

Tuomo Sipola

Frequency phenomena analysis and visualization

Report for eÄlytelli project in Information Technology

March 10, 2021



JAMK University of Applied Sciences

Information Technology

Preface

eÄlytelli project develops knowledge and information about the possibilities of IoT's data analysis for the use of companies and other actors by building a research and testing system for the needs of research and education. The operational environment is developed so that it can be used as a learning platform suitable for the needs of degree programmes. With the project, awareness is raised about the opportunities of IoT among diverse actors and at the same time, the know-how of the project executors is increased, which improves the opportunities to train new experts. The purpose of the project is to develop and pilot concrete data analysis solutions from selected fields of business in order to create new business, achieve growth in the already existing business as well as to increase the number of jobs. The developed testing environment gives good possibilities to do different kinds of AI pilots, integrates users to the trials, supports setup phase and needed changes, shows the advantage of the solutions to the potential customers, and gives a solid foundation for the distribution of the developed methods.

List of Figures

Figure 1. Ramp up measurement and ideal interpolation.....	6
Figure 2. Frequency spectrogram of each window	7
Figure 3. Phenomenon amplitude graph with envelope	9
Figure 4. Phenomenon amplitude going over the envelope, causing an alert.....	10
Figure 5. Evaluation graph	11

Contents

1	WP2 OBJECTIVES	1
1.1	Objectives	1
1.2	Fullfilling the objectives	1
2	INPUT DATA	3
3	TECHNOLOGIES	4
3.1	Hardware recommendations	4
3.2	Execution time.....	4
3.3	Data input dependencies	4
3.4	Data analysis and visualization dependencies	4
3.5	Configurability	5
3.6	Saving results	5
4	FREQUENCY ANALYSIS	6
5	PHENOMENA.....	8
6	VISUALIZATION	11
7	CONCLUSION	12

1 WP2 Objectives

Work package 2 *Data processing and pre-analysis* of the eÄlytelli project had the following objectives, as set in the Project Plan dated 10 March 2020. In the Cooperation Agreement dated 29 January 2020, work package 2 duration is set from Q2/2020 to Q2/2021.

1.1 Objectives

- Task 2A: Data analysis report specs. Deliverable: Specification.
- Task 2B: Data analysis method replication. Deliverable: Code implementing the desired functionality.
- Task 2C: Data visualization. Deliverable: Code implementing the desired functionality to produce figures (no UI).
- Task 2D: Data analysis report generation. Deliverable: Code implementing automated report generation.

1.2 Fullfilling the objectives

Task 2A deliverable was achieved through agile development methodology. There was no separate specification document during the project, as development was communicated via teleconferencing and emails. Actual development tasks were documented as Labranet Gitlab private issues.¹ In addition, this project report describes the needs on which the analysis was based.

Task 2B deliverable has been delivered to Moventas. Code implementing the requested features is available at Labranet Gitlab in a private repository.² These features were frequency analysis and detecting vibration phenomena from the data.

Task 2C deliverable features are available in the code repository as part of the program code. The desired visualization and even some UI elements have been implemented.

1. <https://gitlab.labranet.jamk.fi/ealytelli/moventas/-/issues>

2. <https://gitlab.labranet.jamk.fi/ealytelli/moventas>

Task 2D deliverable feature has been implemented and is available in the code repository.

2 Input data

The input data used in the analysis was in TDMS format. The data contained ramp up test scenario measurements of a windmill gear. The data fields are listed below:

- RPM [$1/min$]
- Power [kW]
- Torque [Nm]
- 18 vibration channels [m/s^2]

These data fields are processed into a data frame, which can be used for analysis later.

3 Technologies

The analysis was carried out using Python 3.8 programming language.

3.1 Hardware recommendations

The following hardware was used for running the program:

- CPU: Intel® Core(TM) i5-8265U CPU @ 1.60GHz (8 processors)
- RAM: 16 GB
- Disk space:
 - Input data TDMS file size: 2.1 GB
 - Output figures, report and data: 65 MB

3.2 Execution time

The execution time of the whole analysis pipeline is about 4.5 min, which is below the expected time.

3.3 Data input dependencies

There were three potential data input methods. UFF files were used initially via the *pyuff* package, but reading from this format was too slow. TDMS files were more usable in practical terms using the *nptdms* package. The third option of using an SQL connection to read the data from the database was only discussed.

3.4 Data analysis and visualization dependencies

Most of the code is related to data analysis. The *process_data.py* module is the main starting point for the analysis. It calls functions in *ramp_model.py* to create ramp model.

The most important outside dependencies include *Pandas* library for efficient data frame

manipulation, *scipy* for frequency analysis, interpolation and pairwise distance calculations, *matplotlib* for drawing graphs and *FPDF* for PDF report generation.

3.5 Configurability

The most used options are configurable in the configuration file *config.ini*. Below is an example of the default configuration file.

```
[LOGGING]
level = INFO

[RESULTS]
save_figures_dir = figures
save_report_dir = report

[DATA]
save_data_dir = data

[ENVELOPE]
envelope_file = envelopes.xlsx
```

Logging level can be set to either *INFO* for normal use or *DEBUG* for copious debug information. The destination directories of figures and PDF report can be set in the *RESULTS* section. The phenomenon amplitude data can also be saved in a directory defined in the *DATA* section. The amplitude envelopes are defined in the *ENVELOPE* section.

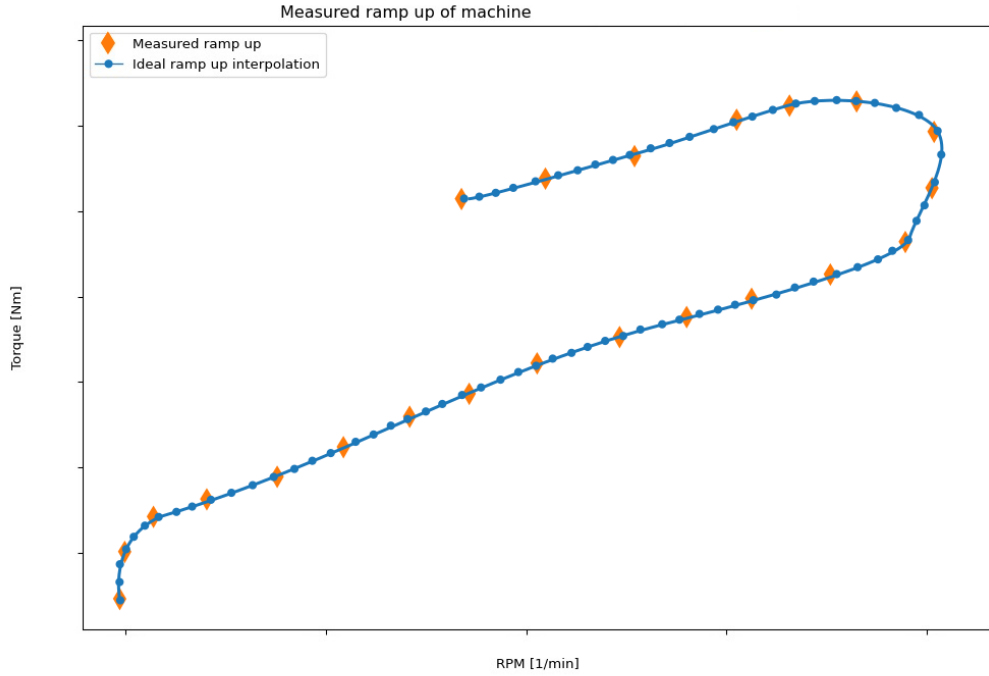
3.6 Saving results

The created visualizations are saved in the directories as indicated by configuration. The default directory for all the figures is *results*. The PDF report can be found in the *report* directory by default.

4 Frequency analysis

Firstly, the measured data needs to be aligned with the ideal ramp. Figure 1 shows the measured ramp up and the ideal interpolation from that data based on ideal ramp up. To speed up the search of the nearest point in the raw data, only every 10240th point is compared. From this interpolated ramp, 300 measurement point are selected.

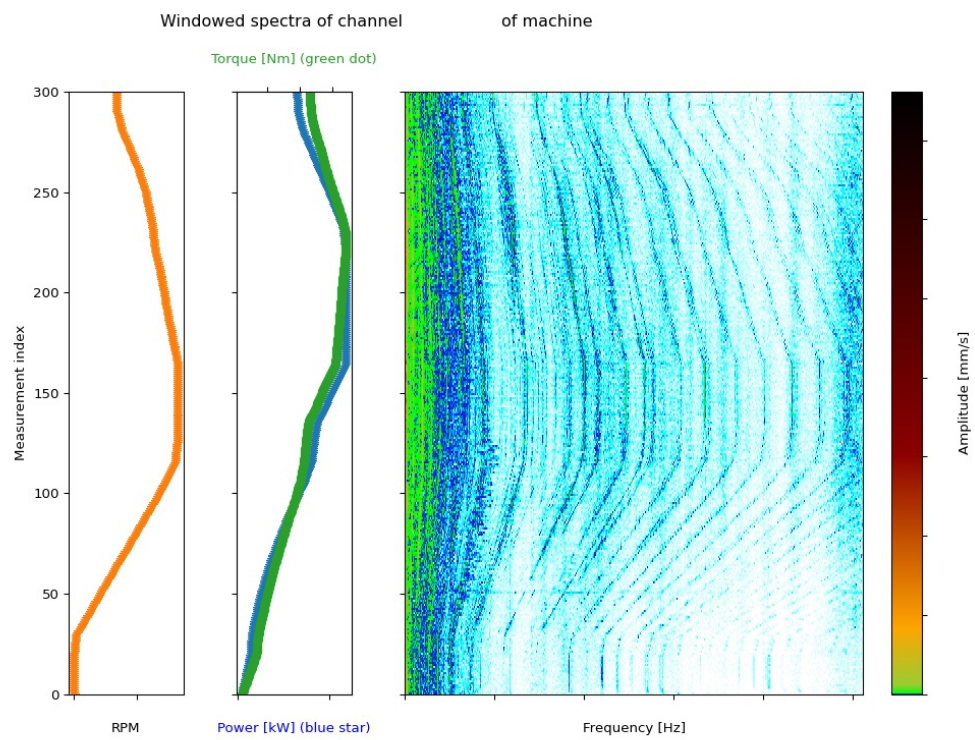
Figure 1. Ramp up measurement and ideal interpolation. (Schematic presentation.)



The window size in samples is $\text{window size} = 0.5s \times f_s$, where $f_s = 10240\text{Hz}$ is the sampling frequency. This number of samples of raw data vibration is selected from each measurement point for a window.

The raw vibration data is arranged into 300 windows, corresponding to the measurement points. Each window is run through fast Fourier transform (FFT) to obtain the frequency distribution during that window. Figure 2 shows the frequency distribution of each measurement point. The figure shows the corresponding RPM, Power and Torque values at each measurement point.

Figure 2. Frequency spectrogram of each window. (Exact values removed from the image.)



As a result, 300 frequency distributions are created. In addition to visual inspection of the spectrogram, further analysis can be performed on this data.

5 Phenomena

Pre-defined interesting phenomena are followed in the spectrogram. This is done by a simple search algorithm. As a result, a phenomenon amplitude time series is created. It represents the evolution of a certain known vibration response in the gear.

Each of the 18 channels is analyzed individually, finding each of the 10 phenomena.

Pre-defined envelopes describing the limits of normal behavior are used for each phenomena. If the found phenomena amplitude goes over the envelope, an alert is triggered. The current alert rules are the following:

- Warning if amplitude goes over the envelope.
- Alert if amplitude is above the envelope 15% of time.
- Alert if the biggest peak amplitude goes over 30% the envelope level.

Figure 3 shows an example of the phenomenon amplitude graph.

Figure 4 shows an example of an alert caused by a peak in the data.

Figure 3. Phenomenon amplitude graph with envelope. (Exact values removed from the image.)

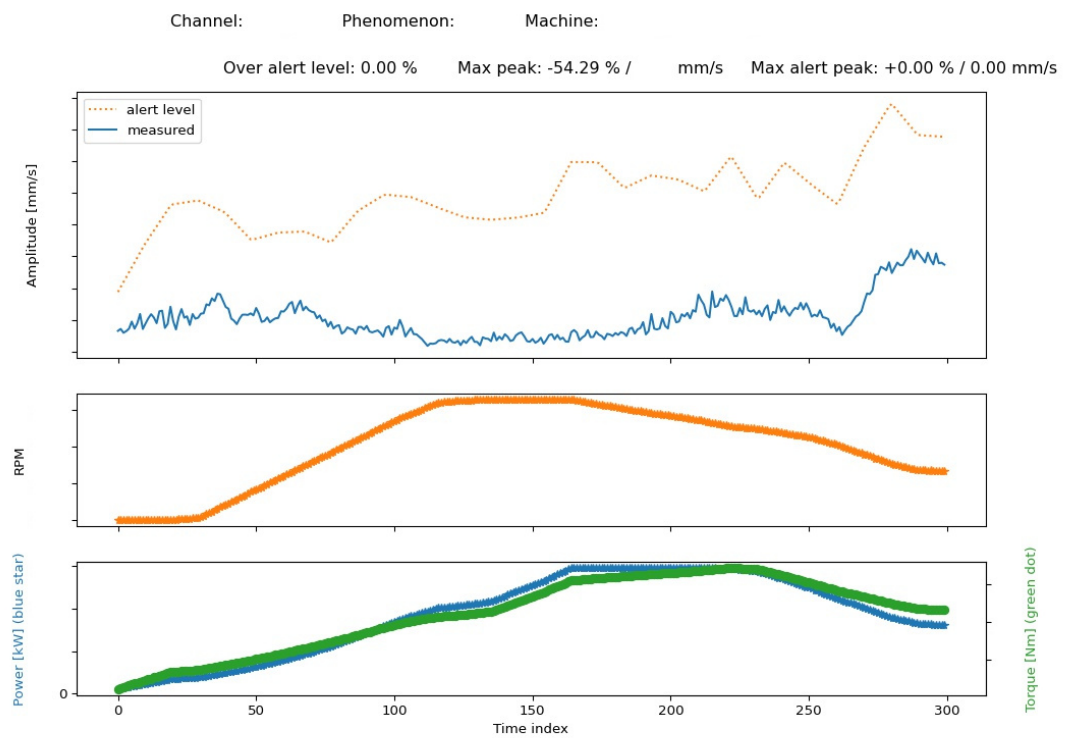
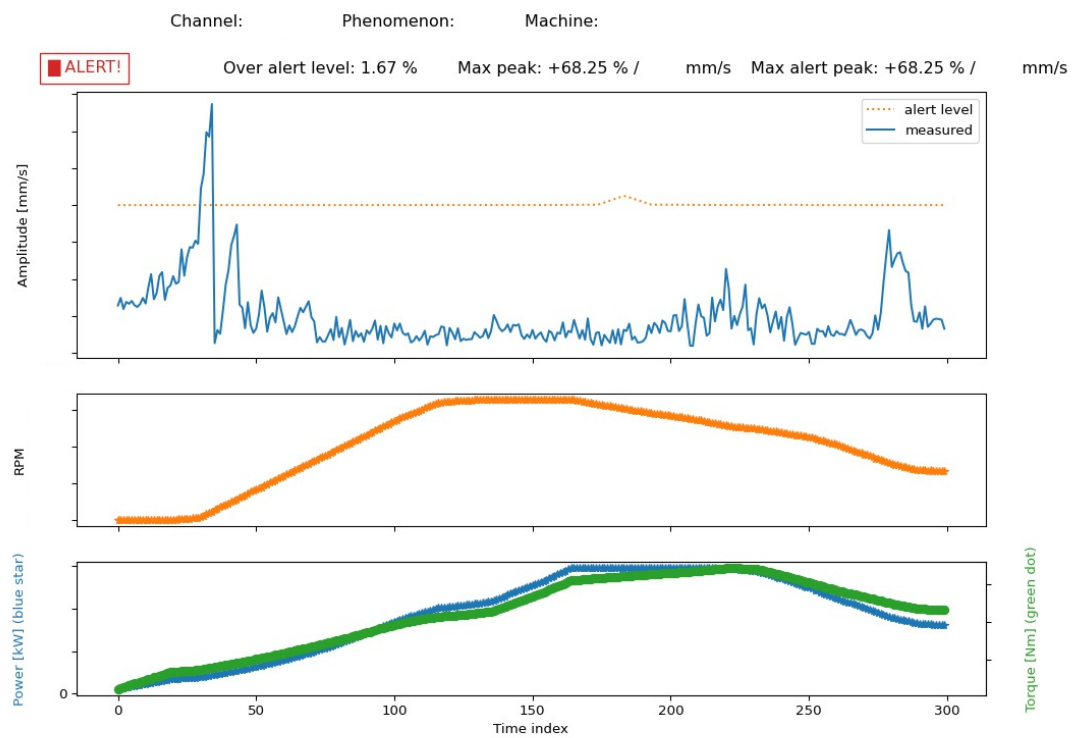


Figure 4. Phenomenon amplitude going over the envelope, causing an alert. (Exact values removed from the image.)



6 Visualization

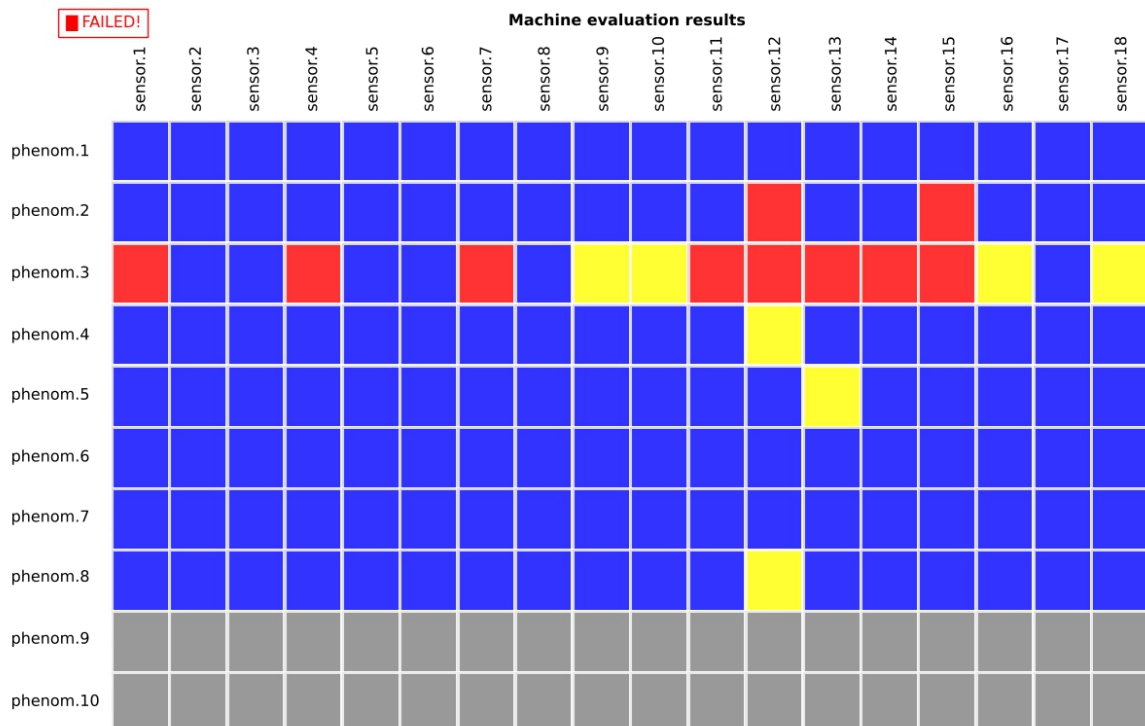
As seen above and below, four types of visualizations were done:

- Ramp up measurement graph,
- Frequency spectrum graph,
- Phenomenon amplitude graph,
- Evaluation matrix graph.

These graphs are created using the *matplotlib* Python library. The graphs are saved as JPEG images and further combined into a PDF report.

As a final result, an overall view of each phenomena is created. This matrix view shows an easy to understand summary of the status of the test scenario, as seen in Figure 5. The pass/fail status of each phenomena is shown here.

Figure 5. Evaluation graph. (Generic names replace the actual values.)



7 Conclusion

The results of work package (WP2) satisfy all the goals that were set in the project plan. The program code produced during the work package takes a ramp up measurement file as input and creates a PDF report. The report contains an overall evaluation graph, along with frequency phenomena analysis graphs.

The project used Python as the programming language to implement an analysis system. The execution time of the analysis from start to finish is below 5 min. Frequency analysis of the ramp up measurement is done, which enables more detailed phenomenon analysis. As an end result, a summary is produced in the form of an evaluation graph.