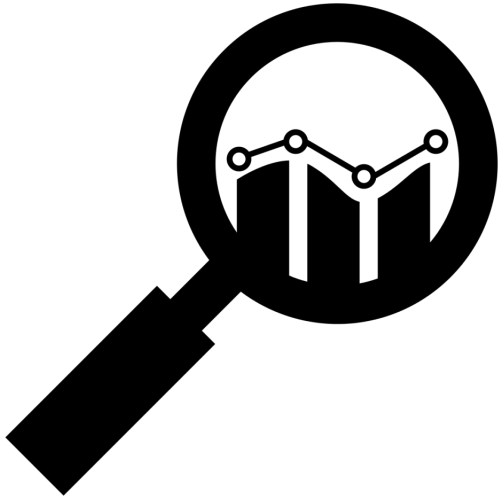
Time series segmentation for

Door movement prediction

Marc Vernet



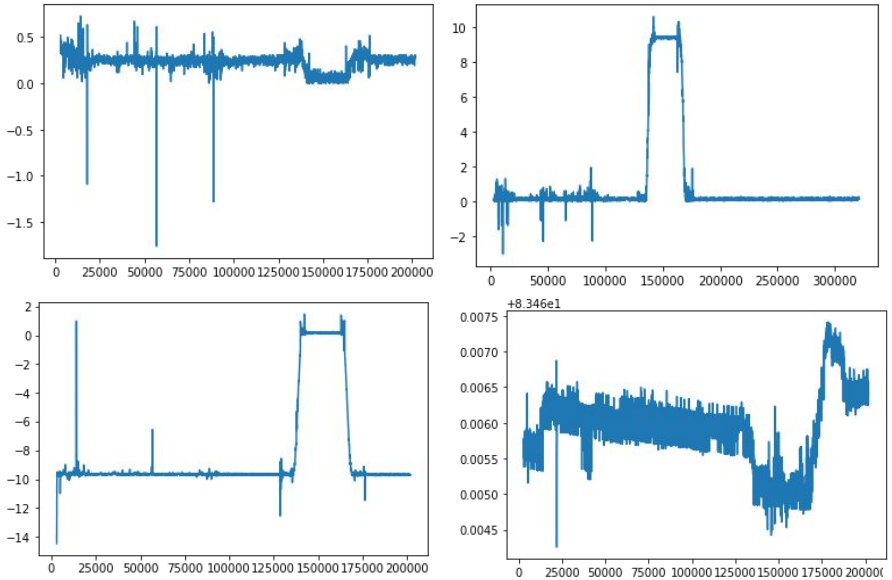
GitHub

https://github.com/marcvernet31/door_opening

TAL TECH

The problem

Example of a door and sensor location



Sample of x-axis, y-axis, z-axis and pressure

INPUT

OUTPUT

-4.651647; 0.478564; 8.231309; 83.471397
-4.613362; 0.344566; 8.269595; 83.471153
-4.651647; 0.401994; 8.307880; 83.471290
-4.632504; 0.421136; 8.403593; 83.471092

- Fully closed door opened fully and closed fully
- Fully closed door opened fully and closed 50%
- Fully closed door opened 50% and closed fully
- 50% opened door opened fully and closed fully
- 50% opened door opened fully and closed 50%

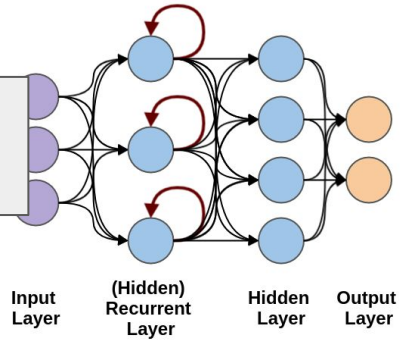
KEY IDEAS

- Data is not labeled
- Only interested in detecting if the door is completely closed or open or at 50%
- Door speed is uniform
- The solution has to work in real time (the algorithm can’t see future samples)

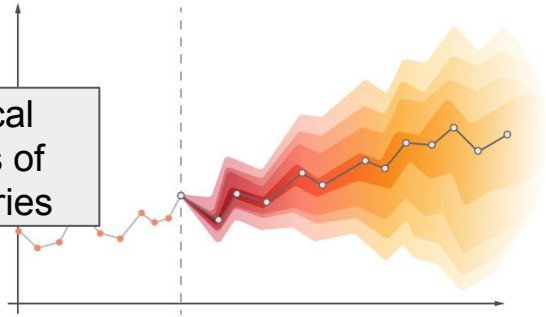
TAL TECH

State of the Art

Recurrent Neural Networks



Statistical Analysis of Time Series

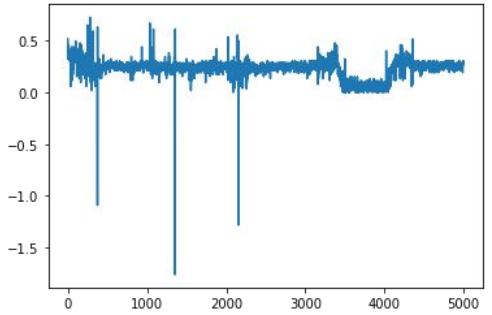


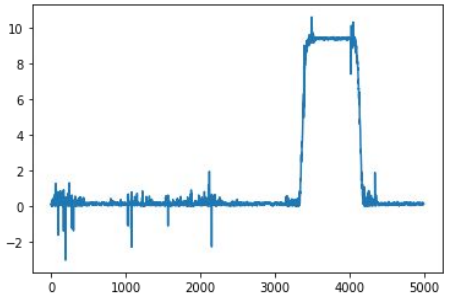
Michèle Basseville and Igor V. Nikiforov (April 1993). *Detection of Abrupt Changes: Theory and Application*. Englewood Cliffs, NJ: Prentice-Hall.

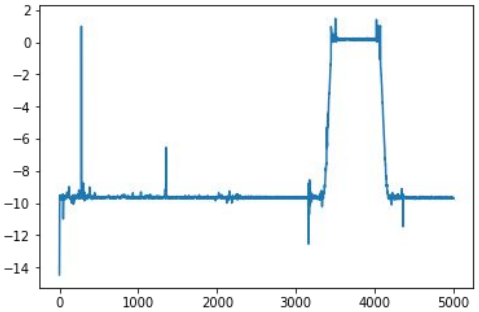
The book can be downloaded here: https://www.researchgate.net/publication/2406812_Detection_of_Abrupt_Change_Theory_and_Application

The idea

- Approach the problem as a time series segmentation problem
- The segmentation can be solved by step detection
- Being able to correctly detect steps allows to predict movement
- Aggregate predictions for each variable in a single response





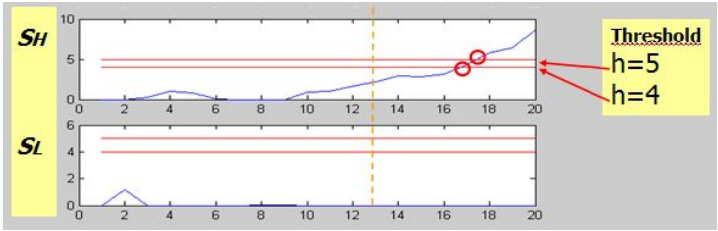


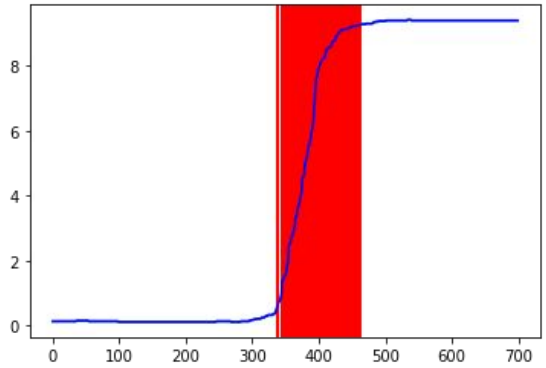
TAL TECH

CUSUM algorithm

- Sequential analysis technique
- Change monitoring
- Depends on two parameters: **drift** and **threshold**

$$S_0 = 0$$
$$S_{n+1} = \max(0, S_n + x_n - \omega_n)$$

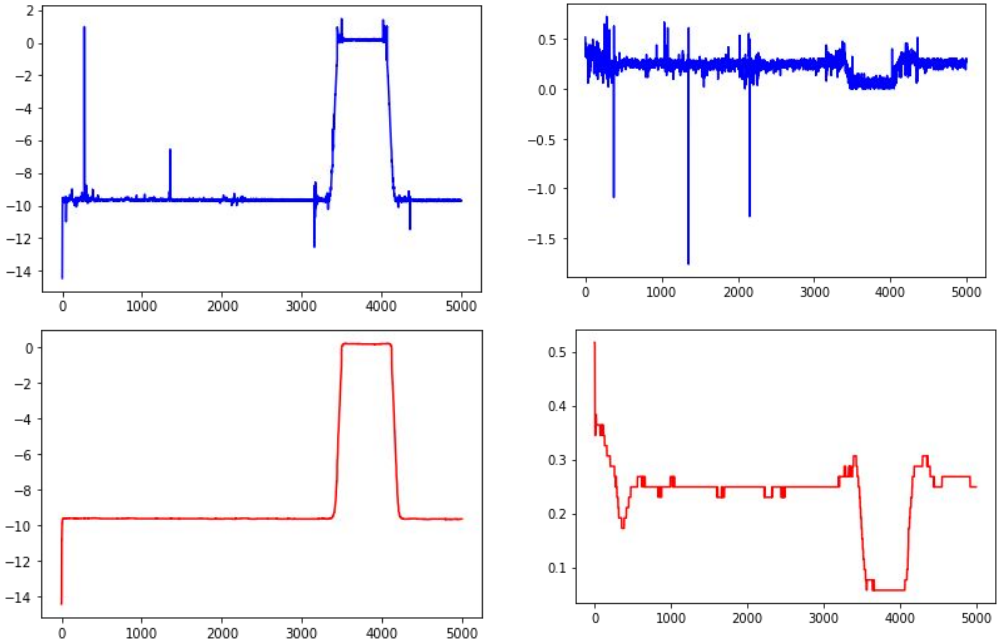




Michèle Basseville and Igor V. Nikiforov (April 1993). *Detection of Abrupt Changes: Theory and Application*. Englewood Cliffs, NJ: Prentice-Hall.

The algorithm

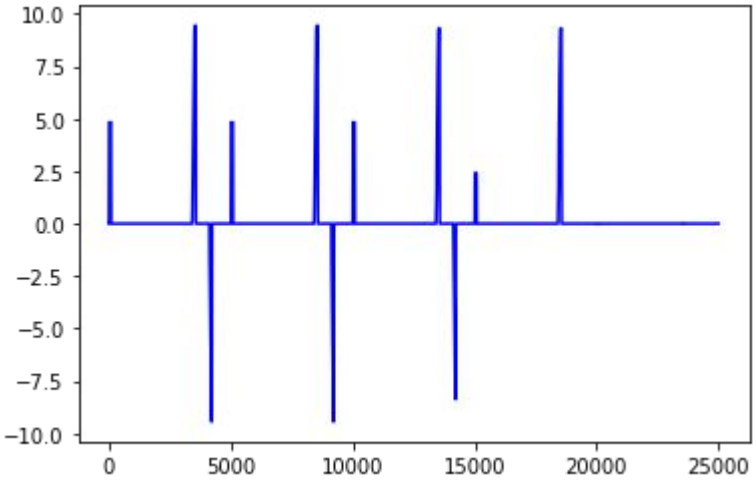
(1) Apply a median filter (perfect for removing noise without damaging the structure)

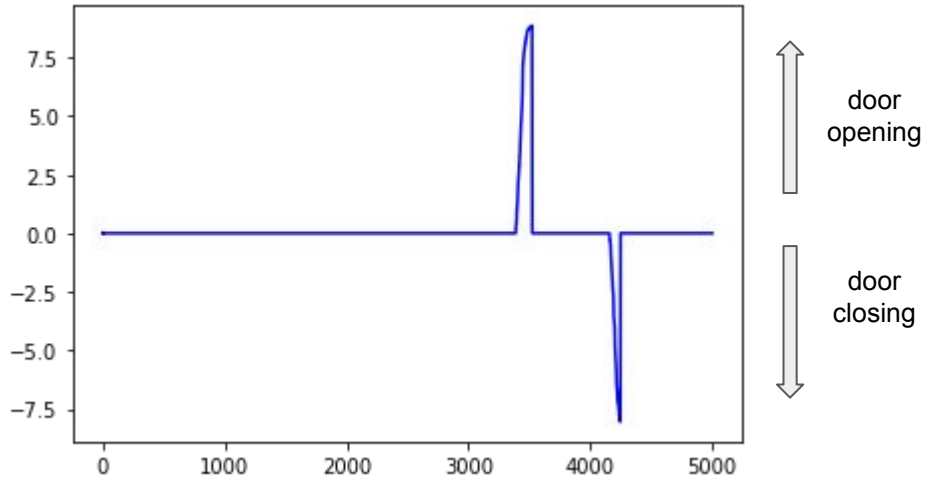


(2) Find CUSUM indicators

(3) Estimate movement from number of consecutive indications

(4) Multiply by slope to considerate speed irregularities

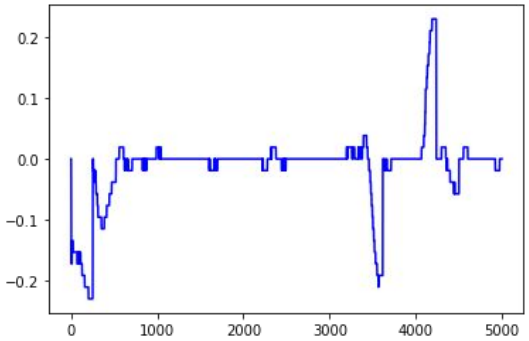


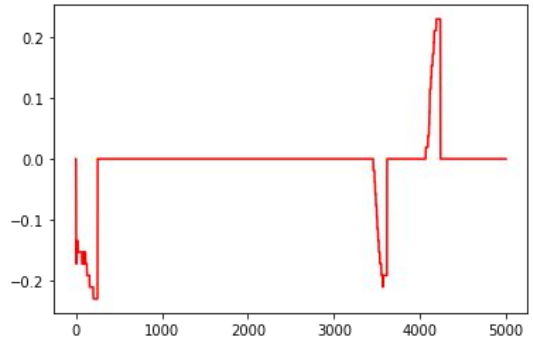


(5) Top cleaning algorithm:

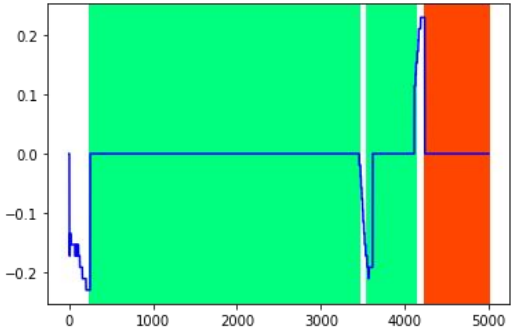
- Keep track of global maximum and minimum and delete smaller that 50% tops
- When a new maximum top is found all the vector has to be iterated.

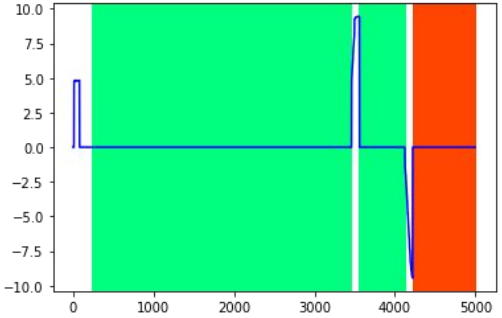
```
for i in val:
    if(localMax):
        isGlobal = change[i-1] > globalMax*coef1
        deleteTop = change[i-1] < globalMax_x*coef2

        # Local max is not global and smaller than 50% global: delete
        if (isGlobal):
            globalMax = change[i-1] # update global max
            deleteSmallTops(...)

        elif(deleteTop):    # Delete local top
            removeMaxTop(...)

    if(localMin):
        isGlobal = change[i-1] < globalMin*coef1
        deleteTop = change[i-1] > globalMin*coef2

        # Local minimum is global
        if (isGlobal):
            globalMin = change[i-1]
            deleteSmallTops(...)

        # Local min is not global and smaller than 50% global: delete
        elif(deleteTop):
            removeMinTop(i, clean_x_change)
```

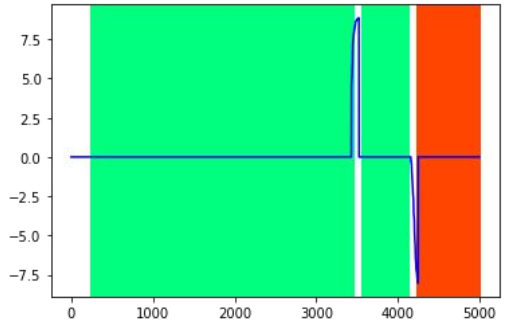


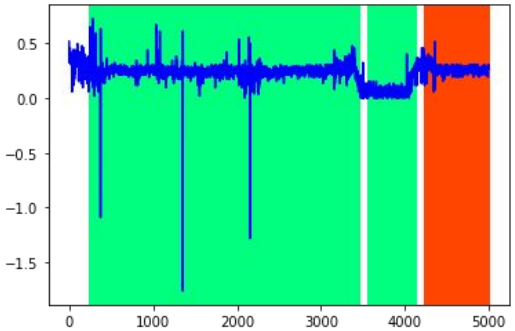


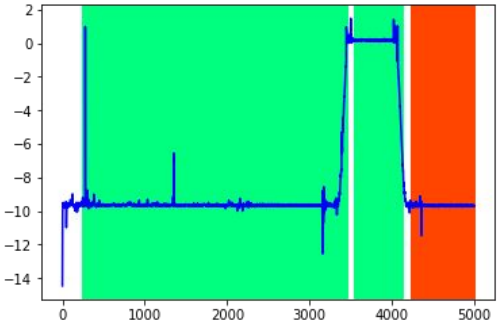
(6) Detect open/close and vote

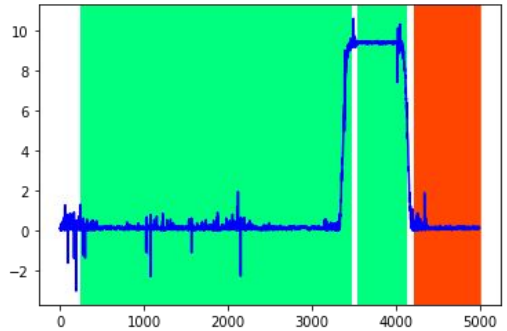












Results

- Correct prediction of all the cycles
- Some classification errors, but it doesnt change the cycles count
- It works in real time, maximum execution time for iteration is 0.043 seconds

Problems

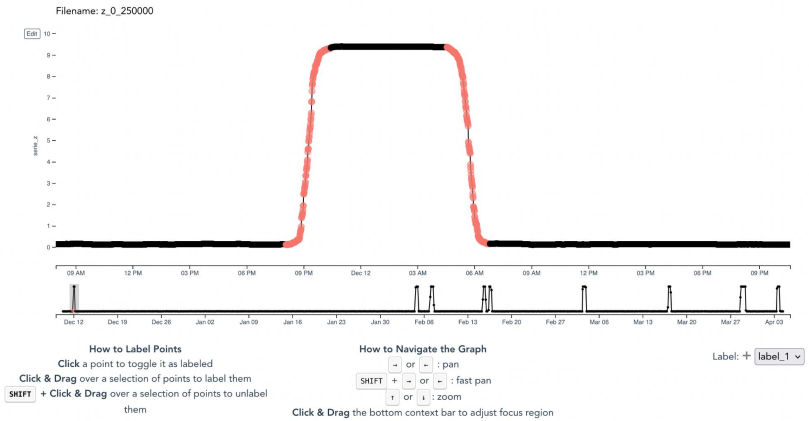
- Good results depends on correctly chosen parameters.
- Parameters depend on the scale of the variable and differ a lot (could need many decimals of precision in some cases)
- Difficult to find a brute force approach to parameter prediction.

TAL TECH

Solution to parameters choice

(1) Manually label a small batch of data to get a baseline

(2) Create a tool to make it easy to choose adequate parameters



TRAINSET

https://trainset.geocene.com/

TAL TECH

Parameter Calculator

Door

Door1

Variable

x

Drift

0.07

Threshold

0.01

Calculate

False alarms: 325 3.43%

Missed Alarms: 3653 38.51%

Correct Alarms: 5834 61.49%

Total Alarms: 9487

Execution Time: 1.2s

False alarms are less than 4%

Correct alarms detected are more than 50%

Plot result

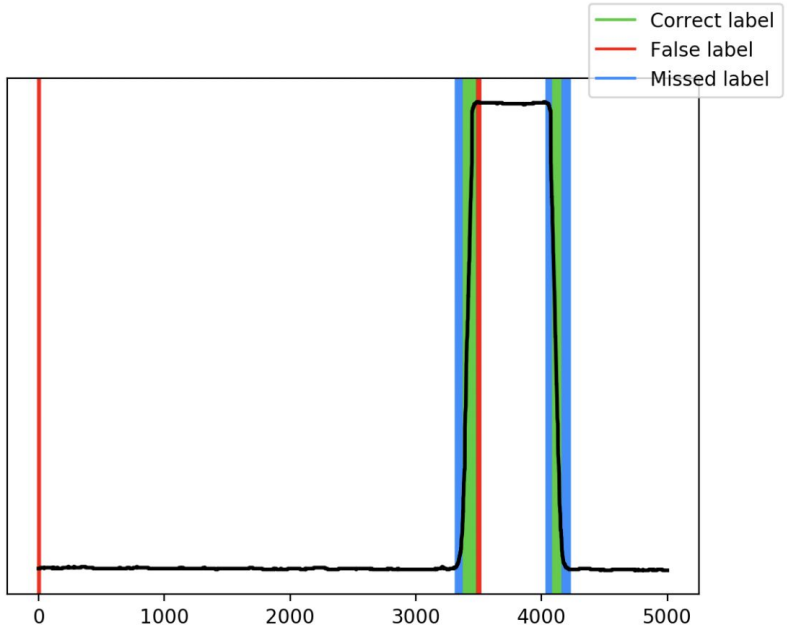
Plot range min

0

Plot range max

5000

Plot



Current status

- Parameters choosing interface
- Real time labeling prediction in Python

```
------
Timestamp:  18784653
x:  -9.806336  | y : 0.248989  | z: 0.095765
closed
------
Timestamp:  18784693
x:  -9.806336  | y : 0.268142  | z: 0.153224
closed
------
Timestamp:  18784733
x:  -9.806336  | y : 0.248989  | z: 0.153224
closed
------
Timestamp:  18784772
x:  -9.806336  | y : 0.19153  | z: 0.153224
closed
------
```

Output of the prediction Python script

To do

- Convert Python code to C to run in STM32

Check out code at Github!
https://github.com/marcvernet31/door_opening

TAL TECH