AI-Assisted Control Systems for Rocket Engine Test Facilities within the ESA FLAME Program Estoril 2022

Kai Dresia[1,4], Eldin Kurudzija[1], Günther Waxenegger-Wilfing[1,2], Hendrik Behler[1], Daniel Auer[1], Karsten Fröhlke[1], Heike Neumann[1], Anja Frank[1], Jérôme Laurent[3] and Luce Fabreguettes[3]

[1] *German Aerospace Center (DLR), Institute of Space Propulsion, 74239 Lampoldshausen, Germany*
[2] *University of Würzburg, Institute of Computer Science, 97074 Würzburg, Germany*
[3] *European Space Agency (ESA), Paris, France*
[4] *Corresponding author: kai.dresia@dlr.de*

KEYWORDS: rocket engine test facilities, artificial intelligence, digital twin, anomaly detection, condition monitoring, intelligent control

ABSTRACT:

The DLR Institute of Space Propulsion has unique expertise in Europe in the development and operation of engine test facilities. Since 1959, engines for rockets and space propulsion systems have been tested and further developed at the Lampoldshausen site. Nevertheless, essential elements of the DLR Lampoldshausen technical center were designed up to 50 years ago. In the frame of the FLAME program, ESA has initiated multiple modernization measures to enhance the competitiveness and the service levels of the engine test facilities. In order to guarantee a smart digital test infrastructure that is ready for the future, the potential of advanced control and monitoring systems are studied. In the long term, such intelligent control systems promise to reduce the development times of engines and engine demonstrators and thus reduce development costs. Furthermore, AI-assisted monitoring systems will further increase safety and reliability.

1. INTRODUCTION

The paper presents two pilot projects which take place within the framework of FLAME (Future LAMpoldshausen Exploitation). The first project evaluates the benefits and usability of artificial intelligence concepts for automated sequence generation in rocket engine test facility operations: Reinforcement learning is used to find optimal test sequences according to user requirements in terms of time-dependent interface conditions and the expected engine operation point.

The second project studies the use of artificial intelligence concepts regarding fault detection and isolation of rocket engine test facilities. Deep neural network models learn the characteristics of the nominal test facility behavior by using historical and synthetic data generated with a digital twin of the system. Afterwards, these models offer real-time detection capabilities of anomalous behavior similar to Park and Ahn [1].

An ideal candidate to evaluate the potential for the AI-methods is given by the LOX system of the P5 test facility at DLR Lampoldshausen. It provides a high amount of actual test data and will be used during Prometheus engine tests scheduled by 2022/2023 [2].

During the project, a detailed system model of the P5 LOX system is built and validated against real test data. This activity paves the way to a future virtual twin of the test bench P5, which can be used to optimized and analyze the test. In addition, a virtual twin enables to study the test bench configuration and control sequences before the real test run, which increases the reliability and precision of the test bench. Finally, a virtual twin can identify bottlenecks that limit the operating range of the test bench, such as slow control valves or saturated pressure regulators. These subsystems can then be replaced or strengthened.

2. OVERVIEW OF THE FLAME PROJECT

The unique test facilities owned by ESA which are located at the DLR test site Lampoldshausen and managed and operated by DLR Lampoldshausen are an important strategic asset for the European launcher sector and guarantee the independent European access to space.

The existing infrastructure and test facilities allow the performance of test activities for research and development as well as production purposes for the European Launchers. However, essential elements were conceived up to 50 years ago. Regular wear as well as technological progress and legislative evolution require the implementation of an extensive (and first!) renewal initiative.

The Future LAMpoldshausen Exploitation (FLAME) program has therefore been approved at the Ministerial Council in November 2019. FLAME is proposing a significant renewal in infrastructure and processes:

- Offering state-of-the-art service responding to the needs of Ariane and Vega to allow maximum use of these public sector operated facilities

- Minimizing future maintenance costs, all borne by the public sector
- Minimizing operation costs, to contribute to the competitiveness of the launch service.

The concept phase for the FLAME project started in early 2020 and the implementation phase is on-going since end of 2021.

3. P5 test facility

At DLR Test center in Lampoldshausen DLR is testing rocket engines on ESA test benches in the frame of Ariane program. 32 years ago, in 1989 the test bench P5 was built up for testing the Vulcain 1 as the new Ariane 5 main stage engine. During the evolution of the Vulcain engine the test bench was modified for the new versions Vulcain 2 and later Vulcain 2.1 for the Ariane 6 rocket.

The rocket engine is fixed inside the test cell at the so called thrust cone. This thrust cone, built of thick steel plates has the shape of the Ariane 5 main stage lower part and can take the over 100 tons of thrust of the Vulcain engines, which use liquid oxygen and liquid hydrogen as propellants.

To provide the high amount of liquid oxygen and liquid hydrogen, the test bench has a 200 m^3 LOX and a 600 m^3 LH2 tank. To feed the engine with fuel, these tanks can be pressurized with gaseous nitrogen for LOX and gaseous hydrogen for LH2. The liquids flow from the tanks via the main feed lines to the turbopump interface at the engine. Currently, the design of the bench allows a flow rate up to 310 $kg\,s^{-1}$ of LOX and 54 $kg\,s^{-1}$ of LH2. With this tank capacity, the P5 can perform tests at full thrust for up to 720 seconds until the tanks are nearly empty.

Performing tests with a rocket engine is very complex and a lot of components like valves and sensor values need to be activated, checked and regulated in a specific way and time. Therefore nearly the whole test is performed by sequences on a test computer. The preparation of tests including programming of these sequences, refilling the tanks and also the mechanical checks of the engine according to customer's requirements can take up to two weeks. Now the test bench P5 is in modification phase again for the next generation rocket engine, such as the upcoming LOX/CH4 engine Prometheus in 2022.

3.1 Overview of the P5 lox system

This section gives an overview of the LOX system of the P5 test bench. The central component is the LOX tank with a volume of 200 m^3. The tank is pressurized with gaseous nitrogen (N2) and is situated above the engine. The tank pressure (PFRO) is controlled via two pressure regulator valves CVO150 and CVO743.

Figure 1 – Test bench P5

Two PI-controllers regulate the valve position based on the tank pressure and the user given reference tank pressure (CVO150ARC).

To quickly lower the tank pressure, three different sized ventilation valves (AVX141, AVX149, AVX3252) are available. The final section of the LOX system is the LOX line from the tank to the engine interface. The engine interface pressure, i.e. the pump inlet pressure, is denoted by POEP. QOPER is the mass-flow measured within the LOX run line.

4. TEST BENCH MODELING

In this study, we use the well-validated simulation software EcoimPro/ESPSS for system modeling. EcosimPro can model 0D or 1D continuous and discrete systems based on differential-algebraic equations. Within a graphical user interface, one can combine different components from several pre-defined libraries. Furthermore, the user can create custom components to implement new correlations or even use data-driven models like artificial neural networks [3–5].

Of particular interest are the European Space Propulsion System Simulation (ESPSS) libraries, which are commissioned by the European Space Agency (ESA). The ESPSS library (used in version 3.3.0) provides implementations of many components relevant for the P5 LOX system (pipes, orifices, valves, tank).

Overall, the simulation model of the P5 LOX system features important physical effects including pressure losses in pipes, elbows, and valves, the heat and mass-transfer between LOX and N2 in the tank, thermodynamic properties of LOX at cryogenic conditions, the static pressure of the LOX caused by gravity, as well as the control characteristics of the pressure regulators.

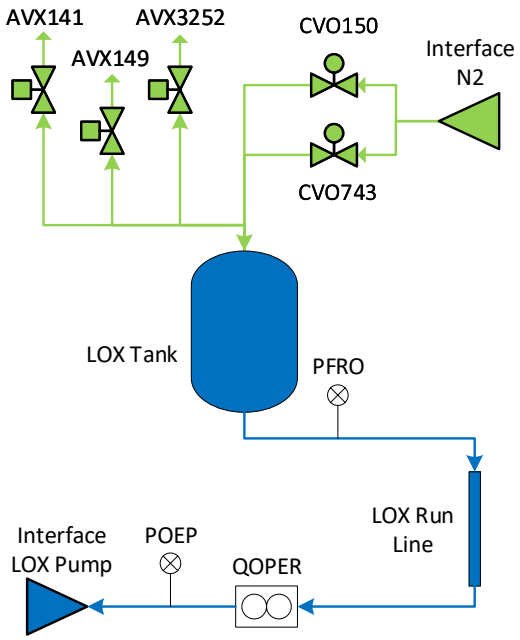

Figure 2 – Simplified scheme of the P5 LOX system

The valve transfer function describes the valve movement in terms of valve control signal and the current valve position. For CVO150 and CVO743, a first-order transfer function is combined with an input delay to model the behavior of the hydraulic regulator valves.

CVO150 and CVO743 are controlled via digital PI-controllers. This control logic is encapsulated in a custom component within EcosimPro, which offers the possibility to adjust the P- and I – parameters of the PI-controllers.

The LOX tank is modeled with the *TankCylDomes* component of the ESPSS tank library. The tank has lower and upper dome and a large cylindrical segment. To model the mass and energy exchange at the gas/liquid interface, the energy-balance approach of the tank component was chosen: "Assuming a very thin layer at T_{sat}, the equilibrium conditions would allow calculating the evaporation (positive or negative) flow rate through this layer" [6].

The interface conditions of the high pressure N2 are provided by a pressure regulator. A data-based modeling approach using real test data is chosen to describe the outlet pressure characteristics of the pressure regulator as a function of the N2 mass flow.

Overall, the system has five boundary conditions:

- The LOX mass flow rate (QOPER), which is imposed at the pump inlet
- The reference tank pressure (CVO150ARC)
- The 3 ventilation valve command signals.

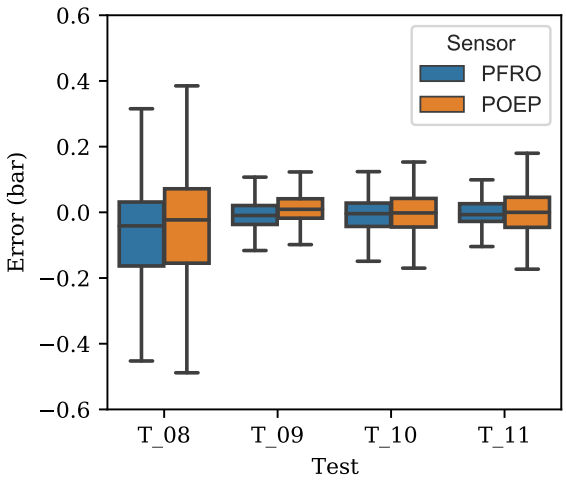

Figure 3 – Comparison between the simulation model and real test data for different test runs

4.1 Model validation

The EcosimPro model of the P5 LOX system is validated against real test data with a total hot-run time of 2700 s. These test runs cover various engine operation points in terms of LOX mass flow rate and pump inlet pressure. The pressurization sequence (CVO150ARC) and the ventilation commands of the real test are imposed to the EcosimPro model and the resulting the pump inlet pressure and tank pressure are compared to their respective real test values.

Figure 3 shows the error of the simulation model for four different test runs denoted by T_{08} to T_{11}. For three of the four test runs, the error is well below ± 0.2 bar, proofing that the model is capable of simulating the time-depended pump inlet pressure, the tank pressure, and the effects of ventilation for different engine operation points.

Compared to the other test runs, the error for T_{08} is larger. The underlying cause was found to be the data-based modeling of the N2 interface pressure regulator: During T_{08}, the N2 interface pressure was unusually low, after several pre-tests had been performed prior to the actual test run. Therefore, in the future, the modeling of the N2 pressure regulator should be replaced by a physics-based modeling to reduce the error for these rare experimental conditions.

5. CONTROL SEQUENCE GENERATION

Controlling the pump inlet pressure with a tolerance of only ± 0.2 bar is challenging given the large operating range of the test bench and the immense mass flows and tank volumes.

Let us first consider only the steady-state case: the resulting pump inlet pressure is the static tank pressure plus the hydrostatic pressure of the LOX fluid volume

minus pressure losses in pipes:

$$p_{\text{pump-inlet}} = p_{\text{tank}} + p_{\text{hydrostatic}}(V_{\text{tank}}) - p_{\text{loss}}(\dot{m}_{\text{engine}}) \quad (1)$$

As indicated in equation 1, the hydrostatic pressure changes with the liquid LOX tank level. Likewise, the pressure losses depend on the LOX mass flow. These factors must be considered in the design of the tank pressurization sequence.

Transient phases, such as major engine operating point changes, are even more difficult to control due to the immense tank volume. Towards the end of the test, for example, there is a lot of compressible gas in the tank, which renders pressure changes time-consuming. Therefore, when changes in operating point are imminent, the tank pressure must be changed some time in advance.

The precise timing of tank pressurization and venting is therefore based on many years of experience of the test bench operators.

5.1 AI-based automated sequence generation

The overarching idea is to formulate the control sequence generation problem as an optimization problem, which can be solved by a machine learning algorithm. The goal is to find the optimal time-dependent tank reference pressure and ventilation commands, that achieves the desired pump inlet pressure.

A similar approach was already studied for the generation of optimal control sequences of an expander-bleed rocket engine [7]. The study demonstrated, that machine learning can find optimal control sequences when combined with a detailed EcosimPro system model.

5.2 Optimization problem

First, the automated sequence generation must be formulated as an optimization problem. It can be stated as following a given reference pump inlet pressure profile as precise as possible.

Mathematically, an objective function can be formulated as minimizing the normalized error between the actual pump inlet pressure (POEP) and the reference pump inlet pressure $POEP_{\text{ref}}$ summed over the entire test run at every time step of the simulation

$$\min \sum_{t=0}^{t=t_{\text{end}}} \left(\frac{|POEP(t) - POEP_{\text{ref}}(t)|}{0.2\,\text{bar}} \right)^x, \quad (2)$$

where x depends on the error function. For example, $x = 1$ is the mean absolute error. For this study, the Huber loss is used, which is quadratic for small errors ($x = 2$), and linear for large errors.

Finally, one must define how often the AI-algorithm is allowed to adjust the tank pressurization. In general, the best performance could be achieved by allowing the AI algorithm to adjust the tank pressure at each time step. On the other hand, the resulting AI-based sequence should be interpretable by humans and comparable to the previously used sequences. Thus, the AI algorithm is only allowed to change the tank pressurization at certain points in time, which are provided by the user.

5.3 Machine learning algorithm

As a suitable machine learning algorithm, deep reinforcement learning [8] is used for solving the optimization problem. It is a machine learning method that derives optimal control sequences through the interaction with a dynamic environment, for example an EcosimPro model. The control law is parameterized in form an artificial neural network.

During training, the algorithm explores different control inputs and evaluates the system response based on the objective function. Then, the algorithm slightly adjusts the parameters of the artificial neural network in the direction of a lower error, thus improving the control law.

In this paper, the off-policy Soft-Actor-Critic (SAC) algorithm [9], implemented within the Ray RLlib framework [10] is used. Compared with other older reinforcement learning algorithms, for example DDPG, SAC is more stable during training and does not need extensive hyper-parameter tuning. Compared with on-policy algorithms like PPO, SAC is more sample efficient, meaning it needs less training data to find the optimal control strategy. SAC can also be trained in a distributed manner to take advantage of multi-CPU machines, further reducing the training time.

A detailed description of reinforcement learning, the SAC algorithm, and its application in the field of liquid propellant rocket engines is given by Waxenegger-Wilfing et al. [11].

5.4 Generic test requirements

The potential of the automated control sequence generation is evaluated for a generic test sequence, which has been developed in the light of a realistic mission profile of a first stage rocket engine used on launchers comparable to the Ariane 5 or Ariane 6. The test sequence combines realistic features of a launcher flight profile with some ambitious testing features challenging the AIs ability to find an optimized solution to the presented pump inlet pressure profile.

Figure 4 illustrates the pressure and LOX mass flow profile of the generic test sequence. The engine start-up begins at $t = 0\,\text{s}$ and lasts until $t = 10\,\text{s}$. During the start-up sequence the pressure inside the tank is usually not adapted in order to avoid any disturbance of the engine start-up transient.

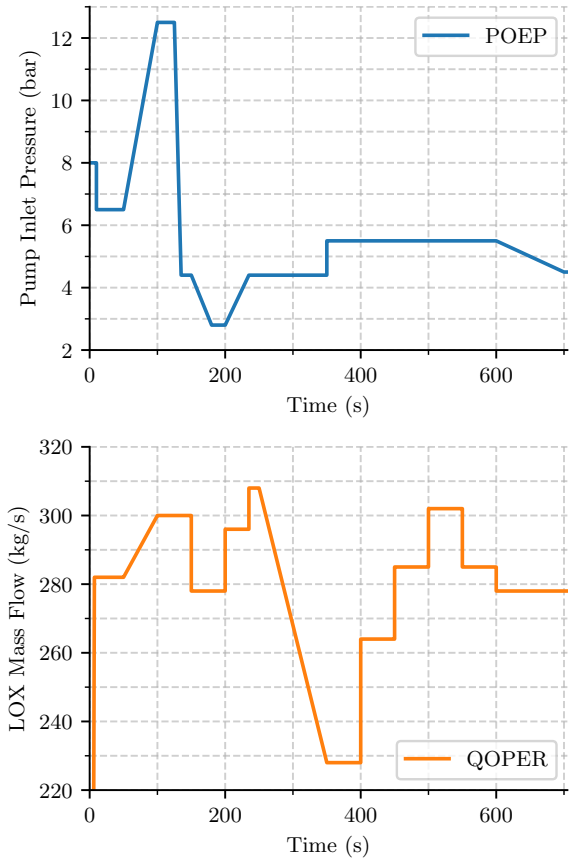

Figure 4 – Generic test requirements

After completing the start-up transient, the tank pressure is adjusted to match the required pump inlet pressure profile as closely as possible within a desired tolerance of ±200 mbar.

First, one could think of a pressure increase at the LOX pump inlet due to the increasing acceleration of a launcher with solid rocket boosters typical for Ariane 5 or Ariane 6. Then, these solid rocket boosters are fictitiously jettisoned at $t = 125$ s, which causes an abrupt pressure drop at the pump inlet.

Between $t = 150$ s and $t = 250$ s, engine throttling is simulated passing through the flight phase of maximum aerodynamic pressure.

The test phase between $t = 250$ s to $t = 350$ s is used to characterize the engines throttling capabilities. Starting with high thrust, the engine is continuously throttled down to an operating point with very low thrust. During this phase, the pump inlet pressure must be kept constant.

In the second half of the test starting from $t = 400$ s until then end of the test at $t = 700$ s, various engine operating points are simulated including a sloping pressure ramp between $t = 600$ s and $t = 700$ s.

The test requirements are prepared for the AI algorithm and the training process is started. On an ordinary workstation with 8 CPU cores, it takes about one day for the reinforcement learning algorithm to find a suitable pressurization sequence. This sequence is then used as an initial guess and slightly post-processed manually.

5.5 Results and discussion

In the following, the AI-based pressurization sequence found by the reinforcement learning algorithm will be discussed. The second row of Figure 5 shows the commands to the three vent valves AVX149, AVX141, and AVX3252.The third row shows the target tank pressure curve. Finally, row four shows the pump inlet pressure curve, and the corresponding reference value.

Overall, the AI-based sequence is promising: it achieves very precise control of the pump inlet pressure throughout the test run and manages to avoid large overshoots altogether. Furthermore, the tank reference pressure profile (CVO150ARC) looks comparable to a sequence that would have been found manually by the test bench operators.

Now, let's take a look at the individual test phases in detail: During the first phase, the simulated acceleration of the launcher followed by a sudden jettison of the solid rocket booster, the pump inlet pressure is first increased as fast as possible to approximately 12.2 bar. Then, the AI-algorithm activates the largest ventilation valve AVX3252 until the pump inlet pressure reaches the desired value of 4.4 bar.

In the following test phase until 250 s, the AI-algorithm uses the medium-sized ventilation valve AVX141 to simulate the lower pump inlet pressure during engine throttling while passing the flight phase of maximum aerodynamic pressure.

Until $t = 350$ s, the engine is throttled down to an operating point with very low thrust at a constant pump inlet pressure. This test requirement is full-filed by precisely adjusting the tank reference pressure approximately every 20 s.

The second half of the test starting from $t = 400$ s simulates different engine operating points. This phase challenges the pressurization system as the pressure of the already half-empty tank cannot be changed instantly. Therefore, the AI-algorithm adjusts the tank reference pressure up to 15 s before the engine changes it operation point. By doing this, large overshoots in the pump inlet pressure can be avoided.

The final test phase requires a sloping pressure ramp from a pump inlet pressure of 5.5 bar to 4.5 bar within 100 s. During this phase, the AI algorithm activates the smallest vent valve AVX149 for about 20 s. Then, the pressure ramp is achieved by repeatedly lowering the tank reference pressure.

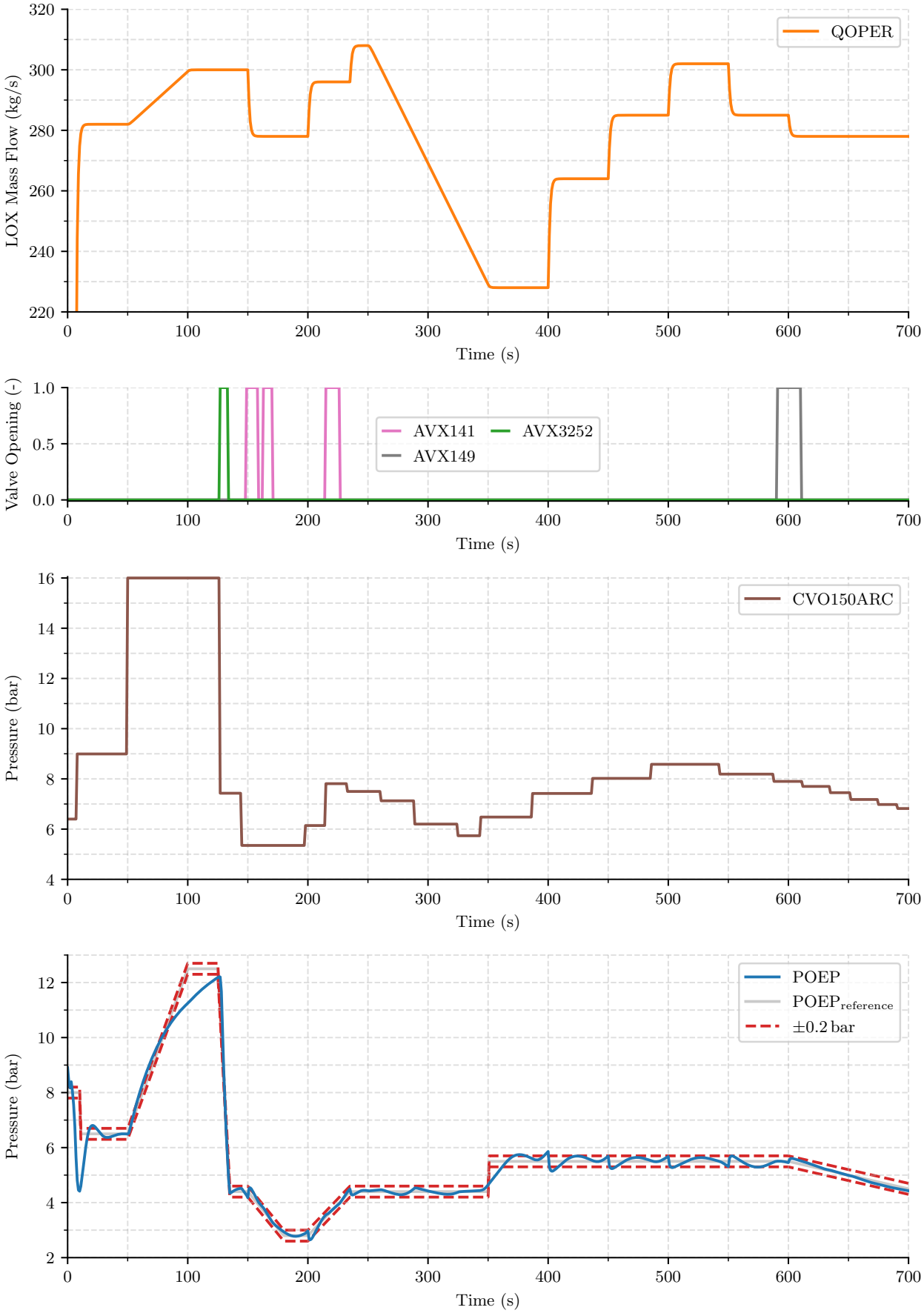

Figure 5 – AI-based pressurization sequence with resulting pump inlet pressure POEP

6. FAULT DETECTION

The current logic for a test abort is based on conventional surveillances. The test is aborted if a recorded sensor signal is indicating a value higher or lower than a given time-dependent maximum or minimum value. This approach offers potential for improvement. A more sophisticated surveillance logic would allow for a better usage of margins and less unintentional test aborts and therefore a more economic usage of available test capabilities.

6.1 AI-based fault detection

With the recent advances in AI and especially the development of efficient deep learning algorithms, novel AI-based fault detection methods have been developed. Depending on the number of available training data, the complexity of the monitored processes and the existence of representative fault cases, the most appropriate method is selected.

The prediction of anomalies does not need innumerable data from different and yet unknown anomalies. This approach follows the assumption that an anomalous event differs significantly from regular test bench operation. With that it is sufficient to train a machine learning model with sensor data belonging to nominal behavior. In this case, the model learns to characterize the nominal test bench operation. Autoencoder models are a special form of neural networks which can be used for this type of detection logic.

Simply put, autoencoder learn how to efficiently compress nominal sensor data in order to reconstruct it from the reduced encoded representation into a decoded representation that is as close as possible to the original input. This can only be done by using characteristic features of the nominal data. More details can be found in [12, 13].

If the sensor data are available as time series, faults can also be detected by forecasting, i.e. by constructing a suitable forward model followed by a step by step comparison of the predicted variables with the measured variables. Different neural network architectures, such as fully-connected feedforward or recurrent networks, can be used to represent the forward model.

In the paper by Waxenegger-Wilfing et al. [14], an AI approach to fault detection is presented that uses explicitly known anomalies (in this case, combustion instabilities in rocket combustion chambers) for training. Such a supervised method often increases the detection accuracy, but is then specifically tailored to the types of anomalies used.

6.2 Training data

Although sufficient real data is available for training of AI models due to the decades of operation of the test bench, synthetic data generated with the help of the developed P5 simulator is used in the context of these pilot study. This allows to investigate the potential of the AI methods without complicating factors such as sensor noise or non-uniform data distribution, and provides a solid foundation for the integration of these more complex real-world effects.

Specifically, 7549 test runs are simulated with different artificially generated pressurization and ventilation sequences as well as changing LOX mass flows. The average length of a simulated run equals 730 s. In total, there are roughly 5.5 million data points available for the training phase.

6.3 Fault cases

Although we do not use data related to faults for training, such data can be used for a quantitative investigation of the expected detection accuracy. Since anomalous events differ from the nominal behavior it is not important to take all error cases into account.

The following fault cases are simulated to examine the ability of the AI models to detect them in a timely manner. Overall, five different fault types are considered and 50 different test runs are simulated for each fault type.

First, a sensor offset could occur for example due to an external voltage that interferes with the measured value. We generate random offsets for the value of PFRO in the range from 0.5 bar to 3.0 bar. Second, a faulty sensor can cause a slowly drifting measurement value. We focus again on pressure readings and assume drift rates from 3 mbar to 5 mbar per second. Third, a malfunctioning sensor may cause a frozen signal.

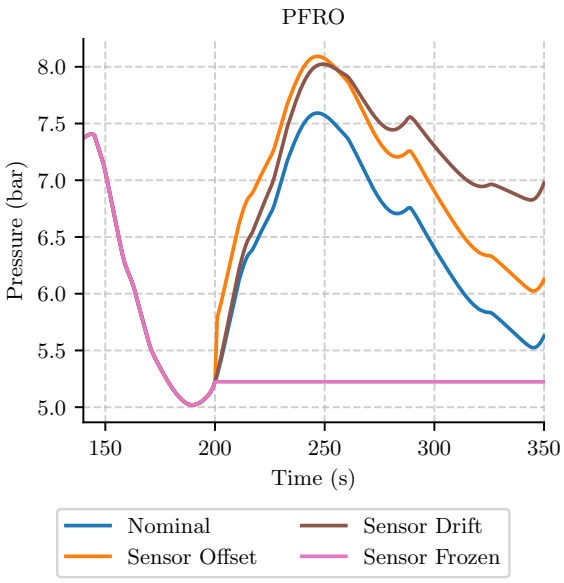

Figure 6 – Simulated sensor faults for the generic test sequence. The faults occur at $t = 200$ s.

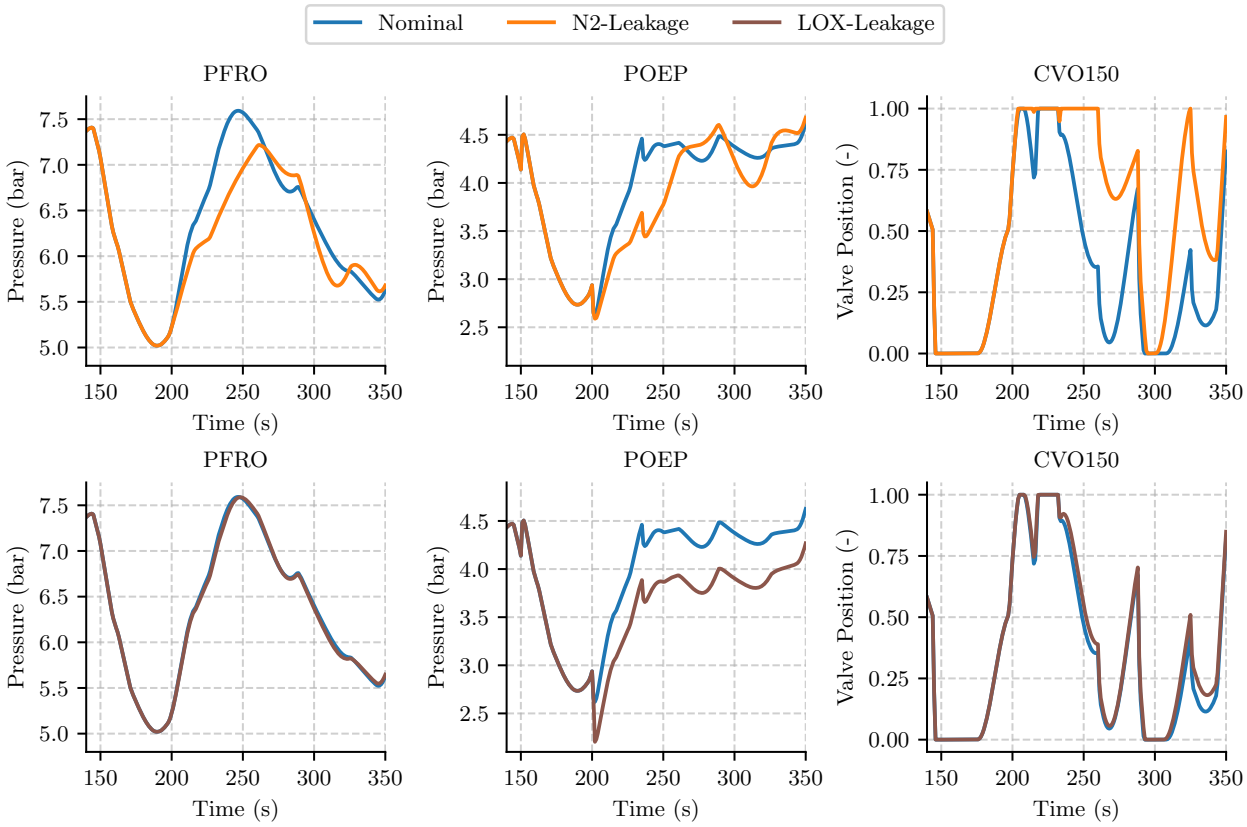

Figure 7 – Simulated system faults for the generic test sequence. The faults occur at $t = 200\,\mathrm{s}$.

In addition to the sensor faults, two representative system faults are also considered. For this we simulate a leak in the LOX or the N2 line with various leakage rates. The mean leakage mass flow rates are given by $22.7\,\mathrm{kg\,s^{-1}}$ and $1.9\,\mathrm{kg\,s^{-1}}$ for a LOX leak and an N2 leak respectively.

All different fault types are introduces into the AI-based generic test sequence generated in section 5. In all cases, the fault occurs at the time $t = 200\,\mathrm{s}$. Figure 6 shows the development of PFRO in the event of sensor faults. Figure 7 shows the effects of a simulated system fault. Characteristic of a LOX leakage is the change of the pump inlet pressure POEP due to the increased pressure drop in the LOX line. In the variables used, an N2 leakage cannot be identified in the same straightforward way and the effects are more complex as can be seen in the first row of the figure.

6.4 Results and discussion

In the following section, first results for AI-based detection of the discussed faults are presented. Two different approaches that only use anomaly-free data are evaluated. In one approach, the training data are used to construct a data-based forward model and the prediction error provides a suitable anomaly score. In the other approach, an autoencoder is trained and the reconstruction error is used as an anomaly score.

In particular, we look at whether the sensor and system faults shown in Figure 6 and 7 respectively are detected.

For both models, a suitable threshold must be defined above which a data point is labeled as belonging to an anomaly, i.e. a fault has occurred. There are several ways to set such thresholds. One possibility is to look at the error distribution for a data set of anomaly-free data and define the threshold such that either none or e.g. only 1 % of the points are labeled as errors. However, if one also has data points available that belong to anomalies, whether they are used for training or not, one can tune the thresholds to achieve the desired detection statistics.

The true-positive rate (TPR) is given by the number of true positives, i.e. the number of data points correctly labeled as belonging to the positive class (sensor or system fault), divided by the total number of data points that belong to the positive class. The false-positive rate (FPR) is given by the number of false positives, i.e. the number of data points incorrectly labeled as belonging to the positive class, divided by the total number of data points that belong to the negative class (anomaly-free test bench operation in our case). A high FPR is equivalent to a large number of false alarms.

In our case, we use the constructed data set of synthetic faults to tune both fault detectors to achieve an FPR of 1 %. Additionally we include the receiver oper-

ating characteristic (ROC) curves which can be used to illustrate the diagnostic ability of a detection logic as its discrimination threshold is varied. It is created by plotting the TPR against the FPR at various threshold settings. Furthermore, the area under the ROC curve, known as AUC, is often used to compare different methods.

6.4.1 Forward model

The forward model is trained to predict the values of PFRO, POEP and the position of CVO150 10 s into the future. As input it additionally receives the values of these variables of the last 10 s, the current values of the tank level and the ventilation as well as the past and future values of the pressurization and LOX mass flow sequences.

Due to the control logic with the PI controllers, it is essential to include past values in the input. This makes it possible to implicitly estimate the values of the I-terms and thus predict the development of the pressures over time. We use a random search and a small validation set to find the best hyperparameters of a fully-connected feedforward neural network that is used for forecasting. As anomaly score we employ the mean squared error of the scaled, i.e. removed mean and scaled to unit variance, predictions.

Figure 8 shows the time evolution of the anomaly score for the discussed generic test sequence and different simulated faults occurring at the time $t = 200\,\mathrm{s}$. It is reassuring to see that no fault is predicted in the nominal sequence. In general, the simulated sensor faults are well detected. Only with sensor drift it takes quite a long time for the threshold to be exceeded. The system faults show a mixed picture. while the LOX leakage is detected perfectly, the N2 leakage is not persistently recognized.

6.4.2 Autoencoder

The autoencoder is trained to reconstruct current and past values of PFRO, POEP, the position of CVO150 and the ventilation valve AVX141, the tank level as well as the pressurization and LOX mass flow sequences. Here, too, we employ a simple fully connected feedforward architecture and optimize the hyperparameters by using a small validation set. The mean squared error of the scaled reconstructions serves as anomaly score.

Figure 9 shows again the time evolution of the anomaly score for the same generic test sequence and simulated faults. Both the sensor faults and the LOX leakage are detected very well with the help of the autoencoder model. However, the simulated N2 leakage is practically not detected at all. A full investigation of the causes is not yet complete, but it appears that the model has not learned any temporal correlations due to the chosen compression and therefore cannot recognize anomalies which are only characterized by changed temporal sequences like the N2 leakage in our case.

Table 1 – Comparison of the true-positive rates and area under the ROC curves for the forward model and autoencoder. The true-positive rates are calculated for fixed thresholds belonging to a false-positive rate of 1 % for the combined fault data set.

Fault Type	Forward Model		Autoencoder	
	TPR*	AUC	TPR*	AUC
All Faults	0.70	**0.95**	**0.71**	0.85
Sensor Offset	**1.00**	**1.00**	**1.00**	**1.00**
Sensor Drift	0.85	**0.98**	**0.89**	**0.98**
Sensor Frozen	0.64	0.86	**0.66**	**0.87**
N2-Leakage	**0.29**	**0.94**	0.01	0.36
LOX-Leakage	0.74	0.99	**0.99**	**1.00**

* for $\mathrm{FPR} = 0.01$

6.4.3 Comparison

Both approaches for AI based fault detection achieve promising results as shown for the TPRs and AUCs for all different fault types in Table 1. The forward model and the autoencoder perform equally good in case of the sensor faults. The autoencoder is the superior model in case of LOX leakage, while the forward model outperforms the autoencoder for the N2 leakage. The latter is the reason for the overall lower AUC of the autoencoder as can be seen in Figure 10.

7. CONCLUSION AND OUTLOOK

The automatically generated pressurization sequences achieve a high control quality of the pump inlet pressure in the evaluated four test runs. Since the AI algorithm is only allowed to change the set point for pressurization of the tank at specific times, the resulting control sequences remain interpretable and verifiable for humans. According to test bed operators, the EcosimPro graphical user interface and associated simulation model will help make test preparation more efficient and accurate in the future.

One next logical step is to replace the EcosimPro model, which is based on physical differential equations with a data-based model. This model will further decrease the necessary calculation time and offer even larger reduction of cadence time. It could also be investigated whether the existing control logic could be replaced entirely by directly controlling the pump inlet pressure of the engine instead of taking the detour via the tank pressure control. The EcosimPro simulation model developed in this study could be a starting point for studying advanced control strategies (reinforcement learning or model predictive control) for this purpose.

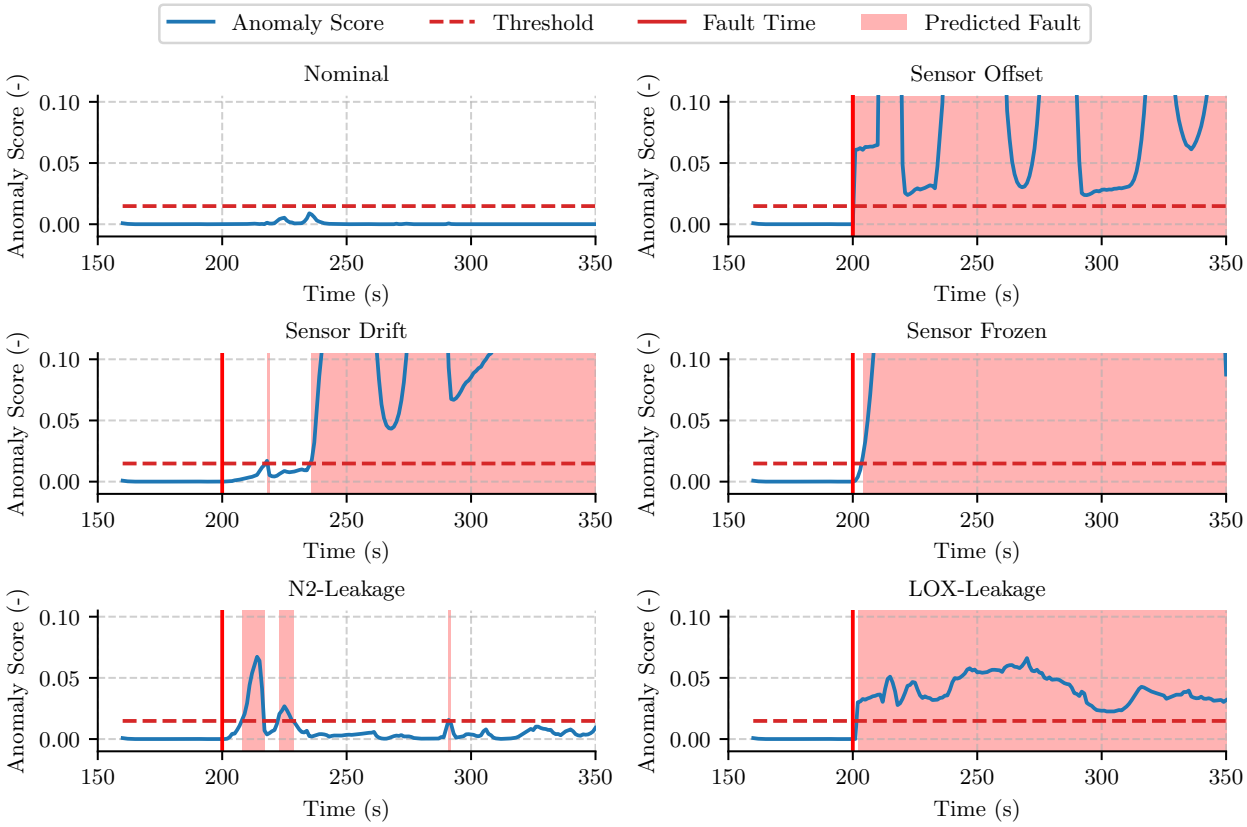

Figure 8 – Fault detection capability of the forward model for the generic test sequence

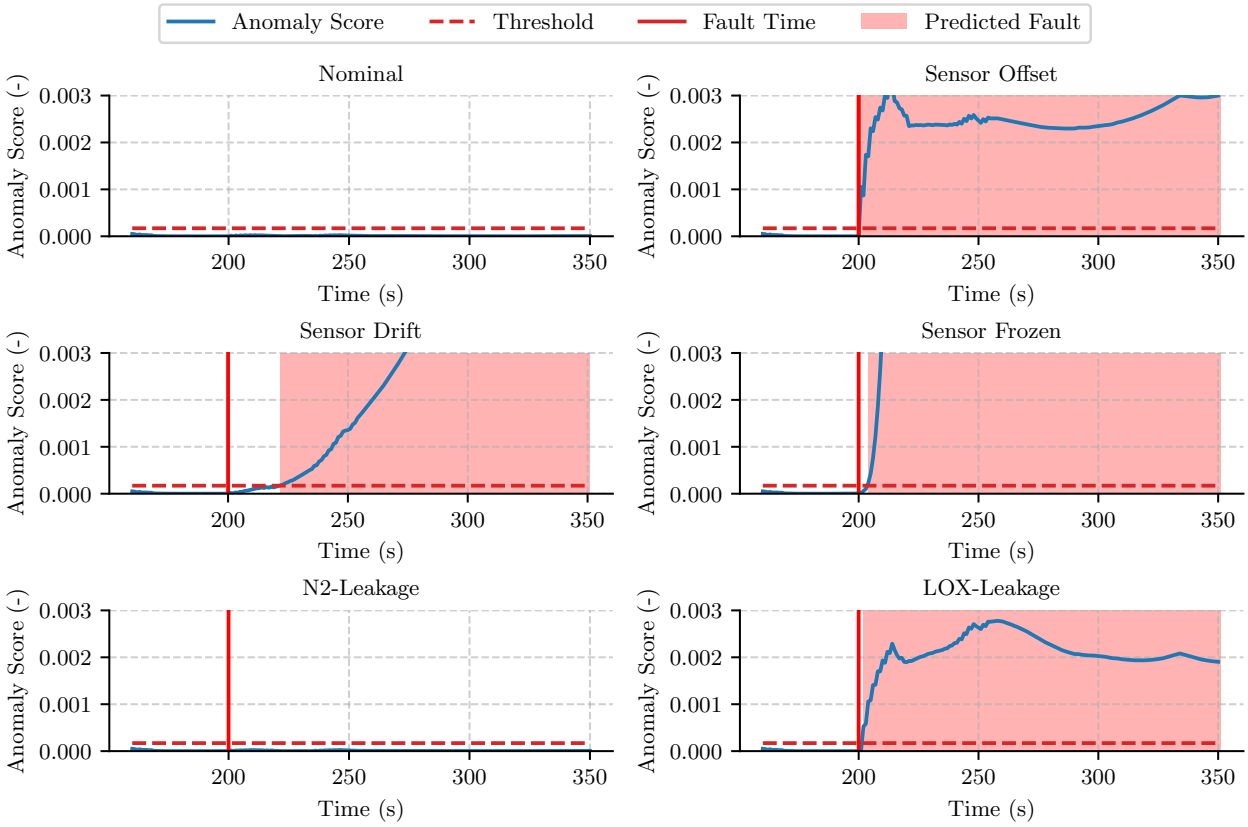

Figure 9 – Fault detection capability of the autoencoder for the generic test sequence

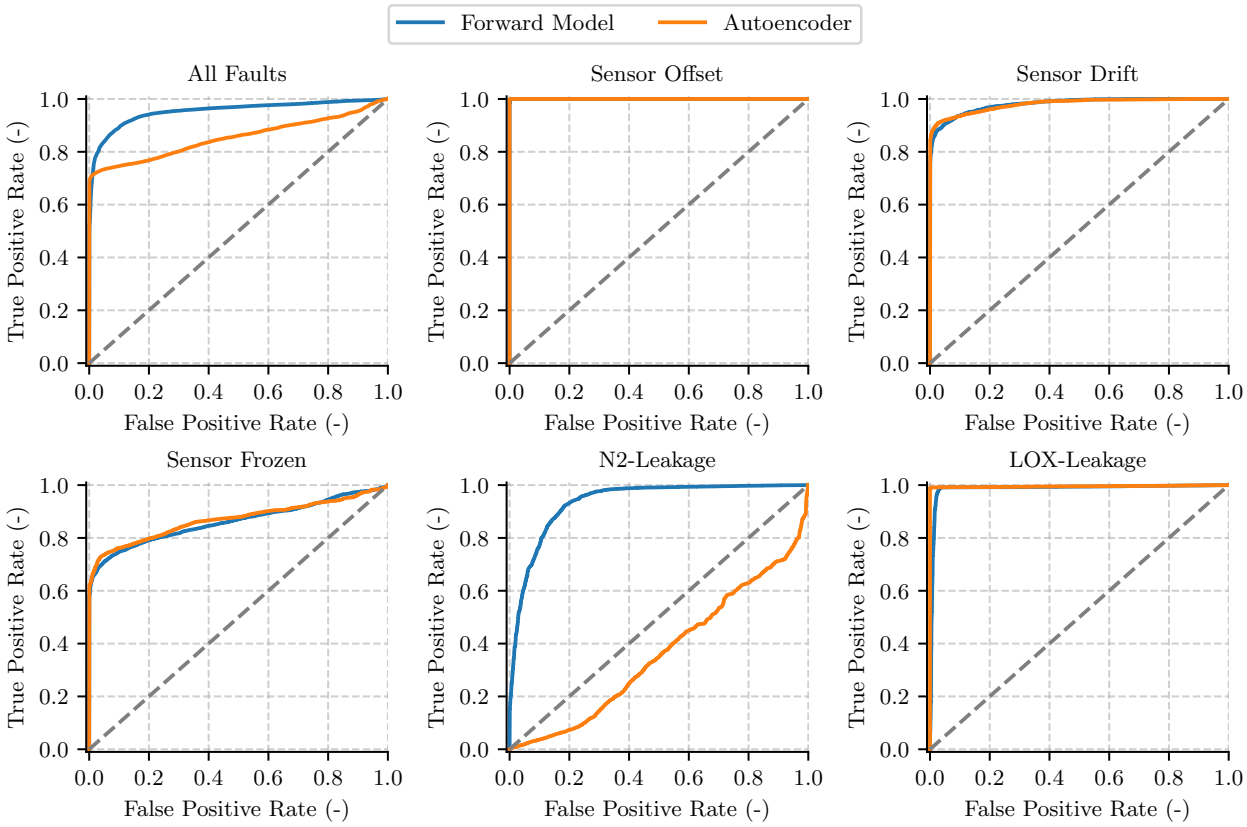

Figure 10 – ROC curves for the simulated fault cases of both models

The next steps in terms of AI-based fault detection are the following: First, the inclusion of sensor noise. Second, evaluation of more complex machine learning architectures like convolutional or recurrent autoencoder. Third, implementation of the AI-based detection logic at the test bench. In addition, the potential of AI for fault isolation will also be quantitatively investigated.

References

[1] S.-Y. Park and J. Ahn, "Deep neural network approach for fault detection and diagnosis during startup transient of liquid-propellant rocket engine," *Acta Astronautica*, vol. 177, pp. 714–730, 2020.

[2] A. Patureau de Mirand, J.-M. Bahu, and O. Gogdet, "Ariane Next, a vision for the next generation of Ariane Launchers," *Acta Astronautica*, vol. 170, pp. 735–749, 2020.

[3] 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.

[4] 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.

[5] S. L. Brunton, J. Nathan Kutz, K. Manohar, A. Y. Aravkin, K. Morgansen, J. Klemisch, N. Goebel, J. Buttrick, J. Poskin, A. W. Blom-Schieber, T. Hogan, and D. McDonald, "Data-Driven Aerospace Engineering: Reframing the Industry with Machine Learning," *AIAA Journal*, pp. 1–26, 2021.

[6] Empresarios Agrupados and ESA, "ESPSS EcosimPro Libraries User Manual," 2019.

[7] K. Dresia, G. Waxenegger-Wilfing, R. H. Dos Santos Hahn, J. Deeken, and M. Oschwald, "Nonlinear Control of an Expander-Bleed Rocket Engine using Reinforcement Learning," in *Space Propulsion 2020+1 Conference*, (Virtual Event), 2021.

[8] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Adaptive Computation and Machine Learning Series, Cambridge, MA: The MIT Press, 2018.

[9] T. Haarnoja, A. Zhou, K. Hartikainen, G. Tucker, S. Ha, J. Tan, V. Kumar, H. Zhu, A. Gupta, P. Abbeel, and S. Levine, "Soft Actor-Critic Algorithms and Applications," *arXiv:1812.05905 [cs, stat]*, 2019.

[10] E. Liang, R. Liaw, P. Moritz, R. Nishihara, R. Fox, K. Goldberg, J. Gonzalez, M. I. Jordan, and I. Stoica, “Ray RLlib: A framework for distributed reinforcement learning,” 2017.

[11] G. Waxenegger-Wilfing, K. Dresia, J. Deeken, and M. Oschwald, “A Reinforcement Learning Approach for Transient Control of Liquid Rocket Engines,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 57, no. 5, pp. 2938–2952, 2021.

[12] L. Ruff, J. R. Kauffmann, R. A. Vandermeulen, G. Montavon, W. Samek, M. Kloft, T. G. Dietterich, and K.-R. Muller, “A unifying review of deep and shallow anomaly detection,” *Proceedings of the IEEE*, vol. 109, no. 5, pp. 756–795, 2021.

[13] G. Pang, C. Shen, L. Cao, and A. V. D. Hengel, “Deep learning for anomaly detection,” *ACM Computing Surveys*, vol. 54, no. 2, pp. 1–38, 2022.

[14] 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,” *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 31, no. 6, 2021.