Lab-04-Optoforce

Students: Alexis GIBERT / Souha GHAZOUANI

1 Goals

Detect terrain bumps using the Optoforce 3D force sensor.

2 Work-lab description

- Software Environment: MATLAB
- Tools: Optoforce 3D force sensor, insulator tape (for creating bumps)
- Preparation: Attach 5-6 insulation stripes to the table and cover the silicon-semisphere of the sensor.
- Measurement Setup: Pull the sensor upside down on a smooth surface containing perpendicular stripes (bumps).
- Measurement Data: The sensor sends measurement data in 14-byte packets through a serial port. Data includes sensorial information and temperature.

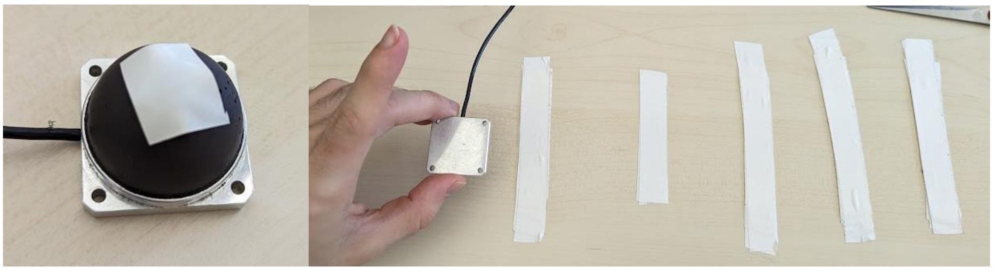

Figure 1 : Left to right - Optoforce sensor and 5 insulation stripes

3 Sensor description

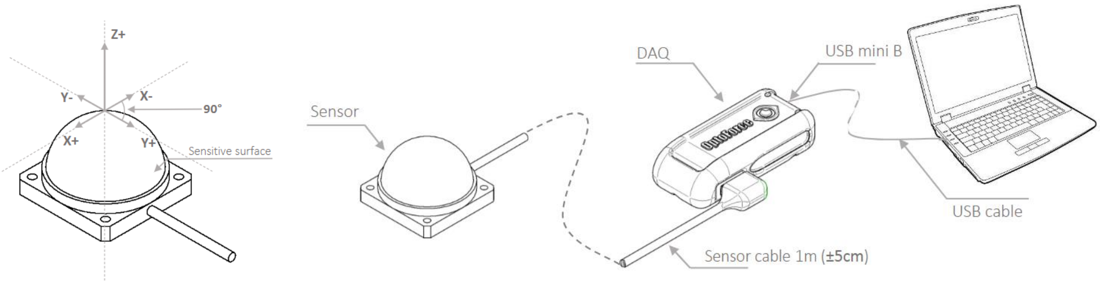

Figure 2 : Optoforce sensor, and their connections.

Unlike in other technologies, OptoForce sensors have only one structure for measuring deformation along the 3-axes (X, Y, Z). In optical force sensors photodiodes are measuring the amount of reflected light, originally emitted by the LED. By comparing the measured values on the photodiodes, the acting forces can be precisely reconstructed – and not just the magnitude, but also the direction.

The analog signals generated by the OMD are digitalized and pre-processed (using advanced noise filtering technology) by the Data Acquisition (DAQ) board to prepare the data for analyzing and recording on a personal computer. The DAQ board must be connected to our computer.

4 Measurement Steps

- Check the allocated port for the sensor and update the sample code (optoforce_test.m).
- Understand and run the sample code to observe sensor sensitivity and time-responsivity.
- Capture sensor output for a few seconds.
- Save recorded data to a .mat file.
- Load the saved archive.
- Detect peaks of the signal for bump detection.

5 Bump Detection Algorithm

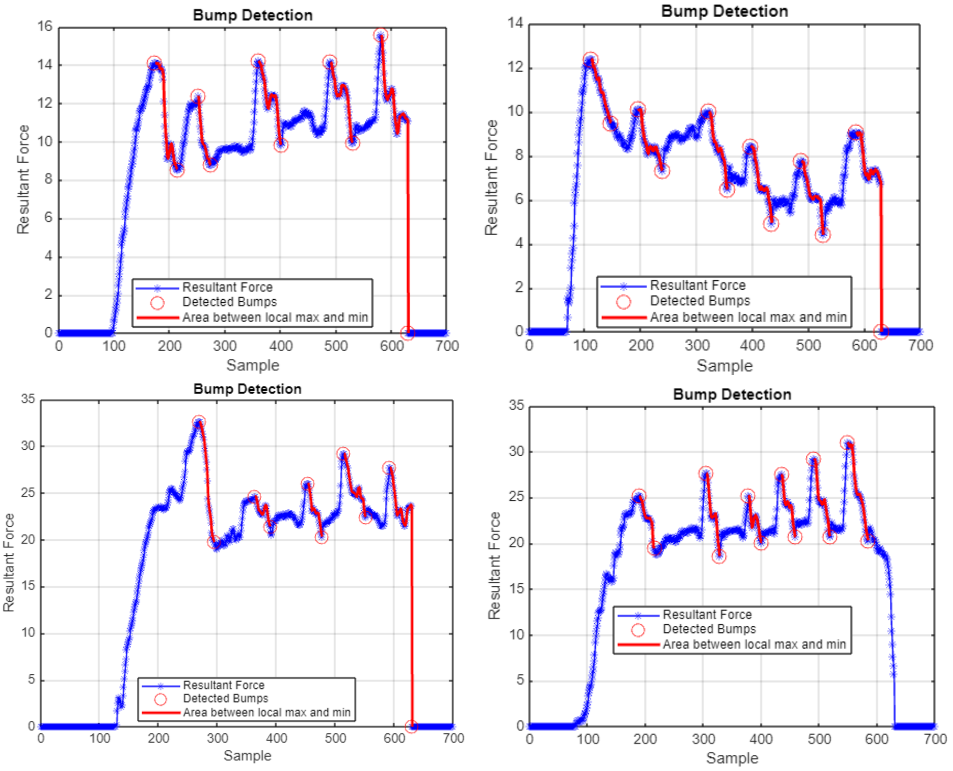

Figure 3 : Left to right, up to down - Bump detection for:

- *optoforce_raw_coords_1 with 5 bands, (threshold = 3, window_size = 70)*
- *optoforce_raw_coords_2 with 6 bands (threshold = 3, window_size = 70)*
- *optoforce_raw_coords_3 with 5 bands (threshold = 3, window_size = 50)*
- *optoforce_raw_coords_4 with 6 bands (threshold = 3, window_size = 50)*

A bump detection algorithm was devised leveraging the identification of local maxima and minima within the force signal. By pinpointing these key points, the algorithm efficiently discerns the presence of terrain irregularities. Additionally, thresholding with constant windowing was integrated into the algorithmic framework, providing a systematic approach to detect bumps across varying conditions. Furthermore, to enhance detection accuracy, zero values within the signal were disregarded, ensuring a more precise analysis of the terrain's surface characteristics. To facilitate interpretation and analysis, the detected bumps were graphically depicted alongside the force signal, offering a comprehensive visualization of the terrain's profile. This approach not only enables effective detection but also facilitates informed decision-making in applications requiring terrain assessment and navigation.

6 Our code

```
load('./optoforce_raw_coords_2.mat');
threshold = 3;    % Threshold value for bump detection
window_size = 70; % Window size for detection
bump_indices = [];

i = window_size + 1;
while i <= length(res_t)
    window_mean = mean(res_t(i - window_size:i));
    if res_t(i) > threshold && res_t(i) > window_mean && res_t(i) ~= 0
        [~, max_index] = max(res_t(i:i + window_size));
        [~, min_index] = min(res_t(i + max_index:i + window_size));
        if min_index > max_index
            bump_indices(end + 1) = i + max_index - 1;
            bump_indices(end + 1) = i + max_index + min_index - 1;
            i = i + max_index + min_index;
        else
            i = i + 1;
        end
    else
        i = i + 1;
    end
end

clf
figure(1);
plot(res_t, 'b*-');
hold on;
plot(bump_indices, res_t(bump_indices), 'ro', 'MarkerSize', 10);
for k = 1:2:length(bump_indices)
    if k < length(bump_indices)
        max_index = bump_indices(k);
        min_index = bump_indices(k+1);
        % Plot red line between local maximum and minimum
        plot(max_index:min_index, res_t(max_index:min_index), 'r', 'LineWidth', 2);
    end
end
xlabel('Sample');
ylabel('Resultant Force');
title('Bump Detection');
legend('Resultant Force', 'Detected Bumps', 'Area between local max and min', 'Location',
'best');
grid on;

disp('Detected Bump Indices:');
disp(bump_indices);
```