University of Sheffield

Robot Localization in Pipe Networks

Rob Worley

Supervisor: Dr. Sean Anderson

A report submitted in partial fulfilment of the requirements
for the degree of PhD in Engineering

in the

Department of Automatic Control and Systems Engineering

November 2, 2022

Declaration

All work contained in this thesis is entirely the author's own work, unless otherwise acknowledged.

Acknowledgements

I would like to thank Sarah for being my best pal for life, for having the most faith in me, and for making every day great.

I would like to thank all of my friends in (and formerly in) Sheffield for being in my life for the last four years. I also want to thank my friends from Swadlincote for being here (virtually) too, my friends from school now scattered around the country, my friends from university now scattered across the world, and my parents and family. Time spent with various combinations of these people walking in the peaks, playing D&D, at karaoke, on the PlayStation, playing board games, playing more D&D, travelling the world, and at the pub, has been invaluable in the production of this thesis.

Finally I would like to thank my supervisor Sean, and my colleagues Rui, Sarah, Mat, Jonathan, Mila, Yicheng, Gavin, Richard, Tanmay, Ashutosh, Will, and Kirill.

Abstract

Pipe networks transporting clean water and wastewater are critical around the world. This infrastructure needs regular inspection and maintenance to reduce the effect of faults on public health, the environment, and the economic cost of operation. These networks are made of pipes mostly less than 300 mm in diameter, buried beneath roads in lengths of around 50 to 200 metres. An autonomous robotic system could pervasively and persistently monitor the infrastructure from within the network. Robot localization, the ability of a robot to estimate its position in the environment, is required as it facilitates autonomous control and allows the localization of faults. The buried pipe environment constrains the robot's sensing, locomotion, and computation, so localization is difficult. Pose estimation using GPS and a magnetometer is unavailable, typical vision and rangefinding sensing is less effective in this environment than usual, and the robot's motion is more uncertain.

This thesis develops both the front-end perception and back-end state estimation. In contrast to approaches found in the literature, it is shown that a hybrid continuous-discrete approach to state estimation is well suited for localization in this application. For lower computational cost, this approach shows an average error rate of 0 for values of uncertainty larger than found in the literature, compared to an average error rate of around 0.25 for a typical approach. It is shown that acoustic echo sensing gives effective perception in this environment, adding to the literature a new means of observing features distant to the robot. Incorporating echoes with a novel localization algorithm gives an average error rate of 0 for larger values of uncertainty than found in the literature. These perception and estimation aspects are shown to be easily integrated, but also function well independently.

Ongoing research develops other aspects of a robot system for this application. The results presented here form part of this progress, informing the design of the overall system. More generally, these results provide some evidence that careful design of the localization system, from front-end to back-end, can provide better performance compared to a typical approach. In particular, the hierarchical approach used here which considers different levels of abstraction, scale, and precision, could be applied more broadly.

Contents

List of Symbols

t	time index, usually at a regular interval
τ	time index, at an irregular interval
k	time index within the short period of time used to make a measurement
T	time index at the end of a sequence of time indices
$\mathbf{x}_t$	the *state* to be estimated in localization, defined more specifically elsewhere, but generally related to the location of a robot
$\mathbf{x}_{0:t}$	the sequence of states to be estimated in localization
$\mathbf{u}_t$	a measurement of a robot's motion, relating the state at time $t-1$ and time t
$\mathbf{z}_t$	a measurement of a robot's surroundings, made at time t
x	a position in the x-axis of a Cartesian coordinate system, or the position along the axis of a location in a local coordinate frame
y	a position in the y-axis of a Cartesian coordinate system
θ	an angle about the z-axis of a Cartesian coordinate system
i	the index of a location in a discrete set of locations
d	the discrete direction in a location relative to a local coordinate frame
$\Delta\xi$, Δx	a measured forward linear motion relative to a robot
$\Delta\theta$	a measured angular motion relative to a robot
$p(\cdot)$	the probability distribution or probability density function of a random variable

$p(\cdot\|\cdot)$	the conditional probability distribution or probability density function of a random variable given another random variable
$p(\cdot,\cdot)$	the joint probability distribution or probability density function of two random variables
$b(x_t)$	the belief, equal to the posterior distribution $p(x_t\|u_{1:t}, z_{1:t}, x_0)$
$f_u(\mathbf{x}_{t-1}, \mathbf{u}_t)$	the probabilistic model of the state $\mathbf{x}_t$ at time t given the previous state and the measurement between the two states
$f_z(\mathbf{x}_t)$	the probabilistic model of the measurement $\mathbf{z}_t$ at time t given the current state
Σ_u	the matrix of uncertainty in the model $f_u(\mathbf{x}_{t-1}, \mathbf{u}_t)$
Σ_z	the matrix of uncertainty in the model $f_z(\mathbf{x}_t)$
w	a value of random noise drawn from a distribution with uncertainty given by Σ_u
v	a value of random noise drawn from a distribution with uncertainty given by Σ_z
Ω	the inverse of a matrix of uncertainty
m	a feature in the map $\mathcal{M}$
s	a measurable spatially varying property
ϕ	a function giving the difference between a function y which is a measurement of a spatially varying property, and f_s which is the expected value of a spatially varying property s
Σ_s	the matrix of uncertainty corresponding to the measurement of spatially varying property y
F_s	the Jacobian of the function f_s of expected value of spatially varying property
J	a cost function to be minimized, defining an optimization problem
$g_{\mathbf{x}}(k)$	acoustic transfer function
ξ_n	distance corresponding to component n of the acoustic echo response
μ	the mean of a normal distribution of the state estimate
$\mathbf{\Sigma}$	the covariance matrix of a normal distribution of the state estimate

μ	the mean of a distribution
σ	the standard deviation of a distribution
Λ	the length of a pipe
$\hat{\Lambda}$	an estimate of the length of a pipe
$\mathbf{x}^h$	the state estimate for particle h in a particle filter
α^h	the weight for particle h in a particle filter
$\mathcal{X}$	the set of state estimates in a particle filter
$\mathcal{A}$ or A	the set of weights in a particle filter
c	the estimated correspondence between features in the map
Ω	information matrix
ξ	information vector
m	measured distance travelled
o	an observation of the number of exits from a junction
a or u	the relative discrete direction that the robot takes
ϵ	error
K	time taken
$\mathcal{I}$	indices of discrete locations
$\mathcal{N}$	indices of discrete nodes
$\mathcal{L}$	indices of discrete links
u_x	magnitude of uniformly distributed noise
β_p	false positive probability
θ_{fp}	false positive number
$\tilde{w}$	sampled noise for the particle filter to estimate w
τ	set of recent times t
D_{kl}	Kullback-Liebler divergence
Υ	the set of paths between places in a network
υ	a path between places in a network
$\mathcal{T}_t$	time indices
$\mathcal{T}_\tau$	time indices

Chapter 1

Introduction

1.1 Context and Motivation

1.1.1 Buried Pipe Networks

Water distribution and wastewater infrastructure is crucial for the supply of clean water and the safe transport of wastewater in urban and rural areas around the world. This critical infrastructure is ageing, with many networks now coming to the end of their expected lifetime [1], and is therefore in constant need of costly inspection and maintenance for reducing impact on public health, pollution, and disruption to services, as well as asset management [2]. This is made difficult by both the size of the pipe networks and the inaccessibility of buried pipes. In the UK, there is around 400,000 kilometres of water supply pipes, and 400,000 kilometres of sewer pipes [3], which is the result of 250 billion pounds of investment [4]. In Germany there is around 600,000 kilometres of sewer pipes in total [5], and in the US the figure is around 2,000,000 kilometres [6]. In the UK, over 3000 million litres of water is lost to leaks every day from water distribution pipes [7], and faults in wastewater pipes can cause pollution and danger to public health, with around 50 serious pollution incidents per year in the UK [7].

Wastewater pipe networks are complex branching systems [2], where pipes connect

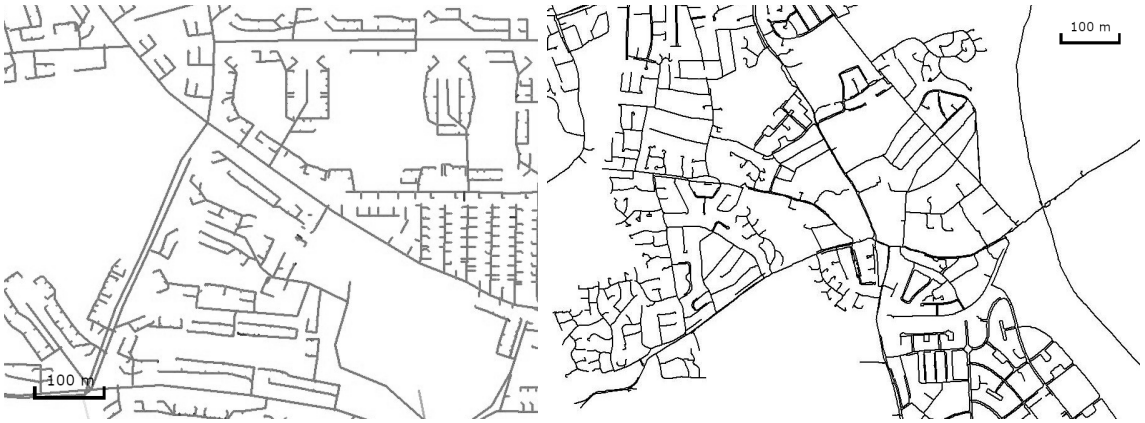

Figure 1.1: *Example maps of buried pipe networks in two parts of the UK*[1]*. Left: A wastewater network. Right: A water supply network.*

individual properties together, and in turn larger pipes connect groups of properties together. Main pipes which are typically buried beneath roads are connected together at *manholes*, and along each main pipe will be *lateral connections* to pipes from properties along the road. Water distribution pipe networks comprise of pipes connecting sources to consumers [1], built with a looped and branched configuration [8], where connections to consumers are made at arbitrary points along the pipeline [8]. While these general configurations can be expected, both types of network will vary considerably in specific construction from location to location. These networks function to move fluid from one location to another. Water distribution networks move water in flat terrain by pressurisation [8] so the pipes are filled with liquid, while water is moved in wastewater pipe networks by gravity with liquid filling varying levels of the pipe cross-section, from empty to full [2]. Overall the two types of network share a number of characteristics, including network structure, scale, function, and conditions both inside and outside of the pipes.

Inspection and maintenance of water distribution and wastewater pipes is challenging. Inspection from within the pipe is desirable as investigation can be done while avoiding disruption from excavations [2], but is difficult due to the dimensions of the pipes. These dimensions are investigated by analysis of pipe data from an area in the UK[2]. This analysis is based on a network which covers an area around 3 km in dimension, which contains around

[2]The data for this analysis were provided by Will Shepherd.

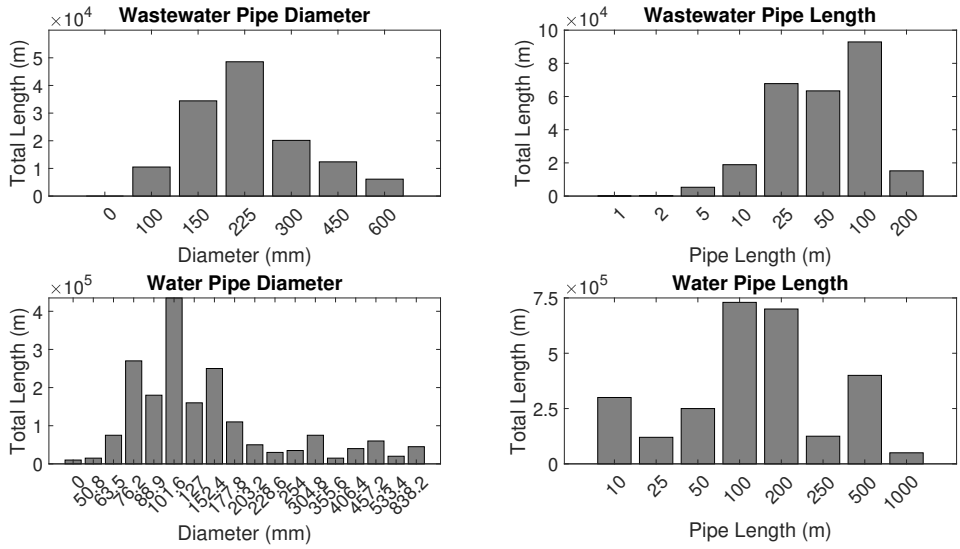

Figure 1.2: *Results from analysis of pipe network data for a small part of the UK. The bar charts show the total length of pipe in the area which is made of pipes of the given length or diameter. Top: Wastewater pipe data. Bottom: Water supply pipe data.*

300 km of pipes in total length.

Figure 1.2 shows the diameter data from this analysis. From this, it can be seen that most of the total length of wastewater pipe has a diameter of 150 mm to 225 mm, while the most common water supply pipe diameter is around 100 mm. These figures are similar to those found in the literature, where it is described that 80% of water supply pipes are less than 150 mm in diameter [1], and 70% of wastewater pipes are less than 300 mm in diameter [2]. This small size limits the size of sensors which can be moved through the pipe.

Figure 1.2 also shows the length data from this analysis. The length of pipes between above-ground access points varies; 85% of wastewater pipes in the UK are around 25 and 100 metres long and 10% are around 5 to 10 metres long, while around 50% of water distribution pipe networks are made of pipes around 100 and 200 metres long. This is supported by the literature, where it is reported that construction standards in the UK for example require above-ground access points to wastewater pipes to be placed at least every 90 metres or 200 metres, depending on the diameter of the pipe [2], for example. Sensors must be able to observe pipes of these lengths while reporting the location of detected faults to a sufficient

precision to allow maintenance.

Maintenance is done by excavating the ground above the pipe. While consistent guidelines for this maintenance are not available, excavation is done using both mechanical excavators and hand tools [9], and excavation will therefore be at the scale of around 1 metre in dimension. This is consistent with the precision expected for the localization of buried pipes according to the British Standards Institution's (BSI) *Publicly Available Specification PAS 128* [10], consistent with example legal requirements for a minimum of 0.5 to 1 metre wide excavations [11], and similar to values from construction standards, which require an excavation width of 1.5 to 3 times the pipe diameter when constructing a wastewater pipe [2]. These expected excavation dimensions can be used to give a target for acceptable fault localization precision.

1.1.2 Pipe Inspection with Robots

Autonomous persistent monitoring of this difficult to access infrastructure could be done using robots, which could pervasively inspect a network of pipes from within the pipes. This could replace current methods which either involve manually controlled robots which have to be physically tethered to the control system at the point of deployment, or involve costly sensing from above ground using technology such as ground penetrating radar [12]. General methods for pipe inspection, including robotics, have been reviewed in the literature [13], where it is concluded that as technology for this application matures, inspection of even relatively low-risk parts of pipe networks will become justified. In the decade since this review, robot technology for this application has matured [14], increasing the likelihood of successful application. The use of robotics for pipe inspection in application to water, wastewater, oil, and gas, has been reviewed elsewhere in more detail [15]. There, it is concluded that autonomous, untethered robots could extend the range of inspection beyond what is possible with existing methods with and without robots, and concluded that highly capable robots are likely not feasible in this environment due to its constraints and size, and instead a system of several less powerful robots is more realistic.

Tables 1.1 and 1.2 give a nonexhaustive list of robots designed for operating in pipes, importantly noting whether they are designed to work in air-filled, water-filled pipes, or both. The diameter range that the robot can operate in is given if available. The type of locomotion, and the sensors available on the robots are listed, and the level of autonomy is described. Table 1.1 lists robots described in academic literature, while 1.2 lists commercially available robots.

It is seen from this that robots exist for application both in air-filled and water-filled pipes, at diameters ranging from 0.05 metres to 0.9 metres. A range of locomotion methods is seen, reviewed elsewhere in more detail [16], which increases the range of pipe network types and configurations which can be traversed by robots. The sizes of developed robots for this application can be compared to the typical diameters of buried pipes, where it is seen that robots exist for this application, however many existing solutions are too large for a substantial amount of pipes, so further miniturisation, and therefore limitations on function, will be needed for full inspection of a pipe network.

1.1.3 Robot Localization in Pipes

Whatever form the robot system takes, robot localization is an essential component when autonomously inspecting buried pipes [14]. Robot localization is the problem of using sensory perceptions of a robot's motion and environment to make an estimate of the robot's state with respect to the environment [30]. Its output is useful for large-scale navigation, small-scale control, for mapping an unexplored environment, and for locating the robot within an environment. In application to pipe inspection robots, all of these functions are required in different parts of the robot operation.

Typically, sensors are needed to improve upon *odometry* (or *dead reckoning*) where only a measurement of the robot's uncertain motion is used to estimate the robot's position, which is subject to accumulating uncertainty, or *drift* [14]. In this application, the robot's motion is expected to be more uncertain than in typical applications, due to challenges in traction [16] from uneven surfaces, slippery surfaces, inclined surfaces, and fluid flow.

Name	Pipe Type	Diameter (min,max) (m)	Autonomy	Locomotion	Sensors
MOGRER [17]	Air	0.13, 0.22	-	Wall-pressing	-
FERRET-1 [18]	Air	0.09, 0.12	-	Wall-pressing	-
KURT [19, 20]	Air	0.3, -	High-level control	Wheels	Camera, radio IMU, rangefinder
Makro [21]	Air	0.3, 0.6	High-level control	Wheels	Cameras, rangefinders
PIRAT [22]	Air	0.6, -	Remote control	Tracks	Laser scanner, camera, IMU, sonar
MRINSPECT VI+ [23]	Air	0.11, 0.11	-	Wall-pressing	Camera
Moritz [24]	Air	0.6, 0.7	Low-level control	Legs	Encoders
Kantaro [25]	Air	0.2, 0.3	High-level control	Wheels	Camera, laser scanner
MagneBike [26]	Air	-	Low-level control	Magnetic Wheels	Camera
MIT-MRL [27]	Water	0.1, -	-	Swimming	-
PIPETRON I, II [28]	Air	0.08, 0.08	Remote control	Wall-pressing	Camera, eddy current, laser projector
PIPETRON VI [28]	Air	0.1, 0.1	Remote control	Wall-pressing	-
PIPETRON VII [28]	Air	0.15, 0.15	Remote control	Wall-pressing	-
Daisy [29]	Water	0.052, 0.052	Passive	Flow	Pressure

Table 1.1: *A list of robots for pipe environments developed for academic research.*

Name	Pipe Type	Diameter (min,max) (m)	Autonomy	Locomotion	Sensors
KA-TE	Air	0.1, 0.8	Remote control	Wheels	Camera
SSET	Air	-	Remote control	Tracks	Camera, scanner, IMU
IBAK	Air	0.15, -	Remote control	Wheels	Camera
VersaTrax	Either	0.2, -	Remote control	Tracks	-
TriTrax	Either	0.2, -	Remote control	Wall-pressing	Camera
Magg	Either	0.3, -	Remote control	Tracks	Camera, magnetic
LineTrax	Either	0.05, 0.15	Remote control	Wall-pressing	Camera
SmartBall	Water	0.15, -	Passive	Flow	Passive acoustics, IMU
Sahara	Water	0.25, -	Passive	Flow, tethered	Passive acoustics, camera, radio
DT320 Mini	Either	0.15, 0.45	Remote control	Tracks	Camera
Pipe Trekker A-200	Either	0.9, -	Remote control	Wheels	Camera

Table 1.2: *A list of robots for pipe environments developed for commercial application.*

Table 1.3 shows the accuracy of linear position estimate using odometry recorded in the literature on robot localization in pipes. Robots in pipe networks will mostly travel on straight paths through pipes. The error in the estimate of linear position is therefore a useful quantity to motivate improvements to robot localization. The full range of literature on robot localization in pipe networks is reviewed in Chapter 2, while here only results where the linear odometry estimate error can be measured are compared. The error in *odometry* estimate as a proportion of the distance travelled reported ranges from 0.8% to 35%, in different environments, using different robots, and in different experimental conditions. Relating to the target of 0.5 metres for localization precision described previously, an error of 5% of distance travelled means that the estimate would likely be outside of the target precision after 5 metres of travel, which is much shorter than the typical wastewater pipe length of around 50 metres described previously.

The pipe conditions of each of these estimates are noted in the table. The dry pipes, especially those in a laboratory setting, are expected to produce less uncertainty than pipes in a field setting. It is not possible without further experimentation to directly estimate the uncertainty in robot motion due to the various parameters describing a pipe network's oper-

Paper	Error (lower, upper bound) (% of distance)	Pipe Diameter (m)	Pipe Length (lower, upper bound) (m)	Pipe Type
[31]	5, 35	0.15	30	Dry, Lab
[32]	-, 15	0.9	3000	Fluid, Field
[33]	7, 18	0.088	5	Water, Simulation
[34, 35]	3, 15	0.2	4, 20	Dry, Lab
[36, 37]	-, 5	unknown	200	Water, Simulation based on Field
[38]	5, 7	4	60	Dry, Field
[39]	0.8, 1.6	0.6	50	Dry, Lab

Table 1.3: *A comparison of estimates of linear odometry error taken from the literature.*

ation. From the literature however, wastewater pipes are often specified to have a minimum flow velocity of 0.7 or 1 metres per second, and a flow velocity of 3 metres per second is considered high [2]. Permanent sediment deposits are present in 80% of wastewater pipes, and effective surface roughness can become as large as 10% of the pipe diameter [2], which would reduce the reliability of locomotion [21]. Clean water distribution pipes would be expected to be clear of substantial obstacles such as sediment, however the flow of water in the pipe would be expected. The small pipe diameter in water distribution networks places further limitations on robots, such as a reduced energy capacity. Robots which move to some extent passively with the flow of water have been developed for energy efficiency [29], but would suffer further uncertainty in motion relative to the static pipe.

Robot localization in pipes has been reviewed elsewhere [40], where it is seen that vision, odometry, and inertial measurements are the most common means of sensing. Vision is a well established sensing mode for robot localization generally, and algorithms can be easily deployed on vision data to assess the effectiveness in this application. From Table 1.2, cameras are seen to be common sensors for commercially available remotely controlled robots, indicating the usefulness of camera data for human operators.

Figure 1.3 gives a set of example images that have been captured in a pipe using a

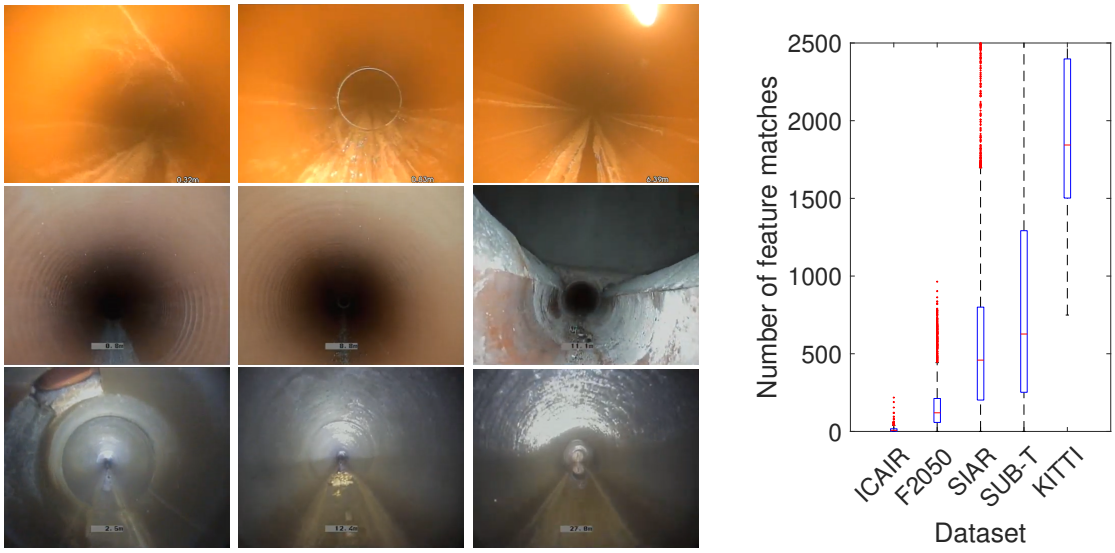

Figure 1.3: *Left: A set of example images that have been captured in buried pipes. Right: A comparison of the number of ORB feature matches in different datasets with similar image resolution: KITTI [41], SUB-T [42], and SIAR [43], and buried pipes (F2050 and ICAIR).*

camera. A *front-end* process would take these raw images as inputs, and produce some kind of measurement or observation which can be used in the *back-end* process. Some images contain detectable features such as a joint between two sections of pipe and a connection to a manhole, which could be used as observations. Most of the images contain some visual texture which might appear similar in each image but could be used to recognise a previously observed location, making a loop-closure measurement, or could be used to make a visual odometry measurement. These images also serve to illustrate some of the challenges in localization in a pipe network; the similarity between locations, relatively feature-sparse images, and restricted scope of vision can all be seen.

Part of the challenge of applying vision sensing to localization in the pipe environment is the feature sparseness compared to other environments. To quantify the challenge, this lack of features can be measured for a range of robot vision datasets. *ORB* features [44] are detected in images and features are matched between consecutive images, which is part of a typical vision-based approach to robot localization. The number of feature matches between consecutive images is compared for different datasets in Figure 1.3. The *KITTI* dataset [41] consists of vision data recorded from a road vehicle moving around both urban

and rural roads, as well as data from a range of other sensors, and represents a typical environment for which robot localization has been developed. The *SUB-T* dataset [42] used here consists of data recorded by a robot during the *DARPA Subterranean Challenge*, which is expected to be more challenging than typical environments due to poor lighting and limited visual structure in the environment. The *SIAR* dataset [43] consists of data recorded by a robot in a sewer tunnel environment, representing data from confined spaces similar to that of the buried pipe environment. The *F2050* and *ICAIR* datasets have been recorded using a remotely controlled robot in buried wastewater pipes[3], giving two examples of the target application.

Figure 1.3 shows the comparison between the number of feature matches between these five datasets. The *KITTI* dataset has a median of around 1800 feature matches between consecutive images, illustrating the feature density of a typical outdoor robot environment. The *SUB-T* and *SIAR* datasets have a median of around 500 features matches between images, indicating an environment which is substantially more challenging for vision-based localization than typical environments. The *F2050* dataset has a median of 120 feature matches between images, while the *ICAIR* dataset has a median of 7 feature matches. The buried pipe environment has a reduction in this measure of vision sensing quality of one or two orders of magnitude compared to typical robot environments and even other confined underground environments. The buried pipe environment also therefore has a range of quality of vision-based sensing. This quantifies the challenge in applying typical robot localization methods to the pipe environment.

Another part of the problem with vision is the reduced perspective, in that all perceivable visual features are positioned ahead of the robot, and there is a small parallax angle between distant features. This can lead to a problem of an apparently shrinking environment [45]. The limited perspective puts a limit on the number of *loop-closures* that can be made. Further problems with vision sensing in the pipe environment include extreme lighting variations between images [46], and visual aliasing, where visually similar features are observed in different locations.

[3]These data were recorded by Mathew H. Evans.

1.2 Problem Statement

Given the above motivation and context, a problem statement can be made which defines the constraints of the research area for this thesis. The aim of this thesis is to develop localization for robots in water supply or wastewater pipe networks, which would be used for autonomous and pervasive inspection of the pipe networks.

Robots in this environment will be constrained by the pipes of diameter 50 mm to 200 mm, and by the need to operate in water. Uncertainty in motion will come from fluid flow and from rough surfaces, with linear position estimate error of up to 35% of distance reported in the literature.

Robots will need to navigate, requiring position estimation to the precision of a single length of pipe. Robots will need to estimate the location of detected faults for subsequent maintenance, requiring position estimation to the precision of around 0.5 metres, which is the scale at which maintenance excavations will be made. This precision is required while the robot travels around 50 m to 200 m between junctions in the network. Network size will vary from location to location, but a network might cover an area with dimensions of several kilometres, with a total length of pipes of hundreds of kilometers.

The variation in dimensions in this environment is larger than in typical robot localization problems, and covers several orders of magnitude from the pipe diameter, to pipe length, to total network length. Figure 1.4 illustrates the different levels of scale in this problem, and how they link to perception, state estimation, and control. This thesis develops approaches for the medium-scale estimation of *position in pipe axis* and large-scale estimation of *location in network*.

1.3 Aims and Objectives

The aim of this thesis is to improve methods for robot localization in a pipe network environment, in terms of accuracy, reliability, and efficiency. The brief investigation above shows the

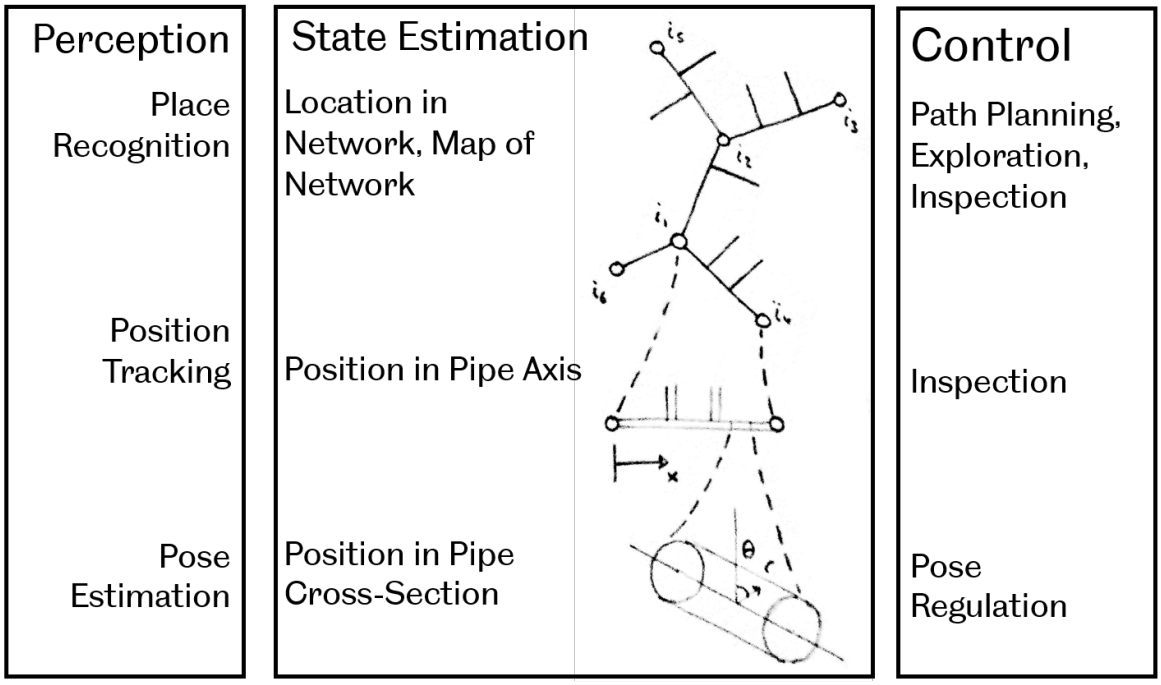

Figure 1.4: *An illustration of the different scales of localization in this problem.*

challenges for perception in the pipe environment, and the limited estimate accuracy found in the literature.

To achieve this aim, the following objectives must be met.

1. Investigate the limitations of existing approaches to localization when applied to the pipe environment, quantifying the effect of difference sources of uncertainty on performance, which will inform the subsequent developments.

2. Investigate how to improve robot front-end perception in the constrained pipe environment, achieving a low rate of error in robot localization estimate along the length of a pipe. This must be over a scale of around 100 metres, with motion uncertainty greater than 35% of the distance travelled.

3. Investigate whether it is possible to improve the back-end estimation over a pipe network by using the constraints of the environment, measured by an improvement to robustness and computational cost. This must be done in a network of pipes covering an area

several kilometres in dimension.

4. Investigate the effect of different front-end perception methods in the pipe network environment, integrating with a back-end localization algorithm at the large network scale.

1.4 Contributions

This thesis presents four key contributions to the literature.

Chapter 4 presents further development of the use of a hydrophone sensor for robot localization in pipes. A novel augmented pose-graph optimization algorithm is proposed and shown to give improved performance than alternative methods in terms of accuracy and efficiency.

Chapter 5 presents the novel application of acoustic echo sensing to robots in pipe networks, and presents a novel algorithm required for processing the sensor data in the challenging environment which is shown to improve results.

Chapter 6 presents an application of the *hybrid continuous-discrete* state representation to localization in pipe networks, and develops on the typical approach with two novel improvements to robustness required in this environment.

Chapter 7 presents a novel localization algorithm developed for the challenging pipe environment, which shows an improvement over previous algorithms in terms of robustness and efficiency.

Chapter 2

Literature Review

2.1 Introduction

In this chapter, three main aspects of the literature relating to this thesis will be reviewed. Firstly, the literature regarding different *state definitions*. Secondly, the literature regarding *state estimation*, which will cover the theoretical estimation formulation and practical implementations of filtering and smoothing. Thirdly, the literature regarding sensing as related to localization.

In each section, the literature in general robot localization will be discussed first, followed by a review of the literature specific to localization of robots in pipes or other similar applications.

2.2 State Definition

Robot localization can be considered as a *state estimation* problem, where variables describing a dynamic state are estimated using measurements over time. This section will review different definitions of this state in the literature. Defining the state to be estimated is the task of specifying the exact question asked to the localization system. For example: *Where*

is the robot now, in relation to its initial position, in [x, y] *coordinates? Where has the robot been previously, and what are its surroundings, relative to a map of the world? Which area of this map is the robot in?* Each question would lead to a different state definition, likely lead to a different set of requirements for sensing, and lead to a different localization algorithm.

A robot application is likely best served by an answer to a particular question. An aerial robot mapping a small, unknown environment could understand where it is in relation to obstacles in the environment in an arbitrary six degree-of-freedom coordinate system. The *state* would be the robot's pose and positions of obstacles in three Cartesian axes and the rotations about these axes. This state definition allows the flexible mapping of an environment which could contain any arbitrary objects, and would facilitate obstacle avoidance from the control system. A robot operating on roads in a city could understand where it is in terms of a known road map, so the *state* could be a discrete index describing which road the robot is on, with continuous variables describing the position and velocities of the robot, vehicles, and pedestrians with respect to the centre of the road.

When defining a robot's state, consideration should be given to how the environment is understood in relation to the robot. The state definition will depend on how it will be used by the robot, and how it will be estimated by the robot. This is true both for the spatial and temporal aspects of the state. The problem statement in Chapter 1 describes two uses of robot localization: to allow autonomous navigation, and to allow estimation of the location of a fault in the pipe network. For navigation, a robot would need to estimate its location to the precision of a single discrete pipe of junction, while usefully localizing a fault requires a high precision of around 1 metre. For a robot to navigate towards a particular point in a pipe network it may only need to estimate its current position; while for a robot to make navigation decisions regarding inspection of a pipe network it may need to estimate where it has been previously. For a robot to navigate it might need to understand the connectivity between places in the environment, while to report the location of a fault in the pipe network it might need to translate the location to global coordinates.

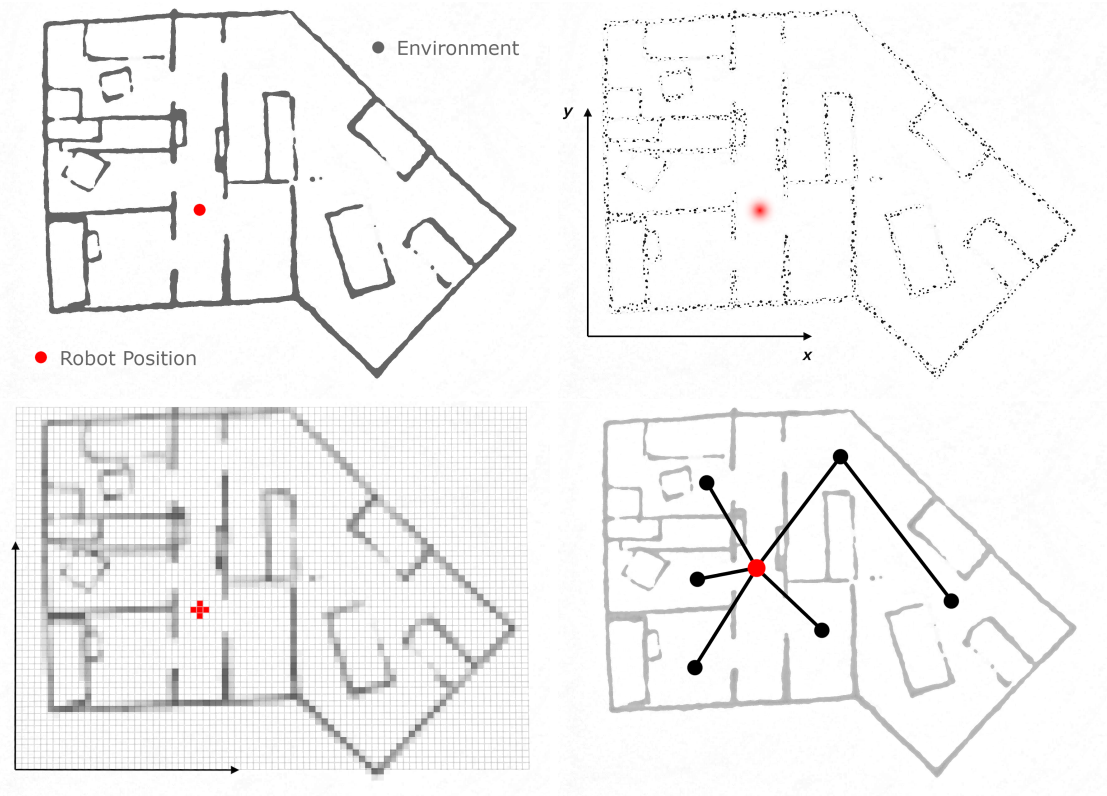

Figure 2.1: *Illustrations of different representations of the robot state and map state.* ***Top-left****: the true map.* ***Top-right****: the map as a set of detected points in an arbitrary Cartesian coordinate system.* ***Bottom-left****: the map represented by a metric occupancy grid.* ***Bottom-right****: a topological map.*

2.2.1 Discrete Representation

A continuous space can be decomposed into a set of discrete, finite sized cells. The state could then be defined as a discrete variable denoting which of the set of cells the robot is in. Some early work in robot localization uses discrete representations of the environment defined by distinctive places in a structured indoor environment [47, 48] or segments between distinctive places, [49, 50]. However, works using a discrete representation of the space diverged: some representations becoming known as *topological* maps, which to use a more coarse representation of the environment, such as dividing the environment into regions separated by narrow passages [51] or into lines equidistant from two objects in the environment [52]; and some representations becoming *metric grid* maps which use a finer grid representation which is independent of the environment [53]. The latter approach can allow the state definition to extend to the position of features in the environment as well as the position of the robot.[1] These two representations are illustrated in Figure 2.1.

In much of the existing work using a *topological* representation, the focus is on obtaining a topological map from metric sensor data, which is less challenging in a pipe environment because it is well described by only a topological map. Therefore, the methods used in localization in a topological map could be easily and usefully applied to the pipe environment. This representation has been applied to localization in small pipe networks [20, 54] where the pipes are connected orthogonally at junctions, however, there is no attempt to understand the effect of uncertainty on this approach in pipe networks in the literature. Recent work using a *topological* space more generally adds the challenges of erroneous repeated observations of the environment at a topological map node, inclusion of information assigned to nearby nodes, and failing to make an observation at a node [55]. Use of geometric information on the robot's orientation has been applied using prior knowledge of the

[1] It should be noted that, as described here, the literature tends to use the term *discrete* to refer to both *topological* and *metric grid* approaches. This may cause confusion where the estimation of *continuous* variables is often referred to as *metric* localization. A better terminology might be to use *discrete* to refer to quantitative variables such as position in an arbitrary grid map, and to use *categorical* to refer to qualitative variables such as one describing the room in a building in which a robot is located. However, this thesis will continue to use the terminology preferred in the literature.

orientation between two topological map nodes [56]. Solutions to challenges such as missing observations are especially applicable where a robot has limited sensing ability as is the case in a pipe environment, and these developments from more general environments could be usefully applied here.

When using a *topological* representation, the state might be precisely defined as

$$\mathbf{x}_t = \Big(i_t, d_t\Big) \tag{2.1}$$

where i_t is the discrete index of the location of the robot, and d_t is the discrete direction of the robot. In a pipe network, each discrete place could be a junction, and each discrete direction could relate to an adjacent pipe.

The *grid* representation allows a flexible representation of the position of features in the map and the position of the robot, however, there is an inevitable loss in precision due to the discretisation of the map. Mitigating this loss of precision can be done by reducing the size of the grid cells, and memory efficient solutions to storing grids have been developed for both two-dimensional [57] and three-dimensional maps [58, 59] to avoid requirements for large amounts of memory. For robots in pipes, the fixed grid approach has more flexibility than necessary, as the environment is expected to be made up of only a small range of components. Unlike common indoor environments for robot operation, which contain many orthogonal walls, the pipe environment is not expected to conform to a grid representation, so a higher resolution may be needed to usefully describe the environment.

2.2.2 Continuous Representation

The use of continuous variables to represent a robot's position was developed early in the field of localization [60, 61]. Filtering methods were later developed which could extend the use of continuous variables to any arbitrary distribution of possible robot positions [53]. In these early works, the robot's state was defined as

$$\mathbf{x}_t = \Big[x_t, y_t, \theta_t\Big] \tag{2.2}$$

representing the robot's position in a two-dimensional Cartesian coordinate frame, and its orientation in rotation about the vertical axis. This could apply to a buried pipe network environment, as networks exist in relatively flat urban and rural areas, and will extend over a far larger area in the horizontal plane than in the vertical axis.

Later development has shown use of a range of state definitions. Some typical state variables are the robot's *pose* (position and orientation) in Cartesian coordinates relative to a global coordinate frame, the *configuration* of the robot's joints (such as the joints in a manipulator arm), the robot's velocity (in terms of pose and configuration), the location and description of objects in the environment, and the velocities of these objects [30]. As well as the use of Euler angles (roll, pitch, and yaw), rotation is often represented by rotation matrices in the Lie group $SO(3)$ [62] (the special orthogonal group of rotations in three dimensions) which can be extended to homogeneous transformation matrices, and by unit quaternions [63].

Variables describing the robot's environment can similarly be described by continuous variables. *Metric map models* have been reviewed elsewhere [64] in detail. Two-dimensional map definitions are mature enough that standards have been developed [65]. The existing representations can be described as sparse landmark-based representations, dense raw data-based representations, and representations of surfaces, volumes, and higher-level objects. The drawbacks of these representations is that they don't give any higher level understanding of the environment and they can inefficiently use a lot of variables to describe a simple environment. This representation is illustrated in Figure 2.1.

The issue of inefficiency is especially problematic in the pipe environment, where power and space for processing and storing data is limited, and where the environment is made up of relatively simple shapes which could be represented simply without much loss of accuracy. Despite this, the continuous variable representation has been applied in robot localization in pipe networks. One-dimensional states have been used in single straight pipes [33, 38], which is seen to be effective. Two-dimensional (three degrees-of-freedom) states [66], and three-dimensional (six degrees-of-freedom) states [67, 68, 69] have been used in pipe networks, however these approaches are seen to require additional computation to constrain

estimates to the lower dimensional space of the pipe network, or additional computation is required to give a useful understanding of the network [70]. Three-dimensional states have been used in long single pipes with corners [35, 32], where the substantial mathematical complexity required to incorporate information from different sensors is seen. Similarly, a three-dimensional state has been used in a single curved pipe [31], where it is seen that the position of the robot is unable to be directly estimated in this three-dimensional state using a tether cable which only measures the one-dimensional distance travelled.

2.2.3 Hybrid Continuous-Discrete Representations

One final type of representation can be described, combining continuous metric variables with discrete topological variables [71]. In this definition, the robot and its surroundings are represented simultaneously in a high-level topological map describing different discrete places of the environment and in a low-level metric map describing the continuous space within the discrete place. Early work using hybrid continuous-discrete representations was done in application to typical indoor environments [72, 73, 74] and in network-like environments [75], the latter of which is especially applicable to the pipe network environment.

A continuous-discrete state might be defined as

$$\mathbf{x}_t = \Big(i_t, x_t, y_t, \theta_t\Big) \tag{2.3}$$

where i_t is the index of the discrete location of the robot, and x_t, y_t, θ_t is the position and orientation of the robot in a two-dimensional plane within location i_t. In a pipe environment, the state might instead be defined as

$$\mathbf{x}_t = \Big(i_t, d_t, x_t\Big) \tag{2.4}$$

where x_t is the position in the one-dimensional axis of a pipe with index i_t, and d_t is the discrete orientation of the robot along that pipe, where it could be oriented in either of two directions. A more detailed state definition in a pipe environment might be given by

$$\mathbf{x}_t = \Big(i_t, x_t, r_t, \varphi_t, \theta_{x,t}, \theta_{r,t}, \theta_{\varphi,t}\Big) \tag{2.5}$$

where the continuous state within discrete location i_t is described in cylindrical coordinates relative to the pipe i_t with axial position x_t, radial position r_t, azimuth angle φ_t, and rotations about each of these three axes $\theta_{x,t}$, $\theta_{r,t}$, $\theta_{\varphi,t}$.

Recent work has applied this state definition to localization of vehicles in road networks [76, 77, 78]. The pipe environment has a number of similarities to the road network environment, so there may be some use in these approaches. However there are a number of large differences including the computation and sensing available for road vehicles in comparison to robots in pipes, and the relatively good quality maps of road networks in comparison to the often poorly mapped pipe networks.

It is recognised in the literature that the hybrid continuous-discrete representation has advantages over the continuous representation when sensory information is unreliable, costly, or unavailable [75], which is the case in the pipe environment, and in terms of efficiency [76]. The literature describes how the hybrid representation allows correction of the robot path without rebuilding a large continuous space map, and difficulty representing the robot state accurately when it is far from the continuous coordinate origin [71]. It is also described in the literature how hybrid continuous-discrete maps are advantageous over continuous maps as they integrate easily with path-planning [71, 70]. In the pipe environment, all of the advantages of the hybrid representation are available, and because the pipe network is well described as a set of simple discrete places the general disadvantages of the representation, such as loss of precision, are mitigated.

2.2.4 Comparison of State Definitions

Various state definitions have been considered: discrete, continuous, and a hybrid discrete-continuous approach. While much of the literature focuses on estimating a continuous state, the limitations of this approach in this application are described in Section 2.2.2, and Section 2.2.3 describes the advantages of alternative approaches in this application.

Further motivation to investigate alternative approaches comes from the specification of inputs and outputs of the localization function in this application. The problem

definition in Chapter 1 describes how the robot needs to estimate its position to the precision of a single pipe for the purposes of navigation, and needs to estimate the position of a fault along the length of a pipe to the precision of around 1 metre to facilitate maintenance.

In typical robot applications, such as the indoor built environment, tasks such as navigation and precision localization in the environment are done at the same approximate scale. Through its sensors, a robot can perceive a large amount of the environment including several rooms. The obstacles in the environment are of a similar scale to the rooms in the building, so the robot's navigation can be done all at the same scale, with path planning and obstacle avoidance achieved by the same system.

The pipe environment however, is very different. As described in Chapter 1, the pipes are approximately 0.3 metres in diameter, around 50 to 100 metres long, and are in a network covering a scale of several kilometres, so there is a difference of as much as two orders of magnitude between the scale and scope of each aspect of the environment. Control regulating the robot's pose at the scale of the pipe cross-section is completely separate from control for path-planning at the scale of a town. Typical sensors which can perceive nearby features of the pipe are limited in scope and cannot perceive features further away.

There is therefore the motivation to develop a localization algorithm, connecting perception to control, which acknowledges this natural variation in scale of the environment. A state space defined as a set of discrete places each with its own continuous space, the *hybrid discrete-continuous* approach, is therefore a sensible approach to take.

2.3 State Estimation

The state must be estimated given some prior information, and a sequence of uncertain inputs and measurements. Acknowledging the unavoidable uncertainty in the measurements, the question asked of the state estimation system can be framed as: *What is the most likely state?* or *What are all of the values the state could likely be, and what is the probability that the system is in each state?* Relating to the previous section, exactly what the state is best

defined as varies depending on the application, and it might include simply the position of the robot, or the full trajectory of the robot and the positions of all features it has observed in its surroundings. In this section, the literature on solutions to this state estimation problem will be described.

First, consideration must be given to what is being estimated. Acknowledging the uncertainty in the measurements means also acknowledging the uncertainty in the state [30]. Therefore, rather than a single value for each state variable, a probability distribution over each state variable is desired. This probability distribution could be defined as

$$p(\mathbf{x}_t|\mathbf{u}_{1:t}, \mathbf{z}_{1:t}, \mathbf{x}_0) \tag{2.6}$$

where $\mathbf{x}_t$ is the state at time t, $\mathbf{x}_0$ is the initial state or prior knowledge of the state, and $\mathbf{u}$ and $\mathbf{z}$ correspond to the measurements given by

$$\mathbf{x}_t = f_u(\mathbf{u}_t, \mathbf{x}_{t-1}) \tag{2.7}$$

$$\mathbf{z}_t = f_z(\mathbf{x}_t) \tag{2.8}$$

respectively. The function in equation 2.6 describes a conditional probability, which is the probability over $\mathbf{x}_t$ given the inputs, measurements, and prior knowledge. This definition implies that only the state at time t is to be estimated. If instead the state trajectory is desired, the required probability distribution would be given by

$$p(\mathbf{x}_{1:T}|\mathbf{u}_{1:T}, \mathbf{z}_{1:T}, \mathbf{x}_0) \tag{2.9}$$

Fundamentally, these two probabilities are different, and this has important practical implications as the derivation of the state estimation can be different in each case. Mathematically, the *state* will be considered as a *Markov state*, where the probability of future states depends only on the value of the current state, and not on the value of previous states.

The probability distribution in equation 2.6 can be estimated recursively through time t as follows [30]

$$p(\mathbf{x}_t|\mathbf{u}_{1:t}, \mathbf{z}_{1:t}, \mathbf{x}_0) \propto p(\mathbf{z}_t|\mathbf{x}_t) \int p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{u}_t) p(\mathbf{x}_{t-1}|\mathbf{u}_{1:t-1}, \mathbf{z}_{1:t-1}, \mathbf{x}_0) dx_{t-1} \tag{2.10}$$

The final term in the integral here is equal to the posterior estimated at the previous time $t-1$, giving a recursive process for estimating the posterior, using only the new information at time t. This is known as a Bayes filter.

Similarly, the probability distribution in equation 2.9 can be estimated recursively through time t as follows.

$$p(\mathbf{x}_{0:T}|\mathbf{u}_{1:T}, \mathbf{z}_{1:T}, \mathbf{x}_0) \propto p(\mathbf{x}_0) \prod_t p(\mathbf{z}_t|\mathbf{x}_t) p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{u}_t) \tag{2.11}$$

The Bayes filter is a purely mathematical construct. In order to practically compute the posterior distribution, an explicit representation of each probability distribution is needed. A number of practical implementations exist for this mathematical Bayes filter. The choice of implementation can depend on a number of factors but principally the state estimation algorithm must be suited to the state to be estimated. Here, the algorithms used for continuous and discrete state variables will be described.

2.3.1 Continuous State Estimation

The *Kalman filter* (KF) [79] represents the posterior distribution of the state as a linear multivariate Gaussian distribution, parameterised using the mean and covariance of the distribution. The *extended Kalman filter* (EKF) [60] and *unscented Kalman filter* (UKF) [80][81] allow representation of nonlinear functions, where the EKF uses linearization and the UKF uses a number of sample points. Referring to equation 2.10, the algorithm first computes a predicted posterior using a physical model of the system and the previous estimated posterior (using the integral of the product of the two distributions, $p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{u}_t) p(\mathbf{x}_{t-1}|\mathbf{u}_{1:t-1}, \mathbf{z}_{1:t-1}, \mathbf{x}_0)$), and then computes a corrected posterior by incorporating measurements (using $p(\mathbf{z}_t|\mathbf{x}_t)$). The Kalman gain is computed at each time and determines the extent to which the updated posterior is influenced by the physical model and by the measurements. The prediction step is computationally simple, however the correction step involves a matrix inversion which is much more computationally expensive.

Similarly to the Kalman filter, the *information filter* (IF) [82] represents the pos-

terior distribution of the state as a multivariate Gaussian distribution, parameterised as an information matrix and an information vector. This is effectively the opposite to the mean and covariance used in the Kalman filter, so the implementations have many similar and opposite properties. The algorithm similarly uses a prediction step and a measurement step, however in this case the prediction step is computationally expensive, while the incorporation of measurements is computationally simple. The relative simplicity of incorporating measurements is a useful feature when applying this algorithm to problems with multiple robots.

The simple Kalman filter and information filter both use a unimodal probability distribution for the posterior, given by a multivariate Gaussian distribution with a single peak. This can be a poor representation of the posterior probability in the case where a robot could equally likely be in one of many distinct places, but not between these places. This might often be the case in the pipe environment, where the robot could be in one of many pipes, but not inbetween them. These filters can be extended to represent the posterior distribution as a mixture of Gaussian distributions, a sum of Gaussian distributions where each Gaussian component is weighted depending on its likelihood. This is known as *multi-hypothesis tracking* [30].

Unlike the *parametric* Kalman and information filters, the *particle filter* (or *Monte-Carlo localization*) [83] uses a *non-parametric* representation of the posterior distribution. Instead of a defined probability distribution, a number of *particles* are used which each represent a hypothetical robot state. These particles are all passed through a filter using a prediction step similar to the parametric filters. Unlike the parametric filters, the correction step is done through resampling, where some of the particles are kept and duplicated and others are removed, depending on an importance factor found using the measurements. Variations of this implementation exist which are structured slightly differently for improved effectiveness or efficiency, including the Rao-Blackwellised particle filter [84] on which the FastSLAM algorithms [85] are based. The particle filter is versatile; it can represent arbitrary probability distributions without incurring error from linearization, including multimodal distributions.

The Kalman filter and information filter are used to solve the *online SLAM* problem,

where state is to be estimated at the current time. When it is desired to estimate the state trajectory over time, a solution is needed for the *full SLAM* problem. From equation 2.11, *pose-graph optimization* [30] can be derived. Applying the Gaussian models for motion and measurement to equation 2.11, and taking the logarithm, gives the log-likelihood which is a sum of quadratic terms. This can be maximised, or minimised in the case of the negative log-likelihood which is in the form of a least squares optimization problem. A number of algorithms for computing this pose-graph optimization have been demonstrated, including *GraphSLAM* [86], *square root smoothing and mapping* ($\sqrt{SAM}$) [87], *incremental smoothing and mapping* (*iSAM*) [88], *iSAM2* [89], *multi-hypothesis iSAM* (*MH-iSAM*) [90].

2.3.2 Discrete State Estimation

When estimating a discrete state, the system is typically referred to as a *Markov Model*. Acknowledging that the state can't be measured directly, and that measurements and motion of the robot will have some uncertainty, the system can be thought of as a *Hidden Markov Model* (*HMM*).

For the filtering problem, the probability distribution $p(\mathbf{x}_T|\mathbf{z}_{1:T}, \mathbf{u}_{1:T}, \mathbf{x}_0)$ is desired. The *forward algorithm* is used to compute this distribution, following an equivalent discrete probability derivation [30] to that that gives equation 2.10 for continuous probability distributions. This algorithm is described by

$$p(\mathbf{x}_t|\mathbf{u}_{1:t}, \mathbf{z}_{1:t}) = p(\mathbf{z}_t|\mathbf{x}_t) \sum_{\mathbf{x}_{t-1}} p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{u}_t)(\mathbf{x}_{t-1}|\mathbf{u}_{1:t-1}, \mathbf{z}_{1:t-1}) \tag{2.12}$$

If the sequence of states is desired, the probability distribution $p(\mathbf{x}_{1:T}|\mathbf{z}_{1:T}, \mathbf{u}_{1:T}, \mathbf{x}_0)$ is to be found. The *Viterbi algorithm* [91] estimates the most likely sequence of states given a sequence of measurements. This is the maximum *a posteriori* estimate, however the full posterior distribution is not estimated. The algorithm estimates the probability of the most likely sequence to each state $\mathbf{x}_t$ in the state space recursively forward through time, using the information in $\mathbf{u}_t$ and $\mathbf{z}_t$, and the probabilities of each state $\mathbf{x}_{t-1}$. The idea here is that only the most likely sub-sequence from state $\mathbf{x}_0$ to state $\mathbf{x}_t$ will need to be considered further

forward from time t, as a most likely sequence overall which passes through state $\mathbf{x}_0$ and state $\mathbf{x}_t$ will also take this most likely sub-sequence. By recording the most likely preceding state to each state at time t, the sequence of states can be simply recovered by stepping backwards through time.

A discrete map representation has been shown to be useful in navigation where a *HMM* localization method is extended to a *Partially Observable Markov Decision Process* (*POMDP*) [48]. Early work on localization in a pipe network [20] also uses a *POMDP* for localization and navigation, where the transition model between states is described, as is the observation model which finds the likelihood of an observation of the robot's surroundings at a junction corresponding to a known discrete type of junction.

2.4 Sensing for Localization

The use of a range of sensing methods has been demonstrated in previous research in in-pipe robot localization, pipe fault detection, and pipe mapping. Different types of sensors provide different types of information from the environment, which can be used in localization in different ways. However, different sensors also suffer from different drawbacks, which could be economic cost, size, power consumption, or reliability.

Conventional robots might use odometry measurements (which are measurements of change in position over time) range finding measurements (which are measurements of distance to objects), feature recognition from images from visual sensors, and other application specific sensing.

In this section, the range of sensing technologies applied to in-pipe inspection, either for localization or for fault detection, will be reviewed with an emphasis on the application to localization. The review will extend to localization with similar sensing methods not yet applied to pipes where relevant.

2.4.1 Visual Sensing

Although there is some overlap in the technology used, visual sensing for localization could be separated into three broad categories: motion detection (or visual odometry), point feature detection, and feature detection.

In *motion detection*, the aim is generally to estimate the instantaneous change in position of the robot between images recorded by the camera. This estimated motion can be integrated to give an estimate of the position of the robot. In *point feature detection*, the aim is to recognise pixel-scale features in camera images that have been detected previously. This might be used to estimate the change in position of the robot between two images as in motion detection, or it might be used to recognise a location visited previously, providing a *loop-closure* measurement. There is some overlap in the range of techniques used to achieve these aims.

Motion detection methods such as LSD-SLAM [92] and DSO (direct sparse odometry) [93] are well developed for robotics, and *optical flow* has been well studied in the context of robotics [94] and for general computer vision [95]. Point feature detection methods including the SIFT [96], SURF [97], and ORB [98] algorithms are well developed in robotics. The latter has been directly applied to localization as ORB-SLAM [44] [99]. Vision-based algorithms can be augmented by inertial sensing [100].

Challenges in applying visual localization methods in pipes such as feature-sparseness, reflection from lighting, and difficulty in focusing on the close surface, have been addressed in the literature. The challenge from feature-sparseness specifically is investigated briefly in Chapter 1. There, Figure 1.3 shows an estimate of the performance of the ORB-SLAM algorithm in different vision datasets, from typical outdoor environments, to somewhat confined tunnel environments, and very confined pipe environments. It is seen that the number of *feature matches* between images is one or two orders of magnitude lower in buried pipes than it is in a typical environment.

For robots with cameras pointing toward the pipe wall, the use of an inertial nav-

igation system (INS) fused with motion detection has been demonstrated [101], where the INS is used to predict the motion which informs the optimization based motion detection of the likely motion, which reduces the computation required to find a match and increases the accuracy of the estimation. Stereo cameras have also been used in this case [102], which have the advantage over single camera systems of being able to estimate depth in images, where SIFT-based feature detection is used to estimate motion. Other applications have been demonstrated where the camera is directed along the axis of the pipe. Transformations are made to account for the radial motion of features observed by a camera moving along the pipe, and SIFT-based [103] and Harris-based [68] methods are used to estimate the motion. Despite the research for vision-based localization in pipes, there are still challenges to overcome. Comparison between different vision-based localization methods shows how ORB-SLAM and DSO methods fail and lose track of the robot position, while methods designed for pipe environments which use prior knowledge of the diameter of the pipes are unable to eliminate the observed phenomenon of a reduction in estimated pipe diameter as the estimation progresses [45].

In *feature detection*, the aim is to detect and locate a feature of interest in the environment. In a water pipe the environment is largely featureless compared to a typical robot's environment, but features do exist in the form of junctions, corners, and faults. These features can be used as landmarks in localization.

Early work on a human-operated robotic system [104] showed how the length of fissures (small faults along the circumference of the pipe) can be detected using image processing techniques. Augmentation of a camera system with a light ring generator is shown [105] to highlight faults in the pipe which can be processed and classified using a neural network for robust detection. With these technologies, a landmark can be created with the detection of a fault, and the recognition of a specific fault might be achievable with the measurement of length or other characteristic, improving the usefulness of the landmark in localization.

More research on this topic has focused on the detection of junctions and corners as landmarks, either for localization or for motion planning. Where the environment can be assumed to have t-junctions, elbows, or straight pipes, these features can be detected

using imaging processing techniques [106]. Time-of-flight imaging has been demonstrated in detection of t-junctions, y-junctions, and elbows, [107] where the image recorded is compared to a synthetic image expected for a straight pipe. Structured light in various forms has been used to detect corners and junctions, using the shadows formed due to the displacement between illuminator and camera [108], using a projected laser spot array [109], and using projected radial laser lines [54]. These methods are generally effective, with some chance of misrecognition (both false positives and false negatives). Classification of junctions into different types would likely be reliable, and although direct recognition of a specific junction would likely not be possible due to the similarity between features of the same type, it may be possible to achieve using data association techniques typical in localization.

The hardware requirement for visual sensing is difficult to assess. Many of the reviewed methods have been demonstrated in larger pipes than would be expected in buried pipe inspection, however cameras can currently be made to fit within this size constraint. The illumination hardware would also likely be small, however these components add to the total size and power consumption of the robot. The computational hardware may also be significant in terms of size, cost and power consumption, however many of the methods using structured light involve extracting features from the images which may result in a lower computational load than processing a full image. Cameras are inherently directional, so it is possible that the methods described here would be less effective if the robot were able to move freely within the pipe, unlike the robots described here which are constrained to be directed along the axis of the pipe. Cameras may be used for detecting and inspecting faults in the pipe network, so use of the same camera and illumination for localization would be efficient.

In conclusion, visual sensing for localization is well developed, and the range of improvements to both accuracy and cost could give good performance in the pipe application as demonstrated in the literature, despite the constraints of the pipe environment. Throughout this thesis vision is treated as a means of making the various measurements described throughout this section, which may or may not be possible in the pipe application. For example, vision is one potential method of obtaining an odometry measurement, and is a

potential method of achieving feature detection or loop-closure.

2.4.2 Scanning Rangefinder Sensing

Rangefinder sensors function by emitting some kind of signal, typically either high frequency acoustic (ultrasonic) signals or a laser (as in a *lidar* sensor), and measuring the time taken for the signal to reflect back to the sensor from objects in the environment. Knowledge of the speed of the wave can then be used to estimate the distance to a reflective object. Often, these sensors are used in a scanning approach, where the emitted signal is sent across a range of angles from the robot, giving either a scan in two or three dimensions.

This type of sensor has been used in robot localization for decades [110, 111], and incorporated into influential work on robot localization [112]. Lidar in particular has been used in localization of vehicles on roads [113, 114].

Rangefinder sensing has also seen limited application to localization of robots in pipe networks [115]. This early work in the application used a scanning ultrasound sensor to make a measurement at junctions in a pipe network which could be used to determine the number and position of exits from the junction. Recent work has shown the use of a scanning lidar sensor integrated with vision and IMU-based sensing to localize a robot in a penstock (a tunnel-like structure which is part of a dam) [116]. In these works, the rangefinder sensor is able to add information regarding the robot's position with respect to the cross-section of the pipe or tunnel, and information regarding a change in the direction of the pipe or tunnel.

The drawback of this type of sensor is that it is unable to give much information about the position or velocity along the axis of the pipe due to the featureless nature of the pipe surface. This is in contrast to typical robot environments where the geometry of rooms and obstacles within them can be observed by a rangefinder sensor, and in contrast to application on roads, which share the network structure of a pipe network, but typically have many more informative features within them. Despite this limitation, some work has shown the use of laser and sonar scanners to find a profile of the pipe wall in sewer pipes [22], and identifying junctions and corners would also be possible using the same process. These

methods offer similar results as found using visual techniques, and would likely come with similar hardware requirements.

Overall, as typical rangefinder sensors are limited in range compared to the scale of the pipe environment, the detection and recognition of features is limited to the robot's immediate surroundings. The sensor type could still be useful in small-scale localization with respect to the pipe cross-section, and in detection and recognition of junctions.

2.4.3 Long-range Acoustic Sensing

Acoustic sensing methods can be used to acquire a range of information from the environment.

General acoustic SLAM has been developed outside of pipes, as a complement to visual SLAM in complex environments. Using only received signals from the environment, bearing-only SLAM (using only the direction, or bearing, of received signals, and not the range) has been demonstrated [117], where multiple sound sources, or features, can be localized even when the sound is intermittent and in the presence of noise and reverberation.

Using acoustic echoes for robot localization, where the robot actively emits a sound and receives the resulting echo, requires an understanding of the acoustic response of the robot's surroundings. Techniques for estimating the acoustic *room impulse response* (RIR) in different environments with variation in the position of acoustic transmitter and receiver have been studied for decades [118], but a number of methods for acoustic robot localization have been demonstrated in the last decade.

Recent work has shown the use of a single co-located acoustic source and receiver to estimate the position of a robot in a structured room [119, 120], where perception from the *room impulse response* is used in a filtering approach and in an optimization approach. In this previous work, it is assumed that each echo is able to be distinguished, that there are no multi-path echoes, and that each detected echo can be associated with a specific wall in the room. These assumptions cannot be made in the pipe environment, where measurement of acoustic echoes is more challenging due to noise from the robot and the environment,

multi-path echoes are likely, and data association is not trivial.

The estimation of the parameters describing the impulse response in the presence of noise has been improved by using an optimization approach in recent work [121, 122]. This method has been applied to localization using a single microphone [123] and using an array of microphones, and the effects of signal to noise ratio, hardware transfer function [124], nongaussian noise, and faulty microphones [121] have been evaluated. This idea has also been applied to using robot ego-noise rather than additional noise for probing the environment [125], although in this case a flying drone was used, which could have much more ego-noise than a small terrestrial robot in a pipe.

The time of travel of sound can be used to estimate the distance it has travelled over. Outside of robotics, the use of acoustic echoes for inspection of pipes is well established as *acoustic pulse reflectometry* (APR). Using stationary sensing equipment, this approach has been taken in gas pipelines, where blockages can be detected at distances up to as far as 10 kilometers [126], and can be done by measuring the difference between acoustic measurements taken before and after a blockage has formed [127]. This approach has also been taken in sewer pipes [128, 129], where the detected signal can be used to locate and characterise a blockage.

Some work using APR in pipes has the acoustic source and receiver placed at the edge of a pipe network, such that the measurement is made in only one direction. The detection of faults in the pipe from the acoustic impulse response can be done by comparing the measurement with a previously measured reference signal [127] or with a simulated signal in the likely case that a reference signal is not available [130]. When making this comparison, some prior knowledge of the pipe network or prior information from the pipe network is needed, and the comparison between reference signal and measured signal is done manually [130].

The challenge in this use of APR is in separating the waves reflecting from features in the environment and reflecting from the acoustic source. This has been addressed by *wave-separation* in the time domain [129, 131] and in the frequency domain [131], or by *source*

termination [131]. *Wave-separation* requires multiple microphones, with requirements on their relative position which might be difficult to achieve on a small robot. *Source termination* requires nanosecond-order computation and emission of a signal to cancel acoustic waves arriving at the source, which is demonstrated in the literature [131] for a static sensor using laboratory-grade data acquisition and processing hardware, but is likely not possible for a small robot platform where the computation time is reduced in comparison to the literature due to the closeness of speaker and microphone and the lower computation power.

Some work on using APR in buried pipes has the sensing equipment placed in the middle of a pipe network, so two directions of measurement are possible, and it is possible to estimate the direction of the reflection using two microphones if sufficient separation is available [130]. Only the distance to features can be measured, so in a network of connected pipes, there is no way to determine the location of a feature beyond a junction between two pipes with a single measurement [127, 130]. Instead, measurements from different locations can be combined to estimate the location, although the required measurement positions will depend on the unknown position of the feature and the complexity of the pipe network [127, 130]. The requirement for multiple measurements at different locations in a network could be met by placing the sensing equipment on a mobile robot.

In wastewater networks, pipes are connected together at manholes, which are described [128] and shown [132] to be highly reflective in the literature. This would limit the sensing range possible with acoustic reflection detection for a static sensor, and motivates the application of the sensing mode to a mobile robot.

With a tethered robot the time of travel of a sound wave from one end of the pipe can be measured using the tether to synchronize the source and receiver [69]. The drawback of use of a tether might be overcome by using echoes from corners or junctions, or by using multiple robots to measure time of travel.

Overall, acoustic reflection measurement has been shown to be effective in measuring the distance to features such as faults in a pipe. The advantage of using this method on a mobile robot is that measurements can be taken autonomously from different locations in

a pipe network, which is needed either to locate a fault in a network, or to inspect pipes which are acoustically isolated from each other by reflective connections. The mobile robot would also be able to use other sensing modes to thoroughly inspect the pipe network, so localization of the robot using acoustics would improve the operation overall. The challenges in the application to a small mobile robot is the limit in quality of acoustic hardware, limit in computation available for active termination of multiple-order reflections, the subsequent reverberant environment. Fundamentally the limitation is in the unknown location of the robot, and therefore acoustic equipment. The challenge is in the estimation of a sequence of robot positions in an unknown environment, rather than the precise estimation of distance of a fault from a known position in a known environment.

In summary, acoustic sensing provides a range of sensing modes which could be useful in the pipe environment. Acoustic echoes in particular have not been explored for use by robots in pipes, while their effectiveness has been shown in sensing in pipe networks. The propagation of acoustic waves through water filled pipes may have some complications compared to air filled pipes which could be both advantageous and disadvantageous. Their popularity in general pipe sensing might make it easier to use existing sensing technology, and make it possible to efficiently use the same sensor for both inspection and localization.

2.4.4 Short-range Acoustic Sensing

As with visual sensing, acoustic methods can be used to locate and identify features in the pipe environment. Ultrasonic sensing can also be used to find features along the length of the pipe by receiving reflections from the ground outside of the pipe [133], which in the case of water pipes contains more features than the pipe interior. Acoustic sensing has also been demonstrated with passive sensing of sound from a leak in a pipe [134]. In this work the sound measured is compared for different flow rates, surrounding media, and position of the sensor relative to the leak. In localization, the distinct measured sound could be used to recognize a previously observed fault, or could be used to estimate the distance or direction to a fault.

Acoustic methods can be used to measure a spatially varying field along the pipe. As well as increasing the number of observable features, the use of ultrasound to measure variation in the ground outside of the pipe can produce a spatially varying profile [133]. In similar work, a hydrophone is used to excite a vibration in the pipe, where the response varies in characteristics along the length of the pipe [135] [136]. Field detection could be used for both precise localization within a single pipe, and recognition of a specific pipe in large scale localization. A drawback in both cases is the possible directionality of sensors [134]. A range of approaches to estimation have been taken in this previous work, where the effectiveness of a particle filter and a Kalman filter in using information from this sensing mode is shown. There is room for further development and evaluation of the related estimation algorithm which could be based on an optimization approach.

This sensing mode could improve the feature density of the environment, as it is able to observe aspects of the surroundings that can't be perceived by vision or other sensors. The use of similar sensing in the passive detection of leaks means that the sensor could be used for localization and inspection, improving efficiency. However, the sensing mode is only able to perceive nearby features, so could be used to recognise previously visited locations, but not to directly measure the position of the robot along a pipe, for example.

2.4.5 Inertial Sensing

There are six components of an object's pose in Cartesian coordinates: x-, y-, and z-position, and roll, pitch, and yaw angles. Inertial navigation systems typically use accelerometers and gyroscopes to measure linear acceleration and angular velocity respectively. These are often combined with magnetometers to measure the absolute angle. These devices are often miniaturized as inertial measurement units (IMUs), and a variety of these devices are commercially available [137]. *Consumer grade* IMUs have a cost of the order of 1 to 10 GBP, and *tactical grade* IMUs have a cost of the order of 1000 to 10000 [138], so can each be used in robotics in different contexts. In consumer grade devices, the low grade noisy accelerometer is used only to measure the direction of gravitational force rather than being integrated to form a position measurement. Even so, the combination of absolute measurements from the

accelerometer and magnetometer is able to give a drift free estimate of attitude (roll, pitch, and yaw), while the gyroscope provides a higher frequency measurement of change in angle and aids in computing the estimates of attitude when the IMU is moving or accelerating. In metal pipes [139] or underground [140], the Earth's magnetic field is distorted such that a magnetometer can't be used. While the accelerometer can still be used to estimate the roll and pitch angles, only the drifting relative measurement of angle from the gyroscope can be used to estimate the heading angle.

A number of implementations have been demonstrated in applications from large oil and gas pipelines to water distribution scale pipes. Inertial measurements can be combined with an encoder on the robot's tether cable which gives a measure of the distance travelled, using the EKF [67] or pose-graph optimization [141] for data fusion. The tether is a limitation of these methods, however there are alternative methods of acquiring similar information. For untethered robots which use some kind of encoder to measure relative change in position, the estimation can be improved by using information about known landmark positions [142] [143] or by using measurements of pipe section length [35]. Landmark positions may be obtained in water pipes as hydrants or valves, however detecting or recognizing a specific landmark may be a challenge. The use of inertial measurements to detect features such as junctions has been demonstrated [32] [144] [145], which is used in combination with known length of pipe between junctions or used to reduce the drifting angular estimate where the pipe between junctions is known to be straight. In a spherical rolling robot, inertial measurements can also be used in combination with a magnetic switch which outputs a signal when in contact with the pipe wall. [146] The phase difference between the inertial and magnetic switch signals can be used to estimate the inclination of the pipe.

Inertial sensing is combined with other sensing methods elsewhere in the literature, and is an important component of many implementations of in-pipe localization. Typically the accuracy achieved using inertial sensing alone increases with the cost, size, and power consumption of the sensors, but results have been demonstrated using the constraints of the pipe as an advantage, which may allow accurate estimation with lower cost sensors. Inertial sensing can be used to estimate the orientation of the sensor in Cartesian coordinates. This

is useful generally in robotics, but may be less directly useful if the state to be estimated is not defined in this coordinate system.

2.4.6 Magnetic Sensing

While the distorted magnetic field inside metal or buried pipes is not expected to be useful for estimating the attitude of the robot [139, 140], the varying field along the length of a pipe could be used as a recognizable spatial field, as has been demonstrated in the similar context of drilling robots [140]. In this application, the robot moves forward and backwards along the same path observing the same magnetic signal twice, which can be useful in loop-closure. The robot also uses two sensors, at each end of the robot, so that an odometry estimate can be made by comparing the out of phase spatial signals.

The detection and recognition of parts of an ambient magnetic field has been used in robot localization in more typical two-dimensional indoor environments [147, 148, 149, 150]. In these applications, previously visited locations are recognised by matching the magnetic field measurements at different points in time.

A similar method may be useful in pipes, especially in precise localization where a robot would similarly move along the pipe to observe a fault two or more times. The characteristics of the magnetic distortion would need further investigation. The sensing mode could be used to increase the density of features in the environment, but it would be limited to observing nearby magnetic properties so would be useful for recognising previously visited locations rather than directly observing the robot's position along a pipe. This sensing mode would be expected to function differently in buried metal pipes and buried pipes made of other materials.

2.4.7 Pressure Sensing

Leaks in water pipes have a range of characteristics, but generally will cause some change in pressure, both locally and across the network, emit some acoustic anomaly, and will be

some physical gap in the pipe wall. Methods to detect a leak using pressure have been implemented where the millimetre-scale millibar pressure difference is detected either directly using pressure sensors [151] or less directly using a membrane connected to mechanical system [152] [29].

Detection of the location of a leak, both along the length of the pipe and the location around the circumference, could offer a means of localization. Although in localization applications this is best suited for high precision localization as the presence of leaks in the network is not guaranteed, a sensor of this type may be required on the robot for leak detection, so using the information for localization where possible would not add to hardware costs. This sensing mode would be limited to observing nearby leaks, so, like magnetic property sensing, would be useful for recognising previously visited locations but not directly observing the robot's position along a pipe.

2.4.8 Radio Sensing

Three implementations of radio wave sensing in pipe localization are reviewed here. The first uses the spatially periodic signal formed in a pipe from the properties of an emitted radio wave as a map which can be used to localize the robot in the pipe [38], similarly to the potential use of acoustic or magnetic spatially varying signals used elsewhere. To implement this method, robots may be able to work in pairs, one transmitting the signal while the other traverses the pipe. The second implementation uses the strength of a signal received from transmitters placed on the ground above pipe [153]. There are challenges in implementing these methods in water pipes. Multiple transmitters may be expensive to install across the network, especially as water pipes are mostly buried below roads. The third implementation uses radio beacons placed at various points in the pipe network to localize the robot when it is nearby [154]. This approach gives an estimate to the precision of the spatial frequency of the radio beacons; the position estimate is likely to be accurate when the robot is within range of a beacon, but the robot has no extra information available for localization when it is far from a beacon. Estimates for the range of radio-frequency communication in buried pipes vary, as the range depends on the material inside and outside of the pipe and on the frequency

of communications [155], as well as on the signal strength required for communication. The range for radio-frequency transmission has been estimated to be around 3.2 metres [154], and it has been concluded that the attenuation of radio-frequency signals along a buried pipe is sufficiently large that relay stations would be required to communicate along a 100 metre pipe [155].

2.4.9 Sensing in other Environments

For a comparison relevant to the pipe environment, this section will review sensing for robot localization in other similarly confined environments. While the pipe application environment is substantially different from other typical applications such as roads (shown in Chapter 1) or the built indoor environment, there are applications that have some similarities to buried pipes.

Recent work on localization of robots in underground mines has been done as part of the *DARPA Subterranean Challenge* (*SubT*), which include *tunnel* and *cave* environments. These have some similarities in structure to the pipe environment, in that typical sensors such as vision and lidar are limited in scope and unable to perceive distant features.

Localization using lidar in the *SubT* mine environments is shown to have a low error of 1.5% of the distance travelled [156], where the success is attributed to the use of the *Ouster OS0-128* lidar scanner (a high-resolution imaging sensor which gives a precision of less than 5 cm over a range of 50 metres, but typically costs more than £10,000 and is around 0.1 metre in each dimension), and plentiful surfaces and features suitable for this sort of sensing [156]. The tunnel widths in the mine environments of around 5 metres [157] are an order of magnitude larger than the typical pipe diameter described in Chapter 1 and this type of sensing would be more challenging to apply to the pipe environment. The pipe environment would present a challenge to this approach as it is made of long sections of uniform pipe which are challenging to this type of localization [156], and it is made of sections with a diameter typically around 0.3 metres, making it difficult for a robot to carry a similar high-quality sensing payload. The cost of this sort of sensor is at the same order of magnitude as the cost

of typical remotely controlled pipe inspection robots, while low-cost pipe inspection robots have been proposed in the literature for 5% of the cost of this sensor alone [36].

Other results using a lidar scanner in the *SubT* mine environment show low estimate error of 0.5% and 0.2% relative to the dimension of the whole environment in two different locations [157, 158]. However, the odometry estimate using only measurements of the robot's motion, which will influence the subsequent improved estimate using lidar, has error of 1% and 3.5% respectively. This is at the lowest end of the relative error observed for linear motion in the literature in the pipe environment shown in Table 1.3. Additionally, the localization system is reported to identify several hundred loop-closure measurements in each case, where the robot recognises a previously visited location. As well as this, the localization is performed using a laptop with Intel i7-8750H 12 core processor, with computation time of around 1 hour [158]. This computation time on a more limited robot in a pipe environment would be expected to be much larger. Finally, the feature density of the *SubT* environment is assessed, where it is found that while some parts of the environment are *featureless* to the lidar sensor, most of the environment has a sufficient number of features to allow recognition of places [158]. While it is difficult to compare between measurements for different sensors, Figure 1.3 shows a comparison of vision-based features for pipe environments and one of the *SubT* environments, where it can be seen that there are one or two orders of magnitude fewer feature matches in the pipe environments. While the results show low error in the mine environment, there are several indications that this environment has many factors which reduce the challenge for localization compared to the pipe environment.

Robot localization has been done in a similar mine environment, where camera and lidar sensor data is combined to get an estimate accuracy of around 1 metre after several traversals along a 2.6 km tunnel [159]. This is done using a sequential mapping and localization approach, where a map is first built manually using prior knowledge of the mine, which is reported as being essential for operation in this environment [159]. The requirement for prior knowledge of the environment indicates the difficulty in localization without this knowledge, which would be expected to be more severe in the pipe environment due to the smaller number of features in pipes compared to mines illustrated in Figure 1.3.

Overall, while localization in similar environments has been shown to be effective, the current requirements for effective localization are not available in the pipe environment, including sensor quality, environment features, low locomotion uncertainty, and computation power. Detailed prior knowledge is seen to be useful in compensating for challenges in localization, which can be applied to an extent in the pipe environment.

2.4.10 Comparison of Sensing Modes

Here, the literature on various sensing modes has been presented both for general robot applications and for application in the pipe environment. Comparing the literature on various sensing modes, it is seen that a range of alternatives to the vision sensing popular in contemporary conventional robots have been developed for application in the pipe environment.

In principle, vision has two main limitations in this environment which reduce its appeal, despite its popularity in application to robot localization generally: Firstly, the quality and quantity of features available is lower in the pipe environment compared to a more general case. Secondly, compared to a typical environment, the perspective of the robot is very limited. In a typical room where the length of each of the walls is likely the same order of magnitude, a robot can likely perceive a substantial amount of the features in the room (whatever the features are) from a large range of poses. This is useful for estimating odometry and for producing regular loop-closure measurements. In the pipe environment however, the length of the pipe is likely to be at least one order of magnitude larger than the diameter, and a robot can likely only perceive relatively nearby features. Therefore each feature can only be observed from a small range of poses which makes loop-closure measurements more sparse.

Alternative sensing approaches explored in the literature could avoid these limitations. Acoustic and magnetic sensing for example could allow detection of more features in the environment, reducing the sparseness of features. Acoustic and radio sensing could be used to detect distant features, which increases the perspective of the robot compared to vision.

Sensing Mode	Sensing Advantages	Sensing Disadvantages
Vision	Precise feature detection, detailed features, easy to interpret by human operator, well developed hardware	Pipe environment lacks features, perspective is limited
Rangefinder	Precise feature detection	Pipe environment lacks features, range is limited
Long-range Acoustic	Long-range feature detection, can observe many features at once	Reverberant environment, difficult feature association
Short-range Acoustic	Increases feature density, sensing can be also used for inspection	Short range, perspective is limited
Inertial	Sensor cannot be occluded	Must be combined with another sensing mode, short range, limited perspective
Magnetic	Increases feature density	Depends on pipe material, short range, limited perspective
Pressure	Increases feature density	Requires pressure variation, short range, limited perspective
Radio	Can transmit information, therefore no data association	Only works when robot is within range

2.5 Literature Review Conclusions

This review of the literature has covered aspects of state definition, state estimation, and sensing for both the general robot localization problem and for the problem of localizing robots in a pipe network. From this, the gaps in the literature can be found.

It was concluded in section 2.2 that the hybrid continuous-discrete approach to state definition is likely the most suitable for the pipe environment, and is also undeveloped in the

literature for this application. This is a clear gap which can be addressed by the work in this thesis, however as it is a hybrid of the alternative approaches (continuous and discrete), these must be assessed in this application too.

It was concluded in section 2.3 that while each of the available fundamental approaches have been applied to localization in the pipe environment, their use has been primarily to demonstrate an aspect of sensing. Therefore, there is a gap in the development of a state estimation approach suited to this environment.

Finally, it was concluded in section 2.4 that acoustic or radio sensing could provide measurements not possible in this environment using typical sensing methods like vision. While acoustic sensing has been applied for localization in pipes, it has not been fully developed and while existing approaches in the literature are effective, they have some limitations. There is therefore a gap in the literature to further develop acoustic sensing for this application.

These gaps motivate the subsequent research presented in this thesis.

Chapter 3

Motivation: Continuous Space Localization in Pipe Networks

3.1 Introduction

The literature review in Chapter 2 discussed the *continuous*, *discrete*, and *hybrid continuous-discrete* state definitions that can be applied to robot localization. As in general robot localization, the *continuous* representation has been used most in the pipe environment. However, the literature develops the *front-end* of the localization problem (from sensing to feature detection), and lacks a sufficient assessment of this localization *back-end* when applied to the pipe environment.

Therefore, this chapter shows the results of the application of typical front-end measurements to a typical back-end estimator, and aims to assess what the strengths and weaknesses of the approach are. The scope of the chapter is both medium and large scale state estimation, as illustrated in Figure 3.1.

The structure of this chapter is as follows. The problem definition used in this chapter will be given, along with the data that will be used as an input to the localization algorithm. The localization algorithm will be defined, and the results of a number of ex-

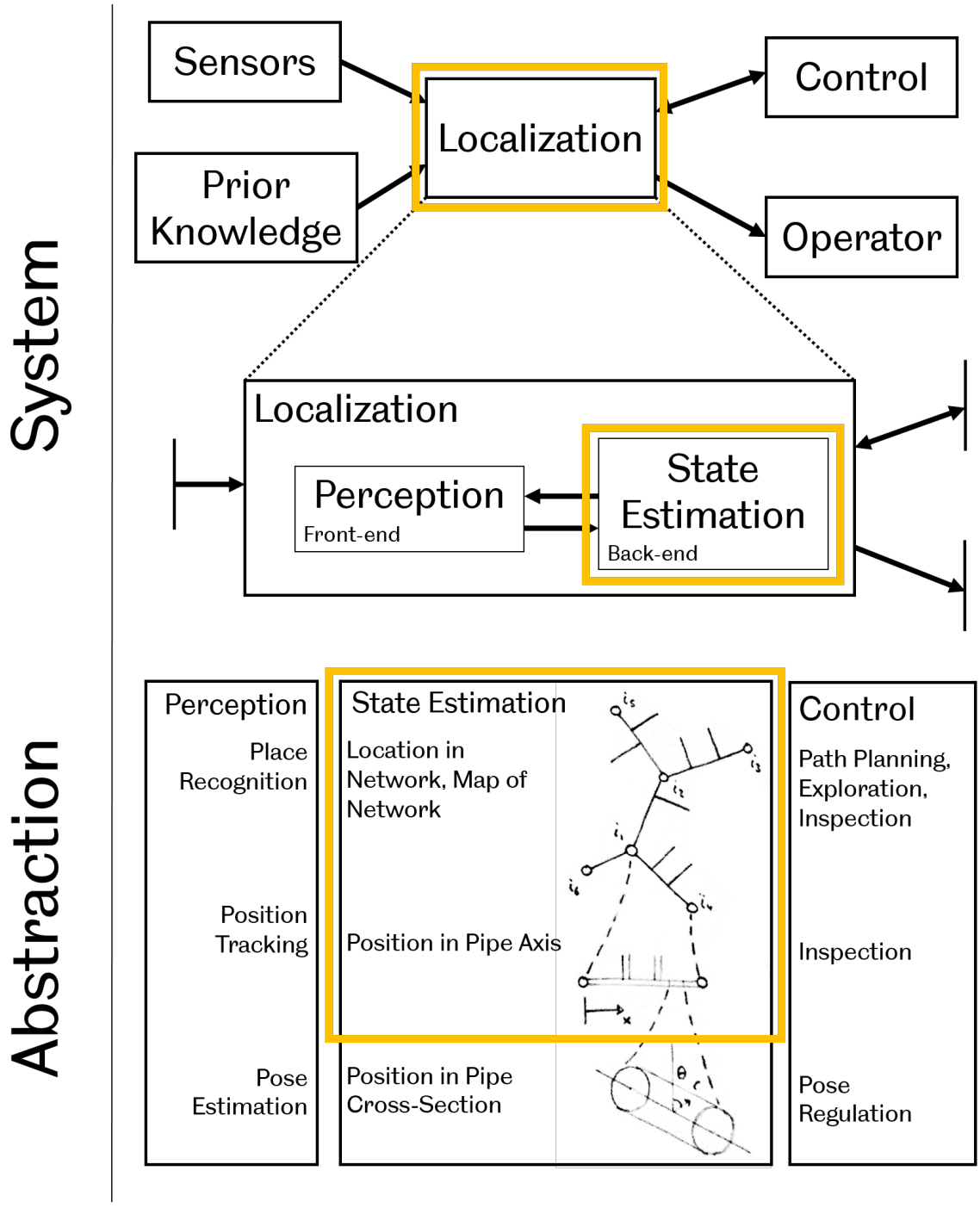

Figure 3.1: *An illustration of the scope of this chapter.*

periments will be shown. Finally, the conclusions will be used to motivate the subsequent research presented in this thesis.

3.2 Methods

In robot localization, typically there are two aspects of the *methodology*: The specification of the localization problem to be solved and the metric of success to be used, and the method of localization used. This section will describe each separately.

3.2.1 Methods: Problem

Robot localization algorithms are typically demonstrated by performing localization on a sequence of recorded sensor and motion inputs, estimating the trajectory and environment map that most likely produced this sequence.

It would be useful to have a large set of data recorded from robots operating in pipe networks on which robot localization can be demonstrated. In practice however, there are two problems which limit the volume of available data:

1. Development of robots for the application to pipe networks is still in progress on all aspects of the robot. The inaccessibility of the environment, unlike more typical environments shared with people, means that a robot needs to have a relatively high level of autonomy and capability in order to move around a large area and acquire data.

2. The buried pipe environment is largely made up of privately owned assets. Deployment of robots in this environment is certainly possible in the future, but asset owners will only allow deployment of a capable, reliable, robot. Therefore, the scale of practical data available is limited to smaller networks of pipes in the laboratory.

The method used here is to use simulation to produce a large volume of simple high-level data, suitable for the analysis required in this chapter. The map, *Net3*, is taken

from the software EPANET, as an example of a relatively large network of connected pipes. This software is typically used to simulate the dynamic behaviour of water inside pressurised pipe networks, usually in application to water distribution. In this case though, the map is used as a convenient representation of a real pipe network.

The motion of a robot through this network is simulated in Matlab. The motion is modelled coarsely, with the robot making a sequence of discrete motions though the network. The robot is constrained to the network, and its motion is limited to effectively one-dimensional movement along the pipes. Modelling simple locomotion, the robot travels some input distance each discrete time step, with added Gaussian distributed uncertainty. In practice this could be achieved by legged or wheeled motion in an air-filled pipe, or by direct movement through the fluid in a water-filled pipe. This model is simple but is fairly typical for models in robot localization in general, and therefore is suitable for this qualitative analysis. The added uncertainty in motion could be made more complex, by adding some time-varying uncertainty to model the effect of fluid flow on robot motion, or to model pipe conditions varying along the length of a pipe.

When the robot's motion would move it past the end of a pipe, the motion is truncated, so that the robot moves precisely to the junction at the end of the pipe. The robot detects a feature here, the junction or manhole, and in this chapter it is assumed that the robot can recognise if it has visited this location previously. This functions as a *loop-closure* measurement at a junction. The robot then turns towards a new pipe, making a measurement of the angle turned through, which is recorded with some Gaussian distributed uncertainty.

This system is repeated for 1000 time steps. The result is therefore a sequence of odometry measurements of the robot's change in position and angle at each time step, a sparse sequence of detection and recognition of features, and a sequence of loop-closure measurements. Prior knowledge of the relative position of features might be available, which could be obtained from an existing map, or by observing features like manholes above the ground in the area of the robot's operation.

In practice, there is some uncertainty in the measurement of each of these variables. The nature of the uncertainty depends on the variable. For the odometry measurements and the prior knowledge of feature position, there could be uncertainty in the continuous measurements which could, for example, be represented by a Gaussian distribution. For the loop-closure measurements and the detection and recognition of features there could be uncertainty in the discrete measurements which would take the form of possible *false positive* or *false negative* measurements. In this chapter, only the continuous uncertainty is applied, and it is assumed that there is no discrete uncertainty.

These sequences and prior knowledge are used as inputs to a localization algorithm which estimates the most likely robot trajectory given the inputs. Without explicit consideration of the sensing required to make these measurements, the input measurements are similar to those found in typical robot localization problems, but with some key differences that motivate the analysis here.

1. The limited perspective of sensors in this problem results in loop-closure measurements only between poses that are very close in space, and in relative odometry measurements only between poses adjacent in time. In other applications, a sensor such as a camera might be able to perceive the same features from a range of positions and more odometry or loop-closure measurements could be found this way.

2. Detection of features is only intermittently available, when the robot arrives at a manhole feature. Subsequently, prior knowledge can only be related directly to a sparse set of poses. In an environment with a more detailed structure, prior knowledge of a more dense set of features such as corners in a room might be available.

3.2.2 Methods: Solution

This chapter aims to investigate the performance of a typical localization algorithm when applied to the localization problem in a pipe network.

The typical pose-graph optimization-based localization algorithm is used to estimate

the robot's trajectory given the whole sequence of input measurements. As described in Section 2.3.1, finding the mostly likely trajectory is equivalent to minimizing the negative log-likelihood, given by

$$-\log p(\mathbf{x}_{0:T}, \mathbf{x}^{0:M} | \mathbf{u}_{1:T}, \mathbf{z}_{1:T}, \mathbf{x}_0) \propto \frac{1}{2}\Bigg[\sum_t \Big[(\mathbf{x}_t - f_u(\mathbf{u}_t, \mathbf{x}_{t-1}))^T \Omega_{u,t} (\mathbf{x}_t - f_u(\mathbf{u}_t, \mathbf{x}_{t-1})) + \sum_m (\mathbf{z}_t^m - f_z(\mathbf{x}_t, \mathbf{x}^m))^T \Omega_{z,m,t} (\mathbf{z}_t^m - f_z(\mathbf{x}_t, \mathbf{x}^m))\Big]\Bigg] \tag{3.1}$$

In this chapter, the state to be estimated is given by $\mathbf{x}_{0:T}$ and $\mathbf{x}^{0:M}$. $\mathbf{x}_{0:T}$ is the robot pose $\mathbf{x}_t = \Big[x_t, y_t, \theta_t\Big]$ on a two-dimensional plane with position given by x_t and y_t and orientation given by θ_t, for all times $t \in \Big\{0, 1, 2, \ldots, T\Big\}$. $\mathbf{x}^{0:M}$ is the positions $\mathbf{x}^m = \Big[x^m, y^m\Big]$ of some arbitrary features in the environment. The variable $\mathbf{u}_t$ relates adjacent poses in time, so in this case is defined as $\mathbf{u}_t = \Big(\Delta\xi_t, \Delta\theta_t\Big)$, where the two elements correspond to the forward motion and the angular motion respectively. The variable $\mathbf{z}_t^m$ relates robot poses with features in the map. In this case it is defined as $\mathbf{z}_t^m = \Big(\Delta\xi_t^m, \Delta\theta_t^m\Big)$. The functions f_u and f_z give the predicted value of the variables $\mathbf{x}_t$ and $\mathbf{z}_t^m$ respectively, given the current model estimate, described by

$$f_u(\mathbf{u}_t, \mathbf{x}_{t-1}) = f_u(\Delta\xi_t, \Delta\theta_t, \mathbf{x}_{t-1}) = \begin{bmatrix} x_{t-1} + \Delta\xi_t \cos\theta_{t-1} \\ y_{t-1} + \Delta\xi_t \sin\theta_{t-1} \\ \theta_{t-1} + \Delta\theta_t \end{bmatrix} \tag{3.2}$$

and

$$f_z(\mathbf{x}_t, \mathbf{x}^m) = \begin{bmatrix} \sqrt{(x^m - x_t) + (y^m - y_t)} \\ \arctan\dfrac{(y^m - y_t)}{(x^m - x_t)} - \theta_t \end{bmatrix} \tag{3.3}$$

Finally, $\Omega_{u,t}$ and $\Omega_{z,m,t}$ are the inverse of the uncertainty in each measurement $\mathbf{u}_t$ and $\mathbf{z}_t^m$, which can vary with time t and measurement m.

Due to the nonlinear functions f_u and f_z, the problem becomes nonlinear least-squares optimization, which can be solved as iterative linear least-squares optimization. This is implemented in Matlab using a Gauss-Newton approach. In effect, the estimate $\Big[\mathbf{x}_{0:T}, \mathbf{x}^{0:M}\Big]$ is iteratively improved to reduced the weighted sum of the squared difference between the uncertain measurements made by the robot and the measurements that would be predicted by the estimate.

3.3 Results

Given the method for simulating a robot trajectory through a pipe network and acquiring measurements representative of the robot's limited sensing, the effectiveness of the method for estimating the most likely trajectory given these uncertain measurements can be analysed.

The magnitudes of linear and angular uncertainty can be varied, as can the proportion of the map which is available as prior knowledge. The estimate error can be measured using the euclidean distance between each estimated pose and the true pose. An error *rate* can be defined as the proportion of estimates which have an error value above a threshold. The threshold used here is 1 metre, based on the problem definition given in Chapter 1.

Figure 3.2 gives an example of the trajectory estimation in this experiment. The *dead-reckoning* trajectory estimate can be seen to *drift* from the true path, with its error increasing over time. The *GraphSLAM* estimate is seen to be improved by the use of prior knowledge of the environment, as the estimate error is lower near to parts of the environment where prior knowledge is available. The *GraphSLAM* estimate is seen to be self-consistent even when the estimate error is nonzero. However, in this example, most of the estimate is above the target accuracy of around 1 metre, so would be ineffective in localizing faults in a pipe network.

Quantitative analysis requires variation over the various parameters which produce uncertainty in the estimate. The trajectory simulation and estimation is repeated 20 times and the average error in estimates can be compared. The range of linear uncertainty used is representative of the uncertainty in odometry estimates found in the literature for robots in pipes of up to around 30% of the distance travelled, described in Chapter 1 in Table 1.3. For this experiment, it is estimated that there is a factor of approximately 0.45 between the linear uncertainty parameter varied here and the relative error in estimate of distance travelled down a single linear pipe section.

Figure 3.3 shows the variation of estimate error with linear noise, with zero angular noise, and a range of amounts of prior knowledge. The estimate error is seen to increase

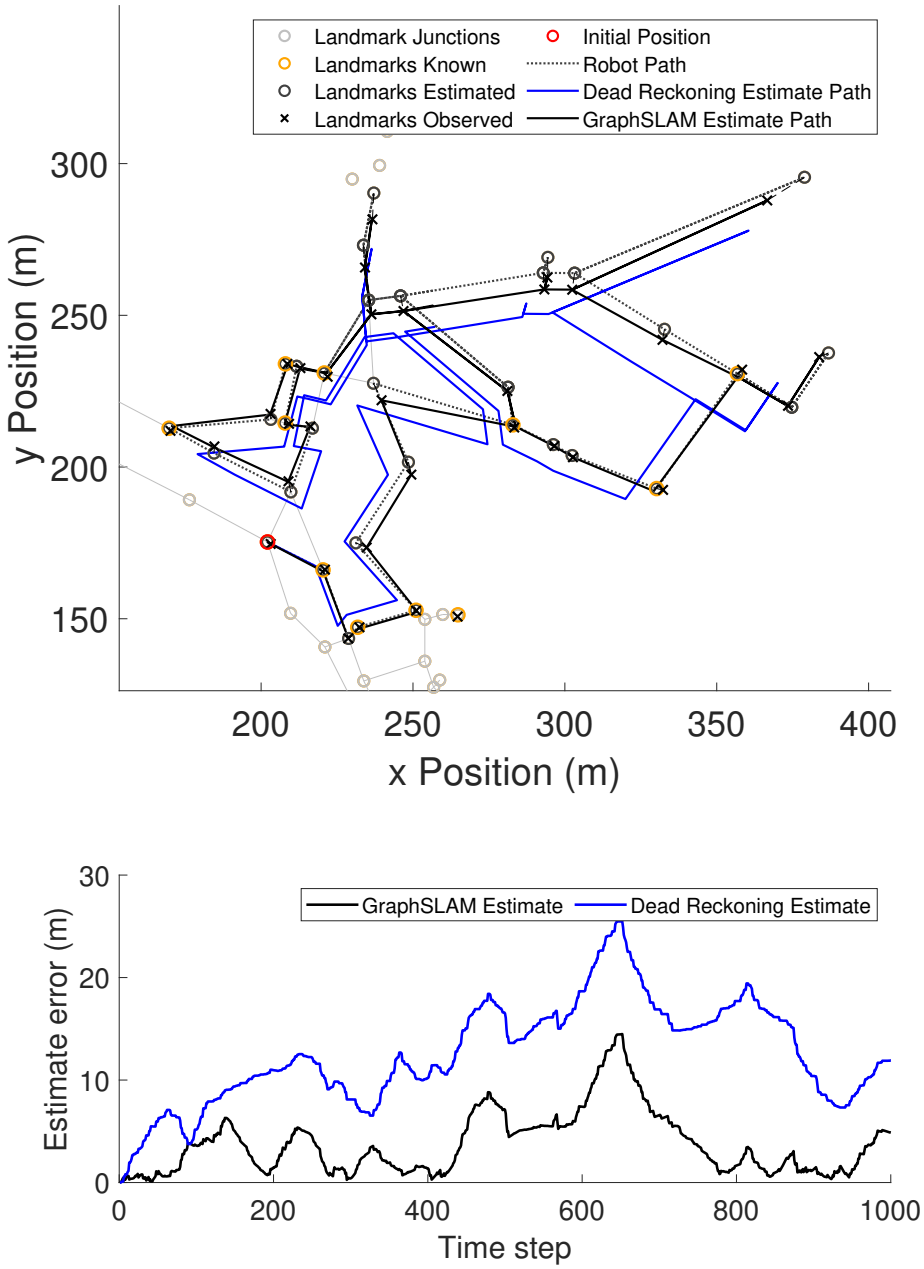

Figure 3.2: *An example of the trajectory estimate made using dead reckoning and using GraphSLAM. Part of the pipe network environment can be seen along with the true robot path. Prior knowledge of some of the landmarks is available. The error in the two estimates is shown over time.*

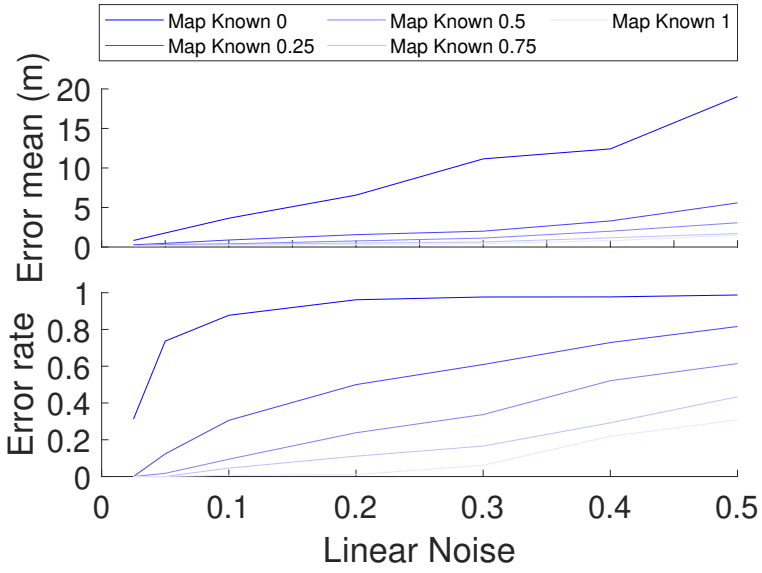

Figure 3.3: *The variation in trajectory estimate error with linear noise, for a range of amounts of prior knowledge.*

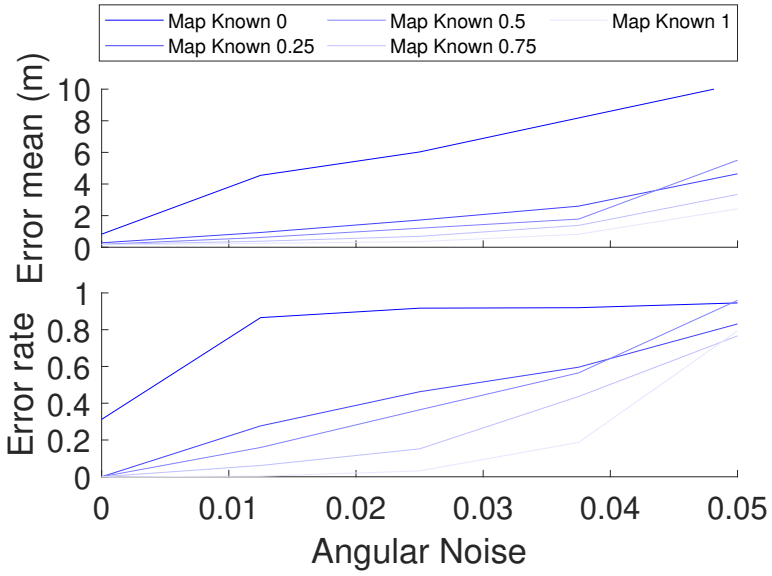

Figure 3.4: *The variation in trajectory estimate error with angular noise, for a range of amounts of prior knowledge.*

linearly with linear noise, by an factor depending on the amount of prior knowledge available. The estimate error rate therefore also increases with linear noise, quickly becoming close to 1 when no prior knowledge is available, and increasing less for larger amounts of prior knowledge. Even for full prior knowledge of the environment the estimate error and error rate increase. This is expected, as while the prior knowledge of network junctions can improve the estimate accuracy at junctions, no additional prior information is available to improve the estimate in between these junctions, and therefore these parts of the estimate will have some nonzero error.

Figure 3.4 shows the variation of estimate error with angular noise, with low linear noise of 0.025 metres, and a range of amounts of prior knowledge. As with the variation in linear noise, the estimate error and estimate error rate increase with increasing angular noise. When full prior knowledge of the environment is available, a low error mean is expected for all values of angular noise, as the prior knowledge should be able to correct any error in angular estimate, an there is no contribution to estimate error in the linear pipe sections between junctions from angular uncertainty. However, it is seen that the error increases substantially, showing that the algorithm is unable to correctly estimate the trajectory in these conditions.

3.4 Conclusions

Overall, this quantitative assessment of a typical localization back end using a range of measurement inputs has given some useful insight.

It can be seen that this typical pose-graph optimization method has low accuracy with the relatively sparse measurements in this environment caused by the limited perspective of sensors. However, while the lack of constraint to the network allows efficient nonlinear least-squares optimization in a continuous space, parts of the estimate will likely be distant from the real pipe network. This will impact the accuracy of the localization of features in the network, such as faults detected by the robot, in global coordinates.

Prior knowledge and recognition of some of the nodes in the network is seen to

be very effective at improving the accuracy of the estimate. This information effectively constrains the estimate at various points, removing the uncertainty in the estimate at these points, and reducing the accumulated uncertainty between these points. Even having prior knowledge of only a small proportion of nodes is seen to give a large improvement to accuracy in terms of the mean error. However, there is less improvement to the error rate, showing that at this scale the even the low end of the range of values of linear uncertainty reported in the literature will cause much of the estimates to be outside of the target accuracy threshold for the medium scale localization along a length of a pipe. Prior knowledge of a large proportion of the network is needed to produce a low error rate for most values of linear motion uncertainty. However, even with full prior knowledge of the environment, angular motion uncertainty has a large effect on the error rate, as the localization algorithm in continuous space is unable to function as expected.

These conclusions inform the remaining work presented in this thesis. Prior knowledge of the network is seen to be very useful, and the cost of acquiring it, which could range from using existing network maps to surveying the area of operation to record the position of the observable features above ground, would likely be worth the subsequent improvement to localization. This prompts the investigation into localization, rather than *localization and mapping*, in Chapter 6. A *localization and mapping* approach could certainly be valuable, but is beyond the scope of this thesis. Problems and inefficiencies are found with the continuous space localization algorithm used in this chapter, which motivates development of discrete and hybrid continuous-discrete space definitions used in Chapter 6 and Chapter 7. The limited perspective of sensors is seen to be detrimental both at the large scale localization over a network and medium scale localization along a pipe. Improvements to sensing are developed in Chapter 4 and Chapter 5.

Chapter 4

Front-End: Hydrophone Localization

4.1 Introduction

Chapter 3 evaluated the typical localization approach in the context of limited sensing in the pipe environment. There it was seen that at a large scale, information found at the connections between pipes in a network has a large effect on overall accuracy of the estimate. However, even with full prior knowledge of the environment, error rate was found to increase with uncertainty in linear motion, due to the error in estimation at the medium scale of localization along a single pipe.

Therefore, detection and recognition of features observed as the robot moves along a single pipe could improve the error rate of the trajectory estimate at the medium and large scale. However, Chapter 1 and Chapter 2 concluded that, compared to typical robot environments, features found using a typical vision approach are quite sparse, so recognition of arbitrary features in the pipe could be difficult using conventional means.

Therefore, this chapter has two aims: firstly, to continue the development of a means of acquiring information from arbitrary features in a pipe which is sparse in typical features,

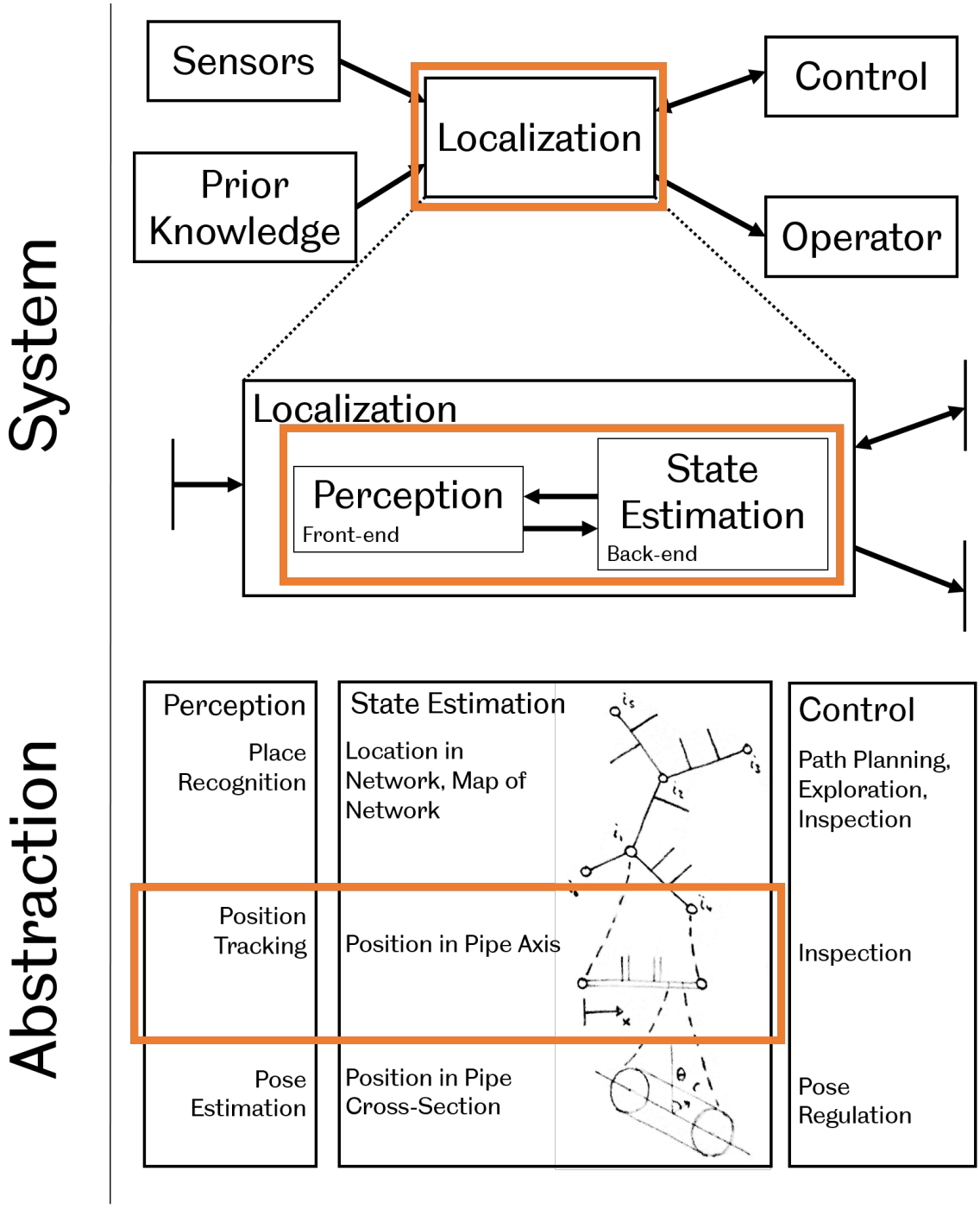

Figure 4.1: *An illustration of the scope of this chapter, with the orange boxes indicating the scope in the context of the rest of the localization system.*

and secondly, to measure the improvement in accuracy that can be found by recognising features in a pipe.

This chapter contains work previously published in the paper Rob Worley et al. "Robot localization in water pipes using acoustic signals and pose graph optimization". In: *Sensors (Switzerland)* (2020), pp. 1–23. DOI: 10.3390/s20195584. URL: https://www.mdpi.com/1424-8220/20/19/5584. This paper continues previously published work [135, 33] of a coauthor Ke Ma. A localization algorithm is developed and evaluated here using experimental and simulated hydrophone data. This experimental data was recorded by Ke Ma, Michele M. Schirru, and Gavin Sailor. Figure 4.1 illustrates that this chapter considers aspects of both perception and of state estimation; this covers both the acquisition of information from an atypical hydrophone acoustic sensor, and the incorporation of this information into a localization estimate. Figure 4.1 also shows that the localization problem is confined only to a medium level of abstraction, considering the position of a robot along the axis of a single pipe.

In previous work, a hydrophone emitter-receiver positioned on a robot has been used to perform localization [135, 33]. The hydrophone emits a sound wave that interacts with a pipe in a way that varies over space and so can be used to recognise location. The limitation of previous work is that only *online* localization is done, which only estimates the current location. Instead, in this work, the full trajectory is estimated which would allow better estimation of the location of faults in the pipe detected by the robot. The objective, information available, and methods used here are therefore different. This chapter presents a novel solution for full trajectory estimation by augmenting a pose graph optimization method.

As described in Chapter 2, pose-graph optimization uses efficient, sparse, nonlinear least squares methods to estimate the robot trajectory. This approach has been used to estimate the large-scale robot trajectory in a network of pipes in Chapter 3, and the same principles can be applied here to the medium scale, more precise localization in a single pipe. These back-end estimation algorithms are commonly usable across application domains. However, the front-end construction of the pose graph from sensor data can depend on the application. Three alternative methods for constructing the pose graph from the spatially

varying hydrophone acoustic signal are proposed and evaluated here: 1. quadratic fit, 2. cross correlation, 3. phase correlation.

Chapter 2 described methods of using measurements of the ambient magnetic field in an environment to improve robot localization. In much of this previous work, the robot is moving in a two-dimensional environment, occasionally revisiting locations, and so the approach is taken to find matches between measurements of the magnetic field. An approach similar to this can be applied to the hydrophone measurements, which vary in a similar way through the similarly one-dimensional environment. An expected limitation of the approach in previous literature is the dependence on reliable matching between measurements, which may not be possible.

The contribution of this chapter is that this work achieves a soft loop-closing effect rather than only explicitly matching subsets of measurements to find loop-closures. This exploits the continuous nature of the measured spatial field to reduce the sensitivity to robustness in feature detection and matching. This also exploits the fact that in this application the robot is revisiting every point along its trajectory, rather than only occasionally revisiting previously observed locations. This is shown to reduce the error rate in localization.

In order to evaluate the pose graph optimization algorithm, data recorded in a 5 metre long metal pipe filled with water, which is an order of magnitude longer than in previous work [135, 33], is used. The performance of the algorithm is analysed in simulation over much larger scales than can be achieved in a laboratory, which relates more closely to real world pipe networks as described in Chapter 1. The simulations are based on synthetic data that are derived from the experimental data to provide realistic evaluation.

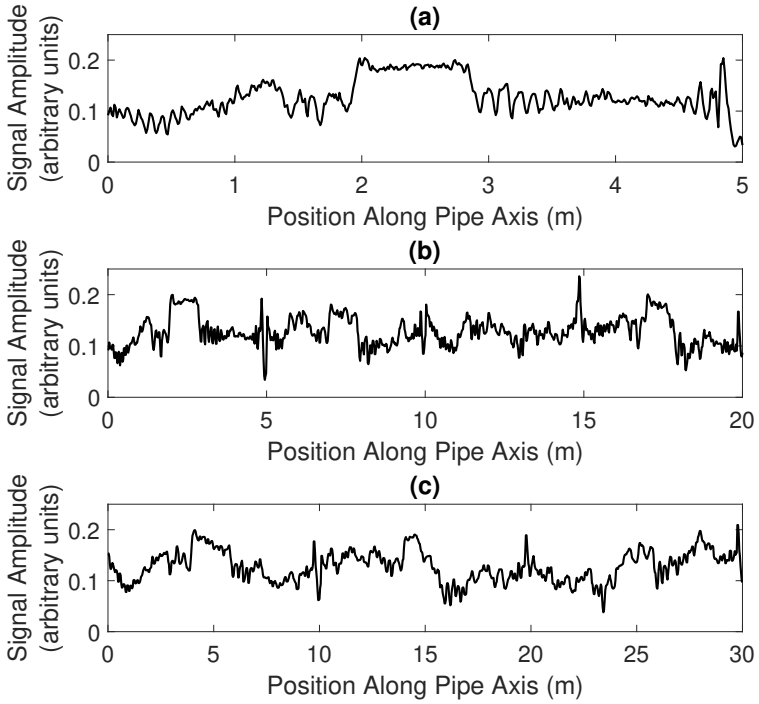

Figure 4.2: *(a) Experimental acoustic signal data. (b) and (c) Synthetic acoustic signal data.*

4.2 Methods: Acoustic Signal for Robot Localisation

4.2.1 Experimental Data

Buried pipes exhibit a vibration when excited by a sound wave. This vibration varies over space and so can be used like a map for localization. The spatial field used in localization is defined as a one-dimensional acoustic signal s_t varying along the pipe, observed at sample time t.

Experimental measurement of a spatially varying acoustic signal s_t is available in the literature [160]. This data was recorded using a hydrophone in a metal pipe to excite a vibration in the kilohertz range in a 5 metre long water-filled pipe at 0.01 metre intervals along the axis of the pipe. Figure 4.2(a) shows the experimentally measured spatial signal [161].

4.2.2 Synthetic Data

To evaluate the robot localization method over additional data sets and a larger spatial scale, synthetic data sets are created using the experimental data. Figures 4.2(b) and 4.2(c) show a signal simulated using the experimental data, which is used to extend the assessment of the developed methods to challenging signals with some periodicity and similarity between sections.

The robot's motion is modelled at discrete time steps along the pipe, considering only the motion along the axis of the pipe, and a measurement of the spatial field is taken at each point. A trajectory is therefore made up of a set of discrete positions $\mathbf{x}^r_{0:T}$ and corresponding measurements of the spatial signal $\mathbf{s}_{0:T}$ and measurements of landmarks $\mathbf{z}_{0:T}$ if the robot is near to one. The robot's velocity between each step is modelled as a random process which results in drift in velocity from integrated additive Gaussian noise, giving uncertainty in position, similar to in other work [69]. This is intended to model the effect of variation in the robot's pipe axis velocity depending on the small robot's real orientation and forward velocity within the pipe.

4.3 Methods: Robot Localisation using Pose Graph Optimization

4.3.1 Pose graph optimization using a spatially varying signal

This section first defines the typical pose-graph optimization problem that can be used to localise a robot in a pipe with respect to conventional features, such as pipe junctions and corners. Secondly, the pose-graph optimization problem is augmented to allow flexible use of a measured spatially varying signal.

It is assumed that a robot moves along a pipe according to the motion model

$$x^r_t = f_u(u_t, x^r_{t-1}) + w_t \tag{4.1}$$

where x_t^r is the robot position in one dimension along the axis of the pipe at sample time t, $u_t = \Delta x_t$ is an input motion, f_u is the state transition function and w_t is Gaussian random state noise where $w_t \sim \mathcal{N}(0, \Sigma_{u,t})$. In this case, f_u is given simply by the linear function

$$f_u(u_t, x_{t-1}^r) = x_{t-1}^r + u_t \tag{4.2}$$

however, for generality, the notation f_u will be used here.

The robot is able to detect *features* such as junctions and corners in the pipe section, where it is assumed that data association, the problem of estimating which measurements correspond to which features, is known, giving the model

$$z_t^m = f_z(x_t^r, x^i) + v_t \tag{4.3}$$

where z_t^m is the measurement at time t of landmark feature m, x^m refers to environment features m, f_z is the nonlinear measurement function and v_t is Gaussian random measurement noise where $v_t \sim \mathcal{N}(0, \Sigma_{z,t})$. The robot only makes measurements of *features* when they are nearby, so for many time indices t there will be no measurements. Here is assumed that there are two junction features, one at each end of a straight pipe.

The terms in equations 4.1 and 4.3 form the typical pose graph optimisation problem [30], which is defined by the cost function

$$\begin{aligned} J = & x_0^{rT} \Omega_0 x_0 \\ & + \sum_t^T (x_t^r - f_u(u_t, x_{t-1}^r))^T {\Sigma_{u,t}}^{-1} (x_t^r - f_u(u_t, x_{t-1}^r)) \\ & + \sum_t^T \sum_m^M (z_t^m - f_z(x_t^r, x^m))^T {\Sigma_{z,t}}^{-1} (z_t^m - f_z(x_t^r, x^m)) \end{aligned} \tag{4.4}$$

where x_0 is the initial state which has uncertainty Ω_0^{-1}, and T and I are the number of time steps and features respectively.

It is desired to simultaneously optimize the trajectory estimate with respect to the feature measurements and acoustic signal measurements. This would allow the incorporation of prior knowledge, more feature measurements, or more signal measurements without further alteration of the problem. Discrete *features* such as recognisable sequences of measurements

could be detected in the signal which and recognised at different points in time, and could be incorporated using equation 4.3. The novel approach taken here is to augment the cost function with an additional term ϕ which defines a measure of inconsistency in the estimated spatial signal measurements along the pipe. This allows the continuous spatial signal information to be incorporated without requiring detection of discrete features. This augmented cost function is given by

$$\begin{aligned} J = & x_0^{rT} \Omega_0 x_0 \\ & + \sum_t^T (x_t^r - f_u(u_t, x_{t-1}^r))^T {\Sigma_{u,t}}^{-1} (x_t^r - f_u(u_t, x_{t-1}^r)) \\ & + \sum_t^T \sum_m^M (z_t^m - f_z(x_t^r, x^m))^T {\Sigma_{z,t}}^{-1} (z_t^m - f_z(x_t^r, x^m)) \\ & + \sum_t^T \phi(t, \mathbf{x}_{0:T}^r, \mathbf{s}_{0:T})^T {\Sigma_{s,t}}^{-1} \phi(t, \mathbf{x}_{0:T}^r, \mathbf{s}_{0:T}) \end{aligned} \tag{4.5}$$

where x_0 is the initial state which has uncertainty Ω_0^{-1}, and T and I are the number of time steps and features respectively. $\mathbf{x}_{0:T}$ and $\mathbf{s}_{0:T}$ are positions and acoustic measurements along the pipe, and $\Sigma_{s,t}$ is the covariance of this field measurement model noise.

The function ϕ generally has the form

$$\phi(t, \mathbf{x}_{0:T}^r, \mathbf{s}_{0:T}) = y(t, \mathbf{x}_{0:T}^r, \mathbf{s}_{0:T}) - f_s(t, \mathbf{x}_{0:T}^r, \mathbf{s}_{0:T}) \tag{4.6}$$

where y is a function giving some measurement of the signal $\mathbf{s}$ and f_s is a function giving the expected value of that measurement, much like the terms z and f_z respectively. Unlike the typical cost function terms, this term can be a measure of difference in the abstract spatial signal quantity, rather than difference in position in space.

The solution is sought as

$$\mathbf{x}^* = \arg\min_{\mathbf{x}} J(\mathbf{x}) \tag{4.7}$$

where $\mathbf{x}$ is given by

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_{0:T}^r \\ \mathbf{x}^{0:M} \end{bmatrix} \tag{4.8}$$

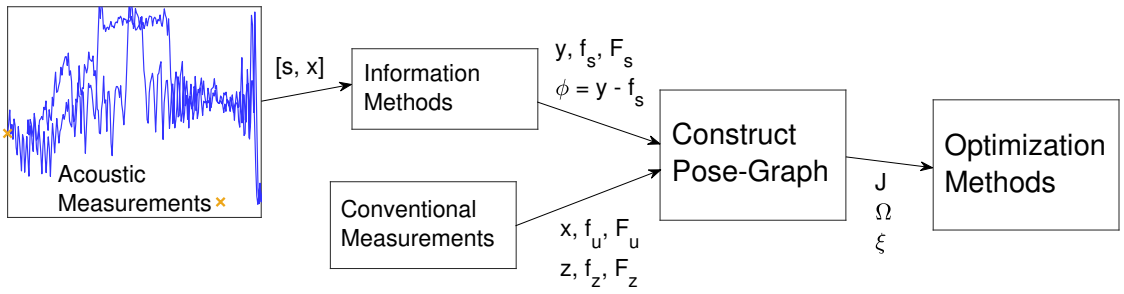

Figure 4.3: *An illustrative block diagram outlining the process of incorporating spatially varying acoustic signal measurements into pose-graph optimization.*

This optimization problem is solved iteratively. The solution can either use analytical methods, where the gradient (or Jacobian) in the solution space is known, or can use numerical methods to compute this gradient.

The process described here and in the rest of Section 4.3 is illustrated in Figure 4.3.

4.3.2 Spatial Signal Information Methods

This section presents two specific functions which can incorporate the spatial signal information into pose-graph optimization. In this application, the robot moves twice along a pipe, recording measurements of the spatial signal of vibration amplitude. These measurements of the spatial signal can be used to improve the estimate of the robot's trajectory.

The measured spatial signals will be distorted along the spatial axis due to the uncertainty in the position along the pipe at which each measurement was taken. Aligning the two signals, within the other constraints of the pose graph, increases the likelihood of accuracy of the corresponding set of poses.

In this section, methods for incorporating information from the spatial signal measurements to the pose graph are described, represented by the *information methods* block in Figure 4.3.

4.3.2.1 Quadratic Fit Prediction

For each pose, the poses and measurements within a chosen distance are used to create a local quadratic fit, parameterized by θ_2, θ_1, and θ_0. It is predicted that $y_t = s_t$ will lie on this quadratic curve, as given by

$$f_{s,t} = \hat{s}_t = \theta_2 x_t^{r2} + \theta_1 x_t^r + \theta_0 \tag{4.9}$$

A multi-scale quadratic fit can be done, where the prediction is made for a number of window sizes and all of the resulting information is incorporated into the optimization.

The Jacobian is needed to incorporate the prediction into the analytical optimization, however in this case the calculation is difficult as the quadratic parameters θ_2, θ_1, and θ_0 are all functions of x_t^r. To simplify the calculation, θ_2 and θ_1 are treated as constants and the Jacobian is computed as

$$F_{s,t} = 2\theta_2 x_t^r + \theta_1 \tag{4.10}$$

This approximation of θ_2, θ_1, and θ_0 as constants means that the cost associated with this prediction is only applied to one pose, x_t^r. There are therefore limitations on the optimization methods that can be used (which are described in section 4.3.3) as the information cannot be incorporated into Matlab's `poseGraph` functions, for example.

In principle a prediction of the position of x_t^r could be made directly from the quadratic fit. However in practice, for two signals of similar value the gradient of the linear fit is very small and the prediction can be very far from the current estimate, leading to instabilities in the optimization. Signals outside of the measurement axis range of the quadratic function would have an undefined position estimate, so it is not always possible to do this at all.

4.3.2.2 Phase-Correlation Prediction

The signal is split into two sets ($[\mathbf{x}_{p1}^r, \mathbf{s}_{p1}]$, $[\mathbf{x}_{p2}^r, \mathbf{s}_{p2}]$), one for each traversal of the pipe. Interpolation is used to create two sets of points ($[\mathbf{x}_{q1}^r, \mathbf{s}_{q1}]$, $[\mathbf{x}_{q2}^r, \mathbf{s}_{q2}]$) of equal number N_q.

These two sets of points can be split into arbitrary smaller sequences of the same length, which could be denoted as $\mathbf{s}_1$ and $\mathbf{s}_2$. Phase-correlation can be used to find the relative shift between two continuous signals, which could usefully be applied here. The phase-correlation method is described by

$$\mathbf{C} = \frac{\mathbf{S}_1 \circ \mathbf{S}_2^*}{|\mathbf{S}_1 \circ \mathbf{S}_2^*|} \tag{4.11}$$

$$\mathbf{c} = \mathcal{F}^{-1}[\mathbf{C}] \tag{4.12}$$

$$\hat{c} = \max_x \mathbf{c}(x) \tag{4.13}$$

$$\Delta x = \arg\max_x \mathbf{c}(x) \tag{4.14}$$

where $\mathbf{S}_1$ and $\mathbf{S}_2$ are the Fourier transforms of $\mathbf{s}_1$ and $\mathbf{s}_2$, $\mathbf{S}_2^*$ is the complex conjugate of $\mathbf{S}_2$, $\mathcal{F}^{-1}$ is the inverse Fourier transform, and $\circ$ denotes the element-wise multiplication of the two vectors.

The values used in equation 4.6 are given by

$$y_t = 0 \tag{4.15}$$

$$f_{s,t} = \Delta x \tag{4.16}$$

The covariance can be set to be inversely proportional to $\hat{c}$. The Jacobian is simply defined as

$$\mathbf{F}_{s,t} = 1 \tag{4.17}$$

4.3.2.3 Cross-Correlation Matching

Similarly to the process used in phase-correlation, the signal is split into two sets ($[\mathbf{x}_{p1}^r, \mathbf{s}_{p1}]$, $[\mathbf{x}_{p2}^r, \mathbf{s}_{p2}]$), one for each traversal of the pipe. Interpolation is used to create two sets of points ($[\mathbf{x}_{q1}^r, \mathbf{s}_{q1}]$, $[\mathbf{x}_{q2}^r, \mathbf{s}_{q2}]$) of equal number N_q. The normalized cross-correlation coefficient, as used in related work [140], is then found between subsets of points ($[\mathbf{x}_1^r, \mathbf{s}_1]$, $[\mathbf{x}_2^r, \mathbf{s}_2]$) of smaller number ($N_s$ and N_y), which are taken across the whole set of points. The normalized cross-correlation coefficient is given by

$$\gamma = \frac{\sum_x (\mathbf{s}_1(x) - \overline{s_1})(\mathbf{s}_2(x) - \overline{s_2})}{\sqrt{\sum_x (\mathbf{s}_1(x) - \overline{s_1})^2 \sum_x (\mathbf{s}_2(x) - \overline{s_2})^2}} \tag{4.18}$$

and is a measure of the match between sets of points. Alternatively, as the magnitude of the signals is seen in Figure 4.2 to be consistent at each point along the pipe, the sum of the difference between the two subsets can also be used as a measure of a match, described by

$$\eta = \left[\sum_{x} (\mathbf{s}_1(x) - \mathbf{s}_2(x))^2 \right]^{-1} \tag{4.19}$$

A value ϵ is found from

$$\epsilon = ((1+\gamma)\eta)^2 \tag{4.20}$$

If ϵ is greater than a threshold $\tau_{\gamma\eta}$, the poses corresponding to $\mathbf{s}_1$ and $\mathbf{s}_2$ are considered matches and are added to the cost function term in equation 4.6 at poses corresponding to the centre of the matching sections, $\tilde{x}_1^r$ and $\tilde{x}_2^r$. The measured distance and expected distance are given by

$$y_t = 0 \tag{4.21}$$

$$f_{s,t} = \tilde{x}_1^r - \tilde{x}_2^r \tag{4.22}$$

Note that the choice to match the poses at the centre of each sequence is unlikely to be accurate, however further matching within these sequences would give an increase in computational complexity.

The correlation process to estimate the matching points $\tilde{x}_1^r$ and $\tilde{x}_2^r$ is dependant on a number of poses in $\mathbf{x}_{0:T}^r$, however, the simple approximation of $f_{s,t}$ in equation 4.22 allows easy computation of the Jacobian as

$$\mathbf{F}_{s,t} = \Big[1, -1\Big] \tag{4.23}$$

The covariance $\Sigma_{s,t}$, can be set to be inversely proportional to the product of γ and η, so that stronger matches are effectively weighted higher in the cost function in equation 4.4.

As described in Section 4.3.1, as this method makes an explicit *match* between two *features*, the information could instead be incorporated into typical pose-graph optimization formulation using equation 4.3.

4.3.3 Optimization Solution Methods

To minimise the cost function in equation 4.5, the typical pose-graph optimization algorithm uses the *information* form described by the parameters $\boldsymbol{\Omega}$ and $\boldsymbol{\xi}$, which is an inverse to the *covariance* form, to parameterize the probabilistic estimation. This relationship is defined as $\boldsymbol{\Omega} = \boldsymbol{\Sigma}^{-1}$, and $\boldsymbol{\xi} = \boldsymbol{\Omega}\boldsymbol{\mu}$, for the covariance, $\boldsymbol{\Sigma}$, and the mean, $\boldsymbol{\mu}$, of the probability distribution of the pose estimate $\mathbf{x}^r_{0:T}$.

The terms f_u, f_z, and f_s in the quadratic cost function in equation 4.5 can be linearized to derive an equation which is quadratic in x, the variable which is to be estimated. This gives the linearised cost function

$$J_l = k + \mathbf{x}^{rT}_{0:T}\boldsymbol{\Omega}\mathbf{x}^r_{0:T} + \mathbf{x}^{rT}_{0:T}\boldsymbol{\xi} \tag{4.24}$$

where $\boldsymbol{\Omega}$ is the information form matrix, which is a function of the uncertainty in measurements and the Jacobian of the expected measurement models, and $\boldsymbol{\xi}$ is the information form vector, which is a function of the same variables and also the measurements, expected measurements, and current pose estimate.

The construction of this cost function corresponds to the *construct pose-graph* block in Figure 4.3. The cost function in equation 4.24 can be minimized by using the relation between the information form and covariance form to iteratively update the estimate of $\mathbf{x}^r_{0:T}$. This optimization corresponds to the final block in Figure 4.3.

The use of analytical optimization requires derivation of an analytical Jacobian for the measurement and motion models. These tend to be known for the case of typical mobile robot models, however the addition of the spatial field to the estimation can require calculations which are difficult to derive in the case where large numbers of poses are used, and can be prone to instability. Therefore, simplifications have to be made when using this optimization method with some of the information methods described in Section 4.3.2. These simplifications may add some uncertainty to the system being optimized, however the iterative optimization is designed to function with uncertainty from the input data and from the linearization required, so this is not a fundamental problem.

The *Gauss-Newton* method, *Levenbert-Marquardt* method, and trust region methods, as examples, can be used to minimize a sum of squares function [162] by numerically computing the Jacobian, rather than needing it to be explicitly defined. These methods, as well as others, are implemented in Matlab's *Optimization Toolbox* in the `lsqnonlin` function, and Matlab's *Navigation Toolbox* in the `optimizePoseGraph` function.

4.4 Results

The novel pose-graph optimization algorithm using an acoustic signal defined in equation 4.5 is evaluated in this section, comparing the three methods proposed above for incorporating the acoustic signal into pose-graph optimization: *quadratic fit*, *cross-correlation* and *phase-correlation*. These are compared to pose-graph optimization without using an acoustic signal, just using landmarks *features* at each end of the pipe, as defined in equation 4.4. These pose-graph optimization methods are also compared to *dead reckoning* (*None*).

The experimental data (shown in Figure 4.2(a)) and synthetic data (which is derived from the experimental data, as described in section 4.2, and shown in Figure 4.2(b)-(c)), are used to compare the effectiveness of the developed localization methods. The objective is to estimate the trajectory of a robot that has travelled twice along a pipe.

The uncertainty in the robot's position is modelled by integrated random normally distributed noise on the robot's velocity, which results in a drifting velocity. The variance of the normally distributed noise added to the motion at each time step is equal to the noise magnitude multiplied by the command motion. The robot's motion is constrained between 0.2 and 1.8 times the command motion. This model is more complex than the model used in the pose-graph optimization, given in equation 4.1, which assumes that the velocity is simply given by a constant with additive normally distributed noise. This difference between the localization model and the simulated reality is useful, as it reflects the practical application where the model and true underlying motion process should be expected to differ.

An example of the use of the optimization methods is shown in Figure 4.4. The

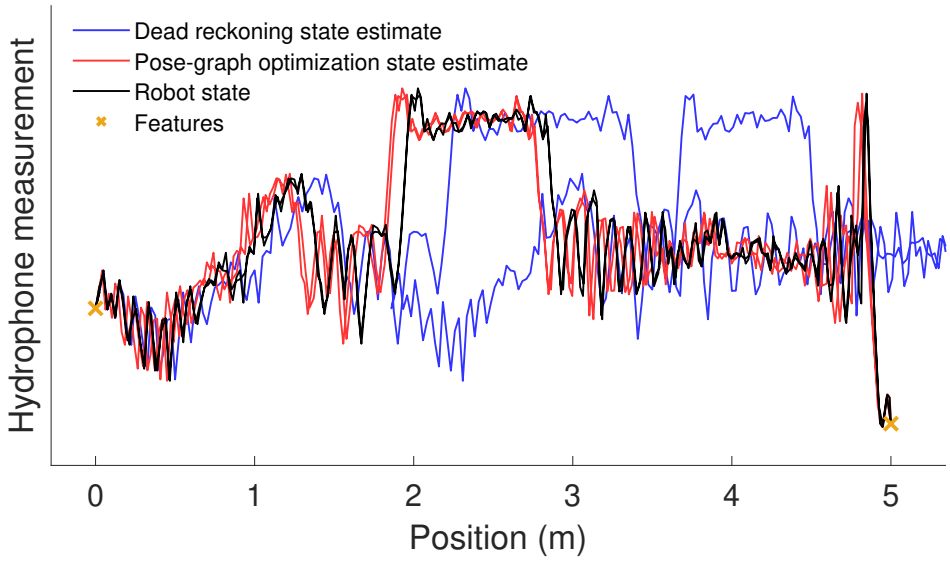

Figure 4.4: *An example of the results of the one-dimensional pipe trajectory optimization illustrated by the resulting spatial signal estimation, using the quadratic fit prediction and phase-correlation methods.*

optimized trajectory shown uses only the features at the ends of the pipe to improve the estimate. The original *dead reckoning* estimate can be seen to differ substantially from the true trajectory; the estimated trajectory is outside of the length of the pipe, and the shape of the acoustic signal is shifted along the pipe. The original estimate can also be seen to be inconsistent as the shapes of the signal recorded in each direction are not aligned. The *pose graph optimization* trajectory uses a combination of the phase-correlation and quadratic fit methods. It is seen that the estimate of spatial signal is more consistent, and more accurate when compared to the true *robot state*.

The main result is shown in Figure 4.5 where the methods are compared using fifty sets of random noise. The *error rate* is used as the metric to measure the performance of the algorithm. The error in the estimate of each point in the trajectory is calculated, and the error rate is defined as the proportion of the trajectory estimate for which the error is above a threshold. In this case, the threshold is 0.5 metres, as established in Chapter 1 as a target precision. This error rate is measured for each estimated trajectory.

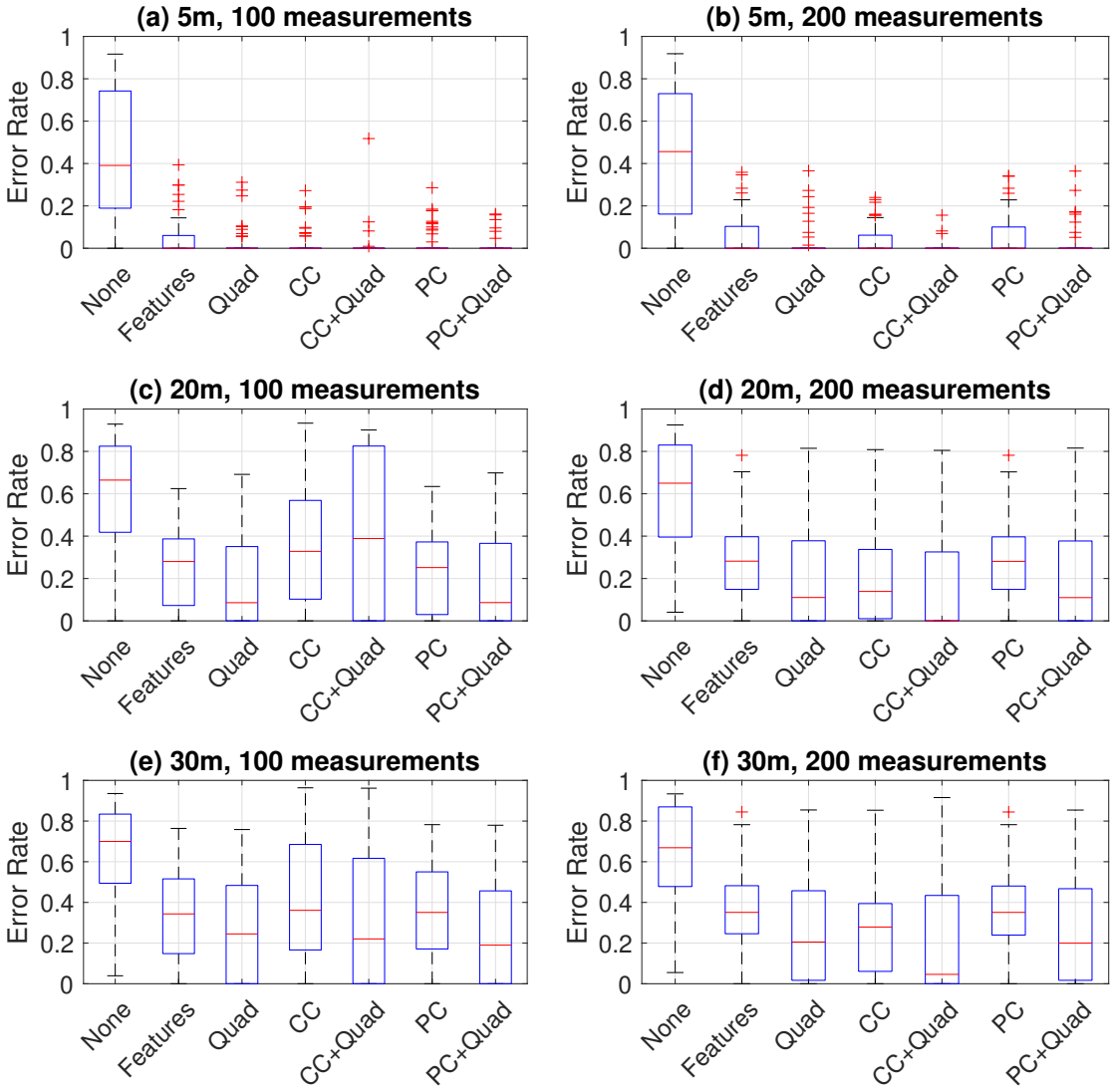

Figure 4.5: *A comparison of results from different methods of incorporating the acoustic spatial signal into localization. Three different pipes are compared, each with two different numbers of measurements.*

Figure 4.5 shows the error rate for each algorithm for three different spatial signals, 5 metres, 20 metres, and 30 metres long respectively, as illustrated in Figure 4.2. In each case, the error rate is measured for 100 and 200 measurements. Boxplots are used to show the distribution of error rate, which shows the quartile values and range, and outlier values as red crosses.

The error rate using no sensing information (*None*) is seen to be substantial in all cases, with a median of 0.4 in the 5 metre case, and with a higher value for the longer pipes. This is expected as the longer trajectories allow for more accumulation of error in motion. The use of *Features* is seen to reduced the error rate. In the 5 metre case, the median error rate is zero, although there are some larger values, while is it reduced to 0.3 and 0.35 for the 20 and 30 metre cases. These values give a baseline for comparison for the methods using acoustic sensing.

In the 5 metre case, the use of acoustic information is seen to reduce the error rate further. Where the quadratic method (*Quad*) is used, the error rate is reduced so that all nonzero values are considered statistical outliers. Similar results are seen for both 100 and 200 measurements. In the 20 metre case, it is seen that the use of acoustic information can give a reduction in error rate, but this depends on the method. The *CC* and *PC* methods give some improvement to performance, but the largest reduction in error rate is found when using the *Quad* method, sometimes in combination with the *CC* or *PC* methods. The *CC+Quad* method is seen to reduce the median error rate to zero when using 200 measurements, but gives an increase in error when using only 100 measurements, showing a sensitivity to this variable. A similar trend can be seen in the results for the 30 metre case, where the *CC+Quad* method has a median error rate of less than 0.1.

For the 20 and 30 metre cases, which represent more realistic pipe lengths as described in Chapter 1, the improvement to error rate can be quantified more easily. In the 20 metre case, the baseline *Features* method gives a median error rate of around 0.3. Relative to this, the *Quad* and *PC+Quad* methods give around a 65% reduction in median error rate, while the *CC* method gives a 50% reduction in median error rate, and the *CC+Quad* method gives a 100% reduction in median error rate when there are sufficient measurements. In the

30 metre case, the baseline *Features* method gives a median error rate of around 0.35. Compared to this, the *Quad* and *PC+Quad* methods give around a 35% reduction in median error rate, while the *CC* method gives a 15% reduction in median error rate, and the *CC+Quad* method gives a 85% reduction in median error rate when there are sufficient measurements.

Each of the methods can also be compared by the upper quartile and range of the error rate. While the median error rate can be reduced in comparison to the baseline *Features* method by using the acoustic measurement information, the upper quartile error rate is similar between the *Features* method and the acoustic methods in the 20 and 30 metre cases. Even in the lowest case of median error rate found using the *CC+Quad* method, around 50% of trajectories have a nonzero error rate. Overall, these results therefore show the limits on the improvement of the error rate of the localization estimate using this approach.

4.5 Discussion

The two aims of this chapter were firstly to develop a method of incorporating arbitrary feature measurements made along a pipe in a localization estimate, and secondly to assess the improvement to accuracy that can be found when using this information. The results are discussed here in the context of these aims.

4.5.1 Summary of Results

This work has developed a pose-graph optimization algorithm that incorporates an acoustic signal for robot localization in a pipe. The pose-graph optimization cost function in equation 4.5 was augmented with a novel term for incorporating information from a measured acoustic property that varies along the robot's trajectory. Three specific implementations methods of incorporating information from an acoustic signal were developed and are compared in the results (quadratic fit, phase-correlation and cross-correlation). It was found that the combination of quadratic fit and cross-correlation gives the best results when there are a sufficient number of measurements. For the 20 and 30 metre pipe respectively, this method gives a re-

duction in median error rate of 100% and 85%, relative to the typical pose-graph optimization method. This improvement is larger than the cross-correlation method alone, which does not require the presented augmentation to the pose-graph optimization, which gives a reduction in median error rate of 50% and 15% for the 20 and 30 metre pipe respectively, relative to the typical pose-graph optimization method. The proposed novel augmentation of the pose-graph optimization method therefore shows an improvement to localization accuracy.

In the 20 metre case, the *Quad* method alone is seen to give an improvement to performance, with a 70% reduction in error rate with 100 measurements, and a 60% reduction in error rate with 200 measurements. The improvement found is smaller in the 30 metre case, which could be due to either the properties of the spatial signal or the longer length of trajectory. In the 20 and 30 metre cases, the *CC* method is seen to give an increase to the median error rate when using only 100 measurements. This could be due to poor robustness in the feature detection and matching process as predicted. Once an incorrect match has been made, the spatial signal can become distorted so that further incorrect matches are more likely to be made in subsequent iterations of the optimization. This effect from the *CC* method varies between different cases, which indicates that the number of measurements required to facilitate this method will vary across the environment. This method therefore may not be reliable in application. In contrast, the *Quad* method is seen to perform similarly with different numbers of measurements. The augmented pose-graph approach therefore shows improved reliability and decreased sensitivity.

As described in Section 4.4, considering the upper quartile of the error rate showed that even in the lowest case of median error rate, 50% of trajectories had a nonzero error rate in the 20 and 30 metre cases. In the context of the application to locating faults in pipes, discussed in more detail in Chapter 1, there is some chance that faults are not located precisely enough to be found during excavation of the pipe.

Overall, the use of acoustic information can reduce the average trajectory estimation error rate when the novel augmentation of pose-graph optimization is used. However, the error rate is not reliably reduced, and many trajectory estimates contain erroneous estimates, which are more common in case of the longer pipes. This could be attributed to

the properties of the signal used in this case, but could be expected as the longer trajectory length allows more error to accumulate. This limited improvement to accuracy comes at the cost of increased complexity in the design of the robot and the operation of the robot. This motivates the use of alternative means of sensing which are able to more reliably improve localization performance with a lower increase in complexity.

4.5.2 Comparison of Results

This quantitative result is difficult to compare to results found in the literature. The most comparable results are from previous work using an acoustic signal to build a map for localization in a pipe [33] and other work using a magnetic field to improve localization in a drilling robot [140]. In both cases, as in this work, the robot travels twice along the same path and uses a continuous spatially varying field to improve localization. However, there are differences in the aims and measures used in these other works that make a useful quantitative comparison difficult to make.

In general, the principal difficulty is that methods in the literature compare the *dead reckoning* approach with an estimate found using both knowledge of *features* and the spatially varying signal. It is therefore difficult to decouple the effects of using a spatially varying signal to better align estimates along the axis of motion, and the effects of simply aligning the two parts of the trajectory using features. A second difficulty is that while previous work uses the *error*, this work uses *error rate* to measure the performance of the method. This measure is more useful as the result can be considered in the context of the application, however it is therefore not possible to make a direct quantitative comparison between these results.

In previous work using an acoustic field in a pipe, the error is reduced by 78%. However, in this previous work the aim is for *online* localization, rather than *full* localization. A measure of the total error would be larger for an *online* method than a *full* method, so therefore a comparison to this result is not conclusive. This previous work also uses deterministic uncertainty in motion, which limits the accumulation of uncertainty in the

trajectory estimate.

In the case of the drilling robot, the error is reported to be reduced by 81% and 98% in the two experiments. However, in this case the robot's position is estimated in three dimensions, rather than one, so the dead reckoning error is very large as the uncertainty in motion in each of the dimensions adds to the error when measured as Euclidean distance. Therefore, pose-graph optimization should give a much larger relative improvement in a larger number of dimensions, and a comparison to the results in this work would not be conclusive.A similar improvement to accuracy might be found simply by aligning the two halves of the trajectory in three-dimensional space, without consideration of the one-dimensional alignment made using magnetic sensing.

Overall, this work therefore contributes a useful evaluation of the general method of using measurements of a spatially varying property in pose-graph optimization, which improves on previous work by using a more appropriate model of uncertainty, and by measuring the trajectory error in the appropriate dimension. The results therefore show some weaknesses and limitations of the general approach which are not clear in previous work.

4.5.3 Limitations

A fundamental limitation of the approach used in this chapter is described here. As described previously, the approach is to construct an optimization problem which directly improves the self-consistency of the estimate, which indirectly improves the absolute accuracy of the estimate. While this approach is shown to be effective, the estimate error rate is not consistently low.

The limited scope of the hydrophone sensor means that while sequences of local features can be aligned to other local sequences, no measurement between distance features in space can be made. This is expected to be the same case for features obtained using vision in this environment; a camera would have a limited scope in a pipe compared to more general applications and would only be able to observe nearby features, much like the hydrophone. The consequence of this is that there is still some relative uncertainty between poses which

is unlikely to be removed by combining the measurements made by only traversing the pipe twice.

A sensor which could observe distant features, such as junctions at the ends of a pipe, would be able to give an absolute, although still uncertain, measurement between poses which would be useful for reducing the accumulated uncertainty.

4.6 Conclusions

The use of pose-graph optimization for localization of a robot in a buried pipe has been demonstrated. As an alternative to visual localization methods, three methods of incorporating information from the measurement of an acoustic spatial field have been presented and described, and compared using simulations based on experimental acoustic measurements. The developed methods are designed to be applicable to any spatially varying property along the robot's trajectory, such as magnetic or electric fields.

The proposed best method for incorporating this acoustic information into localization is a combination of a *cross-correlation* method and a *quadratic fit* method, which requires a novel formulation of the pose-graph optimization problem. In experiments using synthetic data over a 20 and 30 metre long pipe, the median error rate is reduced by 100% and 85% when using this method compared to the typical pose-graph optimization method, when sufficient acoustic measurements are available. In experiment using experimental data in a 5 metre long pipe, the range of error rate values is reduced to zero when using this method.

While the proposed novel approach shows an improvement to median error rate, there is still a nonzero amount of estimate error shown experimentally. This limited improvement comes at the cost of increased complexity, and there is therefore motivation to develop a sensing approach which efficiently reliably improves performance. The proposed sensing approach was concluded to be fundamentally limited in scope, much like vision-based sensing, and therefore methods to overcome this fundamental limitation are discussed in

Chapter 5.

Further work could also be done on a number of aspects of this approach. The use of pose-graph optimization in this work allows easy integration of other sensor measurements, so improvements to the results could be found by integrating visual and inertial sensing. The experimental evaluation could be improved by further real world testing in a number of larger scale pipes, to investigate the robustness of the method to a range of acoustic data. The method could also be extended to acoustic sensing in plastic pipes, where the hydrophone-induced vibration found in metal pipes could be replaced with ultrasonic sensing which can penetrate the plastic pipe. The sensitivity of the method to the experimental and algorithmic parameters could be investigated, and a means of finding an optimal set of parameters could be developed.

Chapter 5

Front-End: Acoustic Echo Localization

5.1 Introduction

Chapter 2 concluded that low-frequency acoustic echoes could provide measurements not possible with typical sensing methods such as video and lidar, which are limited in the pipe environment. As illustrated in Figure 5.1, this chapter develops aspects of both perception and state estimation using acoustic echoes, but focuses on these for medium-scale localization, estimating the position of a robot along the length of a single pipe.

This chapter contains work from the paper Rob Worley, Yicheng Yu, and Sean Anderson. "Acoustic echo-localization for pipe inspection robots". In: *IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems* (2020), pp. 2–7. URL: `https://ieeexplore.ieee.org/document/9235225`, and work from the paper *Acoustic Echo Sensing for Robot Localization in Pipe Networks*, Rob Worley, Yicheng Yu, and Sean Anderson, which is in preparation for submission to a journal. The former paper presents an approach in a single pipe, while the latter paper expands the acoustic echo sensing to a range of pipe configurations. Robot localization algorithms are developed and evaluated

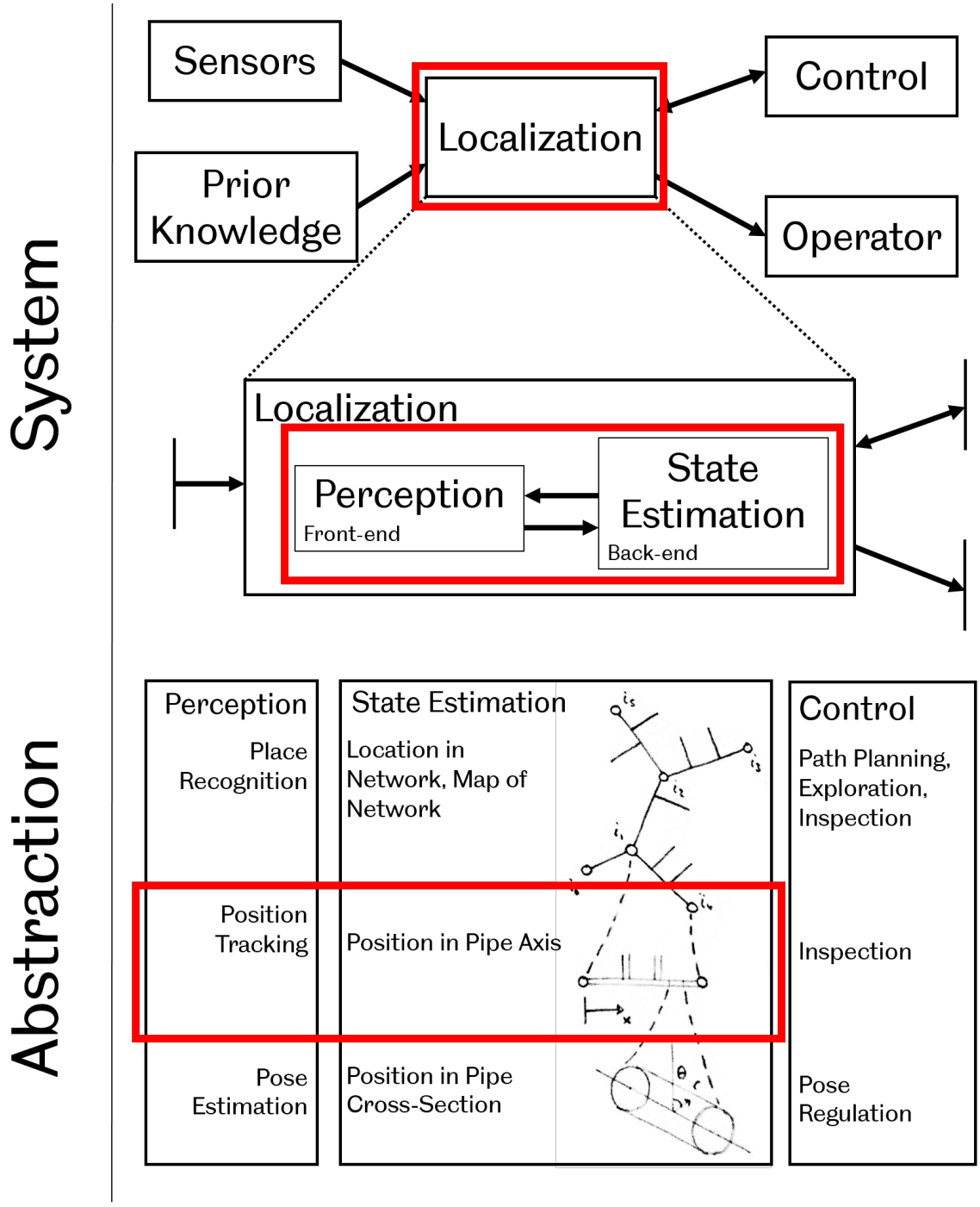

Figure 5.1: *An illustration of the scope of this chapter, with the red boxes indicating the scope in the context of the rest of the localization system.*

here using both experimental and simulated acoustic echo measurements. These acoustic echo measurements have been made in collaboration with Yicheng Yu and Gavin Sailor, which is noted again when the relevant data is discussed.

This chapter will be structured as follows: The general acoustic echo principles will be described, then the localization method and result will be presented.

5.2 Acoustic Echoes in Pipes

If a robot has a loudspeaker and a microphone, it can emit a sound and listen to the response which echoes from the environment, and use this information to make perceptions about its surroundings. The acoustic impulse response, or transfer function, between the loudspeaker and microphone is desired, which is the signal the microphone would detect if the speaker emitted a single impulse of sound.

The received signal $r(k)$ is given in discrete time, with time index k by [118]

$$r(k) = s(k) * g_{\mathbf{x}}(k) = \sum_{j=0}^{t} s(j) g_{\mathbf{x}}(k-j) \tag{5.1}$$

where the received signal r is given by the convolution of the transmitted signal s and the pipe transfer function between the loudspeaker and microphone at robot state $\mathbf{x}$ through the pipe, $g_{\mathbf{x}}$.

In practice it is not possible to directly acquire the impulse response, as an impulse contains very low energy, and it cannot be produced by a speaker. Instead, the impulse response can be estimated by sending a chirp signal containing a range of frequencies from the loudspeaker, and using deconvolution between the measured and emitted sound, given by

$$G_{\mathbf{x}}(\omega) = \frac{R(\omega)}{S(\omega)} \tag{5.2}$$

where each term is the Fourier transform of the corresponding term above.

This impulse response will be a sum of a sequence of impulses, with a time delay depending on the distance travelled between loudspeaker, source of reflection, and microphone

[120, 164, 165]. This is represented as

$$g_{\mathbf{x}}(k) = \sum_{n=1}^{N} g_n \delta(k - \kappa_n) \tag{5.3}$$

where each of N components is a Dirac delta impulse with magnitude g_n and delay κ_n.

The time delay κ_n is converted to a distance to the source of reflection ξ_n by

$$\xi_n = \frac{1}{2} c \kappa_n \tag{5.4}$$

where c is the wave speed. It is assumed that the wave speed is known, requiring calibration in the case of operation in varying temperature and humidity.

The pipe environment can add some complexity to the propagation of acoustics, while also containing the acoustic energy and therefore allowing the propagation over long distances. At higher frequencies, the acoustic propagation in the pipe becomes more complicated due to the *dispersive wave effect*, and the sound can effectively take multiple paths along the pipe, making estimation of the time of arrival of acoustic echoes difficult. Only the *plane wave* behaviour below this frequency is desired, where the sound travels directly along the axis of the pipe. Therefore, a low-pass filter is applied to the transmitted and received signals. A high-pass filter is applied to the response to remove low frequency oscillation in the impulse response, allowing easier detection of each impulse component. This filtering adds delay which is removed using cross-correlation between the filtered pipe impulse response and the impulse response of the filter itself, and oscillation which is removed using envelope detection.

5.3 Echo-Localization Measurements in Single Pipe

This section will describe the acoustic echo measurements made by a robot in a simple single pipe environment, which gives a useful illustration of the properties of the measurements.

Several experimental measurements have been made in the single pipe environment which can be used to demonstrate a localization algorithm based on acoustic echoes. Figure

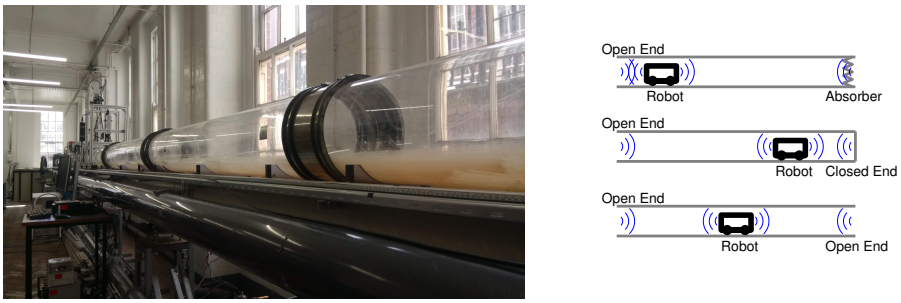

Figure 5.2: *The pipe used to record acoustic echo data, and an illustration of the robot emitting a sound which echoes from the ends of a pipe. Three different scenarios are shown, which have been created experimentally. The ends of the pipe are either open, closed, or filled with absorber.*

5.2 shows the experimental pipe, and an illustration of the acoustic echo concept in a single pipe.[1]

Figure 5.3(a) shows the signal at different states of the signal processing: filtering, correlation, and envelope detection. Figure 5.3(b) shows the response found from measurements made in the 15 metre pipe seen in Figure 5.2, where the robot is 1.5 metres from one end. Figure 5.3(c) shows the estimated impulse response in the form of Equation 5.3, found using *envelope detection*, from which measurements ξ_n can be taken, as found by a *peak detection* function which simply finds the most prominent parts of the impulse signal.

The first impulse corresponds to the direct path between the source and receiver, and is ignored in the subsequent methods. The second and third impulses correspond to first order echoes from the ends of the pipe. Figure 5.3(d) shows that as the robot moves, these impulses move accordingly, so can be used as measurements of map features in localization.

The fourth impulse corresponds to the path equal to twice the length of the pipe, giving a distance to the reflection source equal to the pipe length, 15 metres. In Figure 5.3(d) an impulse at this distance is seen for every robot position, so is a *static* measurement which, if used in the same way as the other measurements, would incorrectly appear to correspond to a feature which varies in position.

The further impulses correspond to the first order echo signals which have then

[1] These measurements were taken in collaboration with Yicheng Yu.

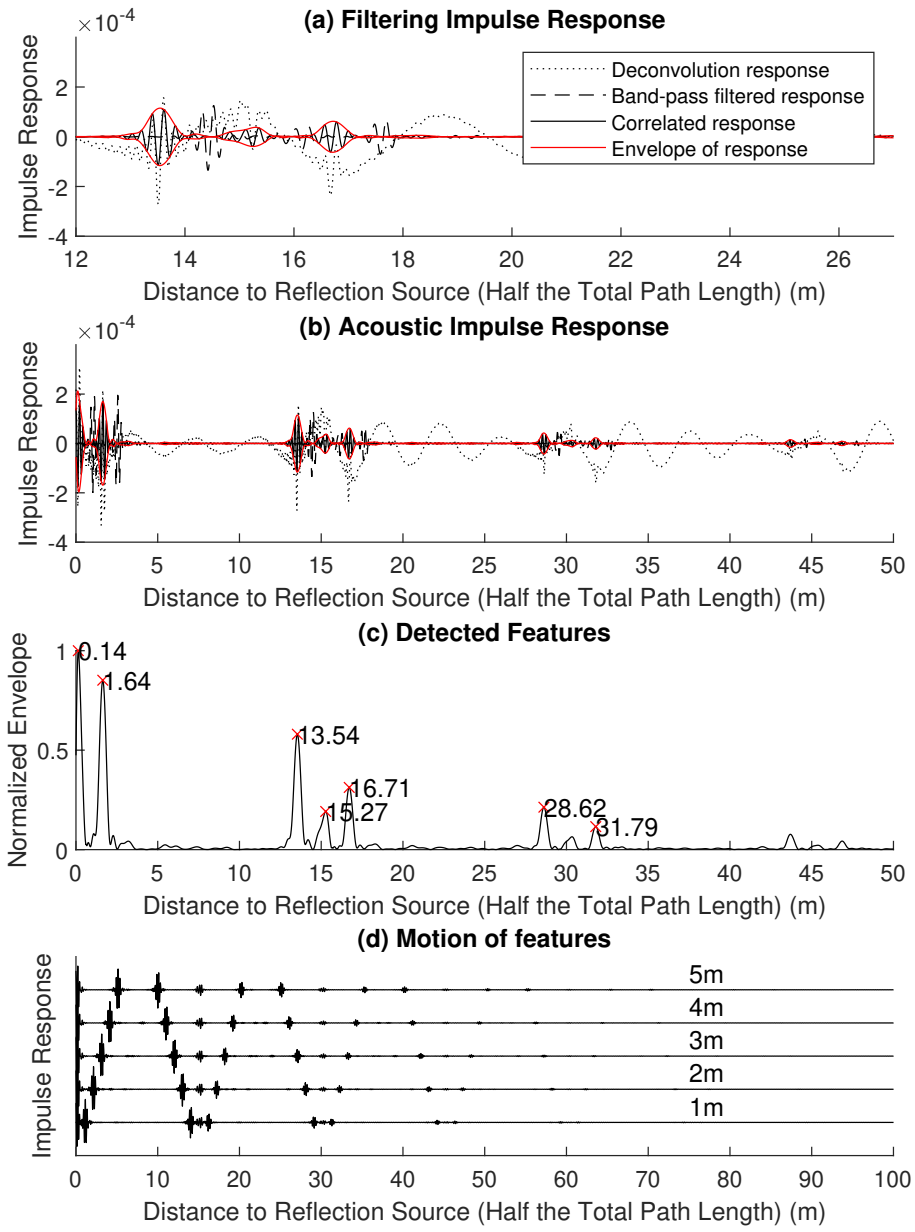

Figure 5.3: *An illustration of the acoustic signal processing used to estimate the distances to reflective features in the pipe from the robot.*

travelled the path of the *static* measurement[2], adding a further delay equal to the length of the pipe. The multiple order echoes are simply offset in distance by multiples of the length of the pipe. Therefore, they could be used as direct measurements of the positions of the echo sources if the offset can be detected and removed, or as measurements of position of fictional echo sources outside the pipe.

Evaluation of this front-end acoustic processing shows that the measured distances are well modelled as normally distributed additive noise on the true distance, with a mean of zero and a standard deviation of 0.09 metres.

5.4 Echo-Localization Methods in a Pipe Network

This section will describe the acoustic measurements, localization methods, and results from experiments in a small network of connected pipes.

5.4.1 Experimental Acoustic Measurements

In this section, the pipe in which acoustic echo measurements are made is expanded in complexity slightly compared to section 5.3. Figure 5.4(a) shows an experimental pipe which is made up of a 27.6 metre long pipe, with a 2.2 metre long lateral connection at 21.6 metres from one end of the pipe. As described in Chapter 1, in practical infrastructure, pipe networks are made up of large pipes like this which each have a number of lateral connections, connecting to each house along a road for example. This experimental pipe is therefore somewhat representative of the real infrastructure. The pipe is made up of separate sections 3.6 metres in length. Several acoustic echo measurements have been made in this pipe.[3]

In this experiment, the robot moved from a starting position of 4.5 metres from one

[2]Equivalently, signals which have travelled the path of the *static* measurement then the path of a first order echo.

[3]Yicheng Yu and Gavin Sailor collected the data presented here.

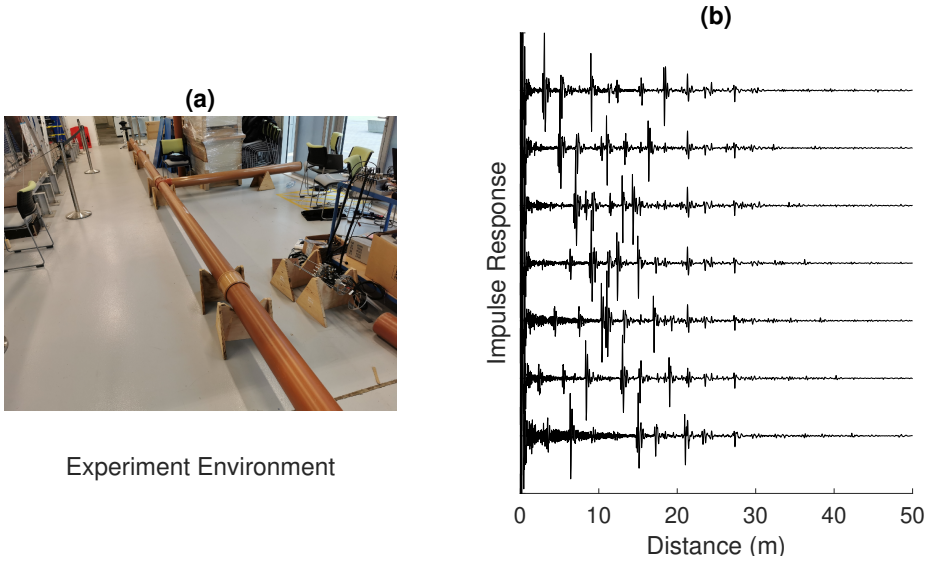

Figure 5.4: *A photograph of the small pipe network used to record experimental acoustic echo data, and an example set of acoustic echo measurements.*

end, to 18.5 metres, making an acoustic echo measurement every 2 metres, and making an odometry measurement between each acoustic measurement. Figure 5.4(b) shows a set of acoustic impulse responses measured at different positions along the pipe. As observed in the single pipe case, it can be seen that some impulses appear closer to the robot as it moves, specifically the group of three impulses corresponding to the lateral connection, the end of the lateral connection, and the end of the main pipe. Similarly, an impulse corresponding to the other end of the main pipe appears further from the robot as it moves. As well as this, impulses which do not vary in distance as the robot moves are seen, which are the *static* impulses described previously. The more complex pipe geometry means that there are more of these static impulses, which appear to correspond to the length of the main pipe, and also the distance along the pipe to the lateral connection, and this same distance plus the length of the lateral connection.

Overall, it can be seen that the slight increase in pipe complexity produces a substantial amount of additional impulses and an increase in the information observed from the environment. The additional information does, however, increase the complexity of the localization problem.

5.4.2 Problem Definition

The robot moves along the axis of a pipe by a known input u_t to position x_t at time t with the state-space model

$$x_t = x_{t-1} + u_t + w_t \tag{5.5}$$

where additive normally distributed noise w_t is added to u_t at each step. This models the uncertainty in motion along the axis of the pipe due to obstruction to motion such as debris and due to unmeasured motion off the axis of the pipe. The aim is to estimate x_t in the presence of noise w_t. After moving, the robot stops and finds the acoustic impulse response as a set of N_t distance measurements $\xi_t = \begin{Bmatrix} \xi_t^1 & \xi_t^2 & \dots & \xi_t^{N_t} \end{Bmatrix}$

In a practical application, the position of each end of the pipe may be available, as the manhole features at each end are observable above ground. As described previously, and illustrated in Figure 5.1, this chapter is focused on localization of a robot in a single pipe, so the assumption that this prior knowledge is available is sensible. In future work, if a robot is moving through a network of pipes, it may not know the length of the pipe with certainty, so this assumption may need to be relaxed. Therefore, the localization algorithms described in this Chapter are able to use prior knowledge of the position of features at each end of the pipe, which is done similarly to the approach in Chapter 3.

5.4.3 Robot Localization Methods

5.4.3.1 Map and Measurement Model

From a computed acoustic echo impulse response, the distance to sources of reflection, or features, in the environment can be estimated. It should be noted that the measurement of distance does not correspond to the euclidean distance to the feature, but instead to the distance along the pipe network to the feature. The distance measurement is therefore one-dimensional, given by ξ_t^j, which is the measurement of distance to feature j at time t.

Here it will be assumed that the pipe network is part of a drainage pipe network.

Therefore, at each end of the pipe there will be a manhole, and along the main pipe there will be an arbitrary number of lateral connections to small pipes, as described in Chapter 1. The manholes and each end of the lateral connections will all cause acoustic echoes, and so the model $p(z_t|x_t, \mathcal{M})$ can be constructed which gives the measurements that are expected at a given position x_t.

Pipe i can be described by

$$\mathcal{M}^i = \left\{ \mathcal{M}_c^i = \left\{ 0, 7, 15, 25 \right\}, \mathcal{M}_l^i = \left\{ 0, 2, 2, 0 \right\} \right\}$$

for a 25 metre long pipe with a manhole at either end, and five lateral connections which are each 2 metres long. Relative to this, the robot's position would be given in the same one-dimensional coordinates, for example $x_t = 10$. The measurement model would consist of three sets of components: The first subset would describe the echoes from each of the lateral connections and manholes, and would be a set of distributions centred around

$$\left\{ |m^j - x_t| : m^j \in \mathcal{M}_c^i \right\} = \left\{ 10, 3, 5, 15 \right\}$$

which describes the distance to features along the main pipe. The second subset would describe the echoes from the features at the end of each of the lateral connections, which is more complex as the distance also includes the length of the lateral connections, described by $\mathcal{M}_{l\Lambda}$. This is given by

$$\left\{ |m_l^j - x_t| + m_l^j : j \in \left\{ 1, \ldots, J \right\}, m_c^j \in \mathcal{M}_c^i, m_l^j \in \mathcal{M}_l^i \right\} = \left\{ 10, 5, 7, 15 \right\}$$

The third subset describes the echoes from two features, where the sound travels from the robot, to one feature, then another, then back to the robot. When the robot is between any two features in the environment, there will always be a *static* echo which travels a total distance equal to twice the distance between those two features. In this case this is given by

$$\begin{aligned} &\left\{ |m_c^j - m_c^k| + n_l^j m_l^j + n_l^k m_l^k : \right. \\ &\quad \left. \left\{ j, k \in \left\{ 1, \ldots, J \right\}, m_c^j, m_c^k \in \mathcal{M}_c^i, m_l^j, m_l^k \in \mathcal{M}_l^i : m_c^j < x_t, m_c^k > x_t \right\}, n_l^j, n_l^k \in \left\{ 0, 1 \right\} \right\} \\ &\quad = \left\{ 15, 25, 8, 18, 10, 20, 17, 12 \right\} \end{aligned}$$

While this description of the set of components is quite complicated mathematically, it is simple to implement algorithmically. It might be desirable to denote each different type of

feature as a subset of the features, for example if it was possible to distinguish between the features. However, here the manhole and lateral connection features are all interpreted by the algorithm to be from the same set of indistinguishable features.

5.4.3.2 Simple Localization Algorithm

Given the measurements z_t and the probability distribution $p(z_t|x_t)$, a practical means of implementing Bayes optimization is needed.

For comparison to the *advanced* algorithm proposed in this Chapter, a *simple* algorithm is implemented which is based on a typical robot localization approach. This simple algorithm is based on pose-graph optimization, but is designed with the assumption that *static* echoes have been detected and removed from the measurement sets, and with the assumption that the robot does not pass lateral connections along the pipe axis.

A pose-graph optimization algorithm is used to estimate the robot's trajectory and the position of reflective features, given by $x_{0:T}$, using all the measurements $z_{1:T}$ and $u_{1:T}$ simultaneously, as in equation 2.11. The algorithm aims to find a value for $x_{0:T}$ which minimizes, in a weighted least squares sense, the difference between the measurements $z_{1:T}$ and $u_{1:T}$, and the equivalent values from the models $p(z_t|x_t)$ and $p(x_t|x_{t-t}, u_t)$.

A *scan-matching* [166] approach is used to estimate correspondence between features detected in different measurement sets. The implementation of this approach here has been specialised to the acoustic echo measurements made in this case, which allows for some simplification as the space has one dimension rather than the typical two, but also requires some additional complexity as the measurements are directionless.

5.4.3.3 Advanced Localization Algorithm

Again, given the measurements z_t and the probability distribution $p(z_t|x_t)$, a practical means of implementing Bayes optimization is needed. This section describes in detail the implementation outlined in Algorithm 1.

Algorithm 1 Echo Localization Algorithm

Get input variables
input $\xi_{1:T} = \{\xi_1, \xi_2, \ldots, \xi_t, \ldots, \xi_T\}$
input $u_{1:T} = \{u_1, u_2, \ldots, u_t, \ldots, u_T\}$
input Q_t, R_t

Estimate measurement location
$[\tau_{0:T}]$ =`estimate-measurements-location`$(\xi_{1:T}, Q_t)$

Classify static measurements
$[\xi^{\Lambda}_{1:T}, \xi^{z}_{1:T}]$ = `estimate-measurements-class`$(\xi_{1:T}, \tau_{0:T}, Q_t)$

Do pose-graph optimization
$[\hat{x}_{0:T}, m]$ = `pose-graph-optimization-1D` $(\xi_{1:T}, u_{1:T}, x_0)$

Classify lateral connections
$[\xi^{c}_{1:T}, \xi^{l}_{1:T}]$ = `estimate-feature-class` $(\hat{x}_{0:T}, u_{0:T}, \xi^{z}_{1:T}, \tau_{0:T}, Q_t, R_t)$

Do pose-graph optimization
$[\hat{x}_{0:T}, m]$ = `pose-graph-optimization-2D` $(\xi^{c}_{1:T}, \xi^{l}_{1:T}, u_{1:T}, x_0)$

As in the *simple* algorithm, a pose-graph optimization algorithm is used to estimate the robot's trajectory and the position of reflective features, given by $x_{0:T}$, using all the measurements $z_{1:T}$ and $u_{1:T}$ simultaneously.

There is a nontrivial data association problem of estimating the correspondence between measurements and features previously observed in the environment. As described in Section 5.4.3.1, the sets of acoustic measurements contain useful acoustic echoes and static acoustic echoes which need to be classified and processed differently, and the useful acoustic echoes are both from features along the main pipe and from lateral connections.

Poses in the same section of pipe need to be identified to classify and process the static acoustic echoes. Algorithm 2 describes a method of doing this using only acoustic echoes. To do this, a pose-graph is constructed, where the edges between each node correspond to the probability that the two nodes, or poses, are within the same section of pipe, and therefore have the same static acoustic echoes. This probability is estimated using the *Kullback-Liebler Divergence*, and *k-means* clustering is used to estimate a which values of divergence correspond to two poses within the same section of pipe. *Spectral graph partition-*

Algorithm 2 Estimate Measurement Location

input $\xi_{1:T} = \{\xi_1, \xi_2, \ldots, \xi_t, \ldots, \xi_T\}$, Q_t

Get static measurement probability distribution over space for each measurement set
for $t = \{1, 2, \ldots, T\}$ **do**
$p(\Lambda_t|\xi_t) = \sum_n \mathcal{N}(\mu = \xi_n, \sigma^2 = Q_t)$
end for

Measure the difference between each distribution
for $(t_1, t_2) = \{(1,1), (1,2), (1,3), \ldots, (2,1), \ldots, (T,T)\}$ **do**
Get Kullback-Liebler Divergence for each pair of time steps, creating matrix D
$D_{t_1,t_2} = D(p(\Lambda_{t_1}|\xi_{t_1})||p(\Lambda_{t_2}|\xi_{t_2})) = \sum_\Lambda p(\Lambda_{t_1}|\xi_{t_1}) \log\left(\frac{p(\Lambda_{t_1}|\xi_{t_1})}{p(\Lambda_{t_2}|\xi_{t_2})}\right)$
end for
$D^\Lambda = \frac{1}{2}\left(D + D^T\right)$

Determine threshold for similar measurement sets based on divergence
$\{D_1^\Lambda, D_2^\Lambda\} = \texttt{k-means}\left(D^\Lambda, N = 2\right)$
$l = \arg\min_{k \in \{1,2\}} \{\texttt{mean}\left(D_k^\Lambda\right)\}$
$\delta^D = \texttt{mean}\left(D_l^\Lambda\right) + \texttt{std}\left(D_l^\Lambda\right)$

Estimate probability that two measurement sets were taken in the same pipe section
$\hat{p}_{t_1,t_2}^\Lambda = \hat{p}(\Lambda_{t_1} = \Lambda_{t_2}|\xi_{t_1}, \xi_{t_2}) = -\mathcal{S}(D_{t_1,t_2}^\Lambda, \mu = \delta^D, \sigma^2 = Q_t)$ *using sigmoid function* $\mathcal{S}$

Construct pose graph with adjacency matrix $\hat{P}_{0:T,0:T}^\Lambda$

Use spectral graph partitioning
Do eigenvalue decomposition
$L^\Lambda = \texttt{graph-laplacian}\left(\hat{P}^\Lambda\right)$
find $\mathbf{v}_2$ *using* $\lambda_2 = {\mathbf{v}_2}^T L^\Lambda \mathbf{v}_2$ *where* λ_2 *is the second smallest eigenvalue of* L^Λ
$t_{i \in 1,2,\ldots,I}^\Lambda = \texttt{find-peaks}\left(\Delta \mathbf{v}_2\right)$
$\tau_i = t_{i:i+1}^\Lambda, i \in 1, 2, \ldots, I$

Algorithm 3 Estimate Measurement Class

input $\xi_{1:T} = \{\xi_1, \xi_2, \ldots, \xi_t, \ldots, \xi_T\}$
input τ_i, Q_t

Estimate the static measurements for each of the sets of time
for $i \in 1, 2, \ldots, I$ **do**
$\xi_{\tau_i}^\Lambda = \texttt{find-peaks}\left(\sum_{t \in \tau_i} p(\Lambda_t|\xi_t)\right)$
$\xi_t^\Lambda = \xi_{\tau_i}^\Lambda, t \in \tau_i$
$\{\xi_t^z\} = \{\xi_t\} - \{\xi_t^\Lambda\}$
end for

Algorithm 4 Estimate Feature Class

input $\hat{x}_{0:T}$, $\xi^z_{1:T} = \left\{\xi^z_1, \xi^z_2, \ldots, \xi^z_t, \ldots, \xi^z_T\right\}$
input τ_i, Q_t, R_t

Create sets of hypothetical features $(\mathbf{x}_t, \mathbf{l}_t)$ *for each measurement set through time*
for $t \in \left\{1, 2, \ldots, T\right\}$ **do**
  for $n \in \left\{1, 2, \ldots, N\right\}$ **do**
    Project each measurement into the two-dimensional space (x, l)
    $\hat{x}_t \pm \xi^n_t = x^n_t + l^n_t$ *defines a subspace occupied by each measurement*
    Hypothesise features in the two-dimensional space using pairs of measurements
    for $m \in \left\{1, 2, \ldots, N\right\}, m \neq n$ **do**
      Compute hypothetical feature based on features n *and* m
      $l^m_t = 0$, $\xi^m_t = x^m_t = x^n_t$, $l^n_t = \xi^n_t - x^n_t$
      Create feature pair $(x_t, l_t) = (x^{m,n}_t, 0), (x^{m,n}_t, l^n_t)$ *and add to vector* $(\mathbf{x}_t, \mathbf{l}_t)$
    end for
  end for
end for

Find the most likely alignment between sets of features given $(\mathbf{x}_t, \mathbf{l}_t)$
for $t_1 \in \left\{1, 2, \ldots, T\right\}$ **do**
  for $t_2 \in \left\{t_1, \ldots, T\right\}$ **do**
    $p^{\xi} = \texttt{corr}\left((\mathbf{x}_{t_1}, \mathbf{l}_{t_1}), (\mathbf{x}_{t_2} + \delta_x, \mathbf{l}_{t_2}), Q_t\right)$
    $p^u = \mathcal{N}(\mu = 0, \sigma^2 = \Sigma_{u,t}\left(t_2 - t_1\right))(\delta_x)$
    $\Delta x_t = \arg\max_{\delta_x} \left\{p^{\xi} p^u\right\}$
    $(\mathbf{x}_{t_2}, \mathbf{l}_{t_2}) \Leftarrow (\mathbf{x}_{t_2}, \mathbf{l}_{t_2}) + (\Delta x_t, 0)$
  end for
end for

Combine sets of features
$p\left(\mathbf{m} | x^m, l^m, \xi^z_{1:T}\right) = \prod_{\tau \in \tau} \sum_{t \in \tau} \sum_n \mathcal{N}\left(\mu = \left(\mathbf{x}^n_{t_2}, \mathbf{l}^n_{t_2}\right), \Sigma^2 = Q_t I\right)$

Measurements ξ^m_t *of features* m *where* $l^m = 0$ *are added to the set* ξ^c_t
Measurements of other features m *are added to the set* ξ^l_t

ing methods are used to separate the pose graph into different discrete regions with different echo measurements.

With this estimation of relevant subgraphs, the static acoustic measurements can be estimated by comparing measurement sets corresponding to nodes in each subgraph. The probability distributions across the one-dimensional pipe axis space are summed, and the largest peaks are identified as those which likely correspond to static measurements, as they have been observed multiple times. This is described in Algorithm 3.

The pose-graph is then constructed and solved in the typical way [30], in a similar way to the approaches in Chapter 3 and Chapter 4. This optimization is denoted as `pose-graph-optimization-1D` in Algorithm 1.

The echo measurements now contain echoes from lateral connections and from the lateral pipes. These again need to be classified to improve localization and allow mapping. From one section of the pipe, lateral connections and laterals will be indistinguishable based on position. Only when observing a lateral from both directions is it possible to make a classification estimate, and it is not trivial. Algorithm 4 describes a method to make this classification. A two-dimensional space (x, l) is defined, where one dimension corresponds to the position along the pipe axis, and one dimension corresponds to distance from the pipe axis. Each echo is projected into this space, and occupies a linear subspace. It is expected that a lateral and lateral connection will exist as a pair at the same position along the pipe axis. The subspace can then be reduced again to a set of points which likely correspond to a lateral echo in the space. Sets of points in this space can be used to create a two-dimensional probability distribution across the space, by applying a Gaussian kernel to each point. The probability distributions corresponding to each set of measurements can be combined and from the resulting distribution, features can be simply classified as either laterals or lateral connections.

A second pose-graph optimization is done, named `pose-graph-optimization-2D` in Algorithm 1. This pose-graph optimization can use the classification of lateral and lateral connection features to reduce the confusion from measurements of the different types of

feature. Simultaneously, the mapping of the environment can be done, giving a result in two dimensions, in the space (x, l) defined above. Using the acoustic measurements described in Section 5.4.3.1 alone, it is not possible to determine the direction of each feature from the axis. This direction could be estimated by incorporating information from other sensors, or by using more advanced acoustic processing, but is beyond the scope of this work.

5.5 Echo-Localization Results

This section presents results which evaluate the proposed localization method. This is done first in comparison to a more simple algorithm for localization using acoustic echoes, and then done over a variation in four parameters which describe the uncertainty in the simulated robot's motion and sensing.

5.5.1 Localization Algorithm Comparison

Figure 5.5 shows the performance of the *Simple* and *Advanced* algorithms measured in simulation based on the experimental data shown in Figure 5.4. Each graph shows the *error rate* of the algorithm when estimating a trajectory along a pipe which is the proportion of time for which the trajectory estimate error is above 0.5 metres, as established as a threshold in Chapter 1. This *error rate* is shown for 25 simulated trajectories as a boxplot, which shows the median and quartile values. This measurement is shown for a range of values of odometry measurement uncertainty, where σ_u can be related to the motion model in equation 5.5 as the motion uncertainty w_t is drawn from a normal distribution with a standard deviation of σ_u. The simulated robot moves forwards by around 2.5 metres before stopping and making an acoustic echo measurement. This motion is repeated along a 50 metre long pipe. In all cases the algorithms incorporating acoustic echo sensing are compared to an estimate made using only odometry.

As described in Section 5.4.3.2, the *Simple* algorithm is designed with the assumptions that *static* measurements are removed and that lateral connections are not passed by the

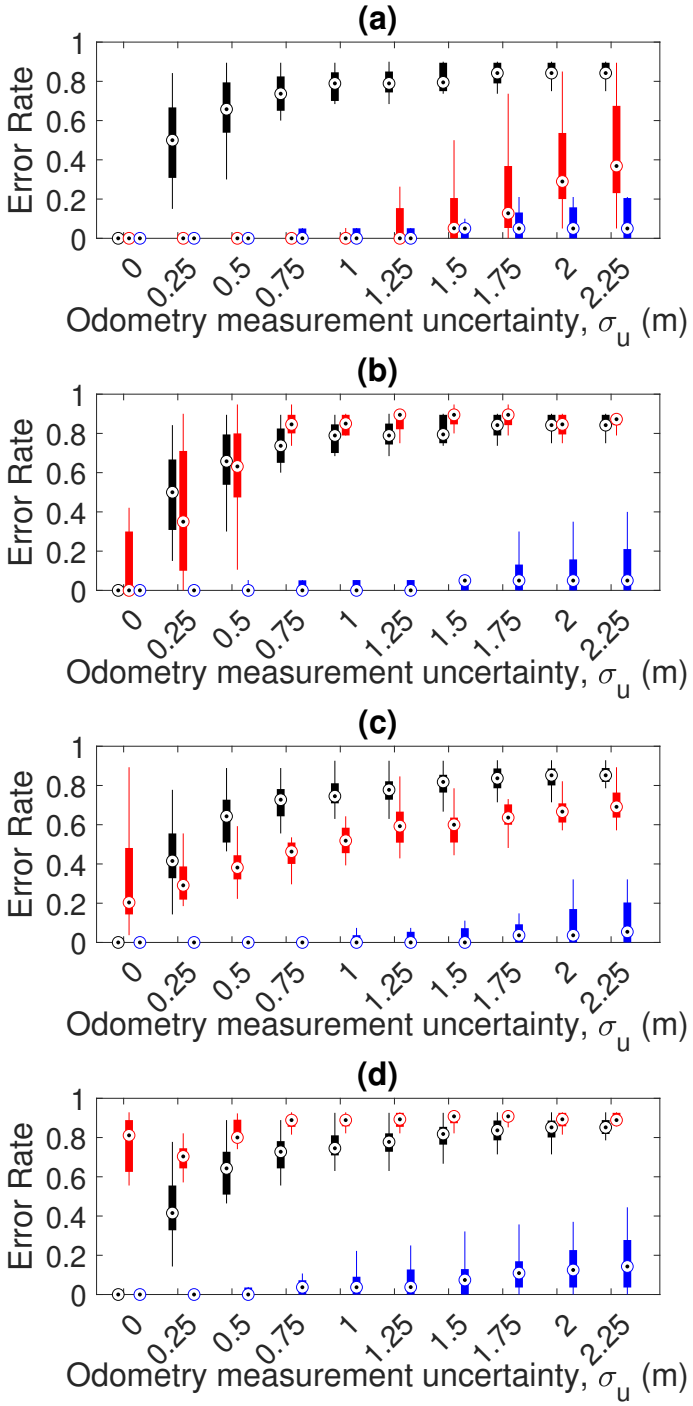

Figure 5.5: *A comparison of the Simple (red) and Advanced (blue) algorithms, in the case: (a) without static measurements or lateral connections; (b) with static measurements; (c) with lateral connections; (d) with both static measurements and lateral connections.*

robot. Figure 5.5(a) shows the performance of the *Simple* and *Advanced* algorithms in a case where these two assumptions are valid. The two algorithms are seen to improve substantially on the estimate made using only odometry for nonzero odometry uncertainty. However, the *Simple* algorithm estimate error rate increases when odometry uncertainty becomes larger. Figure 5.5(b) shows the comparison in the case that static measurements are not removed from the measurement sets. In this case, the error rate of the *Simple* algorithm is still low for very low odometry uncertainty, but increases quickly and substantially as odometry uncertainty is increased, while the *Advanced* algorithm shows a low error rate. Figure 5.5(c) shows the comparison in the case that the robot passes lateral connections along its trajectory. In this case, the *Simple* algorithm error rate becomes substantial for even the lowest odometry uncertainty, and increases as odometry uncertainty increases. Figure 5.5(d) shows the comparison in the case that both of the assumptions are not valid, so static measurements are not removed and the robot passes lateral connections, which represents the most challenging and realistic case. Here the *Simple* algorithm error rate is consistently high, while the *Advanced* algorithm error rate remains low, although it does increase with increasing odometry uncertainty.

These results show the need for the *Advanced* algorithm when incorporating acoustic echo measurements in the most realistic case, and justifies the additional complexity when using the novel approach proposed here.

5.5.2 Localization Algorithm Evaluation

The performance of the *Advanced* algorithm described in Section 5.4.3 is evaluated over a variation in different parameters describing the uncertainty in the robot's motion and sensing.

The uncertainty in the robot's motion σ_u and uncertainty in the measured distance for each measurement σ_z are varied. To model error in the front-end measurement and processing, a random number of false positive measurements can be added to the measurement set at each time step, and a random number of true measurements can be removed from the set, creating false negative measurements. This random number in each case is drawn

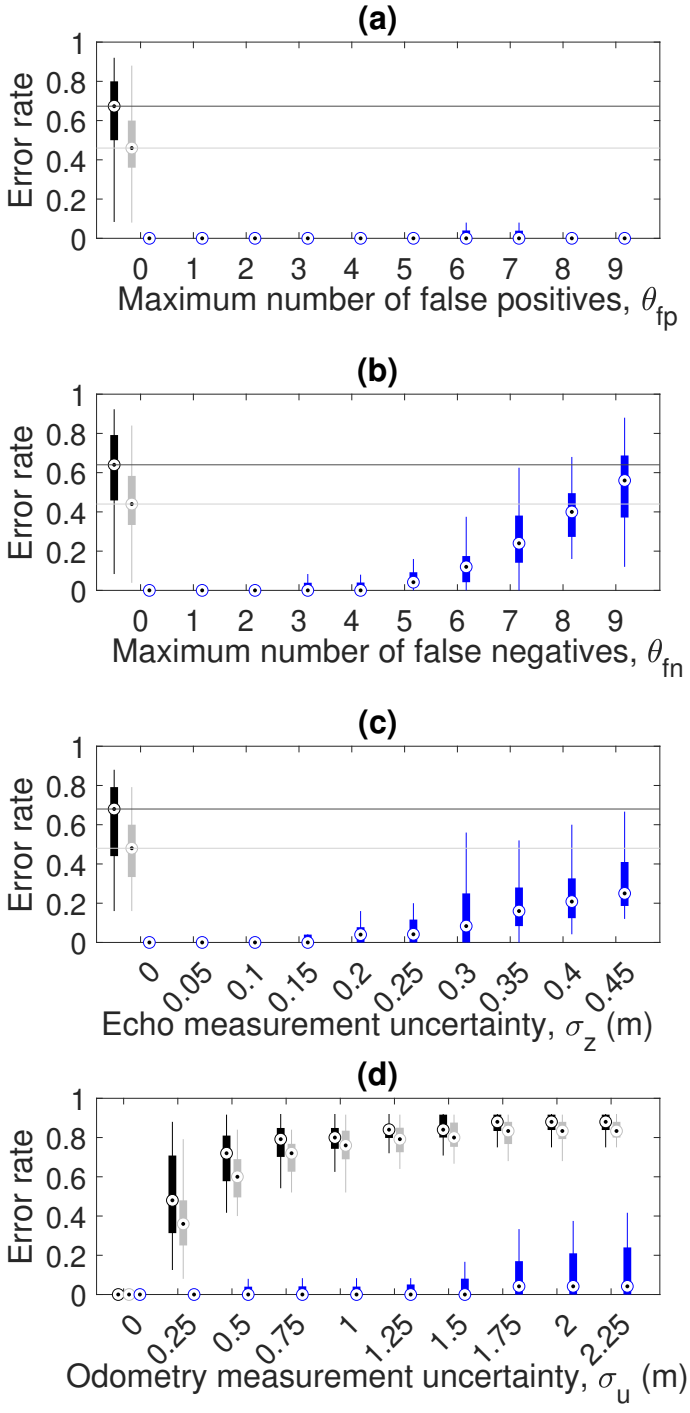

Figure 5.6: *An evaluation of the Advanced algorithm (blue) compared to the use of odometry (black) and odometry and prior knowledge (grey) over variation in magnitude of four different sources of uncertainty.*

randomly from 0 to θ_{fp} and θ_{fn} for false positives and false negatives respectively. Figure 5.6 shows the results for variation in each of these variables. The default values for each variable are $\sigma_u = 0.5$, $\sigma_z = 0.1$, $\theta_{fp} = 1$, and $\theta_{fn} = 1$.

Simulation is used to measure the algorithm error rate over 20 trajectories. A 75 metre long pipe is simulated, with three lateral connections. The robot is set to move around 2.5 metres between making each acoustic echo measurement, so each trajectory is made of around 30 positions. For comparison, two odometry-based methods are also used. The first uses only odometry, and the second uses odometry and prior knowledge of the position of each end of the pipe, as described in Section 5.4.2.

Figure 5.6(d) shows that the error rate of the two odometry-based methods increases quickly with increasing odometry uncertainty. The use of prior knowledge is seen to reduce the error rate in each case slightly, but not substantially. This is different to the results seen in Chapter 4, where the use of prior knowledge of the pipe length decreased error rate. In the case here the pipe is around three times the length used in Chapter 4, and while the estimate error at each end of the pipe will be zero, there is a longer length over which error can accumulate, and the probability is higher that an estimate will be above the threshold of 0.5 metres.

Figure 5.6(a) shows that the *Advanced* algorithm is almost unaffected by false positive measurements, as the error rate does not increase over the tested values. As the false positive measurements of distance to features are drawn from a random distribution, it is unlikely that any of them would take a value which could cause errors in localization. Conversely, Figure 5.6(b) shows a substantial sensitivity to false negative measurements. A function which detects measurements in the echo impulse response would have some balance between probability of false positive and false negative detection. From these results, it can be concluded that such a function should aim to prevent false negatives at the cost of increased likelihood of false positives, to improve the subsequent localization estimate.

Figure 5.6(c) shows that the algorithm's performance in the presence of increasing measurement uncertainty. The error rate is seen to begin to increase when the measurement

uncertainty becomes greater than 0.25 metres. This value can be compared to the uncertainty in measured distance measured from the experimental data in Section 5.3 of 0.09 metres. This shows that there is a large margin in the variation of quality of sensing which can be used for successful localization. This is useful as in practice, measurements may be less precise than in experiment due to a less controlled environment.

Figure 5.6(d) shows that the algorithm's performance in the presence of increasing motion uncertainty. The values of uncertainty are used to determine the noise added to the command robot motion of around 2.5 metres. The localization error rate begins to increase when the odometry uncertainty becomes larger than 1.25 metres. For larger values of odometry uncertainty, the upper quartile error rate increases, while the median error rate remains close to zero. The relative motion uncertainty can be compared to the values from the literature shown in Table 1.3. A range of values of motion uncertainty are seen in the literature, from as low as 1% of distance travelled to 35% of distance travelled. Here, the acoustic echo localization error rate begins to increase when the motion uncertainty standard deviation becomes greater than 50% of the distance travelled, while the odometry localization error rate is greater than 80% at this value of uncertainty. Therefore, while the error measured in the literature can not be compared directly to the standard deviation of uncertainty used in this experiment, this result is an indication that the use of acoustic echoes for localization is effective at all values of motion uncertainty seen in the literature.

While the sensing approaches taken here and in Chapter 4, acoustic vibration measurement and acoustic echo measurement, may not both be possible in a given pipe environment, their performance can be compared. In both cases, the results measure the performance of localization along the length of a single pipe using the *error rate*. The results can be compared by finding values of uncertainty which give similar error rates for the odometry-based methods in Figure 4.5 and in Figure 5.6, and then comparing the error rates for the estimates made using the acoustic sensing.

In Figure 5.6, at the default value of odometry uncertainty of 0.5 metres, the error rate of the odometry-only estimate is similar to the odometry-only estimates in Figure 4.5. In Figure 4.5, the median error rate found using the acoustic vibration measurements can

be as low as zero, while the upper quartile and range of error rates are around 0.4 and 0.8 respectively. In Figure 5.6, at the comparable value of odometry uncertainty, the median error rate is zero, and the upper quartile and range of the error rate is below 0.1 for a large range of values of other uncertainty variables. This shows that the acoustic echo approach proposed in this Chapter gives a more reliable improvement to accuracy than the acoustic vibration measurement proposed in Chapter 4.

Overall, the acoustic echo localization approach is seen to be robust to false positive measurements, measurement uncertainty, and motion uncertainty. In the case of measurement and motion uncertainty, the proposed approach is robust to larger magnitudes than those seen experimentally or reported in the literature, which gives some confidence that the approach will work when applied in practice.

5.6 Conclusions

The aim of this chapter was to develop a novel approach of using acoustic echoes well suited for robot localization and mapping in the pipe environment. The acoustic sensing is able to observe features at a larger distance than is possible for conventional sensors, which increases the perspective of the robot. A typical estimation algorithm, pose-graph optimization, has been augmented with additional processing which is shown to be necessary for realistic measurements. This augmented *Advanced* algorithm has been shown to be able to incorporate information from this acoustic echo sensing, and has been shown to benefit from additional information which can be found from the measurements.

Detecting acoustic echoes from features in the environment has been shown to be well suited for application in pipes as the sound propagates sufficiently long distances through a network, as demonstrated experimentally. The novel *Advanced* localization algorithm has been shown to be necessary in more realistic environments, and has been shown to give a low error rate over the range of conditions expected from the literature. The approach, from sensing to estimation, has been shown to be robust to various sources of uncertainty, and the expected accuracy and reliability of the output estimate has been found experimentally for

a wide range of parameters.

Further work could be done to evaluate the approach on a larger set of data, with a larger range of pipe characteristics and configurations. Further developments are being made in the methods for processing the acoustic echo measurements. Improvements to the signal to noise ratio both from hardware and from the algorithmic approach are being made, allowing more reliable detection of features in the environment. Work is being done to classify or recognise the sources of acoustic echoes by the characteristics of the measured signal, which could be easily and usefully incorporated into the localization estimate described here.

Chapter 6

Back-End: Hybrid Continuous-Discrete Space Localization

6.1 Introduction

As described in Chapter 2, while previous work on robot localization in pipe networks has typically estimated the robot's position in a continuous state space, the environment can be easily described by a discrete state space, where the map is made up of a number of discrete connected places. Chapter 2 also described the idea of a hybrid continuous-discrete state space, where the environment is represented as a set of discrete connected places which each have some continuous space within them. Chapter 3 concluded that these approaches offer improvements over the typical *continuous* approach.

This chapter describes the development of back-end localization algorithms for the pipe network environment, developing from state of the art algorithms for this environment, and using novel approaches to provide substantial improvements to performance in terms of efficiency and robustness. Additionally, the performance of these algorithms is measured

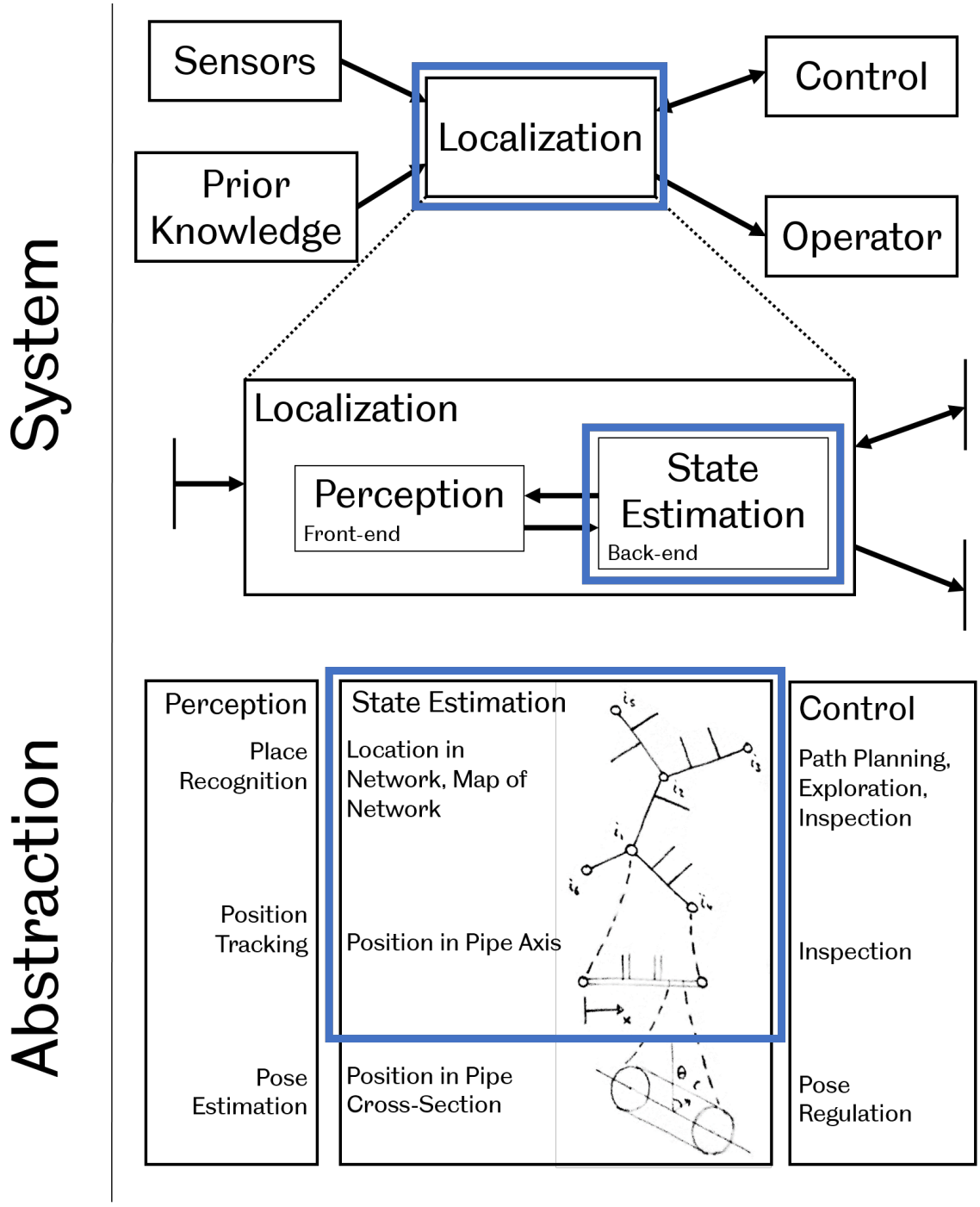

Figure 6.1: *An illustration of the scope of this chapter, with the blue boxes indicating the scope in the context of the rest of the localization system.*

more thoroughly than in related literature.

First, the *discrete* state space approach will be developed, based on work published in Rob Worley and Sean Anderson. "Topological robot localization in a pipe network". In: *UKRAS20 Conference: "Robots into the real world" Proceedings* (2020), pp. 59–60 and Rob Worley and Sean Anderson. "Topological robot localization in a large-scale pipe network". In: *Towards Autonomous Robotic Systems, TAROS 2020* 1 (2020). DOI: 10.31256/zw1wq5m, and its limitations will be described. Secondly, the *hybrid continuous-discrete* state space will be developed, based on work published in Rob Worley and Sean Anderson. "Robust Efficient Localization of Robots in Pipe Networks using a Particle Filter for Hybrid Metric-Topological Space". In: *2021 European Conference on Mobile Robots (ECMR)* (2021) and Rob Worley and Sean Anderson. "Robot Localization in a Pipe Network using a Particle Filter with Error Detection and Recovery in a Hybrid Metric-Topological Space". In: *2021 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)* (2021), pp. 1–8. DOI: 10.1109/mfi52462.2021.9591168, which shows substantial novel improvement over existing comparable approaches.

Figure 6.1 illustrates the focus of this chapter on high and medium level localization, and that the focus is not on the details of how the relevant measurements are acquired.

6.2 Problem Definition

6.2.1 Problem Definition in General

The robot's environment consists of *nodes* (junctions and corners) connected by *links* (pipes). An example of this environment is illustrated in Figure 6.2.

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 relationship between state $\mathbf{x}_t$ and measurements $\mathbf{u}_t$

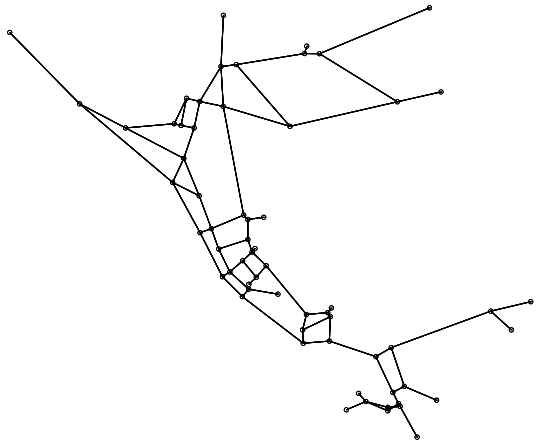

Figure 6.2: *An example network of pipes used in this work.*

and $\mathbf{z}_t$ is defined by

$$\mathbf{x}_t = f_u(\mathbf{x}_{t-1}, \mathbf{u}_t) \tag{6.1}$$

$$\mathbf{z}_t = f_z(\mathbf{x}_t) \tag{6.2}$$

which show that $\mathbf{u}_t$ 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 $\mathbf{x}_t$.

6.2.2 Problem Definition in Detail

6.2.2.1 State Definition

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 continuous-discrete* coordinate system as the triple

$$\mathbf{x}_t = \Big(i_t, x_t, d_t\Big) \tag{6.3}$$

The time steps t are not necessarily at regular intervals. The localization estimate is instead made either after some arbitrary time has elapsed, or when new information is available, which is the case when the robot has executed a turn or when it arrives at a node. $i_t \in \mathcal{I}$ 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 $d_t = d_t^l \in \left\{-1, 1\right\}$, or node, where $d_t = d_t^d d_t^n$, with $d_t^n \in \left\{1, 2, \ldots, D^n\right\}$ where there are D^n adjacent links at node n, and $d_t^d \in \left\{-1, 1\right\}$ so the sign of the index depends on whether the robot is arriving or leaving from a direction.

The robot's pose $\mathbf{x}_\tau$ at time step[1] τ is defined in a *discrete* coordinate system as

$$\mathbf{x}_\tau = \left(i_\tau, d_\tau\right) \tag{6.4}$$

Here, the state is composed of two components. The first component is the robot's discrete position, which is the junction index $i_\tau \in \mathcal{N}$ at time index τ. The second is the robot's discrete direction d_τ which is the index of the link which is has arrived from.

In both cases, the state is therefore effectively defined by a *map*, $\mathcal{M}$, of the environment. This map could be defined specifically in a number of ways, but a minimal representation would describe the neighbouring nodes and links for each node $\mathcal{N}$, and describe the position in continuous space of each node $\mathcal{N}$ and the start and end position of each link $\mathcal{L}$. Because the map $\mathcal{M}$ defines the state space, it does not need to be explicitly referenced in the equations for state estimation. It is assumed here that the robot has prior knowledge of the map $\mathcal{M}$.

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

$$\mathbf{x}_t^g = \left[x_t^g, y_t^g, \theta_t^g\right]^T \tag{6.5}$$

using the map of the poses of each link and node. This could also be extended to a three-dimensional representation of pose.

[1] A different symbol for time index is used, so t refers to a short arbitrary time index, while τ refers to an arbitrary time when the robot arrives at a node.

6.2.2.2 Motion Definition

When the robot's state is defined in the *continuous-discrete* state space according to equation 6.3, the robot's linear position changes according to Δx_t, given by

$$x_t^l = x_{t-1}^l + \Delta x_t d_t + v_t \tag{6.6}$$

Δx_t is usually equal to a fixed command input, except when the robot stops as it arrives at a node. In this work, two models for the uncertain linear robot motion are used. The first uses uncertainty in motion drawn from a Gaussian distribution, where v_t is given by

$$v_t \sim \mathcal{N}(\mu = 0, \sigma = \sigma_x \Delta x_t) \tag{6.7}$$

In the second model, v_t is given by

$$v_t = k_v v_{t-1} + (1 - k_v)\tilde{v}_t \tag{6.8}$$

where $\tilde{v}_t$ is uniformly distributed noise in the range $[-u_x, u_x]$. This aims to model a more challenging case.

If the robot's state is defined in a discrete state space according to 6.4, the measured distance travelled is integrated between junctions, giving a measurement m_t.

When using a *continuous-discrete* state, 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

$$\theta_t^n = \theta_{t-1}^n + \Delta\theta_t + w_t \tag{6.9}$$

with Gaussian uncertainty w_t given by

$$w_t \sim \mathcal{N}(\mu = 0, \sigma = \sigma_\theta \Delta\theta_t) \tag{6.10}$$

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.

When using a *discrete* state, this change in angle can be used to define a discrete *action*, a_τ, which the robot has taken. If the uncertainty given by equation 6.10 is sufficiently large in an instance, then the robot will be moving in a different discrete direction to what would be predicted. Effectively the robot can incorrectly execute a turn action [20].

6.2.2.3 Measurement Definition

At the state $\mathbf{x}_t$ the robot can make some observations of its surroundings. Here, it is assumed that the robot can only make these simple observations.

A discrete measurement $z_t \in \left\{L, N\right\}$ is made, corresponding to the detection of a link or node respectively, with the following probabilities

$$p(z_t = L | i_t \in \mathcal{L}) = 1 - \beta_p \tag{6.11a}$$

$$p(z_t = L | i_t \in \mathcal{N}) = \beta_n \tag{6.11b}$$

$$p(z_t = N | i_t \in \mathcal{L}) = \beta_p \tag{6.11c}$$

$$p(z_t = N | i_t \in \mathcal{N}) = 1 - \beta_n \tag{6.11d}$$

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

When the *discrete* state space approach is taken, this discrete measurement is redundant, as it is instead always $z_t = N$. If a false negative detection of a node is made, then the robot will continue moving in a random unknown direction, without updating the state estimate [55]. However, when using the *discrete* state space approach, it is not possible to incorporate the possibility that the robot makes a false positive detection of a node, as the discrete state space only includes nodes, not the links between them.

Here, when using the *discrete* approach, a measurement o_t, of the number of exits, from a junction is made, which could be done using a camera, sonar, or a number of other sensing modes, as described in Chapter 2.

6.3 Discrete State Space Localization

This section describes a solution for discrete state space localization, and briefly discusses the limitations of the approach.

6.3.1 State Definition

When the state $\mathbf{x}_\tau$ is defined in the *discrete* state space, given by equation 6.4, $\mathbf{u}_\tau$ is described by

$$\mathbf{u}_\tau = \Big\{ a_\tau, m_\tau \Big\} \tag{6.12}$$

and $\mathbf{z}_\tau$ is described by

$$\mathbf{z}_\tau = o_\tau \tag{6.13}$$

The robot's belief in the state is represented as a vector

$$\boldsymbol{b}(\mathbf{x}_\tau) = \boldsymbol{p}(\mathbf{x}_\tau | \mathbf{x}_0, \mathbf{u}_{1:\tau}, \mathbf{z}_{1:\tau}) \tag{6.14}$$

over all possible values of the state, where each value is the estimated likelihood of being in that particular state.

6.3.2 Localization Model

Models for motion and measurements are needed in order to estimate the state $\mathbf{x}_\tau$. Incorporating the discrete variables a_τ and o_τ can be done using discrete probability distributions. To incorporate the continuous measurement of distance travelled, m_τ, the probability of making a given continuous measurement must be found. For motion between two given discrete places, i_τ and $i_{\tau-1}$, there are a number of possible transition lengths, given by the number of possible paths between the two places. The probability distribution over a range of possible measurements is given by a sum of Gaussian distributions. A probability estimate is found

as

$$\tilde{p}(i_\tau^j | i_{\tau-1}^k, m_\tau) = \sum_{v_{i^k}^{ij} \in \Upsilon_{i^k}^{ij}} \frac{1}{\sigma_{v_{i^k}^{ij}} \sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{m_\tau - m_{v_{i^k}^{ij}}}{\sigma_{v_{i^k}^{ij}}}\right)^2} \tag{6.15}$$

where $\tilde{p}(i_\tau^j | i_{\tau-1}^k, m_\tau)$ is the relative probability estimate for a transition from $i_{\tau-1}^j$ to i_τ^j given measurement m_τ, $\Upsilon_{i^k}^{ij}$ is the set of paths between $i_{\tau-1}^j$ and i_τ^j, and $\sigma_{v_{i^k}^{ij}}$ and $m_{v_{i^k}^{ij}}$ are the standard deviation and mean of the distance travelled for path $v_{i^k}^{ij}$.

6.3.3 State Estimation

The forward algorithm is used to compute the discrete probability distribution over the possible robot states. The algorithm in takes the form

$$\boldsymbol{b}(x_\tau) = p(\mathbf{z}_\tau | \mathbf{x}_\tau) p(\mathbf{x}_\tau | \mathbf{x}_{\tau-1}, \mathbf{u}_\tau) \boldsymbol{b}(\mathbf{x}_{\tau-1}) = p(o_\tau | \mathbf{x}_\tau) p(\mathbf{x}_\tau | \mathbf{x}_{\tau-1}, m_\tau, a_\tau) \boldsymbol{b}(\mathbf{x}_{\tau-1}) \tag{6.16}$$

These probabilities are implemented as matrix-vector multiplication, which gives the practical implementation of the Bayes filter the algorithm is derived from.

6.3.4 Results

An example of the result of the localization in the discrete space is shown in Figure 6.3. Figure 6.3(b) and (c) compare the estimation results without and with the use of the measured distance m_τ, which is seen to reduce the error rate from 0.75 to 0.2. This improvement is expected as the the measured distance travelled provides more information, and the information is more precise than the information from the observation of the number of exits from a junction.

Further analysis is presented in the related paper Rob Worley and Sean Anderson. "Topological robot localization in a large-scale pipe network". In: *Towards Autonomous Robotic Systems, TAROS 2020* 1 (2020). DOI: 10.31256/zw1wq5m, which is not repeated here for conciseness. It can be concluded that localization in this discrete state space is effective, however, it has some fundamental limitations. The discrete state space definition

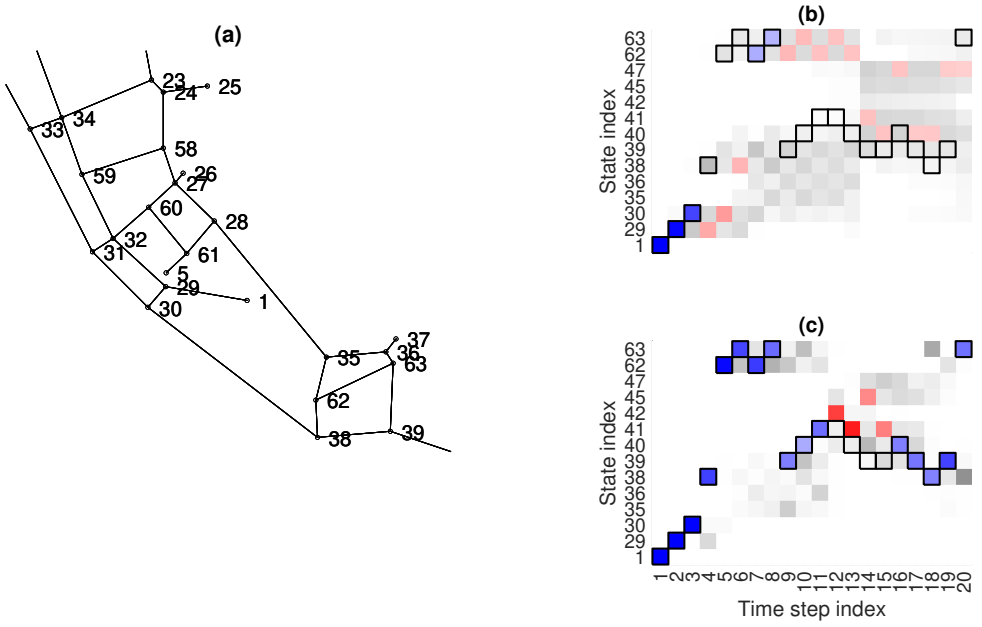

Figure 6.3: *(a) A network map with labelled states. (b), (c) The belief vector over each time step, with and without use of m_t. The darkness corresponds to the belief for each state, with correct estimates in blue, and incorrect in red. Bordered cells show the true state.*

limits the robot behaviour and observations that can be modelled, and localization can only be done at nodes in the network. While the state space is smaller than a *continuous* space in the sense that it is a constrained set of discrete places, it is not possible to simply parameterise an estimate like it is in a continuous state space, so efficiency is not necessarily improved.

A *hybrid continuous-discrete* approach addresses these problems. This expands the restrictive discrete approach presented in this section, which should allow a larger range of robot behaviour and sensing to be incorporated, but which still uses the information of network topology to benefit localization. A particle filter is used in this continuous-discrete approach, which benefits from the reduced size of the state space and provides a compact and flexible representation of the state estimate in the form of a set of particles.

6.4 Hybrid Continuous-Discrete State Space Localization

This section describes a solution to localization in the hybrid continuous-discrete state space approach. Novel approaches to improving performance in this application to robots with high uncertainty in motion and perception are developed.

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 [30]. 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 [171, 172]. The use of map topology facilitates global localization which improves robustness of the particle filter.

6.4.1 Particle Filter in General

Where other Bayes filters describe the posterior distribution parametrically, such as the Kalman filter's Gaussian distribution with a mean and covariance, the *particle filter* describes the posterior distribution as a set of H hypotheses, or *particles* [30]. The likelihood of the

state $\mathbf{x}_t^h$ of particle h being included in the particle set $\mathcal{X}_t$ is proportional to the posterior, or *belief*, $b(\mathbf{x}_t)$, given by

$$\mathbf{x}_t^h \sim b(\mathbf{x}_t) = p(\mathbf{x}_t|\mathbf{u}_{1:t}, \mathbf{z}_{1:t}) \tag{6.17}$$

Like other Bayes filter approaches, the distribution $b(\mathbf{x}_t)$ is recursively estimated using

$$b(\mathbf{x}_t) \propto p(\mathbf{z}_t|\tilde{\mathbf{x}}_t) \int p(\tilde{\mathbf{x}}_t|\mathbf{u}_t, \mathbf{x}_{t-1}) b(\mathbf{x}_{t-1}) d\mathbf{x}_{t-1} \tag{6.18}$$

Implementing this recursive estimation in the particle filter, the state is *predicted* by the set $\tilde{\mathcal{X}}_t$, which is sampled from the *proposal distribution* as

$$\tilde{\mathbf{x}}_t^h \sim p(\mathbf{x}_t|\mathbf{u}_t, \mathbf{x}_{t-1}^h) \tag{6.19}$$

The particles are then *weighted* according to

$$\alpha_t^h \propto p(\mathbf{z}_t|\tilde{\mathbf{x}}_t^h) \tag{6.20}$$

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

$$p(\mathbf{x}_t^h \in \mathcal{X}_t) \propto \alpha_t^h \tag{6.21}$$

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.

Alternatively, the process of *mixture distribution sampling* can be used, where 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

$$\tilde{\mathbf{x}}_t^h \sim p(\mathbf{z}_t|\mathbf{x}_t) \tag{6.22}$$

Subsequently, these particles are weighted according to

$$\alpha_t^h \propto p(\tilde{\mathbf{x}}_t^h|\mathcal{X}_{t-1}, \mathbf{u}_t) \tag{6.23}$$

Further alternative formulations can be used to construct the particle filter. One example is to use an improved *proposal distribution* $p(\mathbf{x}_t|\mathbf{z}_t, \mathbf{u}_t, \mathbf{x}_{t-1}^h)$ which incorporates the

measurement $\mathbf{z}_t$ into the process of sampling particles [30]. Therefore, sampled particles are more likely to represent the robot state, which can improve the algorithm's performance at the cost of more complicated maths needed in both the sampling and weighting steps. This approach is not used here.

6.4.2 Particle Filter Definition - Hybrid Metric-Topological Space

When using the state definition in equation 6.3, the variable $\mathbf{z}_t$ can be defined as $\mathbf{z}_t = z_t$, and the variable $\mathbf{u}_t$ can be defined as $\mathbf{u}_t = \Big\{\Delta x_t, \Delta\theta_t\Big\}$.

The particle filter algorithm functions differently over time depending on the robot's behaviour. In each case the particle filter function follows the general form described in the previous section, however there are differences described here for when the robot believes it is in a link or in a node.

When the robot believes it is in a pipe, or link, i.e. it has made an observation $z_t = L$, the robot makes a linear motion and the algorithm functions as follows.

For linear robot motion, state prediction is done by sampling particles using the motion model as in equation 6.19. For each particle h in set $\mathcal{X}_{t-1}$, the predicted state is sampled using

$$\tilde{x}_t^{l,h} = \tilde{x}_{t-1}^{l,h} + \Delta x_t \tilde{d}_{t-1}^{l,h} + \tilde{v}_t \tag{6.24}$$

where $\tilde{v}_t$ is sampled from a normal distribution with a variance equal to $\tilde{\sigma}_x \Delta x_t$, approximating the noise variable v_t. If $\tilde{x}_t^{l,h}$ is greater than x_i^L or less than zero, the particle has passed a node at the corresponding end of the link. With probability $\tilde{\beta}_n$, which is an estimate of the probability of false negative detection of a node β_n, 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 $\tilde{x}_t^{l,h}$ accordingly. With the remaining probability $1 - \tilde{\beta}_n$, the particle moves to the node at this end of the link, again updating $\tilde{d}_t$ accordingly. Each particle is

then weighted according to equation 6.20, given by

$$\alpha_{t,L}^{m} = \begin{cases} 1 - \tilde{\beta}_n, & \text{if } x_t^h \in \mathcal{L} \\ \tilde{\beta}_n, & \text{if } x_t^h \in \mathcal{N} \end{cases} \tag{6.25}$$

When the robot believes it has arrived at a node, i.e. it has made an observation $z_t = N$, the robot stops and the algorithm functions as follows. For normal operation there are two time steps in this case; in the first time step the robot arrives at the node, and in the second time step the robot turns towards the exit from the node.

When $z_t = N$, mixture distribution sampling is used. With the metric-topological state definition, mixture distribution sampling at $i \in \mathcal{N}$ is simple. Particles can be sampled at all nodes $i \in \mathcal{N}$ in all discrete directions d^n. As will be described shortly, the particle weights, which are found using equation 6.20, will only be nonzero for locations where there are particles nearby prior to this sampling. Therefore, particles can be sampled only at nodes $i \in \mathcal{N}$ which are nearby to existing particles.

The weight of each new particle $\tilde{\mathbf{x}}_t^h$ is given by 6.20. Using the set of particles as a distribution, this weight can be found by applying the motion model like in normal sampling, and somehow estimating the probability that each of the particles in the set could take the value of the new particle sampled from the measurement model. In general, this is described by

$$\alpha_t^h = \sum_k p(\tilde{\mathbf{x}}_t^h | \mathbf{u}_t, \mathbf{x}_{t-1}^k) = \sum_k \alpha_t^{h,k} \tag{6.26}$$

Considering first the case where the linear motion model 6.6 is applicable, where Δx_t is nonzero. A Gaussian kernel is applied to each particle k, therefore the desired probability for each particle k is given by

$$\alpha_t^{h,k} = p(\tilde{\mathbf{x}}_t^k = x_t^h) = \frac{1}{\sigma_g \sqrt{2\pi}} e^{-(x^l - \tilde{\mathbf{x}}_t^k)^2 / 2\sigma_g^2} \tag{6.27}$$

Considering second the case where the angular motion model 6.9 is applicable, where $\Delta\theta_t$ is nonzero. For each particle k that has moved forwards to the node at the previous time step $t-1$, the continuous angle predicted by the input $\Delta\theta_t$ can be found using $\tilde{\theta}_t^n = \tilde{\theta}_{t-1}^n + \Delta\theta_t$.

The angular difference $\delta\theta_{t,k}^{n,h}$ between $\tilde{\theta}_t^n$ and the angle of the newly sampled particle $\Theta_{\hat{d}_t}^h$ can be used to compute the weight as

$$\alpha_t^{h,k} = e^{-\delta\theta_{t,k}^{n,h}{}^2/2\tilde{\sigma}_\theta^2} \tag{6.28}$$

The weights $\alpha_{t,L}^h$, $\alpha_{t,N}^h$, and $\alpha_{t,\theta}^h$ are used in particle resampling as in equation 6.21. To improve on *sequential importance* resampling, a combination of *stratified* and *low variance* resampling is used in this algorithm [30]. The use of the metric-topological state space results in particles being naturally clustered in different discrete places. *Stratified* resampling can then be used, where each *strata* corresponds to each discrete place. This is combined with *low variance resampling*, which systematically selects particles, rather than randomly selecting particles, ensuring all particles with an above average weight are sampled at least once [30]. These resampling methods 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.

6.5 Improvements to the Algorithm: 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 calculate either the difference between the distribution before and after weighting, ($p(\mathbf{x}_t|\mathbf{u}_t, \mathbf{x}_{t-1})$ and $p(\mathbf{x}_t|\mathbf{u}_t, \mathbf{z}_t, \mathbf{x}_{t-1})$) [83], or the difference between the probability of measurements ($p(\mathbf{z}_t|\mathbf{x}_t)$) and the normal measurement probability [30]. 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 $\mathcal{A}_t = \left\{\alpha_t^1, \alpha_t^2, \ldots, \alpha_t^h, \ldots, \alpha_t^H, \right\}$, where most, if not all, elements will be equal or 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 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 a sequence of mean weights $\bar{\alpha}_\tau$ over some set of recent times $\mathcal{T} = \left\{t, t-1, \ldots, t-\tau_\alpha\right\}$, 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 $\bar{\alpha}_\mathcal{T}$ 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 $\mathcal{A}_\mathcal{T}$ could be used, it wasn't seen to be particularly informative here.

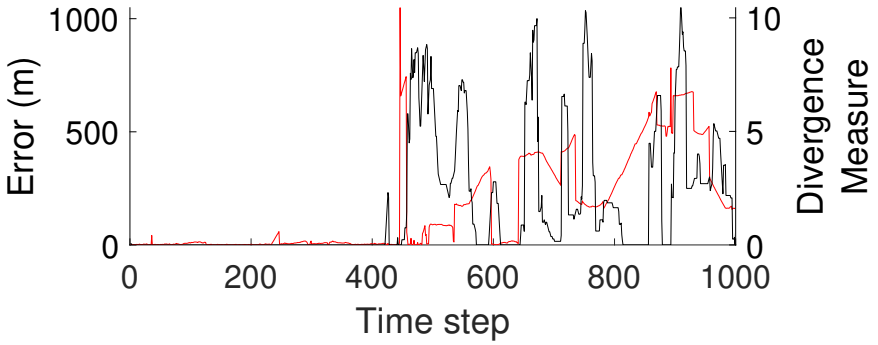

Figure 6.4: *An example of the performance of the algorithm for detecting mislocalization. The error in the particle filter estimate is shown in red, and the divergence measure, used as an indicator of mislocalization, is shown in black.*

To measure the similarity of $\bar{\alpha}_{\mathcal{T}}$ 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 $\bar{\alpha}_{\mathcal{T}}$, $p(\bar{\alpha}_{\mathcal{T}})$, and of the data, $p(\bar{\alpha}^g)$ and $p(\bar{\alpha}^b)$. $D^g_{\mathcal{T}}$ is defined by

$$D^g_{\mathcal{T}} = D_{kl}\left(p(\bar{\alpha}_{\mathcal{T}})||p(\bar{\alpha}^g)\right) = \sum p(\bar{\alpha}_{\mathcal{T}}) \log\left(\frac{p(\bar{\alpha}_{\mathcal{T}})}{p(\bar{\alpha}^g)}\right) \tag{6.29}$$

and $D^b_{\mathcal{T}}$ 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 $p(\bar{\alpha}_{\mathcal{T}})$ is substantially different for $z_t = L$, and $z_t = N$. Therefore, a different probability distribution and corresponding divergence is used for each case (e.g. $p(\bar{\alpha}^L_{\mathcal{T}})$, $D^{gL}_{\mathcal{T}}$). 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 four variable distributions, $p(\bar{\alpha}^L_{\mathcal{T}_s})$, $p(\bar{\alpha}^L_{\mathcal{T}_l})$, $p(\bar{\alpha}^N_{\mathcal{T}_s})$, and $p(\bar{\alpha}^N_{\mathcal{T}_l})$, and four learned distributions, $p(\bar{\alpha}^{gL})$, $p(\bar{\alpha}^{bL})$, $p(\bar{\alpha}^{gN})$, and $p(\bar{\alpha}^{bN})$. The divergence is found between each variable distribution and each learned distribution corresponding to the same value of z_t. Accordingly, eight divergences are calculated, used as follows to create a diver-

gence measure for each value of z_t.

$$D_t^L = \eta^L \left(D_{\mathcal{T}_s}^{gL} - D_{\mathcal{T}_s}^{bL} + D_{\mathcal{T}_l}^{gL} - D_{\mathcal{T}_l}^{bL} \right) \tag{6.30a}$$

$$D_t^N = \eta^N \left(D_{\mathcal{T}_s}^{gN} - D_{\mathcal{T}_s}^{bN} + D_{\mathcal{T}_l}^{gN} - D_{\mathcal{T}_l}^{bN} \right) \tag{6.30b}$$

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 $\eta^N = D_{kl}\left(p(\bar{\alpha}^{gN})||p(\bar{\alpha}^{bN})\right)^{-1}$.

An overall divergence measure is given by

$$D_t = D_t^L + D_t^N \tag{6.31}$$

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 6.4.

6.6 Improvements to the Algorithm: 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.

6.6.1 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_h is given by

$$K_h = \begin{cases} \left(1 + \sqrt{D_t}\right) \frac{H_0}{H_{t-1}}^2, & \text{if } D_t > 0 \\ 1, & \text{otherwise} \end{cases} \tag{6.32}$$

Where D_t is the divergence measure described in section 6.5, H_{t-1} is the number of particles at time $t-1$, and H_0 is the target low number of particles. The number of particles sampled at time t is given by

$$H_t = \begin{cases} K_h H_{t-1}, & \text{if } D_t > 0 \\ H_{t-1} + K_{h0}(H_0 - H_{t-1}), & \text{otherwise} \end{cases} \tag{6.33}$$

where K_{h0} 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 6.32 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.

6.6.2 Multi-Hypothesis Filter for Metric-Topological Space

The typical method of using *mixture distribution sampling* over the whole state space for relocalization [30] is impossible due to the limited information in measurements. Instead, an algorithm is used here which estimates the pose of the robot in parallel to the particle filter described in section 6.4.2, and uses this estimate to find the parts of the state space where *mixture distribution sampling* would be useful. 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 6.4.2 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 $\mathbf{x}_t^h$ and a weight α_t^h.

For $z_t = L$, the hypotheses are each moved in linear motion similar to equation 6.24, but without added uncertainty. When a hypothesis h 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^h/l, and the weight of hypothesis h 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 6.26 using the new hypothesis h and original hypothesis k, and the two hypotheses are given weights $p^n\alpha_t^h$ and $(1-p^n)\alpha_t^h$ respectively. When the robot turns at a node, hypotheses in nodes are weighted according to a similar expression to equation 6.28.

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, $\alpha_t^h < \alpha^0$, 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 H_D (typically a value of 10) particles are sampled at each hypothesis h, each with probability equal to $(\alpha_t^h)^2$ when $z_t = L$ and equal to $(\alpha_t^h)^{1/2}$ when $z_t = N$.

Feedback from the particle filter to the multi-hypothesis filter 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 H_t^i particles, these modes are measured as discrete indices i where H_t^i is greater than $\mu\left(H_t^{i\in\mathcal{I}}\right) + \sigma\left(H_t^{i\in\mathcal{I}}\right)$, which is the mean number of particles with each index plus the standard deviation of number of particles with each index.

Overall, 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.

6.7 Experiments and Discussion: Comparison Between State Definitions

This section contains the experiment and discussion originally published in the paper Worley, R., & Anderson, S., *Robust Efficient Localization of Robots in Pipe Networks using a Particle Filter for Hybrid Metric-Topological Space*, European Conference on Mobile Robots 2021.

Two algorithms are compared in this experiment:

1. Localization in 2D continuous space, as developed in the literature [66].
2. Newly developed localization in hybrid 1D continuous-discrete space, described in section 6.4.2 (*1D*).

Both algorithms use the same prior knowledge of the environment, but incorporate it differently.

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. In simulation,

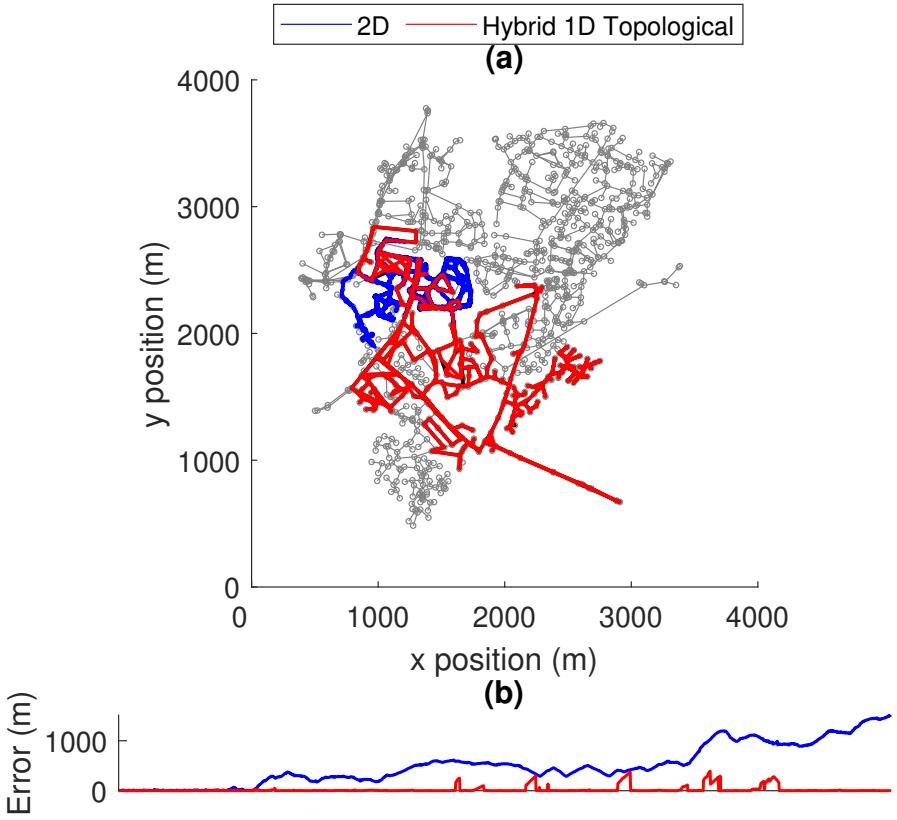

Figure 6.5: *Illustrative results showing the algorithm performance. (a) Two estimates of the robot's trajectory over 10000 steps, corresponding to a total distance of around 44 km. (b) The absolute error of the trajectory estimate over the first 2000 steps of the trajectory.*

uncertainty can be added to the inputs to the localization algorithm, which serves to model both the actual uncertainty in practical inputs, and also the uncertainty in the assumptions made when simulating the robot motion and sensing. Care should be taken when applying results from simulation to practice, but an algorithm shown to work well in simulation with large uncertainty should be likely to work in practice despite uncertainty in modelling.

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

Table 6.1: *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 6.2: *Default parameters for the localization algorithms*

	Parameter	Symbol	Value
2D	Motion model noise	$\tilde{\sigma}_\psi$	$1.2\sigma_x$
	Angular motion model noise	$\tilde{\sigma}_\theta$	$1.2\sigma_\theta$
	Angular motion model noise	$\tilde{\sigma}_{\theta,min}$	0.1 rad
	Link measurement std.	σ_e	Δx m
	Node measurement std.	σ_m	Δx m
1D	Motion model noise	$\tilde{\sigma}_x$	$1.2\sigma_x$
	Node transition std.	σ_n	Δx m
	Angular weight std.	$\tilde{\sigma}_\theta$	$10\sigma_\theta$
	Kernel std.	σ_g	$5\Delta x$ m

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 6.5 (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 in 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. Four sources of uncertainty are tested, and three magnitudes of each source of uncertainty are compared in Figure 6.6.

In each case, the *error rate* is shown, which in this case is defined as the proportion of time for which the estimated robot position is in an incorrect discrete location, This measures the ability of the algorithm to estimate the position of a robot to a single discrete location, which is useful for navigation and approximate localization of faults in the pipe network, as described in Chapter 1. This is estimated by comparing the estimate error to a threshold of 25 metres, which effectively computes the accuracy to the precision of a single discrete location, while neglecting error in the estimate of continuous position within the discrete location, which is not the aim of this experiment.

A *violin* plot is used 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 6.2 are shown in Table 6.1. The default localization parameters are given in Table 6.2, where the symbols relating to the *2D* algorithm are explained in the relevant literature [66]

The four sources of uncertainty are:

1. Gaussian motion noise using the motion model in equation 6.6, with variation in σ_x.

2. Angular measurement noise using the measurement model in equation 6.9, with varia-

tion in σ_θ.

3. Integrated uniform motion noise using the motion model described by equation 6.8, 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.

The values of linear uncertainty used aim to cover the range of linear uncertainty values found in the relevant literature, described in Table 1.3, as well as higher values which are useful in assessing the suitability of the localization algorithm to more limited robots than have previously been developed.

From Figure 6.6, 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.

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* algorithm is robust.

While the error from the *1D* algorithm is lower on average, both algorithms show poor performance at the highest tested magnitudes of either type of motion noise, indicating that good measurement of linear motion is important.

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

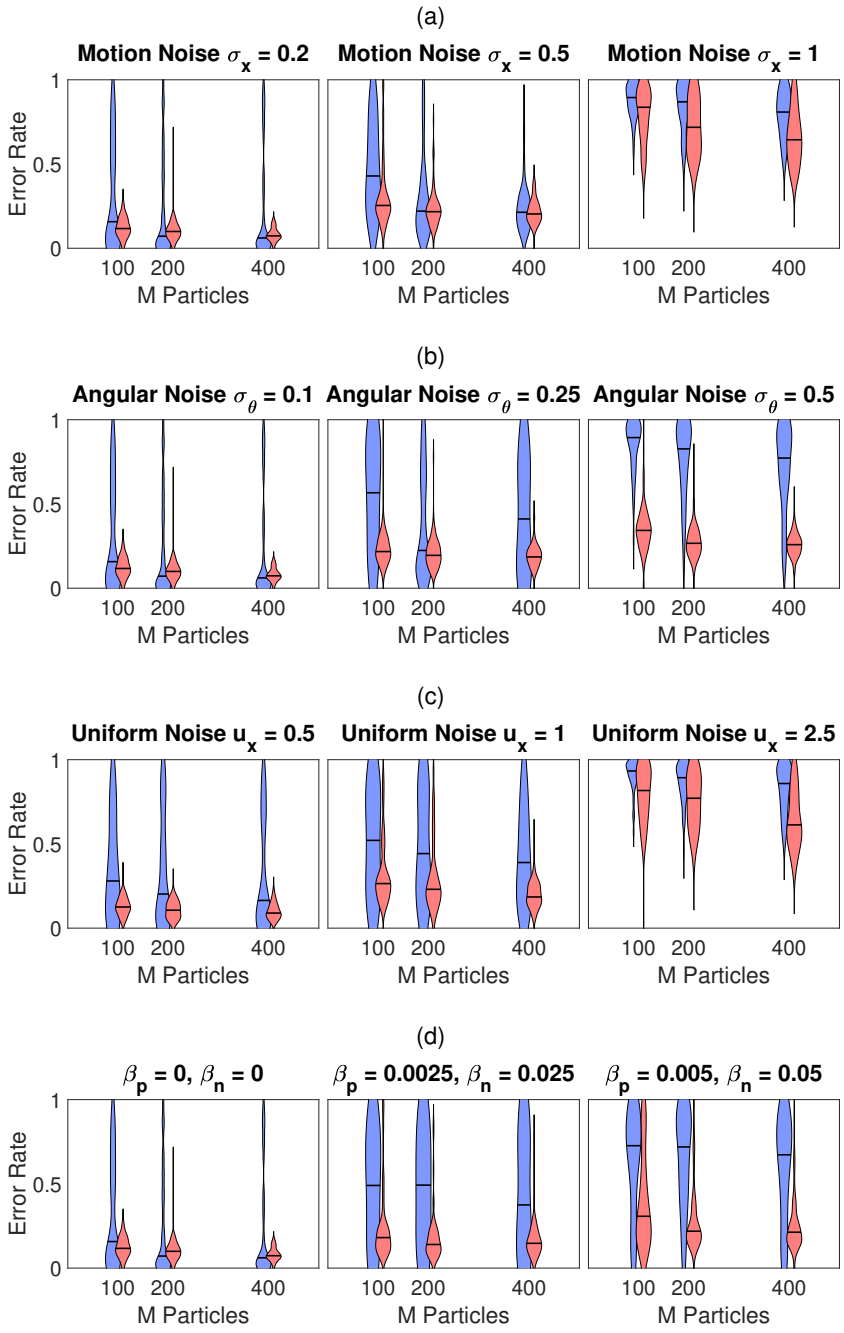

Figure 6.6: *For two algorithms (red: 1D, blue: 2D), the error rate for variation in:* ***(a)*** *Gaussian linear noise magnitude.* ***(b)*** *angular Gaussian noise magnitude.* ***(c)*** *integrated uniform noise in linear motion.* ***(d)*** *measurement error rate.*

Table 6.3: *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

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.

6.8 Experiments and Discussion: Improvements to the Particle Filter Algorithm

This section contains the experimental results and discussion originally published in the paper Worley, R., & Anderson, S., *Robot Localization in a Pipe Network using a Particle Filter with Error Detection and Recovery in a Hybrid Metric-Topological Space*, IEEE MFI 2021 - International Conference on Multisensor Fusion and Integration.

Experiments are done in simulation to evaluate the performance of two algorithms: the *1D* algorithm as described in section 6.4.2, and the *1D+* algorithm which uses the improvements described in sections 6.5 and 6.6.

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*, as defined in section 6.7, and the *failure rate*, which is the proportion of trajectories where the error at the last 100 steps is over a threshold 25 metres, an estimation of when the algorithm has failed and the robot could be considered *lost*. As described in section 6.7, these measures show the effectiveness of the algorithm in localizing a robot to the precision needed for navigation and approximate localization of faults in the pipe network. The computational cost of the algorithms is measured by the *number of particles* used by each algorithm.

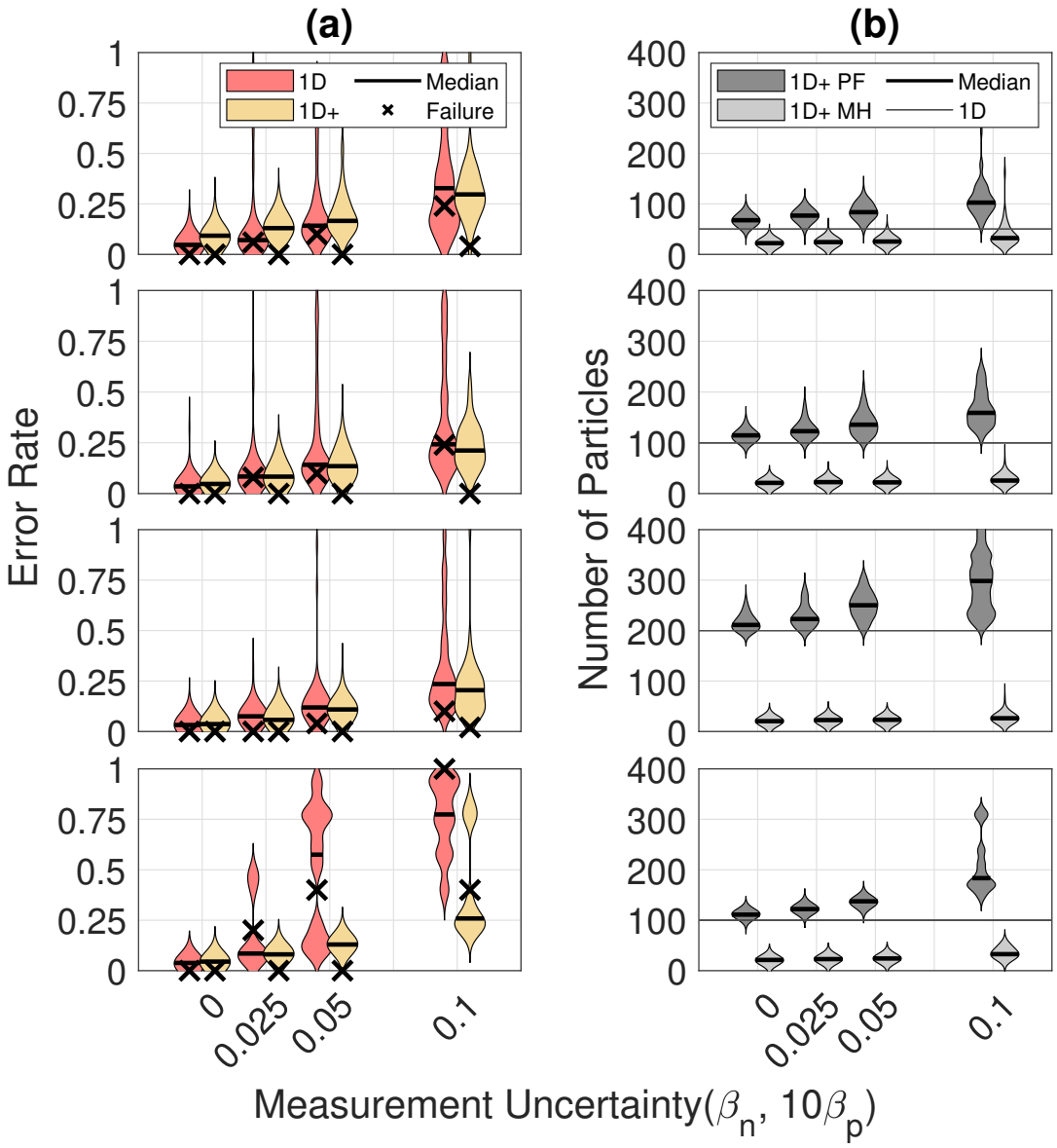

Figure 6.7: ***(a)****: Error rate for the 1D and 1D+ algorithms.* ***(b)****: Number of particles in the 1D+ particle filter and multi-hypothesis filter, and the 1D filter.* ***Top to Bottom****: 50 particles, 100 particles, and 200 particles for 1000 steps, and 50 particles for 10000 steps.*

Table 6.4: *Default parameters for the localization algorithms*

	Parameter	Symbol	Value
1D	Motion model noise	$\tilde{\sigma}_x$	$1.2\sigma_x$
	Node transition std.	σ_n	Δx m
	Angular weight std.	$\tilde{\sigma}_\theta$	$10\sigma_\theta$
	Kernel std.	σ_g	$5\Delta x$ m
1D+	Number of New Particles	H_D	10
	Number of Particles Decay Constant	K_{h0}	0.05
	Divergence Short Time Constant	$\tau_{\alpha s}$	20
	Divergence Long Time Constant	$\tau_{\alpha l}$	100
	Hypothesis Weight Threshold	α^0	0.01

Figure 6.7 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.

In practice, the trajectory length would depend on other aspects of the robot operation. 1000 steps and 10000 steps correspond to a total distance of 4.5 km and 45 km respectively. 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.

Figure 6.7 shows that for increasing measurement uncertainty (β), the error rate increases for both algorithms. For the *1D* algorithm, the chance of failure also increases, 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 median rates of error below 0.25 in most cases.

The computational cost for the *1D+* algorithm is larger than the cost for the *1D* algorithm. The number of particles in both the particle filter and multi-hypothesis filter 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 $H = 50$. As β increases, H 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.

6.9 Conclusions

Overall, this chapter has described and developed ideas for localization in both a discrete state space and a hybrid continuous-discrete state space. The latter approach is concluded to be less limited in this application. This section will summarise the results and discussion around localization in the hybrid continuous-discrete space presented in sections 6.7 and 6.8, and draw some conclusions which will motivate the work in the remainder of this thesis.

Over the two sets of experiments, three algorithms have been compared. Firstly, the *2D* algorithm, which estimates the robot's position in the typical global metric space, which is derived from an algorithm in the existing literature. Secondly, the *1D* algorithm, which estimates the robot's position in the hybrid continuous-discrete space. Thirdly, the *1D+* algorithm, which uses the hybrid continuous-discrete space with novel improvements to robustness.

The two experiments each had different aims. The first experiment looked at the effect of different sources of uncertainty on the localization estimate for the *2D* and *1D* algorithms, measuring the error rate of the algorithms. The second experiment looked at the relative accuracy, reliability, and computational cost of the *1D* and *1D+* algorithms, measuring the error rate and failure rate of the algorithms.

Overall, the first experiment shows that while the *2D* algorithm is seen to be reliable for a more capable robot with low values of each source of error, the *1D* algorithm has better robustness to the various sources of uncertainty measured. The second experiment shows the further improvement to accuracy and reliability from the improvements in the *1D+* algorithm.

The improved algorithm presented in this Chapter is used as a comparison for another novel approach presented in Chapter 7.

Chapter 7

Back-End: Advanced Hybrid Space Localization

This chapter presents the culmination of the work presented previously in this thesis. The back-end estimation approach developed through Chapters 3 and 6 is refined here, and brought together with the front-end sensing and measurement approach described in Chapter 5. The back-end estimation approach takes the *hybrid continuous-discrete* approach developed in Chapter 6, and extends it from a filtering approach to an optimization approach. Efficient optimization is achieved by reducing the sequence of states to a subset where *informative* measurements are made, while the remaining states can be simply estimated given the optimized sequence of *informative* state estimates. This reduction in the sequence of states also allows measurements made over a period of time to be integrated to make an observation of the environment, which facilitates the incorporation of measurements from acoustic echoes as described in Chapter 5, for example.

Figure 7.1 shows the scope of this chapter, which covers both a large and medium scale of estimation. While various front-end measurement types are compared in this chapter, the focus is less on how these measurements could be acquired, and more on how they affect the back-end.

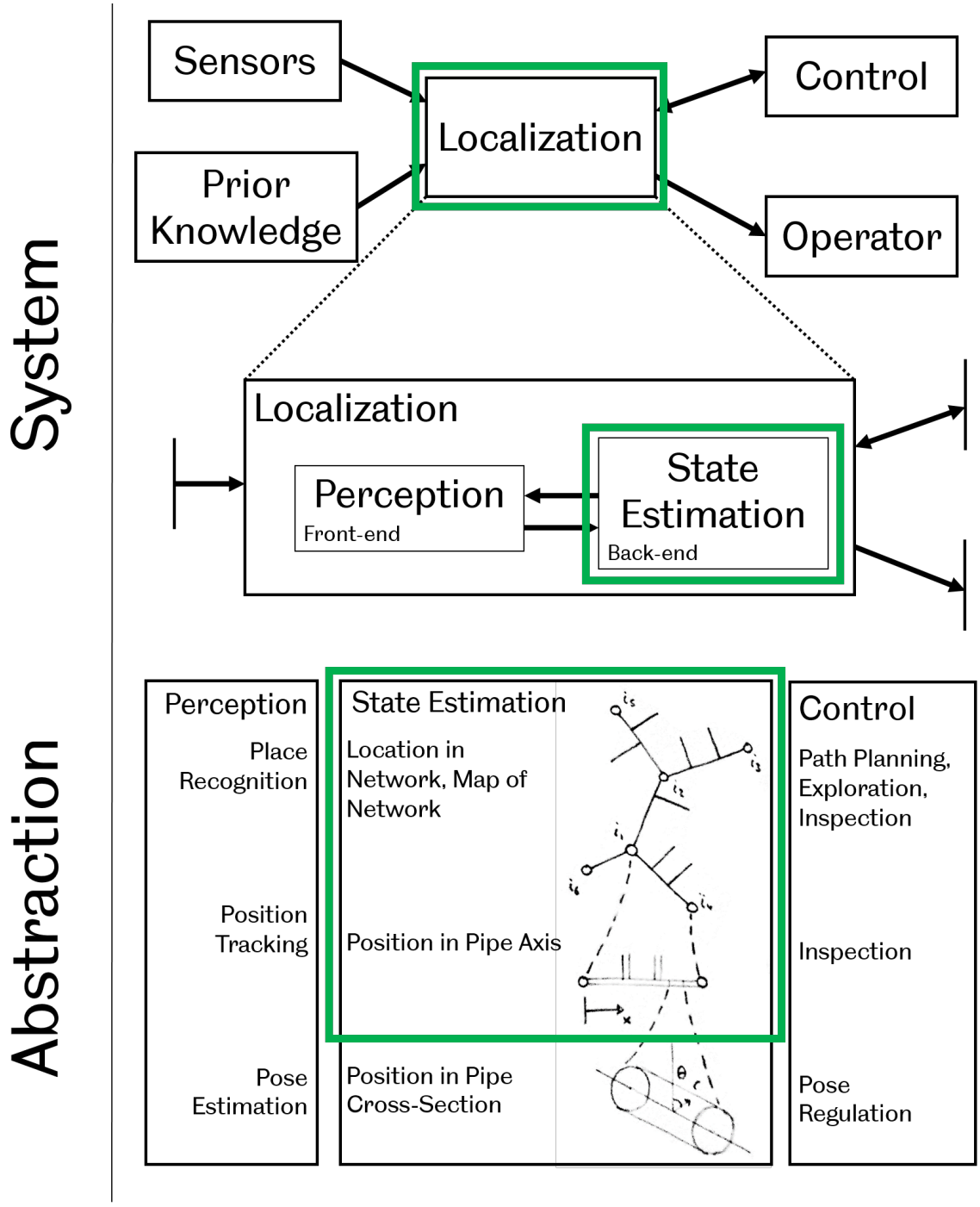

Figure 7.1: *An illustration of the scope of this chapter, with the green boxes indicating the scope in the context of the rest of the localization system.*

This chapter is based on currently unpublished work. Producing a paper based on the approach and results taken here is left for future work.

7.1 Introduction

Throughout this thesis, a number of approaches to the localization back-end estimation have been discussed, developing towards an approach which is best suited for the pipe environment and the constraints on the robot and on sensing that are present.

Chapter 3 described a typical localization approach applied to the pipe environment. There, a robot's trajectory was described in continuous space defined by the two-dimensional position and angle of the robot, and estimated using pose-graph optimization. It was concluded that the approach is effective and flexible as a means of incorporating measurements made by the robot and prior knowledge of the environment. However, the approach is inefficient, as the information from measurements is sparse and much of the computation is spent redundantly estimating poses with only a small amount of information relative to other poses. It was also concluded that incorporating prior knowledge of the environment is useful, but difficult when estimating in the two-dimensional continuous space.

Chapter 6 showed the opposite approach to Chapter 3 and described the estimation of a robot's position in a discrete state space, where each discrete place is a node connecting two or more pipes. Prior knowledge of the environment is given by the pipe network topology, describing the connectivity between discrete places. The state space is therefore reduced in size, aiming to make estimation more efficient. It was concluded that while this approach is effective, it is not flexible, and limits the information that can be incorporated into the estimate. It was also concluded that while the state space is effectively smaller, estimation is not fundamentally more efficient as the robot's pose can't be neatly parameterized as in the approach in the continuous state space.

Chapter 6 then aimed to find a compromise between the continuous and discrete approaches. The robot's pose was estimated in a hybrid continuous-discrete state space,

described by a set of connected places, either pipes or junctions, where each discrete place has some continuous space within it. It was concluded that the approach allows effective, robust, and low-cost localization, but that further improvements to the approach might be found by estimating the robot's trajectory rather than instantaneous position.

Overall, each of the approaches proposed previously are effective, and have both advantageous and disadvantageous characteristics. The aim of this chapter is to develop an approach to localization in a pipe network which takes the best characteristics of each of these approaches. The previously described *hybrid continuous-discrete* approach has been shown to be accurate, robust, and efficient, so will be used as a comparison for further development.

A number of front-end sensing approaches have been developed or discussed throughout this thesis, giving a range of possible measurements which a robot can make relative to its surroundings in the pipe environment. Detection of connections between pipes either from vision, rangefinding, or other methods has been developed extensively in the literature as described in Chapter 2, and these ideas have been applied to the back-end estimation approaches discussed throughout this thesis. Vision-based localization approaches are a means of making both odometry measurements and loop-closing measurements, but are fundamentally limited to observations of only nearby features, as discussed previously. Chapter 4 showed an alternative to vision for loop-closure measurements in feature-sparse environments using a hydrophone sensor to measure an acoustic property of the surrounding pipe. The use of low-frequency acoustic echoes in measuring the distance between the robot and distant features was shown to be very effective in estimating the robot's position in Chapter 5. Overall, these measurement types, as well as others described later in this chapter, offer a range of means of making a localization estimate. This chapter aims to demonstrate and compare the use of these various measurement types.

This chapter will first develop the mathematical formulation of the localization problem, then propose an approach to solving this problem, and finally demonstrate the effectiveness of this approach using a range of types of measurement and prior knowledge of the environment.

7.2 Problem Definition

The problem formulation described here is similar to that used in Chapter 6. The state and measurement definitions are repeated here for clarity.

The pipe network environment is made up of pipes, or *links*, and connections between pipes, or *nodes*. The links are much larger in length than in width and than the size of the nodes, so a close approximation of the network is a set of connected one directional places. The state space of the robot is therefore defined in this hybrid continuous-discrete way, and in this chapter is it assumed that the map of the pipe network, and therefore the state space, is known *a priori*. This map $\mathcal{M}$ would at minimum describe the neighbouring nodes and links for each node $\mathcal{N}$, and describe the position in continuous space of each node $\mathcal{N}$ and the start and end position of each link $\mathcal{L}$. A minimum representation of the map would have low required storage, however there may be some subsequent cost to computing useful information regarding possible paths through the map. Computed information about paths between places in the network is given by variable Υ.

A robot moves around in the network of pipes, modelled as small discrete time *steps*. The aim is to estimate the pose of the robot at each *step*, thereby estimating the robot's trajectory. Specifically, the aim is to estimate the posterior distribution $p(\mathbf{x}_{1:T}|\mathbf{u}_{1:T}, \mathbf{z}_{1:T}, \mathbf{x}_0)$, which is the probability distribution over possible state sequences $\mathbf{x}_{1:T}$ given the sequence of uncertain observations $\mathbf{u}_{1:T}$ and $\mathbf{z}_{1:T}$, and an initial state $\mathbf{x}_0$.

The state sequence is equal to $\mathbf{x}_{0:T} = \left\{\mathbf{x}_0, \mathbf{x}_1, \ldots \mathbf{x}_t, \ldots, \mathbf{x}_{T-1}, \mathbf{x}_T\right\}$. The relationship between instantaneous state $\mathbf{x}_t$ and measurements $\mathbf{u}_t$ and $\mathbf{z}_t$ is defined by

$$\mathbf{x}_t = f_u(\mathbf{x}_{t-1}, \mathbf{u}_t) \tag{7.1}$$

$$\mathbf{z}_t = f_z(\mathbf{x}_t) \tag{7.2}$$

As usual, $\mathbf{u}_t$ 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 $\mathbf{x}_t$. These two measurement variables can be defined more specifically depending on the measurements available.

The pose is defined in this case as the position and direction of the robot within the hybrid continuous-discrete state space, while the small-scale robot pose defined with respect to the pipe cross-section is decoupled from the large-scale estimate and can be estimated independently. The instantaneous state is therefore given by

$$\mathbf{x}_t = \Big(i_t, x_t, d_t\Big) \tag{7.3}$$

where $i_t \in \mathcal{I}$ 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 or node. In a link, $d_t = d_t^l \in \Big\{-1, 1\Big\}$. In a node, $d_t = d_t^d d_t^n$, with $d_t^n \in \Big\{1, 2, \ldots, D^n\Big\}$ 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.

The definition of the time index t is important to consider. When using the *discrete* state space in Chapter 6, the time step was effectively defined as an arbitrary point in time when a robot arrives at a node, so the time step was therefore large. When using the *continuous-discrete* state space in Chapter 6, the time step was defined by a regular interval in time, but where additional time steps were added whenever the robot arrived at a node (or believed it had observed a node). In more general localization, the time step might be defined as a regular interval, small enough that a robot would take several time steps to move though even a small node such as a junction or manhole. The robot's trajectory would then be estimated at a high temporal precision, perhaps as finely as one pose per video frame for example. In this chapter, it is assumed that the interval between time index $t-1$ and t is small, such that it takes many time steps for the robot to move from one node in the pipe network to another. The sequence of time indices is described as

$$\mathcal{T}_t = \Big\{0, 1, 2, \ldots, t, \ldots, T-1, T\Big\} \tag{7.4}$$

7.3 Localization Definition

The approach taken to solving the problem defined in section 7.2 is motivated by the fact that at most instances in time t the robot receives little information from its measurements $\mathbf{u}_t$ and $\mathbf{z}_t$. As was noted in Chapter 3, all odometry measurements in this case simply relate adjacent states in time, and incorporating all of this information into the trajectory estimate is quite costly and does not offer much improvement to the estimate. Therefore, a reduced set of time indices $\tau \in \mathcal{T}_\tau \subset \mathcal{T}_t$ is found which simply gives the set of time indices where an *informative* measurement is made. An *informative* measurement in this case is a measurement which gives more information than other measurements, in some sense. This information could be quantified to make this definition more specific, but here it is defined more simply and intuitively as a measurement which either:

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

For a short trajectory where informative measurements are made at $t = 0$, $t = 9$, $t = 17$, $t = 45$, and $t = 77$, for example, the reduced set of time indices would be given by $\mathcal{T}_\tau = \Big\{0, 9, 17, 45, 77\Big\}$. However, for a clearer description of algorithms used in the rest of this chapter, instead the time indices τ will be redefined and written as a sequence $\mathcal{T}_\tau = \Big\{0, 1, 2, 3, \ldots\Big\}$. An alternative consistent with the previous definition would simply have to use notation like $\tau(k)$, $\tau(k-1)$, when referring to adjacent *informative* points in time, for example.

As for the regular discrete time t, the state at time index τ is defined as

$$\mathbf{x}_\tau = \Big(i_\tau, x_\tau, d_\tau\Big) \tag{7.5}$$

where i_τ and d_τ are discrete variables describing the location and direction of the robot at time τ as defined previously for time t, and x_τ is the continuous position of the robot in a one dimensional pipe or simply equal to zero if i_τ corresponds to a node. The relationship between instantaneous state $\mathbf{x}_\tau$ at the reduced time index τ and measurements $\mathbf{u}_\tau$ and $\mathbf{z}_\tau$ is defined by

$$\mathbf{x}_\tau = f_u(\mathbf{x}_{\tau-1}, \mathbf{u}_\tau) \tag{7.6}$$

$$\mathbf{z}_\tau = f_z(\mathbf{x}_\tau) \tag{7.7}$$

The motion measurement $\mathbf{u}_\tau$ is given by

$$\mathbf{u}_\tau = \Big\{\Delta x_\tau, \Delta\theta_\tau\Big\} \tag{7.8}$$

At the reduced time index τ, it is modelled that the robot turns, making a measurement of change in angle $\Delta\theta_\tau$ by integrating over a period of angular motion, then moves, making a measurement of change in position Δx_τ by integrating over a period of linear motion. This integration can be written as

$$\Delta x_\tau = \sum_{t=\tau-1}^{\tau} \Delta x_t \tag{7.9}$$

The uncertainty in this measurement can be given by

$$\sigma_{x_\tau} = \sqrt{\sum_{t=\tau-1}^{\tau} {\sigma_{x_t}}^2} \tag{7.10}$$

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

$$\sigma_{x_\tau} = \sqrt{T^\tau_{\tau-1}{\sigma_{x_t}}^2} = \sqrt{T^\tau_{\tau-1}}\sigma_{x_t} \tag{7.11}$$

where $T^\tau_{\tau-1}$ is the number of discrete time steps t between reduced time indices $\tau - 1$ and τ. A similar approach can be taken for angular motion, where it assumed that the turning motion is purely angular motion and therefore the motion is *linear* in the angular dimension. The decoupling of the two purely linear motions gives this approach its efficiency. As the robot in this application is constrained to moving in a mostly one-dimensional environment, sequences of purely linear odometry measurements can be simply integrated using a linear motion model without adding the error which would be found when using a linear function to represent the output from a nonlinear model. The cost of estimation can therefore be reduced without adding error to the estimation.

The environment measurements $\mathbf{z}_\tau$ are defined more generally, as the specific measurement depends on what information is available. In the case that no additional information is found from sensors beyond odometry and the detection of nodes, $\mathbf{z}_\tau$ is always simply given by

$$\mathbf{z}_\tau = z_\tau = N \tag{7.12}$$

as, by definition, the time index τ only exists when informative measurements are available, such as the detection of a node. The corresponding measurement of $z_t \in \left\{L, N\right\}$ has some uncertainty, as defined here and in equation 6.11

$$p(z_\tau = L | i_\tau \in \mathcal{L}) = 1 - \beta_p \tag{7.13a}$$

$$p(z_\tau = L | i_\tau \in \mathcal{N}) = \beta_n \tag{7.13b}$$

$$p(z_\tau = N | i_\tau \in \mathcal{L}) = \beta_p \tag{7.13c}$$

$$p(z_\tau = N | i_\tau \in \mathcal{N}) = 1 - \beta_n \tag{7.13d}$$

If information from an observation of a discrete or categorical quality of the robot's surroundings is available, this could be defined as

$$\mathbf{z}_\tau = \left\{z_\tau \in \left\{L, N\right\}, \gamma_\tau, \ldots\right\} \tag{7.14}$$

where z_τ is the detection of either a node or a link, and γ_τ is the colour of the surrounding pipe or junction material, for example. If information from an observation of a metric quality of the robot's surroundings is available, this could be defined as

$$\mathbf{z}_\tau = \left\{ z_\tau \in \left\{ L, N \right\}, \Lambda_\tau, \Theta_\tau \ldots \right\} \tag{7.15}$$

where Λ_τ is an estimate of the length of the surrounding link which could be made using acoustic echoes, and Θ_τ is a set of relative orientations of exits from a junction with respect to the robot, for example. In any case, some of the variables contained in $\mathbf{z}_\tau$ might be empty for some time indices τ, in which case they are not incorporated into the localization estimate.

These informative measurements may require observations that are made and integrated over a sequence of time indices t. Small periods of integration such as the scanning period of a scanning rangefinder sensor are already considered negligible in typical localization definitions. Longer periods of integration such as those for detecting and characterising a junction between pipes based on a point cloud created from visual feature detection are in this case condensed into a single time index τ. Similarly, a sequence of observations $z_t = N$ is condensed into a single time index τ.

Overall this approach condenses the measurements made at frequent time indices t into a smaller set of integrated measurements at time indices τ. The result is a set of useful measurements which shares the simplicity of the approach taken by the *discrete* approach taken in Chapter 6, and has the flexibility of the *continuous* approach taken in Chapter 3 and of the *continuous-discrete* approach in Chapter 6 while also improving on some of the problems found in these approaches. First the trajectory at time indices $\tau \in \mathcal{T}_\tau$ can be estimated, then the sub-trajectories between adjacent indices $\tau \in \mathcal{T}_\tau$ at each index $t \in \mathcal{T}_t$ can subsequently be estimated, where each sub-trajectory should be independent given the estimates at indices $\tau \in \mathcal{T}_\tau$.

7.4 Localization Implementation

Given the approach defined in section 7.3, a practical implementation can be presented. This is presented in two parts, describing first what specific informative measurements can be incorporated into the estimate, and second how the posterior distribution is estimated given these measurements.

The aim here is to find a method of estimating the posterior distribution, which is described as $p(x_{1:T}|x_0, u_{1:T}, z_{1:T})$. Ideally, a complete understanding of the posterior distribution would be found. However, using the hybrid continuous-discrete state space, it is difficult to represent the full distribution without inefficient exhaustive computation. Instead, here a *maximum a posteriori* estimate of the most likely sequence of states is found, which is the mode of the posterior distribution. This optimization problem could be solved by a range of means. Here, the solution is derived from the *Viterbi algorithm* [91], which is known to be an optimal method of estimating a sequence in a discrete-time finite-state Markov process, a description which fits the defined problem here reasonably well.

Compared to the previous approach of estimating the posterior distribution only over the current state at time t, the estimation of the most likely sequence of states has advantages in accuracy and robustness due to the use of information over all times. While instantaneous information might mean that the true state at time t appears unlikely, information from future time steps should, assuming that information in $\mathbf{u}_{1:T}$ and $\mathbf{z}_{1:T}$ is sufficiently informative, mean that the true state at time t appears likely. The estimation of the full sequence of states therefore gives an implicit means of recovering from mislocalization.

7.4.1 Viterbi Algorithm

The algorithm estimates the probability of the most likely sequence to each state $\mathbf{x}_\tau$ in the state space recursively forward through time $\tau \in \mathcal{T}_\tau$, using the information in $\mathbf{u}_\tau$ and $\mathbf{z}_\tau$. The algorithm also records the state at the previous time $\hat{\mathbf{x}}_{\tau-1}$ which most likely proceeds each state at time τ. At the end of this forward recursion, at time T, the state at which

Algorithm 5 The Viterbi algorithm

For each time index forward through time
for each time $\tau \in \mathcal{T}_\tau = \{1, 2, \ldots, T\}$ **do**
  Get the measurements for time τ
  get $\mathbf{u}_\tau$ and $\mathbf{z}_\tau$

  for each state
  for state $\mathbf{x}^j, j \in \{1, 2, \ldots, X\}$ **do**
    Find the probability of the most likely sequence that ends at this state
    $p(\hat{\mathbf{x}}^j_{0:\tau}|\mathbf{x}_0, \mathbf{u}_{1:\tau}, \mathbf{z}_{1:\tau}) = \max_k(p(\mathbf{x}^j_\tau|\mathbf{x}^k_{0:\tau-1}, \mathbf{u}_\tau)p(\mathbf{z}_\tau|\mathbf{x}^j_\tau)), k \in \{1, 2, \ldots, X\}$
    Record the state index at the previous time in this sequence
    $K^j_\tau = \arg\max_k(p(\mathbf{x}^j_\tau|\mathbf{x}^k_{0:\tau-1}, \mathbf{u}_\tau)p(\mathbf{z}_\tau|\mathbf{x}^j_\tau)), k \in \{1, 2, \ldots, X\}$
  end for
end for

Find the trajectory with the highest likelihood
$J_T = \arg\max_j p(\hat{\mathbf{x}}^j_{0:T}|\mathbf{x}_0, \mathbf{u}_{1:T}, \mathbf{z}_{1:T}), j \in \{1, 2, \ldots, X\}$
Get the state that ends this trajectory
$\hat{\mathbf{x}}_T = \mathbf{x}^{J_T}_T$

For each time index backwards through time
for each time $\tau \in \mathcal{T}_\tau = \{T, T-1, \ldots, 2\}$ **do**
  Get the recorded index of the most likely previous state
  $J_{\tau-1} = K^{J_\tau}_\tau$
  Get the state at this index
  $\hat{\mathbf{x}}_{\tau-1} = \mathbf{x}^{J_{\tau-1}}$
end for

the most likely sequence ends, $\hat{\mathbf{x}}_T$, is found, and the most likely sequence is found using the recorded most likely previous states starting with $\hat{\mathbf{x}}_T$ and stepping backwards through time. This function is described by Algorithm 5.

7.4.2 Augmented Viterbi Algorithm

In this implementation, the state is defined as in equation 7.5. The inclusion of this continuous variable means that the state space is infinite in the sense that there are infinitely many possible values of the continuous variables in a given range. Therefore, the Viterbi algorithm requires a small augmentation for use here.

This augmentation is implemented here by computing and maintaining a set of *estimates* or *hypotheses* at each time τ. Each *estimate* has a state $\mathbf{x}_\tau$, a probability describing the likelihood of a sequence from time 0 to time τ ending at the state, and a record of the most likely previous state in this sequence, giving the required information for the function of the Viterbi algorithm. The approach therefore is akin to a multi-hypothesis estimation, as the probability distribution over the state space is computed only for a subset of the whole state space. This approach facilitates the incorporation of the continuous variable x_τ into the estimation in an otherwise discrete state space. The multi-hypothesis approach means that an arbitrary number of estimates could be made in the same link i_τ each with a different continuous position.

To improve efficiency, a threshold for probability is set, below which probabilities are assumed to be zero. If a state at time $\tau - 1$ has a probability of zero, the probability of states at time τ originating from it will also be zero and therefore do not need to be computed. The approach of computing a set of *estimates* facilitates the truncation of probabilities, as sufficiently unlikely estimates can simply be removed from the set. This truncation allows the feasible computation of probabilities in a state space which would otherwise be impractical for this approach.

The states at time $\tau - 1$ and τ could either be nodes or links, which gives four different types of transition between states, each of which are worth describing separately. In each case, the probability of a transition between a pair of states is found given a measured distance travelled Δx_τ with uncertainty given by $\sigma_{x_\tau}\sqrt{T^\tau_{\tau-1}}$, and the measured angle travelled $\Delta\theta_\tau$ with uncertainty given by σ_{θ_τ}. The likelihood of moving from a given state $\mathbf{x}_{\tau-1}$ to a given state $\mathbf{x}_\tau$ can be computed by comparing the odometry measurements to those that would be expected given the known map of the environment. These are described below and also illustrated in Figure 7.2, which shows the computation of probability of transition to a new discrete location given the measured distance travelled, as well as the subsequent new continuous robot position estimated within that discrete location.

- **Node to Node:** With the linear odometry measurement, there is a normally distributed model for distance travelled at time τ. The distance between a pair of nodes

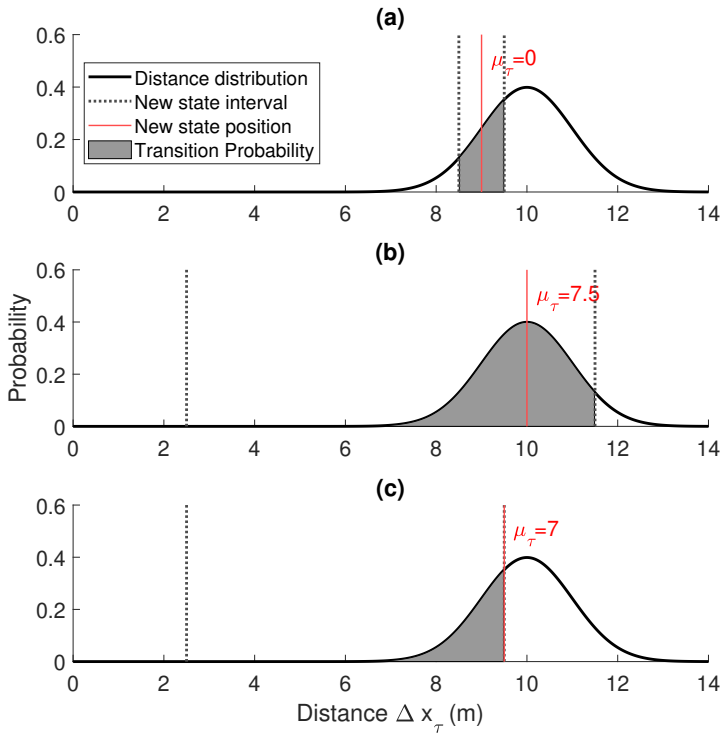

Figure 7.2: *Estimation of the probability of a transition between two states.* ***(a)*** *shows the case for a node state.* ***(b)*** *and* ***(c)*** *show the case for a link state.*

$i_{\tau-1}$ and i_τ can be found, and the likelihood of this transition can be found by comparing the expected distance to the normally distributed measured distance. If the node states are modelled as having infinitesimal size, the probability that a given measured uncertain distance moves the robot from one node state to another is zero. To solve this problem, the node states are considered to have a nonzero size, which allows the calculation of the desired probability by integrating the normally distributed model over the interval in distance corresponding to the node. This is illustrated in Figure 7.2(a). The position x_τ is set to zero. This process is followed for each direction d_τ arrived at. Often there are multiple paths which could be travelled between a given pair of states, so each of these can be considered separately and consolidated later.

- **Node to Link:** The likelihood of moving from a given node $i_{\tau-1}$ to a given link i_τ can be found again by integrating the normally distributed model over the interval in distance travelled, which is given by the distance to the near and far end of the link. This is illustrated in Figure 7.2(b) and (c). As with the transition between a pair of nodes, a number of possible paths may exist between a node and a link so each of these are considered separately before being consolidated later. This process is followed for each direction $d_\tau \in (-1, 1)$ arrived at. The most likely position x_τ is found as the mode of the normally distributed model which is within the interval of the link.

- **Link to Node:** If the previous state $\mathbf{x}_{\tau-1}$ is in a link, it has a continuous position $x_{\tau-1}$. The computation of the likelihood of a transition to a node i_τ can be found similarly to the *node to node* case, where the normally distributed model for distance travelled from the estimated position $x_{\tau-1}$ can be integrated in the interval in distance corresponding to the node, and again each possible path can be considered separately. In this case, the normally distributed model can consider the uncertainty in both the position x_τ and in the distance travelled. The standard deviation of the distance travelled is equal to the square root of the sum of the squared standard deviation of the distributions for position and for distance travelled.

- **Link to Link:** The likelihood of moving from a given link $i_{\tau-1}$ to a given link i_τ can be found by combining the approaches used above, taking the normally distributed model

for distance travelled from position $x_{\tau-1}$ and integrating it over the interval in distance corresponding to the link i_τ.

In all cases, the probability density for the angle taken for a given transition is found from a model of the uncertain angle measurement. A heavy-tailed distribution in the range $[0, 2\pi)$ is used for robustness.

7.4.3 Implementation of Viterbi Algorithm

The approach described previously is implemented as follows, where the probability over states at time τ is computed from the estimated states at time $\tau - 1$. This approach is described in more detail in Algorithm 7. Algorithm 6 describes the implementation of the Viterbi algorithm as a whole, including the forward and backwards parts, as described in general in Algorithm 5.

The map of the environment defines the state space for the localization estimate, so is available to the algorithm. For computational efficiency, the various expected distances between each pair of places are computed prior to the estimation, denoted by Υ, and are available in the map $\mathcal{M}$ used by the algorithm. This reduces computational time at the cost of increased storage required.

For each estimated state $\mathbf{x}_{\tau-1}$, the likelihood of a transition to each other node and link state at time τ is found using the odometry information as described above. A further truncation is used here for efficiency: states at time τ for which the minimum required distance travelled from $\mathbf{x}_{\tau-1}$ is much larger than the measured distance travelled are assumed to have a probability of zero and are not computed exactly. For the transition between the state $\mathbf{x}_{\tau-1}$ and each other place, there will likely be a number of possible paths and corresponding expected distances and angles travelled. For each of these paths, an estimated state and likelihood is computed.

Once each state $\mathbf{x}_{\tau-1}$ has been considered, the resulting estimates $\mathbf{x}_\tau$ can be consolidated. The first of two key aspects of the Viterbi algorithm is that the estimates in the

Algorithm 6 The Viterbi algorithm implementation

For each time index forward through time
for time $\tau = \{1, 2, \ldots, T\}$ **do**
  Get the measurements for time τ
  get $\mathbf{u}_\tau$ and $\mathbf{z}_\tau$
  get the set of estimates $\mathcal{X}_\tau$ using Algorithm 7
  $\tilde{\mathcal{X}}_\tau = \left\{\tilde{\mathcal{X}}_\tau^1, \tilde{\mathcal{X}}_\tau^2, \ldots, \tilde{\mathcal{X}}_\tau^j\right\}$, $\tilde{\mathcal{X}}_\tau^j = \left\{\mathbf{x}_\tau^j = \left[i_\tau^j, x_\tau^j, d_\tau^j\right], k_\tau^j, p(\mathbf{x}_{0:\tau}^j|\mathbf{x}_0, \mathbf{u}_{1:\tau-1}, \mathbf{z}_{1:\tau-1}), \sigma_{x_\tau}^j\right\}$
  consolidate the set of estimates $\tilde{\mathcal{X}}_\tau$ using Algorithm 8
  $\mathcal{X}_\tau \subset \tilde{\mathcal{X}}_\tau$
end for

Find the trajectory with the highest likelihood
$J_T = \arg\max_j p(\mathbf{x}_{0:\tau}^j|\mathbf{x}_0, \mathbf{u}_{1:\tau-1}, \mathbf{z}_{1:\tau-1})$
Get the state that ends this trajectory
$\hat{\mathbf{x}}_T = \mathbf{x}_T^{J_T}$

For each time index backward though time
for time $\tau = \{T, T-1, \ldots, 2\}$ **do**
  Get the recorded index of the most likely previous state
  $J_{\tau-1} = k_\tau^{J_\tau}$
  Get the state at this index
  $\hat{\mathbf{x}}_{\tau-1} = \mathbf{x}^{J_{\tau-1}}$
end for

set are compared and for each subset with equal states $\mathbf{x}_\tau$ only the most likely is kept in the set, including the record of its most likely previous state, while the other estimates are removed. Then, the likelihood of each estimate in the resulting set is compared to a threshold probability, and the estimate is removed if its likelihood is low. This process is followed for each time index $\tau \in \mathcal{T}_\tau$, computing the probabilities of likely sequences of states starting with the known state $\mathbf{x}_0$. The second key aspect of the Viterbi algorithm is that once the most likely sequence ending at time τ is found, the recorded most likely previous state can be used to simply recover the sequence of states back through time.

7.4.4 Additional Measurements

While this algorithm has been derived to use inputs from integrated linear and angular odometry in $\mathbf{u}_\tau$ and observations of either a node or a link in $\mathbf{z}_\tau$, other information could

Algorithm 7 The forward Viterbi algorithm implementation

Compute the likelihood of states for time τ and state estimate at time $\tau - 1$
Get the measurements for time τ
get $\mathbf{u}_\tau$ and $\mathbf{z}_\tau$
initialise $\mathcal{X}_\tau = \emptyset$
for each estimated state at time $\tau - 1$
for state $\tilde{\mathbf{x}}^k_{\tau-1}, \in \mathcal{X}_{\tau-1}$ **do**
  get the state $\tilde{\mathbf{x}}^k_{\tau-1} = \left[\tilde{i}^k_{\tau-1}, \tilde{x}^k_{\tau-1}, \tilde{d}^k_{\tau-1}\right]$ and uncertainty $\sigma^k_{x_\tau}$
  for each discrete index at time τ
  for index $i \in \mathcal{I}$ **do**
    For each discrete direction at time τ
    for direction $d \in \mathcal{D}^i$ **do**
      get discrete state $\mathbf{x}^j_\tau = \left\{i, \emptyset, d\right\}$
      For each path between states
      for $\upsilon^{\mathbf{x}^j}_{\mathbf{x}^k} \in \Upsilon^{\mathbf{x}^j}_{\mathbf{x}^k}$ **do**
        get hypothetical discrete state $\mathbf{x}^j_\tau = \left\{i, \emptyset, d\right\}$
        Get the expected odometry interval and angular odometry for this path
        $\mathbf{u}^{\mathbf{x}^j}_{\mathbf{x}^k} = [\Delta x^{\mathbf{x}^j}_{\mathbf{x}^k,1}, \Delta x^{\mathbf{x}^j}_{\mathbf{x}^k,2}, \Delta\theta^{\mathbf{x}^j}_{\mathbf{x}^k}] = f(\upsilon^{\mathbf{x}^j}_{\mathbf{x}^k})$
        Get the probability distribution over distance travelled
        $p(\Delta x_\tau) = \mathcal{N}(\mu = \Delta x_\tau, \sigma = \sqrt{{\sigma_{x_\tau}}^2 + {\sigma^k_{x_\tau}}^2})$
        Get the probability for discrete state $\mathbf{x}^j_\tau$ and sequence $p(\mathbf{x}^j_{0:\tau}|\mathbf{x}_0, \mathbf{u}_{1:\tau}, \mathbf{z}_{1:\tau})$
        $p(\mathbf{x}^j_\tau|\mathbf{x}^k_{\tau-1}, \mathbf{u}_\tau, \mathbf{z}_\tau) = p(\Delta\theta_\tau|\Delta\theta^{\mathbf{x}^j}_{\mathbf{x}^k}) p(\mathbf{z}_\tau|\mathbf{x}^j_\tau) \int_{\Delta x^{\mathbf{x}^j}_{\mathbf{x}^k,1}}^{\Delta x^{\mathbf{x}^j}_{\mathbf{x}^k,2}} p(\Delta x_\tau) d\Delta x_\tau$
        $p(\mathbf{x}^j_{0:\tau}|\mathbf{x}_0, \mathbf{u}_{1:\tau}, \mathbf{z}_{1:\tau}) = p(\mathbf{x}^j_\tau|\mathbf{x}^k_{\tau-1}, \mathbf{u}_\tau, \mathbf{z}_\tau) p(\mathbf{x}^k_{\tau-1}|\mathbf{x}_0, \mathbf{u}_{1:\tau-1}, \mathbf{z}_{1:\tau-1})$
        Get continuous state and uncertainty
        $\hat{\Delta x}^j_\tau = \arg\max_{\Delta x_\tau} p(\Delta x_\tau), \Delta x^{\mathbf{x}^j}_{\mathbf{x}^k,1} \leq \Delta x_\tau \leq \Delta x^{\mathbf{x}^j}_{\mathbf{x}^k,2}$
        $$\hat{x}^j_\tau = \begin{cases} 0 & \text{if } i^j \in \mathcal{N} \\ \begin{cases} \hat{\Delta x}^j_\tau - \Delta x^{\mathbf{x}^j}_{\mathbf{x}^k,1} & \text{if } d^j = 1 \\ x^{L,i} - (\hat{\Delta x}^j_\tau - x^{\mathbf{x}^j}_{\mathbf{x}^k,1}) & \text{if } d^j = -1 \end{cases} & \text{if } i^j \in \mathcal{L} \end{cases}$$
        $$\sigma^j_{x_\tau} = \begin{cases} 0 & \text{if } i^j \in \mathcal{N} \\ \sqrt{{\sigma_{x_\tau}}^2 + {\sigma^k_{x_\tau}}^2} & \text{if } i^j \in \mathcal{L} \end{cases}$$
        Add to estimate
        $\mathcal{X}^j_\tau = \left\{\mathbf{x}^j_\tau = \left[i^j_\tau, x^j_\tau, d^j_\tau\right], k^j_\tau, p(\mathbf{x}^j_{0:\tau}|\mathbf{x}_0, \mathbf{u}_{1:\tau}, \mathbf{z}_{1:\tau}), \sigma^j_{x_\tau}\right\}$
        $\mathcal{X}_\tau \Longleftarrow \mathcal{X}^j_\tau$
      end for
    end for
  end for
end for

Algorithm 8 The Viterbi algorithm implementation

Compare each state in the set of estimates
for each pair of estimates in the set of estimates
for $\mathcal{X}_\tau^j, \mathcal{X}_\tau^k \in \mathcal{X}_\tau, j \neq k$ **do**
  compare the two states
  if $\mathbf{x}_\tau^j = \mathbf{x}_\tau^k$ **then**
    if the two states are the same, remove the state with the lower probability
    $L = \arg\min_{l \in j,k} p(\mathbf{x}_{0:\tau}^l | \mathbf{x}_0, \mathbf{u}_{1:\tau}, \mathbf{z}_{1:\tau})$
    $\mathcal{X}_\tau^L = \emptyset$
  end if
end for

be found either instantaneously or through integration over the time step τ which could be incorporated into $\mathbf{z}_\tau$.

The first source of additional information used here is the identification of a nearby node which could be done by use of a *beacon* [154] which the robot can detect and uniquely identify. This is implemented in the algorithm presented here as a simple multiplication of the probability of each estimate $\mathbf{x}_\tau$ by a factor given by $p(o_\tau|\mathbf{x}_\tau)$.

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

The third source of additional information is from the estimation of the length of a link Λ_τ using means such as acoustic echoes seen in Chapter 5. This is simply implemented in the algorithm presented here as a multiplication of the probability of each estimate $\mathbf{x}_\tau$ by a factor given by $p(\Lambda_\tau|\mathbf{x}_\tau)$. To reflect the uncertainty in the measurement of the length of a link using acoustic echoes, the following model is used to simulate the length measurement, where a number of possible link lengths are found, as detailed in previously published work Rob Worley, Yicheng Yu, and Sean Anderson. "Acoustic echo-localization for pipe inspection robots". In: *IEEE International Conference on Multisensor Fusion and Integration for Intel-*

Table 7.1: *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	0.5 m
Motion noise constant	k_v	0.8
False positive rate	β_p	0.005
False negative rate	β_n	0.05

ligent Systems (2020), pp. 2–7. URL: `https://ieeexplore.ieee.org/document/9235225`. The true link length is found given the simulated robot's position in the environment. There is some uncertainty to the exact value of link length, there is some probability that the true link length is not found, the set of lengths contains a number of multiples of the true length and a number of random false measurements.

7.5 Results and Discussion: Comparison of Algorithms

This section presents an experiment and discussion comparing two algorithms:

1. The *1Dt* algorithm presented in Chapter 6 (where it is there referred to as the *1D+* algorithm[1]), which is the novel improved version of the novel *1D* algorithm, which itself is an improvement to the *2D* algorithm in the application to high uncertainty robots which is presented in the literature [66].

2. The proposed *1D*$\boldsymbol{\tau}$ algorithm presented in this chapter.

An example of the performance of these two algorithms, compared also with the *2D*

[1]The name of this algorithm has been changed in this chapter to better describe the difference between it and the algorithm it is compared to.

Table 7.2: *Default parameters for the localization algorithms*

	Parameter	Symbol	Value
1Dt	Motion model noise	$\tilde{\sigma}_x$	$1.2\sigma_x$
	Node transition std.	σ_n	Δx m
	Angular weight std.	$\tilde{\sigma}_\theta$	$10\sigma_\theta$
	Kernel std.	σ_g	$5\Delta x$ m
	Number of New Particles	H_D	10
	Number of Particles Decay Constant	K_{h0}	0.05
	Divergence Short Time Constant	$\tau_{\alpha s}$	20
	Divergence Long Time Constant	$\tau_{\alpha l}$	100
	Hypothesis Weight Threshold	α^0	0.01
1Dτ	Motion model noise	$\tilde{\sigma}_x$	$4\sigma_{x_\tau}\sqrt{T^\tau_{\tau-1}}$
	Angular motion model noise	$\tilde{\sigma}_\theta$	$2\sigma_\theta$
	Angular motion model noise minimum	$\tilde{\sigma}_{\theta,min}$	0.2

algorithm described in Chapter 6, is given in Figure 7.3. The trajectory estimated by each algorithm is shown on a map of part of the pipe network used in this experiment, which shows the large scale of the network compared to the length of the trajectory used, which is around 4.5 kilometers. The trajectory estimate error can be compared for each algorithm. The estimate from the *2D* algorithm is seen to eventually become incorrect and fail to relocalize. The *1Dt* algorithm is seen to successfully relocalize over the whole trajectory. The *1Dτ* algorithm is seen to have a consistently accurate estimate, as any periods of mislocalization are relocalized over all time steps. These results are only shown for illustrative purposes. The experiment over a large number of trajectories is described in the rest of this section.

As in Chapter 6, the algorithms are compared by using simulation to create a large number of trajectories through a pipe network, with variation in uncertainty in motion and in measurements. As described in Chapter 6, the two algorithms are compared in terms of *error rate*, which is the proportion of time where the estimated position is in the wrong discrete location. This is estimated as described in Section 6.7.

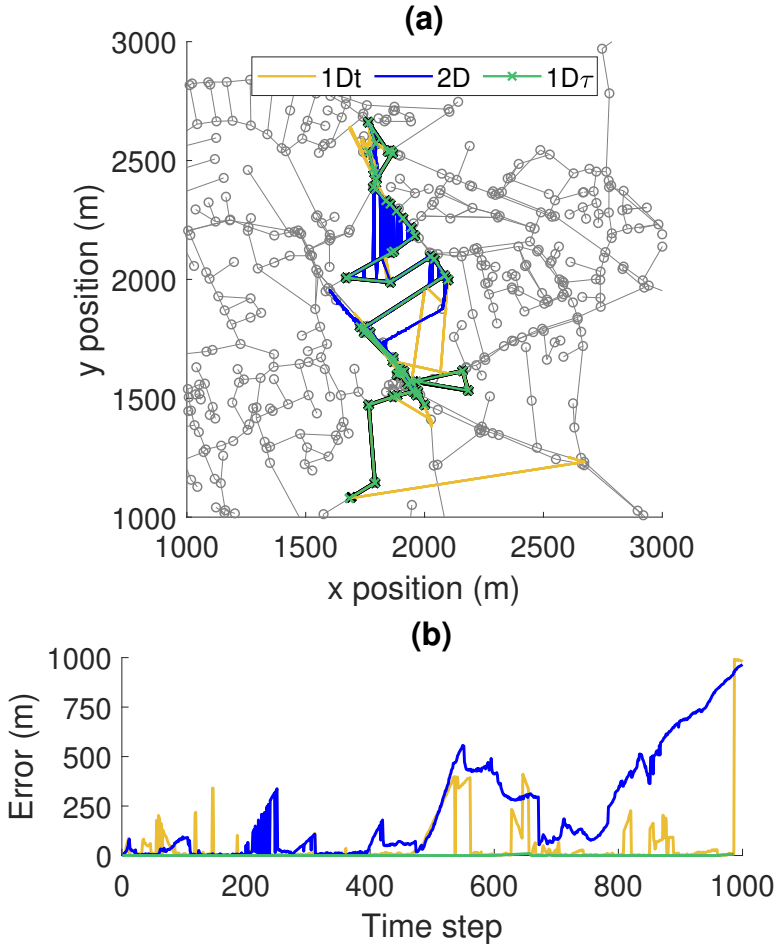

Figure 7.3: *Illustrative example results showing:* ***(a)*** *and estimated trajectory, and* ***(b)*** *the estimate error. Three different estimate algorithms are compared, the 1Dt, 1Dτ, and 2D algorithms.*

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

1. Normally distributed uncertainty in measured linear motion. The robot is modelled as moving forwards where its actual motion is equal to a measured motion with added normally distributed uncertainty with standard deviation σ_x, as described by equations 6.6 and 6.7.

2. Uncertainty in measured angular motion, where the robot's measured angular motion is modelled as being equal to the actual motion with added normally distributed uncertainty with standard deviation σ_θ, as described by equations 6.9 and 6.10.

3. Integrated uniformly distributed uncertainty in measured linear motion. The robot is modelled as moving forwards where its actual motion is equal to a measured motion with added integrated uniformly distributed uncertainty, in the range u_t, as described by equation 6.8. The added uncertainty is therefore dynamic, which is challenging to the algorithms which use a model of uncorrelated uncertainty.

4. Uncertainty in the detection of nodes, given by the rates of false positive and false negative detection, described by equation 6.11.

The error rate is measured over different values of these uncertainty variables. For each value, the error rate is measured over 50 trajectories through the pipe network. The default parameters describing uncertainty in the simulated robot's motion and measurements are given in Table 7.1. The parameters used by each of the two localization algorithms as described earlier in this chapter and in Chapter 6 are given in Table 7.2.

It should be noted that these two algorithms take different approaches to the localization problem, and the results are not easily to directly compare. The principal differences are that the *1Dt* algorithm iteratively estimates the instantaneous position of a robot at a mostly regular interval in time t, while the *1D*τ algorithm estimates the full sequence of robot positions at arbitrary instances in time τ where the robot makes an informative measurement, as discussed previously in this chapter. However, while the two algorithms aim to achieve different things, they can be compared in terms of accuracy, reliability, and efficiency. To best compare the performance of the two algorithms, the estimates given by the *1Dt* and *1D*τ algorithms will be evaluated only at nodes, as this is when the majority of information is acquired by the robot, which avoids biasing the results towards the *1D*τ algorithm.

Figure 7.4 shows the results of this comparison. In each case, the results are similar, and it is seen that the error rate for the *1Dt* algorithm increases with the increase in uncertainty, while the error rate for the *1D*τ algorithm remains constant or increases less.

Figures 7.4(a) and (b) show that the *1D*τ algorithm has a low median error rate of less than 0.05 at the largest measured value of linear uncertainty, where the *1Dt* algorithm's median error rate is above 0.1 and many trajectories have an error rate greater than 0.5.

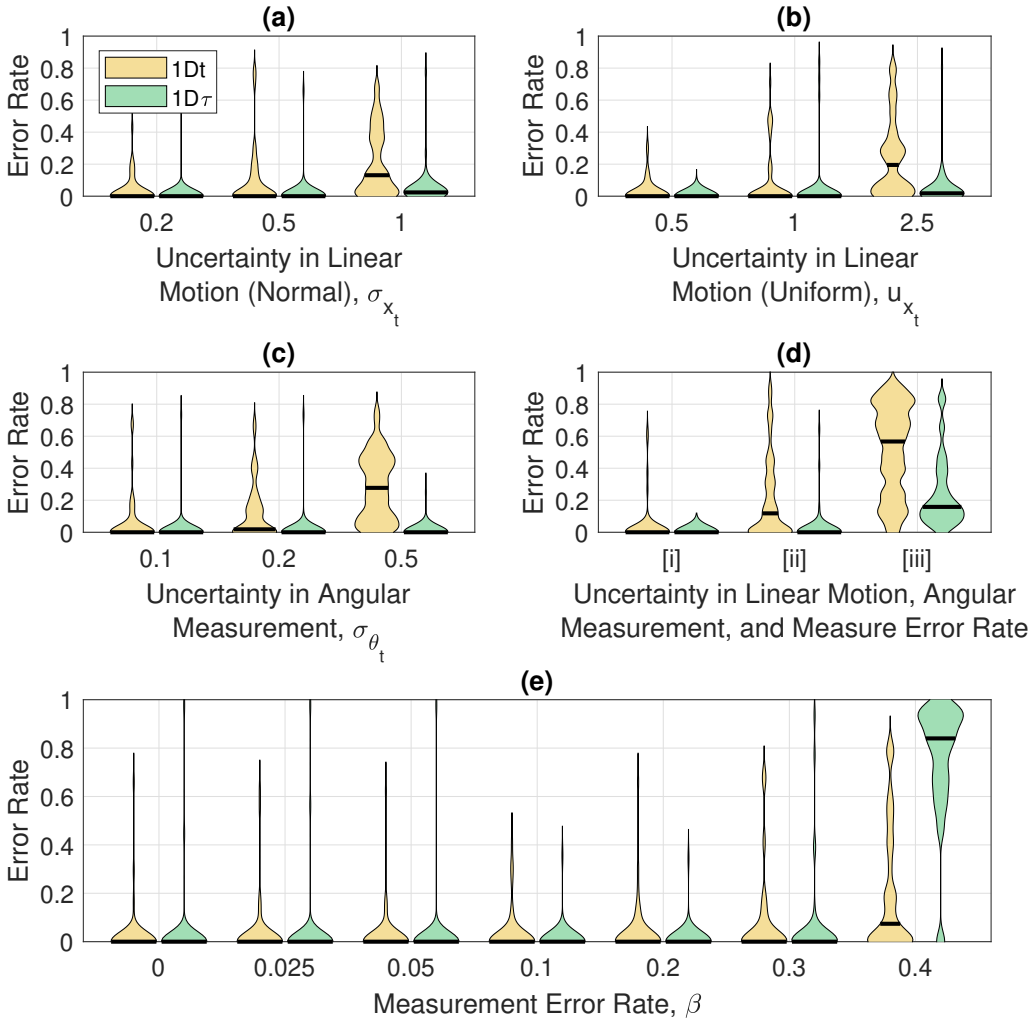

Figure 7.4: *A comparison of the Error rate with variation in:* ***(a)*** σ_{x_t}. ***(b)*** u_{x_t}. ***(c)*** σ_{θ_t}. ***(e)*** β. ***(d)*** *[i]:* $\sigma_{x_t} = 0.2$, $\sigma_{\theta_t} = 0.1$, $\beta = 0$, *[ii]:* $\sigma_{x_t} = 0.5$, $\sigma_{\theta_t} = 0.2$, $\beta = 0.05$, *[iii]:* $\sigma_{x_t} = 1$, $\sigma_{\theta_t} = 0.5$, $\beta = 0.1$.

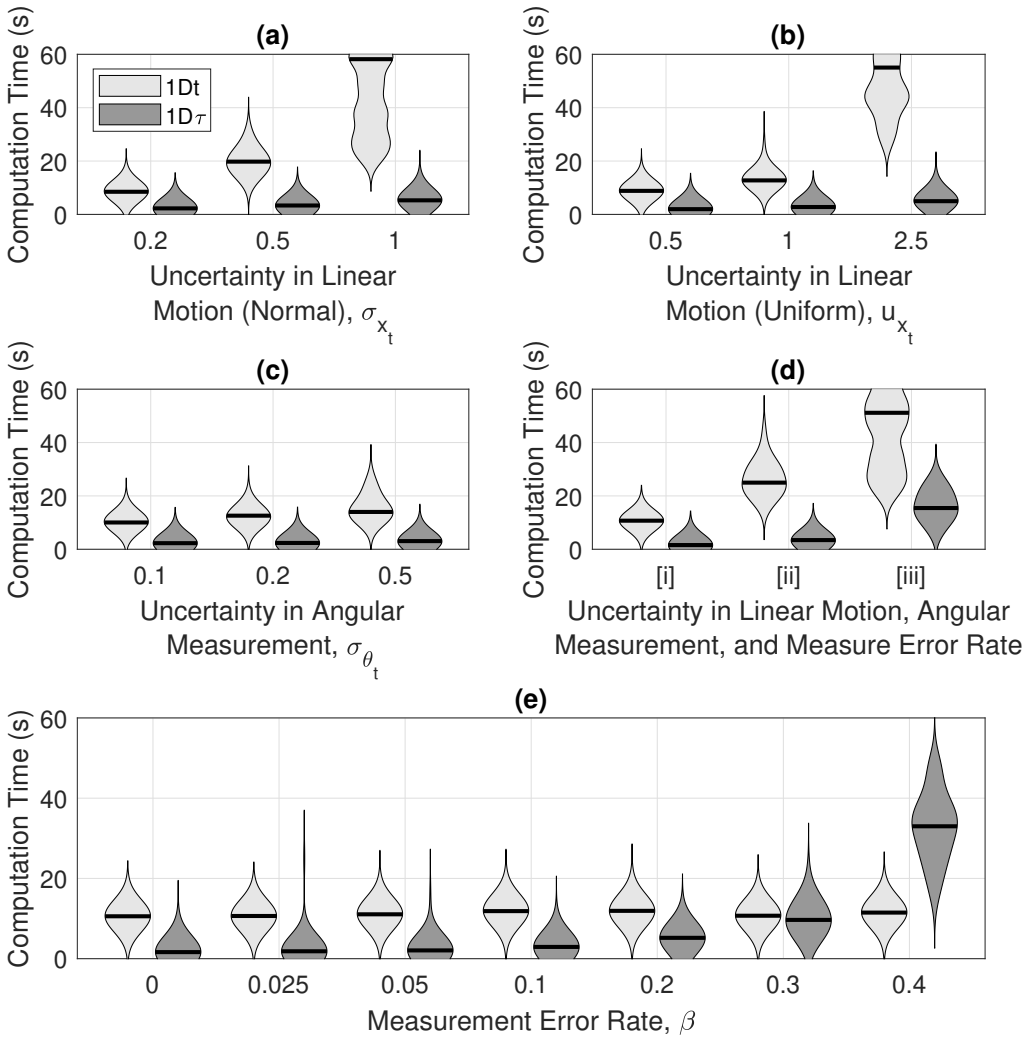

Figure 7.5: *A comparison of the computation time with variation in:* ***(a)*** σ_{x_t}. ***(b)*** u_{x_t}. ***(c)*** σ_{θ_t}. ***(e)*** β. ***(d)*** *[i]:* $\sigma_{x_t} = 0.2$, $\sigma_{\theta_t} = 0.1$, $\beta = 0$, *[ii]:* $\sigma_{x_t} = 0.5$, $\sigma_{\theta_t} = 0.2$, $\beta = 0.05$, *[iii]:* $\sigma_{x_t} = 1$, $\sigma_{\theta_t} = 0.5$, $\beta = 0.1$.

For the linear uncertainty, a value of $2.5\sigma_x$ is approximately equal to u_x in terms of average measurement error produced, although the integrated uniform uncertainty produces dynamic correlated error which could impact the estimate more. Chapter 1 described the range of error in linear motion of robots in pipes in the literature as being up to 35%. The median error rate of zero for linear motion uncertainty of 50% of the distance travelled shows that the algorithm performs well for uncertainty comparable to that expected in practice, while the median error rate of less than 0.05 for linear motion uncertainty of 100% of the distance travelled therefore indicates that this method is robust to uncertainty that might be expected in application in more challenging realistic pipe environments.

Figure 7.4(c) shows that the *1D*$\boldsymbol{\tau}$ algorithm has a median error rate of 0 even at the largest measured value of angular motion, where the median error rate of the *1Dt* algorithm is above 0.25.

Figure 7.4(e) shows the good performance of both algorithms in the presence of large measurement error rate. The error rate of both algorithms stays low or at zero for a measurement error rate of up to 0.3. However, the error rate of the *1D*$\boldsymbol{\tau}$ algorithm increases to a median value greater than 0.8 at larger measurement error rates, and increases much less for the *1Dt* algorithm, showing the expected robustness to measurement error as described in Chapter 6.

Figure 7.4(d) compares the performance of the two algorithms over an increase in all of these uncertainty parameters simultaneously. A similar trend is seen as in the other comparisons, as the error rate of the *1Dt* algorithm increases to a median value of greater than 0.5 with the increase in uncertainty, while the error rate of the *1D*$\boldsymbol{\tau}$ algorithm stays low and only increases to a median value of less than 0.2. From this it can be seen that while the *1D*$\boldsymbol{\tau}$ algorithm is robust to an increase in each source of uncertainty individually, its robustness is still limited when using only the simple measurements used here as inputs. This motivates the incorporation of additional information from observations of the environment.

Figure 7.5 shows a comparison of the computation time used by each algorithm corresponding to the estimate error rate shown in Figure 7.4, over the same variation in

uncertainty parameters. Generally, it can be seen that the computational cost of the *1Dτ* algorithm is lower than that of the *1Dt* algorithm. Figure 7.5(a) and (b) show that as uncertainty in linear motion increases, the cost of the *1Dt* algorithm increases while the cost of the *1Dτ* algorithm remains relatively constant. Figure 7.5(c) shows a similar trend but with a smaller difference between the two algorithms. Figure 7.5(d) shows that the computation time for the *1Dτ* algorithm increases for larger values of all uncertainty, which correlates with the increase in error rate seen for the same uncertainty parameters. Figure 7.5(e) shows that the computation time of the *1Dτ* algorithm increases and becomes larger than that of the *1Dt* algorithm. This corresponds to the substantial increase in error rate of the *1Dτ* algorithm, and the small increase in error rate of the *1Dt* algorithm. Overall, considering both the results for error rate and computation time, it can be seen that the *1Dτ* algorithm is substantially more efficient than the *1Dt* algorithm for most tested values of uncertainty, with a lower error rate and lower computational cost. It is seen here that lower error rate for a given algorithm corresponds to a lower computational cost, due to the less uncertain inputs resulting in fewer likely possible states which reduces the chance of error and reduces cost. However, both algorithms could likely be improved in required computational cost, so this comparison is not necessarily conclusive.

7.6 Results and Discussion: Improvement to the Algorithm

This section presents an experiment and discussion investigating the incorporation of additional information from the environment into the *1Dτ* algorithm. In the previous section, the inputs to the algorithm were simply linear and angular odometry, and the detection of nodes in the pipe network. In Section 7.4.4, three additional sources of information were proposed: unique identification of a node, loop-closure, and estimation of the length of a link. Here, the effect of these additional sources of information is measured in comparison to a estimate using only simple inputs with the higher uncertainty values used previously which give a median error rate of around 0.15. Figure 7.6 shows the results of these comparisons.

For each source of additional information, the frequency of measurements is varied

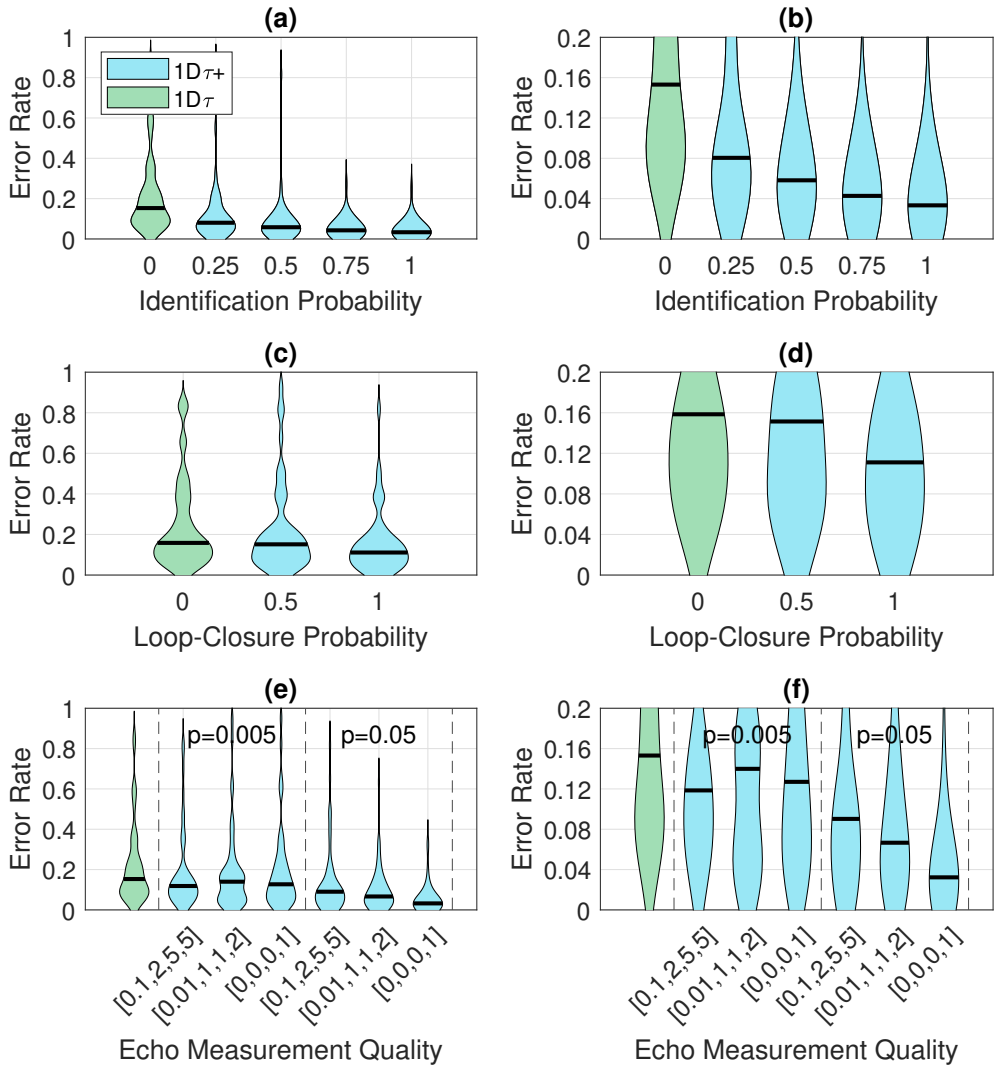

Figure 7.6: *The effect of additional information (1Dτ+) on localization.* ***(a)*** *and* ***(b)*** *Uniquely identification of nodes.* ***(c)*** *and* ***(d)*** *Loop-closure measurements.* ***(e)*** *and* ***(f)*** *Estimates of link length using acoustic echoes.*

to measure the improvement to the error rate for different amounts of cost from additional sensing. The *identification probability* is varied which determines the proportion of nodes in the network which can be identified. A subset of nodes, in the given proportion to the total number of nodes, are chosen to be identifiable. As described previously, this measurement could be obtained by having beacons which the robot can detect and identify placed around the network. The increase in *identification probability* therefore corresponds to an increase in cost of additional hardware. The *loop-closure probability* determines the probability that the robot is able to recognise a location that it has previously visited. The increase in probability therefore corresponds to an increase in the required perceptive ability of the robot, and also to an extent adds some requirements to the path the robot takes. The *echo measurement probability* determines the reliability and precision of the measurement of pipe length made using acoustic echoes. As described in Chapter 5, the estimate of pipe length is made probabilistically, and a probability density is found over the continuous space of possible pipe lengths. This density is typically larger around the true pipe length, around multiples of the pipe length, and around random values smaller than the true pipe length. The precision at which this pipe length can be estimated also depends on the precision of the acoustic echo detection, so the measured length is modelled as having added normally distributed uncertainty.

Figures 7.6(a) and (b) show the effect of adding identification of nodes to the estimation. The error rate is seen to decrease with increasing proportion of nodes being identified. With identification probability of greater than 0.75, there are no measured error rates above 0.4. The median error rate is seen to decrease most substantially with the increase of identification probability from 0 to 0.25, and decreases less so for further increases in identification probability. The modal error rate is seen to decrease almost linearly with increasing identification probability. This shows a clear improvement found for increasing cost of additional sensing, but offers inconclusive evidence as to whether the relationship between cost and effect is linear or not. Further experimentation would be needed to determine this relationship, and further experimentation could also be done to determine the effect of a deterministic rather than stochastic placement of identifiable nodes. The nonzero error rate for an identification probability of 1 might be due to some estimates being made in links for which the estimate

of continuous position could be greater than the threshold for error. It might also be because the current implementation only incorporates the identification measurement into estimates made using the simple odometry and node observation measurements. Therefore, if the correct node is not predicted by these simple measurements, the identification will not be able to improve the estimate.

Figures 7.6(c) and (d) show the effect of increasing probability of loop-closures. The error rate is seen to decrease only slightly, from a median of 0.16 to a median of 0.11, even for a probability of 1 of making loop-closure observations. This could be explained by the low probability that a given loop-closure observation occurs at a time when it improves the estimate. With the current approach, in order to correct an otherwise erroneous estimate at time τ, a loop-closure must occur between the current time and a previous time for which the estimate is correct. The probability of this occurring is lower than the equivalent probability of making an identification observation at a time when the estimate would otherwise be erroneous, which only requires the coincidence of the timing of one observation rather than the coincidence of the timing of two observations. It is possible that loop-closure measurements would be much more impactful in a *simultaneous localization and mapping* application, compared to this localization problem.

Figures 7.6(e) and (f) show the effect of increasing echo measurement quality and rate of echo measurements on the estimate. It can be seen that the error rate generally decreases when echo measurements of any quality are used, although the improvement to accuracy is highest for the highest quality echo measurements. The incorporation of acoustic echoes is likely to share some properties with the identification of nodes. Acquiring the sequence of acoustic echo measurements needed to make an estimate of pipe length has some cost. The estimate of pipe length is likely more accurate with more cost, so here the reduction in error rate is found at increasing cost. The relative cost between making acoustic measurements and placing beacons in the network for identification is beyond the scope of the work presented here. An improvement to estimate accuracy is more likely to be found when the estimate would otherwise have been wrong, which is dependent on the random probability of making an echo estimate. This is shown in Figures 7.6(e) and (f) where the

results found when using two different probabilities are compared, and the error is reduced for a larger probability of echo measurement. Unlike the identification of a node using a beacon placed in the network, acoustic echo measurements can be made by the robot at any time. An improvement to localization efficiency might be found by taking an active localization approach where the robot makes an acoustic measurement when the uncertainty in its localization estimate is large, therefore efficiently acquiring useful measurements when they are more informative.

The incorporation of node identification measurements, echo measurements, and to a lesser extent loop-closure measurements, is seen to improve the performance of the algorithm for the values of uncertainty investigated. The incorporation of these additional measurements has some associated cost, either in additional hardware or in additional time and energy required for sensing. While the experiment here has measured the improvement in algorithm performance with relative cost for each additional measurement, further work would be required to establish the absolute cost so that development towards an optimal solution could be made.

7.7 Conclusions

This section will summarise the results presented in this chapter and draw some conclusions which will motivate future work beyond the scope of this thesis.

The experiments here have compared the newly presented *1D*τ algorithm which uses an approach derived from the Viterbi algorithm to estimate the most likely sequence of states given the sequence of measurements, with the *1Dt* algorithm which uses a particle filter with additional processes to detect and recover from error. The *1Dt* algorithm was shown in Chapter 6 to improve on the reliability of the estimate compared to the *1D* algorithm, which itself was shown to improve upon the performance of the *2D* algorithm in application to limited measurements with large uncertainty. The *1D*τ algorithm is shown to be an effective further development of the idea of making the localization estimate in a suitable low-dimensional space. The *1D*τ algorithm is shown to have higher accuracy and robustness

than the *1Dt* algorithm for a given set of input uncertainty values, and is shown to require less computation. For linear motion uncertainty, the *1D*τ algorithm is shown to have a low median error rate for uncertainty larger than that expected in practical application from the literature. Developments to the *1D*τ algorithm are likely possible which would further improve the accuracy, robustness, and computational cost.

The refinement of the estimation problem to a subset of all time indices brings the problem formulation closer to a form which could be solved analytically. Mathematical tools which are beyond the scope of this thesis likely exist which could develop the localization algorithm further. The design of the algorithm presented here could be improved, or an alternative algorithm might be found. It may be possible to estimate an upper bound on the performance of an algorithm, providing evidence beyond numerical and statistical analysis of the accuracy and robustness of the algorithm. An investigation into the effect on the estimate of the pipe network topology and geometry also might be possible.

Chapter 8

Conclusions

This thesis has improved robot localization systems for application in pipe networks in terms of accuracy, reliability, and efficiency. With the understanding of the useful abstraction to different scales present in this environment: small-scale localization within a pipe cross section, medium-scale localization along a pipe axis, and large-scale localization in a pipe network, localization approaches have been developed across different levels in scope and precision. This achieves the thesis aims described in Section 1.3, as detailed here.

An investigation of the literature on existing front-end approaches for this application in Chapter 2 determined that existing vision or rangefinding sensing is capable of small-scale localization and of making observations of features useful for large-scale localization. However, these typical approaches are limited in effectiveness for medium-scale localization in this environment. Hydrophone sensing is investigated using experimental data and developed beyond previous work using the same data [135, 33] in Chapter 4. It is shown that it can be used to detect more features in the otherwise feature-sparse environment, and a novel augmented pose-graph optimization can be used to give a low error rate in localization, with a median error rate of zero in the best case. However, the error rate is not reliably reduced, with around 50% of the trajectory estimates having a nonzero error rate in the best case, as this sensing approach is still limited in that it is only able to observe nearby features.

A novel *acoustic echo* front-end, well suited for medium-scale localization in this environment, has been developed and integrated into a localization back-end, which results in good performance demonstrated with experimental measurements in Chapter 5. The use of acoustic echoes has been demonstrated here in a complex realistic pipe environment, and the use of a novel extended pose-graph optimization algorithm has been shown to be necessary, and shown to produce a median localization error rate close to zero for values of uncertainty greater than those required by the problem definition made in Chapter 1. Compared to the approach in Chapter 4, the reliability of the acoustic echo approach is higher, as the upper quartile of the error rate is reduced from 0.5 to less than 0.05 for comparable values of robot uncertainty. This adds to the literature regarding alternative sensing approaches in this environment, showing a more direct means of estimating distance to features, and showing improvement to flexibility of operation compared to previous use of acoustic propagation [69] and of other propagating waves [38]. Despite the effectiveness of the proposed method, it should be noted that vision is still most prevalent in the literature on existing front-end approaches. The conclusions here do not imply that vision is unsuitable in this application, only that it is more limited in this application than in more typical environments. Effective use of vision has been demonstrated previously [68], and further development has been seen since then. A more thorough comparison between front-end approaches, considering the robot system as a whole including information about the desired sensing method for pipe fault detection, would be needed to better assess the optimal front-end for a given application.

The investigation of available back-end methods concludes in Chapter 6 and Chapter 7 that a *hybrid continuous-discrete* approach is well suited to large-scale localization of robots in the pipe environment with limited sensor measurements. This approach usefully simplifies and constrains the estimation problem compared to the *continuous* approach explored in Chapter 3. It has been shown that the *continuous-discrete* approach can usefully incorporate information from both existing and novel methods of front-end perception. This challenges the consensus of the literature for this application which largely takes a *continuous* approach. This consensus may be simply a default approach rather than a well evidenced method, and this thesis contributes evidence to the literature that alternative approaches could be considered which are especially well suited to this environment. A direct comparison is made

between a *continuous* space particle filter algorithm presented in recent literature [66] which is well suited for capable robots with low uncertainty perception, and a novel *continuous-discrete* space particle filter algorithm designed for the large uncertainty perception of small robots in this environment, which shows the advantages to accuracy, cost, and robustness which can be gained using the proposed method. The proposed novel algorithm is shown to have an average error rate of zero for problems within the definition made in Chapter 1. This thesis also contributes a useful investigation into the effect and effectiveness of a range of measurement types on localization.

Development of the localization solution for this problem is not yet complete. Localization development is progressing alongside that of every other aspect of this robotic system. This includes locomotion, sensing, control, and communication, and also includes aspects of the system as a whole, such as whether the system should be made up of a group of powerful robots, a swarm of simple robots, or a set of individual robots distributed around a network. This thesis contributes work towards localization of a single robot in a largely known environment which has been mostly assumed to be mapped accurately. However, further work could contribute to the literature as follows.

Implementation of the developed localization approaches in online, real-time localization would better demonstrate the feasibility of the approach to application in industry, and also result in better understanding of the strengths and weaknesses of the approach. Work in this direction would be a substantial contribution to the literature, and could produce data which could be used for further development of the robotic system. However, it would come with the challenge of implementing all aspects of a robot system to a suitable level which was beyond the scope of this thesis, and the challenge of accessing suitable experimental or field conditions.

Substantial work could be done in development of a simultaneous localization and mapping method, which could localize a robot in an unknown map or a map that is partially known either in the sense of some regions of the map being inaccurately mapped previously, or in the sense that the map topology is known but not the metric detail. Similarly, substantial work could be done in the development of localization of multiple robots. Complexity here

could be found from the limits on possible communication and from the possible differences in capabilities between robots. Integration between localization and control of the robot, or robots, could produce more reliable localization than is possible otherwise. While this thesis presented algorithms for detecting a likely error in localization, the subsequent recovery would likely be more effective if a more active approach to recovery was taken. Investigation into the effect of network topology and geometry on the localization result could be useful for improved localization robustness and integration with robot control. In the work here, the presented algorithms have been assessed in their performance over the network as a whole, but it is likely that particular parts of a given network provide a greater or lesser challenge to localization. For example, parts of the network consisting of a number of pipes of similar length may be more challenging to localize in. Incorporation of the dynamics of the environment, either slow changes in pipe condition or fast changes in fluid flow and the position of other robots, could be useful for localization. Depending on the purpose of the network, these aspects might vary periodically with daily use or be influenced by factors such as the weather. The *continuous-discrete* state definition usefully acknowledges the network topology so could facilitate this additional information. Established methods in the field of buried infrastructure modelling could be usefully applied here, as could aspects from the increasingly popular idea of *digital twins*, creating some integration between the robot localization system and the infrastructure system as a whole.

Further work could be done in finding some analytical upper or lower limit on localization performance for a given algorithm. This would improve upon the results presented in this thesis where the performance of the algorithms are evaluated numerically, showing a statistical measure of performance measures such as error rate and failure rate. While it is likely that methods not used in the work presented here would be required for this work, taking it beyond the scope of this thesis, an analytical guarantee on performance would be desirable both from an academic perspective and the perspective of industry.

References

[1] Alan C Twort, Don D Ratnayaka, and Malcolm J Brandt. *Water Supply*. Vol. 15. 2. 2000, pp. 1–23. ISBN: 2013206534.

[2] David Butler and John W Davies. *Urban Drainage, Fourth Edition*. 2004. ISBN: 0203149696. DOI: 10.1201/9781351174305.

[3] GOV.UK. "Water and treated water". In: (2015). URL: https://www.gov.uk/government/publications/water-and-treated-water/water-and-treated-water.

[4] Government Office for Science. "UK Water Research and Innovation Framework:Taking Responsibility for Water". In: (2012). URL: https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment{_}data/file/294176/11-1416-taking-responsibility-for-water-summary.pdf.

[5] Destatis. *Öffentliche Wasserversorgung und öffentliche Abwasserentsorgung*. Vol. 49. 0. 2018. ISBN: 2190213169.

[6] Www.infrastructurereportcard.org. "Wastewater". In: (2021), pp. 151–161.

[7] The Consumer Council for Water. "Water, water everywhere. Delivering resilient water and waste water services everywhere (2017-2018)". In: (2017). ISSN: 00130613. URL: https://www.ccwater.org.uk/research/water-water-everywhere-delivering-resilient-water-and-wastewater-services-2017-18/.

[8] Prabhata K. Swamee and Ashok K. Sharma. *DESIGN OF WATER SUPPLY PIPE NETWORKS*. Vol. 7. 1. John Wiley & Sons, 2008. arXiv: arXiv:1011.1669v3.

[9] HSA. "Code of Practice for Avoiding Danger from Underground Services". In: January (2005), p. 48.

[10] Institution of Civil Engineers. *Specification for underground utility detection, verification and location (PAS128:2014)*. Tech. rep. 2014, p. 36.

[11] STERLING CODIFIERS. *Heyworth, IL Code of Ordinances*. Cincinnati, Ohio, 2011.

[12] S. B. Costello et al. "Underground asset location and condition assessment technologies". In: *Tunnelling and Underground Space Technology* 22.5-6 (2007), pp. 524–542. ISSN: 08867798. DOI: 10.1016/j.tust.2007.06.001.

[13] Zheng Liu and Yehuda Kleiner. "State of the art review of inspection technologies for condition assessment of water pipes". In: *Measurement: Journal of the International Measurement Confederation* 46.1 (2013), pp. 1–15. ISSN: 02632241. DOI: 10.1016/j.measurement.2012.05.032. URL: http://dx.doi.org/10.1016/j.measurement.2012.05.032.

[14] Heesik Jang et al. "A Review : Technological Trends and Development Direction of Pipeline Robot Systems". In: *Journal of Intelligent & Robotic Systems* (2022), pp. 1–20. ISSN: 1573-0409. DOI: 10.1007/s10846-022-01669-2. URL: https://doi.org/10.1007/s10846-022-01669-2.

[15] Josep M Mirats-Tur and William Garthwaite. "Robotic Devices for Water Main In-Pipe Inspection A Survey". In: 27.4 (2010), pp. 491–508. DOI: 10.1002/rob.

[16] George H Mills, Andrew E Jackson, and Robert C Richardson. "Advances in the Inspection of Unpiggable Pipelines". In: *Robotics* 6.4 (2017), p. 36. DOI: 10.3390/robotics6040036.

[17] Tokuji Okada and Tsuyoshi Sanemori. "Mogrer: A Vehicle Study and Realization for In-Pipe Inspection Tasks". In: *IEEE Journal on Robotics and Automation* 3.6 (1987), pp. 573–582. ISSN: 08824967. DOI: 10.1109/JRA.1987.1087149.

[18] Tokuji Okada and Takeo Kanade. "A Three-Wheeled Self-Adjusting Vehicle in a Pipe, FERRET-1". In: *The International Journal of Robotics Research* 6.4 (1987), pp. 60–75. ISSN: 17413176. DOI: 10.1177/027836498700600406.

[19] Frank Kirchner and Joachim Hertzberg. "A prototype study of an autonomous robot platform for sewerage system maintenance". In: *Autonomous Robots* 12 (1997), pp. 319–331. URL: https://link.springer.com/article/10.1023/A:1008896121662.

[20] Joachim Hertzberg and Frank Kirchner. "Landmark-based autonomous navigation in sewerage pipes". In: *Proceedings of the 1st Euromicro Workshop on Advanced Mobile Robots, EUROBOT 1996* (1996), pp. 68–73. DOI: 10.1109/EURBOT.1996.551883.

[21] Erich Rome et al. "Towards autonomous sewer robots: the MAKRO project". In: *Urban Water* 1.1 (1999), pp. 57–70. ISSN: 1462-0758. DOI: http://dx.doi.org/10.1016/S1462-0758(99)00012-6.

[22] Robin Kirkham et al. "PIRAT — A System for Quantitative Sewer". In: *The International Journal of Robotics Research* 19:11.November (2000).

[23] S.G. Roh et al. "Actively steerable in-pipe inspection robots for underground urban gas pipelines". In: *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No.01CH37164)* 1 (2002), pp. 761–766. ISSN: 00400262. DOI: 10.1109/robot.2001.932642.

[24] Andres Zagler and Friedrich Pfeiffer. ""MORITZ" a pipe crawler for tube junctions". In: *Proceedings - IEEE International Conference on Robotics and Automation* 3 (2003), pp. 2954–2959. ISSN: 10504729.

[25] Amir A.F. Nassiraei et al. "Concept and design of a fully autonomous sewer pipe inspection mobile robot "KANTARO"". In: *Proceedings - IEEE International Conference on Robotics and Automation* April (2007), pp. 136–143. ISSN: 10504729. DOI: 10.1109/ROBOT.2007.363777.

[26] Fabien Tâche et al. "MagneBike: Compact magnetic wheeled robot for power plant inspection". In: *2010 1st International Conference on Applied Robotics for the Power Industry, CARPI 2010* May 2014 (2010). DOI: 10.1109/CARPI.2010.5624442.

[27] You Wu et al. "Modeling and parameter estimation for in-pipe swimming robots". In: *Proceedings of the American Control Conference* 2015-July (2015), pp. 2007–2013. ISSN: 07431619. DOI: 10.1109/ACC.2015.7171028.

[28] Paulo Debenest, Michele Guarnieri, and Shigeo Hirose. "PipeTron series-Robots for pipe inspection". In: *Proceedings of the 3rd International Conference on Applied Robotics for the Power Industry, CARPI 2014* (2015), pp. 1–6. DOI: 10.1109/CARPI.2014.7030052.

[29] You Wu et al. "Design of a Leak Sensor for Operating Water Pipe Systems". In: *International Conference on Intelligent Robots and Systems (IROS)* (2017), pp. 6075–6082.

[30] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic Robotics*. The MIT Press, 2006. ISBN: 0262201623.

[31] Andreu Corominas Murtra and Josep M. Mirats Tur. "IMU and cable encoder data fusion for in-pipe mobile robot localization". In: *IEEE Conference on Technologies for Practical Robot Applications, TePRA* (2013). ISSN: 23250526. DOI: 10.1109/TePRA.2013.6556377.

[32] Hussein Sahli and Naser El-Sheimy. "A novel method to enhance pipeline trajectory determination using pipeline junctions". In: *Sensors (Switzerland)* 16.4 (2016), pp. 1–17. ISSN: 14248220. DOI: 10.3390/s16040567. URL: https://www.mdpi.com/1424-8220/16/4/567.

[33] Ke Ma et al. "PipeSLAM: Simultaneous Localisation and Mapping in Feature Sparse Water Pipes using the Rao-Blackwellised Particle Filter". In: *IEEE/ASME International Conference on Advanced Intelligent Mechatronics, AIM* (2017), pp. 1459–1464. DOI: 10.1109/AIM.2017.8014224.

[34] Wasim Al-Masri. "Inertial Navigation system of In-pipe Inspection Robot". In: May (2016). URL: https://dspace.aus.edu:8443/xmlui/handle/11073/8331.

[35] Wasim M.F. Al-Masri, Mamoun F. Abdel-Hafez, and Mohammad A. Jaradat. "Inertial Navigation System of Pipeline Inspection Gauge". In: *IEEE Transactions on Control Systems Technology* PP (2018), pp. 1–8. ISSN: 1558-0865. DOI: 10.1109/TCST.2018.2879628.

[36] You Wu. "Low-cost soft sensors and robots for leak detection in operating water pipes". In: (2018). URL: https://dspace.mit.edu/handle/1721.1/118022.

[37] You Wu et al. "A practical minimalism approach to in-pipe robot localization". In: *Proceedings of the American Control Conference* 2019-July (2019), pp. 3180–3187. ISSN: 07431619. DOI: 10.23919/acc.2019.8814648. URL: https://ieeexplore.ieee.org/document/8814648.

[38] Carlos Rizzo et al. "An alternative approach for robot localization inside pipes using RF spatial fadings". In: *Robotics and Autonomous Systems* 136 (2021), p. 103702. ISSN: 09218890. DOI: 10.1016/j.robot.2020.103702. URL: https://doi.org/10.1016/j.robot.2020.103702.

[39] Amal Gunatilake, Sarath Kodagoda, and Karthick Thiyagarajan. "A Novel UHF-RFID Dual Antenna Signals Combined With Gaussian Process and Particle Filter for In-Pipe Robot Localization". In: *IEEE Robotics and Automation Letters* 7.3 (2022), pp. 6005–6011. ISSN: 23773766. DOI: 10.1109/LRA.2022.3163769.

[40] Saber Kazeminasab et al. "Localization, Mapping, Navigation, and Inspection Methods in In-Pipe Robots: A Review". In: *IEEE Access* 9 (2021), pp. 162035–162058. ISSN: 21693536. DOI: 10.1109/ACCESS.2021.3130233.

[41] Andreas Geiger, Philip Lenz, and Raquel Urtasun. "Are we ready for autonomous driving? the KITTI vision benchmark suite". In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2012), pp. 3354–3361. ISSN: 10636919. DOI: 10.1109/CVPR.2012.6248074.

[42] John G. Rogers et al. "Test Your SLAM! the SubT-Tunnel dataset and metric for mapping". In: *Proceedings - IEEE International Conference on Robotics and Automation* May 2020 (2020), pp. 955–961. ISSN: 10504729. DOI: 10.1109/ICRA40945.2020.9197156.

[43] David Alejo et al. "Into the dirt: Datasets of sewer networks with aerial and ground platforms". In: *Journal of Field Robotics* 38(1).July (2020). ISSN: 15564967. DOI: 10.1002/rob.21976.

[44] Raul Mur-Artal, J. M.M. Montiel, and Juan D. Tardos. "ORB-SLAM: A Versatile and Accurate Monocular SLAM System". In: *IEEE Transactions on Robotics* 31.5 (2015), pp. 1147–1163. ISSN: 15523098. DOI: 10.1109/TRO.2015.2463671.

[45] Sho Kagami et al. "3D Pipe Network Reconstruction Based on Structure from Motion with Incremental Conic Shape Detection and Cylindrical Constraint". In: (2020). arXiv: 2006.10383. URL: http://arxiv.org/abs/2006.10383.

[46] Peter Hansen et al. "Pipe mapping with monocular fisheye imagery". In: *IEEE International Conference on Intelligent Robots and Systems* (2013), pp. 5180–5185. ISSN: 21530858. DOI: 10.1109/IROS.2013.6697105.

[47] Reid Simmons and Sven Koenig. "Probabilistic Robot Navigation in Partially Observable Environments". In: *IJCAI'95: Proceedings of the 14th international joint conference on Artificial intelligence* 2 (1995), pp. 1080–1087.

[48] Anthony R. Cassandra, Leslie Pack Kaelbling, and James A. Kurien. "Acting under uncertainty: discrete Bayesian models for mobile-robot navigation". In: *IEEE International Conference on Intelligent Robots and Systems* 2.May (1996), pp. 963–972. DOI: 10.1109/iros.1996.571080.

[49] Benjamin Kuipers and Yung Tai Byun. "A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations". In: *Robotics and Autonomous Systems* 8.1-2 (1991), pp. 47–63. ISSN: 09218890. DOI: 10.1016/0921-8890(91)90014-C.

[50] David Kortenkamp and Terry Weymouth. "Topological mapping for mobile robots using a combination of sonar and vision sensing". In: *Proceedings of the National Conference on Artificial Intelligence* 2 (1994), pp. 979–984.

[51] Sebastian Thrun. "Learning metric-topological maps for indoor mobile robot navigation". In: *Artificial Intelligence* 99.1 (1998), pp. 21–71. ISSN: 00043702. DOI: 10.1016/S0004-3702(97)00078-7.

[52] H. Choset and K. Nagatani. "Topological simultaneous localization and mapping (SLAM): towardexact localization without explicit localization". In: *IEEE Transactions on Robotics and Automation* 17.2 (2001), pp. 125–137.

[53] Dieter Fox, Wolfram Burgard, and Sebastian Thrun. "Markov localization for mobile robots in dynamic environments". In: *Journal of Artificial Intelligence Research* (1999).

[54] Dong Hyuk Lee et al. "Map building method for urban gas pipelines based on landmark detection". In: *International Journal of Control, Automation and Systems* 11.1 (2013), pp. 127–135. ISSN: 15986446. DOI: 10.1007/s12555-012-0049-6. URL: https://link.springer.com/article/10.1007/s12555-012-0049-6.

[55] C. Gomez et al. "Uncertainty-based localization in a topological robot navigation system". In: *2017 IEEE International Conference on Autonomous Robot Systems and Competitions, ICARSC 2017* (2017), pp. 67–72. DOI: 10.1109/ICARSC.2017.7964054.

[56] Clara Gomez et al. "Qualitative Geometrical Uncertainty in a Topological Robot Localization System". In: *Proceedings - 2018 International Conference on Control, Artificial Intelligence, Robotics and Optimization, ICCAIRO 2018* (2018), pp. 183–188. DOI: 10.1109/ICCAIRO.2018.00038.

[57] Wolfram Burgard et al. "Integrating global position estimation and position tracking for mobile robots: The dynamic Markov localization approach". In: *IEEE International Conference on Intelligent Robots and Systems* 2 (1998), pp. 730–735. DOI: 10.1109/iros.1998.727279.

[58] P. Payeur et al. "Probabilistic octree modeling of a 3-D dynamic environment". In: *Proceedings - IEEE International Conference on Robotics and Automation* 2.April (1997), pp. 1289–1296. ISSN: 10504729. DOI: 10.1109/robot.1997.614315.

[59] Armin Hornung et al. "OctoMap: An efficient probabilistic 3D mapping framework based on octrees". In: *Autonomous Robots* 34.3 (2013), pp. 189–206. ISSN: 09295593. DOI: 10.1007/s10514-012-9321-0.

[60] Randall C Smith and Peter Cheeseman. "On the Representation and Estimation of Spatial Uncertainty". In: *The International Journal of Robotics Research* 5.4 (1986).

[61] Randall C Smith, Matthew Self, and Peter Cheeseman. "Estimating Uncertain Spatial Relationships in Robotics". In: *Proceedings of the Second Conference on Uncertainty in Artificial Intelligence (UAI1986)* (1990). ISSN: 0289-1824. DOI: 10.7210/jrsj.6.83.

[62] F. C. Park, J. E. Bobrow, and S. R. Ploen. *A Lie Group Formulation of Robot Dynamics*. 1995.

[63] Edward Pervin and Jon A. Webb. "Quaternions in Computer Vision and Robotics." In: (1983), pp. 382–383.

[64] Cesar Cadena et al. "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age". In: *IEEE Transactions on Robotics* 32.6 (2016), pp. 1309–1332. ISSN: 15523098. DOI: 10.1109/TRO.2016.2624754. arXiv: 1606.05830.

[65] Francesco Amigoni et al. "A standard for map data representation: IEEE 1873-2015 facilitates interoperability between robots". In: *IEEE Robotics and Automation Magazine* 25.1 (2018), pp. 65–76. ISSN: 10709932. DOI: 10.1109/MRA.2017.2746179.

[66] David Alejo, Fernando Caballero, and Luis Merino. "A robust localization system for inspection robots in sewer networks". In: *Sensors (Switzerland)* 19.22 (2019), pp. 1–28. ISSN: 14248220. DOI: 10.3390/s19224946. URL: https://www.mdpi.com/1424-8220/19/22/4946.

[67] Hoon Lim et al. "SLAM in indoor pipelines with 15mm diameter". In: *Proceedings - IEEE International Conference on Robotics and Automation* (2008), pp. 4005–4011. ISSN: 10504729. DOI: 10.1109/ROBOT.2008.4543826.

[68] Peter Hansen et al. "Visual mapping for natural gas pipe inspection". In: *International Journal of Robotics Research* 34.4-5 (2015), pp. 532–538. ISSN: 17413176. DOI: 10.1177/0278364914550133. URL: https://journals.sagepub.com/doi/abs/10.1177/0278364914550133.

[69] Yoshiaki Bando et al. "Sound-based online localization for an in-pipe snake robot". In: *SSRR 2016 - International Symposium on Safety, Security and Rescue Robotics* (2016), pp. 207–213. DOI: 10.1109/SSRR.2016.7784300. URL: https://ieeexplore.ieee.org/document/7784300.

[70] Dong Hyuk Lee, Hyungpil Moon, and Hyouk Ryeol Choi. "Autonomous navigation of in-pipe working robot in unknown pipeline Environment". In: *Proceedings - IEEE International Conference on Robotics and Automation* (2011), pp. 1559–1564. ISSN: 10504729. DOI: 10.1109/ICRA.2011.5980503.

[71] Jose-luis Blanco, Juan-antonio Fernandez-Madrigal, and Javier Gonzalez. "Toward a unified Bayesian approach to hybrid metric – topological SLAM". In: *IEEE TRANSACTIONS ON ROBOTICS, VOL. 24* 24.2 (2008), pp. 259–270. URL: https://ieeexplore.ieee.org/document/4472721.

[72] Tom Duckett and Ulrich Nehmzow. "Mobile robot self-localization using occupancy histograms and a mixture of Gaussian location hypotheses". In: *Robotics and Autonomous Systems* 34.2-3 (2001), pp. 117–129. ISSN: 09218890. DOI: 10.1016/S0921-8890(00)00116-0.

[73] Haris Baltzakis and Panos Trahanias. "Hybrid mobile robot localization using switching state-space models". In: *Proceedings - IEEE International Conference on Robotics and Automation* 1.May (2002), pp. 366–373. ISSN: 10504729. DOI: 10.1109/robot.2002.1013388.

[74] Yan Zhuang. "Hybrid mobile robot indoor localization using large-scale metric-topological map". In: *Proceedings of the World Congress on Intelligent Control and Automation (WCICA)* 2 (2006), pp. 9057–9062. DOI: 10.1109/WCICA.2006.1713752.

[75] Sylvie Thiebaux and Peter Lamb. *Combining Kalman filters and Markov localization in network-like environments*. 2000. URL: https://link.springer.com/chapter/10.1007/3-540-44533-1{_}75.

[76] P. Merriaux et al. "Fast and robust vehicle positioning on graph-based representation of drivable maps". In: *Proceedings - IEEE International Conference on Robotics and Automation* 2015-June.June (2015), pp. 2787–2793. ISSN: 10504729. DOI: 10.1109/ICRA.2015.7139578. URL: https://ieeexplore.ieee.org/abstract/document/7139578.

[77] Marcus A Brubaker, Andreas Geiger, and Raquel Urtasun. "Map-based probabilistic visual self-localization". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 38.4 (2016), pp. 652–665. DOI: 10.1109/TPAMI.2015.2453975. URL: https://ieeexplore.ieee.org/document/7152950.

[78] Fernando Bernuy and Javier Ruiz-del Solar. "Topological semantic mapping and localization in urban road scenarios". In: *Journal of Intelligent and Robotic Systems: Theory*

and Applications 92.1 (2018), pp. 19–32. ISSN: 15730409. DOI: 10.1007/s10846-017-0744-x. URL: https://link.springer.com/article/10.1007/s10846-017-0744-x.

[79] R.E Kalman. "A New Approach to Linear Filtering and Prediction Problems". In: 21.2 (1992), pp. 125–147. ISSN: 00223611. DOI: 10.1007/BF00248635.

[80] Simon J Julier and Jeffrey K Uhlmann. "A New Extension of the Kalman Filter to Nonlinear Systems". In: *Proceedings - 1997 Signal Processing, Sensor Fusion, and Target Recognition VI* (1997).

[81] Ruben Martinez-Cantin. "Unscented SLAM for Large-Scale Outdoor Environments". In: *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems* (2005).

[82] Michael Csorba. "Simultaneous Localisation and Map Building". PhD thesis. University of Oxford, 1997.

[83] Dieter Fox et al. "Monte Carlo Localization: efficient position estimation for mobile robots". In: *Proceedings of the National Conference on Artificial Intelligence* Handschin 1970 (1999), pp. 343–349.

[84] Arnaud Doucet et al. "Rao-Blackwellised Particle Filtering for Dynamic Bayesian Networks". In: *UNCERTAINTY IN ARTIFICIAL INTELLIGENCE PROCEEDINGS* (2000). ISSN: 17519675. DOI: 10.1049/iet-spr:20070075.

[85] Michael Montemerlo et al. "FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem". In: *Proceedings of the AAAI National Conference on Artificial Intelligence* (2002). ISSN: 14746670. DOI: 10.3182/20070903-3-fr-2921.00092.

[86] Sebastian Thrun and Michael Montemerlo. "The GraphSLAM Algorithm with Applications to Large-Scale Mapping of Urban Structures". In: *The International Journal of Robotics Research* 25.5 (2006), pp. 403–429. ISSN: 08554307. DOI: 10.1177/0278364906065387.

[87] Frank Dellaert and Michael Kaess. "Square root SAM: Simultaneous localization and mapping via square root information smoothing". In: *International Journal of Robotics Research* 25.12 (2006), pp. 1181–1203. ISSN: 02783649. DOI: 10.1177/0278364906072768.

[88] Michael Kaess, Ananth Ranganathan, and Frank Dellaert. "iSAM: Fast incremental smoothing and mapping with efficient data association". In: *Proceedings - IEEE International Conference on Robotics and Automation* April (2007), pp. 1670–1677. ISSN: 10504729. DOI: 10.1109/ROBOT.2007.363563.

[89] Michael Kaess et al. "ISAM2: Incremental smoothing and mapping using the Bayes tree". In: *International Journal of Robotics Research* 31.2 (2012), pp. 216–235. ISSN: 02783649. DOI: 10.1177/0278364911430419.

[90] Ming Hsiao and Michael Kaess. "MH-iSAM2 : Multi-hypothesis iSAM using Bayes Tree and Hypo-tree". In: *IEEE International Conference on Robotics and Automation* (2019), pp. 1274–1280.

[91] G.D. Forney. "The viterbi algorithm". In: *Proc. IEEE* 61 (1973), pp. 268–278. DOI: 10.1049/ic:20060556.

[92] Jakob Engel, Jurgen Sturm, and Daniel Cremers. "LSD-SLAM: Large-Scale Direct Monocular SLAM". In: *Proceedings of the IEEE International Conference on Computer Vision* (2013), pp. 1449–1456. ISSN: 00201693.

[93] Jakob Engel, Vladlen Koltun, and Daniel Cremers. "Direct Sparse Odometry". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40.3 (2018), pp. 611–625. ISSN: 01628828. DOI: 10.1109/TPAMI.2017.2658577. arXiv: 1607.02565.

[94] Haiyang Chao, Yu Gu, and Marcello Napolitano. "A survey of optical flow techniques for robotics navigation applications". In: *Journal of Intelligent and Robotic Systems: Theory and Applications* 73.1-4 (2014), pp. 361–372. ISSN: 15730409. DOI: 10.1007/s10846-013-9923-6.

[95] Denis Fortun, Patrick Bouthemy, and Charles Kervrann. "Optical flow modeling and computation: A survey". In: *Computer Vision and Image Understanding* 134 (2015), pp. 1–21. ISSN: 1090235X. DOI: 10.1016/j.cviu.2015.02.008. URL: http://dx.doi.org/10.1016/j.cviu.2015.02.008.

[96] David G Lowe. "Distinctive Image Features from Scale-Invariant Keypoints". In: *International Journal ofComputerVision* (2004), pp. 1–28.

[97] H Bay, T Tuytelaars, and L Van Gool. "SURF: Speeded Up Robust Features". In: *Proceedings of the 17th Central European Seminar on …* (2006).

[98] Ethan Rublee et al. "ORB: and efficient alternative to SIFT and SURF". In: *2011 IEEE International Conference on Computer Vision (ICCV)* (2011). ISSN: 00043591. DOI: 10.1002/art.38045. URL: http://www.willowgarage.com/sites/default/files/orb{_}final.pdf.

[99] R. Mur-Artal and J.D. Tardós. "ORB-SLAM2 An Open-Source SLAM System for Monocular Stereo.pdf". In: *IEEE TRANSACTIONS ON ROBOTICS* 33.5 (2017), pp. 1255–1262.

[100] Tong Qin, Peiliang Li, and Shaojie Shen. "VINS-Mono: A Robust and Versatile Monocular Visual-Inertial State Estimator". In: *IEEE Transactions on Robotics* 34.4 (2018), pp. 1004–1020. ISSN: 15523098. DOI: 10.1109/TRO.2018.2853729. arXiv: 1708.03852.

[101] Homayoun Najjaran and Dennis Krys. "INS-ASSISTED MONOCULAR ROBOT LOCALIZATION". In: *Proceedings of the ASME 2010 International Mechanical Engineering Congress & Exposition* (2010), pp. 1–8.

[102] Peter Hansen et al. "Stereo visual odometry for pipe mapping". In: *IEEE International Conference on Intelligent Robots and Systems* (2011), pp. 4020–4025. ISSN: 2153-0858. DOI: 10.1109/IROS.2011.6048579.

[103] Rahul Summan et al. "A novel visual pipework inspection system". In: *AIP Conference Proceedings* 1949.April (2018). ISSN: 15517616. DOI: 10.1063/1.5031647.

[104] Jorge Moraleda, Anibal Ollero, and Mariano Orte. "Examining the Interior of Water Pipelines with Minimal Service Interruption Via A New Visual Inspection System Relevant Characteristics of Water Pipelines". In: *IEEE Robotics and Automation Magazine* September (1999).

[105] Olga Duran, Kaspar Althoefer, and Lakmal D. Seneviratne. "Experiments using a laser-based transducer and automated analysis techniques for pipe inspection". In: *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on* 2.3 (2003), pp. 2561–2566.

[106] Wen Zhao et al. "Modeling and simulation of FLC-based navigation algorithm for small gas pipeline inspection robot". In: *IEEE/ASME International Conference on Advanced Intelligent Mechatronics, AIM* 2018-July (2018), pp. 912–917. DOI: 10.1109/AIM.2018.8452416. URL: https://ieeexplore.ieee.org/document/8452416.

[107] Jens T. Thielemann, Gøril M. Breivik, and Asbjørn Berge. "Pipeline landmark detection for autonomous robot navigation using time-of-flight imagery". In: *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops* (2008). DOI: 10.1109/CVPRW.2008.4563167. URL: https://ieeexplore.ieee.org/document/4563167.

[108] Atsushi Kakogawa, Yuki Komurasaki, and Shugen Ma. "Anisotropic shadow-based operation assistant for a pipeline-inspection robot using a single illuminator and camera". In: *IEEE International Conference on Intelligent Robots and Systems* 2017-Septe (2017), pp. 1305–1310. ISSN: 21530866. DOI: 10.1109/IROS.2017.8202306. URL: https://ieeexplore.ieee.org/document/8202306.

[109] Atsushi Kakogawa et al. "Recognition of pathway directions based on nonlinear least squares method". In: *2015 IEEE International Conference on Robotics and Biomimetics, IEEE-ROBIO 2015* (2015), pp. 1596–1601. DOI: 10.1109/ROBIO.2015.7418999. URL: https://ieeexplore.ieee.org/document/7418999.

[110] James L Crowley. "World Modeling and Position Estimation for a Mobile Robot Using Ultrasonic Ranging". In: *Proceedings, 1989 International Conference on Robotics and Automation* (1989).

[111] Ingemar J. Cox. "Blanche - An Experiment in Guidance and Navigation of an Autonomous Robot Vehicle". In: *IEEE Transactions on Robotics and Automation* 7.2 (1991).

[112] F. Lu and E. Milios. "Globally Consistent Range Scan Alignment for Environment Mapping". In: *Autonomous Robots* 4.4 (1997), pp. 333–349. ISSN: 09295593. DOI: 10.1023/A:1008854305733.

[113] Jesse Levinson and Sebastian Thrun. "Robust vehicle localization in urban environments using probabilistic maps". In: *Proceedings - IEEE International Conference*

on Robotics and Automation (2010), pp. 4372–4378. ISSN: 10504729. DOI: 10.1109/ROBOT.2010.5509700.

[114] Alberto Y. Hata and Denis F. Wolf. "Feature Detection for Vehicle Localization in Urban Environments Using a Multilayer LIDAR". In: *IEEE Transactions on Intelligent Transportation Systems* 17.2 (2016), pp. 420–429. ISSN: 15249050. DOI: 10.1109/TITS.2015.2477817.

[115] Frank Kirchner and Joachim Hertzberg. "A prototype study of an autonomous robot platform for sewerage system maintenance". In: *Autonomous Robots* 4.4 (1997), pp. 319–331. ISSN: 09295593. DOI: 10.1023/A:1008896121662.

[116] Tolga Özaslan et al. "Autonomous navigation and mapping for inspection of penstocks and tunnels with MAVs". In: *IEEE Robotics and Automation Letters* 2.3 (2017), pp. 1740–1747. ISSN: 23773766. DOI: 10.1109/LRA.2017.2699790. URL: https://ieeexplore.ieee.org/document/7914761.

[117] Christine Evers et al. "Acoustic SLAM". In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 26.9 (2018), pp. 1484–1498. DOI: 10.1109/TASLP.2018.2828321.

[118] J. Mourjopoulos. "On the variation and invertibility of room impulse response functions". In: *Journal of Sound and Vibration* 102.2 (1985), pp. 217–228. ISSN: 10958568. DOI: 10.1016/S0022-460X(85)80054-7.

[119] Miranda Krekovic, Ivan Dokmanic, and Martin Vetterli. "EchoSLAM : Simultaneous Localization and Mapping with Acousitc Echoes". In: *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2016), pp. 11–15. DOI: 10.1109/ICASSP.2016.7471627.

[120] Miranda Krekovic, Ivan Dokmanic, and Martin Vetterli. "Look, no Beacons! Optimal All-in-One EchoSLAM". In: *ArXiv* (2016). arXiv: 1608.08753. URL: http://arxiv.org/abs/1608.08753.

[121] Usama Saqib, Sharon Gannot, and Jesper Rindom Jensen. "Estimation of acoustic echoes using expectation-maximization methods". In: *Eurasip Journal on Audio,*

Speech, and Music Processing 2020.1 (2020). ISSN: 16874722. DOI: 10.1186/s13636-020-00179-z.

[122] Usama Saqib and Jesper Rindom Jensen. "A Framework for Spatial Map Generation using Acoustic Echoes for Robotic Platforms". In: (2021).

[123] Usama Saqib and Jesper Rindom Jensen. "A model-based approach to acoustic reflector localization with a robotic platform". In: *IEEE International Conference on Intelligent Robots and Systems* (2020), pp. 4499–4504. ISSN: 21530866. DOI: 10.1109/IROS45743.2020.9341437.

[124] Usama Saqib et al. "Robust Acoustic Reflector Localization for Robots". 2021.

[125] Usama Saqib et al. "DETECTING ACOUSTIC REFLECTORS USING A ROBOT'S EGO-NOISE". In: *2021 IEEE International Conference on Acoustics, Speech and Signal Processing* (2021), pp. 466–470.

[126] Xuesong Wang et al. "Detection of hydrate and other blockages in gas pipelines using acoustic reflectometry". In: *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science* 226.7 (2012), pp. 1800–1810. ISSN: 20412983. DOI: 10.1177/0954406211431029.

[127] K. A. Papadopoulou et al. "An evaluation of acoustic reflectometry for leakage and blockage detection". In: *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science* 222.6 (2008), pp. 959–966. ISSN: 09544062. DOI: 10.1243/09544062JMES873.

[128] M T Bin Ali, K V Horoshenkov, and S J Tait. "Rapid Detection of Sewer Defects and Blockages Using Acoustic Based Instrumentation". In: *Water Science & Technology* 64.8 (2010), pp. 1700–1707.

[129] Wenbo Duan et al. "On the use of power reflection ratio and phase change to determine the geometry of a blockage in a pipe". In: *Applied Acoustics* 87.December 2017 (2015), pp. 190–197. ISSN: 1872910X. DOI: 10.1016/j.apacoust.2014.07.002. URL: http://dx.doi.org/10.1016/j.apacoust.2014.07.002.

[130] Linan Tao. "MONITORING GAS DISTRIBUTION PIPELINES". In: (2017).

[131] Omar A Aldughayem. "ACOUSTIC PULSE REFLECTOMETRY FOR POTENTIAL INDUSTRIAL APPLICATIONS". In: (2018).

[132] Dan Murray et al. *Demonstration of Innovative Sewer System Inspection Technology SewerBatt ™*. Tech. rep. January. 2014.

[133] Ke Ma et al. "Robot Mapping and Localisation for Feature Sparse Water Pipes Using Voids as Landmarks". In: *Towards Autonomous Robotic Systems. 16th Annual Conference, TAROS 2015* (2015).

[134] D Chatzigeorgiou et al. "An In-Pipe Leak Detection Sensor : Sensing Capabilities and Evaluation". In: *Proceedings of the ASME 2011 International Design Engineering Technical Conferences & the 2011 ASME/IEEE International Conference on Mechatronic and Embedded Systems and Applications IDETC/MESA 2011 August 28-31, 2011, Washington, DC, USA* September 2015 (2011). DOI: 10.1115/DETC2011-48411.

[135] Ke Ma et al. "Robot mapping and localisation in metal water pipes using hydrophone induced vibration and map alignment by dynamic time warping". In: *Proceedings - IEEE International Conference on Robotics and Automation* (2017), pp. 2548–2553. ISSN: 10504729. DOI: 10.1109/ICRA.2017.7989296. URL: https://ieeexplore.ieee.org/abstract/document/7989296.

[136] Ke Ma et al. "PipeSLAM: Simultaneous Localisation and Mapping in Feature Sparse Water Pipes using the Rao-Blackwellised Particle Filter". In: *IEEE/ASME International Conference on Advanced Intelligent Mechatronics, AIM* (2017), pp. 1459–1464. DOI: 10.1109/AIM.2017.8014224.

[137] Gerasimos G. Samatas and Theodore P. Pachidis. "Inertial Measurement Units (IMUs) in Mobile Robots over the Last Five Years: A Review". In: *Designs* 6.1 (2022). ISSN: 24119660. DOI: 10.3390/designs6010017.

[138] Paul D. Groves. "Navigation using inertial sensors". In: *IEEE Aerospace and Electronic Systems Magazine* 30.2 (2015), pp. 42–69. ISSN: 08858985. DOI: 10.1109/MAES.2014.130191.

[139] Wei Zhao et al. "A detection system for pipeline direction based on shielded geomagnetic field". In: *International Journal of Pressure Vessels and Piping* 113 (2014), pp. 10–14. ISSN: 03080161. DOI: 10.1016/j.ijpvp.2013.11.001. URL: http://dx.doi.org/10.1016/j.ijpvp.2013.11.001.

[140] Byeolteo Park and Hyun Myung. "Resilient Underground Localization Using Magnetic Field Anomalies for Drilling Environment". In: *IEEE Transactions on Industrial Electronics* 65.2 (2018), pp. 1377–1387. DOI: 10.1109/TIE.2017.2733420.

[141] Andreu Corominas Murtra and Josep M. Mirats Tur. "IMU and cable encoder data fusion for in-pipe mobile robot localization". In: *IEEE Conference on Technologies for Practical Robot Applications, TePRA* (2013), pp. 1–6. ISSN: 23250526. DOI: 10.1109/TePRA.2013.6556377.

[142] Douglas Daniel Sampaio Santana, Newton Maruyama, and Celso Massatoshi Furukawa. "Estimation of trajectories of pipeline PIGs using inertial measurements and non linear sensor fusion". In: *2010 9th IEEE/IAS International Conference on Industry Applications, INDUSCON 2010* (2010), pp. 1–6. DOI: 10.1109/INDUSCON.2010.5739911.

[143] Md Sheruzzaman Chowdhury and Mamoun F. Abdel-Hafez. "Pipeline Inspection Gauge Position Estimation Using Inertial Measurement Unit, Odometer, and a Set of Reference Stations". In: *ASCE-ASME J. Risk and Uncert. in Engrg. Sys., Part B: Mech. Engrg.* 2.2 (2016), pp. 021001–1. ISSN: 2332-9017. DOI: 10.1115/1.4030945.

[144] Lianwu Guan et al. "Enhanced MEMS SINS Aided Pipeline Surveying System by Pipeline Junction Detection in Small Diameter Pipeline". In: *IFAC-PapersOnLine* 50.1 (2017), pp. 3560–3565. ISSN: 2405-8963. DOI: 10.1016/j.ifacol.2017.08.962. URL: https://doi.org/10.1016/j.ifacol.2017.08.962.

[145] Lianwu Guan et al. "Micro-inertial-aided high-precision positioning method for small-diameter PIG navigation". In: *Advances in Human and Machine Navigation Systems* (2018). URL: https://cdn.intechopen.com/pdfs/63868.pdf.

[146] Zhang Yu et al. "Pipeline inclination measurements based on a spherical detector with magnetic proximity switches". In: *IEEE Access* PP.August (2018), p. 1. DOI: 10.1109/ACCESS.2018.2856618.

[147] Jongdae Jung, Taekjun Oh, and Hyun Myung. "Magnetic field constraints and sequence-based matching for indoor pose graph SLAM". In: *Robotics and Autonomous Systems* 70 (2015), pp. 92–105. ISSN: 09218890. DOI: 10.1016/j.robot.2015.03.003. URL: http://dx.doi.org/10.1016/j.robot.2015.03.003.

[148] Sen Wang et al. "Keyframe based large-scale indoor localisation using geomagnetic field and motion pattern". In: *IEEE International Conference on Intelligent Robots and Systems* 2016-Novem (2016), pp. 1910–1917. ISSN: 21530866. DOI: 10.1109/IROS.2016.7759302.

[149] Chao Gao and Robert Harle. "MSGD: Scalable back-end for indoor magnetic field-based GraphSLAM". In: *Proceedings - IEEE International Conference on Robotics and Automation* (2017), pp. 3855–3862. ISSN: 10504729. DOI: 10.1109/ICRA.2017.7989444.

[150] Chris Xiaoxuan Lu et al. "Simultaneous localization and mapping with power network electromagnetic field". In: *Proceedings of the Annual International Conference on Mobile Computing and Networking, MOBICOM* (2018), pp. 607–622. DOI: 10.1145/3241539.3241540.

[151] Danial Waleed et al. "An In-Pipe Leak Detection Robot With a Neural-Network-Based Leak". In: *IEEE Sensors Journal* 19.3 (2019), pp. 1153–1165. DOI: 10.1109/JSEN.2018.2879248.

[152] Dimitris Chatzigeorgiou et al. "MIT Leak Detector : An In-Pipe Leak Detection Robot". In: *2014 IEEE International Conference on Robotics and Automation (ICRA)* (2014), p. 2091. DOI: 10.1109/ICRA.2014.6907144.

[153] Dalei Wu, Dimitris Chatzigeorgiou, and Kamal Youcef-toumi. "Node Localization in Robotic Sensor Networks for Pipeline Inspection". In: *IEEE Transactions on Industrial Informatics* 12.2 (2016), pp. 809–819. DOI: 10.1109/TII.2015.2469636.

[154] Martin Haug, Felix Lorenz, and Lauritz Thamsen. "GRAL: Localization of Floating Wireless Sensors in Pipe Networks". In: September (2021), pp. 251–257. DOI: 10.1109/ic2e52221.2021.00042. arXiv: 2109.00294.

[155] Viktor Doychinov and Ian D. Robertson. "Radio Wave Propagation Inside Buried Sewer Pipes for Infrastructure Robotics". In: April (2022), pp. 982–985. DOI: 10.23919/eumc50147.2022.9784221.

[156] Vladimír Kubelka, Maxime Vaidis, and François Pomerleau. "Gravity-constrained point cloud registration". In: (2022). arXiv: 2203.13799. URL: http://arxiv.org/abs/2203.13799.

[157] Yun Chang et al. "LAMP 2.0: A Robust Multi-Robot SLAM System for Operation in Challenging Large-Scale Underground Environments". In: (2022). arXiv: 2205.13135. URL: http://arxiv.org/abs/2205.13135.

[158] Christopher E. Denniston et al. "Loop Closure Prioritization for Efficient and Scalable Multi-Robot SLAM". In: (2022). arXiv: 2205.12402. URL: http://arxiv.org/abs/2205.12402.

[159] Adam Jacobson et al. "What localizes beneath: A metric multisensor localization and mapping system for autonomous underground mining vehicles". In: *Journal of Field Robotics* 38.1 (2021), pp. 5–27. ISSN: 15564967. DOI: 10.1002/rob.21978. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.21978.

[160] Rob Worley et al. "Robot localization in water pipes using acoustic signals and pose graph optimization". In: *Sensors (Switzerland)* (2020), pp. 1–23. DOI: 10.3390/s20195584. URL: https://www.mdpi.com/1424-8220/20/19/5584.

[161] Ke Ma. "Robot Mapping and Localisation in Water Pipes". In: December (2018).

[162] Philip E Gill, Walter Murray, and Margaret H Wright. *Practical Optimization*. The Bath Press, 1981. ISBN: 0-12-283950-1.

[163] Rob Worley, Yicheng Yu, and Sean Anderson. "Acoustic echo-localization for pipe inspection robots". In: *IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems* (2020), pp. 2–7. URL: https://ieeexplore.ieee.org/document/9235225.

[164] Usama Saqib and Jesper Rindom Jensen. "Sound-based distance estimation for indoor navigation in the presence of ego noise". In: *European Signal Processing Conference* 2019-Septe (2019), pp. 1–5. ISSN: 22195491. DOI: 10.23919/EUSIPCO.2019.8902694.

[165] Daniele Salvati, Carlo Drioli, and Gian Luca Foresti. "Sound Source and Microphone Localization from Acoustic Impulse Responses". In: *IEEE Signal Processing Letters* 23.10 (2016), pp. 1459–1463. ISSN: 10709908. DOI: 10.1109/LSP.2016.2601878.

[166] Feng Lu and Evangelos Milios. "Robot Pose Estimation in Unknown Environments by Matching 2D Range Scans". In: *Journal of Intelligent and Robotic Systems: Theory and Applications* 18.3 (1997), pp. 249–275. ISSN: 09210296. DOI: 10.1023/A:1007957421070.

[167] Rob Worley and Sean Anderson. "Topological robot localization in a pipe network". In: *UKRAS20 Conference: "Robots into the real world" Proceedings* (2020), pp. 59–60.

[168] Rob Worley and Sean Anderson. "Topological robot localization in a large-scale pipe network". In: *Towards Autonomous Robotic Systems, TAROS 2020* 1 (2020). DOI: 10.31256/zw1wq5m.

[169] Rob Worley and Sean Anderson. "Robust Efficient Localization of Robots in Pipe Networks using a Particle Filter for Hybrid Metric-Topological Space". In: *2021 European Conference on Mobile Robots (ECMR)* (2021).

[170] Rob Worley and Sean Anderson. "Robot Localization in a Pipe Network using a Particle Filter with Error Detection and Recovery in a Hybrid Metric-Topological Space". In: *2021 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)* (2021), pp. 1–8. DOI: 10.1109/mfi52462.2021.9591168.

[171] Tae Bum Kwon et al. "Efficiency improvement in Monte Carlo localization through topological information". In: *IEEE International Conference on Intelligent Robots and Systems* October (2006), pp. 424–429. DOI: 10.1109/IROS.2006.281962. URL: https://ieeexplore.ieee.org/document/4059089.

[172] Tae Bum Kwon, Ju Ho Yang, and Jae Bok Song. "Efficient and reliable Monte Carlo localization with thinning edges". In: *International Journal of Control, Automation and*

Systems 8.2 (2010), pp. 328–338. ISSN: 15986446. DOI: 10.1007/s12555-010-0219-3. URL: https://link.springer.com/article/10.1007/s12555-010-0219-3.