# INDUSTRIAL IOT WEB APPLICATION FOR MOTOR CHARACTERISTIC MONITORING

***Sami Jradi (Bc)***[1]

[1] ***sami.jradi@fs.cvut.cz***

*Abstract:*

*This paper is based on the technology of web development to build the back-end of an application for motor characteristic monitoring in the field of industrial IoT. The purpose of the web application is to receive data such as velocity, acceleration, position, and torque from an industrial servo motor connected to a Siemens SIMOTION D PLC using MQTT protocol, and then analyze this data in the form of charts. The resulting chart would be known as the load curve which will offer the chance to perform predictive maintenance on the hardware functioning inside the factory.*

*Keywords:*

*speed, torque, IoT, MQTT, PLC, characteristic, backend, java, SQL*

## 1 Introduction

This paper discusses the development of a web application for motor sizing and characteristic monitoring in the field of industrial IoT. The application will connect to the PLC and read the drive data of the motor being controller by the PLC. These data will include the speed, acceleration, position, and torque. This data will be represented in charts. The speed vs torque chart is called the load curve and will be compared to torque characteristic of the motor from the catalogue, additionally the operating point will be plotted as well. This comparison enables the user to identify whether the proper motor has been selected for the respective application.

## 2 Load Profile and Operating Point

The load profile is used for the purpose of motor sizing. Proper sizing is a crucial aspect of motor selection. If a motor is undersized, it will not be able to control the load, leading to overshoot and ringing. If the motor is oversized, it may control the load, but it will also be larger and heavier, as well as more expensive in terms of price and cost of operations. [1]

Every motor will have rated values of voltage, current, speed and power. Normally these are visible on the motor's nameplate and/or given in the documentation. In general, rated values represent the maximum values that the motor should be subjected to in normal conditions. However, the rated torque is often not given on the nameplate but is a very important parameter for appropriately sizing the motor. In both DC and AC induction motors, operating current is proportional to the torque, so exceeding the rated torque is likely to lead to overheating and burnout of the motor windings. Exceeding the rated torque also risks mechanical damage to couplings and the drive shaft. Simply, if the load is constant then sizing the motor consists of choosing a motor whose rated torque is slightly above the torque required by the load. The torque produced by a motor varies with speed and the torque produced by a load also varies with speed. If the motor torque is greater than the load torque, then the load will accelerate. If the load torque is greater than the motor torque, then the load will decelerate.

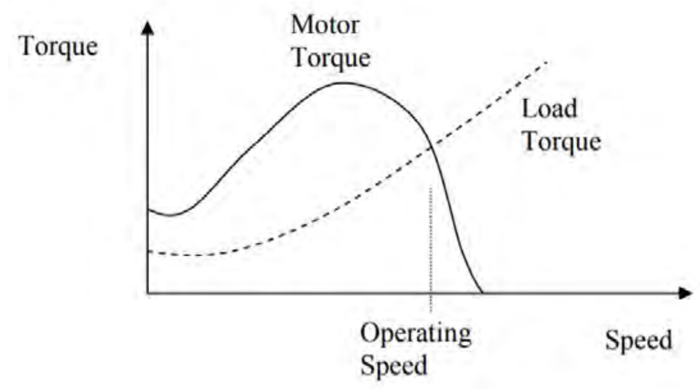


Figure 1 A motor that will start the load and get up to speed correctly

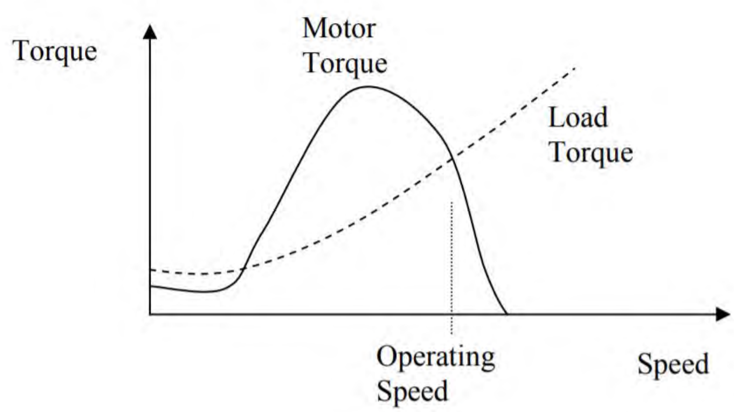


Figure 2 A motor that will never start

Some loads do not present a constant torque even after they have got up to full speed. This presents a variable power to the motor and complicates the sizing problem. In this case we should ensure that:

- Peak load torque < Rated motor torque
- The effective load torque (root mean square load torque) and effective load speed requirements must be less than 100% of the rated motor torque and speed and ideally greater than 75%.
- The motor can start the load and get it up to speed from the initial position.

The point at which the effective torque and effective speed meet is called the load point or operating point. The coordinates of this point can be calculated by the following formulas:

$$M_{eff} = \sqrt{\frac{1}{T} \cdot \sum_{i=1}^{n} M_i^2 \cdot t_i} \qquad (1)$$

$$n_{eff} = \sqrt{\frac{1}{T} \cdot \sum_{i=1}^{n} n_i^2 \cdot t_i} \qquad (2)$$

$T \ldots$ *cycle time of the measurment*
$t \ldots$ *sample time*
$n \ldots$ *number of samples*

If the motor were to be operated at this operating point, the same temperature rise would occur as in the actual load cycle. This point is now entered in its torque-speed diagram for each motor that is considered in the sizing process. If the operating point is below the characteristic curve for S1 operation, the corresponding motor is able to meet the required load cycle from a thermal point of view. All motors that meet this condition are retained for further design steps while all others are eliminated. In figure 8, we can observe that motor 1 is suitable for the load cycle while motor 2 would be thermally overloaded and therefore cannot be used.

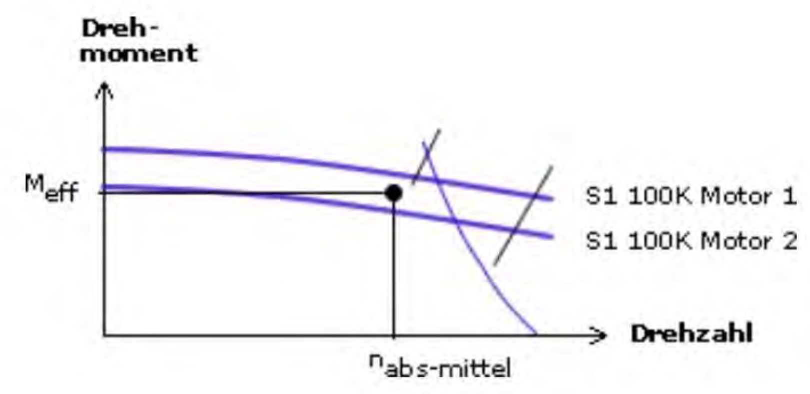


*Figure 1 Characteristic of Motor 1 and Motor 2* [2]

To get a better idea of the required motor for a specific application, then the load torque and speed will be compared to the torque characteristic of the motor. This is done by measuring the torque and speed of the motor during operation and then converting these measurements to their absolute values. We call this the load curve, and it represents the possible effective torque vs the possible effective speed.

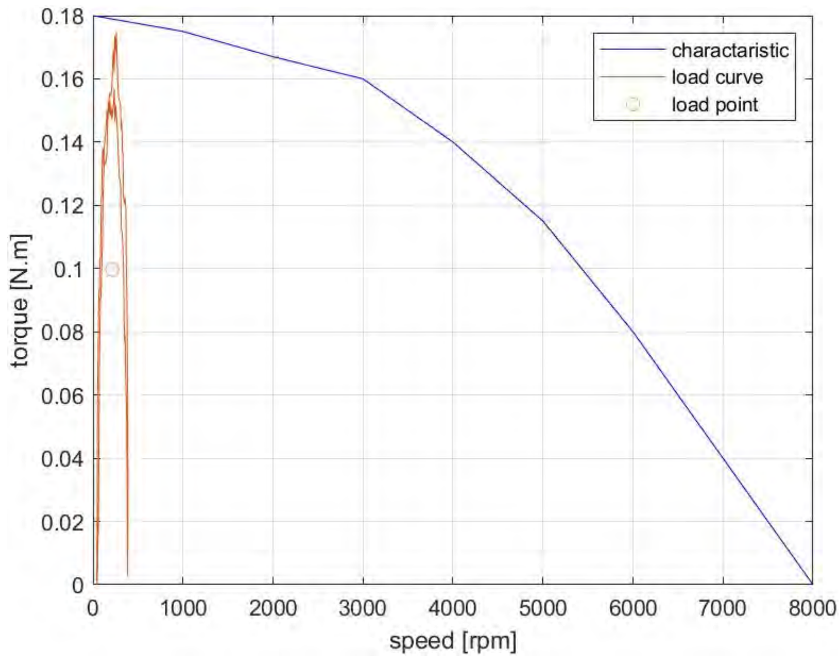


*Figure 2 An example of a load curve below the characteristic*

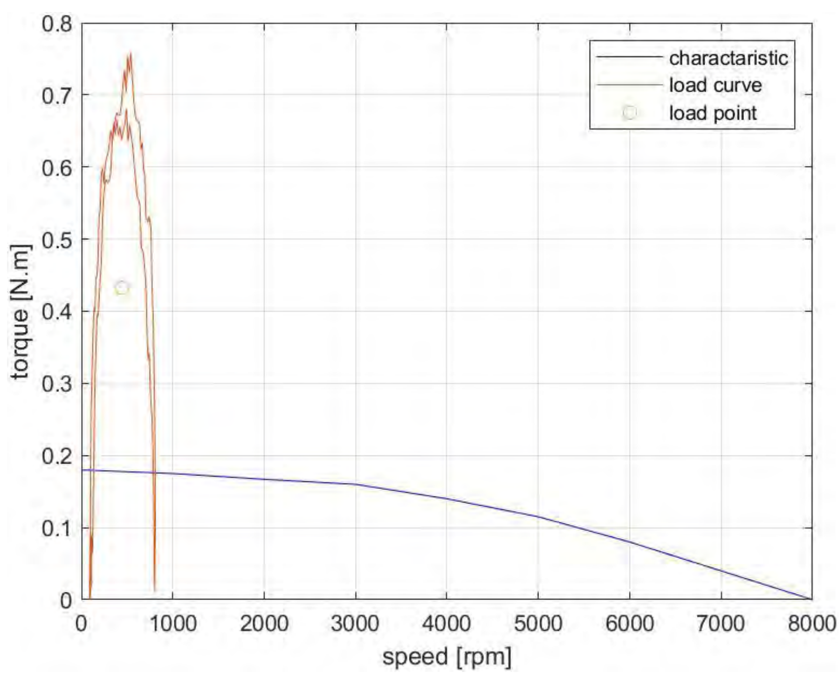


*Figure 3 An example of a load curve above the characteristic*

In figure 2, we can observe that the load curve or load profile is directly lower than the characteristic, this marks this motor as suitable for operation since the load/operating point falls below the characteristic as well. While in figure 3, the chosen motor should not be used at all and more sizing measurements have to be performed to find a better suited one for the required application.

# 3 Architectural Overview

This chapter describes an architectural overview of the application. Based on the requirements of the software, the development of the web application has been considered. The architecture consists of three main components: the back-end and front-end systems communicating with an API controller, and the database. The web application will be built as a “3-tier architecture” which comprises from a data tier, business tier and a presentation tier.

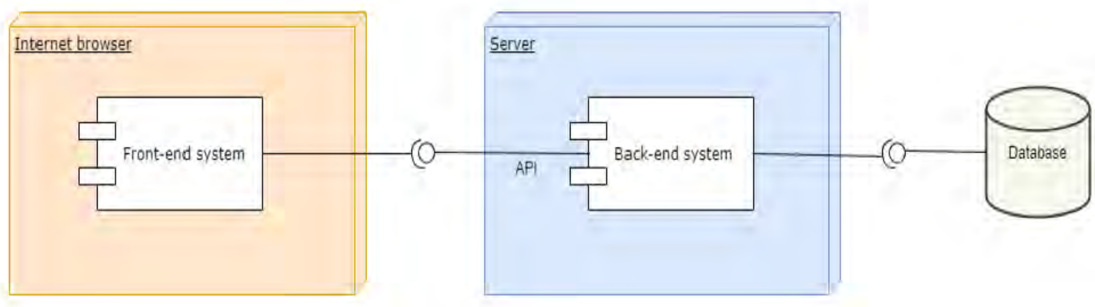


*Figure 4 An overview of the proposed architecture*

## 3.1 The Back-end

The server side of a website is referred to as the backend. It organizes and stores data, as well as ensuring that everything on the client side of the website functions properly. It's the section of the website you can't see or communicate with. It's the part of the app that doesn't interact with users directly. The parts and characteristics developed by backend designers are indirectly accessed by users through a front-end application. The backend includes activities such as writing APIs, developing libraries, and dealing with machine components without user interfaces or even scientific programming systems.

The back-end uses the Model-View-Controller (MVC) architectural pattern. The MVC pattern in Software Engineering Architecture is defined as an application being separated into three logical components: Model, View and Controller.

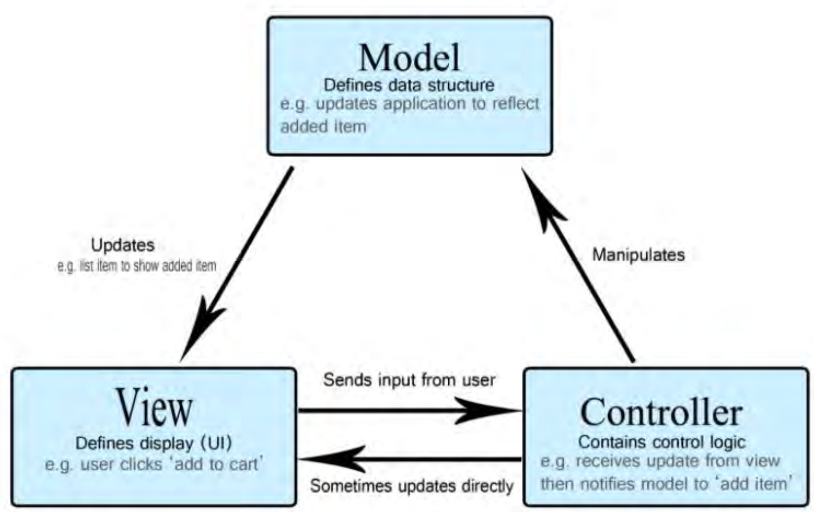


*Figure 5 MVC Architecture* [3]

### 3.2 The Front-end

The front end of a website is the aspect where the user communicates with directly. It is often referred to as the application's "client side." Anything that users see explicitly is included: text colors and styles, images, graphs and tables, buttons, colors, and the navigation menu. The languages used for Front End development are HTML, CSS, and JavaScript. Front End developers create the structure, design, behavior, and content of anything that appears on the browser screen when websites, web applications, or mobile apps are opened. The front end's main goals are responsiveness and results.

### 3.3 Database

SQL stands for Structured Query Language. A query language is a kind of programming language that's designed to facilitate retrieving specific information from databases. To put it in other words, SQL is the language of databases. A relational (SQL) database is a database that stores related information across multiple tables and allows you to query information in more than one table at the same time.

## 4 Implementation and Results

After building the web application, this section will focus on the functionalities of the app and the working principle behind them. The app will consist of four screens and one pop up window for configuration and setup. For the purpose of demonstration, some demo data has been imported into the app.

### 4.1 Configuration Screen

The configuration screen is used to connect to a PLC, browse the variables stored in its memory and create the setup of a motor axis for later measurements.

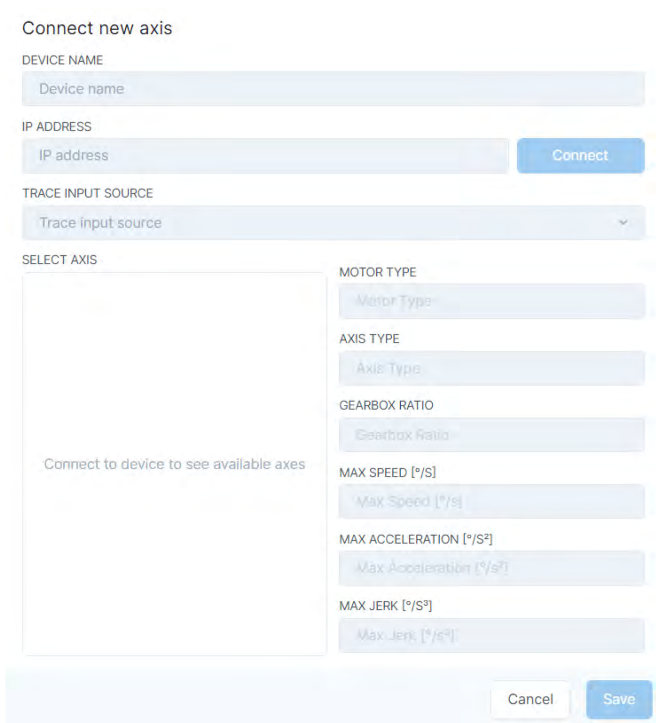


*Figure 6 Connect new axis pop-up window*

In figure 6 we can see the various fields which need to be filled before a setup came be saved. The first step is to choose a name for our machine and then enter its IP address and hit the connect button. This action will send a GET request to the server and the server will respond by connecting to the machine and retrieving the required data. This data is then sent back to the front end as a response body in the form of JSON. An example of the response should be as follows:


```
{
   "type": "D445",
   "ipAddress": "192.168.1.7",
   "serialNumber": "ST-KN6031549",
   "firmware": "V 5.3.1.9",
   "status": "CONNECTED",
   "statusTime": "2021-04-12T20:43:33.663+02:00",
   "machineNodes": [
      {
         "node": "Red_Axis",
         "saved": false
      },
      {
         "node": "Blue_Axis",
         "saved": false
      }
   ]
}
```


The response includes a property called "machineNodes". This property is a list of the available axes inside this machine and each axis contains a Boolean variable showing whether this axis has been already configured and saved or not. If the variable "saved" is false, then we will be able to select the axis from the list in the pop-up window. This action will produce another GET request to the server where the response is a JSON containing all information needed about the axis.


```
{
   "name": "Red_Axis",
   "motorType": "1FK7 synchronous motor",
   "motorMLFB": "1FK7022-5AK7x",
   "motorCode": "23726",
   "axisType": "LINEAR",
   "gearRatio": 10.0,
   "maxSpeed": 200000.0,
   "maxAcceleration": 10000.0,
   "maxJerk": 500.0,
   "dcLinkVoltage": 328.0,
   "possibleVoltages": "600.0,540.0,650.0,720.0"
}
```


## 4.2 Overview Screen

In the overview screen, we will display a list of all measurements of each axis connected. These measurements are known as "Traces" and each "Trace" object will contain information about the axis as well as a list of values of the measurements done. When loading the Overview page, the server will send a response of a list of all available "Trace" object. For this section a demo axis has been created with mock data and it's would be as follows:

```
{
  "id": 92,
  "axisName": "Axis_Demo1",
  "axisType": "LINEAR",
  "motorMLFB": "1FK7011-5AK7x",
  "gearRatio": 200.0,
  "maxSpeed": 7200000.0,
  "maxAcceleration": 360000.0,
  "maxJerk": 18000.0,
  "dcLinkVoltage": 600.0,
  "machineName": "My Demo Machine 1",
  "measurements": {
    "id": 87,
    "measurementName": "Load Curve Edge App",
    "traceStartType": "IMMEDIATE",
    "duration": 166,
    "ipAddress": "192.168.1.8",
    "tracejobId": 101,
    "cycletime": 1,
    "timestamp": 702250513000,
    "machineSerialNumber": "ST-M123456789",
    "machineFirmwareVersion": "V 5.3.1.9"
  },
  "loadCurve": {
    "id": 93,
    "datetime": 702250513000,
    "cycleTime": 1
  },
  "timestamp": 702250513000,
  "duration": 166
}
```


This response contains an object "Measurements" which includes all information about the measurements received over MQTT. However, in this response the list of values of the measurements is being suppressed to avoid the congestion of too many data being sent on each request. To access this data an "Analyse" button will be available next to each measurement which sends two responses. One response will include the values for the position, speed and acceleration charts and the second response will include the values for the torque characteristic, load curve and operating point. The responses are as follows:


```
{
  "velocity": {
    "axisX": [...],
    "axisY": [...]
  },
  "acceleration": {
    "axisX": [...],
    "axisY": [...]
  },
```

```
    "position": {
      "axisX": [...],
      "axisY": [...]
    }
}

{
    "torqueCharacteristic": {
      "axisX": [...],
      "axisY": [...]
    },
    "loadProfile": {
      "axisX": [...],
      "axisY": [...]
    },
    "effectiveSpeed": 606.8984425107654,
    "effectiveTorque": 0.17676690952747628
}
```

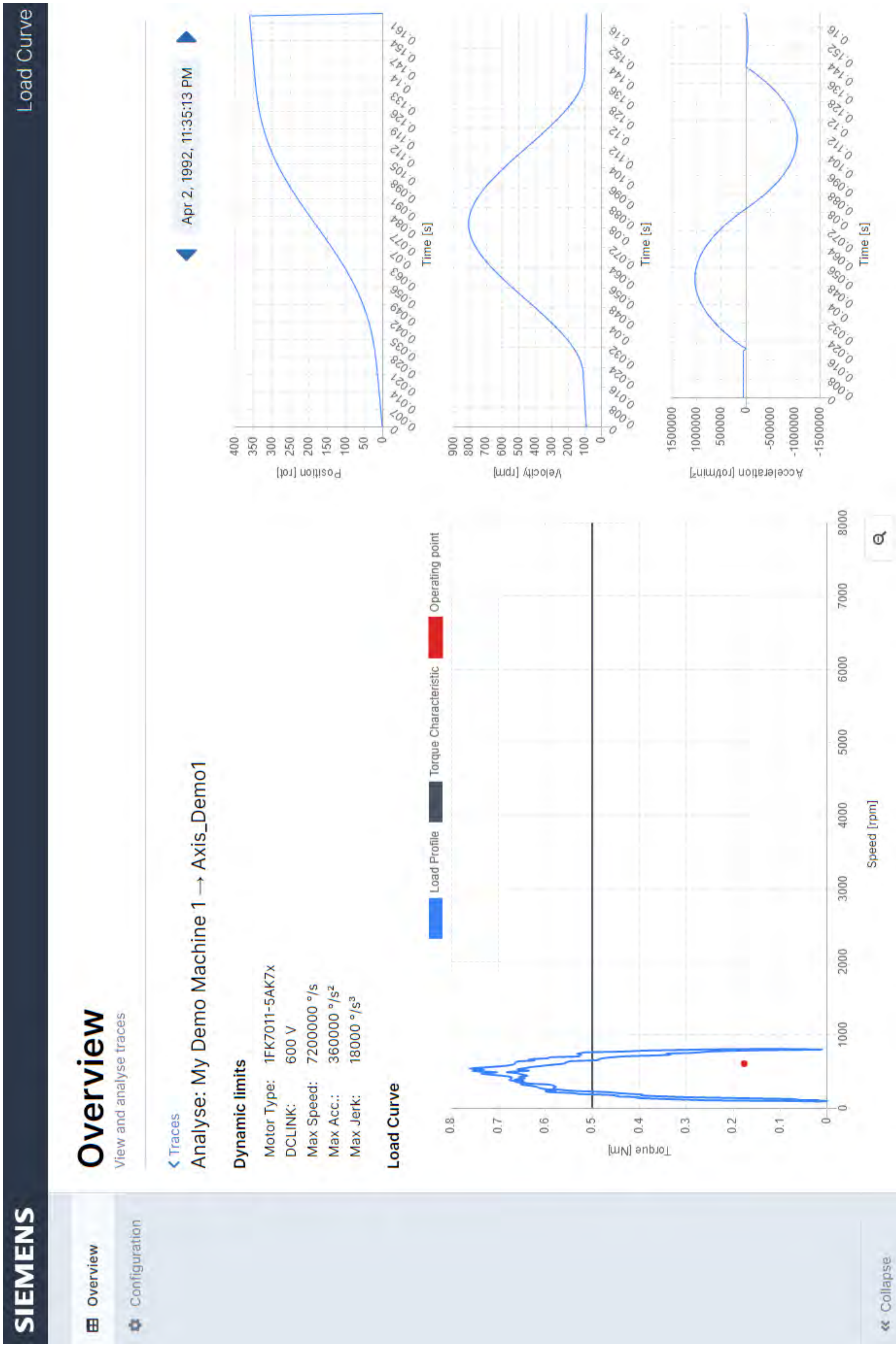


*Figure 7 A full screenshot of the overview Analyse window*

## 5 Conclusion

This paper has examined the possibility of developing an Industrial IoT web application used in the field of factory automation and production machines. The application is able to connect to a PLC using its OPC server and read variables stored inside its memory. By selecting an axis from the list, the application is then able to detect the properties of this axis and allow the user to save the configuration. After finalizing the setup, the user is then able to send a measurement over MQTT to the application and perform visualization of the data in charts. The application has proved to be functional, practical and user friendly when tested with a real-life factory setup. Future work can be done to enhance the usability of the application such as adding compatibility with other types of PLCs other than SIMOTION and allowing the user to trigger measurements directly from the application rather than sending the data from another application using MQTT.

# REFERENCES

[1] G. Kirckof, Servomotor Sizing and Application, International Society of Automation, 2012.

[2] „Servo Technik," [Online]. Available: http://www.servotechnik.de/fachwissen/auslegung/f_beitr_00_708.htm.

[3] „Medium," [Online]. Available: https://medium.com/@eugeneteu/intro-to-software-engineering-architecture-model-view-controller-c29805284de6.

Selected article from

Tento dokument byl publikován ve sborníku

# Nové metody a postupy v oblasti přístrojové techniky, automatického řízení a informatiky 2021

# New Methods and Practices in the Instrumentation, Automatic Control and Informatics 2021

15. 9. – 17. 9. 2021, Žatec

ISBN 978-80-01-06889-2

Web page of the original document:
http://iat.fs.cvut.cz/nmp/2021.pdf

Obsah čísla/individual articles:
**http://iat.fs.cvut.cz/nmp/2021/**

Ústav přístrojové a řídicí techniky, FS ČVUT v Praze, Technická 4, Praha 6