System Identification of an Aeronautical Application Precooler Heat Exchanger Using Neural Networks in an E-TUNI Configuration

R. Peluzio, P. M. Tasinaffo, and C. Y. Nakashima

***Abstract*—Modeling heat and mass transfer in engineering applications is of great importance for the development of efficient systems, especially in aeronautics, where weight and performance are critical. Conventional modeling of heat exchangers using white-box and regression techniques relies heavily on manual adjustment of parameters for high and low fidelity models, which increases cost and limits the direct use of available data. In this article, we apply a black-box approach and present the Euler Type Universal Numeric Integrator (E-TUNI) framework, a novel method for identifying systems. We then compare the performance of 40 topologies of E-TUNI and NARX in reproducing a reference model of the heat exchanger, showing E-TUNI's better performance than conventional direct use of Neural Networks for time-series prediction.**

Link to graphical and video abstracts, and to code:
https://latamt.ieeer9.org/index.php/transactions/article/view/10812

***Index Terms*—System Identification, Multi-Step Time-Series Prediction, Dynamic Systems, Neural Networks, Precooler Heat Exchanger, Aeronautics, Universal Numerical Integrator.**

I. Introduction

DEVELOPING efficient systems has always been an immense challenge for engineers and researchers, but the availability of experimental data in high quantity and quality, combined with the computational resources to process them, are a major difference of the current era. In this context, Artificial Neural Network (ANN) techniques, such as the one studied in this work, provide increasingly widespread tools for the rapid and precise identification, or modeling, of systems, transforming raw data into useful models for the analysis and development of real systems. This paradigm shift becomes particularly important when evaluated alongside several current trends in aeronautical systems development and engineering in general, where not only are systems becoming more complex but the very language used to represent, develop, and certify these systems is demanding that a greater number of elements be modeled with precision.

In this context, a big problem caused by the current "Model more and better" philosophy is that it is very challenging to develop using "white-box" models alone since they require extensive parameter tuning and inevitable large amounts of work hours. This article approaches this problem by analyzing whether ANNs can be used to create physical system models with good enough performance to be used in system integration scenarios and with less manual adjustment.

One such scenario is the dynamic system studied in this work. The offset strip-fin heat exchanger, as shown in Fig. 1, is conventionally used for the Pre-Cooler function in multiple aircraft, conditioning the temperature of pressurized bleed air from the engine for use by other systems. For the design and optimization of the aircraft's pneumatic system, it is necessary to model the dynamics of each subsystem accurately. While experimental data exists for this purpose, this process has historically been performed through a combination of low-fidelity models [1], [2], based on empirical correlations, and high-fidelity models [3], based on numerical methods such as the finite element method. Both types of models require significant engineering effort to be adjusted and optimized to represent the system dynamics accurately, which is a time-consuming and expensive process.

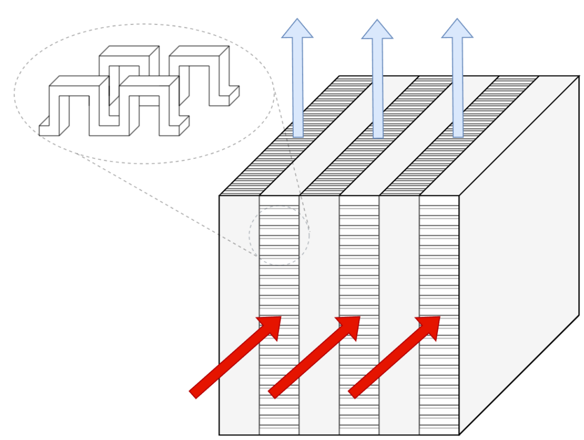


Fig. 1. Schematic of a strip-fin heat exchanger.

The use of ANNs has already been widely demonstrated and has the potential to significantly reduce this effort, allowing the direct use of experimental data to train models that directly represent the dynamics of the system of interest [4], [5].

The associate editor coordinating the review of this manuscript and approving it for publication was Guillermo Valencia-Palomo (*Corresponding author: Rafael Peluzio*).

This work was developed with financial and structural support from the Instituto Tecnológico de Aeronáutica (ITA), Fundação Casimiro Montenegro Filho (FCMF) and Empresa Brasileira de Aeronáutica (Embraer). The authors are grateful for these institutions' financial and structural support in the realization of this work, as well as their continued effort in the development of science and engineering in Brazil.

Rafael Peluzio and Paulo M. Tasinaffo are with the Instituto Tecnológico de Aeronáutica, departments of Aeronautical Engineering and Computer Science, São José dos Campos, SP, Brazil (e-mails: RafaelPeluzio@ita.br and tasinaffo@ita.br).

C. Y. Nakashima is with Embraer S.A., São José dos Campos, SP, Brazil (e-mail: celso.nakashima@embraer.com.br).

As comprehensively detailed in recent studies [6], the literature presents multiple architectures that utilize Feed Forward Neural Network (FFNN), Recursive Neural Network (RNN) and strategies such as Long-Short Term Memory (LSTM), Adaptive Neural-Fuzzy Inference Systems (ANFIS), Physiscs Informed Neural Network (PINN) among others, without a clear indication of a superior method. In this work, the ANNs were generated using the methodology of Universal Numeric Integrators (UNI) [7], which provides a foundation framework for how dynamic systems can be modeled from an ANN connected to a numerical integrator [5] [8].

Given the amplitude of methods for modeling dynamic systems using ANNs, this work argues that there are real gains of using the prediction of derivatives instead of direct prediction for system identification with NNs, based on an extensive simulation performance comparison between these two types of NN implementations for the solution of the Precooler Heat Exchanger problem.

A. Literature Review

The foundation of ANNs was developed throughout the 20th century, starting with the work on the artificial neuron by McCulloch and Pitts and the mathematical definition of its capabilities and methods. The possibility of using an ANN as a universal function approximator [9], [10] was fundamental to its widespread application, including the modeling of dynamic systems as presented by Narendra et al [4], who showed that beyond predicting the future state of a system, it is possible to generate its time series by feedback of the ANN's output. This concept was used in many subsequent works, such as its application in thermal systems modeling [3], [11], [12], general time series [13], pendulum motion [5], pressurized water reactors [14], and the mechanical displacement of a quadrotor vehicle [15], among others.

The UNI concept was derived from the dynamic system identification method. Presented in the 1990s [4], [8], [16], the technique uses the ANN to predict the system derivative instead of the next state, making its behavior easier to learn. The Runge-Kutta Neural Network (RKNN) is implemented to estimate the instantaneous derivative [8], and the E-TUNI (Euler Type Universal Numerical Integrator) implements mean derivatives for the same goal [5]. Some advantages of the UNI framework are that it explicitly uses conventional numerical integration methods in its implementation, which allows this type of ANN implementation for system identification to bridge the gap between black-box models and physical models, since the estimated derivative presents a direct physical interpretation [6], as well as enabling the direct application of advancements in numerical integration to the system identification using NNs [17] [18].

The use of direct prediction in the literature is referred to as NARX (Nonlinear AutoRegressive with eXogenous inputs) [12]–[14], and a characteristic it shares with E-TUNI networks is that the integration/simulation step is mostly limited to its training step, while the RKNN allows a variable integration step intrinsically [8].

The UNI framework is mostly implemented using FFNNs as its approximator, as its external feedback of variables explicitly defines the feedbacks performed. This in turn limits the ability of this type of model to represent systems with time-varying behavior (non-autonomous), in which RNNs are usually employed, such as LSTM [6], [11], [15].

Conventional heat exchanger models are based on physics ("White-Box") and empirical correlations ("Grey-Box") [1], [14] and are adjusted to represent experimental data. In this work, the heat exchanger model by Teruel et al [2] is used as the reference plant for this study and is used to train the E-TUNI and NARX models, which are then evaluated against 34 time-series representative of the operating conditions of the system.

B. Originality

The primary contributions of this work are summarized in the following topics:

- Application of E-TUNI and NARX ANNs to identify the dynamics of a representative model of an aeronautical strip-fin heat exchanger. While ANNs have been applied in several studies for modeling thermal systems, the E-TUNI framework treats the NN as an integrator of mean derivatives.
- Extensive simulation data comparing E-TUNI and NARX performance, showing significant performance gain in E-TUNI compared to the more common NARX configurations.
- Extensive simulation results comparing E-TUNI and NARX performance with 40 different topologies ranging from shallow to deep NNs.
- Sensitivity analysis of E-TUNI and NARX NNs for step variation, showing E-TUNI increased robustness for step sizes different than the one the NN was trained in.

C. Work Structure

This work is subdivided in 5 sections. After the introduction, section II contains the theoretical development of the techniques used, describing FFNN as universal function approximators and the structure of the main implementations of UNIs, with a focus on E-TUNI and NARX, and the classical Heat Exchanger model used to generate the reference data and validate models.

Section III presents the simulation process executed to prepare and generate the datasets, train the ANNs and acquire numeric results. This section also comments on the usage implications of this process in engineering applications of complex systems modeling.

Sections IV and V analyze the performance of the different models trained regarding topologies' width, depth, dataset size and step sensitivity.

II. Theoretical Development

A. Universal Numeric Integrator - UNI

A Universal Numeric Integrator is the union between a Numerical Integrator (e.g., Euler, Runge-Kutta, Adams-Bashforth, among others) and a universal function approximator (e.g., Artificial Neural Networks MLP, RBF, LSTM, Fuzzy

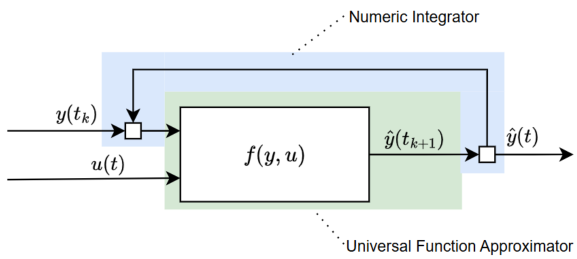


Fig. 2. Visual description of a Universal Numeric Integrator structure.

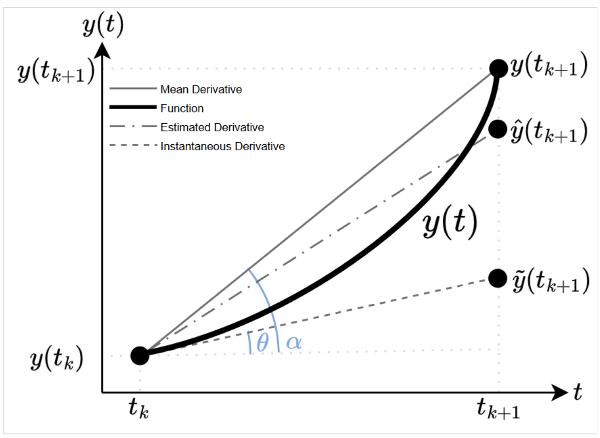


Fig. 3. Difference between instantaneous and mean derivatives.

Inference System, etc.) [7], as presented in Fig. 2. From an engineering perspective, the core concept of this technique is to use the universal function approximator to estimate the system response at a given moment as a function of its current state variables (y) and input variables (u), using a conventional integrator to feed back the calculated system behavior and construct a response in the form of a time series.

In most UNI-related works, the Multi-Layer Perceptron is used as the universal function approximator [7], [19], [20], which is also the case in this work.

Fig. 3 distinguishes between the concepts of mean derivative and instantaneous derivative, which define the main classes of UNIs. In it, a function $y(t)$ varies between two instants t_k and t_{k+1}, which mark a step $\Delta t = t_{k+1} - t_k$ used to discretize the time axis. The instantaneous derivative is defined as

$$\dot{y}(t_k) = \lim_{\Delta t \to 0} \frac{y(t_{k+1}) - y(t_k)}{\Delta t} = \tan(\theta) \tag{1}$$

while the mean derivative is defined as

$$\bar{y}(t_k, \Delta t) = \frac{y(t_{k+1}) - y(t_k)}{\Delta t} = \tan(\alpha) \tag{2}$$

Fig. 3 exaggerates the distance between the predicted results for greater clarity, but in most practical applications, the true future state of the system $y(t_{k+1})$, the state estimated by the specific implementation $\hat{y}(t_{k+1})$, and the state estimated by the instantaneous derivative $\tilde{y}(t_{k+1})$ will be as close as the step size is small and the estimator function is correct.

The basic classification of UNIs refers to the type of derivative estimated by the NN [7], [19]. The three main classes are described as "Direct Prediction," "Instantaneous Derivatives," and "Mean Derivatives," shown in the sub-Figures 1, 2 and 3 in Fig. 4 and detailed as follows:

- Direct Prediction (NARX): The NN directly estimates the next state of the system $\hat{y}(t_{k+1})$. This type of implementation is called NARX due to the type of function used as an approximator and the fact that feeding back the function already constitutes an integration process. Other implementations might also be called NARMAX if they use a Moving Average approach to the time-series feedback.
- Mean Derivatives (E-TUNI): The NN estimates the mean derivative $\hat{\tan}(\alpha)(t_k)$, requiring the use of an Euler-type numerical integration structure to obtain the next state of the system. This type of implementation is called E-TUNI due to its similarity to the Euler method.
- Instantaneous Derivatives (RKNN): The NN estimates the instantaneous derivative $\hat{\tan}(\theta)(t_k)$, which must be integrated similarly to the E-TUNI. An advantage of instantaneous derivative UNIs is that they in theory allow for the use of variable integration steps with no performance degradation. RKNN (Runge-Kutta Neural Network) is an example of an implementation of this type [8].

The three types of UNI mentioned have advantages and disadvantages, and the choice between them depends on the specific problem. Among the three, the direct prediction methodology (NARX/NARMAX) is simpler to implement and can be effective for systems with relatively simple dynamics or when precision is not critical, or if it is a classification problem. However, it has already been demonstrated in works such as [8] that it is simpler for a neural network to learn the dynamics of a system by estimating its variation rather than its absolute value. Of the two methods that use derivative estimation, the mean derivative method (E-TUNI) is simpler to implement, as it uses a single neural network to estimate the mean derivative and an Euler-type integrator to obtain the next state of the system. The instantaneous derivative method, exemplified by RKNN, has a more complex implementation, as it requires multiple neural networks to estimate each of the stages of the Runge-Kutta method. However, the RKNN method can be more accurate and stable for systems where there is a need to simulate with a variable step [8]. Due to its simplicity and robustness, the E-TUNI and NARX methods are used in this work as a benchmark of UNIs ability to reproduce the behavior of training data. The methods are applied exclusively with FFNNs and with a hidden layer architecture containing the same number of neurons at every layer, hereafter referred to as E-TUNIs.

A common feature implemented in NARX, as well as other models, is the input of not only the current value of the time-series $y(t)$, but also multiple past samples of it $y(t-1)$, $y(t-2)$, ()...), $y(t-k)$ and/or a statistic based on these k samples such as the the moving average. This is made to enable the ANN to learn a memory-based behavior. A network

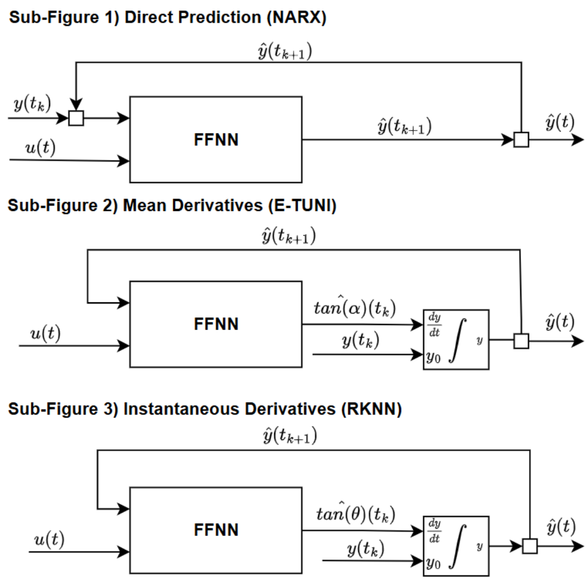


Fig. 4. Functional diagram of direct, mean derivatives and instantaneous derivatives UNIs.

that operates with a large $(t-k)$ window is expected to have a better performance than the one with a smaller window, but at the cost of training cost and complexity.

B. Heat Exchanger Model

Heat exchangers are devices used to efficiently transfer heat between fluids at different temperatures without mixing them. In particular, offset strip-fin heat exchangers are widely used in aircraft due to their high efficiency and low weight. These exchangers consist of thin fins arranged in an offset pattern, which increase the surface area for heat transfer and promotes flow turbulence, enhancing thermal exchange. Fig. 1 illustrates this type of system, showing hot and cold fluids flowing in separate, perpendicular channels, exchanging heat through the fins.

The reference model for this study is a simplified first-order model obtained through the direct application of energy and mass conservation equations, developed in [2]. Its dynamics are defined as

$$\frac{dT_{CM}}{dt} = \frac{1}{m_{CM}c_{p,CM}} \left[\frac{2\dot{m}_H c_{p,H}(T_{H,in}-T_{CM})}{1+2\dot{m}_H c_{p,H} R_{eq,H}} + \frac{2\dot{m}_C c_{p,C}(T_{C,in}-T_{CM})}{1+2\dot{m}_C c_{p,C} R_{eq,C}}\right] \tag{3}$$

This model characterizes the variation of the Temperature at the Core Mass of the exchanger T_{CM} as a function of its previous state and the control variables: Mass Flow Rate and Inlet Temperature of the Hot and Cold fluids ($\dot{m}_H$, $T_{H,in}$, $\dot{m}_C$, $T_{C,in}$). The model also includes the equivalent Thermal Resistance terms for each side of the exchanger ($R_{eq,H}$ and $R_{eq,C}$). These Resistances represent how the physical characteristics of a specific exchanger are incorporated into the generic model. However, obtaining precise values for these Resistances is a complex process involving specific experiments and/or the application of empirical correlations, as they are highly nonlinear values dependent on the specific temperature and flow conditions of the heat exchanger, as well as its geometry and physical characteristics [2], [14].

In (3), the values for the Thermal Resistances appear as constants, but in the simulations performed that implement this model, they are nonlinearly coupled to the other variables. The model for these Resistances was obtained by fitting the heat exchanger model's behavior to 34 experimental tests conducted with the real heat exchanger and the fitted exchanger model is considered as the reference model for this study. The proper adjustment of these Resistances is what provides the model with the ability to reproduce the real system's behavior by approximating complex convection events.

C. Simulation of Experiments

The set of 34 experiments conducted with the real heat exchanger used in this work were made in the context of designing the pneumatic system coupled with the heat exchanger. In these experiments, the inputs (Hot and Cold Flow Rates and Temperatures) were varied to implement the system's entire design space at points of interest. Since the object of this work is to validate the ability and compare the performance of ANNs in predicting time-series, the same 34 experiments are used as references for evaluating the models. Fig. 5 displays an example of how these reference time-series were built, by joining the input conditions of the 34 experiments conducted on the real system, with simulated outputs.

The system's behavior is also described in Fig. 5, where the Hot Inlet Temperature in bright red ($T_{H,in}$) is cooled to the Hot Outlet Temperature in orange ($T_{H,out}$) and the analog occurs for the Cold Inlet Temperature in dark blue ($T_{C,in}$) which is heated to the Cold Outlet Temperature, the light blue curve ($T_{C,out}$). The Heat Exchanger's Core Temperature is presented in black (T_{CM}), and it remains between the Inlet Temperatures of the Hot and Cold fluids. The values of the Hot ($\dot{m}_H$) and Cold ($\dot{m}_C$) Mass Flow Rates are approximately constant in the shown experiment and are presented at the top right of the graph as scalar values of their mean measurements.

It is important to note that Figs. 5 and 7 are presented with Temperature axis non-dimensionalized. This was done by dividing all the absolute temperature measurements by the maximum value measured in the 34 experiments.

III. Numerical and Simulation Procedures

With the scope of effectively evaluating the implementation and performance of the E-TUNI and NARX methods in modeling the complex dynamics of an engineering system of interest, a total of 40 neural network topologies ranging between 30 and 60 neurons per layer and between 1 and 10 hidden layers were trained using two synthetic datasets of 100,000 and 200,000 points, respectively. Their performances were evaluated against the 34 reference time-series, and a mean performance statistic was obtained for each model. All models were implemented using Matlab/Simulink, and the NNs were trained using the Deep Learning Toolbox.

The Neural Network methodology chosen for both the E-TUNI and NARX models was de FFNN. The choice of this method is made for commonality with other UNI related works

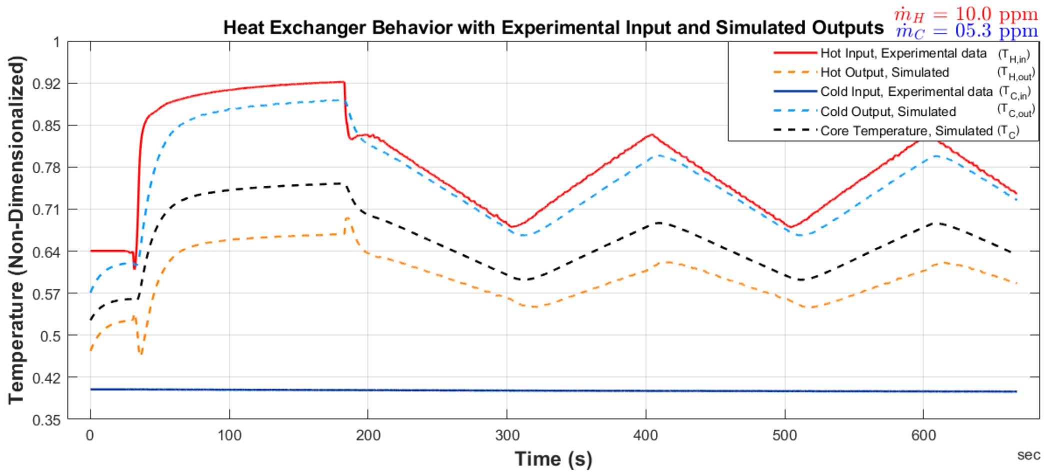


Fig. 5. Simulated Heat Exchanger behavior with experimental inputs.

[5], [20], as well as to provide a benchmark for other methods and a direct comparison between the performance of modeling the derivative of the system and it's output directly. The use of other NN techniques such as LSTM, ANFIS, PINNs, etc is expected to also provide advantageous when in tandem with the UNI framework/.

The algorithms in this work were executed on a NVIDIA GTX 1650 Ti graphics card. The average training time of the ANNs was approximately 1 hour per 5,000 epochs for the largest trained ANNs.

A. Data Preparation

The sets of 100,000 and 200,000 points used for training the models were generated using the classical heat exchanger model presented in Fig. 5. The acquisition of these data followed the process below:

- Lower and upper limits for the four model inputs ($T_{H,in}$, $T_{C,in}$, $\dot{m}_H$, and $\dot{m}_C$) were defined based on the experiments.
- Model excitation signals contained within these limits were generated through the superposition of four sinusoids with semi-random frequencies and amplitudes, normalized to the lower and upper limits of each respective input. This approach was used to sweep the system's design space with continuous and rapidly varying curves, aiming to provide diverse and informative training points.
- These signals were fed into the classical model configured with a fixed step of 5 ms (200 Hz).
- The inputs and outputs were recorded in vectors $\vec{X}$ and $\vec{Y}$, respectively, for training the models. The output vector $\vec{Y}$ was processed in two ways, a "pure" vector $\vec{Y}$ was generated for training NARX NNs and a transformation $\vec{Y}^{\Delta} = (\vec{Y}[N+1] - \vec{Y}[N])/\Delta t$ was performed for training the E-TUNI NNs.

This process was used to generate a synthetic dataset representative of the system's behavior.

In this work, the generated output data was solely used to train the models to predict the Heat Exchanger's Core Temperature, as it is the most representative variable of the system's behavior and directly influences the Hot and Cold Outlet Temperatures. Furthermore, focusing on this variable allows for a simpler analysis of the E-TUNI method's performance.

A second synthetic dataset was generated for retraining the networks due to the need for greater representativeness in the training data regarding the system's behavior under flow conditions very close to zero. For this second set, the process was the same, but with the variation amplitude for the flow rates considerably reduced. To ensure that the network did not become biased toward these low-flow points, the number of samples in this second set was increased by 100,000 points, totaling 200,000, and the values were generated maintaining symmetry with respect to the other variables.

B. Neural Network Training

To conduct the network training, a set of algorithms was implemented to automate the process, which followed these steps:

- The values for W (Width) and D (Depth) within a "for" loop to iterate the 40 topology (T) variations.
- For each T, a neural network was initialized with empty weights and a hyperbolic tangent activation function.
- Training was performed using a GPU with the trainscg (Scaled Conjugate Gradient) training algorithm.
- The stopping criteria are described in Table I and were defined to ensure robust training, preventing overfitting, and guaranteeing model convergence.

Each trained neural network was saved in its own file for subsequent analysis and implementation. The main training parameters are described in Table I. All trained networks used layers of the same width and the same training algorithm to enable comparison between them and analyze the impact of topology on performance.

C. NN Implementation and Performance Analysis

The implementation of the UNIs was performed in the Simulink software according to the Sub-Figs. 1 and 2 of Fig. 4. The numerical integrator structure utilized for the E-TUNI

TABLE I
TRAINING PARAMETERS FOR THE FFNN

Parameter	**Value / Configuration**
Training Algorithm	Trainscg
Performance Function	*Mean Squared Error* with Reg.
Topology (Neurons x No. of layers)	$T = (N, C)$[1]
Data Division (Train/Val/Test)	70%/15%/15%
Division Function	Random (`dividerand`)
Maximum Epochs	$10,000$
Maximum Execution Time	$3,600$ s
Minimum Gradient	1×10^{-10}
Stopping Criterion (Val. Fails)	5

1: $N \in \{30, 40, 50, 60\}, C \in Z, 0 \leq C \leq 10$, where all layers (C) have the same number of neurons (N).

simulation were the conventional integration blocks available on Simulink, since the mean derivative generated by the NN is consistent with the integrator block when using a fixed simulation step. To facilitate the execution of multiple NNs, steps and datasets, the networks were saved in pre-configured Simulink models that, when referenced by name in Simulink, loaded the corresponding NN.

Both the E-TUNI and the NARX models were implemented using a single present measurement of the plant's state to predict it's future state (directly or via derivative).

In total, 160 neural networks were trained, with 40 topologies, 2 synthetic datasets and 2 types of implementations. 34 reference time-series simulations were conducted for each, totaling 5,440 simulations in this study.

Fig. 6 shows the time-series of the core temperature for the same experiment as Fig. 5, but with the predicted results of 4 E-TUNI models with varying performances. The performance of each model is visually represented by how closely its predicted curve follows the experimental curve, with better-performing models showing a closer fit to the reference data.

To quantitatively evaluate and compare the performance of each trained NN topology across all the time-series, the Root Mean Square Error percentage ($RMSE_{\%}$) was computed. As defined in

$$RMSE_{\%} = \frac{100}{\bar{y}} \sqrt{\frac{1}{N_s} \sum_{k=1}^{N_s} \left(\hat{y}(t_k) - y(t_k)\right)^2} \tag{4}$$

, this metric calculates the deviation between the model's estimated state $\hat{y}(t_k)$ and the true state $y(t_k)$ over the total number of samples N_s in each time-series. The result is then normalized by the average value of experimental data ($\bar{y}$) to express the aggregate prediction error as a percentage, allowing for a standardized comparison across time-series with different average values.

Fig. 7 provides an overview of the E-TUNI NNs trained with 100,000 training points, and all the performance results measured in RMSE for the 40 trained topologies across the 34 reference time-series, highlighting the mean as a general indicator of each model's performance. Fig. 8 then uses this concept to plot a comparison of the 40 trained topologies across both E-TUNI and NARX implementations with 100,000 and 200,000 points.

D. Step Variation Performance Analysis

A complementary analysis comparing the E-TUNI and NARX implementations was conducted to evaluate the sensitivity of each method to variations in the simulation step size. For this, the E-TUNI and NARX NNs trained with 200,000 datapoints were simulated for the 34 scenarios again, with step size varying with the gains $\{0.125, 0.250, 0.5, 1, 2, 4, 8, 16, 32, 64, 128, 256\}$ relative to the step size used during training (5 ms). This experiment evaluates the robustness of each method to different simulation step sizes, a desired feature required for high-performance simulations and real-time applications.

The variations in the simulation step size generated variations in the base performance of the models ($RMSE_{\%}$), but in order to effectively compare their performances, another metric was defined, the Relative Performance Variation (RPV), which is calculated as the percentage variation in performance relative to the base performance of each model, as described by

$$RPV_{\%}^{Step} = \frac{RMSE_{\%}^{Step} - RMSE_{\%}^{baseline}}{RMSE_{\%}^{baseline}} \times 100 \tag{5}$$

.

Fig. 9 presents this comparison, showing the RPV for both E-TUNI and NARX implementations across the different step size variations. This plot was made aggregating two statistics: the mean RPV across all 40 topologies for each step size variation and the mean RPV across the 4 best performing topologies for each step size variation. This was made due to the large variation of step x performance behavior of the NARX models that will be explored in the discussion of results.

IV. DISCUSSION OF RESULTS

The main results of this work are presented in Fig. 6, Fig. 7, Fig. 8 and Fig. 9.

For simplification of analysis, the 4 training sessions performed on this work and presented on Fig. 8 will be referred to as follows: E-TUNI trained with 100,000 points (ET1), NARX trained with 100,000 points (NT1), E-TUNI trained with 200,000 points (ET2), and NARX trained with 200,000 points (NT2).

1) Fig. 6 - Performance Levels: The 4 NNs selected for plotting on Fig. 6 were chosen to exemplify the different performance levels that the trained NNs presented on this work achieved. The 4 NNs were selected among ET1 and ET2 networks due to the amplitude of performances for the same reference time-series, but these results are also representative of the NARX models, given that the square-mark curve has a performance similar to the NARX models mean performance. The square-mark curve is the worst-performing model in ET1, the triangle-mark curve is the median-performing model in ET1, the lozenge-mark curve is the best-performing model in ET1, and the circle-mark curve is the worst-performing model in ET2.

The worst-performing model in ET1 is the top curve, with massive errors on the order of 35%. The median-performing curve has errors on the order of 6% and the best-performing

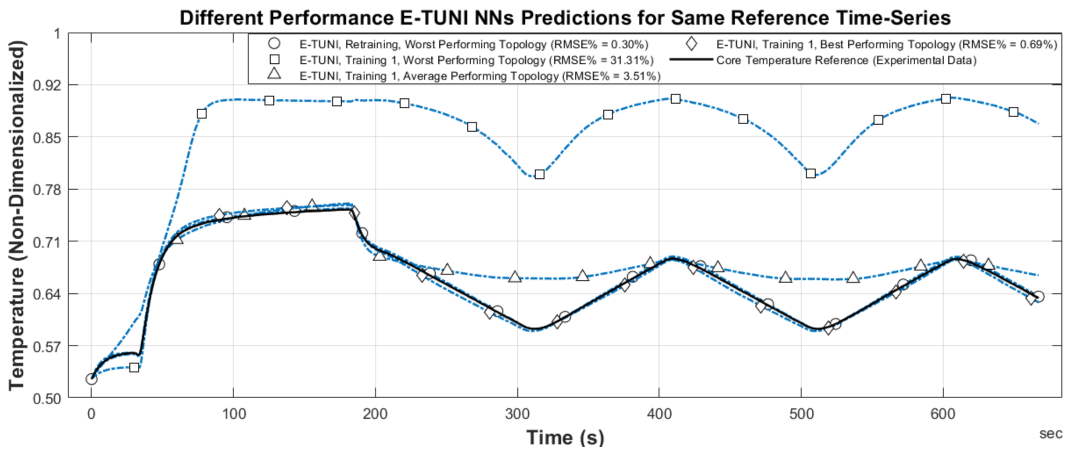


Fig. 6. Time-Series of 4 E-TUNI models with varying performances.

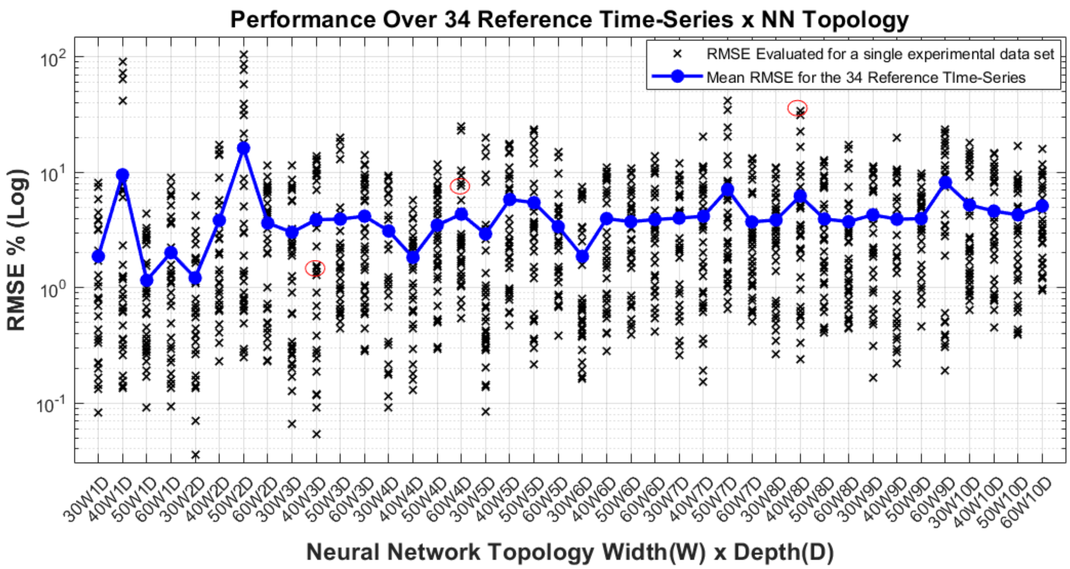


Fig. 7. Neural network topology in relation to model performance in all 34 simulations.

curve in ET1 has errors on the order of 1%, which is similar to the worst-performing model in ET2 with errors on the order of 0.5%. This is consistent with the pattern in Fig. 8, which displays that the performance of the E-TUNI model improved significantly from 4.35% mean performance to 0.41% after the addition of training data.

Performing the same analysis for the NARX models, their mean performance on NT1 was 37.92% and improved to 27.35% after the addition of training data. Even though it was a considerable gain from one session to another, 27.35% is still an unacceptable performance for any practical application. This means that in order to apply the NARX model in this context, the training process would need to be significantly improved or the architecture would need to be revised. The poor performance of the NARX models in this work may be attributed to several factors, such as the complexity of the system being modeled or to the usage of a single sample.

Another result shown by Fig. 6 is the definition of an acceptable performance level for this type of system identification at the order of 1% error, which represents the best-performing model in ET1 and the worst-performing model in ET2. On this case, a performance of around 10% would be considered unacceptable, as it would not be able to represent the system dynamics.

2) Figs. 7 and 8 - Topology Analysis: In Fig. 7 it is possible to observe a trend of deteriorating mean model performance with increased depth. This may indicate that the function to be learned is relatively simple and does not benefit from greater depth, or that the employed training process was unable to exploit the potential of larger networks. However, although the best performances occurred with the smaller networks, the worst performances also occurred with smaller networks, which may indicate the greater sensitivity of smaller networks to the training process. Complementarily, the performances of the larger networks appear to exhibit less dispersion and possibly better stability during training.

For Fig. 8, the addition of new data in ET2 and NT2 was able to significantly improve the average performance, reducing the mean RMSE from 4% to 0.5% and 37.92% to 27.35%. This demonstrates the importance of the quality

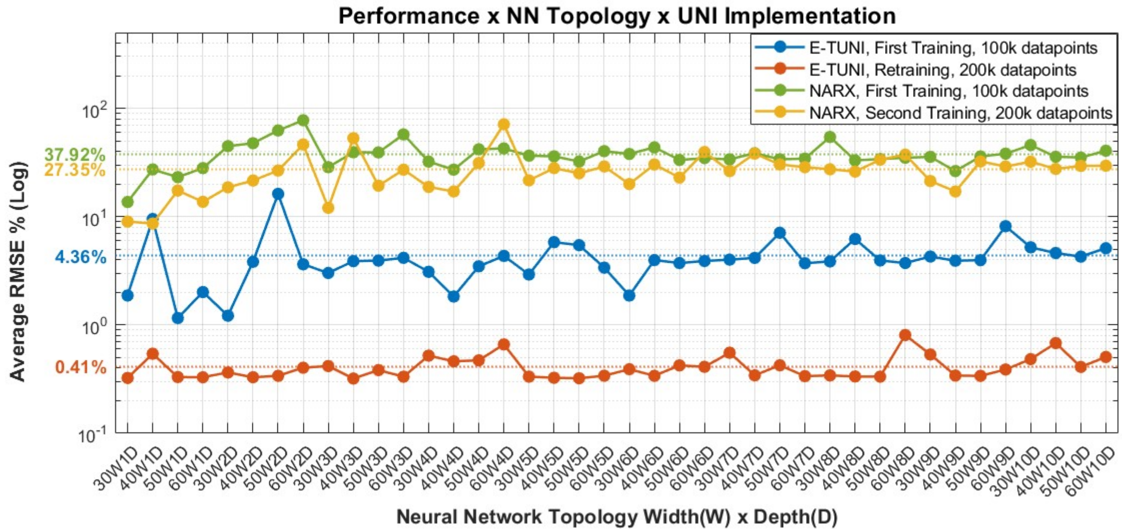


Fig. 8. Comparison between the performance of the first and second training rounds.

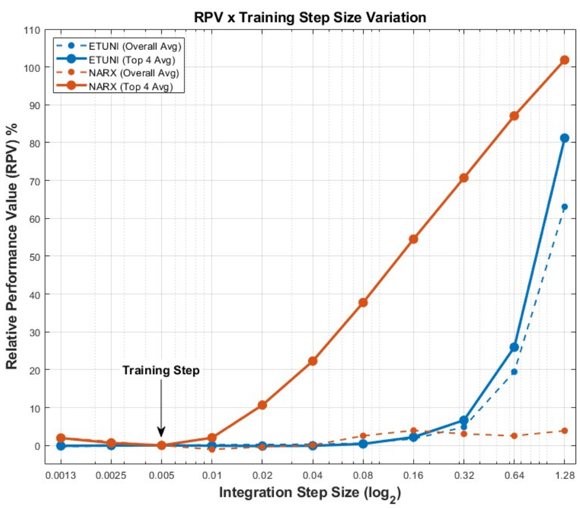


Fig. 9. Relative Performance Variation (RPV) of E-TUNI and NARX implementations for different simulation step sizes.

and quantity of training data for the performance of NN models and suggests that adding more data or utilizing data augmentation techniques could lead to further improvements.

Finally, the main result of Fig. 8 is the significant gap between E-TUNI and NARX models, with the best E-TUNI models achieving an average performance of 0.41% and the best NARX models achieving an average performance of 27.35% when trained with the same data and methods. This strongly indicates the superiority of the E-TUNI method for modeling the dynamics of a precooler heat exchanger system, and suggests that it may be a more effective approach for similar system identification problems compared to the more traditional NARX method.

The optimal topologies identified in this study for each architecture were the 40W3D for the E-TUNI and the 40W1D for the NARX, both trained with 200,000 points.

3) Fig. 9 - Step Variation Analysis: The results of the step variation analysis presented in Fig. 9 show four curves for RPV and step variation. Curves 1,2 and 3 present an expected V-shaped behavior, however, for NARX average performance the RPV curve presented an unexpected behavior of "moving sideways" even improving performance as the step changed. This is attributed to an overall ill-training of most of the NARX models in the training conditions of this work, which may have caused spurious behaviors to have better performances. For this reason, only the top 4 performers of each method will be analyzed in further detail.

For the step variation analysis that resulted in Fig. 9, it is possible to observe that E-TUNI models exhibited a consistent and robust performance even with step sizes significantly different from the one they were trained with, with a mean RPV of around 0% for all step variations until 128 times the training step, while the NARX models exhibited a direct relationship between the step size variation and the performance degradation.

Even though the NARX models exhibited a significant performance degradation with step size variations, both models presented robustness to step variations within applicable levels and the main result is that the sensitivity of NN system identification to training step might be less than expected by theory, which is a positive result for their application in engineering problems. A counter point to this result is that the heat exchanger model used is sufficiently simple and stable, which may have contributed to the robustness of the models to step variations, and that more complex systems may present a greater sensitivity to step variations.

V. Conclusion

This work presented the application of the E-TUNI and NARX ANNs methods to model the dynamics of a complex system such as the strip-fin heat exchanger used in aircraft without the need for parameter tuning. The results demonstrated that the methods are simple and effective for this type of system identification, showing great promise for practical applications.

The E-TUNI method represents the system dynamics with an error metric on the order of 1% relative to the reference data, which far exceeds the performance of the NARX method with the error on the order of 27%. Furthermore, it was observed that small networks, on the order of 100 total neurons, managed to perform better than larger networks in this training configuration. This reinforces the benefit of tailoring the size of the network to be trained to the complexity of the modeled problem.

Both types of NN models exhibited robustness to variations in the simulation step size, with E-TUNI models showing a mean performance variation of around 0.1% even when applied with step sizes significantly different from the training step, while NARX models exhibited a direct relationship between step size variation and performance degradation.

During the development of this work, a few research fronts have been identified as opportunities. It would be interesting to explore the direct application of real data in the training of such systems and to verify if the results presented here are maintained when the models are trained with more complex and higher-order models, as well as to explore the application of these models in real-time control systems, and to compare the generalization of E-TUNI and NARX methods when applied to data ranges outside of their training.

Future research should also investigate the application of this methodology to different physical systems and explore other UNI formulations such as the using a larger input sampling window and RNNs like LSTM for the function approximator of UNIs, improving their capabilities of representing more complex systems.

Finally, as a practical recommendation, researchers and engineers performing system identification using ANNs on lower-order systems are recommended to initially explore wide and shallow topologies, as they may present better performance and faster training times than deeper and larger models according to the results of this work.

REFERENCES

[1] Rahul VR and Ajith Kumar R, "Parametric study of heat transfer and pressure drop characteristics of a rectangular offset strip fin compact heat exchanger," vol. 71, pp. 1381–1386, 2018. [Online]. Available: 10.3303/CET1871231

[2] M. H. Teruel, C. Y. Nakashima, and P. Paglione, "Rectangular Offset Strip-Fin Heat Exchanger Lumped Parameters Dynamic Model ," 2009. [Online]. Available: Unavailable

[3] S. Cai, Z. Wang, S. Wang, P. Perdikaris, and G. E. Karniadakis, "Physics-Informed Neural Networks for Heat Transfer Problems," vol. 143, no. 6, p. 060801, 2021. [Online]. Available: 10.1115/1.4050542

[4] K. Narendra and K. Parthasarathy, "Identification and control of dynamical systems using neural networks," vol. 1, no. 1, pp. 4–27, 1990. [Online]. Available: 10.1109/72.80202

[5] P. M. Tasinaffo, G. S. Gonçalves, J. C. Marques, L. A. V. Dias, and A. M. Da Cunha, "The Euler-Type Universal Numerical Integrator (E-TUNI) with Backward Integration," vol. 18, no. 3, p. 153, 2025. [Online]. Available: 10.3390/a18030153

[6] G. Pillonetto, A. Aravkin, D. Gedon, L. Ljung, A. H. Ribeiro, and T. B. Schön, "Deep networks for system identification: A survey," vol. 171, p. 111907. [Online]. Available: 10.1016/j.automatica.2024.111907

[7] P. M. Tasinaffo, G. S. Goncalves, A. M. Cunha, and L. A. V. Dias, "An introduction to universal numerical integrators," *International Journal of Innovative Computing, Information and Control*, vol. 15, no. 1, pp. 383–406, 2019. [Online]. Available: 10.24507/ijicic.15.01.383

[8] Yi-Jen Wang and Chin-Teng Lin, "Runge-Kutta neural network for identification of dynamical systems in high accuracy," vol. 9, no. 2, pp. 294–307, 1998. [Online]. Available: 10.1109/72.661124

[9] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," vol. 2, no. 5, pp. 359–366, 1989. [Online]. Available: 10.1016/0893-6080(89)90020-8

[10] G. Cybenko, "Approximation by superpositions of a sigmoidal function," vol. 2, no. 4, pp. 303–314, 1989. [Online]. Available: 10.1007/BF02551274

[11] A. Parlos, K. Chong, and A. Atiya, "Application of the recurrent multilayer perceptron in modeling complex process dynamics," vol. 5, no. 2, pp. 255–266, 1994. [Online]. Available: 10.1109/72.279189

[12] M. Mohanraj, S. Jayaraj, and C. Muraleedharan, "Applications of artificial neural networks for thermal analysis of heat exchangers – A review," vol. 90, pp. 150–172, 2015. [Online]. Available: 10.1016/j.ijthermalsci.2014.11.030

[13] R. Katarya and S. Rastogi, "A study on neural networks approach to time-series analysis," pp. 116–119, 2018. [Online]. Available: 10.1109/ICISC.2018.8399024

[14] A. Naimi, J. Deng, A. Abdulrahman, V. Vajpayee, V. Becerra, and N. Bausch, "Dynamic Neural Network-based System Identification of a Pressurized Water Reactor," pp. 100–104, 2020. [Online]. Available: 10.1109/ICCMA51325.2020.9301483

[15] N. Mohajerin and S. L. Waslander, "Multistep Prediction of Dynamic Systems With Recurrent Neural Networks," vol. 30, no. 11, pp. 3370–3383, 2019. [Online]. Available: 10.1109/TNNLS.2019.2891257

[16] A. Rios Neto, "Dynamic Systems Numerical Integrators in Neural Control Schemes," pp. 85–88, 2001. [Online]. Available: 10.21528/CBRN2001-019

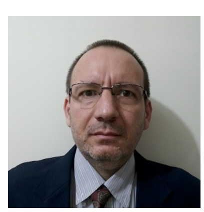
[17] Z. Bin, "Comparison of different order adams-bashforth methods in an atmospheric general circulation model," *Acta Meteorologica Sinica*, vol. 25, no. 6, pp. 754–764, 2011. [Online]. Available: 10.1007/s13351-011-0606-6

[18] E. Misirli and Y. Gurefe, "Multiplicative adams-bashforth-moulton methods," *Numerical Algorithms*, vol. 57, no. 4, pp. 425–439, 2011. [Online]. Available: 10.1007/s11075-010-9437-2

[19] P. M. Tasinaffo, L. A. V. Dias, and A. M. da Cunha, "A qualitative approach to universal numerical integrators (unis) with computational application," *Human-Centric Intelligent Systems*, vol. 4, pp. 1–28, 2024. [Online]. Available: 10.1007/s44230-024-00087-x

[20] P. M. Tasinaffo, G. S. Gonçalves, J. C. Marques, L. A. V. Dias, and A. M. Cunha, "An overview of the euler-type universal numerical integrator (e-tuni): Applications in non-linear dynamics and predictive control," *Algorithms JCR*, vol. 18, pp. 1–28, 2025.

Rafael Machado Peluzio received his Bachelor's degree in Electrical Engineering at the Universidade Federal de Viçosa (UFV) in 2023. He is currently pursuing a Master's degree in Aeronautical Engineering at the Instituto Tecnológico de Aeronáutica (ITA) and is a systems development engineer at Embraer S.A. His research interests include multiphysics simulation, system identification, control systems, numerical integration of dynamic systems, and machine learning.



Paulo Marcelo Tasinaffo received his PhD degree in Space Engineering at the Instituto Nacioanal de Pesquisas Espaciais (INPE) in 2003. Received his MS degree in Mechanical Engineering in 1998 and his Bachelor's degree in Mechanical Engineering in 1996 at Universidade Federal de Itajubá (UNIFEI). He is a Professor at the Instituto Tecnológico de Aeronáutica (ITA). His core expertise encompasses aerospace and computer engineering, focusing deeply on artificial intelligence, nonlinear dynamic system modeling, and artificial neural networks (including both neural control systems and machine learning applications).

Celso Yukio Nakashima received his PhD degree in Mechanical Engineering at the Universidade de São Paulo (USP) in 2005. Received his MS degree in Mechanical Engineering at USP in 2000. Received his Bachelor's degree in Mechanical Engineering at USP in 1997. He is a systems development engineer at Embraer S.A. His research interests include thermodynamics, heat exchanger design, thermal stratification models and dynamic system modeling.