IOP Conf. Series: Materials Science and Engineering **1301** (2024) 012157 doi:10.1088/1757-899X/1301/1/012157

Investigation and perspectives of using Graph Neural Networks to model complex systems: the simulation of the helium II bayonet heat exchanger in the LHC

R Stoklasa[1], **N Calabrese**[1,2] **and L Serio**[1]

[1] Technology Department, CERN, 1211 Geneva 23, Switzerland

[2] Department of Statistical Sciences, Sapienza University of Rome, Via Ariosto 25, Rome, Italy

E-mail: `luigi.serio@cern.ch`

Abstract. Large cryogenic systems, like those installed at CERN, are complex systems relying on many diverse physical processes and phenomena that are difficult to simulate and monitor in detail. With only a limited number of properties measured and made available for monitoring and control purposes, several processes contributing to the dynamics of the systems are ignored. This lack of information can reduce the accuracy and the capability of a model to track, predict, and anticipate the behavior of the system. Accurate analytical or numerical computer modeling can be developed to simulate the non-linear dynamics of the processes but they are complex, computationally intensive, and cumbersome to test, validate, and implement with different configurations and limited measurements of the hidden properties. In this work, we present our investigation of using Graph Neural Networks (GNN) to build a model of the helium II bayonet heat exchanger operating in the LHC at CERN. We are proposing to use a hybrid machine learning approach, where the parameters of the GNN model are estimated by a combination of supervised learning algorithms trained on experimental data and bounding physics equations and parameters. The GNN model was initially trained on data from the experiments performed on the LHC prototype magnet strings and validated on data extracted during the operation of the LHC machine. We demonstrate the model's accuracy, repeatability, and robustness in various configurations. The model is also well inspectable and explainable, providing the time evolution of all variables. We report on the results and expected applications, which include predictive control, diagnostic, and operator training.

1. Introduction

Simulation of complex dynamic systems is always a very challenging task. Large cryogenic systems, like those installed at the European Organization for Nuclear Research (CERN), involve complex processes relying on non-linear physics and diverse components and phenomena. Due to many technical and physical limitations inside such components, monitoring all aspects and properties of the physical processes happening inside is not feasible. This leads to a situation when only a limited number of "boundary" properties are measured and made available for monitoring and control purposes, while there are a lot of other "internal" variables that remain hidden. To mention some of them, these include liquid flow-related properties (e.g., wetted length, the distribution of wetted perimeter, liquid level, and available pipe cross-section for gaseous flow along the bayonet heat exchanger), gas flow-related properties (e.g., gas velocity, pressure distribution (pressure drop) along the pipe, or indication of conditions for droplet formation, etc.), or even heat-related properties (e.g., dynamic heat produced by a strong magnetic field variations, the heat produced by beam-induced radiation, or heat exchange between magnets through their interconnections of helium bath). These unobserved properties are sometimes fundamental for understanding precisely the current state of the system, and, currently, they are only inferred indirectly thanks to the expertise and intuition of skilled cryogenic operators.

One possibility, how to reveal such internal variables of the system, is to use computer modeling. If the model can capture the internal processes correctly, and if the outputs of the model match actual observations for the "boundary" properties, then the internal variables derived from the model are a close approximation of internal processes inside the real system. This requires a model that is explainable, inspectable, and validated to work well in a wide range of situations.

A deep understanding of complex systems is crucial for their optimal and effective control and for the validation of existing, modified, and completely new designs. Models and simulators are beneficial for predicting the systems' response to given hypothetical conditions and may also be utilized for predictive control, anomaly detection, and early recognition of malfunctions.

Mathematical (analytical) simulators are one of the state-of-the-art approaches used for the simulation of complex systems [1, 2]. Thanks to several efficient solvers and software tools, they can be very precise and relatively fast. On the other hand, they do have some disadvantages: they require almost complete knowledge of all equations ruling the processes and are often difficult to extend to be used for other designs or other use cases.

Machine Learning (ML) is another possible approach to process simulation [3, 4]. ML models can derive insightful information from large volumes of data, working even without an in-depth knowledge of the system or phenomena of interest. Machine Learning models proved to be able to reach the same performance as mathematical models [5], and perform even better in some cases, keeping some benefits like scalability and computational efficiency.

The two approaches mentioned above, i.e., analytical simulators and ML models, form two extremes on a virtual scale describing how much expert knowledge is hard-coded inside such models. On one side, analytical simulators rely entirely on the knowledge encoded in the form of mathematical equations, while the other extreme is represented by "fully blind" ML models (e.g., deep generic models, MLPs, etc.), which have no a priori knowledge about the problem and try to learn everything from a large amount of data.

In this work, we propose to use a hybrid machine learning approach [6], where the model is neither a "fully blind" nor a fully analytical implementation. Our model employs simplified equations and constraints, which helps the model to focus on desired physical relationships and trainable parameters that allow for precise and accurate reproduction of the real system's behavior. This hybrid approach enables the model to reach very high reliability while requiring only a limited number of samples of just the "boundary" variables. This strongly contrasts the fully-blind ML models, which would require a very high number of samples of all the "internal" variables with sufficiently high variability. Our approach resembles the Discrepancy Learning [7, 8] where the authors are using the analytic simulator and are training only the corrective ML part on top of it. In contrast to them, embedding everything inside a single ML model eliminates the need for an analytical simulator with complex physics.

The goal is to build a complex superfluid helium cooling loop model for LHC magnets at CERN. The whole complex system was divided into individual smaller and simpler subsystems, which can be modeled separately, and the sub-models can be pre-trained on separate individual datasets. The final model is obtained by "stacking" sub-models together and then fine-tuning them to match the complex system's observed behavior.

Our solution is based on Graph Neural Networks (GNNs) [9, 5], which are a special kind of neural networks that operate on graphs. GNNs have recently become quite popular in the High-Energy Physics field [10]. They use the concept of *message passing* [9] to exchange and transform information between graph nodes, which is the main principle of how GNN derives computational results. The possibility of encoding a lot of various problems into graph structures provides several benefits in using GNN as the central architecture.

The substantial challenge lies in overcoming the *sparsity* of the problem, i.e., only a limited amount of observations of hidden internal variables are available (limited in time, accuracy, and space). Thus, having reasonable constraints in the model helps to regularize the domain of all possible solutions, helps to reduce the parametric space for the model functions, and allows the derivation of a reasonable and robust model. These constraints come from the graph structure, the semantics of messages for message-passing, and physics-inspired equation constraints.

In this paper, we present our investigation and perspective on using GNNs to build a model of complex superfluid helium cooling loop for LHC magnets at CERN. We demonstrate that the developed model's accuracy falls within the measurement errors despite some current simplifications. The model is also well inspectable and explainable – it provides time evolution of all internal variables, which offers valuable insights to uncover what is happening inside the real system. The model is easily extendable and can be used to simulate various systems – it can be directly used to test and validate new designs and their properties just by creating a graph representation of the desired system. The developed model for the LHC has excellent potential to be used in multiple places, e.g., it can be used as part of the predictive control to validate the performance of the whole LHC ring, and for diagnostics.

In the next section, we explain our use-case problem of simulation the helium II bayonet heat exchanger. Our proposed solution is described in section 3, while section 4 presents conducted experiments and achieved results.

2. Problem definition

The LHC cryogenic system at CERN is a remarkable engineering feat, designed to distribute a substantial amount of cooling power along a 3.3 km-long sector of the LHC machine [11]. The system relies on large cryogenic refrigerators that transfer helium at extremely low temperatures to the superconducting magnet strings, which are the heart of the LHC. These magnet strings operate within a bath of pressurized helium II, where a novel cooling scheme called the bayonet heat exchanger (BHX) is employed [12]. The BHX scheme employs a quasi-isothermal heat sink along the length of the magnet string and is represented schematically in Figure 2.

The superconducting magnets are immersed in pressurized helium II, enabling the conduction of the generated heat load to a linear heat sink made up of a heat exchanger tube. The BHX extracts heat by vaporizing a two-phase flow of saturated helium II, maintaining a constant temperature through pumping and controlling the flow rate via Joule-Thomson expansion valves. This dynamic control of the BHX is essential to ensure the magnets' safe and optimal operation, considering heat load variations while ramping up the magnet current, magnetic field changes, and beam operation.

The cooling scheme underwent extensive studies and testing on dedicated test loops and partial/full-scale prototypes of the magnet string, called *String-1* and *String-2*. The knowledge gained from these experiments facilitated the definition of the control parameters necessary for the safe and efficient operation of the 27-km-long LHC machine at temperatures below 1.9 K over several years. This valuable data has also been utilized for training various models, aiding in this intricate cooling scheme's

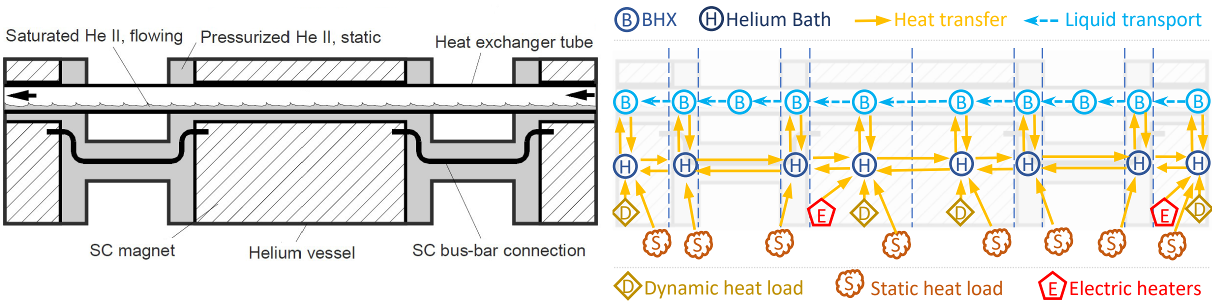

Figure 1. Left: The principle of the LHC superfluid helium cooling scheme (Source: [12]). Right: An example of the graph used to represent the system state, designed for GNN processing.

IOP Conf. Series: Materials Science and Engineering **1301** (2024) 012157 doi:10.1088/1757-899X/1301/1/012157

comprehension and diagnostic capabilities, leading to improved control parameters.

Given the success of Graph Neural Networks (GNNs) in data-driven modeling of complex physics problems, it becomes evident that they offer a suitable approach for modeling complex non-linear systems like the LHC cryogenic system. GNNs are models operating on graph data structures, either simple homogeneous graphs or complex heterogeneous graphs that model various relationships. Heterogeneous graphs come with several disjoint node sets and edge sets, where all the nodes, edges, and the graph itself can have associated multiple features encoding their *state*. The computational procedure is based on the message passing protocol, which exchanges *messages* between neighboring nodes along their connecting edges. These messages are used to update the *state* of edges, nodes, and the graph context.

The LHC cooling system can be decomposed into several sub-systems, which can be modeled separately and are loosely coupled together. Several sub-systems can be easily identified: fluid and gas flow inside the BHX, heat transfer scheme (i.e., the flow of the energy), various heat sources, characteristics of the JT valve, the non-linear properties of the superfluid helium, and possibly others. These sub-systems can be modeled and pre-trained individually and independently (possibly using data from independent experiments), and only later, they may be stacked together to form the complex model of the whole system. This is made possible thanks to the flexibility of heterogeneous graphs and GNNs.

3. Proposed solution

We initially modeled the heat transfer subsystem, including heat sources and cooling by helium evaporation. The liquid flow and fluid dynamics in BHX are emulated, non-linear helium properties are simplified, and the gas flow is ignored in BHX for simplification.

In general, the GNN model $\mathcal{M}$ takes a graph as its input and outputs the same graph with modified attributes (features). To simulate the evolution of the system in time, the input of our model is a suitable graph representation of the *state* of the system $S(t_i)$ at time t_i, and the model computes the new *state* $S(t_{i+1}) = \mathcal{M}(S(t_i))$ at the next time point. This computational step can be repeated as long as desired to produce long-term simulation.

The graph representation of the system's state consists of nodes and edges, where nodes represent individual parts, and edges represent interaction links. The graph encodes both the physical properties of individual parts and links (via associated node and edge *features*), and the structure defining possible interactions between parts. Also, the graph itself can have an associated set of features called *context*, which can be used to encode some global properties of the system.

Modeling the heat transfer subsystem requires two node types: one to represent part of the pressurized helium bath, and another one representing part of BHX that extracts heat by evaporation. Edges between these nodes encode the spatial layout (i.e., how they are adjacent to each other), thus representing the fact they can exchange heat. Each node contains several attributes describing its properties, e.g., the temperature, mass of the helium, dimensions, etc. Based on these properties, the model internally computes some *messages*, containing, e.g., how much energy is transferred via individual edges. Based on these messages, the model updates the properties of nodes (e.g., modifies temperatures or decreases the amount of liquid helium in BHX).

Heat sources are represented by three node types: one for the electrical heaters, one for the static heat load, and another for the dynamic heat loads generated during accelerator operation.

Liquid flow in BHX can be modeled using nodes representing parts of the BHX. Edges between these nodes would encode a spacial layout, i.e., where the remaining liquid helium can flow. These BHX nodes are merged with the BHX nodes in the heat transfer subsystem. Figure 1 depicts graph structure containing nodes for BHX, helium bath, and all three types of heat sources, as well as edges connecting them.

Messages, which drive the updates of node properties, are computed with the help of two suitable constraints: i) constraints of simplified physics and ii) constraints for increased numerical stability. Embedding the former is possible thanks to the fact that well-studied physical relationships rule interactions between different parts of the real system. These constraints lower the dimensionality of the

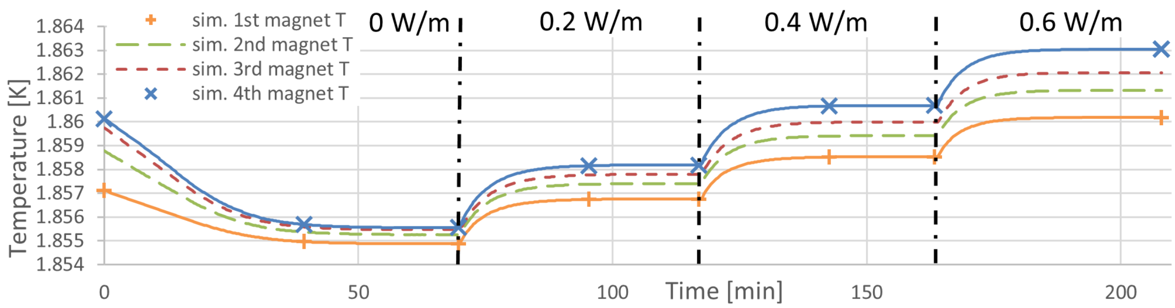

Figure 2. An example of output from our GNN model. The temperature evolution of magnets is extracted over time during 4 phases of an experiment with different applied heat load settings.

modeling problem, making the model more robust and able to learn from less data. The latter constraints are included to fight the problem of discretization and numerical instabilities.

In the computational framework, the following physical constraints are considered: i) Energy Transfer Mechanism: the energy transfer between the helium bath nodes and the heat exchanger nodes is determined by the Kapitza effect, temperature difference, and the wetted surface area inside the BHX. ii) Direction of Heat Flow: heat moves from nodes with higher temperatures to nodes with lower temperatures. iii) Axial Heat Transfer Constraint: temperatures can equalize, but they cannot swap. iv) Maximum Transferable Energy: analytically derived, the maximum transferable energy between two nodes is determined by the temperature difference and the thermal capacities of the two nodes. This constraint prevents temperature oscillations as the simulation time step increases.

The inputs for the model are: the amount of static heat load, the amount of dynamic heat load for each individual heater, the temperature of saturated helium inside BHX (currently a single value for the whole heat exchanger; it will be enhanced once a proper gaseous flow subsystem is implemented), incoming helium mass flow (determining the total cooling power and affecting the emulated liquid state in BHX), and wetted length (which is currently just a static setting; this will be enhanced once we incorporate a proper liquid flow simulation subsystem).

During one model execution, the model iteratively calculates several time steps sequentially, providing the updated system states $S(t_i)$ for each simulated time point. The primary feature within the system state $S(t_i)$ is the temperature of the helium bath nodes. These temperatures in each time point can be retrievable and can be subsequently examined. The model enables the read-out of a graph state at any time step, allowing to track the evolution of any feature of interest.

Figure 2 depicts an example of one simulation run for different heat loads (supplied power). The depicted experiment consists of multiple phases, where each stage has a different applied heat load setting. Simulation of each phase is run until the equilibrium state is reached (i.e., until the time difference of every temperature is smaller than an arbitrary ϵ).

4. Experiments and results

To validate our model, we have simulated the experiments carried out to measure the heat conductivity of BHX over the experimental LHC magnet strings performed before the LHC construction. The experiment consists of several phases: in each stage, the desired value of the applied heat load is set, and then the system is left to find its new equilibrium temperature state. The temperature (or its difference) is noted, and the next step can be executed.

The same experiment protocol was used to derive simulated values, enabling a direct comparison with the results obtained from the real experiments. At the beginning of each phase, the system is set up with initial temperatures, wetted length, helium mass flow, and applied heat loads. Following this, the evolution of the system over time is simulated. Once all the temperatures stabilize, these temperatures

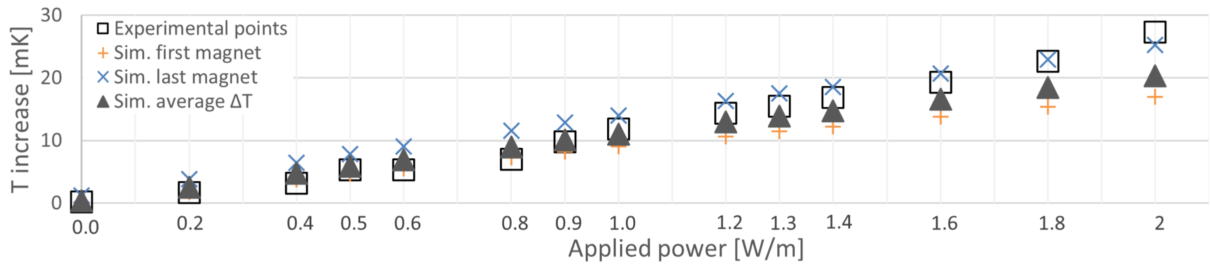

Figure 3. The thermal conductivity experiment of String-1 depicts the differences between the measured and saturation temperatures. The chart shows the experimental data [13] (□) compared with the model simulated average ΔT temperatures (△), together with the ΔT of the first and the last magnet.

are noted, and the simulation moves on to the next phase, i.e., the next applied heat level.

The results of 3 distinct validation steps are presented: i) the thermal conductivity experiment performed on the magnet String-1 [13] composed of 4 prototype magnets; ii) heat exchanger conductivity experiment performed on the magnet String-2 Phase-1 [14] with 5 pre-series magnets; and iii) matching the simulated outputs with the real data from the LHC Standard Cell currently operating in the LHC at CERN, which consists of 8 series magnets. Only the first step (String-1) was used to fit the model's parameters, while the other two demonstrated the robustness of the model – the same model was used to predict the behavior of different configurations and components. From the computational point of view, the only difference between the steps is the input graph, which needs to match the simulated system (e.g., to match the number of magnets and their dimensions, etc.).

String-1 was the first prototype of a string of superconducting magnets cooled by a single BHX. It consisted of three 10 m long dipole magnets and a single 5 m long quadrupole, totaling 35 m in length. The string was inclined with a 1.4% slope and used a corrugated BHX tube. The measurements for thermal conductivity were performed on 14 different settings of applied heat load, varying from 0 to 2 W/m, while ensuring that the entire length of the BHX was wet.

In Fig. 3, the comparison of original results [13] and the simulated ones is presented. As can be seen, the simulated temperatures match very precisely the experimental values up to approx. 1 W/m of applied heat load. For higher heat loads, the predicted values are slightly lower than experimentally measured, which may be caused by some simplifications of the model (these simplifications will be described later). Since this experiment was used to fit the model's parameters, the emphasis was on optimizing the model's accuracy within the range of up to 1 W/m, which is a more realistic range for future designs (String-2 and LHC Standard Cell).

String-2 was the full-sized LHC prototype, which was built in several phases. In Phase 1, when the conductivity experiment was performed, it consisted of 5 magnets: 3 dipoles and 2 quadrupoles. As compared to String-1, both dipole and quadrupole magnets were longer (15 m and 6 m respectively), the string was not inclined, and it used a non-corrugated tube for BHX. Thermal conductivity was estimated

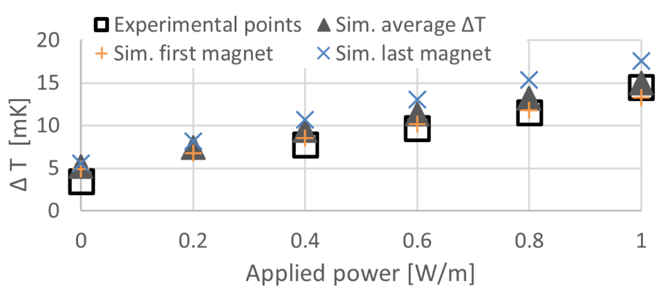

Figure 4. The thermal conductivity experiment of String-2 phase 1 depicts the differences between the measured and saturation temperatures. The chart shows the experimental data [14] (□), compared with the model simulated average ΔT temperatures (△), together with the ΔT of the first and the last magnet.

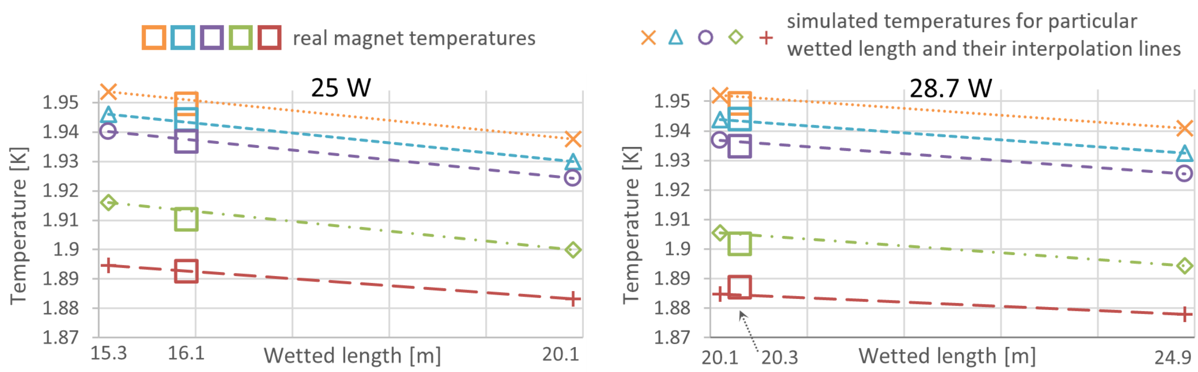

Figure 5. Interpolated steady temperatures of magnets in LHC Standard Cell as predicted by the model, for applied heat loads 25 W (left) and 28.7 W (right) and wetted length of 15.3 m, 20.1 m, and 24.9 m. The predicted wetted length is derived as the position where the real temperatures (□) fit the best the interpolation lines (i.e., 16.1 m and 20.3 m respectively). Please note the chart depicts only 5 out of 8 magnets' temperatures for clarity reasons.

based on five measurements in the range up to 1 W/m of applied heat load, and the BHX was also ensured to be fully wet.

Fig. 4 depicts the comparison of the experimentally measured [14] and our simulated values. Our values match nicely with the experimental ones, except for the part with minimal applied heat load. This discrepancy suggests a potential application of an incorrect assumption regarding the static heat load value. The difference between the first and the last magnet is smaller than measured, which may indicate that our model for very low heat loads over-estimate the cooling capacity of fully-wetted BHX.

The third experiment aimed to replicate the actual data logged from the LHC machine. A specific standard cell (Q15-Q16 of Sector 2) was selected, and two points in time were identified when the cell was supplied with 25 W and 28.7 W of applied power and the cooling system was in a stable condition (these were the periods of the 1.9 K cooling system tests).

The main difference from previous experiments is the lack of information regarding the wetted length. Consequently, the wetted length is inferred using the model through hypothesis testing, posing the question *"What would be the temperature distribution if the wetted length is as specified?"*

Fig. 5 shows the obtained results for applied heat load 25 W and 28.7 W. Due to the discretization procedure, stable temperatures were simulated for hypothetical wetted lengths 15.3 m and 20.1 m, assuming a linear dependency between these two border cases. As can be seen from the chart, the real temperatures observations for 25 W of applied heat load correspond to a wetted length of 16.1 m. Similarly, the estimated wetted length for 28.7 W is 20.25 m (see Fig. 5). The distribution of temperatures along the string is predicted within the sensor's overall absolute accuracy of 5 mK. These derived wetted lengths closely match the expectations of the experts and operators considering the temperature distribution along the magnet string.

4.1. Discussion of Results

For this first investigation, some simplifications and assumptions for the initial model have been adopted. These include static constants for the helium's physical properties, liquid flow emulation with the assumption of constant flow speed, and neglecting the gas flow and pressure drop in the BHX.

In spite of these simplifications, the model successfully captured the dynamics of the system, achieving accurate steady-state values (within the precision limits of the measuring instruments) up to 1 W/m. It demonstrated robustness and applicability across various layouts and configurations of magnet strings.

The model seems to over-estimate magnet temperatures for low heat loads and under-estimate for high

IOP Conf. Series: Materials Science and Engineering **1301** (2024) 012157 doi:10.1088/1757-899X/1301/1/012157

heat loads. This may be explained by the simplification and assumption of the constant liquid flow speed and neglecting the gas pressure drop influencing the temperature profile in the BHX. For higher mass flows, less surface inside BHX will be wetted, higher pressure drop will be experienced, and higher ΔT will be observed. The expectation is that these imperfections will be resolved with the integration of the proper liquid flow simulation component.

5. Conclusions

The developed model is able to show how the system evolves in time for given conditions. Despite some current simplifications, it provides very accurate results inside the tolerance of the actual sensors. The model is well inspectable and explainable, providing time evolution of all internal variables. The model's scalability was demonstrated by simulating various system configurations.

The model can be used directly to test and validate various designs and their properties, for diagnostic purposes to monitor the performance of the whole LHC ring, and also, it can be used as part of advanced predictive control.

We plan to add liquid flow simulation and eliminate current simplifications, e.g., by modeling the proper non-linear characteristics of superfluid helium II. Additionally, a gaseous flow simulation will be added to predict the loop's inverse response phenomenon.

6. References

[1] Bradu B, Blanco Viñuela E and Gayet P 2013 Example of cryogenic process simulation using ecosimpro: Lhc beam screen cooling circuits *Cryogenics* **53** 45–50 URL `https://doi.org/10.1016/j.cryogenics.2012.04.011`

[2] Bradu B 2020 Dynamic simulations and transient management of the hl-lhc cryogenic system during beam operation Tech. Rep. CERN-ACC-NOTE-2020-0060 CERN

[3] Legaard C, Schranz T, Schweiger G, Drgoňa J, Falay B, Gomes C, Iosifidis A, Abkar M and Larsen P 2023 Constructing neural network based models for simulating dynamical systems *ACM Comput. Surv.* **55** ISSN 0360-0300 URL `https://doi.org/10.1145/3567591`

[4] Sanchez-Gonzalez A, Godwin J, Pfaff T, Ying R, Leskovec J and Battaglia P W 2020 Learning to simulate complex physics with graph networks URL `https://arxiv.org/abs/2002.09405`

[5] Sanchez-Gonzalez A 2020 Deep dive on graph networks for learning simulation *4th Inter-experiment Machine Learning Workshop* URL `https://indico.cern.ch/event/852553/contributions/4062226/`

[6] von Rueden L, Mayer S, Sifa R, Bauckhage C and Garcke J 2020 Combining machine learning and simulation to a hybrid modelling approach: Current and future directions *Advances in Intelligent Data Analysis XVIII: 18th International Symposium on Intelligent Data Analysis, IDA 2020, Konstanz, Germany, April 27–29, 2020, Proceedings 18* (Springer) pp 548–560

[7] de Silva B M, Higdon D M, Brunton S L and Kutz J N 2020 Discovery of physics from data: Universal laws and discrepancies *Frontiers in artificial intelligence* **3** 25

[8] Kaheman K, Kaiser E, Strom B, Kutz J N and Brunton S L 2019 Learning discrepancy models from experimental data *arXiv preprint arXiv:1909.08574* URL `https://arxiv.org/abs/1909.08574`

[9] Battaglia P W, Hamrick J B, Bapst V, Sanchez-Gonzalez A, Zambaldi V, Malinowski M, Tacchetti A, Raposo D, Santoro A, Faulkner R, Gulcehre C, Song F, Ballard A, Gilmer J, Dahl G, Vaswani A, Allen K, Nash C, Langston V, Dyer C, Heess N, Wierstra D, Kohli P, Botvinick M, Vinyals O, Li Y and Pascanu R 2018 Relational inductive biases, deep learning, and graph networks URL `https://arxiv.org/abs/1806.01261`

[10] Thais S, Calafiura P, Chachamis G, DeZoort G, Duarte J, Ganguly S, Kagan M, Murnane D, Neubauer M S and Terao K 2022 Graph neural networks in particle physics: Implementations, innovations, and challenges (*Preprint* arXiv: 2203.12852) URL `https://arxiv.org/abs/2203.12852`

[11] Chorowski M, Erdt W, Lebrun P, Riddone G, Serio L, Tavian L, Wagner U and Van Weelderen R 1998 A simplified cryogenic distribution scheme for the large hadron collider *Advances in cryogenic engineering* 395–402

[12] Lebrun P, Serio L, Tavian L and Van Weelderen R 1998 Cooling strings of superconducting devices below 2 K: the helium ii bayonet heat exchanger *Advances in cryogenic engineering* 419–426

[13] Flemsæter B 2000 *Investigation, Modelling and Control of the 1.9 K Cooling Loop for Superconducting Magnets for the Large Hadron Collider* Ph.D. thesis Norwegian U. Sci. Tech.

[14] Blanco E 2003 Superfluid helium cooling loop on string2 *LHC Internal Note, CERN*