Leakage Detection in Thermodynamic Systems

Karl Hansson

A thesis presented for the degree of Master of Science in Mechanical Engineering

Department of Energy Sciences, LTH
Lund University, Spring 2026

Supervisor: Thommie Nilsson
Industry Supervisor: Dennis Järnåsen & Sebastian Lindström
Examiner: Johan Revstedt

ISRN: LUTMDN/TMHP-26/5686-SE
ISSN: 0282-1990

Table of Contents

Acknowledgments

I would like to express my sincere gratitude to my supervisor Thommie Nilsson, for his valuable guidance and support throughout the thesis. Through meetings and insightful discussion, he helped me with guidance and encouraged me to keep going forward. I would also like to thank my supervisors Dennis Järnåsen and Sebastian Lindström at Hitachi Energy, for contributing valuable knowledge related to the subject and for their guidance throughout the study. Furthermore, I am grateful for your introduction to the automation team at Hitachi Energy.

Abstract

Detecting small leakages in thermodynamic cooling systems is a critical challenge in modern industrial applications, particularly in large-scale systems such as HVDC installations. Even minor leakages can reduce efficiency, increase energy consumption, and lead to higher maintenance costs. Detecting small leakages is difficult, because of complex system behavior, measurement noise, and natural variations in operating conditions. This thesis investigates methods for leakage detection and leakage rate estimation in thermodynamic systems while minimizing false alarms. The work is carried out in collaboration with Hitachi Energy, with the goal of developing a general and adaptable method suitable for industrial implementation in Siemens TIA Portal. The proposed approach is based on physical relationships between mass, density, and temperature, combined with filtering and signal processing techniques. Both difference-based and regression-based estimation methods are evaluated, together with the effect of filtering parameters and window lengths. The methods are first developed and validated in a simulation environment using Simcenter Amesim, and later tested on offline data from a real test system using Python. The results show that reliable detection and estimation of leakage rates is possible under clearly detectable conditions. A key finding is the trade-off between responsiveness and stability, where fast detection does not necessarily imply fast stabilization. The study demonstrates how different estimation methods and parameter settings influence detection performance and estimation accuracy.

1 Introduction

1.1 Background

Large-scale thermodynamic cooling and thermal management systems are a fundamental enabling component of modern industrial and energy systems, where their performance directly influences operational efficiency, system stability, and overall reliability [1]. Efficient operation in data centers is strongly dependent on advanced thermal management and cooling strategies aimed at reducing energy consumption and maintaining stable operating conditions [2]. Over the past few decades, the electronic technologies across multiple industries have undergone rapid development [2]. Because of the growth in data center capacity and associated energy demand, effective cooling systems have become increasingly critical to manage heat generation and maintain reliable and energy-efficient operation [2].

Detecting small leakages in thermodynamic systems is important. Even minor leakages can be crucial to find, leakages can significantly reduce system efficiency while increasing energy consumption, environmental impact, and maintenance cost, therefore effective fault diagnosis and leak detection methods are needed [3]. Detecting leakage in complex systems is challenging due to the intricate network behavior, diverse leakage mechanisms, and the difficulty of identifying and localizing leaks within large-scale infrastructures [4]. These factors can distort leakage signatures, increasing detection uncertainty and leading to inaccurate estimation of leakage rate [4]. These factors also makes it important to develop robust detection methods capable of minimizing false alarms while maintaining accurate leakage detection and leakage estimation [5]. This thesis investigates methods for leakage detection and estimation in large thermodynamic systems, with a focus on evaluating detection logic, estimation accuracy, and minimizing false alarms.

1.2 Industrial Context

Hitachi Energy is a global company that operates across multiple segments, including electric power transmission, grid control, and industrial cooling. They design the cooling systems after the customers criteria, cooling systems differ from one installation to another. This thesis will focus on HVDC (High Voltage Direct Current) stations and these may vary according to the customers needs. HVDC installations share several core functions, including power conversion through converter stations, voltage transformation via transformers, and operation of associated power-electronic components [6]. Although these systems operate according to similar principles, their physical layouts sensor configurations, and component placement may vary depending on customer specific requirements.

Hitachi Energy´s systems differ in design, geometry, and instrumentation, and therefore they want a general and adaptable leak detection approach rather than a system-specific solution. The developed method must be able to operate across a wide range of HVDC cooling system configurations and ultimately be implemented into the automation program that Hitachi Energy is using, which is Siemens TIA Portal. To ensure reliability and transferability, the method is first developed and validated in a simulation environment using Simcenter Amesim. An example of a system developed in Amesim is shown in Figure 1. After validating the method using Amesim, the method will be tested on offline data from a real test system and this will be done in Python. The code in Python will be written as the intended PLC deployment. This approach enables consistent translation from model development to practical industrial application, ensuring that leak detection can be deployed uniformly across future projects without requiring extensive system-specific customization.

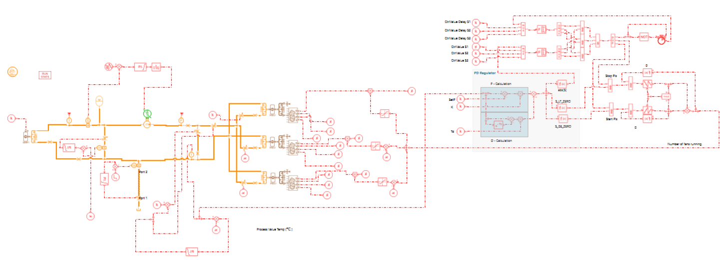

Figure 1: Full Amesim system model.

1.3 Problem Description

Thermodynamic cooling systems are large and complex. They consist of many components that are affecting each other, like pumps, piping networks, heat exchangers, valves, bypasses, and expansion vessels. Some of the interactions are through flow, mass, and temperature changes. This results in complex system behavior that is difficult to analyze [4]. Because of this complexity, it is not possible to isolate individual components and estimate their impact on the overall system performance. Simulations and modeling are required for finding how different signals in the system interact [4]. Another challenge is the operational variability. Changes in pump operating modes, valve positions, load-dependent heat generation, and ambient conditions will cause natural fluctuations in temperature, pressure, and flow signals. These changes in the system will impact the measured signals, that could change like it is looking like a leakage in the system. It could therefore be difficult to distinguish leakage behavior from normal system operation [4]. Time delays in the system are also important to consider when detecting leakages. Long pipelines and large fluid volumes can introduce delays due to thermal inertia and fluid transport time [7]. Changes in heat generation, flow rate, or operating conditions will not affect temperature and pressure immediately throughout the system. These delays can produce variations in the measured signals, making it more difficult to determine whether observed changes are caused by leakage or by normal system dynamics [4]. Small leakages are particularly difficult to detect, because they cause weak and slowly developing changes in measured signals over time. The smaller the leakage, the more similar these signal patterns become to normal operating conditions, making detection more challenging. To make it easier to identify small trends over time, leak detection should not rely entirely on raw sensor signals. Filtering and smoothing methods are required to reduce noise and to easier find longer trends [8]. Another challenge for small leakage detection is the trade-off between false alarm and the leakage rate that will be able to detect. The stricter the thresholds are set, the more sensitive the system becomes to small leakages, but this also increases the risk that normal operating variations will exceed these thresholds and trigger false alarms.

1.4 Research Objectives and Contributions

The purpose of this thesis is to evaluate methods for leakage detection and estimation in thermodynamic cooling systems. Hitachi Energy aims to improve their ability to detect smaller leakages compared to their current implementation. Therefore, the methods developed and evaluated in this work are intended to improve detection sensitivity while minimizing false alarms. This thesis also focuses on leakage size estimation, with the objective of estimating the leakage rate over time. The proposed method should be applicable to different HVDC cooling systems with varying sizes and configurations, while requiring minimal system-specific adjustments. The method should be relatively simple to enable implementation in Siemens TIA Portal PLCs. Based on the requirements, the thesis addresses several research questions related to leakage detection in thermodynamic cooling systems. A central question is whether leakage events can be detected reliably using available system signals including pressure, liquid level in the expansion vessel, and temperature measurements. The study evaluates different methods and approaches that can form the basis for detection logic. Another aspect of the work is to investigate whether simulations in Simcenter Amesim can be effectively used to evaluate methods intended for real-world implementation. A physics-based approach for leakage estimation is developed, based on relationships between mass, density, and temperature. Different filtering methods are evaluated to determine how noise can be reduced while maintaining accurate estimation. The final method is implemented in Siemens TIA Portal, with the goal of providing a practical solution that requires limited system-specific tuning. The developed methods are evaluated under available simulation and measurement conditions, providing insight into parameter selection, detection performance, and estimation behavior, and forming a basis for future work on detecting smaller leakage rates.

2 Theory

2.1 Thermodynamic Properties of the Working Fluid

The working fluid in the small-scale test system that the method is applied on is water, the fluid is operating within a temperature range of approximately 0 °C to 60 °C. In the full-scale HVDC cooling systems that is used, the working fluid is not water, it is water-glycol mixture. Both water and water-glycol mixtures can be treated as incompressible or weakly compressible liquids under moderate pressure conditions, as is commonly assumed for liquids in engineering thermodynamics [9]. Under normal operating conditions, both fluids remain in the liquid phase and can therefore be treated as incompressible liquids. Therefore, the density of the fluid can be approximated as primarily dependent on temperature, and the effect of pressure is not needed to be taken into account [9]. This enables a simplified thermodynamic description, where changes in fluid mass can be estimated from volume measurements, provided that temperature effects on density are properly accounted for.

2.2 Density-Temperature Relationship and Thermal Compensation

The density of liquids varies with temperature change due to thermal expansion and contraction. With increasing temperature the density reduces and decreasing temperature increases the density [9]. Accurately accounting for this effect in real systems is challenging, since the temperature is not homogeneous and is only measured at a limited number of discrete locations. This introduces uncertainty when estimating density changes. For liquids such as water and water-glycol mixtures, changes in fluid volume or level may occur solely due to temperature variations, even when the total mass remains constant [9]. In storage tanks and closed piping systems, such thermally induced volume changes can resemble leakage behavior if temperature effects are not properly accounted for. Although the specific density-temperature relationships differ between fluids, temperature-driven density variations must always be considered when interpreting volume measurements. Temperature compensation must be done to distinguish true mass loss from thermally induced volume variations. This temperature compensation approach provides a physically consistent basis for leakage detection across varying operating conditions and fluid compositions. It enables a unified analysis framework for both the experimental system and real-world systems, despite differences in temperature ranges and fluid properties. The resulting temperature-compensated volume serves as key observable for detecting small and gradually developing leakages. This relationship is consistent with the fundamental expression $m = \rho V$, which links mass, density, and volume in liquid systems [9]. Any change in volume, after temperature compensation, can be interpreted as a change in fluid mass.

2.3 Normal Operating Behavior and Baseline Modeling

Even in the absence of leakage, cooling and fluid systems exhibit variations in measured signals over time. Factors such as load changes, control actions, sensor characteristics, and environmental influences introduce fluctuations in measured signals such as pressure, level, and volume. Therefore, perfect constant signals can not be expected even under normal operating conditions. Even systems that are nominally identical may exhibit different signal patterns due to variations in operating conditions, sensor placement, calibration, and control strategies. This could lead to signal patterns that indicate leakage in one system may correspond to normal operation in another. This makes the use of fixed thresholds unreliable for leakage detection and motivates the use of system-specific references for distinguishing normal behavior from leakage. To address this challenge, leakage detection can be formulated relative to a baseline that represents typical system behavior under healthy operating conditions. The baseline defines the normal range and typical time-dependent behavior of measured signals during healthy operation, allowing deviations to be evaluated relative to what is expected for a specific system. With baseline learning, leakage is identified by sustained deviations from the established baseline behavior, and not by set values. By capturing typical variations and slow system dynamics during healthy operation, it becomes possible to distinguish between true leakage and normal operational fluctuations.

2.4 Stability Conditions for Baseline Identification

Some factors that can cause instabilities in cooling systems are when the system is starting up, system changes, or when a leakage is present. During these phases, the signals are not reliable because they may show trends that do not reflect normal operation. Limiting baseline identification to stable periods ensures that it reflects normal system behavior while avoiding transient and fault-related effects. This ensures that deviations from the baseline can be treated as possible signs of leakage rather than normal variation.

3 Methods

3.1 Method Overview

This section present the methodology used to apply and evaluate different leakage detection methods in HVDC cooling systems. The proposed methods are compared to identify the most effective approach for detecting leakages. The methods are based on physical principles, where leakage is defined as a persistent loss of mass over time. The leakage is estimated from changes in the system volume, which are influenced by operating conditions and measurement uncertainties. The method consists of several steps. The relevant variables that taken into account are level in the expansion vessel, temperature, and pressure. The system mass is estimated by first deriving the liquid volume from level measurements and then combining it with temperature-dependent density. The effect of thermal expansion is removed by temperature compensation. A reference density is defined at a given operating condition, linking the corresponding volume to a fixed reference state. Changes in density then lead to apparent changes in volume, which allows variations caused by temperature to be separated from actual mass loss. Signal filtering techniques are used to reduce measurement noise and extract meaningful trends from the data. Two filtering approaches are considered, exponential moving average (EMA), using different smoothing factors, and a Kalman filter. These filters provide different trade-offs between responsiveness and robustness. Leakage detection is performed using a rule-based approach, where a consistent decrease in mass is combined with supporting trends in level and pressure signals. The detection logic is based on a baseline established during stable operating conditions, where reference values for mass, level, and pressure are learned. From this baseline, the corresponding thresholds are defined and used to identify deviations from normal system behavior. To ensure reliable detection and reduce false alarms, deviations from the baseline indicating mass loss, must occur together with consistent supporting trends in level and pressure signals over multiple consecutive time steps. By deriving the detection thresholds from the baseline, the method adapts to normal system variations and improves robustness. To improve reliability and reduce false alarms more, an alarm is triggered first after consecutive samples have indicated leakage. When a leakage is detected and the alarm is triggered, the leakage rate is estimated using two different approaches. The first method is linear regression, where the slope of the filtered volume signal over a moving time window represents the rate of volume loss. The second method is a difference-based regression where two values are compared. The leakage rate is estimated by comparing the current value with the value at the beginning of the window. The difference between these two values is then divided by the number of samples within the window to get the average rate of change, representing the leakage rate. The performance of the different filtering and estimation methods are evaluated using both simulation and offline data. Simulation data are generated in Siemens Amesim, where controlled leakage scenarios can be created. Measured data from the physical test system are used to evaluate performance under real operating conditions. The method are implemented in Python for offline analysis, replicating the logic intended for deployment in Siemens TIA Portal.

3.2 Test System and Simulation Environment

A simplified system model is developed in Siemens Simcenter Amesim to represent the essential behavior of the cooling system while still keeping it simple and reduce unnecessary complexity. This simplification allows controlled analysis of leakage effects without the influence of unnecessary system details [10]. The model is used to simulate controlled leakage scenarios and to validate the proposed leakage detection methods. This enables systematic evaluation of detection performance under well-defined conditions. The simulation model represents a closed cooling system that includes an expansion vessel, pump, heat source, and fans for cooling. The structure and sensor placement of the simulation model is to look as much as the testing system as possible. This makes the detection logic easier to validate. An overview of the full system layout including thermal, hydraulic, electrical, and auxiliary subsystems is shown in Figure 2. A zoomed-in view of the expansion vessel and the controlled leakage implementation is shown in Figure 3.

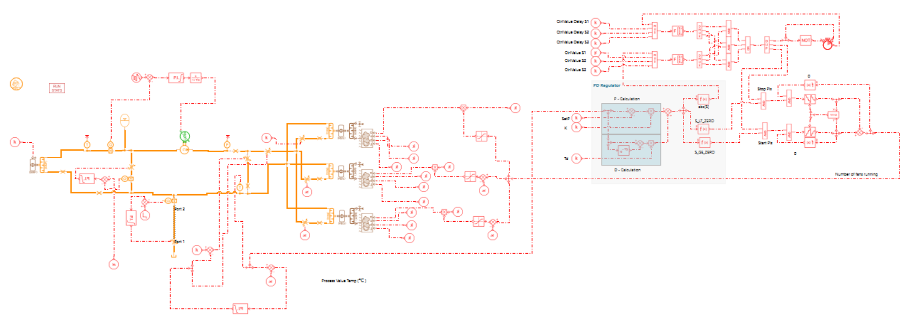

Figure 2: Overall Siemens Simcenter Amesim model of the closed cooling system used for leakage detection evaluation.

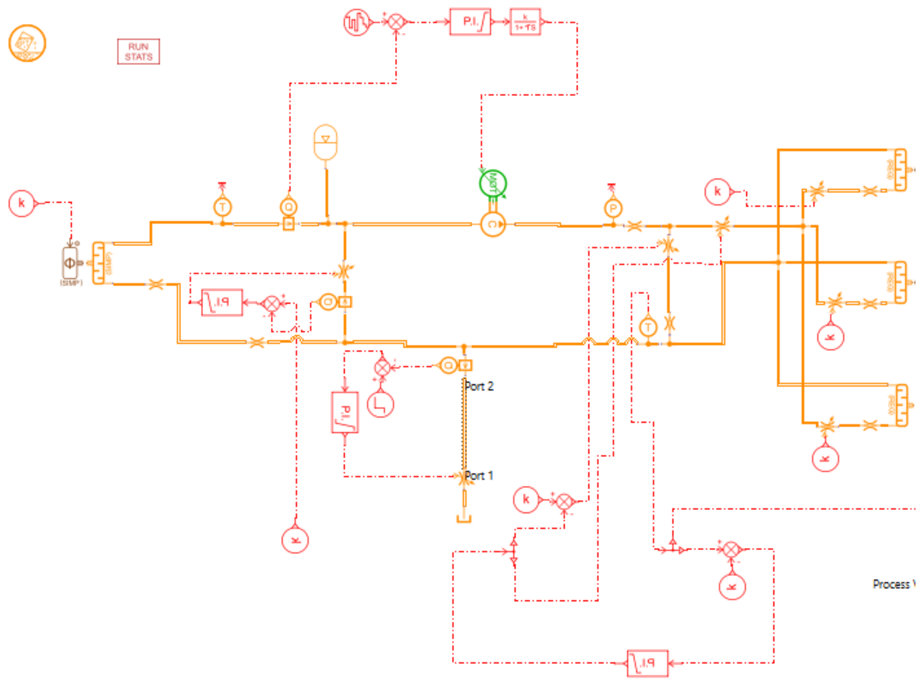

Figure 3: Zoomed view of the Siemens Simcenter Amesim model focusing on the expansion vessel and monitored variables used for leakage detection.

3.3 Leakage Scenarios and Simulation Setup

To evaluate the leakage detection methods under controlled and repeatable conditions, a set of leakage scenarios are simulated in Siemens Simcenter Amesim. The simulation model has a fixed sampling rate of 1 Hz, to make it consistent with the real system, and each simulation run has a total duration of 4400s. The simulations in Amesim have a start-up time of 800s, this is the time before the system is stable. Therefore, the first 800s of each simulation are excluded from the analysis. The leakage detection should only be tested under stable conditions. Leakage scenarios are generated by varying both the leakage rate and the leakage starting time. The simulated leakage rates range from 1 to 5L/min, in order to evaluate the detection capability of different leakage rates. The leakage starting times are 2400s and 3600s into the simulation. The reason for this is to evaluate two different scenarios to see that the results are consistent but also to evaluate the influence of baseline learning duration. Before the leakage starts, each simulation contains a period of leak-free operation during stable conditions. This period is used for baseline learning. Leakages can not be stopped within a single Amesim simulation, therefore is each leakage case implemented as a separate simulation in which the leakage persists until the end. Because the leakage can not be stopped, the alarm clearing behavior can not be analyzed. The analysis will therefore focus on detection performance, consistency, and alarm stability.

The offline data from the real test system is sampled at 1 Hz and the simulation length is approximately 3900s. In the offline evaluation, the leakage starting times are 600s, 1200s, 2400s, and 3600s. This setup enables evaluation of detection delay under varying baseline conditions, but also the alarm clearing behavior following the end of a leakage event. Having many leakage events in a single dataset makes it possible to evaluate both the consistency of leakage detection and the ability of the method to return to normal operation after the leakage has been stopped.

3.4 Measurement Signals and Data Acquisition

Signals available in both the simulation environment and the test system are used, to make the leakage method more consistent. There are some differences in signal collection. In Amesim, a single temperature signal is available and in the test system, three different temperature signals are measured. The temperature signal in the test system are analyzed to determine whether a weighted temperature representation is the best option or a single temperature signal, such as the one closest to the expansion vessel. Another difference concerns the volume in the expansion vessel. In Amesim, the fluid volume is obtained by subtracting the gas volume in the expansion vessel from the total expansion vessel volume. In the test system it is a little different, the volume is derived from the measured liquid level and temperature in the tank. In the simulation environment, the signals are extracted from model components, while in the test system they are obtained from sensors and recorded for offline analysis.

3.5 Temperature Selection for Density Compensation

In systems, such as test system, where multiple temperature measurements are available, selecting the correct representative temperature for density-based compensation is not straightforward. A weighted temperature fusion approach is therefore considered, in which a single effective temperature is computed as a combination of available measurements. Similar weighted temperature representations have been used in thermal system modeling to improve the estimation representative temperature [11]. To determine the weighting factors, a data-driven approach is used based on regression analysis of normal operation data. The available temperature measurements are used as explanatory variables, while variations in the expansion vessel level are used as the response variable. The relationship between the variables is modeled using a linear regression formulation expressed as

$$\Delta h = a\,\Delta T_{\mathrm{EV}} + b\,\Delta T_{\mathrm{Hot}} + c\,\Delta T_{\mathrm{Cold}},$$

where Δh represents the deviation of the expansion vessel level from its mean value, and ΔT_{EV}, ΔT_{Hot}, and ΔT_{Cold} represent the corresponding deviations.

Prior to performing the regression, the mean values of both the expansion vessel level and the temperature signals are removed for the model only to capture variations relative to normal operating conditions. This allows the relationship between temperature and level variations to be analyzed independently of steady-state. The weighting factors from the regression represents the influence each temperature have on the level in the expansion vessel. The regression coefficients are normalized to get relative weighting factors defined as

$$W_{\text{EV}} = \frac{|a|}{|a| + |b| + |c|}, \quad W_{\text{Hot}} = \frac{|b|}{|a| + |b| + |c|}, \quad W_{\text{Cold}} = \frac{|c|}{|a| + |b| + |c|},$$

where a, b, and c are the regression coefficients associated with the expansion vessel, heater outlet, and cooler outlet temperature measurements.

Regression analysis assesses the relationship between variables and explain it statistically [12]. The problem with this approach is that the weighting factors may be influenced by system-level dynamics and does not take the thermodynamic behavior into account. A more simple and maybe accurate approach is also considered, where the temperature measurement located closest to the expansion vessel is used because it´s the one describing the level in the expansion vessel the best.

3.6 Signal Conditioning and Smoothing Methods

Signals obtained from the cooling system are affected by fluctuations caused by sensor noise, measurement uncertainty, and operational variations. These disturbances will affect baseline estimation, and accuracy of both leakage detection and leakage rate estimation. To reduce noise and improve signal stability, smoothing methods are applied to the temperature-compensated volume signal prior to further analysis. Three different approaches are considered and these are, an exponential moving average (EMA) implemented with different smoothing factors, and a Kalman filter. The EMA-based approaches represent different levels of smoothing, where a higher smoothing factor provides faster response with less noise suppression, while a lower smoothing factor results in a smoother but slower signal. The Kalman filter, is a more advanced and adaptive method. These approaches differ in terms of complexity and filtering behavior. Stronger smoothing can improve stability and remove noise better, but at the cost of longer detection delay [13]. The effectiveness of each approach is evaluated based on its performance in leakage detection, detection delay, leakage rate accuracy, and the time required for the estimated leakage rate to stabilize. This enables assessment of both how quickly and reliably a leakage is detected, as well as how accurate and consistent the leakage rate is estimated. All filtering methods are applied under identical conditions to ensure a better comparison. The focus of this study is not detailed parameter optimization for each method, but rather to investigate how variations in smoothing parameters influence estimation accuracy and responsiveness. This allows for a comparison of the relative performance of the different filtering methods in the context of leakage detection.

Exponential Moving Average

The exponential moving average (EMA) is a simple and efficient smoothing method commonly used in time-series analysis and signal processing to reduce noise in measured data [13]. It reduces short-term fluctuations while still preserving long-term trends, which makes it suitable for slowly varying quantities such as leakage [13]. EMA is defined as

$$y(k) = (1 - \alpha)\, y(k - 1) + \alpha\, x(k),$$

where $x(k)$ represents the input signal, $y(k)$ the filtered output, and $\alpha \in (0, 1]$ is the smoothing factor. The value of α determines the balance between past values and the current measurement [13]. Larger values of α result in a more responsive filter that reacts quickly to changes in the signal, while smaller values provide stronger smoothing but slower response. EMA is evaluated using different smoothing factors to investigate how the level of smoothing influences leakage detection performance and leakage size estimation. Two representative values, $\alpha = 0.02$ and $\alpha = 0.05$, are selected to compare the effect of slower and faster filtering on detection delay, leakage rate estimation accuracy, and stabilization time. These representative configurations are further evaluated in combination with different window lengths for both difference-based and regression-based leakage estimation methods to analyze their impact on estimation performance. In addition, a wider range of smoothing factors, $\alpha = 0.01, 0.02, 0.03, 0.05, 0.10,$ and 0.20, is evaluated to further analyze how the smoothing parameter affects the estimated leakage rate and the time required for the estimated leakage rate to converge to the correct leakage value.

Kalman filter

Kalman filter is a common approach in signal processing and estimation theory for estimating the state of a system from noisy measurements [14]. In contrast to the exponential moving average, the Kalman filter combines a model-based prediction with incoming measurements to estimate the underlying signal [14]. This allows the filter to balance noise reduction and responsiveness adaptively. The Kalman filter is described as

$$x(k) = A\, x(k - 1) + w(k), \qquad z(k) = H\, x(k) + v(k),$$

where $x(k)$ represents the true system state, $z(k)$ is the measured signal, and $w(k)$ and $v(k)$ represent process noise and measurement noise, respectively [14]. The process noise $w(k)$ models uncertainties in the system dynamics, while the measurement noise $v(k)$ represents noise affecting the observations. $z(k)$ corresponds to the measured volume signal, while $x(k)$ represents the filtered estimate of the true volume. The estimated state $x(k)$ corresponds to the filtered signal, which was previously denoted as $y(k)$. The filter updates the estimate at each time step by first predicting the next value based on the previous estimate, and then correcting this prediction using the new measurement [14]. The relative influence of the prediction and the measurement is determined by the process noise covariance Q and the measurement noise covariance R. The process noise covariance Q reflects the expected variability of the underlying signal and controls how quickly the estimate is allowed to adapt to changes. The measurement noise covariance R represents the uncertainty in the measured signal and determines how strongly the estimate is influenced by new measurements. R is selected based on the observed noise level in the measurement data. The value of Q is chosen based on the expected physical behavior of the system, the smallest rate of interest which is set to 1L/min in this case. This leakage rate is converted into an equivalent volume change per time step to reflect how quickly the signal is expected to vary. To evaluate the sensitivity of the filter, the chosen value of Q is scaled by factors of 0.5, 1.0, and 2.0. A simplified model is used with $A = 1$ and $H = 1$m which is a common choice for slowly varying signals, where the state is assumed to change gradually over time.

3.7 Baseline Learning and Stability Criteria

A reference describing normal system behavior is required for reliable leakage detection and leakage size estimation. This reference is obtained from baseline learning, where the typical values and variation of the system signals are determined during stable conditions. Stability is evaluated using indicators based on the mass change, level drift, and pressure variation in the expansion vessel. These signals are filtered to reduce short-term fluctuations and highlight persistent trends. The system is considered stable when these indicators are within acceptable limits and also if there is no warning or alarm. The stability limits are determined using percentile-based analysis of data collected during normal system operation. Different percentile levels are considered to define thresholds that represent acceptable range of variation for each indicator. Lower percentile levels correspond to stricter stability conditions, while higher percentile levels classify more data as stable. If all the stability conditions are satisfied, the corresponding data is added to a rolling baseline with a fixed length of 180 samples.

3.8 Leakage Detection Logic

Leakage detection is based on identifying persistent deviations in the mass change signal relative to the learned baseline. The detection logic consists of two main conditions that must be satisfied simultaneously. One condition based on the mass change, and one directional condition based on level and pressure signals. When a sufficient number of baseline samples is available, statistical measures are computed from the baseline data. The baseline consists of the mass change, level drift, and pressure drift signals, which are treated independently. For the mass change signal, the baseline values are first sorted, and the median is extracted as

$$Dm_{\mathrm{Med}} = P_{50},$$

where P_{50} is the 50th percentile of the baseline data. The median represents the typical or expected mass change under normal operating conditions. The natural variability of the signal is then quantified using percentile-based measures. The 10th and the 90th percentile define the range of normal variation observed during stable operation. This allows the expected fluctuations in the signal to be characterized. The variability is then estimated as

$$Dm_{\mathrm{Sig}} = \frac{P_{90} - P_{10}}{2.56},$$

providing a robust measure of the signal variation that is less sensitive to outliers. This enables the detection thresholds to account for normal fluctuations and distinguish them from abnormal deviations. For the level and pressure signals, thresholds are obtained directly from the lower tail of the baseline distributions. Specifically, the 5th percentile is used to define the thresholds

$$dLevel_{\mathrm{Thr}} = P_{Level,5}, \quad dPEV_{\mathrm{Thr}} = P_{Pressure,5}$$

which represent the lower limit of normal signal behavior. If the number of available baseline samples is too small, fixed fallback values are used to define the thresholds.

Based on these baseline-derived quantities, two thresholds are defined for the mass change signal. One threshold is for warning and one threshold is for alarm and they are defined like this

$$\mathrm{Thr}_{\mathrm{Warn}} = Dm_{\mathrm{Med}} - K_{\mathrm{Warn}} \cdot Dm_{\mathrm{Sig}}, \quad \mathrm{Thr}_{\mathrm{Alarm}} = Dm_{\mathrm{Med}} - K_{\mathrm{Alarm}} \cdot Dm_{\mathrm{Sig}},$$

where $K_{\mathrm{Warn}} = 2.0$ and $K_{\mathrm{Alarm}} = 2.8$ are scaling factors that determine the sensitivity of the detection.

The detection logic consists of two conditions that must be satisfied simultaneously. One is a threshold condition based on the mass change signal and the other is a directional condition based on the level and pressure signals. The threshold condition is fulfilled when the smoothed mass change signal falls below the corresponding warning or alarm threshold. The directional condition is satisfied when either the level or the pressure signal falls below its respective threshold. To ensure robust detection, a persistence requirement is introduced. The detection conditions must be satisfied for a number of consecutive time steps before a warning or alarm is triggered. This is implemented using counters that increase when both the threshold condition and the directional condition are satisfied, and reset otherwise. A warning is triggered when the number of consecutive detections exceeds 10. The alarm requires a longer sequence of consecutive detections. The reference configuration for the alarm condition is

30 consecutive samples. Shorter lengths of 20 and 10 samples are also evaluated to analyze the influence of this parameter on detection delay and detection stability.

3.9 Leakage Size Estimation Method

Leakage size is estimated continuously based on the system signals and the temperature-compensated liquid volume. The estimated leakage size is only considered valid when a leakage alarm is active. During periods without detected leakage, the estimated value is set to 0.0 L/min. The estimation is based on the corrected liquid volume, where density variations are compensated using a reference density. Multiple filtered versions of the volume signal are used, based on the previously described smoothing methods. These filtered signals are evaluated to analyze their influence on estimation accuracy, stability, and responsiveness. Two estimation approaches are implemented in parallel, a difference-based estimation and a regression-based estimation.

The difference-based estimators compute the leakage rate from the change over a set time window. The volume in the system now is compared to the volume in the system that was in the first point of the set window. $V_f(t)$ is the filtered volume signal and the leakage rate is estimated as

$$L(t) = -\frac{V_f(t) - V_f(t - \Delta t)}{\Delta t},$$

where Δt is the selected window length. Regression-based estimators are also used, where a linear model is fitted to the filtered volume signal over a set time window and is defined as

$$V_f(t) \approx at + b,$$

where a is the slope of the fitted line. The leakage size is obtained from this slope, which represents the rate of change of the volume over the selected window size. Different window lengths are evaluated for both estimation approaches to analyze the trade-off between responsiveness and stability. Shorter windows provide faster response to changes but are more sensitive to noise, while longer windows produce smoother estimates at the cost of increased delay. Leakage rate estimation is evaluated by comparing the estimated leakage size to a known reference value that is calculated from system measurements. In the evaluation of the methods, the things considered are estimation accuracy, relative error, and stability time. The stability time is the time from activation of the leakage alarm until the estimated leakage rate remains within the set tolerance for a continuous time.

4 Results

4.1 Overview of Results and Evaluation Scope

This section presents the results obtained by applying an offline Python implementation.

Baseline Stability Threshold Configuration

The baseline learning and stability detection are governed by threshold values from percentile-based analysis of normal system operation data. To evaluate the influence of threshold selection, multiple percentile levels are considered, 80th, 90th, 95th, and 99th percentiles. Each percentile configuration defines a corresponding set of stability thresholds for the selected system indicators, which determine whether the system is operating under stable conditions and therefore the baseline is learning. The evaluated stability thresholds for each percentile level are summarized in Table 1.

Table 1: Baseline stability thresholds for different percentile levels

Percentile	$\lvert Dm_{Smooth} \rvert$	$dLevel_{EMA}$	$dPEV_{EMA}$
80th	$< 1.9986 \times 10^{-2}$	$\geq -9.6469 \times 10^{-4}$	$\geq -2.8650 \times 10^{-3}$
90th	$< 2.6644 \times 10^{-2}$	$\geq -1.4858 \times 10^{-3}$	$\geq -4.6708 \times 10^{-3}$
95th	$< 3.4959 \times 10^{-2}$	$\geq -2.0182 \times 10^{-3}$	$\geq -6.5209 \times 10^{-3}$
99th	$< 4.4959 \times 10^{-2}$	$\geq -2.8895 \times 10^{-3}$	$\geq -1.1235 \times 10^{-2}$

Lower percentile thresholds give stricter stability conditions, while higher percentile thresholds allow greater signal variation. The 95th percentile configuration is selected as the baseline setting for the standard evaluation, while other percentile levels are additionally evaluated to analyze the sensitivity of the method to threshold selection.

Weighted Temperature Estimation

In systems where multiple temperature measurements are available, a weighted temperature representation can be used as a single effective temperature for density compensation. The weighted factors are set using regression analysis of normal operation data. The resulting normalized weighted factors are

$$W_{EV} = 0.212, \quad W_{Hot} = 0.415, \quad W_{Cold} = 0.373,$$

where the weights represent the relative influence of the expansion vessel, heater outlet, and cooler outlet temperature signals.

It is observed that the regression-based weighting factors are influenced by system dynamics and may not fully reflect the physical relationship between temperature and volume in the expansion vessel. The temperature closest to the expansion vessel should have the highest impact and therefore this regression based weighted temperature is not used. instead a simpler and more physically motivated approach is considered, where only temperature measurement closest to the expansion vessel is used. Based on this reasoning, the final implementation uses only the expansion vessel temperature, corresponding to

$$W_{EV} = 1.0, \quad W_{Hot} = 0.0, \quad W_{Cold} = 0.0.$$

This simplification reduces model complexity while maintaining physically meaningful temperature compensation.

Kalman Filter Parameter Estimation

The Kalman filter parameters are estimated based on the statistical properties of the measured signals and the expected physical behavior of the system. The measurement noise covariance is estimated directly from normal operation data. To isolate measurement noise, slow variations in the signal are removed using EMA, and the remaining fluctuations are interpreted as measurement noise. The standard deviation of the noise is estimated as

$$\sigma = 0.024 \text{ L},$$

resulting in a measurement noise covariance of

$$R = 5.52 \times 10^{-4}.$$

The process noise covariance is determined based on the smallest leakage rate of interest. A target leakage rate of 1.0 L/min is converted to a volume change per time step, and the corresponding variance is used to define the process noise. This results in a nominal process noise covariance of

$$Q = 2.78 \times 10^{-4}.$$

4.2 Detection Capability in Siemens Amesim

This subsection presents the leakage detection results from Siemens Amesim simulations using the proposed PLC-equivalent algorithm. Leakage rates of 1-5 L/min were simulated in a closed hydraulic system with a total fluid volume of approximately 3 m^3. All simulations were evaluated using the same baseline learning, stability criteria, and alarm logic. The purpose of this evaluation is to analyze the functionality and performance of the leakage detection method. The evaluation first verifies that the underlying logic is working, which means that the baseline learning is working correctly, that no false alarm occur during normal operation, and that leakage events reliably trigger alarms. The evaluation will also determine the smallest possible leakage rate can be consistently detected under the given conditions. The detection delay is also analyzed and detection stability, that the alarm remains active during leakage events. For leakage cases that are more reliable, an evaluation of their leakage size estimation is made. The evaluation include the estimation accuracy and stabilization time. These factors are compared across different filtering approaches, exponential smoothing with different parameters and Kalman filtering.

Detection Results for Different Leakage Sizes

Tables 2 and 3 summarize the leakage detection results for leakage start times of 2400 s and 3600 s, respectively.

Table 2: Leakage detection results for Amesim simulations with leakage start time at 2400 s.

Leakage rate [L/min]	Alarm triggered	Detection stability	Detection delay [s]
1	No	Did not stabilize	-
2	Yes	Intermittent	39
3	Yes	Intermittent	37
4	Yes	Intermittent	36
5	Yes	Stable	35

Table 3: Leakage detection results for Amesim simulations with leakage start time at 3600 s.

Leakage rate [L/min]	Alarm triggered	Detection stability	Detection delay [s]
1	No	Did not stabilize	-
2	Yes	Intermittent	108
3	Yes	Intermittent	104
4	Yes	Intermittent	34
5	Yes	Stable	34

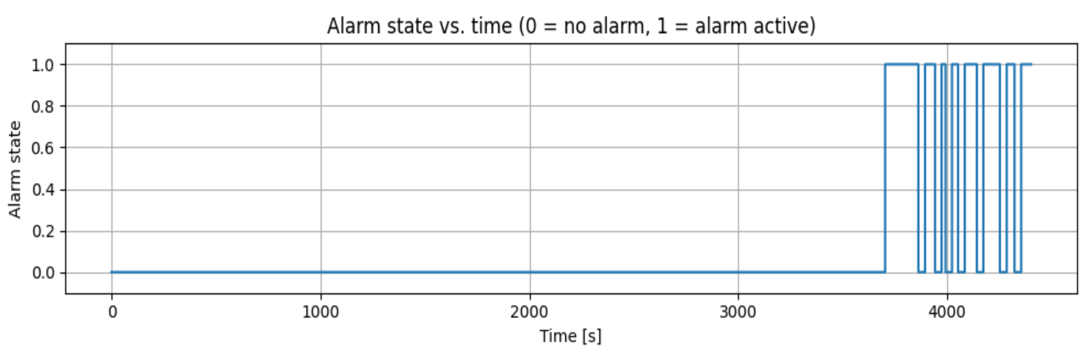

Figure 4: This is an example of intermittent leakage detection. The leakage is constant but the alarm goes between alarm (1) and no alarm (0).

The results show that leakage is successfully detected for leakage rates of 2-5 L/min, while the smallest leakage rate of 1 L/min is not detected. The alarm is intermittent for leakage rates between 2 L/min and 4 L/min. This indicates that the signal is close to the thresholds and is influenced by noise and fluctuations. Stable and continuous detection is achieved when the leakage rate is 5 L/min, here the leakage signal is clearly exceeding the defined thresholds. The quickest detection delays are 34 s, which means that the detection was found quickly and that the consecutive alarm requirement of 30 samples is making the detection delay longer. There were no false alarms prior to the leakage starting time, indicating that the stability criteria and detection thresholds effectively prevent false leakage indications during normal operation.

Leakage Size Estimation for Robustly Detected Leaks

Leakage rate estimation is performed for leakage rates of 4 L/min and 5 L/min, as these represent the lowest leakage levels for which detection is reliable. The 4 L/min case is included even though the alarm is intermittent. It is included in the analysis to evaluate the estimator performance under both partially stable and fully stable conditions. For both leakage rates, the leakage rate is estimated using different-based estimators and regression-based estimator using exponential moving average with different smoothing parameters and Kalman filtering. The stabilization time for the different methods are evaluated, stabilization is defined as the time it takes for the estimated leakage time to remain within $\pm 10\%$ of the true leakage size for a continuous duration of at least 30 s.

Tables 4 and 5 present the leakage size estimation results for leakage start times of 2400 s and 3600 s, respectively. Tables 4 and 5 present the leakage size estimation results for leakage start times of 2400 s and 3600 s, respectively.

Table 4: Leakage size estimation results for robustly detected leaks with leakage start time at 2400 s.

Leak [L/min]	**Estimator**	**Stabil. time [s]**	**Mean [L/min]**	**Error [%]**
4	EMA ($\alpha = 0.05$)	75	4.19	4.8
4	EMA ($\alpha = 0.02$)	134	4.17	4.2
4	Kalman	77	4.19	4.7
4	EMA ($\alpha = 0.05$, reg.)	160	4.17	4.1
4	EMA ($\alpha = 0.02$, reg.)	232	4.17	4.1
4	Kalman (reg.)	161	4.16	4.1
5	EMA ($\alpha = 0.05$)	75	5.22	4.3
5	EMA ($\alpha = 0.02$)	133	5.21	4.1
5	Kalman	77	5.21	4.3
5	EMA ($\alpha = 0.05$, reg.)	159	5.22	4.4
5	EMA ($\alpha = 0.02$, reg.)	203	5.21	4.2
5	Kalman (reg.)	161	5.22	4.4

Table 5: Leakage size estimation results for robustly detected leaks with leakage start time at 3600 s.

Leak [L/min]	**Estimator**	**Stabil. time [s]**	**Mean [L/min]**	**Error [%]**
4	EMA ($\alpha = 0.05$)	82	4.14	3.4
4	EMA ($\alpha = 0.02$)	139	4.13	3.2
4	Kalman	82	4.14	3.4
4	EMA ($\alpha = 0.05$, reg.)	157	4.16	4.0
4	EMA ($\alpha = 0.02$, reg.)	204	4.14	3.4
4	Kalman (reg.)	158	4.16	4.0
5	EMA ($\alpha = 0.05$)	68	5.19	3.9
5	EMA ($\alpha = 0.02$)	137	5.18	3.6
5	Kalman	70	5.19	3.9
5	EMA ($\alpha = 0.05$, reg.)	157	5.20	4.0
5	EMA ($\alpha = 0.02$, reg.)	203	5.17	3.4
5	Kalman (reg.)	159	5.20	4.1

All estimation methods converge toward the true leakage rate with estimation errors below 5%. This indicates that the leakage rate estimation approach works for Amesim simulations. The estimations are all over predicting the leakage. This systematic over prediction can be explained by a mismatch between the fluid density used in the estimation algorithm and the actual fluid used in Amesim. In Amesim, a glycol-water mixture is used, while in Python a water density lookup table is used. The temperature in Amesim is around 30°C which from the table in Python gives

$$\rho_{\text{table}} = 995.65 \text{ kg/m}^3.$$

In the Amesim simulation data, the mass is 3136.85 kg and the volume is 3005.86 L, which gives the fluid density of

$$\rho_{\text{AMESim}} = \frac{3136.85}{3005.86} \approx 1.0435 \text{ kg/L} = 1043.5 \text{ kg/m}^3.$$

This results in a density difference of

$$\frac{1043.5 - 995.65}{995.65} \approx 4.8\%.$$

The leakage rate is calculated based on the rate of change of volume, where the volume is obtained from the mass using an assumed density. This difference directly introduces a scaling error in the estimated leakage. The relationship between the densities leads to

$$\frac{\rho_{\text{AMESim}}}{\rho_{\text{table}}} \approx \frac{1043.5}{995.65} \approx 1.048.$$

Applying this factor to the nominal leakage value gives

$$4.0 \times 1.048 = 4.19 \text{ L/min},$$

which closely matches the estimated leakage that ranges between 4.13 and 4.19 L/min. For the leakage rate of 5.0 L/min, the same factor results in

$$5.0 \times 1.048 = 5.24 \text{ L/min},$$

which closely matches the estimated leakage that ranges between 5.17 and 5.22 L/min. This demonstrates that the over prediction is caused by the density mismatch and not by the estimation method itself.

4.3 Offline Evaluation Using Real System Data

This section presents the offline evaluation of the proposed leakage detection method and leakage size estimation method that is implemented in SIEMENS TIA Portal. The data that is used for this evaluation and testing is from a reduced cooling system. Due to limited access to full-scale HVDC cooling installations and controlled leakage experiments, offline data analysis is very important for evaluating the practical behavior of the method under realistic operating conditions.

Baseline Stability Thresholds and Learning Activity

The stability conditions for baseline learning follow a percentile-based threshold configuration, as described in Table 1 in Section 4.1.1. The influence of different threshold settings on detection and estimation performance is evaluated in this section. Figure 4 illustrates the behavior of baseline learning and alarm activity for the 95th percentile configuration. The figure shows when baseline learning is active, the time periods during which leakage is present, and the corresponding alarm signal over time, where a value of 1 indicates an active alarm and 0 indicates no alarm.

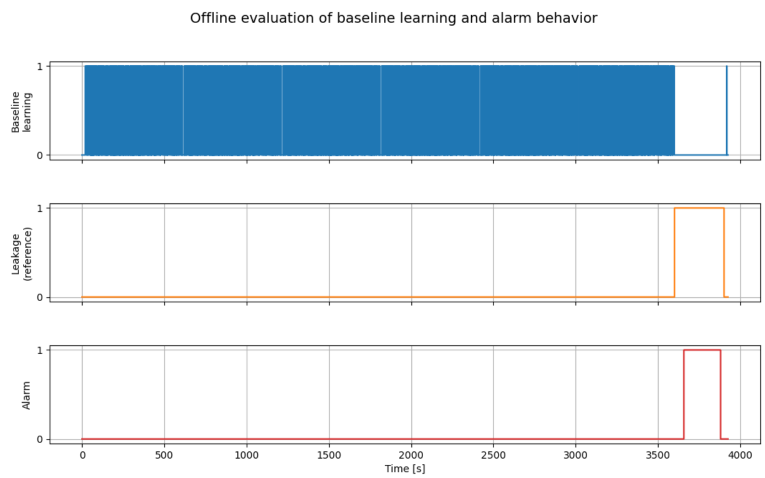

Figure 5: Baseline learning and alarm activity for the 95th percentile configuration.

The figure shows that baseline learning is active only during normal operation. The baseline signal may appear as continuously active due to small fluctuations, as it rapidly switches on and off during stable conditions. This creates a visual effect where the signal looks filled. During leakage and alarm conditions it can be clearly observed that baseline learning is inactive, confirming that no learning takes place during these periods. The alarm signal also behaves as expected, as it is only activated when leakage is present.

Table 6 summarizes the amount of baseline learning obtained for different percentile thresholds. The column "Total" represents the percentage baseline learning relative to the entire simulation time, including both normal operation and leakage periods. The column "Normal" represents the percentage baseline learning relative to normal operation time.

Table 6: Baseline learning activity for different percentile thresholds.

Percentile	**Learning** [s]	**Total** [%]	**Normal** [%]
80th	862	22.0	23.9
90th	1350	34.4	37.5
95th	1837	46.8	51.0
99th	2561	65.3	71.1

The results show that the amount of baseline learning increases with higher percentile levels. For lower percentiles, fewer time intervals are classified as stable, resulting in limited baseline updates. As the percentile increases, more signal variation is accepted as stable, leading to a larger amount of baseline learning over time. The difference between the "Total" and "Normal" arises because baseline learning is inactive during leakage periods.

Leakage Detection Time

Leakage detection time is the time between when the leakage is starting and the first sustained activation of the alarm signal. This metric is evaluated for different percentile configurations and leakage start conditions.

Table 7 presents the detection delay for a leakage event occurring at 3601 s, where a significant amount of baseline learning is available before the leakage starts.

Table 7: Leakage detection time for different percentile configurations.

Percentile	**Leakage onset** [s]	**Detection delay** [s]
80th	3601	56
90th	3601	57
95th	3601	58
99th	3601	58

The results show that the detection delay is very similar across all percentile configurations, ranging from approximately 56 s to 58 s. Lower percentile values lead to slightly faster detection. Even though the variation under these conditions are minimal and only weakly influenced by the percentile, the result still shows that the system is behaving correctly. Lower stability thresholds lead to more conservative baseline learning, where higher quality and more representative baseline data is saved. This is expected to contribute to faster and more reliable detection performance. Under less stable operating conditions, this effect is expected to become more significant, as stricter baseline selection helps to prevent disturbed or non-representative data from being included in the baseline.

Independent Volume-Based Validation

To provide an independent physical reference for the leakage size estimation, the leaked fluid from the experiment used to generate the offline dataset was collected in a bucket. In the experiment, the leakage was active for approximately 247 s, corresponding to the interval from 23 s to 270 s in the offline leakage data file. The total collected volume was 181 L, resulting in an estimated leakage rate of 43.96 L/min. It should be noted that the leakage signal in the offline dataset is constructed by combining a normal operation dataset with a separate leakage dataset. The system therefore directly transitions into a leakage condition when the leakage dataset is introduced. The system considers the total leakage duration to be 270 s, and this duration is used when evaluating the clearing time from leakage to no-leakage conditions. The leakage estimation results obtained from the model are evaluated in a later section, where the estimations are compared to the experimentally determined reference value in order to get the accuracy of the estimation approach.

Leakage Detection and Alarm Clearing Behavior

Table 8 presents the detection delay and alarm clearing behavior for multiple leakage events introduced at 600 s, 1200 s, and 2400 s.

Table 8: Leakage detection and alarm clearing delays for different percentile configurations.

Percentile	Leak start [s]	Detection delay [s]	Clearing delay [s]	Detection type
80th	600	57	11	Stable
80th	1200	57	11	Stable
80th	2400	58	12	Stable
90th	600	58	11	Stable
90th	1200	58	11	Stable
90th	2400	59	12	Stable
95th	600	59	11	Stable
95th	1200	59	11	Stable
95th	2400	60	12	Stable
99th	600	59	11	Stable
99th	1200	59	11	Stable
99th	2400	60	12	Stable

Effect of Consecutive Alarm Duration on Detection and Clearing Performance

Different consecutive alarm durations, 10 s and 20 s, are evaluated to analyze how the detection delay and clearing delay are affected for different percentile configurations and leakage start times.

Table 9: Detection and clearing delays for a consecutive alarm duration of 10 s.

Percentile	Leak start [s]	Detection delay [s]	Clearing delay [s]	Detection type
80th	600	11	11	Intermittent
80th	1200	10	11	Intermittent
80th	2400	38	12	Stable
90th	600	38	11	Stable
90th	1200	11	11	Stable
90th	2400	10	12	Stable
95th	600	11	11	Intermittent
95th	1200	11	11	Intermittent
95th	2400	40	12	Stable
99th	600	11	11	Intermittent
99th	1200	11	11	Intermittent
99th	2400	10	12	Intermittent

The results show the detection and clearing delays when a consecutive alarm duration of 10 s is required. A shorter consecutive duration results in faster detection, but may lead to intermittent alarm behavior.

Table 10: Detection and clearing delays for a consecutive alarm duration of 20 s.

Percentile	Leak start [s]	Detection delay [s]	Clearing delay [s]	Detection type
80th	600	47	11	Stable
80th	1200	47	11	Stable
80th	2400	48	12	Stable
90th	600	47	11	Stable
90th	1200	47	11	Stable
90th	2400	48	12	Stable
95th	600	49	11	Stable
95th	1200	49	11	Stable
95th	2400	50	12	Stable
99th	600	49	11	Stable
99th	1200	49	11	Stable
99th	2400	50	12	Stable

The detection delay increases slightly for the 20 consecutive alarm samples in comparison to 10 consecutive alarm samples. The detection is stable though for all percentiles and leak starting times which is not the case for the 10 consecutive alarm samples, resulting in a a more reliable alarm triggering.

Leakage Size Estimation After Detection

After leakage is detected, the leakage size is estimated using multiple independent estimators based on different filtering and regression approaches. The regression methods are difference-based regression and linear regression. The filtering is used with exponential moving average with different smoothing factors and a Kalman filter. The leakage size estimation runs continuously, but the results are only considered when the leakage alarm is active. Table 11 summarizes the estimated leakage rates. The estimated values represent the average leakage size from the point where the estimate has stabilized after detection until the leakage event ends.

Table 11: Comparison of leakage size estimates with independent volumetric reference (43.96 L/min).

Estimator	Estimated leakage size [L/min]	Difference [L/min]	Difference [%]
EMA ($\alpha = 0.05$)	44.25	0.29	0.66
EMA ($\alpha = 0.02$)	41.49	-2.47	-5.62
Kalman	44.22	0.26	0.59
EMA ($\alpha = 0.05$, reg.)	44.92	0.96	2.18
EMA ($\alpha = 0.02$, reg.)	42.17	-1.79	-4.07
Kalman (regression)	44.89	0.93	2.12

The estimators show leakage sizes close to the reference value of 43.96 L/min. The estimation error is below 6% for all estimators. The EMA estimator with a lower smoothing factor tends to underestimate the leakage, while the estimator with higher smoothing shows values closer to the reference. The difference-based regression show the best results overall.

Leakage Size Estimation Stability Time

In this section the stability time of the leakage size estimators is analyzed. The stability time is evaluated relative to the measured volumetric reference 43.96 L/min. The stability time is defined as the time from when the leakage alarm is activated until the estimated leakage size remains within ±5% of the reference value for a continuous duration of 30 s. An estimator is therefore considered stabilized when its leakage size estimate satisfies

$$|L_i(t) - 43.96| \leq 0.05 \times 43.96$$

for at least 30 s.

Table 12: Leakage rate estimation stability times

Estimator	Stability time [s]
EMA ($\alpha = 0.05$)	78
EMA ($\alpha = 0.02$)	120
Kalman	80
EMA ($\alpha = 0.05$, reg.)	149
EMA ($\alpha = 0.02$, reg.)	Did not stabilize
Kalman (regression)	150

4.4 Parameter Sensitivity Evaluation

This section investigates how different window lengths and filtering parameters affect the estimation accuracy and stability time.

Effect of Sliding Window Length

The sliding window length is the time interval over which volume differences are evaluated for the difference-based leakage size estimators.

Effect of Regression Window Length

The regression window length determines the number of samples used for the regression-based leakage size estimation.

Table 13: Effect of sliding window length (*LeakWinLen*) on leakage size estimation performance for difference-based estimators.

LeakWinLen [s]	Estimator	Estimated leakage [L/min]	Error vs 43.96 [%]	Stability time [s]
30	EMA ($\alpha = 0.05$)	44.51	1.25	155
30	EMA ($\alpha = 0.02$)	42.43	-3.48	101
30	Kalman	44.51	1.25	68
60	EMA ($\alpha = 0.05$)	44.25	0.66	78
60	EMA ($\alpha = 0.02$)	41.49	-5.62	120
60	Kalman	44.22	0.59	80
90	EMA ($\alpha = 0.05$)	43.46	-1.14	94
90	EMA ($\alpha = 0.02$)	40.08	-8.83	139
90	Kalman	43.38	-1.32	95
180	EMA ($\alpha = 0.05$)	38.25	-12.99	160
180	EMA ($\alpha = 0.02$)	33.35	-24.13	Not stabilized
180	Kalman	38.07	-13.40	162

Table 14: Effect of regression window length (*RegWinLen*) on leakage size estimation performance for regression-based estimators.

RegWinLen [s]	Estimator	Estimated leakage [L/min]	Error vs 43.96 [%]	Stability time [s]
30	EMA ($\alpha = 0.05$, reg.)	42.65	-2.98	158
30	EMA ($\alpha = 0.02$, reg.)	43.31	-1.48	99
30	Kalman (regression)	42.72	-2.82	160
60	EMA ($\alpha = 0.05$, reg.)	43.40	-1.27	79
60	EMA ($\alpha = 0.02$, reg.)	43.75	-0.48	117
60	Kalman (regression)	43.49	-1.07	80
90	EMA ($\alpha = 0.05$, reg.)	44.46	1.14	93
90	EMA ($\alpha = 0.02$, reg.)	44.07	0.25	135
90	Kalman (regression)	44.54	1.32	94
180	EMA ($\alpha = 0.05$, reg.)	44.92	2.18	149
180	EMA ($\alpha = 0.02$, reg.)	42.17	-4.07	Not stabilized
180	Kalman (regression)	44.89	2.12	150

Effect of Filtering Parameters

The filtering parameters control how much of the old values and how much of new values that is used. For exponential moving average (EMA), the smoothing factor determines this balance. A higher value uses more of the new data, which leads to faster response. A lower value uses more of old data, which leads to stronger smoothing but slower response.

Table 15: Effect of EMA smoothing factor (*EMA_Alpha*) on leakage size estimation performance. The sliding and regression window lengths are fixed at 60 s.

EMA_Alpha	Estimator	Estimated leakage [L/min]	Error vs 43.96 [%]	Stability time [s]
0.01	EMA	34.74	-20.99	Did not stabilize
0.01	EMA (regression)	39.79	-9.48	Did not stabilize
0.02	EMA	41.49	-5.62	120
0.02	EMA (regression)	43.75	-0.48	117
0.03	EMA	43.31	-1.48	95
0.03	EMA (regression)	43.92	-0.09	95
0.05	EMA	44.25	0.66	78
0.05	EMA (regression)	43.40	-1.27	79
0.10	EMA	44.57	1.39	65
0.10	EMA (regression)	42.73	-2.82	63
0.20	EMA	44.59	1.43	59
0.20	EMA (regression)	42.41	-3.53	55

The results show that lower smoothing factor leads to slow convergence and large estimation errors. The estimate adapts too slowly to the signal changes. With higher smoothing factor, the estimator adapts faster to changes in the signal, which leads to faster convergence and therefore also shorter stabilization time. The result is not always better for higher smoothing factors, for regression-based estimators, larger α values systematically underestimate the leakage rate.

Table 16: Effect of Kalman filter process noise (*KF_Q*) on leakage size estimation performance. The measurement noise is fixed, and both window lengths are 60 s.

KF_Q	Estimator	Estimated leakage [L/min]	Error vs 43.96 [%]	Stability time [s]
0.5×0.000278	Kalman	44.62	1.50	61
0.5×0.000278	Kalman (regression)	42.54	-3.23	57
1.0×0.000278	Kalman	44.61	1.48	60
1.0×0.000278	Kalman (regression)	42.50	-3.32	57
2.0×0.000278	Kalman	44.61	1.48	60
2.0×0.000278	Kalman (regression)	42.47	-3.39	56

The results show that varying the Kalman filter process noise has minor effect on the estimated leakage and stabilization time. The leakage estimates are almost unchanged, this is because of small changes in the process noise parameter. This results in that the Kalman filter provides stable and consistent performance within the evaluated parameter range.

5 Discussion

5.1 Baseline Stability Threshold Selection

The results show that increasing the percentile threshold leads to the system being classified as stable more frequently, resulting in a higher amount of baseline learning. This trend is clearly illustrated in Table 6, where the total learning time increases from 22.0% at the 80th percentile to 65.3% at the 99th percentile. The results are from a test system that is operating under relatively stable conditions. Even if it is operating under stable conditions, the baseline learning remains limited. This can be explained by two main factors. First, the stability thresholds are derived from the same test system, which results in very strict thresholds, since only very small deviations from this reference behavior are accepted as stable. In a full-scale system, larger deviations would occur during normal operation, which would require less restrictive thresholds. This would likely lead to a similar level of baseline learning when applied under comparable conditions. Second, all stability conditions must be satisfied simultaneously for the system to be classified as stable. This combined requirement further limits the amount of baseline learning, as even small deviations in a single condition can prevent the system from updating the baseline.

5.2 Leakage Detection Performance

Simulation Results (Amesim)

The results show that the detection performance in Amesim simulations depends on the leakage rate. The smallest leakage rate of 1 L/min is not detected. For leakage rates between 2 L/min and 4 L/min, the detection is intermittent, while stable and continuous detection is only achieved for the 5 L/min case. The explanation for this is that larger leakage rates produce a more significant deviation from the baseline, resulting in a clearer separation from the detection thresholds. For leakage starting at 2400 s, the detection delay ranges from 35 s for the largest leakage to 39 s for the smaller leakages. Larger leakage rates lead to faster detection, as the system variables deviate more quickly from the baseline conditions. The detection delay closely corresponds to the requirement of 30 consecutive alarm samples, which shows that the leakage is identified almost immediately, and the delay is primarily caused by the persistence requirement before the alarm is triggered. For leakage starting at 3600 s, there is a much larger variation in detection delay. For 4-5 L/min the detection is 34 s, which is fast, but for 2-3 L/min the detection delay is over 100 s. The explanation for this is that the mass in the expansion vessel is not constant during normal operation, there are fluctuations caused by system dynamics. Since the baseline statistics are calculated using a sliding window of 180 s, the estimated baseline mean depends directly on the specific behavior of the signal within this window. Therefore, the baseline mean can vary depending on the state of the system immediately before the leakage onset. If the mass in the expansion vessel is slightly higher within the baseline window, the calculated median value increases. This leads to higher threshold values, which are set for leakage detection. If a leakage occurs under these conditions, the difference between the detection signal and the threshold is reduced, causing the signal to operate closer to the threshold. This makes the alarm condition more sensitive to small fluctuations in the signal. Because the alarm requires a number of consecutive samples to confirm leakage, these small fluctuations can occasionally violate the detection condition and reset the counter. More time and larger accumulated leakage are required before the signal remains consistently beyond the threshold, leading to an increased detection delay. This effect is not dependent on the leakage starting time, but rather on the state of the system within the baseline window prior to leakage. If similar baseline conditions had occurred before an earlier leakage event, a comparable increase in detection delay could have been observed. This result could have changed with larger baseline windows. A larger baseline window would include more samples and therefore provide a more representative estimate of the true baseline behavior. This would reduce the sensitivity to fluctuations, resulting in a more stable baseline mean and more consistent detection performance across different leakage start times.

Offline Real System Results

The detection delay for the offline test system remains very consistent across all percentile configurations. For a leakage event starting at 3601 s, the detection delay ranges between 56 s and 58 s. In simulations with multiple leakage start times, the detection delay varies between 57 s and 60 s, which is comparable to the single leakage scenario. This indicates that the detection performance is stable and repeatable under varying conditions. The detection is relatively fast across all leakage cases. The alarm is triggered after approximately 60 s, but the requirement of 30 consecutive samples indicates that the leakage was identified within 30 s. The alarm clearing was also evaluated. The results show that the clearing logic operates reliably and consistently across different leakage start times and percentile configurations. The system is able to quickly return to a non-alarm state once the leakage condition is removed.

Effect of Consecutive Alarm Duration

The consecutive alarm duration has a significant influence on both detection delay and alarm stability. When the alarm is triggered after 30 consecutive samples, the detection delay ranges between 56-60 s. Reducing the requirement to 10 consecutive samples results in much faster detection, with delays as low as 10-11 s in several cases. This indicates that the leakage is identified quickly, but the alarm is triggered only after the required number of consecutive samples is satisfied. Faster alarm triggering reduces stability, particularly at the onset of leakage. With a shorter confirmation period, only a brief duration of threshold violation is required to trigger the alarm. At the beginning of a leakage event, the detection signal is still close to the threshold, making it sensitive to small fluctuations. These fluctuations can temporarily bring the signal back above the threshold, interrupting the detection condition and causing the alarm to go off. As the leakage continues, the accumulated effect becomes larger, increasing the deviation between the signal and the threshold. This results in a more stable detection, where the alarm condition is continuously satisfied. Therefore, the instability is primarily observed at the onset of leakage. To prevent this intermittent alarm state more consecutive alarm samples is a good option. Increasing the requirement to 20 consecutive samples improves the stability of the detection. The alarm becomes consistent across all tested configurations. The trade-off is an increased detection delay, ranging between 47 s and 60 s. Consecutive alarm samples is a good option, but by observing the intermittent alarm state for the real test system but especially the Amesim simulations there could be an additional improvement for the detection logic. In the smaller leakage rates in Amesim that are operating close to the detection thresholds, the signal may temporarily exceed the threshold for very short periods, sometimes only a single sample. This results in a complete reset of the consecutive alarm counter, even though the overall trend of the signal still indicated leakage. By requiring a number of consecutive samples indicating no leakage before resetting the alarm condition, the detection logic would become more robust to fluctuations. This would make the alarm logic much more stable, even for smaller leakage rates.

5.3 Leakage Size Estimation Performance

Simulation Results (Amesim)

The leakage rate estimation performance is evaluated for leakage rates of 4 L/min and 5 L/min. These leakage rates are selected because the alarm signal is stable for 5 L/min, while the 4 L/min case exhibits only minor fluctuations in detection. The results in Tables 4 and 5 show that the leakage rate estimation is stable for both leakage rates. The estimation error ranges between 3.2% and 4.8%, indicating good estimation of the true leakage. There is a small systematic overestimation across all methods, which as described in the results is due to density mismatch between the Amesim fluid and density used in Python. There is no significant difference in estimation accuracy between the two leakage start times, which indicates that the estimation method provides consistent results and is not that influenced by the system state at the time of the leakage onset. The reason for this is because the simulation in Amesim is very stable. The difference in estimation accuracy between difference-based and regression-based estimators are not very big. The choice of filtering method also has a limited impact. While the estimation accuracy is similar across methods, the stabilization time required to reach a steady leakage estimate differs. The stabilization time is defined as the time required to remain within $\pm 10\%$ of the true leakage for 30 consecutive samples. The shortest stabilization time is observed for difference-based estimators using EMA ($\alpha = 0.05$) and Kalman filtering, which ranges between 70-80 s. The regression-based estimators require longer stabilization times, between 150 s and 200 s. The reason for this is that for regression-based window sizes of 180 samples are used, and for difference-based estimators only a sliding window of 60 samples are used. The longer window provides stronger smoothing but delays the response, resulting in slower stabilization.

Offline Real System Results

The leakage rate estimation performance for the real system is evaluated by comparing the estimated leakage values to an independent volumetric reference of 43.96 L/min, as shown in Table 11. The results show that all estimators provide reliable leakage rate estimates. The estimation error remains below 6% for all methods, indicating that the approach is capable of producing accurate leakage estimates under real system conditions. For the evaluated filtering methods, the EMA ($\alpha = 0.05$) and Kalman estimators provide the most accurate results, with errors below 1%. The EMA with a lower smoothing factor ($\alpha = 0.02$) shows weaker performance, underestimating the leakage size with 4-6%. The underestimation is not caused by the limitation of the estimation method, but by the slow response of the filter combined with the finite length of the estimation window. EMA ($\alpha = 0.02$) will eventually converge to the correct leakage rate, the estimation window still contains samples from the initial transient phase where the signal has not yet fully responded to the leakage. The early underestimated values influence the estimated leakage rate, which leads to an underestimation until the entire window is filled with fully converged values. Both the difference-based and regression-based estimators show good results in leakage rate estimation, indicating that both methods are capable of capturing the leakage rate accurately. The main difference between the methods is the time required for the estimates to stabilize, the difference-based estimators achieve faster stabilization compared to the regression-based estimators. The reason for this is primarily due to the window lengths used in the estimation methods. The difference-based estimators use a shorter sliding window of 60 samples, while the regression-based estimators use a longer window of 180 samples. The regression-based methods therefore require larger number of samples.

5.4 Effect of Window Length Parameters

Impact on Estimated Leakage and Error

The effect of window length on leakage size estimation accuracy is evaluated for both difference-based and regression-based estimators, as shown in Tables 13 and 14. The results indicate that the estimated leakage remains relatively consistent across most of the window lengths, only showing bad results for a few window lengths. For the difference-based estimators, window lengths between 30 s and 90 s provide accurate results, with estimation errors limited to a few percent of the reference value. For the largest window length of 180 s, a significant underestimation is observed. The effect is seen across all estimators, with errors exceeding 12% for EMA ($\alpha = 0.05$) and Kalman, and even larger deviations for EMA ($\alpha = 0.02$). The reason for this result is primarily caused by the relationship between the window length and the leakage duration. In the difference-based estimator, the leakage rate is calculated from the difference between two values separated by the window length. Larger windows are generally better for capturing small leakages over longer periods but in this case the leakage duration is too short compared to the window length. The estimation therefore includes samples from the early transient phase following leakage onset, where the signal has not yet stabilized. Therefore, the first value in the window is too small in comparison to the latest value, which results in an underestimation. When applying the EMA ($\alpha = 0.02$) filter to difference-based estimator, the underestimation increases with increasing window lengths. The reason for this is the slow response of the filter where the first value of the filter is a value that has still not reached the estimated leakage rate, and therefore the error gets bigger with time. The regression-based estimators show a more constant result with smaller errors across all window lengths. The reason for this is because it estimates the leakage from the overall slope of the signal within the window, instead of using two points in the window.

Impact on Stability Time

The window length impacts the stabilization time for both difference-based and regression-based estimators, as shown in Tables 13 and 14. The estimation accuracy remains consistent for shorter and intermediate window lengths, but the stabilization time varies much with window size. Shorter window lengths for difference-based estimators usually give a quicker stabilization time, because the estimator can respond faster to changes in the signal. This trend is consistent across all filtering methods except for EMA ($\alpha = 0.05$), which shows slower stabilization time for window length of 30 s then for window length of 60 s. The reason for this is the convergence characteristics of the leakage estimation. The 30 s window reaches values in the tolerance band quickly, but the signal has not yet fully stabilized, and the estimate continues to evolve and also leaves the tolerance band. This interrupts the stability condition, causing the consecutive counter to reset and delaying the final stabilization time. The 60 s window responds slower and reaches the tolerance band at a later stage, but at that time the signal is more stable and it remains within the tolerance band continuously. This stability condition is never interrupted, so even if it has a slower response it still stabilizes quicker. Similar behavior is shown for regression-based estimations for 30 s windows where the stabilization time is longer than for 60 s windows , for EMA ($\alpha = 0.05$) and Kalman filter. The reason for the longer stabilization time in this case is because of larger estimation errors. The estimation error is larger and therefore closer to the tolerance boundary, small fluctuations and longer time outside of the tolerance boundary make the stability time longer. For 60 s window, the error is smaller and the estimate remains well within the tolerance band. In general, the best results in terms of stabilization time are when using intermediate window lengths. Short windows are too sensitive to changes and variations. The longer windows in this case respond more slowly to changes in the signal, resulting in delayed convergence and longer stabilization times. In the present simulations, the correct window length would therefore be 60s windows or 90s windows. Longer windows may perform better in scenarios with longer leakage durations, where sufficient time is available for the estimator to fully convergence and stabilize.

5.5 Effect of Filtering Parameters

The results show that the filtering parameters have an impact on the leakage estimation and the stabilization time. For the estimations using EMA, it is easy to see how the smoothing factor directly controls how quick the estimate adapts to changes in the signal. Higher smoothing factor results in faster response, which leads to faster stabilization time. Lower smoothing factor results in slower convergence, and also higher stabilization time. Higher smoothing factor leads to faster convergence, but it does not always improve the accuracy of the estimate. In particular for regression-based estimators, where larger α values lead to a systematic underestimation of the leakage rate. The reason for this is because at the end of the regression window the signal changes faster, which reduces the linearity. This results in a smaller estimated slope and an underestimation of the leakage rate. For low smoothing factors, the estimator reacts too slow to the leakage. This slow reaction leads to that the estimate does not always reach the correct leakage rate before the simulated leakage stops. This leads to large errors and with the α value 0.01 the underestimation is that large that it never stabilizes around the correct leakage rate. This indicates that excessive smoothing can delay convergence significantly and reduce the practical usefulness of the estimator. The Kalman filter shows consistent performance across the tested range of process noise values. The reason for this is that the change in parameter values is not very big and therefore leads to no significant difference in result. The results show that the choice of filtering parameters is important. Higher smoothing factors reduce stabilization time but may introduce systematic errors, especially for regression-based estimators. Lower smoothing factors are more suitable for detecting slow leakages with longer duration. Higher smoothing factors are more suitable for rapid detection. Therefore, the filtering parameters should be selected based on the specific requirements of the application.

5.6 Improvements

There are improvements that would be interesting to test after analyzing the results. The most interesting improvement after analyzing the results is the alarm logic. The alarm is triggered after consecutive samples indicate leakage, but the reset of the alarm only requires one single sample to indicate no leakage. This could lead to temporary fluctuations and cause the alarm to reset and after that start to build up consecutive samples that indicate leakage before the alarm is triggered again. By introducing a similar requirement for consecutive non-leakage samples before resetting the alarm, the stability of the detection logic can be significantly improved. Another improvement could be to test a wider range of Kalman filter parameters. Only a limited range is tested in this thesis, it would be interesting to see how more different values would affect the result to easier analyze the results.

The most interesting improvement in the work is to test the logic on a more complex system. The current evaluation is based on a test system that is relatively simple. Testing the estimation methods on a full-scale system would provide a much better insight into their performance. A more representative system would allow for the evaluation of additional effects such as disturbances, varying operating conditions, and longer leakage scenarios, leading to a more comprehensive validation of the approach. It would also have been interesting to test smaller leakage rates. In this thesis, only large leakage rates are tested due to limitations in the available data. Evaluating the methods on smaller leakage rates over time would be a great improvement. By testing smaller and slowly developing leakage rates over longer time periods, would provide a better understanding of the sensitivity and robustness of the estimation methods. Such an analysis would clarify how smaller leakage rates affect the measured signals and the detectability of the leakage. This would make it possible to better analyze the alarm logic and the estimation accuracy under more challenging conditions.

6 Conclusion

From a practical implementation perspective, the EMA-based smoothing would be the best choice. It is simple and it is also easy to tune. The smoothing factor can easily be adjusted depending on the application requirements, allowing the estimator to be configured for either fast detection or more accurate long-term estimation. Higher smoothing factors are recommended when faster detection of larger leakages is required, as they provide faster response and shorter stabilization time. For the purpose of identifying smaller leakages over longer time periods, lower smoothing factors should be used. The choice between difference-based and regression-based estimation methods also depends on the application. The methods differ in how the information within the window is used. Difference-based methods estimate the leakage rate based on the change between two values, which makes it simple to implement into the system. The regression-based method uses all samples within the window to estimate the slope, which generally leads to a more stable and robust estimate. It looks at the overall trend over time which reduces the influence of individual sample variations and provides a smoother representation of the leakage rate. A combined approach of the methods could be a good approach. Implementing a difference-based method with shorter window length that could be used for faster leakage rate estimation and also implementing a regression-based method with larger window length that could more accurately estimate smaller leakage rates over time. This allows the leakage rate estimation to be both fast and accurate over time. The window length determines the trade-off between responsiveness and stability, where smaller windows allow faster detection and larger windows improve detection of small leakage rates over time. Therefore a combination of different window sizes could improve the result as discussed above, but not just between different methods but also within the same logic. The use of simulation tools such as Amesim is a great tool for initial development and validation of the proposed method. Simulations in Amesim are good for testing and tuning the logic, ensuring that the method functions correctly before it is implemented in a real system. This is great, because there is not always access to test systems. Test systems in the production are a good next step for creating simulation data but validation on full-scale systems is necessary to confirm performance under realistic operating conditions, where additional factors such as disturbances, varying loads, and measurement noise influence the results.

7 Appendix

```python
import pandas as pd
import numpy as np
import os

base_dir = r"C:\Users\KH9687\AmesimData\MachineLearning\ExamensarbeteMachineLearning.py"
file_name = "TiaPortalTest.xlsx"
input_path = os.path.join(base_dir, file_name)

if not os.path.exists(input_path):
    raise FileNotFoundError(f"File not found: {input_path}")

df = pd.read_excel(input_path)

V_Total_m3 = 4.7135
DM_Smooth_N = 20
Consecutive_Warning = 10
Consecutive_Alarm = 30
K_Warn = 2.0
K_Alarm = 2.8
W_EV = 1.0
Tau_T_s = 10.0
EMA_Alpha = 0.05
LPF_Alpha = 0.02
KF_Q = 0.000278
KF_R = 0.001
KF_P_est = 1.0
LeakWinLen = 60
RegWinLen = 30
derAlpha = 0.3
BaselineWinLen = 180

BaselineDm = np.zeros(BaselineWinLen)
BaselineDLevel = np.zeros(BaselineWinLen)
BaselineDPEV = np.zeros(BaselineWinLen)

BaselineIndex = 0
BaselineCount = 0

DensityTbl = np.array([
    999.84,999.90,999.94,999.97,999.97,999.97,999.88,999.80,999.72,999.65,
    999.70,999.50,999.30,999.10,998.90,998.70,998.50,998.30,998.10,997.90,
    998.21,997.90,997.58,997.27,996.96,997.05,996.60,996.15,995.90,995.78,
    995.65,995.30,994.95,994.60,994.30,994.00,993.50,993.00,992.60,992.40,
    992.22,991.90,991.60,991.30,991.00,990.20,989.60,989.00,988.60,988.30,
    988.07,987.50,986.90,986.30,985.90,985.70,984.90,984.20,983.70,983.40,
    983.20
])

LastMass = 0.0
LastLevel = 0.0
LastPEV = 0.0

DmBuf = np.zeros(DM_Smooth_N)
DmSum = 0.0
DmIdx = 0
DmCount = 0

dLevel_Ema = 0.0
dPEV_Ema = 0.0


V_EMA = 0.0
V_LPF = 0.0
V_Kal = 0.0
KFV_P = KF_P_est
KFV_X = 0.0

LeakRefInit = False

CWarn = 0
CAlarm = 0
Warning = False
Alarm = False

BufV_EMA = np.zeros(LeakWinLen)
BufV_LPF = np.zeros(LeakWinLen)
BufV_Kal = np.zeros(LeakWinLen)
BufIdx = 0

VReg_EMA = np.zeros(RegWinLen)
VReg_LPF = np.zeros(RegWinLen)
VReg_Kal = np.zeros(RegWinLen)

out = []

Tmix = 0.0

```

```
for i in range(len(df)):

    LevelPercent = df.loc[i, "LogBL_Exp"]
    T_EV = df.loc[i, "LogBT_Inlet"]
    T_Hot = df.loc[i, "LogBT_Outlet"]
    T_Cold = df.loc[i, "LogBT_Coolers"]
    P_EV_kPa = df.loc[i, "LogBP_Exp"]

    Traw = W_EV * T_EV
    Alpha_T = 1.0 / Tau_T_s
    Tmix = Traw if i == 0 else (1 - Alpha_T) * Tmix + Alpha_T * Traw

    TempIndex = int(round(Tmix))
    TempIndex = max(0, min(60, TempIndex))
    Rho = DensityTbl[TempIndex]

    Vliq = (LevelPercent / 100.0) * V_Total_m3
    Mass = Vliq * Rho

    dm = Mass - LastMass
    LastMass = Mass


    if DmCount < DM_Smooth_N:
        DmBuf[DmIdx] = dm
        DmSum += dm
        DmCount += 1
    else:
        DmSum -= DmBuf[DmIdx]
        DmBuf[DmIdx] = dm
        DmSum += dm

    DmIdx = (DmIdx + 1) % DM_Smooth_N
    Dm_Smooth = DmSum / DmCount

    dLevel = LevelPercent - LastLevel
    LastLevel = LevelPercent
    dLevel_Ema = (1 - derAlpha) * dLevel_Ema + derAlpha * dLevel

    dPEV = P_EV_kPa - LastPEV
    LastPEV = P_EV_kPa
    dPEV_Ema = (1 - derAlpha) * dPEV_Ema + derAlpha * dPEV

    Stable = (
        abs(Dm_Smooth) < 1.998624e-02 and
        dLevel_Ema >= -9.646880e-04 and
        dPEV_Ema >= -2.864965e-03 and
        not Warning and not Alarm
    )

    if Stable:
        BaselineDm[BaselineIndex] = Dm_Smooth
        BaselineDLevel[BaselineIndex] = dLevel_Ema
        BaselineDPEV[BaselineIndex] = dPEV_Ema

        BaselineIndex = (BaselineIndex + 1)% BaselineWinLen

        if BaselineCount < BaselineWinLen:
            BaselineCount += 1

    if BaselineCount >= 100:
        arr = np.sort(BaselineDm[:BaselineCount])
        N = BaselineCount

        Dm_Med = arr[int(0.5 * (N-1))]
        P10 = arr[int(0.1 * (N - 1))]
        P90 = arr[int(0.9 * (N - 1))]
        Dm_Sig = (P90 - P10) / 2.56

        arr_level = np.sort(BaselineDLevel[:BaselineCount])
        dLevel_Thr = arr_level[int(0.05 * (N - 1))]

        arr_pev = np.sort(BaselineDPEV[:BaselineCount])
        dPEV_Thr = arr_pev[int(0.05 * (N-1))]
    else:
        Dm_Med = 0.0
        Dm_Sig = 0.0
        dLevel_Thr = -0.0025
        dPEV_Thr = -0.012


    ThrWarn = Dm_Med - K_Warn * Dm_Sig
    ThrAlarm = Dm_Med - K_Alarm * Dm_Sig

    MassFlag_W = Dm_Smooth < ThrWarn
    MassFlag_A = Dm_Smooth < ThrAlarm
    DirFlag = (dLevel_Ema < dLevel_Thr or dPEV_Ema < dPEV_Thr)

    CWarn = CWarn + 1 if MassFlag_W and DirFlag else 0
    CAlarm = CAlarm + 1 if MassFlag_A and DirFlag else 0

```

```
    Warning = CWarn >= Consecutive_Warning
    Alarm = CAlarm >= Consecutive_Alarm

    V_raw_L = Vliq * 1000.0

    if not LeakRefInit:
        Rho_Ref = Rho
        LeakRefInit = True

    V_corr_L = V_raw_L * (Rho / Rho_Ref)

    V_EMA += EMA_Alpha * (V_corr_L - V_EMA)
    V_LPF += LPF_Alpha * (V_corr_L - V_LPF)

    KFV_P += KF_Q
    KFV_K = KFV_P / (KFV_P + KF_R)
    KFV_X += KFV_K * (V_EMA - KFV_X)
    KFV_P *= (1 - KFV_K)
    V_Kal = KFV_X

    VReg_EMA[:-1] = VReg_EMA[1:]
    VReg_LPF[:-1] = VReg_LPF[1:]
    VReg_Kal[:-1] = VReg_Kal[1:]

    VReg_EMA[-1] = V_EMA
    VReg_LPF[-1] = V_LPF
    VReg_Kal[-1] = V_Kal


    if Alarm and i >= RegWinLen:
        t = np.arange(RegWinLen)
        slope_ema, _ = np.polyfit(t, VReg_EMA, 1)
        slope_lpf, _ = np.polyfit(t, VReg_LPF, 1)
        slope_kal, _ = np.polyfit(t, VReg_Kal, 1)

        Leak_EMA_Reg = -slope_ema * 60.0
        Leak_LPF_Reg = -slope_lpf * 60.0
        Leak_Kal_Reg = -slope_kal * 60.0
    else:
        Leak_EMA_Reg = 0.0
        Leak_LPF_Reg = 0.0
        Leak_Kal_Reg = 0.0

    LeakRate_EMA = (V_EMA - BufV_EMA[BufIdx]) / LeakWinLen
    LeakRate_LPF = (V_LPF - BufV_LPF[BufIdx]) / LeakWinLen
    LeakRate_Kal = (V_Kal - BufV_Kal[BufIdx]) / LeakWinLen

    BufV_EMA[BufIdx] = V_EMA
    BufV_LPF[BufIdx] = V_LPF
    BufV_Kal[BufIdx] = V_Kal
    BufIdx = (BufIdx + 1) % LeakWinLen

    Leak_EMA = -LeakRate_EMA * 60.0 if Alarm else 0.0
    Leak_LPF = -LeakRate_LPF * 60.0 if Alarm else 0.0
    Leak_Kal = -LeakRate_Kal * 60.0 if Alarm else 0.0

    out.append([
        Warning,
        Alarm,
        Leak_EMA,
        Leak_LPF,
        Leak_Kal,
        Leak_EMA_Reg,
        Leak_LPF_Reg,
        Leak_Kal_Reg
    ])


output_path = r"C:\Users\KH9687\AmesimData\MachineLearning\plc_leak_output.csv"

pd.DataFrame(out, columns=[
    "Warning",
    "Alarm",
    "LeakageSize_EMA_Lmin",
    "LeakageSize_LPF_Lmin",
    "LeakageSize_Kalman_Lmin",
    "LeakageSize_EMA_Lmin_Reg",
    "LeakageSize_LPF_Lmin_Reg",
    "LeakageSize_Kalman_Lmin_Reg"
]).to_csv(output_path, index=False)
print("Simulation Done")
```

References

[1] A. H. Alami *et al.*, "Expansion cooling prospects for large-scale applications," *International Journal of Thermofluids*, 2023, accessed: 2026-02-15. [Online]. Available: https://doi.org/10.1016/j.ijft.2023.100437

[2] S. Xu, H. Zhang, and Z. Wang, "Thermal management and energy consumption in air, liquid, and free cooling systems for data centers: A review," *Energies*, vol. 16, no. 3, 2023, accessed: 2026-02-15. [Online]. Available: https://doi.org/10.3390/en16031279

[3] S. A. Tassou and I. N. Grace, "Fault diagnosis and refrigerant leak detection in vapour compression refrigeration systems," *International Journal of Refrigeration*, vol. 28, no. 5, 2005, accessed: 2026-02-15. [Online]. Available: https://doi.org/10.1016/j.ijrefrig.2004.12.007

[4] A. L. Sousa, A. F. Brochado, E. Rocha, and A. Andrade-Campos, "A state-of-the-art review of scientific, industrial, and commercial developments in water leakage management for water supply systems," *Water Research*, 2026, accessed: 2026-02-15. [Online]. Available: https://doi.org/10.1016/j.watres.2026.125853

[5] G. An, Z. Huang, and Y. Li, "Constant false alarm rate detection of pipeline leakage based on acoustic sensors," *Scientific Reports*, vol. 13, 2023, accessed: 2026-02-15. [Online]. Available: https://doi.org/10.1038/s41598-023-41177-3

[6] J. M. Maza-Ortega, E. Acha, S. García, and A. Gómez-Expósito, "Overview of power electronics technology and applications in power generation, transmission and distribution," *Journal of Modern Power Systems and Clean Energy*, vol. 5, no. 4, pp. 499–514, 2017.

[7] Z. Li, P. Wang, J. Zhang, and H. Guan, "A model-free method for identifying time-delay characteristics of hvac systems based on multivariate transfer entropy," *Building and Environment*, vol. 217, 2022, accessed: 2026-02-15. [Online]. Available: https://doi.org/10.1016/j.buildenv.2022.109072

[8] Z. Zuo, L. Ma, S. Liang, J. Liang, H. Zhang, and T. Liu, "A semi-supervised leakage detection method driven by multivariate time series for natural gas gathering pipeline," *Journal of Loss Prevention in the Process Industries*, 2023, accessed: 2026-02-15. [Online]. Available: https://doi.org/10.1016/j.jlp.2022.104985

[9] Y. A. Cengel and M. A. Boles, *Thermodynamics: An Engineering Approach*, 9th ed. McGraw-Hill, 2019.

[10] Siemens Digital Industries Software, "Simcenter amesim system simulation software," 2023, accessed: 2026-02-15. [Online]. Available: https://www.sw.siemens.com/en-US/simcenter/amesim/

[11] R. A. Beier and J. D. Spitler, "Weighted average of inlet and outlet temperatures in borehole heat exchangers," *Applied Energy*,

vol. 179, 2016, accessed: 2026-02-15. [Online]. Available: https://doi.org/10.1016/j.apenergy.2016.04.077

[12] D. C. Montgomery, E. A. Peck, and G. G. Vining, *Introduction to Linear Regression Analysis*, 5th ed. Wiley, 2012.

[13] R. J. Hyndman and G. Athanasopoulos, "Forecasting: Principles and practice," 2021, accessed: 2026-02-15. [Online]. Available: https://otexts.com/fpp3/

[14] G. Welch and G. Bishop, "An introduction to the kalman filter," 1997, accessed: 2026-02-15. [Online]. Available: https://www.cs.unc.edu/~welch/media/pdf/kalman_intro.pdf