



Final Dissertation



BACHELOR's DEGREE IN COMPUTER SCIENCE

A WEB GRAPHICAL USER INTERFACE FOR THE PACKET-LOSS-CONCEALMENT TESTBENCH TOOL

Student

Stefano Dallona

Supervisor

Prof. Luca Turchet

Co-Supervisor

Ing. Luca Vignati

Good morning / **afternoon to everyone**, first of all I would like to **thank all the committee members** for attending the discussion of my thesis work.

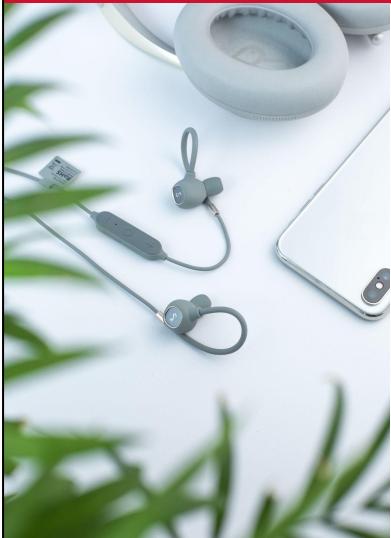
My name is Stefano Dallona, I'm a **student of the Bachelor's Degree in Computer Science** and for the entire duration of my studies I've worked full-time.

The **title** of my thesis is: "**A Web Graphical User Interface for the Packet-Loss-Concealment Testbench Tool**". My work was **supervised by professor Turchet and by Luca Vignati**.

What really attracted me about this thesis was that, being complementary to a Ph.D research carried out by Luca Vignati, it had an **immediate applicability** and the **topic** was totally **new** to me **and** therefore very **challenging**.



Introduction



Context

PLC Testbench, a tool for the comparison of Packet Loss Concealing algorithms



Problems

Interaction only possible at code level in an IDE
Output information not integrated



Solution

Development of a graphical user interface

The **context** in which the idea of this thesis was born is the **PLC Testbench**, which is a tool to compare the effectiveness of different Packet-Loss-Concealment algorithms. **PLC algorithms** try to **make packet loss** in audio streams **unperceivable** by reconstructing the lost portions of the signal to **provide acceptable quality even on lossy connections**.

The **biggest problems** with the PLC Testbench were that:

- the **only way to interact with the tool** was **by directly editing** and executing the **code** in an IDE;
- the **analysis** of the outputs was a bit **cumbersome and inefficient** because the output **information was not integrated**.

The development of a **Graphical User Interface** was seen as a **solution** to these problems.

Purpose of the Thesis

Develop a **Graphical User Interface** for the **PLC Testbench** tool
in order to make

QUALITATIVE and **QUANTITATIVE**
comparison of PLC algorithms

EASY, EFFECTIVE and **EFFICIENT**



The **purpose** of this thesis work was therefore **developing a GUI** for the PLC Testbench to support the qualitative and quantitative comparison of PLC algorithms by **making the interaction** with the tool **easy, effective**, and **efficient**.

Qualitative comparison mainly consists in visually **comparing waveforms and spectrograms**, and in **listening to** portions of the original and reconstructed **audio files**.

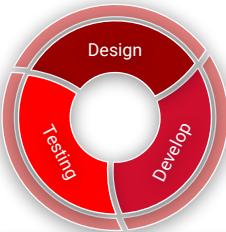
Quantitative comparison is essentially **based on** the figures from the **metrics** calculated by the program on each audio file.



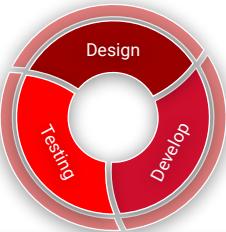
Methodology

Iterative agile development process
to
- shorten release cycles / fast feedback
- minimize risk of useless development

1

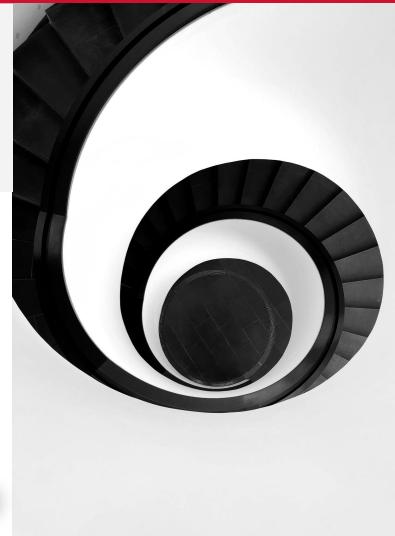
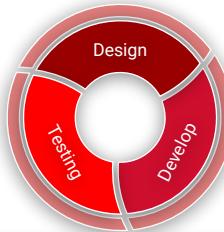


2



...

N

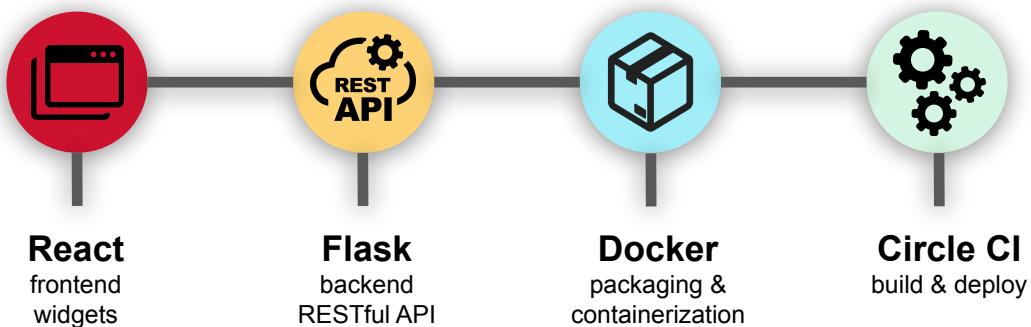


From a methodological perspective, it was decided to adopt an **iterative agile development process based on Kanban** because of its **simplicity** and **flexibility**. The process essentially consisted of **short cycles of design, develop and testing phases**.

The **limited duration of the iterations** allowed keeping the focus on **small and clear objectives** and getting **fast feedback**, thus **minimizing** the risk of wasting time developing **useless functions**.



Technologies



From a technological perspective it was decided to **implement** the PLC Testbench's GUI through a **Web application**, because of the **advantages** offered in terms of **ease of distribution, scalability and accessibility**.

From an architectural point of view the web application was split into **2 layers**: a **frontend** developed in JavaScript and based on React framework, and a **backend**, developed in Python with the help of the Flask framework.

Bundling and distribution were addressed by **containerizing** the application **as a Docker image**, while for **build and deploy** I leveraged the **pipelines as code** solution offered by Circle CI's cloud platform.



GUI - Landing Page

PLC Testbench is ...

PLCTestbench

PLCTestbench is a companion tool for researchers and developers working on Packet Loss Concealment (PLC). It greatly simplifies the process of measuring the reconstruction quality of PLC algorithms and allows to easily test the effects of different packet loss models and distributions.

It features the implementation of some of the most common packet loss models, PLC algorithms and metrics:

Packet Loss Simulation

- **Binomial**: uniform distribution of packet losses, governed by the Packet Error Ratio (PER) parameter.
- **Gilbert-Elliott**: bursty distribution of packet losses, governed by the four probabilities associated to its two states (For each state, the probability of packet loss and the probability of transitioning to the other state) [1].

PLC Algorithms

What would you like to do ?

Configure new Run

Configure a new elaboration, choosing input files, setting algorithms and parameters

Analyze existing Run

Analyze an existing elaboration, navigate waveforms, play files, compare metrics and spectrograms

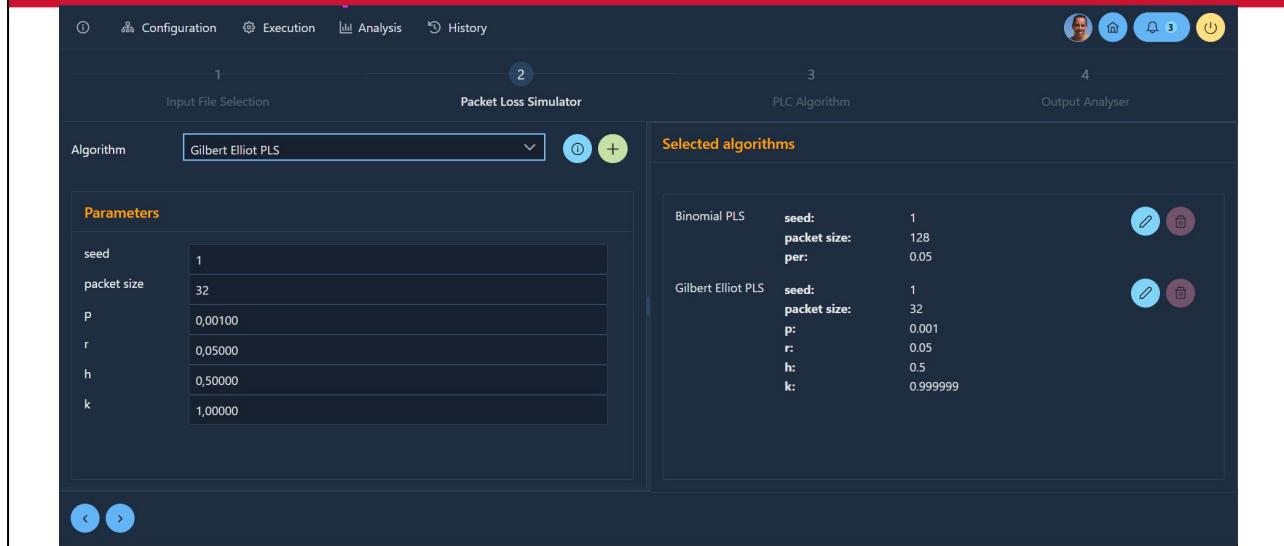
Edit Run

Modify an existing elaboration by changing worker settings or input files

Now time has come to have a look at the application interface.

This is the **landing page** which addresses **two main purposes**. The first one is helping the user to **develop a proper mental model of the software**. To obtain this, a short **help page** about PLC-related concepts and what the application can do is displayed on the left. The second objective of this page is putting the user in condition to **start using the testbench immediately**. To achieve this, a list of the **most relevant operations** supported by the application is displayed **on the right**, with a **direct entry point** to the functions.

GUI - Configuration Module



The screenshot shows a user interface divided into four main sections:

- 1. Input File Selection:** A dropdown menu set to "Gilbert Elliot PLS".
- 2. Packet Loss Simulator:** Displays parameters for the selected algorithm:

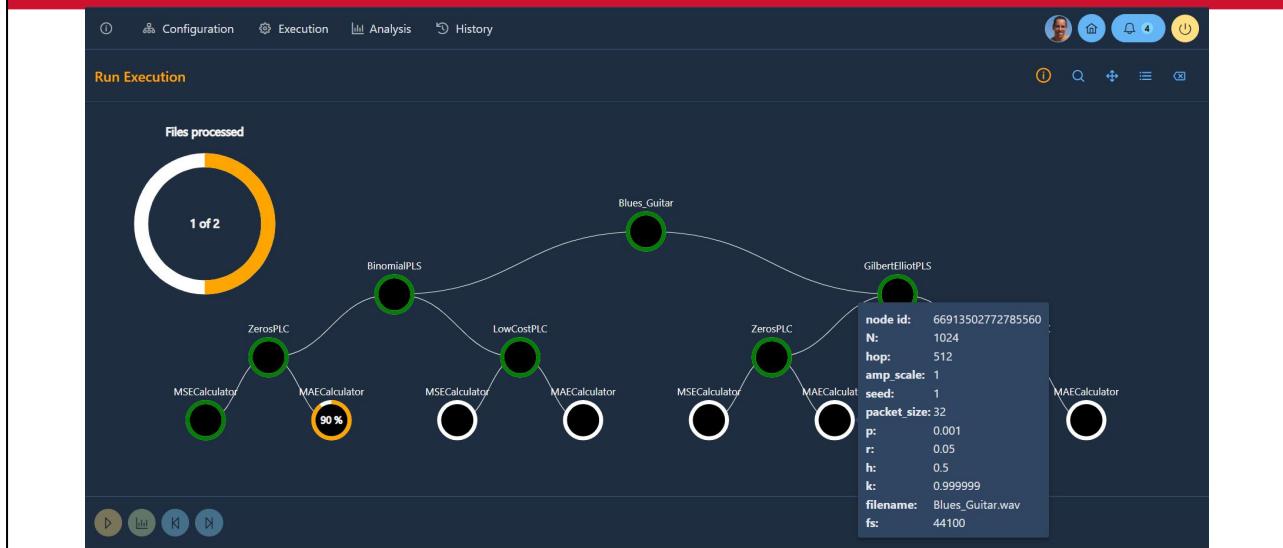
Parameter	Value
seed	1
packet size	32
p	0,00100
r	0,05000
h	0,50000
k	1,00000
- 3. PLC Algorithm:** Shows the "Selected algorithms" table:

Algorithm	seed:	packet size:	per:
Binomial PLS	1	128	0,05
Gilbert Elliot PLS	1	32	0,001
- 4. Output Analyser:** Displays the same algorithm configuration table as section 3.

This screen instead is intended to **guide the user** through the collection of the input data to **configure an elaboration**. Design in this case was focused on three targets:

- **fast and effective interaction:** pursued by providing **sensible defaults for the settings and keyboard shortcuts**;
- **clarity:** pursued by splitting the configuration process into multiple **sequential steps**, each focused on a single aspect;
- **adaptability:** achieved by implementing **dynamic discovery** of the available **algorithms and their settings**, thus supporting testbench extensions without requiring any change to the code.

GUI - Execution Module



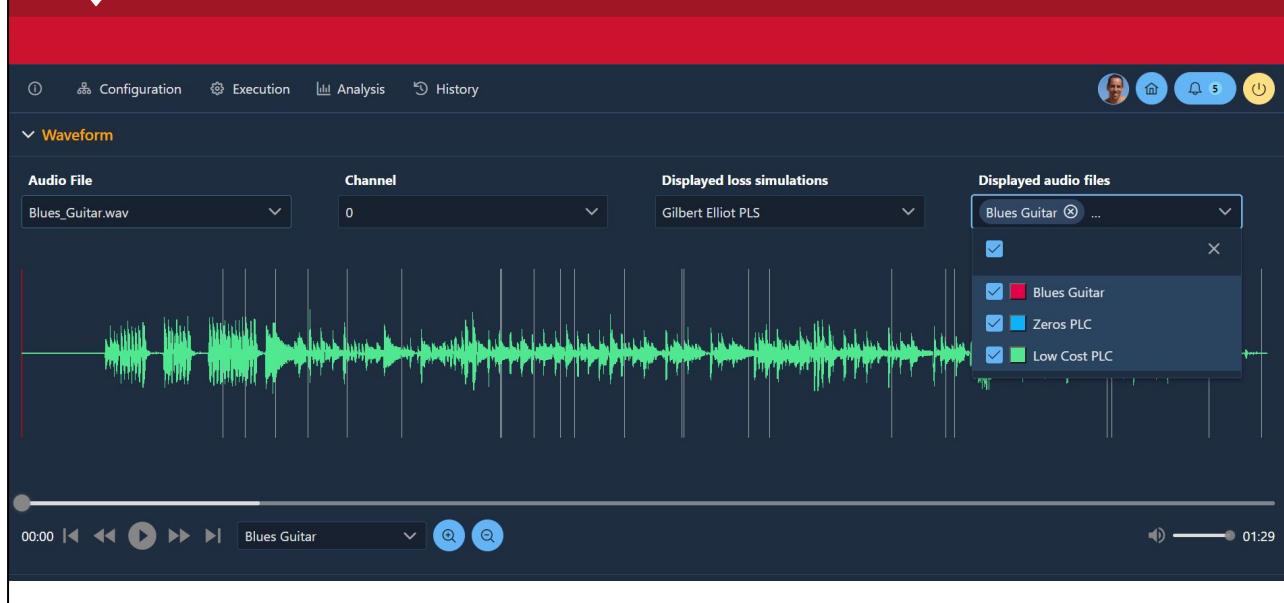
A PLC Testbench **elaboration** can be represented as a **forest of trees**, where each tree corresponds to the processing of a single input file.

Since an elaboration can in some cases take a considerable amount of time, having the **possibility to monitor progress** was **essential**.

Being the **structure** of the elaboration **identical for all the input files** it was decided to display progress at two different levels of detail: **overall progress** and **progress within a single file** processing. The **settings** of each elaboration step can be inspected by placing the mouse pointer over the corresponding **node**.



GUI - Analysis Module - 1



In the analysis module the **waveforms of all the versions of a given input file are displayed on the same chart, together with the lost packets' regions**.

Each **waveform** can be **shown or hidden** by activating or deactivating the corresponding checkbox in the selected audio files dropdown.

Each **audio file** can be **played** by the playback controls and bar below the waveforms chart, **to evaluate the perceived quality of the reconstruction**.

By dragging the mouse pointer over the chart it is **possible to zoom-in** any portion of the file. The zoom operation can be repeated **until the desired level of detail is reached**.



GUI - Analysis Module - 2



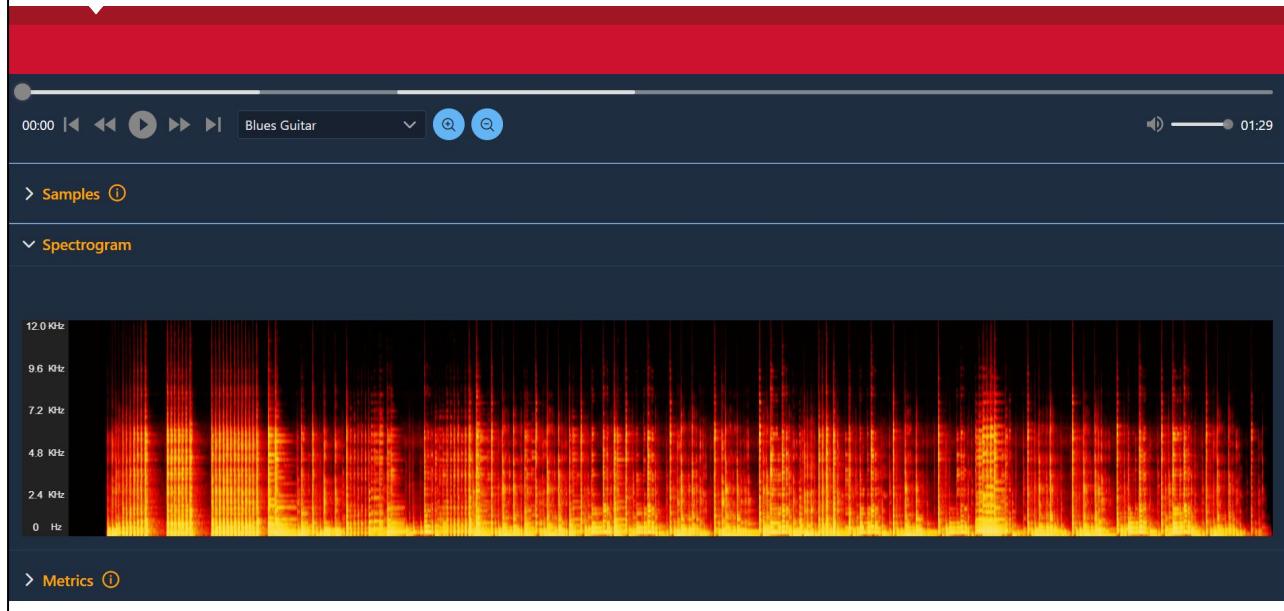
When clicking on a lost packets region in the “Waveform” view, the **detailed waveforms** of all the audio signal’s reconstructions are displayed in the “Samples” view for a **slightly wider area surrounding the lost packets**. Each waveform can be shown or hidden by **clicking on** the corresponding item in the **legend**.

Zoom-in and zoom-out are supported by dragging the handles in the bar below the chart. At maximum zoom level single audio signal’s samples can be **discriminated**.

The more the original and the reconstructed **signals** are **overlappable**, the more the **reconstruction** can be considered **accurate**.

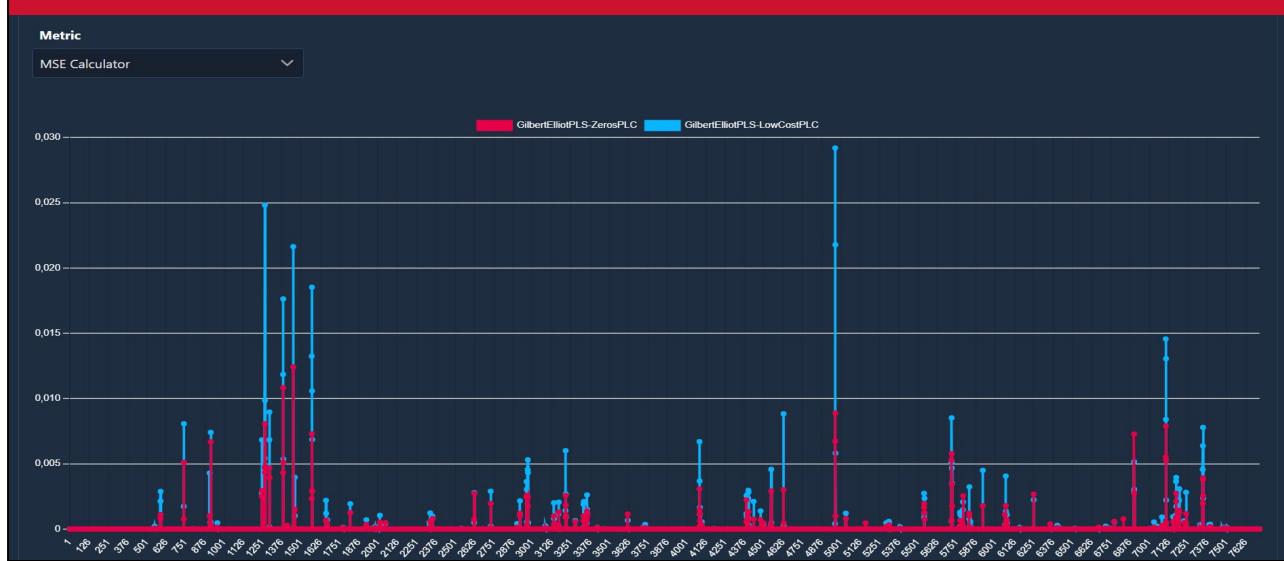


GUI - Analysis Module - 3



For each audio file a **spectrograms** can be displayed, representing the **composition of the audio signal over time as a color map**, where **brighter colors** correspond to greater intensity of the corresponding frequencies.

GUI - Analysis Module - 4



Output metrics are **calculated on each original and reconstructed audio file**. They can be grouped into **two categories**: **linear metrics**, producing a **time series for each audio file**, and **scalar metrics**, producing a **single value for each audio file**. The data to be displayed can be customized by clicking on the legend items.

GUI - Search Module

Search

NOT AND OR

Run ID Starts with 2

Status Any in COMPLETED, FAILED

Select a saved filter

🔍 📁 ⏪ ⏪

Run ID	Files	Created On	Creator	Status	Actions
22025239277524341	• Blues_Bass.wav	2023-10-27T00:58:57	stefano.dallona@gmail.com	✓	🔗
26865775114978227	• Blues_Guitar.wav	2023-10-25T17:43:08	stefano.dallona@gmail.com	✓	🔗

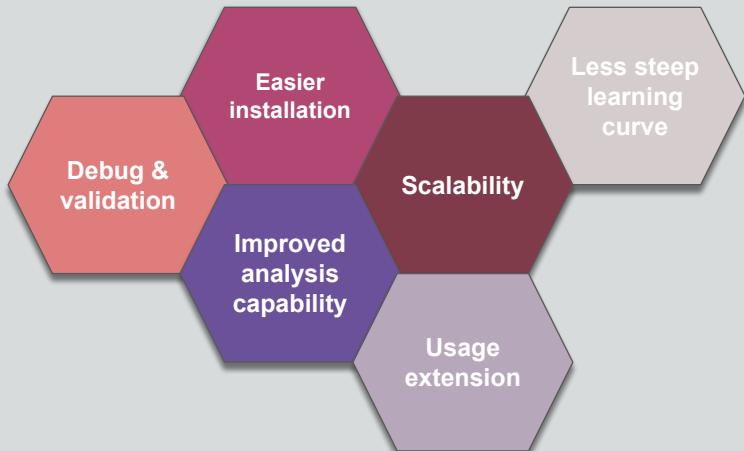
5 < 1 > 1 - 2 of 2

The search module allows the user to **navigate through the list of the elaborations** or to **build, save and re-execute specific queries**. Queries can be built visually by **combining multiple conditions using logical operators**. Each **condition** can be **based on any field** of the elaboration **or** on any **setting** of the referenced algorithms.

Results



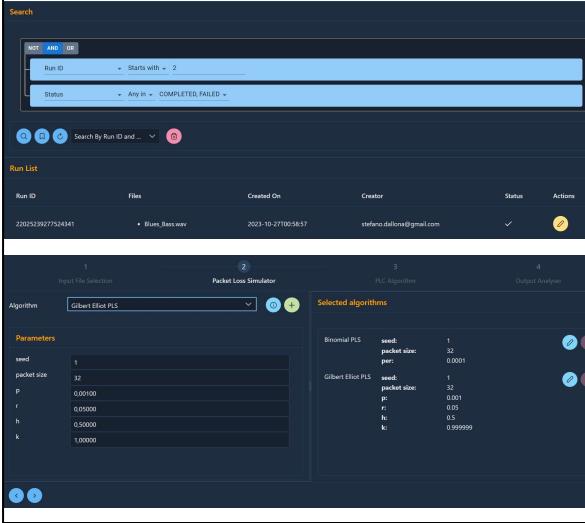
Benefits brought to the PLC Testbench tool



The most relevant benefits brought to the PLC Testbench tool by this thesis work can be summarized as follows:

- a thorough **debug and validation** of the testbench was carried out during of the project;
- the **analysis capabilities** of the tool have been extended and **improved** in terms of **quantity and quality** of the information;
- tool **installation** has been made **easier** by hiding the **complexity** in the **build process of the docker image**;
- the **application** has been made **more scalable** by making **distributed deployment** possible;
- the tool **can now be used** also by **users with no expertise in Python programming**.

GUI vs Native Interaction - 1



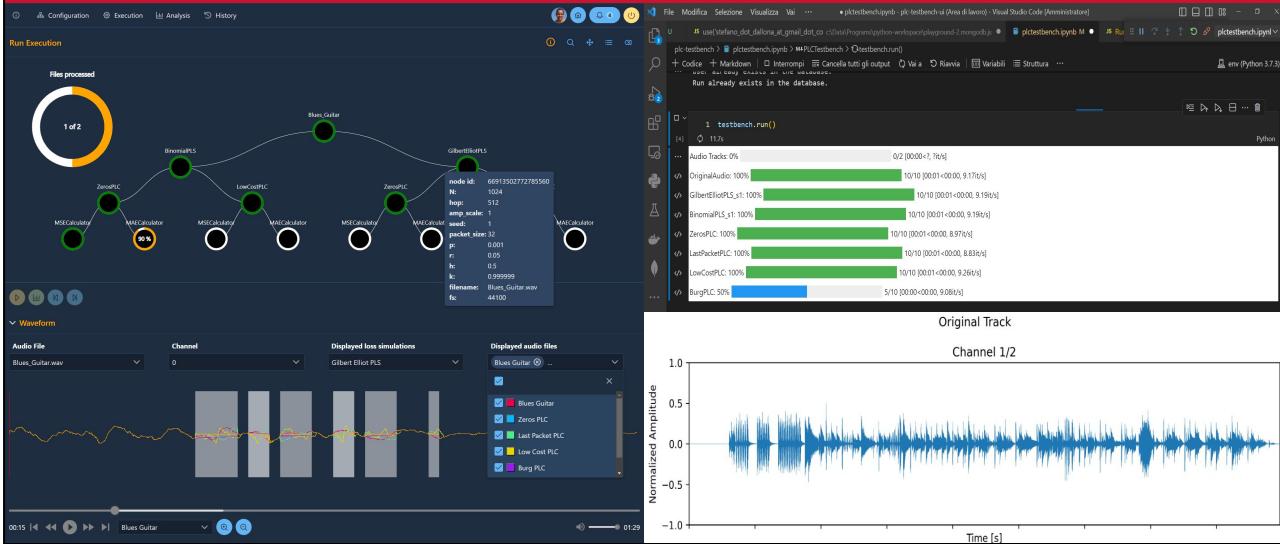
```
use('stefano_dot_dallona_at_gmail_dot_com')
db['RunView'].find({ "run_id": "13292225527426488", "status": { "$in": [ "COMPLETED", "FAILED" ] } })
```

```
1 [
2   {
3     "_id": "13292225527426488",
4     "status": "COMPLETED",
5     "description": "",
6     "classname": "Run",
7     "run_id": "13292225527426488",
8     "creator": "stefano.dallona@gmail.com",
9     "created_on": "2023-10-27T01:00:12",
10    "selected_input_files": [
11      "Blues_Bass.wav",
12      "Blues_Guitar.wav"
13    ],
14    "lost_samples_masks": []
15  }
16]
```

This slide shows a visual **comparison** of how some **operations** are handled **in the web GUI versus** how they are carried out through the **native testbench interface**. On the left we can see how the **GUI** supports **visually** building a queries to find specific elaborations or configuring new ones. On the right, in the **native interaction** the same operations have to be performed **at programming level**.



GUI vs Native Interaction - 2



Here the **same comparison** is displayed for **progress monitoring and result analysis use cases**.

Again the web **GUI** provides a more convenient and effective interaction by presenting **output information in a more detailed and integrated way**. For example only in the **GUI** the **hierarchical structure of the elaboration** can be recognized and the **lost packet regions** are **overlapped to waveforms**.



Future Work

A few ideas

- *Extensive validation with users (*)*
- *Support for additional audio formats*
- *Exposing logs*
- *Search functions on output results*
- *UI Themes*
- *Support for additional languages*
- *User collaboration use cases*
- *Additional authentication & authorization methods*
- *Data export*

(*) A first preliminary but very positive feedback has been received from experts at the IS2 2023 Conference in Pisa



Despite being fully operational and exposing all the current functions of the underlying **PLC Testbench**, the web GUI still has a lot of room for improvement. In this slide I tried to provide some indications on possible future enhancement areas, like:

- Extensive validation with users;
- Support for additional audio formats;
- Exposing logs;
- Search functions on output results;
- Management of UI Themes;
- Support for additional languages;
- User collaboration use cases;
- Additional authentication & authorization methods;
- Data export.



Conclusions

PLC Testbench Native interaction

- limited and raw visual output
- no output integration
- programming-level interaction
- installation complexity

Web GUI

- sophisticated visual output
- highly integrated output
- graphical interaction
- installation ease



Finally this slide summarizes the major achievements of this thesis work by highlighting the most critical limitations of the native PLC Testbench's interface and how they were addressed and overcome in the web GUI.

For example:

- **limited and raw output** has been **replaced by a sophisticated dashboard** where information is highly integrated;
- **from the programming level the interaction** has been turned **into graphical**;
- **installation complexity** has been completely **hidden**.



References



GitHub repository of the project

<https://github.com/cimil/plc-testbench>

public access



Docker image of the project

<https://hub.docker.com/r/cimil/plc-testbench-ui>

public access



Paper presented at the IS2 2023 in Pisa

4th International Symposium on the Internet of Sounds

https://drive.google.com/file/d/1kFY9Hj-G7mWJP7U3CpPZf9TR95M5tQGV/view?usp=drive_link

restricted access



The **results** of this thesis work are **publicly available** in the form of **source code** at the **GitHub** URL mentioned in the slide or as a **pre-built docker image** at the indicated DockerHub URL.

A **paper** about the jointed work of Luca Vignati and me for the respective Thesis was **presented at the 4th International Symposium on the Internet of Sounds**, held in Pisa on the 26-27th of October 2023 and will soon be available in the proceedings of the conference.



UNIVERSITÀ
DI TRENTO

Prizes



BEST DEMO AWARD

at the 4th International Symposium on the Internet of Sounds
October 26-27, 2023, Pisa, Italy

*PLC Testbench: a modular tool for the study and comparison
of audio Packet Loss Concealment algorithms*

by Luca Vignati and Stefano Dallona



Luca Turchet
Luca Turchet
IS² General Chair

During the conference in Pisa the software was **presented to a team of experts** and got a **very positive feedback**.

At the end of the conference Vignati and me had the great honor to be prized with the **“Best Demo Award”**.

Thank you so much for your attention.

Now I am at your disposal in case there are any **questions**.

Acknowledgements



Thanks to

**Prof. Luca Turchet
Ing. Luca Vignati**

**for the invaluable feedback
and support provided**