Simulation of Electro-Thermal Transients in Superconducting Accelerator Magnets with COMSOL Multiphysics ®

L. Bortot[*1], M. Maciejewski[*1,2], M. Prioli[1], A.M. Fernandez Navarro, S. Schöps[3], I. Cortes Garcia[3], B. Auchmann[1] and A.P. Verweij[1]

[1]CERN, Geneva, Switzerland, [2]Łódź University of Technology, Łódź, Poland,
[3]Technische Universität Darmstadt, Darmstadt, Germany

*Corresponding authors: lorenzo.bortot@cern.ch, michal.maciejewski@cern.ch

Abstract: Superconducting accelerator magnets are a cutting-edge technology and require sophisticated monitoring and means of protection, in case superconductivity is lost during operation. Numerical simulations play a crucial role in understanding transient phenomena occurring within the magnet, helping in the prevention of potentially disruptive consequences. We present the electro-thermal model of a twin-aperture superconducting dipole magnet. Due to the complexity of the model, the implementation is achieved through a Java framework that relies on the COMSOL API. We discuss an equivalent magnetization formulation for the cable eddy-currents, and the way it is implemented in COMSOL. The model simulates an emergency stop and provides the temperature map in the magnet, and the voltage evolution on the magnet's terminals.

Keywords: Superconducting magnet, Equivalent magnetization, Eddy-currents, Quench, LHC, COMSOL API.

1 Introduction

Circular accelerators for high-energy particle physics require intense magnetic fields to control the trajectories of particle beams [1]. These fields are obtained using iron-yoked electromagnets (Fig. 1) [2], wound with fully transposed rectangular cables (see Fig. 2): the micro-metric Nb-Ti filaments are embedded in a copper matrix, forming a strand. Strands are twisted and wrapped with a 140-µm-thick polyamide insulation layer. The cable is cooled down to 1.9 K to let the filaments reaching the superconducting state, allowing the cable to carry high current densities, up to 1000 A/mm^2.
A quench is a sudden transition from the superconducting to the normal conducting state, forcing the conduction current to commute from the filaments to the copper matrix. Quenches occur whenever the superconductor's working point moves outside the so-called critical surface (Fig. 3), causing the release of the energy stored in the magnetic field as Ohmic losses. Quenches

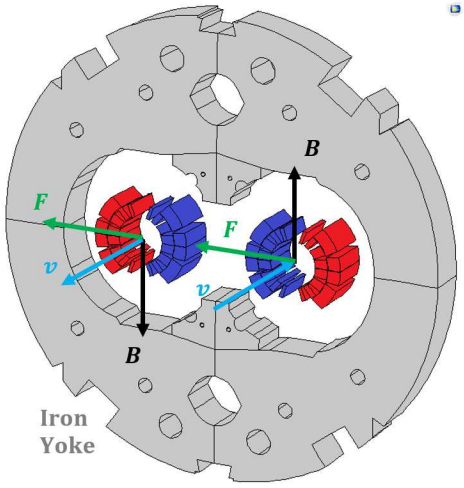

Figure 1: Magnet's cross-section. The interaction of the dipole magnetic field **B** [T] with the motion of the protons **v** [m/s] generates a Lorentz force and **F** [N] that keeps the particles on a circular trajectory. A non-magnetic steel collar (not shown in the picture) keeps the coils in place.

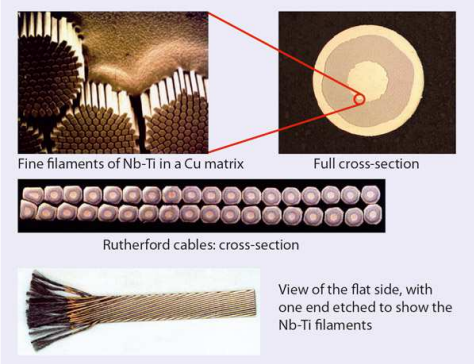

Figure 2: Cable layout (Source: cerncourier.com).

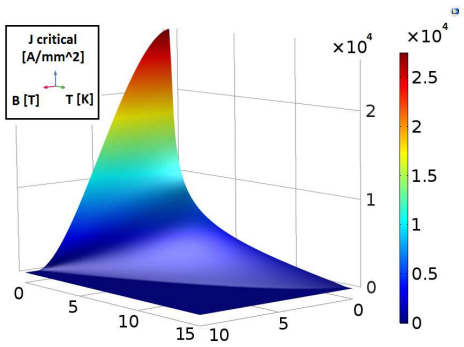

Figure 3: Critical surface for Nb-Ti [3], showing the maximum allowed current density as function of temperature and magnetic field.

cannot always be avoided and must be considered as a possible operational scenario. Generally, dedicated quench detection and protection systems are in place to avoid overheating of the coil. A careful analysis of the ensuing electro-thermal transient is of great importance for the design of the magnet, the quench protection system, as well as for the safe magnet operation.
We present a coupled electro-thermal 2-D model of a superconducting magnet. The model accounts for the non-linear temperature- and field-dependent material properties and for the induced eddy-currents in the superconducting cable, enabling the calculation of quench initiation and propagation. The construction of the magnet's cross-section is realized considering the coil is composed of single turns treated as basic bricks over which material properties and physical laws are homogenized. Since a coil's cross-section usually consists of hundreds of turns, the model is automatically built through an external Java framework that calls the necessary COMSOL API functions.

2 Governing Equations

The proposed electro-thermal formulation combines the magnetic vector potential formulation with the heat balance equation.

2.1 Definitions

The following symbols are introduced:
- $\vec{H}$ Magnetic field;
- $\vec{B}$ Magnetic flux density;
- $\vec{A}$ Magnetic vector potential;
- μ_0 Vacuum permeability;
- μ_r Material permeability;
- σ Electrical conductivity;
- $\vec{M}$ Magnetization;
- P Specific power density;
- $\vec{J}_{ext}$ External current density;
- ρ Density;
- C_p Heat capacity at constant pressure;
- Q Specific heat source;
- k Thermal conductivity;
- T Temperature.

2.2 Magnetic Vector Potential Formulation

Given the standard set of Maxwell equations, the magneto quasi-static approximation is assumed, leading to the following formulation:

$$\begin{cases} \nabla \times \vec{A} = \vec{B} \\ \nabla \times \left(\mu_0^{-1} \mu_r^{-1} \nabla \times \vec{A} \right) = \vec{J}_{ext} + \sigma \dfrac{\partial \vec{A}}{\partial t} + \nabla \times \vec{M} \end{cases}$$

The formulation is not suitable to compute cable eddy-currents, since their contribution requires the model to resolve the micro-metric scale of the cable's superconducting filaments, resulting in an unacceptable number of nodes. For this reason, cable eddy-currents are included in terms of an equivalent cable magnetization $\vec{M}_{eddy}$ [2], [4]. The formulation couples *de-facto* the laws of Faraday-Neumann and Ampere-Maxwell [5], relating the magnetization $\vec{M}$ with the derivative of the magnetic flux density $\vec{B}$ through an equivalent cable time-constant τ_{eq}, which can be explicitly calculated if the induced-currents' paths are known *a-priori*:

$$\mu_0 \mu_r \vec{M}_{eddy} = \tau_{eq} \frac{\partial \vec{B}}{\partial t}$$

The eddy-current contribution appears now in the magnetic constitutive relation:

$$\vec{B} = \mu_0 \mu_r \left(\vec{H} + \vec{M}_{eddy} \right)$$

The magneto-static formulation can hence be modified accordingly, removing $\partial \vec{A}/\partial t$ that is now taken into account by $\vec{M}_{eddy}$:

$$\begin{cases} \nabla \times \vec{A} = \mu_0 \mu_r \vec{H} + \tau_{eq} \dfrac{\partial \vec{B}}{\partial t} \\ \nabla \times \left(\mu_0^{-1} \mu_r^{-1} \nabla \times \vec{A} \right) = \vec{J}_{ext} + \nabla \times \vec{M}_{eddy} \end{cases}$$

The losses associated to the eddy-currents are included in terms of the magnetic power density, representing the specific energy P_{eddy} deposited in the superconductor [6]:

$$P_{eddy} = \vec{M}_{eddy} \cdot \left(\frac{\partial \vec{B}}{\partial t} \right)$$

2.3 Heat Balance

The heat balance equation relates the heat sources with the energy stored in the system, and the one that is transmitted by conduction according to the Fourier equation. The temperature map is obtained from

$$\rho C_p \frac{\partial T}{\partial t} + \nabla \cdot \vec{q} = Q, \qquad \vec{q} = -k \nabla T$$

The heat source Q includes two contributions:

$$Q = P_{eddy} + P_{Joule}$$
$$P_{Joule} = q_{flag} \left(\rho_{Cu} J_{ext}^{\ 2} \right)$$

where P_{eddy} accounts for the eddy-currents losses, and P_{Joule} is related to Joule losses arising in case of a quench, represented by the q_{flag} step function.

3 Automated Model Implementation

3.1 Motivation

A superconducting accelerator magnet is composed of a considerable number of half-turns (320 for the LHC Main Dipole, see Fig. 4), each of which needs to be set up with its respective variables and operators, needed to calculate relevant quantities. To avoid tedious and error-prone manual model creation, an automated workflow in Java based on the COMSOL API has been developed (Fig. 5).

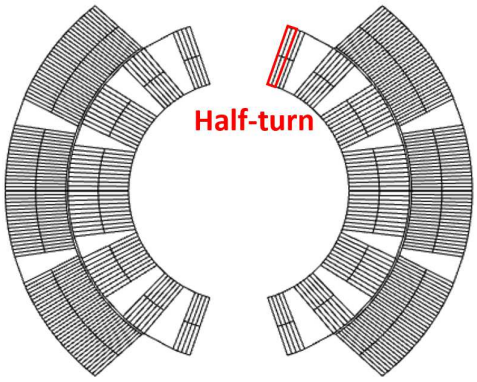

Figure 4: Geometry of a magnet's coil cross-section.

3.2 Architecture

The Java application has been structured following three main functional layers:

1 The top layer allows the user to describe the desired model, via text input files;
2 The middle layer contains all the numerical methods necessary to formulate the parameters needed as input for the API, avoiding the necessity of using explicitly the COMSOL GUI. For example, it is in charge of defining the geometrical position of the half turns;
3 The bottom layer provides the classes that embed the functionalities of the COMSOL API.

In the remainder of the section, we report the middle layer, which implements the COMSOL GUI tree-structure representation of a model.

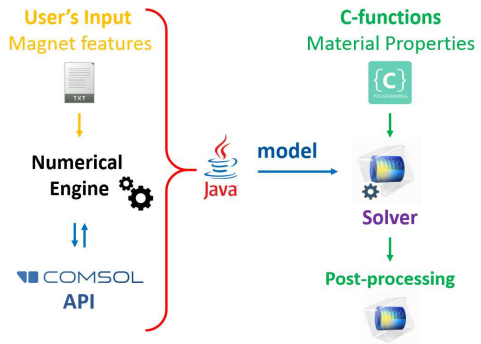

Figure 5: Automated workflow architecture

3.3 General Definitions

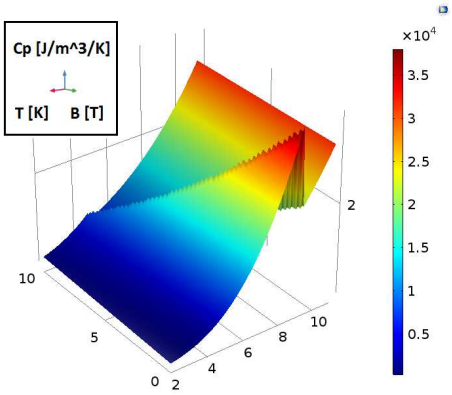

Figure 6: Thermal capacity of Nb-Ti, for a given current of 0 [A].

The general definition node contains global simulation flags allowing to perform various simulation scenarios. The quench transition function and the material properties are implemented in C, then compiled as dynamically linked libraries, and imported to the model using the *External function* node. This implementation has shown to be the most performant, as well as flexible enough to implement highly nonlinear, multi-parametric functions. Fig. 6 shows as example the thermal capacity of Nb-Ti. The highly non-linear behavior is due to the superconducting-to-normal transition as function of T [K], B [T] and I [A], as shown in Fig. 3.

3.4 Definitions

Each coil half-turn has to be characterized with its own set of specific geometrical and physics properties, and dedicated *Component couplings* operators have to be assigned. The half-turns are implemented with the use of the *indexing feature*. This feature allows to redefine variables that have a formulation common to each turn, with individual coefficients depending on turns' position and size. As a consequence, the implementation of materials and physics is lightweight and requires providing only a single variable (like heat capacity, external current density, etc.) for a group of domains (coil, yoke, air gap, etc.) sharing the same property.

3.5 Geometry Module

A dedicated Java geometry module has been developed allowing the user to define geometrical primitives, such as points, lines, and areas. Out of these constructions the superconducting magnet geometry is built. The geometry module incorporates model symmetries, which in turn reduces user input and simplifies the model building process. Geometrical entities in COMSOL are developed with use of *Parametric Curves*, which are connected together by means of the *Convert to Solid* feature in order to form domains.

3.6 Physics

The z-component of the magnetic vector potential is discretized by using second order nodal elements, and solved using the *Magnetic field (mf)* module. To implement the equivalent magnetization formulation for the coil, the *Magnetization* is chosen as constitutive relation for the related Ampere's Law node, and proper (M_x, M_y) coefficients are put in place. Moreover, to disable the $\partial \vec{A}/\partial t$ contribution (Sec. 2.1) in the node, it is necessary to access the *Equation View* editor and modify the variables set as shown in Tab. 1.

Name	Expression	Unit
mf.Jix	0	A/m^2
mf.Jiy	0	A/m^2
mf.Jiz	0	A/m^2

Table 1: Setting of induced currents in the Magnetic Field (mf) module.

To solve the heat balance equation, the *Heat Transfer in Solids (ht)* module has been selected. The key feature is the *Thin layer*, which allows to represent the thermal behavior of the polyamide layer, which insulates each turn. This avoids the necessity of resolving them explicitly in the mesh.

3.7 Meshing

While the model has been meshed with the primary aim of minimizing the number of elements, thus the computational time, a dedicated mesh sensitivity analysis was performed to ensure the accuracy of results. The resulting mesh is a combination of structured and unstructured elements, used for the discretization

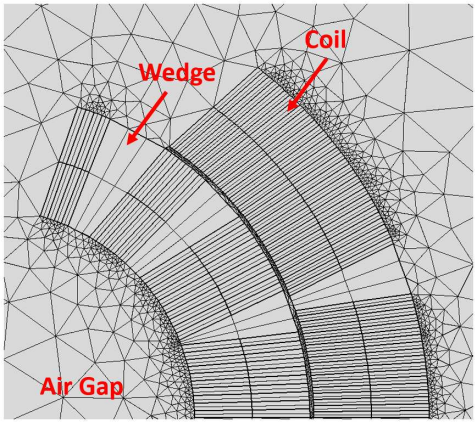

Figure 7: Mesh detail.

of the coil, the wedges and the surrounding air gap region (Fig. 7).
Moreover, for the air gap domain the maximum element growth rate has been raised to 2, to reduce the magnitude of the mesh-packing phenomenon occurring at the coil interface. The mesh details are reported in Tab. 2.

Mesh Element	Quantity
Quadrilateral	2748
Triangular	22928
Edge	6220
Total	**31896**

Table 2: Mesh statistics.

3.8 Computation

The simulation has been organized in two consecutive, time dependent studies. In the first study the current in the magnet is linearly ramped-up to the nominal value. The second study uses the final state of the first one as initial condition and simulates an exponential current decay with a time constant of 0.1 s. To reach the required numerical stability, the numerical solver has been properly adjusted, accordingly to Tab. 3.

Feature	Value
Steps taken by solver	Strict
Max time step	500e-6 s
Jacobian update	On every iteration
Max number of iterations	10
Tolerance factor	1e-3

Table 3: Custom settings for the numerical solver.

4 Results

The results here presented show the magnet's behavior during a fast discharge, as explained in Sec. 3.8. Starting from nominal conditions (Fig. 8), the variation of the field is generating eddy-currents, whose equivalent magnetization is shown in Fig. 9. These losses have two main consequences on the magnet. On the one hand, they are affecting the magnetic field, so they influence the magnet equivalent electrical impedance. On the other hand, they deposit energy in the magnet coil, see Fig. 10, dissipating part of the energy stored in the magnetic field. These losses can be high enough to heat up the superconductor beyond the critical surface,

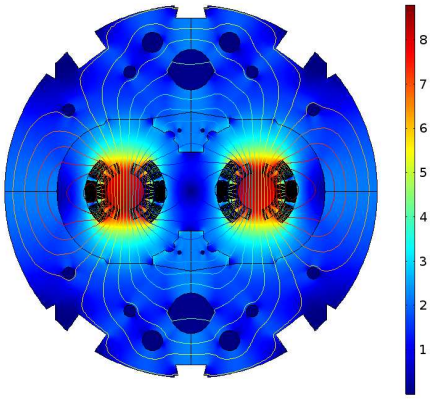

Figure 8: Magnetic field in a superconducting dipole magnet at nominal current, in T.

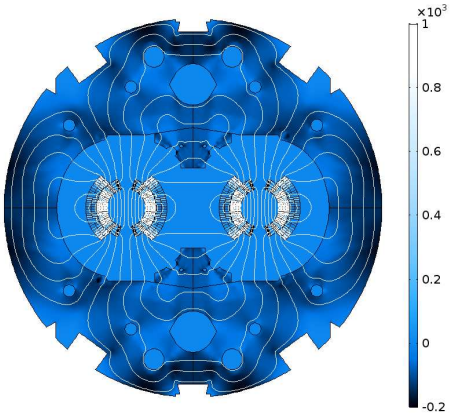

Figure 9: Eddy-currents' equivalent magnetization in A/m, as a difference of two solutions, during a linear ramp-up of 100 A/s, at 8 kA.

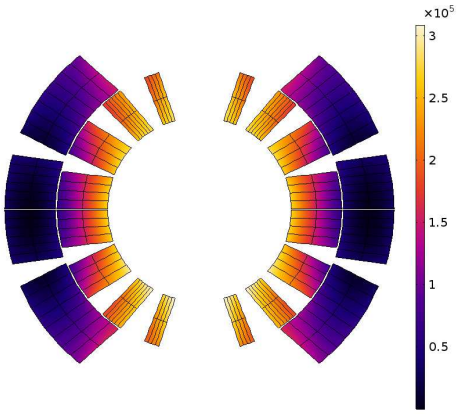

Figure 10: Eddy-currents losses deposited in the coil, in W/m^3.

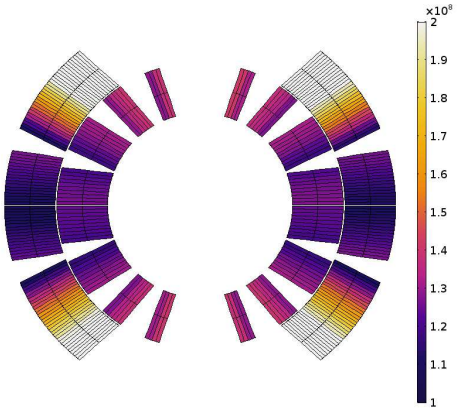

Figure 11: Ohmic losses deposited in the coil, in W/m^3.

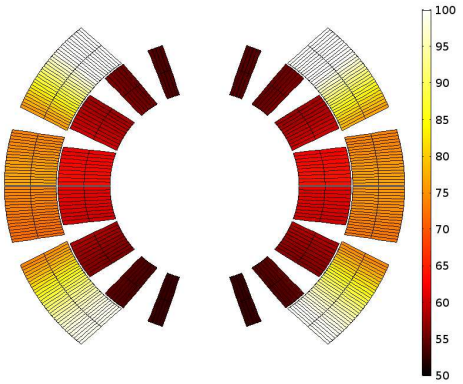

Figure 12: Temperature distribution in the coil, in K.

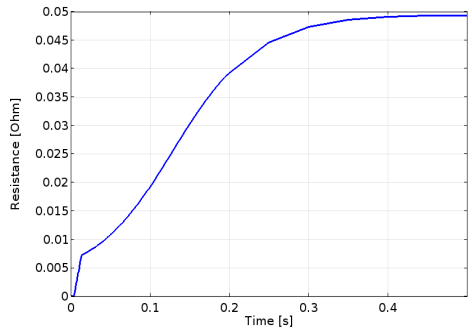

Figure 13: Coil resistance as function of time.

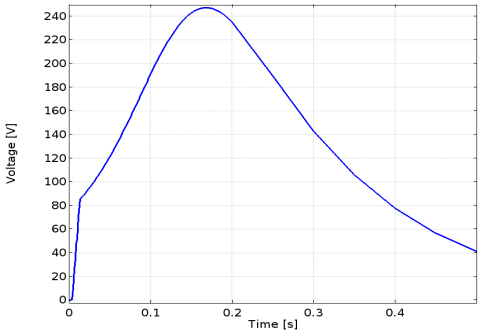

Figure 14: Coil resistive voltage as function of time.

inducing a transition of the conductor to the normal conductive state. At this point, Ohmic losses become predominant in heating up the coil, as shown in Fig. 11. The coil temperature is extracted after 0.5 s and presented in Fig. 12. From the simulation, it is possible to extract the coil resistance (Fig. 13) and voltage (Fig. 14) that can be further used as input for the protection systems.

4 Conclusions

We successfully simulated a quench event in a superconducting magnet, and we extracted the coil resistive voltage that is used as input for protection systems. The COMSOL Java API has proven to be flexible and powerful enough to automatically build the magnet's complex geometry, and characterize it with the necessary set of electro-thermal equations. Thanks to the access provided by COMSOL to the set of governing equations, it has been possible to implement an equivalent magnetization formulation for the eddy-currents. Given the extreme nonlinearities, in physics and material

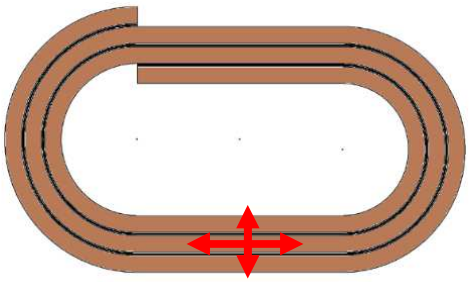

Figure 15: Geometry of a turn with directions of quench propagation.

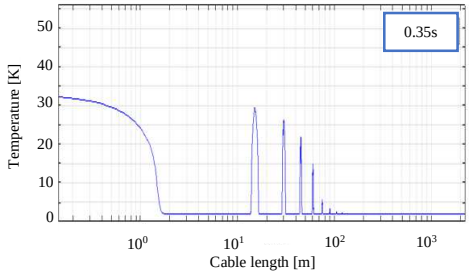

Figure 16: Temperature profile of longitudinal (initial temperature front) and turn to turn (spikes) quench propagation.

properties, the computational time has been kept at a satisfactory level, thanks to a reduced number of mesh nodes and to the use of dynamically linked libraries that give flexibility and performance beyond the capabilities of lookup tables.

5 Future Work

The presented 2-D model provides a good approximation to study electro-thermal transients in superconducting accelerator magnets. Another relevant transient occurring in a superconducting magnet is quench initiation and propagation, both, longitudinally and from turn to turn (See Fig. 15). An accurate prediction of the resistance growth and the resulting voltage is needed to establish an appropriate threshold allowing to detect a quench in time. Simulation of quench propagation requires small temporal and spatial discretization, in the order of 1 millimeter and 1 microsecond, respectively. Preliminary results, shown in Fig. 16, are obtained with an adaptive 1-D model which uses customized boundary conditions to account for turn-to-turn heat propagation. A *Solid Mechanics* module will be added to future models and coupled with the electro-thermal physics modules. Finally we plan to couple the 1-D coil to 3-D domains of bulk material.

Acknowledgements

The authors would like to thank:
- Dr. Sven Friedel and COMSOL Switzerland for continuous support and invaluable suggestions during the development of the project.
- K. Król, and J.C. Garnier from CERN for the help provided during the coding of the Java framework using the COMSOL API.

References

[1] O. Brüning et al., *LHC design report*. European Organization for Nuclear Research, 2004.
[2] M. N. Wilson, *Superconducting magnets*. Clarendon Press Oxford, 1983.
[3] L. Bottura, "A practical fit for the critical surface of Nb-Ti," *IEEE transactions on applied superconductivity*, vol. 10, no. 1, pp. 1054–1057, 2000.
[4] A. P. Verweij, "Electrodynamics of superconducting cables in accelerator magnets," Ph.D. dissertation, Universiteit Twente, 1995.
[5] H. De Gersem and T. Weiland, "Finite-element models for superconductive cables with finite inter-wire resistance," *IEEE transactions on magnetics*, vol. 40, no. 2, pp. 667–670, 2004.
[6] M. Sorbi and V. Marinozzi, "Magnetization heat in superconductors and in eddy-current problems: A classical thermodynamic approach," *IEEE Transactions on Applied Superconductivity*, vol. 26, no. 6, pp. 1–9, 2016.