Estimation of Liquid Level in a Harsh Environment Using Chaotic Observer

Vighnesh shenoy [1], Santhosh KV [2*]
[1,2] Dept. of Instrumentation & Control Engg, Manipal Institute of Technology,
Manipal Academy of Higher Education, Manipal 576104, India
Email: [1] vighneshshenoy112@gmail.com, [2] kv.santhu@gmail.com
*Corresponding Author

***Abstract*—The increased demand for liquid level measurement has been a key factor in designing accurate and reliable control systems. Here, a study was carried out to calculate the liquid level in a tank using a pressure sensor for changes in inlet liquid parameters like temperature, density and velocity. Prediction of their variables for the long term is essential due to the randomness present in the input and measurement. Hence, observer design for state estimation of a non-linear dynamic system with uncertainties in the measurement and process becomes important. This work provides a feedback observer solution for a system with multiple inputs and single measurable output. A full state observer model is developed to estimate a system's states with a sensor placed at a definite position from the pipe's input point through which the liquid flows at different densities and temperatures. Using the observability properties, Luenberger full state observer is designed by various methods, verified using MATLAB and SIMULINK for the system state estimation. To incorporate process noise and measurement noise, the Kalman estimator is integrated with the system. Chaotic systems are susceptible to initial conditions, variations in parameters and are complex dynamic systems. However, providing consistently precise measurements through particular meters necessitates time-consuming computations that can be reduced by employing machine learning approaches that make use of optimizers. The results obtained are compared with the prediction models obtained using Artificial Neural Networks and are validated through the readings obtained from the experimental setup.**

***Keywords*—Artificial Neural Network; Kalman filter; Liquid level; Observer; Orifice; Sensor; State estimation.**

I. Introduction

In many commercial or industrial applications as well as other liquid level monitoring industries like biochemical processing, chemical industry production, and aircraft fueling systems, liquid level sensing is a crucial necessity [1]. In the past decade, numerous level sensing techniques have been proposed. Many different forms of technologies are used to measure liquid level. Below mentioned are the most typical sensor kinds used in the market today. Sensors that can be installed above the liquid, such as ultrasonic [2] sensors, that use the travel time of an electromagnetic wave or sound to calculate the distance from the liquid surface. Some sensors, such as floats that move up and down a liquid-filled tube linked at the bottom of the tank or a pressure sensor inserted in a hole at the bottom of the tank that gauges the pressure created by gravity acting on the liquid, can be mounted externally to liquid-filled tanks.

Another approach is to lower a probe into the tank and submerge it in the liquid. A capacitive probe [3], for example, can be used to detect changes in capacitance charge as the length of the probe is exposed to various liquid concentrations or optical sensors [4], which estimates the liquid level based on its modulation of the intensity, spectrum, phase, or polarization of light passing through an optical fibre.

Finally, the approach based on image processing [5] offers the benefits of great automation and efficiency. The fast development of machine vision to estimate liquid level based on the image has increased interest in detection approaches based on image processing. The currently known methods for measuring liquid levels are either expensive to use or offer a small measurement range. In the worst case, their size, environmental conditions and ongoing maintenance make them difficult to install. A liquid level sensor are also affected by variations in liquid temperature [6], density [7], composition, viscosity, etc. One could compensate for the sensor's sensitivity to these variations but doing so would make the process more complex and time consuming. An ideal liquid level sensor should have good stability, high resolution, and a low-cost device.

In scenarios when hardware sensors are unavailable or incompatible, virtual sensors can be used for estimating conclusions from system observations. Soft sensors [8] also known as virtual sensors are the plant models dedicated to estimate the plant variables in situations where the measurement of the actual variables from the hardware devices is a complicated task. For level estimation, this report uses a soft sensing approach. A sensor that is not directly in contact with the process liquid is always better from a maintenance perspective. Pressure sensors are one of the best alternatives [9] [10] [11].

There are several types of pressure sensors based on their fabrication capacitive pressure sensor [12], inductive pressure sensor [13], piezoresistive pressure [14], etc. Pressure sensors can operate reliably in challenging environmental conditions

[15]. However, calibration must be performed regularly to achieve improved accuracy. Orifice plates are the most common flow sensors, and they work by measuring the pressure drop across the plate to detect the flow of fluid passing through it [16].Here orifice sensor is used to calculate liquid level [17]. The velocity profile and pressure drop across the orifice were investigated using CFD simulation [18] of flow through the orifice plate for specific operating conditions and geometric configurations. The liquid level is computed for different values of temperature and density, for the plant model obtained using the system identification technique. This report explains how to estimate a liquid level sensor using a pressure sensor [19] by just using one vital piece of information about pressure flow to convert sensor output to a liquid height. Here [20], a differential pressure flow meter-based solution to the modeling problem of a contraction process for a liquid flow is reported. The coefficient of contraction is calculated analytically. This research investigates the relationship between the square-shaped orifice plate diameter and the coefficient of flow contraction.

There are few short comings which needs to be addressed. Process variables such variations in liquid density, temperature, inlet velocity, and other factors might have influence on a flow sensor. A few published papers have examined error offsetting, but most of them haven't taken into account change in any of these process factors for analysis. According to reports, the design has either been altered or some of the parameters have been compromised due to the cost of certain sensors. Alternate sensors has been reported in few cases with a few characteristic sacrificed.

In any sensor based system, observers can be used as a replacement for a sensors. Observers are algorithms that generate observed signals by combining sensed signals with additional control system information. Compared to sensed signals from the sensors, these observed signals are more precise, simpler to produce and more dependable. Instead of adding additional sensors or improving existing ones, observers provide designers an attractive option. According to the observer principle, it is possible to predict plant behavior more accurately than by utilizing just the feedback signal by combining this measured feedback signal with knowledge of the system (mainly the plant). The observer can occasionally be utilized to improve system performance. It may be more accurate than sensors or lessen the phase lag a sensor would naturally have. In order to increase disturbance response, observers can also offer observed disturbance signals. In other situations, observers can lower system costs by improving a low-cost sensor's performance so that the two combined can deliver performance comparable to a more expensive sensor. In the most extreme instance, observers can omit a sensor entirely, lowering sensor costs and related wiring [76].

There are different types of observers: reduced-order observers [21] [22], linear functional observers, dual observers, full-order observers [23] etc. An observer can also be designed using a pole placement technique [24]. The design of observers is important in getting system state variables. It is possible to create an observer for a discrete-time system or a continuous-time system [25]. The continuous-time observer design is emphasized in this work. The primary goal is to estimate the state using the input parameters and output measurement. The full state observer designed in the present work also considers measurement noise and unmeasured disturbances in the system. This system can generate estimates of state variables which can or cannot be measured. There are 2 ways of modeling a dynamic system,

1) First principle model
2) Data-driven model

In case the physics and maths behind a problem/system are known, using this knowledge we can build a First principle model whereas no system details are known but only measured data are available in such cases using this data, data-driven [27] [26] models can be constructed. In this report, we have the data from the CFD modeling which is then imported to the MATLAB system identification toolbox for system identification i.e a gray box modeling.

In dynamic systems [28], the response signal depends on both the system's input signal and past behavior. System identification uses certain observed data to estimate the parameters for a defined model structure. The model's accuracy depends on the experimental design and quality of the measured data [29]. Time-domain data contains input and output variables of the system observed over a definite time interval. The model structure relates the output and input variables mathematically in the form of transfer functions. Based on this model structure, a state-space representation of the corresponding system can be obtained. Kalman filter [30] is an optimal estimation algorithm applicable in guidance and navigation systems, computer vision systems, signal processing, etc. When the variable of interest can only be monitored indirectly or when measurements are available from several sensors but may contain noise, Kalman filters are utilized.

Chaotic systems are complex dynamic systems which are extremely sensitive to initial conditions and variations in parameters. So, the prediction of their variables for the long term is important due to the randomness present in the input and measurement [31]. Using unmeasured state dynamics, a novel approach for designing an adaptive state observer for a class of nonlinear systems with unknown parameters was presented in [32]. These were not descriptor systems, though. Similarly, only actuator failures were taken into account when an adaptive observer for a class of Lipschitz nonlinear descriptor systems was proposed in [33]. Although it is common for actuator and sensor defects to occur at the same time in real systems, only one fault type has been considered in many previous research. For example, in [34] [35], only actuator failures were evaluated, whereas in [36] [37], only sensor faults were examined. Hence, observer design for state estimation for the nonlinear dynamic system with uncertainties in the measurement and process becomes very important.

The response from the mathematical model obtained using observer design can be compared with the response's Neural Network (NN) prediction algorithms. Regression models make a useful prediction for solving real-world problems in scientific, industrial, and business environments. Regression analysis helps in understanding the relationship between a dependent variable and an independent variable. Neural networks and deep learning algorithms help in solving complex mathematical problems [38]. Artificial Neural Network (ANN) comprises neurons [39], which can analyze complex problems and help in decision-making by providing an accurate solution.

In [40], mathematical techniques based on Machine Learning (ML) are used to calculate the thruster orifice size. The ML techniques were put into practice using MATLAB. Several ML models are used to estimate the size of the orifice utilized in rocket engine thrusters. This study models the network based on input parameters using 55 data sets.

Following a comparison of the outcomes produced by various models, it was concluded that the LSSVM regression model with Gaussian add kernel outperforms the other models when predicting orifice size given a known flow rate and pressure drop. Ref [41] analyzes machine learning optimizers to anticipate oil flow through orifice plates and avoid problematic discharge coefficient calculations. Based on several input variables, it estimates oil flow through orifice plate meters. To compute oil flow rates (Q_0) through orifice plate meters, a dataset of 6292 data records with 7 input variables including liquid temperature, upstream and differential pressure, β ratio etc was analyzed. Combining different algorithms with a swarm-type optimizer, the two-stage ABC-DWKNN Plus MLP-FF model provided the greatest predicted accuracy for the oil flow rate via the orifice plates with RMS errors of 8.70 stock-tank barrels of oil per day.

In [42], a special deep machine learning model called convolutional neural network (CNN) was built to predict oil flow rate via an orifice plate using the same seven input variables as in [41]. The prediction performance of three alternative machine learning techniques, including the support vector machine (SVM), least absolute shrinkage and selection operator (LASSO), and radial basis function (RBF), was also constructed and compared to that of the CNN model. This was done to demonstrate the proposed CNN model's superior performance versus traditional machine learning models. The result demonstrates that the CNN model, with a Root Mean Squared Error (RMSE) of $0.0341 m^3/s$ and a coefficient of determination of 0.9999, had the highest Q_0 prediction accuracy.

Ref. [44] proposed an exponential polynomial observer, which stabilizes the resulting state estimation error for a nonlinear system. They compare exponential polynomial high gain and extended Luenberger observer. The simulated result of the mentioned approach shows robustness against external noise. Ref [45] reported an approach dealing with a chaotic system's synchronization problem, considering parameter identification simultaneously. The observer here uses an extended Kalman filter and it shows how an observer is used for estimating or synchronizing with different chaotic systems.

Ref. [46] developed a statistically sound and robust method of evaluating multiple input single output systems performance. This method involves the transfer function modeling and fuzzy logic approach. This work considers only two inputs and a single output process, there is a scope for increasing the interaction inputs. In [47], 1-step-ahead prediction models of river water levels are created using an artificial neural network (ANN) model design framework called the Multi-Objective Genetic Algorithm (MOGA). Given the available data, the design technique is a nearly automated method that may divide it into datasets and find a nearly ideal model with the proper topology and inputs, providing superior efficiency on a new dataset, or data not utilized for model creation.

Ref. [48] reported a non-linear gradient-based observer for the synchronization of a chaotic system. This model is based on the Range-Kutta model, where state parameters are derived based on error square minimization. The observer and controller synchronizations are done using the Lyapunov stability approach. Range-Kutta Gradient Observer (RKGO) and Sliding Mode Observer (SMO) are compared using numerical simulation. Ref. [49] reported a full state observer model developed for estimating the states of a system with varying input parameters. The system consists of a pipe fitted with an orifice sensor having liquid flow through the pipe inlet with varying inlet velocity having different liquid densities and temperatures. Luenberger full state observer is designed and verified. To incorporate process noise and measurement noise, the Kalman estimator is integrated with the system. It is observed that error dynamics reduce faster over time.

An accurate state estimate is vital in today's data-intensive control, computing, and power system applications according to [50], since the performance and stability of the system as a whole can be greatly impacted by missing sensor data. In order to provide a more reliable and resilient state estimation method, this article reports a novel second-order fault-tolerant extended Kalman filter estimate framework for discrete-time stochastic nonlinear systems with sensor failures, constrained observer-gain perturbation, external noise and disturbances. The suggested approach may be a good replacement for the current nonlinear estimating methods, according to experimental findings.

Ref. [51] describes a closed-loop Multiple-Input Single-Output (MISO) method for detecting abnormalities during anesthetic induction in clinical trials. It is demonstrated that using more complicated closed-loop identification techniques doesn't increase model accuracy. By employing this method, it can identify and validate a set of models that accurately captures the blood pressure response to propofol infusion as well as the depth-of-hypnosis response for individuals who are at risk of cardiovascular suppression.

The observability analysis and observer design for a nonlinear three-tank system are discussed in [52]. To ensure convergence of the state estimation, a High-Gain Observer (HGO) is created utilizing the equivalence of the initial state-space realization with its observability canonical form. When comparing the HGO response to a Luenberger observer and the Extended Kalman Filter, taking into account non-linearity, interaction, disturbances, and noise, the performance was validated through simulation and experiments in a multipurpose plant installed with real sensors. Despite the HGO's sensitivity to noisy variables in processes like liquid level, theoretical and practical studies demonstrate that the HGO can provide a robust estimate and disturbance rejection.

The presently reported work is based on the influence of the placement of the sensor on the measurements. The adaptation of sensors in terms of placement has not been discussed. Literature reports that the flow sensor's output gets affected by process parameters like the variation in liquid density, temperature, and pressure of the liquid. Though a few reported works are available, which have been discussed to offset the errors, none have discussed making sensing robust.

In this paper, a full-order observer for a Multiple Input Single Output system(MISO) is designed. The designed observer helps calculate the liquid level for a system having a sensor placed at a fixed place from the pipe's input point through which the liquid flows at different densities and temperatures. Luenberger estimator [53] can be used in a deterministic system, whereas if the system characteristics are stochastic, Kalman filters are used. The overall structure achieves stability disturbance rejection. The results are compared with the predictions made by Artificial Neural Networks.

II. Description of System and Calculation of the Liquid Level

Fluid Dynamics is an interesting and complex topic of study and research. Almost everywhere in the actual world, there is fluid. As a result, we require fluid dynamics to describe or simulate these fluid flows. To measure fluid flow through numerical analysis, Computational Fluid Dynamics (CFD) may be used to provide precise insight into the fluid flow characteristics through orifice plates [54]. A CFD simulation for a three-dimensional model [1] is carried out using ANSYS to investigate the laminar type of liquid flow and the pressure profile through the sharp-edge orifice plate with a pipe diameter (D) of $1inch$ and the orifice diameter (d_0) of $0.5inch$. β ratio $= 0.5inch$ and length/depth of the orifice plate is $0.05inch$ [55]. The pipe's total length is kept at $40inch$ having both upstream and downstream flow lengths as $19.975inch.$ The distance between points P_1 and P_2 (Fig. 1) from the point D (Fig. 4) is $0.5334m$ and $0.4953m$ i.e. $20.5inch$ and $19inch.$ At the upstream and downstream of an orifice, appropriate lengths are provided to allow flow to develop fully throughout the pipe. Vena-contracta taps are used in this work. The pressure slightly rises as the fluid moves in the direction of the orifice plate. At the orifice point, pressure will be at its peak before abruptly decreasing when the orifice is passed. This results in a rise in flow velocity and a corresponding fall in pressure. We are taking a pressure tap near the Vena contracta, which is where the minimal flow area—also known as the Vena contracta—occurs, to measure the highest pressure decrease. Static pressure is determined both at the upstream point (P_1) and downstream point (P_2) of the pipe in order to calculate the pressure difference. The flow continues to contract a short distance downstream of the hole called the vena-contracta point. Accordingly, pressure tap P_2 is installed at vena-contracta.

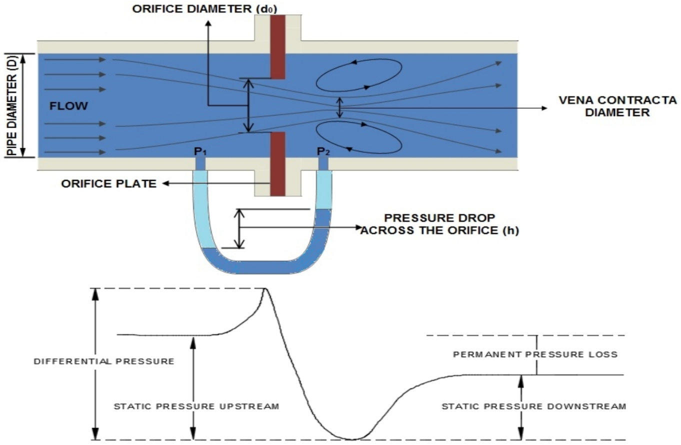

Fig. 1. Model of orifice meter flow lines and their pressure characteristics

Equation (1) is used to calculate the liquid flow rate to the tank,

$$Q = C_d \frac{A_0\sqrt{2}}{\sqrt{1-\beta^4}}\sqrt{\frac{(P1-P2)}{\rho}} \tag{1}$$

where, $'C_d'$ is the coefficient of discharge, $'A_0'$ is the cross-sectional area of an orifice, $'P_1 - P_2'$ i.e., ΔP is the static differential pressure at a distance $'D'$ and $'0.5D'$ respectively, $'\rho'$ is the density of the liquid.

It was reported in [56] that the discharge coefficient declined as the orifice's flow rate increased, and it also did so for the same pressure drop when the orifice's diameter increased. Study on orifice discharge coefficient is done by [57] [58] [59],suggests that the discharge coefficient C_d is a product of three coefficients (the viscosity coefficient(C_v), the contraction coefficient(C_c), and the velocity profile coefficient (C_p) such that $C_d = C_c C_v C_p$.

The C_d typically settles at or around 0.8 for most orifices [60]. Here, the same number is used in Equation (1) to compute the flow rate. As seen in Fig. 2, the fluid flows through the pipe to the $24cm$ long by $30cm$ wide rectangular tank. Calculating the liquid level while maintaining a constant outflow from the tank $Q^{'} = 0.00035m^3/sec$ is done by changing the input velocity. Inlet velocity is increased at a rate of $0.03m/s$ every $100seconds$ (step-size). With steps of $0.03m/s$ per $100seconds$, the input velocity ranges from $0.03m/s$ to $0.6m/s$. As a result, 100 simulations are run for each inlet flow valve, producing 100 ΔP values. Hence, accumulation can be calculated using Equation (2).

$$Q - Q^{'} = Accumulation \tag{2}$$

$$Tank\,Volume = Length(L) \times Width(W) \times Height(H) \tag{3}$$

From Equation (2) and Equation (3),

$$Q - Q^{'} = Length(L) \times Width(W) \times Height(H)$$

$$Height = \frac{Q - Q^{'}}{Length(L) \times Width(W)}$$

$$Height = \frac{Q - Q^{'}}{30cm \times 24cm} \tag{4}$$

Substituting values of Q and $Q^{'}$ in Equation (4) we get the theoretical liquid level in the tank.

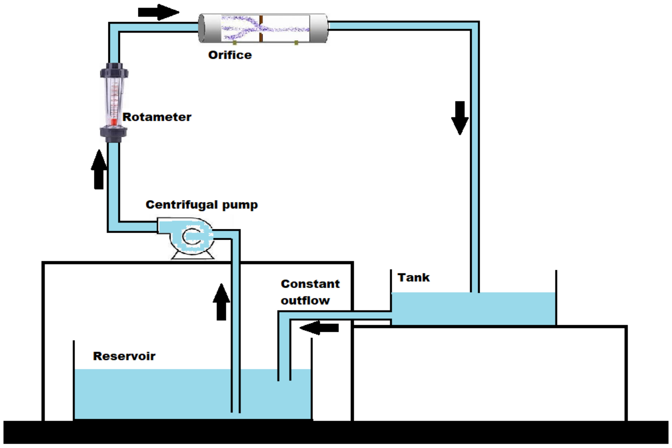

Fig. 2. Experimental setup

A. Setting Up CFD Simulation

The computational mesh created for an orifice pipe needs to be set up for a CFD analysis using ANSYS Fluent. A transient pressure-based solver with SIMPLE pressure velocity is been used here. Material for CFD simulation is set up as water with a default density of $998.2kg/m^3$. A User-Defined Function(UDF), is a code that the ANSYS FLUENT solver may dynamically load to enhance the code's built-in functionality. Here, UDF is defined for a change in inlet velocity via ANSYS. The inlet velocity increases from $0.03m/s$ to $0.6m/s$ in $0.3m/s$ steps every $100secs$. Inlet and outlet boundary conditions (Fig. 3) are set up by selecting inlet velocity magnitude from a previously created UDF file. Since the level is estimated using a pressure sensor, Point P_1 and P_2 are created across the upstream and downstream to calculate the pressure drop. The pipe is modeled with the no-slip condition as a solid wall. The gauge pressure was set as 600000 pascals at the pipe's exit. The output report file is then imported to an excel sheet for system identification purposes. Fig. 1 and Fig: 4, Fig. 5 demonstrate the model and the geometry accordingly. To measure the relative pressure difference between the upstream and downstream, static pressure is assigned at the outlet of the pipe.

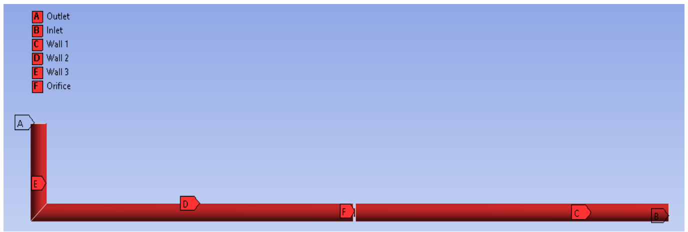

Fig. 3. Boundary conditions for CFD simulation

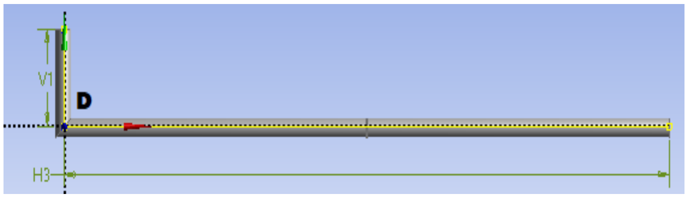

Fig. 4. Geometry of computational domain for the CFD analysis

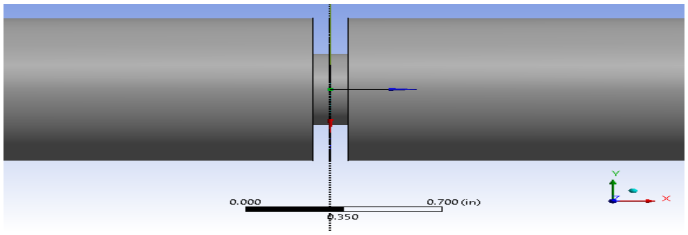

Fig. 5. Orifice geometry

The density of the mesh was increased to get more precise findings. Although the mesh generator comes with a default mesh density, several density trials were conducted to assess the sensitivity in terms of convergence and result accuracy. Here re-meshing was carried out for the same model as shown in Fig. 6 by just increasing the number of elements (finer meshes) in the mesh keeping the same settings in both cases. The result (Delta P) was almost similar to the previous meshed model. This means that the simulation carried out is grid-independent. The meshing gave a total of 15495 nodes and had 69874 elements that consisted of unstructured triangular (tetrahedral) mesh. Contours of static pressure magnitude along the pipe are shown in Fig. 7.

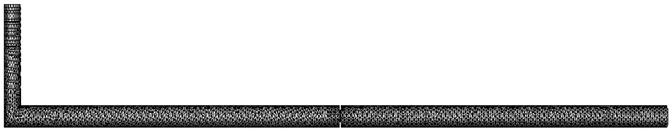

Fig. 6. The Computational mesh for the orifice geometry in ANSYS meshing application

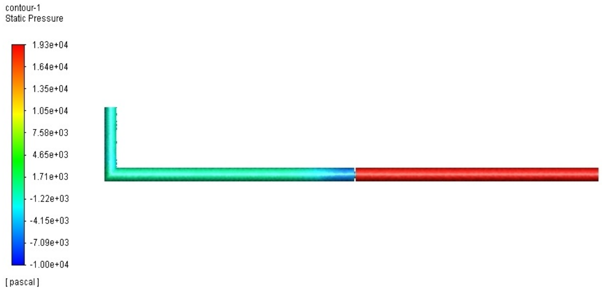

Fig. 7. Contour of Pressure distribution along the pipe

The model consists of a level tank with a constant outflow, and the tank's inlet is supplied by a pipe that is equipped with an orifice-type flow metre that measures the tank's inflow rate. The inflow and outflow values are used to calculate the liquid level in the tank. This model extracts simulated readings for changes in input to the liquid level for various liquid profiles. The data is then used to find the model of liquid level relating to inflow, the temperature of the liquid, type of liquid (density of liquid).

B. Methodology

The MISO system is designed using the calculated liquid level values for various inlet velocities, temperatures and densities for a specific sensor position. System identification is a technique for designing mathematical models of dynamic systems by utilizing the system's input and output signals. Using time-domain signals, the transfer function model of the overall system is built. This iterative process uses an estimation algorithm to estimate the values of the selected model. The generated model can be justified by fitness parameters such as fit to estimation data and final prediction error, which are obtained as 92.17% and 2.64598×10^{-5} correspondingly. In general, the overall system can be represented mathematically as given in the Equation (5)

$$\begin{aligned} y(s) = (u_1(s) \times G_1(s)) + (u_2(s) \times G_2(s)) \\ +(u_3(s) \times G_3(s)) \end{aligned} \tag{5}$$

$$\begin{aligned} y = \frac{3.931}{(s+10.41)}u_1 + \frac{(-2.814 \times 10^{-7})}{(s+0.03977)}u_2 \\ + \frac{(-1.406 \times 10^{-7})}{(s+0.02577)}u_3 \end{aligned} \tag{6}$$

From Equation (6), the differential equation is obtained as,

$$\begin{aligned} \dddot{y} = 3.931\ddot{u_1} + 0.2576\dot{u_1} + (4.0279 \times 10^{-3}u_1) \\ -(2.814 \times 10^{-7}\ddot{u_2}) - (2.936 \times 10^{-6}\dot{u_2}) \\ -(7.548 \times 10^{-8}u_2) - (1.406 \times 10^{-7}\dddot{u_3}) \\ -(1.469 \times 10^{-6}\dot{u_3}) - (5.821 \times 10^{-8}u_3) \\ -10.4757\ddot{y} - 0.6833\dot{y} - 0.011 \end{aligned} \tag{7}$$

Since $y(t)$ represents the liquid's height in the tank, the differential Equation (7) can calculate the same. The Range-Kutta method of order 4 has been used to find the solution of this differential equation.

Equation (5) can be represented as,

$$\dot{y}(s) = \dot{y_1}(s) + \dot{y_2}(s) + \dot{y_3}(s) \tag{8}$$

$$\dot{y_1}(s) = \frac{3.931}{(s+10.41)}u_1(s) \tag{9}$$

$$\dot{y_1} = 3.931u_1 - 10.41y_1 \tag{10}$$

1st iteration:
Taking '0' initial condition and step size of '0.01':
$u_1(0) = 0, y_1(0) = 0, h = 0.01$

$$\begin{aligned} K_1 = f(u_1(0),\ y_1(0)) = f(0,0) = 0 \\ K_2 = f(u_1(0) + \frac{h}{2}, y_1(0) + \frac{k_1}{2}) = f(5 \times 10^{-3}, 0) \\ = 0.01966 \end{aligned} \tag{11}$$

$$\begin{aligned} K_3 = f(u_1(0) + \frac{h}{2}, y_1(0) + \frac{k_2}{2}) = 0.0186 \\ K_4 = f(u_1(0) + h, y_1(0) + k_3) = 0.0374 \end{aligned}$$

$$\begin{aligned} y_1(1) = y_1(0) + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4) \\ = 1.8991 \times 10^{-4} \end{aligned} \tag{12}$$

$$\begin{aligned} \dot{y_1}(1) = f(u_1(1), y_1(1)) = f(0.01, 1.8991 \times 10^{-4}) \\ = 0.03733 \end{aligned} \tag{13}$$

2nd iteration:
$u_1(1) = 0.01,\ y_1(1) = 1.8991 \times 10^{-4},\ h = 0.01$

$$\begin{aligned} K_1 &= f(u_1(1), y_1(1)) = 0.03733 \\ K_2 &= f(u_1(1) + \frac{h}{2}, y_1(1) + \frac{k_1}{2}) = 0.0550 \\ K_3 &= f(u_1(1) + \frac{h}{2}, y_1(1) + \frac{k_2}{2}) = 0.0541 \\ K_4 &= f(u_1(1) + h, y_1(1) + k_3) = 0.0710 \end{aligned} \tag{14}$$

$$\begin{aligned} y_1(2) &= y_1(1) + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4) \\ &= 0.7344 \times 10^{-3} \end{aligned} \tag{15}$$

$$\begin{aligned} \dot{y_1}(2) &= f(u_1(2), y_1(2)) = f(0.02, 0.7344 \times 10^{-3}) \\ &= 0.0709 \end{aligned} \tag{16}$$

Similarly from Equation (6)

$$\dot{y_2}(s) = \frac{-2.814 \times 10^{-7}}{(s + 0.03977)} u_2(s) \tag{17}$$

$$\dot{y_2} = -2.814 \times 10^{-7} u_2 - 0.03977 y_2 \tag{18}$$

1st iteration:
Taking '0' initial condition and step size of '0.01':
$u_2(0) = 0,\ y_2(0) = 0,\ h = 0.01$

$$\begin{aligned} K_1 &= f(0, 0) = 0 \\ K_2 &= -1.407 \times 10^{-9} \\ K_3 &= -1.3790 \times 10^{-9} \\ K_4 &= -2.7592 \times 10^{-9} \end{aligned} \tag{19}$$

$$y_2(1) = -1.3885 \times 10^{-11} \tag{20}$$

$$\dot{y_2}(1) = -2.8134 \times 10^{-9} \tag{21}$$

2nd iteration:
$u_2(1) = 0.01,\ y_2(1) = -1.3885 \times 10^{-11},\ h = 0.01$

$$\begin{aligned} K_1 &= -2.8134 \times 10^{-9} \\ K_2 &= -4.1645 \times 10^{-9} \\ K_3 &= -4.1376 \times 10^{-9} \\ K_4 &= -5.4635 \times 10^{-9} \end{aligned} \tag{22}$$

$$y_2(2) = -5.5353 \times 10^{-11} \tag{23}$$

$$\dot{y_2}(2) = -5.6258 \times 10^{-9} \tag{24}$$

Similarly from Equation (6)

$$\dot{y_3}(s) = \frac{-1.406 \times 10^{-7}}{(s + 0.02577)} u_3(s) \tag{25}$$

$$\dot{y_3} = -1.406 \times 10^{-7} u_3 - 0.02577 y_3 \tag{26}$$

1st iteration:
$u_3(0) = 0,\ y_3(0) = 0,\ h = 0.01$

$$\begin{aligned} K_1 &= 0 \\ K_2 &= -7.03 \times 10^{-10} \\ K_3 &= -6.9394 \times 10^{-10} \\ K_4 &= -1.3881 \times 10^{-9} \end{aligned} \tag{27}$$

$$y_3(1) = -6.9699 \times 10^{-12} \tag{28}$$

$$\dot{y_3}(1) = -1.4058 \times 10^{-9} \tag{29}$$

2nd iteration:
$u_3(1) = 0.01,\ y_3(1) = -6.9699 \times 10^{-12},\ h = 0.01$

$$\begin{aligned} K_1 &= -1.4058 \times 10^{-9} \\ K_2 &= -1.3877 \times 10^{-9} \\ K_3 &= -2.0909 \times 10^{-9} \\ K_4 &= -2.7579 \times 10^{-9} \end{aligned} \tag{30}$$

$$\begin{aligned} y_3(2) &= y_3(1) + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4) \\ &= -2.5505 \times 10^{-11} \end{aligned} \tag{31}$$

$$\begin{aligned} \dot{y_3}(2) &= f(0.02, -2.5505 \times 10^{-11}) \\ &= -2.8113 \times 10^{-9} \end{aligned} \tag{32}$$

Substituting the value of Equations (13), (21) and (29) in Equation (8), we get

$$\begin{aligned} \dot{y}(1) &= \dot{y_1}(1) + \dot{y_2}(1) + \dot{y_3}(1) \\ &= 0.03733 + (-2.814 \times 10^{-9}) + (-1.4058 \times 10^{-9}) \\ &= 0.0373 \end{aligned} \tag{33}$$

For Inlet velocity $u_1 = 0.6 m/s$, liquid temperature $u_2 = 80^0 C$, liquid density $u_3 = 900 kg/m^3$, Height of the liquid in the tank:

$$\dot{y}(1) = 0.6 \times \dot{y_1}(1) + 80 \times \dot{y_2}(1) + 900 \times \dot{y_3}(1) = 0.0224 \tag{34}$$

For a tank height of 24cm,

$$Level = \frac{0.0224 \times 100}{0.24} = 9.3333\% \tag{35}$$

Substituting the value of Equations (16), (24) and (32) in Equation (8), we get

$$\begin{aligned} \dot{y}(2) &= \dot{y_1}(2) + \dot{y_2}(2) + \dot{y_3}(2) \\ &= 0.0709 + (-5.6258 \times 10^{-9}) + (-2.8113 \times 10^{-9}) \\ &= 0.0709 \end{aligned} \tag{36}$$

For Inlet velocity $u_1 = 0.6 m/s$, liquid temperature $u_2 = 80^0 C$, and liquid density $u_3 = 900 kg/m^3$, Height of the liquid in the tank:

$$\dot{y}(2) = 0.6 \times \dot{y_1}(2) + 80 \times \dot{y_2}(2) + 900 \times \dot{y_3}(2) = 0.0425 \tag{37}$$

For a tank height of 24cm,

$$Level = \frac{0.0425 \times 100}{0.24} = 17.7083\% \tag{38}$$

Similarly, the levels can be computed for the next iterations. The system is a Multiple Input Single Output system whose inputs are inlet velocity, temperature, and density and the output is the liquid's height in the tank. The differential equation solution gives the height of the liquid in the tank, which is further converted to the liquid level in the tank. For this system, the observer can be designed using various methods as described in upcoming sections.

III. Luenberger Observer Design

A new parametric observer-based approach for liquid level detection in multivariable linear systems is proposed in this section. Possibly the most useful observer form is the Luenberger observer. The Luenberger Observer (LO) has a very simple design, which makes it an interesting general design method. This section describes the construction of a Luenberger observer. Sensors frequently result in phase lag, attenuation, and noise issues. Phase lag and attenuation are caused by either the sensor's physical design or by sensor filters, which are frequently employed to reduce noise. Selecting a quicker or more accurate sensor will offer advantages that are more predictable and manageable than utilizing an observer. The observer's objective is to make the best use of the sensor that is being utilized, but constraints like cost, size, and reliability will typically oblige the designer to accept sensors with undesired characteristics. The transfer function with respect to input 1(inlet velocity) is,

$$G_1(s) = \frac{1.255}{s + 3.322} \tag{39}$$

The transfer function with respect to input 2(liquid temperature) is,

$$G_2(s) = \frac{0.2907 \times 10^{-4}}{s + 3.322} \tag{40}$$

The transfer function with respect to input 3(liquid density) is,

$$G_3(s) = \frac{-0.2134 \times 10^{-4}}{s + 3.322} \tag{41}$$

From Equations (39), (40) and (41), the overall system can be represented as,

$$y(s) = \frac{1.255}{s + 3.322} u_1(s) + \frac{(0.2907 \times 10^{-4})}{s + 3.322} u_2(s) + \frac{(-0.2134 \times 10^{-4})}{s + 3.322} u_3(s) \tag{42}$$

Given that the order of the system =1= Rank of the system [61], the system is observable.
Open loop settling time,

$$t_s = \frac{4}{a} = \frac{4}{3.322} = 1.2041 seconds \tag{43}$$

For a settling time of $0.5sec$,

$$0.5sec = \frac{4}{a} \tag{44}$$

$$a = 8$$

Pole of the system without an observer,

s=-3.322

Pole of the system with an observer,

s=-8

Reducing the settling time to $0.5sec$, the new characteristic equation is,

$$s + 8 = 0 \tag{45}$$

Therefore $s = -8$ can be chosen as a new pole for designing the observer using the pole placement technique. The state space representation of the system is,

$$\dot{x}(t) = [-3.322]x(t) + [-9.4500 \ -0.0002 \ 0.0002]u(t) \tag{46}$$

$$y(t) = [-0.1328]x(t) + [0\,0\,0]u(t) \tag{47}$$

From equations (46) and (47),
A=$\begin{bmatrix}-3.3220\end{bmatrix}$
B=$\begin{bmatrix}-9.4500 & -0.0002 & 0.0002\end{bmatrix}$
C= -0.1328
D=$\begin{bmatrix}0 & 0 & 0\end{bmatrix}$
The Luenberger observer gain [62] is calculated as shown below. Using Ackermann's formula,

$$L = \phi(A) \begin{bmatrix} C \\ CA \\ . \\ . \\ CA^{n-2} \\ CA^{n-1} \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 0 \\ 0 \\ . \\ . \\ 1 \end{bmatrix} \tag{48}$$

$$\phi(A) = A^n + \alpha_1 A^{n-1} + ... + \alpha_{n-1} A + \alpha_n I \tag{49}$$

$$\phi(A) = A + \alpha = A + 8 = 4.678$$

$$L = \phi(A)[C]^{-1}[1]$$

$$L = [4.678][-0.1328]^{-1}$$

$$L = [-35.2259] \tag{50}$$

The overall representation of the system with the observer is,

$$y(s) = \frac{1.255}{s + 8} u_1(s) + \frac{(0.2907 \times 10^{-4})}{s + 8} u_2(s) + \frac{(-0.2134 \times 10^{-4})}{s + 8} u_3(s) \tag{51}$$

Using Range-Kutta method of order 4 as mentioned earlier,

$$\dot{y}(1) = \dot{y_1}(1) + \dot{y_2}(1) + \dot{y_3}(1) = 0.0121 \tag{52}$$

For $u_1 = 0.6m/s, u_2 = 80^0C, u_3 = 900kg/m^3$,

$$\dot{y}(1) = 0.6 \times \dot{y_1}(1) + 80 \times \dot{y_2}(1) + 900 \times \dot{y_3}(1) = 0.0071 \quad (53)$$

$$Level = \frac{0.0071 \times 100}{0.24} = 2.9583\% \quad (54)$$

$$\dot{y}(2) = \dot{y_1}(2) + \dot{y_2}(2) + \dot{y_3}(2) = 0.0232 \quad (55)$$

For $u_1 = 0.6m/s, u_2 = 80^0C, u_3 = 900kg/m^3$,

$$\dot{y}(2) = 0.6 \times \dot{y_1}(2) + 80 \times \dot{y_2}(2) + 900 \times \dot{y_3}(2) = 0.0136 \quad (56)$$

$$Level = \frac{0.0136 \times 100}{0.24} = 5.6667\% \quad (57)$$

Luenberger observer has been designed for the mathematical obtained from the MISO system. The differential equation obtained from the system with the observer is solved using the Range-Kutta method of order 4, which calculates the tank's liquid level. The error between the actual and estimated output is found to be minimum. The designed observer does not incorporate process and measurement noise [63]. Hence Kalman filter is used to include these uncertainties. One way to look about Kalman Observer (KO) is as an expansion of LO. The KO (filter) is one of the better observers against a variety of disturbances [64].

IV. Kalman Filter

The Kalman filter combines measuring and prediction to provide the best estimate of the liquid level in the tank. It is an iterative technique that employs a series of equations and subsequent data inputs to swiftly estimate the real value of the item being measured when the estimated results contain unpredicted or random error, uncertainty, or variation [65]. It can overcome the non-linear state observation by using a linearized approximation [66]. The working of the Kalman filter is shown in Fig. 8.

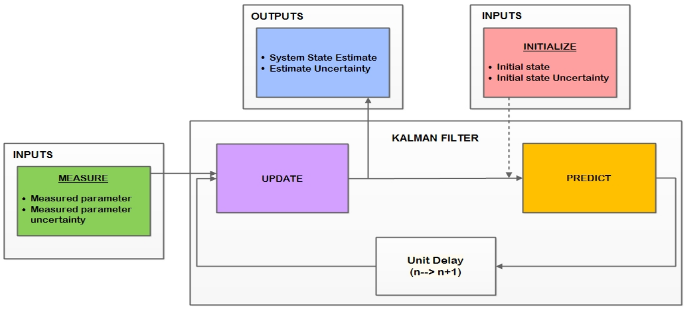

Fig. 8. Working of Kalman filter

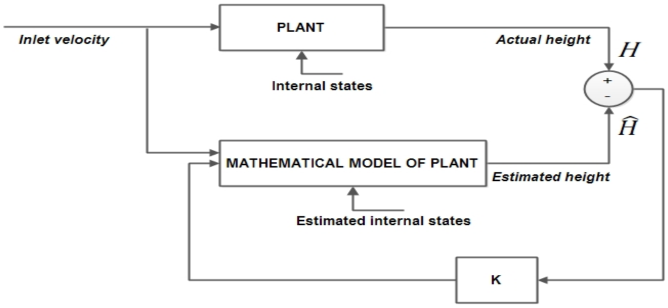

Fig. 9. Estimation of parameters using a mathematical model of the system

The derived mathematical model merely approximates the actual system. It encounters several uncertainties. If the model outputs are perfect with no uncertainties and the actual system model has the same initial conditions, then the calculated and predicted output values exactly match each other. So in this case, the model outputs are without uncertainties and the system's output would match the estimated height of the tank. This is the reason for incorporating a state estimator to estimate the internal states of the system. If the output and estimated value are the same, the model will converge to a real system. Hence it is important to minimize the difference between estimated and measured output values. Fig. 9 shows the closed loop system with the observer having a definite gain (K) which tries to eliminate the error between estimated and measured output values such that the internal states that are estimated drive to their true values. This system estimates the output, i.e., the liquid level in the tank based on the input velocity for different liquid densities, temperature having a definite sensor placement position. There are 3 major steps involved in the process of state estimation when we have a set of measured values.

1) Calculation of Kalman gain.
2) Calculating the current estimate.
3) Calculating the updated value of error in the estimate.

The value of Kalman gain lies between 0 to 1. Its value depends on an error in the estimate and an error in the measurement. It gives more importance to the one whose error is lesser. Based on the calculated value of Kalman gain, the estimate is re-calculated. Kalman gain decides the weightage to be given to the previous estimate and the new measurement value and calculates the new value of the estimate. The newly calculated value of error in the estimated is fed back to re-calculate Kalman gain. Fig. 10 describes the steps involved in the Kalman filter design to update the estimate.

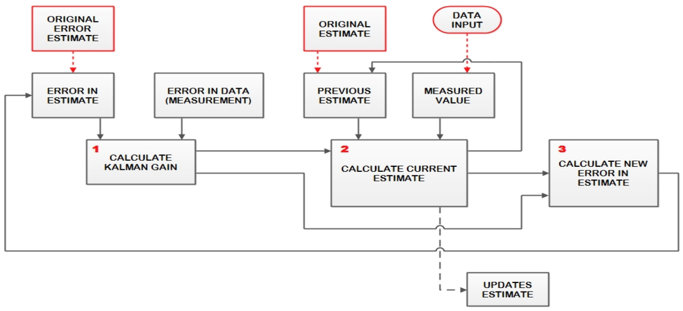

Fig. 10. State estimation using Kalman filter

Kalman filter helps decide how many portions of the prediction and new measurements are to be added to predict new estimates to minimize the error. X_0 and P_0 represent the initial state matrix and initial process covariance matrix respectively. The initial state matrix can be uni-dimensional or multidimensional, depending on the process. The process covariance matrix denotes the error in the estimate. Newly predicted state estimation depends on the physical model and previous state. It depends on control variables of the system (u), predicted state noise (w) , and process noise covariance (Q). It then calculates the measurement of the state using uncertainty in the measurement called measurement noise. The value of Kalman gain is calculated based on process covariance and measurement error [67]. It decides how many fractions of measurement and prediction are to be used for updation. Based on this, the state and process covariance get updated for the next iteration. Fig. 11 describes how the Kalman filter algorithm works to predict the next state from its initial state with Kalman gain.

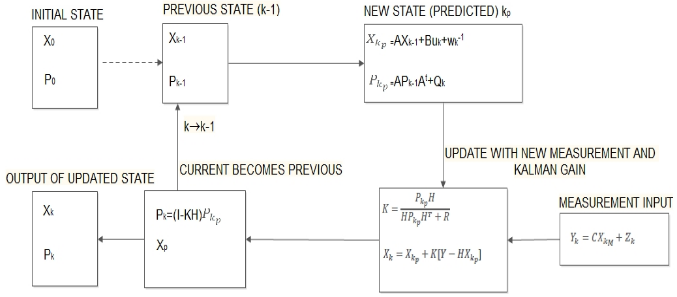
INITIAL STATE
X_0
P_0
PREVIOUS STATE (k-1)
X_{k-1}
P_{k-1}
NEW STATE (PREDICTED) k_p
X_{k_p} =AX_{k-1}+Bu_k+w_k^{-1}
P_{k_p} =$AP_{k-1}A^t$+Q_k
k→k-1
CURRENT BECOMES PREVIOUS
UPDATE WITH NEW MEASUREMENT AND KALMAN GAIN
OUTPUT OF UPDATED STATE
X_k
P_k
P_k=(I-KH)P_{k_p}
X_p
$K = \frac{P_{k_p}H}{HP_{k_p}H^T + R}$
$X_k = X_{k_p} + K[Y - HX_{k_p}]$
MEASUREMENT INPUT
$Y_k = CX_{k_M} + Z_k$

Fig. 11. Kalman filter algorithm

$$Error\,(e) = Actual\,Height(H) - Estimated\,Height(\widehat{H}) \tag{58}$$

Gain K of the estimator must be computed such that the error between measured and estimated output is minimum. To represent the observer mathematically, the problem can be generalized by taking input as $'u'$, output as $'y'$ and state to be estimated as $'x'$. Error dynamics can be computed using the following equations.

$$e = x - \widehat{x} \tag{59}$$

$$\dot{e} = \dot{x} - \dot{\widehat{x}} \tag{60}$$

$$\dot{x} = Ax + Bu \tag{61}$$

$$y = Cx \tag{62}$$

$$\dot{\widehat{x}} = A\widehat{x} + Bu + K(y - \widehat{y}) \tag{63}$$

$$\widehat{y} = C\widehat{x} \tag{64}$$

Subtracting Equation (63) from (61),

$$\dot{x} - \dot{\widehat{x}} = Ax - A\widehat{x} + Bu - Bu + K(y - \widehat{y}) \tag{65}$$

Subtracting Equation (64) from (62),

$$y - \widehat{y} = C(x - \widehat{x}) = C(e) \tag{66}$$

Substituting Equation (59), (60), (62) and (64) in Equation (65), we get

$$\dot{e} = Ae + K(C(x - \widehat{x})) \tag{67}$$

$$\dot{e} = (Ae) + KCe \tag{68}$$

$$\dot{e} = (A + KC)e \tag{69}$$

Solution to Equation (69) is,

$$\dot{e} = e^{(A+KC)t}e(0) \tag{70}$$

If $(A + KC) < 0$, then $e \to 0$ as $t \to \infty$.So $\widehat{x} \to x$.

The significance of having a feedback loop around the observer is that the decay rate of the error function can be controlled by selecting the observer gain $'K'$ accordingly. Having a feedback controller gives more control over Equation (70) and guarantees a faster elimination of the error. Faster the error vanishes, faster is the estimated state $\widehat{x}$ converging to the true state $'x'$. The value of $'K'$ can be optimized by choosing the appropriate Kalman filter. Kalman filter considers measurement noise $'V_k'$ which is a random variable and process noise $'W_k'$ which is also a noise representing the output measurement dynamics. These random variables [39] do not follow a definite pattern, but using probability theory, their average properties can be known.

$$V_k \sim N(0, R) \tag{71}$$

$$W_k \sim N(0, Q) \tag{72}$$

Kalman filter combines measurement and prediction to find the optimal estimate of the height of liquid level in the tank by considering randomness. Optimal state estimation can be obtained by multiplying prediction and measurement probability function together [68]. Hence modified equation with the addition of measurement and process noise of the Kalman filter are as shown in Equation (73).

$$\widehat{x}_t = A\widehat{x}_{t-1} + Bu_t + K_t(y_t - C(A\widehat{x}_{t-1} + Bu_t)) \tag{73}$$

$A\widehat{x}_{t-1} + Bu_t$ predicts the current state using the state estimate from the previous time step and the current input which is calculated before the measurement is estimated. This is similar to Equation (63).

$$A\widehat{x}_{t-1} + Bu_t = \widehat{x}_t^- \tag{74}$$

where, $\widehat{x}_t^-$ is the prior estimate.Hence,

$$\widehat{x}_t = \widehat{x}_t^- + K_t(y_t - C\widehat{x}_t^-) \tag{75}$$

where,

$\widehat{x}_t$ is the posterior estimate, $\widehat{x}_t^-$ is the prediction and $K_t(y_t - C\widehat{x}_t^-)$ is the updation. $K_t(y_t - C\widehat{x}_t^-)$ uses the measurement and incorporates it into the prediction to update the prior estimate and this result is called a posterior estimate [69].

A. Prediction:

$$\widehat{x}_t^- = A\widehat{x}_{t-1} + Bu_t \tag{76}$$

$$P_t^- = AP_{t-1}A^T + Q \tag{77}$$

where, P is the error co-variance of the prior estimate. It is the measure of uncertainty in the estimated state. This variance comes from the process noise $\widehat{x}_{t-1}$. It uses prior estimates calculated in the prediction step to update posterior estimates of the state and error co-variance. The Kalman gain is calculated such that it minimizes the posterior error co-variance.

B. Updation

$$K_t = \frac{P_t^- C^T}{CP_t^- C^T + R}$$

$$\widehat{x}_t = \widehat{x}_t^- + K_t(y_t - C\widehat{x}_t^-)$$

$$P_t = (I - K_t C)P_t^-$$

If the measurement co-variance is zero $(R \to 0)$, the posterior estimate will be equal to the measurement.

$$\lim_{R \to 0} K_t = \lim_{R \to 0} \frac{P_t^- C^T}{CP_t^- C^T + R} \tag{78}$$

$$= \lim_{R \to 0} \frac{1}{C} = 1$$

$$\widehat{x}_t = \widehat{x}_t^- + K_t(y_t - C\widehat{x}_t^-) \tag{79}$$

$$= \widehat{x}_t^- + C^{-1}(y_t - C\widehat{x}_t^-)$$

$$\widehat{x}_t = y_t$$

On the other hand, if the prior error co-variance goes to zero, the Kalman gain is found to be zero.

$$\lim_{P_t^- \to 0} K_t = \lim_{P_t^- \to 0} \frac{P_t^- C^T}{CP_t^- C^T + R}$$

$$= \lim_{P_t^- \to 0} \frac{0}{0 + R} = 0$$

$$\widehat{x}_t = \widehat{x}_t^- + K_t(y_t - C\widehat{x}_t^-)$$

$$= \widehat{x}_t^- + 0(y_t - C\widehat{x}_t^-)$$

$$\widehat{x}_t = \widehat{x}_t^-$$

Using the value of prediction and Kalman gain, the algorithm updates system states even though uncertainties exist. The results can also be verified using ANN prediction algorithms as described in section V.

V. Prediction Using Artificial Neural Network Model

Neural networks are well known for classification and prediction. Neural networks here refer to Artificial Neural Networks (ANNs). ANN is a layered network of artificial neurons. It typically consists of an input layer, one or more hidden layers, and an output layer. Each layer is composed of a number of artificial neurons, also called nodes. The artificial neurons in one layer are connected by weights to the artificial neurons in the next layer as shown in Fig. 12. It explains a simple model of the ANN which helps in solving complex mathematical problems. In the present work, ANN can be used for building a mathematical model of the observations for making predictions [70], [71]. Here X is the input parameters and Y is the liquid level. Normalized output data using standard scaling is used for the supervised learning technique with the output data obtained from ANSYS. Using hyperparameter searching, the total number of parameters (weights and biases) are decided. The trained model is tested for its accuracy and mean squared error. The liquid level is estimated with an optimization algorithm which gives the least mean squared error. The complete details are described below. The plant's NNs model are built during the system identification process, and the system should thereafter be developed or trained utilizing the developed model [72].

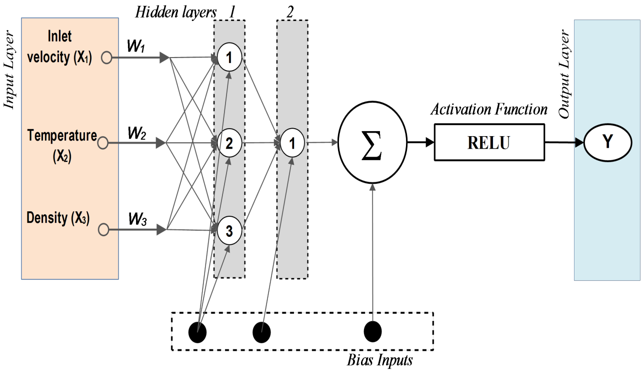

Fig. 12. Architecture of Artificial Neural Network

Steps for building the ANN model using Python are given below.

A. Loading the data

Simulated data in ANSYS are loaded to the pandas data frame. Here input data is a vector consisting of inlet velocity, temperature and density. The output vector comprises the height of the liquid in the tank.

B. Pre-processing of dataset

'Keras' is the most used deep learning framework which makes new experiments run easier. Keras is a simple, consistent interface optimized for normal use. It gives clear and noteworthy feedback on the error made by the end user. Keras models are built by connecting configurable building blocks. It is possible to modify building blocks accordingly to convey new ideas for research by creating new layers, metrics, loss functions, and state-of-the-art model development. In this work, the data collected using ANSYS simulation consists of height calculated as a dependent variable for different values of fluid inlet velocity, temperature and density. After importing the required packages, the data set is imported. The imported data, after installing the required packages, are normalized since they have different units and ranges. This is done to avoid non-correlation between the input and the output values.

C. Training the model and building the model using hyperparameter searching

The model is trained for 100 epochs. A summary of the model built is given in Table I.

TABLE I
MODEL SUMMARY

Layers	Output Shape	Parameters
Input	-	-
Hidden layer 1	(1,3)	12
Hidden layer 2	(1,1)	4
Output	-	-

Total Parameters = 16
Trainable parameters = 16
Activation function = Rectified Linear Unit
Optimizer = Stochastic Gradient Descent (SGD)
Loss function = Mean Squared Error
Learning rate= 0.01

Here the model is built using a neural network having hidden layers between 2 to 6. Here, a random search is carried out within the model to find the best possible neural network that can be created. Among 0.01, 0.001 and 0.0001 best learning rate is chosen based on a random search. An activation function is one of the main parts of the Artificial Neural Network which helps to derive output from a set of input values given to the node (or a layer). By activation function, the weight and bias could express the non-linear relationship. There are majorly 3 types of activation functions used in neural networks namely binary step activation function, linear activation function, and complex non-linear activation function. Here, Rectified linear unit activation function (ReLU) activation function has been applied. Being one of the non-linear Neural Network activation functions, the main intention of using the ReLU activation function is it overcomes the vanishing gradient problems which occur when there are many hidden layers in the networks. It also allows the model to learn faster and perform better than the other two activation functions. The mathematical equation for Relu is shown below in (80)

$$ReLU(x) = max(0, x)$$
$$ReLU'(x) = \begin{cases} 0 & \text{x<0} \\ 1 & \text{x}\geq 0 \end{cases} \tag{80}$$

Here if the input is negative, the output of ReLU is '0' and for positive values, it is 'x'. Optimizers are algorithms or methods for altering the characteristics of a neural network such as weights and learning rate to reduce the losses. By minimizing the function, optimizers are used to solve optimization problems. Adam optimizer is used for compilation as it results in lease Mean Squared Error (MSE) of 6.3487×10^{-7}.

D. Loss functions and Optimizers

The process of minimizing (or maximizing) any mathematical expression is called optimization. Optimizers are algorithms or methods used to change the attributes of the neural network such as weights and learning rate to reduce the losses. Here learning rate is a hyperparameter [73] value. Gradient means slope or rate of inclination. So gradient descent means descending a slope to reach the lowest point on that surface [74]. Stochastic denotes randomness. SGD chooses one of the measured values during every iteration to minimize the number of computations. Data can also be split into a small group of data points called 'batch' rather than just one point at every step which is called the "mini-batch" gradient descent algorithm [75].

$$w_{t+1} = w_t - \eta \nabla w \tag{81}$$

$$\nabla w = \sum_{i=1}^{N} \frac{\partial}{\partial w}(f(x_i) - y_i)^2 \tag{82}$$

Momentum-based Gradient Descent utilizes the history of gradients to filter the update. It calculates an exponentially weighted average of the gradients and then uses that gradient to update the weights. It has a greater execution speed when compared to the gradient descent algorithm.

$$V_t = \gamma \times V_{t-1} + \eta \nabla w_t \tag{83}$$

$$w_{t+1} = w_t - V_t \tag{84}$$

Adagrad increases the robustness of SGD and utilizes it for training neural networks comprising large-scale data.

$$v_t = v_{t-1} + (\nabla w_t)^2 \tag{85}$$

$$w_{t+1} = w_t - \frac{\eta}{\sqrt{v_t} + \epsilon} \nabla w_t \tag{86}$$

The RMSprop Optimizer takes larger steps by increasing the learning rate to restrict the oscillation in a vertical direction to converge faster.

$$V_t = \beta \times V_{t-1} + (1-\beta)(\nabla w_t)^2 \tag{87}$$

$$w_{t+1} = w_t - \frac{\eta}{\sqrt{v_t} + \epsilon} \nabla w_t \tag{88}$$

Adam Optimizer combines RMSprop and SGD with momentum, which squares gradients to scale the learning rate.

$$m_t = \beta_1 \times v_{t-1} + (1-\beta_1)(\nabla w_t) \tag{89}$$

$$v_t = \beta_2 \times v_{t-1} + (1-\beta_2)(\nabla w_t)^2 \tag{90}$$

$$w_{t+1} = w_t - \frac{\eta}{\sqrt{v_t} + \epsilon} m_t \tag{91}$$

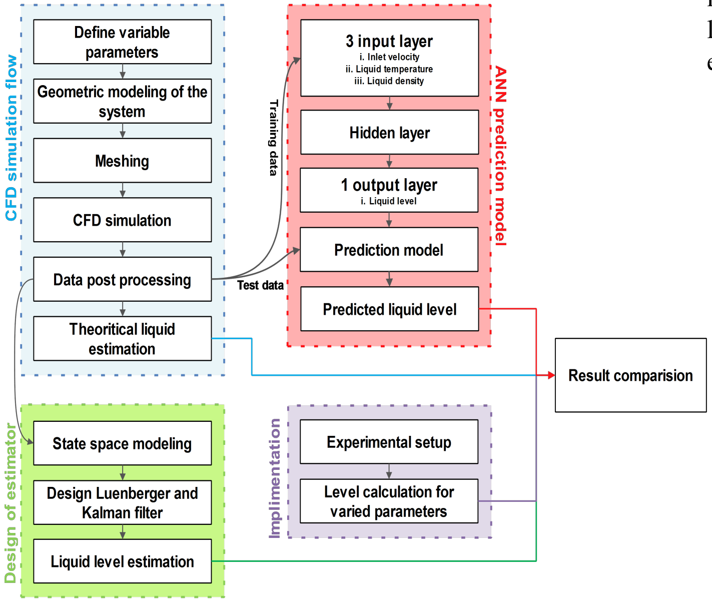

Fig. 13. The workflow of CFD simulation, design of estimator, ANN prediction model and real-time implementation

VI. Results

The previously calculated values of liquid level (height of the tank) for different inlet velocities, temperature, and density for a definite sensor position are used to design multiple input single output (MISO) system. Inlet velocity, temperature, and density are inputs to the system, and liquid level, which is calculated using the tank's height, is the output of this MISO system. System identification helps in designing mathematical models based on the observed data. Based on the experimental design, data is pre-processed and a definite model structure for parameter estimation is obtained. $G_1(s)$,$G_2(s)$ and $G_3(s)$ are the transfer functions(gains) with respect to inputs i.e., inlet velocity, temperature and density respectively. Using this model structure, a chaotic equation of the system is obtained and the resulting differential equation is solved using the Range-Kutta method.

Based on the open loop characteristics, a full state observer is designed for settling time $0.5sec$, which is $1.2041sec$ for an open loop system. Luenberger observer is designed whose gain is calculated theoretically as -35.2259 and matches with MATLAB results. Fig. 14 represents liquid level estimation in a tank using the MISO system with the observer. The level of liquid measured in the tank is 94.0535% and the estimated value of the same is 93.3566% as shown in Fig. 14. The error between the responses is plotted in Fig. 15 and this error is found to be very less settling down to 0.6448%. The mean squared error from the graph is calculated to be 0.00022 for the estimation of the height of the liquid in the tank and 38.24% for the level of liquid in the tank. It is observed that error reduces to minimum value taking very less time($time < 1sec$). The liquid level settles down with very less transients and with minimum error.

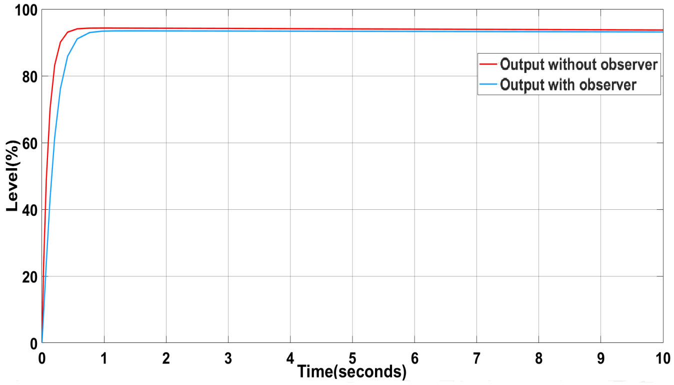

Fig. 14. Estimation of liquid level in the tank with an observer

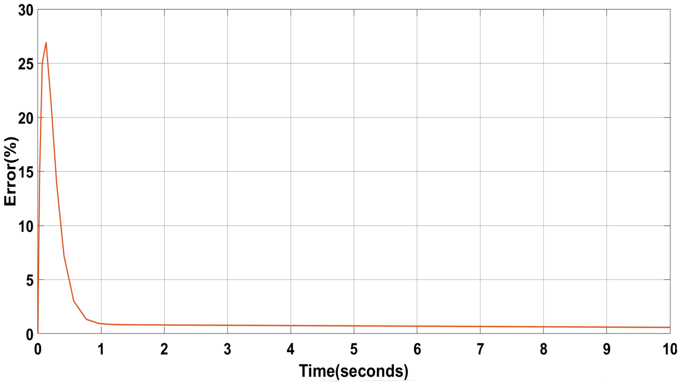

Fig. 15. Error between the plant and output

To incorporate the process and measurement noise, a linear quadratic estimation algorithm based on Kalman filtering is used. It uses a set of observed data consisting of inaccuracies, to estimate unknown variables with high accuracy. Kalman filter designed for the MISO system is as shown in Fig. 16. For the estimation, the value of measurement noise R is chosen to be 0.1 and process noise Q is 0.1. Fig. 17 shows the observed data with high randomness because of the measurement and process noise. Fig. 18 shows the plot of estimated values of tank height which is accurate with less randomness.

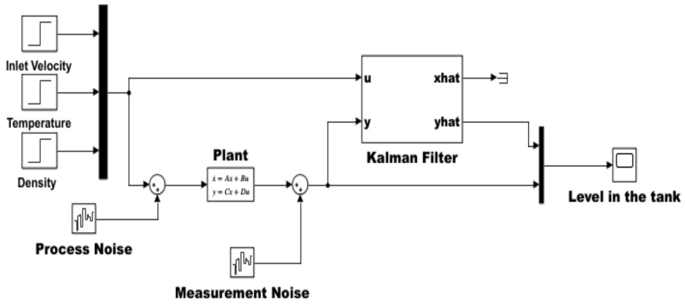

Fig. 16. Plant with Kalman filter

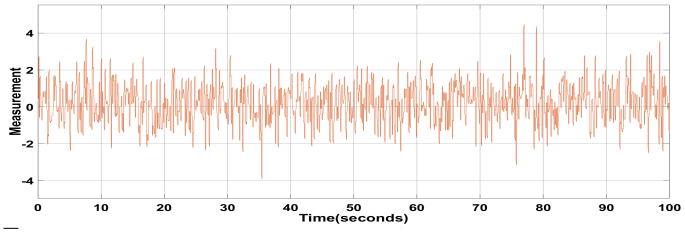

Fig. 17. Measurement with uncertainties

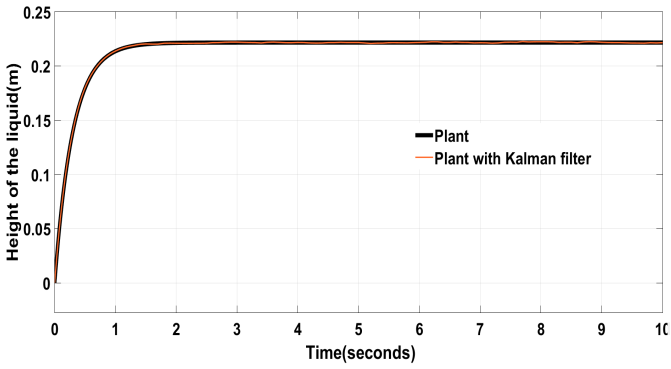

Fig. 18. The output of the system with Kalman filter

The regression model is built using ANN for the observed data. Observed data is imported using the 'Pandas' data frame and pre-processed. Since the independent variables lie in different ranges, they are first normalized using standard scaling and then split into test and train data. ANN model is built with three input layers and one output layer. Rectified linear activation function, which is known as 'RELU' is used as the activation function. By using different optimization algorithms, performance metrics accuracy and loss are determined. The model is run for 100 epochs and tested with test data, and the prediction result is shown in Fig. 19. Similarly, the model is run for different optimization functions the loss functions are plotted for 100 epochs which is shown in Fig. 20. Here in this work, the ANN architecture of 2 layers has been taken for validation purposes. Our approach is based on:

- Formulation of an artificial neural network (ANN) using ANSYS simulation data's input and output variable records
- Using the data, train and test the ANN
- ANN behavior verification using data not utilized for training or testing
- Comparing the ANN's performance against a simulator that was already developed

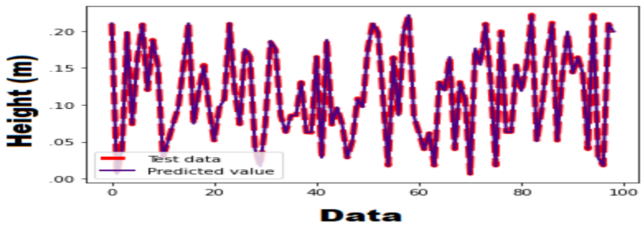

Fig. 19. Prediction for inlet velocity=$0.6 m/s$, temperature=$80°C$ and density=$900 kg/m^3$

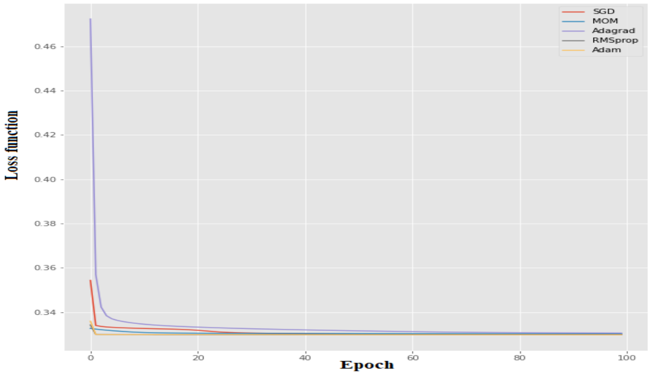

Fig. 20. Loss functions for 100 epochs

Table 2 compares the height of the liquid calculated using different methods.

TABLE II
COMPARISON OF LIQUID HEIGHT IN THE TANK USING VARIOUS METHODS

	Inlet Velocity(m/s)	Temperature(°C)	Density(kg/m^3)	Liquid level(m)	MSE
Actual model	0.6	80	900	0.2250	-
Luenberger observer	0.6	80	900	0.2236	0.0002
Kalman filter	0.6	80	900	0.2215	1.2309
ANN prediction	0.6	80	900	0.2224	6.3487×10^{-7}

VII. VALIDATION USING PRACTICAL SETUP

For validation, a pressure sensor is connected to a panel held vertically as shown in Fig. 21. The flow rate is adjusted by increasing or decreasing the flow bypass manual valve. Rotameter is used to measure the flow rate in LPH. Liquid passing through the pressure sensor is collected in the tank below. Table III compares the difference between actual and measured values for water with different values of inlet velocity at room temperature. Differences in actual and measured height are plotted in Fig. 22. The flow rate is converted to the inlet velocity using the Equation (92). From the volume obtained, the level is calculated using the tank dimensions.

$$Q = V \times A \tag{92}$$

where Q= Flow rate, V= Inlet velocity, A= Cross-sectional area of the measurement location

TABLE III
COMPARISON OF PRACTICAL AND SIMULATED VALUES OF LIQUID HEIGHT IN THE TANK

Inlet velocity(m/s)	Liquid height $h_{actual}(m)$	Practical $h_{measured}(m)$	Simulated $h_{measured}(m)$
0.0109	0.09	0.10	0.084
0.0263	0.15	0.14	0.1463
0.0439	0.21	0.22	0.2087
0.0658	0.27	0.29	0.2711
0.0877	0.35	0.34	0.3519
0.1096	0.42	0.40	0.4250

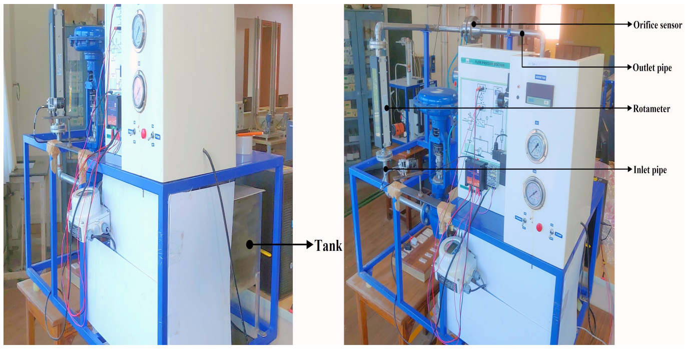

Fig. 21. Real-time setup

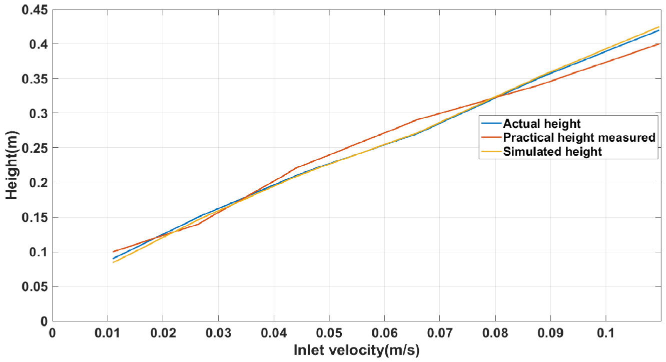

Fig. 22. Comparison of actual and simulated

VIII. CONCLUSION

The estimation of a tank's liquid level is described in this work. To evaluate the laminar type of water flow via an orifice meter with a constant pipe length and orifice diameter, a CFD simulation is performed. For varying inlet velocity, temperature, and density, the flow rate is calculated. By using the Range-Kutta method of order 4, a solution for the differential equation of the MISO system is obtained for the calculation of the liquid level in the tank. Using various approaches, the Luenberger observer is designed for the same system. To incorporate process and measurement noise Kalman filter is added to the system. Results obtained are also verified using an artificial neural network prediction model. For an inlet velocity of $0.6m/s$, temperature $80°C$ and liquid density $900kg/m^3$, the height of the liquid is estimated to be $0.2250m$ by Luenberger observer, $0.2236m$ by Kalman filter and $0.2224m$ by ANN prediction model. Various optimization algorithms are used to minimize the error. The mean squared error calculated was found to be 0.00022,1.2309, 6.3487×10^{-7} for Luenberger observer, Kalman filter, and ANN prediction model respectively. Using the experimental setup, actual values and simulated values are compared and the error is found to be minimum. As a future work, the experiments is carried out for wider range of input parameters like liquid density, temperature, inlet velocity, liquid type etc and study was carried out for different sensor placements i.e., where the sensor needs to be and in what position. This can be extended for estimating the state of a linear system with unknown inputs using an Unknown Input Observer (UIO) [77]. Also Kalman filters are linear quadratic estimators, which assumes that both system and observer models to be linear which may not be real in many situations. Using extended kalman filters can solve the problems with non-linearity for estimation.

REFERENCES

[1] M. S. Shah, J. B. Joshi, A. S. Kalsi, C. S. R. Prasad, and D. S. Shukla, "Analysis of flow through an orifice meter: CFD simulation," Chemical Engineering Science, vol. 71, pp. 300–309, Mar. 2012.

[2] A. Rocchi, E. Santecchia, F. Ciciulla, P. Mengucci, and G. Barucca, "Characterization and optimization of level measurement by an ultrasonic sensor system," IEEE Sens. J., vol. 19, no. 8, pp. 3077–3084, 2019.

[3] T. Islam, O. P. Maurya, and A. U. Khan, "Design and fabrication of fringing field capacitive sensor for non-contact liquid level measurement," IEEE Sens. J., vol. 21, no. 21, pp. 24812–24819, 2021.

[4] R. He, C. Teng, S. Kumar, C. Marques, and R. Min, "Polymer Optical Fiber Liquid Level Sensor: A Review," IEEE Sens. J., vol. 22, no. 2, pp. 1081–1091, 2022.

[5] G. Dou, R. Chen, C. Han, Z. Liu, and J. Liu, "Research on water-level recognition method based on image processing and convolutional neural networks," Water (Basel), vol. 14, no. 12, p. 1890, 2022.

[6] G. Chen, T. Sun, P. Wang, and B. Sun, "Design of temperature compensation system of pressure sensors," 2006 IEEE International Conference on Information Acquisition, 2006.

[7] P. Esmaili, F. Cavedo, and M. Norgia, "Differential Pressure-Based Densitometer in Dynamic Condition," IEEE Transactions on Instrumentation and Measurement, vol. 70, pp. 1–7, 2021.

[8] C. F. Lui, Y. Liu, and M. Xie, "A supervised bidirectional long short-term memory network for data-driven dynamic soft sensor modeling," IEEE Trans. Instrum. Meas., vol. 71, pp. 1–13, 2022.

[9] M. S. Zarnik and D. Belavic, "Study of LTCC-based pressure sensors in water," Sens. Actuators A Phys., vol. 220, pp. 45–52, 2014.

[10] M. S. Zarnik, D. Belavic, and S. Macek, "The warm-up and offset stability of a low-pressure piezoresistive ceramic pressure sensor," Sens. Actuators A Phys., vol. 158, no. 2, pp. 198–206, 2010.

[11] P. Esmaili, F. Cavedo, and M. Norgia, "Characterization of pressure sensor for liquid-level measurement in sloshing condition," IEEE Trans. Instrum. Meas., vol. 69, no. 7, pp. 4379–4386, 2020.

[12] P. Esmaili, F. Cavedo, A. Pesatori, and M. Norgia, "Liquid level measurement through capacitive pressure sensor," in 2020 IEEE International Instrumentation and Measurement Technology Conference (I2MTC), 2020, pp. 1–5.

[13] P. C. Joshi, N. B. Chopade, and B. Chhibber, "Liquid level sensing and control using inductive pressure sensor," in 2017 International Conference on Computing, Communication, Control and Automation (ICCUBEA), 2017, pp. 1–5.

[14] P. Esmaili, P. Esmaili, F. Cavedo, and M. Norgia, "PSO-based auto-calibration for differential pressure level sensor," in 2021 International Conference on Artificial Intelligence of Things (ICAIoT), 2021, pp. 30–35.

[15] R. Perez et al., "Leak localization in water networks: A model-based methodology using pressure sensors applied to a real network in Barcelona [applications of control]," IEEE control syst., vol. 34, no. 4, pp. 24–36, 2014.

[16] M. A. Boillat, A. J. van der Wiel, A. C. Hoogerwerf, and N. F. de Rooij, "A differential pressure liquid flow sensor for flow regulation and dosing systems," Proceedings IEEE Micro Electro Mechanical Systems, 1995.

[17] Y. Hongfeng, L. Xingang, S. Hong, and L. Hong, "CFD simulation of orifice flow of orifice-type liquid distributor," China Pet. Process. Petrochem. Technol., vol. 15, no. 3, p. 70, 2013.

[18] J. F. Wendt, Ed., Computational fluid dynamics: An introduction, 3rd ed. Berlin, Germany: Springer, 2008.

[19] P. Esmaili, F. Cavedo, and M. Norgia, "Characterization of pressure sensor for liquid-level measurement in sloshing condition," IEEE Trans. Instrum. Meas., vol. 69, no. 7, pp. 4379–4386, 2020.

[20] Z. A. Dayev and A. K. Kairakbaev, "Modeling of coefficient of contraction of differential pressure flowmeters," Flow Meas. Instrum., vol. 66, pp. 128–131, 2019.

[21] Y. Guan and M. Saif, "A novel approach to the design of unknown input observers," IEEE Trans. Automat. Contr., vol. 36, no. 5, pp. 632–635, 1991.

[22] M. Hou and P. C. Muller, "Design of observers for linear systems with unknown inputs," IEEE Trans. Automat. Contr., vol. 37, no. 6, pp. 871–875, 1992.

[23] D. Luenberger, "Observers for multivariable systems," IEEE Transactions on Automatic Control, vol. 11(2), pp.190-197, 1966

[24] R. Szabolcsi, "Pole Placement Technique Applied in Unmanned Aerial Vehicles Automatic Flight Control Systems Design," Land Forces Academy Review, vol. 23, no. 1, pp. 88–98, Mar. 2018.

[25] C.-C. Tsui, "Observer design — A survey," International Journal of Automation and Computing, vol. 12, no. 1, pp. 50–61, Feb. 2015, doi: 10.1007/s11633-014-0865-7.

[26] W. Zheng, C. Wang, and D. Liu, "Data-driven based multi-objective combustion optimization covering static and dynamic states," Expert Syst. Appl., vol. 210, no. 118531, p. 118531, 2022.

[27] F. Jafarizadeh et al., "Data driven models to predict pore pressure using drilling and petrophysical data," Energy rep., vol. 8, pp. 6551–6562, 2022.

[28] T. M. Santhi and S. S., "Performance Enhanced Liquid Level Sensing System For Dynamic Environments," 2019 IEEE 5th International Conference for Convergence in Technology (I2CT), pp. 1-5, Mar. 2019.

[29] M. P. Schoen and J.-C. Lee, "Application of System Identification for Modeling the Dynamic Behavior of Axial Flow Compressor Dynamics," International Journal of Rotating Machinery, vol. 2017, pp. 1–14, 2017.

[30] M. R. Ananthasayanam, M. S. Mohan, N. Naik, and R. M. O. Gemson, "A heuristic reference recursive recipe for adaptively tuning the Kalman filter statistics part-1: formulation and simulation studies," Sādhanā, vol. 41, no. 12, pp. 1473–1490, Dec. 2016, doi:10.1007/s12046-016-0562-z.

[31] Y. Li and B. Hou, "Observer-based sliding mode synchronization for a class of fractional-order chaotic neural networks," Advances in Difference Equations, vol. 2018, no. 1, Apr. 2018.

[32] N. Oucief, M. Tadjine, and S. Labiod, "A new methodology for an adaptive state observer design for a class of nonlinear systems with unknown parameters in unmeasured state dynamics," Trans. Inst. Meas. Control, vol. 40, no. 4, pp. 1297–1308, 2018.

[33] Z. Wang, Y. Shen, and X. Zhang, "Actuator fault estimation for a class of nonlinear descriptor systems," Int. J. Syst. Sci., vol. 45, no. 3, pp. 487–496, 2014.

[34] M. Liu, L. Zhang, P. Shi, and Y. Zhao, "Fault estimation sliding-mode observer with digital communication constraints," IEEE Trans. Automat. Contr., vol. 63, no. 10, pp. 3434–3441, 2018.

[35] Y. Wang, V. Puig, and G. Cembrano, "Robust fault estimation based on zonotopic Kalman filter for discrete-time descriptor systems: Robust fault estimation based on zonotopic Kalman filter for discrete-time descriptor systems," Int. J. Robust Nonlinear Control, vol. 28, no. 16, pp. 5071–5086, 2018.

[36] W. Han, Z. Wang, Y. Shen, and Y. Liu, "Fault detection for linear discrete-time descriptor systems," IET Control Theory Appl., vol. 12, no. 15, pp. 2156–2163, 2018.

[37] A. Zemzemi, M. Kamel, A. Toumi, and M. Farza, "Robust integral-observer-based fault estimation for Lipschitz nonlinear systems with time-varying uncertainties," Trans. Inst. Meas. Control, vol. 41, no. 7, pp. 1965–1974, 2019.

[38] J. Schmidhuber, "Deep learning in neural networks: An overview," Neural Networks, vol. 61, pp. 85–117, Jan. 2015.

[39] C. Gershenson, "Artificial neural networks for beginners," arXiv preprint cs/0308031, 2003.

[40] K. Prudviraj, S. Deshmukh, R. K. Tripathy, K. Supradeepan, P. Tandon, and P. K. Jha, "Machine learning-based approach for the prediction of an orifice size of aerospace vehicle RCS thrusters during cold flow calibration," in 2021 IEEE 6th International Conference on Computing, Communication and Automation (ICCCA), 2021, pp. 455–459.

[41] M. Farsi et al., "Prediction of oil flow rate through orifice flow meters: Optimized machine-learning techniques," Measurement (Lond.), vol. 174, no. 108943, p. 108943, 2021.

[42] A. R. Behesht Abad et al., "Predicting oil flow rate through orifice plate with robust machine learning algorithms," Flow Meas. Instrum., vol. 81, no. 102047, p. 102047, 2021.

[43] C. Choi, J. Kim, H. Han, D. Han, and H. S. Kim, "Development of water level prediction models using machine learning in wetlands: A case study of Upo wetland in South Korea," Water (Basel), vol. 12, no. 1, p. 93, 2019.

[44] J. L. Mata-Machuca, R. Martínez-Guerra, and R. Aguilar-López, "An exponential polynomial observer for synchronization of chaotic systems," Communications in Nonlinear Science and Numerical Simulation, vol. 15, no. 12, pp. 4114–4130, Dec. 2010.

[45] L. Torres, G. Besançon, D. Georges, and C. Verde, "Exponential nonlinear observer for parametric identification and synchronization of chaotic systems," Mathematics and Computers in Simulation, vol. 82, no. 5, pp. 836–846, Jan. 2012.

[46] C. C. Nwobi-Okoye, S. Okiy, and A. C. Igboanugo, “Performance evaluation of multi-input–single-output (MISO) production process using transfer function and fuzzy logic: Case study of a brewery,” Ain Shams Engineering Journal, vol. 7, no. 3, pp. 1001–1010, Sep. 2016.

[47] M. L. Lineros, A. M. Luna, P. M. Ferreira, and A. E. Ruano, “Optimized design of neural networks for a river water level prediction system,” Sensors (Basel), vol. 21, no. 19, p. 6504, 2021.

[48] S. Beyhan, “Runge–Kutta model-based nonlinear observer for synchronization and control of chaotic systems,” ISA Transactions, vol. 52, no. 4, pp. 501–509, Jul. 2013.

[49] V. Shenoy and K. V. Santhosh, “Design Of Estimator For Level Monitoring Using Data Driven Model,” 2021 2nd International Conference on Computation, Automation and Knowledge Management (ICCAKM), Jan. 2021.

[50] X. Wang and E. E. Yaz, “Second-order fault tolerant extended Kalman filter for discrete time nonlinear systems,” IEEE Trans. Automat. Contr., vol. 64, no. 12, pp. 5086–5093, 2019.

[51] K. van Heusden, M. Yousefi, J. M. Ansermino, and G. A. Dumont, “Closed-loop MISO identification of propofol effect on blood pressure and depth of hypnosis,” IEEE Trans. Control Syst. Technol., vol. 28, no. 1, pp. 254–263, 2020.

[52] S. Rúa, R. E. Vásquez, N. Crasta, and C. A. Zuluaga, “Observability analysis and observer design for a nonlinear three-tank system: Theory and experiments,” Sensors (Basel), vol. 20, no. 23, p. 6738, 2020.

[53] S. M. N. Arshad, Y. Ayaz, S. Ali, A. R. Ansari, and R. Nawaz, “Experimental study on slosh dynamics estimation in a partially filled liquid container using a low-cost measurement system,” IEEE Sens. J., vol. 22, no. 16, pp. 16212–16222, 2022.

[54] T. J. Chung, Computational Fluid Dynamics, Cambridge University Press, 2002.

[55] V. Shenoy and K. V. Santhosh, “Characterization of orifice performance using Computational Fluid Dynamics,” 2021 IEEE Mysore Sub Section International Conference (MysuruCon), Oct. 2021.

[56] L. K. Bohra, L. M. Mincks, and S. Garimella, “Experimental Investigation of Pressure Drop Characteristics of Viscous Fluid Flow Through Small Diameter Orifices,” Journal of Fluids Engineering, vol. 143, no. 2, Oct. 2020.

[57] R. D. Grose, “Orifice flow at low Reynolds number,” J. Pipelines; (Netherlands), vol. 3, no. 3, 1983.

[58] R. D. Grose, “Orifice contraction coefficient for inviscid incompressible flow,” J. Fluids Eng., vol. 107, no. 1, pp. 36–43, 1985.

[59] K. Ramamurthi and K. Nandakumar, “Characteristics of flow through small sharp-edged cylindrical orifices,” Flow Meas. Instrum., vol. 10, no. 3, pp. 133–143, 1999.

[60] Y. Ding and L. Jiao, “Research on influnence of orifice parameteers on fluid resistance variations,” CSAA/IET International Conference on Aircraft Utility Systems (AUS 2020), 2021, vol. 2020, pp. 227–231.

[61] J. S. Bay, Fundamentals of linear state space systems. Maidenhead, England: Irwin Professional Publishing, 1998.

[62] S. M. Shinners, Modern control system theory and design, John Wiley & Sons, 1998.

[63] H. Sun, R. Madonski, S. Li, Y. Zhang, and W. Xue, “Composite control design for systems with uncertainties and noise using combined extended state observer and Kalman filter,” IEEE Trans. Ind. Electron., vol. 69, no. 4, pp. 4119–4128, 2022.

[64] M. S. Mahmoud, “Observer-based control design: Basics, progress, and outlook,” in New Trends in Observer-Based Control, Academic Press, pp. 143–208, 2019.

[65] G. Welch and G. Bishop, An introduction to the Kalman filter, 2006.

[66] Y. Zahraoui, M. Akherraz, and A. Ma’arif, “A comparative study of nonlinear control schemes for induction motor operation improvement,” International Journal of Robotics and Control Systems, vol. 2, no. 1, pp. 1–17, 2021.

[67] K. Tan, Q. Ji, L. Feng, and M. Torngren, “Shape estimation of a 3D printed soft sensor using multi-hypothesis extended Kalman filter,” IEEE Robot. Autom. Lett., vol. 7, no. 3, pp. 8383–8390, 2022.

[68] R. J. Meinhold and N. D. Singpurwalla, “Understanding the Kalman Filter,” Am. Stat., vol. 37, no. 2, pp. 123–127, 1983.

[69] P. S. Maybeck, “The Kalman filter: An introduction to concepts,” in Autonomous Robot Vehicles, New York, NY: Springer New York, 1990, pp. 194–204.

[70] A. K. Sahoo and S. K. Udgata, “A novel ANN-based adaptive ultrasonic measurement system for accurate water level monitoring,” IEEE Trans. Instrum. Meas., vol. 69, no. 6, pp. 3359–3369, 2020.

[71] J. M. M. Castillo, J. M. S. Cspedes, and H. E. Cuchango, “Water Level Prediction Using Artificial Neural Network Model,” J. Appl. Eng. Res, vol. 13, pp. 14378–14381, 2018.

[72] A. J. Abougarair, M. K. I. Aburakhis, and M. M. Edardar, “Adaptive neural networks based robust output feedback controllers for nonlinear systems,” International Journal of Robotics and Control Systems, vol. 2, no. 1, pp. 37–56, 2022.

[73] S. Blume, T. Benedens, and D. Schramm, “Hyperparameter optimization techniques for designing software sensors based on artificial neural networks,” Sensors (Basel), vol. 21, no. 24, p. 8435, 2021.

[74] G. I. Parisi, R. Kemker, J. L. Part, C. Kanan, and S. Wermter, “Continual lifelong learning with neural networks: A review,” Neural Netw., vol. 113, pp. 54–71, 2019.

[75] S. Ruder, “An overview of gradient descent optimization algorithms,” arXiv [cs.LG], 2016.

[76] G. Ellis, Control System Design Guide: using your computer to understand and diagnose feedback controllers, Butterworth-Heinemann, 2012.

[77] I. Hosseini, A. Khayatian, P. Karimaghaee, M. Fiacchini, and M. A. Davo Navarro, “LMI-based reset unknown input observer for state estimation of linear uncertain systems,” IET Control Theory Appl., vol. 13, no. 12, pp. 1872–1881, 2019.