Gesture Recognition by Pattern Matching using Sensor Fusion on an Internet of Things device

By Sebastien Gios with Peter Van Roy as supervisor 2022-2023

INTRODUCTION

Context

Motivation

Main Technologies

How to achieve it ?

Context

- New technologies
  - More Internet of Things device
    - More Availability & Accessibility
      - Opportunities to help people

Motivation

- **Gesture recognition** algorithm based on **sensor data**
  - A gesture is a movement or rotation of the human arm
- **Communication** between devices
- **Help people** (especially those with reduced mobility)

Main Technologies

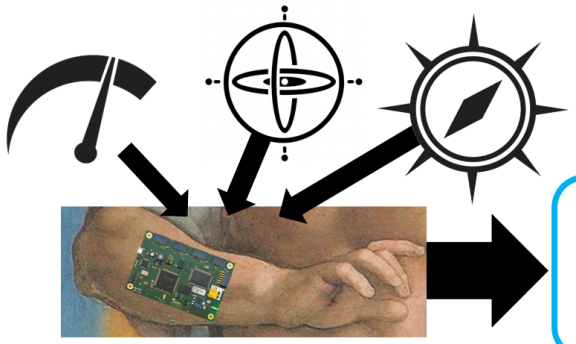
GRiSP device attached to a wrist

Programming Language

Hera

A **Sensor Fusion** By Sébastien Kalbush & Vincent Verpoten

How to achieve it ?



Kalman Filter

Acceleration

Learning

Query

Machine Learning

IoT Interaction

Gesture Recognition

Hardware

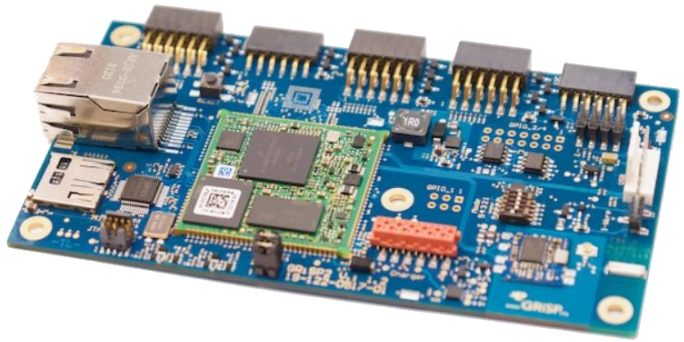
Previous Work

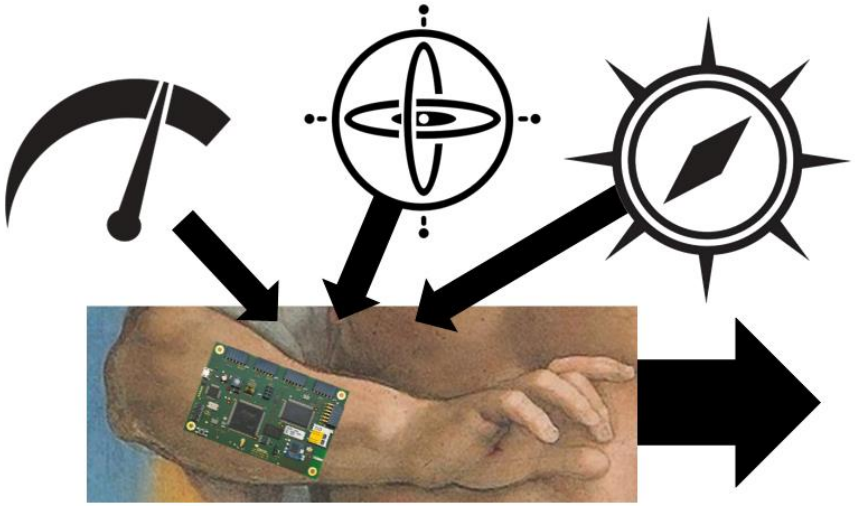
Our Contribution

Future Work

How to achieve it : Sensor Data

Hardware





Accelerometer

+

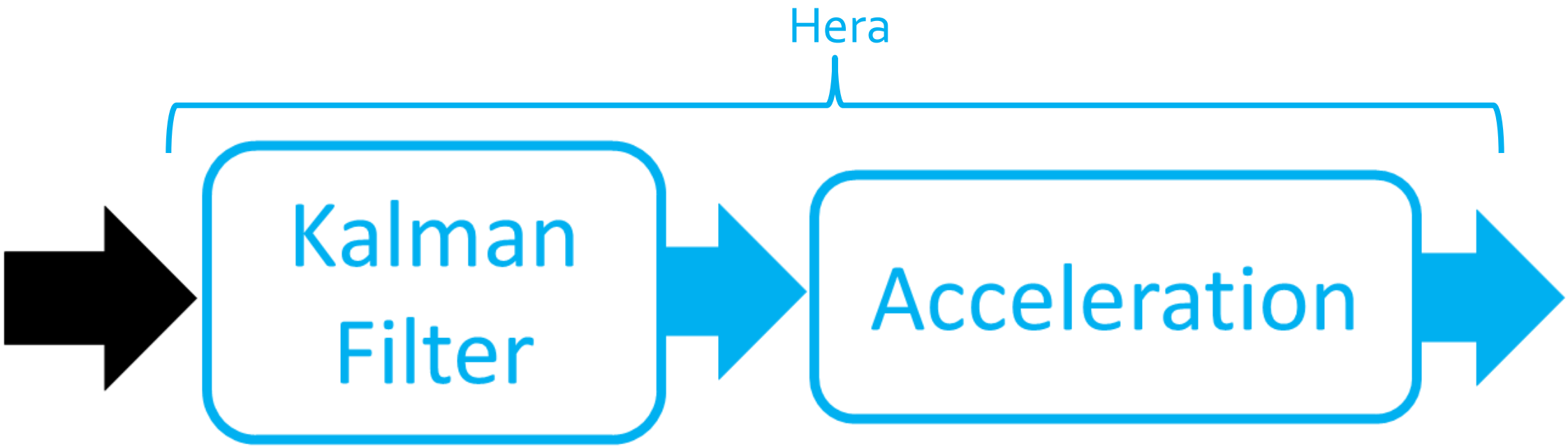
Magnetometer

+

Gyroscope

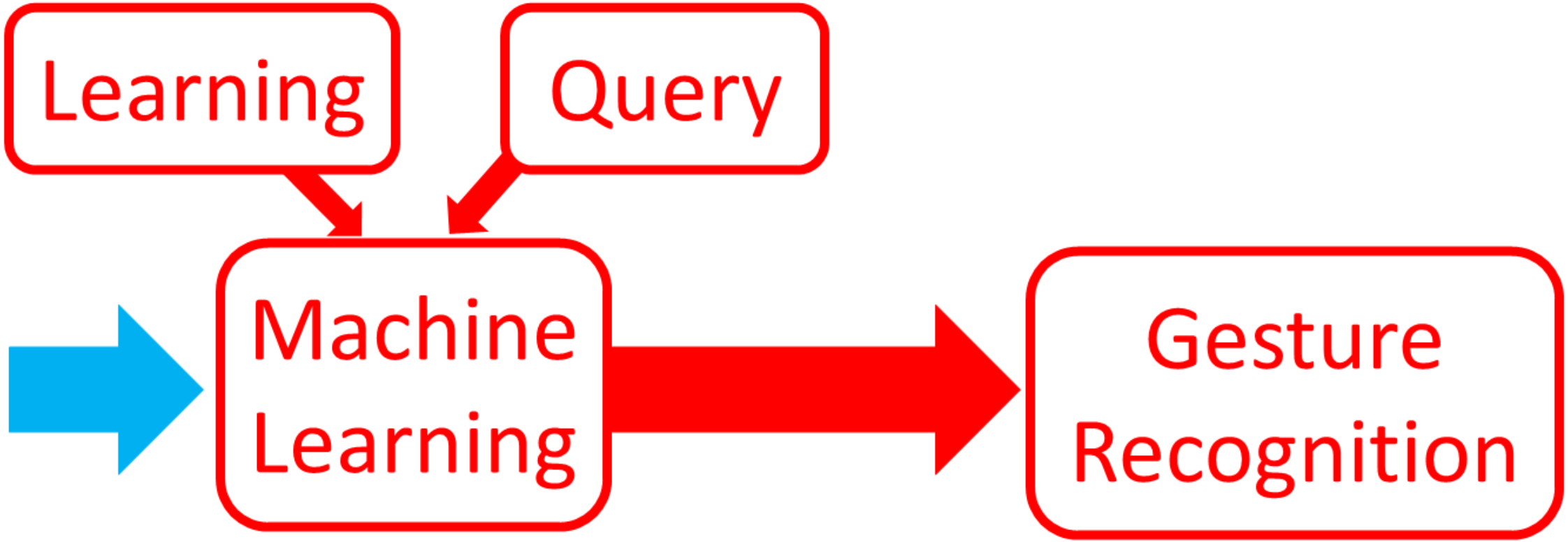
How to achieve it : Sensor Fusion

Previous Work



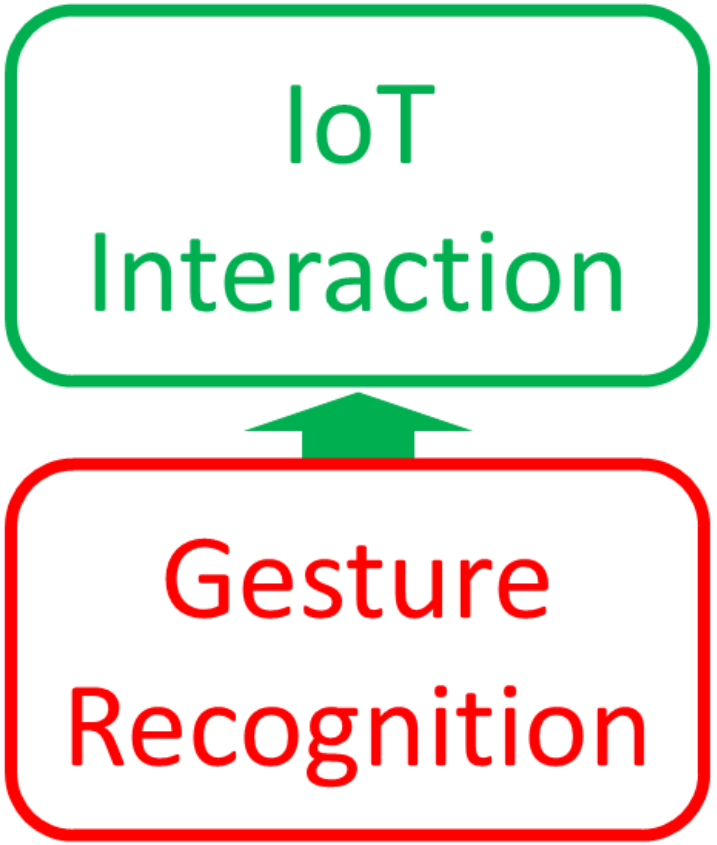
How to achieve it : Classification

Our Contribution



How to achieve it : IoT Interaction

Future Work



CHALLENGES

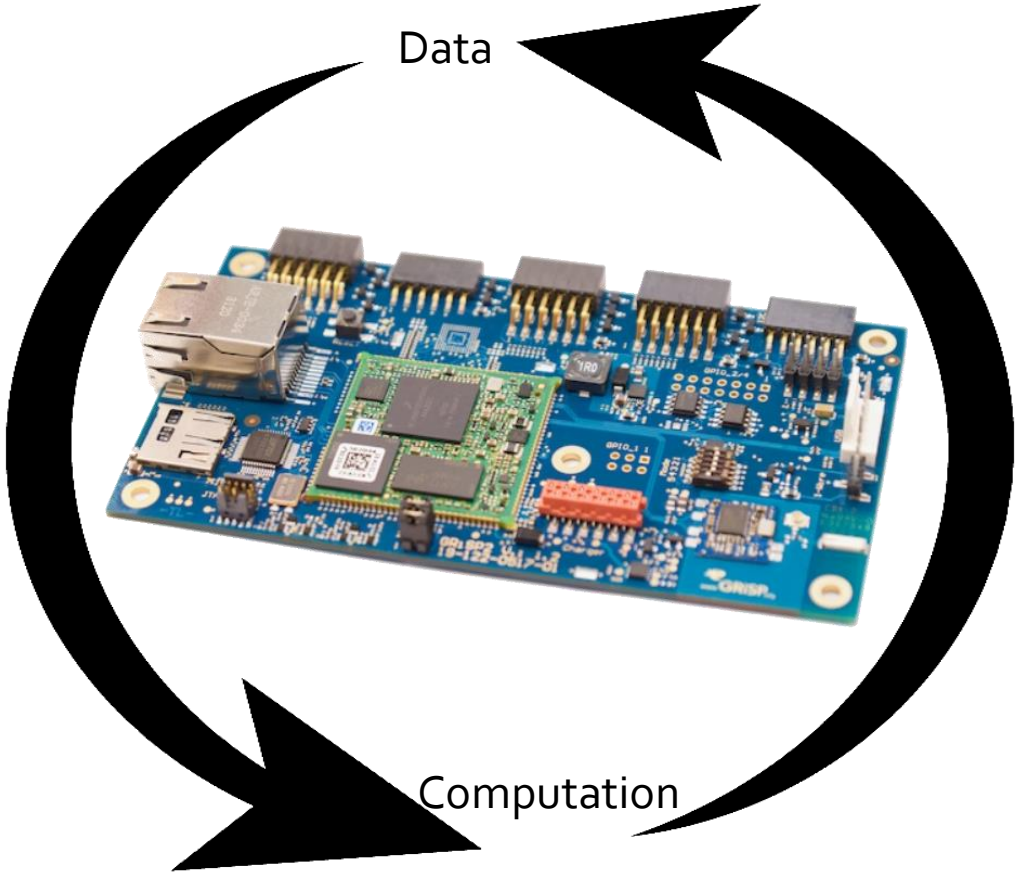
No dependecies

Efficiency

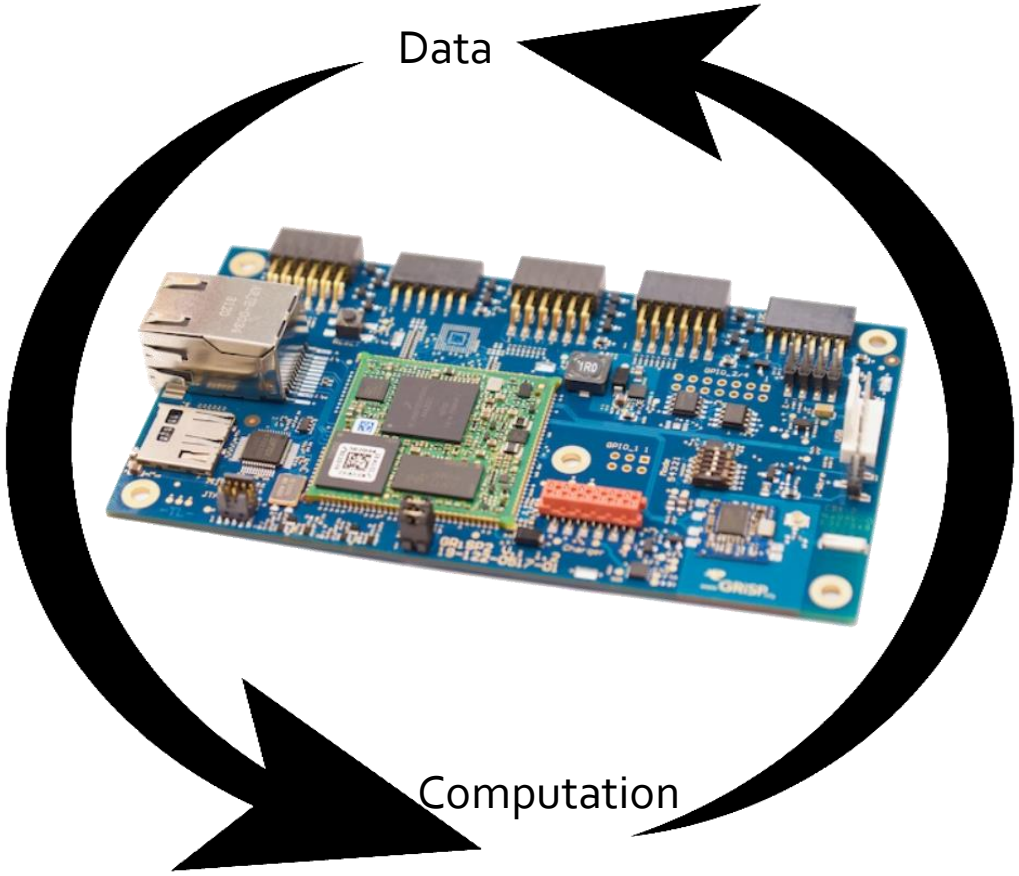
Noise Handling

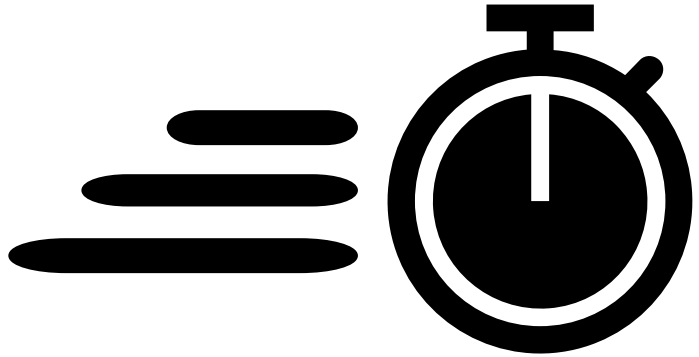
Failure Robustness

Challenges : No dependencies



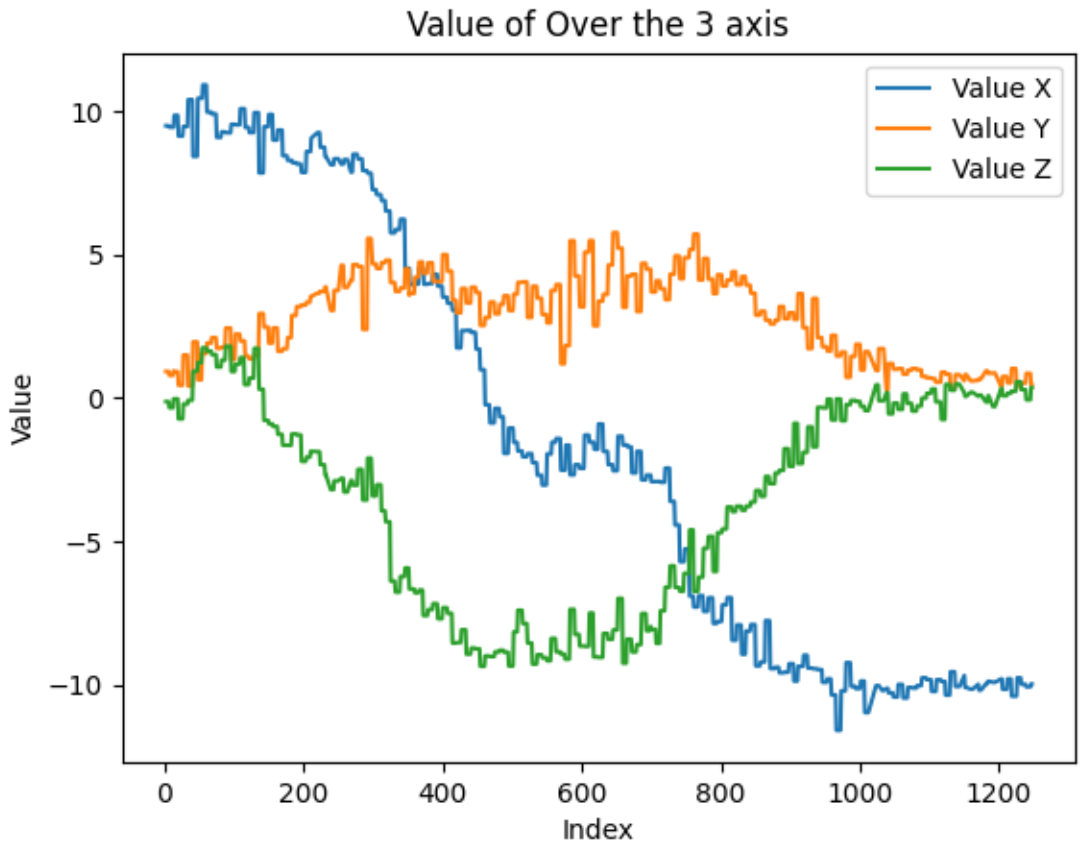
Challenges : Efficiency



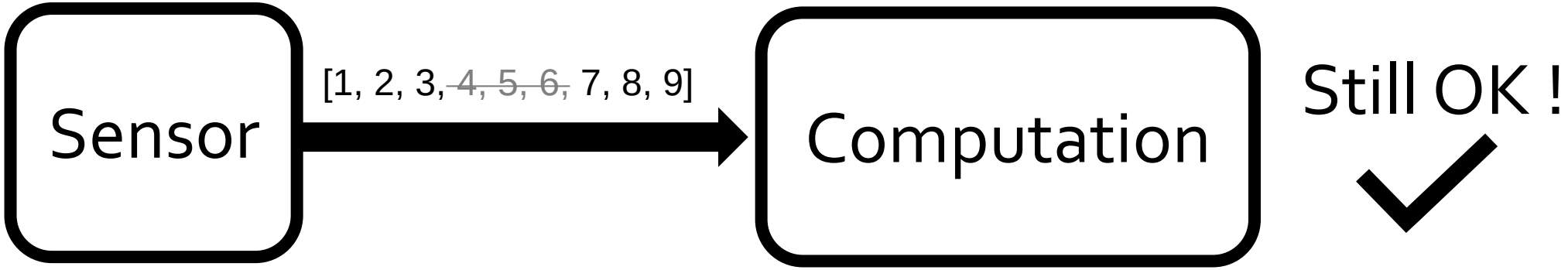


- Adapted to **human environment**
  - Arm movement
- Everyday use
  - Always moving
- **NO heavy computation**
  - Adapted for edge computing

Challenges : Noise Handling



Challenges : Failure Robustness



OUR CONTRIBUTION

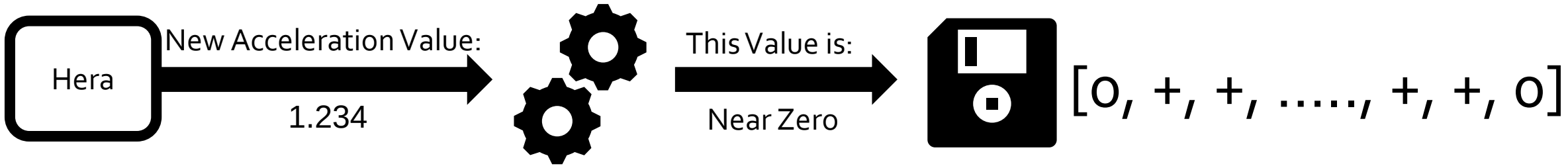
Classification by pattern matching: Learning & Classification

Noise Handling

Realtime Classification

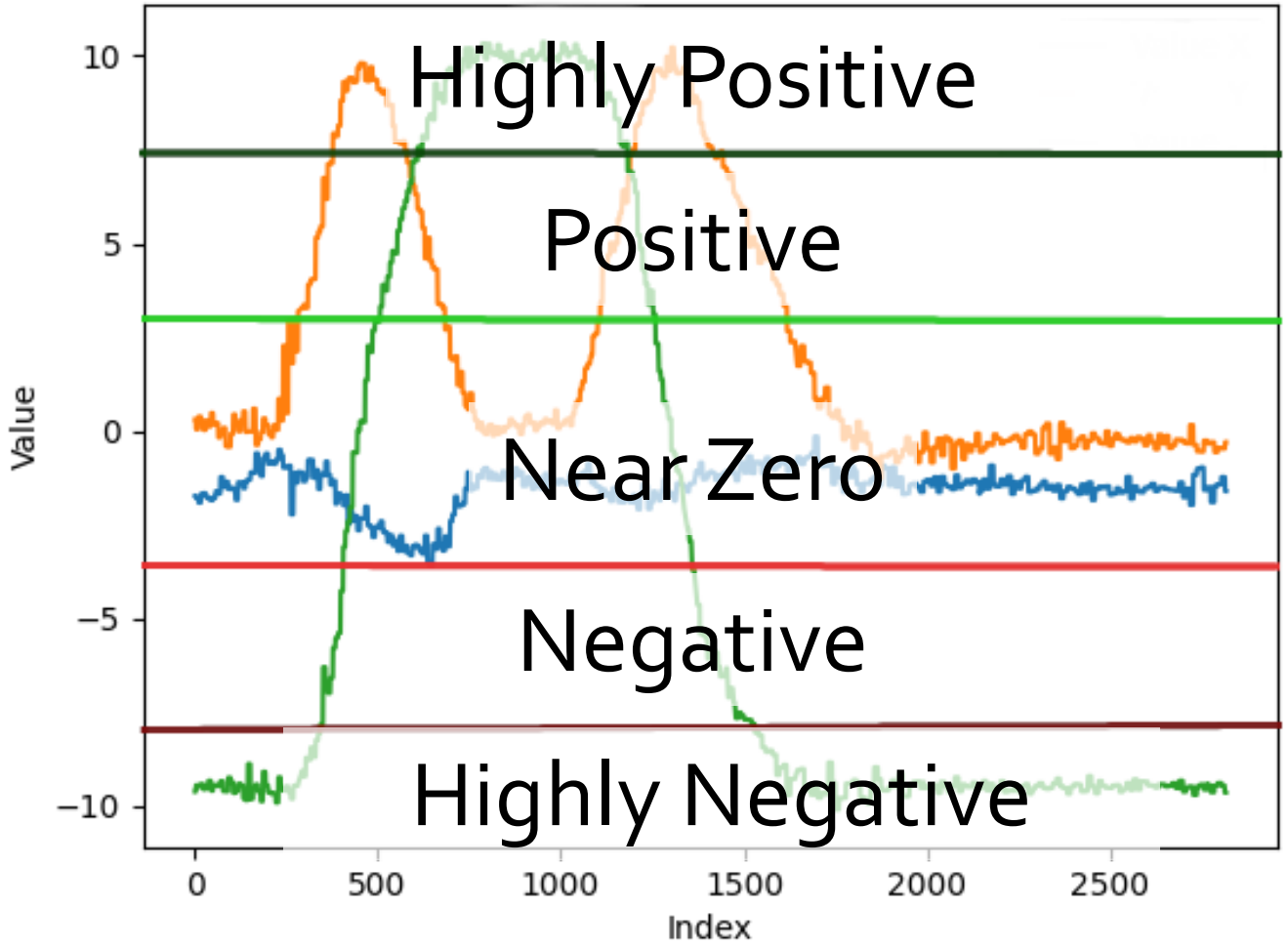
Demonstration

Classification by pattern matching: Learning step 1

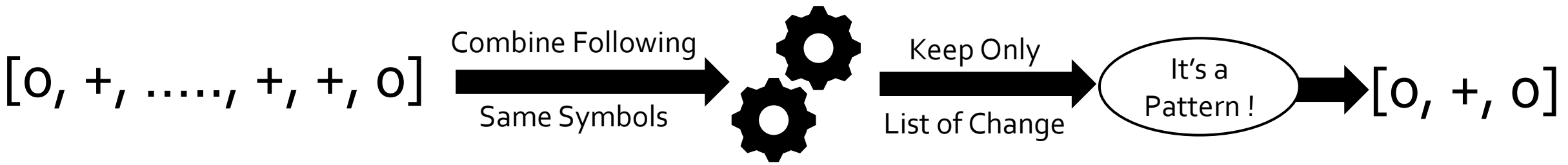


- Data from Hera are **translated** to symbols:
  - Highly Positive, Positive, Near Zero, Negative or Highly Negative
- Saved **in a list**

Symbols

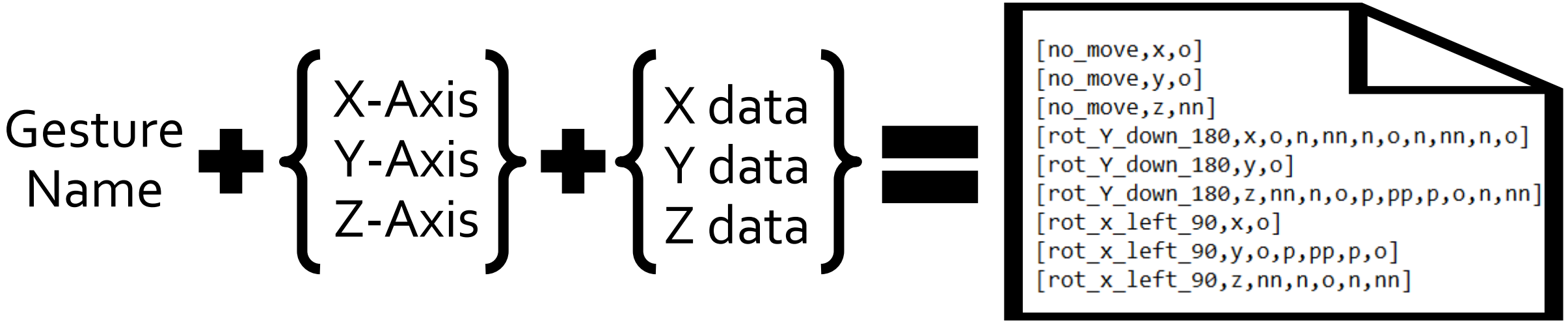

Highly Positive
Positive
Near Zero
Negative
Highly Negative
10
5
0
−5
−10
Value
0
500
1000
1500
2000
2500
Index

Classification by pattern matching: Learning step 2



- To create a pattern :
  1. From a **list of symbols**
  2. Combine following same symbols
  3. Keep only a unique symbols
     - **« Keep only the list of change »**

Classification by pattern matching: Learning step 3



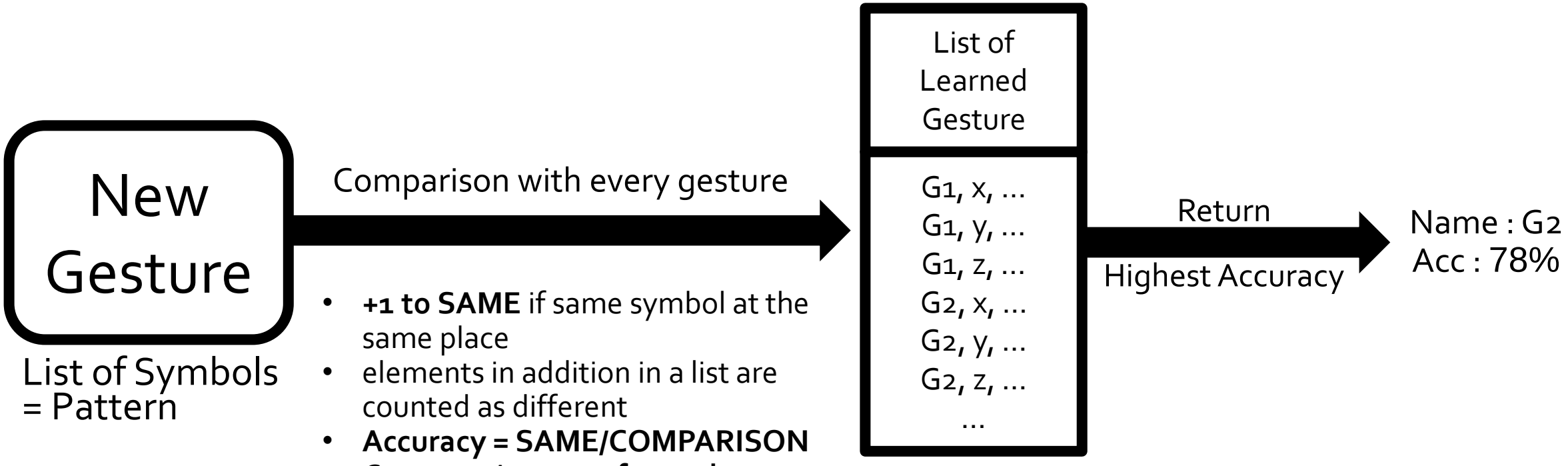
- Name of the gesture + Pattern over the 3 axes
- Saved in a file to be reused

Noise Handling

- **Bad** if [o, o, -, o, o, o] when not moving

- **Regroup** by sub-list: keep most frequent symbol
- [(o, o, -,) (o, o, o)] → [o, o] : No noise

Classification step



- **+1 to SAME** if same symbol at the same place
- elements in addition in a list are counted as different
- **Accuracy = SAME/COMPARISON**
- Compute Accuracy **for each learned gesture**

Realtime classification

One Time

INPUT : Time for the gesture

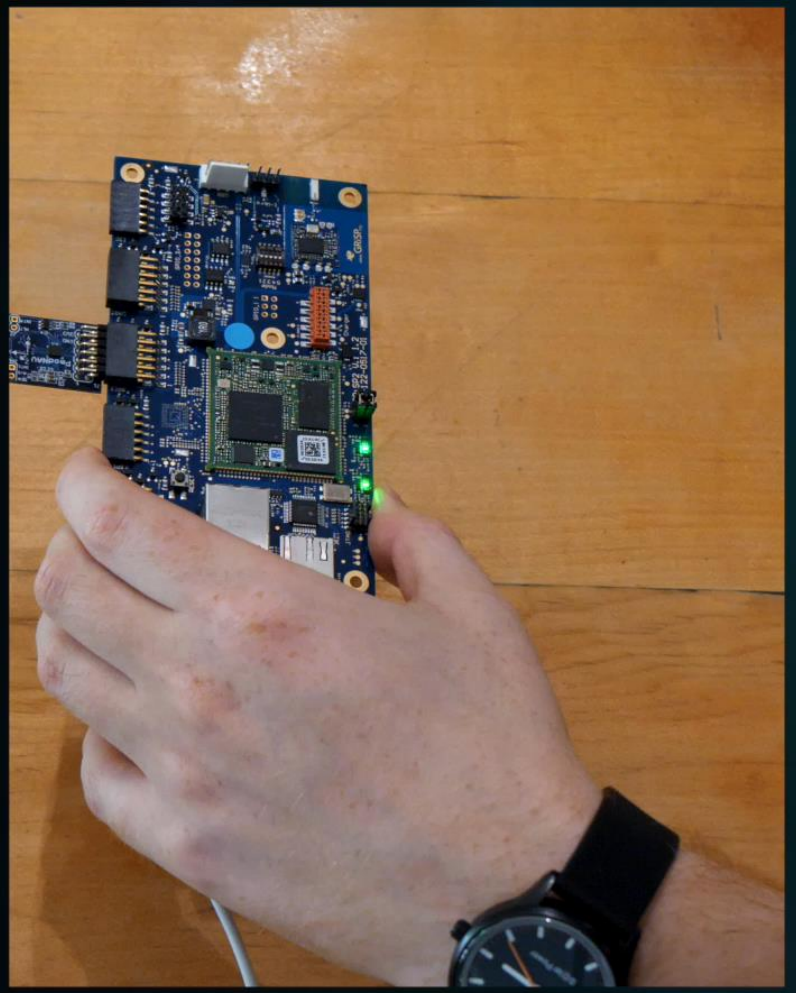
1. After a countdown
2. Perform the Gesture
3. Classification
4. Most accurate gesture
5. Possibility to learn

Multiple Time

INPUT : Time to stop between gesture,
Max time of the classification

1. Perform the Gesture
2. Stop moving
3. Classification
4. Most accurate gesture

Timeout



```
X : 0.0, Y : 0.16666666666666666, Z : 0.0
N : rot_x_right_360
X : 0.0, Y : 0.1111111111111111, Z : 0.0
N : rot_x_right_360
X : 0.0, Y : 0.1111111111111111, Z : 0.0
Too low Accuracy, No gesture recognized
Do you want to learn this gesture? (y/n/ENTER) : n
n
N: learn ok
(sensor_fusion@nav_1)8> sensor_fusion:stop_all().
sensor_fusion:stop_all().
=INFO REPORT==== 1-Jan-1988::00:03:45.961362 ===
    application: hera
    exited: stopped
    type: temporary

Connection established!
ok
(sensor_fusion@nav_1)9> sensor_fusion:launch().
sensor_fusion:launch().
ok
(sensor_fusion@nav_1)10> sensor_fusion:realtime_once(20).
sensor_fusion:realtime_once(20).
```

https://youtu.be/jTJtuDh_4gY

CONCLUSION

Verdict

Future Work

Verdict

- Gesture recognition algorithm **in realtime**
- User-friendly, efficient
- **No cloud** dependency
- Data from 3-axes **acceleration**
  - **Excellent** for rotation

- **Computer** dependency to see the result
- Data from 3-axes acceleration
  - **Not efficient** for linear movement
- Must **stop the movement** to end a gesture

Futur Work

- **Continuous** gesture recognition
- Gesture recognition from the **velocity**
- **Communication** between devices
- **Merging** with the optimized matrix libraries

QUESTIONS & ANSWERS