'Water Underground': Real-time, continuous monitoring of the underground water's quantity and quality.

MAVROFORAKIS M.[1], GEORGIOU H.[1*], MALISIOVAS V.[1], PSYCHIAS C.[1], PAPADIMITRIOU D.[1] and SARIOGLOU K.[2]

[1] InTTrust S.A., 2, Ipirou str., Ag. Paraskevi, 15341 Athens, Greece.

[2] Computer Science Department, University of Crete, 70013 Heraklion, Crete, Greece.

*corresponding author:
e-mail: cgeorgiou@inttrust.gr

Abstract Water resource management is one the most urgent aspects of environmental protection and sustainability policies world-wide. Accurate, real-time remote sensing of the status of underground reservoirs is required for proper regional planning, prevention of droughts, optimized farming etc. 'Water Underground' is a low-cost solution, based on a combination of Internet of Things (IoT) local sensing, Edge computing, Cloud storage, web services and Machine Learning (ML) and predictive analytics, continuously monitoring the level of underground water and its quality. Specifically, water level is monitored via an IoT apparatus providing the Static (SWL) and Pumping Water Level (PWL). Moreover, the quality of water is tracked via measuring the Total Dissolved Solids (TDS), Oxidation-Reduction Potential (ORP), temperature, pH, electrical-conductivity, etc. Local processing in the IoT device includes measurements' transformations and robust adaptive control for the device's actuators. The reservoir dynamics is tracked and modeled using Cloud-based predictive analytics. The corresponding Cloud services include long- and short-term detection of periodic trends, Drawdown (DD) patterns, prediction of SWL, predictive maintenance via PWL tracking, etc. The overall solution has received international recognition in IBM Challenge 2020 as top-7 finalist for Europe. The platform is currently under prototype deployment in several sites in the Attica region of Greece.

Keywords: hydrology, water management, remote sensing, predictive analytics, sustainability.

1. Introduction

Water resources monitoring and management is a very important aspect of environmental and sustainability factors of developmental strategies, especially in urban and suburban regions of high population density, as well as in contexts of spatial (geographic) or temporal (seasonal) scarcity of water. Close monitoring and continuous assessment of water sources and reservoirs is imperative for the policy-makers to employ proper public safety and health protection strategies.

The 'Water Underground' system is a holistic, low-cost, IoT- and ML-enabled solution for sensing, analyzing and monitoring localized water resources, typically wells or reservoirs, which require careful management and long-term sustainability. It combines the currently available low-cost single-board computers (SBC) and modular sensors with networking and Cloud services, in order to provide both on-site Edge computing capabilities together with computationally-heavy back-end processing and storage in the Cloud. The system monitoring includes a user-friendly web-based interface that provides dashboards for tabular and graphical presentation of all the relevant data, status reports and predictions.

In the following sections, the three main components of the Water Underground deployment are described in more detail, namely:

- Sensing I/O, IoT, Edge computing
- Back-end processing, ML/analytics, predictions
- Frond-end application, user interface

2. Methodology

2.1. Data preprocessing

The raw measurements of the water level are usually very noisy and may contain range errors or missing values. The first part of the data processing includes detection and correction of such errors by proper methods. Noise and isolated errors are averaged against neighboring values in the time series; missing values are also averaged if isolated, or estimated via localized linear regression in case of missing blocks. The system pre-processing is designed to be minimally intrusive and all other aspects are addressed in the subsequent steps by other means, e.g., Kalman filtering, as described later on.

2.2. Water reservoir dynamics and state machine

In order to properly design the SWL and PWL estimators, the water reservoir dynamics are employed as the baseline for the state machine design of the controller. More specifically, the water reservoir is modeled as a cylinder-like tank with cross section of arbitrary shape as the base and continuous water inflow along its vertical-dimension walls, inversely dependent on the drainage height during pumping or recovering states (non-stabilized), or approximately in equilibrium otherwise (stabilized). Hence, "pumping" and "stabilized" are the two independent factors and the total number of states are four, as depicted in the STD in Figure 1.

According to Torricelli's law, the parting speed of water inflow is based on the difference of height between the stabilized water level when no pumping (SWL) and the current (lower) water level during pumping, assuming no air resistance, viscosity, or other hindrance to the fluid flow. After some time, the inflow rate may match the outflow rate caused by the pumping, hence establishing a new equilibrium (PWL). The total inflow can be modeled according to the change in water (mass), as in Eq.1:

$$\frac{dm(t)}{dt} = Q_{in}\left(\frac{h_0 - h(t)}{h_0}\right) - Q_{out} \Rightarrow \cdots$$
$$\ldots \Rightarrow \frac{dh(t)}{dt} + \left(\frac{Q_{in}}{\rho A h_0}\right) h(t) = \frac{Q_{in} - Q_{out}}{\rho A} \qquad (1)$$

where $m(t)$ is the mass of water, ρ is the water density, A is the base area of the reservoir, Q_{in} is the inflow rate dependent on the relative difference of the current water level $h(t)$ from the no-pumping water level h_0 and Q_{out} is the fixed outflow rate (pumping). This is an ordinary differential equation of first order that can be solved as:

$$\frac{dy}{dx} + P(x)y = Q(x)$$
$$\Rightarrow y e^{\int P(x)dx} = \int Q(x) e^{\int P(x)dx} dx + C_0 \qquad (2)$$

Substituting the terms in Eq.2 with the model's parameters, the final solution is:

$$\left.\begin{array}{r} x = t \\ y = h(t) \\ P(x) = \dfrac{Q_{in}}{\rho A h_0} = C_1 \\ Q(x) = \dfrac{Q_{in} - Q_{out}}{\rho A} = C_2 \end{array}\right\} \Rightarrow h(t) e^{\int C_1 dt} = \int C_2 e^{\int C_1 dt} dt + C_0$$

$$h(t) = c + \alpha e^{\beta t} \quad , \quad \begin{cases} c = \frac{C_2}{C_1} = \left(1 - \frac{Q_{out}}{Q_{in}}\right) h_0 \\ \alpha = C_0 \\ \beta = -\frac{Q_{in}}{\rho A h_0} \end{cases} \qquad (3)$$

where $h(t)$ is the parameterized estimator for the current water level in the reservoir during the non-stabilized states, i.e., when the drainage (Q_{out}>0) and the recovering (Q_{out} =0) are onset. In stabilized states, the current water level is approximately constant, i.e., a linear function with approximately zero slope. Figure 2 illustrates these concepts and depicts the analytical models the water level estimator for each of the four states of the system. Table 1 provides sample estimates for the parameters of non-stabilized states in Eq.3 using measurements from a two-month time frame, where the constants (c) also indicate the asymptotic values at PWL (22.49 m) and the SWL (30.51 m).

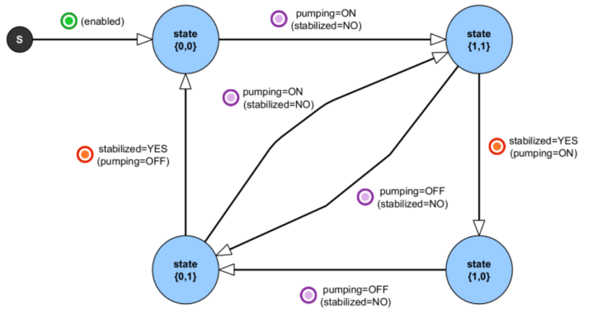

Figure 1. State transition diagram of the controller.

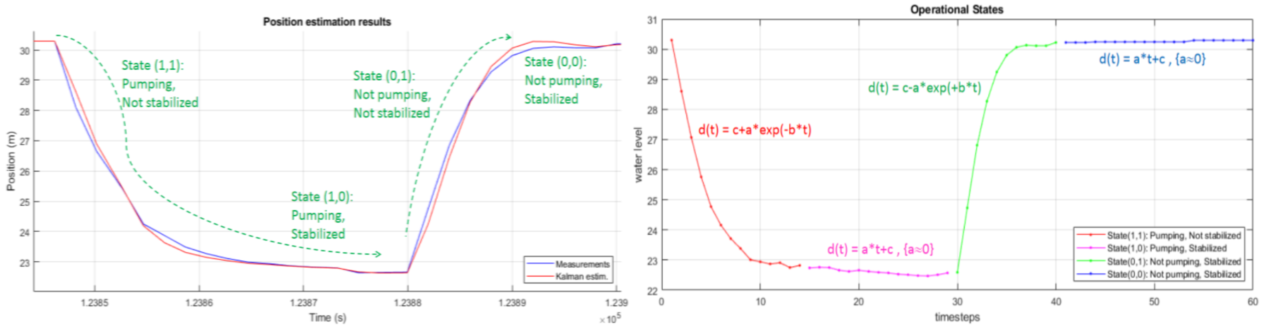

Figure 2. System states and analytical models for the water level in the reservoir, illustrating the actual water level measurements (blue) and the Kalman filter (red) used for adaptive noise-resilient estimators.

Table 1. Sample estimates for the parameters of non-stabilized states (exponential).

	Pumping: state(1,1)	**No pumping: state(0,1)**
scaling: α	+10.84	-2.481e5
exponent: β	-0.3028	-4.208
constant: c	**+22.49**	**+30.51**
fit: R	0.9962	0.9898
error: RMSE	0.1496	0.2619

2.3. Water level and state detection

For the detection of stabilized states, linear regression can be used with proper thresholds over goodness-of-fit (residuals) and slope (change rate). However, the use of exponential fits for the other two non-stabilized states, as depicted in Figure 2, can provide valuable information about the dynamics of the water reservoir, as well as the pumping apparatus in terms of fault detection and predictive maintenance. The typical sampling rate used in the apparatus is in the order of 20-180 seconds, calibrated against the drainage and recovery rates (slopes) for each specific installation. In any case, this is plenty of time for an online simple exponential fit upon a sliding window of 5-15 data points (a few minutes). In practice, each model fit for the next possible state is updated with every new data point and the most probable state transition is realized when the corresponding threshold is met, i.e., when the exponential has reached the PWL or SWL and when the subsequent linear fit exhibits near-zero slope, as Figure 2 shows.

2.4. Kalman and F-test filtering

Two methods have been employed for data filtering in combination with the state detection approach described above, i.e., before employing the exponential or linear fits. Namely, the first is Kalman filtering [Haykin] for robust estimation of the water level and its rate of change, i.e., “velocity”, but with specific known limitations, most importantly its significant processing load if the system model is large, as well as the need for fixed-rate data input. The second alternative is employing a much simpler statistical filter based on F-test [Spiegel], i.e., tracking of running variances. In practice, the variance in the stabilized states is compared with the variance in the current sliding window via a standard F-test at a specific significance level (typically a=0.05). This approach is less accurate than Kalman filter and does not provide proper estimation of the slope, but it can be applied with variable-rate data input. In both Kalman and F-test filtering, the baseline variance is measured separately for PWL and SWL, after the initial system setup and with periodic re-calibration. Additional post-processing may be applied if necessary via median filtering, e.g., when the reservoir dynamics are inherently highly volatile and result in noisy slope estimations.

2.5. Predictive modeling

As described in the previous section, the use of separate model fits for each of the four states enables the extraction of valuable information about the dynamics of the water reservoir and the pumping apparatus. More specifically, the variance measured during the stabilized states s(x,0), i.e., with or without pumping, can reveal possible deficiencies or noise of the water level measurement sensing. Similarly, the goodness-of-fit and the exponential parameters (scale, exponent) during the non-stabilized states s(x,1) can reveal possible leaks and other faults in the pumping apparatus. Using the running best-fit model parameters for each state and comparing them to known periods of nominal operation of the system, persistent deviations can be detected, quantified and characterized in terms of severity. For example, indicators of degrading performance of the pumping apparatus include slower exponential rate when pumping starts, large variance (instability) when approaching the expected PWL, longer-than-expected time required to reach it, etc.

In addition to predictive maintenance, three predictive models have also been designed and implemented in the system, namely: (a) next-day SWL, (b) next-day PWL, (c) 3-hour look-ahead “spot” water level. Models (a) and (b) are examples of medium/long-term predictors, using the aggregated statistics of the previous 24-hour period, including max, min, mean, linear trend (slope), etc, for the entire set and for the specific state, SWL or PWL, respectively. Two ML regression methods have been employed with very similar results, namely Extreme Gradient Boosting (XGBR) [Osman] and Random Forest (RFR) [Breiman]. Model (c) is an example of short-term predictor, using a 9-hour look-back sliding window on the recorded measurements and estimating the water level 3 hours ahead, again employing either XGBR or RFR with similar results. In the specific use case of system deployment and data recording over a period of almost a year, the typical performance (RMSE) of these predictors is in the order of 6.5-8.5 cm for (a) and (b), and 16-17 cm for (c), compared to a value range of about 22.5 m (PWL) to 30.5 m (SWL), as Figure 2 and Table 1 show.

2.6. Long-term analytics and calibration

Besides short-term statistics, long-term tracking of the SWL can reveal the seasonal trends and variability of the water level, which is very important for the optimal management of water resources in the specific site. Similarly, the long-term tracking of the PWL can reveal the seasonal changes in the inflow, which is indicative of the water availability in the larger region. Additionally, these long-term seasonal data are necessary for the proper re-calibration of the system, as PWL and SWL are necessary for the correct model estimations, fault detection and predictive maintenance. Figure 3 illustrates such long-term plot for SWL over several months.

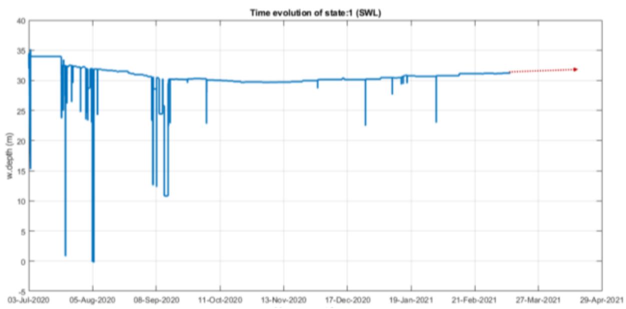

Figure 3. Long-term plot for SWL over a period of several months; sharp drops reveal calibration errors or system malfunctions that were detected and corrected.

3. Implementation and Deployment

3.1 Sensing (I/O) and asynchronous processing

The IoT apparatus consists of several sensors and a water container in order to monitor the level and quality of the water (Figure 4). The SBC used is Raspberry Pi (RPi) 4. The RPi controls the hardware peripherals (e.g., relays, sensors), reads and processes sensors’ measurements (e.g., atmospheric pressure, water depth, pH, Electro-conductivity (EC), Total Dissolved Solids (TDS), Oxidation-Reduction Potential (ORP) and temperature of the water) and communicates with the Cloud to upload data and to receive remote commands and updated configuration and firmware.

Figure 4. IoT apparatus (main hub).

The OS of the IoT apparatus is Raspbian, with the control S/W programmed in Python and using 'asyncio' module for RPi's resources' use maximization. The data collected on the device are pre-processed and transformed locally by using modules such as 'numpy' and 'scipy'. Furthermore, an SQLite3 database is used to store data locally for a configurable period of time, in order to retain data during loss of internet connection. The backbone of the IoT device control is Node-RED, which acts as an orchestrator between the Python scripts and the commands to/from the Cloud.

3.2 Networking

The transportation of Cloud-to-Device messages (C2D) and vice versa is accomplished by using the MQTT protocol. The IBM IoT Platform acts as the broker and is responsible for distributing messages to/from the connected clients. The connection to the Cloud and the subscription to a certain topic is handled though Node-RED. Messages between the IoT apparatus and the Cloud are exchanged in JSON format. This enables remote control and firmware updates, by subscribing to the appropriate topic.

3.3 Front-end and User Interface

The IBM Cloud hosts the front-end application which is a Cloud Foundry App, created using the IBM SDK for Node.js. A Web application is used to access information in a user-friendly layout and perform IoT device operations. A backend application (Express.js) accesses the historical data database and the predictive analytics endpoints. A frontend application (Angular) consumes the data via REST protocol. Water level, quality of water measurements and prediction/trends (next-day SWL, next-day PWL, 3-hour look-ahead "spot" water level) are displayed in a dashboard view (Figure 5). Based on the latest HTML5 features, notifications are used in case of important events. Raw IoT apparatus data are available in graph and table views. The backend application broadcasts commands to IoT devices via a MQTT broker. Each IoT device subscribes to MQTT commands so that a user may transfer objects in JSON format (*configuration*, *firmware*, *etc*) to a device using a Web interface.

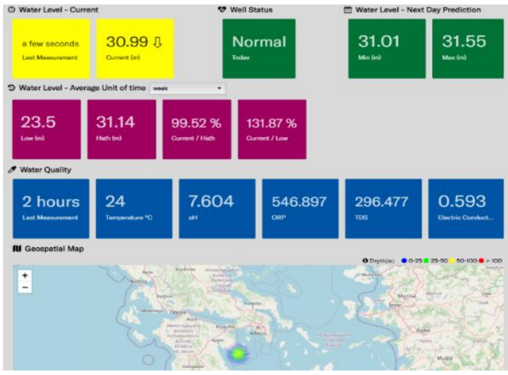

Figure 5. Dashboard view with latest water level measurement with the upcoming trend, well status, next day predictions for max and min level, level average values and the latest water quality measurement.

4. Discussion & Conclusions

One of the unique features of the Water Underground system is the inherently distributed deployment of the various functional modules. Significant amount of the raw data registration and pre-processing is performed on the RPi device itself, configured and fine-tuned for each specific deployment. Thus, global sub-optimal implementations in the back-end are avoided and the quality of the incoming data is ensured. Furthermore, the design and implementation of the Machine Learning modules in the back-end, as well as the front-end Web application, do not need to be closely coupled to a specific RPi device deployment. This is particularly important in the case of analytics and model re-training, since in this way datasets can be constructed across several deployments and generalization is enhanced.

Finally, the overall design, implementation and deployment of Water Underground is a successful use-case and prototyped solution for various other tasks and contexts, which require a wide combination of IoT sensing, robust Edge processing, networking and Cloud services, computationally intense back-end processing, as well as versatile front-end user interfaces.

References

Breiman L. (2001), Random Forests, *Mach. Learn.*, **45**, 5-32.

Haykin S. (2013), Adaptive Filter Theory (5th ed.), Pearson, UK.

Osman A., Ahmed A., et.al. (2021), Extreme gradient boosting (Xgboost) model to predict the groundwater levels in Selangor Malaysia, *Ain Shams Eng. Journal*, **(online)**.

Spiegel M.R., Schiller J., Srinivasan R.A. (2009), Probability and Statistics (3rd ed.), McGraw-Hill, New York.