# Tactile sensing method development for a mobile multi-legged robot in a cave environment

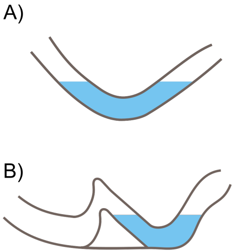
Student: Oleg Bulichev

Supervisor: Alexander Maloletov

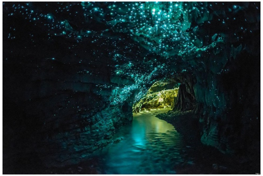
Innopolis University

# Motivation: why do we need to explore caves by robots

*Cave obstacles*




Syphon



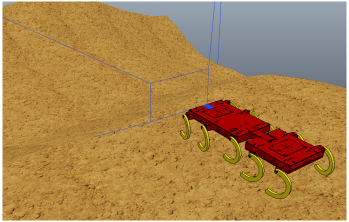
Splash

Glacial caves

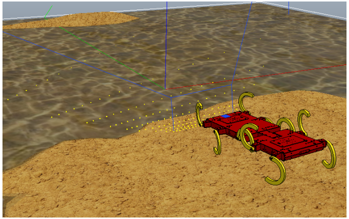
Clay

# Motivation: unsolvable problem for cameras and lidars

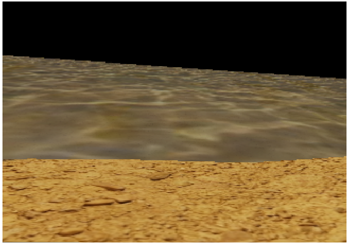
*Question: how to make a terrain map when you have a spash above?*



Terrain without water



Terrain with water, lidar data



Camera view

# Problem statement

**Problem 1**

How to obtain a useful information about terrain, **when we have a SLAM based on lidars, cameras**?

**Problem 2**

How to obtain a useful information about terrain, **when main navigation system died**?

# Problem statement

### Problem 1

How to obtain a useful information about terrain, **when we have a SLAM based on lidars, cameras**?

Obtain map and type of terrain

### Problem 2

How to obtain a useful information about terrain, **when main navigation system died**?

Problem 1 + localization

# Proposed solutions

## Problem 1

*Map can be built* **using tactile sensors** on each leg of the robot and create a **dense point cloud** using sampling from generated mesh using **modified Delaunay triangulation**.

*Terrain type* can be obtained solving **terrain classification** problem using **machine learning**.

## Problem 2

*Localization problem* can be solved by **fused data** from net of **beacons** plus **several IMU** on board and knowing a **kinematics** of the system.

# Literature review

I searched about:

- Cave environment: obstacles, dimensions.
- Robots for cave exploration. From zeppelins, to quadruped robots.
- Methods for map creation using classical way and haptic. Based on cameras, lidars, tactile sensors and etc.

**Summary**

I haven't found any paper, who tried to create a map using tactile sensors in a cave.

# Robot design

*Requirements*

**Problem** – choose robot mover type. This robot should:

- should have *small dimensions* to sneak through holes;
- have enough *mobility* to pass a granular terrain;
- should *pass a small water obstacles*;
- can *climb on a big stones*;

# Robot design

*Requirements*

**Problem** – choose robot mover type. This robot should:

- should have *small dimensions* to sneak through holes;
- have enough *mobility* to pass a granular terrain;
- should *pass a small water obstacles*;
- can *climb on a big stones*;

1-DoF multilegged robot was chosen

# Robot design

*Structural synthesis problem*

**Question**

What the optimal amount of legs should be in such robot?

# Robot design

*Structural synthesis problem*

**Question**

What the optimal amount of legs should be in such robot?

**Answer**

Robot should have **10-12 legs** in total!

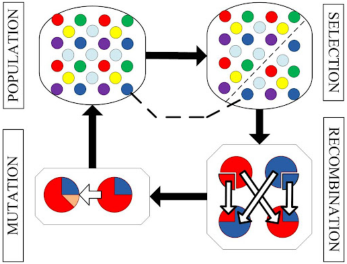
# Robot design

*Technological stack*

Generating terrain approach

GAZEBO

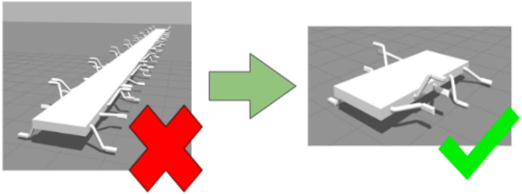
Robot simulator




Genetic algorithm (DEAP Library)

# Robot design

*Proposed solution*



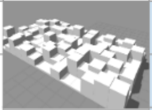
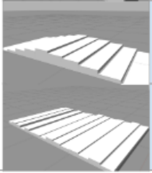
**Idea**: Minimize number of legs without losing cross-road passability

$$F = \beta\left(\omega_1 \cdot \overbrace{\delta}^{\text{Distance}} + \omega_2 \cdot \overbrace{\frac{1}{(\gamma - 1)sin(x)}}^{\text{Body length}}\right) + \\ + (1-\beta)\delta^{\omega_1}\left(\frac{1}{(\gamma - 1)sin(x)}\right)^{\omega_2}$$

$\beta$ is adaptive parameter, $\omega_{1,2}$ are the weight coefficients.

# Robot design

*Results*

| | Terrain types | Number of legs <u>per side</u> | Angle btw neighbor legs | Wave offset btw sides | Number of individuals |
|---|---|---|---|---|---|
| **Phase1** |  | 6 | 73 | 163 | 200 |
| **Phase2** | | 6 | 72 | 165 | 55 |
| |  | 5 | 68 | 177 | |
| | | 6 | 77 | 167 | |

**Result:** robot should have 10-12 legs in total

# Robot design

*Increasing maneuverablity*

**Question**

Long robot can stuck in a cave hole, while rotating. How to avoid it?

# Robot design

*Increasing maneuverablity*

## Question

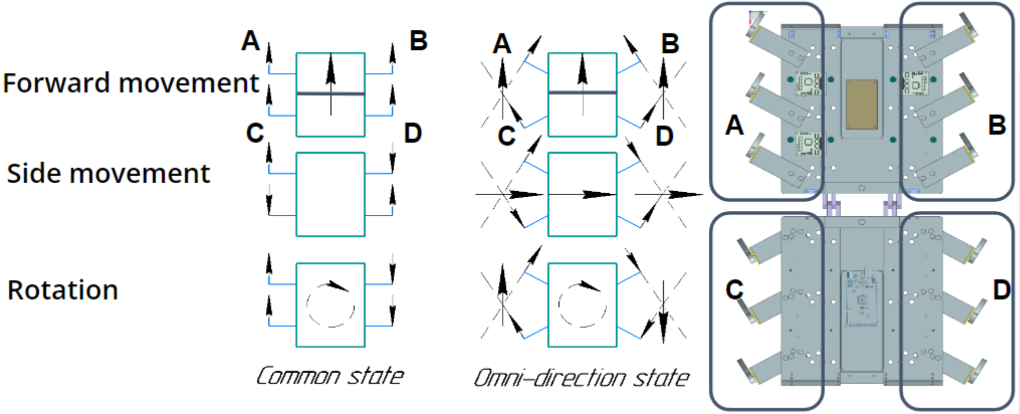
Long robot can stuck in a cave hole, while rotating. How to avoid it?

## Answer

Add an ability to sidestep without changing orientation

# Robot design

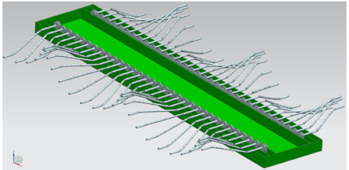
*Proposed solution*



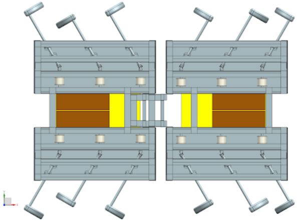

Vector representation of forces in the conventional and omni-direction states

# Robot design

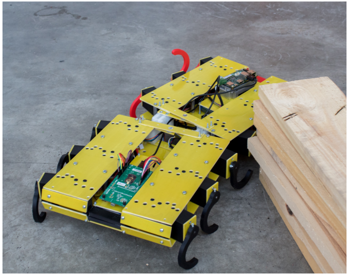
*StriRus prototypes (1)*



1st gen, 54 legs, no segments



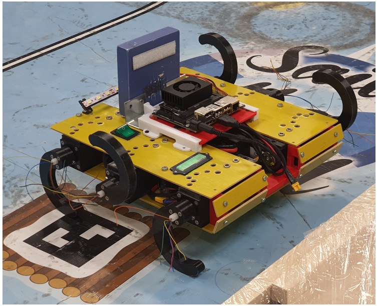
2nd gen, 12 legs, 1 DoF segment, continuous leg inclination up to 45 deg



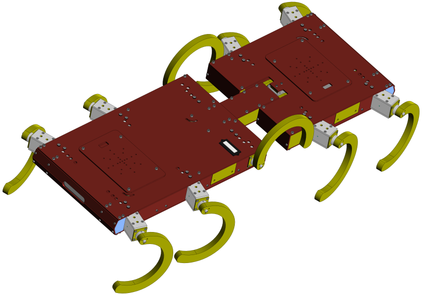
3rd gen, 12 legs, 2 DoF segments, discrete leg inclination up to 45 deg

# Robot design

*StriRus prototypes (2)*



4th gen, 6 big legs, no segments



5th gen, 10 biggest legs, 1 DoF segment, discrete leg inclination up to 15 deg

# Force tranducer design

**Question**

How to receive a reaction force from the ground?

# Force tranducer design

**Question**

How to receive a reaction force from the ground?

**Answer**

Installing a force sensor on each leg

# Force tranducer design

*Force sensor types*

- **F/T sensors**: too big and expencive for small robots.
- **Optical**: too thick.
- **Magnetic**: too thick.
- **Capacitive**: expencive, but the best in terms of requirements.
- **Piezoresistive sensors based on conductive inks or polymers**: inexpencive and robust, but has problems with hysteresis
- **Stain gages**: challenge for wiring between continuously rotating legs and the robot body.

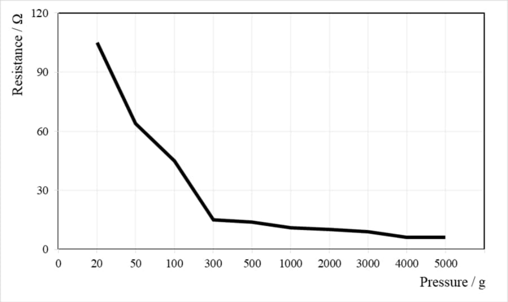
My choice is **Resistive sensor based on Velostat**

It's cheap and it can be made by hand.

# Force tranducer design

*Velostat: Faced problems*

- Hysteresis
- Nonlinearity
- Different values with the same pressure if the square of load is less then the sensor




**Scientific Problem Statement**

To characterize Velostat material for cases when point load is less than sensor size and propose solutions for avoiding such issues.

# Force tranducer design

*Experimental setup requirements*

- Force control.
- Position and force repeatability.
- To have an ability to apply force only on a part of a sensor.
- To change object of the experiment quickly.

# Force tranducer design

*Experimental setup requirements*

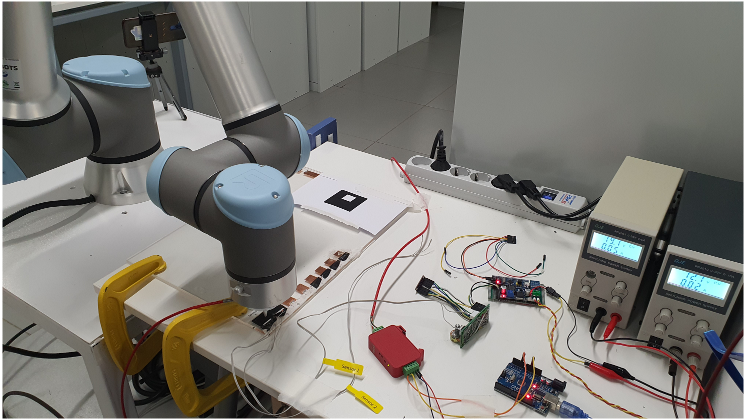
- Force control.
- Position and force repeatability.
- To have an ability to apply force only on a part of a sensor.
- To change object of the experiment quickly.

All requirements are fulfilled

# Force tranducer design

*Experimental Setup: Hardware overall*

1. UR5
2. Futek force sensor
3. Arduino
4. Self-made PCB
5. Velostat sensors
6. Aruco Markers
7. Smartphone camera

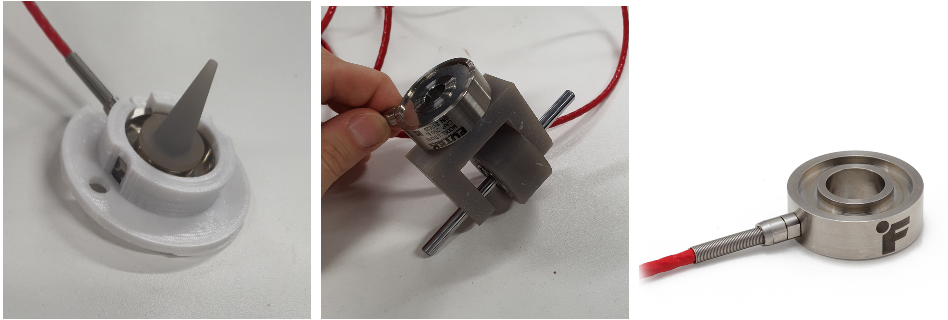


# Force tranducer design

*Experimental Setup: Hardware, aruco markers*

# Force tranducer design

*Experimental Setup: Hardware, end-effector*



(a) Point load (b) Rolling load (c) Futek LTH350

# Force tranducer design

*Experimental Setup: Software*

- **Language**: Python3
- **Frequency**: 450*hz*, using RTDE protocol
- **Control**: Impedance control

# Force tranducer design

*Experiments*

1. Static experiment. The goal is to identify the coefficients for the transducer model.
2. Dynamic experiment. We are representing a transducer as a $4 \times 4$ matrix. We touch with the same pressure using five different end-effectors (area starting from 2mm till 15mm).
3. As the previous, but we are using 1st and last end-effector. We touch with different forces (5, 10, 20, 30, 40 H).

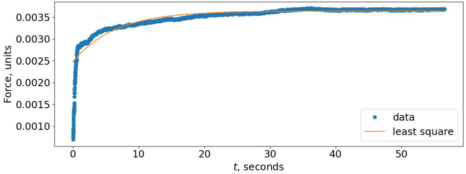
# Force tranducer design

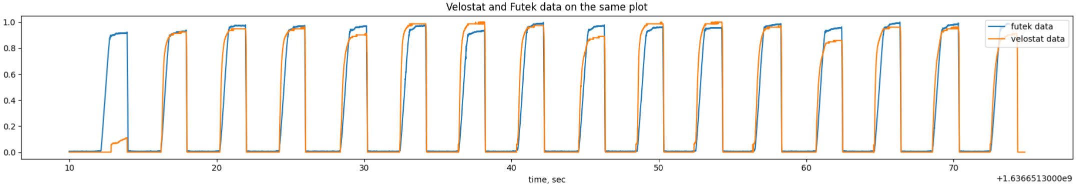
*Result: static experiment*

$$V_{out} = V_0 + p[k_p + k_e(1 - e^{\frac{-(t-t_0)}{\tau_{res}}})](1 - e^{-\frac{A}{p}})$$

$$k_p = A_1 e^{-A_2 p};\ \tau_{res} = B_0 + B_1 e^{-\frac{p}{B_2}}$$

Where $V_0$ – initial voltage, $p,\ A_i,\ B_i,\ \tau_{res},\ k_i$ are constants, $t$ – current time, $t_0$ – the time when the pressure appeared.

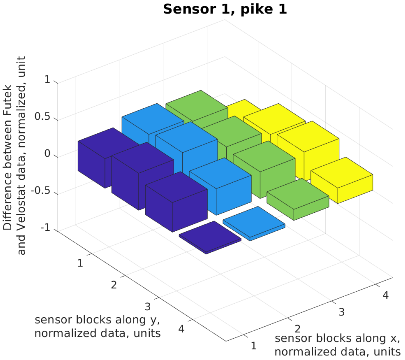


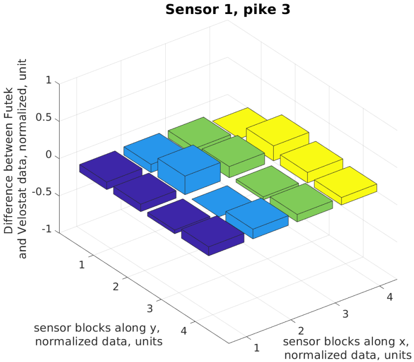

Normalized force data from sensor and transducer

# Force tranducer design

*Result: dynamic experiment*




2mm end-effector diam




8mm end-effector diam

# Force tranducer design

*Results*

1. Static experiment: transducers coefficients were identified.
2. Dynamic experiment. The sensor can be represented as one body, when pressure area is higher than 50 % of the sensor area.

# Terrain classification

**Question**

How to define the terrain type during the movement on such terrain?

# Terrain classification

**Question**

How to define the terrain type during the movement on such terrain?

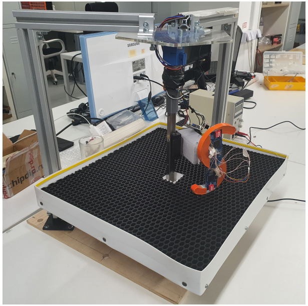
**Answer**

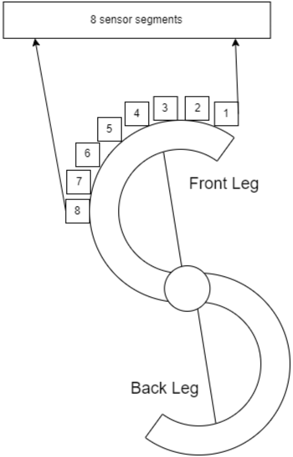
Solving terrain classification problem using Machine learning

# Terrain classification

*Experimental Setup*

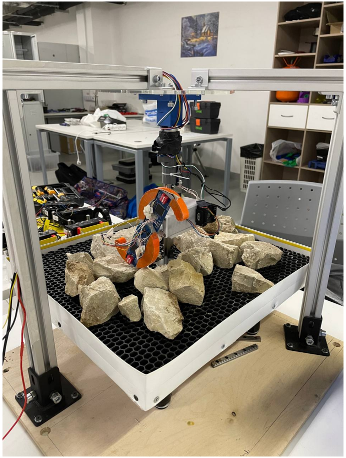
1. Dynamixel MX28 – 47 rev/min
2. Velostat transducer – 25 HZ freq.
3. Experiment duration – 120 sec





# Terrain classification

*Experimental setup: rock terrain*

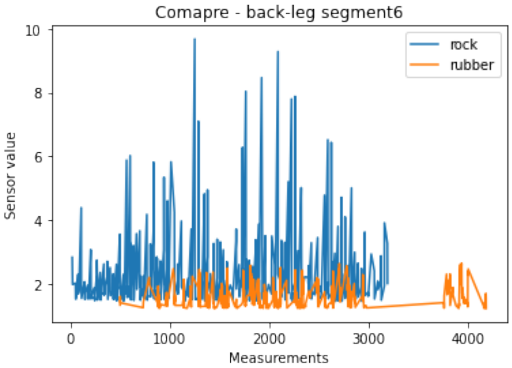


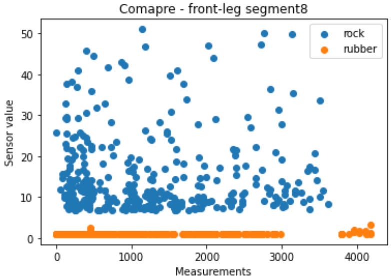
# Terrain classification

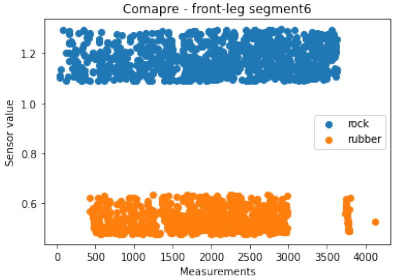
*How it works*

# Terrain classification

*Obtained data from one experiment*



# Terrain classification

*Results*

- Can distinct rubber and rock terrains
- Choose terrain classification parameters
  - RPM
  - Motor Torque
  - Acceleration from IMU
  - Force data which are represented as Sensor valuesegment, Peak amplitude, Average amplitude
- Prove that force transducer is working

# Map creation based on tactile data

**Question**

How to create a dense point cloud using sparse data from legs?

# Map creation based on tactile data

**Question**

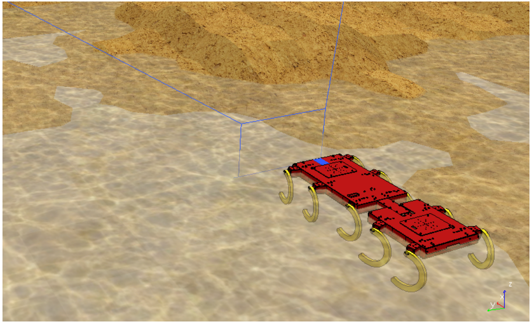
How to create a dense point cloud using sparse data from legs?

**Answer**

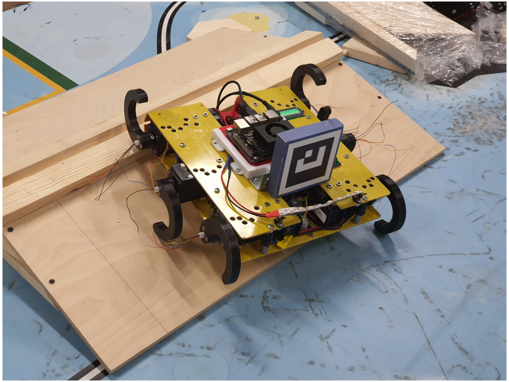
Take this data, create a mesh using concave hull Delaunay triangulation, sampling it and return to the main algorithm

# Map creation based on tactile data

*Experimental setup*



CoppeliaSim simulator, 5th StriRus iteration



4th StriRus iteration

# Map creation based on tactile data

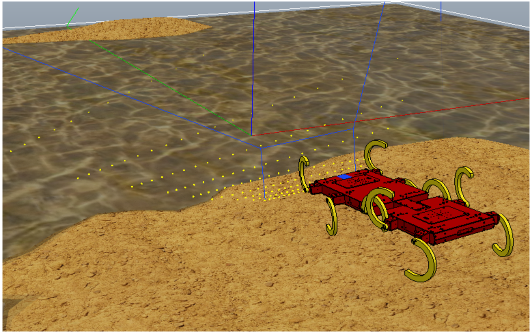
*Assumptions*

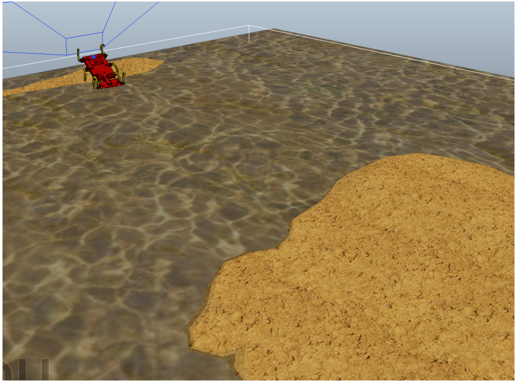
Current solution considering such assumptions:

- Our terrain can be represented $z = f(x, y)$. We can use 2D Delaunay triangulation (projected points on a plane)
- All simulation data are preprocessed by white noise

# Map creation based on tactile data

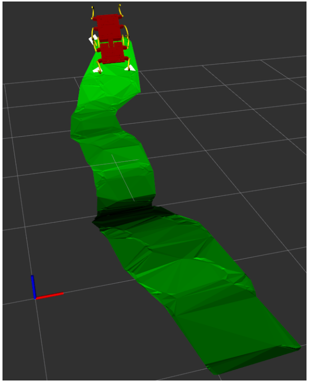
*Result: simulator*



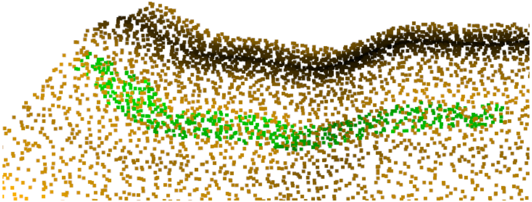


# Map creation based on tactile data

*Result: Mesh*



Mesh created using concave hull 2D Delaunay triangulation



Sampled point cloud

# Map creation based on tactile data

*Why do we need concave hull*

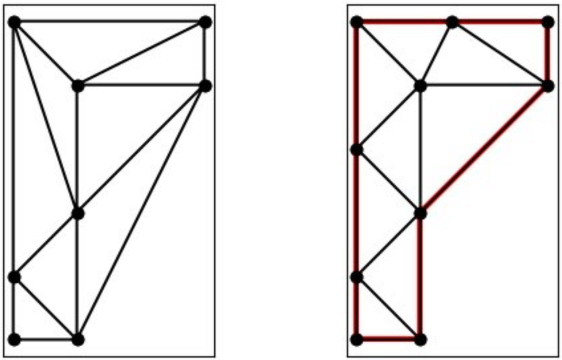
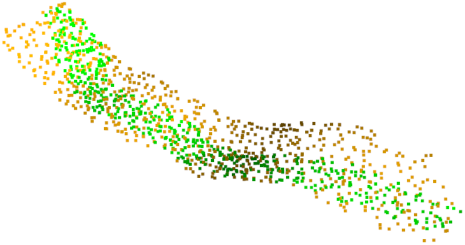
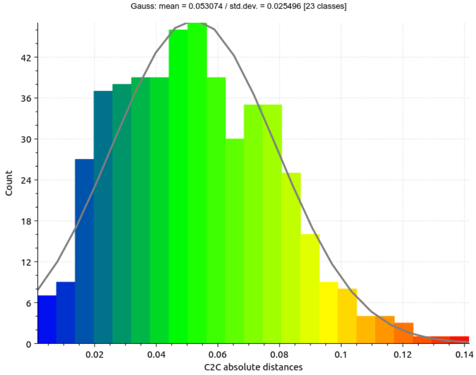


Figure 15: Convex hull VS Concave hull

# Map creation based on tactile data

*Result: point cloud comparison with ground truth*

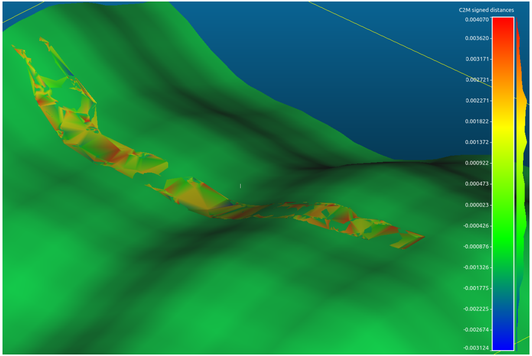


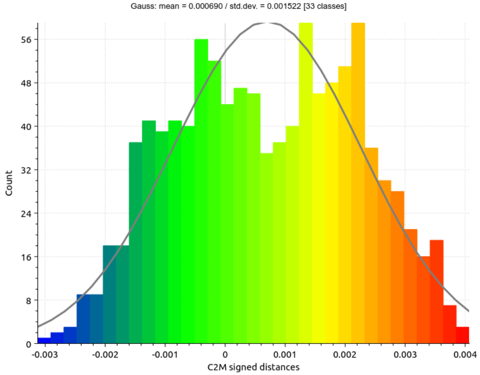



RMSE 0.053074, std 0.025

# Map creation based on tactile data

*Result: mesh comparison with ground truth*




RMSE 0.00167067, std 0.0005

# Summary

- Robot was created
- Force transducer based on Velostat was created and was investigated
- Robot can distinct rubber and rock terrains
- Robot can build a map using tactile sensors

# Deserve “A” grade!

– Oleg Bulichev


o.bulichev@innopolis.ru

@Lupasic

Room 105 (Underground robotics lab)