Sound sensor variability test

2025-04-03

- Two identical sound sensors were used for the test.
- Both sensors were connected to an ADS1115 ADC module.
- A constant sound was played near the sensors for 2 minutes.
- Sensor readings were recorded continuously during this period.
- Each reading was timestamped using the Arduino millis() function.
- Data was saved to an SD card in the format: timestamp, sensor1, sensor2.
- The recorded ADC values were later converted to voltage using the formula: (ADC reading / 32768) * 6.144.
- Readings were filtered to include only data up to 147019 milliseconds.
- Mean and standard deviation were calculated for both sensors.
- The goal was to assess variability and consistency between the two sensors.

```r
knitr::opts_chunk$set(echo = TRUE)
library(readr)
# Read the log file containing time-series sound sensor data.
# Assign column names: timestamp (in milliseconds), sensor1, and sensor2
data <- read_csv("LOG.TXT", col_names = c("timestamp", "sensor1", "sensor2"))
```

```
## Rows: 7420 Columns: 3
## ── Column specification
────────────────────────────────────────────────────────
## Delimiter: ","
## dbl (3): timestamp, sensor1, sensor2
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this
message.
```

```r
# Filter data to include only entries up to 147019 milliseconds
filtered_data <- data[data$timestamp <= 147019, ]

# Define voltage reference based on ADC gain setting (GAIN_TWOTHIRDS =
±6.144V)
VREF <- 6.144

# Convert raw ADC readings to voltages using:
# Voltage = (raw reading / 32768.0) * VREF
filtered_data$Sensor1_Voltage <- (filtered_data$sensor1 / 32768.0) * VREF
filtered_data$Sensor2_Voltage <- (filtered_data$sensor2 / 32768.0) * VREF

# Calculate mean and standard deviation in volts for Sensor 1
```

```r
sensor1_mean <- mean(filtered_data$Sensor1_Voltage, na.rm = TRUE)
sensor1_sd <- sd(filtered_data$Sensor1_Voltage, na.rm = TRUE)

# Calculate mean and standard deviation in volts for Sensor 2
sensor2_mean <- mean(filtered_data$Sensor2_Voltage, na.rm = TRUE)
sensor2_sd <- sd(filtered_data$Sensor2_Voltage, na.rm = TRUE)

# Display summary statistics in volts
cat("Sensor 1 (Volts):\n")
```

```
## Sensor 1 (Volts):
```

```r
cat("  Mean =", round(sensor1_mean, 4), "V\n")
```

```
##   Mean = 1.7448 V
```

```r
cat("  Standard Deviation =", round(sensor1_sd, 4), "V\n\n")
```

```
##   Standard Deviation = 0.0984 V
```

```r
cat("Sensor 2 (Volts):\n")
```

```
## Sensor 2 (Volts):
```

```r
cat("  Mean =", round(sensor2_mean, 4), "V\n")
```

```
##   Mean = 1.7491 V
```

```r
cat("  Standard Deviation =", round(sensor2_sd, 4), "V\n")
```

```
##   Standard Deviation = 0.1121 V
```