DamVis
Visualizing Dam Behaviour

João Carlos da Costa Moura

Thesis to obtain the Master of Science Degree in
Information Systems and Computer Engineering

Supervisor(s):  Prof. Daniel Jorge Viegas Gonçalves
               Prof. Sandra Pereira Gama

Examination Committee
Chairperson:  Prof. Alberto Manuel Rodrigues da Silva
Supervisor:  Prof. Daniel Jorge Viegas Gonçalves
Member of the Committee:  Prof. João Manuel Brisson Lopes

October 2017
Acknowledgments

A special thanks to my supervisors who helped make this project possible. Prof. Daniel Gonçalves, thank you for your insightful guidance and help. Prof. Sandra Gama, thank you for your support, concern and availability. I could not ask for better supervisors. Also thank you to Eng. Sérgio Oliveira, our LNEC supervisor, for your will to help and the opportunity to be part of such a challenging but very rewarding project.

Thank you to all my friends and family for always being there, supporting and helping me through not only this thesis but also my whole academic career and life. Without you, reaching this far would be much more difficult.

For that and much more that I can not describe, my huge thanks.
Resumo

Barragens são estruturas de extrema importância para o meio que as rodeia e que necessitam de ser monitorizadas. Interpretar e compreender os dados recolhidos do comportamento da barragem nem sempre é uma tarefa fácil. Esta tese fornece um sistema de monitorização de barragens atualizado, que se chama DamVis. Este projeto foi desenvolvido em parceria com o Departamento de Barragens de Betão do LNEC (Laboratório Nacional de Engenharia Civil).

DamVis foi desenvolvido com tecnologias web tanto para o back-end (NodeJS, HapiJS, MathJS, etc.) como para o front-end (HTML, CSS, jQuery, D3js, Threejs, etc.). A comunicação entre estes dois componentes é feita através de uma RESTful API, enviando os dados em formato JSON. As novas formas de representação de dados e interação com eles, para este tipo de sistema, foram concebidas com a ajuda dos princípios da visualização de informação (InfoVis).

Foram executados testes de usabilidade (incluindo a pontuação System Usability Scale) e estudos de caso para validar DamVis. Eles comprovaram que este é uma melhor alternativa aos habituais sistemas de monitorização de barragens, como os atualmente utilizados pelo LNEC.

Palavras-chave: Visualização do Comportamento da Barragem, Monitorização de Barragens, Visualização de Informação, InfoVis, Visualização 3D da Barragem, Processamento de dados da barragem
Abstract

Dams are very important structures to their surroundings that need to be monitored. Interpreting and understanding harvested dam behaviour data is not always an easy task. This thesis provides an updated and refreshed system for monitoring dam data, called DamVis. It was developed in partnership with LNEC's Concrete Dam Department.

DamVis was developed using web technologies for both the back-end (NodeJS, HapiJS, MathJS, etc.) and the front-end (HTML, CSS, jQuery, D3js, Threejs, etc.). The communication between both components is made through a RESTful API, sending data in the JSON format. Renewed ways of representing data and interacting with it, for this type of system, was achieved through Information Visualization (InfoVis) principles.

Usability Tests (including a System Usability Scale scoring) and Case Studies were done to validate DamVis. They proved that DamVis is a better alternative to common dam monitoring systems, like those currently used by LNEC.

**Keywords:** Visualizing Dam Behaviour, Dam Monitoring, Information Visualization, InfoVis, Dam 3D Visualization, Dam Data Processing
## Contents

Acknowledgments .................................................. iii
Resumo ................................................................. v
Abstract ............................................................... vii
List of Tables ........................................................ xi
List of Figures ........................................................ xiii

1 Introduction ......................................................... 1
   1.1 Motivation ..................................................... 2
   1.2 Objectives ..................................................... 3
   1.3 Thesis Outline ................................................ 4

2 Theoretical Background ............................................ 5

3 Related Work ...................................................... 9
   3.1 Information Visualization Principles ............................ 9
   3.2 Temporal Data .................................................. 10
   3.3 Hierarchical Data .............................................. 12
   3.4 Monitoring Techniques ........................................ 13
   3.5 Discussion ..................................................... 15

4 DamVis - Visualizing Dam Behaviour ........................... 17
   4.1 Requirements and Data Gathering ............................. 18
   4.2 System Architecture .......................................... 20
      4.2.1 Back-End ................................................ 21
      4.2.2 Front-End ................................................ 24
      4.2.3 System Potential ....................................... 40

5 Evaluation ......................................................... 45
   5.1 Usability Tests ................................................ 45
      5.1.1 Protocol ................................................ 46
      5.1.2 Results .................................................. 47
      5.1.3 Discussion .............................................. 49
   5.2 Case Studies ................................................ 51
List of Tables

<table>
<thead>
<tr>
<th></th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>Comparison between discussed visualization techniques.</td>
<td>16</td>
</tr>
<tr>
<td>4.1</td>
<td>Schematic of the data in the water level file.</td>
<td>19</td>
</tr>
<tr>
<td>4.2</td>
<td>Schematic of the data in the sensor data files.</td>
<td>20</td>
</tr>
<tr>
<td>5.1</td>
<td>Time and Errors for each task</td>
<td>48</td>
</tr>
<tr>
<td>5.2</td>
<td>SUS Questionnaire</td>
<td>48</td>
</tr>
</tbody>
</table>
## List of Figures

1.1 Visualization tool for dam monitoring used by LNEC. ........................................ 1
1.2 Cabrıl's Dam picture. .......................................... 2

3.1 Representation of a Calendar Heat Map [9] ........................................ 11
3.2 Representation of a River View [9] ........................................ 11
3.3 Representation of an Arc View [12] ........................................ 12
3.4 Spectrogram that represents frequency values over time [10] ................................ 12
3.5 Rooted Tree Diagram with complementary bar charts for some nodes [14] ............... 13
3.6 Comparison between predicted (red line) and the real values registered (blue line) [17] . 14
3.7 Schematic side way view of a dam with a spectrogram representation to show higher displacement/movement points [24] ........................................ 15
3.8 Schematic representation of a dam's arch using spectrogram to show higher displacement/movement points [25] ........................................ 15

4.1 Cabrıl dam and the location of its sensors. [18] ........................................ 19
4.2 Basic scheme of DamVis architecture. ........................................ 20
4.3 Paths that the back-end handles. ........................................ 21
4.4 Raw Sensor Data to FFT. ........................................ 23
4.5 FFT data to FDD-SV. ........................................ 24
4.6 LFP - Home Page ........................................ 26
4.7 LFP - Accelerograms and FFTs panel ........................................ 26
4.8 LFP - Frequency Domain Decomposition (FDD) panel ........................................ 27
4.9 LFP - 3D Model Finite Element Panel ........................................ 28
4.10 LFP - Second Version - Home Page ........................................ 28
4.11 LFP - Second Version - Accelerograms and FFTs panel ........................................ 29
4.12 LFP - Second Version - Frequency Domain Decomposition (FDD) panel ........................................ 30
4.13 LFP - Second Version - 3D Model Finite Element Panel ........................................ 31
4.14 Functional Prototype - First iteration of the Water Level Panel. ........................................ 32
4.15 Functional Prototype - Water Level with tool-tip. ........................................ 32
4.16 Functional Prototype - Point in time selected and Sensor Schema appeared. ........................................ 32
4.17 Functional Prototype - Accelerograms and FFTs with Maximum, Median and Minimum value for each sample. ................................................................. 33
4.18 Functional Prototype - Water Level and Sensor Schema new positioning. .......... 34
4.19 Functional Prototype - (a) Mouse hover with tool-tip (b) Mouse click .................. 34
4.20 Functional Prototype - Error Panel. .................................................... 35
4.21 Functional Prototype - Accelerograms and FFTs with Water Level and Sensor Schema context. ................................................................. 35
4.22 Functional Prototype - First iteration of FDD-SV panel. ............................... 36
4.23 Functional Prototype - Second iteration of FDD-SV panel. ............................. 36
4.24 Functional Prototype - Zooming and Tool-tip example. .................................. 37
4.25 Functional Prototype - Building the faces that shape the 3D model .................... 37
4.26 Functional Prototype - Cabril's Dam 3D model with different vertex colours. ........ 38
4.27 Functional Prototype - FDD-SV panel with animated 3D Model ....................... 38
4.28 Functional Prototype - FDD-SV panel with an upper view of the 3D Model .......... 38
4.29 Functional Prototype - New iteration of 3D Model panel ................................. 39
4.30 Functional Prototype - 3D Model panel with difference between observed and calculated behaviour. ................................................................. 39
4.31 Functional Prototype - (a) Date-picker added on top of the water level panel. (b) Date-picker activated. ................................................................. 40
4.32 Functional Prototype - Added sliders for Displacement Multiplier and Animation speed. Added Select All and Clear all buttons for Response in Displacement line chart. ................................. 41
4.33 Selecting a point in time through interaction with the Water Level graph. .......... 42
4.34 Visualizations for the registered response in displacement in 2D and 3D. ......... 42
4.35 Exploration of the 3D scene. ................................................................. 43

5.1 User gender. ......................................................................................... 49
5.2 User age. ............................................................................................ 49
5.3 User level of education. ......................................................................... 50
5.4 Box-plots of each task duration, in seconds. ............................................. 50
5.5 Box-plots of each task errors. ............................................................... 50

A.1 Landing page. ..................................................................................... 63
A.2 Water Level visualization interaction. .................................................... 64
A.3 Water Level Date-Picker. ...................................................................... 64
A.4 Selecting a point in time from Water Level visualization. ....................... 65
A.5 Accelerograms and FFTs panel. ............................................................ 66
A.6 FDD-SV panel. ..................................................................................... 67
A.7 FDD-SV panel interaction. ................................................................. 67
A.8 3D Model panel. .................................................................................. 68
A.9 3D Model panel interaction. ............................................................... 68

xiv
Chapter 1

Introduction

Ensuring that the structure of a dam is in perfect conditions is extremely important. A structural failure in this type of constructions can be catastrophic in many ways. Therefore, it is easy to assume that dam maintenance is a necessary task to carry out through its whole life span. Monitoring dam behaviour is a key aspect in this maintenance task. For these reasons, Laboratório Nacional de Engenharia Civil (LNEC) Concrete Dam Department\(^1\) created their own set of dam monitoring tools. Several years later, LNEC felt that their existing tools for dam monitoring were outdated and inefficient. They needed a new system that relied on newer technologies to perform this task. A visualization used in one of these outdated dam monitoring systems, used by LNEC, is shown in Figure 1.1. The visualization techniques used make it extremely difficult to efficiently interpret monitoring data, which may lead to errors in data readings or even late emergency responses. The lack of intuitive and natural interaction within the system also hinders the ability to explore data in a meaningful way. For these reasons it is crucial to improve the overall effectiveness of this dam monitoring system, taking into account that up-to-date techniques of data representation and interaction can serve an important role for this purpose.

\[\text{Figure 1.1: Visualization tool for dam monitoring used by LNEC.}\]

Visualizing Dam Behaviour (or DamVis) is a project carried out in partnership with LNEC that aims

\(^{1}\text{LNEC’s Concrete Dam Department http://www.lnec.pt/barragens-betao/pt/}\)
to solve these issues. DamVis goal is to provide an updated alternative to dam monitoring, using web technologies and information visualization. LNEC provided data gathered from the Cabril Dam (Figure 1.2) to implement this system. Despite using data from one specific dam, DamVis objective is to be an universal dam monitoring system that can be adapted to most concrete dams, with minor tweaks.

Figure 1.2: Cabril’s Dam picture.

1.1 Motivation

A dam is considered to be of extreme importance to the region it is located on. It is considered that way because of the infrastructure size itself but also for the social and economic impact it has on its surroundings. Usually, constructing a dam and its monitoring is considered to be beneficial to the surrounding population, since it creates a lot of new opportunities. These opportunities can be either creating new job offers or promoting energetic independence for the surrounding community [16].

Despite the advantages mentioned, a dam could be highly dangerous in case of failure. If the structure is compromised and a flood occurs it means that a very costly and time consuming recovery operation for the surrounding region is necessary. Even worse than the negative economical impact is the social impact. It could put the well-being of human and animal life at risk, leading to deaths in some extreme situations. This is why the monitoring and risk analysis of a dam’s structural integrity should be of utmost priority.

Monitoring a dam can be a very difficult task, given its structure size and data complexity. Sensors are used to measure relevant data for the structural response of the dam’s infrastructure. In some cases, tools are used to excite and vibrate constructions under test. However, for dams, these artificial vibration and excitation techniques are just too expensive and impractical which means that they are rarely used [26]. That is why natural values of a dam structural behaviour are used instead of artificial ones. Environmental variables like temperature, reservoir water level and rainfall also have influence
on a dam’s structural behaviour. Unpredictable natural events, like seismic activity or hurricanes, could have an even bigger impact on a dam, which is why the structure should always be in the best condition possible. The main parameters needed to register from a dam are concrete displacement (oscillation and vibration of the structure), hydrodynamic (movement and pressure exercised by the water) and water level.

Traditionally the most common method used to monitor a dam’s integrity is the visual inspection technique. This method has the shortcoming of demanding a significant amount of labour and not giving the possibility to evaluate critical but inaccessible locations [13]. Since this is not a preventive monitoring technique, when a crack or defect is detected its reparation will likely involve a huge amount of work and money. In extreme situations the defect may be irreversible. Visual inspection also relies exclusively on human labour, which is very susceptible to neglect by those involved in the inspection. Automatizing this process using sensors to register concrete oscillation is less likely to have this kind of issues. Relevant data is collected by the sensors and afterwards analysed by one or more specialists. Although this data analysis technique is an improvement it is still not ideal. Raw data interpretation is a challenging task since it is hard to contextualize and relate values registered by the sensors to physical events or movements. This is where Information Visualization (or InfoVis) steps in, helping to solve most of the issues discussed so far.

InfoVis can be found everywhere nowadays. Its main goal is to represent relevant data in a meaningful manner to the user. Visualization is a key component in complex data analysis. It creates encodings of data into visual channels that people can view and assimilate more easily [9]. It enables the user to comprehend data at a higher level, which helps to grasp more complex ideas or larger amounts of information. Using the right visualization techniques is powerful enough to enable analysis of different components while also being able to relate them to each other into a comprehensible and engaging bigger picture [9]. In past decades, visualization techniques have improved and been renewed. With the arrival of problems regarding big data, InfoVis is gaining even more relevance.

Information Visualization is a great way to represent big collections of data in a perceptible manner [23]. This means that users can abstract themselves to a higher level and interact with a full-fledged system that is able to represent data in different ways, exploring or filtering it to their liking and need. Lower level problems, like reading raw data from a table, stop being a matter of concern or inefficiency when interpreting data [23]. Allowing users to disregard low level details empowers them to achieve better results in their original tasks. Understandably, the use of InfoVis can be extremely effective, specially for such sensitive tasks like monitoring dam behaviour that need constant awareness and a fast response in case of imminent failure.

1.2 Objectives

To produce a better final product, some planning must take place. Detailing project objectives is a good way to accomplish it. Given the importance of visualization for this type of systems, DamVis main
objective is:

**Develop a web-based system for visualizing dam behaviour using monitoring data that helps to understand and interpret its structural response.**

The system should be able to:

- Represent external parameters that affect dam behaviour (i.e. reservoir water level).
- Visualize values of structure vibration registered by dam sensors.
- Visualize processed sensor data, using meaningful algorithms to produce derived measures.
- Spatially identify the dam’s structural response for each sensor.

This system can be considered to have two main aspects: Physical Equipments and Software. Physical equipments are the sensors located in Cabril Dam that produce the raw data, which is provided by LNEC. The software component is the one being developed in this thesis, using web-based technologies and the help of InfoVis.

### 1.3 Thesis Outline

This document has six chapters. Chapter 1 is the introduction to the subject, accompanied by the motivation and objectives.

Chapter 2 gives a theoretical background of monitoring dam behaviour, focusing more on the civil engineering context.

Chapter 3 describes what work and research has been done in recent years in the area of InfoVis that can be useful for DamVis. It is divided in Information Visualization Principles, Temporal Data, Hierarchical Data and Monitoring Techniques.

The development process of DamVis is described in Chapter 4. This development process includes requirement and data gathering, system architecture as well as the implementation of the back-end and front-end.

The evaluation of DamVis is presented in Chapter 5. The evaluation process was divided into two main categories: Usability Tests and Case Studies.

In Chapter 6 the conclusions drawn from the process of developing DamVis and its evaluation are exposed, describing its major achievements and future work.

Appendix A displays the screenshots of DamVis final state, presented in this document.
Chapter 2

Theoretical Background

In order to have a meaningful visualization it is mandatory to understand the main goals and issues in the dam behaviour scope. An infrastructure has always some kind of structural response to environmental changes. Mostly, this structural response is very small and imperceptible to the naked eye, even though it is always present. Visual inspection of a dam was used in the past but it is an outdated technique. It does not guarantee that all failures are detected while it also requires a significant amount of labour, exclusively executed by humans. The fact that this technique relies heavily on humans is also a disadvantage due to possible negligence. Locations with extremely difficult or impossible access can not be monitored, which is another major drawback of the visual inspection technique. This method does not support preventive measures, therefore many issues are not detected early enough to be fixed properly, which does not facilitate the prevention of failure [1, 13]. This proves once more that visualization could improve monitoring of dam structural behaviour.

To gather the data needed to evaluate dam behaviour it is imperative to determine what variables are essential to register. The most mentioned variable is concrete displacement [3, 8, 13, 15, 16, 24, 18], which measures the motion or shift of the concrete in a given point of the structure. Dam behaviour is also heavily related to environmental causes, such as hydrostatic load (reservoir water level) and hydrodynamic (pressure applied to the concrete by a body of water) [3, 8, 13, 15, 16, 24]. Air temperature also has a great impact in the construction materials of a dam [3, 13, 15, 24]. Natural catastrophes and other environmental phenomena can have a great impact on a dam’s structure, due to their unpredictability and intensity. Earthquakes or storms with severe rain fall are the natural catastrophes with most impact on a dam [3, 13, 15, 24].

Once determined what values should be registered, it is necessary to choose which tools or sensors are going to capture those values. A specific type of pendulums (accelerometers) are used in real world scenarios to register fluctuation in the concrete position [3]. The measurements registered by a pendulum can be extrapolated to external stimulations like reservoir water level and temperature variation. Non-reversible effects caused by concrete displacement or cracks can be identified if the pendulum position changes over time but does not revert back to the initial position. Other useful sensors are piezometers and an inclinometers [8, 15]. A piezometer is used to measure water level and pressure at
a given location, which is useful to understand the amount of pressure being applied to the concrete in a certain area. An inclinometer is used to measure angles, which can be extrapolated to values similar to those registered by a pendulum. Sensor location on a dam is another key aspect to note. It is necessary to know where the values were observed in order to fully understand them. This way it is possible to identify if some areas of the dam are at risk. The values captured by the sensors are considered to be raw data because they need to be processed in order to produce meaningful values. Raw data is transformed into tabular data by carrying out some data sampling or filtering technique. These techniques may include data set reduction, parameter computation, extraction of derivative values and so on [25]. Captured values can produce helpful data when derived using wavelets transformations [16]. Fast-Fourier Transforms (FFT) are widely used for time-frequency transformations. This kind of operation assumes that the series being transformed is somehow cyclical, meaning that the sensors are excited but revert back to their original state when the excitation ends. Dams can be considered to be a good use case for this technique because it removes noise from the values registered by sensors while also highlighting behaviour that diverts from the expected/usual values.

Vibration of a dam’s structure can be used to measure its behaviour [21]. There are two main methods to do so: Ambient Vibration Tests (AVT) and Forced Vibration Tests (FVT). AVT uses sensors to register vibration caused by the ambient (temperature, wind, hydrodynamics, seismic activity, etc.). On the other hand, FVT uses tools (like a pneumatic hammer) to force vibration while measuring it. For a dam, only the AVT method is considered. It is much simpler and cheaper to execute than FVT because it does not damage the structure or does not require extra tools to force vibration. In the case of dam monitoring, given its mass, it is impractical to move around tools that are big enough to produce a meaningful vibration effect. AVT’s biggest advantage relies on using ambient excitation, which is freely available to vibrate a structure. Nonetheless, given that ambient excitation has an uncontrollable nature this could also be considered as a disadvantage [25].

The structural integrity of a dam is specially important for those that are in need of maintenance or restoration [8]. It is extremely important to assess their condition and prioritize those that need a faster response. For those reasons, risk assessment is fundamental to predict and understand how a dam could behave in a given situation. Usually, to perform this risk assessment, a multidisciplinary team of engineers is involved, such as geological, structural or hydraulic engineers. To accomplish this task, a predictive model is required to guide the risk assessment team on how to evaluate probable behavioural outcomes in the near future [15]. The effort to provide a predicting tool is considerably substantial but critical for dam monitoring, specially those that are already showing irreversible damage. There are different methods that can be used to predict dam behaviour. A statistical and a time-series model can be considered [15] or even an empirical model [4]. Most of these models usually need input parameters that are often selected empirically, which may provide data that is error prone and with a lot of noise [15]. A statistical model provides a purely mathematical modal which considers a series of assumptions contemplating a set of data to generate more data. The assumptions can be described as a collection of probability distributions. This model needs to be tailored to each dam because of their physical characteristics, such as height or thickness, which influences end results [4]. Current conservation state
of the infrastructure, resistance of the construction material and some other values should also be taken into consideration in order to achieve the best prediction results [4]. The time-series model differs from the statistical because it uses discrete-time data (or values with data points) without assumptions [15]. It uses those data points to forecast possible future values. This has the disadvantage of being less flexible to sporadic events such as seismic activity but also means that it is simpler to apply than the statistical model. Lastly, the empirical model tries to correlate eventual dam response with previously gathered information, either for the same construction or other similar dams [4]. This means that comparable infrastructures which suffered from some kind of sporadic event, like an earthquake, are studied in order to try and connect their recorded behaviour to a possible response of the dam under study [4]. The predicted impact of that external event is usually measured in a scale of 5 different levels [4]:

1. Negligible to small damage.
2. Moderate damage.
3. Substantial to high damage.
4. Extremely high damage.
5. Total destruction.

A simplified version of the statistical model can also be considered [3]. It is based on recorded values gathered for that dam so far, instead of a pure mathematical model. This makes it easier to implement than the regular statistical model. However, it has the shortcoming that with this technique, generated statistical coefficients have little to no correlation with the physical response that the structure will produce, making it hard to evaluate expected behaviour for a specific dam. This is where structural identification techniques may become useful, enabling the correlation of statistical coefficients to a dam’s behaviour using its intrinsic characteristics. Those characteristics are height and construction materials used as well as their elasticity and condition. This means that a simple statistical model coupled with structural identification techniques may achieve very similar outputs to a more complex statistical model, purely based on mathematical models [4, 15].

Time is a big factor in dam monitoring [16]. Only with the evolution of time it is possible to relate and understand if the observed structural behaviour of a dam was caused by external variables (air temperature or reservoir water level) and how those changes impact the structure. A case study done in Varosa dam concluded that if no natural catastrophe occurs, time does not have a significant negative impact in the dam infrastructure. This means that deterioration of the structure is probably caused by sporadic natural events. This deterioration is specially apparent if the structure is put through a lot of stress in a short amount of time. In Varosa dam it is possible to state that concrete displacement is highly related to the variation of reservoir water level, since no unexpected natural catastrophe has occurred [16].
Chapter 3

Related Work

Not a lot of academic research has been done combining both dam behaviour monitoring and information visualisation, meaning that there is a shortage of proven methodological approach in this field. Since dams share some characteristics and principles with other large constructions, related work research was broadened to cover this aspect in order to retrieve more useful information.

3.1 Information Visualization Principles

Choosing meaningful visualisation techniques is one of the most important tasks for Information Visualisation (or InfoVis) [23]. This is only possible to do when there is an extended grasp in the field of the problem at hands. Therefore, only after understanding some of the dam’s technicalities (detailed in Chapter 2), one can be capable of delivering powerful visualization tools or techniques. These tailored tools or techniques contribute to better understand the data and its underlying trends, patterns or other type of events while also aiming to be visually pleasing. Visualization can be defined as a key component in complex data analysis, creating encodings of data into visual channels that humans can view and understand [9].

Visualization strives to provide intuitive, immediate and language-independent ways to view data, exploiting the human visual system [9]. The human visual system is considered to be the richest and most immediate of all human senses because of the amount of brain capacity that is devoted to process visual input. With the evolution of technology and arrival of new problems in need to be solved, specially regarding big data, InfoVis is gaining even more relevance. Visualization of individual components is limited to basic data metrics. To overcome this issue, visualization of data that has numerous dimensions or thousands of data points should resort to a mathematical model that reduces data complexity while still capturing its essential characteristics and trends. This means that a powerful visualization system provides some distinct components, isolating visualizations that serve different purposes. Utilizing multiple visualization components grants the ability to relate different metrics or data that otherwise would be difficult with a single component. There are common sets of rules when building monitoring systems that rely on InfoVis but in some specific cases, like dam monitoring, off-the-shelf products are
not enough [5]. Data may need too much processing to adapt it to those off-the-shelf systems, losing some important details or without covering all aspects and needs for specific end-users [5].

Interaction is regarded as one of the most important aspects for this kind of visualization systems [2, 5, 22, 19, 14, 12, 8]. It is considered this way because they are usually used by a multidisciplinary team of engineers [8], which means that different people with distinct tasks will use the system. Obviously one solution could not fit all of their needs, so a system that allows users to filter and explore data, hide or show different components must be considered. This interaction can be treated as user input, which can have multiple forms or granularities, such as hiding or displaying different parameters, changing the time frame or switching to other visualization methods [2]. Visualizations should not be a static demonstration of data and should grant the ability to interact and explore their underlying data [2].

Before displaying the visualization, data should be pre-processed, resulting in a dataset that better adapts its time-frame for the specific visualization (days, mornings, weekdays, months, etc.). For this, concentric circles to represent different events, where the radius is directly related to the number of occurrences, can be used [2]. Interaction with the circles should provide more information that was not possible to represent visually, like a caption or tool-tip with the number of occurrences for the underlying event. There are other common practices for monitoring systems, such as providing a higher-level representation of the monitoring, which enables the exploration to more in depth content [5]. This ensures that only relevant information is displayed for the system as a whole, while also giving the possibility to provide visualizations for specific tasks. System customization could also be very helpful for a better user experience. It can be done by enabling the user to display or hide visualization components on the fly. It offers the possibility of different users with different demands to adapt the system to achieve their individual tasks in a more efficient and simpler manner [5]. User task requirements should help to choose what visualization each component will use. When dealing with more complex or denser visualizations the system’s layout should not be overcrowded, in order to not penalize readability and data fidelity. Tabs could be used to alleviate this density issue but they create yet another problem, by making it much harder to correlate components, since they are not side-by-side any more [5]. Visual analysis should always promote interaction, using parametrization to change the visualization or any other way to explore data at a greater depth [12]. Again, data processing is an important step to help extract more meaningful information rather then just raw numbers. However, it should be done in a concious manner because it is easy to produce processed data that conceals lower granularity events or patterns [12].

3.2 Temporal Data

Displaying time associated data is extremely important to understand patterns, sporadic events and the overall data trend. If the data is known to be cyclical, like weekly or monthly data, radar charts are a good choice [9]. On the other hand, if it is not cyclical but still has time associated, a heat map is a good way to encode it [9, 22, 11]. A heat map conveys progression of the data through time dividing it into equal rectangular shapes, each one corresponding to the defined time span, enabling the possibility to compare them [11]. Colour encoding is then utilized to paint the rectangles. Each colour represents a
value for the encoded data, making it easy to search for trends or clusters in each time interval. The defined time interval should be chosen carefully. Too small of an interval and the visualization ends up with a lot of noise, too big of an interval and some relations or tendencies might get hidden in the process [11]. To alleviate this issue an adjustable threshold for the heat map can be used, which helps improve the data granularity. This results in a more meaningful representation, depending on the user’s usage scenario [22]. If a daily time frame is chosen for the heat map it can be assumed that it takes the form of a calendar, which results in a compact and intuitive visual representation that is good to relate patterns in a weekly or monthly basis [9], as shown in Figure 3.1.

Scatter plots or bar charts can also be useful as complementary views to heat maps. They enable visualization of different data metrics that may not be fully exposed in heat maps [11]. These complementary charts can be global or individual to each time interval and can be displayed when a user interacts with the heat map [22].

Another way to view data at a higher level, giving the chance to showcase trends and outliers, is the stacked line chart (also known as river view) [12, 9]. This visualization is primarily used for cumulative values and is similar to a stacked bar chart but for continuous data, represented in Figure 3.2.
It may be useful to understand when some correlated data values have a significant difference between them as well as their whole progression through time [9]. Arc diagrams, represented in Figure 3.3, is another method to represent cumulative data. It provides an immediate way to identify overlapped events and their relation to each other, which can be helpful for those described scenarios. [12].

Time-frequency data representation is commonly used in analysis of acoustic signals and seismic data, usually employing spectrograms [10, 1]. Different families of algorithms exist to produce this type of representation, Fourier transform or wavelet transforms being the most used. Common graphs with a time axis do not represent well frequency which is bad for some kinds of data. Spectrograms help in that matter, enabling the understanding of frequency progressions through time, while also providing means to view clusters and create relations between them, like the one presented in Figure 3.4 [10].

Spectrograms have the disadvantage of not being able to specify the exact value of the represented signal at a point in time.

### 3.3 Hierarchical Data

Hierarchy visualization is an effective tool to process complex information that is related to each other. It is also helpful to understand the data in a more natural way [14]. Using the node-link technique, more
specifically the rooted tree diagram, allows users to compare different nodes that are on the same layer at a given moment. It can be used either in a vertical or horizontal direction. Each node has a bar chart associated that illustrates the corresponding time-series values, like what is represented in Figure 3.5.

![Figure 3.5: Rooted Tree Diagram with complementary bar charts for some nodes](image)

Each bar represents a point in time and all of them have the same width. This helps to easily compare values between other nodes in the same point in time or the progression of values for the current node. It is also possible to interact with this visualization by expanding and compressing the hierarchy (either by node or level), by changing the time range of the time-series (either by using a sliding window or zooming technique) and also by displaying detailed information with a hovering interaction [14]. This hierarchical visualization may be useful to show data for related sensors, if they have an underlying hierarchy.

### 3.4 Monitoring Techniques

As stated previously, the ability to predict behaviour is key in a monitoring system, which means that the way those predicted values are displayed is crucial. Prediction models can be validated using subsets of data from a past time, comparing data that was previously captured with the predicted values for the same time frame. Matching the shift between these predicted values with real data helps to understand if the model in use is a good fit or not [17]. A simple line chart with two colour coded lines, one for the real and other for the predicted values, such as in Figure 3.6, is enough to understand if the difference between them is negligible or if other prediction model should be considered.

It should be taken into account that datasets, utilized as input for the prediction model with a small
Figure 3.6: Comparison between predicted (red line) and the real values registered (blue line) [17]

time frame, hurt its output effectiveness, since this lack of real data may skew the predicted values [17].

Line charts can be used to monitor the dam response as a whole, with thresholds that define whether
this response is normal or abnormal [15]. If the response is said to be abnormal, complementary bar
charts are displayed to check which sensors are behaving unexpectedly. When the sensor with unusual
behaviour is detected the user should assess if this abnormal value was produced because of a faulty
sensor or local damage in the dam. Once that assessment is made the appropriate action can ensue,
either to repair/replace the faulty sensor or to further inspect the dam’s affect area and work from there.
Line charts are also used to represent external variables that highly affect a dam’s behaviour, like air
temperature or reservoir water level [15]. This is done with distinct line charts, one for each variable,
displayed side by side to easily relate them.

Clustering with colour coded category is also used in order to show different types of sensors [1].
This means that scatter plots are good to relate different types of sensors if each dot represents one
sensor and its colour encoding depicts sensor type.

Some visualizations try to map real life physical characteristics. As already said, some values are
only meaningful when connected to a location. For example, if a sensor location is not known, then
the values it captures are not that useful to monitor because if abnormal behaviour is identified it is
much harder to act accordingly, given the fact that the location where it was registered is unknown.
For those reasons, usage of schematic representations for the dam are helpful, displaying location of
relevant elements, such as sensors [19]. The ability of overlapping those schematic representations
and corresponding real life pictures of the area may also help the user to have a better context of those
schemas.

Combining different visualization techniques with these schematic representations can provide a
powerful tool to monitor a dam. Usage of a sideways schema of the dam (upstream reservoir on one
side and downstream reservoir on the other) combined with a spectrogram, shown in Figure 3.7, is one
of those approaches [24]. This method facilitates the representation of multiple sensors location while also giving information about acquired values. It helps to understand which areas of the dam are going through more stress and with this technique it is possible to easily identify which areas are being affected by abnormal behaviour or which sensor is malfunctioning [24]. Other approach of this technique is to use an aerial view instead of a sideways view, shown in Figure 3.8 [25]. The aerial view represents the arch of the dam and colours it like a spectrogram. Areas with higher displacement are highlighted with red. Using more than one graph side by side, representing different points in time, makes users able to assess if the structure was put through too much stress in that time span. Both techniques have a similar goal but one makes it easier to locate sensor vertically where the other locates them horizontally.

Some 3D techniques may also be used, like a virtual 3D terrain [26]. The ability to explore a 3D environment, simulating real-world constructions with spatial information, could be useful to observe patterns with location awareness. The exploration of the three-dimensional scene may be useful to comprehend how different sides of the dam behave at the same point in time. 3D representations can use animations to simulate the observed dam behaviour in different points in time. This way it can be possible to determine if distinct points of the dam are being put through more stress than others. If that it is true, than an assessment is needed to those specific locations to evaluate the risk of failure.

### 3.5 Discussion

As stated in the beginning of this section, lack of information regarding dam behaviour visualization is a problem. Very little to no real research was done to try and solve this issue from a InfoVis point of view, which is a handicap when trying to develop a powerful and meaningful tool. Some researches
that focus mostly on the dam behaviour monitoring suggest visualization techniques. However, some of those techniques are considered old, outdated or lack user interaction which may hinder the visualization effectiveness [9].

Every visualization has advantages and disadvantages. Those who build a visualization tool are in charge of choosing the right techniques, depending on the context and tasks that are to be fulfilled. For dam behaviour monitoring, as it was already stated, the physical location of sensors is extremely important, in order to be able to understand which areas correspond to the represented values. This means that the visualization with a dam’s schematic representation are very effective in this context, however they do not show changes through time. Other visualization techniques that can represent data evolution through time, like a spectrogram, scatter plot or heat map, could complement the dam schematic to overcome this drawback. It is relevant to provide visualizations that support multiple data represented at the same time, such as values from different sensors, to easily correlate them, which could be achieved using river view, arc diagram, scatter plot or line charts. Heat maps and spectrograms can be considered to be representations with similar function, since both can represent the variation of a sensor through time but without the possibility of pinpointing the exact value at a given instant. On the other hand, river view, arc diagram and scatter plot can also be considered to have similar function. All of them enable multiple data (i.e. various sensors) to be displayed on a single representation. Bar and line charts should also be considered since they could be useful in almost every situation, given their versatility and effectiveness. Table 3.1 summarizes some characteristics of the representations discussed so far and their applications.

Previously it was already referred that a lot of work was done in the Information Visualization area in the past years [9]. This recent work could bring a lot of potential to a new and refreshed way to monitor dam behaviour, not only with new visualization techniques but also with different means of interaction or the usage of complementary visualizations in order to bring the system potential to its maximum. Up to date standards and researches done both in the InfoVis and dam monitoring areas will also contribute in the resolution of this real-world problem. Therefore, it can be considered that DamVis is trying to fulfil this lack of dam behaviour monitoring systems using InfoVis.

<table>
<thead>
<tr>
<th>Visualization Type</th>
<th>Value Quantification</th>
<th>Multiple Data</th>
<th>Change Through Time</th>
<th>Physical Position</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bar Chart</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Line Chart</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Heat map</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Spectrogram</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>River View</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Arc Diagram</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Scatter Plot</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Static Dam Schema with Spectrogram</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>3D Dam Schema with Spectrogram</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Table 3.1: Comparison between discussed visualization techniques.
Chapter 4

DamVis - Visualizing Dam Behaviour

DamVis, as said before, is a system for visualizing dam behaviour that can be tailored to most concrete
dams. Data from Cabril’s dam was used in this DamVis implementation. LNEC’s Concrete Dam De-
partment¹ is responsible for maintaining and monitoring it [18]. They also have made the development
of this system feasible, providing the data and expert knowledge. Therefore, it is possible to affirm that
LNEC’s Concrete Dam Department is the stakeholder for DamVis.

An incremental iterative approach was used in all the process of developing DamVis. This approach
grants flexibility and greater ease to adjust and adapt the work developed to the new needs that arise
over time[6, 20]. This incremental iterative approach could be outlined in three major stages:

1. Requirements and Data gathering.
2. Low Fidelity Prototypes.
3. Functional Prototypes.

The first stage, as in most projects where stakeholders are involved, required to figure out what are
their needs and what tasks should the system accomplish. In this case, the data used in the system
was also collected at this stage. The next step was to build low fidelity prototypes. They are a quick
and easy way to convey and discuss ideas about the layout and visualization techniques to be used at a
higher-level [6, 20]. Functional prototypes were the next stage and, as the name suggests, it was where
all the planning started to take a tangible form. They allowed us to test and interact with the visualization
techniques suggested in the low fidelity prototypes, as well as understanding if they actually fitted our
tasks or not. Since they were functional prototypes, the whole system architecture also started to be
implemented at this stage. The final state of DamVis presented in this document is the latest iteration of
the functional prototypes. In each of these stages one or more testing phase was carried out. This helps
to validate and receive feedback about the work that had been developed more often and in smaller
amounts, which helped to tailor the system to our stakeholders needs.

Section 4.1 details the first stage of the whole process of developing DamVis, requirement and data
gathering. Section 4.2 describes the system architecture, including the back-end (Section 4.2.1) and

¹LNEC’s Concrete Dam Department http://www.lnec.pt/barragens-betao/pt/
the front-end (Section 4.2.2). The front-end section also encompasses the low fidelity prototypes and functional prototypes development stages (Section 4.2.2 and Section 4.2.2 respectively).

4.1 Requirements and Data Gathering

The first stage in the process of developing DamVis, as in most projects where stakeholders are involved, was to meet with LNEC’s Concrete Dam Department and discuss the main goals and functionalities expected of this system. Another predominant purpose of this first stage was to gather requirements and assimilate expert knowledge of the specificities of dam monitoring. The major requirements that the system should meet were defined as:

- Watch the variation in the water-level.
- Choose a point in time to:
  - Visualize values acquired by the sensors.
  - Visualize a Fast Fourier Transform (FFT) of those values.
  - Visualize Frequency Domain Decomposition (FDD) for all sensors at that point in time.

As described previously, water-level is one of the most important variables in dam monitoring, so it is imperative to visualize it in order to understand the dam’s behaviour in a broader scope [18]. A more in-depth monitoring is achieved by narrowing the time frame and visualizing the values registered by the sensors. Accelerograms and graphs containing the FFT values are used if there is a need to view individual sensor values. FDD is a modal analysis of the structure’s health and therefore utilizes data from all sensors in the selected time frame [18]. It builds matrices of complex numbers, with the values obtained in the FFT and afterwards it can be factorized using some other algorithms, like Frequency Domain Decomposition with Random Decrement (FDD-RD) or Frequency Domain Decomposition - Singular Value Decomposition (FDD-SV) which will use the same visualization techniques but with slightly different data. A three-dimensional view of the dam with values outputted by FDD would also be helpful but was considered to be of less priority than the other visualizations mentioned above.

All data used in this system was provided by LNEC and represents real values registered in Cabril’s dam. The data can be divided in three distinct categories:

- 3D Mesh of the Dam.
- Water level.
- Accelerometer Data.

The 3D Mesh, as the name suggests, is a mesh of points in a three-dimensional coordinate system of Cabril’s Dam itself and the surrounding foundation and reservoir. Another category of data is the recorded water level, in meters, for each hour between January the 1st of 2014 and October the 24th of 2014. Lastly, the other category of data contains the values acquired by each of the 19 accelerometers in
Cabril’s Dam, used to measure concrete displacement. From those 19 sensors, 16 are uni-axial, which means that they only register concrete displacement in one axis and the remaining 3 are three-axial. Sensor location is represented in Figure 4.1.

Figure 4.1: Cabril dam and the location of its sensors. [18]

All data files provided by LNEC had a .dad file extension, which is similar to a .csv file but with a space-separator delimiter, meaning that the data is stored in a tabular manner. The file containing the 3D Point Mesh of the Dam contains three arrays, one for each three-dimensional axis and its corresponding values. The file with water level data stores the water level registered each hour from January 1st until the end of October 23rd, as shown in Table 4.1. The files with data registered by the sensors begin with a line defining which year, month, day, hour and minute it corresponds to. After that, sensor data is displayed in 25 columns, one column per uni-axial sensor (from 1 to 16) and three columns per three-axial sensor (X, Y and Z columns for sensor 17, 18 and 19), like shown on Table 4.2. Each of these files contains the values registered by each sensor per hour with 50 samples per second. This means that, per hour, 180,000 values are registered per uni-axial sensor and three times as that per three-axial sensor, resulting in 4.5 million values present in each of these files.

<table>
<thead>
<tr>
<th>Year</th>
<th>Month</th>
<th>Day</th>
<th>Hour</th>
<th>Minute</th>
<th>Water Level (meters)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2014</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>275.14</td>
</tr>
<tr>
<td>2014</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>275.18</td>
</tr>
<tr>
<td>2014</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>275.23</td>
</tr>
<tr>
<td>2014</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>0</td>
<td>275.28</td>
</tr>
<tr>
<td>2014</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>0</td>
<td>275.3</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

Table 4.1: Schematic of the data in the water level file.
Table 4.2: Schematic of the data in the sensor data files.

4.2 System Architecture

From an early stage it was determined that this new monitoring tool should be developed with web technologies, using JavaScript libraries, replacing the existing MATLAB\textsuperscript{2} monitoring tool. This choice was made due to the amount of work done in open-source visualization libraries in the last few years. It facilitates our work because it removes the need to develop a solution from scratch, being able to use templates for the system layout or visualization libraries.

An usual client-server architecture was considered for this system. The back-end (or server-side), described in Section 4.2.1, should process and deliver data to the front-end (or client-side), as shown in Figure 4.2.

![Figure 4.2: Basic scheme of DamVis architecture.](image)

The front-end will be a single-page application that requests data to the back-end and displays it using some visualization techniques, described in Section 4.2.2. The communication between both sides is made through a web Application Programming Interface (API) with standard Hypertext Transfer Protocol (HTTP) requests. The API defines a set of paths, accessible through these HTTP requests in order to exchange data in the JavaScript Object Notation (JSON) format. Therefore, the back-end is responsible for handling requests incoming from the front-end, which may involve data processing and delivering it in a known format, such as JSON. After a certain user interaction with the application on the front-end, a corresponding data request will be sent to the back-end. Once the response from the back-end reaches the front-end, it will manipulate the data to form the visualizations, corresponding to the user request.

This kind of architecture is flexible enough to be scalable if later on new features or methods need

\textsuperscript{2}MATLAB: https://www.mathworks.com/products/matlab.html
to be added. The separation between data processing logic on the back-end and visualization or application logic on the front-end facilitates the project organization and future work. Even though the web API, at this point, is only used internally it can grant to the public the opportunity to develop an entirely different system using the same set of data.

4.2.1 Back-End

DamVis back-end is developed in JavaScript, more specifically using Node.js\(^3\). Node.js is a run-time environment for executing JavaScript server-side code. It is also open-source and cross-platform. Node.js provides a package ecosystem, Node Package Manager\(^4\) (npm), which is one of the largest existing repositories for open source libraries. All the dependency management is also handled by Node.js, which is useful when using several packages from different sources.

The entry-point to start the execution of DamVis instantiates the server and other configuration variables, such as API paths. Hapi\(^5\) was the chosen package to help develop the server logic. This package allows to build a server that responds to the defined paths with a great degree of customization. For DamVis, only requests using the GET method of the HTTP protocol are considered, since the front-end only needs to read data. The paths available to be queried are the ones defined in Figure 4.3.

![Figure 4.3: Paths that the back-end handles.](image)

The root path returns the single-page application in HTML that contains all the front-end logic and implementation. Responding to requests with HTML pages is possible by using the Handlebars\(^6\) package, available in the NPM repository. The remaining three usable paths to communicate with the server belong to the API. This means that their paths are prefixed with /api/. Each of them correspond to the three data categories described in Section 4.1. One for the 3D Mesh Points, one for the Water Level

---

\(^3\)Node.js: https://nodejs.org/en/
\(^5\)Hapi: https://hapijs.com
\(^6\)Handlebars package: https://www.npmjs.com/package/handlebars
and the last one for Sensor Data, given a specific date. When the requests made to these paths are completed successfully a JSON object or array is returned with the corresponding data.

The 3D Mesh Model path reads its corresponding file and returns a JSON array with the 3D points of the dam.

The Water Level path reads its corresponding file and returns a JSON array of objects, structured like what was shown in Table 4.1.

The Sensor Data path is more complex, since it is the one that processes all the raw sensor data into the derived values used in the visualizations. This path only accepts requests when a date is specified, in order to fetch the corresponding data file. The outputted data from this path can be divided into three categories:

1. Raw Sensor Data.
2. Fast-Fourier Transforms (FFT).

Sensor raw data is the base for both of the remaining categories, since FFTs depend on it to be calculated and FDD-SV on the FFTs.

After validating that the requested date exists, the raw sensor data is extracted from the corresponding file into an array, as shown in Table 4.2. Each three-axial sensor is treated as three separate uni-axial sensors, which means that there are a total of 25 sensor values per file. The next step is to calculate the FFT for each sensor, that is done using the NPM package FFT. 25 FFTs are calculated with the help of this package, which returns complex numbers, like what is shown in Figure 4.4. JavaScript does not natively support complex numbers so the output of the FFT package represents them in an array, where the elements with even index represent the real parts and the odd index the imaginary parts. The Mathjs is an extensive mathematics library for Node.js that supports complex numbers, fractions, matrices (among others) and operations between them. This package is used to objectify the complex numbers returned by the FFT package in a more structured way, instead of an integer array.

Once the FFT operation is completed it is now possible to process the FDD-SV values. This algorithm is only applied to the 16 uni-axial sensors. It is completed in two steps. This first step is to generate the Frequency Domain Decomposition values. The second step is to factorize the values obtained in the first step by applying the Singular Value Decomposition algorithm to it, represented in Figure 4.5. The FDD is achieved by building 16 by 16 matrices, each one of its elements is produced by doing the dot multiplication of a vector of FFTs, that was previously multiplied with the frequency of the samples, by a conjugated vector of complex numbers. After calculating the FDD matrices it is now possible to apply the Singular Value Decomposition algorithm. The utilization of the Node-SVD package was considered to help in this task but was quickly discarded. It was the only package registered in the NPM repository to

---

7 One per uni-axial sensor (16 sensors) plus three per three-axial sensor (3 sensors), which is equals to 25.
8 FFT package: https://www.npmjs.com/package/fft
9 Mathjs Package: https://www.npmjs.com/package/mathjs
10 Node-SVD package: https://www.npmjs.com/package/node-svd
implement this algorithm but it was not possible to use it. The Node-SVD package is only able to run on Linux systems, even though Node.js aims to be cross-platform. This issue emerged from the fact that JavaScript does not natively support matrices and matrix operations, so this package was using a library developed in the C programming language, called SVDLibC\textsuperscript{11}, that implemented the Singular Value Decomposition algorithm and that was afterwards compiled to code compatible with JavaScript. SVDLibC uses native functions, provided by header files, that are only available in the Linux OS. The DamVis development environment was based of a Windows machine, due to stakeholders request and limitations, which made the Node-SVD module unusable for this development environment. To overcome this issue the original Node-SVD packaged was forked and re-written into a Windows compatible version. This was done by removing the referenced SVDLibC header files that were unavailable on Windows OS. The new module, to date, was not yet registered in the NPM registry but it is available to be installed in a Node.js project by using the following url: \url{https://github.com/JMoura5/node-svd}.

For each matrix provided to the SVD module it returned two variables: U and V. U is a singular value of the factorization of the FDD matrix and V an array, in this case, of 16 elements that correspond to the factorization for each sensor individually. This is useful to produce values for the Dam as a whole (U) or per sensor, with location awareness (V). An alternative version of the FDD-SV values is produced where a windowed median-filter algorithm is applied. Applying this filter smooths the produced values, removing noisy data and reducing the number of entries, making it more manageable to handle and visualize.

\textsuperscript{11}SVDLibC: \url{http://tedlab.mit.edu/~dr/SVDLIBC/}
In order to improve performance and avoid repeatedly and unnecessarily calculating the same values, the processed sensor data is stored persistently in JSON files with the same name as the file with corresponding raw data. When the server receives a request to the sensor data of a specific date it first checks if it was already processed (first it searches for the JSON file and then the .dad file), in order to avoid doing the same demanding calculations over and over again. Only if the data is still unprocessed this whole processing mechanism is triggered, that stores the produced values in a persistent way afterwards. Another method to improve overall system performance is the ability to filter which sensor data is returned, reducing the amount of data to be transferred between server and client. The sensor data API path is able to filter the returned data by using a query string parameter called "included" that accepts the values "raw" (for the raw sensor data), "fft" (for the raw data and FFT values) and "fddsv" (for the FDD-SV values).

### 4.2.2 Front-End

DamVis front-end was developed in a typical web-application way. It consists of HTML, CSS and JavaScript. Some JavaScript libraries like jQuery\(^{12}\), D3.js\(^{13}\) and Three.js\(^{14}\) were used to help in the implementation of the visualizations and user interaction. The front-end was built in the same incremen-

---

\(^{12}\)jQuery: https://jquery.com/

\(^{13}\)D3.js: https://d3js.org/

\(^{14}\)Three.js: https://threejs.org/
tal iterative approach with lots of feedback from LNEC's Concrete Dam Department. As stated previously in this chapter, after gathering the requirements, low fidelity prototypes (LFP) were developed. Only after validating these LFP with the stakeholders a functional version of DamVis was implemented, the functional prototypes. These functional prototypes were also developed in an iterative approach, adding new visualizations and interactions always with validation from the stakeholders. With this approach it is guaranteed that the final version is useful to LNEC's Concrete Dam Department and fits their needs without requiring changing a lot of things, since the end-users were always aware of the system's state and evolution.

Low Fidelity Prototypes

Following the incremental iterative approach, the next step after gathering the system requirements was to build some low fidelity prototypes (LFP). These kind of prototypes are useful to represent the layout and visualization techniques at a higher level in a quick and easy manner [6, 20]. It is an efficient way to convey ideas and discuss them with the stakeholders, in this case, LNEC's Concrete Dam Department.

Taking into consideration the requirements gathered earlier, a first version of LFP was developed. DamVis was envisioned as a single-page web-application. A dashboard layout was considered for the system to interact with the different tasks it should perform. This kind of layout consists on a sidebar, with a list of the different panels and visualisations that the system offers, and a content area on its side, changing accordingly to user interaction. The landing page (or the primary panel) should show the observed water level of the dam's reservoir, as shown in Figure 4.6. Five buttons are present that allow to regulate the time-span of the chosen point in time, the lowest being hourly. Water level is a cyclical event (the water level is usually lower in the summer months and higher in the winter months) and radar charts are good when used to represent cyclical data. Keeping this in mind, a radar chart was added, adjacent to the line chart, that will represent water level in a cyclical way. A 2D schematic view of the dam with sensor location is also present in this figure. It allows to have spatial awareness of sensor location throughout the dam.

After choosing a point in time, either by clicking in the line or radar chart, this 2D dam schema allows to choose which individual sensor data should be visualized. Clicking the Accelerogram tab will display the respective visualizations, as shown in Figure 4.7. It consists on accelerograms (raw sensor data) in a line chart and Fast-Fourier Transform in a spectrogram for each selected sensor. This tab would also display the water level and sensor location schema visualization as in the last tab. They serve as a time-line and context of the sensor data that is being exhibited.

FDD, FDD(RD), FDD-SV and FDD(RD)-SV tabs consist on the same visualizations with the underlying data being slightly different. Since the data used for these prototypes is mocked and does not have real meaning, only one LFP is needed to represent each of these tabs. Figure 4.8 depictes an example of what each FDD tab should contain. There are three line charts, one for the Response Spectrum (values outputted by the FDD algorithm), one for Modal Assurance Criterion (statistical modal to analyse difference between a set of values) and another for the Displacement Response (decomposition of the FDD data into 16 distinct array of values, each of them corresponding to an uni-axial sensor). Context on
### DamVis

<table>
<thead>
<tr>
<th>Water Level</th>
<th>Accelerograms</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>FDD</td>
</tr>
<tr>
<td></td>
<td>FDD(RD)</td>
</tr>
<tr>
<td></td>
<td>FDD-SV</td>
</tr>
<tr>
<td></td>
<td>FDD(RD)-SV</td>
</tr>
<tr>
<td></td>
<td>3D Model</td>
</tr>
</tbody>
</table>

#### Figure 4.6: LFP - Home Page

#### Figure 4.7: LFP - Accelerograms and FFTs panel
the visualized data is still being provided by the water level representations and sensor location schema, as in the previous panel.

Even though the 3D Model representation of the dam was considered to be an optional part of DamVis, only to be implemented after assuring that the other visualizations were completed, it was still considered in the LFP, represented in Figure 4.9. The objective of this panel was to produce a 3D visualization of the dam that reproduces the calculated displacement response by the FDD algorithm. This way the observed concrete displacement can be represented with a spatial reference of each sensor, enabling to detect which areas of the dam are being put through more stress.

After completing the LFPs a meeting to validate them took place. This validation was performed in an informal way, where an open debate with exchange of ideas emerged with our LNEC supervisor (considered an end-user and stakeholder of DamVis). The LFPs were presented and some ideas for modifications appeared. These changes included different versions of the same visualization techniques or their replacement for something else, due to redundancy or inefficiency executing a specific task. After this feedback, another version of the LFPs was produced. Some of the modifications include the removal of the water level radar chart or the ability to change the time frame of the visualized data. The stakeholders decided that it did not make sense to view data in a time frame other than hourly. Removing the water level radar chart would also help keeping the focus on the more relevant line chart representation. It also helps to simplify the system, since it decreases the amount of data being displayed at once. The modifications to LFP of the water level panel are presented in Figure 4.10.

The Accelerogram tab also suffered a few changes in this new iteration. The spectrogram used to
Figure 4.9: LFP - 3D Model Finite Element Panel

Figure 4.10: LFP - Second Version - Home Page
visualize the FFTs was removed, being replaced with a line chart, as shown in Figure 4.11. This was done, once more, to meet our stakeholders needs. They felt that line charts would facilitate reading exact values of the FFTs, which was more useful to them. The usage of spectrograms was therefore discarded because they have the drawback of hindering the ability to extract exact values, despite usually being helpful to display frequencies over time.

The four FDD tabs were reduced to only one, the FDD-SV. The difference between them was not significant enough to implement them all. It was decided that the Singular Value Decomposition variation of FDD was the most meaningful for the stakeholders needs. Besides that, the visualizations present in this panel were also modified, represented in Figure 4.12. It was considered that the Model Assurance Criterion visualization was expendable and should be replaced by the 3D model of the dam, displaying the Response Displacement in a 3D spatial representation. This way the exhibition of Response Displacement was being complemented by two distinct visualizations, the line chart with 16 lines (each corresponding to the calculated displacement response per uni-axial sensor) and their spatial reference through the 3D model. This 3D representation should be animated and colour coded in order to represent the displacement for each sensor. Using a spectrogram like colour code in the 3D model and displacing the points in the dam that correspond to a sensor location, would achieve this goal of visualizing displacement in a three-dimensional way. The addition of a radar chart with the cyclical representation of the Response Spectrum was also considered.

The 3D Finite Element Model panel also suffered some adjustments, due to the changes in this new LFP iteration. Now this panel should be responsible for displaying two 3D Models with different
One should use the data produced by the FDD-SV algorithm, which is the same as described for the 3D model in the previous panel, while the other 3D model should depict data of the predicted dam behaviour. As referenced in Section 3.4, predicting dam behaviour is a crucial monitoring technique. The predicted dam behaviour data is calculated through the usage of a statistical model that receives an input of different variables related to the dam in cause. Relating the calculated and the observed behaviour is therefore of extreme importance to validate the statistical model being used. If these two differ too much from each other it can mean one of two things: the statistical model used is not good enough or some unpredictable event has occurred (e.g. natural disasters). To give a bit more context to the data that is being viewed in this panel, the Response Spectrum used in the FDD-SV panel is also used here. A complementary comparison method of both the observed and the calculated behaviour can also be useful. This other comparison method can be achieved by using a single 3D model that calculates the difference between the calculated and observed behaviour. If the single 3D model barely changes or moves it means that the calculated and observed behaviour are similar, therefore the statistical model is good and no unpredicted event has occurred.

Another meeting with our stakeholders occurred to validate this new version of the LFPs. The visualizations present in this iteration could now fulfil the tasks the system set out to accomplish. From this point on, the next stage in this iterative process (building the functional prototypes), could start.
After validating the second version of the Low Fidelity Prototypes, the next step was to build a functional prototype. As said in the beginning of this section, a single-page application using HTML, CSS and JavaScript was considered to achieve this purpose. The first step to develop the functional prototype was to build an HTML template that included Bootstrap\textsuperscript{15}. Bootstrap is an open-source toolkit that provides a set of CSS and JavaScript files with a responsive grid system, custom components and other plugins built using jQuery. This template divided the page into two main sections: Sidebar and Content. The Content section is dynamic and should change according to user interaction with the Sidebar. Using D3.js and the data returned by the API Water Level node, a first version of the Water Level panel is shown on Figure 4.14. It consists of a line chart that displays the water level variation through time.

One of the main potentialities of DamVis is to explore the data through a more natural interaction, on the contrary of what LNEC’s previous monitoring tool provided. Considering this, the next iteration of the Water Level visualization was to implement user interaction, like a mouse hover to present a tool-tip, demonstrated in Figure 4.15.

When the Water Level graph is clicked, the corresponding point in time is selected and the Sensor Schema appears, enabling the user to choose which sensors should be displayed in the Accelerogram panel, presented in Figure 4.16. The title above the sensor schema displays the selected time frame.

The next step was to start the Accelerograms and FFT panel, which consists in two line charts side-by-side, for each selected sensor, developed with D3.js. The first one displays the accelerograms, with

\textsuperscript{15}Bootstrap: \url{http://getbootstrap.com}
Figure 4.14: Functional Prototype - First iteration of the Water Level Panel.

Figure 4.15: Functional Prototype - Water Level with tool-tip.

Figure 4.16: Functional Prototype - Point in time selected and Sensor Schema appeared.
the raw sensor data, and the second one with the corresponding FFT, all data being returned by the back-end API. Another informal meeting took place to validate the FP so far. Again, this validation was performed in an informal way, where an open debate with exchange of ideas emerged with our LNEC supervisor. After some feedback, it was decided that another set of two line charts could be useful when two or more sensors are selected. They would display the maximum, minimum and median values for each sample of the selected sensors. This provides a comparison tool for the accelerograms and FFTs being exhibited, represented in Figure 4.17.

Also regarding the feedback from the last meeting, it was decided that the content layout was not optimal. Screen real estate was being wasted with the water level and sensor schema representations, due to their dimensions, as shown in Figure 4.16. For this reason, they were reduced in size and put side-by-side, as in Figure 4.18. Sensor schema is now always visible, like water level, but disabled from any user action if no point in time is selected. Sensor Schema opacity is reduced and all mouse events ignored when it is disabled. Also in Figure 4.18 it is noticeable a visual overhaul of the template’s look and feel. This makeover was done to make the functional prototype feel more polished and closer to a finished product, while also making it more visually pleasing.

Soon after, new ways of displaying feedback to the user after its interaction with the system were implemented, like what is represented in Figure 4.19. A vertical line that touches the X axis complements the tool-tip. If a point in time is selected then this line changes colour and does not disappear until another hour is selected. The title above the water level graph also changes according to the selected hour.

Predicting unwanted user behaviour was also considered. If the user tries to interact with the application, like navigating to any tab other than Water Level, without choosing any point in time, an error panel is displayed, represented in Figure 4.20. The error panel describes why the expected visualizations are
Figure 4.18: Functional Prototype - Water Level and Sensor Schema new positioning.

Figure 4.19: Functional Prototype - (a) Mouse hover with tool-tip (b) Mouse click
not being shown and what the user should do in order to display them.

With this improved layout, the accelerograms and FFTs could be properly displayed beneath the water level and sensor schema, like what was designed in the LFPs, after selecting a point in time. Figure 4.21 shows what it looks like after choosing a point in time and receiving the raw sensor and FFT data from the back-end.

The next step in this iterative process was to build the FDD-SV panel. Figure 4.22 shows its first iteration, with the Response Spectrum and Response Displacement. At this point the 3D model and radar chart are still missing. It is easy to see that the Response Displacement has very noisy data and does not produce a meaningful visualization. This is where the windowed median-filter applied in the back-end becomes handy, since it will reduce the noise and help achieve a more useful visualization.

After discussing this stage of the FDD-SV panel with the stakeholders it was decided, by them, that the radar chart should be expandable, for reasons similar to those that justified the removal of the water level radar chart in the low fidelity prototypes. Another observation made had to do with the usefulness of showing or hiding specific lines in the 2D Response in Displacement visualization.
Each line corresponds to an uni-axial sensor, being able to show or hide each one of them could allow our stakeholders to better interpret its data. Therefore, a new version of this panel was developed, represented in Figure 4.23. Switch style buttons with the corresponding sensor number were added, granting the ability to show or hide each displacement response line.

All graphs grant the user the ability to explore the represented data with interactions such as zoom, pan and mouse hovers. A semantic zooming technique is applied to each graph, which means that the zoom only affects the X axis, which means that the graph shape and Y axis do not change. When zoom is applied, one can pan, with a mechanism to side scroll through the X axis without the need to decrease zoom level. This is done by dragging the graph sideways. The mouse hover interaction provides a tool-tip for the hovered data. A tool-tip can serve various purposes but in DamVis is used to provide details or context on the hovered data, such as an exact value for the corresponding graph. In Figure 4.24 it is possible to observe this behaviour, where a zoom was applied to the Water Level graph and a tool-tip is
It was at this point that the 3D model was developed, using Three.js. After gathering the 3D model points, returned by the back-end API, a custom mesh of polygons, displaying a 3D representation of Cabril's dam, was developed. This custom 3D mesh is built by using a set of 8 three-dimensional points, connecting them to form 6 polygons, which then are used to shape a 3D plane (or parallelepiped face), similar to what is represented in Figure 4.25.

After connecting all points and faces of the mesh, the 3D Model of Cabril's dam is completed, as shown in Figure 4.26. The displayed 3D model presents colour due to tests being carried out with different vertex colours. These vertex colours will be useful later on to apply the colour code of spectrograms.

Since Three.js offers the ability to animate meshes, the next step was to apply the displacement calculated in FDD-SV (which is displayed in the 2D Response Displacement graph) onto the 3D model. Two different variables of the 3D model can help us achieve this: vertex movement (displacement) and vertex colour. Vertex movement can be simulated by simply adding the calculated value of FDD-SV to the corresponding sensor location points. Vertex colour is obtained using a diverging colour scale with 6 steps, ranging from blue to red. The colour scale was obtained in ColorBrewer\textsuperscript{16}, which is an online tool to choose colour schemes. The result of the FDD-SV panel after those changes is displayed on Figure 4.27. Three.js also offers a plugin to move through the three-dimensional scene in a trackball manner, which was implemented at this point, since it is useful to view other sides of the dam, as in Figure 4.28.

At this point, the 3D Model panel could now be completed, which consisted on a combination of the

\textsuperscript{16}ColorBrewer: http://colorbrewer2.org/
Figure 4.26: Functional Prototype - Cabril’s Dam 3D model with different vertex colours.

Figure 4.27: Functional Prototype - FDD-SV panel with animated 3D Model.

Figure 4.28: Functional Prototype - FDD-SV panel with an upper view of the 3D Model.
same Response Spectrum graph used in FDD-SV panel and two equal 3D Models, using two different set of values for Responses in Displacement (one for the observed behaviour - also present in FDD-SV panel - and the other for the calculated behaviour through a prediction algorithm). In Figure 4.29 the new iteration of this panel is shown, as well as some contextual data of the observed dam behaviour, like the maximum water level registered and water weight against the structure. Figure 4.30 shows what is displayed when the user selects option "Difference Between Observed and Calculated Displacement". Now only one 3D model is displayed, instead of the two 3D models side-by-side presented with the "Observed and Calculated Displacement".

After testing the system with LNEC’s Concrete Dam Department, three issues emerged. The first was the fact that time selection should have a more direct method, so that the user would not need to hover through the whole water level graph to reach a specific point in time, when he already has one in mind. To solve this issue a date-picker was added to the top of the water level panel as an alternative to select a date, shown in Figure 4.31. The second issue was that the buttons, on the FDD-SV panel, to
control the number of lines displayed in the 2D Response Displacement would need too many clicks if the user only wanted to see a couple of sensors. For this reason, a Select All and Clear All option was added. The third issue relates to the fact that, sometimes, it can be useful to view the 3D displacement at different speeds or with a scalar multiplier, to amplify the observed displacement. This issue was solved by adding two sliders, bellow each 3D model across the whole system, granting the users the ability to modify these variables to need. In Figure 4.32 these last two changes are displayed, the Select All and Clear All buttons as well as the sliders for the 3D model.

4.2.3 System Potential

A final validation over the last functional prototype of DamVis with LNEC’s Concrete Dam Department got their approval on its current state. It was considered that the tasks they set out for the system to accomplish, described earlier in Section 4.1, were achieved.

DamVis is able to bring an updated version of previous monitoring systems, using state of the art technologies. At first glance, web technologies may not be the most indicated to use on a system that depends heavily on complex mathematical operations. For each file of sensor raw data (one hour time-frame) that the system processes there are 4,5 million values (25 sensors with 180000 samples per hour) that need to be read and processed into other derived measures. If performance is disregarded it can turn the system unusable, given the amount of calculations done. However, no performance loss was registered when compared to the previous monitoring system used by LNEC’s Concrete Dam Department, implemented with MATLAB. In some cases, with the persistent storage of data that was already
processed, DamVis is considerably quicker than this previous used system. In another perspective, user interaction and natural ways to explore data were elements highly lacking on the previous system. Using web technologies it is guaranteed to obtain better results, since web applications usually heavily rely on user interaction. The ability to access DamVis on the go, without the need to run the code locally like in the previous monitoring system, also weighed in the decision of developing it with web technologies.

DamVis improved some use case scenarios that had flaws in the previous monitoring system. One of those scenarios was when a user wanted to know what values the sensors registered given a specific water level. On the old system it was only possible to choose a date through a set of drop-down menus to select the corresponding year, month and hour. This is a direct manner to select a point in time if the user already knows which date he wants to see, however it does not enable the exploration of the visualized data in the water level graph. This means that a trial and error approach is needed when selecting the date from the drop-down in order to reach the desired water level. DamVis has two ways of selecting a point in time, either through clicking in the water level graph (aided by a tool-tip) or through a calendar, displayed in Figure 4.33. The date-picker is similar to what the old system had, since it enables the user to pick a specific date directly, but in a more structured way (the dates are displayed through a calendar instead of drop-down menus). Interacting with the Water Level to select a date is another way to explore the displayed data that complements the calendar. This way the user has the freedom to choose which method of selecting a date is more useful to him, taking into account the task he needs to accomplish.

Comparing the registered response displacement in a graph and 3D visualizations at the same time was another use case that was lacking in the previous system. These two visualizations were in two distinct panels, which removed the ability to compare them at the same time, despite being complementary to each other. In DamVis this is not an issue since the FDD-SV panel contains both visualizations, represented in Figure 4.34.
Figure 4.33: Selecting a point in time through interaction with the Water Level graph.

Figure 4.34: Visualizations for the registered response in displacement in 2D and 3D.
Another major issue present in the old monitoring system was the inability to explore the three-dimensional scene. The 3D model was represented in a static view, which largely reduces the advantages of a three-dimensional representation. DamVis grants the user ability to explore the three-dimensional scene through mouse interaction, like what is shown in Figure 4.35. This way the 3D model representation has the advantage of being able to show how different sides of the dam behaved with the registered concrete displacement.

In Appendix A the latest state of DamVis is represented through screenshots. The evaluation of DamVis is described in the next chapter.
Chapter 5

Evaluation

To get an objective evaluation of DamVis some tests were carried out. These tests can be divided into two different categories:

1. Usability Tests
2. Case Studies

The usability tests, described in Section 5.1, were used to evaluate usability and interactivity of the system through a structured set of tasks. Users that performed these tests were volunteers that do not have expert knowledge of dam monitoring. Results of the usability tests are objective and quantitative.

Case studies, described in Section 5.2, were used to evaluate system potentiality with end-users, in this case, LNEC’s Concrete Dam Department. These tests were done with a less strict structure than the Usability Tests because the test subjects have expert knowledge on the tasks that the system performs. They have the objective of gathering qualitative feedback from the test subjects about DamVis. Given that end-users are the ones who perform the case studies, it is possible to assess DamVis in real-world scenarios and use cases. Therefore, it is possible to attest the importance of case studies to validate DamVis.

5.1 Usability Tests

Usability tests were carried out to assess the quality of system usability and interactivity. Users that performed these tests were volunteers that do not have expert knowledge of dam monitoring. Twenty different users completed the usability tests, guaranteeing meaningful results, given the broader span of test subjects.

The protocol used for these tests is described in Section 5.1.1. Results are presented in Section 5.1.2. Discussion and conclusions about the results are presented in Section 5.1.3.
5.1.1 Protocol

Usability tests followed a protocol that was repeated for each user, to ensure objectivity in the results. The protocol used was divided into four stages:

1. Introduction to the system.
2. Exploration of the system by the user.
3. Task execution.
4. Satisfaction questionnaire.

The first stage, introduction to the system, consisted of explaining to the user what were the objectives for the test and a brief introduction to DamVis. At this stage, it was reinforced the notion that what was being tested was the system and not the user himself. A demonstration of DamVis was performed, giving context on what visualizations were used, why they are important and what tasks do they fulfill.

The second stage granted the user the opportunity to freely explore the system for five minutes. This worked as a consolidation mechanism of the introduction done before and ensured that the user was more comfortable to execute the next stage. The user was also encouraged to clarify anything that he did not fully understand.

The third stages consisted on asking the user to execute five tasks, while recording the amount of time and number of errors it took to complete each of them. The tasks simulated some of the tasks that end-users will be executing. The tasks that were asked to execute were:

1. Check at which month the water level reached its maximum value.
2. Verify what was the maximum FFT value registered by sensor 4, in March 30th.
3. Analyse, for January 12th, what was the maximum frequency registered for the response spectrum.
4. Also for January 12th, examine which sensor registered a bigger amplitude of response displacement, for sensor 9 and 12.
5. Visualize the difference between observed and the calculated displacement in the 3D Model.

These tasks should always be executed in the same order for each user, to prevent disparity between the registered results.

In the fourth and last stage, users filled a questionnaire, which had two main objectives: knowing the users (age, level of education, etc.) and measuring the system using a System Usability Scale\(^1\) (SUS). SUS is a “quick and dirty”, technology independent, usability scale that should be measured after executing the aforementioned tasks [7]. To produce this SUS scoring a questionnaire with a predefined set of questions should be filled. This questionnaire is composed by the following ten items:

1. I think that I would like to use this system frequently.

\(^1\)SUS: https://measuringu.com/sus/
2. I found the system unnecessarily complex.

3. I thought the system was easy to use.

4. I think that I would need the support of a technical person to be able to use this system.

5. I found the various functions in this system were well integrated.

6. I thought there was too much inconsistency in this system.

7. I would imagine that most people would learn to use this system very quickly.

8. I found the system very cumbersome to use.

9. I felt very confident using the system.

10. I needed to learn a lot of things before I could get going with this system.

Answers were given a scale from 1-5, where 1 means “Strongly Disagree” and 5 “Strongly Agree”. After a user gives his responses the SUS scoring is obtained by:

1. Transforming the 1-5 scale to a 0-4 scale
   
   (a) For odd-numbered items: Subtract one from the answer.
   
   (b) For even-numbered items: Subtract the answer from 5.

2. Sum the user responses with this new 0-4 scale (produces a value between 0 and 40).

3. Multiply the sum by 2.5 (turns the sum to a value between 0 and 100)

Once all users complete the questionnaire a median of the individual SUS scoring is calculated, revealing the final SUS score for DamVis.

### 5.1.2 Results

Before analysing the results some context about the enquired users is needed. In Figure 5.1, 5.2 and 5.3 it is displayed the test subjects gender, age and level of education respectively.

The results presented in this section can be divided into two categories: Results from task execution and SUS scoring. In Table 5.1 the amount of time (in seconds) and number of errors for each user when performing the five tasks, described in Section 5.1.1, are presented. An error is considered when a user is trying to find the answer for the corresponding task in an incorrect panel. Figure 5.4 and Figure 5.5 display the box-plots for task duration and errors, respectively. Table 5.2 presents the responses of each user in the SUS scoring format, also described in Section 5.1.1.
Table 5.1: Time and Errors for each task

<table>
<thead>
<tr>
<th>User</th>
<th>Task #1</th>
<th>Task #2</th>
<th>Task #3</th>
<th>Task #4</th>
<th>Task #5</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Time</td>
<td>Errors</td>
<td>Time</td>
<td>Errors</td>
<td>Time</td>
</tr>
<tr>
<td>1</td>
<td>20</td>
<td>0</td>
<td>35</td>
<td>0</td>
<td>45</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>0</td>
<td>63</td>
<td>1</td>
<td>70</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>0</td>
<td>55</td>
<td>0</td>
<td>53</td>
</tr>
<tr>
<td>4</td>
<td>8</td>
<td>0</td>
<td>56</td>
<td>1</td>
<td>55</td>
</tr>
<tr>
<td>5</td>
<td>10</td>
<td>0</td>
<td>24</td>
<td>1</td>
<td>49</td>
</tr>
<tr>
<td>6</td>
<td>4</td>
<td>0</td>
<td>50</td>
<td>1</td>
<td>60</td>
</tr>
<tr>
<td>7</td>
<td>4</td>
<td>0</td>
<td>43</td>
<td>1</td>
<td>58</td>
</tr>
<tr>
<td>8</td>
<td>7</td>
<td>0</td>
<td>40</td>
<td>1</td>
<td>76</td>
</tr>
<tr>
<td>9</td>
<td>6</td>
<td>0</td>
<td>56</td>
<td>1</td>
<td>42</td>
</tr>
<tr>
<td>10</td>
<td>6</td>
<td>0</td>
<td>30</td>
<td>0</td>
<td>37</td>
</tr>
<tr>
<td>11</td>
<td>11</td>
<td>0</td>
<td>33</td>
<td>0</td>
<td>55</td>
</tr>
<tr>
<td>12</td>
<td>6</td>
<td>0</td>
<td>48</td>
<td>0</td>
<td>40</td>
</tr>
<tr>
<td>13</td>
<td>5</td>
<td>0</td>
<td>38</td>
<td>0</td>
<td>59</td>
</tr>
<tr>
<td>14</td>
<td>9</td>
<td>0</td>
<td>29</td>
<td>0</td>
<td>64</td>
</tr>
<tr>
<td>15</td>
<td>12</td>
<td>0</td>
<td>41</td>
<td>0</td>
<td>50</td>
</tr>
<tr>
<td>16</td>
<td>7</td>
<td>0</td>
<td>35</td>
<td>0</td>
<td>39</td>
</tr>
<tr>
<td>17</td>
<td>8</td>
<td>0</td>
<td>49</td>
<td>1</td>
<td>48</td>
</tr>
<tr>
<td>18</td>
<td>4</td>
<td>0</td>
<td>54</td>
<td>0</td>
<td>58</td>
</tr>
<tr>
<td>19</td>
<td>9</td>
<td>0</td>
<td>30</td>
<td>1</td>
<td>45</td>
</tr>
<tr>
<td>20</td>
<td>5</td>
<td>0</td>
<td>42</td>
<td>0</td>
<td>61</td>
</tr>
<tr>
<td>Average</td>
<td>7,45</td>
<td>0</td>
<td>42,55</td>
<td>0,45</td>
<td>53,2</td>
</tr>
</tbody>
</table>

Table 5.2: SUS Questionnaire

<table>
<thead>
<tr>
<th>User</th>
<th>Question</th>
<th>Sum</th>
<th>Final Score (x2.5)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1 2 3 4 5 6 7 8 9 10</td>
<td>40</td>
<td>100</td>
</tr>
<tr>
<td>2</td>
<td>3 4 3 3 4 4 4 4 4 4 4 4 3 3 3 3 3 3 3</td>
<td>36</td>
<td>90</td>
</tr>
<tr>
<td>3</td>
<td>4 3 4 4 4 4 4 4 4 4 4 3 3 3 3 3 3 3 3</td>
<td>38</td>
<td>95</td>
</tr>
<tr>
<td>4</td>
<td>4 3 3 3 4 4 4 4 4 4 4 4 3 3 3 3 3 3 3</td>
<td>37</td>
<td>92.5</td>
</tr>
<tr>
<td>5</td>
<td>4 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3</td>
<td>33</td>
<td>82.5</td>
</tr>
<tr>
<td>6</td>
<td>4 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3</td>
<td>32</td>
<td>80</td>
</tr>
<tr>
<td>7</td>
<td>3 3 2 3 3 4 2 3 2 2 2 2 2 2 2 2 2 2 2 2 2</td>
<td>27</td>
<td>67.5</td>
</tr>
<tr>
<td>8</td>
<td>4 4 3 4 4 4 4 4 3 4 3 4 3 4 3 4 3 4 3 4 3 4 4 4 4</td>
<td>37</td>
<td>92.5</td>
</tr>
<tr>
<td>9</td>
<td>3 1 3 2 3 1 4 2 3 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2</td>
<td>24</td>
<td>60</td>
</tr>
<tr>
<td>10</td>
<td>3 4 4 3 4 4 4 3 4 3 4 3 3 3 3 3 3 3 3 3 3 3 3 3 3</td>
<td>35</td>
<td>87.5</td>
</tr>
<tr>
<td>11</td>
<td>3 4 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3</td>
<td>28</td>
<td>70</td>
</tr>
<tr>
<td>12</td>
<td>3 4 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3</td>
<td>32</td>
<td>80</td>
</tr>
<tr>
<td>13</td>
<td>4 3 3 3 3 4 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3</td>
<td>36</td>
<td>90</td>
</tr>
<tr>
<td>14</td>
<td>4 4 4 4 4 4 4 4 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3</td>
<td>36</td>
<td>90</td>
</tr>
<tr>
<td>15</td>
<td>4 4 4 4 4 4 4 4 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3</td>
<td>29</td>
<td>72.5</td>
</tr>
<tr>
<td>16</td>
<td>4 4 4 4 4 4 4 4 4 4 4 3 3 3 3 3 3 3 3 3 3 3 3 3 3</td>
<td>37</td>
<td>92.5</td>
</tr>
<tr>
<td>17</td>
<td>3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3</td>
<td>30</td>
<td>75</td>
</tr>
<tr>
<td>18</td>
<td>4 4 4 4 4 4 4 4 4 4 4 3 3 3 3 3 3 3 3 3 3 3 3 3 3</td>
<td>38</td>
<td>95</td>
</tr>
<tr>
<td>19</td>
<td>3 3 4 4 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3</td>
<td>32</td>
<td>80</td>
</tr>
<tr>
<td>20</td>
<td>4 4 4 4 4 4 4 4 4 4 4 3 3 3 3 3 3 3 3 3 3 3 3 3 3</td>
<td>38</td>
<td>95</td>
</tr>
<tr>
<td>Average</td>
<td>3,5</td>
<td>3,4</td>
<td>3,25</td>
</tr>
</tbody>
</table>
5.1.3 Discussion

It is possible to say, regarding the user context presented in the last Section, that most of the test subjects are males (70%), most of them are between the ages of 18 and 30 (85%) and also most of them have a Bachelor Degree (60%), as seen in Figure 5.1, 5.2 and 5.3 respectively.

Regarding the tasks executed by the test subjects, Task #1 was the easiest because users did not need to change panels and it was asked to analyse a pattern (month with the highest water level) instead of pin-pointing an exact date. For that reason it was expected to be the one to take the lowest amount of time and register less errors, as it did, with an average duration of 7.45 seconds and 0 errors. For Task #2 it was expected that the user selected a specific date and observed the FFT graph of a determined sensor. This task recorded the second highest average amount of time to accomplish and number of errors with 42.55 seconds and 0.45 errors. The amount of time to complete this task was already expected to be higher than the previous because the user needed to select a specific date and then wait for the back-end to respond with data. However, the task duration could be lower if more users used the Date-picker provided in the top-right corner of the Water Level panel, like what was shown in Figure 4.31. Most of them used the Water Level line graph to reach the desired date, which sometimes meant that the wrong date was selected. This could mean that the date-picker should be highlighted somehow, so that the users do not forget to used it when they have a specific date in mind. Task #3 was
Figure 5.3: User level of education.

Figure 5.4: Box-plots of each task duration, in seconds.

Figure 5.5: Box-plots of each task errors.
assumed to be the most difficult one, since it required the user to select yet another specific date and navigate through a different panel. The recorded values confirmed this assumption, since the average amount of time to complete this task was 53.2 seconds with 0.5 errors. This is justified by the fact that, once again, some users used a less direct way to pick a date but also because this task depends on the biggest amount of data to be sent from the back-end. The most common mistake in this task was to look at a graph that did not correspond to the Response Spectrum, which maybe due to the lack of expert knowledge in dam monitoring and mixing up some of the similar concepts and graphs used in DamVis. Task #4 made the users analyse a different graph on the same panel as before with the same date, which made it take less time to accomplish and with less errors (average duration of 18.2 seconds and 0.35 errors). This task could be almost immediate if the users utilized the “Clear All” button and afterwards clicking in Sensor 9 and 12. However, some users unselected each sensors by hand, which delayed the completion of the task. The most common error was to go back to the Accelerograms and FFTs panel instead of staying in the current FDD-SV panel, which may again be related to the lack of experience in dam monitoring or confusion between concepts. Lastly, Task #5 was the second lowest in duration and number of errors, with an average of 8.45 seconds and 0.2 errors. Users needed to change panels but, at this point, all data was ready so the application response was immediate.

Concluding, the observed results for task execution may point to a need to adjust the way of selecting a date, since it was one of the common mistakes. This could be by highlighting the different ways to select a date, like the date-picker, or through some other form of interaction besides mouse events on the Water Level line graph. The other errors observed, as said before, could be related to lack of experience in the dam monitoring field or simply by a lack of attention from the test subjects, since there are a lot of new concepts presented to them in order to perform this usability test.

When analysing the SUS score it is possible to see that the two lowest scoring answers are number nine (I felt very confident using the system) and ten (I needed to learn a lot of things before I could get going with this system.) which may indicate that the test subjects did not felt comfortable using the system due to the amount of concepts needed to understand the represented data. This supports the conclusion that lack of expertise in this field is a significant handicap in the usability tests. Systems with SUS scores between 68 and 80.3 are considered to be above average [7]. When a system scores 80.3 or more it is considered to have optimal usability and users are more likely to recommend it to others. This means that DamVis, considering a SUS scale, is a system with optimal usability, with a 83.125 score.

5.2 Case Studies

Usability tests are a good way to evaluate the quality of usability and interactivity of a system but they are not a complete set of tests. A system with good usability does not necessarily mean it is useful for its end-users. A great system is one that not only has good usability but one that also is able to give the right information at the right time to help users achieve their objectives. Case studies are useful to assess this because they are targeted at evaluating the context of the system. They are carried out
with end-users, which means that what is being evaluated is the content of the system and how good it helps them to complete their tasks. Since end-users are involved, it removes the amount of context and explanation that an inexpert user needs to start testing the system. Case studies are a much less strict way to test a system than Usability Tests.

The protocol used for these tests is described in Section 5.2.1. Results are presented in Section 5.2.2, they are much more subjective and qualitative than usability tests. In Section 5.2.3 discussion and conclusions about the results are presented.

5.2.1 Protocol

As stated previously, case studies are a very flexible testing methodology. Their purpose is to evaluate system usefulness in real-word scenarios by receiving qualitative feedback from test subjects. In this case, the users that are testing the system are end-users, unlike in the usability tests where volunteers without expert knowledge on dam monitoring were involved. This type of tests does not involve as many people as in usability tests, since there is no need to collect quantitative data. Case studies are done to evaluate usefulness and utility of a system. This means that the focus is less on usability or interaction and more on visualizations, their content, potential and helpfulness when a user performs real-world tasks.

Working with end-users in this testing scenario means that little explanation is needed to introduce the system. After a brief introduction the user has full control over the system, in order to explore it as he likes. This emulates real-word use cases, where the user does not have guidelines to execute a predetermined task, unlike in usability tasks. During system exploration, the user should give feedback about what he is trying to accomplish and how does he feel during that process. Differences while executing similar tasks on both systems (DamVis and the old monitoring tool) should be emphasized. Such differences could describe how the new system helps solve known issues of the old system or, if that is the case, hinders how certain tasks are now executed.

The qualitative feedback received from the users is then interpreted, revealing how well the system responds to real-world scenarios and how useful it can be as a new and improved dam monitoring system.

5.2.2 Results

The results presented in this section correspond to a single end-user that tested our system for the Case Study. More users should have participated in these tests, however this was not possible. Nonetheless, the user that performed the case study was the main point of contact from DamVis to LNEC, being present in all meetings and testing phases of the project.

The main concepts that resulted on feedback were:

1. Ability to view various Accelerograms and FFTs graphs at a time.

2. Side-by-side comparison of the displacement response in a 2D graph and 3D dam model.
3. Capability of exploring the three-dimensional scene of the 3D model.

4. Overall ease of use of the system when reading and exploring graphs through user interaction (zoom, pan, tool-tip).

The ability to view various accelerograms and FFTs graphs simultaneously was considered a helpful feature, since the old monitoring system only allowed to display three sensors simultaneously. With DamVis all 19 sensors can be selected and their corresponding data visualized, scaling the page vertically with the help of a scroll bar. When selecting a 4th sensor our user said "It is possible to select more than three sensors at a time? That is very useful. That was not available in the previous system, we had to always select three sensors, regardless of wanting only one or more.". This statement confirms the flexibility that DamVis provides for different variations of the same task.

Comparing side-by-side the displacement response values calculated with the FDD-SV algorithm in 2D graph and 3D dam model was also commented as a major improvement. In the old monitoring system these visualizations were in separated panels without any context from each other, not enabling to compare them effectively. The user stated that "Before we had these representations in two distinct panels, which hindered the ability to compare them. Now we can easily relate them, since they are side-by-side.". Once again, DamVis proves to be tailored to LNEC’s needs because of the thought and planning that went into the conception of its layout. The frequent testing stages and feedback received through the development process of DamVis also helped achieved this usefulness.

The capability to explore the three-dimensional scene of the 3D model was also lacking in the old system but included in DamVis. This spatial exploration of the three-dimensional scene is one of the greatest advantages when using 3D representations and so it was considered to be very useful. The user commented that "We had a static 3D representation of the dam, which did not allowed us to explore different perspectives. There was an older tool that had this ability to explore the 3D scene but was separate from the main monitoring system, which made it of little use.". This way the user feels that it is easier to complete tasks that were cumbersome before.

Overall user interaction, specially mouse events, is an aspect overlooked in the old monitoring system. Button clicks were the only supported user interaction. As said before, using web technologies to develop DamVis front-end gives a good starting point to improve in this aspect, since most web applications are strongly linked to user interaction in different ways. The ability to explore the visualizations with zoom, pan and tool-tips greatly improves the user interaction with the system in an intuitive way.

### 5.2.3 Discussion

With the positive feedback received from the case studies, we can conclude that DamVis is a useful system for dam monitoring. The results gathered from the case studies support the idea that DamVis provides a new and improved dam monitoring system. It fulfils all the objectives proposed while improving the effectiveness of the dam monitoring task. New patterns can now be discovered due these improvements, such as viewing multiple sensors accelerograms at once, side-by-side comparison of
the displacement response in 2D and 3D or the ability to explore data more in-depth with user interaction. End-users can now complete their tasks more efficiently or even in ways that were not possible before, which is the main goal when creating a new system, specially for such a sensitive task like dam monitoring.

This is corroborated by the received feedback, which is even more meaningful when considering that it was obtained through real-world use cases and from someone with expert knowledge in the matter. Therefore, with the case study, it is possible to confirm the usefulness of DamVis and its ability to provide an updated and meaningful dam monitoring system.
Chapter 6

Conclusions

DamVis had the goal of developing a web-based system for visualizing dam behaviour, using monitoring data that helped to understand and interpret its structural response, which was achieved. It provides a new and improved way of monitoring dam behaviour, fulfilling the needs of LNEC’s Concrete Dam Department. This claim is supported by the results of the system evaluation, both in usability tests and case studies. DamVis implemented a set of features that LNEC’s Concrete Dam Department deemed mandatory in order to effectively perform the dam monitoring, which are:

- Monitor the variation in the water-level.
- Choose a point in time to:
  - Visualize values recorded by the sensors.
  - Visualize a Fast Fourier Transform (FFT) of those values.
  - Visualize Frequency Domain Decomposition (FDD) for all sensors at a point in time.
  - 3D visualization of displacement response calculated by the FDD algorithm.

At the start of this project, the 3D dam model to visualize displacement response was considered as optional, since it could be helpful in monitoring dam behaviour but was not considered a priority. However, this functionality was implemented since it complemented the 2D visualization of the displacement response.

Developing this system with web technologies made it possible to use open-source software that helped developing the back-end and front-end. Some of the algorithms used to process data, for the back-end, were accessible through open-source repositories which helped to achieve all the desired features of the system. All visualizations in the front-end were also implemented with JavaScript open-source libraries (D3js and Threejs). They provide the developer with the ability to configure the respective visualization look and behaviour, given user interaction. Choosing which visualizations and user interaction to use in DamVis was a cooperative work with LNEC’s Concrete Dam Department applied with InfoVis techniques and principles that fitted their tasks and demands to properly monitor dam behaviour. They had the expert knowledge on dam monitoring and the corresponding data that enabled the completion of this project. Therefore it is possible to state that DamVis was tailored to their need.
6.1 Achievements

The greatest achievement of DamVis was to provide a new and improved tool for monitoring dam behaviour using unprecedented technologies in this area allied with InfoVis. No known dam monitoring system has been developed using web-technologies, meaning that these previous systems could only be accessed under specific conditions, where the application could be executed. The technologies that are usually chosen to develop this kind of system have the shortcoming of not being optimized to develop powerful visualizations, besides not providing end-users with the ability to assess dam behaviour on the go.

The technologies used in DamVis, allied with InfoVis principles, facilitated the solution of these common problems for dam monitoring systems. DamVis front-end is much more powerful than what is commonly found in similar systems because it was specifically tailored for dam monitoring while also considering the importance of data visualization and user interaction to explore data in a natural and meaningful way. This is confirmed by the results obtained in the system evaluation. Usability tests results and SUS scoring attest the quality of the usability and interaction of DamVis. The case studies corroborate the usefulness of the system from the end-users perspective, with context awareness and knowledge of the dam monitoring tasks.

Most of DamVis visualizations are not ground-breaking techniques due to LNEC’s Concrete Dam Department specific requests. They rather have line charts to easily identify exact values or patterns instead of other more visually appealing techniques that probably would not be so effective. However, user interaction was something lacking in their previous system, and that was overcome in DamVis. On almost every visualization the user has the ability to zoom and pan, complemented by a tool-tip to help read exact values. The most innovative visualization in DamVis was the 3D dam model. This type of visualizations is not commonly found in web applications. It complements very well the displacement response data in 2D, enabling end-users to draw conclusions that were not possible before. The ability to interact with the 3D model and exploring the three-dimensional scene is also a very important feature that was lacking in LNEC’s previous system.

One of the main concerns before starting to implement DamVis was that its back-end could not be able to process the amount of data that dam monitoring implies in useful time. This was a non-issue since the back-end handles all data processing in roughly the same amount of time that the LNEC’s previous system. DamVis stores persistently the processed data (this was not done in the previous system), which results on a more responsive application, since it does not need to repeatedly do the same calculations.

6.2 Future Work

DamVis can continue to grow and improve in the future to become an even better solution for dam monitoring. These improvements can be either for the back-end or the front-end.

Despite the back-end response time being acceptable it could still be improved. Some other ap-
proaches to share data between the back-end and front-end could improve the system performance, since one of the bottlenecks is the amount of data that needs to flow from one side to another. The back-end already sends only the necessary data for a given request but it is still a considerable extent of data, which increases the amount of time that the front-end is waiting to receive it. Some form of hash function that reduces the size of the data to be transferred can help reduce this bottleneck. Using this approach requires the front-end to reverse the hash in order to be able to use the data. This adds another step to be done before building the visualizations, increasing the front-end complexity, but could produce better response times between it and the back-end.

Another major improvement could be the upgrade to a real-time monitoring system. At this point, DamVis consumes values of the files present in a specific location. This approach relies on someone to populate that location with new data in order to be able to visualize it. If a more automated mechanism was implemented DamVis could be a real-time dam monitoring system. A mechanism that transfers the recorded hourly values of the dam sensors automatically can help achieve this. Once these files coming from the dam reach DamVis, they could be processed and stored. This way they would be ready to be sent as soon as a request for that new date reached the back-end, instead of only being processed after the first time that specific date is requested.

The front-end can also be improved by adding some new features. One useful feature that could be added is the ability to save favourite visualizations configurations, similar to a bookmark system. Storing these favourites configurations can help the end-users comeback to a frequently visited date, with a specific set of sensors or view of the 3D model. Continuous work with LNEC’s Concrete Dam Department could also result in the addition of other visualization techniques that could be useful for monitoring dam behaviour, which would help DamVis to be even more complete.
Bibliography


Appendix A

DamVis Screenshots

This appendix shows screenshots of the final state of DamVis, at time of writing this document. The landing page is shown in Figure A.1. It possible to see the system's dashboard layout, composed by a title, sidebar (for menu navigation) and content. The content in this landing page displays the Water Level visualization and the disabled Sensor Location Schema. When a point in time is not selected, the Sensor Location Schema does not support user interaction and has reduced opacity. Selecting a point in time can be done in two distinct ways. It can be done either by clicking in the Water Level visualization, where a tool-tip is displayed to show which date is being hovered, represented in Figure A.2 or by selecting a point in time using the date-picker, which is used to directly input a specific date, like in Figure A.3. Besides the tool-tip, Water Level visualization also supports user interaction through zoom and pan.

![Figure A.1: Landing page.](image)

After choosing a point in time, the Accelerograms panel is automatically selected and the Sensor Location Schema is enabled, shown in Figure A.4. While the Accelerograms and FFTs data is being retrieved, a loading button is displayed, to provide feedback to the user. It is also possible to see that the Water Level visualization now has an orange line to mark the selected time.

When the sensor data is ready, it is possible to view the Accelerograms and FFTs or change which sensor's data is being displayed, through interaction with the Sensor Schema Location, like in Figure A.5. The page scales vertically, with a scroll-bar, to adjust to the number of sensors being displayed at
Once. If more than one sensor is selected, two more line charts are added, one for the accelerograms data and the other for the FFT values. Each of them display the maximum, median and minimum value for every selected sensor, per time sample. Each of the visualizations in this panel also support user interaction through zoom and pan.

The FDD-SV panel represents the values calculated by the Frequency Domain Decomposition with Singular Value Decomposition. It contains three visualizations, Response Spectrum and Response in Displacement in 2D and 3D, represented in Figure A.6. Zoom and pan is also available to interact with the Response Spectrum and Response in Displacement charts. The 2D Response in Displacement can also hide the lines that are being displayed, each one representing the corresponding sensor values. This is done through interaction with the group of buttons next to it, shown in Figure A.7. The 3D Response in Displacement supports exploration of the three-dimensional scene, to view different sides of the dam, also displayed in Figure A.7. The sliders bellow the 3D visualization are also used to interact with it. The first one, called Displacement Multiplier, is used to control a scalar multiplier applied to the displacement values. The second one, called animation speed, controls the speed at which the frames are updated to show the animation of displacement values.

The last panel is the 3D Finite Element Model. It contains the same Response Spectrum visualization present in FDD-SV, to give some context to the two 3D dam models displayed bellow it, represented in Figure A.8. Some more relevant context is given by displaying the maximum water level registered for all
data and the water weight against the dam’s structure. The Observed 3D displacement model represents the calculated values, which uses the same data as the 3D model displayed in the FDD-SV panel. The Calculated 3D displacement model shows the predicted values for the dam behaviour, obtained through the statistical prediction model. This way it is possible to compare observed values in the real-world with the prediction model. Both 3D models also support the same interaction as the 3D model from the FDD-SV panel, shown in Figure A.9. There is also the ability to display the difference between the displacement values of these two 3D models by choosing the “Difference Between Observed and Calculated Displacement” option, displayed in Figure A.10.
Figure A.5: Accelerograms and FFTs panel.
Figure A.6: FDD-SV panel.

Figure A.7: FDD-SV panel interaction.
Figure A.8: 3D Model panel.

Figure A.9: 3D Model panel interaction.
Figure A.10: 3D Model with Difference between Observed and Calculated Displacement.