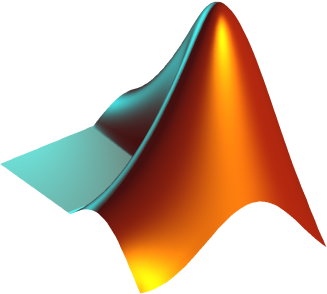
# Abstract

This document presents the detailed information about the program used to compare some of the localization algorithms. The program consists of a Graphical User Interface (GUI) for Matlab which easily set a real case scenario and presents the data gathered by Pozyx device mounted on an Arduino board. The GUI provides a pop-up menu where an algorithm can be chosen in order to compare the estimated position. Also, it may localize an agent in real time, tracking the position of the Arduino board.

# Prerequisites

As the program will provide a communication between the Arduino board and the computer, two parts will be considered in order to make the installation: the Matlab side and the Arduino side.



For the Matlab site it is required:

* Matlab R2016a (older versions may be valid)
* Arduino Support from Matlab: package to communicate with an Arduino. The installation steps are available at reference

<https://es.mathworks.com/hardware-support/arduino-matlab.html>

and is usually installed by double clicking the package download file.

For the Arduino site is required:

* Arduino Uno x1
* USB-B Cable x1
* Pozyx Shield for Arduino
* The Arduino IDE and Pozyx Arduino Library: the installation steps are provided at reference

<https://www.pozyx.io/Documentation/Tutorials/gettingStarted>

* Pozyx Anchor x[3, 6]
* USB-C Cable x[3, 6]

# Script Files

The localize.zip file is provided with the following files

HandbookLocalize.pdf: this file. Contains a description and a tutorial of the program

positioning\_arduino.ino the file to be uploaded onto the Arduino. See section Arduino for more details

# Arduino Setup

Before running the main Matlab file (localize.m), some restrictions about the scenario have to be taken into consideration:

First and most important, the identifiers of the anchors must be specified in the sketch positioning\_arduino.ino. The number of anchors must be comprised between three and six anchors. The external library does not support fewer or more anchors, so it is out of the scope of the program to work with another amount of anchors. This can be done by editing the following lines: METER LSI

Only one tag is possible. This is due to the calibration algorithm, which considers every tag as an anchor. The distance to the tag could be gotten as well as its coordinates. This means that the tag would behave indistinctly to an anchor. Consequently, only non-cooperative algorithms are possible.

The first anchor would define the origin of the coordinates map as the position (0,0). The second one would define the x axis and the third one the y axis, so it is desirable to put them as orthogonal as possible. This has to be taken into consideration if only positive coordinates and a more intuitive map are desired. Nevertheless, it is contemplated negative coordinates and not orthogonal axis. Again, section \ref{Pozyx:localize} is referred in order to get more information.

Coverage between all the devices involving the scenario is assumed. The Pozyx devices have a rangd coverage of about 100 meters in LOS conditions, so larger distances or other conditions have been not contemplated.

The process of the calibration is automatically done. In any case, the doc explains that the coordinates of the anchors can be manually set too if the coordinates are specified and some code is uncommented METER LSI

# Overview

The update button: similar to the ranging. Updates the available ports.

The calibrate button: it sends to the tag a command to automatically (by default) calibrate the position of the anchors. When the calibration is done the coordinates are plotted whit their anchor’s identification. Also, it gets the position of the tag during the specified samples according to the Pozyx algortihm, showing the current progress. After gathering these positions as (x,y) coordinates in millimeters, it plots its mean and shows the positions of the anchors, and also the scenario table is updated (see \ref{se:sc}).

The positioning button: same behaviour for the anchors, calibrating them again. But instead of showing the mean position of the anchor, shows its current position during these samples.

Stop button: stop the execution of the code.

# Control

# Scenario

# Map

# Anchor Data

# Algorithms Comparison

# Future Work

# Bibliography