

Technical Implementation Brief: Tactile Environment Routing Interface (TERI)

Group: Group 5

Dominic Bertolo

October 8th, 2025

1 Introduction

This document aims to provide a thorough explanation of the core software architecture and algorithms used for the Tactile Environment Routing Interface (TERI). TERI is a handheld navigation device with the purpose of addressing the accessibility gap for students with visual impairments, primarily in complex indoor environments. TERI provides real-time, non-visual guidance by utilizing a hybrid indoor localization system that fuses data from an Inertial Measurement Unit (IMU) and WiFi signal fingerprinting. This technical brief details the specific implementation for each core component to demonstrate the project's feasibility.

2 System Architecture Overview

The software is designed to be a modular, multi-layered architecture to guarantee a separation of concerns about function and data handling. The system is driven by a central Navigation Module that commands the user experience from destination input to arrival. A user's input from a tactile keypad is processed by an Input Handler to select a destination in the form of a keycode. This keycode corresponds to a Reference Point (RP), a singular point in a connected graph of RPs that represent the mapped out indoor area being navigated. This corresponding RP will act as the destination to be navigated towards. The user's location is then captured utilizing a WiFi Fingerprinting Module to find which RP the user is closest and choose this RP as a starting point for navigation. With a start and destination RP chosen, the Navigation Module will feed this into the Pathfinding Engine that will utilize the A* algorithm to find the most optimal path to traverse. The Navigation Module then commences navigation for the user, updating them with the appropriate audio cues. Throughout navigation, the Navigation Module continuously queries the Hybrid Positioning System for the user's real-time location and heading. This data is interpreted by the Guidance Logic Module, which selects the proper choice of pre-recorded audio cues to be delivered through the device's speaker. This real-time data interpretation and feedback loop allows for turn-by-turn directions, allowing the user to have guidance navigating complex indoor environments.

3 Core Technology Analysis

3.1 Positioning Subsystem 1: Inertial Measurement Unit (IMU)-based Pedestrian Dead Reckoning (PDR)

This IMU-based PDR subsystem will sample the IMU's accelerometer and gyroscope to detect and interpret user movement. In order to isolate the human movement from the other sources of feedback from the sensors, the raw data of the accelerometer must be filtered before being interpreted (As of now, the exact method for filtering is still being explored in Literature Review). Additionally, this filtering will isolate the movement, such as walking, from a movement such as the user's arm or hand bouncing up and down. Once this data has been filtered, it can be interpreted in order to detect steps taken by the user. By looking for peaks in the magnitude of the acceleration signal, a step can be identified. However, to ensure that the magnitude is not set off by sudden movements or random actions to rapidly displace the device in use, there will be a calibrated threshold that will act as a barrier of confirmation. This threshold will have to be experimented on to find the most consistent results. An additional barrier of confirmation for the identified peaks in acceleration is that it should be followed by a period of low variance in magnitude, indicative of a step being taken. After determining steps, a step cycle can then be interpreted. This step cycle can represent a singular stride and this stride can have its length predicted based off of the minimum and maximum acceleration detected during a single step cycle. The heading calculation can then be made by utilizing the gyroscope's angular velocity data. This data will be integrated over a sampling interval to track changes in the heading (θ). This use of the gyroscope is subject to cumulative drift which is a key consideration for the accumulative error that the IMU will suffer from. This accumulative error is to then be corrected by the Extended Kalman Filter.

3.2 Positioning Subsystem 2: WiFi Fingerprinting

The WiFi Fingerprinting subsystem aims to be utilized as a correction measure against the accumulated error of the IMU-based PDR subsystem. Additionally, it will be used to check the user's estimated closest

RP based on proximity. The subsystem relies on the WiFi module included with the Raspberry Pi. By controlling the Raspberry Pi with a Python script, the WiFi Fingerprinting subsystem will execute wireless networking device scanning commands such as `iw dev wlan0 scan`. In doing so, the subsystem will be able to perceive visible wireless networking devices, identify them by their unique MAC addresses, and measure their Received Signal Strength Indicators (RSSI). This signal strength is measured in dBm and will be the core metric used for the fingerprint vectors to be stored and compared. This process of receiving and interpreting the RSSI will allow for the subsystem to take a real-time snapshot of the surrounding WiFi environment. However, there is a key consideration that signal strength can be inconsistent with many interruptions or interferences. In order to account for random interference, each sample of data collected for each RP to include its WiFi fingerprint will be a sample of data averaged over a set duration of at minimum 10 minutes. An additional consideration should be made that the level of interruption and interference can highly vary in these indoor areas, an example being human foot traffic along with the wireless devices that come with that foot traffic. In order to remedy this, the mapped indoor area will have its data collected at each RP at two time intervals. One of these time intervals will be during a period of high foot traffic, and the other being taken during a period of low foot traffic. Given that TERI is primarily targeted towards university campuses and buildings, a university schedule for classes and passing periods serves as an accurate marker to distinguish "high" and "low" traffic time periods. The WiFi Fingerprinting subsystem will then have localized data that contains information about "high" and "low" traffic times to accurately compare the proper data sets. In order to determine whether it is a "high" or "low" traffic time period, TERI will need to have some form of persistent clock. The Raspberry Pi's clock is not persistent without power, meaning an alternative is desired. A solution to this consideration is to use a Real-Time Clock (RTC) module that is battery-backed and attached to the Raspberry Pi, allowing the Raspberry Pi to read the time from the RTC module during its startup phase upon being switched on.

3.3 Data Fusion: Extended Kalman Filter (EKF)

The Extended Kalman Filter (EKF) is the algorithm that will fuse the two sources of data, being the IMU-based PDR and the WiFi fingerprinting subsystems, into a singular reliable position estimate. In the context of this project, the state we are tracking is a 2D position and a heading in the form of a vector. This vector can be represented as $P = [p_x, p_y, \theta]^T$ where P is a vector made up of an x-axis coordinate, a y-axis coordinate, and a degree measurement to represent the orientation of the user. The model will use the IMU-based PDR subsystem to predict what the expected user-state should be in real-time based off of the calculated step length estimate and the heading degree value. The model then corrects this prediction using the WiFi position estimate. This WiFi position estimate comes from the WiFi Fingerprinting subsystem, utilizing a k-Nearest Neighbors (k-NN) algorithm to determine the likely RP that the user is close to. This RP being validated is what will confirm or deny the prediction the model has made and corrections will be made reflecting the accuracy of the WiFi fingerprinting check. A key consideration is that the EKF incorporates a Process Noise Covariance Matrix and a Measurement Noise Covariance matrix that will represent the uncertainty of the IMU-based PDR subsystem and the uncertainty of the WiFi fingerprinting subsystem respectively. These matrices cannot be predetermined and must be tested and tuned during testing.

3.4 Navigation Guidance Module

The navigation map will be stored locally as a weighted, undirected graph (e.g. $G = (V, E)$), likely in a JSON file for easy parsing. Each vertex in V represents an RP with its absolute 2D coordinates. Each edge in E will represent a direct, walkable path between two RPs. The weight of these edges will be representative of the physical distance between the RPs to be measured in feet due to the targeted user-base to be American university students. Guidance provided by the Navigation Module is determined by vector mathematics. The system will calculate the "direction" a user is coming from based off of their current position and the next way point in their path. Then with this direction vector, a user's heading value can be taken and determine the difference in degrees of the user's chosen path and the suggested, calculated path. Corresponding audio files will be locally stored and will be played based off of the needed audio file. Additionally, all RPs will have a descriptor to define them as a key point that will play as an audio file as the user passes through the RP.