Seamless Navigation: Integration of GPS and UWB-Based Indoor positioning Systems for enhanced Localization for Mobile Robots

Kavin Palanichamy[1], Arkadiusz Perski[2], Rafal[3], Marcin[4]

[1] PIAP Lukasiewicz, Warsaw, Poland
kavin.wut@gmail.com

Abstract. This paper explores the fusion of Global Positioning System (GPS) and Ultra-Wideband (UWB) technology to address the challenges of indoor navigation. Traditional GPS systems struggle with accuracy in indoor settings due to signal obstructions, motivating the incorporation of UWB for centimeter-level precision. Our proposed integration seamlessly transitions between GPS and UWB, ensuring continuous and accurate positioning as users move between outdoor and indoor spaces.

Keywords: Indoor Navigation, Ultra-Wide Band, Mobile Robotics.

1 UWB based Indoor Positioning System

1.1 Introduction

UWB operates by transmitting short pulses of high-frequency radio waves, enabling precise and real-time location determination. In contrast to traditional GPS, UWB is particularly adept at overcoming challenges posed by indoor environments, such as signal interference and attenuation. Deploying UWB anchors strategically within a space allows for accurate triangulation and communication with UWB-equipped devices, enabling centimeter-level accuracy in tracking and positioning.

Fig. 1. DWM 10001-DEV by QORVO

1.2 Programming UWB modules by Qorvo

Qorvo's UWB modules are incorporated into the system, complete with firmware and accessible API featuring standard functions. Qorvo provides the necessary tools for individual programming, allowing modules to be configured as either tags or anchors based on local requirements. The modules can be tailored to specific roles, with tags designed for mobile objects and anchors stationed in the environment. Programming is facilitated through Qorvo's provided firmware and API.

Anchor parameters

Anchors are stationary devices strategically placed in the environment. They receive signals from tags and use the information to calculate the distance or location of the tags. The collective data from multiple anchors is then used to triangulate and determine the position of the tags. While programming the Anchor the following parameters are must:

i. Initiator Flag
ii. X, Y, Z measurements relative to Origin
iii. Quality factor

Tag parameters

A UWB module configured as a tag is usually attached to or integrated into a mobile object that needs to be tracked. Tags actively send signals that can be used by anchors to determine the tag's position. While programming the Anchor the following parameters are must:

i. Update Rate
ii. Quality factor
iii. Tag Flag

Fig. 2. Qorvo module connected to a Raspberry-pi with IO pins via UART.

1.3 Calculating the Local Co-Ordinate

Upon sending short pulses, the tag gauges Time of Flight, and the Qorvo DWM chips embedded onboard undertake the task of coordinate calculations. Retrieving these computed values is streamlined through the utilization of Qorvo's provided API. A program, executed on the Raspberry Pi, orchestrates the periodic retrieval of location data from the DWM chip using the API, which enables us to effectively process the data.

2 Network Structure

The central programs responsible for converting local coordinates to global and determining the optimal navigation mode based on a quality factor are executed on a Raspberry Pi, establishing a connection with the robot through an Ethernet cable. The Raspberry Pi transmits the data stream in NMEA format to the robot, which, in turn, forwards the stream to the console. The console then visualizes and displays the precise position on a high-resolution map.

The subsequent network map elucidates the configuration employed.

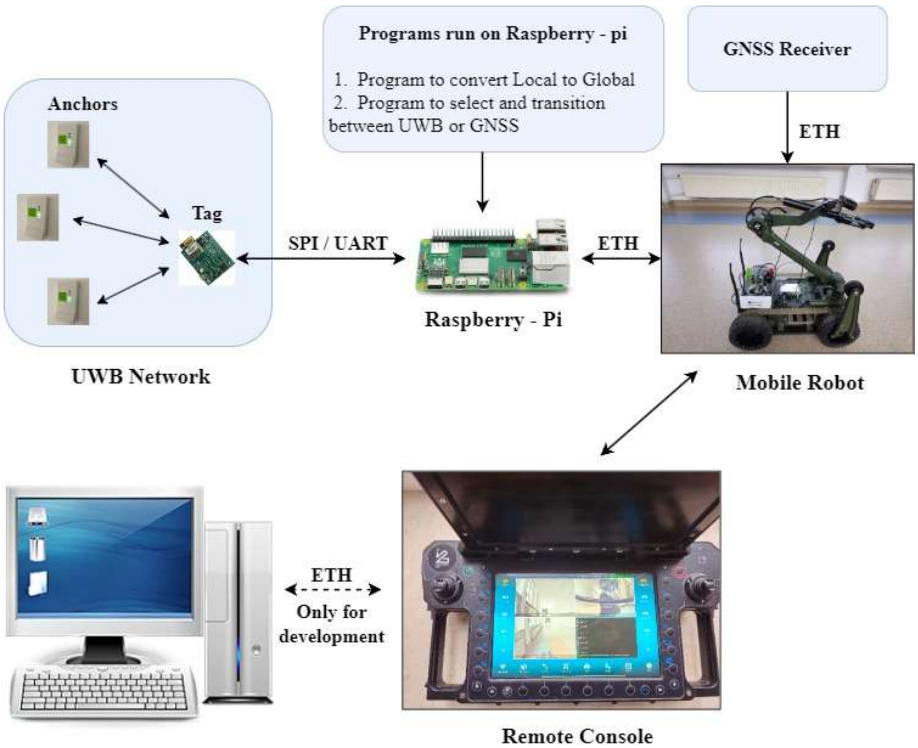

Fig. 3. Network Structure

3 Converting Local Coordinates to Global Coordinates

3.1 Need for Conversion – Map Display

To ensure a seamless representation of the robot's location on a specified map, the conversion of local coordinates data to global coordinates is essential. This transformation is accomplished through the application of the Haversine formula, enabling accurate positioning that can be effortlessly plotted on a digital map.

The parameters required for the conversion process are listed below:

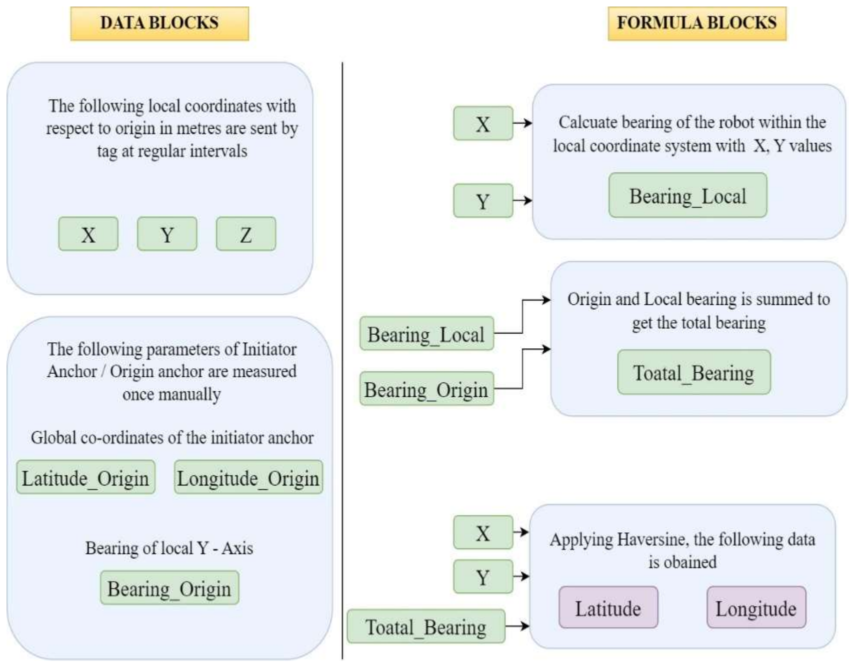

Fig. 4. Schema for converting Local to Global Co-Ordinates

3.2 Calculating bearing within local Co-ordinate system

$$BearingLocal = \tan^{-1}\frac{Y}{X} \quad (1)$$

Variable	Definition	Unit
Y	*Location of the mobile robot in Local Coordinate System along Y-axis.*	**m**
X	*Location of the mobile robot in Local Coordinate System along X-axis.*	**m**
BearingLocal	*The angle between local Y-Axis and the direction at which the robot*	**radians**

3.3 Calculating Total Bearing

The total bearing represents the robot's orientation in relation to Geographic North. It necessitates adjustment based on the specific quadrant in which the value is situated.

$$BearingTotal = BearingLocal + BearingOrigin \tag{2}$$

Variable	Definition	Unit
BearingTotal	*The angle between Geographic North and the direction at which the robot is heading.*	**radians**
BearingOrigin	*The angle between Geographic North and local Y axis*	**radians**
BearingLocal	*The angle between local Y-Axis and the direction at which the robot*	**radians**

3.4 Calculating Euclidean distance of the robot from origin

$$Displacement = \sqrt{X^2 + Y^2} \tag{3}$$

Variable	Definition	Unit
Y	*Location of the mobile robot in Local Coordinate System along Y-axis.*	**m**
X	*Location of the mobile robot in Local Coordinate System along X-axis.*	**m**
Displacement	*The Euclidean distance between the origin and the robot in local coordinate system*	**m**

3.5 Calculating Angular Distance

$$AngularDistance = \frac{Displacement\ from\ origin}{Earth\ radius} \tag{4}$$

Variable	Definition	Unit
Angular Distance	*The angle between two points on a sphere*	**radians**
Displacement from origin	*The Euclidean distance between the origin and the robot in local coordinate system*	**m**
Earth Radius (constant)	6378160	**m**

3.6 Haversine (Inverse) Formula

The modified version of Haversine formula can be used for calculating new coordinates based on an initial position (in this case it is the Coordinates of the Origin), distance traveled, and bearing. The input parameters are obtained from the onboard sensors and definition of initial parameters.

Latitude Calculation

$$LatB = \sin^{-1}(\sin(LatA) \times \cos(\theta) + \cos(LatA) \times \sin(\theta) \times \cos(\alpha)) \tag{5}$$

Variable	Definition	Unit
LatB	*The robot's latitude coordinate*	**degrees**
LatA	*The local Origin's latitude coordinate*	**degrees**
θ	*Angular Distance*	**radians**
α	*BearingTotal*	**radians**

Longitude Calculation

To address the periodicity of longitudes, an intermediary step involves the calculation of an adjusted longitude, ensuring it aligns with the correct quadrant.

$$dLon = \tan^{-1}\frac{\sin(\alpha)\times\sin(\theta)\times\cos(LatA)}{\cos(\theta)-\sin(LatA)\times\sin(LatB)} \tag{6}$$

$$LonB = mod(LonA + dLon + \pi, 2\pi) - 2\pi \tag{7}$$

Variable	Definition	Unit
LatB	*The robot's latitude coordinate*	**degrees**
LatA	*The local Origin's latitude coordinate*	**degrees**
LonB	*The robot's longitude coordinate*	**degrees**
LonA	*The local Origin's longitude coordinate*	**degrees**
dLon	*Intermediate Longitude*	**degrees**
θ	*Angular Distance*	**radians**
α	*BearingTotal*	**radians**

4 Switching Modes between GNSS and UWB

The default localization mode is GNSS signal, which serves as the primary method. However, when the tag identifies the presence of a stable and reliable anchor network with a high-quality factor, the system seamlessly transitions to the active mode, utilizing UWB (Ultra-Wideband). This approach ensures a straightforward and dependable localization process.

5 Testing Setup

The demonstration took place within a linear enclosed passageway, characterized by a weakened and unreliable GNSS signal strength. Toward the terminus of the corridor, an exit door leads to the outdoors. The robot was steered through the entire length of the corridor. A part of the corridor had UWB modules configured and setup.

Fig. 5. The green box indicates the part of the corridor with UWB coverage.

Fig. 6. UWB modules configured and mounted on the wall through the corridor.

Protocols for effective UWB Network configuration

(1) The modules are arranged symmetrically.
(2) The distance between each row was set precisely to 10 m.
(3) Some modules were placed at different heights for coverage of Z-axis

Setting up the hardware network

Fig. 7. The box on the left has raspberry-pi with the tag attached. The device to its right is a GNSS receiver. Both are connected to the robot with ethernet.

Fig. 8. Handheld console to navigate, visualize and monitor the program.

6 Results

In Fig. 9, The green circle denotes the trial positions of the robot during its movement through the corridor section covered by UWB. In contrast, the blue circle represents scattered points despite the robot remaining stationary indoors, relying on GNSS signal. The unreliability of the GNSS signal can be observed.

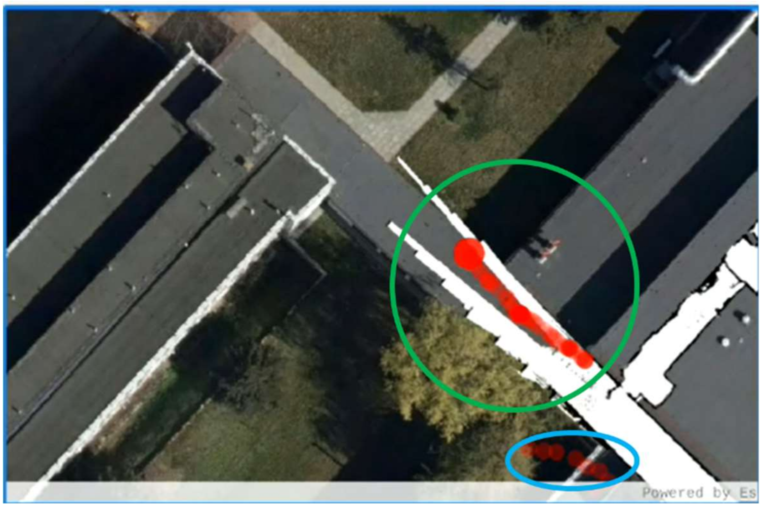

Fig. 9. Navigating the robot indoors using UWB (Green) and GNSS(Blue)

Upon exiting the corridor, the program seamlessly transitioned to the GNSS signal, known for its stability and reliability outdoors.

Fig. 10. The yellow circle signifies the robot's trial positions outdoors, monitored through GNSS signal.

References

1. Qorvo UWB based sensor link, https://www.qorvo.com/products/p/DWM1001-DEV
2. Haversine Formula, https://en.wikipedia.org/wiki/Haversine_formula