Software Requirements Specification for 2D Localizer

Aliyah Jimoh

March 3, 2025

Contents

Revision History

Date	Version	Notes
2025/02/05	1.0	Initial Draft

1 Reference Material

This section records information for easy reference.

1.1 Table of Units

Throughout this document SI (Système International d'Unités) is employed as the unit system. In addition to the basic units, several derived units are used as described below. For each unit, the symbol is given followed by a description of the unit and the SI name.

Symbol	**Unit**	**SI**
m	length	metre
rad	angle	radians

1.2 Table of Symbols

The table that follows summarizes the symbols used in this document along with their units. The symbols are listed in alphabetical order.

Symbol	**Unit**	**Description**
$\mathbf{a_j}$	m	Position for beacon j
$\tilde{\mathbf{d}}$	m	Set of range measurements
$\tilde{d_j}$	m	Measured range with noise from beacon j
$g_j(\mathbf{x_i})$	m	Noisy range measurement of robot's position i taken from beacon j
$\boldsymbol{\mathcal{I}}$	$1/\mathrm{m}^2$	Total Fisher Information Matrix
i	–	Index for robot's positions
j	–	Index for number of beacons
N	–	Total number of beacons used
$\mathbf{T_{robot}}$	m	Pose of robot
$\mathbf{x_i}$	m	Position ith
$\hat{\mathbf{x}}$	m	Estimated position for robot
η_j	m	Sensor noise
θ	rad	Robot's orientation
σ_j^2	m	Noise variance for j

1.3 Abbreviations and Acronyms

Abbreviation/Acronym	Definition
2D	Two-Dimensional
2D Localizer	2D Localization Solution
A	Assumption
CRLB	Cramér-Rao Lower Bound
DD	Data Definition
FIM	Fisher Information Matrix
FM	Fiducial Marker
GD	General Definition
GS	Goal Statement
IM	Instance Model
LC	Likely Change
MLE	Maximum Likelihood Estimation
PDF	Probability Density Function
PS	Physical System Description
R	Requirement
SE(2)	Special Euclidean Group in 2D
SRS	Software Requirements Specification
TM	Theoretical Model
UC	Unlikely Change

1.4 Mathematical Notation

Throughout this document, there will be typographic conventions as well as mathematical operators that are used to distinguish different variables and operations.

Notation	Definition	Description
$\mathbf{A}$	Matrix	Bold capital letter
$\mathbf{a}$	Vector	Bold lowercase letter
a/A	Scalar	Italicized uppercase/lowercase symbol
$\| \ \|$	Euclidean (2) Norm	Vertical brackets

2 Introduction

Mobile robots have been used to traverse areas with various hazards, help collect data whether through its sensors or obtaining samples, and to overall complete challenging tasks. Due to their autonomy, there is no need to constantly monitor them as they have their own means with interacting with the environment. However, it could raise some concern when there is no reliable method to track their movements, especially if they are placed in a vastly large area or if they are difficult to access once they are operational. Risks such as having difficulty retrieving them if there is a malfunction keeping them from returning or possible collisions in the area or with other robots are reasons one would want to find a way to locate them as they carry out their tasks. The program being documented, 2D Localizer, proposes to solve this problem by developing a 2D localization solution that can implement various sensors to accurately localize the robots as they traverse the map provided.

The following section provides an overview of the Software Requirement Specification (SRS) for 2D Localizer. This section explains the purpose of this document, the scope of the requirements, the characteristics of the intended reader, and the organization of the document.

2.1 Purpose of Document

The main purpose of this document is to describe the requirements needed to run 2D Localizer. Information such as the constraints, assumptions and theoretical models used will be provided to help readers get a better understanding of the purpose and computations of this software. Therefore, it can be used as a reference guide on how to plan and set up the requirements needed by the user to get the desired and accurate results.

2.2 Scope of Requirements

The scope of the requirements includes the robot analyzed being in a controlled environment (seen as a 2D environment) to help with potential lighting problems for vision sensors used. The sensors used on the robot and environment are specified for modelling purposes. Some models can have joint formula due to sensors having independent measurements. The noise from the sensors will be considered zero-mean Gaussian for simplicity in calculations.

2.3 Characteristics of Intended Reader

Reviewers of this documentation should have taken a graduate course on linear algebra, estimation theory or matrix computation. They should also have some knowledge in probability.

2.4 Organization of Document

The organization of this document follows the template for an SRS for scientific computing software proposed by Smith and Lai (2005), Smith et al. (2007), and Smith and Koothoor

(2016). The presentation follows the standard pattern of presenting goals, theories, definitions, and assumptions. For readers that would like a more bottom up approach, they can start reading the data definitions and trace back to find any additional information they require.

The goal statements are refined to the theoretical models and the theoretical models to the instance models. The data definitions are used to support the definitions of the different models.

3 General System Description

This section provides general information about the system. It identifies the interfaces between the system and its environment, describes the user characteristics and lists the system constraints.

3.1 System Context

The system context is displayed in Figure 1 below. The circles represent the external aspects related to the software which are the users. The rectangle represents the software system being used (2D Localizer) and the arrows explain what information is being passed between the user and the software.

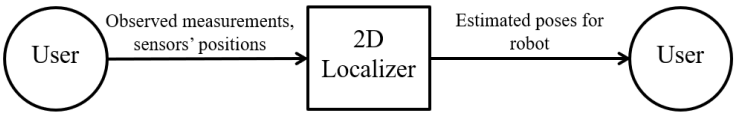

Figure 1: System Context

- User Responsibilities:
  - Provide inputs including the coordinates of each environmental sensor and the measurements taken from them.
  - Evaluate the inputs to ensure all are their respective types
- 2D Localizer Responsibilities:
  - Detect data type mismatch, such as a string of characters instead of a floating point number
  - Estimate the position and orientation of the robot

3.2 User Characteristics

The end user of 2D Localizer should have some familiarity with types of range sensors along with how to read and collect their data. They should also have basic experience with programming.

3.3 System Constraints

There are no system constraints.

4 Specific System Description

This section first presents the problem description, which gives a high-level view of the problem to be solved. This is followed by the solution characteristics specification, which presents the assumptions, theories, definitions and finally the instance models.

4.1 Problem Description

2D Localizer is intended to help keep track of mobile robots while carrying out their tasks.

4.1.1 Terminology and Definitions

This subsection provides a list of terms that are used in the subsequent sections and their meaning, with the purpose of reducing ambiguity and making it easier to correctly understand the requirements:

- **Pose:** Position and orientation of the robot.
- **Localization**: Determining where an object is with respect to its environment.
- **Fiducial Markers (FMs)**: Markers placed around the environment for the robot to determine its location.
- **Beacon**: A range sensor.

4.1.2 Physical System Description

The physical system of the localizer includes the following elements:

PS1: The mobile robot

PS2: The beacons placed in the environment

PS3: The camera sensors on the mobile robot

PS4: The fiducial markers in the environment

4.1.3 Goal Statements

Given the imported 2D map, coordinates of all sensors and fiducial markers in the environment, and the noisy measurements of the sensors, the goal statements are:

GS1: Calculate the estimated pose and error of the robot throughout its trajectory from sensors in the environment and on the robot.

GS2: Display a visual representation of the robot traversing the 2-D map with the sensors placed.

4.2 Solution Characteristics Specification

The instance models that govern 2D Localizer are presented in Subsection 4.2.5. The information to understand the meaning of the instance models and their derivation is also presented, so that the instance models can be verified.

4.2.1 Assumptions

This section simplifies the original problem and helps in developing the theoretical model by filling in the missing information for the physical system. The numbers given in the square brackets refer to the theoretical model [TM], general definition [GD], data definition [DD], instance model [IM], or likely change [LC], in which the respective assumption is used.

A1 : Robot uses camera sensors while the environment uses beacons and FMs

A2 : Localizer is used in a controlled environment (i.e., indoors)

A3 : Each sensor has independent measurements

A4 : Sensor noise is zero-mean Gaussian

4.2.2 Theoretical Models

This section focuses on the general equations and laws that 2D Localizer is based on.

Number	TM1
Label	**Noisy Range Measurement**
Equation	$$g_j(x) = \|\mathbf{x_i} - \mathbf{a_j}\| + \eta_j$$
Description	The equation above gives the noisy range measurement g_j (m) of the beacons placed in the environment (A1) where $\mathbf{x_i}$ is the position of the robot (m), $\mathbf{a_j}$ is the position of the beacon j (m), and $\eta_j \sim \mathcal{N}(0, \sigma_j^2)$ is the noise from the jth beacon (m) (A4).
Source	Sequeira et al. (2024)
Ref. By	DD1
Preconditions	A4
Derivation	None

Number	TM2
Label	**Fisher Information Matrix (FIM)**
Equation	$$\boldsymbol{\mathcal{I}}(\hat{\mathbf{x}}) \cong \sum_{j=1}^{N} \frac{1}{\sigma_j^2} \frac{(\hat{\mathbf{x}} - \mathbf{a_j})(\hat{\mathbf{x}} - \mathbf{a_j})^T}{\|\hat{\mathbf{x}} - \mathbf{a_j}\|^2}$$
Description	This equation gives the FIM which shows how much information the data provides about the robot's position σ_j^2 is the sensor variance used in the noise factor $\mathbf{a_j}$ is the position of beacon j (m), and $\hat{\mathbf{x}}$ is the estimated position in IM2 (m)
Source	Barfoot (2017)
Ref. By	TM3
Preconditions	A4, IM2

Derivation of FIM

TM2 was derived from GD1 and IM1 as the FIM formula from Barfoot (2017) is

$$\boldsymbol{\mathcal{I}}(\mathbf{x}|\boldsymbol{\theta}) = \mathbb{E}\left[\left(\frac{\partial \ln p(\mathbf{x}|\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right)^T \left(\frac{\partial \ln p(\mathbf{x}|\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right)\right]$$

where

$$\frac{\partial \ln p(\mathbf{x}|\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}$$

is the score function which measures how much the likelihood changes from the parameter being estimated $\boldsymbol{\theta}$ from observed data $\mathbf{x}$.

To change it to the software's symbols, the FIM would be

$$\boldsymbol{\mathcal{I}}(\mathbf{x}) = \mathbb{E}\left[\left(\frac{\partial \ln p(\tilde{\mathbf{d}}|\mathbf{x})}{\partial \mathbf{x}}\right)^T \left(\frac{\partial \ln p(\tilde{\mathbf{d}}|\mathbf{x})}{\partial \mathbf{x}}\right)\right]$$

Continuing with Barfoot's equations, their Gaussian Probability Density Function (PDF) can be made with GD1 since it requires a mean and variance. This can be shown as

$$p(\tilde{d}_j|\mathbf{x}) = \frac{1}{\sqrt{2\pi\sigma_j^2}} \exp\left(-\frac{(\tilde{d}_j - \|\mathbf{x} - \mathbf{a}_j\|)^2}{2\sigma_j^2}\right)$$

Linking it to IM1:

$$p(\tilde{\mathbf{d}}|\mathbf{x}) = \prod_{j=1}^{N} \frac{1}{\sqrt{2\pi\sigma_j^2}} \exp\left(-\frac{(\tilde{d}_j - \|\mathbf{x} - \mathbf{a}_j\|_2)^2}{2\sigma_j^2}\right)$$

From here, the log-likelihood function can be calculated:

$$\ln p(\tilde{\mathbf{d}}|\mathbf{x}) = \sum_{j=1}^{N} \left[-\frac{1}{2}\ln(2\pi\sigma_j^2) - \frac{(\tilde{d}_j - \|\mathbf{x} - \mathbf{a}_j\|_2)^2}{2\sigma_j^2}\right]$$

$$\ln p(\tilde{\mathbf{d}}|\mathbf{x}) = -\sum_{j=1}^{N} \frac{(\tilde{d}_j - \|\mathbf{x} - \mathbf{a}_j\|_2)^2}{2\sigma_j^2}$$

The score function can then be written as

$$\frac{\partial \ln p(\tilde{\mathbf{d}}|\mathbf{x})}{\partial \mathbf{x}} = \sum_{j=1}^{N} \frac{(\tilde{d}_j - \|\mathbf{x} - \mathbf{a}_j\|_2)}{\sigma_j^2} \frac{\mathbf{x} - \mathbf{a}_j}{\|\mathbf{x} - \mathbf{a}_j\|}$$

This function is in the expectation of FIM formula so when referring to DD1

$$\mathbb{E}\left[\tilde{d}_j - \|\mathbf{x} - \mathbf{a}_j\|_2\right] = \mathbb{E}\left[\eta_j\right] = 0$$

the noise has zero mean (A4). Cancelling that portion out gives us this model, also known as TM2

$$\boldsymbol{\mathcal{I}}(\mathbf{x}) \cong \sum_{j=1}^{N} \frac{1}{\sigma_j^2} \frac{(\mathbf{x} - \mathbf{a_j})(\mathbf{x} - \mathbf{a_j})^T}{\|\mathbf{x} - \mathbf{a_j}\|^2}$$

Number	TM3
Label	**Cramér-Rao Lower Bound (CRLB)**
Equation	$$Var(\hat{\mathbf{x}}) \geq \frac{1}{\boldsymbol{\mathcal{I}}(\hat{\mathbf{x}})}$$
Description	This equation gives the CRLB which shows how accurate an estimate of a parameter is given the noise in the measurements by providing a lower bound to the estimate's variance. We are able to use TM2 to get this lower bound $Var(\hat{\mathbf{x}})$ is the variance of the unbiased estimator in IM2 $\boldsymbol{\mathcal{I}}(\hat{\mathbf{x}})$ is the FIM calculated from TM2
Source	Barfoot (2017)
Ref. By	None
Preconditions	TM2, IM2
Derivation	None

Number	TM4
Label	**Special Euclidean (SE) 2 Transformation**
Equation	$$T_{robot} = \begin{bmatrix} \mathbf{R}(\theta) & \mathbf{t} \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & x \\ \sin\theta & \cos\theta & y \\ 0 & 0 & 1 \end{bmatrix}$$
Description	This equation shows SE(2) which represents all possible 2D transformations of the robot. Since the robot has cameras placed on it (A1), SE(2) can be used to get the position and orientation through the FMs. $\mathbf{R}(\theta)$ is the rotation represented as a 2×2 matrix $\mathbf{t}$ is the translation represented as a 2×1 matrix [0 1] makes sure that the matrix is homogenous x, y are the positions for the x and y coordinates of the 2D map
Source	Barfoot (2017)
Ref. By	None
Preconditions	None
Derivation	None

4.2.3 General Definitions

This section collects the laws and equations that will be used in building the instance models.

Number	GD1
Label	**Probability Density Function (PDF)**
SI Units	m^{-1}
Equation	$$p_j\left(\tilde{d}_j \vert \mathbf{x}\right) = \mathcal{N}\left(\Vert\mathbf{x} - \mathbf{a_j}\Vert, \sigma_j^2\right)$$
Description	This definition is about the Gaussian PDF (A4) being used in IM1. This provides the PDF of the actual noisy range measurement $\tilde{d}_j$ of beacon j (DD1) that is conditioned on the robot's position $\mathbf{x}$. $\mathcal{N}$ is the normal distribution with $\Vert\mathbf{x} - \mathbf{a_j}\Vert$ as the mean (predicted range) and σ_j^2 as the variance.
Source	Sequeira et al. (2024)
Ref. By	IM1

4.2.4 Data Definitions

This section collects and defines all the data needed to build the instance models. The dimension of each quantity is also given.

Number	DD1
Label	**Actual Range Measurement**
Symbol	$\tilde{d}_j$
SI Units	m
Equation	$\tilde{d}_j = \|\mathbf{x_i} - \mathbf{a_j}\| + \eta_j$
Description	$\tilde{d}_j$ is the actual beacon measurement that will be used compared to TM1 which is just a model. Although it would return a scalar from the sensor, the formulation the equation above. $\mathbf{x_i}$ is the position of the robot (m), $\mathbf{a_j}$ is the position of the jth beacon placed (m), and $\eta_j \sim \mathcal{N}(0, \sigma_j^2)$ is the noise from beacon j (m) (A4).
Sources	Sequeira et al. (2024)
Ref. By	GD1

4.2.5 Instance Models

This section transforms the problem defined in Section 4.1 into one which is expressed in mathematical terms. It uses concrete symbols defined in Section 4.2.4 to replace the abstract symbols in the models identified in Sections 4.2.2 and 4.2.3.

The goal GS1 is solved by IM1 and IM2.

Number	IM1
Label	**Joint Likelihood Function**
Input	$\tilde{\mathbf{d}}$, N
Output	$p(\tilde{\mathbf{d}}\|\mathbf{x})$
Equation	$$p\left(\tilde{\mathbf{d}}\|\mathbf{x}\right) = \prod_{j=1}^{N} p\left(\tilde{d_j}\|\mathbf{x}\right)$$
Description	This equation uses the measurements gotten from each beacon (A3) to calculate the product of how likely $\mathbf{x}$ is the best position to reflect the inputs where $\tilde{\mathbf{d}} = [\tilde{d_1}, \tilde{d_2}, \ldots, \tilde{d_N}]$, N is the number of beacons, and $p(\tilde{d_j}\|\mathbf{x})$ is the Gaussian PDF from GD1
Sources	Sequeira et al. (2024)
Ref. By	IM2

Number	IM2
Label	**Maximum Likelihood Estimation (MLE)**
Input	$p(\tilde{\mathbf{d}}\|\mathbf{x})$
Output	$\hat{\mathbf{x}}$
Equation	$$\hat{\mathbf{x}} = \arg\max_{x} p(\tilde{\mathbf{d}}\|\mathbf{x})$$
Description	Continuing from IM1, the MLE equation gives an estimate based on the maximum probability calculated from the joint likelihood function. $p(\tilde{\mathbf{d}}\|\mathbf{x})$ is the joint likelihood function from IM1
Sources	Sequeira et al. (2024)
Ref. By	TM2, TM3

4.2.6 Input Data Constraints

Table 5 shows the data constraints on the input output variables. The column for physical constraints gives the physical limitations on the range of values that can be taken by the variable. The column for software constraints restricts the range of inputs to reasonable values. The software constraints will be helpful in the design stage for picking suitable algorithms. The constraints are conservative, to give the user of the model the flexibility to experiment with unusual situations. The column of typical values is intended to provide a feel for a common scenario. The uncertainty column provides an estimate of the confidence with which the physical quantities can be measured. This information would be part of the input if one were performing an uncertainty quantification exercise.

Table 5: Input Variables

Var	Physical Constraints	Software Constraints	Typical Value	Uncertainty
$\mathbf{a_j}$	$\mathbf{a_j} > [0\ 0]$	-	[1.5, 2.6] m	-
$\tilde{d}_j$	$\tilde{d}_j > 0$	-	4.76 m	η_j

* The user will set up a series to insert rows of inputs.

4.2.7 Properties of a Correct Solution

A correct solution must exhibit

Table 7: Output Variables

Var	Physical Constraints
$\hat{\mathbf{x}}$	$\hat{\mathbf{x}} > [0\ 0]$

5 Requirements

This section provides the functional requirements, the business tasks that the software is expected to complete, and the nonfunctional requirements, the qualities that the software is expected to exhibit.

5.1 Functional Requirements

R1: Provide the inputs for 2D Localizer which include the 2D map, the coordinates of all sensors and markers located in the environment, and the measurements taken from the sensors.

R2: 2D Localizer will acquire $\hat{\mathbf{x}}$ from the inputs in R1.

R3: 2D Localizer will calculate the estimated posed from the measurements provided for IM2.

R4: 2D Localizer will verify that inputs provided are within their constraints.

R5: 2D Localizer will show a visual animated graph that tracks the mobile robots work while also displaying the sensors' coordinates.

5.2 Nonfunctional Requirements

This problem priorities the accuracy of the estimations along with its performance when it comes to keeping up with the robot's movements. That being said, the nonfunctional requirements are:

NFR1: **Accuracy**: The accuracy of the computed estimations should meet the threshold when comparing to the predicted measurements.

NFR2: **Understandability**: The software should be simple to understand and implement.

NFR3: **Maintainability**: The software should be simple modify some components when needed.

NFR4: **Usability**: The software should be easy to run and have minimal troubleshooting.

5.3 Rationale

This program has rationale for the assumptions mentioned in 4.2.1:

- A1: Assuming the type of sensors assisted in setting up the type of output variables that is needed for each set of estimates.
- A2: Referring to A1, vision sensors will be used on the robot and lighting could affect the way they detect FMs. Another main reason for this assumption is so that the coordinates of each beacon and FM can be structured around the area the user provides.
- A3: To find the sum and products of variables that need data from each beacon and FM placement (TM2, IM1), having each sensor collect independent data from different positions would help with the overall accuracy the program would want to achieve.

- A4: Having a Gaussian noise assumption simplifies computations in the program especially when some sensors used in robotics follow a Gaussian distribution.

6 Likely Changes

There are no likely changes to be made.

7 Unlikely Changes

UC1: A4 There are various models that are derived based on the Gaussian noise meaning that it would be detrimental to change.

8 Traceability Matrices and Graphs

The purpose of the traceability matrices is to provide easy references on what has to be additionally modified if a certain component is changed. Every time a component is changed, the items in the column of that component that are marked with an "X" may have to be modified as well. Table 9 shows the dependencies of theoretical models, general definitions, data definitions, and instance models with each other. Table 10 shows the dependencies of instance models, requirements, and data constraints on each other. Table 11 shows the dependencies of theoretical models, general definitions, data definitions, instance models, and likely changes on the assumptions.

	TM1	TM2	TM3	TM4	GD1	DD1	IM1	IM2
TM1								
TM2					X		X	X
TM3		X						X
TM4								
GD1								
DD1	X							
IM1					X	X		
IM2							X	

Table 9: Traceability Matrix Showing the Connections Between Items of Different Sections

	IM1	IM2	R1	R2	R3	R4	R5
IM1							
IM2	X						
R1							
R2			X				
R3		X					
R4							
R5							

Table 10: Traceability Matrix Showing the Connections Between Requirements and Instance Models

	A1	A2	A3	A4
TM1	X			X
TM2			X	X
TM3				
TM4	X	X		
GD1				X
DD1				X
IM1			X	
IM2				
UC1				X

Table 11: Traceability Matrix Showing the Connections Between Assumptions and Other Items

References

Timothy D. Barfoot. *State Estimation for Robotics*. Cambridge University Press, 2017.

Ethan Sequeira, Hussein Saad, Stephen Kelly, and Matthew Giamou. Towards Optimal Beacon Placement for Range-Aided Localization. *Proceedings of the Conference on Robots and Vision*, may 28 2024. https://crv.pubpub.org/pub/jl15869x.

W. Spencer Smith and Nirmitha Koothoor. A document-driven method for certifying scientific computing software for use in nuclear safety analysis. *Nuclear Engineering and Technology*, 48(2):404–418, April 2016. ISSN 1738-5733. doi: http://dx.doi.org/10.1016/j.net.2015.11.008. URL http://www.sciencedirect.com/science/article/pii/S1738573315002582.

W. Spencer Smith and Lei Lai. A new requirements template for scientific computing. In J. Ralyté, P. Ågerfalk, and N. Kraiem, editors, *Proceedings of the First International Workshop on Situational Requirements Engineering Processes – Methods, Techniques and Tools to Support Situation-Specific Requirements Engineering Processes, SREP'05*, pages 107–121, Paris, France, 2005. In conjunction with 13th IEEE International Requirements Engineering Conference.

W. Spencer Smith, Lei Lai, and Ridha Khedri. Requirements analysis for engineering computation: A systematic approach for improving software reliability. *Reliable Computing, Special Issue on Reliable Engineering Computation*, 13(1):83–107, February 2007.