Robot Mapping and Localisation in Water Pipes

A thesis submitted to the University of Sheffield for the degree of Doctor of Philosophy

Ke Ma

Department of Automatic Control and Systems Engineering

August 2018

To my family

Acknowledgements

First of all, a massive thanks to my supervisor Sean R. Anderson for his kind help, strong support, and loads of great comments and suggestions during the course of this thesis.

I would like to say a big thank you to Richard P. Collins who has helped me a lot during the course of study and the preparation for viva.

Also a big thank you to Tony J. Dodd who has shared loads of valuable comments and suggestions on editing paper publications and also the final thesis.

Many thanks to Gavin Sailor for his great efforts in experiments.

Finally I would like to give my sincere thanks to my family – father and mother, without whom I am not who I am.

Abstract

The demand for inspection and repair technologies for the water industries on their water mains and distribution pipes is increasing. In urban water distribution systems, due to the fact that water pipes are ageing, pipe leakages and serious damage may occur. Compared with the cost of pipe replacement in the underground distribution system, regular pipe inspection and repair is more cost effective to water companies and local communities. However, small-diameter pipes are not accessible to humans because they are small in size and often buried underground. Therefore, inspection robotic systems are more suited to this task in terms of underground pipe networks mapping and damage localisation, in order to target repair from above ground.

There are a number of challenges for robot mapping and localisation in water pipes, which are: 1) feature sparsity in water pipes – lack of navigation landmarks, 2) in-pipe robot can only detect nearby features, and 3) unlike indoor/outdoor SLAM problems, in-pipe robot has less movement flexibility. The main aim of this thesis is to solve these challenges and address the problem of robot mapping and localisation in small-diameter feature-sparse water pipes.

This thesis presents a number of novel contributions. Firstly, for the front end, where raw sensor data is transformed into signals useful for mapping and localisation algorithms, new types of maps are developed here for water pipes: for plastic pipes, ultrasound data is used to map the ground profile through the plastic pipe wall, whilst for metal pipes a hydrophone is used to determine a map based on pipe vibration amplitude over space. Secondly, a new sequential approach to mapping and localisation is developed, based on alignment of multiple maps based on dynamic time warping averaging. Thirdly, a new simultaneous localisation and mapping algorithm is developed, which overcomes the limitation of the sequential approach in that the map is not updated. Finally, a new sensor fusion algorithm is developed that transforms the robot location in the local coordinate frame to the world coordinate frame, which would be essential for targeting repairs from above ground.

Contents

List of Figures

List of Tables

Symbols and Abbreviations

A list of the variables and notation used in this thesis is defined below. The definitions and conventions set here will be observed throughout unless otherwise stated.

Abbreviations

EKF	Extended Kalman filter
KF	Kalman filter
NRMSE	Normalised rooted mean square error
PF	Particle filter
RBPF	Rao-Blackwellised particle filter
SLAM	Simultaneous localisation and mapping

General Notations

$\mathbb{R}, \mathbb{X}, \mathbb{Y}$	Spaces
$\mathbf{A}, \mathbf{F}, \mathbf{G}$	Matrices
$\mathbf{a}, \mathbf{f}, \mathbf{x}, \mathbf{y}$	Vectors
$\mathcal{A}, \mathcal{S}, \mathcal{X}, \mathcal{Y}$	Sets
a, b, c, x, t	Scalars

Notational Conventions

$\mathbf{A}^{-1}$	Matrix inverse
$\mathbf{A}^{T}$	Matrix transpose
$\mathbf{E}[\mathbf{x}]$	Expectation of $\mathbf{x}$

$\mathrm{tr}\mathbf{A}$	Trace of matrix A

Symbols

$\mathbf{F}$	State transition matrix for a time-invariant linear system
$\mathbf{F}_k$	State transition matrix at sample time k for a time-varying linear system
$\mathbf{F}_x$	Jacobian matrix of function $\mathbf{f}(\mathbf{x_k})$, $\mathbf{x}_k \to \mathbf{f}(\mathbf{x}_k)$
$\mathbf{G}$	Control matrix at sample time k for a time-invariant linear system
$\mathbf{G}_k$	Control matrix at sample time k for a time-varying linear system
$\mathbf{H}_k$	Linear observation matrix at time k in the observation model
$\mathbf{H}_x$	Jacobian matrix of function $\mathbf{h}(\mathbf{x_k})$, $\mathbf{x}_k \to \mathbf{h}(\mathbf{x}_k)$
$\hat{\boldsymbol{\theta}}_k$	In SLAM, it is the state estimate posterior describing the map in the state vector at time k
$\mathbf{K}_k$	Kalman gain at time k
$\mathbb{R}$	Space of real numbers
$\mathbb{R}^n$	n-dimensional space model of real numbers
$\mathbf{f}(\cdot)$	Dynamic transition function in a state space model
$\mathbf{h}(\cdot)$	Measurement model function in a state space model
$\mathbf{I}$	Identity matrix
$\mathbf{P}_k^-$	Prediction of the covariance of the Gaussian distribution of a Kalman filter at time step k before measurement $\mathbf{z}_k$
$\mathbf{Q}_k$	Covariance matrix of the process model noise at the jump from time step k to $k+1$
$\mathbf{R}_k$	Covariance matrix of the measurement model noise at the jump from time step k to $k+1$
$\mathbf{u}_k$	Control input vector in state-space model at step k
$\mathbf{v}_k$	Measurement noise at step k
$\mathbf{w}_k$	State noise at step k

$\mathcal{L}(\cdot)$	Likelihood function
$\mu_k^{(i)}$	In FastSLAM, it is the mean of the ith feature at time k
$\omega_k^{(i)}$	The weight of ith particle at time k in a particle filter
$\mathbf{P}_k$	Error covariance estimate of the Gaussian distribution at time k
$\boldsymbol{\theta}$	In EKF-SLAM and PipeSLAM, it is the state variable describing the map in the state vector
$\Sigma_k^{(i)}$	In FastSLAM, it is the error covariance of the ith feature at time k
$\mathbf{x}_k^-$	A prior state estimate at time k with no knowledge of current observation $\mathbf{y}_k$
$\mathbf{x}_k$	State vector in state-space model at step k
$\hat{\mathbf{x}}_k^{(i)-}$	A prior state estimate of ith particle (or ensemble member) at time k with no knowledge of current observation $\mathbf{y}_k$
$\hat{\mathbf{x}}_k^{(i)}$	A posterior state estimate of ith particle (or ensemble member) at time k with no knowledge of current observation $\mathbf{y}_k$
$\hat{\mathbf{x}}_k$	A posterior state estimate at time k with knowledge of current observation $\mathbf{y}_k$
k	Time step number
$N(\cdot)$	Gaussian distribution (i.e., normal distribution)
n_s	Number of samples in a particle filter
n_x	State dimension
n_y	Observation dimension
n_{eff}	Efficient number of particle samples in a particle filter
q	Number of ensemble members in EnKF

Chapter 1

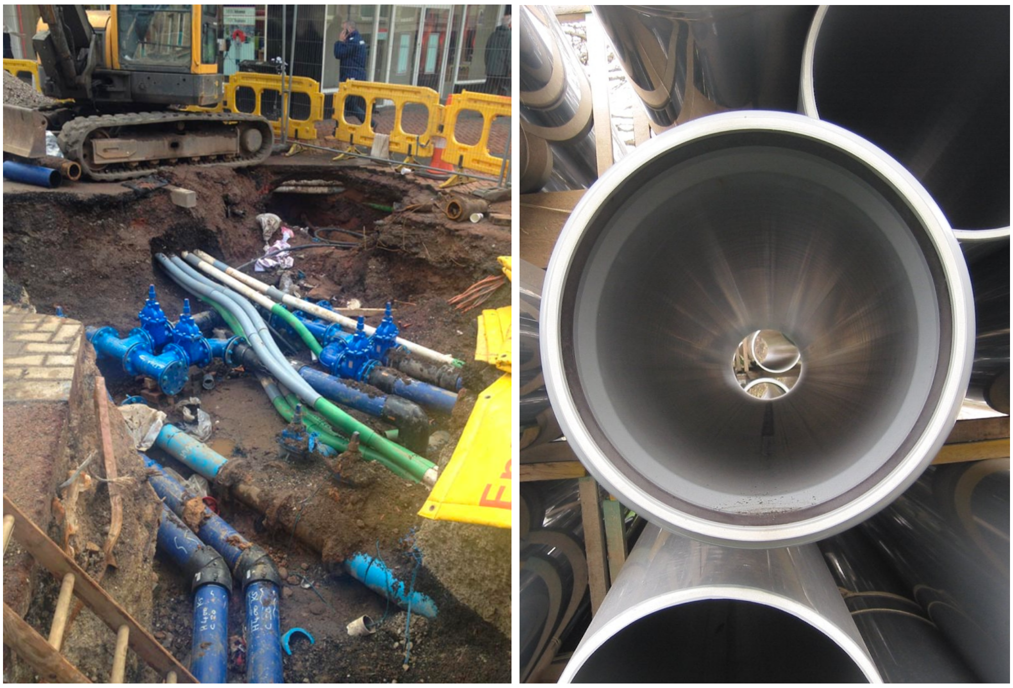
Introduction

1.1 Background

Water is of great importance and is a precious resource for human beings. In developed countries, almost all cities are equipped with complete and sophisticated water supply and purification systems that transform and transfer rain, snow and groundwater to drinking-standard water. In most cases, water supply infrastructure contains various types of components that are used to pump, divert, store, purify and deliver drinking water [121]. Raw (untreated) water is collected in water collection points, such as lakes, rivers, artificial reservoirs and groundwater. This is transferred to purification facilities via uncovered above-ground aqueducts or tunnels. After a number of purification processes, the drinking-standard water is transferred into the underground or above-ground tanks and water towers in cities. Water is then transmitted to private houses or industrial, commercial or institution customers by using underground water pipe networks.

In water distribution systems, because pipe materials are ageing, leakages and damage may happen [121]. These will lead to bacteria infiltration and waste of water. To prevent these, one can use an entirely new set of water distribution system facilities that contains the latest technology, or alternatively, do periodical inspections and regular maintenance before damage occurs. In terms of expense, inspections and maintenance are more acceptable and widely used in the water industry.

The cost of replacing pipes in an underground water distribution system is relatively high [28]. Fig.1.1a shows an example of digging the ground for pipe replacement in an urban area. Therefore regular tests and maintenance are usually preferred from a commercial perspective. In addition, the water distribution system in a city is mostly viewed as a complete system. Damage in one section



(a) (b)

Figure 1.1: (a) Water pipe replacement in urban area [63]. (b) Internal view of a plastic pipe demonstrating the lack of features for mapping and localisation [120].

of the network may have a large negative impact in the community or even result in a wider area being deprived of service. A dramatic example is the famous New York City breakdown in 1998 where a 48-inch, 128 year old water main pipe burst on Fifth Avenue. Several streets were flooded and a 35-foot-wide crater was created.

A less costly way to maintain water distribution, compared to pipe replacement, is more careful maintenance before pipe damage occurs. Therefore, recently researchers and institutions have put more effort into this field [102]. The water industry has spent a large amount of money on early failure detection in their underground networks (water, gas, oil and its derivatives) each year. Successful early failure detection can largely prevent wastage of water, reduce the risk of pollution and improve the stability of the service.

An important problem in water pipe inspection is that many pipelines are inaccessible to humans because they are located underground and some are even placed at the bottom of oceans. This increases the difficulty and cost of inspection around the pipe. Therefore, robotic systems have great potential for inspecting these inaccessible pipelines [102]. Whilst there are many techniques for robot pipe inspection itself, an as yet unsolved problem is accurately locating damage in pipes once found. Further to this, water companies often do not precisely know the location of their pipes. Therefore repairs are difficult to accurately target. This leads to the specification of a key challenge for water pipe inspection: the mapping of pipe networks and localisation of damage using robot navigation algorithms. The focus of this project is therefore the localisation and mapping problem for water pipes using mobile robotic systems.

There are three major challenges for robot navigation in water pipes. The first is the feature sparsity in water pipes, see Fig.1.1b for an inside vision of a pipe. Most current robot navigation systems deal with indoor and outdoor environments, which contain numerous landmark features. However, it is dark in water pipes and pipe walls are smooth and of the same colour in a section of pipeline, which makes it difficult for camera-based methods to distinguish visual patterns and for other sensors (e.g. range sensors) to capture features. Secondly, in pipes the robot can only detect features that are nearby using standard sensors such as laser range finders and ultrasound sensors. This makes it even more difficult for sensors to capture landmark features for navigation. Thirdly, unlike indoor or outdoor navigation, the in-pipe robot has a very restricted route (either moving forward or backward) in the pipe. By contrast, in an indoor environment, the navigation robot can move flexibly and detect landmark features from different positions and angles, reducing uncertainty in the estimate of the map and robot location. However in water pipes, this flexible movement around landmarks is not possible. Therefore, robot navigation in water distribution pipes is a difficult and as yet unsolved problem.

1.2 Aim and Objectives

The aim of this thesis is to build a robot mapping and localisation system for the underground water pipe environment.

The aim is achievable through several objectives:

1. Explore the challenges of water pipe inspection and robot navigation in water distribution pipes. Investigate the buried pipe infrastructure; such as pipe wall materials, pipe surroundings (voids between pipe wall and soil) etc., to find observable features useful for navigation and determine corresponding types of sensors to observe those features.

2. Investigate existing robot navigation algorithms and determine methods for testing them against the needs of the problem as defined in objective 1.

3. Develop and implement a specialised robot navigation algorithm for water

pipe networks by modification and improvement of the most promising algorithms determined in objective 2.

4. Verify and validate the robot navigation algorithm with tests undertaken in controlled laboratory environments using a prototype pipe inspection robot.

1.3 Structure of the Thesis

The thesis is structured as follows.

Chapter 1 gives background and introduction to the water pipe mapping and localisation problem and outlines published contributions arising from this thesis.

Chapter 2 provides a literature review of key areas of relevance, such as robotic systems for water pipe inspection and mapping, simultaneous mapping and localisation (SLAM), and particularly SLAM for water pipes.

Chapter 3 presents the first novel contribution, where new types of robot navigation map are developed for both plastic and metal water pipes. For plastic pipes, the technique is based on using ultrasonic sensing, where the reflections off the soil exterior to the pipe wall are used to construct a terrain-profile map. For metal pipes, the technique is based on hydrophone sensing, where the sound waves excite pipe vibration, which creates a unique profile along the length of the pipe that can be used as a map. Elements of this chapter are published in [96, 98].

Chapter 4 presents a novel sequential approach to mapping and localisation, where a map is first constructed then used for localisation. The map is constructed by a robot making multiple passes up and down a length of pipe - the map from each pass is then averaged using dynamic time warping. The localisation is performed using a robust version of the EKF, previously developed in the aerospace domain for localising off a map of the ground terrain profile [65]. Elements of this chapter are published in [98].

Chapter 5 presents a novel PipeSLAM algorithm, using the Rao-Blackwellised particle filter. The PipeSLAM algorithm addresses the limitation of the sequential approach from chapter 4, which is that the map is not updated in that scheme. Elements of this chapter are published in [97].

Chapter 6 presents a novel method for localising pipes in the world coordinate frame by using heading estimates from an inertial measurement unit (IMU), fused with the PipeSLAM and sequential algorithms from chapters 4 and 5. The limitation of the algorithms from chapters 4 and 5 is that they only estimate distance travelled along the pipe. The IMU gives an estimate of heading, so along with distance travelled the robot location, and therefore the pipe location, can be transformed to the world coordinate frame. However, both the heading and distance travelled estimates incorporate uncertainty, which must be propagated through the nonlinear transformation from the local to the world coordinate frame. This is done using Monte Carlo sampling, which is both simple and effective.

Chapter 7 provides conclusions and directions for future work.

1.4 Contributions Arising from this Thesis

Novelties arising from this thesis are: a novel type of map for navigating in plastic water pipes, where the terrain profile external to the pipe wall is measured by an ultrasonic sensor (see Chapter 3); a novel type of map for navigating in metal water pipes, where a hydrophone is used to induce pipe vibration (see Chapter 3); a novel sequential method for mapping and localisation in water pipes, where map calibration for a set of independent maps is performed using an existing algorithm based on averaged dynamic time warping [116] (see Chapter 4); a novel in-pipe SLAM algorithm – *PipeSLAM*, where a Rao-Blackwellised particle filter based SLAM algorithm is developed for water pipes using the novel types of map described above (see Chapter 5).

During the thesis writing, four papers have been peer reviewed and published in international publications, and are listed below:

1. Ke Ma, Juanjuan Zhu, Tony J. Dodd, Richard Collins, and Sean R. Anderson. "Robot mapping and localisation for feature sparse water pipes using voids as landmarks." In *Conference Towards Autonomous Robotic Systems*, pp. 161-166. Springer, 2015.

   Winner of the IET Robotics and Mechatronics TPN prize for most promising industry technology.

2. Ke Ma, Ali Hassan-Zahraee, Juanjuan Zhu, R. Mills, Joby Boxall, Rob Dwyer-Joyce, Tony Dodd, Richard Collins, and Sean R. Anderson. "Robotic Mapping and Localisation in Feature Sparse Water Pipes." In *19th World Conference on Non-Destructive Testing. German Society for Non-Destructive Testing*, 2016.

3. Ke Ma, Michele M. Schirru, Ali Hassan Zahraee, Rob Dwyer-Joyce, Joby Boxall, Tony J. Dodd, Richard Collins, and Sean R. Anderson. "Robot mapping and localisation in metal water pipes using hydrophone induced vibration and map alignment by dynamic time warping." In *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2548-2553. IEEE, 2017.

4. Ke Ma, Michele Schirru, Ali Hassan Zahraee, Rob Dwyer-Joyce, Joby Boxall, Tony J. Dodd, Richard Collins, and Sean R. Anderson. "PipeSLAM: Simultaneous localisation and mapping in feature sparse water pipes using the Rao-Blackwellised particle filter." In *IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*, pp. 1459-1464. IEEE, 2017.

Chapter 2

Literature Review

2.1 Introduction

In many countries in Europe including the U.K., a great number of cities are still equipped with aged water infrastructure. These water pipes buried underground are often inaccessible for humans and many pipes were buried decades ago. However, mostly because of the high cost in renewing pipes, water companies are not willing to replace these out-of-date pipes with new pipes that contain modern material technologies. Hence, for these aged facilities, additional care and monitoring are essential. For instance, these aged water pipes need inspection more often for leakages and other potential damages using modern inspection technologies such as robotic systems.

This chapter will firstly give an introduction to water infrastructure and water distribution systems, including the latest development of pipe inspection techniques. Secondly, because this research is focused on developing a solution to pipe inspection by using an in-pipe robot, a brief review of the robot simultaneous localisation and mapping (SLAM) techniques and historical development will be given. Thirdly, the robot navigation or SLAM problem can be viewed as probabilistic state estimation. Therefore, probabilistic state estimation methods will be described. Fourthly, a review of the recent developments in robot SLAM in water pipes will be given. Finally, a short summary of this chapter will be given.

2.2 Water Pipes and Inspection

2.2.1 Water Distribution Systems

In a typical water distribution system network, pipes are categorised into two types by their functionality [146]:

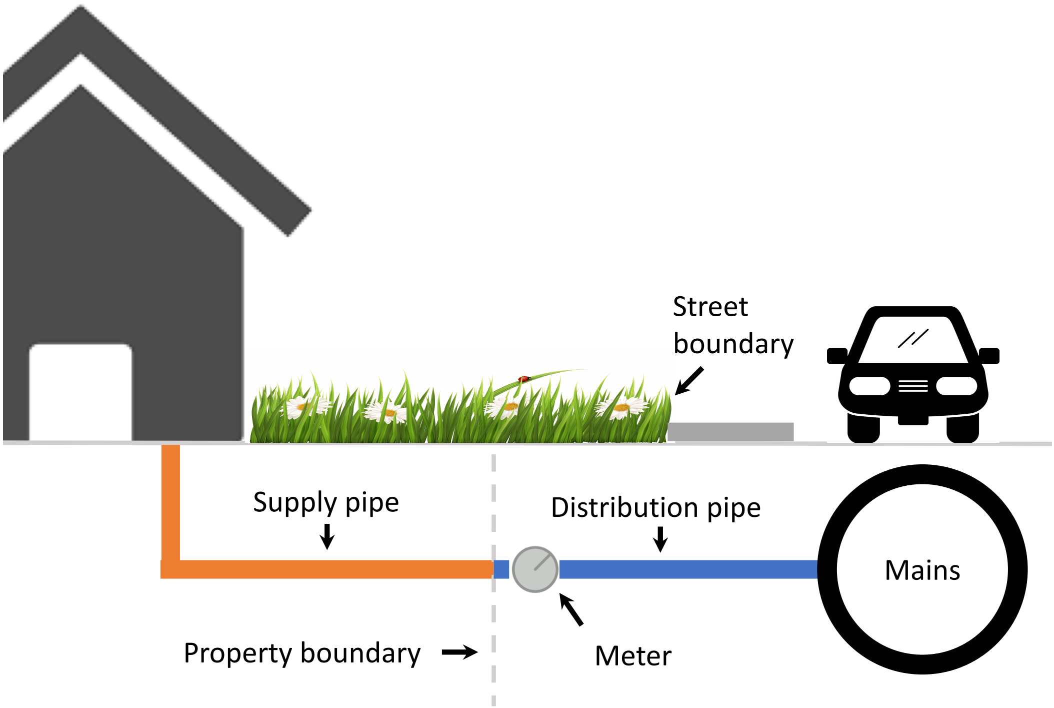


Figure 2.1: Illustration diagram of trunk mains and distribution pipes in water distribution system

.

Trunk mains are usually of large size (typically greater than 18 inches in diameter) that deliver a large amount of water from one area to another. For instance, clean water is firstly delivered from purification facilities, through large-sized trunk mains to water tanks and pumping stations in the cities and towns. Local citizens normally do not get their tap water from water trunk mains, but from smaller sized water distribution pipes.

Distribution pipes, also called communication pipes, transfer water from trunk mains to local communities and business buildings. The diameters of distribution pipes are normally smaller than water trunk mains (typically about 3 inches), but are also determined by the usage of the consumers. For instance, business consumers normally require larger sized pipes compared with home users. Also, fire hydrants on the streets are attached to nearby distribution pipes rather than trunk mains. Fig.2.1 shows a general layout of trunk mains, distribution pipes and supply pipes.

Supply pipes are also called service pipes. Normally, it is a short section of small diameter pipe (about 1 inch), that connects a building to the distribution pipes. In some exceptional cases, a supply pipe could be connected directly to a water trunk main.

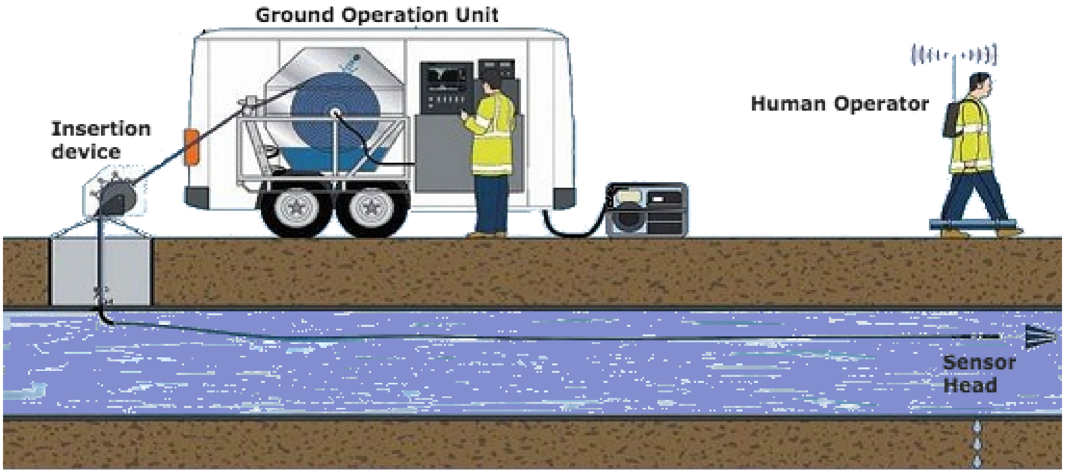


Figure 2.2: Schematic functioning of Sahara system [101]. The system sends a sensor head into a pipe tethered with a cable driven by a motor above the ground. The human operator above the surface tracks the sensor head with a detector.

It is worthwhile to mention that a pipe normally uses the same material across its length and it normally has no distinguishable landmarks or features for robot navigation.

2.2.2 Water Pipe Inspection

In the past decade, much research and development has been put into pipe inspections, however, pipe robots are still in their infancy, with few implementations that can freely travel through typical underground water pipe systems.

In 1998, Roth and colleagues [126] provided a short review on the state-of-art in pipe inspection technology in the late 20th century. That work is focused on inspection technology that is targeted to waste water pipes. They found that many in-pipe inspection devices were tethered and tele-operated from a control station above the ground. Although a fully autonomous pipe inspection robot was the goal, the main emphasis at that time, and even in recent years [92], was the inspection and damage detection systems, rather than damage localisation, which requires robot localisation and mapping.

The Sahara leakage detection system (Fig.2.2) sends a sensor platform into a pipe tethered with a motor above the ground [102]. The surface human operator tracks the sensor platform using a detecting device. This inspection technology is effective in pipelines of diameter larger than 0.3m.

Apart from in-pipe inspection, pipe inspection can also be done from the above ground level by using ground penetrating radar (GPR) [29]. The principle behind

GPR is that it transmits electro-magnetic signal pulses to the ground [29]. When the signal reaches the sub-surface (i.e. the soil-pipe surface), the signal reflects back to a receiver within the GPR. Based on the reflected signal, the location of the sub-surface can be estimated. Recent research has found that such technology is not only able to detect the location of buried containers [154] but can also locate leakages [67, 109].

Safe, reliable water systems are significant to urban communities. There are over 2 million kilometres of water pipes across Europe, of which more than 50 thousand are distribution pipes feeding water distribution systems [102]. Water pipes are very sensitive and expensive components in the water industry as they cannot be regularly replaced or shut down for maintenance, which may disrupt the entire service in a certain area. However, most underground water pipes were built decades ago, therefore, there is an increasing need for efficient and non-disruptive inspection techniques. Hence, the water industry needs fully autonomous in-pipe robots to do regular inspection (i.e. early detection of damage for repair) before significant and disruptive failure can occur.

2.2.3 Inspection Robotics

Since the first in-pipe inspection devices were designed in the 1970s, many robot prototypes of different mechanisms have been proposed. A first review on different robot platforms for water pipe locomotion from the application point of view was given by [130]. Generally, for large-diameter horizontal pipes such as water trunk mains, wheel based robot platforms are most suitable. In terms of small-diameter pipes or those pipes where wheel based platform are not applicable, other types might be suitable, such as legged [16, 110], or snake-like robots [131, 149]. Fig.2.3 shows six different types of in-pipe robot mechanisms that are commonly used in underground water pipe scenarios.

In 1998, a small-sized water pipe inspection robot tailored for 13mm in-diameter water pipe was developed [137]. The robot has a micro arm and an onboard CCD camera for leakage detection using image processing techniques. But this design lacks flexibility and is not capable of smooth motion through sharply curved pipes, such as pipe junctions. In the early 2000s, the mobility restriction was overcome by Muramatsu and Roh [107, 124, 125]. Another image processing based algorithm was published, which used a CCD camera and laser diode to acquire in-pipe images [44, 45]. In 2003, a tele-operation system for pipe inspection was developed [13].

These techniques are capable of robot navigation tasks in the water pipe environment. But due to the darkness in the water pipe environment and the feature

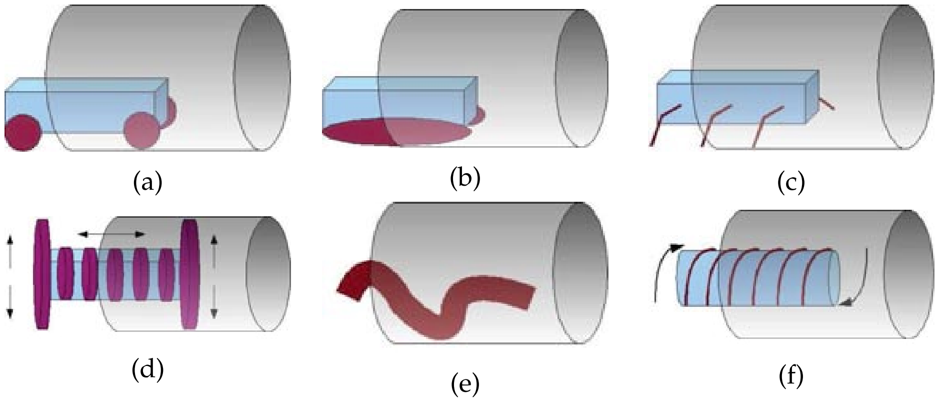


Figure 2.3: Different mechanical types of in-pipe robots [102]. These types are: (a) wheeled, (b) tracks, (c) legged, (d) inchworm, (e) snake, (f) screw.

sparsity of the inner pipe wall (Fig.1.1b), the robustness of such visual based navigation techniques is still in question. Although image processing techniques have been investigated for a while and become relatively mature, such techniques may not be the optimal solution of pipe navigation where pipes lack visual landmarks. Additional sensing technology or navigation methods should be used that can be fused with visual sensing techniques.

2.3 Robot Simultaneous Localisation and Mapping (SLAM) Problem Formulation

In many robot applications, GPS location is not available. For those robots that are used in underground pipes, oceans and other hazardous environments, other techniques need to be developed to solve the navigation problem. Therefore, in such cases, the robot itself needs to build a map of the unknown environment and concurrently localise itself in real-time. A simultaneous localisation and mapping (SLAM) problem asks if it is possible for a mobile robot to simultaneously track its location while incrementally building a consistent map of an unknown environment at the same time. A solution to the problem has been seen as a milestone in the field of robotics as it is the starting point for robots to be truly autonomous [46].

2.3.1 Standard Problem Formulation

The current standard SLAM problem formulation mainly consists of two processes (see Fig.2.4):

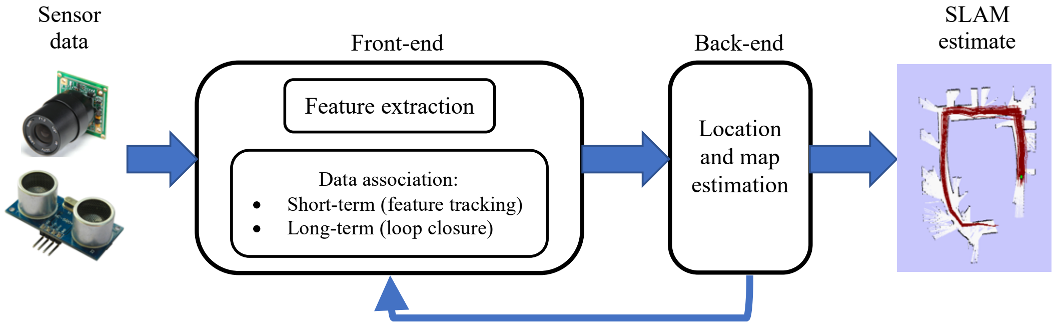


Figure 2.4: Front-end and back-end in a typical SLAM problem

1. **Front-end** process is the sensor data acquisition and filtering process to produce readable data for later back-end process.

2. **Back-end** processes pre-filtered data and uses a suitable SLAM algorithm to process those data to simultaneously estimate robot pose and the map of the environment.

Front-end process: data association and sensor fusion

In real world robotic navigation applications, it is not practical to directly write sensor observations as a mathematical function of robot pose and map parameters. For example, in vision-based SLAM applications (such as [49]), the sensor observations are usually a sequence of image clips captured from a visual camera. Raw images, as the input data, cannot be directly used to drive the SLAM process. The same difficulties are found in other sensors, such as distance measurement-based SLAM, e.g. using lidar sensing [83]. Therefore, the front-end process is to convert or extract information from raw data that can be used later in the back-end SLAM process.

In a typical visual-SLAM system, camera images may contain a large amount of information (pixels). For navigation purposes, not all pixels are of interest but a specific visual pattern can be important. Therefore, the front-end process will firstly extract some distinguishable pixels or patterns in the image. The pattern locations can thus be easier for the back-end module to process. Similarly, laser measurement returns the time-of-flight of a laser beam and it will need to be converted to distance (to an obstacle) before entering the back-end process.

In addition, a front-end process also needs to perform data association from multiple sensors – sensor fusion. Many robot platforms are equipped with multiple sensors for navigation purposes and the information gathered from different sensor observations need to be fused to give a better understanding of the surroundings. Those different types of data (e.g. distances, visual images) need to be fused to distinguish a landmark. For instance, imagine a mobile robot exploring through an indoor environment of multiple obstacles and each obstacle is painted in different colours on its surface. To build a map, a robot needs to determine which obstacle it captures by using a visual image and to determine the distance by using a laser scanner measurement. In such a setting, the front-end module produces an output from raw data comprising the distance and colour of that observed obstacle.

Back-end process: robot location and map estimation

The most used problem formulation for SLAM can be traced back to 1997 in a seminal paper [95]. This *de-facto* standard formulation has been widely adopted in many SLAM research papers [36, 51, 61, 76, 113, 138].

In a typical SLAM problem, the state that needs to be estimated is the combination of robot location and map parameters (where typically map parameters are condidered time-invariant). We denote the time history of robot locations, sensor measurements (produced from the front end process) and control inputs as

$$\mathbf{x}_{0:k} = \{\mathbf{x}_0, \mathbf{x}_1, \cdots, \mathbf{x}_k\} \tag{2.1}$$

$$\mathbf{y}_{1:k} = \{\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_k\} \tag{2.2}$$

$$\mathbf{u}_{1:k} = \{\mathbf{u}_1, \mathbf{u}_2, \cdots, \mathbf{u}_k\} \tag{2.3}$$

where $\mathbf{x}_k \in \mathbb{R}^{n_x}$ is the robot location (or pose) at sample time k, $\mathbf{y}_k \in \mathbb{R}^{n_y}$ is the corresponding sensor measurements, and $\mathbf{u}_k \in \mathbb{R}^{n_u}$ is the vector of control inputs.

The key problem in SLAM is to estimate the joint distribution of the robot location, $\mathbf{x}_k$, and the map parameters, $\boldsymbol{\theta}$, from the history of observed outputs, control inputs, and initial state $\mathbf{x}_0$,

$$p(\mathbf{x}_k, \boldsymbol{\theta} | \mathbf{y}_{1:k}, \mathbf{u}_{1:k}, \mathbf{x}_0)$$

The common algorithmic solutions developed for estimating this joint distribution, based on nonlinear state estimation methods, are described in Section 2.4.

2.3.2 The Evolution of the SLAM Problem

The development and evolution of simultaneous localisation and mapping (SLAM) can be broadly classified into three main time periods that are the *classical age* (1986-2004), the *algorithmic-analysis age* (2004-2015), and the *robust-perception age*

(2015-now), according to a recent review on robot SLAM [20]. Sections below have highlighted some key milestones from each of these periods to show an evolution of SLAM over these time periods.

The classical age (1986-2004)

Modern probabilistic SLAM development can be traced back to the IEEE Robotics and Automation Conference in 1986, California. Since then, probabilistic approaches have been widely adopted in robotic research and artificial intelligence [46]. This historical conference came to a consensus that in the area of robot mapping and localisation, the most fundamental problem that needed to be addressed was the consistent probabilistic mapping [20].

Several years later after this conference, several key papers [48, 135] found a mathematical model that could be used to describe the correlation between landmark locations and robot motion uncertainties. Results also showed that the correlation would increase over an accumulated number of successive measurements.

In the meantime, visual based localisation [4] and sonar based localisation [26, 30] found that there may exist many common characteristics between these two approaches. This had soon been proved by Smith et al. [134], who found that the uncertainties of landmark estimations were correlated with each other. This is because landmark locations are captured and estimated based on the same robot poses, which contain the same errors. The conclusion was significant – the estimation of both localisation and mapping requires a joint state vector consisting of both robot poses and landmark locations.

Gradually, researchers realised that the mapping and localisation problems could be described as a single state estimation problem. On the other hand, to address the mapping problem, the most important part was to find the correlation among landmarks. At the 1995 International Symposium on Robotics Research, the abbreviation SLAM appeared in the robotics community and was presented in a mobile robot paper [47]. At the time, it was also called *concurrent mapping and localisation*, abbreviated as CML, which was equivalent to SLAM. Soon after, a significant improvement was made by Csorba [31, 32], which used a bounded region filter (BRF) that largely reduced the storage requirement to a linear function of N.

Since then, researchers have been focused on improving computational efficiency and solving problems in data association. The International Symposium on Robotics Research in 1999 is another milestone for SLAM in the robotics community. In the symposium, the first SLAM session was held and Thrun et al. [140] presented their formulation of probabilistic SLAM. Then, in 2000, the SLAM workshop at the IEEE International Conference on Robotics and Automation (ICRA) attracted 15 SLAM researchers. Just two years later, the ICRA 2002 workshop attracted over 150 robotic researchers in the SLAM area.

Algorithmic-analysis age (2004-2015)

The second period in SLAM is the *algorithmic-analysis age*. In this period, some problems existed in the fundamental properties of SLAM, such as convergence and observability, which were solved [37]. The paper [37] gives a review on the recent state-of-the-art research mainly on feature based SLAM. In the *algorithmic-analysis age*, SLAM became a much wider subject which integrated many research fields, including computer vision, signal processing, systems identification, sensor fusion, graph theory, optimisation and probability theory [20].

In the past decade, with the fast development in digital memory technology, it became possible to allow complex algorithms to solve large scale SLAM problems while retaining computational efficiency in practical robot applications. In addition, many algorithms based on the extended information filter (EIF) were developed to overcome computational complexity by exploiting sparsity in maps [52, 150, 151]. But the inconsistency issue was not solved (e.g. loop closure – recognising locations that have already been visited).

Robustness can be one of the most important factors for robots achieving full autonomy. A SLAM solution can result in failure mainly due to two reasons: algorithmic failure and hardware-related failure.

The reason for hardware-related failure can be straight forward: no hardware component can be guaranteed to operate correctly at all times, e.g. large cumulative drift errors may occur such as drift in long-term trajectory estimation from an inertial measurement unit (IMU) [112].

For algorithmic failures, incorrect data association can be one of the main reasons for failure. For instance, the data association in the front-end module needs to transform raw sensor measurements to a readable dataset for the back-end module. For instance, in a visual-based SLAM [133], if the front-end module extracts wrong pixels (possibly due to unavoidable noise in the image), incorrect features will be recognised in the image and therefore produce wrong states for the back-end module, which will result in poor estimation.

Many contributions have been made in feature based SLAM problems in the *Algorithmic-analysis age*. To implement high autonomy in mobile robot applications, one important thing is that a robot itself needs to determine whether it has travelled back to a known location which it has passed before. This is usually called loop closure in the field [5]. Due to unavoidable drift in robot pose estima-

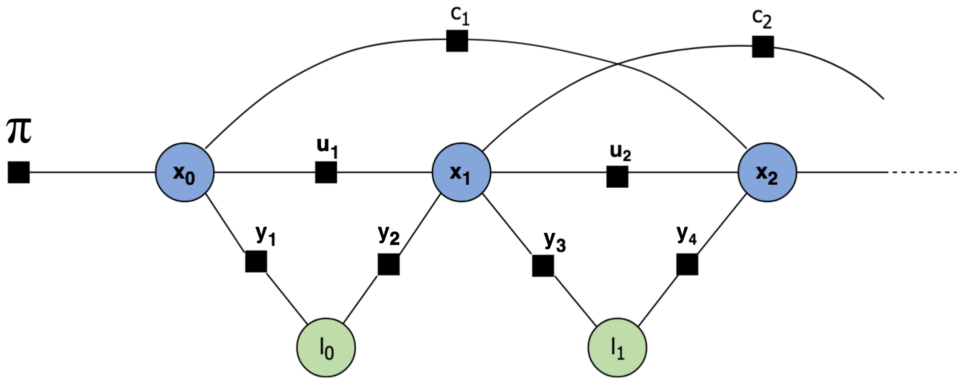


Figure 2.5: SLAM as a factor graph. It gives an insightful visualisation of a SLAM problem. $\boldsymbol{\pi}$ at the very front denotes the initial assumptions and prior knowledge. Blue circles indicate the robot poses $(\mathbf{x}_0, \mathbf{x}_1, \cdots)$. The green circles denote sparse landmarks $(l_0, l_1, \cdots)$. In addition, $(\mathbf{u}_1, \mathbf{u}_2, \cdots)$ marks the control input or robot motion constraints and $(\mathbf{y}_1, \mathbf{y}_2, \cdots)$ indicate the actual sensor measurement of landmarks from each robot pose. Loop closure is abbreviated and marked in $(c_1, c_2, \cdots)$, although it usually happens after a long time when robot travels back to a known position of the map.

tion, the robot needs to move to a known location in the environment in order to close the loop and correct accumulated drift. Therefore, loop closure in a SLAM problem is key to achieve robustness. However, it can be difficult to achieve in some scenarios. Since the real world environment is usually dynamic and landmarks are not static in positions, e.g., moving people, flashing lights. A common assumption in most SLAM problem settings is that the environment is unchanged (static) with no moving objects. Various approaches have been made to correctly determine a loop closure using different SLAM methods and in different scenarios, such as visual place recognition [94] and feature extraction in laser-based SLAM method [144].

Scalability is another important aspect that concerns the size of the environment that needs to be explored. Many modern SLAM algorithms that have been developed so far are mainly targeting indoor environments [54, 55], which are small scaled and bounded. However, real world robot applications include larger scale environment explorations, such as ocean exploration, large scale terrain scanning, agriculture farming and water pipeline inspection. In such scenarios, Thrun and Montemerlo [139] used a method called Graph SLAM to solve the problem by using sparse information by generating a factor graph (see Fig.2.5). All these applications may lead to unbounded state variables to represent the environment as more and more map landmarks are discovered. Because robots in such environments need to continuously travel through new places, observing new landmarks, computational memory usage will be unbounded. In practice, typical SLAM algorithms have a quadratic growth in memory usage [5]. To tackle this problem, an iterative linear solver was introduced, which can reduce the quadratic complexity to linear, which means the memory consumption grows linearly in the number of variables [35]. Among various optimisation approaches, there are three interesting methods to reduce the computational cost:

- *Node and edge sparsification* – this cluster of approaches simply reduces the number of new nodes and edges adding into the graph in graphSLAM [139] in order to solve the scalability problem. Typical methods include the information-theoretic approach [69], inducing new constraints between existing nodes [73], information based criterion graph optimisation [84], generic linear constraint [21] and nonlinear graph sparsification [100]. Also, in order to reduce the number of variables, the discrete robot pose can be replaced by a continuous trajectory by using B-splines [53]. This technique has been used and improved in many SLAM algorithms and applications [1, 10, 12, 43, 79, 105, 145].

- *Parallel SLAM* – uses multiple processors, such as a graphic processing unit (GPU) to simultaneously estimate SLAM variables. In factor graph, the idea is to separate a large graph into several small sub-graphs and each sub-graph is then stored and processed in a single processor. The first attempt was by Bosse et al. [11] followed by sub-graph optimisation methods [111, 155], and hierarchy of sub-maps [62]. In addition, some approaches divide mapping and localisation into two independent computing cores and estimate them in parallel [81], or separate other parts of the SLAM process and run them in parallel [132, 136, 152].

- *Swarm robots SLAM* – another way to reduce computational cost is to use a multirobot system. Individual robots cooperatively travel through a large area and share knowledge with each other simultaneously. This leads to two knowledge sharing structures: 1) a centralised structure – each robot builds its own map and uploads the environment information to a central station which merges all sub-maps and distributes integrated information to individual robots [39, 122]; and 2) a decentralised structure – assume no central station is available and individual robots have to build a consensus and draw a common map [2, 34, 82, 87]. Saeedi et al. [127] give a more detailed review on robot SLAM for multirobot systems.

Metric mapping uses metric map models to represent the environment. Using a suitable map representation is crucial in solving SLAM problem. Generally speaking, metric map models include feature-based maps and occupancy grid-based maps. In 2-dimensional space, assuming landmarks are distinguishable, the former models the environment as a set of sparse landmarks with specific features and latter models the environment as a matrix of cells with the probability of occupation of individual cells. *IEEE RAS Map Data Representation Working Group* has lately released a standard representation of these 2-dimensional maps in robotic navigation [68]. Many current SLAM methods are using feature-based approaches, which models the environment as a set of sparse landmarks with specific features (e.g., points, lines, planes) [106].

Semantic mapping is another way of modelling the environment, which associates semantic elements to geometric properties in the environment. Due to the gradually realised limitations of maps that are merely geometric, semantic mapping has drawn increased attention recently in order to enhance robot robustness in complex environment settings and facilitate autonomous tasks, for instance, SLAM in indoor office environment using semantic objects [15] and human-robot interaction [7, 19, 128]. Except for a few methods discovered so far, the development of semantic mapping is still at its basic level.

Active SLAM contrasts to passive robot SLAM applications where robots perform SLAM using sensor measurements that were not deliberately obtained for the purpose of localisation and mapping. In order to deliver better and more accurate SLAM results, a robot needs to deliberately control its motion to collect the necessary data that can minimise the uncertainty in the localisation and mapping process. This type of SLAM is usually called *active SLAM*. The definition first originated from [6] and has further been explored in [142]. Thrun [143] stated that *active SLAM* includes the dilemma of whether to explore new locations (extend the map) in the environment or to revisit explored positions (reduce mapping and localisation uncertainty). Therefore, the *active SLAM* problem can be viewed as a decision making problem which has several solution frameworks that can be applied to SLAM. One of the well-known decision making frameworks is the theory of optimal experimental design [119]. Several research papers [23, 24] have applied this theory to *active SLAM*, to guide the robot in selecting its next possible action based on the uncertainties in map estimation. Model predictive control has also been applied to *active SLAM* [89, 90]. One popular framework used in *active SLAM* decomposes the problem into three main processes:

- *Select vantage points* – ideally, a robot performing *active SLAM* should evaluate its motion at every time step but the computational cost grows exponentially and is impractical in real-world applications [17, 99]. In practical *active SLAM* algorithms, a small area of the environment is selected, and frontier-based methods are used to explore the space [78, 153]. Recent approaches using a continuous-space domain representation have solved the convergence and optimisation problem in a local area [70, 147, 148].

- *Calculate utility of an action* – when planning an action, a robot needs to consider the robot pose and the map along with future possible sensor measurements and future actions as well. Most of the current works use a linear combination of uncertainties in robot pose and the map [14, 22]. Since the numerical values indicating robot pose errors are considerably lower than the errors of the map (due to the increased space exploration), these two values are not comparable and often need to be tuned manually. But recent approaches partially address the problem in particle filter SLAM [22] and pose graph optimisation [25].

- *Execute an action or stop exploration* – executing an action is often the simplest part in the *active SLAM* algorithm as motion planing has various techniques to support this. However, determining whether or not the exploration is accomplished remains unsolved.

2.4 Probabilistic State Estimation Methods for Robot Navigation

The SLAM problem is usually regarded as state estimation (of the robot pose and map parameters), therefore, state estimation methods have been extensively used to solve robot navigation problems [46]. SLAM state estimation is typically viewed in a probabilistic framework, using recursive Bayesian state estimation methods based on sensor measurements and the system model.

This section first describes probabilistic state estimation algorithms that have been developed in the literature, including the Kalman filter [77], which is an optimal state estimator for linear Gaussian systems that has been widely described in the literature [9, 64, 74]. State estimators for nonlinear systems are also described, including the extended Kalman filter (EKF) [64, 123], the ensemble Kalman filter [18, 50, 57, 66] and the particle filter [58].

The section finishes with a description of two of the most popular state estimation algorithms for SLAM: EKF-SLAM [38] and FastSLAM (which is based on the particle filter) [141].

2.4.1 Kalman Filter

The Kalman filter [77] is a well-known and well-used algorithm in the field of state estimation, for linear, possibly time-varying, Gaussian systems described by the state-space model

$$\mathbf{x}_k = \mathbf{F}_{k-1}\mathbf{x}_{k-1} + \mathbf{G}_{k-1}\mathbf{u}_{k-1} + \mathbf{w}_{k-1} \tag{2.4}$$

$$\mathbf{y}_k = \mathbf{H}_k\mathbf{x}_k + \mathbf{v}_k \tag{2.5}$$

where $\mathbf{x}_k \in \mathbb{R}^{n_x}$ is the state vector, $\mathbf{u}_k \in \mathbb{R}^{n_u}$ is the control input, $\mathbf{F}_k$ is the state transition matrix, $\mathbf{G}_k$ is the control matrix, $\mathbf{w}_k \sim N(\mathbf{0}, \mathbf{Q}_k)$ is Gaussian noise, $\mathbf{y}_k \in \mathbb{R}^{n_y}$ is the measurement, $\mathbf{H}_k$ is the measurement matrix and $\mathbf{v}_k \sim N(\mathbf{0}, \mathbf{R}_k)$ is Gaussian measurement noise.

The Kalman filter can be used to recursively calculate the posterior distribution of the state $\mathbf{x}_k$ at each time step k, given the history of the measurements, $\mathbf{y}_{1:k}$, i.e.

$$p(\mathbf{x}_k|\mathbf{y}_{1:k})$$

Using Bayes theorem, assuming conditional independence of measurements, and recalling the Markov property of the state-space model, then $p(\mathbf{x}_k|\mathbf{y}_{1:k})$ can be expressed as [129],

$$\begin{aligned} p(\mathbf{x}_k|\mathbf{y}_{1:k}) = p(\mathbf{x}_k|\mathbf{y}_k, \mathbf{y}_{1:k-1}) &\propto p(\mathbf{y}_k|\mathbf{x}_k, \mathbf{y}_{1:k-1})p(\mathbf{x}_k|\mathbf{y}_{1:k-1}) \\ &\propto p(\mathbf{y}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{y}_{1:k-1}) \end{aligned} \tag{2.6}$$

where from (2.5),

$$p(\mathbf{y}_k|\mathbf{x}_k) = N(\mathbf{y}_k|\mathbf{H}_k\mathbf{x}_k, \mathbf{R}_k) \tag{2.7}$$

and $p(\mathbf{x}_k|\mathbf{y}_{1:k-1})$ is obtained from the Chapman-Kolmogorov equation [129], which defines the prediction step of the Kalman filter,

$$p(\mathbf{x}_k|\mathbf{y}_{1:k-1}) = \int p(\mathbf{x}_k|\mathbf{x}_{k-1})p(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1})d\mathbf{x}_{k-1} \tag{2.8}$$

where from (2.4),

$$p(\mathbf{x}_k|\mathbf{x}_{k-1}) = N(\mathbf{x}_k|\mathbf{F}_{k-1}\mathbf{x}_{k-1} + \mathbf{G}_{k-1}\mathbf{u}_{k-1}, \mathbf{Q}_{k-1}) \tag{2.9}$$

and where the distribution of the state at the previous time step, $p(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1})$, is assumed known, with mean $\hat{\mathbf{x}}_{k-1}$ and covariance $\mathbf{P}_{k-1}$, i.e.

$$p(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1}) = N(\mathbf{x}_{k-1}|\hat{\mathbf{x}}_{k-1}, \mathbf{P}_{k-1}) \tag{2.10}$$

Therefore, by substitution of (2.9) and (2.10) in (2.8),

$$p(\mathbf{x}_k|\mathbf{y}_{1:k-1}) = \int N(\mathbf{x}_k|\mathbf{F}_{k-1}\mathbf{x}_{k-1} + \mathbf{G}_{k-1}\mathbf{u}_{k-1}, \mathbf{Q}_{k-1})N(\mathbf{x}_{k-1}|\hat{\mathbf{x}}_{k-1}, \mathbf{P}_{k-1})d\mathbf{x}_{k-1} \tag{2.11}$$

where the result of the marginalisation over $\mathbf{x}_{k-1}$ is a Gaussian distribution [129],

$$\begin{aligned} p(\mathbf{x}_k|\mathbf{y}_{1:k-1}) &= N(\mathbf{x}_k|\mathbf{F}_{k-1}\hat{\mathbf{x}}_{k-1} + \mathbf{G}_{k-1}\mathbf{u}_{k-1}, \mathbf{F}_{k-1}\mathbf{P}_{k-1}\mathbf{F}_{k-1}^T + \mathbf{Q}_{k-1}) \\ &= N(\mathbf{x}_k|\mathbf{x}_k^-, \mathbf{P}_k^-) \end{aligned} \tag{2.12}$$

To obtain the posterior distribution of the state, $p(\mathbf{x}_k|\mathbf{y}_{1:k})$, first note that the term on the right hand side of (2.6) can be expressed as the joint distribution

$$\begin{aligned} p(\mathbf{x}_k, \mathbf{y}_k|\mathbf{y}_{1:k-1}) &= p(\mathbf{y}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{y}_{1:k-1}) \\ &= N(\mathbf{y}_k|\mathbf{H}_k\mathbf{x}_k, \mathbf{R}_k)N(\mathbf{x}_k|\mathbf{x}_k^-, \mathbf{P}_k^-) \end{aligned} \tag{2.13}$$

which using Lemma A.1, can be expressed as

$$p(\mathbf{x}_k, \mathbf{y}_k|\mathbf{y}_{1:k-1}) = N\left(\begin{bmatrix}\mathbf{x}_k \\ \mathbf{y}_k\end{bmatrix} \middle| \begin{bmatrix}\mathbf{x}_k^- \\ \mathbf{H}\mathbf{x}_k^-\end{bmatrix}, \begin{bmatrix}\mathbf{P}_k^- & \mathbf{P}_k^-\mathbf{H}_k^T \\ \mathbf{H}_k\mathbf{P}_k^- & \mathbf{H}_k\mathbf{P}_k^-\mathbf{H}_k^T + \mathbf{R}_k\end{bmatrix}\right) \tag{2.14}$$

Finally, the mean $\hat{\mathbf{x}}_k$, and covariance $\mathbf{P}_k$, of the conditional distribution

$$\begin{aligned} p(\mathbf{x}_k|\mathbf{y}_k, \mathbf{y}_{1:k-1}) &= p(\mathbf{x}_k|\mathbf{y}_{1:k}) \\ &= N\left(\mathbf{x}_k|\hat{\mathbf{x}}_k, \mathbf{P}_k\right) \end{aligned} \tag{2.15}$$

can be obtained from (2.14) and Lemma A.2, such that

$$\hat{\mathbf{x}}_k = \mathbf{x}_k^- + \mathbf{P}_k^-\mathbf{H}_k^T(\mathbf{H}_k\mathbf{P}_k^-\mathbf{H}_k^T + \mathbf{R}_k)^{-1}(\mathbf{y}_k - \mathbf{H}_k\mathbf{x}_k^-) \tag{2.16}$$

$$\mathbf{P}_k = \mathbf{P}_k^- - \mathbf{P}_k^-\mathbf{H}_k^T(\mathbf{H}_k\mathbf{P}_k^-\mathbf{H}_k^T + \mathbf{R}_k)^{-1}\mathbf{H}_k\mathbf{P}_k^- \tag{2.17}$$

Hence, the well known Kalman filter recursions for estimating the state vector can now be stated in their usual form, using (2.12), (2.16) and (2.17) as

1. **The prediction step**

$$\mathbf{x}_k^- = \mathbf{F}_{k-1}\hat{\mathbf{x}}_{k-1} + \mathbf{G}_{k-1}\mathbf{u}_{k-1} \tag{2.18}$$

$$\mathbf{P}_k^- = \mathbf{F}_{k-1}\mathbf{P}_{k-1}\mathbf{F}_{k-1}^T + \mathbf{Q}_{k-1} \tag{2.19}$$

2. **The update step**

$$\mathbf{S}_k = \mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k \tag{2.20}$$

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}_k^T \mathbf{S}_k^{-1} \tag{2.21}$$

$$\hat{\mathbf{x}}_k = \mathbf{x}_k^- + \mathbf{K}_k \left(\mathbf{y}_k - \mathbf{H}_k \mathbf{x}_k^- \right) \tag{2.22}$$

$$\mathbf{P}_k = \mathbf{P}_k^- - \mathbf{K}_k \mathbf{H}_k \mathbf{P}_k^- \tag{2.23}$$

Note that the recursion starts from a prior mean $\hat{\mathbf{x}}_0$ and covariance $\mathbf{P}_0$.

2.4.2 Extended Kalman Filter

The extended Kalman filter (EKF) is used to estimate the states of nonlinear systems that are subject to Gaussian process and measurement noise [9, 123]. This is different to the Kalman filter, which can only address linear Gaussian problems. The key difference is that the EKF propagates a Gaussian state estimate by linearising the system for all the updates of the covariance matrices (which preserves Gaussianity). In other regards, the EKF resembles the Kalman filter, so they are closely related.

A nonlinear state-space system with additive Gaussian process and measurement noise can be represented as,

$$\mathbf{x}_k = \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}) + \mathbf{w}_{k-1} \tag{2.24}$$

$$\mathbf{y}_k = \mathbf{h}(\mathbf{x}_k) + \mathbf{v}_k \tag{2.25}$$

where $\mathbf{x}_k \in \mathbb{R}^{n_x}$ is the state, $\mathbf{y}_k \in \mathbb{R}^{n_y}$ is the measurement, $\mathbf{w}_k \sim N(\mathbf{0}, \mathbf{Q}_{k-1})$ is the Gaussian process noise, $\mathbf{v}_k \sim N(\mathbf{0}, \mathbf{R}_k)$ is the Gaussian measurement noise, $\mathbf{f}(\cdot)$ is the dynamic model function, and $\mathbf{h}(\cdot)$ is the measurement model function. The functions $\mathbf{f}(\cdot)$ and $\mathbf{h}(\cdot)$ can also depend on the time step k, but for the convenience of notation, this dependency is omitted.

The idea of the EKF is to assume Gaussian approximations to the filtering densities,

$$p(\mathbf{x}_k | \mathbf{y}_{1:k}) \simeq N(\mathbf{x}_k | \hat{\mathbf{x}}_k, \mathbf{P}_k) \tag{2.26}$$

In the EKF, these approximations are made by using Taylor series approximations to the non-linearities.

Therefore, the steps for the EKF algorithm can be expressed as

1. **Prediction step**

$$\mathbf{x}_k^- = \mathbf{f}(\hat{\mathbf{x}}_{k-1}, \mathbf{u}_{k-1}) \tag{2.27}$$

$$\mathbf{P}_k^- = \mathbf{F}_x \mathbf{P}_{k-1} \mathbf{F}_x^T + \mathbf{Q}_{k-1} \tag{2.28}$$

2. **Measurement update step**

$$\mathbf{S}_k = \mathbf{H}_x \mathbf{P}_k^- \mathbf{H}_x^T + \mathbf{R}_k \tag{2.29}$$

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}_x^T \mathbf{S}_k^{-1} \tag{2.30}$$

$$\hat{\mathbf{x}}_k = \mathbf{x}_k^- + \mathbf{K}_k(\mathbf{y}_k - \mathbf{h}(\mathbf{x}_k^-)) \tag{2.31}$$

$$\mathbf{P}_k = \mathbf{P}_k^- - \mathbf{K}_k \mathbf{H}_x \mathbf{P}_k^- \tag{2.32}$$

where $\mathbf{F}_x$ and $\mathbf{H}_x$ are Jacobian matrices whose elements at row i and column j are given by

$$[\mathbf{F}_x]_{i,j} = \left.\frac{\partial \mathbf{f}_i(\mathbf{x}, \mathbf{u})}{\partial \mathbf{x}_j}\right|_{\mathbf{x}=\hat{\mathbf{x}}_{k-1}, \mathbf{u}=\mathbf{u}_{k-1}} \tag{2.33}$$

$$[\mathbf{H}_x]_{i,j} = \left.\frac{\partial \mathbf{h}_i(\mathbf{x})}{\partial \mathbf{x}_j}\right|_{\mathbf{x}=\mathbf{x}_k^-} \tag{2.34}$$

2.4.3 Ensemble Kalman Filter

The Ensemble Kalman Filter (EnKF) is similar to the Kalman filter but where the state covariance is represented by a sample covariance matrix derived from an ensemble of state vector samples [18, 50, 57, 66]. The EnKF is particularly suited to state estimation in high dimensional systems where it is impractical to work with algebraic operations on the Kalman filter state covariance matrix. Therefore, it would appear that the EnKF is suited to the SLAM problem because of the potential large dimensionality of the state vector in large maps, although this application has not been studied to date.

Again, we assume the same nonlinear system dynamics shown in (2.24) and (2.25). To formulate the EnKF, firstly, at time step k, assume an ensemble of q state estimates with stochastic errors, and the ensemble set is written as $\hat{\mathcal{X}}_k^- \in \mathbb{R}^{n_x \times q}$, and

$$\mathcal{X}_k^- \triangleq \{\mathbf{x}_k^{(1)-}, \cdots, \mathbf{x}_k^{(q)-}\} \tag{2.35}$$

where $\mathbf{x}_k^{(i)-} \in \mathbb{R}^{n_x}$, $i = 1, ..., q$ is the predicted state estimate of ith ensemble member propagated through state equation (shown in (2.24)) as below

$$\mathbf{x}_k^{(i)-} \triangleq \mathbf{f}(\hat{\mathbf{x}}_{k-1}^{(i)}, \mathbf{u}_{k-1}) + \mathbf{w}_{k-1}^{(i)} \tag{2.36}$$

and the estimation mean $\mathbf{x}_k^-$ of the ensemble set is calculated by

$$\mathbf{x}_k^- \triangleq \frac{1}{q}\sum_{i=1}^{q} \mathbf{x}_k^{(i)-} \tag{2.37}$$

Since the true state vector $\mathbf{x}_k$ is unknown, we can approximate state estimates by using ensemble members. The ensemble error matrix and output error matrix are defined around the mean by

$$\mathbf{E}_{\mathbf{x}_k^-} \triangleq [\mathbf{x}_k^{(1)-} - \mathbf{x}_k^-, \cdots, \mathbf{x}_k^{(q)-} - \mathbf{x}_k^-] \tag{2.38}$$

$$\mathbf{E}_{\mathbf{y}_k^-} \triangleq [\mathbf{y}_k^{(1)-} - \mathbf{y}_k^-, \cdots, \mathbf{y}_k^{(q)-} - \mathbf{y}_k^-] \tag{2.39}$$

where $\mathbf{E}_{\mathbf{x}_k^-} \in \mathbb{R}^{n_x \times q}$ and $\mathbf{E}_{\mathbf{y}_k^-} \in \mathbb{R}^{n_y \times q}$. Correspondingly, the predicted error statistics $\mathbf{P}_{\mathbf{x}_k}^-$, $\mathbf{P}_{\mathbf{xy}_k}^-$ and $\mathbf{P}_{\mathbf{yy}_k}^-$ can be approximated as

$$\mathbf{P}_{\mathbf{x}_k}^- \triangleq \frac{1}{q-1}\mathbf{E}_{\mathbf{x}_k^-}(\mathbf{E}_{\mathbf{x}_k^-})^T \tag{2.40}$$

$$\mathbf{P}_{\mathbf{xy}_k}^- \triangleq \frac{1}{q-1}\mathbf{E}_{\mathbf{x}_k^-}(\mathbf{E}_{\mathbf{y}_k^-})^T \tag{2.41}$$

$$\mathbf{P}_{\mathbf{yy}_k}^- \triangleq \frac{1}{q-1}\mathbf{E}_{\mathbf{y}_k^-}(\mathbf{E}_{\mathbf{y}_k^-})^T \tag{2.42}$$

By using (2.31), a posterior estimate of each sample can be calculated by

$$\hat{\mathbf{x}}_k^{(i)} = \mathbf{x}_k^{(i)-} + \mathbf{K}_k(\mathbf{y}_k^{(i)} - \mathbf{h}(\mathbf{x}_k^{(i)-})) \tag{2.43}$$

where we use the classic Kalman gain expression in Kalman filter to express the $\mathbf{K}_k$ and that is

$$\mathbf{K}_k = \mathbf{P}_{\mathbf{xy}_k}^-(\mathbf{P}_{\mathbf{yy}_k}^-)^{-1} \tag{2.44}$$

and each perturbed observation $\mathbf{y}_k^{(i)}$ is given by

$$\mathbf{y}_k^{(i)} = \mathbf{y}_k + \mathbf{v}_k^{(i)} \tag{2.45}$$

where $\mathbf{v}_k^{(i)}$ is zero-mean Gaussian noise variable that is $\mathbf{v}_k^{(i)} \sim N(0, \mathbf{R}_k)$. Similar to the predicted error matrix $\mathbf{E}_{\mathbf{x}_k^-}$, a posterior error matrix can be written as

$$\mathbf{E}_{\hat{\mathbf{x}}_k} \triangleq [\hat{\mathbf{x}}_k^{(1)} - \hat{\mathbf{x}}_k, \cdots, \hat{\mathbf{x}}_k^{(q)} - \hat{\mathbf{x}}_k] \tag{2.46}$$

and the posterior error covariance matrix is

$$\mathbf{P}_k = \frac{1}{q-1}\mathbf{E}_{\hat{\mathbf{x}}_k}\mathbf{E}_{\hat{\mathbf{x}}_k}^T \tag{2.47}$$

The evaluation of covariance matrix can be approximated by using discrete ensemble members.

Hence, the overall EnKF algorithm steps are as follows:

1. **Analysis Step:**

$$\mathbf{K}_k = \mathbf{P}_{\mathbf{xy}_k}^-(\mathbf{P}_{\mathbf{yy}_k}^-)^{-1} \tag{2.48}$$

$$\hat{\mathbf{x}}_k^{(i)} = \mathbf{x}_k^{(i)-} + \mathbf{K}_k(\mathbf{y}_k + \mathbf{v}_k^{(i)} - \mathbf{h}(\mathbf{x}_k^{(i)-})) \tag{2.49}$$

$$\hat{\mathbf{x}}_k = 1/q\sum_{i=1}^{q}\hat{\mathbf{x}}_k^{(i)} \tag{2.50}$$

$$\mathbf{P}_k = \frac{1}{q-1}\mathbf{E}_{\hat{\mathbf{x}}_k}\mathbf{E}_{\hat{\mathbf{x}}_k}^T \tag{2.51}$$

2. **Prediction Step:**

$$\mathbf{x}_k^{(i)-} = \mathbf{f}(\hat{\mathbf{x}}_{k-1}^{(i)}, \mathbf{u}_{k-1}) + \mathbf{w}_{k-1}^{(i)} \tag{2.52}$$

$$\mathbf{x}_k^- = 1/q\sum_{i=1}^{q}\mathbf{x}_k^{(i)-} \tag{2.53}$$

$$\mathbf{E}_{\mathbf{x}_k^-} = [\mathbf{x}_k^{(i)-} - \mathbf{x}_k^-, \cdots, \mathbf{x}_k^{(q)-} - \mathbf{x}_k^-] \tag{2.54}$$

$$\mathbf{E}_{\mathbf{y}_k^-} = [\mathbf{y}_k^{(i)-} - \mathbf{x}_k^-, \cdots, \mathbf{y}_k^{(q)-} - \mathbf{x}_k^-] \tag{2.55}$$

$$\mathbf{P}_{\mathbf{xy}_k}^- = \frac{1}{q-1}\mathbf{E}_{\mathbf{x}_k^-}(\mathbf{E}_{\mathbf{y}_k^-})^T \tag{2.56}$$

$$\mathbf{P}_{\mathbf{yy}_k}^- = \frac{1}{q-1}\mathbf{E}_{\mathbf{y}_k^-}(\mathbf{E}_{\mathbf{y}_k^-})^T \tag{2.57}$$

2.4.4 Particle Filter

The particle filter was first developed by Gordon [58], and was initially known as the Bootstrap filter. The particle filter randomly generates samples to propagate the distribution of the state estimate, which are weighted based on their likelihood as derived from observed data [27, 40, 59, 60]. The particle filter can be applied to nonlinear non-Gaussian systems, which is a key advantage over the EKF. However, a disadvantage is that the particle filter tends to be computationally expensive compared to the EKF because of the need to propagate many samples.

After each time update in estimation, particle weights will vary. Some particles may have lower and lower weights, and become less important. These particles will tend to zero. Since these particles become less important, the entire particle set becomes less efficient. In order to maintain the number of particles at all time, those particles that produced a 'good estimate' (higher weights) are copied to reproduce new particles and particles with low weight values will be removed. This is called particle resampling, which plays a significant role in particle filtering [3].

Assuming a nonlinear state-space model with additive Gaussian noise, as described above,

$$\mathbf{x}_k = \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}) + \mathbf{w}_{k-1} \tag{2.58}$$

$$\mathbf{y}_k = \mathbf{h}(\mathbf{x}_k) + \mathbf{v}_k \tag{2.59}$$

then the basic particle filter for state estimation (the bootstrap filter, which uses the state equation as the proposal distribution) can be described as follows [58] (at each time-step):

1. First calculate the prediction of the state, using the state equation, for each particle $\mathbf{x}_k^{(i)-}$, for $i = 1, \cdots, n_s$, where n_s is the number of particles,

$$\mathbf{x}_k^{(i)-} = \mathbf{f}(\hat{\mathbf{x}}_{k-1}^{(i)}, \mathbf{u}_{k-1}) + \mathbf{w}_{k-1}^{(i)} \tag{2.60}$$

2. Then evaluate the importance weight for each particle sample and multiply by the previous particle weight (which has been normalised),

$$\tilde{\omega}_k^{(i)} = \omega_{k-1}^{(i)} p(\mathbf{y}_k | \mathbf{x}_k^{(i)-}) \tag{2.61}$$

where $\omega_{k-1}^{(i)}$ is the normalised weight, and $\tilde{\omega}_k^{(i)}$ is the un-normalised weight.

3. Then normalise the updated weights

$$\omega_k^{(i)} = \frac{\tilde{\omega}_k^{(i)}}{\sum_{j=1}^{n_s} \tilde{\omega}_k^{(j)}} \tag{2.62}$$

After this procedure, each particle comprises a state estimate for the current time step and a corresponding weight, which indicates the probability that the actual state is the estimated state.

Over time steps, the particle set will diverge and the majority of particles will have very small weights that tend to zero, whilst a few particles will have relatively large weights. To solve this problem of degeneracy, the particles are regularly resampled (as discussed above). A popular resampling method is called sequential importance resampling [129].

In sequential importance resampling, the following procedure is used:

1. The current weights, $\omega_k^{(i)}$, are interpreted as the probability of obtaining the sample index i in the set of current particles $\{\mathbf{x}_k^{(i)-} | \quad i = 1, \ldots, n_s\}$.

2. A total of n_s particles are then resampled from the set $\{\mathbf{x}_k^{(i)-} | \quad i = 1, \ldots, n_s\}$.

3. The corresponding weights are then set to a constant value, $\omega_k^{(i)} = \frac{1}{n_s}$.

This resampling step ensures that multiple copies are made of particles with high weights, and particles with low weights tend to be removed. In practice, at each time step, the algorithm calculates the effective number of particles,

$$n_{\text{eff}} \approx \frac{1}{\sum\limits_{i=1}^{n_s} (\omega_k^{(i)})^2} \tag{2.63}$$

and if n_{eff} is lower than a given value, then the resampling step is performed.

2.4.5 Comparison and Discussion

The performance of each filtering algorithm presented above depends on the type of problem and to a certain extent the computational device used for implementation. To demonstrate differences in performance of each state estimation algorithm a numerical example is presented in this section

The EKF has good performance on low nonlinearity problems with relatively low computational cost. But when it comes to highly nonlinear problems, the estimation results will be inaccurate. In addition, as it is originated from the Kalman filter, it assumes all uncertainties are Gaussian.

The EnKF is suitable for both linear and nonlinear problems and tends to have good performance on nonlinear problems with relatively low computational cost compared to the particle filter. But if one wants to achieve high accuracy without any consideration of computational restrictions, the particle filter is likely to be the better choice. This is because a particle filter does not linearise the model and when the number of particles is significantly large, the estimation will converge to an optimal estimate. In addition, similar to the EKF, the EnKF assumes all uncertainties are Gaussian.

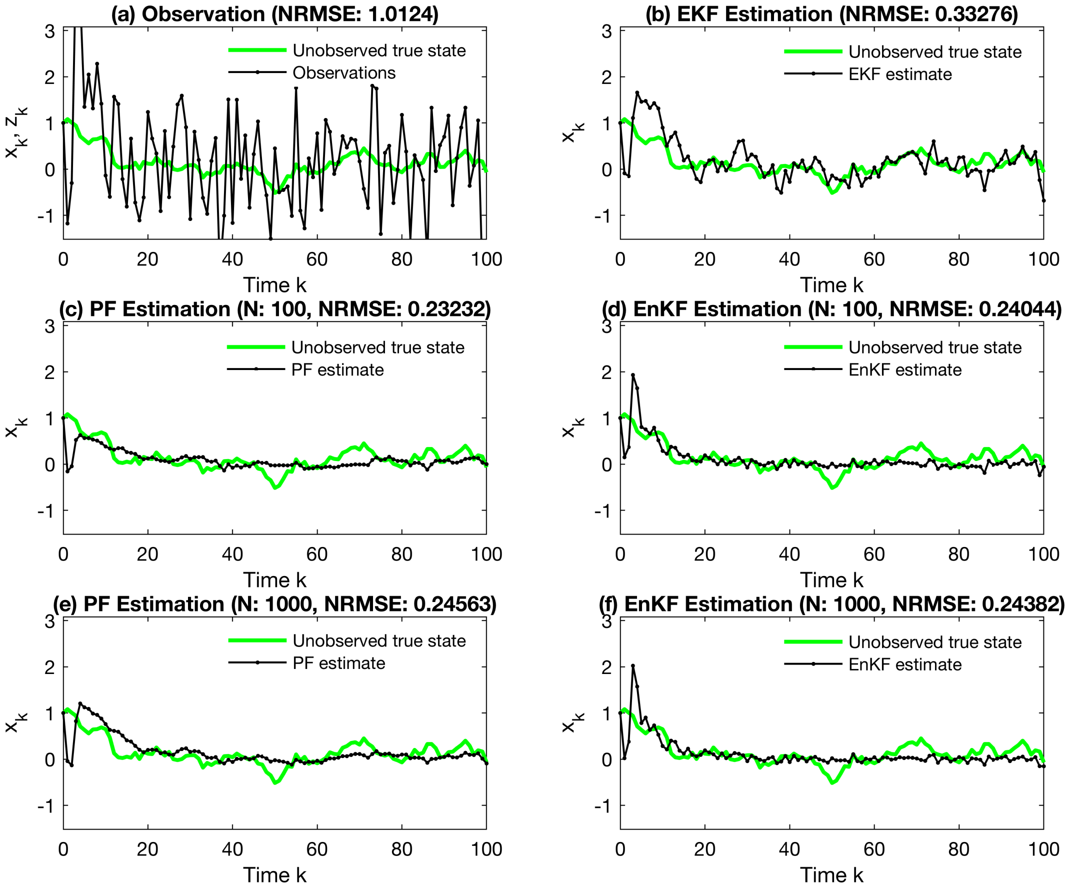


Figure 2.6: Estimation results of a linear model (shown in (2.64) and (2.65)) using EKF, PF, and EnKF methods. Green line in each plot indicates the unobserved true state. (a) noisy observations, (b) state estimation using EKF, (c) state estimation using PF with 100 particles, (d) state estimation using EnKF with a number of 100 ensemble members, (e) state estimation using PF with 1000 particles, (f) state estimation using EnKF with a number of 1000 ensemble members.

The particle filter is able to address both linear and nonlinear problems and has very good performance on problems of very high nonlinearity compared with the EKF and EnKF. The only problem is that the size of the particle set may need to be very large when the state dimension is large, which leads to a dramatically increased computational cost.

To illustrate some of these differences, the algorithms are now compared on some numerical examples. First a linear model is used, so that the optimal state estimate can be used as a benchmark from the Kalman filter. Then a nonlinear system is studied.

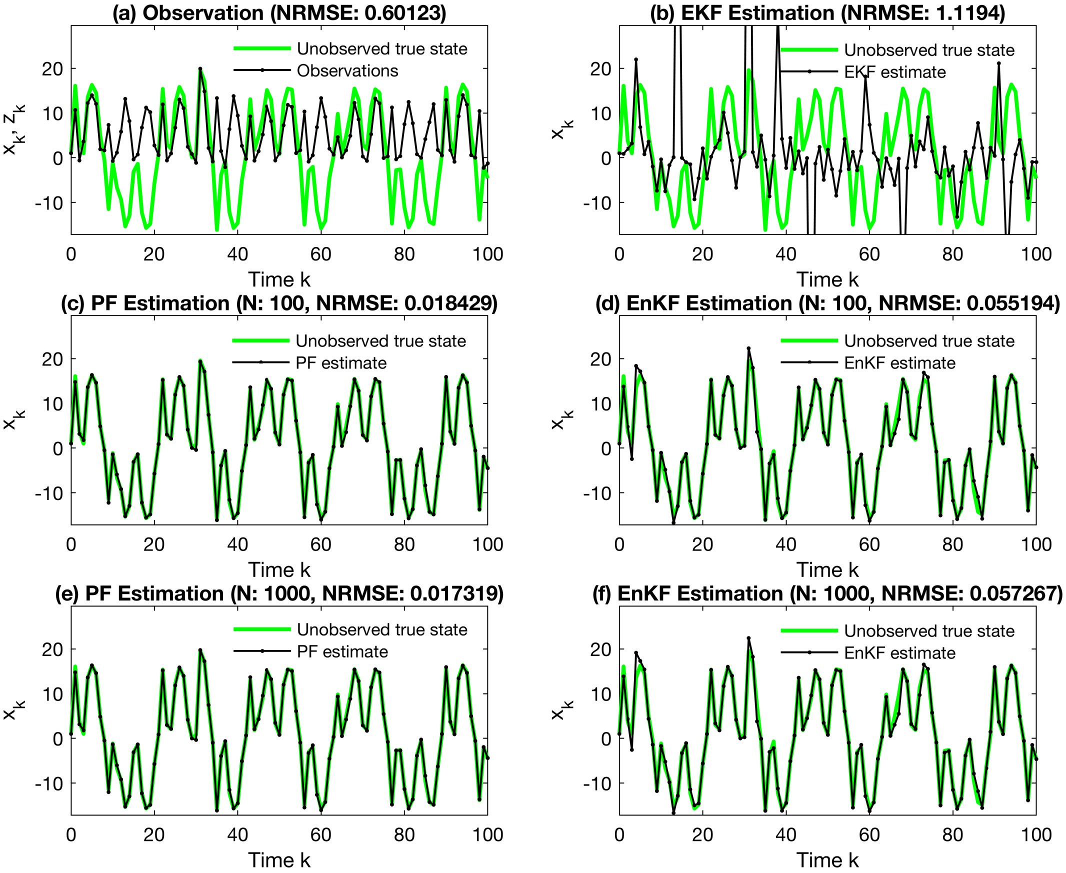


Figure 2.7: Estimation results of a nonlinear model (shown in (2.67) and (2.68)) using EKF, PF, and EnKF methods. Green line in each plot indicates the unobserved true state. (a) noisy observations, (b) state estimation using EKF, (c) state estimation using PF with 100 particles, (d) state estimation using EnKF with a number of 100 ensemble members, (e) state estimation using PF with 1000 particles, (f) state estimation using EnKF with a number of 1000 ensemble members.

Consider the linear system,

$$\mathbf{x}_k = 0.9\mathbf{x}_{k-1} + \mathbf{w}_{k-1} \tag{2.64}$$

$$\mathbf{y}_k = \mathbf{x}_k + \mathbf{v}_k \tag{2.65}$$

with initial condition $\mathbf{x}_0 = 1$, which is simulated over $k = 100$ time steps. The signals $\mathbf{w}_k$ and $\mathbf{v}_k$ are Gaussian noises where $\mathbf{w}_k \sim N(0, 0.1)$ and $\mathbf{v}_k \sim N(0, 1)$.

The estimations using EKF, EnKF and particle filter are compared to the observations in Fig.2.6. To measure the error of the state estimate, normalised rooted mean square error (NRMSE) is used and is

$$NRMSE = \frac{\sqrt{\frac{1}{N}\sum_{k=1}^{N}(\hat{\mathbf{x}}_k - \mathbf{x}_k)^2}}{\mathbf{x}_{max} - x_{min}} \tag{2.66}$$

where $\mathbf{x}_k$ is the true state at time k and $\hat{\mathbf{x}}_k$ is the location estimate. In this linear case, all state estimation algorithms achieve similar performance in terms of NRMSE (Fig.2.6).

For a nonlinear system, which has been analysed before in many papers [58], [80], [3], we have

$$\mathbf{x}_{k+1} = \frac{\mathbf{x}_k}{2} + \frac{25\mathbf{x}_k}{1+\mathbf{x}_k^2} + 8\cos(1.2k) + \mathbf{w}_k \tag{2.67}$$

$$\mathbf{y}_k = \frac{\mathbf{x}_k^2}{20} + \mathbf{v}_k \tag{2.68}$$

with initial assumption $\mathbf{x}_0 = 1$, simulated over 100 time steps, $\mathbf{w}_k$ and $\mathbf{v}_k$ are Gaussian noises that are $\mathbf{w}_k \sim N(0, 0.1)$ and $\mathbf{v}_k \sim N(0, 1)$. The estimations (black lines) are shown in Fig.2.7 along with true states (green lines), from which it is clear in terms of NRMSE and visual inspection that both the EnKF and the particle filter outperform the EKF for this highly nonlinear system.

2.4.6 EKF-SLAM

The basic SLAM problem, as discussed above, is to estimate the joint distribution of the robot location, $\mathbf{x}_k$, and the map parameters, $\boldsymbol{\theta}$, from the history of observed outputs, control inputs, and initial state $\mathbf{x}_0$,

$$p(\mathbf{x}_k, \boldsymbol{\theta} | \mathbf{y}_{1:k}, \mathbf{u}_{1:k}, \mathbf{x}_0)$$

which can be done using the state estimation methods described above. Note that the full state vector here is an augmentation of the robot pose $\mathbf{x}_k$ with the map parameters $\boldsymbol{\theta}$ (where the map parameters are assumed to be time-invariant).

The robot motion model is typically described in a state equation,

$$p(\mathbf{x}_k|\mathbf{x}_{k-1},\mathbf{u}_{k-1}) \Leftrightarrow \mathbf{x}_k = \mathbf{f}(\mathbf{x}_{k-1},\mathbf{u}_{k-1}) + \mathbf{w}_{k-1} \tag{2.69}$$

where $\mathbf{f}(\cdot)$ indicates the motion function or the prediction model, and $\mathbf{w}_k$ is the noise term, which is assumed to be Gaussian. Note that the motion model is not a function of the map parameters, $\boldsymbol{\theta}$.

The measurement model can be described as

$$p(\mathbf{y}_k|\mathbf{x}_k,\boldsymbol{\theta}) \Leftrightarrow \mathbf{y}_k = \mathbf{h}(\mathbf{x}_k,\boldsymbol{\theta}) + \mathbf{v}_k \tag{2.70}$$

where $\mathbf{h}(\cdot)$ indicates the measurement function, and $\mathbf{v}_k$ is the Gaussian noise term. Note that the measurement model is a function of both the robot pose, $\mathbf{x}_k$, and the map parameters $\boldsymbol{\theta}$.

The EKF can be applied directly to this nonlinear state-space model, noting only that the map parameters are omitted from the time update step because the map is assumed to be time-invariant. This results in the well-known EKF-SLAM algorithm [38, 46], which is as follows:

1. **Time update**

$$\mathbf{x}_k^- = \mathbf{f}(\hat{\mathbf{x}}_{k-1},\mathbf{u}_{k-1}) \tag{2.71}$$

$$\boldsymbol{\theta}_k^- = \hat{\boldsymbol{\theta}}_{k-1} \tag{2.72}$$

$$\mathbf{P}_k^- = \mathbf{F}_x\mathbf{P}_{k-1}\mathbf{F}_x^T + \mathbf{Q}_{k-1} \tag{2.73}$$

2. **Measurement update**

$$\mathbf{K}_k = \mathbf{P}_k^-\mathbf{H}_k^T(\mathbf{H}_x\mathbf{P}_k^-\mathbf{H}_x^T + \mathbf{R}_k)^{-1} \tag{2.74}$$

$$\begin{bmatrix}\hat{\mathbf{x}}_k\\ \hat{\boldsymbol{\theta}}_k\end{bmatrix} = \begin{bmatrix}\mathbf{x}_k^-\\ \boldsymbol{\theta}_k^-\end{bmatrix} + \mathbf{K}_k\left[\mathbf{y}_k - \mathbf{h}\left(\begin{bmatrix}\mathbf{x}_k^-\\ \boldsymbol{\theta}_k^-\end{bmatrix}\right)\right] \tag{2.75}$$

$$\mathbf{P}_k = \mathbf{P}_k^- - \mathbf{K}_k\mathbf{H}_x\mathbf{P}_k^- \tag{2.76}$$

2.4.7 FastSLAM

The FastSLAM algorithm [104, 141] is a state estimation method for SLAM that is based on the particle filter, with a simplification known as Rao-Blackwellisation that separates out one set of states (the map parameters, which are high dimensional) from a distinct set of states (the robot pose, which are low dimensional).

Particles are used to represent all possible state trajectories of the robot, and each particle has a unique associated map; each map is updated using an EKF, which lends computational efficiency to the procedure [104, 141]. Without the Rao-Blackwellisation step, the overall state dimension in SLAM, including robot pose and map parameters, would be too large in realistic scenarios to solve using the standard form of particle filter.

The key step in the Rao-Blackwellised particle filter for FastSLAM, is partitioning the joint distribution of robot pose and map parameters, $p(\mathbf{x}_k, \boldsymbol{\theta} | \mathbf{y}_{1:k}, \mathbf{u}_{1:k}, \mathbf{x}_0)$, using the product rule, so that

$$p(\mathbf{x}_k, \boldsymbol{\theta} | \mathbf{y}_{1:k}, \mathbf{u}_{1:k}, \mathbf{x}_0) = p(\boldsymbol{\theta} | \mathbf{x}_k, \mathbf{y}_{1:k}) p(\mathbf{x}_k | \mathbf{y}_{1:k}, \mathbf{u}_{1:k}, \mathbf{x}_0) \tag{2.77}$$

The key point to note is that after application of the product rule, the high-dimensional map parameters $p(\boldsymbol{\theta} | \mathbf{x}_k, \mathbf{y}_{1:k})$ can be updated separately to the low-dimensional robot pose, $p(\mathbf{x}_k | \mathbf{y}_{1:k}, \mathbf{u}_{1:k}, \mathbf{x}_0)$. Therefore, the robot pose can be estimated using a particle filter with a relatively small number of particles, whilst the typically large number of map parameters can be updated using EKFs. This tends to work well in practice because the robot localisation problem tends to be more severely nonlinear than the map update problem [46, 104, 141].

Localisation using the particle filter

For the ith particle, the robot pose estimate $\mathbf{x}_k^{(i)}$ can be described as

$$\mathbf{x}_k^{(i)} \sim p(\mathbf{x}_k | \mathbf{u}_k, \mathbf{x}_{k-1}^{(i)}) \tag{2.78}$$

and the corresponding weight or the so-called importance factor $\omega_k^{(i)}$ of that particle is calculated using the same weights assignment as in the particle filter algorithm, defined in (2.61) and (2.62).

Mapping using the EKF

Unlike the localisation part, the mapping part uses the EKF instead of the particle filter. Since this map update is conditioned on the robot pose, the EKF step is repeated for each individual particle. Hence, the overall estimation posterior in FastSLAM includes the robot pose and landmark locations. The particle set can be represented as

$$\mathcal{X}_k = \{\mathbf{x}_k^{(i)}, \boldsymbol{\mu}_{k,1}^{(i)}, \boldsymbol{\Sigma}_{k,1}^{(i)}, \cdots, \boldsymbol{\mu}_{k,m}^{(i)}, \boldsymbol{\Sigma}_{k,m}^{(i)}\} \tag{2.79}$$

where $\boldsymbol{\mu}_{k,m}^{(i)}$ and $\boldsymbol{\Sigma}_{k,m}^{(i)}$ are the mean and covariance of matrices indicating the mth map parameter in particle i at time k, obtained from each EKF. The FastSLAM algorithm is described in Algorithm 2.1.

Algorithm 2.1 FastSLAM

```
for $i = 1, \cdots, n_s$ do                                    ▷ explore all particles
    Retrieve a robot pose $\mathbf{x}_{k-1}^{(i)}$ from the particle set $\mathcal{X}_k$.
    Update the pose $\mathbf{x}_k^{(i)} \sim p(\mathbf{x}_k|\mathbf{u}_k, \mathbf{x}_{k-1}^{(i)})$.          ▷ sampling
    Update $\boldsymbol{\mu}_{k,1:m}^{(i)}$ and $\boldsymbol{\Sigma}_{k,1:m}^{(i)}$ with $\mathbf{y}_k$ using the EKF.   ▷ Measurement update
    Calculate weight $\tilde{\omega}_k^{(i)}$                  ▷ using likelihood
end for
Normalise weights $\omega_k^{(i)} = \frac{\tilde{\omega}_k^{(i)}}{\sum_{j=1}^{n_s} \tilde{\omega}_k^{(j)}}$
Resample if needed
```

2.5 Robot Simultaneous Localisation and Mapping in Water Pipes

Simultaneous localisation and mapping has been a popular approach to mobile robot navigation for many years. Recently, various new technologies and applications have been presented, in the field of in-door, out-door and underwater environments [5, 46]. However, methods for using SLAM in pipes are still quite new. There have been very few attempts to tackle the problem of robot navigation in pipes in general, and especially in water distribution pipes. Some researchers have implemented SLAM on in-pipe robots that employ visual cameras [85], or inertial measurement units (IMUs) [91].

Many SLAM algorithms employ cameras as they are inexpensive and image processing techniques are quite mature. However, in the pipe environment, there are relatively few visual landmarks and it is dark, and in water filled pipes the images can be noisy. Thus, it is not easy to use visual cameras for navigation in water pipes. IMUs and gyros, although subject to drift, are suitable for obtaining direction in short-length pipes. But the drawbacks are apparent: their estimation uncertainties will increase as the robots travels further and further. Hence, they have to be corrected with other sensors in the long term.

This section describes the main research that has been done in SLAM for robots in water pipes.

Cameras + IMU

An early research paper from Krys and Najjaran [85] used a visual SLAM system equipped with a set of 4 grey-scale CCD cameras, an IMU and a laser range finder (see Fig.2.8). That SLAM system was only tested in a controlled laboratory

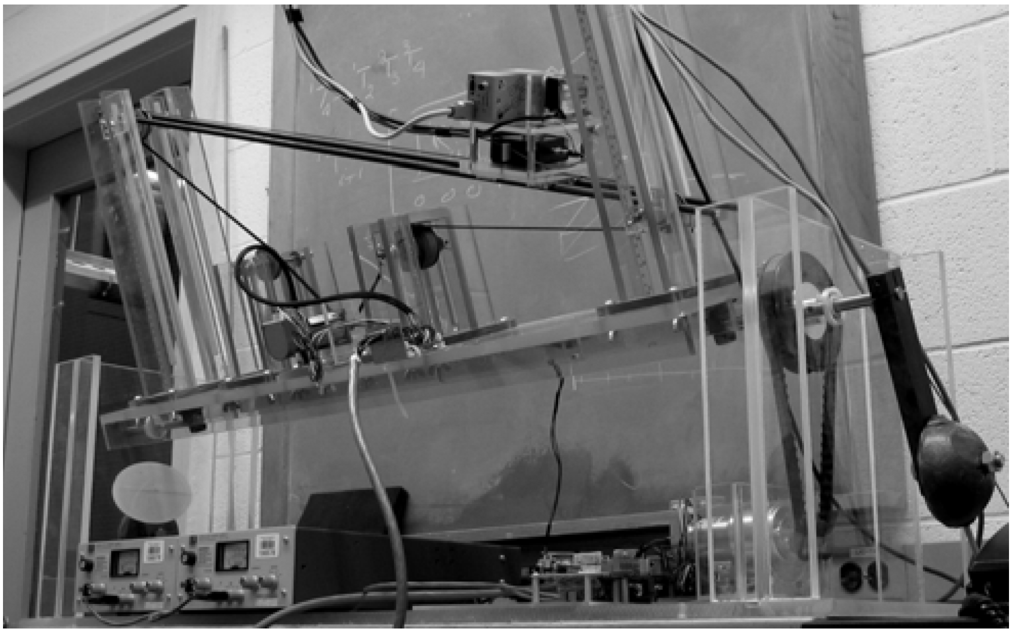


Figure 2.8: Experiment setup in laboratory environment in Krys and Najjaran [85]

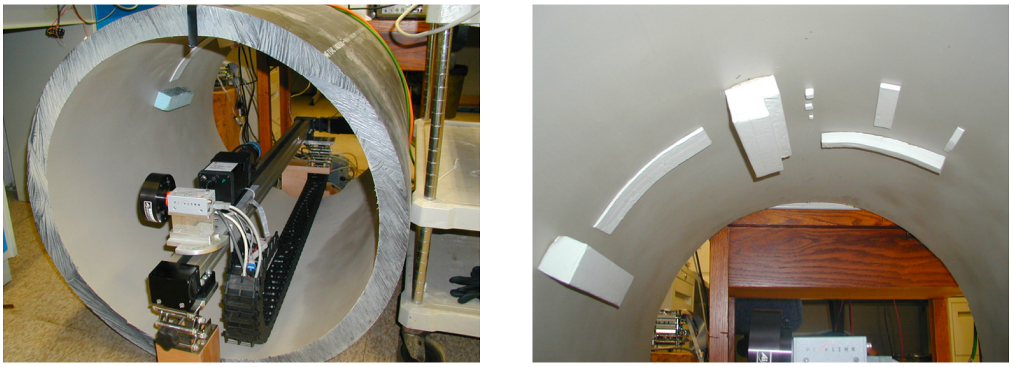
environment with small scale pipe line settings (1.22m in length), and the system was targeted for large in-service water mains.

The 4 spinning CCD cameras capture 360^o images around the sensor carrier in the cross section of the pipe. A laser range finder is used to find the distance between the CCD camera and the image of interest. A gyroscope in the IMU gives direction information and an accelerometer provides dead reckoning distance travelled along the pipe. Due to the well known accelerometer accumulation error, the CCD cameras are used to correct the drift.

Note that this SLAM system platform contains 4 cameras and therefore is targeting large water mains and will not be capable of inspection in small scale water pipes. Most importantly, the system is tested in a transparent pipe in a laboratory environment, which allows external cameras to capture ground truth of the robot movement. However, due to the transparency of the pipe, the light conditions are not realistic, so the testing is not suitable for real-world trunk mains, which are dark. Another limitation is that, in a real-world scenario, the dead reckoning sensor – the accelerometer – can drift rapidly in a large water main full of flowing water, which has not been tested.

Laser + Camera

A recent research paper from Liu and Krys [93] built up a prototype of sensor carrier for internal pipe inspection (see Fig.2.9). The pipe is made of concrete and



(a) (b)

Figure 2.9: Experiment setup in laboratory environment in Liu and Krys [93]: (a) experiment setup, (b) artificial features on the internal pipe wall.

is 82.55 cm in diameter. A laser range finder and an inspection camera are used to perform in-pipe navigation and inspection.

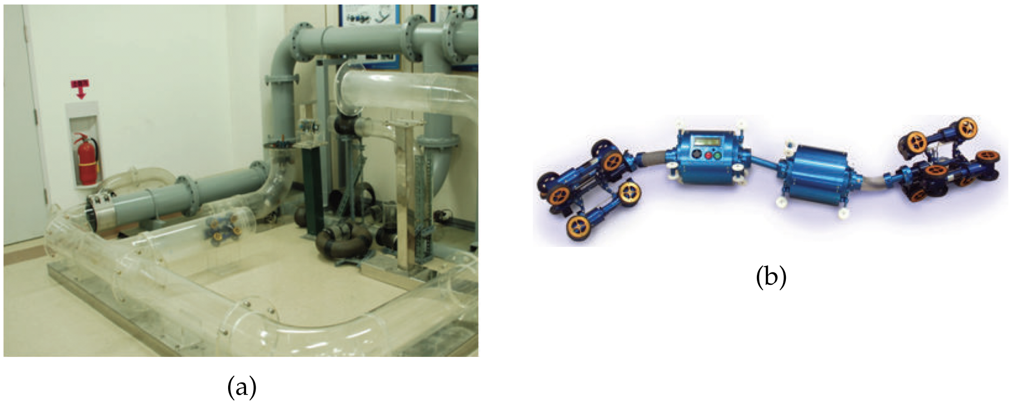
In terms of navigation, the robot uses its onboard camera to capture a sequence of images of the internal pipe wall, as well as distance information from a laser range finder. By assuming the profile of the internal pipe wall is known, localisation is performed by those two sensors. The accuracy and performance of localisation is directly related to the prior knowledge of the pipe wall profile. The inspection is performed by drawing the contour of the internal pipe wall of 360°. A limitation of this work is that the map is assumed to be known.

Camera + IMU + Laser + Motor encoder

Lee et al. [88] also applied a visual-SLAM method to robots in water pipes, by using an IMU, a laser transducer and a camera (see Fig.2.10). The visual camera captures the feature of pipe joints and elbows to localise the robot while producing a 3D map of the pipeline. Whilst apparently effective, testing was limited to a small-scale laboratory environment.

IMU + Motor encoder

Murtra and Mirats Tur [108] used a combination of IMU and a motor encoder to estimate the location of a sensor carrier in water pipe. This is a simple dead reckoning scheme and is not SLAM but is included here for completeness. The other methods discussed above all use some type of corrective sensor such as a camera and/or a laser range finder to observe the environment, which is used in localisation and/or mapping. The limitation of a dead reckoning system is that it can drift over time and unexpected events such as blockages will go undetected.



(a)

(b)

Figure 2.10: Experiment setup in laboratory environment in Lee et al. [88]: (a) experiment setup, (b) sensor carrier: *MRINSPECT V*.

2.6 Summary

In a water distribution system, water pipes are ageing, and have leakages and need regular maintenance and inspection. However, most pipe environments are inaccessible to humans. Thus, a water pipe robot system would be very useful for water companies.

Many robot pipe inspection systems have been developed but a key gap in most of these is that they do not include a navigation system. When a robot is inspecting water pipes, it needs to know its current location and the location of the damage/leakage, which motivates this research on robot navigation systems for water pipes.

Robot navigation using the SLAM technique has been reviewed here, highlighting the main techniques of EKF-SLAM and FastSLAM (based on particle filters). Although there has been much research on robot navigation using SLAM in indoor and outdoor environments, they are mainly applicable to feature-rich environments.

Navigating in water pipes presents a number of different challenges to those commonly encountered in typical SLAM problems, particularly a lack of landmark features. It has been found in this review that only a very small number of attempts have been made to solve the problem of navigating in water distribution pipes, mainly based on cameras and IMUs such as are used in typical indoor/outdoor environments. This motivates further research on robotic navigation systems that are designed specifically for water pipes.

Chapter 3

The Front End: From Sensor Data to Maps for Water Pipes

3.1 Introduction

This chapter will illustrate the *front-end* part in this SLAM project and that includes the experiment setups, sensor modules used, sensor raw data transformation to readable data. As explained in Section 2.3.1, the *front-end* in a typical SLAM problem mainly focuses on feature extraction and data association. The observation equation is usually a nonlinear equation which transfers raw sensor data into readable data for the later *back-end* process. Therefore, the objective of this chapter is to obtain the readable observation data or observation equations.

This thesis contains three different laboratory experiment settings. Modern water distribution pipes are mostly made of plastic, thus we firstly conduct experiment for plastic pipes. However, there still exists pipes that are made of metal. Therefore, experiments on metal pipes have also been conducted from short (1m) to long (5m). Details of these pipes are explained below:

1. For plastic pipes – use a 300 mm width plastic board and 6 mm in its thickness to represent plastic pipe wall and the soil underneath representing the soil outside the plastic pipe wall (see Fig.3.2a).

2. For experiment I in metal pipe, use a 1m long, 88 mm in external diameter pipe vertically placed. Only the 40 cm mid-section is used for scanning (see Fig.3.5).

3. For experiment II in metal pipe, use a 5m long metal water pipe, which is horizontally placed and 70 mm in its internal diameter and 80 mm in external diameter (see Fig.3.6).

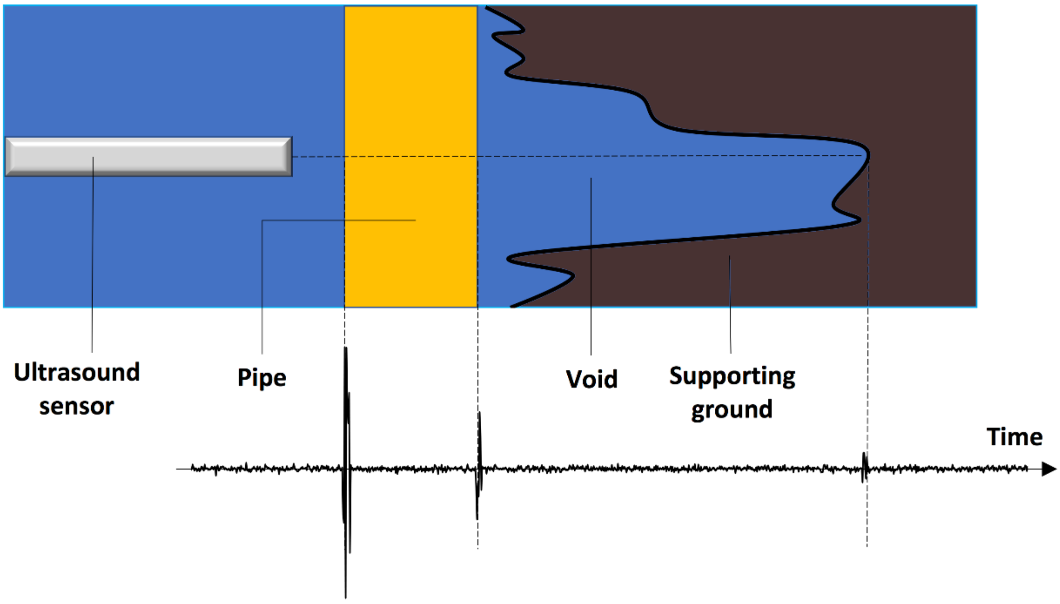


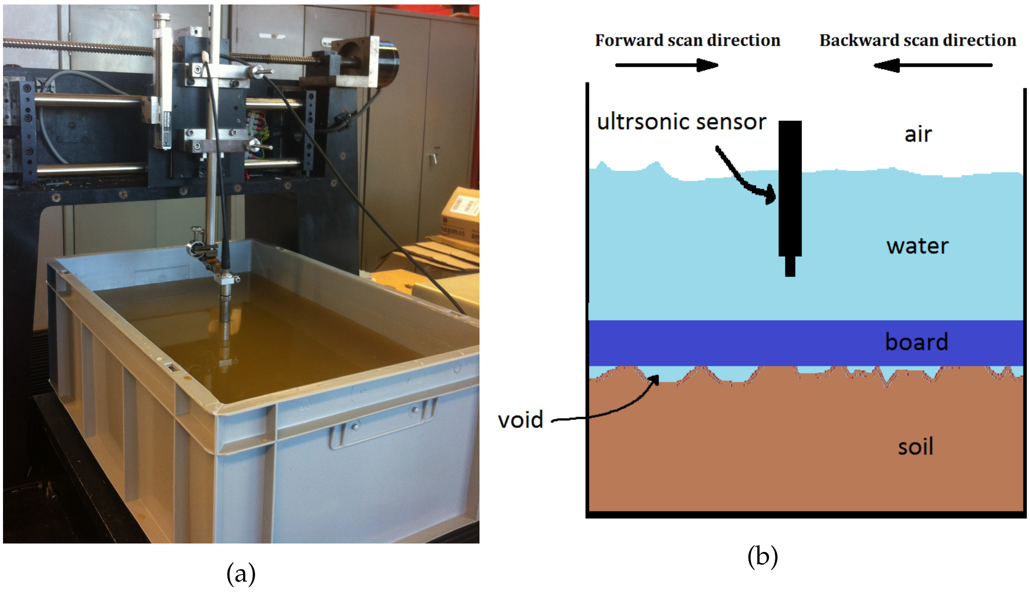
Figure 3.1: Diagram showing the principle of ultrasound scanning in a plastic water pipe. The first reflection is when the ultrasound reaches the inter pipe wall; the second is the reflection when it reaches the outer pipe wall; the third is the reflection when it reaches soil surface outside the pipe.

3.2 From Ultrasound Data to Navigation Maps in Plastic Pipes

3.2.1 Concept

Ultrasound is used to measure the distance to an obstacle by calculating the time of ultrasound flight. The first reflection of the ultrasound is often used in mobile robotics to calculate distance to the nearest object - that would be the inside of the pipe wall for a robot in a pipe. Ultrasound is also used in this way with pipe inspection robots for damage detection.

An additional advantage of ultrasound used in water is that the signal can be transmitted through a plastic pipe wall and be reflected off the ground behind, outside the pipe, although with weaker signal power. This technique has been used to detect voids outside of plastic pipe walls, where the voids might lead to damage of the pipe due to lack of ground support [156]. In this robotics project, a novel use of ultrasonic signals penetrating through the pipe wall is to create a new type of map for navigation based on the ground terrain profile outside the pipe, which is advantageous because the terrain outside the pipe is likely to have more features, i.e. navigation landmarks, than the inside of the pipe wall, which tends to be smooth and featureless.



(a)

(b)

Figure 3.2: Experimental setup in the laboratory environment. (a) Ultrasonic sensing probe, mounted on an x-y motorised arm in a water bath. At the base of the water bath is a layer of soil, over which is a plastic board of similar width and material as a water pipe. (b) Diagram of the lab setup shown in panel (a).

In Fig.3.1, the ultrasound sensor receives an echo after emitting a signal directed at the pipe wall. The echo signal is a time-domain signal wave, which contains 3 pulses (lower time-series plot in Fig.3.1). The first pulse is the strongest pulse, which is the reflection from the inner pipe wall. The second is a smaller reflection from the outer pipe wall. The third pulse is the reflection from the soil outside the pipe. This pulse is the one of interest for creating a navigation map based on the ground terrain profile outside the pipe.

3.2.2 Experiment Settings and Sensor Platform

The ultrasonic transducer was moved through a water bath over plastic pipe material to emulate the water pipe environment. The base of the water bath was covered in soil, with the plastic pipe material resting on top (Fig.3.2). At certain locations in the soil, voids were inserted to create landmark features for evaluating the navigation algorithm.

The ultrasonic transducer had a central excitation frequency of 10 MHz and focal distance of 75 mm, mounted to the gantry of a stepper motor driven scanning table. The transducer was pulsed at a rate of 160 pulse/s using PC mounted pulser-receiver and digitisation cards. The location of the transducer was recorded for each pulse. The reflected ultrasound was windowed such that the reflections

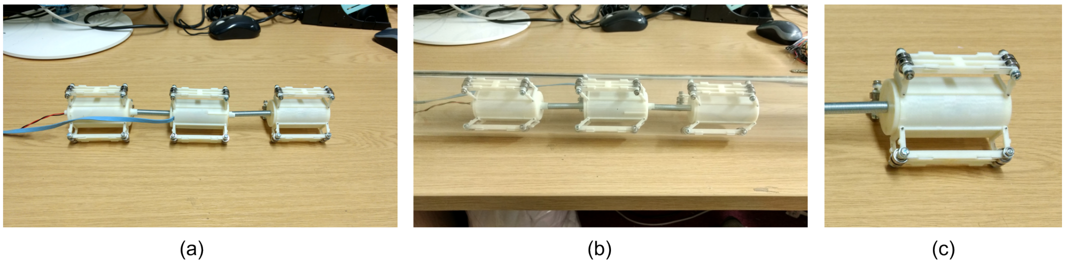


Figure 3.3: Design of a water pipe inspection robot for small diameter water pipes. The prototype shown here is designed to operate in 3 inch pipe and is tethered to simplify recovery in the event of a robot failure (a requirement of water utilities). The use of a tether can also be exploited to supply power and off-board processing. The robot is composed of modules that are flexibly linked by a steel spring. Each module is composed of a 70mm long by 29 mm diameter core unit, which contains sensors and processing units. The flexible arms extend the diameter of each module to the range 65-80mm, bracing against the inner pipe wall for stability. (a) Robot with 3 modules. (b) Robot in 3 inch clear plastic pipe. (c) Zoomed view of one robot module.

extending from the upper pipe surface to approximately 80 mm past the lower surface of the pipe could be observed and digitised at a rate of 100 MSamples/s.

3.2.3 Experiment Data

The experiment data was collected in a simple laboratory setting (see Fig.3.2). The ultrasound sensor head is moving along the plastic board from the left to the right. At each location, an ultrasound time sequence is received. By calculating the index of third reflection in the time sequence (see Fig.3.1), a single number indicating the index of the third reflection can be achieved. After the sensor head reaches the right end, a sequence of data indicating the reflection indices are gathered. A total number of 8000 indices is gathered. After the sensor head returns to the left end, a complete set of 16,000 measurements are obtained.

3.2.4 Ultrasound Derived Maps for Plastic Pipes

The map for navigation generated from ultrasound data is shown in Fig.3.4. Ultrasound transducer moves from one end to the other end above a plastic board Fig.3.2b. A one dimensional sequence of data is measured with 8000 discrete measurement points along the plastic board. The amplitude of the ultrasound data is referred to the time-of-flight. But due to the fact that ultrasound reflection waves travels through water, pipe and air outside the pipe wall, the exact distance is not measurable. On the other hand, in this robot navigation project, it is not important of the physical meaning of the map, as long as the map itself can help on in-pipe robot localisation (in Chapter 4) and SLAM (in Chapter 5). The data shown in Fig.3.4a has some undesirable disturbances noises and they are smoothed out by using median filter and results in Fig.3.4b. Both localisation and SLAM will not work if a long region of the map is too flat. Therefore, an additional artificial soil void (from location 500 to location 750) is added to the map, based on the first big void from location 0 to location 300. Thus, the final ultrasound derived soil depth map is shown in Fig.3.4c.

3.2.5 Discussion

In this Section, a new use of ultrasound sensor data was introduced. Various robot navigation literature have used ultrasound sensor to detect nearest obstacles by measuring the time of flight of the first pulse in the reflection time-domain signals. But in this project, a novel use of this ultrasound reflection data (measuring the third pulse) has proved that the soil void depths outside the plastic pipe wall can be used as an important feature due to its variations along the pipe length.

However, the major drawback of this technique is that the ultrasound transducer has to point to one direction during its observation. Otherwise, it may observe different soil profiles and the soil depth map will be changing because the measured soil depth map is one dimensional while in practical scenarios the pipe is three dimensional in space.

It is worth noting that, compared to other robot navigation projects, the observed 'feature' in this project is still far less – one numeric value at each location along the pipe. However, in a typical visual SLAM problem (either outdoor or indoor environment), an image captured from a visual camera contains too much information (landmarks). Tracking one important landmark while ignoring other landmarks may still do the job. However, in this project, one has to make full use of the one dimensional data and any undesirable disturbances may have unexpected impact on the robot navigation.

3.3 From Hydrophone Data to Navigation Maps in Metal Pipes

3.3.1 Concept

A hydrophone emitting lower frequency sound waves, compared to ultrasound, can be used to excite pipe vibration. When a hydrophone emits sound waves, the metal pipe will vibrate and the vibration will be recorded by another hydrophone

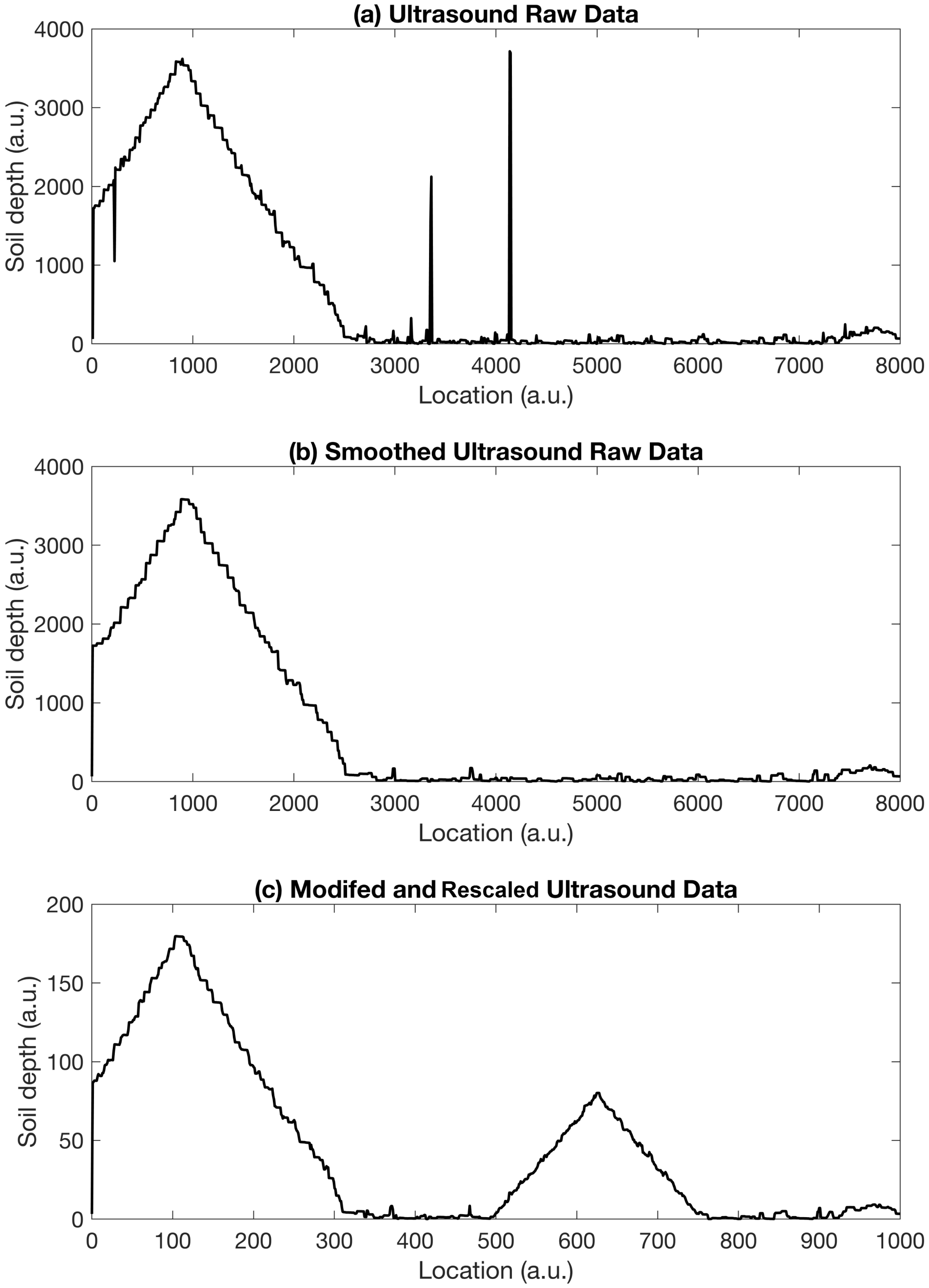


Figure 3.4: Ultrasound data derived map from experiment conducted in laboratory environment. (a) Raw ultrasound transducer measurement in laboratory. (b) Smoothed ultrasound data after applying a median filter to filter out undesirable disturbances. (c) An artificial void (from location 500 to 750) is added to create another feature for localisation and SLAM. It is then rescaled to match the amplitudes in later metal pipe data.

receiver. The recorded echo signal contains various frequency components which are related to the mechanical structure of the pipe and other surroundings. Due to the fact that the underground surroundings along the metal pipe are not perfectly the same, the frequency components of the echo signal at each location may vary. Therefore, this frequency variation along the pipe can be used as another type of feature compared with the smooth and featureless inner pipe wall.

After the hydrophone transducer receives an echo signal in time-domain, the echo signal can be transformed into frequency-domain using Fast Fourier transform (FFT). However, not all frequencies are of interest, and this is because some frequency components are not changing along the pipe while some frequency components will change rapidly as the hydrophone travels through the pipe. Therefore, the front-end process will filter out those frequencies that are not changing much along the pipe and keep those frequency components that are changing. Therefore, by calculating the amplitudes of those frequency components or the mean amplitude of certain frequency band of that signal, a one dimensional map can be obtained.

Similarly to the case of plastic pipes with ultrasound mentioned previously, this one dimensional map obtained from hydrophone data can be used as a navigation map or the offline data for SLAM in metal water pipes.

3.3.2 Experiment Settings and Sensor Platform

The experiments on metal pipes are separated into two different metal pipes. The first experiment is conducted in a short metal pipe (1m length, 88 mm in external diameter). The second experiment is conducted in a longer metal pipe (5m length, 70 mm in internal diameter and 80 mm in external diameter).

In order to demonstrate the feasibility of the mapping and localisation technique described above, a small-scale laboratory experiment was constructed. A steel pipe, of dimensions 1 metre in length, by 88 millimetres in external diameter, was inserted into a concrete mould in a water butt, which was then filled with water (Fig.3.5). The pipe material was chosen to be steel as opposed, to e.g. cast iron, because the acoustical properties of steel are well known. This makes steel a more appropriate material for testing in the early stages of developing this novel technique.

A 3D printed unit was used to house a pair of hydrophones (Bruel&Kjaer type 8103) emitting signals around 3.7×10^3kHz. This unit was then immersed into the steel pipe in an ultrasonic pulse-echo setup. The input signal to the pulser was a waveform generator (Tektronix AFG3022C), amplified by a Bruel&Kjaer type 2713 amplifier, which produced pipe vibration. The receiver unit, the second

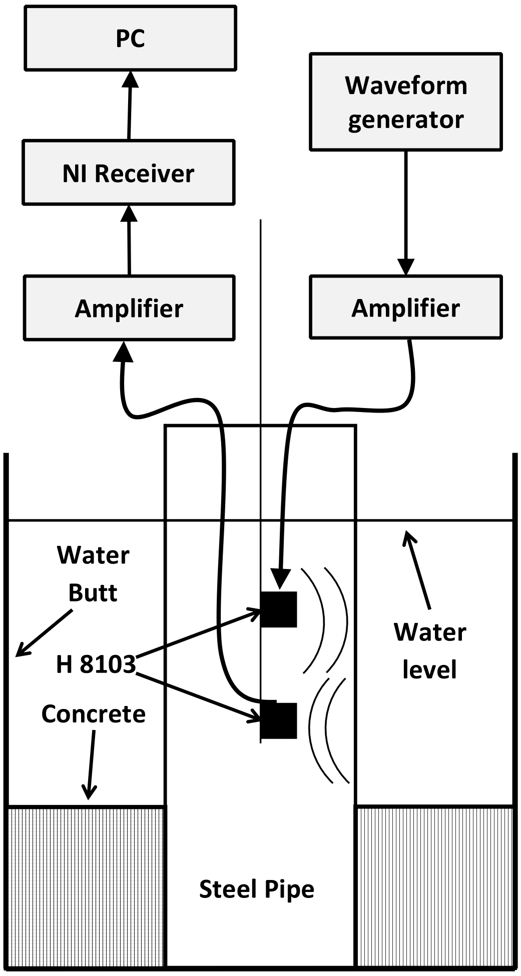


Figure 3.5: Experimental setup for 1m metal pipe: The hydrophone pulser and receiver unit, H8103, travels up and down a one metre steel pipe that is immersed in water. Experimental recording was conducted on a 40 cm mid-section of the pipe.

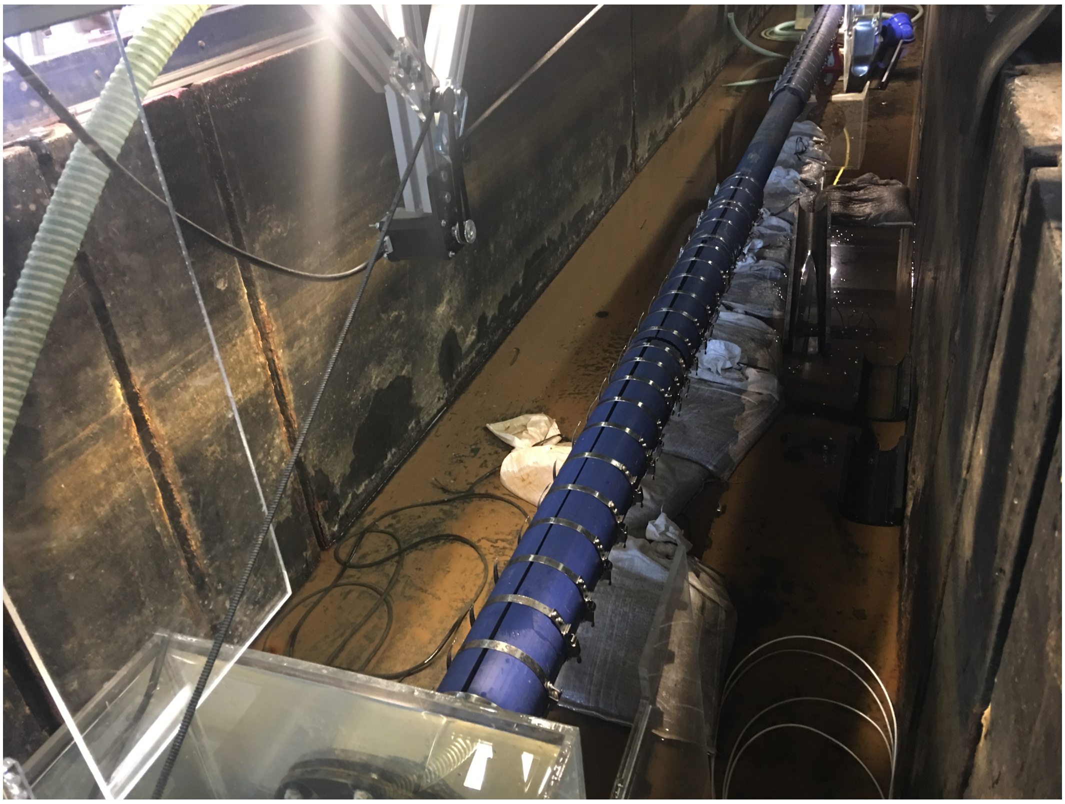


Figure 3.6: Experimental setup for 5m metal pipe: The hydrophone pulser and receiver unit, H8103, travels up and down a one metre steel pipe that is immersed in water. Experimental recording was conducted on a 40 cm mid-section of pipe.

hydrophone connected to an additional amplifier (Bruel&Kjaer 2693), measured the amplitude of the pipe vibration. Finally this output signal was logged on a PC using a National Instruments BNC 2110 receiving unit.

In this preliminary work the sensors were not mounted in the robot in Fig.3.3, which would have over complicated the laboratory experiment. However, the robot is designed to carry a sensor payload such as used here.

3.3.3 Experimental Data

The map of pipe vibration amplitude (measured in arbitrary units, a. u.) was constructed over a 40cm mid-section of a 1m pipe. Data was logged at 0.5 centimetre spacing over this mid section (see Fig.3.7 for experiment data). A fast Fourier transform (FFT) was used to transform the data from the time- to frequency-domain to obtain the amplitude of pipe vibration at each spatial location. The amplitude was averaged over the range 15-25 kHz to produce a one-dimensional map of mean pipe vibration amplitude over space (Fig.3.7c). This one-dimensional function in Fig.3.7c corresponds to the map $h(.)$ defined in (4.2).

3.3.4 Hydrophone Derived Maps for Metal Pipes

Hydrophone derived maps for 40cm mid-section in 1-metre metal pipe is shown in Fig.3.7c. Hydrophone derived maps for 5-metre pipe is shown in Fig.3.8d and Fig.3.9d using different derivation parameters. For 5-metre metal pipe, two sets of experiment data are gathered (Data I in Fig.3.8 and Data II in Fig.3.9).

The experiment in 1m metal pipe is conducted in a one way travel – the sensor platform starts from its original location and travels for 40cm and then stops. However, the experiments for 5-metre metal pipe contains two passes (forward pass and return pass). The sensor platform moves from left end of the pipe to the right end of the pipe and then return. It is worth mentioning that the data collected from 1m metal pipe only contains a "single path" that is left-to-right, but the two sets of data (Data I and Data II) collected in 5m metal pipe contain a "full return path" that is left-to-right-to-left.

On the sensor platform, one hydrophone transmits signals and the other receives response signals. Signals are time-domain waves. By using a Fast Fourier transform (FFT), a frequency-domain response signal can be obtained for each time-domain signal. Using Data II in Fig.3.9 as an example, the brighter in colour map in Fig.3.9b the more frequency components in that frequency range. By taking an average over a certain frequency band, a one dimensional map (green line in Fig.3.9c) can be calculated. As can be seen that it is too noisy, after taking a median filter, the smoothed map is shown in black. In Fig.3.9d, the black smoothed map covers a forward pass (0m to 4.995m) and a backward pass (4.995m to 0m). The red line is the mirrored backward map which overlays on the forward pass map. The reason to plot this is because due to unexpected disturbances, the map derived from the forward pass does not guarantee to match the one from the backward pass. To select an average frequency band, one has to consider two things: 1) the mismatch error of forward map and backward map should be as small as possible; 2) the derived map should contain measurable features, which means the map should not be too flat.

Data I and Data II from the same pipe at same controlled conditions. The reason for the difference between their frequency maps is that an additional noise-filtering process is conducted in Data II that low frequency components in its time-domain signals are filtered out.

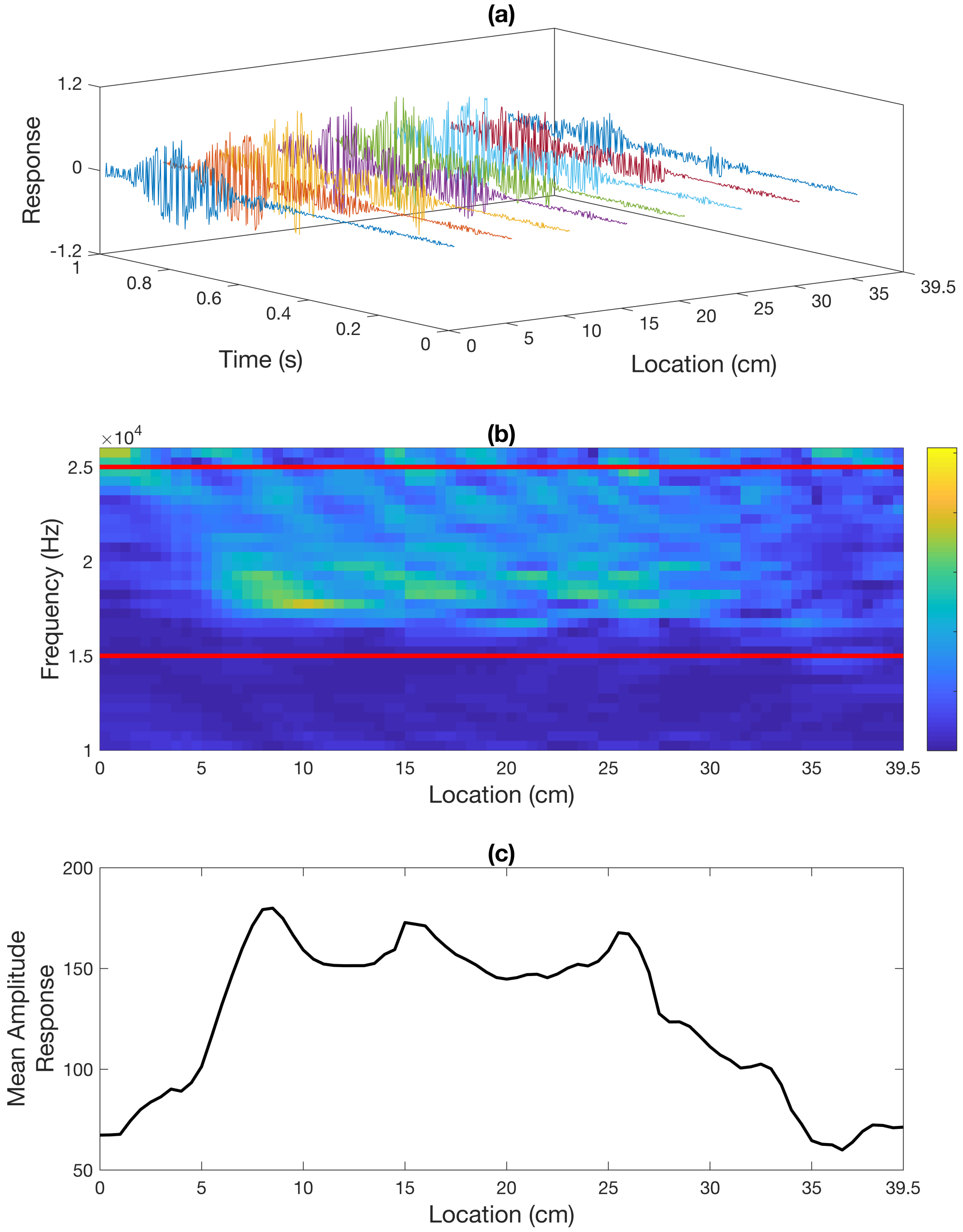


Figure 3.7: Experimental data in 1m length metal pipe: (a): Time-domain signals observed by the hydrophone over 40 cm of pipe. Signals were observed at 0.5 cm intervals but for clarity the graph only shows signals at 5 cm spacing. (b): Space-frequency representation of hydrophone signal amplitude obtained from an FFT of the time-domain signals. The red lines indicate the region over which the average amplitude is taken to form the one dimensional map. (c): The hydrophone map of amplitude over space.

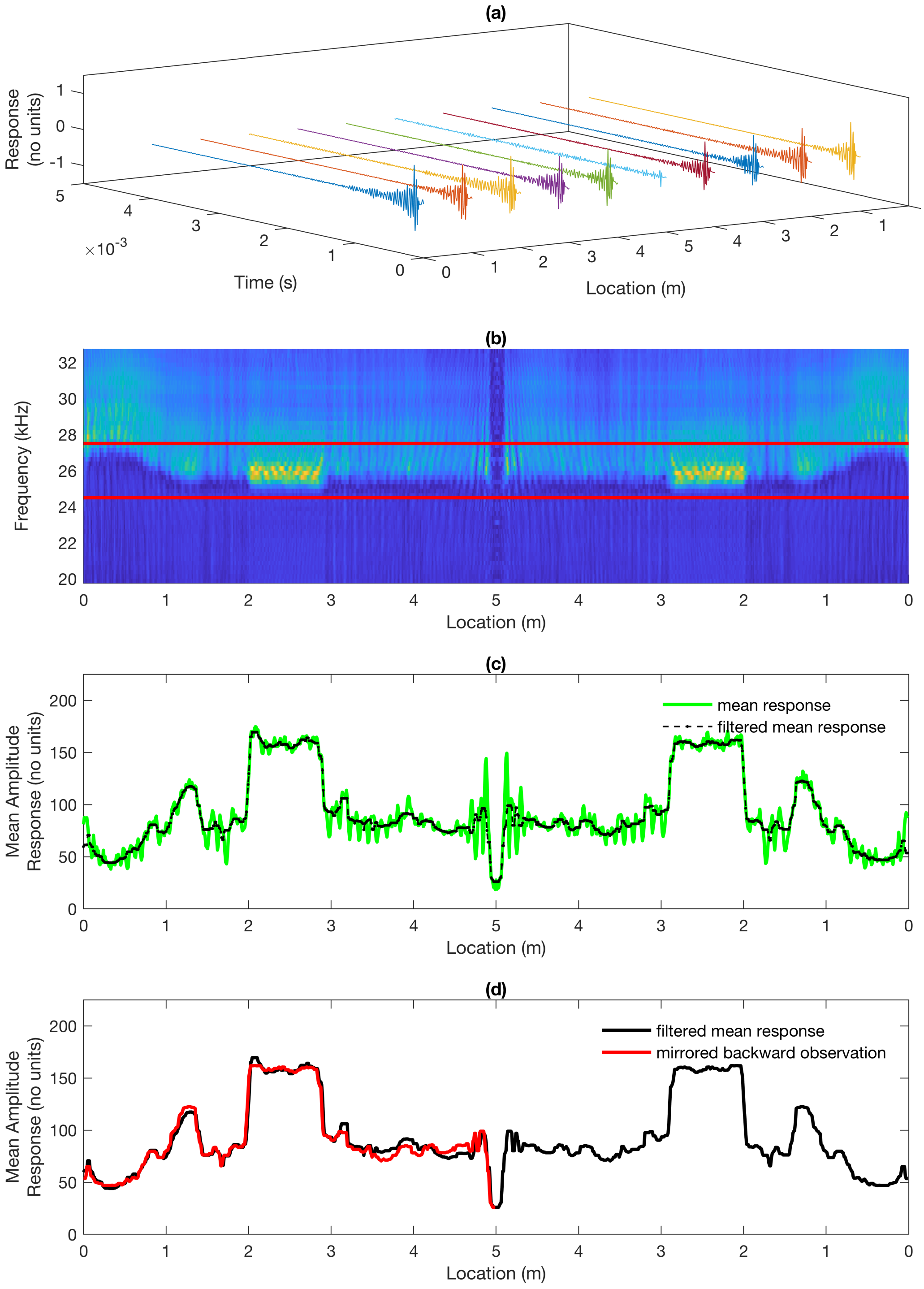


Figure 3.8: Experiment in 5m metal water pipe (data I). (a) Time signals measured by hydrophone. (b) The space-frequency representation of pipe vibration amplitude. The amplitudes over frequency were obtained from an FFT of the time-domain signals. The red lines define the frequency range used to average over to form the map. (c) The map of pipe vibration amplitude over space. (d) The comparison of forward observations and mirrored backward observations.

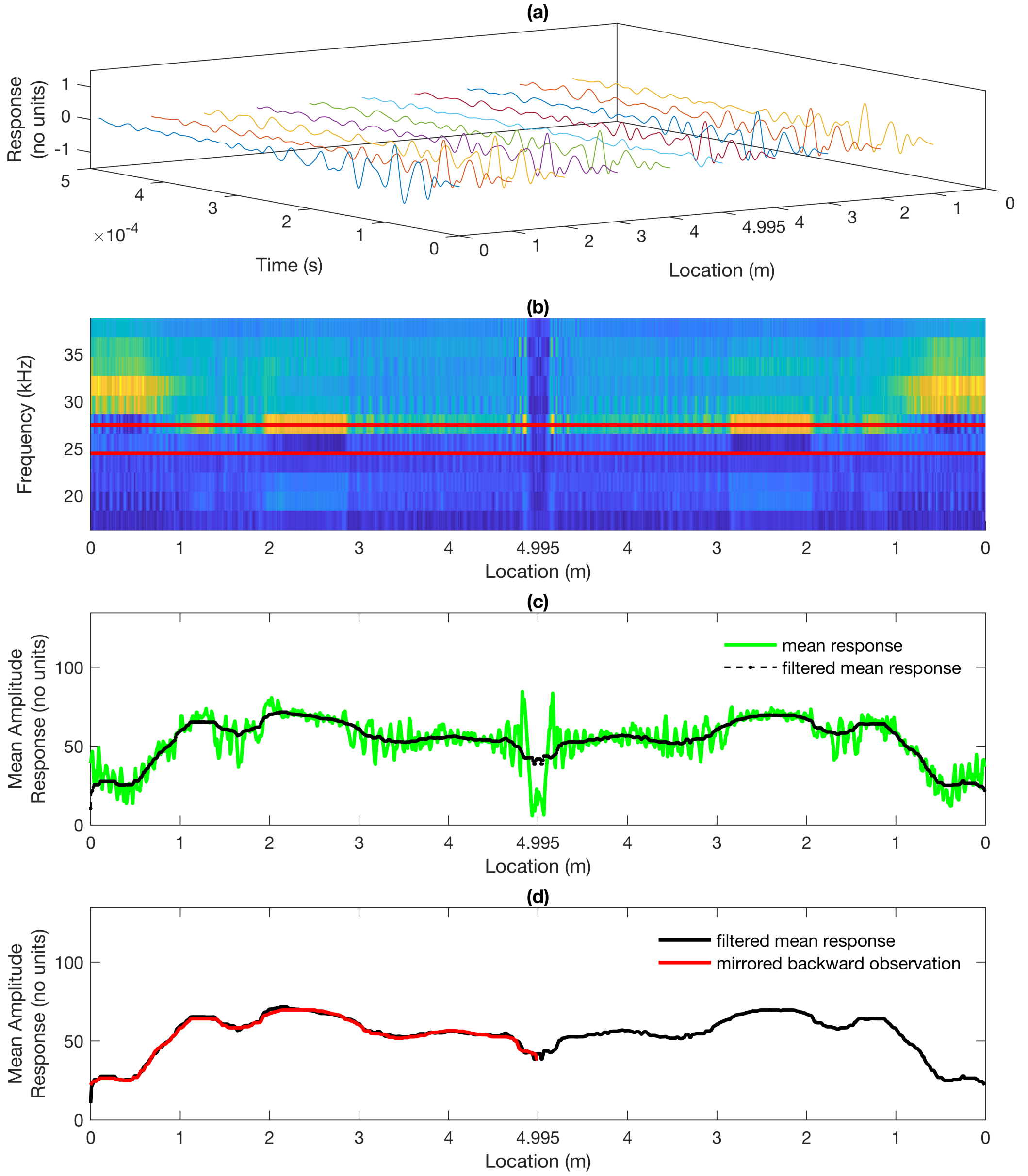


Figure 3.9: Experiment in 5m metal water pipe (data II). (a) Time signals measured by hydrophone. (b) The space-frequency representation of pipe vibration amplitude. The amplitudes over frequency were obtained from an FFT of the time-domain signals. The red lines define the frequency range used to average over to form the map. (c) The map of pipe vibration amplitude over space. (d) The comparison of forward observations and mirrored backward observations.

3.3.5 Discussion

This section introduced how navigation maps are derived from hydrophone data in three experiments in two different metal pipes. The principle behind the derivation of hydrophone data is the same. One hydrophone transmits signals to the metal pipe, and the other hydrophone records the response signals. A frequency colour map is computed by taking a Fast Fourier transform (FFT) to the time-domain signals. By taking an average over a selected frequency band, a mean frequency response map is then obtained.

Compared with literature, the novelty of this work is that hydrophone is firstly used as a navigation sensor in metal water pipes. Similar to the ultrasound sensor, the derived navigation map is in one dimension. Although it creates some measurable features in metal pipes, the 'feature' is still far less. Compared with visual camera tracking landmarks, the one dimensional map contains too little information, where visual image data contains too much information. The advantage of hydrophone sensing is that it can be applied to any metal pipes even if their internal pipe wall are very smooth and are of same colour. However, if the internal metal pipe wall is identical every where, visual camera sensing will fail.

The weakness is that for the first time processing hydrophone signals in an unknown pipe, one cannot easily determine the frequency range to average over. Because averaging over different frequency range will result in different map, and the derived map may not contain enough features if it is not well selected and later navigation may fail if use a bad map.

3.4 Summary

In this chapter, the *front-end* part of robot navigation is introduced. It has illustrated the experiment of using ultrasound sensing technique and hydrophone sensing technique for plastic pipes and metal pipes. For plastic pipes, ultrasound can go through the plastic pipe wall and therefore the reflection can be received by a ultrasound transducer. A soil void depth map can be derived based on the ultrasound refection data. For metal pipes, a hydrophone transmits signals to the metal pipe and another hydrophone records that response signals. The time-domain signals are processed using FFT and averaged over a certain frequency band to derive a one dimensional map for navigation.

The sensor platforms in these experiments are tethered to a motor which provides power supply and sensor measurements are transmitted from the tethered cable to a PC. It is assumed that the motor encoder works perfectly that the motor encoder gives accurate locations during sensor travelling. However, in practical scenarios, motor encoder will not produce 100% accuracy in estimating sensor location.

Chapter 4

Sequential Mapping and Localisation in Water Pipes

4.1 Background

The previous chapter has shown how navigation data in both plastic and metal pipes can be obtained, reducing feature sparsity, so that the robot can in principle produce a map and estimate its location. This problem is addressed in this chapter by using a sequential mapping and localisation method, i.e. first mapping the pipe and then secondly using the map to perform localisation (as opposed to the simultaneous approach used in SLAM in Chapter 5). This sequential approach is appealing because it should be less prone to divergence than SLAM (because only one quantity is estimated at a time, the map then location) and it is just as useful for the purpose of repair from above ground because data can be collected, and then post-processed for the map, and then the location.

The previous chapter described new types of maps that can be derived from ultrasound data for plastic pipes and hydrophone data for metal pipes. However the problem is that the only way of calibrating the spatial component of the map is by a dead reckoning sensor such as a motor encoder, which could be subject to unknown drift. To solve this problem, it is assumed that the robot will make multiple passes up and down the pipe between two known locations (a realistic assumption because known entry points into water pipe networks are spaced by about 100 metres in the UK). This means that the map calibration can be improved by spatial averaging of robot movements up and down the pipe between two known locations.

Taking a direct average of the data would be likely to lead to smoothing of the peaks and troughs in the map, degrading features required for localisation.

Instead, the key novelty in this chapter is to use a signal alignment technique to warp the maps in the spatial direction before averaging. This improves the spatial calibration without overly smoothing the map features. The signal alignment and averaging algorithm is based on dynamic time warping (DTW) and is known as DTW barycentre averaging (DBA) [116].

Section 4.2 defines the mapping and localisation problem, presents the mapping solutions based on DBA and the two alternatives to localisation, one based on the extended Kalman filter (EKF) used in terrain-based localisation [65], and one based on the particle filter (PF) [129]. The experimental details for the hydrophone induced vibration of the metal pipe is also presented in this section. Section 4.3 presents the results on evaluation of the mapping and localisation methods using a combination of experimental and simulation data. Table 4.1 and Table 4.2 show the lists of data that are used to evaluate the localisation algorithm. Table 4.3 have shown the experiment results. Finally, section 4.4 concludes the chapter.

4.2 Methods

4.2.1 Problem Statement

In defining the concept of operations for the pipe inspection robot considered in this work (see Fig.3.3 for a prototype), a number of assumptions are made. The first regards the robot deployment. In consultation with project partners from the water utilities industry, e.g. Yorkshire Water, it requires the robot to enter the water pipe network through existing access points to minimise costs. Assuming fire hydrants could be used, which in the UK, Europe and the USA are separated by approximately 100 metres [121, Chapter 14, Table 14.2].

This leads to the second assumption, that the robot will travel between two points with known location, i.e. two fire hydrants. The third assumption is that robot travel time between two hydrants, in terms of relative time cost to e.g. robot deployment, is relatively trivial, hence it is worth the robot making multiple passes up and down the pipe in order to maximise the mapping and localisation accuracy.

Due to the fact that navigation through the pipe in itself is relatively trivial, i.e. forwards or backwards, the need for a SLAM solution is limited, hence, the sequential approach to mapping and localisation is taken here. In addition, only consider the problem of localising with respect to one dimension, i.e. distance travelled through the pipe: this technique is not suited to correcting heading, which is left to future work.

Define the mapping and localisation problem as follows: 1. to estimate a map $\mathbf{h}(\mathbf{x}_k)$ from hydrophone-induced pipe vibration signals, that transforms robot pose

$\mathbf{x}_k \in \mathbb{R}^{n_x}$ at time-step k to sensor measurements $\mathbf{y}_k \in \mathbb{R}^{n_y}$, where $\mathbf{h} : \mathbf{x}_k \rightarrow \tilde{\mathbf{y}}_k$, where $\tilde{\mathbf{y}}_k$ is the noise-free sensor output, and robot pose for example is $\mathbf{x}_k = \left[x\ y\ \psi_z\right]^T$, i.e. location in x-y co-ordinates and heading ψ_z (yaw); and 2. localise the robot by obtaining the estimate of the pose distribution $p(\mathbf{x}_k|\mathbf{y}_k)$. For a straight pipe, heading is not needed, and we only care about the robot distance travelled from the origin. But for expandability and integrity of this project, y and ψ_z are added to the state $\mathbf{x}_k$.

Assume that the dynamics of the pipe robot can be represented by a state-space model, with state dynamics

$$p(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{u}_{k-1}) \Leftrightarrow \mathbf{x}_k = \mathbf{f}\left(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}\right) + \mathbf{w}_{k-1} \tag{4.1}$$

where $\mathbf{f}(.)$ is the state transition function, $\mathbf{u}_{k-1} \in \mathbb{R}^{n_u}$ is the input, $\mathbf{w}_{k-1} \sim N(\mathbf{0}, \Sigma_w)$ is the state noise. The measurement model is

$$p(\mathbf{y}_k|\mathbf{x}_k) \Leftrightarrow \mathbf{y}_k = \mathbf{h}\left(\mathbf{x}_k\right) + \mathbf{v}_k \tag{4.2}$$

$\mathbf{h}(.)$ is the measurement function, and $\mathbf{v}_k \sim N(\mathbf{0}, \Sigma_v)$ is the measurement noise.

In this part the state vector $\mathbf{x}_k$ can be simplified to contain just the location of the robot along the pipe, $\mathbf{x}_k = x_k$ and $n_x = 1$. The observation $\mathbf{y}_k$ is the processed hydrophone signal, which is the average of the vibration amplitude over some frequency range, $|\bar{a}|_k$, hence, $\mathbf{y}_k = |\bar{a}|_k$, and $n_y = 1$. The state dynamics are assumed to be obtainable from a processed motor encoder reading, which defines distance travelled, m_k, hence

$$\mathbf{f}\left(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}\right) = \mathbf{F}_{k-1}\mathbf{x}_{k-1} + \mathbf{G}_{k-1}\mathbf{u}_{k-1} \tag{4.3}$$

where $\mathbf{F}_{k-1} = 1$, $\mathbf{G}_{k-1} = 1$, and $\mathbf{u}_{k-1} = m_{k-1}$. Although defining this one dimensional state-space model is relatively trivial, it has the advantage that it provides ready extensibility to more state dimensions for representing the pose in two or three dimensions, and also can incorporate more sensors, e.g. camera and IMU data.

4.2.2 Dynamic Time Warping (DTW) and DBA algorithms

In this work it has been demonstrated that the robot can obtain a map of pipe vibration amplitude over space by travelling through the pipe and exciting pipe vibration using a hydrophone. Corresponding locations of the robot can be calibrated using dead reckoning, e.g. from a motor encoder. However, any drift in the dead reckoning estimate will result in an incorrectly spatially calibrated map.

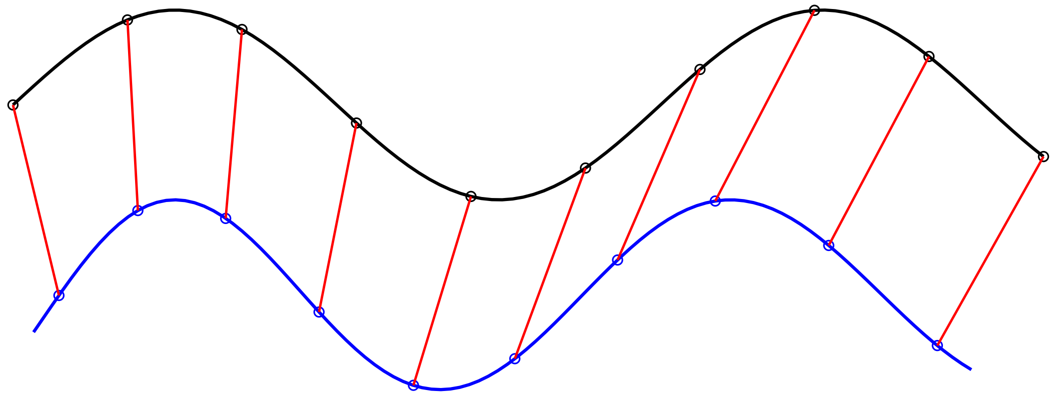


Figure 4.1: Diagram illustration of DTW averaging between two sequences. Taking the black line (as a time sequence) for example, each index in the black sequence must have at least one match in the blue line, and vice versa. The red line indicates the distance between the two indices in the two sequences.

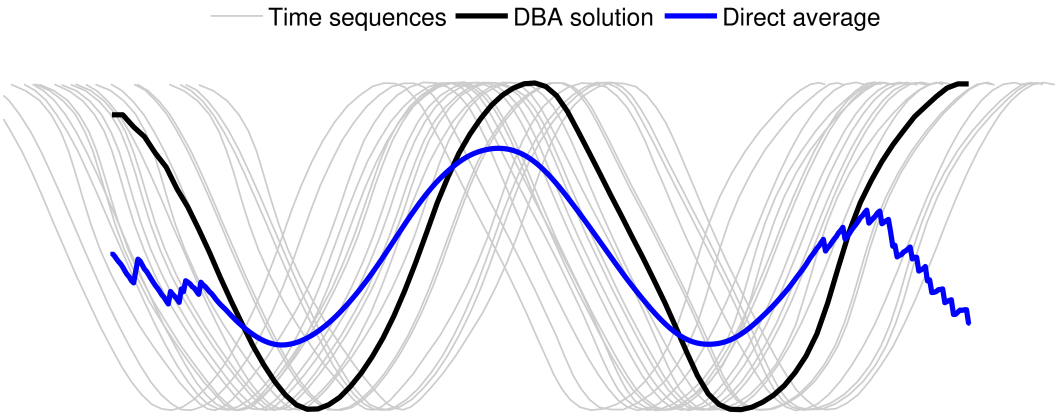


Figure 4.2: Diagram illustration of DTW Barycentre Averaging (DBA) approach. A cluster of light grey lines are the sequence samples that need to be averaged. The DBA map (black) is not vertically compressed over the indices. But the direct averaging result (blue) is compressed vertically. Therefore, direct averaging may lead to sequence distortion while DBA method can produce a more acceptable result.

Algorithm 1: DTW

```
procedure DTW($\mathbf{s}_k$,$\mathbf{s}_l$)
    $M_{1,1} = 0$
    $M_{2:N_k,1} = \infty$
    $M_{1,2:N_l} = \infty$
    for $i = 2$ to $N_k$ do
        for $j = 2$ to $N_l$ do
            $c = d\left(\mathbf{s}_k(i), \mathbf{s}_l(j)\right)$
            $M_{i,j} = c + \min\left(M_{i-1,j}, M_{i,j-1}, M_{i-1,j-1}\right)$
        end for
    end for
end procedure
```

Figure 4.3: Algorithm 1: The dynamic time warping (DTW) algorithm, which takes as input two sequences of data, $\mathbf{s}_k \in \mathbb{R}^{N_k}$ and $\mathbf{s}_l \in \mathbb{R}^{N_l}$, and returns the matrix of similarity measure between the two sequences $M \in \mathbb{R}^{N_k \times N_l}$.

A solution to this problem is for the robot to make multiple passes back and forth through the pipe in order to generate a set S of n_m independent sequences of map data, in order to average out drift errors, where

$$S = \{\mathbf{s}_1, \ldots, \mathbf{s}_{n_m}\} \tag{4.4}$$

where each data sequence is comprised of pipe amplitude response signals over space, $\mathbf{s}_j = (|\bar{a}|_{j,1}, \ldots, |\bar{a}|_{j,n_s})$, where n_s is the number of spatial samples, and for each amplitude datum there is a corresponding dead reckoning estimate of spatial location, $x^{(d)}$, hence we have the data pairs $(x^{(d)}_{j,k}, |\bar{a}|_{j,k})$, for each map $j = 1, \ldots, n_m$ and for each observation within the map $k = 1, \ldots, n_s$.

The sequences in the set S can be combined to reduce the effect of drift, however, a direct averaging of these sequences would be likely to smooth out the map due to sequence misalignment, degrading features required for localisation. Instead, we propose that sequences can be combined into a map using a signal alignment technique that warps the sequences in the spatial dimension before averaging. In this work we use a signal alignment technique based on dynamic time warping (DTW) shown in Fig.4.1, known as DTW barycentre averaging (DBA) shown in Fig.4.2.

The DTW algorithm calculates an alignment cost matrix M, between two sequences $\mathbf{s}_k$ and $\mathbf{s}_l$ (see Algorithm 1). The optimal alignment between the sequences follows a 'valley' in the cost matrix. The approach of DBA is to use DTW to compare a mean sequence estimate, $\bar{\mathbf{s}}$, to each sequence in S and iteratively reduce the total DTW cost,

$$\bar{\mathbf{s}}^* = \arg\min_{\bar{\mathbf{s}}} \sum_{i=1}^{n_m} D^2\left(\bar{\mathbf{s}}, \mathbf{s}_i\right) \tag{4.5}$$

where the quantity $D\left(\bar{\mathbf{s}}, \mathbf{s}_i\right)$ is the cumulative alignment cost calculated by DTW, where $D\left(\bar{\mathbf{s}}, \mathbf{s}_i\right) = M_{N_1,N_2}$ is obtained from the final element computed for the cost matrix M in Algorithm 1 (Fig.4.3).

The algorithm DBA can be used to obtain a solution to the optimisation problem posed in (4.5), i.e. the optimal signal average, $\bar{\mathbf{s}}^*$. The algorithm is iterative and has guaranteed convergence [117]. At each iteration k:

1. Use DTW (Algorithm 1) to iteratively compute the optimal alignment between each data sequence and the current estimate of the signal average $\bar{\mathbf{s}}_k$, i.e. DTW $\left(\bar{\mathbf{s}}_k, \mathbf{s}_j\right)$, for $j = 1, \ldots, n_m$.
2. Use the updated alignment from step 1 to update the signal average to $\bar{\mathbf{s}}_{k+1}$ and set $\bar{\mathbf{s}}^* = \bar{\mathbf{s}}_{k+1}$. Increment k and go to step 1.

The initial mean $\bar{\mathbf{s}}_0$ is defined by using one of the data sequences in S chosen at random. The algorithm is repeated until convergence, which can be monitored by evaluating the cumulative alignment cost in (4.5).

Finally, the optimal sequence of data samples $\bar{\mathbf{s}}^*$ forms the continuous map function $\mathbf{h}(.)$ from linear interpolation of the data pairs $(x_1^*, s_1^*), (x_2^*, s_2^*), \ldots, (x_{n_s}^*, s_{n_s}^*)$, where for a location x_k on the interval $\left(x_j, x_{j+1}\right)$, at sample-time k, we define

$$h(x_k) = s_j^* + (s_{j+1}^* - s_j^*)\frac{x_k - x_j^*}{x_{j+1}^* - x_j^*}. \tag{4.6}$$

Here we use linear interpolation to define the map $h(x_k)$ but an alternative such as splines could equally be used.

4.2.3 Localisation by Extended Kalman Filtering

The approach taken to localisation using the EKF is inspired by a terrain-based navigation algorithm developed for aerospace applications [65]. The steps for the EKF at sample time k consist of:

1. The prediction step for the state vector $\mathbf{x}_k$ and state covariance $\mathbf{P}_k$,

$$\mathbf{x}_k^- = \mathbf{F}_{k-1}\hat{\mathbf{x}}_{k-1} + \mathbf{G}_{k-1}\mathbf{u}_{k-1} \tag{4.7}$$

$$\mathbf{P}_k^- = \mathbf{F}_{k-1}\mathbf{P}_{k-1}\mathbf{F}_{k-1}^T + \mathbf{Q}_{k-1} \tag{4.8}$$

where $\mathbf{Q}_{k-1} = \Sigma_w$ is the state noise covariance.

2. The measurement update step requires the definition of the linearised measurement model, $\mathbf{H}_x$, which is obtained from the derivative of a local quadratic fit to the spatial map of pipe vibration amplitude, where the local quadratic approximation of the spatial map is

$$\mathbf{h}(\mathbf{x}_k^-) = a\left(\mathbf{x}_k^-\right)^2 + b\left(\mathbf{x}_k^-\right) + c \tag{4.9}$$

   where $\mathbf{x}_k^-$ is the current location prediction of the robot and hence the derivative is

$$\mathbf{H}_x = \frac{d}{d\mathbf{x}_k^-}\mathbf{h}(\mathbf{x}_k^-) = 2a\mathbf{x}_k^- + b \tag{4.10}$$

   The parameters of the local quadratic function are obtained from a least-squares fit to a data window centred on the current prediction of the robot location $\mathbf{x}_k^-$: the size of local quadratic fit window was set proportional to the state covariance, $W_{fit} = \alpha\mathbf{P}_k$ [65].

3. The EKF measurement update is performed by

$$\mathbf{K}_k = \mathbf{P}_k^-\mathbf{H}_x^T\left(\mathbf{H}_x\mathbf{P}_k^-\mathbf{H}_x^T + \mathbf{R}_k + \epsilon_k^2\right)^{-1} \tag{4.11}$$

$$\hat{\mathbf{x}}_k = \mathbf{x}_k^- + \mathbf{K}_k\left(\mathbf{y}_k - \mathbf{h}\left(\mathbf{x}_k^-\right)\right) \tag{4.12}$$

$$\mathbf{P}_k = \left(\mathbf{I} - \mathbf{K}_k\mathbf{H}_x\right)\mathbf{P}_k^- \tag{4.13}$$

   where $\mathbf{R}_k = \Sigma_v$ is the measurement noise covariance, and the term ϵ_k^2, in (4.11), is due to the linear fit error ϵ_k, where

$$\epsilon_k = \mathbf{h}(\hat{\mathbf{x}}_k) - 2a\hat{\mathbf{x}}_k - b \tag{4.14}$$

   This time-varying error term is recommended by [65] because it inflates the measurement noise covariance term $\mathbf{R}_k$ in regions of poor fit (typically due to high nonlinearity), reducing the chance of filter divergence and making the EKF more robust.

4.2.4 Localisation by Particle Filtering

As an alternative to the EKF for localisation we also investigated the use of the PF. In this case the PF is based on the bootstrap filter with sequential importance resampling [129] as used in our previous work on mapping and localisation in plastic water pipes [96].

In the first step the particles are initialised,

$$\hat{\mathbf{x}}_0^{(i)} \sim p(\mathbf{x}_0), \text{ for } i = 1, \ldots, n_s \tag{4.15}$$

where n_s is the number of samples, and initial particle weights are set to $w_0^{(i)} = \frac{1}{n_s}$, for $i = 1, \ldots, n_s$. At each sample-time k the PF performs the following steps:

1. The location is predicted by samples drawn from the state equation, Eq. 4.1,

$$\mathbf{x}_k^{(i)-} \sim p(\mathbf{x}_k | \hat{\mathbf{x}}_{k-1}^{(i)}, \mathbf{u}_{k-1}), \text{ for } i = 1, \ldots, n_s \tag{4.16}$$

   where we assume that the state equation can be used as the importance distribution of the particle filter [129].

2. The weight update step is

$$\tilde{\omega}_k^{(i)} = \omega_{k-1}^{(i)} p(\mathbf{y}_k | \mathbf{x}_k^{(i)-}), \text{ for } i = 1, \ldots, n_s \tag{4.17}$$

   where we assume Gaussian noise $\mathbf{v}_k$ on the sensor output,

$$\tilde{\omega}_k^{(i)} = \omega_{k-1}^{(i)} \exp\left(-\frac{1}{2}\left(\mathbf{y}_k - \hat{\mathbf{y}}_k^{(i)}\right)^T \mathbf{R}_k^{-1} \left(\mathbf{y}_k - \hat{\mathbf{y}}_k^{(i)}\right)\right), \tag{4.18}$$

   for $i = 1, \ldots, n_s$, where $\hat{\mathbf{y}}_k^{(i)} = \mathbf{h}\left(\mathbf{x}_k^{(i)-}\right)$. The weights are then normalised to sum to unity to get $\omega_k^{(i)}$ according to (2.62).

3. To avoid degeneracy, resampling is performed if the effective number of particles drops below a threshold, $\gamma = 0.6 n_s$, using stratified resampling [129]. Then use the selected samples to construct the new sample set $\{\hat{\mathbf{x}}_k^{(i)}, i = 1, \cdots, n_s\}$.

4.2.5 Algorithm Evaluation

The mapping and localisation algorithms were evaluated by combining the data from the hydrophone experiment with a simulated robot moving up and down the map, using the state-space model defined in (4.1) and (4.2) for the simulation. The number of map sequences generated was $n_m = 20$, five passes forward along the pipe and five backwards. The input was constant, $m_k = 0.1$ cm or $m_k = -0.1$ cm depending on direction, and drift was added to the simulated robot in the mapping stage in the form of white noise $\mathbf{w}_k$, i.e. state noise covariance $\Sigma_w = 0.05$ cm^2, also the measurement noise term $\mathbf{v}_k$ was set to $\Sigma_v = 0.01$.

For the EKF localisation the size of quadratic fit window was set to $W_{fit} = \alpha \mathbf{P}_k$, with $\alpha = 2.3$; the noise covariances were set to $\sqrt{\mathbf{Q}_k} = 0.5$ cm and $\sqrt{\mathbf{R}_k} = 5$. For the particle filter localisation, the number of particles was set to $n_s = 100$ and noise covariances were set to $\sqrt{\mathbf{Q}_k} = 0.5$ cm and $\sqrt{\mathbf{R}_k} = 5$. To make the localisation more challenging and highlight the benefit of using the map over dead reckoning, we also added a deterministic linear and sinusoidal drift term d_k to the state equation, of the form $d_k = -0.2\bar{m}_k + 0.02\bar{m}_k \sin(1.1\bar{m}_k + 2)$, where $\bar{m}_k = km_k$, where $m_k = 0.0395$ cm.

To calculate the errors in estimations, a normalised rooted mean square error is used and is shown below.

$$NRMSE = \frac{\sqrt{\frac{\sum_{k=1}^{N}(\tilde{s}_k - s_k)^2}{N}}}{s_{max} - s_{min}} \tag{4.19}$$

where s represents the true time series and $\tilde{s}$ represents the estimated time series. The time series s can be a map, a sequence of robot locations.

4.3 Results and Discussion

4.3.1 Mapping Results Using DBA Algorithm

As discussed in Section 4.2.2, a cluster of drifted time series can be calibrated using a method called DTW Barycentre Averaging algorithm, abbreviated as DBA algorithm [116] [115]. To test the performance of the DBA algorithm in map calibration, two examples are used – a simulated nonlinear map of a sine wave and 1m metal pipe experimental data. Results are shown in Fig.4.4 and Fig.4.5.

In this mapping process, the most significant noise that we considered was the drift noise in the dead reckoning (from the motor encoder). In addition, extra noise such as white noise in both sensor measurements and motor encoder were also added. Besides, some sinusoidal noise was also added into the dead reckoning.

The first result was generated using a simulated nonlinear sinusoidal map (green) plotted in Fig.4.4. Fig.4.4c and Fig.4.4d, which show the map estimation errors through a number of robot simulations (From 1 to 20). Fig.4.4a and Fig.4.4b show the DBA map estimate using 20 runs (10 runs forward and 10 backward) and are plotted in grey lines. To more clearly show the estimation errors, the drift is enlarged in the figure. The black DBA map estimation in Fig.4.4a is much better compared with that in Fig.4.4b, although the former also contains a certain amount of errors especially in the far left and far right regions.

One of the most important factors in DBA estimation is the selection of the ini-

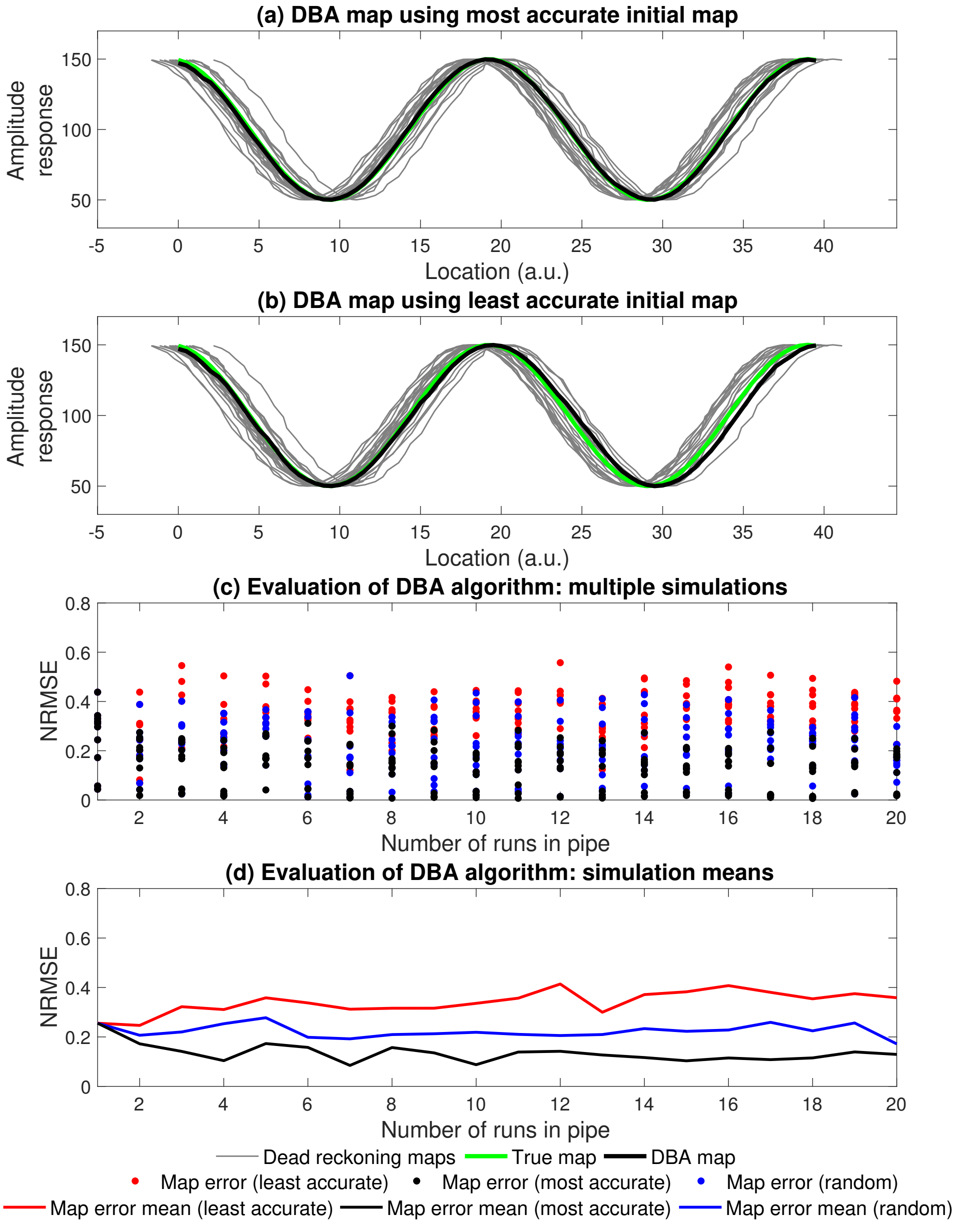


Figure 4.4: Evaluation of DBA algorithm in mapping using a nonlinear map as an example. (a) DBA map generated from 20 runs (10 forward and 10 backward) using the most accurate initial map in DBA iteration. (b) DBA map generated from 30 runs but using the least accurate initial map. (c) mapping error (NRMSE): in the DBA iterations, mapping errors using the most accurate initial map (black), the least accurate initial map (red) and a randomly selected initial map (blue). (d) Mapping error mean.

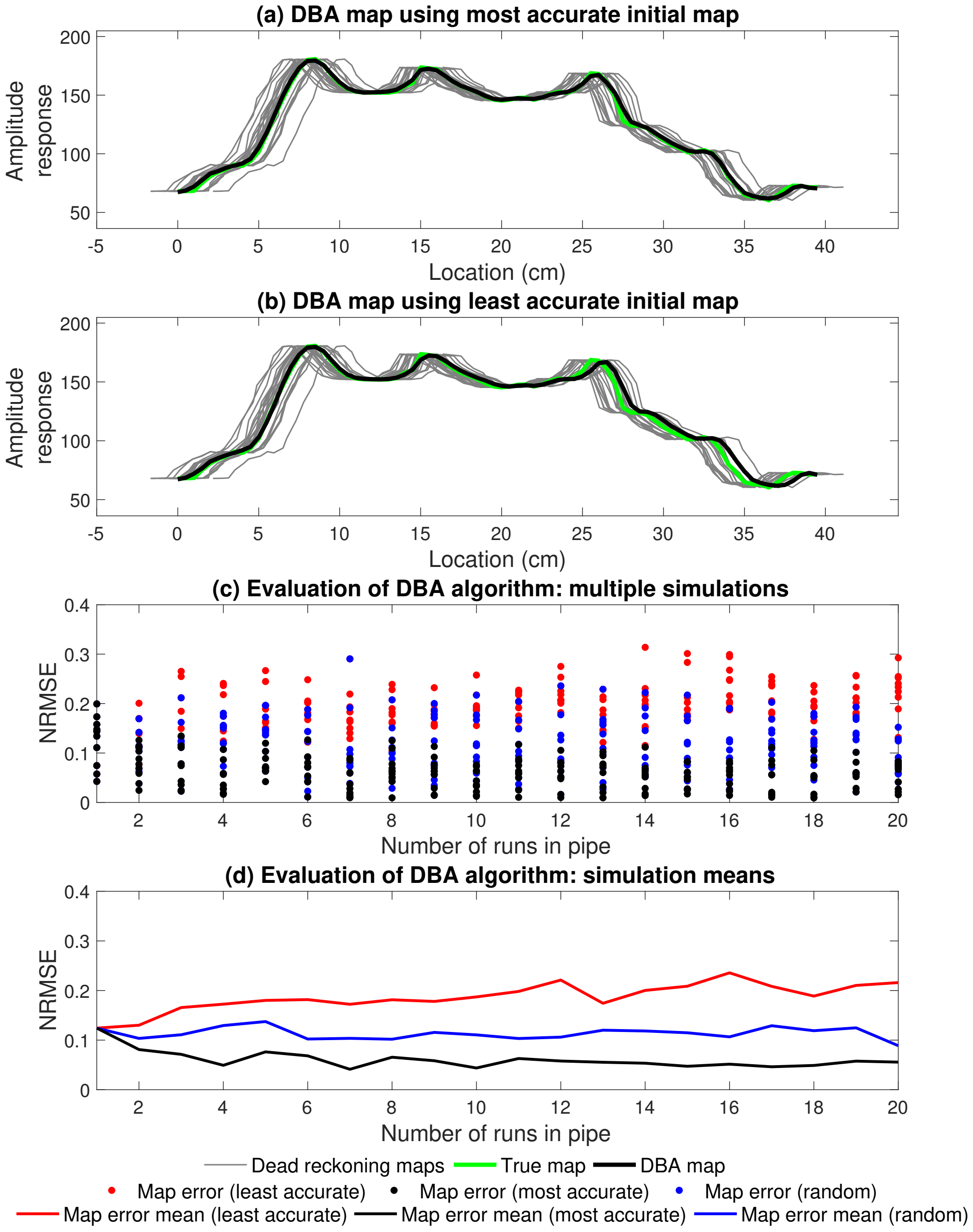


Figure 4.5: Evaluation of DBA algorithm in mapping using 1m metal pipe data as an example. (a) DBA map generated from 20 runs (10 forward and 10 backward) using the most accurate initial map in DBA iteration. (b) DBA map generated from 20 runs but using the least accurate initial map. (c) mapping error (NRMSE): in the DBA iterations, mapping errors using the most accurate initial map (black), the least accurate initial map (red) and a randomly selected initial map (blue). (d) Mapping error mean.

tial average map, which is randomly chosen from the set of observed maps. It is clear that the initial average map will affect the convergence of the DBA algorithm but it is unclear from a theoretical perspective what that affect will be. Hence, the problem now is to evaluate the impact, in simulation, of selecting different initial maps. The DBA algorithm was applied to the mapping simulations described above, but now choosing the initial map at random, repeated across 10 independent simulations. The affect of the random initial map was also investigated as a function of number of runs up and down the pipe (under the assumption that increasing the number of runs might mitigate the affect of initial map selection).

Fig.4.4 and Fig.4.5 show best case and worst case maps along with the mean values of the map estimation errors, versus the number of runs in the pipe. From the results, it can be seen that if the first map selected is the most accurate (black line) then the estimation error is much smaller then the other two (red and blue). With the increased number of runs in the pipe, the black line is smooth and stable. However, if the first map selection is the least accurate (red) then the estimation error is slowly increased. Random selection (blue) lies somewhere between those two. On average, it would appear that the number of runs is relatively independent of the map error when randomly selecting the initial map.

The most important point regarding the random selection of the initial average map, is how it actually affects the localisation accuracy, when the map is used in the localisation algorithm. This point is addressed below in section 4.3.6 (Fig. 4.26).

4.3.2 Localisation Results from Simulation Data

Simulation results are compared with linear and nonlinear map profiles using EKF and particle filter. For linear map profile, results are shown in Fig.4.6. For nonlinear linear map profile, results are shown in Fig.4.7. In this simulation part, the simulated measurements are both for linear case and nonlinear case, i.e. measurements are a smooth line for the linear case and are a smooth sine wave for the nonlinear case. All localisation results are shown in Fig.4.8 - Fig.4.19 and Table 4.1 shows what each figure tells.

In order to generate the DBA map (Fig.4.6a and Fig.4.7a) for the linear map model, 20 runs are simulated with noise in their motions and sensor observations. After the DBA map is achieved, both extended Kalman filter (EKF) and particle filter (PF) are performed to compute the location of the robot at each time sample (Fig.4.6b and Fig.4.7d). The simulation results of EKF locations and PF locations are shown in Fig.4.6c and Fig.4.7e. Compared with dead reckoning (DR) locations as a reference, both EKF locations and PF locations show much improved accuracy.

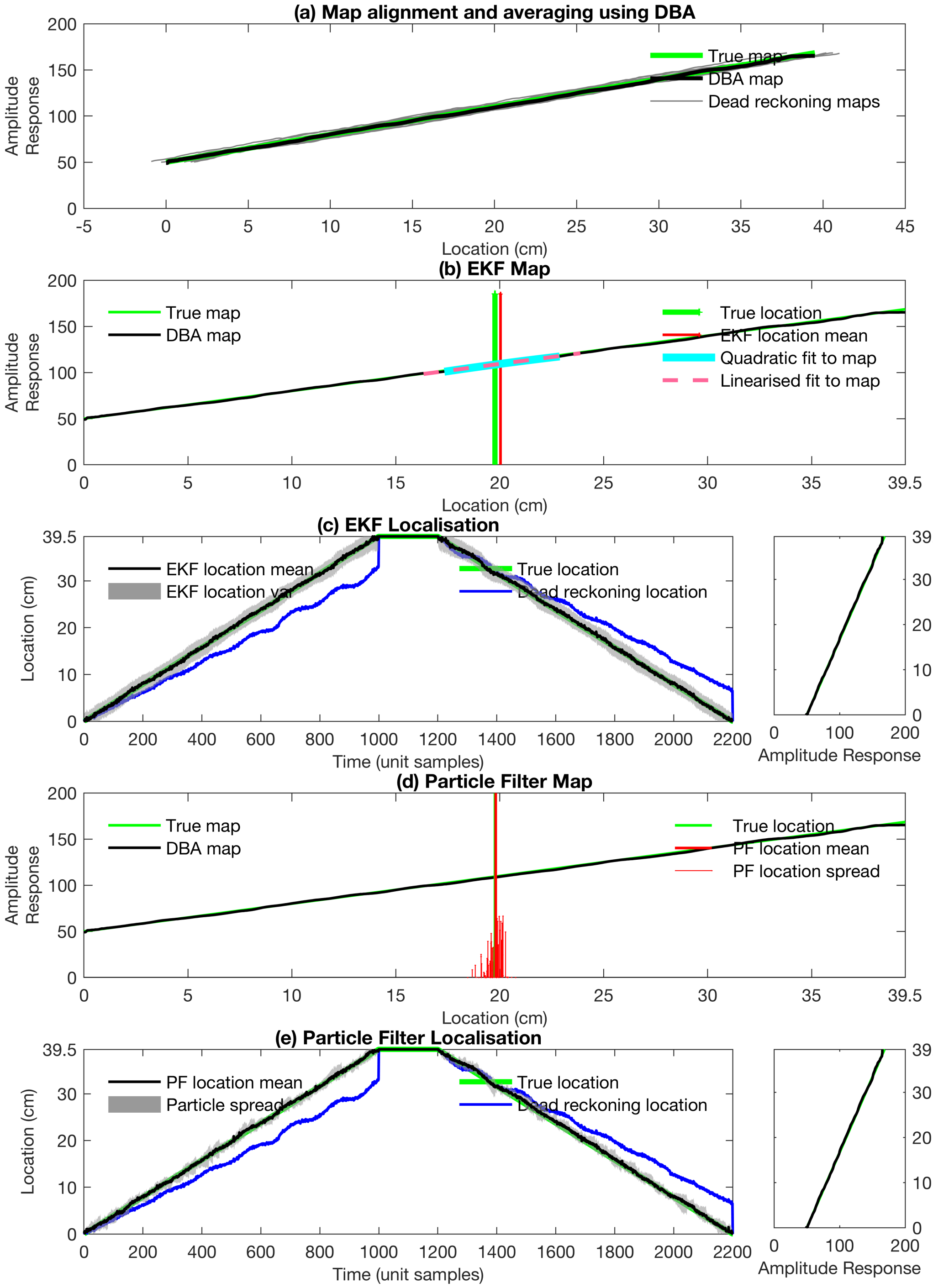


Figure 4.6: Localisation results using simulated linear data. (a) Mapping using DBA method, (b) and (c) Localisation using an extended Kalman filter (EKF). (d) and (e) Localisation using a particle filter (PF). For both the EKF and PF a comparison is given to dead reckoning, showing the clear improvement in localisation accuracy with EKF and PF.

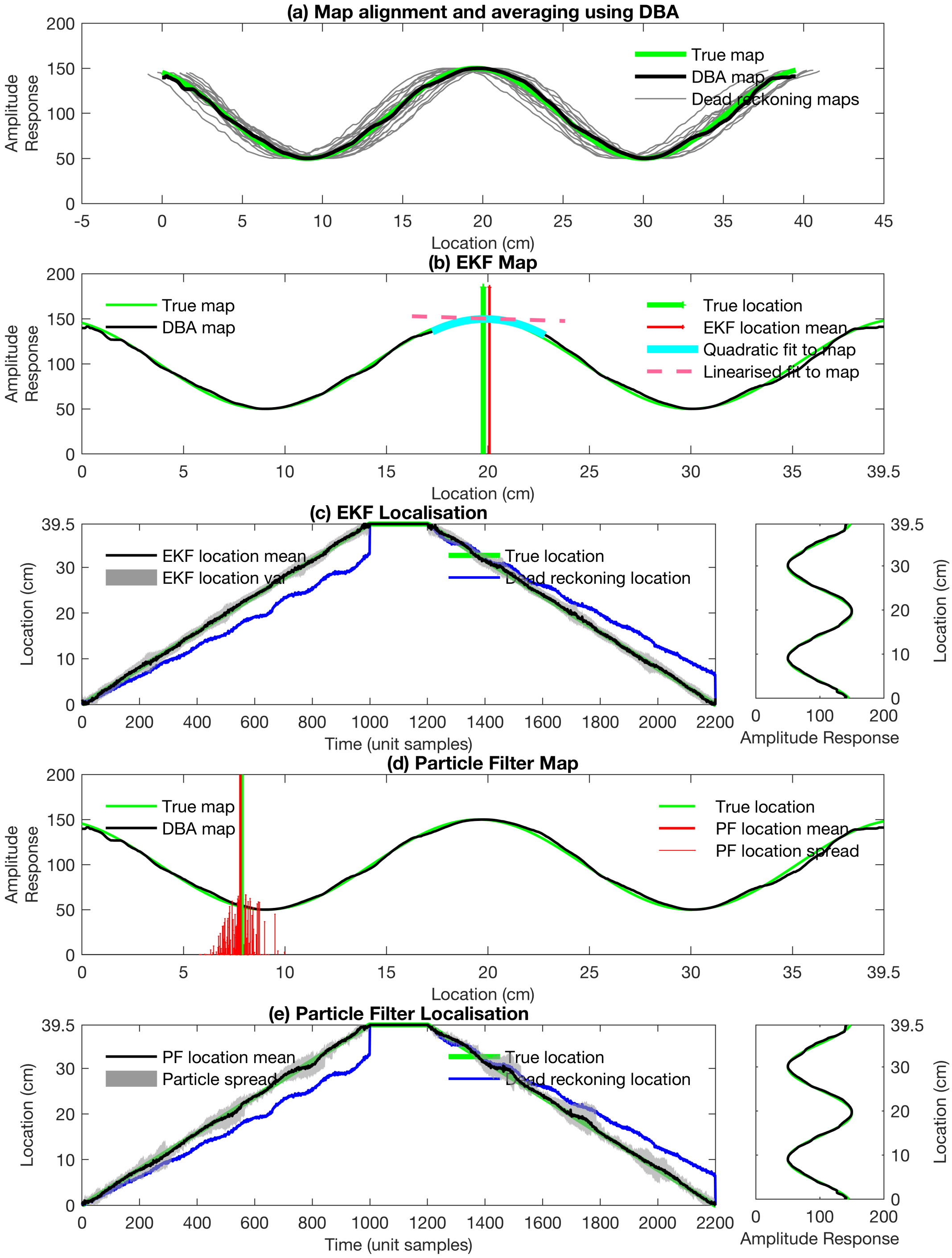


Figure 4.7: Localisation results using simulated nonlinear sine wave data. (a) Mapping using DBA method, (b) and (c) Localisation using an extended Kalman filter (EKF). (d) and (e) Localisation using a particle filter (PF). For both the EKF and PF a comparison is given to dead reckoning, showing the clear improvement in localisation accuracy with EKF and PF.

Table 4.1: List of figures showing localisation results (with Gaussian noise in localisation process) of different data using extended Kalman filter and particle filter. For example, the localisation result of non-linear simulation data using particle filter is shown in Fig.4.11.

Data	**Estimation Method**	
	Extended Kalman filter	Particle filter
Linear data	4.8	4.10
Non-Linear data	4.9	4.11
Plastic pipe data	4.13	4.14
Metal 1m pipe data	4.16	4.17
Metal 5m pipe Data I	4.18	4.19
Metal 5m pipe Data I (reverse)	4.20	4.21

In practical scenarios, the motor encoder (dead reckoning) and robot equipped sensor (ultrasound sensor or hydrophone) may have different accuracy levels. Due to potential motor gear slipping, water flow and tethered cable elasticity, drift noise is unavoidable in dead reckoning estimates. This drift can be very small and can be ignored in a short-length pipe but as the length of the pipe increases, the impact on robot location estimation can be huge. Therefore, to test if the algorithm is robust on this drift noise, additional drift is added to the motor encoder data along with white noise.

Various levels of white noise are also added into the robot equipped sensor (ultrasound sensor or hydrophone) data. The magnitudes of additional added white noise are measured using signal-to-noise ratio (*snr*) in dB, where

$$snr = 10\log_{10}\left(\frac{P_{signal}}{P_{noise}}\right) \tag{4.20}$$

snr is defined as the ratio of the power of a signal and the power of unwanted environment noise. Here the environment noise is assumed to be Gaussian noise with zero mean and different covariance. The additional white noise are added in different *snr* ratios: 15dB, 20dB, 25dB, 30dB, 35dB, 40dB, 45dB and 50dB. $snr = \infty$ indicates no additional white noise. To measure the error of the location estimate, normalised rooted mean square error (NRMSE) is used by replacing s with robot location x in (4.19).

$$NRMSE = \frac{\sqrt{\frac{\sum_{k=1}^{N}(\tilde{x}_k - x_k)^2}{N}}}{x_{max} - x_{min}} \tag{4.21}$$

where x_k is the true robot location at time k and $\tilde{x}_k$ is the location estimate. x_{max} is the furthest location robot travels and x_{min} is the nearest location. In this case, x_{min} and x_{max} are the locations of the two ends of the pipe. Assuming the robot travels from one end of the pipe to the other end of the pipe, then return the starting point, $x_{max} - x_{min}$ indicates the length of the pipe. Since the starting location is assumed to be 0, x_{min} is set to be 0.

Fig.4.8 $\sim$ Fig.4.11 show the effect of the additional white noise added into the simulated sensor measurements. For EKF localisation using linear simulation data shown in Fig.4.8, the increase of the amplitude of additional white noise will make the location estimation covariance larger. For $snr = \infty$ dB, the location estimation will accurately track the simulated line. Purple dots are the simulated observations along the black line (DBA map). The smaller the snr ratio, the wider those purple dots spread. The bold dark purple line is the quadratic fit of the local map. Since the DBA map is a straight line, the quadratic fit looks straight in this case. The dashed red line is the derivative of the quadratic fit at that EKF location. From Fig.4.8b, Fig.4.8d and Fig.4.8e, EKF can estimate fair good locations with small white noise but the uncertainty will go larger with larger white noise. The grey shadow shows EKF covariance and is called the uncertainty region. It goes wider as the robot travels through the pipe in the forward pass. This is because when the measurement uncertainty is large (small snr), dead reckoning location estimates become the dominant part. Since the robot travels further before reaching to the next known location, the uncertainty in dead reckoning location estimates will accumulate and therefore increase. However, if the measurement data is noise free ($snr = \infty$ dB), the measurement data will become dominant in location estimation and therefore the uncertainty area will not increase as the robot travels (see Fig.4.8a and Fig.4.8b). Therefore, EKF locations are more accurate for noise-free measurements (EKF NRMSE $= 0.00681866 = 0.6819\%$), compared with $snr = 15$ dB measurements (EKF NRMSE $= 0.0282015 = 2.8202\%$). But they all outperform dead reckoning (DR) estimates (DR NRMSE $= 0.1126938 = 11.26938\%$).

For a nonlinear sinusoidal map, the dark purple quadratic fit can fit the sinusoidal map very well in a small region. EKF location estimation for this sinusoidal map can still get good estimation results – estimated locations can track the true locations when the robot travels through the pipe. Similar to the linear map, when

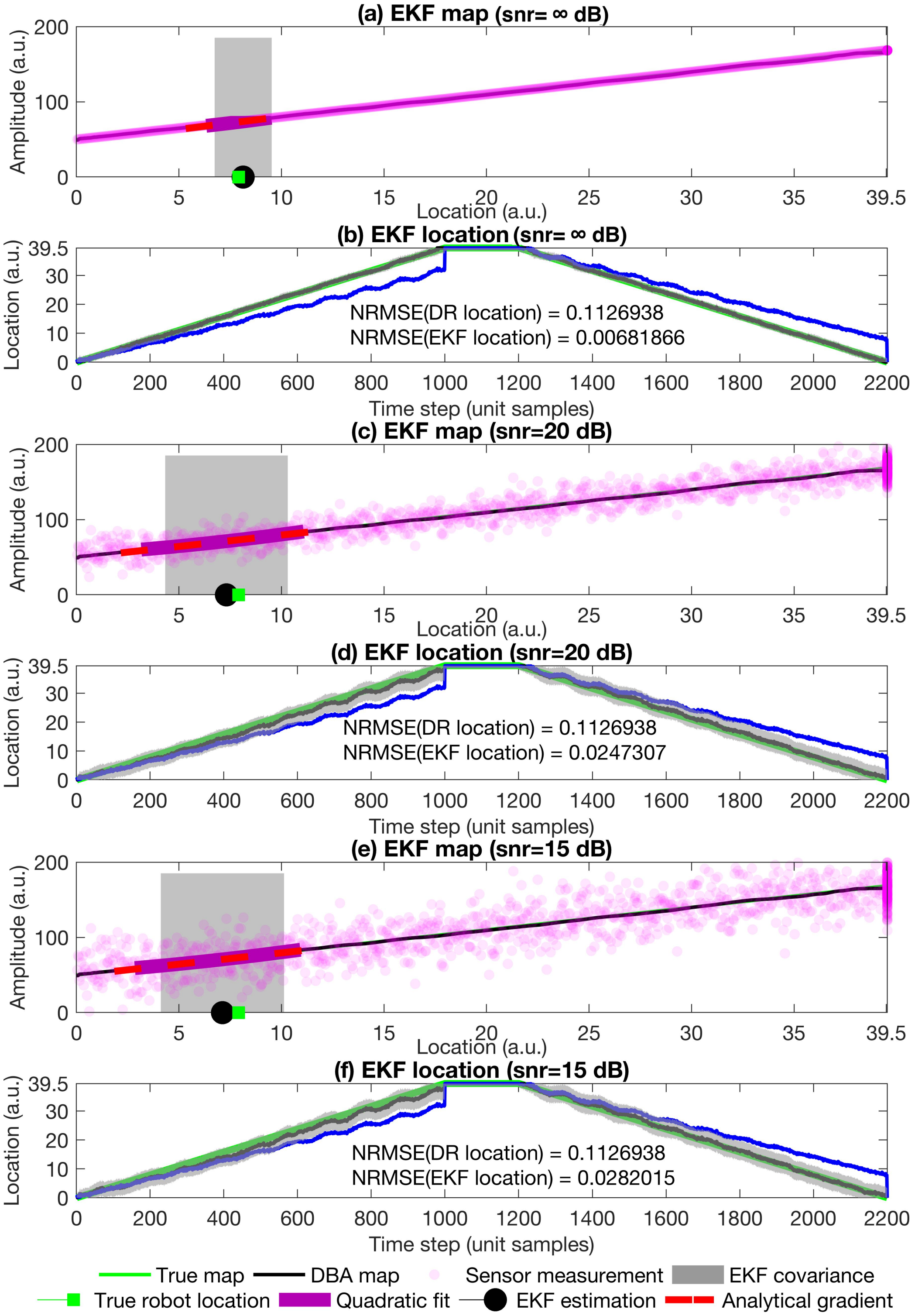


Figure 4.8: EKF location estimation in linear (straight line) observations. (a)-(b) Localisation with additional sensor noise $snr = \infty$ dB (noise free), (c)-(d) Localisation with additional sensor noise $snr = 20$ dB, (e)-(f) Localisation with additional sensor noise $snr = 15$ dB.

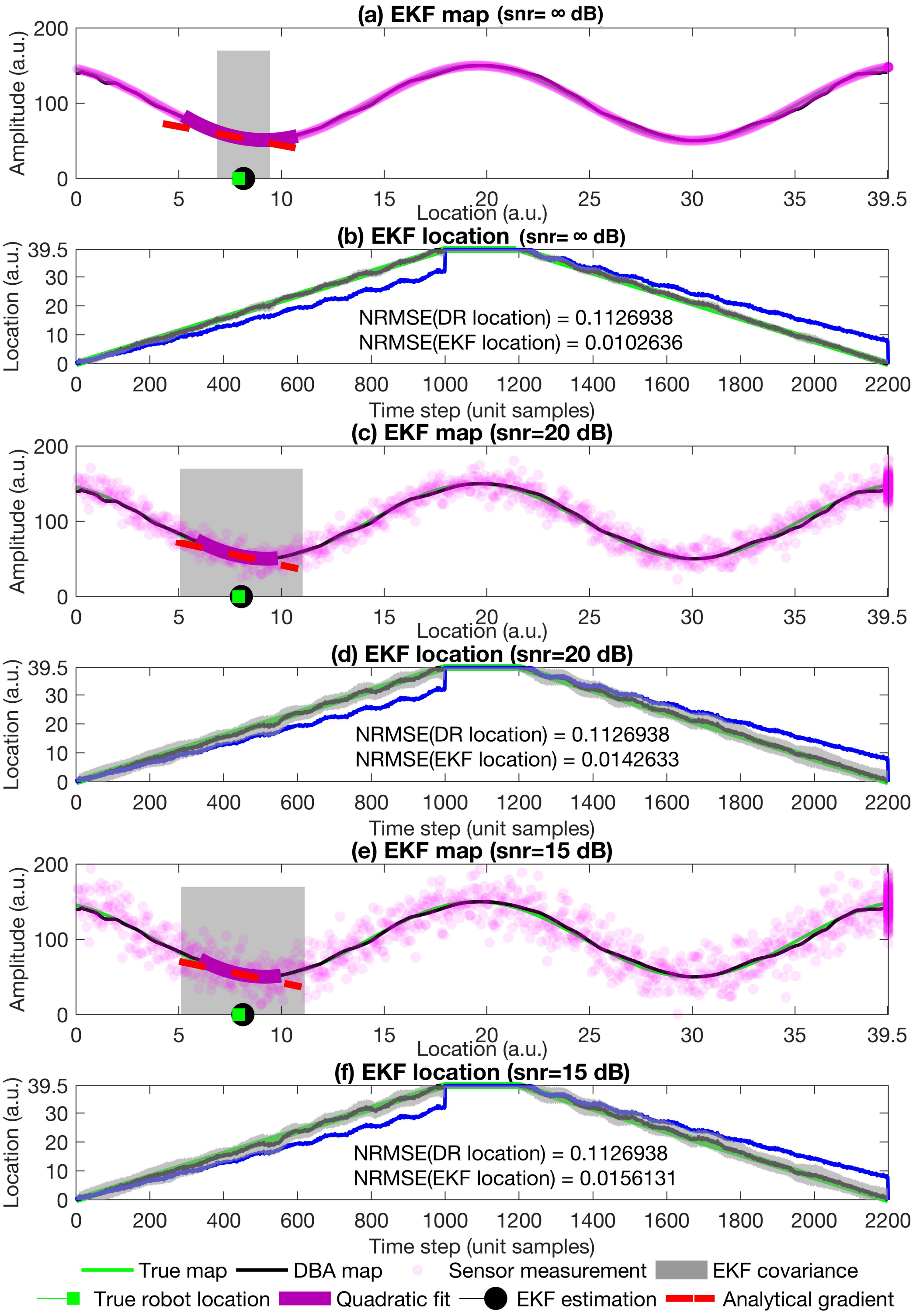


Figure 4.9: EKF location estimation in nonlinear (sine wave) observations. (a)-(b) Localisation with additional sensor noise $snr = \infty$ dB (noise free), (c)-(d) Localisation with additional sensor noise $snr = 20$ dB, (e)-(f) Localisation with additional sensor noise $snr = 15$ dB.

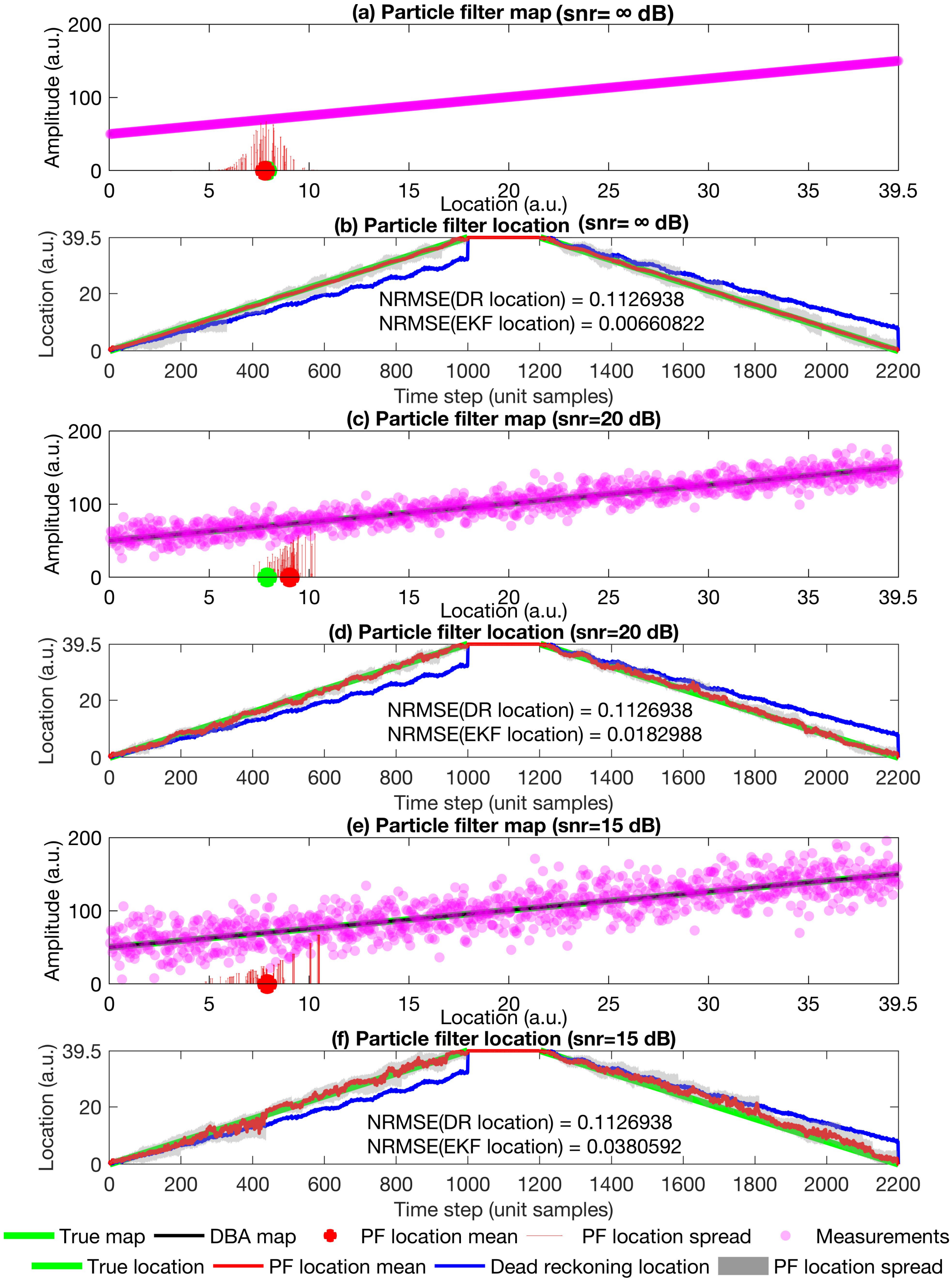


Figure 4.10: PF location estimation in linear (straight line) observations. (a)-(b) Localisation with additional sensor noise $snr = \infty$ dB (noise free), (c)-(d) Localisation with additional sensor noise $snr = 20$ dB, (e)-(f) Localisation with additional sensor noise $snr = 15$ dB.

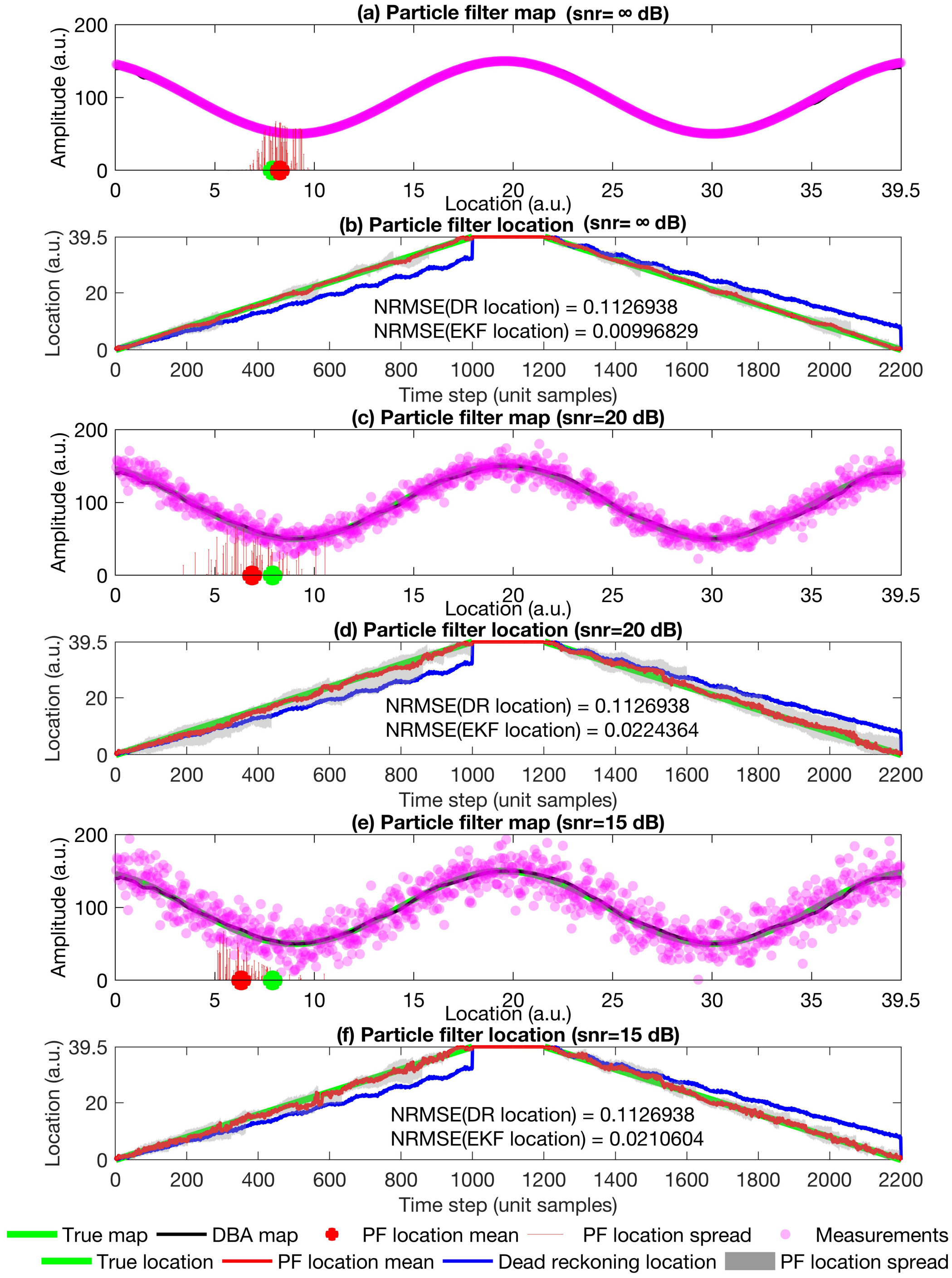


Figure 4.11: PF location estimation in nonlinear (sine wave) observations. (a)-(b) Localisation with additional sensor noise $snr = \infty$ dB (noise free), (c)-(d) Localisation with additional sensor noise $snr = 20$ dB, (e)-(f) Localisation with additional sensor noise $snr = 15$ dB.

the measurement is noise free ($snr = \infty$ dB), the EKF location estimates are very accurate with small error (EKF NRMSE = 0.012636 = 1.2636%), compared with large dead reckoning location error (DR NRMSE = 0.1126938 = 11.26938%). For larger measurement errors ($snr = 15$ dB), the location error in EKF estimates is slightly bigger (EKF NRMSE = 0.0156131 = 1.5613%). Therefore, for a nonlinear sinusoidal map, EKF location estimation still outperforms dead reckoning locations.

Both linear and nonlinear simulation data are also estimated using particle filter (PF) with different level of white noise (see Fig.4.10 and Fig.4.11). Similar to the linear case, particle filter gets better location estimates with smaller white noise. When measurements are noise free ($snr = \infty$ dB), the PF location error is small (PF NRMSE = 0.0066082 = 0.6608%), with same DR location error (DR NRMSE = 0.1126938 = 11.26938%). Compared with EKF estimation, PF estimation slightly outperforms with accuracy increased by 0.0211%. But since both estimation is very accurate, the difference can be ignored. However, when noise increases, the PF location error increases quicker then EKF location error. When $snr = 15$ dB in linear data, EKF location error is 2.8202% while PF location error is 3.8059%. One reason can be the insufficient number of particles which means $n_s = 100$ particles are not large enough.

4.3.3 Localisation Results from Plastic Water Pipe Data

Algorithm evaluation results are also compared in plastic pipes using EKF and particle filter. For plastic pipes, results are shown in Fig.4.12 $\sim$ Fig.4.14.

In order to generate the DBA map (Fig.4.12a and Fig.4.15a) for the plastic pipes and metal pipes, 20 runs are simulated with noise in their motions and sensor observations. After the DBA map is achieved, both EKF and PF are performed to compute the location of the robot at each time sample (Fig.4.12b and Fig.4.15d). The comparison of EKF locations and PF locations are shown in Fig.4.12c and Fig.4.15e. Compared with dead reckoning (DR) locations as a reference, both EKF locations and PF locations show much improved accuracy.

4.3.4 Localisation Results from Metal Water Pipe Data

For metal pipes, EKF estimation and PF estimation results are shown in Fig.4.15-4.19. Two experiment data sets are tested: 1-metre metal pipe (Fig.4.16 and Fig.4.17), 5-metre metal pipe Data I (Fig.4.18 and Fig.4.19)

In order to evaluate the mapping and localisation algorithms described above we used the experimental data to define a ground truth map. We then simulated

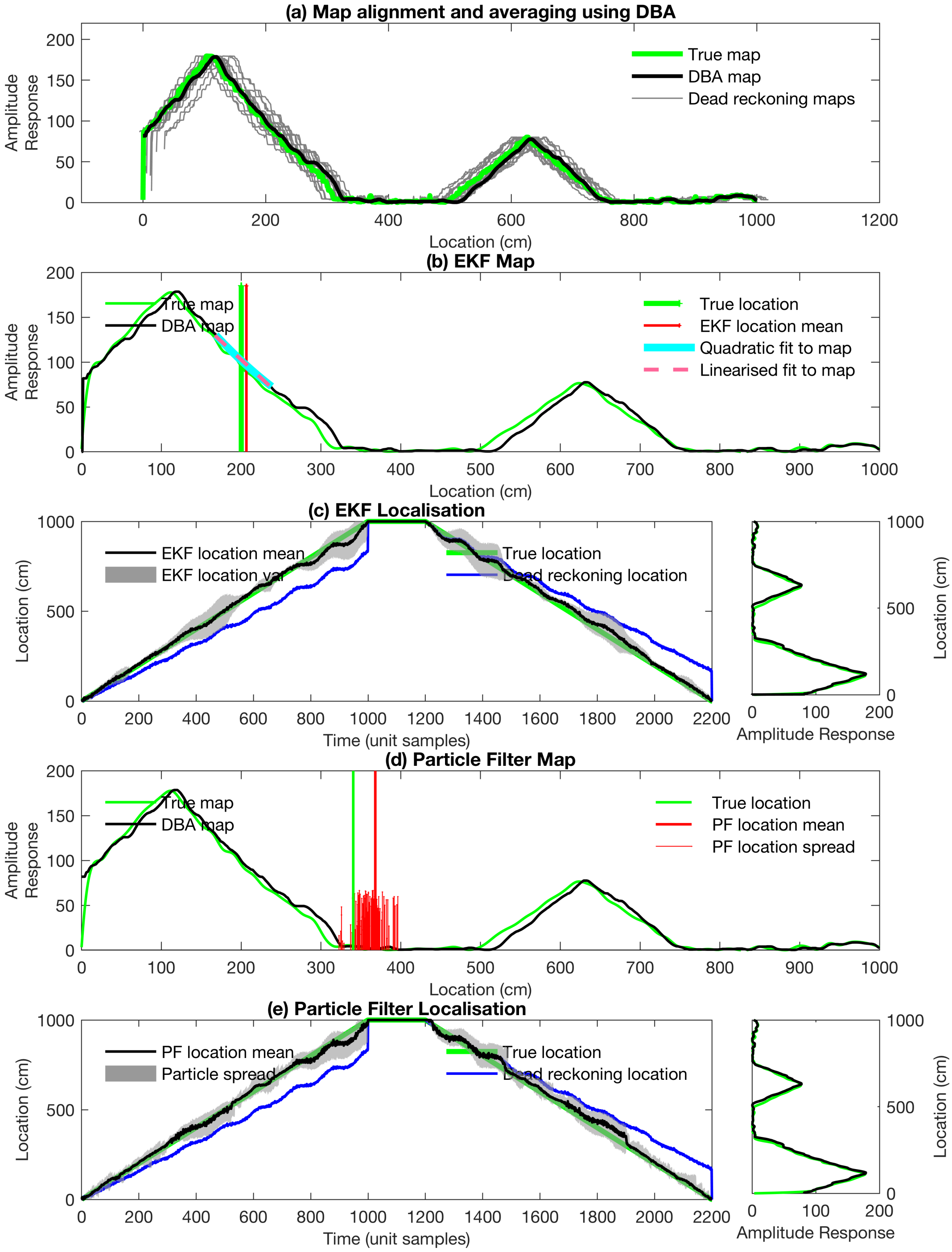


Figure 4.12: Sequential mapping and localisation approach in plastic water pipes. (a) The use of DBA to construct a map estimate from observations with simulated drift. (b) and (c) Localisation using an extended Kalman filter (EKF). (d) and (e) Localisation using a particle filter (PF). For both the EKF and PF a comparison is given to dead reckoning, showing the clear improvement in localisation accuracy with EKF and PF.

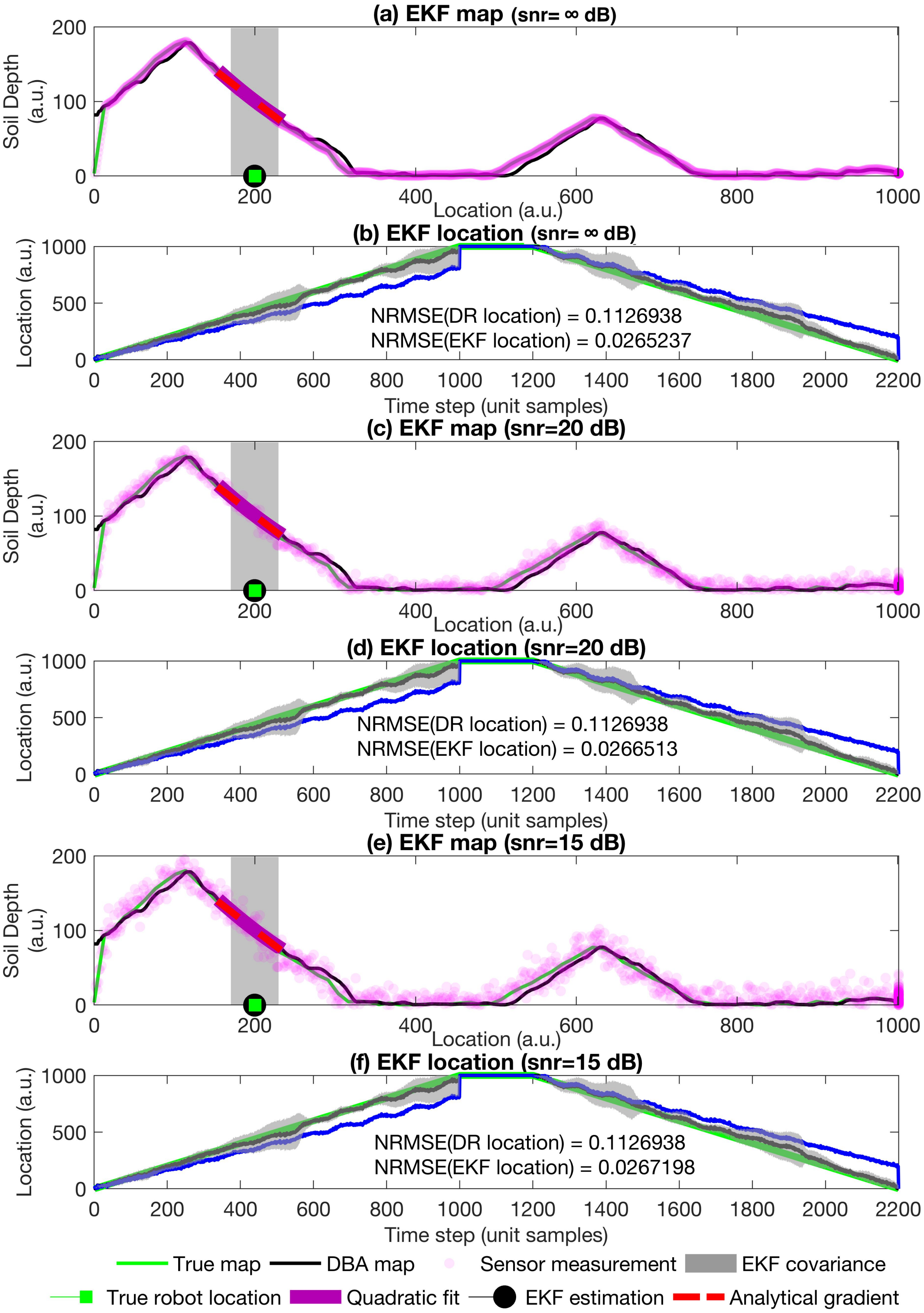


Figure 4.13: EKF localisation approach in plastic water pipes with different level of white noise. (a)-(b) Localisation with additional sensor noise $snr = \infty$ dB (noise free), (c)-(d) Localisation with additional sensor noise $snr = 20$ dB, (e)-(f) Localisation with additional sensor noise $snr = 15$ dB.

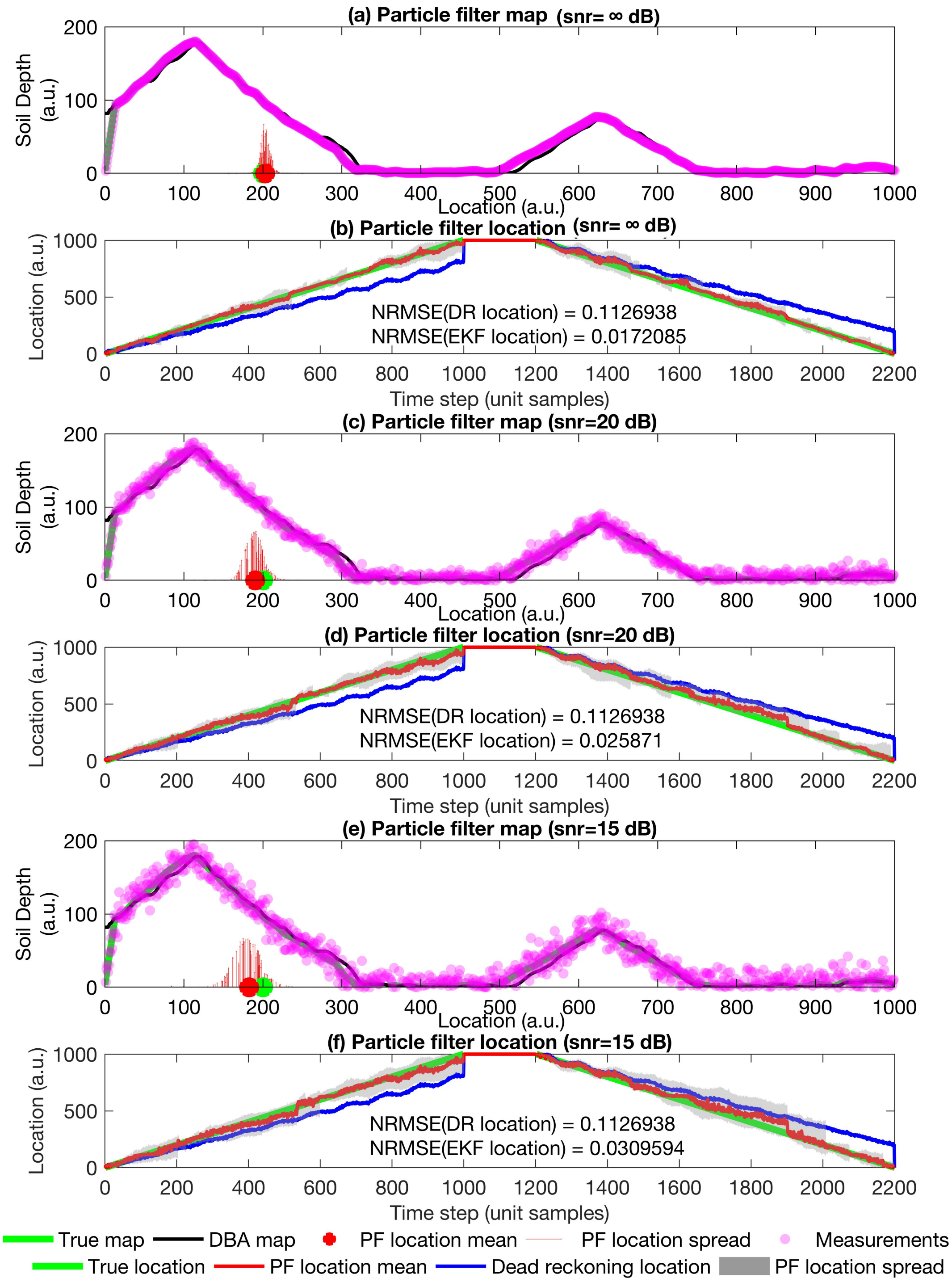


Figure 4.14: PF localisation approach in plastic water pipes with different level of white noise. (a)-(b) Localisation with additional sensor noise $snr = \infty$ dB (noise free), (c)-(d) Localisation with additional sensor noise $snr = 20$ dB, (e)-(f) Localisation with additional sensor noise $snr = 15$ dB.

robot movement up and down this map, with simulated drift, to investigate the effectiveness of the DBA algorithm for constructing the map: we generated a total of 20 maps to align and average using DBA (Fig.4.15a). We then applied both the EKF and the PF localisation algorithms to this estimated map and found that the technique improved on using dead reckoning alone as expected (Fig.4.15b ∼ Fig.4.15e). Hence, both the EKF and PF greatly outperformed the localisation using dead reckoning alone. The EKF slightly outperformed the PF, which along with the efficiency of the EKF approach, makes the EKF more appealing in this application.

In addition, all experiments assume the robot is travelling from the left end to the right end through the length of the metal pipe. An additional evaluation is to use a reverse data set, i.e. by reconstructing data I to assume the robot is travelling from the right end to the left end and then return to the right end. In this case, the right end location is initialised to 0 cm. When the robot reaches the left end, it reaches the maximum distance travelled location (499.5 cm). Both EKF and PF localisation results are shown in Fig.4.20 ∼ Fig.4.21. These results give the first evidence that mapping and localisation using hydrophone induced vibration with map alignment using DBA is feasible, supporting field testing of the sensor on the robot prototype shown in Fig.3.3a. The technique for calibrating the spatial map is also extensible to other types of sensor that would produce similar map data, e.g. through-pipe-wall ultrasonics, proposed in our earlier work [96]. One appealing feature for the hydrophone method we test here, in comparison to the through-pipe-wall ultrasonic method, is that it is omnidirectional in nature, which should make it robust to any robot rotations. A limitation of the method is that it is only useful for correcting drift along the length of the pipe, i.e. distance travelled, not heading estimates. This we leave to future work, but envisage fusing the method with IMU data to solve this problem.

4.3.5 Localisation Results with Blockage Noise

Besides common Gaussian white noise, it is also important to evaluate both localisation algorithms in dealing with blockage noise. There can be many reasons for a robot to get stuck when it travels through a pipe. It may be blocked by some unknown obstacles inside the pipe or facing a strong water flow. This is not only relevant to an in-pipe robot but also to other robot applications such as a wheeled mobile robot, which may get stuck on a slippery surface, e.g. sand. In that case, the spinning wheels imply that the robot is moving, but it may actually stay in the same place.

To represent this scenario, the simulation model of the robot movement is

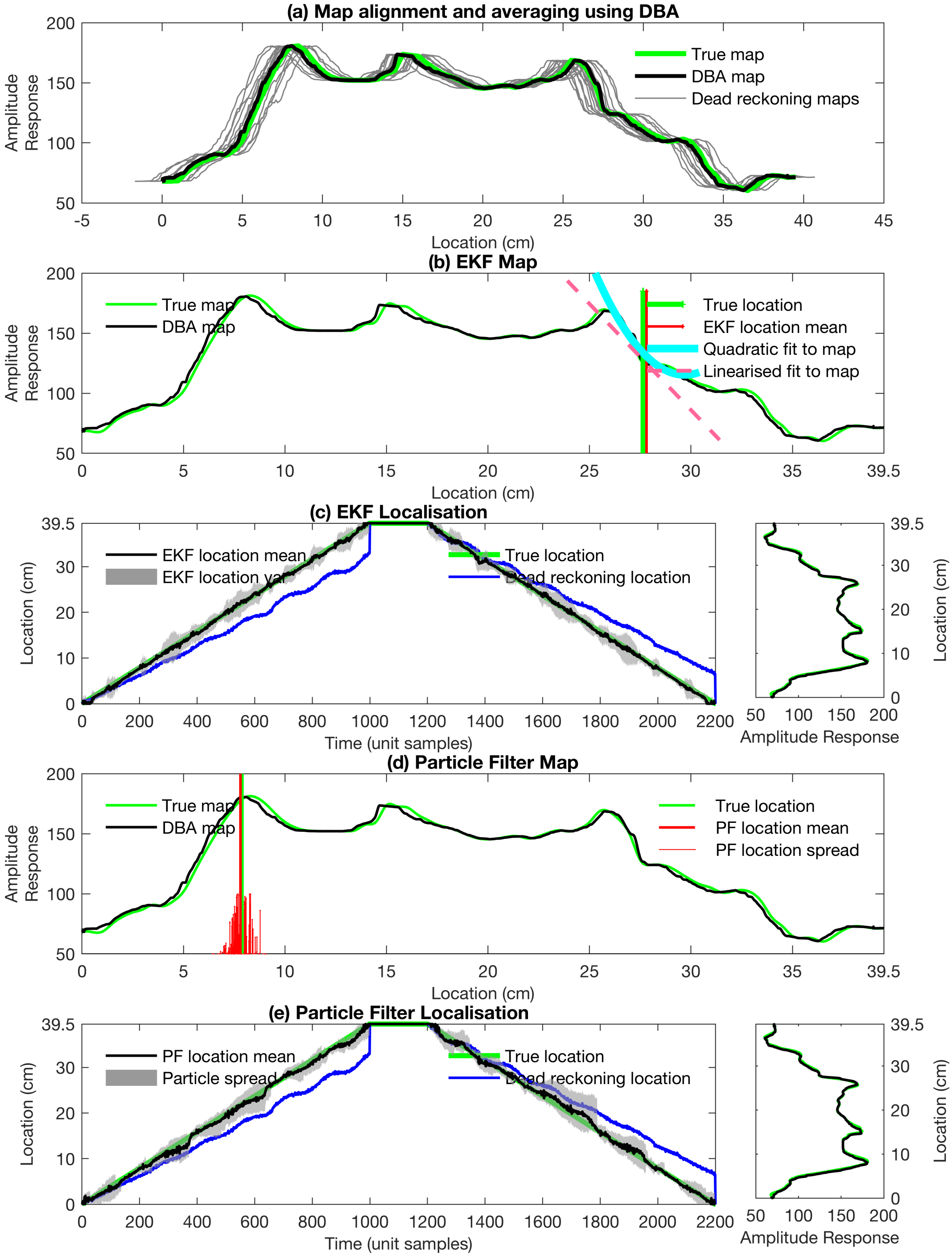


Figure 4.15: Sequential mapping and localisation approach in 1m metal water pipe. (a) The use of DBA to construct a map estimate from observations with simulated drift. (b) and (c) Localisation using an extended Kalman filter (EKF). (d) and (e) Localisation using a particle filter (PF). For both the EKF and PF a comparison is given to dead reckoning, showing the clear improvement in localisation accuracy with EKF and PF.

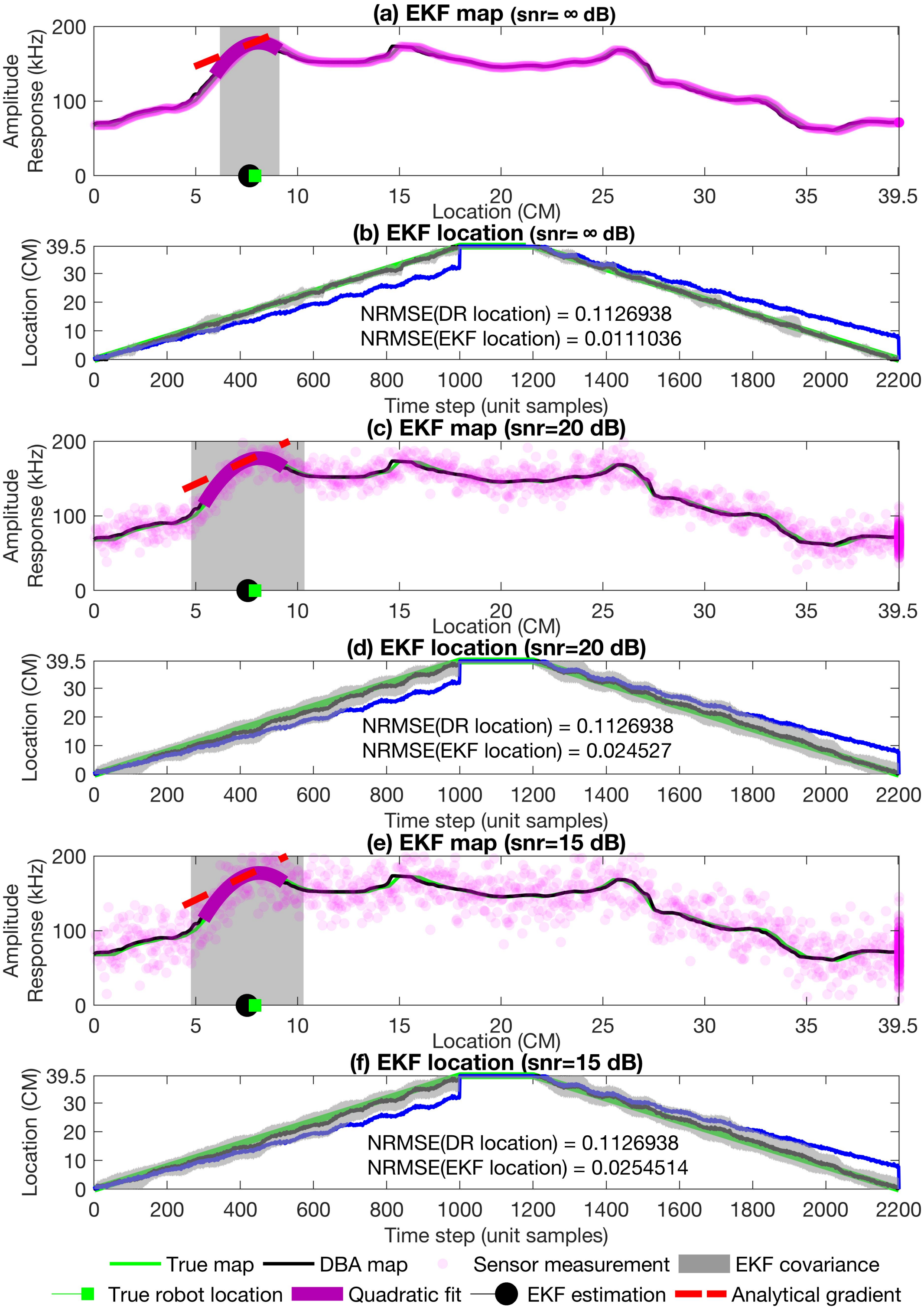


Figure 4.16: EKF localisation approach in 1m metal water pipe. (a)-(b) Localisation with additional sensor noise $snr = \infty$ dB (noise free), (c)-(d) Localisation with additional sensor noise $snr = 20$ dB, (e)-(f) Localisation with additional sensor noise $snr = 15$ dB.

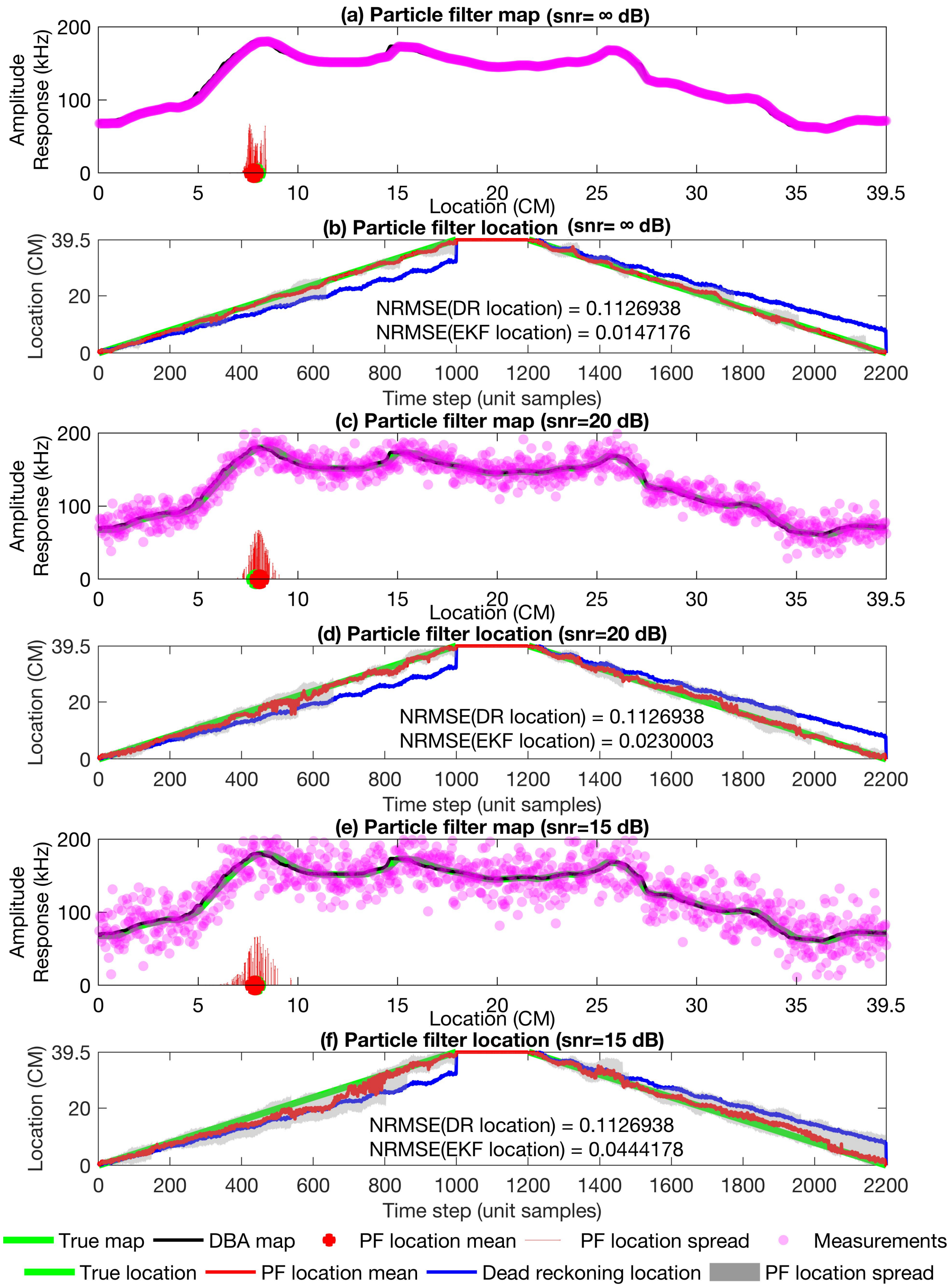


Figure 4.17: PF localisation approach in 1m metal water pipe. (a)-(b) Localisation with additional sensor noise $snr = \infty$ dB (noise free), (c)-(d) Localisation with additional sensor noise $snr = 20$ dB, (e)-(f) Localisation with additional sensor noise $snr = 15$ dB.

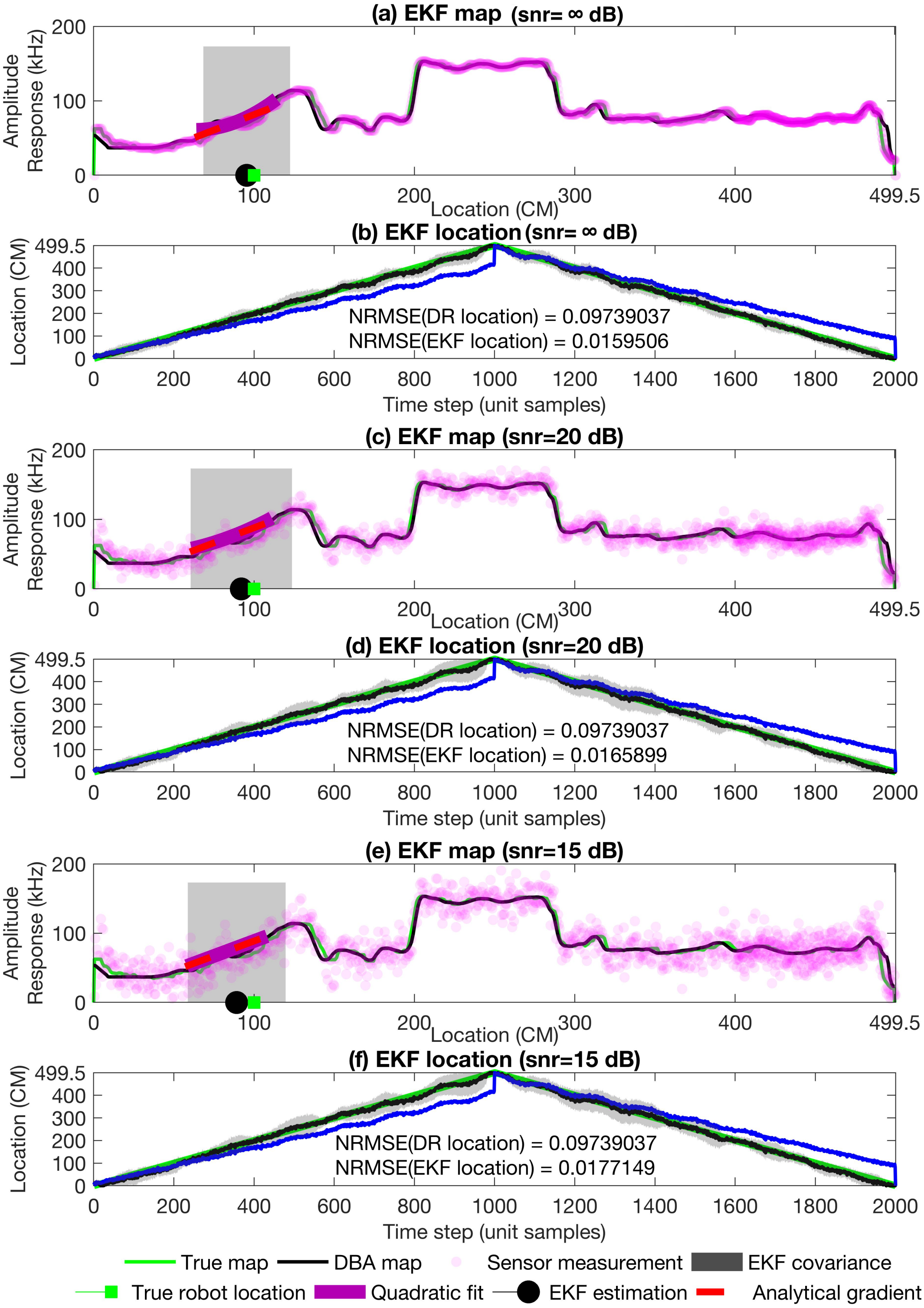


Figure 4.18: EKF localisation approach in 5m metal water pipe data I. (a)-(b) Localisation with additional sensor noise $snr = \infty$ dB (noise free), (c)-(d) Localisation with additional sensor noise $snr = 20$ dB, (e)-(f) Localisation with additional sensor noise $snr = 15$ dB.

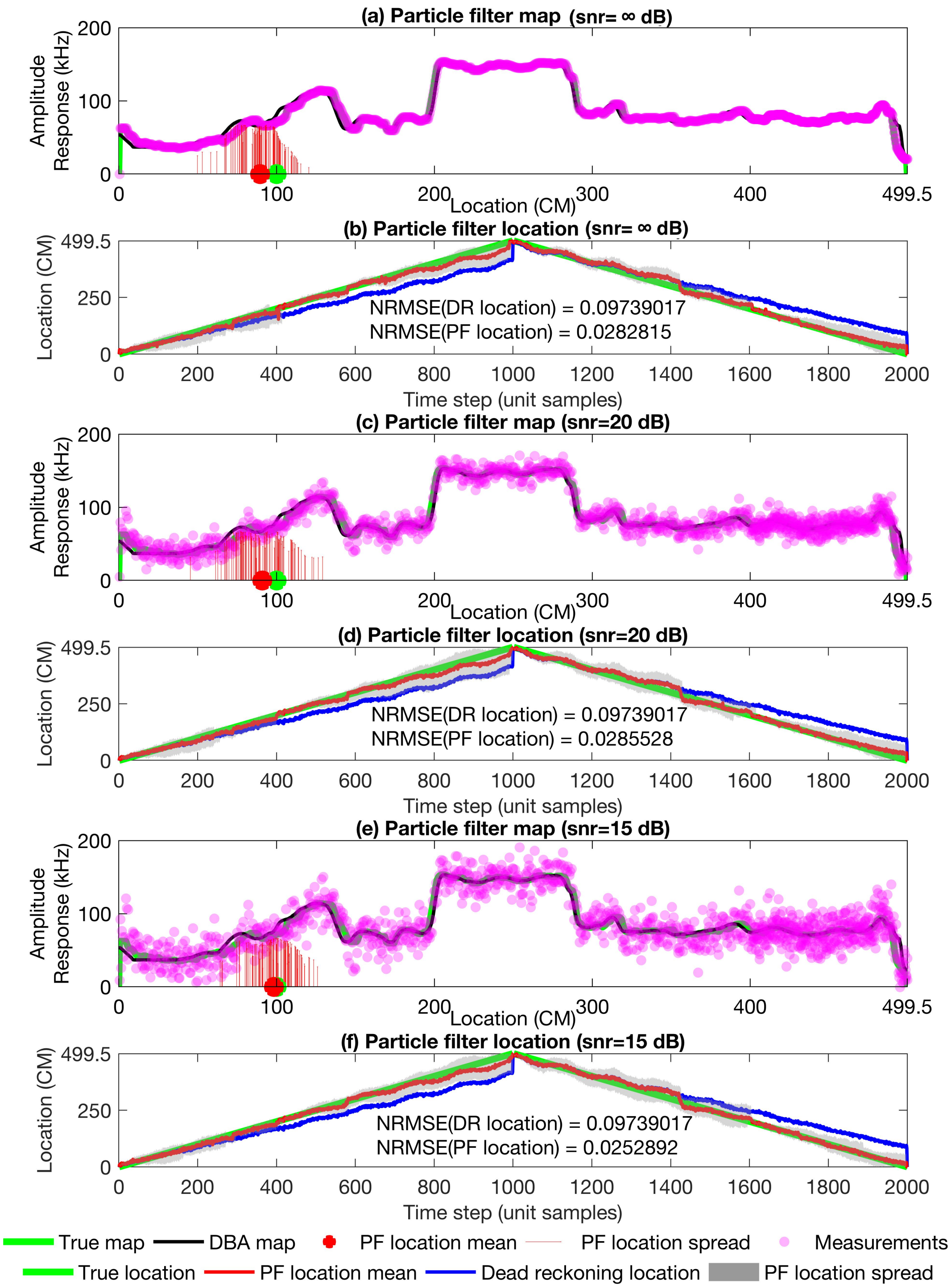


Figure 4.19: PF localisation approach in 5m metal water pipe data I. (a)-(b) Localisation with additional sensor noise $snr = \infty$ dB (noise free), (c)-(d) Localisation with additional sensor noise $snr = 20$ dB, (e)-(f) Localisation with additional sensor noise $snr = 15$ dB.

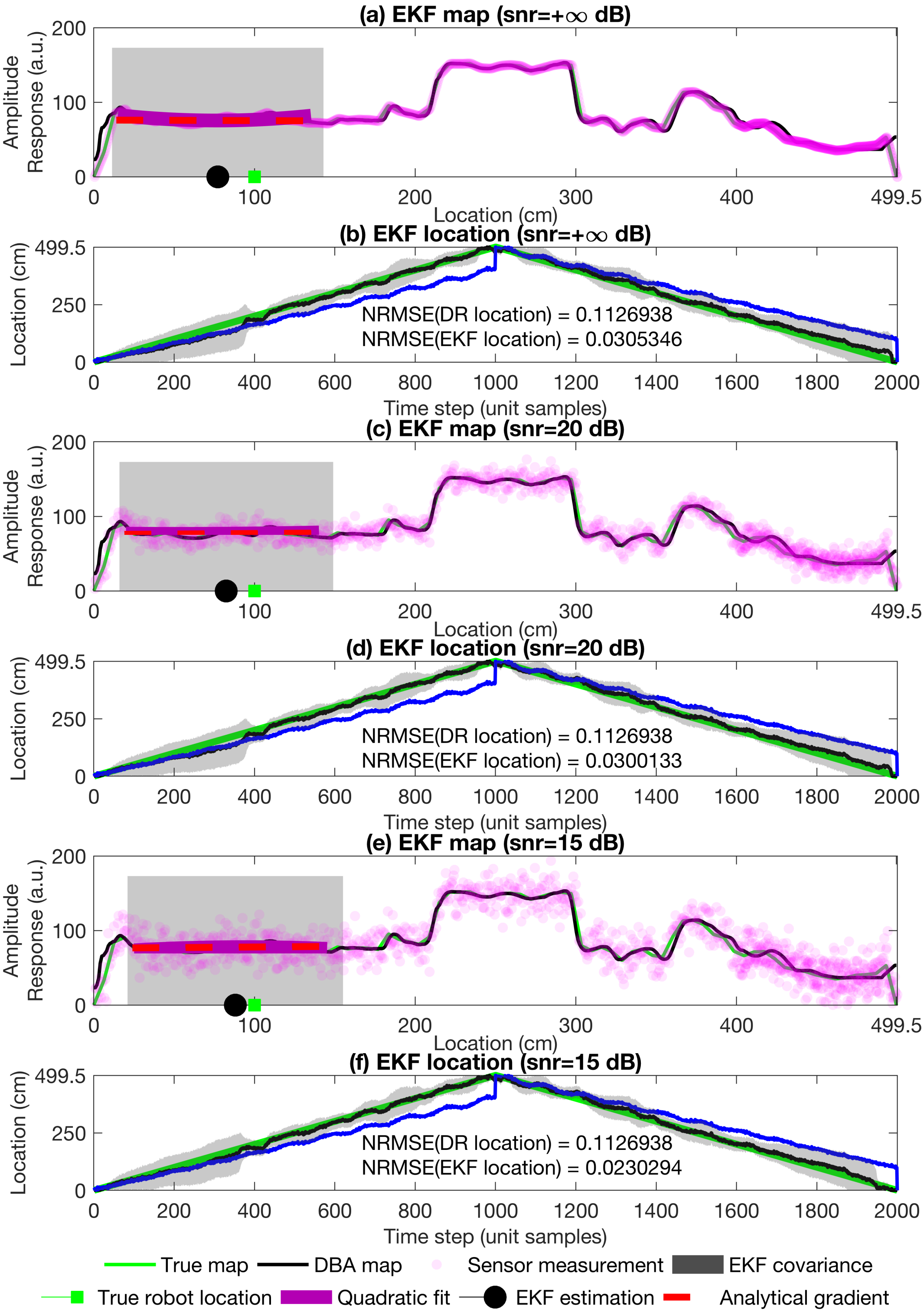


Figure 4.20: EKF localisation approach in 5m metal water pipe data I (reverse path). (a)-(b) Localisation with additional sensor noise $snr = \infty$ dB (noise free), (c)-(d) Localisation with additional sensor noise $snr = 20$ dB, (e)-(f) Localisation with additional sensor noise $snr = 15$ dB.

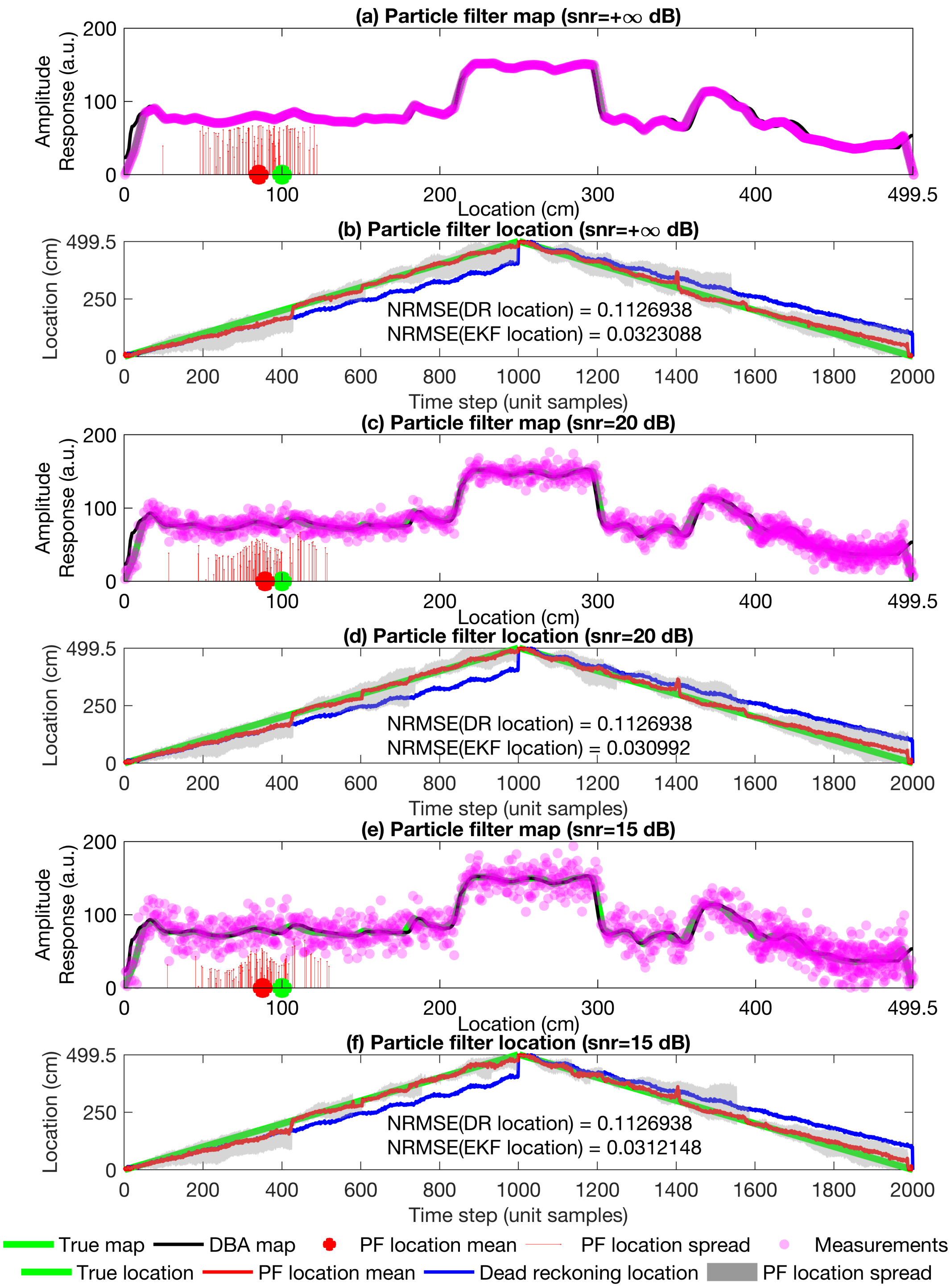


Figure 4.21: PF localisation approach in 5m metal water pipe data I (reverse path). (a)-(b) Localisation with additional sensor noise $snr = \infty$ dB (noise free), (c)-(d) Localisation with additional sensor noise $snr = 20$ dB, (e)-(f) Localisation with additional sensor noise $snr = 15$ dB.

Table 4.2: List of figures showing localisation results (with Gaussian noise and additional blockage noise in localisation process) of different data using extended Kalman filter and particle filter. For example, the localisation result of plastic pipe data (with additional blockage noise) using particle filter is shown in Fig.4.24.

Data	**Estimation Method**	
	Extended Kalman filter	Particle filter
Plastic pipe data	4.22	4.24
Metal 1m pipe data	4.23	4.25

modified to

$$\mathbf{x}_k = \mathbf{F}\mathbf{x}_{k-1} + \mathbf{G}\mathbf{u}_{k-1} + \mathbf{b}_{k-1} \tag{4.22}$$

where the blockage noise term $\mathbf{b}_{k-1} = -\mathbf{u}_{k-1}$ cancels the movement of the robot due to the external input, so the robot remains stuck in place, i.e. $\mathbf{x}_k = \mathbf{x}_{k-1}$. However, note that the state-space model used in the localisation algorithm to represent robot motion remains as previously defined because the presence of a blockage would be unknown.

To test if EKF and PF is capable of dealing with such scenarios, additional blockage noise is added to the experiment data to pretend the robot is stuck for a short period of time during its movement. Results are shown in Fig.4.23 $\sim$ Fig.4.25 and Table 4.2 shows what individual figure tells. To test the blockage noise, data gathered from 1m metal pipe and plastic pipe are used. The maps used for localisation are generated from previous DBA approach. However, it is worth mentioning that the blockage noise is only added into the localisation process not in the mapping process – mapping using DBA does not include blockage noise. For comparison, blockage noise is added into robot forward movement, and the backward movement will not contain any drift noise or blockage noise in motor encoder. But additional white noise will be added into ultrasound sensor and hydrophone measurements.

With the additional blockage noise, although the model remains the same, but some simulation parameters need to be modified. The error covariance in state update equation $\mathbf{Q}_k$ is increased while $\mathbf{R}_k$ remains the same. For different simulation data and experiment data, $\mathbf{Q}_k$ is increased differently.

From Fig.4.23, it can be seen that without additional white noise in hydrophone measurements, the EKF estimation can track the true robot location very well with small error (NRMSE = 0.9776%). Even if the additional white noise in hydrophone

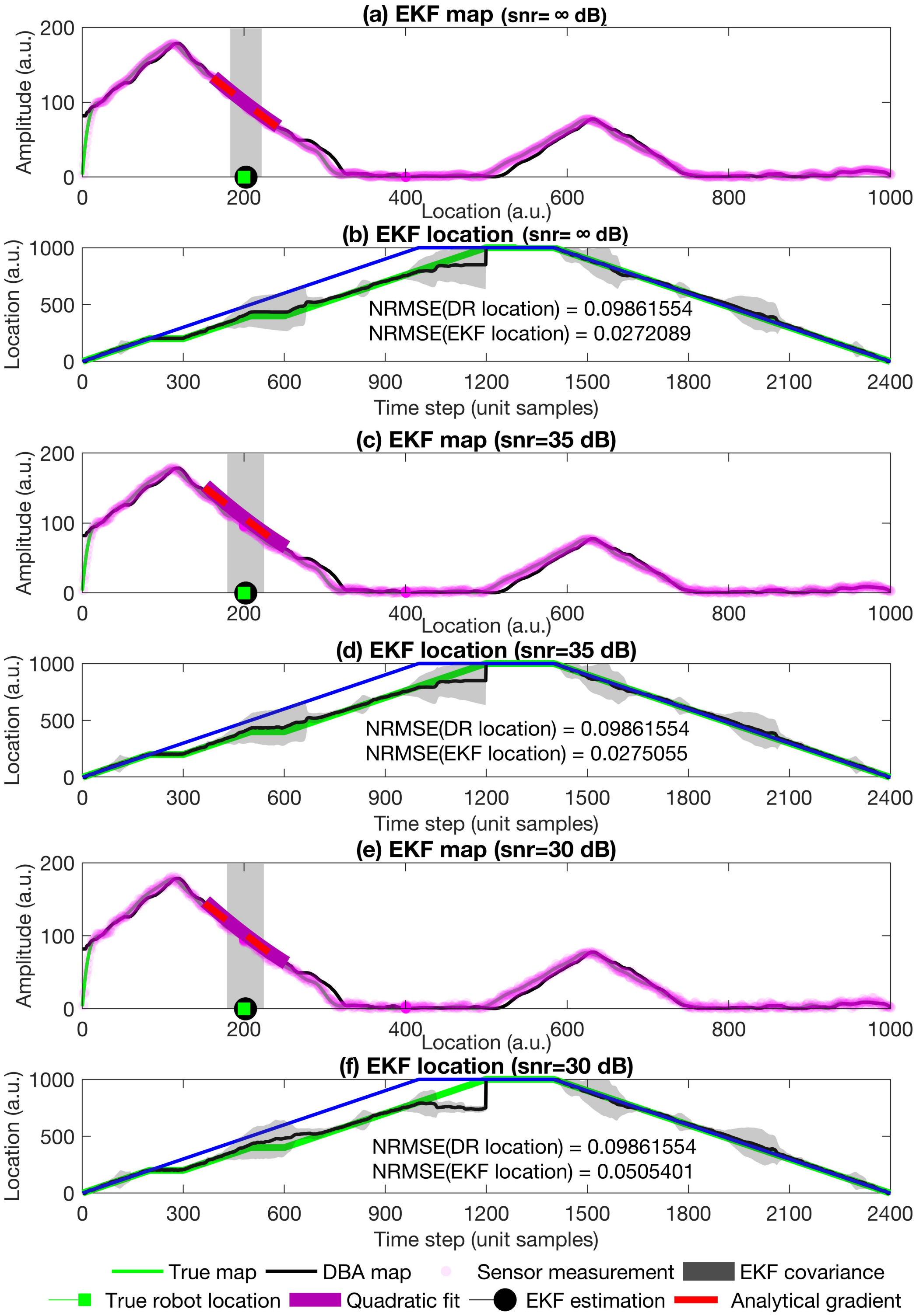


Figure 4.22: EKF localisation approach in plastic water pipe with blockage noise in motor encoder. (a)-(b) Localisation with additional sensor noise $snr = \infty$ dB (noise free), (c)-(d) Localisation with additional sensor noise $snr = 35$ dB, (e)-(f) Localisation with additional sensor noise $snr = 30$ dB.

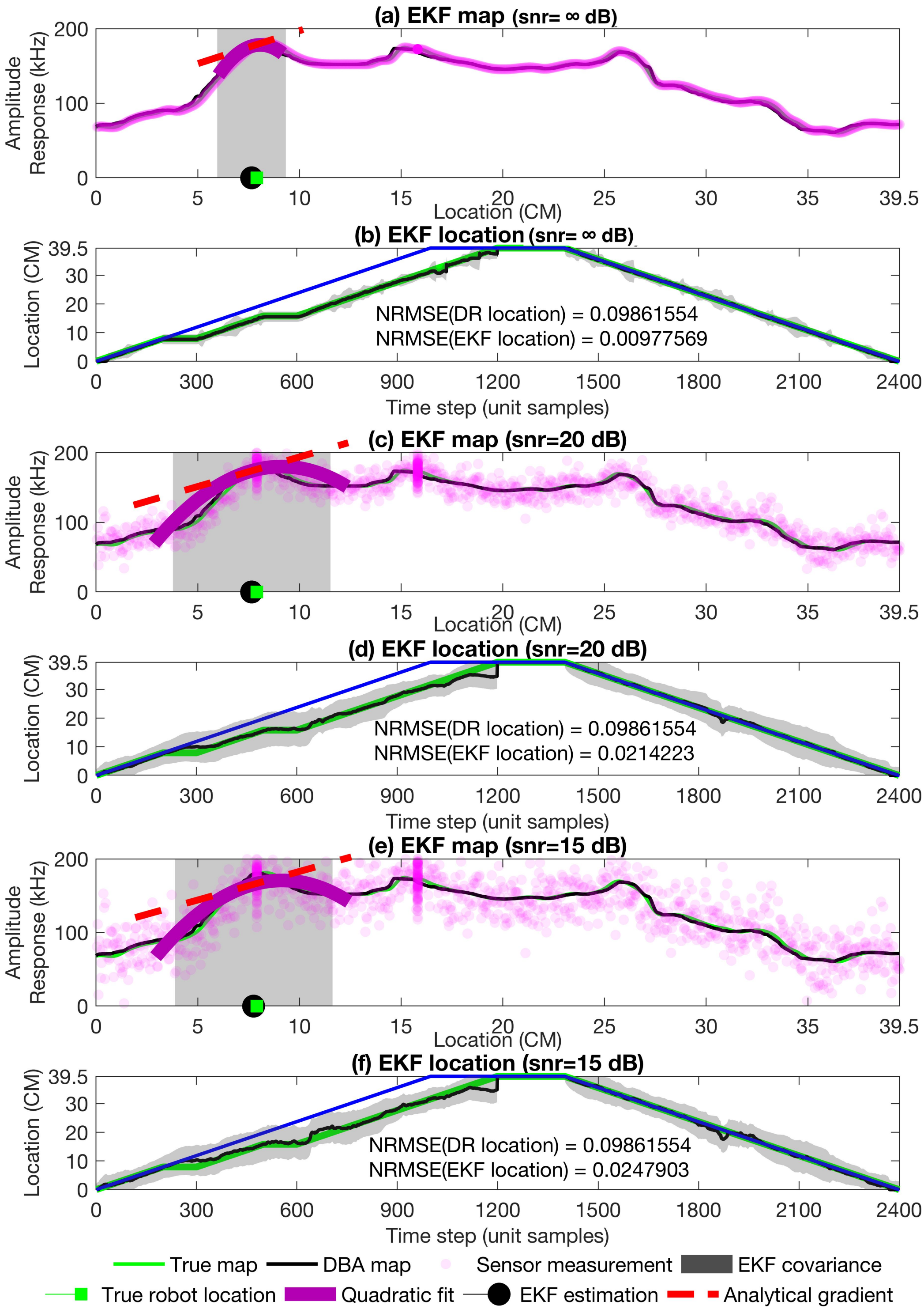


Figure 4.23: EKF localisation approach in 1m metal water pipe with blockage noise in motor encoder. (a)-(b) Localisation with additional sensor noise $snr = \infty$ dB (noise free), (c)-(d) Localisation with additional sensor noise $snr = 20$ dB, (e)-(f) Localisation with additional sensor noise $snr = 15$ dB.

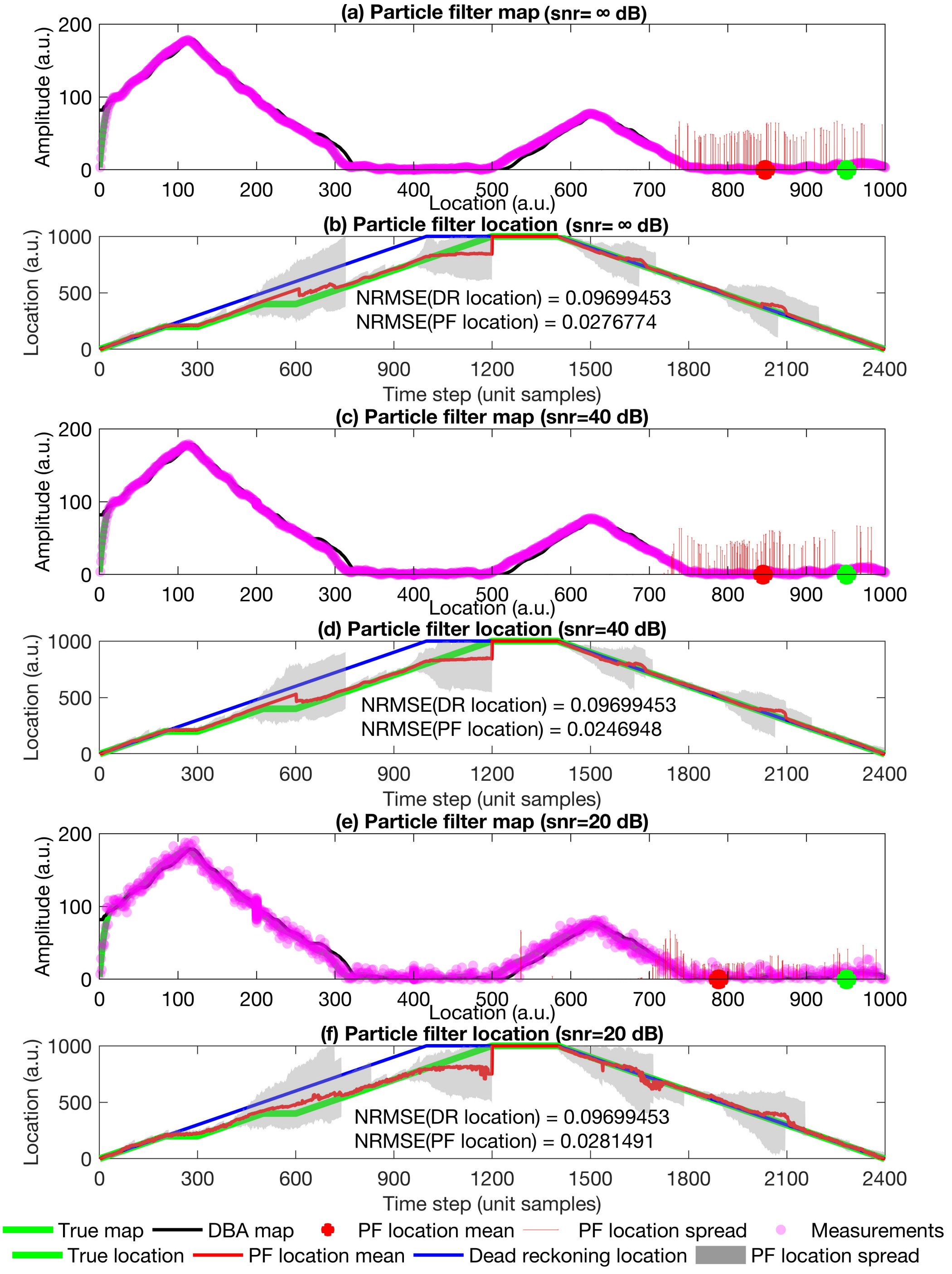


Figure 4.24: PF localisation in plastic water pipe with blockage noise in motor encoder. (a)-(b) Localisation with additional sensor noise $snr = \infty$ dB (noise free), (c)-(d) Localisation with additional sensor noise $snr = 40$ dB, (e)-(f) Localisation with additional sensor noise $snr = 20$ dB.

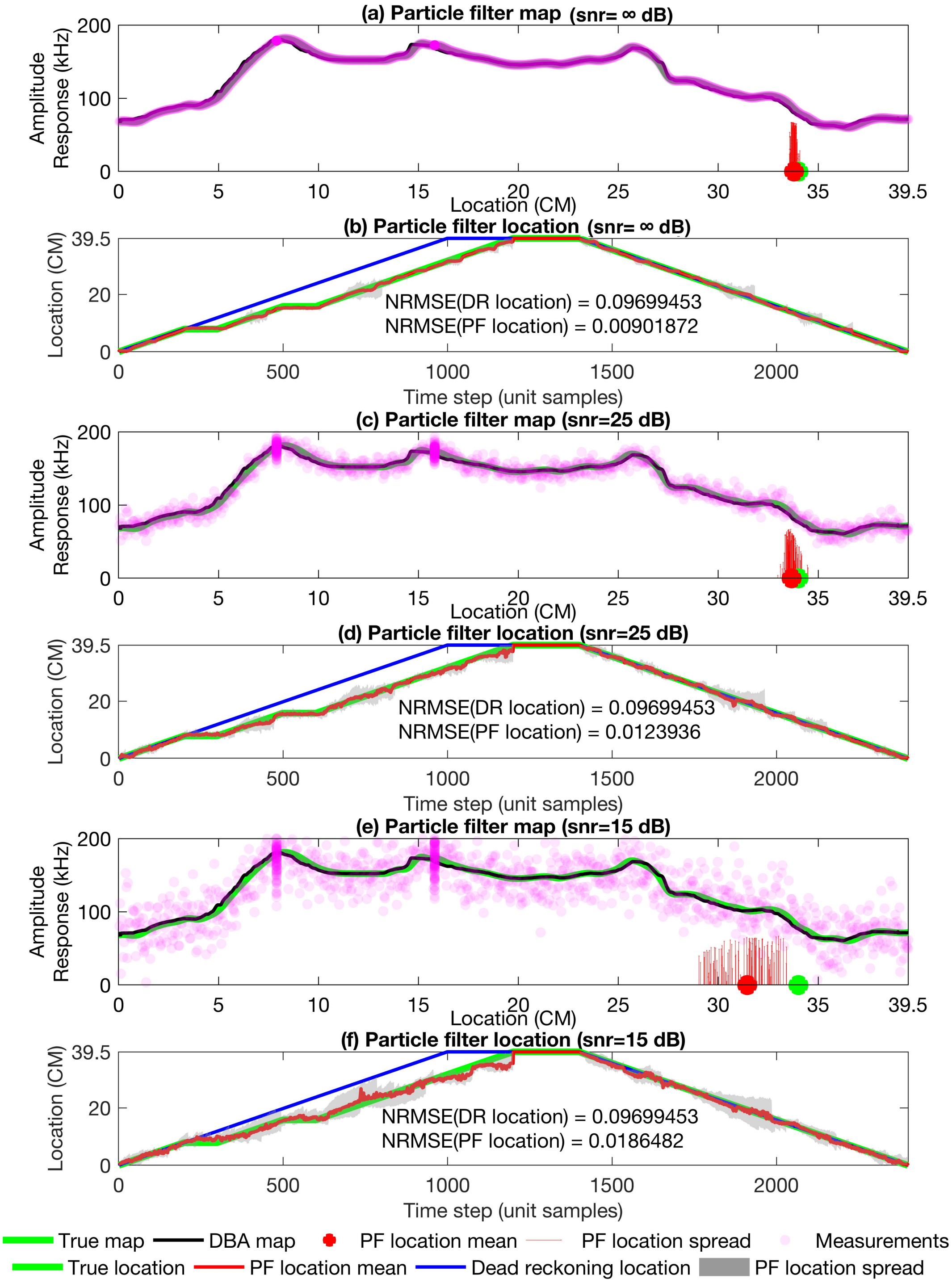


Figure 4.25: PF localisation in 1m metal water pipe with blockage noise in motor encoder. (a)-(b) Localisation with additional sensor noise $snr = \infty$ dB (noise free), (c)-(d) Localisation with additional sensor noise $snr = 20$ dB, (e)-(f) Localisation with additional sensor noise $snr = 15$ dB.

measurements increases to $snr = 15$ dB, the EKF location error is still relatively small ((NRMSE = 2.4790%). But for plastic pipe data (see Fig.4.22), the location error is largely increased (NRMSE = 2.7209% for $snr = \infty$ dB and NRMSE = 5.0540% for $snr = 30$ dB). Note that the EKF estimation for plastic pipe will fail to track the true robot location when the white noise is too big ($snr = 30$ dB) if the blockage noise is added. The reason is because the map contains two large flat regions. If the robot sticks in the flat map region too long, EKF estimation will fail and the grey shadow (EKF covariance or confidence region) will not overlay the true robot location. However, failure will not occur for the 1-metre metal pipe data, as its map is varying along the pipe with no too-flat region.

For the particle filter, however, estimation for plastic pipe data will not fail if a sufficient number of particles is used (see Fig.4.24). When the robot travels through a flat map region, the particles will diverge. When the robot comes out of the flat region, particle filter estimation will quickly converge to the true robot location and largely decrease the uncertainty range (i.e. the particle spread).

4.3.6 Discussion

In this chapter, the effectiveness of sequential mapping and localisation in water pipes has been demonstrated, by using a number of sets of experiment data with additional various types of noise. Both mapping using DBA algorithm and localisation using EKF and PF have shown significant improvement over dead reckoning estimation.

As explained in Section 4.3.1, the accuracy of final map generated using DBA algorithm can be varied if different initial map is used in the DBA iteration. Therefore, additional analysis needs to be carried out to evaluate such uncertainty. It can be seen from Fig.4.26 that dead reckoning location error is constant along different number of runs in pipe. But EKF and particle filter location errors (NRMSE) are varied. For each selected number of runs, 10 simulations have been carried out. Therefore, the location error mean is numerically calculated from these 10 simulations and the error bar represents the standard deviation of the 10 simulations. From the results shown in Fig.4.26, both EKF and PF locations are much accurate compared to the dead reckoning location.

The original dead reckoning data (motor encoder readings) are assumed to be accurate without any noise. However, in order to show the effectiveness and robustness of the mapping and localisation algorithm, additional noises are added into both dead reckoning data and sensor measurements. Table 4.3 shows the full localisation results using either EKF or PF in different data sets with various additional noises. Fig.4.27 $\sim$ 4.29 give a visual illustration of the localisation algorithms

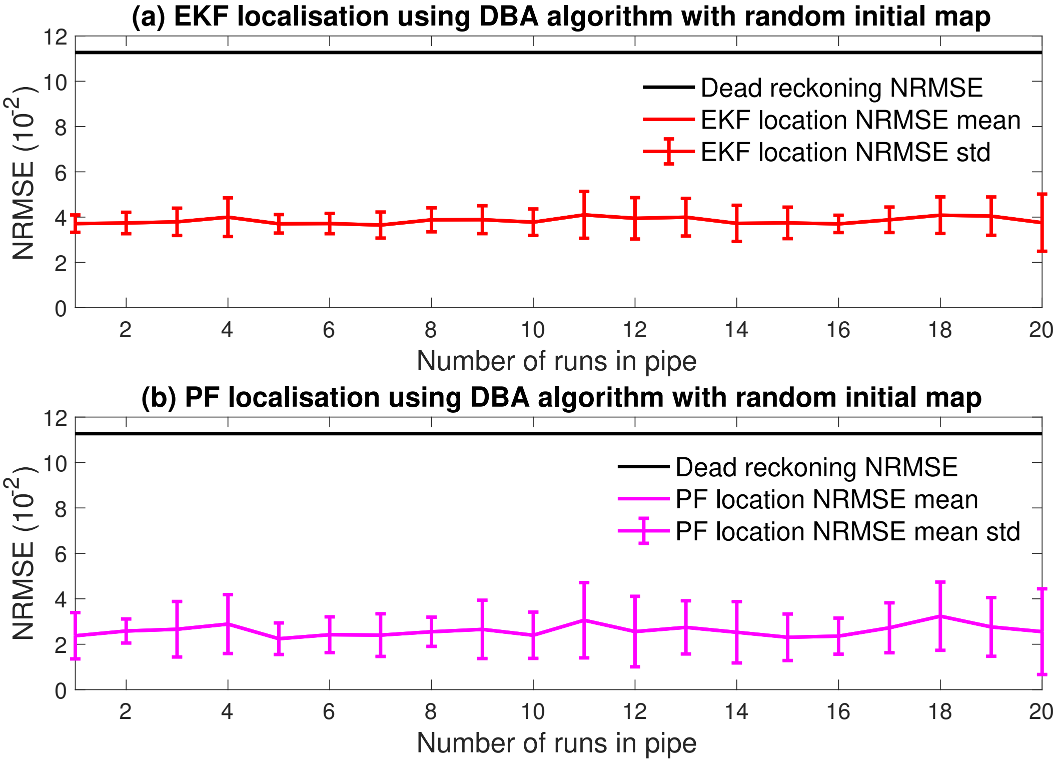


Figure 4.26: Evaluation of the impact of random initial map selection in DBA algorithm on localisation performance. (a) EKF localisation error. (b) Particle filter localisation error.

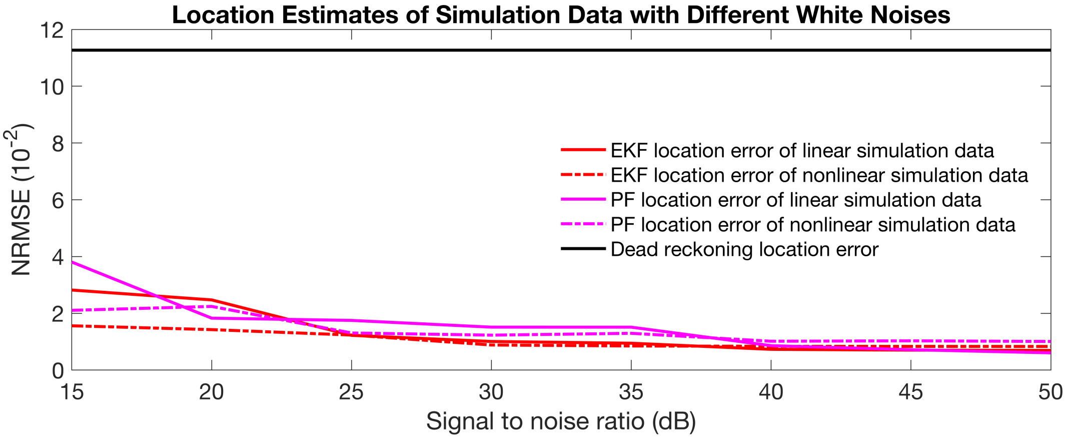


Figure 4.27: Location estimates of simulation data with different levels of white noises. The additional added white noise are measured using signal-to-noise ratio (snr) from 15 dB to 50 dB.

using different data sets with different additional noises.

From Fig.4.27, it can be seen that when additional white noises in sensor measurements are very small ($snr = 50$ dB) both EKF and PF provide better location estimates compared with larger white noises ($snr = 15$ dB). For both linear simulated data (solid lines) and nonlinear simulated data (dashed lines), EKF localisation (red lines) gives more accurate location estimates compared with PF localisation (purple lines). Both EKF and PF location estimation are all outperforming against dead reckoning location estimation (solid black line). Location errors in dead reckoning are assumed to be the same for all simulation and experiment data sets excluding the tests for blockage noise.

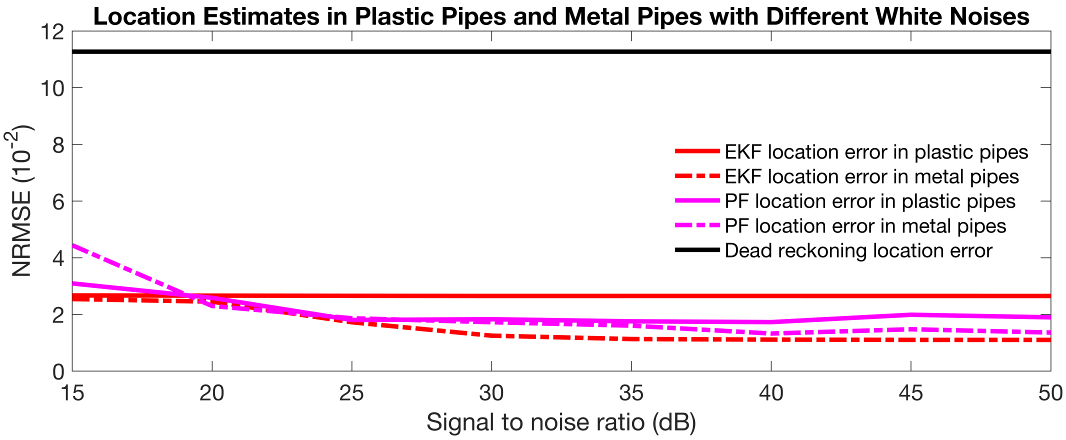


Figure 4.28: Location estimates using data from plastic pipe and 1-metre metal pipe. The additional added white noise are measured using signal-to-noise ratio (snr) from 15 dB to 50 dB.

The localisation results for plastic pipe and 1-metre metal pipe are shown in Fig.4.28. Both EKF and PF estimations provide more accurate location estimates for the 1-metre metal pipe compared with plastic pipe – both dashed red line and dashed purple line are below solid red line and solid purple line for $snr > 25$ dB. As mentioned previously, the reason behind is the navigation maps. The amplitudes of the 1-metre metal pipe is not flat almost for all regions in the map. However, for plastic pipe data, if there is no void outside the pipe wall, the map amplitude will be flat and equals to zero. This is also another drawback of localisation using ultrasound sensor. But for metal pipes, the frequency responses are always changing at different locations along the pipe, therefore the maps of the metal pipes will not contain flat regions. It is important to note that a flat map means it does not contain 'features'. But if the map fluctuates too often with high frequency and amplitude, it will look like a map full of white noises – this is not a good navigation map as well.

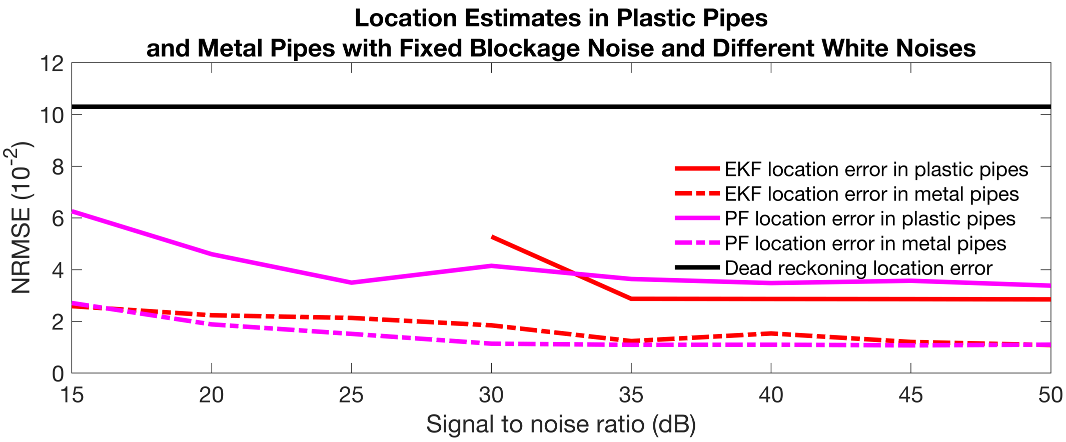


Figure 4.29: Location estimates using data from plastic pipe and 1-metre metal pipe. Fixed blockage noises are added. The additional added white noise are measured using signal-to-noise ratio (snr) from 15 dB to 50 dB.

Similarly, with additional blockage noise added shown in Fig.4.29, localisation in metal pipe is more accurate than that in plastic pipe – dashed lines are below solid line for $snr = 15$ dB to $snr = 50$ dB. It is worth mentioning that, with blockage noise added, when the amplitude of white noise increases ($snr = 30$ dB), EKF location estimation will fail to track the true location for plastic pipe data.

It is important to point out again that the linearisation method used in EKF estimation is by using a quadratic fit to the local map. Therefore, this method is more accurate for maps that contains more quadratic-like regions. For the 1-metre metal pipe map, the shape of each local region can be modelled using a quadratic function. But for the plastic pipe map, although it contains two big voids (features), they all look like triangles with two straight edges that are unlike quadratic functions. Therefore, the EKF estimation with quadratic fit can handle the metal pipe data but for plastic pipes it is not the best solution.

4.4 Summary

This chapter has addressed the problem of robot localisation in simulated linear and nonlinear maps, plastic and metal water pipes. In order to address the problem of spatially calibrating the map using a dead reckoning sensor subject to drift, it introduced the use of a signal alignment and averaging algorithm based on dynamic time warping. The localisation was based on nonlinear state estimation. This chapter evaluated the approach on a combination of experimental and simulation data, demonstrating that the technique is effective.

Table 4.3: EKF and PF location estimation results comparison in different levels of white noises

Data types	**Algorithms**	**Normalised rooted mean square error (NRMSE)** $\times 10^{-2}$									
		DR	snr=15	snr=20	snr=25	snr=30	snr=35	snr=40	snr=45	snr=50	snr=∞
Linear simulation data	EKF	11.2694	2.8201	2.4731	1.2242	1.0088	0.9472	0.7320	0.7035	0.6924	0.6819
	PF	11.2694	3.8059	1.8299	1.7508	1.5135	1.5128	0.8611	0.7184	0.6082	0.6608
Nonlinear simulation data	EKF	11.2694	1.5613	1.4263	1.2348	0.8841	0.8506	0.8384	0.8354	0.8335	1.0264
	PF	11.2694	2.1060	2.2436	1.3081	1.2315	1.2966	1.0169	1.0323	1.0091	0.9968
Plastic pipes data	EKF	11.2694	2.6720	2.6651	2.6581	2.6530	2.6536	2.6537	2.6532	2.6528	2.6524
	PF	11.2694	3.0959	2.5871	1.8070	1.8327	1.7588	1.7311	1.9906	1.8994	1.7208
Metal pipes data (1-metre)	EKF	11.2694	2.5451	2.4527	1.7259	1.2529	1.1337	1.1139	1.1045	1.1041	1.1104
	PF	11.2694	4.4418	2.3000	1.8672	1.7277	1.6064	1.3283	1.4798	1.3629	1.4718
Plastic pipes data (with blockage noise)	EKF	10.8057	-	-	-	5.5378	3.0138	3.0069	3.0021	2.9906	2.9813
	PF	10.8057	6.5655	4.8230	3.6695	4.3515	3.8172	3.6529	3.7456	3.5479	3.8389
Metal pipes data (with blockage noise)	EKF	10.8057	2.7163	2.3473	2.2389	1.9390	1.3004	1.6094	1.2647	1.1388	1.0711
	PF	10.8057	2.8439	1.9771	1.5922	1.1950	1.1472	1.1533	1.1273	1.1552	1.1025

Chapter 5

Simultaneous Localisation and Mapping in Water Pipes

5.1 Introduction

In the previous chapter, a method was presented for sequentially mapping and then localising a robot in a water pipe. This method appeared to work well judging by the simulated and experimental results. However, the sequential method is limited because it does not allow the map to be updated in real-time, which would be essential for future robot systems that could monitor pipes on a longer term basis. This is due to the fact that the pipe and surrounding soil conditions might change, altering the map for either metal or plastic pipes. Therefore, the aim of this chapter is to develop SLAM for feature sparse water pipes based on the sensing techniques developed in chapter 3.

As mentioned in the literature review, mapping and localisation has been developed for water pipes using SLAM techniques with cameras and inertial measurement units (IMUs) [85, 91]. In this chapter, a SLAM algorithm for water pipes, termed PipeSLAM, is developed based on the Rao-Blackwellised particle filter (RBPF) [42]. Specifically, the map is regarded as a continuous function over space that is represented using a weighted basis function decomposition - a radial basis function network. We use a state-space model to describe the robot dynamics, where the state vector is composed of robot pose and is augmented with the basis function parameters to describe the map.

The state-space representation of the robot location and the map parameters in an augmented state vector is well suited to estimation via the RBPF, because the map function is linear-in-the-parameters, and so can be estimated via the Kalman filter, and particles can be used to represent robot poses, solving the localisation problem. In addition, there can be many map parameters, and so this part is not suited to estimation via the particle filter, whilst the robot location is represented by relatively few parameters and so is well suited to estimation via particle filtering.

The approach to mapping is demonstrated experimentally on data from both plastic and metal pipes, as done in the last chapter on sequential mapping and localisation. The results of the PipeSLAM algorithm are also benchmarked against a dead reckoning solution, and demonstrate that the RBPF algorithm can be used to more successfully solve the mapping and localisation problem using the terrain-type map.

Section 5.2 defines the in-pipe SLAM problem, presents the solution of PipeSLAM algorithm using Rao-Blackwellised Particle Filter (RBPF) [41]. Section 5.3 presents the results on evaluation of the PipeSLAM algorithm using a combination of experimental and simulation data. Table 5.2 has shown the list of data that are used to evaluate the PipeSLAM algorithm. Table 5.3 and Table 5.4 show the PipeSLAM results compared with the localisation results using sequential mapping and localisation methods that are introduced in Chapter 4. Finally, Section 5.4 concludes the chapter.

5.2 Methods

5.2.1 Problem Statement

In this work we focus on the problem of robot navigation in relatively small pipes, with diameter of about 3 inches (a current prototype robot design is shown in Fig.3.3). These small pipes are commonly used for water distribution in urban environments. The small diameter means that movement within the pipe is restricted, consisting of back and forth movement only. For entry into these small pipes, fire hydrants can potentially be used, which in the UK, Europe and the USA are spaced approximately every 100 metres [121, Chapter 14, Table 14.2].

We model the dynamics of the pipe robot using a state-space model, with state dynamics

$$p(\mathbf{x}_k|\mathbf{x}_{k-1},\mathbf{u}_{k-1}) \Leftrightarrow \mathbf{x}_k = \mathbf{f}\left(\mathbf{x}_{k-1},\mathbf{u}_{k-1}\right) + \mathbf{w}_{k-1} \tag{5.1}$$

where k is the sample index, $\mathbf{x}_k \in \mathbb{R}^{n_x}$ is the robot pose (including location), $\mathbf{f}(.)$ is the state transition function, $\mathbf{u}_{k-1} \in \mathbb{R}^{n_u}$ is the input, $\mathbf{w}_{k-1} \sim N(0, \Sigma_w)$ is the state noise. In this work we simplify the state vector $\mathbf{x}_k$ to contain just the spatial location of the robot, x, along the pipe, and the input $\mathbf{u}_k$ is obtained from a processed motor encoder reading defining a dead reckoning estimate of distance travelled, hence $\mathbf{f}\left(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}\right) = \mathbf{F}_x \mathbf{x}_{k-1} + \mathbf{G}_{k-1}\mathbf{u}_{k-1}$, where $\mathbf{F}_x = 1$, $\mathbf{G}_k = 1$. The measurement model is

$$p(\mathbf{y}_k|\mathbf{x}_k) \Leftrightarrow \mathbf{y}_k = \mathbf{h}\left(\mathbf{x}_k\right) + \mathbf{v}_k \tag{5.2}$$

where the observation $\mathbf{y}_k \in \mathbb{R}^{n_x}$ is obtained from the hydrophone signal, where vibration amplitude is averaged over some frequency range, and the measurement function $\mathbf{h}(.)$ is a one-dimensional map, which transforms from the spatial location of the robot $\mathbf{x}_k$ to mean pipe vibration amplitude; also $\mathbf{v}_k \sim N(0, \Sigma_v)$ is the measurement noise.

To represent the unknown map, $\mathbf{h}(\mathbf{x})$, where $\mathbf{x}$ is a spatial location, we use a basis function decomposition,

$$\mathbf{h}(\mathbf{x}) = \sum_{j=1}^{M} \theta_j \phi_j(\mathbf{x}) \tag{5.3}$$

where M is the number of basis functions, $\theta_j \in \mathbb{R}$ is the corresponding weight of the jth basis function, and $\phi_j(\mathbf{x})$ is a basis function. Note that this representation of $\mathbf{h}(\mathbf{x})$ is linear-in-the-parameters θ_j. Here we use radial basis functions for $\phi(\mathbf{x})$, specifically the squared exponential function,

$$\phi_j(\mathbf{x}) = \exp\left(-\frac{(\mathbf{x} - c_j)^2}{2\sigma^2}\right) \tag{5.4}$$

where c_j is the centre of the basis function (a spatial location along the pipe) and σ is the width of a basis function, which we assume for simplicity here are the same across all basis functions. We also assume that the centres can be evenly spaced based on some prior knowledge of the likely spatial frequency of variation in pipe vibration amplitude, although it would be possible to use an adaptive technique to place basis functions [75, 118]. Hence, in this case, the parameter vector $\boldsymbol{\theta}$ is the only unknown describing the map, where

$$\boldsymbol{\theta} = \left(\theta_1, \theta_2, \ldots, \theta_M\right)^T \tag{5.5}$$

We define the probabilistic SLAM problem here in the usual way, as computing the joint distribution of robot location and map over all samples times [46],

$$p(\mathbf{x}_k, \boldsymbol{\theta}|\mathbf{y}_{1:k}, \mathbf{u}_{1:k}, \mathbf{x}_0) \tag{5.6}$$

where $\mathbf{y}_{1:k} = \{\mathbf{y}_1, \ldots, \mathbf{y}_k\}$ is the set of observations, $\mathbf{u}_{1:k} = \{\mathbf{u}_1, \ldots, \mathbf{u}_k\}$ is the set

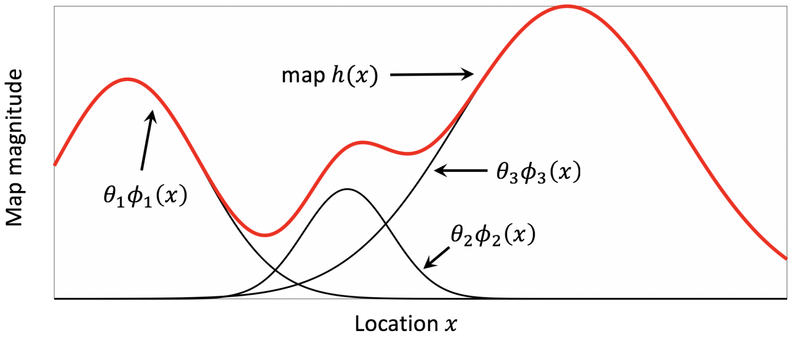


Figure 5.1: A diagram showing the construction of a one dimensional function by using radial basis functions. The red function is the sum of three black radial basis functions.

of all inputs and $\mathbf{x}_0$ is the initial location of the robot.

5.2.2 Estimation via a Rao-Blackwellised Particle Filter

To solve the problem of estimating the joint distribution of robot location and map parameters, $p(\mathbf{x}_k, \boldsymbol{\theta} | \mathbf{y}_{1:k}, \mathbf{u}_{1:k}, \mathbf{x}_0)$, we initially define an augmented state-space model, with linear state dynamics

$$\begin{bmatrix} \mathbf{x}_k \\ \boldsymbol{\theta}_k \end{bmatrix} = \begin{bmatrix} \mathbf{F}_x & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_m \end{bmatrix} \begin{bmatrix} \mathbf{x}_{k-1} \\ \boldsymbol{\theta}_{k-1} \end{bmatrix} + \begin{bmatrix} \mathbf{G}_{k-1} \\ \mathbf{0} \end{bmatrix} \mathbf{u}_{k-1} + \begin{bmatrix} \boldsymbol{\omega}_{k-1} \\ \boldsymbol{\eta}_{k-1} \end{bmatrix} \tag{5.7}$$

where $\mathbf{I}_m$ is the identity matrix of dimension M and $\boldsymbol{\eta}_k \sim N(0, \Sigma_\eta)$ is a noise term accounting for uncertainty in the map parameter evolution. The measurement model is

$$\mathbf{y}_k = \Phi(\mathbf{x}_k)\boldsymbol{\theta}_k + \boldsymbol{\nu}_k \tag{5.8}$$

where

$$\Phi(\mathbf{x}_k) = \Big(\phi_1(\mathbf{x}_k), \ldots, \phi_M(\mathbf{x}_k)\Big) \tag{5.9}$$

The problem of estimating the augmented state vector is nonlinear, and so, for example an extended Kalman filter might be used to obtain the solution [46]. Alternatively, a particle filter might be used, which avoids the linearisation of the measurement function [129]. However, in this case, naive application of a standard particle filter would be computationally infeasible because the number of basis functions used to describe the map, and hence the state dimension, could be very large depending on the length of pipe explored.

A further alternative is the Rao-Blackwellised particle filter (RBPF) [42], which in this case can be used to exploit the linear-in-the-parameters nature of the basis function decomposition of the map: a Kalman filter can be used to update the parameters $\boldsymbol{\theta}_k$, effectively reducing the state-dimension for the particle representation, keeping the computational complexity manageable. A particle filter can still be used to represent the robot location, which in this case is a one-dimensional problem, well suited to particle filtering. A similar formulation has been used in the popular FastSLAM algorithm [141].

The principle of Rao-Blackwellisation is to partition the joint distribution of $p(\mathbf{x}_k, \boldsymbol{\theta}|\mathbf{y}_{1:k}, \mathbf{u}_{1:k}, \mathbf{x}_0)$ using the product rule, where

$$p(\mathbf{x}_k, \boldsymbol{\theta}_k|\mathbf{y}_{1:k}, \mathbf{u}_{1:k}, \mathbf{x}_0) = p(\boldsymbol{\theta}_k|\mathbf{x}_k, \mathbf{y}_{1:k})p(\mathbf{x}_k|\mathbf{y}_{1:k}, \mathbf{u}_{1:k}, \mathbf{x}_0) \tag{5.10}$$

The key point to note is that the function $p(\boldsymbol{\theta}_k|\mathbf{x}_k, \mathbf{y}_{1:k})$ can be represented analytically, as a conditional Gaussian, so that only the function $p(\mathbf{x}_k|\mathbf{y}_{1:k}, \mathbf{u}_{1:k}, \mathbf{x}_0)$ requires sampling [129]. This is why the map parameters $\boldsymbol{\theta}_k$ can be updated in a linear estimation step, i.e. by the Kalman filter.

A further significant point to note when using the RBPF algorithm in SLAM is that due to representing the robot pose via a set of particles, $\mathbf{x}_k^{(i)} \sim p(\mathbf{x}_k|\mathbf{y}_{1:k}, \mathbf{u}_{1:k}, \mathbf{x}_0)$, for $i = 1, \ldots, n_s$, where n_s is the number of particles, each individual particle has its own associated map. This means for every single particle representing robot pose $\mathbf{x}_k^{(i)}$, there is a distinct associated set of map parameters $\boldsymbol{\theta}_k^{(i)}$, and a separate Kalman filter must be computed to update each map.

5.2.3 PipeSLAM Algorithm

In this section the PipeSLAM algorithm is described for estimating the joint distribution $p(\mathbf{x}_k, \boldsymbol{\theta}_k|\mathbf{y}_{1:k}, \mathbf{u}_{1:k}, \mathbf{x}_0)$ using the RBPF.

Initialise particles

Draw initial samples based on $\mathbf{x}_0^{(i)} \sim p(\mathbf{x}_0|\mathbf{y}_0)$, and initialise the parameter settings for the Kalman filter for all particle samples, such that $\boldsymbol{\theta}_0^{(i)} = \hat{\boldsymbol{\theta}}_0$, for $i = 1, \ldots, n_s$, where n_s is the number of particle samples.

Particle filter time update

The particle state $\mathbf{x}_k$ is updated at each time point according to linear state transition model (5.11),

$$\mathbf{x}_k^{(i)-} = \mathbf{F}_x\hat{\mathbf{x}}_{k-1}^{(i)} + \mathbf{G}_{k-1}\mathbf{u}_{k-1} + \mathbf{w}_{k-1}^{(i)}, \text{ for } i = 1, \ldots, n_s \tag{5.11}$$

Kalman filter time update

The basis function weights for each map associated with the i^{th} particle, $\boldsymbol{\theta}_k^{(i)}$, for $i = 1, \ldots, n_s$, are updated at time step k. Due to the assumption that the map is time-invariant, the weights $\boldsymbol{\theta}_k^{(i)}$ are also assumed to be unchanging over time, so the prediction step for $\hat{\boldsymbol{\theta}}_k^{(i)}$ is a constant for the mean, and an inflation of the uncertainty in the covariance,

$$\boldsymbol{\theta}_k^{(i)-} = \hat{\boldsymbol{\theta}}_{k-1}^{(i)} \tag{5.12}$$

$$\mathbf{P}_k^{(i)-} = \mathbf{P}_{k-1}^{(i)} + \Sigma_\eta \tag{5.13}$$

Kalman filter measurement update

The estimation of basis function weights for each map, $\hat{\boldsymbol{\theta}}_k^{(i)}$, for $i = 1, \ldots, n_s$, are corrected using the standard Kalman filter update step,

$$\mathbf{S}_k^{(i)} = \Phi\left(\mathbf{x}_k^{(i)-}\right)\mathbf{P}_k^{(i)-}\Phi^T\left(\mathbf{x}_k^{(i)-}\right) + \Sigma_{\mathbf{v}} \tag{5.14}$$

$$\mathbf{K}_k^{(i)} = \mathbf{P}_k^{(i)-}\Phi^T\left(\mathbf{x}_k^{(i)-}\right)\mathbf{S}_k^{(i)^{-1}} \tag{5.15}$$

$$\hat{\boldsymbol{\theta}}_k^{(i)} = \boldsymbol{\theta}_k^{(i)-} + \mathbf{K}_k^{(i)}\left(\mathbf{y}_k - \Phi\left(\mathbf{x}_k^{(i)-}\right)\boldsymbol{\theta}_k^{(i)-}\right) \tag{5.16}$$

$$\mathbf{P}_k^{(i)} = \mathbf{P}_k^{(i)-} - \mathbf{K}_k^{(i)}\Phi\left(\mathbf{x}_k^{(i)-}\right)\mathbf{P}_k^{(i)-} \tag{5.17}$$

Evaluate and normalise importance weights

Particle weights, $\omega_k^{(i)}$ for $i = 1, \ldots, n_S$, need to be re-evaluated based on the likelihood for the newly arrived observation, and then normalised,

$$\tilde{\omega}_k^{(i)} = \omega_{k-1}^{(i)} p(\mathbf{y}_k | \mathbf{x}_k^{(i)-}, \hat{\boldsymbol{\theta}}_k^{(i)}) \tag{5.18}$$

$$\omega_k^{(i)} = \frac{\tilde{\omega}_k^{(i)}}{\sum_{j=1}^{n_s} \tilde{\omega}_k^{(j)}} \tag{5.19}$$

Particle resampling

When the number of effective particles (n_{eff}) drops to a certain threshold (N_{res}), the particle set needs to be resampled to avoid excessive particle depletion [129],

$$n_{\text{eff}} = \frac{1}{\sum_{j=1}^{n_s}\left(\omega_k^{(j)}\right)^2} \tag{5.20}$$

Table 5.1: Parameter settings for PipeSLAM algorithm evaluation

Parameter description	**Parameters**	**Values**
Number of basis functions	M	100
Width of basis functions	σ	10
State transition matrix of x	$\mathbf{F}_x$	1
State transition matrix of $\boldsymbol{\theta}$	$\mathbf{I}_m$	$\boldsymbol{I}_{100\times100}$
Input matrix	$\mathbf{G}_k$	1
State noise covariance in x	Σ_ω	0.3 cm^2
State noise covariance in $\boldsymbol{\theta}$	Σ_η	$\mathbf{0}_{100\times100}$
Measurement noise covariance	Σ_ν	0.1
Particle sample size	n_s	100
Efficient particles threshold	N_{res}	50

Therefore, when n_{eff} drops to lower than a predefined threshold N_{res}, the particle set $\{\hat{\mathbf{x}}_k^{(i)}, \hat{\boldsymbol{\theta}}_k^{(i)}, \mathbf{P}_k^{(i)}\}$ needs to be resampled by drawing n_s times from the present particle set to a new particle set and the probability of drawing a particle $\{\mathbf{x}_k^{(i)}, \boldsymbol{\theta}_k^{(i)}, \mathbf{P}_k^{(i)}\}$ is proportional to its associated weight $\omega_k^{(i)}$. The algorithm is then repeated from step 2.

5.2.4 Algorithm Evaluation

The PipeSLAM algorithm was evaluated using a combination of experimental and simulation data. We used the map generated by the experiment described above as a ground truth map. We then simulated a robot moving back and forth along this map by the state-space model defined in (5.1) and (5.2).

The PipeSLAM algorithm was used in the simulation phase to simultaneously estimate both the map and robot location. Table 5.1 shows the parameter settings used in the PipeSLAM algorithm. In particular, the number of particle samples, n_s, was set to 100, which is relatively few but in this one-dimensional problem performed well.

To make the SLAM problem more challenging and highlight the benefit of using the map over dead reckoning, we also added a deterministic linear and sinusoidal drift term d_k to the state equation, of the form

$$d_k = -0.2\bar{m}_k + 0.02\bar{m}_k \sin(0.1\bar{m}_k + 2) \tag{5.21}$$

where $\bar{m}_k = km_k$, where $m_k = 1$ (a.u.) for plastic pipe, $m_k = 0.4995$ cm for 5-metre metal pipe and $m_k = 0.0395$ cm other data sets.

Table 5.2: List of figures showing PipeSLAM results with additional Gaussian noise in measurements. The lower the snr value the larger noise but snr=∞ indicates no additional noise. For example, the PipeSLAM result of Metal pipe of 5m long Data I with snr=20 dB is shown in Fig.5.20.

Data	**Signal to noise ratio (snr) in dB**			
	$snr = \infty$	$snr = 30$	$snr = 20$	$snr = 15$
Linear data	5.2	5.3	5.4	5.5
Non-Linear data	5.6	5.7	5.8	5.9
Plastic pipe data	5.10	5.11	5.12	5.13
Metal 1m pipe data	5.14	5.15	5.16	5.17
Metal 5m pipe Data I	5.18	5.19	5.20	5.21
Metal 5m pipe Data I reverse	5.22	5.23	-	-
Metal 5m pipe Data II	5.24	5.25	5.26	5.27

5.3 Results and Discussion

5.3.1 Results from Simulation Data

Result figures of PipeSLAM algorithm using different data sets and with different noise levels are listed in Table 5.2. The order is showing from lower level of noise to higher level of noise, i.e., from no noise $snr = \infty$ to $snr = 15$. It is worth mentioning that $snr = \infty$ indicates no additional noise.

PipeSLAM results are firstly shown using the linear (Fig.5.2 $\sim$ Fig.5.5) and nonlinear (Fig.5.6 $\sim$ Fig.5.9) simulation data with additional white noise added in measurements (at signal to noise ratio $snr = \infty$, $snr = 30$, $snr = 20$ and $snr = 15$). The simulation of robot movement along the pipe is from left (forwards) to right (backwards) over a ground true map generated from linear and nonlinear simulation data. Panels (a)-(d) in those figures are PipeSLAM at increasing time steps of robot movement, where (a)-(b) are the forward movement and (c)-(d) are the backward movement. Panel (e) in those figures shows the PipeSLAM location results with different additional white noise. Compared to dead reckoning estimate, the PipeSLAM locations show much improved performance that exhibits large drift. From time step $t = 1000$ to $t = 1200$, it is assumed that the robot is staying at the right end of the pipe and it starts to move to the left (backwards) at time step $t = 1201$.

In panels (a)-(c) in Fig.5.2 $\sim$ Fig.5.5, the 100 particle maps (light blue lines) are initiated with zero weights and spread out as the robot moves. The black line indicates the specific particle map that is of the largest weight. Some particles share the same largest weight, it randomly picks one of them to plot in black. When the robot reaches to the right end of the pipe (known location), all particle locations will reset to be the true location. This process is similar to the *loop closure* situation in typical SLAM problems. The algorithm will resample all particles when the robot reaches to a known location. Therefore, all particles of small weights will die out and large weighted particle survive. Thus, all particle locations as well as dead reckoning location will be set to the known location. This is why the blue dead reckoning location in panel (e) jumps to 39.5 cm at time step 1000. The grey shaded area in panel (e) shows the particle spread indicating the confident zone in its location estimation. It can be seen that on the forward pass, the particle spread goes wider as the robot moves further. This is because the accumulated error in motor encoder uncertainty. Due to the fact that the robot will not 'see' a 'feature' twice for the first time, this accumulated error cannot be corrected and the particles will continuously spread out. But when the particles spread is too wide, calculated by the number of efficient particles n_{eff} expressed in (5.20), the particle set will resample to reduce the uncertainty. Therefore, in Fig.5.2e, the grey shaded area goes wider and after a short time, it narrows again, both in the forward pass and the backward pass.

Compared PipeSLAM results with different levels of additional added white noise, the larger the additional white noise added, the wider the particle spreads, and the worse the estimated map.

5.3.2 Results from Plastic Water Pipe Data

The PipeSLAM algorithm is also evaluated using plastic pipe data shown in Fig.5.10 $\sim$ Fig.5.13. Same to assumption made in the linear and nonlinear simulation data, the robot starts from very left of the pipe and moves towards the right end of the pipe. When it reaches to the right end, it stops for 200 time steps and move backward to the left.

The main difference between the plastic pipe data and the simulated linear or nonlinear data is that the former contains two large flat regions (i.e. from location 300 to 500 and 750 to 1000) shown in Fig.5.10 $\sim$ Fig.5.13. The 'feature' regions are from location 0 to 300 and 500 to 750, which are like triangles in their shapes similar to simulated linear map. Therefore, the PipeSLAM works well in those

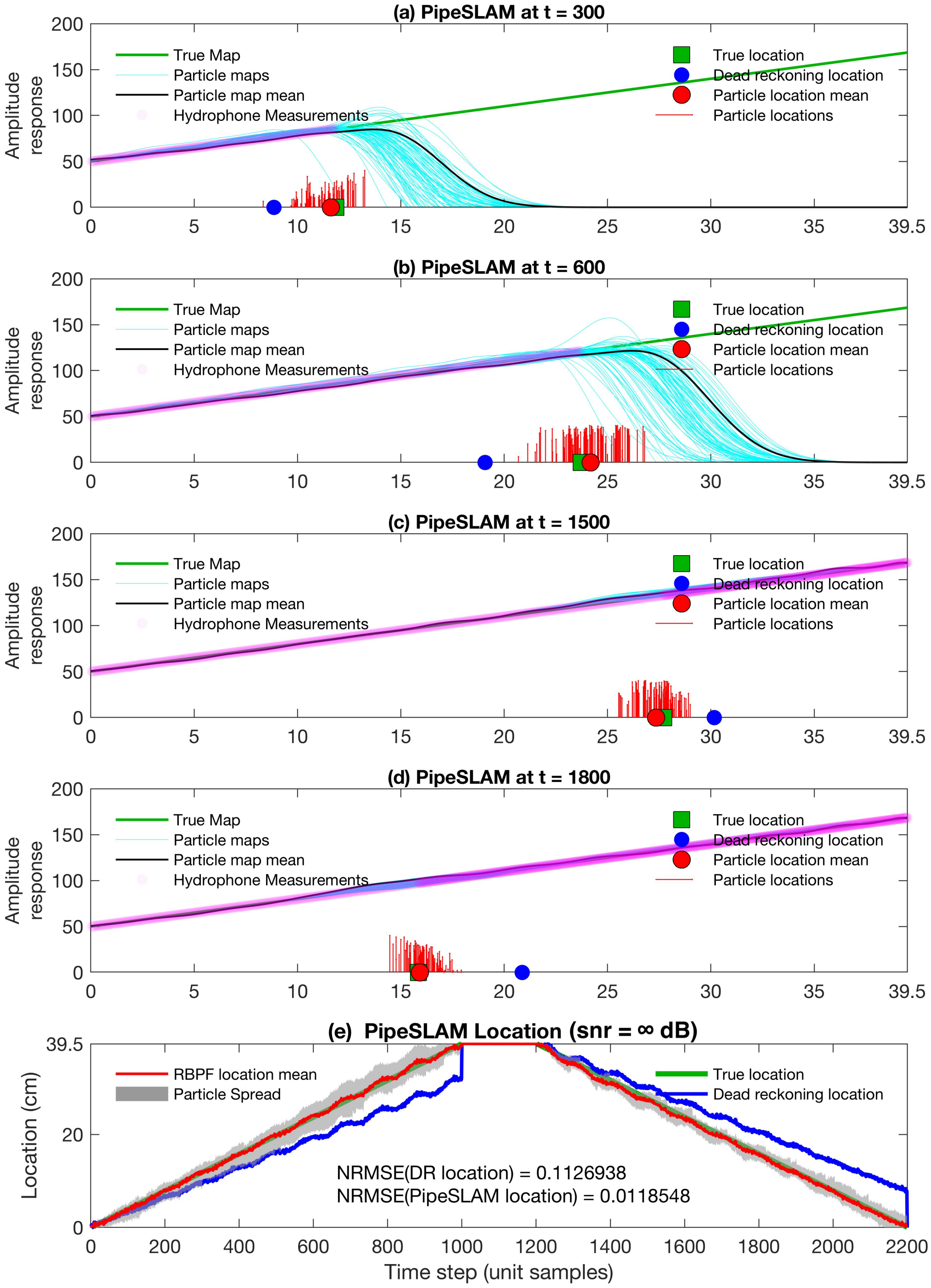


Figure 5.2: PipeSLAM results from simulation of robot movement along a pipe, forwards and backwards, over a ground truth map generated from linear simulation data (snr = ∞ dB). (a)-(d) PipeSLAM at increasing time steps of robot movement (a-b: forward movement, left to right; c-d: backward movement, right to left). (e) PipeSLAM localisation results over time, showing the improvement compared to the dead reckoning estimate, which exhibits significant drift.

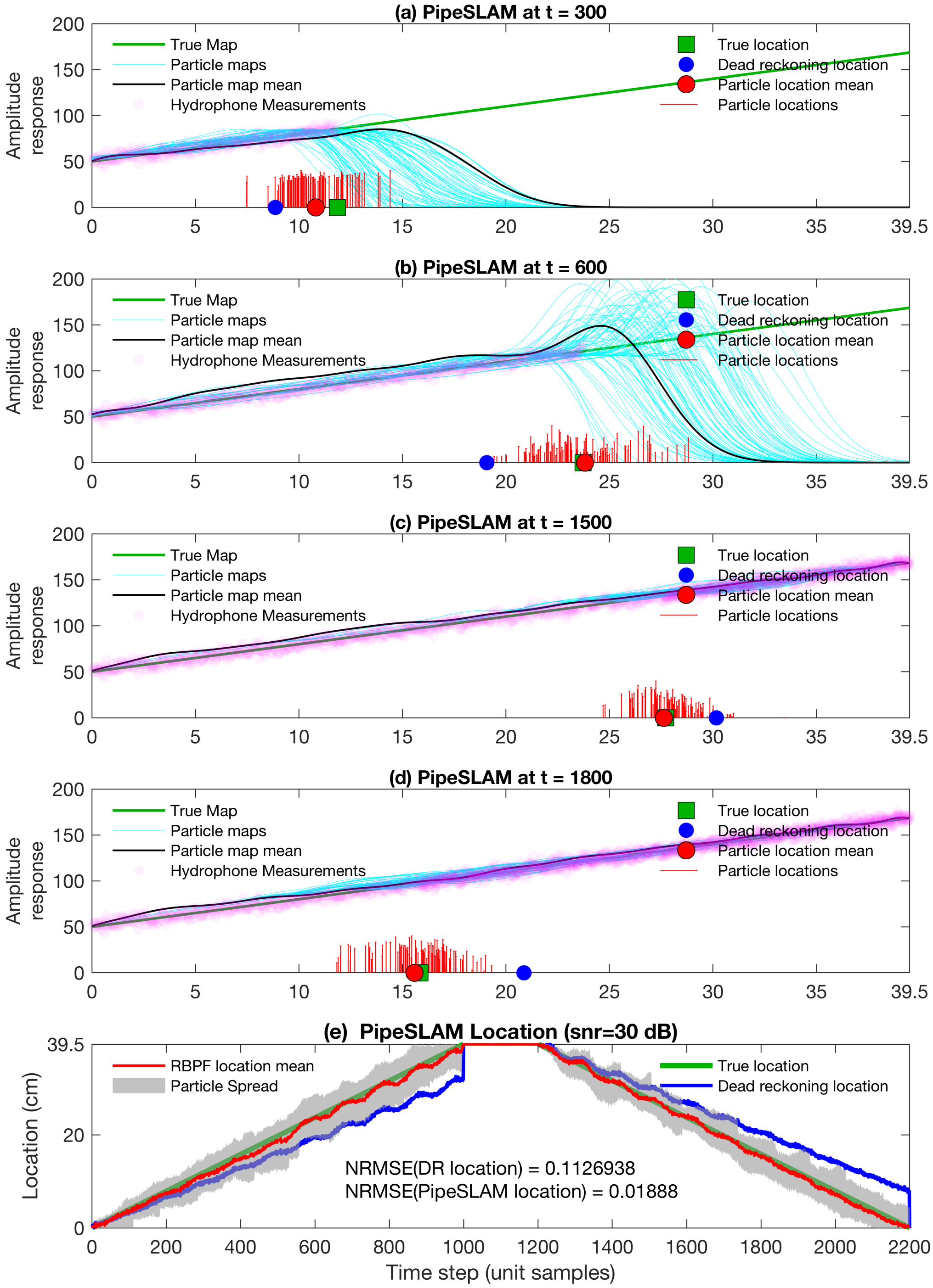


Figure 5.3: PipeSLAM results from simulation of robot movement along a pipe, forwards and backwards, over a ground truth map generated from linear simulation data (snr = 30 dB). (a)-(d) PipeSLAM at increasing time steps of robot movement (a-b: forward movement, left to right; c-d: backward movement, right to left). (e) PipeSLAM localisation results over time, showing the improvement compared to the dead reckoning estimate, which exhibits significant drift.

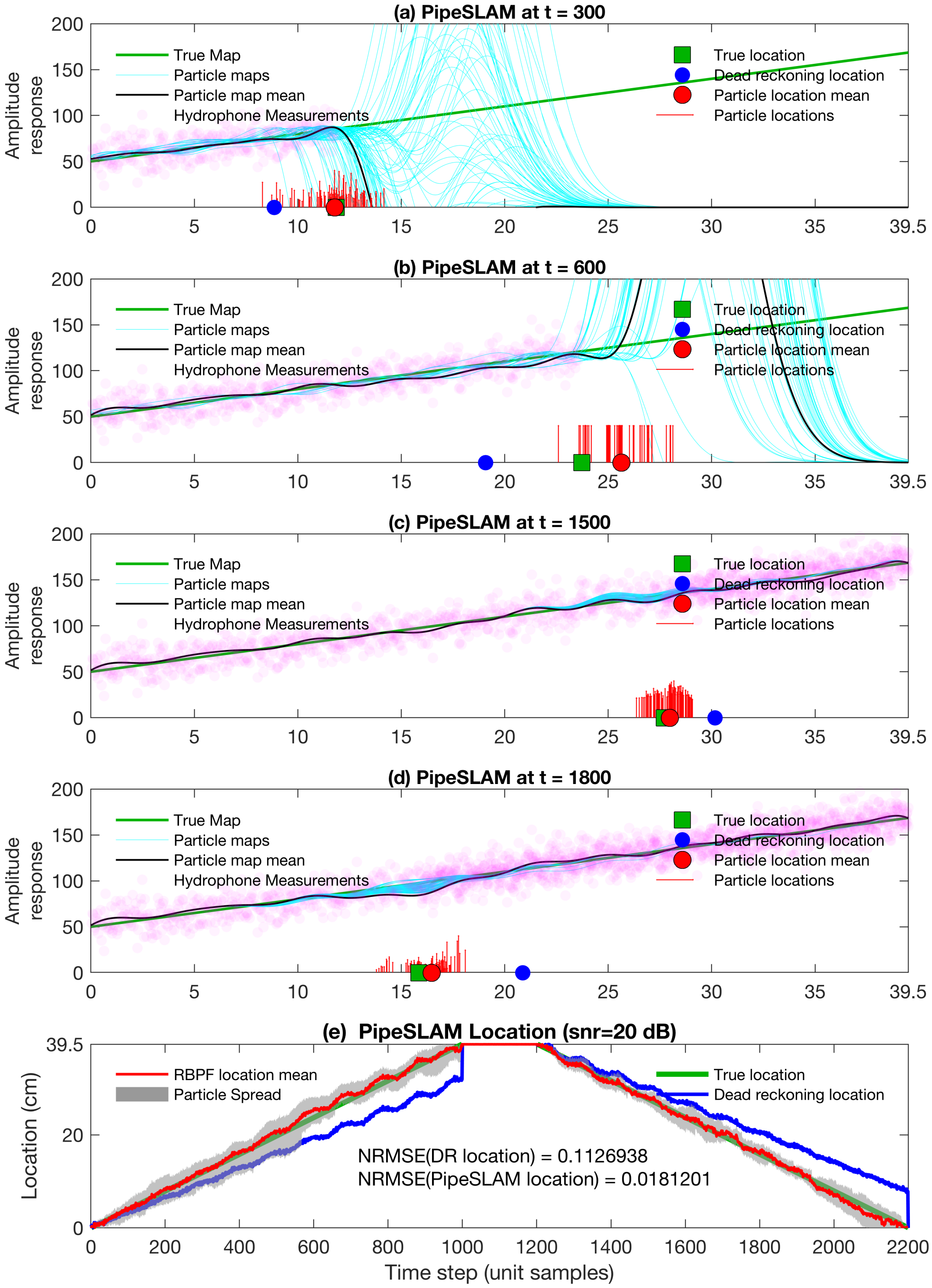


Figure 5.4: PipeSLAM results from simulation of robot movement along a pipe, forwards and backwards, over a ground truth map generated from linear simulation data (snr = 20 dB). (a)-(d) PipeSLAM at increasing time steps of robot movement (a-b: forward movement, left to right; c-d: backward movement, right to left). (e) PipeSLAM localisation results over time, showing the improvement compared to the dead reckoning estimate, which exhibits significant drift.

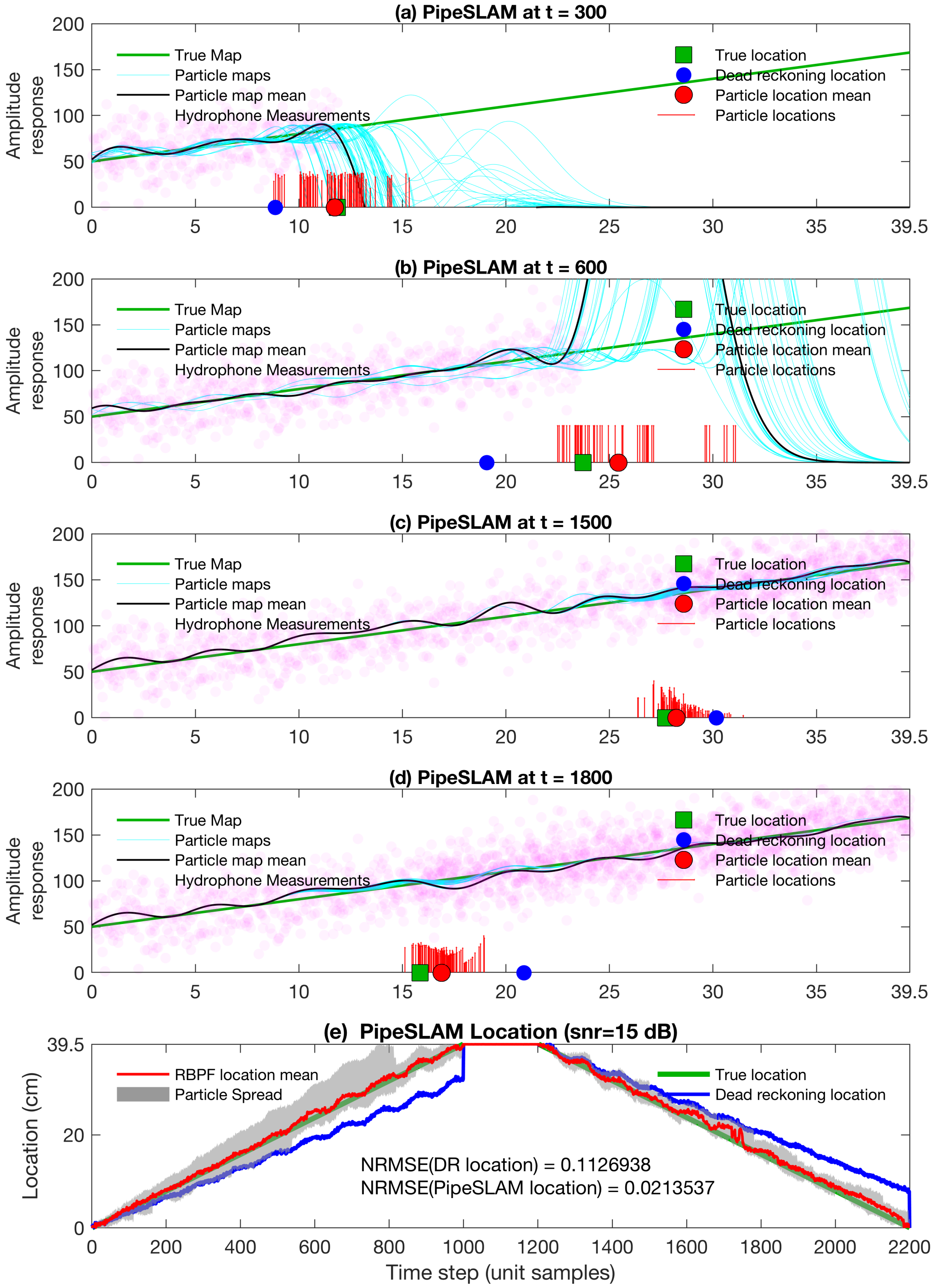


Figure 5.5: PipeSLAM results from simulation of robot movement along a pipe, forwards and backwards, over a ground truth map generated from linear simulation data (snr = 15 dB). (a)-(d) PipeSLAM at increasing time steps of robot movement (a-b: forward movement, left to right; c-d: backward movement, right to left). (e) PipeSLAM localisation results over time, showing the improvement compared to the dead reckoning estimate, which exhibits significant drift.

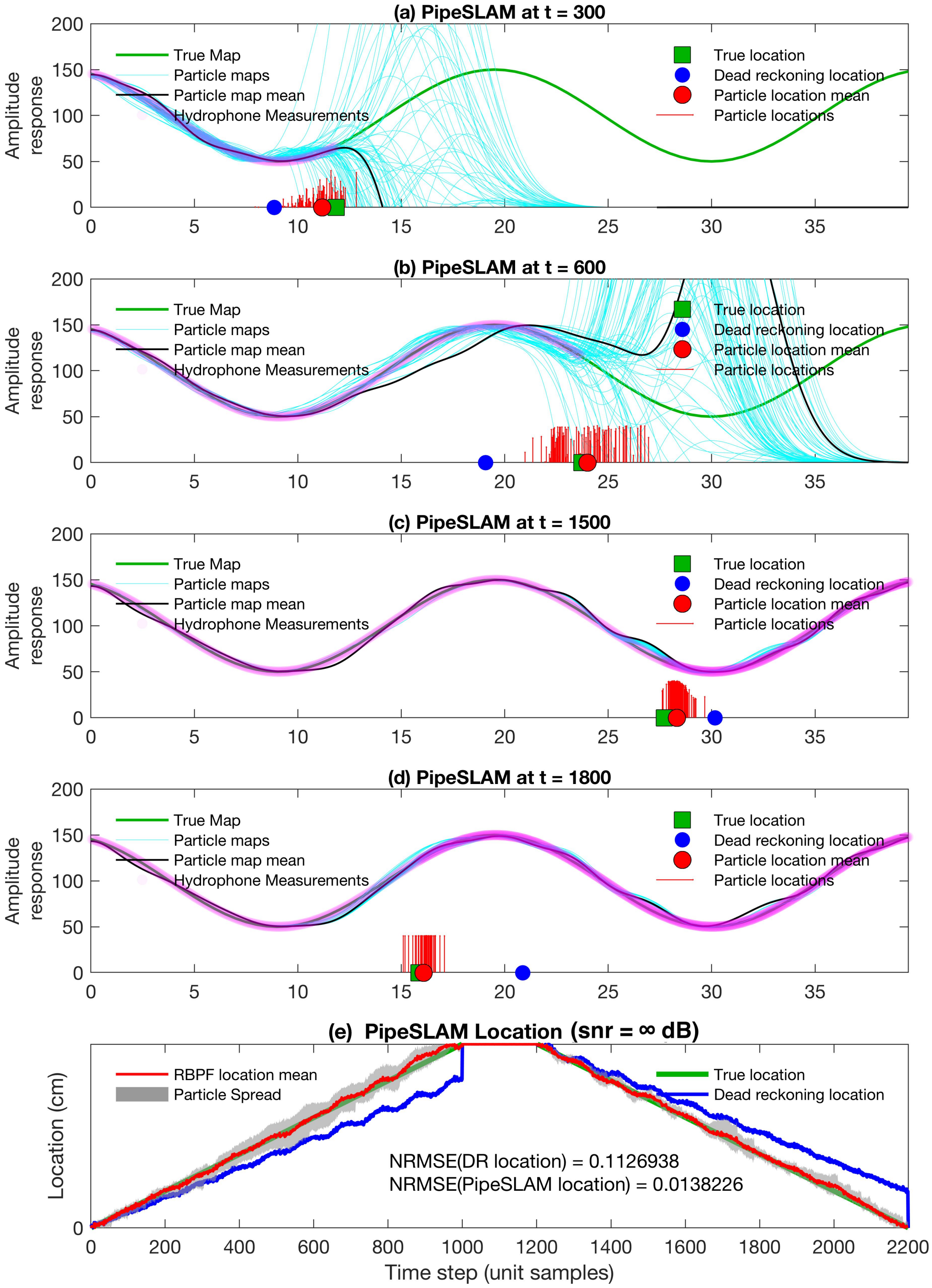


Figure 5.6: PipeSLAM results from simulation of robot movement along a pipe, forwards and backwards, over a ground truth map generated from nonlinear simulation data (snr = ∞ dB). (a)-(d) PipeSLAM at increasing time steps of robot movement (a-b: forward movement, left to right; c-d: backward movement, right to left). (e) PipeSLAM localisation results over time, showing the improvement compared to the dead reckoning estimate, which exhibits significant drift.

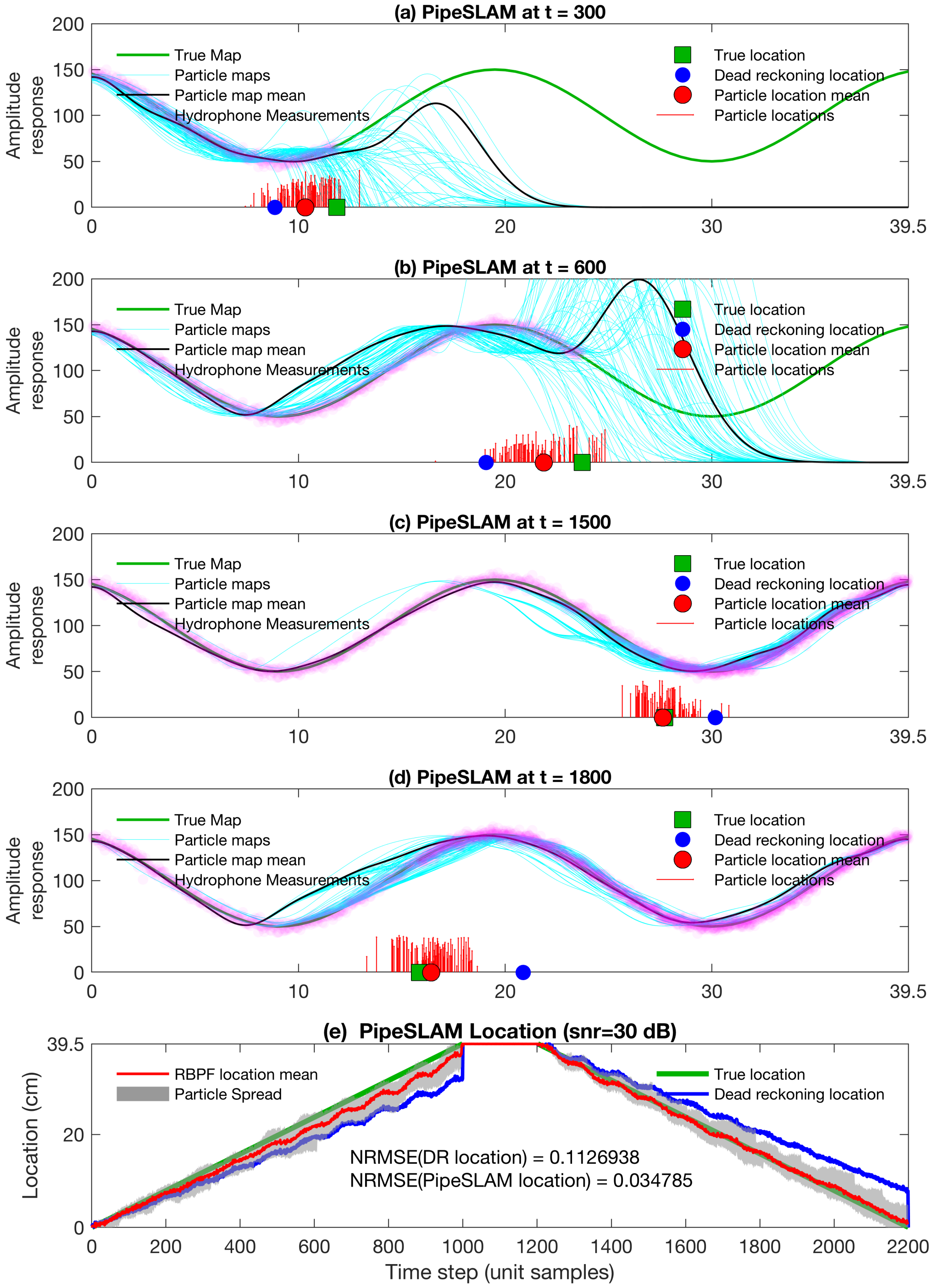


Figure 5.7: PipeSLAM results from simulation of robot movement along a pipe, forwards and backwards, over a ground truth map generated from nonlinear simulation data (snr = 30 dB). (a)-(d) PipeSLAM at increasing time steps of robot movement (a-b: forward movement, left to right; c-d: backward movement, right to left). (e) PipeSLAM localisation results over time, showing the improvement compared to the dead reckoning estimate, which exhibits significant drift.

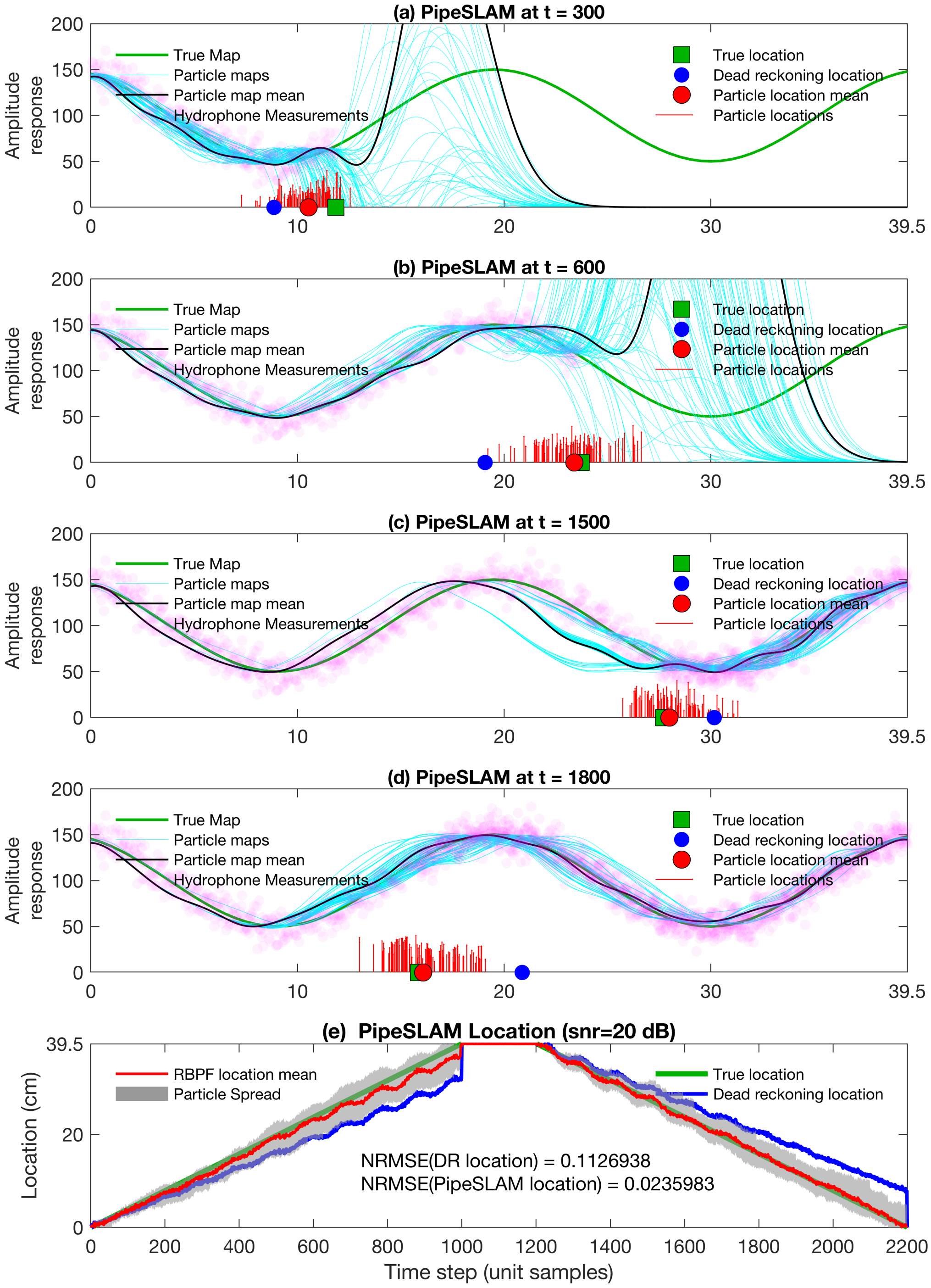


Figure 5.8: PipeSLAM results from simulation of robot movement along a pipe, forwards and backwards, over a ground truth map generated from nonlinear simulation data (snr = 20 dB). (a)-(d) PipeSLAM at increasing time steps of robot movement (a-b: forward movement, left to right; c-d: backward movement, right to left). (e) PipeSLAM localisation results over time, showing the improvement compared to the dead reckoning estimate, which exhibits significant drift.

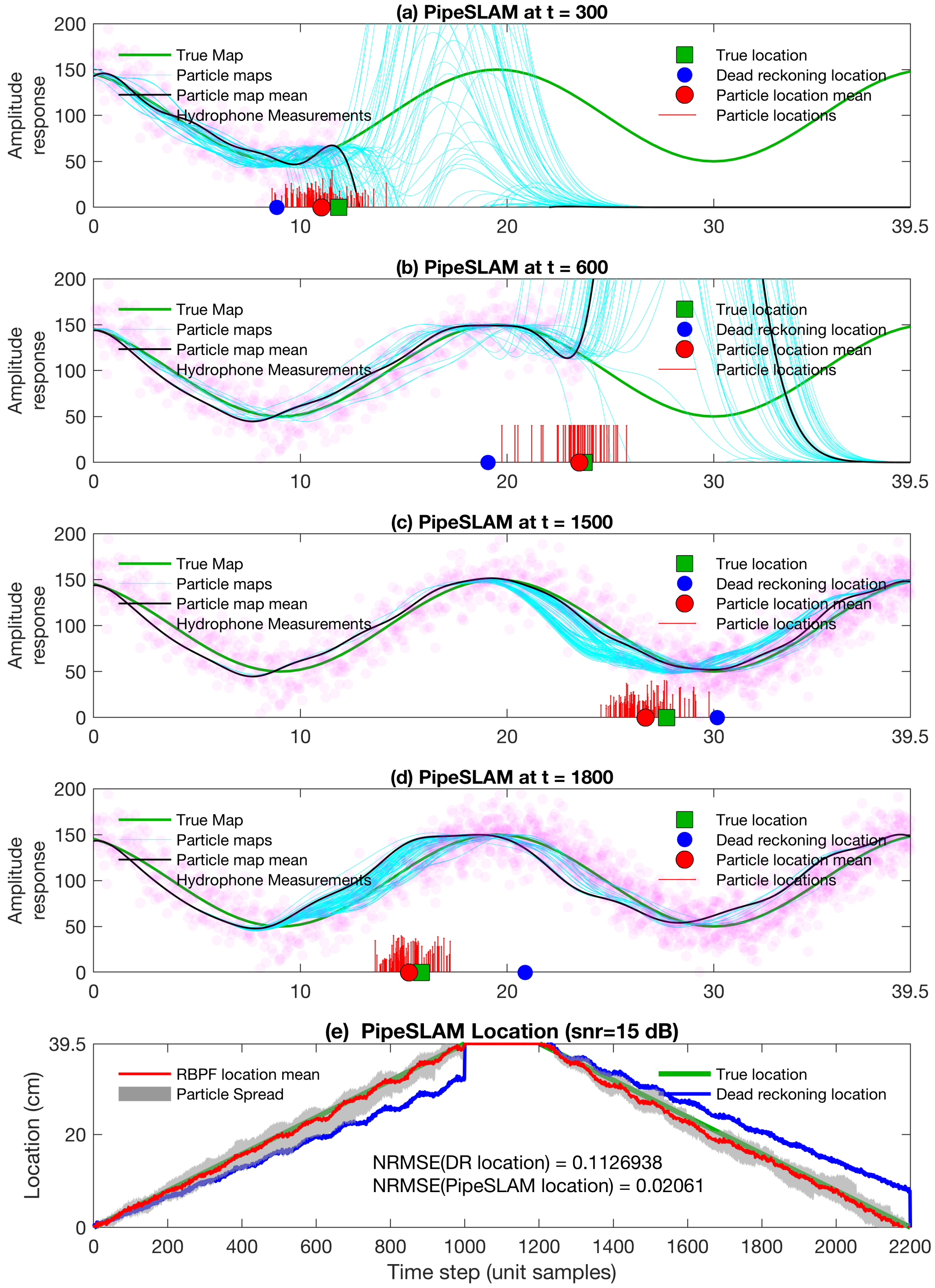


Figure 5.9: PipeSLAM results from simulation of robot movement along a pipe, forwards and backwards, over a ground truth map generated from nonlinear simulation data (snr = 15 dB). (a)-(d) PipeSLAM at increasing time steps of robot movement (a-b: forward movement, left to right; c-d: backward movement, right to left). (e) PipeSLAM localisation results over time, showing the improvement compared to the dead reckoning estimate, which exhibits significant drift.

’feature’ areas but will slightly drift in those ’flat’ regions as expected. The overall map estimation for plastic pipe data is not as good as over the map generated from simulated linear and nonlinear data.

When the robot travels through the pipe for the first time (left to right), it has no prior knowledge of the map. Therefore, the particle uncertainty spreads largely over time. But when it reaches to the right end of the pipe, it resets all particle locations and maps and acquired knowledge of the map. This knowledge can be viewed as a prior when it moves backwards. With a prior of the map, PipeSLAM location uncertainty reduces significantly (shown in the right half in panel (e) in those results figures), while the dead reckoning is still drifted largely over time.

5.3.3 Results from Metal Water Pipe Data

In order to test and evaluate the PipeSLAM algorithm, 3 sets of experimental pipe vibration data is also used to define the ground truth map. These 3 sets of experiment data are: 1) 1-metre metal pipe data, 2) 5-metre metal pipe data I, and 3) 5-metre metal pipe data II. It simulates the robot moving forward and backward over this data to analyse the PipeSLAM algorithm. Drift is applied to the robot simulation to make the SLAM problem more challenging and the map used in the PipeSLAM algorithm is initialised with zero weights. Results are shown in Fig.5.14 $\sim$ Fig.5.27.

In addition, all experiments assume the robot is travelling from the left end to the right end through the length of the metal pipe. An additional evaluation is to use an reverse data set, i.e. by reconstructing data I to assume the robot is travelling from the right end to the left end and then return to the right end. In this case, the right end location is initialised to 0 cm. When the robot reaches the left end, it reaches the maximum distance travelled location (499.5 cm). Results are shown in Fig.5.22 $\sim$ Fig.5.23.

The results obtained from the simulation demonstrate how the PipeSLAM algorithm learns the map: each particle used to represent robot location has an associated map based on the trajectories of the particles – these separate maps are clearly seen in panels (a)-(b) in those result figures. On the backward pass along the simulated pipe the maps have converged to similar values (see panels (c)-(d) in those result figures).

Regarding localisation, on the forward pass, whilst the map is unknown there is relatively large uncertainty in the robot location. Then on the backward pass the uncertainty reduces (see panels (e) in those result figures). Quantitatively, the PipeSLAM algorithm is able to localise more accurately than dead reckoning, which demonstrates the usefulness of the PipeSLAM algorithm.

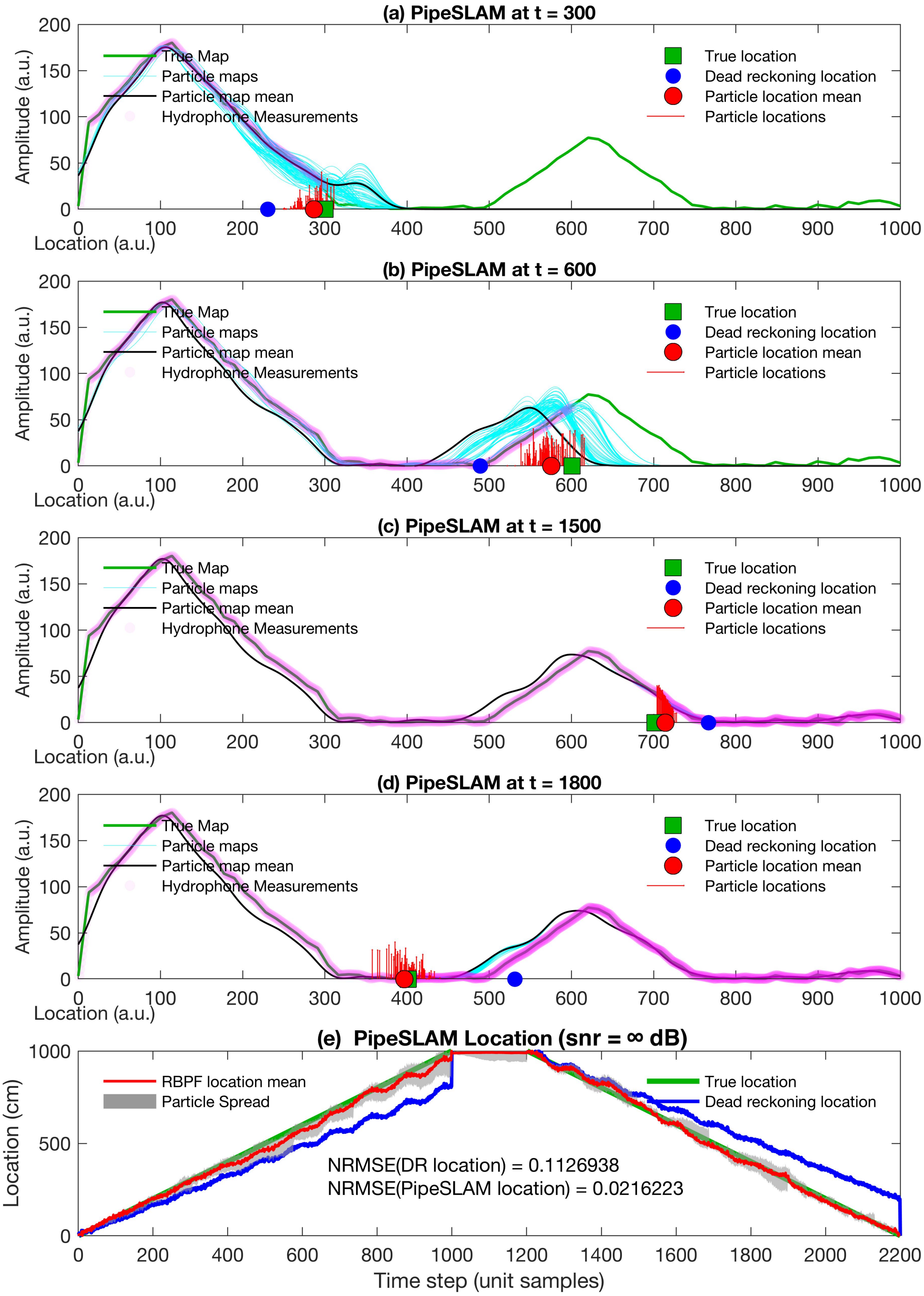


Figure 5.10: PipeSLAM using plastic pipe data. Signal to noise ratio (snr) = ∞ dB. (a)-(d) PipeSLAM at different time steps building the map while localising. (e) PipeSLAM location estimates much improved accuracy compared with dead reckoning locations.

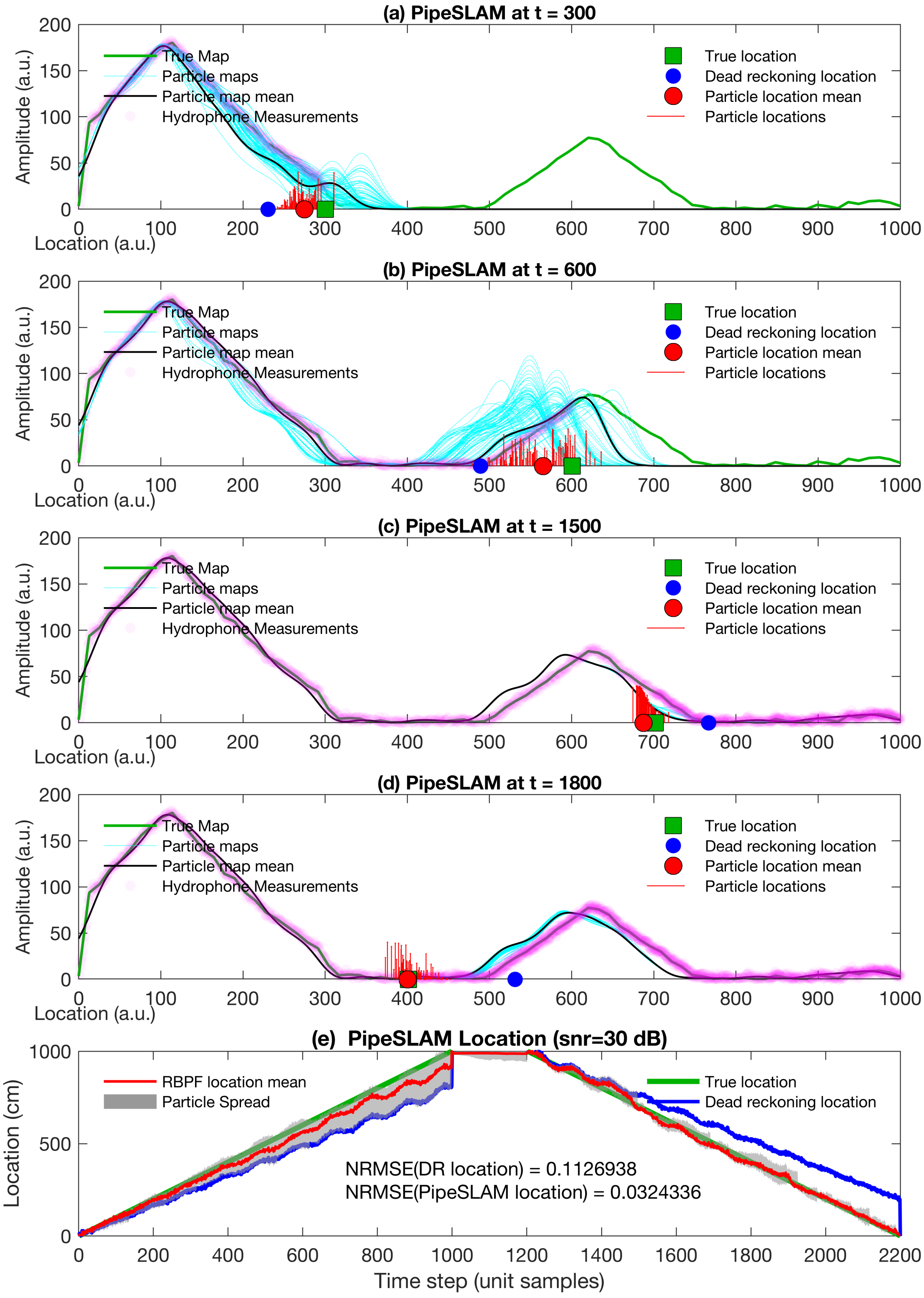


Figure 5.11: PipeSLAM using plastic pipe data. Signal to noise ratio (snr) = 30 dB. (a)-(d) PipeSLAM at different time steps building the map while localising. (e) PipeSLAM location estimates much improved accuracy compared with dead reckoning locations.

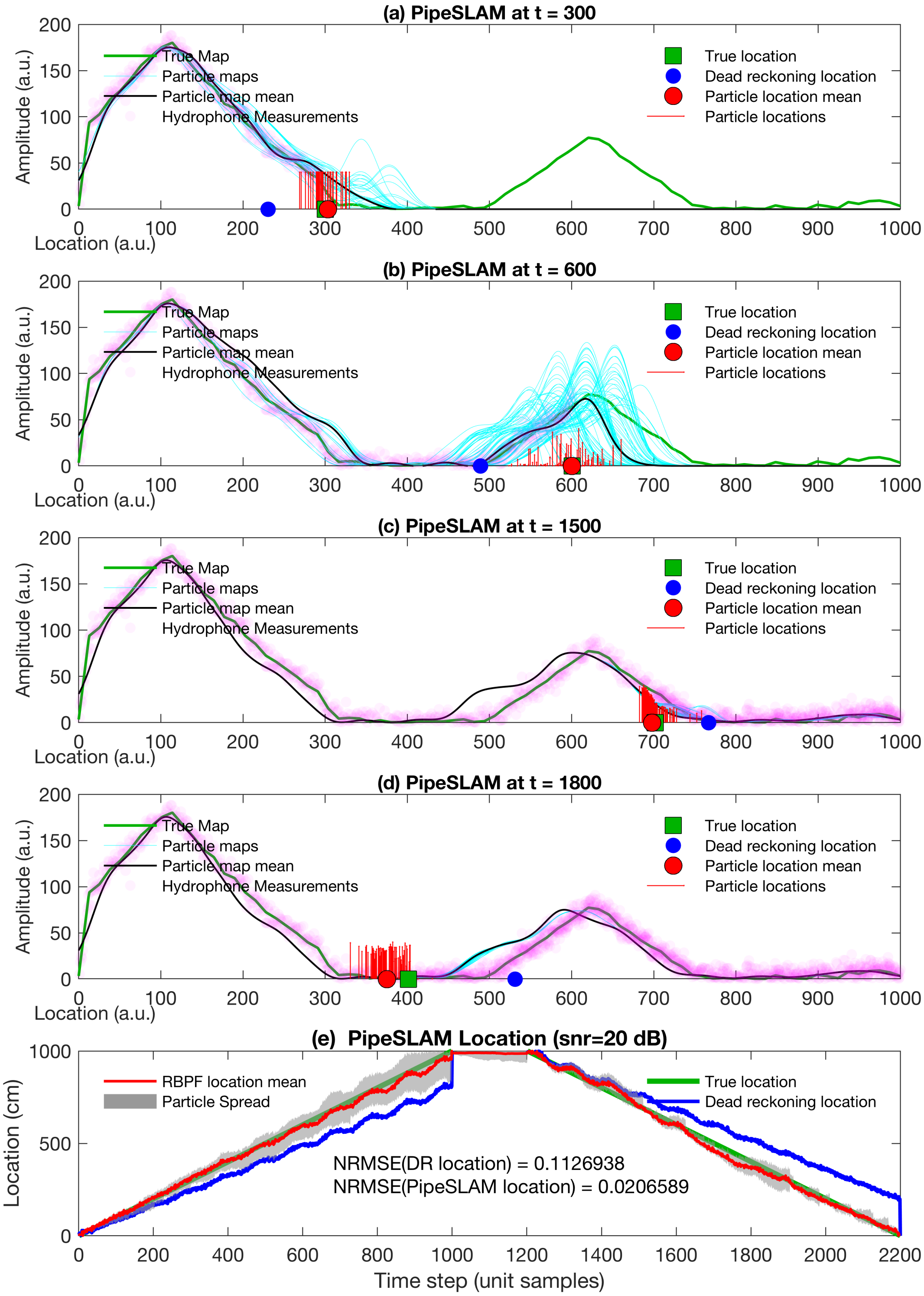


Figure 5.12: PipeSLAM using plastic pipe data. Signal to noise ratio (snr) = 20 dB. (a)-(d) PipeSLAM at different time steps building the map while localising. (e) PipeSLAM location estimates much improved accuracy compared with dead reckoning locations.

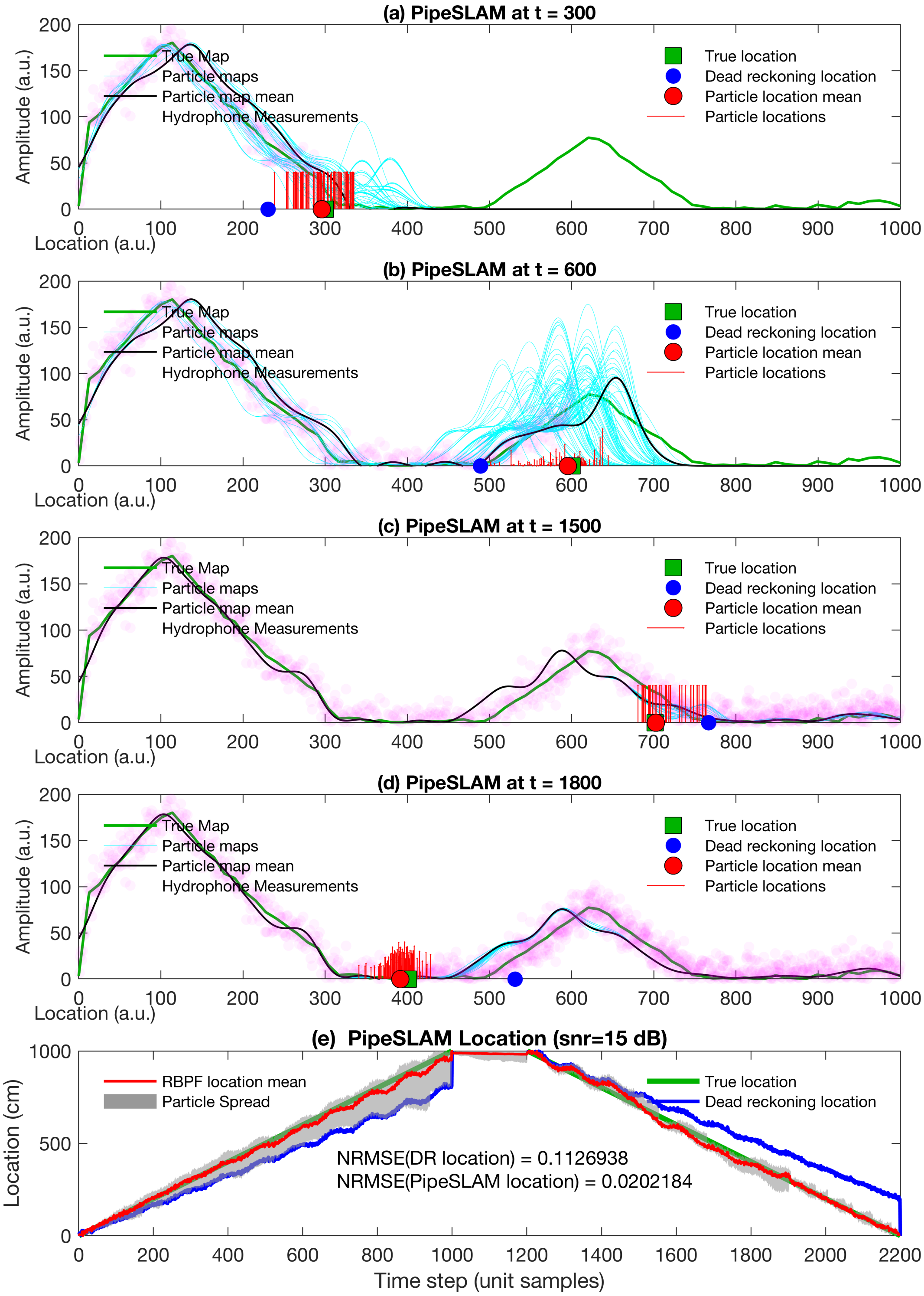


Figure 5.13: PipeSLAM using plastic pipe data. Signal to noise ratio (snr) = 15 dB. (a)-(d) PipeSLAM at different time steps building the map while localising. (e) PipeSLAM location estimates much improved accuracy compared with dead reckoning locations.

Further quantitative results demonstrating the effectiveness of the PipeSLAM algorithm in comparison to dead reckoning are shown in Table 5.3 and Table 5.4.

5.3.4 Discussion

PipeSLAM results using different simulation and experiment data sets with different white noises have been shown above. It is still not quite clear to determine if the PipeSLAM algorithm is better or the sequential mapping and localisation algorithm (in Chapter 4) is better. Therefore, this part will put all results together to show the performances of the two algorithms.

A full location estimation results of the two algorithms are listed and compared in Table.5.3 and Table 5.4. A visual comparison of these results are demonstrated in Fig.5.28 $\sim$ Fig.5.31.

For simulated linear and nonlinear data (see Fig.5.28), PipeSLAM location estimation performance is not as good as EKF and PF location estimation that the blue PipeSLAM location errors are larger than red and purple lines. But compared to the same dead reckoning location error (black line on the top of each figure), PipeSLAM has still performed well. Also, the location errors of both EKF and PF estimation smaller for linear simulation data than nonlinear simulation data (dashed red line is lower than solid red line, dashed purple line is lower than solid purple line). This demonstrates that EKF and PF estimators are good at processing linear maps compared to nonlinear maps. However, this is not the case for PipeSLAM location estimation that dashed blue line (PipeSLAM using nonlinear simulation data) is always on top of solid blue line (PipeSLAM using linear simulation data).

For plastic pipe data (Fig.5.29), 1-metre metal pipe data (Fig.5.30) and 5-metre metal pipe data I (see Fig.5.31), their location errors along different white noises are similar. However, it can be seen that both EKF and PF location errors are relatively smooth along different noise ratios. But PipeSLAM location errors are oscillated along different noise ratios. One possible reason is that the PipeSLAM have not run enough times for different random numbers, as the Rao-Blackwellised particle filter used in PipeSLAM is a probabilistic method. After running large number of times, the mean location error for PipeSLAM may be more smooth. But this also demonstrates that PipeSLAM is not as robust as the sequential mapping and localisation method demonstrated in Chapter 4.

However, compared with sequential mapping and localisation method, which needs to run the robot 20 times forwards and backwards in a pipe, PipeSLAM only needs a single run. This largely saves time and costs if it is deployed in a industrial application. This experiment has not tested the PipeSLAM for multiple runs at

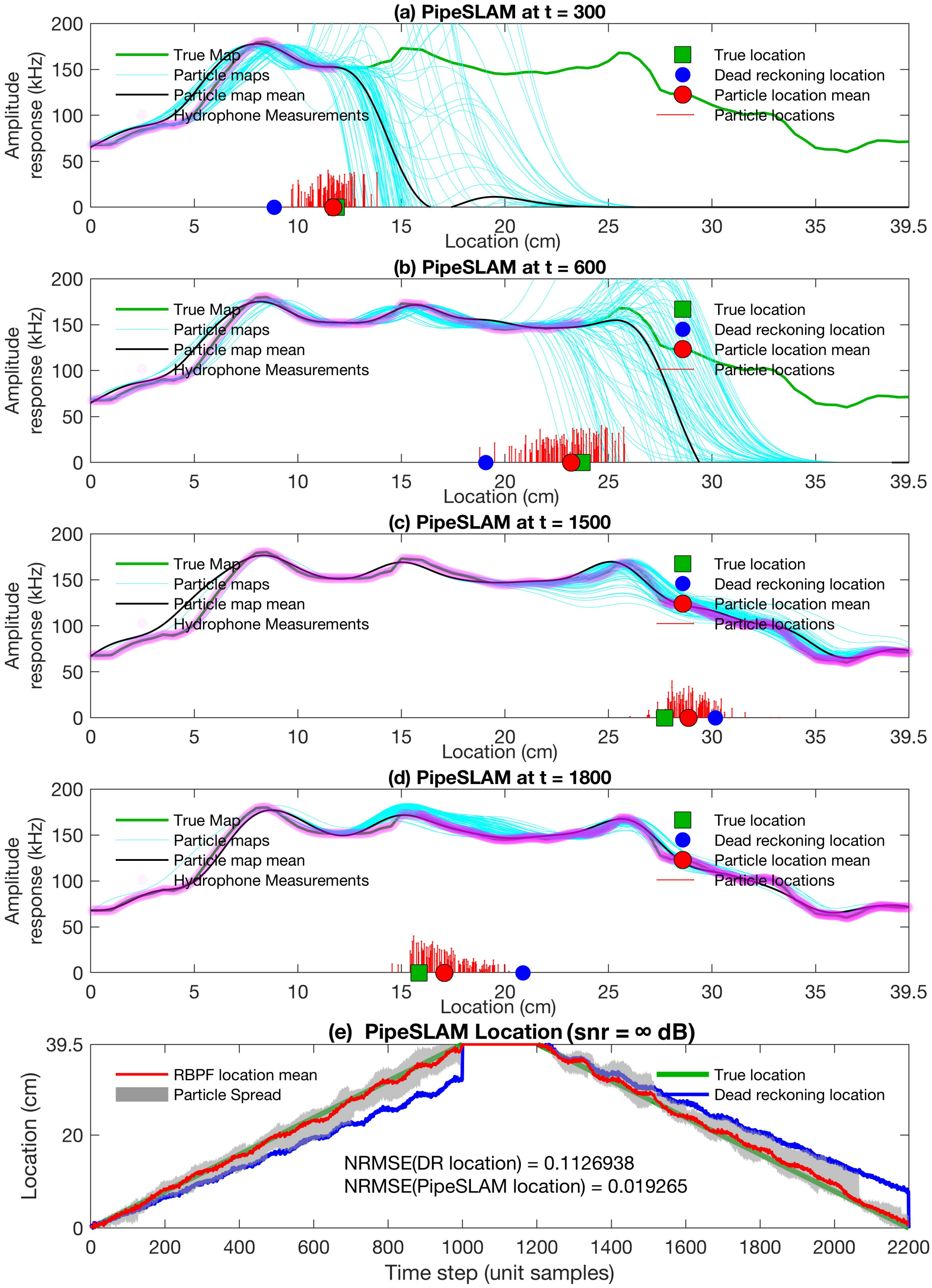


Figure 5.14: PipeSLAM using 1-metre metal pipe data. Signal to noise ratio (snr) = ∞ dB. (a)-(d) PipeSLAM at different time steps building the map while localising. (e) PipeSLAM location estimates much improved accuracy compared with dead reckoning locations.

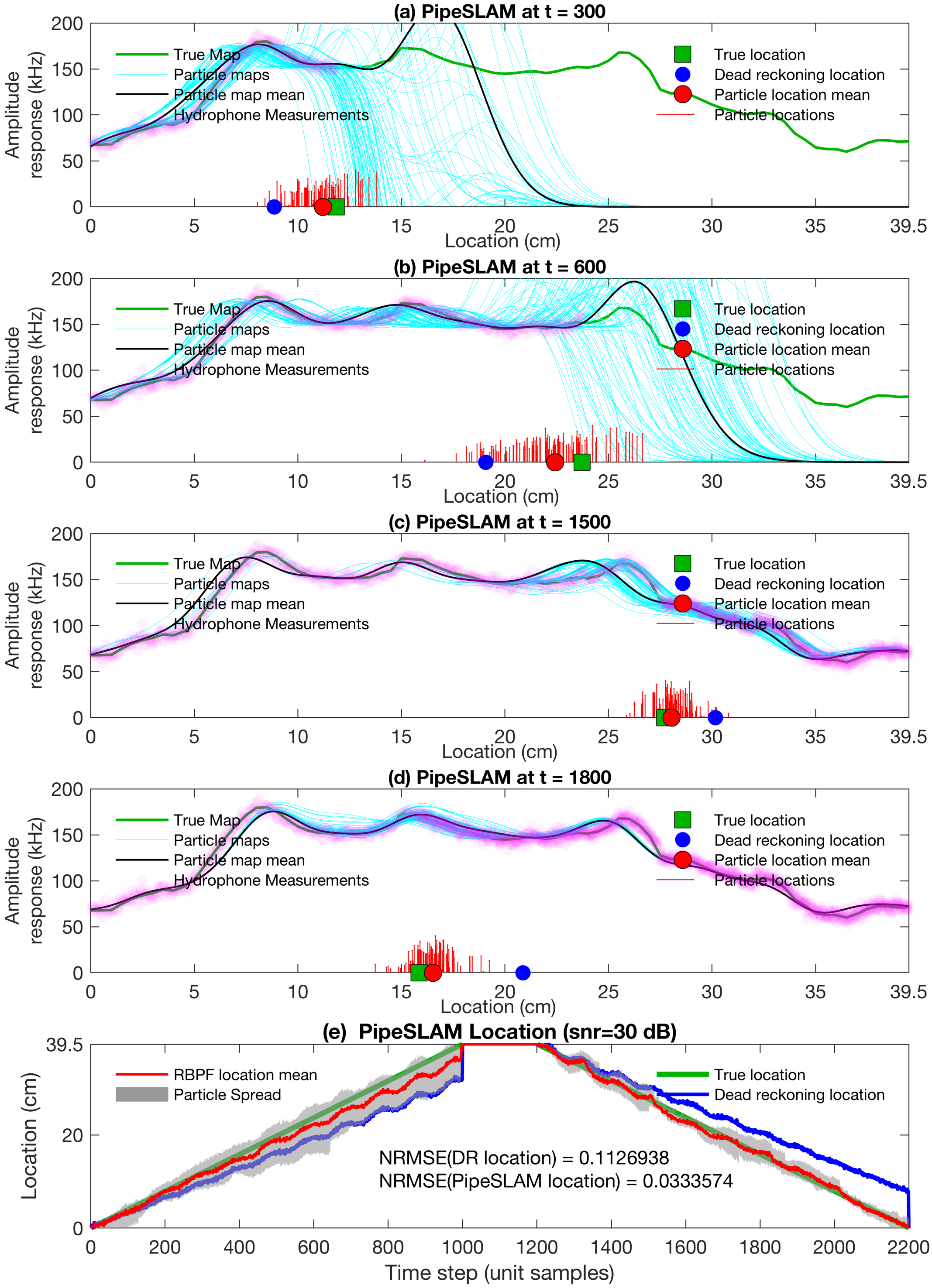


Figure 5.15: PipeSLAM using 1-metre metal pipe data. Signal to noise ratio (snr) = 30 dB. (a)-(d) PipeSLAM at different time steps building the map while localising. (e) PipeSLAM location estimates much improved accuracy compared with dead reckoning locations.

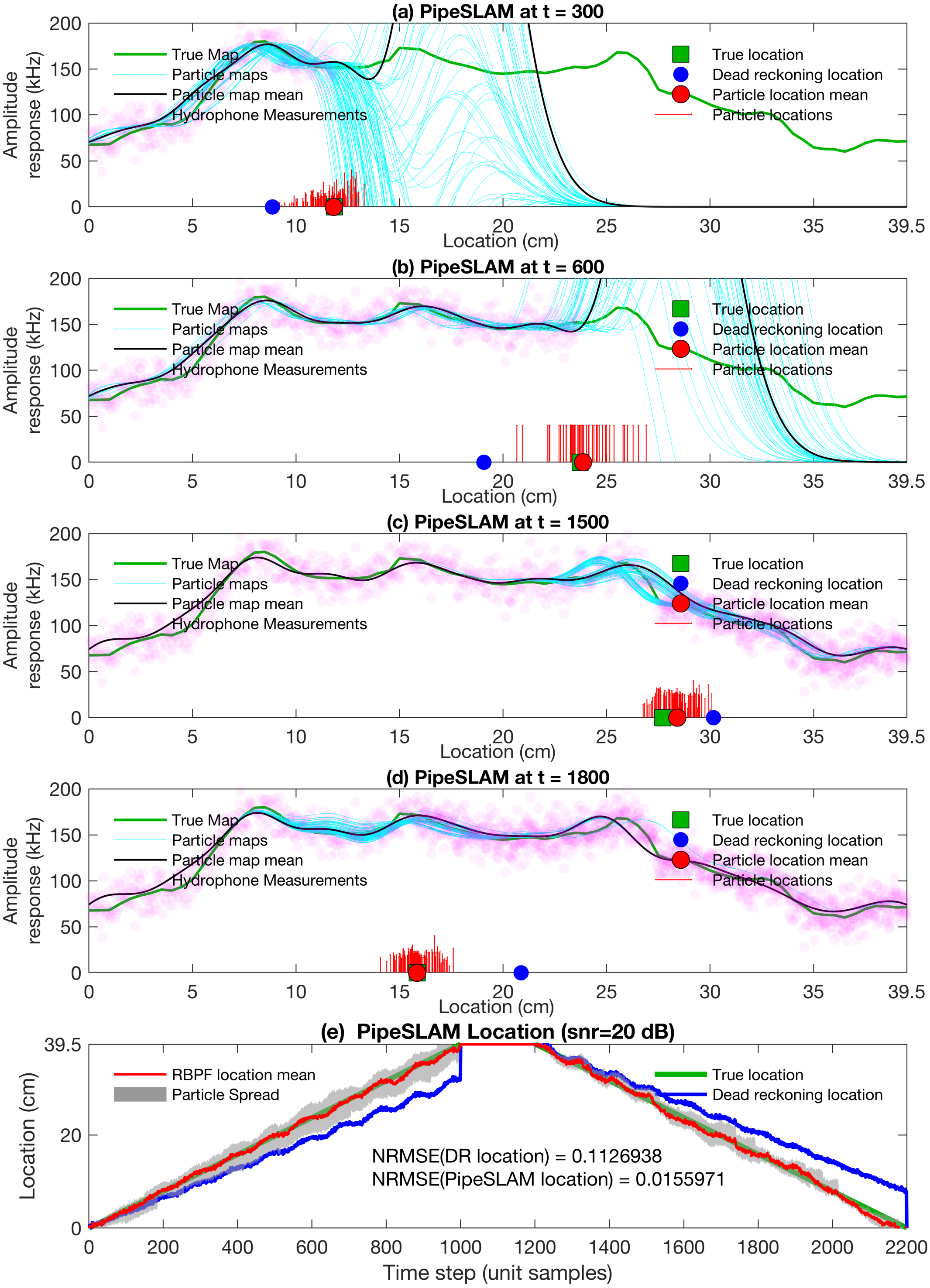


Figure 5.16: PipeSLAM using 1-metre metal pipe data. Signal to noise ratio (snr) = 20 dB. (a)-(d) PipeSLAM at different time steps building the map while localising. (e) PipeSLAM location estimates much improved accuracy compared with dead reckoning locations.

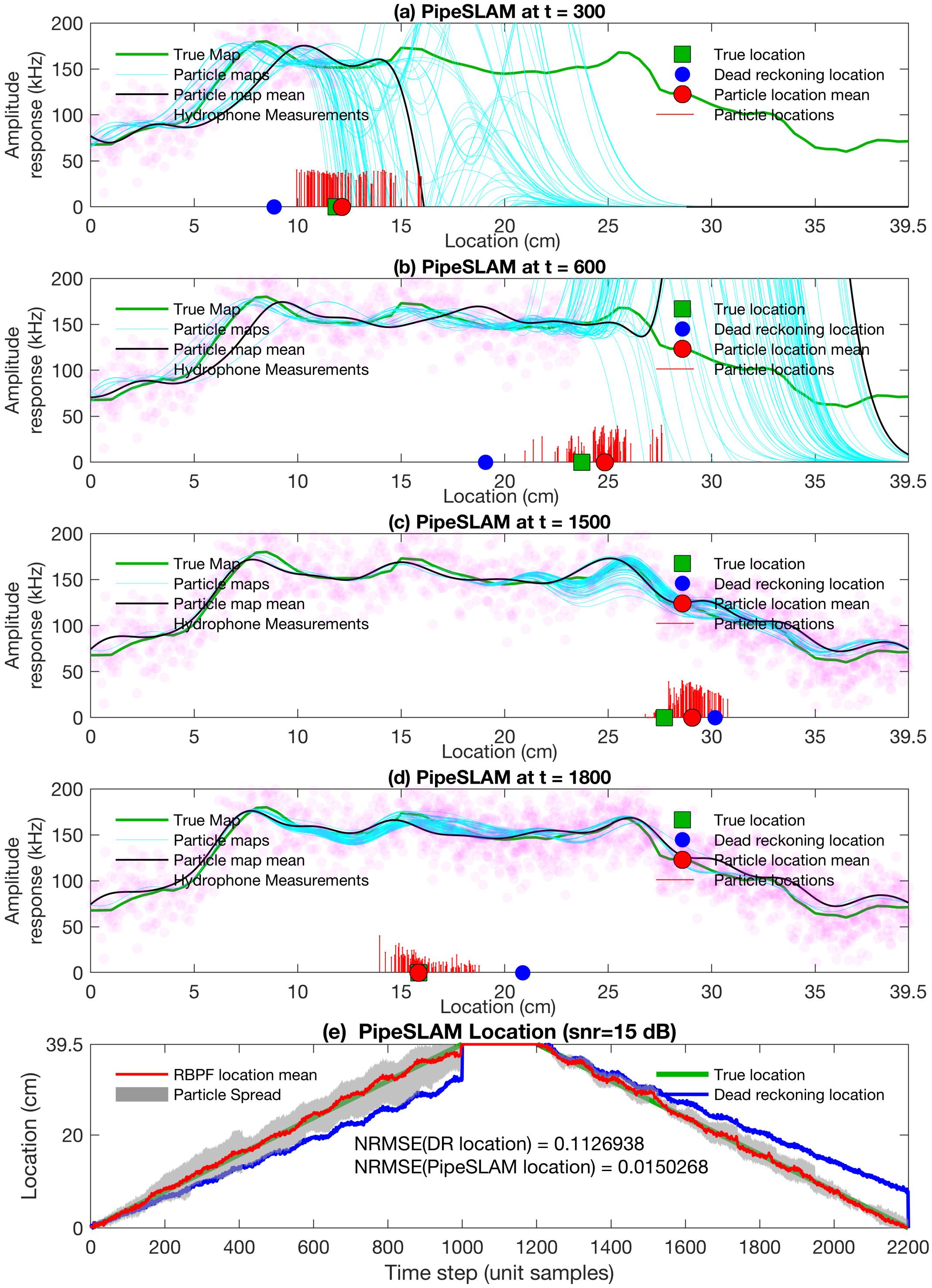


Figure 5.17: PipeSLAM using 1-metre metal pipe data. Signal to noise ratio (snr) = 15 dB. (a)-(d) PipeSLAM at different time steps building the map while localising. (e) PipeSLAM location estimates much improved accuracy compared with dead reckoning locations.

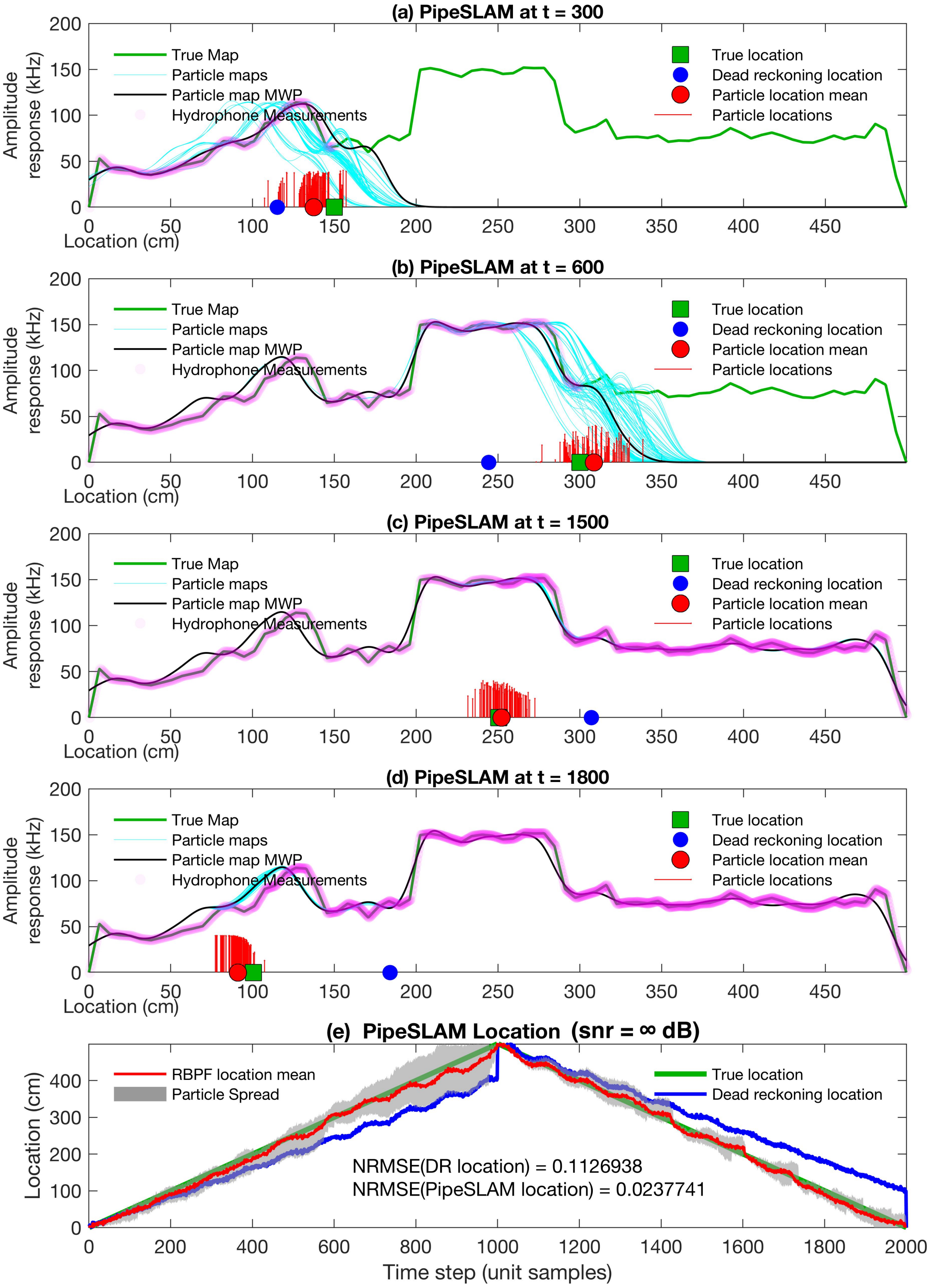


Figure 5.18: PipeSLAM using 5-metre metal pipe data I. Signal to noise ratio (snr) = ∞ dB. (a)-(d) PipeSLAM at different time steps building the map while localising. (e) PipeSLAM location estimates much improved accuracy compared with dead reckoning locations.

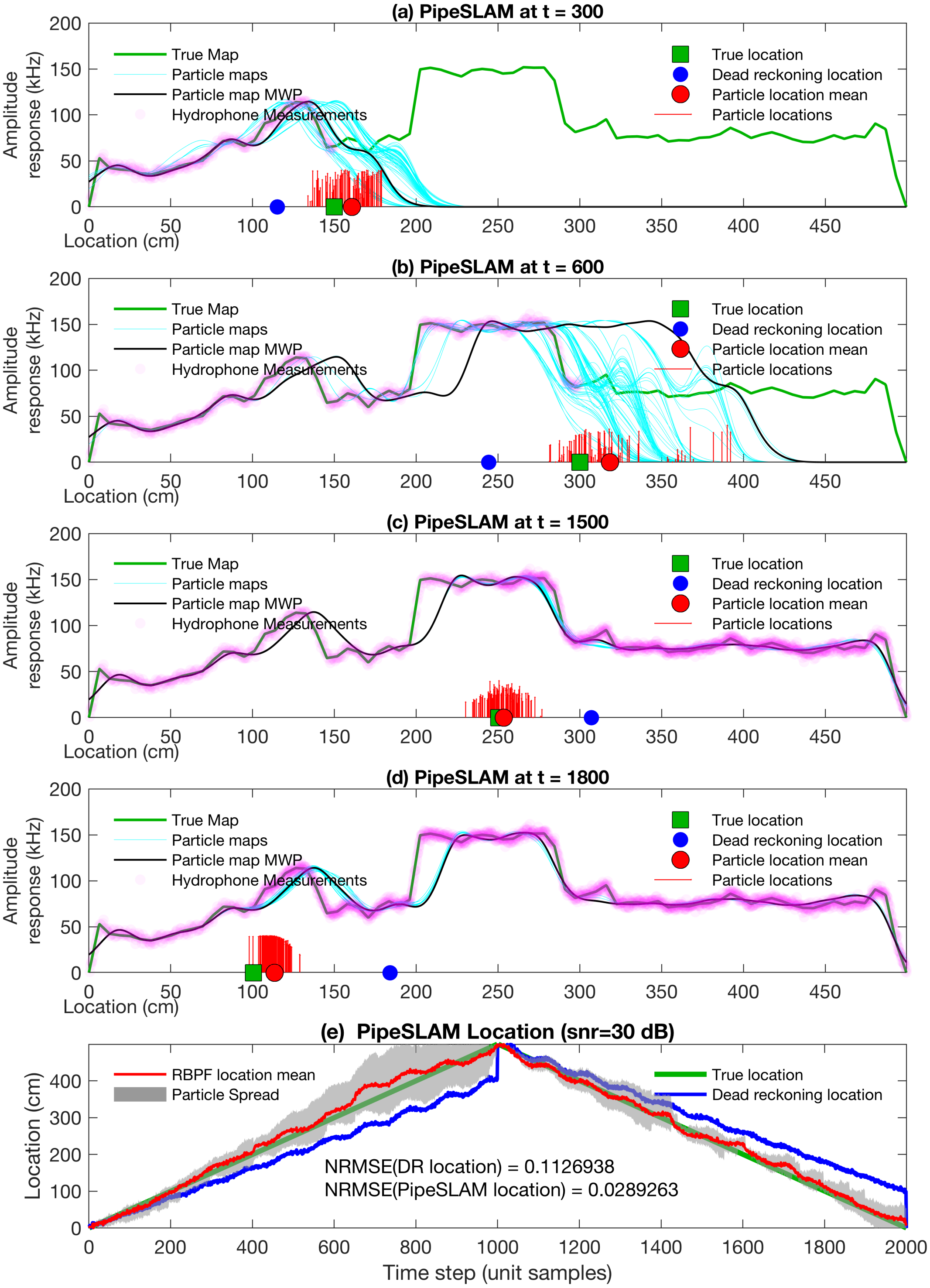


Figure 5.19: PipeSLAM using 5-metre metal pipe data I. Signal to noise ratio (snr) = 30 dB. (a)-(d) PipeSLAM at different time steps building the map while localising. (e) PipeSLAM location estimates much improved accuracy compared with dead reckoning locations.

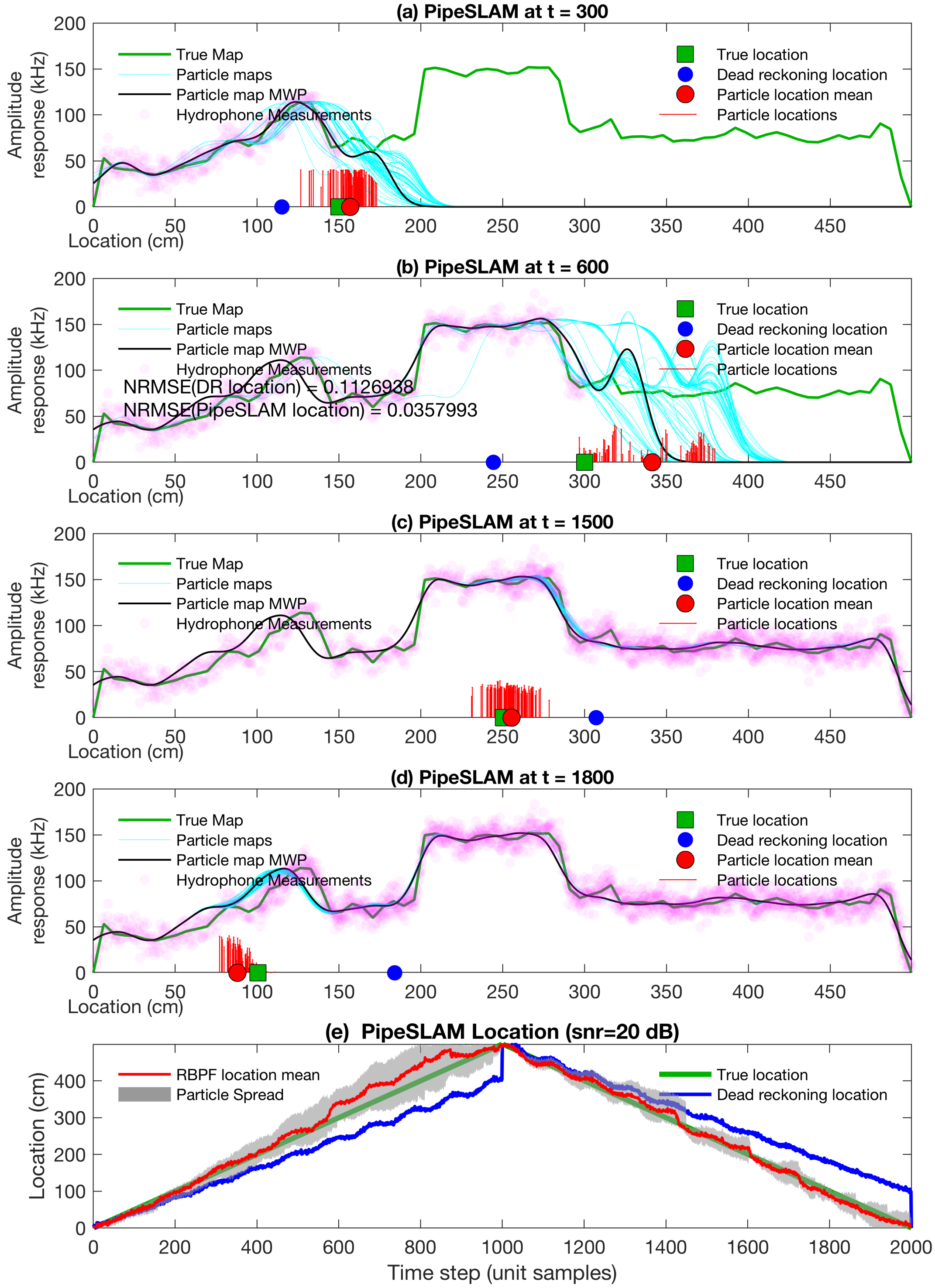


Figure 5.20: PipeSLAM using 5-metre metal pipe data I. Signal to noise ratio (snr) = 20 dB. (a)-(d) PipeSLAM at different time steps building the map while localising. (e) PipeSLAM location estimates much improved accuracy compared with dead reckoning locations.

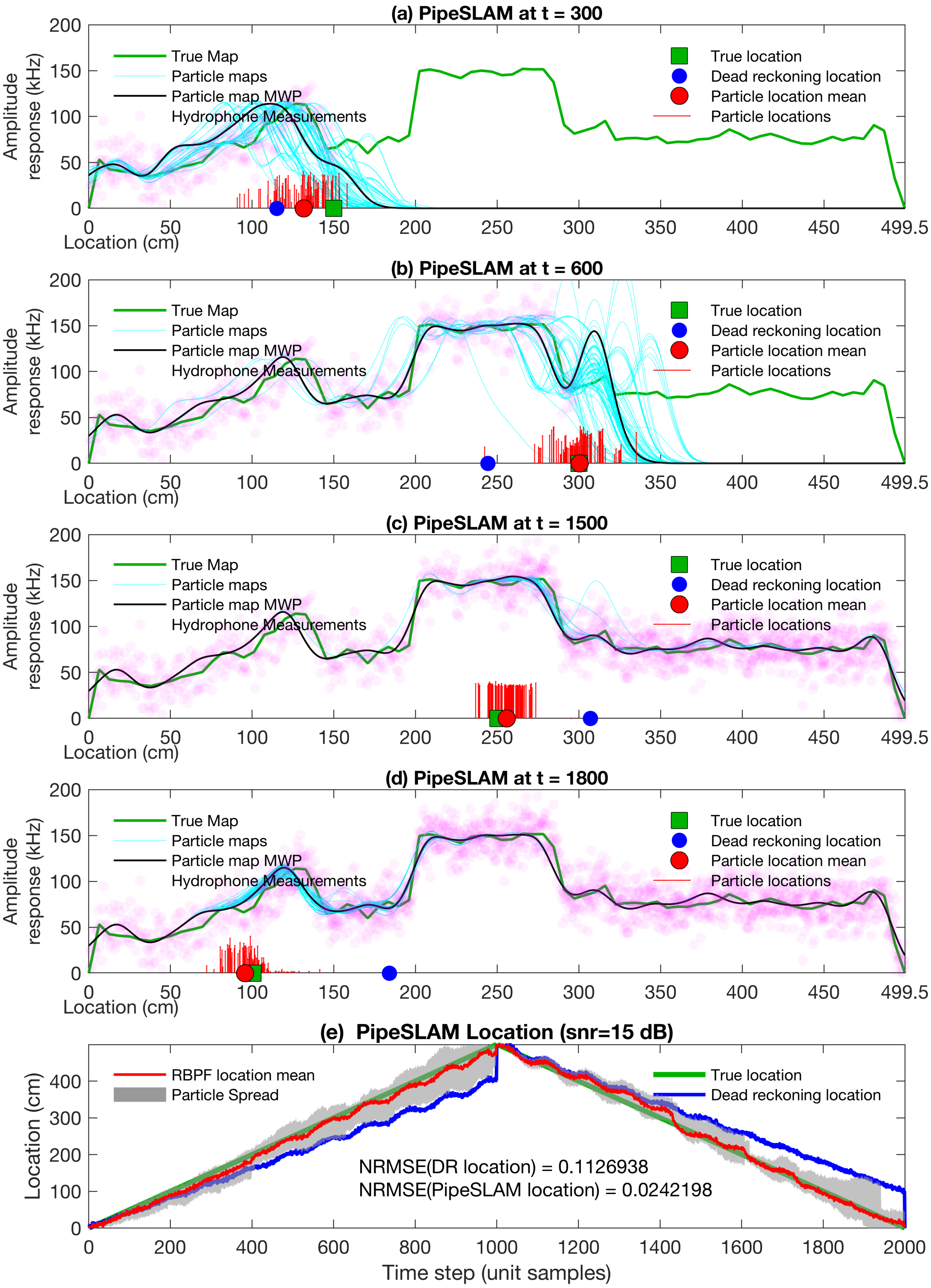


Figure 5.21: PipeSLAM using 5-metre metal pipe data I. Signal to noise ratio (snr) = 15 dB. (a)-(d) PipeSLAM at different time steps building the map while localising. (e) PipeSLAM location estimates much improved accuracy compared with dead reckoning locations.

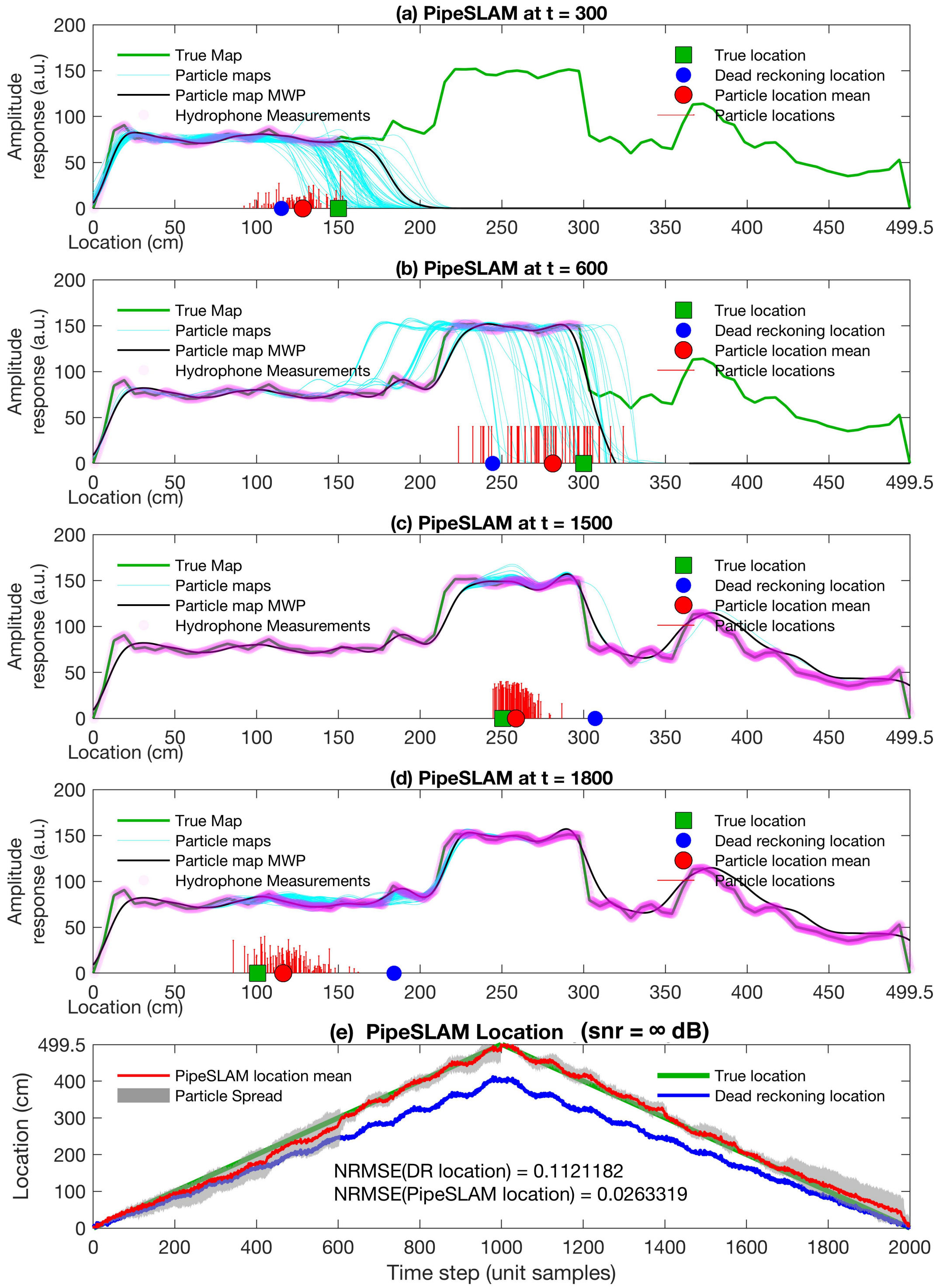


Figure 5.22: PipeSLAM using reverse 5-metre metal pipe data I. Signal to noise ratio (snr) = ∞ dB. (a)-(d) PipeSLAM at different time steps building the map while localising. (e) PipeSLAM location estimates much improved accuracy compared with dead reckoning locations.

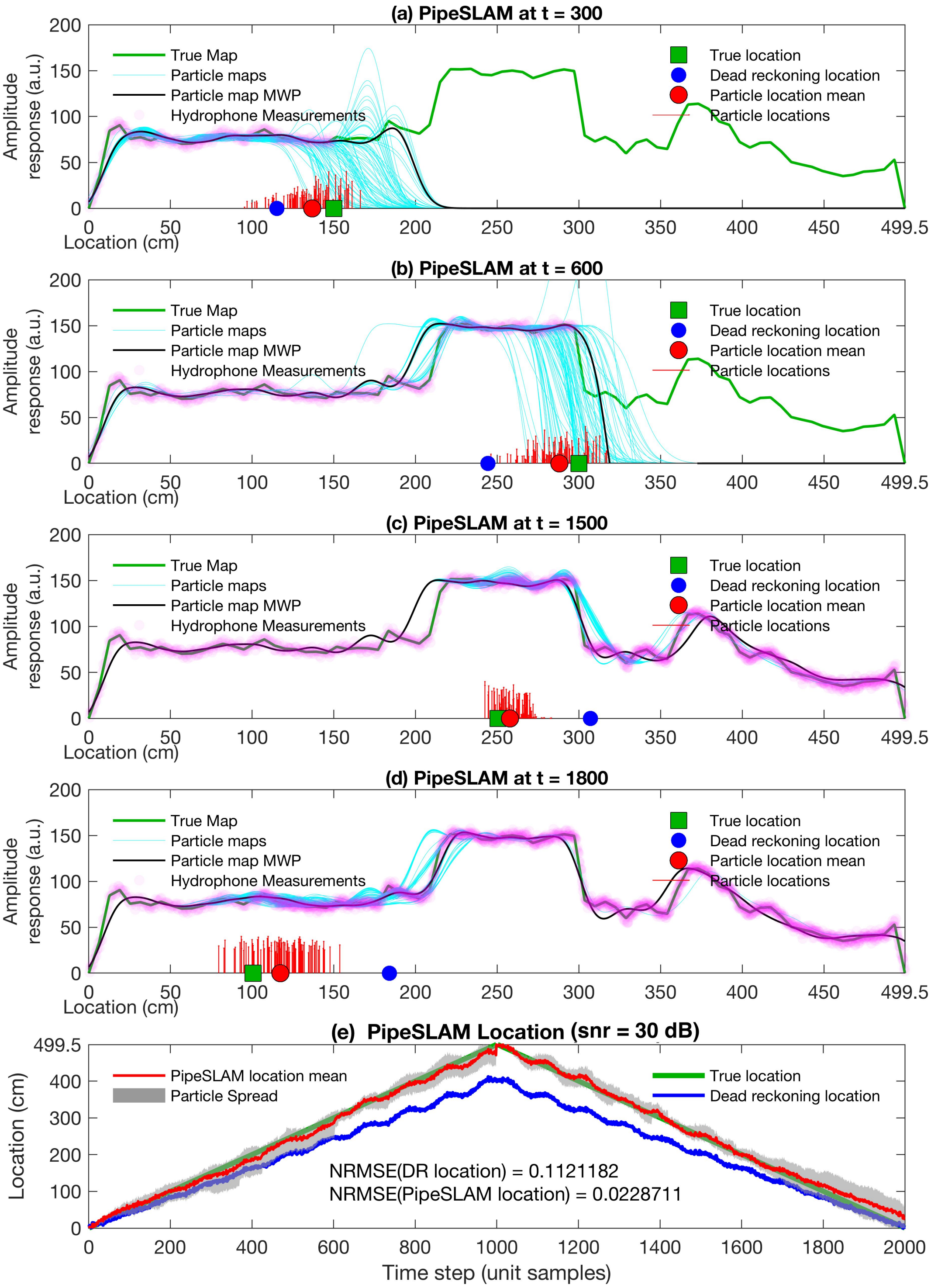


Figure 5.23: PipeSLAM using reverse 5-metre metal pipe data I. Signal to noise ratio (snr) = 30 dB. (a)-(d) PipeSLAM at different time steps building the map while localising. (e) PipeSLAM location estimates much improved accuracy compared with dead reckoning locations.

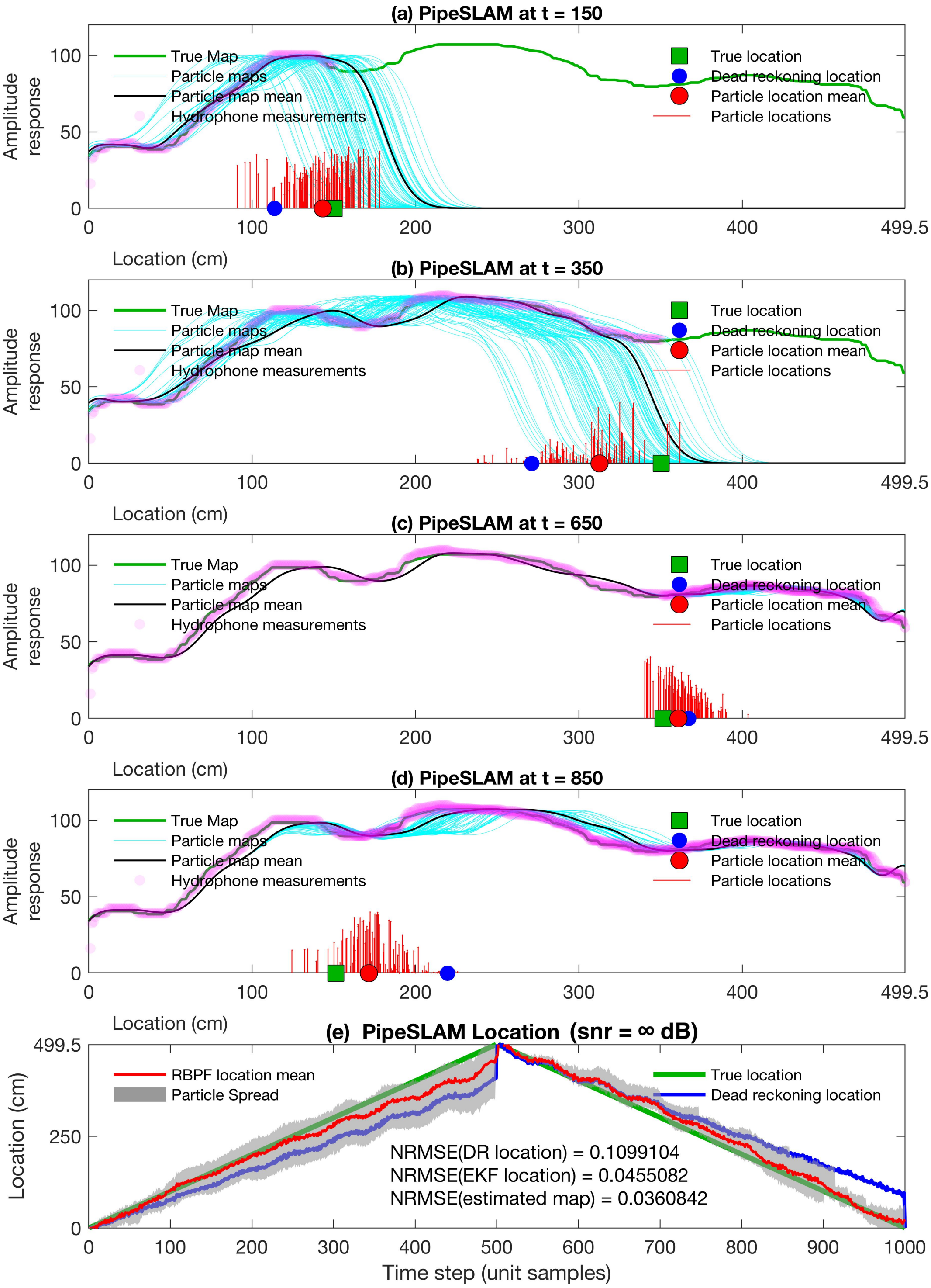


Figure 5.24: PipeSLAM using 5-metre metal pipe data II. Signal to noise ratio (snr) = ∞ dB. (a)-(d) PipeSLAM at different time steps building the map while localising. (e) PipeSLAM location estimates much improved accuracy compared with dead reckoning locations.

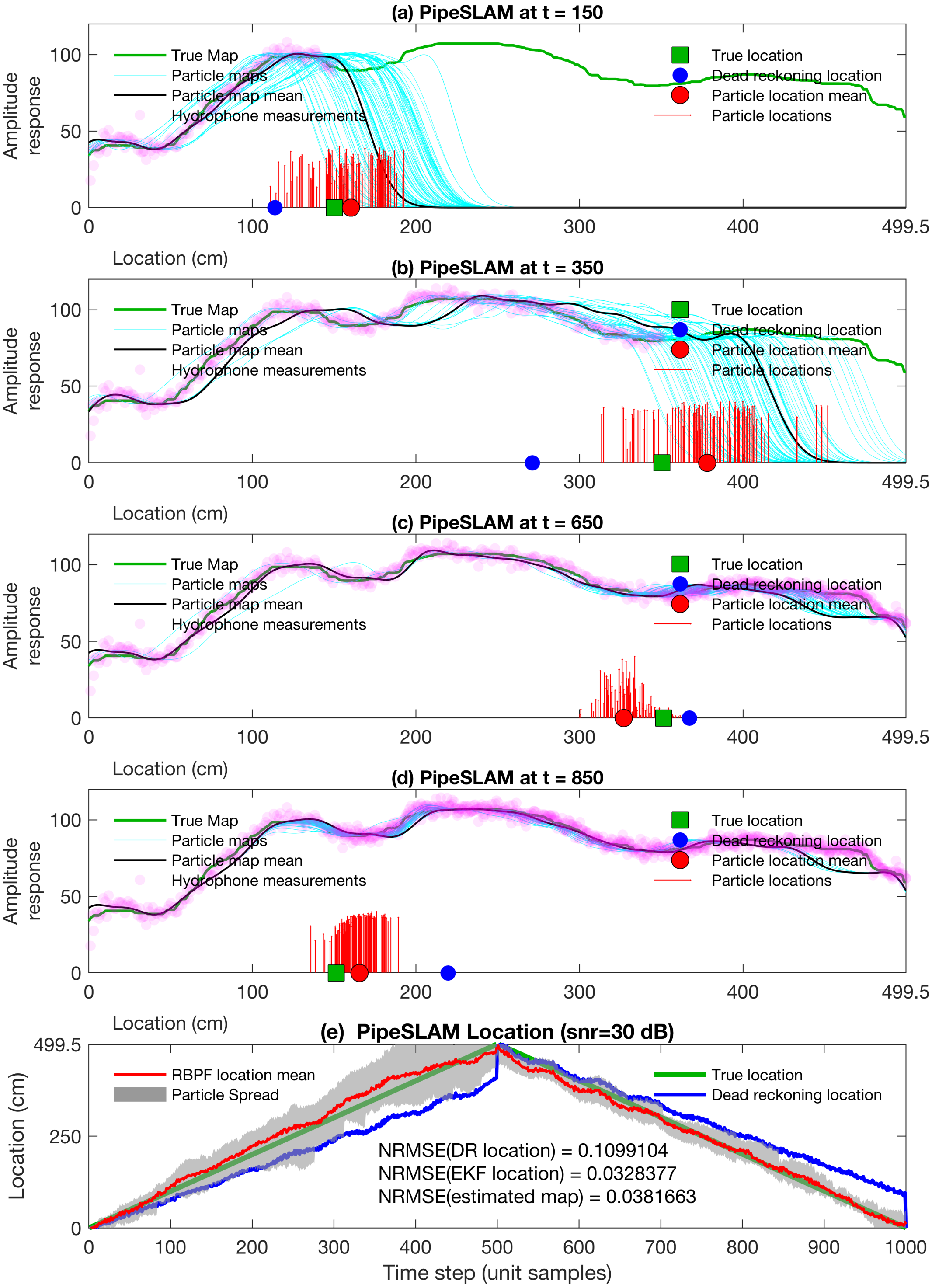


Figure 5.25: PipeSLAM using 5-metre metal pipe data II. Signal to noise ratio (snr) = 30 dB. (a)-(d) PipeSLAM at different time steps building the map while localising. (e) PipeSLAM location estimates much improved accuracy compared with dead reckoning locations.

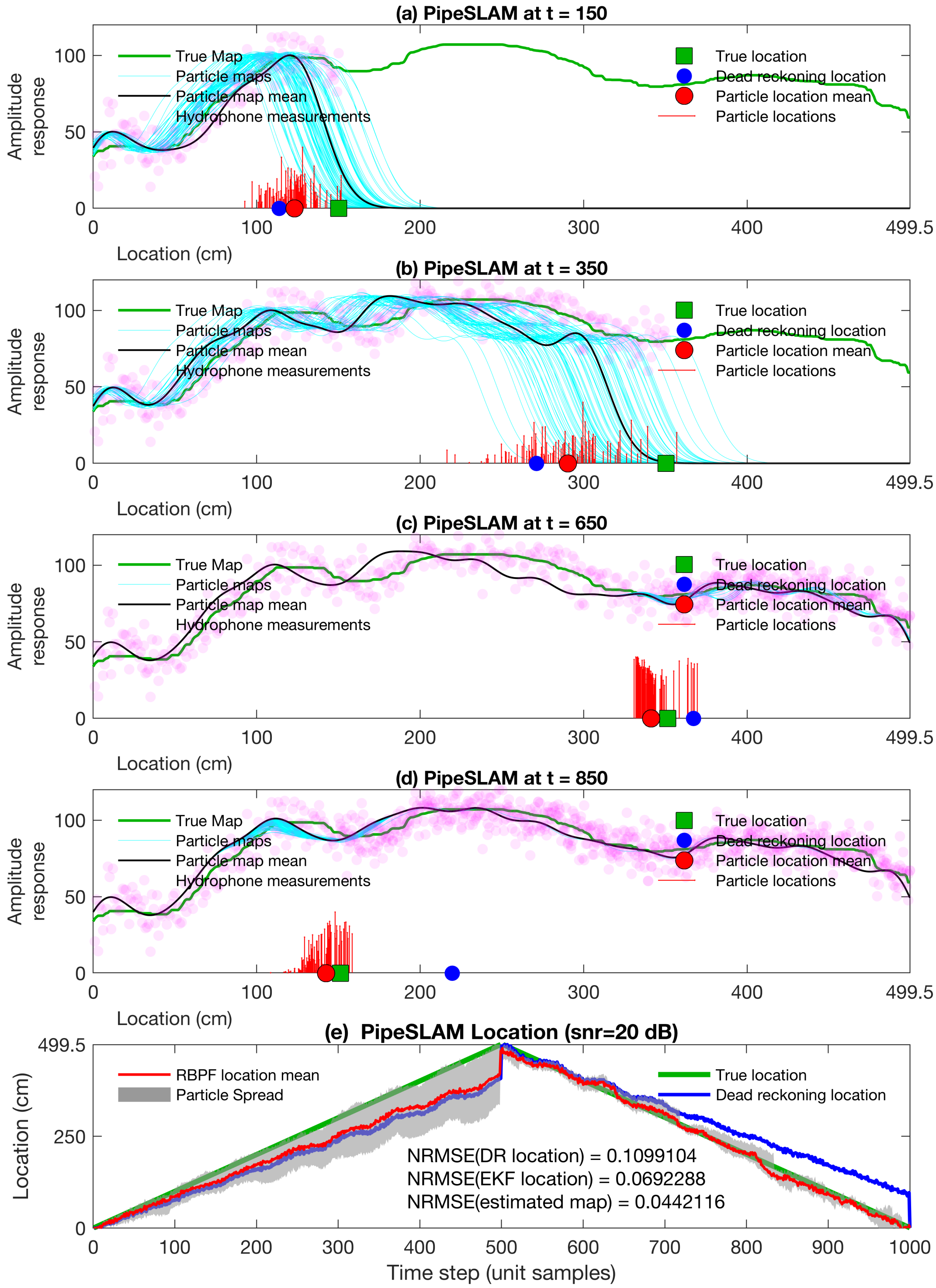


Figure 5.26: PipeSLAM using 5-metre metal pipe data II. Signal to noise ratio (snr) = 20 dB. (a)-(d) PipeSLAM at different time steps building the map while localising. (e) PipeSLAM location estimates much improved accuracy compared with dead reckoning locations.

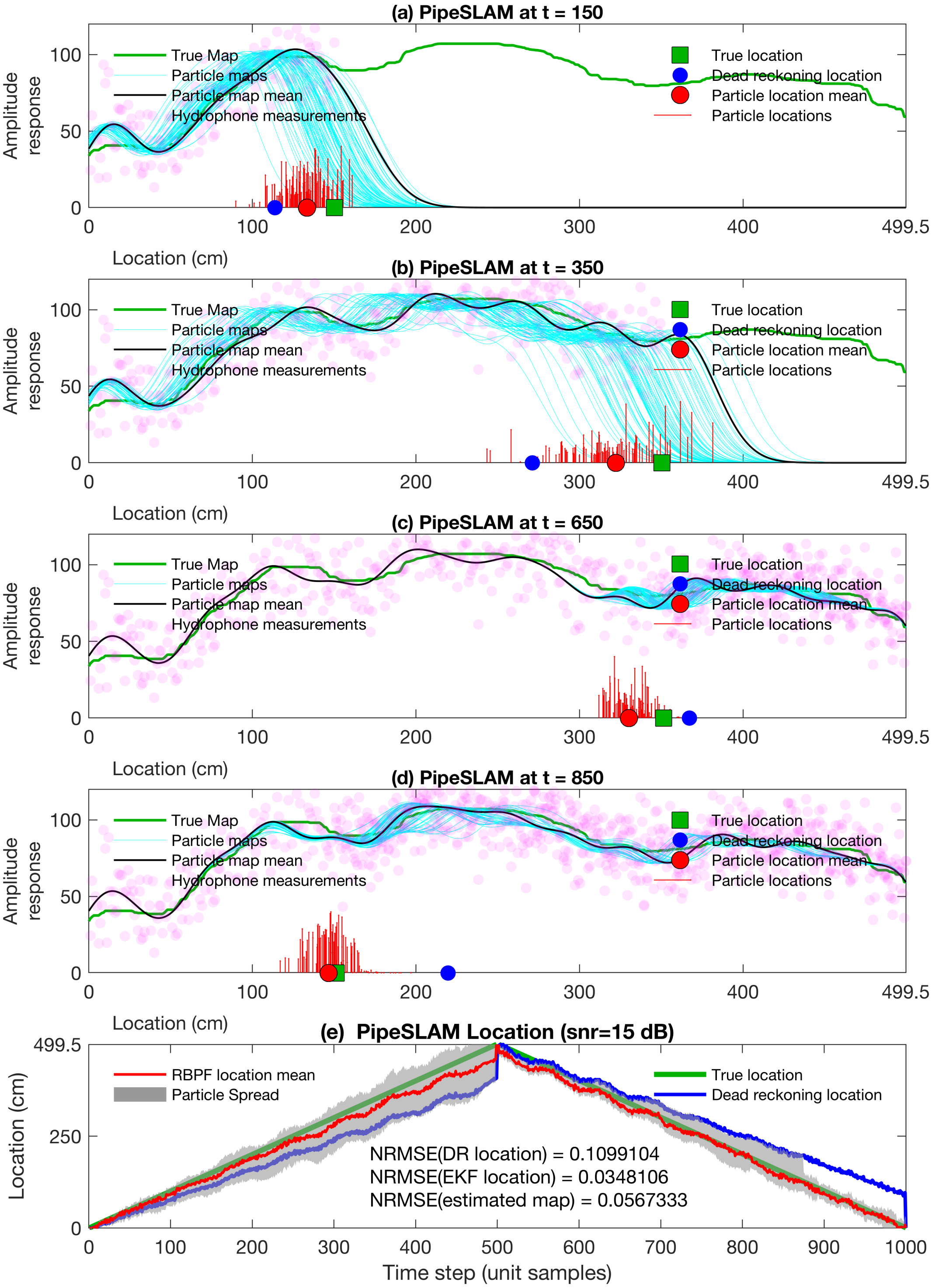


Figure 5.27: PipeSLAM using 5-metre metal pipe data II. Signal to noise ratio (snr) = 15 dB. (a)-(d) PipeSLAM at different time steps building the map while localising. (e) PipeSLAM location estimates much improved accuracy compared with dead reckoning locations.

a same condition. But it is likely that more experiment runs will improve the PipeSLAM performance and its location estimate will be more accurate if increase experiment running times.

Overall, PipeSLAM is a more considerable choice for future in-pipe robot inspections. The first and the most important reason is that it can be upgraded to online application without saving and processing offline data after experiment and the location results are much more accurate than dead reckoning results.

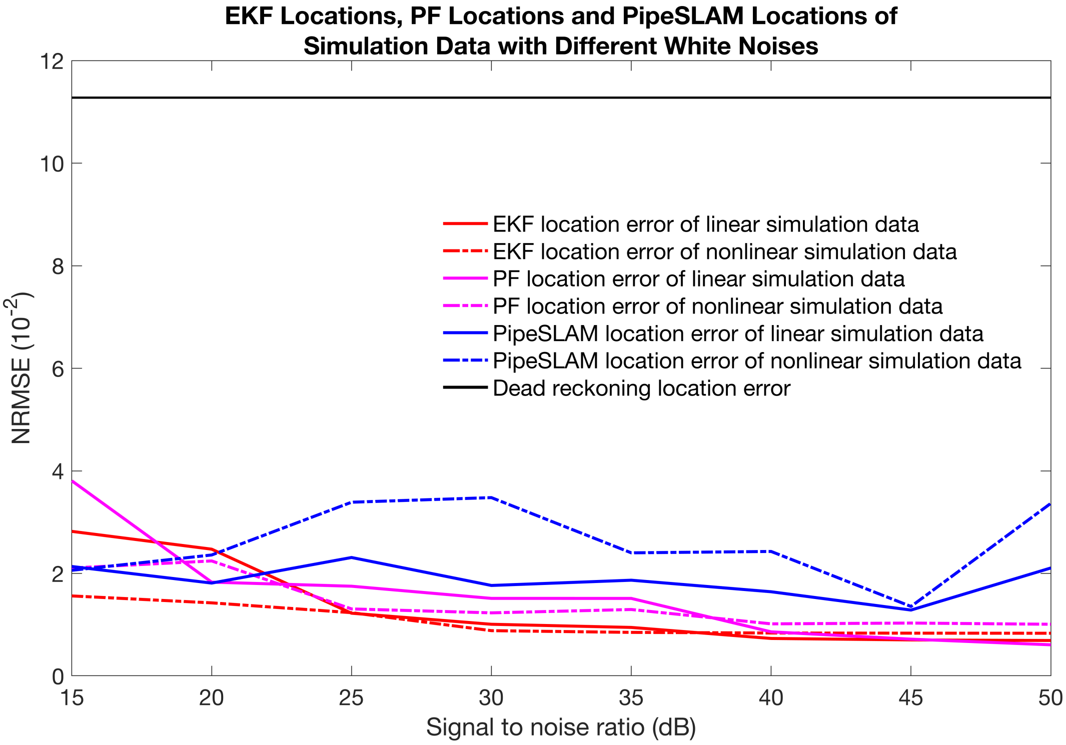


Figure 5.28: Results with additional white noise – a comparison of simulation data using EKF, PF and PipeSLAM algorithms. Results demonstrate EKF estimates, PF estimates and PipeSLAM estimates are outperformed to dead reckoning estimates.

5.4 Summary

This chapter has addressed the SLAM problem for feature-sparse water pipes. In order to track robot location while building the map at the same time, this chapter has introduced a PipeSLAM algorithm based on the Rao-Blackwellised particle filter: the key features of this algorithm are that the map is decomposed using a radial basis function network that is linear-in-the-parameters; these map parameters are estimated using a Kalman filter; robot location is estimated using particles. The feasibility of the algorithm has been tested using a combination of experiment

Table 5.3: EKF, PF and PipeSLAM location estimation results comparison in different noises (Part 1)

Data types	**Algorithms**	**Normalised rooted mean square error (NRMSE)** $\times 10^{-2}$									
		DR	snr=15	snr=20	snr=25	snr=30	snr=35	snr=40	snr=45	snr=50	snr=∞
Linear simulation data	EKF	11.2694	2.8201	2.4731	1.2242	1.0088	0.9472	0.7320	0.7035	0.6924	0.6819
	PF	11.2694	3.8059	1.8299	1.7508	1.5135	1.5128	0.8611	0.7184	0.6082	0.6608
	PipeSLAM	11.2694	2.1354	1.8120	2.3108	1.8880	1.8679	1.6427	1.2860	2.1066	1.1855
Nonlinear simulation data	EKF	11.2694	1.5613	1.4263	1.2348	0.8841	0.8506	0.8384	0.8354	0.8335	1.0264
	PF	11.2694	2.1060	2.2436	1.3081	1.2315	1.2966	1.0169	1.0323	1.0091	0.9968
	PipeSLAM	11.2694	2.0610	2.3598	3.3887	3.4785	2.4016	2.4285	1.3563	3.3671	1.3823
Metal pipe data 1-metre	EKF	11.2694	2.5451	2.4527	1.7259	1.2529	1.1337	1.1139	1.1045	1.1041	1.1104
	PF	11.2694	4.4418	2.3000	1.8672	1.7277	1.6064	1.3283	1.4798	1.3629	1.4718
	PipeSLAM	11.2694	1.5027	1.5597	2.5949	3.3357	3.1268	2.3104	2.8699	2.1783	1.9265
Plastic pipes data	EKF	11.2694	2.6720	2.6651	2.6581	2.6530	2.6536	2.6537	2.6532	2.6528	2.6524
	PF	11.2694	3.0959	2.5871	1.8070	1.8327	1.7588	1.7311	1.9906	1.8994	1.7208
	PipeSLAM	11.2694	2.0218	2.0659	2.4803	3.2434	3.1539	3.6944	1.5161	2.4715	2.1622

Table 5.4: EKF, PF and PipeSLAM location estimation results comparison in different noises (Part 2)

Data types	**Algorithms**	**Location Normalised rooted mean square error (NRMSE)** $\times 10^{-2}$									
		DR	snr=15	snr=20	snr=25	snr=30	snr=35	snr=40	snr=45	snr=50	snr=∞
Metal pipe 5-metre data I	EKF	11.2694	2.8715	2.6387	2.6581	2.6178	2.5622	2.5644	2.5738	2.5861	2.5956
	PF	11.2694	2.8696	2.8297	2.4968	2.8072	2.8163	2.8034	2.6603	2.7876	2.5609
	PipeSLAM	11.2694	2.4220	3.5799	1.7396	2.8926	2.1635	2.5658	3.0831	2.0318	2.3774

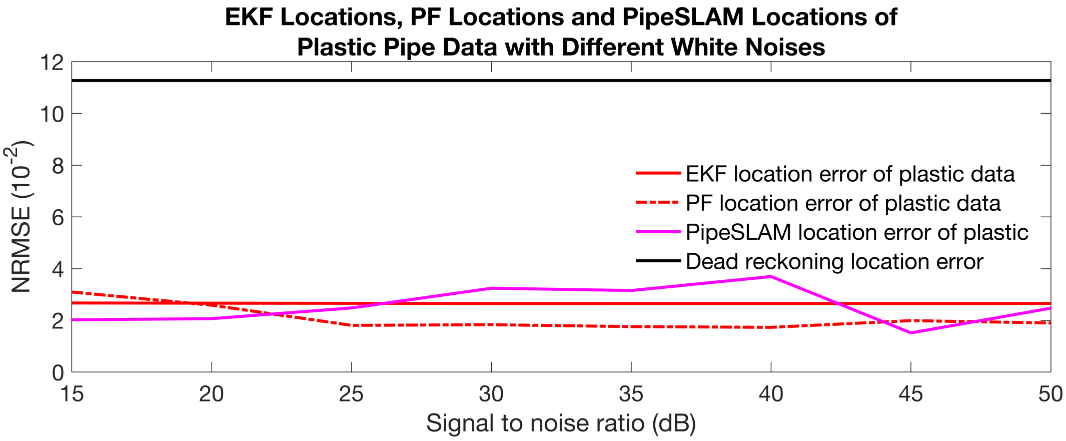


Figure 5.29: Results with additional white noise – a comparison of plastic data using EKF, PF and PipeSLAM algorithms. Results demonstrate EKF estimates, PF estimates and PipeSLAM estimates are outperformed to dead reckoning estimates.

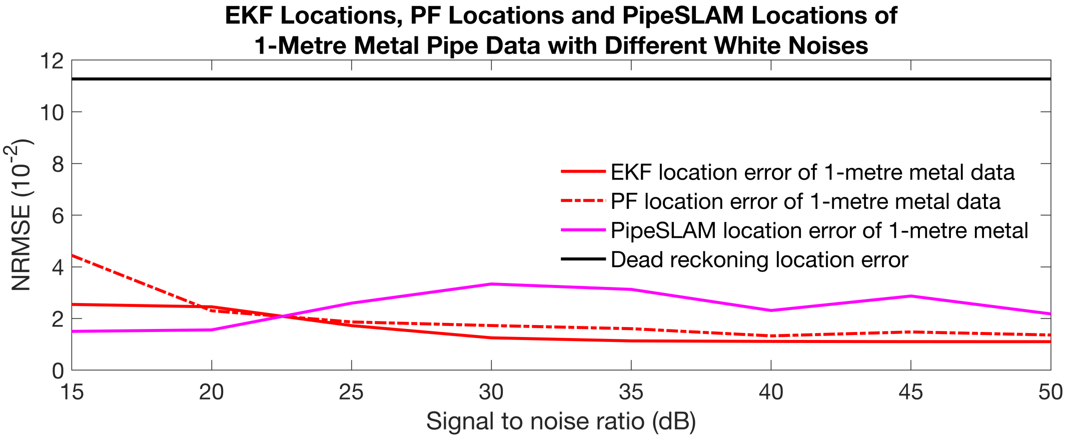


Figure 5.30: Results with additional white noise – a comparison of 1-metre metal pipe data using EKF, PF and PipeSLAM algorithms. Results demonstrate EKF estimates, PF estimates and PipeSLAM estimates are outperformed to dead reckoning estimates.

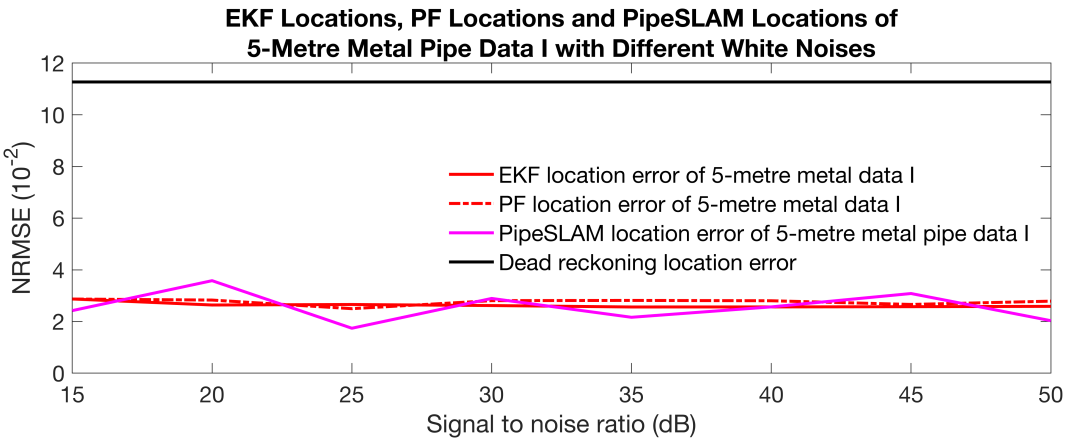


Figure 5.31: Results with blockage noise – a comparison of 5-metre metal pipe data I using EKF, PF and PipeSLAM algorithms. Results demonstrate EKF estimates, PF estimates and PipeSLAM estimates are outperformed to dead reckoning estimates.

and simulation, where the maps are generated in simulation as well as experiments and then the PipeSLAM algorithm has been evaluated on the those maps in simulation. A benchmark against robot localisation using just dead reckoning for all data sets both from simulation and experiments have demonstrated the success of the PipeSLAM method.

Chapter 6

Sensor Fusion For Estimating Pipe Location in the World Coordinate Frame

6.1 Introduction

The work in previous chapters has developed novel solutions to estimating the distance of the robot travelled through water pipes. This information allows the robot to estimate its own location in the local coordinate frame of the pipe (because a small diameter pipe is essentially a one dimensional environment, hence distance travelled is the only measure of location required). However, the distance travelled along a pipe is not sufficient to transform the robot location from the local coordinate frame into the world coordinate frame. The heading of the robot is also required for this transformation. The transformation of the robot into the world coordinate frame is of crucial importance because this also determines the location of the pipe itself, which is needed to effect repairs from above ground.

This chapter will add another sensor to the system that will produce heading information to the robot localisation algorithm, and that is an inertial measurement unit (IMU). Due to the fast development of current IMU technologies in micro-electronic mechanical systems (MEMS) devices, many mobile robot systems have employed IMUs as one of their important sensors (see Section 2.5). The IMU used in this chapter is an *Xsens MTi-3-8A7G6-DK* (see Fig.6.2), which is a commercial product that includes calibration and processing algorithms that fuse gyroscope and accelerometer sensors using an EKF to produce attitude estimates in terms of yaw, pitch and roll.

Therefore, the robot distance travelled can be estimated by PipeSLAM, and the

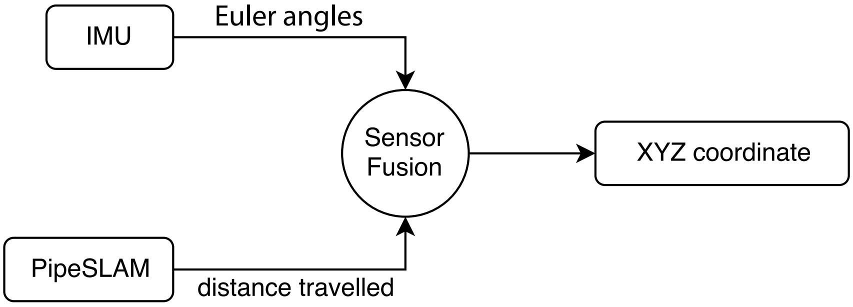


Figure 6.1: Sensor fusion illustration diagram

heading can be estimated by an IMU. This chapter will introduce an approach to fuse IMU and PipeSLAM to estimate XYZ locations in 3D space. The key novel contribution is formulating the sensor fusion algorithm in such a way that the uncertainty of the two distinct estimation algorithms, PipeSLAM for distance travelled and IMU-EKF for heading, is propagated into the world coordinate frame. This uncertainty estimate is generated via Monte Carlo simulations of a stochastic local-world coordinate transformation, which is both simple and effective.

6.2 Methods

From the previous chapters, one-dimensional locations are assumed. This is due to the fact that experiments were conducted in pipes with small diameters. Therefore, the sensor platform is assumed to move either forward or backward with no cross-sectional movements. However, to carry out repairs on pipes from above ground, the location of the pipe must be estimated in the world coordinate frame. Thus, directional information needs to be acquired and an IMU can be used for this purpose. Combined with the distance travelled produced from previous approaches, such as PipeSLAM, XYZ coordinates can be computed after the sensor fusion (see Fig.6.1).

An IMU device can produce rotational Euler angles along x, y and z axes in the world frame with embedded its estimators, such as extended Kalman filter [8]. Thus, in each time stamp, the pose of the robot can be easily calculated by using simple trigonometric functions.

6.2.1 Heading Estimate via Inertial Measurement

IMU technology has been developing rapidly but the principle behind the small chip is rather similar. Mostly used IMU data fusion approaches might be traced back to 1995, according to the work from [8]. The heading of a mobile robot

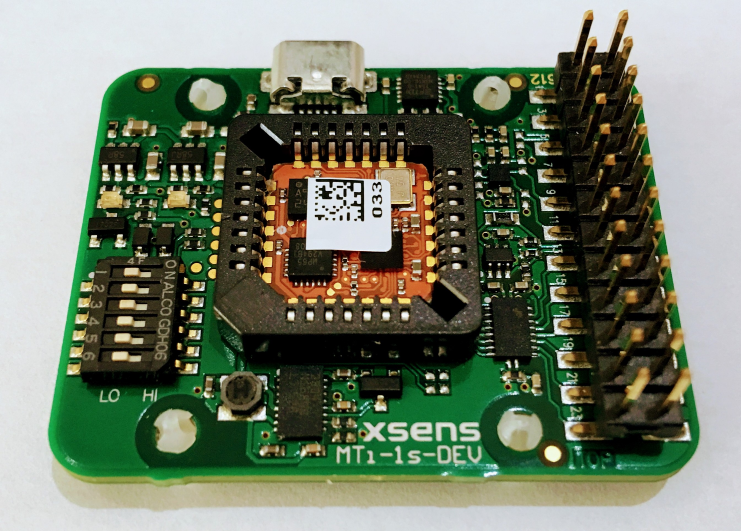


Figure 6.2: *xsens MTi-3-8A7G6-DK* IMU used in the laboratory experiments

can be estimated by an IMU. The IMU uses a gyroscope to measure rotational velocity and an accelerometer to measure linear acceleration. The gyroscope and accelerometer data are then fused using an EKF. The observation equations are

$$\begin{aligned}
z_{G_x}(k) &= \dot{\psi}_x(k) + \epsilon_{\dot{\psi}_x}(k) + v_1(k) \\
z_{G_y}(k) &= \dot{\psi}_y(k) + \epsilon_{\dot{\psi}_y}(k) + v_2(k) \\
z_{G_z}(k) &= \dot{\psi}_z(k) + \epsilon_{\dot{\psi}_z}(k) + v_3(k) \\
z_{A_x}(k) &= \cos\psi_y(k)\cos\psi_z(k)a_x(k) + \cos\psi_y(k)\sin\psi_z(k)a_y(k) \\
&\quad - \sin\psi_y(k)g(k) + \epsilon_{a_x}(k) + v_4(k) \\
z_{A_y}(k) &= [\sin\psi_x(k)\sin\psi_y(k)\cos\psi_z(k) - \cos\psi_x(k)\sin\psi_z(k)]a_x(k) \\
&\quad + [\sin\psi_x(k)\sin\psi_y(k)\sin\psi_z(k) + \cos\psi_x(k)\cos\psi_z(k)]a_y(k) \\
&\quad + \sin\psi_x(k)\cos\psi_y(k)g(k) + \epsilon_{a_y}(k) + v_5(k) \\
z_{A_y}(k) &= [\cos\psi_x(k)\sin\psi_y(k)\cos\psi_z(k) + \sin\psi_x(k)\sin\psi_z(k)]a_x(k) \\
&\quad + [\cos\psi_x(k)\sin\psi_y(k)\sin\psi_z(k) - \sin\psi_x(k)\cos\psi_z(k)]a_y(k) \\
&\quad + \cos\psi_x(k)\cos\psi_y(k)g(k) + \epsilon_g(k) + v_6(k)
\end{aligned} \tag{6.1}$$

The observations from gyroscopes and acceleration measurement unit on the IMU chip are $z_{G_x}(k)$, $z_{G_y}(k)$, $z_{G_z}(k)$, $z_{A_x}(k)$, $z_{A_y}(k)$ and $z_{A_z}(k)$. $\dot{\psi}_x(k)$, $\dot{\psi}_y(k)$, $\dot{\psi}_z(k)$ are the Euler angular rotation rates along x, y and z local axes, and $a_x(k)$, $a_y(k)$ and $a_z(k)$ are the accelerations along x, y and z local axes. Additionally, $\epsilon_{\dot{\psi}_x}(k)$, $\epsilon_{\dot{\psi}_y}(k)$, $\epsilon_{\dot{\psi}_z}(k)$, $\epsilon_{a_x}(k)$, $\epsilon_{a_y}(k)$ and $\epsilon_{a_z}(k)$ are the drift errors added to the measurements. Also, Gaussian noises $v_1(k)$, $v_2(k)$, $v_3(k)$, $v_4(k)$, $v_5(k)$, $v_6(k)$, are assumed here to be added into the equations respectively, in order to use extended Kalman filter. More detailed explanations can be found in [8].

The state update equations are

$$\begin{bmatrix} \mathbf{x}_{G_x}(k+1) \\ \mathbf{x}_{G_y}(k+1) \\ \mathbf{x}_{G_z}(k+1) \\ \mathbf{x}_{A_x}(k+1) \\ \mathbf{x}_{A_y}(k+1) \\ \mathbf{x}_{A_z}(k+1) \end{bmatrix} = \begin{bmatrix} \mathbf{F}_{G_x} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{F}_{G_y} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{F}_{G_z} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{F}_{A_x} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{F}_{A_y} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{F}_{A_z} \end{bmatrix} \begin{bmatrix} \mathbf{x}_{G_x}(k) \\ \mathbf{x}_{G_y}(k) \\ \mathbf{x}_{G_z}(k) \\ \mathbf{x}_{A_x}(k) \\ \mathbf{x}_{A_y}(k) \\ \mathbf{x}_{A_z}(k) \end{bmatrix} + \begin{bmatrix} \mathbf{u}_{G_x} \\ \mathbf{u}_{G_y} \\ \mathbf{u}_{G_z} \\ \mathbf{u}_{A_x} \\ \mathbf{u}_{A_y} \\ \mathbf{u}_{A_z} \end{bmatrix} + \mathbf{w}(k) \tag{6.2}$$

with

$$\mathbf{F}_{G_x} \triangleq \begin{bmatrix} 1 & T_s & \frac{1}{2}T_s^2 & \frac{1}{6}T_s^3 & 0 & 0 \\ 0 & 1 & T_s & \frac{1}{2}T_s^2 & 0 & 0 \\ 0 & 0 & 1 & T_s & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & T_s \\ 0 & 0 & 0 & 0 & 0 & \frac{T_{\dot{\psi}z}}{T_{\dot{\psi}z}+T_s} \end{bmatrix} \tag{6.3}$$

$$\mathbf{F}_{A_x} \triangleq \begin{bmatrix} 1 & T_s & \frac{1}{2}T_s^2 & 0 \\ 0 & 1 & T_s & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & \frac{T_{ax}}{T_{ax}+T_s} \end{bmatrix} \tag{6.4}$$

$$\mathbf{x}_{G_x}(k) \triangleq \begin{bmatrix} \psi_z(k) \\ \dot{\psi}_z(k) \\ \ddot{\psi}_z(k) \\ \dddot{\psi}_z(k) \\ \epsilon_{\psi_z}(k) \\ \epsilon_{\dot{\psi}_z}(k) \end{bmatrix} \tag{6.5}$$

$$\mathbf{x}_{A_x}(k) \triangleq \begin{bmatrix} x(k) \\ v_x(k) \\ a_x(k) \\ \epsilon_{a_x}(k) \end{bmatrix} \tag{6.6}$$

$$\mathbf{u}_{G_x}(k) \triangleq \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \frac{T_s(C_{1\dot{\psi}_z}+C_{2\dot{\psi}_z})}{T_{\dot{\psi}_z}+T_s} \end{bmatrix} \tag{6.7}$$

and

$$\mathbf{u}_{A_x}(k) \triangleq \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{T_s(C_{1a_x}+C_{2a_x})}{T_{a_x}+T_s} \end{bmatrix} \tag{6.8}$$

where C_1 and C_2 are error constants such that $\epsilon(0) = C_2$ and $\dot{\epsilon}(0) = \frac{C_1}{T}$, and T_s is the sampling interval. Others including $T_{\dot{\psi}_z}$, C_{1a_x} and C_{2a_x} are parameters that are to be tuned, according to Barshan and Durrant-Whyte [8]. The other remaining matrices $\mathbf{F}_{G_y}$, $\mathbf{F}_{G_z}$, $\mathbf{F}_{A_y}$, $\mathbf{F}_{A_y}$, and remaining state vectors $\mathbf{x}_{G_y}$, $\mathbf{x}_{G_z}$, $\mathbf{x}_{A_y}$, $\mathbf{x}_{A_y}$ in (6.2) have the similar formats according to the equations above with individual corresponding error parameters substituted in. Therefore, the combined state contains 30 states and the state update equation ((6.2)) can be expressed as

$$\mathbf{x}_k = \mathbf{F}\mathbf{x}_{k-1} + \mathbf{u} + \mathbf{w}_{k-1} \tag{6.9}$$

where

$$\mathbf{F} = \begin{bmatrix} \mathbf{F}_{G_x} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{F}_{G_y} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{F}_{G_z} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{F}_{A_x} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{F}_{A_y} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{F}_{A_z} \end{bmatrix} \tag{6.10}$$

The state equation (6.9) is linear while the observation equation (6.1) is nonlinear.

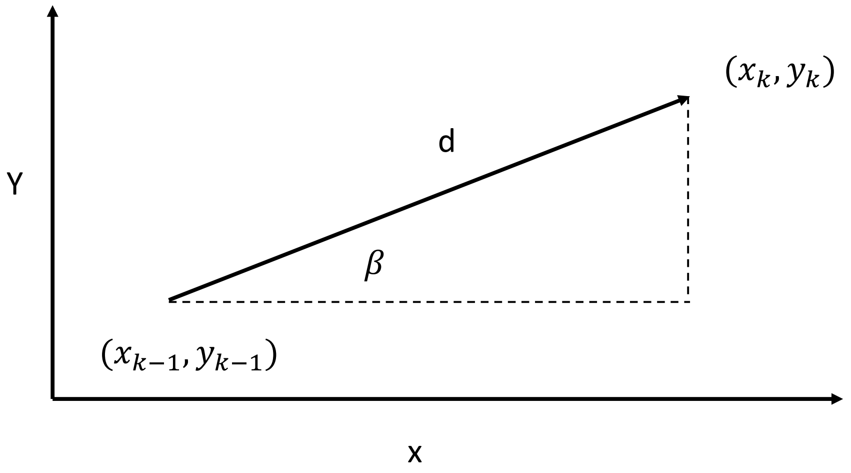


Figure 6.3: An example showing one step increment from $t = k - 1$ to $t = k$ in a 2D case

Therefore, a standard EKF can be applied to the IMU model.

6.2.2 Sensor Fusion

IMU provides the heading estimate, whilst PipeSLAM or DBA method provides distance travelled. The aim of this section is to fuse these methods, into x-y-z location of the pipe/robot in the world coordinate frame, with uncertainty.

In a simple 2D case (see Fig.6.3), assuming at time stamp k, the robot is rotating by β, and moving forward by d. Equations are

$$\begin{aligned} x_k &= x_{k-1} + d\cos\beta \\ y_k &= y_{k-1} + d\sin\beta \end{aligned} \tag{6.11}$$

When the robot rotates by β, its local coordinate frame will also rotate by β and any vector coordinates in the previous local frame needs to be updated after the rotation. This need to require a rotation matrix. Assume a vector $\mathbf{v} = (x, y)$ is in its local frame. When the local frame rotates anti-clockwise by β, the vector $\mathbf{v}$ will all rotate to be $\mathbf{v}'$ and its coordinates in the world frame will be updated following the equations below

$$\begin{aligned} x' &= x\cos\beta - y\sin\beta \\ y' &= x\sin\beta + y\cos\beta \end{aligned} \tag{6.12}$$

or in a matrix form

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \mathbf{R}_{2d}(\beta) \begin{bmatrix} x \\ y \end{bmatrix} \tag{6.13}$$

$$\mathbf{R}_{2d}(\beta) = \begin{bmatrix} \cos\beta & -\sin\beta \\ \sin\beta & \cos\beta \end{bmatrix} \tag{6.14}$$

The square matrix $\mathbf{R}_{2d}$ in (6.14) is called the direction cosine matrix or DCM which transforms a coordinate frame to another coordinate frame. However, in $3D$ cases, rotations could happen in all $x - y - z$ axes and the rotation matrix is relatively more complicated. The Euler angles come from the IMU are $\psi_x \overset{\Delta}{=} \mathit{roll}$, $\psi_y \overset{\Delta}{=} \mathit{pitch}$ and $\psi_z \overset{\Delta}{=} \mathit{yaw}$. Thus the rotation matrix or DCM in 3D space can be expressed as:

$$\mathbf{R}_z(\psi_z) = \begin{bmatrix} \cos\psi_z & -\sin\psi_z & 0 \\ \sin\psi_z & \cos\psi_z & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{6.15}$$

$$\mathbf{R}_y(\psi_y) = \begin{bmatrix} \cos\psi_y & 0 & \sin\psi_y \\ 0 & 1 & 0 \\ -\sin\psi_y & 0 & \cos\psi_y \end{bmatrix} \tag{6.16}$$

$$\mathbf{R}_x(\psi_x) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\psi_x & -\sin\psi_x \\ 0 & \sin\psi_x & \cos\psi_x \end{bmatrix} \tag{6.17}$$

To construct the DCM in 3D, three rotation matrices $\mathbf{R}_x(\psi_z)$, $\mathbf{R}_y(\psi_y)$, $\mathbf{R}_z(\psi_x)$ need to be multiplied together in certain order. It is worth mentioning that different multiplication orders will result in different outcomes. The most commonly used are $\mathbf{R}_{zyx} = \mathbf{R}_z\mathbf{R}_y\mathbf{R}_x$ showing (6.18).

$$\mathbf{R}_{zyx} = \mathbf{R}_z\mathbf{R}_y\mathbf{R}_x =$$
$$\begin{bmatrix} \cos\psi_y\cos\psi_x & \cos\psi_x\sin\psi_z\sin\psi_y - \cos\psi_z\sin\psi_x & \cos\psi_z\cos\psi_x\sin\psi_y + \sin\psi_z\sin\psi_x \\ \cos\psi_y\sin\psi_x & \sin\psi_z\sin\psi_y\sin\psi_x + \cos\psi_z\cos\psi_x & \cos\psi_z\sin\psi_y\sin\psi_x - \cos\psi_z\sin\psi_x \\ -\sin\psi_y & \cos\psi_y\sin\psi_z & \cos\psi_z\cos\psi_y \end{bmatrix} \tag{6.18}$$

$$\mathbf{R}_{xyz} = \mathbf{R}_x\mathbf{R}_y\mathbf{R}_z =$$

$$\begin{bmatrix} \cos\psi_y\cos\psi_x & -\cos\psi_y\sin\psi_x & \sin\psi_y \\ \cos\psi_z\sin\psi_x + \sin\psi_z\sin\psi_y\cos\psi_x & \cos\psi_z\cos\psi_x - \sin\psi_z\sin\psi_y\sin\psi_x & -\sin\psi_z\cos\psi_y \\ \sin\psi_z\sin\psi_x - \cos\psi_z\sin\psi_y\cos\psi_x & \sin\psi_z\cos\psi_x + \cos\psi_z\sin\psi_y\sin\psi_x & \cos\psi_z\cos\psi_y \end{bmatrix} \tag{6.19}$$

The Eq.(6.18) interprets that the space is firstly rotated about z-axis, then y-axis, then x-axis. This is opposite to Eq.(6.19).

Location Estimation using Monte Carlo Method

Now to set DCM matrix notation to be $\mathbf{R}_{dcm}$, which can be $\mathbf{R}_{zyx}$ in (6.18) or $\mathbf{R}_{xyz}$ in (6.19). Then express $\mathbf{R}_{dcm}$ in column vectors such that

$$\mathbf{R}_{dcm} \triangleq \begin{bmatrix} \mathbf{r}^x_{dcm} & \mathbf{r}^y_{dcm} & \mathbf{r}^z_{dcm} \end{bmatrix} \tag{6.20}$$

where $\mathbf{r}^x_{dcm}$ is the first column indicating the direction vector of its $x-$axis in world coordinate frame, $\mathbf{r}^y_{dcm}$ is the second column indicating the direction vector of its $y-$axis in world coordinate frame and $\mathbf{r}^z_{dcm}$ is the third column in $\mathbf{R}_{dcm}$ indicating the direction vector of its $z-$axis in world coordinate frame. Assuming the robot heading is along $x-$axis in its local coordinate frame, which means the $x-$axis in its local coordinate frame is pointing to the robot movement direction. Therefore, by using (6.11), the location coordinates update equation in 3D form can be expressed as

$$\begin{bmatrix} x_k \\ y_k \\ z_k \end{bmatrix} = \begin{bmatrix} x_{k-1} \\ y_{k-1} \\ z_{k-1} \end{bmatrix} + d_k \cdot \mathbf{r}^x_{dcm}(\psi_{x,k}, \psi_{y,k}, \psi_{z,k}) \tag{6.21}$$

To simplify notation, define

$$\mathbf{l}_k \triangleq \begin{bmatrix} x_k \\ y_k \\ z_k \end{bmatrix} \tag{6.22}$$

$$\mathbf{f}(\psi_{x,k}, \psi_{y,k}, \psi_{z,k}) \triangleq \mathbf{r}^x_{dcm}(\psi_{x,k}, \psi_{y,k}, \psi_{z,k}) \tag{6.23}$$

and (6.21) becomes

$$\mathbf{l}_k = \mathbf{l}_{k-1} + d_k \cdot \mathbf{f}(\psi_{x,k}, \psi_{y,k}, \psi_{z,k}) \tag{6.24}$$

In order to calculate variance of the robot location in 3D world coordinates, additional white noise should be added into IMU angular outputs. By using Monte

Carlo simulation method with n_s particles, the particle location set at time k are $\mathbf{l}_k^p$, where $p = 1, \ldots, n_s$. The particle location update in state space form is

$$\mathbf{l}_k^p = \mathbf{l}_{k-1}^p + d_k \cdot \mathbf{f}(\psi_{x,k} + w_{x,k}^{\psi}, \psi_{y,k} + w_{y,k}^{\psi}, \psi_{z,k} + w_{z,k}^{\psi}) \tag{6.25}$$

where $w_{x,k}^{\psi}$, $w_{y,k}^{\psi}$ and $w_{z,k}^{\psi}$ are zero mean Gaussian noise at time k. Then the covariance of the location estimate can be numerically calculated as

$$\Sigma_{\mathbf{l}_k} = \frac{1}{n_s} \sum_{p=1}^{n_s} (\mathbf{l}_k^p - \mathbf{E}[\bar{\mathbf{l}}_k])^2 \tag{6.26}$$

where $\bar{\mathbf{l}}_k = \mathbf{E}[\mathbf{l}_k] = \sum_{p=1}^{n_s} \mathbf{l}_k^p$ is the averaged location at time k in 3D world frame. Therefore, the location estimate at time k is expressed as Gaussian distributed variable that

$$\mathbf{l}_k \sim \mathcal{N}(\bar{\mathbf{l}}_k, \frac{1}{n_s} \sum_{p=1}^{n_s} (\mathbf{l}_k^p - \bar{\mathbf{l}}_k)^2) \tag{6.27}$$

Hence, since the mean location estimation $\bar{\mathbf{l}}_k$ and variance $\Sigma_{\mathbf{l}_k}$ can be calculated, the location estimation results can be plotted (e.g. Fig.6.6).

Location Estimation with Prior Knowledge of the Pipe

Since the pipe used in the experiment is straight, it can be used as a prior knowledge to the estimation. This prior knowledge can be expressed as Gaussian variable that meets the condition $\mathbf{L}_{prior} \sim \mathcal{N}(\bar{\mathbf{L}}_{prior}, \Sigma_{prior})$. Denote previous location estimation to be $\mathbf{L}_{est} \triangleq \mathbf{l}_k$, for $k = 1, \ldots, n_s$, and the combined posterior estimate mean $\bar{\mathbf{L}}_{post}$ and covariance $\Sigma_{\mathbf{L}_{post}}$ can be calculated by using the well-known formulas:

$$\bar{\mathbf{L}}_{post} = \Sigma_{\mathbf{L}_{post}} (\Sigma_{\mathbf{L}_{est}}^{-1} \bar{\mathbf{L}}_{est} + \Sigma_{\mathbf{L}_{prior}}^{-1} \bar{\mathbf{L}}_{prior}) \tag{6.28}$$

$$\Sigma_{\mathbf{L}_{post}} = (\Sigma_{\mathbf{L}_{est}}^{-1} + \Sigma_{\mathbf{L}_{prior}}^{-1})^{-1} \tag{6.29}$$

6.3 Algorithm Evaluation

The experiments are conducted in a 5-metre metal pipe, and two data sets are collected. The robot distance travelled has been estimated in Chapter 4 and Chapter 5. Since the two methods have provided similar distance travelled results, and due to the page limitation, the distance travelled estimated by PipeSLAM are selected to demonstrate the effectiveness of this sensor fusion algorithm. In the Monte

Carlo simulation, the Gaussian noise $w^{\psi}_{x,k}$, $w^{\psi}_{y,k}$ and $w^{\psi}_{z,k}$ in (6.25) is set to be zero means with covariances of 0.3 (radians per unit sample) that added into IMU Euler angle outputs.

6.4 Results and Discussion

The sensor fusion results are shown in Fig.6.6 $\sim$ Fig.6.9 with four panels in each figure. Fig.6.6 shows the result in xy-plane (horizontal plane) and Fig.6.7 shows the result in xz-plane (vertical plane).

Panel (a) shows the direct location estimation of true distance estimates with IMU data. Although distance travelled does not contain any types noise and drift, the location coordinate estimates are still drifted and this is due to the disturbances in IMU measurements. Panel (b) shows the direct location estimation of PipeSLAM distance estimates with IMU data. This estimation is still bad for the same reason of the IMU measurement noise. However, by using Monte Carlo simulation method, and combining forward pass location with backward location, the mean estimated location are more accurate and close to the true pipe location (green line), shown in panel (c). The red lines are the numerically calculated variance of particle locations, which can be regarded as uncertainty estimates. By adding a prior knowledge of the pipe location (panel (d)), the location estimates are much closer to the true pipe location and the uncertainty (red lines) is largely reduced.

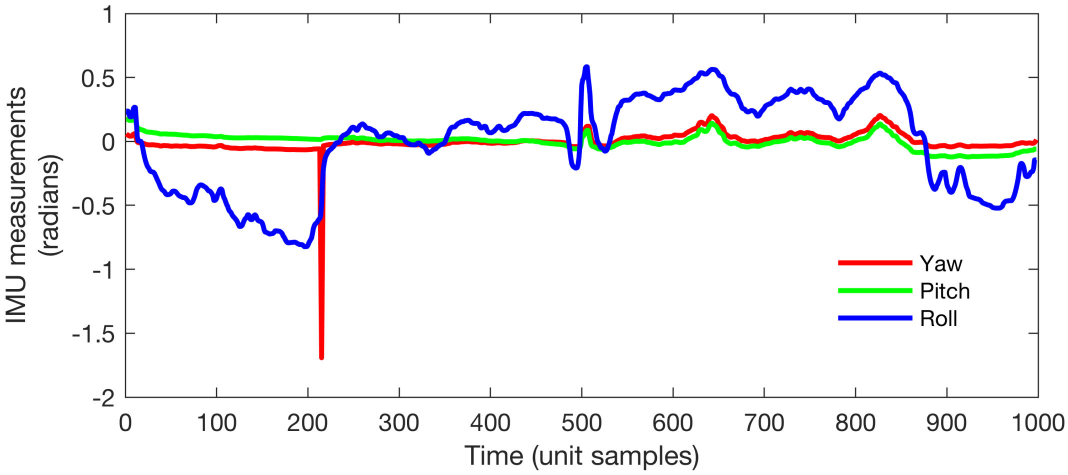


Figure 6.4: Yaw, pitch and roll outputs from IMU Data I

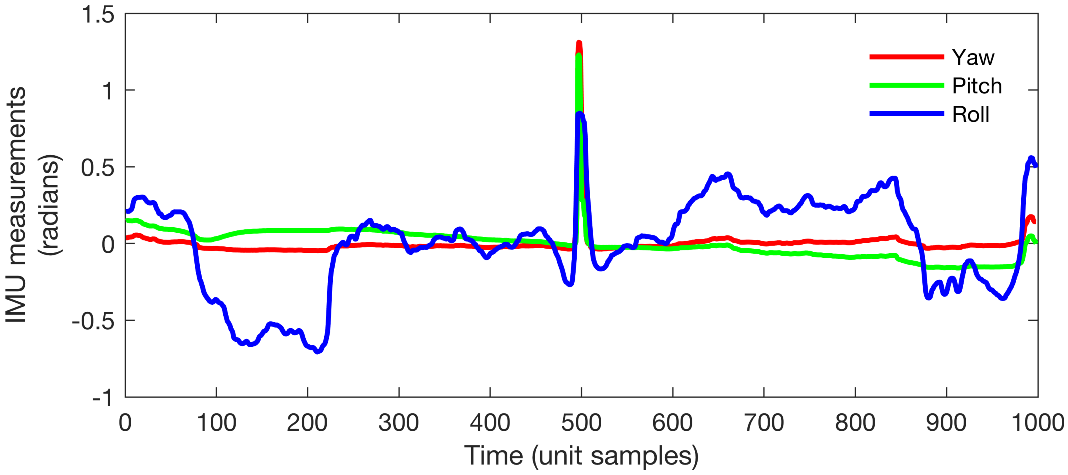


Figure 6.5: Yaw, pitch and roll outputs from IMU Data II

6.5 Summary

In summary, this chapter has demonstrated a sensor fusion technique by adding an IMU to provide heading information. The experiments are tested in the 5-metre metal water pipe with an IMU chip operating on sensor platform. The experiment results have demonstrated that with multi-sensor fusion of one motor encoder, a pair of hydrophones and an IMU can produce a good estimation of pipe location in the 3D world coordinate frame.

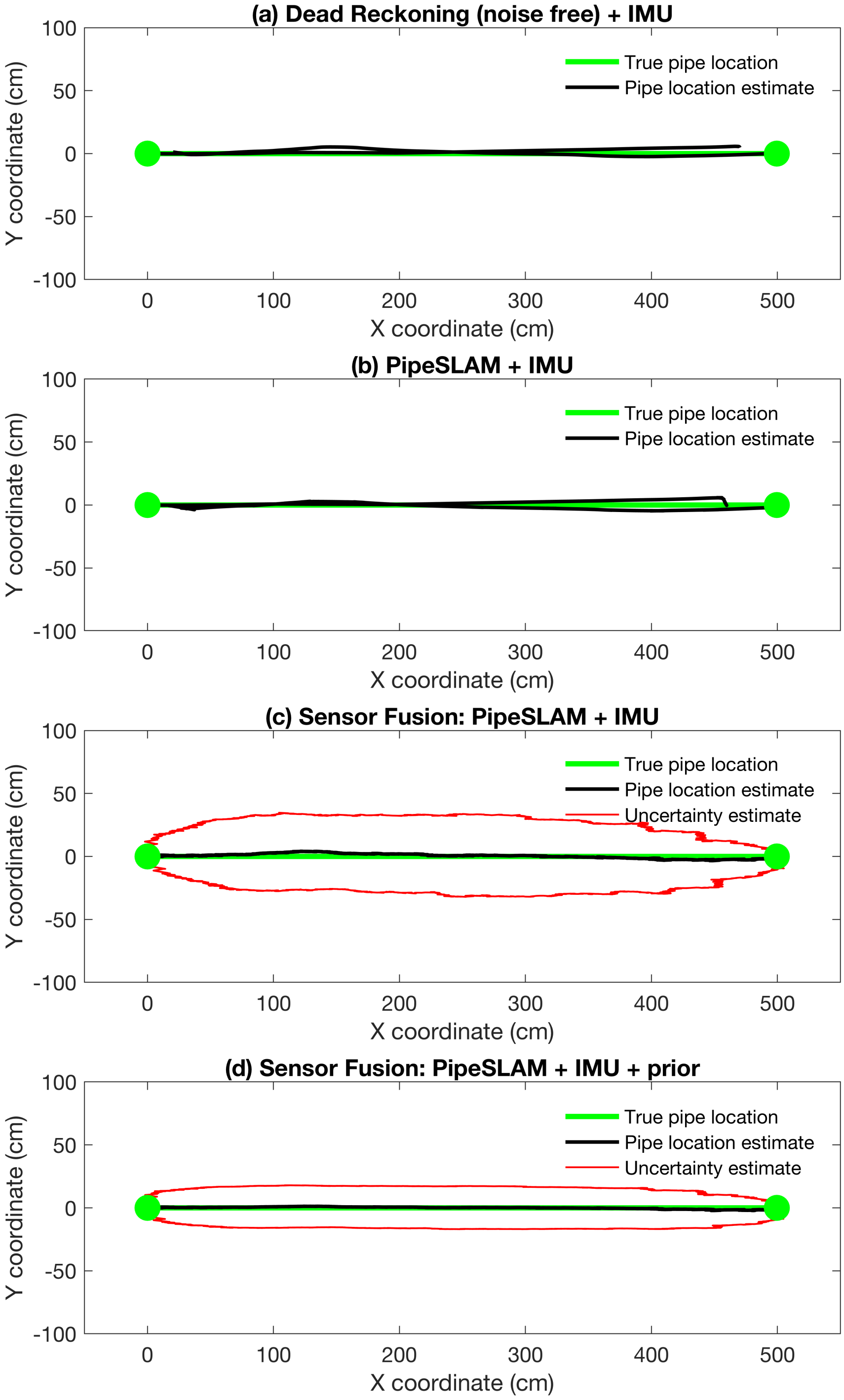


Figure 6.6: Pipe mapping using sensor fusion technique using data set I shown in xy-plane. (a) The estimation using noise free dead reckoning distances and IMU data. (b) Estimation using PipeSLAM distance and IMU data. (c) Estimation using sensor fusion of PipeSLAM distance and IMU data, shown with sensor uncertainties. (d) Along with PipeSLAM distance and IMU data, a prior knowledge (assuming the pipe is straight) is added to the estimation.

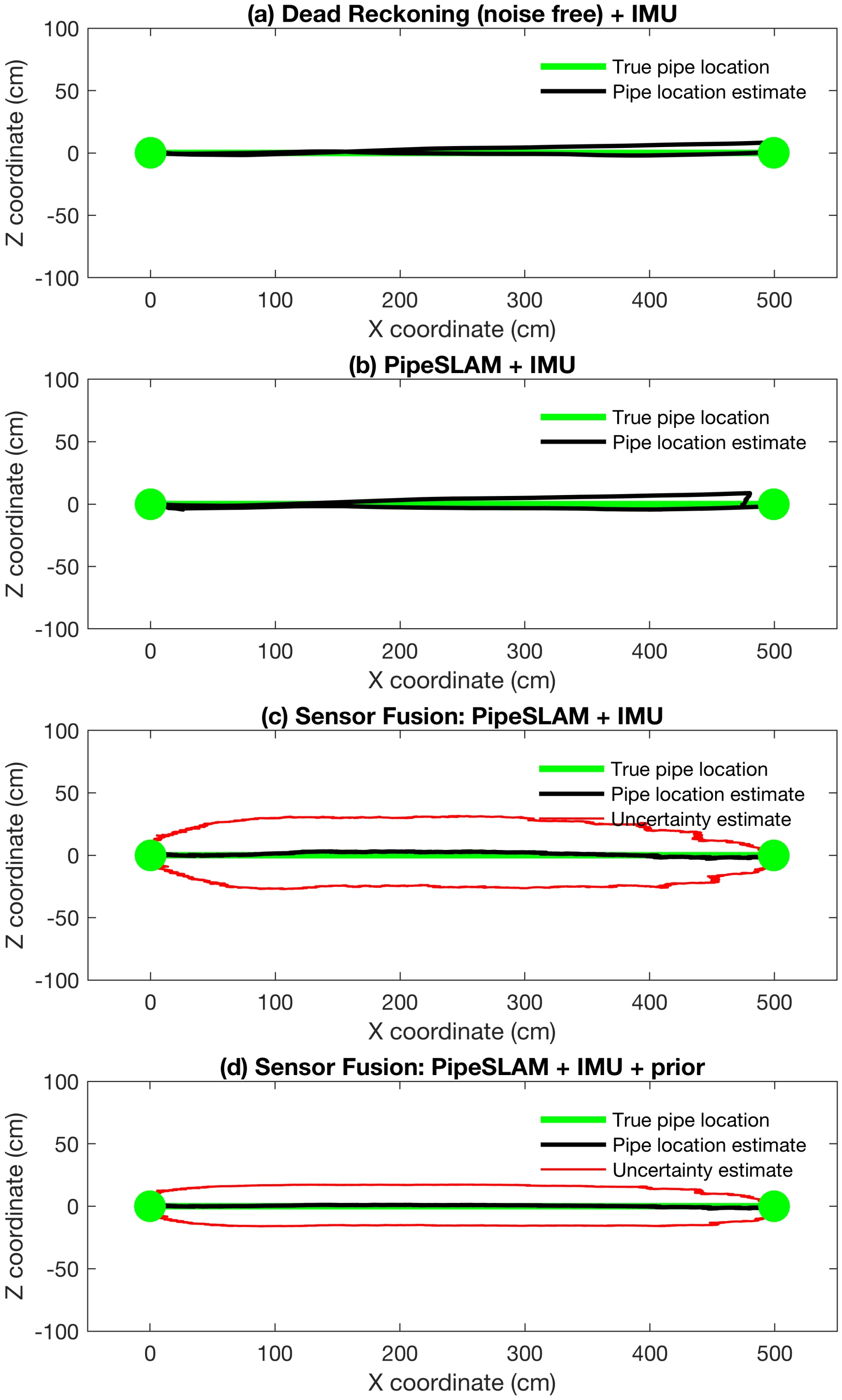


Figure 6.7: Pipe mapping using sensor fusion technique using data set I shown in *xz*-plane. (a) The estimation using noise free dead reckoning distances and IMU data. (b) Estimation using PipeSLAM distance and IMU data. (c) Estimation using sensor fusion of PipeSLAM distance and IMU data, shown with sensor uncertainties. (d) Along with PipeSLAM distance and IMU data, a prior knowledge (assuming the pipe is straight) is added to the estimation.

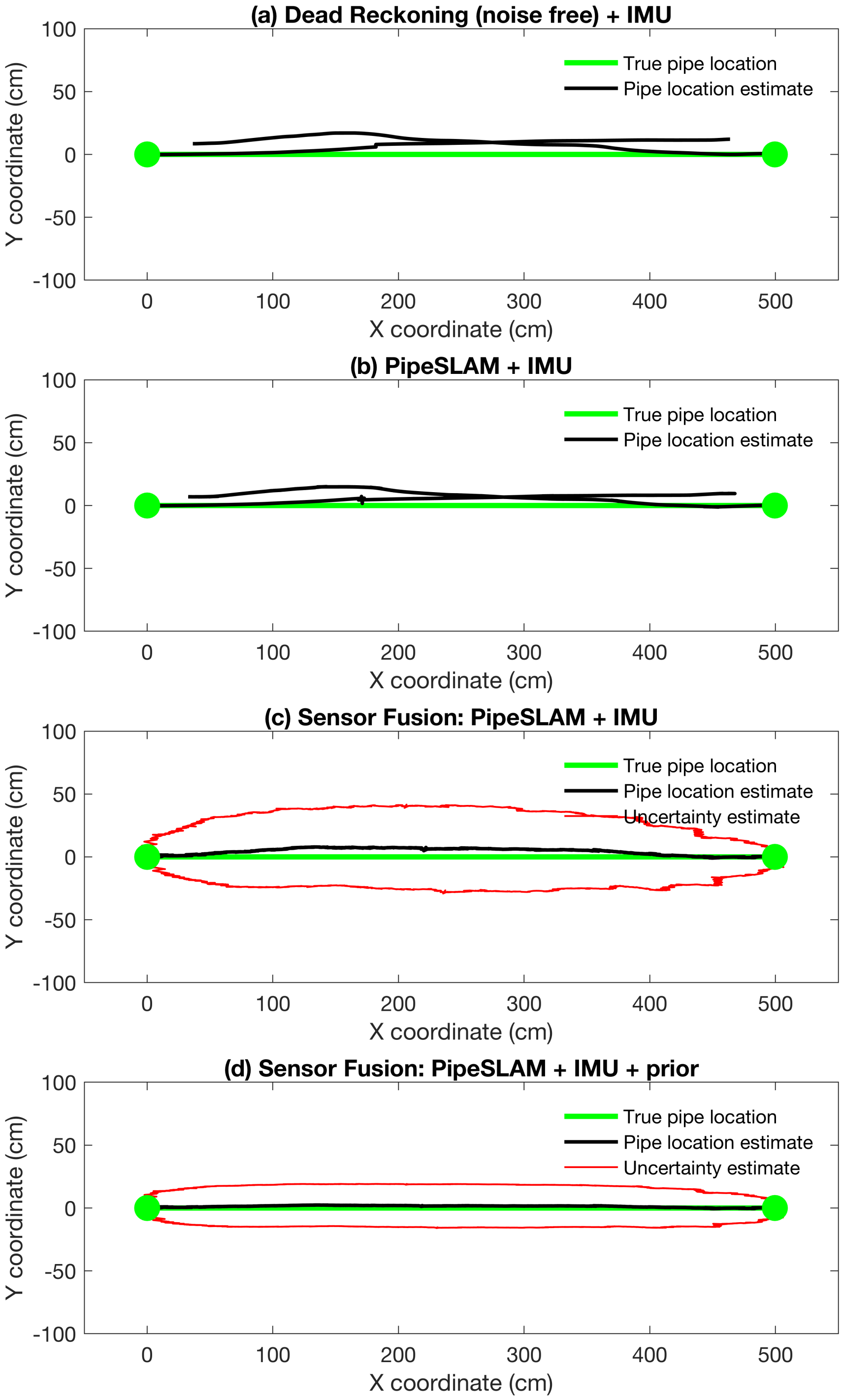


Figure 6.8: Pipe mapping using sensor fusion technique using data set II shown in *xy*-plane. (a) The estimation using noise free dead reckoning distances and IMU data. (b) Estimation using PipeSLAM distance and IMU data. (c) Estimation using sensor fusion of PipeSLAM distance and IMU data, shown with sensor uncertainties. (d) Along with PipeSLAM distance and IMU data, a prior knowledge (assuming the pipe is straight) is added to the estimation.

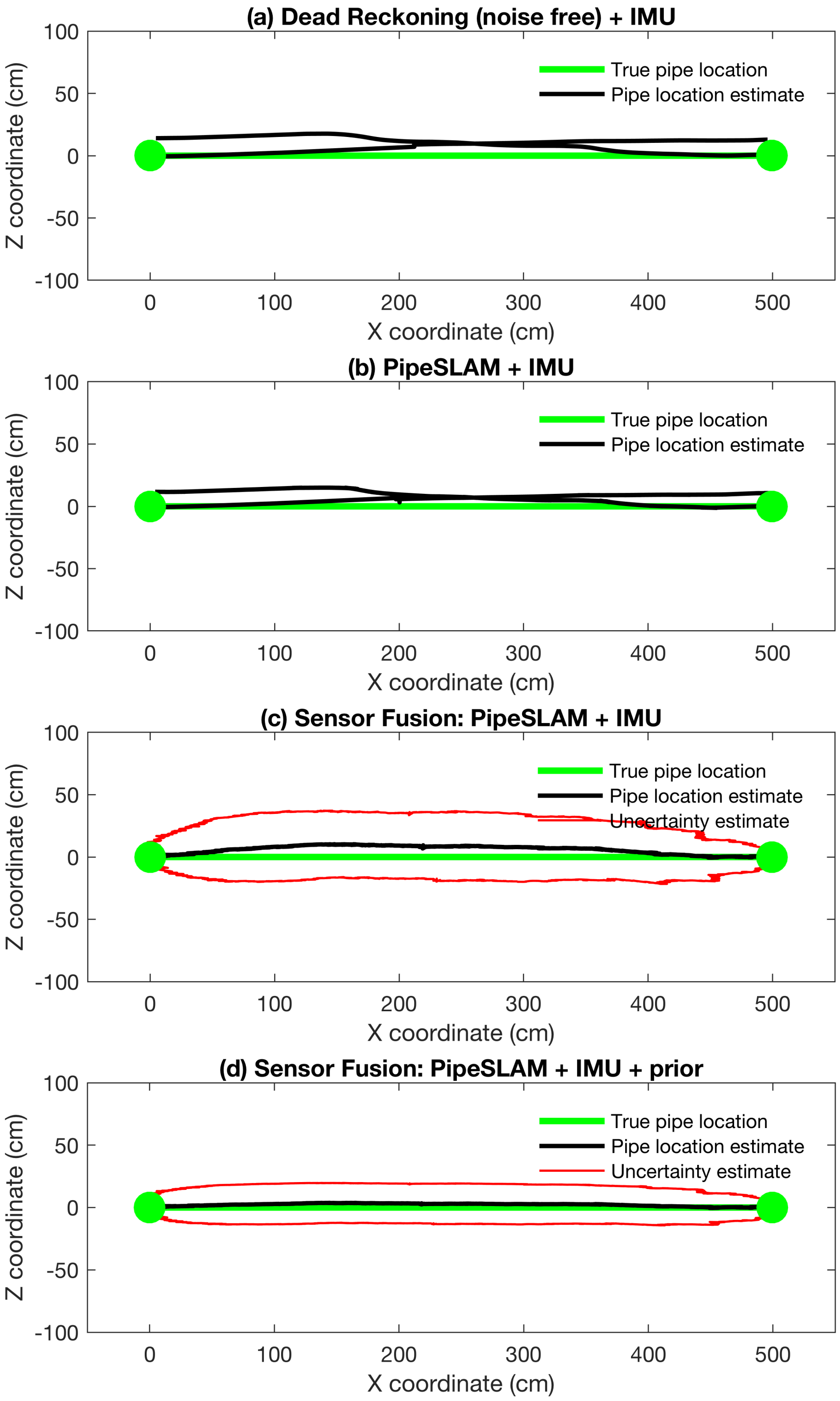


Figure 6.9: Pipe mapping using sensor fusion technique using data set II shown in *xz*-plane. (a) The estimation using noise free dead reckoning distances and IMU data. (b) Estimation using PipeSLAM distance and IMU data. (c) Estimation using sensor fusion of PipeSLAM distance and IMU data, shown with sensor uncertainties. (d) Along with PipeSLAM distance and IMU data, a prior knowledge (assuming the pipe is straight) is added to the estimation.

Chapter 7

Conclusions and Future Work

7.1 Conclusions

The demand for inspection and repair technologies for the water industries on their water mains and distribution pipes is increasing, this can be due to the high cost of damaged infrastructure replacement and more regulations that force water companies to invest on robot inspection [102]. In urban water distribution systems, due to the water pipes are ageing, pipe leakages and serious damage may occur. Compared with the cost pipe replacement in the underground distribution system, regular pipe inspection and repair is more cost effective to water companies and local communities. However, small-diameter pipes are not accessible for human as they are small in size and often buried underground. Therefore, inspection robotic systems are more suitable to this task in terms of underground pipe networks mapping and damage localisation.

This thesis has addressed the problem of mapping and localisation in water pipes and has defined three main challenges for such robot navigation systems. First, due to feature sparsity inside water pipes, robot is not easy to navigate itself based on the featureless in-pipe environment, compared with many indoor and outdoor robot operation environment. Second, in-pipe robot can only detect nearby features by using standard sensors such as laser range finders and ultrasound transducers. Third, unlike other well-developed indoor/outdoor robot navigation systems, an in-pipe robot has less flexibility and can only move either forward or backward.

The aim of this project was address the challenges defined above to build a robot navigation system that could build accurate maps of the pipe while localising itself in the relatively featureless water pipe environment. This aim was successfully achieved through the previous chapters, and the contributions of these chapters are summarised below.

To map the pipe, the robot firstly needs to find observable features in or outside the water pipe. This project has tested two novel approaches to sensing for building maps, in plastic pipes and metal pipes (see chapter 3). One of the novel contributions of this thesis was to use ultrasound sensor to detect the terrain, or soil, profile outside plastic pipe walls, and use the terrain profile as a navigation map. However, such a technique cannot directly translate to metal pipes because ultrasound does not have sufficient energy to penetrate metal pipe walls. Therefore, another novelty of the project was to develop the use of a hydrophone to excite pipe vibration to build a novel type of map for metal pipes. In the hydrophone approach, one hydrophone transmits signals to the metal pipe and the metal pipe will vibrate. The other hydrophone then receives the response signal from the metal pipe vibration. Due to the mechanical characteristics and structure of the pipe, the response signal varies at each location within the pipe, which can be used as map. Therefore, such pipe vibration signals can be used for robot navigation in metal pipes. Both techniques have been tested in different laboratory settings and the navigation maps they produced can be used for sequential mapping and localisation or simultaneous localisation and mapping (SLAM).

In chapter 4 a novel method for sequential mapping and localisation was developed. The maps created from chapter 3 require spatial calibration, which can only be practically performed by a dead reckoning sensor such as a motor encoder. Motor encoder (dead reckoning) measurements may contain various types of noises, such as constant drift and white noise. To spatially calibrate the map therefore, a signal alignment and averaging algorithm named DTW barycentre averaging (DBA) was used. After that the calibrated map was used to perform localisation algorithms. Two localisation algorithms were implemented, which were the extended Kalman filter (EKF) localisation [65] and a standard particle filter (PF) localisation. To test the robustness of the algorithms, additional noises, such as blockage noise in robot motion and extra white noise in ultrasound sensor and hydrophone, were also added in the simulation. The location estimates from both localisation algorithms show much better improvements against the drifted dead reckoning estimation.

In chapter 5 a novel PipeSLAM algorithm was developed for simultaneously localising and mapping in water pipes. The PipeSLAM algorithm was developed based on the Rao-Blackwellised particle filter. The key features of the algorithm are that the map is decomposed by a radial basis function network that is linear-in-parameters; these map parameters are estimated using a Kalman filter; robot location is estimated using particles. The results demonstrated this algorithm could successfully build the map and calculate accurate location estimates along the pipe, compared with dead reckoning location estimates. The key advantage of the PipeSLAM algorithm compared with the previous approach illustrated in Chapter 4 is that PipeSLAM can be further developed into an online SLAM-in-pipe solution in future.

In the previous approaches mentioned above, pipes are assumed to be straight. But it is not always the case in real scenarios, since a pipe network usually contain T junctions and L elbows. Therefore, in order to determine robot location in the xy-plane, directional information is required in addition to robot distance travelled - an inertial measurement unit (IMU) can be used to obtain the directional information. Thus, in addition to sequential mapping and localisation (chapter 4) and PipeSLAM (chapter 5), a sensor fusion with an IMU (Chapter 6) was used to estimate robot location in the world coordinate frame. The robot distance travelled and robot heading are estimated separately. By assuming two known locations (two ends of the pipe), the fused algorithm is able to calculate the shape of the pipe and its location in the world coordinate frame.

7.2 Future Work

Apart from the current development on this robot in-pipe SLAM project, there are many more future research directions to investigate: real-time laboratory and field test; sensor fusion with other different types of sensors; multi-robot operation for mapping entire pipe networks (using cooperative/swarm robotics).

- Further laboratory test and Field test:

    1. Real-time laboratory testing – The experiment data gathered from the laboratory first and the whole set of raw data, including motor encoder readings, hydrophone signals and ultrasound signals, were processed and analysed offline in this project. The sequential mapping and localisation using DBA is designed for offline analysis. But PipeSLAM can be further upgraded for online processing: data can be gathered and processed using the PipeSLAM algorithm along with sensor fusion with IMU, while the robot is travelling through the pipe. As the computer memory and processor technology has been increasing rapidly, a small chip can deal with large work load. This on the other hand also allows all analysis work to be processed onboard.
    2. Field testing – Field testing needs to be performed using the algorithms developed in this thesis. There are sites in the UK where testing can be

performed on buried water pipes, where the location of the buried pipe is known and can be checked for ground truth. The field testing also needs to be done on larger-scale pipes and pipe networks, to demonstrate the effectiveness of PipeSLAM and the sensor fusion technique on more realistic pipe lengths, e.g. 100 metres would be ideal because that is the typical distance between fire hydrants in the UK.

3. From tethered to untethered – At this stage of the project, the sensor platform is tethered to a continuous track powered by two motors on both ends of the pipe. One important reason for tethering is that water companies do not yet have confidence in robotic systems and have concerns that robots might become trapped in the pipe requiring manual recovery (by digging up the road). So for the near future tethering is likely to be standard until the technology is proven. One useful benefit of tethering the robot is that the robot platform effectively has unconstrained power. There are additional advantages in this tethered method: the drive to the system can be effected by actuators outside the pipe to pull the robot back upstream (whilst flow can be used to drive the robot downstream); sensor measurements can be directly transmitted back to the terminal PC for off-board processing with unconstrained computational resources; in the event of the robot becoming stuck in the middle of the pipe, the robot can be recovered by the tethered cable. However, ultimately an untethered in-pipe robot will have much more flexibility and will potentially allow much less human participation. A full autonomous pipe inspection robot is therefore the ultimate goal. To achieve this goal, several difficulties shall be considered: ob-board power system and computational device; wireless communication between the robot and above ground human coordinator; a back up recovery plan in the event of robot failure (possibly using other search and rescue robots).

- Sensor fusion with other types of sensors:

  1. Excluding a motor encoder, the sensor platform is equipped with one pair of hydrophones (or a ultrasound transducer). Using the hydrophone pair as an example, it can only receive one signal at one location at one time. After signal processing, one numeric value indicating the amplitude at that location is the output. Although it can over perform against drifted dead reckoning estimates, it may not always successfully give the true location. Attaching multiple sensor modules (e.g. 3

pairs of hydrophones) of fixed distance between each of them, they can give a set of 3 numeric amplitude readings. This increases the measurement output from 0 dimension (a point) to 1 dimension (a line). This can largely improves the accuracy in both mapping and localisation processes.

2. Sensor fusion with visual cameras, associated visual SLAM algorithms such as a visual SLAM system [85] which uses a set of 4 spinning greyscale CCD cameras capturing 360^o images around sensor platform in the cross section of the pipe. As visual SLAM is a well developed area in robot SLAM, many similar visual SLAM algorithms [56, 86, 103] are also available to integrate into the in-pipe SLAM technology.

   machine learning for recognising objects such junctions, elbows and fire hydrants.

3. Sensor fusion with above ground sensing technology such as ground penetrating radar (GPR) [71, 72, 114].

- A single inspection robot can only travel and inspect a small area of pipe. However, with a group of inspection robots operating in the water pipe network, a global map of the pipe network can be obtained. The health of individual pipe sections within the network can be monitored 24 hours without human participation. Therefore, another direction can be cooperative/swarm robots using distributed SLAM techniques [33]. In addition, the PipeSLAM algorithm developed here is suited to dealing with a section of pipe, at the local scale, but global mapping of the pipe network might be more suited to graph SLAM techniques, which represents the map as vertices and nodes [139], where for instance branches or entry/exit points in the pipe network would be the nodes. This might create an interesting problem/solution of multi-scale SLAM, where small scale mapping-localisation is performed using methods developed in this thesis (between two nodes of a graph), and the larger-scale SLAM problem is addressed through the graph SLAM approach. These techniques would also exploit data from multiple robots cooperating and communicating map data between each other.

Appendix A

Properties of Gaussian Distribution

Lemma A.1 (Joint distribution of Gaussian variables) According to Sarkka [129, p. 209], if random variables $\mathbf{x} \in \mathbb{R}^n$ and $\mathbf{y} \in \mathbb{R}^m$ have the Gaussian probability distributions

$$\begin{aligned} \mathbf{x} &\sim N(\mathbf{m}, \mathbf{P}) \\ \mathbf{y}|\mathbf{x} &\sim N(\mathbf{H}\mathbf{x}, \mathbf{R}) \end{aligned} \tag{A.1}$$

then the joint distribution of $\mathbf{x}$, and $\mathbf{y}$ and the marginal distribution of $\mathbf{y}$ are given as

$$\begin{aligned} \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} &\sim N\left(\begin{pmatrix} \mathbf{m} \\ \mathbf{H}\mathbf{m} \end{pmatrix}, \begin{pmatrix} \mathbf{P} & \mathbf{P}\mathbf{H}^T \\ \mathbf{H}\mathbf{P} & \mathbf{H}\mathbf{P}\mathbf{H}^T + \mathbf{R} \end{pmatrix} \right) \\ \mathbf{y} &\sim N(\mathbf{H}\mathbf{m}, \mathbf{H}\mathbf{P}\mathbf{H} + \mathbf{R}) \end{aligned} \tag{A.2}$$

Lemma A.2 (Conditional distribution of Gaussian variables) According to Sarkka [129, p. 209], if the random variables $\mathbf{x}$, $\mathbf{y}$ have the joint Gaussian probability distribution

$$\begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} \sim N\left(\begin{pmatrix} \mathbf{a} \\ \mathbf{b} \end{pmatrix}, \begin{pmatrix} \mathbf{A} & \mathbf{C} \\ \mathbf{C}^T & \mathbf{B} \end{pmatrix} \right) \tag{A.3}$$

then the marginal and conditional distributions of $\mathbf{x}$, $\mathbf{y}$ are given as follows:

$$\begin{aligned} \mathbf{x} &\sim N(\mathbf{a}, \mathbf{A}) \\ \mathbf{y} &\sim N(\mathbf{b}, \mathbf{B}) \\ \mathbf{x}|\mathbf{y} &\sim N(\mathbf{a} + \mathbf{C}\mathbf{B}^{-1}(\mathbf{y} - \mathbf{b}), \mathbf{A} - \mathbf{C}\mathbf{B}^{-1}\mathbf{C}^T) \\ \mathbf{y}|\mathbf{x} &\sim N(\mathbf{b} + \mathbf{C}^T\mathbf{A}^{-1}(\mathbf{x} - \mathbf{a}), \mathbf{B} - \mathbf{C}^T\mathbf{A}^{-1}\mathbf{C}) \end{aligned} \tag{A.4}$$

Bibliography

[1] S. Anderson, F. Dellaert, and T.D. Barfoot. A hierarchical wavelet decomposition for continuous-time SLAM. In *Proceedings - IEEE International Conference on Robotics and Automation*, pages 373–380, 2014.

[2] R. Aragues, J. Cortes, and C. Sagues. Distributed consensus on robot networks for dynamically merging feature-based maps. *IEEE Transactions on Robotics*, 28(4):840–854, 2012.

[3] M.S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Transactions on Signal Processing*, 50(2):174–188, 2002.

[4] N. Ayache and O.D. Faugeras. Building, Registrating, and Fusing Noisy Visual Maps. *The International Journal of Robotics Research*, 7(6):45–65, 1988.

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

[6] R. Bajcsy. Active perception. *Proceedings of the IEEE*, 76(8):966–1005, 1988.

[7] S.Y. Bao, M. Bagra, Y.W. Chao, and S. Savarese. Semantic structure from motion with points, regions, and objects. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2703–2710, 2012.

[8] B. Barshan and H.F.H. Durrant-Whyte. Inertial navigation systems for mobile robots. *Robotics and Automation, IEEE Transactions on*, 11(3):328–342, 1995.

[9] G. Bishop, G. Welch, G. Bishop, G. Welch, and G. Bishop. An introduction to the Kalman filter. *Proc of SIGGRAPH, Course*, 8(27599-23175):41, 1995.

[10] M. Bosse and R. Zlot. Continuous 3D scan-matching with a spinning 2D laser. *IEEE International Conference on Robotics and Automation*, pages 1–8, 2009.

[11] M. Bosse, P. Newman, J. Leonard, and S. Teller. Simultaneous Localization and Map Building in Large-Scale Cyclic Environments Using the Atlas Framework. *The International Journal of Robotics Research*, 23(12):1113–1139, 2004.

[12] M. Bosse, R. Zlot, and P. Flick. Zebedee: Design of a spring-mounted 3-D range sensor with application to mobile mapping. *IEEE Transactions on Robotics*, 28(5):1104–1119, 2012.

[13] S. Boudjabi, A. Ferreira, and A. Krupa. Modeling and vision-based control of a micro catheter head for teleoperated in-pipe inspection. In *Robotics and Automation, 2003. Proceedings. ICRA. IEEE International Conference on*, volume 3, pages 4282–4287. 2003.

[14] F. Bourgault, A.A. Makarenko, S.B. Williams, B. Grocholsky, and H.F. Durrant-Whyte. Information based adaptive robotic exploration. *IEEE International Conference on Intelligent Robots and Systems*, 1:540–545, 2002.

[15] S.L. Bowman, N. Atanasov, K. Daniilidis, and G.J. Pappas. Probabilistic Data Association for Semantic SLAM. *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1722–1729, 2017.

[16] R. Bradbeer, S. Harrold, F. Nickols, and L.F. Yeung. An underwater robot for pipe inspection. In *Mechatronics and Machine Vision in Practice, 1997. Proceedings., Fourth Annual Conference on*, pages 152–156. 1997.

[17] W. Burgard, M. Moors, C. Stachniss, and F.E. Schneider. Coordinated multi-robot exploration. *IEEE Transactions on Robotics*, 21(3):376–386, 2005.

[18] G. Burgers, P. van Leeuwen, G. Evensen, P. Jan van Leeuwen, and G. Evensen. Analysis scheme in the ensemble Kalman filter. *Monthly weather review*, 126(6):1719–1724, 1998.

[19] C. Cadena, A. Dick, and I.D. Reid. A fast, modular scene understanding system using context-aware object detection. In *Proceedings - IEEE International Conference on Robotics and Automation*, pages 4859–4866, 2015.

[20] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I.D. Reid, and J.J. Leonard. Past, Present, and Future of Simultaneous Localization And Mapping: Towards the Robust-Perception Age. *IEEE Transactions on Robotics*, 32(6):1–27, 2016.

[21] N. Carlevaris-Bianco and R.M. Eustice. Generic factor-based node marginalization and edge sparsification for pose-graph SLAM. In *Proceedings - IEEE International Conference on Robotics and Automation*, pages 5748–5755, 2013.

[22] L. Carlone, J. Du, M. Kaouk Ng, B. Bona, and M. Indri. Active SLAM and exploration with particle filters using Kullback-Leibler divergence. *Journal of Intelligent and Robotic Systems: Theory and Applications*, 75(2):291–311, 2014.

[23] H. Carrillo, Y. Latif, J. Neira, and J.A. Castellanos. Fast minimum uncertainty search on a graph map representation. In *IEEE International Conference on Intelligent Robots and Systems*, pages 2504–2511, 2012.

[24] H. Carrillo, I. Reid, and J.A. Castellanos. On the comparison of uncertainty criteria for active SLAM. In *Proceedings - IEEE International Conference on Robotics and Automation*, pages 2080–2087, 2012.

[25] H. Carrillo, P. Dames, V. Kumar, and J.A. Castellanos. Autonomous robotic exploration using occupancy grid maps and graph SLAM based on Shannon and Rényi Entropy. In *Proceedings - IEEE International Conference on Robotics and Automation*, volume 2015-June, pages 487–494, 2015.

[26] R. Chatila and J. Laumond. Position referencing and consistent world modeling for mobile robots. *Robotics and Automation. Proceedings. 1985 IEEE International Conference on*, 2(APRIL):138–145, 1985.

[27] Z. Chen. Bayesian filtering: From Kalman filters to particle filters, and beyond. *Statistics*, 182(1):1–69, 2003.

[28] R.M. Clark, M. Sivaganesan, A. Selvakumar, and V. Sethi. Cost models for water supply distribution systems. *Journal of Water Resources Planning and Management*, 128(5):312–321, 2002.

[29] S. Costello, D. Chapman, C. Rogers, and N. Metje. Underground asset location and condition assessment technologies. *Tunnelling and Underground Space Technology*, 22(5-6):524–542, 2007.

[30] J. Crowley. World modeling and position estimation for a mobile robot using ultrasonic ranging. *IEEE Conference on Robotics and Automation (ICRA)*, 3:674–680, 1989.

[31] M. Csorba. *Simultaneous localisation and map building*. PhD thesis, University of Oxford, 1998.

[32] M. Csorba, J.K. Uhlmann, and H.F. Durrant-Whyte. New approach to simultaneous localization and dynamic map building. In *Navigation and Control Technologies for Unmanned Systems*, volume 2738, pages 26–37. International Society for Optics and Photonics, 1996.

[33] A. Cunningham, M. Paluri, and F. Dellaert. Ddf-sam: Fully distributed slam using constrained factor graphs. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 3025–3030. IEEE, 2010.

[34] A. Cunningham, V. Indelman, and F. Dellaert. DDF-SAM 2.0: Consistent distributed smoothing and mapping. In *Proceedings - IEEE International Conference on Robotics and Automation*, pages 5220–5227, 2013.

[35] F. Dellaert, J. Carlson, V. Ila, K. Ni, and C.E. Thorpe. Subgraph-preconditioned Conjugate Gradient for Large Scale SLAM. *IEEE Transactions on Robotics*, pages 2566–2571, 2010.

[36] F. Dellaert and M. Kaess. Square Root SAM: Simultaneous Localization and Mapping via Square Root Information Smoothing. *Int. J. Rob. Res.*, 25(12): 1181–1203, 2006.

[37] G. Dissanayake, S. Huang, Z. Wang, and R. Ranasinghe. A review of recent developments in simultaneous localization and mapping. In *Industrial and Information Systems (ICIIS), 2011 6th IEEE International Conference on*, pages 477–482. IEEE, 2011.

[38] M.W.M.G. Dissanayake, P. Newman, S. Clark, H.F. Durrant-Whyte, and M. Csorba. A solution to the simultaneous localization and map building (SLAM) problem. *IEEE Transactions on Robotics and Automation*, 17(3): 229–241, 2001.

[39] J. Dong, E. Nelson, V. Indelman, N. Michael, and F. Dellaert. Distributed real-time cooperative localization and mapping using an uncertainty-aware expectation maximization approach. *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5807–5814, 2015.

[40] A. Doucet and A.M.A.A.M. Johansen. A tutorial on particle filtering and smoothing: fifteen years later. *Handbook of Nonlinear Filtering*, 12(3):656–704, 2009.

[41] A. Doucet, N. De Freitas, K. Murphy, and S. Russell. Rao-blackwellised particle filtering for dynamic bayesian networks. In *Proceedings of the Sixteenth*

conference on Uncertainty in artificial intelligence, pages 176–183. Morgan Kaufmann Publishers Inc., 2000.

[42] A. Doucet, N. de Freitas, N. Gordon, N.D. Freitas, N. Gordon, N. de Freitas, and N. Gordon. An Introduction to Sequential Monte Carlo Methods. In *Sequential Monte Carlo Methods in Practice*, pages 3–14. Springer New York, New York, NY, 2001.

[43] R. Dubé, H. Sommer, A. Gawel, M. Bosse, and R. Siegwart. Non-uniform sampling strategies for continuous correction based trajectory estimation. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4792–4798, 2016.

[44] O. Duran, K. Althoefer, and L.D. Seneviratne. Automated sewer pipe inspection through image processing. In *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, volume 3, pages 2551–2556. 2002.

[45] O. Duran, K. Althoefer, and L.D. Seneviratne. Pipe inspection using a laser-based transducer and automated analysis techniques. *Mechatronics, IEEE/ASME Transactions on*, 8(3):401–409, 2003.

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

[47] H. Durrant-whyte, D. Rye, and E. Nebot. Localization of Autonomous Guided Vehicles. In *Robotics Research*, pages 613–625. Springer, 1996.

[48] H.F. Durrant-Whyte. Uncertain Geometry in Robotics. *IEEE Journal on Robotics and Automation*, 4(1):23–31, 1988.

[49] N. Engelhard, F. Endres, J. Hess, J. Sturm, and W. Burgard. Real-time 3d visual slam with a hand-held rgb-d camera. In *Proc. of the RGB-D Workshop on 3D Perception in Robotics at the European Robotics Forum, Vasteras, Sweden*, volume 180, pages 1–15, 2011.

[50] G. Evensen. The ensemble Kalman filter: Theoretical formulation and practical implementation. *Ocean dynamics*, 53(4):343–367, 2003.

[51] J. Folkesson and H.I. Christensen. Graphical SLAM for outdoor applications. *J. F. Robot.*, 24(1-2):51–70, 2007.

[52] U. Frese. A proof for the approximate sparsity of slam information matrices. In *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, pages 329–335. IEEE, 2005.

[53] P. Furgale, T.D. Barfoot, and G. Sibley. Continuous-time batch estimation using temporal basis functions. In *Proceedings - IEEE International Conference on Robotics and Automation*, pages 2088–2095, 2012.

[54] G. Gallegos and P. Rives. Indoor slam based on composite sensor mixing laser scans and omnidirectional images. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 3519–3524. IEEE, 2010.

[55] S. García, M.E. López, R. Barea, L.M. Bergasa, A. Gómez, and E.J. Molinos. Indoor slam for micro aerial vehicles control using monocular camera and sensor fusion. In *Autonomous Robot Systems and Competitions (ICARSC), 2016 International Conference on*, pages 205–210. IEEE, 2016.

[56] A. Gil, Ó. Reinoso, M. Ballesta, and M. Juliá. Multi-robot visual slam using a rao-blackwellized particle filter. *Robotics and Autonomous Systems*, 58(1): 68–80, 2010.

[57] S. Gillijns, O.B. Mendoza, J. Chandrasekar, B.L.R. De Moor, D.S. Bernstein, and a. Ridley. What is the ensemble Kalman filter and how well does it work? In *2006 American Control Conference*, page 6 pp. 2006.

[58] N.N.J. Gordon, D.D.J. Salmond, and A.A.F.M. Smith. Novel-Approach to Nonlinear Non-Gaussian Bayesian State Estimation. *IEE Proceedings F (Radar and Signal Processing)*, 140(2):107–113, 1993.

[59] G. Grisetti, C. Stachniss, and W. Burgard. Improving grid-based slam with rao-blackwellized particle filters by adaptive proposals and selective resampling. In *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, pages 2432–2437. 2005.

[60] G. Grisetti, C. Stachniss, and W. Burgard. Improved techniques for grid mapping with rao-blackwellized particle filters. *Robotics, IEEE Transactions on*, 23(1):34–46, 2007.

[61] G. Grisetti, C. Stachniss, S. Grzonka, and W. Burgard. A tree parameterization for efficiently computing maximum likelihood maps using gradient descent. *Rss*, page 8, 2007.

[62] G. Grisetti, R. Kummerle, C. Stachniss, U. Frese, and C. Hertzberg. Hierarchical optimization on manifolds for online 2D and 3D mapping. In *Proceedings - IEEE International Conference on Robotics and Automation*, pages 273–278, 2010.

[63] M. Hamilton. Assessing the Underworld, 2013. URL `http://assessingtheunderworld.org/about/`. [Online; accessed December 17, 2017].

[64] S. Haykin. *Kalman filtering and neural networks*. Wiley Online Library, 2001.

[65] L.D. Hostetler and R.D. Andreas. Nonlinear Kalman filtering techniques for terrain-aided navigation. *IEEE Transactions on Automatic Control*, 28(3): 315–323, mar 1983.

[66] P.L. Houtekamer and H.L. Mitchell. Data assimilation using an ensemble Kalman filter technique. *Monthly Weather Review*, 126(3):796–811, 1998.

[67] O. Hunaidi and P. Giamou. Ground-penetrating radar for detection of leaks in buried plastic water distribution pipes. In *International Conference on Ground Penetrating Radar*, 1998.

[68] IEEE RAS Map Data Representation Working Group. IEEE Standard for Robot Map Data Representation for Navigation, Sponsor: IEEE Robotics and Automation Society, 2016. URL `http://standards.ieee.org/findstds/standard/1873-2015.html/`.

[69] V. Ila, J.M. Porta, and J. Andrade-Cetto. Information-based compact pose SLAM. *IEEE Transactions on Robotics*, 26(1):78–93, 2010.

[70] V. Indelman and L. Carlone. Planning in the continuous domain: A generalized belief space approach for autonomous navigation in unknown environments. *The International Journal*, 2015.

[71] W. Jiang, S. Pennock, and P. Shepherd. A novel w-music algorithm for gpr target detection in noisy and distorted signals. In *Radar Conference, 2009 IEEE*, pages 1–6. IEEE, 2009.

[72] W. Jiang, S. Pennock, and P. Shepherd. Novel short-time music in nonlinear fmcw gpr signal analysis. In *Ground Penetrating Radar (GPR), 2010 13th International Conference on*, pages 1–6. IEEE, 2010.

[73] H. Johannsson, M. Kaess, M. Fallon, and J.J. Leonard. Temporally scalable visual SLAM using a reduced pose graph. In *Proceedings - IEEE International Conference on Robotics and Automation*, pages 54–61, 2013.

[74] S.J. Julier and J.K. Uhlmann. New extension of the Kalman filter to nonlinear systems. In *AeroSense'97, International Society for Optics and Photonics*, pages 182–193. 1997.

[75] V. Kadirkamanathan and M. Niranjan. A function estimation approach to sequential learning with neural networks. *Neural Computation*, 5(6):954–975, 1993.

[76] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J.J. Leonard, and F. Dellaert. iSAM2: Incremental smoothing and mapping using the Bayes tree. *Int. J. Rob. Res.*, 31(2):216–235, 2012.

[77] R.E. Kalman. A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, 82(1):35–45, 1960.

[78] M. Keidar and G. Kaminka. Efficient frontier detection for robot exploration. In *The International Journal of Robotics Research*, pages 215—-236. 2014.

[79] J.H. Kim, C. Cadena, and I. Reid. Direct semi-dense SLAM for rolling shutter cameras. In *Proceedings - IEEE International Conference on Robotics and Automation*, volume 2016-June, pages 1308–1315, 2016.

[80] G. Kitagawa. Monte Carlo Filter and Smoother for Non-Gaussian Nonlinear State Space Models. *Journal of Computational and Graphical Statistics*, 5(1): 1–25, 1996.

[81] G. Klein and D. Murray. Parallel tracking and mapping for small AR workspaces. In *2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality, ISMAR*, 2007.

[82] J. Knuth and P. Barooah. Collaborative localization with heterogeneous inter-robot measurements by Riemannian optimization. In *Proceedings - IEEE International Conference on Robotics and Automation*, pages 1534–1539, 2013.

[83] S. Kohlbrecher, O. Von Stryk, J. Meyer, and U. Klingauf. A flexible and scalable slam system with full 3d motion estimation. In *Safety, Security, and Rescue Robotics (SSRR), 2011 IEEE International Symposium on*, pages 155–160. IEEE, 2011.

[84] H. Kretzschmar, C. Stachniss, and G. Grisetti. Efficient information-theoretic graph pruning for graph-based SLAM with laser range finders. In *IEEE International Conference on Intelligent Robots and Systems*, pages 865–871, 2011.

[85] D. Krys and H. Najjaran. Development of visual simultaneous localization and mapping (VSLAM)for a pipe inspection robot. In *Computational Intelligence in Robotics and Automation, 2007. CIRA 2007. International Symposium on*, pages 344–349. 2007.

[86] H. Lategahn, A. Geiger, and B. Kitt. Visual slam for autonomous ground vehicles. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 1732–1737. IEEE, 2011.

[87] M.T. Lazaro, L.M. Paz, P. Pinies, J.A. Castellanos, and G. Grisetti. Multi-robot SLAM using condensed measurements. In *IEEE International Conference on Intelligent Robots and Systems*, pages 1069–1076, 2013.

[88] D.H. Lee, H. Moon, and H.R. Choi. Autonomous navigation of in-pipe working robot in unknown pipeline Environment. In *Proceedings - IEEE International Conference on Robotics and Automation*, pages 1559–1564. 2011.

[89] C. Leung, S. Huang, N. Kwok, and G. Dissanayake. Planning under uncertainty using model predictive control for information gathering. *Robotics and Autonomous Systems*, 54(11):898–910, 2006.

[90] C. Leung, S. Huang, and G. Dissanayake. Active SLAM using model predictive control and attractor based exploration. In *IEEE International Conference on Intelligent Robots and Systems*, pages 5026–5031, 2006.

[91] H. Lim, J.Y. Choi, Y.S. Kwon, E.J.J. Jung, and B.J.J. Yi. SLAM in indoor pipelines with 15mm diameter. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pages 4005–4011. 2008.

[92] Z. Liu and Y. Kleiner. State of the art review of inspection technologies for condition assessment of water pipes. *Measurement*, 46(1):1–15, 2013.

[93] Z. Liu and D. Krys. The use of laser range finder on a robotic platform for pipe inspection. *Mechanical Systems and Signal Processing*, 31:246–257, 2012.

[94] S. Lowry, N. Sunderhauf, P. Newman, J.J. Leonard, D. Cox, P. Corke, and M.J. Milford. Visual Place Recognition: A Survey. *IEEE Trans. Robot.*, 32(1): 1–19, 2016.

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

[96] K. Ma, J. Zhu, T.J. Dodd, R. Collins, and S.R. Anderson. Robot Mapping and Localisation for Feature Sparse Water Pipes Using Voids as Landmarks. In *Towards Autonomous Robotic Systems: 16th Annual Conference, TAROS 2015*, pages 161–166, Liverpool, UK, 2015.

[97] K. Ma, M. Schirru, A.H. Zahraee, R. Dwyer-Joyce, J. Boxall, T.J. Dodd, R. Collins, and S.R. Anderson. Pipeslam: Simultaneous localisation and

mapping in feature sparse water pipes using the rao-blackwellised particle filter. In *Advanced Intelligent Mechatronics (AIM), 2017 IEEE International Conference on*, pages 1459–1464. IEEE, 2017.

[98] K. Ma, M.M. Schirru, A.H. Zahraee, R. Dwyer-Joyce, J. Boxall, T.J. Dodd, R. Collins, and S.R. Anderson. Robot mapping and localisation in metal water pipes using hydrophone induced vibration and map alignment by dynamic time warping. In *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, pages 2548–2553. IEEE, 2017.

[99] R. Martinez-Cantin, N. De Freitas, E. Brochu, J. Castellanos, and A. Doucet. A Bayesian exploration-exploitation approach for optimal online sensing and planning with a visually guided mobile robot. *Autonomous Robots*, 27 (2):93–103, 2009.

[100] M. Mazuran, W. Burgard, and G.D. Tipaldi. Nonlinear factor recovery for long-term SLAM. *The International Journal of Robotics Research*, 35(1-3):50–72, 2016.

[101] B. Mergelas and G. Henrich. Leak locating method for precommissioned transmission pipelines: North american case studies. *Leakage 2005*, pages 12–14, 2005.

[102] J.M. Mirats Tur and W. Garthwaite. Robotic devices for water main in-pipe inspection: A survey. *Journal of Field Robotics*, 27(4):491–508, 2010.

[103] F.M. Mirzaei and S.I. Roumeliotis. A Kalman-filter-based algorithm for IMU-camera calibration: observability analysis and performance evaluation. . *IEEE Trans. Robot.*, 24(5):1143–1156, 2008.

[104] M. Montemerlo, S. Thrun, B. Siciliano, D. Koller, B. Wegbreit, and Others. FastSLAM: A factored solution to the simultaneous localization and mapping problem. In *AAAI/IAAI*, pages 593–598, Berlin Heidelberg, 2002.

[105] E. Mueggler, G. Gallego, and D. Scaramuzza. Continuous-Time Trajectory Estimation for Event-based Vision Sensors. In *Robotics: Science and Systems XI*, 2015.

[106] R. Mur-Artal, J.M. Montiel, and J.D. Tardos. ORB-SLAM: A Versatile and Accurate Monocular SLAM System. *IEEE Transactions on Robotics*, 31(5):1147–1163, 2015.

[107] M. Muramatsu, N. Namiki, R. Koyama, and Y. Suga. Autonomous mobile robot in pipe for piping operations. In *Intelligent Robots and Systems, 2000.(IROS 2000). Proceedings. 2000 IEEE/RSJ International Conference on*, volume 3, pages 2166–2171. 2000.

[108] A.C. Murtra and J.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.

[109] M. Nakhkash and M.R. Mahmood-Zadeh. Water leak detection using ground penetrating radar. In *Ground Penetrating Radar, 2004. GPR 2004. Proceedings of the Tenth International Conference on*, pages 525–528. IEEE, 2004.

[110] W. Neubauer. A spider-like robot that climbs vertically in ducts or pipes. In *Intelligent Robots and Systems 94. Advanced Robotic Systems and the Real World', IROS'94. Proceedings of the IEEE/RSJ/GI International Conference on*, volume 2, pages 1178–1185. 1994.

[111] K. Ni and F. Dellaert. Multi-level submap based SLAM using nested dissection. In *IEEE/RSJ 2010 International Conference on Intelligent Robots and Systems, IROS 2010 - Conference Proceedings*, pages 2558–2565, 2010.

[112] G. Nützi, S. Weiss, D. Scaramuzza, and R. Siegwart. Fusion of imu and vision for absolute scale estimation in monocular slam. *Journal of intelligent & robotic systems*, 61(1-4):287–299, 2011.

[113] E. Olson, J. Leonard, and S. Teller. Fast iterative alignment of pose graphs with poor initial estimates. In *Proc. - IEEE Int. Conf. Robot. Autom.*, volume 2006, pages 2262–2269, 2006.

[114] S.R. Pennock, C.H.J. Jenks, G. Orlando, and M.A. Redfern. In-pipe gpr configuration and the determination of target depth and ground permittivity. In *Geoscience and Remote Sensing Symposium (IGARSS), 2012 IEEE International*, pages 618–621. IEEE, 2012.

[115] F. Petitjean and P. Gançarski. Summarizing a set of time series by averaging: From Steiner sequence to compact multiple alignment. *Theoretical Computer Science*, 414(1):76–91, 2012.

[116] F. Petitjean, A. Ketterlin, and P. Gançarski. A global averaging method for dynamic time warping, with applications to clustering. *Pattern Recognition*, 44(3):678–693, 2011.

[117] F. Petitjean, G. Forestier, G.I. Webb, A.E. Nicholson, Y. Chen, and E. Keogh. Dynamic time warping averaging of time series allows faster and more accurate classification. In *2014 IEEE International Conference on Data Mining*, pages 470–479. IEEE, 2014.

[118] J. Platt. A resource-allocating network for function interpolation. *Neural Computation*, 3(2):213–225, 1991.

[119] A. PÃązman. *Foundations of optimum experimental design*. Springer, New York, NY, 1986.

[120] Rasbak. Sewer plastic pipelines, 2006. URL `https://commons.wikimedia.org/wiki/File:Rioolbuizen_van_kunststof_(Sewer_plastic_pipelines).jpg`. [Online; accessed August 16, 2018].

[121] D.D. Ratnayaka, M.J. Brandt, and K.M. Johnson. *Twort's Water Supply*. Butterworth-Heinemann, Oxford, UK, 6 edition, 2009.

[122] L. Riazuelo, J. Civera, and J.M. Montiel. C2TAM: A Cloud framework for cooperative tracking and mapping. *Robotics and Autonomous Systems*, 62(4): 401–413, 2014.

[123] M.I. Ribeiro. Kalman and extended kalman filters: Concept, derivation and properties. *Institute for Systems and Robotics*, 43(February), 2004.

[124] S.G. Roh, S. Ryew, J.H. Yang, and H.R. Choi. Actively steerable in-pipe inspection robots for underground urban gas pipelines. In *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, volume 1, pages 761–766. 2001.

[125] S.g. Roh and H.R. Choi. Differential-drive in-pipe robot for moving inside urban gas pipelines. *Robotics, IEEE Transactions on*, 21(1):1–17, 2005.

[126] H. Roth, K. Schilling, S. Futterknecht, U. Weigele, and M. Reisch. Inspection and repair robots for waste water pipes a challenge to sensorics and locomotion. In *Industrial Electronics, 1998. Proceedings. ISIE'98. IEEE International Symposium on*, volume 2, pages 476–478. 1998.

[127] S. Saeedi, M. Trentini, M. Seto, and H. Li. Multiple-Robot Simultaneous Localization and Mapping: A Review. *Journal of Field Robotics*, 33(1):3–46, 2016.

[128] R.F. Salas-Moreno, R.A. Newcombe, H. Strasdat, P.H. Kelly, and A.J. Davison. SLAM++: Simultaneous localisation and mapping at the level of objects.

In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1352–1359, 2013.

[129] S. Sarkka. *Bayesian filtering and smoothing*, volume 3. Cambridge University Press, Cambridge, UK, 2013.

[130] H. Schempf. In pipe assessment robot platforms. Technical report, Tech. Rep., The Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, USA, 2004.

[131] E. Shammas, A. Wolf, and H. Choset. Three degrees-of-freedom joint for spatial hyper-redundant robots. *Mechanism and machine theory*, 41(2):170–190, 2006.

[132] D. Sibley and C. Mei. Adaptive relative bundle adjustment. *Robotics: science and systems*, 220(3):1–8, 2009.

[133] R. Sim, P. Elinas, M. Griffin, J.J. Little, et al. Vision-based slam using the rao-blackwellised particle filter. In *IJCAI Workshop on Reasoning with Uncertainty in Robotics*, volume 14, pages 9–16, 2005.

[134] R. Smith, M. Self, and P. Cheeseman. Estimating Uncertain Spatial Relationships in Robotics. *Autonomous Robot Vehicles*, 4(April):167–193, 1990.

[135] R.C. Smith and P. Cheeseman. On the Representation and Estimation of Spatial Uncertainty. *The International Journal of Robotics Research*, 5(4):56–68, 1986.

[136] H. Strasdat, A.J. Davison, J.M. Montiel, and K. Konolige. Double window optimisation for constant time visual SLAM. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2352–2359, 2011.

[137] K. Suzumori, K. Hori, and T. Miyagawa. A direct-drive pneumatic stepping motor for robots: designs for pipe-inspection microrobots and for human-care robots. In *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on*, volume 4, pages 3047–3052. 1998.

[138] S. Thrun and M. Montemerlo. The Graph SLAM Algorithm with Applications to Large-Scale Mapping of Urban Structures. *Int. J. Rob. Res.*, 25(5-6): 403–429, 2006.

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

[140] S. Thrun, W. Burgard, and D. Fox. A probabilistic approach to concurrent mapping and localization for mobile robots. *Autonomous Robots*, 5(3-4):253–271, 1998.

[141] S. Thrun, M. Montemerlo, D. Koller, B. Wegbreit, J. Nieto, and E. Nebot. Fastslam: An efficient solution to the simultaneous localization and mapping problem with unknown data association. *Journal of Machine Learning Research*, 4(3):380–407, 2004.

[142] S. Thrun, W. Burgard, and D. Fox. *Probabilistic robotics*. MIT Press, Cambridge, MA, USA, 2005.

[143] S.B. Thrun. Exploration in active learning. *Handbook of Brain and Cognitive Science*, pages 381–384, 1995.

[144] G.D. Tipaldi and K.O. Arras. Flirt-interest regions for 2d range data. In *Robot. Autom. (ICRA), 2010 IEEE Int. Conf.*, pages 3616–3622. 2010.

[145] C.H. Tong, P. Furgale, and T.D. Barfoot. Gaussian Process Gauss-Newton for non-parametric simultaneous localization and mapping. *The International Journal of Robotics Research*, 32(5):507–525, 2013.

[146] A.C. Twort, D.D. Ratnayaka, M.J. Brandt, and Others. *Water supply*. Butterworth-Heinemann, 2000.

[147] J. Vallvé and J. Andrade-Cetto. Potential information fields for mobile robot exploration. *Robotics and Autonomous Systems*, 69(1):68–79, 2015.

[148] J. van den Berg, S. Patil, and R. Alterovitz. Motion planning under uncertainty using iterative local optimization in belief space. *The International Journal of Robotics Research*, 31(11):1263–1278, 2012.

[149] S. Wakimoto, J. Nakajima, M. Takata, T. Kanda, and K. Suzumori. A micro snake-like robot for small pipe inspection. In *Micromechatronics and Human Science, 2003. MHS 2003. Proceedings of 2003 International Symposium on*, pages 303–308. 2003.

[150] M.R. Walter, R.M. Eustice, and J.J. Leonard. Exactly sparse extended information filters for feature-based slam. *The International Journal of Robotics Research*, 26(4):335–359, 2007.

[151] Z. Wang, S. Huang, and G. Dissanayake. D-slam: A decoupled solution to simultaneous localization and mapping. *The International Journal of Robotics Research*, 26(2):187–204, 2007.

[152] S. Williams, V. Indelman, M. Kaess, R. Roberts, J.J. Leonard, and F. Dellaert. Concurrent filtering and smoothing: A parallel architecture for real-time navigation and full smoothing. *The International Journal of Robotics Research*, 33(12):1544–1568, 2014.

[153] B. Yamauchi. A frontier-based approach for autonomous exploration. In *Proceedings 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation CIRA'97. 'Towards New Computational Principles for Robotics and Automation'*, pages 146–151, 1997.

[154] X. Zeng and G.A. McMechan. Gpr characterization of buried tanks and pipes. *Geophysics*, 62(3):797–806, 1997.

[155] L. Zhao, S. Huang, and G. Dissanayake. Linear SLAM: A linear solution to the feature-based and pose graph SLAM based on submap joining. In *IEEE International Conference on Intelligent Robots and Systems*, pages 24–30, 2013.

[156] J. Zhu, R.P. Collins, J.B. Boxall, R.S. Mills, and R. Dwyer-joyce. Non-destructive in-situ condition assessment of plastic pipe using ultrasound. *Procedia Engineering*, 119:148–157, 2015.