# Evaluation of AI-based Methods for Time-Series Modeling of Measurement Data of a Complex Cryogenic System

**Bryan P. Maldonado**[1], **Frank Liu**[2], **Pradeep Ramuhalli**[3]

[1]Buildings and Transportation Science Division, Oak Ridge National Laboratory*
[2]School of Data Science, Old Dominion University
[3]Nuclear Energy and Fuel Cycle Division, Oak Ridge National Laboratory
maldonadopbp@ornl.gov, fliu@odu.edu, ramuhallip@ornl.gov

## Abstract

The Spallation Neutron Source (SNS) at Oak Ridge National Laboratory produces a high-energy neutron beam to be used for over 20 cutting-edge instruments, offering diverse research capabilities across physics, chemistry, biology, and materials science. The Cryogenic Moderator System (CMS) is responsible for cooling the neutron beam for scientific applications by using cryogenic hydrogen ($H_2$) moderators linked to a helium (He) refrigeration loop through heat exchangers. However, the CMS faces challenges from intermittent beam power losses, known as "beam trips," which cause substantial disruptions to its cooling demands. Modeling the dynamic response of the He refrigeration loop after a beam trip is needed for optimizing the control strategy of the CMS for rejecting beam trip disturbances. This paper discusses several AI-based techniques that were used to identify the system dynamics around the most critical component of the CMS system. Using real-world time series, the advantages and disadvantages of such techniques to assist a large engineering system have been identified and discussed.

## Introduction

The Spallation Neutron Source (SNS) at Oak Ridge National Laboratory (ORNL) is the leading facility for neutron science research, using an accelerator-based system to produce pulsed neutron beams through spallation (Mason et al. 2006). These neutrons are delivered to more than 20 instruments, enabling advances in physics, chemistry, biology, and materials science. The Cryogenic Moderator System (CMS) supports the process of cooling neutrons for different instruments using liquid hydrogen ($H_2$) at cryogenic temperatures ($20\,\mathrm{K}$) (Crabtree 2003). Such thermal and hydraulic conditions are maintained using a helium (He) Brayton refrigeration cycle (Takada et al. 2017). Sudden losses of beam power, called beam trips, disturb the CMS and generate large pressure and temperature transients. In particular, the step change in heat load due to a beam trip, from $1.4\,\mathrm{MW}$ to $0\,\mathrm{MW}$ (measured as "power-on-target") within milliseconds, affects the pressure gradient across the turbine in the Brayton cycle. Large transients in the turbine's rotational speed can cause an eventual shutdown of the CMS for corrective actions. The downtime for a CMS system trip, including repair and recovery, is approximately 5 days. Given that the SNS operates 200 days per year, CMS downtime corresponds to 2.5% of lost beam availability per CMS trip. Moreover, with an SNS availability goal of 90%, a CMS trip has a notable adverse impact on that goal. With the upcoming Proton Power Upgrade (PPU) that increases beam power to $2.0\,\mathrm{MW}$ at the first target station by 2028, the CMS faces greater challenges in maintaining operational integrity under larger transient events (Galambos et al. 2020). One option to improve the transient response of the CMS without performing any hardware enhancements is to recalibrate the control algorithms to better reject disturbances produced by large beam trips. (Zhao et al. 2023; Goth et al. 2024; Maldonado et al. 2024b).

To generate a CMS turbine model for control design, this study explores different methods based on artificial intelligence (AI) to extract the dynamic behavior of systems from real-world time series data. The CMS dataset used corresponds to the time series recorded between October 20–21, 2021 (Maldonado et al. 2024a). The training data correspond to a six-hour continuous time series, while the testing data correspond to a 30-minute window following the onset of a beam trip near the end of the dataset. Figure 1 shows a diagram of the turbine with the following instrumentation:

- Upstream turbine pressure: $P_{6519}$
- Upstream turbine temperature: $T_{6519}$
- Downstream turbine pressure: $P_{6522}$
- Turbine speed: $S_{6501}$

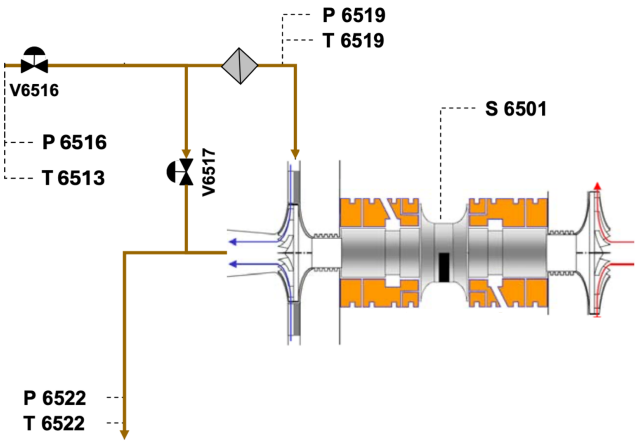


Figure 1: CMS turbine with existing instrumentation

The He flow though the turbine $\dot{m}_T$ can be estimated by analyzing the pressure and temperature drop across the upstream turbine valve $V_{6519}$, which can then be used to further estimate the He flow though the valve $\dot{m}_V$ (Maldonado et al. 2023). Therefore, the following variables around the valve are also considered:

- Upstream valve pressure: $P_{6516}$
- Upstream valve temperature: $T_{6513}$

To model the dynamics of the pressure difference across the turbine, either the upstream or downstream pressure needs to be written as a function of the remaining signals. Scientific Machine Learning methods have demonstrated with great potential to model dynamic behavior of complex systems such as CMS. Besides generic time series modeling methods such as LSTM (Schmidhuber, Hochreiter et al. 1997) and GRU (Cho et al. 2014), more recent approaches have unique advantages. SINDy (Brunton, Proctor, and Kutz 2016) uncovers the governing equations of the dynamic systems from the time series data, by deploying sparse regression method on a library of candidate functions. The outcome is an interpretable ML model. Operator learning (Kovachki, Lanthaler, and Stuart 2024) focuses on approximating mappings between function spaces, often arising in systems governed by partial differential equations or other functional transformations. By leveraging advanced architectures like DeepONet (Lu et al. 2021b) and Fourier Neural Operators (Li et al. 2020), it enables efficient, data-driven modeling of complex, high-dimensional dynamics.

The remainder of this paper is structured as follows. Section *Physics-based model* shows a physics-based model based on thermodynamic principles, which will be used as baseline. Section *SINDy with control* shows the results using sparse identification of nonlinear dynamics (SINDy) and presents an alternative approach for better model fitting. Section *DeepONets* presents the results using a deep operator network (DeepONet). Section *Hankel Matrix* presents the results using the Hankel matrix approach under linear time-invariant assumptions. Finally, *Conclusion* discusses conclusions and future directions for CMS optimization.

## Physics-based model

The physics-based model was developed by applying first-principle physics models of thermal-hydraulic systems to the measurement data of the CMS system. Consider a nonlinear dynamic system for modeling the upstream turbine pressure based on the following thermodynamic principles:

- Compressible turbulent flow through the valve

$$\dot{m}_V = A_V \frac{P_{6516}}{\sqrt{R_{\text{He}} T_{6513}}} \Psi\left(\frac{P_{6519}}{P_{6516}}\right) \tag{1}$$

- Temperature drop through the valve

$$T_{6519} = T_{6513} - \Delta T_V \tag{2}$$

- Pressure drop through the valve

$$P_{6519} = P_{6516} - \Delta P_V \tag{3}$$

- Compressible turbulent flow through the turbine

$$\dot{m}_T = A_T \frac{P_{6519}}{\sqrt{R_{\text{He}} T_{6519}}} \Psi\left(\frac{P_{6522}}{P_{6519}}\right) \tag{4}$$

- Upstream turbine pressure dynamics

$$\frac{d}{dt} P_{6519} = \frac{R_{\text{He}} T_{6519}}{v_T} (\dot{m}_V - \dot{m}_T) \tag{5}$$

Here, $A_V = 1.83\,\text{cm}^2$ is the effective valve area, $A_T = 0.5\,\text{cm}^2$ the effective turbine area, $R_{\text{He}} = 2077\,\text{J/kg K}$ is the ideal gas constant for He, $\Delta T_V = 12.6\,\text{K}$ is the estimated temperature drop through the valve, $\Delta P_V = 0.5\,\text{bar}$ is the estimated pressure drop through the valve, $v_T = 10.8\,\text{cm}^3$ is the volume between the valve and the turbine, and $\Psi(\cdot)$ is the choked flow function defined as:

$$\Psi(\rho) = \begin{cases} \sqrt{\gamma \left(\frac{2}{\gamma+1}\right)^{\frac{\gamma+1}{\gamma-1}}}, & \rho < \left(\frac{2}{\gamma+1}\right)^{\frac{\gamma}{\gamma-1}} \\ \rho^{\frac{1}{\gamma}} \sqrt{\frac{2\gamma}{\gamma-1}\left[1-\rho^{\frac{\gamma-1}{\gamma}}\right]}, & \rho \geq \left(\frac{2}{\gamma+1}\right)^{\frac{\gamma}{\gamma-1}} \end{cases} \tag{6}$$

where $\gamma = 1.66$ is the ratio of specific heats for He. Substituting these relationships in the ordinary differential equation (ODE), we obtain:

$$\frac{d}{dt} P_{6519} = \frac{T_{6519}\sqrt{R_{\text{He}}}}{v_T} \left[ A_V \frac{P_{6519} + \Delta P_V}{\sqrt{T_{6519} + \Delta T_V}} \times \Psi\left(\frac{P_{6519}}{P_{6519} + \Delta P_V}\right) - A_T \frac{P_{6519}}{\sqrt{T_{6519}}} \Psi\left(\frac{P_{6522}}{P_{6519}}\right) \right]. \tag{7}$$

Therefore, the upstream turbine pressure can be modeled by the nonlinear ODE $\frac{d}{dt} P_{6519} = f(P_{6519}, T_{6519}, P_{6522})$, where $x = P_{6519}$ is the state and $u = [T_{6519} \quad P_{6522}]$ is the control. Figure 2 shows the simulated results of the proposed physics based model during a beam trip from $1.14\,\text{MW}$ to $0\,\text{MW}$. Data were collected at different sample rates and synchronized at a samplig rate of $1\,\text{Hz}$. The physics-based model shows good agreement during the heavy transients. However, it is not perfect, and this exploratory study tested different AI techniques to improve the accuracy by using purely a data-driven model.

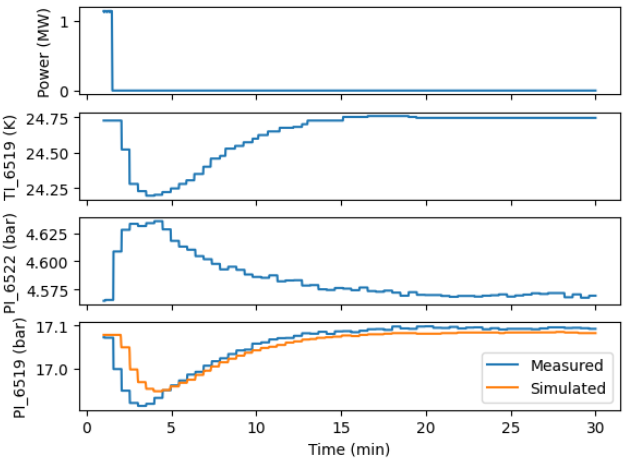


Figure 2: Measured and simulated $P_{6519}$ during a beam tip

## SINDy with control

The sparse identification of nonlinear dynamics (SINDy) algorithm offers a framework for discovering data-driven models that are both interpretable and resistant to overfitting (Fasel et al. 2021; Brunton, Proctor, and Kutz 2016). This is achieved by leveraging sparsity-promoting optimization to extract concise models from limited datasets. SINDy shares similarities with dynamic mode decomposition (DMD), which produces linear models to describe the evolution of key spatio-temporal structures in high-dimensional time series data. Similar to DMD, SINDy has been extended to account for external inputs and control. By integrating these methods, model-based control design can be achieved for nonlinear systems. The SINDy approach, however, is sensitive to the choice of basis functions and sampling rate (Champion, Brunton, and Kutz 2019). Given that the maximum sampling rate for the data is fixed at 1 Hz, special consideration was taken when choosing the basis functions.

The choked flow function for He defined in Eqn. (6) can be simplified to:

$$\widetilde{\Psi}(\rho) = \begin{cases} 0.725, & \rho < 0.5 \\ 1.45\sqrt{\rho - \rho^2}, & \rho \geq 0.5. \end{cases} \tag{8}$$

The identified pressure drop across the valve ($\Delta P_V = 0.5\,\text{bar}$) is small enough to avoid chocking the He flow. Since the turbine is the main component for heat rejection, the pressure drop is significant, causing the He flow to choke. Therefore, the following simplifications can be made:

$$\widetilde{\Psi}\left(\frac{P_{6519}}{P_{6519} + \Delta P_V}\right) = 1.45\frac{\sqrt{P_{6519}\Delta P_V}}{P_{6519} + \Delta P_V} \tag{9}$$

$$\widetilde{\Psi}\left(\frac{P_{6522}}{P_{6519}}\right) = 0.725. \tag{10}$$

Substituting such expressions in Eqn. (7), the simplified

| Parameter | $a_1$ | $a_2$ | $a_3$ | $a_4$ |
|---|---|---|---|---|
| Value | 181,688 | 40,682 | -261 | -160 |

Table 1: Estimated parameters based on physics model

nonlinear ODE can be written as follows:

$$\frac{d}{dt}P_{6519} = 1.45\frac{A_V\sqrt{R_{\text{He}}\Delta P_V}}{v_T}\sqrt{P_{6519}}\frac{T_{6519}}{\sqrt{T_{6519} + \Delta T_V}} - 0.725\frac{A_T\sqrt{R_{\text{He}}}}{v_T}P_{6519}\sqrt{T_{6519}}. \tag{11}$$

Consider the second-order Taylor expansion of the rational expression $\dfrac{T_{6519}}{\sqrt{T_{6519} + \Delta T_V}}$ with respect to $T_{6519}$ around the average He temperature of $T_\mu = 25\,\text{K}$:

$$\left.\frac{T_{6519}}{\sqrt{T_{6519} + \Delta T_V}}\right|_{T_\mu} \approx \frac{3T_\mu^3}{8(T_\mu + \Delta T_V)^{5/2}} + \frac{3T_\mu^2 + 10T_\mu\Delta T_V + 4\Delta T_V^2}{4(T_\mu + \Delta T_V)^{5/2}}T_{6519} - \frac{T_\mu + 4\Delta T_V}{8(T_\mu + \Delta T_V)^{5/2}}T_{6519}^2. \tag{12}$$

Finally, replacing the approximation above in Eqn. (11), we obtain the following nonlinear ODE:

$$\frac{d}{dt}P_{6519} = a_1\sqrt{P_{6519}} + a_2\sqrt{P_{6519}}T_{6519} + a_3\sqrt{P_{6519}}T_{6519}^2 + a_4P_{6519}\sqrt{T_{6519}} \tag{13}$$

where $a_1, a_2, a_3, a_4 \in \mathbb{R}$. Consider the following change of variable:

$$x = \sqrt{P_{6519}}, \quad u = \sqrt{T_{6519}}. \tag{14}$$

Then, the original ODE can be written as a linear combination of polynomial terms as follows:

$$\frac{d}{dt}x = \frac{a_1}{2} + \frac{a_2}{2}u^2 + \frac{a_3}{2}u^4 + \frac{a_4}{2}xu. \tag{15}$$

Table 1 shows the estimated parameters for Eqn. (15) based on the physics-based model. Note that the parameters differ in several orders of magnitude. This is due to the engineering units of the input (Kelvin) compared to the states (Pascal). This issue can be overcome by normalizing the state and control variables. Let $\widehat{u} = \dfrac{u - \mathbb{E}[u]}{\sqrt{\text{Var}[u]}}$ and $\widehat{x} = \dfrac{x - \mathbb{E}[x]}{\sqrt{\text{Var}[x]}}$, then $d\widehat{x}/dt = g(\widehat{x}, \widehat{u}, \theta)$, where $g(\cdot)$ is a polynomial function of order 1 on $\widehat{x}$ and order 4 on $\widehat{u}$ and $\theta$ are the model parameters. The available PySINDy software package was used for implementation (Kaptanoglu et al. 2022). The state derivative $d\widehat{x}/dt$ was estimated by smoothing the measured state $\widehat{x}$ using a third-order Savitzky-Golay filter with a filter window of 120 samples (2 minutes) and then applying a finite difference method. The optimization

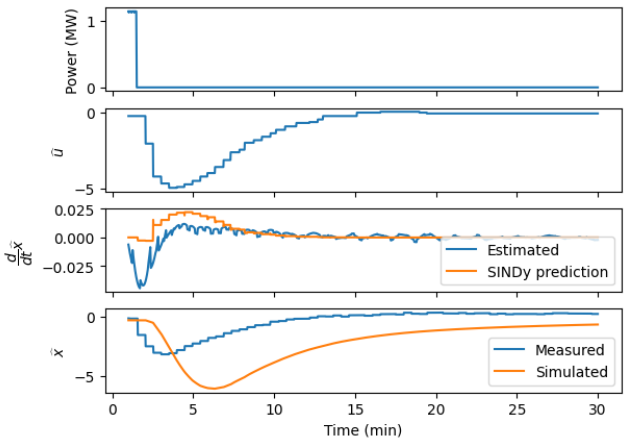


Figure 3: SINDy results using normalized polynomial model

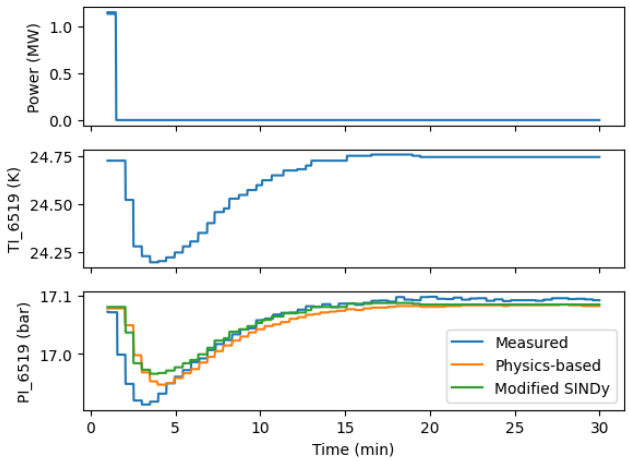


Figure 4: Comparison between measurements and simulation using the physics-based model and the modified SINDy

method chosen was the sequentially thresholded ridge regression (`pysindy.optimizers.STLSQ`), which minimizes the objective function $\|d\widehat{x}/dt - X\theta\|_2^2 + \alpha\|\theta\|_2^2$ where $\alpha = 0.05$ and $X$ is the polynomial library using $\widehat{x}$ and $\widehat{u}$. Figure 3 shows the results using SINDy with an augmented control input for the normalized polynomial ODE. The predicted state derivative using the optimal $\theta$, as well as the resulting state calculated using the `LSODA` SciPy integrator are compared against the estimated state derivative and measured state. Although SINDy can generate results quite fast, they do not perfectly agree with the measurements. This can be caused by the sampling rate as well as the measurement noise. Therefore, an alternative approach is propose to include an ODE solver for better regression performance.

### Modified SINDy approach

Consider the modified cost function:

$$\min_{\theta} \|\widehat{x} - \widetilde{x}\|_2^2 + \alpha\|\theta\|_2^2 \tag{16}$$

$$\text{s.t. } \frac{d}{dt}\widetilde{x} = g(\widetilde{x}, \widehat{u}, \theta) = X\theta \tag{17}$$

$$\widetilde{x} = \texttt{LSODA}\left(t, \frac{d}{dt}\widetilde{x}\right) \tag{18}$$

In this case, the integrator `LSODA` is part of the iteration loop. This approach takes significantly longer than the original SINDy method where no integration is performed. However, Figure 4 shows that including the ODE solver significantly improves the accuracy for real-world signals with low sampling frequency and measurement noise. Based on these findings, approaches where the integrator operator is included in the optimization algorithm for solving the regression problem may be able to better handle issues encountered in real-world time series data.

## DeepONets

The deep operator network (DeepONet) was developed under the principle of approximating any nonlinear continuous operator (Lu et al. 2021a). For this particular application, a DeepONet was used to learn the ODE integrator operator and recover the state from the input signal $\widehat{u}$. The DeepONet architecture is based on the Universal Approximation Theorem for Operators. Consider the problem of finding the antiderivative operator of the following ODE:

$$\frac{d}{dt}\widehat{x}(t) = g(t, \widehat{x}(t), \widehat{u}(t)), t \in \{0, t_1, \ldots, t_n\} \tag{19}$$

with initial condition $x(0) = x_0$. Consider the operator $\mathcal{G} : \widehat{u} \to \widehat{x}$ that maps the trajectory $\{\widehat{u}(0), \widehat{u}(t_1), \ldots, \widehat{u}(t_n)\}$ onto the state space of $\widehat{x}$ as follow:

$$\mathcal{G}(\widehat{u})(t) : \widehat{u}(t) \mapsto \widehat{x}(t) = x_0 + \int_0^t g(\tau, \widehat{x}(\tau), \widehat{u}(\tau))\, d\tau. \tag{20}$$

Then, the DeepONet algorithm consists of a *branch* and *trunk* multilayer perceptrons (MLPs) that map the input sequence $\{\widehat{u}(0), \widehat{u}(t_1), \ldots, \widehat{u}(t_n)\}$ and time domain $\{0, t_1, \ldots, t_n\}$, respectively, to a lower dimensional space $\mathbb{R}^p$ where the universal approximation theorem for integrator operators can be applied:

$$G(\widehat{u})(t) \approx \sum_{k=1}^{p} \text{MLP}_k^{\text{branch}}(\widehat{u}(0), \widehat{u}(t_1), \ldots, \widehat{u}(t_n)) \times \tag{21}$$

$$\text{MLP}_k^{\text{trunk}}(t_0, t_1, \ldots, t_n). \tag{22}$$

Note that this approach is fundamentally different from the physics-based model or the SINDy algorithm. For the previous results, the simulated values of $P_{6519}$ were generated at any time instant $\widehat{x}(t_i)$ using the values of the pressure state at the previous time instant $\widehat{x}(t_{i-1})$ and temperature input at the current time instant $\widehat{u}(t_i)$. Finally, these values are ran through the ODE equation using a numerical integrator. In the DeepONet approach, the input is no longer a scalar value evaluated at a particular time step, but rather and entire sequence of input values. Consequently, the output of the DeepONet is the time series of the estimated state. Although this approach does not need an ODE solver, like `LSODA`, it

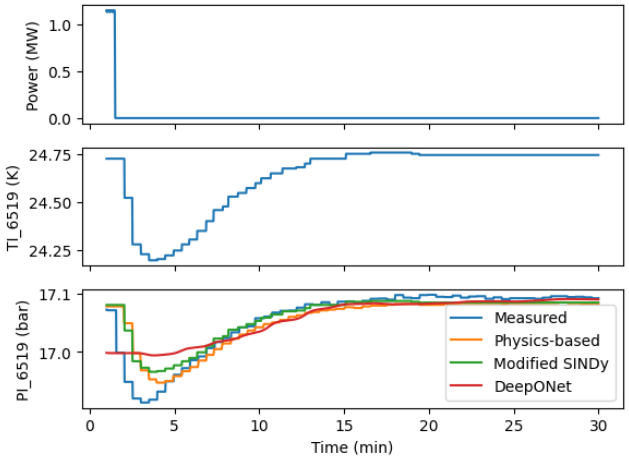


Figure 5: Comparison of DeepONet trajectory prediction versus time series modeling using `LSODA` ODE solver

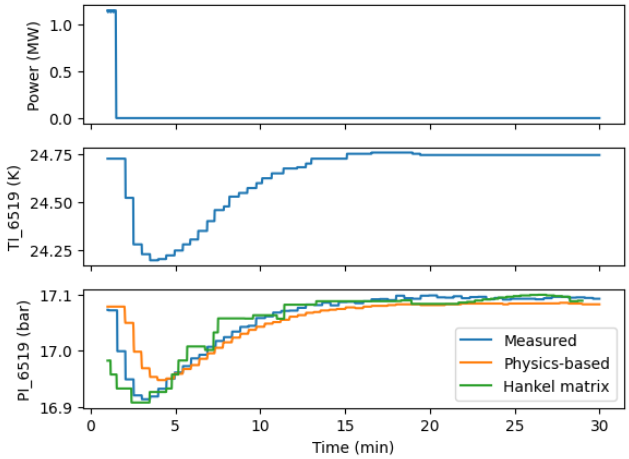


Figure 6: Data-driven time series prediction using the mosaic Hankel matrix under LTI assumptions

requires an entire input sequence to estimate the value of the state at a particular time.

The DeepONet implementation was done using the existing DeepXDE Python package (Lu et al. 2021b). TensorFlow was used for the backend tensor library. The same time series used to train the physics-based mode and the modified SINDy approach were considered for training the DeepONet. The training time series were divided into sequences of 1800 samples (30 minutes). The branch and trunk MLPs were designed as fully connected neural network with a single hidden layer of size 20 and an output layer of size $p = 20$. The activation function chosen was the hyperbolic tangent, and the Adam optimizer was employed with a learning rate of 0.001 for during 15,000 iterations. Figure 5 shows the simulated state using DeepONet compared to the previous methods explored. The results show that DeepONet is able to capture the transients during the beam trip. Note, however, that the change in the state us quite sharp during the beam trip, followed by a smooth recovery to the steady-state value. This approach of mapping input trajectories onto output trajectories can be simplified if the dynamic system in question is linear.

## Hankel Matrix

Modeling time series resulting from dynamic system using data-driven methods has been extensively studied by the control community. One method that has showed promising results is the use of the mosaic Hankel matrix of an input-output time series to predict future trajectories (Markovsky, Huang, and Dörfler 2023). Although this method was designed for linear time-invariant (LTI) systems, the method was applied to the nonlinear CMS equations since the model parameters estimated in Table 1 show a strong preference for linear terms.

Let $\widehat{u}_t^1, \widehat{u}_t^2, \ldots, \widehat{u}_t^N$ be $N$ sequential input trajectory of length $t$. Then, the mosaic Hankel matrix for $\widehat{u}$ of order $T$ and time series length $t$ is defined as:

$$\mathcal{H}_T(\widehat{u}_t^{1:N}) = \begin{bmatrix} \mathcal{H}_T(\widehat{u}_t^1) & \mathcal{H}_T(\widehat{u}_t^2) & \ldots & \mathcal{H}_T(\widehat{u}_t^N) \end{bmatrix} \quad (23)$$

where

$$\mathcal{H}_T(\widehat{u}_t^i) = \begin{bmatrix} \widehat{u}^i(0) & \widehat{u}^i(1) & \ldots & \widehat{u}^i(t-T) \\ \widehat{u}^i(1) & \widehat{u}^i(2) & \ldots & \widehat{u}^i(t-T+1) \\ \vdots & \vdots & \ddots & \vdots \\ \widehat{u}^i(T-1) & \widehat{u}^i(T) & \ldots & \widehat{u}^i(t-1) \end{bmatrix} \quad (24)$$

is the Hankel matrix of each input sequence. Similarly, let $\mathcal{H}_T(\widehat{x}_t^{1:N})$ be the corresponding mosaic Hankel matrix for $\widehat{x}$ of order $T$ and time series length $t$, using the same time intervals as in $\mathcal{H}_T(\widehat{u}_t^{1:N})$. Under observability and persistent excitation conditions on the linear system, for a new input-out time series $\{\widehat{u}_t^*, \widehat{x}_t^*\}$ there exists a linear combination $g$ such that:

$$\begin{bmatrix} \mathcal{H}_T(\widehat{u}_t^{1:N}) \\ \mathcal{H}_T(\widehat{x}_t^{1:N}) \end{bmatrix} g = \begin{bmatrix} \widehat{u}_t^* \\ \widehat{x}_t^* \end{bmatrix} \quad (25)$$

Note that the last value of the state $x^*(t-1)$ can be found at the last row of such vector-matrix equation. Therefore, the linear combination $g$ can be computed from the remaining rows and applied only to the last row for prediction.

Figure 6 shows the results for the testing time series used in previous methods. To simplify and reduce the dimensions of the mosaic Hankel matrix, $t = T = 1800$ was chosen. Even with this dimensioning, the size of the resulting matrix using the training data was $3,600 \times 19,502$, highlighting the memory demand needed for this approach, specially when calculating the pseudo-inverse. The results show that the linear system identified can effectively react to the input temperature sequence, comparable to the results of the nonlinear physics-based model. This indicates that nonlinearities in the system are not significant in this particular application, and that a linear approximation may be enough to generate a successful model simulation and control design and recalibration.

## Conclusions

Two types of data-driven methods where used to identify nonlinear dynamics from real-world, low resolution, noisy data. The first set of methods correspond to those where an ODE integrator is needed for prediction. In this case, the state variable can be predicted using the AI-based model, previous state, current control, and an appropriate ODE integrator. Under this methodology framework, SINDy was tested under real-world conditions. A modified SINDy approach which includes the ODE in the optimization algorithm that determines the model parameters was presented. Results showed that the modified SINDy approach is comparable to the physics-based nonlinear model for the system under investigation. However, the result is contingent on an appropriate choice of a state-input polynomial library.

The second set of AI-based identification methods correspond to operators that map input sequences into state sequences. The first example was the DeepONet method, which approximated the ODE integrator operator. The second method is based on and LTI dynamic system representation and uses a mosaic Hankel matrix to estimate any state-input time series from a finite collection of previously observed state-input time series. Both methods were able to capture the system dynamics during a beam trip, hinting towards the possibility of successfully replicating the dynamic response with a linear system.

The different models listed here were used to simulate the system's state ($\widetilde{x}$), and the following performance metrics were calculated using the measured state values ($x$):

**Root Mean Square Error (RMSE):** Measures the average magnitude of the error between measured and simulated signals, calculated as follows:

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \widetilde{x}_i)^2} \tag{26}$$

**Mean Absolute Error (MAE):** Measures the average absolute difference between measured and simulated values, calculated as follows:

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N}|x_i - \widetilde{x}_i| \tag{27}$$

**Correlation Coefficient:** Assesses the degree of linear correlation between the signals, calculated as follows:

$$R = \frac{\sum_{i=1}^{N}\left(x_i - \mathbb{E}[x]\right)\left(\widetilde{x}_i - \mathbb{E}[\widetilde{x}]\right)}{\sqrt{\sum_{i=1}^{N}\left(x_i - \mathbb{E}[x]\right)^2 \sum_{i=1}^{N}\left(\widetilde{x}_i - \mathbb{E}[\widetilde{x}]\right)^2}} \tag{28}$$

**Nash-Sutcliffe Efficiency (NSE):** Evaluates the predictive power of the model by comparing it to the mean of the measured signal, calculated as follows:

$$\text{NSE} = 1 - \frac{\sum_{i=1}^{N}\left(x_i - \widetilde{x}_i\right)^2}{\sum_{i=1}^{N}\left(x_i - \mathbb{E}[x]\right)^2} \tag{29}$$

Here, $\mathbb{E}[\cdot]$ corresponds to the expected value, $R \in [-1, 1]$ and NSE $\in (-\infty, 1]$. Table 2 shows the performance metrics for each method discussed in the paper. As previously mentioned, the SINDy model did not perform well for this particular application, mainly due to the low sampling frequency and measurement noise in the data. However, the modified SINDy approach performed slightly better than the physics-based model. Although the DeepONet and Hankel matrix approaches are similar, the Hankel matrix performed batter during the particular beam trip transients studied here. This indicates that the subsystem of the CMS consider here has a strong linear component. This characteristic can also be verified looking at the estimated polynomial coefficients in Table 1. In general, the modified SINDY, DeepONet, and Hankel matrix approaches have comparable RMSE, MAE, $R$, and NSE values. From here, we can conclude that the modified SINDY and Hankel matrix approaches performed the best using the data from the complex CMS system.

| | RMSE (kPa) | MAE (kPa) | $R$ | NSE |
|---|---|---|---|---|
| Physics-based | 2.35 | 1.56 | 0.91 | 0.81 |
| SINDy | 14.39 | 12.65 | 0.53 | -6.02 |
| Mod. SINDy | 2.2 | 1.25 | 0.95 | 0.84 |
| DeepONet | 2.36 | 1.54 | 0.91 | 0.81 |
| Hankel Matrix | 2.04 | 1.1 | 0.94 | 0.86 |

Table 2: Performance of data-driven models

Future works involves the testing of such AI methods for learning the dynamics of CMS subsystems where a simple physics-based model is not available, such as the $H_2$ moderators. With a full dynamic model of the CMS, accurate and effective control recalibration can be achieved to improve the reliability and power output of the SNS.

## Acknowledgments

This work was supported by the US Department of Energy (DOE) Office of Advanced Scientific Computing Research's "Data-Driven Decision Control for Complex Systems (DnC2S)" project under FWP ERKJ368.

## References

Brunton, S. L.; Proctor, J. L.; and Kutz, J. N. 2016. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the national academy of sciences*, 113(15): 3932–3937.

Champion, K. P.; Brunton, S. L.; and Kutz, J. N. 2019. Discovery of Nonlinear Multiscale Systems: Sampling Strategies and Embeddings. *SIAM Journal on Applied Dynamical Systems*, 18(1): 312–333.

Cho, K.; van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; and Bengio, Y. 2014. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. In Moschitti, A.; Pang, B.; and Daelemans, W., eds., *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1724–1734. Doha, Qatar: Association for Computational Linguistics.

Crabtree, A. 2003. Hydrogen Moderator System Functional

System Description. Technical report, Oak Ridge National Lab.(ORNL), Oak Ridge, TN (United States).

Fasel, U.; Kaiser, E.; Kutz, J. N.; Brunton, B. W.; and Brunton, S. L. 2021. SINDy with Control: A Tutorial. In *2021 60th IEEE Conference on Decision and Control (CDC)*, 16–21.

Galambos, J. D.; et al. 2020. Final design report proton power upgrade project. Technical report, Oak Ridge National Lab.(ORNL), Oak Ridge, TN (United States).

Goth, N.; Liu, F.; Maldonado, B.; Ramuhalli, P.; Howell, M.; Maekawa, R.; and Cousineau, S. 2024. Dynamic systems modeling of the spallation neutron source cryogenic moderator system to optimize transient control and prepare for power upgrades. *IOP Conference Series: Materials Science and Engineering*, 1301(1): 012088.

Kaptanoglu, A. A.; de Silva, B. M.; Fasel, U.; Kaheman, K.; Goldschmidt, A. J.; Callaham, J.; Delahunt, C. B.; Nicolaou, Z. G.; Champion, K.; Loiseau, J.-C.; Kutz, J. N.; and Brunton, S. L. 2022. PySINDy: A comprehensive Python package for robust sparse system identification. *Journal of Open Source Software*, 7(69): 3994.

Kovachki, N. B.; Lanthaler, S.; and Stuart, A. M. 2024. Operator learning: Algorithms and analysis. *arXiv preprint arXiv:2402.15715*.

Li, Z.; Kovachki, N.; Azizzadenesheli, K.; Liu, B.; Bhattacharya, K.; Stuart, A.; and Anandkumar, A. 2020. Fourier neural operator for parametric partial differential equations. *arXiv preprint arXiv:2010.08895*.

Lu, L.; Jin, P.; Pang, G.; Zhang, Z.; and Karniadakis, G. E. 2021a. Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators. *Nature Machine Intelligence*, 3(3): 218–229.

Lu, L.; Meng, X.; Mao, Z.; and Karniadakis, G. E. 2021b. DeepXDE: A deep learning library for solving differential equations. *SIAM Review*, 63(1): 208–228.

Maldonado, B.; Winder, D.; Ramuhalli, P.; and Blokland, W. 2024a. Process and control variables from the SNS's cryogenic moderator system.

Maldonado, B. P.; Liu, F.; Goth, N.; Ramuhalli, P.; Howell, M.; Maekawa, R.; and Cousineau, S. 2023. Data-Driven Modeling of a High Capacity Cryogenic System for Control Optimization. *IFAC-PapersOnLine*, 56(2): 3986–3993. 22nd IFAC World Congress.

Maldonado, B. P.; Liu, F.; Goth, N.; Ramuhalli, P.; Howell, M.; Maekawa, R.; Degraff, B. D.; and Cousineau, S. 2024b. Transient Optimization of the Cryogenic Moderator System Controller at the Spallation Neutron Source for Improved Performance. In *2024 IEEE Conference on Control Technology and Applications (CCTA)*, 446–451.

Markovsky, I.; Huang, L.; and Dörfler, F. 2023. Data-Driven Control Based on the Behavioral Approach: From Theory to Applications in Power Systems. *IEEE Control Systems Magazine*, 43(5): 28–68.

Mason, T.; et al. 2006. The Spallation Neutron Source in Oak Ridge: A powerful tool for materials research. *Physica B: Condensed Matter*, 385-386: 955–960.

Schmidhuber, J.; Hochreiter, S.; et al. 1997. Long short-term memory. *Neural Comput*, 9(8): 1735–1780.

Takada, H.; et al. 2017. Materials and Life Science Experimental Facility at the Japan Proton Accelerator Research Complex I: Pulsed Spallation Neutron Source. *Quantum Beam Science*, 1(2).

Zhao, X.; Maldonado Puente, B.; Liu, S.; Lim, S.-H.; Gurecky, W.; Lu, D.; Howell, M.; Liu, F.; Williams, W.; and Ramuhalli, P. 2023. Knowledge-Informed Uncertainty-Aware Machine Learning for Time Series Forecasting of Dynamical Engineered Systems. In *13th Nuclear Plant Instrumentation, Control & Human-Machine Interface Technologies (NPIC&HMIT 2023)*.