Politecnico di Torino

Master of science program in Environmental and Land Engineering – Climate Change
A.Y. 2024/2025
Graduation Session March 2025

Analysis and modelling of hydrogeological data in spring systems

Input-output correlations between meteorological and hydrological variables in the case study of the Entrebin spring

Supervisor:
Professor Paolo Dabove

Co-supervisors:
Dr. PhD Martina Gizzi
Professor Glenda Taddia

Candidate:
Davide Salvalaggio
matr. 319647

Table of Contents

Introduction

Study context

Mountain springs represent crucial nodes in the hydrological cycle, acting as interfaces between groundwater and surface systems. In alpine environments, these resources are particularly sensitive to climatic and anthropogenic variations, given their dependence on recharge processes linked to snowfall, glacier melt, and complex geological dynamics. However, understanding their dynamics remains a scientific challenge due to the limited accessibility of sites, data heterogeneity, and seasonal variability amplified by climate change.

The hydrographic basin of the Aosta Valley, characterised by rugged orography and intense past glacial activity, hosts numerous spring systems of strategic interest for local water supply. Among these, the Entrebin spring stands out as an interesting case due to its geographical location, atypical hydrological regime, geological setting, and interaction with historical infrastructures. Located at 981 meters above sea level on the left side of the Dora Baltea, the spring is fed by an aquifer whose recharge is dominated by various components, including snowmelt and rainfall. Unlike conventional systems, which exhibit flow peaks in spring and lows in summer, Entrebin reaches maximum levels between August and September, with minimums between April and May. This "phase-shifted" behaviour suggests delayed recharge mechanisms, likely linked to geological factors, climatic dynamics, and anthropogenic interventions.

The alpine climatic context further amplifies the challenges related to hydrological data. Meteorological stations, such as the one in Roisan-Preyl (3 km from the spring under study), used in this research, suffer from instrumental limitations: non-heated rain gauges underestimate winter precipitation, while altimetric differences (50 m between the station and the spring) can still introduce minor variations in terms of local microclimates. At the same time, the water level data collected by the Politecnico di Torino between 2011 and 2024 present temporal gaps (e.g., interruptions in 2014–2015) and instrumental noise.

This study is part of a research effort aimed at bridging the gap between theoretical hydrogeological models and practical applications, highlighting how advanced data analysis techniques can help overcome the limitations of existing spring monitoring systems. The use of advanced techniques—from the Kalman filter for noise reduction to Fourier functions for identifying seasonal patterns—reflects the need for innovative approaches in data-limited contexts. The choice of Entrebin as a case study is not coincidental: its hydrological anomaly offers a unique opportunity to test methodologies transferable to other mountain springs under climatic stress, thereby contributing to the resilience of water systems in an era of increasing environmental instability.

Aims

The study is structured around a dual scientific and operational axis, aiming on one hand to decipher the uncommon hydrological dynamics of the Entrebin spring and on the other to develop innovative analytical tools managing complex spring systems in mountainous contexts.

Firstly, the research aims to address the challenges related to the quality and discontinuity of hydrological datasets. Through the application of the Kalman filter, it proposes to optimize the reliability of time series, mitigating instrumental noise and compensating for data gaps caused by technical failures or extreme climatic events. This approach not only ensures a solid informational foundation for subsequent analyses but also represents a replicable model for contexts with similar instrumental limitations.

Secondly, the study aims to explore the relationships between meteorological inputs (precipitation, air temperature) and hydrological outputs (water level) through cross-correlation analysis. Unfortunately, this type of approach proves to be non-functional for the case study in question. Instead, the identification of time delays in the spring's response to climatic events, combined with the modeling of seasonal components using Fourier functions, allows for the isolation of recurring patterns and delayed recharge mechanisms, often obscured by the geological complexity of the basin.

Finally, the thesis aims to translate the analytical results into a predictive model capable of anticipating variations in water level and air temperature over multi-year time scales, using the calendar year 2024 as an example. This tool, calibrated on seasonal cyclicity and corrected for residual biases, not only attempts to validate the effectiveness of the adopted techniques but also seeks to provide, for instance, an operational framework for the sustainable planning of water resources, particularly relevant in scenarios of increasing climatic variability.

Thesis structure

The thesis is organized into three main sections:

In the first section, the hydrogeological principles of spring systems, monitoring techniques, and challenges related to data quality are examined, with references to previous studies and cases of interest.

In the second section, the study site, the datasets used (level, precipitation, temperature), the data pre-processing phases (Kalman filter), and the application of mathematical (polynomials, Fourier functions) and statistical (cross-correlations) models are described.

In the third section, the results of the analyses are presented, with a focus on the effectiveness of the Fourier model in predictions and the limitations of cross-correlations. Corrections to reduce bias in the forecasts and critical evaluations of the adopted approaches are included.

The work concludes with a reflection on the limitations of the study, proposals for future research (e.g., the integration of machine learning), and practical implications for the management of water resources in mountainous environments.

Background and State of the Art

Spring Systems and Their Hydrological Dynamics

Spring systems are a crucial component of hydrology, acting as the points where groundwater emerges at the surface. The functioning of these systems is highly complex and influenced by geological, climatic, and hydrological variables.

Recharge Processes

The recharge of aquifers occurs primarily through precipitation infiltration. Rainwater and snowmelt penetrate the soil and, under the force of gravity, reach underground aquifers. This process is affected by:

- Soil and Rock Type: Permeable soils, such as sand and gravel, promote infiltration, while clayey soils hinder it.
- Vegetation: Vegetation enhances infiltration through root systems, which create preferential channels in the soil.

Discharge Processes

Aquifers discharge water when it rises to the surface through springs. This process can occur for various reasons:

- Hydrostatic Pressure: When underground water pressure exceeds atmospheric pressure, water emerges at the surface.
- Geological Features: Fractures or faults in the rock can facilitate the upward movement of water.

Key Variables

The main factors influencing the functioning of spring systems include:

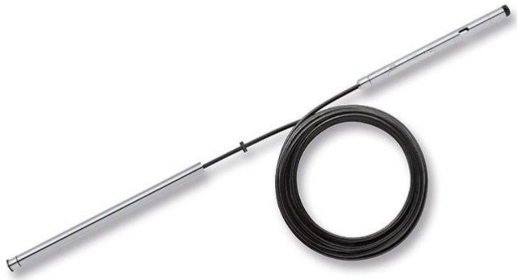
- Precipitation: The amount and distribution of rainfall and snowfall directly impact aquifer recharge.
- Temperature: Temperature fluctuations affect evaporation and transpiration, reducing the amount of water available for infiltration.
- Geology: The composition and structure of the terrain determine its water storage capacity and the speed of groundwater movement.

Monitoring of Levels, Precipitation, and Temperatures

The monitoring of spring levels, precipitation, and temperature is essential for understanding hydrological dynamics and ensuring sustainable water resource management [Lo Russo et al., 2021].

Techniques and Tools

- Water Level Measurement: Hydrometers measure water levels in springs and aquifers. Types include float-operated hydrometers and pressure-based devices.
- Rain Gauges: Rain gauges measure precipitation over a given time period. They range from manual models to automated systems, varying in cost and precision.
- Thermometers: Thermometers monitor air and water temperatures. These include traditional mercury thermometers as well as advanced digital sensors.



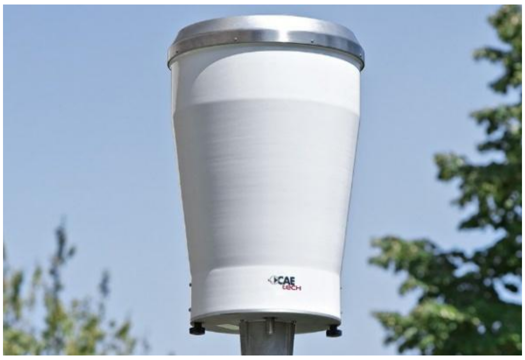


Figure 2 - Hydrometeorological monitoring instrumentation. On the left, the OTT-CTD for measuring water level, water temperature and electrical conductivity (source: https://corr-tek.it/prodotti/livello-idrometrico/ott-ctd/).

Figure 2 - Hydrometeorological monitoring instrumentation. On the right, an example of rain gauge for precipitation measurement (source: https://www.cae.it/eng/products/rain-gauges/pg10-pg10r-rain-gauge-pd-9.html).

Practical Limitations

- Spatial Discrepancies: Monitoring stations are often unevenly distributed, leading to representational gaps [Piersanti et al., 2007]. For instance, a station in a valley might not accurately reflect conditions in nearby hilly regions.
- Instrument Discrepancies: Instruments can differ in calibration and precision, leading to variations in collected data. Standardizing tools and methodologies are crucial to reduce inconsistencies.
- Accessibility and Maintenance: Monitoring stations in remote or mountainous areas can be challenging to access. Regular maintenance is vital to ensure accurate and reliable data collection.

Issues Related to Data Quality

Hydrological datasets often present various challenges that affect the accuracy and validity of hydrological analyses [Mondani et al., 2022]. Identifying and addressing these issues is essential to ensure reliable and precise results [ISPRA Ambiente, 2013].

Instrumental Noise

Instrumental noise is one of the main issues in hydrological datasets. Measurement tools can introduce errors due to several factors, such as malfunctions, environmental interference, and vibrations. For example, hydrometers may record unrepresentative water level variations due to electrical or mechanical interferences. This noise can distort data and negatively impact analyses.

Missing Data

Data gaps are a common problem in hydrological datasets. Missing data can result from instrument failures, extreme weather conditions preventing data collection, or inadequate maintenance. These gaps make temporal analysis challenging and may compromise forecasting accuracy. Methods like interpolation or statistical modelling can be used to estimate missing data, but these solutions do not always guarantee precise results.

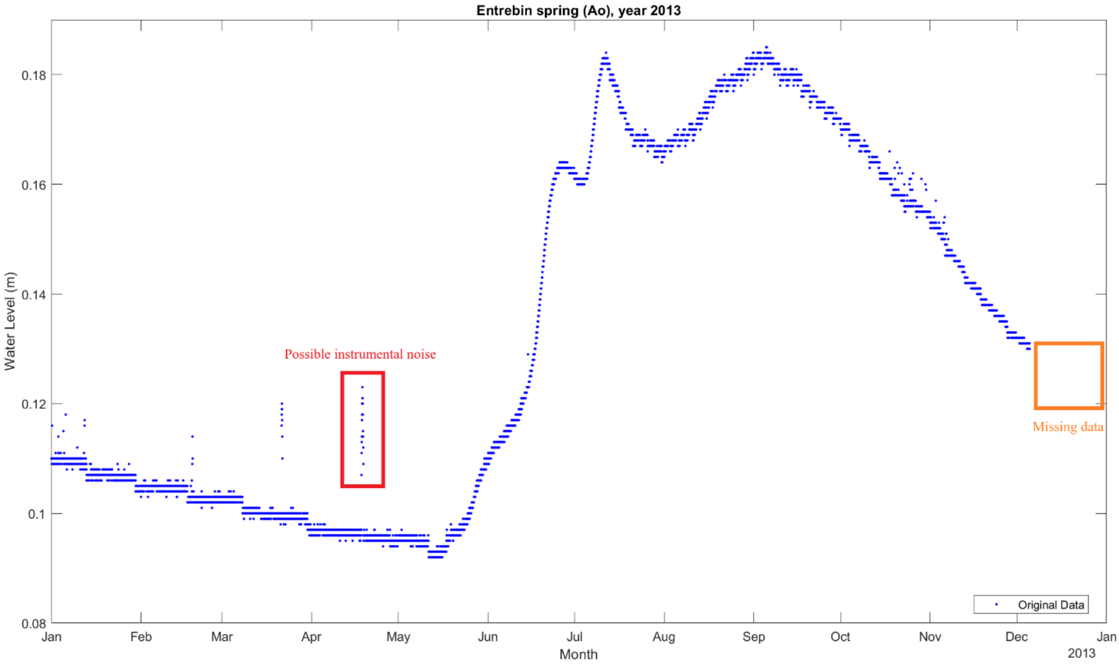


Figure 3 - Water level over time at Entrebin Spring (Ao), 2013. Potential instrument noise is highlighted in red and missing data points are marked in orange.

Spatial Discrepancies

The uneven distribution of monitoring stations can lead to spatial inconsistencies in the data. For instance, a station located in a valley may not accurately reflect hydrological conditions in adjacent hilly or mountainous areas. Addressing this issue requires improving monitoring networks and applying spatial interpolation techniques to estimate data in uncovered regions.

Temporal Discrepancies

Changes in data collection methods over time can result in temporal inconsistencies. For example, technological advancements or shifts in measurement practices may affect the consistency of historical data. Standardizing instruments and data collection methodologies is crucial to ensure consistent and reliable datasets.

Instrument Calibration

Different instruments may vary in precision and calibration, leading to discrepancies in collected data. Regular calibration of instruments is essential to minimize these differences and ensure accurate and reliable data. Calibration procedures should be standardized and rigorously followed to avoid systematic errors.

Techniques for Analysis and Prediction in Hydrogeology

Techniques for analysis and prediction in hydrogeology are vital for understanding and forecasting hydrological processes. These methods include statistical techniques, mathematical models, and machine learning algorithms, which enable the analysis of large datasets and the generation of accurate predictions.

Kalman Filter

The Kalman filter is a recursive algorithm used to estimate the state of a dynamic system in the presence of noise [Reid et al., 2001]. This method is valuable for predicting water levels, as it enhances forecast accuracy and reduces uncertainty. In hydrogeology, the Kalman filter can optimize the processing of historical data and provide more reliable estimates of hydrological parameters [Clark et al., 2008].

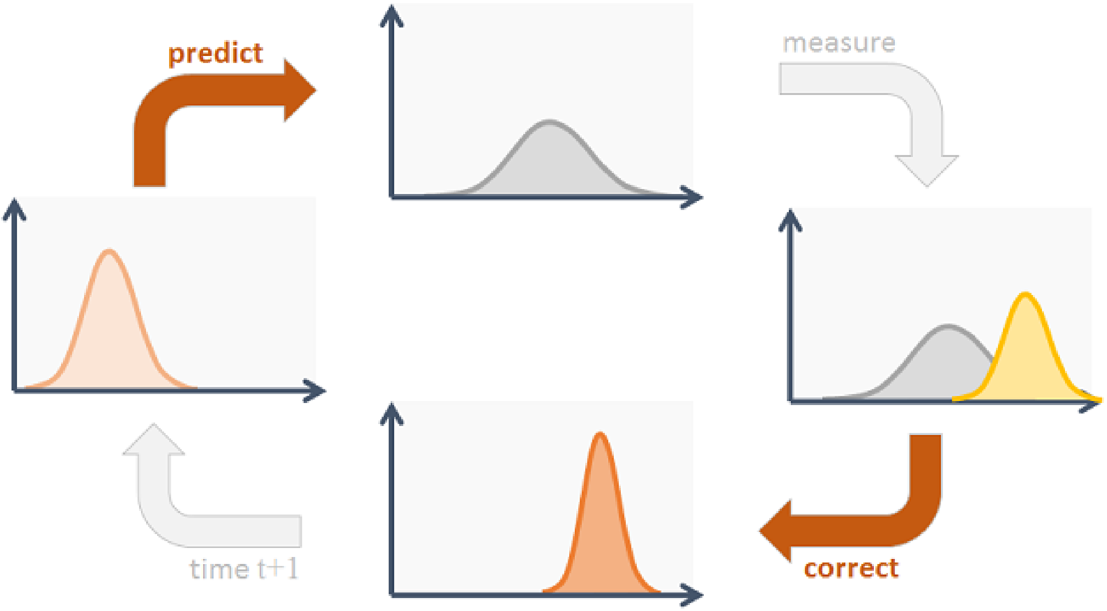


Figure 4 - How work Kalman filter. The filter iteratively predicts the system state through motion model and refines the prediction by incorporating noisy measurements. This process alternates between prediction (propagating state estimates) and correction (updating estimates with new data), continuously improving accuracy.

Mathematical Models

Mathematical models are fundamental tools for simulating hydrological processes. These models can range from simple polynomial equations based on available data to advanced numerical simulations [Viero et al., 2014]. They help reduce noise and improve prediction accuracy. Applying these models allows for detailed examination of spring system behaviors and scenario-based forecasting, supporting sustainable water resource management.

Cross-Correlation Techniques

Cross-correlation techniques measure the relationship between two variables. For example, analysing the correlation between precipitation and runoff can identify time lags between rainfall events and surface water discharge. These techniques help detect patterns and trends in hydrological data, enhancing the understanding of spring system dynamics [Lo Russo et al., 2015].

Machine Learning

Machine learning techniques, such as neural networks and regression algorithms, are increasingly used in hydrogeology to analyse large datasets and make predictions [Pyo et al., 2023]. These methods improve hydrological forecast accuracy and identify complex patterns in the data that might not be apparent with traditional methods. Machine learning enables the development of robust predictive models that adapt to varying conditions and variables.

Materials and Methods

Description of the Study Site

The Entrebin spring, located in the municipality of Aosta, represents a significant example of a mountain water resource subject to seasonal variations. The geographical and hydrological context of the Entrebin spring makes it a valuable yet vulnerable resource. Understanding its hydrological and climatic characteristics is crucial for sustainably managing water supply, especially in the context of climate change.

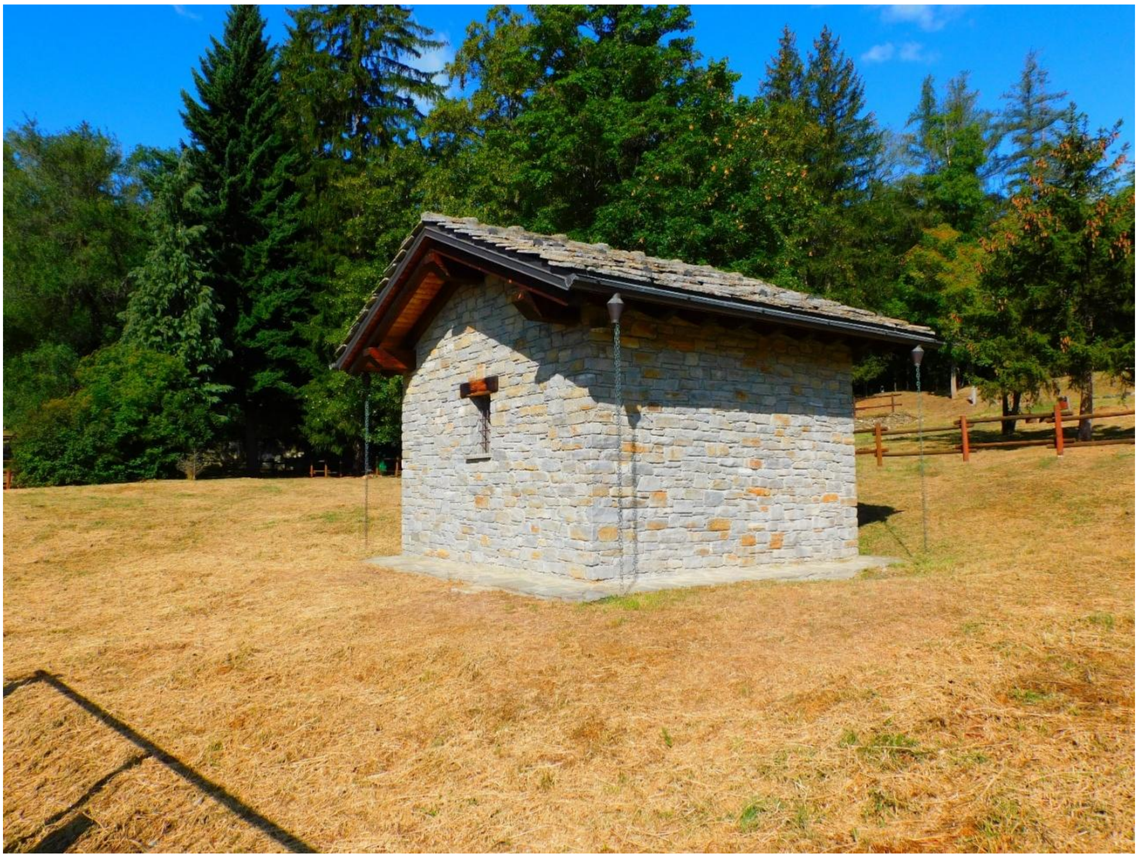


Figure 5 - Monitoring infrastructure housing the spring and associated instrumentation for hydrological data collection.

Geographical Location

The Entrebin spring is situated at approximately 981 meters above sea level, in the village of Entrebin on the left orographic side of the Dora Baltea. It is easily accessible by following the Strada Statale 27 of the Gran San Bernardo and then taking the Strada Regionale 38 towards the Entrebin junction.

Geographical coordinates:

- Latitude: 45.754° N
- Longitude: 7.316° E

The geographical context of the area is essentially characterized by ablation and basal glacial deposits, which are difficult to differentiate due to the complex evolutionary history of the slope at the confluence of two valleys, the Buthier torrent valley and the central valley of the Dora Baltea, both influenced by the action of thick glacier tongues.

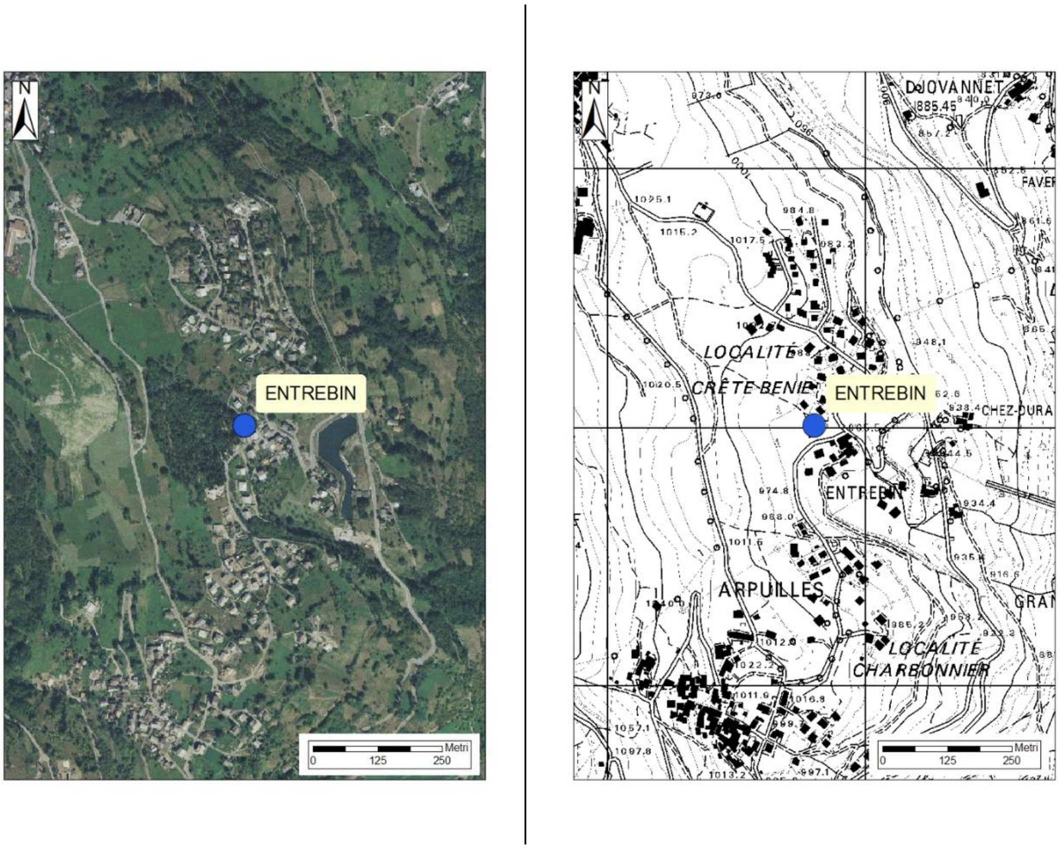


Figure 6 - Geographic location of the Entrebin spring in Aosta Valley. Left: Satellite image of the area. Right: Topographic map to see the spring's position.

Hydrological and Climatic Characteristics

Water Intake Structure

The intake structure of the Entrebin spring plays a crucial role in collecting and making available the water from the hydrogeological system. The spring is captured through a drainage gallery made of concrete, buried at a depth of approximately 2.7 meters. The 39.53 meters long gallery intercepts the groundwater through perforations in the masonry bricks, channelling the flow into longitudinal canals that lead to two accumulation tanks. This infrastructure represents an example of human intervention to optimize the availability of water from a spring characterized by good seasonal regularity, despite the influence of climatic factors such as precipitation and snowmelt. However, the geological complexity of the area, characterized by mixed glacial deposits and schists, affect the functioning of the intake, increasing the risk of clogging or structural inefficiencies.

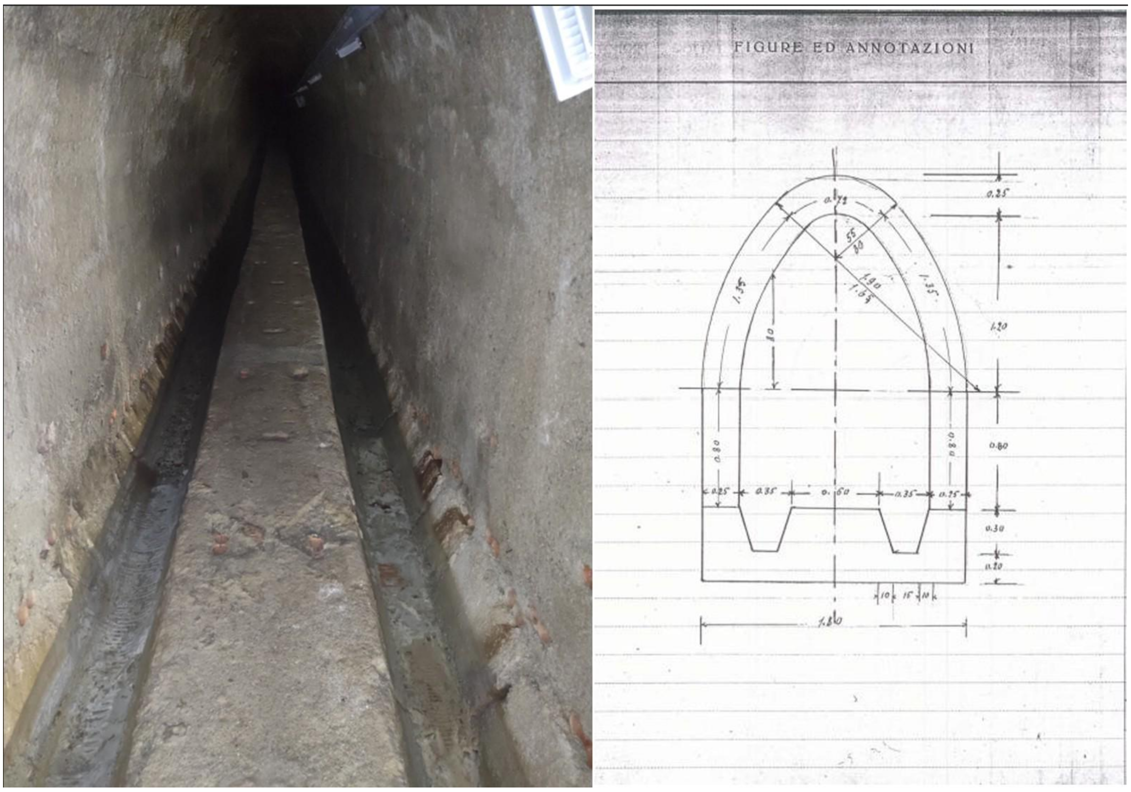


Figure 7 - Drainage gallery at Entrebin. Left: Photograph of the gallery entrance. Right: Technical drawing of the gallery's cross-section, showing dimensions and structural details.

Hydrological Regime

The spring is fed by a porous unconfined aquifer, primarily recharged through snowmelt and atmospheric precipitation. The direction of groundwater flow is oriented from north to south, with a recharge elevation estimated at around 2350 meters above sea level.

Seasonal variations include:

- Minimums: Recorded in the spring months (May-June), when precipitation contributions are limited, and snowmelt is not yet sufficient.
- Maximums: Observed between August and September, corresponding to the peak of snowmelt at high altitudes.

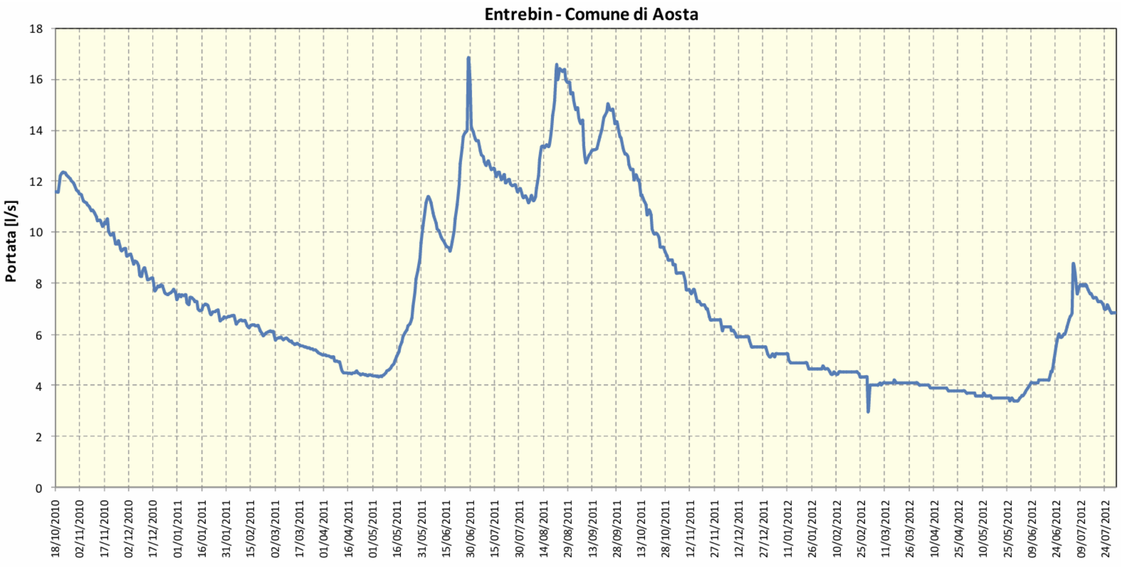


Figure 8 - Discharge over time at Entrebin Spring (Ao), 2010-2012.

Climate and Seasonality

The region's climate is alpine, characterised by cold, snowy winters and cool summers with moderate precipitation. Temperatures significantly influence the spring's hydrological cycle:

- During the winter months, snowfall stores water in solid form, reducing the immediate inflow to the aquifer.
- In the summer months, rising temperatures promote snowmelt, contributing to aquifer recharge.

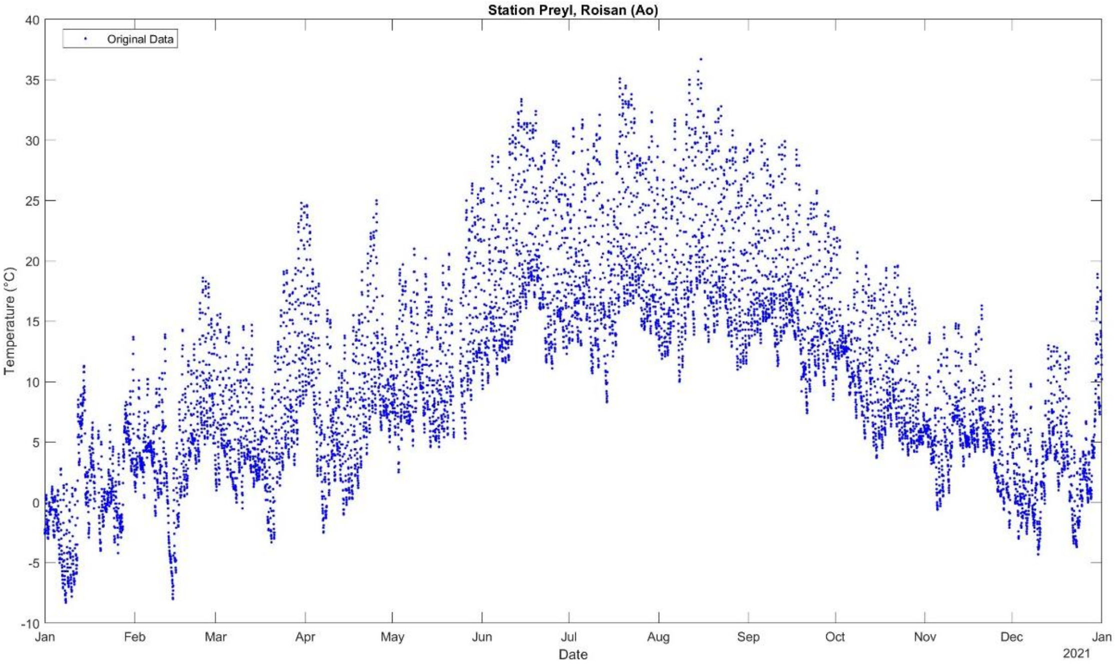


Figure 9 - Temperature over time at Entrebin Spring (Ao), 2021.

Influencing Climatic Factors

In addition to seasonality, extreme phenomena such as prolonged droughts or intense rainfall events can alter the spring's hydrological regime. For example:

- Prolonged rainfall can temporarily increase the spring's flow but may reduce the efficiency of the capture system due to the increase in debris.
- Drought periods can lead to a reduction in flow to critical levels, putting pressure on Aosta's water supply system.

Datasets Used

The analysis of the Entrebin spring was based on data collected and managed by the DIATI department of Politecnico di Torino. The main datasets include measurements related to water level, temperature, and electrical conductivity of the spring (output data). These were integrated with meteorological data (input) from the Roisan-Preyl weather station, located approximately 3 km in a straight air line from the spring. Both data collection areas are within the same watershed, ensuring suitable representation of the climatic phenomena affecting the spring.

The collected data were used to study hydrological dynamics and verify the impact of climatic conditions on the spring's parameters. However, certain limitations in the datasets required the adoption of specific attention to ensure the quality and reliability of the analyses.

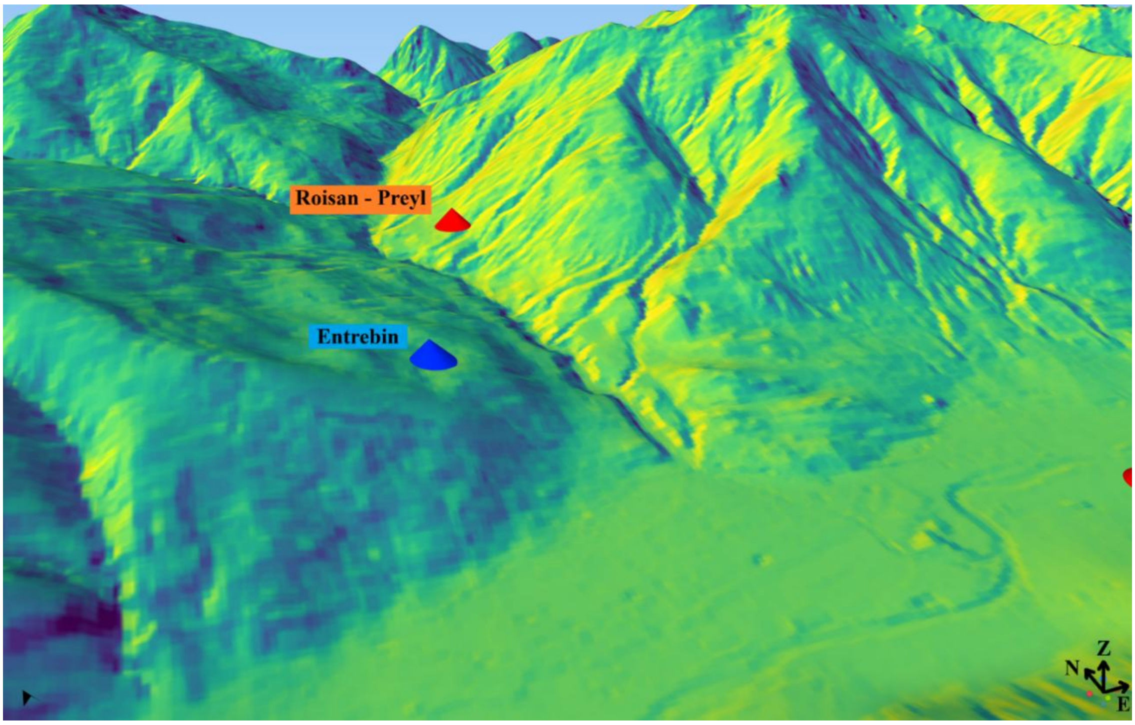


Figure 10 - Digital Terrain Model (DTM) showing the spatial location of Entrebin and Roisan-Preyl.

Water Level Dataset (Output)

The intake work of the Entrebin spring is closely related to monitoring the data on water level, temperature and electrical conductivity. The structure of the drainage gallery, through two channels, communicates with two tanks equipped with overflow pipes and weirs, ensuring a constant water flow. In one of these tanks, a measurement instrument, the OTT-CTD probe, was installed on October 18, 2010, and positioned 150 cm from the main triangular weir. The probe has an hourly acquisition system (one data point every 60 minutes). The overflows and weirs minimize sudden variations that could compromise the quality of the measurements.

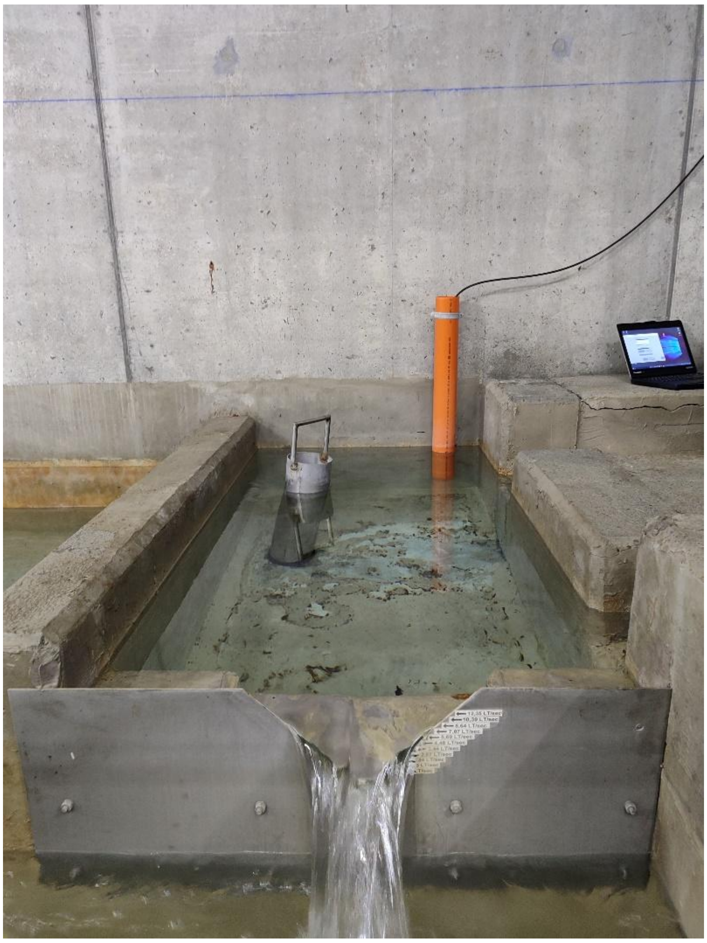


Figure 11 - Monitoring setup at the spring, where it visible a basin equipped with a weir for flow measurement and the sensor installed.

Technical characteristics of the instrumentation

The OTT-CTD probe measures:

- Water level with an accuracy of ±0.05% of full scale, using a ceramic membrane sensor.
- Water temperature with a resolution of 0.01°C and an accuracy of ±0.1°C.
- Electrical conductivity in a range from 0.001 to 100 mS/cm, with an accuracy varying from ±0.5% to ±1.5% of the measured value.

Initial dataset problems

The dataset showed some initial critical issues:

- Instrumental noise: In certain periods, anomalies in the data were observed, probably due to interference or the need for instrument calibration.
- Data gaps: Temporary instrument failures (although extreme weather events cannot be ruled out) led to the loss of some recordings. Notably, there is a lack of data for approximately two years in 2014 and 2015. It was decided not to integrate the data using interpolation techniques, thus only original data were used.
- Clogging in the drains: The old age of the gallery (built in 1930) and the presence of clogs in the drains are critical factors that can affect the accuracy of the collected data. The work requires constant maintenance to ensure the proper functioning of the instruments and the continuity of the measurements.

Precipitation and Temperature Dataset (Input)

The meteorological data used as input for the study comes from the Roisan-Preyl weather station, managed by the Centro Funzionale Regione Autonoma Valle d'Aosta. It is located at an altitude of 935 meters above sea level and approximately 3 km in a straight line from the Entrebin spring. Both areas are within the same water catchment area, specifically the Buthier stream basin, ensuring adequate representation of the climatic phenomena affecting the spring.

Geographical coordinates:

- Latitude: 45.782° N
- Longitude: 7.317° E

Station Characteristics

The Roisan-Preyl station is equipped to measure the following parameters with the following sensors:

- Precipitation: Tipping bucket rain gauge (non-heated).
- Temperature: Digital sensors for air monitoring.
- Relative humidity, wind speed, and direction.

These data are recorded by the station every ten minutes and then made accessible with hourly resolution. This ensures continuous and detailed monitoring of atmospheric variables.

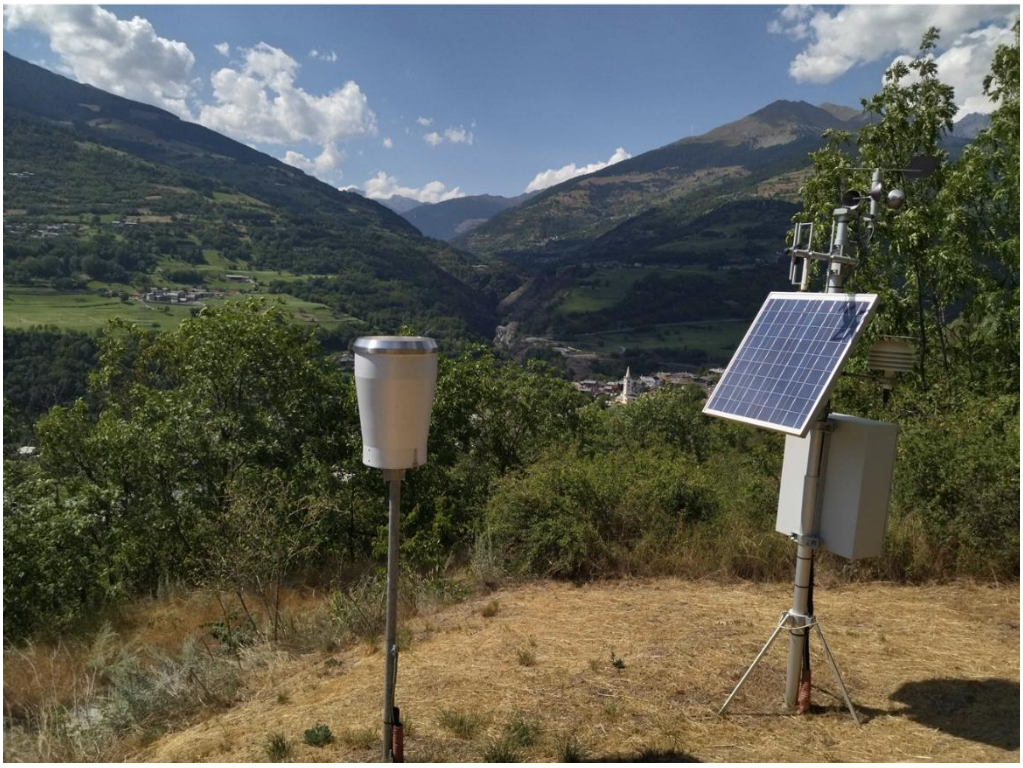


Figure 12 - Weather station of Roisan-Preyl equipped with a rain gauge for precipitation monitoring (on the left).

Limitations and Issues

Despite the suitability of the position relative to the Entrebin spring, some limitations have been identified:

- Non-heated rain gauge: As noted in the station's profile, the lack of heating in rain gauge poses problems during the winter months. Accumulated snow on the sensor can melt afterwards, causing precipitation records that do not correspond to the actual event. This phenomenon can distort precipitation data, especially during periods characterized by temperatures fluctuating around 0°C.
- Winter period issues: During the winter months, particularly from late November to late February, recurring malfunctions of the measurement instruments at the weather station are observed. This issue is especially evident in the precipitation data, likely due to snow accumulation on the non-heated rain gauge, generating erroneous or delayed readings related to snowmelt during cold periods, but also due to the malfunction of the rain gauge itself. On the other hand, for the temperature dataset, anomalies are much more contained, limited to a few nighttime hours on specific winter days.
- Spatial discrepancies: Although the station and the spring belong to the same hydrographic basin, local microclimates and altimetric differences (the station is located approximately 50 meters lower than the spring) could introduce variations compared to the actual meteorological conditions of the study site.
- Temporal discrepancies in data: The availability of data presents significant temporal gaps. For meteorological input data, there are no usable recordings before 2014. For output data related to the spring, there is an interruption of approximately two years, corresponding to the years 2014 and 2015. Therefore, the first data usable for cross-correlation analysis are those starting from 2016. Considering that the analysis could have started in 2011, the absence of usable data for a period of five years resulted in the loss of almost 40% of the originally planned dataset, thus limiting the completeness and reliability of the analyses.

Data Usage

The meteorological data were integrated with those related to the spring to analyze:

- The correlation between precipitation and water levels, to evaluate the aquifer's response times to rainfall events.
- The seasonal and annual variations in air temperature, in relation to water recharge and the behavior of the spring system.

Sources and Data Quality

The analysis of the Entrebin spring is based on data from two main sources:

1. Output data: Provided by direct monitoring of the spring using the OTT-CTD probe, installed and managed by the Politecnico di Torino.
2. Input data: Meteorological data collected from the Roisan-Preyl station, managed by the Region of Valle d'Aosta.

These sources provide a detailed picture of the hydrological and climatic conditions of the spring. However, there are some limitations related to the quality, consistency, and temporal coverage of the data, which required specific methodological interventions to optimize the analyses.

Data Quality

- Output data: The measurements taken by the OTT-CTD probe show a reasonable level of accuracy for the main monitored parameters: water level, temperature, and electrical conductivity. However, some issues have emerged:
  - Presence of anomalous measurement series, attributed to temporary malfunctions of the probe or particularly adverse environmental conditions.
  - Need for regular calibration to maintain the reliability of the readings.

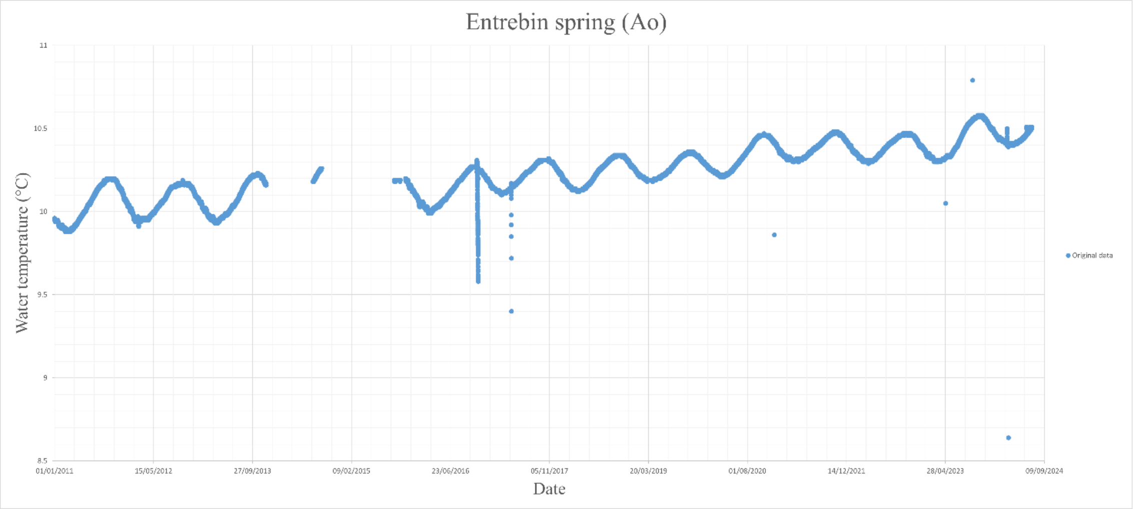


Figure 13 - Water temperature over time for the entire historical dataset at Entrebin.

- Input data: The meteorological data from the Roisan-Preyl station suffer from instrumentation-related limitations:
  - The non-heated rain gauge results in measurement errors during the winter months, with delayed or overestimated precipitation readings due to snowmelt.
  - The air temperature sensors have shown high reliability, with very limited anomalies confined to nighttime hours on certain winter days.

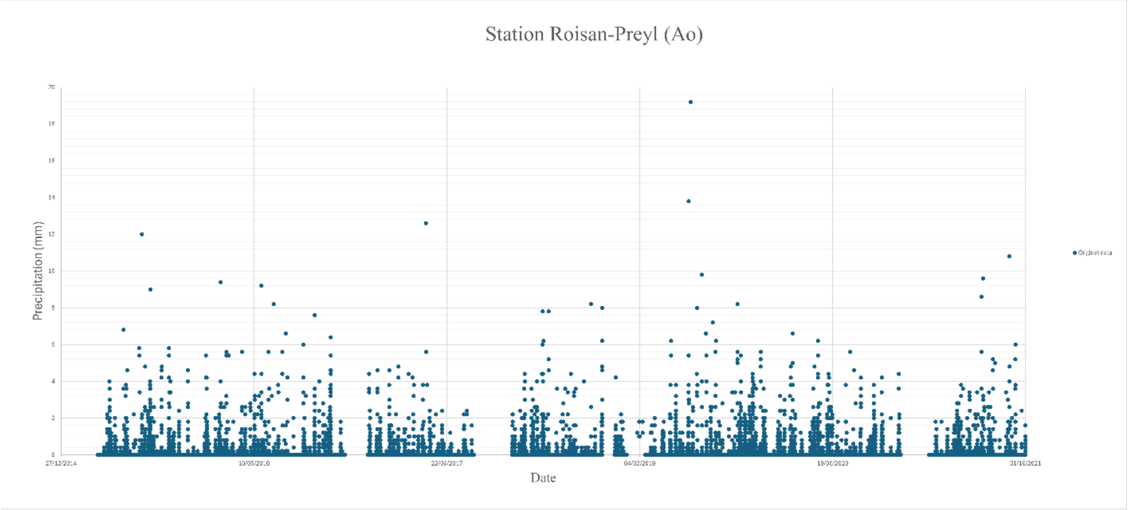


Figure 14 - Precipitation over time from 2015 at Roisan-Preyl.

Data Consistency

- The consistency between input and output datasets was verified over specific periods. The distance between the two stations, about 3 km, could introduce discrepancies.
- Any anomalies in the data were managed:
    - Correction using the Kalman filter for parameters of level, temperatures (water and air), and electrical conductivity.
    - Exclusion of incomplete or unreliable data, such as for the year 2014 in the output data.

Temporal Coverage

- Input data: Available from 2014, with occasional interruptions during the winter months due to the limitations of the rain gauge.
- Output data: Significant temporal gap for the years 2014 and 2015. Consequently, the usable dataset was divided into two main periods:
    - 2011-2013

  - 2016-2023

This temporal fragmentation reduced the amount of data available for long-term analysis and cross-correlation studies between input and output. It also limited the ability to make reliable future forecasts based on an extensive time series.

Observations

Despite the described limitations, the collected datasets provide a solid foundation for studying the hydrological dynamics of the Entrebin spring. To improve the reliability and quality of future analyses, it may be useful to:

- Update the meteorological instrumentation, for example by installing a heated rain gauge.
- Integrate missing data using advanced modeling techniques that consider the correlations observed in complete periods.

Preprocessing and data cleaning

The use of the Kalman filter represents a crucial step in data processing, as it allows for a more reliable representation of the studied phenomenon. This approach integrates perfectly with the data preparation work described below, as pretreated data provides a more solid foundation for the application of the filter.

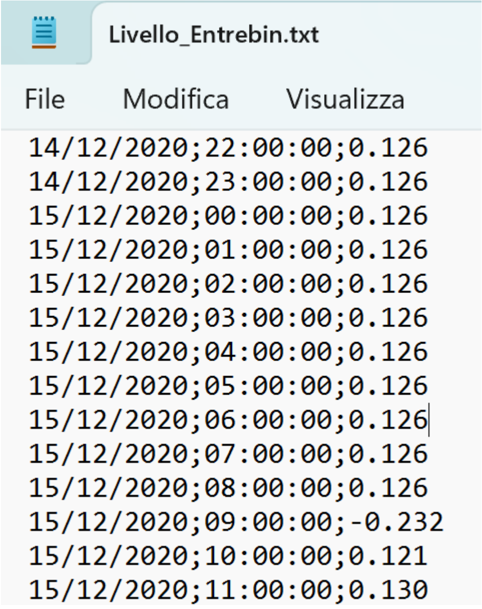
This is a recursive algorithm widely used to estimate the state of a dynamical system in the presence of noise. Its ability to combine noisy measurements with a predictive model makes it particularly suitable for processing temporal data, such as those for the water level of the source in question.

The theoretical operation of the Kalman filter and how it has been applied to datasets to obtain filtered curves is explained. Next, the fundamental process of calibrating the parameters Q (process noise) and R (measurement noise), which are the keys to optimizing filter performance, is described.

Data preparation

The first thing to do was to import to MATLAB data from a text file (esempio: Livello_Entrebin) containing temporal measurements. The file was read using MATLAB's *readtable* function, specifying the delimiter (;) and column format:

- The first column contains dates in dd/MM/yyyy format.
- The second column contains the times in HH:mm:ss format.
- The third column contains the measurement values.



```
14/12/2020;22:00:00;0.126
14/12/2020;23:00:00;0.126
15/12/2020;00:00:00;0.126
15/12/2020;01:00:00;0.126
15/12/2020;02:00:00;0.126
15/12/2020;03:00:00;0.126
15/12/2020;04:00:00;0.126
15/12/2020;05:00:00;0.126
15/12/2020;06:00:00;0.126
15/12/2020;07:00:00;0.126
15/12/2020;08:00:00;0.126
15/12/2020;09:00:00;-0.232
15/12/2020;10:00:00;0.121
15/12/2020;11:00:00;0.130
```

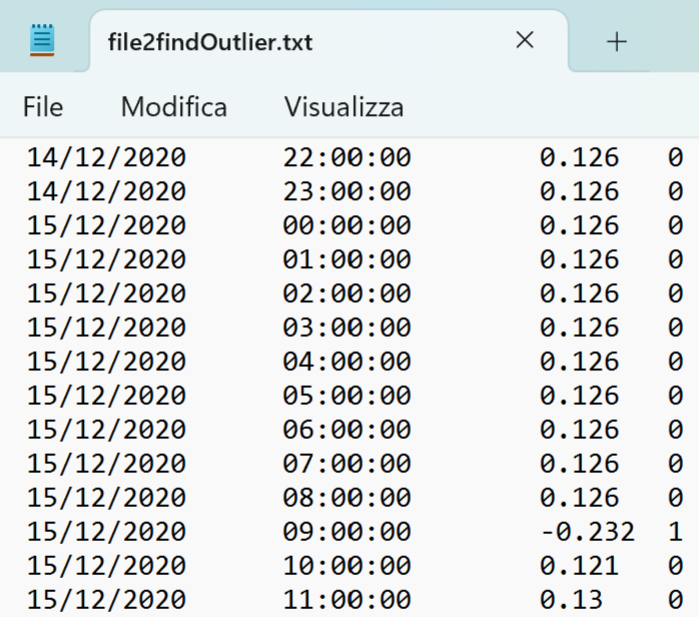
Figure 15 - Example of the raw dataset. Screenshot of the text file opened showing some rows of data.

This operation created a table (data) with three columns: date, time, and measurement value.

The data preparation and cleaning phase was divided into three subphases: identification of non-working periods (No_work), identification of missing or incorrect measurements (No_meas), and removal of incorrect data.

To identify periods when the instrumentation failed, a complete time vector (time_range) was created with one-hour intervals (time_interval) between the first and last available timestamps. Next, this vector was compared with the actual timestamps in the data. If missing timestamps were detected, they were printed on the screen, indicating periods of instrumentation inactivity, to allow the operator to verify the reasons why the instrumentation failed to measure and/or communicate data.

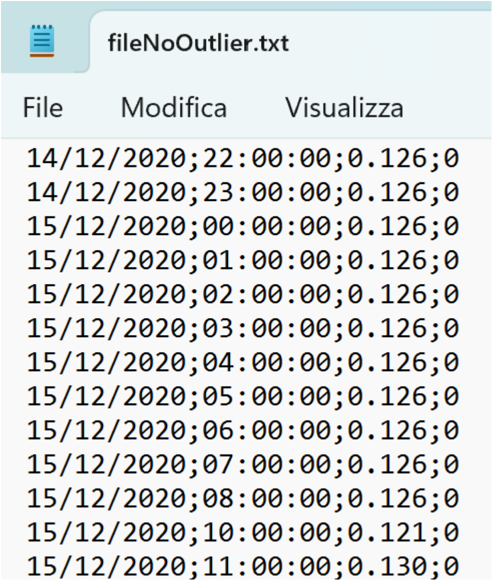
To identify missing or incorrect measurements (in the case of waterlevel calculation, those measurements less than 0 are incorrect), a control column (Var4) was added to the data table. This column was set to 1 if the measurement value was NaN or negative. The data were then saved in a new text file (file2findOutlier) for further analysis, that is, to go in and remove rows with missing or incorrect data.



```
14/12/2020    22:00:00    0.126   0
14/12/2020    23:00:00    0.126   0
15/12/2020    00:00:00    0.126   0
15/12/2020    01:00:00    0.126   0
15/12/2020    02:00:00    0.126   0
15/12/2020    03:00:00    0.126   0
15/12/2020    04:00:00    0.126   0
15/12/2020    05:00:00    0.126   0
15/12/2020    06:00:00    0.126   0
15/12/2020    07:00:00    0.126   0
15/12/2020    08:00:00    0.126   0
15/12/2020    09:00:00    -0.232  1
15/12/2020    10:00:00    0.121   0
15/12/2020    11:00:00    0.13    0
```

Figure 16 - Example of the dataset with control column, text file opened showing some rows of data.

Finally, the data were filtered to remove rows with erroneous values (those with Var4 = 1). Valid rows were saved in a new text file (fileNoOutlier).



```
14/12/2020;22:00:00;0.126;0
14/12/2020;23:00:00;0.126;0
15/12/2020;00:00:00;0.126;0
15/12/2020;01:00:00;0.126;0
15/12/2020;02:00:00;0.126;0
15/12/2020;03:00:00;0.126;0
15/12/2020;04:00:00;0.126;0
15/12/2020;05:00:00;0.126;0
15/12/2020;06:00:00;0.126;0
15/12/2020;07:00:00;0.126;0
15/12/2020;08:00:00;0.126;0
15/12/2020;10:00:00;0.121;0
15/12/2020;11:00:00;0.130;0
```

Figure 17 - Example of the dataset with only valid data, text file opened showing some rows of data.

All this step is essential to ensure the reliability of the data and the correctness of the conclusions drawn from the analysis.

Kalman filter: theory and application

The Kalman filter is a recursive algorithm used to estimate the state of a dynamic system in the presence of noise. It relies on a predictive model that combines information from a theoretical system model (process noise) with experimental observations (measurement noise).

The algorithm operates in two main phases: prediction and correction.

In the first phase, prediction, the system state is estimated using a dynamic model. In our case, the model is simple, with a transition matrix $A = 1$, which assumes that the system state remains constant between one measurement and the next. The estimate of the prediction error covariance is updated by taking into account the process noise Q.

In the correction phase, the predicted estimate is updated using the new measurement. The innovation, which is the difference between the observed measurement and the predicted estimate, is weighted based on the Kalman gain, which depends on the uncertainties of the model (Q) and the measurement (R).

In the MATLAB script, the process begins by importing data from a text file (fileNoOutlier), which contains the preprocessed data. The data is organized into a table with three columns: date, time, and measurement.

At this point, the data needs to be prepared. First, the date and time columns are converted into a single *datetime* vector (dateTimeArray). Next, a list of unique years present in the data is prepared. For the Entrebin data, the years 2014, 2015, and 2024 are excluded from the analysis because they are mostly incomplete and therefore not useful for the analysis being conducted.

Now, the parameters of the Kalman filter for the iterations over each year are defined. In this phase, the state transition matrix A, the observation matrix $H = 1$ (this value is chosen because the measurements are assumed to be directly related to the state), the initial covariance estimate $(initialCovarianceEstimate = 10^{-6})$, the process noise variance Q (related to the uncertainty of the model), and the measurement noise variance R (related to the uncertainty of the observations) are set. The chosen values for the latter two parameters are discussed in detail in the following section.

The Kalman filter is then applied separately for each year. The last parameter that needs to be initialized for each year is set in this part, namely the initial state estimate ($stateEstimate$). For this, the first valid measurement of the year under consideration is used.

The next part of the code involves the actual application of the **prediction** and **correction** phases. Since this is the core of the Kalman filter, this part is explained in even greater detail. The prediction phase projects the state and its uncertainty into "what happens next" – we could define this as the future – while the correction phase updates these estimates using the new measurements. As a final result, a more accurate and reliable estimate of the input measurements is obtained, along with a quantification of the associated uncertainty.

In detail:

$$predictedStateEstimate = A * stateEstimate$$

The predicted state estimate at the next step ($predictedStateEstimate$) is calculated as the product of the state transition matrix (A), which, as mentioned earlier, assumes that the system state remains constant between one measurement and the next, and the current state estimate ($stateEstimate$).

$$predictedCovarianceEstimate = A * covarianceEstimate * A' + Q$$

At this point, the predicted covariance of the estimation error ($predictedCovarianceEstimate$) is computed. This takes into account the current estimation error covariance ($covarianceEstimate$), which represents the uncertainty associated with the state estimate, and the process noise (Q), which represents the uncertainty associated with the dynamic model. This concludes the prediction phase.

$$innovation = yearValues(t) - H * predictedStateEstimate$$

In the correction phase, where the predicted estimate is updated using the actual observed measurements, the first step is to calculate the innovation ($innovation$), which is the information the filter will use to correct the estimate. The innovation is the difference between the observed measurement at time *t* ($yearValues(t)$) and the product of the observation matrix (H), which assumes that the measurement is directly related to the state, and the predicted state estimate ($predictedStateEstimate$).

$$innovationCovariance = H * predictedCovarianceEstimate * H' + R$$

Next, the covariance of the innovation is calculated. This combines the uncertainty of the predicted estimate, i.e., the predicted covariance of the estimation error ($predictedCovarianceEstimate$), and the measurement uncertainty, i.e., the measurement noise variance (R), which represents the uncertainty associated with the observations.

$$kalmanGain = predictedCovarianceEstimate * H' \,/\, innovationCovariance$$

The Kalman gain ($kalmanGain$) is then computed, which determines how much weight to give to the innovation relative to the predicted estimate. Therefore, if the measurement uncertainty (R) is much larger than the uncertainty of the predicted estimate, the Kalman gain will be small, and, as a direct consequence, the filter will give less weight to the new measurement.

$$stateEstimate = predictedStateEstimate + kalmanGain * innovation$$

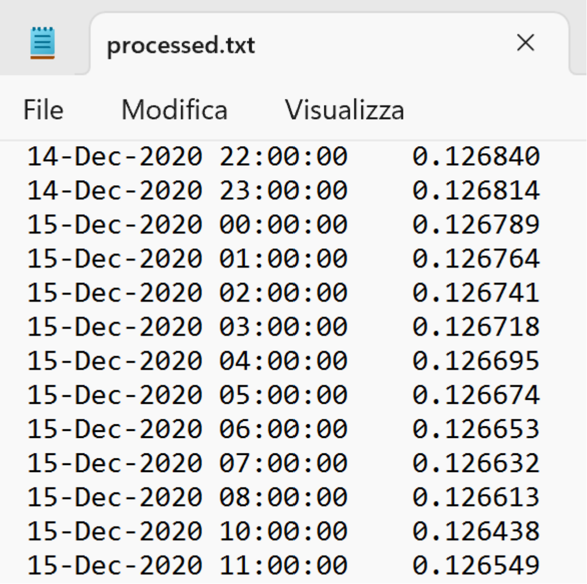
Finally, the updated state estimate ($stateEstimate$) is obtained by correcting the predicted estimate ($predictedStateEstimate$) with the innovation ($innovation$), weighted by the Kalman gain ($kalmanGain$).

$$\begin{aligned} &covarianceEstimate \\ &\qquad = (1 - kalmanGain * H) * predictedCovarianceEstimate \end{aligned}$$

Consequently, the estimation error covariance ($covarianceEstimate$) is also updated, representing the uncertainty associated with the new state estimate. In this calculation, the uncertainty ($predictedCovarianceEstimate$) is reduced based on the Kalman gain ($kalmanGain$) and the observation matrix (H).

Once the iteration over the entire dataset is completed, the results are visualized graphically, one plot for each year considered. This allows for a preliminary comparison between the original data and the data obtained through Kalman estimation.

Finally, the Kalman estimates are saved in a text file (processed). This becomes the new input dataset, cleaned of noise and outliers, for the evaluations that will be carried out in the continuation of the study.



```
14-Dec-2020 22:00:00    0.126840
14-Dec-2020 23:00:00    0.126814
15-Dec-2020 00:00:00    0.126789
15-Dec-2020 01:00:00    0.126764
15-Dec-2020 02:00:00    0.126741
15-Dec-2020 03:00:00    0.126718
15-Dec-2020 04:00:00    0.126695
15-Dec-2020 05:00:00    0.126674
15-Dec-2020 06:00:00    0.126653
15-Dec-2020 07:00:00    0.126632
15-Dec-2020 08:00:00    0.126613
15-Dec-2020 10:00:00    0.126438
15-Dec-2020 11:00:00    0.126549
```

Figure 18 - Example of the processed data after the use of Kalman filter, text file opened showing some rows of data.

Parameters used for Q and R

The choice of parameters related to the process noise (Q) and the measurement noise (R) is crucial for the correct functioning of the Kalman filter. These parameters directly influence the behavior of the filter, determining the relative weight between the predictive model and the observed measurements.

The process noise represents the uncertainty associated with the dynamic model of the system. In the case of water level, a very small value $(Q = 10^{-9})$ was chosen, as it is assumed that the water level varies slowly and that the predictive model is highly reliable. A small value of Q implies that the filter will give more weight to the observations than to the predictions of the model.

The measurement noise represents the uncertainty associated with the measurements. In the case of water level, a value $(R = 0.001^2)$ was chosen, reflecting the precision of the OTT-CTD measuring instrument. A larger value of R would indicate greater uncertainty in the measurements, pushing the filter to give more weight to the predictive model.

A calibration of Q and R was performed by testing different combinations of values to evaluate their effect on the filter's behaviour. The goal was to find a balance between the filter's ability to quickly follow variations in the signal, which we can define as its responsiveness, and the reduction of noisy signals, thus achieving stability.

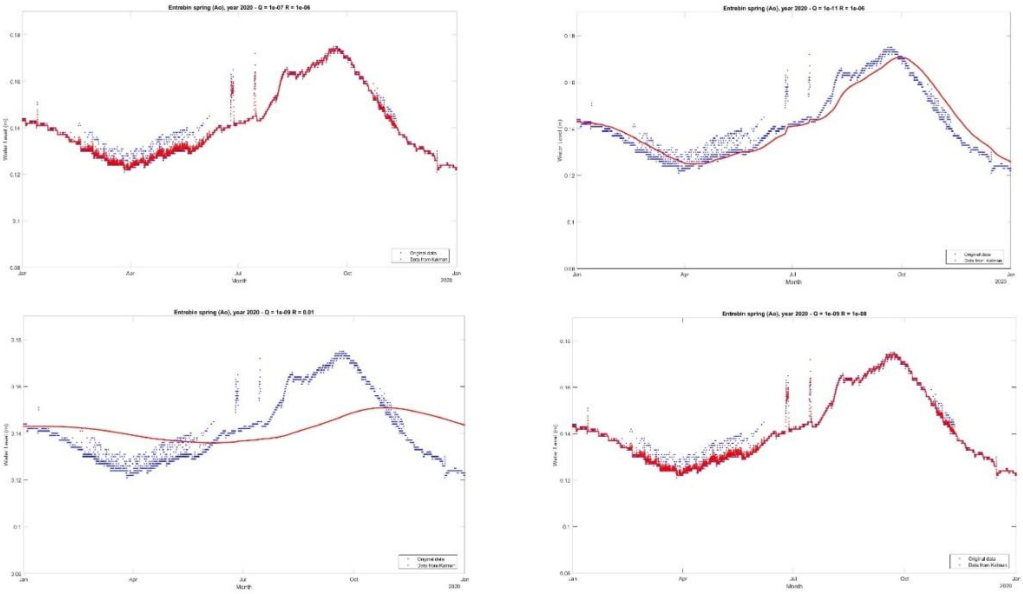


Figure 19 - Example of four Kalman filter models using unsuitable Q and R parameters, resulting in unstable and unacceptable solutions.

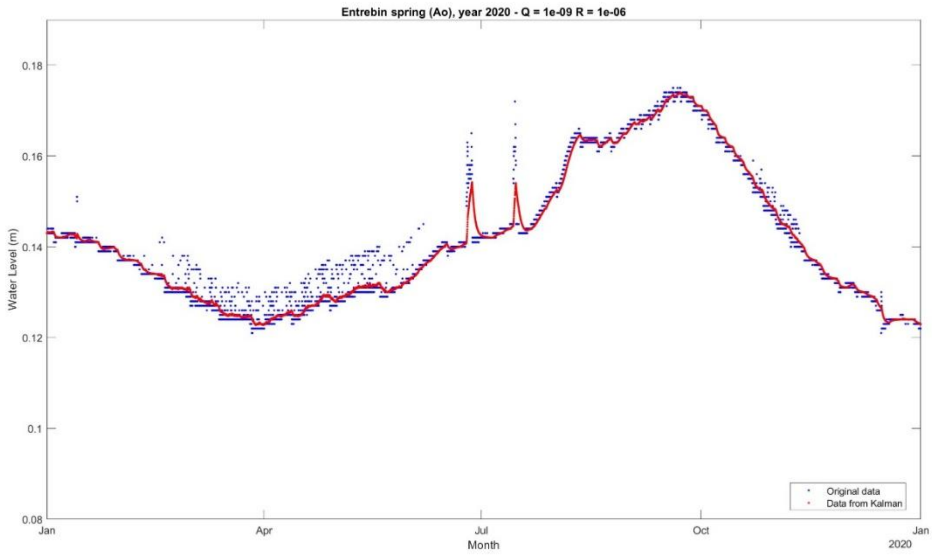


Figure 20 - Example for water level at Entrebin where Kalman filter uses acceptable Q and R parameters.

Mathematical Analysis

At this point, the mathematical models developed to analyze the filtered data related to the water level (and, in general, the output data) are described.

The objective is to use mathematical functions, such as polynomials and Fourier functions, to describe the behaviour of the data and apply these models to the filtered datasets, discussing the results obtained.

In particular, the models were applied to the water level data, and the example images shown in this section focus exclusively on this application.

Mathematical models for the Output datasets

To describe the behaviour of the output data, after some initial testing, the fitting options available in MATLAB involving polynomials of degree n and Fourier functions were chosen.

Polynomials were used in the analysis of data divided by solar year of interest. They were selected for their flexibility in modelling nonlinear relationships between variables.

On the other hand, Fourier functions were chosen to derive a single equation for the entire dataset of interest (usable). These types of functions were employed to capture periodic components in the data, such as seasonal fluctuations.

These initial analyses and models were applied to the water level data to identify trends and predict possible future behaviours, with the intention of testing them on future data to verify their validity.

The MATLAB code used for the analysis begins by reading the filtered data from the previous tabular text file (*processed*), where the columns represent the dates and the associated water level values, respectively. The data is then organized into a *datetime* array to facilitate temporal manipulation.

Subsequently, the script extracts the unique years present in the data and defines the y-axis limits for the graphs. These limits are useful for achieving a good degree of uniformity in the output graphs and must be defined according to the type of dataset being used. Indeed, these limits can be manually adjusted based on the variable being analysed.

Application to Filtered data by solar Year

As mentioned earlier, the analysis was initially divided into two main phases. In the first phase, an annual analysis was conducted. For each solar year, a polynomial fit of degree *n* was applied to the output data to define equations that describe the behavior of the variables over the course of the year.

The choice of the polynomial degree for fitting is a crucial aspect. In this case, examples related to the analysis of water level data are presented. A polynomial of degree 5 was selected as the optimal model to describe the behaviour of the data, while polynomials of degree 3 and 7 were discarded as they were less suitable for capturing the characteristics of the data. Below, the reasoning behind this choice is explained in detail.

The polynomial of degree 5 offers a good balance of flexibility and complexity. It is sufficiently complex to capture nonlinear variations in the water level data, such as seasonal peaks, but not excessively complex to cause what is known as overfitting, i.e., an excessive adaptation to the data, which reduces the model's predictive capability and goes against the goal of finding a general model. It is therefore very important to strike a balance between overfitting and what we might define as underfitting.

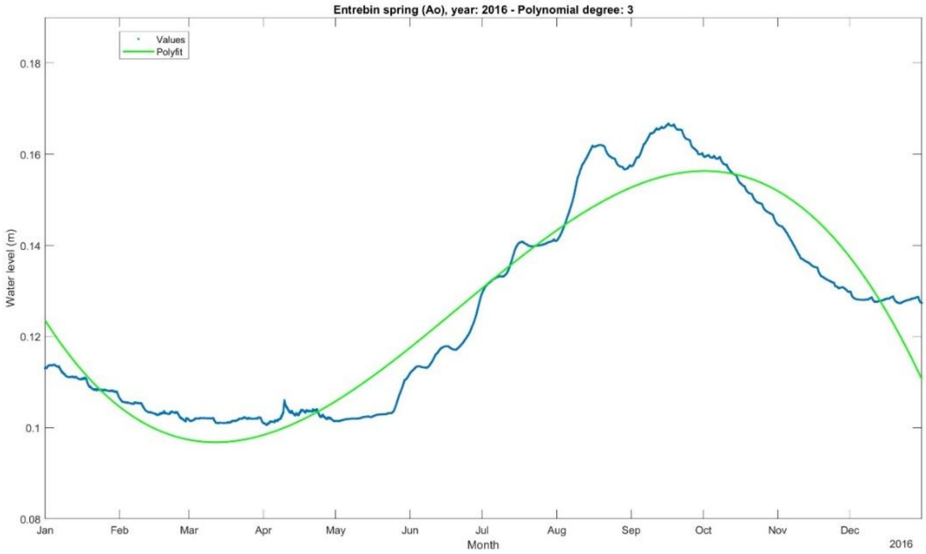


Figure 21 - Fit on filtered water level data using a third-degree polynomial equation for Entrebin, calendar year 2016.

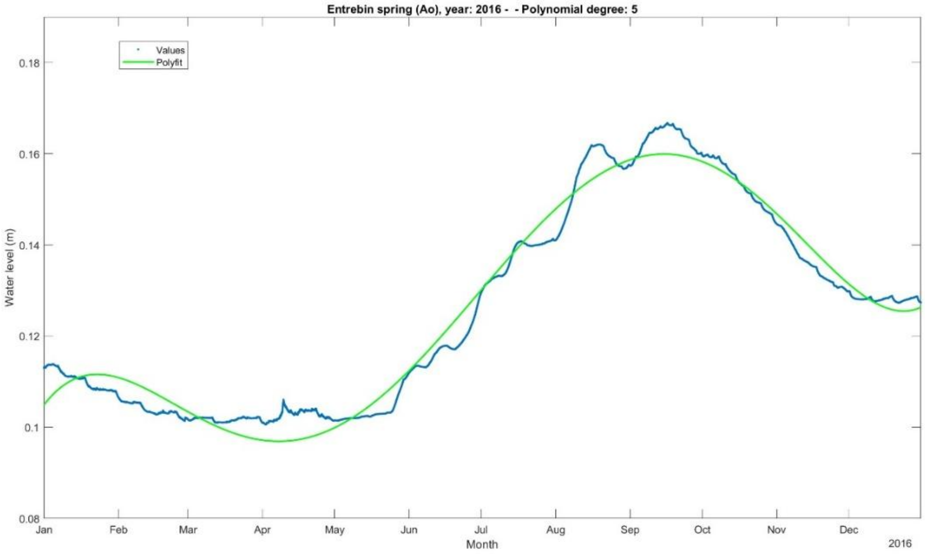


Figure 22 - Fit on filtered water level data using a fifth-degree polynomial equation for Entrebin, calendar year 2016.

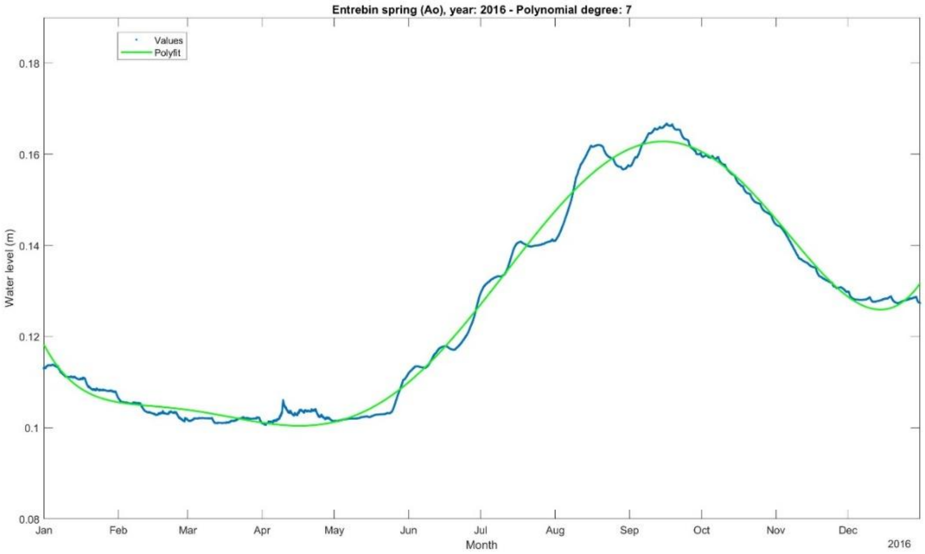
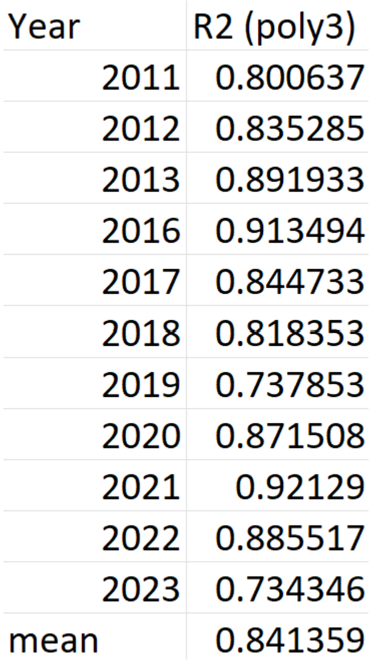


Figure 23 - Fit on filtered water level data using a seventh-degree polynomial equation for Entrebin, calendar year 2016.

From the first image, it can be observed that a polynomial of degree 3 is still too simple to describe the water level data. Indeed the cubic model may fail to adequately capture seasonal variations or the peaks observed in the data, leading to underfitting.

From the third image, particularly in the first months of the analysed year, the polynomial of degree 7 appears too complex. While it fits the existing data very well, it risks capturing noise and random fluctuations, resulting in overfitting. This would reduce the model's ability to generalize and correctly predict future behaviour.

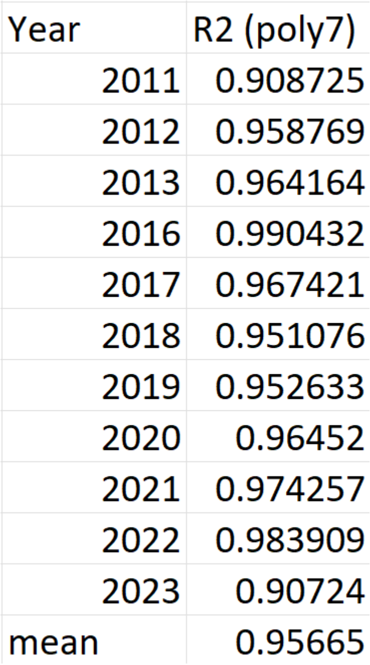
In addition to the graphical analysis, a comparison was also made based on the values of the coefficient of determination R^2 obtained during the tests. R^2 measures how well the model fits the data, so a value close to 1 indicates an almost perfect fit. However, caution is required, as a very high value is not always indicative of a better model, especially if it comes at the disadvantage of the model's generalization capability.



Year	R2 (poly3)
2011	0.800637
2012	0.835285
2013	0.891933
2016	0.913494
2017	0.844733
2018	0.818353
2019	0.737853
2020	0.871508
2021	0.92129
2022	0.885517
2023	0.734346
mean	0.841359

Figure 24 - Values of coefficient of determination for the third-degree polynomial equations for Entrebin, considering all the calendar year available.

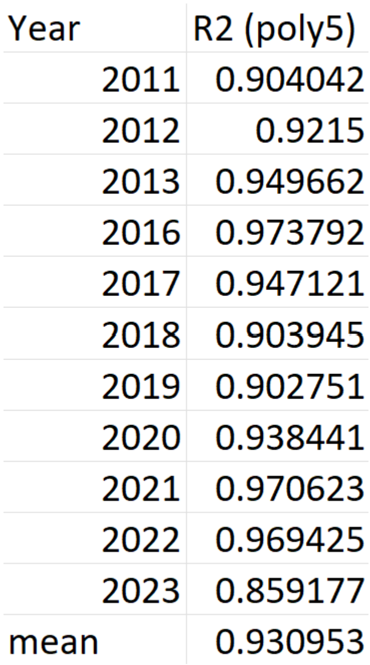
For the polynomial of degree 3 the R^2 values are the lowest, which is due to the rigidity of the model. This model often fails to adequately describe the variations in the water level data. In the years 2019 and 2023, the R^2 value is low, indicating that the model is too simple, leading to a loss of important information.



Year	R2 (poly7)
2011	0.908725
2012	0.958769
2013	0.964164
2016	0.990432
2017	0.967421
2018	0.951076
2019	0.952633
2020	0.96452
2021	0.974257
2022	0.983909
2023	0.90724
mean	0.95665

Figure 25 - Values of coefficient of determination for the seventh-degree polynomial equations for Entrebin, considering all the calendar year available.

For the polynomial of degree 7 the R^2 values are very high, and at first glance, they appear to be the most suitable. However, this comes with the risk of overfitting, as the model adapts excessively to the data and may even capture noise.



Year	R2 (poly5)
2011	0.904042
2012	0.9215
2013	0.949662
2016	0.973792
2017	0.947121
2018	0.903945
2019	0.902751
2020	0.938441
2021	0.970623
2022	0.969425
2023	0.859177
mean	0.930953

Figure 26 - Values of coefficient of determination for the fifth-degree polynomial equations for Entrebin, considering all the calendar year available.

The R^2 values for the polynomial of degree 5 are all above 0.85, which is very high. They are higher than those obtained with the polynomial of degree 3 and, on average, slightly lower than those of degree 7. The difference with the latter is minimal, and the degree 5 polynomial offers a better balance between data fitting and predictive capability. This confirms that the polynomial of degree 5 is the most suitable model for describing the water level data, precisely due to its good generalisation ability.

Application to the Completed filtered dataset starting from 2016

In the second phase, the analysis focused on the complete time series. The available data starts from 2011 but considering that the years 2014 and 2015 lacked data, it was decided to begin from 2016. The goal was to identify a periodic pattern and attempt to predict the behaviour of the spring system, for example, in the following year.

For this part, the Fourier function was tested using MATLAB's *Curve Fitter* tool, along with models based on the Sum of Sine of degree 3 and degree 6. In the end the latter were discarded as they provided unsatisfactory results. Below are some of the reasons why the Fourier function was chosen as the preferred model.

The Fourier function was selected as the ideal model for its ability to capture periodic components in the data, such as seasonal fluctuations, without encountering the issues observed with the Sum of Sine. Specifically, the chosen model is particularly suited for describing data with periodic components, such as the seasonal fluctuations present in the water level data. This makes it ideal for this type of data, which exhibits repetitive patterns over time.

Additionally, the Fourier function uses a limited number of parameters to describe the data, making the model simpler and more interpretable compared to the Sum of Sine. Moreover, unlike the Sum of Sine, it tends not to create unrealistic oscillations, making it more suitable for future predictions due to its better generalization capability. Finally, the Fourier function is less sensitive to noise in the data compared to higher-degree Sum of Sine models, although this aspect is less critical in this case since the data has already been filtered.

The type of fitting chosen for the function in MATLAB was a second-order Fourier model (fourier2), which is defined as follows:

$$f(x) = a_0 + a_1 cos(x\omega) + b_1 sin(x\omega) + a_2 cos(2x\omega) + b_2 sin(2x\omega)$$

Where a_0 is the constant term, representing the mean value around which the data oscillates.

The terms a_1, b_1, a_2, b_2 are the coefficients of the harmonics. The first harmonic describes the periodic component in the data, such as a seasonal fluctuation, allowing the modelling of periodic variations with any phase. The second harmonic captures faster periodic variations due to its structure, characterized by a frequency double that of the first harmonic (2ω). This makes it useful for identifying semiannual fluctuations and other high-frequency components present in the dataset.

The term ω is the fundamental frequency and determines the periodicity of the function. It is fundamentally linked to the period T of the first harmonic through the relationship $T = \frac{2\pi}{\omega}$. Considering that the value obtained for the water level data is $\omega = 0.017308$, the period of the first harmonic is $T \approx 363\, days$. This corresponds to an annual fluctuation, typical of seasonal data. Therefore, the fundamental frequency defines the time scale of the periodic variations.

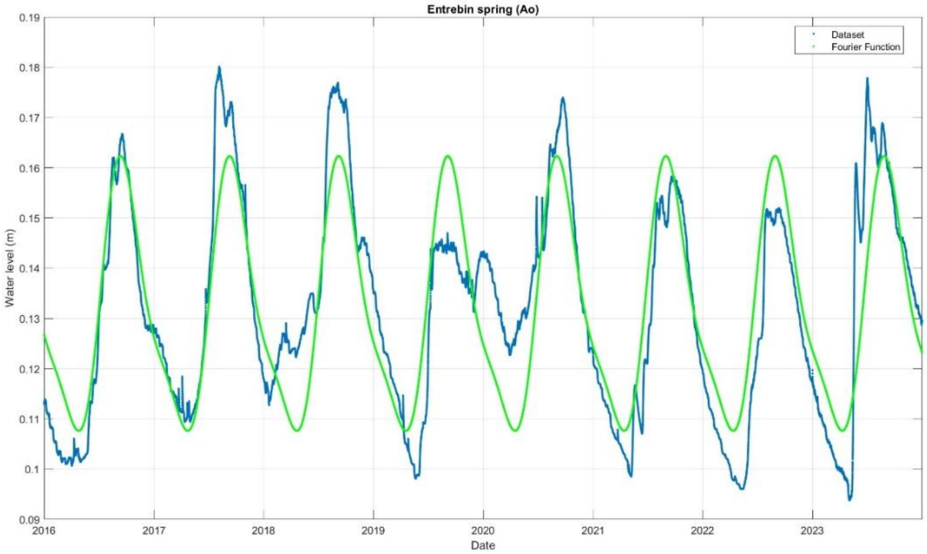


Figure 27 - Fourier of second-order fit applied to the entire historical dataset of filtered water level data for Entrebin, from 2016.

Table 1 - Coefficients of the second-order Fourier function estimated from the water level dataset for Entrebin.

a0	a1	b1	a2	b2	w
0.13195393	0.018048372	0.017595595	0.0052695933	0.0038053172	0.017308

A residual analysis was conducted, which is a fundamental part of evaluating a mathematical model. Residuals represent the difference between the observed values in the data and the values predicted by the model. In other words, residuals measure the model's error for each data point. A thorough analysis of the residuals allows for assessing the goodness of fit and identifying potential issues, such as the presence of outliers or the failure to capture certain patterns in the data.

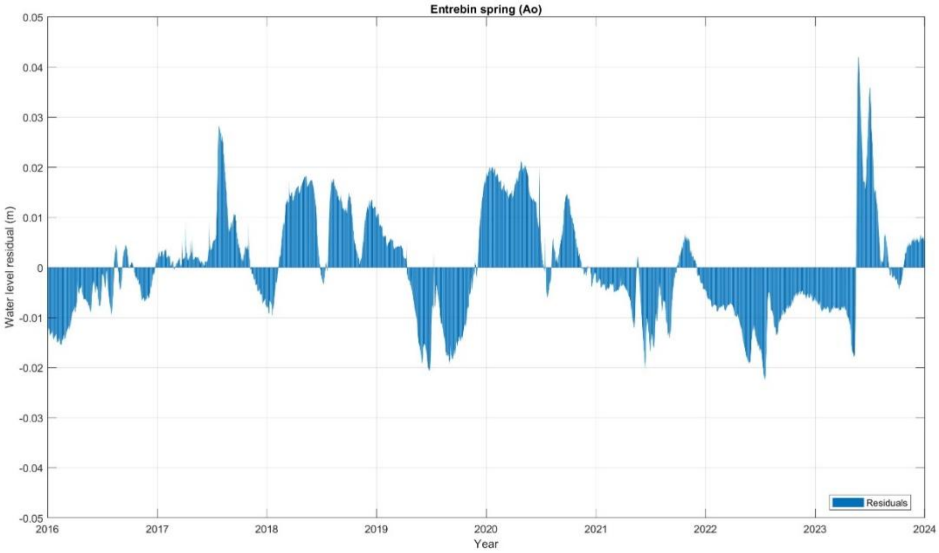


Figure 28 - Residuals from the difference between the model and the original data for the entire historical dataset of filtered water level data for Entrebin, from 2016.

From the bar graph, it can be observed that most of the residuals fall within the range of $\pm 0.02\ m$, indicating a good fit to the model. Of course, there are a couple of years where larger residuals are present, but they do not show a systematic pattern.

This analysis has certainly confirmed that the chosen Fourier model fits the water level data well. The residuals are small and heterogeneously distributed, with no systematic patterns. The model therefore adequately identifies the periodic components of the data, such as seasonal fluctuations, and could be used for future predictions. However, it is always important to continue monitoring the residuals to identify any potential issues with the model and to make further improvements.

Hydrological Analysis and Cross-correlation for the Entrebin spring

The Entrebin spring, located in a complex hydrogeological context and characterized by unconventional dynamics, certainly represents a case study of significant scientific interest.

Different from what is typically expected in other hydrological basins, where the annual cycle follows well-defined seasonal patterns, with autumn recharge and summer recession, Entrebin shows a "phase-shifted" behaviour, in which the phases of spring level recession coincide temporally with periods traditionally associated with recharge.

Also for this reason, the aim is to continue the analysis by attempting to understand the interaction mechanisms between climatic inputs (precipitation and air temperature) and the system's hydrological response (spring water level), as well as to define a methodological framework adaptable to other hydrological contexts.

The study proceeds in two main phases, both implemented again in MATLAB. In the first phase, the dynamic identification of recession periods in the hydrological years is carried out, based on the analysis of level extremes rather than fixed temporal criteria. In the second phase, cross-correlation analysis is performed between time series of precipitation/air temperature (using data from the Roisan-Preyl station published by the Centro Funzionale Regione Autonoma Valle d'Aosta) and water level data of the Entrebin spring, with the intent to determine whether a statistically significant correlation exists and to quantify the time delays (lags) in the hydrological response.

The final objective is double: on one hand, the aim is to identify cause-effect relationships between climatic variables and spring dynamics, with implications for the sustainable management of water resources in climate change scenarios. On the other hand, the results of this analysis are intended to be used to develop a method for predicting future values of air temperature and water levels, which would be crucial in studying future climate change scenarios related to water resources.

Identification of Recession periods in Hydrological years

Unlike previous approaches, where data were analysed in terms of the calendar year, the definition of recession periods in hydrological years represents a fundamental step for the analysis of the dynamics of the Entrebin spring.

This case study required a flexible and adaptive methodology for identifying the recession period, based on the analysis of level extremes. This choice was driven by the peculiarity of the system, which exhibits a "phase-shifted" behaviour compared to theoretical models, where the hydrological year traditionally runs from October 1st to September 30th. This anomaly suggests that delayed recharge mechanisms influence the spring, likely related to geological factors (e.g., low permeability of the terrain) or climatic factors (e.g., late snowmelt).

In MATLAB, the process starts again from the *processed* text file, which is used to import the previously Kalman-filtered level data along with the corresponding dates and times of interest.

The recession period, defined as the time interval between the maximum level of the current year and the minimum level of the following year, is of particular interest because it represents the phase during which the spring gradually loses its water reserve, transitioning from maximum to minimum level conditions.

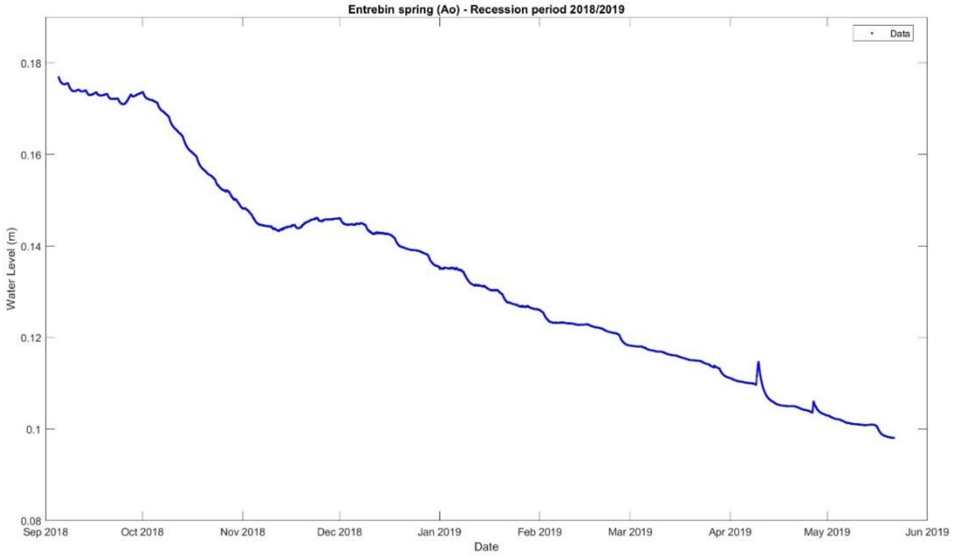


Figure 29 - Recession period (2018–2019) at Entrebin. The graph illustrates the decline in water level, from maximum to minimum, over time during the recession phase.

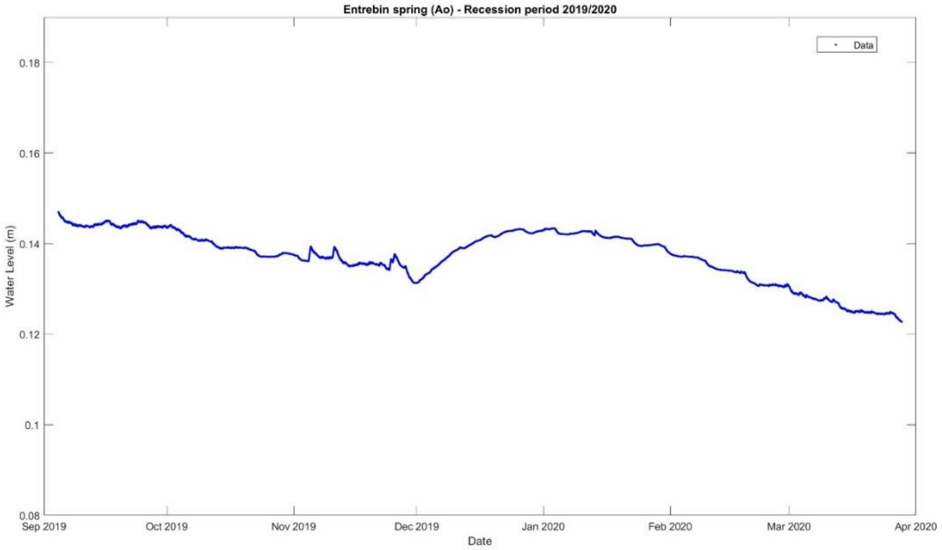


Figure 30 - Recession period (2019–2020) at Entrebin. The graph illustrates the decline in water level, from maximum to minimum, over time during the recession phase.

Cross-correlation analysis between Precipitation, Temperature and Water Level

Cross-correlation analysis represents a powerful statistical tool for quantifying the temporal relationships between climatic variables (precipitation and temperature) and the hydrological response of the Entrebin spring.

However, this approach has proven to be of limited utility, likely due to the specific context under examination, where aquifer dynamics are influenced by complex and not immediately apparent factors. Below, the research conducted is presented.

Cross-correlation is a powerful statistical technique used to measure the similarity between two time series as a function of the time-lag applied to one of them. It is particularly useful in identifying the delay between cause and effect in temporal data. This method is employed to investigate the relationships between meteorological variables (such as precipitation and temperature) and hydrological responses (such as water levels).

The cross-correlation function (CCF) quantifies the correlation between two time series at different lags. It is mathematically represented as follows:

$$CCF(\tau) = \frac{1}{N-\tau}\sum_{t=1}^{N-\tau}\left(\frac{x_t - \mu_x}{\sigma_x}\right)\left(\frac{y_{t+\tau} - \mu_y}{\sigma_y}\right)$$

where:

- (x_t) and (y_t) are the time series.
- (μ_x) and (μ_y) are the means of the time series.
- (σ_x) and (σ_y) are the standard deviations of the time series.
- (τ) is the lag.
- (N) is the length of the time series.

In practice, this means that for each possible time lag, we shift one time series relative to the other and compute the correlation coefficient. The lag that maximizes this coefficient indicates the delay between the input (e.g., precipitation) and the output (e.g., water level). This analysis should help us understand how quickly a system responds to changes in external conditions.

In the MATLAB, two separate analyses were conducted using two different inputs compared to the same output, that is the water level of the Entrebin spring. In the first case, the input used was the daily precipitation data (mm) from the Roisan-Preyl (Ao) station, while in the second case, the air temperature data from the Roisan-Preyl station was used. Below, only the procedure used for the daily precipitation input is analysed, as the one related to temperature is analogous.

It is important to highlight that the water level dataset was adapted. Specifically, from the *processed* text file, the data were extracted, daily averages were calculated and saved in the *processed_daily* text file to allow comparison with the daily precipitation data published by the Centro Funzionale Regione Autonoma Valle d'Aosta.

Therefore, the data used for the cross-correlation analysis include: the Precipitation input, measured daily and acquired from the text file, and the water level output, averaged daily and acquired from the text file.

Before proceeding with the cross-correlation analysis, the datasets were normalized to remove any scale differences between the variables, as they are heterogeneous datasets. This step is crucial because we are comparing variables with different scales and orders of magnitude. Without this operation, the higher values would dominate the correlation calculation, making it difficult to identify significant relationships with variables on a smaller scale. Normalization was performed by subtracting the mean and dividing by the standard deviation, resulting in time series with zero mean and variance equal to 1. This guarantees more reliable and interpretable results for the subsequent phase.

The cross-correlation was calculated using MATLAB's $xcorr$ function, which quantifies the similarity between two time series as a function of the time delay (lag). The normalized option was used to obtain correlation coefficients ranging between -1 and 1, facilitating the interpretation of the results.

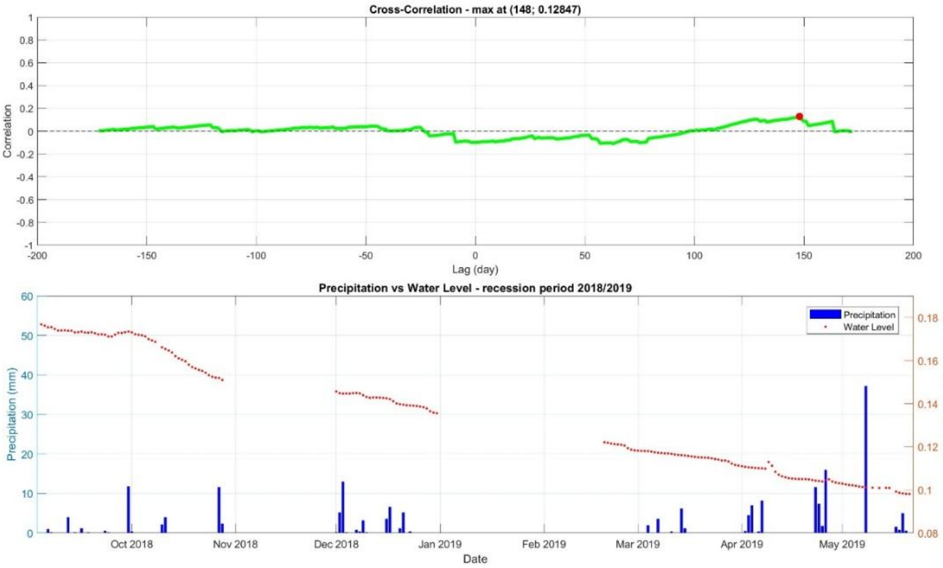


Figure 31 - Cross-correlation analysis of precipitation and water level at Entrebin, for recession period 2018-2019. In the upper subplot, cross-correlation values versus lag time. In the lower subplot, dual-axis graph comparing precipitation (left y-axis) and water level (right y-axis) over time.

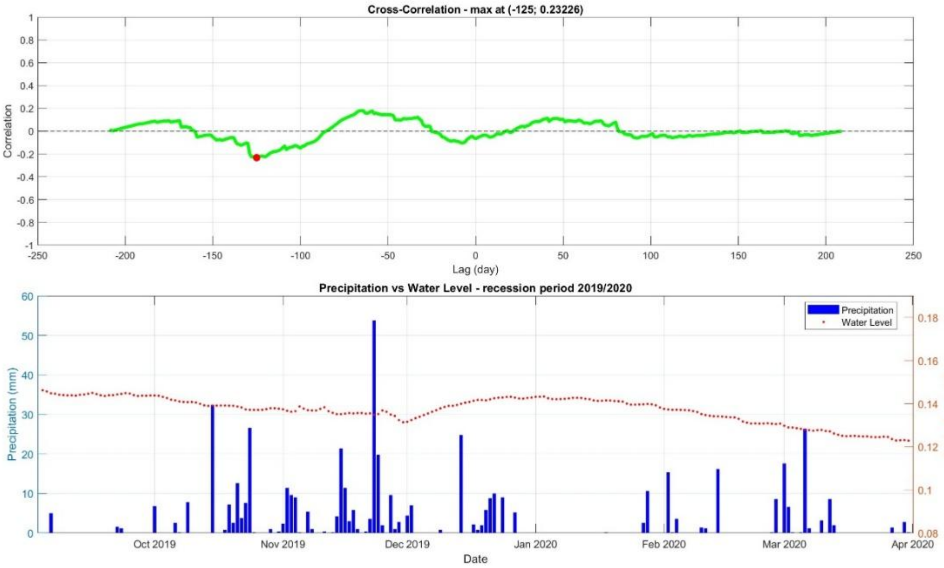


Figure 32 - Cross-correlation analysis of precipitation and water level at Entrebin, for recession period 2019-2020. In the upper subplot, cross-correlation values versus lag time. In the lower subplot, dual-axis graph comparing precipitation (left y-axis) and water level (right y-axis) over time.

The analysis conducted on the relationship between precipitation and water level during the example recession periods, specifically those of 2018/2019 and 2019/2020, does not yield statistically significant results. Although an attempt was made to identify a connection between the two variables through cross-correlation, the results suggest that the relationship is weak or non-existent and, most importantly, inconsistent across the years analysed.

Looking at the first dataset, it can be observed that the maximum correlation between precipitation and water level occurs with a lag of 148 days and has a very low value (0.12847). This indicates a phenomenon that is difficult to interpret in physical terms.

In the same image, in the lower graph, which shows the trend of precipitation and water level over time, a decreasing trend in the water level is highlighted, typical of a recession phase, with sporadic and generally low-intensity precipitation. However, no clear response of the water level to precipitation events is observed, partly due to the presence of some periods (two of which are very evident) where it was not possible to correlate the input data and hydrological output, as daily temperature data from the Roisan-Preyl station were missing. Other causes could depend on additional factors, such as infiltration processes or geological characteristics of the basin, which might have a more significant influence than precipitation itself.

In the second example period, the results of the cross-correlation are different: the maximum correlation occurs at -125 days with a significantly higher value (0.23226), though still not particularly significant. The fact that the lag is negative implies that the water level precedes the precipitation, which lacks a physically coherent explanation in a natural system, unless very complex effects related to water accumulation and release dynamics in the subsurface or the inertia of the hydrological system are considered.

The lower graph also shows a different situation compared to the previous year: precipitation is more frequent and abundant, but the water level exhibits a more gradual and less uniform decline. A small variation in the water level is observed in correspondence with precipitation events, but this link is not confirmed by the correlation evaluation between the two variables, which therefore remains, at the very least, uncertain.

The fact that the cross-correlation results are so different in the two examples suggests that the relationship between precipitation and water level does not follow a stable and repeatable pattern over time. At least for this case study, this is a clear indication of the low statistical significance of the analysis: if a robust relationship between the two variables existed, one would expect to find consistent results across the different periods analysed.

In reality, the correlation values found are very low, and the identified lags cannot be interpreted in a physically plausible way. This leads to the conclusion that the water level during the recession period is likely influenced by a combination of more complex factors, such as basin morphology, soil permeability, and underground drainage dynamics, rather than recent precipitation.

For completeness, the results related to the cross-correlation analysis using air temperature as input are also presented:

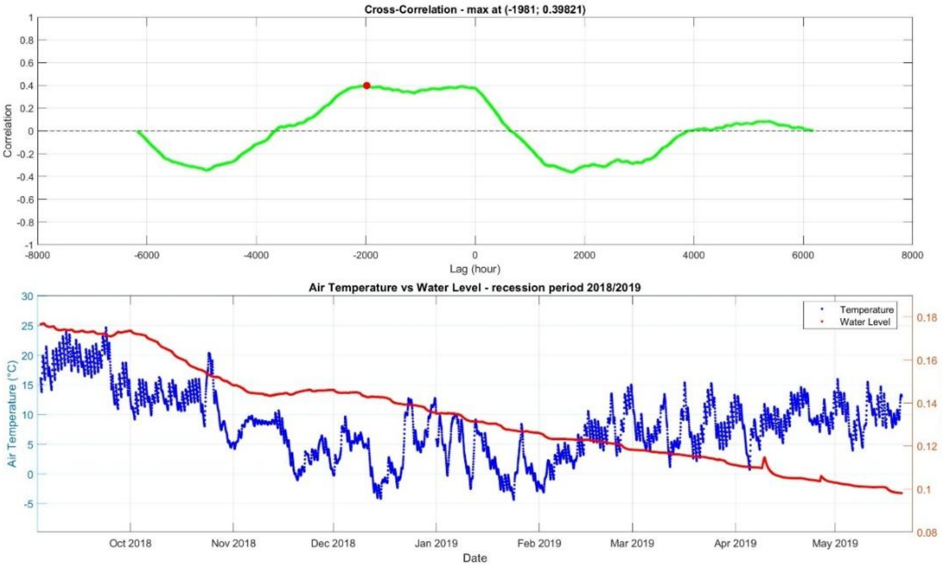


Figure 33 - Cross-correlation analysis of air temperature and water level at Entrebin, for recession period 2018-2019. In the upper subplot, cross-correlation values versus lag time. In the lower subplot, dual-axis graph comparing air temperature (left y-axis) and water level (right y-axis) over time.

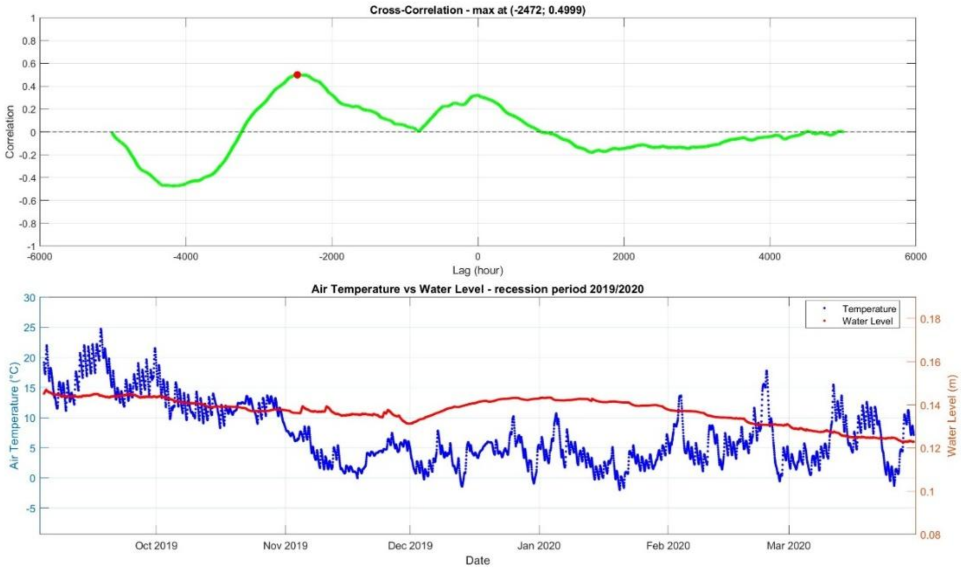


Figure 34 - Cross-correlation analysis of air temperature and water level at Entrebin, for recession period 2019-2020. In the upper subplot, cross-correlation values versus lag time. In the lower subplot, dual-axis graph comparing air temperature (left y-axis) and water level (right y-axis) over time.

Despite a moderate correlation, reaching a value of 0.5, the lag is very long and difficult to interpret. It is likely that the water level is maybe primarily regulated by other phenomena, rather than by a simple thermal effect on water dynamics. In this case as well, no usable result is obtained for the purposes of the ongoing research.

In conclusion, the results show that the cross-correlation analysis does not provide statistically robust evidence to directly link precipitation or air temperature with variations in water level during recession periods. This suggests that the approach that was structured does not work for this case study and does not aid in the hydrological analysis, likely because the analyzed water system is controlled by a combination of more complex processes that would require a more in-depth investigation, possibly considering other factors (e.g., groundwater recharge and geomorphology of the area).

Forecasting Water Level and Air Temperature using the Fourier model

The lack of significance in the cross-correlation highlighted the difficulties in identifying direct relationships between environmental inputs (precipitation, air temperature) and output (spring level). This result, however, does not prevent moving forward with a different approach, based on Fourier analysis, which bypasses the challenges of direct relationships and focuses on the seasonal cyclicity observed in the earlier part of the study.

The results obtained with the Fourier model would demonstrate that, even in complex systems, reliable predictions can be achieved through the analysis of periodic components.

In this section, the focus is on the application of the Fourier model, previously discussed, for predicting the water level of the Entrebin spring and the air temperature. Historical data are used as a base to build a model capable of forecasting the future trends of these two variables.

As mentioned, the model was developed using historical water level data measured at Entrebin (2016–2023), and in this chapter, the aim is to validate it using 2024 data, demonstrating its good predictive capability despite some limitations. This approach represents a step forward compared to the cross-correlation analyses of the previous chapter, which did not yield statistically significant results.

Method and Implementation of the Predictive model

The process continued with the development of the MATLAB script, where the first step was loading the two historical datasets previously filtered using the Kalman method, related to the water level of the Entrebin spring and the air temperature from the Roisan-Preyl station. These data are organized in tables with two main columns: one for the date and time, and one for the measured values.

Subsequently, it was necessary to align the data temporally to ensure that the time series were comparable and to avoid discrepancies due to different measurement times.

Next, the coefficients of the second-order Fourier model, which define its harmonic function, were loaded from a text file. These coefficients are $a_0, a_1, b_1, a_2, b_2, and\ \omega$. They had been calculated and saved in the text file during the earlier phase of model evaluation on the entire historical series.

Table 2 - Coefficients of the second-order Fourier function estimated from the water level dataset for Entrebin.

a0	a1	b1	a2	b2	w
0.13195393	0.018048372	0.017595595	0.0052695933	0.0038053172	0.017308

Before applying the Fourier model to forecast the year 2024, it was necessary to align the model itself with the historical data and the model for 2024. For this reason, a temporal shift was introduced for the water level, $sD\_L = 157\ days$. This value was obtained to ensure that the Fourier function for 2024 aligns with the model's signal on the historical series for the water level. As for the temporal shift for temperature, a different value was used, accounting for the fact that it is an input component and thus requires a greater lead time. Specifically, $sD\_T = 190\ days$.

Therefore, the difference of 33 $days$ reflects the fact that temperature requires a greater lead time compared to the water level, which is effectively expected considering air temperature as an input and water level as an output.

At this point, the forecasts for the year 2024 are made using the Fourier function.

For the water level prediction, the Fourier function is simply applied to the new time series for the year 2024 (t_future):

$$\begin{aligned} future\_L\_fourier &= a0 + a1 * cos(w * (t\_future + sD)) + b1 * sin(w \\ &\quad * (t\_future + sD)) + a2 * cos(2 * w * (t\_future + sD)) \\ &\quad + b2 * sin(2 * w * (t\_future + sD)) \end{aligned}$$

For the air temperature prediction, an additional preliminary step is required. To transform the water level forecast into a temperature forecast, two parameters are introduced: the scaling factor (k) and the offset (m).

A linear transformation was defined to convert the level to temperature, expressed as follows

$$T = k \cdot L + m$$

The scaling factor (k) defines how much the temperature varies in relation to the water level; a higher value of k indicates that small changes in level correspond to large changes in temperature, and vice versa. The offset (m) defines the temperature value when the water level is zero, so this parameter accounts for the "baseline" temperature.

The value of k is calculated as the ratio between the standard deviation of temperature (T_std) and the standard deviation of the level (L_std). This ratio indicates how much the temperature varies relative to the level. If k is large, it means that the temperature is highly sensitive to level variations; if k is small, the temperature is less sensitive.

The coefficient m is obtained as the difference between the mean temperature (T_mean) and the product of the scaling factor k and the mean level (L_mean). This parameter ensures that the linear transformation is correctly centered relative to the historical data.

The mean and standard deviation of the historical datasets are chosen to perform the conversion from one variable to the other because they account for the variability of the data, thus ensuring that the linear transformation is suitable for the dispersion of the measurements. Additionally, they are robust, making the model less sensitive to any remaining outliers.

The scaling factor k describes the sensitivity of the temperature relative to the level and is calculated as $k \approx 360$. This means that for an increase of 0.001 m in the water level, there is a corresponding increase of 0.36 °C.

The linear transformation therefore becomes:

$$T = 360 \cdot L - 36$$

This type of transformation was chosen because it is extremely intuitive and simple. Indeed, it requires the use of two easily calculable parameters and the application of an ordinary linear equation, making it highly adaptable. Additionally, the parameters k and m are easily interpretable, as they have a very clear physical meaning.

The assumptions made, however, come with some limitations. Specifically, a linear relationship between level and temperature is assumed, which could be challenged in more complex systems than the one examined. Also the risk of outliers in the datasets could propagate errors in the calculation of the two parameters of the equation, however this issue is significantly mitigated by the use of pre-filtered data using the Kalman method.

Certainly, the use of only two variables (air temperature and water level) for the calculation of predictions is a limitation, considering how many other potential environmental variables could be included in this type of research, such as precipitation, humidity, and wind.

Finally, it is worth noting that the datasets used do not come from the same site; the two sites are located within the same hydrological basin but are still positioned at a not completely negligible distance from each other. Having datasets measured at the same site would undoubtedly yield more reliable results.

At this point, the linear transformation formula is applied in MATLAB software:

$$future\_T = k * future\_L\_fourier + m$$

The results were visualized on separate graphs and compared with historically measured values for the period 2016–2023 and for 2024.

Visualization of results and corrections for Water level

The model for the water level is as follows:

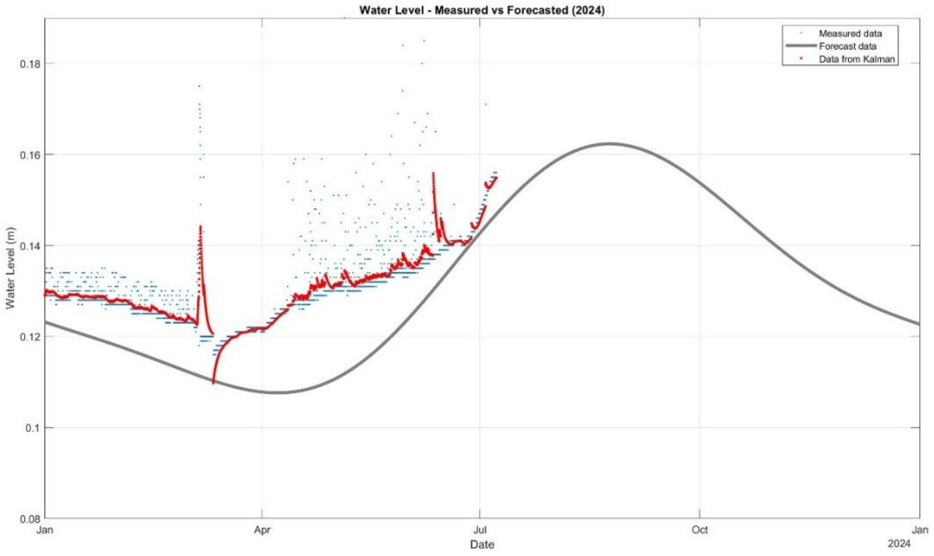


Figure 35 - Comparison of measured water level, Kalman-filtered data and the original forecasted model for Entrebin, 2024.

By comparing it with the rest of the historical series:

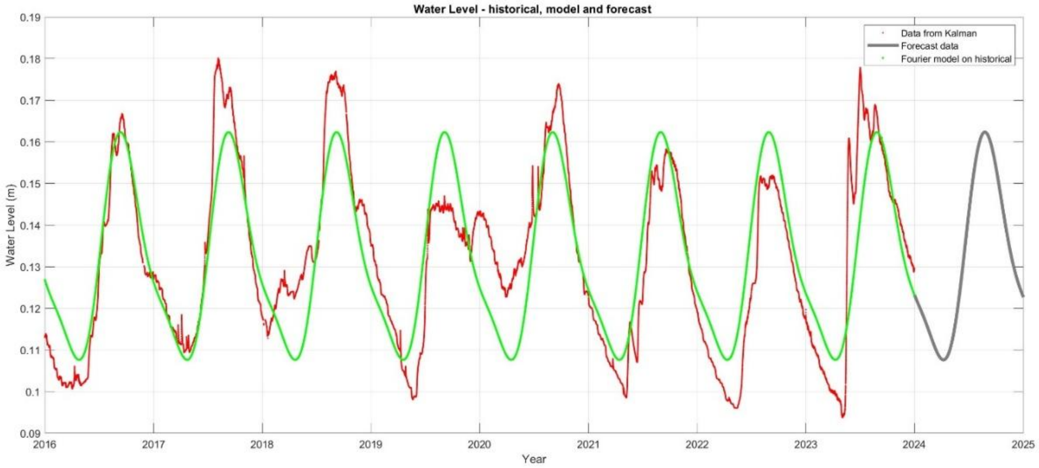


Figure 36 - Comparison of measured Kalman-filtered water level data and the original forecasted model for Entrebin, from 2016.

It can be observed that there is an evident bias between the predictive model and the observations for 2024.

Therefore, the aim is to find a method to ensure that the model accounts for the observed discrepancy. To achieve this, it was decided to use the mean residual value, i.e., the difference between the model and the historical series, from the last 240 historical measurements of 2023, which correspond to the last 10 days of the year.

The use of the residual is intended as a method to correct the bias. By leveraging it, an adjustment is made to align with the real conditions of the system at that moment, with the goal of improving the accuracy of the measurements for the year 2024.

This type of approach, of course, has evident limitations. Specifically, the dependence on the most recent data from the reference series can become problematic: these data could turn out to be outliers, and in any case, they do not account for what happens in the rest of the reference series, potentially leading to incorrect corrections. Additionally, this method assumes stationarity, as the same correction value is consistently applied throughout the model, which is unlikely to be valid.

Therefore, a more sophisticated method is employed, which still involves the use of residuals from the 2016–2023 series but considers all of them this time. A new second-order Fourier model is used to attempt to interpret the trend of these residuals. At this point, the model is applied to the year 2024 as an additive term to correct the bias.

Unfortunately, it will be seen that even this methodology does not correct the original model effectively. It is likely not the best approach, but it was attempted to give weight to the residuals of the series on which the general model is based. It is also important to emphasize that the second-order Fourier model used for the residuals does not provide a good fit but was chosen for continuity and because it is still one of the best models among those tested.

The process continues with the use of the two methods involving residuals for model correction and a comparison with the uncorrected result.

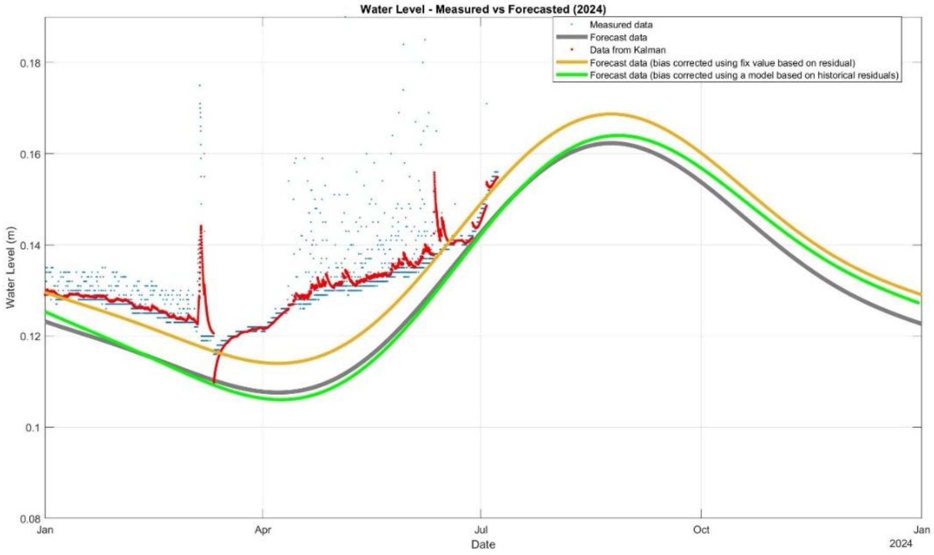


Figure 37 - Comparison of measured water level, Kalman-filtered data, the original forecasted model and the two methods used to correct the bias for Entrebin, 2024.

The use of the average residual from the last days of 2023 as a correction parameter is an approach that leads to a more accurate result compared to the original model. This method accounts for the observed discrepancy between the model and the real data at the end of the calibration period, improving the accuracy of the forecasts for 2024. However, it is important to consider the very evident limitations of this method and explore more advanced approaches for further improvements.

Unfortunately, the curve derived from using a Fourier model on all the data from the historical series does not perform as well; the curve remains very similar to the original one. The (visible) advantages are only evident in the first month and a half of 2024, where the values are closer to those measured. For the rest of the period up to July, the predicted values from the two curves are practically identical. Then, until the end of the year, the corrected model overestimates the original model; unfortunately, it is not possible to determine whether this is beneficial or not, as the corresponding field-measured data for that period are not available.

Visualization of results and corrections for Air Temperature

The forecast obtained for temperature is excellent, and the linear relationship with the water level is very strong. The model is able to reproduce the historical trend with very good accuracy, thus demonstrating the validity of the approach.

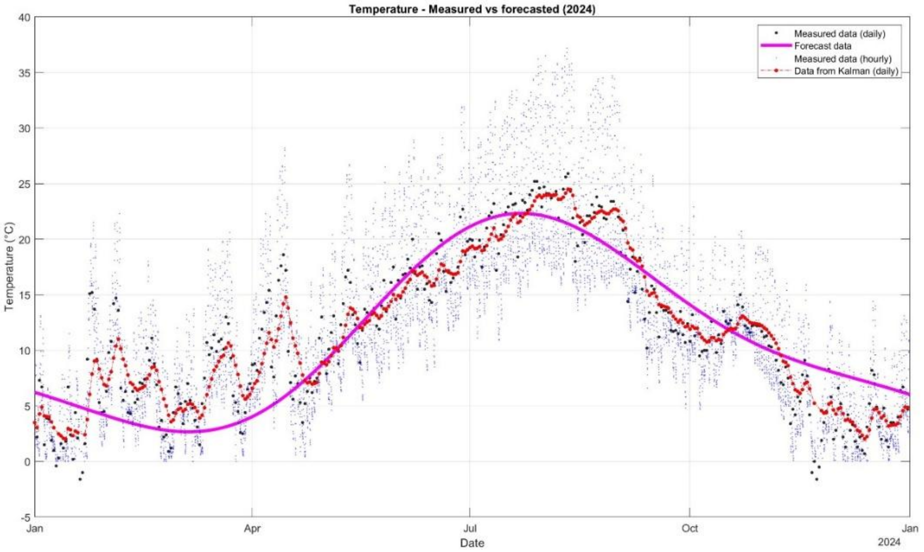


Figure 38 - Comparison of measured air temperature (daily and hourly), Kalman-filtered data and the original forecasted model for Roisan-Preyl station, 2024.

By comparing it with the rest of the historical series:

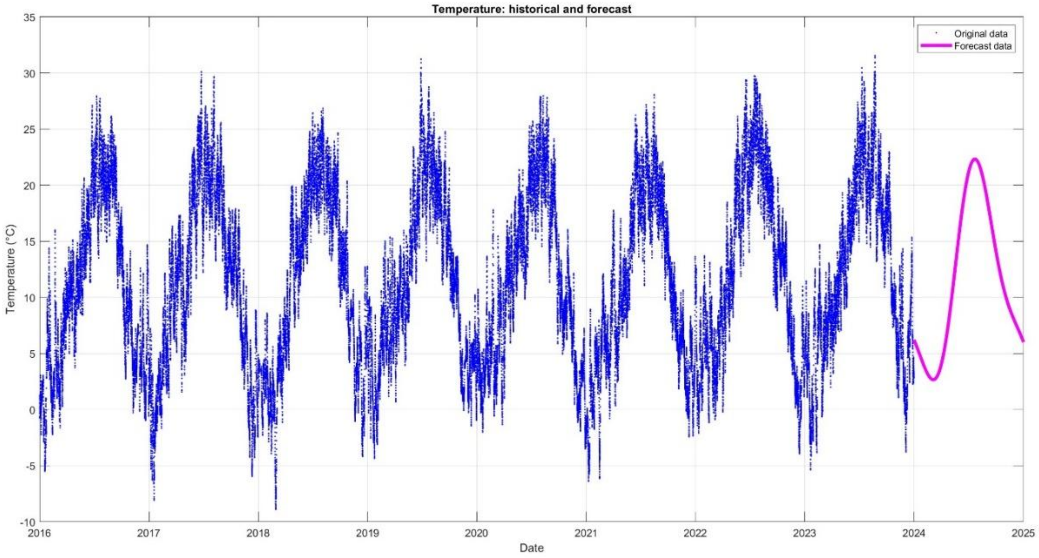


Figure 39 - Comparison of measured air temperature and the original forecasted model for Roisan-Preyl station, from 2016.

The predictive model appears to capture the general trend of temperature very accurately, with a summer peak and a winter minimum. Overall, it seems well-structured and effective in identifying the seasonality of temperature, although limitations exist at the daily and weekly levels.

The average trend of the prediction curve closely follows the historical temperature trend, demonstrating that the linear transformation based on water level was the correct choice.

From the comparison between the historical temperature data filtered with Kalman and the forecast for 2024, it is evident that the distance between the two is very small in the central and final parts, i.e., from May to mid-November. In January and December, an overestimation of temperature is observed, but the largest discrepancy between the two curves occurs in the period from February to April (inclusive), where the forecast significantly underestimates the actual recorded values.

The result regarding the underestimation in the February–April period is very interesting, as it closely mirrors what was observed in the previous section for the water level forecast. Unfortunately, this highlights that the model, at least for this period, did not perform as well as hoped.

For the rest, the temperature forecast results are very encouraging, demonstrating that the Fourier model, combined with a linear transformation based on water level, is capable of correctly reproducing the seasonal trend of temperature. However, there are some limitations, such as the dependence on linearity and the absence of other environmental variables. Certainly, the consideration of the latter could be interesting for future developments aimed at improving the model.

Conclusions

The study conducted on the Entrebin spring has demonstrated how integrating advanced filtering and mathematical modelling techniques can aid in studying complex hydrological dynamics, even in contexts characterized by imperfect data and atypical behaviours.

The application of the Kalman filter enabled the creation of more reliable time series, reducing instrumental noise and mitigating the impact of data gaps, while the use of second-order Fourier functions highlighted the possibility of developing reliable models to simulate the trends of the variables of interest. The water level and temperature forecasts, not without limitations, provided an operational tool to at least anticipate annual trends, highlighting the potential of models based on periodic components.

The results obtained, even if preliminary, provide insights for the future management of water resources in alpine environments. The identification of delays in hydrological response would help better calibrate forecasting models and aquifer conservation measures. In an era of increasing climatic stress, approaches like the one proposed represent an interesting application of how what has been studied in a university context can be implemented in practice for the supervision of at-risk resources, in this case, aimed at the resilience of alpine water systems.

Analysis of limits

The research has highlighted some critical points, particularly concerning:

- Data quality and completeness: Interruptions in the datasets (e.g., 2014–2015) and malfunctions in meteorological instrumentation (non-heated rain gauge) have limited the statistical robustness of the analyses, especially in cross-correlations. This led to the need to create daily datasets to better utilize the data in certain instances. The lack of study on snowmelt for alpine stations like Roisan-Preyl represents a significant limitation in the analysis of water input data in the hydrographic basin of interest.
- Model simplifications: The assumption of linearity between water level and temperature, as well as the use of a second-order Fourier model, overlooked

nonlinear dynamics and multi-variable interactions (e.g., humidity, pressure). These aspects certainly play a role in these systems, and their inclusion would likely have led to more robust and precise results.

- Scalability and generalizability: The geological specificity of Entrebin and the distance between the meteorological station and the spring may raise questions about the transferability of the method to other contexts without significant adaptations. The use of data from the same monitoring station or the integration of multiple stations within the same hydrographic basin would certainly have allowed for more interesting final solutions.
- Dependence on manual corrections: The use of residuals to reduce bias in predictions introduces an element of arbitrariness, limiting the automation of the model. More notable would have been the identification of a more complex corrective model than those currently proposed.

Future developments

To overcome these limitations, future work could try to improve certain aspects of this study by focusing on:

- Strengthening the monitoring network: The installation of additional sensors (e.g., soil moisture, interstitial pressure) and the upgrading of instrumentation (heated rain gauges) would enhance the spatial and temporal resolution of the data. Note, this may not be the only possibility; the implementation of new (or existing) methods to estimate snowmelt based on data related to snowfall/snow accumulation, temperatures, and other input data could prove very useful and interesting. Moreover, it is not always necessary to cover the entire territory with a dense network of sensors. In many cases, it may be sufficient to install the right devices in strategic points, enabling data collection even for areas not directly covered by sensors. Through post-processing techniques, such as triangulation, it is possible to derive information for areas without sensors by leveraging data collected from adjacent stations. This approach will allow for the optimization of resources and cost reduction while maintaining a high level of measurement accuracy.

- Temporal and geographical extension: Replicating the study on other alpine springs and over longer timeframes would validate the methodology and identify trends related to climate change. This study would be very interesting to apply to historical data series that, simply put, begin before 2016. Obviously, the implementation of these techniques in areas characterized by different behaviours would be equally interesting.
- Coupled hydrogeological models: The inclusion of geomechanical parameters was not considered in this study, even though this spring alone is located in an area subject to DGPV (Deep Gravitational Slope Deformation) phenomena, an aspect that was not examined in the study. Additionally, subsurface permeability maps would allow for more accurate simulation of underground flow and offer possibilities for analysing results that are likely different from the current ones.
- Integration of machine learning techniques: Utilizing algorithms based on, for example, recurrent neural networks (RNN) or Long Short-Term Memory (LSTM) models to analyse time series data from hydrological variables.

Sitography and Reference

Bolognini, D., Romani, A., Framarin, M., De Luca D. A. (2010). Hydrogeological study of Entrebin's spring in the municipal district of Aosta (AO). Rev. Valdôtaine Hist. Nat., 64: 5-39.

Box, G. E. P., Jenkins, G. M., & Reinsel, G. C. (2015). Time Series Analysis: Forecasting and Control. John Wiley & Sons.

Centro Funzionale Regione Autonoma Valle d'Aosta. (n.d.). Dettaglio stazione Roisan-Preyl. Retrieved from presidi2.regione.vda.it/str_dataview_station/3560 (Accessed: 14 January 2025).

Clark, M. P., Rupp, D. E., Woods, R. A., Zheng, X., Ibbitt, R. P., Slater, A. G., Schmidt, J., & Uddstrom, M. J. (2008). Hydrological data assimilation with the ensemble Kalman filter: Use of streamflow observations to update states in a distributed hydrological model. Advances in Water Resources, vol. 31(10), 1309-1324

Coltelli, N., Deri, L. (2020). Rilevazioni di correlazioni tra serie temporali. Dipartimento di Informatica, Università di Pisa.

Cucchi, F., Casagrande, G., & Mancà, P. (2000). Chimismo e idrodinamica dei sistemi sorgivi del massiccio del Monte Canin. *Atti e Memorie della Commissione Grotte "E. Boegan"*, Vol. 37, pp. 93-123.

Dabove P. (2023). *Environmental Spatial Analysis* [Lecture notes]. DIATI, Politecnico di Torino.

Gizzi, M., Narcisi, R., Mondani, M., Taddia, G. (2023). Comprehending mountain springs' hydrogeological perspectives under climate change in Aosta Valley (North-Western Italy): new automated tools and simplified approaches. Italian Journal of Engineering Geology and Environment, pp. 73-80.

Grewal M. S., Andrews A. P. (2010). Applications of Kalman Filtering in Aerospace 1960 to the Present. IEEE Control System Magazine, vol. 30, no. 3, pp. 69-78.

ISPRA Ambiente (2013). Linee guida per l'analisi e l'elaborazione statistica di base delle serie storiche di dati idrologici.

Lo Russo, S., Amanzio, G., Ghione, R., & De Maio, M. (2015). Recession hydrographs and time series analysis of springs monitoring data: application on porous and shallow aquifers in mountain areas (Aosta Valley). Environmental Earth Sciences, 73(23), 7415–7434.

Lo Russo, S., Suozzi, E., Gizzi, M., Taddia, G. (2021). SOURCE: a semi-automatic tool for spring-monitoring data analysis and aquifer characterisation. Environmental Earth Science.

Matlab Documentation. (n.d.). Kalman Filter documentations. Retrieved from Matlab Docs.

Matlab Documentation. (n.d.). Curve fitter documentations. Retrieved from Matlab Docs.

Matlab Documentation. (n.d.). Curve fit – Fourier models documentations. Retrieved from Matlab Docs.

Matlab Documentation. (n.d.). Curve fit – Fourier models documentations. Retrieved from Matlab Docs.

Mondani, M., Gizzi, M., & Taddia, G. (2022). Role of Snowpack-Hydrometeorological Sensors for Hydrogeological System Comprehension inside an Alpine Closed-Basin. Sensors, 22(19), 7130.

NOAA National Weather Service. (n.d.). Cross-Correlation. Retrieved from NOAA NWS.

OTT Hydromet GmbH (n.d.). Operating instructions: Groundwater datalogger OTT CTD (version “01-1009”). OTT Messtchnnik GmbH & Co.

Piersanti, A., Righini, G., Russo, F., Cremona, G., Vitali, L., & Ciancarella, L. (2007). Spatial Representativeness of Air Quality Monitoring Stations in Italy. Harmo 15.

Politecnico di Torino, Regione Autonoma Valle d’Aosta. (2012). Scheda sorgente Entrebin. Politecnico di Torino.

Pyo, J., Pachepsky, Y., Kim, S., Abbas, A., Kim, M., Kwon, Y. S., Ligaray, M., & Cho, K. H. (2023). Long short-term memory models of water quality in inland water environments. Water Research X, vol. 21.

Regione Autonoma Valle d’Aosta - Presidenza della Giunta, Dipartimento Protezione Civile e Vigili del Fuoco, Centro funzionale regionale (2022), Bollettino Idrologico - ottobre 2022. Centro funzionale VdA.

Regione Valle d’Aosta. (n.d.). Gestione delle risorse idriche. Retrieved from regione.vda.it (Accessed: 11 January 2025).

Reid, I., Term, H. (2001). Estimation and Kalman Filtering. Oxford University, Department of Engineering Science.

Rogantin, M.P., Sasso E. (n.d.). Trasformazioni lineari. Dipartimento di Matematica, Università di Genova.

Schwatke, C., Dettmering, D., Bosch, W., Seitz, F. (2015). Kalman filter approach for estimating water level time series over inland water using multi-mission satellite altimetry. Hydrology and Earth System Sciences (HESS), 12, pp. 4813-4855.

SIGEA - Società Italiana di Geologia Ambientale. (n.d.). Geologia Urbana di Aosta. Retrieved from sigeaweb.it/documenti/gda-supplemento-2-2017 (Accessed: 11 January 2025).

Viero, D. P., Peruzzo, P., Carniello, L., & Defina, A. (2014). Integrated mathematical modeling of hydrological and hydrodynamic response to rainfall events in rural lowland catchments. Water Resources Research, 50(7), 1-15.

Welch G., Bishop G. (2006). An Introduction to the Kalman Filter. University of North Carolina, USA.

Zhou, L. (2024). Perspective Chapter: Big Data and Deep Learning in Hydrological Modeling. *IntechOpen*.

Appendix

This appendix contains the MATLAB codes developed and used throughout the thesis for data analysis, model simulation and the implementation of algorithms described in the previous chapters. The codes are organized into phases, each corresponding to a specific functionality required for the work carried out.

To execute the codes, MATLAB version R2022a (Update 5) on Windows was used. The Roisan-Preyl data files used as input are available on the website of the Centro Funzionale Regione Autonoma V.d.A [https://presidi2.regione.vda.it/str_dataview_download#]. The data to be used are saved and pre-processed in the folder named "Roisan" located within the scripts folder.

List of Codes

Phase A.m

Description: Imports data from a text file, combines date and time columns, and checks for missing timestamps to identify periods when the instrumentation was not functioning. It then checks the values of the measured data column, flagging NaN or negative values (in the case of water level measurements) as invalid measurements. At the end, it filters the data by removing rows with invalid measurements and saves the results in new files.

Script:

```
clear all
close all
clc

%% DATA IMPORT

filename = 'Livello_Entrebin';
data = readtable(filename, 'Delimiter', ';', 'Format',
'%{dd/MM/yyyy}D%{HH:mm:ss}D%s');

%% DATA PREPARATION (nowork, nomeas, outlier)

%% Trova No_work

% Estrae le colonne di data e ora come unica colonna datetime
```

```
date_time = data.Var1 + timeofday(data.Var2);

% Ordina i dati (con dati PoliTo non utile, lo faccio per sicurezza)
date_time = sort(date_time);

% Determina l'intervallo di tempo tra due righe successive
time_interval = hours(1);

% Crea una serie temporale completa tra il primo e l'ultimo timestamp
time_range = (date_time(1):time_interval:date_time(end))';

% Confronta la serie temporale generata con i timestamp del file
missing_times = setdiff(time_range, date_time);

%% Stampa se tutto ok, se NON presenti NoWork

% Visualizza i risultati
if isempty(missing_times)
    disp('Non ci sono date o orari mancanti.');
else
    disp('Date e orari mancanti:');
    % Trova intervalli di periodi mancanti consecutivi
    start_idx = 1;
    for i = 2:length(missing_times)
        % Se l'intervallo tra due timestamp mancanti consecutivi non è uguale
a time_interval, crea un nuovo periodo
        if missing_times(i) - missing_times(i-1) ~= time_interval
            fprintf('Attenzione! La strumentazione non ha funzionato dal %s
alle ore %s al %s alle ore %s.\n', ...
                    datestr(missing_times(start_idx), 'dd/mm/yyyy'),
datestr(missing_times(start_idx), 'HH:MM:ss'), ...
                    datestr(missing_times(i-1), 'dd/mm/yyyy'),
datestr(missing_times(i-1), 'HH:MM:ss'));
            start_idx = i;
        end
    end

    % Stampa ultimo intervallo
    fprintf('Attenzione! La strumentazione non ha funzionato dal %s alle ore
%s al %s alle ore %s.\n', ...
            datestr(missing_times(start_idx), 'dd/mm/yyyy'),
datestr(missing_times(start_idx), 'HH:MM:ss'), ...
            datestr(missing_times(end), 'dd/mm/yyyy'),
datestr(missing_times(end), 'HH:MM:ss'));
end

%% Individua No_meas

data = readtable("Livello_Entrebin.txt");

% Aggiunge colonne di controllo
```

```
n = height(data);
data.Var4 = zeros(n, 1); % Colonna di verifica per variable

% Controlla i valori per ogni riga
for i = 1:n
    % Effettua controllo sulla colonna variable (colonna 3)
    if isnan(data.Var3(i)) || data.Var3(i) < 0
        data.Var4(i) = 1; % 1 se il valore è NaN o negativo
    end
end

% Salva il nuovo file con le colonne aggiuntive
output_filename = 'file2findOutlier.txt';
writetable(data, output_filename, 'Delimiter', 'tab');

disp('Elaborazione completata e file salvato.')


%% Trova No_Meas

% Carica il file generato precedentemente
input_filename = 'file2findOutlier.txt';
data = readtable(input_filename, 'Delimiter', 'tab');

% Filtra solo le righe con colonna 4 uguale a 0
filtered_data = data(data.Var4 == 0, :);

% Salva il nuovo file con le righe filtrate
filtered_filename = 'fileNoOutlier.txt';
writetable(filtered_data, filtered_filename, 'Delimiter', ';');

disp('Nuovo file con righe filtrate salvato.');
```

Phase B.m

Description: Imports data from a text file, removes years with incomplete data, and applies a Kalman filter to estimate the trend of the data over time. For each considered year, it generates two plots: one comparing the original data with the Kalman filter estimate, and another showing the residuals (differences between the original data and the estimates). The results, including both processed data and plots, are saved in a dedicated folder and a text file.

Script:

```
% NOTA:
%       *  --> Valido per sorgente Entrebin
%       ** --> Valido per dataset di Livello dell'acqua per la sorgente
Entrebin

filename = 'fileNoOutlier';
data = readtable(filename, 'Delimiter', ';', 'Format', '%s%s%f%f%f',
'ReadVariableNames', true, 'HeaderLines', 0);

% Divide la data e l'ora
dateStr = data.Var1;
timeStr = data.Var2;
values = data.Var3;

% Converte le stringhe della data in formato datetime
dates = datetime(dateStr, 'InputFormat', 'dd/MM/yyyy');
times = datetime(timeStr, 'InputFormat', 'HH:mm:ss');

% Combina data e ora in un unico datetime array
dateTimeArray = dates + timeofday(times);

% Definisce gli anni unici
uniqueYearsList = unique(year(dateTimeArray));

% Esclude gli anni 2014, 2015, 2024 (anni i cui dati sono incompleti o
assenti)*
excludeYears = [2014, 2015, 2024];
uniqueYearsList = setdiff(uniqueYearsList, excludeYears);

% % Prealloca la matrice uniqueYears
% uniqueYears = zeros(length(uniqueYearsList), 3);

% Definisce i limiti dell'asse delle ordinate **
yMin = 0.08;
yMax = 0.19;

% Crea una nuova cartella per salvare i grafici
outputFolder = 'Grafici Annuali (post-Kalman)';
if ~exist(outputFolder, 'dir')
    mkdir(outputFolder);
end

% Parametri del filtro di Kalman
A = 1;  % Matrice di transizione dello stato
H = 1;  % Matrice di osservazione
Q = 1e-9;  % Varianza del rumore di processo **
R = 0.001^2;  % Varianza del rumore di misura **
initialCovarianceEstimate = 0.001^2;  % Stima iniziale della covarianza **

% Crea e apre il file processed.txt in modalità scrittura
processed = fopen('processed.txt', 'w');
```

```
fprintf(processed, 'Var1\tVar2\n');

% Calcola filtro di Kalman & crea figura per ogni anno solare
for i = 1:length(uniqueYearsList)
    currentYear = uniqueYearsList(i);

    % Filtra i dati per l'anno corrente
    yearIdx = year(dateTimeArray) == currentYear;
    yearData = dateTimeArray(yearIdx);
    yearValues = values(yearIdx);

    % Stima iniziale dello stato
    initialStateEstimate = yearValues(1,1); % Stima iniziale dello stato

    % Inizializza il filtro di Kalman
    stateEstimate = initialStateEstimate;
    covarianceEstimate = initialCovarianceEstimate;
    kalmanEstimates = zeros(length(yearValues), 1);
    covarianceEstimates = zeros(length(yearValues), 1);

    % Applica il filtro di Kalman
    for t = 1:length(yearValues)
        % Previsione
        predictedStateEstimate = A * stateEstimate;
        predictedCovarianceEstimate = A * covarianceEstimate * A' + Q;

        % Aggiornamento
        innovation = yearValues(t) - H * predictedStateEstimate;
        innovationCovariance = H * predictedCovarianceEstimate * H' + R;
        kalmanGain = predictedCovarianceEstimate * H' / innovationCovariance;
        stateEstimate = predictedStateEstimate + kalmanGain * innovation;
        covarianceEstimate = (1 - kalmanGain * H) *
predictedCovarianceEstimate;

        % Salva la stima del fenomeno
        kalmanEstimates(t) = stateEstimate;
        covarianceEstimates(t) = covarianceEstimate;

        % Scrive i dati nel file
        fprintf(processed, '%s\t%f\n', datestr(yearData(t), 'dd-mmm-yyyy
HH:MM:SS'), kalmanEstimates(t));
    end

    % Crea una nuova figura (dati originali vs modello kalman)
    figure;
    plot(yearData, yearValues, 'b.', 'DisplayName', 'Dati Originali'); % Dati
originali in blu
    hold on;
    plot(yearData, kalmanEstimates, 'r.', 'DisplayName', 'Stima Kalman'); %
Stima Kalman in rosso
```

```
    hold off;

    xlabel('Data');
    ylabel('Livello Acqua (m)');
    title(['Anno: ', num2str(currentYear)])
    legend('Location', 'best');
    ylim([yMin yMax]);

    % Salva la figura nella cartella
    saveas(gcf, fullfile(outputFolder, ['Grafico ' num2str(currentYear)
'.fig']));
    close;


    % Crea una nuova figura per viasualizzare l'incertezza
    figure;
    bar(yearData, (kalmanEstimates-yearValues), 'k', 'DisplayName',
'Residui'); % Stima Kalman in nero
    hold on;
    % Aggiunge barre di errore per rappresentare l'incertezza di stima
    errorbar(yearData, kalmanEstimates, (kalmanEstimates-yearValues), 'r',
'LineStyle', 'none', 'DisplayName', 'Incertezza Stima'); % Barre di errore in
rosso
    hold off;
    xlabel('Data');
    ylabel('Resisui (m)');
    title(['Incertezza di stima anno: ', num2str(currentYear)])
    legend('Location', 'best');

    % Salva la figura nella cartella
    saveas(gcf, fullfile(outputFolder, ['Incertezza anno '
num2str(currentYear) '.fig']));
    close;

end

% Chiude il file
fclose(processed);

disp('File processed.txt creato con successo!');
```

Phase C.m

(to be used with the auxiliary function fitFourier.m)

Description: Imports and analyses a time series dataset, applying two different modelling approaches: a fifth-degree polynomial fit for each calendar year and a second order

Fourier fit for the complete series. In the first case, for each year, it calculates a polynomial that approximates the data, evaluates the goodness of fit using the coefficient of determination R^2 and saves the polynomial coefficients. In the second case, it defines the starting year and applies a Fourier model using a function defined via *Curve Fitter*. Graphical and textual results are saved in a dedicated folder. At the end, it calculates and visualizes the residuals of the Fourier fit and analyzes their mean and standard deviation for bias adjustments in *Phase F*.

Script:

```
% NOTA:
%       ** --> Valido per dataset di Livello dell'acqua per la sorgente
Entrebin

filename = 'processed';
data = readtable(filename, 'Delimiter', '\t', 'ReadVariableNames', true);

% Combina data e ora in un unico datetime array
dateTimeArray = data.Var1;
values = data.Var2;

% Individua gli anni unici
uniqueYearsList = unique(year(dateTimeArray));

% % Definisci i limiti dell'asse y (SE prima non si è eseguita faseB ==> de-
commentare)
% yMin = 0.08;
% yMax = 0.19;

% Crea una nuova cartella
outputFolder = 'Grafici con modelli';
if ~exist(outputFolder, 'dir')
    mkdir(outputFolder);
end

% Prealloca la tabella per i valori di R2post (coefficiente di
determinazione)
R2Tablepost = table('Size', [0 2], 'VariableTypes', {'double',
'double'},'VariableNames', {'Year', 'R2post'});

% Prealloca la tabella per i coefficienti dei polinomi
coeff_pol = table('Size', [0 10], 'VariableTypes', {'double', 'double',
'double', 'double', 'double', 'double', 'double', 'double', 'double',
'double'}, 'VariableNames', {'Year', 'five', 'four', 'three', 'two', 'one',
'zero', 'S', 'mu1', 'mu2'});
```

```
% Crea una figura per ciascun anno e calcola media, deviazione standard e
polinomio
for i = 1:length(uniqueYearsList)
    currentYear = uniqueYearsList(i);

    % Filtra i dati per l'anno corrente
    yearIdx = year(dateTimeArray) == currentYear;
    yearData = dateTimeArray(yearIdx);
    yearValues = values(yearIdx);

    % Regola l'asse delle ascisse per il polinomio
    x = datenum(yearData);  % converte date in numeri seriali

    % Fit polinomiale (grado 5 in questo esempio, puoi cambiare il grado)
    [p, S, mu] = polyfit(x, yearValues, 5);  % coefficienti del polinomio
    % con delta = standard error
    [yFit, delta] = polyval(p, x, S, mu);  % valori del polinomio per l'asse
x

    % Aggiungi i coefficienti e i valori dei coefficienti delle equazioni
sulla tabella
    coeff_pol = [coeff_pol; {currentYear, p(1), p(2), p(3), p(4), p(5), p(6),
S.normr, mu(1), mu(2)}];

    % Calcolo dei residui
        residuals = yearValues - yFit;
        residMean = mean(residuals);
        residStd = std(residuals);

    % Aggiungi i coefficienti e i valori di R2 alla tabella
        rsq = 1 - sum(residuals.^2) / sum((yearValues -
mean(yearValues)).^2);
        R2Tablepost = [R2Tablepost; {currentYear, rsq}];

    % Equazione del polinomio come stringa
    eqn = sprintf('y = %.2fx^5 + %.2fx^4 + %.2fx^3 + %.2fx^2 + %.2fx + %.2f',
p(1), p(2), p(3), p(4), p(5), p(6));

    %%
    % Crea una nuova figura
    figure;
    plot(yearData, yearValues, '.');
    hold on;
    % Aggiungi il fit polinomiale
    plot(yearData, yFit, 'g-', 'LineWidth', 1.5);
    hold off;

    xlabel('Month');
    ylabel('Water level (m)');
    title(['Entrebin spring (Ao), year: ', num2str(currentYear)])
    % Aggiungi l'equazione del polinomio sul grafico
    text(yearData(ceil(end/2)), yMin + (yMax - yMin) * 0.1, eqn, 'Color',
'g');
    legend({'Values', 'Polyfit'}, 'Location', 'best');
    ylim([yMin yMax]);  % Imposta i limiti dell'asse y
    xlim([datetime(currentYear,1,1) datetime(currentYear,12,31)]); % Imposta
i limiti dell'asse x

    % Salva la figura nella cartella
    saveas(gcf, fullfile(outputFolder, ['Grafico ' num2str(currentYear)
'.fig']));
    close;

end

% Salva la tabella R2
writetable(R2Tablepost, fullfile(outputFolder, 'R2tablepost.txt'),
'Delimiter', '\t');
disp('R2 post values salvato in R2tablepost.txt');

% Salva la tabella coeff polinomi
writetable(coeff_pol, fullfile(outputFolder, 'coeffpol.txt'), 'Delimiter',
'\t');
disp('Equations coefficients salvato in coeffpol.txt');


%% ANALISI CON FOURIER
% Filtra i dati per il periodo post-2015
post2015Dates = dateTimeArray(year(dateTimeArray) > 2015);
post2015Values = values(year(dateTimeArray) > 2015);
x_post2015 = datenum(post2015Dates);  % converte le date in numeri seriali

% Richiama la funzione di fitting (fitFourier.m)
[fitresult_post2015, gof_post2015] = fitFourier(x_post2015, post2015Values);

% Calcola i valori del fit per la serie temporale post-2015
post2015YearRange =
datetime(2016,1,1,0,0,0):hours(1):datetime(max(year(post2015Dates)),12,31,23,
59,59);
x_post2015_full = datenum(post2015YearRange);  % converte le date in numeri
seriali
yFit_post2015 = feval(fitresult_post2015, x_post2015_full);

% Crea una nuova figura per il fit Fourier della serie temporale post-2015
figure;
plot(post2015Dates, post2015Values, '.', 'DisplayName', 'Values');
hold on;
plot(post2015YearRange, yFit_post2015, 'g.', 'LineWidth', 1.5, 'DisplayName',
'Fourier Fit');

xlabel('Date');
ylabel('Livello Acqua (m)');
```

```
title('Fit Fourier per la serie temporale post-2015');
legend('Location', 'best');
grid on;
hold off;

% Salva la figura nella cartella
saveas(gcf, fullfile(outputFolder, 'Grafico serie completa post-2015.fig'));
close;

% Salva i coefficienti del fit Fourier per la serie temporale post-2015
coeff_names = {'a0', 'a1', 'b1', 'a2', 'b2', 'w'}; %fourier secondo ordine
coeff_values_post2015 = coeffvalues(fitresult_post2015);
coeff_post2015 = array2table(coeff_values_post2015, 'VariableNames',
coeff_names);
writetable(coeff_post2015, fullfile(outputFolder,
'coeff_post2015_Fourier.txt'), 'Delimiter', '\t');
disp('Coefficienti del modello Fourier salvati in
coeff_post2015_Fourier.txt');

% Calcola i residui
residui = post2015Values - feval(fitresult_post2015, x_post2015);

% Crea una nuova figura per i residui
figure;
bar(post2015Dates, residui, 'DisplayName', 'Residui');
xlabel('Date');
ylabel('Residui (m)');
title('Residui del Fit Fourier per la Serie Temporale Post-2015');
legend('Location', 'best');
grid on;
ylim([-0.05 0.05]);  % Imposta i limiti dell'asse y **
hold off;

% Salva la figura nella cartella
saveas(gcf, fullfile(outputFolder, 'Residui_SeriePost2015_Fourier.fig'));
close;

%% Per aggiustamento BIAS (risultati ultima parte)
% Trova l'ultimo valore del residuo
ultimo_residuo = residui(end);
ultimo_residuo1 = residui(mean(end-240:end));

media_residuo = mean(residui);
std_residuo = std(residui);
```

Phase D.m

Description: Analyses a time series dataset by dividing it into hydrological years, defined as the period between the annual maximum and the maximum of the following year. For each hydrological year, it generates a plot of the data and saves it in a dedicated folder. In addition, it identifies and analyses the recession period, defined as the interval between the annual maximum and the minimum of the following year, creating separate plots for this phase. The results are saved in two distinct folders: one for complete hydrological years and one for recession periods.

Script:

```
% NOTA:
%       ** --> Valido per dataset di Livello dell'acqua per la sorgente
Entrebin

% clear all
% close all
% clc

%%
% Legge i dati dal file processed.txt oppure da processed_daily.txt
data = readtable('processed.txt', 'Delimiter', '\t', 'Format', '%{dd-MMM-yyyy
HH:mm:ss}D%f', 'ReadVariableNames', true); % SE uso dati orari
% data = readtable('processed_daily.txt', 'Delimiter', '\t', 'Format', '%{dd-
MMM-yyyy HH:mm:ss}D%f', 'ReadVariableNames', true); % SE uso dati giornalieri

% Prealloca matrici per gli intervalli anni idrologici
hydrologicYearIntervals = [];
countDays = [];
hydrologicYearIntervals2 = []; %%
countDays2 = []; %%

% Definisce i limiti dell'asse delle ordinate **
yMin = 0.08;
yMax = 0.19;

for currentYear = 2016:(max(year(data.Var1)) - 1)

    yearIdx = year(data.Var1) == currentYear;
    yearValues = data.Var2(yearIdx);
    [~, maxIdx] = max(yearValues);

    % Trova il massimo dell'anno successivo
    nextYearIdx = year(data.Var1) == currentYear + 1;
    nextYearValues = data.Var2(nextYearIdx);
    [~, nextMaxIdx] = max(nextYearValues);
    [~, nextMinIdx] = min(nextYearValues); %%

    % Definisci inizio e fine dell'anno idrologico (il 2 per definire il solo
periodo di recessione
    startDate = data.Var1(find(yearIdx, 1) + maxIdx - 1);
    endDate = data.Var1(find(nextYearIdx, 1) + nextMaxIdx - 1);
    endDate2 = data.Var1(find(nextYearIdx, 1) + nextMinIdx - 1); %%

    % Calcola la durata in giorni tra startDate e endDate
    durationDays = days(endDate - startDate);
    durationDays2 = days(endDate2 - startDate); %%

    % Aggiungi i valori alla matrice degli anni idrologici
    hydrologicYearIntervals = [hydrologicYearIntervals; startDate, endDate];
    countDays = [countDays; durationDays];
    hydrologicYearIntervals2 = [hydrologicYearIntervals2; startDate,
endDate2]; %%
    countDays2 = [countDays2; durationDays2]; %%
end


%%
% Crea una nuova cartella per salvare i grafici anni idrologici
outputHydrologicFolder = 'Grafici Idrologici anni';
if ~exist(outputHydrologicFolder, 'dir')
    mkdir(outputHydrologicFolder);
end

% Genera grafici per ciascun intervallo idrologico
for i = 1:size(hydrologicYearIntervals, 1)
    startDate = hydrologicYearIntervals(i, 1);
    endDate = hydrologicYearIntervals(i, 2);

    % Filtra i dati per l'intervallo corrente
    intervalIdx = data.Var1 >= startDate & data.Var1 <= endDate;
    intervalData = data.Var1(intervalIdx);
    intervalValues = data.Var2(intervalIdx);

    % Crea una nuova figura per l'intervallo idrologico
    figure;
    plot(intervalData, intervalValues, 'b.', 'DisplayName', 'Valori'); % Dati
in blu
    xlabel('Data');
    ylabel('Livello acqua (m)');
    title(['Anno Idrologico: ', datestr(startDate, 'yyyy'), '/',
datestr(endDate, 'yyyy')])
    legend('Location', 'best');

    % Salva la figura nella cartella
    saveas(gcf, fullfile(outputHydrologicFolder, ['Idrologico ',
datestr(startDate, 'yyyy'), '_', datestr(endDate, 'yyyy'), '.fig']));
```

```
    close;
end

disp('Grafici degli anni idrologici creati con successo!');


%%
% Crea una nuova cartella per salvare i grafici perido recessione (da Massimo
a.corrente a Minimo a.successivo)
outputHydrologicFolder2 = 'Grafici Periodo recessione';
if ~exist(outputHydrologicFolder2, 'dir')
    mkdir(outputHydrologicFolder2);
end

% Genera grafici per ciascun intervallo idrologico di recessione
for j = 1:size(hydrologicYearIntervals2, 1)
    startDate = hydrologicYearIntervals2(j, 1);
    endDate2 = hydrologicYearIntervals2(j, 2);

    % Filtra i dati per l'intervallo corrente
    intervalIdx2 = data.Var1 >= startDate & data.Var1 <= endDate2;
    intervalData2 = data.Var1(intervalIdx2);
    intervalValues2 = data.Var2(intervalIdx2);

    % Crea una nuova figura per l'intervallo idrologico di recessione
    figure;
    plot(intervalData2, intervalValues2, 'b.', 'DisplayName', 'Valori'); %
Dati in blu
    xlabel('Data');
    ylabel('Livello acqua (m)');
    title(['Periodo recessione - anno idrologico: ', datestr(startDate,
'yyyy'), '/', datestr(endDate2, 'yyyy')])
    legend('Location', 'best');
    ylim([yMin yMax]);

    % Salva la figura nella cartella
    saveas(gcf, fullfile(outputHydrologicFolder2, ['Recessione - Idrologico
', datestr(startDate, 'yyyy'), '_', datestr(endDate2, 'yyyy'), '.fig']));
    close;
end

disp('Grafici della fase recessione degli anni idrologici creati con
successo!');
```

Phase E_Precipitation.m

Phase E_Temperature.m

Description: Analyses the relationship between input and output during hydrological recession periods, as defined in *Phase D*. It filters the input and output data within the recession interval, normalizes the data, and calculates the cross-correlation between the two datasets to identify time lags and correlation coefficients. The results are visualized in plots showing both the temporal trends of the two variables and the cross-correlation, saved in a dedicated folder. For precipitation, the code works with daily data, while for temperature, it uses hourly datasets.

Script (Precipitation):

```
% Importa input
filename = fullfile('Roisan/', 'processedPRd.txt');
data3 = readtable(filename, 'Delimiter', '\t', 'ReadVariableNames', true);

% Combina data e ora in un unico datetime array
dateTimeArrayP = data3.Var1;
valuesP = data3.Var2;

% Importa output
filename = 'processed_daily';
data4 = readtable(filename, 'Delimiter', '\t', 'ReadVariableNames', true);

% Combina data e ora in un unico datetime array
dateTimeArrayLd = data4.Var1;
valuesLd = data4.Var2;

% Crea una nuova cartella per salvare i grafici
outputFolder = 'Grafici Periodo recessione - Precipitazione
(CrossCorrelation)';
if ~exist(outputFolder, 'dir')
    mkdir(outputFolder);
end

% Inizializza i dati filtrati
filteredDataP = [];
filteredValuesP = [];
filteredDataLd = [];
filteredValuesLd = [];

% Esclude anni specifici
yearsToExclude = [2011 2012 2013];
```

```
% Ciclo for attraverso gli intervalli di anni idrologici, considerando i soli
periodi recessione
for i = 1:size(hydrologicYearIntervals2, 1)

% Imposta le variabili inizio e fine data per l'anno idrologico corrente
startDate = hydrologicYearIntervals2(i, 1);
endDate2 = hydrologicYearIntervals2(i, 2);

% Converte le date da stringa a formato datetime
startYear = datetime(startDate, 'InputFormat', 'dd-MMM-yyyy HH:mm:ss');
currentYear = year(startYear);

% Filtra i dati all'interno dell'intervallo dell'anno idrologico corrente,
escludendo gli anni specificati

currentFilteredDataP = dateTimeArrayP(dateTimeArrayP >= startDate &
dateTimeArrayP <= endDate2 & ~ismember(year(dateTimeArrayP),
yearsToExclude));
currentFilteredValuesP = valuesP(dateTimeArrayP >= startDate & dateTimeArrayP
<= endDate2 & ~ismember(year(dateTimeArrayP), yearsToExclude));

currentFilteredDataLd = dateTimeArrayLd(dateTimeArrayLd >= startDate &
dateTimeArrayLd <= endDate2 & ~ismember(year(dateTimeArrayLd),
yearsToExclude));
currentFilteredValuesLd = valuesLd(dateTimeArrayLd >= startDate &
dateTimeArrayLd <= endDate2 & ~ismember(year(dateTimeArrayLd),
yearsToExclude));

% Aggiunge i dati filtrati agli array finali

filteredDataP = [filteredDataP; currentFilteredDataP];
filteredValuesP = [filteredValuesP; currentFilteredValuesP];

filteredDataLd = [filteredDataLd; currentFilteredDataLd];
filteredValuesLd = [filteredValuesLd; currentFilteredValuesLd];

% Imposta i dati per l'anno corrente

yearDataP = filteredDataP;
yearValuesP = filteredValuesP;

yearDataLd = filteredDataLd;
yearValuesLd = filteredValuesLd;

%% (PROVA: Rimuove tutti i valori minori di 10 mm tra le precipitazioni)

% % Imposta a 0 tutti i valori di precipitazioni minori di 10 mm
% yearValuesP(yearValuesP < 10) = 0;

%%
% Rimuove i valori NaN dai dati comuni
```

```
validIdx = ~isnan(yearValuesP) & ~isnan(yearValuesLd);
valoriP = yearValuesP(validIdx);
valoriLd = yearValuesLd(validIdx);
ascisse_comuni = yearDataP(validIdx);

% Normalizza i valori comuni
valoriP_norm = (valoriP - mean(valoriP)) / std(valoriP);
valoriLd_norm = (valoriLd - mean(valoriLd)) / std(valoriLd);

% Calcola la cross-correlazione
[c, lags] = xcorr(valoriP_norm, valoriLd_norm, 'normalized');

% Trova il massimo valore di correlazione
[max_corr, idx] = max(abs(c));
lag_max = lags(idx);

% Visualizza i risultati
figure;
subplot(2, 1, 1);
yyaxis left
bar(yearDataP, yearValuesP, 'm', 'DisplayName', 'PR');
ylabel('Precipitazioni (mm)');
ylim([0 60]);

yyaxis right
plot(yearDataLd, yearValuesLd, 'r.', 'DisplayName', 'LE', 'MarkerSize', 5);
ylabel('Livello acqua (m)');
ylim([0.08 0.19]);

xlabel('Data e Ora');
title(['Confronto tra PR e LE - Anno ', num2str(currentYear)]);
legend('show', 'Location', 'best');
grid on;

subplot(2,1,2);
plot(lags, c, 'g');
hold on;
plot(lag_max, c(idx), 'ro', 'MarkerSize', 10, 'LineWidth', 2);
title(['Cross-Correlazione (Massima correlazione = ', num2str(max_corr),
')']);
xlabel('Lag');
ylabel('Correlazione');
ylim([-1 1]);
grid on;
yline(0, 'k--');  % Aggiunge una linea di riferimento a 0
hold off;

% Salva la figura nella cartella
saveas(gcf, fullfile(outputFolder, ['Cross-correlation (precipititazione-
livello acqua) ' num2str(currentYear) '.fig']));
close;
```

```
        % Inizializza i dati filtrati
        filteredDataP = [];
        filteredValuesP = [];
        filteredDataLd = [];
        filteredValuesLd = [];
end
```

Script (Temperature):

```
% Importa input
filename = fullfile('Roisan/', 'processedTR2.txt');
data1 = readtable(filename, 'Delimiter', '\t', 'ReadVariableNames', true);

% Combina data e ora in un unico datetime array
dateTimeArrayT = data1.Var1;
valuesT = data1.Var2;

% Importa output
filename = 'processed';
data2 = readtable(filename, 'Delimiter', '\t', 'ReadVariableNames', true);

% Combina data e ora in un unico datetime array
dateTimeArrayL = data2.Var1;
valuesL = data2.Var2;

% Crea una nuova cartella per salvare i grafici
outputFolder = 'Grafici Periodo recessione - Temperatura (CrossCorrelation)';
if ~exist(outputFolder, 'dir')
    mkdir(outputFolder);
end

% Inizializza i dati filtrati
filteredDataL = [];
filteredValuesL = [];
filteredDataT = [];
filteredValuesT = [];

% Esclude anni specifici
yearsToExclude = [2011 2012 2013];

% Ciclo for attraverso gli intervalli di anni idrologici, considerando i soli
periodi di recessione
for i = 1:size(hydrologicYearIntervals2, 1)

% Imposta le variabili inizio e fine data per l'anno idrologico corrente
startDate = hydrologicYearIntervals2(i, 1);
endDate2 = hydrologicYearIntervals2(i, 2);
```

```
% Converte le date da stringa a formato datetime
startYear = datetime(startDate, 'InputFormat', 'dd-MMM-yyyy HH:mm:ss');
currentYear = year(startYear);

% Filtra i dati all'interno dell'intervallo dell'anno idrologico corrente,
escludendo gli anni specificati
currentFilteredDataL = dateTimeArrayL(dateTimeArrayL >= startDate &
dateTimeArrayL <= endDate2 & ~ismember(year(dateTimeArrayL),
yearsToExclude));
currentFilteredValuesL = valuesL(dateTimeArrayL >= startDate & dateTimeArrayL
<= endDate2 & ~ismember(year(dateTimeArrayL), yearsToExclude));

currentFilteredDataT = dateTimeArrayT(dateTimeArrayT >= startDate &
dateTimeArrayT <= endDate2 & ~ismember(year(dateTimeArrayT),
yearsToExclude));
currentFilteredValuesT = valuesT(dateTimeArrayT >= startDate & dateTimeArrayT
<= endDate2 & ~ismember(year(dateTimeArrayT), yearsToExclude));

% Aggiunge i dati filtrati agli array finali
filteredDataL = [filteredDataL; currentFilteredDataL];
filteredValuesL = [filteredValuesL; currentFilteredValuesL];

filteredDataT = [filteredDataT; currentFilteredDataT];
filteredValuesT = [filteredValuesT; currentFilteredValuesT];

% Imposta i dati per l'anno corrente
yearDataL = filteredDataL;
yearValuesL = filteredValuesL;

yearDataT = filteredDataT;
yearValuesT = filteredValuesT;

%%
% Calcola e visualizza la cross-correlazione
yearDatai = yearDataT;
yearValuesi = yearValuesT;

% Trova gli indici comuni tra yearDatai e yearDataL
[~, idxT, idxL] = intersect(yearDatai, yearDataL);

% Considera solo i valori in comune
ascisse_comuni = yearDatai(idxT);
valoriT_comuni = yearValuesi(idxT);
valoriL_comuni = yearValuesL(idxL);

% Normalizza i valori comuni
valoriT_comuni_norm = (valoriT_comuni - mean(valoriT_comuni)) /
std(valoriT_comuni);
valoriL_comuni_norm = (valoriL_comuni - mean(valoriL_comuni)) /
std(valoriL_comuni);
```

```
% Calcola la cross-correlazione sui valori normalizzati in comune
[c, lags] = xcorr(valoriT_comuni_norm, valoriL_comuni_norm, 'normalized');

% Trova il massimo valore di correlazione
[max_corr, idx] = max(abs(c));
lag_max = lags(idx);

% Visualizza i risultati
figure;
subplot(2, 1, 1);
yyaxis left
plot(yearDataT, yearValuesT, 'k.', 'DisplayName', 'TR', 'MarkerSize', 5);
ylabel('Temperatura Aria (°C)');
ylim([-10 35]);

yyaxis right
plot(yearDataL, yearValuesL, 'r.', 'DisplayName', 'LE', 'MarkerSize', 5);
ylabel('Livello (m)');
ylim([0.08 0.19]);

xlabel('Data e Ora');
title(['Confronto tra TR e LE - Anno ', num2str(currentYear)]);
legend('show', 'Location', 'best');
grid on;

subplot(2,1,2);
plot(lags, c, 'g');
hold on;
plot(lag_max, c(idx), 'ro', 'MarkerSize', 10, 'LineWidth', 2); % Evidenzia il punto di massima correlazione
title(['Cross-Correlazione (Massima correlazione = ', num2str(max_corr), ' al lag ', num2str(lag_max), ')']);
xlabel('Lag');
ylabel('Correlazione');
ylim([-1 1]);
grid on;
yline(0, 'k--');  % Aggiunge una linea di riferimento a 0
hold off;

% Salva la figura nella cartella
saveas(gcf, fullfile(outputFolder, ['Cross-correlation (precipititazione-temperatura) ' num2str(currentYear) '.fig']));
close;

        % Inizializza i dati filtrati
        filteredDataL = [];
        filteredValuesL = [];
        filteredDataT = [];
        filteredValuesT = [];
end
```

Phase F.m

Description: Performs future predictions for water level and temperature using a model based on the Fourier function estimated in *Phase C* and a linear relationship between input and output. It imports historical data, aligns them temporally, and applies the model with a 33-day shift between water level and temperature. It generates plots comparing historical data with future predictions and saves them in a dedicated folder. Additionally, it compares the predictions with real 2024 data to evaluate their accuracy.

Script:

```
% Carica i dati
filename = 'processed';
data1 = readtable(filename, 'Delimiter', '\t', 'ReadVariableNames', true);
dateTimeArrayL = data1.Var1; % Data e ora per il livello dell'acqua
valuesL = data1.Var2;       % Valori del livello dell'acqua

filename = fullfile('Roisan/', 'processedTR2.txt');
data2 = readtable(filename, 'Delimiter', '\t', 'ReadVariableNames', true);
dateTimeArrayT = data2.Var1; % Data e ora per la temperatura
valuesT = data2.Var2;        % Valori della temperatura

% Crea una nuova cartella per salvare i grafici
outputFolder = 'Previsioni future';
if ~exist(outputFolder, 'dir')
    mkdir(outputFolder);
end

% Carica i coefficienti della funzione di Fourier per il livello dell'acqua
filename = fullfile('Grafici con modelli/', 'coeff_post2015_Fourier.txt');
coeff = readtable(filename, 'Delimiter', '\t', 'ReadVariableNames', true);
a0 = coeff.a0;
a1 = coeff.a1;
b1 = coeff.b1;
a2 = coeff.a2;
b2 = coeff.b2;
w = coeff.w;

% Allinea i dati (assicurati che L e T abbiano la stessa lunghezza e tempi
comuni)
[commonTimes, idxL, idxT] = intersect(dateTimeArrayL, dateTimeArrayT);
valuesL = valuesL(idxL);
valuesT = valuesT(idxT);
```

```
% Definisce lo shift temporale (157 giorni TOP x livello, 190 giorni TOP x
temperatura)
sD = 157; % Shift di giorni su asse x

% Calcola la componente di Fourier per il livello dell'acqua (con shift)
t = (1:length(valuesL))'; % Tempo (indice numerico)

% Calcola media e deviazione standard
L_mean = mean(valuesL);
L_std = std(valuesL);
T_mean = mean(valuesT);
T_std = std(valuesT);

% Calcola k e m
k = T_std / L_std; % Fattore di scala
m = T_mean - k * L_mean; % Offset

% Definisci le date per il 2024
startDate = datetime(2024, 1, 1); % Data di inizio: 1 gennaio 2024
endDate = datetime(2025, 1, 1); % Data di fine: 31 dicembre 2024 (devo
scrivere 1 gennaio 2025)
futureDates = startDate:endDate; % Array di date per il 2024

% Converte le date future in un indice numerico (giorni dall'inizio)
t_start = datenum(startDate);
t_future = datenum(futureDates) - t_start + 1; % Tempo in giorni dall'inizio
del 2024

% Calcola la componente di Fourier per il livello futuro (con shift)
future_L_fourier = a0 + a1 * cos(w * (t_future + sD)) + b1 * sin(w *
(t_future + sD)) + a2 * cos(2 * w * (t_future + sD)) + b2 * sin(2 * w *
(t_future + sD)); % Componente di Fourier

% Previsioni del livello dell'acqua
future_L = future_L_fourier; % Previsioni del livello dell'acqua (risultato
originale)
%future_L = future_L_fourier + ultimo_residuo; % Previsioni del livello
dell'acqua (correzione fissa del bias)

% Calcola la componente di Fourier per il livello futuro (con shift MA
sD=190)
sD = 190;
future_L_fourier = a0 + a1 * cos(w * (t_future + sD)) + b1 * sin(w *
(t_future + sD)) + a2 * cos(2 * w * (t_future + sD)) + b2 * sin(2 * w *
(t_future + sD)); % Componente di Fourier

 % Previsioni della temperatura
future_T = k * future_L_fourier + m; % Previsioni della temperatura

% Grafico delle previsioni
```

```
figure;

% Livello dell'acqua
subplot(2, 1, 1);
plot(commonTimes, valuesL, 'b', 'DisplayName', 'Data from Kalman');
hold on;
plot(futureDates, future_L, 'r', 'DisplayName', 'Forecast data');
xlabel('Date');
ylabel('Water Level (m)');
legend;
grid on;
title('Water Level:');

% Temperatura
subplot(2, 1, 2);
plot(commonTimes, valuesT, 'g', 'DisplayName', 'Original data');
hold on;
plot(futureDates, future_T, 'm', 'DisplayName', 'Forecast data');
xlabel('Date');
ylabel('Temperature (°C)');
legend;
grid on;
title('Temperature:');

% Salva la figura nella cartella
saveas(gcf, fullfile(outputFolder, ['Modelli di previsione per Livello e
Temperatura.fig']));
close;

%% LIVELLO 2024 (usando sD = 157)

filename = fullfile('Roisan/', '2024L');
data = readtable(filename, 'Delimiter', 'space', 'Format', '%s%s%f',
'ReadVariableNames', true, 'HeaderLines', 0);

% Divide la data e l'ora
dateStr = data.Var1;
timeStr = data.Var2;
values = data.Var3;

% Converte le stringhe della data in formato datetime
dates = datetime(dateStr, 'InputFormat', 'dd/MM/yyyy');
times = datetime(timeStr, 'InputFormat', 'HH:mm:ss');

% Combina data e ora in un unico datetime array
dateTimeArray = dates + timeofday(times);

figure
plot(dateTimeArray, values, ".");
hold on;
```

```
plot(futureDates, future_L, 'r', 'DisplayName', 'Livello previsto
(shiftato)');
ylim ([0.08 0.19]);
xlabel('Date');
ylabel('Water Level (m)');
legend;
grid on;
title('Water Level - 2024');

% Salva la figura nella cartella
saveas(gcf, fullfile(outputFolder, ['Previsione Livello dell acqua (Entrebin,
2024).fig']));
close;

%% TEMPERATURA 2024 (usando sD = 190)

% Carica i dati
filename = fullfile('Roisan/', '2024T');
data1 = readtable(filename, 'Delimiter', '\t', 'ReadVariableNames', true);
dateTimeArrayL = data1.Var1; % Data e ora per il livello dell'acqua
valuesL = data1.Var2;       % Valori del livello dell'acqua

figure
plot(dateTimeArrayL, valuesL);
hold on;
plot(futureDates, future_T, 'm', 'DisplayName', 'Temperatura prevista
(shiftata)');
xlabel('Date');
ylabel('Temperature (°C)');
legend;
grid on;
title('Temperature - 2024');

% Salva la figura nella cartella
saveas(gcf, fullfile(outputFolder, ['Previsione Temperatura (Roisan-Preyl,
2024).fig']));
close;
```