

Machine Learning Methods for the Design and Operation of Liquid Rocket Engines – Research Activities at the DLR Institute of Space Propulsion

Günther Waxenegger-Wilfing^(1,3), Kai Dresia⁽¹⁾, Jan Deeken⁽¹⁾ and Michael Oschwald^(1,2)

⁽¹⁾ German Aerospace Center (DLR), Institute of Space Propulsion, 74239 Lampoldshausen, Germany

⁽²⁾ RWTH Aachen University, Institute of Jet Propulsion and Turbomachinery, 52062 Aachen, Germany

⁽³⁾ Corresponding author: guenther.waxenegger@dlr.de

KEYWORDS: machine learning, surrogate modeling, engine control, neural networks, liquid rocket engines

ABSTRACT:

The last years have witnessed an enormous interest in the use of artificial intelligence methods, especially machine learning algorithms. This also has a major impact on aerospace engineering in general, and the design and operation of liquid rocket engines in particular, and research in this area is growing rapidly. The paper describes current machine learning applications at the DLR Institute of Space Propulsion. Not only applications in the field of modeling are presented, but also convincing results that prove the capabilities of machine learning methods for control and condition monitoring are described in detail. Furthermore, the advantages and disadvantages of the presented methods as well as current and future research directions are discussed.

1. INTRODUCTION

Machine learning (ML) methods use algorithms that can learn from data and make data-driven predictions as well as decisions [1]. Techniques based on learning data representations and especially neural networks (NNs) are achieving outstanding results in recent years [2]. Reasons for the major steps forward are not only theoretical advances but also the availability of a large amount of data and improvements in computer hardware. In addition to well-known applications, e.g. in the fields of computer vision and natural language processing, there are promising applications related to engineering disciplines [3].

Recent research and development activities at the DLR Institute of Space Propulsion prove the feasibility of such methods for supporting the design and operation of liquid rocket engines. So far, NNs are used for the prediction of heat transfer in rocket engine cooling channels and fatigue life estimation. Other applications include the automatic discovery of suitable precursors to combustion instabilities and optimal control of the engines.

2. MACHINE LEARNING BASICS

The field of ML studies algorithms that use datasets to change parts of a mathematical model in order to solve a certain task, instead of using fixed pre-defined rules [1, 2]. The mathematical model is often a function, which maps input data to output data, and the task of the algorithm is to change the adjustable parameters in such a way that the mapping has the desired properties. The sample data used by the ML algorithm to modify the mathematical model or a function is commonly called training data. The central challenge in ML is that the model must perform well on new, previously unseen input data. The field can broadly be divided into three categories, depending on the information available to the algorithm.

2.1 Supervised Learning

In supervised learning, the training dataset contains both the inputs and the desired outputs, and the goal is to learn the corresponding mapping rule. The mathematical model can amongst other things be used for classification or regression. In a classification task, the model is asked to identify to which set of categories a specific input belongs. In a regression task, e.g. with a single explanatory variable, the goal is to predict a numerical value given some input.

2.2 Unsupervised Learning

Unsupervised learning algorithms receive training datasets without target outputs and the goal is to discover the structure or hidden patterns in its input. Unsupervised algorithms can e.g. be used for cluster analysis which groups, or segments, datasets with shared attributes. This approach can help to detect anomalous data points that do not fit into either group.

2.3 Reinforcement Learning

In the reinforcement learning (RL) setting, training data is generated through interaction with a usually dynamic environment [4]. The goal is to optimize the actions in order to maximize the notion of cumulative reward. RL algorithms have achieved impressive results,

e.g reaching super-human performance in games like chess or Go. Besides the sensational results in board games or video games, those algorithms are successfully used in solving complex control problems [5].

2.4 Neural Networks

NNs are a successful family of mathematical models used for ML. NNs are inspired by the functionality of biological brains, which are made of a huge number of biological neurons that work together to control the behavior of animals and humans. A collection of connected units, called artificial neurons, form the basis of an NN. Furthermore, artificial neurons loosely model biological neurons and are usually represented by non-linear functions acting on the weighted sum of its input signals. NNs can represent any smooth function arbitrarily well given enough parameters. Using multiple hidden layers of artificial neurons adds exponentially more expressive power. Each layer can be used to extract increasingly abstract features and hence more suitable representations of the input data. An NN with more than one hidden layer is called a deep NN and the associated learning algorithms are referred to as deep learning algorithms. Deep reinforcement learning is a subfield of ML that combines deep learning and RL.

2.5 Support Vector Machines

Support vector machines are ML models with associated (supervised) learning algorithms [1, 6]. Given a set of training examples, each marked as belonging to one of two categories, a support vector machine maps the inputs to points in space to maximize the width of the gap between the two categories. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall. Support vector machines are one of the most robust prediction methods and are particularly suitable when the features to be used for classification are known.

3. PAST ACTIVITIES

The following is a brief overview of research activities related to ML methods that have taken place at the DLR Institute of Space Propulsion in recent years.

3.1 Modeling

Many engineering problems need accurate models and simulations to evaluate, amongst other things, the implications of design variables and constraints. Furthermore, the computational cost should be moderate to enable optimization loops or even real-time applications. ML models like NNs represent a convincing way to fulfill both criteria. The main disadvantage of

high-fidelity computational fluid dynamics (CFD) or finite element method (FEM) calculations is that they are not suitable for design space exploration and extensive sensitivity analysis due to their large calculation effort [3]. By constructing surrogate models using samples of the computationally expensive calculation, one can alleviate this burden. However, it is crucial that the surrogate model mimics the behavior of the simulation model as closely as possible and generalizes well to unsampled locations while being computationally cheap to evaluate. NNs have been successfully applied as surrogate models in several domains.

3.1.1 NN-based Surrogate Model for the Maximum Wall Temperature

Several methods exist to study the regenerative cooling of liquid rocket engines. A simple approach is to use semi-empirical one-dimensional correlations to estimate the local heat transfer coefficient. However, one-dimensional relations are not able to capture all relevant effects that occur in asymmetrically heated channels like thermal stratification or the influence of turbulence and wall roughness. Especially when using methane as the coolant, the prediction is challenging and simple correlations are not sufficient [7, 8].

An accurate NN-based surrogate model for the maximum wall temperature along the cooling channel is developed by Waxenegger-Wilfing et al. [9]. The training dataset uses results extracted from samples of CFD simulations. The NN employs a fully connected, feed-forward architecture with 4 hidden layers and 408 neurons per layer. It is trained using data from approximately 20 000 CFD simulations. By combining the NN with further reduced-order models that calculate the stream-wise development of the coolant pressure and enthalpy, predictions with a precision similar to full CFD calculations are possible. The prediction of an entire channel segment takes only 0.6 s, which is at least 1000 times faster than comparable three-dimensional CFD simulations.

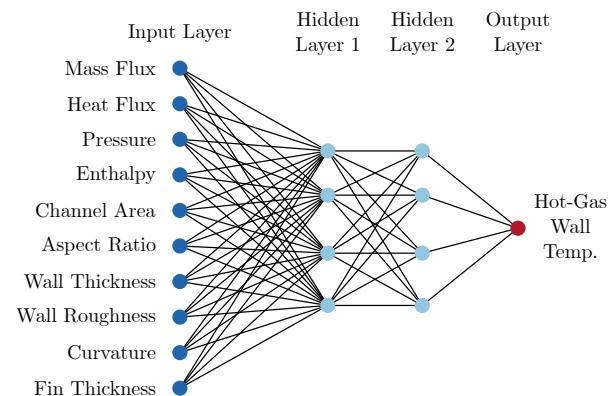


Figure 1: Exemplary NN architecture.

The model is extended for different channel curvatures and rib thicknesses and used to study the cooling channel performance of the LUMEN engine [10]. Fig. 1 shows an exemplary architecture with two hidden layers, four neurons per hidden layer, and all input parameters. Fig. 2 illustrates the result for a chamber pressure of 35 bar and a fixed cooling channel outlet pressure.

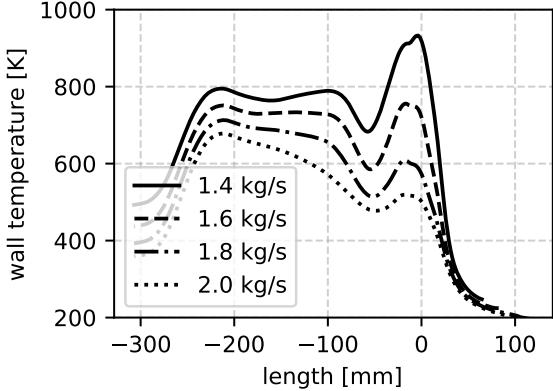


Figure 2: Wall temperature for a combustion chamber pressure of 35 bar.

3.1.2 NN-based Surrogate Model for the Fatigue Life Estimation

Fatigue life prediction is an essential part of the design process of reusable rocket thrust chambers [11]. State of the art FEM calculations are numerically inefficient and prevent more sophisticated multidisciplinary design studies. Modern ML methods offer a potent possibility to reduce the numerical effort. Similar to the method used in [9] NNs are trained by Dresia et al. [12] using samples of the computationally expensive calculation. The training data is generated by a FEM calculation of the first loading cycle followed by a fatigue life estimation during post-processing that includes Coffin-Manson theory and ductile failure. Approximately 120 000 data points are used for training and a cross-validation procedure helps to find the best network architecture as well as hyperparameter combinations.

The network achieves high precision in fatigue life prediction. Overall, the model estimates the number of cycles to failure with a mean squared error (MSE) of 239 on previously unseen data (equal to a mean percentage error of 7 %). The results are compared with the FEM calculation. The predicted effect of varying hot-gas side wall temperature and outer shell temperature on the expected number of cycles to failure is shown in Fig. 3. The predictive error is very small except in the areas where the input data is outside the range of the training data. This circumstance nicely shows that NNs generally cannot extrapolate. Overall, the methodology is well suited for optimization loops and as a component of system analysis tools.

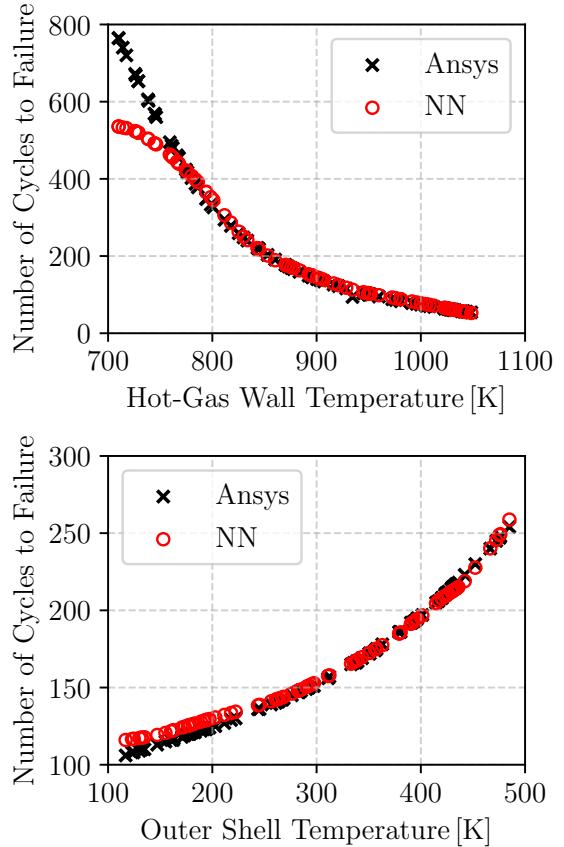


Figure 3: Predictive performance of the NN for the combustion chamber fatigue life.

3.1.3 Advantages and Disadvantages for Modeling Tasks

Advantages:

- similar accuracy as high-fidelity CFD or FEM simulations
- low prediction time, an NN only has to multiply the input vector with its weight matrices to generate the output
- NNs can scale to large datasets and capture the behavior of complicated functions with high-dimensional inputs and outputs
- data fusion and assimilation techniques allow to integrate multiple data sources and to combine simulation and experimental data in a systematic way

Disadvantages:

- depending on the complexity of the problem, the construction of a precise approximation model can require a huge number of data samples
- NNs are not able to extrapolate, but only provide reliable predictions within the region of the input space that is populated with training points

3.2 Control

Key technologies for the successful operation of reusable space transportation systems are the control and condition monitoring of the engines [13]. Space transportation systems that land again with retro-thrust require additional deep thrust throttling and restart capabilities. Optimal engine control can significantly increase the service life and thus contribute considerably to cost-efficient operation. The reliable use of reusable engines requires advanced condition monitoring systems. ML models can analyze sensor data quasi instantaneously, evaluate the current status and calculate the control signals. E.g. using deep reinforcement learning one can train NNs to approximate the optimal nonlinear mapping from sensor signals to actuation commands.

3.2.1 Early Detection of Thermoacoustic Instabilities with Support Vector Machines

Combustion instabilities are particularly problematic for rocket thrust chambers because of their high energy release rates and their operation close to the structural limits [14]. In the last decades, progress has been made in predicting high amplitude combustion instabilities but still, no reliable prediction ability is given. Especially thermoacoustic oscillations are a major hazard, but difficult to predict. An important question is whether features of combustion noise can be used to construct reliable early warning signals for representative rocket thrust chambers. Among other things, instability precursors are needed for active combustion control systems.

Waxenegger-Wilfing et al. [15] study the combination of combustion noise features with support vector machines. First, recurrence quantification analysis is used to calculate characteristic combustion features from short-length time series of dynamic pressure sensor data. The combination of several combustion noise features allows a more accurate estimation of the combustion condition and reduces the influence of outliers. To find the optimal combination and decision criterion respectively, support vector machines are trained to detect the onset of an instability a few hundred milliseconds in advance. The performance of the method is investigated on experimental data from a representative LOX/H₂ research thrust chamber. In most cases, the method is able to timely predict thermoacoustic instabilities on test data not used for training. Compared to the use of only one combustion noise feature, fewer false alarms are generated.

3.2.2 Start-Up Control of Gas-Generator Engines using Deep Reinforcement Learning

Nowadays, liquid rocket engines use closed-loop control at most near steady operating conditions. The control of the transient phases is traditionally performed

in open-loop due to highly nonlinear system dynamics. The situation is unsatisfactory, in particular for reusable engines. The open-loop control system cannot react to external disturbances. It is therefore intended to extend the use of closed-loop control to the transient phases. Only optimal control can guarantee a long life expectancy of the engine without damaging pressure and temperature spikes. The computational effort to calculate a suitable control action must not be too big so that the controller can be used in applications with fast dynamics. A widely recognized shortcoming of standard model predictive control is that it can usually only be used for slow dynamic situations, where the sample time is measured in seconds or even minutes. For small state and input dimensions, one can compute the entire control law offline and implement the online controller as a lookup table. But this does not work for higher state dimensions.

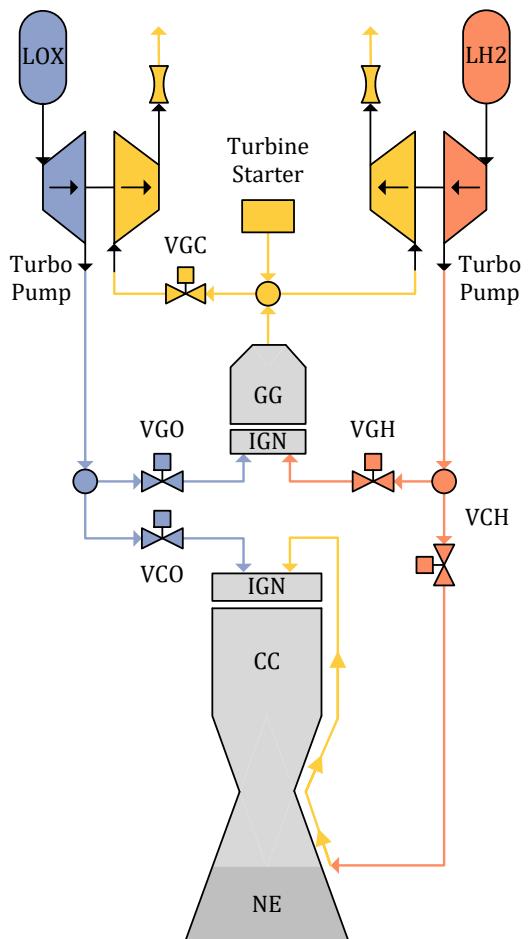


Figure 4: Flow plan of the considered engine architecture. Some of the propellants are burned in an additional combustion chamber, the gas-generator (GG), and the resulting hot-gas is used as the working medium of the turbines which power the engine's pumps. The gas is then exhausted. The engine architecture features five valves, but only three valves (VGH, VGO, VGC) are used for closed-loop control.

A deep reinforcement learning approach is investigated for optimal control of a generic gas-generator engine's continuous start-up phase [16]. The modeling and simulation tool EcosimPro is used as an engine simulator to train the NN controller. The considered engine architecture is similar to the architecture of the European Vulcain 1 engine (see Fig. 4), which powered the cryogenic core stage of the Ariane 5 launch vehicle before it got replaced by the upgraded Vulcain 2 engine. The multi-input multi-output (MIMO) control tasks study the active control of the combustion chamber pressure, the mixture ratio of the gas-generator as well as the global mixture ratio by regulating the gas-generator valves and the turbine valve. The valve actuators are modeled as a first-order transfer function and a linear valve characteristic.

The goal of the controller (RL agent) is to drive the engine state as fast as possible towards the desired reference by adjusting the flow control valve positions. Furthermore, the effect of degrading turbine efficiencies on the start-up transient is studied. This scenario has practical relevance for future reusable engines. The NN controller which is trained by RL achieves the best performance compared with carefully tuned open-loop sequences and PID controllers for different reference states and varying turbine efficiencies. Furthermore, the prediction of the control action takes only 0.7 ms, which allows a high interaction frequency. Fig. 5 shows

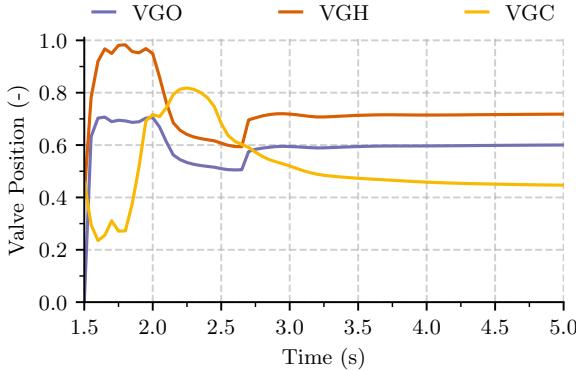


Figure 5: Valve positions for the engine start-up to a chamber pressure of 100 bar.

the manipulated valve positions for the 100 bar nominal start-up. The flow control valves are opened in a non-monotonic way to reduce the start-up duration. The RL agent directly takes the firing of the turbine starter into account. Furthermore, it can handle degrading turbine efficiencies. Deviating efficiencies are detected because the relationship between valve positions and controlled variables changes.

3.2.3 Set-Point Control of Expander-Bleed Engines using Deep Reinforcement Learning

Dresia et al. [17] study an NN based engine controller for the transient control of an expander-bleed liquid

rocket engine. Again, the NN is trained with the combination of modern RL algorithms and the well-validated simulation environment EcosimPro. The engine architecture is the same as used in the LUMEN engine demonstrator [18]. LUMEN is a modular LOX/LNG breadboard engine employing an expander-bleed cycle in the 25 kN thrust class for operation at the new test stand P8.3 in Lampoldshausen. LUMEN is very well suited to investigate advanced control approaches both theoretically and experimentally, as it employs multiple fast and precise flow control valves.

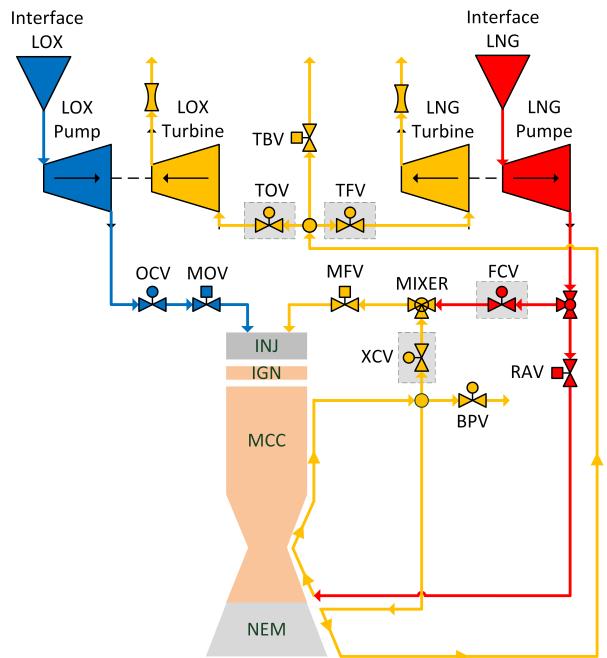


Figure 6: Flow plan of the LUMEN engine architecture.

Fig. 6 shows a schematic representation of the LUMEN engine cycle. The goal is to drive the engine to various combustion chamber pressures and mixture ratios quickly and without overshoot. Furthermore, the engine controller must keep several thermodynamic and mechanical parameters within certain limits to avoid damage to the engine. The evolution of the controlled chamber pressure for a given reference trajectory is presented in Fig. 7. For comparison, a simple open-loop control sequence is shown that linearly operates the valves within 0.5 s.

Compared to the open-loop (OL) control sequence, the NN controller follows the reference values much faster. For the combustion chamber pressure, the NN controller tracks the reference values very precisely. For both major load point changes from 60 to 80 bar and from 80 to 40 bar, the NN controller can adjust the combustion chamber pressure in less than 2 s. The OL system behaves considerably slower. When throttling from 80 to 40 bar, for example, it takes more than 10 s until a nearly steady-state combustion chamber pressure and mixture ratio is reached.

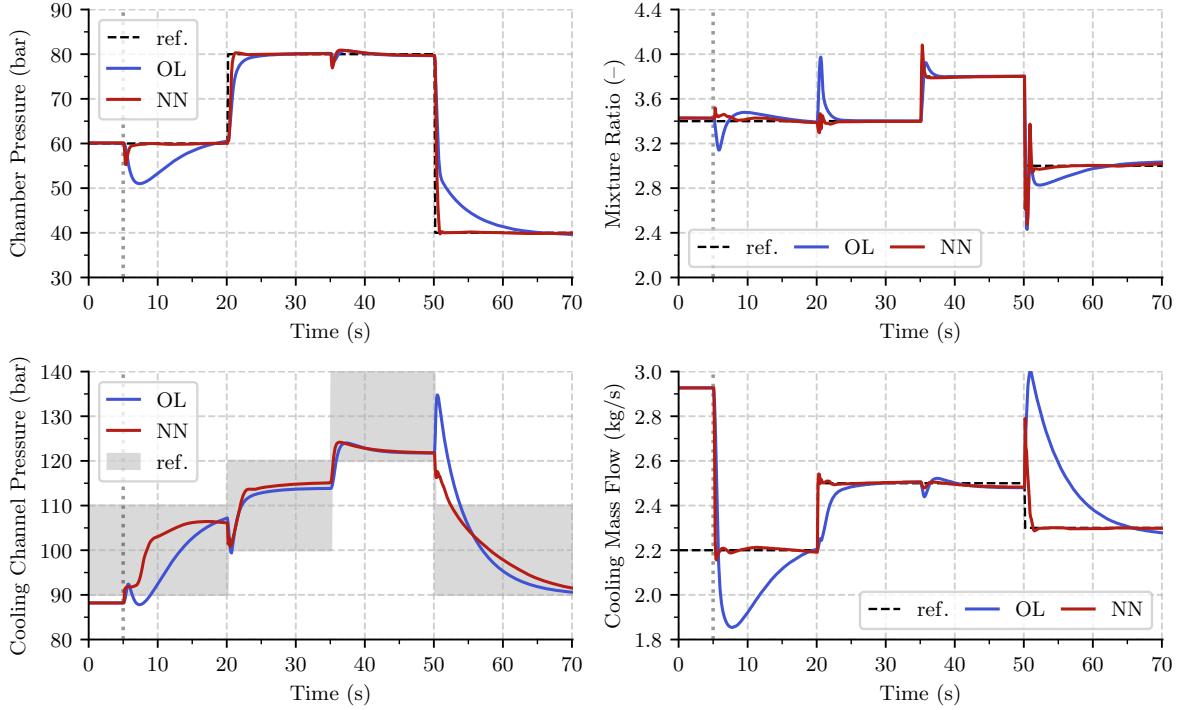


Figure 7: Controlled variables.

3.2.4 Advantages and Disadvantages for Control Tasks

Advantages:

- ML can be used to automatically deduce optimal features of measurement data for control and condition monitoring tasks
- RL control directly uses nonlinear simulation models, no derivation of suitable state-space models, model order reduction or linearization is needed
- ideal for highly dynamic situations (no complex online optimization needed)
- complex reward functions can be included in the RL approach and enable complicated control goals

Disadvantages:

- stability of an NN controller is in general not guaranteed

Concerning the last point (stability), we would like to make a remark. The output of an NN controller can be tested using the simulation environment, and there has been promising recent work on certifying stability of RL policies.

4. CURRENT AND FUTURE ACTIVITIES

Although the completed work impressively reflects the potential of ML methods for the design and operation of liquid rocket engines, they represent just the tip of the iceberg of future applications. For this reason, various research activities related to ML methods are taking place at the DLR Institute of Space Propulsion. The following sections provide a short overview.

the iceberg of future applications. For this reason, various research activities related to ML methods are taking place at the DLR Institute of Space Propulsion. The following sections provide a short overview.

4.1 Integration of Physical Laws into ML Models

Efficient data-based modeling of complex, technical machines or processes should take physical laws into account a priori and should not have to learn them from the training data. In this way, the number of training data could be reduced and the prediction quality increased. The following questions must be addressed: What is the optimal integration of physical laws into machine learning models? How can prior knowledge of physics be included in NNs? How can continuous and discrete symmetries be guaranteed? Recently, so-called Physics-Informed NNs (PINNs) are researched which aim at inferring continuous functions that arise as the solution to a system of nonlinear partial differential equations [19]. Such methods are able to learn among other things velocity and pressure fields from flow visualizations. Another exciting problem is the data-driven discovery of partial differential equations. Especially the implications of successful integration of physical laws for the areas of modeling, optimization, and control of space propulsion systems are analyzed.

4.2 ML-Based Dynamics Modeling

Compared to the modeling of stationary systems, the accurate modeling of dynamic systems is much more

difficult. With the help of so-called Gaussian processes and recurrent NNs, amazing successes have already been achieved. Reservoir computing represents a special kind of recurrent NNs and seems particularly well suited for the prediction of complex nonlinear dynamical systems [20]. Typical applications of dynamics models lie in the optimization of the dynamic system behavior and model-based control.

4.3 Uncertainty Quantification

The predictions of ML models are usually not perfect. However, if the predictions are used for design optimization or control tasks, one would like to know how certain the models are with their predictions. This requires estimating the prediction accuracy for a given input. Bayesian approximation and ensemble learning techniques are the two most widely-used uncertainty quantification methods [21]. In the future, uncertainty estimates will be essential for the acceptance of ML methods in safety-critical applications.

4.4 Life-Extending Control

The goal of life-extending control is to achieve high performance without damaging the system, e.g. by overstraining the mechanical structure [22]. The feasibility of a decision and control system for life extension has already been investigated for the Space Shuttle Main Engine (SSME). The results demonstrate the potential of damage mitigating control, especially for reusable rocket engines with a high number of engine reuses. ML methods can be used to derive optimal transient sequences and to effectively build nonlinear damage models.

4.5 Fault Detection and Diagnosis

Since rocket engines operate at the limits of what is technically feasible, they are inherently susceptible to anomalies [23]. The immense costs associated with the loss of the launch vehicle or a test bench clearly show the importance of a suitable condition monitoring system. Machine condition monitoring systems may have to provide a proper diagnosis in real-time from existing sensor data to detect abnormal behavior and, for example, to trigger an emergency shutdown. Usually, the detection of faults is realized by just monitoring if a sensor signal exceeds a certain threshold or analyzing the discrepancy between sensor readings and expected values, derived from a theoretical model. In cases where the exact theoretical modeling is not possible or would be very costly, statistical methods or direct pattern recognition algorithms are used. ML techniques belong to this second category and received significant attention in recent years.

Various faults can occur during the operation of liquid rocket engines. These faults range from clogging and

ablation of injectors, combustion instabilities, not moving valves to problems with the turbopumps like cracks in the turbine blades, rubbing of the rotors, damage to the bearings and seals as well as cavitation in the propellant pumps. Although condition monitoring of rotating machinery is particularly well developed, including methods like vibration monitoring with spectral analysis, its application in turbopumps of rocket engines is largely unresearched. Some faults that can be detected best by monitoring the interaction of several subsystems. To make matters worse, the diagnosis should work both during steady operation and during transient phases such as the start-up of the engine or a change of operating point.

4.6 Fault-Tolerant Control

The value of fault detection and diagnosis algorithms is enhanced by the presence of optimal responses and countermeasures [24]. The following questions are addressed: What is a suitable fault-tolerant control scheme for rocket engines? Can physical sensors be replaced by virtual sensors? Can a real-time prediction of combustion instabilities be used to realize active combustion control? What role do ML algorithms play in this?

4.7 Application of ML Models in Safety-Critical Situations

The control of space propulsion systems is a safety-critical application. Therefore, the software used must meet high standards. An important research question is how can safety be ensured when ML algorithms are used. Topics like verification, validation, and testing of the critical software will be investigated in the future to finally identify the appropriate steps of a suitable certification.

4.8 Embedded Systems for Modern Rocket Engine Control

It is also planned to focus on the realization of embedded systems that are suitable for the researched control methods. The focus will be on the special challenges to the hardware and software which arise from the space application. The computing power during the operating phase of ML algorithms is typically limited. This is particularly true for space applications, where robust and failsafe computing hardware is used in the harsh environment of space or during a rocket launch. Furthermore, high-performance computing hardware with high weight is not feasible for space launch vehicles as weight is one of the most critical design parameters limiting the performance of the launcher.

In recent years, different techniques were developed to reduce the computational demands during NN training

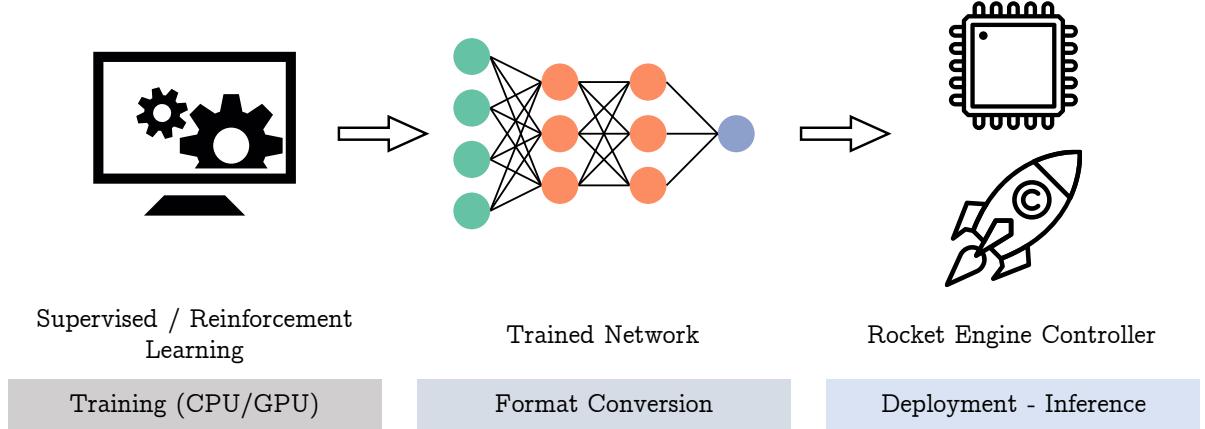


Figure 8: Deployment logic of a neural network controller.

and inference. These techniques can reduce memory usage, and increase inference speed and energy efficiency. However, these improved methods are only necessary for very deep and complex NN architectures used for computer vision (e.g., object detection or video tracking), natural language processing, or when training the neural network directly on the embedded system. For most applications, it is enough to deploy only the trained NN on the embedded system [25]. One would train the NN using suitable training data (supervised learning) or simulation environments (RL) on a dedicated workstation in Python with commonly used ML frameworks such as TensorFlow or PyTorch. Then, one would convert the NN to C/C++, and copy the network to the embedded system on the actual space system. Thus, the embedded system just needs to handle the inference, which has much lower computational demands than the training of the network.

5. CONCLUSION

Although ML methods have already proven to be a valuable tool for the design and operation of liquid rocket engines, these techniques are currently mostly known and used by academics and a few industrial researchers. We are confident that the use will increase strongly in the future. The start of working with ML methods is significantly simplified by freely accessible ML frameworks. These frameworks are characterized by the fact that they are based on linear algebra libraries with additional automatic differentiation routines. This allows to easily differentiate complex mathematical functions that are represented with the help of the framework. Differentiation is important because many ML algorithms essentially solve optimization problems with many parameters, so that gradient-based optimization algorithms are well suited. Based on this, implementations of many standard ML algorithms, such as the training of various types of NNs or modern RL agents, are available.

It is often argued that there is not enough data available in the field of rocket engines to use modern ML

methods. This argument is largely wrong because a lot of the necessary training data can be generated synthetically, e.g. through computer simulations. Furthermore, one does not need innumerable data from occurring anomalies to be able to detect faults precisely. For the pure detection of anomalies, it is sufficient to have or generate data that characterize the regular operation. Ongoing work on the validation and certification of ML models will increase the trust in these models, which are often regarded with skepticism when seen as a black box.

There are many important areas regarding the application of ML methods in rocket propulsion that we have not addressed in this short paper. We would particularly like to mention the field of ML-assisted CFD simulations which has made amazing progress in the last few years [3]. In the case of rocket engines, the modeling of turbulent flows and combustion phenomena is expected to benefit from this.

Acknowledgments

It is a pleasure to thank Charlotte Debus, Philipp Knechtges, Christoph Räth, Alexander Rüttgers, and Ushnish Sengupta for many useful discussions.

References

- [1] C. M. Bishop, *Pattern Recognition and Machine Learning*. Information Science and Statistics, New York: Springer, 2006.
- [2] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Adaptive Computation and Machine Learning, Cambridge, Massachusetts: The MIT Press, 2016.
- [3] M. Frank, D. Drikakis, and V. Charissis, "Machine-Learning Methods for Computational Science and Engineering," *Computation*, vol. 8, p. 15, Mar. 2020.

- [4] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Adaptive Computation and Machine Learning Series, Cambridge, Massachusetts: The MIT Press, second edition ed., 2018.
- [5] D. P. Bertsekas, *Reinforcement Learning and Optimal Control*. Athena Scientific, 2019.
- [6] V. N. Vapnik, “An overview of statistical learning theory,” *IEEE Transactions on Neural Networks*, vol. 10, no. 5, pp. 988–999, 1999.
- [7] M. Pizzarelli, F. Nasuti, M. Onofri, P. Roncioni, R. Votta, and F. Battista, “Heat transfer modeling for supercritical methane flowing in rocket engine cooling channels,” *Applied Thermal Engineering*, vol. 75, pp. 600–607, Jan. 2015.
- [8] J. Haemisch, D. Suslov, and M. Oschwald, “Experimental Study of Methane Heat Transfer Deterioration in a Subscale Combustion Chamber,” *Journal of Propulsion and Power*, vol. 35, pp. 819–826, July 2019.
- [9] G. Waxenegger-Wilfing, K. Dresia, J. C. Deeken, and M. Oschwald, “Heat Transfer Prediction for Methane in Regenerative Cooling Channels with Neural Networks,” *Journal of Thermophysics and Heat Transfer*, vol. 34, no. 2, pp. 347–357, 2020.
- [10] J. Haemisch, D. Suslov, G. Waxenegger-Wilfing, K. Dresia, and M. Oschwald, “LUMEN – Design of the Regenerative Cooling System for an Expander Bleed Cycle Engine using Methane,” in *Space Propulsion 2020+1 Conference*, (Virtual Event), 2021.
- [11] P. Kringe, J. Riccius, and M. Oschwald, “Low-cost life assessment of liquid rocket engines by replacing full-scale engine tests with TMF panel tests,” *Journal of the British Interplanetary Society*, vol. 73, no. 5, pp. 154–162, 2020.
- [12] K. Dresia, G. Waxenegger-Wilfing, J. Riccius, J. Deeken, and M. Oschwald, “Numerically Efficient Fatigue Life Prediction of Rocket Combustion Chambers using Artificial Neural Networks,” in *8th European Conference for Aeronautics and Space Sciences 2019 (EUCASS)*, (Madrid, Spain), 2019.
- [13] S. Pérez-Roca, J. Marzat, H. Piet-Lahanier, N. Langlois, F. Farago, M. Galeotta, and S. Le Gondrec, “A survey of automatic control methods for liquid-propellant rocket engines,” *Progress in Aerospace Sciences*, vol. 107, pp. 63–84, 2019.
- [14] V. Yang, ed., *Liquid Rocket Engine Combustion Instability*. No. 169 in Progress in Astronautics and Aeronautics, Washington, DC: American Inst. of Aeronautics and Astronautics, 1995.
- [15] G. Waxenegger-Wilfing, U. Sengupta, J. Martin, W. Armbruster, J. Hardi, M. Juniper, and M. Oschwald, “Early Detection of Thermoacoustic Instabilities in a Cryogenic Rocket Thrust Chamber using Combustion Noise Features and Machine Learning,” *arXiv:2011.14985 [eess]*, Nov. 2020.
- [16] G. Waxenegger-Wilfing, K. Dresia, J. C. Deeken, and M. Oschwald, “A Reinforcement Learning Approach for Transient Control of Liquid Rocket Engines,” *arXiv:2006.11108 [cs.LG]*, 2020.
- [17] K. Dresia, G. Waxenegger-Wilfing, and M. Oschwald, “Nonlinear Control of an Expander-Bleed Rocket Engine using Reinforcement Learning,” in *Proceedings of the Space Propulsion 2020+1 Conference*, (Virtual Event), 2021.
- [18] J. Deeken and G. Waxenegger, “LUMEN: Engine cycle analysis of an expander-bleed demonstrator engine for test bench operation,” in *Deutscher Luft- Und Raumfahrtkongress*, (Braunschweig, Germany), 2016.
- [19] M. Raissi, P. Perdikaris, and G. Karniadakis, “Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations,” *Journal of Computational Physics*, vol. 378, pp. 686–707, Feb. 2019.
- [20] A. Haluszczynski and C. Räth, “Good and bad predictions: Assessing and improving the replication of chaotic attractors by means of reservoir computing,” *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 29, p. 103143, Oct. 2019.
- [21] M. Abdar, F. Pourpanah, S. Hussain, D. Rezaazadegan, L. Liu, M. Ghavamzadeh, P. Fieguth, X. Cao, A. Khosravi, U. R. Acharya, V. Makarenkov, and S. Nahavandi, “A Review of Uncertainty Quantification in Deep Learning: Techniques, Applications and Challenges,” *arXiv:2011.06225 [cs]*, Jan. 2021.
- [22] A. Ray, M. S. Holmes, and C. F. Lorenzo, “Life extending controller design for reusable rocket engines,” *The Aeronautical Journal*, vol. 105, pp. 315–322, June 2001.
- [23] J. Wu, “Liquid-propellant rocket engines health-monitoring—a survey,” *Acta Astronautica*, vol. 56, pp. 347–356, Feb. 2005.
- [24] C. Sarotte, J. Marzat, H. Piet-Lahanier, G. Ordonneau, and M. Galeotta, “Model-based active fault-tolerant control for a cryogenic combustion test bench,” *Acta Astronautica*, vol. 177, pp. 457–477, Dec. 2020.
- [25] G. Waxenegger-Wilfing, K. Dresia, M. Oschwald, and K. Schilling, “Hardware-In-The-Loop Tests of Complex Control Software for Rocket Propulsion Systems,” in *71th International Astronautical Congress (IAC)*, (Virtual Event), 2020.