Università
degli Studi
di Padova

Dipartimento
di Ingegneria
dell'Informazione

# Identification of Large Scale Systems: the Po River case study

**Laureanda**
Silvia Minucelli

**Relatore**
Gianluigi Pillonetto

**Correlatore**
Damiano Varagnolo

Dipartimento di
Ingegneria
dell'Informazione

Anno 2012

# Contents

**1 Introduction** **1**

**2 The Po River case: the state of the art** **5**
2.1 Structural measures . . . . . 6
2.2 Non-structural measures: the Interregional Agency for the Po River (AIPo) system . . . . . 9
2.3 Management of the predictive uncertainty . . . . . 17

**3 Nonparametric system identification** **23**
3.1 Introduction . . . . . 23
3.2 Reproducing Kernel Hilbert Space (RKHS)-based nonparametric regression – Background . . . . . 26
3.3 Nonparametric Identification of LTI systems . . . . . 34

**4 A prediction system for the Po River and its tributaries** **45**
4.1 Database characteristics and preprocessing . . . . . 46
4.2 Training of the algorithm: settings and choice of training sets . 48
4.3 Test of the algorithm: implementation and results . . . . . 52
4.4 Mean–square error . . . . . 63

**5 Conclusions** **69**

**References** **71**

**Index** **74**

# List of acronyms

**AIPo** Interregional Agency for the Po River

**ARPA** Regional Agency for Environmental Protection

**BMA** Bayesian Model Averaging

**BIBO** Bounded Input - Bounded Output

**ECMWF** European Centre for Medium-Range Weather Forecasts

**EM** Expectation Maximization

**HUP** Hydrological Uncertainty Processor

**LAM** Limited Area Model

**LTI** Linear Time Invariant

**SISO** Single Input - Single Output

**MISO** Multiple Input - Single Output

**MIMO** Multiple Input - Multiple Output

**MCP** Model Conditional Processor

**MSE** Mean Square Error

**NAM** Nedbør Afstrømnings Model

**NQT** Normal Quantile Transform

**HEC-HMS** Hydrologic Modeling System

**HEC-RAS** Hydrologic Engineering Center River Analysis System

**pdf** probability density function

**GP** Gaussian Process

**MAP** Maximum A Posteriori

**MMSE** Minimum Mean Square Error

| | |
|---|---|
| **ML** | Maximum Likelihood |
| **LMMSE** | Linear Minimum Mean Square Error |
| **LS** | Least Squares |
| **WSN** | Wireless Sensor Network |
| **NARX** | Nonlinear AutoRegressive eXogenus |
| **NCS** | Networked Control System |
| **RKHS** | Reproducing Kernel Hilbert Space |
| **RN** | Regularization Network |
| **r.v.** | random variable |
| **r.v.** | random vector |
| **AIC** | Akaike Information Criterion |
| **BIC** | Bayesian Information Criterion |
| **PEM** | Prediction Error Methods |

# 1 Introduction

The supervising and control of the river levels have always been a relevant issue in order to prevent the damages of the flood events. The increasing urbanization and industrialization of many river basins resulted in an growth of the flood risk, therefore in the need for an accurate forecasting system to help the authorities to plan and activate interventions and emergency evacuation procedures. A reliable prediction for the water level and rate of flow is also required to plan hydrological protection works, to manage the water resources and to optimize their use for both industrial and agricultural purposes.

Several hydrologic and hydrodynamic models are currently used in order to get a forecast of the river levels and flood. They involve the processing of a great amount of data, including hydrometric and pluviometric measurements, weather forecasts and hydrogeological maps, in addition to water level, temperature and flow observations.

Such models are usually rather sophisticated, especially when describing large, complex system such as the catchment area of a major river. Most of the models – in particular the hydrodynamic ones, namely those which simulate the propagation of the flow through the basin – are physically based, therefore they aim to obtain a deterministic and physically meaningful description of the real

system. They include a huge number of parameters, and their implementation usually requires a great computational effort.

However, such an approach tends to overestimate the actual amount of information actually available on the system behavior. Moreover, none of the currently used techniques is capable of natively handling parameters uncertainty, since it is assumed that the set of the chosen values perfectly describes the real system.

In the field of Control and System Theory, identification of large scale systems is a classical research topic. It is often the case that identification algorithms tend to disregard the physics of the system, and treat the latter as a black box input–output operator. One of the most recent developments is the so called *nonparametric approach*, which is based on the idea of avoiding the postulation of a priori structures for the result, and searching a model of the system within a space of functions featuring some desirable properties of stability and smoothness.

In this thesis we adopted this approach to forecast water heights and flows on the Po River basin, which is the largest Italian catchment area, and already features a complex supervising and prediction system. Thanks to the observed data, kindly provided by the Regional Agency for Environmental Protection of Emilia–Romagna, we were able to train and test our algorithm on real datasets, and to compare the results of our prediction method with the ones of the current forecasting system.

**Overview of the thesis** The thesis is organized as follows:

- in **Chapter 2** we present the Po River case, along with the main characteristics of the basin, and we review both structural and non structural defense measures, including the current flood forecasting system;
- in **Chapter 3** we introduce the theoretical framework of the nonparametric approach, including Reproducing Kernel Hilbert Spaces theory and regularization techniques, while summarizing the possible benefits of such an approach to the Po River basin forecasts;
- in **Chapter 4** we briefly describe the implementation of the identification algorithm (including its training and testing on real datasets), show our forecasting results on both water heights and flow values, and compare the performances with those of the current system;

- in **Chapter 5** we summarize the obtained results and propose some directions for future research.

# 2

# The Po River case: the state of the art

The Po River is the largest and most important Italian river, which flows 650 km eastward from the Cottian Alps - in the North-West of Italy - into the Adriatic Sea, crossing some of the most industrialized and densely inhabited Italian regions. Through centuries, the river has always been subject to heavy flooding, therefore the need of hindering the flood events has been a major issue for every population settling in the Po Valley. The first human attempts to control the river flow with embankments and channels date back to the Etrurian age, and continued throughout history up to the present time.

It is known that since XVI century the Po river had long and continuous levees from Mantova to the Adriatic sea, covering a stretch of about 150 km. Afterwards, in particular after severe flood events, the development of embankments was extended upstream, as well as along the main tributaries. Nowadays, the levees have reached a length of about 860 km along the main course of the Po River, and about 1420 km along the most important tributaries.

Despite the fact that the overall quantity of water is lower than in the past centuries, flood risks are strongly increasing due to the massive expansion of inhabited areas close to the river path. The growing urbanization, which often involves even flood plains and other reserved areas, required the planning

of several prevention and intervention strategies, both *structural* and *non-structural*.

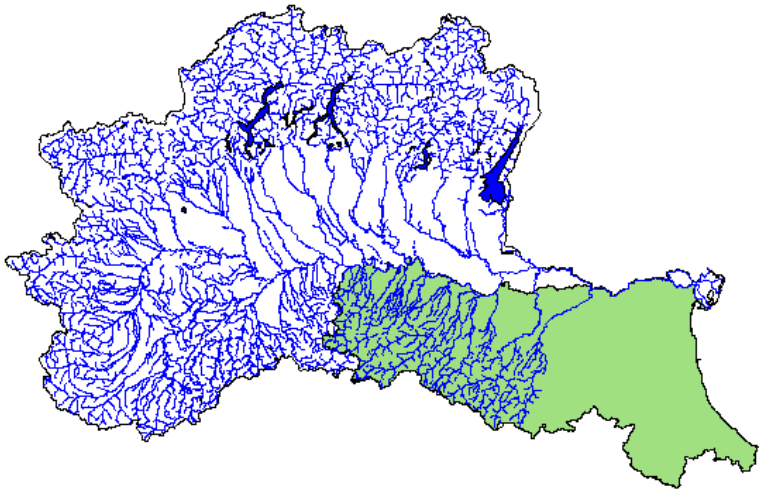

**Figure 2.1:** The Po River basin.

## 2.1 Structural measures

The definition of structural measures includes every kind of physical intervention, from the repair of the levees to the construction of dams and embankments. Several structures and strategies are currently used on the Po River in order to face both seasonal and emergency flood events. An extremely important role in stormwater management is played by *detention basins*, which are storage sites (such as reservoirs or dry ponds) that delay the flow of water downstream. Such basins not only provide general flood protection, but can also help controlling extreme floods as well as extraordinary storm events with very long return period. A detention basin allows the entrance of large flows of water, while limiting the outflow thanks to the very small opening at the lowest point of the structure. The inflow area is obviously subject to high stress, and is therefore designed to be very stout, and to protect the whole structure from damages. For example, concrete blocks are often used to reduce the speed of entering flood water. Most detention basins are built upriver of major cities, in order to protect the population.

*Floodplains* serve a similar function in beheading the flood. They are flat areas adjacent the river stream, stretching from the banks to the base of the

**Figure 2.2:** The 2000 flood of the Po River.

levees. Floodplains experience flooding during periods of high discharge, and since they can extend over very large areas, they are a fundamental resource for emergency water storage. Many of them are in fact *closed*, namely there is a second, lower levee that gets overrun during the flood event, therefore reducing the rate of the flow. The maximum storage volume of the defended floodplains all along the main trunk of the Po River is of about $410 \cdot 10^9\ m^3$. The most important ones are Roncorrente, Revere, San Benedetto Po and Sustinente.

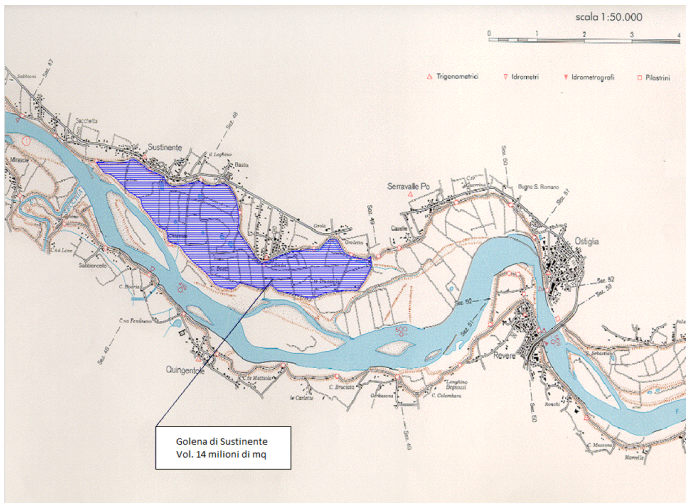


**Figure 2.3:** Sustinente floodplain map.

A major issue concerning these floodplains is that increasing urbanization leads people to settle down even in prohibited areas that are subject to inundation - in particular if the return period of the flood event in that particular site is long. Due to this problem, it is sometimes easier and cheaper - in terms of costs and organizational complexity - to build up brand new defensive structures than to relocate a whole community of settlers.

The main structural defense to contrast flood events is still the presence of a continuous system of *levees* - in the case of the Po River, as already said, the total length of the levees is about 860 km along the main trunk, and 1420 km along the main tributaries and the branching water courses of the river delta. Although the levees offer an effective way to contain the flood and reduce the inundation events, their construction and their growing extension towards the upstream part of the river caused the subtraction of significative floodplain areas, therefore slowing down the discharge process. Not only this determined a progressive and significative rise of water levels and discharge times along the Po main course, but at the same time the steady rise of the height of levees caused the achievement of structural limit conditions. At present time, the size of the embankments has reached its physical maximum along most part of the lower course of the Po River, and can not be augmented anymore.

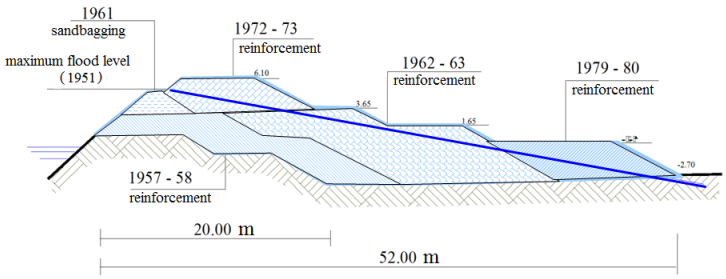


**Figure 2.4:** Evolution of the embankments after the 1951 flood event.

River embankments also need constant maintenance to prevent erosion and collapses, in particular during flood events, that can last up to three or four days. During that period the levees get hardly stressed, therefore requiring

the presence of volunteers teams to keep under surveillance the most risky stretches. Internal erosion of the embankment caused by *seepage* - also known as piping - can be fast enough to form channels underground, that follow paths of maximum permeability and result in extensive field springs. The main strategy to contrast them is that of sandbagging the whole area, in order to increase pressure and consequently reduce water speed.

**Figure 2.5:** Sandbagging around a wide field spring (*fontanazzo*).

Just as the detention basins can be opened in order to behead the flood, it is possible to break the levees at some point to let the water flow out. *Levees cuts* are considered an extreme solution, and are only used to face very serious emergency events. The exact location and timing of the break need to be carefully planned, otherwise the whole intervention might turn out to be either devastating or useless. Moreover, these operations might be rather dangerous for the workers performing the cut, which obviously need to operate under security conditions. This is one of the reasons for the need of an accurate forecasting system, along with other decisional problems such as people relocation, damages minimization etc.

## 2.2 Non-structural measures: the AIPo system

Apart from structural measures, an effective real time flood forecasting system is needed in order to manage emergency situations and defensive strategies. In the case of the Po River, such a need was particularly highlighted during the serious October 2000 flood, and the later inundation of Turin.

The organization that is currently in charge of flood protection and flood damage reduction, and of the whole forecasting system, is the Interregional Agency for the Po River (AIPo), that was established in 2003. AIPo provides engineering and environmental services in support of the Italian regions crossed by the Po river, namely Piemonte, Lombardia, Emilia - Romagna and Veneto.

AIPo efforts range from small, local protection projects to major civil engineering works, such as dams, flood control storage areas, etc., in close cooperation with national and local governments, academic institutions and other concerned groups. Since its establishment, one of the main goals of AIPo was the implementation of a flood early-warning system able to provide river and flood real-time forecasts and information about the drought along the Po River.

The Flood Forecasting and River Monitoring System was the result of a 2005 national and interregional agreement among public administrations, including the Italian Department of Civil Protection, the Po river basin Authority, the AIPo itself and of course the local governments of the interested regions, such as the Regional Agency for Environmental Protection (ARPA). The main goals of the project were:

- developing a reliable model for works management and defensive strategies planning;
- developing a suitable forecasting system for real time applications;
- providing information in advance for the Civil Protection in order to help the organization of flood control services, soil defense strategies, and emergency management.

The system is currently used by local governments in order to reduce territorial vulnerability and to plan alert strategies, and it is connected to external hydrological and meteorological data sources. Imported data include, for example, weather forecasts and telemetry systems, such as observed water levels and precipitations. Besides flood management, the forecasting system is also used to optimize the use of water resources, to provide information for fluvial navigation and to simulate crisis scenarios.

## The inputs of the AIPo forecasting system

The input of the forecasting system is a wide variety of data coming from a complex network of sensors all over the Po River basin. The acquisition takes place by a regular auto-polling via radio from over a thousand stations every thirty minutes - actually coordinated by few major stations.

The most important quantities are obviously *temperatures*, *rainfall* and *water levels*. Rainfalls are measured not only by a thick system of pluviometers (about one per 80 $km^2$, with higher density on hills and mountainous areas), but also by a network of radars. Pluviometers measure the rain depth per time unit, and provide information on both the total amount of water and the hourly intensity of the rainfall, in order to warn on extraordinary precipitation events. The radar network get instead an estimate of the rainfall field by measuring refractivity. Radar information is in general less accurate, as it just provides a rough evaluation on a scale from 1 to 5, and is only used when - for any reason - pluviometric data are not available in real time. It is in fact essential for the forecasting system to be fully and continuously updated 24/7.

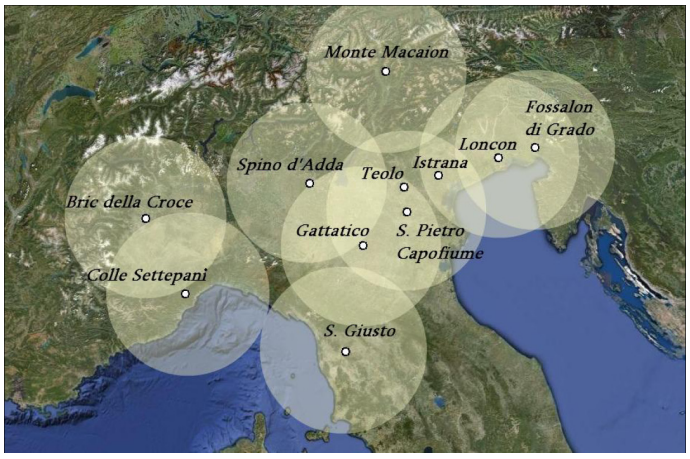


**Figure 2.6:** The radar network covering the Po River basin.

Both pressure and ultrasound sensors are used to measure stream stages along the main trunk and the major tributaries of the Po River, while in some sections flow measurements are available too, thanks to several helix devices or Doppler instruments placed in different points of each section. Without direct flow measures, it is not trivial to compute the actual mass of water flowing through a section, as the stream bed morphology can evolve through years, and therefore a certain amount of water can correspond to rather different water

stages. Moreover, water level measurements are affected by noise, in particular on the upstream part of the basin, where water streams are more turbulent and subject to sudden drifts - the so called *flash floods*. Some attention is thus needed when dealing with historic flow data series.

In addition to hydrometric and pluviometric measures, hydrogeological maps, temperature and flow observations, snow heights and water levels, information on the artificial basins are also used. There are about 180 dams across the whole Italian Alpine chain, which water level and volume data are acquired by the forecasting system with a certain retard, due to the secret required by the hydroelectric stock markets - which anyway is no longer restrictive during flood and crisis events. Information about the industrial and agricultural water use are also collected, and mainly used to optimize seasonal water drainage.

*Salt concentration* along the courses of the delta is measured for the same purpose. Along with an atmospheric circulation model, which processes both astronomic and meteorological forecasts, these measurements aim to estimate saltwater intrusion, in order to identify the most appropriate timing for fresh water drainage, which is obviously a primary concern for irrigation.

Besides *observed* data, *forecasted* data are used too. *Weather forecasts*, in particular, play a fundamental role in the prediction process. The AIPo system uses both forecasts from the European Centre for Medium-Range Weather Forecasts (ECMWF), which are computed in the ECMWF base in Reading (UK), and from a Limited Area Model (LAM). ECMWF predictions are based on a general atmospheric circulation model, namely they are the result of the integration of physics based differential equations over a whole Earth hemisphere, while LAM models only apply to limited regions, and use the information from the global models as an initial frame to develop a more accurate short-range forecast. Although LAM models are unable to perform long term forecasts, as they lack information on boundary conditions, they offer a much more reliable short term predictions, as they compute their forecasts on a very tight lattice - in the case of the Po River basin, the side of the cells is just 2.8 km, while the ECMWF cells side is about 50 km.

## The hydrologic and the hydrodynamic model

The AIPo system uses all the observed data and structural information to simulate the behavior of the Po River basin. Hydrologic models convert

rainfall information into flow forecasts, which are then used as an input for the hydrodynamic models. A cluster of more than 140 CPU cores is used to afford the computational charge of the simulations, coordinated by a master that manages resource allocation according to an open source grid computing technology (Condor). Two independent power systems are available, in order to guarantee the continuity of the system even in breakdown or emergency situations.

### The hydrologic model

All the observed data serve as inputs for the hydrologic runoff models, which aim to convert rainfall information into a flow forecast. Hydrogeological maps, as well as information on soil usage, soil composition and land morphology are used to estimate how much water per time unit is going to reach the main river and its major tributaries. The observed data only allow to get a prediction range shorter than the *time of concentration*, namely the time needed for water to flow from the most remote point in a watershed to the watershed outlet.

In order to perform longer term predictions, weather forecasts are also included in the model. Due to the intrinsic uncertainty on the *future* values of precipitation, a probabilistic approach is required to properly deal with this additional information. *It is fundamental to remark the fact that this type of uncertainty does not arise from a lack of knowledge on the reliability of the model or on the actual value of the parameters, but rather from the use of future quantities, that are therefore inevitably unknown.*

The AIPo system currently includes the predictive uncertainty on weather forecast only by perturbing the initial conditions of the ECMWF general atmospheric circulation model. Over fifty different scenarios are then generated, and subsequently divided into different groups. A single representative is then chosen from each set, according to some kind of meteorological metrics (such as pressure, altitude etc.), and used as an input to the LAM. The initial condition is provided by a mesoscale data assimilation based on a nudging technique. This procedure leads to the generation of sixteen different scenarios - plus the one obtained without any perturbation - which are computed by a CPU cluster located at the CINECA (the largest Italian computing centre).

The seventeen scenarios, along with all of the observed data, get processed by three different hydrologic runoff models:

- the MIKE11-NAM model, a commercial software which is based on Nedbør Afstrømnings Model (NAM) methods. NAMs are lumped rainfall-runoff models that describe the watershed as a single entity with a single rainfall input (mean rainfall). The discharge at the watershed outlet depends on the global dynamic of the system, and the whole drainage basin is represented as a series of storages (including soil water retention, groundwater, artificial basins etc.). Therefore, flow is calculated as a function of the water storage in each of the mutually interrelated storages that model the capacity of the catchment area. In the particular case of the Po River basin, the model includes 488 different storages;

- the Hydrologic Modeling System (HEC-HMS), which is designed to simulate the precipitation-runoff processes of branched watershed systems. A model of the watershed is constructed by separating the hydrologic cycle into single processes, represented as a series of storage layers (canopy interception storage, surface interception storage, soil storage, ground storage etc.).

- The TOPKAPI, a physically-based hydrologic model. The TOPKAPI is fully distributed, namely the river basin is divided into several cells, and a set of different components (such as interception, snowmelt, evapotranspiration, infiltration, percolation, sub-surface flow, surface flow, groundwater flow and channel flow) is applied to each cell. Cells side varies from 200 m on mountain regions up to about 1000 m on plain regions, due to the fact that weather conditions are less uniform over mountainous areas. The TOPKAPI is based upon physically meaningful parameters, and it approximates the horizontal flow of the water over and under the soil by means of a kinematic wave model. It represents flood curves starting from meteorological inputs and from morphological and physical characteristics of the hydrographical basin. The catchment behavior is then obtained by aggregating the non-linear reservoirs into three cascades, representing the soil, the surface and the drainage network, see, e.g., Todini and Ciarapica (2002).

**The hydrodynamic model**

The outputs of the hydrologic models, namely the expected flow values, are used as an input for three different hydrodynamic models, which aim to sim-

ulate the flow propagation along the river network. The models in use are MIKE11-HD (the hydrodinamic module of the Mike11 package), Hydrologic Engineering Center River Analysis System (HEC-RAS) and SOBEK (a commercial suite). They are all based on the so called *Saint-Venant* equations, namely the unidimensional form of shallow water equations:

$$\begin{cases} \dfrac{\partial Q}{\partial t} + \dfrac{\partial}{\partial x}(\alpha \dfrac{Q^2}{A}) + gA\dfrac{\partial h}{\partial x} = 0 \\ \dfrac{\partial Q}{\partial x} + \dfrac{\partial A}{\partial t} = 0 \end{cases} \tag{2.1}$$

where:

- $h$ is the water height (with respect to a fixed level)$(m)$;
- $g$ is the acceleration due to gravity $(m/s^2)$;
- $Q$ is the flow value $(m^2/s)$;
- $A$ is the area of the section $(m^2)$;
- $\alpha$ is the momentum distribution coefficient.

These equation can be derived from the momentum conservation and mass conservation laws applied to each infinitesimal section of the river. The hydrodynamic models represent the river as a series of separate stretches, each one receiving as an input the forecasted flow from the hydrological model. The effect of the incoming water is modeled as a combination of both upstream and lateral inflow. The representation of the hydrographical basin is a network based on topographic surveys coming from over 1,100 stations, in addition to the information on every structure and artificial basin interacting with the Po River catchment area.

A significative example is that of *Isola Serafini*, the largest island in the Po River, which also hosts a hydroelectric power plant. Two main barriers are present on the two branches of the river that surround the island, along with a wide reservoir for water storage. The operational rules of the diversion weir are included in the hydrodynamic models, therefore their contribution is taken into account while simulating the flow propagation.

The three hydrodynamic models (MIKE11-HD, HEC-RAS, SOBEK) use different numerical methods for the integration of shallow water equations, all

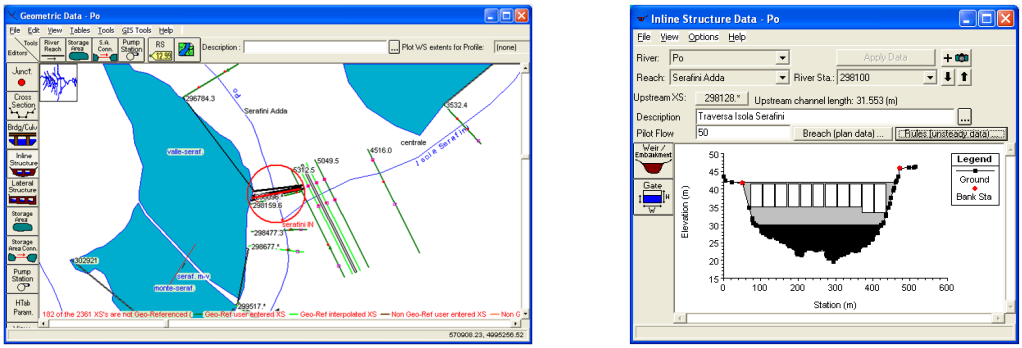


**Figure 2.7:** Managing software of the barriers of Isola Serafini.

based on finite-difference schemes. The models are one-dimensional, meaning that there is no direct modeling of the hydraulic effect of cross section shape changes, turbulence, and other two- and three-dimensional aspects of flow. Only an average value of water height and speed is used to represent each section. Therefore, the territory is modeled as a series of connected segments, and turbulent flows are simulated just on the joints between the main course of the river and its major tributaries. This leads to the so called *quasi-2D applications*, which are capable to catch the most significant aspects of water dynamics.

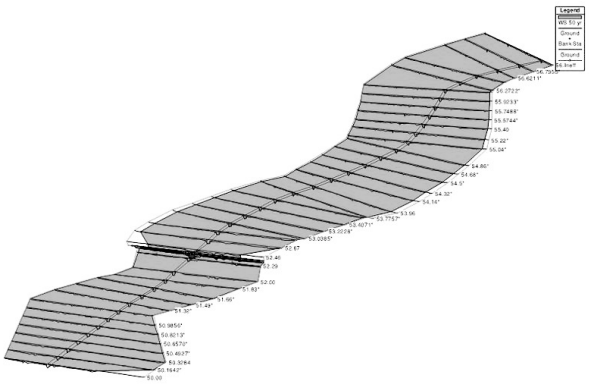

**Figure 2.8:** Graphical representation of a river stretch in the HEC-RAS system.

## How parameters are calibrated and the simulations run

*The calibration process is essential to let the model reproduce as faithfully as possible the real system behavior.* In the case of the AIPo system, the

hydrologic and the hydrodynamic models have been simultaneously calibrated by comparing the simulation results with the observed data. Rainfall and temperature data were used to estimate the parameters of the hydrologic model, which mainly depend on soil usage and composition. The flow data were similarly used to calibrate the hydrodynamical parameters, namely the river bed roughness in each section.

Remarkably, the AIPo system currently does not implement any learning method for parameters update. Only some post processing techniques are used to *correct* the forecasts on the base of the observed data.

The AIPo system features two different types of simulation. A deterministic simulation - based on the *observed* data only - runs once per day, in order to update the initial conditions of the model, namely the flow values and the water content of each reservoir. This daily run has no prediction purposes, and only aims to set the initial values of water level and soil conditions to initialize the various hydrologic models. *This initialization does not concern the model parameters, that are fixed.* A recalibration of the parameters takes place only under extraordinary circumstances, such as the construction of a new dam or watergate, or the survey of a previously unexplored area.

After the initializing run, the predictive simulation starts. A flow forecast is produced once every three hours, by using the historic data series, the real time observed data from the acquisition system, and the weather forecasts.

## 2.3 Management of the predictive uncertainty

When dealing with flood events, an accurate forecast on the future behavior of the river is essential for emergency management and decision making. Since it is impossible to achieve a perfect, deterministic prediction on water and flow values, the need arises for a reliable way to evaluate the predictive uncertainty, namely the probability of any future value conditional upon all the information available up to the present, see, e.g., Todini (2008).

Any river basin is an extremely complex system - expecially the Po River basin m being really wide and heterogeneous - therefore it is impossible to exactly model it, no matter how accurate the model might be. The approximation in the structure of the model, along with measurement errors on input and

outputs, parameters uncertainties and errors on the initial conditions prevent the implementation of a perfect model.

Still, the uncertainty on *future* flow and water stage values is not only due to model errors, but also on the fact that they depend on *future* - therefore unknown - values of rainfalls, temperature, weather conditions etc. This has nothing to do with the uncertainty due to the model, being instead a direct consequence of the fact that random processes are involved in the evolution of the system.

One of the most recent approaches to flood forecasting lies on the attempt of *including in the forecast system a probabilistic description of the quantity of interest* (Todini, 2010). The basic idea is that of providing an optimal decision strategy by maximizing an appropriate utility/damage function, such as, for example,

$$\begin{cases} U(y_t) = 0 & \text{if } y_t \leq y_D \\ U(y_t) = g(y_t - y_D) & \text{if } y_t > y_D \end{cases} \tag{2.2}$$

where $g(\cdot)$ represents a generic function relating the cost of damages and losses to the water stage, and $y_D$ expresses a certain level that should not be exceeded. I.e., $U(y_t)$ might reflect the damages that will actually occur at a certain future time $t$ if the water level $y_t$ overtops the dyke level $y_D$. In flood management operations, the future value $y_t$ is obviously unknown, therefore the manager can only take his decision on the basis of expected utility $\mathbb{E}[U(y_t)]$, which could be computed using a prior assessment of the predictive uncertainty $f_0(y_t)$ as

$$\mathbb{E}[U(y_t)] = \int_0^\infty U(y_t) f_0(y_t) dy_t \ . \tag{2.3}$$

Unfortunately, the a priori probability density $f_0(y_t)$ is generally quite flat, thus resulting in an unreliable estimate of the utility expectation. This leads to the attempt of gathering additional information in order to produce a denser posterior pdf, conditional on all the available information up to the present time (including both direct measurements and additionally generated information, such as model forecasts $\hat{y}_{t|t_0}$). Equation 2.3 can therefore be rewritten as

$$\mathbb{E}\left[U\left(y_t|\hat{y}_{t|t_0}\right)\right] = \int_0^\infty U(y_t) f_{y_t|\hat{y}_{t|t_0}}(y_t|\hat{y}_{t|t_0}) dy_t \ , \tag{2.4}$$

which is a more efficient estimator of the expected utility, as $f_{y_t|\hat{y}_{t|t_0}}(y_t|\hat{y}_{t|t_0})$ is usually less dispersed around its mean than $f_0(y_t)$. In other words, its variance

is smaller, sometimes significantly.

We then remark that there exist thus two different kinds of uncertainties:

**emulation uncertainty,** namely the probability density of a model forecast given the knowledge of the occurred event;

**prediction uncertainty,** namely the probability density of a future event given the knowledge of the model forecast.

In order to highlight the differences between these uncertainties, we plot some observed and model predicted values are a scatter plot in Figure 2.9, and then highlight which are the emulation and prediction uncertainties.

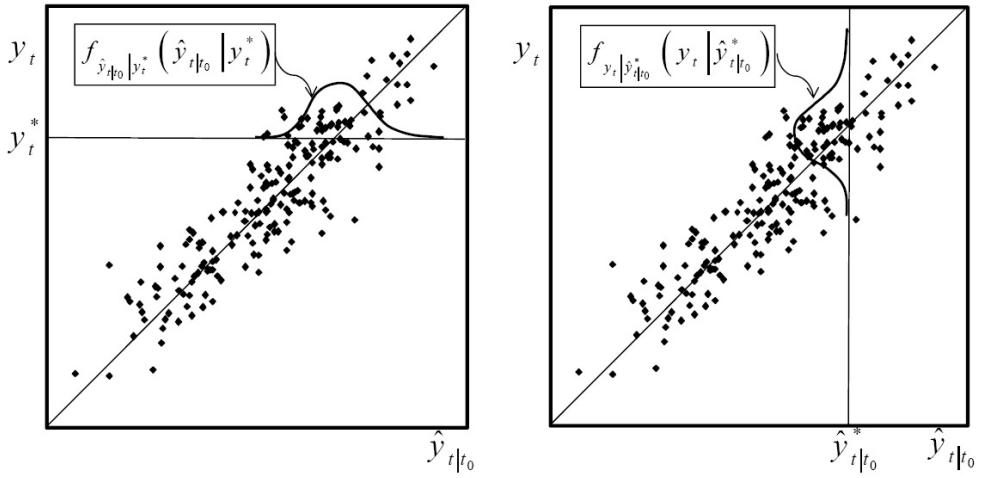


**Figure 2.9:** Graphical representation of *emulation uncertainty* (left) and *predictive uncertainty* (right).

Emulation uncertainty corresponds thus to the spread of the predictions around the real observed value. The emulation probability density, namely the pdf of the model predictions conditional upon the observed value $y_t^*$ (see the left panel of Figure 2.9), can thus be used to reduce errors by properly adjusting the model. Nonetheless, emulation uncertainty can not be used to obtain predictions, since the conditioning variables - namely the observations - are not available for future times. In other words, emulation uncertainty is essential when aiming at model validation or improvement, but meaningless in terms of predictions.

On the contrary, predictive uncertainty $f_{y_t|\hat{y}^*_{t|t_0}}(y_t|\hat{y}^*_{t|t_0})$, namely the pdf of the future and unknown value of $y$ given a specific model prediction $\hat{y}^*_{t|t_0}$, can

be used to extend predictions to the future (see the right panel of Figure 2.9). Notice that in this case the probability of occurrence of the model forecast is obviously 1.

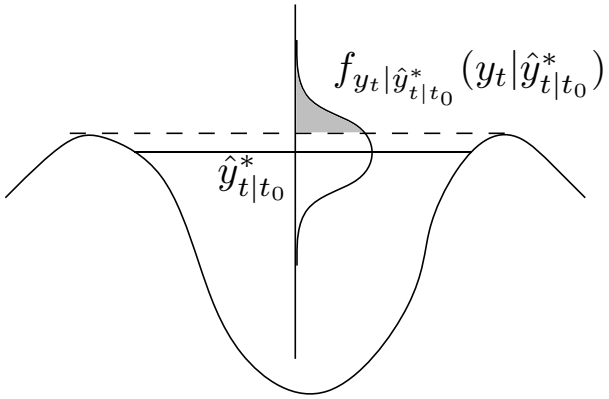


**Figure 2.10:** A graphical representation of the probabilistic measure of flooding conditional upon a predicted water level.

Three different approaches are currently available to assess predictive uncertainty, namely the Bayesian Hydrological Uncertainty Processor (HUP) developed by Krzysztofowicz (1999), the Bayesian Model Averaging (BMA) introduced by Raftery (Bollen and Long, 1993), and the Model Conditional Processor (MCP) due to Todini (2008).

The Hydrological Uncertainty Processor (HUP) aims at estimating predictive uncertainty given a set of historical observations and a hydrological model prediction. It is based on the idea of converting both observations and model predictions into a Normal space by means of the Normal Quantile Transform (NQT) (Van der Waerden, 1952, 1953a,b), in order to exploit the Normal distribution properties to derive the joint distribution and the predictive conditional distribution from an analytically treatable multivariate distribution. The main limitations affecting the HUP are the impossibility to extend it to multi-model forecasts, the fact that it is based on a AR (Auto Regressive) model which seems not to be adequate to represent the rising limb of the flood wave, and the hypothesis of independence of the AR model errors from the prediction model errors (which are usually correlated instead).

The Bayesian Model Averaging (BMA) aims at assessing just the unconditional mean and variance of any future value of the quantity of interest on the basis of several model forecasts. Differently from the HUP assumptions, all the models are here considered as possible alternatives, and weighted according to

the result of the constrained optimization problem

$$\begin{aligned} \max_{w_j} \quad & \log \mathcal{L} = \sum_{i=1}^{n} \log \left( \sum_{j=1}^{m} w_j p_j \left( y_t \;\middle|\; \hat{y}_t^{(j)} \right) \right) \\ \text{s.t.} \quad & \sum_{j=1}^{m} w_j = 1 \\ & w_j \geq 0 \quad \forall j = 1, \ldots, m \; . \end{aligned} \tag{2.5}$$

The BMA assumes all the model forecasts and the variables to be predicted to be approximately Normally distributed. Moreover it computes the unconditional mean on the basis of the estimated weights

$$\mathbb{E}\left[ y_t \mid I_{t_0} \right] = \sum_{j=1}^{m} w_j \mathbb{E}\left[ y_t \;\middle|\; \hat{y}_{t|t_0}^{(j)} \right] \; , \tag{2.6}$$

providing also an approximated value of the unconditional variance.

Still, it was found that the Expectation-Maximization (EM) algorithm proposed by Raftery to solve (2.5) does not always converge to the maximum of the likelihood, therefore requiring the development of additional optimization tool.

To conclude, the Model Conditional Processor (MCP) aims to assess the probability density of the predictand conditional on all the model forecasts available at the present time. Like the HUP, the MCP converts both the observed data and the forecasts into their Normal space images via the NQT, by assuming their joint distribution to be approximatively multivariate Normal. The next step is that of deriving the distribution of the predictand NQT image conditional on the image of the observations. In other words, the MCP is a multivariate extension of the HUP approach, and thanks to the additional hypothesis on the joint distributions is no more limited to the choice of an AR model, therefore allowing a generalization both to physically based models and data driven models.

# 3

# Nonparametric system identification

## 3.1 Introduction

The previously analyzed state of the art refers to techniques that are "parametric", in the sense that the models have all been derived on the basis of Partial Differential Equations, depending on a certain fixed set of parameters. While dealing with such a complex system as the Po River basin, an extremely large number of parameters is used, both physically meaningful and devoid of any physical interpretation. Still, none of the currently used techniques is capable of managing parameters uncertainty and variability in a natural way. The model calibration is performed just once, and only retrained when some major event occurs.

For reasons that will be clear later, these parametric methods implicitly assume a perfect knowledge of the physics of the system, i.e., they assume that the actual model lies on a perfectly known and rather restrictive set of possible hypotheses. When dealing with complex systems as river basins, this assumption is often not sufficiently motivated by the amount of information actually available about the system.

We now introduce the field of "nonparametric" identification and estimation.

Both in regression and system identification, the adjective "nonparametric" usually refers to techniques that do not fix a priori any structure for the result. This lack of structure may initially appear as a negative characteristic, while, on the contrary, years of application on real fields showed that their usage is supported by various practical and mathematical reasons, such as:

- if there is a lack of knowledge on the model to be identified, or if the model is known to belong to a family of different parametric models, then nonparametric identification leads to better estimates (Pillonetto and De Nicolao, 2010). A specific example is Pillonetto et al. (2011), where authors prove that in some practical cases the identification of linear systems through combination of classical model selection strategies, like Akaike Information Criterion (AIC) (Akaike, 1974) or Bayesian Information Criterion (BIC) (Schwarz, 1978), and Prediction Error Methods (PEM) strategies ( Ljung (1999); Söderström and Stoica (1989)) performs worse than identification through nonparametric Gaussian regression approaches;

- nonparametric identification approaches can be consistent where parametric approaches fail to be (Smale and Zhou, 2007; De Nicolao and Ferrari-Trecate, 1999);

- in general, nonparametric approaches require the tuning of very few parameters, allowing the implementation of fast line search strategies (Pillonetto and Bell, 2007);

- for some parametric models, the distributed implementation of Maximum Likelihood (ML) strategies could be infeasible, due to the structure of the likelihood function. An approach is then to convexify - in a sense that will be clear later - the likelihood through the construction of a suitable nonparametric approximated model. This strategy allows the application of generic distributed optimization techniques (Bertsekas and Tsitsiklis, 1997). Under particular choices of the cost and regularization functions, we will show that the ML problem can be distributedly solved through an approximated Regularization Network (RN) requiring small computational and communication efforts and limited memory allocation.

Another important point is the following: the amount of prior information used while using nonparametric techniques (e.g., the kernel functions introduced

below, that can be considered as *covariances* whenever using Bayesian approaches based on Gaussian processes, see Section 3.2) is far less than the total amount of prior information that is given assuming the model to be a certain parametric function. Intuitively, the prior of the nonparametric techniques is *weaker* than the parametric one, and this eventually makes the nonparametric strategies more widely applicable and more robust. Nonetheless this is a tricky point. In fact, should an experiment return a small amount of data, small information would be available to perform the identification. In such a case, if the parametric model at disposal is in some sense *accurate*, the amount of information could be sufficient to obtain an estimate far better than the one that could be obtained with the less informative nonparametric prior, which needs to exploit part of the available information in order to select the model. As an example, should we know a priori that the actual function to be identified is exactly an exponential, and should there be no measurement noise, two samples would be enough to perform an exact identification through parametric techniques. On the contrary, nonparametric techniques tend to obtain better performances when a sufficient number of (eventually) noisy data is available to identify very complex systems.

## The nonparametric identification framework applied to the Po River case

The Po River basin is an extremely complex system, whose behavior depends on a large number of factors. Not only its complicated dynamics involves the interaction of many heterogenous components, but it also evolves through time in a quite unpredictable way (the evolvement can depend on both human interventions and natural processes, such as erosion, sedimentation and so on).

Due to its intrinsic time variability and complexity, the Po River modeling represents a challenging issue. Parametric approaches, in particular, tend to require the setting of a huge number of parameters in order to catch the dynamics of the system, which leads to several problems, such as:

- the need for a great computational capability (e.g. the current AIPo forecasting system exploits a cluster of more than 140 CPUs coordinated by 16 different servers to compute the outputs of the hydrodynamic part only);

- the unfeasibility of parameters retuning, therefore the incapability of taking into account the possible evolution of the basin;
- the impossibility to manage parameters uncertainty, as the model is assumed to be fixed and deterministic.

Our prospect while implementing a nonparametric approach is that of capturing both system variability and parameters uncertainty in a natural way, thus allowing easier forecasts - and more efficient from a computational viewpoint - and also offering the possibility of of on-line retraining procedures, as it will be shown in the following sections.

We now briefly describe the general theory of RKHS-based nonparametric regression and identification of Linear Time Invariant systems

## 3.2 RKHS-based nonparametric regression – Background

From an intuitive point of view, RKHSs are sets of sufficiently-smooth functions with some nice mathematical properties. The theory was founded by Aronszajn (1950). See also Yosida (1965); Cucker and Smale (2002); Poggio and Girosi (1990); Wahba (1990). For an overview of their uses in statistical signal processing see Weinert (1982).

**Definition 3.2.1** (Reproducing kernel Hilbert space)**.** Let $\mathcal{H}_K$ be a Hilbert space of functions[1]

$$f(\cdot) : \mathcal{X} \subseteq \mathbb{R}^d \mapsto \mathbb{R} \tag{3.1}$$

endowed with the inner product $\langle \cdot, \cdot \rangle_{\mathcal{H}_K}$ and norm $\|f\|_{\mathcal{H}_K} := \sqrt{\langle f, f \rangle_{\mathcal{H}_K}}$. If there exists a function

$$K(\cdot, \cdot) : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R} \tag{3.2}$$

such that

(a) $K(x, \cdot) \in \mathcal{H}_K$ for every $x \in \mathcal{X}$

(b) $\langle f(\cdot), K(x, \cdot) \rangle_{\mathcal{H}_K} = f(x)$ for every $x \in \mathcal{X}$ and $f \in \mathcal{H}_K$

then $\mathcal{H}_K$ is said to be a *reproducing kernel Hilbert space* with kernel $K$.

[1] We restrict our analysis only real-valued functions even if the same concepts could be applied to complex-valued functions.

Property (b) is usually called the *reproducing property*. Notice that $\mathcal{L}^2$ is not a RKHS since its representing functions, namely the delta functions, are not in $\mathcal{L}^2$. For the following derivations it is necessary to introduce some definitions.

**Definition 3.2.2** (Positive-definite kernel)**.** A kernel $K$ is said to be *positive-definite* if, for every $N \in \mathbb{N}_+$ and $N$-tuple $x_1, \ldots, x_N \in \mathcal{X}$

$$\begin{bmatrix} K(x_1, x_1) & \cdots & K(x_1, x_N) \\ \vdots & & \vdots \\ K(x_N, x_1) & \cdots & K(x_N, x_N) \end{bmatrix} =: \mathbf{K} \geq 0 \tag{3.3}$$

where the inequality has to be intended in a matricial positive-semidefinite sense.

**Definition 3.2.3** (Symmetric kernel)**.** A kernel $K$ is said to be *symmetric* if $K(x, x') = K(x', x)$ for all $x, x' \in \mathcal{X}$.

**Definition 3.2.4** (Mercer kernel)**.** A symmetric positive-definite kernel $K$ is said to be a *Mercer kernel* if it is also continuous.

The term *kernel* derives from the theory of integral operators, where, given a non-degenerate measure[2] $\mu$ and a function $K$ as in 3.2, it is possible to define the integral operator

$$L_{K,\mu}[g](x) := \int_{\mathcal{X}} K(x, x')\, g(x')\, d\mu(x') \ . \tag{3.4}$$

Operator $L_{K,\mu}[\cdot]$ is said to be *positive definite* if $K$ is positive definite.

The following theorem proves the biunivocity between symmetric positive-definite kernels and RKHSs.

**Theorem 3.2.5** (Moore-Aronszajn Aronszajn (1950))**.** *For every symmetric positive-definite kernel $K$ there exists an unique RKHS $\mathcal{H}_K$ having $K$ as its reproducing kernel. Viceversa, the reproducing kernel of every RKHS $\mathcal{H}_K$ is unique.*

Having in mind our future applications on regression, we focus now on the implications of the spectral theory of compact operators on RKHS theory[3].

[2]We recall that a Borel measure $\mu$ is said to be non-degenerate w.r.t. the Lebesgue measure $\mathcal{L}^2$ if $\mathcal{L}^2(A) > 0 \Rightarrow \mu(A) > 0$ for every $A$ in the Borel $\sigma$-algebra.

[3]See (Zhu, 2007, Chap. 1.3) for more details on compact operators on general Hilbert spaces.

Assume then $\mathcal{X}$ to be compact, $K$ to be Mercer on $\mathcal{X} \times \mathcal{X}$, $\mathcal{L}^2(\mu)$ to be the set of the Lebesgue square integrable functions under the non-degenerate measure $\mu$. A function $\phi$ that obeys the integral equation[4]

$$\lambda \phi(x) = L_{K,\mu}[\phi](x) \tag{3.5}$$

is said to be an *eigenfunction* of $L_{K,\mu}[\cdot]$ with associated eigenvalue $\lambda$. The following result holds.

**Theorem 3.2.6** (Cucker and Smale (2002), see also König (1986))**.** *Let $K$ be a Mercer kernel on $\mathcal{X} \times \mathcal{X}$ and $\mu$ a non-degenerate measure. Let $\{\phi_e\}$ be the eigenfunctions of $L_{K,\mu}[\cdot]$ normalized in $\mathcal{L}^2(\mu)$, i.e. s.t.*

$$\int_{\mathcal{X}} \phi_e(x) \phi_l(x) d\mu(x) \delta_{el} \tag{3.6}$$

*with corresponding eigenvalues $\lambda_e$ ordered s.t. $\lambda_1 \geq \lambda_2 \geq \ldots$. Then*

*(a) $\lambda_e \geq 0$ for all $e$;*

*(b)* $\sum_{e=1}^{+\infty} \lambda_e = \int_{\mathcal{X}} K(x,x) d\mu(x) < +\infty$

*(c) $\{\phi_e\}_{e=1}^{+\infty}$ is an orthonormal basis for $\mathcal{L}^2(\mu)$*

*(d) the RKHS $\mathcal{H}_K$ associated to $\{\phi_e\}_{e=1}^{+\infty}$ is given by*

$$\mathcal{H}_K := \left\{ g \in \mathcal{L}^2(\mu) \;\; s.t. \;\; g = \sum_{e=1}^{\infty} a_e \phi_e \;\; with \;\; \{a_e\} \;\; s.t. \;\; \sum_{e=1}^{\infty} \frac{a_e^2}{\lambda_e} < +\infty \right\} \tag{3.7}$$

*(e) $K$ can be expanded via the relation*

$$K(x,x') = \sum_{e=1}^{\infty} \lambda_e \phi_e(x) \phi_e(x') \tag{3.8}$$

*where the convergence of the series is absolute and uniform[5] in $\mathcal{X} \times \mathcal{X}$.*

[4] In some cases eigenvalues and eigenfunctions can be computed in closed forms, specially in Gaussian cases Zhu et al. (1998). Often it is necessary to perform numerical computations De Nicolao and Ferrari-Trecate (1999), (Rasmussen and Williams, 2006, Chap. 4.3.2).

[5] This has the nice practical implication that it is possible to compute $K$ with the desired level of precision using a finite number of eigenfunctions.

*Remark* 3.2.7. Condition $\sum_{e=1}^{\infty} \frac{a_e^2}{\lambda_e} < +\infty$ expressed in (3.7) can be seen as a smoothness condition. In fact, since the sequence $\lambda_1, \lambda_2, \ldots$ has to vanish because the associated series is convergent, it follows that $a_e^2$ must vanish sufficiently fast.

From the same theorem it follows that if $g_1 = \sum_{e=1}^{+\infty} a_e \phi_e$ and $g_2 = \sum_{e=1}^{+\infty} a'_e \phi_e$ then their inner product is

$$\langle g_1, g_2 \rangle_{\mathcal{H}_K} = \sum_{e=1}^{+\infty} \frac{a_e \cdot a'_e}{\lambda_e} \ . \tag{3.9}$$

Notice that, if $g = \sum_{e=1}^{+\infty} a_e \phi_e \in \mathcal{H}_K$ and $\mathbf{a} = [a_1, a_2, \ldots]^T$, orthogonality of eigenfunctions in $\mathcal{L}^2(\mu)$ implies that

$$\|g\|^2_{\mathcal{L}^2(\mu)} = \sum_{e=1}^{+\infty} \sum_{l=1}^{+\infty} a_e a_l \int_{\mathcal{X}} \phi_e(x) \, \phi_l(x) \, d\mu(x) = \|\mathbf{a}\|_2^2 \ . \tag{3.10}$$

Moreover orthonormality of eigenfunctions in $\mathcal{L}^2(\mu)$ implies orthogonality in $\mathcal{H}_K$, i.e.

$$\langle \phi_e, \phi_l \rangle_{\mathcal{L}^2(\mu)} = \delta_{el} \qquad \Leftrightarrow \qquad \langle \phi_e, \phi_l \rangle_{\mathcal{H}_K} = \frac{1}{\lambda_e} \delta_{el} \ . \tag{3.11}$$

In the following we will use the shorthands $\|\cdot\|_\mu$ for $\|\cdot\|_{\mathcal{L}^2(\mu)}$ and $\|\cdot\|_K$ for $\|\cdot\|_{\mathcal{H}_K}$.

*Remark* 3.2.8. We could have defined $\mathcal{H}_K$ using the so-called *reproducing kernel map construction* (Rasmussen and Williams, 2006, page 131), i.e. starting from the representing functions $K(x, \cdot)$. We preferred to use eigenfunctions-eigenvalues decompositions because these will be heavily used in the following sections.

## Examples of RKHSs

In this section we offer a couple of examples of the some commonly used kernels, focusing on the case $\mathcal{X} = [0, 1]$, namely Gaussian and Laplacian kernels. A third important case is the Spline kernel, but its treatment is postponed to the next section, since it is the one used by the implemented identification algorithm. We send the reader back to (Schölkopf and Smola, 2001, Chap. 13) and references therein for general kernels design techniques.

**Gaussian Kernels** A Gaussian kernel is described by

$$K(x, x') = \exp\left(-\frac{\|x - x'\|_2^2}{2\sigma^2}\right) \tag{3.12}$$

where $x, x' \in \mathcal{X} \subset \mathbb{R}^d$ ($\mathcal{X}$ is a compact). This kernel may have eigenfunctions and eigenvalues in closed forms, depending on $\mu$, see for example Zhu et al. (1998).

In Figures 3.1 and Figure 3.2 we plot the first 4 eigenfunctions for the cases $\mu = \mathcal{U}[0, 1]$ and $\mu = \mathcal{N}(0.5, 0.01)$, both with $\sigma^2 = 0.01$. We notice how the approximation capability of the eigenfunctions is concentrated where it is more probable to have measurements. In Figure 3.3 we show the behavior of the eigenvalues for the two different $\mu$'s. Finally in Figure 3.4 we show 4 different realizations $f_\mu$ relative to the kernel just considered, under the assumptions of Section 3.2.

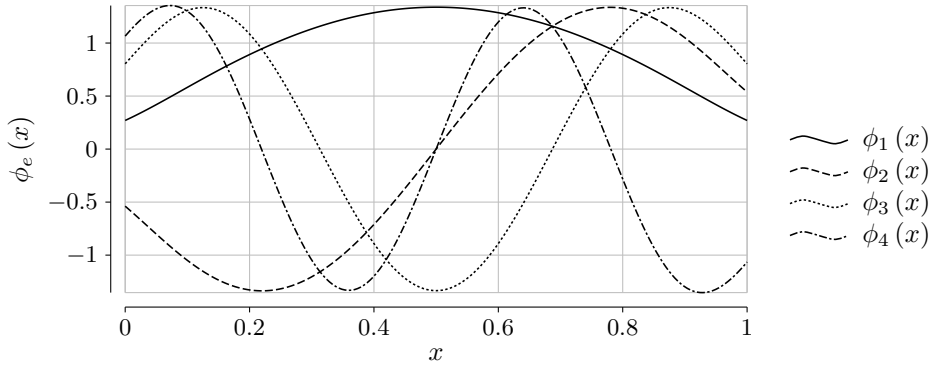


**Figure 3.1:** First 4 eigenfunctions for the Gaussian kernel (3.12), associated to $\mu = \mathcal{U}[0, 1]$ and $\sigma^2 = 0.01$.

**Laplacian Kernels** A Laplacian kernel is described by

$$K(x, x') = \exp\left(-\frac{|x - x'\|}{\sigma}\right) \tag{3.13}$$

where $x, x' \in \mathcal{X} \subset \mathbb{R}^d$, $\sigma \in \mathbb{R}_+$.

In Figure 3.5 we plot the first 4 eigenfunctions for the case $\mu = \mathcal{U}[0, 1]$ with $\sigma = 0.1$. In Figure 3.6 we show the behavior of the eigenvalues for this kernel, and in Figure 3.7 we show 4 different realizations $f_\mu$ relative to the kernel just considered, again under the assumptions of Section 3.2.

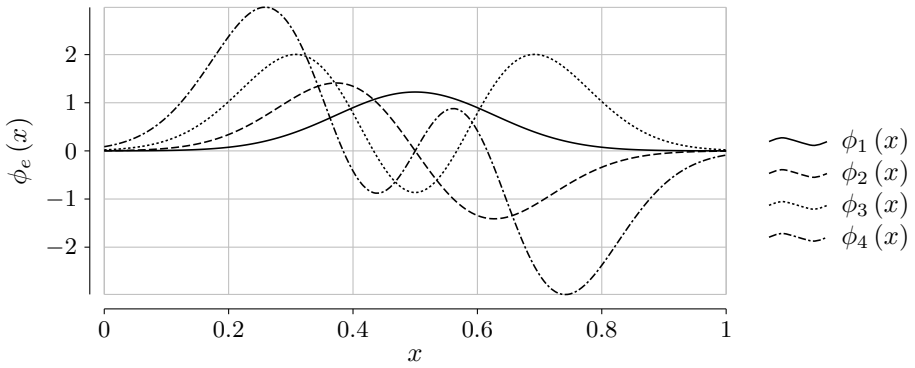


**Figure 3.2:** First 4 eigenfunctions for the Gaussian kernel (3.12), associated to $\mu = \mathcal{N}(0.5, 0.01)$ and $\sigma^2 = 0.01$.

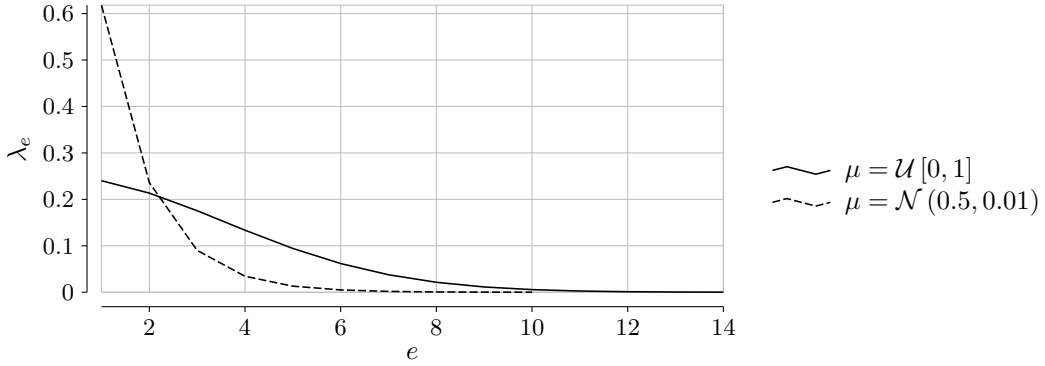


**Figure 3.3:** Eigenvalues of the Gaussian kernel (3.12), associated to $\sigma^2 = 0.01$ and different measures $\mu$.

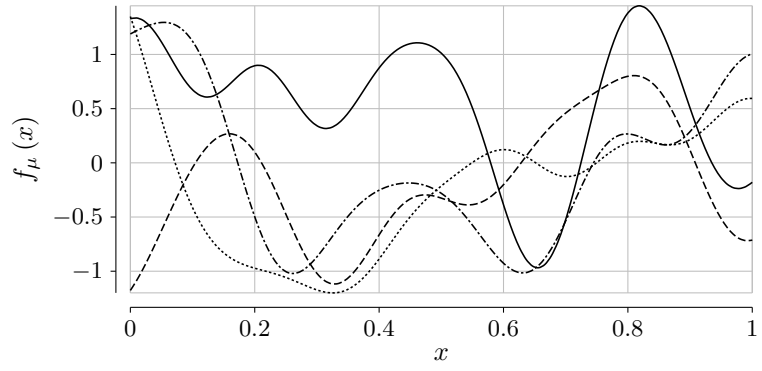


**Figure 3.4:** Independently generated realizations for the Gaussian kernel (3.12), associated to $\sigma^2 = 0.01$.

## Regularized regression

Let $f_\mu : \mathcal{X} \to \mathbb{R}$ denote an unknown deterministic function defined on the compact $\mathcal{X} \subset \mathbb{R}^d$. Assume we have the following $S$ noisy measurements

$$y_i = f_\mu(x_i) + \nu_i, \qquad i = 1, \ldots, S \tag{3.14}$$

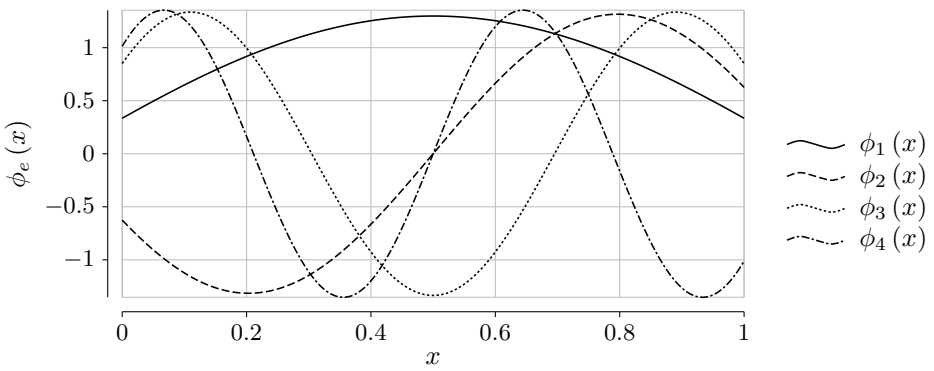


**Figure 3.5:** First 4 eigenfunctions for the Laplacian kernel (3.13), associated to $\mu = \mathcal{U}[0,1]$ and $\sigma = 0.1$.

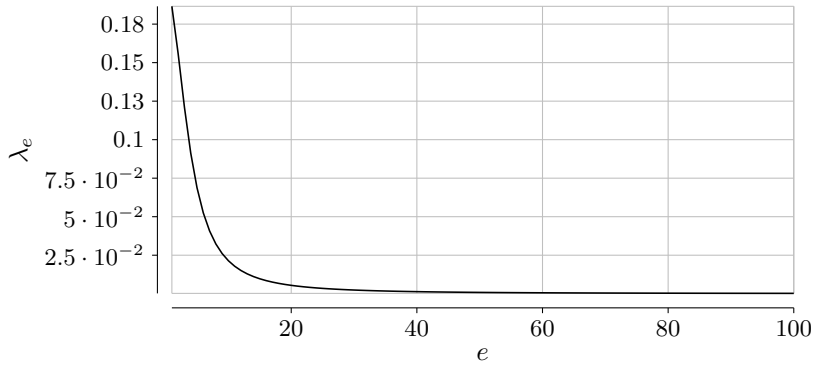


**Figure 3.6:** Eigenvalues of the Laplacian kernel (3.13), associated to $\mu = \mathcal{U}[0,1]$ and $\sigma = 0.1$.

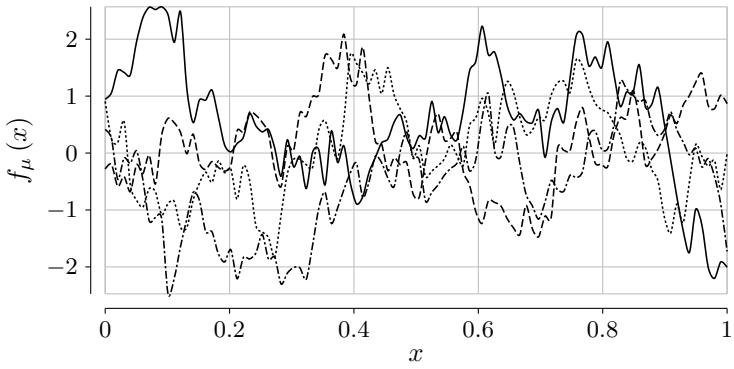


**Figure 3.7:** Independently generated realizations for the Laplacian kernel (3.13), associated to $\sigma = 0.1$.

with $\nu_i$ white noise and $i$ the measurement index. Without any additional assumption, the problem of inferring $f_\mu$ given the data set $\{x_i, y_i\}_{i=1}^S$ is ill-posed in the sense of Hadamard. One of the most used approaches to overcome this

problem relies upon the Tikhonov regularization theory[6][7] Tikhonov and Arsenin (1977), that relies computing the estimate of the unknown function as

$$\widehat{f_{\mathrm{c}}} := \arg \min_{f \in \mathcal{H}_K} Q(f) \tag{3.15}$$

where the functional $Q(\cdot)$ is defined as

$$Q(f) := L\left(f, \{x_i, y_i\}_{i=1}^{S}\right) + \gamma \|f\|_K^2 \tag{3.16}$$

and where the hypothesis space $\mathcal{H}_K$ is typically given by the reproducing kernel Hilbert space induced by the Mercer kernel $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. The first term is a loss function accounting for data-fitting properties of $f$ and related comments), while the second term, usually called *regularizer*, weights the smoothness of $f$, penalizing thus non-smooth solutions[8]. Finally, $\gamma$ is the so called *regularization parameter* that trades off empirical evidence and smoothness information on $f_\mu$.

By using the famous *representer theorem* (introduced in Kimeldorf and Wahba (1971), see (Schölkopf and Smola, 2001, Chap. 4.2) for a generalized version) it is possible to show that each minimizer of $Q(f)$ has the form of a linear combination of $S$ basis functions, i.e.

$$\widehat{f_{\mathrm{c}}} = \sum_{i=1}^{S} c_i K(x_i, \cdot) \tag{3.17}$$

i.e. $\widehat{f_{\mathrm{c}}}$ admits the structure of a *Regularization Network* (RN), term introduced in Poggio and Girosi (1990) to indicate estimates of the form (3.17).

A graphical intuition of (3.17) is that the optimal estimate is given by a combination of some "slices" of the kernel function.

In sight of the Bayesian interpretation that will be introduced in Section 3.2, our choice for the cost function is

$$Q(f) := \sum_{i=1}^{S} (y_i - f(x_i))^2 + \gamma \|f\|_K^2 \tag{3.18}$$

[6]Alternatively one could use explicit prior knowledge, and formulate the problem -for example- through Gaussian Processes formalisms.

[7]Finite-dimensional formulation of this approach is also known as *Ridge regression* Hoerl and Kennard (2000)

[8]See Girosi et al. (1995) for smoothness functionals involving Fourier transforms of the candidate estimating function.

that correspond to obtain the coefficients $c_i$ by means of

$$\begin{bmatrix} c_1 \\ \vdots \\ c_S \end{bmatrix} = (\mathbf{K} + \gamma I)^{-1} \begin{bmatrix} y_1 \\ \vdots \\ y_S \end{bmatrix} \tag{3.19}$$

with

$$\mathbf{K} := \begin{bmatrix} K(x_1, x_1) & \cdots & K(x_1, x_S) \\ \vdots & & \vdots \\ K(x_S, x_1) & \cdots & K(x_S, x_S) \end{bmatrix} . \tag{3.20}$$

### Bayesian interpretation

The estimate $\widehat{f}_c$ in (3.15) computed through (3.19) admits also a Bayesian interpretation. In fact, if $f_\mu$ is modeled as the realization of a zero-mean, not-necessarily stationary Gaussian random field with covariance $K$, if the noises $\nu_i$ are Gaussian and independent of the unknown function and with variance $\sigma^2$, once we set $\gamma = \sigma^2$ it follows that Kimeldorf and Wahba (1970); Zhu et al. (1998)

$$\widehat{f}_c(x) = \mathbb{E}[f_\mu(x) \mid x_1, y_1, \ldots, x_S, y_S] . \tag{3.21}$$

We recall that, using the Bayesian point of view and a Gaussian Processs (GPs) based formulation, it is straightforward to derive not only the estimate (to be intended as the maximum a-posteriori of the conditional density), but also to characterize the uncertainity of the prediction by means of the a-posteriori covariance. Moreover GPs formulation is closely related to *Kriging* techniques Stein (1999), usually used for interpolation of spatial data.

## 3.3 RKHS-based nonparametric identification of Linear Time Invariant (LTI) systems and the `SSpline.m` procedure

This section is devoted to the application of nonparametric identification to the case of linear time–invariant (LTI) systems, as well as to a brief description

of the implemented algorithm, `SSpline`. We refer to Pillonetto and De Nicolao (2012) for a more extended treatment and some nice examples of application of `SSpline`.

## Introduction

As recalled in the Introduction to this Chapter, a classic approach to identification of LTI systems is based on Prediction Error Methods (Ljung, 1999; Söderström and Stoica, 1989), which is in turn a particular application of the Maximum Likelihood estimation technique. As a matter of fact, this approach requires to fix a model for the system, namely, the order of the polynomials in the transfer functions must be known. Under this and other assumptions, e.g. the innovation to be Gaussian white process, the signals to be stationary and so on, it is well known that PEM methods are consistent and correct at least asymptotically, namely, if a large number of data samples is available. As already stated, however, the procedures for the estimate of the model structure, such as AIC or BIM criteria, do not always guarantee an optimal performance, and, moreover, their results are usually hard to analyze from a theoretical point of view.

Here the approach is different, and aims to directly identifying the impulse response of the system. The naive technique for impulsive response identification is to exploit the convolutional representation of LTI systems

$$y(t) = (u * h)(t). \tag{3.22}$$

Once we stack the outputs and the impulse response in vectors

$$Y = \begin{bmatrix} y_1 \\ \vdots \\ y_N \end{bmatrix} \qquad \text{and} \qquad H = \begin{bmatrix} h_1 \\ \vdots \\ h_N \end{bmatrix}$$

and we build the matrix

$$U = \begin{bmatrix} u_1 & 0 & 0 & \cdots & 0 \\ u_2 & u_1 & 0 & \cdots & 0 \\ \vdots & & & & \vdots \\ u_N & u_{N-1} & u_{N-2} & \cdots & u_1 \end{bmatrix}$$

it is easy to see that 3.22 can be rewritten in the form

$$Y = UH$$

Thus in principle one could obtain the impulse response simply inverting the system, namely computing $H = U^{-1}Y$.

Consider however the following definitions:

**Definition 3.3.1** (Ill-posed problem)**.** A problem is said to be ill-posed when:

- the solution is not unique, or
- the solution does not depend continuously on the data.

**Definition 3.3.2** (Ill-conditioned problem)**.** A problem is said to be ill-posed when the solution is much sensitive to small errors in the data.

It is possible to show that the problem of computing $H = U^{-1}Y$ is not only extremely ill conditioned due to the lower triangular structure of $U$, but also suffers from a strong dependence on the data set, namely is ill–posed. Moreover, as a third disadvantage, it does not take into account the eventual dynamical structure of the measurement noise.

To overcome these problems, we use a new Bayesian technique for non-parametric regression. In particular, without imposing any structure on the system (as in the PEM techniques), the impulse response is searched for in an infinite-dimensional space, with some constraints allowing to tune its smoothness. This is obtained looking for the impulse response in a suitable RKHS whose kernel -the so–called Stable Spline kernel, which we review in the next section- imposes smoothness on the functions of the space.

## Identification of LTI systems

We always consider MISO systems, namely, systems in which $m$ inputs are filtered to produce a single output following the rule

$$y_t = \sum_{i=1}^{\infty} f_i u_{t-i} + \sum_{i=0}^{\infty} g_i e_i \tag{3.23}$$

where, for each time instant $t \in \mathbb{Z}$, $y_t \in \mathbb{R}$, $e_t \in \mathbb{R}$ and $u_t \in \mathbb{R}^{1\times m}$, while the coefficients of the impulse responses are such that $f_t \in \mathbb{R}^{1\times m}$ and $g_t \in \mathbb{R}$.

Notice that the system is causal since $f_t = 0$ and $g_t = 0$ for $t < 0$, and in the input–output chain there is always at least one delay step (i.e., $f_0 = 0$). In this model the stochastic process $e_t$ is the Gaussian innovation sequence (namely, $e_t$ is independent from the past of the system up to time $t-1$).

One can easily rewrite the system a form suitable to immediately obtain the one-step ahead predictor as (here $u^k$ is the $k$-th input), namely

$$y_t = \sum_{k=1}^{m} \left[ \sum_{i=1}^{\infty} h_i^k u_{t-i}^k \right] + \sum_{i=1}^{\infty} h_i^{m+1} y_{t-i} + e_t \tag{3.24}$$

in which one can interpret the system as a single output, $y_t$, with $m+1$ inputs, namely the $m$ true inputs and the output sequence up to time $t-1$.

The goal of the algorithm used in this thesis is the reconstruction of the predictor impulse responses $h^k = \{h_t^k\}_{t\geq 0}$.

*Remark* 3.3.3. Formally, $\{y_t\}_{t\geq 0}$ and $\{u_t\}_{t\geq 0}$ are jointly stationary processes related by the model in Eq. 3.23. Here we made a slight abuse of notation and avoided to explicitly distinguish among processes and their realizations.

Under the assumption that the joint spectrum of $\{y_t\}_{t\geq 0}$ and $\{u_t\}_{t\geq 0}$ is bounded away from zero on the unit circle, the predictor impulse responses are BIBO stable. This is taken as a steady assumption from now on.

## Kernel–based identification

Given the set of observed data $\{y_t\}_{t\geq 0}$ and $\{u_t\}_{t\geq 0}$ (now, realization of the corresponding processes), our aim is to reconstruct the $h^k$'s.

The implemented approach consists in the minimization of a regularization functional in a suitable RKHS $\mathcal{H}$ associated with a symmetric positive–definite kernel, as recalled in the previous sections. In particular, we aim to solve (we drop the superscript index $k$ for sake of notation)

$$\hat{h} = \arg\min_{h\in\mathcal{H}} \sum_{t=1}^{N} (y_t - \Gamma_t[h])^2 + \eta ||h||_{\mathcal{H}}^2$$

where $N$ is the number of data samples and, in general, $\{\Gamma_t[h]\}_{t=1,\dots,N}$ are linear and bounded functionals on $\mathcal{H}$. In particular, in our scenario, it holds

$$\Gamma_t[h] = (u * h)(t)$$

which would represent the output at time $t$ with zero innovation and if $h$ were the "true" impulse response of the system.

The already recalled representer theorem allows us to conclude that the solution to the stated problem is a combination of $N$ basis functions defined by the kernel, filtered by the functionals $\{\Gamma_t\}$. As a matter of fact, this implies that the true $h$ can be thought as the realization of an infinite dimensional random vector with zero mean and covariance equal to the kernel, seen as an infinite matrix. From the same perspective, the error on the data $y_t - \Gamma_t[h]$ can be interpreted as a white Gaussian noise independent of $h$, while the solution to the minimization problem represents the minimum variance estimate of $h$ given the data.

**The Stable-Spline kernel** The space of functions $\mathcal{H}$ must satisfy some constraints which are not fully captured by the Gaussian nor Laplacian kernels presented in the previous section.

For sake of simplicity, and without loss of generality, in this paragraph we assume the signals to have domain in $[0, 1] \subset \mathbb{R}$.

A first constraint regards the smoothness of the solution. We restrict to spaces of functions in which the signals and some derivatives are continuous with bounded energy (namely, they belong to a Sobolev space of suitable order).

In the Bayesian interpretation, this type of functions can be recovered by considering the $p$-fold integral of a Gaussian white noise, where $p \geq 1$ is an integer. The corresponding kernel is called Spline kernel, and takes the form

$$W_p(s,t) = \int_0^1 G_p(s,u) G_p(t,u) du$$

with

$$G_p(r,u) = \frac{(r-u)_+^{p-1}}{(p-1)!}$$

where $(x)_+ = \max\{0, x\}$ is the positive part of $x$. Again, in the Bayesian interpretation, the kernel represents the autocorrelation of the signal (in particular, $W_p(s,t)$ increases with $p$, or, from an intuitive point of view, the bigger is $p$, the smoother the signals are).

A particularly important case is the cubic spline kernel ($p = 2$), which leads to

$$W_2(s,t) = \frac{st \min\{s,t\}}{2} - \frac{\min\{s,t\}^3}{6}$$

This kernel is already widely used in literature to treat historic data regressions in several fields (econometrics, biology and so on).

So far we just dealt with smoothness constraints. As we are interested in impulse responses of BIBO stable LTI systems, we also need to require the solution to our minimization problem to decay exponentially to zero. However, the signals in the space defined on the basis of the kernel $W_p(s,t)$ have $h(0) = 0$ and the correlation among $h(t)$ and $h(s)$ increases with the difference among $t$ and $s$. As a drawback, almost any signal in this space diverges.

To overcome this problem, in Pillonetto and De Nicolao (2010) a new type of kernel has been proposed to explicitly handle the problem of exponential stability of the signals in the space. In particular, the *Stable–Spline* kernel is defined as

$$K_p(s,t) = W_p(e^{-\beta s}, e^{-\beta t})$$

and among these kernels, again, particularly important is the case $p = 2$, for which

$$K_2(s,t) = \frac{e^{-\beta(s+t)}e^{-\beta \max(s,t)}}{2} - \frac{e^{-3\beta \max(s,t)}}{6}$$

From Pillonetto and De Nicolao (2010) we known the following proposition, which ensures that the RKHS defined on the bases of $K_2$ is a space of suitable functions for our scopes.

**Proposition 3.3.4.** *Let $h$ be an infinite dimensional Gaussian random vector with zero mean and covariance $K_2$. With probability one, the realizations of $h$ are continuous impulse responses of BIBO stable dynamical systems.*

**Enrichment of the prior** We enrich the previously described prior by modeling the impulse responses $h^k$ as proportional (with unknown scale factors $\lambda_k$) to the convolution of a signal in the space defined on the bases of $K_2$ with a parametric discrete–time impulse response $r$, which is used in order to capture "non–smooth" dynamics, such as high-frequency oscillations. In particular, called $R(z)$ the $z$-transform of such a $r$, we have

$$R(z) = \frac{z^\ell}{P_\theta(z)}, P_\theta(z) = z^\ell + \sum_{i=1}^{\ell} \theta_i z^{\ell-i}$$

which is characterized by a vector of hyperparameters $\theta \in \mathbb{R}^\ell$. The vector $\theta$ belong to a given feasible set $\Theta$ such that the roots of $P_\theta(z)$ belong to the open

left unit semicircle in the complex plane.

We let $K(s,t)$ be the kernel obtained using both $K_2$ and the low–dimensional impulse responses $r$. Overall, this kernel depends on the unknown hyperparameter vector

$$\chi := [\lambda_1, \ldots, \lambda_m, \lambda_{m+1}, \theta_1, \ldots, \theta_\ell, \beta]$$

while the variance of innovation, $\sigma^2$, is estimated from the data as explained in Goodwin et al. (1992).

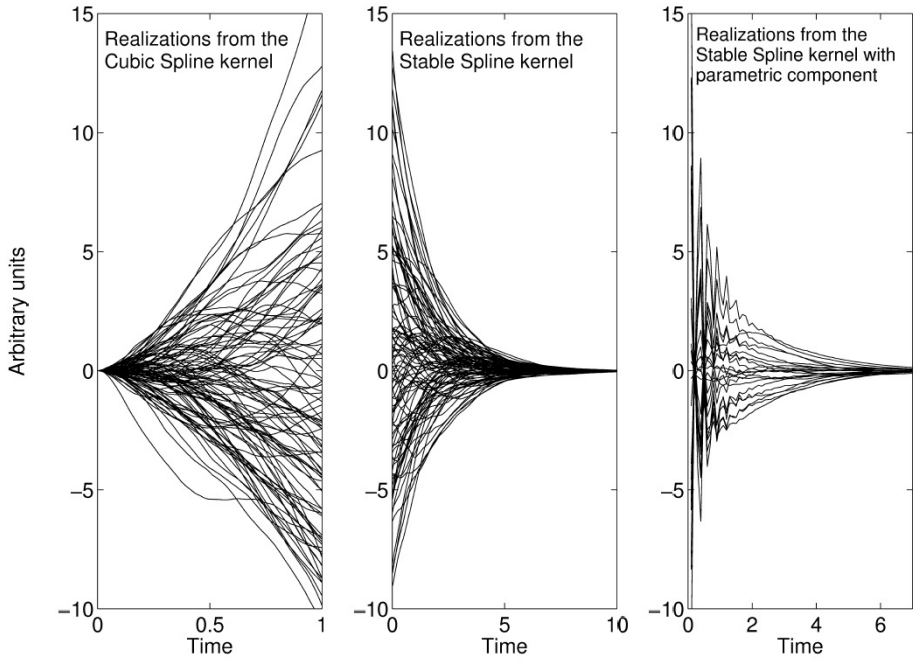


**Figure 3.8:** Realizations of a stochastic process $f$ with autocovariance proportional to the standard Cubic Spline kernel (left), the new Stable Spline kernel (middle) and its sampled version enriched by a parametric component defined by the poles $-0.5 \pm 0.6\sqrt{-1}$ (right).

## The algorithm

The first step to describe the used algorithm is to consider the following vector–form for Eq. 3.24

$$y^+ = \left( \sum_{k=1}^{m} A_k(u^k) h^k \right) + A_{m+1}(y^+, y^-) h^{m+1} + e \tag{3.25}$$

where (the unknown samples of $y^-$ are set to zero in actual implementations)

$$y^+ = \begin{bmatrix} y_1 \\ \vdots \\ y_N \end{bmatrix}, \qquad y^- = \begin{bmatrix} y_0 \\ y_{-1} \\ \vdots \end{bmatrix}, \qquad e = \begin{bmatrix} e_1 \\ \vdots \\ e_N \end{bmatrix}$$

On the basis of such a description of the system, the algorithm exploits the two–steps empirical Bayesian paradigm:

1. the unknown hyperparameter vector $\chi$ is estimated using marginal likelihood optimization in a low–dimensional space,

2. the hyperparameters are set to the just–found estimate, and a minimum variance of the impulse response estimated is computed.

In the next paragraphs we review the two steps. The following approximation is widely used

$$p(y^+, \{h^k\}, y^-|\chi, u) \approx p(y^+|\{h^k\}, y^-, \chi, u)p(\{h^k\}|\chi, u)p(y^-|u) \tag{3.26}$$

which means that $y^-$ is assumed not to carry information on the impulse responses $\{h^k\}$ nor on the hyperparameters $\chi$.

**Estimate of the hyperparameters** $\chi$ The estimate of $\chi$ is obtained by optimizing the marginal likelihood, which is the joint density $p(y^+, \{h^k\}, \chi)$ where $\{h^k\}$ is integrated out. We define

$$V[y^+] = \sigma^2 I_N + \sum_{k=1}^{m+1} \lambda_k A_k K A_k^T$$

were $K$ is seen as an infinite matrix and

$$[A_k]_{ij} = \begin{cases} u^k_{j-i}, k = 1, \dots, m \\ y_{j-i}, k = m+1 \end{cases}$$

Then it holds (Pillonetto et al., 2011)

**Proposition 3.3.5.** *Assume $\{y_t\}_{t\geq 0}$ and $\{u_t\}_{t\geq 0}$ be zero mean, finite variance stationary stochastic processes. Let also hold true the approximation in Eq. 3.26.*

*Then the maximum marginal likelihood estimate of*

$$\chi = [\lambda_1, \dots, \lambda_{m+1}, \theta_1, \dots, \theta_\ell, \beta]$$

*is given by the solution to the problem*

$$\hat{\chi} = \arg\min_{\chi} J(y^+, \chi)$$

*with the constraints* $\theta \in \Theta$, $\beta > 0$ *and* $\lambda_k \geq 0, \forall k = 1, \dots, m, m+1$, *and the cost function is almost surely defined pointwise as*

$$J(y^+, \chi) := \frac{1}{2}\log(\det[2\pi V[y^+]])\frac{1}{2}(y^+)^T(V[y^+])^{-1}y^+$$

**Estimate of the impulse responses $h^k$ given the estimate $\hat{\chi}$** Let $\mathcal{H}_K$ the RKHS defined on the basis of the kernel $K$, which, as already mentioned, takes into account both the structure of $K_2$ and the possible high–frequencies poles of the impulse responses $r$. Denote by $||\cdot||_{\mathcal{H}_K}$ the norm in $\mathcal{H}_K$, and denote also $\hat{h}^k = \mathbb{E}[h^k|y^+, \chi]$, the Bayesian estimate of the impulse responses. The following proposition, again taken from Pillonetto et al. (2011), clarifies the situation.

**Proposition 3.3.6.** *Assume* $\{y_t\}_{t\geq 0}$ *and* $\{u_t\}_{t\geq 0}$ *be zero mean, finite variance stationary stochastic processes. Let also hold true the approximation in Eq. 3.26. Then almost surely*[9]

$$\{\hat{h}\}_{k=1}^{m+1} = \arg\min_{\{h^k \in \mathcal{H}_K\}_{k=1}^{m+1}} \left\{ \left\| y^+ - \sum_{k=1}^{m+1} A_k h^k \right\|^2 + \sigma^2 \sum_{k=1}^{m+1} \frac{||h^k||^2_{\mathcal{H}_K}}{\lambda_k^2} \right\}$$

*In closed form, we have*

$$\hat{h}^k = \lambda_k^2 K A_k^T c$$

*where*

$$c = \left( \sigma^2 I_N + \sum_{k=1}^{m+1} \lambda_k A_k K A_k^T \right)^{-1} y^+$$

**The implemented MatLab function:** `SSpline.m` `SSpline.m` is the MatLab implementation of the algorithm. It takes as inputs both the observed

[9] Here $||\cdot||$ is the Euclidean norm in $\mathbb{R}^N$.

inputs and outputs of the system to be identified, in form of vectors or matrixes. The MatLab function also requires the setting of several other parameters, such as:

- the number $p$ of predictor coefficients to estimate;
- the model type, to be selected between *noise model*

$$A\left(z^{-1}\right) y = B\left(z^{-1}\right) u + e,$$

*outpur error model* (namely $A\left(z^{-1}\right) = 1$)

$$y = B\left(z^{-1}\right) u + e,$$

or *time series* (namely $B\left(z^{-1}\right) = 0$)

$$A\left(z^{-1}\right) y = e;$$

- additional constraints to the hyperparameter vector;
- the number $r$ of input-output data to be used while estimating the hyperparameter vector (this is a key point for computational complexity, as the estimate of the hyperparameters requires the inversion of a $r \times r$ matrix, with complexity $\mathcal{O}(r^3)$);
- (optional) the dimension of the parametric component of the prior, that corresponds to the number of poles introduced in the model.

The outputs of the algorithm are the estimated model and the hyperparameter vector.

# 4

# A prediction system for the Po River and its tributaries

In this chapter we describe the application of the nonparametric identification algorithm proposed in Pillonetto and De Nicolao (2010, 2012); Pillonetto et al. (2011) (and briefly reviewed in the previous chapter) to the case of identification and validation on real data of heights and flows of the Po river and some of its main tributaries.

The application of the algorithm to the observed data has been divided into three steps:

- **data acquisition and data preprocessing**: in the first section we present the raw database which ARPA institution kindly provided. We describe the process of data acquisition and the preprocessing techniques that have been used;
- **training**: in the second section we describe the data set used to train the algorithm, namely to identify the impulse responses of the system;
- **validation**: in the third section we describe the actual implementation of the prediction algorithm for water heights and flows upon various stations

along the Po River and its tributaries. The good performances of the predictor are shown, and some criticalities are discussed. In particular, it is conjectured that the availability of additional data, namely weather forecasts, could significantly improve the prediction performances.

## 4.1 Database characteristics and preprocessing

The database we received from ARPA - AIPo consists of 44 time series correspondent to 11 locations along the Po River main trunk and some of its major tributaries. In particular, as depicted in Figure 4.1, the data acquisition stations of Spessa Po, Piacenza, Cremona, Boretto and Borgoforte are located along the main trunk of the Po river, while Pizzighettone lies upon the Adda River, Borgotaro and San Secondo upon the Taro River, Parma Ponte Verdi upon the Parma River, Marcaria upon the Oglio River and Sorbolo upon the Enza River. Overall, the locations along the main trunk cover about 180 Km through Lombardia and Emilia–Romagna regions.

For each location, indicated with a label $k = 1, \ldots, 11$, the data provided by ARPA are

- the observed heights $\{y_k(t)\}_{t\in\mathcal{I}}$ and the observed flow levels $\{q_k(t)\}_{t\in\mathcal{I}}$, taken each hour (i.e., the integer $t$ indicates hours) in the whole period $\mathcal{I}$ from 00:00, 01 January 2000, up to 24:00, 31 December 2008;

- the ARPA forecasted heights $\{\hat{y}_k(t|t-12)\}_{t\in\mathcal{I}}$ and the forecasted flows $\{\hat{q}_k(t|t-12)\}_{t\in\mathcal{I}}$. The ARPA forecast on time $t$ is done using the available information up to time $t-12$ (namely, the observed data), and the weather forecasts regarding the subsequent 11 hours, up to time $t$.

In total, each time series includes about 71000 data samples.

An important remark is that the river heights are never measured with respect to the stream bed of the river. Instead, the values report the distance of the water-level from the *hydrometric zero quote*, which is an arbitrary altimetric benchmark which zero-level does not refer to any physical quantity. It is interesting to point out that, at any point, the distance between the hydrometric zero and the bed of the river level is not fixed, as the stream

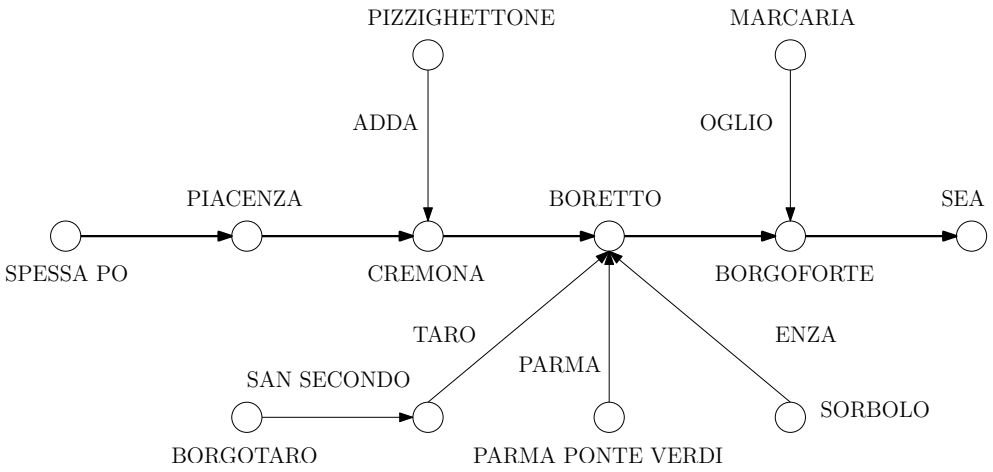


**Figure 4.1:** Locations along the Po river trunk.

bed can evolve due to erosion and sedimentation. On the contrary, flow measurements are not affected by this problem.

Preprocessing of the database was needed due to the presence of spurious data. In particular, for all the 44 time series, we had to deal with:

- **missing data**, that have been linearly interpolated using the closest data at disposal. Namely, assume that $y_k(T), y_k(T+1), \ldots, y_k(T+r)$ are the missing observations of the water height at location $k$, and assume that instead $y_k(T-1)$ and $y_k(T+r+1)$ are at disposal. Then we set

$$y_k(T-1+\alpha) = y_k(T-1) + \frac{\alpha}{r+2}\big(y_k(T+r+1) - y_k(T-1)\big)\ ,$$

  for $\alpha = 0, 1, \ldots, r+2$;

- **outliers**: due to several reasons (e.g., temporary failures of the instruments, random interferences such as passage of boats too close to the sensors) some data subsequences are definitively meaningless. We implemented a simple outlier removal strategy which removes a single data, say $q_k(t)$, the flow level at time $t$ at location $k$, if the increment of $q_k(t)$, call it $dq_k(t)$, exceeds the value

$$|(q_k(t) - q_k(t-1)) - \mathrm{m}(dq_k)| > \kappa\ \mathrm{s}(dq_k)$$

where $\mathrm{m}(dq_k)$ is the mean increment $dq_k(t)$ over the whole dataset, i.e.,

$$\mathrm{m}(dq_k) := \frac{1}{N}\sum_{t=2}^{N}(q_k(t) - q_k(t-1)) \ ,$$

and $\mathrm{s}(dq_k)$ is the standard deviation of the increment $dq_k(t)$, i.e.,

$$\mathrm{s}(dq_k) := \sqrt{\frac{1}{N}\sum_{t=2}^{N}(q_k(t) - q_k(t-1) - \mathrm{m}(dq_k))^2} \ .$$

The threshold $\kappa$ has been set to the value 10 for simplicity. In order to avoid meaningless automatic outlier removals, the outcome of the procedure was to be accepted by the user. In some cases it has been necessary to manually correct the data, since the described procedure was either too mild or too tight.

## 4.2 Training of the algorithm: settings and choice of training sets

As already recalled in Chapter 3, we model the whole river–system as a set of linear time invariant local operators. In particular, we assume that, according to the already presented notation, we can model the height and flow at location $k$ as

$$A_{y,k}(z^{-1})y_k = \sum_{j\in\mathcal{N}_k}(B_{yy,jk}(z^{-1})y_j + B_{qy,jk}(z^{-1})q_j) \tag{4.1}$$

$$A_{q,k}(z^{-1})q_k = \sum_{j\in\mathcal{N}_k}(B_{yq,jk}(z^{-1})y_j + B_{qq,jk}(z^{-1})q_j) \ . \tag{4.2}$$

$\mathcal{N}_k$ is the set of *in–neighbors* of $k$, namely the set of stations assumed to have a relevant influence on $k$. In this thesis we decided to consider the following rule:

*The in–neighbors of a location $k$ are the stations $j$ which are at most two hops upstream with respect to $k$.*

Considering Figure 4.1, the in–neighbors of Cremona are Piacenza, Spessa Po and Pizzighettone, while the in–neighbors of Boretto are, among the others,

Cremona and Piacenza, but not Spessa Po, which is three hops upstream with respect to it.

The choice of considering as possible in–neighbors only the upstream locations lies on obvious physical causality arguments. Moreover, we restricted the influence to the two–hops neighbors only, since we assume that they convey all the important information to predict what happens at a certain location. This choice is also sustained by the empirical observation that, roughly speaking, what happens in $k$ at time $t$ is a delayed version of what already happened at the previous one/two upstream stations, with a delay of at most 10 hours. Since we aim to draw comparisons with the ARPA predictions, we are interested in 12 hours predictions. The definition of in–neighbors as the two–hops upstream stations seems thus enough for our purposes.

## Settings of `SSpline.m`

As already described above, `SSpline.m` accepts as function **inputs** both the output and the inputs of the system we want to identify, plus a set of settings for the algorithm. In this paragraph we briefly describe the choice for these parameters:

- number $p$ of coefficients of the predictor to estimate: as recalled, we are interested in 12-steps ahead predictions. We always set $p = 50$, which means that we assume that a quantity at time $t$ is influenced by its inputs up to time $t - 50$. In other words, $A_{y,k}$ and $B_{yy,jk}, \ldots, B_{qq,jk}$ are polynomials in $z^{-1}$ of degree 50. The comparison with trains and tests on smaller datasets than those presented in the following sections showed that smaller values of $p$ yield to worse results. In principle, one could train and test the algorithm with increasing values of $p$, and optimize over a suitably defined cost which takes into account both the performances in terms of fitting of the data, and the computational load and time. This is left for future design of a more complex identification system;

- the model type: the stations can be divided into two large groups, namely *upstream* stations and *non upstream* stations. In the first group we find Spessa Po, Pizzighettone, Borgotaro, Parma Ponte Verdi, Marcaria and Sorbolo. The main characteristic of these stations is that we have no information on their inputs, thus we model their river heights and flow

levels as time series. In other terms, the model type for these stations is set to `'yn'`, and the algorithm will produce a model, for example for the heights at station $k$, of the type

$$A_{y,k}(z^{-1})y_k = e \ .$$

For the second group of stations, instead, we know what happened at the previous stations along Po and tributaries. We thus interpret this information as an additional input to the system. Thus, the model type for these stations is set to `'yy'`, and the produced model will be of the type in Eq. (4.1);

- additional constraints on the hyperparameter vector: for these constraints we chose standard low computational load settings. Comparing trains and tests suggests that less performing settings do not yield to substantial improvements;

- the number $r$ of input-output data to be used while estimating the hyperparameter vector: this is set to one fifth of the amount of data samples, looking for a trade–off among computational load and accuracy;

- the dimension of the parametric component of the prior: this was set to zero, namely the set of possible impulse responses is not enriched with high frequency components. The reason behind this choice is that the river system appears to be a relatively slow/low pass system.

## Actual training

Once the database had been processed, it was immediately recognized that the data we had at disposal could hardly be seen as inputs and outputs to linear systems. In fact, the height measurements oscillate around the hydrometric zero. Since the hydrometric zeros have no physical meanings, they show fictitious forcing terms. To give an example of this fact, consider Figure 4.2, in which we depicted the measured heights at Cremona and Boretto, two subsequent stations along the main trunk of Po river, in the period 00:00, 17 February 2005 – 16:00, 30 March 2005. As one can see, it seems that a fictitious offset among the heights of the two stations is present. As a second observation, heights are not always strictly positive quantities, as one could expect.

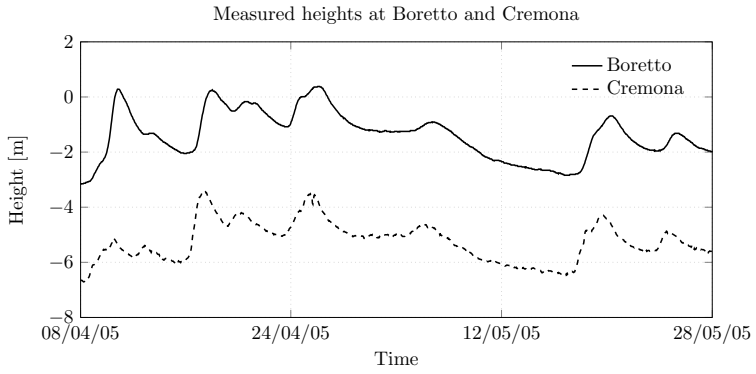


**Figure 4.2:** Offset example.

In order to overcome these problems, we decided to consider three groups of datasets:

- the raw data: in this group of datasets we maintain the original data, without any correction of the observed offsets;
- zero mean data: in this second group of datasets we subtract the mean to all the time series at disposal. This allows to avoid the offsets, and makes the time series more resemblant to inputs and outputs of linear time invariant systems;
- non negative data: in this third group of datasets we subtract the minimum value to all the time series at disposal. This imposes some sort of fictitious positiveness of the signals we deal with.

For each group of datasets, five different training sets have been used:

- 16:00, 16 April 2005 $\mapsto$ 08:00, 8 May 2005 (1000 data samples);
- 16:00, 16 April 2005 $\mapsto$ 24:00, 22 June 2005, (2000 data samples);
- 16:00, 16 April 2005 $\mapsto$ 16:00, 2 August 2005, (3000 data samples);
- 16:00, 16 April 2005 $\mapsto$ 08:00, 13 September 2005, (4000 data samples);
- 16:00, 16 April 2005 $\mapsto$ 24:00, 25 October 2005, (5000 data samples).

The fifth dataset, which is the longest, covers a period which lasts from late spring to early autumn. It thus reflects different weather and river scenarios, such as high levels of water and flow values due to spring rains, low levels in

summer, and again higher levels in autumn. We decided to run several trains covering an enlarging period in order to appreciate whether letting the algorithm learn from larger sets of data could yield to performances improvements. As we will show in the next section, this was indeed the case.

## 4.3 Test of the algorithm: implementation and results

The test of the algorithm consisted in the implementation of a set of functions capable to use the identified models in order to compute, for each station, the forecasts $\hat{y}_k(t|t-12)$ and $\hat{q}_k(t|t-12)$. Namely, we aimed to predict the river height and flow value at time $t$ given all the possible information up to time $t-12$. This is done in order to draw a comparison between the performances of our nonparametric approach and the ARPA prediction system.

We chose as set of samples for validation the period 16:00, 20 April 2007 – 24:00, 12 July 2007, corresponding to 2000 samples of the dataset. This choice for the test set is motivated by the fact that the time distance among the training set and the validation set must be large enough to assume that the samples in these two sets are statistically independent.

### Forecast for upstream stations

As already recalled, the model identified by the algorithm for heights and flow values of upstream stations has no input, namely it is of the type

$$A_{y,k}(z^{-1})y_k = e_{y,k}$$
$$A_{q,k}(z^{-1})q_k = e_{y,k}.$$

The one–step ahead predictors $\hat{y}_k(t+1|t)$ and $\hat{q}_k(t+1|t)$ of these stations is thus simply given by a linear combination of the heights, or the flows, at times $t-1,\ldots,t-p$. The 12–steps ahead predictions can be thus easily computed on the basis of them.

## Forecast for non upstream stations

The 12–steps ahead prediction of non upstream stations is slightly more involved. Assume, e.g., that we want to compute the 12–steps ahead predictions $\hat{y}_k(t+12|t)$ for the Piacenza station. The model identified by `SSpline.m` is of the type

$$A_P(z^{-1})y_P(t) = B_{y,SP}(z^{-1})y_{SP}(t) + B_{q,SP}(z^{-1})q_{SP}(t) + e_P(t) \ ,$$

where $y_P(t)$ is the water height at Piacenza at time $t$, $y_{SP}(t)$ and $q_{SP}(t)$ are respectively the height of the river and the flow at Spessa Po at time $t$, and $e_P(t)$ is Gaussian innovation. Namely, the height of the river at Piacenza is the output of the model in which the inputs are the past samples of the height at Piacenza, and heights and flows at Spessa Po. In particular, we can rewrite the previous equation for time $t + 12$ as

$$\begin{aligned} y_P(t+12) = &\sum_{i=1}^{p} B_{y,SP,i}y_{SP}(t+12-i) + \sum_{i=1}^{p} B_{q,SP,i}q_{SP}(t+12-i) \\ &+ \sum_{i=1}^{p} A_{P,i}y_P(t+12-i) + e_P(t+12) \end{aligned}$$

which in principle yields

$$\begin{aligned} \hat{y}_P(t+12|t) = &\sum_{i=1}^{p} B_{y,SP,i}y_{SP}(t+12-i) + \sum_{i=1}^{p} B_{q,SP,i}q_{SP}(t+12-i) \\ &+ \sum_{i=1}^{p} A_{P,i}y_P(t+12-i) \ . \end{aligned} \tag{4.3}$$

In this equation we see that the predictor $\hat{y}_P(t+12|t)$ would also require the knowledge of the inputs $y_{SP}(t+12-i)$ and $y_{SP}(t+12-i)$ in the interval $[t+1, \ldots, t+11]$, but this is impossible, since these data belong to the future with respect to time $t$. In order to overcome this problem, we substitute for these "actual" inputs their forecasts, under the assumptions that they have already been computed by the station in Spessa Po, as depicted in Figure 4.3.

In other terms, Equation (4.3) turns into

$$\begin{aligned}\hat{y}_P(t+12|t) = &\sum_{i=1}^{11} B_{y,SP,i}\hat{y}_{SP}(t+12-i|t) + \sum_{i=12}^{p} B_{y,SP,i}y_{SP}(t+12-i) \\ &+ \sum_{i=1}^{11} B_{q,SP,i}\hat{q}_{SP}(t+12-i|t) + \sum_{i=12}^{p} B_{q,SP,i}q_{SP}(t+12-i) \\ &+ \sum_{i=1}^{11} A_{P,i}\hat{y}_P(t+12-i|t) + \sum_{i=12}^{p} A_{P,i}y_P(t+12-i|t) \end{aligned} \tag{4.4}$$

and in general

$$\begin{aligned}\hat{y}_P(t+r|t) = &\sum_{i=1}^{r-1} B_{y,SP,i}\hat{y}_{SP}(t+r-i|t) + \sum_{i=r}^{p} B_{y,SP,i}y_{SP}(t+r-i) \\ &+ \sum_{i=1}^{r-1} B_{q,SP,i}\hat{q}_{SP}(t+r-i|t) + \sum_{i=r}^{p} B_{q,SP,i}q_{SP}(t+r-i) \\ &+ \sum_{i=1}^{r-1} A_{P,i}\hat{y}_P(t+r-i|t) + \sum_{i=r}^{p} A_{P,i}y_P(t+r-i|t) \end{aligned} \tag{4.5}$$

which for $r = 12$ gives the previous equation.

Analogously to what has been shown for Piacenza and Spessa Po, for any other non upstream station, the prediction is computed using the actual measured data if available, and the forecasts computed by the in–neighbors if not. Notice that it is thus necessary that the in–neighbors store in memory, for each $t$, the entire sequences $[\hat{y}_k(t+1|t), \ldots, \hat{y}_k(t+11|t), \hat{y}_k(t+12|t)]$ and $[\hat{q}_k(t+1|t), \ldots, \hat{q}_k(t+11|t), \hat{q}_k(t+12|t)]$, which can be computed using the analogous to Equation (4.5). For example, Piacenza is in–neighbor of Cremona. The forecast in Cremona will thus require the whole $[\hat{y}_P(t+1|t), \ldots, \hat{y}_P(t+11|t), \hat{y}_P(t+12|t)]$ and $[\hat{q}_P(t+1|t), \ldots, \hat{q}_P(t+11|t), \hat{q}_P(t+12|t)]$.

Notice moreover that this procedure requires a certain ordering of the stations in terms of forecasts computation, since non upstream locations need their in–neighbors' information in order to process their data. This is made possible by the assumption that the in–neighbors of a station are upstream with respect to that location, since then an iterative algorithm can be implemented. Assume in fact we are at time $t$ and we need to compute the prediction at time $t+12$. Then, as it can be seen in Figure 4.1,

- step 1: the stations Spessa Po, Pizzighettone, Borgotaro, Parma Ponte

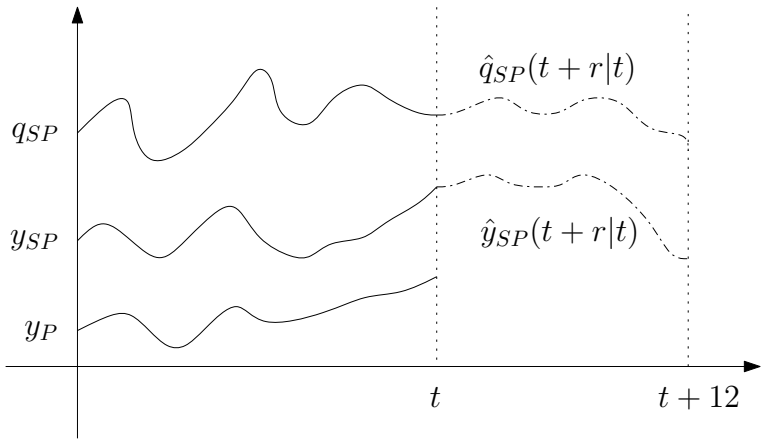


**Figure 4.3:** Pictorial description of the prediction algorithm for non upstream stations. If it is at disposal the information up to time $t$ and the goal is to compute the forecast $\hat{y}_P(t+12|t)$ of the height at Piacenza, the inputs up to $t+11$ are in principle needed. Since the future $[t+1,\ldots,t+11]$ is however unseen, the predictor uses the actual data up to time $t$ (solid line) and the forecasts of the heights and flows at Spessa Po in the unseen future (dashed line). This allows to compute iteratively $\hat{y}_P(t+1|t), \hat{y}_P(t+2|t), \ldots, \hat{y}_P(t+12|t)$. Notice that this sequence must be stored since it will be used when forecasting at Cremona, of which Piacenza is an in–neighbor.

Verdi, Sorbolo and Marcaria, which are all upstream, do not require any information from other locations, thus they can forecast their heights and flows;

- step 2: all the in–neighbors of stations Piacenza and San Secondo have now computed their forecasts. These data are sent to Piacenza and San Secondo, which can make their own forecasts;
- step 3: Cremona is able to compute its forecasts;
- step 4: Boretto is able to compute its forecasts;
- step 5: Borgoforte is able to compute its forecasts.

After the fifth step, all the stations have computed their forecasts, and they can wait for time $t+1$ and the new measurements.

## Test results

The identified models, for each of the 15 obtained databases, have been validated for time constraints reasons on four stations, namely Spessa Po, Pizzighettone,

Piacenza and Cremona. These four locations constitute the most upstream part of the main trunk of the Po river, with the tributary Adda.

The overall computation time required by the algorithm to compute the 12–steps ahead predictions, for all the time instants in the test set, for all the 15 databases, for the four considered stations, has been around 4 hours using the Computation Cluster BLADE at the Department of Information Engineering, University of Padova. The algorithm has been implemented using the programming environment MatLab by MathWorks. The computation time required to obtain the 12–steps ahead predictions for the four stations for a specific time is thus around 1 second. Of course, a real implementation of the algorithm will need to take into account all the stations along the Po river trunk and its tributaries, thus increasing the computation time to obtain the 12–steps ahead prediction up to several minutes. However, we expect a major performances improvement on C implementation and after optimization of the code.

## Results in the upstream locations

In this first paragraph we present some results obtained on the two upstream locations considered, namely Pizzighettone (on the Adda river) and Spessa Po (on the main trunk of the Po river).

In Figure 4.4 and Figure 4.5 we draw a comparison among the predictions using the model identified using nonparametric techniques (in dotted line) and the predictions obtained by ARPA (in dashed line). The periods covered in the two figures are respectively 16:00, 30/04/07 – 16:00, 10/05/07 and 16:00, 25/05/07 – 16:00, 25/05/07, and the training of the algorithm has been performed on 5000 data samples on the dataset in which the mean has been removed. As we show later on, this is arguably the type of dataset which provides the best performance among all our tries.

Analysis of the results show that we can roughly distinct two different conditions which affect the performances of the nonparametric algorithm:

- “stationary regime”: we say that a station is in a stationary regime when heights and flows slowly change in time. For example, in Figure 4.4 the station is in this regime during the first 50 and the last 80 samples. In this situation, the autoregressive component of the model allows the predictor to oscillate around the actually observed (12 steps later) value of height

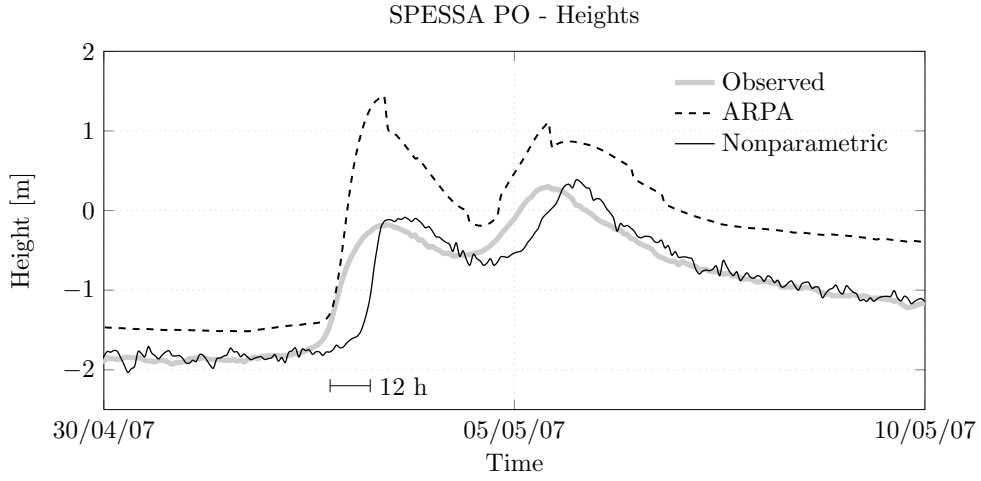


**Figure 4.4:** Prediction of the height at Spessa Po in the period 16:00, 30/04/07 – 16:00, 10/05/07. The nonparametric model is obtained training the algorithm over 5000 samples of the database in which we preprocessed the data such as their mean is zero. Notice two operative regimes, called "stationary regime" and "non–stationary regime". Due to absence of information on rainfall and on upstream stations, in the non–stationary regime the forecasted values show a certain delay (of about 12 hours) with respect the observed data.

(for the flows we obtain an analogous phenomenon).

- "non–stationary regime": we say that a station is not in a stationary regime when heights and flow are subjected to fast changes in time due to the fact that rainfall or upstream floods rapidly increase the quantity of water at the station. Since upstream locations receive no inputs, i.e., have no possibility to know what is happening upstream, the predictor has no way to correctly forecast such increasings/decreasings in heights and flows.

Analogous observations can be done analyzing Figure 4.6 and Figure 4.7. In the former, we compare observed flows and 12–steps ahead predictions at Spessa Po in the period $16:00, 19/06/07 - -16:00, 29/06/07$, while in the latter we compare observed heights and 12–steps ahead predictions at Pizzighettone in the period $16:00, 24/05/07 - -16:00, 15/06/07$. As Spessa Po, Pizzighettone is an upstream station, and in fact the predictor exhibits a certain delay with respect to the observed heights.

Comparison with ARPA predictions shows that in general the nonparametric model allows predictions which are closer to the actual data. However, ARPA system shows much better performances concerning the ability to correctly forecast flood peaks. This is clearly a very important feature since it allows to

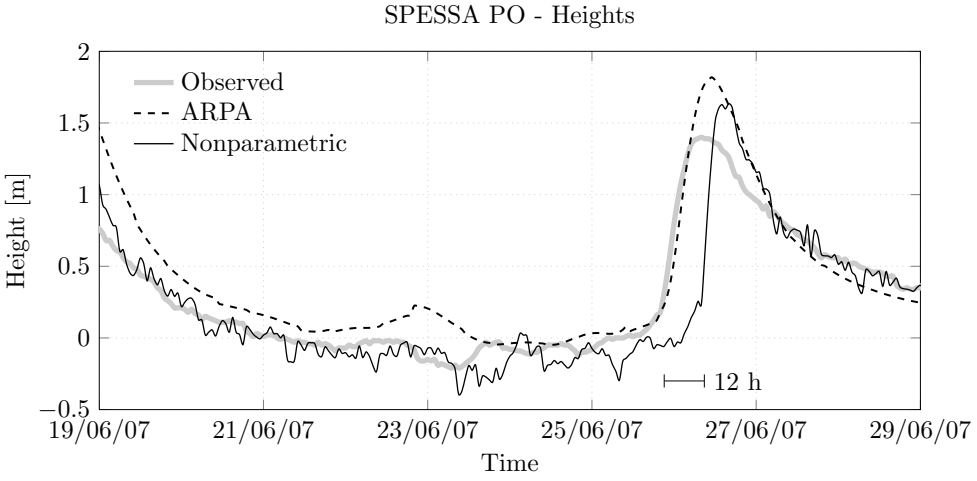


**Figure 4.5:** Prediction of the height at Spessa Po in the period 16:00, 19/06/07 – 16:00, 29/06/07. The nonparametric model is obtained training the algorithm over 5000 samples of the database in which we preprocessed the data such as their mean is zero. We notice again that in stationary regime the performances of the nonparametric model are good, while it is unable to follow fast changes increasing or decreasing of the quantity of interest, in this case the height of the river.

exactly inform authorities about flood risks, thus making the forecast system valuable.

## Results in non upstream locations

This this second paragraph we discuss the result in the two non upstream stations for which forecasts have been computed, namely Piacenza and Cremona.

Figure 4.8 depicts a comparison among ARPA prediction and nonparametric prediction for heights at Piacenza in the period 16:00, 09/06/07 – 16:00, 09/07/07. We use again our best identified model, obtained using 5000 samples for training and the database in which the signals have zero mean.

In case of a non upstream location, in addition to the autoregressive part we have a set of inputs which contribute to heights and flows at the station. In our particular case, inputs for Piacenza are heights and flows at Spessa Po, while inputs for Creamona are heights and flows at Piacenza, Spessa Po e Pizzighettone.

Due to this characteristic of non upstream stations, we expected an improvement in the ability of the nonparametric model to correctly forecast. In fact, one can easily see from the Figure 4.8 that using heights and flows from Spessa Po helps to correctly forecast that a flood will take place. In other terms, the

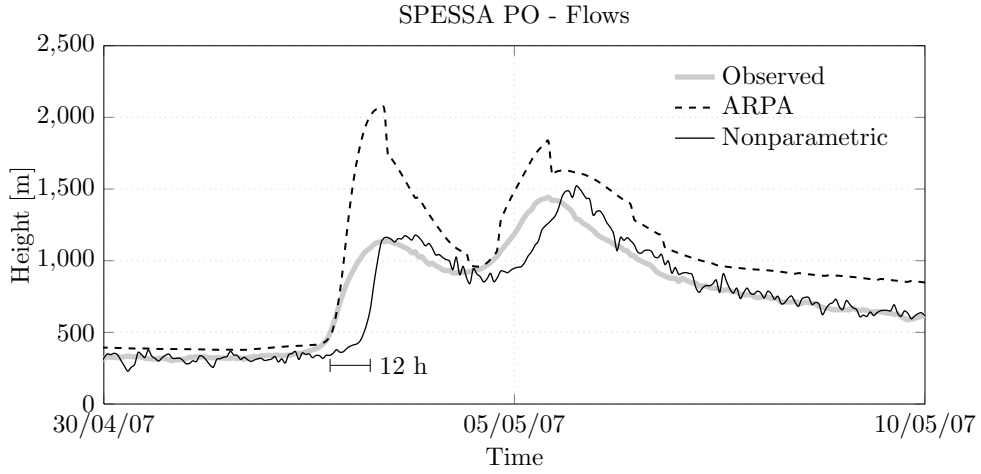


**Figure 4.6:** Prediction of the flow at Spessa Po in the period 16:00, 19/06/07 – 16:00, 29/06/07. The nonparametric model is obtained training the algorithm over 5000 samples of the database in which we preprocessed the data such as their mean is zero.

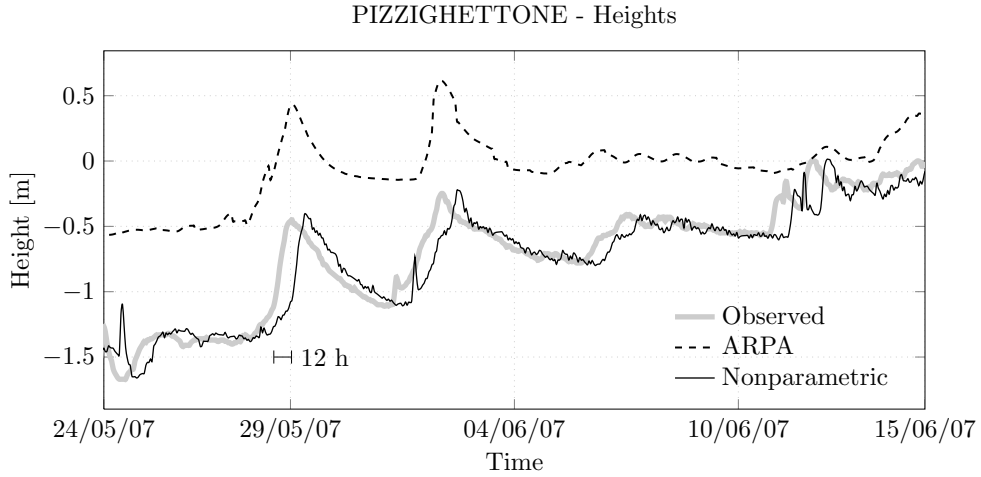


**Figure 4.7:** Prediction of the flow at Pizzighettone in the period 16:00, 24/05/07 – 16:00, 15/06/07. The nonparametric model is obtained training the algorithm over 5000 samples of the database in which we preprocessed the data such as their mean is zero.

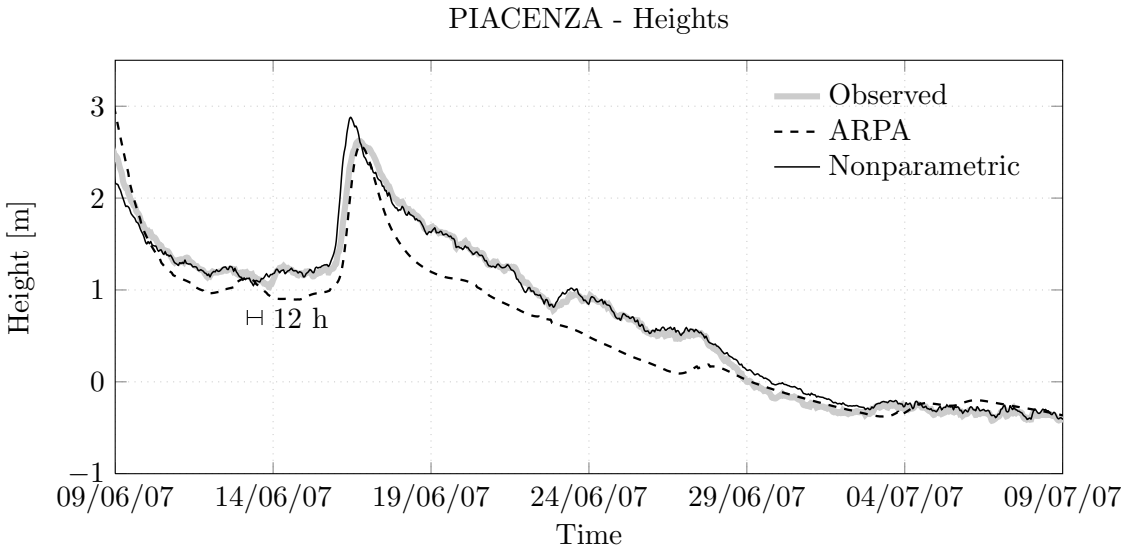


**Figure 4.8:** Prediction of the height at Piacenza in the period 16:00, 09/06/07 – 16:00, 09/07/07. The nonparametric model is obtained training the algorithm over 5000 samples of the database in which we preprocessed the data such as their mean is zero. In case of non upstream stations, we do not observe a clear distinction among stationary and non–stationary regimes. However, in the regime in which heights are subjected to fast increasings and decreasing, forecast are a bit in advance with respect to the actual measured quantity.

predictor does not have to wait 12 steps in order to receive the information "the water level increased/decreased", as it happens in upstream stations. Instead, since the level of the river increased at Spessa Po, a corresponding increasing is expected and forecasted also at Piacenza.

One can also notice that input information is somehow misused by the predictor, yielding forecasts which are a bit in advance with respect to the observed heights. This is probably due to the fact that identification of the impulse response is not ideal, and also because the Po river is far from being a time–invariant system.

Notice that in case of Piacenza ARPA predictions are very accurate. In particular, we notice that peaks are perfectly forecasted.

For completeness we also discuss Figure 4.9 and Figure 4.10, which compare ARPA and nonparametric forecasts for heights and flows, respectively, at Cremona, both in the period 16:00, 04/06/07 – 16:00, 04/07/07. In both cases, the nonparametric forecast is computed according to the model obtained using 5000 samples and the zero mean database.

In this case, Cremona receives information from many stations, and forecasts show good performance.

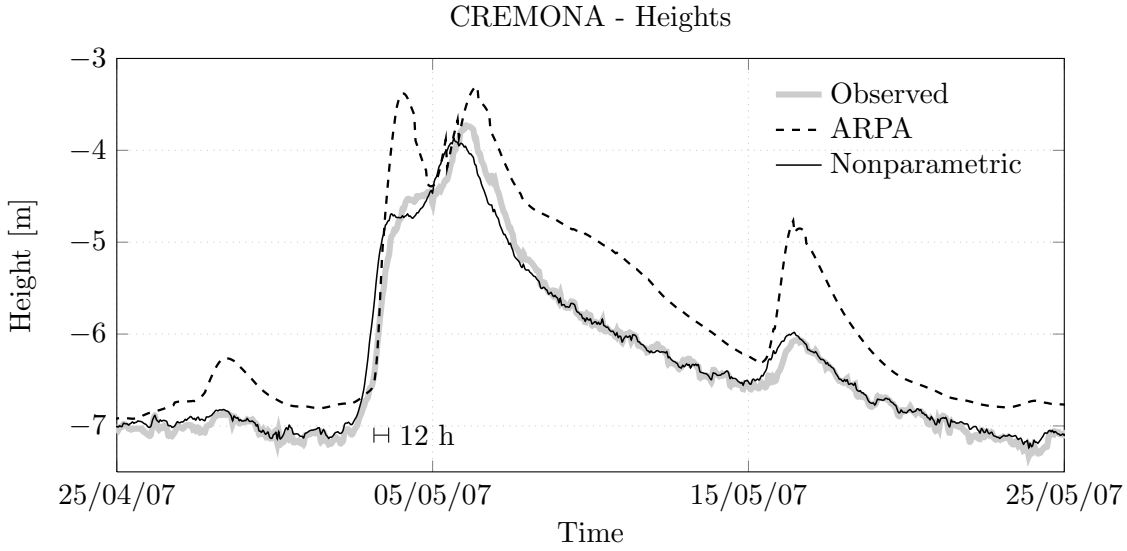


**Figure 4.9:** Prediction of the height at Cremona in the period 16:00,16:00, 04/06/07 – 16:00, 04/07/07. The nonparametric model is obtained training the algorithm over 5000 samples of the database in which we preprocessed the data such as their mean is zero. We can easily appreciate also in this case the improvement with respect to Spessa Po and Pizzighettone.

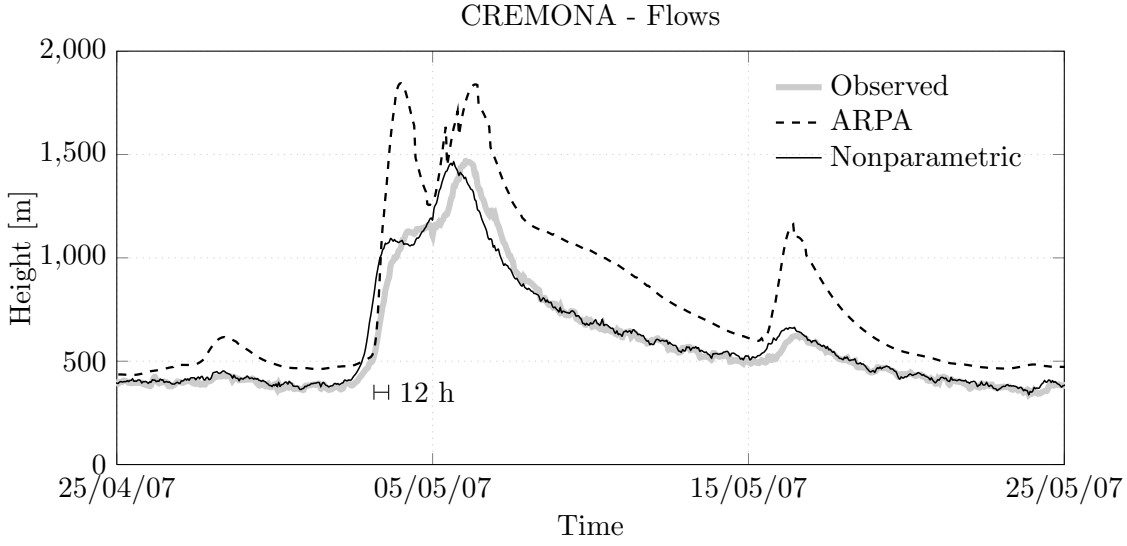


**Figure 4.10:** Prediction of the flow at Cremona in the period 16:00,16:00, 04/06/07 – 16:00, 04/07/07. The nonparametric model is obtained training the algorithm over 5000 samples of the database in which we preprocessed the data such as their mean is zero.

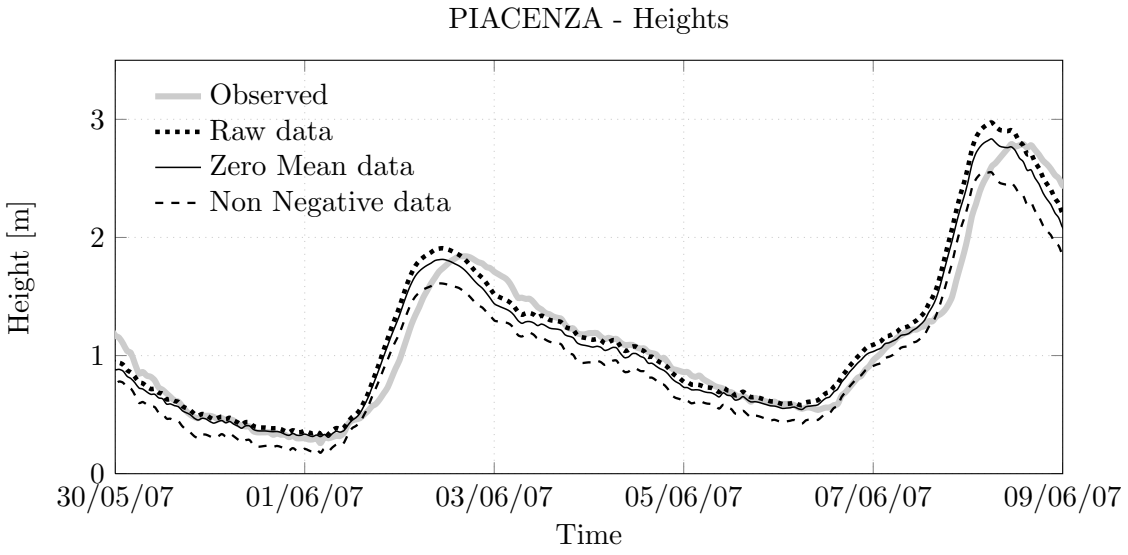


**Figure 4.11:** Prediction of the height at Piacenza in the period 16:00, 29/05/07 – 16:00, 08/06/07. A comparison among forecasts using raw data, zero mean data and non negative data is shown, with train over 4000 samples in the three cases. In this case the forecasts are very similar one each other.

## Comparison among the databases

In this section we briefly compare forecasts obtained using different instances of the 15 databases obtained after the preprocessing.

In the previous sections we showed forecasts computed according to the model identified using 5000 samples and the database in which the signals have zero mean.

In general, models obtained using non negative data show performances comparable with those obtained with zero mean signals. Raw data are instead more subject to the fact that heights are measured with respect to different hydrometric zero quotes, thus showing fictitious offsets.

We only give a couple of examples. In Figure 4.11 we compare forecasts computed using three models, each based on a training set of 4000 samples, with Raw data, zero mean data and non negative data. The problem is prediction of height at Piacenza in the period 16:00, 29/05/07 – 16:00, 08/06/07. This is a lucky case, in which the three models behave approximatively in the same manner, namely, they provide very similar forecasts. We can also notice that in all the three cases the forecast is in advance with respect to the measured heights.

Another example is depicted in Figure 4.12 in which we compare heights

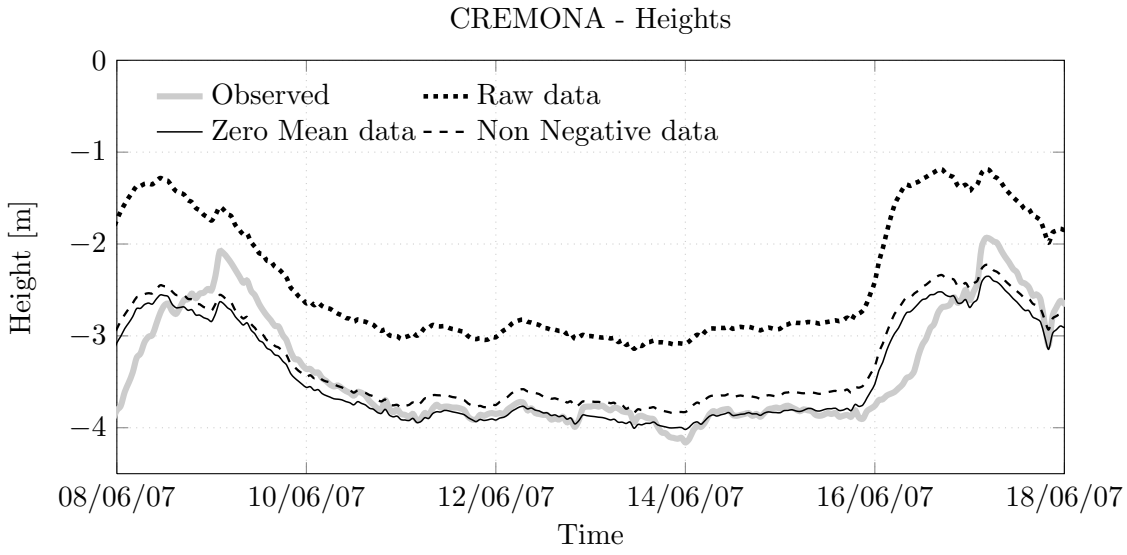


**Figure 4.12:** Prediction of the heigth at Cremona in the period 16:00, 17/07/07 – 16:00, 27/06/07. A comparison among forecasts using raw data, zero mean data and non negative data is shown, with train over 5000 samples in the three cases. In this case forecasts using zero mean signals and non negative signals are similar one each other and show good accordance with the measured heights, while forecasts using raw data show a not–compensated offset.

forecasts at Cremona using three models, each based on a training set of 5000 samples, with raw data, zero mean data and non negative data. One can see that, while forecasts computed using zero mean signals and non negative signals are in good accordance with measured data and similar one each other, the forecasts computed using raw data show a not–compensated offset with measured data. This is probably due to the structure of the identified impulse response using raw data. It is worth noticing, however, that this is not the typical behavior of models obtained from raw data. For example, in the same scenario, when training the dataset using 4000 samples the offset disappears.

## 4.4 Mean–square error

In this section we measure the performances of the models obtained from the 15 databases using the mean–square error as a performance indicator. In particular, if $\{y_t\}_{t\in\mathcal{I}}$ and $\{\hat{y}_{t|t-12}\}_{t\in\mathcal{I}}$ are measured heights and forecasted heights, for a certain station and according to a certain model, the Mean Square

| | 1000 | 2000 | 3000 | 4000 | 5000 | ARPA |
|---|---|---|---|---|---|---|
| SPESSA PO | 0.0612 | 0.0621 | 0.0640 | 0.0620 | 0.0621 | 0.3188 |
| PIZZIGHETTONE | 0.0265 | 0.0260 | 0.0257 | 0.0261 | 0.0260 | 0.9063 |
| PIACENZA | 0.0231 | 0.3222 | 5.0642 | 0.0205 | 6.4899 | 0.1442 |
| CREMONA | 1.6885 | 1.7758 | 2.8075 | 1.4055 | 1.7975 | 0.3423 |

**Table 4.1:** Mean–square errors for heights forecasts of nonparametric models and ARPA model. Rows are indexed by the various stations for which forecasts have been computed, columns are indexed by the numerosity of the dataset used for training. The table refers to models obtained using raw data.

Error (MSE) is defined as

$$\text{MSE} := \frac{1}{|\mathcal{I}|} \sum_{t \in \mathcal{I}} (y_t - \hat{y}_{t|t-12})^2 \ ,$$

where $\mathcal{I}$ is the whole validation period of 2000 samples. Of course, analogous definitions hold for flows and flows forecasts.

The results are shown in Tables 4.1, 4.2 and 4.3 for heights and heights forecasts, and in Tables 4.4, 4.5 and 4.6 for flows and flows forecasts. The following figures graphically depict these tables.

We can notice that

- usually models obtained from raw data behave worse than those obtained from zero mean data and non negative data. This might be due to the offsets in raw data, which are not present, or at least whose influence is much lower, in case of non negative data and, even more, in case of zero mean data;

- even if not as much as expected, there is a slight improvement using larger training sets. We can appreciate this improvement in particular for models obtained using zero mean data;

- nonparametric models usually perform better than ARPA model, at least using large enough databases for training.

We also notice the presence of two outliers in Table 4.1 concerning Piacenza station, for which we have no clear explanation. We expect to have a better understanding of this issue in future analysis.

| | 1000 | 2000 | 3000 | 4000 | 5000 | ARPA |
|---|---|---|---|---|---|---|
| SPESSA PO | 0.0470 | 0.0459 | 0.0460 | 0.0443 | 0.0443 | 0.3188 |
| PIZZIGHETTONE | 0.0180 | 0.0169 | 0.0166 | 0.0170 | 0.0169 | 0.9063 |
| PIACENZA | 0.0283 | 0.0255 | 0.0250 | 0.0245 | 0.0239 | 0.1442 |
| CREMONA | 0.6346 | 0.0861 | 0.0776 | 0.0845 | 0.0756 | 0.3423 |

**Table 4.2:** Mean–square errors for heights forecasts of nonparametric models and ARPA model. Rows are indexed by the various stations for which forecasts have been computed, columns are indexed by the numerosity of the dataset used for training. The table refers to models obtained using zero mean data.

| | 1000 | 2000 | 3000 | 4000 | 5000 | ARPA |
|---|---|---|---|---|---|---|
| SPESSA PO | 0.0443 | 0.0450 | 0.0467 | 0.0449 | 0.0450 | 0.3188 |
| PIZZIGHETTONE | 0.0171 | 0.0165 | 0.0163 | 0.0167 | 0.0166 | 0.9063 |
| PIACENZA | 0.0271 | 0.0155 | 0.0185 | 0.0274 | 0.0143 | 0.1442 |
| CREMONA | 0.5237 | 0.1146 | 0.0907 | 0.0814 | 0.0792 | 0.3423 |

**Table 4.3:** Mean–square errors for heights forecasts of nonparametric models and ARPA model. Rows are indexed by the various stations for which forecasts have been computed, columns are indexed by the numerosity of the dataset used for training. The table refers to models obtained using non negative data.

| | 1000 | 2000 | 3000 | 4000 | 5000 | ARPA |
|---|---|---|---|---|---|---|
| SPESSA PO | 1.6689 | 1.6595 | 1.6865 | 1.5986 | 1.6072 | 5.8165 |
| PIZZIGHETTONE | 0.1226 | 0.1200 | 0.1198 | 0.1354 | 0.1260 | 0.7088 |
| PIACENZA | 1.5370 | 1.1289 | 1.0232 | 1.0359 | 0.9748 | 7.4394 |
| CREMONA | 2.3688 | 1.5308 | 1.4474 | 1.5012 | 1.6361 | 8.9115 |

**Table 4.4:** Mean–square errors for flows forecasts of nonparametric models and ARPA model. Rows are indexed by the various stations for which forecasts have been computed, columns are indexed by the numerosity of the dataset used for training. The table refers to models obtained using raw data. A factor $10^4$ is omitted in the table.

| | 1000 | 2000 | 3000 | 4000 | 5000 | ARPA |
|---|---|---|---|---|---|---|
| SPESSA PO | 1.7432 | 1.6612 | 1.6300 | 1.5549 | 1.5664 | 5.8165 |
| PIZZIGHETTONE | 0.1333 | 0.1255 | 0.1251 | 0.1397 | 0.1323 | 0.7088 |
| PIACENZA | 1.2158 | 1.4768 | 1.0624 | 1.0657 | 1.0256 | 7.4394 |
| CREMONA | 2.0742 | 1.7072 | 1.5551 | 1.4973 | 1.4446 | 8.9115 |

**Table 4.5:** Mean–square errors for flows forecasts of nonparametric models and ARPA model. Rows are indexed by the various stations for which forecasts have been computed, columns are indexed by the numerosity of the dataset used for training. The table refers to models obtained using zero mean data. A factor $10^4$ is omitted in the table.

| | 1000 | 2000 | 3000 | 4000 | 5000 | ARPA |
|---|---|---|---|---|---|---|
| SPESSA PO | 1.6689 | 1.6595 | 1.6865 | 1.5986 | 1.6072 | 5.8165 |
| PIZZIGHETTONE | 0.1226 | 0.1200 | 0.1198 | 0.1354 | 0.1260 | 0.7088 |
| PIACENZA | 1.4698 | 1.0394 | 0.9477 | 0.9872 | 0.9646 | 7.4394 |
| CREMONA | 2.4642 | 1.5208 | 1.4567 | 1.5319 | 1.4603 | 8.9115 |

**Table 4.6:** Mean–square errors for flows forecasts of nonparametric models and ARPA model. Rows are indexed by the various stations for which forecasts have been computed, columns are indexed by the numerosity of the dataset used for training. The table refers to models obtained using non negative data. A factor $10^4$ is omitted in the table.

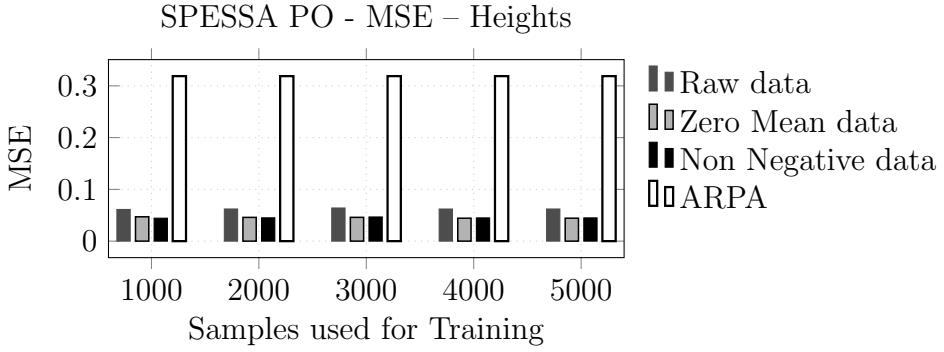


**Figure 4.13:** Mean–square errors for heights forecasts of nonparametric models and ARPA model at Spessa Po. The histogram describes the change in MSE when the dataset used for training grows from 1000 to 5000 samples. It is also shown, for comparison, the MSE of ARPA forecasts.

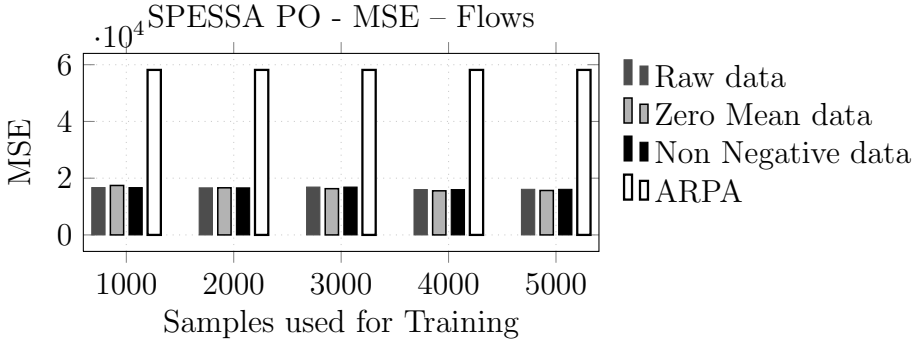


**Figure 4.14:** Mean–square errors for flows forecasts of nonparametric models and ARPA model at Spessa Po. The histogram describes the change in MSE when the dataset used for training grows from 1000 to 5000 samples. It is also shown, for comparison, the MSE of ARPA forecasts.

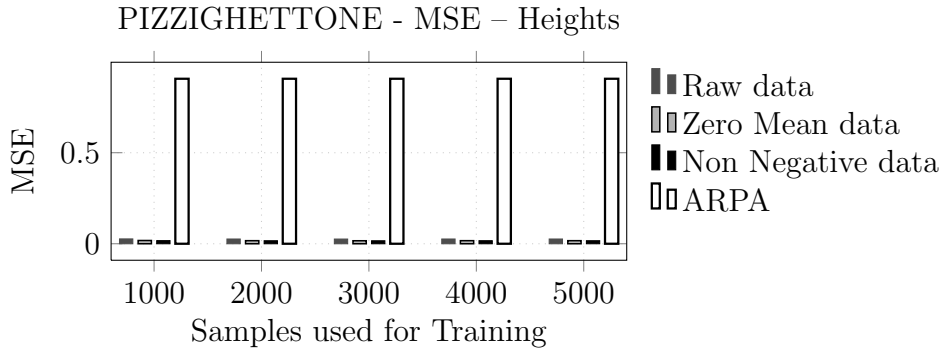


**Figure 4.15:** Mean–square errors for heights forecasts of nonparametric models and ARPA model at Pizzighettone. The histogram describes the change in MSE when the dataset used for training grows from 1000 to 5000 samples. It is also shown, for comparison, the MSE of ARPA forecasts.

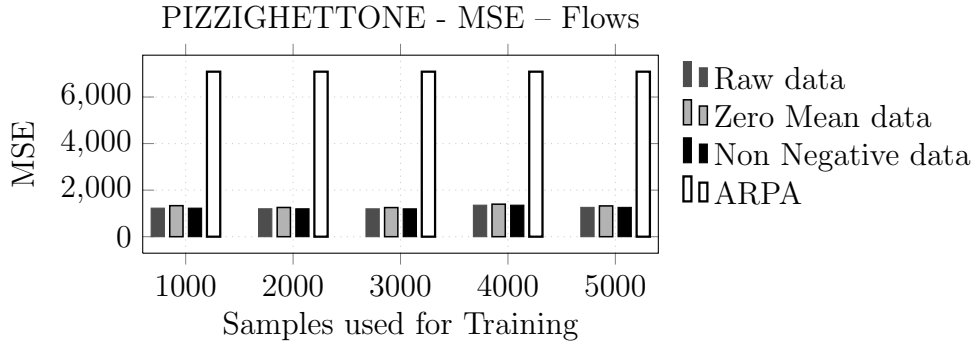


**Figure 4.16:** Mean–square errors for flows forecasts of nonparametric models and ARPA model at Pizzighettone. The histogram describes the change in MSE when the dataset used for training grows from 1000 to 5000 samples. It is also shown, for comparison, the MSE of ARPA forecasts.

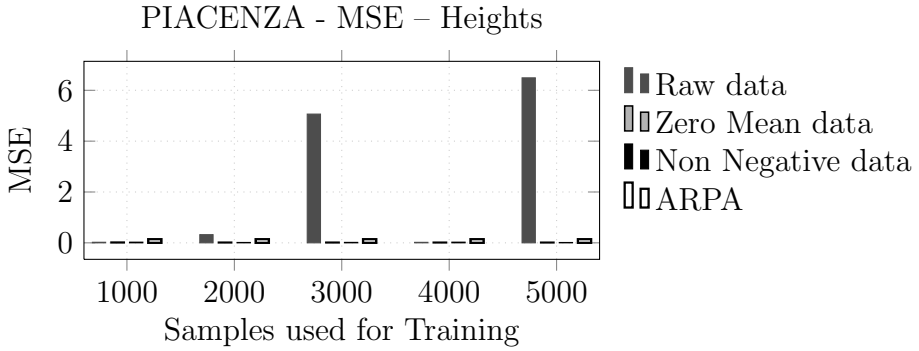


**Figure 4.17:** Mean–square errors for heights forecasts of nonparametric models and ARPA model at Piacenza. The histogram describes the change in MSE when the dataset used for training grows from 1000 to 5000 samples. It is also shown, for comparison, the MSE of ARPA forecasts.

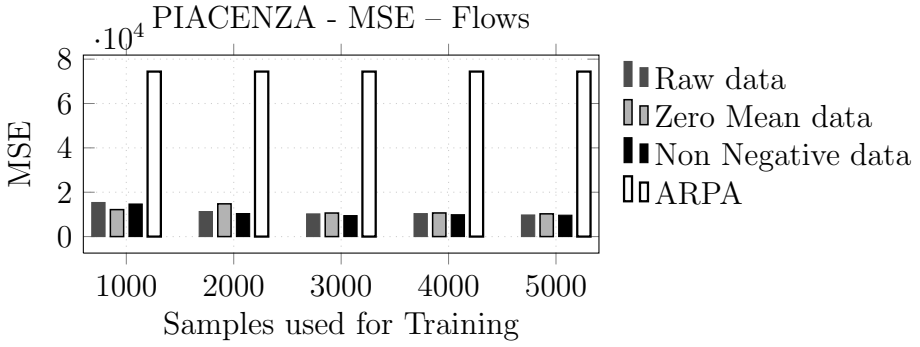


**Figure 4.18:** Mean–square errors for flows forecasts of nonparametric models and ARPA model at Piacenza. The histogram describes the change in MSE when the dataset used for training grows from 1000 to 5000 samples. It is also shown, for comparison, the MSE of ARPA forecasts.

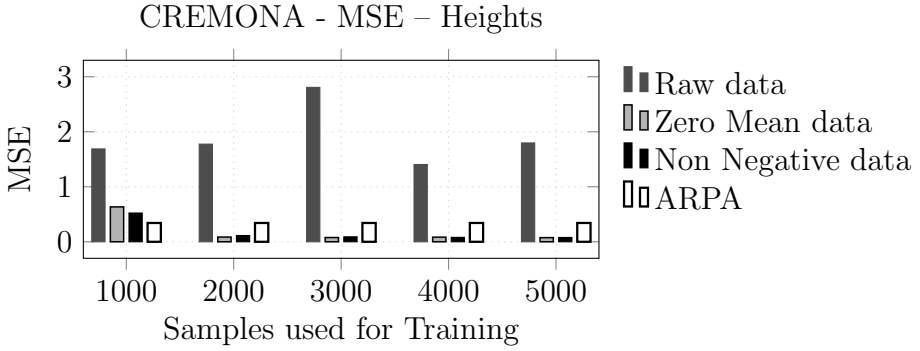


**Figure 4.19:** Mean–square errors for heights forecasts of nonparametric models and ARPA model at Cremona. The histogram describes the change in MSE when the dataset used for training grows from 1000 to 5000 samples. It is also shown, for comparison, the MSE of ARPA forecasts.

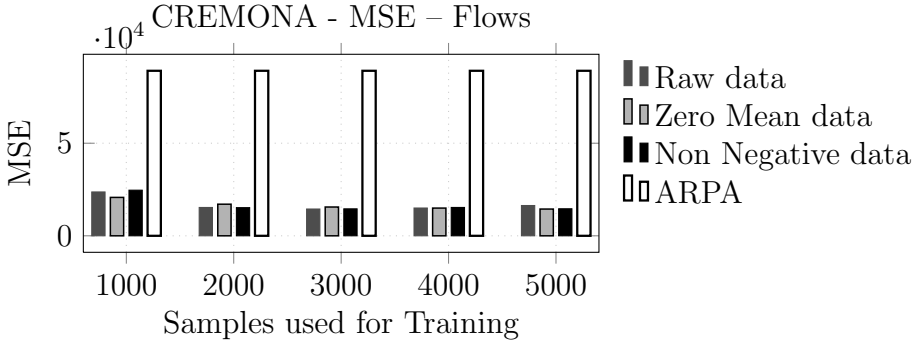


**Figure 4.20:** Mean–square errors for flows forecasts of nonparametric models and ARPA model at Cremona. The histogram describes the change in MSE when the dataset used for training grows from 1000 to 5000 samples. It is also shown, for comparison, the MSE of ARPA forecasts.

# 5

# Conclusions

The main motivation for this thesis was that of testing the performances of a nonparametric approach to the real case of the Po River basin. The extreme complexity of the system suggested that a block box approach could eventually perform better than a deterministic, physically based one. Moreover, the nonparametric approach guarantees the possibility of easy retuning/relearning of the model, which is a particularly useful characteristic when dealing with such an inherently time variant system as a river basin.

Thanks to the kind collaboration of ARPA, we were able to train and test our algorithm on a series of real databases.

The simulation results were satisfactory, both from the point of view of accuracy and efficiency. Despite requiring much lower computational load and time, the nonparametric algorithm obtained – according to MSE comparison – better performances than the current forecasting system.

We obtained best results in the case of stationery regime, namely when height levels and flow values change *slowly*. On the contrary, a certain delay appears in case of sudden variations, especially in the upstream stations. This is explained by the fact that rainfall forecasts are not at disposal of our algorithm, thus implying the modeling of upstream stations as auto–regressive systems

only. The AIPo forecast systems appear to be more capable of predicting the actual water rising time, while less accurate in providing the exact value of the increment.

On the contrary, non upstream stations are capable – in our model – to take into account the information from their upstream neighbors, thus providing a very accurate forecast both in timing and magnitude.

We guess that the inclusion of weather forecast data as additional inputs to our model could greatly help improving the performances, both to reduce (or remove) the upstream prediction delay and to refine all of the results.

A future research topic could be that of including some a priori information to the model, moving back from a completely black box approach to one that features some physical based characteristics.

Another research direction might be that of providing a theoretical framework for computing an approximate probability distribution of the predictions. In particular, assuming that the heights and flows under analysis are stochastic processes, by direct computation of the 12–steps ahead prediction one can in principle also argue that the actual value is distributed as a Gaussian random variable centered in the prediction and with variance given by the model. This could allow providing a reliable uncertainty range to Civil Protection, in order to plan emergency management according to a probabilistic scenario instead than on a single prediction value.

# Bibliography


Akaike, H. (1974). A new look at the statistical model identification. *IEEE Transactions on Automatic Control 19*(6), 716 – 723.

Aronszajn, N. (1950). Theory of reproducing kernels. *Transactions of the American Mathematical Society 68*, 337 – 404.

Bertsekas, D. P. and J. N. Tsitsiklis (1997). *Parallel and Distributed Computation: Numerical Methods*. Belmont, MA: Athena Scientific.

Bollen, K. A. and J. S. Long (1993). *Testing structural equation models*, Chapter Bayesian model selection in structural equation models, pp. 163–180. Newbury Park, CA: Sage.

Cucker, F. and S. Smale (2002). On the mathematical foundations of learning. *Bulletin of the American Mathematical Society 39*, 1 – 49.

De Nicolao, G. and G. Ferrari-Trecate (1999, November). Consistent identification of NARX models via Regularization Networks. *IEEE Transactions on Automatic Control 44*(11), 2045 – 2049.

Girosi, F., M. Jones, and T. Poggio (1995, March). Regularization theory and neural networks architectures. *Neural computation 7*(2), 219 – 269.

Goodwin, G., M. Gevers, and B. Ninness (1992). Quantifying the error in estimated transfer functions with application to model order selection. *IEEE Transactions on Automatic Control 37*(7), 913–928.

Hoerl, A. E. and R. W. Kennard (2000, February). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics 42*(1), 80–86. Special 40th Anniversary Issue.

Kimeldorf, G. and G. Wahba (1971, January). Some results on Tchebycheffian spline functions. *Journal of Mathematical Analysis and Applications 33*(1), 82–95.

Kimeldorf, G. S. and G. Wahba (1970, April). A correspondence between Bayesian estimation on stochastic processes and smoothing by splines. *The Annals of Mathematical Statistics 41*(2), 495–502.

König, H. (1986). *Eigenvalue distribution of compact operators*, Volume 9 of *Operator theory: advances and applications*. Basel-Boston-Stuttgart: Birkhauser Verlag.

Krzysztofowicz, R. (1999). Bayesian theory of probabilistic forecasting via deterministic hydrologic model. *Water Resour. Res. 35*, 2739 – 2750.

Ljung, L. (1999). *System identification: theory for the user*. Prentice Hall PTR.

Pillonetto, G. and B. M. Bell (2007, October). Bayes and empirical Bayes semi-blind deconvolution using eigenfunctions of a prior covariance. *Automatica 43*(10), 1698–1712.

Pillonetto, G., A. Chiuso, and G. De Nicolao (2011, February). Prediction error identification of linear systems: A nonparametric gaussian regression approach. *Automatica 47*(2), 291 – 305.

Pillonetto, G. and G. De Nicolao (2010, January). A new kernel-based approach for linear system identification. *Automatica 46*(1), 81 – 93.

Pillonetto, G. and G. De Nicolao (2012). The stable spline toolbox for system identification. Technical report, University of Padova and University of Pavia.

Poggio, T. and F. Girosi (1990, September). Networks for approximation and learning. *Proceedings of the IEEE 78*(9), 1481 – 1497.

Rasmussen, C. E. and C. K. I. Williams (2006). *Gaussian Processes for Machine Learning*. The MIT Press.

Schölkopf, B. and A. Smola (2001). *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. Cambridge, MA, USA: MIT Press.

Schwarz, G. E. (1978). Estimating the dimension of a model. *Annals of Statistics 6*(2), 461 – 464.

Smale, S. and D.-X. Zhou (2007). Learning theory estimates via integral operators and their approximations. *Constructive approximation 26*, 153–172.

Söderström, T. and P. Stoica (1989). *System Identification*. Prentice Hall.

Stein, M. L. (1999). *Interpolation of spatial data: some theory for Kriging*. Springer.

Tikhonov, A. N. and V. Y. Arsenin (1977). *Solution of Ill-posed Problems*. Wiston.

Todini, E. (2008). A model conditional processor to assess predictive uncertainty in flood forecasting. *Intl. J. River Basin Management 6*(2), 123 – 137.

Todini, E. (2010). Predictive uncertainty in flood forecasting and emergency management.

Todini, E. and L. Ciarapica (2002). A model for the representation of the rainfall-runoff process at different scales. *Hydrological Processes 16*(2), 207 – 229.

Van der Waerden, B. (1952). Order tests for two-sample problem and their power i. *Indagationes Mathematicae 14*, 453 – 458.

Van der Waerden, B. (1953a). Order tests for two-sample problem and their power ii. *Indagationes Mathematicae 15*, 303 – 310.

Van der Waerden, B. (1953b). Order tests for two-sample problem and their power iii. *Indagationes Mathematicae 15*, 311 – 316.

Wahba, G. (1990). *Spline models for observational data*. SIAM.

Weinert, H. L. (1982). *Reproducing Kernel Hilbert Spaces: Applications in Statistical Signal Processing*. Stroudsburg, Pennsylvania: Hutchinson Ross.

Yosida, K. (1965). *Functional Analysis*, Volume 123. Springer-Verlag.

Zhu, H., C. K. I. Williams, R. Rohwer, and M. Morciniec (1998). Gaussian regression and optimal finite dimensional linear models. In *Neural Networks and Machine Learning*. Springer-Verlag.

Zhu, K. (2007). *Operator theory in function spaces*. Number 138 in Mathematical Surveys and Monographs. American Mathematical Society.