Preprints, 10th IFAC International Symposium on
Advances in Automotive Control
Columbus, Ohio, USA, August 28-30, 2022

Fault Diagnosis of Exhaust Gas Treatment System Combining Physical Insights and Neural Networks

Daniel Jung * **Björn Kleman** * **Henrik Lindgren** * **Håkan Warnquist** **

* *Dept. of Electrical Engineering, Linköping University, Sweden.*
(e-mail: daniel.jung@liu.se).
** *Scania CV AB, Södertälje, Sweden*
(e-mail: hakan.warnquist@scania.com)

Abstract: Fault diagnosis is important for automotive systems, e.g., to reduce emissions and improve system reliability. Developing diagnosis systems is complicated by model inaccuracies and limited training data from relevant operating conditions, especially for new products and models. One solution is the use of hybrid fault diagnosis techniques combining model-based and data-driven methods. In this work, data-driven residual generation for fault detection and isolation is investigated for a system injecting urea into the aftertreatment system of a heavy-duty truck. A set of recurrent neural network-based residual generators is designed using a structural model of the system. The performance of this approach is compared to a baseline model-based approach using data collected from a heavy-duty truck during different fault scenarions with promising results.

Keywords: Methods based on neural networks for FDI, Structural analysis and residual evaluation methods, AI methods for FDI, Modeling, supervision, control and diagnosis of automotive systems, Filtering and change detection.

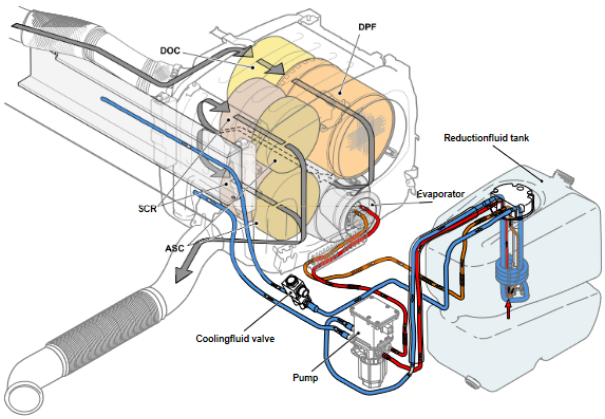

Fig. 1. The exhaust gas aftertreatment system of a Scania heavy-duty truck.

1. INTRODUCTION

One of the main motivations of fault diagnosis in automotive applications is to detect faults in components that could result in increased emissions. One such example is the urea injection in the exhaust aftertreatment system of heavy-duty trucks which is used to reduce NOx emissions, see Fig. 1. With the increasing system complexity and autonomy in modern vehicles, fault diagnosis is also important in new applications, such as predictive maintenance, computer-aided troubleshooting, and self-diagnosis functionality of autonomous systems (Theissler et al., 2021). Two common fault diagnosis approaches are model-based diagnosis and data-driven diagnosis.

In model-based diagnosis, a mathematical model derived from physical insights about the system is used to detect inconsistencies between sensor readings and model predictions due to faults. Developing sufficiently accurate mathematical models of dynamic systems for fault diagnosis is a time-consuming process which has motivated the use of machine learning and data-driven modeling (Qin, 2012). Data-driven fault diagnosis relies on training data from different fault scenarios and operating conditions to classify which fault class that best explains the observed system behavior. However, the performance of a data-driven classifier depends on the quality of training data. Collecting representative data to train such models is complicated by the fact that faults are rare events which means that training data from faults is scarce.

One solution is to use fault-free data to model nominal system behavior to detect anomalies due to faults, e.g., using data-driven residual generators. Even though it is possible to detect faulty behavior, black-box models cannot be used to identify unknown faults since there is no information how different faults will affect the observations. Several researchers have proposed to use recurrent neural networks (RNN) for residual generation, where the network structure is selected based on physical insights. This has been used for fault diagnosis of, e.g., industrial evaporators (Pulido et al., 2019), lithium-ion batteries (Leonori et al., 2021), and internal combustion engines (Jung, 2022). One promising approach is to incorporate physical insights into RNN models to enforce structural properties of the model, see Jung (2019). This can be used to detect and isolate unknown faults that are not represented in the training data by making use of structural information about the system to design residuals that are sensitive to different sets of faults.

This work is based on the results from a master's thesis project performed at Scania CV AB (Kleman and Lindgren, 2021). A diagnosis system is developed for the system illustrated in Fig. 1 by using the Fault Diagnosis Toolbox (Frisk et al., 2017), to automatically design model-based and data-driven residual

generators. A set of RNN-based residual generators is designed using a structural model of the system and the approach is described in Jung (2022).

2. PROBLEM STATEMENT

Developing diagnosis systems can be a time-consuming and expensive process because of extensive model development and the need of collecting data that is representative of the possible realizations of the different fault classes. Hybrid approaches combining model-based and data-driven techniques are attractive since they can take advantage of both physical insights and available training data. The objective in this work is to evaluate the residual generation method proposed in Jung (2022) where a set of RNN-based residual generators are constructed using a structural model of the system. The performance of this approach is compared to a baseline model-based approach. The case study is the system for injecting urea into the aftertreatment system of a heavy-duty truck, see Fig. 1. For the evaluation, data has been collected from a heavy-duty truck on which different fault scenarios have been implemented.

3. BACKGROUND

Before analyzing the case study, some relevant background information regarding model-based diagnosis and RNN-based residual generation is presented.

3.1 Model-Based Diagnosis

Here, a brief introduction to model-based residual generation is given and then how structural analysis methods can be used for fault diagnosability analysis of non-linear systems and design of diagnosis systems.

Residual-Based Fault Diagnosis One useful feature for fault diagnosis is to compare system measurements y with model predictions $\hat{y}$ to compute residuals $r = y - \hat{y}$. A residual generator is a function of known variables (sensor outputs and actuators) that is, ideally, zero in the nominal case and deviates when a fault is present. In practice, fault detection performance is complicated by, e.g., model inaccuracies and sensor noise.

By designing different residual generators to monitor different parts of the system, it is possible to get residual patterns that can be mapped to different faults. When the residual output is insensitive to a specific fault, it is said that the fault is decoupled. The residual patterns can be derived from the mathematical model and summarized in a fault signature matrix, see, e.g., Travé-Massuyès (2014).

To detect significant changes in the residual outputs, different types of change detection algorithms can be used, e.g., the CUSUM (CUmulative SUM) test (Basseville et al., 1993). To handle time-varying noise levels and disturbances, different types of adaptive thresholds can be used that varies with different operating conditions, see, e.g., Pisu et al. (2005).

Structural Methods For non-linear systems, designing residual generators is a complicated task which requires systematic tools to find redundant equation sets. Several algorithms have been proposed for structural models to find redundant equation sets for residual generation. A structural model is a bipartite graph describing the relation between equations and variables in the system model (Frisk et al., 2017). An advantage of structural analysis is that detailed information about analytical models and model parameters is not needed if a structural representation of the system is available.

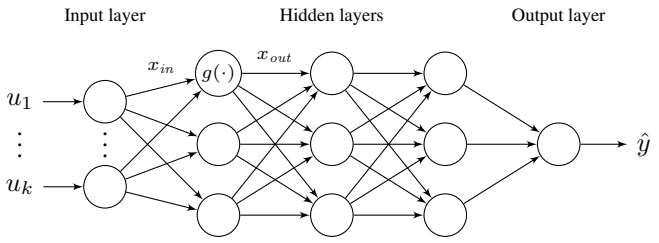

Fig. 2. An illustration of a neural network model structure.

An example of a redundant equation set which has one more equation than unknown variables is called a Minimally Structurally Overdetermined (MSO) set (Krysander et al., 2007). A matching algorithm is used to derive a computational graph for an exactly determined equation set, which is achieved by removing one equation from the MSO set. The computational graph describes in which order to solve the unknown variables in the equation set to compute the predicted variable. Then, the redundant equation can be used as a residual equation. If the analytical relations in the mathematical model are known, the computational graph can be realized as a sequential residual generator, see, e.g., Svärd et al. (2013); Frisk et al. (2017). If the analytical relations or parameter values are unknown, a model-based sequential residual generator cannot be used. However, the computational graph can still be used to design RNN models which is discussed next.

3.2 Residual Generation Using Recurrent Neural Networks

A neural network (NN) is a flexible machine learning model that can capture the behavior of complex non-linear systems (Goodfellow et al., 2016). A NN can be described as a computational graph where the inputs to the nodes in one layer are the outputs from the previous layers. Model inputs are fed to the first layer and the outputs are computed in the final layer, see Fig. 2. Each node represents a non-linear activation function ϕ of the weighted sum of the inputs $x_{in,k}$ to the node $x_{out} = \phi(\sum_k a_k x_{in,k} + b)$ where a_k and b are parameters. In this work, the ReLU activation function $\phi(x) = \max(0, x)$ is used (Goodfellow et al., 2016). For dynamic systems, a special type of neural networks, referred to as recurrent neural networks, are used to model temporal information in time-series data. Conventional NN/RNN models use a general-purpose multi-layered structure and are, in general, data hungry. There is a risk of overfitting if training data is limited and not representative of all relevant operating conditions.

As proposed in, e.g., Jung (2019), deriving computational graphs from a structural model of the system can be used to design RNN structures that capture the structural relations between signals. Thus, the model does not need to learn this from data which reduces the risk of overfitting. Another advantage of deriving the network structure from a structural model is that it is possible to isolate unknown faults, i.e., fault scenarios that are not represented in training data (Jung, 2019). The principle of deriving the RNN model structure is similar to deriving a sequential model-based residual generator but the relations between signals and state variables are modeled in the network structure instead of using the analytical relations.

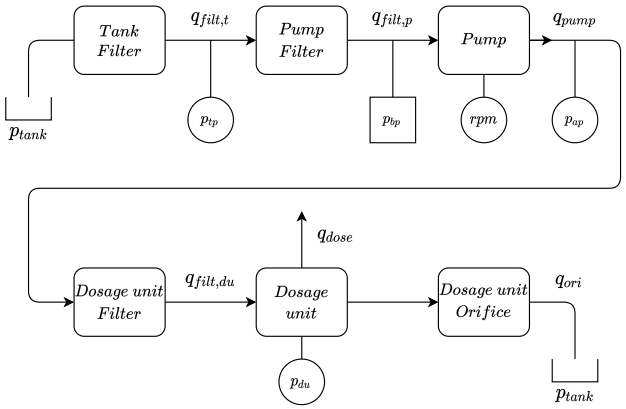

Fig. 3. A schematic of the components in the aftertreatment system model.

The RNN model structures that are considered in this work can be written in the general state-space form

$$\begin{aligned} x_{t+1} &= x_t + Tg(x_t, u_t) \\ y_t &= h(x_t, u_t) \end{aligned} \tag{1}$$

where the dynamics are numerically approximated using Euler forward, T is sampling time, and the functions g and h are approximated by a NN model structure similar to Fig. 2, and the inputs to each function are given by the computational graph. A more detailed description of the methodology that is used here to design the RNN model structure is described in Jung (2022).

4. AFTERTREATMENT SYSTEM

The system used for the experimental setup is part of the exhaust gas aftertreatment system illustrated in Fig. 3. This system is responsible for dosing the right amount of urea into exhausts so that it can react with the harmful NOx gases and form nitrogen and water. The urea is stored in a tank from where it is pumped through a series of hoses to a dosage unit that injects the urea into the exhausts via a nozzle.

4.1 Model

First a mathematical model of the system is developed for the model-based approach based on standard models of the components in Fig. 3. The pump flow is modeled as $q_{pump} = \epsilon_p D_p n_p \eta_{volp}$ [m^3/s], where ϵ_p is the displacement setting $[0, 1]$, D_p is the pump displacement [m^3/rev], n_p is the pump speed [rev/s], and $\eta_{volp} = 1 - C_v \frac{\Delta p}{|\epsilon_p| n_p \eta}$ is the volumetric efficiency $[0, 1]$ where C_v is the laminar leakage losses, Δp the pressure difference over the pump, and η the viscosity.

The flows q_i through the filters in Fig. 3 can be described by the orifice equation as a function of the pressures before and after each filter and the filter's cross-sectional area A_i as

$$q_{filt,t} = C_q A_t \sqrt{(2/\rho)(p_{tank} - p_{bp})} \tag{2}$$

$$q_{filt,p} = C_q A_p \sqrt{(2/\rho)(p_{tp} - p_{bp})}. \tag{3}$$

$$q_{filt,du} = C_q A_{du} \sqrt{(2/\rho)(p_{ap} - p_{du})} \tag{4}$$

$$q_{ori} = C_q A_{ori} \sqrt{(2/\rho)(p_{du} - p_{tank})} \tag{5}$$

The duty cycle (DC) that is used to calculate the dosage flow is calculated as $DC = \frac{q_{dose,req}}{q_{dose,max}}$, that takes the requested dose divided by the maximum dosage flow possible at system pressure. The DC signal is converted to a Pulse Width Modulation (PWM) signal as $PWM_{dose} = f(DC)$ by implementing a PWM conversion function $f()$ in Simulink.

The average dosing flow can be calculated using the orifice equation multiplied by the duty cycle, as

$$q_{dose} = DC \cdot C_q A_{dose} \sqrt{(2/\rho)(p_{du} - p_{exh})} \tag{6}$$

The dynamic of each pressure p_i in Fig. 3, is modeled using the continuity equation as

$$\frac{dp_{tp}}{dt} = \frac{\beta_{tp}}{V_{tp}}(q_{filt,t} - q_{filt,p}) \tag{7}$$

$$\frac{dp_{bp}}{dt} = \frac{\beta_{bp}}{V_{bp}}(q_{filt,p} - q_{pump}) \tag{8}$$

$$\frac{dp_{ap}}{dt} = \frac{\beta_{ap}}{V_{ap}}(q_{pump} - q_{filt,du}) \tag{9}$$

$$\frac{dp_{du}}{dt} = \frac{\beta_{du}}{V_{du}}(q_{filt,du} - q_{ori} - q_{dose}) \tag{10}$$

where β_i is the bulk modulus and V_i the control volume [m^3].

4.2 Modeling of Faults

Typical faults in the aftertreatment system are blockages and leakages. The following faults are considered in the analysis:

- an *actuator fault* in the pump $f_{q,pump}$,
- *blocking or leakages* in hoses after the dosing unit $f_{A,ori}$, in the dosing unit $f_{A,dose}$, before the pump filter $f_{A,t}$, before the pump $f_{A,p}$, and before the dosage unit $f_{A,du}$, and
- *sensor faults* in the pressure sensors at the dosing unit $f_{y,pdu}$, before the pump filter $f_{y,tp}$, and after the pump $f_{y,ap}$.

Each fault is modeled as an additive term in the corresponding model equation.

4.3 Data Collection

Experimental data has been collected from a heavy-duty truck. Different fault scenarios have been evaluated by replacing different components with faulty ones. In each fault scenario, the following signals have been collected:

- three pressure sensors measuring before the pump filter $y_{p,tp}$, after the pump $y_{p,ap}$, and in the dosage unit $y_{p,du}$,
- the converted PWM signal PWM_{dose}, and
- the pump speed n_p.

The training and test data sets consist of approximately 4600 and 2300 samples, respectively, per scenario. Data has been collected from nominal operation NF (No Fault), a blockage before the pump $f_{A,p}$, a blockage after the dosing unit $f_{A,ori}$, a blockage before the dosing unit $f_{A,du}$, and a blockage at the tank $f_{A,t}$.

5. DIAGNOSIS SYSTEM DESIGN

Here, the diagnosis system design process is described including model analysis and residual generation. The fault diagnosability analysis in Section 5.1 is the first step of both the model-based and RNN-based residual generation. The model-based and RNN-based residual generation processes used here are described in Sections 5.2 and 5.3, respectively. In this work,

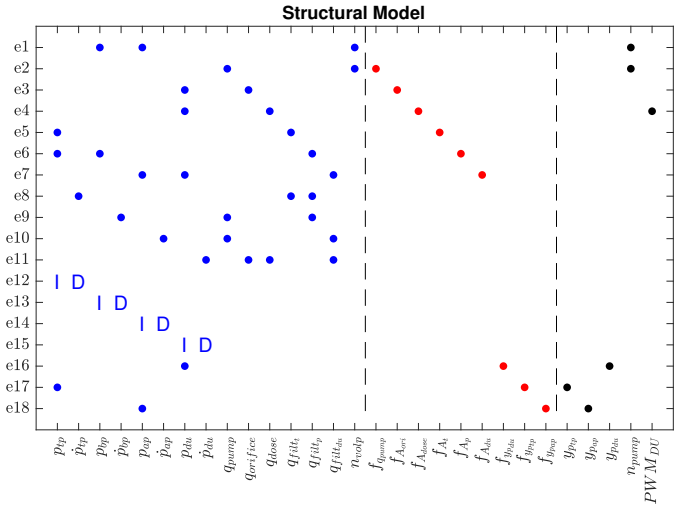

Fig. 4. Structural model of the aftertreatment system.

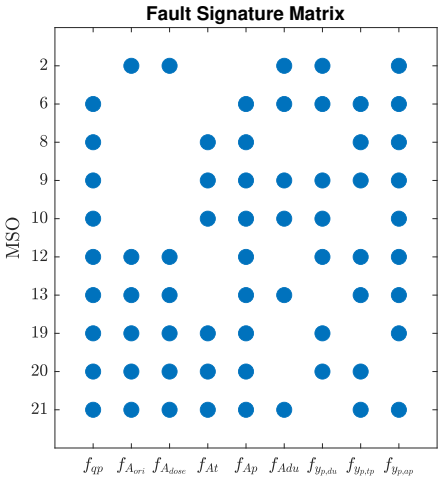

Fig. 5. Fault signature matrix of the selected MSO sets.

only residual generators with integral causality are considered since the current version of the tools used to construct the RNN-models cannot handle differentiation of signals in the network structure (Jung, 2022).

5.1 Fault Diagnosability Analysis

A structural model is formulated based on the component models described in Section 4.1. The structural model is shown in Fig. 4 which has 18 equations, 15 unknown variables including four state variables (marked I) and their derivatives (marked D), 9 faults, and 5 known variables.

From the structural model, 26 MSO sets are found. Out of these, 10 MSO sets are selected that can be used to construct residual generators with integral causality. The fault signature matrix for the selected MSO set describes which faults are affecting any of the equations in each MSO set, see Fig. 5. A residual generator that is based on one of the MSO sets will, ideally, be sensitive to the corresponding faults in the fault signature matrix.

5.2 Model-Based Residual Generation

For each MSO set, one equation is selected as the residual equation and a computational graph is derived from the remaining equations in the MSO set.

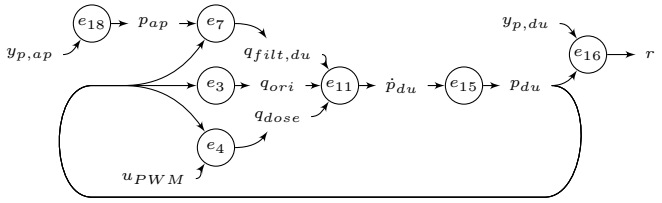

Fig. 6. Computational graph based on MSO_2 when e_{16} is used as residual equation.

For illustration, Fig. 6 shows the computational graph based on the set MSO_2 when e_{16} is selected as residual equation. Similar graphs are derived for the other MSO sets. A set of model-based sequential residual generators is constructed based on the computational graphs for each MSO set and the analytical model equations derived in Section 4.1 using the Fault Diagnosis Toolbox. In the example MSO_2 the residual output r is computed by evaluating the following sequential residual generator in each time step:

$$
\begin{aligned}
e_{18}: &\quad p_{ap} := y_{p,ap} \\
e_7: &\quad q_{filt,du} := \sqrt{2} A_{du} C_q \left| p_{ap} - p_{du} \right| \sqrt{\left| p_{ap} - p_{du} \right| / \rho} \\
e_4: &\quad q_{dose} := \frac{\sqrt{2} A_{dose} C_q PWM_{dose}}{195} \left| p_{du} - p_{exh} \right| \sqrt{\left| p_{du} - p_{exh} \right| / \rho} \\
e_3: &\quad q_{ori} := \sqrt{2} A_{ori} C_q \left| p_{du} - p_{tank} \right| \sqrt{\left| p_{du} - p_{tank} \right| / \rho} \\
e_{11}: &\quad \dot{p}_{du} := -(B_{du}/V_{du}) \left(q_{dose} - q_{filt,du} + q_{ori} \right) \\
e_{16}: &\quad r := p_{du} - y_{pdu} \\
e_{15}: &\quad p_{du} := p_{du} + T\dot{p}_{du}
\end{aligned}
$$

where T is sampling time and the last step (e_{15}) is a numerical integration using Euler forward. A set of sequential residual generators are constructed based on the other MSO sets following the same principles.

5.3 Neural Network-Based Residual Generation

The RNN-based residual generators are derived from the same MSO sets as for the model-based residuals using the same computational graphs. Since the analytical relations are assumed unknown, an RNN model structure is derived based on the dependencies between state variables and known variables by backtracking in the computational graph. As an example, for MSO_2 when e_{16} is selected as residual equation, the resulting recurrent neural network model structure is given as

$$
\begin{aligned}
p_{du,t+1} &= p_{du,t} + Tg(p_{du,t}, y_{p,ap,t}, PWM_{dose,t}) \\
r_t &= h(p_{du,t}) - y_{pdu,t}
\end{aligned}
\tag{11}
$$

where t is used as time index and the state dynamics are evaluated numerically using Euler forward. The functions $g : \mathbb{R}^3 \to \mathbb{R}$ and $h : \mathbb{R} \to \mathbb{R}$ are modeled as NN structures, similar to the structure illustrated in Fig. 2, with three hidden layers, and one scalar output. The number of inputs is given by the inputs to each function in (11). The RNN-based residual generators are automatically implemented in PyTorch (Paszke et al., 2017) using the approach in Jung (2022). Each RNN model is trained on fault-free training data using the ADAM solver (Diederik and Kingma, 2015). One observation from the master's thesis project was that developing the RNN-based residuals was less time-consuming with respect to the model-based residuals.

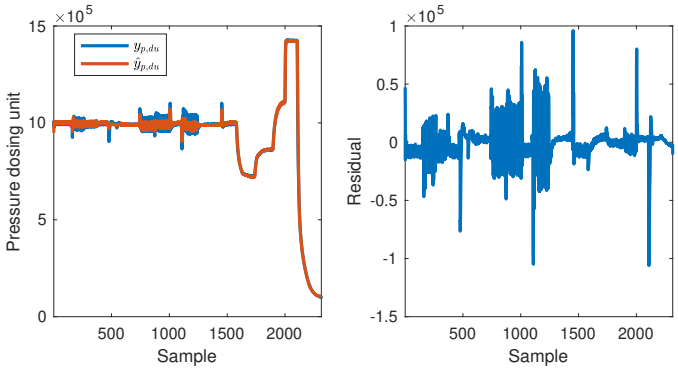

Fig. 7. Left plot show comparison between sensor $y_{p,du}$ and prediction $\hat{y}_{p,du}$ using an RNN model based on MSO_2.

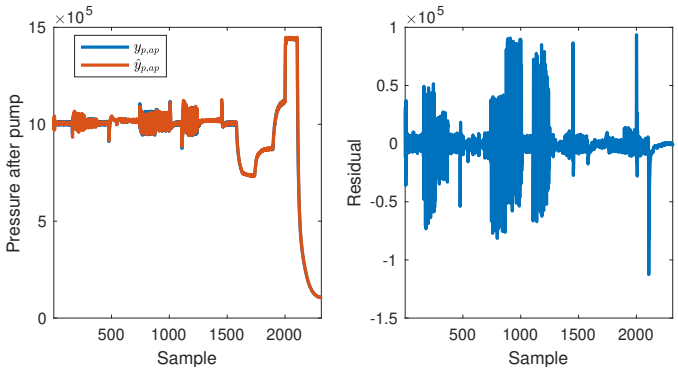

Fig. 8. Left plot show comparison between sensor $y_{p,ap}$ and prediction $\hat{y}_{p,ap}$ using RNN model based on MSO_6.

6. EVALUATION

The resulting predictive performance of the RNN models based on MSO_2 and MSO_6 are shown in Fig. 7 and Fig. 8, respectively. The models capture the general trends in the signals but have difficulties to predict oscillations caused by the dosing unit because of the uncertainties in the PWM_{dose} signal. A more accurate way of modeling the dosing flow would be to include the PWM signal to capture the pump oscillations by replacing the DC with the PWM_{dose} signal in (6).

To evaluate the performance of the RNN models, the residuals are compared to the corresponding model-based residuals derived from the same MSO sets. The parameters of the model-based residuals are optimized using the quasi-Newton algorithm implemented in the Matlab function `fminunc`. The results are shown in Fig. 9 where the model-based residuals achieve better fault-to-noise ratio. Note that the fault $f_{A,p}$ is correctly decoupled for the model-based residual according to the fault signature matrix in Fig. 5. However, this is not the case for the RNN-based residual in the two intervals starting at sample 500 and 1200, respectively. An explanation is that data-driven models do not generalize well. When the fault occurs, the operating conditions deviate from the operating conditions represented in training data, because the blockage affects the pressure levels in the system, which results in false alarms.

To correctly decouple $f_{A,p}$ from the RNN-based residual (MSO_2) in Fig. 5, the first half of the test data set from $f_{A,p}$ is included in the training data. Note that data from fault $f_{A,p}$ still represents nominal system behavior of the residual model since the faulty component is not modeled in the MSO set. The

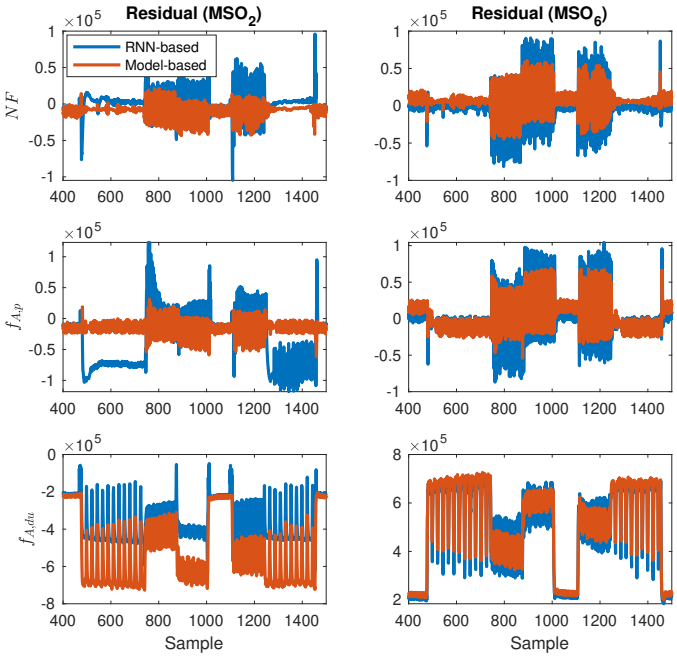

Fig. 9. A comparison between model-based and RNN-based residuals on different fault scenarios. Note that the model-based residual (MSO_2) correctly decouples the fault $f_{A,p}$.

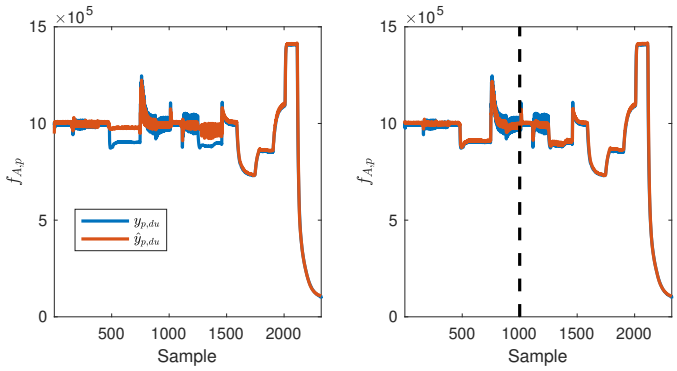

Fig. 10. Comparison of predictive performance when using only nominal training data in the left plot and including data from fault $f_{A,p}$. The right plot show when the data set to the left of the dashed line is included in training data.

RNN model is trained using the expanded training data set. The predictive performance of the original model that is trained using only fault-free data is shown in the left plot in Fig. 10. The updated model is shown in the right plot where data before the dashed line is included in training data. It is visible that the model predictions have improved in the right plot as they better follow the sensor signal compared to the original model.

6.1 Combining CUSUM Test With Adaptive Thresholds

Since there is not enough information in the signals to model the oscillations in the dosage unit, an adaptive threshold is implemented to handle the varying residual noise. The oscillations are correlated with the duty cycle of the PWM signal, and the adaptive threshold is calibrated as a function of the duty cycle using training data. The result in the left plot in Fig. 11 shows the residual output and the adaptive threshold. The right plot shows when the residual is divided with the adaptive threshold to normalize the output.

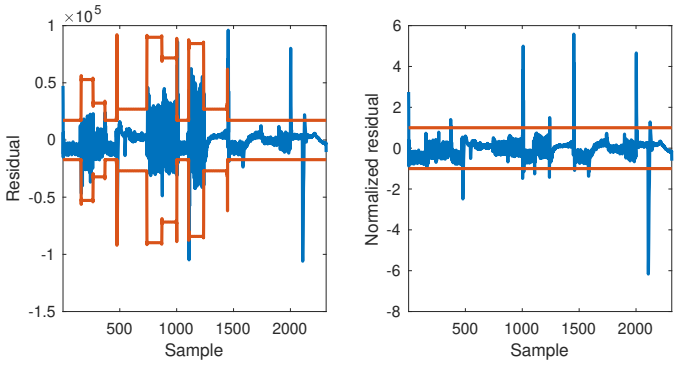

Fig. 11. The left plot shows the residual and the adaptive threshold. The right figure shows the normalized residual.

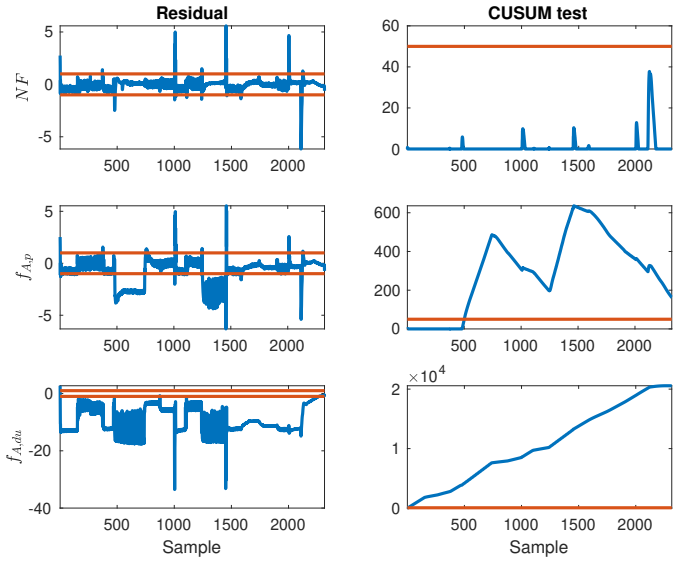

Fig. 12. Evaluation of the CUSUM test for different fault scenarios.

To detect changes in the residual output, a CUmulative SUM test is implemented based on the normalized residual in the form $T_t = \max(0, T_{t-1} + |r_{norm,t}| - 1)$ where $T_0 = 0$, $r_{norm,t}$ is the normalized residual. A fault is detected when $T_t > J$ for some threshold J. Figure 12 shows the normalized residual output of MSO_2 in the left column using data from different fault scenarios and resulting CUSUM tests are shown in the right column. The outputs of the CUSUM test are compared to a threshold chosen as $J = 50$ which shows that the CUSUM test can further improve detection performance of the normalized residuals while reducing the risk of false alarms.

7. CONCLUSIONS

The evaluations from the aftertreatment system case study show promising results when using structural information to design RNN-based residuals for fault diagnosis. As observed in the modeling work, the RNN-based residuals can help reduce development time with respect to the model-based residuals. The fault diagnosis performance of the RNN-based residuals shows the potential with respect to model-based residuals when model information and training data are limited. Still, the quality of training data is important as the predictive performance of data-driven models quickly degrades when the system deviates from the operating conditions represented in training data. Another observation is the ability to take advantage of both model-based and data-driven techniques to improve fault detection performance, for example, combining change detection algorithms, commonly used in model-based diagnosis, with the data-driven residuals to adapt to variations in the residual noise.

REFERENCES

Basseville, M., Nikiforov, I., et al. (1993). *Detection of abrupt changes: theory and application*, volume 104. prentice Hall Englewood Cliffs.

Diederik, J. and Kingma, P. (2015). Adam: A method for stochastic optimization. In *3rd international conference for learning representations, San Diego*.

Frisk, E., Krysander, M., and Jung, D. (2017). A toolbox for analysis and design of model based diagnosis systems for large scale models. *IFAC-PapersOnLine*, 50(1), 3287–3293.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep learning*. MIT press.

Jung, D. (2019). Isolation and Localization of Unknown Faults Using Neural Network-Based Residuals. In *Annual Conference of the PHM Society*, volume 11.

Jung, D. (2022). Automated design of grey-box recurrent neural networks for fault diagnosis using structural models and causal information. In *L4DC Conference*, 8–20. PMLR.

Kleman, B. and Lindgren, H. (2021). *Evaluation of model-based fault diagnosis combining physical insights and neural networks applied to an exhaust gas treatment system case study*. Master's thesis, Linköping University.

Krysander, M., Åslund, J., and Nyberg, M. (2007). An efficient algorithm for finding minimal overconstrained subsystems for model-based diagnosis. *IEEE Trans. Syst. Man Cybern-Part A: Systems and Humans*, 38(1), 197–206.

Leonori, S., Baldini, L., Rizzi, A., and Mascioli, F. (2021). A Physically Inspired Equivalent Neural Network Circuit Model for SoC Estimation of Electrochemical Cells. *Energies*, 14(21), 7386.

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., and Lerer, A. (2017). Automatic differentiation in pytorch. In *NIPS-W*.

Pisu, P., Serrani, A., You, S., and Jalics, L. (2005). Adaptive Threshold Based Diagnostics for Steer-By-Wire Systems. *J Dyn Syst Meas Control*, 128(2), 428–435.

Pulido, B., Zamarreño, J., Merino, A., and Bregon, A. (2019). State space neural networks and model-decomposition methods for fault diagnosis of complex industrial systems. *Engineering Applications of Artificial Intelligence*, 79, 67–86.

Qin, S.J. (2012). Survey on data-driven industrial process monitoring and diagnosis. *Annual reviews in control*, 36(2), 220–234.

Svärd, C., Nyberg, M., Frisk, E., and Krysander, M. (2013). Automotive engine FDI by application of an automated model-based and data-driven design methodology. *Control Engineering Practice*, 21(4), 455–472.

Theissler, A., Pérez-Velázquez, J., Kettelgerdes, M., and Elger, G. (2021). Predictive maintenance enabled by machine learning: Use cases and challenges in the automotive industry. *Reliability engineering & system safety*, 215, 107864.

Travé-Massuyès, L. (2014). Bridging control and artificial intelligence theories for diagnosis: A survey. *Engineering Applications of Artificial Intelligence*, 27, 1–16.