪  iocMXSlowControl/* iocBoot/ioc${iocNAME}/
    ↪  || exit
80 sed -e s/EPICS_HOST_ARCH/${EPICS_HOST_ARCH}/
    ↪  \
81     -e s/YOUR_HOSTNAME/${hostname}/ \
82     -i iocBoot/ioc${iocNAME}/st.cmd
83 chmod +x iocBoot/ioc${iocNAME}/st.cmd
84
85 echo "Transfer autosave example"
86 mkdir -p autosave || exit
87 cp -r ${gitDir}/MXSlowControl/autosave/*
    ↪  autosave/ || exit
88
89 echo "Transfer udev/rules"
90 sudo cp ${gitDir}/MXSlowControl/rules/999- ]
    ↪  mxslowcontrol.rules /etc/udev/rules.d/ ||
    ↪  exit
91
92 # INSTALLING service
93 sudo apt-get install tmux
94 sudo install ${gitDir}/MXSlowControl/ ]
    ↪  systemctl/epics.service
    ↪  /etc/systemd/system/epics.service
95 sudo sed -i 's/\${IOC}/$(echo
    ↪  ${EPICS_ROOT}/MXSlowControl/iocBoot/ ]
    ↪  iocMXSlowControl|sed 's/\/\/\/\/g')/'
    ↪  /etc/systemd/system/epics.service
96 sudo install ${gitDir}/MXSlowControl/ ]
    ↪  systemctl/caArchiver.service
    ↪  /etc/systemd/system/caArchiver.service
97 sudo systemctl daemon-reload
98
99 # update environments
100 cat << EOF >> ${EPICS_ROOT}/thisEPICS.sh
101
102 #MXSlowControl
103 export MXSLOWCONTROL=${EPICS_ROOT}/
    ↪  MXSlowControl
104 export MXSLOWCONTROL_BIN=${MXSLOWCONTROL}/
    ↪  bin/\${EPICS_HOST_ARCH}
105 export MXSLOWCONTROL_LIB=${MXSLOWCONTROL}/
    ↪  lib/\${EPICS_HOST_ARCH}
106 export IOC=${MXSLOWCONTROL}/iocBoot/ ]
    ↪  iocMXSlowControl
107 if [ "" = "\${LD_LIBRARY_PATH}" ]; then
108     export
    ↪  LD_LIBRARY_PATH=${MXSLOWCONTROL_LIB}
109 else
110     export LD_LIBRARY_PATH=${MXSLOWCONTROL ]
    ↪  _LIB}:\${LD_LIBRARY_PATH}
111 fi
112 export PATH=${PATH}:\${MXSLOWCONTROL_BIN}
113
114 #Channel Access List
115 export EPICS_CA_ADDR_LIST="\${<\${EPICS ]
    ↪  _ROOT}/CA_ADDR_LIST.local)"
116
117 #caArchiver
118 export CA_ARCHIVER_LOG=${EPICS_ROOT}/log
119 EOF
120 sudo install ${EPICS_ROOT}/thisEPICS.sh
    ↪  /etc/profile.d/epics.sh
121
122 echo "DONE!"

```

## Code C.3: Installation Script Module MXFrontend

File: `install_module_frontend.sh`

```

1  #!/bin/bash
2  # author: Stefan Lunkenheimer
   ↪ (stelunke@uni-mainz.de)
3
4  # Remark: Check if the export variables are
   ↪ set correctly!
5  RED='\033[1;31m'
6  HINT='\033[1;36m'
7  NC='\033[0m' # No Color
8
9  #source /etc/profile.d/epics.sh
10 moduleName="frontend"
11 modulePath="FRONTEND"
12
13 SOURCE_DIR="$(cd "$(dirname "$0")" && pwd
   ↪ )"
14
15 cd $EPICS_ROOT/modules || exit
16 mkdir -p $moduleName || exit
17
18 # Make Base Application Support Module
19 cd $moduleName
20 echo $moduleName | makeBaseApp.pl -t support
   ↪ || (echo "${RED}Error makeBaseApp" &&
   ↪ exit)
21
22 # Copy source files
23 cp $SOURCE_DIR/src/*.cpp
   ↪ ${moduleName}App/src/. || (echo
   ↪ "${RED}Error while cp cpp" && exit)
24 cp $SOURCE_DIR/src/*.h ${moduleName}App/src/.
   ↪ || (echo "${RED}Error while cp h" &&
   ↪ exit)
25 cp $SOURCE_DIR/src/*.dbd
   ↪ ${moduleName}App/src/. || (echo
   ↪ "${RED}Error while cp dbd" && exit)
26
27 # Configure Makefile
28 sed -e "/${moduleName}_LIBS +=/r
   ↪ $SOURCE_DIR/src/Makefile_ext" \
29 -i ${moduleName}App/src/Makefile || (echo
   ↪ "${RED}Error sed Makefile" && exit)
30
31 # Make all
32 make -j4 || (echo "${RED}Error while make" &&
   ↪ exit)
33
34 # Modify thisEPICS.sh
35 echo "export $moduleName=${EPICS_ROOT}/
   ↪ modules/$moduleName" >>
   ↪ ${EPICS_ROOT}/thisEPICS.sh
36 export $moduleName=${EPICS_ROOT}/modules/
   ↪ $moduleName
37
38 # Modify RELEASE.local
39 echo "$moduleName=${EPICS_ROOT}/modules/
   ↪ $moduleName" >>
   ↪ ${EPICS_ROOT}/RELEASE.local
40
41 # Done
42 echo "${HINT}\n\nmodule $moduleName
   ↪ installed${NC}"

```

## Code C.4: Bash completion for EPICS

File: `epics_BASH_completion.sh`

```

1  # auto completion for a running softIOC
2  #
3  # Tutorials / Sources:
4  # https://web.archive.org/web/
   ↪ 20200507173259/https://debian-
   ↪ administration.org/
   ↪ article/317/
   ↪ An_introduction_to_bash_completion_part_2
5  # https://www.gnu.org/savannah-checkouts/
   ↪ gnu/bash/manual/bash.html
   ↪ #A-Programmable-Completion-Example
6  #
7  # author: Stefan Lunkenheimer
   ↪ (stelunke@uni-mainz.de)
8  # date: 2018-06-13
9  # Version 1.2
10
11 _epics()
12 {
13     local cur prev opts
14     COMPREPLY=()
15     cur="${COMP_WORDS[COMP_CWORD]}"
16     prev="${COMP_WORDS[COMP_CWORD-1]}"
17     if [ -e ${EPICS_ROOT}/dbl_global.txt ]
18     then
19         opts=$(cat ${EPICS_ROOT}/dbl.txt
   ↪ ${EPICS_ROOT}/dbl_global.txt |
   ↪ sort | uniq)
20     else
21         opts=$(cat ${EPICS_ROOT}/dbl.txt)
22     fi
23
24     if [[ ${cur} == *.* ]] ; then
25         local cur1=${cur%.*}
26         #local cur2=${cur##*.}
27         local param="NAME DESC SCAN PREC EGU
   ↪ INPA INPB CALC HOPR LOPR VAL"
28         local fields=$(for x in $param; do
   ↪ echo ${cur1}.${x}; done)
29         COMPREPLY=( $(compgen -W "${fields}"
   ↪ -- "${cur}") )
30         return 0
31     else
32         COMPREPLY=( $(compgen -W "${opts}"
   ↪ -- "${cur}") )
33         return 0
34     fi
35
36     return 0
37 }
38 complete -F _epics caget
39 complete -F _epics caput
40 complete -F _epics camonitor
41 complete -F _epics logreader

```

## Code C.5: ZSH completion for EPICS

File: *epics\_ZSH\_completion.sh*

```

1  # auto completion for a running softIOC
2  #
3  # Tutorials / Sources:
4  # https://github.com/zsh-users/zsh-
  ↪ completions/blob/master/zsh-completions-
  ↪ howto.org
5  # https://stackoverflow.com/questions/
  ↪ 17318913/make-zsh-complete-arguments-
  ↪ from-a-file
6  #
7  # author: Stefan Lunkenheimer
  ↪ (stelunke@uni-mainz.de)
8  # date: 2018-06-13
9  # Version 1.2
10
11 _epics()
12 {
13     cmds=
14     if [ -e ${EPICS_ROOT}/dbl_global.txt ]
15     then
16         cmds=( ${uf}"$(cat ${EPICS_ROOT}/dbl.txt
  ↪ ${EPICS_ROOT}/dbl_global.txt | sort |
  ↪ uniq)" )
17     else
18         cmds=( ${uf}"$(<${EPICS_ROOT}/dbl.txt)"
  ↪ )
19     fi
20
21     # description: The character ':' makes
  ↪ Problems, because the description uses it
  ↪ as seperator for their individual
  ↪ description.
22     # There are two Solutions:
23     # In the file use \: instead of : then
  ↪ everything works fine
24     # use the argument function
25
26     # describe 'a description of the completion
  ↪ options' cmds
27     _arguments '*:foo:({cmds})'
28 }
29
30 compdef _epics caget caput camonitor
  ↪ logreader

```

## Code C.6: Header file of the C++ class for the MUX0 register of the ADS1248 chip.

File: *MUX0.h*

```

1  /*
2  * MUX0.h
3  *
4  * Created on: Oct 16, 2018
5  * Author: stelunke (stelunke@uni-mainz.de)
6  */
7
8  #ifndef MUX0_H_
9  #define MUX0_H_
10
11 #include <stdint>
12
13 namespace MUX0
14 {
15
16     enum class BCS : std::uint8_t
17     {
18         off = 0b00,
19         on_0_5uA = 0b01,
20         on_2uA = 0b10,
21         on_10uA = 0b11
22     };
23
24     enum class AIN : std::uint8_t
25     {
26         AIN0 = 0x0,
27         AIN1 = 0x1,
28         AIN2 = 0x2,
29         AIN3 = 0x3,
30         AIN4 = 0x4,
31         AIN5 = 0x5,
32         AIN6 = 0x6,
33         AIN7 = 0x7
34     };
35
36     class Register
37     {
38     public:
39         Register ( );
40
41         Register (BCS, AIN, AIN );
42
43         void setBCS(BCS);
44
45         void setMUX_SP(AIN);
46
47         void setMUX_SN(AIN);
48
49         std::uint8_t getBitset();
50
51     private:
52         BCS m_bcs;
53         AIN m_muxSP;
54         AIN m_muxSN;
55
56         enum Bit
57         {
58             BCS_BIT=6,
59             MUX_SP_BIT=3,
60             MUX_SN_BIT=0
61         };
62     };
63
64 } /* namespace MUX0 */
65
66 #endif /* MUX0_H_ */

```



**Code C.7:** Source file of the C++ class for the MUX0 register of the ADS1248 chip.

File: MUX0.cpp

```

1  /*
2  * MUX0.cpp
3  *
4  * Created on: Oct 16, 2018
5  * Author: stelunke (stelunke@uni-mainz.de)
6  */
7
8  #include "MUX0.h"
9
10 namespace MUX0
11 {
12
13 Register::Register ( ):
14     m_bcs{BCS::off},
15     m_muxSP{AIN::AIN0},
16     m_muxSN{AIN::AIN1}
17 {
18
19 }
20
21 Register::Register (BCS bcs, AIN muxSP, AIN
22 ↪ muxSN):
23     m_bcs{bcs},
24     m_muxSP{muxSP},
25     m_muxSN{muxSN}
26 {
27 }
28
29 void Register::setBCS(BCS bcs)
30 {
31     m_bcs = bcs;
32 }
33
34 void Register::setMUX_SP(AIN muxSP)
35 {
36     m_muxSP = muxSP;
37 }
38
39 void Register::setMUX_SN(AIN muxSN)
40 {
41     m_muxSN = muxSN;
42 }
43
44 std::uint8_t Register::getBitset()
45 {
46     std::uint8_t bitset = 0x00;
47     bitset |=
48     ↪ static_cast<std::uint8_t>(m_bcs) <<
49     ↪ BCS_BIT;
50     bitset |=
51     ↪ static_cast<std::uint8_t>(m_muxSP) <<
52     ↪ MUX_SP_BIT;
53     bitset |=
54     ↪ static_cast<std::uint8_t>(m_muxSN) <<
55     ↪ MUX_SN_BIT;
56     return bitset;
57 }
58
59 } /* namespace MUX0 */

```

**Code C.8:** Header file of the C++ class for the I<sup>2</sup>C communication of the SiDet device driver.

File: RPiI2C.h

```

1  /**
2  * \class RPiI2C
3  *
4  * \brief I2C connection class for Raspberry
5  ↪ Pi
6  *
7  * Singleton to handle the communication for
8  ↪ the I2C Bus of a Raspberry Pi
9  ↪ (/dev/i2c-1).
10 *
11 * \todo For a RPi4 multiple I2C Bus
12 ↪ connections are possible
13 *
14 * \author Stefan Lunkenheimer
15 * Contact: stelunke@uni-mainz.de
16 *
17 * Created on: Thu Feb 27th, 2020
18 *
19 * Source: https://raspberrypi-
20 ↪ projects.com/pi/programming-in-c/i2c/
21 ↪ using-the-i2c-interface
22 */
23
24 #ifndef RPII2C_H
25 #define RPII2C_H
26
27 #include <stdint>
28 #include <string>
29
30 #include "mlog.h"
31
32 class RPiI2C final
33 {
34 public:
35     // Delete Copy constructor for Singleton
36     RPiI2C(const RPiI2C%) = delete;
37
38     // Delete Move constructor for Singleton
39     RPiI2C(const RPiI2C&&) = delete;
40
41     // Delete Copy assignment operator for
42     ↪ Singleton
43     RPiI2C& operator= (const RPiI2C) =
44     ↪ delete;
45
46     // Default deconstructor
47     ~RPiI2C();
48
49     /**
50     * \brief write data to I2C bus
51     *
52     * write @ref buffer of length @ref len
53     ↪ to a device on I2C bus with the adress
54     ↪ @ref slaveAddr
55     * @param slaveAddr I2C Address of the
56     ↪ device
57     * @param buffer Data to write to device
58     * @param len Length of the data buffer
59     * @return <br>true</br>: write to device
60     ↪ successful; <br>false</br>: write to
61     ↪ device failed
62     */
63     bool

```

```

51     writeI2C(uint8_t slaveAddr, uint8_t*
    ↪     buffer, uint8_t len);
52
53     /**
54     * \brief read data from I2C bus
55     *
56     * read data @ref buffer of length @ref
    ↪     len from device on I2C bus with the
    ↪     address @ref slaveAddr
57     * @param slaveAddr I2C Address of the
    ↪     device
58     * @param buffer Data from device
59     * @param len Length of the data buffer
60     * @return <br>true</br>: write to device
    ↪     succesfull; <br>>false</br>: write to
    ↪     device failed
61     */
62     bool
63     readI2C(uint8_t slaveAddr, uint8_t*
    ↪     buffer, uint8_t len);
64
65     /**
66     * \brief status of the I2C connection
67     *
68     * @return <br>true</br> I2C connection
    ↪     ready; <br>>false</br> I2C connection
    ↪     closed
69     */
70     bool
71     getStatus() const;
72
73     /**
74     * \brief get the class Identifier
75     *
76     * @return class name
77     */
78     std::string
79     IsA() const;
80
81     /**
82     * \brief get Instance of the Singleton
    ↪     class
83     *
84     * The first instance opens the I2C
    ↪     connection.
85     *
86     * @return Instance of RPiI2C
87     */
88     static RPiI2C&
89     device()
90     {
91         if (!m_device)
92             m_device = new RPiI2C();
93         return *m_device;
94     }
95
96     /**
97     * \brief open I2C connection
98     *
99     * @return <br>true</br> I2C connection
    ↪     ready; <br>>false</br> I2C connection
    ↪     failed
100    */
101    bool
102    open();
103
104    /**
105    * \brief reopen I2C connection
106    *
107    * @return <br>true</br> I2C connection
    ↪     ready; <br>>false</br> reopen failed
108    */
109    bool
110    reopen();
111
112    /**
113    * \brief close I2C connection
114    *
115    * @return <br>true</br> I2C closed;
    ↪     <br>>false</br> failed
116    */
117    bool
118    close();
119
120    /**
121    * \brief set log level
122    *
123    * @param new log level
124    */
125    void
126    set_level(spdlog::level::level_enum
    ↪     level);
127
128    /**
129    * \brief set log level
130    *
131    * @param new log level
132    */
133    void
134    set_level(unsigned int level);
135
136
137    private:
138        /// static pointer used to ensure a
    ↪     single instance of the class
139        static RPiI2C* m_device;
140
141        /// private constructor for Singleton
142        RPiI2C();
143
144        /// owm logger
145        mxlog m_logger;
146
147        /// I2C file handler
148        int m_fd_I2C;
149
150    };
151
152
153    #endif // RPII2C_H

```

**Code C.9:** Source file of the C++ class for the I<sup>2</sup>C communication of the SiDet device driver.

File: RPiI2C.cpp

```

1  /**
2  * \file RPiI2C.cpp
3  *
4  * \author Stefan Lunkenheimer
5  * Contact: stelunke@uni-mainz.de
6  *
7  * Created on: Thu Feb 27th, 2020
8  *
9  * Source: https://raspberrypi-projects.com/
    ↪     pi/programming-in-c/i2c/using-the-i2c-
    ↪     interface
10     */
11
12     #include "RPiI2C.h"
13     #include "spdlog/logger.h"
14     #include <stdio.h>

```



```

15 #include <iostream>
16 #include <unistd.h> //Needed for I2C
   ↪ port
17 #include <fcntl.h> //Needed for I2C
   ↪ port
18 #include <sys/ioctl.h> //Needed for I2C
   ↪ port
19 #include <linux/i2c-dev.h> //Needed for I2C
   ↪ port
20
21 RPiI2C* RPiI2C::m_device = nullptr;
22
23 RPiI2C::RPiI2C():
24     m_logger(this->IsA()),
25     m_fd_I2C(-1)
26 {
27     this->open();
28     m_logger.debug("constructed");
29 }
30
31 RPiI2C::~RPiI2C()
32 {
33     this->close();
34     delete m_device;
35     m_logger.debug("deconstructed");
36 }
37
38 bool RPiI2C::writeI2C(uint8_t slaveAddr,
   ↪ uint8_t* buffer, uint8_t len)
39 {
40     if (ioctl(m_fd_I2C, I2C_SLAVE, slaveAddr)
   ↪ >= 0)
41     {
42         if (write(m_fd_I2C, buffer, len) ==
   ↪ len)
43         {
44             if (m_logger.get_level() ==
   ↪ spdlog::level::trace)
45             {
46                 std::string s = "";
47                 for (unsigned int i=0; i<len;
   ↪ i++)
48                     s = fmt::format("{}{:}
   ↪ #04x} ",s,
   ↪ buffer[i]);
49                 m_logger.trace("{}:#04x} write
   ↪ {:} {:}", slaveAddr, len,
   ↪ s);
50             }
51             return true;
52         }
53         else
54         {
55             m_logger.error("Unable to write
   ↪ to slave: {:#04x}",
   ↪ slaveAddr);
56         }
57     }
58     else
59         m_logger.error("Failed to acquire bus
   ↪ access and/or talk to slave
   ↪ {:#04x}", slaveAddr);
60
61     return false;
62 }
63
64 bool RPiI2C::readI2C(uint8_t slaveAddr,
   ↪ uint8_t* buffer, uint8_t len)
65 {
66     if (ioctl(m_fd_I2C, I2C_SLAVE, slaveAddr)
   ↪ >= 0)
67     {
68         if (read(m_fd_I2C, buffer, len) ==
   ↪ len)
69         {
70             if (m_logger.get_level() ==
   ↪ spdlog::level::trace)
71                 {
72                     std::string s = "";
73                     for (unsigned int i=0; i<len;
   ↪ i++)
74                         s = fmt::format("{}{:}
   ↪ #04x} ",s,
   ↪ buffer[i]);
75                     m_logger.trace("{}:#04x} read
   ↪ {:} {:}", slaveAddr, len,
   ↪ s);
76                 }
77                 return true;
78             }
79             else
80             {
81                 m_logger.error("Unable to read
   ↪ from slave {:#04x}",
   ↪ slaveAddr);
82             }
83         }
84         else
85             m_logger.error("Failed to acquire bus
   ↪ access and/or talk to slave
   ↪ {:#04x}", slaveAddr);
86
87         return false;
88     }
89
90 bool RPiI2C::getStatus() const
91 {
92     if (m_fd_I2C >= 0)
93         return true;
94     return false;
95 }
96
97 std::string RPiI2C::IsA() const
98 {
99     return "RPiI2C";
100 }
101
102 bool RPiI2C::open()
103 {
104     if (getStatus())
105     {
106         m_logger.warn("Connection already
   ↪ open, try reopen()");
107         return true;
108     } else if ((m_fd_I2C =
   ↪ ::open("/dev/i2c-1", O_RDWR)) >= 0)
109     {
110         m_logger.info("Connection to bus {}
   ↪ opened", "/dev/i2c-1");
111         return true;
112     }
113
114     m_logger.error("Failed to open the i2c
   ↪ bus {}", "/dev/i2c-1");
115     return false;
116 }
117
118 bool RPiI2C::reopen()
119 {
120     return close() && open();
121 }
122
123 bool RPiI2C::close()
124 {
125     if (m_fd_I2C >= 0)
126     {
127         ::close(m_fd_I2C);
128         m_fd_I2C = -1;
129         m_logger.info("Connection to bus {}
   ↪ closed", "/dev/i2c-1");
130     }
131     return true;

```

```

132 }
133
134 void
135 RPiI2C::set_level(spdlog::level::level_enum
↪ level)
136 {
137     m_logger.set_level(level);
138 }
139
140 void
141 RPiI2C::set_level(unsigned int level)
142 {
143     m_logger.set_level(level);
144 }

```

**Code C.10:** Header file of the C++ class for the `mælog` class for the `SiDet` device driver.

File: `mælog.h`

```

1  /**
2   * \class mælog
3   *
4   * \brief mælogger
5   *
6   * This class allows to specify separat
↪ loggers per object. The advantage is,
7   * that the log_level for each logger can set
↪ seperately.
8   *
9   * \todo Add file_logger
10  * \todo Think about this class, maybe it's
↪ better derivate from this class
11  * \todo How to dis/enable spdlog?
12  *
13  * \author Stefan Lunkenheimer
14  * Contact: stelunke@uni-mainz.de
15  *
16  * Created on: Thu Mar 12th, 2020
17  *
18  */
19
20 #ifndef MXLOG_H
21 #define MXLOG_H
22
23 #include <memory>
24 #include <iostream>
25
26 #include "spdlog/spdlog.h"
27 #include "spdlog/sinks/stdout_color_sinks.h"
28 #include "spdlog/sinks/basic_file_sink.h"
29
30 class mxlog final
31 {
32 public:
33     /**
34      * \brief default constructor
35      *
36      * @param type logger name for
↪ identification, should be unique
37      * @param level initial logger level
38      */
39     mxlog(std::string type = "undefined",
↪ spdlog::level::level_enum
↪ level=spdlog::level::info);
40
41     /**
42      * \brief default constructor
43      *
44      * @param type logger name for
↪ identification, should be unique
45      * @param level initial logger level
46      */
47     mxlog(std::string type, unsigned int
↪ level);
48
49     /**
50      * \brief default constructor
51      *
52      * @param level initial logger level
53
54     * @param type logger name for
↪ identification, should be unique
55     */
56     mxlog(spdlog::level::level_enum level,
↪ std::string type = "undefined");
57
58     /**
59      * \brief default constructor
60      *
61      * @param level initial logger level
62      * @param type logger name for
↪ identification, should be unique
63      */
64     mxlog(unsigned int level, std::string
↪ type = "undefined");
65
66     /// default destructor
67     ~mxlog();
68
69     /**
70      * \brief log string with fmt formatting
↪ @ref format with severity @ref level
71      *
72      * @param level log level
73      * @param format fmt format string
74      * @args the arguments used to format the
↪ string @ref format
75      */
76     template<typename ... Args>
77     void log(int level, const std::string&
↪ format, const Args ... args )
78     {
79         m_logger->log(level, format,
↪ args...);
80     }
81
82     /**
83      * \brief log string with fmt formatting
↪ @ref format with severity @ref level
84      *
85      * @param level log level
86      * @param format fmt format string
87      * @args the arguments used to format the
↪ string @ref format
88      */
89     template<typename ... Args>
90     void log(spdlog::level::level_enum
↪ level, const std::string& format,
↪ const Args ... args )
91     {
92         m_logger->log(level, format,
↪ args...);
93     }
94
95     /**
96      * \brief log string with fmt formatting
↪ @ref format with severity @ref level
97      *
98      * @param level log level
99      * @param format fmt format string
100     */

```

```

101     * @args the arguments used to format the 153     {
↪ string @ref format 154         m_logger->error(format, args...);
102     */ 155     }
103     template<typename ... Args> 156
104     void trace(const std::string& format, 157     /**
↪ const Args ... args ) 158     * \brief log string with fmt formatting
105     { 159     ↪ @ref format with severity critical
106         m_logger->trace(format, args...); 160     *
107     } 161     * @param format fmt format string
108 162     * @args the arguments used to format the
109     /** 163     ↪ string @ref format
110     * \brief log string with fmt formatting 164     */
↪ @ref format with severity debug 165     template<typename ... Args>
111     * 166     void critical(const std::string& format,
112     * @param format fmt format string 167     ↪ const Args ... args )
113     * @args the arguments used to format the 168     {
↪ string @ref format 169     m_logger->critical(format, args...);
114     */ 170     }
115     template<typename ... Args> 171     /**
116     void debug(const std::string& format, 172     * \brief get the current log level
↪ const Args ... args ) 173     *
117     { 174     * @return current log level
118         m_logger->debug(format, args...); 175     */
119     } 176     spdlog::level::level_enum get_level()
120 177     ↪ const;
121     /** 178     /**
122     * \brief log string with fmt formatting 179     * \brief set log level
↪ @ref format with severity info 180     *
123     * 181     * @param log_level
124     * @param format fmt format string 182     */
125     * @args the arguments used to format the 183     void set_level(spdlog::level::level_enum
↪ string @ref format 184     ↪ log_level);
126     */ 185     /**
127     template<typename ... Args> 186     * \brief set log level
128     void info(const std::string& format, 187     *
↪ const Args ... args ) 188     * @param log_level
129     { 189     */
130         m_logger->info(format, args...); 190     void set_level(unsigned int log_level);
131     } 191     // void
132 192     // enableFile(std::string filename);
133     /** 193     //void trace
134     * \brief log string with fmt formatting 194     private:
↪ @ref format with severity warning 195     /// current log level
135     * 196     spdlog::level::level_enum m_level;
136     * @param format fmt format string 197
137     * @args the arguments used to format the 198     /// console sink - equal for all logger
↪ string @ref format 199     static std::shared_ptr<spdlog::sinks::]
138     */ 200     ↪ stdout_color_sink_mt>
139     template<typename ... Args> 201     ↪ m_console_sink;
140     void warn(const std::string& format, 202     /**
↪ const Args ... args ) 203     //static std::shared_ptr<spdlog::sinks::]
141     { 204     ↪ basic_file_sink_mt>
142         m_logger->warn(format, args...); 205     ↪ m_file_sink;
143     } 206
144 207     /// specific logger for each object
145     /** 208     std::shared_ptr<spdlog::logger> m_logger;
146     * \brief log string with fmt formatting 209     };
↪ @ref format with severity error 210
147     * 211     #endif //MXLOG_H
148     * @param format fmt format string
149     * @args the arguments used to format the
↪ string @ref format 212
150     */
151     template<typename ... Args>
152     void error(const std::string& format,
↪ const Args ... args )

```

**Code C.11:** Source file of the C++ class for the `malog` class for the `SiDet` device driver.

File: `malog.cpp`

```

1  /**
2  * \file malog.cpp
3  *
4  * \author Stefan Lunkenheimer
5  * Contact: stelunke@uni-mainz.de
6  *

```

```

7  * Created on: Thu Mar 12th, 2020
8  *
9  **/
10
11 #include <stdio.h>
12 #include <iostream>
13 #include <unistd.h>
14 #include <memory>
15 #include <string>
16
17 #include "mxlog.h"
18
19
20 std::shared_ptr<spdlog::sinks::j
↳ stdout_color_sink_mt>
↳ mxlog::m_console_sink = nullptr;
21 //std::shared_ptr<spdlog::sinks::j
↳ basic_file_sink_mt>
↳ mxlog::m_file_sink=nullptr;
22
23 mxlog::mxlog(std::string type,
↳ spdlog::level::level_enum level):
  m_level(level),
  m_logger(nullptr)
24 {
25     if (!m_console_sink)
26         m_console_sink =
↳ std::make_shared<spdlog::sinks::j
↳ stdout_color_sink_mt>();
27
28     m_logger = std::make_shared<spdlog::j
↳ logger>(type,
↳ m_console_sink);
29     m_logger->set_level(m_level);
30     log(spdlog::level::info, "Logger \"{}\"
↳ created", type);
31 }
32
33
34
35 mxlog::mxlog(std::string type, unsigned int
↳ level):
36     mxlog(type, static_cast<spdlog::level::j
↳ level_enum>(level))
37 {
38 }
39
40
41
42 mxlog::mxlog(spdlog::level::level_enum
↳ level, std::string type):
43     mxlog(type, level)
44 {
45 }
46
47 mxlog::mxlog(unsigned int level, std::string
↳ type):
48     mxlog(type, static_cast<spdlog::level::j
↳ level_enum>(level))
49 {
50 }
51
52
53 mxlog::~mxlog()
54 {
55     m_logger->info("Logger \"{}\" stopped",
↳ m_logger->name());
56     m_logger->flush();
57     m_logger.reset();
58 }
59
60 spdlog::level::level_enum mxlog::get_level()
↳ const {
61     return m_level;
62 }
63
64 void mxlog::set_level(spdlog::level::j
↳ level_enum log_level)
↳ {
65     m_level = log_level;
66     m_logger->info("set_level to {}",
↳ spdlog::level::j
↳ to_string_view(log_level));
67     m_logger->set_level(m_level);
68 }
69
70 void mxlog::set_level(unsigned int log_level)
↳ {
71     if (log_level < spdlog::level::n_levels)
72         set_level(static_cast<spdlog::j
↳ level::level_enum>(log_level));
73     else
74         set_level(spdlog::level::off);
75 }

```

**Code C.12:** The `read_aiSIDET` and `write_aiSIDET` function defined to connect the `SiDet` library with EPICS using the device support interface.

File: `SILICONDETECTOR.cpp`

```

1  #include "aiRecord.h"
2  #include "aoRecord.h"
3  #include "epicsExport.h"
4  #include "iocsh.h"
5  #include <vector>
6  #include <algorithm> // std::find_if
7  #include <string>
8  #include <unistd.h>
9  #include <iostream>
10
11 #include "SiDet.h"
12 #include "RPiI2C.h"
13
14 #include <mutex> // std::mutex
15
16 #define MAX_MSG_LENGTH 100
17
18 static std::mutex m_AccessMutex;
19
20 static SiDet * mySiDet = nullptr;
21
22 static void
23     err_msg (std::string recName, std::string e,
↳ std::string pattern = "\u274C%s
↳ \u25AC\u25AC\u25B6 %s!!!")
24 {
25     char msg[MAX_MSG_LENGTH];
26     sprintf(msg, pattern.c_str(),
↳ recName.c_str(), e.c_str());
27     std::cerr << "\033[1;31m" << msg <<
↳ "\033[0m" << std::endl;
28 }
29
30 extern "C" long
31 init_recordSIDET (struct aiRecord *pai)
32 {
33     std::lock_guard<std::mutex>
↳ lock(m_AccessMutex);
34     if (!mySiDet)
35     {
36         mySiDet = new SiDet();
37     }
38     return ( 0 );
39 }
40

```

```

41  /**
42  *
43  * @param pai
44  * @return error code: 2 := success in
↳ engineering units; 3:= Device not found
45  */
46  extern "C" long
47  read_aiSIDET (struct aiRecord *pai)
48  {
49      if (pai->inp.text)
50      {
51          // Split input String
52          std::string input { pai->inp.text };
53          std::string delimiter = " ";
54          std::vector < std::string > pString;
55
56          size_t pos = 0;
57          std::string token;
58          while (( pos = input.find(delimiter)
↳ ) != std::string::npos)
59          {
60              token = input.substr(0, pos);
61              pString.push_back(token);
62              input.erase(0, pos +
↳ delimiter.length());
63          }
64          pString.push_back(input);
65
66          // Check number of input parameters
67          if (pString.size() == 0)
68          {
69              err_msg(pai->name, "Not Enough
↳ Parameters");
70              return ( 3 );
71          }
72          if
↳ (pString[0].compare("getThreshold")
↳ == 0)
73          {
74              double value = 0;
75              std::lock_guard<std::mutex>
↳ lock(m_AccessMutex);
76              if (!mySidet-
↳ )
↳ >getThreshold(&value))
77              {
78                  err_msg(pai->name, "Can't
↳ Read Threshold");
79                  return ( 3 );
80              }
81              pai->val = value;
82              pai->udf = FALSE;
83              return ( 2 );
84          }
85      }
86      if
↳ (pString[0].compare("getTemperature")
↳ == 0)
87      {
88          double value = 0;
89          std::lock_guard<std::mutex>
↳ lock(m_AccessMutex);
90          if (!mySidet-
↳ )
↳ >getPCBTemperature(&value))
91          {
92              err_msg(pai->name, "Can't
↳ Read getTemperature");
93              return ( 3 );
94          }
95          pai->val = value;
96          pai->udf = FALSE;
97          return ( 2 );
98      }
99      // getResistance Readout
100
101
102      if
↳ (pString[0].compare("getResistance")
↳ == 0)
103      {
104          if (pString.size() < 2)
105          {
106              err_msg(pai->name, "Not
↳ Enough Parameters");
107              return ( 3 );
108          }
109          std::uint8_t channel =
↳ atoi(pString[1].c_str());
110          if (channel >= 2)
111          {
112              err_msg(pai->name, "Channel
↳ Out Of Range");
113              return ( 3 );
114          }
115          double value = 0;
116          std::lock_guard<std::mutex>
↳ lock(m_AccessMutex);
117          if (!mySidet-
↳ )
↳ >getResistance(&value,
↳ channel))
118          {
119              err_msg(pai->name, "Can't
↳ Read Resistance");
120              return ( 3 );
121          }
122          pai->val = value;
123          pai->udf = FALSE;
124          return ( 2 );
125      }
126      }
127      else
128      {
129          err_msg(pai->name, "No input");
130          return ( 3 );
131      }
132      err_msg(pai->name, "Unknowm Command");
133      return ( 3 );
134  }
135  extern "C" long
136  write_aoSIDET (struct aoRecord *pai)
137  {
138      if (pai->out.text)
139      {
140          // Split input String
141          std::string input { pai->out.text };
142          std::string delimiter = " ";
143          std::vector < std::string > pString;
144
145          size_t pos = 0;
146          std::string token;
147          while (( pos = input.find(delimiter)
↳ ) != std::string::npos)
148          {
149              token = input.substr(0, pos);
150              pString.push_back(token);
151              input.erase(0, pos +
↳ delimiter.length());
152          }
153          pString.push_back(input);
154
155          // Check number of input parameters
156          if (pString.size() == 0)
157          {
158              err_msg(pai->name, "Not Enough
↳ Parameters");
159              return ( 3 );
160          }
161          if
↳ (pString[0].compare("setThreshold")
↳ == 0)
162          {
163              double value = 0;
164              std::lock_guard<std::mutex>
↳ lock(m_AccessMutex);
165              if (!mySidet-
↳ )
↳ >setThreshold(&value))
166              {
167                  err_msg(pai->name, "Can't
↳ Set Threshold");
168                  return ( 3 );
169              }
170              pai->val = value;
171              pai->udf = FALSE;
172              return ( 2 );
173          }
174          if
↳ (pString[0].compare("setTemperature")
↳ == 0)
175          {
176              double value = 0;
177              std::lock_guard<std::mutex>
↳ lock(m_AccessMutex);
178              if (!mySidet-
↳ )
↳ >setPCBTemperature(&value))
179              {
180                  err_msg(pai->name, "Can't
↳ Set Temperature");
181                  return ( 3 );
182              }
183              pai->val = value;
184              pai->udf = FALSE;
185              return ( 2 );
186          }
187          if
↳ (pString[0].compare("setResistance")
↳ == 0)
188          {
189              double value = 0;
190              std::lock_guard<std::mutex>
↳ lock(m_AccessMutex);
191              if (!mySidet-
↳ )
↳ >setResistance(&value, channel))
192              {
193                  err_msg(pai->name, "Can't
↳ Set Resistance");
194                  return ( 3 );
195              }
196              pai->val = value;
197              pai->udf = FALSE;
198              return ( 2 );
199          }
200          err_msg(pai->name, "Unknowm Command");
201          return ( 3 );
202      }
203      }
204      err_msg(pai->name, "No output");
205      return ( 3 );
206  }

```

```

165     {
166         std::lock_guard<std::mutex>
167         ↪ lock(m_AccessMutex);
168         double value = (double) pai->val;
169         if (!mySidet-
170         ↪ >setThreshold(value)))
171     {
172         err_msg(pai->name, "Can't Set
173         ↪ Threshold");
174         return ( 3 );
175     }
176     else
177     {
178         err_msg(pai->name, "No input");
179         return ( 3 );
180     }
181     err_msg(pai->name, "Unknowm Command");
182     return ( 3 );
183 }
184 }

```

Code C.13: The definition of the IOC shell command `MXSiDet_SetLogLevel`.File: `SILICONDETECTOR.cpp`

```

1  /*
2  struct iocshFuncDef {
3      const char *name;
4      int nargs;
5      const iocshArg * const *args;
6  };
7
8  struct iocshArg {
9      const char *name;
10     iocshArgType type;
11 }iocshArg;
12
13 void showCallFunc(const iocshArgBuf *args);
14
15 iocshArgInt     int     args[i].ival
16 iocshArgDouble double  args[i].dval
17 iocshArgString  char *  args[i].sval
18 iocshArgPersistentString char * args[i].sval
19 iocshArgPdbbase void *  args[i].vval
20 iocshArgArgv    int     args[i].aval.ac
21                 char ** args[i].aval.av
22 */
23
24 /* drvXxx code, FuncDef and CallFunc
25 ↪ definitions ... */
26 static const iocshArg SetMXSiDetLogLevelArg0
27 ↪ = { "log_level", iocshArgString };
28 static const iocshArg SetMXSiDetLogLevelArg1
29 ↪ = { "device", iocshArgString };
30
31 static const iocshArg *const
32 ↪ SetMXSiDetLogLevelArgs[2] =
33 ↪ { &SetMXSiDetLogLevelArg0,
34 ↪ &SetMXSiDetLogLevelArg1 };
35 static const iocshFuncDef
36 ↪ SetMXSiDetLogLevelDef =
37 ↪ { "MXSiDet_SetLogLevel", 2,
38 ↪ SetMXSiDetLogLevelArgs };
39
40 static void SetMXSiDetLogLevelCallFunc(const
41 ↪ iocshArgBuf *args);
42
43 /* drvXxx registration Function and export
44 ↪ */
45 static void SetMXSiDetLogLevelRegistrar(void)
46 {
47     iocshRegister(&SetMXSiDetLogLevelDef,
48     ↪ SetMXSiDetLogLevelCallFunc);
49 }
50 epicsExportRegistrar
51 ↪ (SetMXSiDetLogLevelRegistrar);
52 /* in dbd add registrar(drvXxxRegistrar) */
53
54 static void printDoc()
55 {
56     std::cout << "\033[1;36m //cyan
57
58     << "Usage: MXSiDet_SetLogLevel
59     ↪ (s,d='all')" <<
60     ↪ std::endl
61     << "Set the minimal Message
62     ↪ log_level s for device d"
63     ↪ << std::endl
64     << "string s: \"trace\",
65     ↪ \"debug\", \"info\",
66     ↪ \"warning\", \"error\",
67     ↪ \"critical\", \"off\" <<
68     ↪ std::endl
69     << "string d: \"all\"
70     ↪ (including: \"SiDet\",
71     ↪ \"MUX\", \"DAC\", \"ADC\",
72     ↪ \"PCB\") or \"I2C\""
73     << "\033[0m" << std::endl;
74     ↪ //back to standard
75 }
76
77 void SetMXSiDetLogLevelCallFunc(const
78 ↪ iocshArgBuf *args)
79 {
80     if (args[0].sval==NULL ||
81     ↪ args[0].sval[0]!='?')
82     {
83         printDoc();
84         return;
85     }
86     std::string arg0{args[0].sval};
87     spdlog::level::level_enum log_level =
88     ↪ spdlog::level::from_str(arg0);
89     if (log_level == spdlog::level::off
90     && arg0.compare("off") != 0)
91     {
92         char msg[50];
93         sprintf(msg, "Severity Level %s
94         ↪ unknown", arg0.c_str());
95         err_msg("MXSiDet_SetLogLevel", msg);
96         printDoc();
97         return;
98     }
99     std::lock_guard<std::mutex>
100    ↪ lock(m_AccessMutex);
101    if (!mySidet)
102    {
103        mySidet = new SiDet();
104    }
105    if (args[1].sval==NULL)
106    {
107        mySidet->set_level(log_level,true);
108        return;
109    }
110    std::string arg1{args[1].sval};
111    if(arg1.compare("all") == 0)

```

```
87     {
88         mySidet->set_level(log_level,true);
89         return;
90     }
91     else if (arg1.compare("SiDet") == 0)
92     {
93         mySidet->set_level(log_level,false);
94         return;
95     }
96     else if (arg1.compare("MUX") == 0)
97     {
98         mySidet->set_level(log_level,SiDet::J
104         ↪ MUX);
105         return;
106     }
107     else if (arg1.compare("DAC") == 0)
108     {
109         mySidet->set_level(log_level,SiDet::J
110         ↪ DAC);
111         return;
112     }
113     }
114     }
115     }
116     }
117     }
118     }
119     }
120     }
121     }
```





---

---

# APPENDIX D

---

## Supplemental Material – Simulation

In this chapter the definition of the TTree structure and the configuration files for the simulation are given.

**Code D.1:** *The TTree structure for the particle and event, stored in a root file at the end of the simulation.*

*File: SFactorFullAnalysisTTree.hh*

```
1  struct Particle{                               9
2    Double_t                                     10  struct Event {
3    vertex_x,vertex_y,vertex_z,mass,            11    Double_t
4    E,pp,p_phi,p_theta,p_ctheta;              12    weight,
5    Int_t PID;                                  13    W, Phi, q2,
6  }m_IS_electron, m_IS_oxygen,                  14    CMS_theta, CMS_phi;
   ↪ m_FS_electron, m_FS_particle1,              15    Int_t nr = 0;
   ↪ m_FS_particle2, m_ImS_oxygen,              16  }m_event;
   ↪ m_ImS_vPhoton;                              17
7
8  const char *TTreestructureParticle =          18  const char *TTreestructureEvent =
   ↪ "vertex_x/D:vertex_y:vertex_z:mass:        ↪ "weight/D:W:Phi:1_q2:CMT_theta:
   ↪ E:pp:p_phi:p_theta:p_ctheta:PID/I";        ↪ CMS_phi:nr/I";
```

**Code D.2:** *Main configuration file of the simulation.*

*File: SFactor\_CrossSectionIntegrator\_ZeroDegTagger.xml*

```
1  <magix>                                         13    <path>@MAGIX_DIR@/lib/MXWare/
2
3    <!-- Enabling the logging system -->         14    ↪ plugins</path>
4    <logging>                                     15    <printlist>true</printlist>
5    <logfile>./log/MXLog.txt</logfile>         16    <stopaterr>>false</stopaterr>
6    <console>true</console>                     17    </pluginmanager>
7    <severity>info</severity>
8  </logging>                                     18    <plugin pluginname="MXUtilityTools">
9
10   <!-- Configuring the plugin manager          19    <datatable>@CMAKE_INSTALL_PREFIX@/
   ↪ -->                                         ↪ share/MXWare/myPartData.mcd</
11  <pluginmanager>                                ↪ datatable>
12    <path>@CMAKE_INSTALL_PREFIX@/lib/          20    <LogPDataTable>true</LogPDataTable>
   ↪ MXWare/plugins</path>                    21    </plugin>
13
14    <path>@MAGIX_DIR@/lib/MXWare/            22    <plugin pluginname=
   ↪ plugins</path>                             ↪ "CrossSectionIntegrator">
```

```

24 <threads>8</threads> 66 <E1data>@CMAKE_INSTALL_PREFIX@/_
25 <engine type="NR_HQGen"/> ↪ share/MXWare/SE1data.csv</_
26 <autoseed>true</autoseed> ↪ E1data>
27 <samples>20000000</samples> 67 <E2data>@CMAKE_INSTALL_PREFIX@/_
28 <accuracy>0.001</accuracy> ↪ share/MXWare/SE2data.csv</_
29 <beam type=_ ↪ E2data>
↪ "IdealBeamParticleGenerator"> 68 </model>
30 <momentum>105</momentum> 69
31 <center>(0, 0)</center> 70 <full_analysis type="SFactor_
32 <pid>11</pid> ↪ FullAnalysisTTree">
33 <current>1.0</current> 71 <rootfile>
34 <helicity>0.2</helicity> 72 <path>@CMAKE_INSTALL_PREFIX@/_
35 </beam> ↪ out/SFactor_TTree_
36 <beam_analysis ↪ ZeroDeg.root</path>
↪ type="InitialBeamRootSummary"> ↪
↪ 73 <openmode>RECREATE</openmode>
37 <rootfile> ↪
38 <path>@CMAKE_INSTALL_PREFIX@/_ ↪
↪ out/SFactor_ZeroDeg.root</_ ↪
↪ path> 74 </rootfile>
↪ 75 <rootfolder>SFactor</rootfolder>
39 <openmode>RECREATE</openmode> 76 </full_analysis>
↪ 77
40 </rootfile> 78 <finalstate_analysis
41 <rootfolder>SFactor/InitialBeam/_ ↪
↪ </rootfolder> 79 <rootfile>
↪ 80 <path>@CMAKE_INSTALL_PREFIX@/_
42 </beam_analysis> ↪ out/SFactor_ZeroDeg.root</_
43 <target type=_ ↪ path>
↪ 81 <openmode>UPDATE</openmode>
44 < "TransverseUniformLayer_TIG"> ↪
↪ <density>0.21219</density> <!-- 82 </rootfile>
↪ 2.1219*1018 particles / 83 <rootfolder>SFactor_FSG</_
↪ cm2--> ↪ rootfolder>
45 <x_extension>[-10:10]</x_ ↪
↪ extension> 84 <seconds>2419200</seconds> <!-- 4
46 <y_extension>[-10:10]</y_ ↪
↪ extension> 85 <histo_CMSEnergy>
47 <z_extension>[-1:1]</z_extension> 86 <name>CMS_Energy</name>
48 <pid>1000080160</pid> 87 <title>Center of Mass Energy of
↪ the outgoing
49 </target> ↪ particles</title>
50 <initialstate_analysis ↪ <nxbins>200</nxbins>
↪ type="InitialStateRootSummary"> 88 <xmin>0.2</xmin>
↪ 89 <xmax>14</xmax>
51 <rootfile> 90 </histo_CMSEnergy>
52 <path>@CMAKE_INSTALL_PREFIX@/_ ↪
↪ out/SFactor_ZeroDeg.root</_ ↪
↪ path> 91 <histo_q2>
↪ 92 <name>q2</name>
53 <openmode>UPDATE</openmode> ↪
↪ 93 <title>Momentum transfer of the
↪ scattered electron to the
54 </rootfile> ↪ target</title>
55 <rootfolder>SFactor/InitialState/_ ↪
↪ </rootfolder> 94 <nxbins>100</nxbins>
↪ 95 <xmin>-1e-6</xmin>
56 </initialstate_analysis> 96 <xmax>0</xmax>
57 <final_state type="Inelastic_ ↪
↪ TwoBodyGenerator_FSG"> 97 </histo_q2>
58 <PID1>1000020040</PID1> 98 <histo_q2Inverse>
59 <PID2>1000060120</PID2> 99 <name>Inverse q2</name>
60 <name>TwoBodyGenerator</name> 100 <title>Inverse Momentum transfer
↪ of the scattered electron to
61 <cutoff>0</cutoff> 101 ↪ the target</title>
62 <dWmax>25</dWmax> 102 <nxbins>200</nxbins>
63 </final_state> 103 <xmin>-15e3</xmin>
64 <model type=_ ↪ <xmax>-1e3</xmax>
↪ "SFactorCrossSectionCalculator"> 104 </histo_q2Inverse>
65 <structure>extrap</structure> 105 <hist_thetavsTkinScat>
↪ 106 <nxbins>320</nxbins>
↪ 107 <xmin>-0.0001</xmin>
↪ 108

```

```

109     <xmax>0.005</xmax>           121     </histo_thetaScat>
110     <nybins>320</nybins>         122
111     <ymin>55</ymin>               123     </finalstate_analysis>
112     <ymin>100</ymax>             124
113     </hist_thetavsTkinScat>      125     <_include>@CMAKE_INSTALL_PREFIX@/_
114                                     ↪ share/MXWare/examples/TwoBody_
115     <histo_thetaScat>              ↪ DetectorSetup_ZeroDegTagger.xml
116     <name>theta_scattered</name>   ↪ </_include>
117     <title>The azimuthal angle of  126
118     ↪ the direction of the         127     </plugin>
119     ↪ scattered particle</title>  128
118     <nxbins>320</nxbins>          129
119     <xmin>-0.0001</xmin>          130 </magix>
120     <xmax>0.015</xmax>

```

**Code D.3:** *Detector setup used in the simulation for Phase 1**File: TwoBody\_DetectorSetup\_Spectrometer.xml*

```

1 <acceptance                       34     </acceptance>
  ↪ type="CoincidenceAcceptance"> 35
2 <acceptance type="                36     <acceptance type="
  ↪ "RotatingSpectrometerAcceptance"> 37     ↪ "SiliconStripAcceptance">
3   <acceptV>true</acceptV>         37     <name>SiliconStripDetector3</name>
4   <acceptancefile>@MAGIX_DIR@/@DATA_ 38     <distance>100</distance>
  ↪ INSTALL_DIR@/MXSpectrometer_ 39     <rotation_angle_deg>150
  ↪ 2013.dat</acceptancefile>      40     ↪ </rotation_angle_deg>
5   <central_momentum>-75</central_ 40     <acceptV>true</acceptV>
  ↪ momentum>                       41     <VBlockSize>10</VBlockSize>
6   <distance>700</distance>        42     <width>50</width>
7   <name/>                          43     <length>50</length>
8   <offplane_angle>0</offplane_angle> 44     <PIDRestrict>1000020040
9   <offsetz>0</offsetz>            45     ↪ </PIDRestrict>
10  <rotation_angle_deg>-10</rotation_ 45     </acceptance>
  ↪ angle_deg>                       46
11 </acceptance>                    47     <acceptance type="
12                                     ↪ "SiliconStripAcceptance">
13 <acceptance type="AcceptanceSet"> 48     <name>SiliconStripDetector5</name>
14   <acceptance type="                49     <distance>100</distance>
  ↪ "SiliconStripAcceptance">      50     <rotation_angle_deg>-90
15   <name>SiliconStripDetector1</name> 50     ↪ </rotation_angle_deg>
16   <distance>100</distance>         51     <acceptV>true</acceptV>
17   <rotation_angle_deg>30</rotation_ 52     <VBlockSize>10</VBlockSize>
  ↪ angle_deg>                       53     <width>50</width>
18   <acceptV>true</acceptV>         54     <length>50</length>
19   <VBlockSize>10</VBlockSize>     55     <PIDRestrict>1000020040
20   <width>50</width>               56     ↪ </PIDRestrict>
21   <length>50</length>             56     </acceptance>
22   <PIDRestrict>1000020040         57
  ↪ </PIDRestrict>                 58     <acceptance type="
23 </acceptance>                    58     ↪ "SiliconStripAcceptance">
24                                     ↪
25 <acceptance                       59     <name>SiliconStripDetector6</name>
  ↪ type="SiliconStripAcceptance"> 60     <distance>100</distance>
26   <name>SiliconStripDetector2</name> 61     <rotation_angle_deg>-150
27   <distance>100</distance>         61     ↪ </rotation_angle_deg>
28   <rotation_angle_deg>90</rotation_ 62     <acceptV>true</acceptV>
  ↪ angle_deg>                       63     <VBlockSize>10</VBlockSize>
29   <acceptV>true</acceptV>         64     <width>50</width>
30   <VBlockSize>10</VBlockSize>     65     <length>50</length>
31   <width>50</width>               66     <PIDRestrict>1000020040
32   <length>50</length>             66     ↪ </PIDRestrict>
33   <PIDRestrict>1000020040         67     </acceptance>
  ↪ </PIDRestrict>                 68     </acceptance>
                                     69     </acceptance>

```

**Code D.4:** *Detector setup used in the simulation for Phase 2**File: TwoBody\_DetectorSetup\_ZeroDegTagger.xml*

```

1 <acceptance                               39         <PIDRestrict>1000020040</j
   ↪ type="CoincidenceAcceptance">        40         ↪ PIDRestrict>
2                                           41         </acceptance>
3 <acceptance type=j                        42         <acceptance
   ↪ "ZeroDegreeSpectrometerAcceptance">  43         ↪ type="SiliconStripAcceptance">
4   <momentumRange>[7:105]</j              44         <name>SiliconStripDetector4</name>
   ↪ momentumRange>                       45         <distance>100</distance>
5   <thetaRange_deg>[0:0.85]</j           46         <rotation_angle_deg>-30</j
   ↪ thetaRange_deg>                       47         ↪ rotation_angle_deg>
6 </acceptance>                            48         <acceptV>>true</acceptV>
7                                           49         <VBlockSize>10</VBlockSize>
8 <acceptance type="AcceptanceSet">        50         <width>50</width>
9   <acceptance type=j                      51         <length>50</length>
   ↪ "SiliconStripAcceptance">            52         <PIDRestrict>1000020040</j
10  <name>SiliconStripDetector1</name>      53         ↪ PIDRestrict>
11  <distance>100</distance>                54         <PIDRestrict>1000060120</j
12  <rotation_angle_deg>30</j              55         ↪ PIDRestrict>
   ↪ rotation_angle_deg>                   56         </acceptance>
13  <acceptV>>true</acceptV>                 57         <acceptance type=j
14  <VBlockSize>10</VBlockSize>            58         ↪ "SiliconStripAcceptance">
15  <width>50</width>                       59         <name>SiliconStripDetector5</name>
16  <length>50</length>                    60         <distance>100</distance>
17  <PIDRestrict>1000020040</j            61         <rotation_angle_deg>-90</j
   ↪ PIDRestrict>                          62         ↪ rotation_angle_deg>
18 </acceptance>                            63         <acceptV>>true</acceptV>
19                                           64         <VBlockSize>10</VBlockSize>
20 <acceptance                               65         <width>50</width>
   ↪ type="SiliconStripAcceptance">        66         <length>50</length>
21  <name>SiliconStripDetector2</name>      67         <PIDRestrict>1000020040</j
22  <distance>100</distance>                68         ↪ PIDRestrict>
23  <rotation_angle_deg>90</j              69         </acceptance>
   ↪ rotation_angle_deg>                   70         <acceptance type=j
24  <acceptV>>true</acceptV>                 71         ↪ "SiliconStripAcceptance">
25  <VBlockSize>10</VBlockSize>            72         <name>SiliconStripDetector6</name>
26  <width>50</width>                       73         <distance>100</distance>
27  <length>50</length>                    74         <rotation_angle_deg>-150</j
28  <PIDRestrict>1000020040</j            75         ↪ rotation_angle_deg>
   ↪ PIDRestrict>                          76         <acceptV>>true</acceptV>
29 </acceptance>                            77         <VBlockSize>10</VBlockSize>
30                                           78         <width>50</width>
31   <acceptance type=j                      79         <length>50</length>
   ↪ "SiliconStripAcceptance">            80         <PIDRestrict>1000020040</j
32  <name>SiliconStripDetector3</name>      81         ↪ PIDRestrict>
33  <distance>100</distance>                82         </acceptance>
34  <rotation_angle_deg>150</j             83         </acceptance>
   ↪ rotation_angle_deg>                   84         </acceptance>
35  <acceptV>>true</acceptV>                 85
36  <VBlockSize>10</VBlockSize>            86
37  <width>50</width>                       87
38  <length>50</length>                    88

```

# List of Figures

2.1	Schematic view of a scattering process . . . . .	7
2.2	Maxwell-Boltzmann distribution for specific elements at the temperatures for specific burning stages in stars . . . . .	10
2.3	Sketch of the Gamow peak, Maxwell-Boltzmann distribution and Coulomb penetration factor . . . . .	13
2.4	Schematic view of the Proton-Proton I-III chains . . . . .	14
2.5	Schematic view of the CNO cycle . . . . .	16
2.6	Reaction scheme of the two most important reactions of the helium burning stage . . . . .	17
2.7	Level scheme of the nuclei involved in the helium burning stage . . . . .	19
2.8	Level diagram of the $^{16}\text{O}$ nucleus . . . . .	20
2.9	The total S-factor compared to the ground state and cascade transitions . . . . .	22
2.10	The E1 and E2 ground state contribution compared to the contribution of cascade transitions . . . . .	23
2.11	Coulomb phase shift $\cos(\Phi_{12})$ . . . . .	23
2.12	Angular contribution of E1 and E2 for the ground state transition . . . . .	26
2.13	Cross section of the time-reversed reaction $^{16}\text{O}(\gamma, \alpha)^{12}\text{C}$ and direct reaction $^{12}\text{C}(\alpha, \gamma)^{16}\text{O}$ in comparison . . . . .	28
2.14	E1 and E2 contribution data . . . . .	33
2.15	Production factor of some key isotopes as function of the S-factor . . . . .	35
3.1	First order Feynman diagram for the electro-disintegration reaction $^{16}\text{O}(e, e'\alpha)^{12}\text{C}$ . . . . .	38
4.1	Overview of the structure of MESA . . . . .	53
4.2	Overview of the MESA hall and the experiments . . . . .	55
4.3	Overview of the P2 experiment . . . . .	55
4.4	Overview of the DarkMESA experiment . . . . .	56
4.5	Overview of the MAGIX experiment . . . . .	57
4.6	Laval Nozzle and Catcher . . . . .	58
4.7	Overview of the gas jet target . . . . .	59
4.8	Overview of the MAGIX spectrometer . . . . .	60
4.9	Schematic principle of a TPC . . . . .	61
4.10	The current 3D concept of the trigger veto system . . . . .	62
4.11	Schematic view of the cross section of the diamond-shaped segmented trigger layer . . . . .	62
4.12	Schematic view of the MAGIX trigger veto system . . . . .	63
4.13	Prototype of the silicon strip detector . . . . .	64

4.14	Schematic representation of the working principle of the silicon strip detector . . . . .	65
4.15	Schematic representation of the zero degree tagger for MAGIX . .	66
4.16	Schematic view of the experimental setup for Phase 0/1 . . . . .	69
4.17	Schematic view of the experimental setup for Phase 2 . . . . .	70
5.1	Overview of the major software components of an IOC . . . . .	73
5.2	Overview of the task distribution for the IOC and the CACs . . .	77
5.3	Overview of the channel access process . . . . .	78
5.4	Example of a CSS OPI . . . . .	82
5.5	General structure of the SiDet test stand from the point of view of the control software . . . . .	94
5.6	Overview of the SiDet readout electronics . . . . .	98
5.7	Class diagram MXSiDet . . . . .	99
5.8	Example output of the <code>mxlog</code> class for the SiDet device driver . .	100
5.10	CSS OPI for the silicon strip detector test stand . . . . .	102
6.1	Illustration of the distribution of the random numbers using a MC integration to calculate $\pi$ . . . . .	106
6.2	Example of a less suitable function for the use of MC integration .	108
6.3	Separation of the Feynman diagram of the electro-disintegration reaction of $^{16}\text{O}$ into two parts . . . . .	113
6.4	Histogram representing the weighted cms energy $E^{\text{cms}}$ for Phase 2	119
6.5	Hitmap for the silicon strip detector array . . . . .	120
6.6	Number of expected events distributed over the cms angular of the $\alpha$ -particle . . . . .	121
6.7	Histogram representing the isotropically distributed cms angle of the $\alpha$ -particle . . . . .	122
6.8	Histogram of the weighted cms angle of the $\alpha$ -particle . . . . .	122
6.9	E1 and E2 simulation results . . . . .	124

# List of Tables

2.1	S-factor results at $E_{\text{cms}} = 300$ keV of previous works . . . . .	32
5.1	Overview of the ICs placed on the SiDet frontend-board . . . . .	98
6.1	Comparison of MC integration for computing $\pi$ with a pseudo-random and a quasi-random generator . . . . .	107
6.2	Results of a MC integration for a less suitable function . . . . .	109
6.3	Summary of the relevant simulation and analysis parameters. . .	123
B.1	List of adopted levels for $^8\text{Be}$ . . . . .	135
B.2	List of adopted levels for $^{12}\text{C}$ . . . . .	135
B.3	List of adopted levels for $^{16}\text{O}$ . . . . .	136
B.4	List of adopted levels for $^{20}\text{N}$ . . . . .	137
B.5	Atomic and core masses . . . . .	138
B.6	Calculated Gamow peaks for the reactions of the helium burning stage . . . . .	138
B.7	Calculated Coulomb penetration factors for the reactions of the helium burning stage . . . . .	139
B.8	Physical Constants . . . . .	139
B.9	Coordinate system for the target and generator system . . . . .	140
B.10	Base units used in the simulation software . . . . .	140
B.11	Ranges for the S-factor extrapolation using AZURE2 . . . . .	140





# List of Codes

5.1	IOC shell example – dbLoadRecords . . . . .	74
5.2	Example calc record . . . . .	75
5.3	Example of a Device Support C interface . . . . .	76
5.4	Example of a asynDriver serial Port configuration . . . . .	78
5.5	Example of a protocol and the corresponding record . . . . .	79
5.6	Example of a record using dev-gpio . . . . .	81
5.7	Example of a <code>thisEPICS.sh</code> file . . . . .	84
5.8	Example of the <code>epics.service</code> file . . . . .	85
5.9	Example using the <code>udevadmin</code> command . . . . .	86
5.10	Example for a udev rule. . . . .	86
5.11	Example of the command <code>dbLoadTemplate</code> . . . . .	88
5.12	Example of the command <code>dbLoadMultiRecords</code> . . . . .	89
5.13	Example of the Profibus StreamDevice bus . . . . .	91
5.14	Example of a record using the ADC1248 module . . . . .	92
5.15	Example of a <code>calc</code> record converting <code>U_Expo_New</code> format to <code>mbar</code> . . . . .	95
5.16	Example of the db and proto file for the HMP4040 device . . . . .	97
5.17	Example of some of the SiDet records . . . . .	101
C.1	Installation Script EPICS . . . . .	141
C.2	Installation Script MXSlowControl . . . . .	143
C.3	Installation Script Module MXFrontend . . . . .	145
C.4	Bash completion for EPICS . . . . .	145
C.5	ZSH completion for EPICS . . . . .	146
C.6	Header file of the C++ class MUX0 . . . . .	146
C.7	Source file of the C++ class MUX0 . . . . .	147
C.8	Header file of the C++ class for the I <sup>2</sup> C communication . . . . .	147
C.9	Source file of the C++ class for the I <sup>2</sup> C communication . . . . .	148
C.10	Header file of the C++ class <code>mxlog</code> . . . . .	150
C.11	Source file of the C++ class <code>mxlog</code> . . . . .	151
C.12	<code>read_aiSIDET</code> and <code>write_aiSIDET</code> function of <code>MXSiDet</code> . . . . .	152
C.13	Definition of <code>MXSiDet_SetLogLevel</code> . . . . .	154
D.1	TTree structure for the particle and the event . . . . .	157
D.2	Main configuration file of the simulation. . . . .	157
D.3	Detector setup used in the simulation for Phase 1 . . . . .	159
D.4	Detector setup used in the simulation for Phase 2 . . . . .	160



# Acronyms

<b>ADC</b>	analog-to-digital converter	90
<b>API</b>	Application Programming Interface	90
<b>CA</b>	Channel Access	73, 77
<b>CAC</b>	CA client	73, 76
<b>CAS</b>	CA server	73, 75
<b>cms</b>	center of mass system	6
<b>CNO cycle</b>	Carbon-Nitrogen-Oxygen cycle	15
<b>CSS</b>	Control System Studio	81
<b>DB</b>	database	73
<b>DBD</b>	database definition file	74
<b>EB</b>	External Beam	54
<b>EPICS</b>	Experimental Physics and Industrial Control System	72
<b>ER</b>	Energy Recovering	54
<b>ERL</b>	Extreme Relativistic Limit	37
<b>FLRM</b>	five-level R-matrix method	29
<b>GEM</b>	Gas Electron Multiplier	61
<b>GPIO</b>	general purpose input-output	81
<b>GUI</b>	Graphical User Interface	81
<b>HRM</b>	hybrid R-matrix method	30
<b>I<sup>2</sup>C</b>	Inter-Integrated Circuit	93
<b>I<sup>2</sup>C-MUX</b>	I <sup>2</sup> C-Multiplexer	93
<b>IC</b>	integrated circuit	96
<b>IOC</b>	Input-Output-Controller	73
<b>KM</b>	K-matrix method	30
<b>LDM</b>	Light Dark Matter	56
<b>MAGIX</b>	Mainz Gas Injection Target Experiment	54, 56
<b>MAMBO</b>	Milliamper Booster	54
<b>MAMI</b>	Mainz Microtron	59, 68
<b>MARC</b>	MESA Arc	54
<b>MC</b>	Monte Carlo	31, 104
<b>MEEC</b>	MESA Enhanced ELBE Cryomodule	54
<b>MELBA</b>	MESA Low-energy Beam Apparatus	54
<b>MESA</b>	Mainz Energy-Recovering Superconducting Accelerator	54
<b>MXSlowControl</b>	MAGIX Slow Control System	71, 82

<b>OPI</b>	Operator Interface	81
<b>PID</b>	particle identification	112
<b>PP chain</b>	Proton-Proton chain	13
<b>Profibus</b>	Process Field Bus	90
<b>PV</b>	Process Variable	75
<b>pyEPICS</b>	EPICS Channel Access for Python	80
<b>RM</b>	R-matrix method	29
<b>RPi</b>	Raspberry Pi	81
<b>SiDet</b>	silicon strip detector	93
<b>SPI</b>	Serial Peripheral Interface	91
<b>STEAM</b>	Small Thermalized Electron Source At Mainz	54
<b>TCP</b>	Transmission Control Protocol	77
<b>TLRM</b>	three-level R-matrix method	29
<b>ToF</b>	Time of Flight	67
<b>TPC</b>	Time Projection Chamber	61
<b>UDP</b>	User Datagram Protocol	77
<b>ZDT</b>	Zero Degree Tagger	65

# Index

- $T_9$ , 13
- $\alpha_{\text{em}}$ , *see* Fine structure constant
- $\alpha$ -capture reaction, 18
- $\delta$ -function, 39
- U\_Expo\_New, 95
- caArchiver, *see* Archiver
- caget, 76
- camonitor, 76
- caput, 76
- dbl, 86
- drvAsynIPPortConfigure, 95
- drvAsynSerialPortConfigure, 93
- makeBaseApp, 100
- systemctl, 83
- systemd, 83
- tmux, 84
- udevadm, 85
- udev, 85
  
- Archiver, 86
- Areal particle density, 57
- AZURE2, 20, 123
  
- Bash, 85
  - completion, 85
- Big Bang nucleosynthesis, 5
- Breit-Wigner, 8
- Byte stream, 78
  
- Carbon-Nitrogen-Oxygen cycle, 15
- Catcher, 58
- Channel access, 73, 77
  - client, 73, 76
  - server, 73, 75
- checksum, 94
- CNO cycle, *see* Carbon-Nitrogen-Oxygen cycle
- Community module, 78
  - asynDriver, 78
  - Autosave, 80
  - dev-gpio, 81
  - pyEPICS, 80
  - StreamDevive, 79
- Control system studio, 81
  - operation interface, 81
- Cosmic background, 26, 125
- Coulomb matrix elements, 51
- Coulomb penetration factor, 11
- Coulomb phase shift difference, 22
- Cross section, 7
  - capture, 8
  - differential, 7
  - elastic, 8
  - inelastic, 8
  - reaction, 8
  - stellar nucleosynthesis, 8
- Current conservation
  - electromagnetic, 40
  - hadronic, 40
  
- DarkMESA, 56
- Data acquisition, 71
- Database, 73
  - definition, 74
- de Broglie wavelength, 8
- Device support, 73, 75
- Driver support, *see* Device support
  
- EB mode, 54
- Elastic scattering, 6
- Elastic scattering phase shifts, 22
- Electro-disintegration, 28
- Electromagnetic current operator, 41
- Electron current, 40
- Electric matrix elements, 51
- Energy transfer, 37
- Energy-conervation, 39
- Environment variable, 83
- EPICS, 72
  - dbLoadRecords, 87
  - dbLoadTemplate, 88
  - module, *see* Community module
- ER mode, 54
  
- Felsenkeller laboratory, 26, 125
- Final state particle generator, 110, 113
- Fine structure constant, 9
- Four-momentum, 37
- Fourier transformation, 41
  
- Gamow peak, 11
  - energy, 11
  - width, 11
- Gas jet target, 58

- Generalized rosenbluth factors, 45
- Graphical user interface, 81
  
- Hadronic current matrix, 40
- Hadronic recoil factor, 40
- Hadronic tensor, 40
- Helium burning stage, 16
- Histogram, 111, 118
  - multi-dimensional, 119
  - two-dimensional, 119
- Hoyle state, 18
- Hydro-Møller, 54
- Hydrogen burning stage, 12
  
- I<sup>2</sup>C master, 96
- I<sup>2</sup>C multiplexer, 96
- I/O hardware, 75
- Importance sampling, 106
- Inelastic scattering, 6
- Initial channel, 6
- Initial state particle generator, 109, 112
- Input-Output-Controller, 73
  - shell, 73
- Invariant mass, 46
- Invariant matrix element, 40
- IOC, *see* Input-Output-Controller
  
- Laval nozzle, 58
- Legendre polynomial, 7, 50
  - associated, 50
- Leptonic tensor, 40
- Lorentz boost, 47
- Lorentz transformation, 46
- Luminosity, 67
- LUNA-MV, 25, 125
  
- MAGIX, 56
  - Phase 0, 68
  - Phase 1, 68, 122
  - Phase 2, 69, 122
- MAMBO, 54
- MARC, 54
- Maxwell-Boltzmann distribution, 9
- Mean value, 104
- MEEC, 54
- MELBA, 54
- MESA, 54
- metastable, 17
- Metric tensor, 38
- Monte Carlo integration, 104
- Multi-pole operator phase, 51
- MXSiDet, 92
  - frontend-board, 96
  - HDMI distribution board, 96
  - library, 96
    - AD5693, 98
    - ADS122C04, 98
    - MPX9808, 98
    - PCA9548A, 99
    - RPiI2C, 97
    - SiDet, 99
  - mxlog, 99
    - documentation, 100
    - unit test, 100
  - module, 100
- MXSlowControl, 71, 82
  - ADC1248, 90
  - dbLoadMultiRecords, 88
  - HMP4040, 95
  - MPT200, 93
  - MXFrontend, 87
  - MXSiDet, *see* MXSiDet
  - StreamDevice Profibus, 90
  - tutorial, 90
  
- Natural system of units, 37
- Nuclear reaction, 5
- Nuclear response functions, 43, 131
  - transverse-longitudinal interference, 43
  - longitudinal, 43
  - transverse, 43
  - transverse-transverse interference, 43
  
- P2 experiment, 55
- Phase shift difference, *see* Coulomb phase shift difference
- Photo-disintegration, 27
- Poisson distribution, 120
- PPI-III chain, *see* Proton-Proton chain
- Process variable, 75
- Protocol, 79
  - file, 79
- Proton-Proton chain, 13
- Python, 80
  - script, 80
  
- Radiative capture, 6
- Random number, 105
  - pseudo-random, 105
  - quasi-random, 105
- Reaction channel, 6
- Reaction rate, 11
  - distribution, 11
- Record, 73
  - field, 74
  - type, 73
- Reduced mass, 9
- Resonance, 8
  - strength, 8
- Rodrigues' formula, 51
- ROOT, 111
  - TTree, 120
  
- S-factor, 9
  - cascade transition, 21
  - E1 contribution, 21, 29, 123
  - E2 contribution, 21, 29, 123
  - ground state transition, 21
- SiDet, 93
- Silicon strip detector, 63
  - test stand, 93
- Simulation, 103

- acceptance, 111, 116
  - coincidence, 118
  - set, 118
  - silicon strip detector, 117
  - spectrometer, 117
  - zero degree tagger, 117
- analysis, 118
- count rate, 119
- cross section, 108, 111
- event generator, 109
- physics model, 110, 115
- Slow control, 71
- Sobol sequence, 105
- Sommerfeld number, 9
- Spherical unit vectors, 41
- STEAM, 54
- Stellar nucleosynthesis, 5
- Three-momentum transfer, 37
- Time Projection Chamber, 61
- Transition charge density, 41
- Transition current density, 41
- Transition matrix element, 41
- Transmission control protocol, 77
- Trigger veto system, 62
- Triple- $\alpha$  reaction, 17
- User datagram protocol, 77
- Variance, 104
- Virtual photon, 37
- Weak mixing angle, 55
- Weinberg angle, 55
- Zero degree tagger, 65
- ZSH, *see* Bash





# Bibliography

- [A5.7] *Autosave R5-7-1*. URL: <https://epics.anl.gov/bcda/synApps/autosave/autosave.html> (visited on 09/10/2021).
- [AD] *Data Sheet AD5693R/AD5692R/AD5691R/AD5693*. Analog Devices. URL: [https://www.analog.com/media/en/technical-documentation/data-sheets/AD5693R\\_5692R\\_5691R\\_5693.pdf](https://www.analog.com/media/en/technical-documentation/data-sheets/AD5693R_5692R_5691R_5693.pdf) (visited on 11/12/2021).
- [ADS] *ADS124x 24-Bit, 2-kSPS, Analog-To-Digital Converters With Programmable Gain Amplifier (PGA) For Sensor Measurement*. Texas Instruments. URL: <https://www.ti.com/lit/gpn/ads1248> (visited on 09/22/2021).
- [ADS2] *ADS122C04 24-Bit, 4-Channel, 2-kSPS, Delta-Sigma ADC With I<sup>2</sup>C Interface*. Texas Instruments. URL: <https://www.ti.com/lit/ds/symlink/ads122c04.pdf> (visited on 11/12/2021).
- [Ago+20] M. Agostini et al. (The Borexino Collaboration). “Experimental evidence of neutrinos produced in the CNO fusion cycle in the Sun”. In: *Nature* 587 (Nov. 2020), pp. 577–582. DOI: 10.1038/s41586-020-2934-0.
- [APS04] *EPICS Lecture Series at APS*. Argonne National Laboratory. URL: <https://epics.anl.gov/docs/GSWE.php> (visited on 09/01/2021).
- [AS74] M. Abramowitz and I. A. Stegun. *Handbook of Mathematical Functions, With Formulas, Graphs, and Mathematical Tables*, USA: Dover Publications, Inc., 1974.
- [Ass+06] M. Assunção et al. “E1 and E2 S factors of  $^{12}\text{C}(\alpha, \gamma)^{16}\text{O}$  from  $\gamma$ -ray angular distributions with a  $4\pi$ -detector array”. In: *Phys. Rev. C* 73 (5 May 2006), p. 055801. DOI: 10.1103/PhysRevC.73.055801.
- [ASYN21] *asynDriver: Asynchronous Driver Support*. URL: <https://epics-modules.github.io/master/asyn/> (visited on 09/01/2021).
- [Aul21] S. Aulenbacher. Personal communication. 2021.
- [Azu+10] R. E. Azuma et al. “AZURE: An  $R$ -matrix code for nuclear astrophysics”. In: *Phys. Rev. C* 81 (4 Apr. 2010), p. 045805. DOI: 10.1103/PhysRevC.81.045805.
- [Azu+94] R. E. Azuma et al. “Constraints on the low-energy E1 cross section of  $^{12}\text{C}(\alpha, \gamma)^{16}\text{O}$  from the  $\beta$ -delayed  $\alpha$  spectrum of  $^{16}\text{N}$ ”. In: *Phys. Rev. C* 50 (2 Aug. 1994), pp. 1194–1215. DOI: 10.1103/PhysRevC.50.1194.
- [B3.14] *EPICS Base Release 3.14.12*. Argonne National Laboratory. URL: <https://epics.anl.gov/base/R3-14/12.php> (visited on 09/01/2021).
- [Bal17] Balabanski, Dimiter L. “Nuclear physics and astrophysics experiments at ELI-NP: The emerging future”. In: *EPJ Web Conf.* 165 (2017), p. 01003. DOI: 10.1051/epjconf/201716501003.
- [BD64] J. D. Bjorken and S. D. Drell. *Relativistic quantum mechanics*. International series in pure and applied physics. New York, NY: McGraw-Hill, 1964.

- [Bec+18] D. Becker et al. “The P2 experiment”. In: *Eur. Phys. J. A* 54.11 (2018), p. 208. DOI: 10.1140/epja/i2018-12611-6. arXiv: 1802.04759.
- [Bem+18] Bemmerer, Daniel et al. “Felsenkeller 5 MV underground accelerator: Towards the Holy Grail of Nuclear Astrophysics  $^{12}\text{C}(\alpha, \gamma)^{16}\text{O}$ ”. In: *EPJ Web Conf.* 178 (2018), p. 01008. DOI: 10.1051/epjconf/201817801008.
- [Ber16] N. Berger. “Tracking and  $Q^2$  determination for the P2 experiment”. In: *Physics of fundamental Symmetries and Interactions*. Poster. PSI2016. Villigen PSI, Switzerland, Oct. 2016. URL: <https://www.blogs.uni-mainz.de/fb08-ag-berger/files/2018/08/P2TrackingPSI2016s.pdf>.
- [Ber21] N. Berger. Personal communication. 2021.
- [Bir21] M. Biroth. Personal communication. 2021.
- [Bru01] C. R. Brune. “Electric-multipole interference effects in the  $^{12}\text{C}(\alpha, \gamma_0)^{16}\text{O}$  reaction”. In: *Phys. Rev. C* 64 (5 Oct. 2001), p. 055803. DOI: 10.1103/PhysRevC.64.055803.
- [BSS14] A. Bisoi, S. Sarkar, and M. S. Sarkar, eds. *Forbidden E1 transitions and isospin mixing in self-conjugate sd-shell nuclei*. Nucl. Phys. 59. DAE Symposium. 2014. URL: [www.sympnp.org/proceedings](http://www.sympnp.org/proceedings).
- [Bur+57] E. M. Burbidge et al. “Synthesis of the Elements in Stars”. In: *Rev. Mod. Phys.* 29 (4 Oct. 1957), pp. 547–650. DOI: 10.1103/RevModPhys.29.547.
- [BW91] J. Blatt and V. Weisskopf. *Theoretical Nuclear Physics*. Dover Books on Physics Series. Dover Publications, 1991.
- [Chr+20a] M. Christmann et al. “Detector response of Cherenkov radiators for calorimetry in the energy range below 14 MeV”. In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 960 (2020), p. 163665. DOI: <https://doi.org/10.1016/j.nima.2020.163665>.
- [Chr+20b] M. Christmann et al. “Instrumentation and optimization studies for a beam dump experiment (BDX) at MESA — DarkMESA”. In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 958 (2020). Proceedings of the Vienna Conference on Instrumentation 2019, p. 162398. DOI: <https://doi.org/10.1016/j.nima.2019.162398>.
- [Chr21a] M. Christmann. Personal communication. 2021.
- [Chr21b] M. Christmann. “The DarkMESA Experiment”. In: *721. WE-Heraeus-Seminar – Light Dark Matter Searches*. Talk. 2021. URL: <https://www.staff.uni-mainz.de/doria/heraeus/christmann.pdf>.
- [CHY92] Y. Chang, Y. Huo, and G. Yu. “The collision-frequency and reaction rate”. In: *Physics of Fluids B: Plasma Physics* 4.11 (1992), pp. 3621–3626. DOI: 10.1063/1.860370.
- [CSS21] *CSS: Control System Studio*. URL: <http://controlsystemstudio.org/> (visited on 09/01/2021).
- [DB74] P. Dyer and C. Barnes. “The  $^{12}\text{C}(\alpha, \gamma)^{16}\text{O}$  reaction and stellar helium burning”. In: *Nucl. Phys. A* 233.2 (1974), pp. 495–520. DOI: 10.1016/0375-9474(74)90470-9.
- [Deb] *Debian - The Universal operating system*. URL: <https://wiki.debian.org/> (visited on 09/17/2021).
- [deB+17] R. J. deBoer et al. “The  $^{12}\text{C}(\alpha, \gamma)^{16}\text{O}$  reaction and its implications for stellar helium burning”. In: *Rev. Mod. Phys.* 89.3 (Sept. 2017), p. 035007. DOI: 10.1103/RevModPhys.89.035007.
- [DiG+15] B. DiGiovine et al. “Bubble chambers for experiments in nuclear astrophysics”. In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 781 (May 2015), pp. 96–104. DOI: 10.1016/j.nima.2015.01.060.

- [DL05] R. M. Dreizler and C. S. Lüdde. *Theoretische Physik 2 Elektrodynamik und spezielle Relativitätstheorie*. 1st ed. Springer-Lehrbuch. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005.
- [Don+17] T. W. Donnelly et al. *Foundations of Nuclear and Particle Physics*. Cambridge University Press, 2017. DOI: 10.1017/9781139028264.
- [Dor+18] L. Doria et al. *Search for light dark matter with the MESA accelerator*. 2018. arXiv: 1809.07168.
- [Dor+19] *Dark Matter at the Intensity Frontier: the new MESA electron accelerator facility*. Vol. 360. ALPS2019. 2019. DOI: <https://doi.org/10.22323/1.360.0022>.
- [Edm57] A. Edmonds. *Angular Momentum in Quantum Mechanics*. Investigations in physics. Princeton University Press, 1957.
- [Emi17] N. Emig. “Machbarkeitsstudie eines EPICS-basierten Slow Control Systems für das Experiment MAGIX”. Unpublished. Bachelor thesis. Johannes Gutenberg University, 2017.
- [Epe+11] E. Epelbaum et al. “Ab Initio Calculation of the Hoyle State”. In: *Phys. Rev. Lett.* 106 (19 May 2011), p. 192501. DOI: 10.1103/PhysRevLett.106.192501.
- [EPICS21] *EPICS: Experimental Physics and Industrial Control System*. URL: <https://epics-controls.org/> (visited on 09/01/2021).
- [FDM19] I. Frišćić, T. W. Donnelly, and R. G. Milner. “A New Approach to Determine Radiative Capture Reaction Rates at Astrophysical Energies”. In: *Phys. Rev. C* 100 (2019), p. 025804. DOI: 10.1103/PhysRevC.100.025804.
- [Fey04] M. Fey. “Im Brennpunkt der nuklearen Astrophysik: Die Reaktion  $^{12}\text{C}(\alpha, \gamma)^{16}\text{O}$ ”. Ph.D. thesis. University of Stuttgart, 2004. DOI: 10.18419/opus-4716.
- [FF14] M. Freer and H. Fynbo. “The Hoyle state in  $^{12}\text{C}$ ”. In: *Progress in Particle and Nuclear Physics* 78 (2014), pp. 1–23. DOI: <https://doi.org/10.1016/j.pnnp.2014.06.001>.
- [Gai12] M. Gai (UConn-Yale-Duke-Weizmann-PTB-UCL Collaboration). “Studies in Nuclear Astrophysics with an Optical Readout TPC (O-TPC) at HI $\gamma$ S”. In: *Journal of Physics: Conference Series* 337 (Feb. 2012), p. 012054. DOI: 10.1088/1742-6596/337/1/012054.
- [Gal+18] S. Gales et al. “The extreme light infrastructure—nuclear physics (ELI-NP) facility: new horizons in physics with 10 PW ultra-intense lasers and 20 MeV brilliant gamma beams”. In: *Reports on Progress in Physics* 81.9 (Aug. 2018), p. 094301. DOI: 10.1088/1361-6633/aacfe8.
- [Gei20] J. Geimer. “Aufbau eines ortsauflösendem Siliziumdetektorsystems zum Nachweis von  $\alpha$ -Teilchen”. Unpublished. Master thesis. Johannes Gutenberg University, 2020.
- [Gei21a] J. Geimer. Personal communication. 2021.
- [Gei21b] J. Geimer. “The silicon strip detector setup for MAGIX”. In: *721. WE-Heraeus-Seminar – Light Dark Matter Searches*. Poster. 2021.
- [Gia+01] L. Gialanella et al. “The E1 capture amplitude in  $^{12}\text{C}(\alpha, \gamma_0)^{16}\text{O}$ ”. In: *The European Physical Journal A* 11 (July 2001), pp. 357–370. DOI: 10.1007/s100500170075.
- [GPIO] F. Feldbauer. *EPICS-devGPIO*. URL: <https://github.com/ffeldbauer/epics-devgpio> (visited on 09/13/2021).
- [Gri+18] S. Grieser et al. “A Cryogenic Supersonic Jet Target for Electron Scattering Experiments at MAGIX@MESA and MAMI”. In: *Nucl. Instr. Meth. A* 906 (2018), pp. 120–126. DOI: 10.1016/j.nima.2018.07.076.
- [Gri18] S. Grieser. “Cluster-Jet Targets for the PANDA-, MAGIX-, and Cryoflash-Experiments at Hadron-, Lepton-, and Laser-Facilities”. Ph.D. thesis. Uni-

- versity of Münster, 2018. URL: [https://www.uni-muenster.de/imperia/md/content/physik\\_kp/agkhoukaz/bamadipdok/dok\\_silke\\_grieser.pdf](https://www.uni-muenster.de/imperia/md/content/physik_kp/agkhoukaz/bamadipdok/dok_silke_grieser.pdf).
- [GS64] I. Gel'fand and G. E. Shilov. *Properties and Operations*. Vol. I. Generalized Functions. New York, NY: Academic Press Inc, 1964.
- [Gug14] Guglielmetti, Alessandra. “The LUNA experiment at Gran Sasso Laboratory”. In: *EPJ Web of Conferences* 78 (2014), p. 07001. DOI: 10.1051/epjconf/20147807001.
- [Gül21] P. Gülker. Personal communication. 2021.
- [Ham+05] J. Hammer et al. “New determination of the  $^{12}\text{C}(\alpha, \gamma)^{16}\text{O}$  reaction rate from  $\gamma$ -ray angular distribution measurements”. In: *Nuclear Physics A* 752 (2005). Proceedings of the 22nd International Nuclear Physics Conference (Part 2), pp. 514–521. DOI: <https://doi.org/10.1016/j.nuclphysa.2005.02.056>.
- [HFP18] R. J. Holt, B. W. Filippone, and S. C. Pieper. *Sensitivity study for the  $^{12}\text{C}(\alpha, \gamma)^{16}\text{O}$  astrophysical reaction rate*. 2018. arXiv: 1809.10176.
- [HFP19] R. J. Holt, B. W. Filippone, and S. C. Pieper. “Impact of  $^{16}\text{O}(\gamma, \alpha)^{12}\text{C}$  measurements on the  $^{12}\text{C}(\alpha, \gamma)^{16}\text{O}$  astrophysical reaction rate”. In: *Phys. Rev. C* 99 (5 May 2019), p. 055802. DOI: 10.1103/PhysRevC.99.055802.
- [Hjo11] M. Hjorth-Jensen. “The carbon challenge”. In: *Physics* 4 (38 May 2011). URL: <http://link.aps.org/doi/10.1103/Physics.4.38>.
- [HM84] F. Halzen and A. D. Martin. *Quarks and Leptons: An Introductory Course in Modern Particle Physics*. Jan. 1984. ISBN: 978-0-471-88741-6.
- [HMP] *R&S@HMP Series Power Supplies User Manual*. Rohde & Schwarz. URL: [https://www.rohde-schwarz.com/webhelp/hmpseries\\_html\\_usermanual\\_en/HMPSeries\\_HTML\\_UserManual\\_en.htm](https://www.rohde-schwarz.com/webhelp/hmpseries_html_usermanual_en/HMPSeries_HTML_UserManual_en.htm) (visited on 09/24/2021).
- [Hua+17] W. Huang et al. “The AME2016 atomic mass evaluation (I). Evaluation of input data; and adjustment procedures”. In: *Chinese Physics C* 41.3 (Mar. 2017), p. 030002. DOI: 10.1088/1674-1137/41/3/030002.
- [Hua+76] K.-N. Huang et al. “Neutral-atom electron binding energies from relaxed-orbital relativistic Hartree-Fock-Slater calculations  $2 \leq Z \leq 106$ ”. In: *Atomic Data and Nuclear Data Tables* 18.3 (1976), pp. 243–291. DOI: [https://doi.org/10.1016/0092-640X\(76\)90027-9](https://doi.org/10.1016/0092-640X(76)90027-9).
- [Hug+17] F. Hug et al. “MESA - an ERL Project for Particle Physics Experiments”. In: *Proc. of Linear Accelerator Conference (LINAC'16), East Lansing, MI, USA, 25-30 September 2016* (East Lansing, MI, USA). Linear Accelerator Conference 28. Geneva, Switzerland: JACoW, May 2017, pp. 313–315. DOI: <https://doi.org/10.18429/JACoW-LINAC2016-MOP106012>.
- [Hug+20] F. Hug et al. “Status of the MESA ERL Project”. In: *Proc. ERL'19* (Berlin, Germany). ICFA Advanced Beam Dynamics Workshop on Energy Recovery Linacs 63. JACoW Publishing, Geneva, Switzerland, June 2020, pp. 14–17. DOI: 10.18429/JACoW-ERL2019-MOCXBS05.
- [ISG13] Izotov, Y. I., Stasińska, G., and Guseva, N. G. “Primordial  $4\text{He}$  abundance: a determination based on the largest sample of H II regions with a methodology tested on model H II regions”. In: *A&A* 558 (Oct. 2013), A57. DOI: 10.1051/0004-6361/201220782.
- [Kem20] R. F. Kempf. “Stabilisation der Strahlparameter für das P2-Experiment an MESA”. Dissertation. Johannes Gutenberg-Universität Mainz, July 2020. DOI: <http://doi.org/10.25358/openscience-4991>.
- [Knu97] D. E. Knuth. *The Art of Computer Programming, Volume 2 (3rd Ed.): Seminumerical Algorithms*. USA: Addison-Wesley Longman Publishing Co., Inc., 1997.

- [Koz14] A. Kozyreva. “Pair instability supernovae: Evolution, explosion, nucleosynthesis”. Ph. D. thesis. University of Bonn, Germany, 2014. URL: [https://astro.uni-bonn.de/~nlangner/thesis/Kozyreva\\_phd.pdf](https://astro.uni-bonn.de/~nlangner/thesis/Kozyreva_phd.pdf).
- [Kre+88] R. M. Kremer et al. “Coincidence measurement of the  $^{12}\text{C}(\alpha, \gamma)^{16}\text{O}$  cross section at low energies”. In: *Phys. Rev. Lett.* 60 (15 Apr. 1988), pp. 1475–1478. DOI: 10.1103/PhysRevLett.60.1475.
- [Kun+01] R. Kunz et al. “ $^{12}\text{C}(\alpha, \gamma)^{16}\text{O}$ : The Key Reaction in Stellar Nucleosynthesis”. In: *Phys. Rev. Lett.* 86 (15 Apr. 2001), pp. 3244–3247. DOI: 10.1103/PhysRevLett.86.3244.
- [Kun+02] R. Kunz et al. “Astrophysical Reaction Rate of  $^{12}\text{C}(\alpha, \gamma)^{16}\text{O}$ ”. In: 567.1 (Mar. 2002), pp. 643–650. DOI: 10.1086/338384.
- [LB04] R. Larson and V. Bromm. “The First Stars in the Universe”. In: *Scientific American Special Edition* 14.4 (Sept. 2004), p. 4.
- [Led20] B. Ledroit. “Target Induced Halo Formation and Collimation Following MAGIX at MESA”. Dissertation. Johannes Gutenberg-Universität Mainz, Dec. 2020. DOI: <http://doi.org/10.25358/openscience-5808>.
- [LPT03] D. Lunney, J. M. Pearson, and C. Thibault. “Recent trends in the determination of nuclear masses”. In: *Rev. Mod. Phys.* 75 (3 Aug. 2003), pp. 1021–1082. DOI: 10.1103/RevModPhys.75.1021.
- [Lyo86] L. Lyons. *Statistics for Nuclear and Particle Physicists*. Cambridge University Press, 1986. DOI: 10.1017/CB09781139167710.
- [Mak+09] H. Makii et al. “ $E1$  and  $E2$  cross sections of the  $^{12}\text{C}(\alpha, \gamma_0)^{16}\text{O}$  reaction using pulsed  $\alpha$  beams”. In: *Phys. Rev. C* 80 (6 Dec. 2009), p. 065802. DOI: 10.1103/PhysRevC.80.065802.
- [Mat+18] C. Matei et al. “Photodisintegration reactions for nuclear astrophysics studies at ELI-NP”. In: *Journal of Physics: Conference Series* 940 (Jan. 2018), p. 012025. DOI: 10.1088/1742-6596/940/1/012025.
- [May02] T. Mayer-Kuckuk. *Kernphysik. Eine Einführung*. ger. 7., überarb. und erw. Aufl. Stuttgart ; Leipzig ; Wiesbaden: Teubner, 2002.
- [MCP]  $\pm 0.5^\circ\text{C}$  Maximum Accuracy Digital Temperature Sensor. Microchip. URL: <https://www.mouser.de/datasheet/2/268/25095A-15487.pdf> (visited on 11/12/2021).
- [Mer+11] H. Merkel et al. “Search for Light Gauge Bosons of the Dark Sector at the Mainz Microtron”. In: *Phys. Rev. Lett.* 106 (2011), p. 251802. DOI: 10.1103/PhysRevLett.106.251802.
- [Mer21] H. Merkel. Personal communication. 2021.
- [MPT200] OPERATING INSTRUCTIONS MPT200. Pfeiffer Vacuum. URL: [https://www.pfeiffer-vacuum.com/filepool/file/digiline/pg0025ben\\_c.pdf](https://www.pfeiffer-vacuum.com/filepool/file/digiline/pg0025ben_c.pdf) (visited on 10/11/2021).
- [Mül14] J. Müller. “Entwurf eines Spektrometers für Niederenergieexperimente an MESA”. Unpublished. Diploma thesis. Johannes Gutenberg University, 2014.
- [Nie92] H. Niederreiter. “Random number generation and Quasi-Monte Carlo methods”. In: *CBMS-NSF regional conference series in applied mathematics*. 1992.
- [NuDat] NuDat (National Nuclear Data Center). *Information extracted from the NuDat 2.8 database*. URL: <https://www.nndc.bnl.gov/nudat2/> (visited on 01/25/2021).
- [Oue+96] J. M. L. Ouellet et al. “ $^{12}\text{C}(\alpha, \gamma)^{16}\text{O}$  cross sections at stellar energies”. In: *Phys. Rev. C* 54 (4 Oct. 1996), pp. 1982–1998. DOI: 10.1103/PhysRevC.54.1982.
- [Phi99] A. C. Phillips. *The Physics of Stars*. Second Edition. New York: Wiley, 1999.

- [Pla+12] R. Plag et al. “ $^{12}\text{C}(\alpha, \gamma)^{16}\text{O}$  studied with the Karlsruhe  $4\pi$  BaF<sub>2</sub> detector”. In: *Phys. Rev. C* 86 (1 July 2012), p. 015805. DOI: [10.1103/PhysRevC.86.015805](https://doi.org/10.1103/PhysRevC.86.015805).
- [Pla+87] R. Plaga et al. “The scattering of alpha particles from  $^{12}\text{C}$  and the  $^{12}\text{C}(\alpha, \gamma)^{16}\text{O}$  stellar reaction rate”. In: *Nuclear Physics A* 465.2 (1987), pp. 291–316. DOI: [https://doi.org/10.1016/0375-9474\(87\)90436-2](https://doi.org/10.1016/0375-9474(87)90436-2).
- [Pov+15] B. Povh et al. *Particles and Nuclei: An Introduction to the Physical Concepts*. Graduate Texts in Physics. Springer Berlin Heidelberg, 2015.
- [PyE3] *PyEPICS*. URL: <https://cars9.uchicago.edu/software/python/pyepics3/> (visited on 09/13/2021).
- [Python3] *Python 3*. URL: <https://docs.python.org/3/> (visited on 09/13/2021).
- [RD89] A. Raskin and T. Donnelly. “Polarization in coincidence electron scattering from nuclei”. In: *Annals of Physics* 191.1 (1989), pp. 78–142. DOI: [https://doi.org/10.1016/0003-4916\(89\)90337-0](https://doi.org/10.1016/0003-4916(89)90337-0).
- [Red+87] A. Redder et al. “The  $^{12}\text{C}(\alpha, \gamma)^{16}\text{O}$  cross section at stellar energies”. In: *Nucl. Phys. A* 462.2 (1987), pp. 385–412. DOI: [10.1016/0375-9474\(87\)90555-0](https://doi.org/10.1016/0375-9474(87)90555-0).
- [ROOT] *ROOT: analyzing petabytes of data, scientifically*. URL: <https://root.cern/> (visited on 10/14/2021).
- [Ros53] M. E. Rose. “The Analysis of Angular Correlation and Angular Distribution Data”. In: *Phys. Rev.* 91 (3 Aug. 1953), pp. 610–615. DOI: [10.1103/PhysRev.91.610](https://doi.org/10.1103/PhysRev.91.610).
- [Rot+99] G. Roters et al. “The E1 and E2 capture amplitudes in  $^{12}\text{C}(\alpha, \gamma)^{16}\text{O}$ ”. In: *Eur. Phys. J. A* 6.4 (1999), pp. 451–461. DOI: [10.1007/s100500050369](https://doi.org/10.1007/s100500050369).
- [Sal52] E. E. Salpeter. “Nuclear Reactions in Stars Without Hydrogen.” In: *Astrophys. J.* 115 (Mar. 1952), pp. 326–328. DOI: [10.1086/145546](https://doi.org/10.1086/145546).
- [SC05] M. Salaris and S. Cassisi. *Evolution of Stars and Stellar Populations*. John Wiley & Sons, 2005.
- [Sch+11] D. Schürmann et al. “Study of the 6.05 MeV cascade transition in  $^{12}\text{C}(\alpha, \gamma)^{16}\text{O}$ ”. In: *Physics Letters B* 703.5 (2011), pp. 557–561. DOI: <https://doi.org/10.1016/j.physletb.2011.08.061>.
- [Sch+12] D. Schürmann et al. “The astrophysical S factor of  $^{12}\text{C}(\alpha, \gamma)^{16}\text{O}$  at stellar energy”. In: *Physics Letters B* 711.1 (2012), pp. 35–40. DOI: <https://doi.org/10.1016/j.physletb.2012.03.064>.
- [Sch+21] B. Schlimme et al. “Operation and characterization of a windowless gas jet target in high-intensity electron beams”. In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 1013 (2021), p. 165668. DOI: <https://doi.org/10.1016/j.nima.2021.165668>.
- [Sch21] S. Schlimme. Personal communication. 2021.
- [SDev2.8] *StreamDevice*. URL: <https://paulscherrerinstitute.github.io/StreamDevice/> (visited on 09/01/2021).
- [SDev21] *StreamDevice*. URL: <http://epics.web.psi.ch/software/streamdevice/> (visited on 09/01/2021).
- [Sim21a] D. Simon. Personal communication. 2021.
- [Sim21b] D. Simon. “Gesamtconcept für den MESA-Teilchenbeschleuniger unter besonderer Berücksichtigung von Strahloptik und Kryogenik”. Dissertation. Johannes Gutenberg-Universität Mainz, Apr. 2021. DOI: <https://doi.org/10.25358/openscience-5809>.
- [Sor17] I. Sorokin. “Parameterization-based tracking for the P2 experiment”. In: *EPJ Web Conf.* 150 (2017). Ed. by C. Germain et al., p. 00012. DOI: [10.1051/epjconf/201715000012](https://doi.org/10.1051/epjconf/201715000012).
- [Ste21a] S. Stengel. Personal communication. 2021.

- [Ste21b] S. Stengel. “The MAGIX trigger veto system”. In: *721. WE-Heraeus-Seminar – Light Dark Matter Searches*. Poster. 2021.
- [Tes+16] O. Tesileanu et al. “Charged particle detection at ELI-NP”. In: *Romanian Reports in Physics* 68 (Jan. 2016), pp. 699–734.
- [Ton+05] A. Tonchev et al. “The high intensity  $\gamma$ -ray source (HI $\gamma$ S) and recent results”. In: *Nuclear Instruments and Methods in Physics Research Section B: Beam Interactions with Materials and Atoms* 241.1 (2005). The Application of Accelerators in Research and Industry, pp. 170–175. DOI: <https://doi.org/10.1016/j.nimb.2005.07.080>.
- [Ud15] E. Uberseder and R. J. deBoer. *AZURE2 User Manual*. JINA. 2015. URL: <https://usermanual.wiki/Document/azure2manual.1252390862.pdf>.
- [Uga+13] C. Ugalde et al. “First determination of an astrophysical cross section with a bubble chamber: The  $^{15}\text{N}(\alpha, \gamma)^{19}\text{F}$  reaction”. In: *Physics Letters B* 719.1-3 (Feb. 2013), pp. 74–77. DOI: [10.1016/j.physletb.2012.12.068](https://doi.org/10.1016/j.physletb.2012.12.068).
- [Wan+17] M. Wang et al. “The AME2016 atomic mass evaluation (II). Tables, graphs and references”. In: *Chinese Physics C* 41.3 (Mar. 2017), p. 030003. DOI: [10.1088/1674-1137/41/3/030003](https://doi.org/10.1088/1674-1137/41/3/030003).
- [Wei00] S. Weinzierl. “Introduction to Monte Carlo methods”. In: *Research School Subatomic Physics*. Amsterdam, Netherlands, June 2000. arXiv: [hep-ph/0006269](https://arxiv.org/abs/hep-ph/0006269).
- [Wel+09] H. R. Weller et al. “Research opportunities at the upgraded HI $\gamma$ S facility”. In: *Progress in Particle and Nuclear Physics* 62.1 (2009), pp. 257–303. DOI: <https://doi.org/10.1016/j.pnpnp.2008.07.001>.
- [WH07] S. Woosley and A. Heger. “Nucleosynthesis and remnants in massive stars of solar metallicity”. In: *Physics Reports* 442.1-6 (Apr. 2007), pp. 269–283. DOI: [10.1016/j.physrep.2007.02.009](https://doi.org/10.1016/j.physrep.2007.02.009).
- [WHW02] S. E. Woosley, A. Heger, and T. A. Weaver. “The evolution and explosion of massive stars”. In: *Rev. Mod. Phys.* 74 (4 Nov. 2002), pp. 1015–1071. DOI: [10.1103/RevModPhys.74.1015](https://doi.org/10.1103/RevModPhys.74.1015).
- [Won98] S. S. Wong. *Introductory Nuclear Physics*. A Wiley-Interscience publication. Wiley, 1998.
- [WW93] T. A. Weaver and S. E. Woosley. “Nucleosynthesis in massive stars and the  $^{12}\text{C}(\alpha, \gamma)^{16}\text{O}$  reaction rate”. In: *Physics Reports* 227.1-5 (May 1993), pp. 65–96. DOI: [10.1016/0370-1573\(93\)90058-L](https://doi.org/10.1016/0370-1573(93)90058-L).
- [Zim+13] W. R. Zimmerman et al. “Unambiguous Identification of the Second  $2^+$  State in  $^{12}\text{C}$  and the Structure of the Hoyle State”. In: *Phys. Rev. Lett.* 110 (15 Apr. 2013), p. 152502. DOI: [10.1103/PhysRevLett.110.152502](https://doi.org/10.1103/PhysRevLett.110.152502).
- [Zyl+20] P. A. Zyla et al. (Particle Data Group). “Review of Particle Physics”. In: *Progress of Theoretical and Experimental Physics* 2020 (2020). 083C01. DOI: [10.1093/ptep/ptaa104](https://doi.org/10.1093/ptep/ptaa104).