universität innsbruck

Fakultät für Technische Wissenschaften

3D Modelling and Transient Simulation of the Superconducting High Order Corrector Magnets for the High Luminosity Upgrade of the LHC

In collaboration with CERN

Daniel Mayr, BSc

Innsbruck, January 24, 2024

CERN-THESIS-2023-325
08/01/2024

Masterarbeit

verfasst im Rahmen eines gemeinsamen Masterstudienprogramms von LFUI und UMIT – Joint Degree Programme

eingereicht an der Leopold-Franzens-Universität Innsbruck, Fakultät für Technische Wissenschaften zur Erlangung des akademischen Grades

Diplomingenieur

Beurteiler:

Univ.-Prof. Dr. Clemens Zierhofer

Institut für Mechtronik
Arbeitsbereich für Signalverarbeitung und Hochfrequenztechnik

Betreuer:	Univ.-Prof. Dr. Clemens Zierhofer, Universität Innsbruck, Institut für Mechtronik, Arbeitsbereich für Signalverarbeitung und Hochfrequenztechnik
Mitbetreuer:	Dr. Emmanuele Ravaioli, CERN, Technology Department - Machine Protection and Electrical Integrity, Performance assessment and Electrical quality assurance

Acknowledgment

First and foremost, I wish to express my profound gratitude to my supervisor at CERN, Emmanuele. Not only has he imparted invaluable knowledge and wisdom, but his patience throughout this journey has been unwavering. Amongst a pool of many applicants, I'm immensely grateful for the trust he placed in me by selecting me for this unique opportunity to spend almost a year at CERN.
A heartfelt thank you goes to my supervisor at LFU, Prof. Zierhofer. His flexibility and support made it possible for me to undertake this thesis at CERN, an opportunity for which I am profoundly thankful.
I would also like to thank my colleagues, especially Marvin, Erik, Arjan, and Mariusz. Their readiness to answer questions, offer help, and share their expertise greatly enhanced my learning experience.
A special acknowledgement goes to the team at INFN. Their promptness in sharing data, simulation results, and knowledge was pivotal to the success of my research. Furthermore, their commitment to taking time out for meetings ensured a smooth and efficient collaboration. In particular, I would like to extend my deepest appreciation to Marco Prioli, Marco Statera, and Samuele Mariotto for their significant contributions.
Last but by no means least, my heartfelt thanks to my friends and family who visited me at CERN, adding joy and respite to my intense academic journey. Most importantly, I owe a world of gratitude to my girlfriend, Isabel, for her unwavering and unequivocal support in every aspect of my life.

Kurzfassung

In dieser Arbeit werden die supraleitenden, superferrischen, selbsgeschützten Korrektormagneten höherer Ordnung, die Teil des High Luminosity Upgrades des Large Hadron Colliders sind, modelliert und Quench Transienten Simulationen werden durchgeführt. Zu diesen Magneten gehören Sechspol-, Achtpol-, Zehnpol- und Zwölfpolmagneten, die von LASA (INFN) entworfen und hergestellt wurden. Die Aufgabe dieser Korrektormagneten ist es, die magnetischen Feld Fehler der fokussierenden Vierpolmagnete in den zwei Haupt-Kreuzungsbereichen im LHC (ATLAS und CMS) zu korrigieren.
Zur Quench-Protection Analyse wird das Python Framework STEAM-SDK verwendet, das am CERN entwickelt wird und mehrere Softwares zur (co-)simulation von supraleitenden Magneten durch einheitliche Modelle verbindet. Im Rahmen der Modellbildung wird die 3D "Racetrack"-Geometrie der betrachteten Magneten in das Framework implementiert und für alle STEAM tools verfügbar gemacht. Elektromagnetische und thermische Transienten in den Magneten werden mit STEAM-LEDET simuliert, das die 3D-Wärmediffusion entlang des Leiters und zwischen den Windungen, den ohmschen Verlust und den Kopplungsverlust zwischen den Filamenten mit der Finite-Differenzen-Methode modelliert.
Weiters wird STEAM-SDK hinsichtlich zweier Automatisierungsfunktionen weiterentwickelt. Zum einen sollen die Parameter aller Leiter automatisiert für jede Spule einzeln berechnet und im Modell angepasst werden. Zum anderen sollen auch alle auf ein Quench Event bezogenen Parameter auf Basis automatisiert erstellter Messungsdateien berechnet und im Modell angepassen werden. Mit dieser Automatisierung wird auch jede Spulen der HL-LHC HOC Magneten von den Design-Parametern an gemessene Werte angepasst.
Im Rahmen der Modellbildung müssen mehrere Annahmen getroffen werden, die durch geeignete Simulationen und Messungen validiert und begründet werden. Dabei werden spezifische Aspekte wie die Quench-Propagation zwischen Spulen, die Anpassung der Helium Kühlung oder die Approximation des Magnetschaltkreises behandelt.
Zur Validierung stehen elf Messungen aus Training-Quenches bei LASA mit verschiedenen Magneten und unterschiedlichen Stromstärken zur Verfügung. Da die initiale Quench Position eine unbekannte Größe bleibt, wird dafür eine zielgerichtete Parameterstudie durchgeführt. Außerdem wird das Auftreten von quench back untersucht und eine Analyse der simulierten hot-spot Temperaturen durchgeführt um die Effektivität der Quench-Protection Strategie zu evaluieren.
Für die verschiedenen Korrektormagneten kann ein einheitlicher Satz von Annahmen gefunden werden, bei welchen die Modelle prädiktive Simulationsergebnisse für alle Magneten und bei verschiedenen Stromstärken liefern, wobei nahezu alle Simulationsergebnisse in Bezug auf quench load innerhalb der Unsicherheiten der Quench Position liegen.

Abstract

In this work, the superconducting superferric self-protected high order corrector magnets that are part of the High Luminosity Upgrade of the Large Hadron Collider are modelled and quench transient simulations are performed. These magnets include sextupole, octupole, decapole, and dodecapole magnets that were designed and manufactured by LASA (INFN). The purpose of these magnets is to correct the magnetic field errors of the focusing quadrupole magnets in the two main intersection regions in the LHC (ATLAS and CMS). To perform a quench protection analysis, the Python framework STEAM-SDK is used, which is developed at CERN and connects several softwares for (co-)simulation of superconducting magnets by unified models. As part of the modelling, the 3D racetrack geometry of the considered magnets is implemented in the framework and is now available across various STEAM tools. Electromagnetic and thermal transients in the magnets are simulated using STEAM-LEDET, which models 3D thermal diffusion along the conductor and between turns, ohmic loss, and inter-filament coupling loss using the finite difference method. Furthermore, STEAM-SDK is further developed with regard to two automation functions. On the one hand, the parameters of all conductors are to be automatically calculated for each coil individually and adjusted in the model. Secondly, all parameters related to a quench event will be calculated and adjusted in the model on the basis of automatically created measurement files. With this automation, each coil of the HL-LHC HOC magnets will also be adjusted from the design parameters to measured values. Within the context of modelling, several assumptions have to be made, which are validated and justified by appropriate simulations and measurements. Specific aspects such as the quench propagation between coils, the adjustment of the helium cooling or the approximation of the magnetic circuit are addressed. Eleven measurements from training quenches at LASA with different magnets at different current levels are available for validation. Since the initial quench position remains an unknown quantity, a targeted parameter study is performed for this purpose. Furthermore, the occurrence of quench back is investigated and an analysis of the simulated hot-spot temperatures is performed to evaluate the effectiveness of the quench protection strategy. For the different high order corrector magnets, a consistent set of assumptions was found for which the models provide predictive simulation results for all magnets and at different current levels, with almost all simulation results being within the uncertainties of the quench position with respect to quench load.

Contents

List of Figures **ix**

List of Tables **xv**

1 Introduction **1**
1.1 About the project . . . 1
1.1.1 CERN[1] and its accelerator complex . . . 1
1.1.2 High-Luminosity Upgrade . . . 3
1.1.3 SM18 . . . 4
1.2 Superconducting magnets . . . 5
1.2.1 Superconductivity . . . 5
1.2.2 Superconducting cables . . . 8
1.2.3 Superconducting magnets and their applications . . . 9
1.3 Quench in superconducting magnets . . . 10
1.3.1 Causes and risks of quenches . . . 10
1.3.2 Quench protection systems . . . 12
1.4 Technical details of the HL-LHC HOC magnets . . . 16
1.5 Research objective and thesis structure . . . 19

2 Modelling and simulation **21**
2.1 STEAM framework . . . 21
2.1.1 Overview . . . 21
2.1.2 Implementation . . . 23
2.1.3 Simulation setup and minimal example . . . 23
2.2 STEAM-LEDET . . . 25
2.2.1 Finite difference and lumped element modelling . . . 25
2.2.2 Electro-magnetic model . . . 26
2.2.3 Thermal model . . . 27
2.2.4 Coupling current model . . . 29
2.2.5 Quench model . . . 30

3 Geometry modelling in STEAM **33**
3.1 3D racetrack geometry . . . 33
3.1.1 Defining parameters of a racetrack coil and magnet . . . 35

[1] Conseil européenne pour la recherche nucléaire (french for: European Organization for Nuclear Research)

3.1.2 Implementing coil geometry in FIQUS and generating .cond files 36
3.1.3 Dividing coil geometry from .cond files in conductor geometries 40
3.1.4 Generating PCB coil connection 43
3.1.5 Generating LEDET geometry input files 49
3.2 Setting Non-adiabatic parameters 54

4 Automatic test campaign validation 57
4.1 Introduction and motivation 57
4.2 Multiple changes of several variables: ParametricSweep 58
4.3 Automation of conductor specific simulation setup 59
4.3.1 Introduction of ParsimConductor step 59
4.3.2 Preparing and reading conductor data 60
4.3.3 Data processing 62
4.3.4 Applying the changes to the model and implementation 64
4.4 Automation of event specific simulation setup 65
4.4.1 Introduction of ParsimEvent step 65
4.4.2 Reading event data 66
4.4.3 Data processing 68
4.4.4 Setting up Viewer configuration files 70
4.4.5 Applying the changes to the model and implementation 71
4.5 Full test campaign analysis procedure 72

5 Identification of model assumptions 75
5.1 Introduction, conductor parameters adjustments and analysis of measurement data 75
5.2 Quench parameters 77
5.2.1 Quench initialisation 77
5.2.2 Quench location 78
5.3 Cooling of the conductor between coils 79
5.4 Magnet circuit and power supply 81
5.5 Epoxy impregnation 84
5.6 Cooling correction factor f_c 85
5.7 Setting U_d and finalising f_c 87

6 Simulated vs measured transients 91
6.1 Quench location analysis 91
6.2 Quench back 104
6.3 Peak temperature 105

7 Discussion and Conclusion 109

Bibliography 111

List of Figures

1.1 Schematic representation of the CERN accelerator complex, showing the particle accelerators, the used particles and the various experiments and detectors [11]. 2

1.2 Schematic schedule for the further development of LHC under the HL-LHC project, including collision energy and luminosity at project stage. [19] . . 4

1.3 Critical surface for the parameters temperature, magnetic field and current density below which the superconductor transitions to its superconducting phase 5

1.4 Timeline of observed critical temperatures of different superconducting materials. BCS-Superconductors (dark green circle), Heavy-fermions-based (light green star) [27], Cuprate (blue diamond) [28], Buckminsterfullerene-based (purple inverted triangle) [29], Carbon-allotrope (red triangle) [30], Iron-pnictogen-based (orange square) [31], Strontium-ruthenate (grey pentagon) [32], Nickel-based (pink six-point star) [33]. Figure reproduced from [34]. 6

1.5 Course of magnetic field lines inside superconductors of types I and II for different magnetic fields and temperatures [36] 7

1.6 various examples of wire-cross-sections of round superconducting wires. a. Nb-Ti filaments in copper matrix. b. Nb_3-Sn filaments in copper matrix. c. 20 328 Nb-Ti filaments in a copper matrix [37]. Figures a,b reproduced from:[38] 8

1.7 Example of a Nb-Ti Rutherford cable used in various accelerator magnets a. Top view (reproduced from: [40]). b. Sketch of the cross section and strand twist pitch (reproduced from: [41]) 9

1.8 Examples of cross-sections for different types of magnets. Colors represent the direction of the transport current, arrows indicate the generated magnetic field's orientation. a. Solenoid. b. Dipole. and c. Quadrupole. [14] 10

1.9 Illustration of heat generation and redistribution of current around a quench hot-spot. Sc...superconductor, Cu...copper (usual matrix material). [36] . . 11

1.10 Qualitative evolution of the temperature in a superconducting cable at an energy perturbation just below (blue curve) and just above (red curve) the critical energy. [45] 11

1.11 Quench heating strips, consisting of heating stations and copper conductor, attached to parts of an insulated coil of a short magnet model [58] 15

1.12 Circuit diagram of a magnet protected with Energy Extraction. Power supply consisting of voltage source with switch and parallel diode. L_M...magnet inductance, R_C...magnet and circuit resistance, EE Energy Extraction. Figures reproduces from [14] 16
1.13 Schematics depicting the cross section of the HL-LHC HOC magnets, showcasing the iron yoke in blue, superconducting coils in red, and the coil fastening mechanism in grey and brown [67, 66]. 17
1.14 Table showing the final design parameters of the HL corrector magnets [67, 66] 18
1.15 Simplified schematic of the protection circuit applied to the other corrector magnets, left to rigth: power converter, crowbar, magnet [79] 18

2.1 Schematic representation of the STEAM framework with all tools and input files used. Some functionalities are under active developement. *COMSOL license needed. **Commercial circuit solver from Cadence Design Systems. ***Free tools from Sandia Labs. Figure reproduced from [87] 22
2.2 Schematic representation of the mutual coupling and energy exchange among the three interconnected sub-networks (electrical, thermal, and transitory) in LEDET [14]. 26
2.3 Electric network used in LEDET to reproduce the electro-dynamic behaviour of a superconducting magnet and its surrounding components. Dashed elements can be optionally included for quench protection. [108] 27
2.4 Analogous thermal circuit diagram of a thermal element b used in the thermal lumped element sub-network of LEDET. Element b is exchanging heat with an exemplary element h coupled through $R_{b,h}$. [86] 28
2.5 Electric network modelling a single coupling current used in the coupling sub-network of LEDET [108]. 29

3.1 Technical drawing of the racetrack coil geometry of the MQSXF magnet including geometrical specifications. Figure reproduced from [114]. 33
3.2 Top, front and right side view of the old MCSXF geometry in LEDET displayed in 3D prespective. The color represents the length of the conductor along the electrical direction starting from blue to green then to yellow and ending in red. 34
3.3 UML diagram of the `RacetrackMagnet` class that is used to store all parameters related to the geometry and geometry processing of a magnet consisting of racetrack coils. 35
3.4 Schematic representation of the chosen geometrical parameters of the definition of a racetrack coil in STEAM-FiQuS. 36
3.5 Generated hexahedron models of every HL-LHC HOC magnet. Each coil is coloured differently. 39
3.6 Schematic drawing of a arbitrary concave quadrilateral in 3D space with different points needed for the designed subdivision algorithm 40
3.7 Frames number 1, 2, 3, 4, 6 and 11 (l.t.r.) of the subdivision of coil hexahedrons into conductor hexahedrons. Conductor starting point is 7 (see listing 3.2). 42

3.8 Two technical drawings of the PCB used to connect the individual coils of MCSXF at the rear end. On the left with dimensions, on the right the coil cross sections have been added in red and the cable simplification approximation in blue. Shown is the x-y plane with the z-axis running along the magnet. Figure reproduced from [117] and [114]. 43
3.9 Recording of the entry and exit points of the conductor of an MCOXF coil towards the PCB [122]. 44
3.10 Center points and numbered vertices of the hexahedrons start and end faces of an inner coil connection starting from the start hexahedron ending at the end hexahedron. 45
3.11 Center points and numbered vertices of the hexahedrons start and end faces of an outer coil connection starting from the start hexahedron ending at the end hexahedron. 46
3.12 Coil connection hexahedrons for inner and outer coil connection. Last element of first coil in blue, coil connection in green, first element of second coil in red. 47
3.13 Hexahedron geometry models created in FiQuS for the HL-LHC HOC magnets. Racetrack coils composed of red hexahedrons and coil connections composed of blue hexahedrons. 48
3.14 Top, front and right side view of the new MCSXF geometry in LEDET displayed in 3D prespective. The color represents the length of the conductor along the electrical direction starting from blue to green then to yellow and ending in red. 49
3.15 Absolute magnitude of the magnetic field B in each element of a 3D LEDET model of MCSXF at operating current (99 A) generated by the geometry input CSV file. B derived from different prior simulations. 51
3.16 Absolute magnitude of the magnetic field B in each element of the 3D LEDET models for the remaining HL-LHC HOC magnets at operating current created by the geometry input CSV file and derived from preceding OPERA simulations. 52
3.17 Cross section of an imaginary magnet with 2x3 windings to demonstrate different numbering formalities. ROXIE numbering in red, electrical order numbering in black, conductors in grey. 53
3.18 Transverse section of a MCSXF coil with insulation including geometrical specifications. Magnet aperture is at the bottom. Figure reproduced from [114]. 54
3.19 Different sized cross-section through MCTXF showing all conductors with masks for insulation parameters needed for non-adiabat simulations. 55

4.1 UML diagram of the `DataParsimConductor` class used to store conductor specific parameters 61
4.2 UML diagram of the `ParsimConductor` class used to run a `ParsimConductor` step 65
4.3 UML diagram of the `DataEventMagnet` class used to store event specific parameters 67

4.4 Cross secions of the magnet MQXFB02 with Quench Heater information needed to set up the dict_QH_circuits_to_QH_strips entry in the `ParsimEvent` step definition. 68
4.5 Example of a voltage discharge curve of a quench heater. High frequency measurement of a MBRD Heater in blue. Threshold measured trigger time and voltage in red. Approximated discharge curve in grey. Corrected trigger time and voltage in green. 69
4.6 UML diagram of the `ParsimEventMagnet` class used to run a `ParsimEvent` step 72
4.7 Schematic of a complete test campaign analysis using LEDET including output and input files. Sequence of steps used in the centre of the image. More detailed schematic of steps in coloured background. Filetype figures from [141, 142, 143, 144, 145, 146] 73

5.1 Time evolution of the temperature profile (temperature versus position) at the quench start location using the example of an MCSXF08 simulation with $T_{p,pulse} = 20\,\mathrm{K}$ and $w_{pulse} = 0.03\,\mathrm{m}$ in the case of a propagating quench. . . 77
5.2 3D CAD renderings from the construction drawings of the magnet type MCSXF showing. Figures reproduced from [150, 151]. 79
5.3 3D temperature distribution at four selected time points in two MCSXF08 simulations with high cooling of the conductor between coils enabled or disabled. Initial current is 110.375 A. 80
5.4 Electrical circuit diagram illustrating the test configuration used for HL-LHC HOC magnet measurements, detailing the interconnected components including the power converter (PC), the IGBT polarity switch, SCR diodes in a by-pass-crowbar, cable resistance R_c and the magnet (HOC magnet) with ammeter (ltr.). 82
5.5 Comparison of measurement No. 1 with corresponding simulation result with initial (ideal) and revised modelling of the power converter (PC). . . . 82
5.6 Comparison of measurement No. 1 with corresponding simulation result with revised modelling of the power converter (PC). 83
5.7 Photography of a scrapped MCTSXF cross-section showing single isolated cables in silver, epoxy in dark brown and coil insulation in light brown. . . 85
5.8 Simulation results with different f_c for corresponding measurement No. 10 showing the adjustment process of the starting time according to resistive voltage at 1 V. 86
5.9 Simulated resistance with different f_c values in comparison to corresponding measurement No. 10. 86
5.10 Measured resistive voltage smoothed with gaussian filters of different standard deviation σ to determine the quench start time $t_{q,s}$. 88
5.11 Comparison of relative quench load errors, mean relative error (MRE), mean absolute error (MAE) and mean squared error (MSE) of each measurement with high-field-location-simulations of different cooling correction factors f_c and excluded or imperfectly included U_d. 88
5.12 Comparison of measurement No. 1 with corresponding simulation result with the final simulation model. 89

6.1 Comparison of simulation results corresponding to measurement No. 1 showing the maximum, minimum and median values of 34 selected quench locations and high field location. 92
6.2 Comparison of simulation results corresponding to measurement No. 2 showing the maximum, minimum and median values of 34 selected quench locations and high field location. 93
6.3 Comparison of simulation results corresponding to measurement No. 3 showing the maximum, minimum and median values of 34 selected quench locations and high field location. 94
6.4 Comparison of simulation results corresponding to measurement No. 4 showing the maximum, minimum and median values of 34 selected quench locations and high field location. 95
6.5 Comparison of simulation results corresponding to measurement No. 5 showing the maximum, minimum and median values of 34 selected quench locations and high field location. 96
6.6 Comparison of simulation results corresponding to measurement No. 6 showing the maximum, minimum and median values of 34 selected quench locations and high field location. 97
6.7 Comparison of simulation results corresponding to measurement No. 7 showing the maximum, minimum and median values of 34 selected quench locations and high field location. 98
6.8 Comparison of simulation results corresponding to measurement No. 8 showing the maximum, minimum and median values of 34 selected quench locations and high field location. 99
6.9 Comparison of simulation results corresponding to measurement No. 9 showing the maximum, minimum and median values of 34 selected quench locations and high field location. 100
6.10 Comparison of simulation results corresponding to measurement No. 10 showing the maximum, minimum and median values of 34 selected quench locations and high field location. 101
6.11 Comparison of simulation results corresponding to measurement No. 11 showing the maximum, minimum and median values of 34 selected quench locations and high field location. 102
6.12 Boxplot of the relative quench load error for simulations with 34 selected quench locations corresponding to the measurements at hand. 103
6.13 2D temperature distribution of MCTSXF simulation with high field location (measurement no. 9) at time 0.41 s (close to the time where T_{cl} is maximum). 104
6.14 Simulated peak hot-spot temperatures for different locations for the simulated MCSXF and MCOXF events. 105
6.15 Simulated peak hot-spot temperatures for different locations for the simulated MCDXF, MCTSXF and MCTXF events. 106
6.16 Boxplot of the peak temperatures reached for simulations with 34 selected quench locations corresponding to the measurements at hand. 107

List of Tables

1.1 Overview of common quench protection systems, categorised according to their activation and their discharge location 13

4.1 Minimal example of an input table for the `ParametricSweep` step for two simulations with different initial current and different trigger times of the quench heaters. 58
4.2 Minimal example of an input table for the `ParsimConductor` step with two coils "c1" and "c2" for the magnet "mag1". Different parameters (diameter, RRR) and differen measurements (room temperature resistance, critical current) are given. 60
4.3 Minimal example of a input table for a `ParametricSweeper` step created automatically by the `ParsimConductor` step. Strand diameter, RRR and the CUDI1 parameter C_1 of different conductors are changed at once. 65
4.4 Minimal example of a QuenchDictionary input table for a `ParsimEvent` step. Initial temperature and Quench Heater properties for two measurements are recorded. 66
4.5 Minimal example of a input table for a `ParametricSweeper` step created automatically by the `ParsimEvent` step. Initial Temeprature and Quench Heater trigger time and voltage are changed for two events. 72

5.1 Overview of performed quench measurements including quench event information of the HL-LHC HOC magnets. 76
5.2 Key parameters of the finite difference models in LEDET. The simulation times refer to a machine with 32GB RAM with a 2.19 GHz processor. . . . 90
5.3 Overview of needed pulse height $T_{p,pulse}$ and width w_{pulse} of the Gaussian temperature distribution used to start a quench in the simulation for each measurement at hand. 90

Acronyms

Abbriviation	Explanation
AD	Antiproton Decelerator
ALICE	A Large Ion Collider Experiment
ATLAS	A Toroidal LHC Apparatus
CERN	Conseil européenne pour la recherche nucléaire (french for: European Organization for Nuclear Research)
CMS	Compact Muon Solenoid
HL-LHC	High Luminosity Large Hadron Collider
LHC	Large Hadron Collider
LHCb	LHC-beauty
SPS	Super Proton Synchrotron
PSB	Proton Synchrotron Booster
PS	Proton Synchrotron
Linac4	Linear accelerator 4
HOC	High Order Corrector
LEDET	Lumped-Element Dynamic Electro-Thermal model
INFN	Istituto Nazionale di Fisica Nucleare (italian for: National Institute for Nuclear Physics)
HTS	High Temperature Superconductors
LTS	Low Temperature Superconductors
MRI	Magnetic Resonance Imaging
CLIQ	Coupling Loss Induced Quench
IFCC	Inter-Filament Coupling Currents
ISCC	Inter-Strand Coupling Currents
LASA	Laboratorio Acceleratori e Superconduttività Applicata (italian for: Laboratory for Accelerators and Applied Superconductivity)
STEAM	Simulation of Transient Effects in Accelerator Magnets
SIGMA	STEAM Integrated Generator of Magnets for Accelerators
BBQ	BusBar Quench
PyBBQ	Python BusBar Quench
FiQuS	Finite Elements Quench Simulator
ProteCCT	Protection of Canted-Cosine-Theta
YAML	YAML Ain't Markup Language, originally: Yet Another Markup Language
STEAM-SDK	Simulation of Transient Effects in Accelerator Magnets - Software Developement Kit

RRR	Residual-Resistance Ratio
CSV	Comma-Separated Values
CI/CD	Continuous Integration and Continuous Delivery
UML	Unified Modeling Language
TDMS	Technical Data Management System
HTML	Hypertext Markup Language
CCT	Canted Cosine Theta
PCB	Printed Circuit Board
IGBT	Insulated-Gate Bipolar Transistor
SCR	Silicon Controlled Rectifier
FAT	Factory Acceptance Test

1 Introduction

1.1 About the project

1.1.1 CERN[2] and its accelerator complex

On September 29, 1954, the European Organisation for Nuclear Research CERN (French: Conseil européenne pour la recherche nucléaire) was founded by 12 European states (founding Member States) to build a central laboratory for nuclear and particle physics for Europe [1]. Today, CERN is one of the largest and most respected centres for scientific research and counts 23 Member States, 10 Associate Member States and several nations and international organisations with Observer status [2]. At CERN numerous experiments are carried out across diverse fields of physics: from the Standard Model to super-symmetry and from exotic isotopes to cosmic rays [3, 4].
Since the foundation CERN has made numerous groundbreaking discoveries and advancements in the field of particle physics [5] and engineering. Following is a list of the key achievements of CERN:

1. In 1983, the elementary particles mediating the weak nuclear interaction, the W^+, W^- and Z bosons, were discovered at CERN with the SPS[3]. This discovery, led by Carlo Rubbia and Simon van der Meer, provided empirical validation of the electroweak theory, which unifies electromagnetic and weak forces [6]. They were awarded the 1984 Nobel Prize in Physics.
2. In 2002, a small amount of anti-hydrogen atoms were successfully decelerated and stabilised for the first time at CERN by the AD[4], which had been installed two years earlier. In 2011, the experiment was able to stabilise anti-hydrogen for 1000 s, studying its properties more precisely [7].
3. In 2012, CERN announced the discovery of a new particle at 126 GeV by the ATLAS[5] and CMS[6] detectors. The discovered particle is the Higgs Boson, theoretically predicted in the 1960s, which plays a central role in the Standard Model of particle physics as part of the Higgs mechanism. This mechanism explains how some particles acquire their mass by interacting with the Higgs field, a ubiquitous invisible field. [8, 9]
4. In addition to the many physical discoveries, CERN also contributed significantly to many technological developments. For example, the world-wide-web was de-

[2] Conseil européenne pour la recherche nucléaire (french for: European Organization for Nuclear Research)
[3] Super Proton Synchrotron
[4] Antiproton Decelerator
[5] A Toroidal LHC Apparatus
[6] Compact Muon Solenoid

veloped by the software engineer Tim Berners-Lee in 1989 at CERN, originally to meet the demand for automatic sharing of information between researchers, universities and institutes all over the world [10].

Many of these breakthroughs were made possible mainly by the CERN accelerator complex, which was built and further developed over decades and is shown schematically in Figure 1.1.

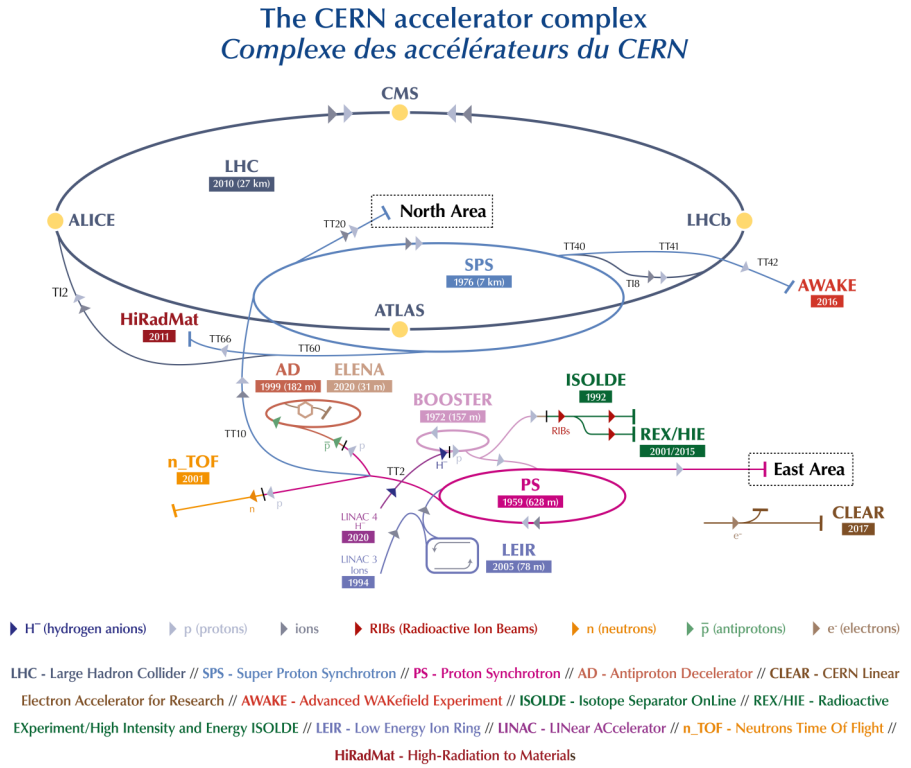


Figure 1.1: Schematic representation of the CERN accelerator complex, showing the particle accelerators, the used particles and the various experiments and detectors [11].

The accelerator complex consists of a chain of individual particle accelerators that accelerate particles to ever higher energies. The particle speeds become so high that a relativistic consideration is necessary and the relationship for kinetic energy $E_{kin} = 0.5 \cdot mv^2$ (E_{kin}[J]), where v[ms^{-1}] is the speed and m[g] the mass, no longer holds. The relativistic kinetic energy is given by: $E_{kin} = \frac{mc^2}{\sqrt{1-(\frac{v}{c})^2}} - mc^2$, where c [ms^{-1}] stands for the speed of light. This essentially has the consequence that an extremely large amount of energy must be applied for a relatively small increase in speed at very high velocities.
In the LHC[1], the last link and thus the most energetic part of this chain, particle beams are accelerated to up to 6.5 TeV per beam resulting in a velocity of 99.9999991% the speed of light. Linac4[2], CERN's proton source since 2020, linearly accelerates negative

[1] Large Hadron Collider
[2] Linear accelerator 4

hydrogen ions (H^-) to 160 MeV before feeding them into the PSB[1] (built in 1972). In the process, the electrons are removed from the hydrogen ions, leaving only protons. The protons are then accelerated by applying electric fields to 26 GeV and 450 GeV via the PS[2] (built 1959) and SPS[3] (built 1976), respectively, before entering the LHC rings. Here, the proton beams move in opposite directions in two separate circular beam pipes under ultra-high vacuum to avoid collisions with gas molecules. Targeted control of the accelerating electric fields ensures that the particles are not accelerated in a continuous stream, but in closely spaced bunches. The two tubes cross exactly 4 times in the 26.6 km long tunnel creating 40,000,000 collisions per second. The detectors ALICE[4], ATLAS, CMS and LHCb[5] are build around those four intersection points. The maximum collision energy at these detectors currently amounts to maximum 13.6 TeV [12].
In addition to protons, lead ions are also accelerated in the LHC. After acceleration in Linac3 and the Low Energy Ion Ring (LEIR), they follow the same path as protons [12].

1.1.2 High-Luminosity Upgrade

The two most commonly used metrics of an accelerator are luminosity and collision energy [13].
The goal of the accelerators is to reproduce circumstances similar to the beginning of our universe at the collision points. Due to the high energy density of the early universe, the collision energy of an accelerator is of central importance. For the collision energy roughly the relation $E \approx 0.3 \cdot R \cdot B$ is valid, where R stands for the radius of the accelerator ring and B for the magnetic flux density of the main dipoles bending the particle beam[6] [14]. So to increase the collision energy, the radius can be increased and stronger magnets can be developed.
Luminosity is a measure of how many particles pass through a certain area in a certain time in a particle accelerator. It has a significant influence on the probability of collisions, but does not correspond to the collision rate, since it only indicates how many particles enter per cross-section and time. Typically, luminosity is expressed in $\frac{1}{\mathrm{cm^2 s}}$ or $\frac{1}{\mathrm{fb}}$(often referred as inverse femtobar). A higher luminosity allows for more collisions and increases the chances of observing rare events such as the production of a Higgs particle or other new physical phenomena. Also, higher luminosity can help to distinguish between noise and potentially interesting signals. [13, 15]
The HL-LHC[7] project now aims to increase the integrated luminosity by a factor of a 5-7.5 (from 450 fb^{-1} to 3000 fb^{-1}) beyond the original design value of the LHC, thereby enabling the experiments to enlarge their data sample by one order of magnitude during the 12 years of HL-LHC operation compared with the LHC baseline programme [16]. The design phase started back in 2011, and in 2013 the development of the project was announced as a top priority in the European Strategy for Particle Physics [17], and in 2015 the prototyping and construction phase of the upgraded components began. All

[1] Proton Synchrotron Booster
[2] Proton Synchrotron
[3] Super Proton Synchrotron
[4] A Large Ion Collider Experiment
[5] LHC-beauty
[6] In section 1.2 the different magnets of an accelerator and their tasks are explained in more detail
[7] High Luminosity Large Hadron Collider

upgrades are expected to be completed by 2029, ushering in a new era of particle physics research [18, 19]. Figure1.2 shows a detailed schematic diagram of the schedule for the project. New parts will always be developed during the year and installed during the winter break.
Practically, the higher luminosity is achieved by making the particle beams more intense and focused, especially in the ATLAS and CMS regions. Several precautions are taken to achieve this [20]. Of central importance are stronger focusing magnets in the area of the detectors. For this reason, the inner triplet magnets, i.e. the magnets before and after the intersection points, are exchanged within the framework of the High Luminosity project. In the inner triplet region, quadrupoles are used that have the desired focusing effect. These will be replaced with Nb_3-Sn magnets as part of the High Luminosity project. Compared to the previously used Nb-Ti alloy, magnetic flux densities of up to 12 T instead of 8 T can now be achieved with the new quadrupole magnets [20]. To correct the errors produced by these new quadrupoles, several superferric magnets of higher order are needed. These magnets are the subject of this work. More information about them can be found in the section 1.4.

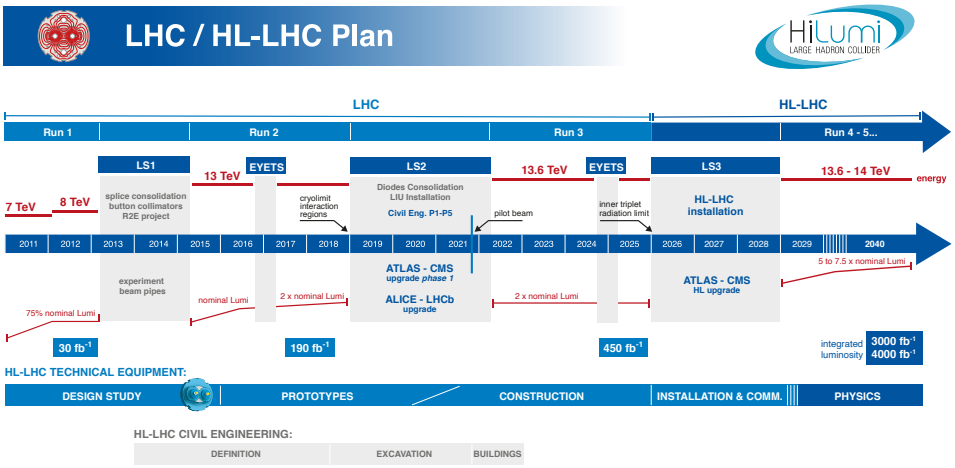


Figure 1.2: Schematic schedule for the further development of LHC under the HL-LHC project, including collision energy and luminosity at project stage. [19]

1.1.3 SM18

Before magnets are installed in the LHC tunnel, the complex magnets are tested under near-operational conditions. These tests are carried out in SM18, the CERN magnet and magnet circuit test facility. At SM18, originally built for the Large Electron-Positron Collider, more than 1700 magnets and radio frequency cavities[1] have been tested, including those for the LHC. In addition to evaluating magnets before they are deployed, the facility will be used to refine future prototype magnets and protections. The facility

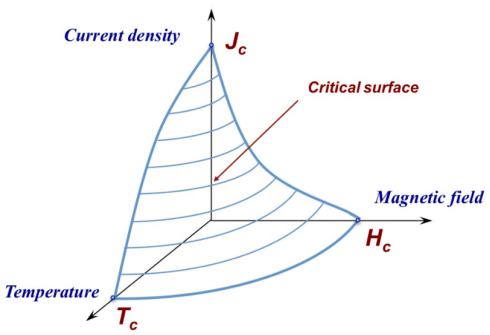
[1]RF cavities are metal chambers arranged along the accelerator that resonate at specific frequencies. They allow radio waves to interact with particle bunches for energy transfer, accelerating the particles.

has several vertical (for magnets up to 4 m in length) and six horizontal (for magnets up to 15 m in length) test stations, which can reach a maximum current of up to 30 kA and cryogenic temperatures ranging from 1.9 K to 80 K. The validation of systems, procedures, and hardware commissioning will be crucial milestones for the HL-LHC. [21]

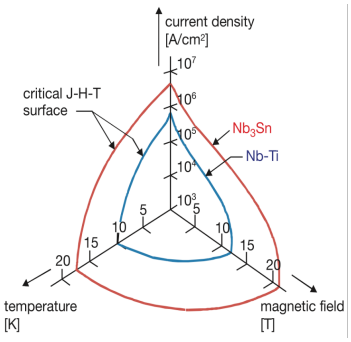
1.2 Superconducting magnets

1.2.1 Superconductivity

Three years after the Dutch physicist Heike Kamerlingh Omnes succeeded in liquefying helium and thus reaching temperatures of a few K, he carried out an experiment with mercury at extremely low temperatures. At temperatures of 4.1 K no resistance could be observed. Furthermore, Omnes discovered that certain other materials such as lead and indium below a certain critical temperature T_c no longer exhibit resistance. The phenomenon he discovered was superconductivity [22]. It was not until the 1950s and 1960s that a solid theoretical basis for this phenomenon could be defined - but only for part of it [23, 24].



(a) Typical critical surface [25]



(b) critical surface of Nb_3-Sn, Nb-Ti [26]

Figure 1.3: Critical surface for the parameters temperature, magnetic field and current density below which the superconductor transitions to its superconducting phase

During further experiments it was discovered that these superconducting properties depend on several factors, not only on the critical temperature T_c. Essentially, it is an interplay of mainly 3 physical quantities: temperature, magnetic flux density (magnetic field strength respectively) and current density. T_c is defined as the critical temperature at zero magnetic field and current density. In addition, there is also a critical magnetic flux density B_c and a critical current density J_c when the other quantities are assumed to be zero, respectively [24]. If this is plotted in three dimensions, with the axes representing these three quantities, a phase-transition-surface can be defined. This surface can be seen in Figure 1.3a. Below this surface, the material is in its superconducting state. For each material, this surface looks different. Looking at figure 1.3b, one can see the reason why the ductile metal alloy Nb-Ti is replaced with the inter-metallic compound Nb_3-Sn for the new inner triplet quadrupole magnets. This material can withstand higher magnetic fields before returning to its normal state. It also withstands this at higher current densi-

ties, which is also important for superconducting magnets to generate the high magnetic fields in the first place.
At this point, it should be mentioned that also pressure plays a role at the transition to the superconducting phase. Technically, however, this has no real relevance, as very high pressures are required for this effect, which are difficult to implement. For the sake of completeness, however, it will be briefly discussed in the following.
The greater the pressure, the earlier the entry into the superconducting state, meaning the Volume under the Surface in figure 1.3 increases [24]. This fact can also be seen in Figure 1.4, where various superconducting materials over time and their measured critical temperatures T_c are plotted. Indeed, particularly high T_c values could only be achieved under very high pressure (in the order of eleven Pa magnitudes). In the 1.4 figure, it can also be observed that the superconducting properties of Nb_3-Sn were discovered as early as the 1960s. The fact that superconducting magnets made of this material are only now being used at the LHC is testimony to the technical difficulties in manufacturing cables and magnets from this brittle material. Mote details can be found in the next subsection.

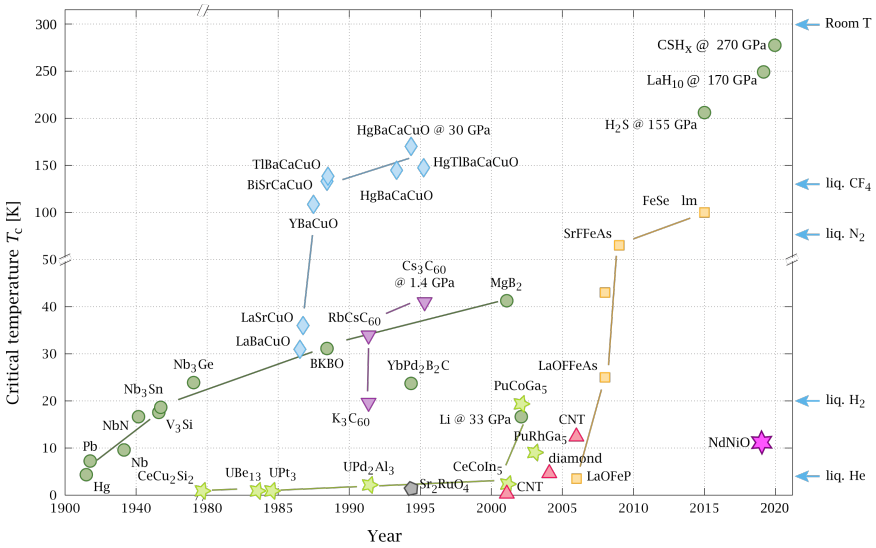
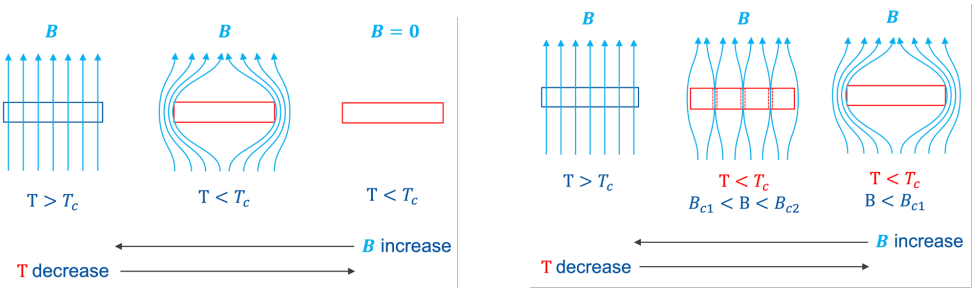


Figure 1.4: Timeline of observed critical temperatures of different superconducting materials. BCS-Superconductors (dark green circle), Heavy-fermions-based (light green star) [27], Cuprate (blue diamond) [28], Buckminsterfullerene-based (purple inverted triangle) [29], Carbon-allotrope (red triangle) [30], Iron-pnictogen-based (orange square) [31], Strontium-ruthenate (grey pentagon) [32], Nickel-based (pink six-point star) [33]. Figure reproduced from [34].

The simple critical surface shown before only accounts for type I superconductors. This applies to most metallic elements and T_c is typically in the range of a few K [24]. Type II superconductors have what is called a lower critical magnetic field B_{c1} and an upper critical magnetic field B_{c2}. Below B_{c1} type II superconductors behave identically to type I superconductors. This phase is called Meissner phase after its discoverer. In this phase a phenomenon can be observed in which the magnetic flux is excluded from the interior of the material and all the current flows in a surface layer of a thickness of about 0.1 µm [22] (Meissner effect). Above B_{c1} but below B_{c2} magnetic field lines

can penetrate the material in the form of so-called flux tubes or fluxoids (Schubnikow state). Thus, the material is not entirely in superconducting phase. If B_{c2} is exceeded, the material is no longer superconducting. Graphically, one can imagine this as if there were an upper and a lower critical surface with two different B_c. Typical materials for Type II superconductors are metal alloys, complex oxide ceramics or elements such as niobium, vanadium, technetium, diamond and silicon. Also Nb-Ti, which is used for the 1700 large magnets and the 8000 corrector magnets of the LHC, and Nb_3-Sn, the superconductor used for the novel quadrupole magnets of the HL-LHC, fall into this category [35]. In figure 1.5 the difference between type I and type II superconductors is shown again pictorially. In figure 1.5a it can be seen that in type I superconductors all magnetic field lines are excluded from the interior until T_c is intersected. On the other hand, in figure 1.5b one can see that type II superconductors can form magnetic field lines in the form of fluxoids in their interior when $B_{c1} < B < B_{c2}$ [36].



(a) Type I (b) Type II

Figure 1.5: Course of magnetic field lines inside superconductors of types I and II for different magnetic fields and temperatures [36]

Superconductors are often divided into HTS[1] and LTS[2]. Depending on the definition, a critical temperature limit of 30 K is used, below which one speaks of LTS and above which one speaks of HTS. A very simplistic distinction that is sometimes used is the required coolant. Accordingly, LTS must be cooled with liquid helium, while HTS can be cooled with liquid nitrogen (77 K), which is less expensive and complex [24].
As already mentioned in the introductory paragraph, the phenomenon of superconductivity is theoretically not yet fully understood. LTS are based on the formation of Cooper pairs, in which two electrons (belonging to the group of fermions) combine to form bosonic pairs. This occurs according to the BCS theory of Bardeen, Cooper and Schrieffer [23]. This pair formation and delocalization in the common quantum state prevents energy dissipation to the crystal lattice, allowing current to flow without resistance. Because the attractive phonon (vibrations in the lattice) force does not exist at higher temperatures, the BCS theory describes only the behaviour of LTS, also called conventional superconductors. Indeed, the theoretical limit of the BCS theory is around 30 K [36]. Because the attraction force between two electrons at high temperatures cannot be explained, HTS are also called unconventional superconductors.

[1] High Temperature Superconductors
[2] Low Temperature Superconductors

1.2.2 Superconducting cables

Technically used superconductors are mostly of type II, since type I superconductors typically only withstand magnetic fields of maximum 0.1 T. Due to this property, Type I superconductors are impractical for wire and cable applications [35]. Type II superconductors can also be used as cables in the mixed phase, so the applied magnetic field must be less than the upper critical magnetic field B_{c2}, which is typically in the range of several tens of T. In addition, J_c can be increased or decreased by certain manufacturing processes [36].
In contrast to T_c and B_{c2}, the height of J_c is not a material property but essentially dependent on the manufacturing process. According to Lorentz, the fluxoids, which form a magnetic field perpendicular to the movement of the current, experience a force which pushes them outwards (drift). Due to this flux motion $\frac{d\mathbf{B}}{dt}$, energy is dissipated. By incorporating impurities and resistive defects or other pinning centres[1] into the structure, the motion of the fluxoids is blocked. Strong density pinning centres are therefore very important for superconducting cables to allow high J_c. As soon as the Lorentz force, which depends proportionally on J_c, exceeds the pinning force, losses occur. Especially for accelerator magnets it is crucial to withstand high J_c as it can reduce the magnet volume and lower the cost [36].

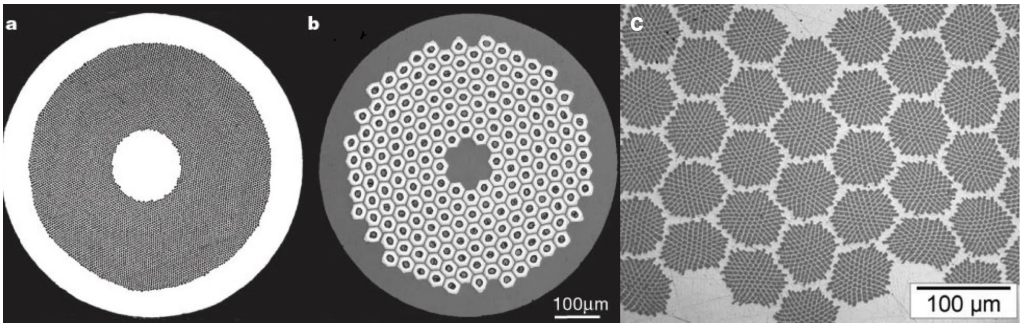


Figure 1.6: various examples of wire-cross-sections of round superconducting wires. a. Nb-Ti filaments in copper matrix. b. Nb_3-Sn filaments in copper matrix. c. 20 328 Nb-Ti filaments in a copper matrix [37]. Figures a,b reproduced from:[38]

In cable applications, LTS are usually realised in the form of thin filaments with diameters in the micrometer range (for Nb_3Sn tens of micrometer). This avoids not only magnetic field errors but also flux jumps. Since superconductors have a relatively high electrical resistance in their normal state, these filaments are placed in a matrix of a highly conducting material. Mostly this matrix consists of copper, but also aluminium or silver are used. Above T_c, therefore, the current flows through the matrix because it experiences less resistance there. Below T_c, the path with the least resistance goes through the superconductor itself. In addition to this advantage, the matrix material also conducts heat away from the superconductor and protects it from transitioning into the normal state. A wire can consist of hundreds to tens of thousands of filaments in a stabilising matrix. In figure 1.6 one can see exemplary cross sections of different wires. Figure 1.6a shows the cross section of an Nb-Ti wire, with the copper matrix in white and the

[1] examples would include precipitates in alloys, grain boundaries in inter-metallic compounds, doping in ceramic compounds

Nb-Ti in dark. Figure 1.6c shows the magnification of another Nb-Ti strand where the individual 4.8 µm filaments can be seen. Figure 1.6b shows the cross section of a Nb_3-Sn strand. There are usually no filaments in the centre of the cross section to allow their transposition. Filaments are twisted in most cables to reduce coupling losses, resulting in higher magnetic field uniformity and lower overall losses [39]. This is usually described with the parameter filament twist pitch length l_f [m].
Several such wires are often bundled into a cable to further prevent transitory losses and flux jumps. Also the conductor current capability is increased in that way. When wires are part of a cable they are called strands and the measure of their twist is called strand twist pitch length l_s [m]. There are different ways to align the strands and therefore different cable types have been proposed[1]. Because of his hight packing factor and good packing possibilities the most used cable type in accelerator magnets is the Rutherford cable. Figure 1.7a shows a top view of such a cable. Figure 1.7b shows the cross section of the cable and the twisting of the strands in three dimensions.

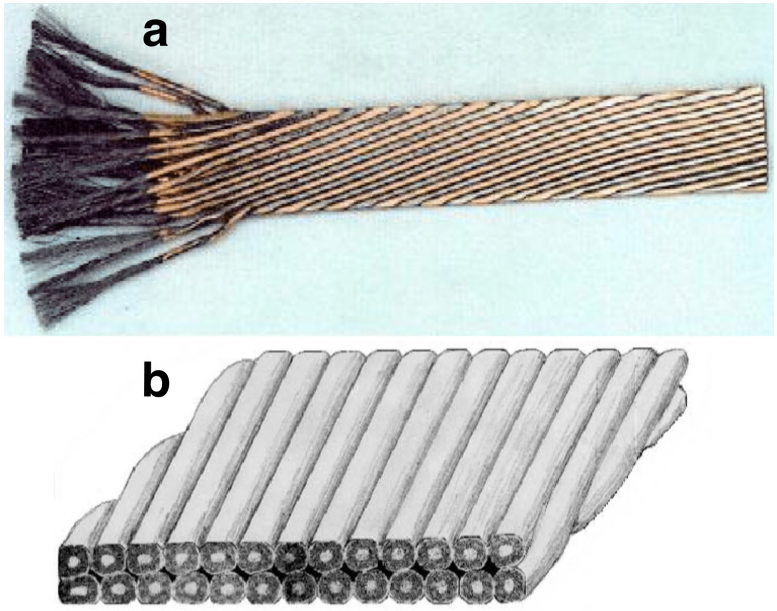


Figure 1.7: Example of a Nb-Ti Rutherford cable used in various accelerator magnets a. Top view (reproduced from: [40]). b. Sketch of the cross section and strand twist pitch (reproduced from: [41])

1.2.3 Superconducting magnets and their applications

Superconducting cables can be wound into conductor loops to act as electromagnets that can achieve magnetic fields far above the saturation of iron. These magnets are used primarily in high energy physics and medical applications, such as in MRI[2] technique. In MRI research, there are already systems that reach magnetic fields above 10 T [42, 43].

[1] Some examples include: Roebel cables, nuclotron cables, and cable-in-conduit conductors [14]
[2] Magnetic Resonance Imaging

There are different designs for such magnets, which differ in how the superconductor is wound. The simplest design of such a magnet would be a solenoid, which has the geometry of a coil. This is shown in figure 1.8a. These magnets generate a very uniform magnetic field and are able to generate very high magnetic fields. The problem however is that the field is parallel to the aperture and not perpendicular, as would be needed for accelerator applications. For that reason and due to their compactness dipole magnets (shown in Figure 1.8b) are used in the LHC [14]. These magnets perform the task of bending the particle beam and keeping it on the approximate circular path. Besides the 1232 dipole magnets installed, there are also 392 main quadrupole magnets in the LHC. A simple quadrupole design is shown in figure 1.8c. A combination of these magnets shifted in space focuses the beam especially near the intersection points to allow more collisions.

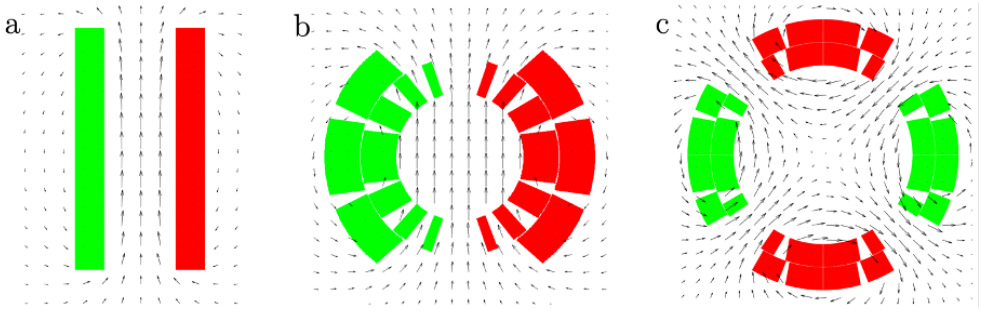


Figure 1.8: Examples of cross-sections for different types of magnets. Colors represent the direction of the transport current, arrows indicate the generated magnetic field's orientation. a. Solenoid. b. Dipole. and c. Quadrupole. [14]

Since the focusing quadrupole magnets and the deflecting dipole magnets are not ideal, several HOC[1] magnets are used for field correction. These magnets are called HOC magnets and can come in various harmonic types, such as (skew) quadrupole, sextupoles, octupoles, decapoles, and dodecapoles. Together with the HOCs, a total of about 10 000 superconducting magnets are installed in the LHC [44].

1.3 Quench in superconducting magnets

1.3.1 Causes and risks of quenches

Superconducting materials can experience a phenomenon called quench, where parts of the superconducting material suddenly and irreversibly transition to the normal state because the conditions introduced in previous sections are locally no longer satisfied (see figure 1.3a). The transition to the normal conducting state of a small part of the superconductor develops a resistance in the superconducting filaments. This initial quench location is called a quench hot-spot. The resistive zone in the superconductor redistributes the current within the strands: the stabilising matrix, normally made of copper, now exhibits the lower resistance than the superconducting material leading the matrix to carry the current. This process is shown schematically in figure 1.9 and is called the current sharing regime, which is reached at the current sharing temperature T_{cs} [K]. The

[1] High Order Corrector

temperature in the affected region rises further due to energy dissipation caused by the ohmic loss in the copper matrix, bringing even larger parts of the superconductor above the critical conditions.

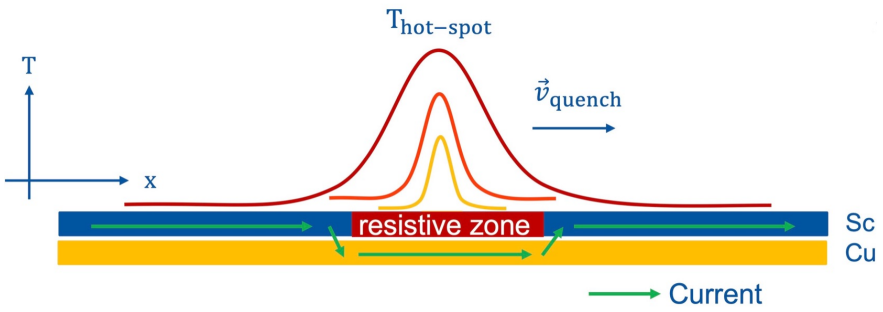


Figure 1.9: Illustration of heat generation and redistribution of current around a quench hot-spot. Sc...superconductor, Cu...copper (usual matrix material). [36]

For a quench to occur and propagate, a large enough pertubation is needed. With small pertubations, the superconductor can recover and reverse the momentary transition to the non-superconducting phase. The blue curve in figure 1.10 represents this case schematically. However, if the perturbation exceeds the critical energy density, which is essentially dictated by the conductor and its operating condition, thermal runaway occurs. The quench spreads throughout the entire magnet, leading to its discharge [45]. The red curve in figure 1.10 represents this case schematically.

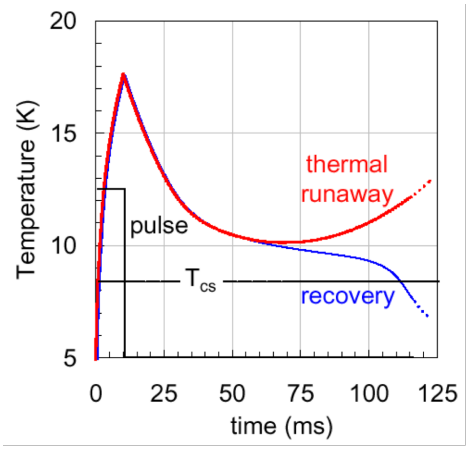


Figure 1.10: Qualitative evolution of the temperature in a superconducting cable at an energy perturbation just below (blue curve) and just above (red curve) the critical energy. [45]

In the LHC, the cooling is done with liquid helium. Depending on the magnet, the helium bath has a temperature of 1.9 K or 4.2 K. At 1.9 K, helium is in superfluid phase. In this phase helium has a very large thermal conductivity and wets very well due to its viscosity of zero [46].
Very small energy densities, typically in the range of a few $\mu\frac{J}{mm^3}$, are sufficient to initiate

a quench. The occurrence of quenches is therefore part of normal operations and must be considered in magnet design [14]. Causes for such low energy densities can be disturbances of various kinds: flux jumps, mechanical events such as conductor movements or cracks in the epoxy resin, electromagnetic transients due to hysteretic losses or eddy current losses, heat loads for example due to radiation, heat leaks, but also operational events such as power cuts, vacuum leaks or coolant loss [47, 48].
The propagation of a quench occurs longitudinal along the conductor and transversal between the conductors of a coil. Transversal propagation depends mainly on the thickness of the insulation layer. For longitudinal propagation, a propagation velocity v_q [$\frac{\mathrm{m}}{\mathrm{s}}$] can be defined, as shown in 1.9. Under adiabatic conditions the following relation is valid for this velocity [49]:

$$v_q = \frac{J}{\bar{c}} \cdot \sqrt{\frac{2 \cdot p \cdot \kappa}{T_{cs} + T_c + 2T}} \tag{1.1}$$

Where J [$\frac{\mathrm{A}}{\mathrm{m}^2}$] represents the current density in the insulated cable, p [m] symbolises its perimeter, κ [$\frac{\mathrm{W}}{\mathrm{Km}}$] denotes the thermal conductivity, $\bar{c}$ [$\frac{\mathrm{J}}{\mathrm{Km}^3}$] is the volumetric specific heat capacity of the volume. Typical values for v_q in Nb-Ti wires at LHC operating conditions range from 10 $\frac{\mathrm{m}}{\mathrm{s}}$ to 30 $\frac{\mathrm{m}}{\mathrm{s}}$ [50].
With currents up to 16.23 kA, magnetic fields up to 12 T are induced in superconducting magnets in the HL-LHC. Due to the large magnetic field energy stored in superconducting magnets, quenches can be a major risk as much of this energy is converted to heat. Some potential causes of damage are listed below:

- The maximum temperature in the hot spot can become so great that the insulation melts or breaks down or the conductor itself is damaged.
- Thermal energy can heat the helium until it evaporates. The resulting pressure can cause structural damage.
- The fact that certain parts of the magnet are heated faster than others can cause thermally induced mechanical stresses between these parts.
- An inhomogeneous transition to the non-superconducting phase can generate high voltages between coils and ground. These voltages are then discharged in the form of short circuits or even arcs and can damage insulation layers.

To demonstrate how much magnetic field energy is involved, a long LHC main dipole is considered. This magnet has a length of 15 m and stores about 8 MJ of energy. This is enough to melt 1.5 dm^3 of copper or to evaporate about 10cm of a LHC dipole coil [50]. This main dipole is installed in the LHC with several other magnets in series forming a chain of 154 magnets. If a quench would occur in only one magnet and if no protection systems are in place, the total field energy of the magnets of the circuit would be dissipated in this one resistive magnet. The total magnetic field energy of the circuits is about 1 GJ.

1.3.2 Quench protection systems

Due to the risks associated with quenching (see previous subsection), it is important to set up protection systems for superconducting magnets. In this context, the hot-spot temperature is of central importance. Since this causes many of the previously introduced risks, this value should be within a certain limit in order not to damage the magnet. For this limit different values can be found in the literature. Conservative studies recommend a limit of 100 K to 150 K for series-production magnets, which may be exceeded to about

300 K only in exceptional cases and R&D models [51]. Most present, however, is the limit of 300 K [52] or room temperature [53]. For Nb_3-Sn magnets even a maximum allowed limit of 350 K [53] was applied. Due to this evidence, in this thesis 300 K is taken as the maximum allowable hot-spot temperature for Nb-Ti magnets.
To keep the hot-spot temperature as low as possible, the field energy stored in the magnet must be dissipated as quickly as possible at another location and not locally at the hot-spot itself. This can be achieved by dissipating the energy in a controlled manner via another resistance. Quench protection systems can be divided into internal and external discharge mechanisms, where this resistance is located inside or outside the magnet itself.
Another possible subdivision of quench protection systems is the type of activation. Active systems must be triggered, for which a quench must first be detected. A simple way to detect a quench in a magnet coil would be to monitor the voltage across individual areas of the coils. As shown in figure 1.9, the resistance increases while quenching due to the copper carrying the current, which again leads to an increased voltage in this area. Passive systems, on the other hand, activate themselves and do not require measurements, which are associated with delays and inaccurecies. A simple example of a passive quench protection system would be a resistive diode that switches when the resistance in the magnet exceeds a certain value.
Table 1.1 summarises the most common quench protection methods [14] and categorises them according to the classification just introduced. The different systems are explained in the following.

	Internal discharge	**External discharge**
Passive	Self-protection	By-pass elements Coupled secondary coils
Active	Quench Heater Coupling Loss Induced Quench (CLIQ)	Energy Extraction

Table 1.1: Overview of common quench protection systems, categorised according to their activation and their discharge location

Self-protection

Magnets can be designed in such a way that a quench propagates so quickly that it does not damage the magnet. For this, the longitudinal and transverse heat conduction must be sufficiently large and the cooling of the helium bath must be sufficiently designed. Thus a quench leads to a fast discharge of the magnet before the hot-spot temperature can reach the critical limit of 300 K. Therefore, the magnetic field energy is rapidly dissipated in the whole magnet and not only locally at the hot-spot.
This oversizing achieves very safe and reliable quench protection at low cost, but these magnets are very large because only low current densities can be set [54]. Since compactness is particularly important for magnets used in underground areas, this quench protection system is not used for accelerator magnets but rather for other small magnets.

By-pass elements and coupled secondary coils

By-pass elements, such as diodes or resistors, are connected in parallel to the magnet. If there is a resistance in the magnet, part of the current flows through these by-pass elements, so that part of the magnetic field energy is dissipated outside. In most cases simple protection diodes are used, but also more complex by-pass structures consisting of several resistors and single or back-to-back diodes are used [55].
This method is also very reliable, since it relies only on the fact that the resistance of the magnet increases, which is unavoidable during a quench. However, the disadvantage is that the by-pass elements change the behaviour of the circuit, since leakage currents flow through these elements when magnetic field is built up or dissipated. Diodes can reduce these effects, but they lead to a limitation of the maximum current change $\frac{dI}{dt}$ in the magnet [14]. Therefore, the design of the resistor is a trade off between effectiveness of quench protection (resistance as large as possible) and reduction of loss in operation (resistance as small as possible).
Instead of electrically connecting an external resistor to the magnet, a connection can also be achieved magnetically with a coupled secondary coil. A change of the current $\frac{dI}{dt}$ in the main magnet induces a current in the secondary coil, which is converted into heat by ohmic resistance in the secondary coil, thus outside the main magnet. In accelerator applications this method is hardly used, because very large secondary magnets, which again lead to transitory losses, are needed for good protection.

Quench Heaters and CLIQ[1]

Quench Heaters [56, 57] are strips of resistive material that are attached to the magnet. When a quench is detected, a current is sent through these strips from a power supply. The strips heat up due to their designed high ohmic resistance and conduct this heat through the outer insulation layer to large parts of the magnet. This causes the parts of the magnet that are connected to the quench heaters to go into the non-superconducting phase and become resistive. The magnetic field energy is thus dissipated in a controlled manner in large areas of the magnet.

In practice, only parts of the entire magnet are heated, so quench heaters usually consist of resistive heating stations made of stainless steel connected by well-conducting copper surfaces. These strips are then bonded to the surface of the magnet, as shown in 1.11. Particularly efficient but not very often used due to maintenance issues is the placement of the heating stations between coils. With the distance of the heating stations the discharge velocity can be influenced. The local quenches forced by the heating stations propagate with v_q along the two wire directions. The closer the heating stations are to each other, the faster the discharge. According to the equation 1.1, the greater the current density, the faster the discharge.
The power supply must immediately provide as large a current as possible. For this reason single capacitors or capacitor banks are used as power supplies. By default, a series connection of two identical parallel circuits of three 4.7 mF capacitors is used in the LHC. In total, this arrangement has a capacitance of 7.05 mF and with a nominal operating voltage of $U_{nom} = \pm 450\,\mathrm{V}$, this power supply can provide an energy of up to

[1] Coupling Loss Induced Quench

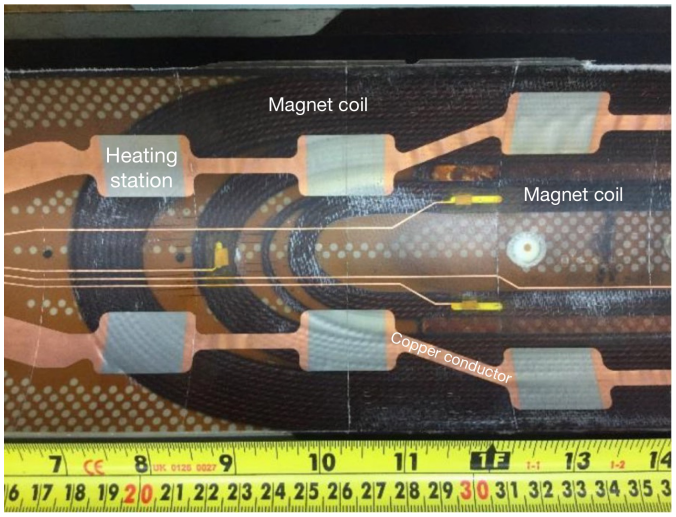


Figure 1.11: Quench heating strips, consisting of heating stations and copper conductor, attached to parts of an insulated coil of a short magnet model [58]

$E = 2.86\,\text{kJ}$ to the quench heaters [59, 60].
Quench heaters are very compact and are therefore often used in accelerator magnets. However, the disadvantage is that a quench detection system is needed that causes delays and can be corrupted. In the LHC this delay is about 10 ms [61], the HL upgrade will reduce this to below 1 ms [62].
A similar and still relatively new quench protection system is CLIQ [63, 14]. Conductors of Superconducting magnets are inherently subject to magnetic field changes. Due to their twist, these changes induce currents between individual filaments in a strand (IFCC[1]) and also between the individual strands in a cable (ISCC[2]). With a capacitor-thysistor design, sufficiently large coupling currents can be generated quickly and for a short time so that their joule losses are sufficient to quench large parts of the magnet in a controlled manner. Discharging the capacitor after quench detection thereby leads to oscillations in the current through the conductor, which in turn causes strong changes in the magnetic field. These changes cause the coupling currents leading to the desired loss. [14]

Energy Extraction

Energy extraction systems use an external ohmic resistor R_{EE} [Ω]. When a quench is detected, a switch flips to direct the current flow across R_{EE}. This dissipates some of the magnetic field energy of the magnet in the resistor. Figure 1.12 shows a simple energy extraction circuit.
Assuming that the self-inductance of the magnet and R_{EE} remain constant, the current discharges following an exponential curve. The time constant for this discharge τ_{EE} [s] can be calculated as follows:

$$\tau_{EE} = \frac{L_{tot}}{R_{EE}} \tag{1.2}$$

[1] Inter-Filament Coupling Currents
[2] Inter-Strand Coupling Currents

where L_{tot} is the total inductance of the circuit.
Energy extraction is another protection system often used in accelerator magnets. Like other active systems, it has the advantage that it is very compact and can extract a lot of energy from the system right from the start.
In order for as much energy as possible to be dissipated outside the magnet and for this to happen as quickly as possible, the greatest possible resistance R_{EE} is required (see equation 1.2), but this can lead to another problem. During a discharge, the magnet and R_{EE} are connected in series and form a closed circuit. The maximum voltage on R_{EE} is therefore equal to the maximum voltage on the magnet. These two effects must be taken into account in the design. In accelerator magnets, R_{EE} is in the range of some mΩ.

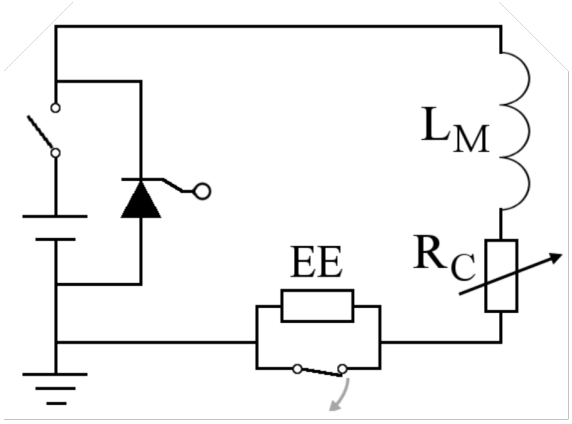
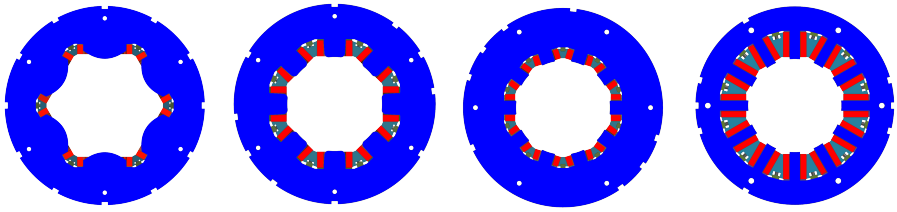


Figure 1.12: Circuit diagram of a magnet protected with Energy Extraction. Power supply consisting of voltage source with switch and parallel diode. L_M...magnet inductance, R_C...magnet and circuit resistance, EE Energy Extraction. Figures reproduces from [14]

1.4 Technical details of the HL-LHC HOC magnets

For the HL-LHC, 54 corrector magnets of different harmonic order are needed: from quadrupole to dodecapole [64]. Specifically, these are the following magnets: MQSXF, MCSXF, MCOXF, MCDXF, MCTXF. The naming convention for these magnets is as follows: M for magnet, C for corrector, Q/S/O/D/T indicating the number of poles (QS/SS/OS/DS/TS indicating that they are short magnets), X representing their position near the intersection, and F, which stands for focusing. High order corrector (HOC) magnets are those with more than four poles, thus all of the above except MQSXF. The new generation of HOC magnets is designed to achieve magnetic fields between 1.5 T and 3.6 T. Since prototypes and real magnets already exist, the HOC magnets are in terms of manufacturing at the most advanced stage of any magnets in the HL project [65].
The first conceptual design of these magnets has been published in 2015 [66]. In the design phase, the main focus was on facilitating series production, which is why a superferric design was chosen. The magnets are built modularly from relatively simple coils and Nb-Ti, which is already widely used in industry, is used as superconductor. A cross-section of the magnet design is shown in Figure 1.13. The iron yoke is constructed from hot-rolled, 5.8 mm thick plates. The laminations are manufactured with laser cutting and

subsequent electrical discharge machining to meet the required tolerances in the range of 10 µm. The same material is also used for the dipoles of the LHC and represents the state of the art for a magnetic iron [66]. For the insulation and impregnation of the coils, different materials are used, such as S-2 fibreglass and G11, which must be able to absorb the radiation of 25 MGy expected over the lifetime of the magnet [66].



(a) MCSXF (b) MCOXF (c) MCDXF (d) MCTXF

Figure 1.13: Schematics depicting the cross section of the HL-LHC HOC magnets, showcasing the iron yoke in blue, superconducting coils in red, and the coil fastening mechanism in grey and brown [67, 66].

INFN[1] was commissioned to produce a short model prototype for each of the corrector magnets. In 2016, the first coil for MCSXF was produced and tested stand alone [68]. One year later, MCSXF could be fully tested [69] and based on this, the design could be iterated [70]. The LASA[2] laboratory at INFN simultaneously fabricated and tested an MCOXF, an MCDXF and an MCTXF prototype [71, 72, 73]. Subsequently, first magnetic measurements [74] and power test [75] with the MCSXF prototype were performed at CERN. After completion of the test phase for this magnet [76], it was then possible to work on optimisations for series production [77, 78] and start series production together with industry. The first batch of magnets was already delivered to CERN and tested under operational conditions. Field qualities were within the specifications and magnetic fields at nominal current were even 1-3% larger than the calculated values [64].
The final design values of the corrector magnets are summarised in table form in Figure 1.14. The 54 corrector magnets are combined into a total of six cold masses, called corrector packages. This is a novelty for superferric magnets in a collider [64].

In the final design, the HOC magnets are self-protected. In the first prototypes produced, spot heaters were installed for R&D purposes, which showed that quench heaters are not necessary [66]. The quench will be detected by the current control loop of the power converter in maximal 3 time-steps, which corresponds to 60 ms to 180 ms according to the sample rate. The power converter will be turned off and the magnets are discharged on the crowbar resistance. An additional dump resistor is only put into place for the skew quadrupole. Measurement validation at LASA showed a maximum quench detection time of 180 ms [79]. A simplified schematic of this protection system can be seen in figure 1.15.

[1] Istituto Nazionale di Fisica Nucleare (italian for: National Institute for Nuclear Physics)

[2] Laboratorio Acceleratori e Superconduttività Applicata (italian for: Laboratory for Accelerators and Applied Superconductivity)

		MQSXF	MCSXF/MCSSXF	MCOXF/MCOSXF	MCDXF/MCDSXF	MCTXF	MCTSXF
Order		2	3	4	5	6	6
Aperture	(mm)	150	150	150	150	150	150
Integrated strength[1]	(T m)	0.700	0.095	0.069	0.037	0.086	0.017
Coil length[2]	(mm)	457	192	172	172	498	123
Magnet length	(mm)	614	254	233	233	575	200
Magnetic length	(mm)	401	168	145	145	469	99
Gradient	(T/m^{n-1})	34.8	224	3680	40480	585600	550400
Number of apertures		1	1	1	1	1	1
Number of circuits		1	2	2	2	1	1
Units needed		4	8	8	8	4	4
Spares		2	4	4	4	2	2
Cable data							
Strand diameter	(mm)	0.700	0.500	0.500	0.500	0.500	0.500
Insulation thickness	(mm)	0.070	0.070	0.070	0.070	0.070	0.070
Cu/No_Cu		2.1	2.1	2.1	2.1	2.1	2.1
Coil design							
Material		Nb-Ti	Nb-Ti	Nb-Ti	Nb-Ti	Nb-Ti	Nb-Ti
N. turns/pole		754	288	372	228	432	432
Cable length/pole	(m)	1230	104	105	59	439	112
Operational parameters							
Coil peak field	(T)	3.6	2.23	2.09	1.63	1.57	1.5
Temperature	(K)	1.9	1.9	1.9	1.9	1.9	1.9
Current	(A)	174	99	102	92	85	84
j overall[3]	(A/mm^2)	314	308	317	286	264	261
Loadline fraction		0.44	0.31	0.31	0.26	0.27	0.27
Differential inductance	(mH)	1530	213	220	120	805	177
Stored energy	(kJ)	30.8	1.72	1.55	0.668	3.627	0.732

Figure 1.14: Table showing the final design parameters of the HL corrector magnets [67, 66]

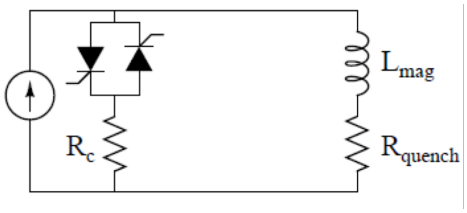


Figure 1.15: Simplified schematic of the protection circuit applied to the other corrector magnets, left to rigth: power converter, crowbar, magnet [79]

Only a few quench simulations were performed with these magnets. In the design phase, some simulations were carried out with the very unfavourable assumption of no quench propagation [66]. Further much more accurate simulations were then carried out by INFN [79, 80]. These simulations were performed as co-simulations of two different tools using the Steam framework [81], which is discussed in more detail in section 2.1. For the thermal domain QLASA [82] was used to simulate the evolution of the quench in the magnet coils and for the electrical domain LTSpice [83] was used to simulate the evolution of the circuit voltages. This also allowed the non-linear behaviour of inductance and current due to the saturation of iron to be modelled. However, the influence of a quench location was neglected, a scaling factor for the propagation velocity was introduced as a free variable and the resin distribution was considered homogeneous [79]. Therefore, the zone with the highest magnetic field was used as quench location. Furthermore, these simulations were adiabatic and quenches can only propagate within one coil and cannot transfer to another coil. Nevertheless, validations with measurements showed good agreement. In the simulations, the parameters hot-spot temperature and voltages to

ground remained within safe values even when the power supply is not switched off [79]. Based on these simulations, a non-adiabatic simulation with variable calculated propagation velocity is performed in this thesis.

1.5 Research objective and thesis structure

In this thesis, the superconducting HOC magnets of the HL-LHC are modelled using CERNs simulation framworke STEAM-SDK [81, 84] with LEDET[1] [85, 86]. Prototypes of these magnets, manufactured by INFN are tested in SM18 and based on this measurement data the conducted simulation are evaluated and further improved. In addition, the simulation tool STEAM-SDK will be further developed, especially with regard to automation.
This thesis is structured as follows:

- Chapter 1 provides an overview of the basic principles related to CERN, superconductivity, superconducting magnets, quench and quench protection. This section also introduces the HL-LHC HOC magnets and their technical details.
- Chapter 2 explains the details of the modelling, along with a comprehensive explanation of the operational mechanisms underlying the utilised finite difference software.
- Chapter 3 outlines the methodologies employed in implementing and approximating the geometry of the HL-LHC HOC magnets.
- Chapter 4 depicts the implemented automation framework for adjusting all event and conductor specific parameters with new steps implemented in STEAM-SDK.
- Chapter 5 addresses the necessary assumptions due to missing information and limitations of the used simulation software, including the justification and validation of these assumptions through targeted simulations and measurements.
- Chapter 6 presents the simulation results and compares them with corresponding measurement results. In addition, the influence of the unknown quench location is investigated.
- Finally, in chapter 7 the results are discussed and a conclusion is drawn.

[1] Lumped-Element Dynamic Electro-Thermal model

2 Modelling and simulation

2.1 STEAM framework

2.1.1 Overview

Modelling and simulating superconducting magnets is a complex task. A superconducting cable alone can consist of strands, which in turn can consist of filaments, both of which are twisted, can have different geometric dimensions and can be made of different materials. In addition to different cable types, different magnet types, such as simple solenoid or pancake coils or more complex cos-theta or canted cos-theta magnets, must also be modelled. In addition to the possible complex geometries, various quench protection systems are also built into the magnets and magnets themselves can either be stand-alone or built into more complex circuits. In addition, different fields of physics, such as electromagnetism, mechanics and thermodynamics, are needed in combination. Moreover, for different magnet designs, different numerical approaches are more suitable than others. Hereby, not only the time, but also the geometry must be discretised. Essentially, there are two modelling approaches for geometry discretisation: finite element method and finite difference method. When modelling, a compromise must always be made between computational effort and simulation accuracy.
Because modelling is such a complex and multifaceted problem, there is not one simulation tool that combines all these features, but several different tools are used. To guarantee consistency and repeatability between the different tools, the STEAM[2] framework [81, 84] has been developed as CERN. The aim of this framework is to "Achieve specialised, trusted, consistent, repeatable and sustainable software tools and models for rapid Simulation of Transient Effects in Accelerator superconducting Magnet circuits." [87]. The various tools used in the background and the models are validated, maintained and further developed by the STEAM team. Figure 2.1 shows the structure of the STEAM framework schematically. In the centre the three levels of detail and below them the tools used for each level can be seen: Conductor, Magnet and Circuit. STEAM is used by many renowned research institutions around the world, such as the Fermi National Accelerator Laboratory, the Lawrence Berkeley National Laboratory, INFN, ect.
The following is a brief overview of some of the simulation tools:

- **BBQ**[3] [88] is used to simulate 1D+1D quench propagation simulations in superconducting busbars using a COMSOL [89] finite element model. Heat flow can be modelled along the conductor and also through the insulation at certain nodes.

[2] Simulation of Transient Effects in Accelerator Magnets
[3] BusBar Quench

- **PyBBQ**[1] is a newly developed Python complement to BBQ, but uses finite differences. It mainly supports compatibility with other tools.
- **LEDET**[2] [85, 84] can be used for both conductors and magnets. It can simulate electro-magnetic and thermal transients in superconducting magnets in 2D and 3D geometry based on finite-differences method using MATLAB [90]. LEDET also uses the concept of lumped element modelling. Since LEDET is mainly used in this thesis, this concept will be explained further in the following section.
- **SIGMA**[3] [91, 84] is used to model electro-magnetic and thermal transients in a 2D magnet geometry using a COMSOL finite-elements model.
- **FiQuS**[4] [92, 84, 93] is used to perform quench simulations with 2D and 3D magnetic models using finite element methods and can also calculate magnetic fields at each time step. In the background FiQuS uses Gmsh [94] for geometry and meshing and GetDP [95] for solving and postprocessing.
- **ProteCCT**[5] [96, 84] is a tool specialised for canted-cosine-theta (CCT) for electro-magnetic and thermal transients using finite-differences method.
- **PSpice** [97] and **XYCE** [98] are used to solve circuits from different magnets. Both of these tools are developed and maintained commercially and are only incorporated into STEAM.

Since different tools have different specialties (physical domains, magnet types or magnitudes), co-operative simulations can be performed with **COSIM**. [99]. This tool ensures a consistent signal exchange between the simulation tools, taking possibly different temporal or spatial scales into account [100].

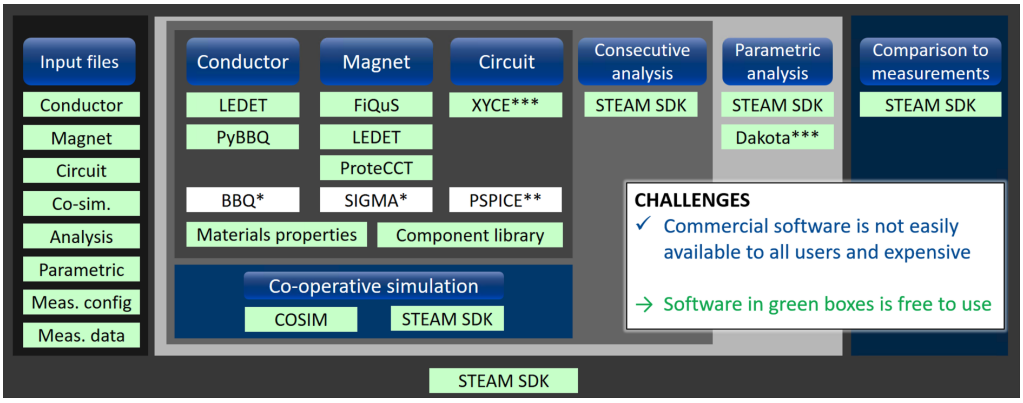


Figure 2.1: Schematic representation of the STEAM framework with all tools and input files used. Some functionalities are under active developement. *COMSOL license needed. **Commercial circuit solver from Cadence Design Systems. ***Free tools from Sandia Labs. Figure reproduced from [87]

[1] Python BusBar Quench
[2] Lumped-Element Dynamic Electro-Thermal model
[3] STEAM Integrated Generator of Magnets for Accelerators
[4] Finite Elements Quench Simulator
[5] Protection of Canted-Cosine-Theta

2.1.2 Implementation

In the stand-alone version controlled repository **STEAM-models** the validated models of conductors, magnets and circuits are stored and further developed [101]. For magnet models, in addition to .iron, .data, .map2d files from the magnet design software ROXIE [102], a YAML[1] file is also stored, which holds all important parameters of a magnet. These include: model status, design and geometry, specifications of windings, information about used cables (type, materials, strand and filament parameters), properties of the power supply, parameters of the used quench protection systems (quench heater, energy extraction, ect.) and also tool specific options. These models are in fact unified models used by each of the tools.
The python package **STEAM-SDK**[2] [103] combines the stand-alone tools in a simple way so that users who do not know the individual tools or python can run simulations. The main tasks of STEAM-SDK include:

- Read data from STEAM models and generate tool-specific models (each tool has differently structured models it can simulate, which can be realised in different languages)
- Generate, read and execute input files for simulations
- Read and parse simulation results (simulation results of different tools are only available in different formats and file types)
- Display 2D and 3D plots for different conductor, magnet and circuit types
- Compare graphically measurements and simulations in reports

Material parameters depend on different variables and have to be calculated at each time step and by the different tools. Since this is computationally very intensive and therefore time-consuming, the STEAM framework also features the **STEAM-material-library** implemented in `C++` [104]. This repository makes functions for material parameters accessible throughout the framework in various languages and tools.

2.1.3 Simulation setup and minimal example

STEAM analyses are organised in steps. In order to run simulations in STEAM, these steps (and their sequence) must be defined by the user in a YAML file. In addition to the step information, general settings, input paths for tools used and output paths for simulation results are also defined there. Listing 2.1 contains the Python code needed to start a STEAM analysis with a YAML input file by using the class `AnalysisSTEAM`.

```
1 from steam_sdk.analyses.AnalysisSTEAM import AnalysisSTEAM
2
3 if __name__ == "__main__":
4     AnalysisSTEAM(file_name_analysis='file.yaml').run_analysis()
```

Listing 2.1: Python Code example on how to run a steam analysis file named file.yaml

The most common steps are listed and briefly explained below:
SetUpFolder: This step generates the folder structure for the required simulation softwares.

[1] YAML Ain't Markup Language, originally: Yet Another Markup Language
[2] Simulation of Transient Effects in Accelerator Magnets - Software Developement Kit

MakeModel: This step generates the models required for the specified programes. If a simulation number is assigned to the step, a simulation input file for the reference model is also generated.

ModifyModel: This step changes the value of a specific parameter of the model. If a simulation number is assigned to the step, a simulation input file is also generated for the modified model.

ModifyModelMultipleVariables: This step is analogous to ModifyModel, except that multiple parameters can be changed in one step. Internally this step also reuses code from the ModifyModel step.

RunSimulation: This step finally runs the simulations. From all simulation numbers that are defined in the step definition, the simulation input files are executed with the specified tools.

RunViewer: This step compares measurements and simulations graphically. It requires a CSV[1] input file that links a measurements with their corresponding simulation and a configuration file that defines which plots are created. The plots are then saved individually in the desired file format and combined into a PDF or HTML report.

Listing 2.2 shows the YAML input file for a minimal example in which the magnet MCSXF is simulated in LEDET once with the default parameters from STEAM-models and again with a modified current of 120 A. Under the key `AnalysisStepDefinition` the steps are defined and under the key `AnalysisStepSequence` their sequence is determined.

```yaml
AnalysisStepDefinition:
  setup_folder_LEDET:
    type: SetUpFolder
    software: [LEDET]
  makeModel_ref:
    type: MakeModel
    model_name: BM
    file_model_data: MCSXF
    case_model: magnet
    software: [LEDET]
    simulation_number: 1
  modifyMaxPCvoltage:
    type: ModifyModel
    model_name: BM
    variable_to_change: current
    variable_value: 120 # A
    simulation_numbers: [2]
    new_model_name: [] # if specified model will be copied
    software: [LEDET]
  RunSimList:
    type: RunSimulation
    software: LEDET
    simulation_numbers: [1, 2]
    simFileType: .yaml
AnalysisStepSequence:
  - setup_folder_LEDET  # make LEDET model folder
  - makeModel_ref       # make simulation files for ref model
  - modifyMaxPCvoltage  # make simulation files with changed current
```

[1] Comma-Separated Values

```
29   - RunSimList           # Run simulation input files
```

Listing 2.2: Minimalistic example of a YAML input file for a STEAM analysis. Some flags and simulation_name are not shown for reasons of simplicity.

2.2 STEAM-LEDET

2.2.1 Finite difference and lumped element modelling

The finite element method is often used to model physical phenomena. In this method, a system is subdivided into small volumes or, assuming uniform properties in the third direction, a 2D cross section is subdivided into small areas. However, for the non-linear behaviour of superconducting magnets and the necessary combination of different physical domains, this method becomes computationally very intensive. The LEDET tool [86] used for the simulations in this thesis therefore uses lumped element modelling. In this type of modelling, a system is divided into smaller and smaller subsystems until the desired accuracy is achieved. These subsystems are less complex chunks that have significantly fewer input arguments and are small and discrete. Partial differential equations, as well as the space model, are thereby transformed into a set of ordinary differential equations with a finite number of parameters [105].
Lumped elements are very often used in modelling, such as in the field of electrical engineering. All symbols in a circuit, such as resistors, capacitors or coils, are basically lumped elements. In the case of a resistor, for example, the complex interaction between electrons and atomic nuclei is approximated with the simple equation $U = R \cdot I$. This type of modelling can be used for many problems from a wide variety of sub-areas of physics. For this, the system only has to be broken down into sufficiently small and sufficiently accurately modelled elements and combined in an equivalent circuit to approximate the system behaviour.
A great advantage of lumped elements is that different physical subdomains can be easily connected. Interactions and mutual influences can be modelled by coupling individual subnetworks that are specialised in one physical domain. This modelling approach is therefore particularly well suited for the simulation of superconducting magnets, since many different domains have to be modelled. In LEDET, there are three coupled subnetworks: one electromagnetic, one thermal and one for transitory coupling. Figure 2.2 shows these three domains as well as their coupling and energy exchange. Elements that belong to one subnetwork are in the colour of the subnetwork (yellow, red and blue respectively) and elements that belong to several subnetworks in different colours (green, orange and pink respectively). Each subnetwork consists of source, storage and sinks. In the following sections, these individual subclasses are discussed in more detail.
The simulations in this thesis are carried out with LEDET in 3D [106]. How the geometry is created for this is explained in more detail in subsection 3.1. The LEDET version used for the simulations is 2.02.26. The differential equations of the coupled lumped elements are solved by using the finite difference method[1] with a semi-implicit Euler implementation [107].

[1] The finite difference modelling method is practically using a network of non-linear lumped elements. However, in LEDET not all lumped elements are non-linear.

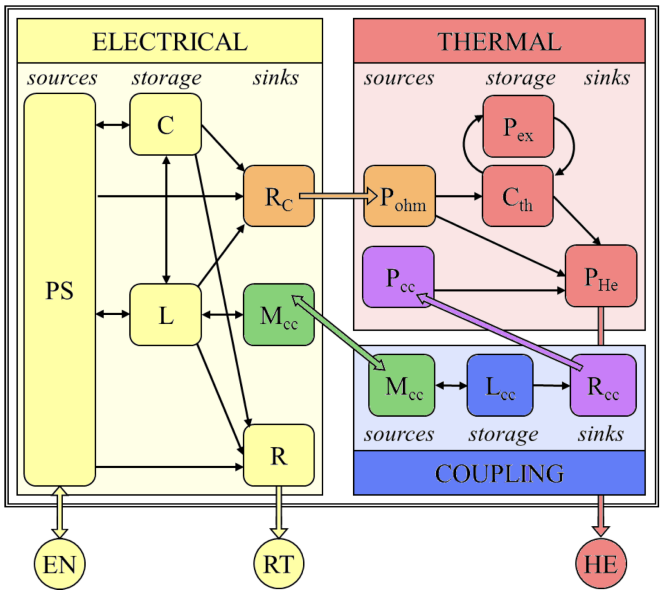


Figure 2.2: Schematic representation of the mutual coupling and energy exchange among the three interconnected sub-networks (electrical, thermal, and transitory) in LEDET [14].

2.2.2 Electro-magnetic model

In the following, the electrical subnetwork (see figure 2.2) will be further described. The source there is a power supply (PS) that is connected to the rest of the electrical network (EN). The energy provided by PS can be temporarily stored and exchanged by the storage elements, namely by capacitors (C) or coils (L). Sinks of this electrical energy are ohmic losses (R_C), which are simultaneously heat sources in the thermal subnetwork (P_{ohm}), coupling losses, which in turn enter the thermal subnetwork as heat sources (P_{cc}) via the coupling subnetwork, and a resistor (R) connected to a room temperature (RT) environment modelled as an infinite sink. The coupling current network is connected to the electrical network via mutual inductances (M_{cc}), which models energy exchange in both directions.
LEDET provides the usual lumped elements for electrical modelling, including resistors, inductances, capacitors, as well as diodes, switches and voltage or current sources. The circuit model of the lumped elements used to model a superconducting magnet is shown in Figure 2.3. A conductor section of a coil is modelled by two parameters. The first parameter is a variable resistance $R_{Coil}[\Omega]$, which is zero in the superconducting state and non-zero in the normal state. The second parameter is a self-inductance L[H] which is mutually coupled (M[H]) to other self-inductances of other parts. In the figure, the two elements a and n are shown as examples. In a true magnet model, however, not only two elements are used, but as many as necessary to achieve the desired accuracy. In 2D models, one such element is usually simply applied for a half-turn. Also the decrease of the differential inductance with increasing magnet transport current, due to the saturation of the iron yoke of the superferric HL-LHC HOC magnets, is modelled [85, 14, 106]. The fact that IFCC reduce the magnet differential inductance due to the local magnetic field changes they induce in the conductor which provokes a faster magnet discharge is

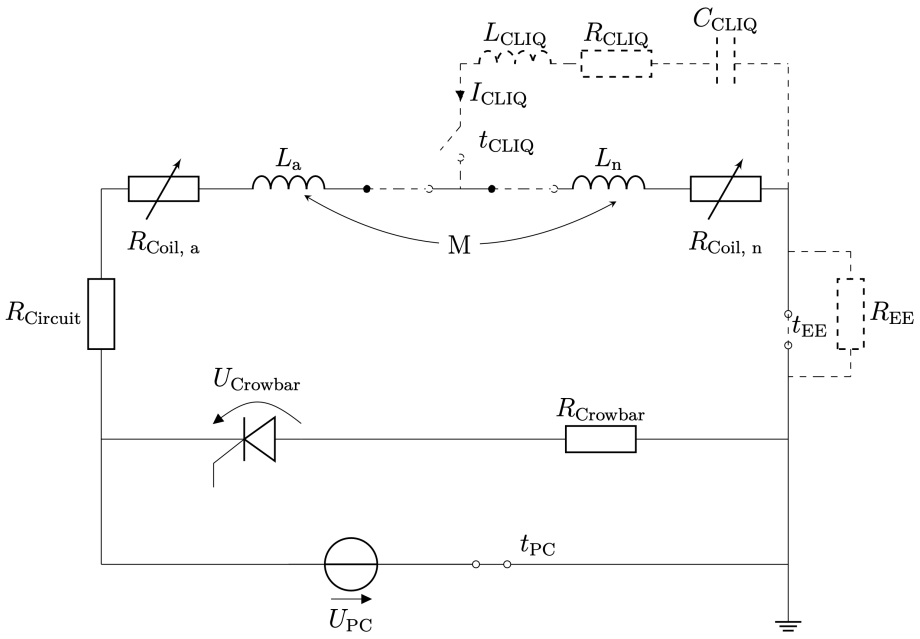


Figure 2.3: Electric network used in LEDET to reproduce the electro-dynamic behaviour of a superconducting magnet and its surrounding components. Dashed elements can be optionally included for quench protection. [108]

also modelled for 2D models [109, 85].
In addition to the elements modelling the magnet, the circuit is also modelled in the figure. The power source is an ideal voltage source U_{PC}[V] with a switch to turn it off at the time t_{PC}[s]. Above the power source is a crowbar, which carries the current in the case of a power abort when a quench is detected. The crowbar is described by the threshold voltage of the thysistor $U_{Crowbar}$[V] and the warm resistance of the crowbar $R_{Crowbar}$[Ω]. $R_{Circuit}$[Ω] describes the warm resistance of the circuites and various optional CLIQ or Energy Extraction (EE) parameters are specific to these quench protection systems. Many of the parameters shown in the figure are dependent on magnetic field, transport current, temperature or other parameters. This is also accounted for in LEDET [85].

2.2.3 Thermal model

As shown in Figure 2.2, there are two sources of thermal power: P_{cc} dissipated by coupling currents in resistors and P_{ohm} due to ohmic losses in the conductor. The thermal subnetwork has thermal energy storage elements (C_{th}) that can exchange energy with each other (P_{th}). The only heat sink of this subnetwork is the power dissipated in the helium cooling (P_{He} in figure 2.2). This power is associated with the helium environment (HE), which is modelled as a thermal sink at constant temperature.
As in the electrical model, a LEDET model consists of many of these thermal elements. The thermal behaviour of the conductor, insulation and any resin is thus approximated for specific volumes by thermal resistances and storage elements. Over each of these

individual volumes, a thermal balance is introduced, defined as [85]:

$$P_{if} + P_{is} + P_{ohm} + P_{ex} + P_{sink} = \bar{c}(T) \cdot \frac{d(T(t) - T_{He})}{dt} \tag{2.1}$$

Where P_{if}[W] and P_{is}[W] are the powers due to inter-filament and inter-strand coupling losses, P_{ohm}[W] is the power generated by the heat losses in the conductor, P_{ex}[W] is the power transferred via heat exchange with other elements, P_{sink}[W] is the power dissipated by the helium bath, $\bar{c}[\frac{\mathrm{J}}{\mathrm{K}}]$ is the average heat capacity of the element weighted by cross-sectional size, T[K] is the variable temperature of the element, and T_{He}[K] is the constant temperature of the helium bath (modelled by the thermal sink).
Also for thermal modeling a circuit diagram can be made, which has analogous symbols to an electrical circuit. Here the heat flow corresponds to the electric current and the temperature gradient to the electric voltage. The thermal circuit diagram of a LEDET model with one element b coupled to another element h is shown in Figure 2.4. Here, the current sources $I_{if,b}$, $I_{is,b}$ and $I_{ohm,b}$ model the heat sources P_{if}, P_{is} and P_{ohm} from equation 2.1. If the superconductor is in the superconducting phase, energy is supplied only via P_{if} and P_{is} because the superconductor has zero resistance; if it is in the normal phase, energy is supplied via P_{ohm}. Therefore, the circuit diagram contains a switch q_b which changes between these two modes depending on the state of the conductor. Furthermore the voltages U_b and U_h stand for the temperatures of the elements b and h. Through thermal resistances $R_{b,h}$, which model insulation layers, energy is transferred to other elements (corresponding to P_{ex} in equation 2.1). The helium bath is modeled by U_{He} and R_{He}. The heat energy stored in the element is stored in the heat capacity $C_{th,b}$.

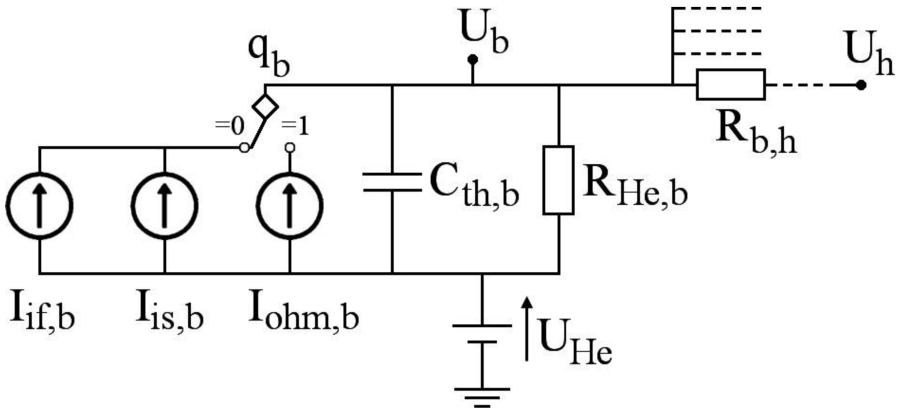


Figure 2.4: Analogous thermal circuit diagram of a thermal element b used in the thermal lumped element sub-network of LEDET. Element b is exchanging heat with an exemplary element h coupled through $R_{b,h}$. [86]

To couple the electromagnetic model with the thermal one, two effects have to be considered. First, the resistive power in the conductor P_{ohm} depends on the current in the conductor I_b[A]: $P_{ohm} = R_b I_b^2$ [14, 85]. On the other hand, the electrical resistance of the conductor $R_b[\Omega]$ depends not only on the magnetic flux density B_b[T] but also on the temperature of the element T_b[K] and on the switch position q_b. This change of R_b further leads to a build up of voltage across the element.

To model the conductor heat capacity, the cross section is divided into four parts: $f_{strand}[\%]$, $f_{insulation}[\%]$, $f_{internalvoids}[\%]$ and $f_{externalvoids}[\%]$. These describe the percentage cross sectional area of strands, insulation and internal and external voids. In superconducting cables, porosities and other openings are often created to allow the helium to cool the cable from the inside. These openings are divided into internal and external voids. Internal voids are empty spaces between strands and external voids are empty spaces between strands and cable insulation. These voids can be filled either with helium or with Kapton insulation or epoxy impregnation [14]. With a further parameter (f_{He}) is noted how much of these voids are filled with helium. By this notation the helium cooling is modelled in LEDET.

2.2.4 Coupling current model

The third sub-network models the two coupling currents between strands (ISCC) and between filaments (IFCC) resulting from the change of the applied magnetic field. These coupling currents are modeled as stand-alone electrical circuits consisting of only a resistor and a coil. This arrangement is shown in Figure 2.5 as L_{cc} and R_{cc}. The coil *Lcc* is coupled to the inductance of each element (L_a ... L_n) from the electrical sub-network via M_{cc}. This configuration allows energy to be exchanged with the electrical network (M_{cc}), stored in the coupling network (L_{cc}), and dissipated (R_{cc}) into the thermal sub-network (see 2.2).

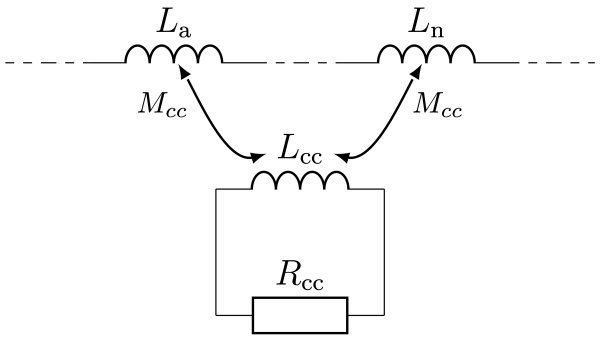


Figure 2.5: Electric network modelling a single coupling current used in the coupling sub-network of LEDET [108].

To model the effects of a changing magnetic field in x- and y- direction[1], two of the coupling current loops shown in 2.5 must be used for each electrical element. However, for ISCC only one is needed, since the inter-strand coupling losses due to magnetic field changes perpendicular to the cable broad face are significantly higher than due to magnetic field change in the parallel direction [110, 85]. The total power dissipation of the two coupling currents is as follows [85]:

$$P_{if} = R_{if,x} \cdot {I_{if,x}}^2 + R_{if,y} \cdot {I_{if,y}}^2 \tag{2.2}$$

$$P_{is} = R_{if} \cdot {I_{if}}^2 \tag{2.3}$$

[1]The longitudinal axis of the magnet, to which the particle beam will be parallel, usually is the z-axis. Thus, x- and y-axis correspond here in top and bottom of the magnet cross section.

The required parameter values for equivalent resistance, self- and mutual inductance are geometry dependent and therefore have to be calculated for each strand and cable independently [85]. How dominant P_{if} and P_{is} are depends on various factors. IFCC are mainly dependent on the effective transverse resistivity of the strand and thus on filament-twist-pitch, the matrix material and the type oft he superconductor. ISCC, on the other hand, depend essentially on the cable width, number of strands and strand twist-pitch. In detail the more strands or the lower the strand-twist-pitch the higher P_{is}. [14, 110]
This sub-network is coupled to the other two networks in two ways (see figure 2.2). First, the equivalent coupling current loops are coupled to the inductances in the electrical sub-network via M_{cc}. In this way the magnet differential inductance is influenced by IFCC and ISCC. Second, the energy dissipated in R_{cc} is fed into the thermal subnetwork as input power P_{cc}.

2.2.5 Quench model

As explained in the previous section, the temperature of an element increases due to ISCC, IFCC or due to heat exchange with neighbouring elements. If more energy is introduced into an element in this way than can be removed by the heat sink or other neighboring elements, the temperature will continue to rise. It can get to the point where the temperature rises above the critical value T_c and a quench occurs. This critical temperature T_c also depends on the magnetic field and the current density (see section 1.2.1 for more information). The larger these two physical quantities are, the lower temperatures are sufficient to start the transition to the normal phase.
This quench can then spread through the magnet in 3 ways: coupling losses, transverse heat diffusion and longitudinal quench propagation. The effects of coupling losses are already modelled by the electrical and coupling current sub-networks with M and M_{ii} respectively and can cause the temperature to increase due to ohmic losses, resulting in a quench. Also the effects of transverse heat diffusion are already modelled by the thermal sub-network as the elements are connected via thermal resistors that model the insulation of the cables and can transfer heat. However, longitudinal quench propagation is not yet considered in the introduced model.
In 2D LEDET models the analytical formula (see 1.1) for this longitudinal quench propagation velocity v_q is used to run 2D+1D simulations. This already leads to a good approximation while also having a computationally efficient model [111]. Starting from elements that are no longer superconducting, the quench front propagates as a running wave in both directions of the conductor with $v_q[\frac{\mathrm{m}}{\mathrm{s}}]$. Thus, if the length along the current direction of each element is described by $ds_i[m]$, the quench propagates from element b into element h in time $t_{q,bh}[s]$:

$$t_{q,bh} = \frac{0.5(ds_b + ds_h)}{v_q} \tag{2.4}$$

At this point the switch q_b is set to 1. This modelling of the longitudinal quench propagation causes the quench to propagate also longitudinally from the first coil-block of the coil to the second one also from the quenching coil to the neighbouring coils, which are so far apart that a transversal propagation is negligible.
3D models in LEDET however do not rely on this analytical formula for v_q, but rather solve the 3D thermal diffusion problem [107, 112, 113]. This leads to an increase in

computational effort, but also increases the accuracy of the simulation, especially for short magnets, as in the case of the HL-LHC HOC magnets.
To position a quench at a specific location, a Gaussian temperature profile is applied to that very location. Specifically, in LEDET, for each half-turn, the width of this Gaussian distribution (`lengthHotSpot_iStartQuench`) must be specified and at what time it is applied (`tStartQuench`). The selection of this specific temperature profile was dictated by its enhanced representational accuracy of real-world conditions, as compared to the overly simplistic approach of merely assigning a value to the flag q_b.

3 Geometry modelling in STEAM

3.1 3D racetrack geometry

In STEAM, there are various geometric models for magnets, such as CCT[2] (straight and curved), pancake or solenoid. The HL-LHC HOC magnets, however, are constructed from radially arranged racetrack coils. The racetrack coils of each HL-LHC HOC magnet have different dimensions and a different number of racetrack coils is used. This number determines the harmonic order of the magnet. For example, the sextupole has six racetrack coils, the octupole has eight ect. This racetrack geometry is shown in Figure 3.1 as an example for MCSXF. The sextupole MCSXF consists of six of these coils which are arranged radially around the z-axis with the same spacing.

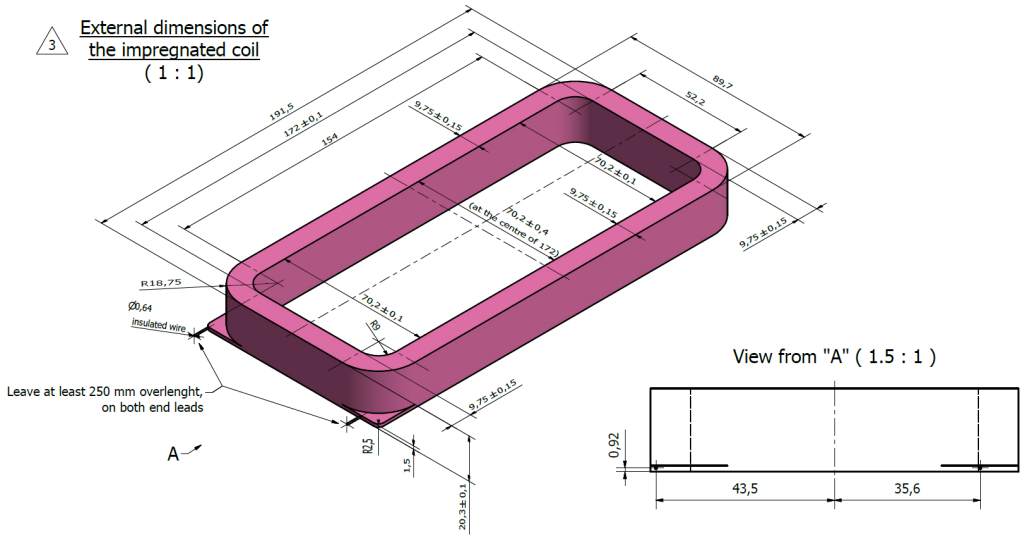


Figure 3.1: Technical drawing of the racetrack coil geometry of the MQSXF magnet including geometrical specifications. Figure reproduced from [114].

The implementation of this racetrack geometry in LEDET is not accurately corresponding to the geometry in figure 3.1. Figure 3.2 shows this inaccurate representation by the example of MCSXF. It can be seen, that the sections of a coil parallel to the z-axis are interconnected using arcs or half-circles instead of the desired racetrack geometry. In addition, the connection of the individual coils is implemented with simple Bézier

[2]Canted Cosine Theta

curves. In reality, however, a much more complex PCB[1] connection is used. Furthermore, this incorrect geometry overestimates the total conductor length by some 20 % (slightly different for each magnet), which can lead to larger errors in simulations due to the additional cable length. It is also noticeable that from some viewpoints, such as the top view, the geometry corresponds to a semicircle.
To rectify this error, the geometry is re-implemented. However, in order to implement the racetrack coil in a consistent way across different STEAM tools, the geometry is not generated in LEDET (MATLAB), but in STEAM-FiQuS. In this Python STEAM tool, there are already several geometry generators, some of whose methods can be repurposed. Moreover, the future objective is to consistently generate geometries within STEAM-FiQuS and then import them into the relevant tools, a functionality that is also available in STEAM-LEDET and other STEAM tools.

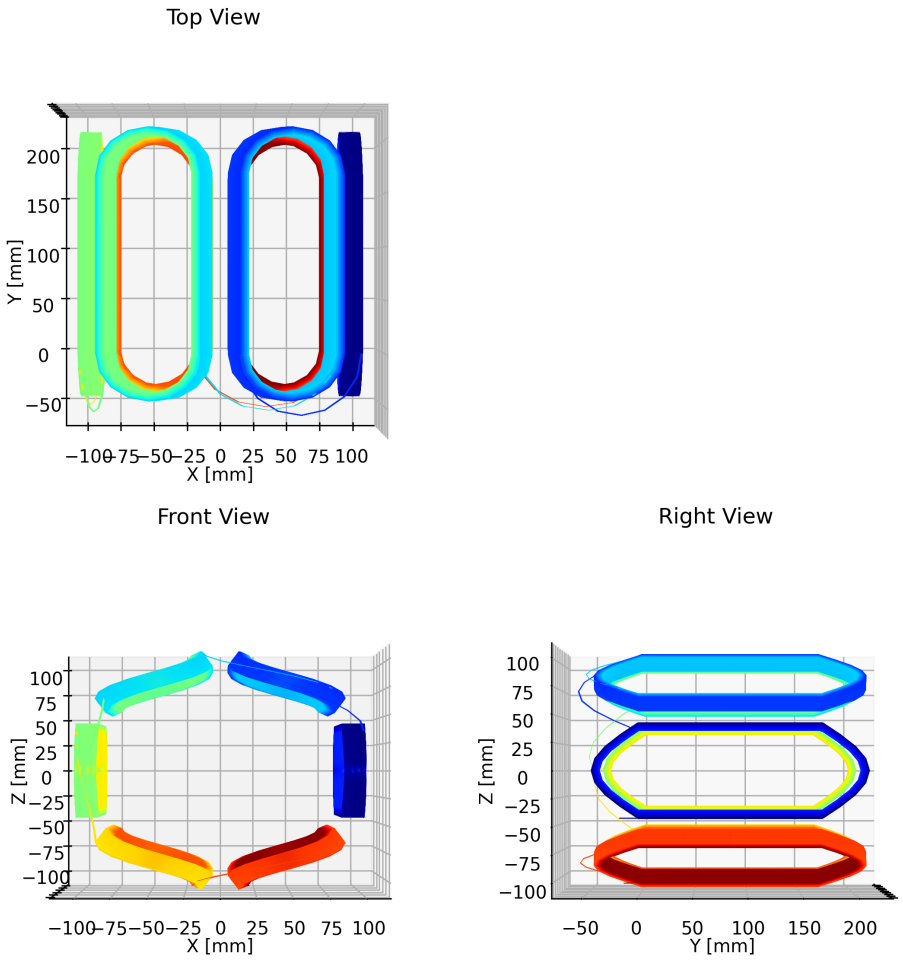


Figure 3.2: Top, front and right side view of the old MCSXF geometry in LEDET displayed in 3D prespective. The color represents the length of the conductor along the electrical direction starting from blue to green then to yellow and ending in red.

[1] Printed Circuit Board

In the following it is explained by which parameters the racetrack geometry is defined, how the coil geometry is implemented in FiQuS and stored in `.cond` files, how the conductor geometries are created from the coil geometries stored in the `.cond` files, how the conductor ends of individual coils are connected and finally how the input files for LEDET are generated. The recourse to saving the coil geometry in `.cond` files was deliberately adopted due to the necessity, within the context of STEAM, for a distinct capability to partition `.cond` files into individual conductors. This requirement stems from the fact that a geometry, when created using the OPERA software [115], can be exported in the form of a `.cond` file.

3.1.1 Defining parameters of a racetrack coil and magnet

The geometry of a magnet must be specified by a user in the STEAM model. To read these specifications the class `RacetrackMagnet` was implemented, whose UML diagram is shown in figure 3.3. The parameters under the entries `racetracks` and `placement` under the property `geometry` are explained in this section. The rest of the parameters are explained in the following sections.

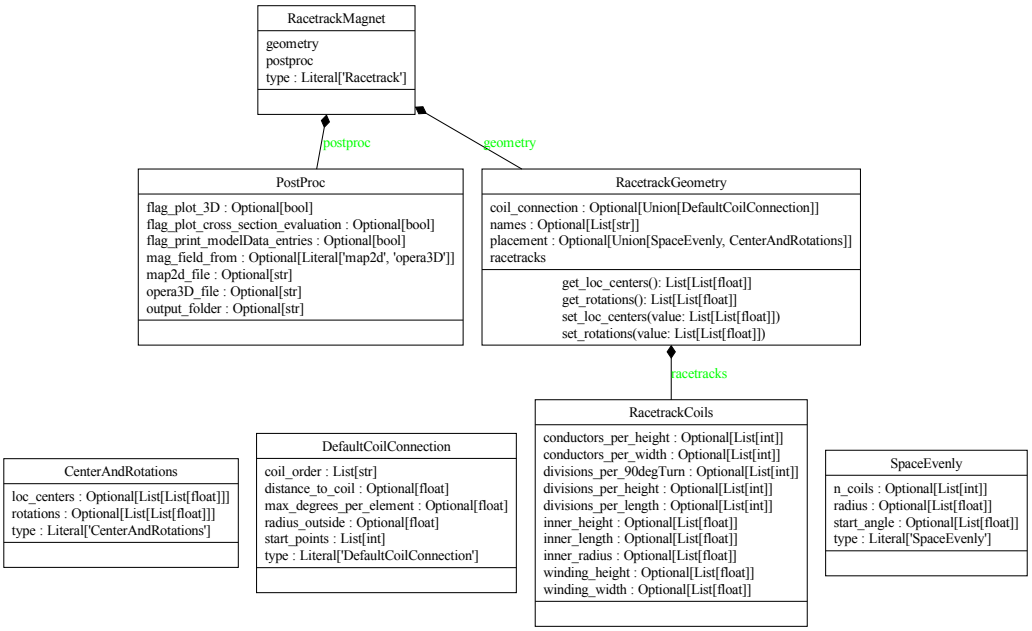


Figure 3.3: UML diagram of the `RacetrackMagnet` class that is used to store all parameters related to the geometry and geometry processing of a magnet consisting of racetrack coils.

The geometry of a racetrack coil can be represented in different ways. The chosen parameters are shown in figure 3.4 and are inspired by the representation from ROXIE [102, 116]. For each of these parameters it is also specified how many parts the corresponding length or 90 degree turn is divided into and how many conductors are present in height and length direction. In the class `RacetrackMagnet` these parameters are stored in the class `RacertrackCoils`.

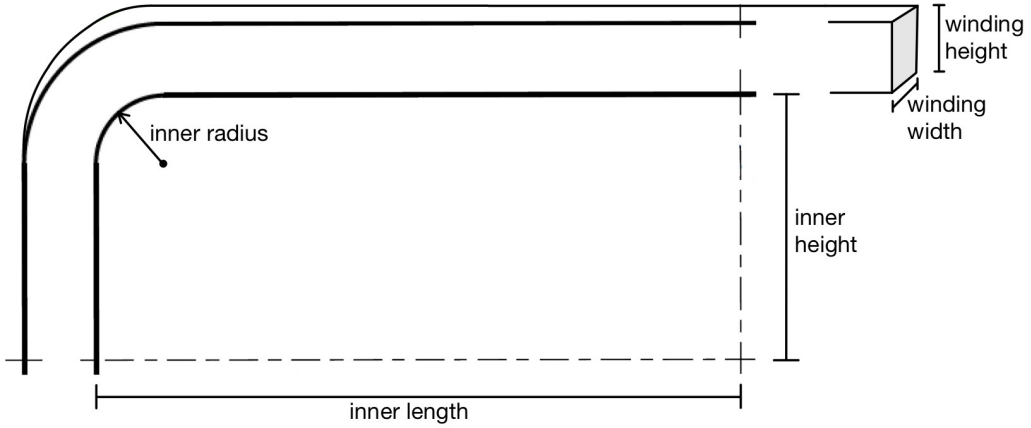


Figure 3.4: Schematic representation of the chosen geometrical parameters of the definition of a racetrack coil in STEAM-FiQuS.

Usually magnets consist of several racetrack coils which are arranged differently. In order to uniquely define their position in space, for each coil three center coordinates and three rotation angles are required. The center coordinates define the center of gravity of a racetrack coil. The rotation angles define how many degrees the coil is rotated around the x-, y- and z-axis. The center of rotation is the coil's center of gravity. With these two time three location variables it is possible to define every possible position and orientation of a coil. Due to their flexibility, however, this definition method is also cumbersome for a user. Center point and rotation angle are therefore only implemented as private properties of the class `RacetrackGeometry` (see figure 3.3).
To make the input easier for the user, the entry `positions` was added to the class. Here, different types can be implemented for a user to determine the position of the coil. To facilitate the input of geometry for magnets consisting of racetrack coils evenly distributed around the z-axis, the type `SpaceEvenly` is implemented. By having a user define the number of coils (`n_coils`), their distance from the z-axis (`radius`), and the angle of the first coil to the x-z-plane (`start_angle`), a once-defined racetrack coil can be arranged `n_coils` times radially around the z-axis. In the implementation, these parameters are simply converted to the private parameters center coordinates and rotation angle. Also `CenterAndRotation` was implemented as a type where a user can directly enter these two time thress variables.
All HL-LHC HOC magnets exhibit a uniform coil geometry, consistent across all poles and symmetrically distributed around the z-axis, thus the models employ the `SpaceEvenly` placement type.

3.1.2 Implementing coil geometry in FIQUS and generating `.cond` files

From the parameters of the `RacetrackMagnet` class, the geometry consisting of hexahedrons must be generated. Consistent with existing FiQuS implementations, the hexahedron points are enumerated in the following manner:

```
                y (v)
             3----------2
             |\      ^   |\
             | \     |   | \
             |  \    |   |  \
             |   7------+---6
             |   |  +-- |-- | -> x (u)
             0---+---\--1   |
              \  |    \  \  |
               \ |     \  \ |
                \|     z(w)\|
                 4----------5
```

Listing 3.1: Naming convention of Hexahedron vertices in FiQuS.

To build a coil of such hexahedrons in the x-z-plane, the classes `Hexahedrons` and `Coil` were implemented. The class `Hexahedrons` stores the coordinates of the hexahedrons and provides functions to add new elements or change the order of the hexahedrons. The `Coil` class inherits from `Hexahedrons` and creates the hexahedrons using the parameters of `RacetrackGeometry`.
To achive this, first the length of the elements in x- and z-direction and the degrees per element in the 90 degree curve are calculated. Then the coordinates of the first element are calculated directly from the given parameters. This calculation is easiest for the first length element in z-direction, because this is a straight section and will always exist[1].
After the first hexahedron is generated, the inherited method `_add_elem_to_elem()` from `Hexahedrons` adds as many elements as required along the z-axis with the previously calculated element length. This method always takes the four endpoints of the previous hexahedron as starting points and the four endpoints are calculated by moving the starting points by the specified length in the specified direction – so in this case by the calculated element length in z-direction. The inherited method `_add_elem_with_rot()` is then used to build the 90 degree curve with hexahedrons. Center of rotation is at a distance of `inner_radius` in the negative x direction. This method takes the four endpoints of the previous hexahedron as the starting points and as the endpoints, these points are rotated about the specified axis by the center of rotation by the specified angle. Once the 90 degree curve is completed `_add_elem_to_elem()` is called again to move in the x direction. After building straight section and 90 degree curve four times in a row the coil in the x-z-plane is completed.
To facilitate post-processing and further procedure, two more changes are made to the list of hexahedrons. First, coils whose polarity is reversed must be rotated 180 degrees around the z-axis. In postprocessing the coil is processed along the hexahedrons and by rotating the electrical direction is always correct. By the coil order specified in the class `DefaultCoilConnection` the polarity can be determined. Second, the start of the coil must still be placed exactly at the start of a half-turn. Starting from the middle of the straight height section the first half of the hexahedrons should belong to one half-turn and the second half always to the next one, so that each hexahedron can be assigned the halfturn index in electrical order very easily. The start is therefore shifted by a 90 degree turn and by half of a straight section in x-direction.
The racetrack geometry thus generated has its center of gravity at the coordinate center

[1] Since the MCTXF and MCTSXF magnets do not have a straight piece in the x-direction, such a piece is not always available. Starting with a curved piece would be computationally more expensive.

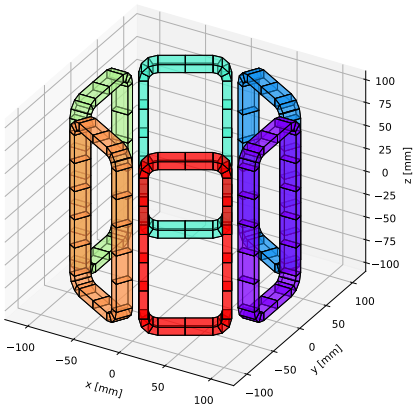
and must therefore be correctly positioned in space and rotated. To do this, all points are moved by the vector to the desired center of gravity by adding the coordinates. To then rotate the coils, a method is implemented that subtracts the center of the coil and then rotates all points about the 3 different axes using the rotation matrices. The rotation of the points around the center of gravity coinciding with the coordinate origin can be described as follows:

$$\begin{pmatrix} x_{\text{rot},\,x} \\ y_{\text{rot},\,x} \\ z_{\text{rot},\,x} \end{pmatrix} = R_x(\theta) \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & -\sin(\theta) \\ 0 & \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

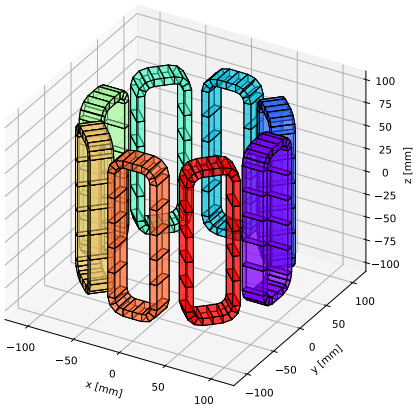
$$\begin{pmatrix} x_{\text{rot},\,y} \\ y_{\text{rot},\,y} \\ z_{\text{rot},\,y} \end{pmatrix} = R_y(\theta) \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

$$\begin{pmatrix} x_{\text{rot},\,z} \\ y_{\text{rot},\,z} \\ z_{\text{rot},\,z} \end{pmatrix} = R_z(\theta) \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

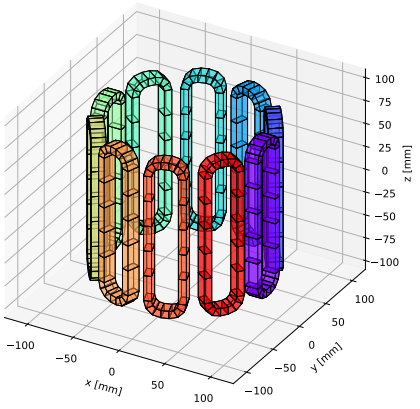
Here $R_x(\theta)$, $R_y(\theta)$ and $R_z(\theta)$ are the rotation matrices, x, y and z are the original coordinates of the vertices of the hexahedrons and $x_{red,i}$, $y_{red,i}$ and $z_{red,i}$ are the coordinates rotated around the axis i. After this rotation the centroid coordinates are added again.
Each coil defined in `RacetrackMagnet` is generated, positioned and rotated by this procedure. The hexahedrons of new coils are always appended to the hexahedron list until the whole magnet is created. A plot interface has been implemented for this geometry, showcasing the resulting constructed structure. Figure 3.5 shows the resulting geometry of all five HL-LHC HOC magnets consisting only of hexahedrons.
The list of hexahedron coordinates can be exported to a `.cond` file using an already implemented class in STEAM-FiQuS. To ensure the validity of the created `.cond` files, they were loaded into OPERA and verified.



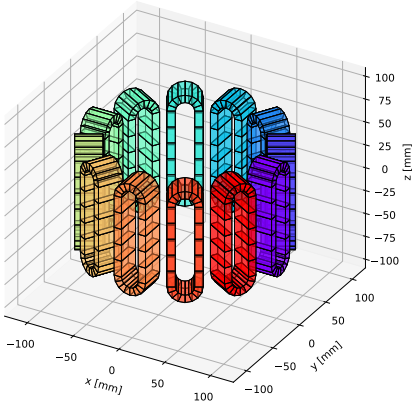
(a) MCSXF



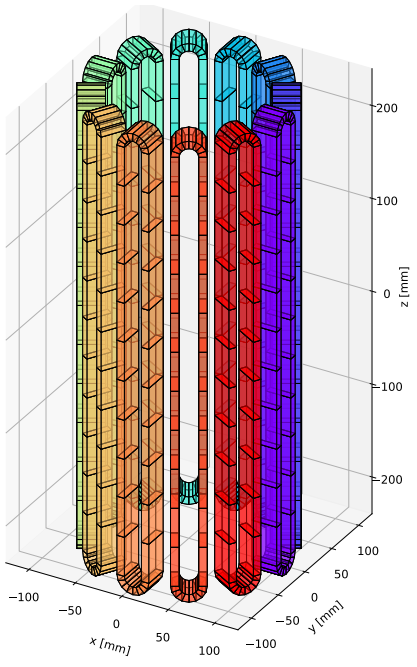
(b) MCOXF



(c) MCDXF



(d) MCTSXF



(e) MCTXF

Figure 3.5: Generated hexahedron models of every HL-LHC HOC magnet. Each coil is coloured differently.

3.1.3 Dividing coil geometry from `.cond` files in conductor geometries

For the generation of STEAM-LEDET geometry input files, not the entire coil, but each individual conductor must be broken down into hexahedrons. For example, if a coil consists of 10 hexahedrons and has 288 turns, $10 \cdot 288 = 2880$ smaller conductor hexahedrons are needed for this coil. Therefore, for each coil, a loop is executed for each turn, processing the conductor in electrical sequence.
The start- and end-surfaces of each hexahedron must therefore be divided by the number of windings in the width and height directions. Given the lack of practical benefits, we assume that these surfaces can manifest as arbitrary concave quadrilaterals. An efficient algorithm must be established to decompose these surfaces into sub-surfaces, both vertically and horizontally. This algorithm may be invoked thousands, or even tens of thousands, of times.
This subdivision is calculated by cutting linear straight lines. First, the contour lines are subdivided into the desired number of points. The lines $\overrightarrow{P_0P_1}$ and $\overrightarrow{P_3P_2}$ are divided into `conductors_per_width+1` equidistant points and the lines $\overrightarrow{P_1P_2}$ and $\overrightarrow{P_0P_3}$ are divided into `conductors_per_height+1` points. Mathematically, a weighted average can be formed for this:

$$P_{01,i} = P_0 + \overrightarrow{P_0P_1} \cdot \frac{i}{N} \quad \text{with} \quad \overrightarrow{P_0P_1} = P_1 - P_0 \quad \text{and} \quad i \in \{0, 1, \dots N\} \tag{3.1}$$

Where $P_{01,i}$ is the set of desired points on the route $\overrightarrow{P_0P_1}$ and N is the number of desired points. Thus, $N =$`conductors_per_width+1` (or $N =$`conductors_per_height+1` for distances along height) holds. Figure 3.6 shows the points obtained in black for an example where 3 conductors per width and 2 conductors per height are applied.

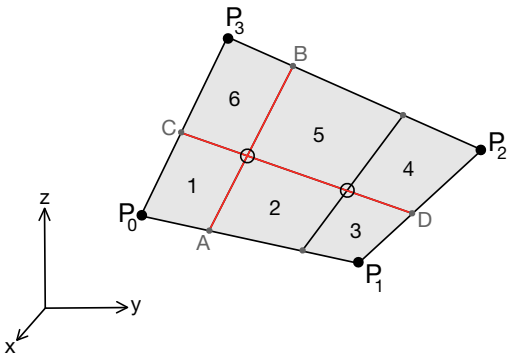


Figure 3.6: Schematic drawing of a arbitrary concave quadrilateral in 3D space with different points needed for the designed subdivision algorithm

The thus obtained associated black points can now be connected by straight lines, which are shown in red in figure 3.6. The missing points inside the quadrilateral, are the intersections of these straight lines. To calculate these intersection points, the function `__find_intersection_point` is implemented. This function takes 4 points A, B, C, and D (see figure 3.6) and calculates the intersection of the lines through A and B and the lines through C and D. The formulas for calculating the intersection points are explained below.

In 3D space, two lines do not necessarily intersect – they can be skew or parallel. Given two lines defined by four points A, B, C, and D, we can represent these lines as:

$$A + s(B - A) = P, \qquad \text{where} \quad 0 \leq s \leq 1,\ s \in \mathbb{Q} \tag{3.2}$$

$$C + t(D - C) = P, \qquad \text{where} \quad 0 \leq t \leq 1,\ t \in \mathbb{Q} \tag{3.3}$$

where P is the point of intersection and s and t are scalars. In these two equations only s and t are unknown. To calculate the intersection point P, the equations 3.2 and 3.3 are subtracted from each other:

$$A + s(B - A) - C - t(D - C) = 0$$

Rearranging terms and denoting $(B - A)$ as $\vec{u}$ and $(D - C)$ as $\vec{v}$, the equation can be written as:

$$s\vec{u} - t\vec{v} = \vec{w}, \quad \text{where} \quad \vec{w} = C - A$$

This equation can be solved by cross-multiplying both sides of the equation by v:

$$\vec{v} \times s\vec{u} - \vec{v} \times t\vec{v} = \vec{v} \times \vec{w}$$

Since the cross product is distributive over addition, we can rewrite the left-hand side of the equation as:

$$s(\vec{v} \times \vec{u}) = \vec{v} \times w + t(\vec{v} \times \vec{v})$$

Given that the cross product of any vector with itself is zero $\vec{v} \times \vec{v} = \vec{0}$, the equation simplifies to:

$$s(\vec{v} \times \vec{u}) = \vec{v} \times \vec{w}$$

The two vectors $\vec{v} \times \vec{u}$ and $\vec{v} \times \vec{w}$ are thus colinear, where s is the scaling factor. Since division of vectors is not defined, the equation is transformed in which the identity $\frac{1}{\vec{v} \times \vec{u}} = \frac{1}{\|\vec{v} \times \vec{u}\|} \cdot \overrightarrow{e_{vu}} = \frac{1}{\|\vec{v} \times \vec{u}\|} \cdot \frac{\vec{v} \times u}{\|\vec{v} \times \vec{u}\|} = \frac{v \times u}{\|\vec{v} \times \vec{u}\|^2}$ is applied[1]:

$$s = \frac{(\vec{v} \times \vec{w}) \cdot (\vec{v} \times \vec{u}}{|\vec{v} \times \vec{u}|^2})$$

Returning to the original variables, we obtain the formula for s:

$$s = \frac{(C - A) \times (D - C) \cdot ((B - A) \times (D - C))}{\|((B - A) \times (D - C))\|^2} \tag{3.4}$$

In the same way, the relation for t is obtained:

$$t = \frac{(C - A) \times (B - A) \cdot ((B - A) \times (D - C))}{\|((B - A) \times (D - C))\|^2} \tag{3.5}$$

In the implementation, both t and s are calculated and it is checked whether the intersection points obtained in this way are within a certain tolerance. Furthermore, the function checks whether there is an intersection point, i.e. whether the lines are parallel or skew. The subdivision points obtained by intersecting are shown as circles in Figure 3.6. The surface can thereby be divided into the desired number and distribution of sub-surfaces. These sub-surfaces are numbered from one to six in the figure. The order in which they are numbered is determined by the specified starting points in `coil_connection` under `start_points`. In the STEAM model of the racetrack coil the start point is specified for each coil. The numbering of this start point follows the following logic:

[1] This has the further advantage that no singularities arise since there is no division by zero, NaN or inf.

```
      ^                7---------6
      |                |         |  upper coil block
    y |                4---------5
      |
      |
      ------------------> x
      |
      |                3---------2
      |                |         |  lower coil block
      |                0---------1
```

Listing 3.2: Naming convention for the different possible start points of the coil winding

To further reduce the computational effort, all these intersection points are calculated at the first loop through the coil and stored in a dictionary. The calculation of common intersection points is therefore not repeated. This also avoids the double calculation for start and end surfaces.
These calculations were implemented in STEAM-FiQuS in the course of a post processing step. The coil hexahedrons are read from the previously generated `.cond` and finally subdivided. To visualise this subdivision, a 3D plot was implemented that updates live with the calculation and visualises the subdivision. Figure 3.7 shows some of the first frames of this visualisation again for the case with 2x3 conductors in the coil. Turn by turn the coil is processed along the width direction. When the whole width is divided, the next height position is selected. For practical reasons in the winding of the coil, the direction is reversed for each level. If the winding starts on the left and ends on the right, then it starts on the right and ends on the left in the next layer.

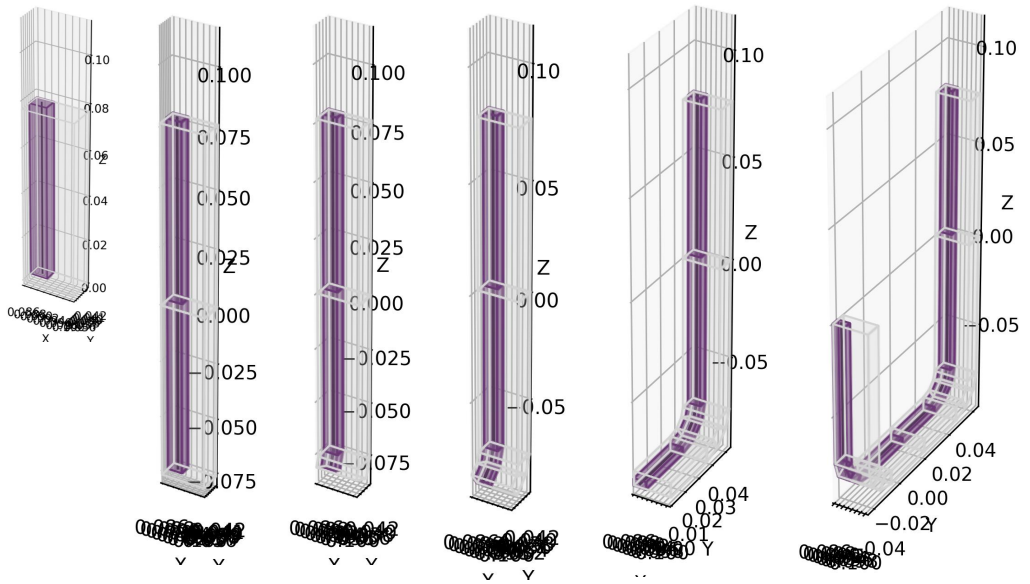


Figure 3.7: Frames number 1, 2, 3, 4, 6 and 11 (l.t.r.) of the subdivision of coil hexahedrons into conductor hexahedrons. Conductor starting point is 7 (see listing 3.2).

3.1.4 Generating PCB coil connection

The resulting geometric model, which is composed of hexahedrons, has only one limitation: the connection of the coils. The last hexahedron of a coil simply ends with the last turn of the coil and the first hexahedron of the next coil simply starts in the next coil. The next step is therefore the connection of the last hexahedron of a coil with the first hexahedron of the following coil.
The connection of each coil may vary from magnet to magnet and cannot be predicted. Also the different HL-LHC HOC magnets have slightly different geometries of the coil connection [117, 118, 119, 120, 121]. Due to time constraints, only the connection geometry of MCSXF is implemented as `DefaultConnection`[1], since this also leads to no overlaps for all other magnets. The error of this simplification is very small, because the connection is not in contact with other parts of the magnet and the magnetic field outside the magnet is very small anyway. It is only important that the length of the cable outside is about the same, because it can be that the conductor runs so long in the helium bath that a transversal propagation of the quench from one coil to the next is prevented by the cooling.

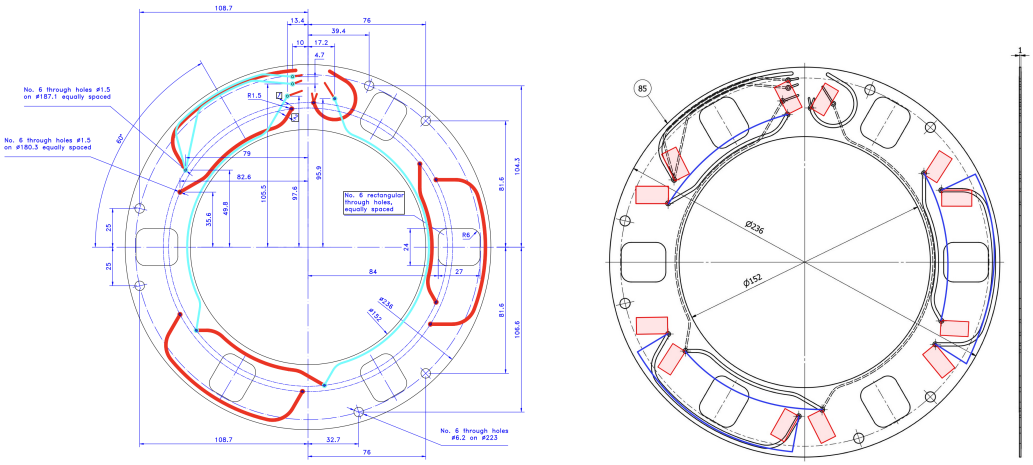


Figure 3.8: Two technical drawings of the PCB used to connect the individual coils of MCSXF at the rear end. On the left with dimensions, on the right the coil cross sections have been added in red and the cable simplification approximation in blue. Shown is the x-y plane with the z-axis running along the magnet. Figure reproduced from [117] and [114].

The HL-LHC HOC magnets are connected busbars hosted in PCB[2]s, which have different geometries [117, 118, 119, 120, 121]. The geometry of the PCB of MCSXF is exemplified in Figure 3.8 on the left. In figure 3.8 on the right, the underlying cross-sections of the coils are shown in red. It can be seen that conductors ending inside the coil are connected with a shorter cable and conductor endings outside with a longer cable. This more complex geometry is simplified to make it easier to approximate with hexahedrons. The inner traces connecting the innermost turns are approximated as a circular arc around the z-axis, and the outer traces connecting the outermost turns are approximated

[1]The code was designed in such a way that another type of coil connection can be easily retrofitted.
[2]Printed Circuit Board

by going outward in a radial direction, and then again connecting these pieces with a circular arc around the z-axis. These approximated traces are shown in blue in Figure 3.8 on the right.
The entry and exit points of the superconducting cable of the coil are not at the beginning of the half-turn, but are executed at the end of the penultimate and at the beginning of the second straight section from the coil. Figure 3.9 shows the exiting conductors of a MCOXF coil as an example. For the `DefaultCoilConnection`, the hexahedrons of the half height section and in the 90 degree curve are skipped at the beginning of the first turn. At the last turn the last 90 degree curve and the last half height section are skipped. For this reason, the construction of the conductor hexahedrons in Figure 3.7 starts at this desired point and not at the top center. Now that the entry and exit points are correctly adjusted, the last hexahedrons of one coil can now be connected to the first hexahedrons of the next coil.

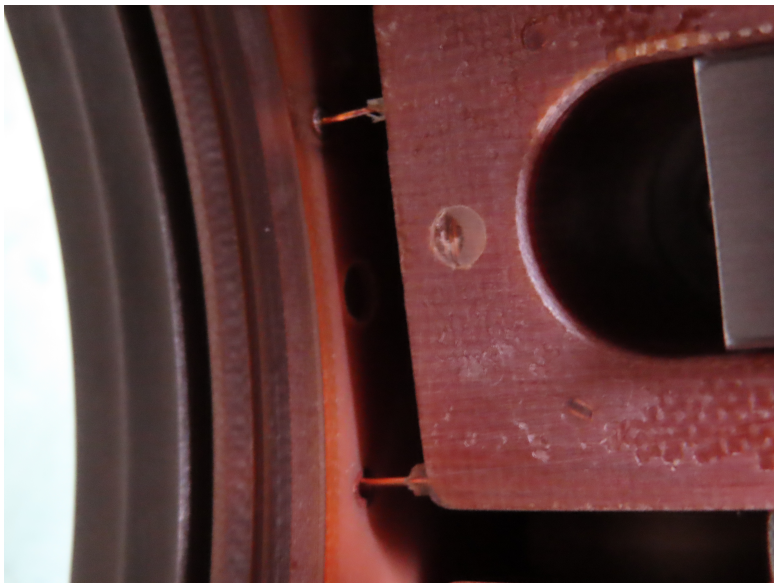


Figure 3.9: Recording of the entry and exit points of the conductor of an MCOXF coil towards the PCB [122].

The connection of the inner conductor ends is accomplished with a circular arc on PCB level. In `DefaultCoilConnection` two parameters are specified for this. The first parameter `distance_to_coil` specifies how far the PCB is away from the coil (along the z-axis) and the second parameter `max_degrees_per_element` defines the maximum hexahedron length (in degrees) in the circular arc. To calculate the vertices of the connecting hexahedrons, first center points are defined through which the conductor runs. Based on these center points the required four vertices are calculated in the matching surface through the center point and finally the hexahedrons can be defined from the vertices calculated in this way. For this the correct numbering of all vertices according to the FiQuS naming convention is necessary. In figure 3.10 start and end hexahedrons, center points and vertices with numbering are shown for the calculation of the vertices of the inner coil connection.

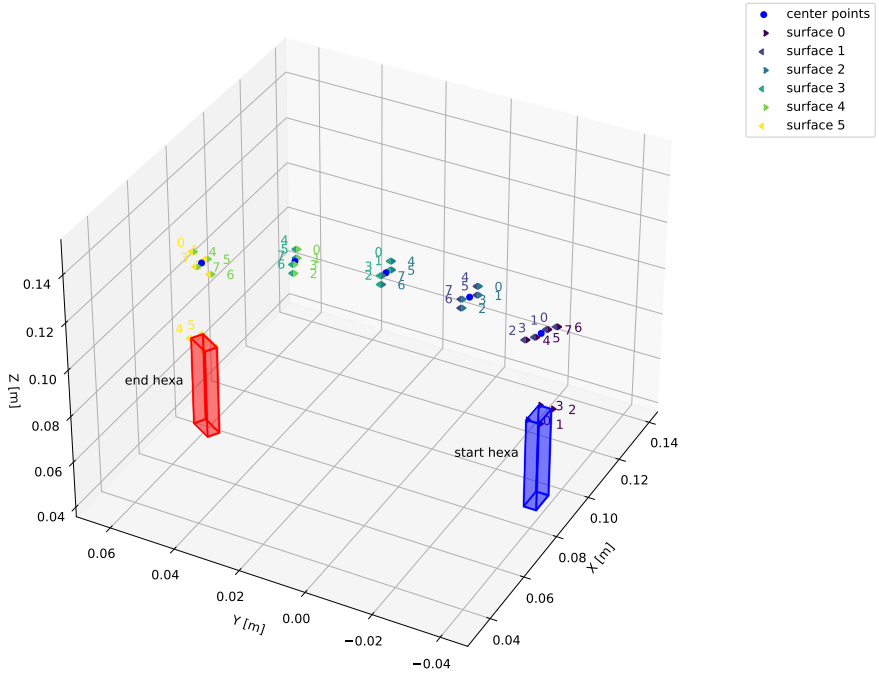


Figure 3.10: Center points and numbered vertices of the hexahedrons start and end faces of an inner coil connection starting from the start hexahedron ending at the end hexahedron.

The first center point C_0 is calculated by shifting the center point of the end face of the start hexahedron (the last hexahedron of the first coil) by `distance_to_coil` in positive z-direction. Similarly, the last center point C_N is obtained by shifting the end face of the end hexahedron (the first hexahedron in the second coil) by the same amount in the positive z-direction. The number of hexahedrons needed for the circle element N is determined by:

$$N = \left\lceil \frac{\Delta\varphi}{\texttt{max_degrees_per_element}} \right\rceil \quad \text{where} \quad \Delta\varphi = \varphi_N - \varphi_0 = \tan\frac{C_{N,y}}{C_{N,x}} - \tan\frac{C_{0,y}}{P_{0,x}}$$

Since Hexahedrons share start and end faces, $N+1$ different faces are needed for N hexahedrons. After the first and last center point of these faces are already defined, $N-1$ points are still needed. The missing center points C_i are calculated by:

$$P_i = \begin{pmatrix} \sqrt{C_{0,x}^2 + C_{0,y}^2} \cdot \cos(\varphi_0 + i \cdot \Delta\varphi) \\ \sqrt{C_{0,x}^2 + C_{0,y}^2} \cdot \sin(\varphi_0 + i \cdot \Delta\varphi) \\ C_{0,z} \end{pmatrix} \quad \text{where} \quad i \in \{1, 2, \ldots, N-1\}$$

Now that all center points are defined, the four vertices around each center point are calculated. Since height h and width w of the racetrack winding are known, the corner points around the center points C_i are easy to calculate. Starting from each center point,

the corner points are calculated by adding $\pm 0.5h$ in z-direction and $\pm 0.5w$ in radial direction ($\vec{e_r} = (\cos(\varphi_i), \sin(\varphi_i), 0)^\intercal$). The vertices of C_0 and C_N are at a kink point of the conductor and are therefore inclined 45 degrees. For this, $\pm 0.5h$ is added to the previous calculation in tangential direction ($\vec{e_t} = (-\sin(\varphi_i), \cos(\varphi_i), 0)^\intercal$). If the obtained points are numbered in FiQuS order and the correct start and end faces are used, the hexahedrons from Figure 3.12a for the inner coil connection are obtained.
For the outer coil connection the calculation of the corner points is very similar. Again, the center points and corner points are shown in the figure 3.11.

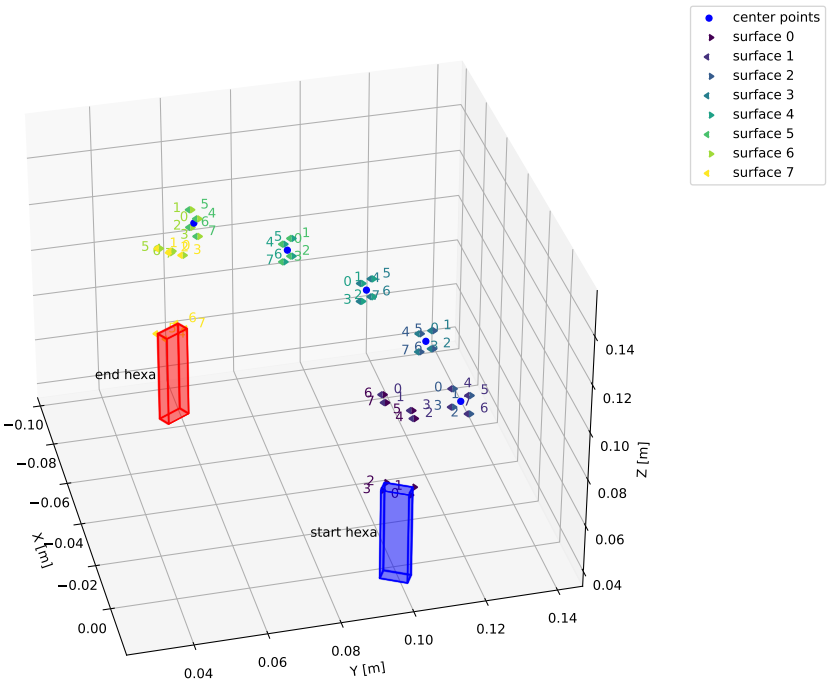
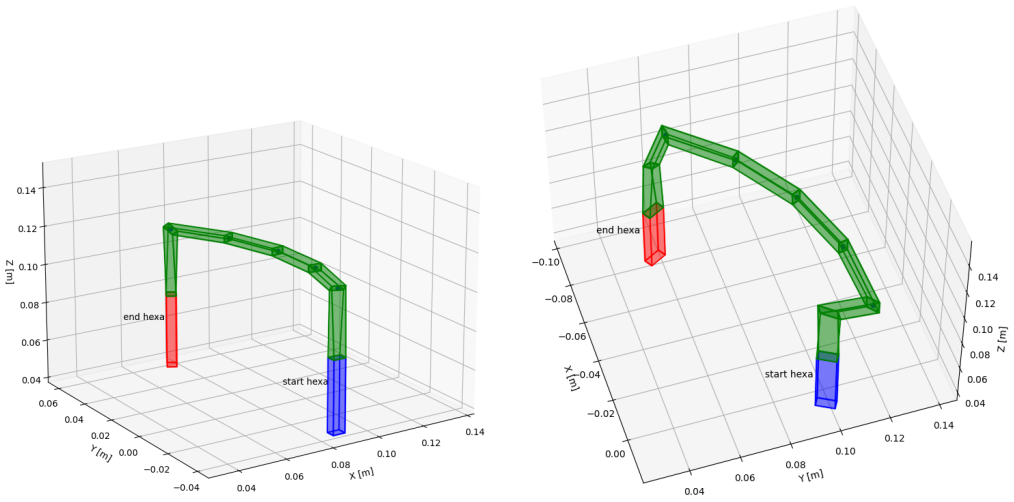


Figure 3.11: Center points and numbered vertices of the hexahedrons start and end faces of an outer coil connection starting from the start hexahedron ending at the end hexahedron.

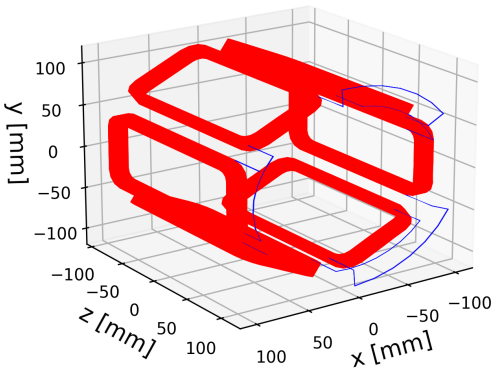
The main difference is that two more kink points have to be introduced. The first and last surfaces are not inclined in tangential direction, but radially outward. To reach the outer radius a hexahedron is added in radial direction. How far the second center point (center of the end face of this hexahedron) is outside is determined by the parameter `radius_outside` from the `DefaultCoilConnection` class. Since at the second center point, as well as at the penultimate center point, a kink point is located, this surface is also inclined. For this $\pm 0.5w$ is added in tangential direction. Different from before at the inner connection the conductor is twisted once more by the second kink, therefore w and h must be exchanged for the circle segment. After numbering according to naming convention, this gives the hexahedrons shown in Figure 3.12b. Depending on which `start_points` are defined for the coils, it can be distinguished whether it is an inner or an outer coil connection.



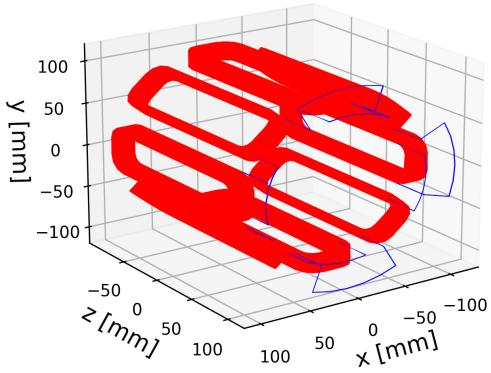
(a) inner coil connection (b) outer coil connection

Figure 3.12: Coil connection hexahedrons for inner and outer coil connection. Last element of first coil in blue, coil connection in green, first element of second coil in red.

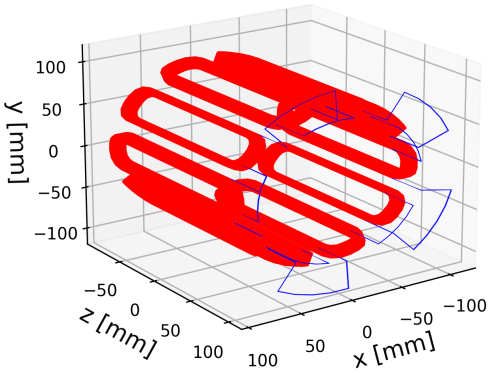
Figure 3.13 shows the complete geometry models of all five HL-LHC HOC magnets. Magnet coils are shown in red, the coil connections are shown in blue. All connection paths were checked for intersections. The final number of hexahedrons per magnet type increases with the harmonic order of the magnet. MCSXF is approximated with 44 930 hexahedrons, MCDXF with 54 714, MCOXF with 77 369 and the dodecapoles MCTSXF and MCTXF with 138 508 and 202 442 respectively.



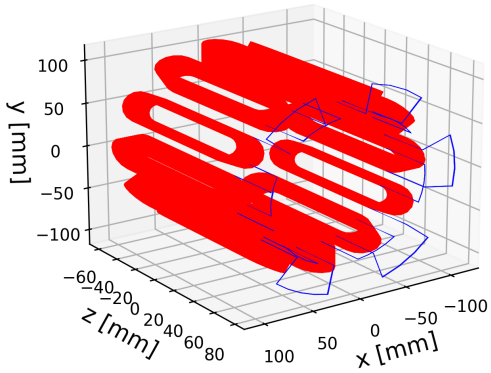
(a) MCSXF



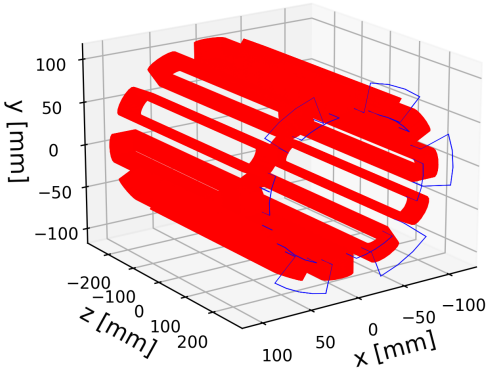
(b) MCOXF



(c) MCDXF



(d) MCTSXF



(e) MCTXF

Figure 3.13: Hexahedron geometry models created in FiQuS for the HL-LHC HOC magnets. Racetrack coils composed of red hexahedrons and coil connections composed of blue hexahedrons.

To better compare the revised geometry with the corrupted one from Figure 3.2, Figure 3.14 shows the top front and right view of MCSXF again. In the previously implemented geometry, the connection of the coils was made with simple arcs, some of which also overlapped.

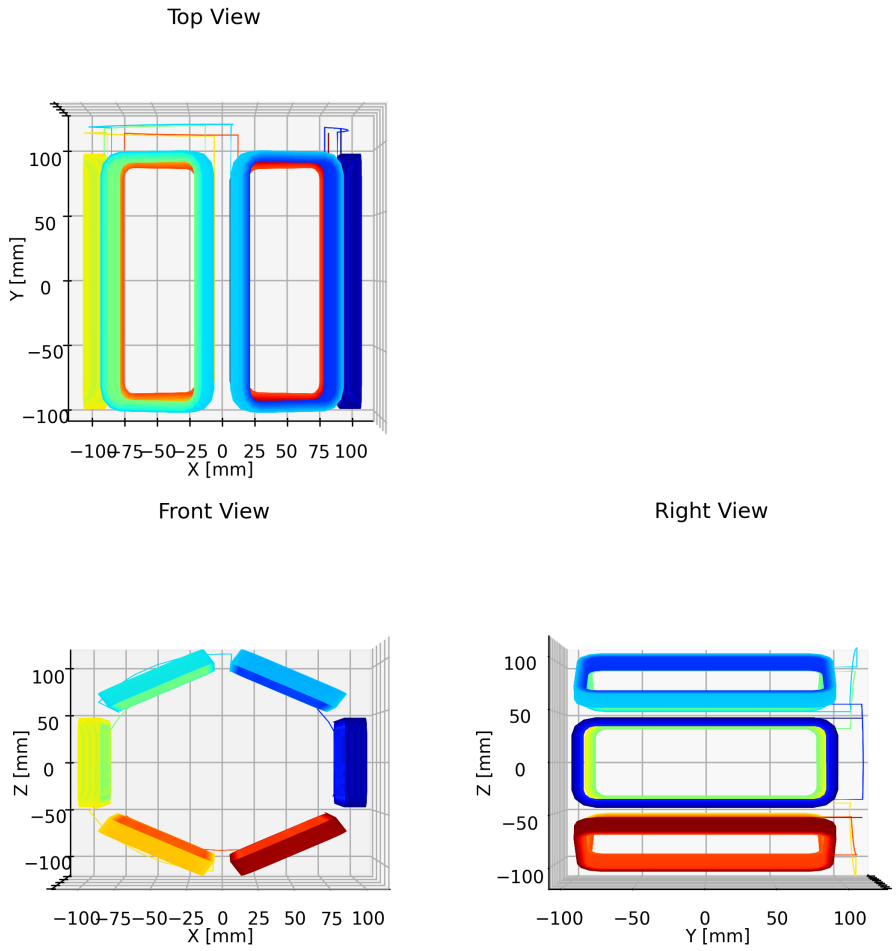


Figure 3.14: Top, front and right side view of the new MCSXF geometry in LEDET displayed in 3D prespective. The color represents the length of the conductor along the electrical direction starting from blue to green then to yellow and ending in red.

3.1.5 Generating LEDET geometry input files

For LEDET 3D simulations the geometry can be given as input by means of a CSV file. This file holds several values for each conductor hexahedron. These are: Element length, element position along the electical order, the point coordinates of the lateral and upper surface of the element (for 3D visualisations), half-turn index in electrical order and index identifying to which ROXIE half-turn the element belongs to, transport current through the element and the magnetic field in the element in x, y and z direction but also

in l, h and w direction (length, height and width direction - specific to every element) [123]. From these values, LEDET can create and simulate the lumped element model of the magnet by calculating, among other things, the mutual inductance of the magnet and all needed parameters for the full semi-implicit finite difference scheme for the thermal model. After subdividing the geometry, these parameters are calculated for each element and stored in a pandas dataframe [124] to be written to the CSV file in the specified format [123]. The magnetic field strength, must thereby be passed to LEDET, because LEDET can only simulate electro-magnetic and thermal transients. The prevailing magnetic field at the beginning of the simulation must be determined with another tool [85]. The following explains how the various required parameters are calculated.

Geometric parameters

The element length is calculated as the distance between the two centers of entry and exit faces $ds = \|(\frac{P1+P2+P3+P4}{4} - \frac{P5+P6+P7+P8}{4})\|$. The element position along the conductor is calculated as a distance through all surface centers of entry surfaces starting with the first entry surface to the center of the element. The elements are traversed in electric order and always half of the previous element length ds_{prev} is added to the current element length ds: $s = s_{prev} + 0.5 \cdot ds_{prev} + 0.5 \cdot ds$. The point coordinates can be easily adopted. The points 1 and 2 for `3Dsurf` are P1 and P2 and the points 1 and 2 for `3Dsurf_2` are P3 and P2 (following the FiQuS numbering). In one row of the CSV file only the start points of the surface are given and in the following row the end points are given. Therefore, a line must be added at the end of the file, which defines the last end points. The remaining values in this line are to be set to `NaN` [123].

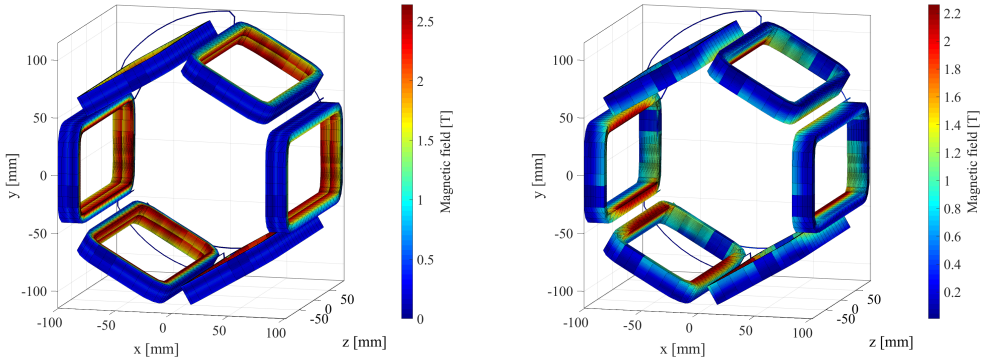
Magnetic field

Calculating the magnetic field of the magnet can be done in different tools, which can be basically divided into 2D and 3D tools. For the case of the HL-LHC HOC magnets, 2D simulations were performed in ROXIE [102] and 3D simulations in OPERA [115].
To adopt the magnetic field strengths from ROXIE, the simulation file in `.map2d` format is read. In this file x- and y-coordinates of the half-turns, their indexes and their magnetic fields and transport current are recorded for the 2D magnetic cross section. With the help of the previously calculated STEAM indexes the correct conductor is found and their field and current values are parsed. The x- and y-components of the 2D field are adopted and the z-component is assumed to zero. This results in the magnetic field which is shown in figure 3.15a as an example for MCSXF.
To adopt the magnetic field strengths from OPERA, first the center points of all conductor hexahedrons are beeing exported to a CSV file. The simulation results of Opera are stored in a `.op3` database and if points are known, they can be exported with a single command. This approach has the advantage that OPERA evaluates the magnetic field at the hexahedron centers using the Biot-Savart law, making it more accurate than an interpolation over exported spatial points. The command used in OPERA to export the magnetic field values can be seen in listing 3.3 and outputs the data in the format: x-coordinate, y-coordinate, z-coordinate, field component in x, field component in y, field component in z , absolute magnetic field.

```
TABLE INFILE='InputFile.table' OUTFILE='OutputFile.table' COLUMNS=7
    F1=X UNIT1T=LENGU F2=Y UNIT2T=LENGU F3=Z UNIT3T=LENGU F4=Bx
    UNIT4T=FLUXU F5=By UNIT5T=FLUXU F6=Bz UNIT6T=FLUXU F7=B UNIT7T=
    FLUXU UNIT8T=1 UNIT9T=1 UNIT10T=1 UNIT11T=1 UNIT12T=1
```

Listing 3.3: OPERA command that exports magnetic field values to an output table (OutputFile.table) for given points in space specified in an input table (InputFile.table).

As datatype of the output table `.mat` is used since there are many parsing functions implemented in STEAM. When generating the geometry input files for LEDET, a bool is introduced to only output the center points. This can then be used to create the `.mat` file and thus generate the geometry with the 3D field from OPERA. This results in the magnetic field shown in figure 3.15b as an example for MCSXF.



(a) 2D ROXIE simulation

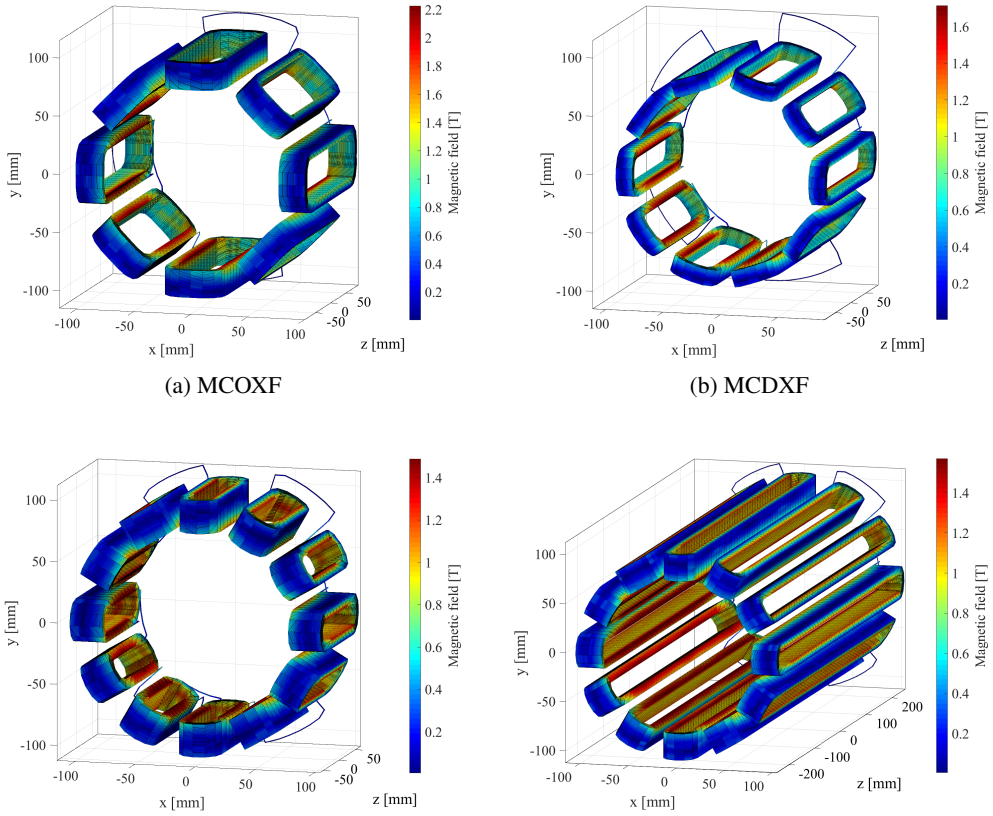
(b) 3D OPERA simulation

Figure 3.15: Absolute magnitude of the magnetic field B in each element of a 3D LEDET model of MCSXF at operating current (99 A) generated by the geometry input CSV file. B derived from different prior simulations.

The geometric model must match both the ROXIE and OPERA models. However, when checking this requirement, it turned out that the two models did not have identical geometry. In the cross-section, the distance between individual conductors in the ROXIE model was significantly smaller than in the OPERA model, which corresponds exactly to the technical drawings [114, 125, 126, 127, 128]. As it turned out, only the insulation of the cable was used as the distance between the individual conductors of a coil in the ROXIE model for reasons of plausibility. However, the individual cables are wound in epoxy, which is irregularly distributed around the cables and results in larger gaps. INFN has determined empirical packing densities for modelling, which were not considered in the ROXIE model.[1]. According to this issue, the ROXIE model was revised, resulting in a larger magentic field. For MCSXF, the maximum value of the field increased from 2.250 T to 2.639 T. In figure 3.15a the revised simulation result is shown.
Comparing the two magnetic fields, it can be seen that the ROXIE model with a maximum magnetic field of 2.639 T has a significantly higher maximum value than the

[1] In chapter 5, this packing density will be discussed in more detail, especially with respect to the thermal influence of the epoxy.

OPERA model with 2.262 T. The older ROXIE model is much closer to the OPERA model. The possible reason for this will be explained in more detail in chapter 5. Also in the distribution it can be seen that the ROXIE model overestimates the magnetic field: At the beginning and end of the magnet, the magnetic field in the 3D model is significantly lower at the inner most turns than when the 2D field is applied for the whole turn. Since this is a comparatively short magnet, these edge effects are expected to have a significant impact on the simulation results.
For the reasons just mentioned, and also because the OPERA simulations performed by INFN have been checked against experimental results, the magnetic field from OPERA is used for the simulations. Figure 3.16 shows the magnetic fields imported from OPERA for the remaining HL-LHC HOC magnets. Their maximum magnetic field at operating current is 2.226 T for MCOXF, 1.713 T for MCDXF, 1.489 T for MCTSXF and 1.570 T for MCTXF.



(a) MCOXF (b) MCDXF

(c) MCTSXF (d) MCTXF

Figure 3.16: Absolute magnitude of the magnetic field B in each element of the 3D LEDET models for the remaining HL-LHC HOC magnets at operating current created by the geometry input CSV file and derived from preceding OPERA simulations.

Comparing the 3D OPERA field of MCSXF and the rest of the magnets, it can be seen that it does not appear continuous in some places. The input file was examined and

visualised separately in Python. No such spots were found. Probably the error is due to the rendering of the plot in MATLAB or to an interpolation in LEDET, since LEDET again interpolates the geometry from the input file. Furthermore, it was recognised in this comparison that all magnetic field values are shifted along the conductor length. Looking at the dark red area in Figure 3.15b, one can see that it is not perfectly placed in the magnet center, but that it seems to be shifted by one element along the conductor. The reason for this is most probable only an affect fo the visualisation, since LEDET displays the magnetic field value on the starting edges but uses the center points for the simulation [123].

Indexing

Since the conductor hexahedrons are passed through in electrical order, the half-turn index in electrical order is easy to assign. If a turn consists of N hexahedrons, the electrical order half-turn index is always incremented after $\frac{N}{2}$ hexahedrons. From the electrical order half-turn index, the corresponding STEAM-ordered half-turn can be calculated. These two index conventions are shown in Figure 3.17 for the case of a sextupole with 2 conductors in the height direction and 3 conductors in the width direction.

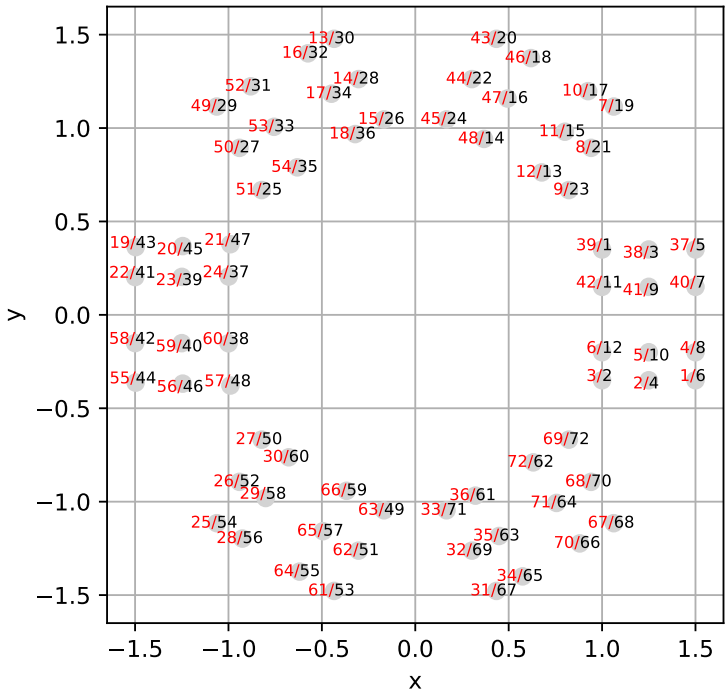


Figure 3.17: Cross section of an imaginary magnet with 2x3 windings to demonstrate different numbering formalities. ROXIE numbering in red, electrical order numbering in black, conductors in grey.

The total number of half-turns n_{ht} is calculated by summing the products of the number of conductors in height and width of each coil, multiplied by 2. The electrical order half-turn index is then assigned for each hexahedron in each coil and stored in a temporary list. These indices are then incremented with the highest index of the previous

coil to avoid overlap. Finally, the list is converted to the STEAM half-turn by adding an offset of $\frac{n_{ht}}{2}$ for alternating coils. The result is a unique mapping of the ladder hexahedra to their corresponding STEAM half-turn index.

3.2 Setting Non-adiabatic parameters

STEAM models have already been generated for the HL-LHC HOC magnets. However, these models are adiabatic and not all information about windings and conductors are given and correct. The following explains how the existing models have been expanded. For non-adiabatic simulations, the thickness of the insulation layers towards the heat sink is of central importance. In a STEAM model, the relative thickness of the insulation layer in all four directions must be defined for each half-turn of a magnet. For half-turns that are not on the outside of a coil and therefore not in contact with the He bath this relative insulation thickness has to be set to zero. For half-turns, that are on the outside however, this relative insulation thickness $f_{cooling}$[] can be calculated as follows:

$$f_{cooling} = \frac{th_{wire}}{th_{coil} + th_{wire}} \tag{3.6}$$

Where th_{wire}[m] is the thickness of the insulation layer of the superconducting wire[1] and th_{coil}[m] is the thickness of the insulation layer of the additional insulation layer around the coil. th_{coil} can be determined form the construction drawing shown in Figure 3.18 for MCSXF. Although the geometry of the coils of the HL-LHC HOC magnets is different, the insulation thickness is the same for all orders [114, 125, 126, 127, 128]. As shown in figure 3.18 for MCSXF this is 1.5 mm towards the aperture, 0.15 mm on the coil inside and 1 mm on the outside of the coil and in on the side facing away from the aperture.

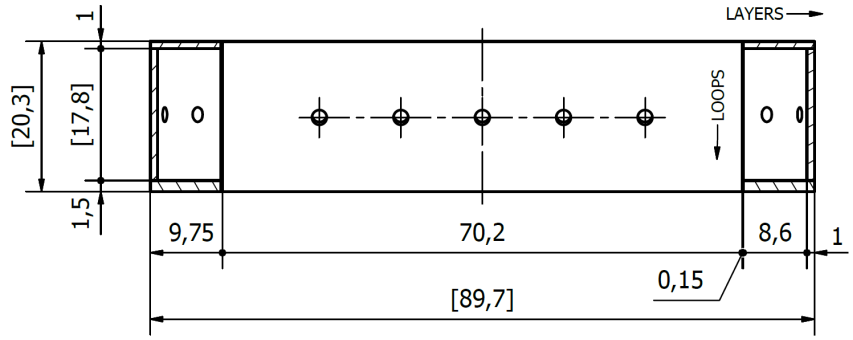


Figure 3.18: Transverse section of a MCSXF coil with insulation including geometrical specifications. Magnet aperture is at the bottom. Figure reproduced from [114].

The scaling factors for the thermal diffusion towards the heat sink $f_{cooling}$ in a STEAM model are recorded in the four lists `sim3D_f_cooling_left`, `sim3D_f_cooling_right`, `sim3D_f_cooling_up`, and `sim3D_f_cooling_down`. The length of these lists corresponds to the number of half-turns. Ordered by electrical sequence, the relative insulation thickness of the first half-turn in the corresponding direction must be written to the first position in the list. Since each magnet has thousands of half-turns, the creation of these

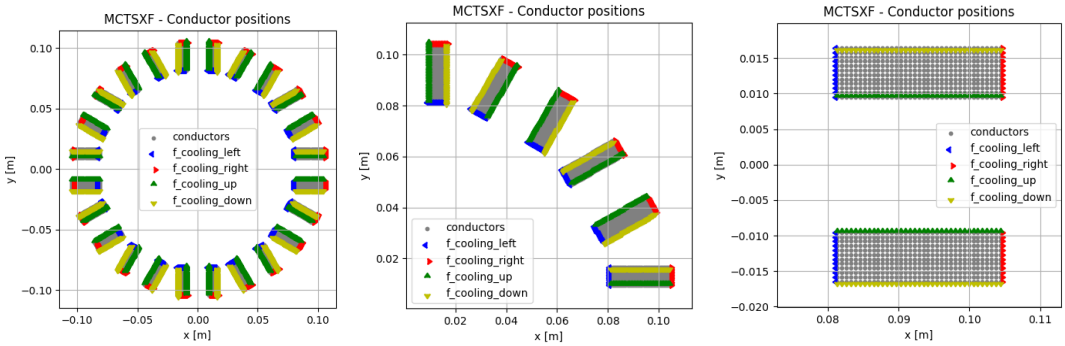
[1] For the wires of all HL-LHC HOC magnets $th_{wire} = 70\,\mu$m.

lists was automated. The numbering pattern corresponds to the STEAM order already introduced in figure 3.17, which was visually checked with the mplcursors library [129]. This numbering convention can be used to determine the filter masks for each direction. The algorithm used for this is as follows [130]:

Algorithm 1 Compute filter masks for all for insulation sides

plot_mask_left ← array
for $i \in [0, \lfloor \frac{\texttt{n_ht}}{\texttt{n_w}} \rfloor]$ **do**
    append "n_w $-1+$ n_w $\times i$" to plot_mask_left
end for
plot_mask_right ← array
for $i \in [0, n_sg]$ **do**
    append "n_w $\times i$" to plot_mask_right
end for
arr_down ← $[0, \ldots,$ n_w$]$
plot_mask_down ← array
for $i \in [0, \lfloor \frac{n_sg}{n_h} \rfloor]$ **do**
    append "arr_down $+$ n_w $\times$ n_h $\times i$" to plot_mask_down
end for
plot_mask_up ← array
for val ∈ plot_mask_down **do**
    append "val $+$ n_w $\times$ (n_h -1)" to plot_mask_up
end for

Where `n_h` stands for the number of conductors in height direction, `n_w` for the number of conductors in width direction and `n_ht` for the number of half turns.
Figure 3.19 shows the filter masks determined in this way using the MCTSXF magnet as an example. The algorithm just described works for all HL-LHC HOC magnets.



(a) Full cross section (b) First quadrant (c) Single Coil

Figure 3.19: Different sized cross-section through MCTXF showing all conductors with masks for insulation parameters needed for non-adiabat simulations.

4 Automatic test campaign validation

4.1 Introduction and motivation

For this thesis, a large number of simulations must be generated and validated based on measurements. If this were done with the current functionality, i.e. with the steps introduced in section 2.1, it would be very time-consuming. For each individual measurement, up to 100 parameters would have to be adjusted. This means that for each measurement a step `ModifyModelMultipleVariables` would have to be defined and all variable names and their new values would have to be noted there manually. In order to compare measurements and corresponding simulations, a `RunViewer` step should also be carried out. The necessary setup CSV file for this step must also be prepared manually. In addition to the large amount of time required, it is also possible that errors may occur during manual insertion, which are very difficult to find. In addition, the YAML input file for the simulation becomes long and therefore unreadable. The aim, however, is to keep this file as clear as possible to ensure user-friendliness.
Adjusting parameters for large magnet test campaigns is not only needed for the validation of models, but has many other use cases. For example, if one wants to compare simulations and measurements to identify unusual behaviour, such as quenches during operation, or if one wants to analyse large groups of tests in a consistent manner, one reaches the limits of the framework. The analysis of large test campaigns is also required by CERN external users, such as magnet manufacturers and designers, e.g. INFN.
The parameters that have to be changed for each test campaign can be divided into conductor specific parameters and event specific parameters. Conductor specific parameters are the parameters that are specific to each conductor in the magnet. This concerns the deviations of the manufactured magnet from its design values. When a magnet is manufactured, measurements and known parameters of each conductor and coil are recorded. Parameters such as the copper-to-no-copper ratio, RRR[2] or the filament-twist-pitch must therefore be adjusted for each individual magnet on the basis of these measurements. Event-specific parameters, on the other hand, are parameters that define the underlying conditions of a simulation. Since each measurement has different underlying conditions, these must also be adapted for every simulation. For example, in each measurement the power converter switches off at a different time, or the different quench heaters trigger at different times. Also, parameters such as coil current or initial temperature have to be adjusted from simulation to simulation.
Due to the problems just highlighted and the numerous use cases outlined, the aim is to automate test campaign analyses. For this automation, a way was found to change both

[2]Residual-Resistance Ratio

conductor and event specific parameters. The implementation is flexible enough to cover all possible cases, i.e. all possible combinations of quench protection systems, different numbers of measurements for different parameters, etc. In section 4.3 the automation of the process to adjust conductor specific parameters is explained and in section 4.4 the same is done for the event specific parameters. For these automations, common denominators were first sought so that the same functionalities do not have to be implemented twice. This common denominator is explained in section 4.2. Finally, section 4.5 gives an overview of how these steps are combined to perform a full test campaign analysis.

4.2 Multiple changes of several variables: ParametricSweep

For both automations, many parameters must be changed for many simulations. For this purpose, a new step was implemented in STEAM-SDK, which is used in both automations and can also be used stand alone.
In order to be able to change many parameters for many simulations, a table representation is particularly suitable in which a row is generated for each simulation and a column for each parameter. To read these tables and write the simulation input files (which can be executed later) the step `ParametricSweep` was implemented in STEAM. This step ignores all columns whose names do not exactly correspond to the name of a parameter in the STEAM model that is being changed. Table 4.1 shows a minimal example of such a table inputted in CSV format. The columns `simulation_name` and `simulation_number` are mandatory fields, as they are needed to run a STEAM simulation. Since a model consists of hundreds of parameters and in principle an infinite amount of simulations can be performed, this file can consist of hundreds of columns and a large number of rows. The stand alone implementation of this step is important not only because it can be used for the two automations in this chapter, but also because the future implementation of parametric analyses will require this step. Already during the model validation, this step proved to be particularly useful for checking the impact of individual parameters.

simulation_name	simulation_number	Power_Supply. I_initial	Quench_Protection. Quench_Heaters. t_trigger
mag1_V1	1	12321	[0.01, -0.04, 0.0, 0.3]
mag1_V2	2	9453	[0.02, -0.06, 0.02, 0.0]

Table 4.1: Minimal example of an input table for the `ParametricSweep` step for two simulations with different initial current and different trigger times of the quench heaters.

The `ParametricSweep` step reads the CSV file, loops through the rows and executes a `ModifyModelMultipleVariables` step for each row. It is assumed that all numerical values are given in the standard system. As already mentioned, only columns are considered whose name is an attribute of the used model class. Depending on whether it is a conductor, magnet or circuit simulation, this model class is either `DataModelConductor`, `DataModelMagnet` or `DataModelCircuit`[1]. The implementation had to take into ac-

[1]This step works for all three models. The steps introduced in the following subsections are for the automation of magnet test campaigns and can therefore only change magnet models.

count that nested attribute names, such as `GeneralParameters.name` (using dot separation), and index accesses, such as `Conductors[2].strand.diameter` (in squared brackets) are read as strings from the CSV file and are addressed correctly. Furthermore, values that are actually of the data type list are read as string, e.g. `'[1.1, 2.2]'` or `'[normal, special]'`. For these cases, separate parsing functions have been implemented. The executed `ModifyModelMultipleVariables` step does two main tasks: it changes the model and it writes simulation input files that can be executed by a `RunSimulation` step for every row. The model in this context is an instance of the `BuilderModel` class within STEAM-SDK. Since it is sometimes desirable to change the model permanently and sometimes not, a bool is passed to the step that specifies whether the model should be changed or restored. If this bool is set to true, changed parameters are restored by copying all parameters before changing them in order to reassign them afterwards.
An example step definition of the `ParametricSweep` step for a magnet model simulated with LEDET is shown in listing 4.1. Under the key input_sweep_file the relative path to the CSV file is defined.

```
par_sweep:
  type: ParametricSweep
  input_sweep_file: input/paramTable.csv
  model_name: BM
  case_model: magnet
  software: [LEDET]
  verbose: True
```

Listing 4.1: Example step definition of a `ParametricSweep` step in YAML format for a magnet model using LEDET

4.3 Automation of conductor specific simulation setup

4.3.1 Introduction of ParsimConductor step

To automate the adjusting of conductor specific parameters for simulations, the step `ParsimConductor` is implemented. How this step reads the conductor data, processes it and finally changes the model is explained in the following. An example step definition of the step is shown in listing 4.2.

```
runParsimConductor:
  type: ParsimConductor
  input_file: input/conductor_database.xlsx
  path_output_sweeper_csv: output/sweeper.csv
  model_name: BM
  case_model: magnet
  magnet_name: mag1 v1
  software: [LEDET]
  simulation_number: 1
  groups_to_coils:
    c1: [1,2]
    c2: [3,4]
  length_to_coil:
    c1: 703.8
```

```
      c2: 704.3
    strand_critical_current_measurements:
      - column_name_I_critical: Ic meas1
        reference_mag_field: 12
        reference_temperature: 4.22
        column_name_CuNoCu_short_sample: 'Ave Cu/noCu in SS'
        coil_names: [ c2, c3 ]
      - column_name_I_critical: Ic meas2
        reference_mag_field: 15
        reference_temperature: 4.22
        column_name_CuNoCu_short_sample: 'Ave Cu/noCu in SS'
        coil_names: [ c1, c3 ]
```

Listing 4.2: Example definition of a `ParsimConductor` step in YAML format for a magnet model using LEDET

4.3.2 Preparing and reading conductor data

During the production of the magnets, measurements are taken for each used conductor and for each wounded coil. Currently, these are recorded in reports in PDF format. In order to make past measurements accessible, the values from the reports must be summarised in CSV[1] files.[2]. A minimal example of such a file is shown in table 4.2. For each coil (with a different conductor) in a magnet a row was written in the file and for each coil or conductor parameter a column was written. Several samples of a conductor can be measured per coil, therefore some values are available as a list (with square brackets).

Magnet	Coil	D [mm]	RRR [Ohm]	weight factor	RT resistance	Ic meas1
mag1	c1	0.821	165.5		0.834	719
mag1	c2	0.804	[190, 193]	[1,1]	[0.73, 0.74]	723

Table 4.2: Minimal example of an input table for the `ParsimConductor` step with two coils "c1" and "c2" for the magnet "mag1". Different parameters (diameter, RRR) and differen measurements (room temperature resistance, critical current) are given.

Initially, the step reads the data of the CSV file into `DataParsimConductor`, a Pydantic [132] data class. A UML[3] diagram of this data class is shown in Figure 4.1. For each coil, i.e. for each row, a new key-value pair is added to the coils dictionary. The key is the name of the coil and the value is a new instance of the class Coil. All values in the table that can be directly assigned to a coil, such as those of the room temperature resistance measurements, are directly assigned to the coil. Values resulting from measurements of a conductor sample are stored with the help of the class `ConductorSample`. For each `Coil`, a list of `ConductorSample` instances is created. As explained in the previous paragraph, there can be several short sample measurements for one coil. To read float or string lists from the table, the parsing functions implemented for the step `ParametricSweep`

[1] Besides CSV format, XLSX (Microsoft Excel Open XML Spreadsheet) format is also supported.

[2] For future measurements, this should also be automated by storing the data in the CERN internal MTF database [131] after the measurement and reading it from there. Read-requests to the MTF API (Application Programming Interface) have already been implemented in Python.

[3] Unified Modeling Language

are reused. A weighting factor is also introduced for each coil. This allows the user to weight the significance of several conductor samples. Before changing the values in the magnet model, the attributes of all `ConductorSample` are weighted according to this list. By default, the arithmetic mean is calculated.

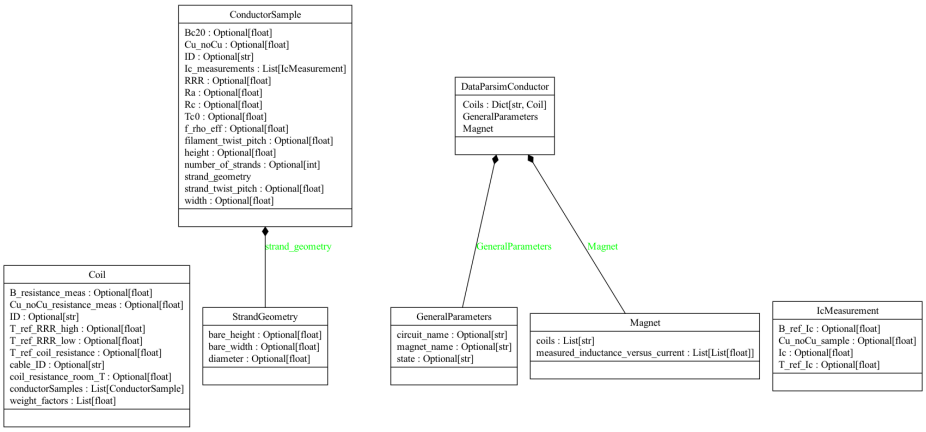


Figure 4.1: UML diagram of the `DataParsimConductor` class used to store conductor specific parameters

Since the input CSV file is not generated automatically and uniformly, different spellings must be permitted for the column names. For this purpose, a YAML file was compiled in which the attribute names of the `DataParsimConductor` class are noted as key and a list of the possible spelling strings is the value. This YAML file separates functionality and data cleanly and also facilitates maintenance, as new notations can simply be added to the file without changing any code. With these notations, different dimensions are also allowed. In the column name, square brackets are searched for and if they are found, the string between them is interpreted as a unit and converted to standard units before being assigned to the instance of the `DataParsimConductor` class.
In addition, the conductor information in the magnet model must be prepared. This information is stored in two lists. In the first list, the conductors of the magnet with their parameters are stored and in the second list it is recorded which STEAM group has which conductor (by the index of the conductor in the first list). However, for many magnets, such as the HL-LHC HOC magnets, only one conductor is defined in the magnet model, since by design, all coils should have the same conductor. But because the parameters of the conductor of each coil have to be changed separately, the defined conductor has to be copied in the magnet model. Since STEAM does not know which STEAM groups make up a coil, the user must provide information about which STEAM groups make up the coil. This is done with the groups_to_coils dictionary in the step definition (see listing 4.2). Based on this dictionary, the conductor is copied for each coil and then changed individually. Also the second list assigning the conductor to STEAM groups has to be adapted accordingly. In the implementation, care had to be taken that a coil can be generated from STEAM groups with different conductors.

4.3.3 Data processing

Many values of the CSV file do not need to be processed further, except for weighting and conversion into standard units, as they directly correspond to parameters of the magnet model. However, this is not the case for all measurements. From the measurements of the critical current, the parameters for the fitting function of the critical current density J_c must be calculated and with the measurement of the resistance at room temperature, parameters with a certain uncertainty must be optimised.

J_c fitting parameters

The critical current density J_c $[\frac{\mathrm{A}}{\mathrm{m}^2}]$ is dependent on the magnetic field B and the temperature T. There are various fitting functions for this calculation, many of which are implemented in the STEAM-material-library and are in the form $f(B,T)$. Common fitting functions include Bottura, CUDI1, CUDI3 for Nb-Ti conductors and Summers and Bordini for Nb_3Sn conductors. These fitting functions are listed below:[1]:

- Bottura fit [133]:

$$J_c(B,T) = \frac{J_{c,ref} C_0}{B} \left(\frac{B}{B_{c2}(T)}\right)^{\alpha} \left(1 - \frac{B}{B_{c2}(T)}\right)^{\beta} \left(1 - \left(\frac{T}{T_{c0}}\right)^{1.7}\right)^{\gamma} \tag{4.1}$$

  where $B_{c2}(T) = B_{c20}\left(1 - \left(\frac{T}{T_{c0}}\right)^{1.7}\right)$ [T].
- CUDI1 fit [134]:

$$I_c(B,T) = (C_1 + C_2 B)\left(1 - \frac{T}{T_c(B)}\right) \tag{4.2}$$

  where $T_c(B) = T_{c0}(1 - \frac{B}{B_{c20}})^{0.59}$ [K].
- CUDI3 fit [134]:

$$J_c(B,T) = \left(c_1 + c_2 B + c_3 \cdot e^{-c_4 B} + c_5 \cdot e^{-c_6 B}\right)\left(1 - \frac{T}{T_c(B)}\right) \tag{4.3}$$

  where $T_c(B) = T_{c0}(1 - \frac{B}{B_{c20}})^{0.59}$ [K].
- Summers fit [135]:

$$J_c(B,T) = \frac{J_{c0}}{\sqrt{B}} \cdot \left(1 - \frac{B}{B_{c2}(T)}\right)^2 \cdot \left(1 - \left(\frac{T}{T_c 0}\right)^2\right)^2 \tag{4.4}$$

  where $B_{c2}(T) = B_{c20} \cdot \left(1 - \left(\frac{T}{T_{c0}}\right)^2\right)\left(1 - 0.31 \cdot \left(\frac{T}{T_{c0}}\right)^2 \left(1 - 1.77 \cdot \ln \frac{T}{T_{c0}}\right)\right)$ [T].
- Bordini fit [136, 137, 138]:

$$J_c(B,T) = \frac{C(T)}{B} \cdot \sqrt{\frac{B}{B_{c2}(T)}} \cdot \left(1 - \frac{B}{B_{c2}(T)}\right)^2 \tag{4.5}$$

  where $B_{c2}(T) = B_{c20} \cdot \left(1 - \left(\frac{T}{T_{c0}}\right)^{1.52}\right)^{-1}$ [T]

  and where $C(T) = C0 \cdot \left(1 - \left(\frac{T}{T_{c0}}\right)^{1.52}\right)^{\alpha} \cdot \left(1 - \left(\frac{T}{T_{c0}}\right)^2\right)^{\alpha}$ [K].

[1] Since in STEAM for CUDI1 the constants for the critical current and not for the critical current density are used, this function is listed here accordingly.

In these equations T_{c0}[K] stands for the superconductor critical temperature, B_{c02}[T] for the superconductor critical magnetic field and the remaining parameters are the sought fitting parameters. So far, the automatic calculation of the parameters for Summers and CUDI1 has been implemented.
Since several critical current measurements can be performed with each conductor sample, a list of instances of the class `IcMeasurement` (see figure 4.1) is implemented in the class `ConductorSample`. In this class all parameters of the critical current measurement are stored. In addition to the measured critical current, this also includes the magnetic flux density and temperature at the time of the measurement. For most fits, the copper-to-no-copper ratio is also required. Namely, to determine the current density from the measured current, it must be divided by the area of the superconductor A_{sup} according to the following equation:

$$J_c = \frac{I_c}{A_{sup}} \qquad \text{with} \qquad A_{sup} = A_{strand} \cdot \left(\frac{1}{CuNCu + 1} \right) \tag{4.6}$$

Where A_{strand}[m^2] is the cross-section area of the strand and $CuNCu$[] is the copper-to-non-copper ratio.
To obtain this additional information about the circumstances of a critical current measurement, the user must specify this information in the step definition under the key strand_critical_current_measurements, as shown in listing 4.2. For the used columns of critical current measurements the user has to define at what temperature and at which magnetic flux density the measurement was performed. Furthermore, the user must specify in which column the corresponding copper-to-non-copper ratio is recorded and for which coils this measurement is to be used. These correlations and background information cannot be read from the CSV file and cannot be integrated into the CSV file in a user-friendly and flexible way.
When calculating the respective fitting parameters, a maximum of three values can be available for each parameter: First, the "default" value from the magnet model, second, the modified value from the CSV file and finally, the value can be specified again in the CSV file for this specific measurement. In reverse order, the first defined value is used for the calculation, as this should be the most accurate. The same applies to the room temeperature resistance measurements.
Depending on the fitting function, a different number of parameters are sought and thus a different number of measurements are necessary to solve the system of equations, where each measurement inserted into the fitting function results in an equation. In general, an error is given for over- or underdetermined systems of equations. However, since CUDI1 occurs very frequently, further considerations were made there. If a measurement is missing to solve the system, the usual ratio of the two parameters C_1 and C_2 for Nb-Ti superconductors is used as the missing equation: $\frac{C_1}{C_2} = -0.0134$. If no measurement for CUDI1 is given, a common value for the two parameters is scaled using the area of the superconductor, where $C_{1,rel} = 10\,257\,504\,\mathrm{Am}^{-2}$ and $C_{2,rel} = -765\,285\,544\,\mathrm{AT}^{-1}\mathrm{m}^{-2}$ was used. For both the ratio and the values per cross-sectional area, the values from MB inner-layer conductor were used.
Depending on the complexity of the fitting function, the system of equations was either converted to its matrix representation and then solved as a linear system or solved with the algorithm `fsolve` from the scipy library [139].

Room temperature resistance measurements

After manufacturing each coil, the electrical resistance at room temperature $R_{RT}[\Omega]$ is measured. The following relationship applies to this resistance:

$$R_{RT} = \rho \cdot \frac{L}{(f_{Cu} \cdot A_{cond})} \cdot f_{tp} \qquad \text{with} \qquad f_{tp} = \frac{\sqrt{w^2 + (0.5 \cdot l_s)^2}}{0.5 \cdot l_s} \tag{4.7}$$

Where ρ [Ωm] stands for electrical resistivity, L[m] for the total length of the conductor of the coil, f_{Cu}[] for the fraction of copper, A_{cond}[m^2] for the total area of the conductor, f_{tp} for strand twist pitch correction factor, w[m] for the bare conductor width and l_s[m] for the strand twist pitch.
If the parameters of the new magnet model are inserted into the equation 4.7, the measured value is never exactly obtained. This deviation between measurement and calculation can be compensated by optimising a value from the equation with the measurement. The two values that could not be precisely known or even unknown are L and f_{Cu}. A_{cond} and f_{tp}, on the other hand, can be determined from precisely measurable geometric values, and the parameters required for ρ are also known and can be used in the ρ function from the STEAM-materials-library.
If the length of the coils is known exactly to the user, the length of each coil can be inserted into the length_to_coil dictionary in the step definition (see listing 4.2). In this case, f_{Cu} is optimised and then converted to *CuNCu*, since this is the parameter in the model[1]. For this, the equation 4.7 is transformed to f_{Cu} and solved for each resistance measurement. Then the mean value of the individual calculated copper-to-no-copper ratios is calculated.
If the length of the coils is not known, as is the case with 2D models that assume equal lengths for all half-turns, the length_to_coil dictionary can also be left empty in the step definition. In this case the length of the coil is optimised. Since the length of a half-turn[2] is specified in the magnet model for each STEAM group, the equation 4.7 must be transformed to L for each coil and then divided by the number of half-turns of the coil. To do this, the number of half-turns of the coil is read from the magnet's .map2d file using the groups_to_coils dictionary. Again, if there are several measurements, the average of the halfturn lengths is calculated.

4.3.4 Applying the changes to the model and implementation

Now that all values have been calculated, the STEAM magnet model must be changed accordingly. The already implemented `ParametricSweep` step is used for this by writing a CSV file that can be used as input by this step. In this file, a column is written for each parameter to be changed with the exact attribute name of the parameter in the magnet model. It must be noted that for each coil the correct index of the conductor in magnet model is used in the column name. This is done with the groups_to_coils dictionary. Since all parameters have to be changed in one step, the file in this case has only one row of values. In table 4.3 a minimal example of such a CSV file is shown. Since the

[1] These two parameters can be converted into each other, because $f_{Cu} = \frac{A_{copper}}{A_{total}}$ and $CuNCu = \frac{A_{copper}}{A_{sup}}$ where $A_{total} = A_{copper} + A_{sup}$.

[2] A half-turn is basically a half of a turn of the coil. A magnet can be considered on the basis of its half-turns, as one usually has geometric cross-sections at hand.

magnet model is to be permanently modified with the updated conductor data, the bool to restore the model to the original values after each row is set to false.

simulation _name	simulation _number	Conductors[0]. strand.diameter	Conductors[2]. strand.RRR	Conductors[1]. Jc_fit.C1_CUDI1
mag1_V1	1	0.821	190.0	66914.5

Table 4.3: Minimal example of a input table for a `ParametricSweeper` step created automatically by the `ParsimConductor` step. Strand diameter, RRR and the CUDI1 parameter C_1 of different conductors are changed at once.

The functionalities of the `ParsimConductor` step were all tested several times with different magnets. Existing values calculated in different ways could be reproduced exactly. These tests were further implemented in a CI/CD[1] pipeline in GitLab to continuously check the integrity of the calculations.
To be able to execute the step easily, the class `ParsimConductor` was implemented. A UML diagram of this class is shown in Figure 4.2. To execute this step, first the method `read_from_input()` and then the method `write_conductor_parameter_file()` is executed from the outside. This way the step can be executed in a few lines. The first method reads the data from the CSV file with the private read methods and the second method writes the parameters with the private write methods into the CSV file for the `ParametricSweep` step. For parameters that need to be calculated, the write functions call the private calculate functions.

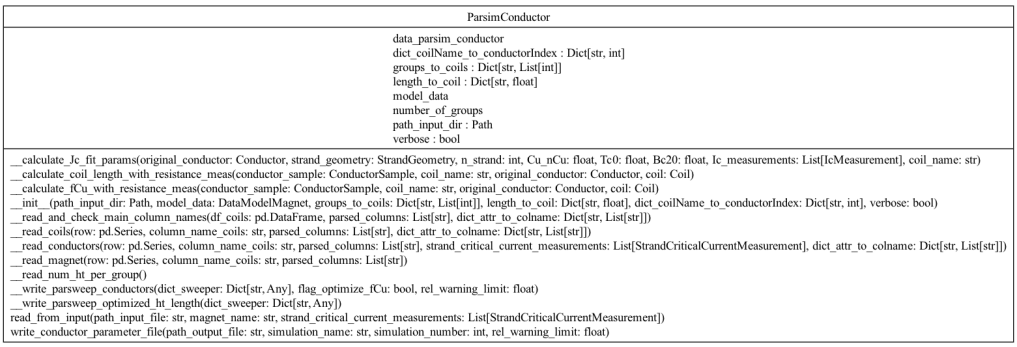


Figure 4.2: UML diagram of the `ParsimConductor` class used to run a `ParsimConductor` step

4.4 Automation of event specific simulation setup

4.4.1 Introduction of ParsimEvent step

As already shown, many parameters have to be set for the simulation of a quench event. The `ParsimEvent` step was implemented to automate this process. In the following it is explained how this step reads and processes data, how it writes the setup file for the `RunViewer` step and how simulation input files are finally generated. An example step definition is shown in listing 4.3.

[1] Continuous Integration and Continuous Delivery

```yaml
runParsimEvent:
  type: ParsimEvent
  input_file: input/QuenchDictionary.xlsx
  path_output_viewer_csv: output/setUpViewer.csv
  path_output_event_csv: output/sweeper.csv
  simulation_numbers: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
  model_name: BM
  case_model: magnet
  simulation_name: mag1
  software: [LEDET]
  t_PC_off: 0.02
  rel_quench_heater_trip_threshold: 0.99
  dict_QH_circuits_to_QH_strips:
    U(PS_1)_YT111: [ 1,5 ]
    U(PS_3)_YT112: [ 2,6 ]
    U(PS_4)_YT113: [ 3,7 ]
    U(PS_2)_YT114: [ 4,8 ]
    U(PS_5)_YT121: [ 12,16 ]
    U(PS_7)_YT122: [ 11,15 ]
    U(PS_8)_YT123: [ 10,14 ]
    U(PS_6)_YT124: [ 9,13 ]
  default_keys:
    local_LEDET_folder: 'steam\steam_sdk_data\LEDET'
    path_config_file: '../../configurations/config_file.yaml'
    default_configs: ['conf_1', 'conf_2']
    path_tdms_files: 'sm18-magnettest\Data_TDMS\'
    test_campaign_name: 'HCLMBRDP001-CR000001_2'
    path_output_measurement_files: 'steam\measurement_database\'
    path_output: 'output/figures'
```

Listing 4.3: Example definition of a `ParsimEvent` step in YAML format for a magnet model using LEDET

4.4.2 Reading event data

For measurements in SM18, a CSV file is automatically written for each measurement, which records the parameters of each event. This CSV file is called QuenchDictionary by SM18 and a minimal example of such a file is shown in table 4.4. The columns here represent various parameters and other informations about the measurement and each row represents a measurement. This file usually has tens of columns and hundreds of rows.

File name	Temperature	QH Name	QH Start time	QH Voltage
f202211241342	1.902	Heater1; Heater2	-0.014; 0.003	885.9; 882.5
f202209120816	1.91	Heater1; Heater2	-0.004; 0.003	859.6; 855.9

Table 4.4: Minimal example of a QuenchDictionary input table for a `ParsimEvent` step. Initial temperature and Quench Heater properties for two measurements are recorded.

Analogous to `ParsimConductor`, the data from the CSV file is read into a Pydantic [132] data class. This class is called `DataEventMagnet` and stores all data of an event. So for each row in the input CSV, an instance of this class is stored in a list. Figure 4.3 shows a UML diagram of this class. In the subclass `Powering` all information about

the power supply is stored, in `Circuit` about the circuit, in `General` general parameters are stored, in the class `QuenchProtection` parameters of common quench protection systems (Quench Heaters, CLIQ, Energy Extraction) can be stored and in `Quench` information like location and start time of the quench itself can be stored. Some of the used classes could already be adopted from other STEAM-SDK data classes.

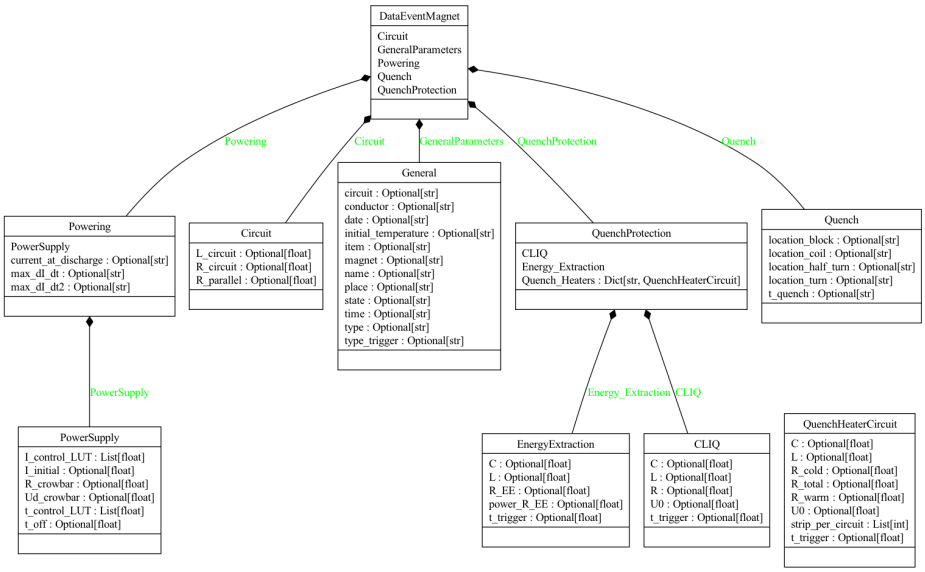
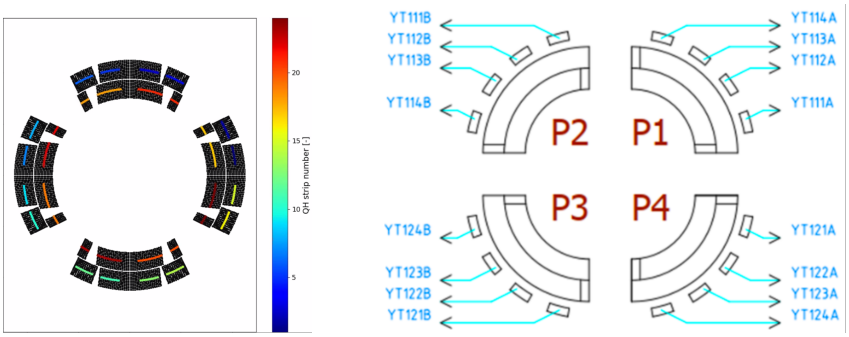


Figure 4.3: UML diagram of the `DataEventMagnet` class used to store event specific parameters

For reading in the parameters, a YAML file is used again, which records possible spellings of a parameter in the CSV file. The `ParsimEvent` step should not only be used for QuenchDictionaries from SM18, but should also be flexible enough to read files from other institutions, such as KEK or INFN. In this YAML again a dictionary is stored in which each key corresponds to an attribute name of a parameter in `DataEventMagnet` and each value is a list of possible spellings. If a new input file has to be read, the new column names can simply be entered into this YAML and no code has to be changed.
Furthermore, some parameters, such as those for CLIQ or the Quench Heaters, are spread over several columns. As can be seen in table 4.4, the names of the Quench Heaters are defined in one column and then the parameters of the individual Quench Heaters in the following columns. When reading, this is taken into account and for each read Quench Heater an instance of the class `QuenchHeaterCircuit` is created and stored with its name in the Quench_Heaters dictionary of the `QuenchProtection` class (see figure 4.3). For CLIQ parameters, for example, there is a column in which the names of parameters with semicolon separation are defined and in another column the values of these parameters with semicolon separation are listed. This has also been implemented in the reading functions. However, the names of these parameters are hardcoded and not in an external YAML file, as it is assumed that different parameters are normally already to be found in different columns.
Furthermore, in STEAM models it must be assigned which Quench Heater is attached to what parts of the magnet. This poses a difficulty for automation, as only the names of the heaters can be read from the Quench Dictionary. To read the information specify-

ing to which magnet turns these heaters are attached, the dict_QH_circuits_to_QH_strips dictionary was included in the step definition. For each quench heater name (key), this dictionary records the STEAM heating strip positions to which it is attached (value). These heating strip positions are numbered[1] in STEAM. Figure 4.4a shows the numbering of the strips for the magnet MQXFB02 as an example[2]. Figure 4.4b shows the arrangement of the quench heaters for this magnet taken from a design report of the magnet. In figure 4.4b, the names of the heaters can be seen and where they are placed, and in figure 4.4a, the corresponding STEAM number of this location can be read. By matching names with locations in these plots, dict_QH_circuits_to_QH_strips can be obtained. The correct dict_QH_circuits_to_QH_strips for this example is defined in listing 4.3.



(a) STEAM strip numeration

(b) Location of named heaters [140]

Figure 4.4: Cross secions of the magnet MQXFB02 with Quench Heater information needed to set up the dict_QH_circuits_to_QH_strips entry in the `ParsimEvent` step definition.

4.4.3 Data processing

To measure the quench heater trigger time the discharge voltage is often considered. Since this signal can be noisy, it could be that in practice a relative threshold $\gamma[]$ for the trigger time detection in the voltage is used to avoid false detection. For example, looking at the blue discharge curve and the red cross in Figure 4.5, it can be seen that the trigger time is not measured correctly because of the delay caused by γ.
For example, if γ is 99%, the measured trigger time t_m[s] is the time at which the voltage has already dropped by 1% and not the true trigger time t_t[s]. Also the measured quench heater voltage U_m[V] is only measured at the time of detection. The correct voltage U_t[V] can consequently be calculated by resolving the definition $\gamma = \frac{U_m}{U_t}$ to U_t: $U_t = \frac{U_m}{\gamma}$.
To correct the trigger time, the exponential discharge curve of the supplying capacitors must be considered. This is defined as follows:

$$U(t) = U_0 \cdot e^{-\frac{t+T_s}{\tau}} \qquad \text{with} \qquad \tau = C \cdot R \tag{4.8}$$

[1]Usually they start on the right side and are numbered counter-clockwise, but with `steam-notebooks/steam-ledet-input/MagName/sigma2ledet_MagName.ipynb` this can be checked. The implementation of this plot in STEAM is planned.

[2]Note that in figure 4.4a also inner layer quench heaters are numbered. Howevery, in the baseline of MQXF there are no heaters on the inner layer but for research purposes they ware added in the past.

Where U_0[V] stands for the voltage at $t = -T_s$, T_s[s] for the time shift, τ[s] for the discharge time constant, C[F] for the capacitance of the capacitors and R[Ω] for the resistance of the entire quench heater circuits.
Substituting equation 4.8 into the definition of γ, the unknowns U_0 and T_s are eliminated from the equation and the delay of the measurement $\Delta t = t_m - t_t$[s], can be calculated:

$$\gamma = \frac{U_m}{U_t} = \frac{U_0 \cdot e^{-\frac{t_m+T_s}{\tau}}}{U_0 \cdot e^{-\frac{t_t+T_s}{\tau}}} = e^{\frac{1}{\tau}((t_t-T_s)-(t_m-T_s))} = e^{\frac{t_t-t_m}{\tau}} = e^{-\frac{\Delta t}{\tau}} \quad \Rightarrow \quad \Delta t = \tau \cdot \ln \gamma^{-1}$$

The true Quench Heater trigger time is then simply $t_t = t_m - \Delta t$.
Since γ can vary and is not commonly known, the entry rel_quench_heater_trip_threshold was included in the step definition. For example, in listing 4.3 this value was set to 99%. In figure 4.5 the corrected value for the shown measurement is marked in green and the interpolated curve in grey. Even if the measurement and the interpolated curve do not match perfectly, the result is still satisfactory because the corrected point would only change marginally.

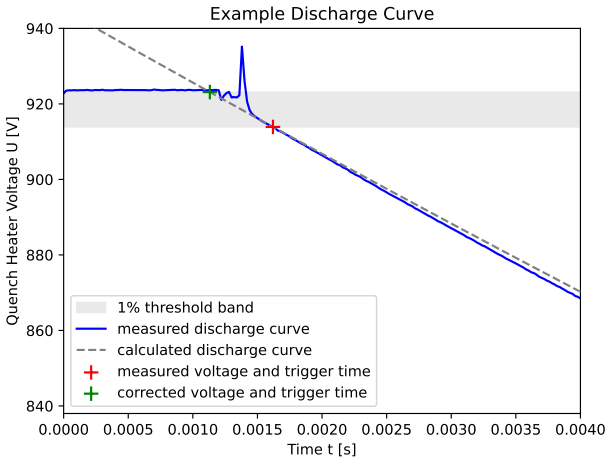


Figure 4.5: Example of a voltage discharge curve of a quench heater. High frequency measurement of a MBRD Heater in blue. Threshold measured trigger time and voltage in red. Approximated discharge curve in grey. Corrected trigger time and voltage in green.

In addition to these calculations, some less complex aspects are also adapted. For example, the time vector for the simulation, which defines the intervals and their step size, is adapted to the trigger times of the quench heaters. This ensures that the simulation always starts before the first trigger of a quench heater. To prevent simulations from starting extremely early due to incorrect input data, a minimum value of -2 s is set for the simulation start time. In addition, the parameter cold resistance R_{cold}[Ω] is calculated for each Quench heater that is not directly given and is needed in the model. The following formula is used for this:

$$R_{cold} = \frac{\rho}{w \cdot h} \cdot l \cdot f_{cover} \tag{4.9}$$

Where ρ [Ωm] stands for the specific electrical resistance of the heater, w[m] for the cross-sectional width of a heater, h[m] for the cross-sectional height of the heater, l[m]

for the length of the heater and f_{cover}[] for the fraction of strip length covered by heating stations, i.e. not copper-plated.

4.4.4 Setting up Viewer configuration files

Hundreds of simulations can be carried out in a test campaign. Due to this large number, it is important to automate the visualisation of the simulation data and, if necessary, a comparison with the measurement data. Therefore the `ParimEvent` step generates an input CSV file for a `RunViewer` step. Before this automation can be discussed, it must be explained how a `RunViewer` step works. This step essentially uses two files to generate plots: a configuration file in YAML format that determines which signals are plotted and a CSV file that determines which simulation is plotted with which measurement and with which configuration. The generated plots are saved individually in the file formats specified by the user in the `RunViewer` step definition and finally summarised in a PDF and/or HTML[1] report.
Signals are defined in the config file, for each of which `RunViewer` generates and saves a plot. For a signal, the names of a simulation variable and a measurement variable can be defined. Several variables can be added together and basic calculations can be performed. Furthermore, plotting options such as titles, intervals and labels are specified for each signal. These signals are then combined into configurations. Thus, in a configuration file there are several configurations that define which plots are created. The preparation of this configuration file is not automated for several reasons[2].
The input CSV file, often referred as the Viewer setup file, stores all other information necessary for the generation of the plots. In this file, a row is created for each measurement-simulation pair and columns represent different parameters. These parameters can be summarised as follows:

1. Path to the config file and name of the configuration to be used: This determines which plots are created.
2. Path to the measurement data folder and file-& campaign name of the measurement[3]: This determines which measurement is plotted.
3. Flag and path for the conversion of the measurement data: This determines if the measurement data (present in TDMS[4] format) should be converted to CSV and where the CSV files should be saved.
4. Path to the simulation data folder and simulation file name: This determines which simulation is plotted.
5. Path where the generated plots and reports are saved.

In order to automate the writing of this CSV file, these five points will now be addressed. For this, some extra information is needed from the user, which he has to specify in the step definition of the `ParsimEvent` step under the entry default_keys.

[1] Hypertext Markup Language

[2] For one thing, these files have already been prepared for almost all the magnets in the STEAM library, and for another, this involves an immense amount of work and may not always work perfectly. Predicting which combination of variables a user wants to compare is only possible to a limited extent. In addition, it must be possible to deal with a large number of exceptions and a large number of possible spellings of different variable names without mix-ups

[3] The files of the measurement data are combined in campaign folders and these folders are again combined in a measurement data folder. The data is stored in TDMS format.

[4] Technical Data Management System

1. The path to the used config file cannot be determined and is therefore provided by the user in the default_keys. Furthermore, the user specifies a list of configurations to be used. For each measurement-simulation pair, this list is processed from the beginning and it is checked whether all the required variables for a configuration can be found in the measurement and simulation. The first matching configuration is written into the viewer setup file. Therefore, the configurations should be ordered from the most detailed to the least detailed. Since measurement data can be given in TDMS as well as in CSV format, this functionality must be implemented for both cases.
2. In each line of the Quench Dictionary there is a column with the file name of the measurement. Since these files are often generated automatically, it is assumed that a timestamp in the format YYYY/MM/DD/HH/mm is present in all file names. This timestamp is extracted and used to search for the file. In the default_keys, the user specifies the path of TDMS and CSV measurements folders. In all subfolders all filenames are searched for the timestamp and in case of a match the path to the parent folder, the name of the measurement campaign and the file name are written into the setup file for the `RunViewer` step. The TDMS file is searched for first. If this is not found, the CSV file is searched for and if this is also not available, no measurement will be plotted. A configuration with only simulation data should therefore be defined.
3. If the measurement data are in TDMS format, the flag for conversion is set to true and if only CSV data are available to false. The path to store the CSV files is read from the default_keys.
4. Since the simulations are generated in this step, their paths and file names are known.
5. The output path for the plots has to be inserted to the default_keys.

4.4.5 Applying the changes to the model and implementation

Analogously to `ParsimConductor` (see section 4.3.4), the `ParametricSweep` step is used to write simulation input files. This time, however, the bool to restore the model to the original values is set to true because each line stands for an event and if a parameter is not specified, the default value from the model should be taken and not a value from a previous event. In summary, `ParametricSweeper` is primarily[1] used by `ParsimConductor` to change the model and is used by `ParsimEvent` only to write simulation input files. Another difference to previous usage is that that the parameters are not adjusted once, but several times for every event. The CSV file for the `ParametricSweep` step therefore has as many rows as the input file for `ParsimEvent`. The step is automatically run by `ParsimEvent`. Table 4.5 shows a minimal example of such an `ParametricSweep` input file created by `ParsimEvent`.
To simplify the execution of the step, the class `ParsimEventMagnet` was implemented. A UML of this class is shown in Figure 4.6. To execute the `ParsimEvent` step for magnets the three public functions `read_from_input()`, `write_event_file()` and `set_up_viewer()` must be executed in sequence from the outside. This way the step can be executed in a few lines. The first method reads the data of the CSV file into an

[1] The simulation input files created with `ParsimConductor` can also be executed if desired.

simulation _name	simulation _number	General Parameters. T_initial	Quench_Protection. Quench_Heaters. t_trigger	Quench_Protection. Quench_Heaters. U0
mag1	1	1.902	[-0.015, 0.002]	[894.8, 891.4]
mag1	2	1.91	[-0.005, 0.004]	[868.3, 864.5]

Table 4.5: Minimal example of a input table for a `ParametricSweeper` step created automatically by the `ParsimEvent` step. Initial Temeprature and Quench Heater trigger time and voltage are changed for two events.

instance of the data class `DataParsimEvent` with the private read methods and corrects trigger time and voltage of the quench heaters. The second method then writes the CSV file for the `ParametricSweep` step using the calculate function. After this function is executed a `ParametricSweep` step is automatically run with the generated input file. The third method writes the setup file for the `RunViewer` step and uses the private get functions to determine the most detailed suitable configuration for different file types.
The functionalities of the `ParsimEvent` step were tested with several campaigns. These tests were also integrated into a CI/CD pipeline.

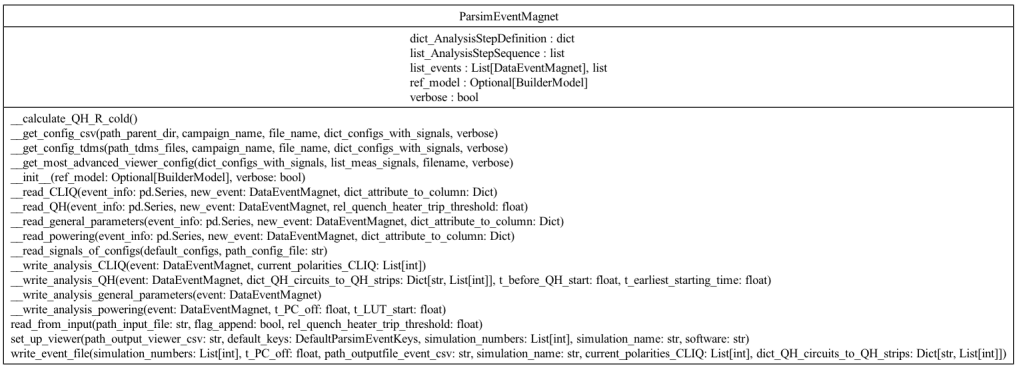


Figure 4.6: UML diagram of the `ParsimEventMagnet` class used to run a `ParsimEvent` step

4.5 Full test campaign analysis procedure

In the following, the procedure of a complete test campaign analysis, as it was also carried out for the HL-LHC HOC magnets, is explained. Figure 4.7 schematically shows the sequence of the steps used and how they work.
To begin with, the folders must be prepared and the model generated from the STEAM-models library. The steps `SetUpFolder` and `MakeModel` are used for this. Then the conductor parameters of the model are adjusted. To do this, the `ParsimConductor` step, which is highlighted in blue in the illustration, is executed. This reads the conductor measurement data from a CSV file into a local data class and writes an input file for the `ParametricSweeper` step which is executed automatically. This step modifies the model and writes a simulation input file that can be executed later.

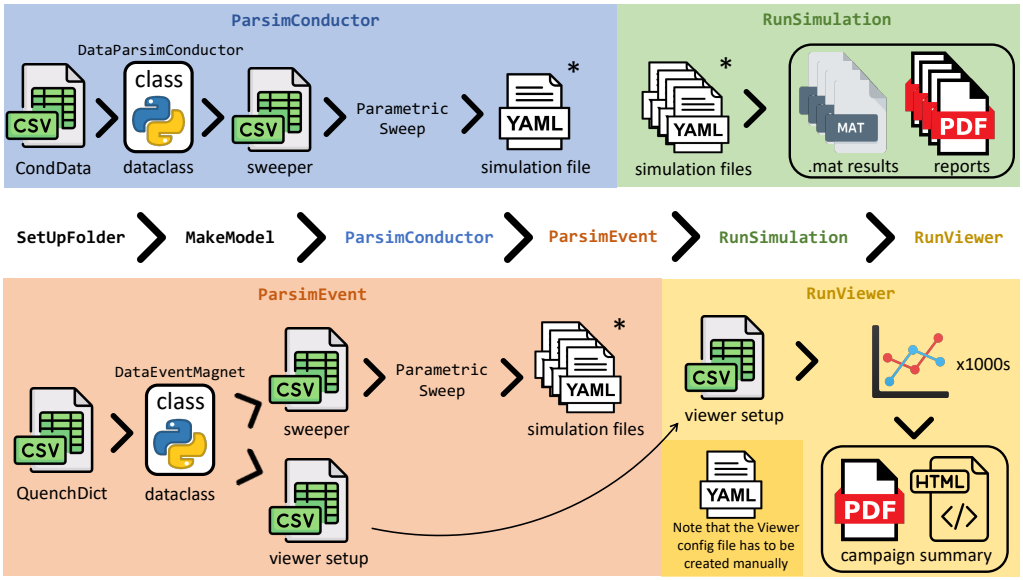


Figure 4.7: Schematic of a complete test campaign analysis using LEDET including output and input files. Sequence of steps used in the centre of the image. More detailed schematic of steps in coloured background. Filetype figures from [141, 142, 143, 144, 145, 146]

After the conductor parameters of the model have been adjusted from the design values to the measured values, the simulations for all quench measurements of the magnet are prepared. For this purpose, the step `ParsimEvent`, which is highlighted in orange in the figure, is used. It reads the necessary simulation parameters from a CSV file into a local data class. The step then writes an input file for a `ParametricSweep` step and executes it. This writes a simulation input file for each quench measurement performed. Furthermore, the step writes the setup file for a `Viewer` step, which is executed below.
Once the simulation input files have been generated, the simulations can be run. The step `RunSimulation`, highlighted in green in the illustration, is used for this. The simulation numbers of the desired simulations can be passed to this step. It then reads the simulation input files that `ParsimConductor` and `ParsimEvent` have written (marked with * in the figure) and starts the simulations with the specified tools. Depending on the tool, the simulation results are then available in different file formats. In the illustration, for example, LEDET was specified, so the results are output in .mat format. LEDET also writes a PDF report for each simulation in which the most relevant results are plotted.
Finally, a `RunViewer` step is performed, which is highlighted in yellow in the figure. The setup file automatically generated by `ParsimEvent` writes the specified plots for each measurement simulation pair. Since `ParsimEvent` can run hundreds of simulations and for each simulation tens of plots can be defined, this step may need to create and save thousands of plots. After these plots are generated, they are summarised in a PDF and in an HTML file. In this campaign summary, all desired signals for all measurements can now be compared visually. Note that the definition of these desired signals must be manually defined once in a configuration file.
With the simulation setup just described, large test campaigns can be automatically simulated and evaluated in one go.

5 Identification of model assumptions

5.1 Introduction, conductor parameters adjustments and analysis of measurement data

For the simulations some assumptions have to be made, because on the one hand not every detail can be modelled accurately in LEDET and on the other hand there are unknown parameters. In this chapter assumptions are made, validated and justified by suitable simulations and measurements or. These assumptions are, for example, whether a quench propagates from one coil to the next or how the magnet circuit is approximated. In the following chapter 6 the simulation results are validated for all magnets and the influence of the quench location is shown again.
Before the validation of the model assumptions can be started, the measured data are first analysed. For each magnet type, except MCTSXF, there are several measured quenches at different initial currents. These measurements were performed on different magnets. For example, for MCOXF there are measurements with MCOXF05 and MCOXF12. 05 and 12 are the model numbers of two built magnets of the type MCOXF. Since for each coil was recorded from which cables the coils were wound, a `ParsimConductor` step can be executed. The conductor database is prepared manually since documentation was on paper and scanned in from cable property sheets [147]. Winding documents for every coil in every magnet were used to find out which conductor was used for which coil (by ID) [148][2]. The correct order of the coils for every magnet can be found in the FAT[3] repots [149][4]. For each coil of each individual magnet critical current measurements, RRR, copper-to-superconductor-ratio, filament-twist-pitch and room-temperature-resistance-measurements are taken. Copper-to-superconductor-ratio ranges from 2.18 to 2.265; critical current at 4.2 K and 5 T ranges from 173 A to 182 A; filament-twist-pitch ranges from 14 mm to 16.5 mm and RRR ranges from 168 to 218. The room-temperature-resistance-measurements were used to control the known copper-to-superconductor-ratio and the already known conductor length. Minimal relative errors in the range of 10^{-2}-10^{-5} were found.
When training/testing a magnet it normal to linearly ramp up or ramp down the magnet current, while a quench during a plateau would be the exception. Thereby, the constant slope of the current is called dI/dt $[\frac{A}{s}]$ and is automatically determined from the measured data and a linear regression analysis. The current at the quench starting time can

[2]Citation only leads to first MCSXF08 coil. Other coils can be retrieved two folders up other magnets can be selected.

[3]Factory Acceptance Test

[4]Citation only leads to MCSXF08 FAT. FATs of other magnets can be retrieved four folders up.

be easily read at the point where the nonlinear current decay due to the quench start. In the measurements it can also be seen that as soon as the current reaches almost zero, the total voltage across the magnet suddenly drops to zero. This happens because the current falls below the holding current of the diode so that it does not conduct any more. Therefore, the total magnet voltage drops by the diode forward voltage to 0.0 V. Since this behaviour is not modelled and the magnet discharged the measurements are taken into account until that exact time.
Table 5.1 contains all quench measurements considered in this thesis including the parameters just introduced. Due to the adaptation of only a few parameters for each quench event, mainly because of the absence of more complex quench protection systems, and because no quench dictionary was automatically generated during the measurement, it was decided not to use a `ParsimEvent` step in the final simulation setup. However, such a step was used for parameter studies.
For each of these measurements, the following variables were measured:

- VMA [V]: voltage across the first half of the magnet
- VMB [V]: voltage across the second half of the magnet
- VMT [V]: total voltage across the magnet
- DCCT [A]: magnet current

Through these measured quantities, even more can be calculated:

- VMR [V]: resistive voltage (calculated by the absolute difference of VMA and VMB[1])
- R [Ω]: magnet resistance (calculated by VMR/DCCT)

Measurements and simulations are compared below using these six physical quantities.

No.	Magnet	t_{end} [s]	initial current [A]	dI/dt [A/s]
1	MCSXF08	1.32	110.375	0.50
2	MCSXF08	4.14	50.740	0.47
3	MCOXF05	3.40	73.600	0.50
4	MCOXF05	2.98	82.300	0.50
5	MCOXF05	2.23	100.700	0.43
6	MCOXF12	2.77	86.000	0.50
7	MCDXF11	1.65	101.600	-0.50
8	MCDXF11	1.46	111.100	0.00
9	MCTSXF06	2.72	99.100	0.00
10	MCTXF04	>10	41.000	0.50
11	MCTXF04	4.20	96.000	0.38

Table 5.1: Overview of performed quench measurements including quench event information of the HL-LHC HOC magnets.

[1]This calculation is only correct if the quench does not propagate from one magnetic half to the other. In section 5.3 this is examined more closely.

5.2 Quench parameters

5.2.1 Quench initialisation

To start a quench in the simulation, a temperature profile can be applied in LEDET at a certain position along the conductor. For this profile a Gaussian distribution is used, whose peak temperature $T_{p,pulse}$ [K] and width w_{pulse} [m] can be specified. The position along the conductor is also given in meters and is defined using s from the geometry input file created in Chapter 3. $T_{p,pulse}$ and w_{pulse} should be chosen high enough so that there is a propagating quench but also not to high so that an implausible amount of energy is introduced from the simulation start on that leads to a different simulation outcome. So the smallest values are searched, which still trigger a quench. This way a realistic pertubation can be modelled. Depending on the magnet and event parameters, these are to be set differently and must be determined with several simulations. The current has the greatest influence: the lower this is, the more intense the pulse must be to cause a propagating quench[1]. For the simulation of MCSXF08 at 110.375 A, $T_{p,pulse} = 20\,\mathrm{K}$ and $w_{pulse} = 0.03\,\mathrm{m}$ is sufficient to start a quench, whereas for the simulation at 50.7 A $T_{p,pulse} = 40\,\mathrm{K}$ and $w_{pulse} = 0.15\,\mathrm{m}$ had to be applied.
Figure 5.1 shows the time evolution of the temperature profile near the quench start location using an MCSXF08 simulation at 110.375 A as an example. At the beginning of the simulation in dark blue, one can see the Gaussian profile, which then decays over time. During this time, however, the conductor locally is quenching, causing resistance to appear in this conductor section. Due to the resulting ohmic heat, the conductor continues to heat up and the quench spreads longitudinally and transversely due to thermal diffusion. If the profile is too weak to cause a quench, the Gaussian profile simply decays over time and the conductor section returns to the operating temperature of 1.9 K.

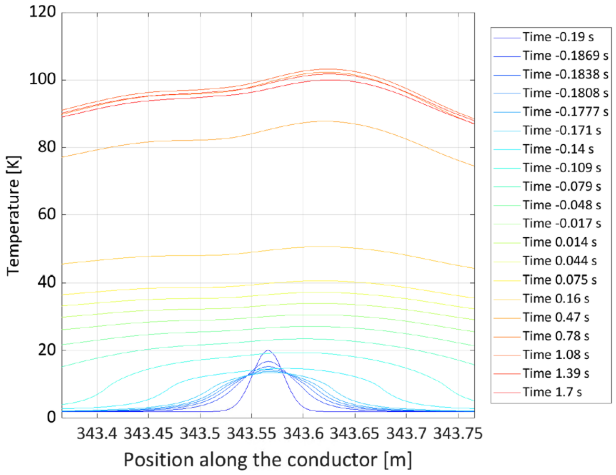


Figure 5.1: Time evolution of the temperature profile (temperature versus position) at the quench start location using the example of an MCSXF08 simulation with $T_{p,pulse} = 20\,\mathrm{K}$ and $w_{pulse} = 0.03\,\mathrm{m}$ in the case of a propagating quench.

[1]This is partly because lower current leads to a lower field (which is further away from B_c) and also because you are further away from the critical current J_c. So to lift the superconductor above its critical area it needs a higher temperature.

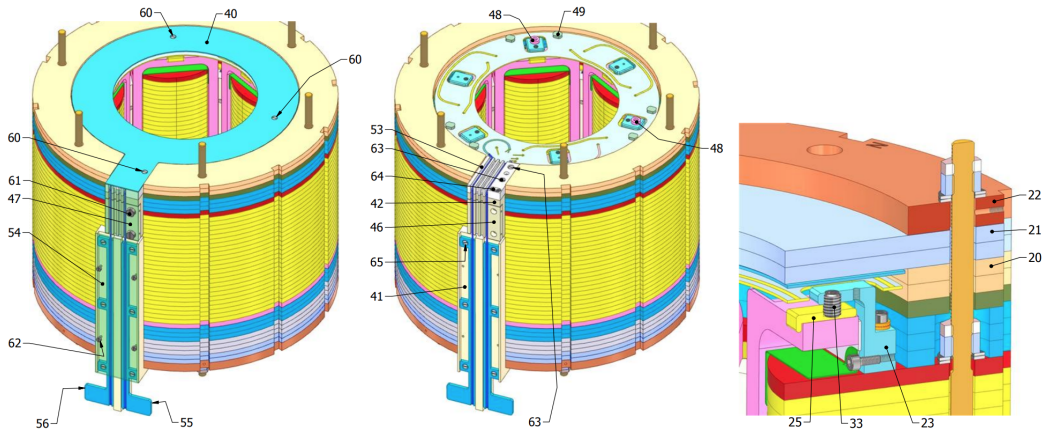
The start time of the quench is another unknown quantity that must be used for the simulation. This time is determined by first running a simulation with an estimated start time. For this estimation the first detectable drop of the current in the measured data was used. The resistive voltage of the simulation obtained in this way can be compared with the measurement and aligned. Since this rises relatively steeply, the time difference of the two curves can be determined exactly. The simulations are aligned to the time at which the resistive voltage reaches a value of 1.0 V. A second simulation with the corrected starting time, shows a good agreement of the quench starting time. Thus, for each measurement, a shorter simulation is first performed, which is used to adjust the starting time. Later, in figure 5.8 this alignment is shown at the example of a MCTXF04 measurement.

5.2.2 Quench location

The quench location cannot be measured or calculated and therefore cannot be adapted to the measured quench event for each simulation. As will be shown in the subsequent chapter, the quench location has a great influence on the simulation results. A quench in a low field region propagates much slower than a quench in a high field region. In addition, there are turns that are cooled more than others, which in turn slows down propagation. As explained in the previous section, each coil has different parameters due to the adjustments made by the `ParsimConductor` step. Parameters like RRR or critical current, which can be up to 30% and 5% different respectively, also have significant influence on the simulation result.
In this chapter, one location is selected to determine different model assumptions: the location with the highest magnetic field. In an adiabatic model, this is expected to be the location with the highest probability for a quench since the conductor needs smaller disposed energy to quench due to the higher field. The discharge at this location is expected to be relatively fast because the quench spreads faster due to the high magnetic field in the neighbouring half-turns making it easier for a quench to propagate (less temperature needed). Due to this faster propagation the field energy is dissipated in a bigger resistive region leading to a relatively low hot-spot temperature. In the non-adiabatic model, the high field location is not expected to be the location with the highest quench probability. This location is at the inner edge of the coil, which is closer to the aperture (see figures 3.15b and 1.8). This location is in contact with the He bath via the coil insulation and is therefore cooled more than other half-turns. So the most probable location would be further inside the coil, where the field is slightly smaller, but the cooling is lower. Knowing that this is not the most likely location, it is used for this chapter to validate assumptions that have a large impact.
To automatically calculate the location with the highest magnetic field the geometry input CSV is used. The locations with the highest magnetic field in each coil are indexed and the electrical center of the magnet is selected. A location close to the electrical center of the magnet is important because, among other things, the propagation of the quench into other coils is investigated and therefore electrically adjacent coils in both directions are needed.

5.3 Cooling of the conductor between coils

In the case of the HL-LHC HOC magnets cooling is very difficult to model for several reasons (discussed in detail in section 5.6). Since due to various limitations of the software it cannot be reliably simulated whether a quench propagates from one coil to its electrical neighbour. Therefore, an assumption has to be made how the cooling of the coil connection wires is modelled.
Depending on this cooling, a quench typically propagates to different numbers of neighbouring coils. Figure 5.3 shows the 3D temperature distribution at different times of two simulations with the extreme cases of strong cooling (left) and no cooling (right). With strong cooling ($f_{cooling} = 1$) there is no propagation of the quench. Thus, only the coil in which a quench starts heats up. If the cooling is reduced, a quench only propagates to a neighboring coil along the coil connection with the higher magnetic field. If the cooling is further reduced, the quench propagates in both directions and if the cooling is further reduced, the quench propagates over more and more neighboring coils (as seen in the figure). Efforts have been made to conduct simulations with various numbers of quenched coils. While in most cases no extreme differences were observed, a significant variation in outcomes was nonetheless noted.
Essentially, the coil connection is made with the same superconducting wire that is used for the coil winding. However, this cable is routed to the PCB without insulation and is thus entirely in the He bath [117, 114]. To validate how good the He feed is in this area, construction drawings are considered. In Figure 5.2a two 3D CAD designs of a complete MCSXF magnet are shown. Marked with the number 40 in the picture is an insulation cover that is close to the PCB, which is shown on the right side in white with yellow tracks. In the magnification in figure 5.2b it can be seen that the insulation cover, which is intended to prevent short circuits, and also the other components have a visible distance from the PCB not blocking any He to flow around it. Due to the very good wetting properties of helium, it is assumed that the cable lies in the He bath without additional insulation.



(a) Entire magnet (b) Magnification

Figure 5.2: 3D CAD renderings from the construction drawings of the magnet type MCSXF showing. Figures reproduced from [150, 151].

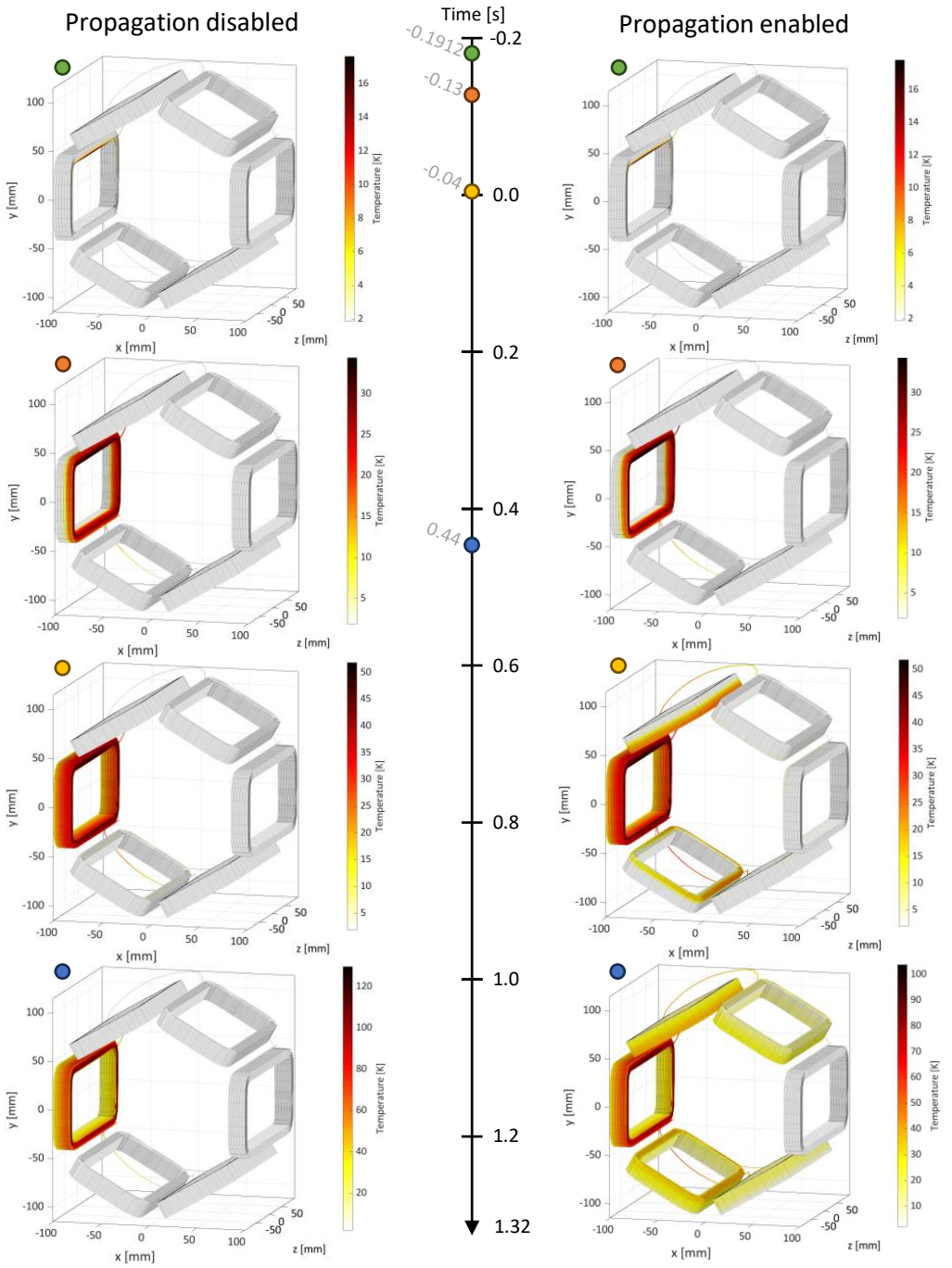


Figure 5.3: 3D temperature distribution at four selected time points in two MCSXF08 simulations with high cooling of the conductor between coils enabled or disabled. Initial current is 110.375 A.

This cooling in the busbars between coils cannot be modeled ideally in LEDET due to limitations of the software. As explained in chapter 3, the relative insulation thicknesses can only be specified for each half turn. The indices of the coil connection wires were chosen when generating the geometry input files in such a way that the first half of

the coil connection belongs to the previous half turn (the last half turn of the previous coil) and the second half to the succeeding half turn (first half turn of the succeeding coil). Therefore, only one insulation thickness can be defined for the last half turn and the first half coil connection or for the second half coil connection and the first half turn. If the adopted insulation thickness is correct for the half turns (according to chapter 3), a quench propagates only along the coil connection with higher magnetic field for a single coil. If no insulation thickness is applied, as would be correct according to the coil connection, the quench does not propagate to other coils. However, this strong cooling within a coil affects the discharge of the coil at the last and first half turn.
In the final simulation setup, a combination of the two extreme cases is therefore chosen: the first and last half turn of a quenching coil get the correct insulation thickness (together with the connected coil connections). The two other coil connections leading from other coils to this coil (and their connected half turns in the neighbouring coils) get an insulation thickness of 0. The first and last half turns of the remaining coils keep their correct insulation thickness, because there is no significant propagation of the quench and the coils should be as realistic as possible for coupling currents. With this setup, the quenching coil is minimally influenced and there is no propagation of the quench to neighbouring coils. When setting up the simulation files, the isolation is automatically adapted to the quench location by going through the half turn indices in electrical order and determining the coil connection indices for each magnet. For the simulation in figure 5.3 on the left exactly this setup has been used.
With this setup, which is as close to reality as possible without modification to the LEDET program, no propagating quench between coils occurred with any magnet type. This is also the more conservative approach, since peak temperatures are significantly higher with a discharge in only one coil than when propagation occurs. This can also be seen in Figure 5.3. At the fourth time step at 0.44 s, the maximum temperature without propagation is about 127 K, while it reaches about 104 K with propagation.

5.4 Magnet circuit and power supply

Figure 5.4 presents the electric circuit design used for each HL-LHC HOC magnet measurements. The magnet under consideration is interconnected at the "HOC magnet" designation. The Power Coneverter (PC) links to this magnet via an IGBT[1] polarity switch, a design choice allowing for easy change of the current flow direction. The warm resistance of the current leads, represented as R_c, measures 6.8 mΩ. Incorporating two central SCR[2] thyristor in this setup ensures an alternative route for the current when a maximum allowed voltage is crossed—thus bypassing the magnet. With each potential current direction in mind, a distinct diode is strategically placed. The forward voltage U_{fwd} of these identical diodes is 850 mV. To ensure consistent measurements, the current's intensity coursing through the magnet (known as DCCT) is precisely captured directly at the magnet.

In LEDET, it is not possible to model a more complex electrical circuit in which the magnet is connected. However, some parameters can be set for switching off the power converter: the switch-off time $t_{PC,off}$[s], the maximum applied voltage when controlling

[1] Insulated-Gate Bipolar Transistor
[2] Silicon Controlled Rectifier

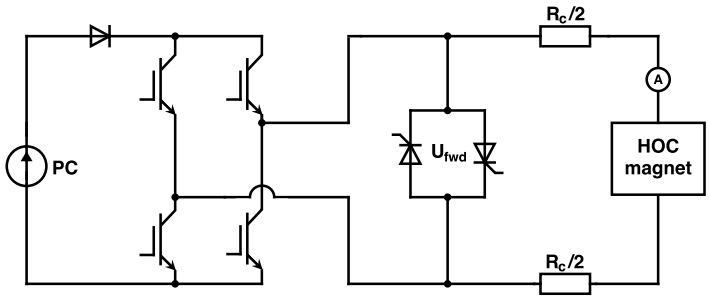


Figure 5.4: Electrical circuit diagram illustrating the test configuration used for HL-LHC HOC magnet measurements, detailing the interconnected components including the power converter (PC), the IGBT polarity switch, SCR diodes in a by-pass-crowbar, cable resistance R_c and the magnet (HOC magnet) with ammeter (ltr.).

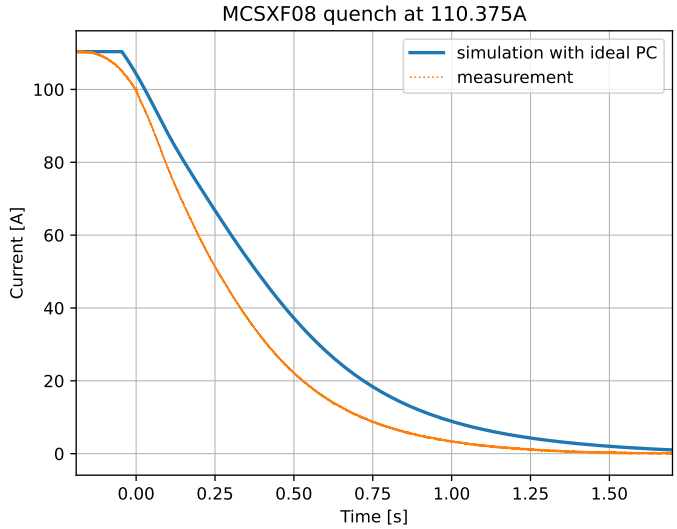


Figure 5.5: Comparison of measurement No. 1 with corresponding simulation result with initial (ideal) and revised modelling of the power converter (PC).

the current $U_{PC,max}$[V] (PC is modelled as a current source), the warm resistance in the crowbar $R_{crowbar}$[Ω] and the circuit $R_{circuit}$[Ω], the forward voltage of the diodes U_d[V] in the crowbar, and a look-up table in which the current to be held at each point in time is to be specified (linear interpolation in between). $U_{PC,max}$ is set to 10 V, a standard value for power converters in the LHC that has also been validated with measurements and simulations [152]. $t_{PC,off}$ was set to a very large number, so that the power converter does only trip when $U_{PC,max}$ is reached. The diode forward voltage U_d is given by 850 mV and the resistance in the wire R_c is given by 6.8 mΩ. Since this resistance is $R_{crowbar}$ and $R_{circuit}$ combined and there is no need to distinguish between them $R_{crowbar}$ is set to 6.8 mΩ and $R_{circuit}$ is set to 0.0Ω). The look-up-table for the given current at each time point is set so that the initial current is at the beginning of the simulation, then increases/decreases linearly with dI/dt until simulation end time.

With these settings, several quench events were simulated for MCSXF08. Figure 5.5 compares a representative simulation with corresponding measurement based on the current discharge. It can be seen that the current in the measurement decreases earlier then in the simulation. Also, the current falls slower initially in the measurement whereas in the simulation it drops suddenly after the power converter tripps since $U_{PC,max}$ =10 V is reached. The reason for this is that the power converter in LEDET is modelled as ideal. In reality, however, the power converter does not regulate fast enough to keep the current at the desired level straight after a quench start.
To simulate the power supply not regulating until it trips, there is unfortunately no ideal solution in the LEDET version used (2.02.26). The closest to reality is to set $U_{PC,max}$ to a very small number, like 0.001 V. This way, the power supply does not regulate immediately after the simulation start (which is also the start time of the quench) until it trips at 0.0 s. However, no resistance is introduced into this circuit at simulation starting time, as the power converter compensates for the voltage drop caused by the resistance of the circuit. So this resistance is only applied when the power converter trips at 0.0 s. $R_{crowbar}$ and $R_{circuit}$ are therefore set to 0.0Ω at simulation start. To introduce the warm resistance at time 0.0 s, an energy extraction system is included in the simulation, which introduces a resistance of 6.8 mΩ at this time. In the look-up table, the current is set to the desired value until 0.0 s is reached and then switched off. The introduction of the forward voltage at the diode into this model is problematic. Either this is ignored or it is introduced with the tripping of the power supply (i.e. with simulation start and not at 0.0 s). Applying this voltage too early or not applying it at all does not correspond to the real case. In section 5.7 one of the options is selected with the help of a quench integral comparison.
Figure 5.5 again shows the simulation results with this modelling of the power converter. It can be seen that the behaviour of the power converter is simulated much more realistically.

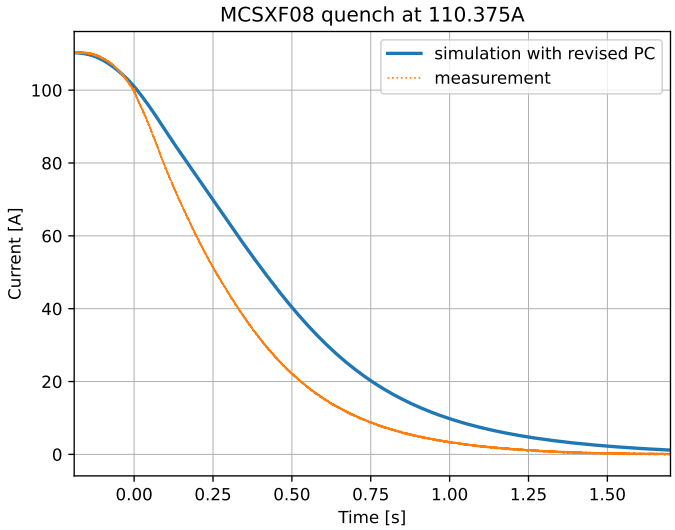


Figure 5.6: Comparison of measurement No. 1 with corresponding simulation result with revised modelling of the power converter (PC).

5.5 Epoxy impregnation

As can be seen in figures 5.5 and 5.6, the simulated current discharge is clearly too slow. The same behaviour can also be observed with other locations, other conductor parameters, with and without U_d and also other magnet types and current levels. The model is therefore still inaccurate. Since almost all parameters of the electrical and coupling current sub-models are known exactly (e.g. conductor parameters, winding length, coil resistances), the error must lie in the thermal model. One error could be the overestimated modelled cooling. However, it could be shown in several simulations that an adapted cooling alone is not sufficient to accelerate the discharge sufficiently, at least not with plausible values. Therefore, the adaptation of the cooling will be discussed in the following two sections.
Another uncertainty is the thermal influence of the epoxy in the coils. The superconducting cable is wound in epoxy and in the current model this epoxy is simply distributed homogeneously over all cable gaps. Since LEDET is primarily designed for squared conductors, a cable is thermally modelled in LEDET in such a way that a round conductor is filled with epoxy until it is square and the insulation layer is then applied around it (looking at a cable cross section). The parameters are the diameter of the conductor, the height and width of the epoxy square and the thickness of the insulation of the individual conductors. Conductor diameter and insulation thickness are known as 0.5 mm and 0.07 mm [147]. The dimensions of the epoxy square were calculated assuming homogeneously distributed epoxy. Thus, the cross-sectional area of all conductors including their insulation was subtracted from the cross-section (without outer insulation) and this remaining area was distributed evenly over all epoxy squares.
Three incorrect assumptions are made in this modelling. The first error is that the epoxy is only applied inside the insulation and not outside. The second error is that the size of the contact area is further overestimated because the conductor cross-section is modelled as square and not round. The third error is due to the fact that the parameters of the epoxy are not precisely known. The only insulation materials implemented in the used LEDET version are the widely used kapton and G10. For this model, G10, a fiber glass, was used for the epoxy, whose material properties are not identical.
Furthermore, the assumption of the uniformly distributed epoxy is not very good. Figure 5.7 shows the cross-section of a scrapped MCTSXF coil. It can be seen that the majority of the conductors are very close together and in many cases directly touching. The epoxy is found much more in larger pockets, which do not have such a large thermal impact, as many thermal resistances between conductors become small and a few become very large. Looking at the top of the coil in the picture, it can also be seen that there is a larger epoxy build-up at the edge of the coil insulation. In the orientation of the picture, the lower conductors are the first winding level and the coil is then wound from bottom to top. The epoxy at the top edge was probably filled in after the coil was wound to insulate it. Furthermore, it can be observed that the first winding levels are still relatively orderly, while subsequent ones appear increasingly disorderly. The exact modelling of such a cross-section is extremely challenging, since it must be expected that the cross-section of each built coil looks different along the conductor direction and here only a cross-section of a coil of a single magnet type can be considered, which is probably not representative. As a consequence, the thermal model in particular is not expected to be very accurate. The error in the geometrical model and the resulting errors in the coupling current model

due to the calculation of the mutual inductance is difficult to quantify. However, since no better data are available, a uniform distribution of the conductors is assumed.



Figure 5.7: Photography of a scrapped MCTSXF cross-section showing single isolated cables in silver, epoxy in dark brown and coil insulation in light brown.

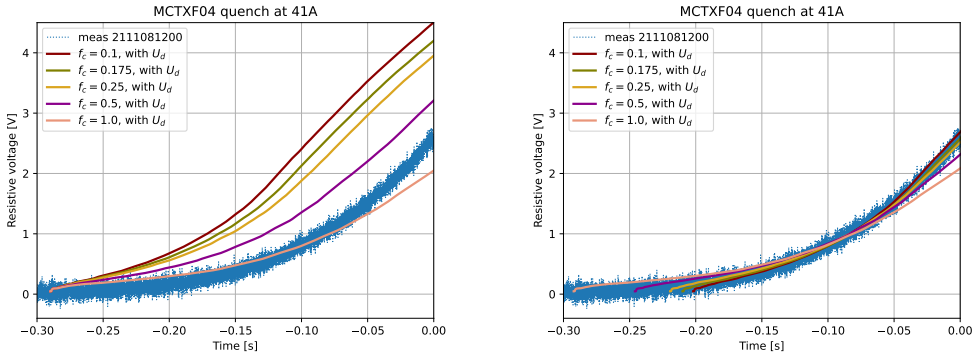
Modelling the epoxy would therefore be inaccurate and the distribution of the epoxy shows that neglecting its thermal capacity is a far better assumption than including it. Therefore, the epoxy is not included in the model. The height and width of the epoxy square is set so that it has the same area as the round conductor, without any extra epoxy.

5.6 Cooling correction factor f_c

In the current model cooling is overestimated due to different reasons. Firstly, the epoxy is neglected, which means that in the model, after the insulation of the cable, there is directly the insulation of the coil. However, as shown in the previous section, a denser epoxy layer is present, especially at the bordering region. In addition, LEDET also assumes that the cooling to the He bath is a squared conductor. The size of the surface is therefore overestimated and the proximity of this surface to the bath is underestimated. Both effects lead to increased cooling. In addition, the He bath in LEDET is modelled as an infinite heat sink. Specifically, the outer surface of the coil insulation is set to a constant temperature of 1.9 K as a boundary condition. Effects such as convective cooling, Helium heating up, possible local evaporation or the fixation of the coil itself (not perfectly surrounded by He bath) are therefore not modelled.
For these reasons, a cooling correction factor $f_c[]$ is introduced. Since there is no easy way to scale the cooling of the coils by a constant factor in LEDET, the relative insulation thickness $f_{cooling,calc}$ calculated in chapter 3 is scaled: $f_{cooling,new} = f_c \cdot f_{cooling,calc}$, where $f_{cooling,new}$, is used in LEDET.
Since the insulation thickness of the wire is fixed to 70μm, this factor scales the thickness of the insulation layer of the coil to the He bath, according to the definition in equation 3.6. Thus, a factor of $f_c = 0.5$ would mean that this insulation layer is approximately doubled[1]. However, this factor must model the surface area that is significantly too large due to the squared conductor modelling and the layer of epoxy that is neglected. Therefore, it is expected that this factor can also be significantly less than 1. At this point it should also be mentioned that the introduction of such a factor introduces further assumptions into the model. On the one hand, it is assumed that this factor is constant for

[1] Since $th_{cable} << th_{coil}$ equation 3.6 can be approximated for this estimate: $f_{cooling} = \frac{th_{cable}}{th_{coil}}$

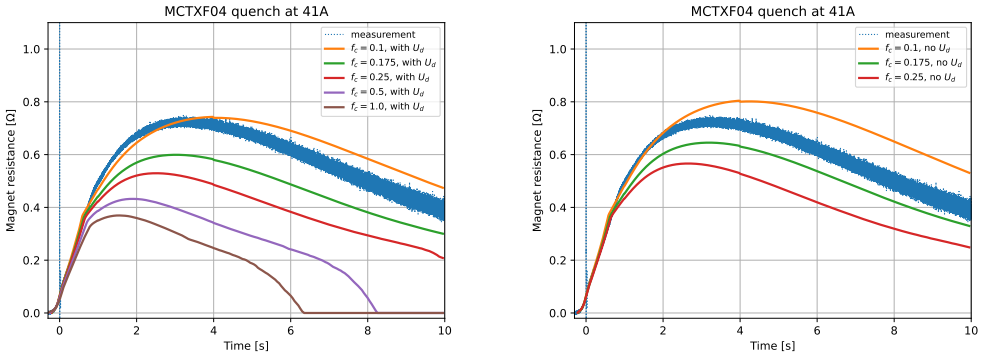
the entire discharge and on the other hand, it is assumed that this factor is the same for the different magnet types. However, since He cooling involves very complex non-linear physical processes and since the magnet types have different geometries, these two assumptions are not ideal. Due to the lack of alternatives with realistic computational effort and also because the clearly overestimated cooling has to be corrected, this approach is nevertheless chosen.
To determine f_c, measurement no. 10 (see table 5.1) is considered. As this measurement is clearly the longest, cooling is expected to have the greatest effect. Since f_c has a great influence on the simulation, each simulation must be carried out twice, as the starting time must be adjusted. Figure 5.8 shows the adjustment process of the simulations. Figure 5.8a shows the resistive voltage of the first simulation results, which were then used to adjust the starting times of the second simulations so that they have the same resistive voltage at 1 V (see Figure 5.8b).



(a) first simulations with estimated start time

(b) second simulation with adjusted start time

Figure 5.8: Simulation results with different f_c for corresponding measurement No. 10 showing the adjustment process of the starting time according to resistive voltage at 1 V.



(a) including U_d at simulation start

(b) not including U_d

Figure 5.9: Simulated resistance with different f_c values in comparison to corresponding measurement No. 10.

Since it has not yet been determined whether U_d will be introduced at the start time, simulations are carried out with and without including U_d. The resulting resistance curves are shown in figure 5.9. In figure 5.9a it can be seen that the factors $f_c = 1.0$ and $f_c = 0.5$ cool so much that the magnet returns to the superconducting state without discharging completely. However, since this cannot be seen in the measurement, the number must be significantly smaller. This evidence confirms that cooling is clearly overestimated. For both cases, with and without U_d, the measurement lies between the two simulations with $f_c = 0.175$ and $f_c = 0.1$. In the following section, f_c and U_d are specified more precisely using the quench load criterion.

5.7 Setting U_d and finalising f_c

In section 5.6, the value f_c was restricted when comparing simulated resistances: $0.1 < f_c < 0.175$. Furthermore, in section 5.4 an implementation problem of the power converter was encountered. It was left open whether U_d is introduced too early (at simulation start) into the model or whether it is neglected. In this section, simulations for all measurements are carried out with $f_c = 0.1$, $f_c = 0.14$ and $f_c = 0.175$ each with and without U_d to be included - i.e. six simulations each for eleven measurements. Based on a comparison of quench load [A^2 s], also called quench integral, the most suitable model is then chosen.
The quench load is an important quantity in the modelling of superconducting magnets. For a safe operation of the magnet, the hot-spot temperature T_{max}[K] must be limited. By derivation via a heat diffusion differential equation and assuming adiabatic conditions, a conservative estimate for the hot-spot temperature can be made [153, 154]:

$$\int_{T_{op}}^{T_{max}} \frac{\bar{C}}{\bar{\eta}} dT = \int_{t_{q,s}}^{\infty} J_{op}^2 dt \tag{5.1}$$

Where $\bar{C}[\frac{J}{kg \cdot K}]$ stands for the averaged heat capacity, $\bar{\eta}$ for the average resistivity, T_{op}[K] for the operating temperature, T_{max}[K] for the hot-spot temperature, T[K] for the temperature, t[s] for time, $t_{q,s}$[s] for the quench starting time, and $J_{op}[\frac{A}{m^2}]$ for the operating current density. Except for T_{max}, all parameters on the left side are known because they are properties of the materials in the cable. The right-hand side therefore determines how large T_{max} is and contains the reaction of the circuit. If one does not use a cross-section-independent formulation by multiplying by the cross-sectional area, the right-hand side is the quench load.
As a reference value for the error calculation, quench load is determined for each measurement. To determine the quench start time $t_{q,s}$, the noisy resistive voltage from the measurement is smoothed with a Gaussian filter. In order to achieve a sufficiently strong smoothing for the 20 kHz measurements, different standard deviations σ for the filter are visually checked. These are shown in figure 5.10 exemplarily for measurement No 1. It was found that $\sigma = 200$ is the most suitable for the measurements and together with the threshold of 0.1 V it leads to a satisfactory calculation of the area under the current curve.

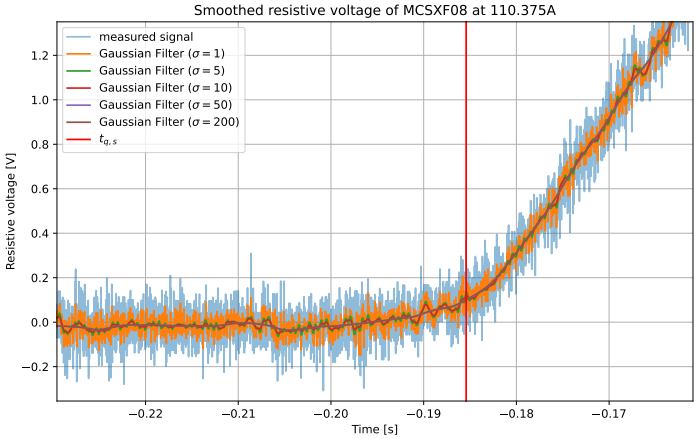


Figure 5.10: Measured resistive voltage smoothed with gaussian filters of different standard deviation σ to determine the quench start time $t_{q,s}$.

For numerical integration, linear interpolation was used between the data points and then the trapezoidal integration rule was applied. This procedure was also used to calculate the integral of simulation results. However, since the resistive voltage is not noisy, no smoothing is needed to determine the quench starting time.
The deviations of all simulated quench load from experimentally determined ones are shown in figure 5.11.

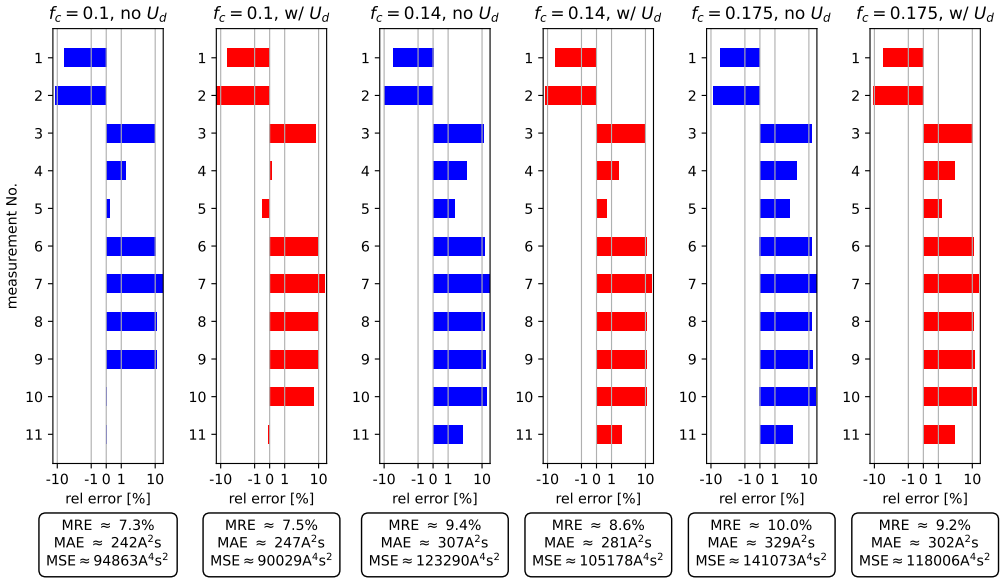


Figure 5.11: Comparison of relative quench load errors, mean relative error (MRE), mean absolute error (MAE) and mean squared error (MSE) of each measurement with high-field-location-simulations of different cooling correction factors f_c and excluded or imperfectly included U_d.

The figure shows that all errors are lowest for $f_c = 0.1$ with U_d at the start of the simulation. Furthermore, it is noticeable that for each f_c the variant with included U_d shows a lower error. This is probably due to the fact that U_d does not matter as much at the beginning when the current is high as it does at the end when the current is very low. It is also noticeable that quench load of MCSXF are underestimated, while quench load of the other measurements with the model are mostly overestimated. This might be due to the fact that the high field location of MCSXF could be in a coil that discharges particularly quickly. However, this effect is not expected to be as pronounced. Therefore, the epoxy area of the magnets was considered again. Per strand, the epoxy in the cross-section of MCSXF has an area of about $0.210\,\text{mm}^2$, while those of MCOXF, MCDXF, MCTXF, MXTSXF are $0.150\,\text{mm}^2$,$0.154\,\text{mm}^2$, $0.159\,\text{mm}^2$ and $0.159\,\text{mm}^2$, respectively. To put this into perspective, the wire cross section is $0.79\,\text{mm}^2$ for all magnet types. So the approximation to neglect the epoxy is a worse approximation for MCSXF than for the other magnets, because there is up to 40% more epoxy in MCSXF coils. This leads to a faster discharge in the simulation, as the cables should be further apart and heat should have a higher thermal resistance.
Since the aim of this thesis is to create a model with a set of model assumptions for the HL-LHC HOC series, $f_c = 0.1$ is used for the final simulations with U_d at the start of the simulation. However, magnets of type MCSXF are expected to discharge rather faster then in the experiments, while the remaining magnets discharge slightly too slow. Resulting voltages should therefore be higher for MCSXF in the simulation than in the measurement.
Figure 5.12 shows again the simulation result of measurement No. 1 with the now final simulation model. This model is still subject to the uncertainty of the quench location, which will be examined in more detail in the following chapter.

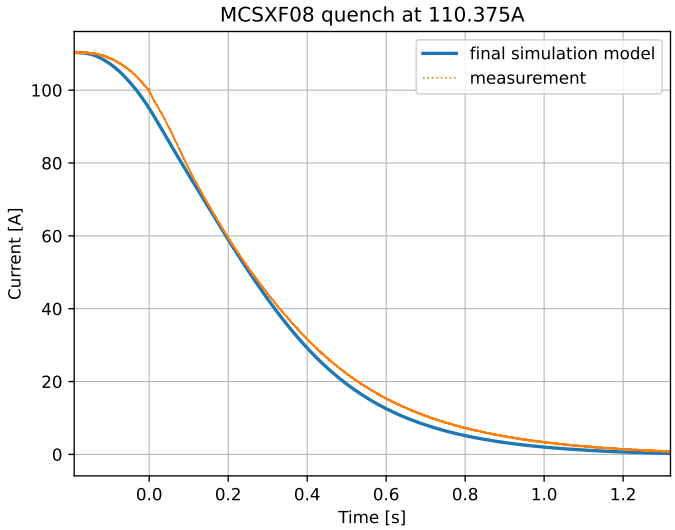


Figure 5.12: Comparison of measurement No. 1 with corresponding simulation result with the final simulation model.

In table 5.2 an overview of the most important parameters of the final simulation models is given. The number of time steps varies due to different starting times and simulation lengths for different measurements but also because LEDET internally has quality criterion that have to be met, otherwise the step size will be reduced automatically. The long simulation times make a simulation study, e.g. regarding the quench location, very time consuming.

Magnet type	time steps	half turns	nodes	simulation time
MCSXF	600-700	3456	88977	$\approx$ 40 min
MCOXF	700-800	5952	120849	$\approx$ 100 min
MCDXF	600-700	4560	87385	$\approx$ 100 min
MCTSXF	600-700	10656	142153	$\approx$ 180 min
MCTXF	800-1000	10656	544538	$\approx$ 240 min

Table 5.2: Key parameters of the finite difference models in LEDET. The simulation times refer to a machine with 32GB RAM with a 2.19 GHz processor.

Finally, table 5.3 shows which pulse parameters were needed in the final model for a quench start in all selected locations (see section 5.2).

No.	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**
$T_{p,pulse}$ [K]	20	30	20	20	20	20	20	20	20	40	20
w_{pulse} [m]	0.03	0.25	0.2	0.2	0.2	0.2	0.03	0.03	0.15	0.5	0.1

Table 5.3: Overview of needed pulse height $T_{p,pulse}$ and width w_{pulse} of the Gaussian temperature distribution used to start a quench in the simulation for each measurement at hand.

6 Simulated vs measured transients

6.1 Quench location analysis

The last simulation parameter yet not determined is the exact quench location. Since this value cannot be easily determined by voltage taps or other diagnostics, the three-dimensional magnet models with 31^2 uniformly distributed quench locations were simulated. Due to the symmetry of the HL-LHC HOC magnets, however, very similar locations were placed in different coils, which led to very similar simulation results. Only the variations in conductor parameters led to different results. Furthermore, 31 is not a particularly large number, considering that the magnets have up to 12 poles, i.e. less than three locations per pole. Also, in some magnets low field and high field regions are not included with this method.

To enhance the efficiency of the simulation while ensuring a thorough analysis of both best and worst-case scenarios in terms of hot-spot temperature, locations are selected more strategically. For each magnet, the resistances of all its coils at 10 K are calculated by dividing the measured room temperature resistance by the measured RRR. Then the two coils with the largest and smallest resistance at 10 K are selected, as these are expected to lead to the fastest and slowest discharges. In these two coils, 17 equidistantly distributed locations are then set and simulated. The number 17 is used again because, as a prime number, it has no common divisors or multiples with the number of halfturns in the height and width directions of the HL-LHC HOC coil blocks. Visually, it was checked again that the selected locations are distributed as widely as possible within a coil (especially that high field and low field regions are included). Since the starting times are not influenced as much by other locations as, for example, with a different f_c, and also since the simulation effort would increase significantly, the starting times determined by the simulations in the high field location were used. However, this could be improved.

In the future, it would even be conceivable to have an analysis that estimates the probability of a quench location. Since the gaussian pulse parameters needed to start a quench depend on the location, many locations could be carried out with incrementing these parameters until a propagating quench occurs. This makes it possible to estimate for which locations the smallest possible perturbations are necessary and for which larger ones. In a plot similar to the one in this chapter with min/max values, every simulation result could then be marked in colour (heat map) representing the needed pulse parameters. In theory, the locations that require smaller perturbations should also be the most likely locations.

[2]The number 31 was chosen because the simulation times are realistic (depending on the magnet and the hardware used (Windows PCs with 32-128GB RAM) simulations take 1-4 hours.) and 31 is a prime number and thus no totally identical locations are simulated in different coils for symmetry reasons.

A prerequisite for this, however, may be a proper inclusion of the epoxy and also a more realistic quench start since in the real world a gaussian distribution of power deposition is more likely than the gaussian temperature profile.
In figures 6.1, 6.2, 6.3, 6.4, 6.5, 6.6, 6.7, 6.8, 6.9, 6.10 and 6.11 the simulation results with 34 locations and the high field location are compared to the corresponding measurements. Minimum and maximum values and the median of the 34 locations are shown.

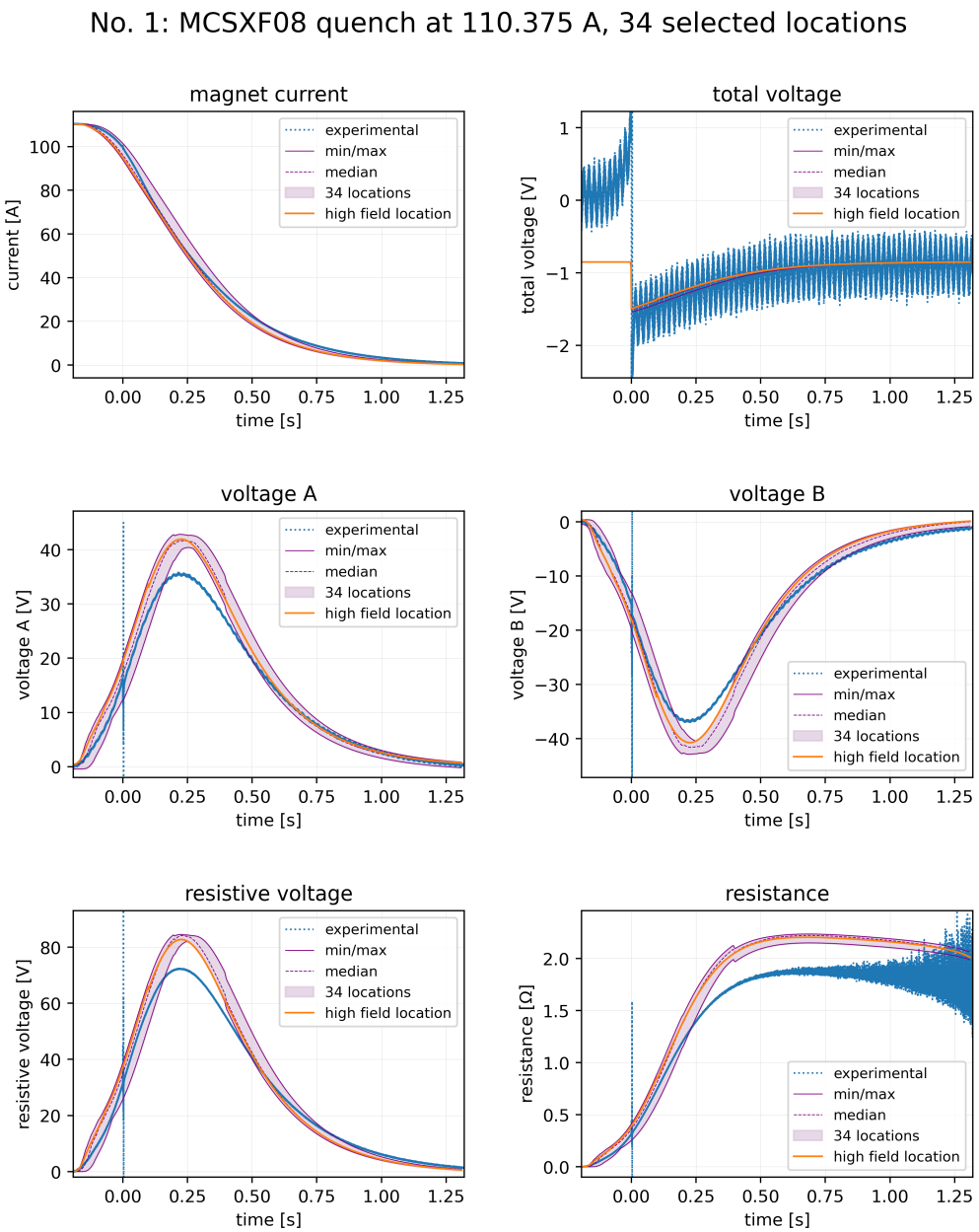


Figure 6.1: Comparison of simulation results corresponding to measurement No. 1 showing the maximum, minimum and median values of 34 selected quench locations and high field location.

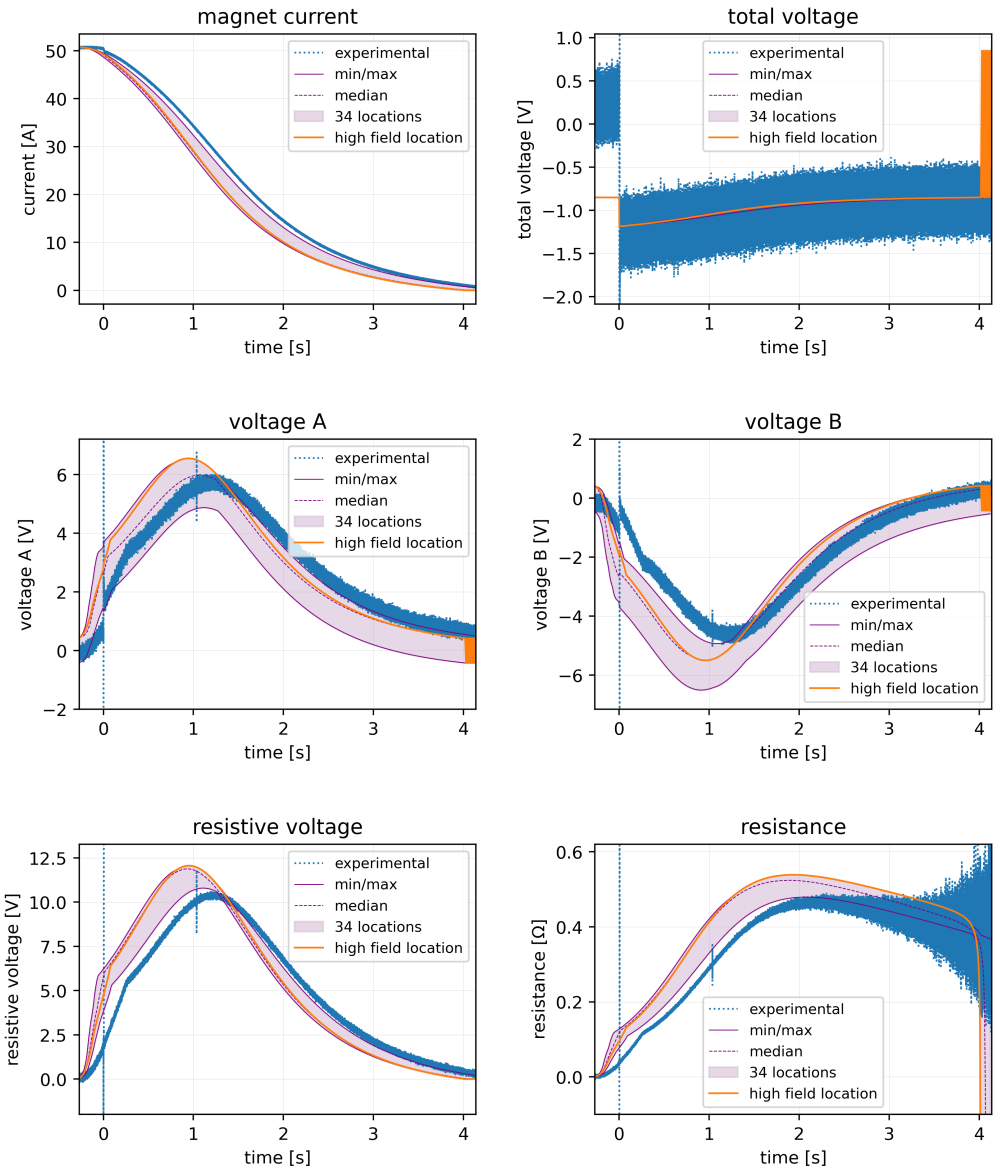


Figure 6.2: Comparison of simulation results corresponding to measurement No. 2 showing the maximum, minimum and median values of 34 selected quench locations and high field location.

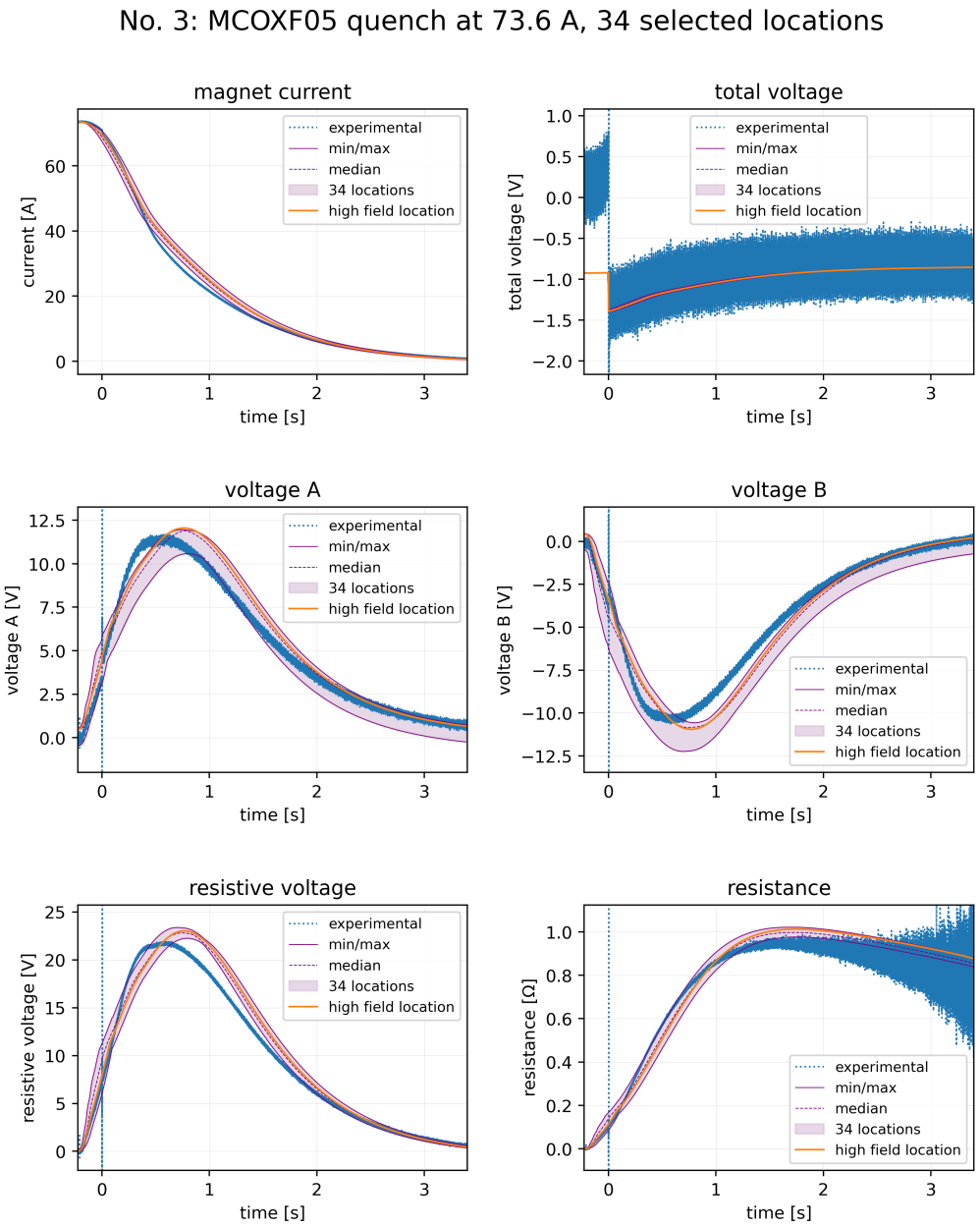


Figure 6.3: Comparison of simulation results corresponding to measurement No. 3 showing the maximum, minimum and median values of 34 selected quench locations and high field location.

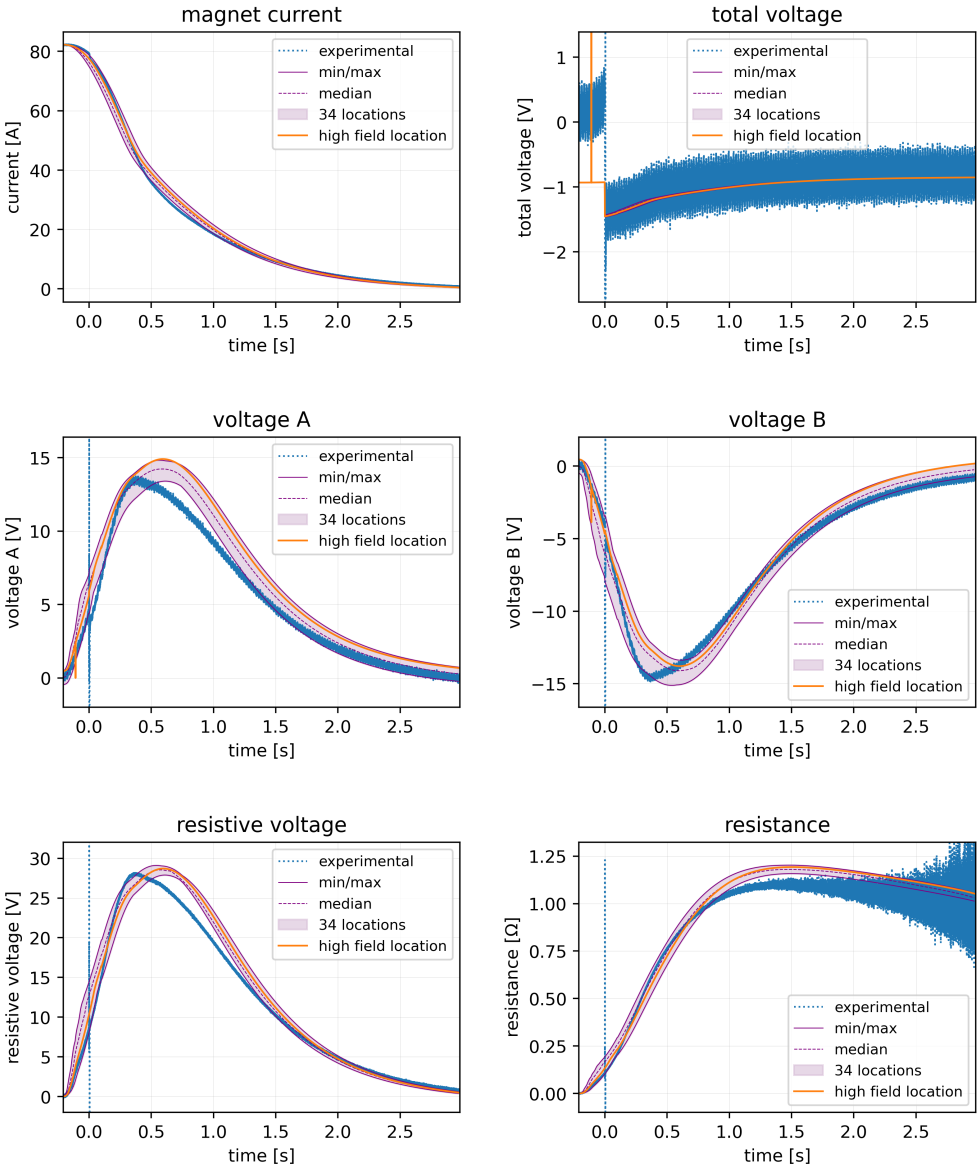


Figure 6.4: Comparison of simulation results corresponding to measurement No. 4 showing the maximum, minimum and median values of 34 selected quench locations and high field location.

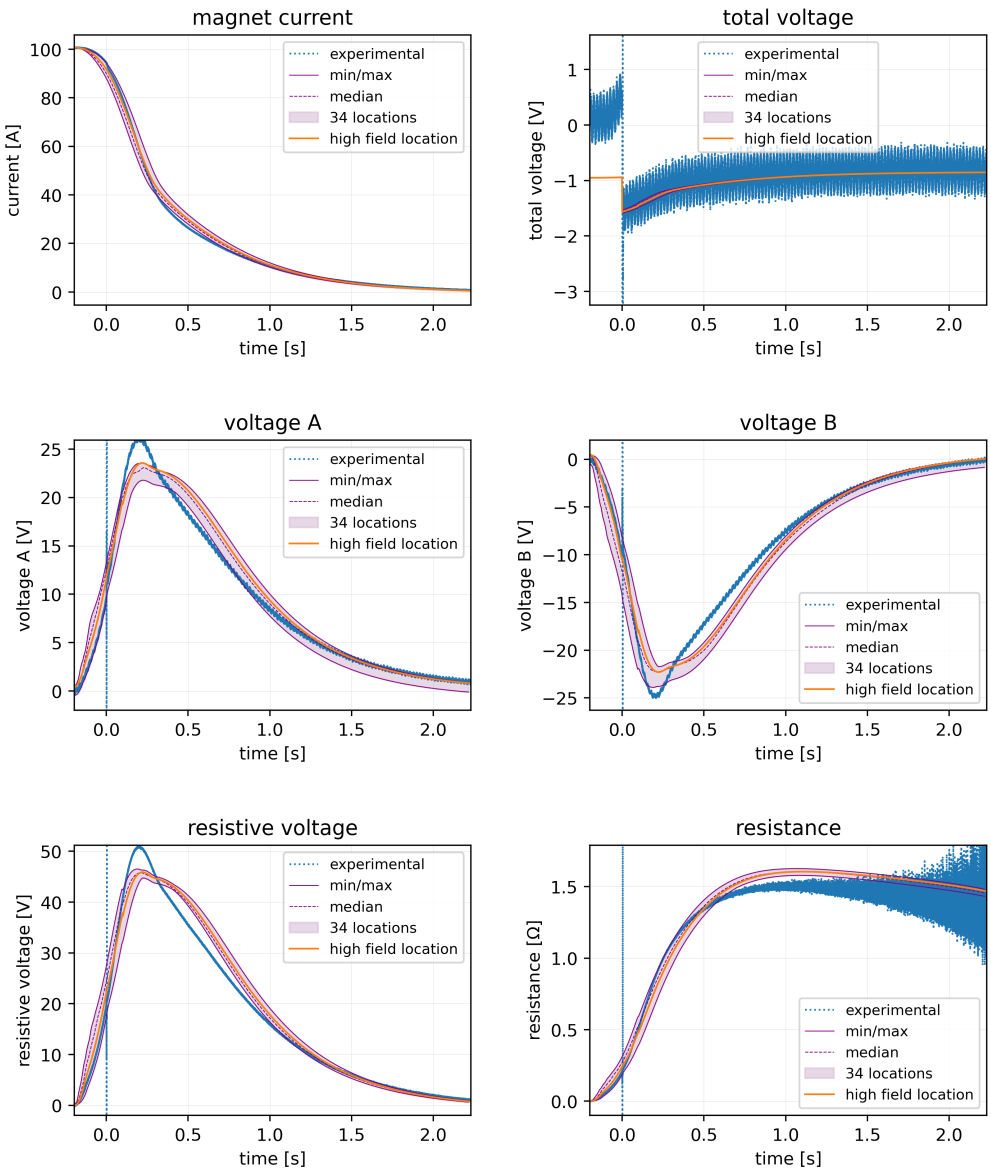


Figure 6.5: Comparison of simulation results corresponding to measurement No. 5 showing the maximum, minimum and median values of 34 selected quench locations and high field location.

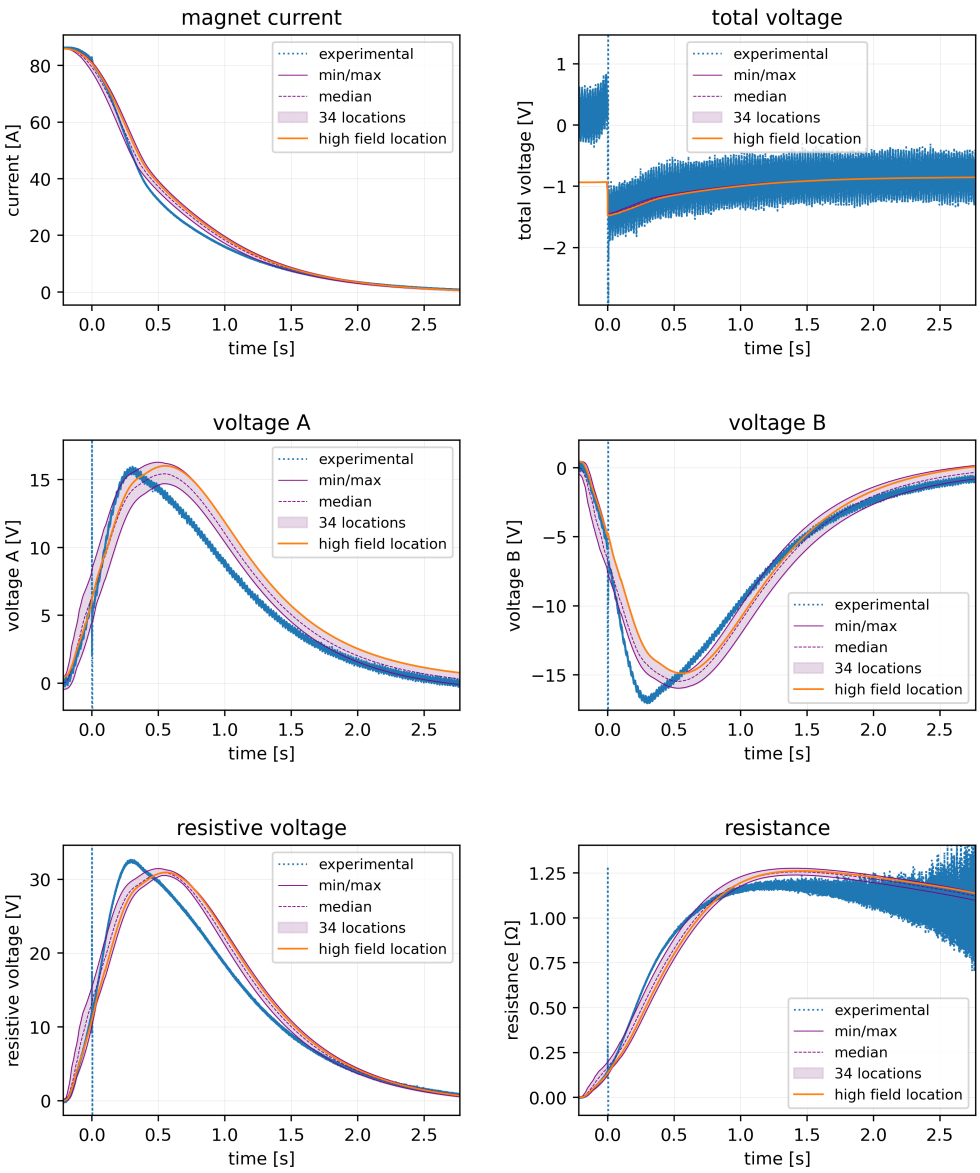


Figure 6.6: Comparison of simulation results corresponding to measurement No. 6 showing the maximum, minimum and median values of 34 selected quench locations and high field location.

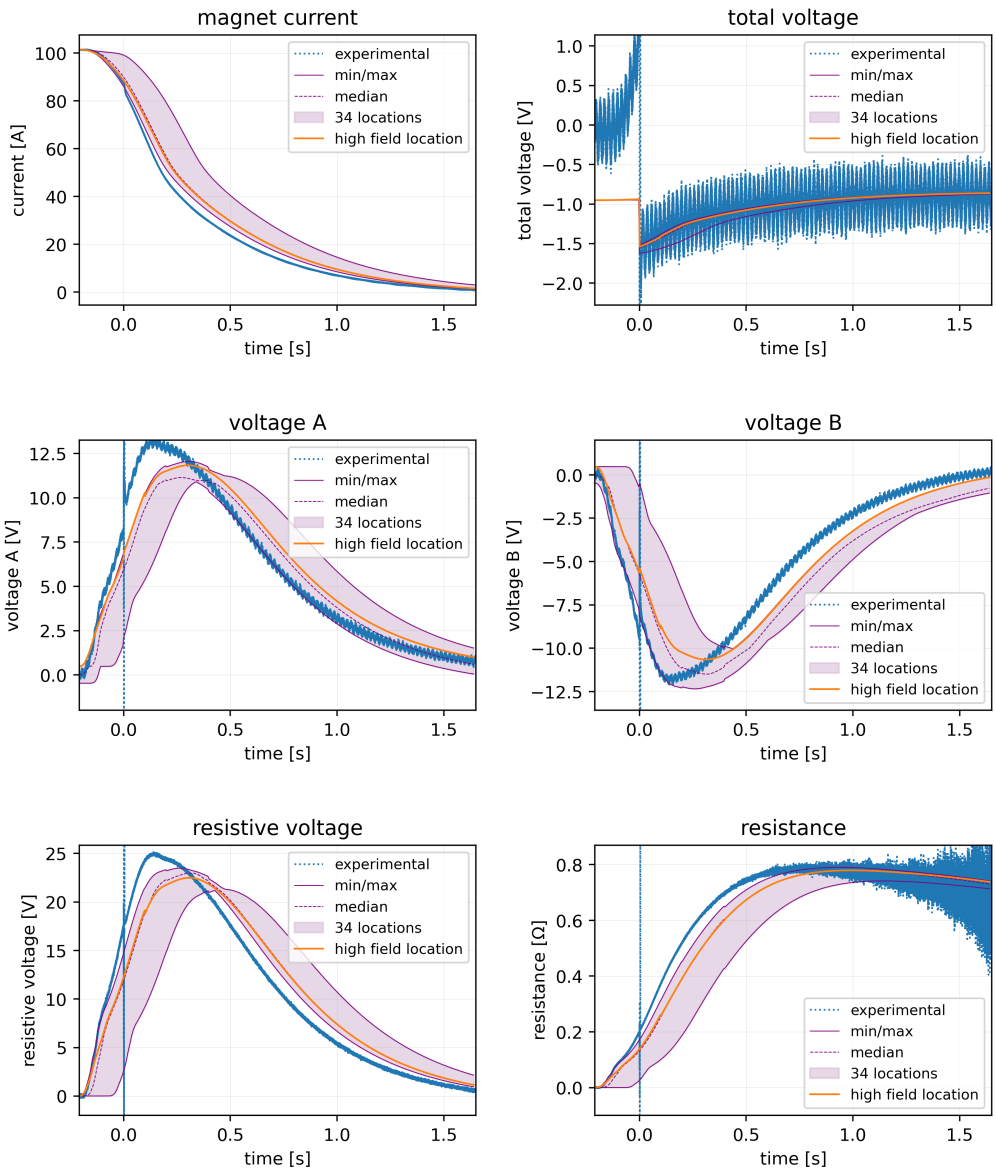


Figure 6.7: Comparison of simulation results corresponding to measurement No. 7 showing the maximum, minimum and median values of 34 selected quench locations and high field location.

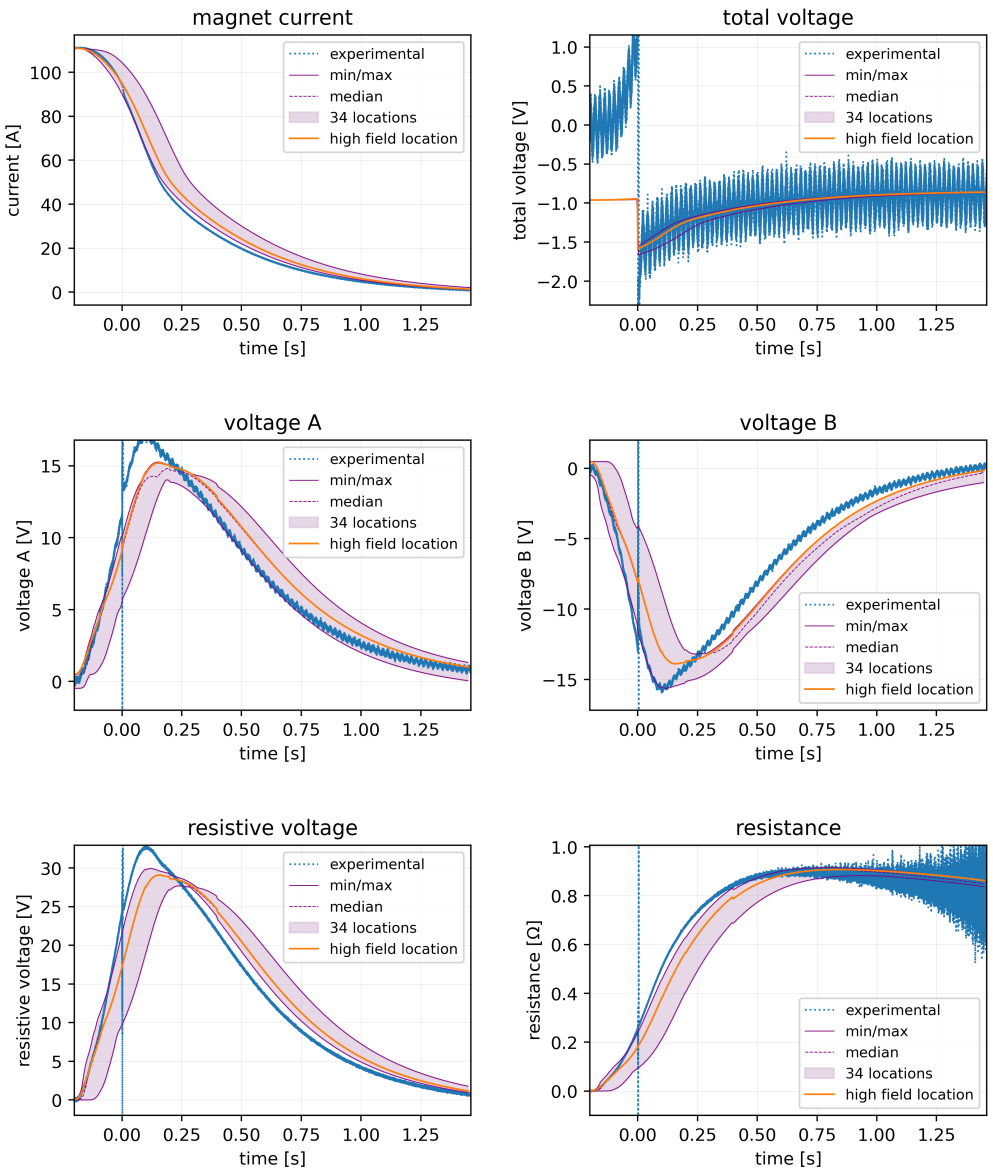


Figure 6.8: Comparison of simulation results corresponding to measurement No. 8 showing the maximum, minimum and median values of 34 selected quench locations and high field location.

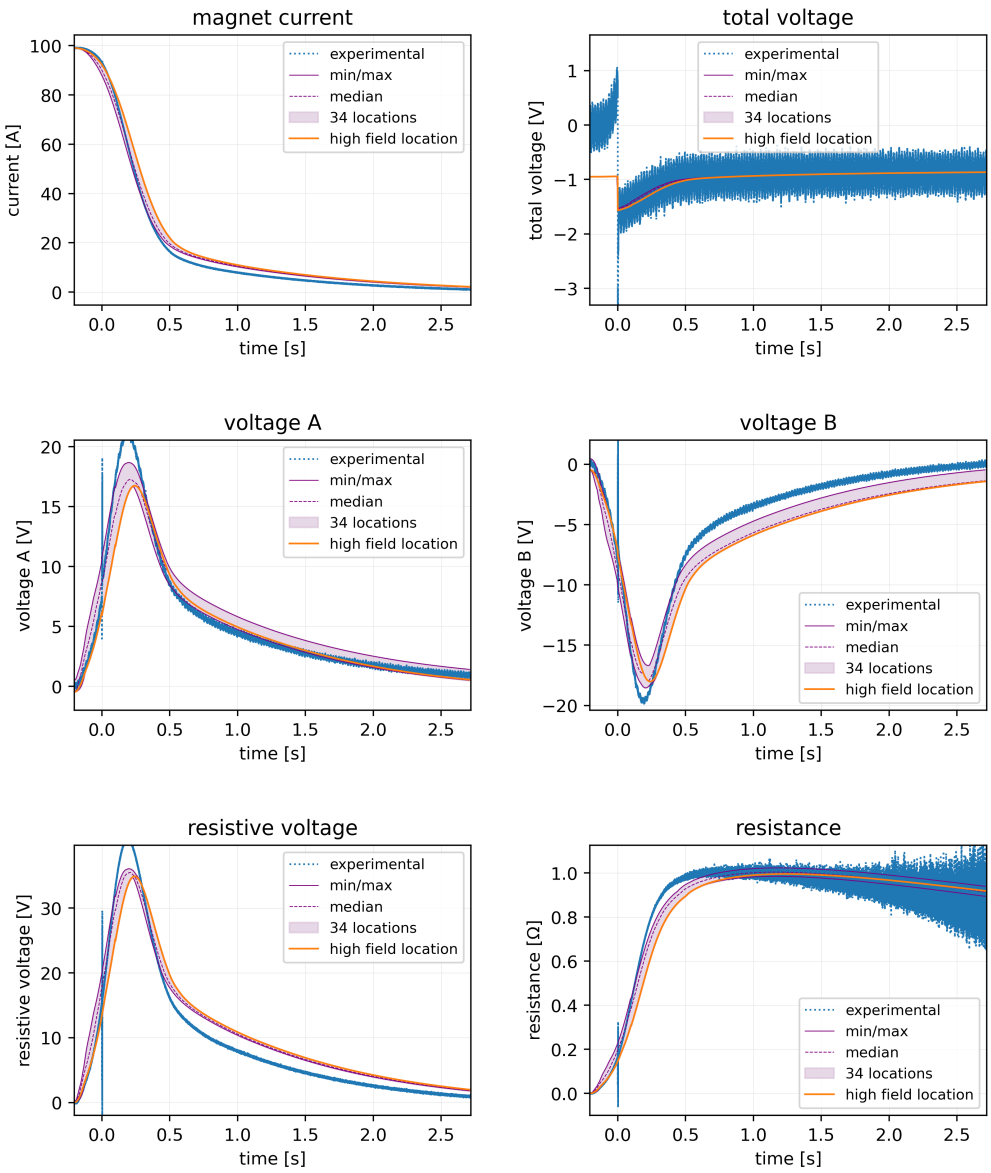


Figure 6.9: Comparison of simulation results corresponding to measurement No. 9 showing the maximum, minimum and median values of 34 selected quench locations and high field location.

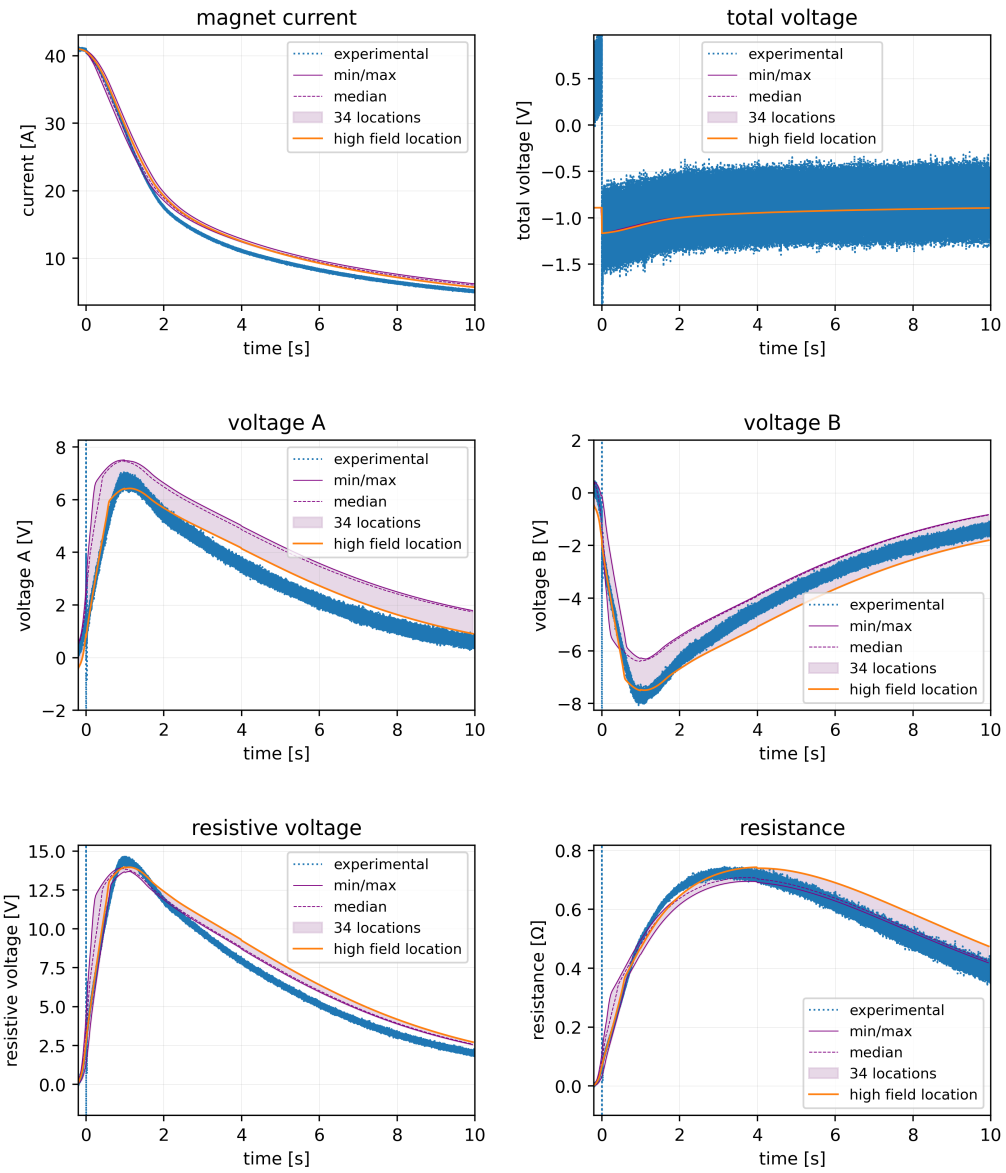


Figure 6.10: Comparison of simulation results corresponding to measurement No. 10 showing the maximum, minimum and median values of 34 selected quench locations and high field location.

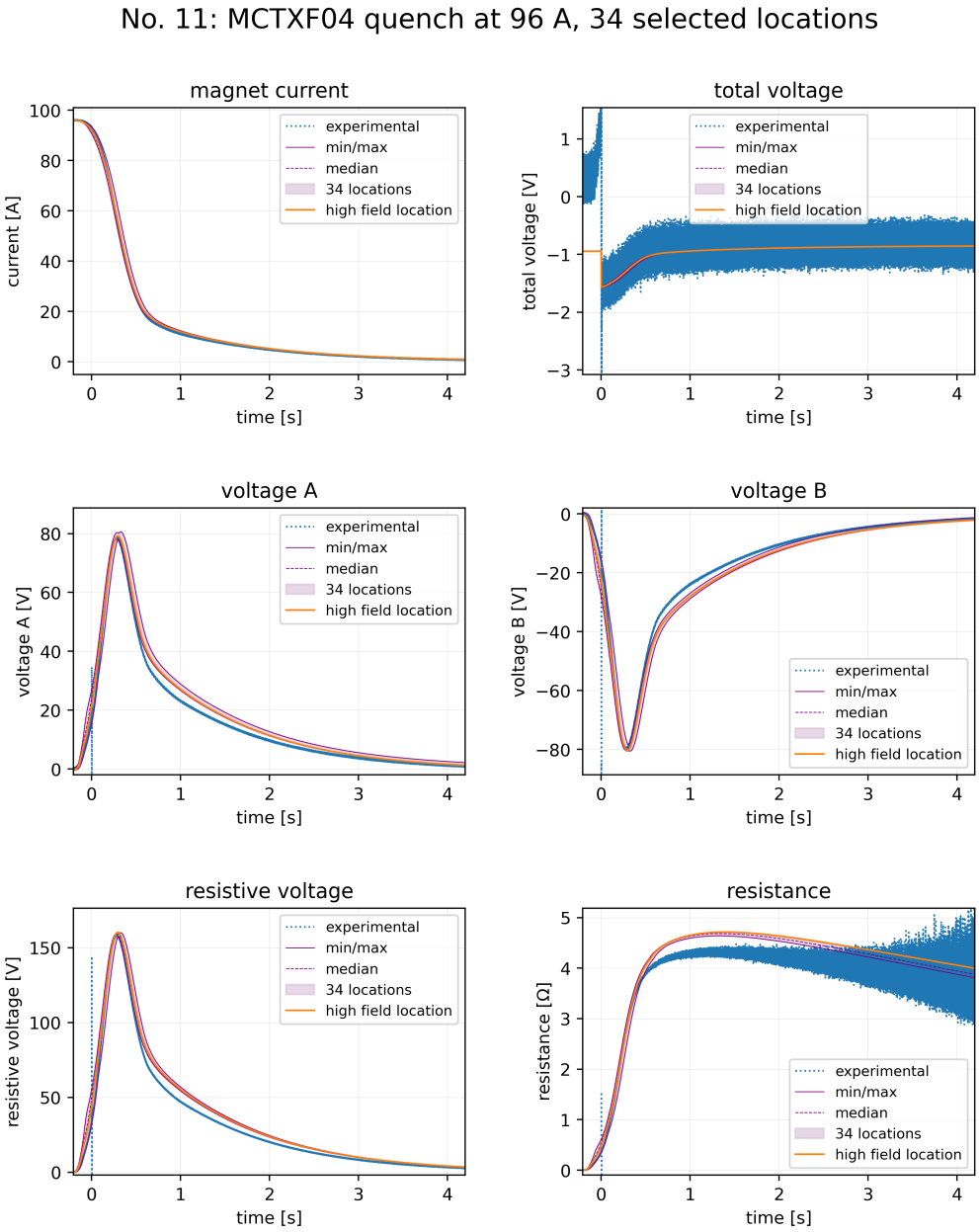


Figure 6.11: Comparison of simulation results corresponding to measurement No. 11 showing the maximum, minimum and median values of 34 selected quench locations and high field location.

In the figures it can be seen that the current is always relatively well matched. The total voltage is 0.85 V too low until $t = 0.0$ s where the power supply is switched off because the forward voltage of the diode is already introduced at simulation start. Also the total voltage does not increase until the power supply is switched off, because the power supply already tripped at simulation start since the maximum allowed voltage was set to a very small number to model the non-ideal behaviour. Also the remaining voltages A, B and the resistive voltage are relatively well matched. Furthermore, the resistance is also

well matched, especially in the initial range, which is the most critical for an estimation of the peak temperature.
Additionally, it can be seen that the width of the band between minimum and maximum of the simulations depends on the parameters of the Gaussian pulse for the quench start. These parameters were increased until a propagating quench occurs for the respective magnet at all locations. Now it can be seen that the smaller the pulse width and height (see measurement no. 7 and 8 in table 5.3 and figures 6.7 and 6.8), the larger the difference between minimum and maximum simulation results. This means that the model with lower gaussian parameters is more sensitive to smaller magnetic field and therefore takes longer to start a propagating quench. This leads to some inaccuracies with the simulations, since the simulations were not run twice to adjust the simulation start time in each case due to the large effort involved. Earlier investigations showed that this error is negligible, but this was only investigated for higher gausssian parameter. Since the strength of the perturbation, like the quench location, is an unknown quantity that is difficult to quantify, a separate parameter study would be needed to investigate these uncertainties. It is therefore expected that adjusting the simulation start time will considerably improve the agreement between measurements and simulations.

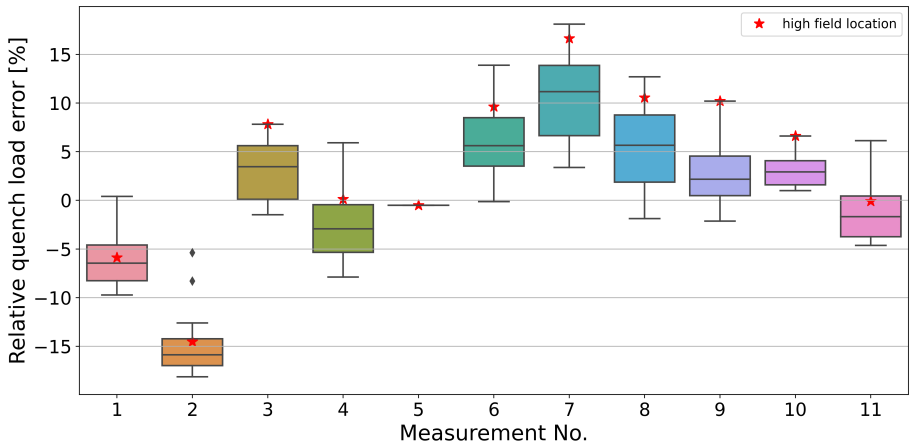


Figure 6.12: Boxplot of the relative quench load error for simulations with 34 selected quench locations corresponding to the measurements at hand.

For these 34 simulations the quench load error is considered. Figure 6.12 shows the distribution of the relative error in form of a box plot[1]. It can be seen that the simulated quench load for MCSXF (measurement No. 1 and 2) is, as expected, underestimated with respect to measurements, whereas the quench loads of the other magnets are in general overestimated. As mentioned before, the reason for this is probably that the assumption to neglect the heat capacity of the epoxy is worse for MCSXF than for the other magnets. Furthermore, it can be seen in the figure that almost all simulations for the five magnet types and at various current levels results are within the uncertainty of the quench loca-

[1]A box plot visually depicts data distribution using a rectangular box spanning from the first to third quartiles, with a line inside indicating the median, and "whiskers" extending from the box to represent data variability, with points beyond the whiskers signifying outliers. Outliers are defined as values that are more than 1.5 times the difference of first and third quartiles outside of the box.

tion. Only the results of measurement numbers 2, 7 and 10 fdeviate further. However, these are also the measurements that deviate the most from nominal conditions.
In summary, after evaluation of the model's capabilities, it is evident that the model validation aligns effectively with the observed data. Therefore, the model is deemed fully suitable for predicting transients during the operation of the specific magnets within the LHC circuits.

6.2 Quench back

In the simulations, the only coils where a resistance is developed are those where the quench initially started. In all other coils, no quench occurs due to thermal propagation or due to quench back (coupling loss-induced quenches). Since inter-filament coupling loss is modelled in LEDET (see chapter 2), the maximum temperatures reached in the turns due to this effect can be considered[1]. Quench back only occurs when the superconductor of other coils changes to the normal phase due to coupling losses and this happens when the critical surface of the superconductor is exceeded. Since current density and magnetic field strength are known, the necessary current sharing temperature T_{cs} can be calculated for the worst case for each measurement. For this, the CUDI1 formula 4.2 [134] with the constants calculated by `ParsimConductor` can simply be used. The coil of a magnet with the lowest measured critical current is also used for this calculation, as this represents the worst-case scenario.
As an example, figure 6.13 shows the temperature distribution within the individual turns in the 2D cross-section using the example of high field location simulation results of MCTSXF. In this figure it can be seen that the temperature in the coils where the quench did not start remains below 7.2 K.

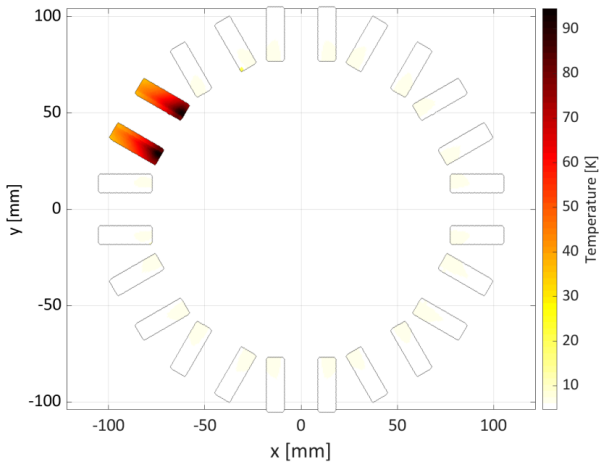
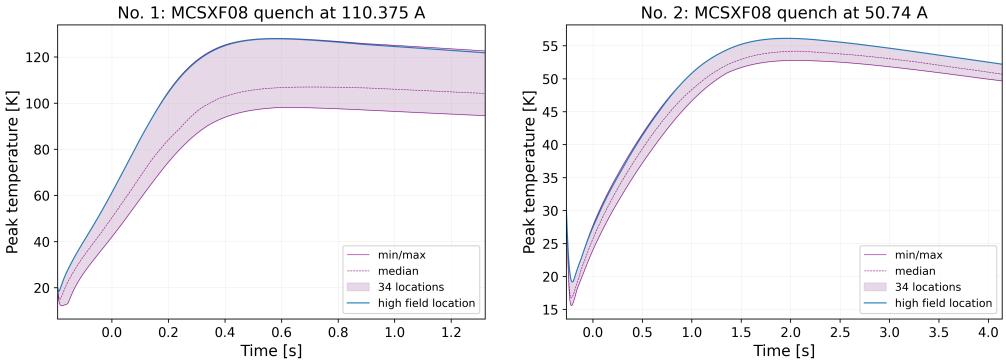


Figure 6.13: 2D temperature distribution of MCTSXF simulation with high field location (measurement no. 9) at time 0.41 s (close to the time where T_{cl} is maximum).

[1]Since there is no propagation of the quench from one coil to the next in the model (see chapter 5), the temperatures of the coils where the quench did not start can be considered. These only arise due to the coupling currents.

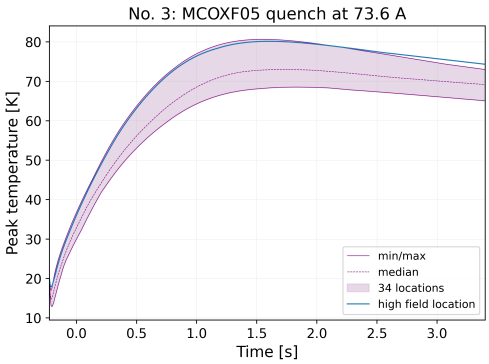
6.3 Peak temperature

As introduced in chapter 1, the peak hot-spot temperature is a central parameter when it comes to the protection of superconducting magnets. Since it cannot be measured easily, accurate simulation models are needed. The simulation results of the peak temperatures in the different chosen locations are shown in 6.14 and 6.15.

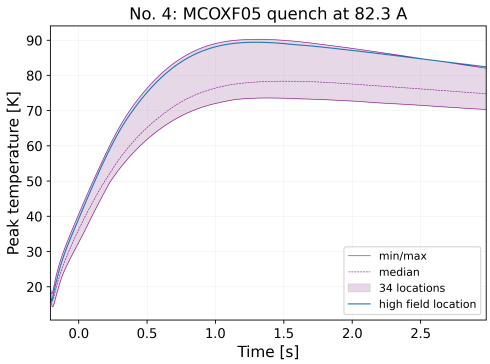


(a) No. 1

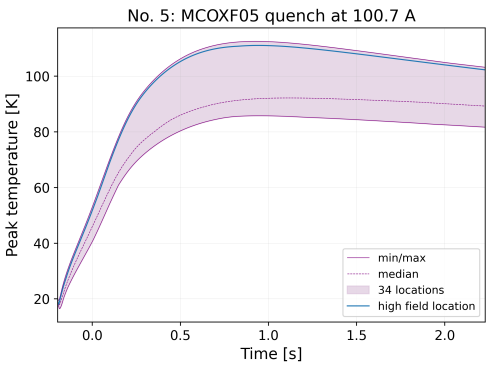
(b) No. 2



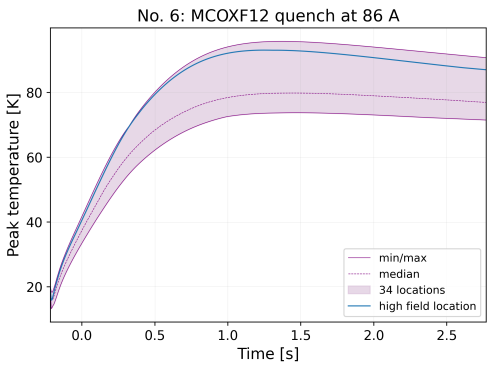
(c) No. 3



(d) No. 4

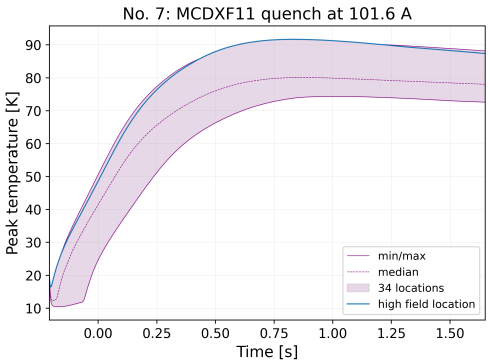


(e) No. 5

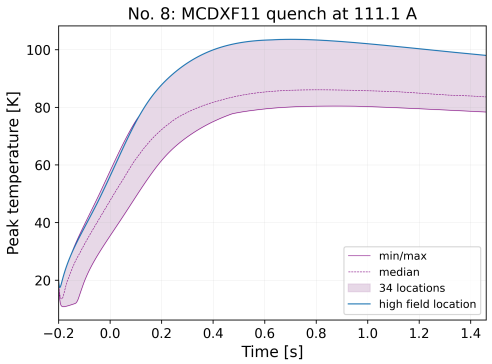


(f) No. 6

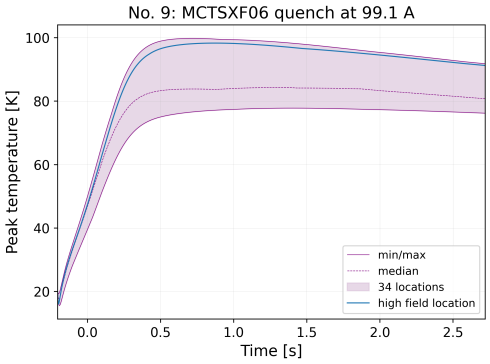
Figure 6.14: Simulated peak hot-spot temperatures for different locations for the simulated MCSXF and MCOXF events.



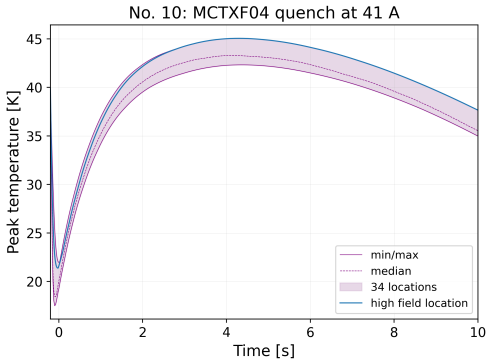
(a) No. 7



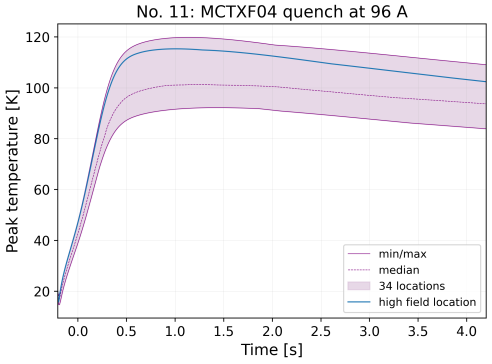
(b) No. 8



(c) No. 9



(d) No. 10



(e) No. 11

Figure 6.15: Simulated peak hot-spot temperatures for different locations for the simulated MCDXF, MCTSXF and MCTXF events.

Figure 6.16 provides an overview of the distribution of peak hot-spot temperatures by location for all measurements. It can be seen that all temperatures are well below a critical peak hot-spot temperature. Regarding the peak hot-spot temperature the self-protection design of the HL-LHC HOC magnets is therefore sufficient for quench protection. The fact that, with the exception of the two measurement, all current levels are well above the operating temperature confirms this conclusion.

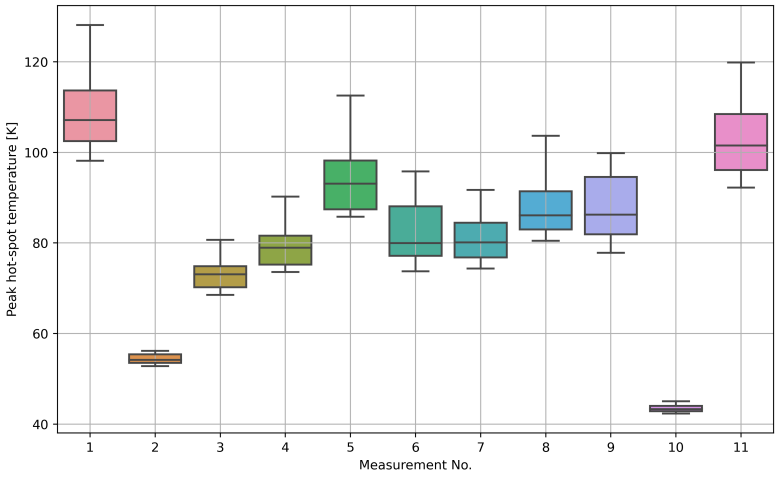


Figure 6.16: Boxplot of the peak temperatures reached for simulations with 34 selected quench locations corresponding to the measurements at hand.

7 Discussion and Conclusion

For the High Luminosity LHC self-protected high order corrector magnets were designed and built by LASA (INFN). This includes sextupole, octupole, decapole and dodecapole magnets made from superconducting Nb-Ti wires wound to racetrack coils. At an operating temperature of 1.9 K these magnets produce magnetic fields between 1.5 T and 2.3 T, and their nominal currents range from 84 A to 102 A.
Quench trasnsients in these magnets were simulated using the STEAM framework, currently under development at CERN. For this work, STEAM-FiQuS and STEAM-LEDET were used. STEAM-FiQuS was extended to generate racetrack geometries for general purposes and for the case of the HL-LHC HOC magnets. From this geometry together with the validated 3D magnetic field from OPERA, STEAM-LEDET was used to simulate the electromagnetic and thermal transients are then simulated in LEDET. Multiple interdependent physical effects are considered in the simulation program, including 3D thermal diffusion between turns, ohmic loss and inter-filament coupling loss.
Within the uncertainties of unknown parameters, the limitations of the used software and the irregular arrangement of the turns in the cross-section, the considered HL-LHC HOC magnets could be modelled accurately enough to be considered fully suitable to predict transients during the operation of these magnets in the LHC circuits. This was achieved by carefully searching for a single set of assumptions with the smallest possible deviations in the quench load. With this optimal set of assumptions, the models are accurate enough that almost all simulations are within the uncertainty of the unknown quench location when comparing the quench load to experimental data. Thus, a single set of assumptions could be found that is predictive for all magnet types and different current levels. The final magnet models reproduce the electro-magnetic and thermal transients sufficiently well so that the relative error in quench load is approximately within $\pm$ 17%, whereas the unknown quench location introduces uncertainties in the quench load of about 9-16%.
For MCSXF the simulated quench load in general is underestimated with respect to its measured value. The reason for this is probably that this type of magnet was wound in about 40% more epoxy, which acts as thermal resistance between turns and should decelerate the discharge. However, due to the uncertainty of the non-homogeneously distribution of the epoxy, the epoxy was neglected, also because the conductors are in very close contact to each other making the neglect of the epoxy more accurate than assuming a homogeneous distribution. A possible improvement of the model would be to investigate the epoxy distribution further, but needs more data that is not trivial to collect.
Additionally, a few potential improvements were identified in the thermal model. Conductors are approximated as rectangular and the conductive cooling imposed as boundary condition and ignoring helium effects significantly overestimates the cooling effect, and

needed to be scaled. Another improvement could be to adjust the simulation starting time for every quench location. Given that some simulations with a smaller temperature profile initialising the quench, show a notable delay in quench propagation. However, this would lead to significant increase in computational effort.
In contrast, the electrical model and coupling current model is accurate due to two main reasons. On the on hand, the `ParsimConductor` step automatically computes and adapts conductor parameters (such as RRR, fraction of copper, critical current fitting parameters) and thereby eliminates unknowns associated to these parameters. On the other hand, the implemented geometry with the racetrack coils and lead ends helped to model the 3D thermal diffusion and quench propagation along the conductor.
An analysis of the peak temperatures demonstrated that the self-protection strategy of the considered magnets is sufficient, and does not reach critical values. This supports the current research suggesting that the magnets can be safely operated in the LHC. In the simulations, most of which were above the operating current, the maximum achieved hot-spot temperatures never exceeded a value of 130 K for any considered quench location.
Further analysis of the temperature of the non-quenching coils showed that no quench back occurs in the examined HL-LHC HOC magnets. This analysis, along with the relatively short simulation times, is one of the main reasons for choosing STEAM-LEDET as the software in the first place.
The validated models are now part of the STEAM-models library and can be used by researchers across CERN and from various other institutions worldwide to simulate transients occurring in these LHC magnets. In addition, with the two implemented steps `ParsimEvent` and `ParsimConductor`, the adaptation of all necessary simulation parameters could be automated to obtain models tailored for each individual magnet with different conductor parameters. These two contributions bring the STEAM framework one step closer to the final goal of simulating all LHC events in an automated and reliable way.

Bibliography

[1] CERN. "The history of CERN". In: (2019). URL: https://timeline.web.cern.ch/timeline-header/89#:~:text=The%20European%20Organization%20for%20Nuclear%20Research%20is%20born,-At%20the%20sixth&text=On%2029%20September%201954%2C%20following,dissolved%20but%20the%20acronym%20remained..

[2] CERN. "Our Member States". In: (2023). URL: https://home.cern/about/who-we-are/our-governance/member-states.

[3] CERN. "Experiments". In: (2023). URL: https://home.cern/science/experiments.

[4] CERN. "The CERN Experimental Programme". In: (2023). URL: https://greybook.cern.ch.

[5] CERN. "Key Achivements". In: (2023). URL: https://home.cern/about/key-achievements.

[6] Carlo Rubbia. "Experimental observation of the intermediate vector bosons W^+, W^-, and Z^0". In: *Rev. Mod. Phys.* 57 (3 July 1985), pp. 699–722. DOI: 10.1103/RevModPhys.57.699. URL: https://link.aps.org/doi/10.1103/RevModPhys.57.699.

[7] The ALPHA Collaboration. "Confinement of antihydrogen for 1,000 seconds". In: *Nature Phys* 7 (2011), pp. 558–564. DOI: https://doi.org/10.1038/nphys2025.

[8] CERN. "Discovery of the Higgs Boson". In: (2023). URL: https://home.cern/science/physics/higgs-boson/how.

[9] ATLAS Collaboration. "Observation of a new particle in the search for the Standard Model Higgs boson with the ATLAS detector at the LHC". In: *Physics Letters B* 716.1 (2012), pp. 1–29. ISSN: 0370-2693. DOI: https://doi.org/10.1016/j.physletb.2012.08.020. URL: https://www.sciencedirect.com/science/article/pii/S037026931200857X.

[10] CERN. "Where the web was born". In: (2023). URL: https://home.cern/science/computing/where-web-was-born.

[11] Esma Mobs. "The CERN accelerator complex in 2019. Complexe des accélérateurs du CERN en 2019". In: (2019). General Photo. URL: https://cds.cern.ch/record/2684277.

[12] CERN. "CERN's accelerator complex". In: (2023). URL: https://home.cern/science/accelerators/accelerator-complex#:~:text=CERN%27s%20accelerator%20complex,-There%27s%20more%20to&text=The%20accelerator%20complex%20at%20CERN,next%20machine%20in%20the%20sequence..

[13] Werner Herr and B Muratori. "Concept of luminosity". In: (2006). DOI: 10.5170/CERN-2006-002.361. URL: https://cds.cern.ch/record/941318.

[14] E. Ravaioli. *CLIQ: A New Quench Protection Technology for Superconducting Magnets*. University of Twente, 2015. ISBN: 9789036539081. URL: https://books.google.fr/books?id=IvcMswEACAAJ.

[15] CERN. "Luminosity". In: (2023). URL: https://home.cern/news/opinion/cern/luminosity-why-dont-we-just-say-collision-rate.

[16] CERN. "High Luminosity LHC Project". In: (2023). URL: https://hilumilhc.web.cern.ch.

[17] CERN. "High-Luminosity LHC". In: (2023). URL: https://home.cern/science/accelerators/high-luminosity-lhc.

[18] Alonso et. al. *High-Luminosity Large Hadron Collider (HL-LHC): Technical design report*. Vol. 10. 2020. DOI: https://doi.org/10.23731/CYRM-2020-0010.

[19] CERN. "LHC / HL-LHC Plan". In: (updated in February 2022). URL: https://hilumilhc.web.cern.ch/article/ls3-schedule-change.

[20] CERN. "High-Luminosity LHC". In: (2023). URL: https://home.cern/resources/faqs/high-luminosity-lhc.

[21] CERN. "The SM18 test facility in the HL-LHC era". In: (2022). URL: https://home.cern/news/news/engineering/sm18-test-facility-hl-lhc-era.

[22] Michael Tinkham. *Introduction to superconductivity*. Courier Corporation, 2004.

[23] John Bardeen, Leon N Cooper, and John Robert Schrieffer. "Theory of superconductivity". In: *Physical review* 108.5 (1957).

[24] Charles P Poole, Horacio A Farach, and Richard J Creswick. *Superconductivity*. Academic press, 2013.

[25] Rafael Navarro. "Hundred years of Superconducting Materials". In: Dec. 2010.

[26] Lyndon Evans. "THE LARGE HADRON COLLIDER: A MARVEL OF TECHNOLOGY. CERN AND EPFL PRESS". In: (2009), p. 78. URL: https://cds.cern.ch/record/2645935/files/Evans2018.pdf.

[27] F. Steglich et al. "Superconductivity in the Presence of Strong Pauli Paramagnetism: $CeCu_2Si_2$". In: *Phys. Rev. Lett.* 43 (25 Dec. 1979), pp. 1892–1896. DOI: 10.1103/PhysRevLett.43.1892. URL: https://link.aps.org/doi/10.1103/PhysRevLett.43.1892.

[28] J. G. Bednorz and K. A. Mueller. "Possible highT c superconductivity in the Ba?La?Cu?O system". In: *Zeitschrift fuer Physik B Condensed Matter* 64.2 (June 1986), pp. 189–193. DOI: 10.1007/bf01303701. URL: https://doi.org/10.1007/bf01303701.

[29] E.A. Katz. "Chapter 13 - Fullerene Thin Films as Photovoltaic Material". In: *Nanostructured Materials for Solar Energy Conversion*. Ed. by Tetsuo Soga. Amsterdam: Elsevier, 2006, pp. 361–443. ISBN: 978-0-444-52844-5. DOI: https://doi.org/10.1016/B978-044452844-5/50014-7. URL: https://www.sciencedirect.com/science/article/pii/B9780444528445500147.

[30] Ying Yu et al. "Cubic C20: An intrinsic superconducting carbon allotrope". In: *Applied Physics Express* 16.6 (June 2023), p. 063003. DOI: 10.35848/1882-0786/acdc02. URL: https://dx.doi.org/10.35848/1882-0786/acdc02.

[31] Charles Day. "Iron-based superconductors". In: *Physics Today* 62.8 (Aug. 2009), pp. 36–40. DOI: 10.1063/1.3206093. URL: https://doi.org/10.1063/1.3206093.

[32] Gertjan Koster et al. "Structure, physical properties, and applications of $SrRuO_3$ thin films". In: *Rev. Mod. Phys.* 84 (1 Mar. 2012), pp. 253–298. DOI: 10.1103/RevModPhys.84.253. URL: https://link.aps.org/doi/10.1103/RevModPhys.84.253.

[33] Qiangqiang Gu and Hai-Hu Wen. "Superconductivity in nickel-based 112 systems". In: *The Innovation* 3.1 (2022), p. 100202. ISSN: 2666-6758. DOI: https://doi.org/10.1016/j.xinn.2021.100202. URL: https://www.sciencedirect.com/science/article/pii/S2666675821001272.

[34] Pia Jensen Ray. "Structural investigation of La(2-x)Sr(x)CuO(4+y)". In: (2016). URL: https://doi.org/10.6084/m9.figshare.2075680.v2.

[35] CERN. "Superconductivity in short". In: (2023). URL: https://www.lhc-closer.es/taking_a_closer_look_at_lhc/0.superconductivity_in_short.

[36] L. Bortot. "Introduction to Superconductivity and Superconducting Magnets". In: (2020). URL: https://indico.cern.ch/event/884989/contributions/4031687/attachments/2150900/3626515/2020_11_MiniLecture_Bortot.pdf.

[37] D. Leroy. "Review of the R&D and Supply of the LHC Superconducting Cables". In: *IEEE Transactions on Applied Superconductivity* 16.2 (2006), pp. 1152–1159. DOI: 10.1109/TASC.2005.864273.

[38] Larbalestier D. Gurevich A. Feldmann D. et al. "High-Tc superconducting materials for electric power applications". In: 414 (2001), pp. 368–377. DOI: https://doi.org/10.1038/35104654.

[39] Y Oz et al. "Influence of twist pitch on hysteretic losses and transport Jc in overpressure processed high Jc Bi-2212 round wires". In: *Superconductor Science and Technology* 35/6 (2022). DOI: 10.1088/1361-6668/ac68a8.

[40] T Schreiner and Andrzej Siemko. “Current distribution inside Rutherford-type superconducting cables and impact on performance of LHC dipoles”. In: (May 2023).

[41] Martin Wilson. “Advances in low loss nbti strand and cable”. In: *Proceedings - Workshop on Accelerator Magnet Superconductors, Design and Optimization, WAMSDO 2008* (Jan. 2009).

[42] Alternative Energies and Atomic Energy Commission (CEA). “The most powerful MRI scanner in the world delivers its first images”. In: (2021). URL: https://www.cea.fr/english/Pages/News/premieres-images-irm-iseult-2021.aspx.

[43] Anna Nowogrodzki. “The world’s strongest MRI machines are pushing human imaging to new limits”. In: *Nature* 563.7729 (Oct. 2018), pp. 24–26. DOI: 10.1038/d41586-018-07182-7. URL: https://doi.org/10.1038/d41586-018-07182-7.

[44] Marco Statera et al. “THE LARGE HADRON COLLIDER 2008–2013”. In: *International Journal of Modern Physics A* Vol. 28, No. 25, 1330035 (2013). DOI: 10.1142/S0217751X13300354.

[45] L. Bottura. “Cable Stability”. In: (2014). Comments: 51 pages, contribution to the CAS-CERN Accelerator School: Superconductivity for Accelerators, Erice, Italy, 24 April - 4 May 2013, edited by R. Bailey, pp. 401–451. DOI: 10.5170/CERN-2014-005.401. arXiv: 1412.5373. URL: https://cds.cern.ch/record/1974064.

[46] P Lebrun and L Tavian. *Cooling with Superfluid Helium*. 2014. DOI: 10.5170/CERN-2014-005.453. URL: http://cds.cern.ch/record/1974065.

[47] L Bottura. *Cable Stability*. 2014. DOI: 10.5170/CERN-2014-005.401. URL: http://cds.cern.ch/record/1974064.

[48] Linda Imbasciati. “Studies of Quench Protection in Nb_3Sn Superconducting Magnets for Future Particle Accelerators”. PhD thesis. Vienna, Tech.U., 2003. DOI: 10.2172/1419205.

[49] H. ten Kate, H. Boschman, and L. Van de Klundert. “Longitudinal propagation velocity of the normal zone in superconducting wires”. In: *IEEE Transactions on Magnetics* 23.2 (1987), pp. 1557–1560. DOI: 10.1109/TMAG.1987.1065020.

[50] Herman ten Kate. “Superconducting Magnets Quench Propagation and Protection”. In: (2013). URL: https://indico.cern.ch/event/194284/contributions/1472819/attachments/281522/393603/TenKate_-_CAS_-Handout-Quench-Erice-2103.pdf.

[51] Herman ten Kate. “(Multi) Normal Zone Propagation Velocity in high current density high field magnets”. In: (2013). URL: https://indico.cern.ch/event/199910/contributions/381641/attachments/297506/415833/TenKate_-_WAMSDO_2013_Propagation15Jan13.pdf.

[52] Stephan Russenschuck. *Field Computation for Accelerator Magnets*. Wiley, 2010. DOI: 10.1002/9783527635467. URL: https://doi.org/10.1002/9783527635467.

[53] G. Ambrosio. "Maximum allowable temperature during quench in Nb_3Sn accelerator magnets". In: (2013). Comments: 4 pages, Contribution to WAMSDO 2013: Workshop on Accelerator Magnet, Superconductor, Design and Optimization; 15 - 16 Jan 2013, CERN, Geneva, Switzerland, pp. 43–46. DOI: 10.5170/CERN-2013-006.43. arXiv: 1401.3955. URL: https://cds.cern.ch/record/1643437.

[54] K. H. Mess, Peter Schmueser, and S. Wolff. *Superconducting accelerator magnets*. 1996.

[55] J.H. Schultz. "Protection of superconducting magnets". In: *IEEE Transactions on Applied Superconductivity* 12.1 (2002), pp. 1390–1395. DOI: 10.1109/TASC.2002.1018662.

[56] F. Rodriguez-Mateos and F. Sonnemann. "Quench heater studies for the LHC magnets". In: *PACS2001. Proceedings of the 2001 Particle Accelerator Conference (Cat. No.01CH37268)*. IEEE. DOI: 10.1109/pac.2001.988141. URL: https://doi.org/10.1109/pac.2001.988141.

[57] K. Dahlerup-Petersen et al. "The protection system for the superconducting elements of the Large Hadron Collider at CERN". In: *Proceedings of the 1999 Particle Accelerator Conference (Cat. No.99CH36366)*. IEEE. DOI: 10.1109/pac.1999.792249. URL: https://doi.org/10.1109/pac.1999.792249.

[58] E. Todesco. "Unit 13 Protection systems". In: (2020). URL: https://indico.cern.ch/event/940961/attachments/2104840/3540058/13.pdf.

[59] F Rodríguez-Mateos et al. "Quench Heater Experiments on the LHC Main Superconducting Magnets". In: (2000). URL: https://cds.cern.ch/record/466520.

[60] F. Rodriguez-Mateos and F. Sonnemann. "Quench heater studies for the LHC magnets". In: *PACS2001. Proceedings of the 2001 Particle Accelerator Conference (Cat. No.01CH37268)*. Vol. 5. 2001, 3451–3453 vol.5. DOI: 10.1109/PAC.2001.988141.

[61] Renilde Vanden Broeck. "PROTECTING THE LHC SUPERCONDUCTING MAGNETS. PROTÉGER LES AIMANTS SUPRACONDUCTEURS DU LHC". 2019. URL: https://cds.cern.ch/record/2693576.

[62] L. Coull et al. "LHC magnet quench protection system". In: *IEEE Transactions on Magnetics* 30.4 (1994), pp. 1742–1745. DOI: 10.1109/20.305593.

[63] E. Ravaioli et al. "New, Coupling Loss Induced, Quench Protection System for Superconducting Accelerator Magnets". In: *IEEE Transactions on Applied Superconductivity* 24.3 (2014), pp. 1–5. DOI: 10.1109/TASC.2013.2281223.

[64] E. De Matteis et al. "Magnetic Measurements Results and Analysis of the First Batches of Superferric Magnets for the HL-LHC High Order Field Correction". In: *IEEE Transactions on Applied Superconductivity* 32.6 (2022), pp. 1–5. DOI: 10.1109/TASC.2022.3163118.

[65] M. Statera G. Volpini M. Sorbi. "The HL-LHC Nb-Ti magnets towards series production". In: (). URL: https://indico.psi.ch/event/8694/attachments/16116/27784/2021-01_wp3_iter.pdf.

[66] G. Volpini et al. “NbTi Superferric Corrector Magnets for the LHC Luminosity Upgrade”. In: *IEEE Transactions on Applied Superconductivity* 25.3 (2015), pp. 1–5. DOI: 10.1109/TASC.2014.2378377.

[67] M. Prioli M. Statera M. Sorbi. “HL-LHC Project Intranet: High Order Correctors”. In: (). URL: https://espace.cern.ch/HiLumi/WP3/SitePages/High%20Order%20Correctors.aspx.

[68] G. Volpini et al. “Development of the Superferric Sextupole Corrector Magnet for the LHC Luminosity Upgrade”. In: *IEEE Transactions on Applied Superconductivity* 26.4 (2016), pp. 1–4. DOI: 10.1109/TASC.2016.2543468.

[69] Marco Statera et al. “Construction and Cold Test of the First Superferric Corrector Magnet for the LHC Luminosity Upgrade”. In: *IEEE Transactions on Applied Superconductivity* 27.4 (2017), pp. 1–5. DOI: 10.1109/TASC.2017.2650957.

[70] Massimo Sorbi et al. “Status of the Activity for the Construction of the HL-LHC Superconducting High Order Corrector Magnets at LASA-Milan”. In: *IEEE Transactions on Applied Superconductivity* 28.3 (2018), pp. 1–5. DOI: 10.1109/TASC.2017.2772887.

[71] Marco Statera et al. “Construction and Cold Test of the Superferric Octupole for the LHC Luminosity Upgrade”. In: *IEEE Transactions on Applied Superconductivity* 28.4 (2018), pp. 1–5. DOI: 10.1109/TASC.2018.2809561.

[72] Marco Statera et al. “Construction and Cold Test of the Superferric Decapole for the LHC Luminosity Upgrade”. In: *IEEE Transactions on Applied Superconductivity* 29.5 (2019), pp. 1–5. DOI: 10.1109/TASC.2019.2907197.

[73] Massimo Sorbi et al. “Construction and Cold Test of the Superferric Dodecapole High Order Corrector for the LHC High Luminosity Upgrade”. In: *IEEE Transactions on Applied Superconductivity* 29.5 (2019), pp. 1–5. DOI: 10.1109/TASC.2019.2897113.

[74] L. Fiscarelli et al. “Magnetic Measurements on the Prototype Magnets of the High-Order Correctors for HL-LHC”. In: *IEEE Transactions on Applied Superconductivity* 29.5 (2019), pp. 1–5. DOI: 10.1109/TASC.2019.2899984.

[75] Marco Statera et al. “Construction and Power Test of the Superferric Skew Quadrupole for HL-LHC”. In: *IEEE Transactions on Applied Superconductivity* 30.4 (2020), pp. 1–5. DOI: 10.1109/TASC.2020.2979159.

[76] M. Prioli et al. “Completion of the Test Phase for the Hilumi LHC Skew Quadrupole Corrector Magnet”. In: *IEEE Transactions on Applied Superconductivity* 31.5 (2021), pp. 1–5. DOI: 10.1109/TASC.2021.3059986.

[77] Marco Statera et al. “Optimization of the High Order Correctors for HL-LHC Toward the Series Production”. In: *IEEE Transactions on Applied Superconductivity* 31.5 (2021), pp. 1–5. DOI: 10.1109/TASC.2021.3070904.

[78] M. Statera et al. “The HL-LHC High Order Correctors Series Production and Powering Tests Status”. In: *IEEE Transactions on Applied Superconductivity* 32.6 (2022), pp. 1–5. DOI: 10.1109/TASC.2022.3159318.

[79] M. Prioli et al. “Engineering specification: quench protection study for the hi-luminosity LHC hich order corrector magnets”. In: (). URL: https://edms.cern.ch/document/2370274/1.

[80] Matthias Mentinik. “Report: cross-check of protection studies of the HL-LHC high order corrector magnets”. In: (). URL: https://edms.cern.ch/document/2222750/1.3.

[81] “CERN steam framework webpage”. In: (). URL: https://espace.cern.ch/steam.

[82] Lucio Rossi and Malou Sorbi. “Qlasa: A Computer Code For Quench Simulation In Adiabatic Multicoil Superconducting Windings”. In: (Jan. 2004). DOI: 10.15161/oar.it/1448979849.04.

[83] “LTSpice webpage”. In: (). URL: https://www.analog.com/en/design-center/design-tools-and-calculators/ltspice-simulator.html.

[84] EMMANUELE RAVAIOLI. “STEAM user agreement”. 2020. URL: https://edms.cern.ch/document/2024516.

[85] “Steam-LEDET webpage”. In: (). URL: https://espace.cern.ch/steam/_layouts/15/start.aspx#/LEDET/Home.aspx.

[86] E. Ravaioli et al. “Lumped-Element Dynamic Electro-Thermal model of a superconducting magnet”. In: *Cryogenics* (2016). ISSN: 0011-2275. DOI: http://dx.doi.org/10.1016/j.cryogenics.2016.04.004. URL: http://www.sciencedirect.com/science/article/pii/S0011227516300832.

[87] Mariusz Wozniak Emmanuele Ravaioli. “STEAM Software Framework for Simulation of Transient Effects in Accelerator Magnets and Circuits”. In: (). URL: https://indico.cern.ch/event/1161448/.

[88] Matthias Mentink. “BBQ (BusBar Quench)”. In: (). URL: https://indico.cern.ch/event/808547/contributions/3367239/attachments/1861112/3059472/STEAM-BBQ.pdf.

[89] “COMSOL Multiphysics webpage”. In: (). URL: https://www.comsol.com.

[90] “MATLAB webpage”. In: (). URL: https://www.mathworks.com/products/matlab.html.

[91] “SIGMA webpage”. In: (). URL: https://gitlab.cern.ch/steam/steam-sigma.

[92] “FiQuS webpage”. In: (). URL: https://pypi.org/project/fiqus/.

[93] “FiQuS developer repository”. In: (). URL: https://gitlab.cern.ch/steam/steam-fiqus-dev/-/tags/2023.9.1.

[94] “Gmsh webpage”. In: (). URL: https://gmsh.info.

[95] “GetDP webpage”. In: (). URL: https://getdp.info.

[96] Matthias Mentink. “STEAM-ProteCCT (Protection of Canted-Cosine-Theta-type Magnets)”. In: (). URL: https://indico.cern.ch/event/817839/contributions/3414305/attachments/1839116/3049992/STEAM-ProteCCT.pdf.

[97] “PSpice webpage”. In: (). URL: https://www.pspice.com.

[98] Sandia National Laborities. “XYCE webpage”. In: (). URL: https://xyce.sandia.gov.

[99] “STEAM-COSIM webpage”. In: (). URL: https://espace.cern.ch/steam/COSIM.

[100] Michal Maciejewski. “Co-Simulation of Transient Effects in Superconducting Accelerator Magnets”. Presented 10 May 2019. 2018. URL: https://cds.cern.ch/record/2675039.

[101] “steam_models repository”. In: (). URL: https://gitlab.cern.ch/steam/steam_models.

[102] “ROXIE webpage”. In: (). URL: https://roxie.docs.cern.ch.

[103] “steam_sdk repository”. In: (). URL: https://gitlab.cern.ch/steam/steam_sdk/-/tags/2023.9.1.

[104] “steam-material-library repository”. In: (). URL: https://gitlab.cern.ch/steam/steam-material-library.

[105] Frank P Incropera et al. *Fundamentals of heat and mass transfer*. en. 6th ed. Nashville, TN: John Wiley & Sons, Sept. 2007. URL: https://lccn.loc.gov/2005058360.

[106] Emmanuele Ravaioli et al. “Quench Transient Simulation in a Self-Protected Magnet with a 3D Finite-Difference Scheme”. In: *IEEE Trans. Appl. Supercond.* 32.6 (2022), p. 4005205. DOI: 10.1109/tasc.2022.3162798. URL: https://cds.cern.ch/record/2842571.

[107] E. Ravaioli et al. “Quench Transient Simulation in a Self-Protected Magnet With a 3-D Finite-Difference Scheme”. In: *IEEE Transactions on Applied Superconductivity* 32.6 (2022), pp. 1–5. DOI: 10.1109/TASC.2022.3162798.

[108] Lennard Bender. “Simulation of quench behaviour of the recombination dipole magnet for the LHC High Luminosity upgrade ”. Presented 08 Nov 2022. 2022. URL: https://cds.cern.ch/record/2838807.

[109] E. Ravaioli et al. “Modeling of Interfilament Coupling Currents and Their Effect on Magnet Quench Protection”. In: *IEEE Transactions on Applied Superconductivity* 27.4 (2017), pp. 1–8. DOI: 10.1109/TASC.2016.2636452.

[110] A P Verweij. “Electrodynamics of superconducting cables in accelerator magnets”. Presented on 15 Sep 1995. 1995. URL: https://cds.cern.ch/record/292595.

[111] Marvin Janitschke et al. “A Simplified Approach to Simulate Quench Development in a Superconducting Magnet”. In: *IEEE Transactions on Applied Superconductivity* 31.5 (2021), pp. 1–5. DOI: 10.1109/TASC.2021.3059980.

[112] Emmanuele Ravaioli. “Simulation of the 3D magnet quench process with the finite-difference method in STEAM-LEDET”. 2022. URL: https://edms.cern.ch/document/2454468/2.

[113] Emmanuele Ravaioli Ola Tranum Arnegaard. “3D simulations of quench transients in the MQSX magnet using STEAM-LEDET”. 2021. URL: https://edms.cern.ch/document/2613181/1.

[114] LASA (INFN). "Construction drawing of the HL-LHC sextupole corrector coil". In: (2020). URL: https://edms.cern.ch/ui/#!master/navigator/document?D:100561726:100561726:subDocs.

[115] "OPERA webpage". In: (). URL: https://www.3ds.com/products-services/simulia/products/opera/.

[116] Stephan Russenschuck. "ROXIE: the Routine for the Optimization of Magnet X-sections, Inverse Field Computation and Coil End Design". In: (Jan. 1998).

[117] LASA (INFN). "HL-LHC SEXTUPOLE CORRECTOR CONNECTION PCB". 2020. URL: https://edms.cern.ch/ui/file/2317706/AC/lhcmcsxf0004-vAC.pdf.

[118] LASA (INFN). "HL-LHC OCTUPOLE CORRECTOR CONNECTION PCB". 2020. URL: https://edms.cern.ch/ui/file/2317701/AC/lhcmcoxf0004-vAC.pdf.

[119] LASA (INFN). "HL-LHC DECAPOLE CORRECTOR CONNECTION PCB". 2020. URL: https://edms.cern.ch/ui/file/2317696/AC/lhcmcdxf0004-vAC.pdf.

[120] LASA (INFN). "HL-LHC DODODECAPOLE (LONG) CORRECTOR CONNECTION PCB". 2020. URL: https://edms.cern.ch/ui/file/2317716/AC/lhcmctxf0004-vAC.pdf.

[121] LASA (INFN). "HL-LHC DODODECAPOLE (SHORT) CORRECTOR CONNECTION PCB". 2020. URL: https://edms.cern.ch/ui/file/2317711/AC/lhcmctsxf0004-vAC.pdf.

[122] Olivier HOUSIAUX. "Visual Inspection @CERN - HCMCTSXF001-X5000006 ". 2022. URL: https://edms.cern.ch/ui/#!master/navigator/document?P:100441947:101093180:subDocs.

[123] Emmanuele Ravaioli. "LEDET manual". 2022. URL: https://edms.cern.ch/ui/file/2046824/2.02.11/LEDETmanual.pdf.

[124] "pandas library webpage". In: (). URL: https://pandas.pydata.org.

[125] LASA (INFN). "Construction drawing of the HL-LHC octupole corrector coil". In: (2020). URL: https://edms.cern.ch/ui/#!master/navigator/document?P:1118913318:100561721:subDocs.

[126] LASA (INFN). "Construction drawing of the HL-LHC decapole corrector coil". In: (2020). URL: https://edms.cern.ch/ui/#!master/navigator/document?P:1118913318:100561716:subDocs.

[127] LASA (INFN). "Construction drawing of the HL-LHC dodecapole corrector coil (long)". In: (2020). URL: https://edms.cern.ch/ui/#!master/navigator/document?P:1118913318:100561736:subDocs.

[128] LASA (INFN). "Construction drawing of the HL-LHC dodecapole corrector coil (short)". In: (2020). URL: https://edms.cern.ch/ui/#!master/navigator/document?P:1118913318:100561731:subDocs.

[129] "mplcursor library webpage". In: (). URL: https://mplcursors.readthedocs.io/en/stable/.

[130] Daniel Mayr. "Permanent link to filter mask computation code". 2023. URL: https://gitlab.cern.ch/steam/steam_models/-/blob/7ae156e633e426dfb5db77a53a6aa871resources/magnets/useful_python_scripts/calculate_fcooling_params.py.

[131] CERN. "MTF database". In: (). URL: https://edms5.cern.ch/pls/asbuilt/mtf_eqp_sel.adv_list_top.

[132] "pydantic library webpage". In: (). URL: https://docs.pydantic.dev/1.10/.

[133] L. Bottura. "A practical fit for the critical surface of NbTi". In: *IEEE Transactions on Applied Superconductivity* 10.1 (2000), pp. 1054–1057. DOI: 10.1109/77.828413.

[134] Arjan Verweij. "CUDI manual". In: (2007). URL: https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&ved=2ahUKEwizi8DjgI7_AhWsgv0HHRI_A3YQFnoECBcQAQ&url=https%5C%3A%5C%2F%5C%2Fcern-verweij.web.cern.ch%5C%2FManual%5C%2520CUDI%5C%2520v13.doc&usg=AOvVaw2i5M14pf-xjb3w3qEVboCM.

[135] L. T. Summers et al. "A model for the prediction of Nb3Sn critical current as a function of field, temperature, strain, and radiation damage". In: *IEEE Trans. Magnetics* 27 (1991), pp. 2041–2044. DOI: 10.1109/20.133608.

[136] Gianluca De Marzi et al. "Magnetic and Electromechanical Characterization of a High-JC RRP Wire for the HL-LHC MQXF Cable". In: *IEEE Transactions on Applied Superconductivity* 32.6 (2022), pp. 1–5. DOI: 10.1109/TASC.2022.3167655.

[137] Jack Ekin et al. "Extrapolative Scaling Expression: A Fitting Equation for Extrapolating Full Ic (B,T,) Data Matrixes From Limited Data". In: *IEEE Transactions on Applied Superconductivity* 27.4 (2017), pp. 1–7. DOI: 10.1109/TASC.2017.2647852.

[138] B Bordini et al. "An exponential scaling law for the strain dependence of the Nb3Sn critical current density". In: *Superconductor Science and Technology* 26.7 (May 2013), p. 075014. DOI: 10.1088/0953-2048/26/7/075014. URL: https://dx.doi.org/10.1088/0953-2048/26/7/075014.

[139] "scipy library webpage". In: (). URL: https://scipy.org.

[140] Mangiarotti Franco. "HCLMQXFBT01-CR000005 (MQXFB02) test plan". In: (), p. 3. URL: https://edms.cern.ch/ui/#!master/navigator/document?D:101203197:101203197:subDocs.

[141] "CSV icon image source". 2023. URL: https://www.freepik.com/icon/csv_6133884.

[142] "YAML icon image source". 2023. URL: https://www.vectorstock.com/royalty-free-vector/yaml-file-document-icon-vector-24671866.

[143] "Python icon image source". 2023. URL: https://www.iconfinder.com/icons/4518857/python_icon.

[144] "PDF icon image source". 2023. URL: https://en.m.wikipedia.org/wiki/File:PDF_icon.svg.

[145] "HTML icon image source". 2023. URL: https://www.freepik.com/icon/html-document_103077.

[146] "MAT icon image source". 2023. URL: https://www.file-extension.info/pl/format/mat.

[147] INFN. "HL-LHC HOC cable reception assets". 2020. URL: https://edms.cern.ch/ui/#!master/navigator/document?D:100631278:100631278:subDocs.

[148] INFN. "Manufacturing records, MCSXF assets: Winding - HCMCSXFC07-X5000013". 2020. URL: https://edms.cern.ch/file/2367316/1/06_013.pdf.

[149] INFN. "HL-LHC DODODECAPOLE (SHORT) CORRECTOR CONNECTION PCB". 2020. URL: https://edms.cern.ch/file/2511825/1/FAT-18333-M06-08_EDMS_2511825.xlsx.

[150] LASA (INFN). "HL-LHC SEXTUPOLE CORRECTOR MAGNET 1". 2020. URL: https://edms.cern.ch/ui/#!master/navigator/document?P:1129466932:100631650:subDocs.

[151] LASA (INFN). "HL-LHC SEXTUPOLE CORRECTOR MAGNET 2". 2020. URL: https://edms.cern.ch/ui/#!master/navigator/document?P:1129466932:100631651:subDocs.

[152] Y Thurel, D Nisbet, and B Favre. "Four quadrant 120 A, 10 V power converters for LHC". In: (2007). URL: https://cds.cern.ch/record/1058542.

[153] L. Bottura. "Magnet Quench 101. Magnet Quench 101". In: (2013). Comments: 9 pages, Contribution to WAMSDO 2013: Workshop on Accelerator Magnet, Superconductor, Design and Optimization; 15 - 16 Jan 2013, CERN, Geneva, Switzerland, pp. 1–9. DOI: 10.5170/CERN-2013-006.1. arXiv: 1401.3927. URL: http://cds.cern.ch/record/1643429.

[154] Yukikazu Iwasa and Kenneth R. Marken. "Case Studies in Superconducting Magnets: Design and Operational Issues". In: 1994, pp. 472–474.

Verpflichtungs- und Einverständniserklärung

Ich erkläre, dass ich meine Masterarbeit selbständig verfasst und alle in ihr verwendeten Unterlagen, Hilfsmittel und die zugrunde gelegte Literatur genannt habe.

Ich nehme zur Kenntnis, dass auch bei auszugsweiser Veröffentlichung meiner Masterarbeit die Universität, das/die Institut/e und der/die Arbeitsbereich/e, an dem/denen die Masterarbeit ausgearbeitet wurde, und die Betreuerin/nen bzw. der/die Betreuer zu nennen sind.

Ich nehme zur Kenntnis, dass meine Masterarbeit zur internen Dokumentation und Archivierung sowie zur Abgleichung mit der Plagiatssoftware elektronisch im Dateiformat pdf ohne Kennwortschutz bei der/dem Betreuer/in einzureichen ist, wobei auf die elektronisch archivierte Masterarbeit nur die/der Betreuerin/Betreuer der Masterarbeit und das studienrechtliche Organ Zugriff haben.

Innsbruck am . . .