Dataset on Energy Consumption of Fischertechnik Quality Control

Aaditya Neupane

June 2025

1 Dataset Overview

This dataset contains detailed energy consumption measurements from the **Fischertechnik Quality Control.** The data was collected at 100hz, 400hz, and 800hz sampling rates. For each sampling rate, 4000 experimental runs was done where colored Tabs (`Nio`, `Bio`, `Wio`, and `Rio`) were sorted by the system. Each experimental run generated a separate CSV file containing time-series measurements of electrical parameters.

2 Experimental Setup

2.1 Physical System Components

- **Fischertechnik Quality Control (9v):** An industrial training model that simulates a real-world sorting system.
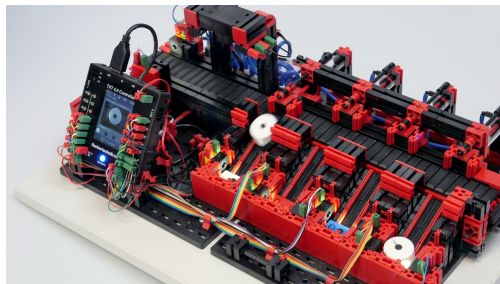

Figure 1: Fischertechnik Quality Control

  - **Vision System:** Includes a camera for Tab identification.
  - **Conveyor Belt:** Transports Tabs from the starting point to the end point of the conveyor belt.
  - **Pneumatic cylinders:** Pushes the Tabs on conveyor belt to the sorting destination.
  - **Phototransistors:** At the starting point and at the sorting destination to start/stop conveyor belt.

- **Power Supply:** Supplies 12v DC

- **DC to DC Converter:** Converts 12v DC to 9v DC

- **Arduino UNO:** Connected to a computer; it starts recording the electrical parameters from 'DC to DC Converter' when a Tab is detected by Phototransistor at starting point until it reaches (detected by Phototransistor) at sorting destination.
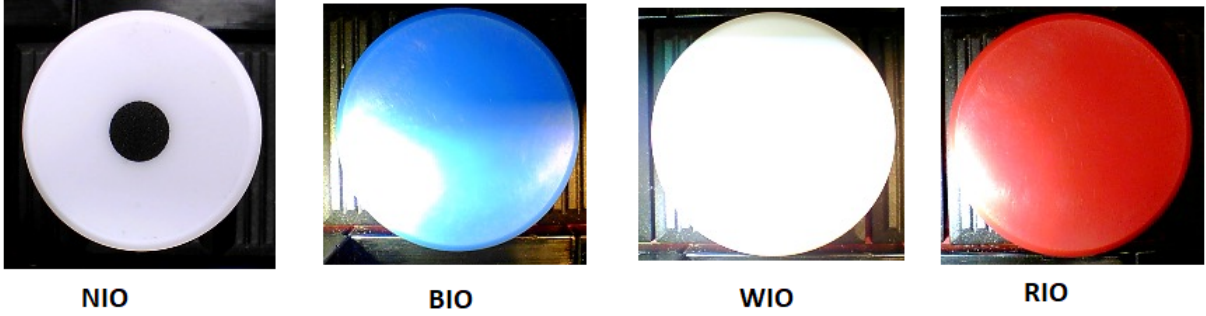
- **Tabs:** Nio, Bio, Wio, and Rio

Figure 2: Tabs

2.2 Data Collection Methodology

- **Trigger Mechanism:** Data recording begins when a Tab is placed at the starting point and stops when it reaches the sorting destination.

- **Sampling Frequencies:** Data is collected at 100 Hz, 400 Hz, and 800 Hz

- **Repetition:** Each Tab was processed 1,000 times (4 Tab types × 1,000 runs = 4,000 datasets) for each sampling frequency.

- **Automatic File Generation:** Each sorting operation generates a separate CSV file.

3 Data Description

3.1 File Structure

- **File Naming Convention:** [Frequency]Hz_Sort_[Tab-Type-Name]_[Date]_[Count].csv
  example: 400Hz_Sort_biO_06092024_404.csv

- **Total Files:** 1000 files per Tab type for each sampling frequency = 1000 x 4 x 3 = 12,000 files

- **Directory Structure**

```
├──100hz
│      ├──biO─────────├──100Hz_Sort_biO_02052024_1.csv
│      ├──niO         ...
│      ├──riO         └──100Hz_Sort_biO_02052024_(...).csv
│      └──wiO
├──400hz
│      ├──biO
│      ├──niO
│      ├──riO
│      └──wiO
└──800hz
       ├──biO
       ├──niO─────────├── 800Hz_Sort_niO_03052024_1.csv
       ├──riO         ...
       └──wiO         └── 800Hz_Sort_niO_03052024_(...).csv
```
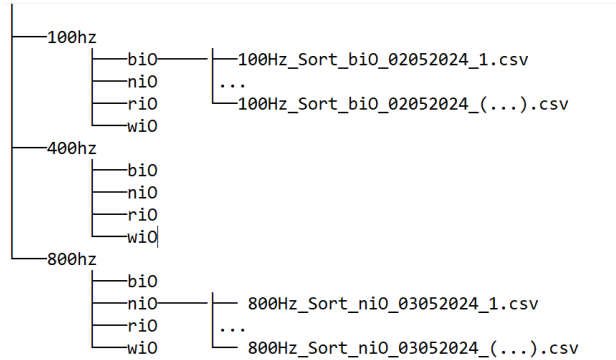
Figure 3: Directory Stucture

- **File Organization:** Separated by Sampling frequency and by Tab types.

2

3.2 Data Columns

- `Datum`: Date of recording (format: MM/DD/YYYY)

- `Uhrzeit`: Timestamp with microsecond precision (format: YYYY-MM-DD HH:MM:SS.microseconds)

- `I_In (A)`: Input current in Amperes

- `V_In (V)`: Input voltage in Volts

- `P_In (W)`: Input power in Watts (calculated as V_In × I_In)

- `I_Out (A)`: Output current in Amperes

- `V_Out (V)`: Output voltage in Volts

- `P_Out (W)`: Output power in Watts (calculated as V_Out × I_Out)

- `Temp (°C)`: System temperature of Arduino Uno in Celsius

- `Energie_In (Wh)`: Cumulative input energy in Watt-hours

- `Energie_Out (Wh)`: Cumulative output energy in Watt-hours
  Calculated as:
  $$\Delta E = P \times \Delta t$$
  $$E_{\text{new}} = E_{\text{previous}} + \Delta E$$

3.3 Measurement Characteristics

- Each file represents one complete sorting operation from start to finish.

- Various sampling frequencies (100hz, 400hz, 800hz)

- Synchronized and Standardized measurements across all 12,000 runs.

- Includes both instantaneous and cumulative measurements

- Timestamps with Microsecond Precision for time-series analysis

- Includes natural variations from repeated operations

4 Potential Research Applications

- **Energy Efficiency Analysis:** Study power consumption patterns during sorting operations.

- **Machine Learning:** Develop models to predict Tab types/colors based on Energy Consumption data.

- **Process Optimization:** Identify inefficiencies in the sorting process.

- **Frequency Impact Analysis:** Examine how the sampling rate affects the performance of the ML models.

- **Anomaly Detection:** Identify unusual power consumption patterns that may indicate system issues.