IOP Conference Series: Materials Science and Engineering

PAPER • OPEN ACCESS

Water Pipeline Monitoring and Leak Detection using Flow Liquid Meter Sensor

To cite this article: R F Rahmat *et al* 2017 *IOP Conf. Ser.: Mater. Sci. Eng.* **190** 012036

Water Pipeline Monitoring and Leak Detection using Flow Liquid Meter Sensor

R F Rahmat[1], I S Satria[1], B Siregar[1], R Budiarto[2]

[1] Department of Information Technology, Faculty of Computer Science and Information Technology University of Sumatera Utara, Medan, Indonesia
[2] Department of Information System College of Computer Science and Information Technology, Albaha University, Saudi Arabia

romi.fadillah@usu.ac.id, 1114083iss@gmail.com, baihaqi@usu.ac.id, rahmat@bu.edu.sa

Abstract. Water distribution is generally installed through underground pipes. Monitoring the underground water pipelines is more difficult than monitoring the water pipelines located on the ground in open space. This situation will cause a permanent loss if there is a disturbance in the pipeline such as leakage. Leaks in pipes can be caused by several factors, such as the pipe's age, improper installation, and natural disasters. Therefore, a solution is required to detect and to determine the location of the damage when there is a leak. The detection of the leak location will use fluid mechanics and kinematics physics based on harness water flow rate data obtained using flow liquid meter sensor and Arduino UNO as a microcontroller. The results show that the proposed method is able to work stably to determine the location of the leak which has a maximum distance of 2 metres, and it's able to determine the leak location as close as possible with flow rate about 10 litters per minute.

1. Introduction

Water is an essential element for every organism, the needs for providing a good water distribution system is a must. Sometimes, the condition in certain location does not allow us to create a good water distribution system on the ground and the development of constructions causes the current water distribution system to residentials, offices, and industry premises through pipes under the ground.

Flow liquid meter sensor is one of the sensors that is used for this monitoring process. This sensor uses Hall Effect sensor inside it to measure the water flow rate and is placed on a pipe that has a diameter equal to the diameter of the sensor. The sensor will retrieve water flow data by analysing rotation count of the wheel. A microcontroller is required to process this data in order to know how the rate of water flow through [1]. Then the data will be sent by the microcontroller to the end node via internet access.

Clean water is a critical requirement in human life. Commonly, water distribution is done through underground pipes, where the water pipeline will become more difficult to control than the one in the open space. This situation will cause a permanent loss if there is a disturbance in the pipeline such as leakage. Leaks in pipes may be caused by several factors, such as the age of the pipeline, improper installation, and the condition of the environment (natural disaster, etc.). Therefore, an efficient solution is required to detect and to locate the damage in underground pipeline system.

2. Previous Research

Previous researches on water pipeline monitoring and detection in water pipes have been conducted before. Reference [2] conducted a research in monitoring the level of water flow, by utilizing web services and Zigbee as a communication device as well as some sensors such as level sensors, water flow sensors, and temperature sensor. In addition to web monitoring, the owner of the sensor can also get important information about the flow of water via SMS to a personal mobile phone number of the owner.

Reference [3] was conducted to detect leakage of water pipeline. The research investigated the impact of various pipe diameter on pressure of the water flow in the pipes and the temperature changes around the pipe. FSR sensor is used to measure changes in the pipe diameter, and temperature sensors are used to measure the temperature around the pipe. In this research, they used 40 mm diameter PVC pipe with a constant pressure of 3 bar.

The other work in Reference [4] analysed vibration in the pipe wall caused by collisions between the water flows to the pipe wall. Vibration is measured using an MEMS sensor. Leakage is analyzed by comparing the vibration of the normal water flow and the vibration when there is a leak in the pipe. Tests are conducted by varying the pressure from 3 to 10 bar with a constant water flow rate of 300 m^3/hr.

A research that compared the consumption savings of tap water using wireless sensor network has been conducted in Reference [5]. This research uses Rfbee sensors to gather data received from the water flow rate transducer, as a sender and a recipient of the data. Data collected by Rfbee sensors will be transmitted by wireless network to a computer connected directly to the sensor Rfbee. A research to monitor and control the water flow through a web server is carried out in Reference [1]. Monitoring and controlling is done by using Hall Effect Flow Sensors, Arduino, Raspberry PI, and Solenoid Electro Valve. Hall Effect Flow Sensor with Arduino will measure fluid flow, while Raspberry PI will control solenoid electro valve, which is used to close or to open the flow of fluid through the pipe.

3. Methodology

General architecture of the experiment set up in this research shown in figure 1.

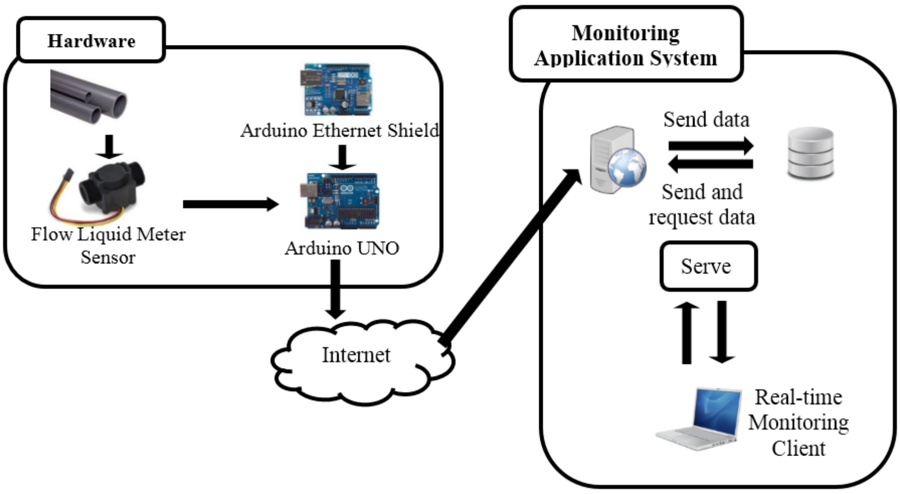


Figure 1. General System Architecture

Hardware

This section shows data retrieved by the sensor and then transmitted by Arduino to the monitoring application system using the Arduino Ethernet shield. The process starts from water that flows in the pipeline and go through the liquid flow meter sensor placed at the end of the pipe. The sensors then will collect data from the water which flows through it, and will be sent to Arduino. Arduino has some pins that serve as a place for data processing and power input. Flow Liquid Meter Sensor will transmit data to Arduino via digital pin 2, while GND pin and 5V pin will be connected to the sensor as the power supply. Then, Arduino calculates the water flow per second based on this data. The obtained data will be stored in a server on the PC's localhost first, then it will be immediately sent to the real-time monitoring application. Data from Arduino is stored to localhost server on the PC via an Ethernet shield, an additional module used to establish connection to the PC by accessing the PC's IP address [6]. When the connection is established, Arduino will access the PC and send the data using the POST method. Stackable Ethernet shield will be directly connected to the PC using the RJ-45.

A program to detect whether there is a leak in the pipe also will be created in Arduino. Leakage will be detected by Arduino when the normal initial flow of water continues to decrease slowly until the water flow stops at a certain speed, and until it is neither continues to decrease nor return to normal anymore. Arduino will process the data in order to obtain the location of the leak and send it to the monitoring application system. The data is measured from wheel rotation count inside the sensor caused by flow of water through sensor. The wheel rotation count is further processed in order to get the data of water flow passing through the sensor every second using equation (1), where n = number of wheel rotation, Q = liquid flow rate (m^3/sec), and c = calibration factor. Calibration Factor has its own constant value of 7.5, depends on the type of sensor used.

$$Q = \frac{n}{c} \tag{1}$$

Every water flow rate data received per second will be compared by Arduino. If there is a significant reduction in normal water flow rate, then the data of it will be stored temporarily until the flow is stable, stability is measured by comparing the normal and potential leakage data. Once the data is stable, the normal and the potential leakage data will be processed to get the velocity of the normal water and the flow after leakage by using equation (2) [7], where v is water velocity (m/s), Q is water flow rate (m^3/s), and A is area of the pipe (m^2).

$$v = \frac{Q}{A} \tag{2}$$

The data of water velocity and time during the reduction will be processed to know how much acceleration does the flow generated by using equation (3) [7]. And based on the acceleration and time data, the location of the leakage on the pipe can be obtained by using the equation (4) [7], where x is distance (m), v is final velocity (m/s), u is initial velocity (m/s), a is acceleration (m/s^2), and t is time (s).

$$v = u + at \tag{3}$$

$$x = \frac{v^2 - u^2}{2a} \tag{4}$$

The pseudocode of the process to detect location of the leak using Arduino can be described as in figure 2.

```
Keep the number of wheel rotation sensor ();
Calculate water flow rate ();
If (water flow rate  < normal water flow rate)
        If (water flow rate == leak water flow rate and
         the sending of leak detection data != 1)
                Calculate normal velocity () ;
                Calculate leak velocity () ;
                Calculate water acceleration ();
                Calculate location of leak ();
                Send leak detection data = 1 ;
        Else
                Leak water flow rate = water flow rate ;
                Increase time ;
Else if (water flow rate > normal water flow rate)
        normal water flow rate = water flow rate ;
        Time = 0 ;
        The sending of leak detection data = 0 ;
Else
        Time = 0 ;
        The sending of leak detection data = 0 ;
Send data to server ();
```

Figure 2. Pseudocode to detect the leakage location.

The location of the leak detection process in accordance with the pseudocode using sample data of the wheel rotation number on sensors that are sent sequentially to Arduino (79, 66.75, 66.75) can be illustrated as follows.

1. Keep the number of wheel rotation sensor (*pulseCount* = 79)
2. Change the number of wheel rotation sensor to get water flow rate (*flowrate* = 10.53 L/min)
3. Check whether the water flow rate data is smaller, larger, or equal normal flow rate data.
4. If value of water flow rate that first received is greater than the normal water flow rate data which initially is 0 and further if the water flow rate data is larger than normal water flow rate data, data will be saved as a normal water flow rate (*flowrateNormal* = 10.53 L / min). Time and leak detection data variable set to 0 to state that the data of leakage have not been sent.
5. If the water flow rate is not smaller and larger than normal, the data is considered as stable and does not have any problem. Time and leak detection data variable is set to 0 to state that the data about leakage have not been sent yet.
6. Water flow rate data has been checked and will be sent to the server.
7. Keep the next number of wheel rotation sensor (*pulsecount* = 66.75).
8. Repeat steps 2 to obtain water discharge (*flowrate* = 8.9 L / min) and step 3.
9. If the water flow rate data is smaller than the normal water flow rate, leakage is detected, check whether the data is the same as the water flow rate after leakage, check the data have been sent or not. If water flow rate is not the same as the water flow rate after leak, store it into the water flow rate after leak data (*flowrateLeak* = 8.9 L / min) and the time will increase. The time will continue to increase until water flow rate declared stable. Then repeat step 6.
10. Keep the next number of wheel rotation sensor (*pulsecount* = 66.75).
11. Repeat steps 2 to obtain water discharge (*flowrate* = 8.9 L/min) and step 3
12. Repeat steps 9. If water flow rate is equal to the water flow rate after leak, indicates that water flow rate as stable and check if the data of leakage has not been sent yet, then the Arduino will calculate the velocity of normal water flow rate (*velocityNormal* = 0.55 m/s), the speed of the water flow rate after leak (*velocityLeak* = 0.46 m/s), the water acceleration (acceleration = -0.09 m/s^2) and the location of the leak (*leakDistance* = 0.50 m) and then stores the value (*datasent* = 1) to declare that the data leakage has been sent to server. Then repeat step 6
13. The leak location detection will be done continuously until the sensor stops transmitting the data of the wheel rotation.

The location of leakage in this pipe will be sent by Arduino to localhost server on PC to be stored and will immediately transmit this data to the web server to then is represented back to the client in the form of notification and information that there had been a leak. This notification contains information about the date, time, position and distance of the leak location in pipe from the sensor.

Web Server

The Web server will act as the server and perform data processing within Arduino, databases, and client. Web server will receive the water flow rate data sent by the PC. When the Web server receives data of the leakage from the Arduino, the web server will store the information of the time when the leakage was detected, in accordance with the time on the database server side. Information about the time when a leak is detected and the location of the leak in the pipe will be sent by a web server to the client in a notification.

Real-time Monitoring Client

The client will be accessing the web page on the web server to monitor and only specific client can get permission to access the page. The page will contain the chart of the water flow through the pipe, and each chart is automatically updated every second and displayed in a certain time interval. In addition, the information about used sensor will be displayed on the web page. The client can also see a notification if there is a leak in the pipe and see the data of the previous water flow on the page.

4. Result and Discussion

System performance testing is conducted to determine the performance in monitoring and detecting the location of the leak in an underground pipe water flow. A prototype is designed to test the system performance by using PVC pipe ½" and with 5 meters' length. This pipe has five holes with a distance of each other 0.77 meters, 1.55 meters, 2.08 meters, 2.58 meters and 3.1 meters from the sensor. Every hole on this pipe will be tested with three different variations of water flow rate which is 5 litters per minute, 10 litters per minute and 15 litters per minute. The design of the prototype is shown in Figure 3.

Figure 3. Prototype design

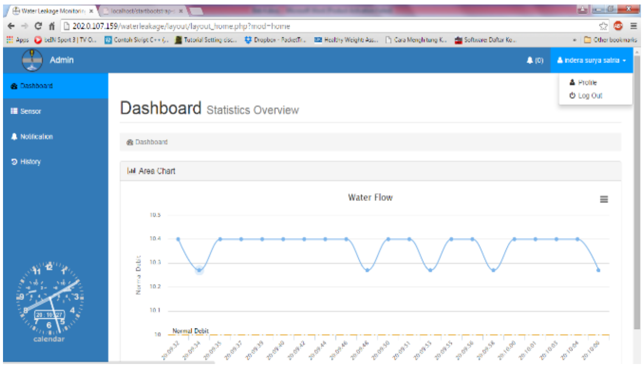
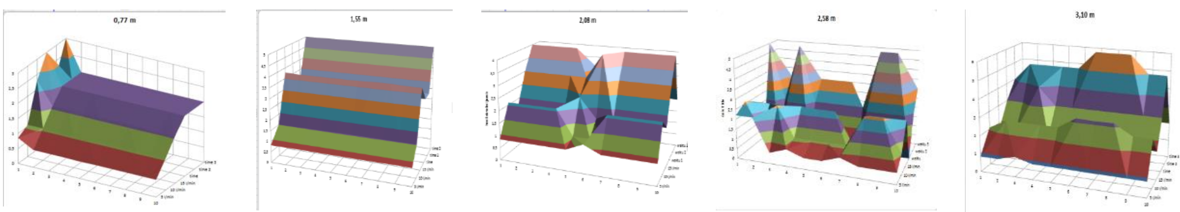


Figure 4. Results of water flow rate monitoring

At the first time of testing, normal water flows through the pipe and the sensor, with the condition of each holes are closed. A hole will be opened and the Arduino will detect that a leak is occurring. This detection result will appear on the page notification.

Testing the performance of the system at the time of water flow rate monitoring shows that the graph automatically moves every two seconds, which is supposed to be one. This caused by the delay time when data is sent from the PC to the web server and it is normal. The results of the monitoring system performance in the water flow is displayed in figure 4.

The system performance tests in detecting the location of the leak is performed 10 times for each variation of the water flow rate on every hole. Each of the test data is converted into the form of 3D graphics, as can be seen, there is a comparison of this data from every hole. The graph in the hole of 0.77 meters, 1.55 meters and 2.08 meters more stable when compared to the graph on the hole of 2.58 meters and 3.1 meters. Stable in this case is the results of the leak detection not changing yet. 3D graphics of the data produced by system testing is depicted in figure 5.



(a) (b) (c) (d) (e)

Figure 5. 3D graphics testing results (a) Distance hole 0.77 meters; (b) Distance hole 1.55 meters; (c) Distance hole 2.08 meters; (d) Distance hole 2.58 meters; (e) Distance hole 3.10 meter

The average results of the leakage location are compared with the actual location of the leak, to see how much difference it made. Based on the difference of average results for the leakage location, it can be seen that at a distance of 1.55 meters, 2.08 meters and 3.10 meters, smallest difference of average results belongs to water flow rate of 10 litters per minute. At a distance of 0.77 meters, smallest difference of average results belongs to water flow rate of 5 litters per minute and at a distance of 2.58 meters, smallest difference of average results belongs to water flow rate of 15 litters per minute. The difference of average results is summarized in table 1.

Table 1. Difference of average results

Actual Location (m)	Flow Rate (l/min)	Difference of Detection Result (m)
0,77	5	0,19
	10	0,284
	15	0,871
1,55	5	0,728
	10	0,025
	15	2.209
2,08	5	1,18
	10	0,22
	15	0,464
2,58	5	1,64
	10	0,92
	15	0,432
3,10	5	2,227
	10	0,514
	15	1,541

Based on figure 5, and the difference of average results in table 1, it can be observed that the system is able to work stably to determine the location of the leak that comes closest to the actual location of the leak in a maximum distance of 2 meters and using water flow rate of 10 litters per minute. This is due to the fact in determining the processing time of water flow rate reduction from the normal until the stable water flow rate after leak is more accurate if the water flow rate is 10 L/min.

5. Conclusions and Future Work

Computation of science, especially fluid mechanics and kinematics physics is implemented on microcontroller. Arduino chip-based ATmega328P can be used to detect the location of leaks in pipes by using a data rate of water flow and the system is able to work stably to determining the location of the leak with a maximum distance of 2 meters and it can determine the location of the leak closest to the actual location of the leak with an average flow rate of 10 litters per minute.

One plan in the future work is to extend the detection ability to more than 2 meters, which later on can be developed by considering the time duration of the reduction in water flow rate when there is a leak so the accuracy of the leakage location is improved. Another future work is to implement the detection system to branched pipelines as well as using various water flow rates.

References

[1]. Suresh N, Balaji E, Anto K J and Jenith J 2014 Raspberry PI based liquid flow monitoring and control *Int. J. Research in Engineering and Technology (IJRET)* **3** 122-125

[2]. Rani M U, Kamalesh S, Preethi S, Shri C K C and Sungaya C 2013 Web based service to monitor water flow level in various applications using sensors *Int. J. Biological, Ecological and Environmental Sciences (IJBEES)* **2** 119-122

[3]. Sadeghioon A M, Metje N, Chapman D N and Anthony C J 2014 SmartPipes: Smart wireless sensor networks for leak detection in water pipelines *J. Sensor and Actuator Network (JSAN)* **3** 64-78

[4]. Rizwan M and Paul I D 2015 Leak detection in pipeline system based on flow induced vibration methodology in pipeline *Int. J. Science and Research (IJSR)* **4** 3326-3330

[5]. Yano I H, Oliveira V C, Araujo E V Campagnuci A G, Fabiano B and Demanboro A C 2014 Wireless sensor networks for measuring the consumption of save water taps *American J. Applied Sciences* **11** 899-905

[6]. Chaudari R P and Chakraborty S 2014 Ethernet based, industrial furnace control and data acquisition *Int. J. Emerging Technology and Advanced Engineering (IJETAE)* **2** 124-128

[7]. Verma H C 1999 *Concepts of Physics* (Bharati Bhawan: Kanpur)