Construction des espaces de représentation RPA pour l'analyse des signaux transitoires

Florin-Marian Birleanu

UNIVERSITÉ DE GRENOBLE

THÈSE

pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ DE GRENOBLE

Spécialité : Signal, Image, Parole, Télécoms

Arrêté ministériel : 7 août 2006

Présentée par

Florin-Marian BIRLEANU

Thèse dirigée par **Jocelyn CHANUSSOT** et codirigée par **Cornel IOANA**

préparée au sein du

Laboratoire GIPSA-lab (Grenoble Image Parole Signal Automatique)

dans l'**École Doctorale EEATS (Électronique, Électrotechnique, Automatique et Traitement du Signal)**

Construction des espaces de représentation RPA pour l'analyse des signaux transitoires

Thèse soutenue publiquement le **25 septembre 2012**, devant le jury composé de:

M. Alexandru SERBANESCU
PR Académie technique militaire de Bucarest - Roumanie, Président

M. André QUINQUIS
Déléqué Régional CNRS – Région Bretagne, Rapporteur

M. Philippe RAVIER
MCF Université d'Orléans, Rapporteur

M. Jocelyn CHANUSSOT
PR Institut Polytechnique de Grenoble, Membre

M. Guy D'URSO
ING EDF R&D Chatou, Membre

Mme Anne-Catherine FAVRE
PR Institut Polytechnique de Grenoble, Membre

M. Cornel IOANA
MCF Institut Polytechnique de Grenoble, Membre

Design of RPA representation spaces for the analysis of transient signals

Florin-Marian BIRLEANU

PhD Thesis

On n'enseigne pas ce que l'on sait ou ce que l'on croit savoir :
on n'enseigne et on ne peut enseigner que ce que l'on est.

(Jean Jaurès)

Contents

Chapter 1

Introduction

This thesis treats the subject of transient signal analysis in the context of ultrasonic monitoring of water flow in pipes. The following section presents the context and outlines the objectives. In brief, the thesis deals with the development of robust and efficient tools for transient signal detection and characterization. We present then the state of the art in transient signal analysis. We point out the limits of current methods, which motivate the study of new approaches. We also briefly discuss in this chapter the approach proposed by this thesis, which is based on operating with vector samples and computing generalized recurrences. The chapter closes with a brief guide of the thesis, which gives an overview of the content of the next chapters.

1.1 Context and purpose

Water is a fundamental element of biological life. For instance, the human body is formed in proportion of 60% of water. Likewise, over 70% of the surface of Earth is covered with water. Hence, is no wonder that water is ubiquitous in everyday life. Whether it is necessary for food or hygiene, or for irigations, or for cooling nuclear reactors, water must be transported from one place to another – often, on very long distances. Transport of water is achieved with the aid of pipes, and therefore pipe flow monitoring is an important technological need.

Monitoring water pipes targets detection of various abnormalities that may occur in the operation of such a system, e.g. dangerous pressure levels, fissures (caused by: temperature changes, vibrations, ground movements, abrupt pressure changes inside the pipe, corrosion, etc.), deposits on the inner walls of the pipe, etc. Most of the methods that are currently used for detecting these problems are invasive methods, and they require the insertion of some measuring intrument (e.g. for measuring temperature and flow rate) in the transport chain. Other methods require a complete shutdown of the system, and possibly even emptying of the pipes (as in the case of inspecting the inner walls of the pipe with the aid of robots).

In this context noninvasive methods are the most desirable ones, as they do not require shutting down the system or inserting additional elements inside the pipe. Among the noninvasive techniques, those based on ultrasounds occupy a very important place. They

require mounting of some ultrasonic transducers on the outer walls of the investigated pipe. These transducers emit and receive signals that propagate through the water in the pipe and are affected by the state parameters of this water. The emitted signals are short impulsive signals. The received signal will contain an impulse corresponding to the emitted burst (but having various parameters (eg. amplitude, envelope, frequency, etc.) modified), as well as eventual reflections or propagations on other trajects.

Figure 1.1[1] shows a sample setup of measuring flow rate using ultrasounds. It also shows both the emitted and the received signals. It can be noticed that the emitted signals arrive at the receiver with certain delays, that depend on the velocity of the water flowing inside the pipe. Estimating water velocity requires, therefore, an accurate estimation of the times of arrival of these signals. (This task is even more difficult when the received signals contain noise.)

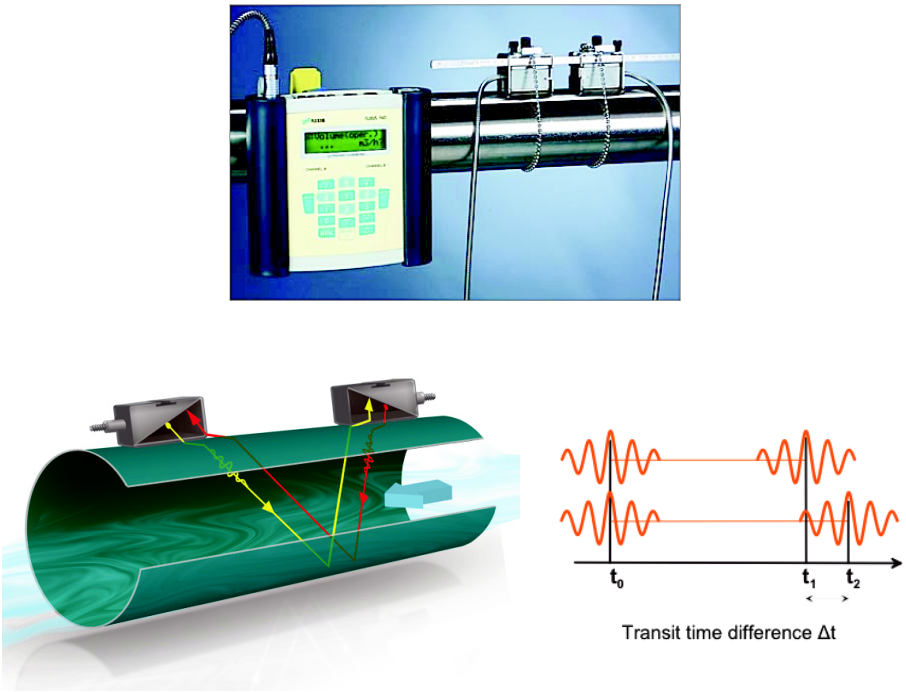


Figure 1.1: Sample setup for measuring flow rate in a pipe using ultrasounds.[1]

Consequently, signal processing is an essential component of ultrasound based measuring techniques. Transient signal procesing, to be more precise. Throughout this paper we call *transients* those signals that are impulsive, ephemeral, and have a length that is finite and most often very short compared to the total length of the recorded signal. For instance, we consider transients the impulsive signals in Figure 1.1.

The purpose of this thesis is to develop new signal analysis tools that are capable to offer accurate and robust results in the detection and characterization of transient signals. When speaking about transient signal detection, we mean the identification of its time support. Clasically, detection refers to estimating the probability that some observed signal contains a certain (usually, a priori known) signal (in this case, a transient signal). However, the target of our detection is to obtain, for each signal sample, an

[1]We used figures from `http://www.flexim.com/us/fluxus-f601` and `http://www.flexim.com/us/measurement-principle`.

estimation of the probability that that sample belongs to a transient. On the other hand, when speaking about transient signal characterization we refer to extracting the defining characteristics of a transient. Some such characteristics may be: energy, frequency band, maximum amplitude, envelope, initial phase, histogram, etc. It is not difficult to notice that detection is actually a special case of characterization – as what detection actually does is estimating two defining parameters of the transient, i.e. its time of arrival and its length. Nevertheless, given the paramount importance of these two parameters (at least for ultrasound based measuring techniques), we treat detection separately from characterization.

In the next section we review the state of the art in ultrasound based intrumentation, as well as in transient signal analysis methods. We present the basic principles of ultrasonic measurements, and then we briefly discuss the main approaches for analyzing the specific signals involved.

1.2 State of the art

Before presenting the state of the art in transient signal analysis, we consider necessary to first outline the state of the art in ultrasonic based inspection of water pipes. In this way we show more clearly where the analysis of transient signals occurs in this type of measurement techniques, and also why it is necessary to develop the arsenal of tools that can be used for this analysis.

1.2.1 Ultrasonic measurements

Since more than forty years, ultrasonic measuring techniques have been used in various fields and they continue to develop. Their main advantage compared to the classic measuring instruments is noninvasiveness. They do not directly touch the investigated liquid[2], and their mounting does not require shutting down the system. Contact with the investigated environment can be done by simply placing two acoustic transducers on the outer walls of the pipe.

Two of the possible applications of ultrasounds in the field of fluid analysis are:

- level measurements, which in turn can be divided into presence/absence tests, and (proper) level measurements
- and flow rate measurements.

Liquid level measurements aim to measure the level of the liquid inside containers or pipes. In certain situations ultrasonic measurement is the only viable solution – as is the case for containers that can not be opened for safety reasons, or when the nature of the chemical product or process does not allow mounting a classic level transducer[3].

[2]We point out that although throughout this paper we refer strictly to liquid (and especially water) measurements, similar methods can be used for gas measurements.

[3]See: `www.olympus-ins.com`, `www.rickly.com`.

There are a large number of fields where ultrasonic level measurements can be used, from detecting stationnary liquids in pipes that are being opened for maintenance purposes, to measuring the level of reactive liquids in chemical applications, or measuring the thickness of a petroleum layer floating over water in an oil processing system.

Both presence/absence tests and level measurements use a similar measurement principle. It consists in first sending an acoustic signal from an ultrasonic transducer placed on the bottom of the container (or pipe) and then analyzing the signal received (by the same transducer, that plays a double, time-multiplexed, role: first it acts as emitter, then it acts as receiver). The emitted signal penetrates the wall and propagates through the liquid until it reaches the liquid-air interface (or the interface between the liquid and the opposite wall, if the container is full), where it suffers reflection. The time instant when this reflection arrives back at the transducer offers information about the distance travelled by the signal (when the parameters of the wall and those of the liquid are known). In addition, the lack of this reflection in the received signal is an indicator of the fact that the container is empty (– fact that the measuring device may signal by optical or acoustical means). We must mention that even when the container is empty, the receiver may still record some reflections, i.e. the reflections caused by the propagation through the container wall (especially when it is a thick metallic wall). We also mention that measuring level can be perfomed indirectly as well, by performing several presence/absence tests transversally on the container, by placing the transducer at several positions on the vertical wall of the container.

Ultrasonic measurements are very attractive in many ways, but some cautions must be taken when using them. For ensuring a good transfer of the acoustic energy between the transducer and the pipe wall, an ultrasonic couplant must be used (as air attenuates very strongly the acoustic wave). This couplant may be water, grease or hair gel, but special couplants are the most efficient (– for instance, nucler industry uses couplants whose halogen and sulfur is limited and well documented). The acoustic properties of the liquid are very importand, as well – a liquid that is very viscous strongly attenuates the acoustic wave. Besides these, the thickness, shape and state of the wall surface may negatively influence the measurement accuracy. In addition, one must take care that for the measurement method to work properly there must not be any obstructions in the path of the acoustic wave through the liquid.

The most spread of the two classes of ultrasonic applications mentioned above is flow rate measurement. Various kinds of liquids may be concerned (starting from water, waste water, hot water and beverages, to chemicals, crude oil, oil products and fuels), in fields such as: water distribution, irrigations, monitoring energy consumption, food industry, metallurgical and mining applications, thermal and nuclear plants. Additionally, ultrasonic flow meters can be used for measuring flow rate in rivers and open channels.

There are two kind of ultrasonic flow meters:

- Doppler based ultrasonic flow meters
- and transit-time difference based ultrasonic flow meters.
- (There are also hybrid ultrasonic flow meters, that use both technologies[4].)

[4]Such a hybrid ultrasonic flow meter is Duosonics from Fuji Electric (`http://www.fujielectric.`

Doppler based ultrasonic flow meters require the presence of sonic reflectors in the liquid to be investigated. These reflectors can be air bubbles, mud or any other particles that reflect acoustic waves. Flow rate of clean liquids may be measured as well, by using reflections caused by the eddies (that appear inside the liquid when flow becomes turbulent). In all cases, it is required that the pipes be completely full.

The measurement principle of a Doppler flow meter consists in sending an acoustic wave into the liquid. This wave will be reflected by the sonic reflectors in the liquid (that move (hopefully!) with the same velocity as the liquid). According to the Doppler effect, the received signal will exhibit a frequency deviation from the original frequency of the emitted signal. This frequency deviation depends on the velocity of the reflective particles in the liquid, hence it can be used for computing the velocity of the liquid.

Unlike Doppler flowmeters, transit-time difference flow meters are useful when investigating liquids that are relatively clean, i.e. that do not contain particles that could significantly obstruct the acoustic wave path. These flowmeters can use various positions for the acoustic transducers – the most common are shown in Figure 1.2[5]. It can be noticed that what actually varies between the different configurations is the number of sound paths that the wave has to travel from the emitter to the receiver. More sound path are needed for pipes with small diameter, in order for the acoustic wave to be *visibly* affected by the flow.

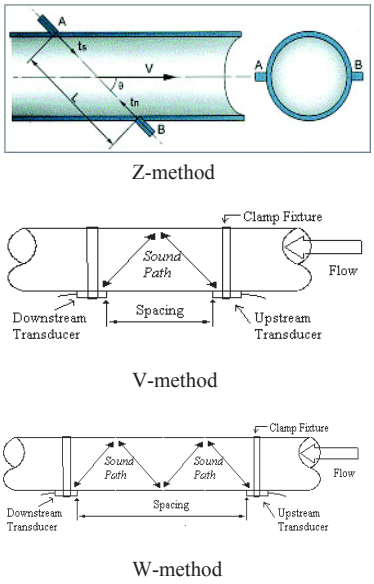


Figure 1.2: Typical measuring configurations for transit-time difference based ultrasonic flow meters: Z configuration (used especially for large pipes), V configuration (used especially for medium size pipes), W configuration (used especially for small pipes).[5]

Regardless of the measuring configuration, all transit-time based flow meters use a similar measurement principle. The two transducers act in turn as emitter and receiver. First, the signal is emitted in one direction. It propagates through the pipe and reaches the

com/products/instruments/library/catalog/pdf/ECXNO632d.pdf).

[5]See http://www.shenitech.com/support/Brochure_STUF-300H.pdf.

receiver, and then it is emitted in the opposite direction. This second signal propagates through the pipe and reaches the first emitter (which now acts as receiver). In the case where the liquis is static, the two received signals will be nearly identical. Otherwise, there will be a difference between the time instants when the emitted bursts reach the receivers. This time difference depends on the liquid velocity, hence it can be used for computing it.

Two of the parameters that most strongly affect the measurement accuracy are the precision of the geometric data of the setup, and the method used for determining the transit times. Besides that, one must also account for the fact that the flow profile is not known with certainty[6]. Therefore, the relation between the speed with which the wave is slown down or accelerated on its path and the mean flow velocity is not known with certainty. We must also not forget that the speed of wave propagation in the liquid is not a constant[7], but depends on pressure and temperature. Changes of this propagation speed affect the accuracy of computing the flow rate.

As for determining transit-times, we must point out that in present day ultrasonic flow meters[8] it is performed either based on measuring the first ascending edge of the signal, or by correlating the received signal with the emitted one.

By combining the two classes of ultrasonic measurement applications presented in this section (i.e. level measurement, and flow rate measurement) flow rate measurements in rivers and open channels can be performed as well. More precisely, an ultrasonic flow meter will be used for measuring water velocity (posibly, at various depths – in order to provide increased accuracy), and a level detector will measure level. Having these measures and given the channel geometry, the flow rate can be easily computed.

Despite the variety of fields where they can be useful, we notice that ultrasonic measurement equipments have similar operating principles. Regardless of their precise purpose, they ultimately lead to similar signals, out of which we must extract as accurately as possible some parameters. Hence, they all need signal analysis method that are able to extract these parameters in an automated manner, i.e. with minimum human operator assistance, and (ideally) with better efficiency.

1.2.2 Transient signal detection and characterization

Although we encounter them in various fields, transient signals do not have an unanimously accepted definition. It is easier to state what transients are not than what they are. Transients are not the infinite length signals that mathematicians would love. Transients are not stationnary, *well-behaved* signals. Transients are not signals that can be easily associated to some signal model. Transients may appear in recordings of short sounds, heart beats, partial discharges in electric cables, etc. Any signal that is charac-

[6]This is why some companies use in their flow meters (e.g. KROHNE UFM 3030 – `http://www.instrumart.com/assets/UFM3030_datasheet.pdf`) several ultrasonic fascicles, in order to better estimate the flow profile by sectioning it on several directions.

[7]There are ultrasonic flow meters that use proprietary algorithms for automatically tracking the sound propagation speed in the liquid (e.g. Panametrics DF 868 (`http://www.ge-mcs.com/en/flow/ultrasonic-liquid/digitalflow-df868.html`) uses ATW – Automatic Tracking Window).

[8]See the ultrasonic measuring products of GE Panametrics (`http://www.ge-mcs.com/`) and Flexim (`http://www.flexim.com/`).

terized by an abrupt change, a short nonstationnarity, a fast change of rythm, may be considered to be a transient signal. Figure 1.3[9] illustrates some examples of transient signals coming from different sources. The common feature of all transient signals is their transiency – they all have a short lifetime compared to the observation time. This feature turns their analysis into a difficult task, that require special approaches.

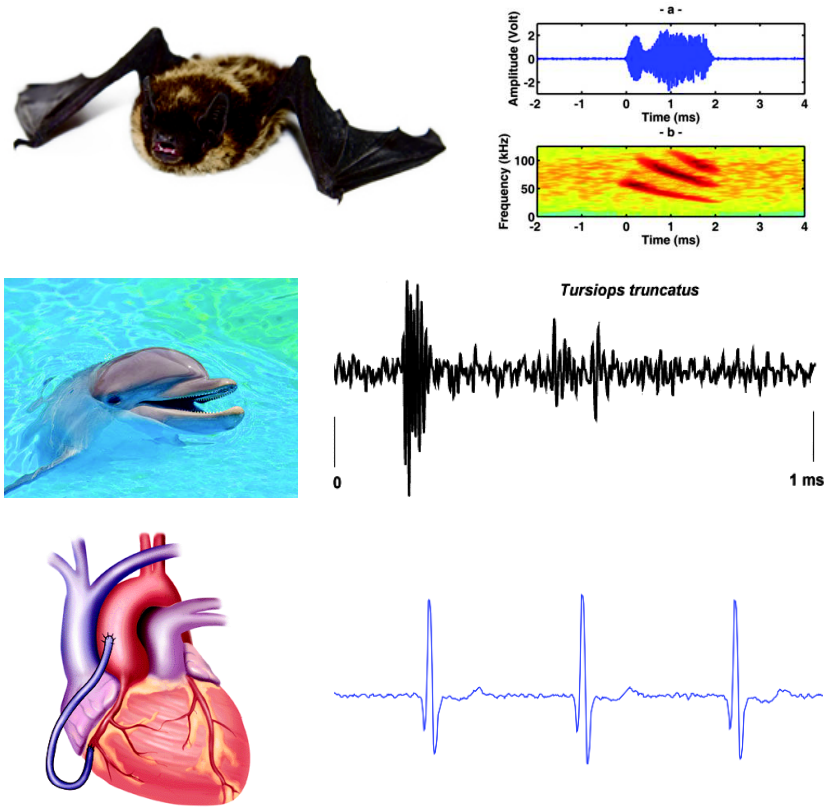


Figure 1.3: Examples of different biological transient signals: (top) bat signal, (middle) dolphin signal, (bottom) heart signal. While bats and dolphins use transients for doing echolocation, doctors use the heart signals to find specific health problems.[9]

The impulsive signals that were involved in the ultrasonic measurements discussed in the previous subsection are transient signals, as well (due to their duration, that is finite and short compared to the duration of the entire recorded signal). As we have control on the emitted signal (i.e. it is known), it would seem that for this particular application (i.e ultrasonic pipe measurement) a method that is based on a model for the transient signals would be the most desirable one. However, this thesis does not aim to develop methods that are adapted only for a very narrow class of transient signals. On the contrary, its purpose is that the proposed methods work fine also outside the borders of the application context, which it embraced only as a pretext for developing generic and robust transient signal analysis methods. Therefore, we are mostly interested in the case where no a priori knowledge is available about the transients to be analyzed.

[9]We used images from `http://www.about-bats.com/graphics/bat.jpg`, `http://www.acoustics.org/press/154th/Echolocation_Pulse.jpg`, `http://l.yimg.com/ea/img/-/110201/vdg_red_balloon_dolphins_16kehpi-16kehq5.jpg`, `http://www.wiltonlibrary.org/uploads/heart.gif`, and `http://www.swharden.com/blog/images/diy_ecg3.png`.

There are several classes of approaches when it comes to detecting transient signals. We list next the most notable ones.

- Energy detection

  This is the classic solution to the problem of detection. Basically, such a detector is based on computing the signal energy on a time-shifted window of signal samples. The energy detector is usually used as a lower limit reference when studying the detection efficiency of new detectors.

- Time domain analysis

  This class of approaches comprises solutions such as: thresholding the temporal shape of the signal with a threshold that exceeds the noise level, detecting the envelope of the analyzed signal, and correlating the analyzed signal with a knwon reference signal. The first two solutions we mentioned have the advantage of low computational requirements, but they are efficient only when noise is very weak. Otherwise, they do not work well at all. However, the correlation[10] method works well even under noise conditions. In order to use it, we must know in advance the temporal shape of the transient we are searching for. Correlating this a apriori transient with the analyzed signal will generate maxima at the time instants where it is found. The method is based on the suppositions that noise and transients are completely decorrelated and that the actual transients are almost identical to the reference one. Hence, the received transient must be similar to the transmitted one, which is not the case in most of the real applications.

- Time-frequency analysis

  Having in mind that nonstationnarity and time localization are some important features of transient signals, a time-frequency approach seems to be very legitimate. Still, in general the frequency band of transients is unknown, and their short duration prevents obtaining a good localization in frequency. Besides that, they are difficult to detect under noise conditions, especially when their energy is widespread in the time-frequency plane.

- Time-scale analysis

  Time-scale analysis offers the possibility of locally inspecting the analyzed signal with the aid of transient-like signals, using, for example, the concept of wavelets (that have the main features of a transient, i.e. they are brief and time-localized). This kind of analysis provides good results when the wavelets are well adapted to the transients in the analyzed signal. Unlike time-frequency analysis, which performs a rigid decomposition of the signal in the time-frequency plane, wavelet analysis allows building time-frequency bases that are more adapted to the signal, acording to some (most often, energetic) criteria.

- Statistic analysis

  Statistic approach is mainly based on the supposition that transients do not have a Gaussian histogram. This is the feature that distinguishes them from noise, which

[10]We make the observation that most of the transit-time ultrasonic flow meters available today rely on correlation for estimating the transit times. The emitted signals are built such that they be as decorrelated with noise as possible and have an autocorrelation function that is as *peaked* as possible.

is supposed Gaussian. Hence, measuring the local deviation from Gausianity of the analyzed signal can reveal the transients that it contains. This can be done, for instance, with the aid of higher order statistics like the kurtosis [LAC97]. An important constraint in the statistic approach is related to time localization. On one hand, we want to obtain a very precise identification of the time support of transients, and, on the other hand, statistic estimators have no relevance if they are computed using an insufficient amount of signal samples.

The list is not comprehensive, but it contains the main classes of appraches. Besides these, interesting solutions were obtained by combining some of these aproaches. For instance, the author of [Rav98] used higher order statistics as a criterion for separating transient coefficients from noise coefficients in a wavelet decomposition of the analyzed signal.

Characterization of transient signals aims to go further than their simple detection in time. It aims to extract additional knowledge from the detected transients (i.e. not only the time and duration, that are generally given by detection). Based of this additional knowledge, one would then be able to distinguish or classify different types of transients. For instance, in the context of underwater signal analysis one question would be to distinguish the sounds generated by a ship from those generated by marine mammals.

A relatively recent actor that entered the scene of signal processing is recurrence plot analysis (RPA) [MCRTK07]. The thesis will show that RPA has some interesting words to say regarding transient signal detection and characterization.

1.3 Proposed approach

In this thesis we address the problem of transient signal analysis by using phase space *recurence*, a concept that comes from dynamical systems theory. This concept, although similar to frequency, is more flexible, being able to extract additional information from the analyzed signals. For instance, two different signals having similar histograms and spectra might be distinguished easily by checking their recurrence patterns. We show in this thesis how by involving this concept we may define new representation spaces for signals, where various transient signal related problems can be easily solved. We also provide comparisons with classic and existing solutions.

Despite the significant number of results of recurrence plot based methods in various fields involving time series analysis, addressing typical signal processing problems in terms of recurrences is not a common practice yet. There have been, though, some works regarding recurrence based signal detection [ZGWJ98, ZGWJ00, MNB08, RNDB08, MK09] and fundamental frequency estimation [Ter02]. At a higher scale, this thesis aims to contribute to paving the way of RPA into signal processing, by trying to show that RPA is nothing but a generic framework that (usefully) extends several classic signal processing tools (by also making room for new tools, as well).

The following section provides an overview of the thesis content.

1.4 Thesis outline

While our central application for transient signal processing is the ultrasonic investigation of water flow inside pipes, Chapter 2 addresses the problem of numerically simulating the interaction between ultrasounds and (arbitrarily) flowing water. We present HydroSimUS, a software application that we built in order to simulate acoustic wave propagation in water. We later use this application for performing "offline measurements" on different simulated pipe flow configurations.

The central chapters of the thesis, Chapters 3 and 4, are dedicated to the analysis of signals. In Chapter 3 we enter into the details of analyzing the transient signals that we will have already met in the previous chapter, we discuss the difficulties encountered in their analysis, as well as the existing solutions. Then we present the recurrence plot analysis (RPA) method, a more recent approach (based on the concept of phase space recurrence) which we adopt and generalize in signal processing terms – by introducing the vector samples processing (VeSP) concept. Afterwards, we show that by combining the concepts of *vector sample* and *recurrence* we can extend tools such as the autocorrelation function and the histogram. We show in Chapter 4 that these new tools lead to some representation spaces where different signal features are highlighted, thus facilitating the detection and characterization of transient parts in a signal. We also show that they provide good results in various transient signal analysis related tasks, and we compare them to existing solutions.

In Chapter 5 we make use of the tools and methods presented in the previous three chapters in order to illustrate solving of some practical problems related to ultrasonic investigation of water pipes. First, by using the software tools described in Chapter 2, we simulate various ultrasonic measurements, targeting problems such as estimating the level and the velocity of water ($\leftarrow$ direct problem). Then, we use the signal analysis methods developed in Chapter 3 (and illustrated in Chapter 4) for estimating from the obtained signals the real parameters of the water inside the pipe ($\leftarrow$ inverse problem).

Chapter 6 presents the conclusions of the thesis, the main contributions that it brings, as well as some interesting directions for broadening and deepening the research work described in this manuscript.

Information about the software tools that we built while working on this thesis, grouped into two public Matlab toolboxes – *HydroSimUS*, and *VeSP* – can be found in the Appendix.

Chapter 2

Modeling of acoustic wave propagation in arbitrary pipe flows

The nonintrusive analysis of pipes constitutes a challenging problem that transient signal processing must face. In order to define the appropriate analysis methodology for extracting the parameters of the flow, an accurate modeling of acoustic wave propagation must be done. This is the subject of the current chapter, where we treat the problem of numerically simulating ultrasonic wave propagation in water pipes. We start with an overview of the fundamentals of fluid dynamics and acoustics, and then we get into the implementation details of the simulator (that we called *HydroSimUS*). We address topics such as the numerical solution of the wave equation in a moving environment, the simulation of the emitter, and the computation of the different water velocity profiles.

2.1 Preliminaries

Turbulence has a strong impact on the accuracy of any ultrasonic flow meters. They only measure the average effect of the flow on the propagation time of the ultrasonic wave. But the overall flow can be highly irregular, which leads to inaccurate estimations of the global average velocity of the liquid. Under normal operating conditions, the apparition of turbulence may indicate abnormalities, such as unwanted deposits on the inner walls of the pipe, or leakages. Besides measuring flow rate, it would also be very interesting to develop ultrasonic nonintrusive techniques for detecting (hopefully, as in advance as possible) the apparition of such problems.

Turbulent flow is a physical phenomenon that still raises problems, as it is not yet fully known. Even though the Navier-Stokes equations model very well fluid dynamics, they could not be analytically solved due to the nonlinearities they contain[1] [TL72, Tri77, Pop00, Man04, Les08, Tsi09]. In the case of turbulent motion, the contribution of the nonlinear term is significant. As the micro-scale local interactions between the liquid particles are strongly nonlinear, the global (macro-scale) behaviour becomes practically

[1]The Clay Mathematics Institute of Cambridge, Massachusetts, USA, offers a prize of $1 million for "*substantial progress toward a mathematical theory which will unlock the secrets hidden in the Navier-Stokes equations*" (`http://www.claymath.org/millennium/Navier-Stokes_Equations/index.php`).

unpredictable.

Our approach to studying turbulence in a pipe consists in simulating the propagation of an ultrasonic signal in a *frozen* flow, i.e. by taking into consideration the local velocities of the liquid in each point at a specific time instant. That is, we compute the local velocities of the liquid in each point for a certain turbulent situation at a certain time instant, and then we solve numerically a modified version of the wave propagation equation that takes into consideration these local velocities. The connection between fluid dynamics and signal propagation, that are merged in our simulation toolbox, is the main contribution of this part. More details about the simulator can be found in Section 2.2. The current section continues with presenting some fundamentals of fluid dynamics and acoustic wave propagation.

2.1.1 Fluid dynamics

Flow regime in a pipe is usually characterized with the aid of a dimensionless number, the Reynolds number, Re, which quantifies the ratio between inertial forces and viscous forces in the liquid. Its value is directly proportional to the pipe diameter and to the mean velocity of the liquid, and is inversely proportional to the kinematic viscosity of the liquid. While for small values of the Reynolds number fluid flow through a pipe is modeled well by a Poiseuille velocity profile (see Section 2.2.3), when Re increases the flow becomes more complicated, turning turbulent [Tri77]. In this case, small irregular phenomena are impredictabilly amplified, due to the flow of the fluid. A fluid particle that starts from a same point at different time instants will follow different paths. If we follow many such paths, we can observe that they mix in a very complicatd manner. Globally, all the fluid particles go in the direction of the flow, but some of them will locally move even in the opposite direction. A fully developed turbulence produces a good mixing of the fluid.

Any flow, whether laminar or turbulent, is modeled by the Navier-Stokes [Deb97, BHSL02] equations:

$$\begin{cases} \nabla \cdot \vec{v} = 0 \\ \frac{\partial \vec{v}}{\partial t} + (\vec{v} \cdot \nabla)\vec{v} = -\frac{1}{\rho}\nabla p + \nu \nabla^2 \vec{v} + \frac{1}{\rho}\vec{F}_{ext} \end{cases}, \tag{2.1}$$

where: $\vec{v}$ is the local fluid speed, p is the local fluid pressure, ρ is the fluid density, ν is the kinematic viscosity, and $\vec{F}_{ext}$ represents the external forces that act on the fluid. If we add boundary conditions and initial conditions, we obtain a complete equation system, whose solving would give the state of the fluid in every point, at every time instant.

Up to the present date, however, these equations have not been solved analytically. Instead, numerical methods for their solving have been developed [Wil93, GHL96, GDN98, CCY09]. They allow to obtain the values of the state variables of the fluid on a fixed mesh, for discrete time instants[2]. A first approach is represented by DNS (Direct Numerical Simulation). It consists in solving the equation system on a mesh that is fine enough to capture even the smallest details of turbulent flow. The problem of this approach is that such a fine mesh would involve an enormous computation power. In order to simplify the computations (with the cost of decreasing the precision of the results), an averaging of

[2]Software applications like ANSYS Fluent are able to simulate fluid flows through complicated geometries. (`http://www.ansys.com/Products/Simulation+Technology/Fluid+Dynamics/ANSYS+Fluent`)

the Navier-Stokes equations has been done (RANS – Reynolds Averaged Navier-Stokes), together with introducing a turbulence model, e.g. the k-ε model. This is the most used method. Another direction consists in large eddy simulation (LES) [WHS06]. Besides these, current efforts are being made, grouped into a very active research field.

2.1.2 Acoustics

Sound is an elastic wave that consists in the propagation of a pressure oscillation. The human ear can generally perceive sounds having frequencies between 20 Hz and 20 kHz. An ultrasound is an acoustic wave [KFCS00, EH02, Rai06] whose frequency is above the audible range, i.e. above 20 kHz.

In order to model the propagation of an acoustic wave in a dynamic liquid environment, we must reconsider how the wave equation [KP77, Pai05] is obtained:

$$\nabla^2 p - \frac{1}{c^2}\frac{\partial^2 p}{\partial t^2} = 0. \tag{2.2}$$

In this equation p denotes the acoustic pression, and c is the speed at which sound propagates through that environment. We must introduce in this equation the fact that the liquid moves with a certain speed, assumed, for the moment, constant in the whole volume. Under this condition, we have the following three equations [Bru98]:

- the law of mass conservation:

$$\left(\frac{\partial}{\partial t} + \vec{v}\cdot\nabla\right)\rho' + \rho_0 \cdot \nabla \cdot \vec{u} = 0, \tag{2.3}$$

- Euler's equation:

$$\left(\rho_0\frac{\partial}{\partial t} + \vec{v}\cdot\nabla\right)\vec{v} + \nabla p = 0, \tag{2.4}$$

- the law of adiabatic transform:

$$p = c^2 \cdot \rho'. \tag{2.5}$$

By applying the gradient to the law of mass conservation, taking the derivative of Euler's equation and using also the law of adiabatic transform, we obtain the linear equation of acoustic pression (see [Bru98], page 502):

$$\nabla^2 p - c^{-2} \cdot \left(\frac{\partial}{\partial t} + \vec{v}\cdot\nabla\right)^2 p = 0, \tag{2.6}$$

where: p is the acoustic pressure perturbation, c is the (constant) sound propagation velocity inside the fluid, and $\vec{v}$ is the local fluid velocity.

This equation models wave propagation in the case of a constant velocity, $\vec{v}$, of the liquid. Hence, it is not valid for any velocity profile. But we can use it by meshing the whole mass of liquid. We split this mass into small volumes whose local velocity is assumed to be constant. Therefore, an implementation of this equation can be done in two steps:

- obtaining of the local velocities of the liquid particles, in any domain point where we want to model the propagation of the wave
- and then meshing the equation, by using finite differences.

We detail these steps in the following section.

2.2 Numerical simulation

2.2.1 Wave equation in flowing environment

We recall that the propagation of ultrasounds in water can be mathematically modeled by the wave equation, which was shown in (2.2). This equation is valid only in static environment (that is, when water does not flow). If water flows at a constant velocity (that is the same in the whole volume of water), then (2.2) becomes (2.6) (as discused in the previous section). Again, Equation (2.6) is valid only if we assume that the water velocity, $\vec{v}$, is constant. Besides the space constancy, we also make the supposition of time constancy (during the propagation simulation time). We base this last supposition on the fact that ultrasounds travel through water with a speed that is much higher than the speed at which the local velocities of the fluid change.

By using (2.6) we can model the propagation of an acoustic wave through liquids having arbitrary flow if we consider the whole volume of liquid split into small volumes, as in Figure 2.1. The dimensions of this volume must be sufficiently small for the conditions on which the obtaining of Equation (2.6) was based to remain valid. Otherwise said, these small volumes must be sufficiently small such that the local water velocity can be considered to be the same in the whole small volume. As for the time constancy of water velocity, it can be assured if the wave propagation speed is much higher than the maximum local flowing speed of the water.

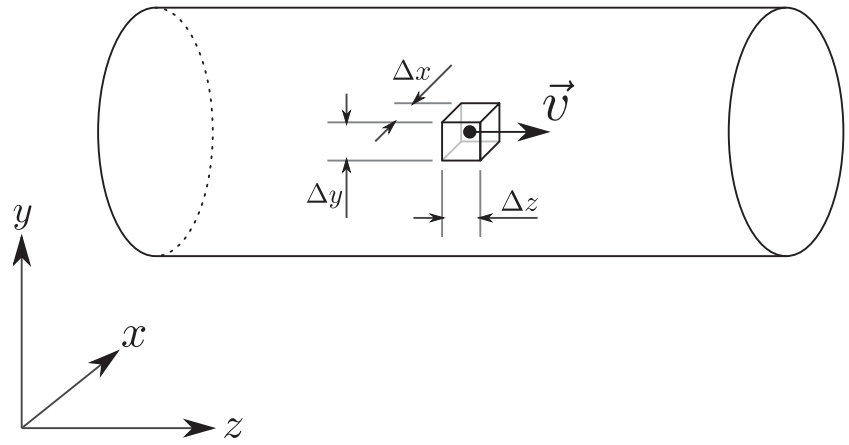


Figure 2.1: The simulaton domain is split into sufficiently small sub-volumes such that the local velocity of water can be considered to be the same in the whole sub-volume.

The numerical solving of Equation (2.6) can be performed by using different approaches [GDN98, HC00]: finite difference, finite volume, finite element. We opted for

the finite difference method, due to its simplicity. It consists in approximating the derivatives in (2.6) with finite differences, i.e. differences between the values of elements located at discrete positions in the simulation domain. In other words, we do not perform simulation on the entire (continuous) domain, but only on a (discrete) network of points spread over the entire domain and sufficiently close to each other.

The fundament of the finite difference method can be found in the definition of the derivative itself. The derivative of a function f in a point x is defined as the limit:

$$f'(x) = \lim_{\Delta \to 0} \frac{f(x+\Delta) - f(x)}{\Delta}. \tag{2.7}$$

Hence, if Δ is very small, then the ratio $(f(x+\Delta) - f(x))/\Delta$ approximates the value of the derivative of f in x, i.e. $f'(x)$. The smaller Δ is, the better is this approximation.

The finite differences used for approximating derivatives are obtained in a more theoretically sound manner by using Taylor series expansion [HC00]. This also allows estimating the approximation precision for the different finite difference approximation schemes. For instance, let us consider the case of approximating the first order derivative in x for function f. The Taylor series expansion of function f around point x is the following:

$$f(x+\Delta) = f(x) + f'(x) \cdot \Delta + \frac{1}{2} \cdot f''(x) \cdot \Delta^2 + \frac{1}{6} \cdot f'''(x) \cdot \Delta^3 + ... \tag{2.8}$$

We can obtain an approximation for $f'(x)$ by considering only the first two terms in the above expansion. Equation (2.8) becomes:

$$f(x+\Delta) = f(x) + f'(x) \cdot \Delta + \mathcal{O}\left(\Delta^2\right), \tag{2.9}$$

where $\mathcal{O}(\Delta^2)$ is a residual which indicates that truncation of the expansion was performed starting from the term that contains Δ^2. From (2.9) immediately follows that $f'(x)$ is:

$$f'(x) = \frac{f(x+\Delta) - f(x)}{\Delta} - \mathcal{O}(\Delta), \tag{2.10}$$

where by $\mathcal{O}(\Delta)$ we noted the quantity $\mathcal{O}(\Delta^2)/\Delta$, thus indicating that the most significant term in the residual is the term that contains Δ. Otherwise said, if we approximate $f'(x)$ with the finite difference in (2.10) (which is the same as in (2.7)), then we make an approximation error whose value is dictated mostly by the value of the term that contains Δ from the Taylor series expansion (in (2.8)). However, this is not the only way $f'(x)$ can be approximated. Let us consider besides (2.8) the following Taylor series expansion for f:

$$f(x-\Delta) = f(x) + f'(x) \cdot (-\Delta) + \frac{1}{2} \cdot f''(x) \cdot (-\Delta)^2 + \frac{1}{6} \cdot f'''(x) \cdot (-\Delta)^3 + ... \tag{2.11}$$

By considering only the first three terms in the expansions in (2.8) and (2.11) and by performing the difference between these two equations we obtain:

$$f(x+\Delta) - f(x-\Delta) = 2 \cdot f'(x) \cdot \Delta + \mathcal{O}\left(\Delta^3\right), \tag{2.12}$$

where by $\mathcal{O}(\Delta^3)$ we noted (as we did before) the approximation residual, whose most significant term is in this case the term that contains Δ^3. From (2.12) we can obtain another finite difference approximation for $f'(x)$:

$$f'(x) = \frac{f(x+\Delta) - f(x-\Delta)}{2\Delta} - \mathcal{O}\left(\Delta^2\right). \tag{2.13}$$

Compared to the approximation in (2.10), this one here has the advantage that the most significant term in the approximation error does not depend on Δ, but on Δ^2 (which is smaller than Δ when $\Delta < 1$). Otherwise said, (2.13) offers a better approximation for $f'(x)$ than (2.10).

Approximating second order derivatives can be done in a similar manner. One such approximation of second order derivatives by using centered finite differences looks like:

$$f''(x) = \frac{f(x+\Delta) - 2f(x) + f(x-\Delta)}{\Delta^2} - \mathcal{O}\left(\Delta^2\right). \tag{2.14}$$

In order to apply the finite difference method on Equation (2.6), we must first write it in expanded form. Equation (2.6) is just a compact form for the following equation:

$$\nabla^2 p - \frac{1}{c^2} \cdot \left(\frac{\partial^2 p}{\partial t^2} + 2 \cdot \vec{v} \cdot \frac{\partial}{\partial t} \{\nabla p\} + (\vec{v})^2 \cdot \nabla^2 p \right) = 0. \tag{2.15}$$

Further, we must define the domain on which we want to numerically solve this equation. Although a more precise simulation can be obtained by solving Equation (2.15) on a 3D domain, for feasibility reasons concerning the computation time we chose to limit ourselves to 2D domain simulations. That is, we will not simulate the acoustic wave propagation in the entire volume of liquid, but only in a bidimensional section of it. Although this way we lose a little in terms of precision of the results, we win much in terms of computation time (our objective being to make a real-time propagation simulator). However, when simulating wave propagation we are mostly interested in the propagation in the plane defined by the two transducers. Hence, we can replace computations on the entire volume with computations on various bidimensional sections, as shown in Figure 2.2.

Regardless of the shape of the section, we can perform the computations on a 2D matrix of discrete points. We must just take care that the values of the elements located outside the border of the section are not updated in the computation process (but instead they are left equal to 0 during performing computations). We note that we can easily perform this way simulations on bidimensional sections of arbitrary shapes.

Having defined the simulation grid, we can proceed to the meshing of Equation (2.15). First, we explicit the ∇ operator for the 2D case. We obtain:

$$\frac{\partial^2 p}{\partial x^2} + \frac{\partial^2 p}{\partial y^2} - \frac{1}{c^2} \cdot \left(\frac{\partial^2 p}{\partial t^2} + 2 \cdot v_x \cdot \frac{\partial^2 p}{\partial t \partial x} + 2 \cdot v_y \cdot \frac{\partial^2 p}{\partial t \partial y} + v_x^2 \cdot \frac{\partial^2 p}{\partial x^2} + v_y^2 \cdot \frac{\partial^2 p}{\partial y^2} \right) = 0. \tag{2.16}$$

We separate the term that contains the second order time derivative, as this is the term that will allow us to obtain the next state for p (i.e. the acoustic pressure at Δt seconds later in time):

$$\frac{\partial^2 p}{\partial t^2} = (c^2 - v_x^2) \cdot \frac{\partial^2 p}{\partial x^2} + (c^2 - v_y^2) \cdot \frac{\partial^2 p}{\partial y^2} - 2 \cdot v_x \cdot \frac{\partial^2 p}{\partial t \partial x} - 2 \cdot v_y \cdot \frac{\partial^2 p}{\partial t \partial y} \tag{2.17}$$

By using second order centered finite differences for approximating the time derivative as well as the space derivatives (except for the first order time derivatives, for which we use

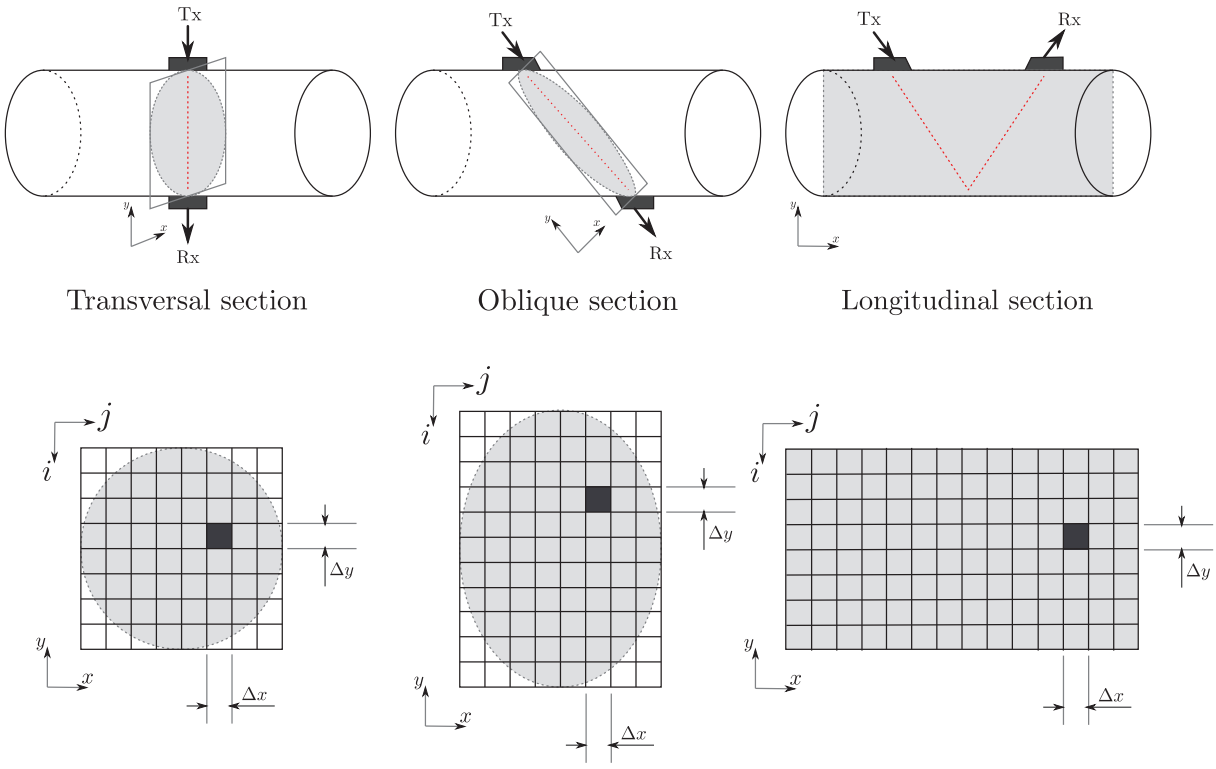


Figure 2.2: Various manners of bidimensionally sectioning the simulation domain, as well as the measuring configurations where they are particularly useful.

backwards differences), we obtain the finite difference form of Equation (2.17), that is:

$$\begin{aligned}\frac{p_{i,j}^{(t+1)} - 2p_{i,j}^{(t)} + p_{i,j}^{(t-1)}}{(\Delta t)^2} &= (c_{i,j}^2 - v_{x_{i,j}}^2) \cdot \frac{p_{i,j+1}^{(t)} - 2p_{i,j}^{(t)} + p_{i,j-1}^{(t)}}{(\Delta x)^2} + \\ &(c_{i,j}^2 - v_{y_{i,j}}^2) \cdot \frac{p_{i-1,j}^{(t)} - 2p_{i,j}^{(t)} + p_{i+1,j}^{(t)}}{(\Delta y)^2} - \\ &2 \cdot v_{x_{i,j}} \cdot \frac{1}{\Delta t} \cdot \left(\frac{p_{i,j+1}^{(t)} - p_{i,j-1}^{(t)}}{2 \cdot \Delta x} - \frac{p_{i,j+1}^{(t-1)} - p_{i,j-1}^{(t-1)}}{2 \cdot \Delta x} \right) - \\ &2 \cdot v_{y_{i,j}} \cdot \frac{1}{\Delta t} \cdot \left(\frac{p_{i-1,j}^{(t)} - p_{i+1,j}^{(t)}}{2 \cdot \Delta y} - \frac{p_{i-1,j}^{(t-1)} - p_{i+1,j}^{(t-1)}}{2 \cdot \Delta y} \right). \end{aligned} \quad (2.18)$$

In this equation Δx and Δy are the spatial meshing steps (that is, the distances between successive points on the meshing grid), and Δt is the temporal meshing step (that is, the distance between two successive simulaton time instants). In order to provide a wide range of simulation features, we considered that c and $\vec{v}$ $(= (v_x, v_y))$ are not constant over the entire simulation domain, but only around meshing grid points. Besides, we notice that the acoustic pressure values from the previous simulation time step must be stored, as they are being used in the equation. The only unknown variable in (2.18) is $p_{i,j}^{(t+1)}$, i.e. the value of the acoustic pressure in the current point at the following simulation time step (that is, Δt seconds later). Solving Equation (2.18) can be done by "simultaneously" computing $p_{i,j}^{(t+1)}$ for every point (i, j) in the spatial grid (except for the points that are located outside the contour of the section, as mentioned earlier).

Before moving to implementation, we first check the consistency of the finite difference equation (2.18) [HC00]. This would ensure us that when the spatial and temporal meshing steps (i.e. Δx, Δy, and Δt) tend to zero the meshed equation tends to its continuous form. In order to do that we first write the Taylor series expansions of the discrete variables in (2.18), around the simulation point (i, j, t). Thus we obtain:

- for the elements that correspond to $\frac{\partial^2 p}{\partial t^2}$:

$$\begin{aligned} p_{i,j}^{(t+1)} &= p_{i,j}^{(t)} + \frac{\partial p}{\partial t} \cdot \Delta t + \frac{1}{2} \cdot \frac{\partial^2 p}{\partial t^2} \cdot (\Delta t)^2 + \mathcal{O}\left((\Delta t)^3\right) \\ p_{i,j}^{(t)} &= p_{i,j}^{(t)} \\ p_{i,j}^{(t-1)} &= p_{i,j}^{(t)} + \frac{\partial p}{\partial t} \cdot (-\Delta t) + \frac{1}{2} \cdot \frac{\partial^2 p}{\partial t^2} \cdot (\Delta t)^2 + \mathcal{O}\left((\Delta t)^3\right) \end{aligned} \tag{2.19}$$

- for the elements that correspond to $\frac{\partial^2 p}{\partial x^2}$:

$$\begin{aligned} p_{i,j+1}^{(t)} &= p_{i,j}^{(t)} + \frac{\partial p}{\partial x} \cdot \Delta x + \frac{1}{2} \cdot \frac{\partial^2 p}{\partial x^2} \cdot (\Delta x)^2 + \mathcal{O}\left((\Delta x)^3\right) \\ p_{i,j}^{(t)} &= p_{i,j}^{(t)} \\ p_{i,j-1}^{(t)} &= p_{i,j}^{(t)} + \frac{\partial p}{\partial x} \cdot (-\Delta x) + \frac{1}{2} \cdot \frac{\partial^2 p}{\partial x^2} \cdot (\Delta x)^2 + \mathcal{O}\left((\Delta x)^3\right) \end{aligned} \tag{2.20}$$

- for the elements that correspond to $\frac{\partial^2 p}{\partial y^2}$: similarly as above.

- for the elements that correspond to $\frac{\partial^2 p}{\partial t \partial x}$:

$$\begin{aligned} p_{i,j+1}^{(t)} &= p_{i,j}^{(t)} + \frac{\partial p}{\partial x} \cdot \Delta x + \frac{1}{2} \cdot \frac{\partial^2 p}{\partial x^2} \cdot (\Delta x)^2 + \mathcal{O}\left((\Delta x)^3\right) \\ p_{i,j-1}^{(t)} &= p_{i,j}^{(t)} + \frac{\partial p}{\partial x} \cdot (-\Delta x) + \frac{1}{2} \cdot \frac{\partial^2 p}{\partial x^2} \cdot (\Delta x)^2 + \mathcal{O}\left((\Delta x)^3\right) \\ p_{i,j+1}^{(t-1)} &= p_{i,j}^{(t-1)} + \left(\frac{\partial p}{\partial x}\right)_{i,j}^{(t-1)} \cdot \Delta x + \frac{1}{2} \cdot \left(\frac{\partial^2 p}{\partial x^2}\right)_{i,j}^{(t-1)} \cdot (\Delta x)^2 + \mathcal{O}\left((\Delta x)^3\right) \\ p_{i,j-1}^{(t-1)} &= p_{i,j}^{(t-1)} + \left(\frac{\partial p}{\partial x}\right)_{i,j}^{(t-1)} \cdot (-\Delta x) + \frac{1}{2} \cdot \left(\frac{\partial^2 p}{\partial x^2}\right)_{i,j}^{(t-1)} \cdot (\Delta x)^2 + \mathcal{O}\left((\Delta x)^3\right) \\ \text{and } \left(\frac{\partial p}{\partial x}\right)_{i,j}^{(t-1)} &= \frac{\partial p}{\partial x} + \frac{\partial^2 p}{\partial t \partial x} \cdot (-\Delta t) + \mathcal{O}\left((\Delta t)^2\right) \end{aligned} \tag{2.21}$$

- for the elements that correspond to $\frac{\partial^2 p}{\partial t \partial y}$: similarly as above.

By replacing all these in Equation (2.18), after some straightforward computations we

obtain:

$$\begin{aligned}\frac{\partial^2 p}{\partial t^2} + \mathcal{O}(\Delta t) &= (c^2 - v_x^2) \cdot \left(\frac{\partial^2 p}{\partial x^2} + \mathcal{O}(\Delta x)\right) + \\ &\quad (c^2 - v_y^2) \cdot \left(\frac{\partial^2 p}{\partial y^2} + \mathcal{O}(\Delta y)\right) - \\ &\quad 2 \cdot v_x \cdot \left(\frac{\partial^2 p}{\partial t \partial x} + \mathcal{O}(\Delta t) + \frac{\mathcal{O}\left((\Delta x)^2\right)}{\Delta t}\right) - \\ &\quad 2 \cdot v_y \cdot \left(\frac{\partial^2 p}{\partial t \partial y} + \mathcal{O}(\Delta t) + \frac{\mathcal{O}\left((\Delta y)^2\right)}{\Delta t}\right). \end{aligned} \tag{2.22}$$

Now it is easy to note that if all the meshing steps (Δx, Δy, and Δz) simultaneously tend to 0, (2.22) tends to (2.17). Therefore, Equation (2.18) is consistent.

Solving of Equation (2.18) can be performed using a coupled map lattice (CML) [Kan93] – a structure that is often used in simulating partial difference equations and is very similar to cellular automata (CA) [CD98]. A cellular automaton is a computational structure that is composed of a certain number of identical cells disposed in a fixed topology. Each cell is characterized by a state (chosen from a finite set of values). All cells update their state in the same time, by using an evolution law that takes into consideration the states of the cells located in the neighbourhood of the current cell. The shape of the neighbourhood is defined by the topology (i.e. spatial localization and interconnections of the cells). All in all, CAs are discrete space and time computational structures that are able to perform paralell local computations. This makes them appropriate for modeling physical phenomena described by finite difference equations. Unlike CAs, though, CMLs have an infinite number of states for every cell (while CAs usually have two), and also they can link the future state of a cell not only to the current states of the neighbours, but also to previous states, depending on the equations to be solved.

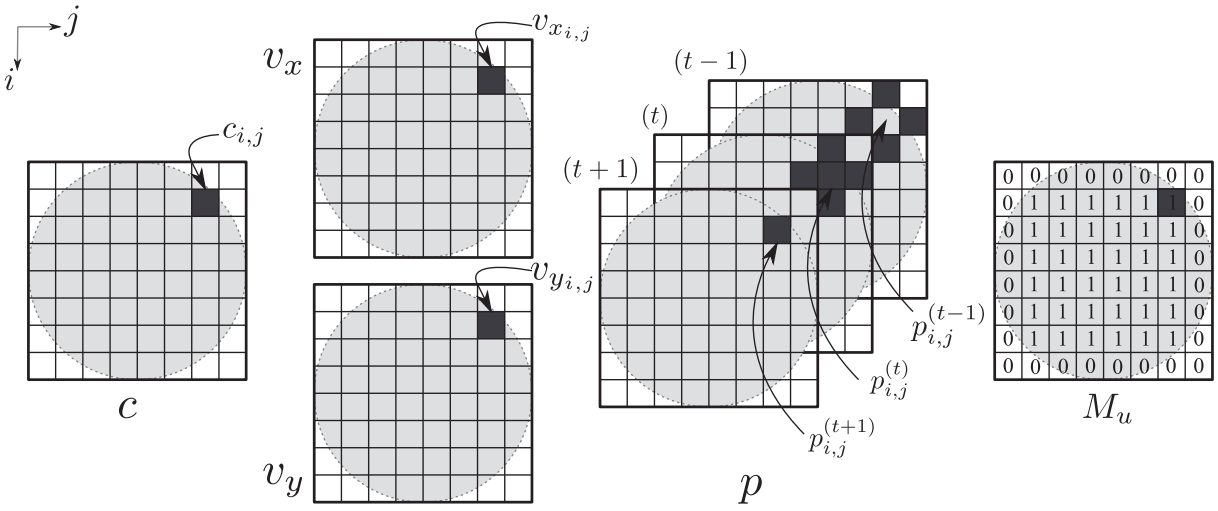


Figure 2.3: The seven matrices involved in numerically solving the acoustic wave propagation equation. We pointed the elements used for computing $p_{i,j}^{(t+1)}$.

The parameters of the liquid are given by three matrices, c, v_x, and v_y, and for computing the *new* value of the acoustic pressure three other matrices are needed: one matrix for storing the current values of the acoustic pressure ($p^{(t)}$), another matrix for

storing the previous values ($p^{(t-1)}$), and another one for storing the values of the acoustic pressure at the next time instant ($p^{(t+1)}$). Not all the elements of these matrices are used in computations, but only those that correspond to the desired section shape. We code the elements whose values update during computing the acoustic pressure values by using another matrix, M_u (*mask of update*) which contains values of 1 in the positions of the elements that correspond to liquid particles that belong to the section (see again Figure 2.2), and values of 0 otherwise. Computing the value of the acoustic pressure in point (i, j) at time instant $t + 1$, i.e. $p_{i,j}^{(t+1)}$, involves elements in all of these matrices, as indicated in Figure 2.3. (In order to reduce the computation time, we implemented these computations as C language functions to be called from Matlab as "*.mex*" files.)

We described in this section the essence of the *HydroSimUS* simulator. In the following sections we discuss about:

- the implementation of the directional emitter
- the computation of the water velocity profiles
- and the constraints and precautions to account for in choosing the simulation parameters

2.2.2 The acoustic transducers

We simulated the emitter by placing the signal sample from the current time instant at the emitter position, on the border located in the immediate vicinity of the liquid. Thus the value of that sample is involved in computing the acoustic pressure value of the adjacent liquid cell, but its value is not updated in the numerical equation solving process.

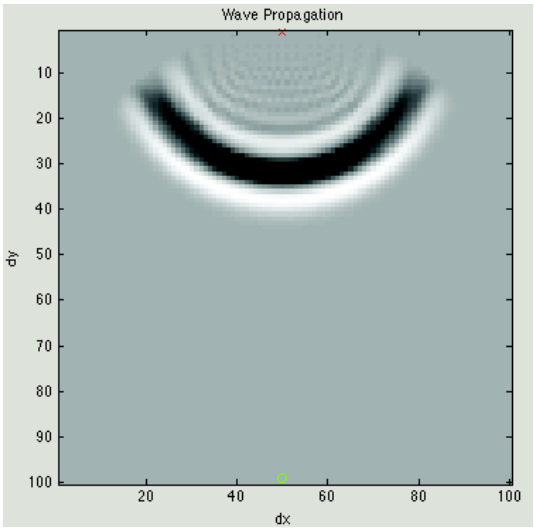


Figure 2.4: The wavefront obtained for an omnidirectional emitter in static water ($v_x = 0$, and $v_y = 0$, and c = constant). The emitted signal is a sinusoidal pulse.

Such an emitter is omnidirectional, as the simulation in Figure 2.4 shows. Actually, the real ultrasonic transducer does not have an omnidirectional directivity diagram, but it emits with maximum power only inside an angle that is inferior to 90°. We can obtain

a quasi-directional emitter by using several omnidirectional emitters as the one described above. Each of them will emit the same signal with the same power in all directions, but with properly chosen time delays. In the points where two or more waves with the same phase superpose, contructive interferences occur. By properly placing the desired interference points, we can favor certain propagation directions, along which the wave will have maximum amplitude (according to the concept of beamforming[3] [VB88, Jae06]). Given that the positions of the elementary emitters are fixed, the positions of these interference points are dictated only by the time delays between the signals emitted by each emitter. Therefore, if we want to emit a signal $s(t)$ in a certain direction, then each elementary emitter will emit a time-shifted attenuated version of $s(t)$, i.e. $\alpha \cdot s(t+d)$. These time delays will be computed according to the desired positions for the points where contructive interference must occur. This calculation is shown below.

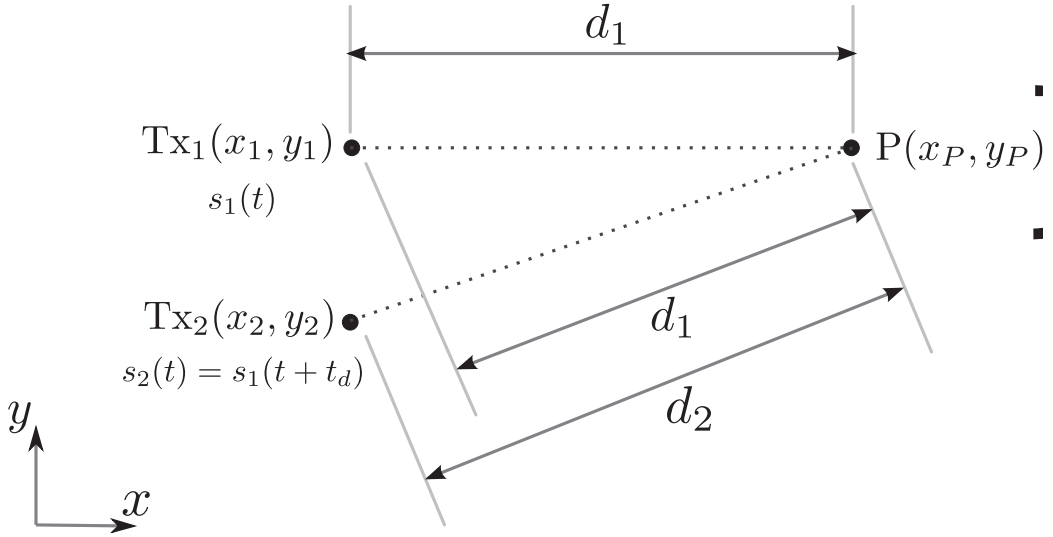


Figure 2.5: Two omnidirectional punctual emitters. The emitted signals are shifted in time so that constructive interference occurs in point P.

Let us have two punctual (omnidirectional) emitters placed at positions (x_1, y_1) and (x_2, y_2) respectively, as indicated in Figure 2.5. The desired propagation direction is in this case the $\mathrm{Tx}_1-\mathrm{P}$ direction. The Tx_2 emitter will contribute to favoring this propagation direction by emitting a similar signal, shifted in time so that signals s_1 and s_2 arrive synchronously in point P (thus leading to the occurence of a contructive interference). As the distance that signal s_2 has to travel to point P (d_2), is longer than the distance that signal s_1 has to travel to that same point (d_1), the signal emitted by Tx_2 must be advanced in time compared to the signal emitted by Tx_1 (as both signals propagate with the same speed, i.e. c). Hence, the time advance t_d must be equal to the time required for the signal to travel the distance $d_1 - d_2$, that is:

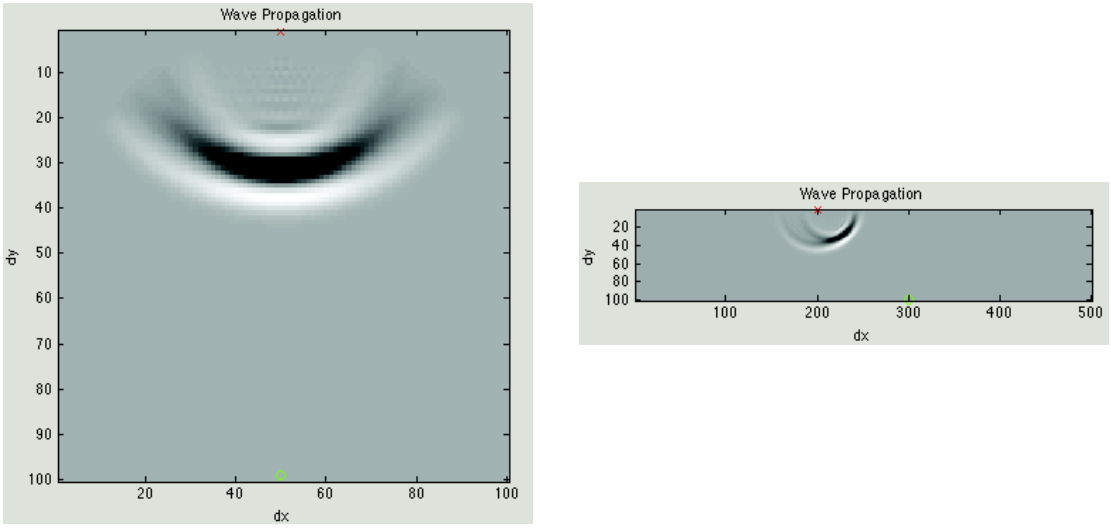
$$t_d = \frac{d_2 - d_1}{c}, \tag{2.23}$$

[3]See also `http://www.spectrumsignal.com/news-events/webcasts/beamforming/`.

where the two distances noted with d are computed with:

$$\begin{aligned} d &= \|(x_P, y_P) - (x, y)\| \\ &= \|(x_P - x, y_P - y)\| \\ &= \sqrt{(x_P - x)^2 + (y_P - y)^2}. \end{aligned} \tag{2.24}$$

(We must bear in mind that in the actual implementation of the simulator we do not work with continuous variables as those in (2.23)) and (2.24), but with discrete variables. Passing from discrete space to continuous space is done by multiplying the discrete variables with the values of the meshing steps, i.e. Δx and Δy. Figure 2.2 clearly illustrates this transition from continuous to discrete (and vice versa) for the spatial variables.)



(a) Transversal section

(b) Longitudinal section

Figure 2.6: The wavefronts obtained in the case of directional emitters formed by the superposition of seven individual omnidirectional emitters that interfere in one point.

Figure 2.6.(a) shows the waveform obtained for the same simulation configuration as the one in Figure 2.4, with the only difference that in this case the signal is not emitted by only one punctual emitter, but by seven such emitters. On each side of the emitter used in Figure 2.4 we additionally placed three such emitters, which emitted the signal with a time advance computed such that contructive interference occured in the reception point. It can be noticed that the resulting wave is focalized along the main propagation direction. Figure 2.6.(b) offers another example, for the case of a longitudinal section of the pipe. The interference point was placed in this case such that the main direction of the emitter be 45 degrees.

2.2.3 Laminar water velocity profile

In the simulation examples that we showed so far (in Figures 2.4 and 2.6) we considered that the velocity of water is equal to 0 in every point in the simulation domain. But the power of Equation (2.18) lies in the fact that it allows simulating wave propagation in a

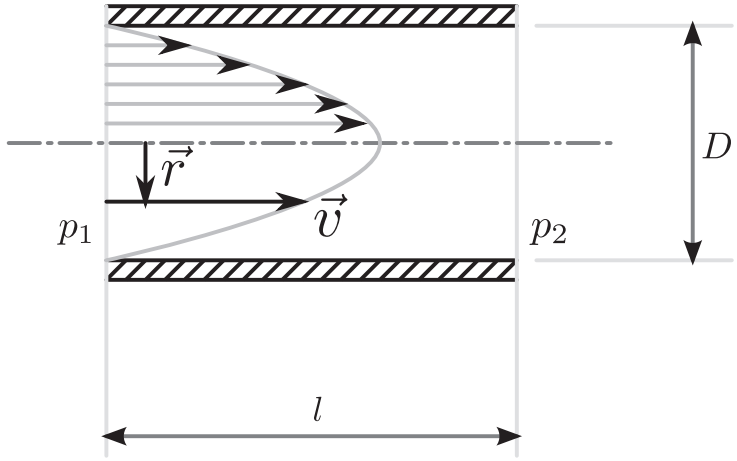


Figure 2.7: Poiseuille velocity profile in a pipe.

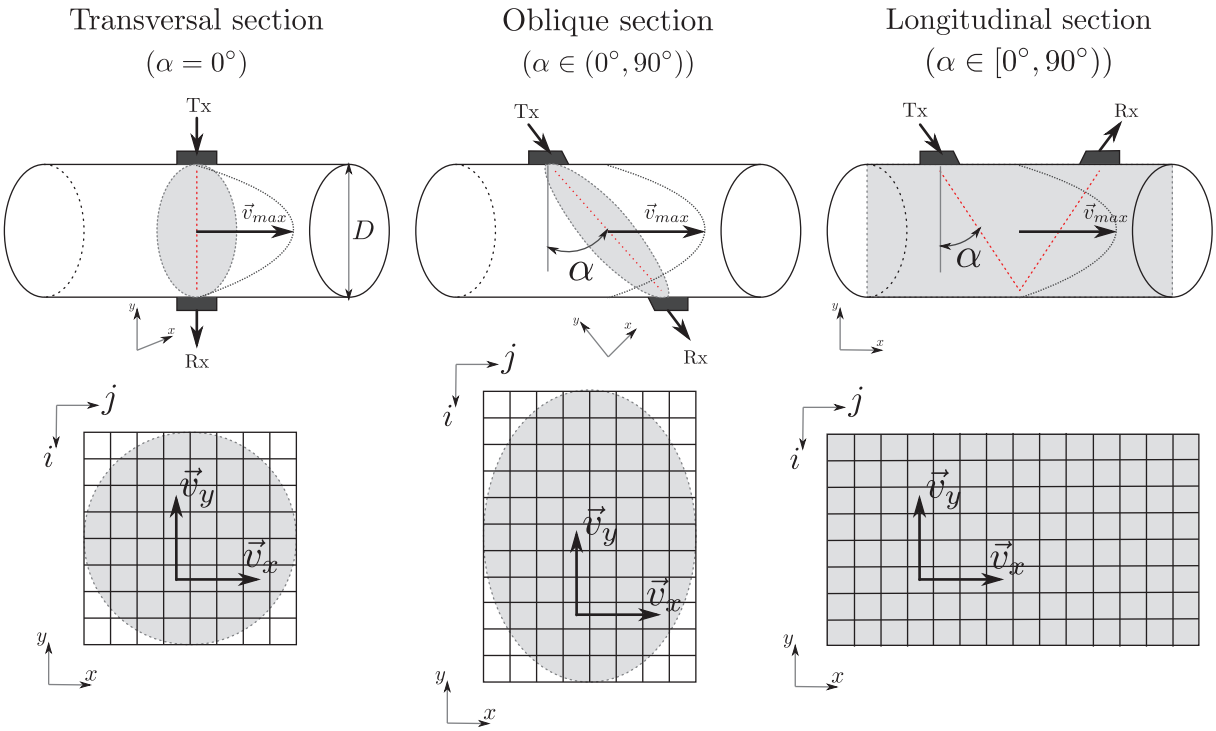


Figure 2.8: Computing the Poiseuille velocity profile for the three main pipe section types.

moving liquid, i.e. when the local water velocities in the discrete points of the simulation domain are nonzero. (Otherwise, i.e. when $v_{x_{i,j}} = 0$ and $v_{y_{i,j}} = 0$, Equation (2.6) (starting from which we obtained the meshed form in (2.18)) becomes Equation (2.2) – the (classic) wave equation.) The simulation software allows, therefore, any velocity profile for liquid – laminar, as well as turbulent.

In particular, in the case of a laminar flow the local water velocities are well modeled

by a Poiseuille profile [Tri77]:

$$v = \frac{G}{4\mu} \cdot \left(\frac{D^2}{4} - r^2 \right), \tag{2.25}$$

where: $G = (p_1 - p_2)/l$, μ is the dynamic viscosity, D is the pipe diameter, and $\vec{r}$ is the distance from the center of the pipe (as shown in Figure 2.7).

From (2.25) it follows that the maximum velocity is obtained for $r = 0$ and it is equal to:

$$v_{max} = \frac{G \cdot D^2}{16 \cdot \mu}. \tag{2.26}$$

Hence, by choosing the maximum velocity and given the pipe dimensions, the local water velocities in every point of the simulation domain can be obtained by using (2.25) and (2.26), as outlined in Figure 2.8.

2.2.4 Turbulent water velocity profile

In order to simulate wave propagation in a turbulent velocity profile, one possible solution would be to simulate a turbulent flow in Fluent, and then import in HydroSimUS (our simulator) the water velocities obtained for the desired section.

However, in order to facilitate the interface with our simulator, we also implemented a bidimensional flow simulator that lets us easily simulate eddies having arbitrary positions and sizes. In this way we removed the difficulties raised by the Fluent-Matlab interface. As we simulate flow only on the 2D section of interest, the computational cost is also drastically reduced.

The software for simulating water flow in a 2D pipe section was obtained by meshing (as we did for Equation (2.6)) the Navier-Stokes equations and then following the methodology described in [Sei08].

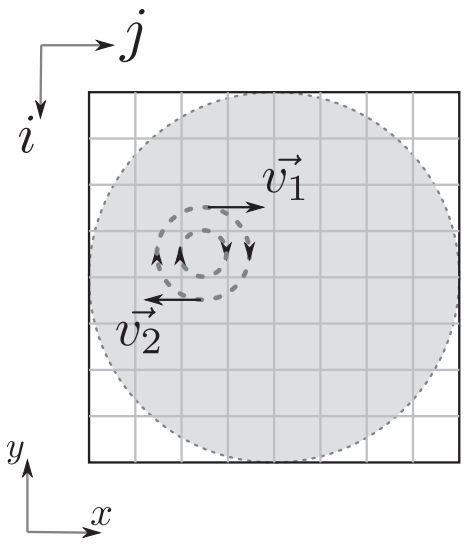


Figure 2.9: Simulation of a 2D eddy can be done by introducing two fixed velocity sources ($\vec{v}_2 = -\vec{v}_1$).

We can simulate an eddy by introducing two fixed velocity sources, as shown in Figure 2.9. Figure 2.10 exemplifies what happens with the wavefront when it passes through such an eddy.

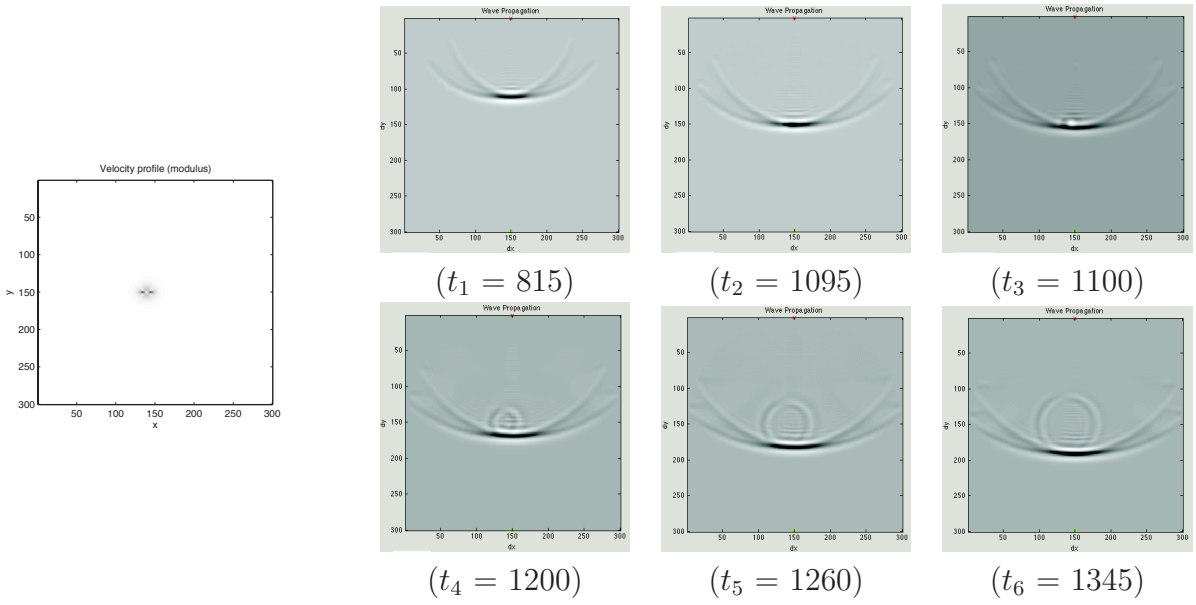


Figure 2.10: Simulation of acoustic wave propagation through an eddy. The modulus of the local water velocity profile is shown on the left side plot. Time instants $t_1, t_2, ..., t_6$ are expressed in number of time steps, Δt.

2.2.5 Choice of the simulation parameters

For the results of the simulation to be realistic, the finite difference equation (2.18) should approximate as good as possible the continuous equation (2.17). Therefore, the meshing steps (Δx, Δy, and Δt) should be as small as possible.

Besides that, we must take into account the frequency of the emitted signal. The results of the simulation are valid if during a wavelength enough discrete points are present in the simulation domain. Otherwise said, we must have:

$$\begin{cases} \Delta x \ll \lambda \\ \Delta y \ll \lambda \end{cases}, \tag{2.27}$$

where: $\lambda = c/f$, with f being the signal frequency.

As for the time step, Δt, it must be at least as small as the desired sampling period for the signal, i.e.:

$$\Delta t \leq T_s. \tag{2.28}$$

Therefore, it would seem that in order to perform a valid simulation it would be sufficient to choose Δx, Δy and Δt as small as possible. However, we should also take into account the fact that the displacement of the liquid particles in a time step must not significantly alter the positions of the grid points, that is:

$$\begin{cases} v_x \cdot \Delta t \ll \Delta x \\ v_y \cdot \Delta t \ll \Delta y \end{cases}. \tag{2.29}$$

Besides this, a valid simulation would require that the distance travelled by the wave in a time step is smaller than the space step, that is:

$$\begin{cases} c \cdot \Delta t < \Delta x \\ c \cdot \Delta t < \Delta y \end{cases} . \tag{2.30}$$

Therefore, some generic guidelines for choosing the meshing steps would be the following:

- choose Δx and Δy as large as possible (in order to have a small number of discrete points in the simulation domain, which means smaller matrices in the simulation software), but keep in mind that they should be much smaller than the wavelength of the emitted acoustic signal.
- and then choose Δt as large as possible (in order to have a small number of iterations in the simulation software), but not larger than the desired sampling period for the signal and also keeping in mind that the distance travelled by the acoustic wave in a time step, i.e. $c \cdot \Delta t$ should be smaller than Δx and Δy.

More details about the implementation and use of our simulator for acoustic wave propagation in water pipes can be found in the Appendix.

2.2.6 Comparison between simulated and experimental signals

Finally, in order to check the validity of HydroSimUS we consider the test case illustrated in Figure 2.11. We emitted a 2 MHz sinusoidal signal containing 11 cycles in a pipe filled with water, and we received it on the other side of the pipe. The received signal is shown in Figure 2.12(a), while Figure 2.12(b) shows the signal obtained by simulation. The simulation was performed on the diagonal section marked in Figure 2.11, by using space meshing steps (Δx and Δy) of 0.05 mm and a time meshing step (Δt) of 0.01 μs.

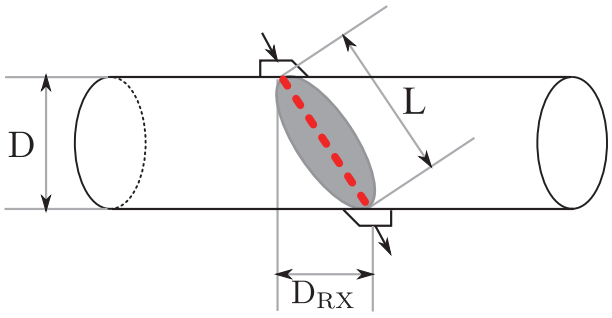
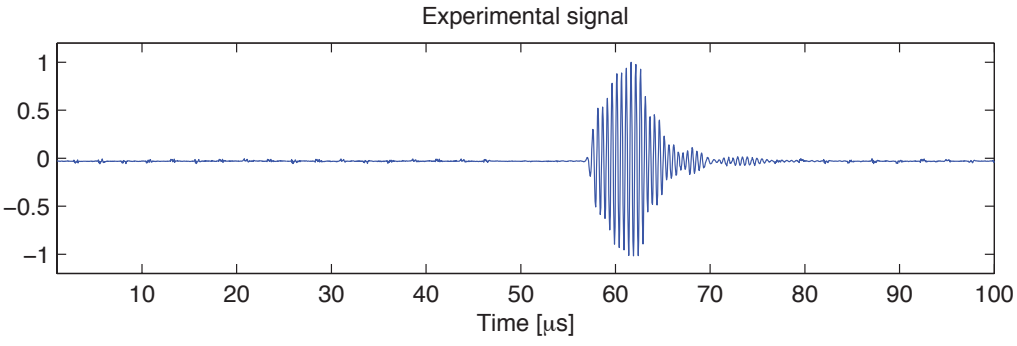
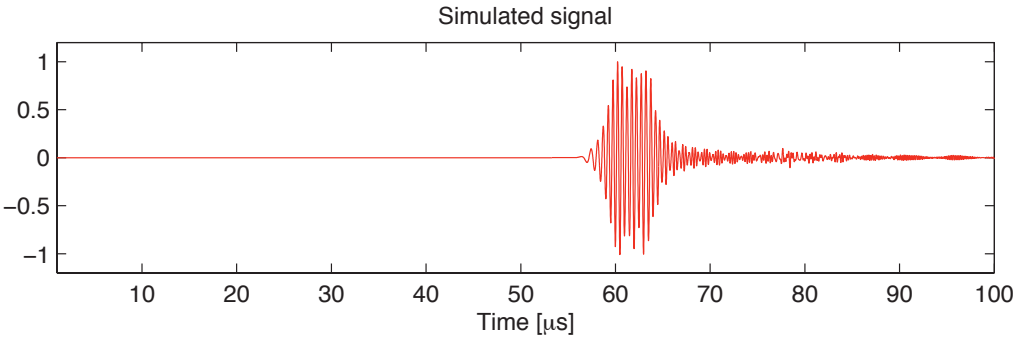


Figure 2.11: Schematic description of the test case we used for comparing an experimental measurement with a software simulation. The pipe has a diameter (D) of 5 cm, and the horizontal distance between the two transducers (D_{RX}) is 3.5 cm.

The amplitude spectra of the two signals (experimental, and simulated) are shown in Figure 2.13. While both Figures 2.12 and 2.13 show that the two signals are similar in the time domain as well as in the frequency domain, they also show some differences –

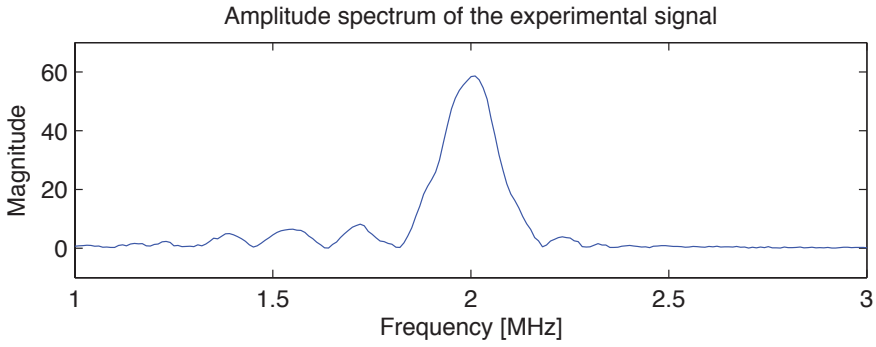


(a)

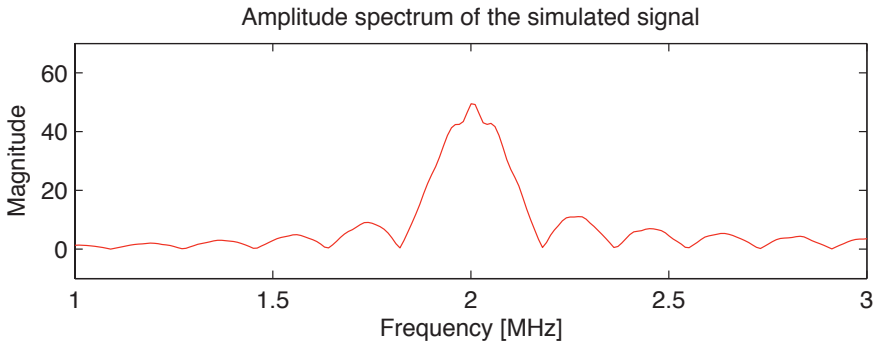


(b)

Figure 2.12: Comparative illustration of an experimental signal and a simulated signal obtained for a similar configuration. (a) The measured signal. (b) The simulated signal.



(a)



(b)

Figure 2.13: Comparative illustration of the amplitude spectra of the two signals – (a) the measured signal, and (b) the simulated signal.

which are mainly due to the approximations we did when simulating the emitter and the attenuation at the water-wall interface.

A more comprehensive experimental validation of HydroSimUS can be found in Chapter 5, where we also use it for simulating different configurations for ultrasonic based estimation of water velocity.

2.3 Summary

Simulating the propagation of ultrasounds in flowing liquids is very useful when it comes to developing ultrasonic intrumentation, as performing experimentations on real life platforms is both expensive and time consuming. We showed in this chapter that this task is not straightforward, and we proposed a method for solving it. The resulted software application, *HydroSimUS*, allows us to perform offline "measurements" in various configurations. Some of these configurations are practically impossible to be studied experimentally – as is the case of acoustic wave propagation through a single and very well localized eddy.

Further, we will use HydroSimUS for obtaining various "realistic" test signals. This will later give a more practical taste to the tasks of transient signal detection and characterization, which are the subject of the following two chapters.

Chapter 3

Analysis of transient signals using vector samples and recurrences

The debatable lack of elegance of nonlinear methods often conceals their power and robustness when it comes to performing different signal processing tasks. From this realm of nonlinearity comes the concept of recurrence, which is the main refrain of the current chapter. We consider the generalized meaning of this concept and we show that some special cases of it are in fact more than common in classic signal processing. We show thus that RPA is an unifying framework for various traditional signal processing tools, by also providing generalizations as well as the possibility of building completely new tools.

3.1 Preliminaries

Transient signals having a short time span are very common in signal processing applications. Telecommunications, reflectometry, ultrasonic testing, biomedical engineering, radar, sonar, as well as the study of marine mammals are only some of the applications where transient signals play a key role. There is no exact definition of what transient signals are, but the common feature of all these different signals is their short duration. In [RA01] transients are defined as "short time impulsive signals". As noted in [RA98], they are characterized by a short duration compared to the observation duration, they have unknown waveform and unknown arrival time.

There are certain applications where models can be used to describe transients, but we are interested in a generic approach. Hence, we call transient any signal having a finite (and short) time support. Estimating this time support is the main concern in transient signal analysis. This can be put as a problem of signal detection at every time instant, which results into a time detection curve.

As we are mostly interested in the generic case where nothing is known a priori about the transients to be detected, we consider only nonparametric approaches. Nonparametric statistic approaches generally start from the assumption that noise is white, Gaussian and stationnary, while transients are not. Therefore, they usually detect transient signals either based on the deviation from Gaussianity of their probability density function (with the aid of χ^2 tests or higher order statistics [LAC97]), or by checking signal stationnar-

ity [RFA+10]. On the other hand, time-frequency [PS02, Sta02, OSA11] and time-scale [Mal09, IQ03] approaches are based on the assumption that transients usually have an oscillatory waveform. They detect them based on the spectral differences between transients and noise. As mentioned in the first chapter, another existing solution consists in combining these two approaches, in order to take advantage on the benefits of them both (as the authors of [RA98] did, by combining higher order statistics with time-scale analysis). Each of these classes of solutions has its limitations, especially due to the fact that in practice noise may differ from its theoretical model. For instance, it may not have a Gaussian distribution and a flat power spectral density. Besides that, complex deterministic signals may have similar properties to noise, making it even more difficult to discriminate between the two of them.

In addition to detecting the time support of transients, we might also want to obtain robust and effective estimates of some of their parameters, such as the duration of the first rising edge, or the fundamental frequency.

Although initially developed for the analysis of dynamical systems, recurrence plot analysis (RPA) later turned into a generic tool for nonlinear time series analysis [MCRTK07]. Some applications in signal detection were reported in the last fifteen years. They showed that RPA based statistics are able to detect deterministic signals buried in noise [ZGWJ98, ZGWJ00, MNB08, RNDB08, MK09], as well as to estimate the fundamental frequency of signals [Ter02]. Recurrence plots are also able to distinguish deterministic chaotic signals from random signals, as well as to estimate dynamical invariants [MCRTK07].

Despite these studies, the use of RPA in signal processing is yet far from becoming a common practice. However, in this chapter we point out two important concepts that RPA brings into signal processing – *vector samples* and *recurrence* – and then we involve them in developing tools for performing different transient signal processing tasks.

3.2 Recurrence plot analysis

In the current section we present the methodology of RPA. We start with an overview of the steps involved in recurrence plot analysis, by highlighting the concepts of *vector sample* and *recurrence*, which are of particular interest in signal processing. (In the following subsections we return to the steps of RPA by providing more in-depth discussion about phase space representations, recurrence plots, and recurrence quantification analysis (RQA).) We discuss the idea of representing signals as trajectories in phase space from a signal processing point of view, we discuss the concept of (generalized) recurrence, and we discuss as well the relationship between RQA and traditional signal processing. We close the current section by mentioning some variations of recurrence plots, as well as some of their uses currently found in literature.

3.2.1 Overview

Recurrence plot analysis is based on the concept of recurrence, which signifies the return of the state of a system close to a previously visited point. If we plot the evolution of

the system state as a trajectory, a recurrence is obtain each time this trajectory nearly intersects itself. The study of the recurrence pattern (obtained by checking the recurrence between each pair of points) may reveal useful information about the analyzed trajectory. The method of time-delay embedding provides the link to signal processing, by allowing us to turn a monodimensional signal into a multidimensional trajectory – which can later be analyzed with the aid of recurrences.

There are three main milestones in the developement of RPA: time-delay embedding (1980-1981) [PCFS80, Tak81], recurrence plot (1987) [EOKR87], and recurrence quantification analysis (RQA) (1992-2002) [ZWJ92, WJZ94, MK02]. More details about the developement of RPA can be found in [Mar08]. The authors of [MCRTK07] greatly contributed to the popularization of this method, which found applications in various fields involving the (nonlinear) analysis of time series. Among others, different signal processing applications can be found in literature, e.g. speech processing, denoising, and signal detection.

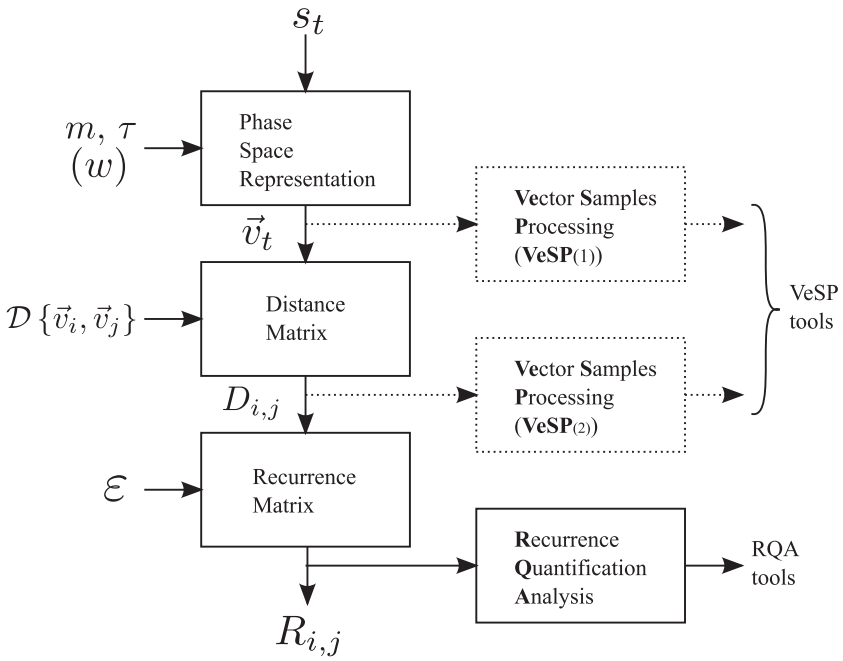


Figure 3.1: Summary of recurrence plot analysis (RPA). Extensions of the classic methodology are shown in dotted line.

Figure 3.1 summarizes the steps of the RPA methodology. The classic *flow* is the one marked by bold arrows in the figure. The analyzed signal, s, is first turned into a trajectory, $\vec{v}$, by forming vectors from groups of samples. In classical time-delay embedding, phase space vectors are constructed as:

$$\vec{v}_t = \left(s_t, s_{t+\tau}, s_{t+2\tau}, ..., s_{t+(m-1)\tau}\right), \tag{3.1}$$

where m is the embedding dimension (i.e. the dimension of the phase space), τ is the time delay between successive vector components, and by s_x we note the value of signal s at time x. Figure 3.2 illustrates this. However, we slightly modify (3.1) in order for vector $\vec{v}_t$ to more precisely characterize signal s around time instant t:

$$\vec{v}_t = \left(s_{t-\lfloor\frac{w-1}{2}\rfloor}, s_{t-\lfloor\frac{w-1}{2}\rfloor+\tau}, ..., s_{t-\lfloor\frac{w-1}{2}\rfloor+(m-1)\tau}\right), \tag{3.2}$$

where $w = 1 + (m-1)\tau$. We call $\vec{v}_t$ the *vector sample of signal s at instant t*. We note that $\vec{v}_t$ in (3.2) does not contain all signal samples between time instants $t - \lfloor \frac{w-1}{2} \rfloor$ and $t - \lfloor \frac{w-1}{2} \rfloor + (w-1)$, but only some of them. Hence, we may call it a *sparse* vector sample (because it only contains some (and, hopefully, the most significat) of the signal samples in the neighbourhood of s_t). On the other hand, a (full) vector sample is obtained by setting τ equal to 1 (which means that it will contain all the signal samples loated in a range of size w around the time instant t).

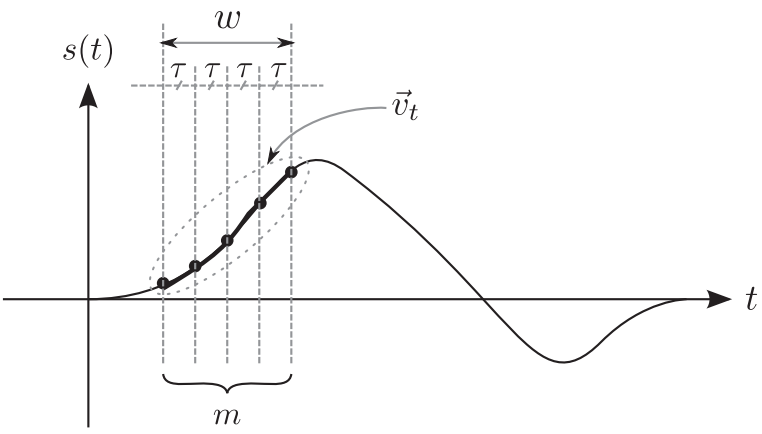


Figure 3.2: Illustration of phase space vectors building in RPA. The m components of a phase space vector span a time interval of length w.

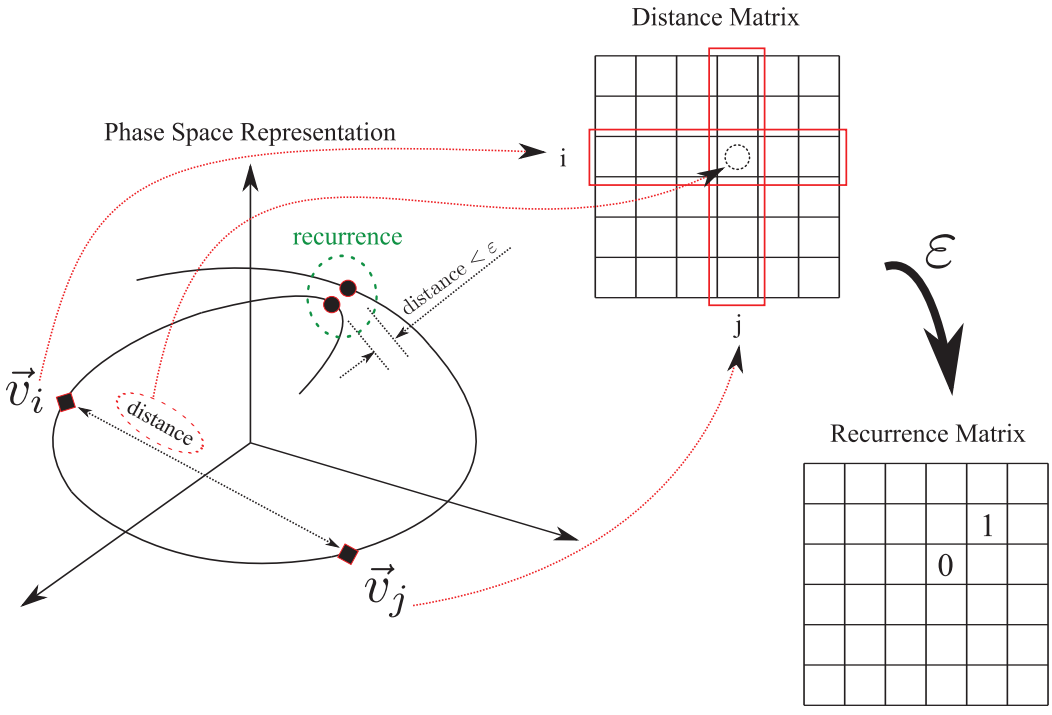


Figure 3.3: Computation of the recurrence plot (RP) from a phase space trajectory.

The following step in classic RPA is the computation of the distance matrix. It usually contains in the element located at row i and column j the distance between vectors $\vec{v}_i$ and $\vec{v}_j$:

$$D_{i,j} = \|\vec{v}_j - \vec{v}_i\|\,, \tag{3.3}$$

where $\|\cdot\|$ is most often the Euclidean metric. (The use of other metrics is reported in literature as well [MCRTK07].) In order to make things more generic, we define the distance matrix as:

$$D_{i,j} = \mathcal{D}\left\{\vec{v}_i, \vec{v}_j\right\}, \tag{3.4}$$

where $\mathcal{D}$ is not necessary a metric, but any operator that produces a scalar from two vectors. We call $\mathcal{D}\left\{\vec{v}_i, \vec{v}_j\right\}$ *the distance between vectors* $\vec{v}_i$ *and* $\vec{v}_j$ (although this may not always be its real meaning). In Section 3.3 we discuss three such new "distances".

The third step in RPA is the computation of the recurrence matrix (see Figure 3.3). It is obtained by (hard) thresholding the distance matrix D with a threshold ε:

$$R_{i,j} = \Theta\left(\varepsilon - D_{i,j}\right). \tag{3.5}$$

Here Θ denotes the Heaviside step function. It produces a value of 1 for positive arguments and a value of 0 for negative arguments. The threshold ε is called recurrence threshold. (In the original definition [EOKR87] ε was not a constant, but in common practice it is used as such.) If the distance between vectors $\vec{v}_i$ and $\vec{v}_j$ is less than or equal to the recurrence threshold, then we have a recurrence, i.e. a value of 1 at $R_{i,j}$. "Recurrence plot" (RP) is the name used for the graphical representation of the recurrence matrix (by using black for 1 and white for 0).

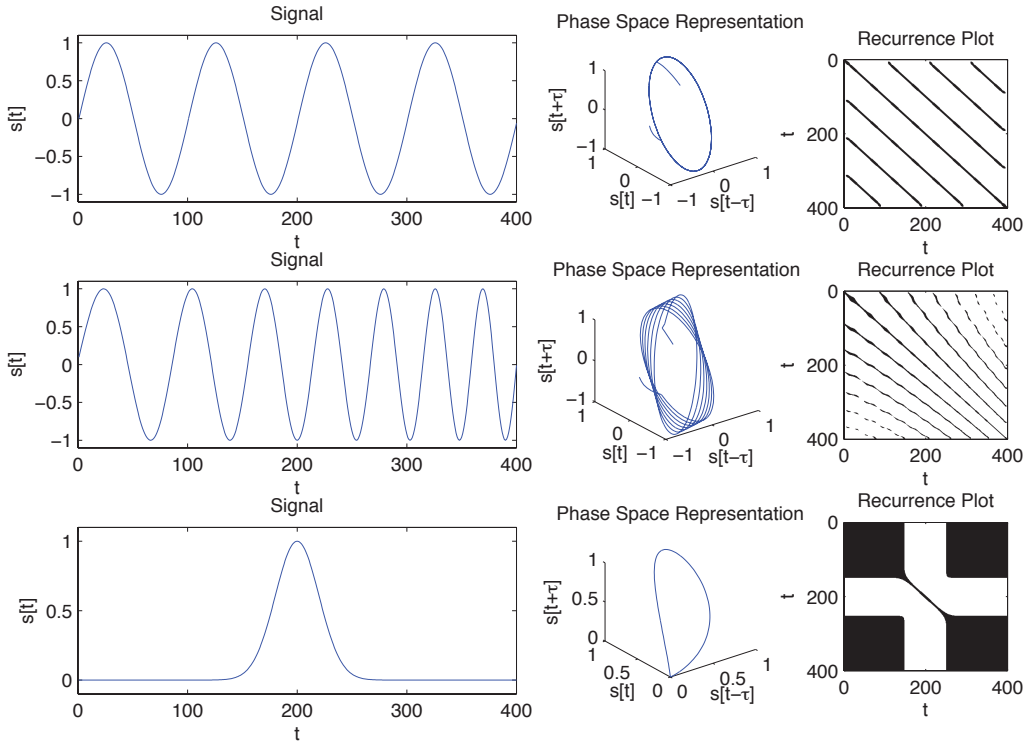


Figure 3.4: Examples of typical signals and their corresponding recurrence plots. (top) Sinusoidal signal. (middle) Chirp signal. (bottom) Gaussian pulse.

Figure 3.4 shows some examples of phase spaces trajectories and recurrence plots computed for some test signals. Some other examples of RPs, as well as discussion on the meaning of the different patterns can be found in [MCRTK07]. We just mention that impulsive signals generally lead to light bands in the recurrence plots, and that periodic

signals lead to diagonal lines that are parallel to the main diagonal (as it can be noticed from Figure 3.4).

Finally, the last step in RPA is the computation of the recurrence quantification analysis (RQA) measures on the recurrence plot. Several types of such measures exist in literature. They either quantify black point density, diagonal lines, or vertical lines [MCRTK07]. For instance, the most common RQA measure, the recurrence rate (RR) is defined as:

$$RR = \frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} R_{i,j}, \tag{3.6}$$

where R is the recurrence matrix and $N \times N$ is the size of R.

The classical RPA path presented here might as well be *shortcut* as shown (with dotted line) in Figure 3.1. As there is no recurrence involved, but only vector samples processing (VeSP), we called these alternatives $\text{VeSP}_{(1)}$ and $\text{VeSP}_{(2)}$.

$\text{VeSP}_{(1)}$ involves reformulating operations with signal samples in terms of vector samples. This may be useful in generalizing classical signal processing tools. For instance, we generalize in this way the correlation coefficient (see Section 3.4).

$\text{VeSP}_{(2)}$ consists in performing a quantification of the distance matrix (instead of quantifying the recurrence matrix, as in classical RQA). For instance, the energy of signal s may be computed in terms of vector samples by adding the elements on the main diagonal of the distance matrix D (computed with the distance between vector samples $\vec{v}_i$ and $\vec{v}_j$ defined as their dot product). This VeSP version of the energy of a signal is detailed in Section 3.4.

Some of the classical RQA measures already proved their utility in signal detection [ZGWJ98, ZGWJ00, MNB08]. In [MK09] the author additionally proposed in this context the use of RQA measures computed on variations of the RP (i.e. cross recurrence plot, and joint recurrence plot). We propose in Section 3.5 a new RQA measure, which allows us to build a transient signal detector based on the analysis of the time distribution of the phase space vectors histogram. We also propose methods for noise reduction (see Section 4.2.2), fundamental frequency estimation (see Section 4.2.3), and slow-rate chirp identification (see Section 4.2.4).

Next, we get into some details related to aspects only briefly discussed in this overview, namely time-delay embedding, recurrence plots, and recurrence quantification.

3.2.2 Phase space representation and vector samples

Time-delay embedding

We have shown what time-delay embedding is in the previous subsection. In an alternative formulation, the phase space trajectory may look like this [SSB11]:

$$\vec{r}(t) = \sum_{k=1}^{m} s\left(t + (k-1)\tau\right) \cdot \vec{e_k}, \tag{3.7}$$

where $s(t)$ is the signal, $\vec{r}(t)$ denotes the phase space trajectory, and $\vec{e}_k$ are versors indicating the phase space axes.

The idea of building trajectories from successive samples of a signal was supported by Takens' theorem, which states that a dynamical system attractor can be reconstructed from a series of (scalar) observations of the state of the dynamical system, embedded in a dimension that is greater than twice the size of the attractor. More detailed information can be found in [PCFS80, Tak81, SYC91, CEFG91, Huk06].

The conditions of Takens' theorem are not satisfied by real world signals, which are usually not noise-free and do not have infinite resolution. However, it is remarcable that although they are just a series of monodimensional observations, signals may hide inside them information about the global dynamics of the (multidimensional) system that produced them. The difficulty in revealing this information through time-delay embedding consists in choosing proper values for the m and τ parameters.

Various techniques for choosing the m and τ embedding parameters can be found in literature [Sma05], such as false nearest neighbours or successive embeddings – for the embedding dimension, and autocorrelation or mutual information – for the embedding delay. However, there is no universal *best choice* for m and τ, but it depends on the specific application.

Next, we discuss the relationship between τ and the time scale of the analyzed signal. This discussion is motivated by the necessity to analyse the signal at different scales, which is useful in practical applications (in particular, related to the propagation of transient signals in liquids).

Time scaling

If $s(t)$ is the signal and $\vec{r}(t)$ is its corresponding phase space trajectory (using time-delay embedding with parameters m and τ), the time-scaled version of it is $s'(t) = s(\beta \cdot t)$, where β is a real number. And the corresponding phase space trajectory will be:

$$\vec{r'}(t) = \sum_{k=1}^{m} s\left(\beta t + (k-1)\beta\tau'\right) \cdot \vec{e_k}. \tag{3.8}$$

We notice that scaling the time axis of the signal leads to an appropriate scaling of the time delay (the τ parameter). For the trajectory to remain unchanged, τ' should be chosen such that:

$$\tau' = \frac{\tau}{\beta}. \tag{3.9}$$

Figure 3.5 shows that the trajectories indeed remain unchanged when τ is properly scaled.

However, Equation (3.8) shows that $\vec{r'}(t)$ is not identical to $\vec{r}(t)$ even if (3.9) is valid. Figure 3.5 does not reflect the time evolution, and therefore $\vec{r'}(t)$ and $\vec{r}(t)$ seem identical. In fact, trajectory $\vec{r}(t)$ is "drawn" β times faster (or slower, depending on whether β is greater than or less than 1) than trajectory $\vec{r'}(t)$. This leads to the contraction/dilatation of the structures in the recurrence plot, as shown in Figure 3.6.

Failure to meet the constraint in Equation (3.9), though, leads to significant changes in the trajectory – redundance (when $\beta < 1$) or irrelevance (when $\beta > 1$) [Mar03]. It

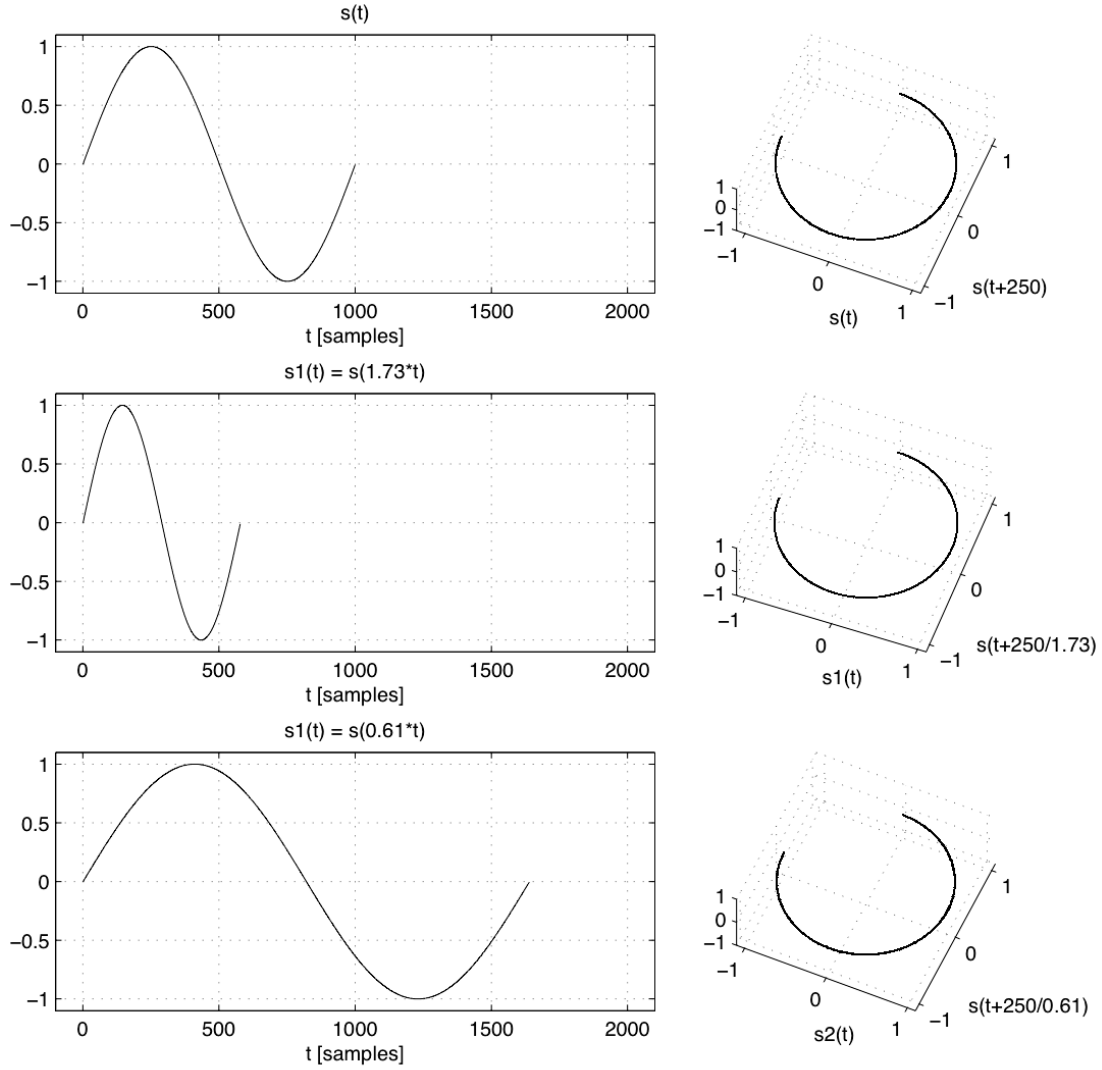


Figure 3.5: Time-scaled one-period sinusoidal signal and the corresponding trajectories. From top to bottom, β is: 1, 1.73, 0.61, and τ is: 250, 250/1.73, 250/0.61. The shape of the trajectory remains unchanged, although it is "drawn" with different speeds.

follows that the recurrence patterns will also be modified. However, when dealing with dilated signals, i.e. when $\beta < 1$, the trajectory does not change its main characteristics, except that it is "attracted" to the first diagonal of the coordinate system. (In this case the changes in the recurrence pattern are mostly due to the use of recurrence balls that have the same dimensions as those used for the non-dilated signal.)

Figure 3.5 would suggest that the optimum value for τ is the value for which the trajectory is as wide as possible (or as round as possible, in the case of a sinusoidal signal). However, a τ that is optimum for all signals is impossible to find. For instance, let us consider a chirp having its frequency linearly varying from f_0 to f_1. Hence, its period varies linearly from T_0 to T_1. If we chose for τ the optimum value corresponding to

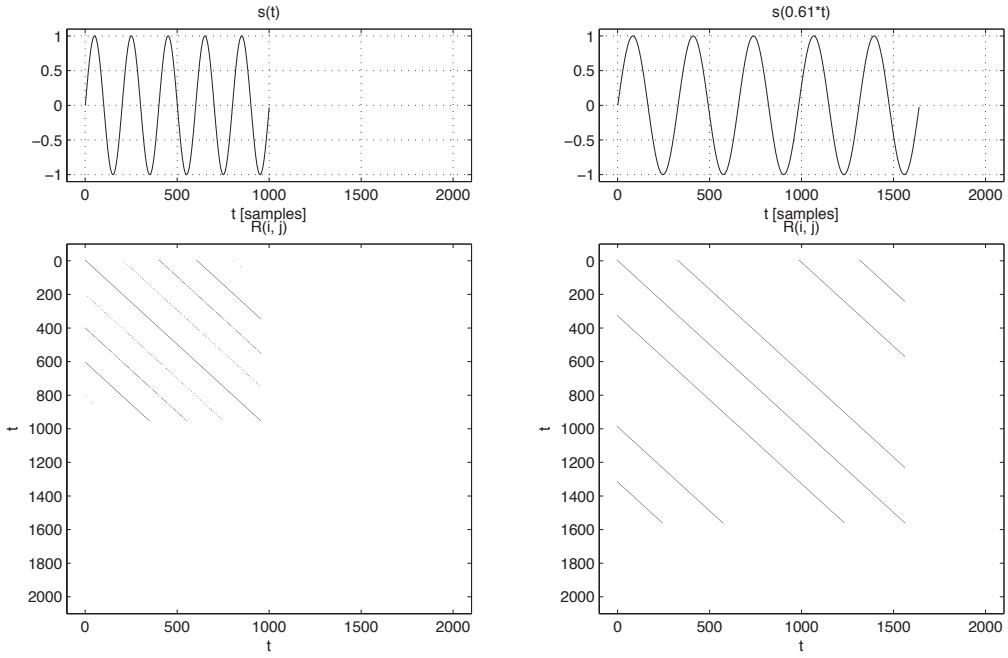


Figure 3.6: The effect of dilatation on the recurrence plot, for a sinusoidal signal. (left) The recurrence plot of signal $s(t)$ is obtained using $m = 2$ and $\tau = 50$. (right) The recurrence plot of signal $s(0.61 \cdot t)$ is obtained using $m = 2$ and $\tau' = 82$ $(= \tau/0.61)$. As τ is scaled appropriately, the pattern remains unchanged, except for the fact that it is stretched.

period T_0, then this τ is not optimum any more as the instantaneous period advances to T_1. This clearly shows that trying to find a τ that is *optimum* for the entire signal leads only to a rough approximation. The most desirable it would be to have a τ that varies for the duration of the analyzed signal, in order to adapt to its local features. This would allow us to investigate only those scales that are relevant for the analyzed signal.

Use of vector samples in signal processing

As discussed in Section 3.2.1, we propose to call the phase space vectors *vector samples* and take them centered on the current time instant. We can also abandon the use of m and τ, by using only the quantity $w = 1 + (m - 1) \cdot \tau$ instead. We call it the size of the vector samples, or the vector sampling window. We obtain thus a signal representation in which every sample additionally contains memory and prediction. It is easy to notice that vector sampling, while being a special case of time-delay embedding, is basically equivalent to multiplying the signal with a rectangular window of unit value and size w, whose time center is slided over each signal sample.

Filtering signals by using such a sliding window is not a new thing in signal processing. What is new about the concept of *vector sample* is that it allows the development of operations that are usually performed on samples. Clasically, two signal samples are usually only added or multiplied. But if we think in terms of vector samples of size w, we see that the range of options greatly extends. For instance, in order to compare the

similarity of two vector samples, we can compute the distance between them – and we can define this distance in many ways.

In addition to the flexibility they bring in defining the distance between them, operating with vector samples promises to be more noise robust than operating with classic samples, as for every time instant we consider not only the signal value at that time instant, but its values in an entire neighbourhood. We elaborate on this topic in Section 3.4.

But RPA is not only about vector samples. It is a methodology that is centered, as its name implies, on the recurrence plot, which returns to stage in the following subsection.

3.2.3 Recurrence quantification analysis

The recurrence plot was introduced in [EOKR87] as a tool for computing dynamical system parameters from time series, and it was defined as a square binary matrix that contains ones for the pairs of points that are close to each other on the trajectory, and zeros for the rest. As its authors state, the recurrence plot "*describes natural (but subtle) time correlation information*". They also mention the large-scale typologies as well as the small-scale textures that appear in the recurrence plots and can provide (visual) information about the underlying dynamical system.

Analyzing recurrence plots visually is not very useful, as it involves human intervention (as well as a trained eye). Recurrence quantification analysis (RQA) appeared as an attempt to perform computationally the interpretation of recurrence plots [ZWJ92, WJZ94]. A typical example of using the classical RQA measure of the recurrence rate (i.e. black point density) is shown in Figure 3.7.

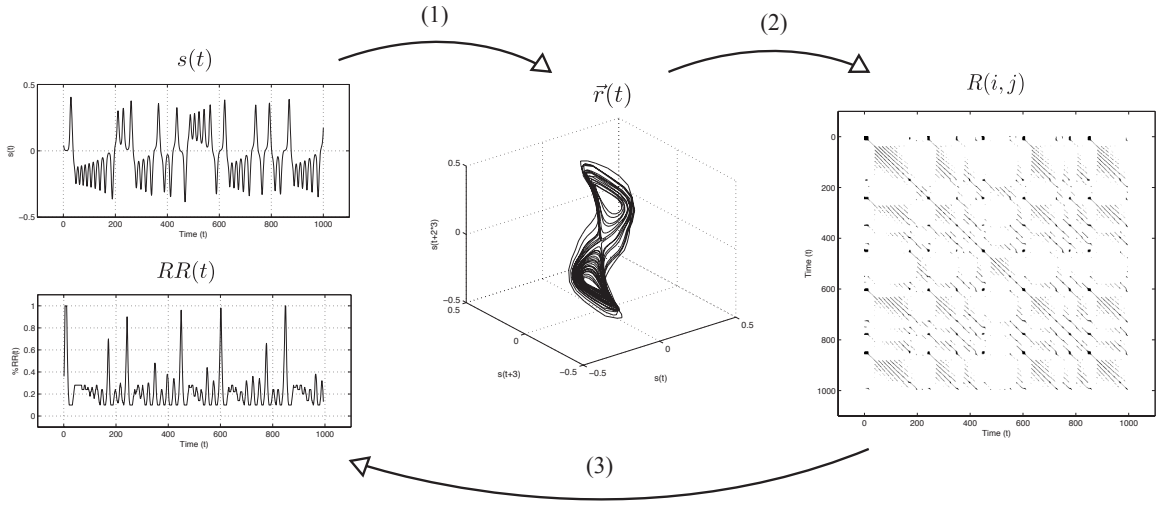


Figure 3.7: Sample use of RQA: (1) phase space representation of the signal; (2) computation of the recurrence plot; (3) quantification of the recurrence plot. In this example, signal $s(t)$ is obtained from a chaotical Lorenz system. The trajectory $\vec{r}(t)$ is obtained using time-delay embedding with a dimension $m = 3$ and a delay $\tau = 3$. It can be noticed that $\vec{r}(t)$ has strong similarities with the well-known Lorenz "butterfly" attractor. The transitions of the system from one unstable periodic orbit to the other can be identified using the significant maxima of the locally-computed (by using a window of size 10) recurrence rate, $RR(t)$.

Although the recurrence rate is the most common RQA measure, other quantification methods have been developped. The traditional RQA measures can be grouped into the following three classes[1] [MCRTK07]:

- RQA based on quantifying (black) point density in the recurrence plot. This class contains in fact only one measure, that is the recurrence rate (RR) that we already mentioned. It computes the black point density of the RP by counting the number of black points (i.e. the number of ones in the recurrence matrix), and then dividing it by the total number of points in the RP (i.e. N^2, where N is the size of the recurrence matrix).

- RQA based on quantifying diagonal lines. We mention determinism, average diagonal line length, maximum diagonal line length, divergence, entropy, trend, and ratio.

- RQA based on quantifying horizontal(/vertical) lines. The measures in this class were introduced by [Mar03] and are laminarity, trapping time, and maximum vertical line length.

Different methods related to recurrence plots can be found in literature, such as cross recurrence plots, joint recurrence plots, perpendicular recurrence plots, isodirectional recurrence plots, order patterns recurrence plots, isometric recurrence plots, etc. Besides these, there are as well different other methods that use similar principles to recurrence plots, such as close return plots, contact maps, dot plots, first return maps, order matrices, similarity plots, etc. Initially developed as a dynamical systems analysis tool, recurrence plots have been since then used in a variety of fields. To name just a few, we mention: physiology, geology, molecular biology, speech recognition, astrophysics, financial data analysis, etc. More detailed information about the different methods related to recurrence plots, as well as about the uses of recurrence plots in various fields can be found in [MCRTK07], as well as in the programmes of the four editions of the international symposium dedicated to recurrence plots[2], which is organized every two years starting in 2005.

3.3 New distances in vector samples processing

We are used to operate with signal samples. We are used, for example, to add, substract, and multiply them. By rethinking all these operations in terms of vector samples, we get to the notion of *distance*. This opens the way to a very wide range of new operations that can be performed on the signal. In the current section we discuss three such new operations: the angular distance, the dynamic range distance, and the dot product distance. These contributions will serve to define new detection/characterization methods, some of which will be detailed in the next section.

[1] More information about the classic RQA measures, as well as about variations of recurrence plots and related methods can be found online at `http://recurrence-plot.tk/`.

[2] See `http://recurrence-plot.tk/workshop.php`.

3.3.1 Angular distance

As the phase space representation of the signal is a linear operation, scaling the signal amplitude will result in an appropriate scaling of the trajectory. If for signal $s(t)$ we have trajectory $\vec{r}(t)$, then for signal $s'(t) = \alpha \cdot s(t)$ we have:

$$\vec{r'}(t) = \sum_{k=1}^{m} \alpha \cdot s\left(t + (k-1)\tau\right) \cdot \vec{e_k} = \alpha \cdot \vec{r}(t). \tag{3.10}$$

Let us see whether this scaling of the trajectory has any effect on the recurrence plot. If the recurrence plot corresponding to signal $s(t)$ is $R(i,j)$, then the recurrence plot corresponding to signal $s'(t)$ will be:

$$R'(i,j) = \Theta\left(\varepsilon'(i) - \mathcal{D}\left\{\alpha \cdot \vec{r}(i), \alpha \cdot \vec{r}(j)\right\}\right). \tag{3.11}$$

(Notice that here we considered the generic case where ε is not a constant.)

If $\mathcal{D}$ is the Euclidean metric, then in order for $R'(i,j)$ to be identical to $R(i,j)$, $\varepsilon'(i)$ should be chosen such that:

$$\varepsilon'(i) = \alpha \cdot \varepsilon(i). \tag{3.12}$$

However, this scaling of the recurrence radius is no more needed when using an "angular distance", defined as:

$$\mathcal{D}\left\{\vec{r}(i), \vec{r}(j)\right\} = \arccos\left(\frac{\vec{r}(i) \cdot \vec{r}(j)}{\|\vec{r}(i)\| \cdot \|\vec{r}(j)\|}\right), \tag{3.13}$$

for non-zero vectors (otherwise vectors can be considered paralell, i.e. the angular distance between them is 0). This definition for $\mathcal{D}$ leads to a recurrence plot that is invariant to amplitude scalings of the analyzed signal. We note that this angular distance generates conical neighbourhoods (instead of the spherical ones generated by the Euclidean distance). Figure 3.8 illustrates this observation. We also note that by using such a conical neighbourhood the concept of recurrence loses its initial meaning (that is returning close to a previously visited point) – therefore, we are actually working with a generalized recurrence.

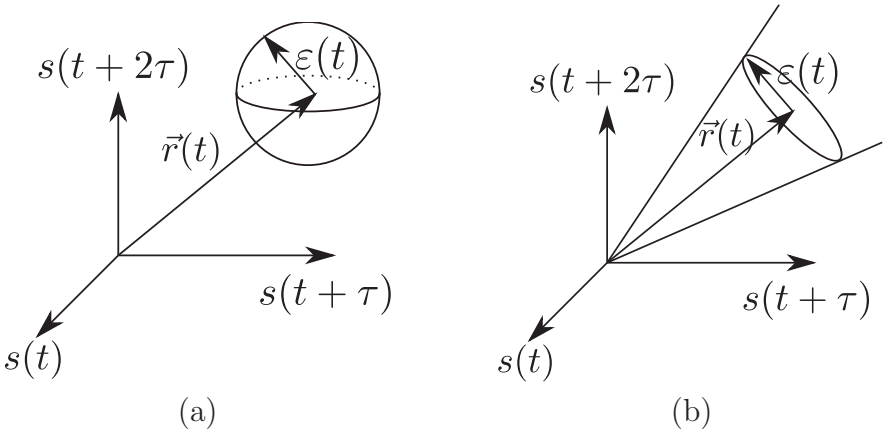


Figure 3.8: The shape of the neighbourhood (in a tridimensional phase space) is a sphere for the Euclidean distance (a) and a cone for the angular distance (b).

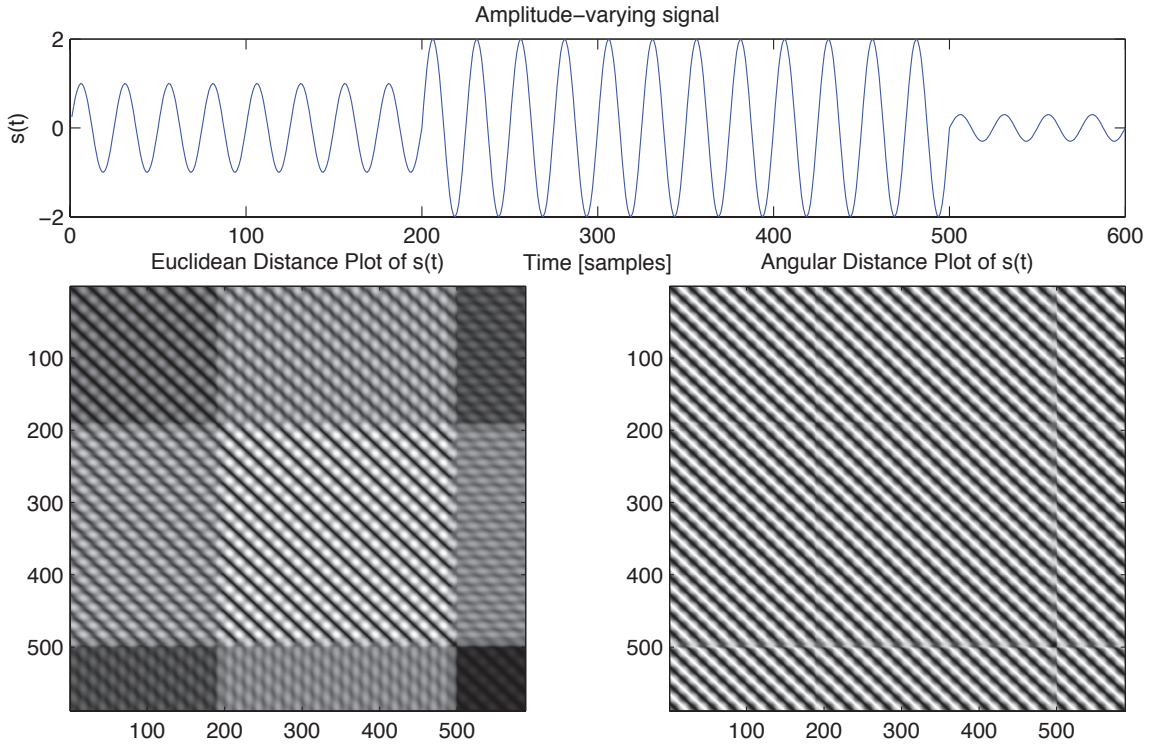


Figure 3.9: Euclidean distance plot and angular distance plot for an amplitude-varying signal. ($m = 3$, $\tau = 6$)

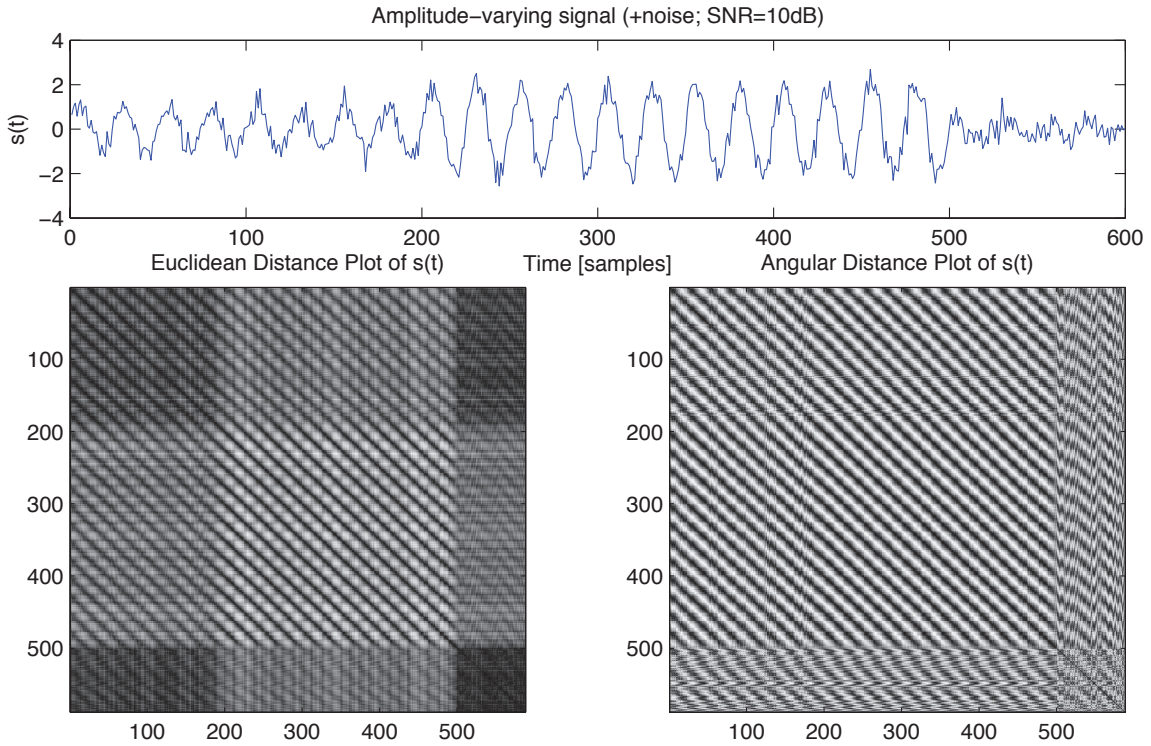


Figure 3.10: Euclidean distance plot and angular distance plot for an noisy amplitude-varying signal. (SNR = 10 dB, $m = 3$, $\tau = 6$)

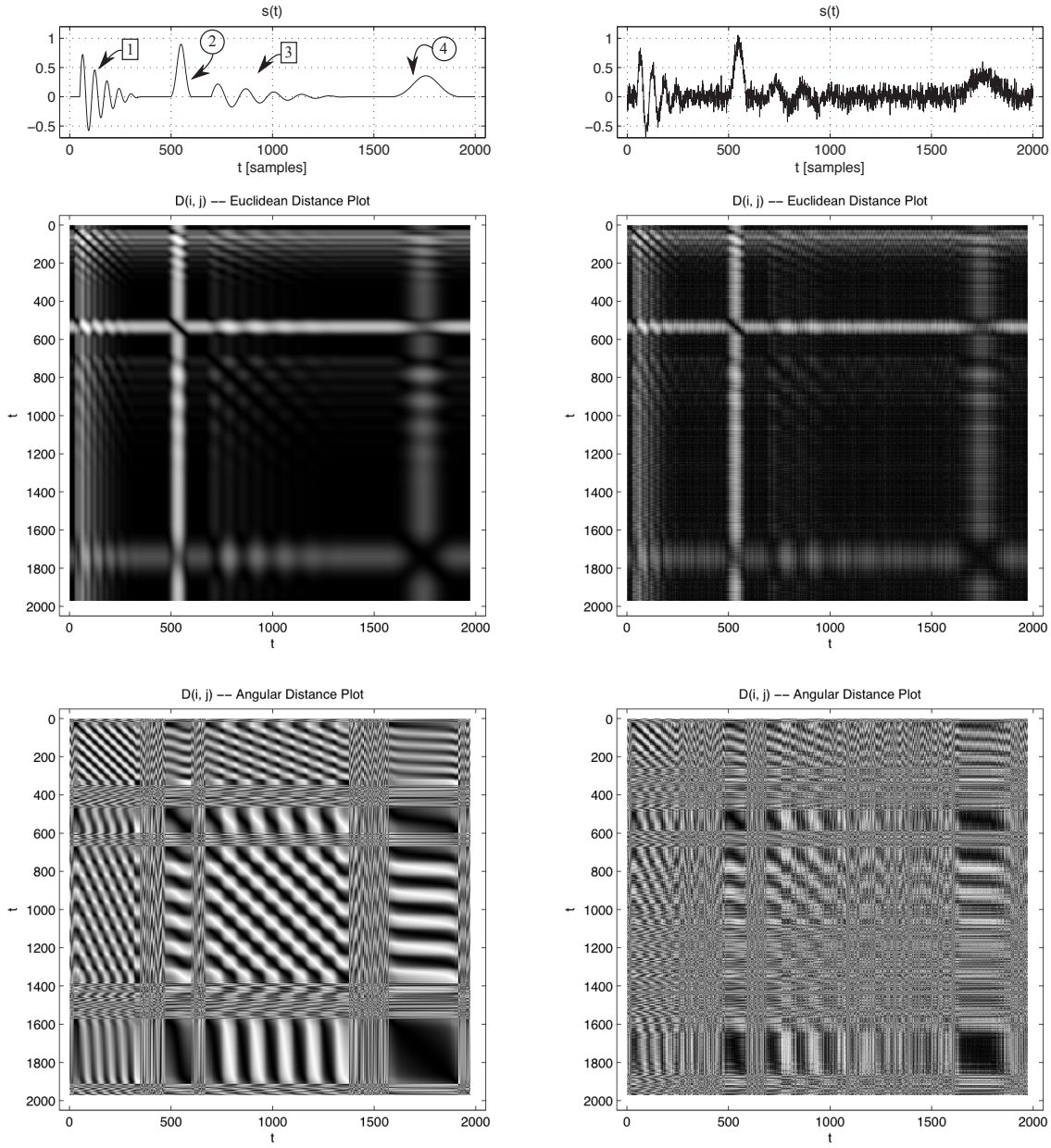


Figure 3.11: EDP and ADP for a synthetic signal composed of four transients coming from two different sources. It is computed using $m = 3$ and $\tau = 15$. The four transients can be easily localized in time and characterized as well by a visual analysis of this image. (a) The SNR is 60 dB. (b) The SNR is 5 dB. Noise has a strong impact on the resulted ADP image, but time localization and characterization of the four transients can still be done to some extent.

As the recurrence matrix is nothing but a thresholded version of the distance matrix, we will use the latter for illustrating the effect of the angular distance. Figure 3.9 shows both the Euclidean distance plot (EDP) and the angular distance plot (ADP) for an amplitude-varying sinusoidal signal. It is obvious from the figure that the ADP is almost invariant to the amplitude variation of the original signal. The ADP of the signal is almost identical to the EDP that would have been obtained if the sinusoid had not been

amplitude modulated. It can also be noticed in this case that the ADP keeps the textures in the EDP unchanged, except for a difference in "brightness" and "contrast". However, when adding noise to the analyzed signal things change, as Figure 3.10 shows. Even if the global SNR is constant (i.e. 10 dB), it is locally different in different areas of the signal – the larger the signal amplitude, the higher the SNR. This difference in the SNR during the signal leads to an ADP whose texture is no longer constant, but is more or less affected by noise, according to the local amplitude of the signal.

Figure 3.11 shows another example of an angular distance plot computed for a synthetic signal. The signal is composed of four transients coming from two different sources – the first two of them come from direct propagation between the sources and the receiver, and the last two come from reflexions (i.e. they are attenuated and dilated). As the figure shows, the angular distance plot of this signal allows a good time localization of the four transients, regardless of their scale differences. More than that, it allows a visual characterization of the transients. It can be easily observed (by analyzing the blocks on the main diagonal, that correspond to the four transients) that we are dealing with two types of signals (i.e. two different sources). It can also be easily observed that the first transient is of the same type as the third, and that the second is of the same type as the fourth. We can make this statement either as a result of a comparative visual analysis of their corresponding blocks in the angular distance plot, or as a result of the fact that their cross-term blocks are composed of diagonally oriented structures (e.g. the third rectangular block on the first line of $D(i,j)$ in Figure 3.11 is the cross-term block between the first and the third transients; all the black structures in it are oriented along the first diagonal of the block).

These observations (together with those we made in Section 3.2.2, regarding time scale) may constitute a good starting point for developing a robust automatic method for detecting and classifying transient signals having different scales.

3.3.2 Dynamic range distance

The same as for the Euclidean distance, in order to compute the dynamic range (DR) distance between two vector samples, $\vec{v}_i$ and $\vec{v}_j$, we must first compute the difference vector:

$$\vec{v}^{(-)} = \vec{v}_j - \vec{v}_i. \tag{3.14}$$

Then, we compute the *norm* of this vector, which in the case of the DR distance is defined as:

$$\mathcal{D}\left\{\vec{v}_i, \vec{v}_j\right\} = \max_k \left\{\vec{v}^{(-)}(k)\right\} - \min_k \left\{\vec{v}^{(-)}(k)\right\}, \tag{3.15}$$

where $\vec{v}^{(-)}(k)$ denotes the k-th component of vector $\vec{v}^{(-)}$.

To show the utility of this distance, let us consider the signal s in Figure 3.12(a). If we resample this signal using vector samples of size w, we obtain $\vec{s}^{(w)}$. Then, if for each vector sample we compute its distance from the origin (i.e. from vector $\vec{0}$), that is:

$$DR0[n] = \max_k \left\{\vec{s}_n^{(w)}(k)\right\} - \min_k \left\{\vec{s}_n^{(w)}(k)\right\}, \tag{3.16}$$

we obtain the signal in Figure 3.12(b). It can be noticed that the effect of this operation is somewhat similar to the effect of a (first order) derivative. Nevertheless, we do not

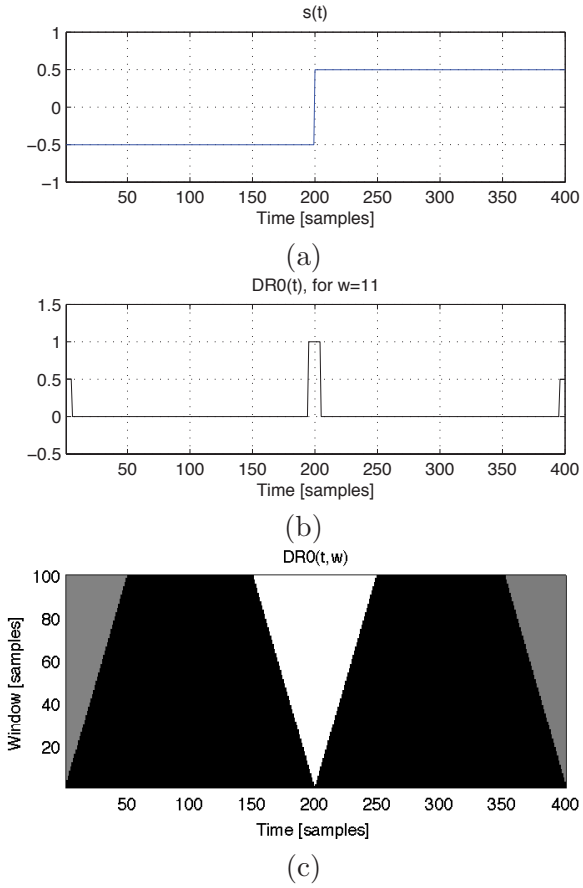


Figure 3.12: A step signal, and the dynamic range "norm" computed for each of its vector samples (of size $w = 11$). (a) The signal. (b) The DR norm of vector samples. (c) A t-w representation of the DR norm.

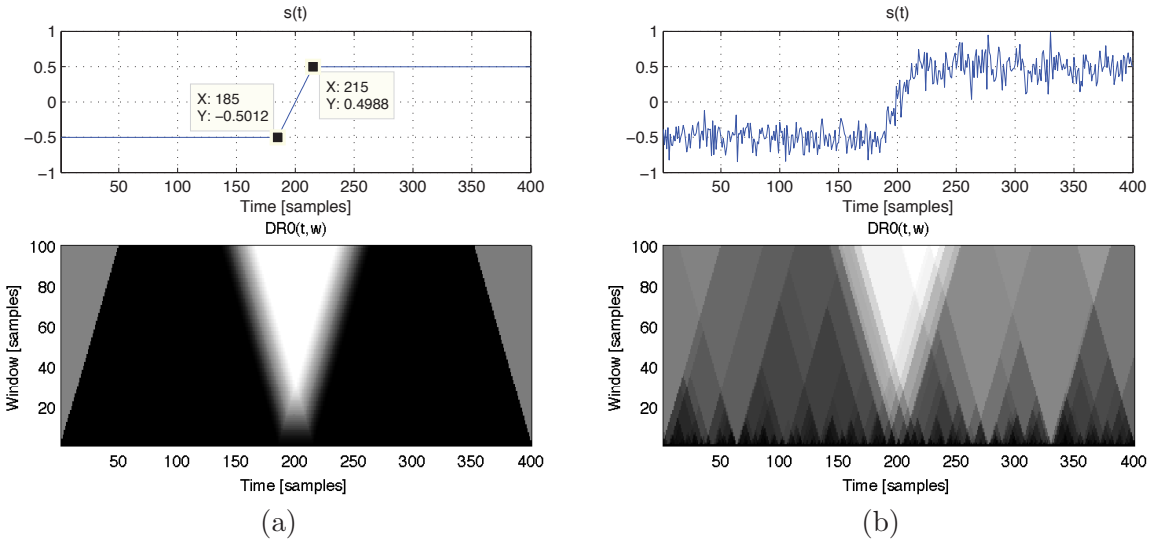


Figure 3.13: A signal containing two stable states and a linear transition between them, and the dynamic range norm computed for each of its vector samples, in a t-w representation. The length of the plateau is minimal for values of w around 30, i.e. the duration of the ramp. (a) No noise case. (b) SNR = 10 dB case.

obtain only a peak indicating the position of the rising edge, but a plateau lasting for w samples. The dependance of $DR0$ on w is illustrated globally in Figure 3.12(c), where every horizontal line in the image corresponds to a fixed value of w.

However, if the transition between the two stable values in the signal is not sudden, but is done linearly during a time T, things look like in Figure 3.13. We notice that, (the same as for Figure 3.12) the sharpest peak is obtained for that value of w that is equal to T.

The dynamic range distance will be later used for defining the VeSP derivative of a signal (see Section 3.4.4).

3.3.3 Dot product distance

If we think about how the correlation compares signals, we see that it relies on multiplication between signal samples. We recall that the cross-correlation of signals x and y is computed as:

$$X[n] = \sum_{i=1}^{N} x_i \cdot y_{i+n}. \tag{3.17}$$

When the two signals are similar, their corresponding samples will have similar values (of the same sign). Hence, by summing these products a large value will be obtained. Instead, if the corresponding samples of the two signals are decorrelated, there will also be a significant number of pairs of samples having opposite signs – and therefore the sum of all the sample products will have a smaller value. Consequently, *correlation* measures how similar signals are, i.e. *how close* they are, while *distance* measures *how far* they are. Hence, these two notions are complementary – the more similar two signals are, i.e. the larger the similarity, the smaller the distance between them is, and vice versa.

Having in mind that a scalar is a particular case of vector and that correlation measures similarity (which can be seen, as discussed, as a (reversed) *distance*) with the aid of multiplications, we can define a *distance* between vector samples by using the dot product, as follows:

$$\mathcal{D}\left\{\vec{v}_i, \vec{v}_j\right\} = \vec{v}_i \cdot \vec{v}_j, \tag{3.18}$$

where $\vec{v}_i$ and $\vec{v}_j$ are vector samples of size w.

The dot product distance will be later used for defining the VeSP correlation coefficient (see Section 3.4.1) and the RQA autocorrelation function (see Section 3.5.1).

3.4 Vector samples based signal processing measures

In the current section we define VeSP tools that generalize well-known traditional signal processing tools. First, we address two extreme cases of transient signal detection – signal known completely, and signal completely unknown – by developing VeSP based generalizations for the correlation coefficient and for the signal energy. Then, we develop a VeSP based technique for computing the envelope of a signal, as well as a noise robust generalization of the derivative of a signal. The section closes with a VeSP generalization of the average magnitude difference function (AMDF).

3.4.1 VeSP Correlation

The correlation coefficient of signals s and r is computed as:

$$C = \frac{1}{N}\sum_{i=1}^{N}\left(s_i - \langle s\rangle\right)\cdot\left(r_i - \langle r\rangle\right), \tag{3.19}$$

where $\langle x\rangle$ denotes the mean value of signal x, and N is the length of the two signals. It provides a measure of the similarity between the two signals.

By replacing signals s and r in (3.19) with their vector sampled versions (see Equation (3.2)), we obtain the vector samples based generalization of the correlation coefficient:

$$C^{(w)} = \frac{1}{N}\sum_{i=1}^{N}\left(\vec{s}_i^{(w)} - \langle\vec{s}^{(w)}\rangle\right)\cdot\left(\vec{r}_i^{(w)} - \langle\vec{r}^{(w)}\rangle\right). \tag{3.20}$$

In this equation $\vec{x}^{(w)}$ is the vector samples version of signal x using a vector sample size of w (i.e. $\vec{x}_i^{(w)} = \left(x_{i-\lfloor(w-1)/2\rfloor}, x_{i-\lfloor(w-1)/2\rfloor+1}, ..., x_{i-\lfloor(w-1)/2\rfloor+(w-1)}\right)$), and the product of samples is replaced by a dot product of vector samples. (Note that the signal is zero padded in order to obtain the same number of vector samples as the number of samples in the original signal.) The quantity $\langle\vec{x}^{(w)}\rangle$ is the average of the vectors in $\vec{x}^{(w)}$ (i.e. $\langle\vec{x}^{(w)}\rangle = \frac{1}{N}\sum_{i=1}^{N}\vec{x}_i^{(w)}$).

It is easy to note that for $w = 1$ Equations (3.20) and (3.19) are equivalent, i.e. $C^{(1)} \equiv C$. Hence, (3.20) is a generalization of (3.19). The natural question is whether this generalization is useful or not. In order to answer to this question we consider the test signal in Figure 3.14. It has a total length of 100 samples and it contains a 10 samples one-period sinusoidal transient. If we know the waveform of the transient, we can search for its occurence in the analyzed signal by computing the correlation coefficient between the analyzed signal and the reference signal at every time instant. (The signals are appropriately zero-padded in order to obtain N values for the correlation coefficient.) Then we take the position of the maximum.

Figures 3.14(c) and 3.14(d) show the results of such actions for a signal-to-noise ratio (SNR) varying from -15 dB to -5 dB with a step of 0.25 dB. Details about the way we computed the SNR can be found in the Appendix. For each SNR a number of 201 iterations were performed. Figure 3.14(c) displays the median values of the positions of the maximum of correlation, while Figure 3.14(d) displays their median absolute deviations. It can be noticed from these figures that we obtain better results by using the VeSP correlation than by using classic correlation. The value we used for the vector sample size is $w = 10$. (We recall that the median value of a set of values is obtained by sorting the values in ascending order and then taking the value located in the middle position. We also recall that the median absolute deviation of a set of values is computed as the median value of the absolute values of the differences between each element and the median value of all elements. We could have computed instead the mean value and the standard deviation, but they are less robust to outliers.) In Figure 3.14(b) we show how the *peakedness* of the VeSP correlation coefficient depends on the value of w (in the SNR $\to \infty$ case). We compute this peakedness as the difference between two maxima – the first one is computed on the time support of the transient, while the second is

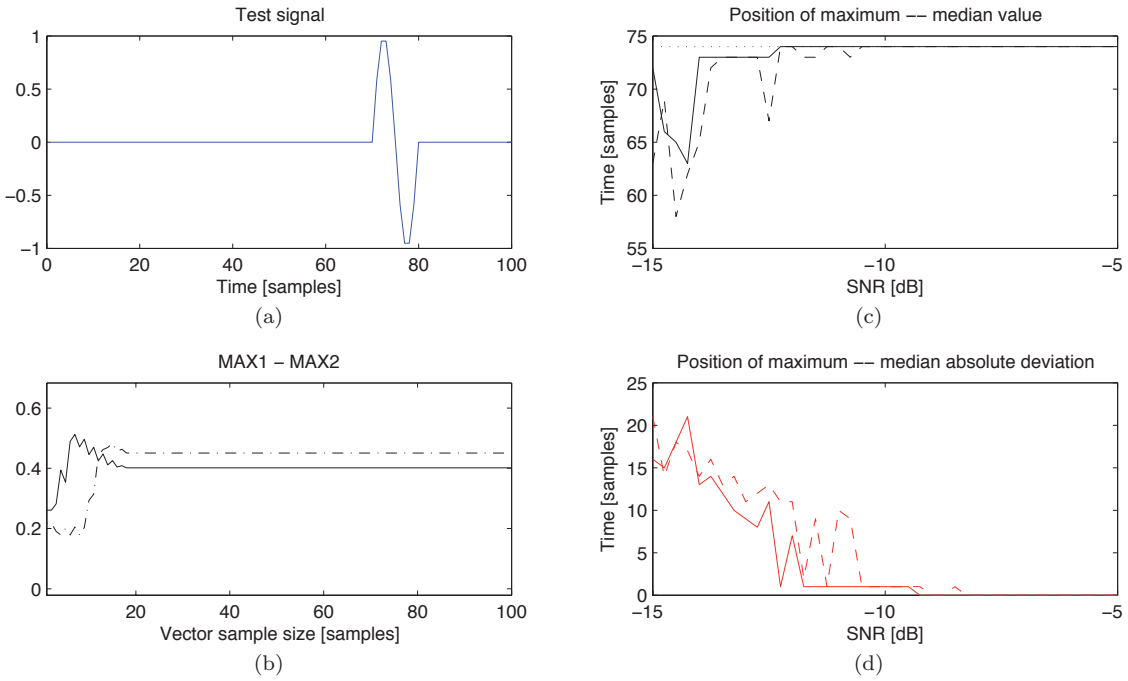


Figure 3.14: Vector samples processing (VeSP) based correlation coefficient. (a) The test signal, containing a one-period sinusoidal transient of $N_r = 10$ samples. (b) Dependance of the *peakedness* of $C^{(w)}$ on the vector samples size, w, for the test signal (continuous line) and for a similar signal with $N_r = 20$ samples (dash-dotted line). $MAX1$ is the maximum value computed on the time support of the transient, while $MAX2$ is computed outside the time support of the transient. (c) Position of the correlation maximum – median value – for $C^{(10)}$ (continuous line), and for $C^{(1)}$ (dashed line). (d) Median absolute deviation of the position of the correlation maximum (same as for (c)).

computed outside this time interval. We note that for a sufficiently large value of w the difference of these two maxima increases compared to the $w = 1$ case. The tests we performed revealed that this observation holds true also in noise conditions. This is the underlying cause of the improvements shown in Figures 3.14(c) and 3.14(d).

The results presented here motivated our research for a VeSP based improvement of the traditional energy detector. This is the subject of the next subsection.

3.4.2 VeSP Energy

The classical solution for detecting unknown transients in noise is estimating how the energy of the observed signal varies in time. This solution does not make any assumptions with regard to the transient or the background noise.

The energy of signal s is computed as:

$$E = \sum_{i=1}^{N} s_i^2 \tag{3.21}$$

By replacing signal s with its vector sampled version $\vec{s}^{(w)}$ (as we did in the previous

subsection), we obtain the vector samples based generalization of the energy:

$$E^{(w)} = \sum_{i=1}^{N} \vec{s}_i^{(w)} \cdot \vec{s}_i^{(w)}, \tag{3.22}$$

where w is the vector sample size. For $w = 1$, (3.22) is equivalent to (3.21), i.e. $E^{(1)} \equiv E$.

In order to study if $E^{(w)}$ improves the detection performance of E, we consider the test signal in Figure 3.14(a). The purpose of our detection is to obtain a time detection curve that has the same number of samples as the analyzed signal and has values of 1 for the positions corresponding to the time support of the transient and values of 0 for the rest of time. The operating principle of the energy detector consists in sliding the center of a rectangular unitary window of W samples through every position in the analyzed signal and computing the energy of that piece of signal. That is, the value at index i in the thus obtained time detection curve is the energy of the signal computed by using only W samples around the signal sample at index i.

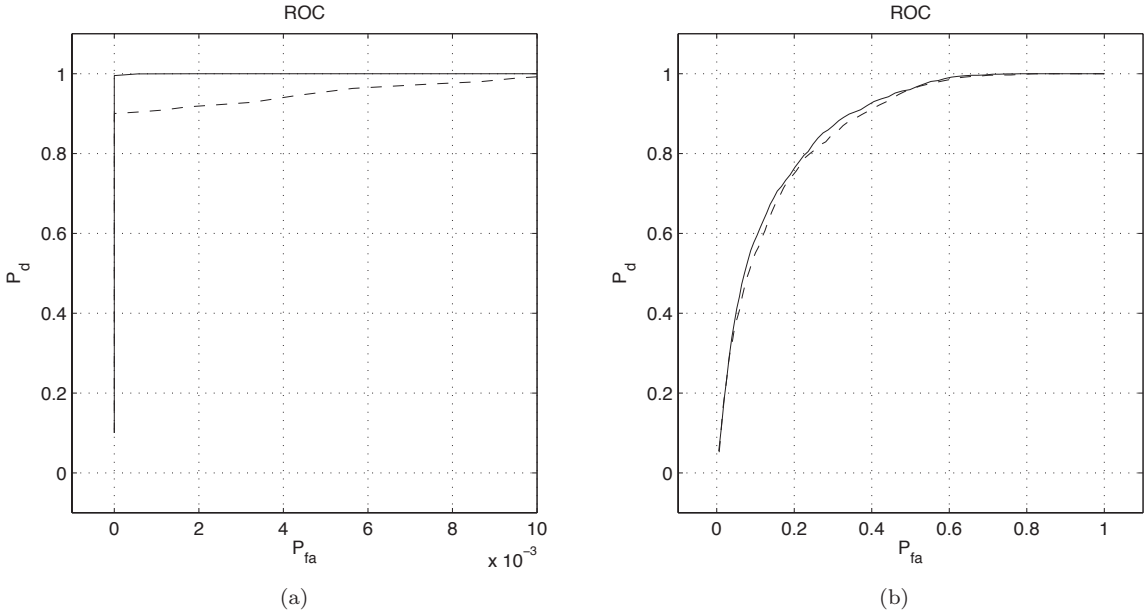


Figure 3.15: Receiver operating characteristic (ROC) curves for VeSP energy detector using the test signal in Figure 3.14(a). ROCs for $E^{(10)}$ are represented with continuous line, and those for $E^{(1)}$ with dashed line. The size of the window used for computing the local energy is $W = 15$. A number of $N_{points} = 100$ points is computed for each ROC, and each of them is the mean value obtained for $N_{rep} = 201$ realizations of noise. (a) SNR = 10 dB. (b) SNR = -10 dB.

In Figure 3.15 we present comparatively the receiver operating characteristic (ROC) curves obtained with the two detectors for two SNRs (10 dB, and -10 dB). (Details about how we computed the ROCs can be found in the Appendix.) The figure shows that $E^{(10)}$ performs better than E both at high SNR and at low SNR.

The results presented in this subsection and in the previous one show that the VeSP philosophy applied to well-known signal processing tools such as correlation and energy improve the efficiency of these tools with respect to detection of transient signals.

3.4.3 VeSP Envelope

When the analyzed signal is not too noisy, a rough detection of transients can be performed by simply computing the signal envelope and then comparing it to a threshold. There are various methods for computing the envelope of a signal, e.g. computing the analytic signal and taking the amplitude envelope (i.e. the absolute value of the analytic signal), finding the local maxima (or minima) and interpolating them, etc.

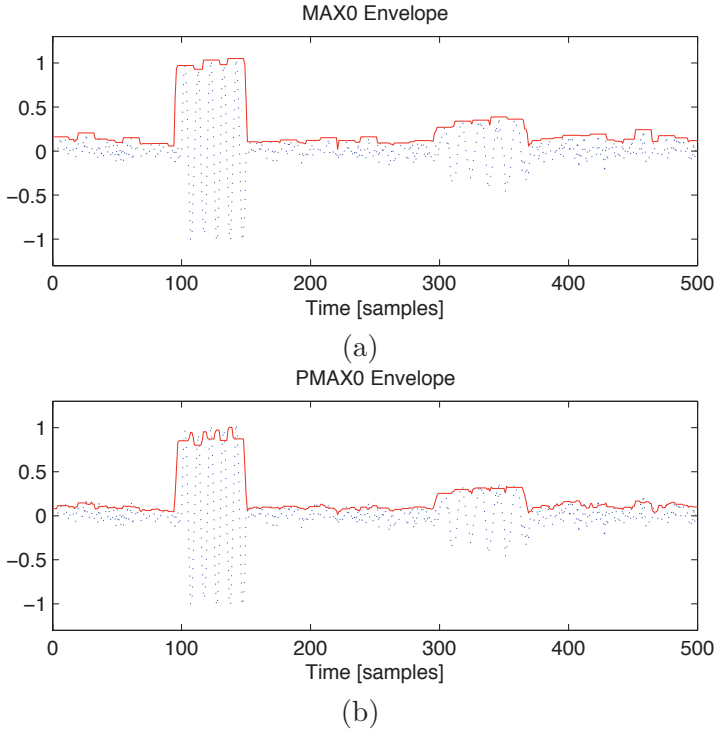


Figure 3.16: Envelopes computed for a test signal containing two sinusoidal transients (plus a 10 dB noise), by using $MAX0$ with $w = 13$ (a), and by using $PMAX0$ with $w = 13$ and $p = 0.1$ (b).

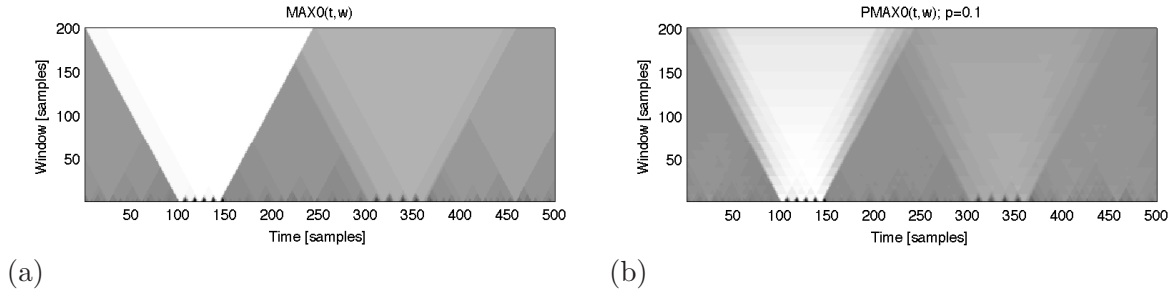


Figure 3.17: Envelopes computed for the test signal with (a) $MAX0$ and (b) $PMAX0$, shown in t-w representations.

One very simple technique for roughly estimating the envelope of a signal consists in sliding a fixed size window over each sample of the analyzed signal and keeping only the maximum value of the signal portion that lies under the analysis window. This is equivalent in VeSP terms to vector sampling the signal s with a vector sampling window of size w, thus resulting the trajectory $\vec{v} = \vec{s}^{(w)}$, and then computing for each vector $\vec{v}_n$ a

norm defined as the maximum value of its components, i.e. computing the measure:

$$MAX0^{(w)}[n] = \max_k \{\vec{v}_n(k)\}. \tag{3.23}$$

(We named it $MAX0$ instead of just MAX in order to indicate that it computes distances not between vectors, but between each vector and the origin, which is vector $\vec{0}$. We did the same for the DR norm, defined in Section 3.3.2).)

This technique can be improved by replacing the Maximum norm with a p-Maximum norm, that would first require choosing a percent of confidence, p, and then sorting in ascending order the components of vector $\vec{v}_n$, discarding the first $w - \lceil p \cdot w \rceil$ values and computing the mean of the remaining values. It can be written as:

$$PMAX0^{(w)}[n] = \text{mean}_{i>w-\lceil p\cdot w\rceil} \{\text{sort}_k \{\vec{v}_n(k)\}[i]\}. \tag{3.24}$$

Despite their simplicity, these methods provide good results when the noise level is not too high. Let us consider a test signal like the one in Figure 4.12. Figure 3.16 shows the envelopes computed using $MAX0$ and $PMAX0$ respectively.

Figure 3.17 illustrates how these envelopes depend on the value of the w parameter. As expected, the larger w, the wider the time interval covered by the envelope.

3.4.4 VeSP Derivative

The (discrete) derivative of a (digital) signal s is computed as the difference between successive signal samples, that is:

$$\Delta\{s\}[n] = s_{n+1} - s_n. \tag{3.25}$$

It can be generalized to vector samples by computing the distance between vectors $\vec{s}_{n+1}^{(w)}$ and $\vec{s}_n^{(w)}$ instead of the difference between samples s_{n+1} and s_n:

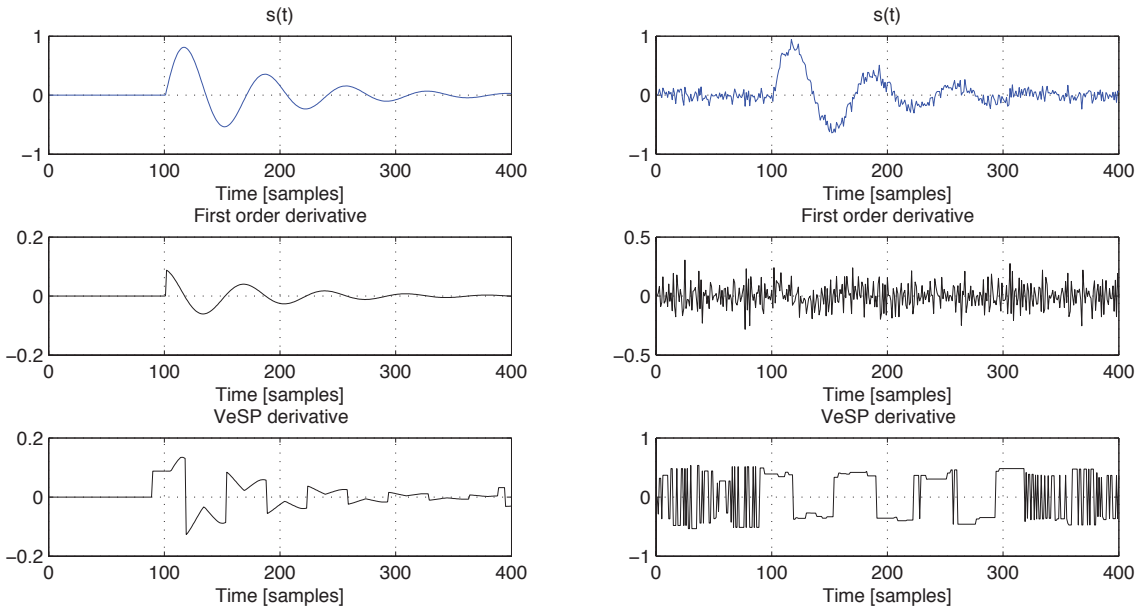
$$\Delta^{(w)}\{s\}[n] = \mathcal{D}\left\{\vec{s}_{n+1}^{(w)}, \vec{s}_n^{(w)}\right\}. \tag{3.26}$$

It is obvious that in this case $\Delta^{(1)} \equiv \Delta$ only if the "distance" $\mathcal{D}$ is defined such that:

$$\mathcal{D}\{s_{n+1}, s_n\} = s_{n+1} - s_n. \tag{3.27}$$

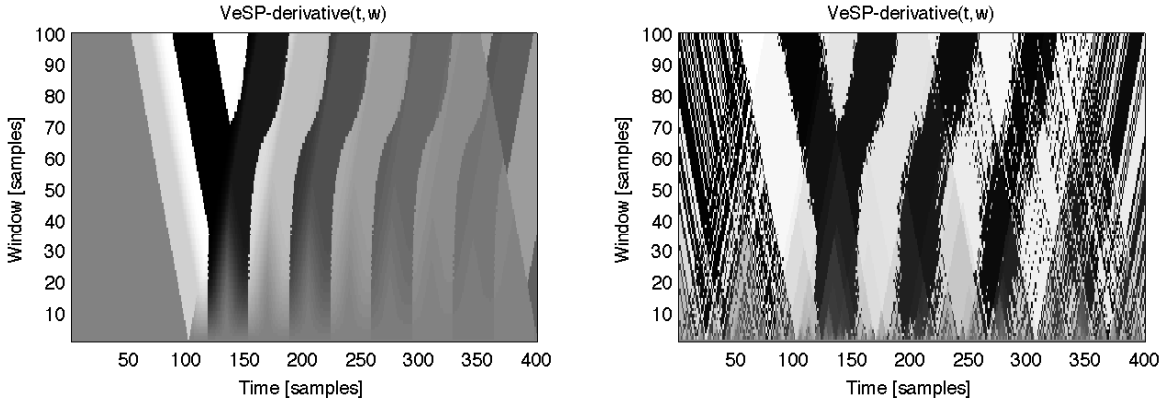
We mention that the Euclidean distance (as well as any distance in the mathematical meaning) does not have this property – instead, $\Delta^{(1)} \equiv |\Delta|$. However, we can make the Euclidean distance comply with (3.27) by multiplying it with a sign. The sign can be chosen in several ways. For instance, it can be chosen as the sign of the majority of the components of vector $\vec{s}_{n+1}^{(w)} - \vec{s}_n^{(w)}$ (*majority sign*), or as the sign of the sum of the components of this vector (*cumulative sign*).

Figure 3.18(a) illustrates the VeSP derivative computed by using the dynamic range distance (defined in Section 3.3.2) and the sign defined as the sign of the sum of the components of the difference between successive vector samples. The test signal is an attenuated transient sinusoid. The classic derivative is also plotted, for comparison purpose.



(a) (b)

Figure 3.18: Illustration of the VeSP derivative computed using the dynamic range distance and cumulative sign, for $w = 25$. (a) No noise case. (b) SNR = 10 dB case.



(a) (b)

Figure 3.19: The VeSP derivative (computed using the dynamic range distance and cumulative sign) for the signals in Figure 3.18, in a t-w representation. (a) No noise case. (b) SNR = 10 dB case.

It can be noticed that zero-crossings are more clearly visible for the VeSP derivative than for the classical one. This suggests that we could use the VeSP derivative for finding the local extrema of signals. Figure 3.18(b) shows that even when the analyzed signal is quite noisy, the zero-crossings of the derivative are still visible in the VeSP derivative.

Of course, there is a relationship between the maximum frequency of the signal and the largest value for w that can be *safely* used. In addition, the larger w is, the less precision we have in estimating the time instants of the zero-crossings. Figure 3.19 illustrates how the VeSP derivative depends with respect to the value of w. It can be easily noticed that the positions of the zero-crossings (i.e. the edges, in the figures) do not change while w is

less than half of the period of the sinusoid.

3.4.5 VeSP AMDF

The average magnitude difference function (AMDF) [Ger03, Muh11] is a common time-domain approach to estimating the fundamental frequency of acoustic signals. For a signal s, the AMDF is computed as follows:

$$AMDF[n] = \frac{1}{N-n} \sum_{i=1}^{N-n} |s_{i+n} - s_i| \,. \tag{3.28}$$

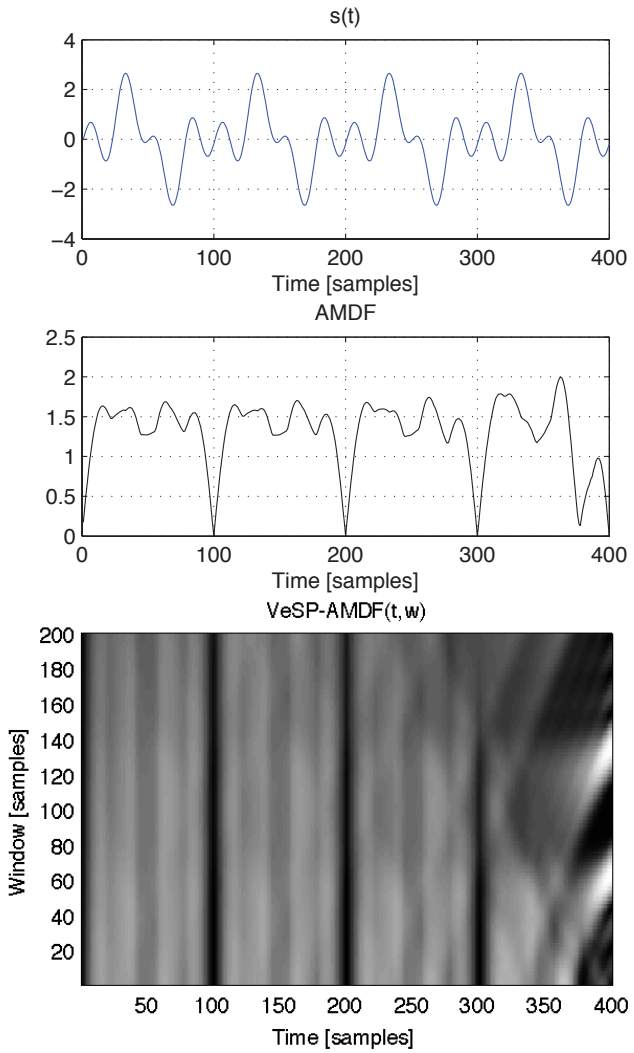


Figure 3.20: Comparative illustration of classic AMDF and VeSP-$AMDF$ (shown in a t-w representation).

By passing to vector samples of size w, we obtain the VeSP version of AMDF, that is:

$$AMDF^{(w)}[n] = \frac{1}{N-n} \sum_{i=1}^{N-n} \left\| \vec{s}_{i+n}^{(w)} - \vec{s}_i^{(w)} \right\| . \tag{3.29}$$

The absolute value used in (3.28) was replaced here with the Euclidean norm. This equation is identical with the previous one when $w = 1$, i.e. $AMDF^{(1)} \equiv AMDF$. Figure 3.20 illustrates the newly obtained $AMDF^{(w)}$. It can be seen that (unlike the other VeSP tools defined so far) it depends very little on the value of w.

In the following section we further extend AMDF by involving recurrences.

3.5 Recurrence-based signal processing measures

We showed in the previous section that VeSP generalizes and in certain situations improves well-known classical signal processing tools. In the current section we study the use of recurrences in further extending some of these tools, as well as in building a new tool that will prove to be very useful in detecting transient signals.

3.5.1 RQA Autocorrelation

The autocorrelation function (ACF) of signal s can be computed as:

$$ACF[n] = \sum_{i=1}^{N} s_i \cdot s_{i+n}. \tag{3.30}$$

By using the dot product distance defined in Section 3.3.3 we can extend ACF to work with vector samples of size w, as follows:

$$ACF^{(w)}[n] = \sum_{i=1}^{N} \vec{s}_i^{(w)} \cdot \vec{s}_{i+n}^{(w)}. \tag{3.31}$$

Or, otherwise put, $ACF^{(w)}$ can be seen as:

$$ACF^{(w)}[n] = \sum_{i=1}^{N} \mathcal{D}\left\{\vec{s}_i^{(w)}, \vec{s}_{i+n}^{(w)}\right\}, \tag{3.32}$$

which is in fact a quantification of the distance matrix (defined in Section 3.2.1). In other words, it is some kind of RQA, except that is it performed not on the recurrence matrix, but on the distance matrix. (In Section 3.2.1 we referred to this kind of quantification as $\text{VeSP}_{(2)}$.)

A proper RQA ACF can be obtained by passing from the distance matrix to the recurrence matrix. Usually this is done by setting to 0 the distances that are above a chosen (recurrence) threshold ε, and setting to 1 all the other distances (i.e. those that indicate recurrences). However, as discussed in Section 3.3.3, the dot product distance is complementary to a *distance* from the point of view of recurrences. This means that the recurrence matrix will actually be reversed. Having said that, the RQA ACF would be defined as:

$$ACF_n^{(w)} = \sum_{i=1}^{N} R_{i,i+n}, \tag{3.33}$$

with the recurrence matrix R computed a little differently, i.e.:

$$R_{i,j} = \Theta\left(\mathcal{D}\left\{\vec{s}_i^{(w)}, \vec{s}_j^{(w)}\right\} - \varepsilon\right). \tag{3.34}$$

3.5.2 RQA AMDF

The AMDF is basically an extension of the ACF. Instead of computing products between samples, it computes differences (as Equation (3.28) shows). We showed in Section 3.4.5 how AMDF can be generalized by using vector samples. The discussion in the previous section regarding ACF remains valid for the AMDF as well – VeSP AMDF is in fact a quantification of the distance matrix, as Equation (3.29) can be also written as:

$$AMDF^{(w)}[n] = \frac{1}{N-n}\sum_{i=1}^{N-n}\mathcal{D}\left\{\vec{s}_i^{(w)}, \vec{s}_{i+n}^{(w)}\right\} = \frac{1}{N-n}\sum_{i=1}^{N-n} D_{i,i+n}. \tag{3.35}$$

Hence, RQA AMDF can be defined as:

$$AMDF_n^{(w)} = \frac{1}{N-n}\sum_{i=1}^{N-n} R_{i,i+n}, \tag{3.36}$$

where the recurrence matrix R is computed as:

$$R_{i,j} = \Theta\left(\varepsilon - \mathcal{D}\left\{\vec{s}_i^{(w)}, \vec{s}_j^{(w)}\right\}\right). \tag{3.37}$$

Except for the normalizing factor in $AMDF_n^{(w)}$, Equations (3.36) and (3.33) are identical. The only real difference between RQA-$AMDF$ and RQA-ACF if the definition they use for the *distance* $\mathcal{D}\{\}$ – ACF uses the dot product distance, while AMDF uses the Euclidean distance. Applications are discussed in Section 4.2.3.

3.5.3 RQA TDH

The idea of time-distributed histogram (TDH) (first introduced in [BISC11]) was obtained empirically, starting from the observation that we detect transients visually by using their *distance* from the rest of the signal (which is only noise). Therefore, transients are detected visually by taking advantage of their *rareness*. That roughly translates to low values in the histogram for transient signal samples and high values for noise samples. This last statement suggests that the idea of TDH might be interesting for transient signal detection – by normalizing TDH and then taking the complement of it we should obtain a rough time detection curve.

We define the TDH of signal s at time instant n as the number of signal samples whose values are in the same histogram bin as sample s_n. Figure 3.21 sketches the graphical meaning of this *definition*. The most interesting part in this definition is the meaning of the statement "two samples have their values in the same bin". An implication of the original definition of the histogram of a signal is that two signal samples are in the same bin if the distance between them is at most the bin size. (Let us note it by ε.) Although

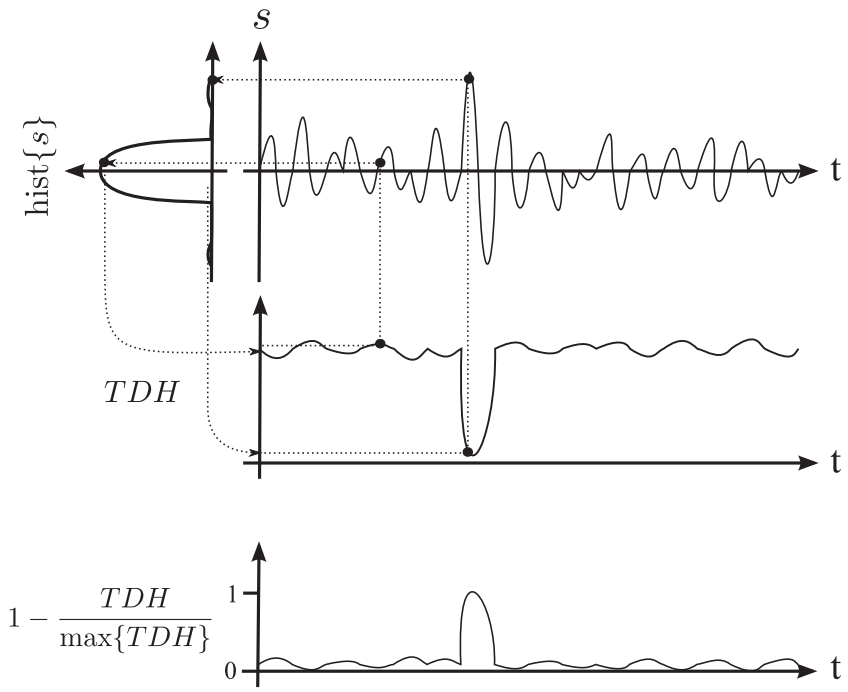


Figure 3.21: Principle of the use of a time-distributed histogram (TDH) for transient signal detection. Rare samples lead to small values in the TDH. A normalized and complemented version of TDH can actually be used for detection purpose.

the converse implication is not valid from a rigorous point of view, we use it to define a new version of TDH, which slightly differs from the TDH defined above. We compute the value of TDH_n by counting the number of samples that are located at a distance of at most ε from the signal sample at instant n, s_n. Figure 3.22 illustrates this distance both from a single samples and from a vector samples points of view.

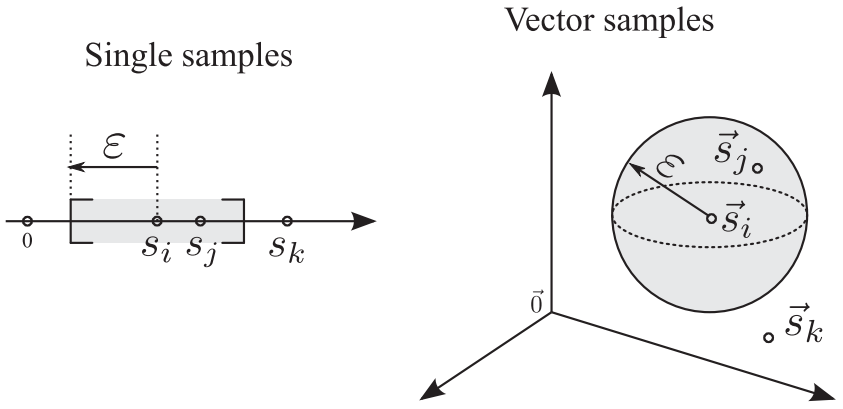


Figure 3.22: Bin shapes of radius ε for single samples and for three-dimensional vector samples. (Vector) samples i and j form a recurrence, while (vector) samples i and k do not.

It is easy to note that by computing the TDH in this way, the "H" in "TDH" refers to some kind of histogram where there are as many bins as different signal amplitudes. Additionally, these bins have a size of 2ε and are centered on the values of the signal samples that they correspond to. It is also easy to note that by this new definition TDH_n

count the number of recurrences around sample s_n (or around vector sample $\vec{s}_n$). In other words, it counts the number of values of 1 in line n in the recurrence matrix (defined in (3.5)). Hence, this new TDH is in fact a RQA measure.

We define the RQA TDH measure as:

$$TDH_n^{(w)} = \frac{1}{N}\sum_{i=1}^{N} R_{i,n}, \tag{3.38}$$

where N is the length of the signal and R is the recurrence matrix (defined in (3.5)) computed from signal vector samples of size w, using the Euclidean distance and a threshold ε. For detection purpose, we actually use a normalized and complemented version of TDH defined here (as shown in Figure 3.21):

$$TDH_n'^{(w)} = 1 - \frac{TDH_n^{(w)}}{\max\{TDH^{(w)}\}}. \tag{3.39}$$

From now on, in the context of detecting transients, by TDH we will actually refer to TDH' defined here.

Besides the vector sampling window size, w, the recurrence threshold, ε, is another parameter that needs to be chosen before computing the RQA-TDH measure (as well as the other RQA measures defined in the previous subsections). The following subsection addresses the problem of automatically choosing the *proper* ε.

3.5.4 Choice of the recurrence threshold

The choice of the recurrence threshold should take into consideration the level of noise in the signal, as noise alters the phase space trajectory that corresponds to the signal. Let us consider two vector samples, $\vec{s}_i$ and $\vec{s}_j$. This pair of vector samples is a recurrence if $\|\vec{s}_j - \vec{s}_i\| \leq \varepsilon$. Now let us consider adding some noise, z, to the signal:

$$s' = s + f_z \cdot z, \tag{3.40}$$

that translates in vector samples representation to:

$$\vec{s'} = \vec{s} + f_z \cdot \vec{z}. \tag{3.41}$$

The recurrence relation (i.e. $\|\vec{s}_j - \vec{s}_i\| \leq \varepsilon$) becomes:

$$\|(\vec{s}_j - \vec{s}_i) + f_z \cdot (\vec{z}_j - \vec{z}_i)\| \leq \varepsilon'. \tag{3.42}$$

Considering the limit case $\|\vec{s}_j - \vec{s}_i\| = \varepsilon$, in order to make sure this recurrence is still caught we should choose an ε' that satisfies:

$$\varepsilon' \leq \varepsilon + f_z \cdot \|\vec{z}_j - \vec{z}_i\|. \tag{3.43}$$

The previous equation shows that ε should increase when the SNR decreases (that is, when f_z increases). However, the SNR is usually unknown a priori. We propose to choose

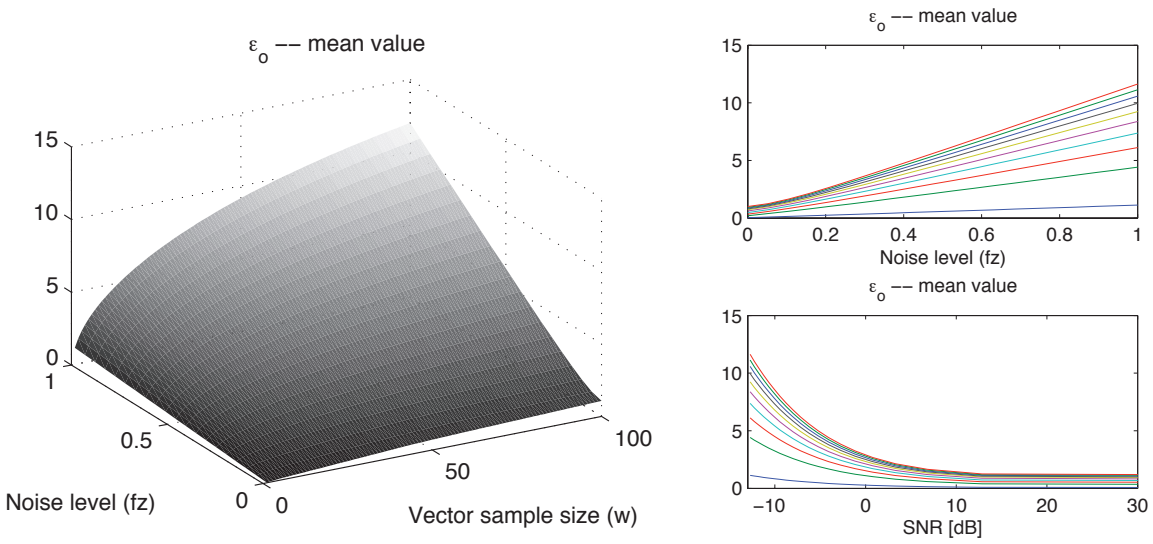


Figure 3.23: Dependence of ε_0 on noise level f_z and vector sample size w. The mean value of ε_0 is shown, for 10 realizations of noise.

ε as the average distance between successive vector samples. If we note this value by ε_0, it is defined as:

$$\varepsilon_0 = \frac{1}{N-1} \sum_{i=1}^{N-1} \|\vec{s}_{i+1} - \vec{s}_i\| \,. \tag{3.44}$$

(To be more generic, we can use $\mathcal{D}\{\vec{s}_{i+1}, \vec{s}_i\}$ instead of $\|\vec{s}_{i+1} - \vec{s}_i\|$.)

First of all, we note that this value automatically adapts to signal amplitude scalings, i.e.:

$$\varepsilon_0\{\alpha \cdot s\} = |\alpha| \cdot \varepsilon_0\{s\}, \tag{3.45}$$

where α is a real valued scaling factor, and $\varepsilon_0\{x\}$ denotes the computation of ε_0 for signal x.

Considering the noised signal in (3.40), we are interested in how ε_0 varies with respect to the level of noise f_z. It is also interesting to study how ε_0 depends on the size of the vector samples, w. Figure 3.23 shows this variation for the test signal in Figure 3.14. It can be noted that ε_0 increases as the SNR decreases.

The ε_0 measure will be used in the following chapter in conjuction with RQA TDH for performing automatic detection of transient signals.

3.6 Summary

The keywords of this chapter were *vector sample* and *recurrence*. They summarize the entire philosophy of recurrence plot analysis. We have shown how they can be involved in generalizing and extending various signal processing tools. Therefore, RPA is not an "exotic" nonlinear time series analysis tool, but it can be seen as a framework that brings together in a common formulation different well-known signal processing tools, which it also generalizes (through vector samples) and extends (through distances between vector

samples). We have also shown that although they lose the linearity property, some of these tools outperform their linear counterparts.

We put these tools to work in the following chapter, where we address different problems regarding the analysis of transient signals.

Chapter 4

Applications of RPA tools in transient signal processing problems

The different recurrence quantification measures developed in the previous chapter are generalized in the current one into some signal representation spaces that highlight transient features in the analyzed signal. We discuss the time - recurrence radius representations and the time - vector sampling size representations. Next, we study their performance in dealing with (transient) signal processing tasks such as detection, noise reduction, fundamental frequency estimation, and slow-rate chirp identification.

4.1 Signal representations based on RPA

In this section we show that the RQA measures defined in the previous chapter practically come from two methods of quantification of the recurrence plot, namely diagonal-wise summing, and column-wise summing. We obtain thus two generic RQA measures that essentially depend on two parameters – the size of the vector sampling window, and the size of the recurrence radius. By setting one of these parameters equal to a fixed value and making the other one go through a range of values, we obtain bidimensional signal representation (where one of the axes is, of course, the time axis). We show that these representations emphasize some signal features as the time localization of transients and their fundamental period.

4.1.1 The σ_d and σ_c RQA measures

If we look carefully at the expressions of the RQA measures defined in the previous section, we notice that what they do in fact is computing a normalized sum of the elements of the recurrence matrix, R, either diagonal-wise (as RQA AMDF does) or column-wise (as RQA TDH does). We note by σ_d the normalized sum of the diagonals in R, and by σ_c the measure obtained by computing the normalized sum of the columns in R. Figure 4.1 schematically illustrates these two generic RQA measures.

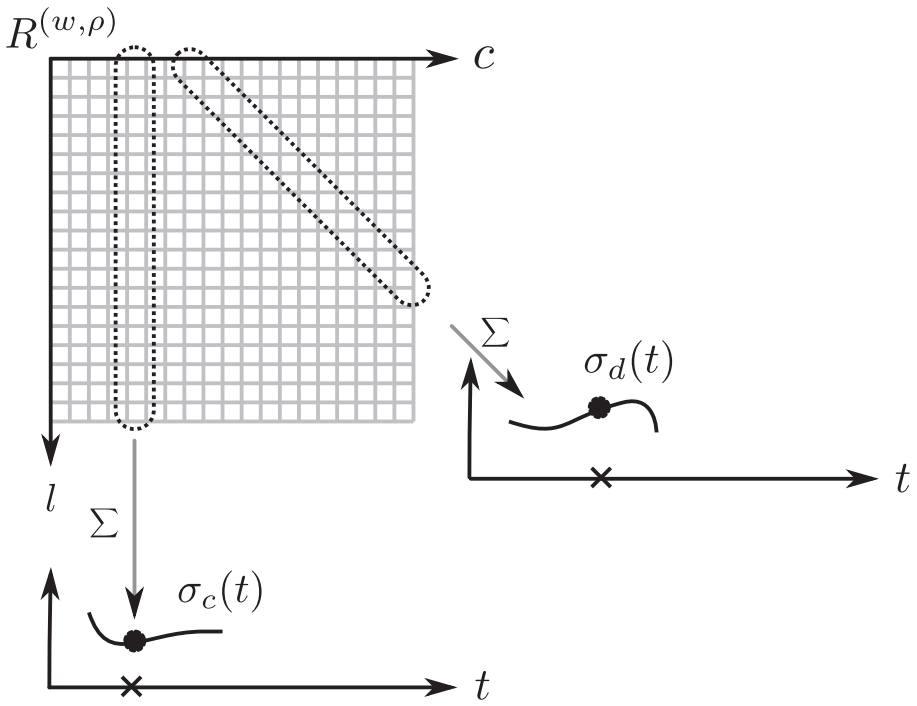


Figure 4.1: Obtaining the σ_c and σ_d generic RQA measures from the recurrence plot computed from signal vector samples of length w.

Mathematically, σ_d is computed as:

$$\sigma_d(t) = \frac{1}{N-t} \sum_{i=1}^{N-t} R_{i,i+t}, \tag{4.1}$$

while σ_c is computed as:

$$\sigma_c(t) = \frac{1}{N} \sum_{i=1}^{N} R_{i,t}. \tag{4.2}$$

We note by $\bar{\sigma}_c$ the complementary value of σ_c, that is:

$$\bar{\sigma}_c(t) = 1 - \sigma_c(t). \tag{4.3}$$

Both $\bar{\sigma}_c$ and σ_d require two parameters: w, i.e. the vector sampling window, and ε, i.e. the recurrence threshold. If we use a fixed value for w while sweeping the whole range of values for ε, we obtain two t-ε representations: $\bar{\sigma}_c^{(w)}(t,\varepsilon)$ and $\sigma_d^{(w)}(t,\varepsilon)$. In the same manner, if we use a fixed value for ε while sweeping the whole range of values for w, we obtain two t-w representations: $\bar{\sigma}_c^{(\varepsilon)}(t,w)$ and $\sigma_d^{(\varepsilon)}(t,w)$.

In order to illustrate these representations, we use the test signal in Figure 4.2.(a). It contains a 50 samples sinusoidal burst starting at time $t = 100$ samples and having a period of 10 samples. The signal-to-noise ratio is 10 dB. The second signal only contains the chopped burst, with the time axis zoomed in.

4.1.2 The t-ε representations

Figure 4.2 illustrates the two t-ε representations, obtained for a w of 10 samples. The $\bar{\sigma}_c^{(w)}(t,\varepsilon)$ representation is computed for the whole test signal, while $\sigma_d^{(w)}(t,\varepsilon)$ is computed for the chopped burst only.

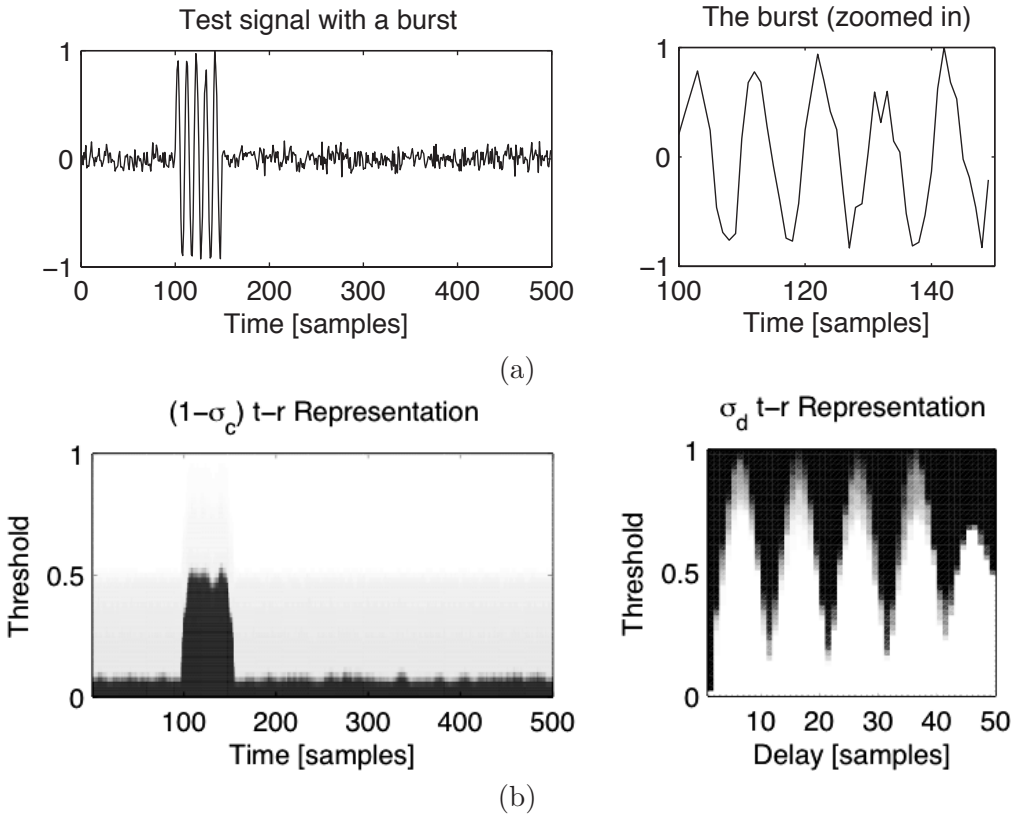


Figure 4.2: (a) The test signals: signal with a sinusoidal burst, and zoomed in burst. (b) The $\bar{\sigma}_c^{(w)}$ and $\sigma_d^{(w)}$ t-ε representations, computed using $w = 10$ samples.

It can be noticed that choosing the optimum ε is not straightforward. It should be sufficiently large to leave the noisy area in $\bar{\sigma}_c^{(w)}(t,\varepsilon)$, but in the same time it should be sufficiently small to get even the last peak in $\sigma_d^{(w)}(t,\varepsilon)$. A measure that fulfills these criteria is the average distance between all the successive vector samples of the signal, i.e. the ε_0 measure defined in the previous section.

4.1.3 The t-w representations

The two t-w representations are illustrated in Figure 4.3. They are computed for the same signals used in Figure 4.2, by using $\varepsilon = \varepsilon_0$.

It is to be noticed from these two representations that the bigger w is, the more the influence of noise is reduced. However, big values for w lead to an artificial incrementation in the duration of the burst detected in $\bar{\sigma}_c$, while for σ_d they lead to a decrementation in the number of maxima. Choosing w around half of the expected period of the transient signal seems to be a good compromise, as we show in the next section, which illustrates the use of $\bar{\sigma}_c$ and σ_d in some transient signal processing problems.

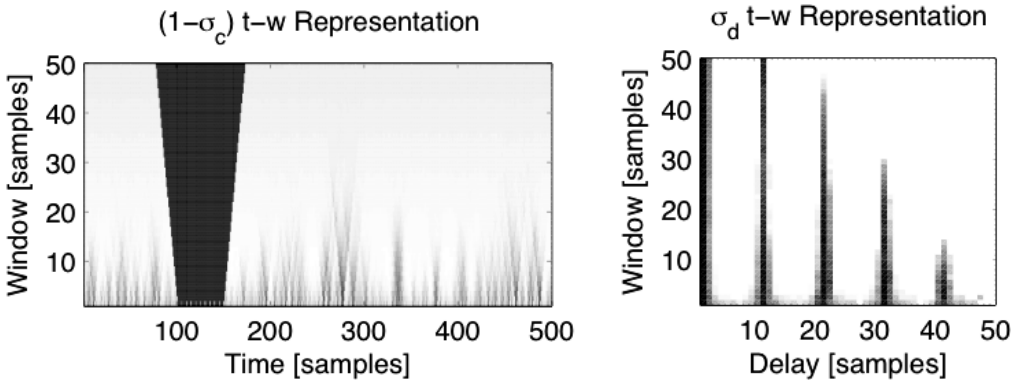


Figure 4.3: The $\bar{\sigma}_c^{(\varepsilon)}$ and $\sigma_d^{(\varepsilon)}$ *t-w* representations, computed using $\varepsilon = \varepsilon_0$.

4.1.4 Discussion

The σ_d and σ_c defined in the current section seem to be ssentially nothing else but the RQA AMDF and RQA TDH respectively, but they are in fact generalizations (as they do not necessarily rely on the Euclidean distance, although this was the distance that we used here). Even more, they can be computed not only on the recurrence matrix, but also on the distance matrix. Hence, RQA TDH can be seen as a special case of σ_c, while RQA AMDF (as well as RQA ACF) can be seen as a special case of σ_d.

We should mention that besides the two *t-w* representations defined here, two more such representations were discussed – in Section 3.3.2 (– the $DR0$ norm) and in Section 3.4.4 (– the VeSP derivative).

These tools are put to work in the next section, where several transient signal processing tasks are addressed.

4.2 Transient signal processing using RPA tools

The current section tries to study the practical utility of the signal processing tools discussed so far, by focusing on different applications where transient signals are involved. While the main focus is detection, it also addresses problems such as noise reduction, fundamental period estimation, and slow chirp identification.

4.2.1 Detection

This subsection is devoted to the study of the performance of the RQA-TDH measure in detecting transient signals. (We recall that here by TDH we refer in fact to TDH' defined in (3.39).) As we are interested in developping a detector that requires no a priori knowledge on the signal to detect, the only detectors left for comparison are those based on the energy of the signal and those based on the histogram of the signal. The second class of detectors mentioned here is based either on assuming Gaussian noise and measuring signal deviation from Gaussianity, or on computing the histogram of the signal and comparing it with a reference one. In both cases, the probability density function of

the noise is assumed to be known in advance. This is not the case of RQA-TDH, which makes no such assumptions. Therefore, the only fair comparison of TDH can be done with energy based detectors. We compare the performance of $TDH^{(w)}$ with that of the classical energy detector, E, and VeSP energy detector, $E^{(w)}$ (discussed in Section 3.4). We already showed (Figure 3.15) that VeSP energy detector improves in certain cases the classical energy detector, but these improvements are limited to the fact that the VeSP version of the energy time detection curve is actually a time filtered version of the classical one.

Synthetic signals

We begin our study with the synthetic signal in Figure 4.4(a). It consists in two one-period sinusoidal transients of different amplitudes and durations. (The ratio between their amplitudes is 0.1.) As there is no noise in the signal, the two energy detectors provide quasi-identical time detection curves (Figure 4.4(b)). They provide a good localization for the two transients, but the difference between their energies is highly visible in the detection curve, which has significant differences when compared to the ideal one.

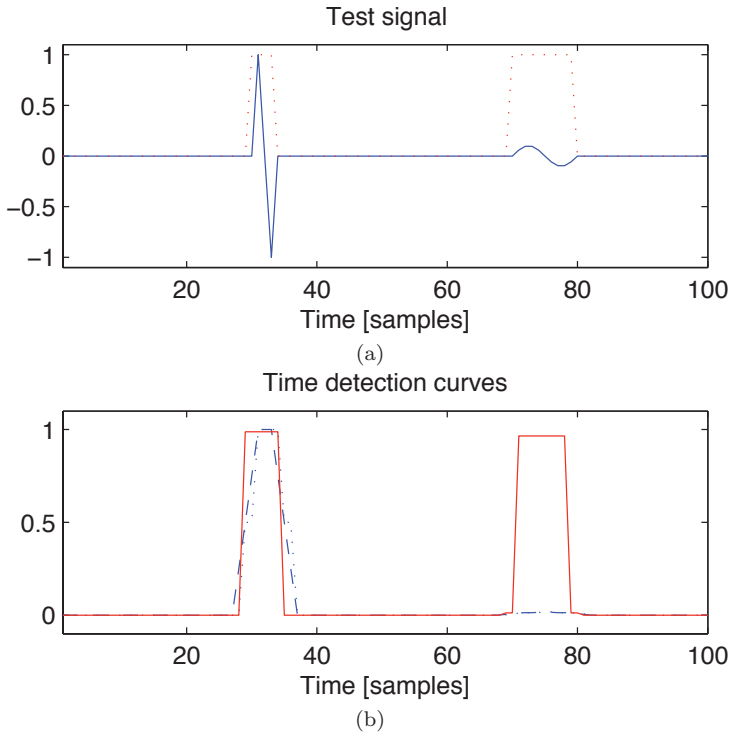


Figure 4.4: Synthetic test signal and time detection curves. The first transient has 4 samples, and the second has 10 samples. (a) Test signal (continuous line) and ideal time detection curve (dotted line). (b) Time detection curves obtained with $E^{(1)}$ (dotted line), $E^{(2)}$ (dashed line), and $TDH^{(4)}$ (continuous line). The size of the window used for computing energy is $W = 6$ for both $E^{(1)}$ and $E^{(2)}$.

The parameter values we used for this detectors were obtained experimentally, by searching to optimize detection performance. On the other hand, it is remarcable how well the time detection curve provided by $TDH^{(4)}$ (with the recurrence radius $\varepsilon = \varepsilon_0$ defined in Section 3.5.4) approaches the ideal one. Hence, from a qualitative point of

view, the time detection curve given by RQA-*TDH* has superior performance (at least for the analyzed signal, at SNR $\rightarrow \infty$).

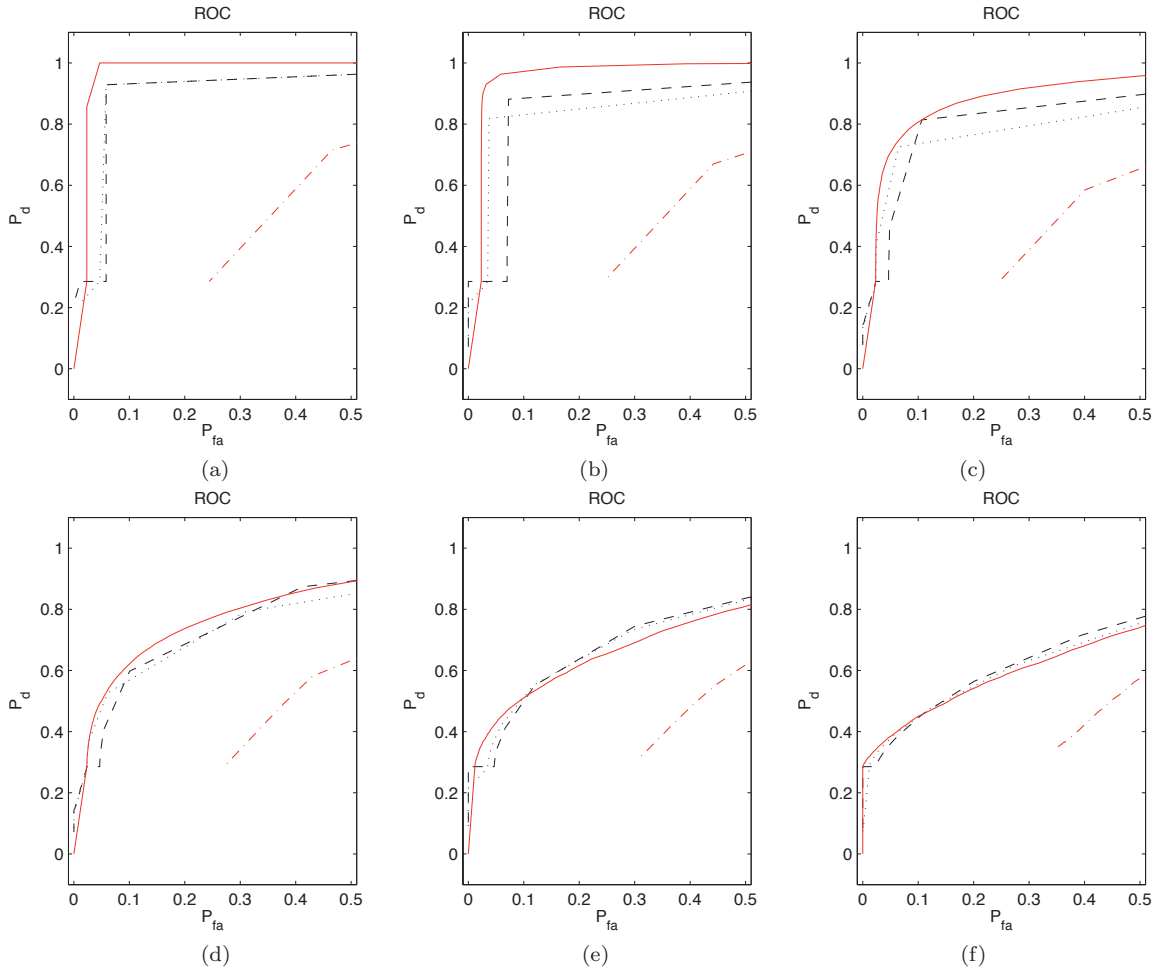


Figure 4.5: ROCs computed for synthetic test signal in Figure 4.4(a) using $E^{(1)}$ (dotted line), $E^{(w)}$ (dashed line), $TDH^{(w)}$ (continuous line), and wavelet + higher order statistics (dash-dotted line). $N_{points} = 100$ points are computed on each ROC curve, which is the average of $N_{rep} = 500$ repetitions. (a) SNR $\rightarrow \infty$; $W = 6$; $E^{(2)}; TDH^{(4)}$. (b) SNR = 15 dB; $W = 5$; $E^{(4)}; TDH^{(4)}$. (c) SNR = 10 dB; $W = 4$; $E^{(3)}; TDH^{(4)}$. (d) SNR = 7 dB; $W = 4$; $E^{(3)}; TDH^{(4)}$. (e) SNR = 5 dB; $W = 5$; $E^{(2)}; TDH^{(3)}$. (f) SNR = 3 dB; $W = 3$; $E^{(2)}; TDH^{(2)}$.

However, in the classical problem of signal detection we are not interested in having a time detection curve that approaches as much as possible the ideal one, but in being able to choose a fixed threshold for binarizing this primary time detection curve in order to obtain the final exploitable detection curve. This can be studied in terms of ROCs, which provide, therefore, a more quantitative view on the performance of the time detection curves. As mentioned before, details about the computation of ROCs can be found in the Appendix. Figure 4.5 shows ROCs obtained using the three detectors for six SNRs on the test signal in Figure 4.4. We also plotted the ROCs obtained for a detector based on wavelets and higher order statistics [RA98], to show that even if this kind of approaches generally have high performance, they can not be used as a potential solution to our problem here (i.e. detecting very short transients having very different energies). For each case, we tried to choose for each detector those parameter values for which it performed

best. (For the wavelet + higher order statistics detector we used a time resolution of 16 samples and a confidence percentage of 0.01.) The ROCs are zoomed in order to focus on the area of interest (i.e. small probability of false alarm, P_{fa}). The results provided in the figure show that while VeSP energy performs better than classical energy starting from a certain P_{fa}, generally $TDH^{(w)}$ outperforms both of them up to a SNR of around 5 dB.

Experimental signals

We continue the performance study of our detectors with the experimental signal in Figure 4.6.

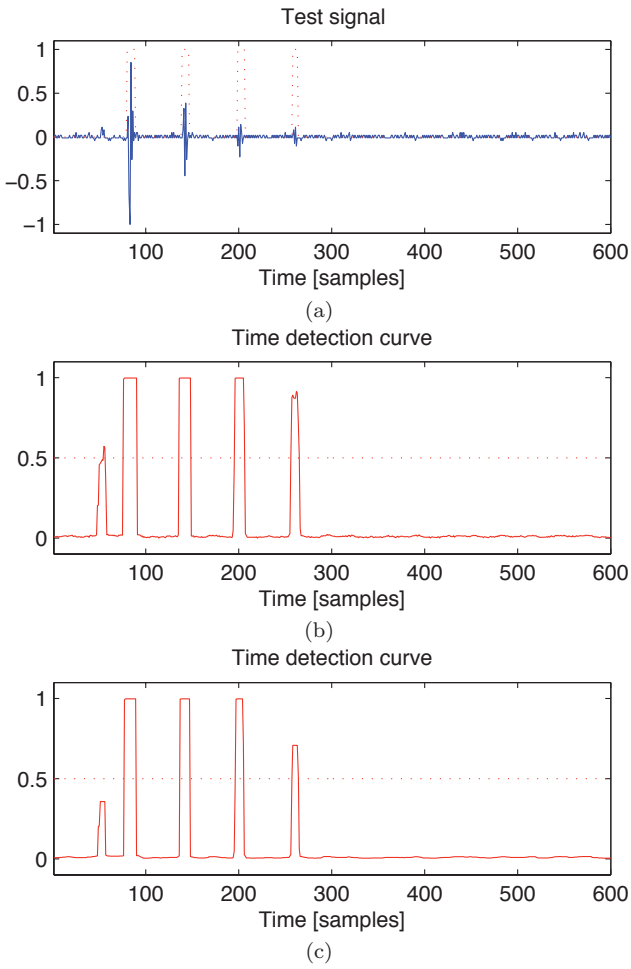


Figure 4.6: (a) Experimental test signal (from ultrasonic nondestructive testing) and ideal time detection curve. (b) Time detection curve obtained with $TDH^{(10)}$. (c) Time detection curve obtained with $TDH^{(10)}$ followed by a median filtering using a window of size $w_{filt} = 6$.

It is a typical signal for ultrasonic nondestructive testing. We obtained it by emitting a short sinusoidal ultrasonic pulse through a metallic block. The first (and strongest) pulse received corresponds to the first arrival of the emitted signal, and the following pulses are reflexions produced when the signal reaches the boundaries of the tested object. These pulses provide information about the thickness of the tested object. They may also

provide information about the presence and position of the eventual holes or cracks inside the object. Estimating the time support of these pulses is, hence, very important.

The middle plot in Figure 4.6 shows that RQA-TDH is able to properly detect positions in time and durations of the pulses (by binarizing the obtained time detection curve with a threshold of 0.5). But it also detects an unwanted short noise burst (at the beginning). This can be solved by using a median filtering (with a filtering window of size w_{filt}) of the time detection curve given by $TDH^{(w)}$. This filtering reduces the short duration transients originally detected by $TDH^{(w)}$, as the bottom plot in Figure 4.6 shows. (A brief recall of the median filter can be found in Section 3.4.1.)

Combining RQA-TDH with median filtering provides good results even in more complex situations, as the signal in Figure 4.7(a). It contains echolocation clicks generated by a Tursiops truncatus dolphin, measured in coastal environment in Cotentin, France. Detection of clicks is the first step in the analysis of such signals. Figure 4.7(b) shows that RQA-TDH followed by a median filtering is able to produce a time detection curve that approaches significantly the desired one.

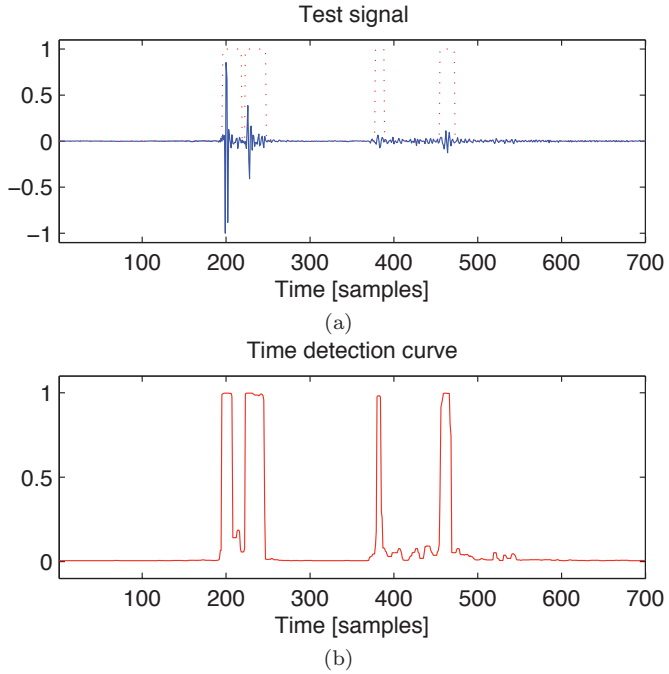


Figure 4.7: Time detection curve for dolphin echolocation signal. (a) Test signal and desired time detection curve. (b) Time detection curve obtained with $TDH^{(4)}$, plus median filtering with $w_{filt} = 6$.

In Figure 4.8 we show what happens to this detection curve when adding artificial noise. The detection curves provided by the energy detector are very difficult to appreciate visually, as the energies for the last two transients are very low compared to the energy of the first one. Figure 4.9 facilitates this comparison, providing ROCs computed for the two detectors on this signal.

RQA-TDH detection performs better than energy detection up to an SNR of 5 dB, which is the SNR up to which the third transient is still detected well, as Figure 4.8(b) shows. However, the ROCs in Figure 4.9 are computed by considering a fixed set of parameters for RQA-TDH based detection. Figure 4.10 shows that by properly choosing

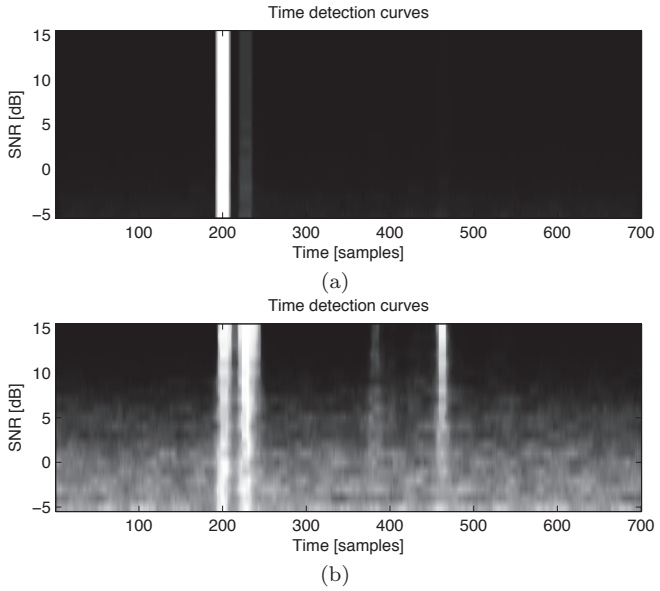


Figure 4.8: Time detection curves for dolphin echolocation signal at various SNRs, for (a) energy detector using a window of $W = 16$ samples, and for (b) RQA-TDH detector ($w = 4$; $w_{filt} = 6$). Each horizontal line is the average of 10 time detection curves computed at the corresponding SNR value. (Black is 0 and white is 1.)

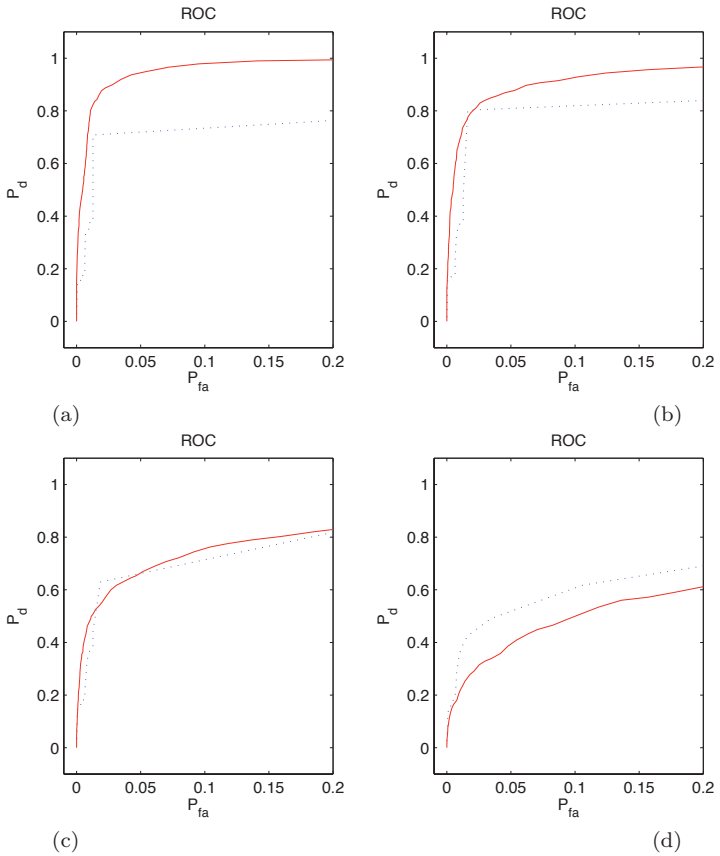


Figure 4.9: ROCs ($N_{points} = 100$ and $N_{rep} = 50$) obtained for $TDH^{(w)}$ (continuous line; $w = 4$; $w_{filt} = 6$) and E (dotted line; $W = 16$), using dolphin echolocation signal. (a) SNR = 15 dB. (c) SNR = 10 dB. (c) SNR = 5 dB. (d) SNR = 0 dB.

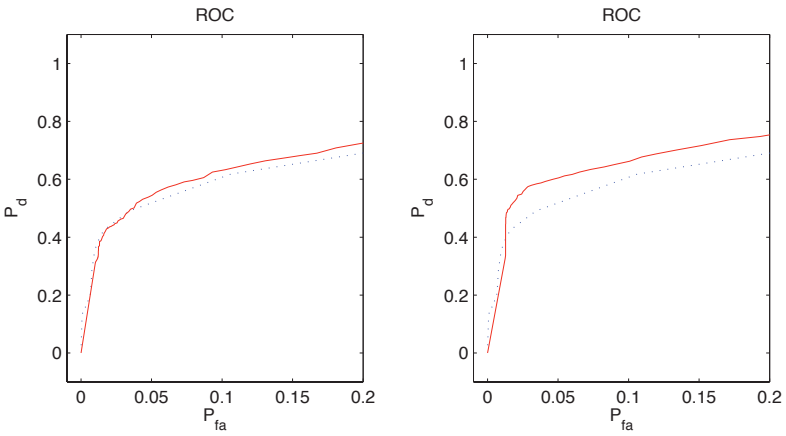


Figure 4.10: ROCs at SNR = 0 dB, computed for $TDH^{(w)}$ (continuous line) and for E (dotted line; $W = 16$), using dolphin echolocation signal. (a) $w = 16$; $w_{filt} = 16$. (b) $w = 16$; $w_{filt} = 32$.

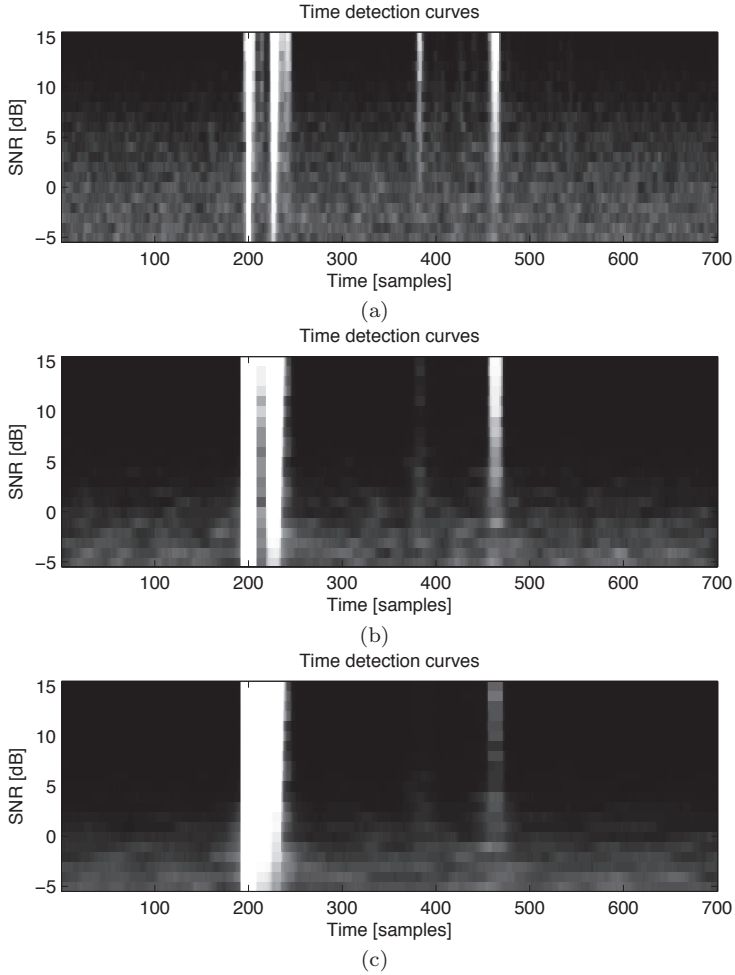


Figure 4.11: Time detection curves at SNR = 0 dB for different configurations of $TDH^{(w)}$ detection, using dolphin echolocation signal. (a) $w = 4$; $w_{filt} = 0$. (a) $w = 16$; $w_{filt} = 16$. (c) $w = 16$; $w_{filt} = 32$.

the parameters, TDH may outperform E even at low SNR. However, ROCs should be regarded with care, as those in Figure 4.10 hide a possibly important fact about the detection curve, i.e. that it may fail to separate two transients that are close in time. This is the case in Figure 4.11(c) – although the detector performance increases in terms of ROC (as Figure 4.10 shows), it may decrease in terms of detecting transients of very short duration and separating transients that are close to each other in time. This last fact may be of increased interest in certain applications, such as the analysis of ECG signals.

4.2.2 Noise reduction

Now let us consider the test signal in Figure 4.12. It contains two transients having different start times, durations, and frequencies. As we said before, this kind of signals may appear in various applications (e.g. ultrasonic testing, telecommunications) where the common point consists in emitting short burst signals that propagate through a certain environment and finally reach the receiver. Reflexions and multipath propagation may lead to different delays and attenuations in the received signal. Regardless of which is the exact application, for a signal like the one in Figure 4.12 we are interested in estimating the different parameters of the transients it contains.

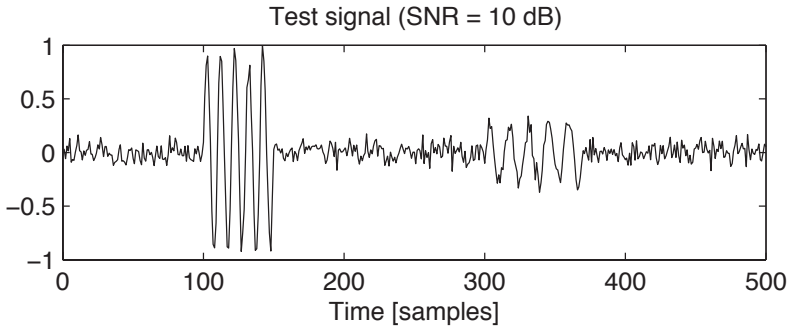


Figure 4.12: The test signal: two sinusoidal bursts with different amplitudes and frequencies.

We suggested in Section 4.1 that the t-ε and t-w representations based on the $\bar{\sigma}_c$ RQA measure can be helpful in detecting the time supports of transients in a signal, and in the previous section we showed a practical implementation of a transient signal detector (based on TDH and ε_0). Figure 4.13 illustrates the time detection curve obtained by computing the $\bar{\sigma}_c$ measure (by using the parameters w=7 and $\varepsilon = \varepsilon_0$).

The detection problem was treated in the previous section. Here we only make the remark that although the time detection curve in Figure 4.13 has highly irregular behaviour in the noise areas, it has the remarcable property that it has high values (very close to 1) in the transient areas, and has low values in the noise areas. This property is not found in any energy based detector (as low energy transients generate low values in the time detection curve), and we can take advantage of it in order to reduce noise in the analyzed signal.

Noise in the test signal (Figure 4.12) can be reduced by multiplicating it with the time detection curve (Figure 4.13). The transients will remain almost unchanged, while noise

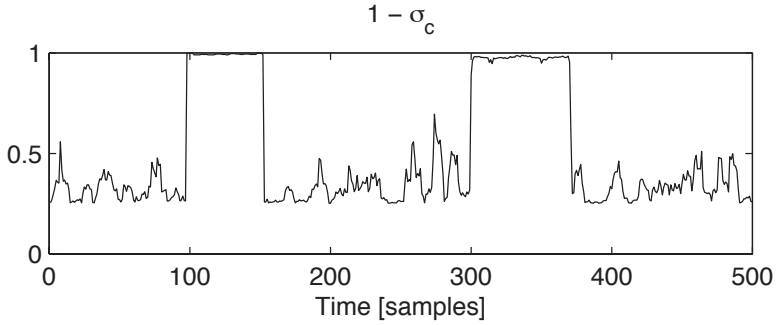


Figure 4.13: Time detection curve obtained for the test signal by computing the RQA measure $\bar{\sigma}_c^{(w=7,\varepsilon=\varepsilon_0)}$.

in the noise-only areas will be attenuated. Hence, this is not actually a noise removal – but a reduction of noise in the uninteresting areas of the signal. Further, if we compute the $\bar{\sigma}_c$ measure on the new signal, we obtain a cleaner time detection curve. For instance, if we repeat the whole process one more time (i.e. if we multiply the new signal with this new $\bar{\sigma}_c$, and then we compute $\bar{\sigma}_c$ for the result), we obtain the time detection curve in Figure 4.14.

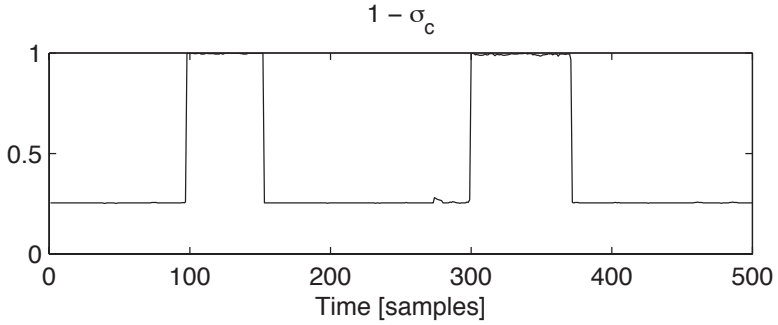


Figure 4.14: Time detection curve obtained by using $\bar{\sigma}_c^{(w=7,\varepsilon=\varepsilon_0)}$, after two iterations.

The newly obtained signal is cleaner, as well. Figure 4.15 shows it.

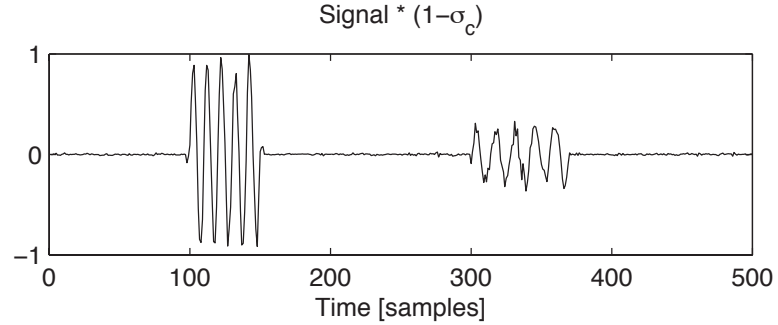


Figure 4.15: Noise reduced signal obtained by multiplicating the analyzed signal with $\bar{\sigma}_c^{(w=7,\varepsilon=\varepsilon_0)}$, after the second iteration.

In order to globally analyze the behaviour of this iterative technique for noise reduction, we study what happens for different SNRs between 30 dB and -30 dB. Figure 4.16

shows the results obtained with this technique (i.e. compute $\bar{\sigma}_c$, multiply it with the signal, then compute again $\bar{\sigma}_c$ on the new signal, multiply it with the new signal, an so on) by using a number of four iterations.

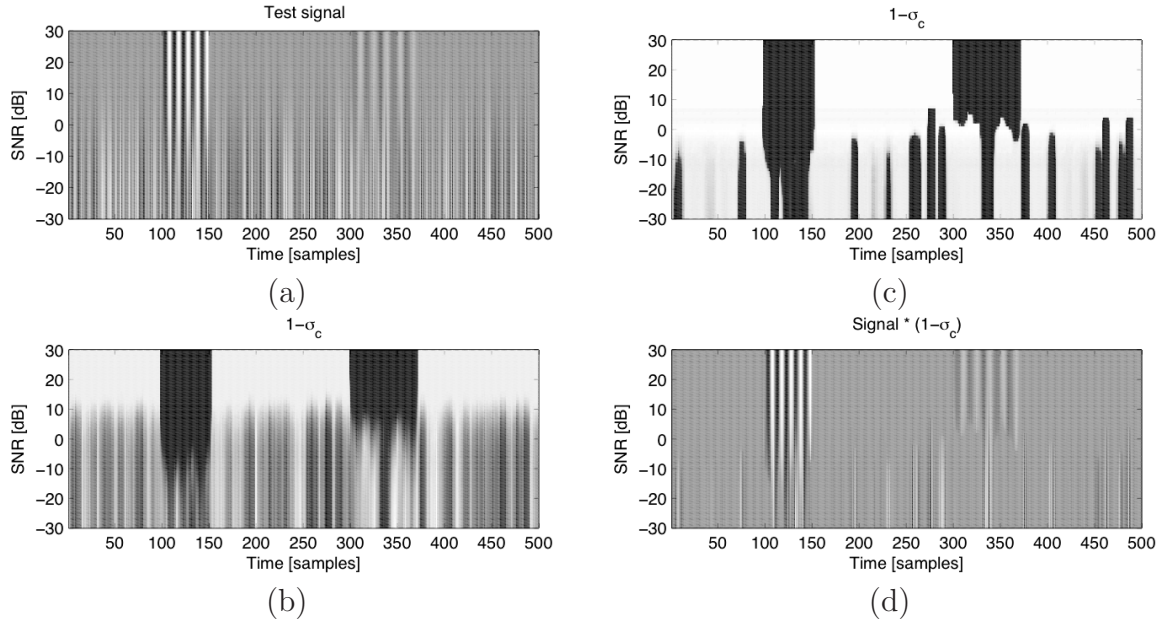


Figure 4.16: (a) The test signal in Figure 4.12, for multiple signal-to-noise ratios. (b) Time detection curve obtained with $\bar{\sigma}_c$. (c) Time detection curve obtained with $\bar{\sigma}_c$, after four iterations. (d) Noise reduced signal, obtained after four iterations. $(w = 7, \varepsilon = \varepsilon_0)$

It it obvious from the figure that this kind of iterations do not improve too much the detection performance of the method with respect to SNR. Still, some minor improvements can be seen. And it can also be seen that the new time detection curves (see Figure 4.16.(c)) are much cleaner than the original ones (see Figure 4.16.(b)). This highly simplifies their postprocessing in order to obtain a segmentation of the analyzed signal into its constitutive transients.

4.2.3 Fundamental frequency estimation

In addition to detecting the transients in a signal, we might also be interested in estimating their fundamental frequencies (for instance, in order to detect the frequency deviation in Doppler ultrasonic instrumentation). There is a broad range of fundamental frequency estimation (also known as *pitch detection*, in speech processing) techniques in literature [Ger03]. Among them, time domain techniques like the AMDF are very common. They are actually used for estimating the fundamental period, as they generate maxima (or minima, depending on the actual measure used) that mark the positions in time where the signal repeats itself. We already showed in Section 3.4.5 a vector samples based generalization of the AMDF. In Section 3.5.2 we further extended it by also making use of RQA.

If we go back to Section 4.1 and look at the t-ε and t-w representations based on the σ_d RQA measure, we see that the repetition points of the signal are very well marked. Now

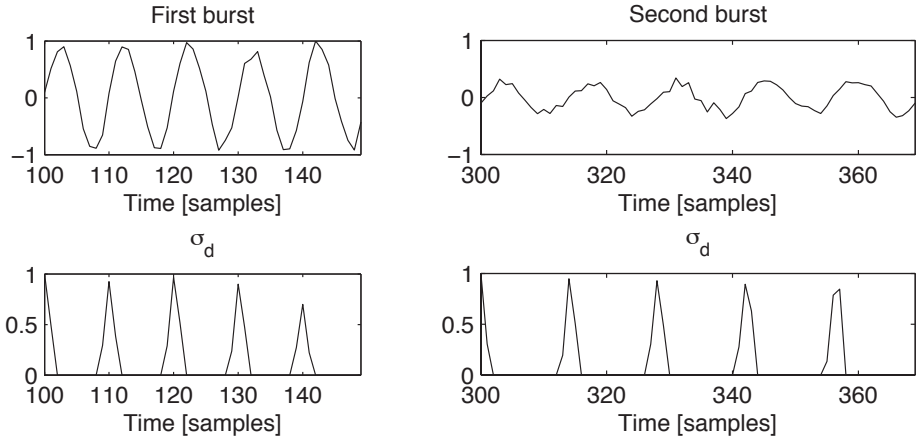


Figure 4.17: The RQA measure σ_d, computed for the two transients in the test signal in Figure 4.12, using $w = 7$ and $\varepsilon = \varepsilon_0$.

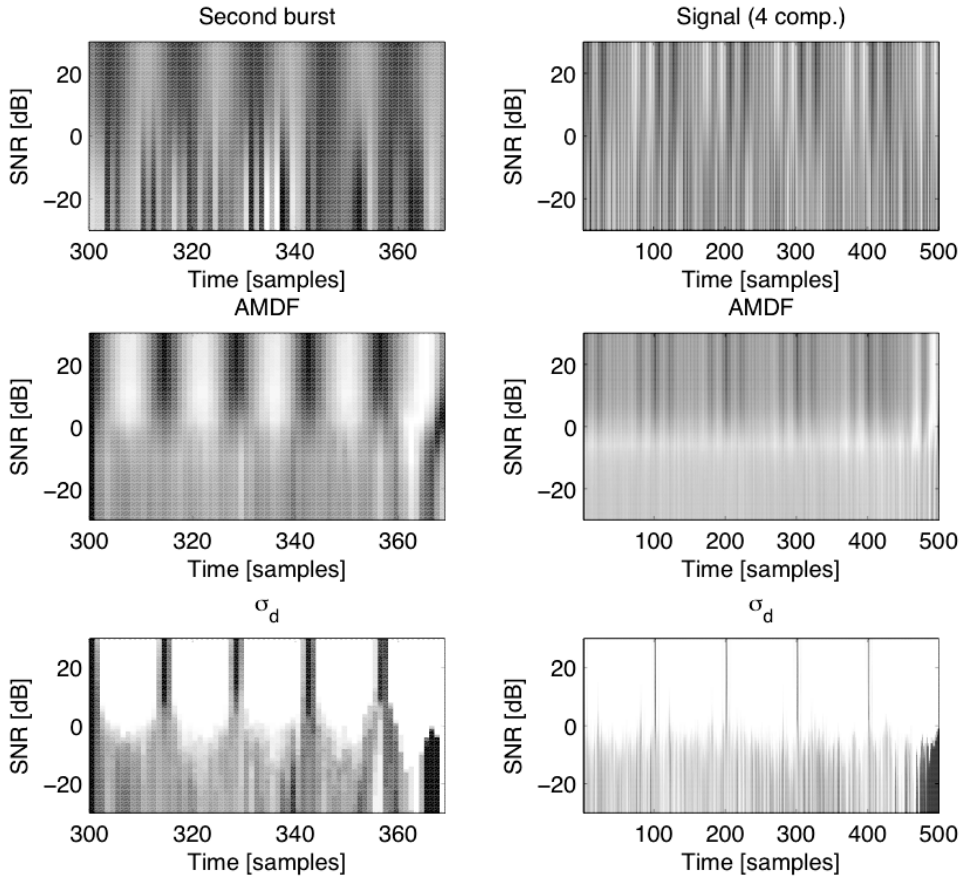


Figure 4.18: The average magnitude difference function (AMDF) and the σ_d measure, computed at multiple signal-to-noise ratios for the second transient in Figure 4.12 (left – $\sigma_d^{(w=7,\varepsilon=\varepsilon_0)}$) and for a periodic signal composed of four sinusoids (right – $\sigma_d^{(w=50,\varepsilon=\varepsilon_0)}$).

let us consider the two transients in the test signal in Figure 4.12. Computing $\sigma_d^{(w=7,\varepsilon=\varepsilon_0)}$ for these two transients gives the results in Figure 4.17. It can be noticed that σ_d has sharp equally spaced maxima, and the distance between these maxima is the fundamental period of the analyzed burst signal. Hence, the estimation of this distance leads to an

estimation of the fundamental frequency.

We show next that σ_d improves AMDF when the w and ε parameters are properly chosen. Figure 4.18 shows a comparative view on the results provided by AMDF and σ_d for two different signals. The first one (left side of the figure) is actually the second transient in the test signal in Figure 4.12, while the second one (right side of the figure) is a periodic signal composed by the superposition of 4 in phase sinusoidal signals of equal amplitudes (and having the following periods (expressed in number of samples): 100, 25, 20, and 5.). We note that unlike σ_d AMDF is represented using white for large values and black for small values, in order to facilitate the visual comparison with σ_d.

It is visually clear that σ_d sharpens the maxima (see plots on the bottom), as well as eliminates the additional local maxima caused by the higher frequency components in the signal (see plots on the right).

4.2.4 Slow-rate chirp identification

Finally, in this subsection we show that the σ_d measure can be used for discriminating short sinusoids from short chirps (i.e. linear frequency modulations). This can be useful, for instance, in distinguishing beaked whale clicks (which are known to be short linear frequency modulations) [Mel08] from dolphin clicks.

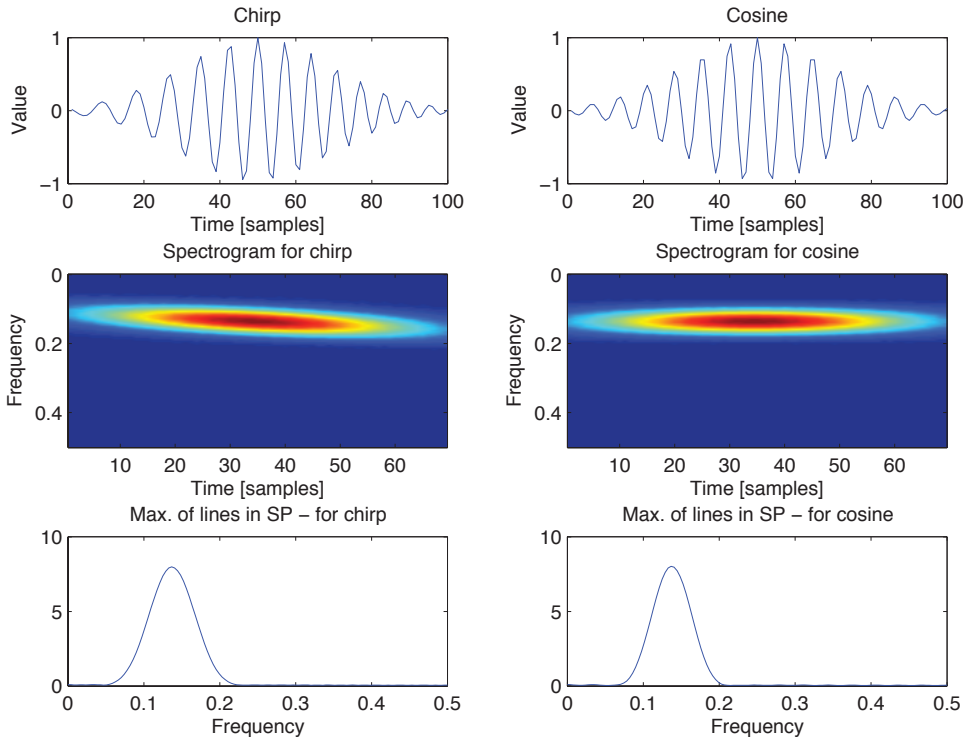


Figure 4.19: Two test signals, together with their spectrograms – a windowed chirp (left), and a windowed cosine (right).

Let us consider the synthetic signals in Figure 4.19. Their spectrograms were computed using windows of 32 samples, with an overlapping of 31 samples. They are almost identical, except that the signal on the left has a very slow variation in its instantaneous

frequency. The question is whether we can computationally distinguish the two signals. The bottom plots show that the lines of maxima in the two spectrograms do not offer the answer to this question.

At SNR = 0 dB things are worse, as the two signals become practically indistinguishable in the spectrogram, as Figure 4.20 shows.

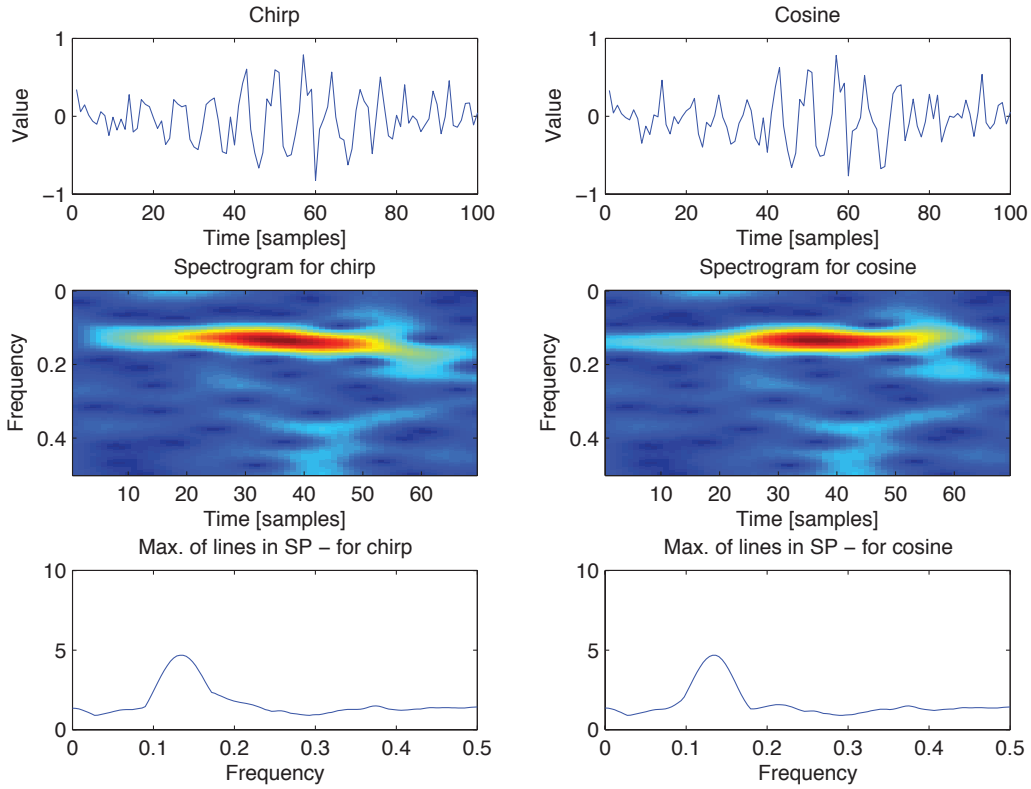


Figure 4.20: The two test signals (chirp (left), and cosine (right)), for SNR = 0 dB.

Given the nature of the two test signals (which are, both of them, special cases of linear frequency modulations), the fractional Fourier transform [OZK01, SS05] would seem to be a solution to our problem here. However, as Figures 4.21 and 4.22 show, the very close slopes of the two signals make it difficult to differentiate them even when no noise is present.

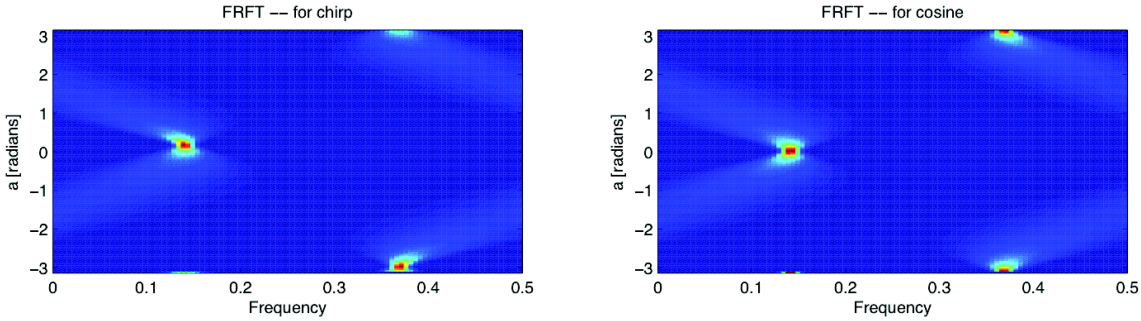


Figure 4.21: The fractional Fourier transforms of the two signals – chirp (left), and cosine (right).

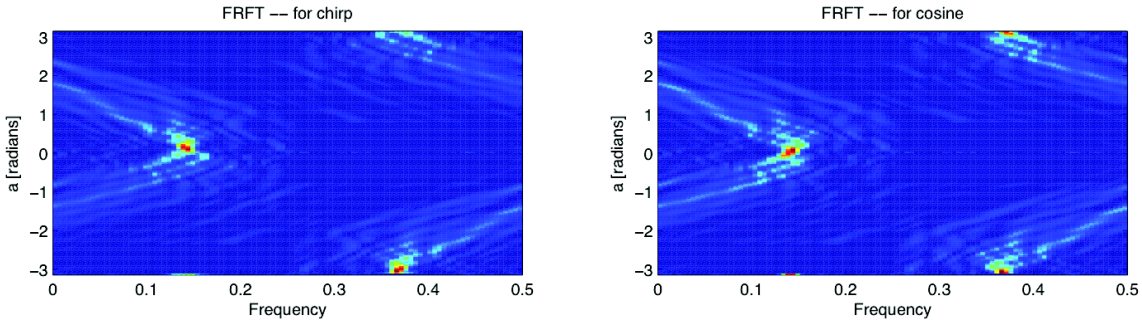


Figure 4.22: The fractional Fourier transforms of the two signals – chirp (left), and cosine (right) (for SNR = 0 dB).

Figure 4.24 illustrates the distance matrices for the two test signals, computed by using vector samples of size $w = 3$, and Euclidean distance. It can be noticed that the diagonal lines in the distance matrix corresponding to the chirp signal are not parallel to the main diagonal, but are slightly inclined.

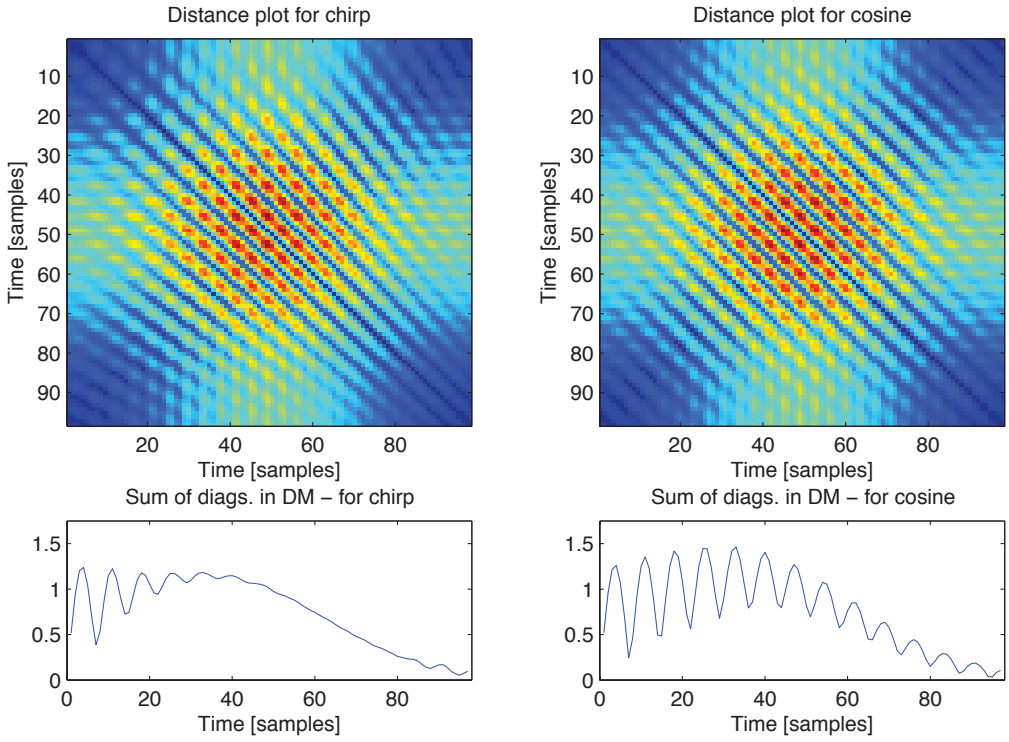


Figure 4.23: The distance matrices corresponding to the two test signals, and the σ_d measures computed by quantifying them.

The bottom plots were obtained by summing diagonal-wise the elements of these matrices (– see the σ_d measure defined in Section 4.1). A visual analysis of these two plots clearly shows that the two signals are very different. The last statement holds true also for noisy signals, as Figure 4.24 shows. It can be noticed that even at a SNR of 0 dB, the σ_d measure (computed on the distance matrix) is able to distinguish the two signals. The noise behaviour of σ_d is illustrated in Figure 4.25, which comparatively shows the two signals, and the corresponding σ_d measures, computed for a range of SNRs between -20 dB and 30 dB.

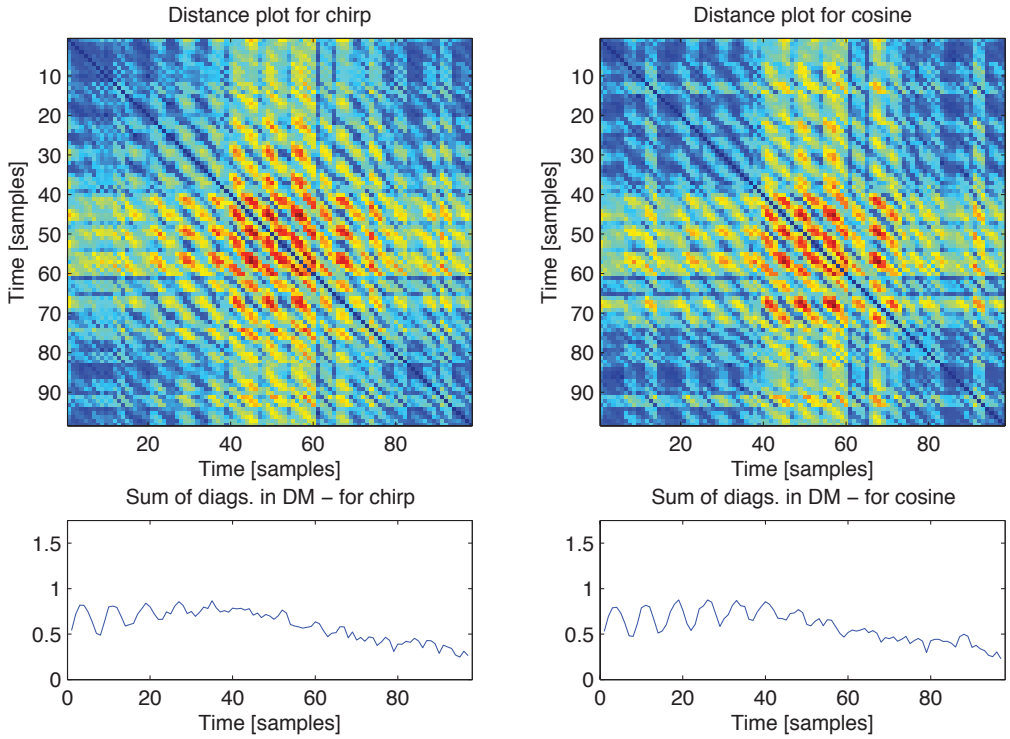


Figure 4.24: The distance matrices corresponding to the two test signals, and the σ_d measures computed by quantifying them (– SNR = 0 dB).

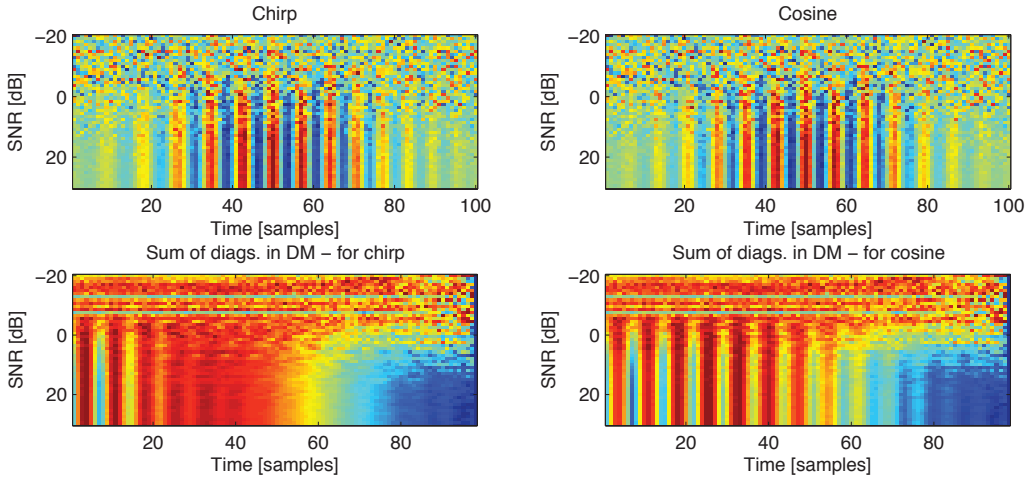


Figure 4.25: The behaviour of the σ_d measures computed for the two test signals, for a range of SNRs.

4.3 Summary

Throughout this chapter we encountered transient signals coming from very different sources, from ultrasonic nondestructive testing of materials to recordings of sounds produced by marine mammals. Regardless of their source, the analysis of transient signals faces similar problems. Our focus here was the detection of the time supports of the transients, and the estimation of some parameters related to their instantaneous frequency. We addressed these problems by making use of the methods developed in the previous

chapter.

Detection of transient signals will also be the main focus in the following chapter, where we address some practical issues concerning ultrasonic investigation of water flow inside pipes.

Chapter 5

Results in ultrasonic investigation of water pipes

In this chapter we put to work the HydroSimUS software application described in Chapter 2, as well as a transient signal detection method introduced in Chapter 3. First, we describe some experimental measurements that we performed on a water pipe and we compare them with the results obtained by offline simulations. This allowed us to highlight the strengths as well as the weaknesses of our software simulator. Then, we make use of HydroSimUS in order to simulate two experimental setups than can be used for estimating the water level in the pipe and the velocity of the water. Finally, we analyze the resulted signals, aiming to obtain the propagation time. We compare our method with a classic state of the art method.

5.1 Experimental signals

The current section presents some experimental ultrasonic measurements and tries to reproduce them in offline mode by using HydroSimUS.

5.1.1 Description of the experiment

The experimental measurements were performed in the experimental facility available at ENSE[3]/Grenoble INP[1]. We had a five centimeters diameter plexiglass pipe filled with water, on which we used two types of ultrasonic transducers: straight beam, and angle beam. The two experimental setups are sketched in Figure 5.1 (for the straight beam transducer) and in Figure 5.2 (for the angle beam transducers).

The signal processing equipment is similar for the two experiments and is described next:

→ The signal generator produces the signal to be emitted.

[1]Ecole Nationale Supérieure de l'Energie, l'Eau et l'Environment / Institut polytechnique de Grenoble

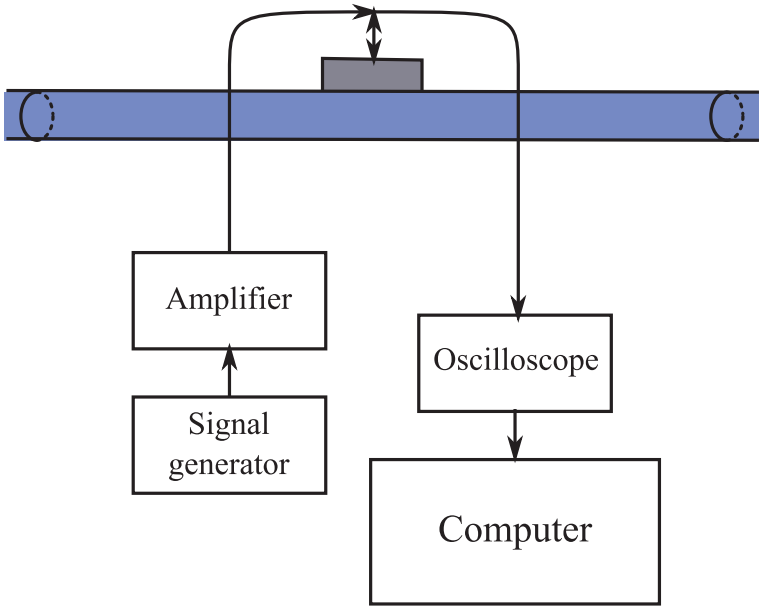


Figure 5.1: Schematic illustration of the first experimental setup – straight beam transducer.

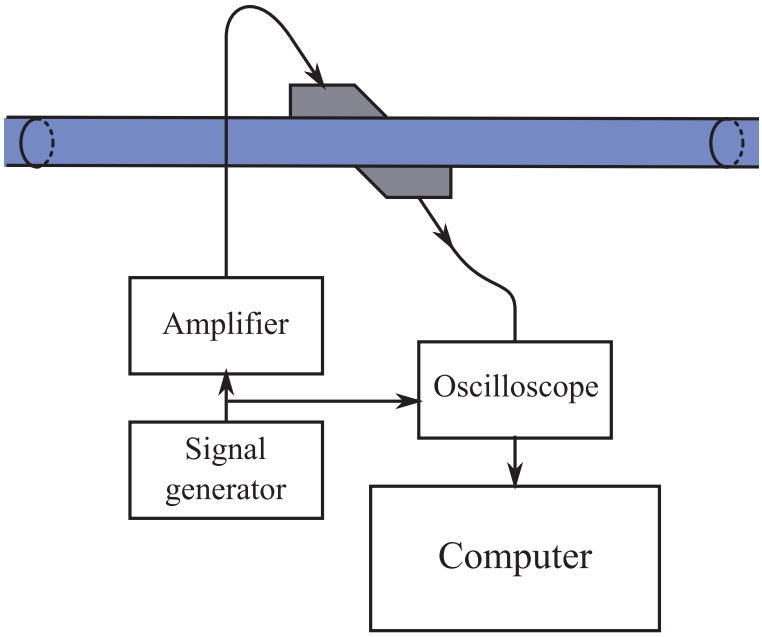


Figure 5.2: Schematic illustration of the second experimental setup – angle beam transducers.

→ This signal is then amplified (by the signal amplifier) in order to bring it to a level that is high enough to resist the attenuation that it will be subject to.

→ The signal reaches the emission transducer, where it is converted from electrical tension to mechanical pressure.

→ The very rapid variation (i.e. about 2 MHz) of mechanical pressure (i.e. the ultrasonic wave) propagates first through the pipe wall, then through the water in the pipe,

then through the opposite wall of the pipe, and then (if applicable, after a few reflections) it finally reaches the reception transducer.

→ The ultrasonic signal reaches the reception transducer (which is the same as the emission transducer, in the case of the first experimental setup), where it is converted back from mechanical pressure variation to voltage variation.

→ The electrical signal is then digitized (at a sampling frequency of 20 MSa/s and a resolution of 12 bits) and sent to the computer, where it can be visualized, stored, and subsequently processed.

Figure 5.3 shows some pictures that were taken while performing the experimental measurements described in this section.

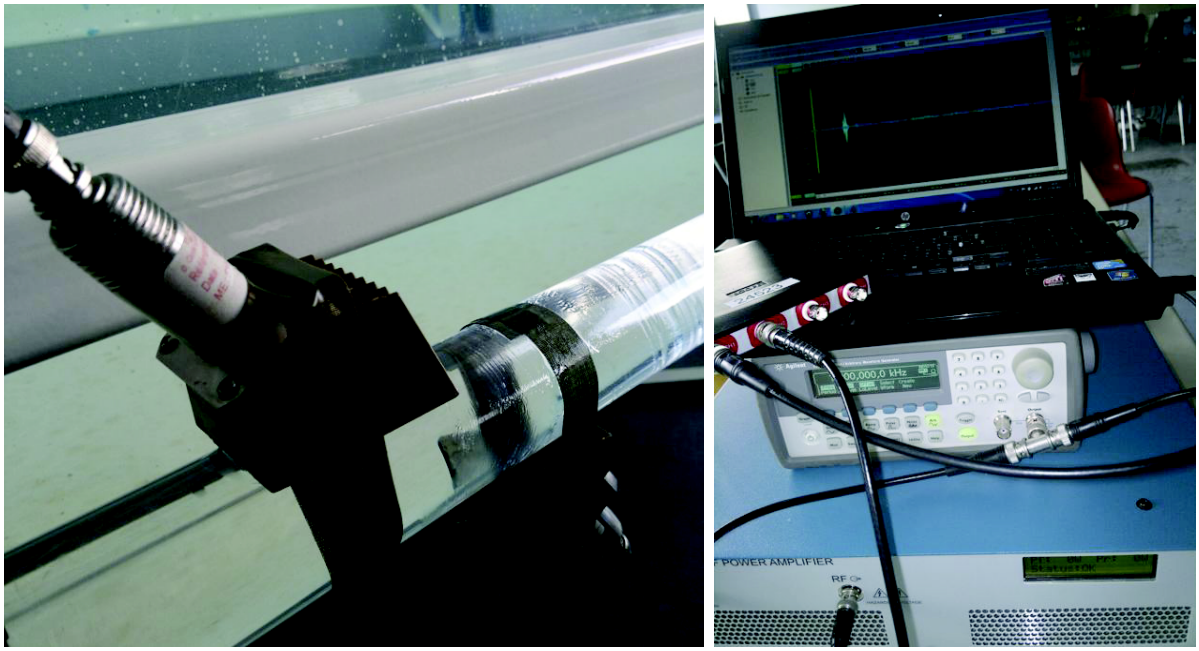


Figure 5.3: Pictures taken while performing the experimental measurements.

The first experiment

For the straight beam transducer measurement we used a sinusoidal signal with a frequency of 2.25 MHz and a duration of two cycles. The received signal is shown in Figure 5.4. It contains four clearly visible reflections, that we noted by R_1, R_2, R_3, and R_4. The source of these reflections can be found by taking into consideration the section of the pipe (which is illustrated in Figure 5.5) and then by following the wave along its path. Hence, it is clear that R_1 is the reflection at the interface point P_0. It is also quite clear that R_2 is the reflection generated at P_1 and that R_3 is the reflection at P_2.

Knowing that the diameter of the pipe is $D = 50$ mm and that the thickness of the pipe wall is $G = 5$ mm, we can use the start times of the reflections in the signal in Figure 5.4 to estimate the speed of the acoustic wave in water, c_{water}, and in the pipe wall, c_{wall}.

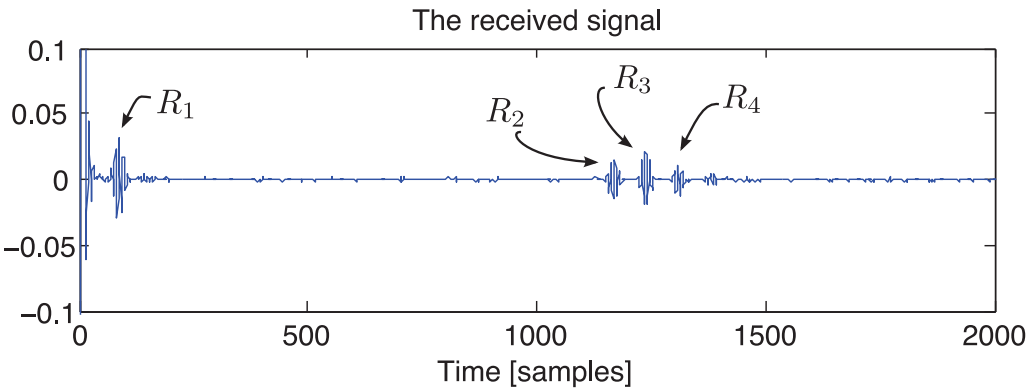


Figure 5.4: The received signal obtained for the straight beam transducer experiment.

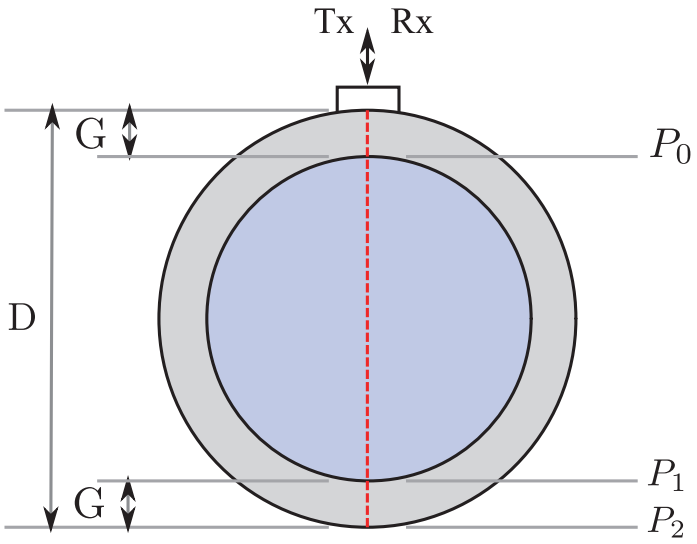


Figure 5.5: The the pipe section, for the straight beam transducer experiment.

As transient R_1 comes from the reflection at P_0, its propagation time is twice the propagation time of the acoustic wave through the pipe wall, that is:

$$t_{\text{propag}(R_1)} = 2 \cdot t_{\text{wall}}, \tag{5.1}$$

which means that the propagation time through the pipe wall is:

$$t_{\text{wall}} = \frac{1}{2} \cdot t_{\text{propag}(R_1)}. \tag{5.2}$$

But it is known that:

$$t_{\text{wall}} = \frac{G}{c_{\text{wall}}}. \tag{5.3}$$

From the last two equations it follows that the speed of sound in the pipe wall is:

$$c_{\text{wall}} = \frac{G}{\frac{1}{2} \cdot t_{\text{propag}(R_1)}}. \tag{5.4}$$

By manually estimating $t_{\text{propag}(R_1)}$ to be equal to 70 samples times the sampling period T_{samp} ($= 1/20$ μs) and making the computations we obtain that $c_{\text{wall}} = 2857.1\,\text{m/s}$. It

comes naturally the question of how the value obtained for c_{wall} is affected by the precision we had in estimating $t_{\text{propag}(R_1)}$. For $t_{\text{propag}(R_1)}$ equal to $69 \cdot T_{\text{samp}}$ we obtain $c_{\text{wall}} = 2898.6$ m/s, while for $t_{\text{propag}(R_1)}$ equal to $71 \cdot T_{\text{samp}}$ we obtain $c_{\text{wall}} = 2816.9$ m/s. Hence, the precision of $t_{\text{propag}(R_1)}$ has a strong impact on the value obtained for c_{wall}. More precisely, an additive error ξ on $t_{\text{propag}(R_1)}$ leads to an absolute error of $(2 \cdot G \cdot \xi) / (t \cdot (t + \xi))$ on c_{wall}.

As we mentioned before, R_2 comes from the reflection at interface P_1. This means that:

$$t_{\text{propag}(R_2)} = 2 \cdot t_{\text{wall}} + 2 \cdot t_{\text{water}}, \tag{5.5}$$

which means that the propagation time through the water (from top to bottom) is:

$$t_{\text{water}} = \frac{1}{2} \cdot \left(t_{\text{propag}(R_2)} - 2 \cdot t_{\text{wall}} \right). \tag{5.6}$$

But it is known that:

$$t_{\text{water}} = \frac{D - 2 \cdot G}{c_{\text{water}}}. \tag{5.7}$$

From the last two equations it follows that the speed of sound in the water inside the pipe is:

$$c_{\text{water}} = \frac{D - 2 \cdot G}{\frac{1}{2} \cdot \left(t_{\text{propag}(R_2)} - 2 \cdot t_{\text{wall}} \right)}, \tag{5.8}$$

where

$$t_{\text{wall}} = \frac{1}{2} \cdot t_{\text{propag}(R_1)}. \tag{5.9}$$

By manually estimating $t_{\text{propag}(R_2)}$ to be equal to $1152 \cdot T_{\text{samp}}$ and $t_{\text{propag}(R_1)}$ to be equal to $70 \cdot T_{\text{samp}}$ and by making the computations we obtain that $c_{\text{water}} = 1478.7\,\text{m/s}$. The same as for c_{wall}, the value obtained for c_{water} is affected by the precision we had in estimating the propagation times. However, we note that the value we obtained for c_{water} corresponds well to the theoretical value of sound speed in water at 20° C, which is 1480 m/s.

The second experiment

For the angle beam transducer experiment we used transducers with an angle of $\alpha = 35°$. We performed measurements in three different configurations, that can be seen in Figure 5.6. We have also sketched in the figure the paths of the acoustic waves. However, these paths are only approximate, as in practice deviations may occur due to the refractions produced at the interfaces between water and the pipe walls. And we must also take into consideration the fact that the ultrasonic beam is not perfectly linear, but has a certain spreading angle.

We emitted a 2 MHz frequency sinusoidal signal of 5 μs duration. The received signals are shown in Figure 5.7. It can be noticed that for the signal measured in the second configuration we receive a lower energy transient before the actual arrival of the emitted signal. It is due to the spreading angle of the emission beam, which makes it possible for a fraction of the signal energy to propagate directly (i.e. without any reflections) from the emitter to the receiver.

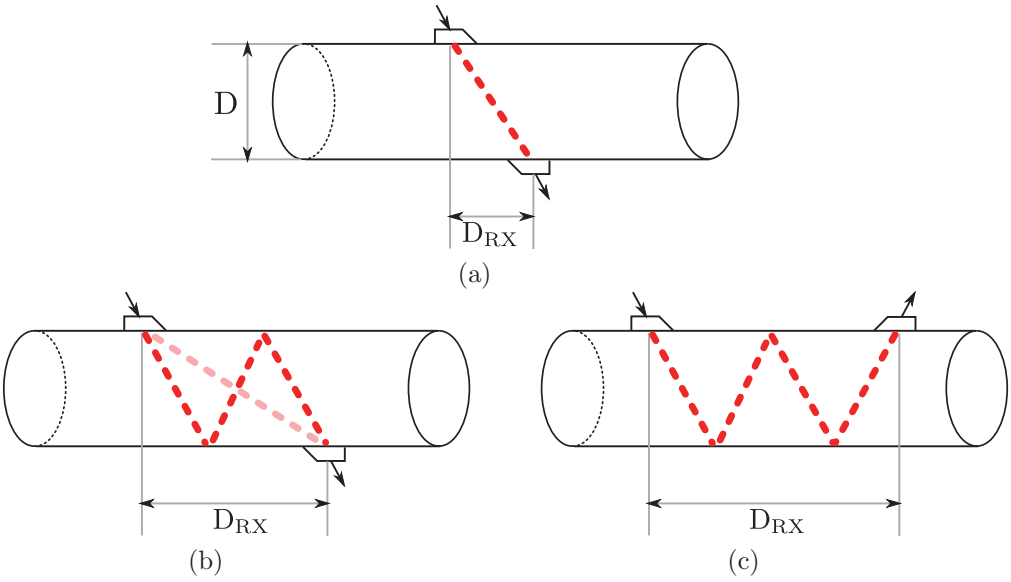


Figure 5.6: The three measurement configurations used in the angle beam transducers experiment. D= 50 mm. (a) D_{RX} = 35 mm. (b) D_{RX} = 60 mm. (c) D_{RX} = 90 mm.

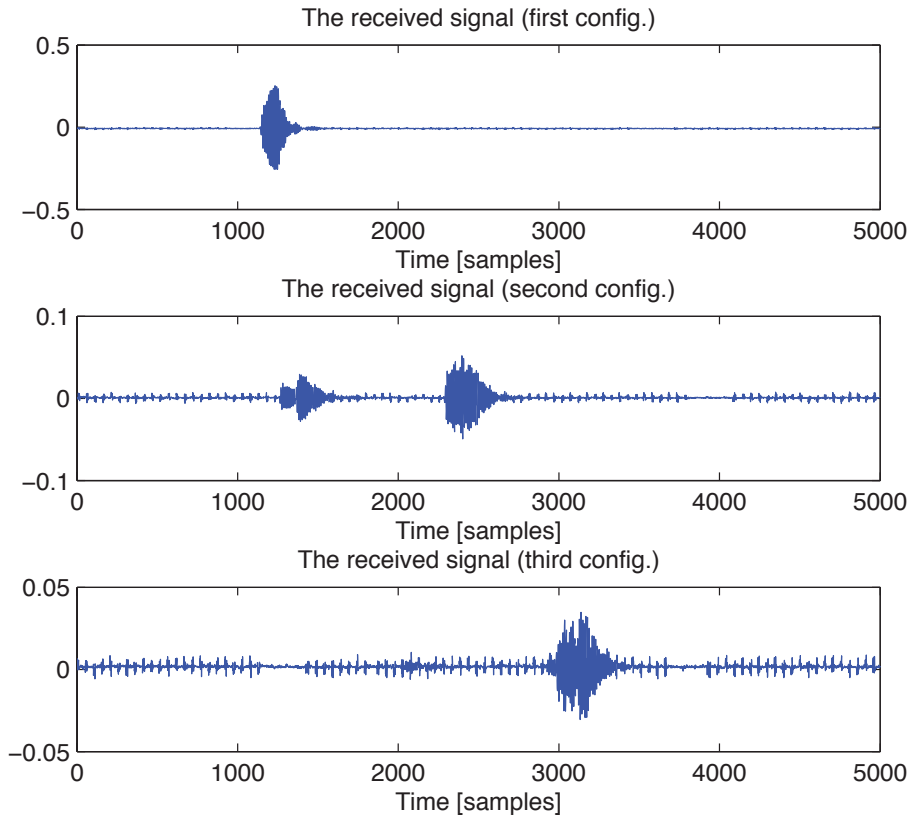


Figure 5.7: The received signals obtained for the angle beam transducer experiment.

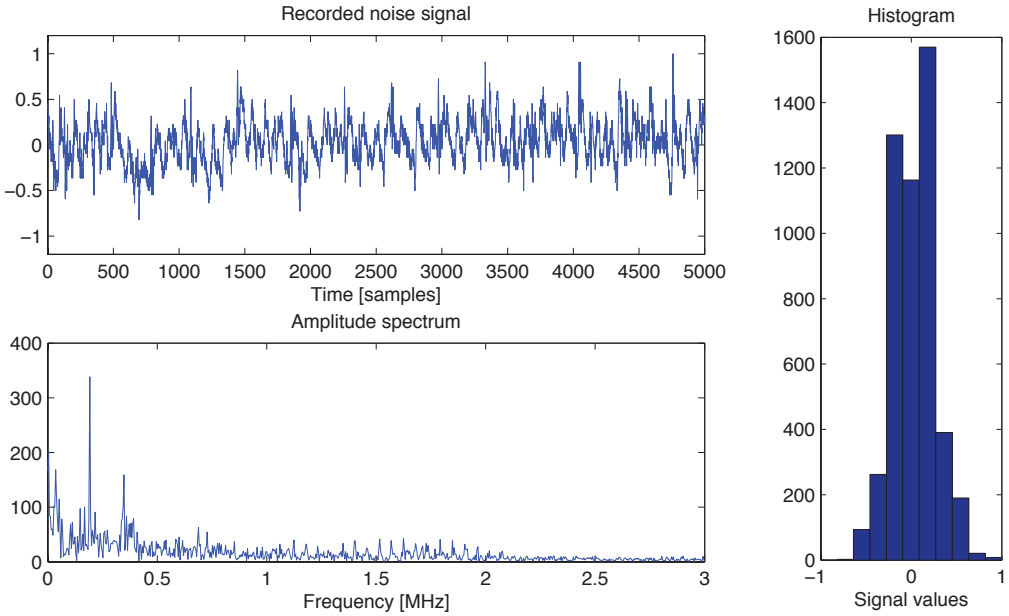


Figure 5.8: The recorded noise, its amplitude spectrum, and its histogram.

Besides, in order to have recordings of real noise, we also recorded the signal that was "heard" by the receiver when no signal was emitted. As it can be seen in Figure 5.8, this noise is colored and Gaussian.

5.1.2 Simulation with HydroSimUS

Simulation of the first experiment

Figure 5.9 shows the parameters that we used for simulating the first of the two experiments discussed so far. For the speed of sound, c, we used the values shown in Figure 5.10. (We took into consideration the computations we made in the previous subsection.)

We recall next the constraints that must be met when choosing the time and space meshing steps, i.e. Δt, Δx, and Δy:

$$\begin{cases} \Delta x \ll \lambda \\ \Delta t \leq T_{\text{samp}} \\ c \cdot \Delta t < \Delta x \end{cases} . \tag{5.10}$$

(For Δy the same constraints apply as for Δx.)

In order to reduce the size of the matrices involved in performing the computations, Δx should be as large as possible. (And, of course, Δy as well.) But, as discussed in Chapter 2, in order to perform a valid simulation Δx should be much less than the wavelength of the emitted signal. As the wavelength of the signal is:

$$\lambda = \frac{c}{f}, \tag{5.11}$$

(where c is the speed of sound and f is the emitted signal frequency), is means that for the pipe wall λ is $\lambda_{\text{wall}} = 2.85$ mm, and for the water inside the pipe it is $\lambda_{\text{water}} = 1.49$ mm.

Simulation paramters:

Parameter	Description	Value
Δx	Space meshing step on the j axis	0.05 mm
Δy	Space meshing step on the i axis	0.05 mm
N_1	Number of mesh points on i axis	1000
N_2	Number of mesh points on j axis	1000
N_{tx}	Number of punctual emitters	20
D_{tx}	Number of points between emitters	4
Δt	Time meshing step	0.01 μs
N_t	Number of time steps in simulation	10000
f_{sig}	Frequency of the emitted signal	1 MHz
N_{cyc}	Number of cycles in the emitted signal	1

Positions of the transducers:

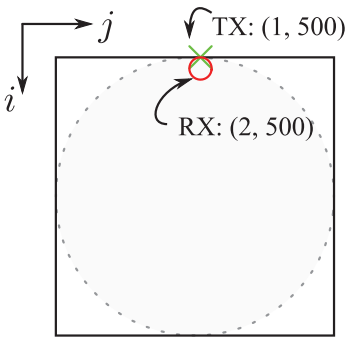


Figure 5.9: The HydroSimUS parameters used for simulating the first experiment.

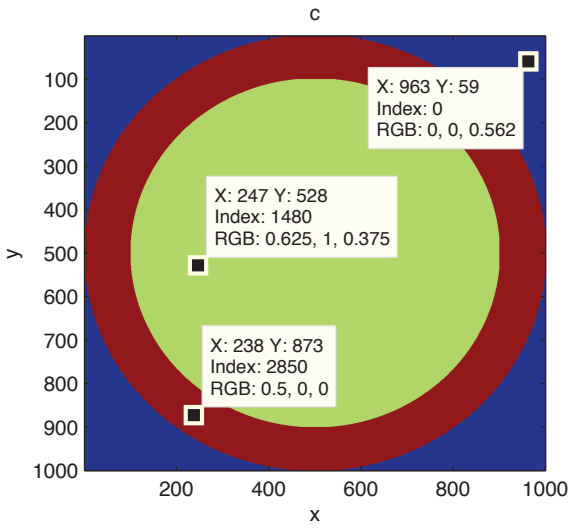


Figure 5.10: The velocity of the acoustic wave in each point of the simulation section, for the first experiment.

We obtained these values by considering that $f = 1$ MHz instead of the 2.25 MHz we used in the real experiment. By reducing in simulation the frequency of the emitted signal we could obtain larger values for λ, which allowed us to choose larger values for Δx and Δy (which leads to smaller size matrices, thus leading to faster simulations). Hence, we emitted in simulation one cycle sinusoidal signal of 1 MHz amplitude.

We chose $\Delta x = \Delta y = 0.05$ mm, which is much less than the values obtained for λ. By taking into consideration that D = 50 mm, we obtained matrices of 1000x1000 elements, i.e. $N_1 = N_2 = 1000$. As G = 5 mm, it means that the pipe wall is meshed into 100 points.

As for the time step Δt, it should be less than the sampling period T_{samp} (which is equal to $1/20\,\mu$s) and less than $\Delta x/c$ (which is equal to $1.75 \cdot 10^{-2}\mu$s for the pipe wall, and is equal to $3.38 \cdot 10^{-2}\,\mu$s for the water inside the pipe). We chose for the time meshing step the value $\Delta t = 0.01\,\mu$s.

Given that $T_{\text{samp}}\Delta t = 5$, in order to have the same number of samples in the simulated signal as in the experimental signal, i.e. 2000 samples, the number of simulation time steps must be 5 times greater (i.e. N_t=10000).

In Figure 5.11 we show some snapshots taken during the simulation of the wave propagation for the first experiment. It can be noticed how the reflections are produced at the interfaces water - pipe wall and pipe wall - air.

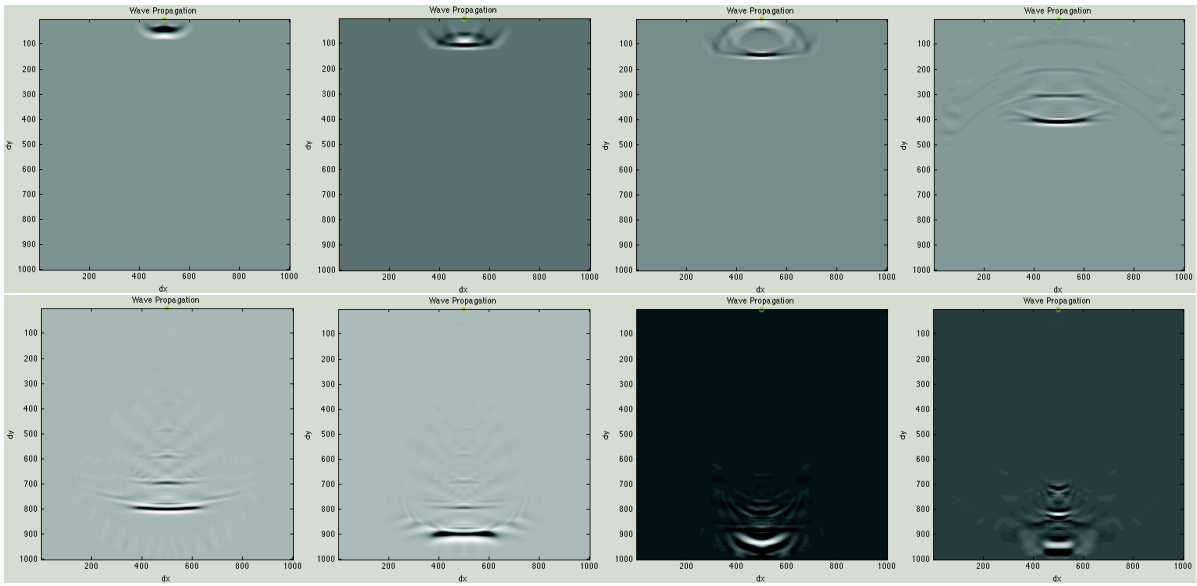


Figure 5.11: Snapshots of wave propagation for the simulation of the first experiment. (The time advances from left to right, and from top to bottom.)

The received signal is shown in Figure 5.12. The experimental signal is shown as well, for comparison purpose.

It can be noticed that we obtained in simulation the same propagation times for the different reflections. However, the amplitudes of the transients are not exactly the same. Besides, in the areas between the transients the signal the simulated signal looks noisier. This is due to the fact that the wave propagation equation did not take into account the attenuation of the wave, which is why the signal contains many reflections (which are highly attenuated in the experimental signal). This is also the reason why the last transients have higher amplitudes – as the acoustic wave is not properly attenuated, constructive interferences occur between reflexions that propagate on different paths. Another reason for the differences between the simulated and the experimental signal is that the directional emitter implemented in HydroSimUS models the real emitter only partially.

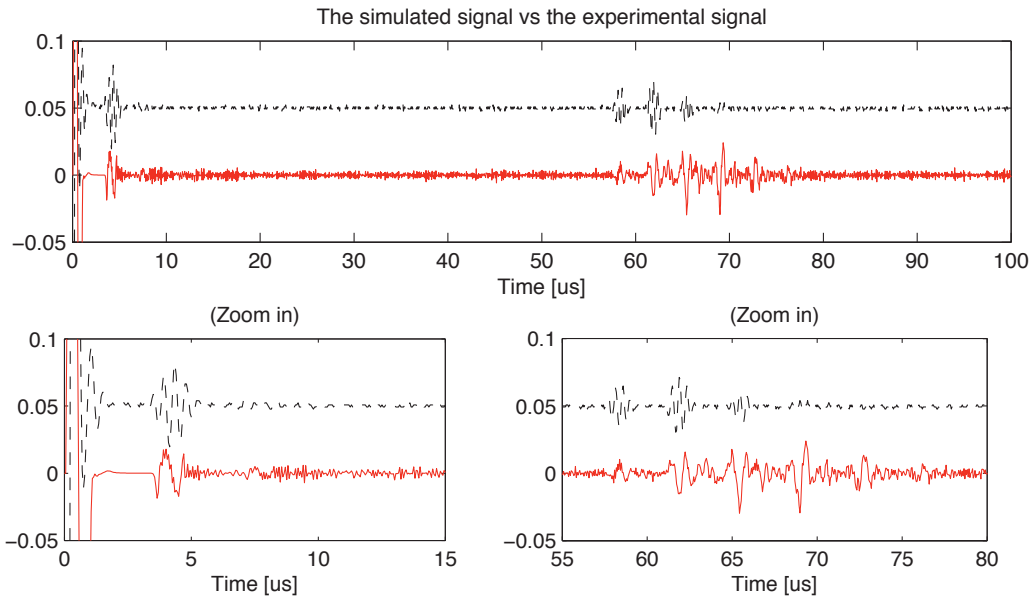


Figure 5.12: The simulated signal, for the first experiment. (The experimental signal is shown with dotted line.)

Nevertheless, we note that the propagation times obtained in simulation are very similar to those measured experimentally. In addition to this, we note that HydroSimUS allows to visualize the acoustic wave during its propagation. Hence, the different reflections can be followed individually. This can be useful when studying complex configurations, where it is difficult to predict the sources of the different reflections that appear in the recorded signal. For instance, in the case of the signal recorded during our first experiment we could have not figure out where the transients R_1, R_2, R_3 and R_4 come from. However, the visualization of the wave propagation (see Figure 5.11) makes things clearer.

Simulation of the second experiment

In order to simulate the second of the two experiments we can use a longitudinal section in the pipe. The length of the pipe section must be at least equal to the maximum distance between the two transducers (the emitter, and the receiver). Ideally, this length should be as long as possible, in order for the reflexions produced at the ends of the pipe section to affect as little as possible the simulation results. We would require, therefore, matrices of at least 1000x5000 elements, which would lead to heavy computations and slow simulations. However, in order to be able to illustrate this kind or simulation, we decided to perform the simulation on a scaled configuration (ten times smaller). The parameters we used for simulation are shown in Figure 5.13, and the values of the sound speed are shown in Figure 5.14.

The same as for the previous simulation, we used a signal of $f = 1$ MHz and we chose $\Delta t = 0.01\,\mu$s. We used a number of simulation time steps of $N_t = 2500$, in order for the duration of the simulation to correspond to the aquisition duration.

Figure 5.15 shows how the wave propagates in time, and Figure 5.16 shows the simulated signals recorded in the three reception points.

Simulation paramters:

Parameter	Description	Value
Δx	Space meshing step on the j axis	0.05 mm
Δy	Space meshing step on the i axis	0.05 mm
N_1	Number of mesh points on i axis	100
N_2	Number of mesh points on j axis	1000
N_{tx}	Number of punctual emitters	5
D_{tx}	Number of points between emitters	8
Δt	Time meshing step	0.01 μs
N_t	Number of time steps in simulation	2500
f_{sig}	Frequency of the emitted signal	1 MHz
N_{cyc}	Number of cycles in the emitted signal	1

Positions of the transducers:

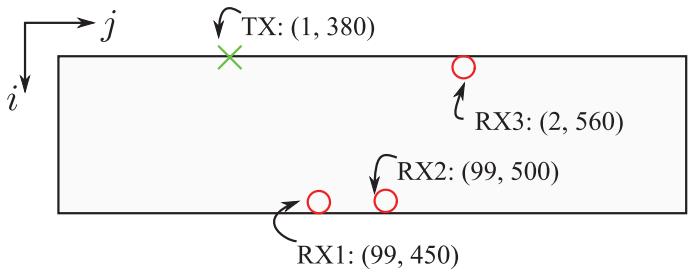


Figure 5.13: The HydroSimUS parameters used for simulating the second experiment.

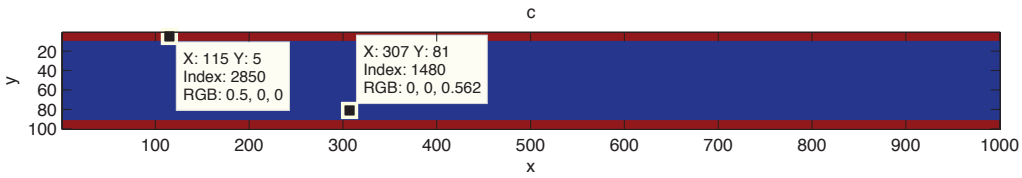


Figure 5.14: The velocity of the acoustic wave in each point of the simulation section, for the second experiment.

The first observation is that the simulated signals contain more transients than the experimental signals. The reason is the same as in the case of the previous simulation, namely the fact that we did not consider any attenuation in the wave propagation equation. It can be also noted that in the simulated signals the propagation times differ from those in the measured signals. This is caused by the fact that we did not appropriately model neither the acoustic wave propagation in the pipe walls (but we used the same equation as for the propagation in water), nor the phenomena that occur at the interfaces between different propagation media.

Further, we make use of HydroSimUS for performing some simulations that were experimentally unavailable.

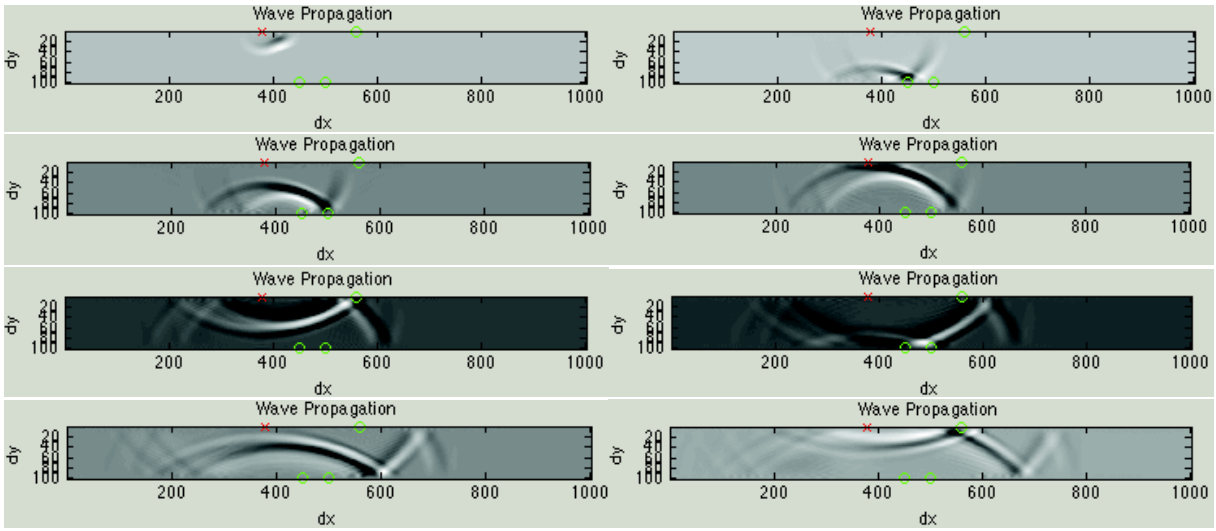


Figure 5.15: Snapshots of wave propagation for the simulation of the second experiment. (The time advances from left to right, and from top to bottom.)

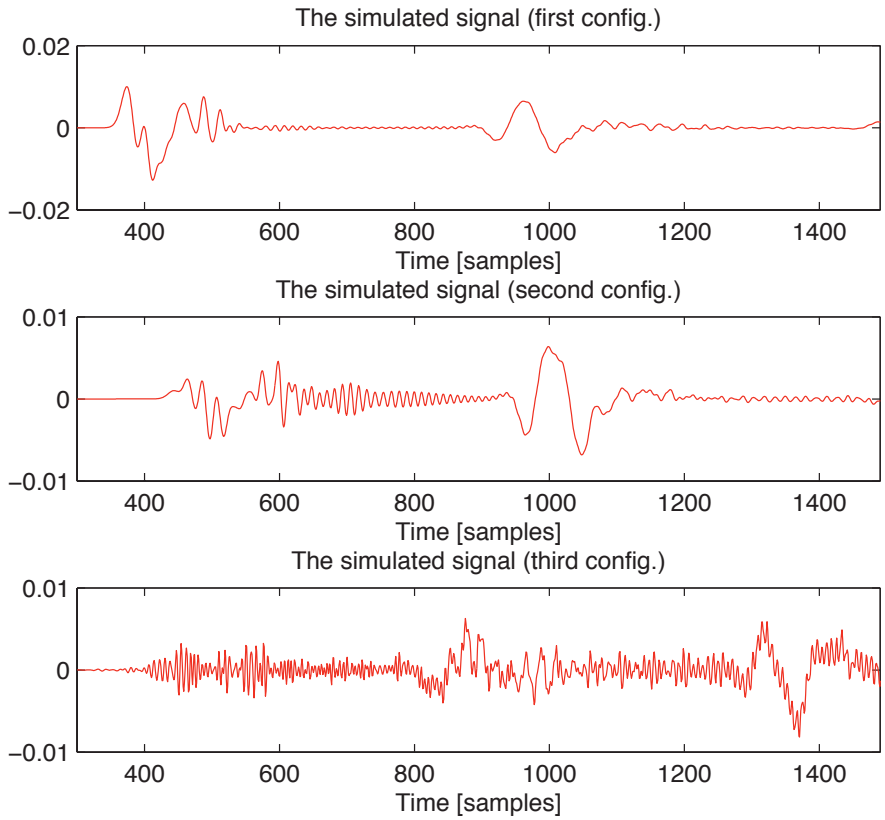


Figure 5.16: The received signals obtained for the angle beam transducer experiment. (The real signals were shown in Figure 5.7.)

5.2 Synthetic signals

In this section we show how HydroSimUS can be useful for simulating experimental configurations used in measuring the level and the velocity of water inside a pipe.

5.2.1 Water level

Let us consider a pipe having the dimensions of the pipe we used in the experimental configurations presented in the previous section, i.e. D = 50 mm and G = 5 mm. We take $c_{\text{water}} = 1480$ m/s and $c_{\text{wall}} = 2850$ m/s, and we consider the level of the water in the pipe to be equal to 7.5 mm. We intend to simulate with HydroSimUS a water level measurement scheme based on placing an ultrasonic transducer on the bottom of the pipe wall. This transducer will act as both an emitter and a receiver – it will first emit a 1 MHz sinusoidal signal through the pipe and then it will listen for the reflections. The simulation parameters are shown in Figure 5.17.

Simulation paramters:

Parameter	Description	Value
Δx	Space meshing step on the j axis	0.05 mm
Δy	Space meshing step on the i axis	0.05 mm
N_1	Number of mesh points on i axis	1000
N_2	Number of mesh points on j axis	1000
N_h	Level of water (on j axis)	250
N_{tx}	Number of punctual emitters	20
D_{tx}	Number of points between emitters	4
Δt	Time meshing step	0.01 μs
N_t	Number of time steps in simulation	10000
f_{sig}	Frequency of the emitted signal	1 MHz
N_{cyc}	Number of cycles in the emitted signal	1

Positions of the transducers:

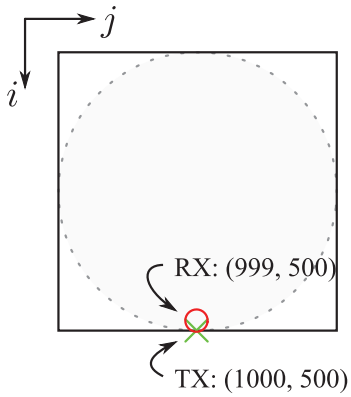


Figure 5.17: The HydroSimUS parameters used for simulating the water level measurement configuration.

Figure 5.18 shows some snapshots taken during the propagation of the wave. The received signal is shown in Figure 5.19. The first transient (around sample number 500) is the reflection that occured at the interface between the pipe wall and the water in the

pipe, and the second transient (around sample number 1500) is the reflection that occured at the top of the water, i.e. at the interface water-air.

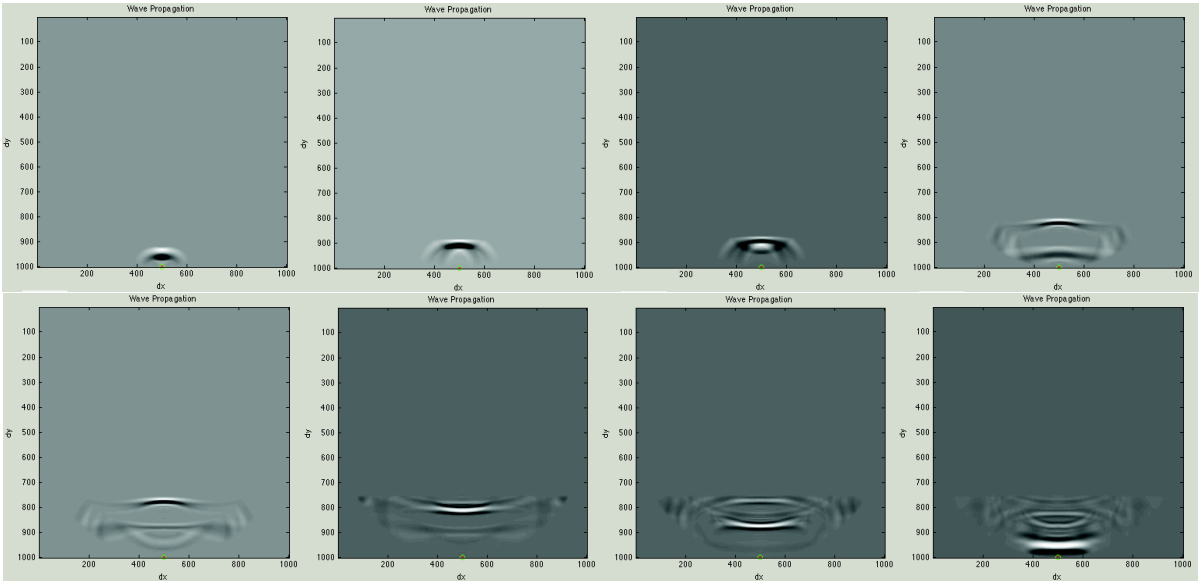


Figure 5.18: Snapshots of wave propagation for the simulation of the water level measurement configuration. (The time advances from left to right, and from top to bottom.)

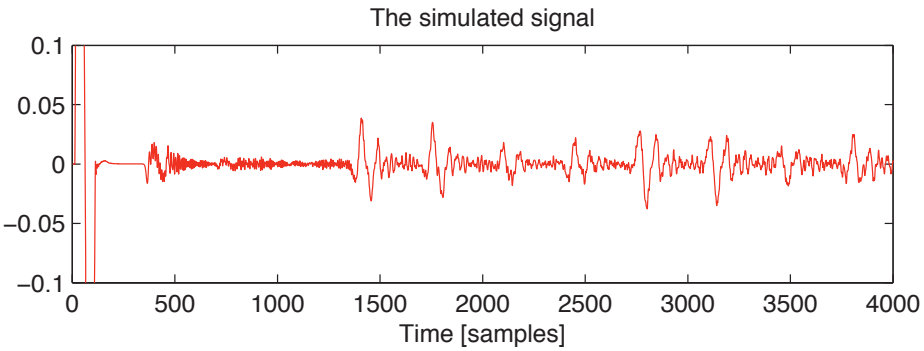


Figure 5.19: The simulated signal obtained for the water level measurement configuration.

By visually estimating the propagation times of these two transients, we obtain that:

$$t_{\text{propag}(1)} = 350 \cdot \Delta t \tag{5.12}$$

and

$$t_{\text{propag}(2)} = 1350 \cdot \Delta t. \tag{5.13}$$

Knowing that $\Delta t = 0.01, \mu$s, we obtain that $t_{\text{propag}(1)} = 3.5\,\mu$s and $t_{\text{propag}(2)} = 13.5\,\mu$s. The difference between these two times is twice the propagation time through the water column, that is:

$$t_{\text{propag}(2)} - t_{\text{propag}(1)} = \frac{2 \cdot d_{\text{water}}}{c_{\text{water}}}. \tag{5.14}$$

This equation allows us to obtain the height of the water, which is:

$$d_{\text{water}} = \frac{1}{2} \cdot c_{\text{water}} \cdot \left(t_{\text{propag}(2)} - t_{\text{propag}(1)}\right). \tag{5.15}$$

After performing the computations we obtain for the water level the value:

$$d_{\text{water}} = 7.4\,\text{mm}, \tag{5.16}$$

which is close to the desired value, i.e. 7.5 mm.

We notice, however, that the signal contains many reflections, which can make its analysis difficult to perform. The cause of this reflections was discussed in the previous section. Now we repeat the simulations using the same parameters, except that we consider that the pipe wall is extremely thin, i.e. G = 0 mm. The propagation of the acoustic wave in this case is shown in Figure 5.20, and the received signal is shown in Figure 5.21.

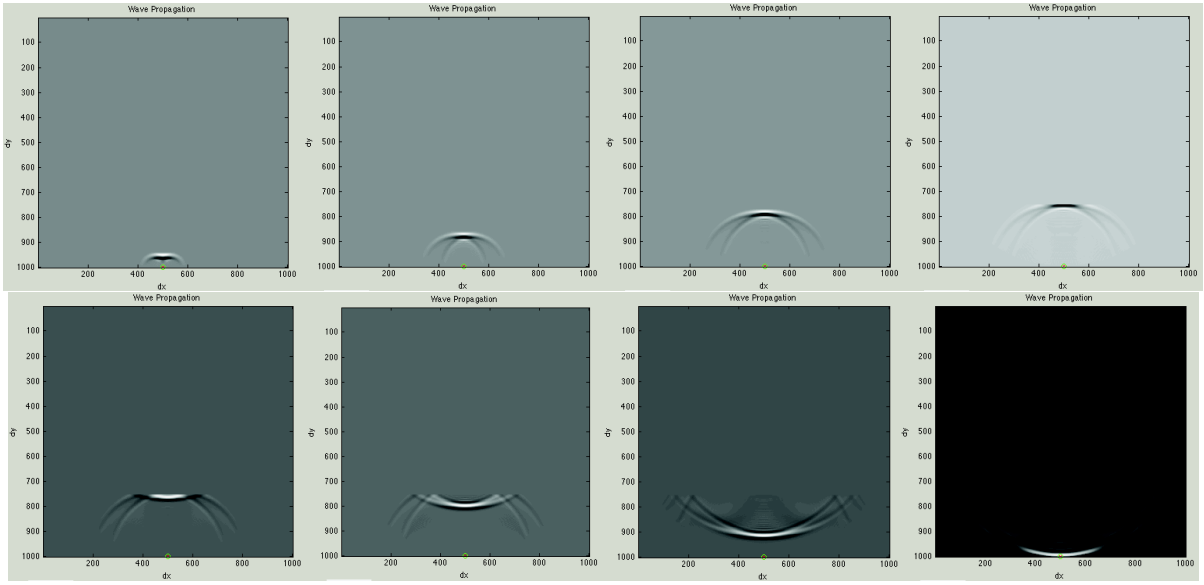


Figure 5.20: Snapshots of wave propagation for the simulation of the water level measurement configuration, in the no pipe walls case. (The time advances from left to right, and from top to bottom.)

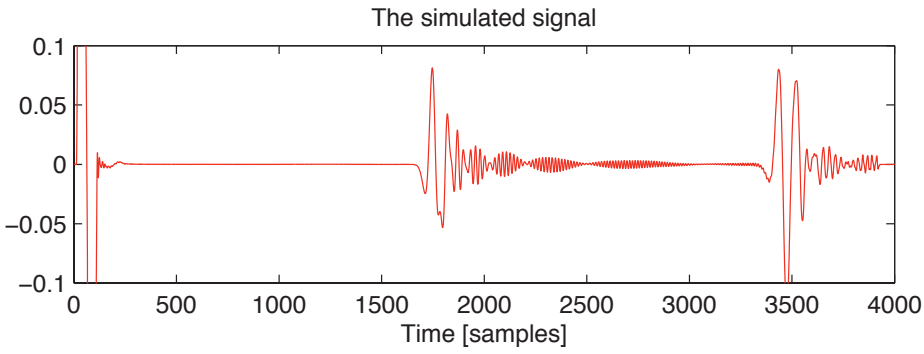


Figure 5.21: The simulated signal obtained for the water level measurement configuration, in the no pipe walls case.

In this case we obtain, as expected, in the simulated signal only the reflection that was produced at the interface water-air. By measuring (visually) the propagation time of this reflection (i.e. the first transient in the signal), we obtain:

$$t_{\text{propag}} = 1670 \cdot \Delta t. \tag{5.17}$$

As this time is the time required for the emitted signal to propagate twice through the water column, i.e.

$$t_{\text{propag}} = \frac{2 \cdot c_{\text{water}}}{c_{\text{water}}}. \tag{5.18}$$

we obtain that the water level can be computed with:

$$d_{\text{water}} = \frac{1}{2} \cdot c_{\text{water}} \cdot t_{\text{propag}}. \tag{5.19}$$

Performing the computations give a water level of:

$$d_{\text{water}} = 12.4\,\text{mm}. \tag{5.20}$$

Hence, we obtained a water level that is about 5 millimeters higher than the one obtained in the previous case (where we considered the pipe walls as well). This difference is exactly the 5 millimeter pipe wall that is missing in the current simulation.

We showed so far how HydroSimUS can be used for simulating a pipe water level measurement configuration, by using a transversal section simulation. Further, we show how we can use a longitudinal section simulation for perfoming the simulation of a water velocity measurement configuration.

5.2.2 Water velocity

Let us now consider a pipe with a diameter of D = 30 mm. We neglect the thickness of the pipe wall. We consider that the angle of the ultrasonic transducers is $\alpha = 45°$, the speed of sound in water is $c_{\text{water}} = 1480$ m/s, and the frequency of the emitted signal is $f = 1$ MHz. The whole set of simulation parameters is shown in Figure 5.22.

Wave propagation is shown in Figure 5.23, and the received signal is shown in Figure 5.24. As in this case the water inside the pipe does not flow, the signal that propagates in the opposite direction (i.e. from the receiver to the emitter) is the same as the direct signal.

Next, we consider the case where water flows with a certain speed. Then, we compute the water velocity by using the difference between the propagation time of the direct signal and the propagation time of the reverse signal [Lyn89]. To begin, we consider that water flows (from right to left) with a constant speed (on the x axis) in every mesh point. We denote this velocity by $\vec{v}$ (see Figure 5.25).

In the direct configuration (i.e. from top to bottom) the acoustic wave propagates with the speed $c - v \cdot \sin\alpha$, while in the opposite direction (i.e. from bottom to top) it propagates with the speed $c + v \cdot \sin\alpha$. It means that the propagation times are:

$$\begin{cases} t_{\text{propag(d)}} = d_{\text{water}}/(c - v \cdot \sin\alpha) \\ t_{\text{propag(r)}} = d_{\text{water}}/(c + v \cdot \sin\alpha) \end{cases}. \tag{5.21}$$

Simulation paramters:

Parameter	Description	Value
Δx	Space meshing step on the j axis	0.1 mm
Δy	Space meshing step on the i axis	0.1 mm
N_1	Number of mesh points on i axis	300
N_2	Number of mesh points on j axis	900
c	Speed of acoustic wave	1480 m/s
N_{tx}	Number of punctual emitters	20
D_{tx}	Number of points between emitters	4
Δt	Time meshing step	0.01 μs
N_t	Number of time steps in simulation	10000
f_{sig}	Frequency of the emitted signal	1 MHz
N_{cyc}	Number of cycles in the emitted signal	1

Positions of the transducers:

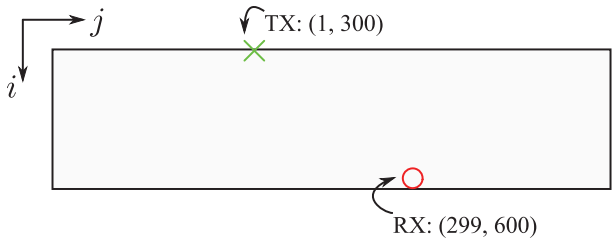


Figure 5.22: The HydroSimUS parameters used for simulating the water velocity measurement configuration (for the $v = 0$ m/s case).

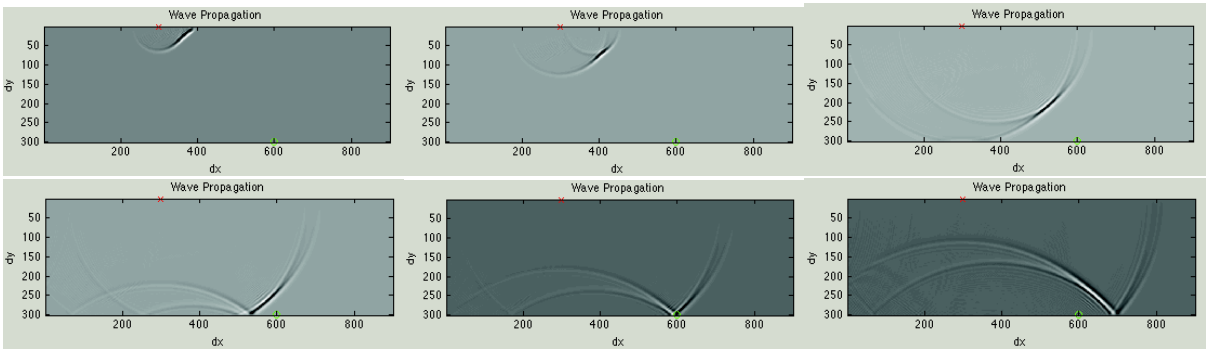


Figure 5.23: Snapshots of wave propagation for the simulation of the water velocity measurement configuration (for the $v = 0$ m/s case). (The time advances from left to right, and from top to bottom.)

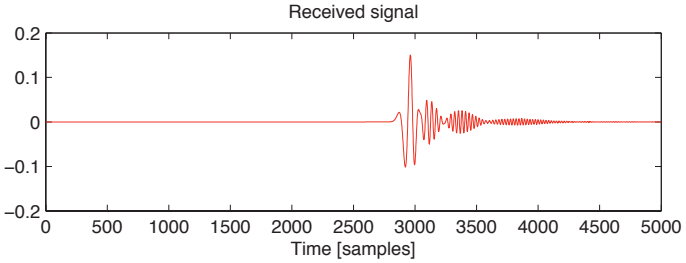


Figure 5.24: The received signal obtained for the simulation of the water velocity measurement configuration (for the $v = 0$ m/s case).

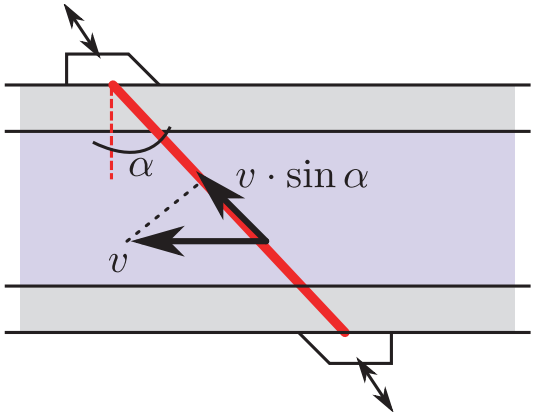


Figure 5.25: Illustration of the influence of water flow on the propagation time of the acoustic signal.

By computing the time difference $\Delta t_{\text{propag}} = t_{\text{propag(d)}} - t_{\text{propag(r)}}$, we obtain:

$$\Delta t_{\text{propag}} = d_{\text{water}} \cdot \left(\frac{1}{c_{\text{water}} - v \cdot \sin\alpha} - \frac{1}{c + v \cdot \sin\alpha} \right), \tag{5.22}$$

which can be rewritten as a quadratic equation in $(v \cdot \sin\alpha)$:

$$\frac{\Delta t_{\text{propag}}}{d_{\text{water}}} \cdot (v \cdot \sin\alpha)^2 + 2 \cdot (v \cdot \sin\alpha) - \frac{\Delta t_{\text{propag}}}{d_{\text{water}}} \cdot c_{\text{water}}^2 = 0. \tag{5.23}$$

This equation has two solutions, but we keep the one that has the same sign as Δt_{propag}, that is:

$$v \cdot \sin\alpha = \frac{-b + \sqrt{\Delta}}{2 \cdot a}, \tag{5.24}$$

where:

$$\Delta = \sqrt{b^2 - 4 \cdot a \cdot c}, \tag{5.25}$$

and:

$$\begin{cases} a = \Delta t_{\text{propag}} / d_{\text{water}} \\ b = 2 \\ c = -\Delta t_{\text{propag}} / d_{\text{water}} \cdot c_{\text{water}}^2 \end{cases}. \tag{5.26}$$

Hence, the water velocity can be computed with the following expression:

$$v = \frac{1}{\sin\alpha} \cdot \frac{-2 + \sqrt{4 + 4 \cdot \left(\frac{\Delta t_{\text{propag}}}{d_{\text{water}}} \right)^2 \cdot c_{\text{water}}^2}}{2 \cdot \frac{\Delta t_{\text{propag}}}{d_{\text{water}}}}. \tag{5.27}$$

By using this expression we compute the values of v for a range of values of Δt_{propag} (expressed in samples (with $\Delta t = 0.01\,\mu$s)). The result is shown in Figure 5.26.

It can be noticed from the dependence of v on Δt_{propag} that in order to have a sufficiently large number of samples for Δt_{propag} (while keeping Δt unchanged) v should have high values. We will use in our simulations values of v around 10 m/s. Even if these values are somewhat unrealistic, they will allow us to obtain a sufficiently large Δt_{propag} in the given simulation constraints (i.e. reduced pipe dimensions, fast simulations).

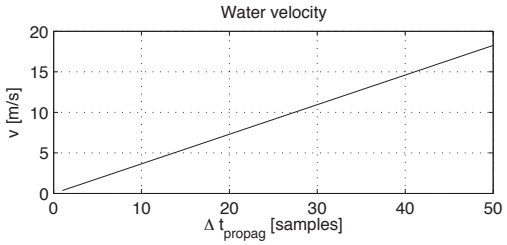


Figure 5.26: The values of the water velocity, v, with respect to the propagation time difference, Δt_{propag}.

We begin our water flow simulations with a constant water velocity profile. We consider the case $v = 10$ m/s. The HydroSimUS simulation parameters are shown in Figure 5.27.

Simulation paramters:

Parameter	Description	Value
Δx	Space meshing step on the j axis	0.1 mm
Δy	Space meshing step on the i axis	0.1 mm
N_1	Number of mesh points on i axis	300
N_2	Number of mesh points on j axis	900
c	Speed of acoustic wave	1480 m/s
N_{tx}	Number of punctual emitters	20
D_{tx}	Number of points between emitters	4
Δt	Time meshing step	0.01 μs
N_t	Number of time steps in simulation	5000
f_{sig}	Frequency of the emitted signal	1 MHz
N_{cyc}	Number of cycles in the emitted signal	1

Positions of the transducers:

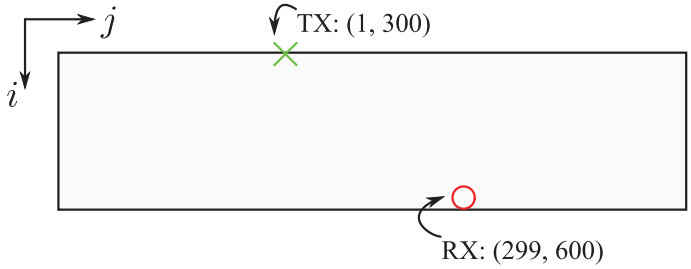


Figure 5.27: The HydroSimUS parameters used for simulating the water velocity measurement configuration (for the $v = 10$ m/s case, constant profile). (Direct): $v_x = -10$ m/s, $v_y = 0$ m/s; constant profile. (Reversed): $v_x = 10$ m/s, $v_y = 0$ m/s; constant profile. (v_x is the water velocity along the j axis, while v_y is the water velocity along the $-i$ axis.)

The obtained signals are depicted in Figure 5.28. The time difference between the two signals is about 24 samples, which means that $\Delta t_{\text{propag}} = 0.24\,\mu$s, thus leading to a water velocity of:

$$v = 9.99\,\text{m/s}. \tag{5.28}$$

(We note that this value was obtained by considering d_{water} to be the distance travelled by the acoustic wave from the rightmost point of the emitter to the reception point. Given

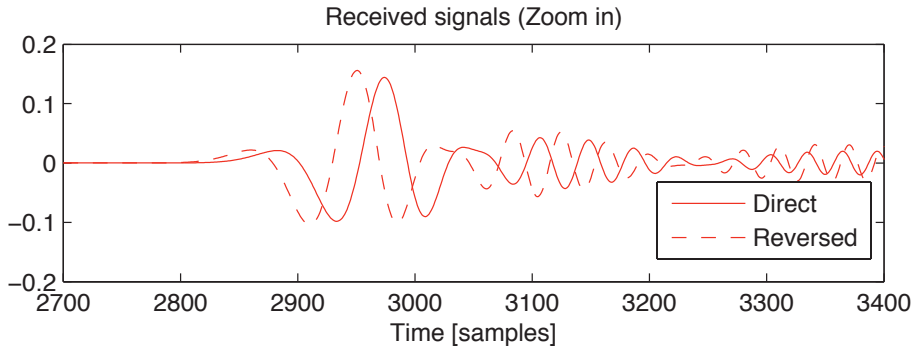


Figure 5.28: The received signals obtained for the simulation of the water velocity measurement configuration (for the $v = 10$ m/s case, constant profile).

the values of the parameters in Figure 5.27, this distance is 37.2 mm.)

Now we repeat the simulation of this water measurement configuration, by considering a Poiseuille velocity profile with a maximum value of 10 m/s. Except for the type of the velocity profile (which is Poiseuille instead of constant, in this case), all the other simulation parameters are the same as those shown in Figure 5.27. Figure 5.29 shows the received signals.

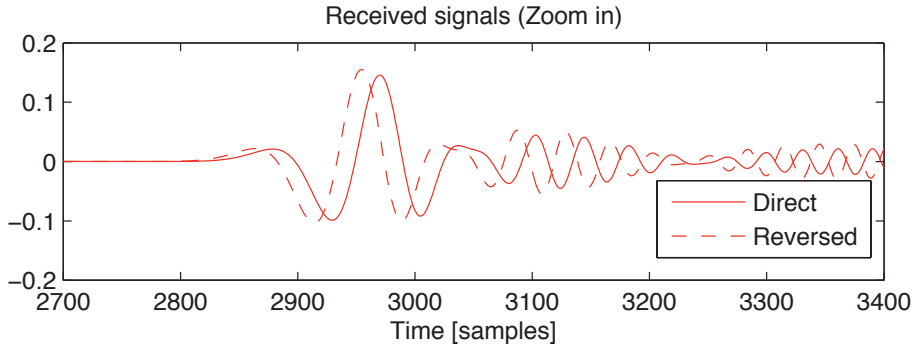


Figure 5.29: The received signals obtained for the simulation of the water velocity measurement configuration (for the $v = 10$ m/s case, Poiseuille profile).

The time difference between the two signals is of about 16 samples. Hence, $\Delta t_{\text{propag}} = 0.16\,\mu$s, which leads to a water velocity of:

$$v = 6.66\,\text{m/s}. \tag{5.29}$$

This is indeed the mean value of the Poiseuille velocity profile with a maximum value of 10 m/s, that we used in our simulation.

5.3 Automatic estimation of the propagation times

We showed in the previous section how the mean velocity of the water flowing through a pipe (and, hence, the flow rate as well) can be estimated by using the difference between the times required for the signal to travel the distance between the two transducers against the flow and with the flow, respectively.

In the previous section we estimated these propagation times visually. However, in the current section we intend to compute them automatically, by analyzing the received signals. The classic method for estimating the transit time of the signal consists in computing the signal envelope (which can be done by taking the absolute value of the analytic signal). This envelope can then be used for estimating the transit time with the aid of a threshold. The time instant when the first sample of the envelope exceeds the threshold is considered to be the transit time, t_{propag}. In Figure 5.30 we show the results we obtained for one of the signals in Section 5.2.2. We used as a threshold the value:

$$Th = \frac{\text{mean}\{Env\} + \max\{Env\}}{2}, \tag{5.30}$$

where Env is the envelope of the signal.

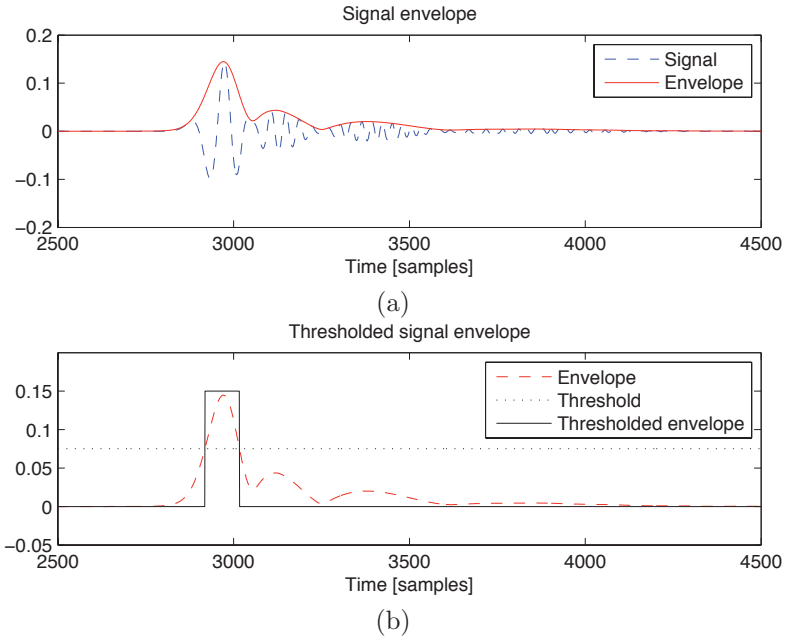


Figure 5.30: Estimation of the propagation time by using the envelope of the signal. (a) The envelope of the signal, computed as the amplitude of the analytic signal. (b) The thresholded envelope of the signal.

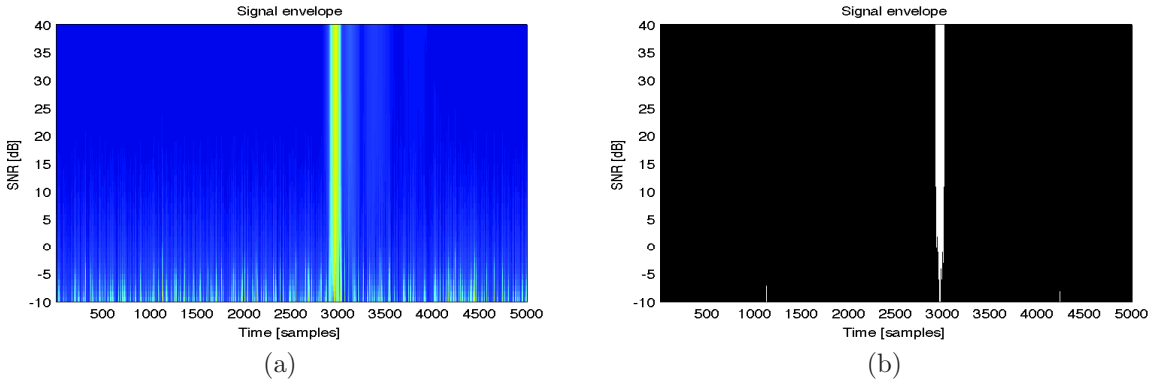
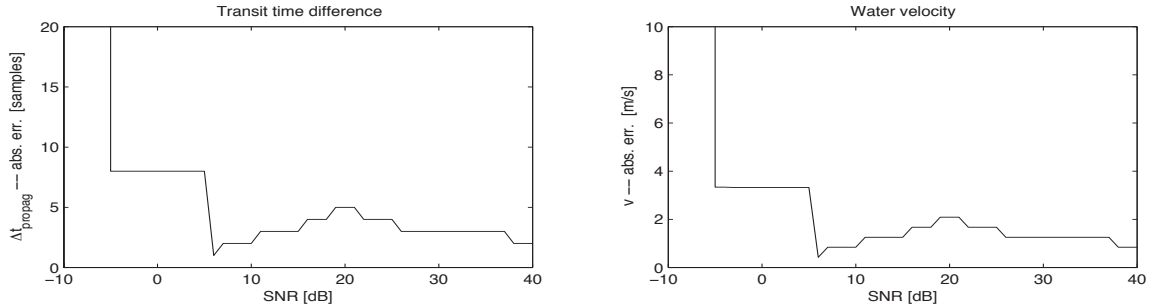


Figure 5.31: (a) The envelope of the signal, for a range of SNRs. (b) The thresholded version of the envelope.

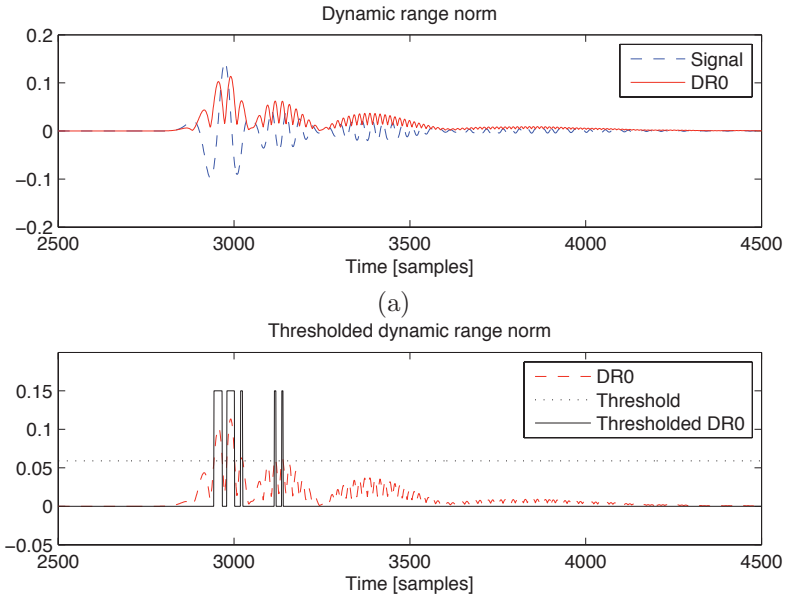
Now let us take the first pair of signals obtained in Section 5.2.2, i.e. those that correspond to a 10 m/s constant velocity profile. We estimate as discussed t_{propag} for the two signals, and we compute the transit time difference Δt_{propag} (out of which we obtain the mean water velocity, v). Then we add a synthetic white Gaussian noise and we repeat the computations. The results are shown in Figures 5.31 and 5.32.



(a) (b)

Figure 5.32: (a) The propagation time difference absolute error obtained with the signal envelope, for a range of SNRs. (b) The corresponding absolute errors obtained for the water velocity.

It can be noticed that down to a SNR of about 5 dB, the values obtained for the water velocity are not very far from the desired value.



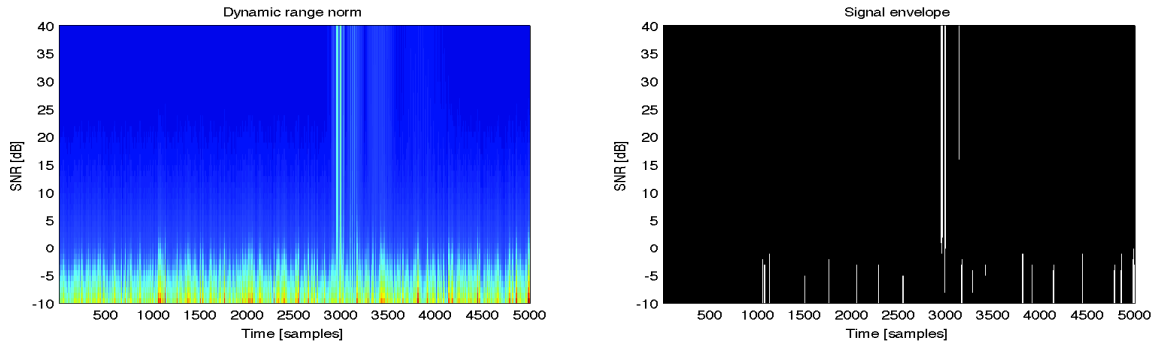
(a)

(b)

Figure 5.33: Estimation of the propagation time by using the dynamic range norm of the vector sampled signal (using $w = 12$). (a) The dynamic range norm. (b) The thresholded dynamic range norm.

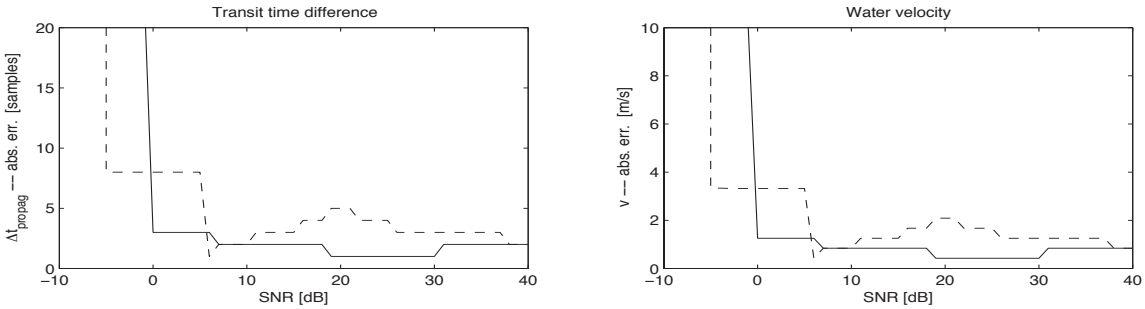
Next, we try to obtain all these results with a VeSP based estimation of the transit time. We chose for this purpose the dynamic range norm discussed in Section 3.3.2. Figure 5.33 illustrates this measure, computed for the signal in Figure 5.30. The threshold was computed as in Equation (5.30).

By repeating the computation of the $DR0$ measure (see Section 3.3.2) for a range of SNRs using synthetic white Gaussian noise, we obtain the results in Figure 5.34. Figure 5.35 shows the water velocities obtained by computing the transit time difference between the two signals with the (thresholded) dynamic range norm computed for each of them. It can be noticed that v is not far from the desired one until a SNR of 0 dB (which means a 5 dB improvement compared to the results obtained with the signal envelope (see Figure 5.32)).



(a) (b)

Figure 5.34: (a) The dynamic range norm of the signal (for $w = 12$), for a range of SNRs. (b) The thresholded dynamic range norm.



(a) (b)

Figure 5.35: (a) The propagation time difference absolute error obtained with the dynamic range norm (for $w = 12$), for a range of SNRs. (b) The corresponding absolute errors obtained for the water velocity. (The errors obtained when using the Hilbert envelope (see Figure 5.32) are shown with dashed line, in order to facilitate the comparison.)

Figure 5.36 illustrates comparatively the water velocities obtained with the two methods (i.e. Hilbert envelope, and DR0) for the noise recorded in Section 5.1.1, i.e. real colored Gaussian noise. The same as for the case of synthetic noise, DR0 improves with about 5 dB the noise robustness when estimating v.

Finally, we repeat the comparison between the two methods by using the second pair of signals in Section 5.2.2, i.e. the signals obtained for the Poiseuille velocity profile with $v_{\max} = 10$ m/s. Figure 5.37 shows the comparative results obtained with the two methods for synthetic noise, while Figure 5.38 shows the results obtained for real noise.

As noticed before, the dynamic range norm outperforms the Hilbert envelope when it comes to estimating the transit time of the signal. It is also easier to compute. However, its use requires the choice of a parameter – the vector sampling size, w.

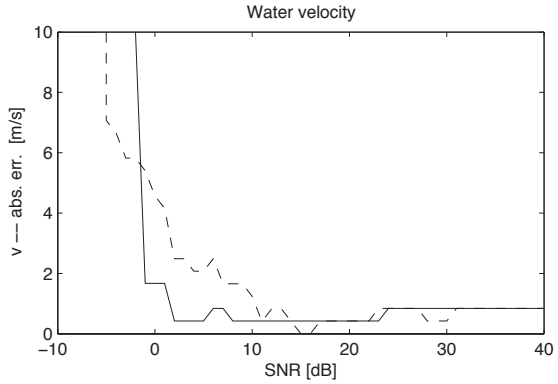


Figure 5.36: Comparative results obtained for real noise (colored Gaussian noise), for a range of SNRs. (Dashed line) The water velocity obtained with the signal envelope. (Continuous line) The water velocities obtained with the dynamic range norm (for $w = 12$).

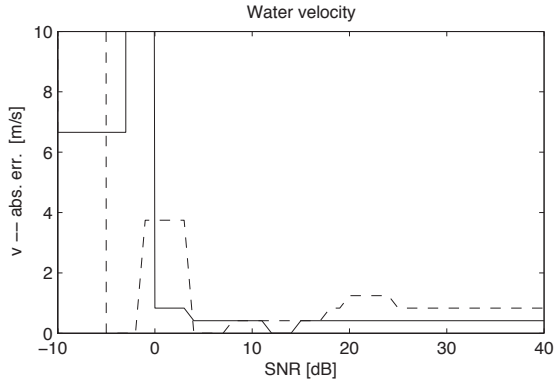


Figure 5.37: Comparative results obtained for synthetic noise (white Gaussian noise) on the signals obtained in the Poiseuille velocity profile case, for a range of SNRs. (Dashed line) The water velocity obtained with the signal envelope. (Continuous line) The water velocity obtained with the dynamic range norm (for $w = 12$).

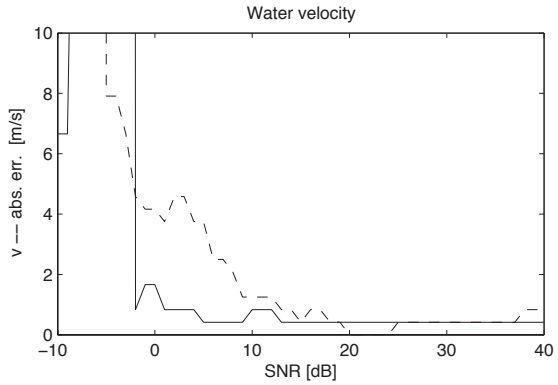


Figure 5.38: Comparative results obtained for real noise (colored Gaussian noise) on the signals obtained in the Poiseuille velocity profile case, for a range of SNRs. (Dashed line) The water velocity obtained with the signal envelope. (Continuous line) The water velocity obtained with the dynamic range norm (for $w = 12$).

5.4 Summary

The first part of this chapter dealt with the experimental validation of the HydroSimUS software simulator that was described in Chapter 2. To this end, we started by describing some experimental measurements that we performed on a plexiglass water pipe. Then, we repeated these experiments in simulation and we discussed the similarities as well as the differences between the simulated signals and the experimental ones.

The chapter continued with HydroSimUS simulations of some experimental setups that are used in measuring the level and velocity of the water inside the pipe. Discussion on the obtained signals is provided, as well as on the means to compute the desired water parameters from the analysis of these signals.

As the key part in the ultrasonic measurements that were discussed was the estimation of the propagation time of the emitted signal, in the last part of the chapter we performed a comparative analysis of the performance of a classic envelope based method and a VeSP based method that we developed in Chapter 3. The results have shown that our method outperforms the classic one, provided that a *proper* vector sampling size is chosen first.

In the following chapter we conclude the manuscript by summarizing the conclusions of this thesis, as well as the contributions of the author.

Chapter 6

Conclusion

Although they do not belong to a very well defined class of signals, transient signals are found in many practical fields (from biological signals to signals used in telecommunication systems), so the development of robust and efficient tools for their analysis is an important objective in signal processing. In particular, this thesis was focused mainly on solving some signal analysis problems raised by the ultrasonic investigation of water in pipes. However, this applicative context was used only as a basis for developing generic tools that can find their usefulness in various other applications.

The thesis addressed the issue of (transient) signal analysis from the perspective of the phase space recurrence concept, borrowed from dynamical systems theory. We showed that recurrence plot analysis (RPA) brings into the field of signal processing two important concepts: vector sample, and recurrence. The use of these concepts allowed us to generalize and extend well-known signal processing tools. Based on these concepts we could build signal representations that proved to be useful in various tasks related to transient signal detection and characterization.

Basically, this thesis has shown that RPA can be seen as a generic framework that unifies in a common formulation various well-known classic signal processing tools, which it generalizes (by using the concept of vector samples) and extends (by using the concept of recurrence). We have shown in this thesis that the new tools thus obtained outperform the traditional tools in certain cases.

We detail next the main contributions brought by this thesis, and then we outline some directions for further research.

6.1 Contributions

We split the contributions in three categories that will be listed separately: theoretical contributions, applicative contributions, and technical contributions.

- Theoretical contributions

  In this category we include those contributions that resulted from the study and development of methods for the analysis of transient signals. We mention first an

original presentation of the RPA method, which highlighted the aspects that are interesting from a signal processing perspective. This resulted into the introduction of the vector samples processing (VeSP) concept. We also extended the generic RPA methodology by defining new "distances" between vector samples (the angular distance, the dynamic range distance, the dot product distance), and by introducing the ε_0 recurrence threshold, which automatically adapts in order to couteract the effect of noise on the analyzed signal.

Another theoretical contribution is the VeSP-based generalization of some signal processing tools (the correlation coefficient, the energy, the envelope, the derivative, the averaged magnitude difference function). We further extended some of these tools, by defining them as recurrence quantification analysis (RQA) measures (the autocorrelation function, the averaged magnitude difference function, the time distributed histogram).

One more theoretical contribution that we mention here is the definition of the t-ε and t-w representations, based on the introduction of the new generic RQA measures σ_c and σ_d. (We showed that these representations highlight some useful features in the analyzed signals.)

- Applicative contributions

  We include in this category the contributions that are related to the solving of different signal processig tasks. First, we mention the time detection of transients in a signal, aiming to obtain a time detection curve that indicates the time support of transients. To this end, we made use of the σ_c time-recurrence representation (and the ε_0 recurrence threshold) that we proposed. The obtained time detection curve also allowed us to perform the attenuation of the noise areas in a signal that contains trasients. Besides, we also performed the detection of the envelope of a signal, by using the vector samples concept and the dynamic range "norm".

  Two more contributions that we mention in this category are related to the estimation of the instantaneous frequency of the analyzed signal by making use of the σ_d representation. We illustrated the use of this tool in estimating the fundamental frequency of a signal, as well as in the identification of transient chirps having low slopes.

- Technical contributions

  Finally, in this category we mention the contributions involving software coding and experimental work:

  - The development of a software application that allows to simulate the propagation of acoustic waves in liquid environments having arbitrary velocity profiles and situated in pipes or containers with arbitrary section shape – the HydroSimUS Matlab toolbox.
  - The implementation as Matlab functions of the signal processing tools developed in this thesis – the VeSP toolbox.
  - The experimental validation of the results obtained by HydroSimUS simulations, and the illustration of the working principle of ultrasonic flow meters based on the difference between transit times.

6.2 Future work

The analysis of transient signals is a signal processing topic that is both actual and important. It is particularly important in ultrasonic imaging applications, as well as in characterization and localization of natural sources of transient signals (such as whales and dolphins). Regardless of the application context, current theoretical research aims to build signal representation spaces that are able to represent transients by a number of parameters that is as small as possible, but still sufficient for ensuring a pertinent characterization of the phenomenon being studied.

This thesis has shown that phase space analysis of signals allows the contruction of representation spaces in which characteristics of transients such as the time support and the instantaneous period are highlighted. The generic RPA methodology is flexible, thus allowing as immediate developments the definition of new ways to compute distances between vector samples, as well as the definition of new VeSP/RQA measures. Of course, this will later require a detailed study of the efficiency of these new tools.

The two software tools developed during this thesis, i.e. the HydroSimUS and VeSP Matlab toolboxes, can be improved by optimizing their implementations. Although they have already served their purpose in illustrating the operation of different equations and algorithms, increasing their speed and reducing their memory requirements is an interesting direction from a technical point of view. The use of hardware systems that allow to perform highly parallel computations (such as the field programmable gate arrays) is a promising idea for this purpose.

An interesting research direction from a practical point of view is involving the transient signal processing tools presented in this thesis in detecting leakages in water pipe systems. Terrain movement, soil vibrations due to human activity, thermal expansion and contraction of the material, corrosion, as well as internal tensions due to abrupt changes in the water pressure can compromise the integrity of the pipes at localized spots, causing leakages along the pipe. Designing novel techniques for accurate and timely leakage detection and localization, as well as decision support systems that can exploit the inputs provided by such techniques for providing a real-time knowledge of the status of the pipe is, hence, a key strategic objective for water system management. Transient signal processing will play an important role here, in conjunction with nonintrusive ultrasonic monitoring equipments. The idea would be to place several acoustic transducers around the investigated pipe section, in order to obtain not only the mean value of the water velocity, but also an estimation of the flow profile. It would then be possible to identify structural problems in the pipe (such as leakages and corrosion) that are related to modifications of the flow profile.

Another interesting application for the tools that we studied and developed in this thesis is the development of smart sensor networks, i.e. distributed sensor networks with integrated processing. They are used, for example, for monitoring large size environments such as the underwater environment, where the presence of transients is very rich in information about the investigated phenomena. The need for autonomy, as well as the inherent synchronization problems involve the implementation of signal processing at the sensor – which will then send to the processing center only those information that are relevant for the phenomena being studied. In this context, the tools described in this

thesis can be used for detecting the transients and for extracting their significant features (such as the envelope and the instantaneous frequency).

From a theoretical point of view there are as well several interesting research directions. For instance, operations such as interpolation, decimation, as well as various (nonlinear) filtering operations that are traditionally performed on signals can be performed on multidimensional trajectories composed of vector samples. These new VeSP filters would benefit on one hand from the increased information contained in the vector samples (compared to regular samples), and on the other hand from a wide range of possible definitions for the notion of "distance" between vector samples.

The most interesting research direction from a theoretical point of view, however, is the development of a multiscale time-recurrence analysis. The idea is to extend the tools from time-frequency and time-scale analysis with the aid of the concepts of vector samples and recurrences. At first, it would be interesting to study if a VeSP extension of the spectrogram would improve the resolution when it comes to localizing time-frequency atoms. Then, it would be challenging to propose a definition of a recurrence-based (generalized) frequency as well as a technique for adapting the vector sampling window size to the local features of the signal being investigated. Practically, the fact that a trajectory composed of vector samples of size 1 is the signal itself suggests that any operation that is currently performed on the signal can be generalized by using vector samples of size greater than 1 (and by replacing the operations performed with regular samples with operations performed on vector samples). The most important question that raises here is whether (and under which conditions) this generalization leads to results that are better than those obtained by using the traditional signal processing tools.

Bibliography

[BHSL02] W. Benenson, J. W. Harris, H. Stoker, and H. Lutz. *Handbook of Physics*, chapter 5, pages 153–210. Springer-Verlag New York, Inc., 2002.

[BISC11] F.-M. Birleanu, C. Ioana, A. Serbanescu, and J. Chanussot. A time-distributed phase space histogram for detecting transient signals. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 3844–3847, Prague, Czech Republic, 22-27 May 2011.

[Bru98] M. Bruneau. *Manuel d'acoustique fondamentale*. Ed. Hermes, Paris, France, 1998.

[CCY09] H. Choi, H.G. Choi, and J.Y. Yoo, editors. *Computational Fluid Dynamics*. Springer-Verlag Berlin Heildelberg, 2009.

[CD98] B. Chopard and M. Droz. *Cellular automata modeling of physical systems*. Cambridge University Press, Cambridge, UK, 1998.

[CEFG91] M. Casdagli, S. Eubank, J. Doyne Farmer, and J. Gibson. State space reconstruction in the presence of noise. *Physica D*, 51:52–98, 1991.

[Deb97] L. Debnath. *Nonlinear Partial Differential Equations for Scientists and Engineers*. Birkhauser Boston, USA, 1997.

[EH02] B. O. Enflo and C. M. Hedberg. *Theory of Nonlinear Acoustics in Fluids*. Kluwer Academic Publishers, Dordrecht, The Netherlands, 2002.

[EOKR87] J.-P. Eckmann, S. Oliffson Kamphorst, and D. Ruelle. Recurrence plots of dynamical systems. *Europhysics Letters*, 4(9):973–977, November 1987.

[GDN98] M. Griebel, T. Dornseifer, and T. Neunhoeffer. *Numerical Simulation in Fluid Dynamics. A Practical Introduction*. Society for Industrial and Applied Mathematics, Philadelphia, USA, 1998.

[Ger03] D. Gerhard. Pitch extraction and fundamental frequency: History and current techniques. Technical report, University of Regina, Regina, Canada, 2003.

[GHL96] T. B. Gatski, M. Yousuff Hussaini, and J. L. Lumley. *Simulation and Modeling of Turbulent Flows*. Oxford University Press, Inc., UK, 1996.

[HC00] K. A. Hoffmann and S. T. Chiang. *Computational Fluid Dynamics*, volume I. Engineering Education System, Wichita, KS, USA, fourth edition, 2000.

[Huk06] J. P. Huke. Embedding nonlinear dynamical systems: A guide to Takens' theorem. Technical report, The University of Manchester, Manchester, UK, 2006.

[IQ03] C. Ioana and A. Quinquis. Transient signal detection using overcomplete wavelet transform and high-order statistics. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, volume VI, pages 449–452, Hong Kong, China, 6-10 April 2003.

[Jae06] O. Jaeckel. Strengths and weaknesses of calculating beamforming in the time domain. In *The 1st Berlin Beamforming Conference*, Berlin, Germany, 22-23 November 2006.

[Kan93] K. Kaneko. *Theory and applications of coupled map lattices*. John Wiley & Sons, New York, USA, 1993.

[KFCS00] L. E. Kinsler, A. R. Frey, A. B. Coppens, and J. V. Sanders. *Fundamentals of Acoustics*. John Wiley & Sons, Inc., New York, USA, fourth edition, 2000.

[KP77] J. B. Keller and J. B. Papadakis, editors. *Wave Propagation and Underwater Acoustics*. Springer-Verlag Berlin Heidelberg, New York, USA, 1977.

[LAC97] J.-L. Lacoume, P.-O. Amblard, and P. Comon. *Statistiques d'Ordre Supérieur pour le Traitement du Signal*. Masson, 1997.

[Les08] M. Lesieur. *Turbulence in Fluids*. Springer, Dordrecht, The Netherlands, 2008.

[Lyn89] L.C. Lynnworth. *Ultrasonic measurements for process control: theory, techniques, applications*. Academic Press, 1989.

[Mal09] S. Mallat. *A Wavelet Tour of Signal Processing. The Sparse Way*. Academic Press, 2009.

[Man04] P. Manneville. *Instabilities, Chaos and Turbulence. An Introduction to Nonlinear Dynamics and Complex Systems*. Imperial College Press, London, UK, 2004.

[Mar03] N. Marwan. *Encounters with neighbours. Current developments of concepts based on recurrence plots and their applications*. PhD thesis, Institut für Physik, Fakultät Mathematik und Naturwissenshaften, Universitat Potsdam, May 2003.

[Mar08] N. Marwan. A historical review of recurrence plots. *The European Physical Journal Special Topics*, 164:3–12, 2008.

[MCRTK07] N. Marwan, M. Carmen Romano, M. Thiel, and J. Kurths. Recurrence plots for the analysis of complex systems. *Physics Reports*, 438:237–329, January 2007.

[Mel08] D.K. Mellinger. A neural network for classifying clicks of blainville's beaked whales (*Mesoplodon densirostris*). *Canadian Acoustics*, 36(1):55–59, 2008.

[MK02] N. Marwan and J. Kurths. Nonlinear analysis of bivariate data with cross recurrence plots. *Physics Letters A*, 302:299–307, 2002.

[MK09] N. Marwan and J. Kurths. Comment on "stochastic analysis of recurrence plots with applications to the detection of deterministic signals" by rohde et al. [physica d 237 (2008) 619-629]. *Physica D*, 238:1711–1715, 2009.

[MNB08] J. V. Michalowicz, J. M. Nichols, and F. Bucholtz. Signal detection based on recurrence matrix statistics. *Physics Letters A*, 372:7172–7178, 2008.

[Muh11] G. Muhammad. Extended average magnitude difference function based pitch detection. *The International Arab Journal of Information Technology*, 8(2):197–203, April 2011.

[OSA11] I. Orović, S. Stanković, and M. Amin. A new approach for classification of human gait based on time-frequency feature representations. *Signal Processing*, 91(6):1448–1456, June 2011.

[OZK01] H. M. Ozaktas, Z. Zalevsky, and M. A. Kutay. *The Fractional Fourier Transform with Applications in Optics and Signal Processing*. John Wiley & Sons, 2001.

[Pai05] H.J. Pain. *The Physics of Vibrations and Waves*. John Wiley & Sons Ltd., New York, USA, sixth edition, 2005.

[PCFS80] N. Packard, J. Crutchfield, D. Farmer, and R. Shaw. Geometry from a time series. *Physical Review Letters*, 45(9):712–716, 1980.

[Pop00] S. B. Pope. *Turbulent Flows*. Cambridge University Press, Cambridge, UK, 2000.

[PS02] A. Papandreou-Suppappola. *Applications in Time-Frequency Signal Processing*. CRC Press, 2002.

[RA98] Ph. Ravier and P.-O. Amblard. Combining an adapted wavelet transform with 4th order statistics for transient detection. *Signal Processing*, 70:115–128, 1998.

[RA01] Ph. Ravier and P.-O. Amblard. Wavelet packets and denoising based on higher-order statistics for transient detection. *Signal Processing*, 81(9):1909–1926, August 2001.

[Rai06] D. R. Raichel. *The science and applications of acoustics*. Springer Science+Business Media, Inc., New York, USA, second edition, 2006.

[Rav98] P. Ravier. *Détection des transitoires par ondelettes adaptées. Critères d'adaptation fondés sur les statistiques d'ordre supérieur*. PhD thesis, Intitut National Polytechnique de Grenoble, 1998.

[RFA+10] C. Richard, A. Ferrari, H. Amoud, P. Honeine, P. Flandrin, and P. Borgnat. Statistical hypothesis testing with time-frequency surrogates to check signal stationnarity. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 3666–3669, Dallas, TX, USA, 14-19 March 2010.

[RNDB08] G. K. Rohde, J. M. Nichols, B. M. Dissinger, and F. Bucholtz. Stochastic analysis of recurrence plots with applications to the detection of deterministic signals. *Physica D*, 237:619–629, 2008.

[Sei08] B. Seibold. A compact and fast matlab code for solving the incompressible navier-stokes equations on rectangular domains, March, 31 2008.

[Sma05] M. Small. *Applied nonlinear time series analysis. Applications in physics, physiology and finance.* World Scientific Publishing, Singapore, 2005.

[SS05] R. Saxena and K. Singh. Fractional fourier transform: A novel tool for signal processing. *Journal of the Indian Institute of Science*, 85:11–26, January-February 2005.

[SSB11] A. Serbanescu, O. Stanasila, and F.-M. Birleanu. *Analiza neliniară a seriilor de timp: aplicaţii în prelucrarea semnalelor (Nonlinear time series analysis: applications in signal processing).* Military Technical Academy, Bucharest, Romania, 2011.

[Sta02] LJ. Stanković. Time-frequency distributions with complex argument. *IEEE Transactions on Signal Processing*, 50(3):475–486, March 2002.

[SYC91] T. Sauer, J. A. Yorke, and M. Casdagli. Embedology. *Journal of Statistical Physics*, 65(3/4):579–616, 1991.

[Tak81] F. Takens. Detecting strange attractors in turbulence. In D. A. Rand and L.-S. Young, editors, *Dynamical Systems and Turbulence. Lecture Notes in Mathematics*, volume 898, pages 230–242. Springer-Verlag, 1981.

[Ter02] D. E. Terez. Robust pitch determination using nonlinear state-space embedding. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 1, pages 345–348, Orlando, FL, USA, 13-17 May 2002.

[TL72] H. Tennekes and J. L. Lumley. *A first course in turbulence.* Kingsport Press, USA, 1972.

[Tri77] D.J. Tritton. *Physical Fluid Dynamics.* Cambridge University Press, Cambridge, UK, 1977.

[Tsi09] A. Tsinober. *An informal conceptual introduction to turbulence.* Springer Science+Business Media B.V., 2009.

[VB88] B. D. Van Veen and K. M. Buckley. Beamforming: A versatile approach to spatial filtering. *IEEE ASSP Magazine*, pages 4–24, April 1988.

[WHS06] C. Wagner, T. Huttl, and P. Sagaut. *Large Eddy Simulation for Acoustics.* Cambrigde University Press, New York, USA, 2006.

[Wil93] D. C. Wilcox. *Turbulence Modeling for CFD.* DCW Industries, Inc., California, USA, 1993.

[WJZ94] C. L. Webber Jr. and J. P. Zbilut. Dynamical assessment of physiological systems and states using recurrence plot strategies. *Journal of Applied Physiology*, 76(2):965–973, 1994.

[ZGWJ98] J. P. Zbilut, A. Giuliani, and C. L. Webber Jr. Detecting deterministic signals in exceptionally noisy environments using cross-recurrence quantification. *Physics Letters A*, 246:122–128, 1998.

[ZGWJ00] J. P. Zbilut, A. Giuliani, and C. L. Webber Jr. Recurrence quantification analysis as an empirical test to distinguish relatively short deterministic versus random number series. *Physics Letters A*, 267:174–178, 2000.

[ZWJ92] J. P. Zbilut and C. L. Webber Jr. Embeddings and delays as derived from quantification of recurrence plots. *Physics Letters A*, 171:199–203, 1992.

Appendix A

Appendix

A.1 Computation of SNR

The signal-to-noise ratio in decibels is defined as:

$$SNR_{[dB]} = 20 \cdot \log_{10} \frac{\text{rms}\{s\}}{\text{rms}\{z\}}, \tag{A.1}$$

where s is the signal, z is the noise, and $\text{rms}\{x\}$ is the root mean square amplitude of signal x, i.e.:

$$\text{rms}\{x\} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} x_i^2}. \tag{A.2}$$

Hence, in order to obtain a desired SNR, a certain fraction of noise must be added to the signal, as follows:

$$s' = s + f_z \cdot z, \tag{A.3}$$

where f_z is computed as:

$$f_z = \left(\frac{\text{rms}\{s\}}{\text{rms}\{z\}}\right) / 10^{\frac{SNR}{20}}. \tag{A.4}$$

A.2 ROC evaluation

Receiver operating characteristics describe the performance of a signal detector at a certain SNR in terms of the dependance between the probability of detection, P_d, and the probability of false alarm, P_{fa}. For a fixed P_{fa}, the higher P_d, the better the detector.

The algorithm we used for computing ROCs has the following steps:

- take N_{points} equally spaced threshold values between 0 and 1; this will lead to N_{points} on the resulting ROC curves
- for N_{rep} realizations of noise, repeat the following steps
- add noise to signal and compute time detection curve

- threshold the obtained time detection curve and compare it to the ideal one
- compute P_d as the ratio between the number of samples correctly detected and the total number of samples to detect
- compute P_{fa} as the ratio between the number of samples erroneously detected and the total number of samples not to detect.
- finally, return the average of the N_{rep} computed ROCs.

A.3 HydroSimUS Toolbox

This toolbox contains our implementation of a simulator for acoustic wave propagation in water flows in pipes. Details about it can be found in Chapter 2. It can be obtained by contacting the author at `florinmarianb@gmail.com`. It is also available for download at `https://docs.google.com/open?id=0B0KhcFTdgVm5aHN1RGMzTkJFRUE`. We present next some details about the graphical user interface of the simulator, as well as about its use for performing simulations on different pipe section types.

A.3.1 Graphical user interface of the simulator

In order to facilitate the input of the wave propagation simulation parameters, we built a graphical user interface for *HydroSimUS* (using Matlab version R2008a for Mac OS X). It can be seen in Figure A.1. This graphical user interface allows to introduce the simulation parameters (and it also roughly checks their validity), as well as to visualize the simulation results in real time. It shows the field of acoustic pressure as it evolves in time, as well as the emitted and the received signals. In the following section we make use of this user interface for performing some sample simulations that highlight the potential of *HydroSimUS*.

A.3.2 Simulation setups

In the current section we address the three main classes of pipe sections (see Figure 2.2) that can be simulated with HydroSimUS (i.e. transversal, oblique, and longitudinal), by illustrating some sample setups for each of them.

Transversal section simulation

Figure A.2 illustrates a wave propagation simulation on a perpendicular transversal pipe section. The picture is a snapshot taken at a certain time step while the simulation was running. It shows the parameter values being used, the acoustic pressures over the entire simulation domain, as well as the emitted and the received signal. The position of the emitter is marked with a (red) cross, and the position of the receiver is marked with a (green) circle on the plot showing the acoustic pressures. The emitted signal is shown (in red) in the top plot on the right, and the received signal is shown (in green) in the bottom

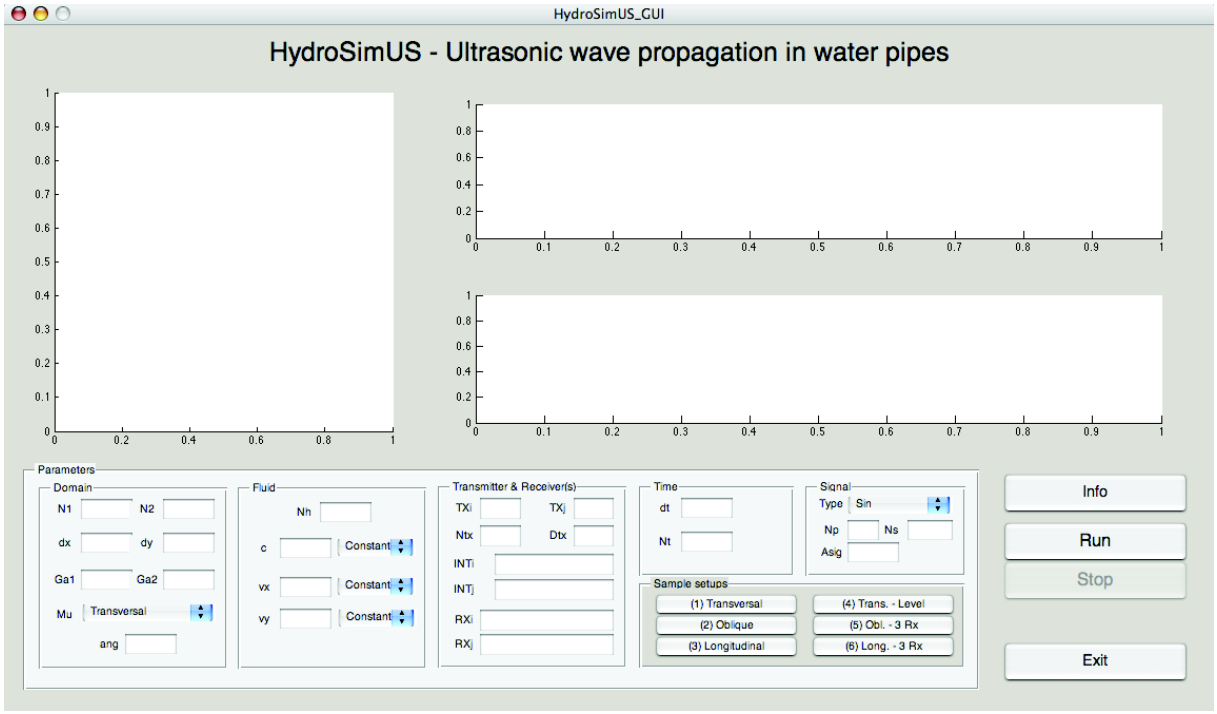


Figure A.1: The graphical interface of *HydroSimUS* – our software simulator of acoustic wave propagation in arbitrary velocity profiles.

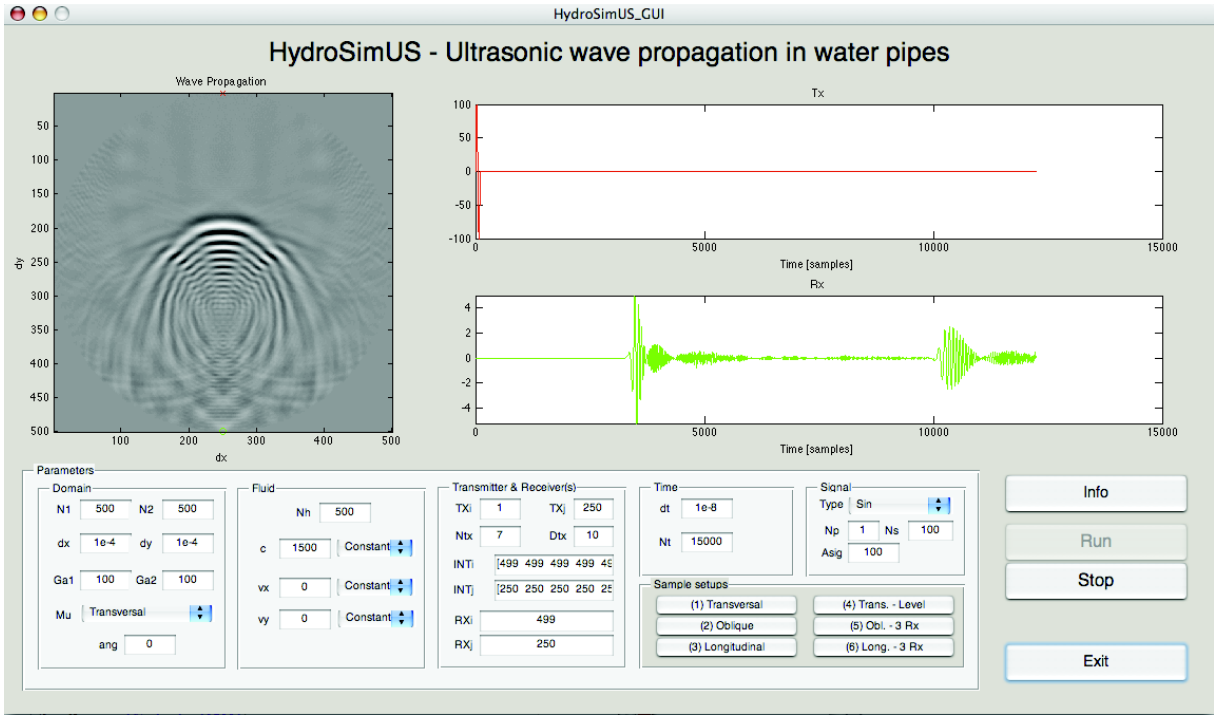


Figure A.2: Sample simulation on perpendicular transversal section of the pipe.

plot on the right. The pipe diameter is 5 cm, the frequency of the emitted signal is 1 MHz, and the wave propagation speed is 1500 m/s. Hence, by taking into account the considerations in Section 2.2.5, we chose a value of 10^{-4} m for the space meshing steps and a value of 10^{-8} for the time meshing step. For the emitter we used 15 omnidirectional

emmiters placed at a distance of 1 mm between each other (see Section 2.2.2).

Two trasient signals can be clearly identified visually in the received signal in Figure A.2. The first one (starting around time sample 3300) represents the arrival of the emitted signal at the reception point, while the second one (starting around time sample 10000) represents the arrival of the reflected signal. The time of arrival of the second trasient is three times greater than the time of arrival of the first transient because the reflected signal traveled a distance that equaled three times the pipe diameter (while the first one only travelled a distance equal to the pipe diameter).

Oblique section simulation

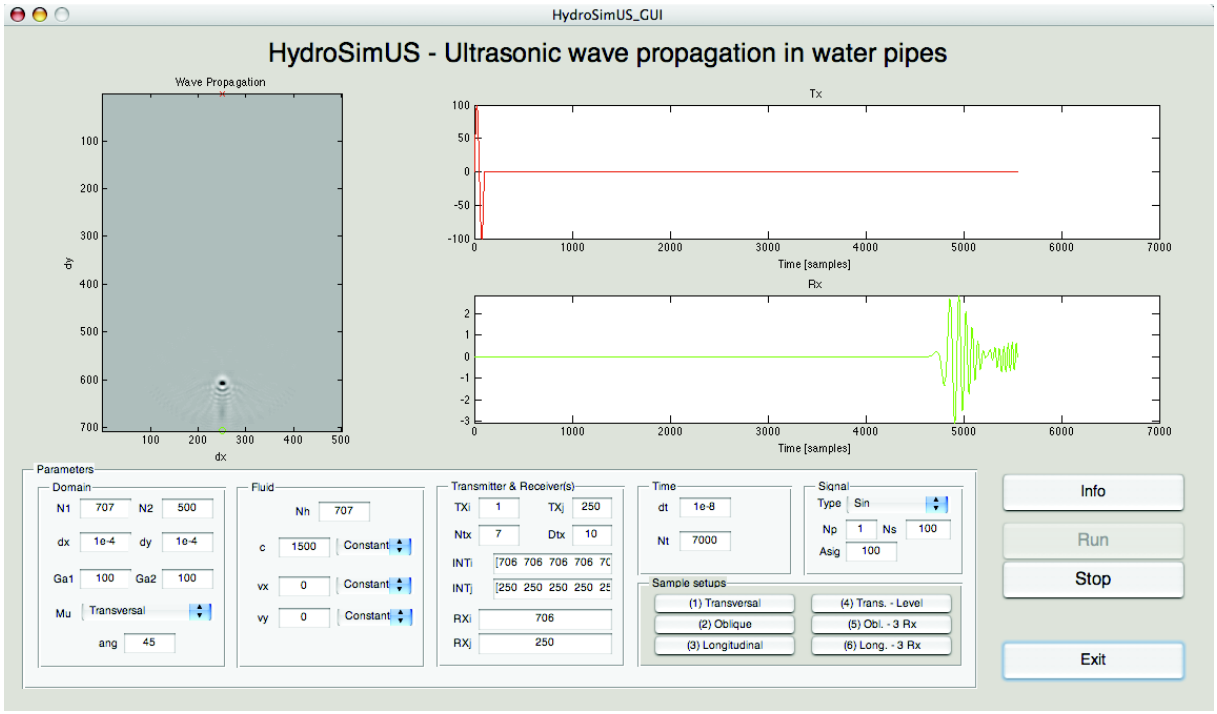


Figure A.3: Sample simulation on oblique transversal section of the pipe.

The configuration simulated in Figure A.3 is very similar to the previous one. The values of all the parameters are the same, except that in this case the ultrasonic transducers do not act perpendicularly on the investigated water, but they operate under an angle of 45 degrees.

In this case we are not interested in any reflection of the signal (as in fact the reflection leaves the oblique section), but only on the arrival of the signal coming directly from the emitter. As we can see from the figure, this signal arrives around time sample 4700, which corresponds to a distance that is $1/\cos 45°(=\sqrt{2})$ times greater than the pipe diameter (which is equal to $N_2 \cdot \Delta x$, i.e. 5 cm).

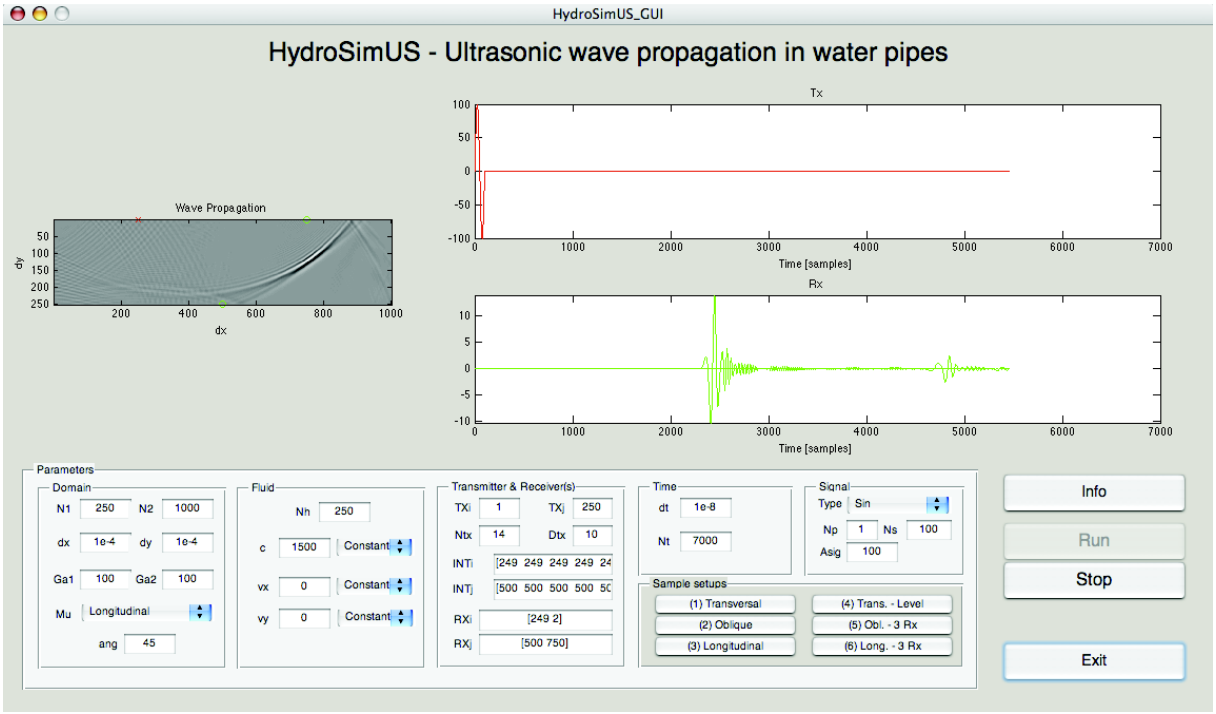


Figure A.4: Sample simulation on longitudinal section of the pipe.

Longitudinal section simulation

Finally, Figure A.4 illustrates wave propagation on a longitudinal section of the pipe. In this case we considered two receivers: the first one is placed on the other side of the pipe (as in the previous case), and the second one is placed on the same side of the pipe as the emitter. That is, the first receiver aims to receive the direct arrival of the emitted signal, while the second receiver aims to receive the signal reflection coming from the bottom wall of the pipe. We considered in this case a pipe diameter of 2.5 cm and we performed simulation on a pipe chunk that is 10 cm long. The values for the rest of the parameters are similar as in the previous case. The same as for the previous two simulation cases, we considered that the pipe is full of water and that there is no water flow, i.e. the local water velocities are 0 in the x-axis as well as in the y-axis direction.

Both received signals are superposed on the same plot (i.e. the green, bottom plot). The first transient signal (with the high amplitude) is received by the first receiver, and the second trasient (with the low amplitude) is received by the second receiver. The time of arrival of the second transient is two times greater than the time of arrival of the first one, as it traveled a double distance.

A.4 VeSP Toolbox

This toolbox contains implementations for the vector samples based signal processing tools described in Chapters 3 and 4. It can be obtained by contacting the author at `florinmarianb@gmail.com`. It is also available for download at `https://docs.google.com/open?id=0B0KhcFTdgVm5dmlTMDJjMmI4WTg`.

A.5 Publications of the author

- Florin-Marian Birleanu, Cornel Ioana, Alexandru Serbanescu, Gheorghe Serban, Emil Sofron. *Estimation of thermo-hydrodynamic parameters in energy production systems using non-stationary signal processing.* In Fourty-Fourth Asilomar Conference on Signals, Systems and Computers, pages 1353–1356, Monterey, CA, USA, 7-10 November 2010.

- Florin-Marian Birleanu, Cornel Ioana, Alexandru Serbanescu, and Jocelyn Chanussot. *A time-distributed phase space histogram for detecting transient signals.* In IEEE International Conference on Acoustics, Speech and Signal Processing, pages 3844–3847, Prague, Czech Republic, 22-27 May 2011.

- Florin-Marian Birleanu, Cornel Ioana, Cedric Gervaise, Jocelyn Chanussot, Alexandru Serbanescu, Gheorghe Serban. *On the Recurrence Plot Analysis method behaviour under scaling transform.* In 2011 IEEE Statistical Signal Processing Workshop (SSP), pages 793–796, Nice, France, 28-30 June 2011.

- Florin-Marian Birleanu, Cornel Ioana, Cedric Gervaise, Alexandru Serbanescu, Jocelyn Chanussot. *Caractérisation des signaux transitoires par l'analyse des récurrences de phase.* In 23ème Colloque GRETSI, Bordeaux, France, 5-8 September 2011.

- Florin-Marian Birleanu, Cornel Ioana, Cedric Gervaise, Yvan Simard. *Recurrence plot analysis (RPA): a new tool for click detection and clustering clicks into trains.* In Fifth International Workshop on Detection, Classification, Localization, and Density Estimation of Marine Mammals using Passive Acoustics, Timberline Lodge, Mount Hood, Oregon, USA, 21-25 August 2011.

- Florin-Marian Birleanu, Ion Candel, Cornel Ioana, Cedric Gervaise, Alexandru Serbanescu, Gheorghe Serban. *A Vector Approach to Transient Signal Processing.* In The 11th International Conference on Information Science, Signal Processing and their Applications, pages 1174–1179, Montreal, Quebec, Canada, 2-5 July 2012.

- Florin-Marian Birleanu, Cornel Ioana, Gheorghe Serban, Alexandru Serbanescu, Jocelyn Chanussot. *Modeling of acoustic wave propagation through turbulent fluids.* Submitted to Journal of the Acoustical Society of America, 2011.

- Florin-Marian Birleanu, Cornel Ioana. *Transient Signal Detection using Vector Approach.* Submitted to IEEE Transactions on Signal Processing, 2012.

- Florin-Marian Birleanu, Cornel Ioana. *Modélisation de la propagation des ultrasons dans un écoulement turbulent.* Rapport de recherche, 14 juin 2010.

Résumé :

Même s'ils n'appartiennent pas a une classe de signaux bien définie, les signaux transitoires se retrouvent dans plusieurs domaines pratiques très différents (comme les signaux médicaux et les signaux utilisés dans des systèmes de télécommunications), donc le développement des outils robustes et efficients pour faire leur analyse est un objectif important dans le traitement du signal. En particulier, cette thèse s'est concentrée surtout sur la résolution de quelques problèmes de traitement du signal posés par l'investigation ultrasonore de l'eau dans des conduites. Cependant, ce contexte applicatif a été utilisé seulement comme une base pour développer des outils génériques qui peuvent trouver leur utilité dans plusieurs d'autres applications. La thèse a abordé la question du traitement des signaux (transitoires) du point de vue du concept de récurrence dans l'espace des phases, qui est emprunté de la théorie des systèmes dynamiques. Nous avons montré que l'analyse du diagramme des recurrences (RPA) apporte dans le traitement du signal deux concepts importants : l'échantillon vectoriel, et la récurrence. L'utilisation de ces concepts nous a permis de généraliser et d'étendre des outils de traitement du signal bien connus, et aussi de construire des représentation de signal qui se sont révélées utiles dans des différentes tâches liées à la détection et à la caractérisation des transitoires. Essentiellement, dans cette thèse nous avons montré que la RPA peut être vue comme un cadre générique qui unifie dans une formulation commune des diverses outils bien connus dans le traitement du signal. En plus, elle généralise ces outils (à l'aide du concept des échantillons vectoriels) et les étend (en utilisant le concept de récurrence).

Mot-clés : Signal transitoire, mesure par ultrasons, analyse des récurrences.

Abstract:

Although they do not belong to a very well defined class of signals, transient signals are found in many practical fields (from biological signals to signals used in telecommunication systems), so the development of robust and efficient tools for their analysis is an important objective in signal processing. In particular, this thesis was focused mainly on solving some signal analysis problems raised by the ultrasonic investigation of water in pipes. However, this applicative context was used only as a basis for developing generic tools that can find their usefulness in various other applications. The thesis addressed the issue of (transient) signal analysis from the perspective of the phase space recurrence concept, borrowed from dynamical systems theory. We showed that recurrence plot analysis (RPA) brings into the field of signal processing two important concepts: vector samples, and recurrence. The use of these concepts allowed us to generalize and extend well-known signal processing tools, as well as to build signal representations that proved to be useful in various tasks related to transient signal detection and characterization. Basically, in this thesis we have shown that RPA can be seen as a generic framework that unifies in a common formulation various well-known classic signal processing tools, which it generalizes (by using the concept of vector samples) and extends (by using the concept of recurrence).

Keywords: Transient signal, ultrasonic measurement, recurrence plot analysis.