Correspondenceless scan–to–map-scan matching of homoriented 2D scans for mobile robot localisation

Alexandros Filotheou\textsuperscript{a,1,*}

\textsuperscript{a}Agnostou Stratiotou 10, 54631, Thessaloniki, Greece

Abstract

The objective of this study is improving the location estimate of a mobile robot capable of motion on a plane and mounted with a conventional 2D LIDAR sensor, given an initial guess for its location on a 2D map of its surroundings. Documented herein is the theoretical reasoning behind solving a matching problem between two homoriented 2D scans, one derived from the robot’s physical sensor and one derived by simulating its operation within the map, in a manner that does not require the establishing of correspondences between their constituting rays. Two results are proved and subsequently shown through experiments. The first is that the true position of the sensor can be recovered with arbitrary precision when the physical sensor reports faultless measurements and there is no discrepancy between the environment the robot operates in and its perception of it by the robot. The second is that when either is affected by disturbance, the location estimate is bound in a neighbourhood of the true location whose radius is proportional to the affecting disturbance.

Keywords: robot localisation, scan–to–map-scan matching

1. Introduction

Mobile robot localisation in one plane is a well-studied field in robotics, and several diverse approaches have been proposed in the past. Probabilistic methods, e.g. the Kalman filter \cite{1}, or Monte Carlo Localisation (MCL) methods \cite{2, 3}, have been applied to the task of localisation and proved their success with respect to tracking accuracy, and their robustness with respect to sensor noise, discrepancies between the robot’s environment and its corresponding map, motion model mismatch with regard to the true kinematics of the robot, and pose uncertainty \cite{4}.

As for sensors, apart from encoders, Light Detection And Ranging devices (LIDARs) have become popular in robot localisation due to their measurement precision, real-time operability, and virtually no need for preprocessing.

In practice, due to an abundance of reasons—range scan measurements being corrupted by noise, the map the robot navigates in does not match the environment perfectly, the map is expressed as a finite resolution grid, noisy or faulty but ever-drifting odometry—the resulting localisation estimate is beset by an error which is often measured in centimetres or even decimetres \cite{5, 6}. Apart from the conceptual challenge of reducing this error, in certain conditions, such as industrial ones \cite{7, 8}, this order of magnitude of the estimate’s error is not acceptable, and therefore prosthetic methods have been employed in tandem with well-established sturdy probabilistic localisation methods, with most utilising the onboard pre-existing LIDAR sensors due to their aforementioned merits.

Such a method is introduced in \cite{8}: it uses (a) MCL combined with Kullback-Leibler Divergence (KLD) sampling \cite{9} as its base probabilistic localisation method, (b) scan–to–map-scan matching for improving the orientation estimate first, and then (c) a method that rests on the Discrete Fourier Transform (DFT) of the two now almost homoriented scans in order to improve the location estimate. However, the mathematical reasoning behind the deduction of the offset estimate between the location estimate and the robot’s real location accounts only for robot orientations equal to zero, which means that only when the robot’s heading is that of the map’s positive x-axis can the location estimate be corrected. What is more, the method aiming to recover the unknown (translation) offset is essentially a discrete-time controller whose objective is to stabilise the robot’s location estimate to the robot’s real pose, but no convergence or stability guarantees are given.

This paper focuses on the third step of the pipeline localisation method expressed in \cite{8} and, specifically, it aims to supplement it so that it is effective over all robot orientations, by deriving the necessary relations used in facilitating the extraction of an improved location estimate over the entire rotation space. What is more, this paper provides guarantees of the produced method’s convergence and stability by means of analysis resting on Lyapunovian notions of stability.

The proposed method’s natural adversaries are those
which can also extract the relative translation between homoriented scans: (a) traditional, ICP-based, scan-matching methods [10, 11, 12], and (b) correspondenceless and probabilistic approaches, such as the Normal Distributions Transform (NDT) scan-matcher [13, 14, 15, 16]. However, the former are subject to the perplexities delimitted by the underlying process of establishing correspondences between the two scans, and those posed by the plethora of parameters governing the accuracy of their behaviour, where these must be set by hand and address, for instance, sensor noise levels, outlier-rejection-related variables, or particularities of the ad-hoc environment in which scan-matching will be performed; namely: inefficient and ever-wanting tuning. Approaches that rest on the ND transform, on the other hand, operate by discretising the $x - y$ plane, and this fact limits their desirable accuracy and/or increases their execution time [17, 18]. In contrast to ICP methods, and similarly to the NDT method, the proposed method does not deal in correspondences. Furthermore, it requires no parameters to be tuned at all—apart from the maximum number of iterations, a parameter that trades accuracy for execution time, and a numerical threshold for stopping the process of iteration short (should the error reach a low enough level, which would make subsequent iterations redundant). That said, the proposed method runs in real time in modern processors. Furthermore, and most crucially, pitted against the best-performing, state-of-the-art, ICP-based scan-matching method, and the equally correspondenceless scan-matching method of NDT, the proposed method achieves better accuracy and increased robustness to sensor noise and map-to-real-world mismatch.

The remainder of the paper is structured as follows: Section 2 formulates the problem under purpose of solution and the solution’s objective, defines necessary notions, and finally provides a bibliographical exposition of the current state-of-the-art solutions to the problem of performing scan-to-map scan matching in order to improve the pose estimate of a range-sensor-mounted robot capable of motion in the 2D plane. Section 3 illustrates the method of solving the stated problem that this paper introduces. The theorems on which the method rests are proved, and then its algorithmical statement follows, accompanied with insights into its systemic operation, convergence, and meant stability. Section 4 presents the experimental setup: a benchmark dataset is used as the sole source and means of testing the proposed method against (a) the most accurate correspondence-finding state-of-the-art method, and (b) the equally correspondenceless approach of scan-matching via NDT. Additionally, it provides a characterisation of the proposed method in conditions arising from reality. Finally, section 5 offers a recapitulation.

## 2. The overall problem & current solutions

This section offers the formulation of the overall problem and objective aimed at considered in this study (subsection 2.1), necessary definitions that will be useful hereafter (subsection 2.2), and a collection of the considered problem’s current solutions (subsection 2.3).

### 2.1. Problem and objective formulation

**Problem 1.** Let a mobile robot capable of motion in the $x - y$ plane be equipped with a coplanarily mounted range scan sensor, whose pose with respect to the robot’s frame of reference is fixed, known, and of the same orientation. Let also at time $t \geq 0$ the following be available or standing:

- The map $M$ of the environment the robot operates in
- A range scan $S_t^r$, captured from its range scan sensor’s (unknown and sought for) pose $p_t(l_t, \theta_t)$, $l_t = (x_t, y_t)$
- An initial estimate of the range scan sensor’s pose $\hat{p}_t(l_t^0, \hat{\theta}_t)$, where $l_t^0 = (\hat{x}_t^0, \hat{y}_t^0)$ is in a neighbourhood of $l_t$, expressed in the map’s frame of reference
- $|\theta_t - \hat{\theta}_t| = 0$, i.e. the real and estimated poses are homoriented

Then, the objective is to reduce the 2-norm of the sensor’s location error $e(l_t, \hat{l}_t) \triangleq l_t - \hat{l}_t$ from its initial value

$$
\|e(l_t, \hat{l}_t^0)\|_2 = (\|x_t - \hat{x}_t^0\|^2 + \|y_t - \hat{y}_t^0\|^2)^{1/2}
$$

by improving the sensor’s location estimate to $\hat{l}_t(\hat{x}_t^0, \hat{y}_t^0)$ so that

$$
\|e(l_t, \hat{l}_t^0)\|_2 < \|e(l_t, \hat{l}_t^0)\|_2
$$

Assuming that the sensor’s pose with respect to the robot’s frame of reference is fixed at all times (as is customary in mobile robotics), this improvement in the sensor’s pose equals that of the robot’s pose with respect to the map’s frame of reference.

### 2.2. Definitions

**Definition 1.** Definition of a range scan captured from a 2D LIDAR sensor A 2D LIDAR sensor captures finitely-many ranges, i.e. distances to objects within its range, on a horizontal cross-section of its environment at regular angular and temporal intervals over a defined angular range [19]. We define a range scan $S$, consisting of $N$ rays over an angular range $\lambda$, to be an ordered sequence of $N$ pairs of (a) one range measurement and (b) one angle, i.e. the ray’s angle relative to the sensor’s heading, expressed in the sensor’s frame of reference, ordered by increasing angle:

$$
S \equiv \{(d_n, -\lambda_n \frac{\lambda}{2} + \frac{\lambda n}{N})\}, n = \{0, 1, \ldots, N - 1\}
$$
Remark II. The solution to (1) cannot, in general, be found in closed form due to the arbitrary nature of the nonlinearity of the “$\oplus$” operator.

Definition II. Scan-to-scan matching using a 2D LIDAR sensor (adapted for use in two dimensions from [12]) Let two range scans as defined by Definition I, $S_r$ and $S_v$, be captured from a LIDAR sensor operating in the same environment at both capturing times. Let $p_v(x_v, y_v, \theta_v)$ be the pose from which the sensor captured $S_v$, expressed in some coordinate system (usually a past pose estimate of the sensor). The objective of scan-to-scan matching in two dimensions is to find the roto-translation $q = (t, \theta)$, $t = (\Delta x, \Delta y)$ that minimises the distance between the endpoints of $S_v$ roto-translated by $q$ to their projection on $S_r$. Denoting the endpoints of $S_v$ by $\{p_v^i\}$, in formula:

$$\min_q \sum \|p_v^i \oplus q - \prod\{S_r, p_v^i \oplus q\}\|^2$$  \hspace{1cm} (1)

The symbol “$\oplus$” denotes the roto-translation operator $p_v^i \oplus (t, \theta) \triangleq R(\theta)p_v^i + t$, where $R(\theta)$ is the 2D rotation matrix for argument angle $\theta$, and $\prod\{S_r, p_v^i \oplus q\}$ denotes the Euclidean projector on $S_r$.

Remark II. The solution to (1) cannot, in general, be found in closed form due to the arbitrary nature of $S_v$ and the nonlinearity of the “$\oplus$” operator.

Remark III. Scan-to-scan matching is employed in robotics as a form and means of odometry, primarily in non-wheeled robots where no encoders can be utilised, or as a useful ameliorator of the ever-drifting encoder-ed odometry: scans captured at consecutive time instances, inputted to a scan-matching algorithm, convey an estimate as to the pose of the scan sensor at the second capture time relative to that captured first. It is being successfully employed in the tasks of Simultaneous Localisation and Mapping [20, 21, 22], local map construction [23, 24, 25], or in people-tracking systems [26].

Definition III. Definition of a map-scan A map-scan is a virtual scan that encapsulates the same pieces of information as a scan derived from a physical sensor; only their underlying operating principle is different due to the fact the map-scan refers to distances to obstacles within the map of the robot’s environment (hence its virtuality) rather than within the environment itself. A map-scan is captured from a virtual sensor whose pose relative to the robot’s virtual frame of reference is the same as that of the physical sensor relative to the real robot’s frame of reference, and derived by means of locating intersections of rays emanating from the estimate of the virtual sensor’s pose with boundaries demarcating obstacles in the map.

Definition IV. Scan-to-map-scan matching in two dimensions Scan-to-map-scan matching is defined in the same way as scan-to-scan matching but with $S_v$ now derived not from the physical environment of the robot but from its map. This subtle difference makes $p_v$, the pose from which the map-scan $S_v$ was captured (Definition II), continually expressible in the map’s frame of reference, and therefore in absolute terms, rather than relative to its previous estimate (recursively relative to a convention of the robot’s starting pose).

Remark IV. The advantage of matching a scan derived from a physical sensor from its actual pose and a map-scan derived from a virtual sensor from its estimated pose comes now into light: Assume that robot localisation is performed on a mobile robot equipped with a 2D range-scan sensor via some localisation method which produces its pose estimate; assuming that the range sensor is fixed at the same pose relative to the robot in both real and virtual environments, the roto-translation of the virtual scan’s endpoints that minimises their distance to their projection on the physical scan equals the roto-translation that, when applied to the robot’s estimated pose will minimise its distance to its real pose. Therefore, extracting the relative roto-translation of the virtual scan with respect to the real scan can be used as a correction of the localisation estimate of the robot’s pose within the map, thereby making the reduction of the inevitable localisation error inherent to localisation approaches (either probabilistic or other) conditionally possible.

Definition V. Admissible sensor location estimates & admissible estimate errors Admissible sensor location estimates are those whose line of sight to the true sensor location are uninterrupted by the map—e.g. the line segment that connects the two does not intersect obstacles it. The estimate errors of admissible sensor location estimates are called admissible estimate errors. The term “in a neighbourhood of the sensor’s true location” or variations of it implies that all location estimates interior to that neighbourhood are admissible.

2.3. State-of-the-art approaches

The bibliography on solving Problem I is anything but vast, and, in their majority, state-of-the-art solutions do
not require that the estimated and true poses are aligned with regard to orientation.

For example, in [27] an elementary stochastic search algorithm is employed to correct the robot’s translational and rotational pose error due to its inevitable odometric drift. This auxiliary localisation behaviour is activated whenever an error measure that is based on the relative deviation in detected distances between rays from a real scan and a map-scan is found to be above a preset threshold. To avoid having to correct for the motion of the robot while scan-matching, the robot is assumed to be standing still for the whole duration of its pose correction. Therefore, whenever the error measure is found to be above its preset threshold, the algorithm halts the robot’s motion, and picks a random pose in the neighbourhood of its estimated pose. It then takes a virtual range scan from that pose, and computes the new error. If the error is lower than the one found for the previous estimated pose, a new iteration starts, this time centered around the newly found pose; if not, the algorithm keeps guessing poses until it finds one whose error is lower than the previous one. The final pose is then taken as the true pose of the robot, allowing for a correction of the odometry. Experiments performed with this method showed that it was able to correct a radial pose error of 0.3 m to 0.07 m, and an angular pose error of 0.393 rad to 0.01 rad.

By contrast, in order to solve Problem I, the authors of [8] assume that the real and the virtual scans have already been angularly aligned through the use of scan-to-map-scan matching, and specifically through the all-encompassing, highly accurate, efficient, and outperformer of the state-of-the-art scan-matchers: PL-ICP [12]. Their findings indicate that the improvement of the location estimate through scan-matching is unstable, and therefore that utilising ICP-variants in order to extract the relative translation between LIDAR-extracted scans and map-scans is precarious and hence unsuitable in the context of localisation of autonomous forklifts in industrial warehouse settings where milli-meter accuracy is required. In the technical report they outline how at each localisation step they (a) acquire a forklift’s pose estimate through the use of MCL with KLD sampling, (b) compute the relative rotation between that step’s real scan and map-scan given the forklift’s pose estimate through the use of scan-to-map-scan matching, and then, given that the forklift’s orientation error has decreased to as much as 0.13° (0.0023 rad), they (c) correct the displacement error by iteratively performing scan-to-map-scan matching through a process that approximates the displacement error at each iteration by a function of the first element of the Discrete Fourier transform of the difference in ranges between that step’s real scan and that iteration’s map-scan. However, in their proof of how this process is feasible and convergent, the implication of a non-zero robot orientation is missing, which, as we shall see in subsection 3.1 is actually a non-trivial matter, since incorporating it leads to a result different to that of simply rotating the location estimate vector by the robot’s orientation.

A similar pipeline is presented in [28]. Instead of using PL-ICP off-the-shelf, the authors develop a scan-matching algorithm that also aligns real scans with map-scans taken from MCL’s estimate pose using the Gauss-Newton method during their optimisation of scan-alignment, but they do so layer-by-layer in increasing map resolution. Experiments conducted with a real robot in unstructured environments show that the scan-to-map-scan matcher achieves an average location accuracy of 0.017 m and an average rotation accuracy of 0.0095 rad. Interestingly, by feeding back the improved estimate to MCL in the form of one discrete particle they manage to decrease the location estimate error even further to achieve milli-meter accuracy.

Likewise, in [29], a matching algorithm that deals in range scan features is introduced. It works by detecting rotation- and translation-invariant features that are only computable in real-time (such as extreme values in the polar representation of a range scan) in both real and virtual scans before establishing correspondences between them. The roto-translation between the two is then computed as the optimal transformation for mapping the latter’s features to the former’s.

In [30] scan-to-map-scan matching is employed in tandem with a particle filter. From the pose estimate of the latter, a map-scan is computed and then matched against the range scan captured from the physical sensor using PLICP. Feeding back the resulting pose estimate to the population of the particle filter in the form of a multitude of particles is shown to exhibit lower pose errors compared to [28], where the resulting pose estimate is fed back in the form of only one particle. Furthermore it is shown that this method of feedback exhibits increased robustness compared to [8], where the particle filter is initialised anew around the resulting estimate.

3. The proposed method

We first study the unreal situation where no disturbances are acting on the range measurements of either the real or the virtual scan. This is performed not for theoretical reasons but because through it we establish robust stability of the proposed solution to Problem I in real conditions where disturbances are present in both scans.

3.1. Without disturbances

**Theorem I.** Let the assumptions of Problem I hold at time $t$, and the angular range of the range scan sensor be $\lambda = 2\pi$. Let a map-scan, denoted by $S_v^M(p_t)$, be captured from $p_t$ within map $M$. Assume that both $S_v(p_t)$ and $S_v^M(p_t)$ range scans are disturbance-free, that is, the distances to obstacles the rays of the real scan capture correspond to the true distance of the sensor to said obstacles, and that the map of the environment captures the latter perfectly. Then, treating the estimate of the location of the
sensor \( \hat{\mathbf{l}}_t = [\hat{x}_t, \hat{y}_t]^\top \) as a state variable and updating it according to the difference equation

\[
\hat{\mathbf{l}}_t[k + 1] = \hat{\mathbf{l}}_t[k] + \mathbf{u}[k]
\]  

where \( \hat{\mathbf{l}}_t[0] = \hat{\mathbf{l}}_0^0 = [\hat{x}_0^0, \hat{y}_0^0]^\top \), i.e. the supplied initial location estimate,

\[
\mathbf{u}[k] = \frac{1}{N} \begin{bmatrix} \cos \hat{\theta}_t & \sin \hat{\theta}_t \\ \sin \hat{\theta}_t & -\cos \hat{\theta}_t \end{bmatrix} \begin{bmatrix} X_{1,r}(\mathbf{S}_t(\mathbf{p}_t), \mathbf{S}_v^M(\hat{\mathbf{p}}_t[k])) \\ X_{1,i}(\mathbf{S}_t(\mathbf{p}_t), \mathbf{S}_v^M(\hat{\mathbf{p}}_t[k])) \end{bmatrix}
\]

is the two-dimensional vector hereafter referred to as the control vector, \( X_{1,r}(\cdot) \) and \( X_{1,i}(\cdot) \) are, respectively, the real and imaginary parts of the complex quantity \( X_1 \):

\[
X_1(\mathbf{S}_t(\mathbf{p}_t), \mathbf{S}_v^M(\hat{\mathbf{p}}_t[k])) = X_{1,r}(\mathbf{S}_t(\mathbf{p}_t), \mathbf{S}_v^M(\hat{\mathbf{p}}_t[k])) + i X_{1,i}(\mathbf{S}_t(\mathbf{p}_t), \mathbf{S}_v^M(\hat{\mathbf{p}}_t[k]))
\]

\[
= \sum_{n=0}^{N-1} (d_n^r - \tilde{d}_n^r) \cdot e^{-i \frac{2\pi n}{N}}
\]

where \( d_n^r \) and \( d_n^i \) are, respectively, the ranges of the \( n \)-th ray of the real \( \mathbf{S}_t(\mathbf{p}_t) \) and virtual \( \mathbf{S}_v^M(\hat{\mathbf{p}}_t[k]) \) scans, and \( \hat{\mathbf{p}}_t[k] = (\hat{\mathbf{l}}_t[k], \hat{\theta}_t) \) then \( \hat{\mathbf{l}}_t[k] \) converges to \( \mathbf{l}_t \) uniformly asymptotically as \( k \to \infty \).

The proof is found in subsection 3.3.

In practice, the control system (2) is let to iterate either until the norm of the control vector \( \mathbf{u}[k] \) reaches a sufficiently small quantity: \( ||\mathbf{u}[k]||_2 \leq \varepsilon_u \), where \( \varepsilon_u \) is sufficiently small—e.g., \( \varepsilon_u < 10^{-3} \), or for \( k_{\text{max}} \geq 0 \) iterations (a sufficiently large, externally-supplied maximum iterations threshold—e.g., \( k_{\text{max}} \geq 20 \)). Therefore, if we denote by \( k_{\text{stop}} \in [0, k_{\text{max}}] \) the last index of iteration, and by \( \hat{\mathbf{l}}_t^* = \hat{\mathbf{l}}_t[k_{\text{stop}}] \) the final estimate of the sensor’s location, and by \( B \) the ultimate bound of the error. If \( ||\mathbf{e}(\mathbf{l}_t, \hat{\mathbf{l}}_t^*)||_2 > B \), Theorem II guarantees the satisfaction of objective (7) if \( k_{\text{stop}} \geq k_0 \). If, on the other hand, \( ||\mathbf{e}(\mathbf{l}_t, \hat{\mathbf{l}}_t^*)||_2 \leq B \), it is not certain that \( ||\mathbf{e}(\mathbf{l}_t, \hat{\mathbf{l}}_t^*)||_2 < ||\mathbf{e}(\mathbf{l}_t, \hat{\mathbf{l}}_t^*)||_2 \), and therefore objective (7) is attained.

3.2. With disturbances

**Theorem II.** Let the assumptions of Problem 1 hold at time \( t \), and the angular range of the range scan sensor be \( \lambda = 2\pi \). Let a map-scan, denoted by \( \mathbf{S}_v^M(\hat{\mathbf{p}}_t) \), be captured from \( \hat{\mathbf{p}}_t \) within map \( M \). Assume that the ranges of both \( \mathbf{S}_t(\mathbf{p}_t) \) and \( \mathbf{S}_v^M(\hat{\mathbf{p}}_t) \) range scans are affected by additive, bounded disturbances. Then, treating the estimate of the location of the sensor \( \mathbf{l}_t = [x_t, y_t]^\top \) as a state variable and updating it according to the difference equation

\[
\hat{\mathbf{l}}_t[k + 1] = \hat{\mathbf{l}}_t[k] + \hat{\mathbf{u}}[k]
\]

where \( \hat{\mathbf{l}}_t[0] = \hat{\mathbf{l}}_0^0 = [\hat{x}_0^0, \hat{y}_0^0]^\top \), i.e. the supplied initial location estimate,

\[
\hat{\mathbf{u}}[k] = \frac{1}{N} \begin{bmatrix} \cos \hat{\theta}_t & \sin \hat{\theta}_t \\ \sin \hat{\theta}_t & -\cos \hat{\theta}_t \end{bmatrix} \begin{bmatrix} \bar{X}_{1,r}(\mathbf{S}_t(\mathbf{p}_t), \mathbf{S}_v^M(\hat{\mathbf{p}}_t[k])) \\ \bar{X}_{1,i}(\mathbf{S}_t(\mathbf{p}_t), \mathbf{S}_v^M(\hat{\mathbf{p}}_t[k])) \end{bmatrix}
\]
Here we make the observation that \(-x_n - x_s\) and \(y_n - y_s\) are, respectively, the real and imaginary parts of the complex quantity
\[
d_n e^{-i(\theta + \frac{2\pi n}{N})} = d_n \cos(\theta + \frac{2\pi n}{N}) - i \cdot d_n \sin(\theta + \frac{2\pi n}{N})
\]
and, therefore, that
\[
d_n e^{-i\frac{2\pi n}{N}} = e^{ith}(-x_n - x_s) + i \cdot (y_n - y_s) \tag{9}
\]
and, therefore, that
\[
d_n e^{-i\frac{2\pi n}{N}} = e^{ith}(-x_n - x_s) + i \cdot (y_n - y_s) \tag{10}
\]
Ergo, denoting with the superscript \(r\) quantities which correspond to the real scan \(S_r\), which has been captured from the unknown sensor pose \(p(x'_r, y'_r, \theta'_r)\), and with \(v\) those which correspond to the virtual scan \(S_v\), which has been captured from pose \(p(x'_v, y'_v, \theta'_v)\), where \(\theta'_r = \theta'_v = \theta\):
\[
d_n e^{-i\frac{2\pi n}{N}} = e^{ith}(-x'_n - x'_s) + i \cdot (y'_n - y'_s) \tag{11}
\]
\[
d_n e^{-i\frac{2\pi n}{N}} = e^{ith}(-x'_n - x'_s) + i \cdot (y'_n - y'_s) \tag{12}
\]
The first term of the Discrete Fourier Transform of the signal that consists of the difference of the two signals (11) and (12) is
\[
X_1 = \sum_{n=0}^{N-1} (d_n^r - d_n^v) \cdot e^{-i\frac{2\pi n}{N}}
\]
\[
= \sum_{n=0}^{N-1} (d_n^r e^{-i\frac{2\pi n}{N}} - d_n^v e^{-i\frac{2\pi n}{N}})
\]
\[
(11) \leq (12) \sum_{n=0}^{N-1} (e^{ith}(-x'_n - x'_s) + i \cdot (y'_n - y'_s))
\]
\[
- e^{ith}(-x'_n - x'_s) + i \cdot (y'_n - y'_s))
\]
\[
= \sum_{n=0}^{N-1} ((-x'_n - x'_s) + i \cdot (y'_n - y'_s))
\]
\[
- ((-x'_n - x'_s) + i \cdot (y'_n - y'_s))
\]
\[
e^{ith} \sum_{n=0}^{N-1} (-x'_n - x'_s) + i \cdot \sum_{n=0}^{N-1} (y'_n - y'_s)
\]
\[
e^{ith} \sum_{n=0}^{N-1} (x'_n - x'_s) + i \cdot \sum_{n=0}^{N-1} (y'_n - y'_s)
\]
\[
= e^{ith} \sum_{n=0}^{N-1} (x'_n - x'_s) + i \cdot \sum_{n=0}^{N-1} (y'_n - y'_s)
\]
\[
+ e^{ith} \sum_{n=0}^{N-1} (x'_n - x'_s) + i \cdot \sum_{n=0}^{N-1} (y'_n - y'_s)
\]
\[
= e^{ith} \sum_{n=0}^{N-1} (x'_n - x'_s) + i \cdot \sum_{n=0}^{N-1} (y'_n - y'_s)
\]
(13)
In the first summand of (13), the quantities under summation express the \(x\)-wise and \(y\)-wise offsets of the endpoint of the virtual scan’s \(n\)-th ray from the real scan’s \(n\)-th ray, expressed in the map’s frame of reference. In general, they cannot be calculated in closed form since the sensor’s true location is unknown, and the environment—even if it is of absolute fidelity to the map—is arbitrary. For posterior convenience we denote the first summand of (13) by
\[
e^{ith} \sum_{n=0}^{N-1} (x'_n - x'_s) + i \cdot \sum_{n=0}^{N-1} (y'_n - y'_s)
\]
\[
= N e^{ith}(-\delta_x + i \cdot \delta_y) \tag{14}
\]
where
\[
\delta_x = \frac{1}{N} \sum_{n=0}^{N-1} (x'_n - x'_s)
\]
\[
\delta_y = \frac{1}{N} \sum_{n=0}^{N-1} (y'_n - y'_s)
\]
(15)
As regards the second summand of (13), let us denote the components of the error between the true location \((x'_r, y'_r)\) and the estimated one \((x'_v, y'_v)\) by \(e_x = x'_r - x'_s\) and \(e_y = y'_r - y'_s\). The quantities under summation in the second summand of the right-hand side of (13) do not depend on \(n\), and therefore
\[
\sum_{n=0}^{N-1} (x'_n - x'_s) = \sum_{n=0}^{N-1} e_x = N \cdot e_x \tag{16}
\]
\[
\sum_{n=0}^{N-1} (y'_n - y'_s) = \sum_{n=0}^{N-1} e_y = N \cdot e_y \tag{17}
\]
Then, equation (13) is transformed as follows:
\[
X_1 (14),(17),(18) = N e^{ith}(e_x - \delta_x - i \cdot (e_y - \delta_y)) \tag{19}
\]
Since \(X_1\) is, in general, complex, it can be written in the form \(X_1 = X_{1,r} + i \cdot X_{1,i}\), where \(X_{1,r}\) and \(X_{1,i}\) are known quantities \((X_1\) is the first term of the DFT of the difference between sequences of known real numbers). Then, from equation (19), the following equality is established:
\[
X_1 = N e^{ith}(e_x - \delta_x - i \cdot (e_y - \delta_y)) = X_{1,r} + i \cdot X_{1,i}
\]
from which the expressions for the positional errors between the estimated and true location of the sensor \(e_x\) and \(e_y\) can be derived:
\[
\begin{bmatrix}
e_x \\
e_y
\end{bmatrix} = \frac{1}{N} \begin{bmatrix}
\cos \theta & \sin \theta \\
-\sin \theta & \cos \theta
\end{bmatrix} \begin{bmatrix}X_{1,r} \\
X_{1,i}
\end{bmatrix} + \begin{bmatrix}\delta_x \\
\delta_y
\end{bmatrix}
\]
\[
= \begin{bmatrix}u_x \\
u_y
\end{bmatrix} + \begin{bmatrix}\delta_x \\
\delta_y
\end{bmatrix}
\]
(20)
and this is the positional error between the true and estimated poses of the sensor. Since
\[
\begin{bmatrix}e_x \\
e_y
\end{bmatrix} = \begin{bmatrix}x'_r \\
y'_r
\end{bmatrix} - \begin{bmatrix}x'_s \\
y'_s
\end{bmatrix}
\]
and therefore

\[
\begin{bmatrix} x'_u \\ y'_u \end{bmatrix} + \begin{bmatrix} e_x \\ e_y \end{bmatrix} = \begin{bmatrix} x'_e \\ y'_e \end{bmatrix}
\]

adding (20) to the estimated position of the sensor would allow us to extract its true position in one step. However, although the first summand of the right-hand side of (20) does consist of known quantities, the second one does not, and therefore it is impossible to extract the sensor’s true position in one step. This is the reason an iterative process of updating the sensor’s location estimate is necessary.

Let us now denote by \( \hat{\mathbf{u}}[k] = [x'_u[k], y'_u[k]] \) the estimated location of the range sensor at iteration \( k \), and by \( \mathbf{u}[k] = [x_u[k], y_u[k]] \). The claim of Theorem I is that by updating the location estimate with

\[
\hat{\mathbf{u}}[k + 1] = \hat{\mathbf{u}}[k] + \mathbf{u}[k]
\]

when \( \hat{\mathbf{u}}[0] = [x'_u[0], y'_u[0]] \) is set to the initially supplied location estimate, the location estimate converges asymptotically to the sensor’s true pose \([x'_u, y'_u] \) as \( k \to \infty \).

In order to investigate the convergence and stability of (21), we first rewrite it so that it reflects the dynamics of the error between the estimated range sensor location and the true one. We denote this error by \( \mathbf{e} = [e_x, e_y] \). Then, by simply multiplying both sides of (21) with \( -1 \) and adding \([x'_e, y'_e] \) to both sides, (21) is transformed to:

\[
\mathbf{e}[k + 1] = \mathbf{e}[k] - \mathbf{u}[k]
\]

We shall begin by examining the \( x \)-wise component of the error (the analysis regarding the \( y \)-wise component is analogous) and state the two main equations available:

\[
e_x[k + 1] = e_x[k] - u_x[k] \tag{23}
\]

\[
e_x[k] = u_x[k] + \delta_x[k] \tag{24}
\]

from which we obtain that \( \delta_x[k] = e_x[k + 1] \).

When the estimated location of the sensor is in a neighbourhood of its true location and space is sampled sufficiently densely \((N \gg 4)\), \( \delta_x \) is strictly increasing with respect to the \( x \)-wise location error between subsequent iterations:

\[
\frac{\delta_x[k + 1] - \delta_x[k]}{e_x[k + 1] - e_x[k]} > 0 \tag{25}
\]

Our first point shall be to prove that the derivative of \( u_x \) with respect to the error \( e_x \) is bounded above. From

\[
e_x[k + 1] = u_x[k + 1] + \delta_x[k + 1]
\]

\[
e_x[k] - e_x[k] = u_x[k + 1] - e_x[k] + \delta_x[k + 1]
\]

\[
\delta_x[k + 1] = (e_x[k] - u_x[k + 1]) + (e_x[k + 1] - e_x[k])
\]

\[
\delta_x[k + 1] - \delta_x[k] = (e_x[k] - \delta_x[k] - u_x[k + 1]) + (e_x[k + 1] - e_x[k])
\]

\[
\frac{\delta_x[k + 1] - \delta_x[k]}{u_x[k + 1] - u_x[k]} = -1 - \frac{u_x[k]}{u_x[k + 1] - u_x[k]}
\]

\[
\frac{\delta_x[k + 1] - \delta_x[k]}{u_x[k + 1] - u_x[k]} = -1 - \frac{e_x[k + 1] - e_x[k]}{u_x[k + 1] - u_x[k]}
\]

\[
\frac{\delta_x[k + 1] - \delta_x[k]}{u_x[k + 1] - u_x[k]} > 0
\]

(24) for \( k \leftarrow k + 1 \):

\[
e_x[k + 1] = u_x[k + 1] + \delta_x[k + 1]
\]

\[
e_x[k + 1] - e_x[k] = u_x[k + 1] - e_x[k] + \delta_x[k + 1]
\]

\[
\delta_x[k + 1] - \delta_x[k] = (e_x[k] - \delta_x[k] - u_x[k + 1]) + (e_x[k + 1] - e_x[k])
\]

\[
\delta_x[k + 1] - \delta_x[k] = (e_x[k] - \delta_x[k] - u_x[k + 1]) + (e_x[k + 1] - e_x[k])
\]

\[
\frac{\delta_x[k + 1] - \delta_x[k]}{u_x[k + 1] - u_x[k]} = -1 - \frac{u_x[k]}{u_x[k + 1] - u_x[k]}
\]

\[
\frac{\delta_x[k + 1] - \delta_x[k]}{u_x[k + 1] - u_x[k]} = -1 - \frac{e_x[k + 1] - e_x[k]}{u_x[k + 1] - u_x[k]}
\]

\[
\frac{\delta_x[k + 1] - \delta_x[k]}{u_x[k + 1] - u_x[k]} > 0
\]

Now, given from (16) that \( \delta_x = 0 \) when \( e_x = 0 \) (when the virtual sensor is posed at the pose of the true sensor, both the virtual and real scan perceive exactly the same points in space, and therefore \( x'_u = x'_v \) for all \( n \in [0, N] \)), and that \( u_x = 0 \) when \( e_x = 0 \) (by the same reasoning, from the definition of \( X_1 \), \( X_1 = 0 \) because \( d_n^r = d_n^v \) for all \( n \in [0, N] \)), we conclude that \( \delta_x = 0 \) when \( u_x = 0 \). Since \( \delta_x \) increases as \( u_x \) increases, we conclude that the input \( u_x \) exhibits the same sign as \( \delta_x \). Hence, from (24) we conclude that \( e_x, u_x, \delta_x \) all exhibit the same sign, and therefore that \( |u_x| \leq |e_x| \) and \( |\delta_x| \leq |e_x| \) for all admissible \( e_x \).

Let now \( V_x : \mathbb{R} \to \mathbb{R} \) be \( V_x(e_x) = e_x^2 \), where \( e_x(|e_x|) \leq V_x(e_x) \leq \beta_x(|e_x|) \), with \( \alpha_x \) \( (\cdot) \) \( \beta_x \) \( (\cdot) \) class \( K \) functions \([31]\) for all admissible \( e_x \). It is clear that \( V_x \) is a \( C^1 \) function,
\[ V_z(0) = 0, \text{ and positive everywhere else. Then} \]
\[ \Delta V_z = V_z(e_x[k+1]) - V_z(e_x[k]) \]
\[ \geq \delta^2_x[k] - \delta_x[k] + u_x[k])^2 \]  

\[ \leq 0 \]  

because \(|\delta_x + u_x| \geq |\delta_x|\) since \(\delta_x\) and \(u_x\) are of the same sign. We now rewrite (27):

\[ \Delta V_z = V_z(e_x[k+1]) - V_z(e_x[k]) \]
\[ = \delta^2_x[k] - e^2_x[k] \]
\[ \geq -e^2_x[k] + (e_x[k] - u_x[k])^2 \]  

\[ = u_x[k](u_x[k] - 2e_x[k]) \]  

(28)  

Since \(\Delta V_z \leq 0\), expression (29) is nonpositive, and therefore, omitting brackets since all variables refer to iteration \(k\), if \(u_x \geq 0 \Rightarrow 0 \leq u_x \leq 2e_x\), since \(e_x\) and \(u_x\) share the same sign. Alternatively, if \(u_x \leq 0 \Rightarrow 0 \leq 2e_x \leq u_x\) for the same reason. Now consider the former case:

\[ 0 \leq u_x \leq 2e_x \]
\[ -e_x \leq u_x - e_x \leq e_x \]

Since \(|u_x| \leq |e_x|\) and both \(u_x\) and \(e_x\) are positive: \(u_x \leq e_x\), which means that \(u_x - e_x \leq 0\). Then, multiplying all sides with \((u_x - e_x)\):

\[ -e_x \leq u_x - e_x \leq e_x \]
\[ -e_x(u_x - e_x) \leq (u_x - e_x)^2 \leq e_x(u_x - e_x) \]
\[ e_x(u_x - e_x) \leq (u_x - e_x)^2 \leq -e_x(u_x - e_x) \]

Therefore, from (28):

\[ \Delta V_z = -e^2_x + (e_x - u_x)^2 \]
\[ \leq -e^2_x - e_x(u_x - e_x) \]
\[ = -u_x e_x \]

Now consider the latter case:

\[ 2e_x \leq u_x \leq 0 \]
\[ e_x \leq u_x - e_x \leq -e_x \]

Since \(|u_x| \leq |e_x|\) and both \(u_x\) and \(e_x\) are negative: \(u_x \geq e_x\), which means that \(u_x - e_x \geq 0\). Then, multiplying all sides with \((u_x - e_x)\):

\[ e_x \leq u_x - e_x \leq -e_x \]
\[ e_x(u_x - e_x) \leq (u_x - e_x)^2 \leq -e_x(u_x - e_x) \]

Therefore, from (28):

\[ \Delta V_z = -e^2_x + (e_x - u_x)^2 \]
\[ \leq -e^2_x - e_x(u_x - e_x) \]
\[ = -u_x e_x \]  

Therefore, for all admissible \(e_x\): \(\Delta V_z \leq -u_x e_x\). The proof is analogous for the \(y\)-wise component, and hence omitted.

Let now \(V : \mathbb{R}^2 \rightarrow \mathbb{R}\) be \(V(e) = ||e||_2^2\), where \(\alpha(||e||) \leq V(e) \leq \beta(||e||)\), with \(\alpha(\cdot)\), \(\beta(\cdot)\) class \(\mathcal{K}\) functions for all admissible \(e\). It is clear that \(V = \) a \(C^1\) function, \(V(0) = 0\), and positive everywhere else. Then

\[ \Delta V = V(e[k+1]) - V(e[k]) \]
\[ = \left\| \begin{bmatrix} \delta_x[k] \\ \delta_y[k] \end{bmatrix} \right\|_2^2 - \left\| \begin{bmatrix} e_x[k] \\ e_y[k] \end{bmatrix} \right\|_2^2 \]

Dropping brackets for readability since all terms refer to the same iteration:

\[ \Delta V = (\delta^2_x - e^2_x) + (\delta^2_y - e^2_y) \]
\[ = -e^2_x + (e_x - u_x)^2 - e^2_y + (e_y - u_y)^2 \]  

\[ \leq -e^2_x - e_x(u_x - e_x) \]
\[ = -u_x e_x - u_y e_y \]
\[ = -u^\top e \]
\[ = -\gamma(||e||) \]

where \(\gamma(\cdot)\) is a class \(\mathcal{K}\) function. The latter equation stands because \(u^\top e = ||u|| ||e|| \cos \phi \geq 0\), where \(\phi\) is the angle between vectors \(u\) and \(e\), which, because \(u_x e_x \geq 0\) and \(u_y e_y \geq 0\), is acute: \(\phi \in [0, \pi/2]\), and therefore \(\cos \phi \geq 0\). Hence, from [32] (Theorem 2.3 p.26, points A2 p.27 and B1 p.28) the origin of the system (22) is uniformly asymptotically stable and, therefore, the range scan sensor’s location estimate \(l_t[k]\) converges to its real location \(l_t\) as \(k \rightarrow \infty\). \(\blacksquare\)

3.4. Proof of convergence and stability when disturbances are present

Suppose now that the ranges of the real scan \(S_r\) are corrupted by additive, bounded disturbances \(w_i^n\), where \(0 < |w_i^n| \leq W_i\), for all \(n \in [0, N]\), and that those of the virtual scan are similarly affected by \(w^n_v\), \(0 < |w^n_v| \leq W_v\) for all \(n \in [0, N]\). Then, denoting by \(\theta^n_r = \theta^n_r + w^n_r\) the range of ray \(n\) measured from the real range scan sensor, and by \(\theta^n_v = \theta^n_v + w^n_v\) the range obtained from the virtual sensor, and applying the same rationale as in subsection 3.3, we arrive at the analogous expression to (20):

\[ \begin{bmatrix} e_x \\ e_y \end{bmatrix} = \frac{1}{N} \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} \tilde{X}_{1,r} \\ \tilde{X}_{1,i} \end{bmatrix} + \begin{bmatrix} \delta_x \\ \delta_y \end{bmatrix} - \begin{bmatrix} w_x \\ w_y \end{bmatrix} \]

\[ = \begin{bmatrix} u_x \\ u_y \end{bmatrix} + \begin{bmatrix} \delta_x \\ \delta_y \end{bmatrix} - \begin{bmatrix} w_x \\ w_y \end{bmatrix} \]
where
\[
\begin{align*}
    w_x &= \frac{1}{N} \sum_{n=0}^{N-1} (w_r^n - w_{\theta}^n) \cos(\theta + \frac{2\pi n}{N}) \\
    w_y &= \frac{1}{N} \sum_{n=0}^{N-1} (w_r^n - w_{\theta}^n) \sin(\theta + \frac{2\pi n}{N})
\end{align*}
\]

0 < |w_x| \leq W_x < \infty, 0 < |w_y| \leq W_y < \infty, i.e. \( w_x \) and \( w_y \) are bounded since they are products of finite operations on finite quantities, \( X_{1,r} \) and \( X_{1,i} \), are respectively, the real and imaginary parts of the complex quantity \( \tilde{X}_1 \):
\[
\tilde{X}_1 = \sum_{n=0}^{N-1} (\tilde{d}_n^r - \tilde{d}_n^i) \cdot e^{i\frac{2\pi n}{N}}
\]

and
\[
\begin{bmatrix}
\tilde{u}_x \\
\tilde{u}_y
\end{bmatrix} = \begin{bmatrix}
\tilde{u}_x \\
\tilde{u}_y
\end{bmatrix} + \begin{bmatrix}
w_x \\
w_y
\end{bmatrix}
\]

(31)

where \( \tilde{u} = [u_x, u_y]^T \) is the control vector of the unperturbed system (22) and \( \tilde{u} = [\tilde{u}_x, \tilde{u}_y]^T \) is the input to the (now perturbed) system. Let us now denote the disturbance by \( w = [w_x, w_y]^T \); then the dynamics of the error of the perturbed system become:
\[
e[k+1] = e[k] - \tilde{u}[k] \tag{31} = e[k] - u[k] - w[k] \tag{32}
\]

Since \( w \) is bounded, and the origin of the unperturbed system (22) is uniformly asymptotically stable, the solution \( e[k] \) of the perturbed system (32) is uniformly bounded for \( k \geq k_0 \) and uniformly ultimately bounded with an ultimate bound which is a function of \( W_x, W_y \), where \( W_x \leq W_x, W_y \leq W_y \) [32] (Theorem 2.7, p.29).

3.5. In algorithmic form

The process of estimating the location of the sensor’s pose by the proposed method—dubbed Iterative Correspondenceless Translation Estimator (ICTE) — is more clearly illustrated in pseudocode, where it is described in Algorithm I. The process’ inputs are the map \( M \) of the robot’s environment, the range scan captured from the true pose of the real sensor \( S_t^i \) at time \( t \), its properties, the estimated pose of the sensor \( \hat{p}_t(x_t, y_t, \theta_t) \), the threshold of maximum iterations to run \( k_{max} \) and the error threshold \( \varepsilon_u \). The map-scan \( S_t^i \) for the \( k \)-th iteration is computed in line 4 (Algorithm II), and, together with the real scan, the two are inputted to the \textit{diffDFT} routine (Algorithm III). The latter’s output is a complex number, \( X_1 \), whose real and imaginary parts are used in lines 7,8 to compute the \( x \)-wise and \( y \)-wise corrections of the sensor’s pose estimate (lines 9,10). If the \( 2 \)-norm of the control vector \( \mathbf{u}_k = [u_x, u_y]^T \) is found to be below the preset threshold \( \varepsilon_u \) (line 11), the algorithm halts and returns its last estimate; otherwise it iterates through the same sequence of steps until this last condition is met, or until it runs out of iterations.

\textbf{Algorithm I.} \textit{icte}

\textbf{Input:} \( M, S_t^i, \text{sensorProperties}, \hat{p}_t(x_t, y_t, \theta_t), k_{max}, \varepsilon_u \)

\textbf{Output:} \( \hat{p}'_t(x'_t, y'_t, \theta'_t) \)

\begin{itemize}
    \item[1:] \( k \leftarrow 0 \)
    \item[2:] \( N = \text{sensorProperties.N} \)
    \item[3:] while \( k < k_{max} \) do
    \item[4:] \( S_t^i[k] \leftarrow \text{scanMap}(M, \hat{p}_t, \text{sensorProperties}) \)
    \item[5:] \( X_1[k] \leftarrow \text{diffDFT}(S_t^i[k]) \)
    \item[6:] \( (X_{1,r}, X_{1,i}) \leftarrow (\text{Re}(X_1), \text{Im}(X_1)) \)
    \item[7:] \( u_x \leftarrow (\cos \theta_t \cdot X_{1,r} + \sin \theta_t \cdot X_{1,i})/N \)
    \item[8:] \( u_y \leftarrow (\sin \theta_t \cdot X_{1,r} - \cos \theta_t \cdot X_{1,i})/N \)
    \item[9:] \( u_k \leftarrow (u_x, u_y) \)
    \item[10:] \( \hat{p}_t[k] \leftarrow \hat{p}_t[k] + u_k \)
    \item[11:] if \( \|u_k\|_2 < \varepsilon_u \) then
    \item[12:] break
    \item[13:] end if
    \item[14:] \( k \leftarrow k + 1 \)
    \item[15:] end while
    \item[16:] \( \hat{p}'_t \leftarrow \hat{p}_t[k] \)
    \item[17:] return \( \hat{p}'_t \)
\end{itemize}

Algorithms II and III respectively describe the inner processes \textit{scanMap}(\cdot) and diffDFT(\cdot) of Algorithm I.

In Algorithm II, the routine for computing a map-scan is outlined: in line 5, \( \lambda_n \) represents the angle that the map’s \( n \)-th ray forms with respect to the sensor’s \( x \) axis in its local frame of reference, and in line 6 it is expressed with respect to the map’s frame of reference (\( \theta_n \)). The intersection point of that ray, with a starting location that of the sensor’s pose estimate, and an orientation equal to \( \theta_n \) and the map is then computed at line 7, and its range from the estimated pose of the sensor is calculated in line 8. This process is carried out sequentially for each ray \( n \in [0, N] \).

Algorithm III illustrates the elementary routine for computing the first term of the Discrete Fourier transform between ranges of homologous rays of \( S_t^i \) and \( S_t^{i[k]} \). \( \Delta \) is a one-dimensional vector of equal size to the number of rays of each scan. The \( n \)-th element of \( \Delta \) holds the range difference between the \( n \)-th ray of \( S_t^i \) and that of \( S_t^{i[k]} \). The directive DFT(\( \Delta \)) in line 8 computes the (complex) terms of the Discrete Fourier transform of \( \Delta \) and, finally, in line 9, the first term (equation (4)) is extracted as the second element of the returned vector (assuming zero-based indexing).

3.6. Commentary

The convergence of the location estimate and its final error depend mainly on the disturbance acting on the
Algorithm II scanMap

Input: \( M, \hat{p}_t(x_t, y_t, \theta_t), \) sensorProperties
Output: \( S^{[k]}_v \)
1: \( N = \text{sensorProperties}.N \)
2: \( \lambda = \text{sensorProperties}.\lambda \)
3: \( S^{[k]}_v \leftarrow \{\} \)
4: \textbf{for} \( n = 0 : \text{sensorProperties}.N - 1 \) \textbf{do}
5: \( \lambda_n \leftarrow \frac{\lambda}{2} + n \frac{\lambda}{N} \)
6: \( \theta_n \leftarrow \lambda_n + \hat{\theta} \)
7: \( (x_n, y_n) \leftarrow \text{intersect}(M, (\hat{x}_t, \hat{y}_t, \theta_n)) \)
8: \( d_n^v \leftarrow ||(\hat{x}_t - x_n, \hat{y}_t - y_n)||_2 \)
9: \text{Append} \( (d_n^v, \lambda_n) \) to \( S^{[k]}_v \)
10: \textbf{end for}
11: \textbf{return} \( S^{[k]}_v \)

Algorithm III diffDFT

Input: \( S^t_v, S^{[k]}_v \)
Output: \( X_1 \)
1: \text{Assert that size of} \( S^t_v \) \text{equals size of} \( S^{[k]}_v \)
2: \( N \leftarrow \text{size of} \ S^t_v \)
3: \( \Delta \leftarrow \{\} \)
4: \textbf{for} \( n = 0 : N - 1 \) \textbf{do}
5: \( d \leftarrow S^t_v \cdot d_n - S^{[k]}_v \cdot d_n \)
6: \text{Append} \( d \) to \( \Delta \)
7: \textbf{end for}
8: \( X \leftarrow \text{DFT}(\Delta) \)
9: \( X_1 \leftarrow X[1] \)
10: \textbf{return} \( X_1 \)

Readings of the real and the virtual range sensor. With respect to the former, range inaccuracy with respect to real distances to obstacles in the sensor’s environment is not only due to the real sensor’s finite range resolution, but depends also on the material of the target surface, its distance with respect to the obstacle, the sensor’s temperature, and others [19]. As for the virtual scan sensor, we refer to acting disturbances in the sense of map inaccuracies: when the map does not accurately correspond to the environment, ranges captured from within the map are inaccurate with respect to the environment and, therefore, subject to perceived noise.

In the absence of disturbances, ranges from both sensors capture the environment perfectly and the sensor’s (robot’s) location estimate can be brought arbitrarily close to the true location of the sensor (robot). Figure 2 illustrates the evolution of the location estimate within a sample map, while figure 3 illustrates the evolution of the norms of the location error \( e = [e_x, e_y]^T \) and the control vector \( u = [u_x, u_y]^T \) as a function of time. In this case, convergence is asymptotic, and the location estimate error is driven exclusively by the control vector.

![Figure 2](image-url)

Figure 2: Estimate correction illustration in the absence of disturbances: the map captures the environment perfectly and sensor noise is absent. The true location of the sensor is depicted with a cross, the initial estimate with a circle, and the intermediate location estimates with × marks.

Figure 4 illustrates the evolution of the error \( e \), control vector \( u \), and function \( \delta = [\delta_x, \delta_y]^T \) per-axis components as a function of time, all converging to zero as time grows. Figure 5 depicts the components of \( \delta \) as a function of the error \( e \) in the corresponding axis; both functions are strictly increasing.

By contrast, when noise is present in the ranges of either the real or the virtual range scan sensor, the control vector does not unilaterally determine the estimate error. Figure 6 depicts the trajectory of the location estimate near the target location when both real and virtual scans are perturbed by zero-mean, normally-distributed noise with standard deviation equal to \( \sigma = 0.05 \) m, given...
the same map and initial location as in the case of absent disturbances. Due to the presence of noise, the location estimate is offset and unable to converge arbitrarily close to it. Here, $\epsilon_y$, the error threshold for stopping, has been set high enough so that the algorithm terminates only due to reaching the maximum number of iterations, set here to $k_{max} = 600$. Figure 7 illustrates the evolution of the norms of the location error $\epsilon$ and the control vector $u$ as time progresses, where it is evident that both are not strictly decreasing from some iteration forward (for $k \geq k_0 = 13$ in particular). In parallel, for all $k \geq k_0$, the error is bound, as guaranteed by Theorem II: the trajectory of the estimate location (figure 6) does not diverge for any $k \in [0, k_{max}]$.

Finally, the proposed method is akin to the update step of the (Extended) Kalman filter: both assume as inputs a prior pose estimate, a measurement vector at the current time step, and the map of the environment, and both attempt to improve the accuracy of this prior estimate by utilising a measure of divergence of measurements from their estimated values (compare eq. (4) to the calculation of the innovation in the update step of the Kalman filter). In the Kalman filter’s case this prior is issued from the robot’s motion model, but in the case of the proposed method this prior may be issued from additional sources. However, their differences lie in the facts that the proposed
method (a) assumes that the orientation of the robot is known, and (b) does not make assumptions on the distribution of noise affecting the range measurements of either the physical or the virtual range sensor, apart from it being additive and bounded.

4. Experimental Procedure

This section serves to illustrate the efficacy and performance of the proposed method. The following experiments were conducted using a benchmark dataset consisting of 778 laser scans obtained from a Sick range-scan sensor mounted on a robotic wheel-chair emitting 360 rays over a 180° field of view. The same dataset was used to evaluate the performance of IDC [33], ICP, and MBICP in [11], and of PLICP in [12], wherein the latter was found to be the best-performing among the four correspondence-finding state-of-the-art scan-matching methods. For each scan, the dataset reports 360 range measurements and a pose \( p(x, y, \theta) \) from which it was captured.

The conducted experiments test for performance in two regards and with two discrete objectives: (a) to query on the difference of mean error between the proposed method, the optimal correspondence-finding scan-matching method (PLICP), and the equally correspondenceless but probabilistic scan-matching method of NDT when employed in a scan-to-map-scan matching context, and (b) to query on how the proposed method’s mean error varies with respect to common and varying limitations of the range-scan sensor, such as varying field of view or number of rays emitted, and real-life occurrences of failure, such as retrieval of rays with invalid range or the reasonable non-perfection of the method that aligns the sensor’s orientation estimate with the sensor’s true orientation, typically needed before the proposed method can be employed. The results of tests relative to the first category are found in subsection 4.1, while those relative to the second in subsection 4.2.

In the following we describe how the inputs of the proposed method (which are the same as those of PLICP and NDT) are constructed given only a dataset instance.

The proposed method requires as inputs: (a) a map \( M \) of the robot’s surroundings, (b) the sensor’s pose estimate \( p \) within \( M \), (c) a range scan \( S \), ranging over \( 2\pi \), (d) an upper threshold of iterations \( k_{\text{max}} \), and (e) a threshold for stopping \( \varepsilon_u \).

As the proposed method requires a map, dataset instances were used to construct a map as follows. Let a dataset \( D \) instance \( d \) comprising \( N_d = 360 \) range measurements \( R_d = \{ r_{d,i} \}, r_{d,i} \in \mathbb{R}, i = \{0, 1, \ldots, N_d-1\} \) be captured from pose \( p_d = (px_d, py_d, p\theta_d) \in \mathbb{R}^2 \times [-\pi, \pi) \). A map \( M_d \) corresponding to the \( d \)-th scan, \( d = \{0, 1, \ldots, N_d-1\} \), \( N_d = 778 \), is constructed as a collection of points, where the coordinates of each point in the \( x-y \) plane are respectively \( x_{d,i} = px_d + r_{d,i} \cos(-\pi/2 + i\pi/N_d + p\theta_d) \) and \( y_{d,i} = py_d + r_{d,i} \sin(-\pi/2 + i\pi/N_d + p\theta_d) \). Since \( M_d \) ranges over a field of view of \( \pi < 2\pi \) rads for the particular dataset, there is a range of options on how to supplement it so that it ranges over an angular field of view of \( 2\pi \) and bridge the gap non-arbitrarily that we identify: (a) mirror the points of \( M_d \) with respect to \( p_d \) with the axis of symmetry set to that corresponding to the \( x \) axis of the robot (assuming the right-handed 3D coordinate frame convention) (b) mirror them with the axis of symmetry that of the \( y \) axis of the robot, and (c) draw a semicircular arc around \( p_d \) with radius set to the minimum range between the two extreme rays. All three have been found equivalent with respect to the method’s performance.

The initial estimated pose of the virtual sensor \( \hat{p} \) is obtained for each dataset instance by perturbing the \( x \) and \( y \) axis components of \( p_d \) with quantities extracted from a uniformly distributed error distribution \( U(-\alpha, \alpha), \alpha \in \mathbb{R} \).

A virtual scan \( S_v \) is then produced by locating the intersections of rays emanating from the virtual sensor’s estimated pose \( \hat{p} \) and ranging over \( \lambda = 2\pi \) within the produced map \( M_d \) with the lines connecting its points. The corresponding real laser scan \( S \) is obtained by augmenting vector \( R_d \) by appending it with the range of rays captured from \( p_d \) within \( M_d \) from the missing \( \{\pi/2, \pi\} \cup [-\pi, -\pi/2] \) range. Lastly, the iterations threshold was set to \( k_{\text{max}} = 60 \), and the threshold for stopping to \( \varepsilon_u = 10^{-5} \) m.

In order to test for the performance of the proposed method, we test for four discrete values of \( \alpha \), in order to progressively test it in the range of pose estimate error values typically reported in the bibliography: \( \alpha \in \{0.05, 0.10, 0.15, 0.20\} \) m.

Furthermore, we test for five different levels of disturbances acting on the range measurements of the real and virtual scans, as these may manifest themselves in real conditions: Range scans are affected by additive zero-
mean normally distributed noise with standard deviation equal to \( \sigma \in \{0.0, 0.005, 0.01, 0.02, 0.05\} \) m. For each displacement value, each noise level, and each case, we run Algorithm I for \( I = 100 \) times for each dataset, totaling \( 4 \times 5^2 \times 6 \times 100 \times 788 \sim \mathcal{O}(10^7) \) runs. PLICP and NDT were ran only for the nominal case, i.e., without querying their performance under limitations or failures, for a total of \( 4 \times 5^2 \times 100 \times 778 \sim \mathcal{O}(10^6) \) runs. All experiments and all algorithms ran serially, on a single thread, in a machine with a CPU frequency of 4.00 GHz.

Unless otherwise noted, in the following figures circles (○) denote the performance of either the proposed method, PLICP, or NDT when \( \alpha = 0.05 \) m, stars (⋆) when \( \alpha = 0.10 \) m, downward-facing triangles (▽) when \( \alpha = 0.15 \) m, and squares (□) when \( \alpha = 0.20 \) m.

4.1. Comparison against the state of the art

Figure 8 illustrates the mean error of the proposed method on the tested dataset for different levels of sensor position displacement, real scan noise levels, and virtual scan noise levels, over \( I = 100 \) runs. Figures 9 and 10 illustrate the mean error of the PLICP and NDT methods over the same dataset for the same levels of displacement and scans’ noise and over the same number of runs.

The proposed method’s mean errors are consistently invariant across different levels of position displacement, i.e. in a neighbourhood of the true sensor position, in terms of the position error, its performance does not depend on the sensor’s initial position estimate. With regard to PLICP, the proposed method’s mean errors are lower; their difference increases the more noise is present in either real or virtual scan for the same level of position displacement, or the greater the initial position error is for the same levels of scans’ noises. Figure 11 depicts a direct comparison between the proposed method and PLICP when \( \alpha = 0.05 \) m. With regard to NDT, the proposed method’s mean errors are significantly lower. NDT’s performance is also dependent on the initial displacement, but it demonstrates a higher degree of robustness compared to PLICP, as its mean position errors are almost invariant to either sensor noise or map-to-environment discrepancy for a given level of initial displacement.

Figure 12 illustrates the corresponding mean execution times of the proposed method. Since, in general, what ultimately determines the length of execution is the combination of the maximum iterations threshold \( k_{\text{max}} \) and the threshold for stopping \( \varepsilon_u \), it is evident from figures 8 and 12 that the configuration \( (k_{\text{max}}, \varepsilon_u) = (60, 10^{-5}) \) is able to provide improved position estimation at real-time for realistic levels of estimation error and noise in scans. The longest overall execution did not take more than 25 ms, and therefore the highest frequency at which the method can be run is at 40 scan-matchings per second. PLICP’s mean execution time varied between 4 ms in the case of minimal position estimate displacement and noise levels and 28 ms in the case of their maximal values. The cor-
4.2. A characterisation over limitations posed by reality

Commercially available LIDAR sensors vary in terms their field of view $\lambda$ and number of rays they emit $N$. In terms of the latter, the position error exhibited by the proposed method decreases in proportion to the number of available rays. This result stands to reason as, the denser the sampling of the environment and the map, the better a virtual scan may approximate the real scan, and therefore the closer the robot’s position estimate may approximate its actual position. In terms of a sensor’s field of view, the proposed method’s location errors increase when the field of view is $\lambda < 2\pi$. In that case location errors become proportional to the initial displacement between the sensor’s location estimate and its true location. Furthermore, the proposed method reports reduced errors for a sensor with a field of view of $\lambda = 3\pi/2$ in comparison to a panoramic sensor which emits 360 rays or fewer.

In real conditions, the range reported for a number of rays may be invalid due to sensor fault. The proposed method reports a slight increase in position errors for a percentage of randomly invalid rays in the range of 10%-20% compared to the case of all rays being valid. Random invalidation of half of the total rays emitted produces slight increase in position errors compared to those reported for a sensor emitting all-valid rays but half in number. This fact suggests that even sampling of space fares better than uneven sampling.

Another limitation that manifests in real conditions is that, due to the existence of a maximum detectable range by a LIDAR sensor, the range reported for a number of
consecutive rays may peak at its maximum range (e.g., when a robot is placed at one end of a long corridor), thus reporting a misrepresentation for some portion of the environment. As expected, the larger these portions are in size, the greater the position error that the proposed method exhibits. However, the results also show that for increasing position displacements, the proposed method performs better when invalid rays are concentrated over one region compared to their being scattered unevenly.

Finally, the orientation estimate provided to the proposed method may be accurate to a certain extent. Simulations show that for the range of orientation estimate errors reported in the bibliography, the proposed method’s position errors may increase by as much as five times compared to the case of complete orientation coincidence, if these orientation errors are left untreated.

The complete results for the above cases are reported in the appendix, in Figures 13-19.

5. Conclusions

The premise of this study was improving the location estimate of a mobile robot capable of motion on a plane and mounted with a conventional 2D LIDAR sensor, given an initial guess for its location on a 2D map of its surroundings. The preceding analysis provided the theoretical reasoning behind solving a matching problem between two homoriented 2D scans, one derived from the robot’s physical sensor and one derived by simulating its operation within the map, in a way that does not require the establishing of correspondences between their constituting rays. Two results were proved and subsequently shown through experiments. The first is that the true position of the sensor can be recovered with arbitrary precision when the physical sensor reports faultless measurements and there is no discrepancy between the environment the robot operates in and its perception of it by the robot. The second is that when either is affected by disturbance, the location estimate is bound in a neighbourhood of the true location whose radius is proportional to the affecting disturbance.

References

[1] P. Maybeck, “Stochastic Models, Estimation and Control”, Volume 1, Academic Press, New York, 1979
[2] F. Dellaert, D. Fox, W. Burgard and S. Thrun, “Monte Carlo localization for mobile robots,” Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No.99CH36288C), Detroit, MI, USA, 1999, pp. 1322-1328 Volume 2, doi: 10.1109/ROBOT.1999.772544
[3] Thrun, S., Particle Filters in Robotics, Proceedings of the 17th Annual Conference on Uncertainty in AI (UAI), 2002
[4] Sebastian Thrun, Wolfram Burgard, and Dieter Fox, “Probabilistic Robotics” (Intelligent Robotics and Autonomous Agents), The MIT Press, 2005
[5] Zhu, J., Zheng, N., and Yuan, Z., “An Improved Technique for Robot Global Localization in Indoor Environments”, International Journal of Advanced Robotic Systems, 2011, https://doi.org/10.5722/10525
[6] Kumagai, I., Ueda, R., Sugai, F., Nozawa, S., Kakuchi, Y., Okada, K., and Inaba, M. (2016). Achievement of localization system for humanoid robots with virtual horizontal scan relative to improved odometry fusing internal sensors and visual information. 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). doi:10.1109/ios.2016.7759124
[7] J. Röwekämper, C. Sprunk, G. D. Tipaldi, C. Stachniss, P. Pfaff and W. Burgard, “On the position accuracy of mobile robot localization based on particle filters combined with scan matching”, 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, Vilamoura, 2012, pp. 3158-3164, doi: 10.1109/ROS.2012.6385988
[8] Vasilevic, G., Miklic, D., Draganjac, I., Kovacic, Z., and Lista, P. “High-accuracy vehicle localization for autonomous warehousing”, Technical Report, 2016
[9] Fox, D., “Adapting the Sample Size in Particle Filters Through KLD-Sampling”, The International Journal of Robotics Research, 22(12), 2003, pp. 965-1003, https://doi.org/10.1177/0278364903221001
[10] P. J. Besl and N. D. McKay, “A method for registration of 3-D shapes”, IEEE Transactions on Pattern Analysis and Machine Intelligence, 1992, volume 14, number 2, pp. 239-256, doi: 10.1109/34.121791, ISSN 0162-8828
[11] J. Mingal, F. Lamiraux and L. Montesano, “Metric-Based Scan Matching Algorithms for Mobile Robot Displacement Estimation,” Proceedings of the 2005 IEEE International Conference on Robotics and Automation, Barcelona, Spain, 2005, pp. 3557-3563, doi: 10.1109/ROBOT.2005.1570661
[12] A. Censi, “An ICP variant using a point-to-line metric,” 2008 IEEE International Conference on Robotics and Automation, Pasadena, CA, 2008, pp. 19-25, doi: 10.1109/ROBOT.2008.4543181
[13] P. Biber and W. Strasser, “The normal distributions transform: a new approach to laser scan matching,” Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No.03CH37453), Las Vegas, NV, USA, 2003, pp. 2734-2748 vol.3, doi: 10.1109/IROS.2003.1249285.
[14] Saarinen, Jari & Andreasssen, Henrik & Stoyanov, Todor & Lilenthal, Achim. (2013). “Normal distributions transform Monte-Carlo localization (NDT-MCL).” Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE/RSJ International Conference on Intelligent Robots and Systems, 382-389. 10.1109/ROS.2013.6696380.
[15] Ahtiani, Juhana & Stoyanov, Todor & Saarinen, Jari. (2016). “Normal Distributions Transform Traversability Maps: LIBAR-Only Approach for Traversability Mapping in Outdoor Environments: Normal Distributions Transform Traversability Maps,” Journal of Field Robotics. 34. 10.1002/rob.21657.
[16] Bouraine, S., Bougouffa, A. & Azouaoui, O. “Particle swarm optimization for solving a scan-matching problem based on the normal distributions transform.” Evol. Intel. (2021). https://doi.org/10.1007/s12065-020-00545-9
[17] H. Hong and B. H. Lee, “Probabilistic normal distributions transformation representation for accurate 3D point cloud registration,” 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, 2017, pp. 3333-3338, doi: 10.1109/RSJ.2017.8206170.
[18] Liu T, Zheng J, Wang Z, Huang Z, Chen Y. “Compositional clustering normal distribution transform algorithm.” International Journal of Advanced Robotic Systems. May 2020. doi:10.1177/1729881420912142
[19] Cooper, M.A.; Raquet, J.F.; Patton, R. Range Information Characterization of the Hokuyo UST-20LX LIDAR Sensor. Photometrics 2018, 5, 12.
[20] J. Mingal, F. Lamiraux and L. Montesano, “Metric-Based Scan Matching Algorithms for Mobile Robot Displacement Estimation,” Proceedings of the 2005 IEEE International Conference on Robotics and Automation, Barcelona, Spain, 2005, pp. 3557-3563, doi: 10.1109/ROBOT.2005.1570661
cient fastSLAM algorithm for generating maps of large-scale cyclic environments from raw laser range measurements,” Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No.03CH37453), Las Vegas, NV, USA, 2003, pp. 206-211 vol.1. doi: 10.1109/IROS.2003.1250629

[22] Chieh-Chih Wang, C. Thorpe and S. Thrun, “Online simultaneous localization and mapping with detection and tracking of moving objects: theory and results from a ground vehicle in crowded urban areas,” 2003 IEEE International Conference on Robotics and Automation (Cat. No.03CH37422), Taipei, Taiwan, 2003, pp. 842-849 vol.1. doi: 10.1109/ROBOT.2003.1241698

[23] Lacroix, S., Mallet, A., Bonnafous, D., Bauxil, G., Fleury, S., Herrb, M., and Chatila, R. (2002). “Autonomous Rover Navigation on Unknown Terrains: Functions and Integration”. The International Journal of Robotics Research, 21(10–11), 917–942. https://doi.org/10.1177/0278364902021010841

[24] Minguez, J., Montesano, L., and Montano, L. (2004). “An architecture for sensor-based navigation in realistic dynamic and troublesome scenarios”. 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566), 3, 2750-2756 vol.3.

[25] Montesano, Luis, Minguez, Javier and Montano, Luis, “Modeling dynamic scenarios for local sensor-based motion planning”, Autonomous Robots, 2008, Volume 25, pp 231-251.

[26] D. Schulz, W. Burgard, D. Fox and A. B. Cremers, “Tracking multiple moving targets with a mobile robot using particle filters and statistical data association”, Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No.01CH37164), Seoul, South Korea, 2001, pp. 1665-1670 vol.2. doi: 10.1109/ROBOT.2001.9327850

[27] Sandberg, David and Wolff, Krister and Wahde, Mattias, “A Robot Localization Method Based on Laser Scan Matching”, Advances in Robotics, 2009, Springer Berlin Heidelberg, pp. 171–178, isbn 978-3-642-03983-6

[28] Gang Peng, Wei Zheng, Zezao Lu, Jinhua Liao, Lu Hu, Gongyue Zhang, and Dingxin He, “An Improved AMCL Algorithm Based on Laser Scanning Match in a Complex and Unstructured Environment”, Complexity, Volume 2018, Article ID 2327637, 11 pages, 2018, https://doi.org/10.1155/2018/2327637.

[29] Lingemann, Kai, Andreas Nüchter, Joachim Hertzberg and Hartmut Surmann. “High-speed laser localization for mobile robots.” Robotics and Autonomous Systems 51 (2005): 275-296.

[30] Filotheou, A., Tsardoulias, E., Dimitriou, A. et al. “Pose Selection and Feedback Methods in Tandem Combinations of Particle Filters with Scan-Matching for 2D Mobile Robot Localisation”. J Intell Robot Syst 100, 925–944 (2020). https://doi.org/10.1007/s10846-020-01253-6

[31] H. Khalil, Nonlinear Systems,1996, Prentice-Hall, New Jersey

[32] C. Cruz-Hernández, J. Álvarez-Gallegos and R. Castro-Linares, “Stability of discrete nonlinear systems under vanishing perturbations: application to a nonlinear model-matching problem,” in IMA Journal of Mathematical Control and Information, vol. 16, no. 1, pp. 23-41, March 1999. doi: 10.1093/imamci/16/1.23

[33] Feng Lu and Milios, “Robot pose estimation in unknown environments by matching 2D range scans,” 1994 Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 1994, pp. 935-938.
6. Appendix

This section houses the results of simulations of the proposed method which are conducted over a range of constraints posed by real conditions (section 4.2).

Figure 13 depicts the proposed method’s mean position errors for varying number of sensor rays: nominal (720), half (360), and one third (240) when \( \alpha = 0.2 \text{ m} \) (in fact all displacement configurations yield the same results, as in the nominal case). Evidently, the error decreases in proportion to the number of available rays.

![Figure 13: Mean position errors of the proposed method over I = 100 runs for each combination of real scan noise standard deviation \( \sigma_r \), virtual scan standard deviation \( \sigma_v \), for displacement \( \alpha = 0.2 \text{ m} \), for different number of rays emitted from the range sensor. Dots signify the nominal configuration (720 rays), circles a reduction by half (360 rays), and diamonds by a third (240 rays).](image)

Figure 14 illustrates the proposed method’s mean position errors when the range-finder sensor’s field of view is \( \lambda = 3\pi/2 < 2\pi \text{ rad} \), distributed evenly over the sensor’s \( z \) axis. Evidently, what determines the independence of the method’s performance from the initial location error is whether or not the range sensor has a panoramic field of view.

Figure 15 depicts the proposed method’s mean position error for varying levels of randomly invalidated rays—scenario of rather uncommon failures (10% – 20% of the nominal number of rays, which is 720) and extreme failures (50%). When a ray is detected as invalid, the value of its range along with the corresponding one from the map-scan is zeroed out but included in the computation of the \( X_1 \) term (equation (4)). For up to uncommon levels of sensor failure to retrieve range, the proposed method’s performance is relatively unaffected.

Figures 16-18 illustrate the proposed method’s mean position errors for varying levels of consecutively invalidated rays—common cases where obstacles are farther

![Figure 14: Mean position errors of the proposed method over I = 100 runs for each combination of real scan noise standard deviation \( \sigma_r \), virtual scan standard deviation \( \sigma_v \), and displacement \( \alpha = 0.2 \text{ m} \) when the range sensor’s field of view is reduced to \( \lambda = 3\pi/2 \text{ rad} \).](image)

![Figure 15: Mean position errors of the proposed method over I = 100 runs for each combination of real scan noise standard deviation \( \sigma_r \), virtual scan standard deviation \( \sigma_v \), for displacement \( \alpha = 0.2 \text{ m} \), for different levels of randomly invalidated number of rays: dots denote the nominal configuration (none invalid), circles denote a configuration where 10% of all rays are randomly invalidated, diamonds when 20% are invalidated, and pentagrams when 50% are invalidated](image)
away from the sensor than its maximum range. The index of the first ray from which an invalid block of rays is established is chosen at random among all rays. Analogously to the case where the index of an invalid range is chosen at random between all, the performance of the proposed method is not significantly affected when invalid blocks are relatively small in size, but deteriorates to twice the mean error compared to the nominal case when half of all rays are consecutively invalid.

Figure 19 depicts the proposed method’s mean errors when the sensor’s estimate is rotationally misaligned with regard to the sensor’s true pose. The rotational displacement for each experiment was chosen from the uniform distribution $U_{\alpha} = U(-\rho_1, -\rho_0) \cup U(\rho_0, \rho_1)$, where $\rho_0 = 0.003$ rad and $\rho_1 = 0.01$ rad. The values of $\rho_\alpha$ were selected from the high end of the spectrum of mean rotational errors reported in the literature of scan–to–map-scan matching, as documented in subsection 2.3. Overall, from figure 19 it is quite obvious that the performance of the proposed method relies heavily on the precision of its antecedent method that angularly aligns the two scans.
Figure 19: Mean position errors of the proposed method over $I = 100$ runs for each combination of real scan noise standard deviation $\sigma_r$, virtual scan standard deviation $\sigma_v$, and displacement $\alpha$, when the estimated pose of the sensor is rotationally misaligned with respect to its true pose. The rotation error is chosen at random from a uniform distribution $U_r = U(-\rho_1, -\rho_0) \cup U(\rho_0, \rho_1)$, $\rho_0 = 0.003$ rad and $\rho_1 = 0.01$ rad.