TÉCNICO LISBOA

Robotics

MEEC

Orientation Estimation and Trajectory Analysis Using Sensor Data

---

Authors:

Vladimiro José de Medeiros Roque (98589) vladimiro.roque@tecnico.ulisboa.pt
José Pedro da Cunha Rodrigues (113234) jose.p.rodrigues@tecnico.ulisboa.pt
Miguel Rodrigues Ferreira (113289) miguel.r.ferreira@tecnico.ulisboa.pt
Pedro Alexandre Ferreira Dias Lopes (103194) pedroafdlopes@tecnico.ulisboa.pt

Group 9

2024/2025 – 1º Semester, P2

Contents

1 Group Members Contribution

2 Introduction

This laboratory assignment focuses on two core objectives: understanding the estimation of orientation using data from a rate-gyro sensor and an accelerometer, and demonstrating the application of an industrial-grade serial manipulator. By addressing these objectives, students will gain practical experience in processing sensor data, applying mathematical models, and reconstructing trajectories in both Cartesian and orientation spaces. The work begins with the analysis of sensor data, provided in unique datasets for each group, containing approximately 20 seconds of measurements. The initial 5 seconds represent a static phase where the sensor remains stationary, providing a baseline for filtering and processing. These datasets include accelerometer readings (in milli-g) and rate-gyro readings (in degrees per second), formatted to facilitate computational analysis. The assignment is divided into multiple tasks, starting with data visualization, filtering, and theoretical trajectory reconstruction equations. Further tasks involve the graphical representation of reconstructed trajectories and their interpretation. In the later stages, the focus shifts to the use of the Scorbot VII manipulator. Here, students will derive the robot's direct kinematics equations and assess whether the reconstructed trajectories can be executed by the manipulator. By combining theoretical concepts with practical implementation, this lab offers an in-depth understanding of sensor data processing and robotic manipulator operations.

3 Tasks

3.1 Task 1

Task 1 involves visualizing sensor data by plotting the components of the accelerometer and rate-gyro measurements along their respective axes. This step is critical for understanding the behavior of the sensor during the data collection process and identifying any initial patterns or anomalies in the raw data. The dataset provided contains time-stamped measurements from an accelerometer and a rate-gyro, with values distributed across three axes: x, y, and z. The accelerometer data (measured in milli-g) reflects linear acceleration along each axis, while the rate-gyro data (measured in degrees per second) provides angular velocity along the same axes. The first few seconds of the dataset capture the sensor in a static configuration, offering a baseline for comparison against later movements. Through the visualization of these components in a combined plot, we aim to:

- Distinguish the data trends for each axis.
- Identify any irregularities or noise in the data.
- Provide a foundation for further processing in subsequent tasks.
- This initial analysis will serve as the starting point for understanding the sensor's performance and the nature of the motion captured.

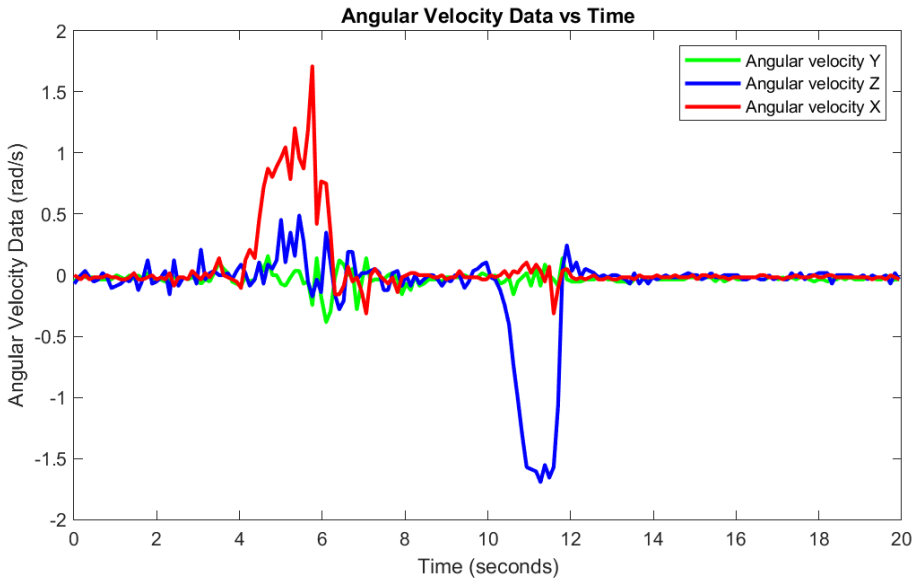

Figure 1: Angular velocity vs time.

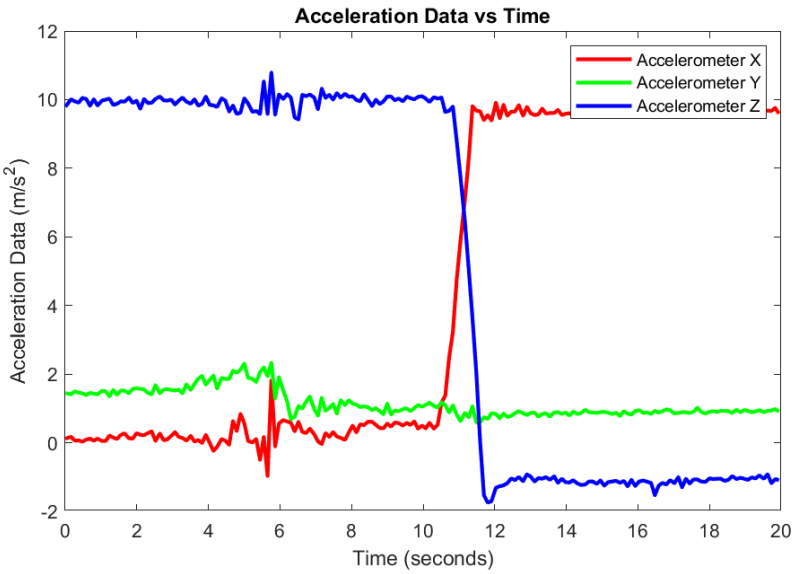

Figure 2: Acceleration vs time

Z-Axis (Blue Line)

In the acceleration graph, the blue line starts near $10\,\mathrm{m/s^2}$, which corresponds to the gravitational acceleration. This confirms that the blue line represents the z-axis, as it aligns with gravity during the static phase of the robotic arm.

In the angular velocity graph, the blue line represents ω_z (angular velocity around the z-axis). As expected, there is a strong correlation between the changes in acceleration and angular velocity along this axis:

- For example, at around 12 seconds, both acceleration and angular velocity show a significant negative dip, confirming the z-axis mapping.

X-Axis (Red Line)

In the acceleration graph, the red line shows values near $0\,\mathrm{m/s^2}$ during the static phase, confirming that it is not aligned with gravity. However, during the dynamic phase (after ~ 5 seconds), it shows significant positive and negative variations due to movement along this axis.

In the angular velocity graph, the red line represents ω_x (angular velocity around the x-axis). We observe that:

- When the angular velocity (ω_x) increases positively (e.g., around 6 seconds), the corresponding acceleration (a_x) also shows a positive variation.
- This synchronized behavior validates that the red lines in both graphs represent the x-axis.

Y-Axis (Green Line)

In the acceleration graph, the green line remains near a low positive value during the static phase, indicating that it is neither aligned with gravity nor orthogonal to it. During the dynamic phase, it exhibits moderate oscillations due to translational and rotational effects.

In the angular velocity graph, the green line represents ω_y (angular velocity around the y-axis). Observing the dynamic phase:

- Positive peaks in ω_y (e.g., around 6 seconds) correspond to similar positive peaks in a_y, confirming the mapping of the green line to the y-axis.

Alignment in Variation

For all three axes (x, y, and z), we observe that when the angular velocity increases positively, the corresponding acceleration also increases positively. Similarly, when the angular velocity decreases or becomes negative, the corresponding acceleration exhibits a similar negative variation. This consistent relationship occurs because:

Rotational Dynamics Influence Linear Dynamics

- In a robotic arm, each joint's motion generates a combination of tangential and centripetal accelerations, which are directly tied to the angular velocity and its rate of change.
- The tangential acceleration depends on the rate of change of angular velocity ($\dot{\omega}$), while the centripetal acceleration depends on the square of angular velocity (ω^2) and acts perpendicular to the direction of rotation.

Controlled System Dynamics

- Robotic arms are designed for precise and coordinated movements. Each joint is controlled by motors that apply predictable torques, resulting in synchronized changes in angular velocity and acceleration.
- When a joint rotates positively (e.g., $\omega_x > 0$), it induces a linear acceleration along the same axis as part of the controlled rotational motion.

Fixed Axis of Rotation

- Unlike free-moving objects, the axes of rotation in a robotic arm are fixed, ensuring that the measured angular velocities and accelerations align in their respective directions without interference from external or random forces.

This predictable alignment between angular velocity and acceleration highlights the consistency of the robotic arm's motion and validates the observed variations in the provided graphs.

3.2 Task 2

Preprocessing sensor data is a vital step to ensure accurate and meaningful analysis. Raw data, collected from accelerometers and rate-gyros, often contain noise and occasional outliers caused by sensor imperfections, environmental factors, or abrupt movements. These irregularities can obscure the true motion dynamics and negatively affect subsequent analyses. In this case, a median filter was applied to clean the data effectively. This filtering technique is well-suited for handling outliers and reducing noise, as it replaces each data point with the median of its neighbors within a defined window. Unlike other filtering methods, the median filter preserves sharp transitions and edges in the data while removing unwanted spikes, making it ideal for dynamic motion data. The results achieved with the median filter demonstrated significant improvements, yielding smooth and reliable datasets without distortions. This preprocessing ensures that the data is well-prepared for reconstructing accurate trajectories and performing further analysis with confidence in the validity of the underlying patterns.

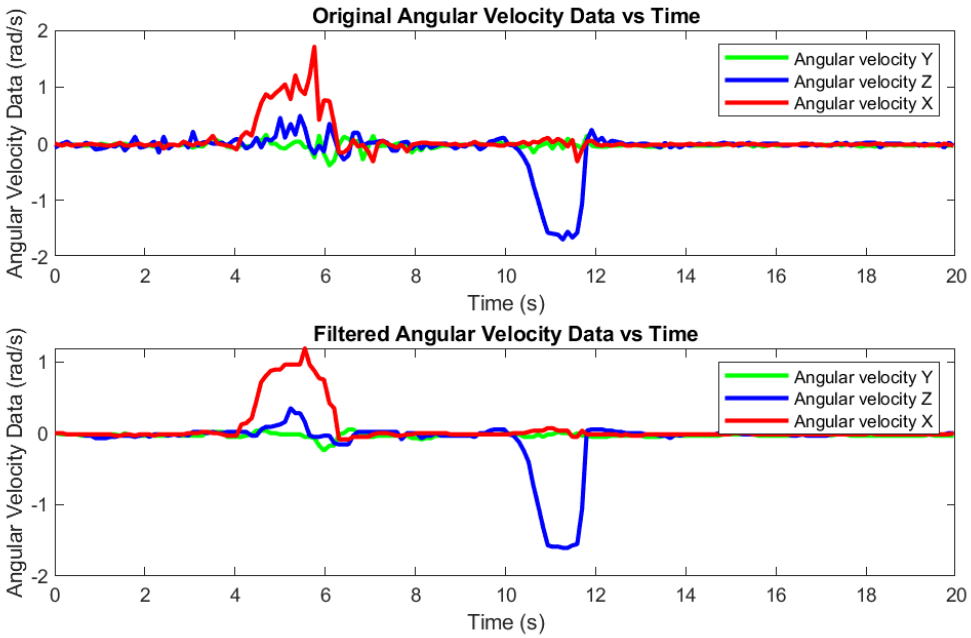

Figure 3: Filtered angular velocity vs original (using median filter).

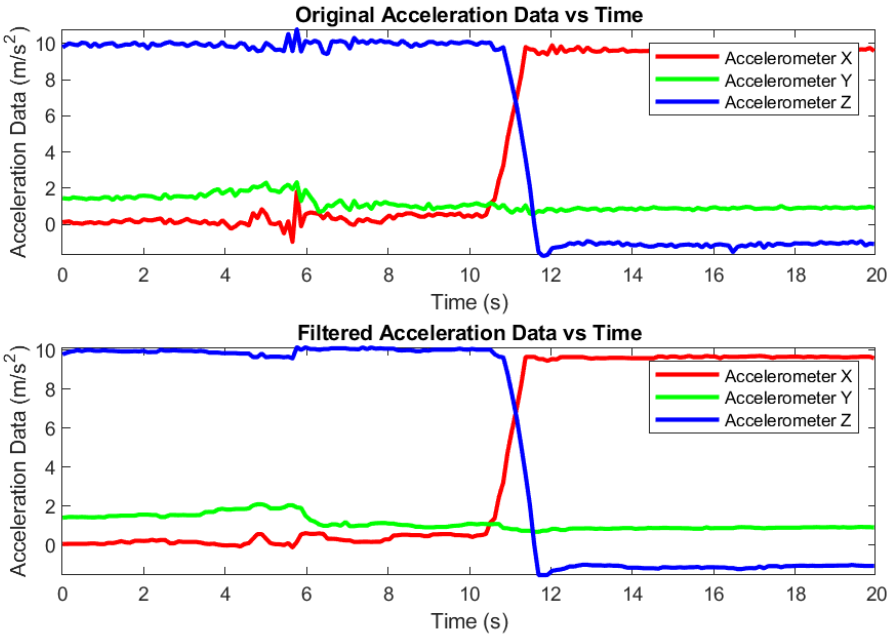

Figure 4: Filtered acceleration vs original (using median filter)

Comparison of Filters

1. Sliding Window Average/Median Filters

- **Average Filter:** Calculates the average of data points within a sliding window, effectively reducing high-frequency noise. However, it is sensitive to outliers, as extreme values can significantly skew the average.
- **Median Filter:** Replaces each point with the median of its neighbors within the sliding window. This makes it more robust than the average filter, as it can effectively handle outliers without distorting the signal.

2. Gaussian Filters

- Smooths data by applying a weighted average, where weights follow a Gaussian distribution. This reduces noise while preserving overall trends better than simple averaging.
- It is less robust to outliers compared to the median filter, as the weights are still influenced by extreme values.
- Requires the selection of a standard deviation parameter (σ), which controls the level of smoothing, making it less straightforward for general applications.

3. Savitzky–Golay Filters

- Fits a polynomial to a subset of data points within a sliding window, then replaces the center point with the value predicted by the polynomial.

- Effective at preserving sharp transitions and trends while reducing noise, making it ideal for smooth but non-linear signals.
- Sensitive to outliers, as the polynomial fit can be distorted by extreme values within the window.
- Requires careful tuning of the polynomial order and window size for optimal results.

4. Smooth Filters

- General-purpose filters that reduce noise by averaging data points or applying weighted smoothing.
- These filters can smooth data effectively but often blur sharp transitions, making them less suitable for dynamic systems like robotic arms.
- They do not specifically address outliers, which can still influence the smoothed result.

5. Outliers Filters

- Specifically designed to detect and remove outliers based on statistical thresholds (e.g., Z-Score or Interquartile Range methods).
- While effective at identifying and removing anomalies, these filters often leave gaps in the data by replacing outliers with `NaN`, requiring additional interpolation steps.
- Do not inherently smooth the data or reduce noise in non-outlier regions.

Why Use the Median Filter?

The Median Filter stands out compared to these alternatives due to its:

- **Robustness:** Unlike the Average, Gaussian, or Savitzky–Golay filters, the Median Filter is highly robust against outliers and extreme values, as it is unaffected by the magnitude of data points outside the sliding window.
- **Preservation of Trends:** The Median Filter preserves sharp transitions and dynamic changes better than smooth or Gaussian filters, which tend to over-smooth the data.
- **Simplicity:** Requires only a window size as a parameter, making it simpler to implement than Gaussian or Savitzky–Golay filters, which require tuning of additional parameters (e.g., σ or polynomial order).
- **Completeness:** Unlike Outliers Filters, the Median Filter does not leave gaps in the data and simultaneously reduces noise and removes anomalies.

3.3 Task 3

3.4 Task 4

To reconstruct the sensor orientation in Euler angles (α, β, γ) using rate-gyro data, we analyzed the relationship between angular velocities $(\omega_x, \omega_y, \omega_z)$ and the derivatives of the Euler angles. The **ZYX Euler angle convention** was selected because it provides an intuitive sequence for controlling robotic manipulators, such as the Scorbot VII, allowing separate adjustments for direction, height, and tool orientation.

3.4.1 Theory and Mathematical Formulation

In the ZYX convention, the angles represent rotations applied in a specific order:

- **Yaw (γ)** — rotation about the global Z-axis. This initial rotation aligns the manipulator to face the target, orienting its base towards the goal.
- **Pitch (β)** — rotation about the new Y-axis after the yaw rotation. This step adjusts the vertical alignment of the manipulator relative to the goal.
- **Roll (α)** — rotation about the new X-axis following the yaw and pitch rotations. This final adjustment ensures the gripper or tool is properly oriented for interaction with the target.

By properly parameterizing these rotations, we ensure predictable and precise control of the manipulator's movements, essential for tasks involving accurate positioning and orientation.

3.4.2 Parameterization

To reconstruct the orientation accurately, we define the following parameters:

- α, β, γ: Euler angles representing the sensor's orientation in 3D space.
- $\omega_x, \omega_y, \omega_z$: Angular velocities provided by the rate-gyro, indicating rotational speeds around the sensor's local axes.
- Δt: Time interval between consecutive measurements.
- $\mathbf{T}(\alpha, \beta)$: Transformation matrix that relates the Euler angle rates to the angular velocities.

3.4.3 Derivation of the Transformation Matrix

To relate the angular velocities $(\omega_x, \omega_y, \omega_z)$ to the time derivatives of the Euler angles $(\dot{\alpha}, \dot{\beta}, \dot{\gamma})$, we derive the transformation matrix based on the sequence of rotations in the ZYX Euler angle convention.

1. Rotation Matrices
The individual rotation matrices for each Euler angle are:

- Rotation about Z-axis (γ):

$$\mathbf{R}_z(\gamma) = \begin{bmatrix} \cos\gamma & -\sin\gamma & 0 \\ \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{1}$$

- Rotation about Y-axis (β):

$$\mathbf{R}_y(\beta) = \begin{bmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{bmatrix} \tag{2}$$

- Rotation about X-axis (α):

$$\mathbf{R}_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{bmatrix} \tag{3}$$

2. Combined Rotation Matrix

The overall rotation matrix $\mathbf{R}$ representing the orientation is obtained by multiplying the individual rotation matrices in the order of rotations:

$$\mathbf{R} = \mathbf{R}_z(\gamma)\mathbf{R}_y(\beta)\mathbf{R}_x(\alpha). \tag{4}$$

3. Angular Velocity in Terms of Euler Angle Rates

In the ZYX Euler angle convention, the roll angle α is the last rotation applied. Consequently, its rate of change $\dot{\alpha}$ represents rotation about the current X-axis, which remains fixed relative to the body frame after the yaw (γ) and pitch (β) rotations. As a result, $\dot{\alpha}$ directly contributes to the angular velocity vector along the X-axis without requiring any transformation.

The angular velocity vector $\boldsymbol{\omega}$ in the body frame can be expressed in terms of the Euler angle rates as follows:

$$\boldsymbol{\omega} = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} = \begin{bmatrix} \dot{\alpha} \\ 0 \\ 0 \end{bmatrix} + \mathbf{R}_x(-\alpha) \begin{bmatrix} 0 \\ \dot{\beta} \\ 0 \end{bmatrix} + \mathbf{R}_x(-\alpha)\mathbf{R}_y(-\beta) \begin{bmatrix} 0 \\ 0 \\ \dot{\gamma} \end{bmatrix} \tag{5}$$

The matrices $\mathbf{R}_x(-\alpha)$ and $\mathbf{R}_y(-\beta)$ are the inverse rotations of $\mathbf{R}_x(\alpha)$ and $\mathbf{R}_y(\beta)$, respectively. Since rotation matrices are orthogonal, the inverse is equivalent to their transpose:

$$\mathbf{R}_x(-\alpha) = \mathbf{R}_x(\alpha)^T, \quad \mathbf{R}_y(-\beta) = \mathbf{R}_y(\beta)^T.$$

Expanded Form of Angular Velocity:

Expanding the terms in Equation 5 provides:

$$\boldsymbol{\omega} = \begin{bmatrix} \dot{\alpha} \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & \sin\alpha \\ 0 & -\sin\alpha & \cos\alpha \end{bmatrix} \begin{bmatrix} 0 \\ \dot{\beta} \\ 0 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & \sin\alpha \\ 0 & -\sin\alpha & \cos\alpha \end{bmatrix} \begin{bmatrix} \cos\beta & 0 & -\sin\beta \\ 0 & 1 & 0 \\ \sin\beta & 0 & \cos\beta \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ \dot{\gamma} \end{bmatrix}. \tag{6}$$

This equation forms the foundation for deriving the transformation matrix $\mathbf{T}(\alpha, \beta)$, which relates the angular velocities $(\omega_x, \omega_y, \omega_z)$ to the Euler angle rates $(\dot{\alpha}, \dot{\beta}, \dot{\gamma})$.

4. Transformation Matrix
Combining the terms in 5 we get the transformation matrix

$$\begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 & -\sin\beta \\ 0 & \cos\alpha & \sin\alpha\cos\beta \\ 0 & -\sin\alpha & \cos\alpha\cos\beta \end{bmatrix}}_{\mathbf{T}(\alpha,\beta)} \begin{bmatrix} \dot{\alpha} \\ \dot{\beta} \\ \dot{\gamma} \end{bmatrix}. \tag{7}$$

5. Inverting the Transformation Matrix
To compute the Euler angle rates from the angular velocities, we invert the 7

$$\begin{bmatrix} \dot{\alpha} \\ \dot{\beta} \\ \dot{\gamma} \end{bmatrix} = \mathbf{T}^{-1}(\alpha, \beta) \cdot \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \tag{8}$$

And we get the inverse transformation matrix $\mathbf{T}^{-1}(\alpha, \beta)$:

$$\mathbf{T}^{-1}(\alpha, \beta) = \begin{bmatrix} 1 & \sin\alpha\tan\beta & \cos\alpha\tan\beta \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \dfrac{\sin\alpha}{\cos\beta} & \dfrac{\cos\alpha}{\cos\beta} \end{bmatrix} \tag{9}$$

3.4.4 Final Equations

Using Equation (9), the rates of change of the Euler angles are derived as follows:

$$\dot{\alpha} = \omega_x + \sin\alpha\tan\beta \cdot \omega_y + \cos\alpha\tan\beta \cdot \omega_z, \tag{10}$$

$$\dot{\beta} = \cos\alpha \cdot \omega_y - \sin\alpha \cdot \omega_z, \tag{11}$$

$$\dot{\gamma} = \frac{\sin\alpha}{\cos\beta} \cdot \omega_y + \frac{\cos\alpha}{\cos\beta} \cdot \omega_z. \tag{12}$$

These equations relate the angular velocities $(\omega_x, \omega_y, \omega_z)$ to the rates of change of the Euler angles $(\dot{\alpha}, \dot{\beta}, \dot{\gamma})$ in the ZYX convention, and they will serve as the foundation for reconstructing the sensor's orientation.

Appendices