Relax and Recover: Guaranteed Range-Only Continuous Localization

Michalina Pacholska[†] Frederike Dümbgen[†] Adam Scholefield

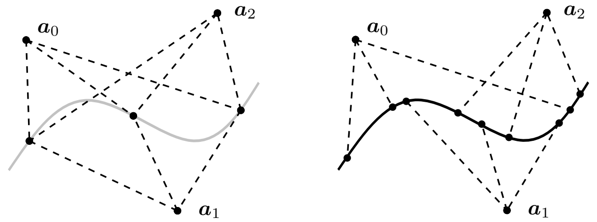
***Abstract*— Range-only localization has applications as diverse as underwater navigation, drone tracking and indoor localization. While the theoretical foundations of lateration—range-only localization for static points—are well understood, there is a lack of understanding when it comes to localizing a moving device. As most interesting applications in robotics involve moving objects, we study the theory of trajectory recovery. This problem has received a lot of attention; however, state-of-the-art methods are of a probabilistic or heuristic nature and are not well suited for guaranteeing trajectory recovery.**

In this paper, we pose trajectory recovery as a quadratic problem and show that we can relax it to a linear form, which admits a closed-form solution. We provide necessary and sufficient recovery conditions and in particular show that trajectory recovery can be guaranteed when the number of measurements is proportional to the trajectory complexity. Finally, we apply our reconstruction algorithm to simulated and real-world data.

I. Introduction

A robot's ability to localize itself accurately is essential for applications such as exploration, rescue and delivery. In many environments, visual (camera-based) simultaneous localization and mapping (SLAM) is the go-to solution for reliable localization [1]. However, various settings exist in which visual SLAM is not practical, for example when scanning the environment is impossible (e.g. passive indoor localization) or when the environment does not exhibit enough reliable features (for example under water [2], [3], at high altitudes [4], or in large exhibition-style rooms [5]). In such situations, it is sometimes more feasible to install a few fixed anchors which can provide the robot with distance measurements. Given sparse range measurements from multiple anchors, the robot can calculate its position through multilateration.

While position recovery guarantees exist for traditional lateration in static setups (see Figure 1 (a)), the problem is less understood when the robot is moving. To date, practical systems predominately recover trajectories by coupling partial lateration with filtering techniques. While these approaches lead to good performance, they offer little hope of providing fundamental guarantees for the recovery of the robot's continuous trajectory. In these cases, under which conditions is it possible to uniquely recover the trajectory?



(a) Conventional lateration (b) Proposed trajectory estimation

Fig. 1. Two different approaches for recovering a trajectory $\boldsymbol{r}(t)$ (solid line) from distance measurements (dashed lines) to anchors $\boldsymbol{a_m}$. In conventional lateration (a), we recover single points at which we have at least $D+1$ distance measurements. The proposed method (b) recovers the continuous representation of the trajectory $\boldsymbol{r}(t)$ from non-synchronized measurements.

We answer this question in the particular setting in which a moving robot obtains range measurements from static and known anchors. In particular, we do not require the measurements to be perfectly synchronized, nor to be uniformly distributed in time. We make the realistic assumption that the robot can only measure one range at a time and limit ourselves to smooth trajectories—in particular, we focus on bandlimited and polynomial trajectories. An example setup is sketched in Figure 1 (b). It is straight-forward to see that traditional lateration cannot provide us with recovery algorithms, and even less with uniqueness guarantees. One can instead resort to trajectory estimation algorithms, which provide either a probabilistic or deterministic description of the continuous trajectory, but no guarantees for perfect recovery exist.

In this paper, we obtain a novel closed-form solution to the trajectory estimation problem, by relaxing the quadratic constraints. In addition, by studying the obtained linear system, we deduce necessary and sufficient conditions for trajectory recovery of the relaxed problem. This also provides a sufficient condition for trajectory recovery of the original (non-relaxed) problem.

II. Problem Formulation

To enable us to more accurately compare our contribution with existing techniques, we define our problem setup before discussing related work in the next section. Throughout the paper, we use regular lower case letters for variables (t), regular upper case letters for constants (K), bold lower case letters for column vectors ($\boldsymbol{c}$) and bold upper case for matrices ($\boldsymbol{C}$). By D we denote the robot's embedding dimension, which is fixed but in principle could be arbitrary. In this paper, we usually use $D=2$.

Our aim is to recover the position, $\boldsymbol{r}(t) \in \mathbb{R}^D$, of a moving device (e.g. a robot), for t in some given interval, $t \in \mathcal{I} \subset \mathbb{R}$. At a set of time instances $\{t_n : n = 0, \ldots, N-1\}$,

[†] Authors have equal contributions. Detailed contributions: AS designed research; MP, FD and AS designed the algorithm; FD and MP implemented the algorithm and simulations; FD performed literature review and worked on real data; MP proved algorithm properties, and analyzed simulations; FD, MP and AS wrote the paper.

All authors are with the Faculty of Computer and Communication Sciences, Ecole Polytechnique Fédérale de Lausanne, 1015 Lausanne, Switzerland `firstname.lastname@epfl.ch`

$t_n \in \mathcal{I}$, we measure the distance from the robot's (unknown) position $\boldsymbol{r_n} := \boldsymbol{r}(t_n)$ to one of M fixed anchors. We denote the anchor positions by $\boldsymbol{a}_m \in \mathbb{R}^D$, $m = 0, \ldots, M-1$, and assume that their locations are known. The measured distances are thus $d_n = ||\boldsymbol{r}_n - \boldsymbol{a}_{m_n}||$, where $||\cdot||$ is the Euclidean norm and m_n is the index of the anchor used at time n. In practice, we assume that we can measure distances $\tilde{d}_n$ corrupted by additive zero-mean Gaussian noise: $\tilde{d}_n = d_n + \epsilon_n$, where $\epsilon_n \sim \mathcal{N}(0, \sigma^2)$.

For ease of analysis, we assume that each t_n is different; in fact, this is a strength of our formulation as it means measurements from different anchors are not assumed to be synchronized and, since the t_n are real numbers, two consecutive t_n's can be arbitrarily close together.

For noisy measurements, the maximum likelihood estimator (MLE) of the device's position, at one time instant t_n, is given by the solution to the following optimization problem:

$$\arg\min_{\hat{\boldsymbol{r}}_n} \sum_{n=0}^{N-1} \left(\tilde{d}_n - ||\hat{\boldsymbol{r}}_n - \boldsymbol{a}_{m_n}|| \right)^2 . \quad (1)$$

In the noiseless case, the optimal $\hat{\boldsymbol{r}}_n$ are solutions to the following system of equations:

$$||\boldsymbol{a}_{m_n}||^2 - d_n^2 = 2\boldsymbol{a}_{m_n}^\top \hat{\boldsymbol{r}}_n - \hat{\boldsymbol{r}}_n^\top \hat{\boldsymbol{r}}_n, \quad (2)$$

for $n = 0, \ldots, N-1$. These problems are non convex and difficult to solve. In the following section, we will review some of the numerous techniques that have been devised to simplify these problems.

III. RELATED WORK

Note that although the focus of the paper is localization, we include the SLAM literature, as localization is a core part of SLAM and thus many methods can be transferred from one problem to the other.

A. Basic concepts for range-only localization

A core concept of many range-based localization algorithms is lateration, or how to estimate an object's location from distances to anchor points of known position.

This problem, formalized in (1), is not convex and to date, no algorithm is guaranteed to find the optimal solution. Therefore, (1) is commonly tackled with a standard nonlinear least squares solver, which in general leads to a local minimum. Alternatively, by squaring the two terms inside the brackets, the relaxed problem (dubbed Squared Range Least Squares, SRLS), can be optimally solved as shown in [6]. However, the result is no longer the MLE.

An important variation of the classical lateration problem is when the anchor locations are also unknown. Historically, this problem has often been analyzed through Euclidean Distance Matrices (EDMs), which contain the squared distances between points, and exhibit characteristic properties which can be exploited for denoising, completion and point recovery. [7]. Although EDMs by definition are more closely related to the SLAM problem, knowledge of anchor locations can be incorporated through the Procrustes transform, or in a semidefinite program, as proposed in [8].

B. Non-parametric trajectory recovery

Using the above methods, a moving object can only be localized at discrete time instances. This imposes a strong requirement on the number of measurements available at each such time instance, and does not ensure consistency between the subsequent position estimates.

Numerous algorithms solve these two issues by combining range measurements with movement estimates from inertial measurement units (IMUs) in standard filtering methods such as particle or Kalman filters [9]–[13]. Note that the obtained accuracy depends strongly on the sampling rate at which position updates can be computed, and problems can arise when IMU measurements are delivered at a much higher frequency than other modalities [14].

Sampling rate issues can be solved by continuous-time non-parametric models. A widely used approach [15]–[17] is to impose time consistency using Gaussian processes. Despite the continuous-time paradigm this method is considered similar to the filter based methods, because the objects position is predicted directly from the measurements. This means that no explicit parameter estimation is required. Numerous research efforts have been invested to make these computationally expensive methods more efficient, using for example Bayes trees for incremental reordering and just-in-time relinearization [18].

C. Parametric trajectory recovery

In this paper, we aim to recover a parametric model of the robot's position. A number of other works have proposed recovery of parametric trajectory models, predominantly using splines. A comprehensive review of this field is given in [14].

Li *et al.* [19] solve the classical SLAM problem, replacing the position update of the usual state-space equations with a continuous, parametric trajectory $x(t) = F(C_k, t)$. The authors consider polynomial basis functions identical to ours, and update the coefficients C_k for sliding time windows at time index k. They use a standard iterative solver which in general converges to a local minimum. Other methods [14], [20], [21] solve the same trajectory estimation problem, parameterizing the trajectory with B-spline basis functions. As B-splines have a local support, they automatically offer more flexibility in fitting complex trajectories without recursively updating the coefficients.

As opposed to these methods solving the more general SLAM framework with arbitrary measurement models, we show that by focusing on the modality of range measurements, a closed-form solution and recovery guarantees can be deduced.

Recently, trajectory estimation has been integrated in the traditional EDM framework with so-called *Kinetic EDMs* [22], where all points are considered to move on trajectories. In a similar spirit, in this paper we extend the traditional lateration framework to a device moving on a trajectory, however we incorporate knowledge of anchor points and we provide recovery guarantees.

IV. Method

In this section, we give an outline of our recovery algorithm. We first introduce the trajectory model and reformulate (2) to include it. Then, we relax the problem by reformulating it into a linear system of equations that can be solved with any linear solver.

We assume that the robot trajectory coordinates belong to some K-dimensional linear space of functions $\mathcal{F}$:

$$\boldsymbol{r}(t) = \sum_{k=0}^{K-1} \boldsymbol{c}_k f_k(t), \tag{3}$$

where $\{f_k : k = 0, \dots, K-1\}$ is a basis for $\mathcal{F}$, and the vectors $\boldsymbol{c}_k \in \mathbb{R}^D$ are the multidimensional basis coefficients.

In this work, we focus on bandlimited functions and polynomials. Both these models can approximate naturally occurring trajectories well. For example, bandlimited trajectories describe the oscillatory motion of a body around a stationary point. Polynomials cover constant speed motion ($K = 1$), constant acceleration (e.g. free fall, $K = 2$) and linearly changing acceleration ($K = 3$). For more complex trajectories, polynomials are the essential ingredient to the commonly used spline approximation.

For the space of bandlimited functions, we define the basis functions for odd degree K as

$$f_k(t) = \begin{cases} 2\cos(2\pi k t/\tau) & \text{for } k \text{ odd}, \\ 2\sin(2\pi k t/\tau) & \text{for } k \text{ even}, \ k > 0, \\ 1 & \text{for } k = 0, \end{cases}$$

where τ is the fixed period of the trajectory. For the space of polynomials we simply use the monomial basis $f_k(t) = t^k$.

We can now reformulate (2) in terms of the coefficients $\boldsymbol{c}_k$. By setting $\boldsymbol{f}_n := [f_0(t_n), \dots f_{K-1}(t_n)]^\top$ and $\boldsymbol{C} = [\boldsymbol{c}_0, \dots \boldsymbol{c}_{K-1}] \in \mathbb{R}^{D \times K}$, we can express the sampled positions in matrix form: $\boldsymbol{r}_n = \boldsymbol{C}\boldsymbol{f}_n$. The distances thus become $d_n = ||\boldsymbol{C}\boldsymbol{f}_n - \boldsymbol{a}_{m_n}||$. In the noiseless case would like to solve (2), which can be rephrased as the following system of equations:

$$\begin{aligned} ||\boldsymbol{a}_{m_n}||^2 - d2_n &= 2\boldsymbol{a}_{m_n}^\top \hat{\boldsymbol{C}} \boldsymbol{f}_n - \boldsymbol{f}_n^\top \boldsymbol{L} \boldsymbol{f}_n \\ \hat{\boldsymbol{L}} &= \hat{\boldsymbol{C}}^\top \hat{\boldsymbol{C}}, \end{aligned} \tag{4}$$

where $\boldsymbol{L}$ is introduced to separate the terms linear in $\boldsymbol{C}$ from the quadratic, non-convex terms in $\boldsymbol{C}$. A common approach is to make the problem convex, for example via semidefinite relaxation [23]. We propose to simply discard the second constraint entirely, which at first sight introduces an additional $K(K+1)$ independent variables. However, as we will show in Section V, the effective increase is in fact linear in K. Since we drop constraints, we know that even if there exists a unique solution to (4), the solution to the relaxed equation might not be unique. However, if the solution to the relaxed problem is unique, it is also the unique solution to (4).

Algorithm 1: Trajectory Reconstruction

```
Data: Anchor coordinates $\boldsymbol{a}_m$, distance measurements
      $d_m$, times and anchor indices $t_n$, $m_n$
Result: Trajectory coefficients $\hat{\boldsymbol{C}}$, empty if not unique
$\boldsymbol{f}_n \leftarrow [f_0(t_n) \quad \dots f_{K-1}(t_n)]$;
set up $\boldsymbol{T}_A, \boldsymbol{T}_F, \boldsymbol{b}$ as in (5);
$\hat{\boldsymbol{C}} \leftarrow [\,]$;
if condition (7) satsified then
    $\boldsymbol{U}, \boldsymbol{\Sigma}, \boldsymbol{V} \leftarrow$ SVD $(\boldsymbol{T}_F)$;
    $\boldsymbol{A} \leftarrow$ concatenate $(\boldsymbol{T}_A, \boldsymbol{U}\boldsymbol{\Sigma})$;
    $\hat{\boldsymbol{C}} \leftarrow$ linsolve $(\boldsymbol{A}, \boldsymbol{b})$;
end
```

In order to devise a recovery algorithm, we write the remaining equality constraint of (4) in linear form. To this end, we introduce the vectorized forms $\operatorname{vect}(\boldsymbol{C})$ and $\operatorname{vect}(\boldsymbol{L})$. Since $\boldsymbol{a}_{m_n}^\top \boldsymbol{C} \boldsymbol{f}_n$ is a scalar, it is equal to its trace and thus

$$\boldsymbol{a}_{m_n}^\top (\boldsymbol{C}\boldsymbol{f}_n) = \operatorname{tr}(\boldsymbol{a}_{m_n} \boldsymbol{f}_n^\top \boldsymbol{C}^\top) = \operatorname{vect}\left(\boldsymbol{a}_{m_n} \boldsymbol{f}_n^\top\right)^\top \operatorname{vect}(\boldsymbol{C}),$$

where the first equality comes from the commutativity of the trace, $\operatorname{tr}(\boldsymbol{A}^\top \boldsymbol{B}) = \operatorname{tr}(\boldsymbol{A}\boldsymbol{B}^\top)$, and the second from the fact that $\operatorname{tr}(\boldsymbol{A}^\top \boldsymbol{B}) = \operatorname{vect}(\boldsymbol{A})^\top \operatorname{vect}(\boldsymbol{B})$. Similarly,

$$\boldsymbol{f}_n^\top \boldsymbol{L} \boldsymbol{f}_n = \operatorname{vect}\left(\boldsymbol{f}_n \boldsymbol{f}_n^\top\right)^\top \operatorname{vect}(\boldsymbol{L}).$$

Therefore, the first equality constraint of (4) can be rewritten as

$$b_n = \begin{bmatrix} \operatorname{vect}\left(\boldsymbol{a}_{m_n} \boldsymbol{f}_n^\top\right)^\top & \operatorname{vect}\left(\boldsymbol{f}_n \boldsymbol{f}_n^\top\right)^\top \end{bmatrix} \begin{bmatrix} \operatorname{vect}(\boldsymbol{C}) \\ \frac{1}{2}\operatorname{vect}(\boldsymbol{L}) \end{bmatrix},$$

where we introduce $b_n := \frac{1}{2}\left(||\boldsymbol{a}_{m_n}||^2 - d_n^2\right)$ for simplicity. By concatenating the above equations, we obtain

$$\boldsymbol{b} = \begin{bmatrix} \boldsymbol{T}_A & \boldsymbol{T}_F \end{bmatrix} \left. \begin{bmatrix} \operatorname{vect}(\boldsymbol{C}) \\ \frac{1}{2}\operatorname{vect}(\boldsymbol{L}) \end{bmatrix} \right\}_{DK+K(K+1)}, \tag{5}$$

where we have introduced $[\boldsymbol{T}_A]_n = \operatorname{vect}\left(\boldsymbol{a}_{m_n} \boldsymbol{f}_n^\top\right)$ and $[\boldsymbol{T}_F]_n = \operatorname{vect}\left(\boldsymbol{f}_n \boldsymbol{f}_n^\top\right)$.

Since we only need to recover $\operatorname{vect}(\boldsymbol{C})$, we apply a Singular Value Decomposition (SVD) to $\boldsymbol{T}_F$, which reduces the dimension of the system. As we show in the next section, this does not affect the solution of $\hat{\boldsymbol{C}}$ and the dimensionality of $\boldsymbol{T}_F$ is reduced to $O(K)$ dimensions. The reduced system becomes

$$\boldsymbol{b} = \begin{bmatrix} \boldsymbol{T}_A & \boldsymbol{U}\boldsymbol{\Sigma} \end{bmatrix} \left. \begin{bmatrix} \operatorname{vect}(\boldsymbol{C}) \\ \frac{1}{2}\boldsymbol{V}^\top \operatorname{vect}(\boldsymbol{L}) \end{bmatrix} \right\}_{DK+r}, \tag{6}$$

which we solve with the right inverse. Here, $\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V} = \boldsymbol{T}_F$ is the SVD decomposition of $\boldsymbol{T}_F$ and $\boldsymbol{\Sigma}$ is an $r \times r$ diagonal matrix, with r the rank of $\boldsymbol{T}_F$.

This system in (6) can be solved with any linear solver. For clarity, the straight-forward recovery algorithm is summarized in Algorithm 1. Note that we check invertibility before solving the linear system, which is the topic of the next section,.

Based on the known complexities of the most costly components of our algorithm (that is SVD, matrix multiplications and solving the linear system) we calculate that the complexity of our algorithm is $O(N^2K + K^4)$.

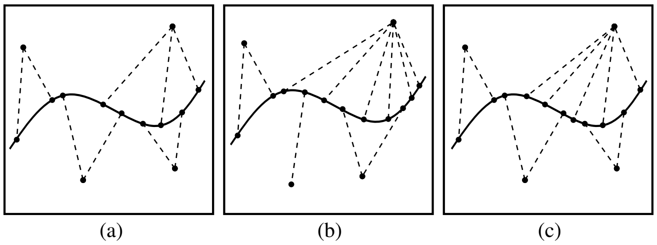


Fig. 2. Examples of sufficient and insufficient measurements, with model degree $K = 3$. (a) Condition (7) is satisfied, but we do not have $(D+2)K-1 = 11$ measurements. (b) We have 11 measurements, but condition (7) is not satisfied because too many of the measurements involve the same anchor. (c) Both conditions are satisfied and the recovery is possible.

V. Guarantees

In this section, we present necessary and sufficient conditions for trajectory recovery from the relaxed linear system. Since the unique solution to the relaxed problem is also the solution to the original problem, we obtain a sufficient condition for the original trajectory recovery from (4).

The crucial operation in Algorithm 1 is solving the linear system. This step is equivalent to finding the right inverse of $\begin{bmatrix}\boldsymbol{T}_A & \boldsymbol{U\Sigma}\end{bmatrix}$. Note that, since we assume we know the anchor positions perfectly, this matrix is not affected by noise. Thus, the whole analysis of this section is noise-agnostic. We do not, however, consider numerical errors.

First, let us observe that since $\boldsymbol{U\Sigma}$ is full rank, for $\begin{bmatrix}\boldsymbol{T}_A & \boldsymbol{U\Sigma}\end{bmatrix}$ to be full rank, we need

a) $\boldsymbol{T}_A$ to be full column rank, and
b) columns of $\boldsymbol{U}$ to be independent of the columns of $\boldsymbol{T}_A$.

When $\begin{bmatrix}\boldsymbol{T}_A & \boldsymbol{U\Sigma}\end{bmatrix}$ is invertible we can recover $\operatorname{vect}(\boldsymbol{C})$ (and $\boldsymbol{V}^\top \operatorname{vect}(\boldsymbol{L})$, which we discard). This means that the smallest number of measurements for which the recovery of $\boldsymbol{C}$ could be possible is $DK + r$, where r is the rank of $\boldsymbol{T}_F$.

Now, maintaining the assumptions of Section IV and adding two additional ones, we can precisely characterize the sets of measurements sufficient for recovery of $\boldsymbol{C}$. First, we additionally assume that the times t_n are generated by some continuous time process. In practice, any non-adversarial times, such as uniform or uniformly random times, can be used. Second, we also assume that no D anchors lie on the same (affine) subspace. This assumption is only slightly stronger than the very common requirement that the anchors span the (extended) space. Randomly placed anchors will satisfy these conditions almost surely. Under these weak assumptions, we can now introduce the main theoretical contribution of this paper.

Theorem 1. *Given at least $K(D+2)-1$ measurements (at different times), the matrix $\boldsymbol{C}$ can be uniquely recovered with probability one if*

$$\sum_{m=0}^{M-1} \min(k_m, K) \geq K(D+1), \tag{7}$$

where k_m is the number of measurements in which the m-th anchor is used. Moreover, if (7) is not satisfied, $\boldsymbol{C}$ cannot be uniquely reconstructed using Algorithm 1.

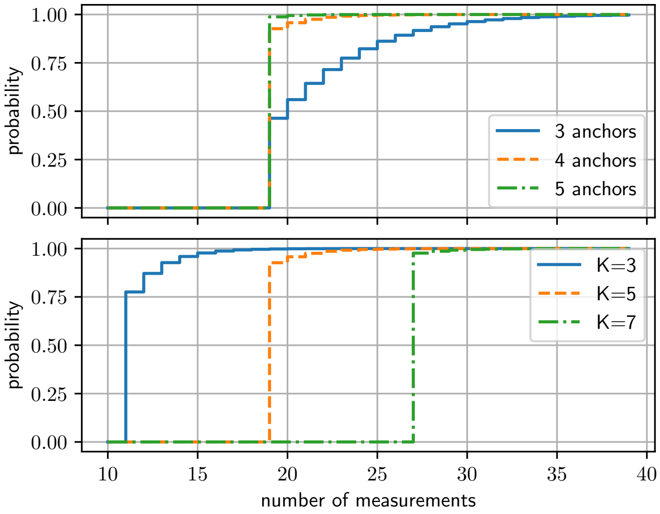


Fig. 3. Probability of recovering $\boldsymbol{C}$, with dimension $D = 2$. Upper plot: the trajectory degree is set to $K = 5$ and the number of anchors vary. We observe that the probability of recovering $\boldsymbol{C}$ grows with the number of anchors. This is because, the more anchors we use the higher the chance that the measurements spread uniformly and satisfy (7). Lower plot: the number of anchors is set to $M = 4$ and the trajectory degree varies. For Algorithm 1, (7) states that the minimum number of required measurements is $11, 19$ and 27 for trajectory complexities of $3, 5$ and 7, respectively.

Intuitively, (7) means that measurements cannot be arbitrarily distributed between anchors. In particular, if an anchor provides more than K measurements, only the first K have an effect on uniqueness. Moreover, unique recovery is not possible with measurements from less than $D+1$ anchors. To give a better understanding of this condition, we depict three examples in Figure 2.

Theorem 1 also implies that, in principle, we need only $O(DK)$ measurements to localize, even though we use a relaxation that would normally lead to a quadratic increase in the required data. Even without the relaxation, one would expect to need $(D+1)K$ measurements for recovery. Indeed, to recover a trajectory of complexity K, one can independently localize K points along it, and to localize a single point, we need $D+1$ distance measurements.

In the interest of clarity and space, a sketch of the proof of Theorem 1 can be found in the Appendix. The full proof will appear in a more abstract, extended publication that is currently under preparation.

A natural question to ask is, how likely it is to obtain a measurement set sufficient to recover $\boldsymbol{C}$. Unfortunately, the probability of the set of measurements satisfying Theorem 1 does not seem to have a closed form formula. Fortunately, it depends only on the partition of measurements between the M anchors, so it can be easily calculated numerically by counting partitions that satisfy (7).

Clearly, the probability of recovering $\boldsymbol{C}$ is non decreasing, because adding a new measurement can only increase the rank of $\begin{bmatrix}\boldsymbol{T}_A & \boldsymbol{T}_F\end{bmatrix}$. In practice, we observe that the probability is already large for $(D+2)K-1$ measurements, and grows with the number of measurements to 1. The probability also depends on the degree of the trajectory K and number of anchors (see Figure 3).

Finally, note that, in practice, the matrix might be ill conditioned when the anchors are almost co-linear, or many measurements to the same anchor are taken at almost the same time.

VI. Results

In the previous section, we have established conditions under which unique recovery of $\boldsymbol{C}$ is possible. In this section, we assume that those conditions are satisfied and investigate the trajectory recovery under more realistic, noisy conditions. We test the robustness of our algorithm to noise on simulated and real data.

A. *Noise model*

As mentioned before, we assume additive Gaussian noise on distance measurements $\tilde{d}_n = d_n + \epsilon_n$, where ϵ_n are i.i.d. random variables, $\epsilon_n \sim \mathcal{N}(0, \sigma)$.

An immediate problem comes from the fact that we use squared distances for recovery. This means that the additive noise becomes multiplicative and additive:

$$\tilde{d}_n^2 = d_n^2 + 2d_n\epsilon_n + \epsilon_n^2,$$

while our recovery method implicitly assumes only additive noise. Indeed, the pseudoinverse, in Algorithm 1, leads to an Ordinary Least Squares (OLS) optimisation, which is the MLE in the presence of additive Gaussian noise. To alleviate this problem, we propose a Weighted Least Squares (WLS) approach.

More precisely, the noisy version of the n-row of (5) is

$$d_n\epsilon_n + \frac{1}{2}\epsilon_n^2 + \begin{bmatrix}\boldsymbol{T}_A & \boldsymbol{T}_F\end{bmatrix}_n \begin{bmatrix}\operatorname{vect}(\boldsymbol{C}) \\ \frac{1}{2}\operatorname{vect}(\boldsymbol{L})\end{bmatrix} = b_n.$$

If the noise is small, the $\epsilon_n^2/2$ term is negligible. The term $d_n\epsilon_n$ can be normalized to follow an i.i.d. distribution more closely by dividing both sides of the equation by the measured distances $\tilde{d}_n$:

$$\frac{d_n}{\tilde{d}_n}\epsilon_n + \frac{1}{\tilde{d}_n}\begin{bmatrix}\boldsymbol{T}_A & \boldsymbol{T}_F\end{bmatrix}_n \begin{bmatrix}\operatorname{vect}(\boldsymbol{C}) \\ \frac{1}{2}\operatorname{vect}(\boldsymbol{L})\end{bmatrix} = \frac{b_n}{\tilde{d}_n}.$$

Usually, $d_n/\tilde{d}_n$ will be closer to one than d_n, and thus the weighed linear system is more suited to our model. In practice, to avoid dividing by extremely small numbers, we can add some small regularisation γ to the distance and divide by $\hat{d}_n + \gamma$.

B. *Simulations*

In this section, we report the root squared error E of reconstructing $\boldsymbol{C}$:

$$E(\boldsymbol{C} - \hat{\boldsymbol{C}}) = ||\boldsymbol{C} - \hat{\boldsymbol{C}}||_F, \tag{8}$$

where $||\cdot||_F$ is the Frobenius norm. For the chosen basis of bandlimited functions, this norm is equivalent to the power of the signal: using power as opposed to energy makes errors comparable between trajectories with different periods. For more intuition, see Figure 4.

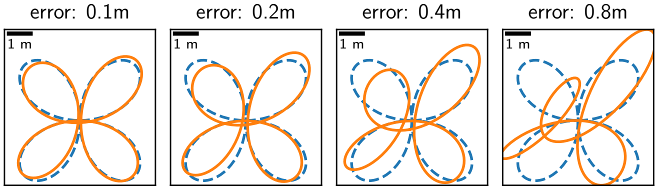


Fig. 4. Visualisation of the distance between trajectories. Original bandlimited trajectory of order $K = 7$ (dashed blue) and a randomly perturbed trajectory (solid orange). The Frobenius distance between the trajectory coefficients is displayed above each subplot.

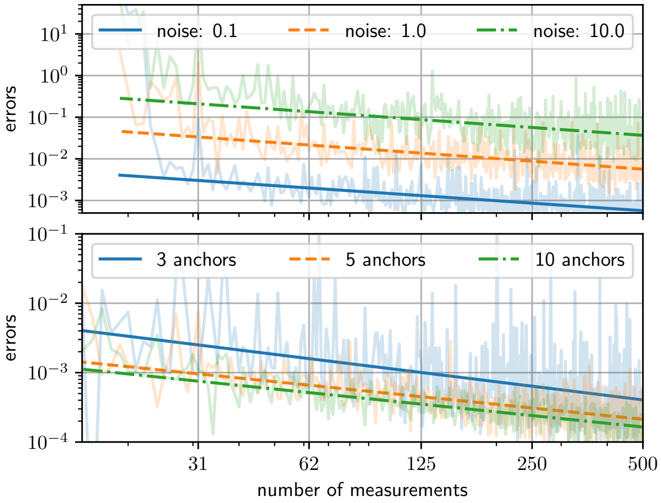


Fig. 5. Reconstruction from noisy measurements using the weighted system of equations. For clarity, slopes (dark lines) were fitted to the averages over 1000 simulations (light lines). The simulated distances were between 0.1 m and 10 m. Upper plot: the trajectory degree was set to $K = 5$ and $M = 4$ anchors were used; the magnitude of the noise is changing. In this setup, the minimum number of measurements required is 19. We can see that the algorithm is robust to noise starting from about 3× oversampling. Lower plot: the trajectory degree was set to $K = 3$ and noise magnitude was set to $\sigma = 1\,\text{m}$; the number of anchors is changing. The reconstruction error does not depend heavily on the number of anchors, but it has much higher variance for $D + 1 = 3$ anchors.

In the simulations, we take samples t_n uniformly in the interval $[0, \tau]$, and at each time we choose anchors uniformly at random. If we do not use all available measurements, we discard some of them uniformly at random. We fix $\tau = 2$. In Figure 5, we can see the coefficients reconstruction error obtained using the weighted reconstruction. We can see that the error decreases with oversampling, and the fitted slope is roughly -0.6. This means that for $10\times$ oversampling we get more than a $5\times$ reconstruction improvement. The regular (non-weighted) solver performed similarly, with a smaller improvement from oversampling.

C. *Real-world experiments*

We test our trajectory estimation algorithm on a dataset provided by Djugash *et al.* [24]. It consists of an autonomous lawnmower moving on a grass field, using ultra-wideband (UWB) signals to 4 stationary anchors for range measurements, and densely sampled kinematic GPS for ground truth. The distance measurements have an average standard deviation of ca. $0.5\,\text{m}$, with a tendency to overestimate [24].

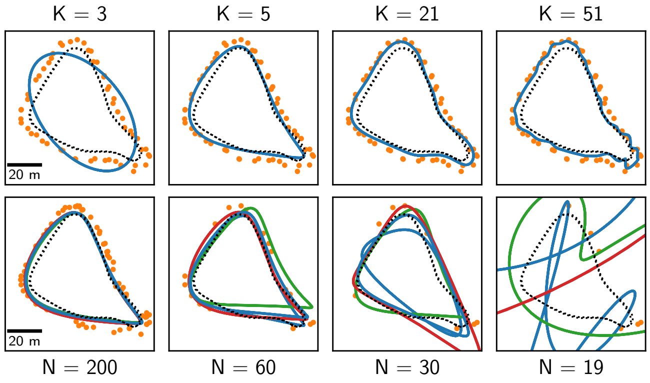


Fig. 6. Reconstruction accuracy of lawnmower trajectory using period $\tau = 54\,\text{s}$. In dotted black is the ground truth trajectory (from GPS), in blue is our reconstruction, and the orange dots show the SRLS estimates. The top row shows different complexities K using all 500 available distance measurements. In the bottom row, we fix K to 5 and drop measurements uniformly at random. The different colors show the reconstructions for different sets of uniformly chosen measurements.

The trajectory completed by the robot does not perfectly fit our models, but we will see that it can be approximated by the bandlimited model. We can estimate its period τ by visual inspection[1]. Using all range 500 measurements, we use Algorithm 1 to estimate the coefficients for different complexities K, and report the obtained trajectories in the first row of Figure 6. We see that the trajectory is well approximated with model degrees of $K = 5$ or higher.

Next, we fix K to the smallest sensible value for the given trajectory ($K = 5$) and test the performance of our algorithm when dropping measurements down to the minimum number required to satisfy Theorem 1 (19 measurements). The second row in Figure 6 shows that the obtained reconstruction quality remains satisfactory down to 30 measurements only, and is not so sensitive to the specific distance measurements selected. The estimates start to break down for smaller N; this is in accordance with our simulations, where we also observed a very noisy regime for few measurements.

For comparison, we also plot localization results with the point-wise SRLS method [6], using the latest distance measurements from $D+1$ anchors. The trajectory only starts to be recognizable from $N = 200$ measurements onwards, and the individual estimates are more noisy than ours.

VII. Conclusion and Future Work

In this work we have proposed a closed-form trajectory estimation method. Even though the method is based on a relaxation, the number of required measurements stays modest. Furthermore, our theory provides recovery guarantees and the framework eliminates the impractical assumption of perfectly synchronized measurements and dense deployment of anchors. We have demonstrated the performance of our method both on simulated and real data, showing in particular the advantage over point-wise lateration.

To the best of our knowledge, this paper presents the first recovery guarantees for trajectory recovery from range measurements. As such, we have focused on the theoretical aspects. In the future, we believe that the model can be extended to build more sophisticated and practically relevant localization algorithms. One natural extension would be to allow for a dynamically changing trajectory model, for example by using the proposed theory in spline approximations, or by dynamically updating the period τ. Finally, it would be interesting to investigate if the reconstruction could be improved with standard linear regression tricks, for example incorporating different regularizations or random projections.

Appendix

A. *Sketch of the proof of Theorem 1*

In order to prove Theorem 1, we need to prove that, given the right measurements (as stated in the theorem), the following three conditions are satisfied:

1) $\boldsymbol{T}_A$ is full rank,
2) $2K-1$ (independent) columns of $\boldsymbol{T}_F$ are independent of the columns of $\boldsymbol{T}_A$,
3) $\boldsymbol{T}_F$ has rank $2K-1$.

It is equivalent to prove these conditions for the modified matrices $\boldsymbol{T}'_A$ and $\boldsymbol{T}'_F$ created by moving the first K columns of $\boldsymbol{T}_F$ to $\boldsymbol{T}_A$. We use the three following facts to prove the above conditions:

1) A polynomial either has a discrete set of zeros or is zero everywhere. In the former case, the polynomial is non zero at a random point with probability 1.
2) The product of two polynomials of degree $< K$ is a polynomial of degree $< 2K-1$, and the same holds for bandlimited functions.
3) Matrix rank is sub-additive and, if $\boldsymbol{f}$ and $\boldsymbol{g}$ are independent, the matrices $\boldsymbol{f}\boldsymbol{f}^\top$and $\boldsymbol{g}\boldsymbol{g}^\top$are independent too.

The first condition is proved from the properties of the determinant of $\boldsymbol{T}'_A$, which is a polynomial in its entries. After expanding it in f_k, we can apply the first fact giving us a set of algebraic equations that $\boldsymbol{a}_m$ must not satisfy.

Due to the second fact, the columns of $\boldsymbol{T}'_F$ contain polynomials of order up to $2K-1$. Then, the first fact can be applied inductively over (some) columns of $\boldsymbol{T}'_F$ to obtain the second condition.

Finally, to prove the third condition, we use the second fact to bound the rank from above, and the third fact to bound the rank from below.

Since the first two conditions require the first fact, they are probabilistic in nature, and only the last condition is a purely deterministic result.

Acknowledgment

The authors would like to thank Ivan Dokmanić for many fruitful discussions and Martin Vetterli for his suggestions and constant guidance. We also thank Karen Adam for her constructive comments on the manuscript.

[1] This estimate could be improved through an iterative algorithm, but we did not find it necessary.

References

[1] F. Schenk and F. Fraundorfer, “RESLAM : A real-time robust edge-based SLAM system,” in *IEEE International Conference on Robotics and Automation (ICRA)*, 2019, pp. 154–160.

[2] A. Bahr, J. J. Leonard, and M. F. Fallon, “Cooperative localization for autonomous underwater vehicles,” *International Journal of Robotics Research*, vol. 28, no. 6, pp. 714–728, 2009.

[3] G. Vallicrosa, P. Ridao, D. Ribas, and A. Palomer, “Active Range-Only beacon localization for AUV homing,” in *IEEE International Conference on Intelligent Robots and Systems*, 2014, pp. 2286–2291.

[4] F. R. Fabresse, F. Caballero, L. Merino, and A. Ollero, “Active perception for 3D range-only simultaneous localization and mapping with UAVs,” in *International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, 2016, pp. 394–398.

[5] F. Dümbgen, C. Oeschger, M. Kolundzija, A. Scholefield, E. Girardin, J. Leuenberger, and S. Ayer, “Multi-Modal Probabilistic Indoor Localization on a Smartphone,” in *International Conference on Indoor Positioning and Indoor Navigation (IPIN) (to be presented)*, 2019. [Online]. Available: https://infoscience.epfl.ch/record/270245?ln=en

[6] A. Beck, P. Stoica, and J. Li, “Exact and approximate solutions of source localization problems,” *IEEE Transactions on Signal Processing*, vol. 56, no. 5, pp. 1770–1778, 2008.

[7] I. Dokmanic, R. Parhizkar, J. Ranieri, and M. Vetterli, “Euclidean Distance Matrices: Essential theory, algorithms, and applications,” *IEEE Signal Processing Magazine*, vol. 32, no. 6, pp. 12–30, 2015.

[8] S. Sremac, F. Wang, H. Wolkowicz, and L. Pettersson, “Noisy Euclidean Distance Matrix Completion with a Single Missing Node,” *Journal of Global Optimization*, 2019.

[9] J. L. Blanco, J. A. Fernández-Madrigal, and J. González, “Efficient probabilistic range-only SLAM,” in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, 2008, pp. 1017–1022.

[10] E. Menegatti, A. Zanella, S. Zilli, F. Zorzi, and E. Pagello, “Range-only slam with a mobile robot and a wireless sensor networks,” in *IEEE International Conference on Robotics and Automation (ICRA)*, 2009, pp. 8–14.

[11] Y. Song, M. Guan, W. P. Tay, C. L. Law, and C. Wen, “UWB/LiDAR Fusion For Cooperative Range-Only SLAM,” in *IEEE International Conference on Robotics and Automation (ICRA)*, 2019, pp. 6568–6574.

[12] A. Torres-González, J. R. Martinezdedios, and A. Ollero, “Range-only SLAM for robot-sensor network cooperation,” *Autonomous Robots*, vol. 42, no. 3, pp. 649–663, 2018.

[13] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. MIT press, 2005.

[14] P. Furgale, C. H. Tong, T. D. Barfoot, and G. Sibley, “Continuous-time batch trajectory estimation using temporal basis functions,” *International Journal of Robotics Research*, vol. 34, no. 14, pp. 1688–1710, 2015.

[15] C. H. Tong, P. Furgale, and T. D. Barfoot, “Gaussian Process Gauss-Newton for non-parametric simultaneous localization and mapping,” *International Journal of Robotics Research*, vol. 32, no. 5, pp. 507–525, 2013.

[16] J. Dong, M. Mukadam, B. Boots, and F. Dellaert, “Sparse Gaussian processes on matrix lie groups: A unified framework for optimizing continuous-time trajectories,” *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6497–6504, 2018.

[17] T. D. Barfoot, C. H. Tong, and S. Särkkä, “Batch nonlinear continuous-time trajectory estimation as exactly sparse Gaussian process regression,” *Robotics: Science and Systems*, 2014.

[18] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. J. Leonard, and F. Dellaert, “iSAM2: Incremental Smoothing and Mapping Using the Bayes Tree,” *International Journal of Robotics Research*, vol. 31, no. 2, pp. 216–235, 2012.

[19] T. Li, H. Chen, S. Sun, and J. M. Corchado, “Joint Smoothing and Tracking Based on Continuous-Time Target Trajectory Function Fitting,” *IEEE Transactions on Automation Science and Engineering*, vol. 16, no. 3, pp. 1476–1483, 2019.

[20] E. Mueggler, G. Gallego, and D. Scaramuzza, “Continuous-time trajectory estimation for event-based vision sensors,” *Robotics: Science and Systems*, 2015.

[21] D. Droeschel and S. Behnke, “Efficient continuous-time SLAM for 3D lidar-based online mapping,” in *IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 5000–5007.

[22] P. Tabaghi, I. Dokmanić, and M. Vetterli, “Kinetic Euclidean Distance Matrices,” *arXiv preprint arXiv:1811.03193*, 2018.

[23] Z. Q. Luo, W. K. Ma, A. So, Y. Ye, and S. Zhang, “Semidefinite relaxation of quadratic optimization problems,” *IEEE Signal Processing Magazine*, vol. 27, no. 3, pp. 20–34, 2010.

[24] J. Djugash, B. Hamner, and S. Roth, “Navigating with Ranging Radios: Five Data Sets with Ground Truth,” *Journal of Field Robotics*, vol. 26, no. 9, pp. 689–695, 2009.