

Real Time Leak Detection and Isolation in Pipelines: A Comparison Between Sliding Mode Observer and Algebraic Steady State Method

G. Espinoza-Moreno, O. Begovich and J. Sanchez-Torres

Center for Research and
Advanced Studies of the
National Polytechnic Institute
Zapopan, Jalisco 45019
Telephone:+52(33)3777-3600

Email: [gespinoza, obegovi, dsanchez]@gdl.cinvestav.mx

Abstract—The purpose of this paper is to compare two different algorithms used to detect and isolate water leaks in a pipeline. One method is based on a Sliding Mode Observer and the second method is an Algebraic method obtained from the pipeline model in steady state. Because of the simplicity of both methods, they can be easily implemented. The methods were tested offline with real time data and the Algebraic method was also implemented online. Satisfactory results are shown through some experiments.

I. INTRODUCTION

Leak detection and isolation is an important issue due to the environmental and economical impact. For this reason several works have been developed over the last years [1]–[5]. For most of the cases, leak detection can be easily realized by using a mass balance. However, the leak location along the pipe can result in a difficult procedure.

Leak detection in pipelines can be divided in two categories, external methods and internal methods. External methods detect the leak outside the pipeline and can be expensive and slow, such as inspection by line patrols, dielectric cables and ultrasonic technologies. On the other hand, internal methods use sensing instruments to monitor pressure, flow, temperature, etc., which provide information to a computational pipeline monitoring system. These methods are fast, accurate, cheap and more sensitive. There are two types of Internal Methods, the first is the Fault Sensitive Approach (FSA), where the leak does not appear in the model and is based on residual correlation techniques like the method introduced by L. Billmann and R. Isermann [1]. The second Internal Method type is a method which uses a model that contains the faults and is called Fault Model Approach (FMA) and is more commonly used in recent investigation [2], [3], [6].

Both analytical methods (FSA and FMA) use a nonlinear model deduced from the Water Hammer equations [7] which can be discretized applying multiple methods such as Characteristics Method or Finite Differences [8], [9]. In this work, the model is spatially discretized by using Finite Differences and two different algorithms are used to detect and isolate a leak: A Sliding Mode Observer based on a Super Twisting Algorithm (STA) [10], to be precise an Uniform

Robust Exact Differentiator (URED) [11] and an Algebraic Steady State estimator [12]. They are tested in real time in a pipeline prototype which has flow and pressure sensors at its ends. Both algorithm performances will be compared. More precisely, the URED is a differentiator that yields finite-time and theoretically exact convergence to the derivative of an input signal with any initial condition, whenever the derivative is Lipschitz. It is important to point out that for leak isolation in [13] a STA differentiation was implemented, with the difference that in [13] the convergence time is not bounded by some constant independent of the initial conditions of the differentiator error. Another difference is that the experiments in [13] were made only in simulation, while in this work they are performed in real time.

To design the Algebraic algorithm we follow [12]. In that research an Algebraic algorithm is proposed to locate multiple non-concurrent leaks, but the algorithm was tested only in simulation. In our work the algorithm is applied in real time in order to isolate one leak. It is important to analyze the real time performance of the Algebraic algorithm due to its simplicity can be easily implemented in hardware like PLC, DSP, etc.

The paper is organized as follow: Section II introduces the model of the flow dynamics. In Section III the URED and the Algebraic algorithms are presented. A brief description of the prototype is presented in Section IV. Results in real time are provided in Section V and finally in Section VI some conclusions are stated.

II. MODEL

A. Modeling equations

In this Section the model that describes the behavior of a fluid in a pipeline, often known as Water Hammer equations, is introduced. The model consists of two equations, the conservation of mass equation (continuity equation) and the momentum equation. Assuming the fluid to be slightly compressible and the duct walls slightly deformable; the convective changes in velocity to be negligible; the cross section area of the pipe and the fluid density to be constant, then the dynamics of the pipeline fluid can be described by the following partial differential equations [7]:

Momentum Equation

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

Continuity Equation

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

where Q is the flow rate [m^3/s], H 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], a the speed of the pressure wave in the fluid [m/s], $\mu = \frac{f}{2DA}$, D the diameter [m] and f the friction factor.

On the other hand, the general equation which describes the behavior of a leak in a pipeline is deduced from the Bernoulli equation as

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

where Q_L is the flow through the leak, H_L is the pressure head at the leak point and the parameter λ is a constant related to the leak magnitude, which is a function of the orifice area and the discharge coefficient.

Due to the pipeline dynamics is affected by the leak, the flow through the leak is included in (2) as a mass balance

$$Q_1 = Q_2 + Q_L \quad (4)$$

where Q_1 and Q_2 are the flow before and after the leak, respectively.

Other important aspect is the friction coefficient f and the pressure wave speed in the fluid a . The friction coefficient can be calculated with the Haaland equation [14] which is defined as

$$f = \left(-1.8 \log_{10} \left[\left(\frac{\epsilon/D}{3.7} \right)^{1.11} + \frac{6.9}{Re} \right] \right)^{-2} \quad (5)$$

where ϵ is the roughness height [m], D is the pipe diameter [m] and Re the Reynolds number given by

$$Re = \frac{QD}{vA} \quad (6)$$

with v as the kinematic viscosity of the flow [m^2/s].

The pressure wave speed in the fluid is given by

$$a = \sqrt{\frac{\frac{K}{\rho}}{1 + \frac{DK}{eE}}} \quad (7)$$

where K is the elastic modulus and ρ the density of the fluid, while e is the wall thickness and E the elastic modulus of the pipe.

B. Spatial Discretization of the Model

A spatial discretization is introduced in order to obtain ordinary differential equations as follows:

$$\begin{aligned} \frac{\partial H}{\partial z} &\approx \frac{H_{j+1} - H_j}{z_j} \\ \frac{\partial Q}{\partial z} &\approx \frac{Q_j - Q_{j-1}}{z_j} \end{aligned} \quad (8)$$

Fig. 1 shows a spatial discretization of the pipeline in two sections. In this figure z_j , ($j = 1, 2$) represent the distance from the pipe beginning to the leak point and from the leak point to the pipe ending, respectively. Observe that $z_2 = L - z_1$ where L is the total length of the pipeline.

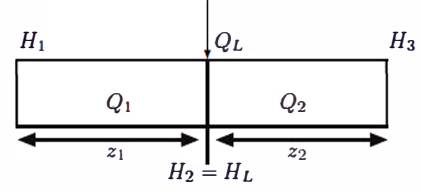


Fig. 1. Spatial discretization of pipeline in two sections with a leak.

Being the used pipeline prototype a non-straight pipe with several fittings, the length L is approximated by the Darcy-Weisbach equation as follows:

$$L = L_{eq} = \frac{2h_f Dg}{fV^2} \quad (9)$$

with L_{eq} as the Equivalent Straight Length [15], h_f the pressure drop by friction losses, which is the difference between the inlet and the outlet pressure head, and V the velocity of the flow.

Finally, using equations (8) and (9) in equations (1) and (2) and incorporating the model of the leak presented in equations (3) and (4), the following equations in state variables are obtained [4]:

$$\begin{aligned} \dot{Q}_1 &= -\frac{gA}{z_1}(H_2 - u_1) - \mu_1 Q_1 |Q_1| \\ \dot{H}_2 &= -\frac{a^2}{gAz_1}(Q_2 - Q_1 + \lambda \sqrt{H_2}) \\ \dot{Q}_2 &= -\frac{gA}{L-z_1}(u_2 - H_2) - \mu_2 Q_2 |Q_2| \end{aligned} \quad (10)$$

where $u_1 = H_1$ (inlet pressure), $u_2 = H_3$ (outlet pressure) and $y = [Q_1 \ Q_2]^T$. Notice that μ_1 and μ_2 are calculated with the friction factor at the inlet and outlet of the pipe, respectively.

III. URED AND ALGEBRAIC ALGORITHMS.

A. URED Algorithm

First, the equations to calculate the derivative of a signal with the URED [11] will be presented. Consider $f(t) = f_0(t) + v(t)$, where $f_0(t)$ is the base signal to be differentiated and $v(t)$ a bounded noise. Note that the derivative of $f_0(t)$ is Lipschitz. A state representation of the base signal is given by

$$\begin{aligned} \dot{\zeta}_0 &= \zeta_1 \\ \dot{\zeta}_1 &= \dot{f}_0 \end{aligned} \quad (11)$$

with $\zeta_0 = f_0(t)$ and $\zeta_1 = \dot{f}_0(t)$, the next URED to observe the system (11) is proposed:

$$\begin{aligned} \dot{\xi}_0 &= -k_1 \phi_1(\sigma_0) + \xi_1 \\ \dot{\xi}_1 &= -k_2 \phi_2(\sigma_0) \end{aligned} \quad (12)$$

where k_1 and k_2 can be chosen from the set

$$\kappa = \left\{ (k_1, k_2) \in \mathbb{R}^2 \mid 0 < k_1 \leq 2\sqrt{\mathcal{L}}, k_2 > \frac{k_1^2}{4} + \frac{4\mathcal{L}^2}{k_1^2} \right\} \cup \left\{ (k_1, k_2) \in \mathbb{R}^2 \mid k_1 > 2\sqrt{\mathcal{L}}, k_2 > 2\mathcal{L} \right\} \quad (13)$$

with $\mathcal{L} \geq |f_0''(t)|$ as a positive constant, $\sigma_0 = \xi_0 - \zeta_0$,

$$\begin{aligned} \phi_1(\sigma_0) &= |\sigma_0|^{1/2} \text{sign}(\sigma_0) + \beta |\sigma_0|^{3/2} \text{sign}(\sigma_0), \\ \phi_2(\sigma_0) &= \frac{1}{2} \text{sign}(\sigma_0) + 2\beta\sigma_0 + \frac{3}{2}\beta^2 |\sigma_0|^2 \text{sign}(\sigma_0) \end{aligned}$$

and $\beta \geq 0$.

Once the estimation of the derivatives is obtained, these can be applied to the equations which will be described bellow. From (10)

$$\dot{Q}_2 = -\frac{gA}{L - z_1} (u_2 - H_2) - \mu_2 Q_2 |Q_2| \quad (14)$$

we can see that

$$H_2 = \frac{L - z_1}{gA} (\dot{Q}_2 + \mu_2 Q_2 |Q_2|) + u_2. \quad (15)$$

Then H_2 is substituted in the first equation from (10),

$$\dot{Q}_1 = -\frac{gA}{z_1} (H_2 - u_1) - \mu_1 Q_1 |Q_1|. \quad (16)$$

Hence the variable z_1 can be derived as follows:

$$z_1 = \frac{-L(\dot{Q}_2 + \mu_2 Q_2 |Q_2|) - gA(u_2 - u_1)}{\dot{Q}_1 - \dot{Q}_2 + \mu_1 Q_1 |Q_1| - \mu_2 Q_2 |Q_2|}, \quad (17)$$

now if z_1 is known, it is easy to see in (16) that H_2 is also known, whenever Q_2 is available. Another variable that will be calculated is the constant λ obtained from the next equation introduced in (10)

$$\dot{H}_2 = -\frac{a^2}{gAz_1} (Q_2 - Q_1 + \lambda\sqrt{H_2}) \quad (18)$$

as

$$\lambda = \frac{1}{\sqrt{H_2}} \left(\frac{gAz_1 \dot{H}_2}{a^2} + Q_1 - Q_2 \right) \quad (19)$$

Since \dot{Q}_1 and \dot{Q}_2 are estimated by the URED, they need to be substituted by \hat{Q}_1 and \hat{Q}_2 , respectively in (15) and (17). Notice that Q_1 and Q_2 are available due to flow sensors installed in the prototype. Once the derivatives of these signals are obtained, z_1 and H_2 , which are substituted by \hat{z}_1 and \hat{H}_2 in (15, 17, 19), can be calculated. Finally differentiating \hat{H}_2 , λ (seen in (19) as λ) can be estimated. Note that \hat{H}_2 is continuous but not differentiable due to its estimation is a function of signals which result from an URED. To solve this problem \hat{H}_2 is filtered by a first order filter, making \hat{H}_2 differentiable.

B. Algebraic Algorithm

As it is well known, pressure in pipelines working at certain operating point is generally constant, consequently the flow in the pipeline is also constant. On the other hand, when a leak or a change at the operating point occurs there exists a short transient after which all the states of the system become constant. This leads us to design a leak detection and isolation algorithm where all the derivatives are neglected like the proposed in [12].

Using equations (15), (17) and (19) and considering that $\dot{Q}_1 = 0$, $\dot{Q}_2 = 0$ and $\dot{H}_2 = 0$, the next equations are obtained to be implemented in the Algebraic method:

$$\begin{aligned} z_1 &= \frac{-L(\mu_2 Q_2 |Q_2|) - gA(u_2 - u_1)}{\mu_1 Q_1 |Q_1| - \mu_2 Q_2 |Q_2|} \\ H_2 &= \frac{L - z_1}{gA} (\mu_2 Q_2 |Q_2|) + u_2 \\ \lambda &= \frac{Q_1 - Q_2}{\sqrt{H_2}} \end{aligned} \quad (20)$$

IV. PROTOTYPE

Before presenting the results, a brief description of the pipeline prototype built at the Center for Research and Advanced Studies in Guadalajara, Mexico (CINVESTAV-Guadalajara) [5] will be given. The prototype consists of a pump, a non-straight plastic pipeline, where seven sensors are installed. Also three valves are distributed along the pipe in order to simulate leaks to test the algorithms and finally we have a tank to store the fluid.

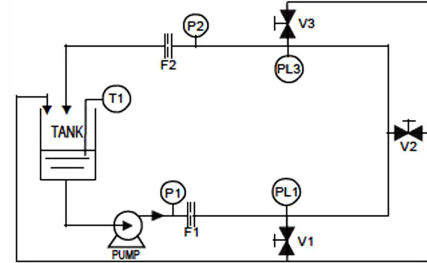


Fig. 2. Schematic diagram of the pipeline prototype.

In Fig. 2 all the parts that integrate the prototype are shown, where a pressure sensor (P1) and a flow sensor (F1) can be seen at the inlet of the pipe. At the outlet of the pipe the P2 and F2 sensors are installed, which have the same specifications as P1 and F2. In order to verify the Equivalent Straight Length of a leak position, by using Darcy-Weisbach, a pressure sensor (PL1) is mounted at the first valve (V1) and another one (PL3) is mounted at the third valve (V3). Ultimately, a temperature sensor is installed in the tank to monitor fluid temperature, which is very useful to adjust some parameters like the kinematic viscosity of the flow, the elastic modulus and density of the fluid, among others.

TABLE I. POSITION OF THE VALVES

	Real distance from the inlet of the pipe
Valve 1	17 m
Valve 2	33.5 m
Valve 3	50 m

The real length between inlet and outlet sensors is 68 meters. The distances from inlet sensors to pressure sensors mounted at the first and third valve are shown in Table I.

V. RESULTS

In order to test the URED and the Algebraic methods some experiments were performed with real data. Both methods were programmed in MATLAB environment, giving better results the Algebraic method. So in the first part of this section offline results are presented, while in the second part online implementation results obtained by the Algebraic method are reported through the LabView software.

A. Offline Implementation

To execute the offline implementation some synthesis parameters have to be given to the program. Table II shows the initial conditions of the experiment, and values like D , g , among others which remain constants during the experiment.

The algorithm starts executing the equations obtained in Section III once a threshold is over-passed, for both the URED algorithm and the Algebraic algorithm. This threshold was set in $8 \times 10^{-5} m^3/s$ and it is compared with the difference between inlet (Q_1) and outlet (Q_2) flow when the system is not leaking. When a leakage occurs that threshold is over-passed and the alarm to detect the leak is activated.

TABLE II. INITIAL CONDITION OF THE EXPERIMENT

Parameter	Value	Units
Q_1, Q_2	0.00828	m^3/s
H_{in}	17.063	mH_2O
H_{out}	8.995	mH_2O
D	0.06271	m
g	9.7819	m/s^2
Temperature	37.72	$^{\circ}C$
L_{eq}	88.28	m
e	0.01307	m
ϵ	7×10^{-6}	m
K	2.256×10^9	N/m^2
E	5.039×10^8	N/m^2
ν	6.8817×10^{-7}	m^2/s
ρ	993.054	Kg/m^3

Fig. 3 shows inlet and outlet flows in the pipeline and their changes at the moment of the leak occurrence. Note that inlet flow increases and outlet flow decreases when the leak appears as it is mentioned in [7].

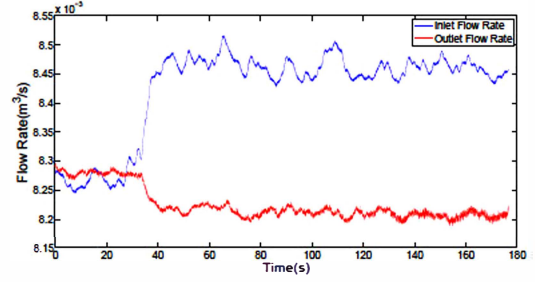


Fig. 3. Inlet and outlet flow of the pipe.

Fig. 4 presents inlet pressure (H_1) and outlet pressure (H_3) and in Fig. 5 the temperature of the experiment is displayed. As it can be seen, all the signal contain noise, being more significant in flow measurements.

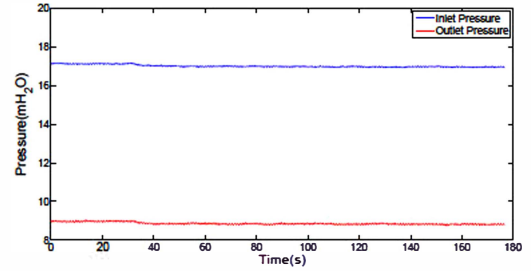


Fig. 4. Pressure head at inlet and outlet of the pipe.

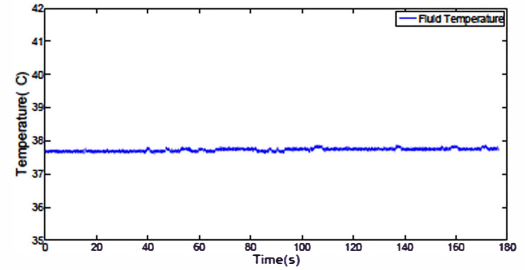


Fig. 5. Temperature of the water in the tank.

In both algorithms, as in other proposed in the literature [1]–[5], the pipe Equivalent Straight Length needs to be frozen at the moment the leak appears, otherwise the algorithm generates incorrect results.

The constants k_1 and k_2 need to be set in URED. In the case of Q_1 and Q_2 estimations, k_1 and k_2 are set in 0.02 and 0.0001. For the H_2 estimation the constants are fixed as $k_1 = 0.64$ and $k_2 = 0.2$. Observe that the constants are set in small values due to the dynamic of the system is slow, that means that the second derivative of the system states are also small, thus the differentiator become robust with respect to noise.

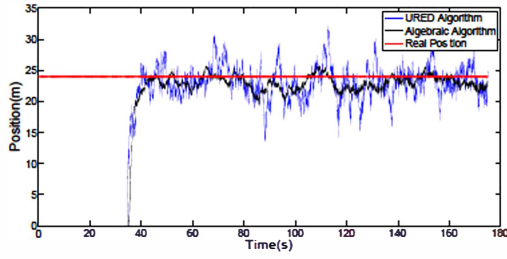


Fig. 6. Leak Position and its estimation.

Fig. 6 shows the leak estimation given by URED and Algebraic algorithms and the real position of the leak. Real leak position is located at 24 meters and is given by Darcy-Weisbach formula using the pressure of the sensor installed at valve 1. URED algorithm estimation is very noisy but the mean value of the signal is 23 meter. Algebraic algorithm estimation is also noisy but less than the URED algorithm estimation. It is easy to see that both estimations have similar dynamic and after a statistical study, the mean value of the Algebraic algorithm estimation is also 23 meters. Note that URED algorithm estimation is noisier due to not modeled and not desired dynamics are differentiated.

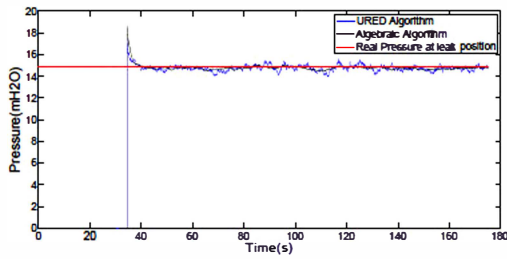


Fig. 7. Pressure at leak point and its estimation.

Estimation of pressure at leak point is displayed in Fig. 7. The real pressure at the leak point is $14.9 \text{ mH}_2\text{O}$ and both estimations resulted in approximately $14.68 \text{ mH}_2\text{O}$. In Fig. 8 the estimation of λ is shown giving as a result the same value as the real λ .

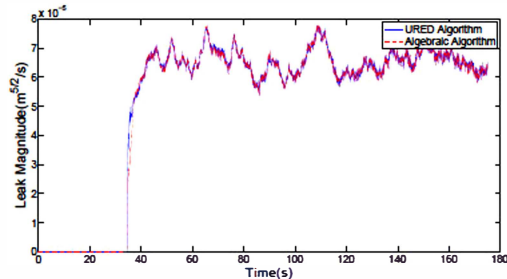


Fig. 8. Estimation of λ .

B. Online Implementation

Online implementation code was programmed in LabView software. Only Algebraic algorithm was implemented because of the good result obtained offline.

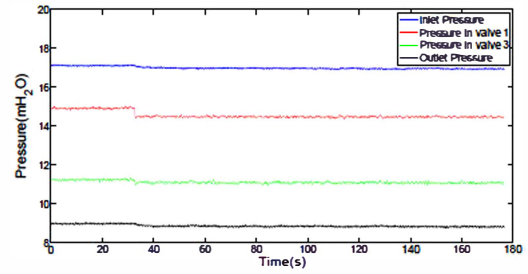


Fig. 9. Measured pressure by the sensors.

In Fig. 9 pressures provided by the sensors are observed. Four signals are displayed, inlet and outlet pressure, pressure at valve 1 and pressure at valve 3. In Fig. 10 we can see that the Equivalent Straight Length of the pipe is 88.28 meter before the leak occurrence and 86.6 meters after the leak appears, but for the algorithm equations 88.28 meters is used. Leak position is estimated at 25.7 meters while real position is at 24 meters. Remember that all the distances are calculated using Darcy-Weisbach formula, using measured pressures from the sensors.

Fig. 10 also shows an estimation with less noise, this can be possible thanks to a moving average filter, with the disadvantage of a delay in leak position estimation. There is a small difference between offline and online result, this is because the frozen length was different during online and offline test.

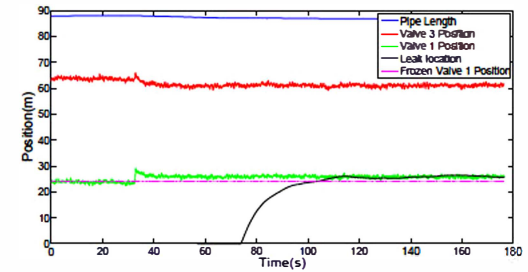


Fig. 10. Position of the valves and leak estimation.

Flows and temperature graphs for this experiment can be seen in Offline Subsection, in Fig. 3 and Fig. 5 respectively.

VI. CONCLUSION

Two algorithms were tested in real time, giving better results the Algebraic algorithm, then this algorithm is a good candidate for PLC and DSP applications. The URED algorithm performance was also acceptable. URED algorithm could give better results that Algebraic algorithm in a system where flows and pressures are not constants, for example a sine wave. As future work these leak detection and isolation systems will be tested simultaneously in order to obtain analytic redundancy in leak isolation.

ACKNOWLEDGMENT

The authors would like to thank for the financial support to the CONACYT CB project 177656.

REFERENCES

- [1] L. Billman and R. Isermann, "Leak detection methods for pipelines," *Automatica*, vol. 23, no. 3, pp. 381–385, May 1987.
- [2] A. Benkherouf and A. Y. Allidina, "Leak detection and location in gas pipelines," *Control Theory and Applications, IEE Proceedings D*, vol. 135, no. 2, pp. 142–148, March 1988.
- [3] C. Verde, "Minimal order nonlinear observer for leak detection," *Journal of Dynamic Systems, Measurement, and Control*, vol. 126, no. 3, pp. 467–472, Dec. 2004.
- [4] C. Verde, "Accommodation of multi-leak location in a pipeline," *Control Engineering Practice*, vol. 13, no. 8, pp. 1071 – 1078, 2005.
- [5] O. Begovich, A. Pizano, and G. Besançon, "Online implementation of a leak isolation algorithm in a plastic pipeline prototype," *Latin American Applied Research*, vol. 42, no. 2, pp. 131–140, 2012.
- [6] G. Besançon, I. Rubio Scola, M. Guillen, J.-F. Dulhoste, R. Santos, and D. Georges, "Observer-based detection and location of partial blockages in pipelines," in *Control and Fault-Tolerant Systems (SysTol), 2013 Conference on*, Oct 2013, pp. 694–699.
- [7] J. Roberson, J. Cassidy, and H. Chaudhry, *Hydraulic engineering*. Wiley, 1998.
- [8] Z. Kowalczyk and K. Gunawickrama, *Leak Detection and Isolation for Transmission Pipelines via Nonlinear State Estimation*. IFAC, 2000, pp. 921–926.
- [9] J. Korbicz, J. M. Kościelny, Z. Kowalczyk, and W. Cholewa, *Fault Diagnosis: Models, Artificial intelligence and Applications*. Springer, 2004.
- [10] A. Levant, "Sliding order and sliding accuracy in sliding mode control," *International Journal of Control*, vol. 58, no. 6, pp. 1247–1263, 1993.
- [11] E. Cruz-Zavala, J. Moreno, and L. Fridman, "Uniform robust exact differentiator," *Automatic Control, IEEE Transactions on*, vol. 56, no. 11, pp. 2727–2733, Nov 2011.
- [12] E. Garcia-Malacara, O. Begovich, and A. Pizano-Moreno, "Diagnosis of multiple non-concurrent leaks in pipelines based on a static model," in *World Automation Congress (WAC), 2012*, June 2012, pp. 1–6.
- [13] A. Navarro, O. Begovich, J. D. Sanchez-Torres, G. Besançon, and J. A. P. Murillo, "Leak detection and isolation using an observer based on robust sliding mode differentiators," in *World Automation Congress (WAC), 2012*, June 2012, pp. 1–6.
- [14] S. E. Haaland, "Simple and explicit formulas for the friction factor in turbulent pipe flow," *Journal of Fluids Engineering*, vol. 105, no. 1, pp. 89–90, 1983.
- [15] A. Navarro, O. Begovich, and G. Besançon, "Calibration of fitting loss coefficients for modelling purpose of a plastic pipeline," in *Emerging Technologies Factory Automation (ETFA), 2011 IEEE 16th Conference on*, Sept 2011, pp. 1–6.