# COMP0217 Final Project Report: Sub-Terranean Navigation Project

Ethan Hocquellet, Rohan Kishore, Xavier Parker, Ashraya Poudel
(zcabelp, zcabrki, zcabmp, zcabapo) @ ucl.ac.uk
*Department of Computer Science, University College London, London, UK*

March 2026

## Abstract

The following report recounts our state estimation approach for the COMP0217 Sub-Terranean Navigation Project. Using accelerometer, gyroscope, magnetometer and Time-of-Flight (ToF) sensors, we aim to estimate the 2D position and yaw of a wheeled robot without GPS. We detail the construction, calibration, and sensor integration of our estimator. By testing the system on the provided datasets and benchmarking the outputs against ground truth, we thoroughly analyse its accuracy, durability, and limitations.

***Index Terms***— Sensor Fusion, State Estimation, Extended Kalman Filter, Inertial Sensing, Time-of-Flight, Robotics

## 1 Introduction

In this project, we were tasked with accurately estimating the planar position $(x, y)$ and yaw $\psi$ of a mobile robot for both simple and complex trajectories. This robot is equipped with inertial sensors and three ToF range sensors, and is bounded in an indoor arena of dimensions 2.44m by 2.44m.

This report will describe the estimation problem and explain our sensor fusion algorithm. We will also describe our motivation for modelling choices, and provide an evaluation of the algorithm's performance on the given datasets.

## 2 Project Setup and Problem Formulation

### 2.1 Available Sensor Streams

Table 1 summarises the main logged data streams available in the provided datasets.

Table 1: Available logged data streams used by the estimator.

| Signal | Nominal Rate |
|---|---|
| GT position, GT rotation, GT time | 200 Hz |
| Sensor ACCEL, Sensor GYRO | 104 Hz |
| Sensor LP ACCEL, Sensor Temp | 100 Hz |
| Sensor MAG | 50 Hz |
| Sensor ToF1, Sensor ToF2, Sensor ToF3 | 10 Hz |

Ground truth rotation is provided as quaternions. In this project, we use it only for initialisation and evaluation, while the estimator itself relies on onboard sensing.

### 2.2 Estimation Objective

The estimator must produce a real-time estimate of the robot pose while also keeping track of other properties of the robot to maintain accuracy on the pose. The state vector we used is written as:

$$\mathbf{x}_k = \begin{bmatrix} x_k & \dot{x}_k & \ddot{x}_k & y_k & \dot{y}_k & \ddot{y}_k & \psi_k & \dot{\psi}_k \end{bmatrix}^\top, \quad (1)$$

Notably, the state vector is all in the world frame rather than the local frame of the robot. This is to ensure the readings we get are interpretable.

### 2.3 Tasks

The estimator is evaluated on two core tasks:

1. **Task 1:** Straight-line motion with forward and backward segments.

2. **Task 2:** A more complex circuit trajectory with varying motion.

# 3 Methodology

## 3.1 System Model

We model the estimator dynamics in discrete time as

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k, \Delta t) + \mathbf{w}_k, \tag{2}$$

where $f$ contains the individual kinematic equations denotes the control-relevant sensor input or dead-reckoning input, $\Delta t$ is the sampling interval, and $\mathbf{w}_k$ is zero-mean process noise with covariance $\mathbf{Q}_k$. We made the decision to treat the acceleration between readings constant as a method of linearising the system. By linearising the system in the prediction step, we did not have to compute any Jacobians as the transition matrix of all the functions inside $f$ is the same as the Jacobians of the system. Therefore our $F$ matrix can be written as

$$\mathbf{F} = \begin{bmatrix} 1 & \Delta t & \frac{1}{2}\Delta t^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & \alpha_v & \Delta t & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & \Delta t & \frac{1}{2}\Delta t^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & \alpha_v & \Delta t & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \tag{3}$$

where $\alpha_v$ is a coefficient mimicking drag and is a hyper parameter we can tune, the value we settled on was 0.98.

The measurement model is written as

$$\mathbf{z}_k = h(\mathbf{x}_k) + \mathbf{v}_k, \tag{4}$$

where $\mathbf{z}_k$ contains the available sensor observations and $\mathbf{v}_k$ is zero-mean measurement noise with covariance $\mathbf{R}_k$.

For the ToF sensors, the $(x, y)$ distance of the robot to the nearest wall can be calculated following this set of equations:

$$\begin{bmatrix} x_{sensor} \\ y_{sensor} \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} \cos(\psi) & -\sin(\psi) \\ \sin(\psi) & \cos(\psi) \end{bmatrix} \begin{bmatrix} dx \\ dy \end{bmatrix} \tag{5}$$

$$\begin{bmatrix} x(t) \\ y(t) \end{bmatrix} = \begin{bmatrix} x_{sensor} \\ y_{sensor} \end{bmatrix} + t \cdot \begin{bmatrix} \cos(\psi_{offset}) \\ \sin(\psi_{offset}) \end{bmatrix} \tag{6}$$

Where $\psi_{offset}$ is calculated by adding the respective offset from each sensor onto the psi from the state vector. Producing the vector $(x, y)$ where the wall is according to the sensor. Furthermore, because $\begin{bmatrix} \cos(\psi_{offset}) & \sin(\psi_{offset}) \end{bmatrix}^\top$ is a unit vector, $t$ can be considered the distance travelled. $x_{max}$, $x_{min}$, $y_{max}$ and $y_{min}$ are defined as the borders of the arena. The distance to each wall is then calculated for each respective wall as

$$t = \frac{x_{max} - x_{sensor}}{\cos(\psi_{offset})} \tag{7}$$

where $x_{max}$, $x_{sensor}$ and cos change with respective to which wall. The smallest distance can then be used as the value d. To calculate the Jacobian, H, of the time of flight sensors, they are only affected by $\psi$ so only the state referring to the change in $\psi$ will be non-zero.

For the rest of the sensors, because we instantly turn the readings from the local frame of the robot to the world frame, that allowed all our Jacobians to be very simple and only have values of 1 or 0 depending on which state was affected by that sensor.

## 3.2 Estimator Design

The selected estimator design featured an Extended Kalman Filter (EKF), with the following prediction and update equations

$$\hat{\mathbf{x}}_{k|k-1} = f(\hat{\mathbf{x}}_{k-1|k-1}, \Delta t), \tag{8}$$

$$\mathbf{P}_{k|k-1} = \mathbf{F}_k \mathbf{P}_{k-1|k-1} \mathbf{F}_k^\top + \mathbf{Q}_k, \tag{9}$$

$$\mathbf{S}_k = \mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k^\top + \mathbf{R}_k, \tag{10}$$

$$\mathbf{K}_k = \mathbf{P}_{k|k-1} \mathbf{H}_k^\top \mathbf{S}_k^{-1}, \tag{11}$$

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k \left( \mathbf{z}_k - h(\hat{\mathbf{x}}_{k|k-1}) \right), \tag{12}$$

$$\mathbf{P}_{k|k} = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_{k|k-1}. \tag{13}$$

where $\mathbf{H}$ denotes the Jacobian of h.

## 3.3 Calibration and Pre-processing

Calibration parameters were determined offline using the provided calibration datasets and embedded as constants in the estimator.

**Gyroscope:** Column 1 was identified as the yaw axis by correlating integrated heading with ground-truth ro-

tation. The static bias of +0.00188 rad/s and measurement noise standard deviation of 0.005 rad/s were estimated from stationary data.

**Accelerometer:** Column 1 contains gravity ($\approx 10$ m/s$^2$). Columns 2 and 3 measure lateral and forward acceleration respectively, with biases of $+0.028$ m/s$^2$ and $-0.396$ m/s$^2$ estimated from stationary data.

**ToF Sensors:** Orientation mapping was determined by comparing initial readings against expected wall distances from known start positions. The mapping differs from physical labelling: ToF1→Left, ToF2→Backward, ToF3→Right.

**Magnetometer:** Hard-iron and soft-iron calibration was attempted, but the corrected heading exhibited 78° standard deviation, rendering it unusable. Magnetometer updates are disabled.

**ToF Rejection:** Measurements are rejected if: status $\neq 0$, range outside $[0.05, 2.5]$ m, signal strength $< 300$, angular rate $> 0.3$ rad/s (suppressed during rotation to avoid wall-hole artefacts), or innovation exceeds $\min(3\sigma, 0.8\text{ m})$.

**Time Synchronisation:** All sensors are logged at 200 Hz via zero-order hold; duplicate ToF readings are detected and used only for change-detection, not filtered out.

**Initial Heading:** A two-pass brute-force scan (1° coarse, then 0.05° fine) finds the heading minimising ToF prediction error at startup, refined after 5 unique ToF readings.

**Zero-Velocity Updates:** When $|\omega| < 0.08$ rad/s, ToF readings have not changed for $> 0.2$ s, and all accelerometer axes are near their stationary values, pseudo-measurements $v_x = v_y = 0$ and $a_x = a_y = 0$ are injected to prevent drift during stationary periods.

### 3.4 Implementation Details

The myEKF function operates as a single-step, 200 Hz extended Kalman filter where logged .mat sensor data is fed row-by-row from an external wrapper script, which also supplies the ground-truth initialisation on the first call. To smoothly handle asynchronous sensor streams, the filter utilises strict validation gates that ignore NaN values and skip redundant, zero-order hold readings from slower sensors by comparing them to previous values. Within each update cycle, the estimator follows a rigid execution order: it predicts the next state using a constant-acceleration kinematic model, sequentially applies measurement updates from the gyro, magnetometer, accelerometer, and Time-of-Flight sensors (restricting ToF to position only), and concludes with a zero-velocity update (ZUPT) if the robot is stationary. At the end of this sequence, the updated state estimate and covariance matrices are returned to the external caller, which is responsible for storing the iterative outputs for post-simulation trajectory evaluation.

## 4 Experimental Setup

### 4.1 Datasets and Evaluation Protocol

**Calibration:** Two calibration datasets were used to determine sensor parameters:

- `calib1_rotate.mat`: 60 seconds of continuous rotation (5 full turns) used to identify the gyroscope yaw axis, estimate gyro bias, and attempt magnetometer calibration.
- `calib2_straight.mat`: 60 seconds of stationary then straight-line motion used to estimate accelerometer biases and determine ToF sensor orientations.

**Parameter Tuning:** Process noise covariance $Q$, measurement noise covariances $R$, and gating thresholds were tuned iteratively on `task2_3 1.mat`, chosen as a representative circuit trajectory.

**Evaluation:** The estimator was evaluated on all provided datasets without modification:

- Task 1: `task1_1 1.mat`, `task1_2 1.mat`, `task1_3.mat`
- Task 2: `task2_1 1.mat`, `task2_2 1.mat`, `task2_3 1.mat`, `task2_4.mat`

**Initialisation:** The initial position $(x_0, y_0)$ for the datasets given are taken from the first ground-truth sample, as permitted by the project specification. However, the initial heading $\psi_0$ is *not* taken directly from ground truth. Instead, a 360° brute-force scan is performed at startup: for each candidate heading at 1° resolution, the sum-squared error between measured ToF distances and predicted wall distances is computed, and

the heading minimizing this error is selected. This approach simulates a practical scenario where the robot knows its approximate starting location but must determine its own orientation from sensor measurements. For the unseeen coursework dataset tests we have assumed to initial start at the middle of the left wall.

### 4.2 Performance Metrics

Performance of the state estimator is evaluated using the Root Mean Squared Error (RMSE) over a trajectory of length N:

$$\mathrm{RMSE}_{\mathrm{pos}} = \sqrt{\frac{1}{N}\sum_{k=1}^{N}\left[(\hat{x}_k - x_k^{\mathrm{gt}})^2 + (\hat{y}_k - y_k^{\mathrm{gt}})^2\right]}, \tag{14}$$

and yaw RMSE can be computed as

$$\mathrm{RMSE}_{\psi} = \sqrt{\frac{1}{N}\sum_{k=1}^{N}\left[\mathrm{wrap}(\hat{\psi}_k - \psi_k^{\mathrm{gt}})\right]^2}, \tag{15}$$

Additionally, the final position error was evaluated:

$$\mathrm{FPE}_{\mathrm{pos}} = \sqrt{(\hat{x}_N - x_N^{gt})^2 + (\hat{y}_N - y_N^{gt})^2} \tag{16}$$

## 5 Results

### 5.1 Quantitative Results

| Dataset | Pos. RMSE (m) | Yaw RMSE (deg) | Final Err. (m) |
|---|---|---|---|
| task1_1 | 0.0340 | 2.58 | 0.0090 |
| task1_2 | 0.0326 | 2.04 | 0.0385 |
| task1_3 | 0.0337 | 2.20 | 0.0326 |
| task2_1 | 0.0891 | 6.89 | 0.0494 |
| task2_2 | 0.0671 | 6.98 | 0.0497 |
| task2_3 | 0.0340 | 5.04 | 0.0153 |
| task2_4 | 0.1090 | 5.99 | 0.0203 |

Table 2: Quantitative results

Position accuracy degrades roughly 2–3× from Task 1 to Task 2, driven almost entirely by heading drift, small heading errors compound over the longer circuit trajectories. Notably, Task 2_3 achieves Task 1-level position RMSE (0.034 m) despite being a circuit course - its shorter path and fewer turns limit heading accumulation.

### 5.2 Trajectory and Time-Series Plots

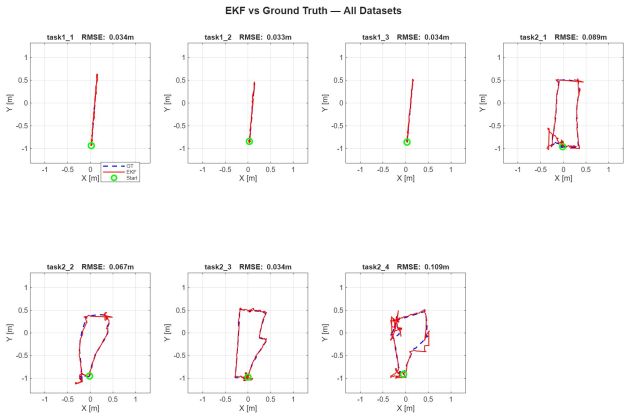


Figure 1: Trajectory comparison between estimated pose and ground truth.

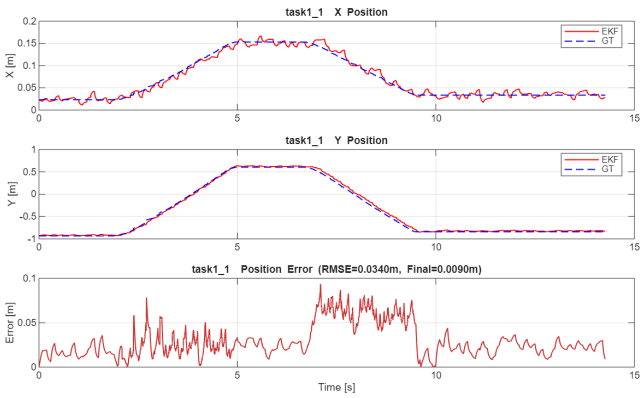


Figure 2: Time-series plot of x, y, position error for task1.1

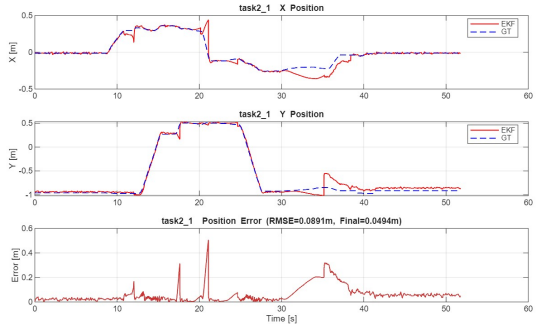


Figure 3: Time-series plot of x, y, position error for task2.1

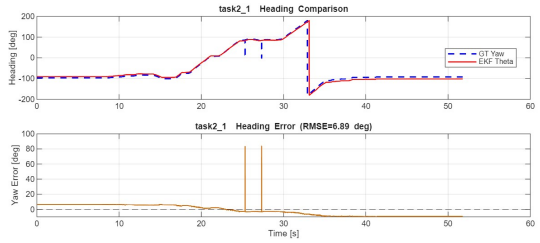


Figure 4: Time-series plot of heading error for task2.1

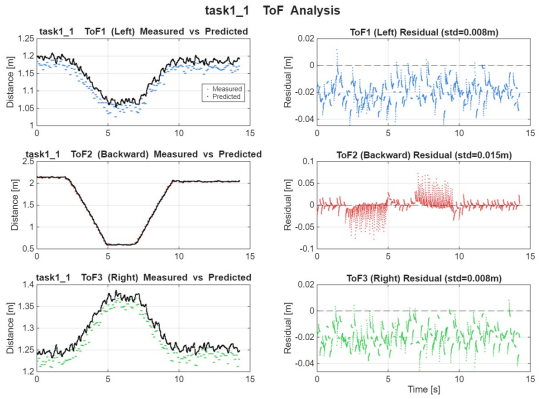


Figure 5: Time-series plot of ToF readings for task1.1

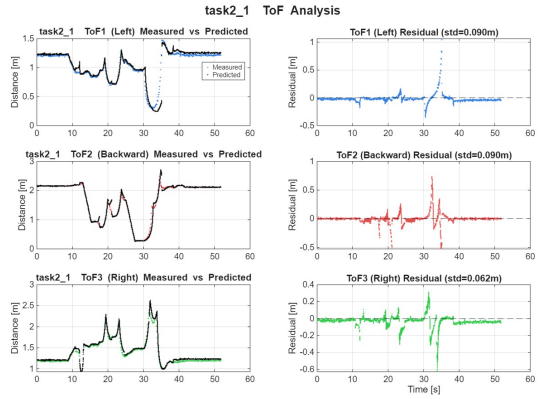


Figure 6: Time-series plot of ToF readings for task2.1

## 5.3 Task-by-Task Analysis

**Task 1 (Straight-line, Pos RMSE ∼0.03m):** The estimator performed very well on Task 1 because the robot follows a simple straight-line trajectory with no turns. The heading remains essentially constant throughout, meaning the initial heading scan from ToF provides a reliable angle that doesn't accumulate drift. With a fixed heading, the accelerometer forward/lateral decomposition into world-frame $X/Y$ stays accurate, and all three ToF sensors maintain consistent, perpendicular lines-of-sight to the arena walls - giving strong, unambiguous position corrections at every 10 Hz update. There are no wall holes in the direct sensor paths during straight travel, so ToF outlier rejection is rarely needed.

**Task 2 (Circuit course, Pos RMSE 0.03–0.11m):** Task 2 introduced multiple 90° turns, which changed the problem fundamentally. The gyroscope becomes critical for tracking heading during rotations, but it integrates bias and noise over time, leading to a $\sim 5°$–$7°$ heading drift that compounds into position error. During turns, the ToF sensors sweep across the arena walls and can briefly point through wall holes, producing erroneous range readings that must be gated out (we skip ToF updates when angular rate exceeds 0.3 rad/s). After turns complete, the constant-acceleration model retains residual velocity that the ZUPT must damp down. The longer and more dynamic the trajectory (`task2_4` being the worst at 0.11 m), the more these errors accumulate.

**Most helpful sensor in each regime:**

- Straight-line segments: ToF dominate - they provide direct, absolute position corrections to the walls and are the primary reason RMSE stays below 0.04 m.
- During turns: The gyroscope is essential-it's the only sensor tracking heading changes in real-time. The accelerometer provides little useful information during rotation (centripetal effects contaminate it), and ToF readings become unreliable as sensors sweep past wall holes.
- *Stationary periods:* The accelerometer enables zero-velocity detection (ZUPT), which prevents velocity drift from inflating position error between movements.

**Where the largest errors occurred:** The largest position errors (0.1-0.3 m spikes) occurred immediately after turns, in the 1-2 seconds before the ToF sensors re-establish reliable wall corrections. During the turn itself, ToF updates are suppressed (to avoid wall-hole artefacts), so the EKF relies solely on dead-reckoning from the gyro and accelerometer. Any heading error from the turn directly projects into position error on the subsequent straight segment. In `task2_4` specifically, the robot traverses the full arena at high speed,

causing the innovation gate (0.8 m cap) to temporarily reject valid ToF corrections-creating a brief 0.2 m error spike until the filter reconverges. The magnetometer was found to be unusable (78° heading standard deviation) and was disabled entirely.

# 6 Discussion

**Sensitivity to bias, drift, and imperfect calibration:** An incorrect sampling rate assumption (104 Hz vs true 200 Hz) amplified gyroscope integration and was a huge damaging calibration error. After correction, residual gyro bias drift still accumulated $\sim 6°$ heading error over 50-second runs, remaining the dominant error source. The magnetometer was unusable (78° heading std after calibration) and was disabled entirely.

**Robustness to invalid ToF readings and wall holes:** ToF sensors required multiple protection layers: duplicate detection prevented zero-order-hold resampling from applying each 10 Hz reading 20× at 200 Hz, while angular-rate gating (suppressing updates when $|\omega| > 0.3$ rad/s) avoided wall-hole artefacts during turns. The innovation gate cap (0.8 m) represented a key trade-off-tighter values protected against outliers but caused divergence in `task2_4` where fast motion produced legitimately large innovations.

**Effect of filter tuning and model mismatch:** High acceleration process noise ($Q_a = 3.0$) was necessary to absorb the mismatch between the constant-acceleration model and real wheel-driven dynamics. Velocity damping ($\alpha = 0.95$/step) and ZUPT were both required to prevent phantom velocities persisting after the robot stopped.

**Benefits and limitations of the state representation:** The 8-state vector $[x, v_x, a_x, y, v_y, a_y, \theta, \omega]^\top$ keeps prediction linear except for the rotation projection, and including acceleration as a state allows motion to propagate smoothly between the 100 ms ToF gaps. However, with heading corrected only by the gyroscope (no usable absolute heading sensor), the filter is fundamentally vulnerable to long-term yaw drift.

**Failure cases:** The worst performance (`task2_4`, 0.11 m RMSE) occurred during fast full-arena traversals where ToF suppression during turns combined with innovation gating created correction blackouts. A heading reacquisition feature had to be disabled after producing catastrophic jumps from incorrect local minima.

**Future improvements:** An innovation gate that widens after turns would prevent the `task2_4` divergence spiral. Fusing the unused low-power accelerometer could improve velocity estimation during motion transitions.

# 7 Conclusion

We implemented an 8-state Extended Kalman Filter fusing IMU (accelerometer, gyroscope) and three ToF rangefinders for 2D robot localisation in an arena. The filter uses a constant-acceleration kinematic model with velocity damping, a ToF ray-casting measurement model with numerical Jacobians, and several robustness mechanisms including duplicate-reading detection, angular-rate gating, zero-velocity updates ect.

The estimator achieved a position RMSE of 0.03 m on straight-line tasks and 0.03–0.11 m on circuit tasks, with final position errors consistently below 0.05 m. The main findings were: (1) ToF rangefinders are by far the most valuable sensor for absolute position correction, but require careful outlier rejection near wall holes; (2) gyroscope bias drift ($\sim 6°$ over 50 s) is the primary error source, compounded by the absence of a usable absolute heading reference after the magnetometer proved unusable; and (3) filter robustness decisions-particularly the innovation gate cap-involve direct trade-offs between outlier rejection and tracking agility during fast motion.

The most impactful next steps would be using ToF signal-strength for adaptive measurement noise weighting instead of hard gating, implementing an innovation gate that widens during periods of high uncertainty after turns, and exploring analytical ToF Jacobians to improve both accuracy and computational efficiency.