Hybrid Metric-Topological Localization for Robots in Pipe Networks

Rob Worley, Sean R. Anderson

***Abstract*—Accurate, reliable, and efficient robot localization is essential for long-term autonomous robotic inspection of buried pipe networks. It is necessary for path planning and for locating detected faults in the network. This paper proposes a novel localization algorithm designed for limited, high-uncertainty sensing in network environments. The localization method is developed from the Viterbi algorithm, which efficiently searches for the most likely robot trajectory amongst multiple hypotheses. It is augmented to facilitate hybrid metric-topological localization, and it is improved to efficiently spend computation on useful points in time. Results using field robot data from a sewer network demonstrate the algorithm's practical applicability, as the algorithm is shown to robustly produce a coherent trajectory estimate with low error in estimated location, compared to a particle filter alternative that incorrectly jumps between parts of the network. Results using simulated data demonstrate the algorithm's robust performance at large spatial and temporal scales. In 79% of trajectories, the algorithm produces less error than a particle filter, while requiring a median of 0.18 times the computation time, demonstrating a substantial improvement in computational efficiency with comparable or superior accuracy. The flexibility of the algorithm is also demonstrated in simulation by incorporating measurements representing acoustic echo sensing and pipe gradient sensing, which is shown to reduce the error rate from 28% to 7% or below, in the case of large uncertainty in all other inputs. These results demonstrate that the proposed localization method improves on the computational efficiency, accuracy and robustness of a particle filter specialised to the pipe environment, even in the presence of limited and high-uncertainty sensing.**

I. Introduction

Autonomous robots could improve the effectiveness and efficiency of buried pipe network inspection. These robots would operate within the pipes, detecting and monitoring faults over a large area and time period. Long-term autonomy in the pipe network requires reliable localization, which is the problem of how a robot estimates its position in the environment. This is more challenging here than in other robotics applications as GPS cannot be used underground to measure position, and the scale of the environment, at 400,000 kilometres of wastewater pipes in the UK (Department for International Trade, 2015) with 400,000 kilometres of water supply pipes, precludes alternative external position sensors.

Pipe networks are a special case of environment for robot localization. The discrete pipe and junction locations, as depicted in Figure 1, lend themselves to a topological environment representation. Moreover, a topological representation has been shown to be memory-efficient for localization and mapping at large scales (Aguiar et al., 2023), which would be required in application to pipe networks. However, while a number of sensing approaches have been developed in the literature (Aitken et al., 2021) for this environment, these approaches are limited to making observations of only the robot's motion or immediate surroundings, and a lack of distinguishing features (Edwards et al., 2023) makes recognising discrete locations challenging. The sensing available is therefore limited to metric measurements of distance travelled between detected locations. This motivates a hybrid metric-topological environment representation.

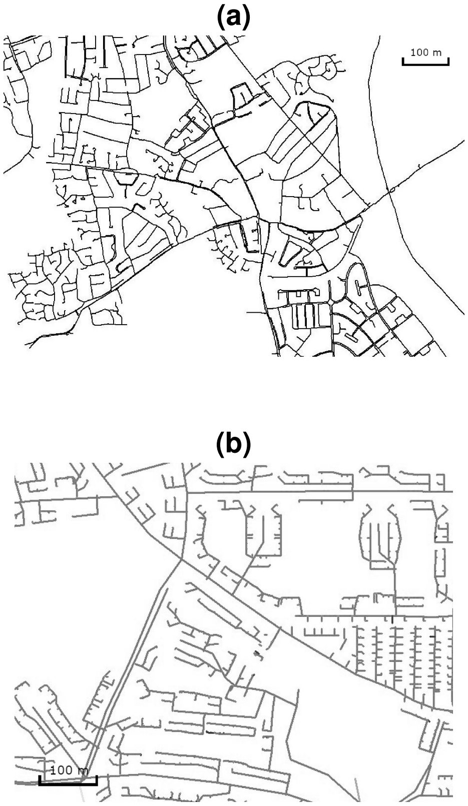

Fig. 1. Examples of **(a)** water distribution, and **(b)** wastewater pipe networks in towns in the UK.

Hybrid metric-topological localization using filtering has been developed generally (Blanco et al., 2008), and similar approaches have been applied to tracking targets in agricultural applications (Khan et al., 2020) (Polvara et al., 2021), and

This work is supported by EPSRC grant EP/S016813/1 Pervasive Sensing for Buried Pipes (Pipebots).

R. Worley and S. R. Anderson are with the Department of Automatic Control and Systems Engineering, University of Sheffield, Sheffield, UK. e-mail: {r.worley, s.anderson}@sheffield.ac.uk

localization of vehicles in road networks, where estimation is done using particle filtering (Merriaux et al., 2015). A combination of particle filtering and Kalman filtering with a Gaussian mixture model has been used elsewhere for vehicles in road networks (Brubaker et al., 2016), and a similar approach was augmented with semantic information to more uniquely identify places (Bernuy and Ruiz-del Solar, 2018). Filtering is motivated in application to road vehicles where estimation at a high rate is required for control and possibly path planning. Good results are found in that application, but improvements are found using semantic maps unavailable in the pipe environment, and accuracy and robustness in the case of limited, high-uncertainty sensing is not shown.

Methods based on *smoothing* or *optimization* can be used, which have the advantage of estimating the full robot trajectory using information over all time steps, rather than incremental position. Smoothing for hybrid metric-topological robot localization has been developed for more general robot environments where the discrete variables are not limited to the location of the robot (Doherty et al., 2022). The proposed solution is to alternately iteratively solve the continuous and the discrete parts of the problem. In the pipe environment, robot observations do not generally contain precise information regarding the discrete location, so the estimation of the discrete state alone must generally involve the continuous state. Similarly, the continuous robot pose depends heavily on the discrete state. Therefore, there is a need for an approach specialised to this environment.

A distinct approach to metric-topological localization is based on *map-merging* (Huang and Beevers, 2005); in this application merging a network map and a robot's trajectory, or robot map. In this application, the problem is a specific instance of this map-merging problem defined previously; the robot map is entirely contained within the network map, rather than partially overlapping, and the initial pose of the robot, part of the robot map, is known exactly in the network map. At the same time, the accumulating uncertainty from odometry means that the robot map would be relatively deformed, such that the rigid transform of the robot map would not bring it in alignment with the network map. These substantial differences and challenges motivate a different approach to the solution.

This paper addresses the general research gap in developing and evaluating hybrid metric-topological state space estimation in application to robots in pipe networks, where there is high uncertainty in sensing and significant limitations on computation. Robot localization in pipe networks has been done using particle filtering in a 2D metric state space (Alejo et al., 2019), and our previous work improved upon the robustness of this approach by constraining the estimate to a 1D hybrid metric-topological state space (Worley and Anderson, 2021b), (Worley and Anderson, 2021a). However, despite its improvements to robustness, our previous approach is shown in experiment to suffer from occasional mislocalization from which it fails to recover, producing large overall error in trajectory estimation. Furthermore, the computational cost of the particle filtering approach increases as measurement uncertainty increases, such that a robot constrained in sensing and computation will be unable to produce a satisfactory estimate with this approach. Therefore, further improvements to robustness and computational cost are required in this application.

This paper addresses the specific problem of developing an algorithm which improves localization robustness in the pipe environment while simultaneously reducing computational cost. Improving estimation robustness by more precise or thorough calculation of localization hypotheses is antithetical to the reduction in computation, posing a difficult challenge, but a challenge which must be solved to allow the deployment of localization algorithms on robots in this environment.

Computation for each step of the robot trajectory must be reduced so that a more thorough set of trajectory hypotheses can be considered while keeping the total computational cost low. Improving from previously developed particle filter approaches, the proposed algorithm utilises the low dimensionality of the network environment to represent hypothetical robot locations as a small set of estimates with some continuous position uncertainty, rather than a large set of discrete particles. This allows the use of a more thorough distribution of localization hypotheses, improving robustness, without simply adding to the computational cost. The efficient *Viterbi algorithm* is used as a starting point for computing the localization estimate. This algorithm is augmented for the continuous part of the state space by using the set of estimates, but is usefully constrained by the discrete part of the state space, such that a small set of distinct estimates can sufficiently represent the hypothetical robot trajectories. Moreover, the algorithm is improved by making estimates only for the subset of times at which the robot makes an informative measurement. Therefore, computation is spent for points in space and time at which it is most useful, allowing a more thorough, robust localization estimate while reducing computational cost.

The localization method is demonstrated on real-world robot data in a sewer network (Alejo et al., 2020) using the odometry measurements, manhole detection, and *a priori* map of the environment. The performance of the algorithm is benchmarked against particle filter methods designed for the pipe environment in metric (Alejo et al., 2019) and hybrid metric-topological (Worley and Anderson, 2021b) space. The robustness of the hybrid metric-topological approaches to measurement uncertainty is investigated using simulation data. These experiments show the superior accuracy and robustness of the proposed optimization-based trajectory estimation approach. Finally, there is an investigation into the use of different additional measurement types with this estimation method, including the use of measurement of pipe gradient in a three-dimensional pipe network.

II. Problem Definition

The pipe network environment is made up of pipes, or *links*, and connections between pipes, or *nodes*. These nodes are positioned in three-dimensional space, with a latitude, longitude, and elevation. The links are much larger in length than in width, and much larger than the nodes, so a close approximation of the network is a set of connected one dimensional places. The state space of the robot is therefore defined in this hybrid metric-topological way, and in this paper it is assumed that the map of the pipe network is known *a priori*. In practice, this map would be obtained from utility managers in the area in which the robot is operating, or from a resource like the National Underground Asset Register (Department for Science, Innovation and Technology, 2022) in the UK. This map $\mathcal{M}$ describes the neighbouring nodes and links for each node $n \in \mathcal{N}$, and describe the position in continuous space of each node $n \in \mathcal{N}$ and the start and end position of each link $l \in \mathcal{L}$.

A robot moves around the network of pipes, modelled as small discrete time steps. It is assumed that the robot moves purely linearly along the straight pipes in the network, and makes purely angular motion when turning at a connection between pipes. The robot's state is therefore defined by

$$\mathbf{s}_t = \left(\boldsymbol{\xi}_t = (i_t, d_t), \mathbf{x}_t = x_t\right) \tag{1}$$

where $i_t \in \mathcal{I}$ is the discrete location of the robot, represented by the index of the link (pipe) or node (junction or manhole), from the set of all locations $\mathcal{I} = \{\mathcal{L}, \mathcal{N}\}$ where $\mathcal{L}$ and $\mathcal{N}$ are the sets of all links and nodes. x_t is the distance from the origin of the link or node, $x_t \in \{x \in \mathbb{R} : 0 \leq x \leq X_{i_t}\}$, where X_{i_t} is the maximum distance for a location i_t. d_t is the discrete direction in the link or node. In a link, $d_t = d_t^l \in \{-1, 1\}$. In a node, $d_t = d_t^d d_t^n$, with $d_t^n \in \{1, 2, \dots, D^n\}$ where there are D^n adjacent links at node n, and $d_t^d \in \{-1, 1\}$ so the sign of the direction depends on whether the robot is arriving or leaving from a direction.

The aim is to estimate the robot's trajectory, given by $\mathbf{s}_{0:T} = \{\mathbf{s}_0, \mathbf{s}_1, \dots \mathbf{s}_t, \dots, \mathbf{s}_{T-1}, \mathbf{s}_T\}$. Specifically, the aim is to estimate the trajectory which maximises the posterior distribution $p(\mathbf{s}_{0:T} | \mathbf{u}_{1:T}, \mathbf{z}_{1:T}, \mathbf{s}_0)$, given by

$$\hat{\mathbf{s}}_{0:T} = \underset{\mathbf{s}_{0:T}}{\arg\max} \{p(\mathbf{s}_{0:T} | \mathbf{u}_{1:T}, \mathbf{z}_{1:T}, \mathbf{s}_0)\} \tag{2}$$

which is the probability distribution over possible state sequences $\mathbf{s}_{0:T}$ given an initial state $\mathbf{s}_0$ and the sequence of uncertain observations $\mathbf{u}_{1:T}$ and $\mathbf{z}_{1:T}$.

The observations are defined by

$$\mathbf{s}_t = f_u(\mathbf{s}_{t-1}, \mathbf{u}_t) \tag{3}$$

$$\mathbf{z}_t = f_z(\mathbf{s}_t) \tag{4}$$

$\mathbf{u}_t$ contains information about the transition between states $\mathbf{s}_{t-1}$ and $\mathbf{s}_t$, given by

$$\mathbf{u}_t = \{\Delta x_t, \Delta\theta_t\} \tag{5}$$

These *odometry* measurements could be obtained from the robot's command motion, or wheel encoders and inertial sensing. These measurements could be modelled as having normally distributed uncertainty, given by $\sigma_{\Delta x_t}$ and $\sigma_{\Delta\theta_t}$. This uncertainty accounts for deviations between the robot's measured motion and its actual motion, which could be due to the uneven terrain in the environment. Alternatively, the uncertainty might be better modelled as a time-varying bias, which might account for persistent deviations in motion due to the flow of water or obstacles like blockages in the pipes. The effect of both models are investigated in Section IV-B.

$\mathbf{z}_t$ contains information about the observations made in the current state $\mathbf{s}_t$. The simplest observation is the robot detecting whether it is at a pipe junction, or node $\mathcal{N}$, where $z_t = N$, or in a pipe, or link $\mathcal{L}$, where $z_t = L$. The measurement uncertainty is described by

$$p(z_t = L | i_t \in \mathcal{L}) = 1 - \beta_p \tag{6a}$$

$$p(z_t = L | i_t \in \mathcal{N}) = \beta_n \tag{6b}$$

$$p(z_t = N | i_t \in \mathcal{L}) = \beta_p \tag{6c}$$

$$p(z_t = N | i_t \in \mathcal{N}) = 1 - \beta_n \tag{6d}$$

where β_p and β_n are the probabilities of false positive and negative detection respectively.

These two measurement variables can be defined more specifically depending on the measurements available. Nearby junctions between pipes can be detected from vision with structured lighting (Lee et al., 2013; Kakogawa et al., 2017; Zhao et al., 2018; Kakogawa et al., 2015) or depth images (Thielemann et al., 2008), or from rangefinding (Kirchner and Hertzberg, 1997). Vision can also be used make odometry measurements which could be incorporated into $\mathbf{u}_t$ and into loop-closing measurements, although this is challenging due to the feature sparseness compared to other environments (Edwards et al., 2023). Lidar sensors are well developed for the task of object detection and classification in other environments (Wu et al., 2022), so could likely be used for detecting manholes, junctions between pipes, or other informative features in this application. Acoustic echoes can be used to measure the distance between the robot and distant features, and measure the length of the surrounding pipe (Worley et al., 2020; Worley et al., 2024). For long-term localization in realistic environments, these feature observations need to be robust to changing environmental conditions (Sousa et al., 2023). The pipe environment is dynamic, with conditions changing over a daily period, changing with weather, and degrading over time. Therefore, this work will assume initially that only simple binary detection of junctions, as described above, are available.

It is assumed that the interval between time index $t - 1$ and t is small, such that it takes many time steps for the robot to move from one node in the pipe network to another. The sequence of time indices is described as $\mathcal{T}_t = \{0, 1, 2, \dots, t, \dots, T-1, T\}$.

III. Methods

This section describes the proposed localization approach. In Section III-A it is proposed to transform from the real state space to a discrete state space using multi-hypothesis estimation, in which the *Viterbi algorithm* can be applied. An altered form of the state which is most suitable for this estimation is defined in Section III-B. The implementation is described, including the computation of the discrete multi-hypothesis state space (Section III-C2), the calculation of the required probabilities in this state space (Section III-C3), and the execution of the Viterbi algorithm using these discrete states and probabilities. The estimation is transformed back to the real state space in Section III-D, and the estimation of the remaining continuous variables using *Rauch-Tung-Striebel* smoothing is described. Finally, the use of additional sensing information is described in Section III-E.

A. Estimation Definition

Ultimately, the *maximum a posteriori* estimate of the most likely sequence of states $\hat{\mathbf{s}}_{0:T}$ is desired, given by equation 2. A particle distribution was used in previous work (Worley and Anderson, 2021b) to represent the distribution of pose $\mathbf{s}_t$ over the hybrid metric-topological state space. This sort of distribution is illustrated in Figure 2(a). This could be extended to estimation over all time, however:

- The reliability of particle-based estimation is difficult to determine, and although methods have been developed to address this (Akai, 2023), they require more informative observations than those described in equation 6.
- The application to small, resource-constrained robots limits available computation, and a particle distribution with a small number of estimates is desirable.
- Because much of the environment comprises one-dimensional linear sections, a small number of estimates each with some continuous uncertainty can equally well represent the pose distribution over the hybrid metric-topological state space.

These drawbacks to a particle-based approach motivate the improvements proposed in this paper.

The proposed improved approach is based on *multi-hypothesis* estimation, which is implemented here by maintaining a set $\mathcal{S}_t = \left\{\mathcal{S}_t^1, \mathcal{S}_t^2, \ldots, \mathcal{S}_t^j, \ldots\right\}$ of *estimates* or *hypotheses* of the state at each time t. Each estimate $\mathcal{S}_t^j$ contains a robot pose $\mathbf{s}_t$ and a corresponding probability. This is illustrated in Figure 2(b).

These estimates are computed iteratively over time. For time t, estimates are made by applying the measured odometry $\mathbf{u}_t$ to estimates $\mathcal{S}_{t-1}^j$ using the odometry model in equation 5. When the odometry gives a continuous position x_t beyond the end of a link location $i_t \in \mathcal{L}$, estimates are made in each adjacent link location. Estimates are made in nearby node locations $i_t \in \mathcal{N}$, the likelihood of which depends on estimates in $\mathcal{S}_{t-1}$ and the observation $\mathbf{z}_t$ and the model in equation 6. This produces a set of estimates $\mathcal{S}_t$ for all time steps.

Much of the state space is discrete, and the continuous space is effectively discretised by the set of estimates (Godsill et al., 2001). This gives a purely discrete state space $\mathbf{S}_{0:T}$ defined for each time t as

$$\mathbf{S}_t \in \left\{\mathbf{S}_t^1, \mathbf{S}_t^2, \ldots, \mathbf{S}_t^j, \ldots, \mathbf{S}_t^J\right\} \tag{7}$$

where the number of possible states J may be different for each time t. Each possible state $\mathbf{S}_t^j$ is an instance of the state $\mathbf{s}_t$ defined in equation 1.

Given a good set of estimates, the solution to the *maximum a posteriori* estimation problem in this discrete state trajectory gives a solution approximately equal to the *maximum a posteriori* estimate of the real state trajectory $\mathbf{s}_{0:T}$

$$\hat{\mathbf{s}}_{0:T} \approx \hat{\mathbf{S}}_{0:T} = \underset{\mathbf{S}_{0:T}}{\arg\max} \left\{P(\mathbf{S}_{0:T}|\mathbf{u}_{1:T}, \mathbf{z}_{1:T}, \mathbf{S}_0)\right\} \tag{8}$$

In the purely discrete state space $\mathbf{S}_{0:T}$, the *Viterbi algorithm* (Forney, 1973) can estimate $\hat{\mathbf{S}}_{0:T}$. This algorithm finds the most likely trajectory which arrives at each state $\mathbf{S}_t^j \in \mathbf{S}_t$ for $t \in \mathcal{T}_t$, thus finding $\hat{\mathbf{S}}_{0:T}$ at time T. The algorithm efficiently only considers the most likely trajectory which arrives at each state $\mathbf{S}_t^j$; therefore many hypotheses in $\mathcal{S}_t$ can be discarded, saving redundant computation.

However, the Viterbi algorithm does not usefully work in this case when estimating at time t, which motivates some improvement. Most states in $\mathbf{S}_t$ are in link locations $i_t \in \mathcal{L}$, with a nonzero estimate of continuous position $\mathbf{x}_t$, as illustrated in Figure 2(b). For these states $\mathbf{S}_t^j$ and corresponding estimates $\mathcal{S}_t^j$ the most likely previous state will simply be the estimate in $\mathcal{S}_{t-1}$ from which it was created, so the algorithm's process of discarding unlikely trajectories which have arrived at the same state $\mathbf{S}_t^j$ will not function. However, the algorithm still requires the computation of many estimates and probabilities which add little information to the estimation of the overall trajectory. Therefore, the set of time indices over which estimation is done can be reduced to those at which the Viterbi algorithm can be usefully applied, reducing the computational cost without affecting accuracy, and therefore improving the algorithm's performance. This is illustrated in Figure 2(c).

B. State Definition

To improve upon the Viterbi algorithm, a reduced set of time indices $\mathcal{T}_t(\tau \in \mathcal{T}_\tau) \subset \mathcal{T}_t$ is found which simply gives the set of time indices where an *informative* measurement is made. Here, this is a measurement which gives more information than other measurements, when the robot either:

- Makes an observation between the robot pose at time t and another robot pose that is not time $t-1$, and therefore provides more information than odometry, for example a loop-closure measurement.
- Makes an observation $\mathbf{u}_t$ between the robot poses $\mathbf{x}_t$ and $\mathbf{x}_{t-1}$ which includes some angular change, rather than purely linear odometry.
- Makes an observation which gives some information about the location of the robot $\mathbf{x}_t$ at time t given the understanding of the environment, i.e. results in a nonuniform distribution $p(\mathbf{z}_t|\mathbf{s}_t)$. This could be either:
  - A simple observation that the robot is in a node. The majority of the space in the network is made up of links and a small minority of the space (an infinitesimal volume in the model, and small in reality) is made up of nodes, so an observation of a node carries far more information than an observation of a link. This has been noted in the literature on localization on road networks (Bernuy and Ruiz-del Solar, 2018).
  - An observation of some discrete or categorical quality of the robot's surroundings, such as the material or colour.
  - An observation of some continuous metric quantity of the robot's surroundings, such as the length of a pipe, or the relative angles at which pipes leave a connection.

The reduced set of time indices is described as $\mathcal{T}_\tau = \left\{0, 1, 2, \ldots, \tau, \ldots, T\right\}$.

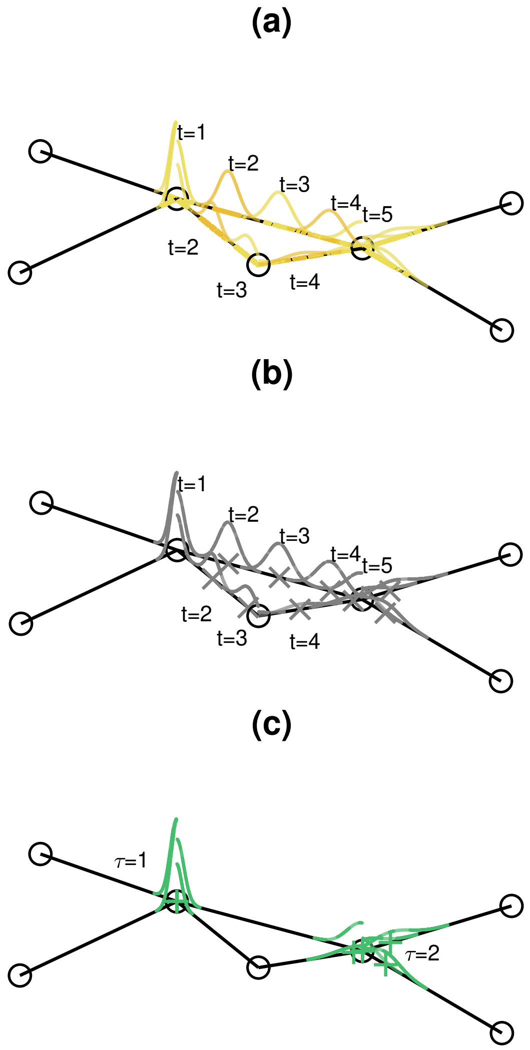

Fig. 2. An illustration of an example set of probability distributions across a pipe network over time. The pipe network nodes $\mathcal{N}$ are shown as circles, and the links $\mathcal{L}$ between nodes as black lines. These distributions are represented by particle distributions (yellow crosses, with an approximation of the distributions in yellow lines) as used in previous work, and by multi-hypothesis distributions (grey and green crosses and lines). Distributions are shown for the original time indices $t \in \{1, 2, 3, 4, 5\}$, and for the reduced set of time indices $\tau \in \{1, 2\}$.

The state space at time index τ is defined as

$$\mathbf{S}_\tau \in \left\{\mathbf{S}_\tau^1, \mathbf{S}_\tau^2, \ldots, \mathbf{S}_\tau^j, \ldots, \mathbf{S}_\tau^J\right\} \tag{9}$$

like the definition for $\mathbf{S}_t$ given in equation 7. Each possible state $\mathbf{S}_\tau^j$ is an instance of the state $\mathbf{s}_{\mathcal{T}_t(\tau)}$ defined as in equation 1. The *maximum a posteriori* estimate of the trajectory is desired, given by

$$\hat{\mathbf{S}}_{0:T} = \underset{\mathbf{S}_{0:T}}{\arg\max} \left\{P(\mathbf{S}_{0:T} | \mathbf{u}_{1:T}, \mathbf{z}_{1:T}, \mathbf{S}_0)\right\} \tag{10}$$

The motion measurement, like in equation 5, is $\mathbf{u}_\tau = \left\{\Delta x_\tau, \Delta\theta_\tau\right\}$. At time index τ, it is modelled that the robot turns, making a measurement of change in angle $\Delta\theta_\tau$ by integrating over a period of angular motion, then moves, making a measurement of change in position Δx_τ by integrating over a period of linear motion, given as

$$\Delta x_\tau = \sum_{t=\mathcal{T}_t(\tau-1)}^{\mathcal{T}_t(\tau)} \Delta x_t \tag{11}$$

The uncertainty in this measurement can be given by

$$\sigma_{\Delta x_\tau} = \sqrt{\sum_{t=\mathcal{T}_t(\tau-1)}^{\mathcal{T}_t(\tau)} {\sigma_{\Delta x_t}}^2} \tag{12}$$

If the value of uncertainty $\sigma_{\Delta x_t}$ at each time t is constant, this can be simplified to

$$\sigma_{\Delta x_\tau} = \sqrt{T_{\tau-1}^{\tau} {\sigma_{\Delta x_t}}^2} = \sqrt{T_{\tau-1}^{\tau}} \sigma_{\Delta x_t} \tag{13}$$

where $T_{\tau-1}^{\tau}$ is the number of discrete time steps t between reduced time indices $\tau-1$ and τ. As the robot is constrained to move in a mostly one-dimensional environment, sequences of purely linear odometry measurements can be simply integrated using a linear motion model without adding the error which would be found when using a linear function to represent the output from a nonlinear model. The cost of estimation can therefore be reduced without adding error to the estimation.

Given the definition of the informative measurements, the environment measurement is generally simply given by $\mathbf{z}_\tau = z_\tau = N$ with uncertainty defined by equation 6.

C. Discrete Localization Implementation

1) Viterbi Algorithm: The *maximum a posteriori* estimate of the most likely sequence of discrete states $\mathbf{S}_{0:T}$ over time $\tau \in \mathcal{T}_\tau$ is desired, given by equation 10. The discrete state space $\mathbf{S}_\tau \in \left\{\mathbf{S}_\tau^1, \mathbf{S}_\tau^2, \ldots, \mathbf{S}_\tau^j, \ldots, \mathbf{S}_\tau^J\right\}$ is defined by a set of *estimates* or *hypotheses* $\mathcal{S}_\tau = \left\{\mathcal{S}_\tau^1, \mathcal{S}_\tau^2, \ldots, \mathcal{S}_\tau^j, \ldots\right\}$.

The Viterbi algorithm estimates the probability $Q(\mathbf{S}_{0:\tau}^j)$ of the most likely sequence which ends at each possible state $\mathbf{S}_\tau^j$ in the state space. This is done recursively forward through time $\tau \in \mathcal{T}_\tau$, using the information in $\mathbf{u}_\tau$ and $\mathbf{z}_\tau$. The algorithm also records the index K_τ^j of state at the previous time step which most likely proceeds each state $\mathbf{S}_\tau^j$. At the end of this forward recursion, at time T, the state at which the most likely sequence ends, $\hat{\mathbf{S}}_T$, is found, and the most likely sequence $\hat{\mathbf{S}}_{0:T}$ is found using the recorded most likely previous states K_τ^j starting with $\hat{\mathbf{S}}_T$ and stepping backwards through time, achieving the estimation given by equation 8. This is described by Algorithm 1.

To include the necessary information for the Viterbi algorithm, each *estimate* in $\mathcal{S}_\tau^j \in \mathcal{S}_\tau$ is described by

$$\begin{aligned}\mathcal{S}_\tau^j = \Big\{\mathbf{S}_\tau^j = \left(\boldsymbol{\xi}_\tau^j = \left(i_\tau^j, d_\tau^j\right), \mathbf{x}_\tau^j = x_\tau^j\right), \\ K_\tau^j, Q(\mathbf{S}_{0:\tau}^j), \sigma_{\mathbf{x}_\tau}^j\Big\}\end{aligned} \tag{14}$$

which includes the uncertainty $\sigma_{\mathbf{x}_\tau}^j$ of the estimate of the continuous variable $\mathbf{x}_\tau^j$.

The estimates are computed iteratively through time τ, so the discrete state space $\mathbf{S}_\tau$ and the state probability are calculated simultaneously forwards through time. The computation of state and probability at each time step is described in the following two sections.

Algorithm 1 The Viterbi algorithm implementation

```
Initialise tables S, Q, K
to store states and probabilities through time $\tau$
S$[\tau \in \mathcal{T}_\tau][j \in \{1, 2, \dots\}]$
Q$[\tau \in \mathcal{T}_\tau][j \in \{1, 2, \dots\}]$
K$[\tau \in \mathcal{T}_\tau][j \in \{1, 2, \dots\}]$
For each time index forward through time
for time $\tau \in \mathcal{T}_\tau = \{1, 2, \dots, T\}$ do
    Get the measurements for time $\tau$
    get $\mathbf{u}_\tau$ and $\mathbf{z}_\tau$
    for state index $j \in \{1, 2, \dots\}$ do
        Compute the estimated state
        $\texttt{S}[\tau][\texttt{j}] \leftarrow \{\mathbf{S}^j_\tau = (\boldsymbol{\xi}_\tau = (i_\tau, d_\tau), \mathbf{x}_\tau = x_\tau), \sigma^j_{\mathbf{x}_\tau}\}$
        Compute the probability of the most likely sequence
                of states ending at this state
        $\texttt{Q}[\tau][\texttt{j}] \leftarrow \texttt{max}_\texttt{k}\{P(\mathbf{S}^j_\tau | \mathbf{u}_\tau, \mathbf{z}_\tau, \mathbf{S}^k_{\tau-1})\texttt{Q}[\tau - 1][\texttt{k}]\}$
        Compute the index of the most likely previous state
        $\texttt{K}[\tau][\texttt{j}] \leftarrow \texttt{argmax}_\texttt{k}\{P(\mathbf{S}^j_\tau | \mathbf{u}_\tau, \mathbf{z}_\tau, \mathbf{S}^k_{\tau-1})\texttt{Q}[\tau - 1][\texttt{k}]\}$
    end for
end for

Initialise table J to store the indices corresponding to the most
likely trajectory
J$[\tau \in \mathcal{T}_\tau]$
Find the estimate at time $T$ with the highest likelihood
Find the largest value in Q[T]
$\texttt{J[T]} \leftarrow \texttt{argmax}_\texttt{j}\{\texttt{Q[T][j]}\}$

For each time index backward though time
for time $\tau \in \mathcal{T}_\tau = \{T, T - 1, \dots, 2\}$ do
    Get the index of the most likely previous state
    $\texttt{J}[\tau - 1] \leftarrow \texttt{K}[\tau][\texttt{J}[\tau]]$
    Get the state at this index
    $\hat{\mathbf{S}}_{\tau-1} \leftarrow \texttt{S}[\tau - 1][\texttt{J}[\tau - 1]]$
end for
```

2) Computing Hypothetical Discrete States: Hypotheses of the state $\mathbf{S}^j_\tau = \left(\boldsymbol{\xi}_\tau = \left(i^j_\tau, d^j_\tau\right), \mathbf{x}^j_\tau = x^j_\tau\right)$ at time τ, are desired. To allow recovery of the trajectory at time indices t, it is useful to estimate the most likely sequence of discrete locations $\boldsymbol{\xi}^j_{\tau-1:\tau}$ between states $\mathbf{S}^k_{\tau-1}$ and $\mathbf{S}^j_\tau$. Therefore, hypotheses of the state sequence $\mathbf{S}^j_{\tau-1:\tau} = \left(\boldsymbol{\xi}^j_{\tau-1:\tau} = \left(i^j_{\tau-1:\tau}, d^j_{\tau-1:\tau}\right), \mathbf{x}^j_\tau = x^j_\tau\right)$ are found.

For each estimate $\mathcal{S}^k_{\tau-1} \in \mathcal{S}_{\tau-1}$, hypotheses of the state sequence $\mathbf{S}^j_{\tau-1:\tau}$ can be found by searching through the state space defined by the pipe network map $\mathcal{M}$. The discrete network environment usefully constrains the state space such that the hypothetical state sequences can be represented by a small number of distinct estimates.

For a given discrete state $\boldsymbol{\xi}^j_\tau = \left(i^j_\tau, d^j_\tau\right)$, where $i^j_\tau \in \mathcal{I}$ and $d^j_\tau \in \mathcal{D}^i$, there are multiple possible values for continuous position $\mathbf{x}^j_\tau = x^j_\tau$.

A predicted value of the continuous state $\tilde{x}^k_\tau$ is given by

$$\tilde{x}^k_\tau = x^k_{\tau-1} + \Delta x^j_\tau \tag{15}$$

where Δx^j_τ is part of the odometry measurement $\mathbf{u}_\tau$.

The predicted value $\tilde{x}^k_\tau$ is transformed to the coordinate frame of the discrete state $\boldsymbol{\xi}^j_\tau$ using knowledge of the path $\boldsymbol{\xi}^j_{\tau-1:\tau}$ between states $\mathbf{S}^k_{\tau-1}$ and $\boldsymbol{\xi}^j_\tau$.

$$x^j_\tau = f^v_x(\tilde{x}^k_\tau, \boldsymbol{\xi}^j_{\tau-1:\tau}) \tag{16}$$

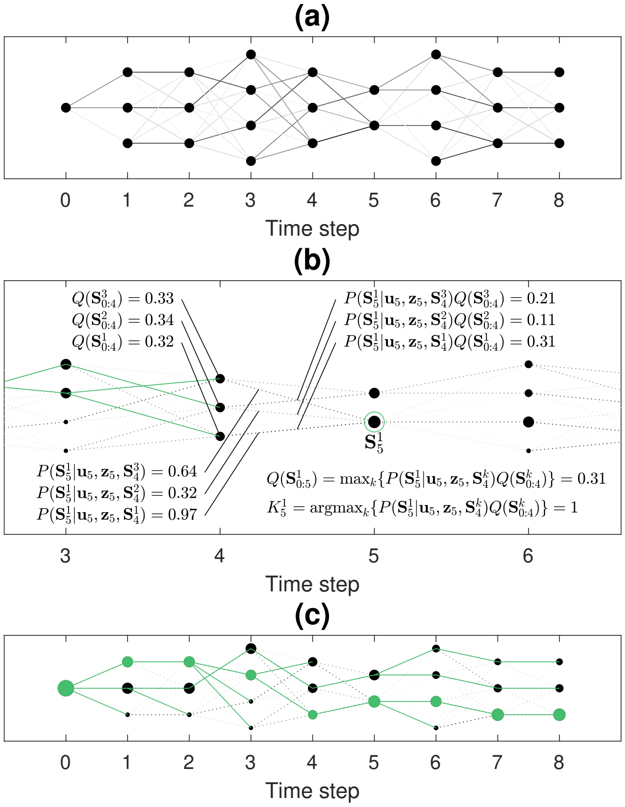
(a)
0
1
2
3
4
5
6
7
8
Time step
(b)
$Q(\mathbf{S}^3_{0:4}) = 0.33$
$Q(\mathbf{S}^2_{0:4}) = 0.34$
$Q(\mathbf{S}^1_{0:4}) = 0.32$
$P(\mathbf{S}^1_5|\mathbf{u}_5, \mathbf{z}_5, \mathbf{S}^3_4)Q(\mathbf{S}^3_{0:4}) = 0.21$
$P(\mathbf{S}^1_5|\mathbf{u}_5, \mathbf{z}_5, \mathbf{S}^2_4)Q(\mathbf{S}^2_{0:4}) = 0.11$
$P(\mathbf{S}^1_5|\mathbf{u}_5, \mathbf{z}_5, \mathbf{S}^1_4)Q(\mathbf{S}^1_{0:4}) = 0.31$
$\mathbf{S}^1_5$
$P(\mathbf{S}^1_5|\mathbf{u}_5, \mathbf{z}_5, \mathbf{S}^3_4) = 0.64$
$P(\mathbf{S}^1_5|\mathbf{u}_5, \mathbf{z}_5, \mathbf{S}^2_4) = 0.32$
$P(\mathbf{S}^1_5|\mathbf{u}_5, \mathbf{z}_5, \mathbf{S}^1_4) = 0.97$
$Q(\mathbf{S}^1_{0:5}) = \max_k\{P(\mathbf{S}^1_5|\mathbf{u}_5, \mathbf{z}_5, \mathbf{S}^k_4)Q(\mathbf{S}^k_{0:4})\} = 0.31$
$K^1_5 = \text{argmax}_k\{P(\mathbf{S}^1_5|\mathbf{u}_5, \mathbf{z}_5, \mathbf{S}^k_4)Q(\mathbf{S}^k_{0:4})\} = 1$
3
4
5
6
Time step
(c)
0
1
2
3
4
5
6
7
8
Time step

Fig. 3. Illustrations of the discrete state space $\mathbf{S}_\tau$ and the Viterbi algorithm. Each discrete state is shown as a point. **(a)**: The likelihood of transition between each state, where a darker line illustrates a more likely transition. **(b)**: The calculation at time $\tau = 5$ of the likelihood $Q(\mathbf{S}^1_{0:5})$. **(c)**: The most likely sequences ending at each state $\mathbf{S}^j_\tau$ illustrated with green lines, and the most likely sequence using information up to time step $\tau = 8$ illustrated as green points.

The ultimate *maximum a posteriori* estimate $\hat{x}^j_\tau$ must be within the boundary $\mathbf{X}^{i^j} = (0, X^{L,i^j})$ of the discrete state ξ^j_τ. However, the estimate x^j_τ can be outside of this range to allow accurate computation of future estimates.

This gives the state $\mathbf{S}^j_\tau = \left(\boldsymbol{\xi}^j_\tau = \left(i^j_\tau, d^j_\tau\right), \mathbf{x}^j_\tau = x^j_\tau\right)$ in equation 14.

3) Computing State Probability: For each estimate $\mathcal{S}^j_\tau$, with a state $\mathbf{S}^j_\tau$, the probability $Q(\mathbf{S}^j_{0:\tau})$, which is the probability of the most likely sequence of states which arrives at state $\mathbf{S}^j$ at time τ, is found. Where $\mathbf{S}^j_{0:\tau}$ contains the sequence of states at times $\mathcal{T}_\tau$ as well as intermediate discrete locations, this probability can be written (Thrun et al., 2006) as

$$\begin{aligned} Q(\mathbf{S}^j_{0:\tau}) &= P(\mathbf{S}^j_{0:\tau}|\mathbf{u}_{1:\tau}, \mathbf{z}_{1:\tau}, \mathbf{S}_0) \\ &= P(\mathbf{S}^j_{\tau-1:\tau}|\mathbf{u}_\tau, \mathbf{z}_\tau, \mathbf{S}^k_{\tau-1})Q(\mathbf{S}^k_{0:\tau-1}) \\ &= P(\mathbf{z}_\tau|\mathbf{S}^j_{\tau-1:\tau})P(\mathbf{S}^j_{\tau-1:\tau}|\mathbf{u}_\tau, \mathbf{S}^k_{\tau-1})Q(\mathbf{S}^k_{0:\tau-1}) \\ &= P(\mathbf{z}_\tau|\mathbf{S}^j_\tau)P(\mathbf{S}^j_{\tau-1:\tau}|\mathbf{u}_\tau, \mathbf{S}^k_{\tau-1})Q(\mathbf{S}^k_{0:\tau-1}) \end{aligned} \tag{17}$$

The motion model term can be written as

$$\begin{aligned} P(\mathbf{S}^j_{\tau-1:\tau}|\mathbf{u}_\tau, \mathbf{S}^k_{\tau-1}) &= P(\boldsymbol{\xi}^j_{\tau-1:\tau}, \mathbf{x}^j_\tau|\mathbf{u}_\tau, \mathbf{S}^k_{\tau-1}) \\ &= p(\mathbf{x}^j_\tau|\boldsymbol{\xi}^j_{\tau-1:\tau}, \mathbf{u}_\tau, \mathbf{S}^k_{\tau-1})P(\boldsymbol{\xi}^j_{\tau-1:\tau}|\mathbf{u}_\tau, \mathbf{S}^k_{\tau-1}) \end{aligned} \tag{18}$$

using the definition of conditional probability.

The continuous component is a probability density over the value of the predicted state given by equation 16

$$p(\mathbf{x}_\tau^j|\boldsymbol{\xi}_{\tau-1:\tau}^j,\mathbf{u}_\tau,\mathbf{S}_{\tau-1}^k) = \mathcal{N}(\mu = x_\tau^j, \sigma = \sqrt{{\sigma_{\Delta x_\tau}}^2 + {\sigma_{\mathbf{x}_{\tau-1}}^k}^2}) \tag{19}$$

This density is the uncertainty in the estimated continuous position $\mathbf{x}_\tau^j$ for hypothesis $\mathcal{S}_\tau^j$.

The discrete motion model term can be written as

$$\begin{aligned} P(\boldsymbol{\xi}_{\tau-1:\tau}^j|\mathbf{u}_\tau,\mathbf{S}_{\tau-1}^k) & \\ \propto P(\mathbf{u}_\tau|\boldsymbol{\xi}_{\tau-1:\tau}^j,\mathbf{S}_{\tau-1}^k)P(\boldsymbol{\xi}_{\tau-1:\tau}^j|\mathbf{S}_{\tau-1}^k) & \\ = P(\Delta x_\tau, \Delta\theta_\tau|\boldsymbol{\xi}_{\tau-1:\tau}^j,\mathbf{S}_{\tau-1}^k)P(\boldsymbol{\xi}_{\tau-1:\tau}^j|\mathbf{S}_{\tau-1}^k) & \\ = P(\Delta x_\tau|\boldsymbol{\xi}_{\tau-1:\tau}^j,\mathbf{S}_{\tau-1}^k)P(\Delta\theta_\tau|\boldsymbol{\xi}_{\tau-1:\tau}^j,\mathbf{S}_{\tau-1}^k) & \\ P(\boldsymbol{\xi}_{\tau-1:\tau}^j|\mathbf{S}_{\tau-1}^k) & \end{aligned} \tag{20}$$

using Bayes' theorem.

The linear motion component is proportional to the integral of the probability density over predicted continuous position, given by equation 19, over the range $\mathbf{X}^{i^j} = \left(0, X^{L,i^j}\right)$ (Brubaker et al., 2016)

$$\begin{aligned} & P(\Delta x_\tau|\boldsymbol{\xi}_{\tau-1:\tau}^j,\mathbf{S}_{\tau-1}^k) \\ & \propto \int_0^{X^{L,i^j}} p(\mathbf{x}_\tau^j|\boldsymbol{\xi}_{\tau-1:\tau}^j,\mathbf{u}_\tau,\mathbf{S}_{\tau-1}^k)d\mathbf{x}_\tau^j \end{aligned} \tag{21}$$

The angular motion component $P(\Delta\theta_\tau|\boldsymbol{\xi}_{\tau-1:\tau}^j, S_{\tau-1}^k)$ is calculated using the expected angular measurement for the path $\boldsymbol{\xi}_{\tau-1:\tau}^j$ by assuming there is heavy-tailed normally distributed uncertainty in the measurement $\Delta\theta_\tau$.

The path $\boldsymbol{\xi}_{\tau-1:\tau}^j$ from $\mathbf{S}^k$ to $\mathbf{S}^j$ may be via other discrete node locations in the environment, because of the nonzero probability β_n that the robot fails to perceive that it is in a node, described in equation 6. The number of missed nodes for a given path is computed

$$\bar{\mathbf{z}}^{j,k} = f_z^\xi(\boldsymbol{\xi}_{\tau-1:\tau}^j, \mathbf{S}_{\tau-1}^k) \tag{22}$$

and the probability of this is

$$P(\bar{\mathbf{z}}^{j,k}) = {\beta_n}^{\bar{\mathbf{z}}^{j,k}} \tag{23}$$

The probability of the path $\boldsymbol{\xi}_{\tau-1:\tau}^j$ is reduced by a factor equal to the product of the reciprocal of the *degree* $\Theta^{j,k}$, or number of neighbours, of each missed node $\boldsymbol{\xi}^{j,k}$.

$$\Theta^{j,k} = \left\{\Theta_1^{j,k}, \Theta_2^{j,k}, \dots\right\} = f_\Theta^\xi(\boldsymbol{\xi}_{\tau-1:\tau}^j, \mathbf{S}_{\tau-1}^k) \tag{24}$$

$$P(\Theta^{j,k}) = \prod_{a=1}^{\bar{\mathbf{z}}^{j,k}} {\Theta_a^{j,k}}^{-1} \tag{25}$$

The path component $P(\boldsymbol{\xi}_{\tau-1:\tau}^j|\mathbf{S}_{\tau-1}^k)$ is given by

$$P(\boldsymbol{\xi}_{\tau-1:\tau}^j|\mathbf{S}_{\tau-1}^k) = P(\bar{\mathbf{z}}^{j,k})P(\Theta^{j,k}) \tag{26}$$

The probability $P(\mathbf{z}_\tau|\mathbf{S}_\tau^j)$ of the observation $\mathbf{z}_\tau$ is found using the measurement model in equation 6.

Algorithm 2 The Viterbi estimate comparison

```
Compare each state in the set of estimates
for each pair of estimates in the set of estimates
for $\tilde{\mathcal{S}}_\tau^{j_1}, \tilde{\mathcal{S}}_\tau^{j_2} \in \tilde{\mathcal{S}}_\tau, j_1 \neq j_2$ do
    compare the two states
    Get the estimated state sequences $\mathbf{S}_{\tau-1:\tau}^{j_1}$, $\mathbf{S}_{\tau-1:\tau}^{j_2}$
    Get the final states in each sequence $\mathbf{S}_\tau^{j_1}$, $\mathbf{S}_\tau^{j_2}$
    if $\mathbf{S}_\tau^{j_1} = \mathbf{S}_\tau^{j_2}$ then
        if the two states are the same, remove the
                state with the lower probability
        $K = \arg\min_{k \in \{j_1, j_2\}} p(\mathbf{S}_{0:\tau}^k|\mathbf{u}_{1:\tau}, \mathbf{z}_{1:\tau}, \mathbf{S}_0)$
        $\mathcal{S}_\tau^K = \emptyset$
    end if
end for
```

The probability of arriving at state $\mathbf{S}_\tau^j$ from state $\mathbf{S}_{\tau-1}^k$ via path $\boldsymbol{\xi}_{\tau-1:\tau}^j$ is found by multiplying the observation, path, and motion probabilities together, as well as the model for uncertainty in continuous position given by equation 19

$$\begin{aligned} & P(\mathbf{S}_{\tau-1:\tau}^j|\mathbf{u}_\tau, \mathbf{z}_\tau, \mathbf{S}_{\tau-1}^k) = \\ & p(\mathbf{x}_\tau^j|\boldsymbol{\xi}_{\tau-1:\tau}^j, \mathbf{u}_\tau, \mathbf{S}_{\tau-1}^k)P(\mathbf{z}_\tau|\mathbf{S}_\tau^j) \\ & P(\boldsymbol{\xi}_{\tau-1:\tau}^j j|\mathbf{S}_{\tau-1}^k)P(\Delta\theta_\tau|\boldsymbol{\xi}_{\tau-1:\tau}^j, S_{\tau-1}^k) \\ & P(\Delta x_\tau|\boldsymbol{\xi}_{\tau-1:\tau}^j, \mathbf{S}_{\tau-1}^k) \end{aligned} \tag{27}$$

After computing the set of estimates $\mathcal{S}_\tau$, some of the estimates $\mathcal{S}_\tau^j$ and corresponding sequences $\mathbf{S}_{\tau-1:\tau}^j$ may have identical values for state $\mathbf{S}_\tau^j$, because much of the state space is purely discrete. For the Viterbi algorithm, only the most likely sequence arriving at a state $\mathbf{S}_\tau^j$ is needed and less likely estimates are removed from the set. Conceptually, this is computed by

$$Q(\mathbf{S}_{0:\tau}^j) = \max_k \left\{P(\mathbf{S}_{\tau-1:\tau}^j|\mathbf{u}_\tau, \mathbf{z}_\tau, \mathbf{S}_{\tau-1}^k)Q(\mathbf{S}_{0:\tau-1}^k)\right\} \tag{28}$$

and

$$K_\tau^j = \arg\max_k \left\{P(\mathbf{S}_{\tau-1:\tau}^j|\mathbf{u}_\tau, \mathbf{z}_\tau, \mathbf{S}_{\tau-1}^k)Q(\mathbf{S}_{0:\tau-1}^k)\right\} \tag{29}$$

which give the likelihood $Q(\mathbf{S}_{0:\tau}^j)$ of the most likely trajectory ending at state $\mathbf{S}_\tau^j$ and the index K_τ^j most likely preceding state $\mathbf{S}_{\tau-1}^k$. Practically, this is computed by Algorithm 2. Figure 3 illustrates this process with an example, showing the discrete state space, computation of equations 28 and 29, and the resulting estimated trajectory through the discrete state space.

The distribution $p(\mathbf{x}_\tau^j|\boldsymbol{\xi}_{\tau-1:\tau}^j, \mathbf{u}_\tau, \mathbf{S}_{\tau-1}^k)$ over continuous position can be parameterised by the uncertainty

$$\sigma_{\mathbf{x}_\tau}^j = \begin{cases} 0 & \text{if } i^j \in \mathcal{N} \\ \sqrt{{\sigma_{\Delta x_\tau}}^2 + {\sigma_{\mathbf{x}_{\tau-1}}^k}^2} & \text{if } i^j \in \mathcal{L} \end{cases} \tag{30}$$

The probabilities and indices calculated are sufficient for the computation of the Viterbi algorithm as defined in equation 14 and therefore the estimation of the most likely discrete state trajectory $\hat{\mathbf{S}}_{0:T}$. Because it is assumed that the state $\mathbf{S}_\tau^j$ has the *Markov property*, which is that the likelihood of future states is independent of past states given $\mathbf{S}_\tau^j$, the sequence $\mathbf{S}_{\tau-1:\tau}^j$ can be estimated without any change to the Viterbi algorithm.

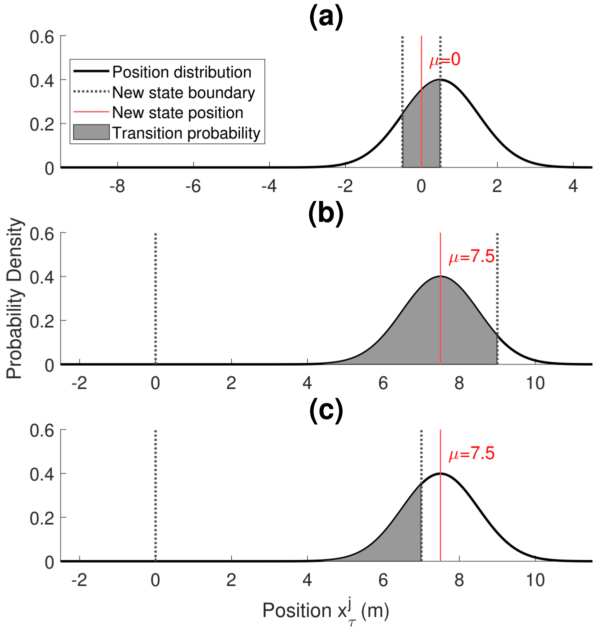

Fig. 4. Estimation of the probability of a transition between two states. **(a)** shows the case for a *node* state. **(b)** and **(c)** show the case for a *link* state.

4) Implementation: The locations $i_{\tau-1}$ and i_τ could either be nodes or links, which gives four different types of transition between states. The considerations in each case are described here. Figure 4, shows the computation of probability of transition to a new discrete location given the measured distance travelled, as well as the subsequent new continuous robot position estimated within that discrete location.

- **Node to Node:** If the node states are modelled as having infinitesimal size, the probability that a given measured uncertain distance moves the robot from one node state to another is zero. Therefore, the node states are considered to have a nonzero size, which gives a nonzero interval in distance corresponding to the node. This is illustrated in Figure 4(a). The position x_τ is set to zero.
- **Node to Link:** The interval in distance travelled is given by the distance to the near and far end of the link. This is illustrated in Figure 4(b) and (c). The most likely position x_τ is found using equations 15 and 16. The probability distribution of position x_τ over the space of the location $\boldsymbol{\xi}_\tau$ is created by limiting a Gaussian distribution in the range $\left(0, X^{L,i}\right)$, so is nongaussian, especially when the Gaussian distribution mode is close to the boundaries of the location $\boldsymbol{\xi}_\tau$. Related research addresses this by using a sampling approach when the estimate is close to this nonlinearity, and approximating the sample distribution as a Gaussian distribution (Brubaker et al., 2016). Here, the aim is in accurate trajectory estimation rather than estimation of the state at time τ. Therefore, this problem is addressed by modelling the distribution as a Gaussian distribution for which much of the probability mass may be outside of the location $\boldsymbol{\xi}_\tau$. This gives an imprecise distribution for time τ but allows the exact computation of subsequent distributions without the need for sampling or approximation.
- **Link to Node:** If the previous state $\mathbf{S}_{\tau-1}$ is in a link, it has a continuous position $x_{\tau-1}$. In this case, the normally distributed model can consider the uncertainty in both the position $x_{\tau-1}$ and in the distance travelled Δx_τ. The standard deviation of the distance travelled is equal to the square root of the sum of the squared standard deviation of the distributions for position and for distance travelled. This is seen in equation 19.
- **Link to Link:** The likelihood of moving from a given link $i_{\tau-1}$ to a given link i_τ can be found by combining the considerations used above.

A general consideration is made regarding efficiency and allow feasible computation. A threshold for probability is set below which probabilities are assumed to be zero, and therefore sufficiently unlikely estimates $\mathcal{S}^j_\tau$ can be removed from the set $\mathcal{S}_\tau$, and sufficiently unlikely paths $\boldsymbol{\xi}^j_{\tau-1:\tau}$ between states do not need to be considered. The latter is achieved by exploiting the constraints of the discrete network environment, and computing the possible paths Ξ between discrete states for a given map $\mathcal{M}$ using a depth-first search prior to robot operation. From each starting state, possible paths can be computed incrementally, and the ending state at each step is recorded. One factor of the probability of a transition between two states is independent of the odometry measurements $\mathbf{u}_\tau$, described by equation 23. This factor can therefore be compared to a threshold probability, and the finite set of possible paths Ξ is made of only paths with a sufficient probability.

D. Continuous Localization Implementation

As stated in equation 2, the aim is the *maximum a posteriori* estimate

$$\hat{\mathbf{s}}_{0:T} = \underset{\mathbf{s}_{0:T}}{\arg\max} \left\{ p(\mathbf{s}_{0:T}|\mathbf{u}_{1:T}, \mathbf{z}_{1:T}, \mathbf{s}_0) \right\} \tag{31}$$

where the state at time t is $\mathbf{s}_t = (\boldsymbol{\xi}_t = (i_t, d_t), \mathbf{x}_t = x_t)$.

This posterior distribution can be factorised as

$$\begin{aligned} &p(\mathbf{s}_{t=0:T}|\mathbf{u}_{1:T}, \mathbf{z}_{1:T}, \mathbf{s}_0) \\ &\quad = p(\boldsymbol{\xi}_{t=0:T}, \mathbf{x}_{t=0:T}|\mathbf{u}_{1:T}, \mathbf{z}_{1:T}, \mathbf{s}_0) \\ &\quad = p(\mathbf{x}_{t=0:T}, \boldsymbol{\xi}_{t=0:T}|\boldsymbol{\xi}_{\tau=0:T}, \mathbf{u}_{1:T}, \mathbf{z}_{1:T}, \mathbf{s}_0) \\ &\qquad\qquad p(\boldsymbol{\xi}_{\tau=0:T}|\mathbf{u}_{1:T}, \mathbf{z}_{1:T}, \mathbf{s}_0) \end{aligned} \tag{32}$$

by conditioning the posterior on the discrete set of states at times τ. These values $\boldsymbol{\xi}_{\tau=0:T}$ are what has been estimated precisely by the Viterbi algorithm when performing the *maximum a posteriori* estimation

$$\hat{\mathbf{S}}_{0:T} = \underset{\mathbf{S}_{0:T}}{\arg\max} \left\{ P(\mathbf{S}_{0:T}|\mathbf{u}_{1:T}, \mathbf{z}_{1:T}, \mathbf{S}_0) \right\} \tag{33}$$

in the discrete space of the estimates $\mathcal{S}_{0:T}$ at reduced time indices τ, as described in Section III-C.

Given the estimates of the discrete trajectory $\boldsymbol{\xi}_{\tau=0:T}$, the state space reduces to a single continuous linear space $x^{\boldsymbol{\xi}}$ along the trajectory. In this continuous linear space the *maximum a posteriori* estimates $\hat{\mathbf{x}}_{t=0:T}$ and $\hat{\boldsymbol{\xi}}_{t=0:T}$ can be found, therefore completing the *maximum a posteriori* estimate $\hat{\mathbf{s}}_{0:T}$.

When the estimated location is $\hat{\imath}_t \in \mathcal{N}$, the continuous position $\mathbf{x}_t = x_t$ is defined as equal to zero, so further estimation is not needed. When the estimated location is $\hat{\imath}_t \in \mathcal{L}$, the continuous position $\mathbf{x}_t$ can be estimated using the odometry measurements $\mathbf{u}_t$ and $\mathbf{u}_{t+1}$. This smoothing can be done for sequences in time from t_a to t_b where $\hat{\imath}_t \in \mathcal{L}$ for $t_a \leq t \leq t_b$, and where $\hat{\imath}_{t_a-1} \in \mathcal{N}$ and $\hat{\imath}_{t_b+1} \in \mathcal{N}$. For these sequences, the continuous space of $\hat{\imath}_{t_a:t_b}$ can be considered as a single linear continuous space, over which continuous probability distributions for position $\mathbf{x}_t$ can be simply described by one-dimensional linear Gaussian distributions. Therefore, *Rauch-Tung-Striebel* (RTS) smoothing can be used for this estimation.

The forward recursion of the smoothing is done using the method for computing hypothetical states described in Section III-C2, except using measurements $\mathbf{u}_t$ at time t rather than $\mathbf{u}_\tau$. Equations 15 and 16 calculate the estimate of the continuous position x_t using an odometry measurement and an estimate of the state at the previous time step. Equation 30 calculates the uncertainty in this estimate of continuous position. The uncertainty in estimates $\sigma_{x_{t_a}}$ and $\sigma_{x_{t_b}}$ is set to a small nonzero value.

The backward recursion of the smoothing is done starting at $t = t_b - 1$ using

$$\hat{x}_{t|t_b} = x_t + c_t(\hat{x}_{t+1|t_b} - x_{t+1}) \tag{34}$$

$$\hat{\sigma}_{x_{t|t_b}} = \sigma_{x_t} + {c_t}^2(\hat{\sigma}_{x_{t+1|t_b}} - \sigma_{x_{t+1}}) \tag{35}$$

where

$$c_t = \sigma_{x_{t|t}} \sigma^{-1}_{x_{t+1|t}} \tag{36}$$

and $\hat{x}_{t|t_b}$ and $\hat{\sigma}_{x_{t|t_b}}$ refer to the estimate and uncertainty for time t given information from time t_a to time t_b.

The estimates $\hat{x}_{t_a|t_b}$ to $\hat{x}_{t_b|t_b}$ in the single coordinate frame $x^{\boldsymbol{\xi}}$ can be used to find the corresponding discrete locations $\hat{\boldsymbol{\xi}}_{t_a|t_b}$ to $\hat{\boldsymbol{\xi}}_{t_b|t_b}$. This is simple to do as it is known how the continuous space $x^{\boldsymbol{\xi}}$ maps to the discrete locations along the trajectory. The estimates $\hat{x}_{t_a|t_b}$ to $\hat{x}_{t_b|t_b}$ are then transformed to the coordinate frames of the locations $\hat{\boldsymbol{\xi}}_{t_a|t_b}$ to $\hat{\boldsymbol{\xi}}_{t_b|t_b}$ for the output. This change of coordinate frames is done with an altered form of equation 16. This is straightforward to implement algorithmically, but the detail is omitted here.

This Rauch-Tung-Striebel smoothing estimate of the continuous variables $\hat{\mathbf{x}}_t$, combined with the Viterbi algorithm estimate of the discrete states $\hat{\boldsymbol{\xi}}_{0:T}$, gives the estimation of the most likely state trajectory $\hat{\mathbf{s}}_{0:T}$ for time indices t.

E. Additional Measurements

While this algorithm has been derived to use inputs from integrated linear and angular odometry in $\mathbf{u}_\tau$ and observations of either a node or a link in $\mathbf{z}_\tau$, other information could be found either instantaneously or through integration over the time step τ which could be incorporated into $\mathbf{z}_\tau$.

The first source of additional information used here is the identification of a nearby node which could be done by use of a *beacon* (Sahli and El-Sheimy, 2016; Haug et al., 2021) which the robot can detect and uniquely identify. This is implemented in the algorithm presented here as a simple multiplication of the probability of each state $\mathbf{S}_\tau$ by a factor given by $P(o_\tau|\mathbf{S}_\tau)$.

The second source of additional information is in loop-closures, where the robot detects and recognises a previously observed feature, thereby observing that the state at two different time steps is likely the same. This is denoted as $z^l_\tau = \tau_l$. If one estimate of the state is $\mathbf{S}_\tau$, and a loop-closure has been made between time τ and time τ_l, then the state $\mathbf{S}_{\tau_l}$ is found using the same method that is used to find the full sequence from time 0 to T. The likelihood of the estimated sequence given the loop-closure can be found by computing $P(z^l_\tau|\mathbf{S}_{\tau_l:\tau})$.

The third source of additional information is from the estimation of the length of a link Λ_τ using means such as acoustic echoes (Worley et al., 2020; Worley et al., 2024). This estimate gives a probability density over the continuous space of possible pipe lengths. This density is typically largest around the true pipe length (which is measured with probability p_Λ), but also around $N^+_{\bar{\Lambda}}$ multiples of the pipe length and around $N^-_{\bar{\Lambda}}$ random values smaller than the true pipe length. The true pipe length may not be in the set of measurements, with probability $\bar{p}_\Lambda$. The precision of this estimate depends on that of the acoustic echo detection, so the measured length has normally distributed uncertainty σ_Λ. This is simply implemented in the algorithm presented here as a multiplication of the probability of each estimate $\mathbf{S}_\tau$ by a factor given by $P(\Lambda_\tau|\mathbf{S}_\tau)$.

The fourth source of additional information is found from the three-dimensional nature of buried pipe networks. Pipe networks are constructed to follow the three-dimensional topology of the area to some extent, and some networks are designed to use gravity to direct fluid as desired. Therefore, there is a variation in elevation of nodes in pipe networks, and therefore a variation in gradient of pipes. The pipe gradient g_τ can be measured using an inertial sensor (Yu et al., 2018; Luo et al., 2021; Cui and Liang, 2023), or *inclinometer*, which measures the direction of gravity relative to the robot. Here, it is assumed that the robot will only sometimes be able to confidently measure the gradient of a pipe, due to variation in gradient along a pipe, variation in the robot's orientation with respect to the pipe, and sensor uncertainty σ_g. This uncertainty is reported as less than $1.5°$ (Cui and Liang, 2023), less than $0.5°$ (Yu et al., 2018), and around $0.1°$ (Hu et al., 2018), for different devices. Some of these values are relatively large; in the pipe data used in this paper, 35% of pipes have a gradient of less than $0.5°$, while 66% of pipes have a gradient of less than $1.5°$. Different values of measurement uncertainty σ_g will be investigated, although it is assumed that the robot will be able to correctly measure the sign of the gradient.

When incorporating all of these measurements, $\mathbf{z}_\tau$ would be defined as

$$\mathbf{z}_\tau = \left\{ z_\tau \in \left\{ L, N \right\}, o_\tau, z^l_\tau, \Lambda_\tau, g_\tau \right\} \tag{37}$$

IV. Results and Discussion

This section presents an experiment and discussion primarily comparing three algorithms:

1) The *2D* algorithm presented in the literature (Alejo et al., 2019).

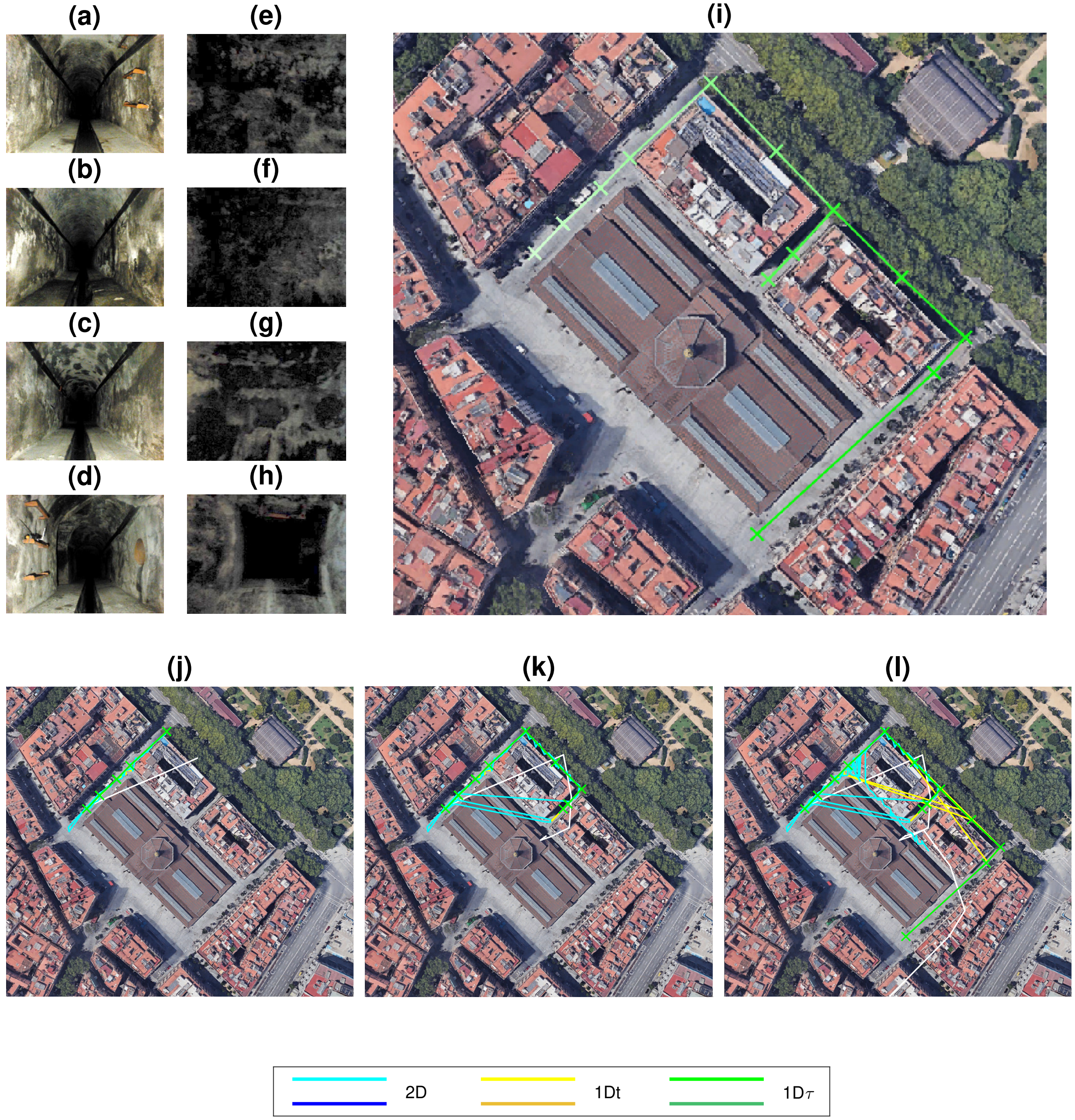

Fig. 5. **(a)**-**(h)**: Example data from the SIAR dataset, showing images from the front and upward facing cameras. **(i)** Illustrative example results showing the estimated trajectory based on experimental data from the SIAR dataset (Alejo et al., 2020). **(j)**-**(l)**: The trajectory estimated using the *1Dt*, *1Dτ*, and *2D* algorithms, at three different points in time. For comparison, an estimate made only using the odometry input is shown in white. The trajectory estimated using the *1Dτ* algorithm follows the true trajectory, so the true trajectory is not seen. *Satellite imagery: Google, ©2023.*

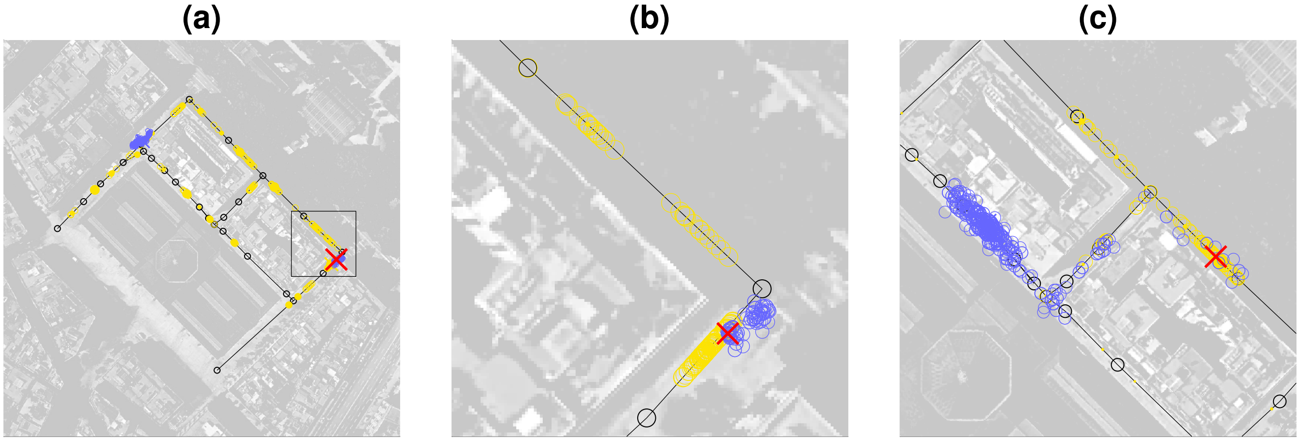

Fig. 6. Illustrative results showing the computation of the *1Dt* (yellow) and *2D* (blue) particle filter algorithms, using data from the SIAR dataset with added uncertainty. In all cases, circles show the location of *particles* used to represent the probability distribution over robot position, and the approximate true robot position is shown with a red cross. **(a)**: The full particle distributions, shortly after the robot has turned approximately 90 degrees at a corner. **(b)**: A close view of part of the two particle distributions, at the same point in time as in **(a)**. **(c)**: Part of the particle distributions some time after the robot has made several consecutive changes in direction. *Satellite imagery: Google, ©2023.*

2) The *1Dt* algorithm presented in previous work (Worley and Anderson, 2021a) (where it is there referred to as the *1D+* algorithm. The name of this algorithm has been changed in this paper to better describe the difference between it and the *1Dτ* algorithm.), which is an improvement in the application to high uncertainty robots to the *2D* algorithm.
3) The proposed *1Dτ* algorithm presented in this paper.

A. Localization on Real-World Experimental Data

Figure 5 shows a demonstration and comparison of localization algorithms using data from the SIAR dataset (Alejo et al., 2020) as an input. This data is recorded on a robot moving through a sewer network beneath part of a city in Spain. The robot records data using a range of sensors, and here, the output from the odometry and *manhole detection* functions are used as inputs to the localization algorithms, along with a map of the pipe network. Figures 5(a-d) show example images from the forward-facing camera on the robot, showing a relatively feature-dense environment compared to those described in other pipes in the literature (Edwards et al., 2023). To test the algorithms in the context of high-uncertainty measurements, only the odometry estimated using wheel encoders and and IMU is used, rather than visual odometry, to get $\mathbf{u}_t$ as in equation 5. Figures 5(e-h) show images from the upward-facing camera used in the dataset as part of a *manhole detection* system. The output from this manhole detection is used to get $\mathbf{z}_t$.

Due to differences between the dataset environment and robot operation to that specified in Section II, the data is altered somewhat. It was assumed here that a robot would only turn at discrete connections between pipes, but in the experimental environment there are sections of curved tunnel so the angular motion is gradual. To reconcile the data with the problem specified here, periods where the robot is turning are manually identified and the angular motion is integrated and applied at a single time step t. The position of the manholes in space and in time are adjusted to reflect this change. The experimental robot is able to move in either direction, while it was assumed here that the robot would only move forwards and change direction at manholes. The odometry data is updated to add this change of direction as a turn of 180 degrees.

Figure 5(i) shows the trajectory estimated using the *1Dτ* algorithm, along with an aerial image of the location above the buried pipes. The crosses show the positions of the detected manholes. Figures 5(j-l) show the trajectories estimated using the *1Dτ* and *1Dt* algorithms, and the *2D* algorithm presented in the literature (Alejo et al., 2019), incrementally through time.

Figure 6 illustrates the computation of the *1Dt* and *2D* algorithms. These algorithms are based on the particle filter, and the different particle distributions for each algorithm are shown. Figure 6(a) shows an occasion where the two distributions are usefully modelling the robot position. Much of the probability distribution is near to the true robot location. The increased robustness of the *1Dt* algorithm is achieved by creating a very diverse distribution; particles are spread across the environment, increasing the likelihood that the algorithm is making an estimate near the true robot location. Conversely, the *2D* particle distribution is clustered around only two locations. Figure 6(b) shows a closer view of part of the particle distribution. The computational drawback of the *1Dt* algorithm is illustrated, as it can be seen that each cluster of particles in the distribution could likely be well represented by a single estimate with some parameterised uncertainty. Figure 6(c) shows an occasion where the *2D* is failing to estimate the correct location. Due to uncertainty in the input to the algorithm, most of the probability distribution is clustered

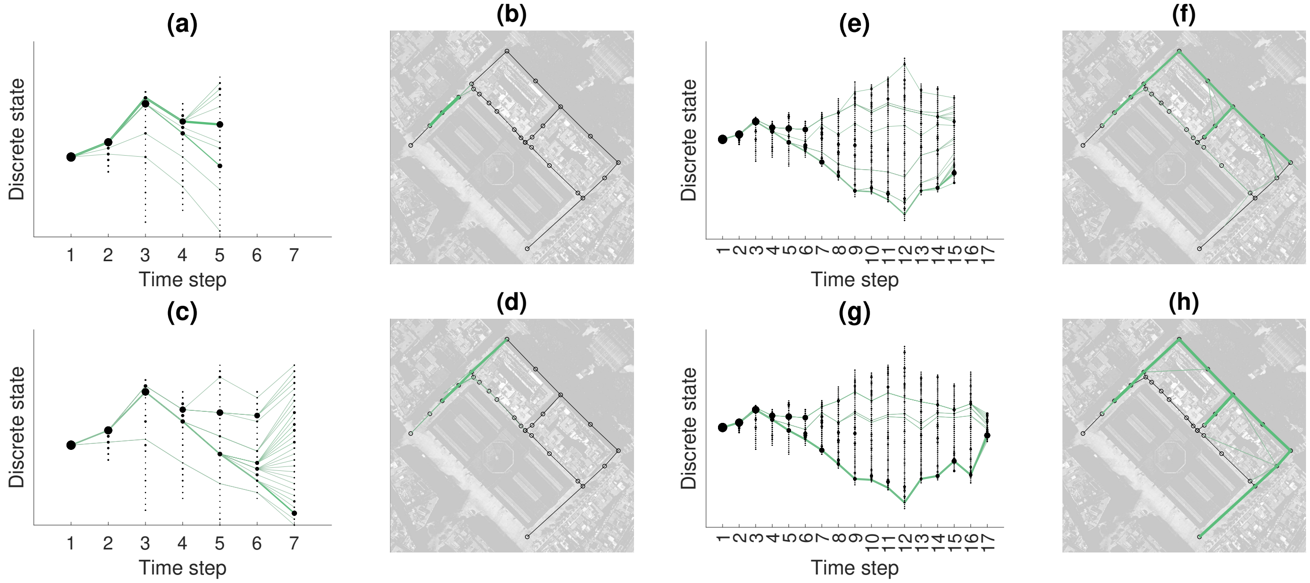

Fig. 7. Illustrative results showing the computation of the *1Dτ* algorithm using data from the SIAR dataset. **(a)**, **(c)**, **(e)**, and **(g)** show the discrete state space constructed by the algorithm for times $T = \{5, 7, 15, 17\}$. Each discrete state k at time $\tau = \{0, \ldots, T\}$ is denoted as a circle, the size of which corresponds to the likelihood $Q(\mathbf{S}^k_{0:\tau})$. Green lines show the likely trajectories through this state space at time T, where the width of the lines corresponds to the likelihood. **(b)**, **(d)**, **(f)**, and **(h)** show the likely trajectories estimated by the *1Dτ* algorithm drawn on the map. *Satellite imagery: Google, ©2023.*

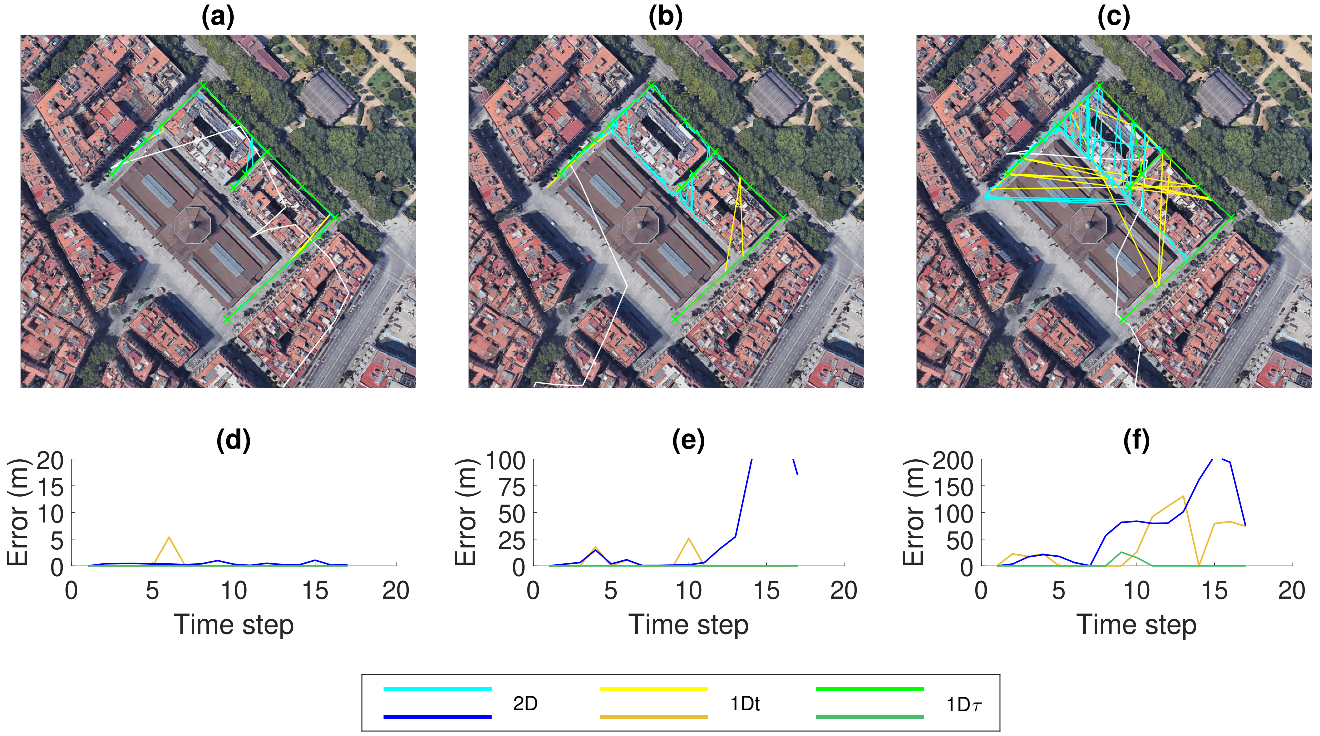

Fig. 8. **(a)**-**(c)**: Illustrative example results using experimental data from the SIAR dataset with increasing additive uncertainty in the odometry input: $\sigma_{\Delta x_t} = 0.2$, 0.5, and 0.8 for **(a)**, **(b)**, and **(c)** respectively. Four different estimate algorithms are compared, the *1Dt* (yellow), *1Dτ* (green), and *2D* (blue) algorithms, and an estimate made only using the odometry input (white). The true robot trajectory is shown with a dashed white line, slightly offset from the real position for clarity. **(d)**-**(f)**: The estimate error corresponding to each trajectory, at time steps τ. Note the variation in y-axis scale. *Satellite imagery: Google, ©2023.*

around the wrong location. There are some particles near the correct location, so the algorithm may be able to recover from this error; however, this would require a fortunate sequence of observations from the robot, which is unlikely in this environment with many similar appearing locations.

Figure 7 illustrates the computation of the *1Dτ* algorithm. Figures 7(a), (c), (e), and (g) show the discrete state space, in the same way as in Figure 3. The likelihood $Q(\mathbf{S}_{0:\tau}^k)$ corresponding to each state $\mathbf{S}_{\tau}^k$ is shown by a circle, and the likely trajectories through the state space are indicated by lines. These lines are transformed into the physical state space and shown on the maps in Figures 7(b), (d), (f), and (h). In all cases, the estimated trajectory approximately matches the real trajectory.

Comparing Figure 7(a), and (c), corresponding to $\tau = 5$ and $\tau = 7$, it is seen how the most likely trajectory at $\tau = 5$ is found to no longer be most likely at $\tau = 7$ given new information. This shows the innate recovery from error achieved by this algorithm. Comparing Figure 7(e), and (g), corresponding to $\tau = 15$ and $\tau = 17$, it is seen seen how the observation of a long pipe at $\tau = 17$ increases the certainty of the estimation of the trajectory, and reduces the number of states in the state space on which computation must be spent at future time steps. This dynamic computation of likely states based on the varying usefulness of the input measurements provides efficiency to the algorithm.

Figure 8 shows a comparison of the trajectories estimated by each algorithm, measured by the estimate error, for three different amounts of noise added to the input data. Figures 8(a) and (d) show the case with no additional noise, where all algorithms are seen to give a small amount of error. Figures 8(c) and (f) show the case with the largest additional noise, where only the *1Dτ* algorithm is seen to give a performance with low error.

The *1Dτ* algorithm produces a smooth trajectory estimate, while the trajectory estimate from the *1Dt* and *2D* algorithms is incoherent, exhibiting large, incorrect, jumps around the network. These jumps become more common as input uncertainty increases, while the robust *1Dτ* estimate remains coherent and low in error.

TABLE I

DEFAULT PARAMETERS FOR THE ROBOT MOTION AND MEASUREMENT.

Parameter	Symbol	Value
Command input motion	Δx (normal)	5
Normal motion noise	$\sigma_{\Delta x_t}$	0.2
Angular measurement noise	$\sigma_{\Delta\theta_t}$	0.1
Uniform motion noise	$u_{\Delta x_t}$	0.5 m
Motion noise constant	k_v	0.8
False positive rate	β_p	0.005
False negative rate	β_n	0.05

TABLE II

DEFAULT PARAMETERS FOR THE LOCALIZATION ALGORITHMS

	Parameter	Symbol	Value
2D	Motion model noise	σ_{ψ}	$1.2\sigma_{\Delta x_t}$
	Angular motion model noise	σ_{ω}	$1.2\sigma_{\Delta\theta_t}$
	Angular motion model noise	$\sigma_{\omega,min}$	0.1 rad
	Link measurement std.	σ_l	Δx m
	Node measurement std.	σ_n	Δx m
1Dt	Motion model noise	$\tilde{\sigma}_{\psi}$	$1.2\sigma_{\Delta x_t}$
	Node transition std.	σ_n	Δx m
	Angular weight std.	$\tilde{\sigma}_{\omega}$	$10\sigma_{\Delta\theta_t}$
	Kernel std.	σ_g	$5\Delta x$ m
	Number of Particles	M	100
	Number of New Particles	M_D	10
	Number of Particles Constant	K_{h0}	0.05
	Divergence Short Time Constant	$\tau_{\alpha s}$	20
	Divergence Long Time Constant	$\tau_{\alpha l}$	100
	Hypothesis Weight Threshold	α^0	0.01
1Dτ	Motion model noise	$\tilde{\sigma}_{\Delta x_\tau}$	$4\sigma_{\Delta x_\tau}\sqrt{T_{\tau-1}^{\tau}}$
	Angular motion model noise	$\tilde{\sigma}_{\Delta\theta_\tau}$	$2\sigma_{\Delta\theta_\tau}$
	Model noise minimum	$\tilde{\sigma}_{\Delta\theta,min}$	0.2

B. Localization on Simulated Data Analysing Robustness to Noise

An example of the performance of the localization algorithms is shown for simulated large-scale data in Figure 9. In Figure 9(a), the trajectory estimated by each algorithm is shown on a map of part of the pipe network used in this experiment. This network is obtained from a utility map of a town in the UK. The trajectory estimate error is compared for each algorithm in Figure 9(b). The estimate from the *2D* algorithm eventually becomes incorrect and fails to relocalize, indicated by a persitantly large error. The *1Dt* algorithm successfully relocalizes over the whole trajectory, indicated by periods of large error followed by periods of low error. The *1Dτ* algorithm has a consistently accurate estimate.

The algorithms are compared by using simulation to create a large number of trajectories through a pipe network, each of 1000 time steps in length. As in previous work (Worley and Anderson, 2021b), the algorithms are compared in terms of *error rate*, which is the proportion of time where the error in estimated position is above a threshold 25 metres. Figure 9(c) illustrates the calculation of the estimate error rate. The coloured sections of the graphs indicate time steps at which the estimate is in an incorrect discrete location. This error is computed only at time steps τ, to make a fair comparison between the algorithms.

The error rate is compared for a range of values of different variables describing the uncertainty in the input measurements:

1) Normally distributed uncertainty in measured linear motion, with additive noise given by $v_t \sim \mathcal{N}(\mu = 0, \sigma = \sigma_{\Delta x_t}\Delta x_t)$.
2) Uncertainty in measured angular motion, with additive noise given by $w_t \sim \mathcal{N}(\mu = 0, \sigma = \sigma_{\Delta\theta_t}\Delta\theta_t)$.
3) Integrated uniformly distributed uncertainty in measured

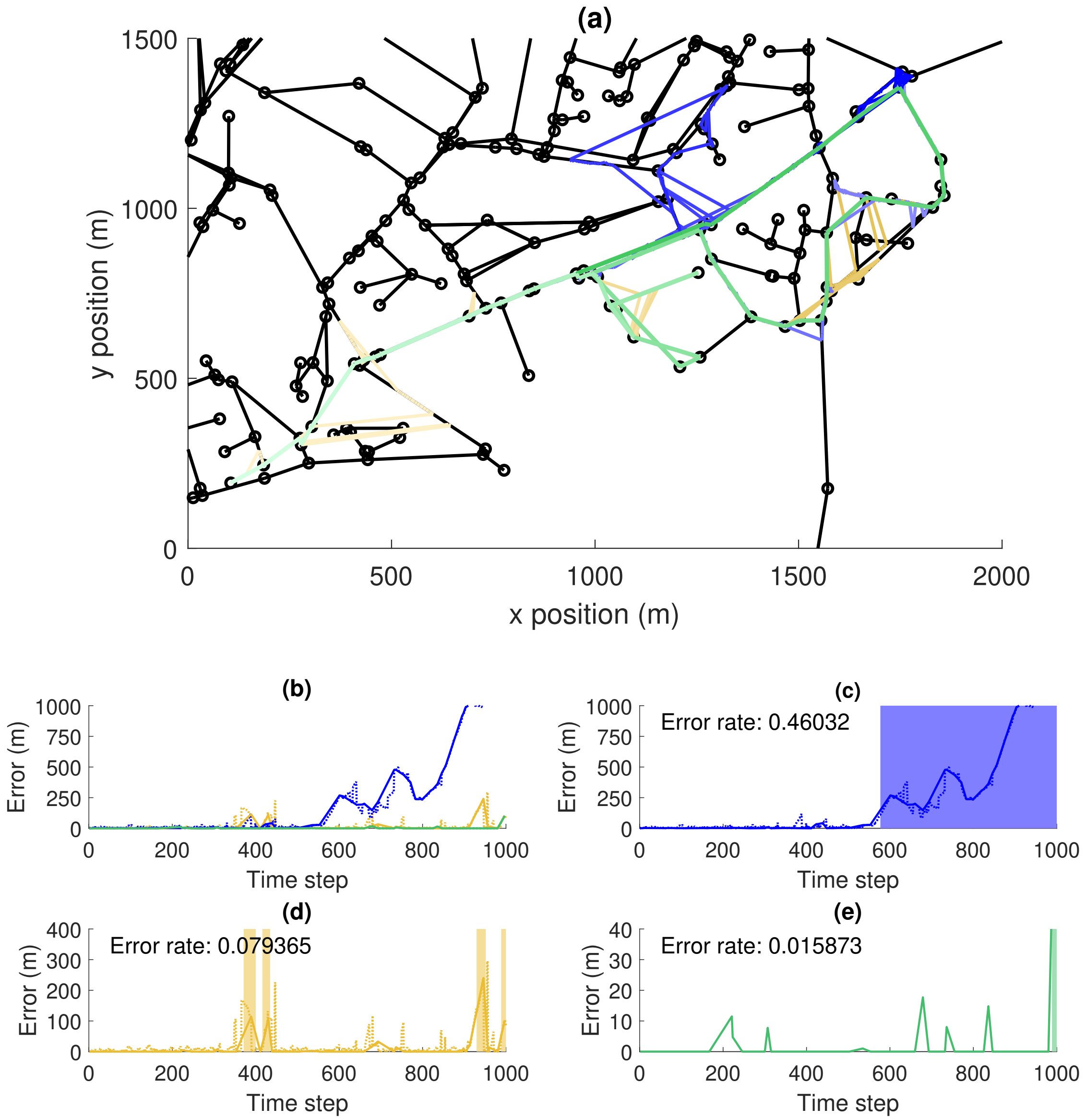

Fig. 9. **(a)**: Illustrative example results from simulation showing an estimated trajectory. Three different estimate algorithms are compared, the *1Dt* (yellow), *1D*τ (green), and *2D* (blue) algorithms. The trajectory estimated using the *1D*τ algorithm follows the true trajectory, so the true trajectory is not seen. The trajectories over time are indicated by the colour gradient from light to dark. **(b)**: The estimate error for the trajectories in **(a)**. **(c)**-**(e)**: Illustrations of the calculation of the *error rate* from the estimate error in **(b)**, which is the proportion of time for which the error is above a threshold, indicated by the coloured sections. The dotted line shows the error at time steps t, while the solid line shows the error at time steps τ for which the error rate is compared here. *Satellite imagery: Google, ©2014.*

linear motion, with additive noise given by $v_t = k_v v_{t-1} + (1 - k_v)\tilde{v}_t$, where $\tilde{v}_t \sim \mathcal{U}(-u_{\Delta x_t}, u_{\Delta x_t})$. The added uncertainty is therefore dynamic, and could model persistent deviations in the robot's motion caused by obstacles or the flow of water in the pipes. This is challenging to the algorithms which use a model of uncorrelated uncertainty.

4) Uncertainty in the observation of nodes, given by the rates of false positive and false negative detection, β_p and β_n, described by equation 6. Both are varied simultaneously, according to $\beta_p = \beta/10$, $\beta_n = \beta$, with the observation error rate β.

The error rate and computation time are measured over different values of these uncertainty variables. For each value, the error rate and computation time are measured over 50 trajectories through the pipe network, and the distributions of error rate and computation time over the 50 trajectories are presented. The median of each distribution is shown as a measure of the average value unbiased by the high outliers. The 90$^{\text{th}}$ percentile is shown as a measure of the measure of the higher, or worse, values in each distribution.

The default parameters describing uncertainty in the simu-

TABLE III

COMPARISON OF ALGORITHMS OVER ALL TRAJECTORIES.

		Algorithm A		
		2D	1Dt	1Dτ
Alg. B	2D	-	55%	76%
	1Dt	36%	-	79%
	1Dτ	7%	6%	-

Percentage of trajectories where the error rate of Alg. A $<$ error rate of Alg. B

lated robot's motion and measurements are given in Table I. The parameters used by each of the localization algorithms are given in Table II. The meaning of the parameters for the *1Dt* algorithm are described in detail in previous work (Worley and Anderson, 2021a), as is the meaning of the parameters for the *2D* algorithm (Worley and Anderson, 2021b). The parameter values for the *1Dt* and *2D* algorithms, for example number of particles, are chosen based on the results from previous work. It was found that increasing the number of particles would reduce the error rate somewhat; however, there is no guarantee that the error rate would continue to decrease for very large numbers of particles. Moreover, this reduction was be smaller for each further increase in number of particles, therefore giving a reduction in efficiency. 100 particles was chosen as a balance between performance and computational cost.

Table III compares the results over all measured trajectories. For 55% of trajectories, the *1Dt* algorithm error rate is less than that of the *2D* algorithm, while for 36% of trajectories, the *1Dt* algorithm error rate is more than that of the *2D* algorithm, indicating the better performance of the *1Dt* algorithm. For 79% of trajectories, the *1Dτ* algorithm error rate is less than that of the *1Dt* algorithm, while for only 6% of trajectories, the *1Dτ* algorithm error rate is more than that of the *1Dt* algorithm, which illustrates the further improved performance of the *1Dτ* algorithm.

Figure 10(a-e) shows the results of the comparison for each value of uncertainty variables. In most cases, the error rate for the *1Dτ* algorithm increases less with the increase in uncertainty than that for the *1Dt* algorithm, which in turn increases less than that for the *2D* algorithm. In some cases, for the lower values of uncertainty variables, the error rate of the *2D* algorithm is lower than that of the *1Dt* algorithm, indicating a suitability for estimation in the case of low uncertainty.

Figures 10(a) and (b) show that the *1Dτ* algorithm has a median error rate of less than 0.025, and a 90th percentile error rate of less than 0.15, at the largest measured value of linear uncertainty, where the *1Dt* algorithm's median error rate is above 0.4 and the *2D* algorithm's median error rate is above 0.6. The literature gives a range of values of error in linear motion estimation for robots in pipes; with measurements of 15% (Sahli and El-Sheimy, 2016), 5-35% (Murtra and Mirats Tur, 2013), and 3-15% (Al-Masri et al., 2018) of distance travelled. Figure 10(a) shows that the median error rate is 0 and 0.02 for linear motion uncertainty of 50% and 100% of distance travelled respectively. Therefore, the algorithm performs well at uncertainty comparable to that expected in practice, and is robust to the larger uncertainty that might be expected in application in more challenging realistic pipe environments.

Figure 10(c) shows that the *1Dτ* algorithm has a median error rate of 0, and a 90th percentile error rate of 0, even at the largest measured value of angular motion, where the median error rate of the *1Dt* algorithm is around 0.4 and the median error rate of the *2D* algorithm is above 0.6.

Figure 10(d) shows that as measurement error rate increases up to a measurement error rate of 0.3, the estimate error rate remains low for the *1Dτ* algorithm, while it increases steadily for the *1Dt* algorithm. At a measurement error rate of 0.3, the error rate for all three algorithms increases considerably.

The results presented here can be translated to some degree to the further challenge of *simultaneous localization and mapping* (SLAM) in this application, where an exact map of the pipe network is not necessarily available. The effect of metric error in the pipe network map, such as incorrect pipe lengths or angles at junctions, would be similar to that of metric error in the sensing. Therefore, similar relative magnitudes of linear and angular uncertainty in the network map could produce similar results to those in Figure 10(a)-(c). One element of topological error in the map would be a missing junction. The effect of this may be similar to that of false positive junction detection as implemented in this experiment. Further topological error, such as entire missing pipes, would require an augmentation to the approach. Hybrid metric-topological approaches to SLAM have been proposed generally (Blanco et al., 2008) and in application to pipe networks (Lee et al., 2013), although the accuracy of the resulting map given the large uncertainties in sensor input in this application is not known. In a case where a map is available but has some missing information, it may be possible to use the robust and efficient approach proposed in this paper most of the time, and use a more computationally complex approach when it is estimated that the map is likely missing some information. Overall, addressing the SLAM problem for unknown environments would be an interesting area for future work.

Figure 10(e) compares the performance of the algorithms over an increase in all uncertainty parameters simultaneously. A similar trend is seen as in the other comparisons, as the error rate of the *1Dt* algorithm and the *2D* algorithm increase to a median value close to 1 with the increase in uncertainty, while the error rate of the *1Dτ* algorithm stays low and only increases to a median value of less than 0.1. However, the 90th percentile error rate increases to above 0.4 for the *1Dτ* algorithm. Therefore, while it is robust to an increase in each source of uncertainty individually, its robustness is still limited when using only the simple measurements used here as inputs. This motivates the incorporation of additional information from observations of the environment, as described in III-E.

Overall, the proposed algorithm can be compared to the existing localization methods for robots in pipe networks, which are the *2D* (Alejo et al., 2019) and *1Dt* (Worley and

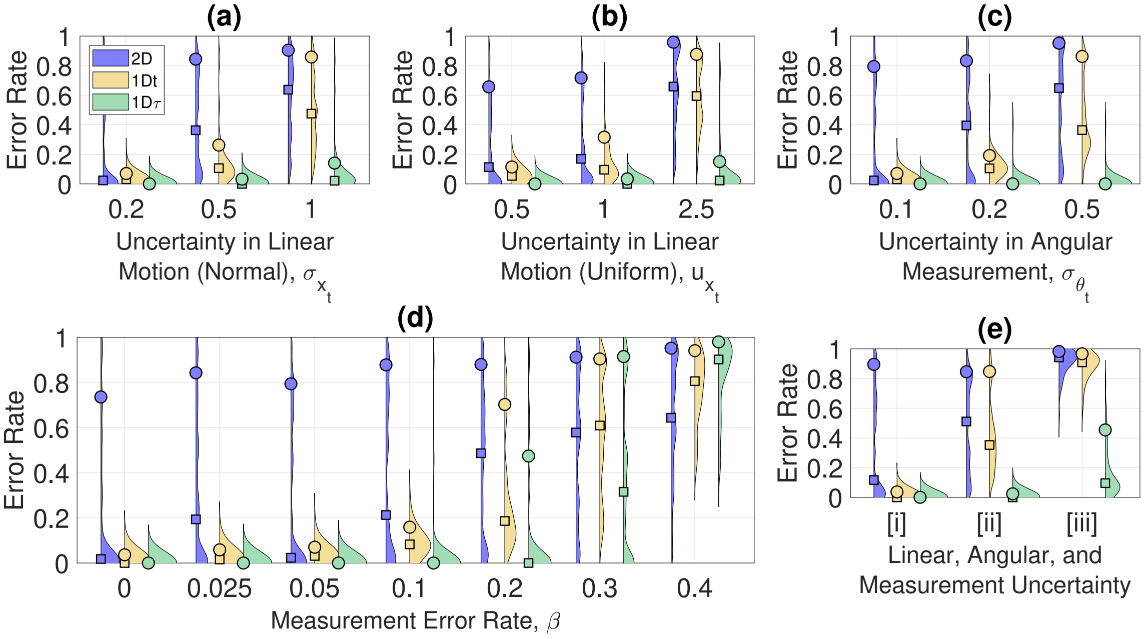

Fig. 10. A comparison of the error rate of the *2D*, *1Dt*, and *1D*τ algorithms with variation in: **(a)** σ_{x_t}. **(b)** u_{x_t}. **(c)** σ_{θ_t}. **(d)** β. **(e)** `[i]`: $\sigma_{\Delta x_t} = 0.2$, $\sigma_{\Delta\theta_t} = 0.1$, $\beta = 0$, `[ii]`: $\sigma_{\Delta x_t} = 0.5$, $\sigma_{\Delta\theta_t} = 0.2$, $\beta = 0.05$, `[iii]`: $\sigma_{\Delta x_t} = 1$, $\sigma_{\Delta\theta_t} = 0.5$, $\beta = 0.1$. The probability density of error rate and computation time over 50 sample trajectories is estimated using *kernel density estimation* with a Gaussian kernel *bandwidth* of 0.05 error rate. Squares show the median and circles show the 90th percentile of the distributions.

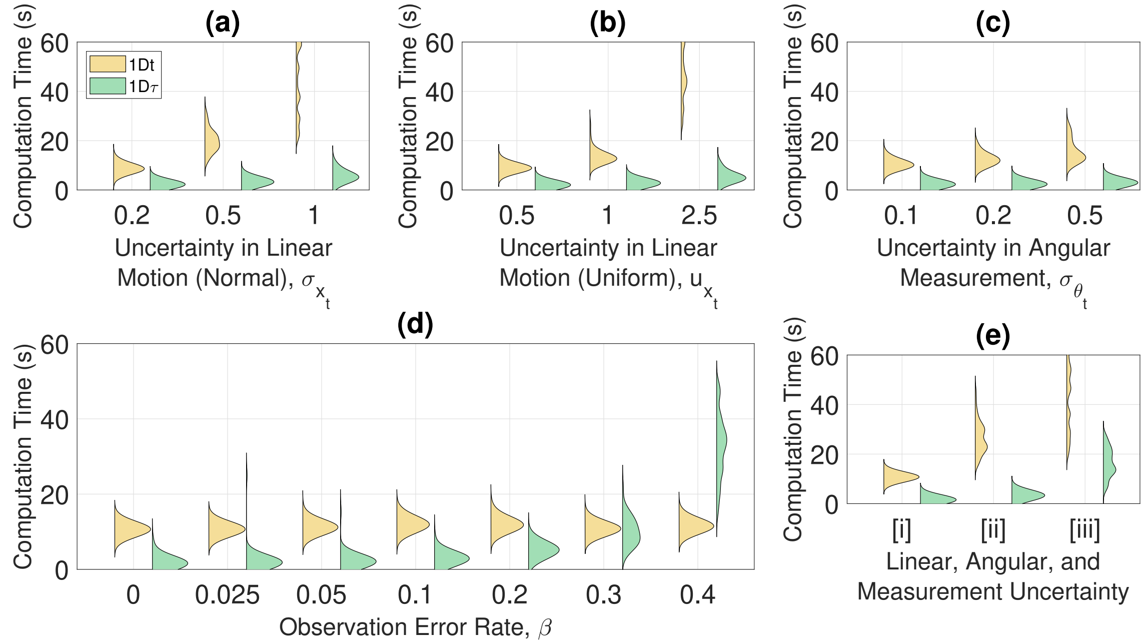

Fig. 11. A comparison of the computation time of the *1Dt* and *1D*τ algorithms with variation in: **(a)** σ_{x_t}. **(b)** u_{x_t}. **(c)** σ_{θ_t}. **(d)** β. **(e)** `[i]`: $\sigma_{\Delta x_t} = 0.2$, $\sigma_{\Delta\theta_t} = 0.1$, $\beta = 0$, `[ii]`: $\sigma_{\Delta x_t} = 0.5$, $\sigma_{\Delta\theta_t} = 0.2$, $\beta = 0.05$, `[iii]`: $\sigma_{\Delta x_t} = 1$, $\sigma_{\Delta\theta_t} = 0.5$, $\beta = 0.1$. The probability density of computation time over 50 sample trajectories is estimated using *kernel density estimation* with a Gaussian kernel *bandwidth* of 2 seconds.

Anderson, 2021a) algorithms, as well as similar approaches based on the particle filter (Kazeminasab et al., 2021). The proposed *1Dτ* algorithm is more accurate, and specifically more robust, as it maintains a low estimate error in cases of uncertainty which cause the alternative algorithms to fail. Other developments in the literature are complementary to the algorithm proposed here. Sensing to improve the precision of position estimation along a single pipe, such as the use of vision (Hansen et al., 2015; Zhang et al., 2023), acoustic echoes (Worley et al., 2020; Worley et al., 2024), or radio wave sensing (Rizzo et al., 2021), are well suited to function alongside the approach for robust localization across a large pipe network proposed in this paper.

These results can be translated to pipe environments with higher complexity. The hybrid metric-topological space is embedded in a three-dimensional space, and while the pipe networks investigated here are relatively flat, the approach could be applied equally to networks of pipes in industrial applications where the network extends considerably in all three dimensions. The approach proposed here explicitly represents the connectivity of pipes. Therefore, even if many pipes exist close together, this approach could be useful in estimating which pipes the robot has traversed through, if the connectivity between them is known. Evaluation of the algorithm in more complex environments, for example with overlapping pipes such as would be found in process plants, would be an interesting area for future work.

C. Analysis of Computational Cost of Localization Algorithms

Figure 11(a-e) shows a comparison of the computation time used by the *1Dt* and *1Dτ* algorithms over a variation in uncertainty parameters. Table IV presents statistics on the computation time over all cases, excluding outliers. The median computation time over all cases tested here is 10.98 seconds for the *1Dt* algorithm, and 2.52 seconds for the *1Dτ* algorithm. The computational time of the *1Dτ* algorithm is a median of 0.18 times than that of the *1Dt* algorithm.

There is a correlation between computation time and error rate as shown in Figure 10, seen in Figure 11(a), (b), (d), and (e), and in Figure 11(c) to a lesser extent. Lower uncertainty inputs result in fewer likely possible states which simultaneously reduces the chance of error and reduces cost.

Overall, considering both the results for error rate in Section IV-B and computation time, it can be seen that the *1Dτ* algorithm is substantially more efficient than the *1Dt* algorithm for most tested values of uncertainty, with a lower error rate and lower computational cost. The proposed *1Dτ* algorithm therefore has a considerable advantage in applicability to robots in real challenging pipe networks, compared to particle filter approaches, such as those in the literature (Alejo et al., 2019; Kazeminasab et al., 2021).

Further improvement to the computation time might be found by using a more efficient adaptive approach to estimation, rather than the *breadth-first* approach taken here. Inspiration could be taken from the field of robot path planning, for example, where the state space can be efficiently searched for the most promising region for improvements to the trajectory (Li et al., 2023).

TABLE IV

Computation Time Statistics Excluding Outliers

Statistic	*1Dt*	*1Dτ*	*1Dτ / 1Dt*
Lower Quartile	10.33	1.91	0.14
Median	10.98	2.52	0.18
Upper Quartile	12.21	3.58	0.24
Mean	11.38	2.95	0.19
Standard Deviation	2.06	1.46	0.079

D. Additional Sensing for Localization

This section investigates the incorporation of four additional sources of information proposed in Section III-E: unique identification of a node, loop-closure, estimation of the length of a link, and estimation of the gradient of a link, into the *1Dτ* algorithm. This is compared to a estimate using only simple inputs with the higher uncertainty values used previously which give a median error rate of around 0.3 and a 90th percentile error rate of around 0.8 . Figure 12 shows the results of these comparisons. Here, the estimate error is only measured at times at which the robot observes correctly, through $\mathbf{z}_\tau$, that it is at a node. Otherwise, there will be a component of the error rate which corresponds to the error in position estimate along a link, which this algorithm and additional sensing cannot address, and therefore the effect of additional sensing would not be properly measured.

The frequency of measurements is varied to measure the improvement to the error rate for increasing cost from additional sensing. The *identification probability* is the proportion of nodes in the network which are chosen to be identifiable. This measurement could be obtained by placing identifiable beacons placed around the network, so increase in *identification probability* might increase the cost of hardware. However, these identifiable beacons might be placed alongside other necessary hardware such as recharging stations, depending on the overall system design, with only a small increase in cost. The *loop-closure probability* determines the probability that the robot is able to recognise a location that it has previously visited. The increase in probability therefore corresponds to an increase in the required perceptive ability of the robot, and adds a requirement to the path the robot takes. The *echo measurement quality* determines the accuracy and precision of the measurement of pipe length made using acoustic echoes, with parameters $\bar{p}_\Lambda$, $N^+_{\bar{\Lambda}}$, $N^-_{\bar{\Lambda}}$, and σ_Λ defined in Section III-E. The *gradient measurement uncertainty* σ_g is the standard deviation in gradient measurement, where $\sigma_g = 10^{-2}$ corresponds to around 0.5°, $\sigma_g = 10^{-3}$ corresponds to around 0.05°, and $\sigma_g = 10^{-5}$ corresponds to around 0.0005°.

Figure 12(a) shows that the error rate decreases with increasing proportion of nodes being identified. The median and 90th percentile error rate decreases most substantially with the increase of identification probability from 0 to 0.25,

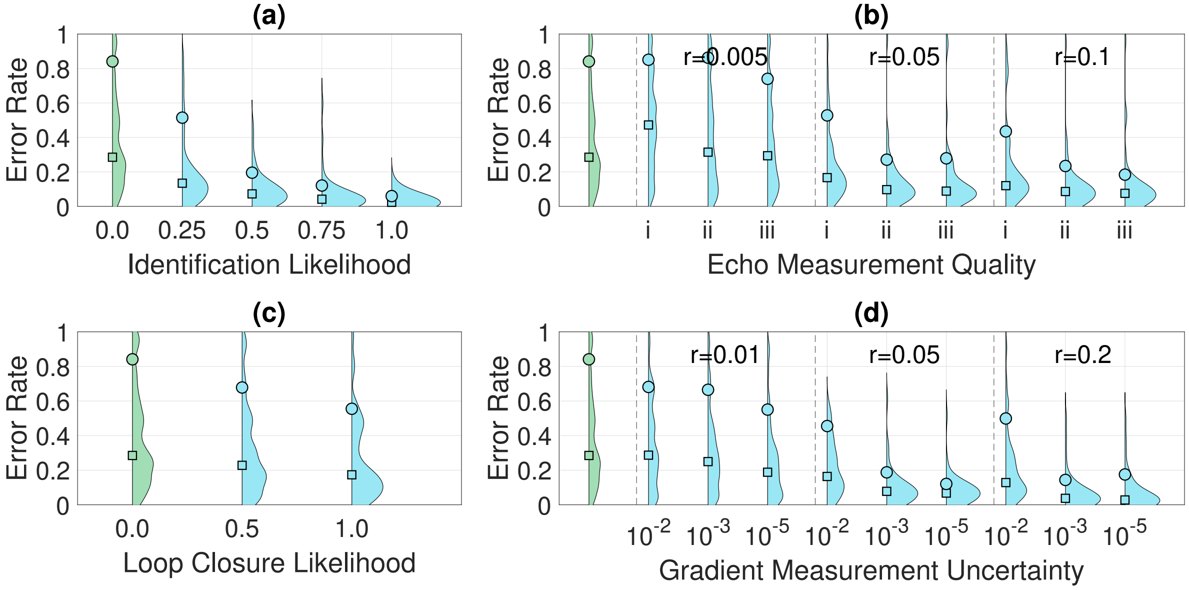

Fig. 12. The effect of additional information on localization. **(a)** Unique identification of nodes. **(b)** Estimates of link length using acoustic echoes. The quality is defined by $\left[\bar{p}_\Lambda, N^+_{\bar{\Lambda}}, N^-_{\bar{\Lambda}}, \sigma_\Lambda\right]$. For the lowest quality, i, these are $[0.1, 1, 5, 5]$. For ii, these are $[0.01, 0, 1, 2]$. For the highest quality, iii, these are $[0, 0, 0, 1]$. r is the proportion of times t at which the robot makes an echo measurement. **(c)** Loop-closure measurements. **(d)** Measurements of link gradient, with variation in measurement uncertainty standard deviation σ_g. r is the proportion of times t at which the robot makes a gradient measurement. The probability density of error rate over 50 sample trajectories is estimated using *kernel density estimation* with a Gaussian kernel *bandwidth* of 0.05. Squares show the median and circles show the 90th percentile of the distributions.

and decreases less so for further increases in identification probability. For an identification probability of 1, the median error rate is 0.03. Further experimentation would be needed to better understand this relationship, and could also explore the effect of a deterministic rather than stochastic placement of identifiable nodes. The nonzero error rate for an identification probability of 1 may be due to some estimates being made in links for which the estimate of continuous position could be greater than the threshold for computing the error rate. It may also be because the identification measurement can only improve the estimate if the relevant node was predicted by the simple odometry and node observation measurements.

Figure 12(b) shows the effect of increasing echo measurement quality and rate on the estimate. The error rate generally decreases when echo measurements are made regularly, but can actually decrease when low quality measurements are made only at a low rate. In the best case, with a higher rate of high quality measurements, the median error rate is reduced from 0.28 to 0.075, and the 90th percentile error rate is reduced from 0.84 to 0.18. Similar to the identification observations, acquiring the sequence of acoustic echo measurements needed to make an estimate of pipe length has some cost. The relative cost between making acoustic measurements and placing beacons in the network for identification is beyond the scope of the work presented here. Unlike the identification observations, acoustic echo measurements can be made by the robot at any time. An active localization approach where the robot makes an acoustic measurement when the uncertainty in its localization estimate is large could more efficiently acquire useful measurements when they are more informative.

Figure 12(c) shows the effect of increasing probability of loop-closures. The error rate is seen to decrease only slightly, from a median of 0.28 to a median of 0.17, and the 90th percentile decreases from 0.83 to 0.55, even for a probability of 1 of making loop-closure observations. This could be explained by the low probability that a given loop-closure observation occurs at a time when it improves the estimate. In order to correct an otherwise erroneous estimate at time τ, a loop-closure must occur between the current time and a previous time for which the estimate is correct. In comparison, a useful *identification* observation requires only requires the coincidence in timing of one observation rather than the coincidence in timing of two observations.

Figure 12(d) shows the effect of increasing rate and quality of gradient measurements. The error rate generally decreases when gradient measurements are used, and the improvement to accuracy is larger for the lower uncertainty gradient measurements. If measurements are made frequently enough, for measurement uncertainty of 10^{-2}, around the worst of the values from the literature, the error rate can be reduced somewhat, from a median of 0.28 to a median of 0.13. For measurement uncertainty of 10^{-3}, which is close to the best value from the literature, the error rate can be reduced substantially, to a median of 0.037. The error rate is only reduced by a small additional amount for further reduced measurement uncertainty, to a median of 0.028, so useful improvement to localization estimates could be found with sensing technology similar to what has been reported on in the existing literature. Incorporating this three-dimensional gradient information is therefore useful, and may be even more impactful in pipe networks in locations with a bigger variation in topography.

These results can be compared to alternative approaches to localization in pipe networks. The use of beacons for identifying a robot (or sensor) location has been investigated in the literature (Haug et al., 2021; Kazeminasab and Banks, 2021), and these results are complementary to those presented here, as it has been shown that incorporating the identification of a robots location can improve the localization estimate. However, it has been shown that alternative sensing, specifically acoustic echo sensing and pipe gradient sensing, can be used to produce a similar improvement to localization estimate, without the need for the additional cost of adding beacons to a huge number of locations around the pipe network. These alternative sensing approaches might therefore offer an advantage in terms of cost effectiveness.

V. Conclusions

This paper develops an approach for estimating a robot trajectory through a network of buried pipes, aiming for robustness to the large uncertainty in input measurements which is expected in field applications, while achieving low computational cost which is necessary in application to small robots. Experiments here have compared the novel *1D*τ algorithm which improves on the Viterbi algorithm to estimate the most likely sequence of states given the sequence of measurements, with the *1Dt* algorithm which uses a particle filter to incrementally estimate the most likely state with additional processes to detect and recover from error.

The *1D*τ algorithm is shown to have higher accuracy and robustness than the *1Dt* algorithm for almost all measured input uncertainty values, and is shown to require an average of 0.18 times the computation. For linear motion uncertainty, the *1D*τ algorithm is shown to have a low median error rate of less than 0.025 for uncertainty up to 100% of the distance travelled, larger than that expected in practical application from the literature. The median error rate is 0 for motion uncertainty up to 50% of the distance travelled and for a observation error rate of 0.2, and the 90% percentile error rate is 0 for the largest measured value of angular motion uncertainty. A median error rate of 0 means that the trajectories are estimated without error more often than not, and these results give some insight into the requirements for front-end sensing needed to achieve this performance. These results show that the principle of making the localization estimate in a suitable low-dimensional space is effective in this application.

The *1D*τ algorithm is challenged by larger values of uncertainty for motion and measurement simultaneously, which motivated the incorporation of additional sensing information. In this case, the error rate was reduced from 0.28 to 0.023 by incorporating the ability to *identify* locations in the network, and similarly reduced by incorporating *echo* sensing proposed in the literature and incorporating pipe gradient measurement, while *loop-closure* measurements were shown to be less useful.

Developments to the *1D*τ algorithm could further improve the accuracy, robustness, and computational cost. The effect on the estimate of the pipe network topology and geometry could be considered, as could the impact of active sensing or active localization. It may be possible to estimate an upper bound on the performance of an algorithm, providing evidence beyond numerical and statistical analysis of the accuracy and robustness of the algorithm.

Acknowledgements

This work is supported by EPSRC grant EP/S016813/1 Pervasive Sensing for Buried Pipes (Pipebots).

References

Aguiar, A. S., Santos, F. N. d., Santos, L. C., Sousa, A. J., and Boaventura-Cunha, J. (2023). Topological map-based approach for localization and mapping memory optimization. *Journal of Field Robotics*, 40(3):447–466.

Aitken, J. M., Evans, M. H., Worley, R., Edwards, S., Zhang, R., Dodd, T., Mihaylova, L., and Anderson, S. R. (2021). Simultaneous Localization and Mapping for Inspection Robots in Water and Sewer Pipe Networks: A Review. *IEEE Access*, 9:140173–140198.

Akai, N. (2023). Reliable monte carlo localization for mobile robots. *Journal of Field Robotics*, 40(3):595–613.

Al-Masri, W. M., Abdel-Hafez, M. F., and Jaradat, M. A. (2018). Inertial Navigation System of Pipeline Inspection Gauge. *IEEE Transactions on Control Systems Technology*, PP:1–8.

Alejo, D., Caballero, F., and Merino, L. (2019). A robust localization system for inspection robots in sewer networks. *Sensors (Switzerland)*, 19(22):1–28.

Alejo, D., Chataigner, F., Serrano, D., Merino, L., and Caballero, F. (2020). Into the dirt: Datasets of sewer networks with aerial and ground platforms. *Journal of Field Robotics*, 38(1)(July).

Bernuy, F. and Ruiz-del Solar, J. (2018). Topological semantic mapping and localization in urban road scenarios. *Journal of Intelligent and Robotic Systems: Theory and Applications*, 92(1):19–32.

Blanco, J.-l., Fernandez-Madrigal, J.-a., and Gonzalez, J. (2008). Toward a unified Bayesian approach to hybrid metric – topological SLAM. *IEEE TRANSACTIONS ON ROBOTICS, VOL. 24*, 24(2):259–270.

Brubaker, M. A., Geiger, A., and Urtasun, R. (2016). Map-based probabilistic visual self-localization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(4):652–665.

Cui, S. and Liang, J. (2023). Three-dimensional mapping of pipelines using laser ranging and a gyroscope. *Scientific Reports*, 13(1):20330.

Department for International Trade (2015). Water and treated water.

Department for Science, Innovation and Technology (2022). National Underground Asset Register (NUAR).

Doherty, K. J., Lu, Z., Singh, K., and Leonard, J. J. (2022). Discrete-Continuous Smoothing and Mapping. *IEEE Robotics and Automation Letters*, 7(4):12395–12402.

Edwards, S., Zhang, R., Worley, R., Mihaylova, L., Aitken, J., and Anderson, S. R. (2023). A robust method for approximate visual robot localization in feature-sparse sewer pipes. *Frontiers in Robotics and AI*, 10(March):1–13.

Forney, G. (1973). The viterbi algorithm. *Proc. IEEE*, 61:268–278.

Godsill, S., Doucet, A., and West, M. (2001). Maximum a posteriori sequence estimation using monte carlo particle filters. *Annals of the Institute of Statistical Mathematics 53, 82–96*.

Hansen, P., Alismail, H., Rander, P., and Browning, B. (2015). Visual mapping for natural gas pipe inspection. *International Journal of Robotics Research*, 34(4-5):532–538.

Haug, M., Lorenz, F., and Thamsen, L. (2021). GRAL: Localization of Floating Wireless Sensors in Pipe Networks. (September):251–257.

Hu, C., Li, M., Zeng, D., and Guo, S. (2018). A survey on sensor placement for contamination detection in water distribution systems. *Wireless Networks*, 24(2):647–661.

Huang, W. H. and Beevers, K. R. (2005). Topological map merging. *International Journal of Robotics Research*, 24(8):601–613.

Kakogawa, A., Komurasaki, Y., and Ma, S. (2017). Anisotropic shadow-based operation assistant for a pipeline-inspection robot using a single illuminator and camera. *IEEE International Conference on Intelligent Robots and Systems*, 2017-Septe:1305–1310.

Kakogawa, A., Yamagami, T., Tian, Y., and Ma, S. (2015). Recognition of pathway directions based on nonlinear least squares method. *2015 IEEE International Conference on Robotics and Biomimetics, IEEE-ROBIO 2015*, pages 1596–1601.

Kazeminasab, S. and Banks, M. K. (2021). A Localization and Navigation Method for an In-pipe Robot in Water Distribution System through Wireless Control towards Long-Distance Inspection. *IEEE Access*, PP:1.

Kazeminasab, S., Janfaza, V., Razavi, M., and Banks, M. K. (2021). Smart Navigation for an In-pipe Robot Through Multi-phase Motion Control and Particle Filtering Method. *ArXiv*, pages 342–349.

Khan, M. W., Das, G. P., Hanheide, M., and Cielniak, G. (2020). Incorporating Spatial Constraints into a Bayesian Tracking Framework for Improved Localisation in Agricultural Environments. *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2440–2445.

Kirchner, F. and Hertzberg, J. (1997). A prototype study of an autonomous robot platform for sewerage system maintenance. *Autonomous Robots*, 4(4):319–331.

Lee, D. H., Moon, H., Koo, J. C., and Choi, H. R. (2013). Map building method for urban gas pipelines based on landmark detection. *International Journal of Control, Automation and Systems*, 11(1):127–135.

Li, C., Meng, F., Ma, H., Wang, J., and Meng, M. Q.-H. (2023). Relevant region sampling strategy with adaptive heuristic for asymptotically optimal path planning. *Biomimetic Intelligence and Robotics*, 3(3):100113.

Luo, S., Li, W., Lv, F., Cai, D., Kong, L., et al. (2021). Research and evaluation of the measurement uncertainty with the pipeline robot. 1894(1):012036.

Merriaux, P., Dupuis, Y., Vasseur, P., and Savatier, X. (2015). Fast and robust vehicle positioning on graph-based representation of drivable maps. *Proceedings - IEEE International Conference on Robotics and Automation*, 2015-June(June):2787–2793.

Murtra, A. C. and Mirats Tur, J. M. (2013). IMU and cable encoder data fusion for in-pipe mobile robot localization. *IEEE Conference on Technologies for Practical Robot Applications, TePRA*.

Polvara, R., Del Duchetto, F., Neumann, G., and Hanheide, M. (2021). Navigate-and-Seek: A Robotics Framework for People Localization in Agricultural Environments. *IEEE Robotics and Automation Letters*, 6(4):6577–6584.

Rizzo, C., Seco, T., Espelosín, J., Lera, F., and Villarroel, J. L. (2021). An alternative approach for robot localization inside pipes using RF spatial fadings. *Robotics and Autonomous Systems*, 136:103702.

Sahli, H. and El-Sheimy, N. (2016). A novel method to enhance pipeline trajectory determination using pipeline junctions. *Sensors (Switzerland)*, 16(4):1–17.

Sousa, R. B., Sobreira, H. M., and Moreira, A. P. (2023). A systematic literature review on long-term localization and mapping for mobile robots. *Journal of Field Robotics*, 40(5):1245–1322.

Thielemann, J. T., Breivik, G. M., and Berge, A. (2008). Pipeline landmark detection for autonomous robot navigation using time-of-flight imagery. *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops*.

Thrun, S., Burgard, W., and Fox, D. (2006). *Probabilistic Robotics*. The MIT Press.

Worley, R. and Anderson, S. (2021a). Robot Localization in a Pipe Network using a Particle Filter with Error Detection and Recovery in a Hybrid Metric-Topological Space. *2021 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, pages 1–8.

Worley, R. and Anderson, S. (2021b). Robust Efficient Localization of Robots in Pipe Networks using a Particle Filter for Hybrid Metric-Topological Space. *2021 European Conference on Mobile Robots (ECMR)*.

Worley, R., Yu, Y., and Anderson, S. (2020). Acoustic echolocalization for pipe inspection robots. *IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, pages 2–7.

Worley, R., Yu, Y., Horoshenkov, K. V., and Anderson, S. R. (2024). Acoustic echo sensing for robot localization in buried pipe networks. *IEEE Sensors Journal*, 24(16):26506–26521.

Wu, D., Liang, Z., and Chen, G. (2022). Deep learning for lidar-only and lidar-fusion 3d perception: a survey. *Intelligence & Robotics*, 2(2).

Yu, Z., Yameng, X., Xinjing, H., Jian, L., and Shili, C. (2018). Pipeline inclination measurements based on a spherical detector with magnetic proximity switches. *IEEE Access*, PP(August):1.

Zhang, R., Worley, R., Edwards, S., Aitken, J., Anderson, S. R., and Mihaylova, L. (2023). Visual simultaneous localization and mapping for sewer pipe networks leveraging cylindrical regularity. *IEEE Robotics and Automation Letters*, 8(6):3406–3413.

Zhao, W., Kamezaki, M., Yoshida, K., Konno, M., Onuki, A., and Sugano, S. (2018). Modeling and simulation of FLC-based navigation algorithm for small gas pipeline inspection robot. *IEEE/ASME International Conference on Advanced Intelligent Mechatronics, AIM*, 2018-July:912–917.