Nonlinear Control of an Expander-Bleed Rocket Engine using Reinforcement Learning
Virtual Conference 2021

Kai Dresia[1,3], Günther Waxenegger-Wilfing[1], Robson dos Santos Hahn[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: kai.dresia@dlr.de*

KEYWORDS: engine control, LUMEN, machine learning, liquid rocket engine

ABSTRACT:

Intelligent engine control could be one of the most important innovations in the development of future reusable engines, facilitating a safer and more economical engine operation. In this work, we investigate the closed-loop control of the LUMEN expander-bleed engine by combining machine learning with a transient simulation environment. The controller can dynamically change the set-point of the engine between a chamber pressure of 40 bar to 80 bar by adjusting up to six flow control valves while maintaining several boundary conditions at any given time.

1. INTRODUCTION

There is considerable interest in advanced control algorithms for future liquid propellant rocket engines. Advanced engine control offers the possibility for a safer and more economical engine operation and enables more complex mission scenarios, e.g. deep throttling for the landing of first stages. For future reusable engines, advanced control aspects such as life-extending control, re-configurable control, and adaptive control gain in importance.

Liquid propellant rocket engines typically control the thrust and mixture ratio (oxidizer to fuel ratio) by adjustable valves. Traditionally, carefully tuned and predefined valve sequences are used [1]. This may be fine for expandable engines that are only flown once and mostly operate at a constant thrust level during the entire mission. In the context of (partially), reusable launch systems more advanced engine control is mandatory [2], for example for deep throttling maneuvers.

The advantages of an appropriate control system have been highlighted already in 1984 by Bellows et al. [3] for the Space Shuttle main engine. Nevertheless, until today most engines still use either open-loop control with predefined valve sequences or closed-loop control, but only close to steady-state conditions [4].

One reason for such rudimentary engine control is the pneumatic flow control valves, which are too inaccurate and slow for sophisticated closed-loop control. The development of an all-electric control system started in the late 1990s partly to address this problem [5]. This concept replaces the pneumatic valves with electrically operated valves and introduces more sophisticated engine controllers. For example, the future European Prometheus engine might have such a system [6].

In terms of classical engine control, PI-based solutions have mostly been used, e.g. in the Space Shuttle main engine to maintain a constant thrust and mixture ratio at steady-state conditions [7, 8]. PI controllers are extensively studied in literature benefiting from a long tradition of control theory. On the other hand, it is strenuous to design them for highly nonlinear systems with multiple inputs or outputs and their performance is limited [9, 10]. Also, PI-based controllers cannot include advanced control goals such as life-extending control. A survey by Perez-Roca et al. [4] provides an overview of various control strategies for liquid rocket engines, including PI, PID, LQR, model predictive control, and robust control methods.

In recent years, model-free reinforcement learning algorithms [12] - a subset of artificial intelligence - have proven very effective in various fields of research [13, 14]. By utilizing a simulation environment, these algorithms can learn the optimal control strategy even for systems with highly non-linear dynamics. The algorithm samples the environment by carrying out exploratory actions and uses the experience to directly learn the optimal control strategy in form of a neural network. Reinforcement learning is extensively studied for robotics and autonomous driving.

In this work, a neural network-based engine controller is studied for the transient control of an expander-bleed liquid rocket engine by combining modern reinforcement learning with the well-validated simulation environment EcosimPro/ESPSS. The test case at hand is the LUMEN [15, 16] (Liquid Upper Stage Demonstrator Engine) engine demonstrator. LUMEN is a modular LOX/LNG breadboard engine with an expander-bleed cycle in the 25 kN thrust class for operation at the

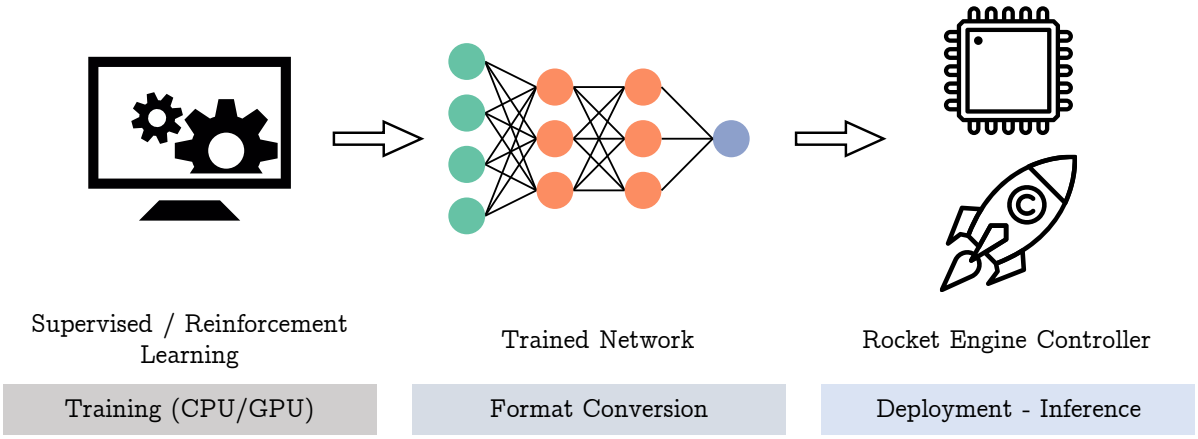


Figure 1: Deployment logic of a neural network controller [11].

new test stand P8.3 in Lampoldshausen [17]. LUMEN is very well suited to investigate advanced control approaches both theoretically and experimentally, as it uses fast and precise electrical valves.

An analogous approach was already studied for a 1000 kN gas-generator engine [18]. The previous study showed the advantages of the neural network controller compared to traditional PI-based controller.

2. REINFORCEMENT LEARNING

Reinforcement Learning [12] attempts to emulate the learning process of intelligent living beings. The central building block is the so-called agent or neural network controller, which tries to solve the task given to it by interacting with its environment. The agent learns a decision rule, also called the policy, which returns a suggested action when given the current state of the system. The action changes the system state, and the agent receives a reward based on the quality of the previously performed action. The goal of reinforcement learning is now to maximize the cumulative reward for a given task. Therefore, a suitable reward function enables the agent to learn complex behaviors without having to be explicitly programmed. Fig. 2 shows the interaction between environment and agent, which is repeated at each time step.

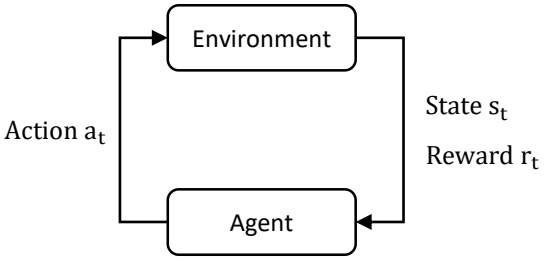


Figure 2: Schematic of reinforcement learning.

To illustrate the concepts of reinforcement learning, let us consider the example of a quadrocopter with the task of hovering at a certain position in space. The state of the system could be described by the current position, attitude, and velocity of the quadrocopter. Rewards could be given for reducing the distance to the target and penalties for energy consumption, forcing the agent to learn an energy-efficient strategy, without explicitly defining how this could be achieved. The agent's policy could yield the optimal rotor speeds as a function of the system state.

The combination of reinforcement learning with deep neural networks as function approximators is called deep reinforcement learning. Here, neural networks are used for representing the policy, allowing the agent to achieve impressive results, such as reaching superhuman performance in the game of Go. Besides sensational results in board games or video games, those algorithms are currently studied in areas like robotics or autonomous driving.

2.1 REINFORCEMENT LEARNING ALGORITHM

Due to the rapid development in reinforcement learning, a large variety of different algorithms with individual strengths and weaknesses exist. Otto et al. [14] compares current state-of-the-art algorithms and discusses their applications. Riedmiller [19] furthermore provides practical guidance on how to apply reinforcement learning to real-world control problems.

In this study, we use the off-policy Soft-Actor-Critic (SAC) algorithm [20], implemented in the Ray RLlib framework [21]. SAC, released in 2019, can handle continuous state and action spaces. Compared with other off-policy algorithms, for example DDPG or TD3, SAC is more stable during training and does not need extensive hyperparameter tuning. Compared with on-policy algorithms like PPO, SAC is more sample efficient, meaning it needs less training data to find the optimal policy. SAC can also be trained in a distributed manner to take advantage of multi-CPU machines, further reducing the training time.

2.2 Deployment and Testing

Training deep reinforcement learning algorithms on real hardware is expensive, and may even not be feasible at all for liquid rocket engines due to safety concerns. Ground tests of the entire propulsion system are extremely expensive and cannot address all nominal and off-nominal behavior that could occur during the mission.

An alternative is to train the controller in a simulated environment and transfer the learned policy afterwards to the on-board embedded computer of the rocket engine. Fig. 1 shows exemplarily how to develop and deploy such a neural network controller. One would train the neural network with a simulation environment on a dedicated workstation. Then, one would convert the neural network to C/C++, and copy it to the embedded computer of the actual space system.

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 libraries. Of particular interest are the European Space Propulsion System Simulation (ESPSS) libraries, which are commissioned by the European Space Agency (ESA).

Finally, EcosimPro models can be combined with typical hardware-in-the-loop simulators to demonstrate the reliability of a neural network-based controller on a space-graded embedded system [11]. This will be an important step in the future to prove the robustness and applicability for real-world applications.

3. LUMEN ENGINE DEMONSTRATOR

The LUMEN engine demonstrator burns LOX/LNG in an expander-bleed cycle. The nominal combustion chamber pressure is 60 bar at a mixture ratio of 3.4. Fig. 3 shows a schematic representation of the engine cycle [15].

For combustion chamber wall cooling, a portion of the LNG mass flow is pumped into the cooling channels in a counter-flow arrangement. The heated cooling mass flow is partially remixed into the main LNG mass flow (mixer) to actively control the LNG injection (INJ) temperature. The remaining cooling mass flow is further heated within the nozzle extension (NEM) and then divided between the LOX and LNG turbines. Afterwards, the turbine exhausts are vented, without being combusted in the main combustion chamber (MCC).

In contrast to flight-like engines, the LUMEN demonstrator as a research platform offers a maximum amount of possibilities for engine control. Electrically actuated flow control valves are used instead of fixed orifices at multiple locations allowing the flexible control of the engine.

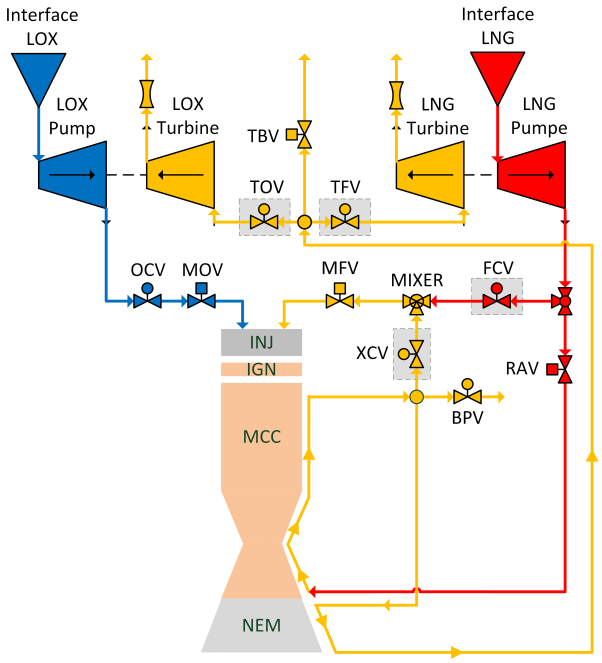


Figure 3: Flow plan of the LUMEN engine architecture.

MOV and MFV are the main oxidizer and fuel valves and remain completely open during testing. TOV and TFV are the turbine oxidizer and fuel valves, which divide the heated LNG between the LOX and LNG turbines. FCV and XCV are placed close to the fuel mixer and allow to independently control the pressure level in the cooling channels and the LNG injection temperature. This valve arrangement allows super-critical conditions within the cooling channels while operating the combustion chamber at lower pressure. OCV is the oxidizer control valve.

LUMEN also has two additional bypass valves: BPV allows discharging LNG after the chamber cooling, reducing the mass flow within the nozzle extension. BPV is required when the necessary mass flow for adequate combustion chamber cooling is greater than the combined mass flow for the combustion chamber and turbines. In this case, LNG can be vented; thereby adjusting the mass flow in the nozzle extension to match the turbines' needs. TBV is a safety valve used only in the event of an emergency shutdown to vent the heated methane and reduce the turbine power quickly.

3.1 LUMEN Engine Control

The fundamental control problem of a rocket engine is chamber pressure and mixture ratio control. The goal is to drive the engine to various set-points without overshoot and then to maintain the set-point with a minimal steady-state error. LUMEN's adaptive engine architecture with multiple control valves makes it possible to further adjust the coolant mass flow and pressure level of the regenerative cooling circuit independently from the combustion chamber pressure.

Apart from controlling the chamber pressure and mixture ratio, the engine controller must keep several thermodynamic and mechanical parameters within certain limits to avoid damaging components or inducing combustion instabilities. Tab. 1 lists all the constraints used in this study.

Table 1: Engine constraints.

Parameter	Unit	Minimum	Maximum
$T_{\text{injection, LNG}}$	K	190	–
$p_{\text{turbine, inlet}}$	bar	30	–
$T_{\text{turbine, inlet}}$	K	$f(p_{\text{turbine, inlet}})$	550
n_{turbine}	[–]	–	n_{max}
$T_{\text{wall, chamber}}$	K	–	900
$p_{\text{pump, LNG}}$	bar	–	150

First, LNG must be injected in a gaseous state, which imposes a minimum LNG temperature at the injection head. Second, the supersonic turbines require a minimum inlet pressure and temperature to prevent LNG from condensing in the turbines. Finally, the turbine inlet temperature, the turbine rotational speed, and the combustion chamber wall temperature are limited for mechanical reasons. The combustion chamber wall temperature is estimated with a neural network trained on CFD data [22]. The original neural network is extended to also include curvature effects.

3.2 Reference Trajectory

In the following, the control system has to drive the engine to different operating points, which are specified in a predefined sequence. Fig. 4 shows the reference values for combustion chamber pressure $p_{\text{cc,ref}}$, mixture ratio $\text{MR}_{\text{cc,ref}}$, cooling channel pressure $p_{\text{RC,ref}}$, and cooling channel mass flow $\dot{m}_{\text{RC,ref}}$. For the cooling channel pressure, a band of $p_{\text{cc,ref}} \pm 10$ bar is defined around the reference value, which the controller has to maintain. The goal of the control system is to follow these reference values as precisely as possible.

First, the design operating point of the LUMEN engine should be reached at a combustion chamber pressure of 60 bar, a mixture ratio of 3.4, and a cooling channel mass flow of 2.3 kg s^{-1}. Then, the combustion chamber pressure should be increased to 80 bar while keeping a constant mixture ratio of 3.4. To ensure sufficient cooling, the cooling mass flow must also be increased to 2.5 kg s^{-1}.

At $t = 35$ s, the mixture ratio should be increased to 3.8 while keeping a constant combustion chamber pressure of 80 bar. At $t = 50$ s, the engine should be throttled down to from 80 to 40 bar. At the same time, the mixture ratio should be reduced to 3.0. Cooling channel pressure and cooling mass flow are adjusted as required.

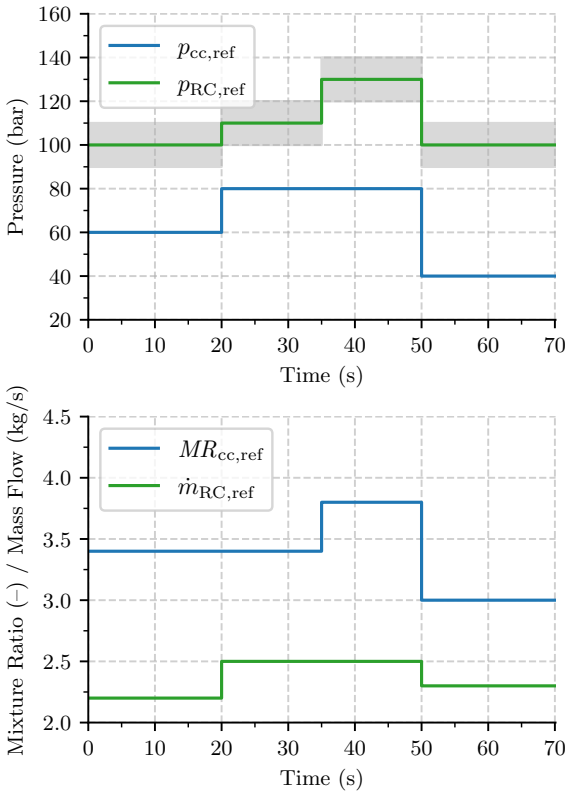


Figure 4: Reference trajectory

3.3 Neural Network Control

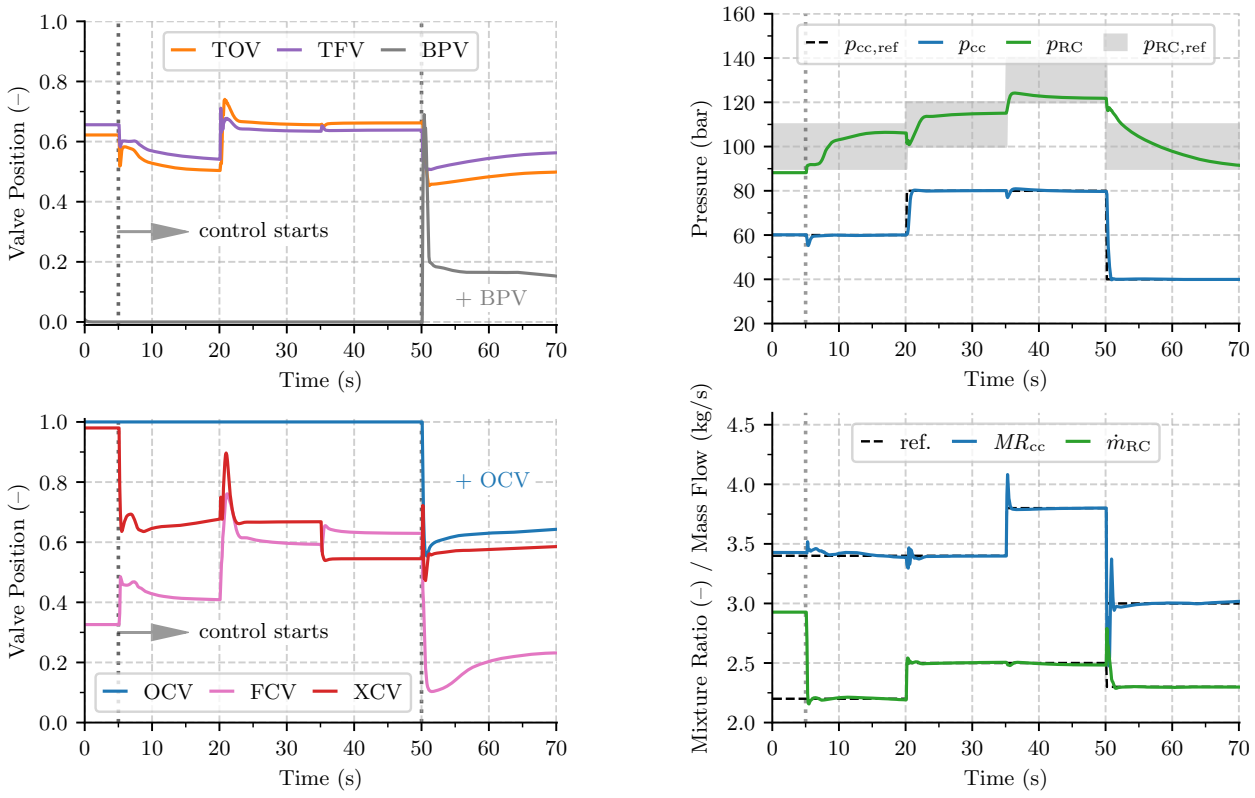
The LUMEN engine control is a multi-input multi-output (MIMO) task, i.e. the controller has to fulfill multiple control goals at the same time by adjusting multiple valves. The control goal is given by matching the reference trajectory for the combustion chamber pressure, mixture ratio, cooling channel mass flow, and the cooling channel pressure with minimal deviations. At all times, the controller must comply with the constraints from Tab. 1.

To train the neural network, one needs to define the observation and action space. The observation space, i.e. the variables the controller receives from the environment at each time step must contain sufficient information to unambiguously define the state of the system. In our set-up, the observation space contains 11 variables:

$$X = [p_{\text{cc,ref}},\ p_{\text{cc}},\ MR_{\text{cc,ref}},\ MR_{\text{cc}},\ p_{\text{RC}},\ T_{\text{inj,LNG}},\ p_{\text{LOX turbine, inlet}},\ p_{\text{LNG turbine, inlet}},\ T_{\text{LNG turbine, inlet}},\ T_{\text{LOX turbine, inlet}},\ \dot{m}_{\text{RC,ref}},\ \dot{m}_{\text{RC}}] \quad (1)$$

Here, p_{RC} is the outlet pressure of the regenerative cooling circuit.

The observation space is normalized with typical steady-state values. All observations are measurable during real engine operation at the test bench, which allows using a similar methodology for future applications of the neural network controller at the test bench.



(a) Valves sequences

(b) Controlled variables

Figure 5: Tracking results of the neural network controller.

The frequency of interaction between the controller and the engine is 10 Hz. At each time step, the controller receives observations from the environment and sends control signals to the flow control valves of the engine.

The objective function, or reward function, is used to train the neural network controller. The reward function calculates a scalar value that is used as a feedback during and it measures whether a state of the environment is considered to be good or bad. The reward function must be defined so that the controller achieves the behavior desired by the human. For the problem of engine control, the reward function assesses how good or bad the controllers follows the given trajectory.

The total reward r at each time step is calculated as follows:

$$reward = r_1 + r_2 + r_3 + r_4 + p \tag{2}$$

Here, r_i, for $i = 1, ..., 4$, are the individual reward components for the controlled variables:

$$r_1 = -\sqrt{\frac{|p_{cc} - p_{cc,\,ref}|}{5\,\text{bar}}} \qquad r_2 = -\sqrt{\frac{|MR - MR_{\,ref}|}{0.1}}$$

$$r_3 = -\sqrt{\frac{|p_{RC} - p_{RC,\,ref}|}{10\,\text{bar}}} \qquad r_4 = -\sqrt{\frac{|\dot{m}_{RC} - \dot{m}_{RC,\,ref}|}{0.5\,\text{kg s}^{-1}}}$$

If the cooling channel pressure is within a band of $p_{cc,ref} \pm 10$ bar around the reference value, r_3 is set to 0. Furthermore, if a constraint from Tab. 1 is violated, the controller receives a penalty of $p = -2n$, where n is the number of violated constraints. Altogether, the reward function allows the controller to trade-off between reaching the desired reference point as fast as possible, avoiding steady-state errors, and minimizing overshoots.

4. Results

This section evaluates the neural network controller (NN) and compares the results against an simple, open-loop (OL) sequence that linearly operates the valves within 0.5 s.

4.1 Neural Network Control

Fig. 5a shows how the neural network controller adjusts the different flow control valves. The neural network controller is activated at $t = 5$ s. Between 5 s and 50 s, the controller can adjust four of the six flow valves (TOV, TFV, FCV, XCV) for closed-loop control. The oxidizer and bypass valve (OCV, BPV) remain completely opened and closed during this period.

At $t = 50$ s, OCV and BPV are also enabled for control. The goal behind these two different control regimes

is to show that the neural network controller can also handle a higher dimensional actuator space, meaning that the controller has to regulate more valves at the same time to achieve its goal. The different control regimes are highlighted with vertical dotted lines in the figure.

In the following, the closed-loop control is evaluated. In general, the neural network controller leverages the dynamics of the LUMEN system for better and faster engine control. For example, at $t = 20\,\mathrm{s}$, the controller opens the valves for a few seconds beyond the final steady-state valve positions to increase the combustion chamber pressure as quickly as possible. The same effect can be observed when the engine is throttled down: In this case, BPV is widely opened for a few seconds to reduce the mass flow to the turbines, which in turn throttles down the engine quickly.

The neural network controller follows the reference trajectory with high precision, while avoiding overshoots and steady-state errors. For both major load point changes from 60 bar to 80 bar and from 80 bar to 40 bar, the neural network controller can adjust the combustion chamber pressure in less than 2 s. One extremely important point is that the controller regulates the cooling channel mass flow quickly and precisely. An insufficient mass flow would lead to high combustion chamber wall temperatures and possibly damage.

The optimal control strategy is often counter-intuitive due to all the system's nonlinearities and constraints. The neural network controller helps to better understand the system dynamics. For example, at $t = 35\,\mathrm{s}$, the controller should change the mixture ratio from 3.4 to 3.8. The most intuitive approach for mixture ratio control is as follows: Increase the mass flow to the fuel turbine via TOV and TFV to increase the power of the fuel turbine and thereby increase the mixture ratio.

However, this control approach via TOV and TFV would violates the constraint for the minimum LNG injection temperature as the system is strongly coupled. Therefore, another control strategy is necessary. The neural network controller uses FCV and XCV for mixture ratio control in this specific case, leaving TOV and TFV essentially unchanged. By altering the pressure level in the cooling channel, the controller increases the power demands on the fuel pump and as a result, increases the mixture ratio.

4.2 Comparison with simple open-loop

The results of the neural network controller are now compared against an simple, naive open-loop (OL) sequence (see Fig. 6) that linearly operates the valves within 0.5 s between the steady-state valve positions.

The results of the neural network controller and the open-loop sequence are compared in Fig. 7. Compared to the open-loop control sequence, the neural

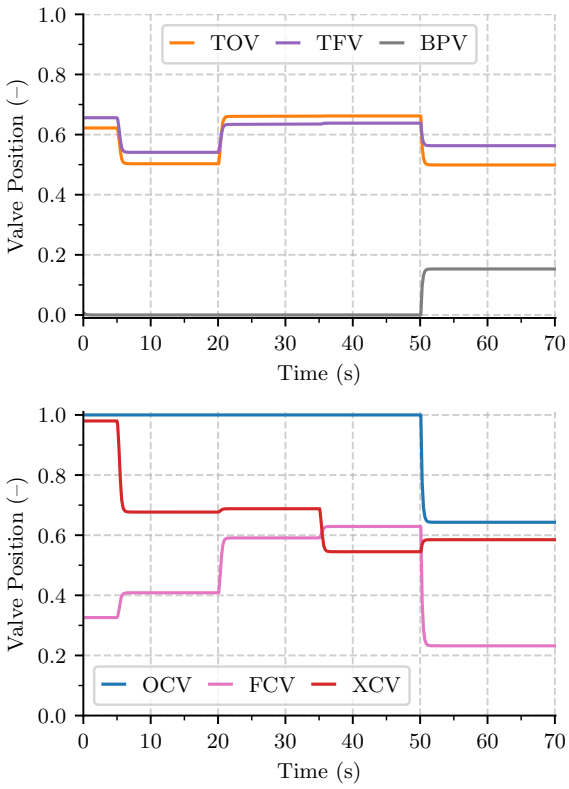


Figure 6: Simple, linear open-loop trajectory.

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

The mixture ratio temporarily exhibits relatively sharp peaks when changing the load point, whereby the peaks for the open-loop are more pronounced. The reason for this is that the neural network controller tries to adjust all target variables as quickly as possible, which means that it accepts short-term peaks. In real engines, damping effects in the combustion chamber and the injection head might substantially mitigate these peaks. Nevertheless, for real-world applications, it is of course necessary to investigate whether there are problems with flame anchoring [23, 24] or combustion instabilities. In this case, the reward function of the reinforcement learning setting can be adjusted in such a way that the controller is slower but avoids short-term peaks.

The cooling channel mass flow shows the shortcomings of simple open-loop control. If the valves are operated linearly, the cooling channel mass flow temporarily can drop significantly and it takes nearly 20 s to

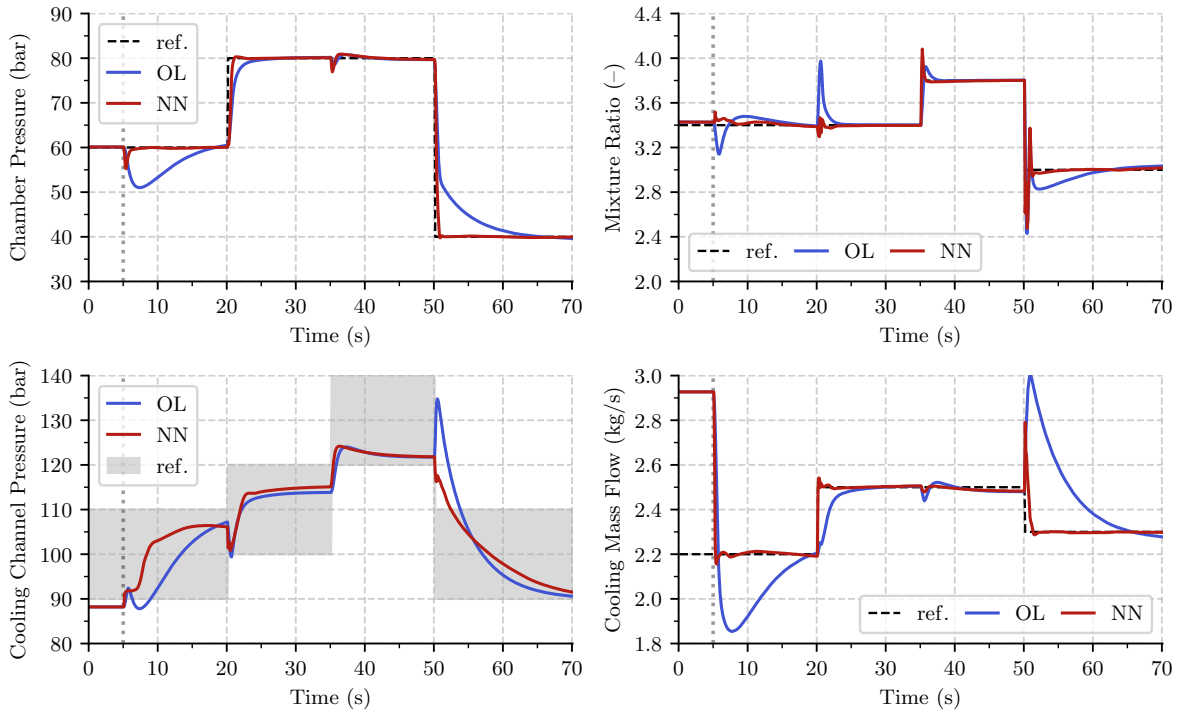


Figure 7: Comparison of the tracking results for the neural network controller and the open loop sequence.

reach steady-state conditions. The insufficient cooling leads to high combustion chamber wall temperatures outside their tolerable specifications [25] and consequently to severe damage of the combustion chamber.

Tab. 2 compares the performance of the neural network controller and the open-loop sequence quantitatively. For this purpose, the summed rewards from equation 2 from $t = 5\,\text{s}$ to $t = 70\,\text{s}$ are compared.

Table 2: Summed reward.

Reward	Variable	OL (–)	NN (–)
$\sum_{t=5\,\text{s}}^{70\,\text{s}} r_1(t)$	p_{cc}	−63	−31
$\sum_{t=5\,\text{s}}^{70\,\text{s}} r_2(t)$	MR_{cc}	−30	−22
$\sum_{t=5\,\text{s}}^{70\,\text{s}} r_3(t)$	p_{RC}	−17	−1
$\sum_{t=5\,\text{s}}^{70\,\text{s}} r_4(t)$	$\dot{m}_{\text{RC}}$	−45	−18
$\sum_{t=5\,\text{s}}^{70\,\text{s}} p(t)$	constraints	−162	0
$\sum_{t=5\,\text{s}}^{70\,\text{s}} [\sum_i r_i(t) + p(t)]$	total	−317	−72

Overall, the neural network controller yields a smaller error for all metrics (smaller summed deviations between reference and current set-point).

4.3 Constraints

As already discussed, all constraints of the engine and its components (see Tab. 1) must be respected at all times to avoid damage or unstable combustion.

Fig. 8 shows that the simple open-loop control violates this requirement. At $t = 5\,\text{s}$, the turbine inlet pressure drops below the allowed minimum value of 30 bar for more than 5 s. The controller, on the other hand, succeeds in complying with all constraints.

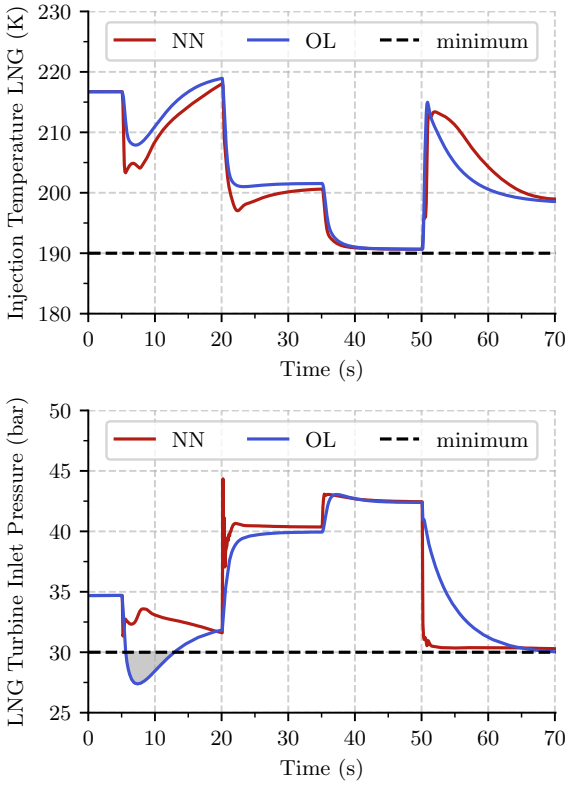


Figure 8: Exemplary engine constraints.

4.4 Sensor noise

Measurements are always subject to sensor noise in real applications at the test bench. To investigate the robustness of the controller, Gaussian noise with a standard deviation of $\pm 1\,\%$ is added to all measurements.

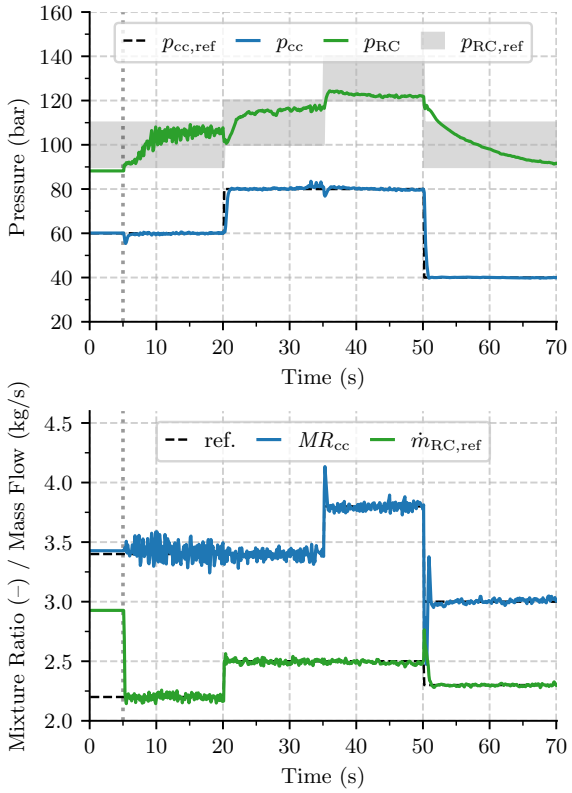


Figure 9: Neural network control with sensor noise.

Fig. 9 shows that the controller provides good tracking performance even in this case. The sensor noise causes only valve jittering, which could be eliminated by a low pass filter in the future.

5. SUMMARY AND OUTLOOK

A neural network-based engine controller was studied for the transient control of an expander-bleed liquid rocket engine, namely the LUMEN engine demonstrator. The controller was trained with a well-validated simulation environment in EcosimPro/ESPSS.

The controller had to follow a given reference trajectory, which specified the combustion chamber pressure, mixture ratio, cooling channel pressure, and coolant mass flow. Compared to a simple open-loop sequence, the neural network controller achieves a much faster tracking response and avoids overshoots in the controlled variables, which could potentially damage the engine.

In the future, it is planned to extend the control task: The controller should optimize the engine for maximum efficiency (*ISP*). Furthermore, the controller should be coupled to a dedicated engine monitoring system. Based on modern machine learning, the detection system can detect the rising of a thermoacoustic instability [26], which allows the engine control to counteract this or at least safely shut-down the engine. Also, life extending control techniques [27] with suitable damage models [28] could increase the engine reusability by considering fatigue life concerns [29].

Additionally, the neural network-based controller will be tested for an existing engine on a test bench at DLR Lampoldshausen in the future. For this real-world application, the robustness of the controller against external disturbances and model uncertainties needs to be further evaluated.

Finally, we will use the existing simulation model to study the steady-state and transient behavior of the LUMEN Engine in more detail.

Acknowledgments

It is a pleasure to thank Michael Börner for discussions about the combustion processes in LOX/LNG flames. We also want to thank Karina Einicke for her analysis of the EcosimPro LUMEN model.

A. SAC Hyperparameters

Tab. 3 lists the SAC parameters that differ from the original paper [20].

Table 3: SAC Hyperparameters

Parameter	Value
discount factor (γ)	0.9
replay buffer size	100 000
total time steps	200 000

References

[1] C. F. Lorenzo and J. L. Musgrave, “Overview of rocket engine control,” in *AIP Conference Proceedings*, vol. 246, (Albuquerque, NM), pp. 446–455, AIP, 1992.

[2] S. Colas, S. L. Gonidec, P. Saunois, M. Ganet, A. Remy, and V. Leboeuf, “A point of view about the control of a reusable engine cluster,” in *8th European Conference for Aeronautics and Space Sciences (EUCASS)*, (Madrid, Spain), 2019.

[3] J. Bellows, R. Brewster, and E. Bekir, “OTV liquid rocket engine control and health monitoring,” in *20th Joint Propulsion Conference*, (Cincin-

nati,OH), American Institute of Aeronautics and Astronautics, 1984.

[4] S. Pérez-Roca, J. Marzat, H. Piet-Lahanier, N. Langlois, F. Farago, M. Galeotta, and S. Le Gonidec, "A survey of automatic control methods for liquid-propellant rocket engines," *Progress in Aerospace Sciences*, vol. 107, pp. 63–84, 2019.

[5] J.-N. Chopinet, F. Lassoudiere, G. Roz, O. Faye, S. Le Gonidec, P. Alliot, and G. Sylvain, "Progress of the development of an all-electric control system of a rocket engine," in *48th AIAA/ASME/SAE/ASEE Joint Propulsion Conference*, (Atlanta, GA), American Institute of Aeronautics and Astronautics, 2012.

[6] P. Simontacchi, R. Blasi, E. Edeline, S. Sagnier, N. Ravier, A. Espinosa-Ramos, J. Breteau, and P. Altenhoefer, "PROMETHEUS: Precursor of new low-cost rocket engine family," in *8th European Conference for Aeronautics and Space Sciences (EUCASS)*, (Madrid, Spain), 2019.

[7] M. Mattox and J. B. White, "Space Shuttle Main Engine Controller," NASA Technical Paper 1932, NASA, 1981.

[8] P. F. Seitz and R. F. Searle, "Space Shuttle Main Engine Control System," SAE Technical Paper 730927, SAE International, Warrendale, PA, 1973.

[9] D. Atherton and S. Majhi, "Limitations of PID controllers," in *Proceedings of the 1999 American Control Conference (Cat. No. 99CH36251)*, (San Diego, CA), pp. 3843–3847 vol.6, IEEE, 1999.

[10] R. Anandanatarajan, M. Chidambaram, and T. Jayasingh, "Limitations of a PI controller for a first-order nonlinear process with dead time," *ISA Transactions*, vol. 45, no. 2, pp. 185–199, 2006.

[11] 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.

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

[13] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[14] F. Otto, "Model-Free Deep Reinforcement Learning—Algorithms and Applications," in *Reinforcement Learning Algorithms: Analysis and Applications* (B. Belousov, H. Abdulsamad, P. Klink, S. Parisi, and J. Peters, eds.), vol. 883, pp. 109–121, Cham, Switzerland: Springer International Publishing, 2021.

[15] T. Traudt, J. C. Deeken, M. Oschwald, and S. Schlechtriem, "Liquid Upper Stage Demonstrator Engine (LUMEN): Status of the Project," in *70th International Astronautical Congress (IAC)*, (Washington D.C.), 2019.

[16] 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.

[17] D. Lindner, D. Suslov, J. Hardi, J. Deeken, G. Brümmer, A. Frank, M. Oschwald, and S. Schlechtriem, "European Research and Technology Test Facility P8.3 for full Cycle Investigations of Subscale Rocket Engines," in *Aeospace Europe Conference*, (Bordeaux, France), 2019.

[18] 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.

[19] M. Riedmiller, "10 Steps and Some Tricks To Set Up Neural Reinforcement Controllers,"

[20] 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.

[21] E. Liang, R. Liaw, R. Nishihara, P. Moritz, R. Fox, K. Goldberg, J. Gonzalez, M. Jordan, and I. Stoica, "RLlib: Abstractions for distributed reinforcement learning," in *Proceedings of the 35th International Conference on Machine Learning* (J. Dy and A. Krause, eds.), vol. 80, (Stockholm Sweden), pp. 3053–3062, PMLR, 2018.

[22] 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.

[23] M. Börner, C. Manfletti, J. Hardi, D. Suslov, G. Kroupa, and M. Oschwald, "Laser ignition of a multi-injector LOX/methane combustor," *CEAS Space Journal*, vol. 10, no. 2, pp. 273–286, 2018.

[24] M. Börner, J. Martin, J. Hardi, J. Deeken, and M. Oschwald, "Flame Anchoring For Shear Coaxial Injectors," in *Space Propulsion Conference 2021*, (Virtual Event), 2021.

[25] 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.

[26] 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.SP]*, 2020.

[27] A. Ray, M. S. Holmes, and C. F. Lorenzo, "Life extending controller design for reusable rocket engines," *The Aeronautical Journal*, vol. 105, no. 1048, pp. 315–322, 2001.

[28] 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.

[29] 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.