From Estimation to Control for Robotic Navigation: Probabilistic and Optimal Approaches

by

Pedram Agand

M.Sc., K. N. Toosi University of Technology, 2017
B.Sc., K. N. Toosi University of Technology, 2014

Thesis Submitted in Partial Fulfillment of the
Requirements for the Degree of
Master of Science

in the
Department of Computing Science
Faculty of Applied Sciences

SIMON FRASER UNIVERSITY
Summer 2021

Declaration of Committee

Name:	**Pedram Agand**
Degree:	**Master of Science**
Thesis title:	**From Estimation to Control for Robotic Navigation: Probabilistic and Optimal Approaches**
Committee:	**Chair:** Manolis Savva Assistant Professor, Computing Science
	Mo Chen Supervisor Assistant Professor, Computing Science
	Angelica Lim Committee Member Assistant Professor, Computing Science
	Hang Ma Examiner Assistant Professor, Computing Science

Abstract

Nowadays, mobile robots capable of autonomous navigation and interaction in unfamiliar and dynamic environments have received great attention among researchers. The robot must be able to precisely perceive its environment, make appropriate inference, plan its path, and travel around safely in order to achieve this goal. In robotics, maneuvering in a complex setting has been challenging. Several methods propose robust architectures in which the agent acts conservative in respect to uncertainty by considering worst case scenario, while others provide adaptive policies which try to adjust the actions given the concurrent knowledge. The usually suffers from guaranteed stability and efficiency in data. The novelty in this report is two folded: the first is to suggest a probabilistic framework for estimating environmental hazards and dynamic models. By improving MCMC, we propose an online method to obtain the model parameters distribution. The second novelty, is to propose an inference model and update framework for human navigational intent. We will discuss how one can apply these insights in a safe path planning problem by considering the environment's uncertainty in a probabilistic manner.

Keywords: Human navigation, Optimal control, MCMC, Bayesian optimization, Probabilistic approach, Path planning, Inference model.

Dedication

To my family and numerous friends, I dedicate my dissertation effort. I owe a special debt of appreciation to my devoted parents, whose words of encouragement and insistence on perseverance still reverberate in my ears. My sisters and brother are extremely precious to me and have never left my side.

This dissertation is also dedicated to my numerous friends and lab colleagues who have helped me along the way. I'll be eternally grateful for everything they've done to help me improve my technical skills, for the countless hours of proofreading, and for teaching me how to master the leader dots.

Acknowledgements

First of all, I would like to warmly thank Dr. Mo Chen, my senior supervisor for the technical help he provided throughout the entire project, and his invaluable comments and guidance. He paid close attention to the progress of my work. This project would not have been possible without his support. I would like to express my greatest gratitude to my academic supervisors Angelica Lim, for her answers to my questions and all support they have given me in the human intent prediction during the master thesis. Finally, many thanks to my examiner Dr. Hang Ma for his interest in the master thesis project and for taking the time to give me feedback and suggestions to improve the quality of my work.

I would like to thank all the members of the MARS Lab team; Mahdi Taher Ahmadi in particular, for collaborating to integrate the human navigational model and processing the synthetic and real-world data from the very beginning. Navigational intent inference model was a joint project funded by Huawei in MARS lab. I also thank Xubo Lyu, Payam Nikdel, Mohammad Mahdavian, Geoff Nagy and all the other members of the Lab for having maintained a pleasant and studious atmosphere at the Research Lab during the month we spent together beside the remote work situation in the unusual context of the pandemic and all the additional challenges it brought.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

This chapter introduces the concern that this thesis investigates, as well as the problem's context, the study's objectives, and the report's structure.

1.1 Background

Exploration in an unfamiliar area by a mobile robot is a typical concern in robot applications. Normally, robots require full knowledge about obstacles and topological mapping of environments while using information from sensors [43, 44]. Machine learning has recently received a lot of recognition to tackle the problem of unavailability of the information. Robots that successfully manoeuvre in their space use a variety of sensors such as GPS, LiDARs, ultrasonic sensors, and sometimes a camera, as the primary means of tracking and localization. Some UAVs and wide-range robotics use measurement data to (i) localize the robot in the environment and (ii) identify specific environmental characteristics such as the location of obstacles or other objects that may be of concern. Consequently, the autonomous agent depends on the information it gathers to attempt to derive valuable information from the world that can be used to make decisions.

The aim of research on autonomous mobile robots is to both understand and construct systems that demonstrate intelligent behaviour. Obstacle avoidance is one of the most important functions of autonomous mobile robots. In order to respond appropriately in diverse, unfamiliar, and dynamic situations, the autonomous mobile robot must be able to adjust its skills. Reinforcement learning (RL) is a strong tool for achieving this goal since it does not require a full understanding of the world and allows the robot to learn online. Reinforcement learning is a concept borrowed from the study of animal learning and is analogous to control engineering's theory of learning automation. Reinforcement learning can be used to learn unknown target outcomes by providing autonomous mobile robots with appropriate performance evaluations, as well as to allow robots to learn online. Any robot learning algorithm running in evolving and unstructured environments where the robot explores its environment to obtain appropriate samples of the required experience should have online

learning and adaptation as desirable traits [24]. It is necessary to conduct online learning when interacting with the natural world in a dynamic environment. We would concentrate our efforts on learning how to avoid obstacles. If seen with an autonomous mobile robot, this fact is an example of a trait that can be reinforced.

When mobile robots work in the real world, they must be able to avoid obstacles, which is one of the most basic skills they need. Obstacle avoidance is a long-standing challenge in robotics that is commonly addressed using ranging sensors such as laser scanners and sonar [34]. However, ranging sensors only collect a small amount of data, and some of them are too costly or heavy/power-consuming for a specific platform, such as a UAV. Monocular cameras, on the other hand, offer details about the robot's working environment, are inexpensive, lightweight, and can be used on a variety of platforms.

In reality, some of the existing techniques for regulating and directing the robot to achieve full coverage are computationally intensive and susceptible to noise and inconsistencies in sensor data [9]. Controlling and preparing a trajectory often necessitates the discovery of an endless space of possible solutions. Machine Learning in general, and reinforcement Learning in particular, which encourages robots to be represented as software agents, has opened up new possibilities for improving robot control strategies. For reasonably complicated scenarios, motion preparation can be done, and suitable trajectories can be extracted in a fair amount of time.

1.2 Goal and novelty

In general, the goal of developing smart robots capable of acquiring and transferring knowledge and/or expertise gained from data analysis to better their navigation strategies and actions is the goal when designing autonomous and intelligent mobile systems. Robots must benefit not just from the analysis of sensor input data, but also from the effects of their behaviour in this sense [31].

The primary goal of this thesis is to discuss how information is perceived in the environment and how it can be applied to path planning tasks. To our knowledge, no previous work has addressed full probability density estimation of complex dynamic models in an online setting. Probabilistic relations are used to propose an inference paradigm for predicting human movement based on past actions.

1.3 Research methodology

Perception of the environment is critical for meticulous manipulation. Hazard awareness, constant obstacles, moving obstacles or other agents or humans, safety constraints, map, and so on are all examples of perception. This knowledge can necessitate sophisticated and diligent architectures in a complex setting. There are several ways to perceive information

from the environment, and one classification can be made between model-based and model-free methods. The issue is in the identification stage for model-based methods, where the aim is to determine the model parameters in order to estimate/predict the desired performance. In model-free methods, on the other hand, function approximate can be used to mimic the underlying model.

Model-based approaches, in particular, may be solved using either a classical optimization approach or probabilistic methods. Probabilistic methods have shown promise in an uncertain, dynamic world. The frequentest and Bayesian paradigms can be used to perform probabilistic approaches. Unlike frequents, Bayesian frameworks consider uncertainty in data rather than model parameters. This basic distinction enables researchers to have a degree of confidence in a model as a byproduct, paving the way for dealing with dynamic, hybrid, non-deterministic systems. We target complex model in this thesis by defining the full probability distribution of model parameters and inferring human navigational intent. One may use this information for path planning.

1.4 Outline

The remainder of this thesis is structured as follows: in chapter 2, we provide a quick rundown of the technological preliminaries that will be included in this thesis. We will discuss optimal control, Bayesian optimization, identification problems. We will present probabilistic estimation in Chapter 3, known as ARMCMC as an online method for rendering full probability density of model parameters in complex models. In chapter 4, we propose HNIPO as an inference tool for predicting human navigational intend. Finally, the thesis is concluded and future directions are presented in chapter 5. We show that these knowledge of current probabilistic information from the environment can be used in a near-optimal collision free path planing. One can use a hierarchical approach for path following, which combines global awareness from the world map with locally obtained information from the robot sensory data.

Chapter 2

Background

This chapter is devoted to background knowledge required for the methodology presented in chapter 3 and 4 of this thesis. The preliminaries besides the related work of literature are presented here.

2.1 Notations

Throughout this document, the set of real numbers is denoted by $\mathbb{R}$. Vectors are denoted by small letters whereas matrices are denoted by capital letters. The transpose of a matrix A is denoted by A^T. For $A \in \mathbb{R}^{n\times n}, A_{i,j}$ denotes the entry in row i and column j. The i^{th} component of a vector x is denoted by x_i. $\mathbb{E}\{\cdot\}$ represents the expectation of its argument.

2.2 Time-to-reach in optimal control

Optimal control theory seeks the best control policies for dynamical systems by optimizing user-defined cost functions that represent the user's design goals [35]. By solving a partial differential equation known as the Hamilton–Jacobi–Bellman (HJB) equation derived from dynamic programming (DP), a global optimality is obtained [49]. Traditional optimal control methods are employed offline and require full knowledge of the system dynamics, but they do not perform well when faced with uncertainty and changes in dynamics. Consider the following dynamical system in $\mathbb{R}^n$ with two competing inputs:

$$\dot{X} = f(X(t), u(t), d(t)), \tag{2.1}$$

where $X(\cdot)$ is the state and $u(\cdot), d(\cdot)$ are control and disturbance injected into system, respectively. The two competing inputs $u(\cdot)$ and $d(\cdot)$ are considered as controls for player I and player II, respectively. We assume that f is continuous and there exist $L > 0$ such that $\|f(x_1, a, b) - f(x_2, a, b)\| \leq L\|x_1 - x_2\|$ for all $x_1, x_2 \in R^n$. Under these assumptions, the dynamic system Eq. (2.1) has a unique solution [74].

The TTR [45] problem's objective is to find the shortest time for an agent to reach a target set of states from any initial state X by using optimal control, when player II wants to maximize the time, while player I uses a strategy $\theta(d)$ to minimize the time with the knowledge of player II's current and past decisions. So, the problem can be formulated as follows:

$$\phi(X) := \min_{\theta \in \mathcal{U}} \max_{d \in \mathcal{D}} T_X[d, \theta[d]], \tag{2.2}$$

where ϕ is a TTR value function, T is the time and X is the initial state. The TTR function (ϕ) can be obtained by solving the stationary Hamilton-Jacobi (HJ) partial differential equation (PDE) of the form Eq. (2.3).

$$H(X, \nabla\phi(X)) = \min_{d \in \mathcal{D}} \max_{u \in \mathcal{U}} \{-\nabla\phi(X)^T f(X, u, d) - 1\}, \tag{2.3}$$

where H denotes the Hamiltonian and f is the system dynamics. For detailed discussion on derivation of Eqs. (2.2), (2.3) one can refer to [74]. This will be used in Sec. 3.5 of this thesis.

2.3 Bayesian inference

Bayesian methods are effective tools for obtaining not only a numerical estimation but also a level of confidence in that estimate ([41, 8, 29]) by calculating the probability distribution of parameters rather than a point estimate, which is prevalent in frequentist paradigms [65]. One of the main advantages of probabilistic frameworks is that they enable decision making under uncertainty [52]. Furthermore, in probabilistic systems, information fusion is greatly facilitated; diverse sets of evidence or assumptions may be merged in a principled manner based on their degree of certainty [4]. Using a credible interval instead of confidence interval [41], absence of over parameterized phenomena [8], and evaluation in the presence of limited number of observed data [29] are other distinct feature of this framework. Nonetheless, Bayesian inference requires a significant amount of computation to estimate the entire probability distribution, as well as prior knowledge of the noise distribution.

For a time, the big hassle for deploying Bayesian methods in system identification was the deprivation of computational tools to solve the associated multi-dimensional optimization problem. Consequently, some restrictive assumptions were made to simplify the relations (e.g. calculating specific features of the model parameters' distribution rather than the whole distribution). One of the most effective methods for Bayesian inferences is Markov Chain Monte Carlo (MCMC). In the field of system identification, MCMC variants such as the one recently proposed by [22] are mostly focused on offline system identification. This is partially due to computational difficulties that preclude it from being used in real time [39]. Another source of difficulty exist in the case of model change which can happen in hybrid systems. Since different model candidates do not share the same parameter set, the

regular MCMC algorithm is unsuitable for model variability. The authors in [21] introduced reversible jump Markov chain Monte Carlo (RJMCMC) as a method to address the model selection problem. To solve dimension mismatch as a result of model variation, an extra pseudo-random variable is specified in this procedure. There are more MCMC variants in the literature (e.g. [20]), but there aren't any that are useful for online estimation.

2.3.1 Markov Chain Monte Carlo (MCMC)

Markov chain Monte Carlo (MCMC) techniques are a type of sampling procedure used in statistics. The posterior pdf can be sampled by recording states from a Markov chain with the desired distribution as its equilibrium distribution. The more steps there are, the closer the sample distribution is to the intended distribution. The samples are taken from a separate distribution called the proposal distribution, denoted $q(\cdot)$, which is easier to sample from than the posterior distribution. Author in [10] discusses several categories of MCMC implementations, including those that use various proposal distributions and approval conditions. This will be used in Sec. 3.2 of this thesis. The main steps of the Metropolis-Hastings algorithm are listed as follows [51]:

1. Set initial guess θ_0 such that $P(\theta_0|Y) > 0$ for iteration $k = 1$,
2. Draw candidate parameter θ_{cnd}, at iteration k, from the proposal distribution, $q(\theta_{cnd}|\theta_{k-1})$
3. Compute the acceptance probability,

$$\alpha(\theta_{cnd}|\theta_{k-1}) = \min\Big\{1, \frac{P(\theta_{cnd}|D)q(\theta_{k-1}|\theta_{cnd})}{P(\theta_{k-1}|D)q(\theta_{cnd}|\theta_{k-1})}\Big\}, \tag{2.4}$$

4. Generate a uniform random number γ in $[0, 1]$,
5. 'Accept' candidate if $\gamma \leq \alpha$ and 'ignore' it if $\gamma > \alpha$,
6. Set iteration to $k + 1$ and go to step 2.

2.3.2 Precision and reliability

Two important notions in probabilistic frameworks to compare results are precision (ϵ) and reliability (δ). The former represents the proximity of a sample to the ground truth, and the latter represents the probability that an accepted sample lies within ϵ of the ground truth. This will be used in Sec. 3.2.3 of this thesis.

Lemma: Let P_k be k samples from MCMC, and $\mathbb{E}(P_k)$ denote their expected value. According to Chernoff bound [63], to achieve the given precision and reliability given $\epsilon, \delta \in [0, 1]$, if the minimum number of samples (k) satisfies

$$k \geq \frac{1}{2\epsilon^2} \log(\frac{2}{1-\delta}), \tag{2.5}$$

then $Pr\Big\{\{P_k - \mathbb{E}(P_k)\} \leq \epsilon\Big\} \geq \delta$.

2.4 Filtering and prediction

Robotic motion filtering and force estimation are critical fields of research with fascinating problems that Bayesian inference can solve [57]. Measurements are inherently noisy in this situation, which is undesirable for monitoring. Correspondingly, inaccuracy, inaccessibility, and costs are common issues that make force measurements unusable for realistic applications [5]. For linear and Gaussian noise, various environmental detection approaches have been proposed in the literature [70]; however, in cases of nonlinear models like the Hunt-Crossley model that does not have Gaussian noise (e.g. impulsive disturbance), there is no optimal solution for the identification problem.

Human modelling is required for autonomous robots to navigate safely and smoothly in the presence of humans. In a variety of indoor and outdoor environments, knowing where people are likely to go in the next few seconds is critical for secure decision-making and autonomous navigation [38]. There have been extensive studies on robot trajectory planning [27], and human-aware planning is one of the most challenging problems in this area [38]. Forecasting future human states is one of the most important tasks in human-aware preparation. Maintaining a parameterized model of the system that involves complex entities such as humans and constantly modifying these parameters based on observations is one way to ensure stable and normal robot action [67].

2.4.1 Hunt-Crossley estimation

Authors in [17] suggested a two-stage bootstrapped approach for the Hunt-Crossley model's online recognition, which is vulnerable to parameter initial conditions. Authors in [12] proposed a method to determine the damping term in the Hunt-Crossley model. A neural network-based approach was introduced to control the contact/non-contact Hunt-Crossley model in [7]. A single-stage method for estimation of the Hunt-Crossley model is proposed by [23]; in this case, some restrictive conditions must be met to calculate the parameters. Furthermore, the system fails to provide a solution when the dynamic model undergoes discontinuous shifts, which is normal during the transition from contact to free motion and vice versa [73]. This will be used in Sec. 3.3.2 of this thesis.

2.4.2 Human navigational prediction

Methods for predicting human trajectory can be classified into many groups based on the input modalities available, the modelling technique used, and the optimal type of prediction, which varies depending on the application scope [56].

Deep learning-based methods are the subject of a lot of recent research. Some make observations based on contextual/dynamical clues or social interactions in the scene using a

series of scene frames. Authors in [6] used visionary data and recurrent neural networks to create a human trajectory predictor that can capture the social interactions between humans in the scene. A constant-velocity motion for humans is considered in [33] which forecasts trajectories of occluded pedestrians in crowded scenes. Authors in [11] propose a three-stage network. They predict a series of possible 2D targets in the image space while taking into account the scene background and create a sequence of 3D human poses that follow paths to the goals. To forecast dynamically-feasible agent trajectories, [58] integrates environmental dynamics as scene graph representations and semantic maps in a novel graph-based recurrent network. To encode the past of human motion, [16] proposes using a generative adversarial network (GAN). They create predicted trajectories by predicting goals in collaboration with a soft attention network for scoring. Some of the works ventured into the field of multi-modality. [75] combined human head and body pose modalities with a gesture function to forecast the end-point and direction in the real world using a hybrid CNN-LSTM network. Some studies focused on the end-point rather than the whole trajectory. [47] and [46] propose a novel pooling strategy called social-pooling to incorporate historical backgrounds of all pedestrians to predict socially compliant long trajectories using a conditional variational autoencoder technique. Using a socially aware LSTM model, authors in [13] show how the path and distance of moving agents influences the trajectory prediction of a targeted pedestrian. Although many of the studies in this field rely on visual stimuli, some have attempted to learn probabilistic models based on human observational evidence. Given the target and the proper objective function, proposes a maximum entropy-based probabilistic model for humans and adds a conviction parameter as a model confidence measure that is modified in real time in a Bayesian paradigm [18]. Authors in [37] and [68] focused on Model Predictive Control (MPC) based on Inverse Reinforcement Learning (IRL) to model human behaviour. It is possible to account for the human motion uncertainties by creating a temporal social cost map dependent on Gaussian distributions around humans [36]. [67] incorporates a social force model [26] with a velocity-based approach and adaptive parameters to tackle this problem.

While several studies in the literature have shown encouraging results in human trajectory simulation, there are no comparable studies on online projections using Optimal Control and Bayesian filtering that we are aware of. As a result, this paper represents the first step in that direction.

2.4.3 Human motion dynamic

We use a 4D extended Dubins car model to describe human motion:

$$\begin{aligned} \dot{x} =& v\cos(\theta t) \\ \dot{y} =& v\sin(\theta t) \\ \dot{\theta} =& \omega \\ \dot{v} =& a \end{aligned} \tag{2.6}$$

where the human state $X_H = (x, y, \theta, v)$ consists of position, heading (direction of travel), and speed. The human actions, $u_H = (\omega, a)$ are the turn rate and acceleration. This will be used in Sec. 3.5 of this thesis.

2.5 Collision avoidance path planning

One approach for dealing with the challenge of a complex world is to schedule an initial path and then re-plan once its implementation becomes infeasible [14]. Owing to the lack of motion knowledge of dynamic obstacles, re-planning may suffer from time inefficiency (frequent re-planning) and path inefficiency (oscillating motions and detours). If the possible motions of obstacles are unpredictable, one may try to model their actions and simulate their path using dynamic obstacles [54]. Separating the navigation problem through disjointed prediction and preparation phases, on the other hand, may result in the "freezing robot" problem. This happens when the environment becomes too complicated, the planner decides that all forward paths are unsafe, and the robot freezes (or makes useless turns) to avoid collisions [66]. This behaviour is suboptimal, because a feasible path is usually available. In that case, the robot would be unable to take any action because the projected paths will render a significant portion of the space impassable.

Learning-based approaches have been recently receiving intensive interest to target dynamic environments [50]. However, these methods are hampered by the following flaws, which limit their contribution to path planning.

1. Sparse reward in the case of extremely large environment
2. Increased training effort and inefficient learning method [61]
3. Over-fitting, since robots are frequently restricted to training environments and have low generalizability in unfamiliar environments [53]
4. Scalability, the extension to a multi-robot framework is implausible due to the complicated architecture and increasingly growing observation space [60].

In a two-step approach, first some researchers use sensor data to create a map of environment using techniques like Simultaneous Localization and Mapping (SLAM), and then

use path-planning algorithms like RRT^* to navigate certain spaces while avoiding obstacles. However, path-planning on-the-fly with SLAM can be challenging as these methods struggle to automatically respond to non-stationary conditions [30]. In autonomous obstacle avoidance, where LiDAR sensor is available, using Q-learning and/or SARSA on ϵ-greedy policy with Linear Value Function Approximation (LVFA) for obstacle avoidance has been suggested [42, 64].

Chapter 3

Probabilistic parameter estimation

A probabilistic approaches for model estimation in an uncertain environment is discussed in this chapter. In order to obtain full probability distribution of model parameters in a Bayesian framework, an online version of MCMC is introduced. This is done by proposing model mismatch index and variable jump policy.

3.1 Full probability density estimation

Bayesian methods offer a robust mechanism for estimating the full probability distribution over uncertain parameters, but due to high computational costs, its online application can be difficult. We propose **A**daptive **R**ecursive **M**arkov **C**hain **M**onte **C**arlo (ARMCMC) which estimates full probability densities of model parameters while alleviating the shortcomings of conventional online approaches. These shortcomings include being limited to considering Gaussian noise, being applicable to systems with linear in the parameters (LIP) constraints, and having persistence excitation requirements (PE). We propose a variable jump distribution in ARMCMC which is based on a temporal forgetting factor. This helps one to vary the trade-off between exploitation and exploration based on whether the parameter being estimated changes abruptly. In comparison to traditional MCMC methods, we show that ARMCMC needs fewer samples to achieve the same precision and reliability. We demonstrate our method on two difficult benchmarks: parameter estimation in a soft bending actuator, and the Hunt-Crossley dynamic model, which is a well-known model for soft contact force. As compared to recursive least squares and classical MCMC, our approach achieves a 70% increase in parameter point estimation precision, and a 55% reduction in tracking error of the value of interest.

3.1.1 Objectives

We propose ARMCMC, a method that addresses many flaws in conventional online estimation approaches, such as being only applicable to linear in parameters (LIP) model,

requiring persistent excitation (PE), and assuming Gaussian noise. By the cost of computation, Bayesian optimization systems encompass a broader group of models.

Whenever the parameter distribution does not change dramatically, ARMCMC is an online approach that takes advantage of the previous posterior distribution. To achieve this, we define a new *variable jump distribution* that accounts for the degree of model mismatch using a *temporal forgetting factor*. The temporal forgetting factor is computed from a model mismatch index and determines whether ARMCMC employs modification or reinforcement to either restart or refine the estimated parameter distribution. As this factor is a function of the observed data rather than a simple user-defined constant, it can effectively adapt to the underlying dynamics of the system. We demonstrate our method using two different examples: a soft bending actuator, and the Hunt-Crossley model. We show favorable performance compared to state-of-the-art baselines.

3.1.2 Problem statement

In the Bayesian paradigm, estimates of parameters are given in the form of the posterior probability density function (pdf); this pdf can be continuously updated as new data points are received. Consider the following general model:

$$Y = F(X(t), \theta(t)) + \nu, \tag{3.1}$$

where Y, X, θ, and ν are concurrent output, input, model parameters and noise vector, respectively. To calculate the posterior pdf of θ, the observed data (input/output pairs) along with a prior distribution are combined via Bayes' rule [32]. We will be applying updates to the posterior pdf using batches of data points; hence, it will be convenient to partition the time series data as follows:

$$D^t = \{(X,Y)_{t_m}, (X,Y)_{t_m+1}, \cdots, (X,Y)_{t_m+N_s+1}\}, \tag{3.2}$$

where $N_s = T_s/T$ is the number of data points in each data pack with T, T_s being the data and algorithm sampling time periods, respectively. This partitioning is useful for online applications because D^{t-1} should have been obtained previously and the algorithm runs from t_m to $t_m + N_s + 1$, which we will call the algorithm time interval t. Ultimately, inference is completed at $t_m + N_s + 2$. Fig. 3.1 illustrates the timeline for the data and the algorithm. The algorithm is mainly composed of three consecutive steps; data acquisition (phase A), training (phase B) and, execution (phase C). It is worth noting that each interval does not necessarily have to start after the previous run is done. More specifically, it will run in parallel with the earlier phase of next algorithm interval (e.g. phase (A) of algorithm t, phase (B) of algorithm $t-1$, and phase (C) of algorithm $t-2$ will all run simultaneously). According to Bayes' rule and assuming data points are independent and identically distributed (i.i.d)

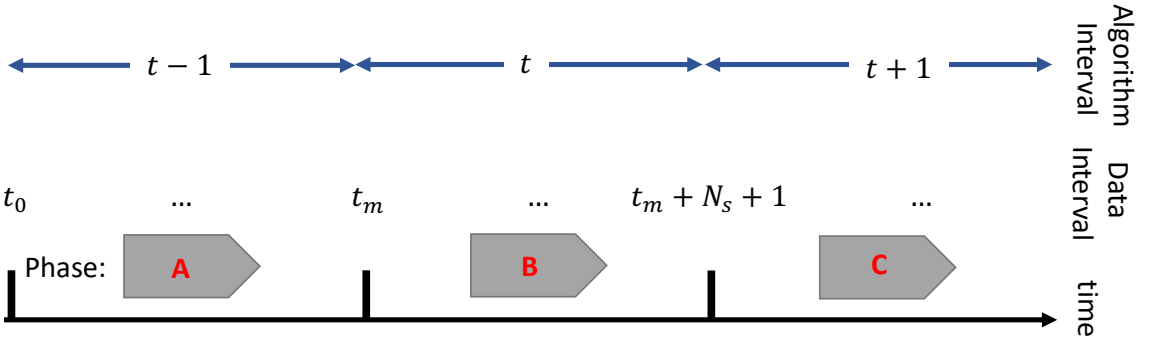

Figure 3.1: Data timeline and different phases of ARMCMC algorithm. For the algorithm at time t: Phase (A) data collection (pack N_s data points), phase (B) running (apply the method on the data pack), phase (C) execution (update the algorithm results on parameters).

in Eq. (3.1), we have

$$P(\theta^t|[D^{t-1}, D^t]) = \frac{P(D^t|\theta^t, D^{t-1})P(\theta^t|D^{t-1})}{\int P(D^t|\theta^t, D^{t-1})P(\theta^t|D^{t-1})d\theta^t}, \tag{3.3}$$

where θ^t denotes the parameters at the current algorithm time step. $P(\theta^t|D^{t-1})$ is the prior distribution over parameters, which is also the posterior distribution at the previous algorithm time step. $P(D^t|\theta^t, D^{t-1})$ is the likelihood function which is obtained by sampling from the one-step-ahead prediction:

$$\hat{Y}^{t|t-1} = F(D^{t-1}, \theta^t), \tag{3.4}$$

where $\hat{Y}^{t|t-1}$ is the prediction of the output in (3.1). If the model in (3.4) is accurate, then the difference between the real output and predicted should be measurement noise, (i.e., $Y^{t|t-1} - \hat{Y}^{t|t-1} = \nu$). Therefore, the model parameter may be updated as follows:

$$P(D^t|\theta^t, D^{t-1}) = \prod_{t=t_m+1}^{t_m+N_s+1} P_\nu(Y^{t|t-1} - \hat{Y}^{t|t-1}), \tag{3.5}$$

where P_ν is the probability distribution of noise. Note that there is no restriction on the type of noise probability distribution.

Remark 1: To avoid numerical issues, the summation of probability logarithms are calculated. In addition, each data pair in the algorithm time sample is weighted based on its temporal distance to the current time. Therefore, Eq. (3.5) is modified as

$$\begin{aligned} \log\left(P(\cdot)\right) &= \sum_{t_m+1}^{t_m+N_s+1} \log P_\nu(e^t), \\ e^t &= \Big(Y_n^t - F^t(D^{t-1}(n), \theta^t)\Big) e^{-\rho(N_s-n)}, \end{aligned} \tag{3.6}$$

where $\rho \in [0, 1]$ is a design parameter that reflects the volatility of the model parameters, and $e^t = [e_1^t, ..., e_n^t, ..., e_{N_s}^t]$. For systems with fast-paced parameters, ρ should take larger values.

3.2 ARMCMC algorithm

ARMCMC calculates the posterior pdf by drawing samples recursively at each algorithm time interval. The desired precision and reliability, as well as the requirement for real-time applications, restrict the number of samples drawn. The maximum number of data points in each data pack, Ns, is limited by how quickly model parameters vary over time, while the minimum is constrained by the shortest required time for the algorithm to be real-time.

We propose a variable jump distribution that allows enrichment as well as exploration. To represent the current underlying dynamics of the data, we define a temporal forgetting factor as a model mismatch measure. We also show that, as compared to conventional MCMC, ARMCMC achieves the same precision and reliability with fewer samples. ARMCMC is summarised in Algorithm 1.

3.2.1 Variable jump distribution

We propose a variable jump distribution (also known as a proposal distribution) to achieve faster convergence, thereby enabling real-time parameter estimation:

$$q_k^t(\theta^t|\theta_{k-1}^t) = \begin{cases} P(\theta^{t-1}|D^{t-1}) & \lambda_k \leq \lambda^t \\ N(\mu_D, \sigma_\nu) & \lambda_k > \lambda^t \end{cases}, \tag{3.7}$$

where θ_{k-1}^t is the $(k-1)$-th parameter sample, which is given by the t-th data pack throughout the MCMC evaluation. In each algorithm time sample, we divide the data pack in two halves, and θ^t is equal to the average of the second half. $P(\theta^{t-1}|D^{t-1})$ is the posterior pdf of the parameters at the previous algorithm time step, and $N(\mu_D, \sigma_\nu)$ is a Gaussian distribution with its mean and variance μ_D, σ_ν computed using the empirical mean and variance of D^{t-1}.

The adaptive threshold for the λ^t (*temporal forgetting factor*), is based on classical system identification and regulates how previous experience affects the posterior pdf. A lower value of λ^t implies that there may be a sudden change in the ground truth value of θ, necessitating further investigation. When θ is changing slowly, however, a greater value of λ^t

Algorithm 1 ARMCMC

Assumptions: 1) roughly noise mean (μ_ν) 2) roughly noise variance (σ_ν) 3) desired precision and reliability (ϵ_0, δ_0) 4) desired threshold for model mismatches (ζ_{th})
Goal: Online calculation of parameters posterior distribution given the consecutive t-th pack of data ($P(\theta^t|D^t)$)
Initialization: Prior knowledge for θ_1^0, $n = 0$
Consider desire precision and reliability (ϵ, δ)
repeat
  Put $t_0 = n * N_s + 1$ from (3.2), $n++$
  Add a new data pack to dataset D^t
  Model mismatch index: ζ^t from (3.9)
  if $\zeta^t < \zeta_{th}$ **then**
    Reinforcement: set prior knowledge equal to the latest posterior of previous pack
    Temporal forgetting factor: λ^t from (3.8)
  else
    Modification: set prior knowledge θ_1^n
    Temporal forgetting factor: $\lambda^t = 0$
  end if
  Set minimum iteration k_{min} from (3.10)
  for $k = 1$ **to** k_{min} **do**
    Proposal distribution:
      • draw $\lambda_k \sim U(0, 1)$
      • *Variable jump distribution:* $q_k^t(.)$ from (3.7)
    Draw $\theta_k^{t*} \sim q_k^t(.)$
    Acceptance rate: $\alpha(.)$ from (2.4)
    Draw $\gamma \sim U(0, 1)$
    if $\gamma \leq \alpha$ **then**
      'Accept' the proposal
    end if
  end for
  Wait to build $D_{t_0}^{t_{m+N_s+1}}$ (algorithm sample time)
until No data is obtained

is sufficient, allowing previous information to be exploited. Better precision and reliability would result from using this information.

3.2.2 Temporal forgetting factor

Depending on whether the distribution of the parameter θ has changed significantly, a new sample is drawn according to the *modification* or the *reinforcement* mode in ARMCMC. When the identified parameter probability distribution is not changing rapidly, reinforcement is used to make it more accurate. Otherwise, modification is used to re-identify the distribution from the ground up. As a result, we define a *model mismatch index*, denoted ζ^t, so that when it exceeds a predefined threshold ζ_{th}, modification is applied. Otherwise,

ζ^t is used to determine λ^t as follows:

$$\lambda^t = e^{-|\mu_\nu - \zeta^t|}, \tag{3.8}$$

where μ_ν is an estimation of the noise mean, which can be calculated from the expected value in Eq. (3.1). Note that employing modification is equivalent to setting $\lambda^t = 0$. The model mismatch index ζ^t itself is calculated by averaging the errors of the previous model given the current data:

$$\zeta^t = \frac{1}{N_s} \sum_{n=1}^{N_s} \Big(y_n^t - \mathop{\mathbb{E}}_{\theta \sim \theta^{t-1}} (F(D^t(n), \theta)) \Big), \zeta^0 = \infty \tag{3.9}$$

By taking into account the usefulness of previous information in disposal, the temporal forgetting aspect makes it possible to handle the retrieval of multiple models. Intuitively, receiving more data pairs will make the parameters' distribution less dispersed if the system dynamic does not adjust over time. As a result, previous posterior distributions are still approximately accurate, and minor adjustments are sufficient. Modification, on the other hand, captures a sudden shift in model parameters or model alternation. The algorithm will be able to investigate new spaces as a result of this.

Remark 2: The model mismatch index accounts for all sources of uncertainty in the system. To determine a reasonable model mismatch threshold ζ_{th}, one needs to precalculate the persisting error between the predicted and measured data. In other words, ζ_{th} is basically an upper bound for the unmodeled dynamics, disturbances, noises, and any other source of uncertainty in the system.

3.2.3 Minimum required evaluation

Theorem 3.2.1. *Let ϵ and δ be the desired precision and reliability. Furthermore, assume that the number of evaluations (k) for the initial data pack at first algorithm time step satisfies Eq. (2.5). To satisfy this for the rest of the data, the minimum number of samples in ARMCMC is given by:*

$$k_{min} = \frac{1}{2\epsilon^2} \log\left(\frac{2}{\lambda^t(1-\delta) + 2(1-\lambda^t)e^{-2\epsilon^2(1-\lambda^t)k_{min}}}\right). \tag{3.10}$$

Proof. Samples from previous pdf: According to the variable jump distribution in (3.7), given k samples, the expected number of samples drawn from the previous posterior pdf $(P(\theta|D^t))$ is $\lambda^t k$. By assumption, the algorithm has already drawn at least k samples in the previous algorithm time-step. Consequently, by (2.5), the expected number of samples with distances less than ϵ from $\mathbb{E}(P_k)$ drawn from a previous distribution is at least $\lambda k \delta$.

Samples from Gaussian: By (3.7), there are $k_0 = (1-\lambda^t)k$ samples drawn in expectation. According to (3.11), we have $P\Big\{\{P_k - \mathbb{E}(P_k)\} \leq \epsilon\Big\} \geq \delta_0$, where δ_0 is given by rearranging (2.5):

$$\delta_0 = 1 - 2e^{-2\epsilon^2 k_0}. \tag{3.11}$$

Thus, the expected number of samples with distances less than ϵ from $\mathbb{E}(P_k)$ are at least $\delta_0(1-\lambda^t)k$.

Overall reliability: The total expected number of samples with distances less than ϵ from $\mathbb{E}(P_k)$ is the sum of the two parts mentioned above. Hence, it is obtained through dividing by k:

$$\delta_1 = \frac{(\lambda^t k\delta) + (\delta_0(1-\lambda^t)k)}{k} \tag{3.12}$$

Given the new obtained reliability, which is greater than the desired one, helps us decrease the number of evaluations. For the sake of illustration, Fig. 3.2 presents the minimum required number of evaluations with respect to λ for different precisions and reliabilities. As it can be seen, the MCMC is equal to ARMCM if λ is always set to one. The number of evaluations in ARMCMC mitigates as the validity of the previous model increases. □

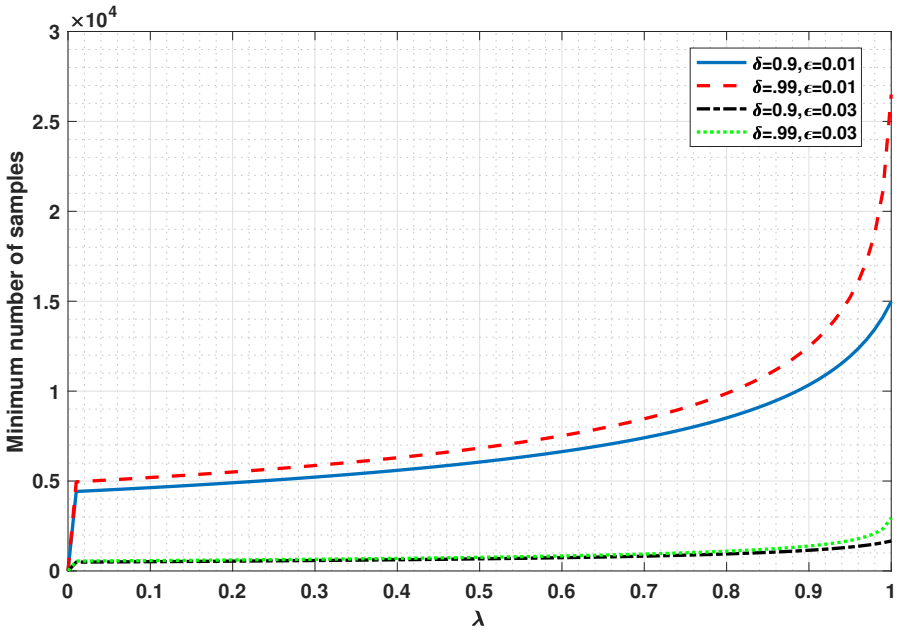

Figure 3.2: K_{min} with respect to λ for some values of ϵ, δ in ARMCMC. (for $\lambda = 1$ evaluation for ARMCMC is equivalent to MCMC)

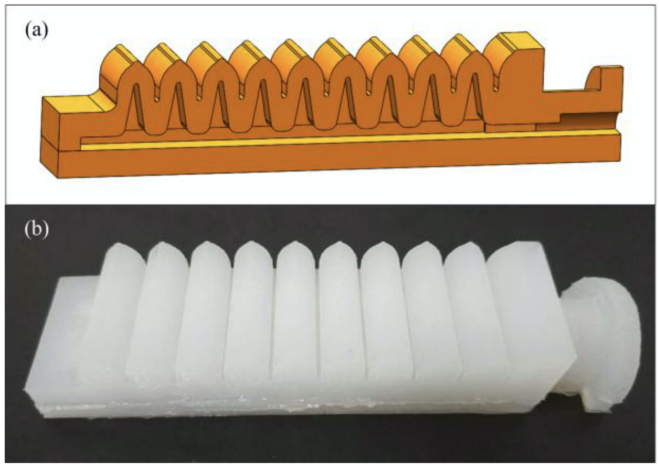

Figure 3.3: Studied fluidic soft bending actuator. (a) Cut view of the three- dimensional model. (b) Prototype picture [72].

3.3 Results

In this section, we demonstrate the performance of the proposed approach on two examples. First, we employ the proposed method to identify parameters in the soft bending actuator model and compare the results with identification using Recursive Least Squares (RLS). In the second example, we evaluate our method on the Hunt-Crossley model given a reality-based model and compare it with simple MCMC and RLS.

3.3.1 Fluid soft bending actuator

Consider the dynamic model of a fluid soft bending actuator in Fig. 3.3, given by [72]:

$$\begin{aligned} \ddot{\alpha} &= q_1(p - p_{atm}) - q_2\dot{\alpha} - q_3\alpha, \\ u_c \,\mathrm{sign}(p_s - p)\sqrt{|p_s - p|} &= q_4\dot{p} + q_5\dot{p}p, u_d = 0, \\ u_d \,\mathrm{sign}(p - p_{atm})\sqrt{|p - p_{atm}|} &= q_6\dot{p} + q_7\dot{p}p, u_c = 0, \end{aligned} \tag{3.13}$$

where α is the angle of the actuator, and u_c, u_d are the control inputs for retraction and contraction, respectively. Also p, p_s, p_{atm} are the current, compressor, and atmosphere pressure respectively. For this example, we assume $q_1 = 1408.50, q_2 = 132.28, q_3 = 3319.40$ are known and $p_{atm} = 101.3$ kPa, $p_s = 800$ kPa. We are trying to identify the four other parameters $(q_4, ..., q_7)$. The true parameters are $q_5 = -2.14 \times 10^{-4}, q_6 = 6.12 \times 10^{-9}, q_7 = -9.76 \times 10^{-5}, q_8 = -1.90 \times 10^{-9}$, which are used to generate simulated data. To this end, we assume the hybrid model below:

$$u \;\; \mathrm{sign}(\Delta p)\sqrt{|\Delta p|} = \theta_1\dot{p} + \theta_2\dot{p}p, u = \{u_c, u_d\}, \tag{3.14}$$

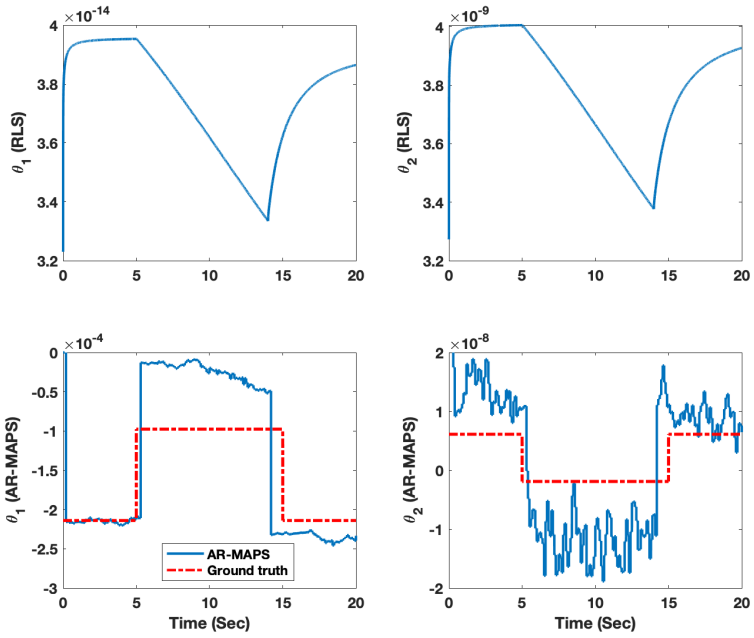

Figure 3.4: Parameter variation for RLS and AR-MAPS

where Δp is either $(p_s - p)$ for retraction or $(p - p_{atm})$ for contraction. As the range of parameters are small, we scale the input vector by a factor of 10^7 for RLS. Given the input (u_c, u_d) and the output $(p, \dot{p})$, we want to identify the parameter and estimate the current angle of actuator assuming that its initial position is at the origin. The data sample time is $T = 1$ ms and each data pack includes 100 samples, which results in an algorithm sample time equal to $T_s = 0.1$ sec. Point estimation is obtained by considering the mode at the modification phase and the median during the reinforcement phase; this estimate is denoted as AR-MAPS. The point estimate results for the parameter estimation are shown in Fig. 3.4. The estimation errors are $0.0235, 6.0053 \times 10^{-7}$ for θ_1, θ_2 in RLS and $0.0089, 1.1840 \times 10^{-7}$ in AR-MAPS, respectively. Moreover, the estimation of the angle is plotted in Fig. 3.5. As it can be seen, the AR-MAPS estimation is closer to the ground truth compared to RLS.

3.3.2 Hunt-Crossley model

In this section, we demonstrate ARMCMC by identifying parameters of the Hunt-Crossley model, which represents an environment involving a needle contacting soft material. The needle is mounted as an end-effector on a high-precision robotic arm, which switches between two modes: free motion and contact. For robotic applications, estimating the exact force on the needle's tip is vital, but direct measurement of this force is difficult due to the cost and noise of force sensors. Owing to sterilisation problems, installing force sensors on the needle is normally not possible in medical applications. Due to abrupt changes in the

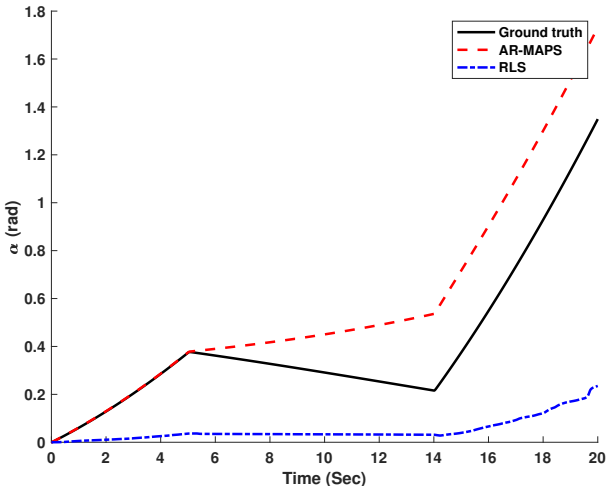

Figure 3.5: Angle of the actuator comparison of RLS and AR-MAPS for soft bending actuator.

model parameters when the contact is established or lost, online estimation of the force is extremely challenging.

Contact dynamic model

Consider the dynamics of contact as described by the Hunt-Crossley model, which is more consistent with the physics of contact than classical linear models such as Kelvin-Voigt [23]. In order to overcome the shortcomings of linear models, [28] proposed the following hybrid nonlinear model:

$$f_e(x(t)) = \begin{cases} K_e x^p(t) + B_e x^p(t)\dot{x}(t) & x(t) \geq 0 \\ 0 & x(t) < 0 \end{cases}, \tag{3.15}$$

in which, $K_e, B_e x^p$ denote the nonlinear elastic and viscous force coefficients, respectively. The parameter p is typically between 1 and 2, depending on the material and the geometric properties of contact. Also, $x(t), \dot{x}(t), f_e$ are the current position, velocity (as input X) and contact force (as output (Y in Eq. (3.1)) of a needle near or inside the soft material, with $x \geq 0$ representing the needle being inside. This needle can move freely in open space or penetrate the soft contact; the forces on this needle are modeled using the Hunt-Crossley model. The practical problem we consider is to estimate the force at the tip of the needle by identifying the model parameters. K_e, B_e, p are three unknown parameters (θ in Eq. (3.1)) that needs to be estimated. An online estimate of contact force plays a pivotal role in the stable interaction between robotic manipulators and unknown environments.

$$\begin{aligned} \log(f_e) &= \log(K_e x_s^p + B_e \dot{x}_s x_s^p), \\ \log(f_e) &= p \log(x_s) + \log(K_e + B_e \dot{x}_s). \end{aligned} \tag{3.16}$$

For RLS, we also need to make the assumption that $B_e/K_e\dot{x}_s << 1$. Note that the vector of parameters (θ) in the following relation are not independent, which may lead to divergence. With this assumption, we have

$$\begin{aligned} \log(1 + B_e/K_e\dot{x}_s) &\approx B_e/K_e\dot{x}_s, \\ \log(f_e) &= p\log(x_s) + \log(K_e) + B_e/K_e\dot{x}_s. \end{aligned} \tag{3.17}$$

$$\begin{aligned} \phi &= [1, \dot{x}_s, \log(x_s)], \\ \theta &= [\log(K_e), B_e/K_e, p]^T. \end{aligned} \tag{3.18}$$

Setup

The data structure is same as previous simulation. Prior distribution of all three parameters (K_e, B_e, p) are initialized to $N(1, 0.1)$ (a normal distribution with $\mu = 1$ and $\sigma = 0.1$) Furthermore, as more data is obtained, the posterior pdf's distribution narrows. After about 5 seconds, the needle exits the soft material and feels zero force; this is the same as setting all parameters to zero. For the three parameters in Fig. 3.7, a color-based visualization of probability distribution over time is used. There is no interaction and the parameter values are equal to zero during the period when the entire space is blue (zero probability density).

Since we are taking a Bayesian approach, we are able to estimate the entire posterior pdf. However, for the sake of illustration, the point estimates are computed from the ARMCMC algorithm by using AR-MAPS method. The results are shown in Fig. 3.8 for the time-varying parameters $\theta_1 = K_e, \theta_2 = B_e, \theta_3 = p$. During the times that RLS results are chattering due to the use of saturation (if not, the results would have diverged), the needle is transitioning from being inside the soft material to the outside or vice versa. In addition, due to the assumption (3.17), performance can deteriorate even when there is no mode transition. Furthermore, in the RLS approach, estimated parameters suddenly diverged during free motion, since the regression vectors are linearly dependent. In contrast, with ARMCMC this is not an issue we need to consider. The result of ARMCMC is presented in Fig. 3.6, which shows the force estimation with two different identification approaches. This probability of interest can be easily obtained by deriving the parameter density at one's disposal.

Quantitative Comparison

Quantitative details of comparing a naive point estimate of the ARMCMC algorithm by averaging the particles (denoted as AR-APS) and the RLS method are listed in Table 3.1. This reveals more than a 70% improvement in the precision of all model parameters throughout the study by using the Mean Absolute Error (MAE) criteria and also more than a 55% improvement in the force estimation error. Amongst parameters, the viscose (B_e) has the largest error in the RLS method since it is underestimated due to the restrictive

Table 3.1: Comparison of RLS [23] and point estimate of ARMCMC and MCMC for environment identifications.

Errors (MAE)	K_e	B_e	p	F_e (mN)
RLS	0.5793	0.9642	0.3124	51.745
MCMC-1	0.6846	0.8392	0.3783	76.695
MCMC-2	0.7294	0.9964	0.4195	101.88
AR-APS	0.0774	0.0347	0.0945	33.774
AR-MAPS	0.0617	0.0316	0.0756	31.659

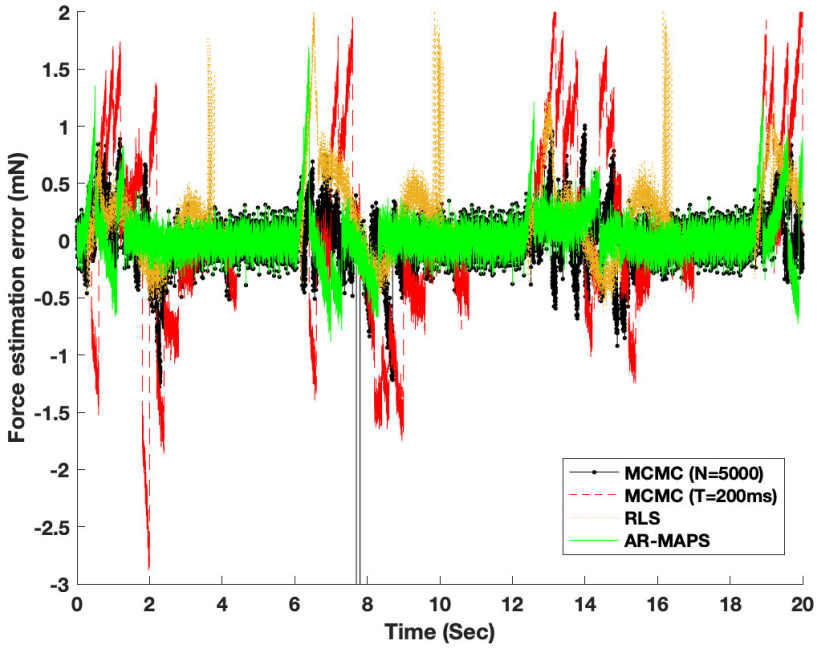

Figure 3.6: Force prediction error in RLS, AR-APS, and MCMC.

assumption in Eq. (3.17). The AR-MAPS approach uplifts the performance of the parameter identification and the force estimation. We also compare ARMCMC to MCMC. For the algorithm to run in real-time, MCMC requires more time to converge. For this example, with $\lambda = 0.7$, the value of k_{min} is 15000 for MCMC but only 6000 for ARMCMC with $\epsilon = 0.01, \delta = 0.9$. Two possible ways to address this drawback in MCMC are to reduce the number of samples to 5000 per algorithm iteration (denoted MCMC-1 in Table 3.1), which results in worse precision and reliability compared to ARMCMC, or to increase the algorithm sample time to 0.2 (denoted MCMC-2 in Table 3.1) which would cause more delay in the estimation result and slower responses to changes in the parameter.

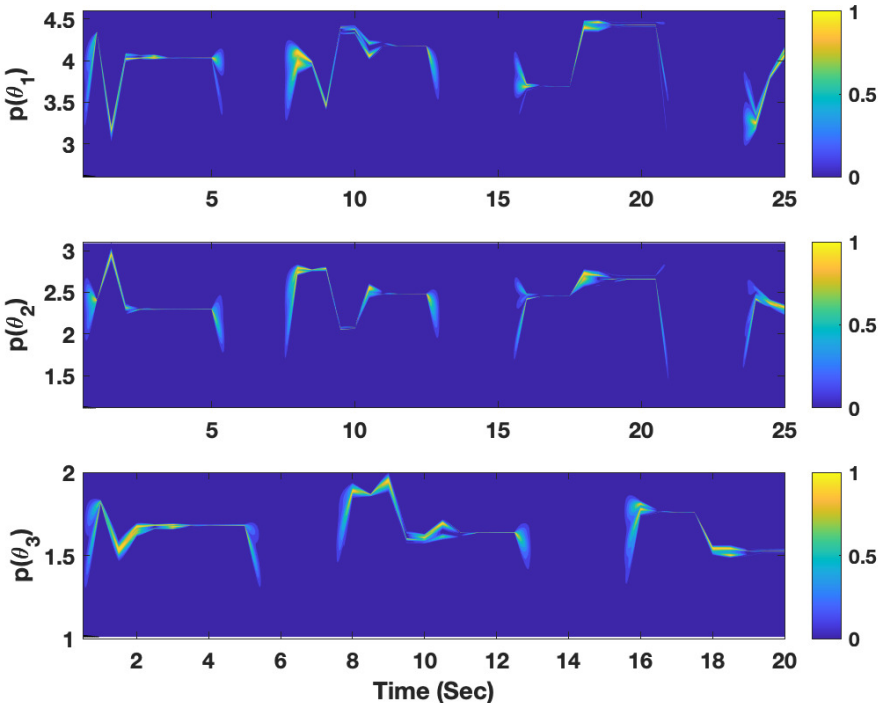

Figure 3.7: Probability distribution of parameters ($\theta_1 = K_e, \theta_2 = B_e, \theta_3 = p$) using ARM-CMC.

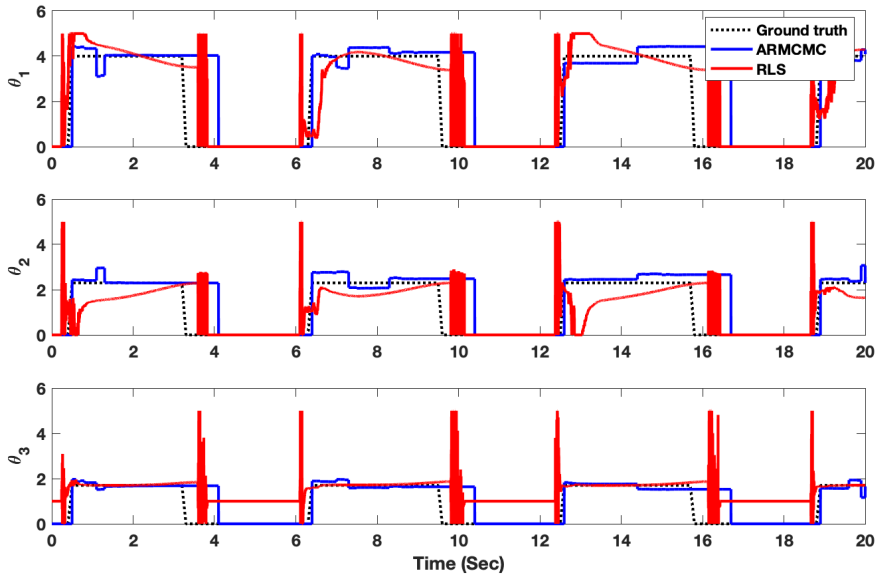

Figure 3.8: Model parameters ($\theta_1 = K_e, \theta_2 = B_e, \theta_3 = p$) point estimation in AR-MAPS.

Chapter 4

Human navigational intent inference

In this chapter, an inference framework to predict the probability distribution over future human states. The navigational model is parametrized by three internal human parameters: behaviour optimality, farsightedness, and navigational goal. We estimate the probability distribution of these parameters and predict future human states simultaneously.

In contrast to previous work [19], which used a simple two-dimensional model of human dynamics, our model accounts for the effects of human orientation and speed on future human states. However, the more realistic human dynamic model introduces two additional challenges. First, online prediction of four-dimensional future states is intractable using a simple Bayes' filter as was done in [19]. Second, the human navigational cost function, assumed to be time in [18] and our work, no longer leads to a simple, analytical policy; instead, the human policy now needs to be derived based on a value function that is also intractable to compute in real time.

To address the first challenge stemming from the more complex human dynamic model, we use a particle representation of the probability distributions over human states and over the human navigational goal parameter, and estimate the distributions using an adapted particle filter. In addition to resolving the computational challenge, the use of particles also enables our approach to estimate arbitrary goal distributions, rather than assuming a set of fixed potential goal positions as was done in [19].

To address the second challenge, we first pre-compute a time-to-reach TTR value function, assuming the goal is known and at the origin. Leveraging the position and orientation invariance of the extended Dubins car dynamics, we can derive in real time the TTR function for goals at arbitrary positions and orientations via simple translation and rotation. Using properties of geometric series, this TTR function can then be converted into an action-value (Q) function, which allows us to introduce the farsightedness parameter, and using which the noisily rational policy can be computed.

In summary, our contributions are as follows: Owing to human unpredictability, robot navigation and decision-making in the presence of humans is complex. In the literature, various probabilistic or hybrid models for capturing human navigational purposes have been presented. With a prediction of the intent, one may forecast future human states in order to avoid collisions. In this section, we create a general framework for predicting human navigational states in the future based on their current location, heading, velocity, and action. To this end, we employ a probabilistic model that incorporates three parameters including: a) their conceptual point of interest, referred to as their goal; b) the human optimality factor, which shows how optimal the human's actions are in relation to their goal; and c) the farsightedness parameter, which is inspired by the discount factor in reinforcement learning and prioritize between current and future gains. We assume a pre-computed time-to-reach function as a reward function and 4D Dubins car model as human motion dynamic. The model parameters are refined to represent the credibility of the individual hypotheses during the update stage, when we observe an action taken by the person. Due to our aforementioned assumptions, the computation is expensive. Hence, we use particle filters to describe goals and human states to make the computation tractable. We compare our findings to those of a recent related methodology using synthetic data to show the efficacy of our method. We also put our approach to the test in the real world.

4.0.1 Overview

In this thesis, we present a probabilistic model of human navigation that can be used to forecast possible human states in real time. This is done by assuming a human motion model and constantly estimating three internal human parameters. The first internal parameter is a person's possible navigational purpose, or the point of interest they hope to visit in the future. Various known and unknown factors, such as mental state, social pressures, and complex environmental restrictions, all influence such motives. The second internal parameter is the degree of optimality; the higher the value, the more efficient the human's strategy for achieving its target. A farsightedness metric is used to measure how often current gains are sacrificed in order to obtain future gains. This method proposes a few possible human goals and formulates a prediction based on that hypothesis; when the human takes action, the model evaluates the reliability of each goal assumed previously and attempts to adjust the hypothetical goals.

To make predictions more reliable, we use a Bayesian approach, updating beliefs over model parameters, as described in [18]. We use the 4D dynamical model presented in Eq. (2.6) in order to capture human motions more accurately. Prior cognitive science research has shown that a person can be viewed as an agent acting on the basis of a utility function [48]. Thus, we incorporate a noisy-rationality model of the human, which determines a distribution over the possible human actions given the utility function. We use a discounted TTR function, parameterized by the goal and farsightedness parameter, as the

utility function. Using a noisily rational control policy, the human actions is taken from a Boltzmann distribution, parameterized by the optimality parameter, according to a corresponding action-value (Q) function, which is inspired by RL. In addition, we use a Bayesian framework to estimate the *distribution* of all model parameters, and also use this distribution to perform prediction.

4.0.2 Problem statement

Let $X_H \in \mathbb{R}^n$ be the human state, where n is the dimension of the state space. We assume that the human dynamics is given by the ODE given in Eq. (2.1) and it is moving toward an unknown target state in an unobstructed space. Hence, we have $\dot{X}_H = f_H(X_H, u_H)$. Our objective is to estimate the probability distribution of the human goal (g), optimality factor (β), and farsightedness parameter (γ) by observing the current behaviour. We also evolve the particles for the next steps by resampling from a mixture of previous particles, and a specified set of goals. We can forecast human future states by bringing internal parameters together.

4.1 HNIPO model

In this section, we propose a model for **H**uman **N**avigational **I**ntent inference with **P**robabilistic **O**ptimal approach (HNIPO). The graphical model of the system to present the causality and connectivity of different components of the model is shown in Fig. 4.1. The internal human parameters for the inference model are g, β, γ, which respectively represent goal state, optimality factor, and farsightedness and affects the action taken by human. According to the bottom part of the Fig. 4.1, at every time step, given the distribution of human actions u_H from any state X_H, we can compute a distribution of future human states with the recursive update:

$$P(X_H^{\tau+1}|X_H^\tau; \beta, g, \gamma) = \textstyle\sum_{u_H^\tau} P(X_H^{\tau+1}|X_H^\tau, u_H^\tau; \beta, g, \gamma) P(u_H^\tau|X_H^\tau; \beta, g, \gamma), \tag{4.1}$$

where

$$P(X_H^{\tau+1}|X_H^\tau, u_H^\tau; \beta, g) = I(X_H^{\tau+1} = \tilde{f}_H(X_H^\tau, u_H^\tau)), \tag{4.2}$$

where I is the indicator function.

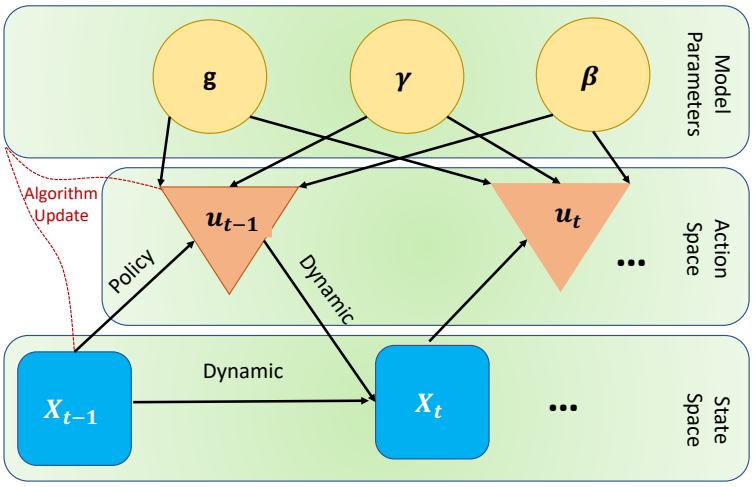

Figure 4.1: Graphical model for human navigation; parameters: goal state (g), farsightedness parameter (γ), optimality factor (β). Action space: turn rate and acceleration; state space: position, heading, velocity

4.1.1 Time-to-reach and Q function with goal at origin

In general, we believe that humans want to get to their destination reasonably fast. As a result, humans behave in accordance with an action-value (Q) function, which reflects the discounted penalty over time.

$$Q^{\pi}(X_H, u_H, \gamma) = \mathbb{E}_{\pi}\{G_t | s_t = X_H, u_t = u_H\}, \tag{4.3}$$

where G_t is the total discounted reward defined in Eq. (**??**) and

$$G_t = R_{t+1} + \gamma V(X_H), \tag{4.4}$$

where $r = -1$ for any state transition and V is the value function. Since the state space is 4D and the dynamics Eq. (2.1) are non-trivial, online computation of Q is not tractable. To overcome this challenge, we pre-compute a TTR function by solving Eq. (2.2) to compute value function $V(X_H) = -\phi(X_H)$. This TTR function represents the time it takes the human to reach the target set of the origin in position and heading space, and is obtained by solving Eq. (2.2) with

$$\begin{aligned} H &= -P^T f \\ P^T &= [V_x, V_y, V_\theta, V_v] \\ H &= v.cos(\theta)V_x + v.sin(\theta)V_y + \omega.V_\theta + a.V_v \end{aligned} \tag{4.5}$$

where V_μ is the partial derivative of V with respect to μ. Observing that the TTR is the *undiscounted* sum of reward, we can compute the Q function as follows:

$$Q(X_H, u_H, \gamma) = V\frac{1-\gamma^\tau}{1-\gamma}, \tag{4.6}$$

4.1.2 Map Q function for arbitrary goals

The goal denotes the location and direction in which the person can come to a stop. As a result, the person will arrive at its destination and the velocity will be zero. This issue forces us to project the pre-calculated TTR map to the current goal. Since the dynamics in Eq. (2.1) is invariant in translation and rotation, we can transform the Q function corresponding to the target set being at the origin to a Q function corresponding to a target set being at any position and heading.

More conveniently, we can assume that the goal is always at the origin, and define the human position and heading in the reference frame of the goal (x_g, y_g, θ_g):

$$\begin{bmatrix} x_m \\ y_m \\ \theta_m \end{bmatrix} = \begin{bmatrix} cos(\theta_g) & -sin(\theta_g) & 0 \\ sin(\theta_g) & cos(\theta_g) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x - x_g \\ y - y_g \\ \theta - \theta_g \end{bmatrix} \tag{4.7}$$

where $(x_g, y_g), \theta_g$ are respectively the position and heading of the goal and $(x_m, y_m), \theta_m$ are the mapped position and heading of the human (X_{Hm}). We assume that when the goal is reached, the human will have a speed of zero while facing it. Hence, the Q function is computed as follows:

$$Q(X_H, u_H, \gamma, g) = Q(X_{Hm}, u_H, \gamma, (0, 0, 0)) \tag{4.8}$$

4.1.3 Noisily rational policy

The human's policy is a probability distribution of actions conditioned on the state. We assume the following noisily rational policy [18]:

$$P(u_H^t | X_H^t; \beta, g, \gamma) = \frac{e^{\beta Q_H(X_H^t, u_H^t; g, \gamma)}}{\sum_{\tilde{u}} e^{\beta Q_H(X_H^t, \tilde{u}; g, \gamma)}} \tag{4.9}$$

The optimality parameter β is an internal human parameter that can be inferred from observed human motion and its predictive model marginalising over this hidden state. Eq. (4.9) is a Boltzmann distribution that gives probability of a system to be in a certain state as a function of that state's energy [62].

Remark 3: To tackle the soft-max overflow problem, we rearrange Eq. (4.9) as follows:

$$P(u_H^t | X_H^t; \beta, g, \gamma) = \frac{e^{\beta(Q_H^t - Q_{H,max})}}{\sum_{\tilde{u}} e^{\beta(Q_H^t - Q_{H,max})}} \tag{4.10}$$

where $Q_{H,max} = \max_a\{Q_H\}$ and Q_H^t is the Q value for the action u_H^t.

4.2 HNIPO internal parameter updates

In this section, we address a framework to update HNIPO internal parameters. For the sake of simplicity, we compute joint probability distribution of parameters as follows:

$$p(g, \beta, \gamma) = p(g|\beta, \gamma)P(\beta|\gamma)P(\gamma) \tag{4.11}$$

4.2.1 Estimating γ

The update formula for γ is obtained using the Bayes' rule as follows:

$$b^{t+1}(\gamma) = P(\gamma|X_H^{0:t+1}, g, \beta) = \frac{P(u_H^t|X_H^t; \gamma, g, \gamma)b^t(\gamma)}{\sum_{\hat{\gamma}} P(u_H^t|X_H^t; \hat{\gamma}, g, \gamma)b^t(\hat{\gamma})} \tag{4.12}$$

where $\gamma = 0$ is for short-sighted humans while $\gamma = 1$ is for far-sighted humans. Weighted average (Expected value) of the variable set is used as the representative of that variable.

4.2.2 Estimating β

The update relation of β is given based on the measured human action. At every time step t, we obtain a new measurement of the human's action u_H^t which is used as evidence to update β:

$$b^{t+1}(\beta) = P(\beta|X_H^{0:t+1}, g, \gamma) = \frac{P(u_H^t|X_H^t; \beta, g, \gamma)b^t(\beta)}{\sum_{\hat{\beta}} P(u_H^t|X_H^t; \hat{\beta}, g, \gamma)b^t(\hat{\beta})} \tag{4.13}$$

where $P(u_H^t|X_H^t; \beta, g, \gamma)$ is obtained from Eq. (4.10). While in theory $\beta \in [0, \infty]$, in practice we only maintain a Bayesian belief over a relatively small set of β values.

4.2.3 Estimating g

The update relation for the goals state are same as other internal parameters. According to Bayes' rule we have:

$$b^{t+1}(g) = \frac{P(u_H^t|X_H^t; \beta, g, \gamma)b^t(g)}{\sum_{\hat{g}} P(u_H^t|X_H^t; \hat{g}, g, \gamma)b^t(\hat{g})}. \tag{4.14}$$

Since goal states are in 3D continuous random variable, we use particle filter to represent its distribution. We assume that the particle weights are modified in accordance with Eq. (4.14). We assume a random walk for the evolution of the goal particles. Furthermore, the particles are created using a combination of previous particles, and a fixed set of predefined goals to accommodate changes in human goals. The fixed sets are selected in such a way that the entropy is maximised. As a result, we use a uniform distribution to generate a parameter $\zeta \in [0, 1]$.

$$g_t^{(i)} = \begin{cases} G^{(k)} & \zeta \geq \zeta^{th} \\ g_{t-1}^{(i)} + \nu & \zeta < \zeta^{th} \end{cases} \tag{4.15}$$

where, $g_t^{(i)}$ is the i-th particle at time t, $G^{(k)}$ is the k^{th} element chosen randomly from the fixed set of particles mentioned above, $\nu \sim N(0, \sigma)$ controls the random walk variance, and $\zeta^{th} \in [0, 1]$ is the model mismatch index defined in Eq. (3.9) [3]. This controls the probability of particles chosen from a fixed subset.

$$\zeta^{th} = \mathbb{E}\{G\} = \sum_k b^{t+1}(g) P(u_H^t | X_H^t, \theta, g_k) \tag{4.16}$$

Therefore, the probability of goal states are computed as follows:

$$P(g | X_H^{0:t+1}, \beta, \gamma) = \frac{1}{N} \sum_{i=1}^{N} \delta_{xi}(g_t^{(i)}) \tag{4.17}$$

where N is the number of particles and δ_{xi} is the Dirac measure.

Remark 4: The aim in here is to take into account several sequences of behaviour taken by the person to reason on their goal using the particle filter. However, given a set of actions, there are a number of states that can be called goal candidates. The explanation for this is that given a human's current state, there are many final states of (sub)optimal paths with the same trajectory in a given time frame. On the other hand, the fact that this approach has several goals, has little effect on its primary goal of predicting human navigational intent. Conceptual goals, as the name implies, are pivots that allow human prediction, not necessarily the human actual final state.

4.2.4 Human state prediction

In this section, we will apply some probabilistic relations to simplify the human state prediction. Dynamic model is generally given by $p(X_{t+1}|X_t, u_t)$ and control policy is available in $p(u_t|X_t, \beta_t, \gamma_t, g_t)$, therefore we have

$$\begin{aligned} p(X_{t+1}|X_t) &= p(X_{t+1}|X_t, u_t) p(u_t|X_t) \\ &= p(u_t|X_t, \beta_t, \gamma_t, g_t) \sum_{\beta_t, \gamma_t, g_t} p(\beta_t, \gamma_t, g_t|X_t) \end{aligned} \tag{4.18}$$

To further simplify the probability distribution of human occupancies overtime we have:

$$\begin{aligned} p(X_{t+1}|X_t) &= \sum_{u_t} p(X_{t+1}|X_t, u_t) p(u_t|X_t) \\ &= \sum_{u_t} p(X_{t+1}|X_t, u_t) \\ &\sum_{\beta_t, \gamma_t, g_t} p(u_t|X_t, \beta_t, \gamma_t, g_t) p(\beta_t, \gamma_t, g_t|X_t) \end{aligned} \tag{4.19}$$

Belief is defined as: $b_t(\beta, \gamma, g) = p(\beta, \gamma, g|X_{0:t})$

$$
\begin{aligned}
p(\beta,\gamma,g|X_{0:t+1}) &= p(\beta,\gamma,g|X_{t+1},X_{0:t}) \\
&= \eta p(X_{t+1}|X_{0:t},\beta,\gamma,g)p(\beta,\gamma,g|X_{0:t})
\end{aligned}
\tag{4.20}
$$

$$
\begin{aligned}
b_{t+1}(\beta,\gamma,g) &= \eta p(X_{t+1}|X_{0:t},\beta,\gamma,g)b_t(\beta,\gamma,g) \\
&= \eta p(X_{t+1}|X_{0:t},u_t,\beta,\gamma,g)p(u_t|X_{0:t},\beta,\gamma,g)b_t(\beta,\gamma,g) \\
&= \eta p(X_{t+1}|X_{0:t},u_t,\beta,\gamma,g)p(u_t|X_t,\beta,\gamma,g)b_t(\beta,\gamma,g) \\
&= \eta p(X_{t+1}|X_{0:t},u_t)p(u_t|X_t,\beta,\gamma,g)b_t(\beta,\gamma,g)
\end{aligned}
\tag{4.21}
$$

By substituting Eq. (4.21) in (4.19), we have

$$
p(X_{t+1}|X_t) = \sum_{u_t} I(X_{t+1} = f(X,u_t)))p(u_t|X_t,\beta_t,\gamma_t,g_t)b_t(\beta,\gamma,g) \tag{4.22}
$$

The direct computation of this relation is intractable due to the 4D state and grid points. Hence, we use particle filter representation to forecast the human state when taking into account all cases in the grid. We must draw samples from the probability distribution computed in Eq. (4.10) since the control input in this problem is unknown. A fair representation of the true state will be provided by sufficient particles with random inputs. The weights are modified depending on the current state's measurements.

$$
\begin{aligned}
u_i &\sim p(u_H^t|X_H^t;\beta,g,\gamma) \\
X_t^{(i)} &= FX_{t-1}^{(i)} + Bu_i + \nu \\
w^{(i)} &= e^{(X_t^{(i)} - \hat{X}_t)}
\end{aligned}
\tag{4.23}
$$

where, $w^{(i)}$ is the weight of the i-th particle and $\hat{X}_t$ is the measurement of X at time t. Since we do not have access to the control inputs, u_i is the i-th sample from the control distribution. The other parameters are:

$$
F = \begin{pmatrix} 1 & 0 & 0 & \delta t\cos(\theta_{t-1}^i) \\ 0 & 1 & 0 & \delta t\sin(\theta_{t-1}^i) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, B = \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ \delta t & 0 \\ 0 & \delta t \end{pmatrix} \tag{4.24}
$$

Remark 5: For n-steps ahead prediction when $n > 1$, we select a few goal candidates based on their probability and make the (sub)optimal path to achieve them individually. Finally, we represent the weighted average as a point estimate and in a heat map illustration, which includes all particles.

4.3 Results

We now demonstrate our method using synthetic and real-world data. Initially, we assume that $\beta \in \{0.1, 10\}$ and the discount factor $\gamma \in \{0.9, 0.99\}$ have a uniform distribution in all experiments. The following represent the TTR configuration: The grid boundaries are

$$|x| < 8, |y| < 8, |\theta| < \pi, |v| < 2.2 \tag{4.25}$$

In order to be able to take gradients from TTR, the grid needs to include negative values for V. However, as any negative value is not physically meaningful, we need to set $v < 0$ as an obstacle. Therefore the control constraints are

$$a \in [-1, 1], \omega \in [-1.5, 1.5]. \tag{4.26}$$

Target setting is $(0, 0, 0, 0)$ with radius 0.2, and obstacles' radius is 0.15 in $[-0.2, 2]$.

4.3.1 Synthetic data

For the numerical simulation involving synthetic data, we assumed the ground truth parameters are $\Delta = 0.2, T_f = 20$, where Δ is the sampling interval and T_f is the time frame. We also considered the input bounds of human controls to be $|a| \leq 1, |\omega| \leq 1.5$. We discretized it with a precision of 0.5 for the simulation, resulting in 35 variations. These parameters were used to create synthetic trajectories on which our prediction algorithm was implemented. The goal distribution was represented by 50 particles in the prediction stage, while the human states were represented by 200 particles.

The particles reflect distributions over the targets for horizon one and three, respectively, in Fig. 4.2, 4.3. As more insight is gained by watching the human's behaviour, the states shift with time. The goal particles, eventually converge to their true values. Fig. 4.4 reflects the heat map in four different time slots to display the distribution over different behaviour in each time step. The probability distributions of γ and β over time are illustrated in Fig. 4.5. Both begin at 0.5 and gradually approach the ground truth.

The goal particles are shown in green, and their point estimates are shown in black in Fig. 4.7 (which is the weighted average of the particles). The estimated future of human states is depicted in Fig. 4.6, which includes both the particle representation and the weighted average. The weighted average is intractable with horizons greater than one since the number of evaluations grows exponentially. The particle representation approach, however, is scalable with longer time horizons.

To compare our results with a recent literature ([19]), Fig. 4.8 shows predictions made in 4 different time steps to derive the route of a human. Fig. 4.9 shows an updated profile of the β parameter.

To have a detailed comparison, Table 3.1 compares the final result of this method with [19]. The brute force (BF) and particle filter point estimate (PF) using our approach, with the result of the weighted average of confidence aware (WA CA) in [19] are listed in Table 3.1. The result of adding various inputs to the dynamic and multiplying them by their corresponding probability (after marginalization over other parameters such as β, γ) is brute force. The point estimate for the particle filter is a simple average of the remaining particles after re-sampling, while the point estimate for the particle filter is a simple average of the surviving particles after re-sampling (so we may have multiple selection of same inputs, and miss some other inputs). BF produces more reliable results when it considers all possible outcomes; however, it is not analytic tractable for predictions of more than one stage. As a result, we can use a particle filter as a sub-optimal solution. As can be seen, incorporating the human's 4D state into the inference process expands the source of information available to the model and, as a result, improves prediction accuracy. If the horizon lengthens, this distinction becomes more apparent.

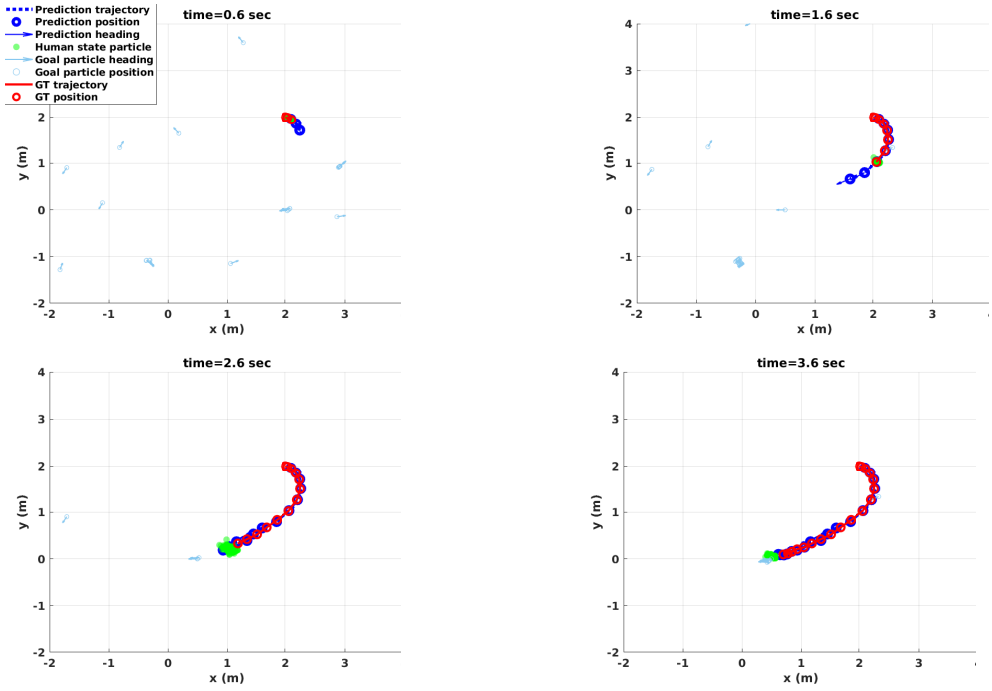

Figure 4.2: Prediction results for $\tau = 1$. Particle filter of states (green dots), particle filter of goal (purple arrows), ground truth path of human (solid red line), and black dotted arrows predicted position and orientation of human.

4.3.2 Real-world data

This dataset includes translational data from a motion capture device of people travelling in an indoor environment in real-world coordinates. To minimise noise derivatives and correct inconsistencies in time intervals in the results, we use a Kalman filter. The speed and

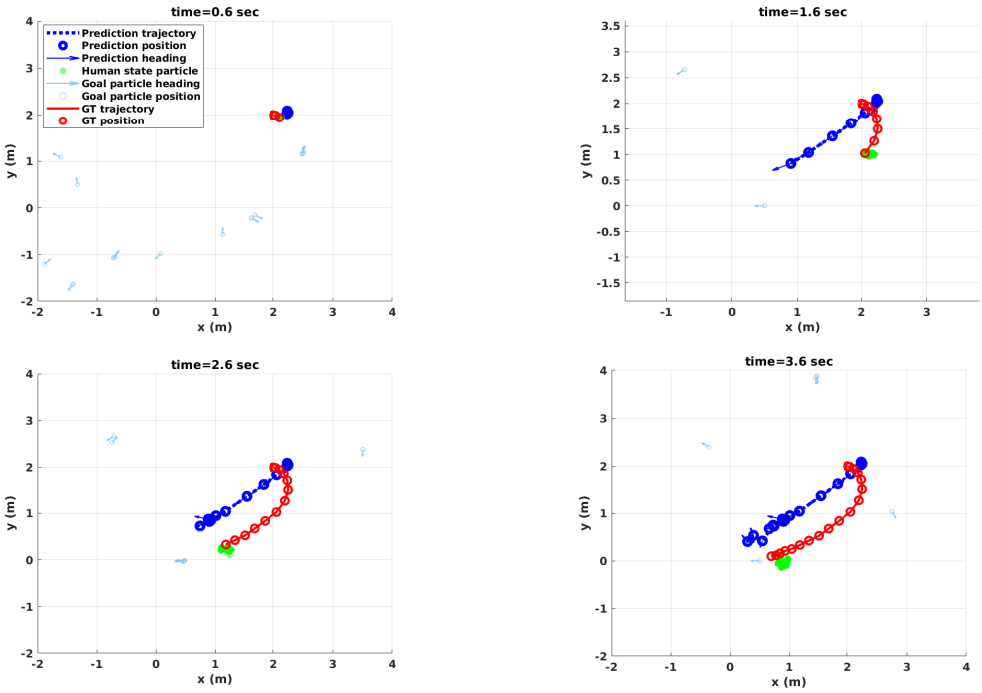

Figure 4.3: Prediction results for $\tau = 3$. Particle filter of states (green dots), particle filter of goal (purple arrows), ground truth path of human (solid red line), and black dotted arrows predicted position and orientation of human.

heading angle were then determined by combining positional data, the sampling time is 0.1 Sec. Finally, we use the same procedure to approximate the controls (acceleration and turn rate). The particle filter of goal and state is shown in Fig 4.10. When an individual is moving towards a distant destination, they can move faster than if they are moving towards a closer one. In real-time, the BF solution, which considers all possible combinations of goals/parameters given system states/actions, is computationally unfeasible. The profile of ground truth, BF, and PF over time is depicted in Fig. 4.12. As can be shown, PF is capable of estimating the underlying likelihood with reasonable accuracy.

Table 4.1: Comparison of our approach with [19] simulation results

Errors	x (m)	y (m)	θ (rad)	v (m/sec)
PF ours ($\tau = 1$)	0.3796	0.3738	0.7980	0.7193
BF ours ($\tau = 1$)	0.1462	0.1261	0.7043	0.5903
WA CA ($\tau = 1$)	0.2813	0.2408	NA	NA
PF ours ($\tau = 3$)	0.8491	0.8552	1.9848	1.6163
BF ours ($\tau = 3$)	NA	NA	NA	NA
WA CA ($\tau = 3$)	1.5234	1.4256	NA	NA

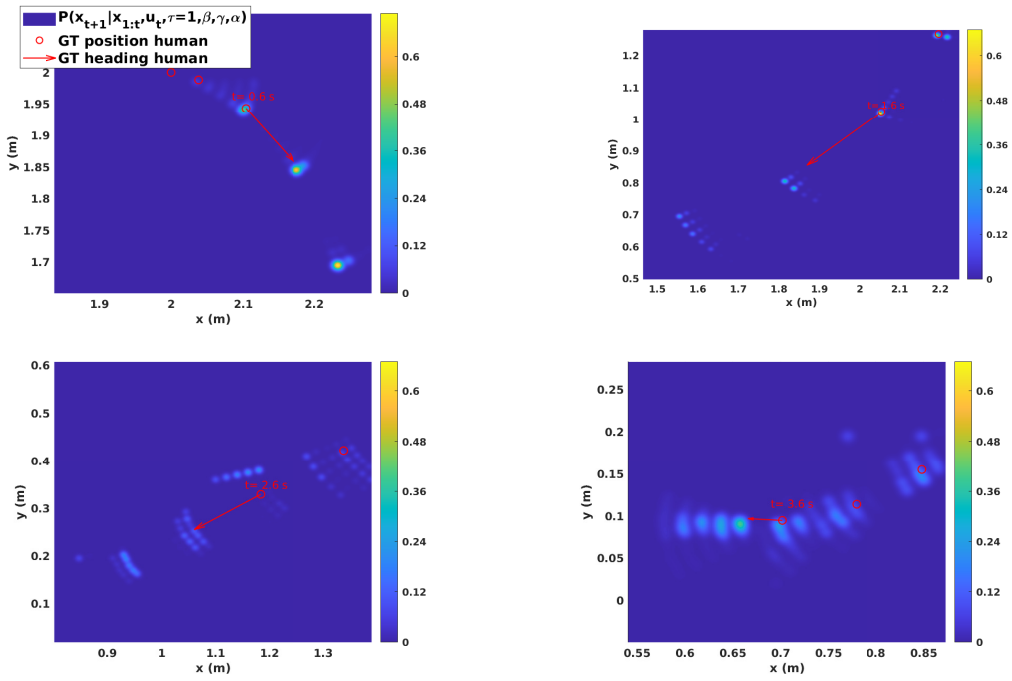

Figure 4.4: The heat map for future state at each time step for one step ahead prediction ($\tau = 1$) and different goals. Ground truth positions in red circle.

Table 4.2: Comparison of our approach with [19] for real data

Errors	x (m)	y (m)	θ (rad)	v (m/sec)
PF ours ($\tau = 1$)	0.0231	0.0229	0.0384	0.0312
BF ours ($\tau = 1$)	0.0083	0.0079	0.0104	0.0087
WA CA ($\tau = 1$)	0.0323	0.0367	NA	NA

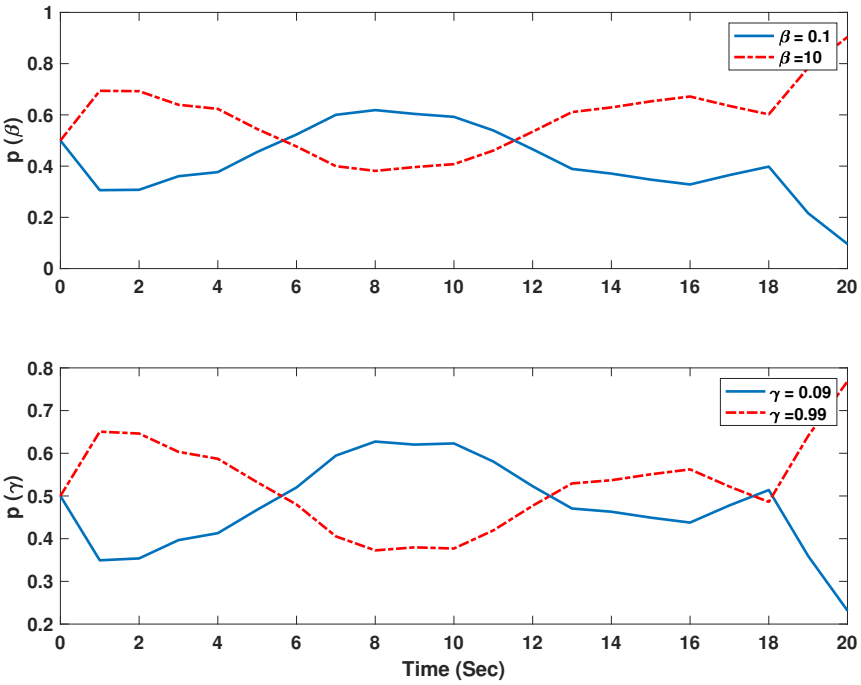

Figure 4.5: The probability of two values of β, γ parameters in the experiment with synthetic data generated with $\beta = 10$.(γ only inferred during prediction). As the probability for one of them is reaching 1 at the end, means that the framework identified the true internal parameter with a high confidence.

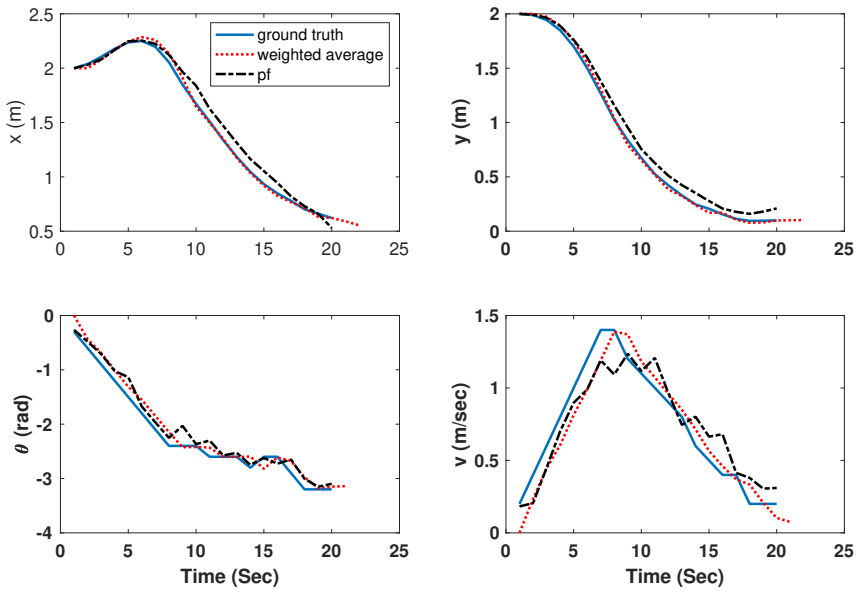

Figure 4.6: Evolution of the particle filter of human states over time. The particle filter give reasonable estimation of probability as the point estimate is close to the ground truth.

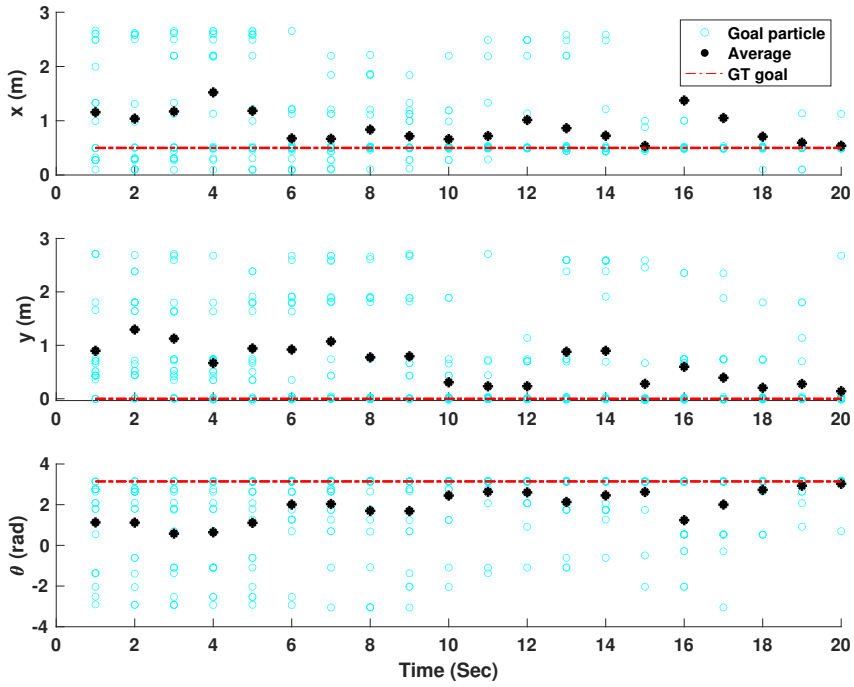

Figure 4.7: Evolution of the particle filter of the goal over time. At the end of the simulation, the particle filters are less disperse and converged to the ground truth.

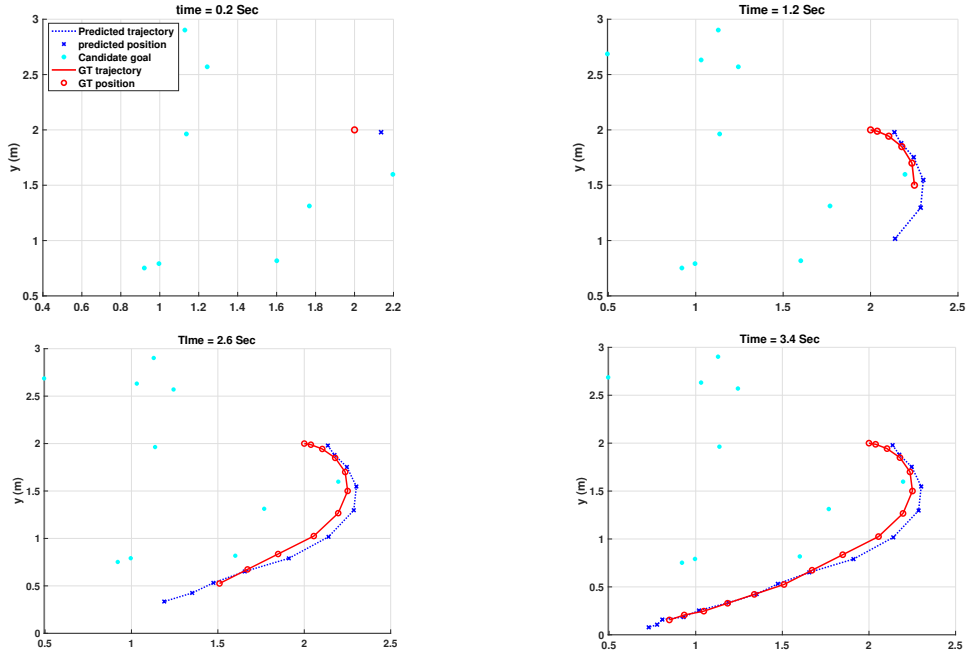

Figure 4.8: The ground truth trajectory of the human (solid red line), and predicted trajectory of the human using [19] shown in blue dashed line.

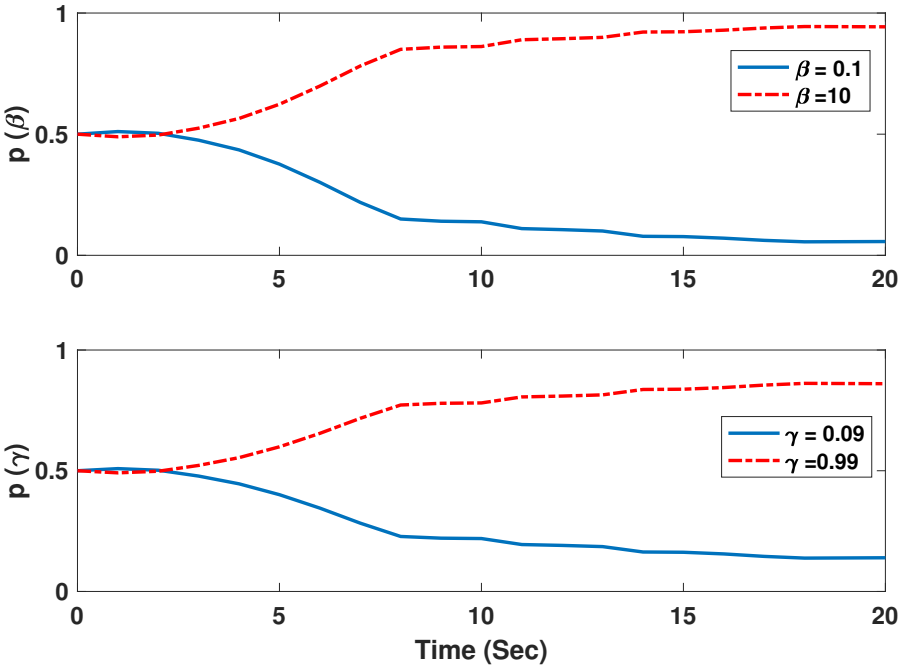

Figure 4.9: The probability of β, γ using [19]

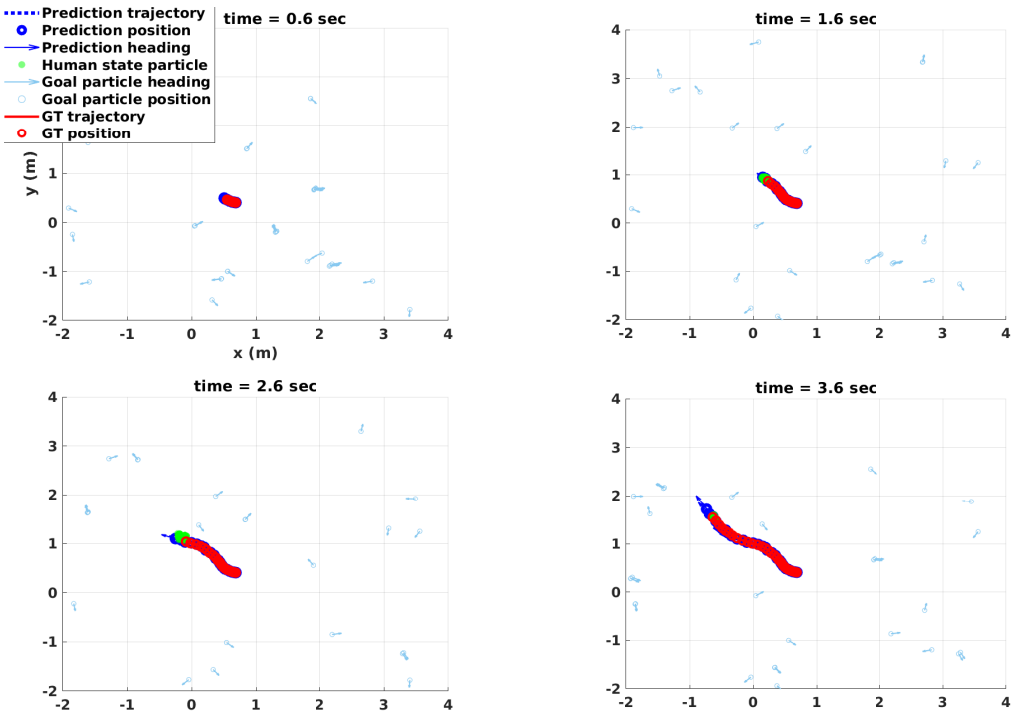

Figure 4.10: Particle filter of states (green dots), particle filter of goal (purple arrows), ground truth path of human (solid red line), and black dotted arrows predicted position and orientation of human, real data

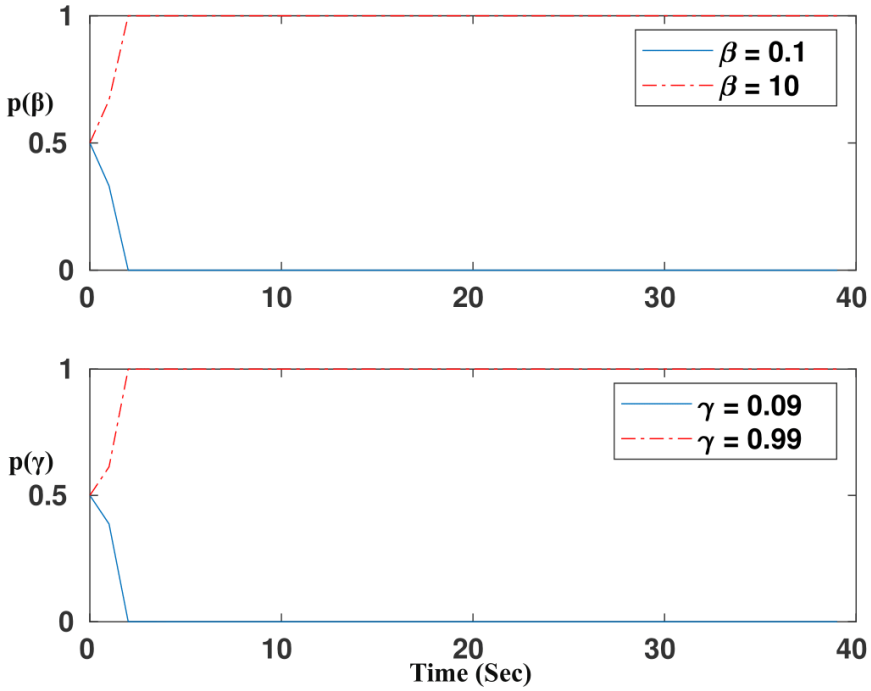

Figure 4.11: Probability of β and γ for real data

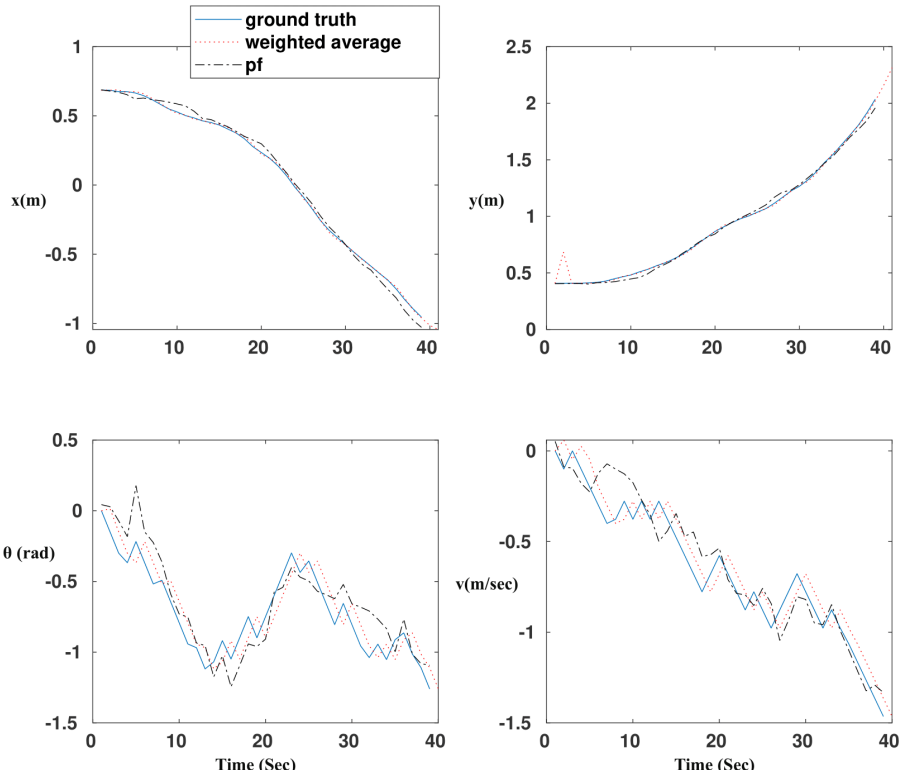

Figure 4.12: Evolution of the particle filter of human states over time, real data k

Chapter 5

Conclusion

5.1 Discussion

Using a Bayesian paradigm, this thesis proposed an adaptive recursive MCMC (ARMCMC) algorithm for online estimation of the complete probability distribution of model parameters. Conventional methods work poorly when applied to systems involving sudden shifts in model parameters, which may occur when contact with a soft environment is formed or lost. The Hunt-Crossley model parameters, which is a nonlinear hybrid dynamic model, was identified with ARMCMC and compared to a well-known conventional identification. The results show superiority of the proposed algorithm. The proposed approach adapts easily to sudden shifts, allowing the method to be applicable to a wider range of systems.

We also proposed HNIPO model and update framework, a probabilistic approach to predict human future states. The use of a 4D state space for human dynamics addressed technical problems for a principled derivation of the Q function. It's also used to calculate the distribution of model parameters and individual states over time. To conclude, we suggested a probabilistic human navigation model based on optimal control assumptions in this thesis. The key aspects of this model are as follows: 1) We compute discounted action-value functions (Q) functions in real time by transforming a TTR value function in the position and heading space, and mapping the TTR value to corresponding Q value, 2) we use the farsightedness parameter to decide how the model sacrifices the current rewards to achieve the future ones, inspired by the discount factor in reinforcement learning, 5) we use a particle representation of distributions over model parameters and human states to reduce the computational burdens associated with the 4D state space.

Adding velocity and heading to the dynamics is supposed to make a difference between close and far targets appearing in the same direction. Furthermore, using TTR instead of geographical distance will provide a more precise calculation of path optimality. A particle filter is proposed to estimate the likelihood of the state, and the goal to handle the algorithm's complexity given the 4D model and three tuning parameters. It is also worth noting that a large number of particles can be ignored because they are associated with a low

confidence index. For real-time applications, one can sample a few surviving particles each time to compute $P(u)$ and the corresponding value of interest. The only thing that should be considered is the bias of the estimation.

The down-sampling method from the current probability distribution of the particles can be used to address the aforementioned problem. Only certain sampled particles are manipulated during the updating steps.

5.2 Future directions

To fully explore the potentials of Bayesian optimization, deploying a fully probabilistic framework from detection to control and decision-making is being seen as future work. It would also be considered whether or not to use a system to compensate for the delay. To compensate for the two algorithm time delays, one potential change is to provide a prediction stage in the ARMCMC algorithm. Using an expected maximizing method is one approach.

One of the missing parts of the HNIPO architectures is an evolution model for the hypothetical desires aka goal position. We need to think about how to deal with a goal shift in the middle of an episode. Finding an approach to confine the possibility of accepted goals given a series of control inputs may be seen as a second potential work.

Collision avoidance is critical for path planning in dynamic uncertain environments. One can use the knowledge given by the ARMCMC method for other robots in the scene or the HNIPO approach for humans based on the previous chapters to provide a collision-free navigation in complex setting. There are hierarchical framework in literature which utilizes both global and local information; a high-level viewpoint is given by a global path planar that encapsulates static obstacles and goal reaching information offline, while a local observer provides a real time probabilistic occupancy map for all obstacles (including dynamic obstacles) in the field of view (FOV). FOV in here refers to the range of environment perception via any data acquisition tool (e.g. LiDAR, vision). A RL method can be utilized to provide a safe path for the robot that maneuvers collision-free in the environment with static and dynamic obstacles with the existing probabilistic knowledge.

State-of-the-art methods (e.g. [40, 77, 25]) often use re-planning techniques in the presence of dynamic obstacles, which re-call a planning algorithm to look for an alternate direction if the robot finds a new obstacle. However, such approaches sometimes result in unneeded detours [71]. Meanwhile, some approaches (e.g. [76, 1, 59]) consider a deterministic constraint around dynamic obstacles in a robust planning scheme which mainly suffers from performance due to being too conservative. There are a few probabilistic approaches in the literature (e.g. [69, 55]) that require a complete knowledge of the obstacles to be ready at the beginning of the simulation. Not only are not available these level of knowledge in many cases, but also in practice, they could only be available when the robot approaches them

(not in advance). To address these problems, one can propose a learning-based approach that uses environmental spatio-temporal knowledge in the form of consecutive local maps.

In addition, ARMCMC can be utilizes as a parameter estimator to obtain probability distribution of goal parameters in the HNIPO framework. To this end, we need an evolution model for the goal candidates which could be the mean and variance of the normal distribution of the random walk.

Bibliography

[1] Baqir Nasser AbdulSamed, Ammar A Aldair, and Auday Al-Mayyahi. Robust trajectory tracking control and obstacles avoidance algorithm for quadrotor unmanned aerial vehicle. *Journal of Electrical Engineering & Technology*, 15(2):855–868, 2020.

[2] Niki Abolhassani, Rajni Patel, and Mehrdad Moallem. Needle insertion into soft tissue: A survey. *Medical Engineering and Physics*, 29(4):413–431, 2007.

[3] Pedram Agand, Mo Chen, and Hamid D. Taghirad. Armcmc: Bayesian online full density estimation of model parameters. In *Submited: Thirty-fifth Conference on Neural Information Processing Systems*, 2021.

[4] Pedram Agand and Mahdi Aliyari Shoorehdeli. Adaptive model learning of neural networks with uub stability for robot dynamic estimation. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–6. IEEE, 2019.

[5] Pedram Agand, Hamid D Taghirad, and Ali Khaki-Sedigh. Particle filters for non-gaussian hunt-crossley model of environment in bilateral teleoperation. In *4th International Conference on Robotics and Mechatronics (ICROM)*, pages 512–517. IEEE, 2016.

[6] Alexandre Alahi, Kratarth Goel, Vignesh Ramanathan, Alexandre Robicquet, Li Fei-Fei, and Silvio Savarese. Social lstm: Human trajectory prediction in crowded spaces. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 961–971, 2016.

[7] S Bhasin, K Dupree, PM Patre, and WE Dixon. Neural network control of a robot interacting with an uncertain hunt-crossley viscoelastic environment. In *Dynamic Systems and Control Conference*, volume 43352, pages 875–882, 2008.

[8] Christopher M Bishop. Pattern recognition. *Machine Learning*, 128, 2006.

[9] Omar Boufous. Deep reinforcement learning for complete coverage path planning in unknown environments, 2020.

[10] Steve Brooks, Andrew Gelman, Galin Jones, and Xiao-Li Meng. *Handbook of markov chain monte carlo*. CRC press, 2011.

[11] Zhe Cao, Hang Gao, Karttikeya Mangalam, Qi-Zhi Cai, Minh Vo, and Jitendra Malik. Long-term human motion prediction with scene context. In *European Conference on Computer Vision*, pages 387–404. Springer, 2020.

[12] André S Carvalho and Jorge M Martins. Exact restitution and generalizations for the hunt–crossley contact model. *Mechanism and Machine Theory*, 139:174–194, 2019.

[13] Zhixian Chen, Chao Song, Yuanyuan Yang, Baoliang Zhao, Ying Hu, Shoubin Liu, and Jianwei Zhang. Robot navigation based on human trajectory prediction and multiple travel modes. *Applied Sciences*, 8(11):2205, 2018.

[14] Liron Cohen, Tansel Uras, Shiva Jahangiri, Aliyah Arunasalam, Sven Koenig, and TK Kumar. The fastmap algorithm for shortest path computations. *arXiv preprint arXiv:1706.02792*, 2017.

[15] Noah J Cowan, Ken Goldberg, Gregory S Chirikjian, Gabor Fichtinger, Ron Alterovitz, Kyle B Reed, Vinutha Kallem, Wooram Park, Sarthak Misra, and Allison M Okamura. Robotic needle steering: Design, modeling, planning, and image guidance. In *Surgical Robotics*, pages 557–582. Springer, 2011.

[16] Patrick Dendorfer, Aljosa Osep, and Laura Leal-Taixé. Goal-gan: Multimodal trajectory prediction based on goal position estimation. In *Proceedings of the Asian Conference on Computer Vision*, 2020.

[17] Nicola Diolaiti, Claudio Melchiorri, and Stefano Stramigioli. Contact impedance estimation for robotic systems. *IEEE Transactions on Robotics*, 21(5):925–935, 2005.

[18] Jaime F Fisac, Andrea Bajcsy, Sylvia L Herbert, David Fridovich-Keil, Steven Wang, Claire J Tomlin, and Anca D Dragan. Probabilistically safe robot planning with confidence-based human predictions. *arXiv preprint arXiv:1806.00109*, 2018.

[19] David Fridovich-Keil, Andrea Bajcsy, Jaime F Fisac, Sylvia L Herbert, Steven Wang, Anca D Dragan, and Claire J Tomlin. Confidence-aware motion prediction for real-time collision avoidance1. *The International Journal of Robotics Research*, 39(2-3):250–265, 2020.

[20] Mohammed Geda and CK Kwong. An mcmc based bayesian inference approach to parameter estimation of distributed lag models for forecasting used product returns for remanufacturing. *Journal of Remanufacturing*, pages 1–20, 2021.

[21] Peter J Green. Reversible jump markov chain monte carlo computation and bayesian model determination. *Biometrika*, 82(4):711–732, 1995.

[22] Peter L Green. Bayesian system identification of a nonlinear dynamical system using a novel variant of simulated annealing. *Mechanical Systems and Signal Processing*, 52:133–146, 2015.

[23] Amir Haddadi and Keyvan Hashtrudi-Zaad. Real-time identification of hunt-crossley dynamic models of contact environments. *IEEE transactions on robotics*, 28(3):555–566, 2012.

[24] Hani Hagras, Victor Callaghan, and Martin Colley. Learning and adaptation of an intelligent mobile robot navigator operating in unstructured environment based on a novel online fuzzy–genetic system. *Fuzzy Sets and Systems*, 141(1):107–160, 2004.

[25] Mehmet Hasanzade and Emre Koyuncu. A dynamically feasible fast replanning strategy with deep reinforcement learning. *Journal of Intelligent & Robotic Systems*, 101(1):1–17, 2021.

[26] Dirk Helbing and Peter Molnar. Social force model for pedestrian dynamics. *Physical review E*, 51(5):4282, 1995.

[27] Michael Hoy, Alexey S Matveev, and Andrey V Savkin. Algorithms for collision-free navigation of mobile robots in complex cluttered environments: a survey. *Robotica*, 33(3):463–497, 2015.

[28] KH Hunt and FRE Crossley. Coefficient of restitution interpreted as damping in vibroimpact. *Journal of applied mechanics*, 42(2):440–445, 1975.

[29] Dominik Joho, Gian Diego Tipaldi, Nikolas Engelhard, Cyrill Stachniss, and Wolfram Burgard. Nonparametric bayesian models for unsupervised scene analysis and reconstruction. *Robotics: Science and Systems VIII*, page 161, 2013.

[30] Oskar Karlsson. Lidar-based slam: Investigation of environmental changes and use of road-edges for improved positioning, 2020.

[31] Amna Khan, Iram Noreen, and Zulfiqar Habib. On complete coverage path planning algorithms for non-holonomic mobile robots: Survey and challenges. *Journal of Information Science & Engineering*, 33(1), 2017.

[32] Shima Khatibisepehr, Biao Huang, and Swanand Khare. Design of inferential sensors in the process industry: A review of bayesian methods. *Journal of Process Control*, 23(10):1575–1596, 2013.

[33] Tarasha Khurana, Achal Dave, and Deva Ramanan. Detecting invisible people. *arXiv preprint arXiv:2012.08419*, 2020.

[34] Dong Ki Kim and Tsuhan Chen. Deep neural network for real-time autonomous indoor navigation. *arXiv preprint arXiv:1511.04668*, 2015.

[35] Bahare Kiumarsi, Kyriakos G Vamvoudakis, Hamidreza Modares, and Frank L Lewis. Optimal and autonomous control using reinforcement learning: A survey. *IEEE transactions on neural networks and learning systems*, 29(6):2042–2062, 2017.

[36] Marina Kollmitz, Kaijen Hsiao, Johannes Gaa, and Wolfram Burgard. Time dependent planning on a layered social cost map for human-aware robot navigation. In *2015 European Conference on Mobile Robots (ECMR)*, pages 1–6. IEEE, 2015.

[37] Henrik Kretzschmar, Markus Spies, Christoph Sprunk, and Wolfram Burgard. Socially compliant mobile robot navigation via inverse reinforcement learning. *The International Journal of Robotics Research*, 35(11):1289–1307, 2016.

[38] Thibault Kruse, Amit Kumar Pandey, Rachid Alami, and Alexandra Kirsch. Human-aware robot navigation: A survey. *Robotics and Autonomous Systems*, 61(12):1726–1743, 2013.

[39] Scott Kuindersma, Roderic Grupen, and Andrew Barto. Variational bayesian optimization for runtime risk-sensitive control. *Robotics: Science and Systems VIII*, 2012.

[40] Geesara Kulathunga, Roman Fedorenko, Sergey Kopylov, and Alexandr Klimehik. Real-time long range trajectory replanning for mavs in the presence of dynamic obstacles. In *2020 5th Asia-Pacific Conference on Intelligent Robot Systems (ACIRS)*, pages 145–153. IEEE, 2020.

[41] Tomasz Kuśmierczyk, Joseph Sakaya, and Arto Klami. Variational bayesian decision-making for continuous utilities. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 6392–6402. Curran Associates, Inc., 2019.

[42] Xiaoyun Lei, Zhian Zhang, and Peifang Dong. Dynamic path planning of unknown environment based on deep reinforcement learning. *Journal of Robotics*, 2018, 2018.

[43] Ming Liu, Francis Colas, Luc Oth, and Roland Siegwart. Incremental topological segmentation for semi-structured environments using discretized gvg. *Autonomous Robots*, 38(2):143–160, 2015.

[44] Ming Liu, Francis Colas, François Pomerleau, and Roland Siegwart. A markov semi-supervised clustering approach and its application in topological map extraction. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4743–4748. IEEE, 2012.

[45] Xubo Lyu and Mo Chen. Ttr-based reward for reinforcement learning with implicit model priors. *arXiv preprint arXiv:1903.09762*, 2019.

[46] Karttikeya Mangalam, Yang An, Harshayu Girase, and Jitendra Malik. From goals, waypoints & paths to long term human trajectory forecasting. *arXiv preprint arXiv:2012.01526*, 2020.

[47] Karttikeya Mangalam, Harshayu Girase, Shreyas Agarwal, Kuan-Hui Lee, Ehsan Adeli, Jitendra Malik, and Adrien Gaidon. It is not the journey but the destination: Endpoint conditioned trajectory prediction. *arXiv preprint arXiv:2004.02025*, 2020.

[48] Kenneth O May et al. R. duncan luce, individual choice behavior, a theoretical analysis. *Bulletin of the American Mathematical Society*, 66(4):259–260, 1960.

[49] William M McEneaney. *Max-plus methods for nonlinear control and estimation*. Springer Science & Business Media, 2006.

[50] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.

[51] Brett Ninness and Soren Henriksen. Bayesian system identification via markov chain monte carlo techniques. *Automatica*, 46(1):40–51, 2010.

[52] Ali Noormohammadi-Asl and Hamid D Taghirad. Multi-goal motion planning using traveling salesman problem in belief space. *Information Sciences*, 471:164–184, 2019.

[53] Xinlei Pan, Weiyao Wang, Xiaoshuai Zhang, Bo Li, Jinfeng Yi, and Dawn Song. How you act tells a lot: Privacy-leaking attack on deep reinforcement learning. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*, pages 368–376, 2019.

[54] Mike Phillips and Maxim Likhachev. Sipp: Safe interval path planning for dynamic environments. In *2011 IEEE International Conference on Robotics and Automation*, pages 5628–5635. IEEE, 2011.

[55] Ankit A Ravankar, Abhijeet Ravankar, Takanori Emaru, and Yukinori Kobayashi. Hpprm: Hybrid potential based probabilistic roadmap algorithm for improved dynamic path planning of mobile robots. *IEEE Access*, 8:221743–221766, 2020.

[56] Andrey Rudenko, Luigi Palmieri, Michael Herman, Kris M Kitani, Dariu M Gavrila, and Kai O Arras. Human motion trajectory prediction: A survey. *The International Journal of Robotics Research*, 39(8):895–935, 2020.

[57] Kaur Aare Saar, Fabio Giardina, and Fumiya Iida. Model-free design optimization of a hopping robot and its comparison with a human designer. *IEEE Robotics and Automation Letters*, 3(2):1245–1251, 2018.

[58] Tim Salzmann, Boris Ivanovic, Punarjay Chakravarty, and Marco Pavone. Trajectron++: Multi-agent generative trajectory forecasting with heterogeneous data for control. *arXiv preprint arXiv:2001.03093*, 2020.

[59] Baskın Şenbaşlar, Wolfgang Hönig, and Nora Ayanian. Robust trajectory execution for multi-robot teams using distributed real-time replanning. In *Distributed Autonomous Robotic Systems*, pages 167–181. Springer, 2019.

[60] Chuangchuang Sun, Macheng Shen, and Jonathan P How. Scaling up multiagent reinforcement learning for robotic systems: Learn an adaptive sparse communication graph. *arXiv preprint arXiv:2003.01040*, 2020.

[61] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[62] John Bradbury Sykes, LD Landau, MJ Kearsley, LP Pitaevski, and EM Lifshits. *Statistical Physics (Course of Theoretical Physics)*. Butterworth-Heinemann, 1980.

[63] Roberto Tempo, Giuseppe Calafiore, and Fabrizio Dabbene. *Randomized algorithms for analysis and control of uncertain systems: with applications*. Springer Science & Business Media, 2012.

[64] Luca Tiseni, Domenico Chiaradia, Massimiliano Gabardi, Massimiliano Solazzi, Daniele Leonardis, and Antonio Frisoli. Uv-c mobile robots with optimized path planning: Algorithm design and on-field measurements to improve surface disinfection against sars-cov-2. *IEEE Robotics & Automation Magazine*, 28(1):59–70, 2021.

[65] Felipe Tobar. Bayesian nonparametric spectral estimation. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 10127–10137. Curran Associates, Inc., 2018.

[66] Peter Trautman and Andreas Krause. Unfreezing the robot: Navigation in dense, interacting crowds. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 797–803. IEEE, 2010.

[67] Xuan-Tung Truong and Trung Dung Ngo. Toward socially aware robot navigation in dynamic and crowded environments: A proactive social motion model. *IEEE Transactions on Automation Science and Engineering*, 14(4):1743–1760, 2017.

[68] Dizan Vasquez, Billy Okal, and Kai O Arras. Inverse reinforcement learning algorithms and features for robot navigation in crowds: an experimental comparison. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1341–1346. IEEE, 2014.

[69] Abraham P Vinod and Meeko MK Oishi. Probabilistic occupancy function and sets using forward stochastic reachability for rigid-body dynamic obstacles. *arXiv preprint arXiv:1803.07180*, 2018.

[70] Beilun Wang, Arshdeep Sekhon, and Yanjun Qi. A fast and scalable joint estimator for integrating additional knowledge in learning multiple related sparse Gaussian graphical models. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 5161–5170, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018. PMLR.

[71] Binyu Wang, Zhe Liu, Qingbiao Li, and Amanda Prorok. Mobile robot path planning in dynamic environments through globally guided reinforcement learning. *IEEE Robotics and Automation Letters*, 5(4):6932–6939, 2020.

[72] Tao Wang, Yunce Zhang, Zheng Chen, and Shiqiang Zhu. Parameter identification and model-based nonlinear robust control of fluidic soft bending actuators. *IEEE/ASME Transactions on Mechatronics*, 24(3):1346–1355, 2019.

[73] Grady Williams, Brian Goldfain, Paul Drews, Kamil Saigol, J Rehg, and Evangelos A Theodorou. Robust sampling based model predictive control with sparse objective information. In *Robotics Science and Systems*, 2018.

[74] Insoon Yang, Sabine Becker-Weimann, Mina J Bissell, and Claire J Tomlin. One-shot computation of reachable sets for differential games. In *Proceedings of the 16th international conference on Hybrid systems: computation and control*, pages 183–192, 2013.

[75] Zhitian Zhang, Jimin Rhim, Angelica Lim, and Mo Chen. A multimodal and hybrid framework for human navigational intent inference. 2020.

[76] Xiaolin Zhao, Yu Zhang, and Boxin Zhao. Robust path planning for avoiding obstacles using time-environment dynamic map. *Measurement and Control*, 53(1-2):214–221, 2020.

[77] Boyu Zhou, Fei Gao, Jie Pan, and Shaojie Shen. Robust real-time uav replanning using guided gradient-based optimization and topological paths. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1208–1214. IEEE, 2020.

Appendix A

Hunt-Crossley Model

According to [2], a nonlinear hybrid model based on a reality-based soft environment is considered as follows:

$$f_e = f_{\text{st}}(x, t, t_p) + f_{\text{fr}}(x, \dot{x}) + f_{\text{ct}}(x, t, t_p), \tag{A.1}$$

where x is the needle tip position and t_p is the latest time of puncture. Initial position of the environment is assumed to be at the origin. The stiffness of the force (f_{st}) belongs to a pre-puncture and the friction (f_{fr}) and cutting forces (f_{ct}) belong to a post-puncture. The stiffness force is modeled using a nonlinear Hunt-Crossley model:

$$f_{st}(x, t, t_p) = \begin{cases} 0 & x < 0 \\ K_e x^p(t) & 0 \leq x \leq x_1, t < t_p \\ 0 & x > x_2, t \geq t_p \end{cases} \tag{A.2}$$

where K_e, p are the same parameters defined in (3.15). The maximum depth that the soft environment yields before the puncture and its position after it is denoted by x_1, x_2, respectively ($0 < x_2 < x_1$). In this study, the needle can insert up to 16.65, 10.21 mm before and after penetration. A friction model is inspired from modified Karnopp model.

$$f_{fr}(x, \dot{x}) = \begin{cases} C_n sgn(\dot{x}) + B_e x^p \dot{x} & \dot{x} \leq -\Delta v/2 \\ \max(D_n, F_a) & -\Delta v/2 < \dot{x} \leq 0 \\ \max(D_p, F_a) & 0 < \dot{x} < \Delta v/2 \\ C_p sgn(\dot{x}) + B_e x^p \dot{x} & \dot{x} \geq \Delta v/2 \end{cases} \tag{A.3}$$

where $C_n = -11.96 \times 10^{-3}$ and $C_p = 10.57 \times 10^{-3}$ are negative and positive values of dynamic friction, $D_n = -0.01823$ and $D_p = 0.01845$ are negative and positive values of static friction, and B_e, p are same as Eq. (3.15). The relative velocity between the needle and tissue is denoted by $\dot{x}$, and $\Delta v/2 = 0.005$ is the value below where the velocity is considered to be zero, and F_a is the sum of non-frictional applied forces. The cutting force is considered as a static force constant for the tissue and the needle geometry if soft tissues

are not inhomogeneous and anisotropic [15]

$$f_{ct}(x,t,t_p) = \begin{cases} 0 & x \leq x_1, t < t_p \\ 0.94 & x > x_2, t \geq t_p \end{cases}. \tag{A.4}$$

According to the previous relations, the system is considered as a hybrid model while providing both free motion and in-contact environment. The manipulator is a translational mechanism with a friction, slip, and hysteresis loop for the actuator. To present the superiority of the proposed algorithm, the results are compared with the RLS method presented in [23]. To prevent the results of RLS from divergence in model mismatch sequences, saturation is applied in the outputs of the identifier.