DINAME 2017 - Proceedings of the XVII International Symposium on Dynamic Problems of Mechanics
A. T. Fleury, D. A. Rade, P. R. G. Kurka (Editors), ABCM, São Sebastião, SP, Brazil, March 5-10, 2017

A Neural Network Observer for Injection Rate Estimation in Common Rail Injectors with Nozzle Wear

Oliver Hofmann, Manuel Kiener, and Daniel Rixen

Chair of Applied Mechanics, Technical University of Munich, Germany,
oliver.hofmann@tum.de, manuel.kiener@tum.de, rixen@tum.de

Abstract: The objective of this study is to present a neural observer estimating changing injection behavior due to wear and aging effects within the nozzle of a common rail diesel injector. Using a dynamic identification system in combination with a modified learning rule, the neural observer is applicable to a wide range of problem sets. A multilayer perceptron (MLP) network with three-layers and few neurons in the hidden layer ensures fast computing and high efficiency, and learning is based on quasi-Newton optimization and an additional line search algorithm. Modeling the bottom part of the injector introduces a simulation model, which is validated with experimental data from a solenoid common rail diesel injector. Estimation results conform well with to altered plant and therefore demonstrate the significant benefit of using the proposed neural network observer concept.

Keywords: neural network observer, diesel injector, injection rate estimation, nozzle wear

NOMENCLATURE

Latin symbols

$\boldsymbol{C}$:	system output matrix	m:	number of network outputs
$\boldsymbol{H}$:	Hessian approximation matrix	n:	number of network inputs
$\boldsymbol{I}$:	identity matrix	p:	pressure
$\boldsymbol{L}$:	observer design matrix	q:	number of hidden layer neurons
$\boldsymbol{V}$:	network weight matrix	A:	area
$\boldsymbol{W}$:	network weight matrix	J:	cost function
$\boldsymbol{b}$:	vector of network biases	∇J:	cost function gradient
$\boldsymbol{e}$:	vector of system output errors	$\nabla^2 J$:	cost function Hessian
$\boldsymbol{p}$:	vector of optimization parameters	K:	bulk modulus
$\boldsymbol{u}$:	vector of system inputs	L:	length
$\boldsymbol{x}$:	vector of system states	Q:	volume flow rate
$\boldsymbol{y}$:	vector of system outputs	V:	volume
$\boldsymbol{z}$:	vector of network inputs		

Greek symbols

α:	line search step size
ε:	network output error
μ:	convergence tolerance
ξ:	loss factor
ρ:	density
σ:	network transfer function
ϕ:	observer fault approximation
Φ:	plant fault model

INTRODUCTION

Injection rate significantly influences combustion performance in common rail diesel injectors. Aging effects such as coking or or wear due to cavitation, which develop in the injector nozzle over its lifetime, deteriorate the injection's behavior and result in increasing soot and nitrogen oxide (NO_x) emissions. These effects cause large uncertainties in injection rate estimation, which need to be considered when modeling. Krogerus et al. (2016) present a survey of analysis, modeling, and diagnostics of diesel fuel injection systems, which shows the proposed topic's great relevance. Specific investigations into fault effects such as nozzle wear in diesel injectors, which affect the injection's behavior over its lifetime, are emphasized. Several research studies demonstrate the negative impact of injection faults on combustion behavior, resulting in reduced combustion efficiency and greater emissions. Neural networks are frequently applied to investigate injection behavior and its impacts on combustion performance.

Concepts that combine state observers and neural networks to identify nonlinearities can be found in various publications. Abdollahi et al. (2006) and Talebi et al. (2010) derived a stable neural network observer for coping with unknown system faults. Using multilayer perceptron (MLP) networks, the observer scheme allows an application to nonlinear MIMO systems assuming observability. The learning rule for neural network weight adaption is based on a stable back-propagation training algorithm using Lyapunov's direct method. Hintz (2003) introduced another state observer concept employing radial-basis-function (RBF) networks. Learning rules based on the failure models of Narendra and Annaswamy made an identification of static and dynamic nonlinearities possible. Stringent requirements imposed on the system structure, such as the occurrence of isolated nonlinearities, limit the approach's applicability.

In this paper, a neural network observer scheme to estimate the injection rate under fault caused by aging effects is proposed. The observer's layout uses a neural network model approximation to take additive fault into consideration. In contrast to state observers without adaptive models, the proposed neural network observer can cope with large uncertainties in the estimated states. It therefore ensures a robust estimation behavior considering wear and aging effects within the injector nozzle.

System identification provides information to the observer about the existing network output error. We determine a multilayer perceptron (MLP) feed-forward neural network structure with three layers of neurons, which is trained using a modified quasi-Newton optimization approach to minimize a predefined cost function. A line search algorithm helps to optimize the observer performance adapting the learning step size.

The common rail diesel injector model focuses on the nozzle area, as aging effects are expected to be most evident there. We derive a nonlinear state space model using measurement signals of the rail pressure and needle lift as model inputs and the pressure within the lower feed line as output. Simulation results of the state space model proposed in this study are compared to available measurement data, and show good agreement. Finally, we analyze the performance of the injection rate estimation, comparing the results to simulation data with modified nozzles.

NEURAL OBSERVER DESIGN

We consider the general nonlinear plant model, which is assumed to be observable

$$\begin{aligned} \dot{\boldsymbol{x}} &= \boldsymbol{f}(\boldsymbol{x},\boldsymbol{u}) \\ \boldsymbol{y} &= \boldsymbol{C}\boldsymbol{x} \end{aligned} \tag{1}$$

where $\boldsymbol{u}$ is the input, $\boldsymbol{y}$ is the output and $\boldsymbol{x}$ is the state vector of the system. We suppose the model $\boldsymbol{f}(\boldsymbol{x},\boldsymbol{u})$ with additive fault $\Phi(\boldsymbol{x},\boldsymbol{u})$, which we want to approximate using a neural network model. The nonlinear neural network observer is given by

$$\begin{aligned} \dot{\hat{\boldsymbol{x}}} &= \boldsymbol{f}(\hat{\boldsymbol{x}},\boldsymbol{u}) + \phi(\hat{\boldsymbol{x}},\boldsymbol{u}) \\ \hat{\boldsymbol{y}} &= \boldsymbol{C}\hat{\boldsymbol{x}} \end{aligned} \tag{2}$$

where $\hat{\boldsymbol{x}}$ is the observed state. The neural network approximation $\phi(\hat{\boldsymbol{x}},\boldsymbol{u})$ aims to identify the additive fault occurring within the plant. Consequently, the output error between plant and observer is defined as

$$\boldsymbol{e} = \boldsymbol{y} - \hat{\boldsymbol{y}} \tag{3}$$

The introduced observer system is depicted in Fig. 1. Additive fault $\Phi(\boldsymbol{x},\boldsymbol{u})$ influences the original plant model Eq. (1). The neural observer estimates the derived state vector $\dot{\hat{\boldsymbol{x}}}$ by summing up the results of the nonlinear vector valued function $\boldsymbol{f}(\hat{\boldsymbol{x}},\boldsymbol{u})$ and the neural network output $\phi(\hat{\boldsymbol{x}},\boldsymbol{u})$. The three layer MLP feed-forward neural network is illustrated next to the observer structure. Weight adjustments within the neural observer are based on the network output error ε. The transfer function $\boldsymbol{g}(\boldsymbol{u},\hat{\boldsymbol{x}},\tilde{\boldsymbol{x}},\boldsymbol{e})$ describes the transition of the system error $\boldsymbol{e}$ to the network error ε and is examined below.

The observer's objective is to minimize the network output error defined as

$$\varepsilon = \Phi(\boldsymbol{x},\boldsymbol{u}) - \phi(\hat{\boldsymbol{x}},\boldsymbol{u}) \tag{4}$$

The neural network thereby relies on the information of this specific error to apply a learning algorithm. Further information needs to be supplied because fault on the plant $\Phi(\boldsymbol{x},\boldsymbol{u})$, is an unknown quantity. We introduce a separate dynamic system to identify the system fault:

$$\dot{\tilde{\boldsymbol{x}}} = \boldsymbol{f}(\tilde{\boldsymbol{x}},\boldsymbol{u}) - \boldsymbol{C}^T(\boldsymbol{y} - \boldsymbol{C}\tilde{\boldsymbol{x}}) \tag{5}$$

where $\tilde{\boldsymbol{x}}$ denotes the state vector of the novel identification system. We obtain approximated information about the fault behavior within the plant, which we use for further investigation. Subsequently, comparing the network output to the identified fault provides the required error. After applying mathematical transformations, the network output error is given by

$$\varepsilon = \boldsymbol{L}\left[\dot{\boldsymbol{e}} + \boldsymbol{C}\boldsymbol{f}(\hat{\boldsymbol{x}},\boldsymbol{u}) - \boldsymbol{C}\boldsymbol{f}(\tilde{\boldsymbol{x}},\boldsymbol{u})\right] \tag{6}$$

where $\boldsymbol{L}$ describes an observer design matrix, to assign ε using a gain to the neural network output ϕ. To cope with faults in the physical plant, we model the identified error using neural network approximation. The considered neural network is a multilayer perceptron (MLP) feed-forward network with three layers of neurons, which we define as

$$\phi(z) = \boldsymbol{W}\sigma(\boldsymbol{V}z) + \boldsymbol{b} \qquad\qquad z = \begin{bmatrix}\boldsymbol{x} & \boldsymbol{u} & 1\end{bmatrix}^T \tag{7}$$

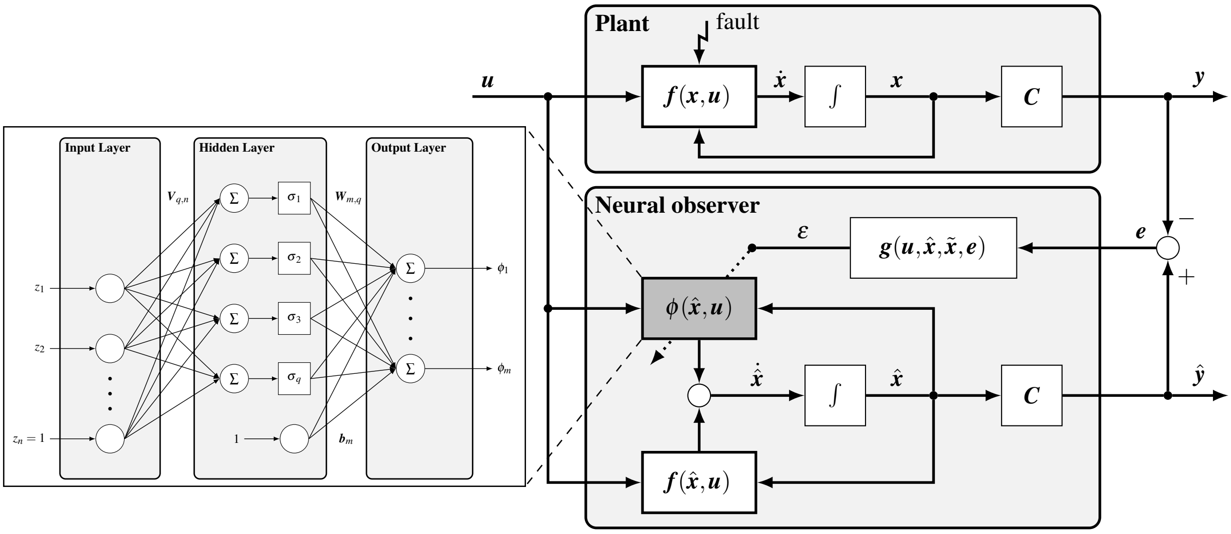


Figure 1 – Neural observer design with three-layer MLP feed-forward network

The configurable weights $\boldsymbol{V}$ and $\boldsymbol{W}$, as well as the bias of the output layer $\boldsymbol{b}$ are given by

$$\boldsymbol{V} = \begin{bmatrix} \boldsymbol{v}_1{}^T \\ \vdots \\ \boldsymbol{v}_q{}^T \end{bmatrix} \in \mathbb{R}^{q \times n} \qquad \boldsymbol{W} = \begin{bmatrix} \boldsymbol{w}_1 & \cdots & \boldsymbol{w}_q \end{bmatrix} \in \mathbb{R}^{m \times q} \qquad \boldsymbol{b} \in \mathbb{R}^{m \times 1} \tag{8}$$

where n is the number of network inputs, m is the number of network outputs, and q is the number of hidden layer neurons. Note, that the bias of the hidden layer is included in the weight matrix $\boldsymbol{V}$. The weight matrices consist of parameters that the learning algorithm determines, and we use a tangent hyperbolic as transfer function

$$\sigma_i = \frac{2}{1 + \exp\left(-2\boldsymbol{v}_i{}^T \boldsymbol{z}\right)} - 1 \tag{9}$$

The training of the network weights and biases aims to minimize the network outputs's remaining approximation error ε. In order to measure the goodness of the fit between identified fault data and network output, we use a real time recurrent learning algorithm (Williams and Zipser, 1989) for the observer application. Unlike during conventional back-propagation, the network parameters are not updated while sweeping through the training data set from t_0 to t_{end}. Therefore, the cost function J characterizes the total sum-squared error of all time samples in the network output (Nelles, 2001)

$$J = J(t_0) + J(t_1) + \cdots + J(t_{end}) = \sum_{t=0}^{t_{end}} \frac{1}{2} \varepsilon^T \varepsilon \tag{10}$$

The objective of network training is to minimize the cost function without constrains $\min_{\boldsymbol{p}} J(\boldsymbol{p})$, which results in optimality conditions for the problem

$$\nabla J(\boldsymbol{p}^*) = \boldsymbol{0} \qquad \text{and} \qquad \nabla^2 J(\boldsymbol{p}^*) \geq \boldsymbol{0} \tag{11}$$

where $\bullet^*$ denotes the parameter set with optimal conditions and all network parameters to be optimized are arranged within the parameter set

$$\boldsymbol{p} = \begin{bmatrix} \boldsymbol{v}_1{}^T & \cdots & \boldsymbol{v}_q{}^T & \boldsymbol{w}_1{}^T & \cdots & \boldsymbol{w}_q{}^T & \boldsymbol{b}^T \end{bmatrix}^T \tag{12}$$

The cost function gradient, $\nabla J(\boldsymbol{p})$, of the given optimization problem is derived by applying the chain rule and results in

$$\nabla J(\boldsymbol{p}) = \sum_{t=0}^{t_{end}} \frac{\partial J}{\partial \boldsymbol{p}} = \sum_{t=0}^{t_{end}} \left(\frac{\partial J}{\partial \varepsilon} \cdot \frac{\partial \varepsilon}{\partial \phi} \cdot \frac{\partial \phi}{\partial \boldsymbol{p}} \right) = \sum_{t=0}^{t_{end}} \begin{bmatrix} -\frac{\varepsilon^T \boldsymbol{w}_1 (\sigma_1 + 1)^2}{\exp(2\boldsymbol{v}_1 \boldsymbol{z})} \boldsymbol{z} \\ \vdots \\ -\frac{\varepsilon^T \boldsymbol{w}_q \left(\sigma_q + 1\right)^2}{\exp\left(2\boldsymbol{v}_q \boldsymbol{z}\right)} \boldsymbol{z} \\ -\varepsilon_1 \sigma \\ \vdots \\ -\varepsilon_m \sigma \\ -\varepsilon \end{bmatrix} \tag{13}$$

To solve the optimization problem, we iteratively update the parameters using a line search approach with search direction $\Delta\boldsymbol{p}^k$ and step size α^k

$$\boldsymbol{p}^{k+1} = \boldsymbol{p}^k + \alpha^k \Delta\boldsymbol{p}^k \tag{14}$$

The search direction is obtained by approximating the cost function using a quadratic model from the second-order Taylor series (Nocedal and Wright, 2006)

$$J(\boldsymbol{p}+\Delta\boldsymbol{p}) \approx J(\boldsymbol{p}) + \Delta\boldsymbol{p}^T \nabla J(\boldsymbol{p}) + \frac{1}{2}\Delta\boldsymbol{p}^T \nabla^2 J(\boldsymbol{p})\Delta\boldsymbol{p} \tag{15}$$

which results in the so called Newton search direction, $\Delta\boldsymbol{p}^k$, assuming that the Hessian $\nabla^2 J(\boldsymbol{p}^k)$ is positive definite

$$\Delta\boldsymbol{p}^k = -\left[\nabla^2 J(\boldsymbol{p}^k)\right]^{-1} \nabla J(\boldsymbol{p}^k) \tag{16}$$

As direct calculation of the Hessian is numerically expensive, an approximation of the Hessian, $\boldsymbol{H}^k \approx \nabla^2 J(\boldsymbol{p}^k)$, is often used. The very common BFGS updating method proposed by Broyden, Fletcher, Goldfarb, and Shanno is given by

$$\boldsymbol{H}^{k+1} = \boldsymbol{H}^k - \frac{\boldsymbol{H}^k \boldsymbol{d}^k (\boldsymbol{d}^k)^T \boldsymbol{H}^k}{(\boldsymbol{d}^k)^T \boldsymbol{H}^k \boldsymbol{d}^k} + \frac{\boldsymbol{v}\boldsymbol{v}^T}{\boldsymbol{d}^T \boldsymbol{v}} \tag{17}$$

where

$$\boldsymbol{v} = \boldsymbol{p}^{k+1} - \boldsymbol{p}^k \qquad \boldsymbol{d} = \nabla J(\boldsymbol{p}^{k+1}) - \nabla J(\boldsymbol{p}^k)$$

The step size is obtained using the Amijio backstepping algorithm. Starting with an initial value $\alpha^k = \alpha_0^k$, the step size is reduced to satisfy the inexact line search condition for sufficient decrease with the constant value $\rho \in (0,1)$

$$J(\boldsymbol{p}^k + \alpha^k \Delta\boldsymbol{p}^k) < J(\boldsymbol{p}^k) + \rho\alpha^k \nabla J(\boldsymbol{p}^k)\Delta\boldsymbol{p}^k \tag{18}$$

The complete algorithm used for the neural network training is summarized in Alg. 1. We initialize the network weights, as well as the Hessian approximation at $k = 0$ using scaled identity $\beta\boldsymbol{I}$ and $\gamma\boldsymbol{I}$, respectively. Furthermore, the convergence tolerance is set to $\mu > 0$, which is used as the optimization abort criterion. In each iteration, search direction $\Delta\boldsymbol{p}^k$ and step size α^k are computed according to Newton direction with Hessian approximation, Eq. (16), and backstepping method to satisfy the Amijio condition, Eq. (18). The parameter set, $\boldsymbol{p}^{k+1}$, is updated using line search according to Eq. (14) and the Hessian, $\boldsymbol{H}^{k+1}$, of the next iteration is approximated using the BFGS algorithm, Eq. (17). When reaching the convergence tolerance, μ, the optimal network weights are calculated from the optimal parameter set, $\boldsymbol{p}^*$.

Algorithm 1: neural network learning for fault identification

Input : $\boldsymbol{p}^0$;	// Initial network weights $\boldsymbol{V}^0 = \beta\boldsymbol{I}, \boldsymbol{W}^0 = \beta\boldsymbol{I}$, and $\boldsymbol{b}^0 = \beta\boldsymbol{I}$
$\boldsymbol{H}^0 = \gamma\boldsymbol{I}$;	// Initial Hessian approximation
$\mu > 0$;	// Convergence tolerance
$k = 0$;	
while $\lvert\lvert\nabla J^k\rvert\rvert > \mu$ **do**	
compute search direction $\Delta\boldsymbol{p}^k$;	// Newton direction with Hessian approximation, Eq. (16)
compute step size α^k ;	// Backstepping to satisfy Amijio condition, Eq. (18)
update parameter set $\boldsymbol{p}^{k+1}$;	// Line search update, Eq. (14)
compute $\boldsymbol{H}^{k+1}$;	// BFGS update, Eq. (17)
$k = k + 1$;	
end	
Output: $\boldsymbol{p}^* = \boldsymbol{p}^k$;	// Optimal solution of the weights $\boldsymbol{V}^*, \boldsymbol{W}^*$, and $\boldsymbol{b}^*$

INJECTION RATE OBSERVER

Aging effects within the nozzles of common rail diesel injectors result in deteriorating injection behavior. We use the neural network observer to estimate the injection rate of a diesel injector, taking the changed dynamics due to nozzle wear into consideration. The injector of interest is a solenoidal CRIN3.18 injector manufactured by Bosch, which is depicted is Fig. 2. Modeling the bottom part of the injector is sufficient because we are investigating the effects of nozzle wear. Measurement data at different operating conditions is available for the marked signals rail pressure, p_R, lower feed-line pressure, p_L, needle lift x_N, and the injection rate, Q_H, which is used to analyze the observer's performance.

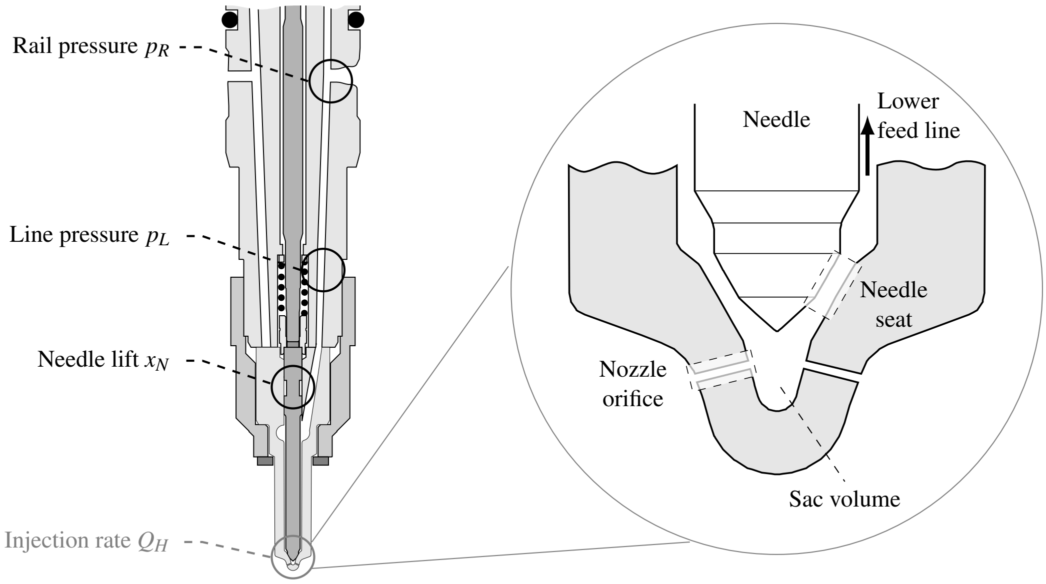


Figure 2 – Bottom part of the considered common rail diesel injector CRIN3.18 with available measurement signals and detailed nozzle section

A simplified model describing the injection dynamics is used for observer design. We model the one-dimensional flow of incompressible diesel fluid in the lower feed line using the physical principles of mass conservation, Newton's second law, and energy conservations. To obtain an analytical solution of these equations several simplifications are applied. Neglecting the convective terms as well as gravity effects and applying the Galerkin method (Pfeiffer and Borchsenius, 2004) with steady friction in the pipe leads to the following equation

$$\dot{Q}_L = \frac{A_L}{\rho_L L_L}(p_R - p_L) - \frac{\xi}{2A_L L_L} Q_L|Q_L| \tag{19}$$

The storage capability of the lower feed line is considered using a compressible model of the cavity

$$\dot{p}_L = \frac{K_L}{V_L}(Q_L - Q_H) \tag{20}$$

The volumetric flow rate at the needle seat Q_S and the injected flow rate through the nozzle orifices are obtained using Bernoulli equations. The sac volume is considered by applying the continuity equation to the cavity

$$Q_S = \operatorname{sgn}(p_L - p_S)\alpha_S A_S \sqrt{\frac{2\,|p_L - p_S|}{\rho_C}} \tag{21}$$

$$\dot{p}_S = \frac{K_C}{V_S}(Q_S - Q_H) \tag{22}$$

$$Q_H = \operatorname{sgn}(p_S - p_C)\alpha_H A_H \sqrt{\frac{2\,|p_S - p_C|}{\rho_C}} \tag{23}$$

where p_C is the pressure in the combustion chamber, which is supposed to be constant.

We obtain a normalized state space representation of the equations using rail pressure and area at the needle seat as system inputs, and line pressure, line flow rate and sac pressure as states.

$$\boldsymbol{u} = \begin{bmatrix} \frac{p_R - p_C}{p_0} & \frac{A_S}{A_0} \end{bmatrix}^T \qquad \boldsymbol{x} = \begin{bmatrix} \frac{p_L - p_C}{p_0} & \frac{Q_L}{Q_0} & \frac{p_S - p_C}{p_0} \end{bmatrix}^T \tag{24}$$

Equation (19) to (23) can then be rewritten for $A_S > 0$ as

$$\begin{aligned} \dot{\boldsymbol{x}} &= \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}) = \begin{bmatrix} a_1 x_2 - a_2\sqrt{x_3} \\ a_3(u_1 - x_1) - a_4 x_2 |x_2| \\ a_5 u_2 \sqrt{x_1 - x_3} - a_6\sqrt{x_3} \end{bmatrix} \\ \boldsymbol{y} &= \boldsymbol{C}\boldsymbol{x} \end{aligned} \tag{25}$$

The measurement of the line pressure is used to identify nozzle wear, which results in $y = x_1$. Based on the state space model, we design a neural network observer according to Eq. (2) to estimate the injection behavior

$$\begin{aligned} \dot{\hat{\boldsymbol{x}}} &= \boldsymbol{f}(\hat{\boldsymbol{x}}, \boldsymbol{u}) + \boldsymbol{\phi}(\hat{\boldsymbol{x}}, \boldsymbol{u}) \\ \hat{\boldsymbol{y}} &= \boldsymbol{C}\hat{\boldsymbol{x}} \end{aligned} \tag{26}$$

To analyze the observer's performance using a simulated experiment, the system plant was modeled by adding a fault model representing nozzle wear as follows

$$\Phi(\boldsymbol{x},\boldsymbol{u}) = \begin{bmatrix} -\nu_H a_2 \sqrt{x_3} \\ 0 \\ \nu_S a_5 u_2 \sqrt{x_1 - x_3} - \nu_H a_6 \sqrt{x_3} \end{bmatrix} \tag{27}$$

The fault model describes parameter variation within the needle seat and the nozzle orifices with $\nu_S, \nu_H \in (-\frac{1}{2}, \frac{1}{2})$ and results in additional nonlinear system dynamics to be identified.

RESULTS

The validation results of the simplified state space model are presented in Fig. 3. We use measurement signals of the rail pressure and the needle lift as input signals for the model and compare the resulting feed-line pressure and the mass injection rate to measurement data. The figure shows the results at energizing time 1.9 ms and set rail pressures of 60 MPa, 100 MPa, and 140 MPa. It is obvious that despite the model simplifications the simulation results show good agreement with the measurement data at all operating points. The model is able to predict significant signals during the injection process and is therefore used for injection rate observation.

We use the neural network observer with $q = 5$ hidden layer neurons. The observer design vector is experimentally determined to be $\boldsymbol{L} = \begin{bmatrix} 1 & 0 & 12 \end{bmatrix}^T$. To compare the results, we use a scaled identity instead of random numbers to initialize the network weights. Furthermore, network training according to Alg. 1 is stopped after 30 training epochs for the same reason. As an example, Fig. 4 shows a state-approximation result at energizing time 1.9 ms and set rail pressure 100 MPa using the network observer. The fault model according to Eq. (27) is added to the initial plant model using $\nu_S = 0$ and $\nu_H = 0.3$. The estimation of state x_3, which is relevant for estimating the injection rate, is depicted in Fig. 4(a) compared to the output of the model without observer. The neural network observer obviously predicts the state of interest much better than the initial model. Figure 4(b) shows the identified network output ϕ_1 to ϕ_3. The observer tracks the artificial plant fault quite well, using the considered design vector $\boldsymbol{L}$. Note, that faults with more complex dynamic behavior are difficult to identify, as $\boldsymbol{L} = \text{const.}$ is used to distribute the measured error to all states.

We analyzed the performance of the network observer by means of the mean squared error (MSE) of the state x_3, which is defined as

$$\text{MSE} = \frac{1}{N} \sum_{i=0}^{N} \left[x_3(t_i) - \hat{x}_3(t_i) \right]^2 \tag{28}$$

The observer is tested varying the nozzle parameter within $\nu_H \in (-\frac{1}{2}, \frac{1}{2})$. The mean squared error is evaluated for each configuration, which is depicted in Fig. 5. Using the neural network observer, the estimation error improves at each fault condition that was analyzed compared to the initial model. Additionally, the resulting state of interest is shown for the extreme nozzle aging parameters, $\nu_H = \pm 0.5$. The observer predicts the shape of the faulty state in both cases, but slightly better for $\nu_H = 0.5$ as the observer is unable to estimate the modified end of injection for $\nu_H = -0.5$. Regarding nozzle wear, which can be described as linear parameter change of the nozzle orifice discharge coefficient, the neural network observer exhibits very good overall performance.

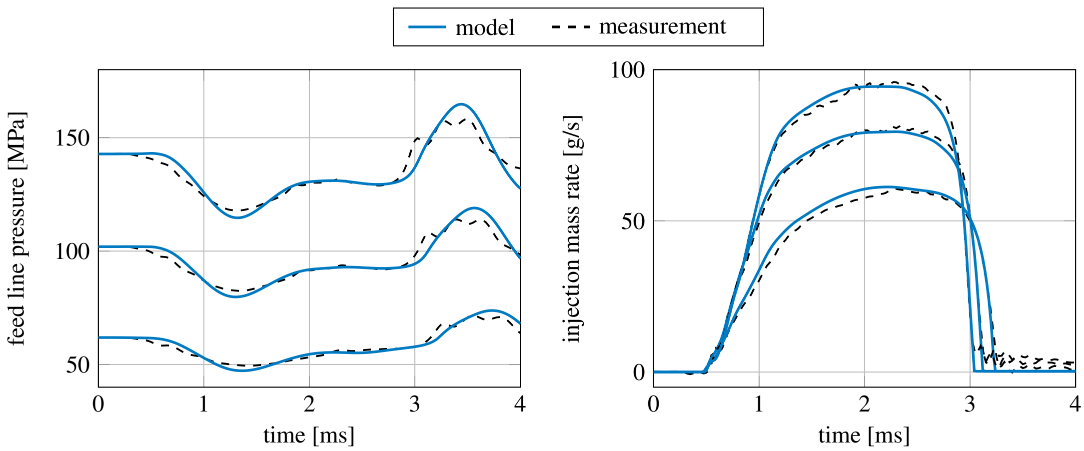
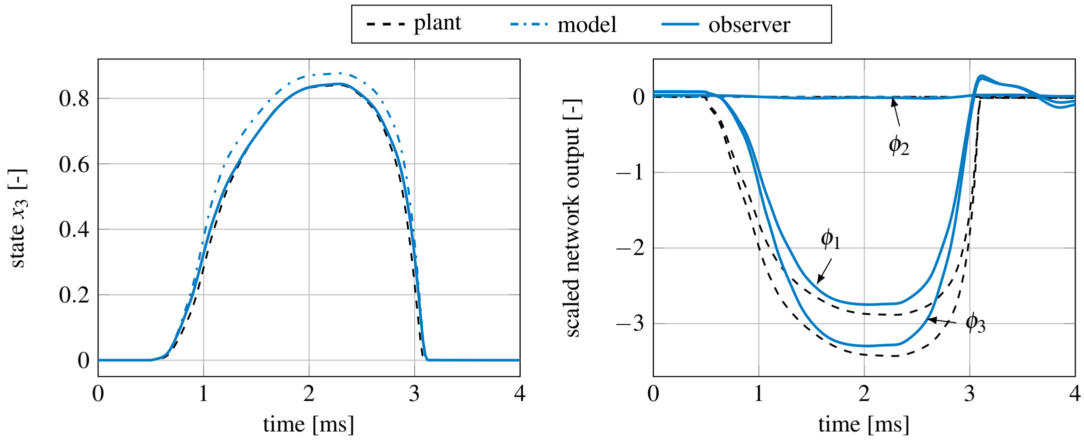


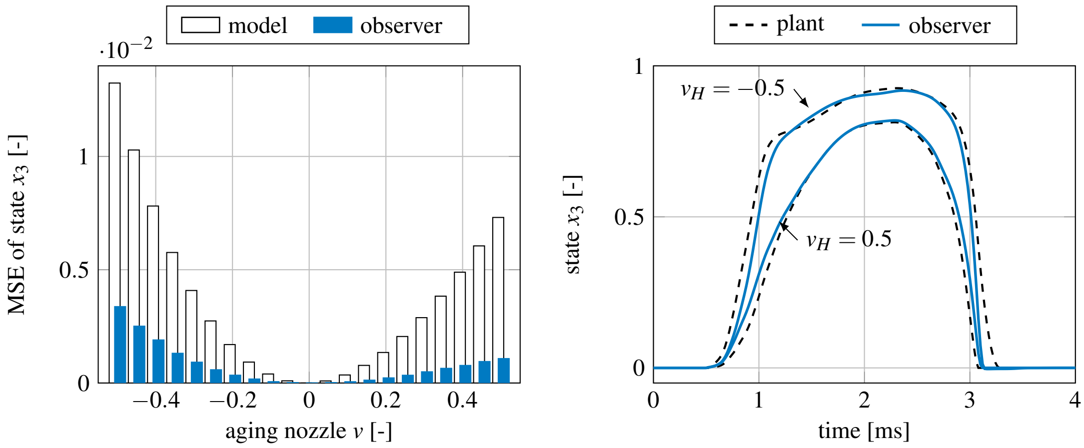
Figure 3 – Validation of the simplified state-space injector model with experimental data at energizing time 1.9 ms and set rail pressures 60 MPa, 100 MPa, and 140 MPa.



(a) State approximation with the original model and the neural network observer

(b) Identified network output compared to the plant fault

Figure 4 – Identification of the nonlinearity due to nozzle wear at energizing time 1.9 ms and set rail pressure 100 MPa. Artificial nozzle wear of $v_H = 0.3$ is identified using the injection rate observer algorithm after 30 training epochs.



(b) Mean squared error with nozzle aging $v_H \in (-\frac{1}{2}, \frac{1}{2})$

(a) State estimation of state x_3 with $v_H = \pm 0.5$

Figure 5 – Performance of the network observer with simulated aging parameter in the nozzle $v_H \in (-\frac{1}{2}, \frac{1}{2})$

CONCLUSION

In this paper, a neural state observer scheme was proposed for application to a common rail diesel injector exhibiting aging phenomena. The observer design includes a three-layer MLP feed-forward network for fast, flexible adoptions of the observer's characteristics. Applying a Newton optimization procedure combined with a backstepping method yields good learning-process efficiency. Separate fault identification guarantees correct network output error for the weight adoption process. As the presented concept combines system identification with a classical observer scheme, a dynamic model of the fault in the plant can be obtained. Simulation results confirm the applicability of the proposed observer to the injector model under wear and aging effects. Because of reliable performance and the identified fault model, the neural network observer is beneficial to be used for control methods in future investigations.

ACKNOWLEDGMENTS

This research is supported by the "Deutsche Forschungsgemeinschaft" (German Research Foundation, project RI2451/1-1). The authors would like to thank the Chair of Internal Combustion Engines, Technical University of Munich, for providing the experimental data used in this paper.

REFERENCES

Abdollahi, F., Talebi H.A. , and Patel R. V., 2006, "A stable neural network-based observer with application to flexible-joint manipulators", IEEE Transactions on Neural Networks, Vol. 17, No. 1, pp. 118–129.

Hintz, C., 2003, "Identifikation nichtlinearer mechatronischer Systeme mit strukturierten rekurrenten Netzen". PhD thesis, Lehrstuhl für Elektrische Antriebssysteme, Technische Universität München.

Krogerus, T.R., Hyvönen M. P. , and Huhtala K. J., 2016, “A Survey of Analysis, Modeling, and Diagnostics of Diesel Fuel Injection Systems”, Journal of Engineering for Gas Turbines and Power, Vol. 138, No. 8, pp. 081501–081501.

Nelles, O., 2001, ”Nonlinear System Identification: From Classical Approaches to Neural Networks and Fuzzy Models”, Springer, Berlin Heidelberg, Germany, ISBN: 978-3-540-67369-9.

Nocedal, J. and Wright, S., 2006, ”Numerical optimization”, Springer Science & Business Media, New York, USA, ISBN: 978-0387-30303-1.

Pfeiffer, F. and Borchsenius, F., 2004, “New Hydraulic System Modelling”, Journal of Vibration and Control Vol. 10, No. 10, pp. 1493–1515.

Talebi, H.A. et al., 2010, “Neural Network-Based State Estimation of Nonlinear Systems”, Lecture Notes in Control and Information Sciences, Vol. 395, pp. 15–35.

Williams, R.J. and Zipser D., 1989, “A Learning Algorithm for Continually Running Fully Recurrent Neural Networks”, Neural Computing, Vol. 1, No. 2, pp. 270–280.

RESPONSIBILITY NOTICE

The authors are the only responsible for the material included in this paper.