Securing Oil Tank Level Control System against Adversary Attacks using Fuzzy Logic Technique

P. Enyindah
Department of Computer Science
University of Portharcourt, Nigeria
Rivers State, Nigeria

C. Onukwugha
Department of Computer Science
Federal University of Technology, Owerri
Imo State, Nigeria

E.N. Osegi
Department of Information Technology
National Open University of Nigeria (NOUN)
Nigeria

ABSTRACT
A framework is proposed for a secure industrial process control system for an oil tank using a fuzzified technique in Matlab-Simulink. Specified start messages were encrypted and decrypted using a generic equation in Matlab m-file. Industrial process is simulated using a fuzzy-logic inference system in a Simulink model. There are two inputs to the fuzzy controller: the rate of change of the oil level and the difference between the actual and the required oil levels. The controller uses these inputs to control the inflow rate. A Rule Viewer is used to show dynamically which rule the fuzzy controller is using to set the flow rate. The required oil level is also set dynamically at run-time once the correct decoder equation has been specified. The results of the simulation showed high degree of stability in process variables before and after simulations of adversary attacks on the system.

General Terms
Pattern Recognition, Security, Algorithms

Keywords
Cyptographic model, fuzzified encryption-decryption, generic equation, oil level, adversary attacks

1. INTRODUCTION
The objective of process control in the industries mainly is about maintaining regulatory control and achieving certain economic goals in the face of measured and immeasurable differences within the product and process quality constrains. One of these core goals is "Security". The increased use of computer networked devices in control systems and the use of internet as key communication backbone have brought benefits in communication and for process control in general. The ability to share information, decision making concerning production stoking and distribution has been greatly improved [1].

New information technologies have recently been introduced for control systems based on open standards like Visual Basic, Java, Open Database Connectivity (ODBC), Ethernet communication and the use of internet based control system. Internet or web based control system uses some internet protocols in the application layer to monitor and control process and production plants from remote distances. Older control systems were based on closed protocols implemented by vendors of Supervisory Control and Data Acquisition (SCADA), Distribution Control Systems (DCS), and Programmable Logic Controller (PLC) systems.

However, the new common technology and the use of internet introduced many challenges in the area of industrial security, such as cyber threats [2, 3]. In order to secure such systems, reliable algorithms, models, and/or frameworks that are dynamic and incorporate knowledge of process control parameters is thus needed to bring security in the production process. Such an algorithm or framework should be able to enforce the goals of security- namely; prevention, detection and recovery, model the process, as well as capture potential security breaches or threats before they occur.

As a consequence of the technological improvements in process control and production operations, cyber-engineered sensor and actuator networks (CSANs) are being introduced in safe-critical control applications. That means that, their failure can cause irreparable harm to the physical system being controlled and the people who depend on it. Some examples of these critical systems are given by [4, 5, 6] and they include:

i. Electric power systems and distribution;
ii. Oil and gas storage and transportation;
iii. Water supply and waste water distribution systems;
iv. Transportation systems;
v. Telecommunications;
vi. Emergency services (including medical, police, fire, and rescue);
vii. Banking and finance.

Most of the analyses of the possible failures in those systems have been done in terms of random failures (reliability) [7]. However, the wireless nature of CSANs introduce some vulnerabilities from the security point of view, an example is, the fact that a malicious agent (hacker) could take remote control of some of the wireless nodes or sensors (motes) and use them deliberately to cause a wrong behavior or damage to the plant. In general, the attacks launched by an adversary can be more dangerous than random failures as these malicious attacks will be calculated to damage or produce specific operating errors.

It is of utmost importance to guarantee the stability and performance of these networked control systems prone to those malicious attacks, particularly when it comes to critical infrastructures such as power networks, oil infrastructure and water distribution systems. Moreover, as it is said in [4], there is a need of developing realistic and rational adversary models in order to test, investigate the vulnerabilities and improve the security of existing cyber-physical systems. One step in this direction has been in the development of standard centralized controllers framework using realistic adversary models as

attack mechanism [8].
This paper proposes a Fuzzified Encryption Decryption (FED) model for possible cyber-connected oil-tank level controller systems.

2. PROBLEM STATEMENT

Since most modern process automation systems are networked, there is the possibility of cyber-automated crimes. Currently there is a need for a more secure control system solution due to the frequent sabotage on automated process control plants by hoodlums and cyber criminals. In light of these challenges, a framework, FED, is developed to serve as a platform for protecting the integrity of potential threatened industrial process systems.

3. FED METHODOLOGY

In this section, a fuzzified encryption-decryption framework is presented. This framework makes use of some specific assumptions for a oil-level controller system, and builds on the theory of liquid tank -level controllers and fuzzy sets for its formulations.

The purpose of this section is to provide details of methodology and approaches to completing this research. An analytic framework for the study and implementation of encryption/decryption algorithms for the oil tank level control system were developed. A fuzzy logic based Tank System model was also implemented to study its resistance to system-level parameter control attacks. The system will take into account system limitations/assumption.

3.1 Assumptions

The following assumptions were proposed:

i. Number of users (channel capacity, n) is finite
ii. Normal operation is defined as normal operator flow rate setting between 0.65 and 0.66 m^3/s
iii. Normal tank level is defined between 7 and 9 m high
iv. Control is possible on any one channel
v. Attack is possible on any one channel
vi. Attack on system occur at system time as normal operation
vii. Attack on system flow rate is additive
viii. Maximum number of attack on a system at any given time is (n-1).

3.2 Tank Model Development for Simulation

3.2.1 Liquid Level Model

Consider a generic liquid level control application shown in Figure 1.

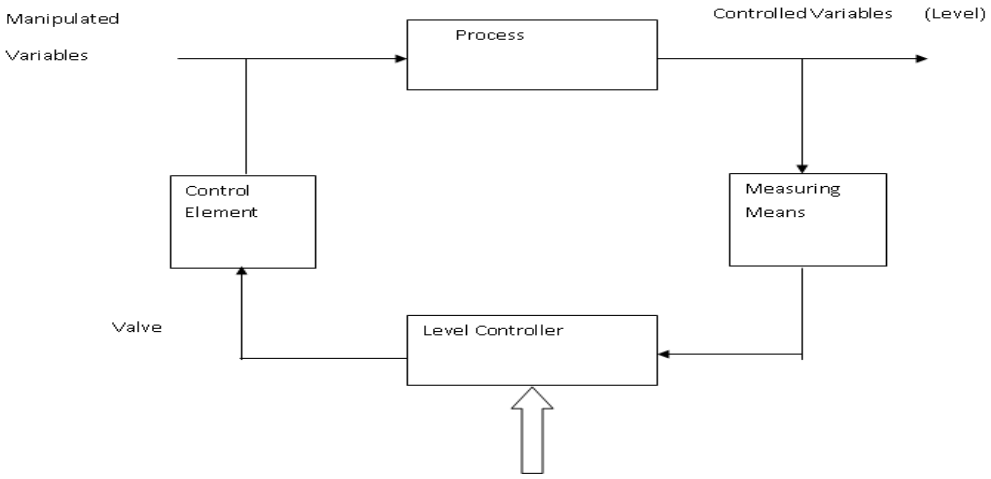

Fig 1: A typical industrial Liquid Level control problem

Also, consider the liquid-tank system shown in Figure 2:

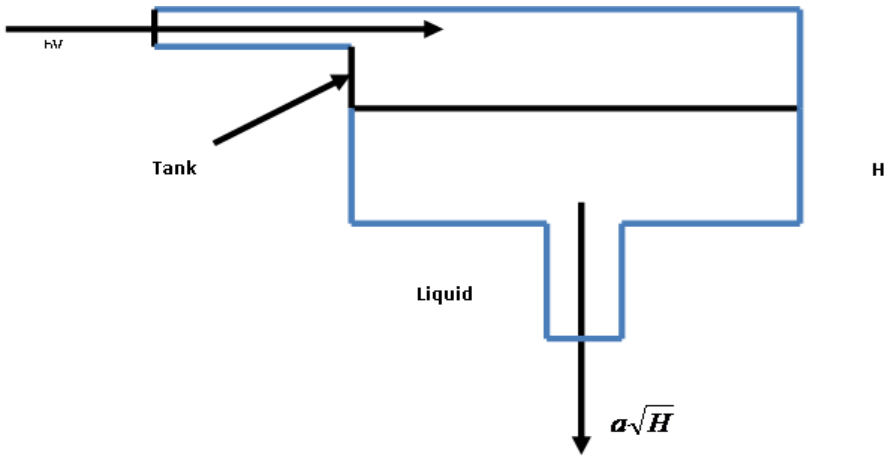

Fig 2: Schematic Diagram for a typical Liquid-Tank System

A differential equation for the height of liquid in the tank, H, is given by:

$$\frac{d}{dt}Vol = A\frac{dH}{dt} = bV - a\sqrt{H} \quad \text{..... (1)}$$

Where
Vol: is the volume of liquid in the tank,
A: is the cross-sectional area of the tank
b: is a constant related to the flow rate into the tank, and
a: is a constant related to the flow rate out of the tank.
The equation describes the height of liquid, H, as a function of time, due to the difference between flow rate into and out of the tank. The equation contains one state, H, one input, V, and one output, H. It is nonlinear due to its dependence on the square-root of H, linearizing the model, using Simulink Control Design, simplifies the analysis of this model [9].
The level is sensed by a suitable sensor and converted to a signal acceptable to the controller. The controller compares the level signal to the desired set-point temperature and actuates the control element. The control element alters the manipulated variable to change position of the valve so that the quantity of liquid being added can be controlled in the process. The objective of the controller is to regulate the level as close to the set point as possible.

3.2.2 Modelling the Single Tank System (Analytic Framework)

Initially, consider a coupled tank with an imaginary valve A being closed, and real valve B open. This system is a single tank process that can be drawn as shown in Figure 3.

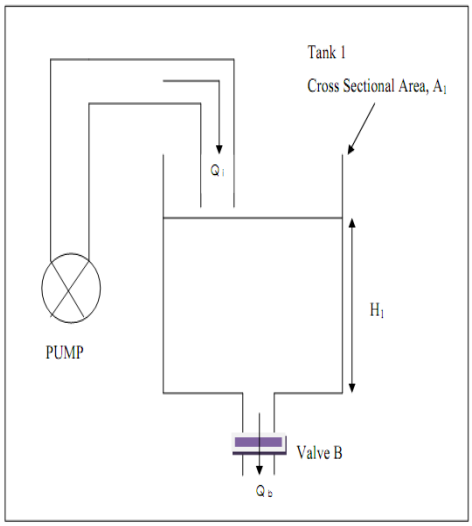

Fig 3: Single Tank System

The system model is determined by relating the flow into the tank to that leaving via valve B. Hence, Qi – Qb = rate of change of liquid level volume and

$$Q_i - Q_b = \frac{dV}{dt} = A\frac{dH_1}{dt} \quad \text{..... (2)}$$

where,
A: is cross sectional area of tank 1.
V1: is volume of liquid in tank 1 (V1 = A.H1).
Q i: is pump flow rate.
Q b: is flow rate out of valve B.
If valve B is assumed to behave like a standard sharp edged orifice, then the flow through valve B will be related to the liquid level in the tank, H1, by the expression,

$$C_{db.ab}\sqrt{2g.H_1} = Q_b \quad \text{..... (3)}$$

where,
a_b: is cross sectional area of the orifice. This represents the dimensions of valve B and the flow channel in which it is mounted. Because this dimension changes along the length of the channel, a_b would have to be taken to be the mean value.
c_{db}: is discharge coefficient of valve B. This coefficient takes into account all liquid characteristics, losses and irregularities in the system such that two sides of the equation balance.
g: is gravitational constant = 0.98 m/sec^

The orifice relationship (3) assumes C_{db} is a constant and therefore that Q_b is proportional to the square root of the level H1 for all possible operating condition. In a practical valve the flow rate Q_b will be some general non linear function of level H1.

$$Q_b = f(H_1) \quad \text{..... (4)}$$

Combining equation 2 and 4 gives,

$$A.\frac{dH_1}{dt} + f(H_1) = Q_i \quad \text{..... (5)}$$

The system model, Equation (5) is a first order differential equation relating input flow rate Qi, to the output liquid level, H1. In order to make it useful for control systems purposes it must be linear equation by considering small variations about a desired operating level of liquid in the tank.

3.2.3 Linearizing the Liquid Level Operating System

Let, H1 = H1' + h1
From the Figure 4, H1' is the normal operating level and is a constant H1 is a small change about that level. Then, for small variations of h1 about H1', can approximate the function f(H1) by the straight line tangent at H1'.
Let the inflow Qi consist of a steady component Qi' plus a small change qi, then if Qb' is the steady state outflow corresponding to h1, we can rewrite equation 5 as:

$$A.\frac{dH_1}{dt} + Q'_b + q_b = Q'_i + q_i \quad \text{..... (6)}$$

This can be rewritten, with reference to figure 3 as,

$$A.\frac{dH_1}{dt} + f(H'_1) + h_1.D = Q'_i + q_i \quad \text{..... (7)}$$

where the coefficient is the slope of the valve characteristics at the level H1'

$$D = \frac{\delta f(H'_1)}{\delta H_1} \quad \text{..... (8)}$$

When the level is constant, with qi=0 and h1=0, then equation 9 gives the steady state relation for flow and level,

$$f(H_1') = Q_i' \quad \text{..... (9)}$$

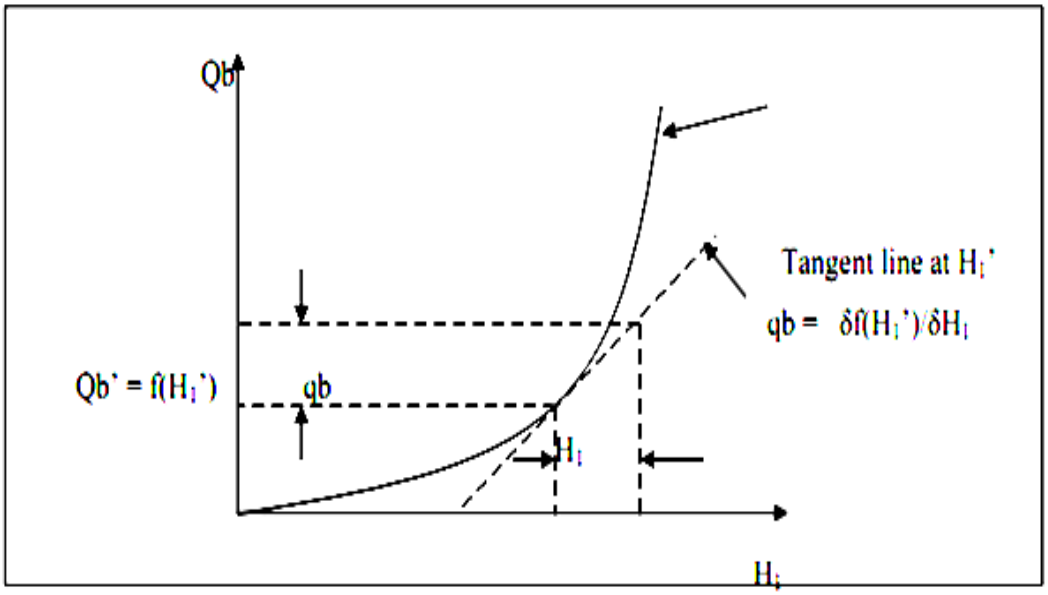

Fig 4: Linearization of the Liquid Level Operating System

Subtracting equation 9 from equation 7 and then rearranging gives the linear first order differential equation for the single tank system,

$$A.\frac{dh_1}{dt} + h_i.D = q_i \quad \text{..... (10)}$$

Given;

$$K_b = D^{-1} = 1/D$$

$$D = 1/k_b$$

Time constant T = A/D; A = T.D

$$(T.D\frac{dh_1}{dt}) + h_1.\frac{1}{k_b} = q_i \quad \text{..... (11)}$$

$$(T.\frac{1}{k_b}\frac{dh_1}{dt}) + h_1.\frac{1}{k_b} = q_i \quad \text{..... (12)}$$

$$\frac{1}{k_b}[(T.\frac{dh_1}{dt}) + h_1] = q_i \quad \text{..... (13)}$$

$$(T.\frac{dh_1}{dt}) + h_1 = k_b q_i \quad \text{..... (14)}$$

$$[T.s\,h_1] + h_1 = k_b.q_i$$

$$h_1[T_s + 1] = k_b.q_i$$

Taking Laplace transforms gives the single tank system transfer function,

$$h_1(s) = \frac{k_b}{T_s + 1}.q_i(s) \quad \text{..... (15)}$$

where, T is the Time Constant of the system
given by, T = A/D and kb is given by; Kb = D-1
where;

A = cross sectional area of tank A
D = slope at the normal operating level.
T = time constant.

3.3 Modelling for Encryption and Decryption

Encryption is modelled using classical method based on a generic mathematical formula, [11]:

$$C_{encrypt} = (A_m * 21) + 21 \quad \text{..... (16)}$$

The decryption of control messages (flow rate specification(s)) is thus simply the inversion of Equation 16 above:

$$D_{encrypt} = (C_{encrypt} - 21)/21 \quad \text{..... (17)}$$

where,

A_m = the message (system-control parameter) signal

Encryption and Decryption algorithms are modelled in an Embedded Matlab Function block as an Ethernet transmit and receive subsystems.Users can access the system and control system flow rate depending on which channel they are operating from. The input channels can be modelled with the aid of multiple Constant and Slider gain blocks in Simulink (see Figure 5). These channels then form a network of possible nodes through which control signals may transit. With this structure, it is thus possible to simulate multiple attacks on the system by varying one or more of the channel input flow rate parameters.

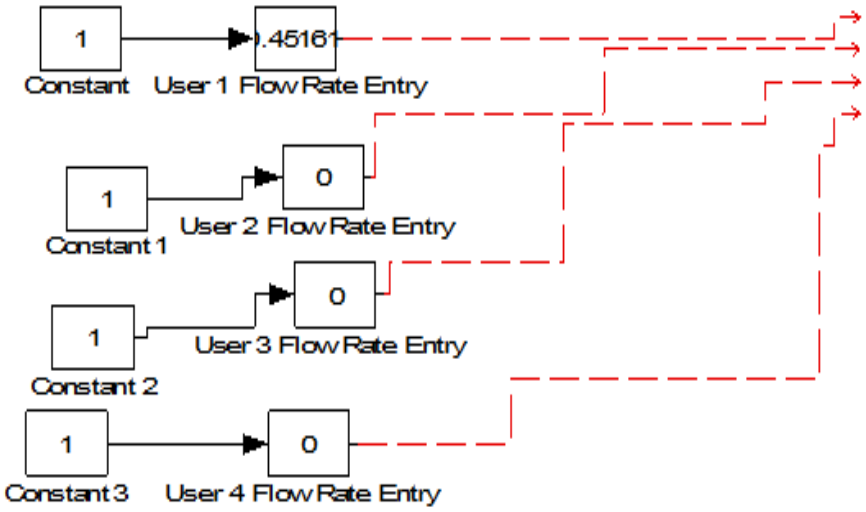

Fig 5: Input Channel Modelling in Simulink

3.4 Modelling the Fuzzy Logic Control (FLC) and Fuzzy Inference System (FIS)

The FLC developed is a single-input single-output controller. The input is the deviation from set point flow rate and system output flow rate fi (t), and fs (t) respectively. The FLC is implemented in a discrete-time form as shown in Figures 6 and 7 respectively.

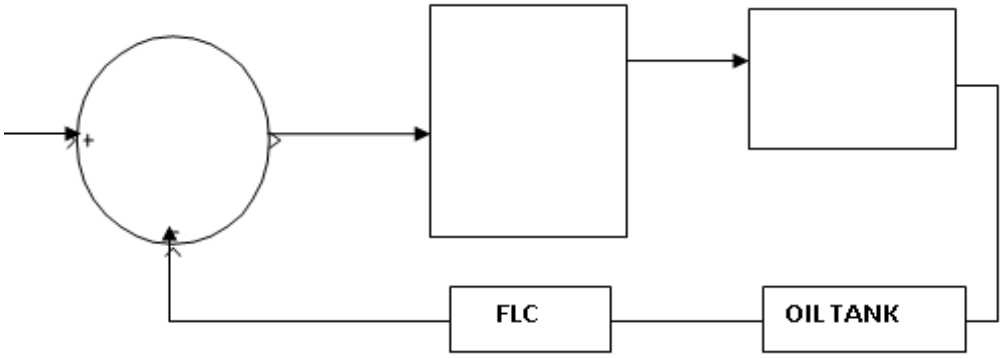

Fig 6: FLC Model

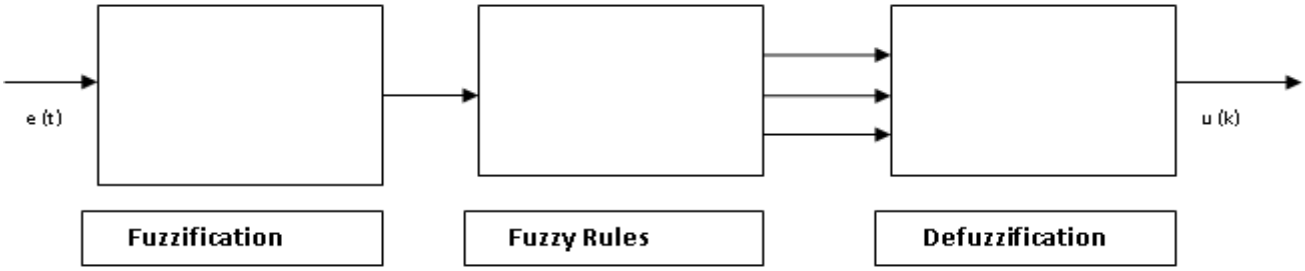

Fig 7: FLC Operational Structure

3.4.1 Rule Development

Our rule development strategy for systems with time delay is to regulate the overall loop gain to achieve a desired step response. The output of the FLC is based on the current input, e (k) without any knowledge of the previous input and output data or any form of model predictor. The main idea is that if the FLC is not designed with specific knowledge of mathematical model of the plant, it will not be dependent on it. The rules developed in this paper are able to compensate for varying time delays on-line by tuning the FLC output membership functions based on system performance. The rules and membership functions of the FLC are developed using an intuitive understanding of what a PI controller does for a fixed delay on a first order system [11]. Rules are implemented in the Matlab using Matlab Rule-Viewer.

3.4.2 Membership Functions

The FLC membership functions are defined over the range of input and output variable values and linguistically describes the variable's universe of discourse as shown in Figures 8 to 10 (for the input case only) while the set of rules for fuzzy logic control are shown in Figure 11.

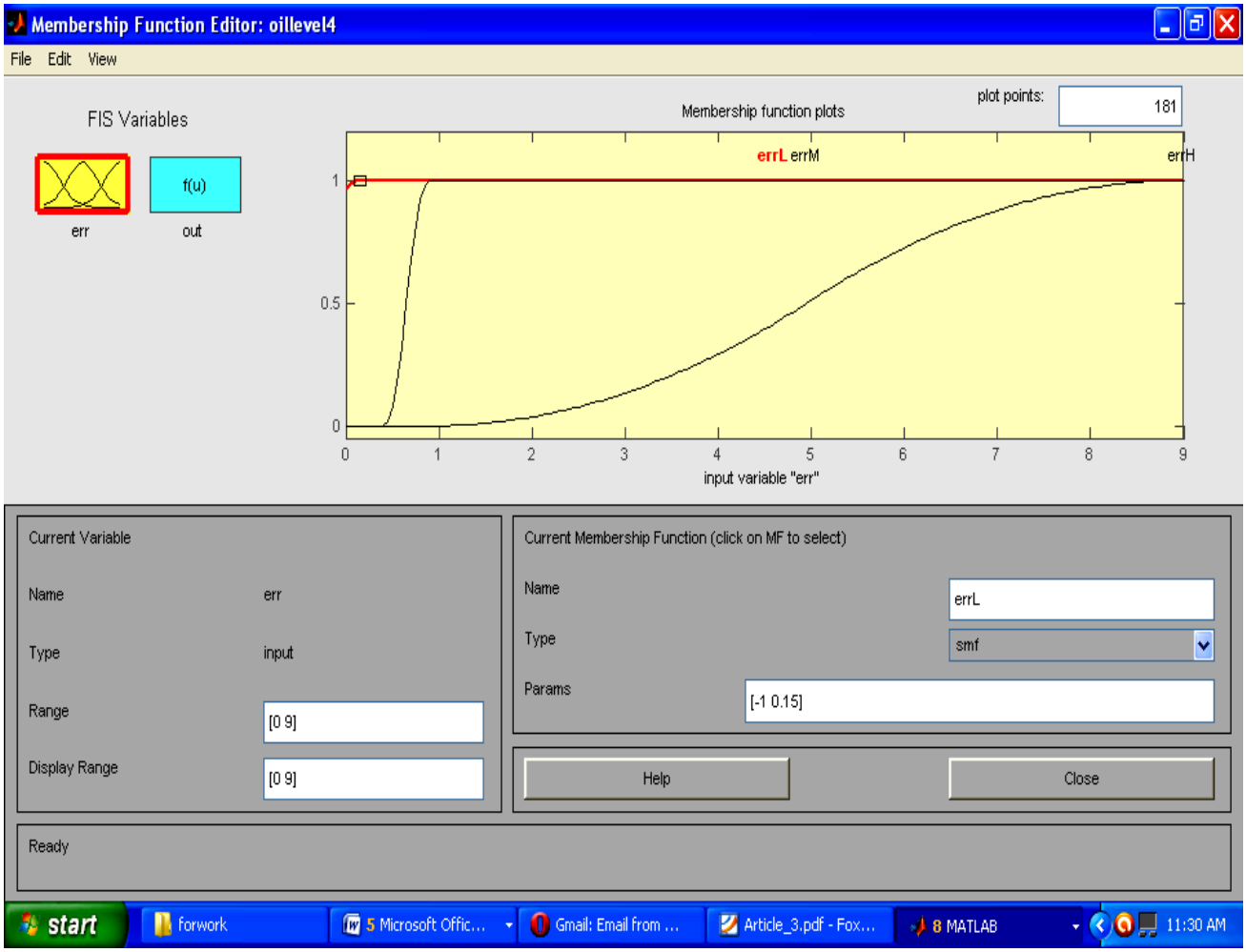

Fig 8: Input Membership Function Plot for Low Flow rate case (errL)

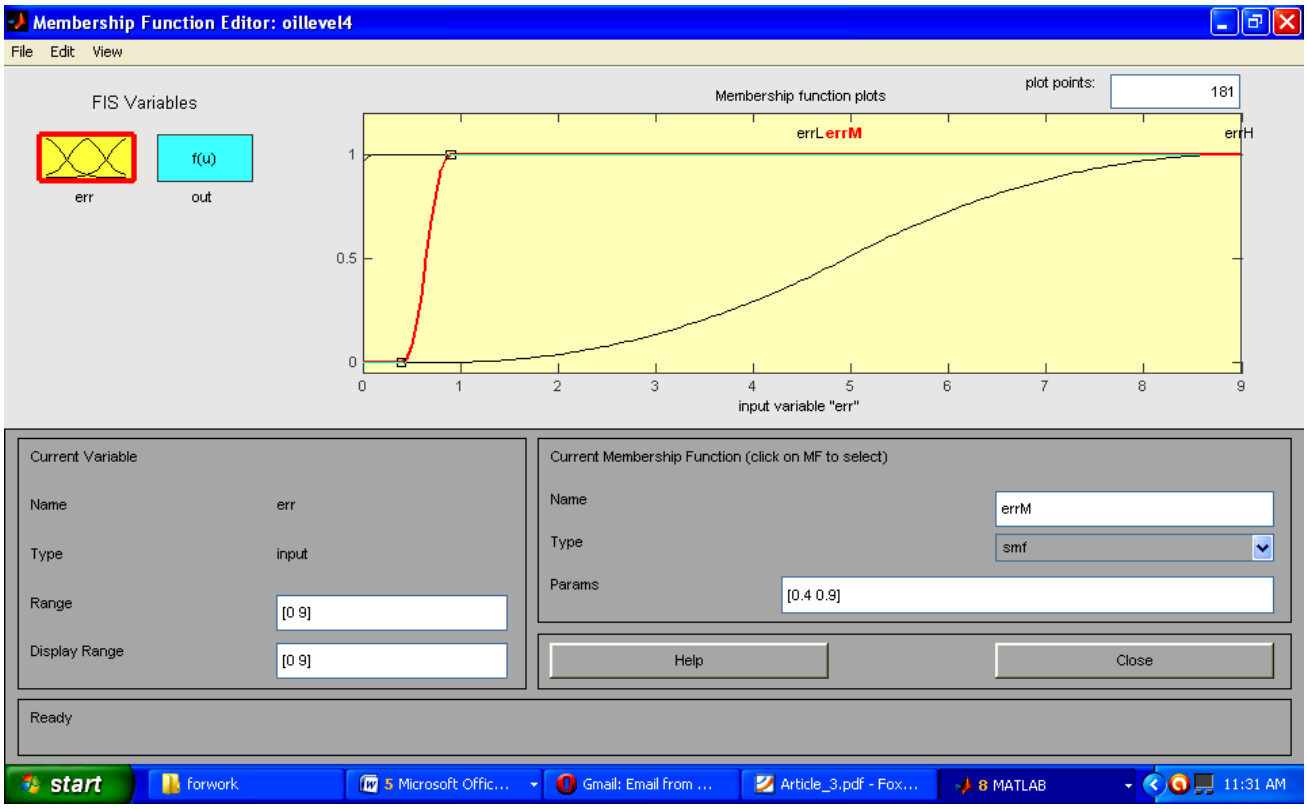

Fig 9: Input Membership Function Plot for Middle Flow rate case (errM)

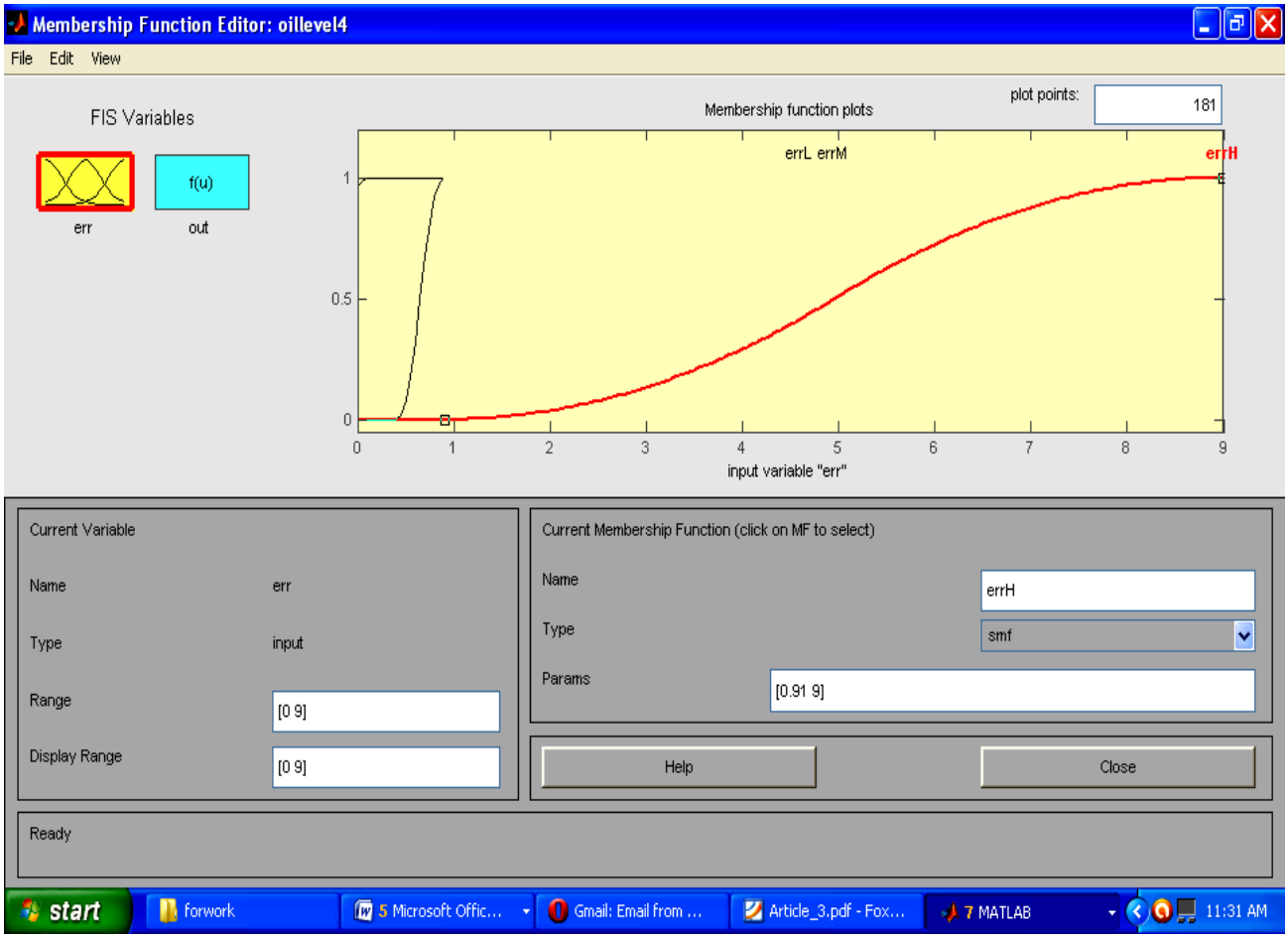

Fig 10: Input Membership Function Plot for High Flow rate case (errH)

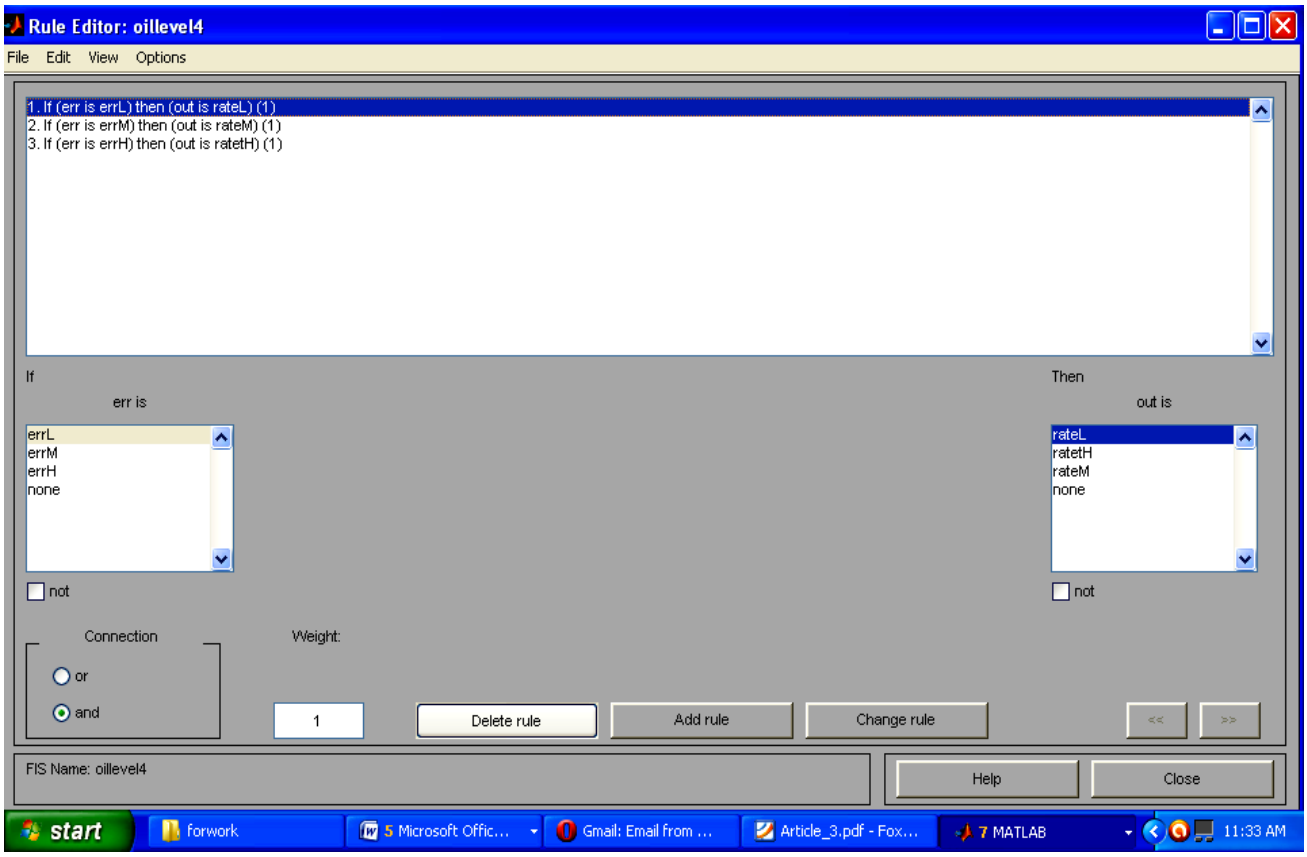

Fig 11: Rules for the FLC

3.5 Integrated Solution (FED Framework)

The model of our proposed framework is as shown in Figure 12. It includes four constant (initializing) blocks and four corresponding slider gain blocks which serves as blocks for user (and of course an adversary) flow rate entries. Thus multiple users can be modelled in this way. Two subsystems have been implemented defining the Encryption/Ethernet Transmission Algorithm (EETA) and Ethernet Receive Algorithm (ERA) respectively. Also included is the Fuzzy Logic Controller (FLC), the oil tank subsystem, display blocks and a To Workspace block for data capture in-system.

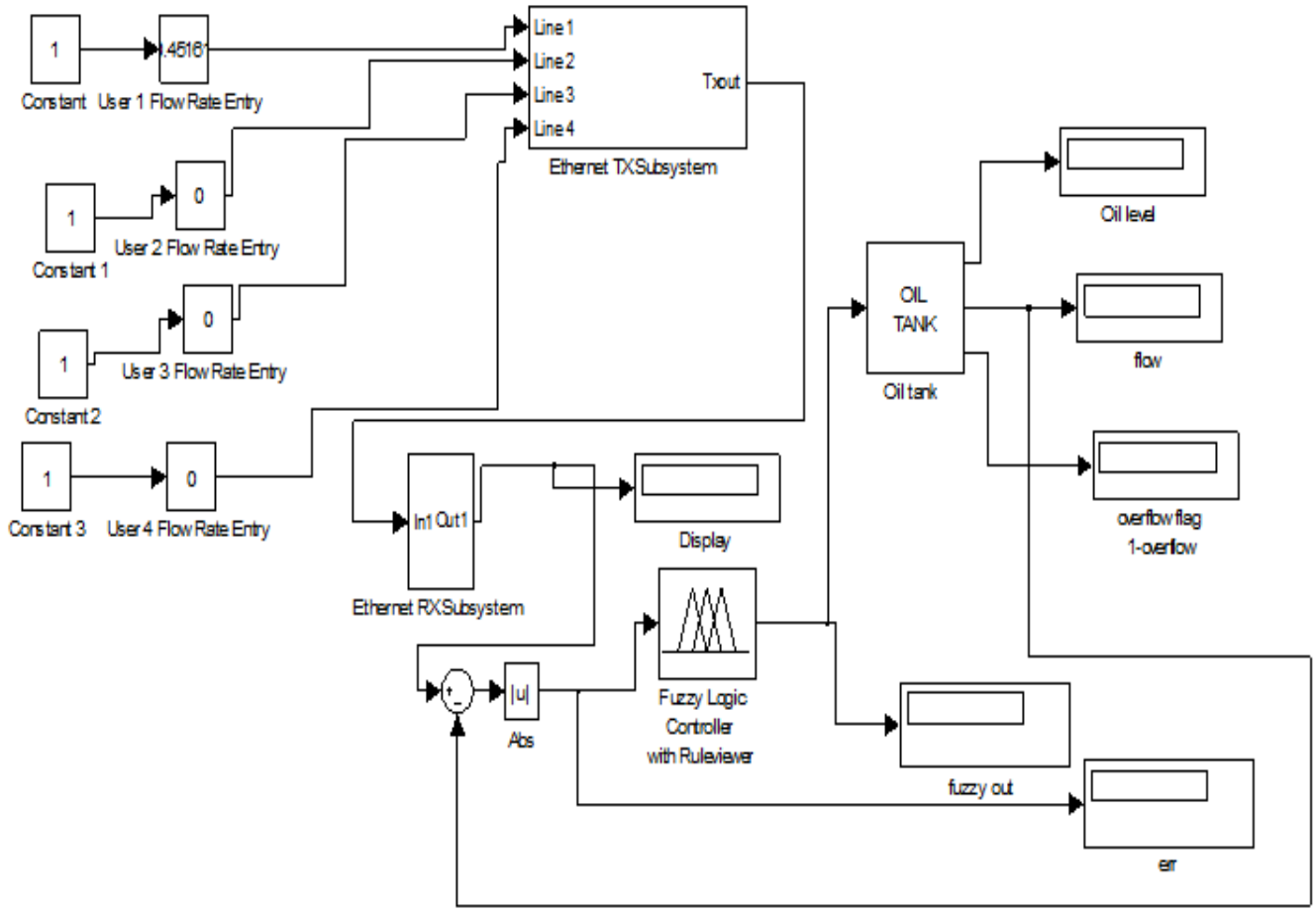

Fig 12: FED system framework for an Oil-Level Tank in Simulink

3.5.1 Encryption/Ethernet Transmission Algorithm (EETA) Subsystem

The EETA subsystem includes 4 Embedded MATLAB function blocks, 4 Math Function blocks and a Mux (Multiplexor) block. Input flow rate signals enter into this system through 4 inports labeled (lines 1 to 4). The Encryption algorithm is hardcoded in an Embedded Matlab block; these blocks are connected in a parallel configuration- so we have parallel computing functionality implemented. The signal chain is distributive. The Math Function block does Matrix transposition on the encrypted signals which in turn passes it to a 4-input channel Mux block which performs signal concatenation on the input signal train. The Mux block then routes the mixed signal to the main subsystem through an outport labeled Txout.

3.5.2 Ethernet Receive Algoritgm (ERA)

The ERA subsystem receives the encrypted signals from EETA subsystem. It includes an Embedded Matlab Function block, Mux/Demux blocks, Decryption Logic subsystems, Sum of Elements blocks. Also added is a data type conversion block, display blocks for visualizing numeric data and inport and outport blocks. Inputs (Encrypted signals) are received into these subsystems through inport 1 (labeled In1). The Decryption algorithm is hardcoded in an Embedded Matlab block; these blocks is fed to a data type conversion block of type 'double', and fed to a 4-channel Mux block. The signal chain is distributive and works in parallel computing topology. 4 Decryption Logic subsystems handle the actual decoding routines and the decoded signal train and summed up for each case using Sum of Elements blocks 1 to 4 respectively. The decoded system is multiplexed and summed up by
single Sum of Elements blocks. This signal is then routed to main subsystem through an outport (labeled Out1).

3.5.3 Simulink Implementation of Oil Tank Model

A simulink model of oil tank is presented. This model consists of the following main features:

i. The oil-tank system itself
ii. Input blocks (Constant/slider gain blocks) for line inputs (flow rate settings)
iii. A Fuzzy-based Controller subsystem to control the height of oil in the tank by varying the flow rate
iv. A reference signal that sets the desired oil level
v. An Encryption/Decryption Subsystem for encryption and decryption of control signals
vi. A Scope block that displays the height of oil (oil level) as a function of time

The oil tank is the process that needs to be protected or

monitored. In order to effectively accomplish this task there is need for a simulation model that best describes the key parameters that define a tank of this nature and that will enable the determination of its flow rate. Among the simulation parameters considered for the tank model include the following:

- Signal flow inputs
- Tank volume – using integration this can be achieved
- Tank Area – as defined in the model
- Height of oil in tank
- Gravitational Constant: 9.8m/s^2 is assumed.

These parameters will be useful in the determination of the flow rate of oil into the tank.

Principle of Operation

The oil flow level can be controlled by using limited integrator in the simulated oil tank subsystem as in the Water tank demo [10] and this is shown in Figure 13:

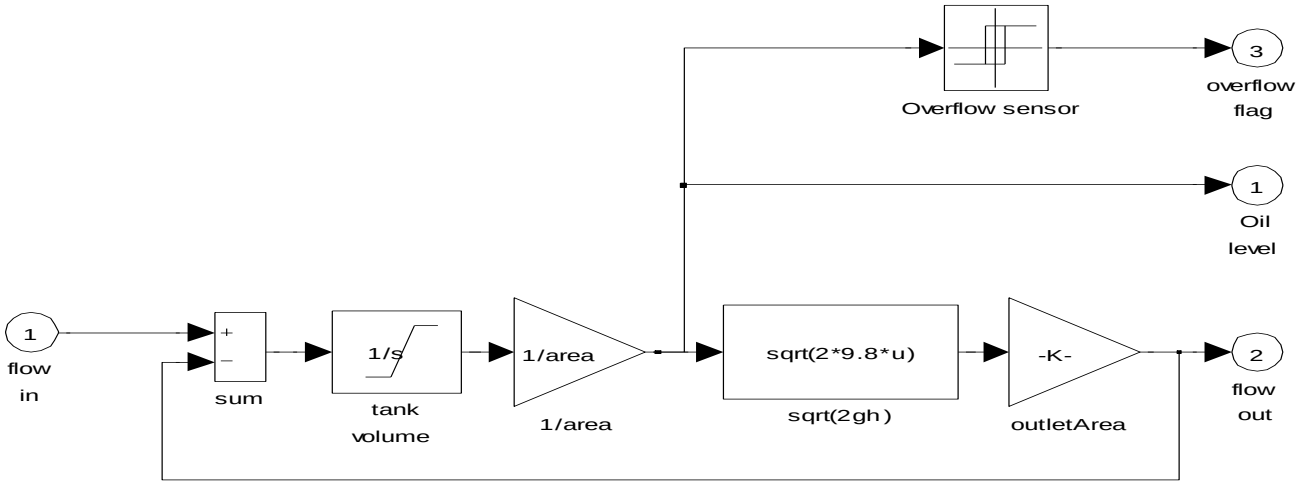

Fig 13: Block diagram model of Oil-tank subsystem in Simulink

Oil flowing through the inport (inport 1 labeled as 'flow in') is summed with the output using a Simulink summation block and fed to an integrator block that models volume of oil in tank. The height of oil in the tank can be obtained by taking the inverse of the area of tank (1/area) using a Gain block and multiplying by the tank volume using a direct feed through technique (See Fig. 13). The output flow rate can thus be obtained using Equation 3 implemented in a matlab function block and is defined by outport 2 labeled as 'flow out'. From Figure 13 an overflow sensor has been incorporated to monitor height of oil and is detected at outport 3 labeled as 'overflow flag'. An additional outport (outport 1) is included so that the height of oil in tank can be read. It is labeled 'Oil level'.

4. RESULTS

Using the model developed in section 3, we present computer simulations for the normal flow rate, below and above normal flow rates response curves against time shown graphically below (Figures 14, 15 and 16 respectively). For the purposes of this analysis attacks are assumed to emanate from user entry 1, who can change flow rate values at will, though this is also possible at any other user entry channels. Figure 17 also show a concatenated plot system for the simulated case.

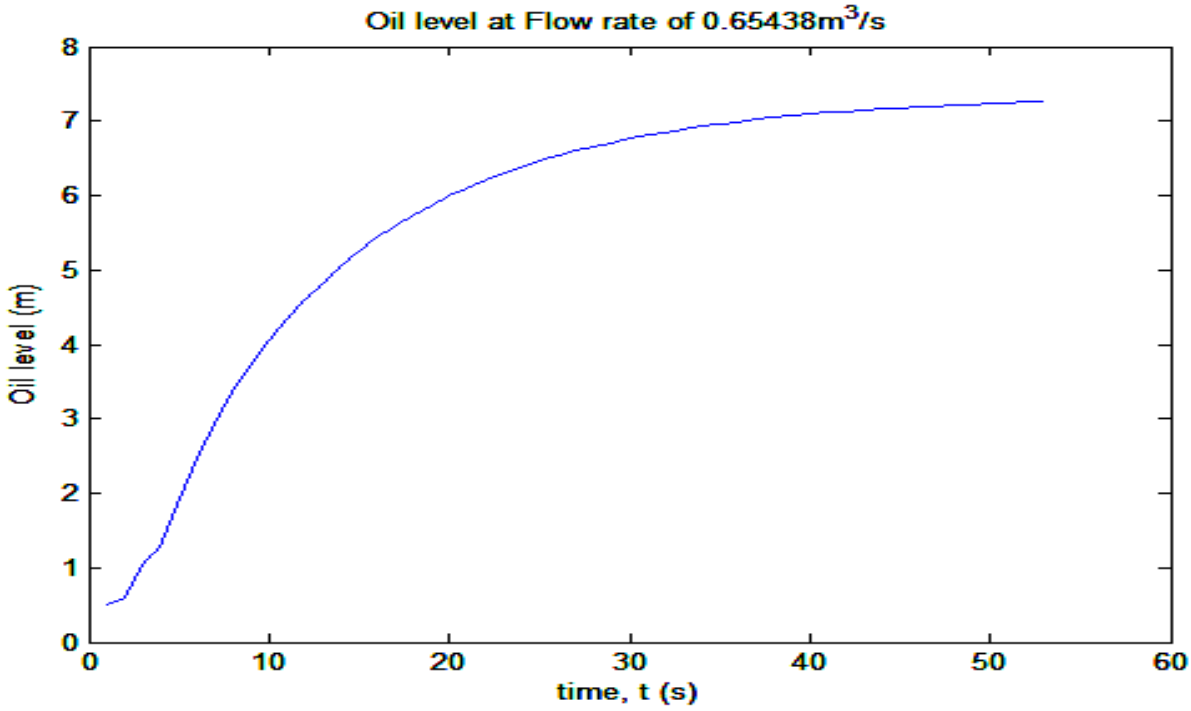

Fig 14: Response curve at flow rate of 0.6543m^3/s (normal flow rate

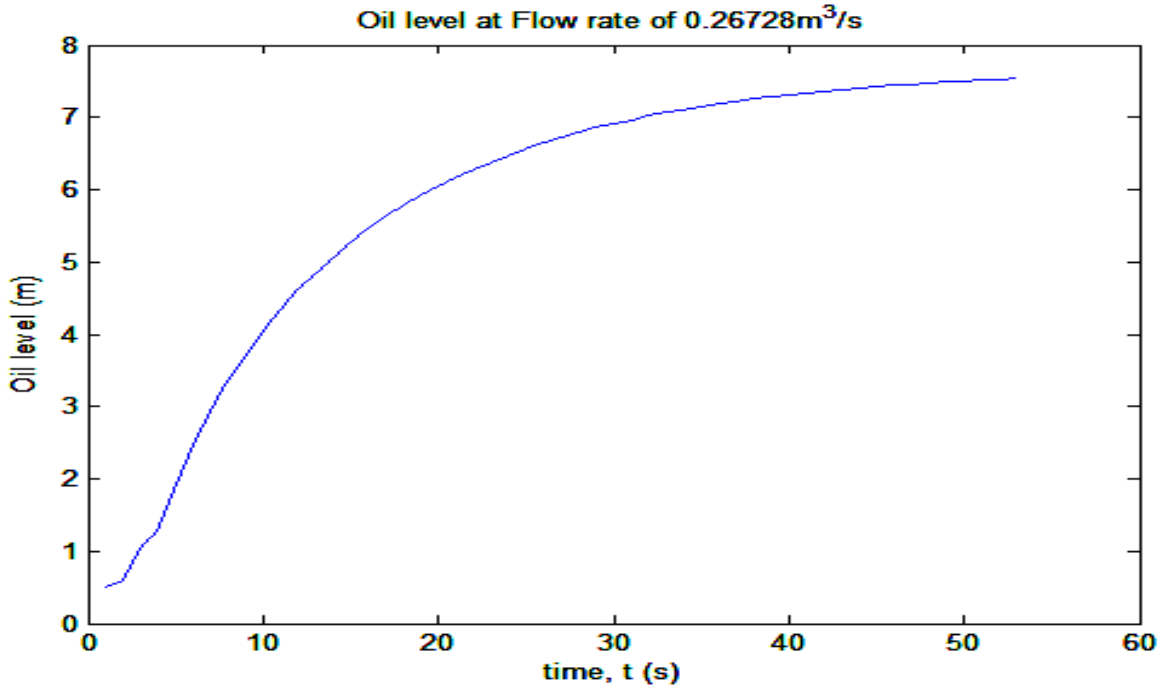

Fig 15: Response curve at Flow rate of 0.26728m^3/s (below normal flow rate)

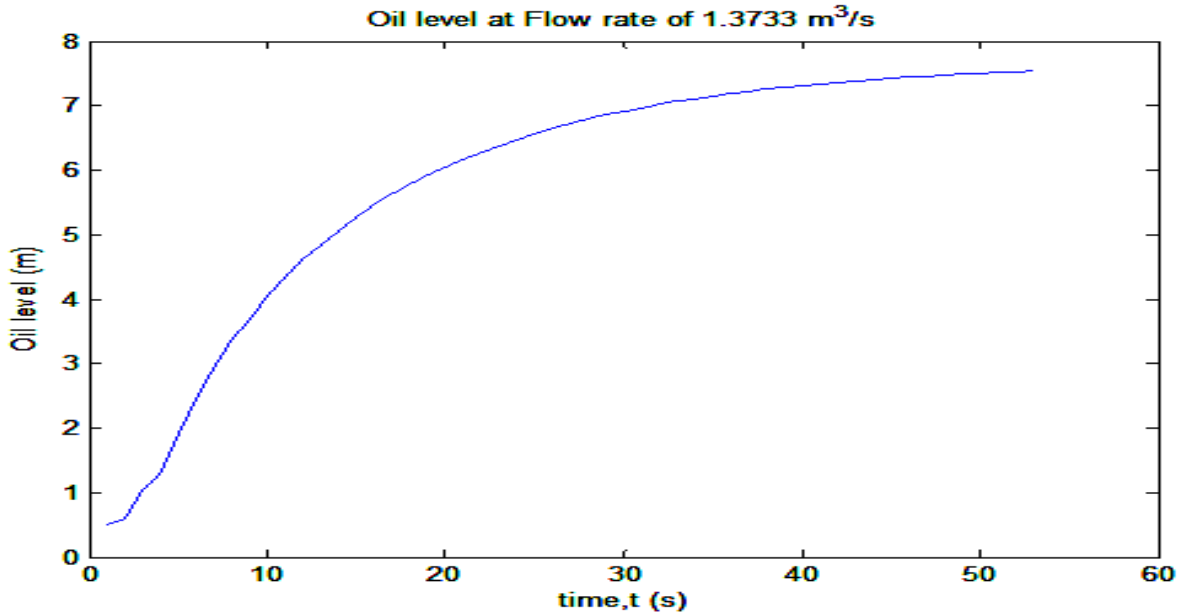

Fig 16: Response curve at Flow rate of 1.3733m^3/s (above normal flow rate)

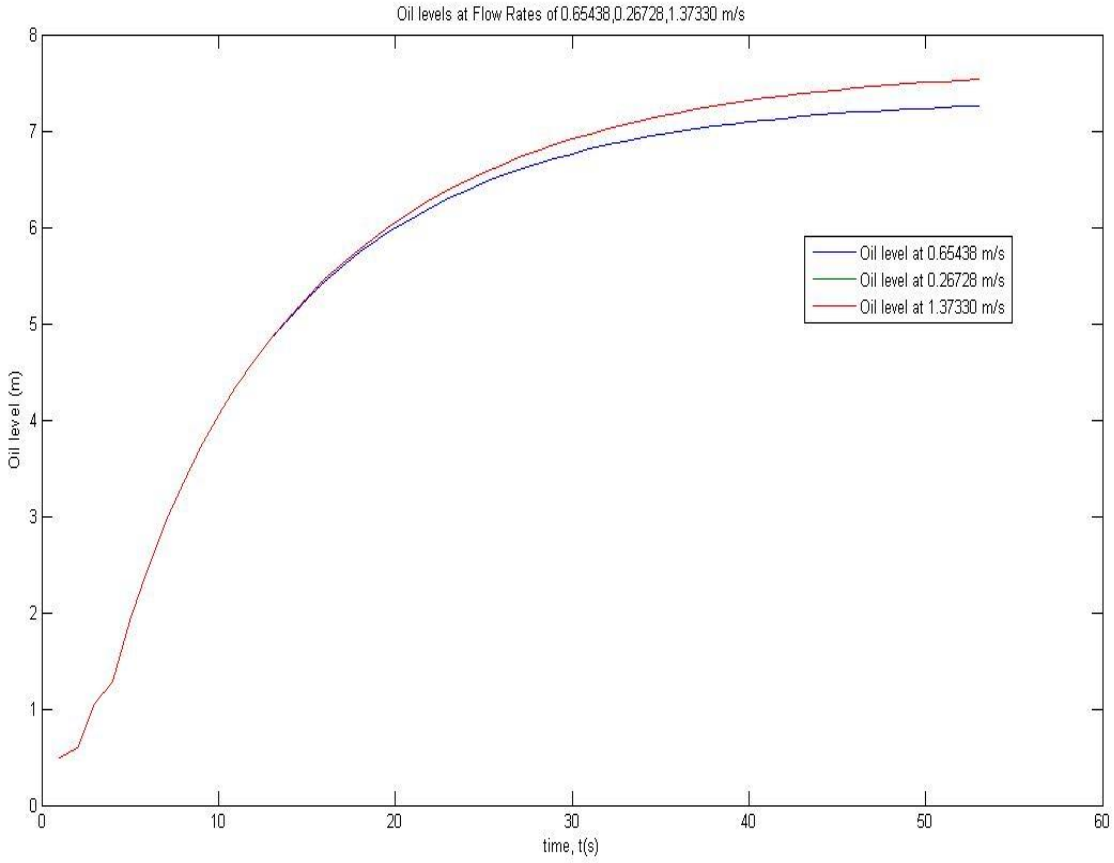

Fig 17: Response curve at Flow rates of 0.65438, 0.26728, and 1.3733m^3/s

5. DISCUSSIONS

As can be seen from the graphs, for a flow rate 0.65438m^3/s the oil level rises steadily from a stationary level of about 0.5m to a peak of about 7.3m which is a tolerable limit. For abnormal flow rates of 0.26728m^3/s and 1.3733m^3/s, the peak oil level is about 7.5m which still falls within tolerable limits. Also, with multiple attacks using abnormal flow rate of 0.26728m^3/s the peak oil level is still about 7.3m. The same is also true for abnormal flow rate of 1.3733m^3/s. This is evident in the combined systems plot of Figure 17. Thus, the system maintains reasonable stability even with large variations in flow rate due to multiple attacks.

6. CONCLUSIONS

The results of the proposed secure model for oil-level tank control clearly show the advantage of using a fuzzy based controller in the designed encryption/decryption system. Based on the simulations, the effect of abnormal attacks gave negligible effect on the oil level. Unlike some encryption/decryption logic with hundreds, or even thousands, of layers running on dedicated computer systems, a unique fuzzy based encryption/decryption system with dynamic and mutable capabilities while yet using a small number of rules and straightforward implementation is proposed to solve a class of level control problems with unknown dynamics or variable time delays commonly found in industry.

7. RECOMMENDATIONS FOR FUTURE WORK

The fuzzy based encryption/decryption system can be easily programmed into many currently available industrial process controllers. The FLC encryption/decryption system simulated on a level control problem with promising results can be applied to an entirely different industrial level controlling apparatus.

As a software framework, Fuzzy Logic provides a completely different, unorthodox way to approach a control security problem. This method focuses on what the system should do rather than trying to understand how it works. One can concentrate on solving the problem rather trying to model the system mathematically, if that is even possible. This almost invariably leads to quicker, cheaper solutions. The Fuzzy Logic Controller approach is recommended where a secure and accurate control of the liquid level in any industrial application is required. An open-source version of the framework together with the algorithm source is planned.

8. ACKNOWLEDGMENTS

Many thanks to the SurePay Foundations Group for providing programming support..

9. REFERENCES

[1] R. L. Krutz, 2006. “Securing SCADA Systems,” Wiley Publishing Inc., Indianapolis.

[2] J. Pollet, 2002. “Developing a Solid SCADA Security Strategy,” Proceedings of the Second ISA/IEEE Sensors for Industry Conference, Houston, USA, pp. 148-156.

[3] J. Caswell, 2011. Survey of Industrial Control Systems Securiy, http://www.cse.wustl.edu/~jain/

[4] Alvaro A. Cárdenas, Saurabh Amin, Shankar Sastry. Secure Control: Towards Survivable Cyber-Physical Systems, the 28th International Conference on Distributed Computing Systems and Workshops University of California, Berkeley.

[5] Parfomak, John Moteff and Paul (2004). Critical Infrastructure and Key Assets: Definition and identification. CRS Report for Congress, USA, CRS-4.

[6] John Moteff, Claudia Copeland, and John Fischer (2003). Critical Infrastructures: What Makes an Infrastructure Critical? : CRS Report for Congress (EEUU), Resources, Science, and Industry Division.

[7] R.Dawson, 2008. Secure Communications for Critical Infrastructure Control Systems. Information Security Institute, Faculty of Information Technology, Queensland University of Technology, Brisbane, Australia.

[8] D.P. Huertas, 2011. Cyber-Security and Safety Analysis of Interconnected Water Tank Control Systems. KTH, School of Electrical Engineering (EES), Automatic Control Electrical Engineering, Stockholm, Sweden.

[9] Z. Zhi, H. Lisheng, (2011). "Performance assessment for the water level control system in steam generator of the nuclear power plant", “IEEE/CCC”. pp. 5842-5847.

[10] Matlab version R2007b, 2007. Mathworks Inc., USA, http://mathworks.com.

[11] M.Bishop (2004).Introduction to Computer Security: Addison Wesley Publishers.