Robot Localization in a Pipe Network using a Particle Filter with Error Detection and Recovery in a Hybrid Metric-Topological Space

Rob Worley[1], Sean Anderson[1]

***Abstract*— Estimation of a robot's pose in the constrained environment of a buried pipe network is difficult due to uncertainty in motion, limited and unreliable sensing, and limited computational capability on board the robot. An efficient localization algorithm is described here, along with two novel improvements: the detection of mislocalization using the Kullback-Leibler divergence between the distribution of recent particle filter weights and a distribution learned from data for good algorithm performance, overcoming the problem of low information in each individual set of particle filter weights; and the capability for relocalization, using multi-hypothesis estimation to overcome the problems of limited information in sensing. The algorithm uses the low-dimensional metric-topological space of the pipe network to give efficient performance and to facilitate relocalization. Experimental results show that the localization algorithm is effective and that the presented improvements reduce the chance of unrecoverable mislocalization, therefore improving robustness to error in measurements made by the robot.**

I. Introduction

The inspection and maintenance of buried pipe networks is a critical part of managing the risk of blockages and leaks, and subsequent damage to public health and to other infrastructure. Current inspection methods either involve imprecise or expensive sensing from above the surface, or involve costly manual inspection using remotely controlled robots. Inaccurate detection of faults in the pipe network can result in excavation of large sections of road above the buried pipes. There is scope to improve the effectiveness and efficiency of buried pipe inspection, which might be achieved using autonomous robots for pervasive and persistent inspection.

There are challenges in the development of all aspects of this proposed robotic system. The localization of a robot, and subsequently the localization of any faults detected in the network, is especially challenging in this environment. The constraints of the environment limit the size, power available, and therefore quality, for sensors on the robot. The environment also prohibits the use of GPS sensing, reduces the reliability of magnetic field sensing from an IMU, and limits the scope of sensors such as cameras and rangefinders which in other environments might be able to observe features at a distance or from different perspectives.

Despite the challenges, sensing methods for this environment have been reported. Vision has been used for precise odometry and surface reconstruction [1], and also lower dimensional detection of features such as manholes and junctions [2], [3], [4], [5], [6]. Detection of these features has also been achieved using inertial [7], [8], and acoustic [9], [10] sensing, which has also be applied more generally to robot localization in pipes [11], [10], [12], [13], while radio waves have been used for position estimation both directly [14] and via communication with a base station [15].

The constraints of the environment can be advantageous. The robot is constrained to the pipe network, and the small-scale robot pose estimation within a pipe cross-section is decoupled from the large-scale robot position estimation within the pipe network, and the complexity of the state to be estimated is therefore reduced. Maps of the constrained environment are available which aids localization [16], [17]. These advantages have been used previously for localization in pipe networks [17] and road networks [18], [19], [20], [21]. However, these algorithms are applied to more capable robots with less concern about efficiency or robustness to high uncertainty, and the use of these advantages for limited robots in pipes has not been investigated.

Robot localization can estimate a hybrid metric-topological state [22], an idea which has been applied previously to network environments [23]. While this previous implementation uses a measurement of the position not available in the pipe environment, the concept is a good basis for development of algorithms for this application.

Despite the available sensing, efficient robust localization is not trivial. The various sensing methods in the literature give limited information about the environment, and false positive or false negative detection of features is possible. Typical localization algorithms are not inherently robust to this sort of sensor error, and do not facilitate detecting and recovering from error with this limited of sensing.

This paper describes developments to a localization algorithm with low sensing and computational requirements, suitable for online localization of resource-constrained robots in pipes. This is developed from an algorithm reported previously [24], with some improvements to the formulation here. The algorithm uses only linear and angular odometry and simple binary detection of features in the environment, which could be acquired using sensing methods reported in the literature. The advantages of the constraints of the network environment allow efficient large-scale localization over a pipe network, facilitating navigation and the reporting of pipe faults with online localization.

The novelty of this paper is in error detection and recovery, improving the robustness of the previously reported algo-

*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

rithm. Detection of mislocalization in the application to limited sensing is done using the Kullback-Leibler divergence between the distribution of recent particle filter weights, to overcome the problem of limited information in the individual particle weights, described in section V. Efficient, effective relocalization is achieved using a multi-hypothesis estimator which is robust to sensing errors, allowing recovery from error despite the inability to recognise any location with the limited sensor information, described in section VI. These solutions are applied to robots in pipes here, but could be applied generally to robots with limited sensing.

II. Problem Definition

In this section, the specific problem addresssed by the presented algorithm is defined.

A. Environment and State Definition

Pipe networks are comprised of pipes connected at junctions, corners, and manholes. A single network that could be traversable for a robot might cover tens of square kilometers of area, and tens of kilometers of total distance. In this work, a simulated environment is created from a map of a large pipe network in the UK, partly illustrated in Figure 1 , allowing experimentation over a scale that could not be done practically at present.

The trajectory of the robot is modelled as discrete time steps where the robot moves and then makes a localization estimate. The robot's pose $\mathbf{x}_t$ at time step t is defined in a hybrid metric-topological coordinate system as the triple

$$\mathbf{x}_t = \big(i_t, x_t, d_t\big) \tag{1}$$

$i_t \in \mathcal{I}$ is the discrete index of the link (pipe) or node (junction or manhole), 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 in the link, where $d_t = d_t^l \in \big\{-1, 1\big\}$, or node, where $d_t = d_t^d d_t^n$, with $d_t^n \in \big\{1, 2, \dots, D^n\big\}$ where there are D^n adjacent links at node n, and $d_t^d \in \big\{-1, 1\big\}$ so the sign of the index depends on whether the robot is arriving or leaving from a direction.

B. Motion and Measurement Definition

The robot's linear position is changed according to Δx_t, which is usually equal to a fixed command input, except when the robot stops as it arrives at a node, given by

$$x_t^l = x_{t-1}^l + \Delta x_t d_t + v_t \tag{2}$$

where $v_t \sim \mathcal{N}(\mu = 0, \sigma = \sigma_x \Delta x_t)$ is Gaussian uncertainty and Δx_t is either known from a control input or measured using odometry, and can be used in localization. Gaussian uncertainty is used here for simplicity, but in previous work, experimentation with a more complex uncertainty model was done [24], as the particle filter implementation is able to use any arbitrary noise model.

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. Change in angle, $\Delta\theta_t$, is measured, given by

$$\Delta\theta_t = \theta_t^n - \theta_{t-1}^n + w_t \tag{3}$$

with Gaussian uncertainty $w_t \sim \mathcal{N}(\mu = 0, \sigma = \sigma_\theta \Delta\theta_t)$ which models the uncertain relative angular measurement that could be made using a gyroscope in an IMU, where the absolute angular measurement from the IMU is unavailable due to the unreliable magnetic field in this environment.

A discrete measurement $z_t \in \big\{L, N, T\big\}$ is made with the following probabilities (considering $z_t \in \big\{L, N\big\}$ briefly)

$$p(z_t = L | i_t \in \mathcal{L}) = 1 - \beta_p \tag{4a}$$

$$p(z_t = L | i_t \in \mathcal{N}) = \beta_n \tag{4b}$$

$$p(z_t = N | i_t \in \mathcal{L}) = \beta_p \tag{4c}$$

$$p(z_t = N | i_t \in \mathcal{N}) = 1 - \beta_n \tag{4d}$$

where β_p and β_n are the probabilities of false positive and false negative detection of nodes respectively. A measurement $z_t = T$ can only, and always will, immediately follow a measurement $z_t = N$, superseding these probabilities. In this case, the robot will have stopped due to its observation of a node, and will turn (or believe it has turned, in the case of a false positive).

This represents the minimal measurement that could be made in order for the robot to be able to turn at a node, and move straight in a link. 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.

C. Topological Map Definition

For localization, the robot uses a hybrid *metric-topological* map, $\mathcal{M}_{MT}$. For each link, the map contains the index l, the two adjacent node indices, and the length x_l^L, For each node, the map contains the node index, n, the indices of adjacent links, and the directions to adjacent links, Θ^n. It is assumed that the metric and topological aspects of the map are correct.

The position of each link and node is known so transforming between the metric-topological space, $\mathbf{x}_t$, and a two-dimensional coordinate space, $\mathbf{x}_t^g = \big[x_t^g, y_t^g, \theta_t^g\big]$ is simple.

Measurements ξ_t, ω_t, and ζ_t can be made between an estimated robot pose $\hat{\mathbf{x}}_t = \big(\hat{i}_t, \hat{x}_t, \hat{d}_t\big)$ and this map. ξ_t is the distance to the nearest node, found using

$$\xi_t = \min\big\{|\hat{x}_t|, |x_{\hat{i}}^L - \hat{x}_t|\big\} \tag{5}$$

ω_t is the minimum difference in angle between a continuous direction and the discrete directions at a node, found using

$$\omega_t = \min\Big\{|\Theta_{\hat{d}_t}^n - (\Theta_{\hat{d}_{t-1}}^n + \Delta\theta_t)|\Big\} \tag{6}$$

where $\Theta_{\hat{d}_t}^n$ is the continuous direction corresponding to the estimated discrete direction $\hat{d}_t$. ζ_t is made as the estimated state moves from $\hat{\mathbf{x}}_{t-1}$ to $\hat{\mathbf{x}}_t$, given by

$$\zeta_t = \begin{cases} \bar{N} & \text{if } \hat{i}_t \neq \hat{i}_{t-1},\, \hat{i}_t \in \mathcal{L},\, \hat{i}_{t-1} \in \mathcal{L} \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

and is used to measure if a robot in the estimated position would have made an observation of a node.

D. Problem Summary and Motivation for Novelty

Overall, the robot makes three observations: Δx_t, $\Delta\theta_t$, and z_t, and three types of map observation: ξ_t^m, ω_t^m, and ζ_t^m, that are available for use in localization.

Compared to observations made from typical sensors such as cameras and lidar, the observations Δx_t, $\Delta\theta_t$, and z_t contain limited information. Subsequently, the *prediction* and *weighting* models (equations 10 and 11) can give limited information about the quality of the estimate made by the particle filter, which is commonly used to detect error, or *mislocalization*. This is seen in Section IV in the limited values which can be assigned to weights α_t^m, and motivates the development of the algorithm for detecting mislocalization given in Section V.

Similarly, the limited observations affect recovery from error, or *relocalization*. In general particle filtering for localization, in the case of mislocalization it is desired to sample particles using *mixture distribution sampling* at poses which match the current observation $\mathbf{z}_t$. However in this problem definition, the observation gives very little information, and sampling in this way would produce a huge number of particles distributed across the whole state space. This motivates the development of the relocalization algorithm in Section VI, where the metric-topological map $\mathcal{M}_{MT}$ facilitates relocalization by constraining the estimate to the map, and reducing the dimension, and therefore size, of the state space.

III. Particle Filter Localization Algorithm

The robot localization algorithm estimates the posterior distribution $p(\mathbf{x}_t|\mathbf{u}_{1:t}, \mathbf{z}_{1:t})$, which is the probability distribution over possible current states $\mathbf{x}_t$ given the sequence of uncertain observations $\mathbf{u}_{1:t}$ and $\mathbf{z}_{1:t}$.

The *particle filter* describes the posterior distribution as a set of M samples, or *particles* [25]. The likelihood of the state $\mathbf{x}_t^m$ of particle m being included in the particle set $\mathcal{X}_t$ is proportional to the posterior, or *belief*, $b(\mathbf{x}_t)$, given by

$$\mathbf{x}_t^m \sim b(\mathbf{x}_t) = p(\mathbf{x}_t|\mathbf{u}_{1:t}, \mathbf{z}_{1:t}) \tag{8}$$

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

$$b(\mathbf{x}_t) \propto p(\mathbf{z}_t|\tilde{\mathbf{x}}_t) \int p(\tilde{\mathbf{x}}_t|\mathbf{u}_t, \mathbf{x}_{t-1}) b(\mathbf{x}_{t-1}) d\mathbf{x}_{t-1} \tag{9}$$

Typically, $\mathbf{u}_t$ contains information about the transition between states $\mathbf{x}_{t-1}$ and $\mathbf{x}_t$, and $\mathbf{z}_t$ contains information about the observations made in the current state $\mathbf{x}_t$.

The state is *predicted* by the set $\tilde{\mathcal{X}}_t$, which is sampled as

$$\tilde{\mathbf{x}}_t^m \sim p(\mathbf{x}_t|\mathbf{u}_t, \mathbf{x}_{t-1}^m) \tag{10}$$

The particles are then *weighted* according to

$$\alpha_t^m \propto p(\mathbf{z}_t|\tilde{\mathbf{x}}_t^m) \tag{11}$$

Resampling creates $\mathcal{X}_t$ by drawing particles from $\tilde{\mathcal{X}}_t$ using

$$p(\mathbf{x}_t^m \in \mathcal{X}_t) \propto \alpha_t^m \tag{12}$$

Finally, the mode of the particle distribution is estimated, and output as the estimated robot state. This process is repeated at each time step t, when new information is available.

IV. Particle Filter for the Hybrid Metric-Topological State

A localization algorithm using the particle filter for metric-topological space has been designed for using these minimal measurements, improved in formulation from previous work [24]. The use of the algorithm is visualized in Figure 1.

The posterior is expanded to $p(\mathbf{x}_t|\mathbf{u}_{1:t}, \mathbf{z}_{1:t}, \mathcal{M}_{MT})$ to include the map $\mathcal{M}_{MT}$. $\mathbf{u}_t$ and $\mathbf{z}_t$ incorporate the observations made by the robot and from the map, defined as

$$\mathbf{u}_t^m = \left(\Delta x_t, z_t, \xi_t^m, \omega_t^m, \zeta_t^m\right) \tag{13}$$

$$\mathbf{z}_t^m = \left(\Delta\theta_t, z_t\right) \tag{14}$$

Observations ξ_t^m, ω_t^m, and ζ_t^m can be made for each particle m. z_t appears in both $\mathbf{z}_t$ and $\mathbf{u}_t$, as it is used as a typical observation in the former case, and also determines what model is used for the state transition.

A. Linear Motion

When $z_t = L$, the particles are moved linearly as follows. For particles where $i_{t-1}^m \in \mathcal{L}$, the position is updated according to equation 10, given by

$$\tilde{x}_t^{l,m} = x_{t-1}^{l,m} + \Delta x_t d_{t-1}^{l,m} + \psi_t \tag{15}$$

where ψ_t is an estimate of the Gaussian distributed noise and is drawn from a Gaussian distribution with a variance equal to $\sigma_\psi \Delta x_t$. If $\tilde{x}_t^{l,m}$ is greater than x_i^L or less than zero, the particle is moved to a neighbouring link $\tilde{i}_t \in \mathcal{L}^n$ with uniform probability, and $\tilde{d}_t$ and $\tilde{x}_t^{l,m}$ are updated accordingly.

Each particle is then weighted according to equation 11 and 10 in a form of mixture distribution sampling, given by

$$\alpha_{t,L}^m = \begin{cases} 1 - \beta_n, & \text{if } \zeta_t^m = 0 \\ \beta_n + \epsilon_\alpha, & \text{if } \zeta_t^m = \bar{N} \end{cases} \tag{16}$$

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 equation 16 is that the particles are weighted high, $\alpha_t^m = 1 - \beta_n$, when the particle state matches the measurement, and low, $\alpha_t^m = \beta_n + \epsilon_\alpha$, when the particle has moved past a node when the robot has not detected one.

B. Mixture Distribution Sampling at Nodes

Mixture distribution sampling [25] is used when $z_t = N$, which is simple with the metric-topological state definition. If $z_t = N$, particles may move to a node index. The probability of a particle moving to the nearest node i^n is given by

$$p(i_t^m = i^n) = e^{-\xi_t^{m2}/2\sigma_n^2} \tag{17}$$

If the particle moves to a node the corresponding direction index $\tilde{d}_t^n$ is chosen. A negative index is chosen if the particle has moved forwards into the node, and a positive index is chosen if the particle has moved backwards.

Each particle is then weighted according to

$$\alpha_{t,N}^m = \begin{cases} 1 - \beta_p, & \text{if } \tilde{i}_t^m \in \mathcal{N} \\ \beta_p + \epsilon_\alpha, & \text{if } \tilde{i}_t^m \in \mathcal{L} \end{cases} \tag{18}$$

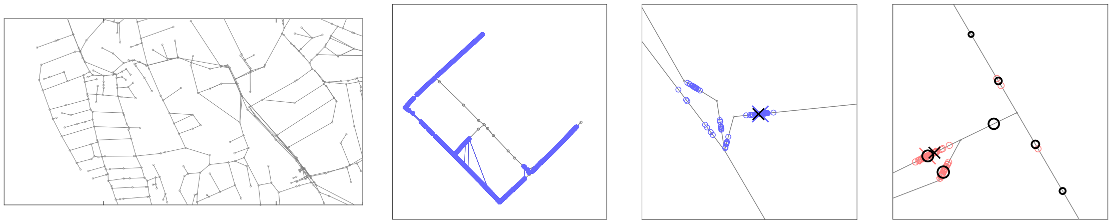

Fig. 1. **Left**: An example of a map of a pipe network used in this work. **Middle-Left**: An example of the performance of the algorithm described in Section IV on experimental data from the SIAR dataset [26]. The grey circles and lines represent the manholes and pipes in the map, and the blue circles show the robot poses estimated in this map, with the blue lines indicating adjacent poses in time. **Right**: An illustration of the use of two localization algorithms, compared to the robot pose given by the black cross. **Middle-Right**: The algorithm described in Section IV. The blue circles are the particles in the filter, and the blue cross is the particle filter estimate. **Far-Right**: The improved localization algorithm, described in Section V and VI. The red circles are the particles in the filter, the black circles are the hypotheses in the multi-hypothesis filter with size depending on their weight, and the red cross is the particle filter estimate.

where again ϵ_α has a value around 0.1. The effect of this sampling and weighting is that particles are moved and weighted high, $\alpha_t^m = 1 - \beta_p$, when their state is likely to match the measurement, and weighted low otherwise.

C. Angular Motion

After the robot observes a node, and $z_t = T$, it will make an angular motion to move to a new link. The particle distribution is moved accordingly. For prediction, the particles m for which $i_t^m \in \mathcal{N}$ are moved with uniform probability to one of the possible discrete angles at that node.

The particles are weighted as follows. If the particle transitioned backwards into the node ($\tilde{d}_t^{n,m} > 0$), its weight is set to a fixed value. If the particle transitioned forwards into the node ($\tilde{d}_t^{n,m} < 0$), a measurement ω_t^m is available. The weight is therefore given by

$$\alpha_{t,T}^m = \begin{cases} e^{-\omega_t^{m2}/2\sigma_\omega^2}, & \text{if } \tilde{d}_t^{n,m} < 0 \\ 1/2, & \text{if } \tilde{d}_t^{n,m} > 0 \end{cases} \tag{19}$$

where σ_ω determines the width of the distribution in angle.

D. Resampling

A set of M particles is sampled from the existing set using a combination of *stratified resampling*, where sampling is done separately for each subset of particles in different discrete places, $i \in \mathcal{I}$, and *low variance resampling*, where systematically selects particles, ensuring all particles with an above average weight are sampled at least once [25]. These are especially useful here where a small number of particles is used, and losing important particles due to simple *sequential resampling* is more likely and more detrimental. Particles are drawn from $\tilde{\mathcal{X}}_t$ creating $\mathcal{X}_t$, which is distributed approximately according to equation 8, using $p(\mathbf{x}_t^m \in \mathcal{X}_t) \propto \alpha_t^m$. This is achieved using the weights $\alpha_{t,L}^m$, $\alpha_{t,N}^m$, and $\alpha_{t,T}^m$.

E. Estimation

The modal index $\hat{i}_t$ is found from the indices i_t^m for all particles m. The modal value of $\hat{d}_t$ is then found for all particles $i_t^m = \hat{i}_t$. The median value of x_t^m is then found within these modal discrete indices, as an approximation to the mode, which is used as an estimate of the robot pose.

V. Detection of Mislocalization

In this section, a novel method for detecting failure in the previously described algorithm is given.

If the parameters for the localization algorithm are accurate models of the robot motion, it should be able to track the robot's position. However, there are sources of uncertainty which could cause *mislocalization*, where the algorithm loses track of the robot's position. The chance of mislocalization is increased by the possibility of false positive and negative measurements. A long trajectory gives plenty of opportunity for unfortunate coincidences to cause mislocalization, such as a false positive occurring at a time where an incorrect hypothesis predicts a positive measurement, thus making it appear as though the incorrect hypothesis is correct.

Typical methods of estimating that the distribution $\mathcal{X}_t$ poorly represents the robot's pose are to use the difference between the distribution before and after weighting, ($p(\mathbf{x}_t|\mathbf{u}_{1:t}, \mathbf{z}_{1:t})$ and $p(\mathbf{x}_t|\mathbf{u}_t, \mathbf{z}_t, \mathbf{x}_{t-1})$) [27], or the difference between the probability of measurements ($p(\mathbf{z}_t|\mathbf{x}_t)$) and the normal measurement probability [25]. The principle is that if the robot's measurements are unlikely given the estimated distribution, then the estimated distribution is likely a poor representation of the robot pose. Practically, these probabilities can be found using the particle weights.

However, in the problem defined here, the measurements have limited information to use to detect error. Most information is gained infrequently, when $\mathbf{z}_t = N$. This is reflected in the set of weights $\mathrm{A}_t = \{\alpha_t^1, \alpha_t^2, \ldots, \alpha_t^m, \ldots, \alpha_t^M,\}$, where most, if not all, elements will be equal and close to 1 for a large proportion of times t.

The instantaneous, or temporally filtered, set of weights therefore can't be used in the same way as they would be in a more conventional problem where the weights are based on sensors which give useful information at most times t. Using the weights for this detection only when $z_t = N$ gives only infrequent information, so the particle filter estimate could have degraded further before mislocalization can be detected. This motivates a different approach for this problem.

While the values of the weights can't be used directly to estimate the estimate quality, they do still contain useful information. The novel approach proposed here is to compare

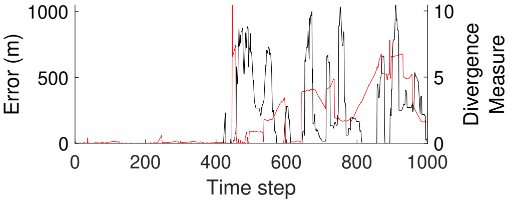

Fig. 2. An example of the performance of the algorithm for detecting mislocalization. The error in the particle filter estimate is shown in red, and the divergence measure, used as an indicator of mislocalization, is shown in black.

a sequence of mean weights $\bar{\alpha}_\tau$ over some set of recent times $\tau = \{t, t-1, \ldots, t-\tau_\alpha\}$, to the weights expected for a *good* performance, $\bar{\alpha}^g$, and for a *bad* performance, $\bar{\alpha}^b$, which can be learned from data where the performance has been measured. This is easy to do in simulation, where the performance can be easily measured for a large set of data. This allows useful information to be found from the weights indirectly from their values. The intuition is that if the sequence $\bar{\alpha}_\tau$ appears to be more similar to what is expected for *bad* performance than *good* performance, mislocalization is likely. In this algorithm, the mean weight is used, and while the set of all weight values A_τ could be used, it wasn't seen to be particularly informative here.

To measure the similarity of $\bar{\alpha}_\tau$ to the *good* and *bad* data, the *Kullback-Leibler Divergence*, D_{kl}, or *relative entropy*, is used as a measure of difference between the probability distribution of $\bar{\alpha}_\tau$, $p(\bar{\alpha}_\tau)$, and of the data, $p(\bar{\alpha}^g)$ and $p(\bar{\alpha}^b)$. D^g_τ is defined by

$$D_{kl}\left(p(\bar{\alpha}_\tau)||p(\bar{\alpha}^g)\right) = \sum p(\bar{\alpha}_\tau) \log\left(\frac{p(\bar{\alpha}_\tau)}{p(\bar{\alpha}^g)}\right) \tag{20}$$

and D^b_τ is defined similarly. This measure is in the range $[0, \infty)$, where two identical distributions give a value of 0.

It is seen that the distribution of weights $p(\bar{\alpha}_\tau)$ is substantially different for $z_t = L$, $z_t = N$, and $z_t = T$. Therefore, a different probability distribution and corresponding divergence is used for each case (e.g. $p(\bar{\alpha}^L_\tau)$, D^{gL}_τ). It is also seen that both the short-term and long-term sequence of weights can be useful for detecting mislocalization; using the short-term weights allows the detection to be faster, while using the long-term weights allows the detection to be more reliable. Therefore, in the proposed algorithm, two sets of time τ_s and τ_l are used, with different time lengths $\tau_{\alpha s}$ and $\tau_{\alpha l}$.

In total, there are therefore six variable distributions, $p(\bar{\alpha}^L_{\tau_s})$, $p(\bar{\alpha}^L_{\tau_l})$, $p(\bar{\alpha}^N_{\tau_s})$, etc., and six learned distributions, $p(\bar{\alpha}^{gL})$, $p(\bar{\alpha}^{bL})$, etc.

The divergence is found between each variable distribution and each learned distribution corresponding to the same value of z_t. Therefore twelve divergences are calculated, used as follows to create a divergence measure for each value of z_t.

$$D^L_t = \eta^L \left(D^{gL}_{\tau_s} - D^{bL}_{\tau_s} + D^{gL}_{\tau_l} - D^{bL}_{\tau_l}\right) \tag{21a}$$

$$D^N_t = \eta^N \left(D^{gN}_{\tau_s} - D^{bN}_{\tau_s} + D^{gN}_{\tau_l} - D^{bN}_{\tau_l}\right) \tag{21b}$$

$$D^T_t = \eta^T \left(D^{gT}_{\tau_s} - D^{bT}_{\tau_s} + D^{gT}_{\tau_l} - D^{bT}_{\tau_l}\right) \tag{21c}$$

where η is a constant which normalizes the measure by dividing by the divergence measured between $p(\bar{\alpha}^g)$ and $p(\bar{\alpha}^b)$. For example $\eta^N = D_{kl}\left(p(\bar{\alpha}^g)||p(\bar{\alpha}^b)\right)^{-1}$.

An overall divergence measure is given by

$$D_t = D^L_t + D^N_t + D^T_t \tag{22}$$

In this algorithm, mislocalization is detected by comparing these divergence measures to a threshold of zero. If the measure is above zero, the distributions $p(\bar{\alpha})$ are closer to the *bad* data than the *good* data, and mislocalization is likely.

The performance of this algorithm is shown in Figure 2.

VI. Relocalization

In this section, a novel method for recovering from failure of the previously described algorithm is given, based on the divergence measures described in the previous section. Two aspects are described: firstly a simple algorithm for varying the number of particles, and therefore the ability of the particle filter to model more broad distributions, based on the divergence measures; secondly, an algorithm for estimating where to sample particles in the case of mislocalization.

A. Varying the Number of Particles

This algorithm aims to increase the number of particles when the estimate is poor, temporarily increasing the computational cost, to decrease the severity of mislocalization.

A gain K_m is given by

$$K_m = \begin{cases} \left(1 + \sqrt{D_t}\right)^{\frac{M_0}{M_{t-1}}^2}, & \text{if } D_t > 0 \\ 1, & \text{otherwise} \end{cases} \tag{23}$$

Where D_t is the divergence measure described in section V, M_{t-1} is the number of particles at time $t-1$, and M_0 is the target low number of particles. The number of particles sampled at time t is given by

$$M_t = \begin{cases} K_m M_{t-1}, & \text{if } D_t > 0 \\ M_{t-1} + K_{m0}(M_0 - M_{t-1}), & \text{otherwise} \end{cases} \tag{24}$$

where K_{m0} gives the rate of decay of the number of particles.

Therefore, when the estimate is poor, $D_t > 0$, the number of particles increases. The exponent in equation 23 reduces the increase in number of particles as the number of particles increases, designed to avoid an exponential increase in the number of particles. When the estimate is good, $D_t < 0$, the number of particles exponentially decays towards the target number. The slow change in the number of particles gives the algorithm robustness to noise in the mislocalization estimate.

B. Multi-Hypothesis Filter for Metric-Topological Space

The typical method of using *mixture distribution sampling* for relocalization [25] is impossible due to limited measurements. Instead, an algorithm is used here which estimates the pose of the robot in parallel to the particle filter described in section IV. This *multi-hypothesis filter* algorithm is designed to be approximate, but robust to false positive and negative measurements, so that the overall algorithm should have improved robustness to this source of error. If other sources of error were to be considered, the *multi-hypothesis* algorithm could be modified to give robustness to the other sources too.

The proposed *multi-hypothesis* algorithm is based on the same particle filter framework described in section IV with some key modifications, keeping within familiar methods for particle filtering. Instead of using a large set of particles which is weighted and resampled at each time t, a small number of hypotheses $\mathcal{X}_t^{MH}$ with persistent weights over time is used, each representing a more distinct hypothesis. Like the particle filter algorithm, each hypothesis has a state $\mathbf{x}_t^m$ and a weight α_t^m. Measurements ξ_t^m, ζ_t^m, and ω_t^m can be made from each hypothesis.

For $z_t = L$, the hypotheses are each moved in linear motion similar to equation 15, but without added uncertainty. When a hypothesis m moves past the end of a link, instead of uniformly randomly moving to an adjacent link, new hypotheses are created in each of the l adjacent links $i \in \mathcal{L}^n$. Each is given a weight equal to α_t^m / l, and the weight of hypothesis m is set to zero.

For $z_t = N$, hypotheses are sampled at nodes near existing hypotheses. A probability p^n is found using the same function as equation 17, and the new hypothesis and original hypothesis m are given weights $p^n \alpha_t^m$ and $(1 - p^n)\alpha_t^m$ respectively. For $z_t = T$, hypotheses in nodes are weighted according to a similar expression to equation 19.

Unlike the particle filter, resampling is not done. Instead, the weights of the hypotheses are normalized so their sum equals one, and then hypotheses with a weight lower than a threshold, $\alpha_t^m < \alpha^0$, are removed. Therefore, the approximate poses with a likelihood greater than α^0 are tracked through this algorithm. The algorithm is more robust to false positive and negative measurements z_t than the particle filter, as the hypotheses are not weighted lower when they pass a node without making a measurement $z_t = N$, and hypotheses are not reduced in weight when they are in a link and there is a measurement $z_t = N$.

When $D_t > 0$ particles in the normal particle filter can be sampled at the set of approximate hypotheses, $\mathcal{X}_t^{MH}$, given by the multi-hypothesis filter. Up to M_D (typically a value of 10) particles are sampled at each hypothesis m, each with probability equal to $(\alpha_t^m)^2$ when $z_t = L$ and equal to $(\alpha_t^m)^{1/2}$ when $z_t = N$ or $z_t = T$.

Feedback from the particle filter to the multi-hypothesis is used so the two estimates do not completely diverge. If there is no hypothesis there already, one is created at each of the modes of the particle filter distribution. If each index contains M_t^i particles, these modes are measured as discrete indices i where M_t^i is greater than $\mu\left(M_t^{i \in \mathcal{I}}\right) + \sigma\left(M_t^{i \in \mathcal{I}}\right)$, the mean number of particles with each index plus the standard deviation of number of particles with each index.

The effect of this algorithm is that the approximate likely poses of the robot are tracked using an algorithm which is robust to error in measurements, and these hypotheses are used to improve the more precise particle filter estimate when necessary. The algorithm's use is visualized in Figure 1.

VII. Experiments

Experiments are done in simulation to evaluate the performance of two algorithms: the *1D* algorithm as described in section IV, and the *1D+* algorithm which uses the improvements described in sections V and VI.

The algorithm presented in this paper takes minimal measurements as inputs, in practice from a front-end feature detection algorithm using measurements from the robot's sensors. Figure 1 shows a trajectory estimated using the algorithm described in section IV from wheel encoder and IMU odometry and feature detection data from the SIAR dataset [26]. As the front-end is not investigated here, instead of limiting testing to a set of practical experimental data, simulation of a large set of data is done. This allows experimentation over a much larger sequence of inputs than would be possible experimentally, and good investigation of the effectiveness and efficiency of the algorithms.

The experiment here aims to evaluate the two algorithms over an increase in measurement uncertainty. The effectiveness of the algorithms is measured by the *error rate*, which is the proportion of time for which the error in the estimate is lower than a threshold 25 metres, and the *failure rate*, which is the proportion of trajectories where the error at the last 100 steps is over the threshold 25 metres, representing when the algorithm has failed and the robot could be considered *lost*. The computational cost of the algorithms is measured by the *number of particles* used by each algorithm.

Figure 3 shows the results over sets of data recorded for several robot trajectories through a pipe network. The performance of each algorithm on 50 trajectories of 1000 steps is shown for a target number of particles of 50, 100, and 200 particles. For 100 particles, the performance is shown for 5 trajectories of 10000 steps.

The number of steps is arbitrary, but important. If there is a chance of mislocalization at each time step, and a small chance of relocalization, then a longer trajectory is likely to show more error as the algorithm will eventually fail and not

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

TABLE II

Default parameters for the localization algorithms

	Parameter	Symbol	Value
1D	Motion model noise	σ_ψ	$1.2\sigma_x$
	Node transition std.	σ_n	2.8 m
	Angular weight std.	σ_ω	$10\sigma_\theta$
	Kernel std.	σ_g	25 m
1D+	Number of New Particles	M_D	10
	Number of Particles Decay Constant	K_{m0}	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

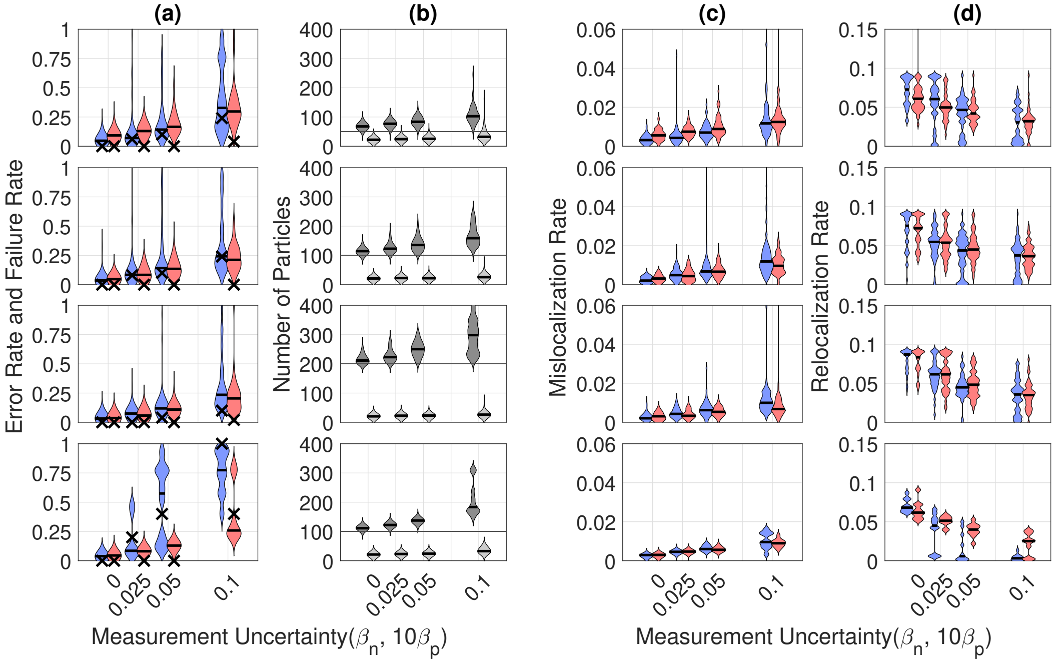

Fig. 3. Measures of performance and cost for the *1D* algorithm (particle filter in metric-topological space) and *1D+* algorithm (with detection of mislocalization and relocalization). For the error rate, violin plots show the normalized probability density, the black bars show the median over a set of tests. Black crosses show the failure rate. In all cases, blue is the *1D* algorithm and red is the *1D+* algorithm, and results are for varying uncertainty β. **(a)**: The error rate and failure rate. **(b)**: The number of particles used in the *1D+* particle filter algorithm (dark grey), and the *1D+* multi-hypothesis relocalization algorithm (light grey), for varying uncertainty β. The number of particles for the *1D* algorithm is shown by the horizontal line. **(c)**: The mislocalization rate. **(d)**: The relocalization rate. **Top to Bottom**: Each row shows results for: 50 particles over 50 1000 step trajectories, 100 particles over 100 1000 step trajectories, 200 particles over 50 1000 step trajectories, and 50 particles over 5 10000 step trajectories, respectively.

recover, while a short trajectory is more likely to avoid any error, and reduce the error rate. 1000 steps appears to balance these effects, but 10000 steps is also used for comparison. These values correspond to a total distance of 4.5 km and 45 km. In the pipe network used, the median pipe length is 64 m, so the robot travels through around 70 pipes on average over 1000 steps. Considering the trajectory lengths in terms of pipes, and therefore measured features, might help translate results from this experiment to other pipe configurations.

Error and *failure* rate are not the only options. Use of total absolute error is common in the literature, but in this case the absolute error depends on the network geometry as well as performance, so is not suitable. To address these problems, Figure 3 shows the *mislocalization* rate, the number of steps where the estimate becomes incorrect relative to the number of steps with correct estimates, and *relocalization* rate, the number of steps where the estimate becomes correct relative to the number of steps with incorrect estimates.

Figure 3 shows that for increasing measurement uncertainty (β), the error rate increases for both algorithms. For the *1D* algorithm, the chance of failure increases substantially, especially for the 10000 step trajectories, which is expected as a failure is followed by a longer period of error. The *1D+* algorithm is shown to be more robust; the failure rate remains low, mostly zero. Both algorithms are seen to be largely effective however, with low median rates of error.

The mislocalization and relocalization rates reflect these results. For both algorithms, with increasing β, the mislocalization rate increases but remains low, and the relocalization rate decreases. The *1D+* algorithm shows a slightly higher mislocalization rate; the algorithm's attempts at relocalization can cause mislocalization after a false detection. Relocalization rate is similar for the two algorithms, although the *1D* algorithm has more very low values, which correspond to failure, which the *1D+* algorithm avoids.

The computational cost for the *1D+* algorithm is larger than the cost for the *1D* algorithm. The two numbers of particles are summed to give the total number of particles used in the *1D+* algorithm. While the number of particles in the multi-hypothesis estimator is only around 25, this is substantial when $M = 50$. As β increases, M increases as the number of detected mislocalizations increases.

Overall, it can be seen that the *1D+* algorithm has better performance, but higher cost, than the *1D* algorithm. Whether this extra cost is acceptable depends on how much error is allowed, or the allowed likelihood of unrecoverable mislocalization. In practice, this would depend on the robot's parameters and other aspects of the robot's operation.

The algorithms presented here, including the novel means of detecting mislocalization and relocalization, prove to effectively estimate the pose of a robot in a network environment using a very minimal set of measurements, which could be usefully applied to simple robots or sensors moving in buried pipe networks. Filtering methods like those shown here, which recursively estimate the state using new information, may be limited in this problem definition, and substantial improvements might be more likely to be found using a smoothing or optimization based method using sequences of measurements to estimate a sequence of states.

VIII. Conclusions

This paper has presented two improvements to a particle filter algorithm which allow detection of mislocalization and relocalization in the case of minimal binary sensor observations, which is applied here to localization of a robot in a buried pipe networks, but could be applied more generally where measurements are limited. Error, or mislocalization, is detected by using the Kullback-Leibler divergence to compare sequences of observations, overcoming the problem of limited information in each individual observation. Recovery, or relocalization, is facilitated by describing the robot's pose as a hybrid metric-topological state. This reduces the state space in dimension compared to a typical two or three-dimensional state space, which allows approximate hypothetical robot poses to be tracked efficiently, and used to improve the particle filter estimate in the case of error. Results show that the improvements effectively reduce the rate of error and failure of the localization algorithm.

References

[1] 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

[2] 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

[3] 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

[4] 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

[5] 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

[6] 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

[7] 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

[8] 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

[9] 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.

[10] 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

[11] 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

[12] 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

[13] 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

[14] 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

[15] S. Kazeminasab and M. K. Banks, "A Localization and Navigation Method for an In-pipe Robot in Water Distribution System through Wireless Control towards Long-Distance Inspection," *IEEE Access*, vol. PP, p. 1, 2021. [Online]. Available: http://arxiv.org/abs/2105.10447

[16] S. Kazeminasab, V. Janfaza, M. Razavi, and M. K. Banks, "Smart Navigation for an In-pipe Robot Through Multi-phase Motion Control and Particle Filtering Method," pp. 342–349, 2021.

[17] 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

[18] 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

[19] 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

[20] 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

[21] S. Edwards, L. Mihaylova, J. M. Aitken, and S. Anderson, "Toward robust visual odometry using prior 2D map information and multiple hypothesis particle filtering," *Towards Autonomous Robotic Systems Conference (TAROS)*, pp. 1–5, 2021.

[22] 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

[23] 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

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

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

[26] 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.

[27] D. Fox, W. Burgard, F. Dellaert, and S. Thrun, "Monte Carlo Localization: efficient position estimation for mobile robots," *Proceedings of the National Conference on Artificial Intelligence*, no. Handschin 1970, pp. 343–349, 1999.