Robust Efficient Localization of Robots in Pipe Networks using a Particle Filter for Hybrid Metric-Topological Space

Rob Worley[1], Sean Anderson[1]

***Abstract*— Water distribution and drainage pipe inspection and maintenance is costly, and could be improved by using robots to locate faults from within the pipes. Robot localization is critical in this operation, but is challenging due to the constraints of the pipe environment. An efficient, robust algorithm is needed for localization using limited sensors. A novel particle filter algorithm is proposed for localization, which estimates the robot's position in a hybrid metric-topological state space, allowing efficient computation and relocalization. The algorithm is demonstrated in simulation at a large scale, considering substantial uncertainty in motion, measurements, and the map of the environment, showing an improvement over a benchmark algorithm developed for this application.**

I. Introduction

Buried pipe infrastructure such as water and drainage pipes needs constant inspection and maintenance, the efficiency of which might be improved by using robots to persistently monitor a pipe network. However, this environment is challenging for robots due to constraints on power and size. An important aspect of the robot operation is localization, which allows the robot to navigate autonomously, accurately locate faults in the network, and avoid becoming unrecoverable.

For long-term operation, a localization algorithm back-end should be robust to false positives and false negatives in measurements, and recover from outlier measurements and mislocalization [1]. This is especially important in pipes as the robot has limited front-end information, and an unrecoverable robot would be a critical failure in the operation.

Localization in pipes is challenging due to the unavailability of GPS, the lack of a reliable magnetic field for a magnetometer, the limited perspective of sensors, and the sparseness of recognisable features. Despite this, localization has been demonstrated using a range of sensors, including vision [2], acoustics [3], [4], [5], [6], and radio waves [7]. However, in practice, front-end sensor information is expected to be uncertain and unreliable. Metric measurements will inevitably have some error, which is exacerbated in this application as the quality of typical sensing is limited by constraints on the hardware and computing power. Measurements may also be infrequent, occluded, and result in false positives and negatives. However, the robot's environment is reasonably well mapped and the pipe network topology and some metric information is available, as in Figure 1.

*R. Worley is supported by an EPSRC Doctoral Training Partnership Scholarship, R. Worley and S. Anderson are supported by EPSRC grant EP/S016813/1 Pervasive Sensing for Buried Pipes (Pipebots)

[1]R. Worley and S. Anderson are with Department of Automatic Control and Systems Engineering, University of Sheffield, Sheffield, UK rfworley1@sheffield.ac.uk

978-1-6654-1213-1/21/$31.00 ©2021 IEEE

Fig. 1. An example water distribution network map from a town in the UK. The map is approximately 1.6 km from top to bottom.

The aim of this paper is to evaluate an efficient and robust robot localization algorithm specialized for pipe networks. As in related work [8], our approach assumes the use of only odometry and a binary measurement of detection of features such as corners and junctions, along with a map of the environment. The aim is to show that effective localization in a pipe network is possible even with this limited set of measurements. Additional challenges presented in this work are from larger uncertainty in measurements, false positive and negative measurements, and error in the environment map. The approach developed is therefore different from previous methods for pipe networks, recent demonstrations of localization methods for pipe networks [9], [10], and other recent developments in localization in similar environments such as mines [11] and tunnels [12], which show effective localization with a large range of sensors. Here the aim of localization is to provide a sufficient position estimate to allow navigation and reporting of faults in the pipes, rather than for precise mapping and control.

Hybrid metric-topological localization [13], as applied previously in network environments [14], would be useful in a pipe environment; as the pipe network is well described as a set of connected spaces, a hybrid metric-topological representation offers advantages to performance from the use of topology, but without less loss in precision from discretization. Inspired by this previous work, this paper describes the pipe network as a hybrid metric-topological map, with both discrete and continuous aspects. This idea has been applied to localization in road networks [15], [16], [17], and similarly in pipe applications this can be done without much abstraction, as the networks are well described by topological connections and metric distances alone.

The algorithm presented here uses a particle filter to estimate the robot position, as the non-parametric distribution works in the discontinuous network environment and can give a multi-modal estimate [18]. The hybrid metric-

topological representation reduces the dimension of the state space compared to a continuous metric representation. This gives a reduction in size of the space, which has been shown to improve the efficiency of particle filtering [19], [20]. The use of map topology facilitates global localization which improves robustness of the particle filter.

This paper contributes the development and evaluation of a robot localization algorithm with several key distinctions:

1) The particle filter in hybrid metric-topological space is applied where sensing is limited to only binary detection of features such as corners and junctions.
2) Robot motion is highly uncertain, including a time-varying bias in the error between motion and odometry representing fluid flow, gradient, and varying pipe conditions not found in previous work.
3) Robot sensing is prone to false positives and negatives, which have not been investigated in previous work.
4) Error in the knowledge of the environment is included, investigating the performance of the algorithm in the case that the buried network has been mapped poorly due to difficulties in observing the pipes.

II. Problem Definition

A. Environment Definition

Water and drainage networks are made up of pipes connected by corners, junctions, valves and manholes. These networks cover tens of kilometers in combined length. In this work, maps are derived from a water distribution pipe network map for a 100 square kilometer part of the UK.

This map is divided into towns and simplified somewhat: Valves, which are inaccurately represented as sections of pipe, are combined with adjacent pipes, and curved pipes, which are inaccurately represented as several short pipe segments, are replaced with straight pipes. In practice, valves might be undetected by a robot, or might be useful features which aid localization. Calculating the distance from a point to a curved pipe might pose some extra challenge to the *continuous space* algorithm described later, while the *hybrid space* algorithm could simply consider a curved pipe the same as any other one-dimensional pipe.

An environment therefore consists of around 1000 *nodes* (junctions and corners) connected by a similar number of *links* (pipes). Robot motion can be simulated in this environment, allowing experimentation over a scale and topology that is difficult to create experimentally.

B. State Definition

The robot's motion is evaluated at discrete time steps where the robot moves and makes a localization estimate. The robot's pose $\mathbf{x}_t$ at time t is defined in a hybrid one-dimensional-two-dimensional coordinate system as the triple

$$\mathbf{x}_t^h = \left(i_t, x_t, d_t\right) \tag{1}$$

where $i_t \in \mathcal{I}$ is the discrete index of the link or node the robot is in from the set of all indices $\mathcal{I} = \{\mathcal{L}, \mathcal{N}\}$ where $\mathcal{L}$ and $\mathcal{N}$ are the sets of all link and node indices. x_t is the distance from the origin of the link or node. d_t is the discrete direction of the robot in the link, where $d_t^l \in \left\{-1, 1\right\}$, or node, where $d_t^n \in \left\{1, 2, \ldots, D^n, -1, -2, \ldots, -D^n\right\}$ where there are D^n adjacent links at node n and where the negative and positive indices correspond to the robot arriving from or leaving from a direction respectively. In this case only the position of the robot along the axis of the pipe is considered, and the position of the robot in three dimensions with respect to this axis is not estimated. Similar definitions of robot state are found in the literature [16], [13].

The time steps t do not need to be at regular intervals, and instead a localization estimate is made either after a period of time or whenever new information is available. This could be after the robot has executed a turn or when it arrives at a node.

The pose in this coordinate system can be converted to a two-dimensional representation of pose in global coordinates

$$\mathbf{x}_t^g = \left[x_t^g, y_t^g, \theta_t^g\right]^T \tag{2}$$

using the map of the poses of each link and node.

C. Motion and Measurement Definition

Two models for linear robot motion are used in this work. In both models, the robot moves along link l according to

$$x_t^l = x_{t-1}^l + \Delta x_t d_t + v_t \tag{3}$$

where Δx_t is used as an odometry measurement for localization, and where v_t is the motion noise. In the first model, v_t is normally distributed noise with zero mean and covariance $\sigma_x \Delta x_t$, giving the uncertainty in motion typical in mobile robots. In the second model, v_t is given by

$$v_t = k_v v_{t-1} + (1 - k_v)\tilde{v}_t \tag{4}$$

where $\tilde{v}_t$ is uniformly distributed noise in the range $[-u_x, u_x]$. This models a more challenging noise distribution, acting like a slowly changing bias which could model fluid flow, gradient, or varying pipe surface conditions. Δx_t is usually equal to a fixed command input, except when the robot stops as it arrives at a node, where it will be smaller.

After arriving at a node, the robot turns according to $d_t^n \in D^n$, and correspondingly $\theta_t^n \in \Theta^n$ where Θ^n is the set of directions at node n. A measurement of change in angle $\Delta\theta_t$ is made for use in localization, given by

$$\Delta\theta_t = \theta_t^n - \theta_{t-1}^n + w_t \tag{5}$$

with normally distributed noise w_t with variance $\sigma_\theta \Delta\theta_t$. This models the relative angular measurement that could be made using a gyroscope in an IMU, without the use of an absolute measurement from an IMU compass which is expected to be unavailable in the application environment.

There is therefore a double of motion observations

$$\mathbf{u}_t = \left(\Delta x_t, \Delta\theta_t\right) \tag{6}$$

Finally, a binary measurement z_t is made, equal to L if the robot detects that it is in a link, and equal to N if it detects that it is at a node. This kind of measurement can be

made using visual [21], [22], [23], [24], [25], acoustic [26], [4], or inertial [27], [28] sensing, as has been demonstrated in previous work. This binary measurement contains no other information about the link or node it has detected, and represents the minimal measurement of this type that could be made. To model errors in this measurement, a false positive and false negative rate (β_p and β_n) can be specified. It is assumed that if the measurement is a false negative, then there is also no measurement of change in angle $\Delta\theta_t$, since this could otherwise be used to detect the node.

D. Metric Map Definition

For localization, the robot can use a map, $\mathcal{M}_{metric}$, of the environment. This map contains the positions of each node $\mathbf{x}_n^g = \begin{bmatrix} x_n, y_n \end{bmatrix}$ which represent junctions in the pipes, and the positions of each end of each link $\mathbf{x}_l^g = \begin{bmatrix} x_{l,1}, y_{l,1}, x_{l,2}, y_{l,2} \end{bmatrix}$ which represent the pipes connecting each node.

Measurements ζ_t can be made between an estimated robot pose $\hat{\mathbf{x}}_t^g = \begin{bmatrix} \hat{x}_t, \hat{y}_t, \hat{\theta}_t \end{bmatrix}$ and the map, given by

$$\zeta_t = \begin{cases} \min\limits_{n \in \mathcal{N}} f(\hat{\mathbf{x}}_t^g, \mathbf{x}_n^g), & \text{if } z_t = N \\ \min\limits_{n \in \mathcal{N}} g(\hat{\mathbf{x}}_t^g, \mathbf{x}_l^g), & \text{if } z_t = L \end{cases} \tag{7}$$

where the function f gives the distance between two points, finding this distance to a node, and the function g gives the distance between a point and a line of finite length, finding the distance to a link.

There is therefore a double of environment observations

$$\mathbf{z}_t = \begin{pmatrix} z_t, Z_t \end{pmatrix} \tag{8}$$

which can be used in localization, where Z_t is the set of measurements ζ_t for each estimated robot pose.

E. Topological Map Definition

An alternative to a *metric* map is a *hybrid metric-topological* map, $\mathcal{M}_{hybrid}$. The map is made up of two parts: a map of the *links*, and a map of the *nodes*. For each link, the map contains the link index i, the two adjacent node indices, the length of the link x_i^L, the position of the origin of the link, and the heading angle of the link. Every position in every link is therefore uniquely defined, with one end of each link defined as a relative position of $x_t = 0$, and the other defined as a relative position of $x_t = x_i^L$. For each node, the map contains the node index, the position of the node, the indices of each adjacent node, the indices of each adjacent link, and the direction to each adjacent node.

A measurement ξ_t can be made between an estimated robot pose $\hat{\mathbf{x}}_t^h = \begin{pmatrix} \hat{i}_t, \hat{x}_t, \hat{d}_t \end{pmatrix}$ and features in this map. The only measurement is the distance to the nearest node which is found when z_t equals N. This can be simply found using

$$\xi_t = \min \left\{ |\hat{x}_t|, |x_{\hat{i}}^L - \hat{x}_t| \right\} \tag{9}$$

There is therefore a double of environment observations

$$\mathbf{z}_t = \begin{pmatrix} z_t, \Xi_t \end{pmatrix} \tag{10}$$

which can be used in localization, where Ξ_t is the set of measurements ξ_t for each estimated robot pose.

III. Localization Algorithm in Continuous Space

A localization algorithm similar to what has been previously reported [8] is described here. The state of the robot is estimated in the two-dimensional global coordinate frame $\mathbf{x}_t^g$ defined by equation 2, with the map $\mathcal{M}_{metric}$, and environment observations $\mathbf{z}_t$ defined by equation 8. In this section, the state is written as $\mathbf{x}_t = \mathbf{x}_t^g$ for simplicity.

As in typical robot localization, the posterior distribution $p(\mathbf{x}_t | \mathbf{u}_{0:t}, \mathbf{z}_{0:t})$ is desired, which is the probability distribution over possible states $\mathbf{x}_t$ given the sequence of observations $\mathbf{u}_{0:t}$ and $\mathbf{z}_{0:t}$. A particle filter is used to estimate the pose of the robot [18], which uses a set of M particles as random samples of the posterior distribution. The likelihood of a hypothetical state $\mathbf{x}_t^m$ for particle m being included in the particle set $\mathcal{X}_t$ is proportional to the posterior, given by

$$\mathbf{x}_t^m \sim b(\mathbf{x}_t) = p(\mathbf{x}_t | \mathbf{u}_{1:t}, \mathbf{z}_{1:t}) \tag{11}$$

The distribution $b(\mathbf{x}_t)$ is recursively estimated using

$$b(\mathbf{x}_t) \propto p(\mathbf{z}_t | \tilde{\mathbf{x}}_t) \int p(\tilde{\mathbf{x}}_t | \mathbf{u}_t, \mathbf{x}_{t-1}) b(\mathbf{x}_{t-1}) d\mathbf{x}_{t-1} \tag{12}$$

with four steps, as follows.

1) *Prediction*. The predicted state is sampled from

$$\tilde{\mathbf{x}}_t^m \sim p(\mathbf{x}_t | \mathbf{u}_t, \mathbf{x}_{t-1}^m) \tag{13}$$

$$\begin{bmatrix} \tilde{x}_t^m \\ \tilde{y}_t^m \\ \tilde{\theta}_t^m \end{bmatrix} = \begin{bmatrix} \tilde{x}_{t-1}^m \\ \tilde{y}_{t-1}^m \\ \tilde{\theta}_{t-1}^m \end{bmatrix} + \begin{bmatrix} (\Delta x + \psi_t) \cos(\theta_{t-1}^m) \\ (\Delta x + \psi_t) \sin(\theta_{t-1}^m) \\ (\Delta\theta + \omega_t) \end{bmatrix} \tag{14}$$

for each particle m, where ψ_t and ω_t are estimates of the normally distributed noise variables v_t and w_t, and are sampled from normal distributions with a mean of zero and a variance equal to $\sigma_\psi \Delta x_t$ and $\sigma_\omega \Delta\theta_t + \sigma_{\omega,min}$ respectively. This gives a set of particles $\tilde{\mathcal{X}}_t$.

2) *Weighting*. Each particle is then weighted according to

$$\alpha_t^m \propto p(\mathbf{z}_t | \tilde{\mathbf{x}}_t^m) \tag{15}$$

$$\alpha_t^m = \begin{cases} (\sigma_l \sqrt{2\pi})^{-1} e^{-\zeta_t^2 / \sigma_l^2}, & \text{if } z_t = N \\ (\sigma_n \sqrt{2\pi})^{-1} e^{-\zeta_t^2 / \sigma_n^2}, & \text{if } z_t = L \end{cases} \tag{16}$$

where ζ_t is defined in equation 7, and σ_l and σ_n are parameters which control how strongly the algorithm weights particles that are closer to the elements in the map.

Rather than searching all links and nodes to find the nearest to each particle separately, the efficiency is improved by using *k-means* clustering to find the centroid of any distinct parts of the particle distribution, and a subset of the links or nodes near to these centroids is searched to find the value for ζ_t as in equation 7. The number of clusters is chosen arbitrarily here, $k = 10$, and is simply sufficiently high so that all particles are likely to be near to a cluster centroid.

3) *Resampling*. M particles are sampled from the existing set using *sequential importance* resampling. Particles are drawn from $\tilde{\mathcal{X}}_t$ creating $\mathcal{X}_t$, which is distributed approximately according to equation 11, using

$$p(\mathbf{x}_t^m \in \mathcal{X}_t) \propto \alpha_t^m \tag{17}$$

4) *Estimation*. The mode of the distribution is used as an estimate of robot pose. In this case, the mode is estimated by an approximation of the median particle in two dimensions, which is the particle with the smallest total distance to all other particles. Clustering could also be used to do this more efficiently, but would require the optimal number of clusters k.

IV. Robust Localization in Hybrid Metric-Topological Space

A novel algorithm is described in this section which gives a more efficient, robust estimate. The previously reported particle filter in continuous space functions well. The particle representation of the posterior distribution is useful in the discontinuous network environment, where parametric representations such as a Kalman filter might fail. The algorithm is effective even with the reduced sensing compared to the original application with a powerful robot [8].

However, the algorithm has some weaknesses in the application to smaller robots. Firstly, a large number of particles is required, with a large computational cost prohibitive to a robot with limited power. Secondly, the algorithm is seen to have low robustness with the limited sensing in this problem definition. Despite the general ability of the particle filter to model an arbitrary distribution where particles could be spread in any configuration across the state space, with this implementation the particles tend to be distributed in a single region, observed anecdotally. Therefore, if mislocalization occurs it is unlikely that the algorithm can relocalize.

The state of the robot is estimated in the hybrid metric-topological coordinate system as $\mathbf{x}_t = \mathbf{x}_t^h$ given by equation 1, using the map $\mathcal{M}_{hybrid}$. In this case, the posterior distribution $p(\mathbf{x}_t|\mathbf{u}_{0:t}, \mathbf{z}_{0:t}, \mathcal{M}_{hybrid})$ is desired, where $\mathbf{z}_t$ is in this case defined by equation 10, and a particle filter is used to recursively estimate this posterior using equations 11,12, 13, and 15. Each particle in the particle set $\mathcal{X}_t$ is modelled as having a Gaussian distributed uncertainty in state, resulting in an estimator which shares properties between a particle filter and a multi-hypothesis filter. The steps of the algorithm and the function of each aspect will be described here.

A. Improved Efficiency: Particle Prediction and Weighting

1) Prediction: The particles are constrained to the network, using the topological information contained in the map $\mathcal{M}_{hybrid}$. For each particle m in set $\mathcal{X}_{t-1}$, the predicted state is sampled as in equation 13 using

$$\tilde{x}_t^{l,m} = \tilde{x}_{t-1}^{l,m} + \Delta x_t \tilde{d}_{t-1}^{l,m} + \psi_t \tag{18}$$

where ψ_t is sampled from a normal distribution with a variance equal to $\sigma_\psi \Delta x_t$, approximating the noise variable v_t. If $\tilde{x}_t^{l,m}$ is greater than x_i^L or less than zero, the particle has passed a node at the corresponding end of the link. The particle moves to a new link by choosing an index $\tilde{i}_t$ with uniform probability from the links adjacent to the appropriate node, and updating $\tilde{d}_t$ and $\tilde{x}_t^{l,m}$ accordingly.

This prediction algorithm constrains each of the particles to be in the network, and the movement of particles to different adjacent links results in a multimodal distribution which should be more robust than a unimodal distribution.

When z_t indicates that a node has been detected, particles may be transitioned to a node index. The probability of a particle transitioning to the nearest node i^n is given by

$$p(i_t^m = i^n) = e^{-\xi_t^{m\,2}/2\sigma_n^2} \tag{19}$$

If the particle transitions to a node, or if a particle is in a node and the robot turns, the direction index $\tilde{d}_t^n$ is updated.

2) Weighting: The particle weights depend on z_t. In this algorithm, the weights are computed using both equation 15 and 13 in a form of mixture distribution sampling, using the available information in $\mathbf{x}_{t-1}^m$ and $\mathbf{u}_t$ where it is suitable as well as the typical information in $\tilde{\mathbf{x}}_t^m$ and $\mathbf{z}_t$.

For $z_t = N$ and $z_t = L$ respectively the weights are

$$\alpha_{t,N}^m = \begin{cases} 1-\beta_p, & \text{if } \tilde{i}_t^m \in \mathcal{N} \\ \beta_p + \epsilon_\alpha, & \text{if } \tilde{i}_t^m \in \mathcal{L} \end{cases} \tag{20}$$

$$\alpha_{t,L}^m = \begin{cases} 1-\beta_n, & \text{if } \tilde{i}_t^m \in \mathcal{L},\, i_{t-1}^m \in \mathcal{N} \\ 1-\beta_n, & \text{if } \tilde{i}_t^m = i_{t-1}^m,\, i_{t-1}^m \in \mathcal{L} \\ \beta_n + \epsilon_\alpha, & \text{if } \tilde{i}_t^m \neq i_{t-1}^m,\, i_{t-1}^m \in \mathcal{L} \end{cases} \tag{21}$$

where ϵ_α has a value around 0.1 which gives all particles a nonzero weight, adding robustness to noise and errors. The function of equations 20 and 21 is that the particles are weighted high, $\alpha_t^m = 1-\beta$, when the particle state matches the measurement, and low, $\alpha_t^m = \beta + \epsilon_\alpha$, when the particle state does not match the measurement or when the particle has moved past a node when the robot has not detected one.

When $z_t = N$ and the robot has turned, giving a measurement of angle $\Delta\theta_t$, the weights of the particles are determined as follows. If the particle transitioned backwards into the node, its weight is set to a fixed value. If the particle transitioned forwards into the node, the predicted continuous direction is found using the uncertain $\Delta\theta_t$ with

$$\tilde{\theta}_t^n = \tilde{\theta}_{t-1}^n + \Delta\theta_t \tag{22}$$

The angular differences $\delta\theta_{t,k}^{n,m}$ between $\tilde{\theta}_t^n$ and the angles of each particle are found and are used to compute the weights. The weights for each particle are therefore given by

$$\alpha_{t,\theta}^m = \begin{cases} e^{-\delta\theta_{t,k}^{n,m\,2}/2\sigma_\rho^2}, & \text{if } \tilde{d}_t^{n,m} < 0 \\ 1/2, & \text{if } \tilde{d}_t^{n,m} > 0 \end{cases} \tag{23}$$

depending on whether the particle transitioned forwards or backwards given by the particle direction $\tilde{d}_t^{n,m}$, where σ_ρ determines the width of the distribution in angle. Note that this weight is computed using both equation 15 and 13.

The particles are therefore weighted higher when the angle they have turned is close to the measured angle $\Delta\theta_t$. Over all

nodes, this has the effect of giving lower weights to particles in nodes where there is no possible turn with an angle close to $\Delta\theta_t$, allowing the algorithm to determine to some extent how likely it is that the robot is at a given node.

The weights $\alpha^m_{t,L}$, $\alpha^m_{t,N}$, and $\alpha^m_{t,\theta}$ are used in particle resampling as in equation 17. To improve on *sequential importance* resampling, a combination of *stratified* and *low variance* resampling is used in this algorithm [18].

A comparison of efficiency between the two algorithms can be made. The core particle filter algorithms have the same order of computational complexity in terms of the number of particles, which is linear, $O(M)$, for the prediction and weighting, and linear using low variance resampling, and log-linear, $O(M\log(M))$, using sequential importance resampling [18]. However, the lower dimensionality of the 1D hybrid space compared to the 2D continuous space means that the number of particles needed to estimate the posterior distribution in hybrid space is smaller [29].

B. Improved Robustness: Mixture Distribution Sampling

One method of achieving relocalization in the case of estimation error is to use a *mixture proposal distribution* in the particle filter [30], [18]. The roles of the prediction and weighting processes from the typical particle filter are reversed for a subset of the particles. As well as predicting particles from the particle set $\mathcal{X}_{t-1}$, particles can also be predicted from the measurement model as

$$\tilde{\mathbf{x}}^m_t \sim p(\mathbf{z}_t|\mathbf{x}_t) \tag{24}$$

This is done at every time step, in case of estimation error, and a particle is sampled in each discrete state adjacent to each discrete state currently containing a particle.

In other applications this can be somewhat difficult, however in this case is very simple: when $z_t = N$, particles can be sampled at all nodes $i \in \mathcal{N}$ in all discrete directions d^n. A similar use of mixture distribution sampling could also be applied to the algorithm in continuous space as well as in discrete space. However, if the knowledge of the environment's discrete nature and topology is not used, sampling will be inefficient as particles could be placed at nodes which are nearby but not connected, and placed in directions which do not correspond to any of the discrete links, which will likely be soon weighted low. With this knowledge, the resulting algorithm would be similar to the discrete space algorithm presented by this paper.

The corresponding weight can be calculated using

$$\alpha^m_t = \int p(\tilde{\mathbf{x}}^m_t|\mathbf{u}_t, \mathbf{x}_{t-1})b(\mathbf{x}_{t-1})d\mathbf{x}_{t-1} \tag{25}$$

With the particle representation of the posterior distribution $p(\mathbf{x}_{t-1}|\mathbf{u}_{1:t-1}, \mathbf{z}_{1:t-1})$, the integration over the possible pose estimates at time $t-1$ can be done as follows by summing over all particles $\mathbf{x}^k_{t-1}$ in $\mathcal{X}_{t-1}$, giving

$$\alpha^m_t = \sum_k p(\tilde{\mathbf{x}}^m_t|\mathbf{u}_t, \mathbf{x}^k_{t-1}) \tag{26}$$

This probability is found for each new particle $\tilde{\mathbf{x}}^m_t$ for each particle $\mathbf{x}^k_{t-1}$ using a similar measure to equation 19.

For this weight computation, a Gaussian kernel is applied to each particle. The weight is then computed as the integral of the product of this kernel and equation 19 over space x^l. The Gaussian kernel probability density is given by

$$p(\tilde{x}^k_t = x^l) = \frac{1}{\sigma_g\sqrt{2\pi}}e^{-(\tilde{x}^k_t - x^l)^2/2\sigma^2_g} \tag{27}$$

and therefore the weight for the new particle $\tilde{\mathbf{x}}^m_t$ is given by

$$\alpha^m_t = \sum_k \int p(i^k_t = i^m)p(\tilde{x}^k_t = x^l)dx^l \tag{28}$$

The particle position $\tilde{x}^k_t$ is taken after the particle has moved according to the motion model, and before the particle transitions to a node. This integral can be simply computed as the product of two Gaussian functions is a Gaussian function, for which the integral is known analytically. It takes a value in the range 0 to ${\sigma_g}^2{\sigma_n}({\sigma_g}^2 + {\sigma_n}^2)^{-\frac{1}{2}}$.

Only particles in links adjacent to each node add to the weight of a particle at that node, and particles sampled in nodes with no particles in adjacent links will have a weight equal to 0. For efficiency, particles can be sampled as in equation 24 only in nodes with a possible non-zero weight.

The weighted set of particles from both the typical prediction and the mixture distribution prediction are combined.

V. Experiments and Discussion

Two algorithms are compared in this experiment:

1) Localization in 2D continuous space, described in section III (*2D*), based on previous work.
2) Newly developed localization in hybrid 1D metric-topological space, described in section IV (*1D*).

The algorithms are designed for minimal sensor measurements which could be extracted from a range of sensors in a practical dataset. An example of this is shown in Figure 2, which shows a trajectory estimated from sensor data in a dataset collected by the *SIAR* platform [9]. The trajectory is estimated using the odometry measurements from wheel encoders and IMUs, and the detection of manhole features.

However, rather than attempting to evaluate the algorithms on a small set of practical data, simulation is used to assess the performance of the algorithms over a large number

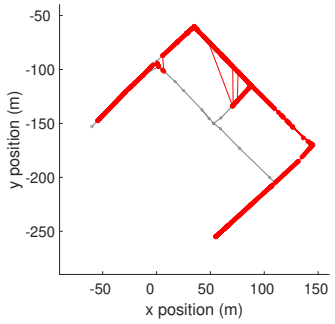

Fig. 2. A trajectory estimated using the *1D* algorithm using the experimental dataset acquired from the *SIAR* platform. Odometry and manhole detection data is input to the algorithm.

of measurements, allowing testing on a large number of trajectories with a range of magnitudes of uncertainty. The aim of the experiments is to demonstrate the algorithms with a large set of data, to determine how susceptible they are to various sources of uncertainty, and to determine what magnitudes of uncertainty cause them to fail. The simulated environment is built from a map of a real pipe network.

In the following experiments, the number of particles is assumed to reflect the computational cost required. However, it should be noted that it is difficult to translate between these measures. In testing, the computation time per particle for the *1D* algorithm is seen to be approximately half that of the *2D* algorithm. The computation time depends on the implementation of the algorithms, and the efficiency of both algorithms may be improved further. It is assumed here that, as both algorithms are based on the particle filter, they will have approximately the same computational cost per particle.

For illustration, simulation over 10000 time steps, around 44 km in distance, is shown in Figure 3 (for 1000 and 100 particles respectively for the *2D* and *1D* algorithms). For evaluation of the localization algorithms, simulation of 100 trajectories of 1000 time steps, around 4.5 km of distance each trajectory, is used. These distances are of the same order of magnitude as distances covered by inspection systems such as Pure's *SmartBall* and WRc's *Sahara*.

The two localization algorithms are used to estimate the robot's position, each with 100, 200, and 400 particles. Five sources of uncertainty are tested, and three magnitudes of each source of uncertainty are compared in Figure 4. In each case, the *error rate*, defined as the proportion of time for which the error is greater than a threshold 25 m, is shown, using a *violin* plot to show the normalized probability density and median of the error rate for the 100 trajectories. The default parameters used to describe the robot operation as defined in section II are shown in Table I. The default localization parameters are given in Table II.

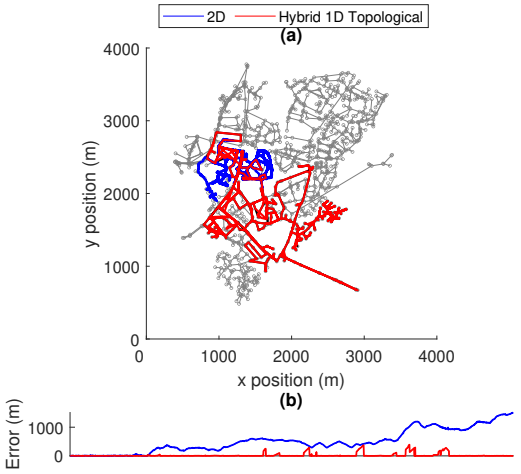

Fig. 3. Illustrative results showing the performance of the two algorithms. (a) An illustration of the two estimates of the robot's trajectory over 10000 steps, corresponding to a total distance of around 44 km. (b) The absolute error of the trajectory estimate over the first 2000 steps of the trajectory.

The five sources of uncertainty are:

1) Gaussian motion noise using the motion model in equation 3, with variation in σ_x.
2) Angular measurement noise using the measurement model in equation 5, with variation in σ_θ.
3) Integrated uniform motion noise using the motion model described by equation 4, with variation in u_x. A given value of u_x gives approximately the same overall noise frequency as 0.2 times the same value of σ_x.
4) False positive and false negative rates of detection of nodes, β_p and β_n.
5) Error in the maps $\mathcal{M}_{metric}$ and $\mathcal{M}_{hybrid}$. In reality, maps of pipe networks will be largely correct with occasional errors. However, in this experiment the entire map is somewhat distorted to usefully test the performance of the algorithms in the case of an error in the map. Map errors are expected to be smooth over space, i.e. nearby points will all have similar error, and are expected to be relatively small in magnitude compared to the size of the map. To model this, each node with position $\mathbf{x}_n^g = \left[x_n, y_n\right]$ is moved in both x_n and y_n with displacement given by

$$\delta\mathbf{x}_n^g = \Delta\mathbf{x}_{\mathcal{M}}(\sin k_{\mathcal{M}} x_n + \sin k_{\mathcal{M}} y_n) \qquad (29)$$

where $k_{\mathcal{M}} = 0.01$ is a constant determining the spatial frequency of the distortion, and $\Delta\mathbf{x}_{\mathcal{M}}$ is the magnitude of the distortion. For nonzero distortion, the particle filter uncertainty parameters are increased as follows: $\sigma_\psi = 0.4$, $\sigma_\omega = 0.2$, $\sigma_l = 10$, $\sigma_n = 10$, $\sigma_\rho = 2.5$.

From Figure 4, it can be seen that in almost all the measurements, the average error rate decreases with increasing number of particles, but with less improvement at larger number of particles. In practice, the number of particles could be chosen based on the required performance and cost.

TABLE I

DEFAULT PARAMETERS FOR THE ROBOT MOTION AND MEASUREMENT.

Parameter	Symbol	Value
Command input motion	Δx (normal)	5
Normal motion noise	σ_x	0.2
Angular measurement noise	σ_θ	0.1
Uniform motion noise	u_x	1 m
Motion noise constant	k_v	0.8
False positive rate	β_p	0
False negative rate	β_n	0
Map distortion	$\Delta x_{\mathcal{M}}$	0

TABLE II

DEFAULT PARAMETERS FOR THE LOCALIZATION ALGORITHMS

	Parameter	Symbol	Value
2D	Motion model noise	σ_ψ	$1.2\sigma_x$
	Angular motion model noise	σ_ω	$1.2\sigma_\theta$
	Angular motion model noise	$\sigma_{\omega,min}$	0.1 rad
	Link measurement std.	σ_l	Δx m
	Node measurement std.	σ_n	Δx m
1D	Motion model noise	σ_ψ	$1.2\sigma_x$
	Node transition std.	σ_n	Δx m
	Angular weight std.	σ_ω	$10\sigma_\theta$
	Kernel std.	σ_g	$5\Delta x$ m

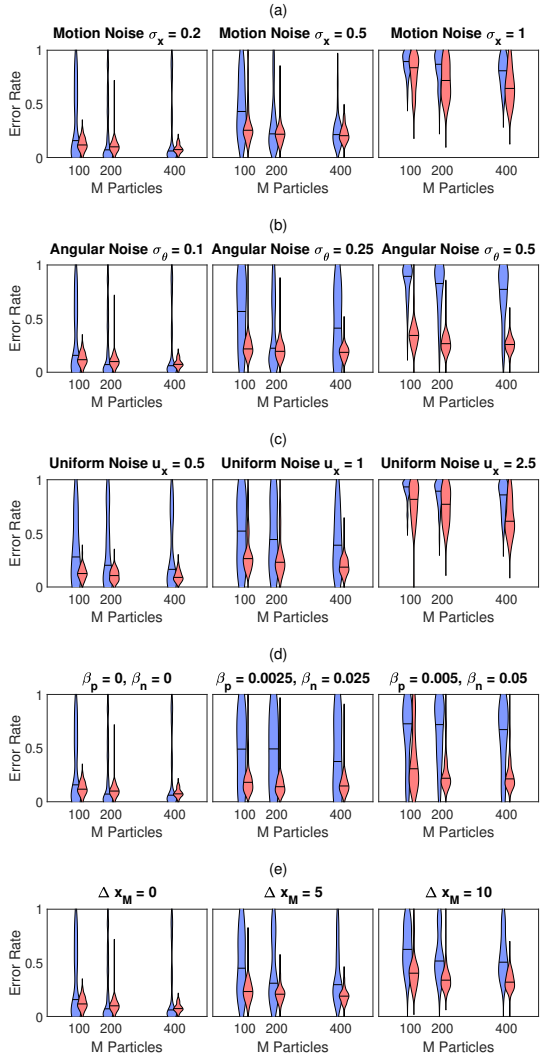

Fig. 4. For the two algorithms (red: *1D*, blue: *2D*), the error rate for variation in: (a) the Gaussian motion noise magnitude. (b) the angular measurement Gaussian noise magnitude. (c) the magnitude of integrated uniform noise in motion. (d) the measurement error rate. (e) the error in the knowledge of the environment map.

For the lowest uncertainty in all aspects when Gaussian motion noise is used, the *2D* algorithm performs equal to or better on average than the *1D* algorithm, showing that it is an effective algorithm when uncertainty is low, which is more likely to be the case for larger, more capable robots. For integrated uniform noise, which models unmeasured drift in robot velocity, the *1D* algorithm performs better on average.

Larger magnitudes of uncertainty in all aspects reduces the performance of both algorithms, but less so for the *1D* algorithm. This is most prominent for angular measurement noise and false measurement rate, where the *1D* is robust.

While the error from the *1D* algorithm is lower on average, both algorithms show poor performance at high magnitudes of either motion noise, indicating that good measurement of linear motion is important. Both algorithms give only a slow increase in error for increasing map error, showing that both are robust even in this extreme case of error over the entire map, although the *1D* algorithm performs better on average.

Overall, it can be seen that the two algorithms presented here have similar performance when uncertainty is low. As uncertainty increases in all aspects tested, the proposed *1D* algorithm is seen to have a better performance, at a similar computational cost, showing that the algorithm is more efficient than the *2D* algorithm in these cases.

Considering the practicality of the proposed algorithm, acquiring the minimal measurements required as inputs is possible with any of a range of possible sensors; reference to the literature on robotic sensing for pipes is given in section II-C. Similarly, with small modifications to the algorithm, the robot's locomotion could be modelled for a variety of designs not considered in this work. For example, if the robot were able to turn around in a pipe, this could be incorporated into the algorithm as a possible change in direction d_t.

The proposed algorithm uses only input from odometry and binary feature detection. However, use of more sensing developed for pipes is allowed by the robot state definition, such as visual odometry and junction classification [2], [25], [23], tactile sensing of pipe joints [31] and corners [32], detection of junctions using acoustic echoes [4], and junction classification using a scanning rangefinder [26].

The results have practical implications. The aim with these algorithms is to provide an efficient estimate of the robot's position, which is of sufficient accuracy for navigation and for reporting the approximate location of faults found in the pipe network. An acceptable rate of error in the position estimation depends on the operation and broad characteristics of the robotic system. For example, a high rate of error in localization might be acceptable if the robot is extracted from the network frequently for data collection, or if beacons are installed in the pipe network to facilitate relocalization. If a very low rate of error is required for a particular application, these results offer a measure of the level of uncertainty that is therefore required. Conversely, the results offer a measure of the error that would be expected for a given level of uncertainty, even when the rate of error is high.

Active detection of mislocalization and a subsequent method of attempting relocalization would improve robustness to the various sources of uncertainty. The particle filtering (and perhaps any filtering) approach might be limited in performance, and smoothing or other formulations of the use of a sequence of measurements rather than recursive estimation may be needed to achieve a high robustness.

VI. Conclusions

Robot localization is challenging in a buried pipe environment due to restricted sensing and computation power. This paper has presented an efficient, robust particle filter localization algorithm where the position of the robot is defined by a hybrid metric-topological state. This state definition allows efficient prediction and weighting of particles, robust mixture distribution sampling and relocalization. Experiments in simulation over a very large scale over variation in a range of sources of uncertainty show the algorithm has a lower rate of error than a benchmark algorithm.

REFERENCES

[1] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard, "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age," *IEEE Transactions on Robotics*, vol. 32, no. 6, pp. 1309–1332, 2016.

[2] P. Hansen, H. Alismail, P. Rander, and B. Browning, "Visual mapping for natural gas pipe inspection," *International Journal of Robotics Research*, vol. 34, no. 4-5, pp. 532–538, 2015. [Online]. Available: https://journals.sagepub.com/doi/abs/10.1177/0278364914550133

[3] K. Ma, M. M. Schirru, A. H. Zahraee, R. Dwyer-Joyce, J. Boxall, T. J. Dodd, R. Collins, and S. R. Anderson, "Robot mapping and localisation in metal water pipes using hydrophone induced vibration and map alignment by dynamic time warping," *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 2548–2553, 2017. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/7989296

[4] R. Worley, Y. Yu, and S. Anderson, "Acoustic echo-localization for pipe inspection robots," *IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, pp. 2–7, 2020. [Online]. Available: https://ieeexplore.ieee.org/document/9235225

[5] R. Worley, K. Ma, G. Sailor, M. M. Schirru, R. Dwyer-joyce, J. Boxall, T. Dodd, R. Collins, and S. Anderson, "Robot localization in water pipes using acoustic signals and pose graph optimization," *Sensors (Switzerland)*, pp. 1–23, 2020. [Online]. Available: https://www.mdpi.com/1424-8220/20/19/5584

[6] Y. Bando, H. Suhara, M. Tanaka, T. Kamegawa, K. Itoyama, K. Yoshii, F. Matsuno, and H. G. Okuno, "Sound-based online localization for an in-pipe snake robot," *SSRR 2016 - International Symposium on Safety, Security and Rescue Robotics*, pp. 207–213, 2016. [Online]. Available: https://ieeexplore.ieee.org/document/7784300

[7] T. Seco, M. T. Lazaro, C. Rizzo, J. Espelosin, and J. L. Villarroel, "Graph-based robot localization in tunnels using RF fadings," in *Robot 2019: Fourth Iberian Robotics Conference*, 2019. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-030-35990-4{_}47

[8] D. Alejo, F. Caballero, and L. Merino, "A robust localization system for inspection robots in sewer networks," *Sensors (Switzerland)*, vol. 19, no. 22, pp. 1–28, 2019. [Online]. Available: https://www.mdpi.com/1424-8220/19/22/4946

[9] D. Alejo, F. Chataigner, D. Serrano, L. Merino, and F. Caballero, "Into the dirt: Datasets of sewer networks with aerial and ground platforms," *Journal of Field Robotics*, no. July, 2020.

[10] F. Chataigner, P. Cavestany, M. Soler, C. Rizzo, J.-P. Gonzalez, C. Bosch, J. Gibert, A. Torrente, R. Gomez, and D. Serrano, "ARSI: an aerial robot for sewer inspection," *Springer Tracts in Advanced Robotics*, vol. 132, no. July, pp. 249–274, 2020. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-030-22327-4{_}12

[11] A. Jacobson, F. Zeng, D. Smith, N. Boswell, T. Peynot, and M. Milford, "What localizes beneath: A metric multisensor localization and mapping system for autonomous underground mining vehicles," *Journal of Field Robotics*, vol. 38, no. 1, pp. 5–27, 2021. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.21978

[12] T. Özaslan, G. Loianno, J. Keller, C. J. Taylor, V. Kumar, J. M. Wozencraft, and T. Hood, "Autonomous navigation and mapping for inspection of penstocks and tunnels with MAVs," *IEEE Robotics and Automation Letters*, vol. 2, no. 3, pp. 1740–1747, 2017. [Online]. Available: https://ieeexplore.ieee.org/document/7914761

[13] J.-l. Blanco, J.-a. Fernandez-Madrigal, and J. Gonzalez, "Toward a unified Bayesian approach to hybrid metric – topological SLAM," *IEEE TRANSACTIONS ON ROBOTICS, VOL. 24*, vol. 24, no. 2, pp. 259–270, 2008. [Online]. Available: https://ieeexplore.ieee.org/document/4472721

[14] S. Thiebaux and P. Lamb, "Combining Kalman filters and Markov localization in network-like environments," 2000. [Online]. Available: https://link.springer.com/chapter/10.1007/3-540-44533-1{_}75

[15] P. Merriaux, Y. Dupuis, P. Vasseur, and X. Savatier, "Fast and robust vehicle positioning on graph-based representation of drivable maps," *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2015-June, no. June, pp. 2787–2793, 2015. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/7139578

[16] M. A. Brubaker, A. Geiger, and R. Urtasun, "Map-based probabilistic visual self-localization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 4, pp. 652–665, 2016. [Online]. Available: https://ieeexplore.ieee.org/document/7152950

[17] F. Bernuy and J. Ruiz-del Solar, "Topological semantic mapping and localization in urban road scenarios," *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 92, no. 1, pp. 19–32, 2018. [Online]. Available: https://link.springer.com/article/10.1007/s10846-017-0744-x

[18] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. The MIT Press, 2006.

[19] T. B. Kwon, J. H. Yang, J. B. Song, and W. Chung, "Efficiency improvement in Monte Carlo localization through topological information," *IEEE International Conference on Intelligent Robots and Systems*, no. October, pp. 424–429, 2006. [Online]. Available: https://ieeexplore.ieee.org/document/4059089

[20] T. B. Kwon, J. H. Yang, and J. B. Song, "Efficient and reliable Monte Carlo localization with thinning edges," *International Journal of Control, Automation and Systems*, vol. 8, no. 2, pp. 328–338, 2010. [Online]. Available: https://link.springer.com/article/10.1007/s12555-010-0219-3

[21] J. T. Thielemann, G. M. Breivik, and A. Berge, "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*, 2008. [Online]. Available: https://ieeexplore.ieee.org/document/4563167

[22] W. Zhao, M. Kamezaki, K. Yoshida, M. Konno, A. Onuki, and S. Sugano, "Modeling and simulation of FLC-based navigation algorithm for small gas pipeline inspection robot," *IEEE/ASME International Conference on Advanced Intelligent Mechatronics, AIM*, vol. 2018-July, pp. 912–917, 2018. [Online]. Available: https://ieeexplore.ieee.org/document/8452416

[23] D. H. Lee, H. Moon, J. C. Koo, and H. R. Choi, "Map building method for urban gas pipelines based on landmark detection," *International Journal of Control, Automation and Systems*, vol. 11, no. 1, pp. 127–135, 2013. [Online]. Available: https://link.springer.com/article/10.1007/s12555-012-0049-6

[24] A. Kakogawa, T. Yamagami, Y. Tian, and S. Ma, "Recognition of pathway directions based on nonlinear least squares method," *2015 IEEE International Conference on Robotics and Biomimetics, IEEE-ROBIO 2015*, pp. 1596–1601, 2015. [Online]. Available: https://ieeexplore.ieee.org/document/7418999

[25] A. Kakogawa, Y. Komurasaki, and S. Ma, "Anisotropic shadow-based operation assistant for a pipeline-inspection robot using a single illuminator and camera," *IEEE International Conference on Intelligent Robots and Systems*, vol. 2017-Septe, pp. 1305–1310, 2017. [Online]. Available: https://ieeexplore.ieee.org/document/8202306

[26] F. Kirchner and J. Hertzberg, "A prototype study of an autonomous robot platform for sewerage system naintenance," *Autonomous Robots*, vol. 4, no. 4, pp. 319–331, 1997.

[27] H. Sahli and N. El-Sheimy, "A novel method to enhance pipeline trajectory determination using pipeline junctions," *Sensors (Switzerland)*, vol. 16, no. 4, pp. 1–17, 2016. [Online]. Available: https://www.mdpi.com/1424-8220/16/4/567

[28] L. Guan, X. Xu, Y. Gao, F. Liu, H. Rong, M. Wang, and A. Noureldin, "Micro-inertial-aided high-precision positioning method for small-diameter PIG navigation," *Advances in Human and Machine Navigation Systems*, 2018. [Online]. Available: https://cdn.intechopen.com/pdfs/63868.pdf

[29] P. B. Quang, C. Musso, and F. Le Gland, "An insight into the issue of dimensionality in particle filtering," *13th Conference on Information Fusion, Fusion 2010*, no. May, 2010. [Online]. Available: https://ieeexplore.ieee.org/document/5712050

[30] S. Thrun, D. Fox, W. Burgard, and F. Dellaert, "Robust Monte Carlo localization for mobile robots," *Artificial Intelligence*, vol. 128, no. 1-2, pp. 99–141, 2001. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0004370201000698

[31] Y. Wu, E. Mittmann, C. Winston, and K. Youcef-Toumi, "A practical minimalism approach to in-pipe robot localization," *Proceedings of the American Control Conference*, vol. 2019-July, pp. 3180–3187, 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8814648

[32] L. Brown, J. Carrasco, S. Watson, and B. Lennox, "Elbow detection in pipes for autonomous navigation of inspection robots," *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 95, no. 2, pp. 527–541, 2019. [Online]. Available: http://dx.doi.org/10.1007/s10846-018-0904-7