




**IMT MINES ALÈS**

**Department of Computer Science and Artificial Intelligence**

# Case Study

# Project OROWAN

**PREPARED BY:**

Dalin SOUN

Oumaïma BOUNHAR

Wenyao LAN

IMT Mines Alès
École Mines-Télécom

2ia
informatique et
intelligence artificielle

# 2022-2023

# OROWAN

## 1. INTRODUCTION

The studies of this project are technically on industrial applications based on friction. The scope of work is an integration of steel production that starts from a resource like a hot metal into the production line to get a hot strip mill. In a hot strip mill, slabs are transformed into coils. The hot strip mill's main objective is to reduce production weakness and width. Moreover, it can obtain the surface quality and the mechanical properties the customer requires. The four main steps of the hot strip mill process are reheating, rolling, cooling, and coiling. In a work-roll implementation, cooling down the hot strip mill and lubricating the bite is essential.

The main objective of this project is to develop the related system in a hot strip mill based on online friction coefficient calculation:

- Work-rolls degradation indicator
- Threading refusal prediction and prevention system
- Lubrication efficiency indicator and recommendation for change in preset of lubrication
- Preset model for roll force, forward slip, and torque.

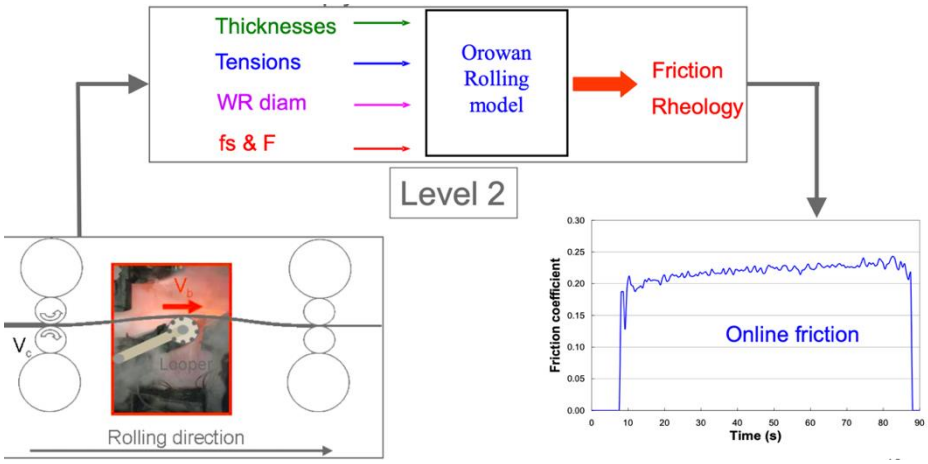


*Figure 1: Diagram of Online friction Back-Calculation*

Since the objectives are based on the computer implementation of the rolling force model, this project uses the Orowan application. It is written in C++ and can generate a library for PC using the visual studio development environment. The model includes two functions: direct calculation and backward calculation. In this case, online friction back calculation is a robust solution. Because forward slip measurement increases the encoder on the looper, which allows good friction precision. Furthermore, an online calculation model (OROWAN) based on force and forward slip to obtain the friction coefficient and strip yield stress.

## 2. SYSTEM REQUIREMENT

This project requires many steps or approaches to design and assist during the implementation. First, this project is given the dataset in a TXT file that can either convert to CSV or remain as a TXT file. There are two given TXT files in names: "1339351_F2.txt" and "1339351_F3.txt", respectively. Then, it requires reading and storing those data in the H2 database.

The second step is to improve the Orowan model by computing the Orowan model in real-time every 200ms with the last available value. Moreover, computing the average of the last 5 Orowan friction coefficients every 1s and store in the database.

The third step is to design the interface that shows the graph or curves for each output, like friction coefficient, sigma, and roll speed. Besides the interface, it should deliver password protection for the users.

The final step is to store level II data in a time series, store Orowan results in a time series database, and show data from a time series database.

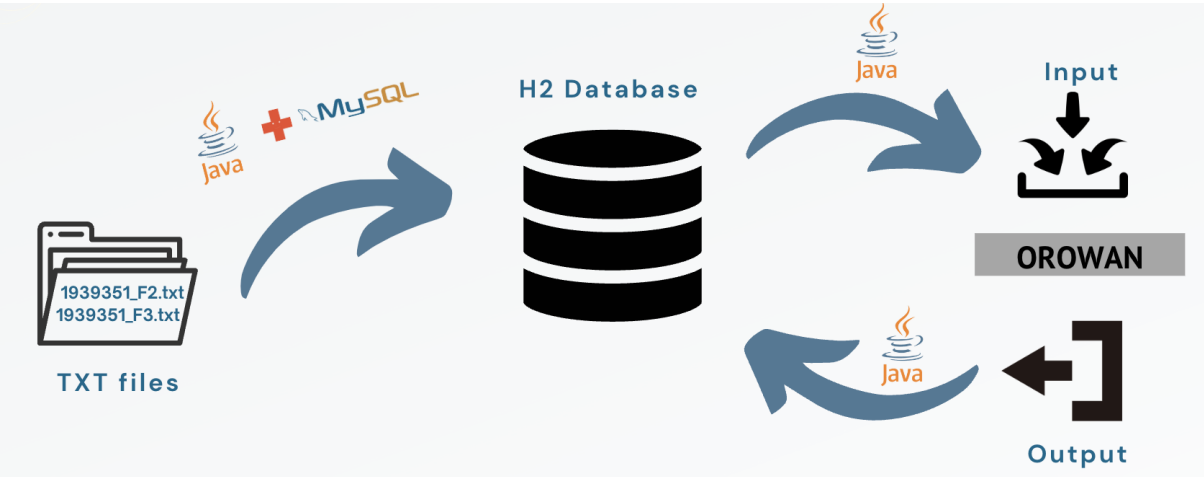


*Figure 2: Description of the Project*

## 3. DESIGN OF A SYSTEM

### 3.1 Conception

In the system, we considered the concept of the data model that can help us narrow down the existent input and outputs. There are many input parameters, as shown in Figure 3 on the left-hand side. However, the crucial outputs we want to study are the friction coefficient and rheology (sigma). On the other hand, the package diagram shows the arrangement of various model element structures. The application diagram contains the database, interface, and Orowan application. Moreover, the database is required to import or use the SQL programming language. Furthermore, the JavaFX application has a heritage from the whole application package.

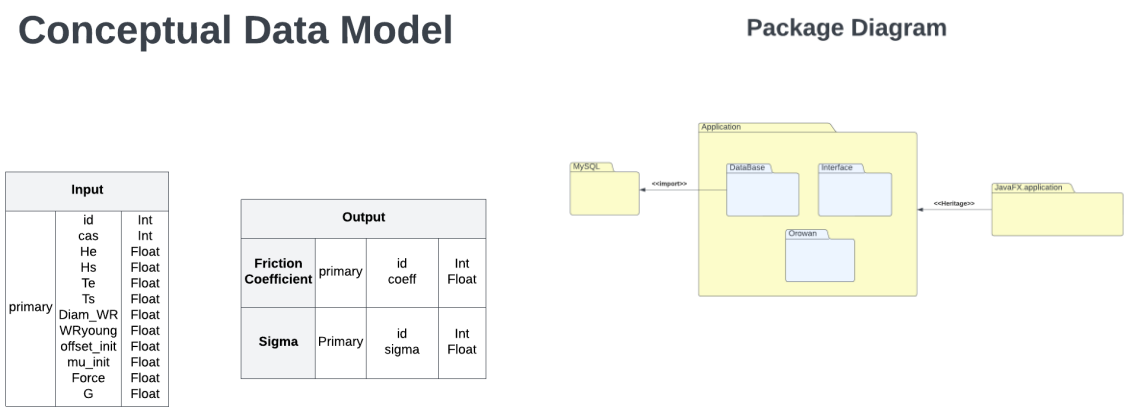


*Figure 3: Conceptual diagram (left hand sided) Package Diagram (right-hand sided)*

## 3.2 Diagram Class

This system involves several classes and processes for managing and analyzing data. It includes a database for storing input and output values and executing SQL queries on the data. The SystProcess class calls Orowan.exe and calculates the average of the 5 values of each output. The Graph class controls a graphical interface that displays different types of graphs based on user input, and the SqlGraph class is similar but has a button for launching the interface Graph_and_SQL.fxml.

The NewFile class is used to rewrite the files about reels so that they become compatible with the software "orowan". The User class creates engineer and worker passwords, while the Password class controls the interface "Login", which is an identification interface that requests a password. There is a page for each password (engineer or worker).

The SqlQuery class controls the interface "SQL_Query", which opens when we click on the button sql from "Graph_and_SQL", allowing engineers to write SQL queries. Finally, the Main class connects to the database and launches the Login interface.

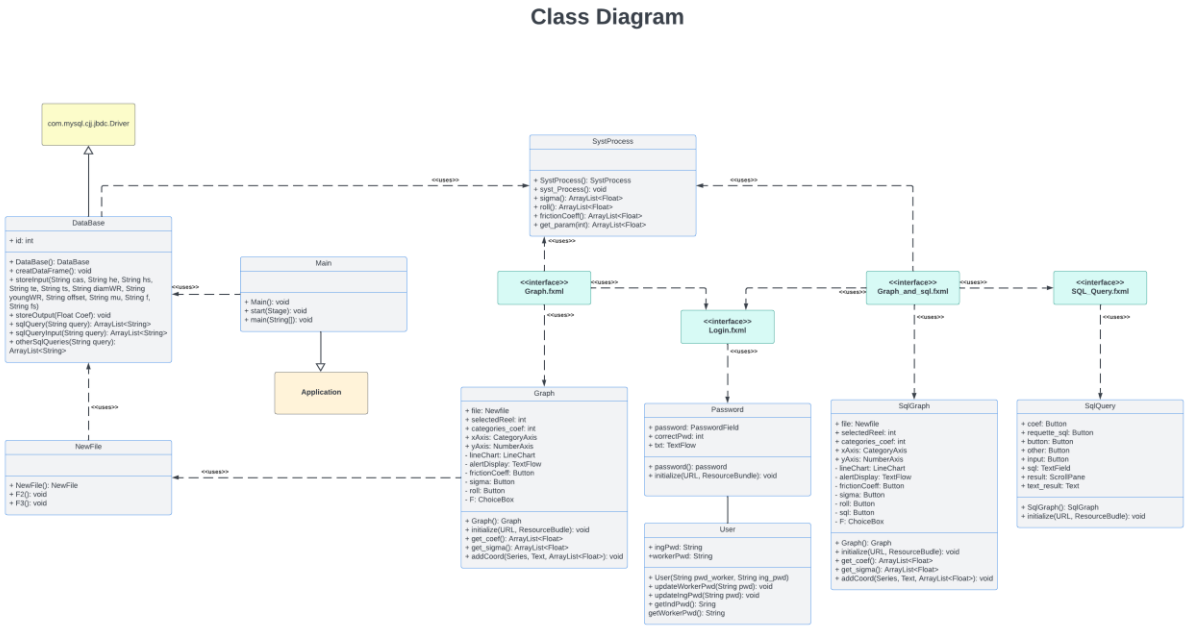


*Figure 4: Class diagram*

## 3.3 Specification

We create the use case diagram to identify the interactions between the system and its actors in the specification that precise the requirements. For example, in this case, there are three crucial actors, the worker, engineer, and chief operator, as shown in Figure below. Following the use case diagram is the Kaos diagram, which captures the requirement engineering approach. It is specifically a goal modelling, the responsibility of a model, the operation model, and the objective model. As we can see, it mainly describes the system requirement in the previous section of the diagram, “Kaos”.

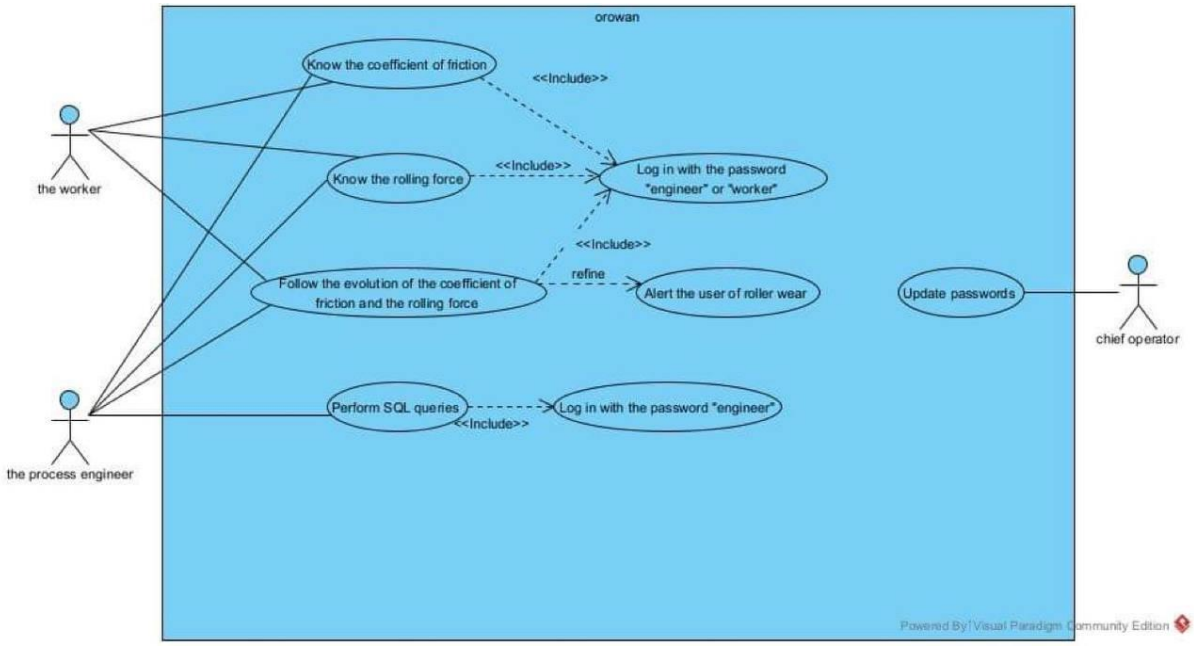


*Figure 4: Use Case Diagram*

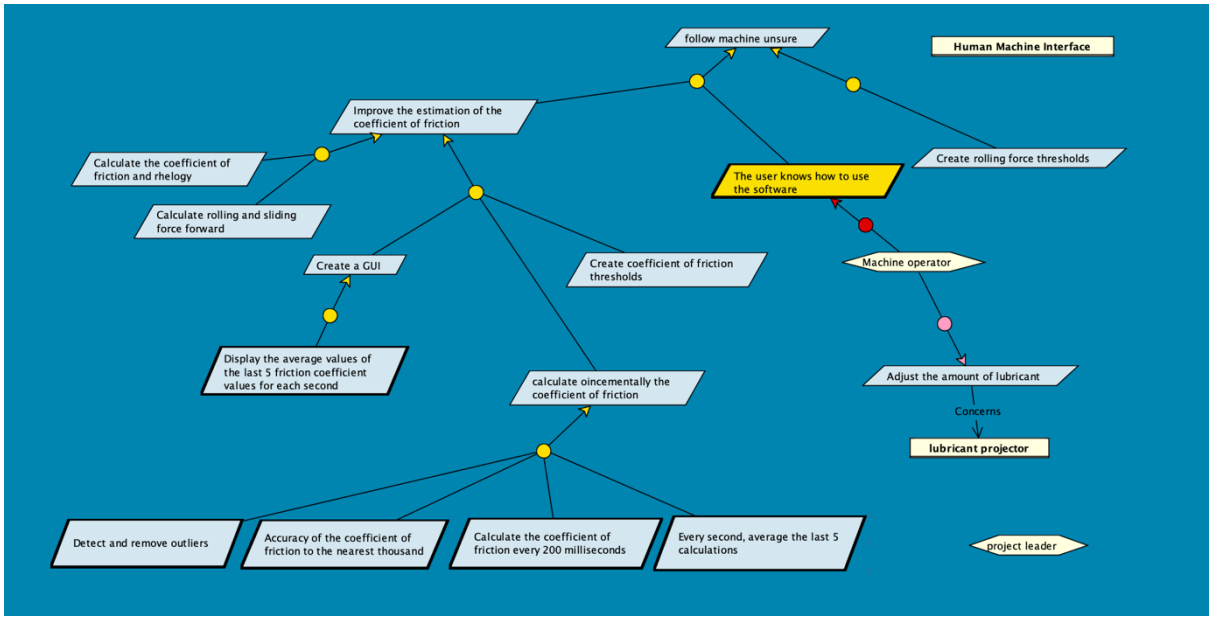


*Figure 5: Kaos Diagram*

In the implementation of software modelling, it encounters errors or broken security in some parts. Therefore, we can use an alloy analyzer to find a hole in the system's security mechanism. Figure 6 shows Alloy's implementation, which is mainly focused on the signature of the interaction between systems. Unfortunately, we haven't yet carried out the check or predicate function to verify if an issue might occur.

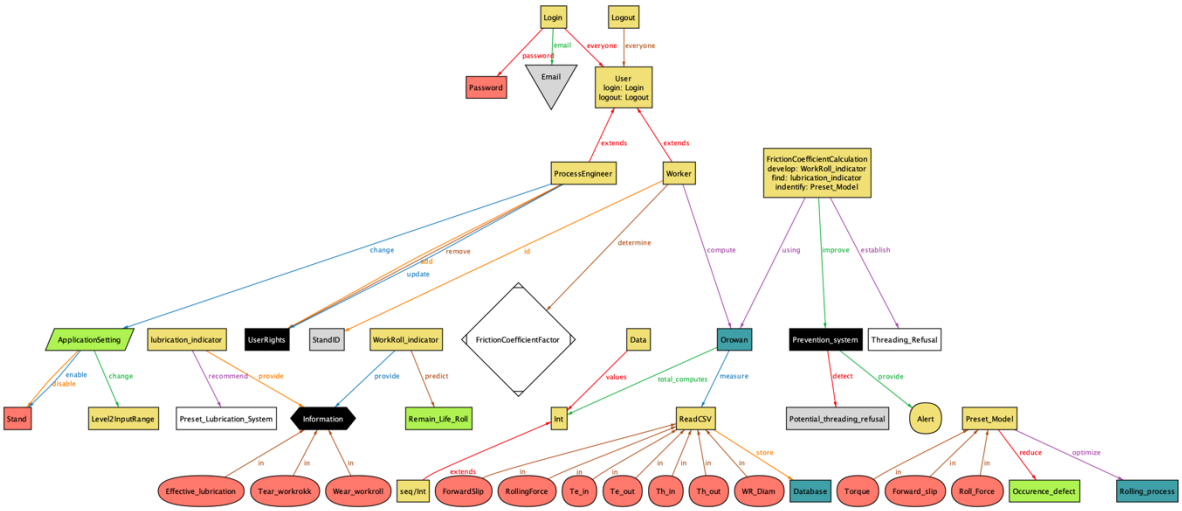


*Figure 6: Alloy*

# 4. RESULT

## 4.1 Interface

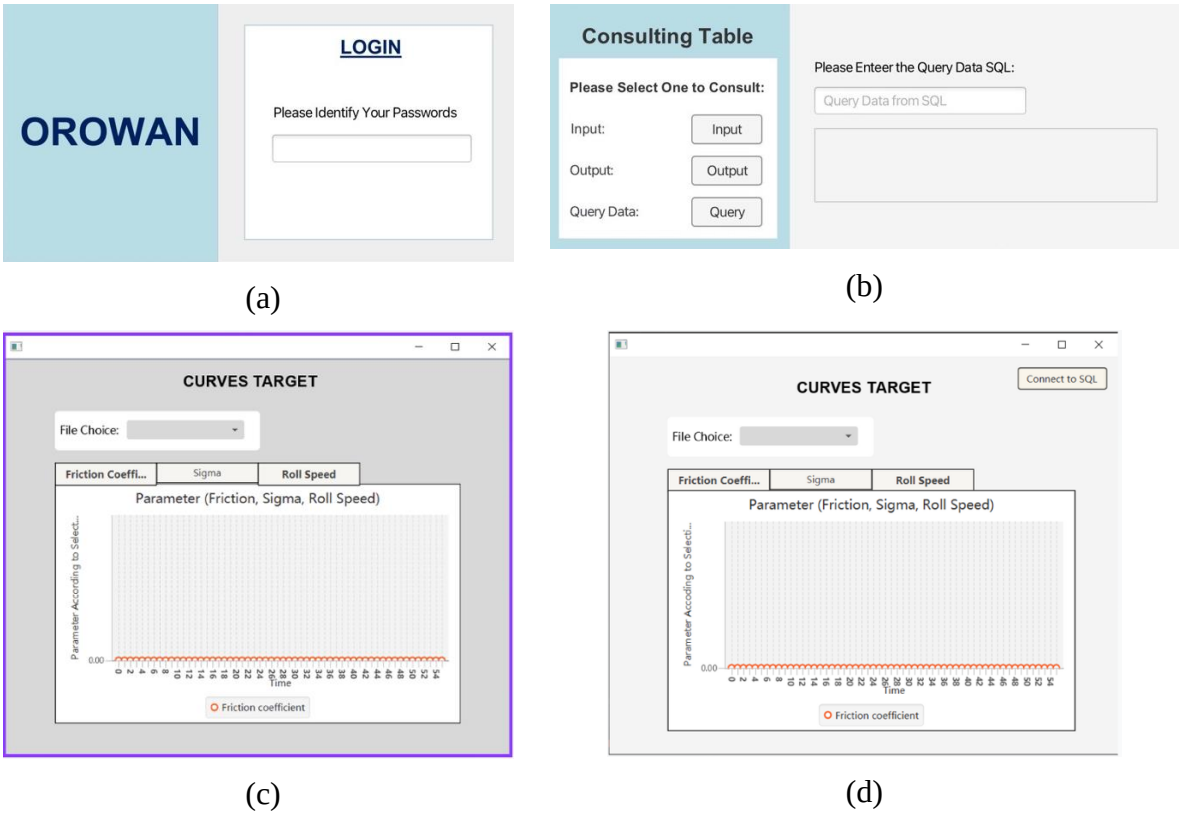


*Figure 7: Interface (a). Login, (b). Query Data SQL, (c). Graph Interface for Worker, (d). Graph Interface for Engineer*

## 4.2 Input in Database

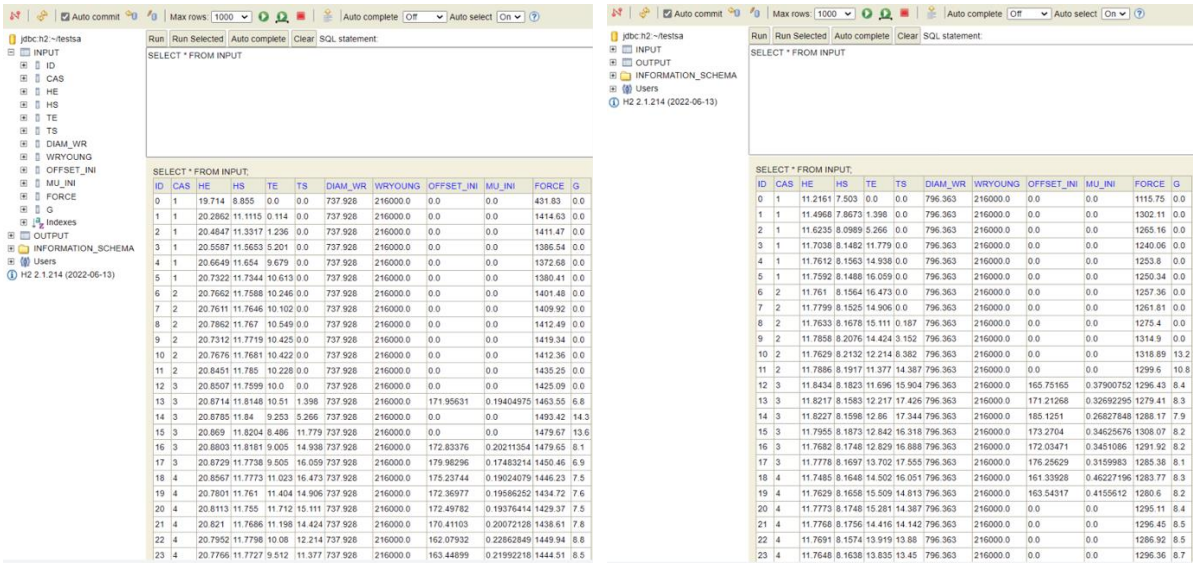


*Figure 8 : Input F2 (left-hand sided), Input F3 (right-hand sided)*

## 4.3 Output in Database

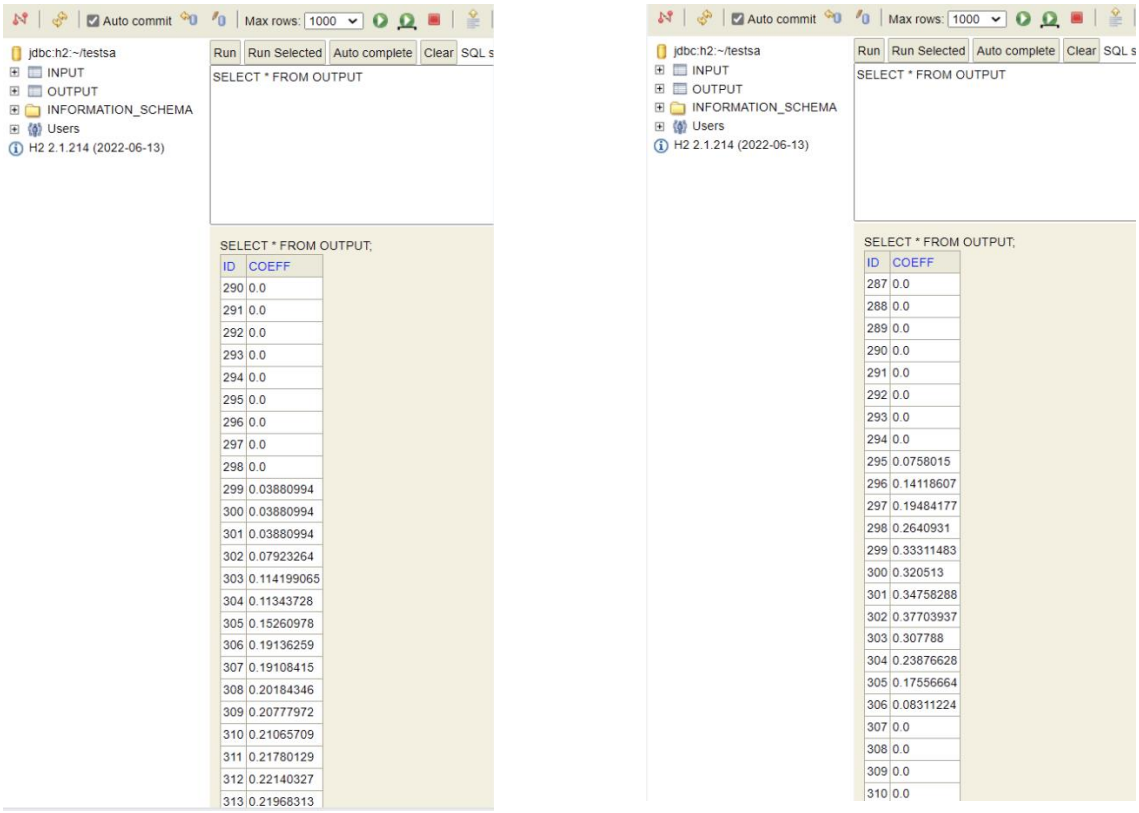


*Figure 9: Output F2 (left-hand sided), Output F3 (right-hand sided)*

## 4.4 Plot Graph

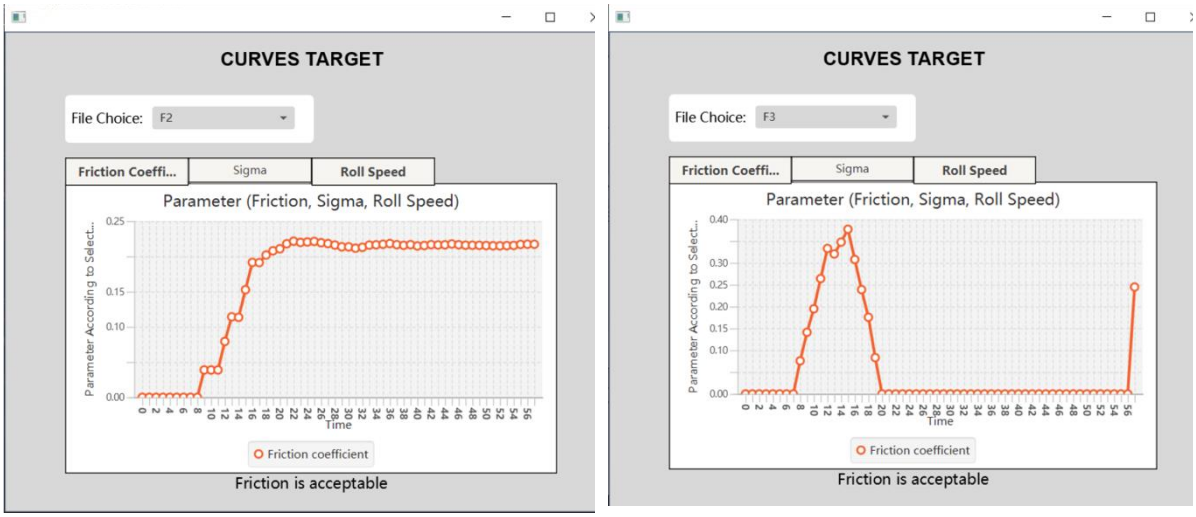


*Figure 10: Friction Coeffiecient F2 (left-hand sided), Friction Coeffiecient F3 (right-hand sided)*

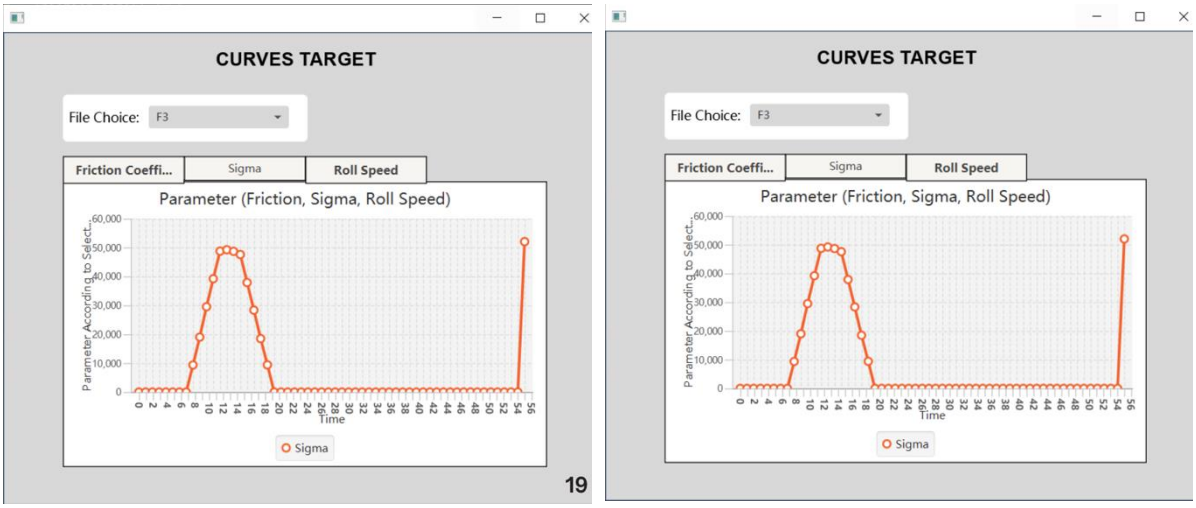


*Figure 11: Sigma F2 (left-hand sided), Sigma F3 (right-hand sided)*

## 5. CONCLUSION

In conclusion, the project was wrapped up by achieving the requirement from step one to storing data in a database. The interface design was considered for both workers and engineers. The protection passwords for workers and engineers are different and allow the login by the engineer to lead to the interfaces that can connect to the SQL database. The graph GUI can be selected the button to show the desired plot output like friction coefficient or sigma. However, the roll speed output has yet to be achieved.

Moreover, the outcome (friction coefficient and sigma) are stored in the H2 database. Unfortunately, the project still needs to carry out the real-time calculation of the output that took each 200ms and got the last available value. Initially, we faced difficulties understanding the project description, connection to H2, and not enough time for implementation; that's the major confusion made up during realization. Therefore, this project concludes only with the read-store input and computes the output inside the Orowan application directly without real-time. Nevertheless, it could be an improvement for the future after our group's achievements.