*Article*

The International Journal of Robotics Research
2017, Vol. 36(12) 1363–1386

DOI: 10.1177/0278364917732639
journals.sagepub.com/home/ijr

# TSLAM: Tethered simultaneous localization and mapping for mobile robots

**Patrick McGarey, Kirk MacTavish, François Pomerleau, and Timothy D Barfoot**

**Abstract**

*Tethered mobile robots are useful for exploration in steep, rugged, and dangerous terrain. A tether can provide a robot with robust communications, power, and mechanical support, but also constrains motion. In cluttered environments, the tether will wrap around a number of intermediate 'anchor points', complicating navigation. We show that by measuring the length of tether deployed and the bearing to the most recent anchor point, we can formulate a* tethered simultaneous localization and mapping *(TSLAM) problem that allows us to estimate the pose of the robot and the positions of the anchor points, using only low-cost, nonvisual sensors. This information is used by the robot to safely return along an outgoing trajectory while avoiding tether entanglement. We are motivated by TSLAM as a building block to aid conventional, camera, and laser-based approaches to simultaneous localization and mapping (SLAM), which tend to fail in dark and or dusty environments. Unlike conventional range-bearing SLAM, the TSLAM problem must account for the fact that the tether-length measurements are a function of the robot's pose and* all *the intermediate anchor-point positions. While this fact has implications on the sparsity that can be exploited in our method, we show that a solution to the TSLAM problem can still be found and formulate two approaches: (i) an online particle filter based on FastSLAM and (ii) an efficient, offline batch solution. We demonstrate that either method outperforms odometry alone, both in simulation and in experiments using our TReX (Tethered Robotic eXplorer) mobile robot operating in flat-indoor and steep-outdoor environments. For the indoor experiment, we compare each method using the same dataset with ground truth, showing that batch TSLAM outperforms particle-filter TSLAM in localization and mapping accuracy, owing to superior anchor-point detection, data association, and outlier rejection.*

**Keywords**

SLAM, nonvisual, localization, mapping, tethered robot, field robotics

## 1. Introduction

Mobile robots operating in extreme environments require robust power and communication, but are limited to finite, onboard resources. When outfitted with an electro-mechanical tether, a robot is able to leverage an off-board power source, while maintaining communication with a remote base station over a reliable wired network (Wettergreen et al., 1993). The tether also provides safety, stability, and support, allowing a tethered robot to access terrain that is dangerous or impossible for conventional platforms (Matthews and Nesnas, 2012). For example, steep cliffs, dams, and disaster sites can be safely explored by tethered robots. While a supportive tether can reduce onboard requirements for the robot, it introduces challenging constraints to navigation (Sinden, 1990).

When a robot navigates through cluttered environments, its tether will contact and wrap around obstacles, forming intermediate anchor points (IAPs). To return to its base station or start position, the robot must sequentially unwrap from each added IAP (excluding its initial anchor). To avoid tether entanglement, the robot must return along a route that is homotopic to its outgoing path (i.e., retracing the outbound trajectory) (Teshnizi and Shell, 2014). Figure 1 illustrates such a scenario in a simple test environment. Accordingly, safe navigation requires localization of both the robot and any 'active' IAPs in the environment. An active IAP refers to any obstacle currently in contact with the tether. To address the problem of safe navigation for a tethered robot, we introduce tethered simultaneous localization and mapping (TSLAM), a novel estimation technique utilizing low-cost, nonvisual sensors to jointly estimate the state of the robot and IAPs. TSLAM incorporates measurements of

Institute for Aerospace Studies, University of Toronto, Canada

**Corresponding author:**
Patrick McGarey, Institute for Aerospace Studies, University of Toronto, 4925 Dufferin Street, Toronto, Ontario M3H 5T6, Canada.
Email: patrick.mcgarey@robotics.utias.utoronto.ca

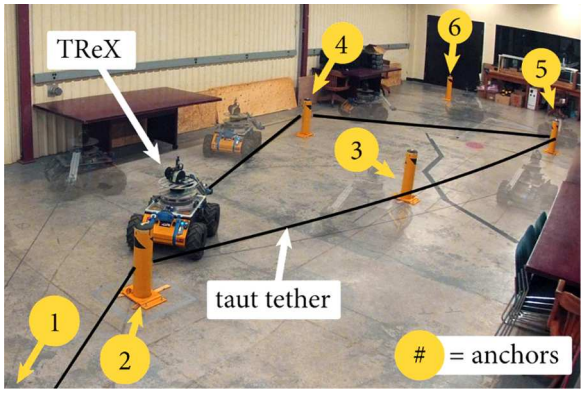


**Fig. 1.** *The tether problem.* The goal of TSLAM is to estimate the robot pose as well as positions of anchors based on measurements of tether length, bearing-to-anchor, and odometry for the purpose of avoiding tether entanglement. TReX is shown navigating through a cluttered test environment, which illustrates the importance of localizing the robot and mapping anchors (indicated with yellow markers). The active anchor sequence is 1,2,5,4, where 1 is not visible.

deployed tether length and bearing to the current anchor as a means to reduce odometry errors for a wheeled robot. The same measurements are also employed to determine the presence, position, and sequence of added IAPs within a local map. TSLAM differs from conventional range-bearing simultaneous localization and mapping (SLAM) in that our tether-length measurement involves *all* IAPs currently in contact with the tether. In conventional range-bearing SLAM, exteroceptive measurements are a function of a single robot pose and landmark, which yields a sparsity that can be exploited in estimation. In TSLAM, the measurement contribution is dense. Therefore, our observation model must either (i) make an approximation that the position of prior IAPs contributes negligible uncertainty or (ii) properly account for the position uncertainty across all IAPs touching the tether at any given time. In visual SLAM, where many thousands of landmarks might be observed, this type of density would make the estimation problem intractable. However, since we expect to observe substantially fewer IAPs in a single traverse, we show that the TSLAM problem is solvable using (i) a particle-filter implementation, which we originally proposed in McGarey et al. (2016) or (ii) an efficient batch formulation, introduced herein.

We evaluate the particle-filter and batch-TSLAM approaches using simulated and experimental data with the Tethered Robotic eXplorer (TReX) platform operating in cluttered environments. TReX, seen in Figure 1, is a custom-built, multi-sensor, tethered robot that can be used to explore extreme environments (McGarey et al., 2015). The sensors used in our estimation are low-cost and nonvisual, which means that they work in a wide variety of environments. Therefore, we are motivated to develop TSLAM in order to aid conventional SLAM in environments that are known to be problematic (e.g., dark and dusty). Our main contributions are to (i) formulate the general TSLAM problem, (ii) present two approaches to solve the problem, and (iii) compare the accuracy of each approach using the same dataset with ground truth. Overall, the results demonstrate that either method outperforms odometry alone, while providing a useful map of IAPs using only nonvisual measurements.

The organization of this article is as follows. Section 2 provides a brief background of SLAM, prior work on tethered mobile robots, and current developments in tether-based localization. Section 3 sets up the general TSLAM problem and details the specific methodology behind the particle-filter and batch-TSLAM implementations. Section 4 evaluates both approaches to TSLAM in simulation and experiments using TReX. Section 5 offers a discussion of the experiment results. Section 6 provides concluding remarks and outlines future work. Appendix A includes supplemental results and additional math.

## 2. Related work

In this paper, we are interested in developing an approach to landmark-based SLAM that is suited for nonvisual tether measurements. Smith et al. (1990) originally formulated a solution to the general SLAM problem based on the use of an extended Kalman filter (EKF). A modern survey of this technique, showing advances to landmark-based SLAM, was later made by Durrant-Whyte and Bailey (2006) and Bailey and Durrant-Whyte (2006). In our case, we propose a solution to the tethered-SLAM problem that is similar to GraphSLAM (Thrun and Montemerlo, 2006), a well-known example of the batch-SLAM technique pioneered by Lu and Milios (1997). SLAM itself is a modern version of the classic bundle-adjustment problem introduced for automated photogrammetry by Brown (1958), whose work was later popularized by Triggs et al. (2000). The principal difference between TSLAM and SLAM is that tether-length measurements involve a single robot pose and potentially all of the known IAPs, instead of just one landmark.

With respect to tethered mobile robots, the most notable examples from the recent literature include Dante II (Wettergreen et al., 1993) and Axel II (Matthews and Nesnas, 2012), two climbing robots developed to access nearly vertical terrain in planetary environments. McGarey et al. (2015) provides an extensive review of other tethered mobile robots.

Tether measurements have been used in line-of-sight localization for robots by Rajan et al. (2014) and Kumar and Richardson (2008). However, these experiments employed brute-force localization and did not use probabilistic machinery. Later, Corominas Murtra and Tur (2013) used wheel odometry, tether length, and an inertial measurement unit to localize a pipe-inspection robot, where tether length was used to limit uncertainty about the distance traveled in

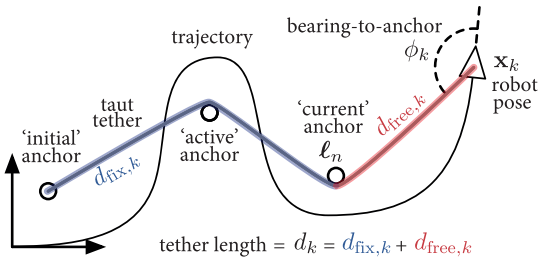


**Fig. 2.** *The TSLAM problem.* The pose of the robot and positions of the intermediate anchor points (IAPs) are unknown and will be estimated using tether length, bearing-to-anchor angle, and odometry gathered along the trajectory. We assume a planar environment and that the tether remains taut as it wraps around IAPs (represented as points and illustrated as circles). The tether length is divided into 'fixed' and 'free' lengths (i.e., tether contacting active IAPs and tether between the current IAP and robot), which are highlighted in blue and red, respectively. The total deployed tether length is $d_k$.

a pipe. However, this work was concerned with localization alone and no attempt was made to detect and map tether contacts (i.e., IAPs) within the pipe.

The key idea is that we are interested in estimating the state of the tether in our SLAM formulation, which is similar in respect to performing SLAM in the robot's configuration space (Klingensmith et al. (2016) provides a popular example for robotic arms). In McGarey et al. (2016), we introduced the first tether-based localization and mapping problem and proposed a particle filter inspired by FastSLAM (Montemerlo et al., 2002), where many particles, each with their own map of IAPs (represented as discrete Gaussians), capture the multi-modal uncertainty related to measuring IAPs (i.e., was an IAP added, reobserved, or detached from the tether?). We expand on that paper here and propose a novel, batch formulation. Furthermore, we compare the performance of each approach using the same dataset with ground truth.

## 3. TSLAM

In this section, we formulate the TSLAM problem and introduce several variables that will be used throughout this paper. The basic TSLAM problem is illustrated in Figure 2, where the robot's trajectory on a plane is given as

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_1 \\ \vdots \\ \mathbf{x}_K \end{bmatrix}, \quad \mathbf{x}_k = \begin{bmatrix} x_k \\ y_k \\ \theta_k \end{bmatrix} \tag{1}$$

with $k$ the discrete time-step index and $K$ its maximum. The robot's pose on the plane is described by a 2D position, $x_k$ and $y_k$, and orientation, $\theta_k$. We assume that the initial robot pose is $\mathbf{x}_0 = (0, 0, 0)$.

The ordered list of IAPs contacted by a tether is

$$\boldsymbol{\ell} = \begin{bmatrix} \boldsymbol{\ell}_1 \\ \vdots \\ \boldsymbol{\ell}_N \end{bmatrix}, \quad \boldsymbol{\ell}_n = \begin{bmatrix} x_n \\ y_n \end{bmatrix} \tag{2}$$

where $n$ is the 'current' IAP (first along the tether from the robot) and $N$ is the maximum number of IAPs observed in a sortie. We assume that the tether remains taut throughout movement and approximate the IAPs as zero-radius points, where a series of points would be used to represent a large obstacle. The (exteroceptive) tether measurements are

$$\mathbf{y} = \begin{bmatrix} \mathbf{y}_1 \\ \vdots \\ \mathbf{y}_K \end{bmatrix}, \quad \mathbf{y}_k = \begin{bmatrix} d_k \\ \phi_k \end{bmatrix} \tag{3}$$

which are a nonlinear function, $\mathbf{g}(\cdot)$, of the current robot pose and (potentially) *all* of the IAPs

$$\mathbf{y}_k = \mathbf{g}(\mathbf{x}_k, \boldsymbol{\ell}) + \mathbf{n}_k \tag{4}$$

where $\mathbf{n}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_k)$ is zero-mean, Gaussian measurement noise with covariance $\mathbf{R}_k$. The robot makes tether measurements according to the model shown in Figure 3. The bearing-to-anchor measurement model, $\phi_k$, which is a function of a single IAP, $\boldsymbol{\ell}_n$, and robot pose, $\mathbf{x}_k$, is written as

$$\phi_k = \text{atan2}(y_n - y_k, x_n - x_k) - \theta_k \tag{5}$$

The tether-length measurement model, $d_k$, is made up of 'fixed' and 'free' lengths, where

$$d_k = d_{\text{fix},k} + d_{\text{free},k} \tag{6}$$

The fixed length of tether wrapped around active IAPs is written as a sum of normal distances

$$d_{\text{fix},k} = \|\boldsymbol{\ell}_1 - \boldsymbol{\ell}_2\| + \cdots + \|\boldsymbol{\ell}_{n-1} - \boldsymbol{\ell}_n\| \tag{7}$$

where the active IAPs are those currently in contact with the tether. Active IAPs include $\boldsymbol{\ell}_1$ and $\boldsymbol{\ell}_n$ (the initial and current IAP). If only one IAP is active (i.e., when $\boldsymbol{\ell}_n$ is $\boldsymbol{\ell}_1$), the fixed length is zero. We note that the sequence of IAPs between, but not including, $\boldsymbol{\ell}_1$ and $\boldsymbol{\ell}_n$ can be represented differently, depending on the estimation method used. We detail how each method handles this sequence in Sections 3.1 and 3.2.

The free length of tether is the distance between the current IAP and the robot's position

$$d_{\text{free},k} = \left\| \boldsymbol{\ell}_n - \begin{bmatrix} x_k \\ y_k \end{bmatrix} \right\| \tag{8}$$

The fact that $d_k$ involves *all* active IAPs highlights the subtle difference between TSLAM and range-bearing SLAM; in conventional, range-bearing SLAM, each measurement is typically a function of the current robot pose and *just one* landmark.

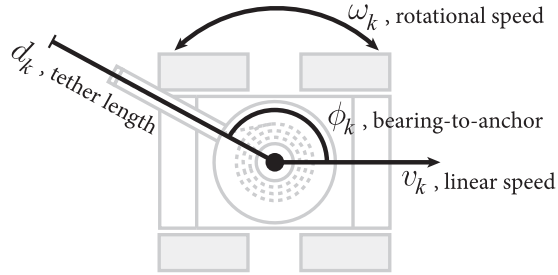


**Fig. 3.** *Tethered robot model.* This model is based on the TReX robot and shows the measurements and inputs used in TSLAM. We assume the robot to be a skid-steered robot with a kinematic model represented by a unicycle. At some time, $k$, the robot can measure its tether's bearing-to-anchor, $\phi_k$, with respect to the robot's forward driving direction. The tether length, $d_k$, is measured from the center of the robot, through all added intermediate anchor points (IAPs), to the initial anchor. The linear and rotational speeds, $v_k$ and $\omega_k$, are the body-centric input velocities given by equation (9).

With respect to the motion model, the inputs are provided as body-centric velocities, $\mathbf{v}_k$

$$\mathbf{v} = \begin{bmatrix} \mathbf{v}_1 \\ \vdots \\ \mathbf{v}_K \end{bmatrix}, \quad \mathbf{v}_k = \begin{bmatrix} v_k \\ \omega_k \end{bmatrix} \tag{9}$$

where $v_k$ is the linear translational speed and $\omega_k$ is the rotational speed, as illustrated in Figure 3. The nonlinear motion model, $\mathbf{f}(\cdot)$, is of the form

$$\mathbf{x}_k = \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{v}_k, \mathbf{w}_k) \tag{10}$$

where $\mathbf{w}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_k)$ is zero-mean, Gaussian process (i.e., measurement) noise with covariance $\mathbf{Q}_k$.

With general terminology defined, we would like to estimate

$$p(\mathbf{x}, \boldsymbol{\ell} | \mathbf{y}, \mathbf{v}) \tag{11}$$

the joint likelihood of the robot's trajectory and encountered IAPs, given the nonvisual measurements.

In this article, we evaluate two different solutions to the TSLAM problem: (i) an online particle-filter implementation (McGarey et al., 2016) and (ii) a new, offline[1] batch formulation. Supplemental math for TSLAM is provided in Appendix A.

## 3.1. Particle-filter method

Our first approach to the TSLAM problem uses an online particle filter, which is motivated by the idea that uncertainty about the real-time detection of new IAPs and tether wrap direction around a newly added IAP is multi-modal. In other words, if we are unsure about the direction that the robot drives around an obstacle, we want to use particles to represent the multi-modal decision and make data associations at the current time. Accordingly, we can later remove

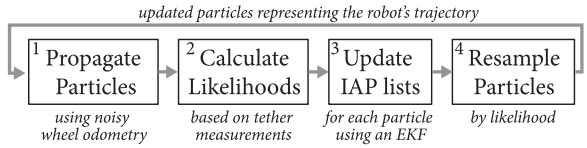


**Fig. 4.** *Particle-filter pipeline.* Our particle-filter approach is based on the FastSLAM algorithm and iterates through the steps shown. Each particle, representing a belief of the robot's trajectory, stores and updates its own list of IAPs similar to the map update step in FastSLAM.
EKF: extended Kalman filter; IAP: intermediate anchor point.

particles with poor data associations and keep particles that best represent the true state of the robot and the IAP map. Later, we will show a batch solution to TSLAM that uses an easier uni-modal representation; the batch solution benefits from making data associations with all the data at once so a multi-modal representation is no longer necessary. For the online TSLAM problem, we implement a particle-filter approach based on FastSLAM (Montemerlo et al., 2002), which uses discrete particles to represent the robot's state and Gaussians for individual IAP positions. FastSLAM is a specific approach to the more general Monte Carlo localization problem (Thrun et al., 2001), which uses discrete samples (e.g., particles), each with varying maps of the environment, to jointly represent the state. The FastSLAM algorithm iterates through a series of steps, as shown in Figure 4. The underlying motivation to use FastSLAM is that the joint likelihood of the robot's trajectory and the list of IAPs can be factored according to

$$p(\mathbf{x}, \boldsymbol{\ell} | \mathbf{y}, \mathbf{v}) = \underbrace{p(\mathbf{x} | \mathbf{y}, \mathbf{v})}_{\text{particles}} \underbrace{p(\boldsymbol{\ell} | \mathbf{x}, \mathbf{y})}_{\text{Gaussians}} \tag{12}$$

where, as per FastSLAM, landmarks (i.e., IAPs) are represented as Gaussians having both a mean (i.e., physical location) and an associated covariance. This setup allows for estimating and bookkeeping IAP positions using an EKF. For the particle-filter problem, we consider $\boldsymbol{\ell}$ to contain a set of Gaussians representing all active IAPs

$$\boldsymbol{\ell} = \begin{bmatrix} \boldsymbol{\ell}_1 \\ \vdots \\ \boldsymbol{\ell}_N \end{bmatrix}, \quad \boldsymbol{\ell}_n \sim \mathcal{N}(\boldsymbol{\mu}_n, \boldsymbol{\Sigma}_n) \tag{13}$$

where $N$ is the maximum number of observed IAPs, and a single Gaussian, $\boldsymbol{\ell}_n$, has a $2 \times 1$ mean, $\boldsymbol{\mu}_n$ (i.e., position on the plane), and a $2 \times 2$ covariance, $\boldsymbol{\Sigma}_n$.

*3.1.1. Adapting FastSLAM.* This section describes how we adapt our problem to work with the FastSLAM algorithm. In FastSLAM, measurements depend on just one landmark (i.e., IAP) and one robot pose, meaning that landmark measurements are conditionally independent (e.g., a range or

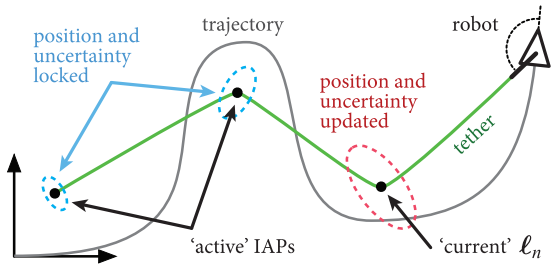


**Fig. 5.** *Updating intermediate anchor points (IAPs).* The approximation made to preserve measurement independence limits the ability to update the position and uncertainty to just the current anchor, $\boldsymbol{\ell}_n$. The 'active' anchors remain locked to updates until they are reobserved on the robot's return trajectory.

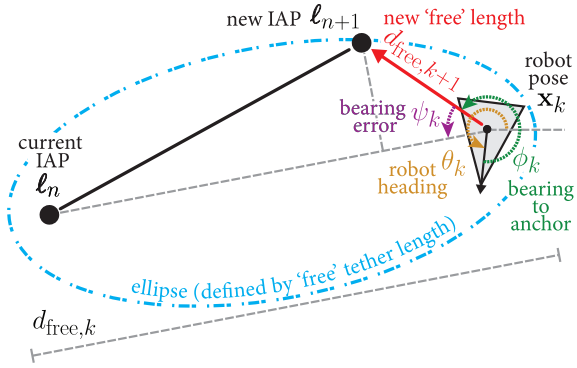


**Fig. 6.** *Ellipse measurement model.* An ellipse contains all possible locations for a newly added intermediate anchor point (IAP), $\boldsymbol{\ell}_{n+1}$, given the fixed length, $d_{\text{free},k}$, pose, $\mathbf{x}_k$, and current IAP, $\boldsymbol{\ell}_n$. The bearing-to-anchor measurement, $\phi_k$ and robot heading, $\theta_k$, combine to give a global bearing measurement to a unique location on the ellipse. The difference between this global angle and the bearing to $\boldsymbol{\ell}_n$ gives the bearing-to-anchor error, $\psi_k$. With $\psi_k$, $\mathbf{x}_k$, $\boldsymbol{\ell}_n$, and $d_{\text{free},k}$, the new free distance, $d_{\text{free},k+1}$, from the robot to the newly added IAP is found using equation (17). Lastly, the mean of $\boldsymbol{\ell}_{n+1}$, $\boldsymbol{\mu}_{n+1}$, can be calculated as in equation (16).

bearing sensor measurement to a landmark is not influenced by any other landmark). This fundamental assumption allows the further factorization of $p(\boldsymbol{\ell}|\mathbf{x},\mathbf{y})$ into a product of individual Gaussians, one for each IAP. To preserve this factorization, we make an approximation that IAPs are conditionally independent. The measurement likelihood is approximated as

$$p(\mathbf{y}_k|\mathbf{x}_k,\boldsymbol{\ell}) \approx p(\mathbf{y}_k|\mathbf{x}_k,\boldsymbol{\ell}_n) \tag{14}$$

where just the current IAP, $\boldsymbol{\ell}_n$ (subscript $n$ denotes the IAP closest along the tether from the robot), influences the measurement at time $k$. Figure 5 illustrates the impact of this approximation on updating the list or map of IAPs.

Given equation (14), it follows that the IAP map likelihood becomes

$$p(\boldsymbol{\ell}|\mathbf{x},\mathbf{y}) \approx \prod_{n=1}^{N} p(\boldsymbol{\ell}_n|\mathbf{x},\mathbf{y}) \tag{15}$$

where $N$ is the maximum number of IAPs observed, noting that only tens of IAPs will be encountered during any given sortie, as opposed to thousands of visual landmarks in conventional SLAM. We note that this approximation is not normally made in FastSLAM, but is required for our tethered approach. As a consequence of limiting measurement uncertainty to just the current IAP, we acknowledge that the method is vulnerable to localization errors stemming from noisy measurements, and increasing with the quantity of IAPs encountered, which is further discussed in Section 5.

The approximation in this formulation highlights the critical difference between the particle filter and the batch method described in Section 3.2; in the batch method, tether measurements are allowed to be conditionally dependent (i.e., measurements can involve one robot pose and potentially all active IAPs), which better captures the true position and is more robust to outliers.

*3.1.2. Anchor detection.* In TSLAM, each measurement involves a single, 'direct' bearing-to-anchor measurement to the current IAP and an 'indirect' length measurement through all the active IAPs. Recalling equation (6), the tether length comprises 'fixed' and 'free' lengths. In this approach, IAPs are ordered by time of detection, which means that the sequence of 'active' IAPs from equation (7) is numerically ordered. However, this is not the case for the batch method, as we will later show.

Owing to the approximation made in the preceding section, our measurement model is tailored to estimate the position of a newly added IAP using the 'free' length of tether joining the robot and current anchor. The ellipse model in Figure 6 is used because it contains all possible locations for an, as yet, undetected IAP, given the robot pose, current list of IAPs, and tether measurements at time $k$. The robot pose and known IAP location define the foci of the ellipse, the 'free' length defines the ellipse boundary, and the bearing-to-anchor measurement points in the direction of a unique point on the ellipse (i.e., the newly added IAP). The process for determining a new IAP is:

1. Calculate $d_{\text{free},k} = d_k - d_{\text{fix},k}$.
2. A new IAP has been detected if $d_{\text{free},k}$ is greater than the distance between $\mathbf{x}_k$ and $\boldsymbol{\ell}_n$ (if not, the formed ellipse will be ill-conditioned).
3. Set up the ellipse with values from $\mathbf{x}_k$, $\boldsymbol{\ell}_n$, $d_{\text{free},k}$, and $\phi_k$ according to Figure 6.
4. Compute the position of the new IAP.

The mean (i.e., position) of $\boldsymbol{\ell}_{n+1}$ is

$$\boldsymbol{\mu}_{n+1} = \begin{bmatrix} x_k \\ y_k \end{bmatrix} - d_{\text{free},k+1} \begin{bmatrix} \cos(\phi_k + \theta_k) \\ \sin(\phi_k + \theta_k) \end{bmatrix} \tag{16}$$

which first requires calculating $d_{\text{free},k+1}$, the new 'free' distance from $\mathbf{x}_k$ to $\boldsymbol{\ell}_{n+1}$. To calculate $d_{\text{free},k+1}$, we need the angle, $\psi_k$, which is the bearing-to-anchor measurement error between the expected and measured $\phi_k$ in the global frame (see Figure 6). We solve for $d_{\text{free},k+1}$ as

$$d_{\text{free},k+1} = \frac{\left\| \begin{bmatrix} x_k \\ y_k \end{bmatrix} - \boldsymbol{\mu}_n \right\|^2 - (d_{\text{free},k})^2}{2(d_{\text{free},k}) - \left\| \begin{bmatrix} x_k \\ y_k \end{bmatrix} - \boldsymbol{\mu}_n \right\| \cos \psi_k} \tag{17}$$

*3.1.3. Applying particle weights.* To calculate individual particle weights, we use the corresponding uncertainty from the last update, where the weight of a particle, denoted by superscript $m$, is determined by the likelihood function

$$w_k^m \approx \int \underbrace{p(\mathbf{y}_k|\mathbf{x}_k^m, \boldsymbol{\ell}_n^m)}_{\substack{\text{observation} \\ \text{model}}} \underbrace{p(\boldsymbol{\ell}_n^m|\mathbf{x}^m, \mathbf{y}^m)}_{\substack{\text{Gaussian from} \\ \text{last update}}} \mathrm{d}\boldsymbol{\ell}_n^m \tag{18}$$

A weight is calculated for all $M$ particles representing the likelihood of the robot's trajectory. Weights are used during resampling in order to select (with greater likelihood) particles that best estimate the robot's trajectory and IAPs.

*3.1.4. Adding, updating, and removing anchors.* Recalling that data associations are made in real time, we take a probabilistic approach to determining whether a given particle should add, update, or remove an IAP from its list in order to represent all the possibilities. We split all particles into three different copies immediately after resampling. One copy adds a new IAP to its list using the ellipse measurement model, another will update its current IAP position, and yet another will remove an IAP from its list (analogous to loop closure in SLAM). The splitting approach is also beneficial because it is difficult to detect changes to IAPs, given that our ellipse model can be degenerate when measurement noise is prevalent. As an example, a particle with a list of $n$ current IAPs will spawn three copies of itself with lists of length $n+1$, $n$, and $n-1$ IAPs. Assuming that our physical robot added an anchor, the subset of split particles that added IAPs to their list will be more likely to survive iterative resampling. If the robot drives straight just after adding an anchor it could be quite a bit later (when the robot turns again) that the different data associations manifest themselves in different particle weights. Adding IAPs is similar to landmark initialization in FastSLAM.

In between resampling, just after the particles have split, every particle's list of IAPs is updated using a Kalman filter. As opposed to EKF-SLAM, which incorporates the robot's pose and all landmarks into the update, we only update the 2D position of the current IAP in a particle's list at any given time, since we are using many discrete particles to represent the robot's pose. Therefore, we only require that a sufficient number of particles be used to properly represent the likelihood of the robot's pose (in practice 500–1000 particles are used). During the update period, no particles add or remove IAPs from their list. A nonlinear observation model from equation (4) is used to supply an updated position estimate (i.e., mean) for the current IAP. We take the Jacobian of that observation model to linearize and compute a new covariance as per FastSLAM. Further details of FastSLAM can be found in Thrun et al. (2004). Additional details regarding our implementation can be found in Appendix A.2.

## 3.2. Batch method

Our batch-TSLAM formulation improves on the particle-filter approach by (i) considering uncertainty across all active IAPs and (ii) providing superior IAP detection, data association, and outlier rejection. Accordingly, we no longer need a multi-modal approach, as we did with the particle filter; we can now use a Gaussian (uni-modal) representation of measurement uncertainty. The following section describes the methodology behind our offline batch approach and outlines critical differences from the particle-filter method.

We take the usual maximum-a-posteriori approach and turn equation (11) into an optimization problem of the form

$$\left\{\mathbf{x}^\star, \boldsymbol{\ell}^\star\right\} = \arg\min_{\mathbf{x},\boldsymbol{\ell}} J(\mathbf{x}, \boldsymbol{\ell}) \tag{19}$$

where $J(\mathbf{x}, \boldsymbol{\ell}) = -p(\mathbf{x}, \boldsymbol{\ell}|\mathbf{v}, \mathbf{y}) + C$ is our objective function and constant $C$ represents the terms in the negative log-likelihood that are independent of $\mathbf{x}$ or $\boldsymbol{\ell}$. We use the classic approach to solve this maximum-a-posteriori optimization problem with a variant of the Gauss–Newton algorithm, where we start with initial guesses, $\mathbf{x}_{\text{op}}$ and $\boldsymbol{\ell}_{\text{op}}$, and write the design variables as perturbations of these

$$\mathbf{x} = \mathbf{x}_{\text{op}} + \delta\mathbf{x}, \quad \boldsymbol{\ell} = \boldsymbol{\ell}_{\text{op}} + \delta\boldsymbol{\ell} \tag{20}$$

where $\delta$ represents a small perturbation to the state that minimizes the cost function from equation (19). At each iteration, we arrive at a linear system of equations that we solve for the optimal perturbations, $\delta\mathbf{x}^\star$ and $\delta\boldsymbol{\ell}^\star$, to our initial guesses

$$\underbrace{\begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{12}^{\mathrm{T}} & \mathbf{A}_{22} \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} \delta\mathbf{x}^\star \\ \delta\boldsymbol{\ell}^\star \end{bmatrix}}_{\mathbf{z}} = \underbrace{\begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \end{bmatrix}}_{\mathbf{b}} \tag{21}$$

The optimal perturbations are then applied to the initial guesses

$$\mathbf{x}_{\text{op}} \leftarrow \mathbf{x}_{\text{op}} + \delta\mathbf{x}^\star, \quad \boldsymbol{\ell}_{\text{op}} \leftarrow \boldsymbol{\ell}_{\text{op}} + \delta\boldsymbol{\ell}^\star \tag{22}$$

and the entire procedure is iterated to convergence.

Looking to equation (21) for the SLAM case, we recall that $\mathbf{A}_{11}$ will be block-tridiagonal and $\mathbf{A}_{22}$ will be block-diagonal. It is the block-diagonal nature of $\mathbf{A}_{22}$ that is frequently exploited (e.g., Schur complement, sparse Cholesky) to arrive at an efficient SLAM solution. The reason that $\mathbf{A}_{22}$ is block-diagonal is that each measurement, $\mathbf{y}_k$, is a function of one robot pose and just one landmark, which is no longer true for TSLAM, as previously stated.

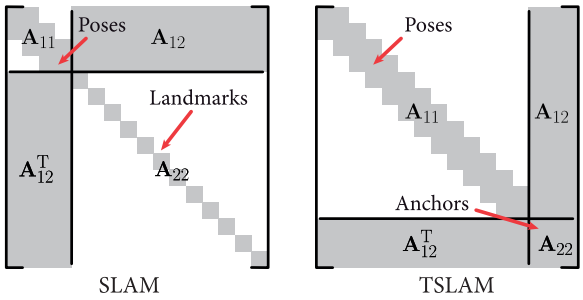


**Fig. 7.** *Comparing sparsity.* The primary sparsity patterns of the **A** matrix in equation (21) for the batch-SLAM and TSLAM problems. In batch TSLAM, the $\mathbf{A}_{22}$ block is usually dense but we have a relatively small number of intermediate anchor points (IAPs) and therefore propose to exploit the sparsity of $\mathbf{A}_{11}$ in our method.

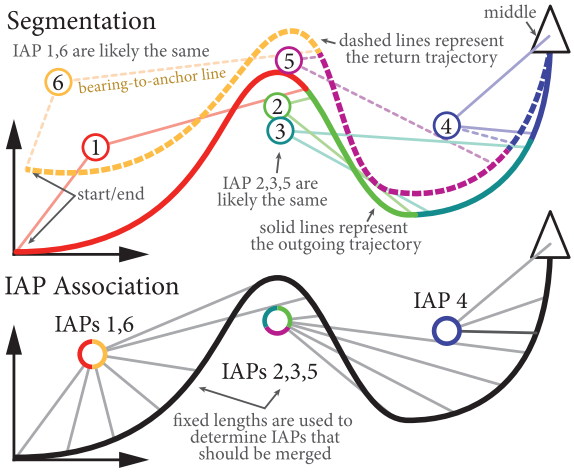


**Fig. 8.** *Segmentation and intermediate anchor point (IAP) association example.* The top map provides a sample output of subtrajectory segmentation and IAP detection. As the robot moves, the 'current' anchor point (first along tether from robot) changes. We segment the robot's trajectory into color-coded subtrajectories that share common anchors. IAPs are numbered in the chronological order of observation. Since odometry is prone to drift, more IAPs have been detected than actually exist. The bottom map is an example of a repaired trajectory after IAP association and merging is performed. This figure is best viewed in color.

In TSLAM, the tether-length measurement is a function of one robot pose and all active IAPs, as captured by equation (7). The result is that $\mathbf{A}_{22}$ is potentially dense for TSLAM. However, $\mathbf{A}_{11}$ is still block-tridiagonal, since it is only a function of the odometry measurements, which still involves just two consecutive robot poses.

In practice, owing to the finite length of tether spooled on the robot, a relatively small number of IAPs will be encountered during a sortie, as compared with the number of robot poses: $N \ll K$. This is the opposite of the usual SLAM case, where, for example, there may be a very large number of unique visual landmarks compared with the number of robot poses. Figure 7 compares the primary sparsity patterns of the batch-SLAM and TSLAM problems.[2] We propose to exploit the sparsity of $\mathbf{A}_{11}$ in this batch implementation of TSLAM to solve equation (21) efficiently. We do so using a sparse Cholesky decomposition of **A**

$$\mathbf{A} = \mathbf{L}\mathbf{L}^{\mathrm{T}} \tag{23}$$

with **L** the sparse, lower-triangular Cholesky factor. Then we solve $\mathbf{L}\mathbf{c} = \mathbf{b}$ for **c** and $\mathbf{L}^{\mathrm{T}}\mathbf{z} = \mathbf{c}$ for **z**. The complexity[3] of this entire procedure is $O(N^3 + N^2K)$ (at each iteration). More details on this and the general batch formulation are provided in Appendix A.2.

Thus far, we have described the general batch-TSLAM problem and our solution strategy, but have not addressed the detection of active IAPs, and the determination of the IAP sequence or tether interaction history (i.e., the order of tether-to-obstacle contacts, tracing from the robot back to the initial anchor).

*3.2.1. Anchor detection.* The previous section describes the overall strategy of setting up and solving the batch-TSLAM problem, but there are some additional issues to consider. Specifically, we need to be able to handle two important situations:

(i) *New IAP.* We need to detect when the tether encounters a new IAP that has never previously been touched.

(ii) *Old IAP.* We need to detect when the tether re-encounters an old IAP that has previously been touched (analogous to loop closure in SLAM).

Our approach to handling situation (i) is to carry out a segmentation of the robot's trajectory into subtrajectories that share a common, active IAP, as depicted in Figure 8. This approach is inspired by the 'clique' technique proposed by Howard (2008), which is used to separate sets of mutually consistent features. During segmentation, we make no attempt to deal with situation (ii) and instead assume that whenever the tether switches to a different IAP, it is a new one that has not been encountered previously. We then address situation (ii) during the batch solve by first minimizing the distance between IAPs with similar fixed lengths of tether, then merging those IAPs in close proximity using the concept of switchable constraints from Sünderhauf and Protzel (2012), implemented as a dynamic-covariance-scaling robust cost function (Agarwal et al., 2013). An overview of the batch-TSLAM pipeline is illustrated in Figure 9. The segmentation and IAP association steps are discussed next.

*3.2.2. Segmentation.* Trajectory segmentation is essentially the discretization of the environment that occurs when a tether contacts an anchor (Teshnizi and Shell, 2014). Prior to solving the batch-estimation problem, the trajectory of the robot must first be segmented into subtrajectories, each associated with an individual IAP. With the assumption that wheel odometry is a reasonable approximation of the

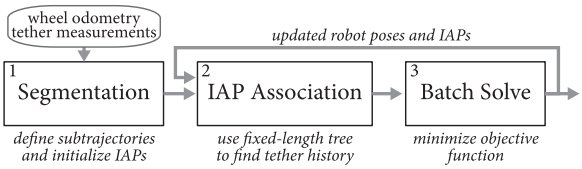


**Fig. 9.** *Batch pipeline.* (1) Using wheel odometry and tether measurements, the robot's trajectory is divided into subtrajectories, each associated with an individual intermediate anchor point (IAP). Our current pipeline performs the segmentation just once at the beginning because it is computationally expensive. (2) The tether interaction history is used to build the observation model, which changes depending on the IAP that is being observed. (3) The batch solve outputs the optimal perturbations to robot poses and IAP positions by minimizing the objective cost function in equation (19). Steps 2 and 3 iterate until a desired cost is reached.

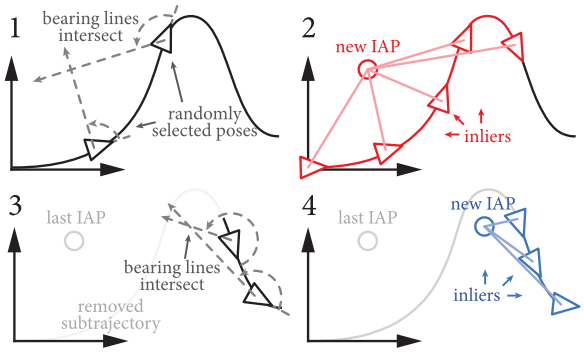


**Fig. 10.** *Random sample consensus (RANSAC) segmentation.* Using the trajectory from wheel odometry as a prior, segmentation requires the following steps. (1) Two poses are randomly selected and the intersection point of their bearing-to-anchor lines is used as a proposal intermediate anchor point (IAP). (2) Inliers will have a predicted bearing measurement to the proposed IAP that closely matches their recorded bearing measurement. (3) Consecutive inliers, or subtrajectories, are removed from the pool of samples in order to find other IAPs. (4) The process iterates until all poses have been associated with a single subtrajectory and IAP.

trajectory over short distances, our method uses random sample consensus (RANSAC) (Fischler and Bolles, 1981) to generate sample IAPs using only bearing-to-anchor measurements and odometry. Proposal IAPs are determined from the intersection of bearing-to-anchor lines from two sampled poses in a step-by-step process detailed in Figure 10. This type of RANSAC is similar in spirit to single-point RANSAC introduced by Scaramuzza et al. (2009). As opposed to the particle filter's real-time detection approach, with segmentation we (i) use *all* the measurements, and (ii) only require the 'direct' bearing-to-anchor measurement and odometry to determine potential IAP locations.

Once all poses have been matched to individual IAPs, a local optimization (Chum et al., 2003) is performed to resolve the position of each IAP using all bearing-to-anchor measurements for poses within a given subtrajectory. An example problem illustrating segmentation is shown in Figure 8. The segmentation step is executed just once at the beginning of the pipeline because it is computationally expensive; through future speedups, it should be possible to iterate this step as well.

*3.2.3. Anchor association.* The association or merging of re-encountered IAPs is required for loop closure. Owing to odometry bias and measurement noise, revisited IAPs may be initialized in different locations, as shown in Figure 8. For this reason, the approach of naively merging closely spaced IAPs is not initially helpful.

Instead, we use tether-length measurements to decide whether associations between IAPs should be made (e.g., a revisited IAP). When the robot revisits an IAP that was previously observed, it is possible that the two observations will be offset in position. However, the fixed-length measurements between the two IAPs should be similar. Making fixed-length associations allows us to determine the tether interaction history from any given IAP back to the initial anchor, which is used to build the fixed-length observation model in equation (7). In other words, since it is possible that a single IAP can be represented by two points on our map, we allow for a nonsequential ordering of the tether history.

The tether history from an IAP back to the initial anchor is found using a binary search tree (Bentley, 1975) constructed from fixed tether lengths. At any time, $k$, the fixed length can be determined from the total length measurement, $d_k$

$$d_{\text{fix},k} = d_k - d_{\text{free},k} \tag{24}$$

Since each subtrajectory involves a group of consecutive robot poses, we calculate an estimated fixed length to use in the search tree from the average value of $d_{\text{fix},k}$ for all poses in the subtrajectory group, with the initial IAP having a fixed length of zero. The specific method used to determine tether history is explained by the illustrated fixed-length tree in Figure 11.

Tether history impacts the $\mathbf{A}$ matrix in the linear system from equation (21). Recall from batch SLAM that

$$\mathbf{A}_{12} = \mathbf{G}_1^{-\mathrm{T}}\mathbf{R}^{-1}\mathbf{G}_2\,, \quad \mathbf{A}_{22} = \mathbf{G}_2^{-\mathrm{T}}\mathbf{R}^{-1}\mathbf{G}_2 \tag{25}$$

where the Jacobians of the measurement model with respect to all robot poses and IAP positions are $\mathbf{G}_1$ and $\mathbf{G}_2$, respectively. $\mathbf{R}^{-1}$ is the block-diagonal, inverse covariance on measurement noise. Since the Jacobian with respect to the robot's pose only involves the 'free' tether length and bearing-to-anchor, as in equations (5) and (8), $\mathbf{G}_1$ will still be block-diagonal. However, $\mathbf{G}_2$, which is dependent on tether history and all active IAPs, will potentially be dense. For our example problem, the sparsity patterns of $\mathbf{G}_1$ and $\mathbf{G}_2$ are illustrated in Figure 12. Details of the components in the batch linear system can be found in Appendix A.2.

With regard to loop closure, we assume that (i) IAPs are static and (ii) the robot's outgoing and return trajectories

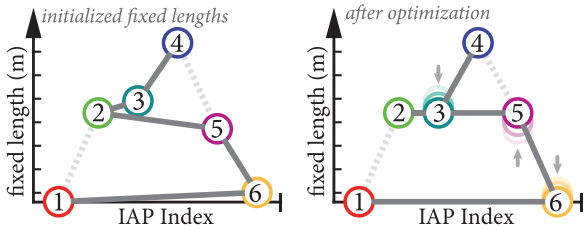


**Fig. 11.** *Fixed-length tree.* The left tree plot shows the initial state of the fixed-length tree for the example problem in Figure 8. The tether interaction history is drawn with a gray solid line between intermediate anchor points (IAPs). For any given IAP, the history can be determined by descending along the line as far as the initial IAP. For example, the tether history from IAP 4 back to the initial IAP 1 is the sequence 4, 3, 2, 5, 6, 1. The right tree plot is the optimized result of association and merging, where IAPs with similar fixed lengths are horizontally aligned.

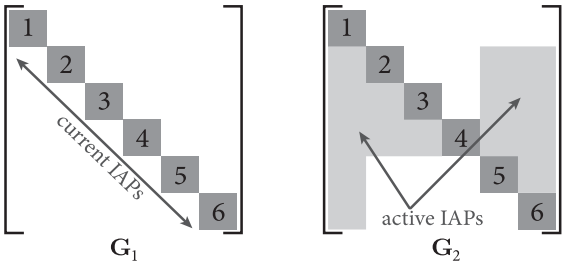


**Fig. 12.** *Block Jacobian sparsity.* The Jacobians of the measurement model with respect to robot poses, $\mathbf{G}_1$, and intermediate anchor points (IAPs), $\mathbf{G}_2$ for the example problem in Figure 8. $\mathbf{G}_1$ is block-diagonal, while $\mathbf{G}_2$ is more dense. The sparsity pattern of $\mathbf{G}_2$ follows from the tether history outlined in Figure 11. $\mathbf{G}_1$ and $\mathbf{G}_2$ are used to calculate the $\mathbf{A}$ matrix from equation (21).

are similar. Accordingly, we know that all active IAPs will be reobserved as the robot backtracks to the initial IAP. Revisited IAPs will have similar fixed lengths because they are initialized from tether measurements and not reliant on tether history. Building the fixed-length observation model from equation (7) allows IAPs with similar fixed lengths to be 'pulled' together as the measurement error is minimized, regardless of the sequence of observation. Recall that the particle-filter approach makes explicit loop closures when a particle removes an IAP from its sequential list—subsequent resampling causes a lack of IAP map diversity, as many particles share the same parent particle over time, making earlier choices hard to reverse. However, the implicit batch data association allows for superior recovery from early mapping mistakes, since loop closures can be reversed if they reduce the cost function.

We note that it is possible for physically distinct IAPs to have similar fixed lengths. This can happen if the robot backtracks on a path and then drives off in a different direction. As such, we do not make associations between newly detected IAPs and IAPs that were formerly observed but are no longer on the 'active' list. In this situation, the fixed-length tree would have two branches, where merges between branches are not allowed in our current implementation. In practice, we run the optimization as described so far, then later introduce a robust cost function between each link in the tree to merge only closely spaced IAPs. MacTavish and Barfoot (2015) provide an overview of different robust cost functions, demonstrating that switchable constraints (Sünderhauf and Protzel, 2012), implemented as dynamic covariance scaling (Agarwal et al., 2013), are ideal for its narrow region of influence (i.e., good outlier rejection) and fast convergence. The influence of this cost function approaches zero for IAPs that are distant, and is large for nearby IAPs that should be merged. This robust cost is added as an additional term in the overall optimization problem. For details on how dynamic covariance scaling is implemented, see Appendix A.2.

*3.2.4. Batch implementation.* As shown in Figure 9, the batch method is initialized with odometry and tether bearing-to-anchor measurements to segment the trajectory and initialize IAP estimates using RANSAC. Next, robot poses, IAP positions, and fixed lengths are optimized by gradient descent. We use a robust cost function on loop closures to merge IAPs with similar fixed lengths. The optimization iterates until a convergence criterion is met. In practice, we use an average change-in-cost threshold over iterations.

## 4. Experiments

We have performed a series of experiments, which range from simulation to outdoor tests using TReX, to evaluate these approaches to the TSLAM problem. First, we test the functionality and robustness of TSLAM with simulated data to show statistical performance over 100 trials. Then we experiment with TReX in a cluttered, indoor test environment to evaluate TSLAM with ground-truth data. Finally, we show TSLAM working outdoors during a field test in Sudbury, Ontario, Canada, where TReX drove through a sloped forest environment and added a series of IAPs around tree trunks.

### *4.1. Simulation*

Our particle-filter and batch-TSLAM approaches were initially tested and compared using simulated data. The datasets were generated using a random set of 20 potential IAPs distributed in a plane with five random waypoints for the robot to follow. The closest IAP to the robot is selected as the initial anchor. The robot drove a random path through the waypoints, adding anchors, and sensor measurements were recorded, to which we added Gaussian noise. The quantity and position of all active IAPs, including the initial IAP, are unknown to the estimator. For analysis, 100 trials were run on simulated data. For each trial, a unique trajectory and a set of measurements were generated with

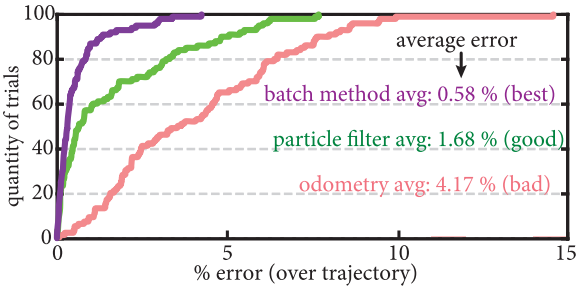


**Fig. 13.** *Simulation statistics.* (100 trials) The root mean square error at the end of each trajectory is divided by the distance traveled to show percentage error over the total trajectory. For each method, a cumulative distribution function shows the error over all trials. The batch method outperforms the particle filter, which outperforms odometry.

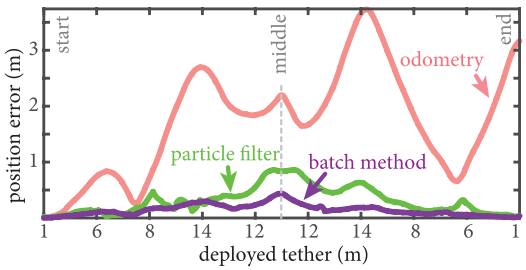


**Fig. 14.** *Simulated trial error.* For a single example from the 100 trial simulation, the root mean square position error shows that the particle-filter and batch-estimation errors both return to zero once all intermediate anchor points (IAPs) have been removed at the end of a trajectory. The batch method outperforms the particle filter. The trajectory maps in Figure 15 illustrate the estimated paths compared with ground truth.

ground truth. We use the same 100 simulated datasets to test the batch and particle-filter methods in order to make a fair statistical comparison.

To verify that TSLAM statistically outperforms wheel odometry, Figure 13 compares each method's localization accuracy over all 100 trials as percentage error over the full trajectory (i.e., how much the solution drifted after adding or removing anchors and returning to the original start position). We see that either approach to the TSLAM problem outperforms odometry alone, while the batch method is statistically more accurate than the particle filter overall. The reason, as previously stated, is related to superior IAP detection, data association, and outlier rejection. We show an example result from a single trial in Figures 14 and 15, which provide a position error plot and trajectory maps, respectively. Again, we see that each estimation approach outperforms odometry, while the batch method more closely aligns with ground truth than the particle-filter method. Supplemental results from the simulation, including a time-lapse animation of the particle filter and an illustration of batch segmentation and association, are available in Appendix A.1.

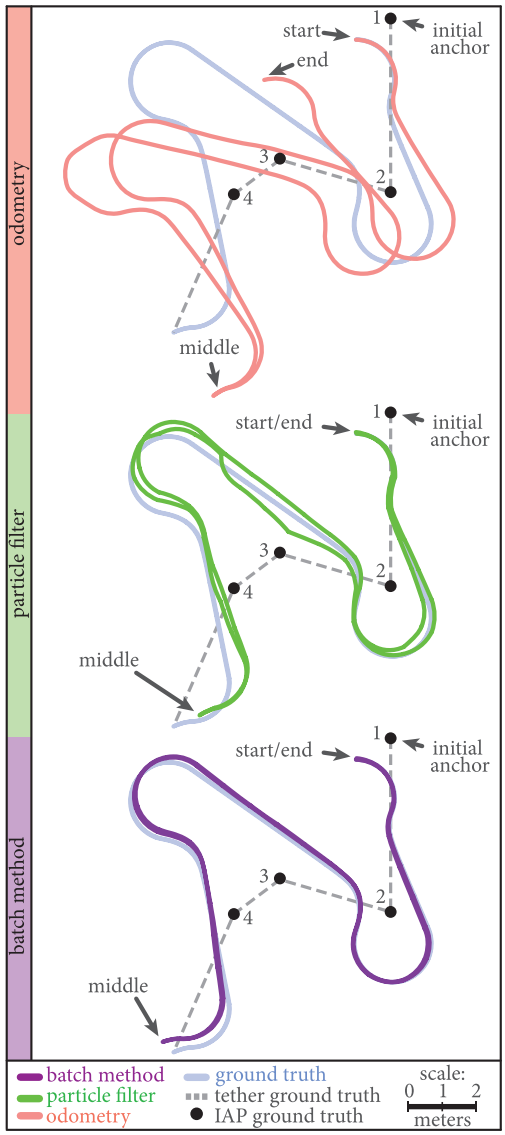


**Fig. 15.** *Simulated trial trajectory maps.* The maps compare each estimation method and odometry with ground truth. Each tethered simultaneous localization and mapping (TSLAM) approach outperforms odometry by reducing accumulated error. The batch method is most accurate. Position errors are shown in Figure 14. IAP: intermediate anchor point.

## 4.2. Test platform: TReX

We evaluated TSLAM with data collected from our Tethered Robotic eXplorer (TReX), a mobile robot capable of capturing 3D point-clouds of extreme environments, managing an onboard electromechanical tether, and measuring its tether length and bearing-to-anchor (McGarey et al., 2015). Figure 16 shows the platform setup (top image) and examples of TReX exploring steep terrain aided by an electromechanical tether (bottom image). TSLAM is possible, owing to the robot's unique, passively rotating tether arm, which enables bearing-to-anchor measurements to be made


**Fig. 16.** *TReX platform.* Top: major system components. Most importantly, the locations of tether-length and bearing-to-anchor measurements are indicated. Bottom: TReX navigating a set of rock faces. When the tether comes into contact with the rock, intermediate anchor points (IAPs) are created. We note that while the robot is equipped with a stereo camera, inertial measurement unit, and lidar, these sensors are not used for TSLAM because we are motivated by the idea of nonvisual SLAM as a building block for future work.

when the tether is taut, regardless of the vehicle's orientation or direction of applied tension. McGarey et al. (2015) provides more details of the system design and testing of TReX.

## 4.3. Indoor experiment

Three indoor experiments were conducted using TReX in a planar test environment with ground truth. Six static bollards (serving as IAPs) were distributed in our indoor workspace, with one serving as an initial anchor. The ground-truth position of all IAPs was determined using a Leica total station, which also provided position-only ground truth for a prism mounted on the rotational axis of TReX with millimeter accuracy.

Figure 17 shows the experiment setup using a time-lapse image for each trial. The images indicate the paths taken and also the ground-truth IAP sequences in order of contact.

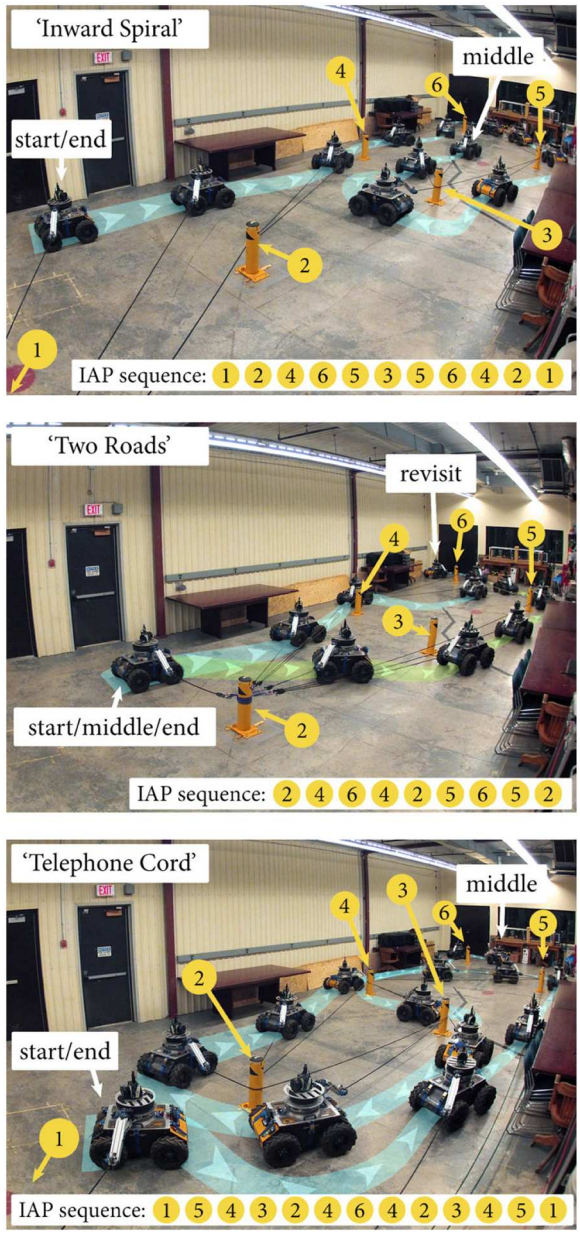


**Fig. 17.** *Indoor-experiment setup.* A time-lapse image for each experiment shows the robot's trajectory marked by a color overlay with outgoing direction arrows. All intermediate anchor points (IAPs) are indicated by numbered yellow markers. Each dataset features a different trajectory and IAP sequence.

## 4.4. Indoor results

What follows is an evaluation of three datasets collected using TReX, each featuring a different trajectory and IAP sequence. Results are presented in the same manner as in Section 4.1, with supplementary results available in Appendix A.1. A set of position error plots in Figure 18 shows the position error of each TSLAM method and dead-reckoned odometry, as compared with ground truth. The errors are further illustrated by the collection of trajectory maps in Figure 19, which show the trajectory from our

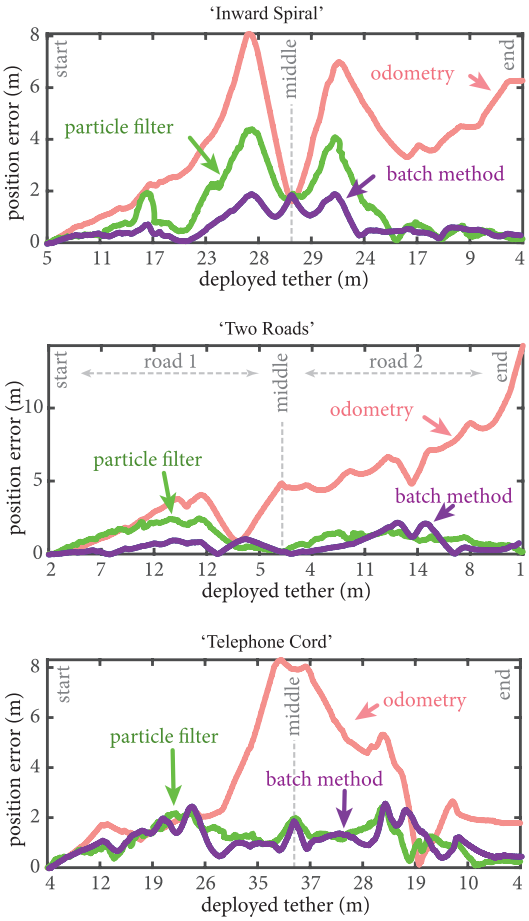


**Fig. 18.** *Root mean square position error.* For each experiment, errors are compared with odometry as a function of deployed tether length. Both approaches to the TSLAM problem outperform odometry by incorporating tether measurements. Best viewed in color.

TSLAM estimates, odometry, and ground truth for each experiment. In general, we are able to outperform odometry, as in simulation. Although errors in wheel odometry are made worse, owing to driving with a taut tether, causing the robot to slide and violate its kinematic motion model (sliding motion is not observable), TSLAM is able to leverage tether measurements to repair the global trajectory. In all experiments, we were able to completely roll off accumulated error from odometry while returning to and observing the initial anchor. The batch estimate best approximates ground truth, owing to superior IAP detection, data association, and outlier rejection, as well as the ability to optimize the entire trajectory and all IAPs simultaneously.

## 4.5. Indoor mapping

In this section, we discuss indoor mapping results, evaluate batch mapping accuracy, and present a computational cost analysis. Figure 20 provides a qualitative example of IAP mapping accuracy for each indoor experiment, which distinguishes the performance difference between the batch and particle-filter methods. The particle filter performs distinctly worse than the batch method when mapping IAPs. The reason for the particle filter's deficient mapping ability can be attributed to (i) the use of a measurement model that is degenerate in noisy conditions (i.e., the model is not suited to handling outliers) and (ii) sample impoverishment, leading to poorer outlier rejection (i.e., we cannot recover from early mapping mistakes because many particles have the same parent particle, owing to iterated resampling). Furthermore, if the detection of an IAP happens too late in the particle filter, the robot position can be pushed further away (i.e., the free length measurement must be satisfied). If the detection happens too early, false positive IAPs can be created, which requires the use of many more particles in order to reject outliers. Not restricted to past measurements, the batch method uses *all* tether measurements to optimize the trajectory or map and uses RANSAC to detect new IAPs and limit the influence of outliers. The key point is that the batch approach is better at capturing the true posterior. The exception to this point is seen in the 'Telephone cord' dataset, which shows a greater variation in performance. For one, the odometry error approaches zero at the second 19 m tether mark. Unlike the other experiments, this is an example of dead-reckoned luck, as odometry shows greater error in the middle of the trajectory. With respect to the batch and particle-filter results, we expect that excessive turning caused a poorer initialization of batch anchor estimates, leading to a similar performance when compared with the particle filter. What follows is a discussion of specific mapping results from individual indoor experiments.

*4.5.1. Inward spiral.* Since IAP 2 is passed at a shallow angle, the change in bearing-to-anchor measurements during observation was small. This stems from IAP 2 being in near alignment with the initial anchor and IAP 4. *Particle-filter method.* This map is a clear example of the particle filter's inability to recover from early mapping mistakes (after a poor initialization of IAP 4, there are many false detections). The estimated map is fortunate to realign with ground-truth IAP 3. *Batch method.* Although the impact is small, IAP 2 was not detected during segmentation, as a result of the near collinearity of IAPs 1, 2, and 4.

*4.5.2. Two roads.* The trajectory is divided into two discrete routes, leading to and around IAP 6. The robot returned to the initial IAP before taking a different route back to IAP 6. *Particle-filter method.* Near the start of each route (heading toward IAP 6) clusters of false detections were made, which cause the map to be skewed toward the initial anchor. *Batch method.* We did not merge multiple instances of IAP 6, since the first observation was made on the 'first road' and associations are only possible between IAPs on the 'active' list. The map indicates a falsely detected IAP in a position where similar false detections occurred in all three maps using the batch method.

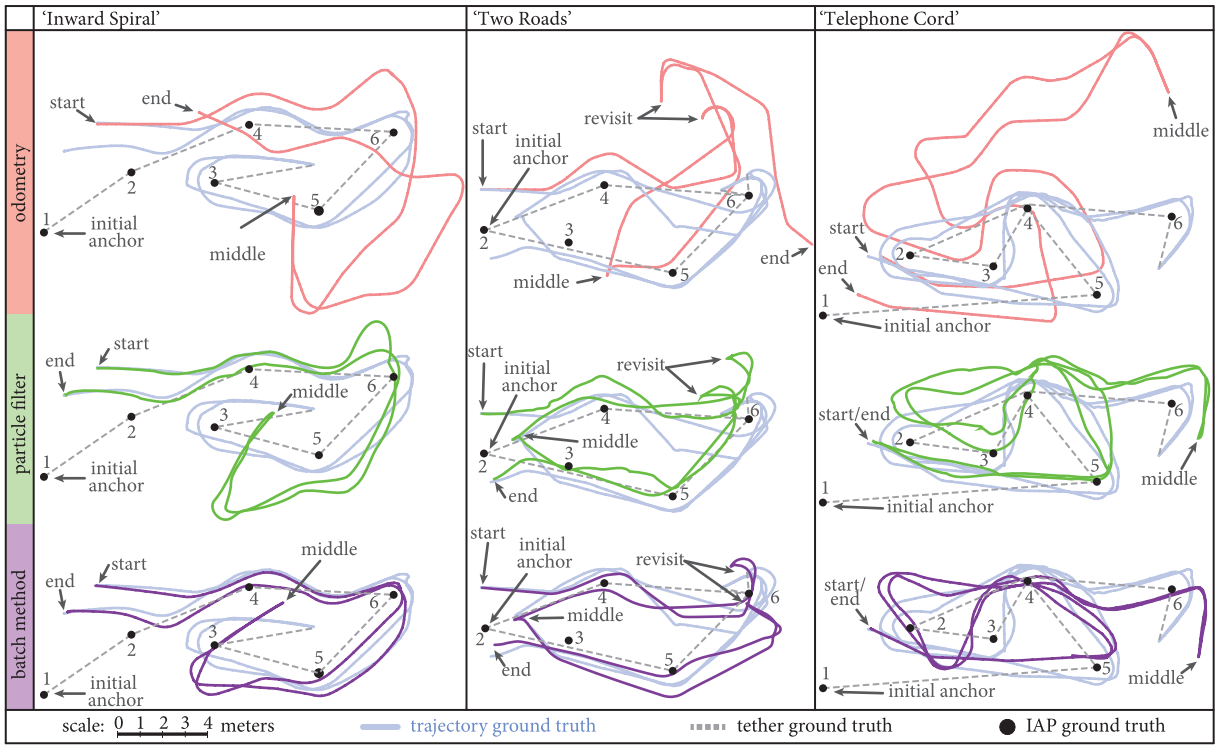


**Fig. 19.** *Trajectory maps.* Localization results from the indoor experiment show that both tethered simultaneous localization and mapping (TSLAM) methods outperform odometry. Each method is able to reduce accumulated error from dead-reckoned wheel odometry on revisiting the initial anchor point, owing to our use of tether measurements. In general, the batch method's estimated trajectory better matches ground truth than does the particle filter.
IAP: intermediate anchor point.

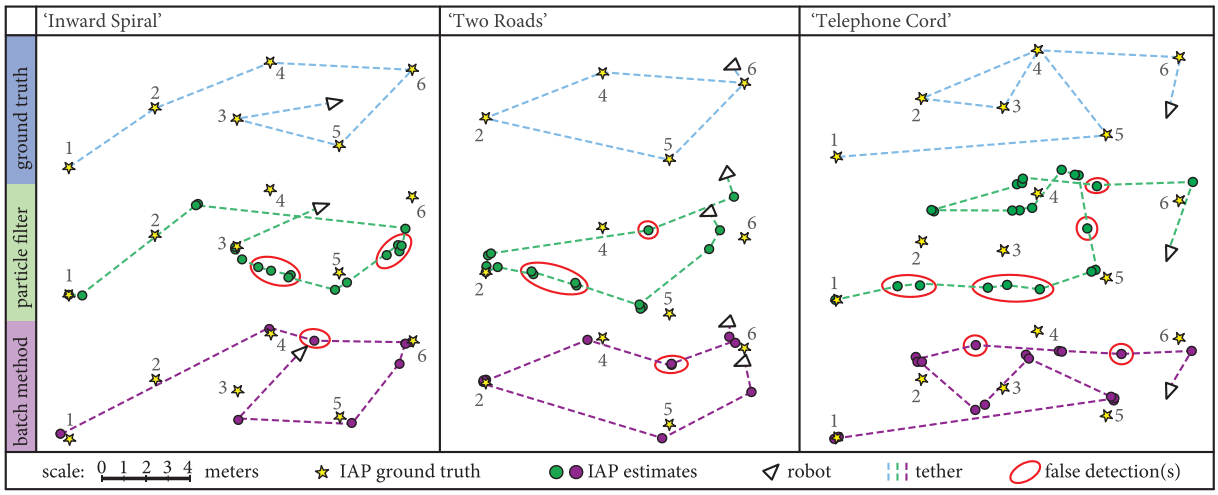


**Fig. 20.** *Mapping comparison.* For each dataset we compare the intermediate anchor point (IAP) mapping accuracy between the particle-filter and batch-TSLAM methods with respect to ground truth. The map generated by batch TSLAM (using only nonvisual sensors) is more accurate than the particle-filter map, owing to a more robust measurement model and added outlier rejection. The batch-method map better matches ground truth, with significantly fewer false detections than the particle filter. False detections were manually determined by visual comparison with the ground-truth map. The variance in mapping performance between the batch and particle-filter methods is greatest for the 'Telephone cord' experiment.

The false detection was probably caused when the robot turned in place (i.e., wheel odometry was poor, owing to slippage, and tether bearing-to-anchor measurements were biased, owing to a drop in tension).

**Table 1.** Batch-method mapping accuracy.

| Dataset | $\mu$ (m) | $\sigma$ (m) | True | Estimated | False |
|---|---|---|---|---|---|
| Inward spiral | 0.63 | 0.45 | 6 | 11 | 1 |
| Two roads | 0.68 | 0.41 | 4 | 11 | 2 |
| Telephone cord | 0.88 | 0.44 | 6 | 19 | 3 |

*4.5.3. Telephone cord.* IAP 4 is revisited a total of four times (twice per outgoing and return trajectories). *Particle-filter method.* There are simply too many false detections in this map (especially early on in the trajectory), which causes mapping accuracy to degrade as a function of tether deployed. *Batch method.* Since the estimate was only able to merge observations of IAPs with similar fixed tether lengths, we end up with two discrete proposals for IAP 4 that cannot be merged. Notwithstanding, the resulting trajectory and map reflect ground truth with reasonable accuracy, given the length of tether deployed.

*4.5.4. Indoor mapping accuracy.* Here we provide batch mapping results only. Table 1 shows mapping errors for each experiment, where the error is determined by first finding the average error for a single IAP with multiple, merged estimates (association is by proximity). We then take the average of individual IAP errors to produce a mean error. For all experiments, the mean, $\mu$, and standard deviation, $\sigma$, are shown in meters. The quantities of true and estimated IAPs are indicated. Any IAPs that were manually determined to be false detections (illustrated with red circles in Figure 20) were not included in the error calculation, as they could not be associated with a ground-truth IAP.

As shown qualitatively in Figure 20 and quantitatively in Table 1, our cumulative mapping error over multiple trials is in the range of 1 m, which, in consideration of the non-visual sensors used and the inherent noise involved, should be sufficient for safe navigation.

*4.5.5. Computational cost.* The most challenging indoor traverse, 'Telephone cord', was recorded over the course of 15 min (900 s). *Particle-filter method.* Each iteration of the particle filter running on the dataset took approximately 0.16 s/iteration. The total run time of the particle-filter algorithm was 24 min (1440 s), which is slower than real time, but it is expected that it could run online if the implementation were moved from *Matlab* to a compiled language. *Batch method.* The primary sparsity pattern for the **A** matrix, which is used in the 'Telephone cord' optimization, is illustrated in Figure 21. Even though the measurement contribution is dense, the number of anchors relative to poses is small. Thus, we can still solve the problem efficiently by exploiting the block-tridiagonal nature of pose estimates using the sparse Cholesky decomposition from equation (23). Each iteration of the batch solver required 0.2 s. The total run time of the full batch pipeline (segmentation, association, and solve to convergence) required only 40 s. The batch method is also prototyped in *Matlab*; thus, we expect that solve times can be further reduced. Furthermore, batch TSLAM could be adapted for online use using an incremental approach similar to that of Kaess et al. (2008, 2011).

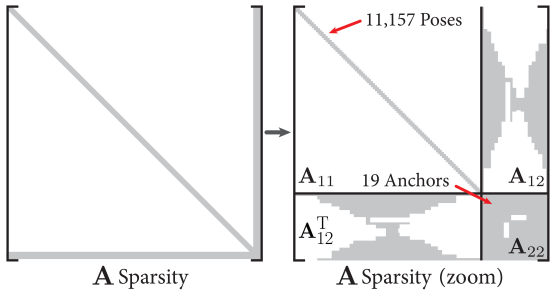


**Fig. 21.** *'Telephone cord' sparsity.* Left: primary sparsity pattern for the **A** matrix used to solve the batch, 'Telephone cord' estimation. Right: for clarity, a nonuniform, zoomed view of the sparsity pattern, which shows that over 11 thousand poses were computed, compared with just 19 intermediate anchor point (IAP) estimates. A total of 19 IAPs were initialized to represent six ground-truth anchors. Even though $\mathbf{A}_{22}$ is mostly dense, we can efficiently solve the problem by exploiting the block-tridiagonal nature of $\mathbf{A}_{11}$. We note that it may be possible to further exploit the hourglass pattern of $\mathbf{A}_{12}$ and $\mathbf{A}_{12}^{\mathrm{T}}$, but that is beyond the scope of this work.

## 4.6. Outdoor experiment

While the indoor test is useful for evaluating the performance of TSLAM with respect to ground truth, it is less relevant to the desired operational environment for TReX (i.e., steep and outdoors). Therefore, we collected data as TReX was piloted in a sloped forest environment through a set of trees on terrain covered with a mixture of soft vegetation and hard rock. The outdoor test serves as an example of TSLAM operating in a challenging field environment. For this experiment, we were not able to collect ground truth for error comparison. Instead, we compare an aerial view of the trajectory from odometry with the batch and particle-filter methods and provide examples of adding IAPs in Figure 22. As shown, both estimation approaches align the outgoing and return trajectories as they roll off the accumulated drift from wheel odometry. In lieu of ground truth, we can use path alignment as a qualitative metric for accuracy outdoors because TReX was piloted along similar outgoing and incoming trajectories in order to prevent tether entanglement. Wheel odometry exhibits characteristic drift (as it did indoors), causing the incoming trajectory to be misaligned with the outgoing trajectory. In terms of mapping, the batch method has initialized IAPs corresponding to each of the images in Figure 22. The particle filter produced poorer mapping results than batch (as it did indoors), so we *do not* show those results here, for clarity. Since the inclination ($\approx$20°) was generally constant during the test, our 2D

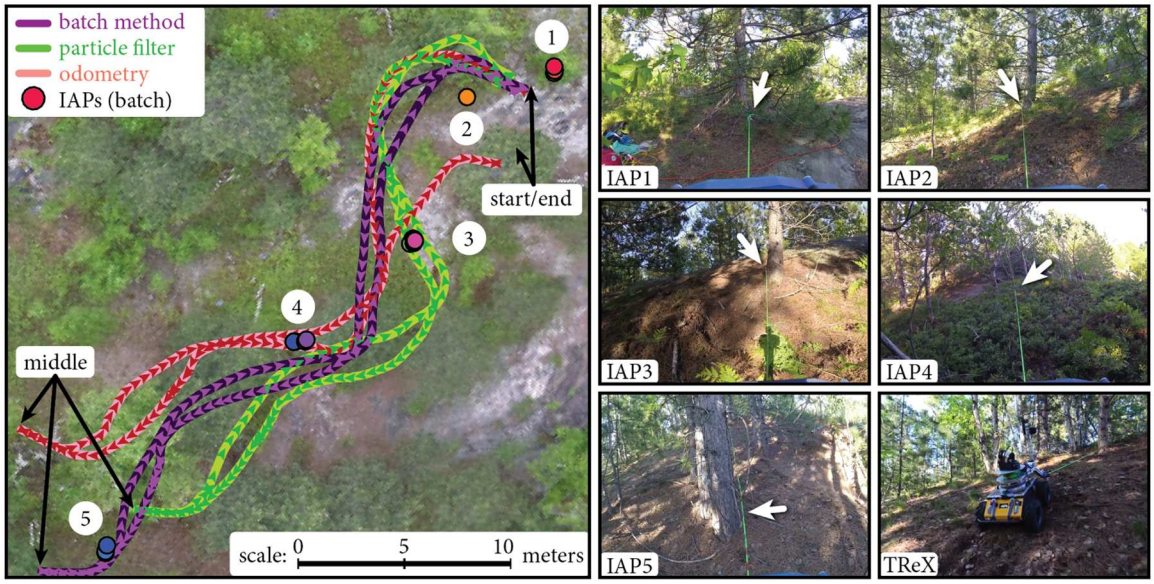


**Fig. 22.** *TSLAM outdoors.* TReX was driven though an inclined forest with trees serving as intermediate anchor points (IAPs). The trajectory map (*left*), which is overlaid on an aerial image of the test site, shows how both TSLAM methods align the outgoing and return trajectories (see direction arrows). The IAP map was made using the batch method in lieu of having ground truth. The IAP sequence (1,2,3,4,5) is numbered according to observation during descent, with 1 being the initial anchor. The images (*right*) were taken in the direction of the bearing-to-anchor (attached to the tether arm) and show the IAPs added along the traverse. The electromechanical tether is clearly visible (light-green line) in all images. The outdoor example shows that TSLAM can work in an operationally relevant environment.

TSLAM formulation provides a reasonable approximation of the robot pose on the slope (assumed to be a plane).

## 5. Experiment discussion

Given our experience with the particle-filter and batch-TSLAM algorithms running on data collected from TReX, we have determined a set of common limitations and outstanding problems.

*Assumptions.* As formulated, we assume that the tether is perfectly taut and that anchors can be represented as zero-radius points. The tautness assumption ignores the fact that a stretched cable or tether will always exhibit some catenary curve, regardless of the tension, which contributes to the length measurement. Dealing with this problem would be difficult because the curve would need to be modeled and the tension between each anchor would need to be estimated. The anchor-point assumption simplifies the representation of large obstacles to a series of points. However, if the tether fully wraps around a small obstacle, that extra length is not accounted for in our formulation.

*Length drift.* The current approach to length measurement on TReX uses an optical, rotatory encoder. This sensor, like any nonabsolute encoder, is prone to drift over time, owing to pulley or tether slippage, causing length error to increase with the amount of tether deployed. Drifting tether length impacts the accuracy of IAP initialization (by the particle filter) and association (in batch loop closure).

*Bearing-to-anchor bias.* The rotational freedom of TReX is reliant on rotary bearings that allow the tether arm to swing and align in the direction of tension. The bearing requires force in the form of a taut tether to overcome stiction (static friction) and rotate. Consequently, the tether arm exhibits bias, which causes outlier measurements. Accounting for bias is nontrivial because it is a function of tension, IAP wrap direction, slippage due to decreased wheel traction, and inclination of the driving terrain (stiction is easier to overcome in steep environments, owing to increased tension influenced by the force due to gravity). The batch method handles the bias better than the particle filter, owing to the RANSAC IAP detection step, but in the worst-case scenario, when the surface is smooth and flat (e.g., our indoor testing environment) and the tension is very low, bias can cause false positive IAPs to be initialized.

*Small-scale loop closure.* It is possible that closely spaced anchors with similar fixed lengths will be pulled on top of one another once associated. This action can be problematic when a series of IAPs representing a larger obstacle (e.g., a large tree trunk) are pulled together, because we lose useful information about the physical shape of obstacles in the environment.

*Large-scale loop closure.* Large-scale loop closure is not addressed in either TSLAM approach. If an IAP were added, removed, and later revisited, we do not make an association, given the nonvisual nature of the problem (i.e., nonvisual place recognition is not possible). In an experiment such as ‘Two roads’ (Figure 19), where two observations

were made of the same IAP at different times and from different approaches, large-scale loop closure would greatly increase our mapping accuracy. It may be possible to rely further on the switchable constraint approach to simply add all possible loop closures, but we have not tried this. Alternatively, since the IAP map resembles a constellation, it may be possible to use constellation matching to associate a new detection of an IAP with those that have been observed before.

*Dynamic anchors.* We make a strong assumption that our IAPs are fixed in the environment. Given our limited set of available measurements in TSLAM, it would be difficult, but not impossible, to discover whether an IAP changes position or is removed entirely. In practice, we observe that IAPs can be removed, owing to tension alone (e.g., a tree trunk breaks under force and the tether detaches), so this will need to be addressed in future work.

Notwithstanding these challenges, we have shown that the TSLAM problem is not only solvable with basic, nonvisual measurements (wheel odometry, tether length, and bearing-to-anchor) but, through experiment, have demonstrated that we are capable of producing accurate localization and mapping results for a real tethered robot with noisy sensors in cluttered environments.

## 6. Conclusions and future work

This paper has formulated the TSLAM problem and proposed two methods that use wheel odometry and nonvisual tether measurements to jointly estimate the state of a robot and any IAPs resulting from the tether coming into contact with obstacles. TSLAM is similar to range-bearing SLAM, with a major difference being that each measurement is a function of one robot pose and potentially all active IAPs. We expand on an online particle-filter approach to TSLAM (McGarey et al., 2016), which uses particles to represent the robot's trajectory and individual Gaussians for the map of IAPs. With lessons learned from this approach, we demonstrate an efficient batch method capable of solving the proposed estimation problem. We motivate each method by outlining the estimation pipeline, how new IAPs are detected, and whether or not a new observation corresponds to a previously observed IAP (i.e., loop closure). Experiments are performed in simulation and in tests with TReX. Either approach to TSLAM outperforms odometry and provides a nonvisual means to determine the position and quantity of IAPs in cluttered environments. We show that the batch approach is more accurate than the particle filter, owing to superior IAP detection, data association, and outlier rejection.

Future versions of the TSLAM pipeline will consider a sliding-window batch variation. An incremental approach will allow real-time updates to our batch solution as soon as new measurements arrive (Kaess et al., 2008, 2011). Using our newly developed, visual-route following pipeline for tethered robots, we can now drive trajectories on steep terrain and automatically repeat them (McGarey et al., 2017). This ability enables the robot to autonomously reobserve all added IAPs, which is important for loop closure in TSLAM. Therefore, future experiments can be performed in a more operationally relevant environment (e.g., steeper, with varying slopes), which will require adapting planar TSLAM to three dimensions. We will investigate the use of inertial measurements of the robot's pose with respect to gravity to constrain the robot's heading on steep terrain to aid in 3D pose estimation. Methods of handling IAP change detection will also be investigated. Ultimately, nonvisual TSLAM is an interesting problem, which, when integrated with visual SLAM, will aid in robot localization and anchor mapping in environments that are dark and dusty.

### Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was supported by Fulbright Canada, the Natural Sciences and Engineering Research Council of Canada (NSERC), the NSERC Canadian Field Robotics Network, and the Collaborative Research and Training Experience Program.

### Notes

1. Our batch-TSLAM algorithm could be adapted for online use, but that is beyond the scope of this article.

2. A real-life example of the **A**-matrix sparsity is provided in Figure 21.

3. In a conventional SLAM formulation, the complexity would be $O(K^3 + K^2N)$, where $N$ would represent the maximum landmarks observed.

### References

Agarwal P, Tipaldi GD, Spinello L, et al. (2013) Robust map optimization using dynamic covariance scaling. In: *2013 IEEE international conference on robotics and automation (ICRA)*, Karlsruhe, Germany, 6–10 May 2013, pp. 62–69. Piscataway: IEEE.

Bailey T and Durrant-Whyte H (2006) Simultaneous localization and mapping (SLAM): Part II. *IEEE Robotics & Automation Magazine* 13(3): 108–117.

Bentley JL (1975) Multidimensional binary search trees used for associative searching. *Communications of the ACM* 18(9): 509–517.

Brown DC (1958) A solution to the general problem of multiple station analytical stereotriangulation. RCA-MTP Data Reduction Technical Report No. 43 (or AFMTC TR 58-8). Patrick Airforce Base, Florida, USA.

Chum O, Matas J, and Kittler J (2003) Locally optimized RANSAC. In: Michaelis B and Krell G (eds.) *Pattern Recognition* (*Lecture Notes in Computer Science*, vol. 2781). Berlin: Springer, pp. 236–243.

Corominas Murtra A and Tur JMM (2013) IMU and cable encoder data fusion for in-pipe mobile robot localization. In: *2013*

*IEEE international conference on technologies for practical robot applications (TePRA)*, Woburn, MA, 22–23 April 2013. Piscataway, NJ: IEEE.

Durrant-Whyte H and Bailey T (2006) Simultaneous localization and mapping: Part I. *IEEE Robotics & Automation Magazine* 13(2): 99–110.

Fischler MA and Bolles RC (1981) Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM* 24(6): 381–395.

Howard A (2008) Real-time stereo visual odometry for autonomous ground vehicles. In: *2008 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, Nice, France, 22–26 September 2008, pp. 3946–3952. Piscataway, NJ: IEEE.

Kaess M, Johannsson H, Roberts R, et al. (2011) iSAM2: Incremental smoothing and mapping using the Bayes tree. *The International Journal of Robotics Research* 31(2): 216–235.

Kaess M, Ranganathan A, and Dellaert F (2008) iSAM: Incremental smoothing and mapping. *IEEE Transactions on Robotics* 24(6): 1365–1378.

Klingensmith M, Sirinivasa SS, and Kaess M (2016) Articulated robot motion for simultaneous localization and mapping (ARM-SLAM). *IEEE Robotics and Automation Letters* 1(2): 1156–1163.

Kumar TR and Richardson RC (2008) Tether monitoring techniques for environment monitoring, tether following and localization of autonomous mobile robots. In: *2008 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, Nice, France, 22–26 September 2008, pp. 2109–2114. Piscataway, NJ: IEEE.

Lu F and Milios E (1997) Globally consistent range scan alignment for environment mapping. *Autonomous Robots* 4(4): 333–349.

McGarey P, MacTavish K, Pomerleau F, et al. (2016) The line leading the blind: Towards nonvisual localization and mapping for tethered mobile robots. In: *2016 IEEE international conference on robotics and automation (ICRA)*, Stockholm, Sweden, 16–21 May 2016, pp. 4799–4806. Piscataway, NJ: IEEE.

McGarey P, Polzin M, and Barfoot TD (2017) Falling in line: Visual route following on extreme terrain for a tethered mobile robot. In: *2017 IEEE international conference on robotics and automation (ICRA)*, Singapore, 29 May–3 June 2017, pp. 2017–2034. Piscataway, NJ: IEEE.

McGarey P, Pomerleau F, and Barfoot TD (2015) System design of a tethered robotic explorer (TReX) for 3D mapping of steep terrain and harsh environments. In: Wettergreen D and Barfoot T (eds.) *Field and Service Robotics*. (*Springer Tracts in Advanced Robotics*, vol. 113). Cham: Springer, pp. 267–281.

MacTavish K and Barfoot TD (2015) At all costs: A comparison of robust cost functions for camera correspondence outliers. In: *2015 12th conference on computer and robot vision (CRV)*, Halifax, NS, 3–5 June 2015, pp. 62–69. Piscataway, NJ: IEEE.

Matthews JB and Nesnas IA (2012) On the design of the Axel and DuAxel rovers for extreme terrain exploration. In: *2012 IEEE aerospace conference*, Big Sky, MT, 3–10 March 2012. Piscataway, NJ: IEEE.

Montemerlo M, Thrun S, Koller D, et al. (2002) FastSLAM: A factored solution to the simultaneous localization and mapping problem. In: *18th national conference on artificial intelligence*, Edmonton, Alberta, Canada, 28 July–1 August 2002, pp. 593–598. Menlo Park, CA: American Association for Artificial Intelligence.

Rajan VAKT, Nagendran A, Dehghani-Sanij A, et al. (2014) Tether monitoring for entanglement detection, disentanglement and localisation of autonomous robots. *Robotica* 34(3): 527–548.

Scaramuzza D, Fraundorfer F, and Siegwart R (2009) Real-time monocular visual odometry for on-road vehicles with 1-point RANSAC. In: *2009 IEEE international conference on robotics and automation (ICRA)*, Kobe, Japan, 12–17 May 2009, pp. 4293–4299. Piscataway, NJ: IEEE.

Sinden F (1990) The tethered robot problem. *The International Journal of Robotics Research* 9(1): 122–133.

Smith R, Self M, and Cheeseman P (1990) Estimating uncertain spatial relationships in robotics. In: Cox IJ and Wilfong GT (eds.) *Autonomous Robot Vehicles*. New York: Springer, pp. 167–193.

Sünderhauf N and Protzel P (2012) Switchable constraints for robust pose graph SLAM. In: *2012 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, Vilamoura, Portugal, 7–12 October 2012, pp. 1879–1884. Piscataway, NJ: IEEE.

Teshnizi RH and Shell D (2014) Computing cell-based decompositions dynamically for planning motions of tethered robots. In: *2014 IEEE international conference on robotics and automation (ICRA)*, Hong Kong, 31 May–7 June 2014, pp. 6130–6135. Piscataway, NJ: IEEE.

Thrun S and Montemerlo M (2006) The graph SLAM algorithm with applications to large-scale mapping of urban structures. *The International Journal of Robotics Research* 25(5–6): 403–429.

Thrun S, Fox D, Burgard W, et al. (2001) Robust Monte Carlo localization for mobile robots. *Artificial Intelligence* 128(1): 99–141.

Thrun S, Montemerlo M, Koller D, et al. (2004) FastSLAM: An efficient solution to the simultaneous localization and mapping problem with unknown data association. *Journal of Machine Learning Research* 4(3): 380–407.

Triggs B, McLauchlan PF, Hartley RI, et al. (2000) Bundle adjustment—a modern synthesis. In: Triggs B, Zisserman A, and Szeliski R (eds) *Vision Algorithms: Theory and Practice. IWVA 1999*. (*Lecture Notes in Computer Science*, vol. 1883). Berlin: Springer, pp. 298–372.

Wettergreen D, Thorpe C, and Whittaker R (1993) Exploring Mount Erebus by walking robot. *Robotics and Autonomous Systems* 11(3): 171–185.

# A. Appendix

## A.1. Supplemental results

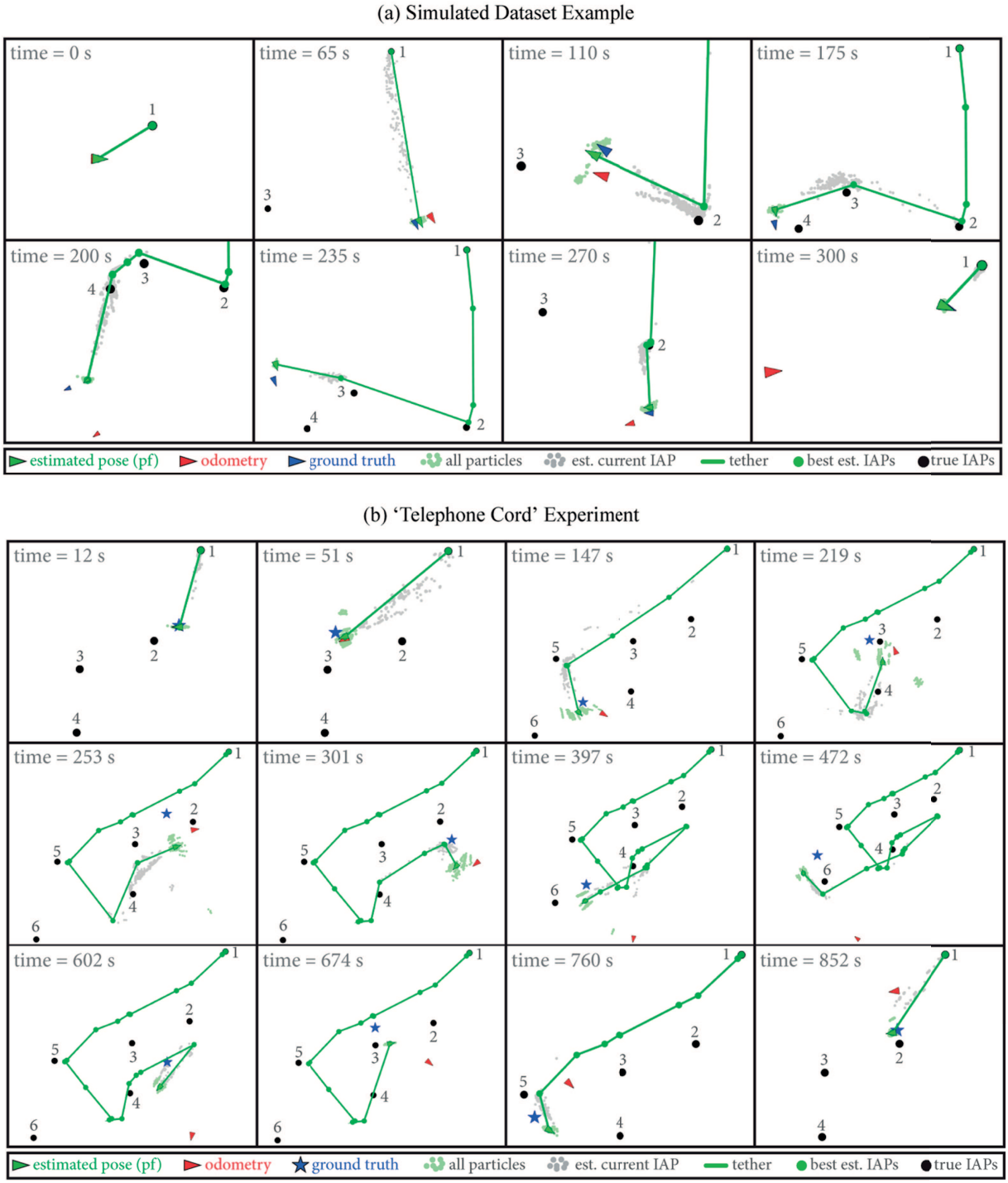


**Fig. 23.** *Particle-filter results.* A set of time-lapse sequences show the particle filter running on the (a) simulated trial and (b) 'Telephone cord' (b) datasets. Each particle, represented as small light-green circles, stores its own discrete map of IAPs. Newly detected IAPs are shown by gray circles (each particle estimates its belief of the current IAP). The estimated robot pose (green triangle) comes from the mean of the highest weighted particles at any given time. The most probable map is shown (from the top weighted particle), where large green circles are the encountered IAPs, and the line connecting them represents the tether configuration. Best viewed in color.
est.: estimated; IAP: intermediate anchor point; pf: particle filter.

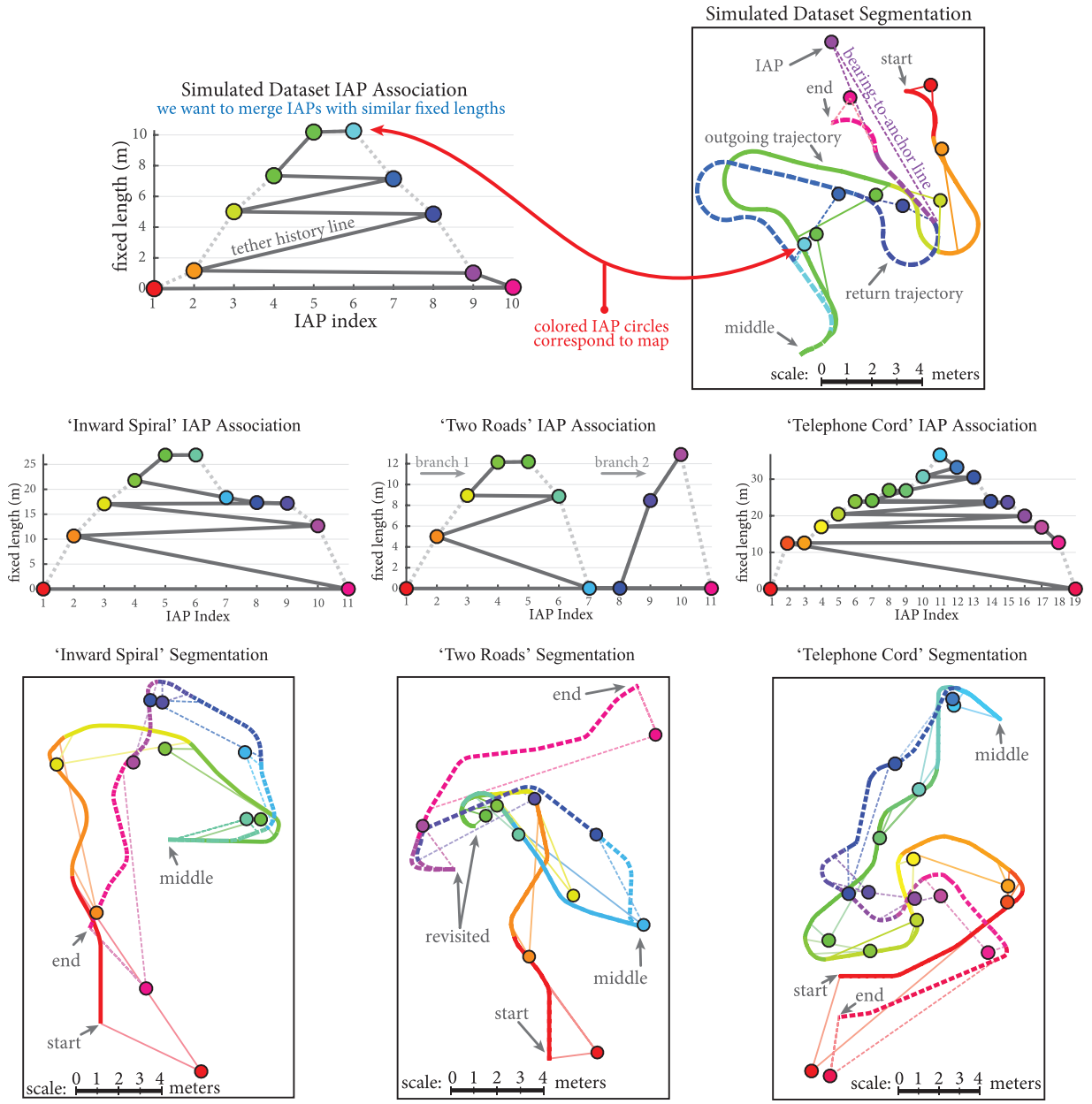


**Fig. 24.** *Batch results.* The process for data association and trajectory segmentation is illustrated for the batch method on both simulated and indoor-experiment datasets. *Intermediate anchor point (IAP) association.* The fixed-length tree is used to determine tether history by associating IAPs with similar fixed tether lengths that should be merged in the map (Section 3.2.3 provides details). *Segmentation.* The trajectory from odometry is divided into subtrajectories, each associated with a single IAP of matching color using the RANSAC method described in Section 3.2.2. Solid and dashed lines represent outgoing and return trajectories. Dashed, colorized bearing-to-anchor lines are shown. Best viewed in color.

## *A.2. Supplemental math*

### General TSLAM

*Robot trajectory.*

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_1 \\ \vdots \\ \mathbf{x}_K \end{bmatrix}, \quad \mathbf{x}_k = \begin{bmatrix} x_k \\ y_k \\ \theta_k \end{bmatrix} \tag{26}$$

*Body-centric velocities.*

$$\mathbf{v} = \begin{bmatrix} \mathbf{v}_1 \\ \vdots \\ \mathbf{v}_K \end{bmatrix}, \quad \mathbf{v}_k = \begin{bmatrix} \upsilon_{x,k} \\ \upsilon_{y,k} \\ \omega_k \end{bmatrix} \tag{27}$$

*Motion model.*

$$\mathbf{x}_k = \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{v}_k, \mathbf{w}_k), \quad \mathbf{w}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_0), \quad \mathbf{Q}_0 = \begin{bmatrix} \sigma_{\upsilon_x}^2 & & \\ & \sigma_{\upsilon_y}^2 & \\ & & \sigma_{\omega}^2 \end{bmatrix} \tag{28}$$

$$\begin{bmatrix} x_k \\ y_k \\ \theta_k \end{bmatrix} = \begin{bmatrix} x_{k-1} \\ y_{k-1} \\ \theta_{k-1} \end{bmatrix} + T \begin{bmatrix} \cos\theta_{k-1} & -\sin\theta_{k-1} & 0 \\ \sin\theta_{k-1} & \cos\theta_{k-1} & 0 \\ 0 & 0 & 1 \end{bmatrix} \left( \begin{bmatrix} \upsilon_{x,k} \\ 0 \\ \omega_k \end{bmatrix} + \begin{bmatrix} w_{\upsilon_{x,k}} \\ w_{\upsilon_{y,k}} \\ w_{\omega_k} \end{bmatrix} \right), \quad T \text{ is a sample rate (e.g., 10 Hz)} \tag{29}$$

*Tether measurements.*

$$\mathbf{y} = \begin{bmatrix} \mathbf{y}_1 \\ \vdots \\ \mathbf{y}_K \end{bmatrix}, \quad \mathbf{y}_k = \begin{bmatrix} d_k \\ \phi_k \end{bmatrix} \tag{30}$$

*IAP list (two different representations are used).*

$$\boldsymbol{\ell} = \begin{bmatrix} \boldsymbol{\ell}_1 \\ \vdots \\ \boldsymbol{\ell}_N \end{bmatrix}, \quad \boldsymbol{\ell}_n \to \underbrace{\mathcal{N}\left( \begin{bmatrix} x_n \\ y_n \end{bmatrix}, \begin{bmatrix} \sigma_d^2 & \\ & \sigma_\phi^2 \end{bmatrix} \right)}_{\text{particle filter (see next section)}}, \quad \boldsymbol{\ell}_n = \underbrace{\begin{bmatrix} x_n \\ y_n \\ d_{\text{fix},\boldsymbol{\ell}_n} \end{bmatrix}}_{\text{batch method}}, \quad d_{\text{fix},\boldsymbol{\ell}_n} \text{ is the fixed tether length for } \boldsymbol{\ell}_n \tag{31}$$

*Measurement model.*

$$\mathbf{y}_k = \mathbf{g}(\mathbf{x}_k, \boldsymbol{\ell}_n) + \mathbf{n}_k, \quad \mathbf{n}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_0), \quad \mathbf{R}_0 = \begin{bmatrix} \sigma_d^2 & \\ & \sigma_\phi^2 \end{bmatrix} \tag{32}$$

$$\begin{bmatrix} d_k \\ \phi_k \end{bmatrix} = \begin{bmatrix} \|\boldsymbol{\ell}_1 - \boldsymbol{\ell}_2\| + \ldots + \|\boldsymbol{\ell}_{n-1} - \boldsymbol{\ell}_n\| + \left\| \boldsymbol{\ell}_n - \begin{bmatrix} x_k \\ y_k \end{bmatrix} \right\| \\ \text{atan2}(\ell_{y,n} - y_k, \ell_{x,n} - x_k) - \theta_k \end{bmatrix} + \begin{bmatrix} n_{d,k} \\ n_{\phi,k} \end{bmatrix} \tag{33}$$

*Pose errors.*

$$\mathbf{e}_{\mathbf{x},k}(\mathbf{x}) = \begin{cases} \mathbf{0} & k = 0 \\ \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{v}_k, \mathbf{0}) - \mathbf{x}_k & k = 1, \ldots, K \end{cases} \tag{34}$$

*Measurement errors.*

$$\mathbf{e}_{\mathbf{y},k}(\mathbf{x}, \boldsymbol{\ell}) = \mathbf{y}_k - \mathbf{g}(\mathbf{x}_k, \boldsymbol{\ell}), \quad k = 1, \ldots, K \tag{35}$$

## Particle-filter TSLAM

*Inverse measurement model (initializing the mean, $\boldsymbol{\mu}_{n+1}$, of Gaussian, $\boldsymbol{\ell}_{n+1}$).*

$$\mathbf{g}^{-1}(\mathbf{x}_k, \mathbf{y}_k, \boldsymbol{\ell}_n) = \boldsymbol{\mu}_{n+1} = \begin{bmatrix} x_k \\ y_k \end{bmatrix} - d_{\text{free},k+1} \begin{bmatrix} \cos(\phi_k + \theta_k) \\ \sin(\phi_k + \theta_k) \end{bmatrix} \tag{36}$$

*Free length to a newly detected IAP.*

$$d_{\text{free},k+1} = \frac{\left\| \begin{bmatrix} x_k \\ y_k \end{bmatrix} - \boldsymbol{\mu}_n \right\|^2 - (d_{\text{free},k})^2}{2(d_{\text{free},k}) - \left\| \begin{bmatrix} x_k \\ y_k \end{bmatrix} - \boldsymbol{\mu}_n \right\| \cos\psi_k}, \quad \text{where } \psi_k \text{ is the bearing-to-anchor measurement error} \tag{37}$$

*Linearized inverse measurement model with respect to tether measurements.*

$$\mathbf{G}_k^{-1} = \frac{\partial \mathbf{g}^{-1}(\mathbf{x}_k, \mathbf{y}_k, \boldsymbol{\ell}_n)}{\partial \mathbf{y}_k} = \begin{bmatrix} \frac{\partial \mathbf{g}_1^{-1}(\mathbf{x}_k, \mathbf{y}_k, \boldsymbol{\ell}_n)}{\partial d_k} & \frac{\partial \mathbf{g}_1^{-1}(\mathbf{x}_k, \mathbf{y}_k, \boldsymbol{\ell}_n)}{\partial \phi_k} \\ \frac{\partial \mathbf{g}_2^{-1}(\mathbf{x}_k, \mathbf{y}_k, \boldsymbol{\ell}_n)}{\partial d_k} & \frac{\partial \mathbf{g}_2^{-1}(\mathbf{x}_k, \mathbf{y}_k, \boldsymbol{\ell}_n)}{\partial \phi_k} \end{bmatrix}$$

$$\begin{aligned}
\frac{\partial \mathbf{g}_1^{-1}(\mathbf{x}_k, \mathbf{y}_k, \boldsymbol{\ell}_n)}{\partial d_k} &= \frac{2\left(\left\|\begin{bmatrix} x_k \\ y_k \end{bmatrix} - \boldsymbol{\mu}_n\right\|^2 - d_{\text{free},k+1}^2\right)\cos(\phi_k + \theta_k)}{\left(2d_{\text{free},k+1} - 2\left\|\begin{bmatrix} x_k \\ y_k \end{bmatrix} - \boldsymbol{\mu}_n\right\|\cos(\psi_k)\right)^2} + \frac{2d_{\text{free},k+1}\cos(\phi_k + \theta_k)}{2d_{\text{free},k+1} - 2\left\|\begin{bmatrix} x_k \\ y_k \end{bmatrix} - \boldsymbol{\mu}_n\right\|\cos(\psi_k)} \\
\frac{\partial \mathbf{g}_2^{-1}(\mathbf{x}_k, \mathbf{y}_k, \boldsymbol{\ell}_n)}{\partial d_k} &= \frac{2\left(\left\|\begin{bmatrix} x_k \\ y_k \end{bmatrix} - \boldsymbol{\mu}_n\right\|^2 - d_{\text{free},k+1}^2\right)\sin(\phi_k + \theta_k)}{\left(2d_{\text{free},k+1} - 2\left\|\begin{bmatrix} x_k \\ y_k \end{bmatrix} - \boldsymbol{\mu}_n\right\|\cos(\psi_k)\right)^2} + \frac{2d_{\text{free},k+1}\sin(\phi_k + \theta_k)}{2d_{\text{free},k+1} - 2\left\|\begin{bmatrix} x_k \\ y_k \end{bmatrix} - \boldsymbol{\mu}_n\right\|\cos(\psi_k)} \\
\frac{\partial \mathbf{g}_1^{-1}(\mathbf{x}_k, \mathbf{y}_k, \boldsymbol{\ell}_n)}{\partial \phi_k} &= \frac{\left(\left\|\begin{bmatrix} x_k \\ y_k \end{bmatrix} - \boldsymbol{\mu}_n\right\|^2 - d_{\text{free},k+1}^2\right)\sin(\phi_k + \theta_k)}{2d_{\text{free},k+1} - 2\left\|\begin{bmatrix} x_k \\ y_k \end{bmatrix} - \boldsymbol{\mu}_n\right\|\cos(\psi_k)} - \frac{\left\|\begin{bmatrix} x_k \\ y_k \end{bmatrix} - \boldsymbol{\mu}_n\right\|\left(\left\|\begin{bmatrix} x_k \\ y_k \end{bmatrix} - \boldsymbol{\mu}_n\right\|^2 - d_{\text{free},k+1}^2\right)\cos(\phi_k + \theta_k)\sin(\psi_k)}{\left(2d_{\text{free},k+1} - 2\left\|\begin{bmatrix} x_k \\ y_k \end{bmatrix} - \boldsymbol{\mu}_n\right\|\cos(\psi_k)\right)^2} \\
\frac{\partial \mathbf{g}_2^{-1}(\mathbf{x}_k, \mathbf{y}_k, \boldsymbol{\ell}_n)}{\partial \phi_k} &= -\frac{\left(\left\|\begin{bmatrix} x_k \\ y_k \end{bmatrix} - \boldsymbol{\mu}_n\right\|^2 - d_{\text{free},k+1}^2\right)\cos(\phi_k + \theta_k)}{2d_{\text{free},k+1} - 2\left\|\begin{bmatrix} x_k \\ y_k \end{bmatrix} - \boldsymbol{\mu}_n\right\|\cos(\psi_k)} - \frac{\left\|\begin{bmatrix} x_k \\ y_k \end{bmatrix} - \boldsymbol{\mu}_n\right\|\left(\left\|\begin{bmatrix} x_k \\ y_k \end{bmatrix} - \boldsymbol{\mu}_n\right\|^2 - d_{\text{free},k+1}^2\right)\sin(\phi_k + \theta_k)\sin(\psi_k)}{\left(2d_{\text{free},k+1} - 2\left\|\begin{bmatrix} x_k \\ y_k \end{bmatrix} - \boldsymbol{\mu}_n\right\|\cos(\psi_k)\right)^2}
\end{aligned} \tag{38}$$

*Linearized measurement model with respect to IAPs.*

$$\mathbf{G}_k = \frac{\partial \mathbf{g}(\mathbf{x}_k, \boldsymbol{\ell}_n)}{\partial \boldsymbol{\mu}_n} = \begin{bmatrix} \frac{\partial g_1}{\partial \mu_{x,n}} & \frac{\partial g_1}{\partial \mu_{y,n}} \\ \frac{\partial g_2}{\partial \mu_{x,n}} & \frac{\partial g_2}{\partial \mu_{y,n}} \end{bmatrix} = \begin{bmatrix} \frac{\mu_{x,n} - x_k}{\left\|\begin{bmatrix} x_k \\ y_k \end{bmatrix} - \boldsymbol{\mu}_n\right\|} & \frac{\mu_{y,n} - y_k}{\left\|\begin{bmatrix} x_k \\ y_k \end{bmatrix} - \boldsymbol{\mu}_n\right\|} \\ \frac{-\mu_{y,n} + y_k}{\left\|\begin{bmatrix} x_k \\ y_k \end{bmatrix} - \boldsymbol{\mu}_n\right\|^2} & \frac{\mu_{x,n} - x_k}{\left\|\begin{bmatrix} x_k \\ y_k \end{bmatrix} - \boldsymbol{\mu}_n\right\|^2} \end{bmatrix} \tag{39}$$

*Initializing an IAP.*

(1) $\boldsymbol{\mu}_{n+1} = \mathbf{g}^{-1}(\mathbf{x}_k, \mathbf{y}_k, \boldsymbol{\ell}_n)$ ⇐ initialize the mean of Gaussian $\boldsymbol{\ell}_{n+1}$

(2) $\mathbf{G}_k^{-1} = \frac{\partial \mathbf{g}^{-1}(\mathbf{x}_k, \mathbf{y}_k, \boldsymbol{\ell}_n)}{\partial \mathbf{y}_k}$ ⇐ Jacobian of inverse measurement model

(3) $\boldsymbol{\Sigma}_{n+1} = \mathbf{G}_k^{-1}\mathbf{R}_0(\mathbf{G}_k^{-1})^{\mathrm{T}}$ ⇐ initialize measurement covariance

(4) $w_k = w_0$ ⇐ initialize weight

(5) $\boldsymbol{\ell}_{n+1} \to \mathcal{N}(\boldsymbol{\mu}_{n+1}, \boldsymbol{\Sigma}_{n+1})$ ⇐ Gaussian with IAP position and covariance

*Update an IAP.*

(1) $\hat{\mathbf{y}}_k = \mathbf{g}(\mathbf{x}_k, \boldsymbol{\ell}_n)$ ⇐ estimated measurement

(2) $\mathbf{G}_k = \frac{\partial \mathbf{g}(\mathbf{x}_k, \boldsymbol{\ell}_n)}{\partial \boldsymbol{\mu}_n}$ ⇐ Jacobian of measurement model

(3) $\mathbf{R}_k = \mathbf{G}_k \Sigma_n \mathbf{G}_k^{\mathrm{T}} + \mathbf{R}_0$ ⇐ update measurement uncertainty

(4) $w_k = |2\pi\mathbf{R}_k|^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}(\mathbf{y}_k - \hat{\mathbf{y}}_k)^{\mathrm{T}}\,\mathbf{R}_k^{-1}(\mathbf{y}_k - \hat{\mathbf{y}}_k)\right\}$ ⇐ update weight

(5) $\mathbf{K}_k = \Sigma_n \mathbf{G}_k^{\mathrm{T}} \mathbf{R}_k^{-1}$ ⇐ Kalman gain

(6) $\boldsymbol{\mu}_n = \boldsymbol{\mu}_n + \mathbf{K}_k(\mathbf{y}_k - \hat{\mathbf{y}}_k)$ ⇐ update IAP position

(7) $\Sigma_n = (\mathbf{I} - \mathbf{K}_k\mathbf{G}_k)\,\Sigma_n$ ⇐ update covariance, $\mathbf{I}$ is identity

(8) $\boldsymbol{\ell}_n \to \mathcal{N}(\boldsymbol{\mu}_n, \boldsymbol{\Sigma}_n)$ ⇐ update Gaussian

*Removing an IAP.*
Remove an IAP from the list and then perform an update (as above) on the next IAP in the list.

**Batch TSLAM**

*Linear system to solve.*

$$\underbrace{\begin{bmatrix}\mathbf{A}_{11} & \mathbf{A}_{12}\\ \mathbf{A}_{12}^{\mathrm{T}} & \mathbf{A}_{22}\end{bmatrix}}_{\mathbf{A}}\underbrace{\begin{bmatrix}\delta\mathbf{x}^{\star}\\ \delta\boldsymbol{\ell}^{\star}\end{bmatrix}}_{\mathbf{z}}=\underbrace{\begin{bmatrix}\mathbf{b}_1\\ \mathbf{b}_2\end{bmatrix}}_{\mathbf{b}} \tag{40}$$

*Components of* $\mathbf{A}$ *(Hessian).*

$$\mathbf{A}=\mathbf{H}^{\mathrm{T}}\mathbf{W}^{-1}\mathbf{H}=\begin{bmatrix}\mathbf{F}^{-\mathrm{T}}\mathbf{Q}^{-1}\mathbf{F}^{-}+\mathbf{G}_1^{-\mathrm{T}}\mathbf{R}^{-1}\mathbf{G}_1 & \mathbf{G}_1^{-\mathrm{T}}\mathbf{R}^{-1}\mathbf{G}_2\\ \mathbf{G}_2^{-\mathrm{T}}\mathbf{R}^{-1}\mathbf{G}_1 & \mathbf{G}_2^{-\mathrm{T}}\mathbf{R}^{-1}\mathbf{G}_2\end{bmatrix} \tag{41}$$

*Components of* $\mathbf{b}$ *(Jacobian).*

$$\mathbf{b}=\mathbf{H}^{\mathrm{T}}\mathbf{W}^{-1}\mathbf{e}(\mathbf{x}^{\star},\boldsymbol{\ell}^{\star}),\quad \mathbf{e}(\mathbf{x}^{\star},\boldsymbol{\ell}^{\star})=\begin{bmatrix}\mathbf{e}_{\mathbf{x}}(\mathbf{x}^{\star})\\ \mathbf{e}_{\mathbf{y}}(\mathbf{x}^{\star},\boldsymbol{\ell}^{\star})\end{bmatrix} \tag{42}$$

*Components of* $\mathbf{H}$ *and* $\mathbf{W}^{-1}$.

$$\mathbf{H}=\begin{bmatrix}\mathbf{F}^{-1} & \mathbf{0}\\ \mathbf{G}_1 & \mathbf{G}_2\end{bmatrix},\quad \mathbf{W}^{-1}=\begin{bmatrix}\mathbf{Q}^{-1} & \\ & \mathbf{R}^{-1}\end{bmatrix} \tag{43}$$

*Linearized motion model (inverse).*

$$\mathbf{F}^{-1}=\begin{bmatrix}\mathbf{1} & & & & \\ -\frac{\partial\mathbf{f}(\mathbf{x}_0,\mathbf{v}_1,\mathbf{w}_1)}{\partial\mathbf{x}_0} & \mathbf{1} & & & \\ & -\frac{\partial\mathbf{f}(\mathbf{x}_1,\mathbf{v}_2,\mathbf{w}_2)}{\partial\mathbf{x}_1} & \ddots & & \\ & & \ddots & \mathbf{1} & \\ & & & -\frac{\partial\mathbf{f}(\mathbf{x}_{K-1},\mathbf{v}_K,\mathbf{w}_K)}{\partial\mathbf{x}_{K-1}} & \mathbf{1}\end{bmatrix} \tag{44}$$

*Motion model Jacobian.*

$$\frac{\partial\mathbf{f}(\mathbf{x}_{k-1},\mathbf{v}_k,\mathbf{w}_k)}{\partial\mathbf{x}_{k-1}}\bigg|_{\mathbf{x}_{k-1},\mathbf{v}_k,\mathbf{0}}=\begin{bmatrix}\frac{\partial f_1}{\partial x_{k-1}} & \frac{\partial f_1}{\partial y_{k-1}} & \frac{\partial f_1}{\partial\theta_{k-1}}\\ \frac{\partial f_2}{\partial x_{k-1}} & \frac{\partial f_2}{\partial y_{k-1}} & \frac{\partial f_2}{\partial\theta_{k-1}}\\ \frac{\partial f_3}{\partial x_{k-1}} & \frac{\partial f_3}{\partial y_{k-1}} & \frac{\partial f_3}{\partial\theta_{k-1}}\end{bmatrix}=\begin{bmatrix}1 & 0 & -T(\sin\theta_{k-1}v_k)\\ 0 & 1 & T(\cos\theta_{k-1}v_k)\\ 0 & 0 & 1\end{bmatrix} \tag{45}$$

*Linearized motion noise (inverse).*

$$\mathbf{Q}^{-1}=\begin{bmatrix}\mathbf{Q}_0^{-1} & & & \\ & (\mathbf{F}_{\mathbf{w},1}\mathbf{Q}_0\mathbf{F}_{\mathbf{w},1}^{\mathrm{T}})^{-1} & & \\ & & (\mathbf{F}_{\mathbf{w},2}\mathbf{Q}_0\mathbf{F}_{\mathbf{w},2}^{\mathrm{T}})^{-1} & \\ & & & \ddots \\ & & & & (\mathbf{F}_{\mathbf{w},K-1}\mathbf{Q}_0\mathbf{F}_{\mathbf{w},K-1}^{\mathrm{T}})^{-1}\end{bmatrix} \tag{46}$$

*Motion noise Jacobian.*

$$\mathbf{F}_{\mathbf{w},k}=\frac{\partial\mathbf{f}(\mathbf{x}_{k-1},\mathbf{v}_k,\mathbf{w}_k)}{\partial\mathbf{w}_k}\bigg|_{\mathbf{x}_{k-1},\mathbf{v}_k,\mathbf{w}_k}=\begin{bmatrix}\frac{\partial f_1}{\partial w_{v_{x,k}}} & \frac{\partial f_1}{\partial w_{v_{y,k}}} & \frac{\partial f_1}{\partial w_{\omega_k}}\\ \frac{\partial f_2}{\partial w_{v_{x,k}}} & \frac{\partial f_2}{\partial w_{v_{y,k}}} & \frac{\partial f_2}{\partial w_{\omega_k}}\\ \frac{\partial f_3}{\partial w_{v_{x,k}}} & \frac{\partial f_3}{\partial w_{v_{y,k}}} & \frac{\partial f_3}{\partial w_{\omega_k}}\end{bmatrix}=\begin{bmatrix}T\cos\theta_{k-1} & -T\sin\theta_{k-1} & 0\\ T\sin\theta_{k-1} & T\cos\theta_{k-1} & 0\\ 0 & 0 & T\end{bmatrix} \tag{47}$$

*Linearized measurement model with respect to robot pose.*

$$\mathbf{G}_1=\begin{bmatrix}\frac{\partial\mathbf{g}(\mathbf{x}_1,\boldsymbol{\ell})}{\partial\mathbf{x}_1} & & & \\ & \frac{\partial\mathbf{g}(\mathbf{x}_2,\boldsymbol{\ell})}{\partial\mathbf{x}_2} & & \\ & & \ddots & \\ & & & \frac{\partial\mathbf{g}(\mathbf{x}_K,\boldsymbol{\ell})}{\partial\mathbf{x}_K}\end{bmatrix} \tag{48}$$

*Measurement model Jacobian with respect to pose.*

$$\frac{\partial \mathbf{g}(\mathbf{x}_k, \boldsymbol{\ell})}{\partial \mathbf{x}_k}\bigg|_{\mathbf{x}_k, \boldsymbol{\ell}, \mathbf{0}} = \begin{bmatrix} \frac{\partial g_1}{\partial x_k} & \frac{\partial g_1}{\partial y_k} & \frac{\partial g_1}{\partial \theta_k} \\ \frac{\partial g_2}{\partial x_k} & \frac{\partial g_2}{\partial y_k} & \frac{\partial g_2}{\partial \theta_k} \end{bmatrix} = \begin{bmatrix} \frac{-\ell_{x,n}+x_k}{\left\|\boldsymbol{\ell}_n - \begin{bmatrix} x_k \\ y_k \end{bmatrix}\right\|} & \frac{-\ell_{y,n}+y_k}{\left\|\boldsymbol{\ell}_n - \begin{bmatrix} x_k \\ y_k \end{bmatrix}\right\|} & 0 \\ \frac{\ell_{y,n}-y_k}{\left\|\boldsymbol{\ell}_n - \begin{bmatrix} x_k \\ y_k \end{bmatrix}\right\|^2} & \frac{-\ell_{x,n}+x_k}{\left\|\boldsymbol{\ell}_n - \begin{bmatrix} x_k \\ y_k \end{bmatrix}\right\|^2} & -1 \end{bmatrix} \tag{49}$$

*Linearized measurement model with respect to IAPs.*

$$\mathbf{G}_2 = \begin{bmatrix} \frac{\partial \mathbf{g}(\mathbf{x}_1, \boldsymbol{\ell})}{\partial \boldsymbol{\ell}} \\ \frac{\partial \mathbf{g}(\mathbf{x}_2, \boldsymbol{\ell})}{\partial \boldsymbol{\ell}} \\ \vdots \\ \frac{\partial \mathbf{g}(\mathbf{x}_K, \boldsymbol{\ell})}{\partial \boldsymbol{\ell}} \end{bmatrix} = \begin{bmatrix} \frac{\partial \mathbf{g}(\mathbf{x}_1, \boldsymbol{\ell})}{\partial \boldsymbol{\ell}_1} & \frac{\partial \mathbf{g}(\mathbf{x}_1, \boldsymbol{\ell})}{\partial \boldsymbol{\ell}_2} & \cdots & \frac{\partial \mathbf{g}(\mathbf{x}_1, \boldsymbol{\ell})}{\partial \boldsymbol{\ell}_{N-1}} & \frac{\partial \mathbf{g}(\mathbf{x}_1, \boldsymbol{\ell})}{\partial \boldsymbol{\ell}_N} \\ \frac{\partial \mathbf{g}(\mathbf{x}_2, \boldsymbol{\ell})}{\partial \boldsymbol{\ell}_1} & \frac{\partial \mathbf{g}(\mathbf{x}_2, \boldsymbol{\ell})}{\partial \boldsymbol{\ell}_2} & \cdots & \frac{\partial \mathbf{g}(\mathbf{x}_2, \boldsymbol{\ell})}{\partial \boldsymbol{\ell}_{N-1}} & \frac{\partial \mathbf{g}(\mathbf{x}_2, \boldsymbol{\ell})}{\partial \boldsymbol{\ell}_N} \\ \vdots & \vdots & & \vdots & \vdots \\ \frac{\partial \mathbf{g}(\mathbf{x}_K, \boldsymbol{\ell})}{\partial \boldsymbol{\ell}_1} & \frac{\partial \mathbf{g}(\mathbf{x}_K, \boldsymbol{\ell})}{\partial \boldsymbol{\ell}_2} & \cdots & \frac{\partial \mathbf{g}(\mathbf{x}_K, \boldsymbol{\ell})}{\partial \boldsymbol{\ell}_{N-1}} & \frac{\partial \mathbf{g}(\mathbf{x}_K, \boldsymbol{\ell})}{\partial \boldsymbol{\ell}_N} \end{bmatrix} \tag{50}$$

*Measurement model Jacobian with respect to IAPs.*

$$\frac{\partial \mathbf{g}(\mathbf{x}_k, \boldsymbol{\ell})}{\partial \boldsymbol{\ell}_n}\bigg|_{\mathbf{x}_k, \boldsymbol{\ell}, \mathbf{0}} = \begin{bmatrix} \frac{\partial g_1}{\partial \ell_{x,n}} & \frac{\partial g_1}{\partial \ell_{y,n}} & \frac{\partial g_1}{\partial d_{\text{fix},\boldsymbol{\ell}n}} \\ \frac{\partial g_2}{\partial \ell_{x,n}} & \frac{\partial g_2}{\partial \ell_{y,n}} & \frac{\partial g_2}{\partial d_{\text{fix},\boldsymbol{\ell}n}} \end{bmatrix} = \begin{bmatrix} \frac{\ell_{x,n}-x_k}{\left\|\boldsymbol{\ell}_n - \begin{bmatrix} x_k \\ y_k \end{bmatrix}\right\|} & \frac{\ell_{y,n}-y_k}{\left\|\boldsymbol{\ell}_n - \begin{bmatrix} x_k \\ y_k \end{bmatrix}\right\|} & 1 \\ \frac{-\ell_{y,n}+y_k}{\left\|\boldsymbol{\ell}_n - \begin{bmatrix} x_k \\ y_k \end{bmatrix}\right\|^2} & \frac{\ell_{x,n}-x_k}{\left\|\boldsymbol{\ell}_n - \begin{bmatrix} x_k \\ y_k \end{bmatrix}\right\|^2} & 0 \end{bmatrix} \tag{51}$$

*Linearized measurement noise (inverse).*

$$\mathbf{R}^{-1} = \begin{bmatrix} (\mathbf{G}_{\mathbf{n}_1}\mathbf{R}_0\mathbf{G}_{\mathbf{n}_1}^{\mathrm{T}})^{-1} & & & \\ & (\mathbf{G}_{\mathbf{n}_2}\mathbf{R}_0\mathbf{G}_{\mathbf{n}_2}^{\mathrm{T}})^{-1} & & \\ & & \ddots & \\ & & & (\mathbf{G}_{\mathbf{n}_K}\mathbf{R}_0\mathbf{G}_{\mathbf{n}_K}^{\mathrm{T}})^{-1} \end{bmatrix} \tag{52}$$

*Measurement noise Jacobian.*

$$\mathbf{G}_{\mathbf{n}_k} = \frac{\partial \mathbf{g}(\mathbf{x}_k, \boldsymbol{\ell})}{\partial \mathbf{n}_k} = \begin{bmatrix} \frac{\partial g_1}{\partial n_{d,k}} & \frac{\partial g_1}{\partial n_{\phi,k}} \\ \frac{\partial g_2}{\partial n_{d,k}} & \frac{\partial g_2}{\partial n_{\phi,k}} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{53}$$

## Batch loop closure

*Switchable constraint error:* Requires modifying the original cost function (not shown above).

$$e_{\mathrm{c}}(\boldsymbol{\ell}_*) = (d_{\text{fix},\boldsymbol{\ell}_*} - d_{\text{fix},\boldsymbol{\ell}_{**}}) - \|\boldsymbol{\ell}_* - \boldsymbol{\ell}_{**}\| \tag{54}$$

Note: the symbols $*$ and $**$ represent two arbitrary IAPs.

*Linearized switchable constraint noise.*

$$\mathbf{S}^{-1} = \begin{bmatrix} S_0^{-1} & & \\ & \ddots & \\ & & S_0^{-1} \end{bmatrix}, \quad S_0 = \sigma_{e_{\mathrm{c}}}^2 \tag{55}$$

*Dynamic covariance scaling.*

$$\hat{\mathbf{S}}^{-1} = w(\mathbf{e}_{\mathrm{c}})\,\mathbf{S}^{-1}, \quad \text{where } \mathbf{e}_{\mathrm{c}} \text{ is a vector of constraint errors} \tag{56}$$

$$w(\mathbf{e}_{\mathrm{c}}) = \begin{cases} e_{\mathrm{c}}^2 \le s \\ e_{\mathrm{c}}^2 > s \end{cases} \Bigg| \begin{array}{l} 1 \\ \frac{4s^2}{(s+e_{\mathrm{c}}^2)^2}, \end{array} \quad \text{where } w \to [0, 1] \tag{57}$$

**Batch solve**

*Lower Cholesky decomposition.*

$$\begin{aligned}
&\mathbf{A} = \mathbf{L}\mathbf{L}^{\mathrm{T}} \\
&\mathbf{L}\mathbf{p} = \mathbf{b} \quad \text{(solve for } \mathbf{p}\text{)} \\
&\mathbf{L}^{\mathrm{T}}\delta\mathbf{z} = \mathbf{p} \quad \text{(solve for } \delta\mathbf{z}\text{)} \\
&\mathbf{L}^{\mathrm{T}}\delta\mathbf{z} = \mathbf{L}^{-1}\mathbf{b}
\end{aligned} \tag{58}$$

$$\underbrace{\begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{12}^{\mathrm{T}} & \mathbf{A}_{22} \end{bmatrix}}_{\mathbf{A}} = \underbrace{\begin{bmatrix} \mathbf{L}_{11} & \mathbf{0} \\ \mathbf{L}_{12}^{\mathrm{T}} & \mathbf{L}_{22} \end{bmatrix}}_{\mathbf{L}} \underbrace{\begin{bmatrix} \mathbf{L}_{11} & \mathbf{L}_{12} \\ \mathbf{0} & \mathbf{L}_{22}^{\mathrm{T}} \end{bmatrix}}_{\mathbf{L}^{\mathrm{T}}} = \begin{bmatrix} \mathbf{L}_{11}^2 & \mathbf{L}_{11}\mathbf{L}_{12} \\ \mathbf{L}_{11}\mathbf{L}_{12}^{\mathrm{T}} & \mathbf{L}_{12}\mathbf{L}_{12}^{\mathrm{T}} + \mathbf{L}_{22}\mathbf{L}_{22}^{\mathrm{T}} \end{bmatrix}$$

$$\mathbf{L}^{-1} = \begin{bmatrix} \mathbf{L}_{11}^{-1} & \mathbf{0} \\ -\mathbf{L}_{11}^{-1}\mathbf{L}_{12}^{\mathrm{T}}\mathbf{L}_{22}^{-1} & \mathbf{L}_{22}^{-1} \end{bmatrix}$$