Real Time Leak Isolation in Pipelines Based on a Time Delay Neural Network

A. Navarro
Sch. of Engineering and Science
ITESM
Zapopan, México
adrian.navarro@tec.mx

O. Begovich
Dept. of Electrical Eng.
CINVESTAV
Zapopan, México
obegovi@gdl.cinvestav.mx

J.A. Delgado-Aguiñaga
CIIDETEC-UVM
UVM, Guadalajara Sur
Tlaquepaque, México
jorge.delgado@uvmnet.edu

J. D. Sánchez-Torres
Dept. of Mathematics and Physics
ITESO
Tlaquepaque, México
dsanchez@iteso.mx

***Abstract*—In this paper, the one leak insulation issue in a water pipeline is addressed through a Time-Delayed Neural Network. This scheme is an alternative to better computing performance since conventional model-based methods usually have high workloads due to the complexity of the mathematical model of the pipeline compared to the speed of the dynamics of leakage. Besides, the design of the neural network could offer an improved time efficiency by using the parallel architecture of some electronic systems such as the FPGA.**

The current proposal is based on a scheme in which a mathematical model of the pipeline is used to generate synthetic training data. These training data are obtained using different leak sizes and leak positions and are also corrupted by random noise in order to emulate the actual data pipe. Finally, to show the potential of this method, some results are presented using real-noise databases from a pipeline prototype. Finally, only the flow and pressure sensor at both ends of the aqueducts are used for procedure following the classic leak diagnosis hypothesis.

***Index Terms*—Artificial Neural Network, Pipelines, Leak Isolation, Dynamic Systems.**

I. Introduction

Pipelines are widely used for the distribution of liquids in many areas: the petroleum industry, drinking water distribution systems, mining, to name but a few. For this reason, the design of techniques and algorithms for detecting and isolating leaks in real-time is an active study issue, given the undesirable effects of significant losses of water or oil. For instance, in recent years, the issue of illegal extraction of oil derivatives such as diesel and gasoline from distribution pipelines has increased, leading in significant economic losses for the Mexican oil company (PEMEX).

Leaks can be detected and located using external equipment such as acoustic devices, electronic audio sticks, tracer gas methods, infrared thermography, ultrasonic methods, electromagnetic techniques, among others. These external techniques are useful, but generally, involve specialized users, and it is also necessary to use such sophisticated hardware throughout the pipeline and sometimes even to empty it.

An alternative technique for detecting leaks is using computing algorithms that apply a mathematical pipeline model to predict the size and place of the leak, i.e., a model-based leak detection system. In general, these methods use the measurement of the flow and pressure heads upstream and downstream of the pipe.

Several works on model-based methods have been suggested to reduce economic losses and environmental pollution. It is appropriate to mention [1] as a pioneer in this field. In that reference, a discrete model of water dynamics is used to predict the flow and the head at each interior point of discretization. Thus, an estimate of the coefficient of friction can also be obtained. As a standard hypothesis, the steady-state head and flow conditions at the ends of the pipe are used to estimate the leakage intensity and the leakage location.

More recent works can be mentioned. Notably, in [2], a leak detection an isolation algorithm based on [1] has been implemented and tested with accurate results and under steady-state conditions. In [3], the authors propose an approach where the leak location is done by a hybrid technique based on real-time transient modeling method and negative pressure wave. The work presents successful results in locating gas pipeline leak. The work [4] deals with the leak isolation problem in plastic pipelines; for this case, the proposed algorithm uses a robust exact differentiation method for state variables and leak parameters. As the main feature of the previous results, the authors improve the leak detection accuracy taking the temperature variation into account. The reference [5] shows a detection method for a pressurized liquid ammonia pipeline with a leak. Here, the proposed approach consists of a leak indicator together with the one-dimensional steady-state flow model in order to detect the leak. Experiments on different leak positions and ratios from liquid R22 and ammonia pipelines are carried out to validate this method.

Although these works have been designed and tested either in pilot plants or in a simulation setting, the dynamics of the pipeline is generally fast, making the sample time to be smaller and therefore increasing the need for complicated and costly computing devices. In order to prevent this scenario, this note proposes a leak diagnosis system based on a neural network since these structures are naturally appropriate to specific electronic devices such as the FPGA. This suitable hardware is widely employed to accelerate software such as in the field of artificial intelligence (training and implementation of neural networks and machine learning algorithms) and massively parallel computing applications, [6].

The paper continues as follows: Section II provides the neural network fundamentals. Section III describes the pipeline mathematical model in order to generate the synthetic data useful to train the net. The proposed neural network scheme is presented in Section IV. Section V simulation results are presented an discussed. Finally, Section VI concludes the paper.

II. Pipeline Mathematical Model

This section is organized into two parts. The first part describes a couple of Partial Differential Equations and the corresponding leak model, which describes the pipeline dynamics. In the second one, a finite-dimensional approximation is obtained from a space discretization, useful to apply a model-based leak detection algorithm.

A. Pipeline Partial Differential Equation

Under the following assumptions: a straight-horizontal pipe with constant cross-section area and without fittings, fluid and wall duct being slightly deformable, convective changes in velocity being neglected and the fluid density being constant, then the fluid transient response can be described by a couple of quasilinear hyperbolic partial differential equations (*PDE's*) as [7]:

Momentum Equation

$$\frac{\partial Q(z,t)}{\partial t} + gA\frac{\partial H(z,t)}{\partial z} + \mu Q(z,t)\left|Q(z,t)\right| = 0 \quad (1)$$

Continuity Equation

$$\frac{\partial H(z,t)}{\partial t} + \frac{b^2}{gA}\frac{\partial Q(z,t)}{\partial z} = 0 \quad (2)$$

where Q is the flow rate $[m^3/s]$, H is the pressure head $[m]$, z the length coordinate $[m]$, t the time coordinate $[s]$, g the gravity acceleration $[m/s^2]$, A the cross-section area $[m^2]$, b the pressure wave speed in the fluid $[m/s]$, $\mu = f/2DA$, with D the inner diameter $[m]$ and f the friction factor.

Leak model: One leak can arbitrarily appear at any position z_1 at any time $t_l > 0$ (see Fig. 1) and it can be modeled as follows [7]:

$$Q_L = \lambda\sqrt{H_L} \quad (3)$$

where the constant λ is a function of the orifice area and the discharge coefficient $[m^{5/2}/s]$; Q_L is the flow through the leak and H_L is the head pressure at the leak point [7].

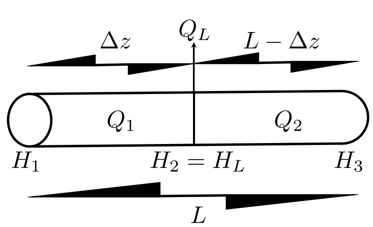

Fig. 1. Discretization of the pipeline with a leak Q_L

This leak produces a discontinuity on the pipe changing the mathematical model (1) and (2). Furthermore, due to the law of mass conservation, Q_L must satisfy the next relation:

$$Q_b = Q_a + Q_L \quad (4)$$

where Q_b and Q_a are the flows in an infinitesimal length before and after of the leak, respectively.

B. Spatial Discretization of the Modeling Equations

In order to obtain a finite dimensional approximation of from (1) and (2), those *PDE's* are discretized with respect to the spatial variable z, as in [8], [9], by using the following relationships:

$$\frac{\partial H(z_i,t)}{\partial z} \backsimeq \frac{H_{i+1} - H_i}{\Delta z_i} \quad \forall i = 1, \cdots, n \quad (5)$$

$$\frac{\partial Q(z_{i-1},t)}{\partial z} \backsimeq \frac{Q_i - Q_{i-1}}{\Delta z_{i-1}} \quad \forall i = 2, \cdots, n \quad (6)$$

where H_i, Q_i stand for $H(z_i,t)$ $Q(z_i,t)$, and $\Delta z_i = z_{i+1} - z_i$. Assuming only two partitions in the pipeline as shown in Fig. 1, Δz_j $(j = 1, 2)$ becomes the distance from the upstream end until the leak position and from the leak position until to the downstream end of the pipe, respectively. Notice that $\Delta z_2 = L - \Delta z_1$ (for simplicity, $\Delta z_1 = \Delta z$, so $\Delta z_2 = L - \Delta z$) where L is the total length of the pipeline. The leak position is assumed to be different from 0 and L in this description (i.e. the leak position is neither at the beginning nor at the end of the duct). Applying approximations (5) and (6) to equations (1) and (2) together with (3) and (4), we get:

$$\begin{bmatrix} \dot{Q}_1 \\ \dot{H}_2 \\ \dot{Q}_2 \end{bmatrix} = \begin{bmatrix} \frac{-gA}{\Delta z}\left(H_2 - u_1\right) - \frac{f(Q_1)}{2DA}Q_1|Q_1| \\ \frac{-b^2}{gA\Delta z}\left(Q_2 - Q_1 + \lambda\sqrt{H_2}\right) \\ \frac{-gA}{L-\Delta z}\left(u_2 - H_2\right) - \frac{f(Q_2)}{2DA}Q_2|Q_2| \end{bmatrix} \quad (7)$$

Here, the input vector is $u = [H_{in}\ H_{out}]^T = [u_1\ u_2]^T$, and the output vector is $y = [Q_1\ Q_2]^T = [Q_{in}\ Q_{out}]^T$.

III. Neural Network Fundamentals

In this section, a brief review of Time Delayed Neural Network (TDNN) is presented as well as a short discussion about the training process of the net.

A. Time Delayed Neural Network

An Artificial Neural Network can be defined as interconnected processors massively (neurons), operate in parallel and learn from their own experience [10]. Such interconnection is known as "the network architecture," and it is associated with the learning process used to train the net.

The TDNN arise as an extension of Static Neural Networks which are designed to explicitly includes time relationship between input-output mappings. Specifically, in a TDNN, a tapped delay line is given in the input followed by a multilayer perceptron. Figure 2, shows the first hidden layer structure of a TDNN (for more information see [11]).

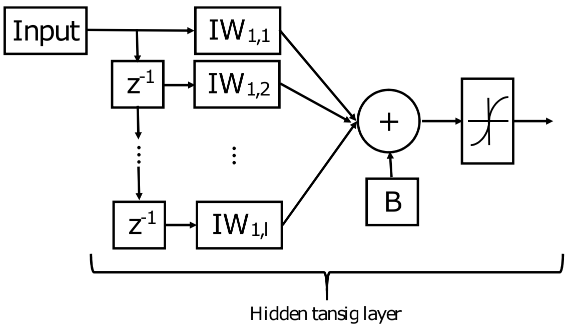

Fig. 2. Hidden layer structure of the Multilayer Perceptron

The TDNN has demonstrated excellent performance for classifying a temporal pattern that consists of a sequence of fixed dimensional feature vector such as phonemes [12] and acoustic modeling [13]. For this reason, a neural network with this architecture is profiled as a promising candidate to solve the leak diagnosis problem.

B. Supervised Training Frame Work

Achieving a desirable set of synaptic weights for a given predefined network architecture requires a training process. That process is generally based on an optimization scheme to adjust the network parameters, mainly the weights, to a set of input-to-output to be matched by the neural network model; namely, a supervised learning scheme. The well-known backpropagation algorithm based on a gradient descent technique [11] has been widely applied for general neural network training.

Like the quasi-Newton methods, the Levenberg-Marquardt algorithm was designed to approach second-order training speed without having to compute the Hessian matrix. When the performance function has the form of a sum of squares (as is typical in training feedforward networks), then the Hessian matrix can be approximated as $\mathbf{H} = \mathbf{J}^T\mathbf{J}$ and the gradient can be computed as $\mathbf{g} = \mathbf{J}^T\mathbf{e}$, where $\mathbf{J}$ is the Jacobian matrix that contains first derivatives of the network errors with respect to the weights and biases, and $\mathbf{e}$ is a vector of network errors. The Jacobian matrix can be computed through a standard backpropagation technique (see [14]) that is much less complex than computing the Hessian matrix.

The Levenberg-Marquardt algorithm uses this approximation to the Hessian matrix in the following Newton-like update (more information could be found in [11].):

$$\mathbf{w}_{k+1} = \mathbf{w}_k - \left[\mathbf{J}^T\mathbf{J} + \varphi\mathbf{I}\right]^{-1}\mathbf{J}^T\mathbf{e} \tag{8}$$

IV. Proposed Neural Network Scheme

This section presents the proposed leak isolation process (i.e., the task of determining the leak location Δz). It is carried out by the design of a TDNN described in section III. 400 neurons constitute the NN architecture in the input layer, 100 neurons in the hidden layer (with the logistic activation function) and, finally, one neuron in the output layer (with purelin activation function).

The 400 input layer is due to each input (remember that the net inputs are the inlet and outlet flows and pressures - Q_{in}, Q_{out}, H_{in} and H_{out}-) has 100 delays per inputs (a time window of 0.3085 seconds, since the sampling time is the 3.085×10^{-3} milliseconds).

A. Training the TDNN

Due to the difficulty in obtaining training data from a real pipe (it is technically impossible to generate holes along the whole pipe), the authors propose to generate synthetic data directly obtained from the pipeline mathematical model, equation (7). Such training data was generated as follows: The pipeline was divided into 40 sections equally distributed, 40 sections of length. The first section at a distance of $L/40$, the second of $2L/40$, until covering the whole pipeline, this division length is chosen arbitrarily by the designer. Then, the leak magnitude search space was selected such that the maximum value, λ_{max}, induce a leak 10% of the nominal flow. A leak bigger than this percent is considered as a failure (a catastrophic breakdown of the systems ability to perform a required function under specified operating condition [15]) and this topic is beyond the scope of this paper. In the same way, the Search Space was divided in 40 section equally separated, so, a vector of the form $[\lambda_{max}/40 \; 2\lambda_{max}/40 \; \cdots]$. Now, the cross product between fictitious hole positions and the vector of leak size, yield 1600 training data, enough to obtain satisfactory results.

Finally, TDNN was trained with the well-known Levenberg-Marquardt algorithm discussed in Section III-B. In order to find an adequate number of hidden layers, the net is trained varying its number until 110 hidden layers are reached. Figure 3 shows the overall accuracy of each network. From this figure, we can see that having less than 70 hidden layers (i.e., a shallow neural network) is not very helpful. The most substantial improvement is obtained when 100 neurons to the network hidden layer are added, and adding more neurons do not have a significant impact on the accuracy of the network.

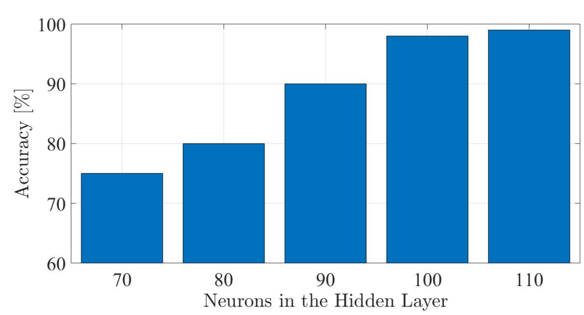

Fig. 3. Overall fault detection accuracy of neural networks with different number of neurons in the hidden layer

B. Simulation Results

In order to test the proposed approach, the pilot water pipeline designed at the CINVESTAV in Guadalajara, Mexico, was considered. The pilot pipeline is equipped with two water-flow (FT) and two pressure-head (PT) sensors at inlet and outlet of the pipeline, a 5 HP centrifugal pump connected to a variable-frequency driver fixed at 50 Hz, and three valves to emulate the effect of a leak. Fig. 4 below depicts a schematic diagram of the pipeline prototype (more information can be found in [2]).

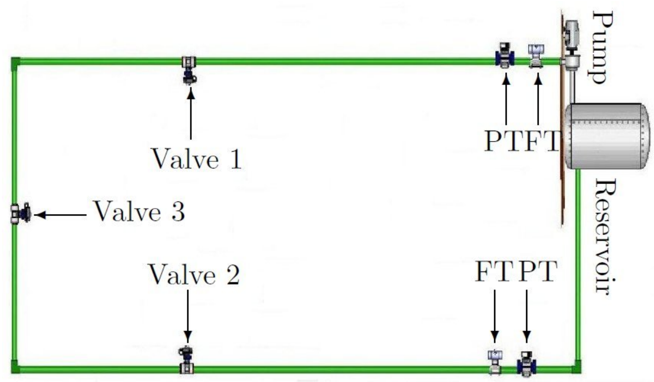

Fig. 4. Schematic diagram of the pipeline prototype

The main parameters of the pipeline system are shown in Table I

TABLE I
PIPELINE PROTOTYPE PARAMETERS

Parameter	Symbol	Value
Length between sensors	L	105.21 m
Internal diameter	D	6.54×10^{-2} m
Friction factor	f	2.062×10^{-2}
Gravity acceleration	g	9.81 m/s^2
Pressure wave speed	b	1435 m/s
Valve 1 distance	Δz_1	30.92 m
Valve 2 distance	Δz_2	43.64 m
Valve 3 distance	Δz_3	62.99 m

Ten simulation tests were carried out to validate the performance of the net. The scenario of each experiment is as follows. The mathematical pipeline model (7) serves to generate the synthetic data (The net inputs). When the simulation starts, λ is equal to zero (since the pipeline is no leaking) and $\Delta z = L/2$ is fixed arbitrarily (the initial condition of the system). The initial condition of the rest of the states, Q_1, H_2 and Q_3, were chosen such that the system is in steady-state conditions. Table II shows such values.

TABLE II
INITIAL CONDITION OF THE MODEL

Parameter	Symbol	Value
Pressure Head upstream end	H_{in}	$20.74; m$
Pressure Head downstream end	H_{out}	$10.56; m$
Inflow	Q_{in}	9.00×10^{-3} m^3/s
Outflow	Q_{out}	9.00×10^{-3} m^3/s
Leak Pressure Head	H_2	15.41 m

Once the experiment begins, a leak was induced at 20 seconds. The position of the leak for each test are $\Delta z_1 = 10\ m$, $\Delta z_2 = 20\ m$, $\Delta z_3 = 30\ m$, $\Delta z_4 = 40\ m$, $\Delta z_5 = 50\ m$, $\Delta z_6 = 60\ m$, $\Delta z_7 = 70\ m$, $\Delta z_8 = 80\ m$, $\Delta z_9 = 90\ m$ and $\Delta z_{10} = 100\ m$.

Random noise was added to the system input to check the algorithm robustness. Figure 5 shows the neural network prediction expected values for the leak positions. As can be seen, in all cases, the leak positions were well estimated, despite the noise in the input measurements. It is essential to point out that the prediction error increases when the leak is near the ends of the pipeline. The maximum gap is around three meters, less than 3% of the whole pipe; this is a very acceptable error in this sort of problems regarding the pipeline length and the fact that he input signals are corrupted by noise.

C. Real Time Experimental Results

In this section, the proposed LDI methodology is evaluated off-line by using some database coming from the pilot pipeline building at CINVESTAV previously mentioned. Three experiments were carried out on three different databases in order to assess the effectiveness of the method. The experiment starts in a free-leak condition and at time $t_l \approx 40\ s$ a leak is induced by opening the valve 1. The neural network start once the leak is detected i.e. $|Q_{in} - Q_{out}| > \delta$ ($\delta = 1.55 \times 10^{-}4\ m^3/s$ is chosen as the detection threshold which was selected considering the noise variance of the flow rate measurements). The Signal-to-noise Ratio or SNR (the ratio of the signal power to the background noiseSNR) of each input and output signal are shown in table III. The SNR was calculated as the ratio of the signal power to the background noise [16]:

$$SNR = \frac{E\left[s^2\right]}{\sigma^2}$$

where $E\left[\bullet\right]$ refers to the expected value and σ stands for the standard deviation of the noisy signal.

TABLE III
SIGNAL-TO-NOISE RATION OF THE INPUT AND OUTPUT SIGNALS

Variable	SNR
H_{in}	2.008×10^1
H_{out}	6.689×10^1
Q_{in}	1.338×10^2
H_{out}	1.432×10^2

Figures 6 and 7 depicts the evolution of the neural network inputs (upstream and downstream pressure head and flow rate, respectively). The results of the *LDI* scheme are shown through Figs. 8-10. Particularly, Fig. 8, Fig. 9 and Fig. 10, depict the leak position estimation for experiment one, two, and three (valve 1, 2 and 3) respectively. As can be seen, the leak position in all three cases is well estimated despite signal noise.

V. Conclusions

This paper addressed the problem of one leak isolation in a water pipeline using TDNN. This approach is an alternative to better computing performance application, as a neural network structure can have better efficiency using the parallelism of some electronic systems such as the FPGA.

The implemented model was trained only with flow and pressure measurements at the inlet and outlet of the pipe. Such training signals have been corrupted by random noise in order to emulate an actual scenario. It should be observed that synthetic data were produced by a simulator based on a mathematical pipeline model.

Despite the presence of noise in the measurements, the simulation results showed an outstanding network performance. Finally, the TDNN algorithm performs a proper leak position assessment using real-noise databases from a real pilot system.

In the context of future work, two points should be addressed: (i) the efficiency of the system will be tested in the FPGA device and (ii) the extension of the strategy to the location of two leaks.

References

[1] L. Billman and R. Isermann. Leak detection methods for pipelines. *Automatica*, 23:381–385, 1987.

[2] Ofelia Begovich, Alejandro Pizano, and Gildas Besançon. Online implementation of a leak isolation algorithm in a plastic pipeline prototype. *Latin American Applied Research*, 57(6):131–140, 2012.

[3] Tiantian Zhang, Yufei Tan, Xuedan Zhang, and Jinhui Zhao. A novel hybrid technique for leak detection and location in straight pipelines. *Journal of Loss Prevention in the Process Industries*, 35:157 – 168, 2015.

[4] J.A. Delgado-Aguiñaga, O. Begovich, and G. Besançon. Exact-differentiation-based leak detection and isolation in a plastic pipeline under temperature variations. *Journal of Process Control*, 42:114 – 124, 2016.

[5] Shen Tian, Juanli Du, Shuangquan Shao, Hongbo Xu, and Changqing Tian. A study on a real-time leak detection method for pressurized liquid refrigerant pipeline based on pressure and flow rate. *Applied Thermal Engineering*, 95:462 – 470, 2016.

[6] Yufeng Hao. A general neural network hardware architecture on FPGA. *CoRR*, abs/1711.05860, 2017.

[7] J. A. Roberson, J. J. Cassidy, and M. H. Chaudhry. *Hydraulic Engineering*. Wayne Anderson, 1998.

[8] M Hanif Chaudhry. *Applied Hydraulic Transients*. Springer-Verlag New York, 2014.

[9] G. Besançon, D. Georges, O. Begovich, C. Verde, and C. Aldana. Direct observer design for leak detection and estimation in pipelines. In *European Control Conference, ECC'07*, pages 5666–5670, Kos, Greece, 2007.

[10] Donald F Specht. A general regression neural network. *IEEE transactions on neural networks*, 2(6):568–576, 1991.

[11] Simon Haykin. *Neural Networks and Learning Machine*. Pearson, 2008.

[12] Alexander H. Waibel, Toshiyuki Hanazawa, Geoffrey E. Hinton, Kiyohiro Shikano, and Kevin J. Lang. Phoneme recognition using time-delay neural networks. *IEEE Trans. Acoustics, Speech, and Signal Processing*, 37:328–339, 1989.

[13] Xiaorong Huang, Weibin Zhang, Xiangmin Xu, Ruixiang Yin, and Dongpeng Chen. Deeper time delay neural networks for effective acoustic modelling. *Journal of Physics: Conference Series*, 1229:012076, may 2019.

[14] M. T. Hagan and M. B. Menhaj. Training feedforward networks with the marquardt algorithm. *IEEE Transactions on Neural Networks*, 5(6):989–993, 1994.

[15] Rolf Isermann. *Fault Diagnosis Systems An Introduction from Fault Detection to Fault Tolerance*. Springer, Heidelberg, Berlin, Germany, 2006.

[16] Athanasios Papoulis and S. Unnikrishna Pillai. *Probability, Random Variables, and Stochastic Processes*. McGraw Hill, Boston, fourth edition, 2002.

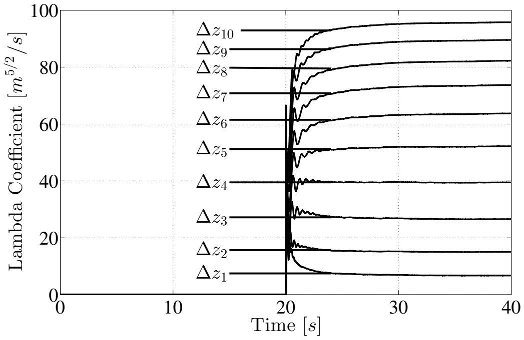

Fig. 5. Leaks position estimation, Δz, for the benchmark experiments

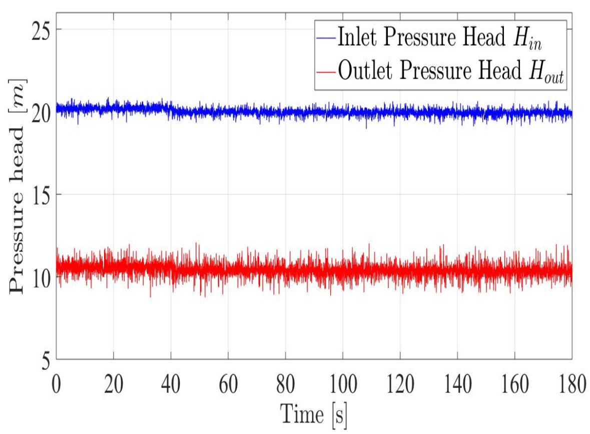

Fig. 6. Pressure head at inlet an outlet of the pipeline.

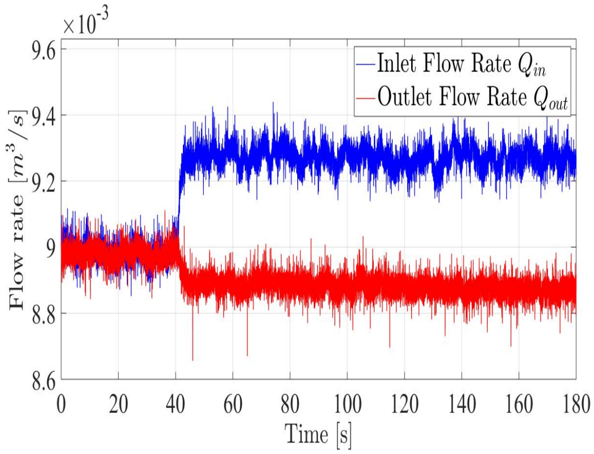

Fig. 7. Flow rate at inlet an outlet of the pipeline.

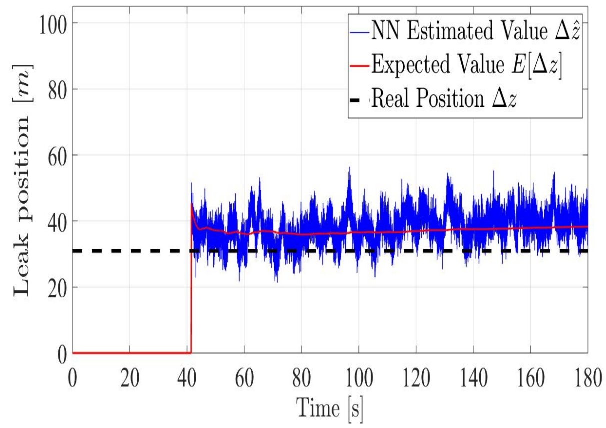

Fig. 8. Leak position estimation, Δz related to valve 1.

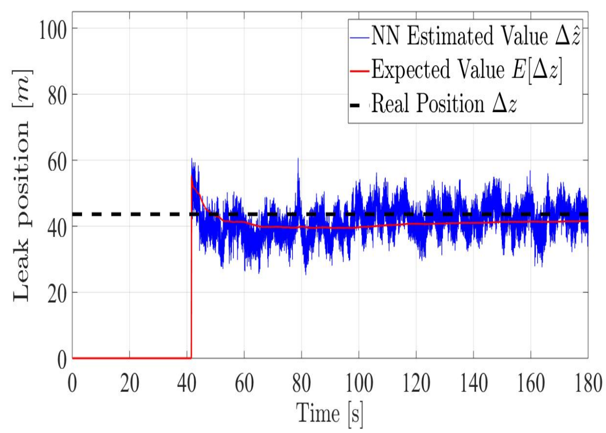

Fig. 9. Leak position estimation, Δz related to valve 2.

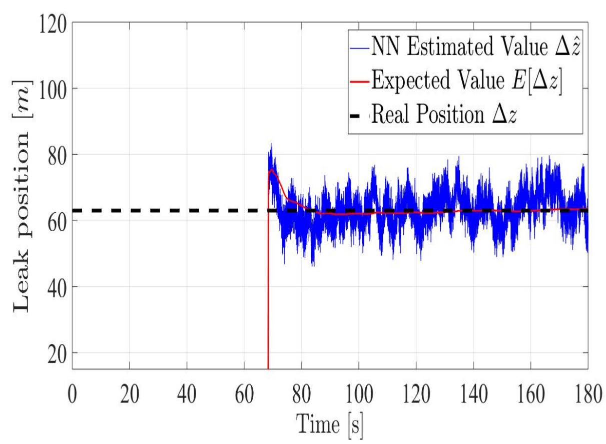

Fig. 10. Leak position estimation, Δz related to valve 3.