



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

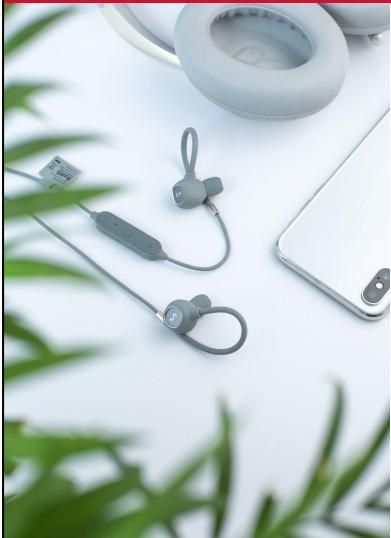
My name is Stefano Dallona, I'm a **student of the Bachelor's Degree in Computer Science** and for the entire duration of my studies I've worked full-time.

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 the **immediate applicability** and the fact that the **topic** was completely **new** to me.

# Introduction



## Context

PLC Testbench, a tool to compare 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 this thesis was born is the **PLC Testbench**, a tool to compare Packet-Loss-Concealment algorithms developed by Luca Vignati.

**PLC algorithms** try reconstruct the lost portions of audio streams to **provide acceptable quality on lossy connections**.

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

- the **interaction with the tool** was **possible only at code level**;
- the output **information** was **not integrated**, making the **analysis** inefficient

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

# Purpose of the Thesis

Developing a **GUI** for the **PLC Testbench** tool  
to boost up

**QUALITATIVE** and **QUANTITATIVE**  
comparison of PLC algorithms

by making the **INTERACTION** with the tool  
**EASY, EFFECTIVE** and **EFFICIENT**



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

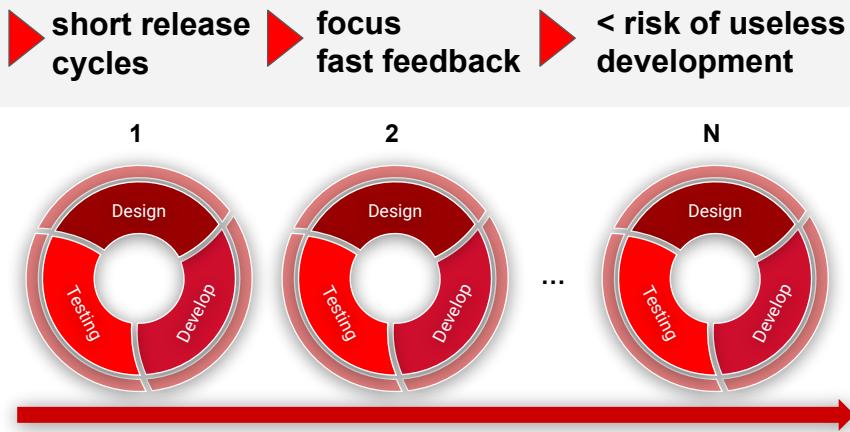
**Qualitative comparison** mainly consists in visually **comparing waveforms and spectrograms**, and in **listening to audio files**.

**Quantitative comparison** is essentially **based on metrics** calculated the reconstructed audio files.



# Methodology

## Iterative agile development process



From a methodological perspective, it was decided to adopt an **iterative agile development process** composed of **short cycles of design, develop and testing phases**.

This helped keeping the focus on **small and clear objectives** and getting **fast feedback**, thus **minimizing** the risk of developing **useless functions**.



## Technologies



**Front-end**  
Javascript + React



**Back-end**  
Python + Flask



**Packaging &  
Distribution**  
Docker



**Build & Deploy**  
Circle CI pipelines

The GUI was developed through a **Web application**, made of **two layers**: a **frontend, implementing the graphical part of the UI**, written in JavaScript and based on React framework and a **backend, implementing services over a REST API**, written in Python and based on Flask framework.

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



# 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**

Configuration    Execution    Analysis    History

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**:

- helping the user to **develop a proper mental model of the software** by providing on the left a **short help page**;
- putting the user in condition to **start using the tool immediately** by displaying on the right a **list of the most relevant operations** supported by the application, with a **direct entry point** to the functions.



# GUI - Configuration Module

The screenshot shows a user interface for configuration, 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 algorithm with its parameters:

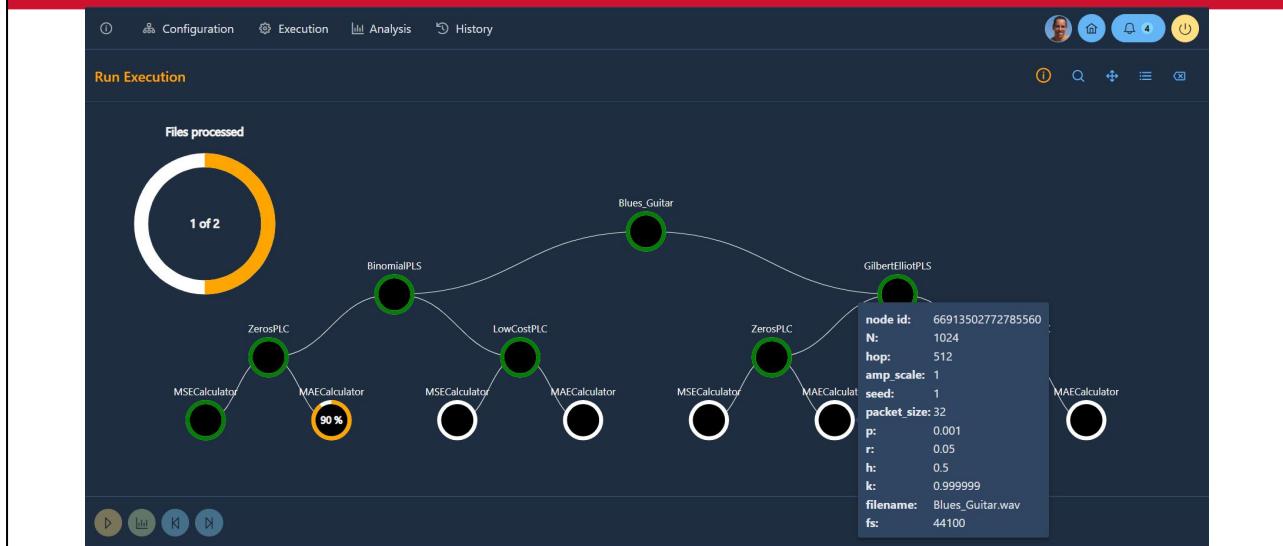
Algorithm	seed:	1
	packet size:	128
	per:	0,05
- 4. Output Analyser:** Shows the selected algorithm with its parameters:

Algorithm	seed:	1
	packet size:	32
	p:	0,001
	r:	0,05
	h:	0,5
	k:	0,999999

This screen instead is intended to **guide the user** through the **configuration of an elaboration** focusing on three objectives:

- **fast and effective interaction:** pursued with **sensible defaults for the settings and keyboard shortcuts**;
- **clarity:** pursued by splitting the configuration process into **multiple sequential steps**;
- **adaptability:** achieved by implementing **dynamic discovery of the algorithms and settings**, thus supporting testbench extensions with no changes to the code.

# GUI - Execution Module

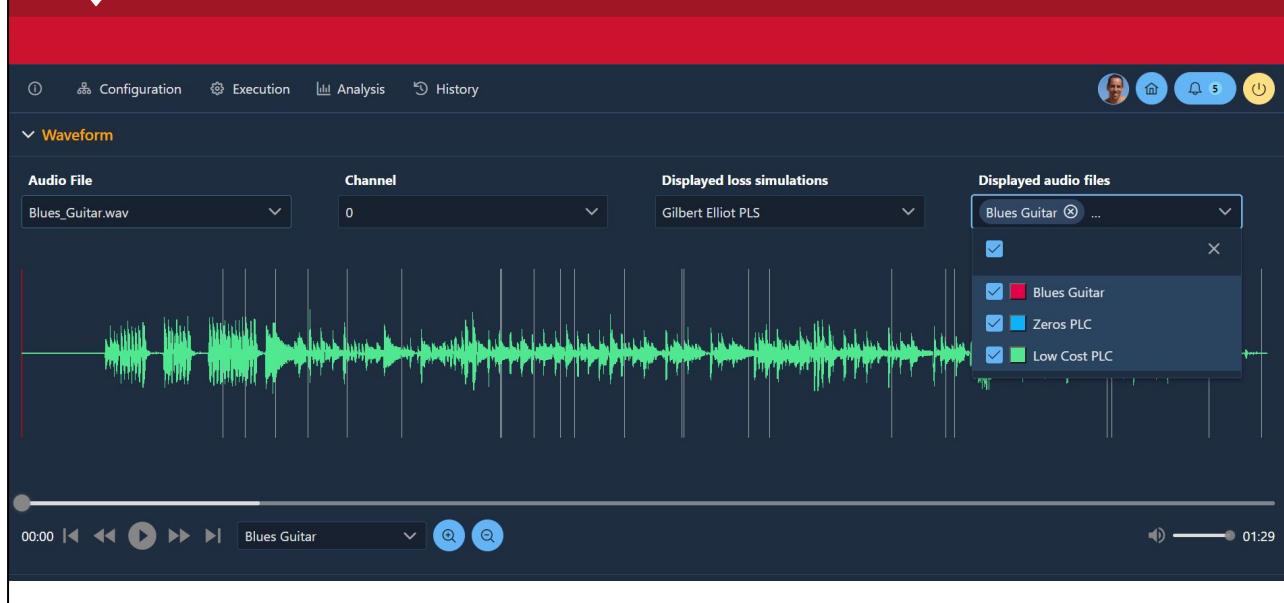


A PLC Testbench **elaboration** can be represented as a **forest of trees**, each corresponding to a single input file. Processing can take a long time, thus **progress monitoring** is essential.

Since the **structure 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 independently**.

Each version of the **audio file** can be **played to evaluate the perceived quality of the reconstruction**.

Zoom-in function supports any **level of detail**.



## GUI - Analysis Module - 2

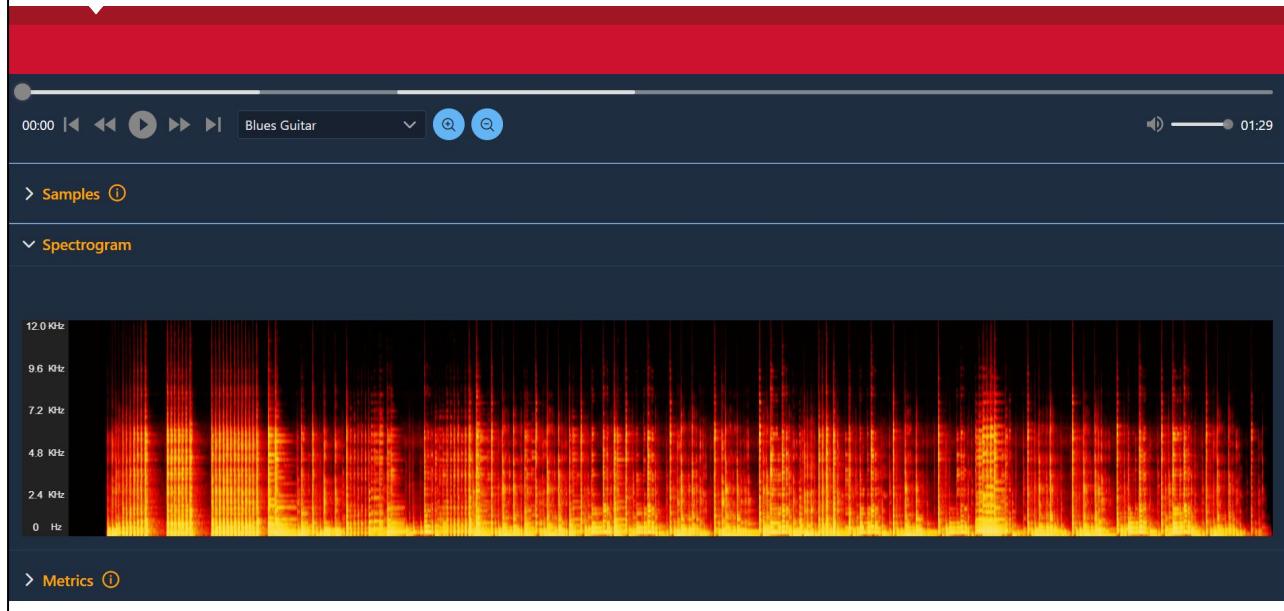


In the “Samples” view in the analysis module at the maximum zoom level **single audio signal’s samples** can be discriminated. **Zoom-in and zoom-out** are supported by dragging the handles in the bar below the chart. Each **waveform can be shown or hidden** independently.

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

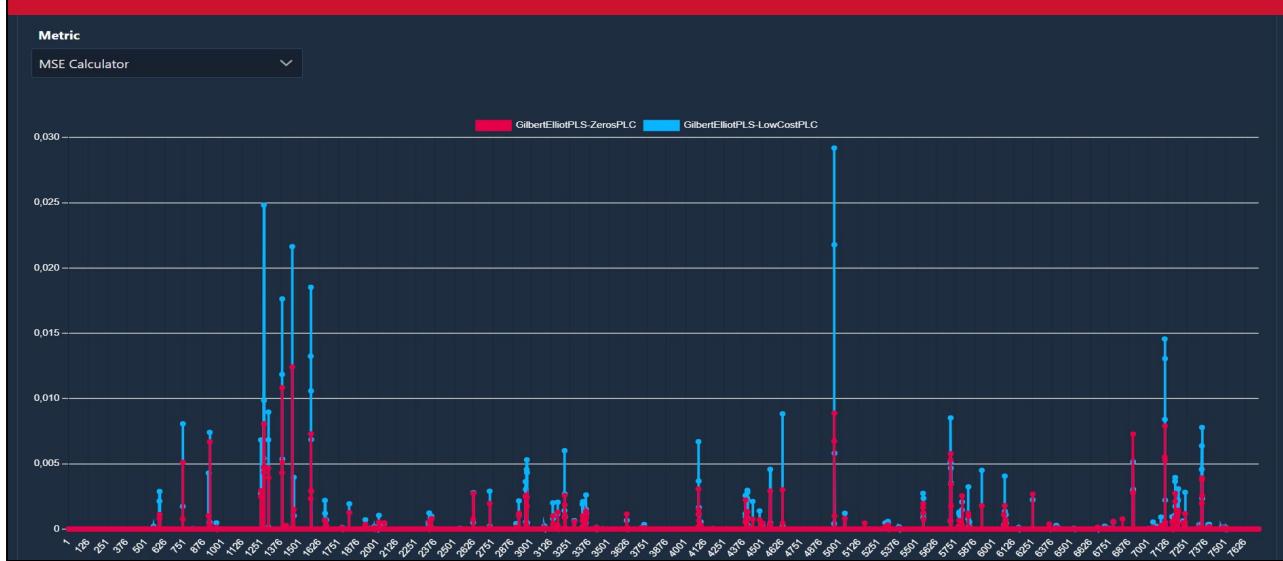


# GUI - Analysis Module - 3



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** indicate a 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 List

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.



# GUI vs Native Interaction - 1

The screenshot displays two side-by-side interfaces for a testbench. On the left is a **web-based GUI**, and on the right is a **native Python interface**.

**Left (Web GUI):**

- Run List:** Shows a table with columns: Run ID, Files, Created On, Creator, Status, Actions.
- Run View:** Shows a search bar for Run ID and a dropdown for Status (Any, COMPLETED, FAILED).
- Input File Selection:** A dropdown set to "Blues\_Bass.wav".
- Packet Loss Simulator:** A dropdown set to "Gilbert Elliot PLS".
- RLC Algorithm:** A dropdown set to "Selected algorithms".
- Output Analyzer:** A dropdown set to "None".
- Parameters:** A table showing configuration values for Gilbert Elliot PLS.

**Right (Native Python Interface):**

```
use('stefano_dot_dallona_at_gmail_dot_com')  
db['RunView'].find()  
  
{"run_id": "13292225527426488",  
 "status": "COMPLETED",  
 "description": "",  
 "classname": "Run",  
 "run_id": "13292225527426488",  
 "creator": "stefano.dallona@gmail.com",  
 "created_on": "2023-10-27T01:00:12",  
 "selected_input_files": [  
     "Blues_Bass.wav",  
     "Blues_Guitar.wav"  
 ],  
 "lost_samples_masks": [
```

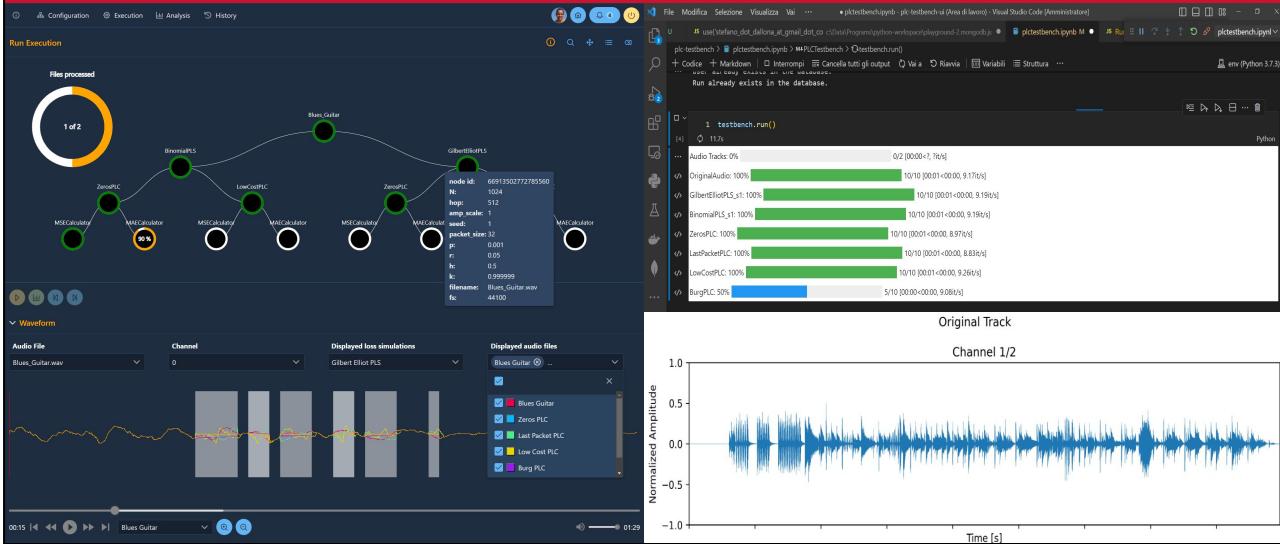
This comparison highlights the visual and interactive nature of the GUI versus the procedural and code-oriented nature of the native interface.

This slide shows a visual **comparison of** how elaborations inquiry and configuration are performed **in the web GUI versus** how they are carried out through **the native testbench interface**.

On the left in the **GUI** supports the interaction is completely **visual** while on the right, in the **native mode**, the same operations have to be performed **at programming level**.



## GUI vs Native Interaction - 2



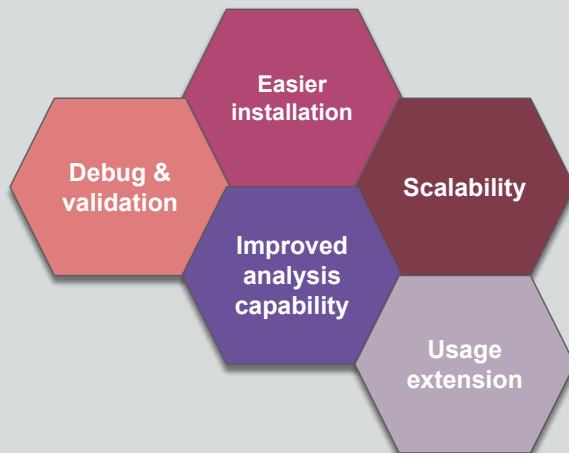
Here the **same comparison** is applied to the **progress monitoring and results analysis use cases**.

Also in this case the web **GUI** provides a more convenient and effective interaction by presenting **output information** in a **more detailed and integrated way**.

# 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 encapsulating the **complexity** in the **build process of the docker image**;
- the **application** has been made **more scalable** by making **distributed deployment** possible;
- the possibility to **use** the tool has been **extended** to **users with no expertise in Python programming**.



## Future Work

### A few ideas

- *Extensive validation with users (\*)*
- *Support for additional audio formats*
- *Search functions on output results*
- *User collaboration use cases*
- *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.**

A few ideas for **possible future enhancements** are:

- Extensive validation with users;
- Support for additional audio formats;
- Search functions on output results;
- User collaboration use cases;
- Data export.



## 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](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 4<sup>th</sup> 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  
IS<sup>2</sup> 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**.



# 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**;
- **interaction mode** has been turned from **programming level** into **graphical**;
- **installation complexity** has been completely **hidden**.

## Acknowledgements



Thanks to

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

**for the invaluable feedback  
and support provided**