



Neuro-fuzzy-based Electronic Brake System Modeling using Real Time Vehicle Data

Ana Farhat¹, Kyle Hagen², Ka C Cheok¹ and Balaji Boominathan²

¹Oakland University, Rochester, MI, ²Continental Automotive Systems Inc. Auburn Hills, MI.

anafarhat@oakland.edu, kyle.hagen@continental-corporation.com,
cheok@oakland.edu, Balaji.boominathan@continental-corporation.com

Abstract

Electronic Brake System (EBS) is considered as one of the most complicated systems whose performance depends on the subsystems parameters. Usually these parameters are difficult to predict. Based on the task to improve the EBS performance, this article presents a mathematical modeling approach based on neuro-fuzzy network method to model a subsystem of EBS. For the model parameters identification, a neuro-fuzzy network has been implemented based on Least Square Error (LSE) and Levenberg-Marquardt Algorithm (LMA) as the optimization algorithms. Finally, the performance of identified model has been evaluated.

1 Introduction

Recent advances within the field of Electronic Brake System (EBS) have come in the form of increasingly capable electrohydraulic actuators. These actuators allow for increased performance in braking and weight savings over conventional hydromechanical brake systems. These new actuators also enable by-wire braking and future technologies like automated driving. Continental Automotive has developed an electrohydraulic brake system for the next generation of EBS. The architecture presented in Figure 1 shows an overview of the major components used within the MKC1 system [1]. The architecture can be used to create a simulated model which can analyze numerous situations and scenarios that the brake system will experience. Here we explain how the system operates in sufficient detail.

The driver applies on a brake pedal which connects to an internal hydraulic actuator. This actuator determines through travel and pressure measurements, how much brake fluid the electrohydraulic actuator needs to supply. The boost controller drives the electrohydraulic actuator by commanding a certain amount of output. The electrohydraulic actuator supplies a pressure and volume and splits up to individual wheels. These final wheel actuators of the foundation brake system provide torque on the wheels which translates to deceleration of the vehicle.

Due to the importance of this system and its control applications, this paper will detail brake model identification method. There are different identification approaches that can be selected according to

the system's specifications. In the identification and modeling process, the first step is choosing the right model structure, and the next step will be selecting appropriate optimization algorithm. One of the model structures that can identify nonlinear dynamic systems with high accuracy is adaptive network-based fuzzy inference system (ANFIS). This model is used to identify single-input single-output (SISO) systems.

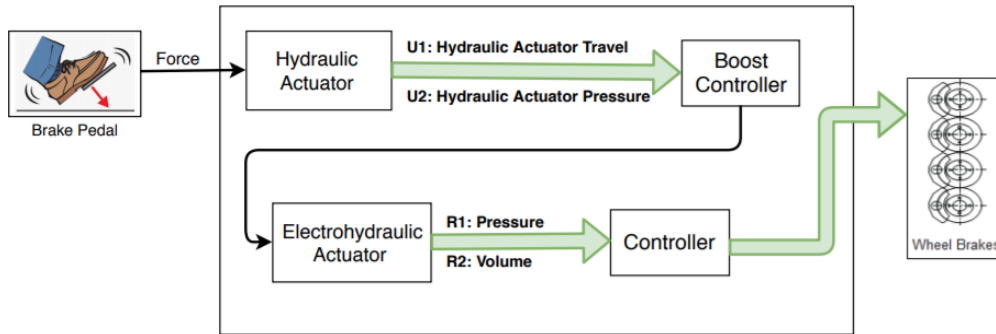


Figure 1: Architecture of brake system

As is shown in Figure 1, the studied model has two inputs and two outputs. The inputs (U1, U2) are taken as the output of the hydraulic actuator block, and the outputs (R1, R2) are the outputs of electrohydraulic actuator. Therefore, this model is a multi-input multi-output (MIMO) model. To be able to model a MIMO system with ANFIS structure, ANFIS should be modified to coactive neuro-fuzzy inference system (CANFIS).

As mentioned before, the second step in identification is choosing the optimization method. Different optimization algorithms can be selected to optimize the parameters of the model. Chatterjee and Watanabe [2] use particle swarm optimization (PSO) to update whole parameters of the ANFIS. In [3], a hybrid optimization method which presents PSO for antecedent part and gradient descent (GD) for consequence part is used. Reference [4] compares different training algorithms; the algorithms which are compared are GD, Resilient Propagation, Quick prop, and LMA. The LMA was originally designed [5,6] to serve as an intermediate optimization algorithm between the Gauss–Newton (GN) method and gradient descent algorithm.

In this paper CANFIS model has been selected to identify two-input two-output model. As the optimization algorithm, Levenberg-Marquardt algorithm (LMA) is employed to update the parameters in the antecedent part of CANFIS. The consequence parameters are identified by least square estimation (LSE). The article is organized as follows. In section 2, CANFIS structure and applied optimization algorithms are reviewed. In section 3, the performance of trained model is verified. Section 4 presents conclusion, the references are presented at the end.

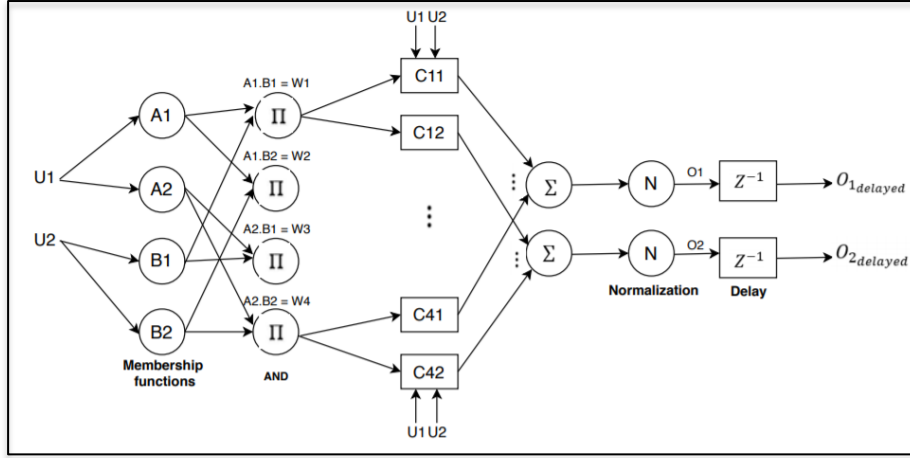


Figure 2: CANFIS network

2 CANFIS Model as an Identifier

To identify above mentioned model which is a MIMO system, CANFIS model has been selected as an identifier. CANFIS has the capability of modeling the systems which have more than one output. The structure of CANFIS is similar to ANFIS. Detailed information regarding ANFIS can be found in [8]. CANFIS rules comprise a set of IF-antecedent and THEN-consequent, which are aggregated to produce an output.

An example of CANFIS model with two inputs (U_1, U_2) and two outputs ($O_{1_delayed}, O_{2_delayed}$) is presented in Figure 2. For simplicity of presentation, the number of membership functions for each input is set to two. The equations corresponding to Figure 2 is presented in (1).

$$\begin{aligned}
 & \text{If } U_1 \text{ is } A_1 \text{ and } U_2 \text{ is } B_1, \text{ then } \begin{cases} C_{11} = p_{11}U_1 + q_{11}U_2 + r_{11} \\ C_{12} = p_{12}U_1 + q_{12}U_2 + r_{12} \end{cases} \\
 & \text{If } U_1 \text{ is } A_1 \text{ and } U_2 \text{ is } B_2, \text{ then } \begin{cases} C_{21} = p_{21}U_1 + q_{21}U_2 + r_{21} \\ C_{22} = p_{22}U_1 + q_{22}U_2 + r_{22} \end{cases} \\
 & \text{If } U_1 \text{ is } A_2 \text{ and } U_2 \text{ is } B_1, \text{ then } \begin{cases} C_{31} = p_{31}U_1 + q_{31}U_2 + r_{31} \\ C_{32} = p_{32}U_1 + q_{32}U_2 + r_{32} \end{cases} \\
 & \text{If } U_1 \text{ is } A_2 \text{ and } U_2 \text{ is } B_2, \text{ then } \begin{cases} C_{41} = p_{41}U_1 + q_{41}U_2 + r_{41} \\ C_{42} = p_{42}U_1 + q_{42}U_2 + r_{42} \end{cases}
 \end{aligned}$$

and the outputs of the CANFIS model are

$$\begin{aligned}
 O_1 &= \frac{C_{11}W_1 + C_{21}W_2 + C_{31}W_3 + C_{41}W_4}{W_1 + W_2 + W_3 + W_4} \\
 O_2 &= \frac{C_{12}W_1 + C_{22}W_2 + C_{32}W_3 + C_{42}W_4}{W_1 + W_2 + W_3 + W_4}
 \end{aligned} \tag{1}$$

where W_1 is $\mu_{A_1}(U_1) \cdot \mu_{B_1}(U_2)$ and so on.

The generalized form of CANFIS rules for i^{th} rule:

If U_1 is A and U_2 is B, then $C_{ij} = p_{ij}U_1 + q_{ij}U_2 + r_{ij}$ (2)

where $i = 1, 2, \dots, n^2$, n is number of membership functions for each input and n^2 is total number of rules. And $j = 1, \dots, k$, k is number of outputs. A and B, membership functions of input1 and 2, are Gaussian functions as defined in (3).

$$\begin{aligned} \mu_{A_f}(U_1) &= \exp^{-0.5\left(\frac{U_1 - c_f}{s_f}\right)^2} \\ \mu_{B_d}(U_2) &= \exp^{-0.5\left(\frac{U_2 - c_d}{s_d}\right)^2} \end{aligned} \tag{3}$$

where $f = 1, \dots, n$ and $d = n+1, \dots, 2n$. According to (2) and (3), CANFIS model has some unknown parameters which should be estimated.

2.1 Least Square Error (LSE)

Unknown parameters in consequence part of (2) are defined in matrix θ

$$\theta = [p_{11} \ q_{11} \ r_{11} \ \dots \ p_{n^2k} \ q_{n^2k} \ r_{n^2k}]^T$$

According to (1), generalized form of first output of CANFIS is

$$O_1 = \frac{c_{11}W_1 + \dots + c_{n^2_1}W_{n^2}}{W_1 + \dots + W_{n^2}} \tag{4}$$

According to (4) we have

$$\begin{matrix} \begin{bmatrix} W_1U_{11} & W_1U_{21} & W_1 & \dots & W_{n^2}U_{11} & W_{n^2}U_{21} & W_{n^2} \\ & \vdots & & \ddots & & \vdots & \\ W_1U_{1N} & W_1U_{2N} & W_1 & \dots & W_{n^2}U_{1N} & W_{n^2}U_{2N} & W_{n^2} \end{bmatrix} & \begin{bmatrix} p_{11} \\ q_{11} \\ r_{11} \\ \vdots \\ p_{n^2_1} \\ q_{n^2_1} \\ r_{n^2_1} \end{bmatrix} & \approx & \begin{bmatrix} (W_1 + \dots + W_{n^2})R_{11} \\ \vdots \\ (W_1 + \dots + W_{n^2})R_{1N} \end{bmatrix} \\ \text{Matrix } M & \text{Matrix } \theta & & \text{Matrix } K \end{matrix} \tag{5}$$

$$\Rightarrow M\theta \approx K \tag{6}$$

which N is the number of samples, and R_1 is the desired output. In (5), O_1 is replaced by R_1 . Actually, $O_1 = [O_{11} \ \dots \ O_{1N}]^T$ as the first output of the model should match up with $R_1 = [R_{11} \ \dots \ R_{1N}]^T$. By LSE

$$\hat{\theta} = \begin{bmatrix} \hat{p}_{11} \\ \hat{q}_{11} \\ \hat{r}_{11} \\ \vdots \\ \hat{p}_{n^2_1} \\ \hat{q}_{n^2_1} \\ \hat{r}_{n^2_1} \end{bmatrix} = (M^T M)^{-1} M^T K$$

The unknown parameters of second output of model (O_2) can also be estimated in a similar way.

2.2 Levenberg-Marquardt Algorithm (LMA)

In this paper, the unknown parameters of antecedent part of CANFIS which are parameters of Gaussian membership functions are estimated using LMA as the learning algorithm. Letting $\mathbf{e}_1 = R_1 - O_{1_{delayed}}$ and $\mathbf{e}_2 = R_2 - O_{2_{delayed}}$, cost function can be defined as

$$T = 0.5(\mathbf{e}_1^T \mathbf{e}_1 + \mathbf{e}_2^T \mathbf{e}_2)$$

where \mathbf{e}_1 and \mathbf{e}_2 are the error vectors corresponding to first and second output respectively. By considering N sample data and one sample time delay in the output, error vectors are represented as $\mathbf{e}_1 = [e_{12} \ \dots \ e_{1N}]^T$ and $\mathbf{e}_2 = [e_{22} \ \dots \ e_{2N}]^T$. Therefore, total error vector will be

$$E = [\mathbf{e}_1 \ \mathbf{e}_2]^T$$

LMA uses Jacobian matrix J which is a gradient matrix representing the partial derivatives of error vector E with respect to V [7]. Here we have the Jacobian matrix equation [7].

$$J = \frac{\partial E}{\partial V} = \frac{\partial \begin{bmatrix} \mathbf{e}_1 \\ \mathbf{e}_2 \end{bmatrix}}{\partial \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_{4n} \end{bmatrix}} = \begin{bmatrix} \frac{\partial e_{12}}{\partial v_1} & \dots & \frac{\partial e_{12}}{\partial v_{4n}} \\ \vdots & \ddots & \vdots \\ \frac{\partial e_{1N}}{\partial v_1} & \dots & \frac{\partial e_{1N}}{\partial v_{4n}} \\ \frac{\partial e_{22}}{\partial v_1} & \dots & \frac{\partial e_{22}}{\partial v_{4n}} \\ \vdots & \ddots & \vdots \\ \frac{\partial e_{2N}}{\partial v_1} & \dots & \frac{\partial e_{2N}}{\partial v_{4n}} \end{bmatrix}$$

By using Jacobian matrix, LMA update is expressed as

$$V_{new} = V - ((J^T J) + \mu I)^{-1} J^T E \quad (7)$$

For detailed information about μ please refer to [8].

3 Training Results

To evaluate the accuracy of the proposed model, input/output data has been taken from brake module. As mentioned before, this model is designed with two inputs two outputs. Inputs are pedal travel and pressure change measurements caused by driver, and the outputs are volume and pressure change in the brake actuator. Before training the model we are going to evaluate the effects of number of membership functions on the accuracy of the model.

3.1 Effects of the number of membership functions on the accuracy of the model

The number of membership functions in CANFIS model directly affects the accuracy of the model. In this paper different numbers of membership functions (n) have been considered for CANFIS model. We started the modeling with $n = 2$, and by checking the accuracy of the model, eventually $n = 4$ has been selected as the number of membership functions for this model. Figures 3 and 4 represents the results. In Figure 3, 2 membership functions are considered for each input in the model. In this figure desired outputs ($R1, R2$) are compared with outputs of the model ($O1, O2$). As is shown, the second output of the model is not mimicking the data. Same comparison has been implemented in Figure 4 with 4 membership functions for each input. According to these results, 4 membership functions have been selected to model the data.

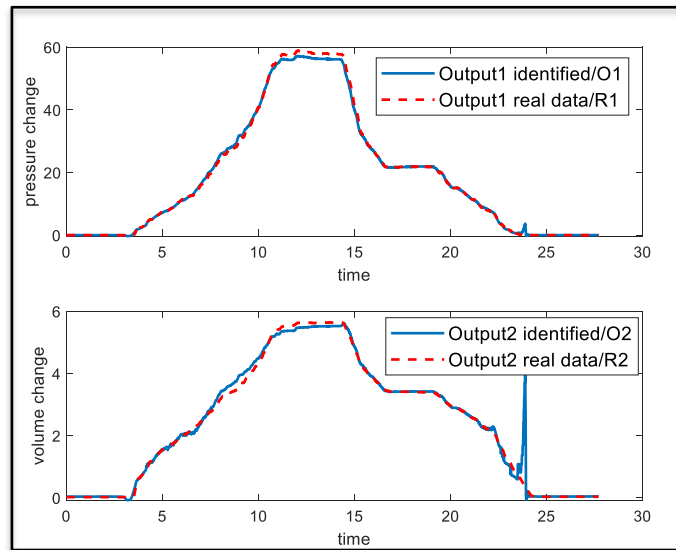


Figure 3: performance of the model with $n = 2$

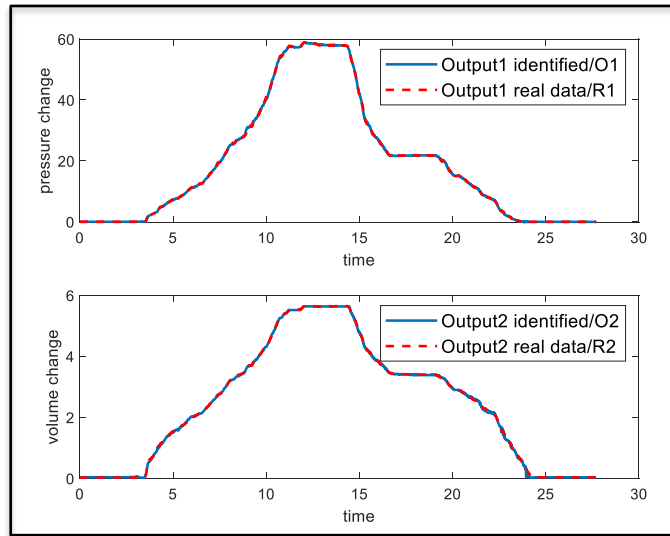


Figure 4: performance of the model with $n = 4$

3.2 Training the model

In this section, with the selected number of membership functions, model will be trained. Figure 5 shows the training dataset having been used to train the model.

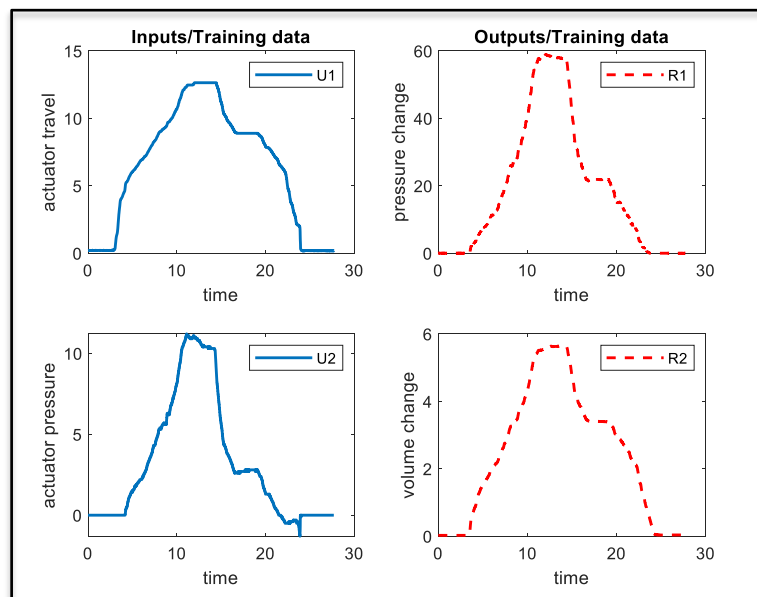


Figure 5: Input/Output training data

The accuracy of trained model is shown in Figure 6. In this figure the desired outputs (R1, R2) are compared to the estimated outputs (O1, O2).

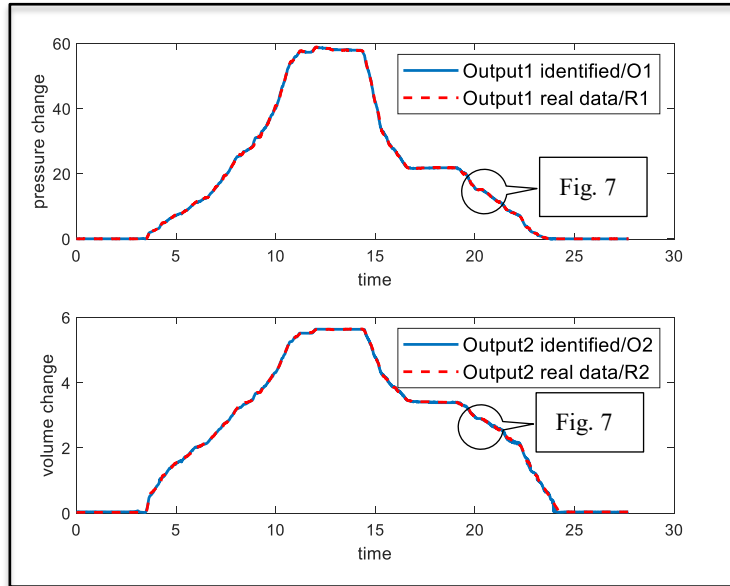


Figure 6: Identified outputs by CANFIS

A magnification of Figure 6 is shown in Figure 7. And error plots corresponding to Figure 6 show the difference between training data and output of the model presented in Figure 8.

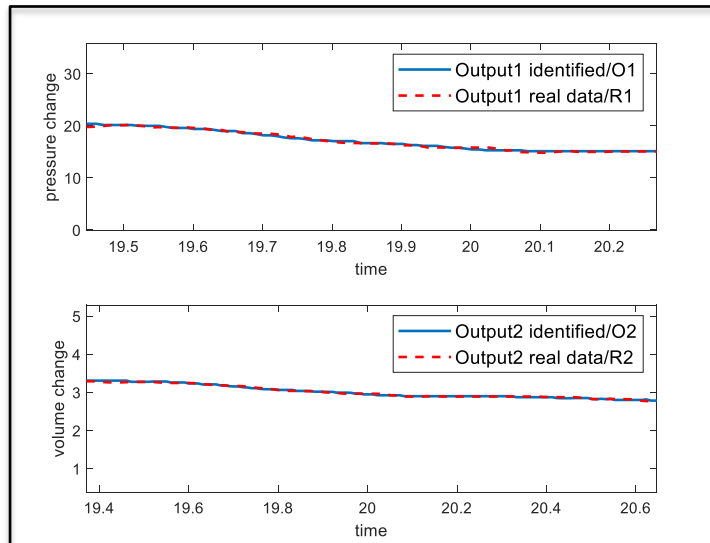


Figure 7: magnification of part of fig. 6

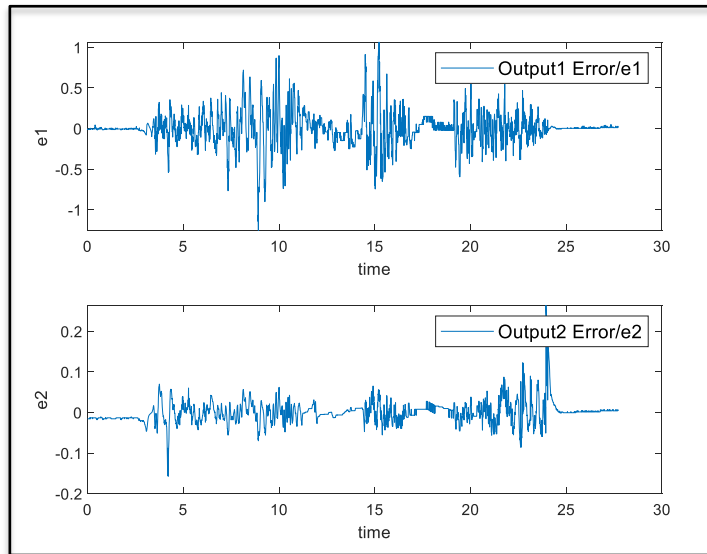


Figure 8: error plots

3.3 Validating the model

The trained model has been validated by using three different datasets. As an example one set of validation data is shown in Figure 9.

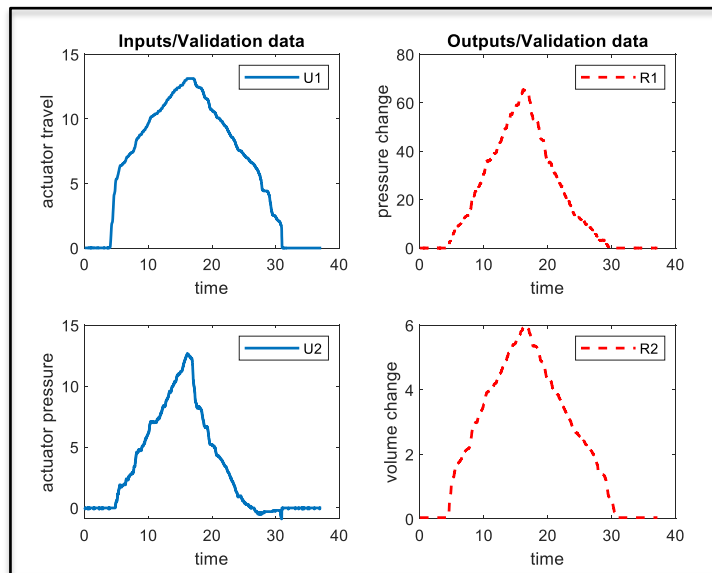


Figure 9: validation data

The outputs of trained model using validation input data presented in Figure 10. As it's shown in this figure, trained model has acceptable accuracy.

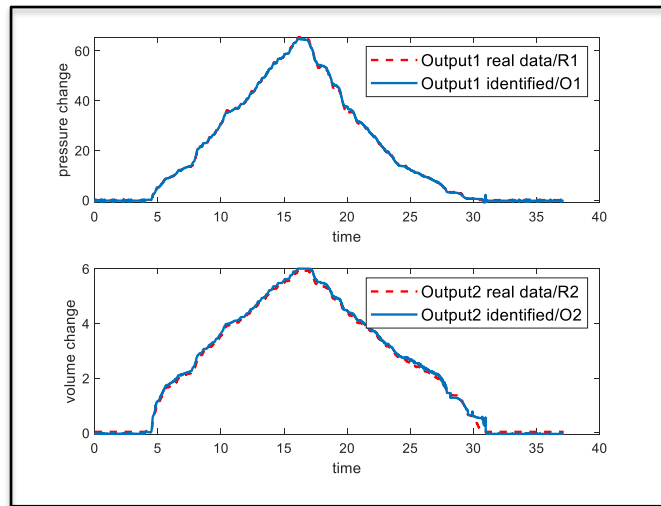


Figure 10: outputs comparison using validation data

4 Conclusion

Currently, EBS systems are modeled using physics-based simulation environments. These models are beneficial for predicting system component behavior according to their predefined model. When it comes to predicting the entire system behavior, these methods cannot determine the efficiencies of real-world parts or interactions that cause off-nominal behavior from the model. Neuro-fuzzy network modeling strategies determines system behavior with better accuracy since the correlation of real-world data and the model is more direct. These strategies, like CANFIS, are a viable method for EBS modeling and provide tight accuracies.

References

- [1] <https://www.continental-automotive.com/en-gl/Passenger-Cars/Chassis-Safety/Brakes/Electronic-Brakes/MK-C1/MK-C1>
- [2] Chatterjee A, Watanabe K (2006) An optimized TakagiSugeno type neuro-fuzzy system for modeling robot manipulators. *Neural Comput Appl* 15(1):55–61
- [3] Aliyari M Sh, Teshnehlab M, Sedigh AK (2006) A novel training algorithm in ANFIS structure. *Proceedings of the American Control Conferences*
- [4] Chen MS (1999) A comparative study of learning methods in tuning parameters of fuzzy membership functions. *IEEE Conf on Syst Man Cybern* 3:40–44
- [5] D. Marquardt, An algorithm for least squares estimation of non-linear parameters, *J. Ind. Appl. Math.*(1963) 431–441
- [6] Levenberg–Marquardt Training by Hao Yu and Bogdan M. Wilamowski
- [7] Hao Yu, Bogdan M. Wilamowski, Levenberg–Marquardt Training, Auburn University
- [8] Farhat Ana, Cheok Ka C, Improving Adaptive Network Fuzzy Inference System with Levenberg–Marquardt Algorithm *2017 Annual IEEE International Systems Conference, April 2017, pp. 1-6*