Robot Motion Planning in Dynamic, Cluttered, and Uncertain Environments: the Partially Closed-Loop Receding Horizon Control Approach

Thesis by

Noel E. Du Toit

In Partial Fulfillment of the Requirements
for the Degree of
Ph.D

California Institute of Technology
Pasadena, California

2010
(Defended November 10, 2009)

Aan my Ouers, vir hulle eindelose inspirasie, opoffering, liefde, en ondersteuning.

To my Parents, for their endless inspiration, sacrifice, love, and support.

Acknowledgments

I wish to extend my gratitude to the various people who have contributed to my graduate career and to making it a rewarding and constructive experience:

A big thank you to my advisor, Dr. Joel Burdick, for his mentorship, encouragement, understanding, and unwaivering support. Your commitment to your students and family truly is an inspiration.

I am exceedingly grateful to Dr. Richard Murray for his mentorship and commitment during the DARPA Urban Challenge. To the rest of my thesis committee members, Dr. Pietro Perona, Dr. Lars Blackmore, and Dr. Jim Beck, for their guidance. A special thanks to Maria Koeper for her efforts on my behalf.

To my friends, the Robotics research group (past and present), and especially my office mate, Jeremy Ma: thanks for all the discussions, laughes, and lunches.

To the SOPS members that make Thomas such a unique and special place, and the Caltech community at large.

To my family who have inspired and supported me during my (seemingly endless) student career: I am eternally grateful for your love and sacrifice.

Finally, to Andrea, for keeping me sane and fed, and for sharing this journey.

Abstract

This thesis is concerned with robot motion planning in dynamic, cluttered, and uncertain environments. Successful and efficient robot operation in such environments requires reasoning about the future system evolution and the uncertainty associated with obstacles and moving agents in the environment. Current motion planning strategies ignore future information and are limited by the resulting growth of uncertainty as the system is evolved. This thesis presents an approach that accounts for future information gathering (and the quality of that information) in the planning process. The Partially Closed-Loop Receding Horizon Control approach, introduced in this thesis, is based on Dynamic Programming with imperfect state information. Probabilistic collision constraints, due to the need for obstacle avoidance between the robot and obstacles with uncertain locations and geometries, are developed and imposed. By accounting for the anticipated future information, the uncertainty associated with the system evolution is managed, allowing for greater numbers of moving agents and more complex agent behaviors to be handled. Simulation results demonstrate the benefit of the proposed approach over existing approaches in static and dynamic environments. Complex agent behaviors, including multimodal and interactive agent-robot models, are considered.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This work is concerned with robot motion planning in *Dynamic, Cluttered,* and *Uncertain Environments* (DCUEs). Current approaches result in either very aggressive, in the case of reactive planners, or overly cautious, in the case of open-loop approaches, robot behavior. This research is aimed at generating efficient, safe motion plans for a robot in these complex environments.

1.1 Motivation

In the DCUE problem, robots must work in close proximity with many other moving agents whose future actions and reactions are possibly not well known or characterizable. The robot is plagued by uncertainty or noise in its own state measurements and in the sensing of obstacles and moving agents. Moreover, the large number of moving agents may offer many distractors to the robot's sensor processing systems. An example of a DCUE application is a service robot which must cross through a swarm of moving humans in a cafeteria during a busy lunch hour in order to deliver food items. Clearly, the future human trajectories cannot be predicted with any certainty, and the motion planning and social interaction rules used by each of the humans may be different, thereby complicating the planning problem. The clutter provided by so many humans is also likely to severely strain the robot's visual, ladar, and sonar sensor processing systems.

A motion planning framework is desired that can account for the different sources of uncertainty in the DCUE problem. The uncertainty associated with the prediction of future positions of the large number of moving agents in cluttered environments is especially constraining when solving the motion planning problem in a DCUE. Current approaches that account for the prediction uncertainty do not take future measurments into account, which results in an undesired growth in the uncertainty. Additionally, as the complexity of the behavior of these agents increase, the prediction uncertainty increases. It becomes very important to manage the uncertainty growth in the system due to these predictions, otherwise, very cautious motion plans are produced, which lower potential performance. This work introduces a motion planning framework that allows for the impact of future information gathering operations to be accounted for during the planning process.

Furthermore, the quality of information that will be obtained in the future is affected by the nature of the planned trajectory. This coupling between the motion plan and the information gathering quality necessitates the combined solving of the motion planning, estimation, and prediction problems. For simplicity, most approaches to date have purposely separated these processes. For the application of interest here, this separation is not possible in general, nor is it desirable.

Lastly, due to the uncertainty in the robot and obstacle locations and geometry, a probabilistic collision avoidance framework is necessary to ensure safe operation. Prior strategies have been limited to systems with known obstacle locations and geometries. An approach is needed that accounts for both the robot and obstacle uncertainties. This thesis presents some initial results for such a framework.

1.2 Literature Review

Robot motion planning in dynamic environments has recently received substantial attention due to the DARPA Urban Challenge [20] and growing interest in service and assistive robots (e.g., [47, 51]). In urban environments, traffic rules define the expected behaviors of the dynamic agents, and this information can be used to partially constrain expected future locations. In other applications, agent behaviors are less well defined, and the prediction of their future trajectories is more uncertain. To date, various proposed motion planning algorithms and frameworks handle only specific subsets of the DCUE problem.

Classical motion planning algorithms [17, 34, 35] mostly ignore uncertainty when planning in static or dynamic environments. When the future locations of moving agents are known, the two common approaches are to add a time-dimension to the configuration space, or velocity tuning (the spatial and temporal planning problems are separated) [35]. When the future locations of the moving agents are unknown, the planning problem is either solved locally [17, 22, 24] (reactive planners in an assumed static environment), or a global planner guides the robot towards the goal and a local planner reacts to the dynamic component of the environment [17, 18, 35]. One attempt to extend the local planner to uncertain environments is the Probabilistic Velocity Obstacle approach [26], where uncertainty associated with the obstacle geometry and velocity vector is used to artificially grow the velocity obstacle in an open-loop fashion.

Planning algorithms for stochastic systems have been developed, but in static environments. Two types of stochastic systems are distinguished: systems where the uncertainties lie in a bounded set[1], and probabilistic systems where the uncertainties are discribed by probability distribution functions [35]. Probabilistic systems are of interest here. The first stochastic planning approach was pre-image

[1] This is often referred to as non-deterministic systems (e.g., [35])

back-chaining [41]. Since then, discrete search strategies have also been extended to probabilistic systems where the problem is solved in an extended state space of pose×covariance (e.g., [16, 27, 45]). These algorithms are only applied in static environments.

Dynamic Programming (DP) is a well established and mature field, and it is not the intention of the author to exhaustively review the field. For this, the reader is referred to texts by Bertsekas [6, 7]. The topic is treated in more detail in Section 2.1. DP is geared towards decision making under uncertainty, and starts at the goal state and recursively chooses optimal action in a backwards temporal direction. A control policy, giving the optimal control action for every reachable state, is obtained. Initially, only process noise was considered in the *perfect state information* (PSI) problem: the outcome of the applied action is uncertain, but the state at each stage is perfectly measured. Later, the approach was extended to systems with *imperfect state information* (ISI): the outcome of the applied control is uncertain and the state can be only imperfectly measured. The DP ISI problem is reformulated in the belief space [53], where the stochastic properties of the original system are encapsulated in the new (belief) states. This approach is very general, but the major disadvantage is the computation requirements. Very few problems allow closed form solution [1, 6], and approximations must normally be used [6, 53]. The second shortcoming is that hard constraints are not readily handled in this framework. Depending on the system at hand and the cost function, different approximations have been proposed.

When the system equations (dynamic and measurement) are time-invariant and the cost incurred at each stage is constant, then the DP ISI problem can be solved using Partially Observable Markov Decision Process (POMDP) methods such as value function iteration and policy iteration [6, 53]. This also is a mature field, and the reader is referred to Thrun et al. [53] for a detailed treatment.

When the system equations and cost do not satisfy these conditions (as is the case in this research), the DP ISI problem can be approximately solved with the Rollout algorithm (a limited lookahead strategy) or a restricted information approach [6]. The latter is of particular interest here: the information set over which the planning problem is solved is restricted, effectively reducing the set of reachable states in the problem (since the complete set of reachable states are defined by all the possible future measurements). The effect of this simplification is that a *control sequence* is obtained for the restricted set of reachable states, instead of a control policy over all reachable states. The problem must be recursively solved to incorporate the true new information. These approximations are detailed in Section 3.2.2. This framework is relevant here since it can be used to formulate the stochastic receding horizon control approach, which then allows for constraints to be incorporated.

It is often necessary to impose constraints on the states and the controls of the system. *Receding horizon control* (RHC), also known as *Model Predictive Control*, is a suboptimal control scheme which is motivated by the need to handle nonlinear systems, and to incorporate state and control constraints explicitly in the planning problem. The reader is referred to recent review papers on the topic, for example [42, 43]. The (sub)optimal planning problem is solved from the current stage over a shorter horizon than that which the problem is posed on, and then a small portion of the plan is executed. New measurements are obtained, the system states are updated, and the problem is re-solved. The RHC approach was originally developed for deterministic systems, but has been extended to a stochastic RHC formulation in the particular case of robot localization uncertainty (e.g., [**?**, 14, 32, 38, 48, 60, 61]). Again, two types of stochastic systems are considered: nondeterministic systems (e.g., [14, 32]), and probabilistic systems. The latter is of interest here. The stochasticity of the system has two effects on the constrained optimization problem: the uncertainty needs to be accounted for in the cost function, and the constraints on the states of the system must be imposed as chance constraints.

Notebly, Yan and Bitmead [61] incorporated state estimation into the RHC problem. A linear, Gaussian system in a static environment is assumed, and all future information is ignored. This results in the open-loop prediction of the evolution of the system states for which the uncertainty grows. This growth in uncertainty results in overly cautious solutions. In an attempt to manage this uncertainty growth, the *closed-loop covariance* is introduced: the predicted state covariance is *artificially* fixed at the one-step ahead covariance. There is a need to manage this uncertainty growth in a more formal manner, which is the main topic of the current research.

In a similar fashion, Blackmore's RHC-based motion planning approach ignored the future measurements in the assumed linear system, but used a particle approach to estimate the predicted distributions of the robot location [8], thus moving away from the strictly Gaussian noise assumption. A piecewise linear cost function and a static environment are assumed. This work was extended in [12] to nonlinear systems, using linearization and then applying the same particle method. Again, due to the use of open-loop predictions, the uncertainty of the evolved states grows, and a formalism is needed to incorporate the anticipated future measurements in the planning process.

Accounting for the anticipated future information in the planning problem is not a novel concept *when applied to static environments*. In fact, Fel'dbaum first introduced the *dual effect* of control actions: actions affect the information gathered about the system, as well as progress the system towards the goal. This field is known as *dual control*, and the reader is referred to the survey by Filatov and Unbehauen [21] and the paper by Londoff et al. [39]. Historically, the dual control research community has been interested in controlling a system with unknown parameters and states, so that the controls must be chosen to optimally gather information about the model parameters, and drive the system towards the goal. The work by Bar-Shalom and Tse [1, 2, 3] is of particular interest since the problem is posed in the DP ISI framework. The generally nonlinear system equations and cost function are approximated with second-order Taylor expansions, and it is shown that the cost function decomposes into deterministic, caution, and probing parts. This work does not consider constraints on the planning problem and the problem is posed in a static environment, but the formulation of the problem as a stochastic DP is built upon in this work.

A major consequence of the dual effect of control is that the estimation and planning processes become coupled. In many common systems, the estimation and planning problems are separable according to the separation principle (e.g., [25]). The reader is referred to [1] for a discussion of the connection between the dual effect in control and the separation principle. Research into the combined solution of these problems results in a trade-off between exploration and exploitation (e.g., [40, 49]). These conflicting problems are commonly solved seperately, and then iterated to obtain the optimal solution (e.g., [37, 54]). These approaches use the dynamic programming framework and do not consider state constraints in the formulation.

An alternative approach to incorporating future information has been suggested by Bemporad [4] who used the *closed-loop* state prediction to reduce the conservativeness of solutions for a constrained linear system with Gaussian noise. This notion is utilized by Van Hessem and Bosgra [56, 60, 59]: by explicitly feeding back the future measurements in the planning process, the effect of the controls on the quality of the future measurements can be captured. The approach converts the system equations into trajectories, for which the linear system equations are necessary. Gaussian noise is assumed, and the control sequence is selected that minimizes the overall trajectory covariance. This formulation is also used in [9]. One drawback of the method is that the optimization problem must be solved using a large gain matrix over the whole trajectory. As a result, it is unclear how this method will scale with trajectory length and state-space size. Additionally, it is not clear how the anticipated future measurements are actually used during the planning process, making it hard to extend to more general scenarios.

Chance constraints have been commonly applied in stochastic control and stochastic RHC (e.g., [9, 38, 48, 59, 61]). Generally linear constraints are enforced on the assumed Gaussian distributed system states. The chance constraints are converted into an amount of *back-off* that is a function of the covariance, and the constraint is then applied to the system mean state. Blackmore et al. [10] applied this framework to probabilistic obstacle avoidance, assuming that the obstacle geometry and location are known. Multiple obstacles are also handled, necessitating the joint collision evaluation over trajectories, instead of each separate time step. This resulted in very conservative constraint enforcement [11]. Ono and Williams [44] and Blackmore [11] have since introduced the notion of risk allocation, where the joint constraint assigns more probablity mass to parts of the trajectory where collisions are more likely, and less mass to parts where the robot is far from obstacles. The part of probabilistic obstacle avoidance that is missing in these prior efforts is the accounting for the uncertainty in obstacle geometry and location (currently, only the robot uncertainty is considered). This problem is addressed in Section 4.1.3.

The problem of predicting future dynamic agent states has received some attention. Short term predictors evolve the future state of the dynamic agents using a simple model such as a constant velocity model (e.g., [33, 62]). Longer term predictions can be constrained by learning the dynamic agents' preferred paths and using these paths to predict future states (e.g., [5, 28]), or by inferring structure in the environment which can inform the prediction process (e.g., [23, 52]). It is desirable to incorporate this structure that becomes known about the environment in the planning and prediction processes. In Section 5.1, a first attempt is made to incorporate this prior information in the agent models by accounting for multiple possible destinations in the agent models.

1.3 Thesis Contributions and Layout

While individual components of the DCUE problem have been previously considered, a comprehensive framework that integrates planning, prediction, and estimation as needed to solve the DCUE planning problem has been missing. This thesis represents the first formal effort to incorporate the effect of anticipated future measurements in the motion planning process in dynamic environments. As shown by example, the proper inclusion of these effects can improve robot performance in the presence of uncertain agent behavior. Additionally, a novel analysis of the probabilistic collision constraints between moving objects whose positions are uncertain is presented. Because the exact DCUE solution is generally intractable, the Partially Closed-Loop Receding Horizon Control (PCLRHC) approach is derived, based on the stochastic RHC framework, which in turn is motivated from a dynamic programming point of view. This approach successfully manages the growth in uncertainty as the system is propagated. As a result, feasible solutions can be obtained in environments with more moving agents and agents with more complex behaviors. Simulation results for a robot navigating in a static environments and dynamic environments of increasing complexity are presented to illustrate some of the characteristics of this method.

The thesis is organized as follows:

- Chapter 2: Necessary background information on the dynamic programming approach is presented, including the definition of information states and spaces, the belief state and space, and the formulation of dynamic programming in terms of the belief state. Additionally, receding horizon control and chance constraints are introduced. Finally, the optimal state estimation problem is formulated, and standard results are presented.

- Chapter 3: Approximations to the stochastic dynamic programming problem ignore future information and do not readily handle constraints, but are used to motivate and derive the Partially Closed-Loop Receding Horizon Control (PCLRHC) approach. This novel approach accounts for the anticipated future measurements in the system and is equally applicable in static and dynamic environments. The benefit of this algorithm over standard approaches is demonstrated for a simple example in a static environment.

- Chapter 4: The PCLRHC algorithm is applied to dynamic scenarios where the agent behavior is independent of the robot actions. A novel approach to probabilistic obstacle avoidance is presented that accounts for the uncertainty associated with the agent location and geometry. Estimation results for agents with unknown parameters are presented. Simulation results demonstrating the benefit of the PCLRHC approach over standard approaches are presented.
- Chapter 5: The PCLRHC algorithm is applied to dynamic scenarios with more complicated agent behaviors, but which are still independent of the robot actions. These behaviors are induced by incorporating environment structure in the agent models, or by using multiple simple models. The resulting predicted agent states have multimodal distributions, and extensions to estimation results and the probabilistic obstacle avoidance approach are presented. Simulation results demonstrate the benefit of the PCLRHC approach over standard approaches.
- Chapter 6: The PCLRHC algorithm is applied to dynamic scenarios with agent behaviors which are dependent on the robot actions. The extension of the probabilistic obstacle avoidance approach to capture this dependence is presented. First, a state-dependent measurement noise term is used to capture the effect of the robot actions on the quality of agent information, and simulation results demonstrate the ability of the robot to choose actions that improve information quality (active learning). Second, friendly and adversarial agent models are used that capture simple interaction between the robot and the agent. The robot can use knowledge of this anticipated interaction to improve the obtained solution, as demonstrated by example.
- Chapter 7: Conclusions and future work are presented.

Chapter 2

Background

The objective of the current research is to make progress towards a planning approach for complex dynamic, uncertain environments where all aspects of the problem can be easily and elegantly incorporated. The general problem requires the combined solution of the estimation, prediction (using anticipated measurements), and the planning problems. These problems must be solved simultaneously since the estimation and prediction processes depend on the proposed plan — the quality of information is in general a function of the executed path. Additionally, it is necessary to impose constraints on the states and controls of the system to ensure safe operation.

Dynamic programming addresses some of these issues: the combined planning, prediction, and estimation problem can be formulated. The major disadvantages of the dynamic programming approach is that hard constraints are not readily handled, and the approach is computationally expensive (often intractable). Still, it is very useful to investigate the dynamic programming problem to understand the use of information in the formulation and during the approximations to this problem. Receding horizon control was formulated in an attempt to incorporate constraints in the optimal control problem, and to handle nonlinear systems. The basic receding horizon control approach for deterministic systems is presented in this chapter. When imposing constraints on the state of a stochastic system, these constraint must be imposed as chance constraints. Standard results for linear chance constraints are presented. Lastly, standard results for recusive estimation theory are presented.

2.1 Dynamic Programming

Dynamic programming (DP) is a mathematical formulation for making sequential decisions under uncertainty due to disturbances (process noise) and measurement noise ([6], p. 2). The planning objective is to minimize a cost function. Decisions cannot be viewed in isolation, but instead the cost incurred at the current step needs to be traded off against cost incurred in the future. The basic model for the dynamic programming problem studied in this thesis has two features:

1. An underlying, discrete time dynamic system (both continuous state or discrete state problems can be handled),
2. An additive cost function.

The *perfect state information* planning problem and the *imperfect state information* planning problem are considered. In decision theory, the former is referred to as a Markov Decision Process (MDP), and the latter as a Partially Observable Markov Decision Process (POMDP) [35, 53]. The basic dynamic programming algorithm is developed for the perfect state information case. The imperfect state problem can be converted into the perfect state problem by reformulating the problem in terms of the information state.

2.1.1 Feedback Control with Perfect State Information

Let the state, $x_k \in \mathbb{X}$, be an element of the *state space*, $\mathbb{X} \subseteq \mathbb{R}^{n_x}$. The control, $u_k(x_k) \in \mathbb{U}(x_k)$, is an element of the *action space,* $\mathbb{U}(x_k) \subseteq \mathbb{R}^{n_u}$. The disturbance, $\omega_k(x_k, u_k) \in \mathbb{W}(x_k, u_k)$, is described by the conditional distribution

$$\omega_k(x_k, u_k) \sim p(\omega_k | x_k, u_k) \tag{2.1}$$

and is an element of the set $\mathbb{W}(x_k, u_k) \in \mathbb{R}^{n_\omega}$. The disturbance does not depend on the previous disturbances, $\omega_{0:k-1}$. *Perfect state information* (PSI) refers to the fact that x_k is known at stage k, however, disturbances cause the outcome of a dynamic process after the application of a control, u_k, to be uncertain.

Definition 2.1: *Feedback Control Problem with Perfect State Information*

Consider the following discrete-time dynamic system

$$x_{k+1} = f\left(x_k, u_k, \omega_k\right) \tag{2.2}$$

where $f : \mathbb{X} \times \mathbb{U} \times \mathbb{W} \rightarrow \mathbb{X}$ *is the* state transition function *which is assumed to be* C^2 *(continous twice differentiable).*

Consider a class of control policies[1], Π, *which consists of a set of functions,* $\pi : \mathbb{X} \rightarrow \mathbb{U}$, *that map the states into controls:* $u_k = \pi_k(x_k)$.

$$\Pi = \{\pi_0(x_0), \ldots, \pi_{N-1}(x_{N-1})\} \tag{2.3}$$

[1] A distinction between open-loop planning and closed-loop planning is necessary. For open-loop planning, the *sequence of controls* are selected at time 0, $\{u_0, \ldots, u_{N-1}\}$. In contrast, closed-loop planning aims to select a policy Π which applies control $u_k = \pi_k(x_k)$ at stage k *with knowledge of* x_k. Dynamic programming is a formulation of closed-loop planning.

Assume a stage-additive cost *function written in terms of the control policy:*

$$L(x_0, \Pi, \omega_{0:N-1}\}) = l_N(x_N) + \sum_{k=0}^{N-1} l_k(x_k, \pi_k(x_k), \omega_k). \tag{2.4}$$

The expected cost associated with a control policy can be calculated:

$$C_\Pi(x_0) = E_{\omega_{0:N-1}} \left[l_N(x_N) + \sum_{k=0}^{N-1} l_k(x_k, \pi_k(x_k), \omega_k) \right] \tag{2.5}$$

where the expectation is over the joint distribution of the random variables, $\omega_{0:N-1}$. The optimal policy, $\Pi^{(*)} = \{\pi_0^{(*)}(x_0), \ldots, \pi_{N-1}^{(*)}(x_{N-1})\}$ minimizes the expected cost over the set of admissible policies., $\widetilde{\Pi}$:

$$\begin{aligned} \Pi^{(*)} &= \arg\min_{\Pi \in \widetilde{\Pi}} \; C_\Pi(x_0) \\ &= \arg\min_{\Pi \in \widetilde{\Pi}} \; E_{\omega_{0:N-1}} \left[l_N(x_N) + \sum_{k=0}^{N-1} l_k(x_k, \pi_k(x_k), \omega_k) \right] \end{aligned}$$

and the optimal cost is given by

$$C^{(*)}(x_0) \quad = \quad E_{\omega_{0:N-1}} \left[l_N(x_N) + \sum_{k=0}^{N-1} l_k(x_k, \pi_k^{(*)}(x_k), \omega_k) \right]. \tag{2.6}$$

2.1.2 Principle of Optimality and Dynamic Programming

The dynamic programming algorithm is derived from the *principle of optimality* ([6] p. 18).

Proposition 2.1: *Principle of Optimality*

Let $\Pi^{(*)} = \{\pi_0^{(*)}, \ldots, \pi_{N-1}^{(*)}\}$ be the control policy that minimizes Eq. (2.5), and assume that during the execution of this optimal policy, a state x_i is reached with positive probability. Now consider the subproblem of starting at stage i, state x_i, and solving the minimization:

$$C^{(*)}(x_i) = \min_{\pi_{i:N-1}} E_{\omega_{i:N-1}} \left[l_N(x_N) + \sum_{k=i}^{N-1} l_k(x_k, \pi_k(x_k), \omega_k) \right].$$

Then the truncated policy $\{\pi_i^{(*)}(x_i), \ldots, \pi_{N-1}^{(*)}(x_{N-1})\}$ is optimal for this subproblem.

Proof: Refer to Bertsekas ([6] p. 18). ■

Intuitively, if it was possible to obtain a lower cost policy for the subproblem of planning from x_i, then the policy $\Pi^{(*)}$, which passes through x_i, can also be improved and was not optimal. Thus, all subproblems of the optimization must be optimal as well. This principle is used in DP to first solve the 'tail end' of the problem, and then to sequentially extend the optimal policy using a backward recursion. The DP algorithm is stated as:

Proposition 2.2: *Dynamic Programming*

For every initial state x_0, the optimal cost $C^{(*)}(x_0)$ for the Feedback Control problem with Perfect State Information (Definition 2.1) in Eq. (2.13) is equal to $J_0(x_0)$, obtained from the final step of backwards-recursive algorithm:

$$J_N(x_N) = l_N(x_N) \tag{2.7}$$

$$J_k(x_k) = \min_{u_k \in U_k(x_k)} E_{\omega_k}\left[l_k(x_k, u_k, \omega_k) + J_{k+1}(f(x_k, u_k, \omega_k))\right]. \tag{2.8}$$

The expectation is taken with respect to the conditional probability distribution $p(\omega_k|x_k, u_k)$. Furthermore, if $u_k^{(*)} = \pi_k^{(*)}(x_k)$ minimizes the right-hand side of Eq. (2.8), for $k = 0, \ldots, N-1$, then the policy $\Pi^{(*)} = \{\pi_0^{(*)}, \ldots, \pi_{N-1}^{(*)}\}$ is optimal.

Proof: This can be shown by induction ([6] p. 23). ■

2.1.3 Dynamic Programming with Perfect State Information

Consider the feedback control problem of Definition 2.1. According to Proposition 2.2, the optimal control problem can be solved using the Dynamic Programming with *Perfect State Information* (DP PSI) algorithm.

Definition 2.2: *Dynamic Programming with Perfect State Information*

Consider a discrete-time dynamic equation, Eq. (2.2), with the states, controls, and disturbances defined in Section 2.1.1. Assume a class of policies that map the states into control actions, Eq. (2.3), and a stage-additive cost functional, Eq. (2.4).

The DP algorithm in Proposition 2.2 is used to solve this optimization problem by using the backwards recursion of Eq. (2.7) and Eq. (2.8).

2.1.4 Planning with Imperfect State Information

In most practical applications, the system state is not directly observable, but must be inferred from noisy measurements. This is what is known as the *imperfect state information* (ISI) problem. The ISI problem can be cast as a PSI problem by using an information space as the state space. For the PSI problem, a class of control policies that map the (known) state, x_k, into a control was considered. Now, this state is unknown, but noisy measurements are available and can be used instead to help define the controls. In this section, the information state and space are defined, the belief state is introduced, and the DP problem is formulated for the case of ISI. First, the feedback control problem with imperfect state information (FC ISI) is defined ([6], p. 218, and [35], p. 523).

The state, control, and disturbance was defined in Section 2.1.1. The measurement, $y_k \in \mathbb{Y}(x_k)$, is defined to be in the measurement space, $\mathbb{Y}(x_k) \in \mathbb{R}^{n_y}$, and is corrupted by measurement noise, $\nu_k(x_k) \in \mathbb{V}(x_k)$, which is described by the conditional distribution

$$\nu_k(x_k) \sim p(\nu_k | x_k). \tag{2.9}$$

Definition 2.3: *Feedback Control Problem with Imperfect State Information*

Consider the feedback control problem with perfect state information (Definition 2.1) where the dynamics of the system are governed by Eq. (2.2). However, the state, x_k, is not directly measurable. A sensor mapping, $h : \mathbb{X} \times \mathbb{V} \to \mathbb{Y}$, is C^2 and maps every state into a measurement:

$$y_k = h(x_k, \nu_k). \tag{2.10}$$

A class of control policies, $\Pi = \{\pi_0(I_0), \ldots, \pi_{N-1}(I_{N-1})\}$, that maps at each stage the information available to the controller, I_k, to a control, $u_k = \pi_k(I_k)$, is used. This information, I_k, is formally defined below.

A stage-additive cost function, similar to Eq. (2.4), is defined in terms of the control policy:

$$L(x_{0:N}, \Pi, \omega_{0:N-1}) = l_N(x_N) + \sum_{k=0}^{N-1} l_k(x_k, \pi_k(I_k), \omega_k). \tag{2.11}$$

The expected cost for a control policy can be calculated:

$$C_\Pi(I_k) = E\left[l_N(x_N) + \sum_{k=0}^{N-1} l_k(x_k, \pi_k(I_k), \omega_k)\right]. \tag{2.12}$$

The expectation is taken with respect to the joint distribution: $p(x_0, \omega_{0:N-1}, \nu_{1:N})$. The optimal policy, $\Pi^{(*)} = \{\pi_0^{(*)}(I_0), \ldots, \pi_{N-1}^{(*)}(I_{N-1})\}$, minimizes the expected cost over the set of admissible policies, $\widetilde{\Pi}$:

$$\begin{aligned}\Pi^{(*)} &= \arg\min_{\Pi \in \widetilde{\Pi}} C_\Pi(x_0) \\ &= \arg\min_{\Pi \in \widetilde{\Pi}} E_{\omega_{0:N-1}}\left[l_N(x_N) + \sum_{k=0}^{N-1} l_k(x_k, \pi_k(I_k), \omega_k)\right].\end{aligned}$$

The optimal cost is given by

$$C^{(*)}(x_0) = E_{\omega_{0:N-1}}\left[l_N(x_N) + \sum_{k=0}^{N-1} l_k(x_k, \pi_k^{(*)}, \omega_k)\right]. \tag{2.13}$$

2.1.4.1 Information Spaces and States

Information states summarize the information available to the system at a given stage. For brevity, information state and space will be abbreviated as I-state and I-space, respectively.

Definition 2.4: *Information State, Space, and State Transition Function*

The I-state*, I_k, captures all the information available to controller at stage k and is a member of the I-space $\mathbb{I}_k$. An* I-state transition function*, which describes the evolution of the I-states, is defined as:*

$$I_{k+1} = g(I_k, u_k, y_{k+1})$$

where $g : \mathbb{I}_k \times \mathbb{U} \times \mathbb{Y} \to \mathbb{I}_{k+1}$ is C^2.

It is convenient to think of the I-space as just another state space, albeit a larger one. The states in this space are the I-states and are known (similar to the PSI problem). Different I-states can be defined. Let the *history of measurements* be $y_{0:k} \triangleq \{y_0, y_1, \ldots, y_k\}$ and the *history of controls* be $u_{0:k-1} \triangleq \{u_0, u_1, \ldots, u_{k-1}\}$. At the k^{th} stage, the information that is available is the history of controls and the history of measurements. Furthermore, knowledge of the initial state of the system is assumed. The history I-state is the most basic and intuitive I-state, summarizing the information available to the system at stage k.

Definition 2.5: *History Information State and Space*

The history information state *at stage k, η_k, is defined as the combination of the control- and measurement histories and the initial I-state, η_0:*

$$\eta_k = \{\eta_0, u_{0:k-1}, y_{0:k}\}. \tag{2.14}$$

The history I-space at stage k, $\mathbb{I}_k$, *is defined as the set of all possible history I-states at stage k. This is defined in terms of the sets of measurement histories: the Cartesian product of k copies of the measurement space, $\widetilde{\mathbb{Y}}_k = \mathbb{Y} \times \mathbb{Y} \ldots \times \mathbb{Y}$. The set of control histories is similarly defined as the Cartesian product of $k-1$ copies of the control space. Let $\mathbb{I}_0$ be the I-space of the initial I-state, η_0. Then, the history I-space at stage k is defined as: $\mathbb{I}_k = \mathbb{I}_0 \times \widetilde{\mathbb{U}}_{k-1} \times \widetilde{\mathbb{Y}}_k$. The* history I-space *is defined as the union of the stage I-spaces: $\mathbb{I}_{hist} = \mathbb{I}_0 \cup \mathbb{I}_1 \cup \ldots \cup \mathbb{I}_k$.*

In order to derive the I-state transition function for the history I-state, consider the following thought experiment: Suppose that there are no sensors (and thus no measurements), then the history I-states at the next stage is predictable when u_k is fixed: the control is concatenated to the previous history I-state, $\eta_{k+1} = g_{hist}(\eta_k, u_k) = \{\eta_k, u_k\}$. Now suppose that there are measurements available, which are unpredictable due to measurement noise. In Section 2.1.1, ω_k is used to model the unpredictability of state x_{k+1}. In terms of the history I-state, y_{k+1} serves the same purpose: the outcome, η_{k+1}, is unpredictable because y_{k+1} is random. However, once the control is chosen and applied, the measurement y_{k+1} is obtained and the history I-state η_{k+1} becomes known at stage $k+1$. The history I-state transition function is the concatenation of u_k and y_{k+1} to the previous I-state.

$$\eta_{k+1} = g_{hist}(\eta_k, u_k, y_{k+1}) = \{\eta_k, u_k, y_{k+1}\} \tag{2.15}$$

Remark 2.1: *The disturbance in the PSI problem (Definition 2.1) depends only on the current state and the control, (see Eq. (2.1)) for the derivation of the dynamic programming algorithm to hold, and not on the previous disturbances,* $\omega_{0:k-1}$. *Here,* y_{k+1} assumes the role of the disturbance in the I-space. *For* y_{k+1} *to correspond to the disturbance in the Definition 2.1, it is necessary that this measurement depend only on the current I-state and the control and not on the previous measurements,* $y_{k+1} \sim p(y_{k+1}|\eta_k, u_k)$. *Per definition, this is true since* $y_{0:k}$ *is imbedded in* η_k *and* $p(y_{k+1}|\eta_k, u_k, y_{0:k}) = p(y_{k+1}|\eta_k, u_k)$. *The distribution of* y_{k+1} *explicitly depends only on the current I-state and the control, as required.*

2.1.4.2 Derived Information States and Spaces

One disadvantage of the history I-state is that the length of the sequence defining the I-state grows at each stage. To overcome this problem, it is convenient to map the history I-states into a simpler space, called a *derived I-space*, $\mathbb{I}_{der}$. This mapping, $\kappa : \mathbb{I}_{hist} \rightarrow \mathbb{I}_{der}$, is known as an *information mapping*, or I-map. Let the *derived I-state be*:

$$\zeta_k \triangleq \kappa(\eta_k) \in \mathbb{I}_{der}.$$

With the current definition, the full history I-state is necessary to define the derived I-state. However, it will be convenient to discard the histories and work in the derived space entirely. Let the I-state transition function in the derived I-space be $\zeta_{k+1} = g_{der}(\zeta_k, u_k, y_{k+1})$. For an I-map to be *sufficient*, the following two calculations must be equivalent:

$$\kappa(g_{hist}(\eta_k, u_k, y_{k+1})) \iff g_{der}(\kappa(\eta_k), u_k, y_{k+1}).$$

Stated more clearly, the following sequence of mapping are equivalent:

$$: \eta \overset{g_{hist}}{\longrightarrow} \eta_{k+1} \text{ then } \eta_{k+1} \overset{\kappa}{\longrightarrow} \zeta_{k+1} \iff \eta_k \overset{\kappa}{\longrightarrow} \zeta_k \text{ then } \zeta_k \overset{g_{der}}{\longrightarrow} \zeta_{k+1}.$$

When the I-map is sufficient, there is no need to store the history I-space. The derived I-space of interest in this work is the *belief space* (or probabilistic I-space).

2.1.4.3 Belief State and Space

The belief space is derived from the history I-space with a sufficient I-map ([35], p. 474). A Markov probabilistic model is assumed: the past and future data are only related through the current system state. In other words, the past states, measurements, and controls contain no information that will help predict the evolution of the system. The current state is the best predictor of the future states. See Thrun et al. ([53] pp. 21 and 33) for a detailed discussion on the Markov assumption. The measurement, y_{k+1}, assumes the role of the disturbance in the I-space (Remark 2.1) and depends on the current I-state and the control and not the previous states or controls.

Definition 2.6: *Belief State, or Probabilistic Information State*

The probabilistic I-map, κ_{prob}, is used to define the belief state, ζ_k:

$$\zeta_k = \kappa_{prob}(\eta_k) \triangleq p(x_k|\eta_k). \tag{2.16}$$

The belief state is the distribution of the state, conditioned on the information available stage k.

Bayes' rule is used to determine the belief state transition function, $\zeta_{k+1} = g_{prob}(\zeta_k, u_k, y_{k+1})$:

$$\begin{aligned} \zeta_{k+1} &= p(x_{k+1}|\eta_{k+1}) \\ &= p(x_{k+1}|\eta_k, u_k, y_{k+1}) \\ &= \frac{p(y_{k+1}|x_{k+1}, \eta_k, u_k)p(x_{k+1}|\eta_k, u_k)}{p(y_{k+1}|\eta_k, u_k)}. \end{aligned}$$

Note that $p(y_{k+1}|x_{k+1}, \eta_k, u_k) = p(y_{k+1}|x_{k+1})$ since the I-state, η_k, does not give any additional information about y_{k+1} once the state x_{k+1} is given (Markov assumption). Also,

$$\begin{aligned} p(x_{k+1}|\eta_k, u_k) &= \int_{x_k \in X_k} p(x_{k+1}|x_k, \eta_k, u_k)p(x_k|\eta_k, u_k)dx_k \\ &= \int_{x_k \in X_k} p(x_{k+1}|x_k, u_k)\zeta_k dx_k \end{aligned}$$

which is obtained through marginalization, and using the fact that η_k does not contain additional information regarding the prediction of x_{k+1} once x_k is given (Markov assumption). Finally, the law of total probability is used to obtain an expression for $p(y_{k+1}|\eta_k, u_k)$:

$$p(y_{k+1}|\eta_k, u_k) = \int_{x_{k+1} \in X_{k+1}} p(y_{k+1}|x_{k+1}, u_k)p(x_{k+1}|\eta_k, u_k)dx_{k+1}.$$

Table 2.1: Converting the ISI problem to a PSI problem

ISI variable	PSI variable	Description
ζ_k	x_k	State is the probabilistic I-state
$\mathbb{I}_{prob}$	$\mathbb{X}$	State space is the probabilistic I-space
u_k	u_k	Control remains the same
$\mathbb{U}$	$\mathbb{U}$	Control space remains the same
y_{k+1}	ω_k	Measurement at $k+1$ is the disturbance
$\mathbb{Y}$	$\mathbb{W}(x_k, u_k)$	Measurement space is the disturbance space
$g_{prob}(\zeta_k, u_k, y_{k+1})$	$f(x_k, u_k, \omega_k)$	Information state transition function
η_0	x_0	Initial state is the initial I-state
$L^{ISI}(\zeta_{0:N}, u_{0:N-1}, y_{1:N})$	$L(x_k, u_k, \omega_k)$	Cost functional is derived from $L(x_k, u_k, \omega_k)$
$l_k^{ISI}(\zeta_k, u_k, y_{k+1})$	$l_k(x_k, u_k, \omega_k)$	Stage cost derived from $l_k(x_k, u_k, \omega_k)$
$l_N^{ISI}(\zeta_N)$	$l_N(x_N)$	Terminal cost derived from $l_N(x_N)$

The denominator is independent of x_{k+1}, and can be treated as a normalizing factor, β. The belief state transition function becomes:

$$g_{prob}(\zeta_k, u_k, y_{k+1}) \quad = \quad \beta\, p(y_{k+1}|x_{k+1}, u_k) \int_{x_k \in X_k} p(x_{k+1}|x_k, u_k)\zeta_k dx_k. \tag{2.17}$$

2.1.4.4 Converting ISI to PSI

The ISI planning problem can now be reformulated as a PSI planning problem in terms of the belief states. To make this explicit, consider the definitions in Table 2.1 where equivalent PSI variables are defined for the ISI variables. The measurement at the next stage assumes the role of the disturbance in the system, since it is assumed that the belief state is known. Thus, this future measurement is the only unknown variable.

The cost function in Eq. (2.11), $L(x_k, u_k, \omega_k)$, must be converted into an equivalent cost function in terms of the belief states, $L^{ISI}(\zeta_k, u_k, y_{k+1})$. At each stage, the stage cost is the expected value of $l_k(x_k, u_k, \omega_k)$, conditioned on all the information available at that stage (i.e., the history I-state):

$$\begin{aligned} l_k^{ISI}(\zeta_k, u_k, y_{k+1}) &= E_{x_k, \omega_k}\left[l_k(x_k, u_k, \omega_k)|\eta_k\right] \\ &= l_k^{ISI}(\zeta_k, u_k). \end{aligned}$$

The expectation is taken with respect to the conditional distribution $p(x_k, \omega_k|\eta_k)$. The terminal cost for the reformulated problem is:

$$l_N^{ISI}(\zeta_N) = E_{x_N}\left[l_N(x_N)|\eta_N\right].$$

The expectation is taken with respect to the conditional distribution $p(x_N|\eta_N)$. The cost function for the reformulated problem becomes:

$$L^{ISI}(\zeta_{0:N}, u_{0:N-1}, y_{1:N}) = l_N^{ISI}(\zeta_N) + \sum_{k=0}^{N-1} l_k^{ISI}(\zeta_k, u_k). \tag{2.18}$$

The expected cost in the belief space is given by:

$$C^{ISI}(\zeta_0) = E_{y_{1:N}} \left[l_N^{ISI}(\zeta_N) + \sum_{k=0}^{N-1} l_k^{ISI}(\zeta_k, u_k) \right]. \tag{2.19}$$

2.1.5 Dynamic Programming with Imperfect State Information

The dynamic programming problem for the imperfect state information case (DP ISI) is formulated. Consider the FC ISI problem of Definition 2.3.

Definition 2.7: *Dynamic Programming with Imperfect State Information*

Consider a discrete-time dynamic equation, Eq. (2.2), with the states, controls, and disturbances defined in Section 2.1.1. Consider a measurement model, Eq. (2.10) with the measurement and measurement noise defined in Section 2.1.4. The problem is solved in the belief space, using the belief states defined in Definition 2.6 and the history I-state defined in Definition 2.5. The belief state evolves according to

$$\zeta_{k+1} = g_{prob}\left(\zeta_k, u_k, y_{k+1}\right) \tag{2.20}$$

with the state transition function defined in Eq. (2.17). Consider a class of control policies that map the belief states into control actions:

$$\Pi = [\pi_0(\zeta_0) \; \dots \; \pi_{N-1}(\zeta_{N-1})] \tag{2.21}$$

and a stage-additive cost function in terms of the belief states, $L^{ISI}(\zeta_{0:N}, u_{0:N-1}, y_{1:N})$ *(defined in Eq. (2.18)). The dynamic programming algorithm in Proposition 2.2 is used to solve this optimization problem by using the backwards recursion of Eq. (2.7) and Eq. (2.8) in terms of the belief states:*

$$J_N(\zeta_N) = l_N^{ISI}(\zeta_N) \tag{2.22}$$

$$J_k(\zeta_k) \quad = \quad \min_{\pi_k} l_k^{ISI}(\zeta_k, u_k) + E_{y_{k+1}}\left[J_{k+1}(g_{prob}(\zeta_k, u_k, y_{k+1}))|\eta_k\right]. \tag{2.23}$$

2.2 Receding Horizon Control

It is often necessary to impose constraints on the states and the controls of the system. In general, these constraints are described as non-linear inequality functions ($c(x_{k:M}, u_{k:M-1}) \leq 0$). *Receding* *horizon contro*l (RHC), also known as *Model Predictive Control*, is a suboptimal control scheme which is motivated by the need to handle nonlinear systems and to incorporate state and control constraints explicitly in the planning problem [42, 43]. The problem is solved from the current stage, k, on a shorter (finite) horizon, $M \leq N$, and a portion of the plan is executed before new measurements are obtained, the system states are updated, and the problem is re-solved. The approach was originally developed for deterministic systems.

Consider the feedback control problem of Definition 2.1, with the states and controls defined in Section 2.1.1, but without disturbances. Let the deterministic, nonlinear dynamic equation of the system be $x_{i+1} = f(x_i, u_i)$. The cost function of Eq. (2.4) without the disturbances is to be used and the states and controls are to be constrained.

Definition 2.8: *Basic Receding Horizon Control*

Find the sequence of control actions*, $u_{k:M-1}$, that minimizes the cost, satisfies the system dynamics, and satisfies the constraints of the problem:*

$$\begin{aligned} \min_{u_{k:M} \in U_C} \quad & \sum_{i=k}^{M-1} l_i(x_i, u_i) + l_M(x_M) \\ s.t. \quad & x_{i+1} = f(x_i, u_i) \\ & c(x_{k:M}, u_{k:M-1}) \leq 0. \end{aligned}$$

Note that the problem is changed from a *feedback* solution for all possible states that the system can reach, to finding a *control sequence* over a finite horizon that is executed from the current state. Extensions of the basic RHC to stochastic systems are described in Section 3.4.

2.3 Chance Constraints

It is often necessary to impose constraints on the states of the system when solving a motion planning problem. In general, these constraints are described as either linear ($Ax \leq b$) or non-linear ($c(x) \leq 0$) inequality functions. For these constraints, it is assumed that the state is perfectly known. However, in many applications the state of the system consists of stochastic variables with unbounded distributions (e.g., normal distributions). The constraints in the above form cannot be evaluated since there is no guarantee that the constraint can be met for all possible realizations of the states. It is instead necessary to introduce *chance constraints*, or stochastic state constraints, which are of the form $P(x \notin \mathbb{X}_{free}) \leq 1 - \alpha$ or $P(x \in \mathbb{X}_{free}) \geq \alpha$. α is the *level of confidence* and $\mathbb{X}_{free}$ denotes the set of states where the original constraints are satisfied, for example $\mathbb{X}_{free} = \{x : Ax \leq b\}$. The constraints are specified as limits on the probability of constraint violation.

There are generally two approaches to evaluating chance constraints: (i) assume (or approximate) jointly Gaussian distributed state variables and convert the chance constraints into constraints on the means of the states (e.g., [10, 59]), and (ii) evaluate the constraints by Monte Carlo simulation (the state distribution is sampled and the number of samples that violate the deterministic constraint are counted, from which the probability of constraint violation can be approximated (e.g., [8, 12])). Approach (i) is investigated in this work.

Two types of chance constraints are considered: (i) linear constraints: $P(Ax \leq b) \geq \alpha$ (e.g., velocity constraints) and (ii) collision constraints (e.g., between the robot and other agents). Standard results for linear constraints using approach (i) above are presented in this section.

2.3.1 Single Linear Chance Constraints

For jointly Gaussian distributed state variables, the linear chance constraints are relatively easy to evaluate since the probability function is scalar (for single linear constraints). Let the linear chance constraint be of the form $P(a^T x \leq b) \geq \alpha$, where $a^T x \in \mathbb{R}$ and $b \in \mathbb{R}$. Let $\hat{x} \triangleq E[x]$ be the mean of the state, and the covariance matrix of x be $\Sigma \triangleq E[(x - \hat{x})(x - \hat{x})^T]$.

Lemma 2.1: *The chance constraint $P(a^T x \leq b) \geq \alpha$ is satisfied iff*

$$a^T \hat{x} + \mathrm{icdf}(\alpha) \times \sqrt{a^T \Sigma a} \leq b$$

where $\mathrm{icdf}(\alpha)$ *is the inverse of the cumulative distribution function for a* standard *scalar Gaussian variable with zero mean and unit standard deviation.*

Proof: Let $\xi \triangleq a^T x$. Since x is assumed to be jointly Gaussian, the new variable is normally distributed $\xi \sim N(\xi; a^T \hat{x}, a^T \Sigma a)$. Convert this to a standard Gaussian variable by introducing $\xi_n \triangleq \frac{\xi - a^T \hat{x}}{\sqrt{a^T \Sigma a}}$, which has a zero mean and unit variance. Then

$$\begin{aligned} a^T x \leq b \quad &\Longleftrightarrow \quad \xi \leq b \\ &\Longleftrightarrow \quad \frac{\xi - a^T \hat{x}}{\sqrt{a^T \Sigma a}} \leq \frac{b - a^T \hat{x}}{\sqrt{a^T \Sigma a}} \\ &\Longleftrightarrow \quad \xi_n \leq \frac{b - a^T \hat{x}}{\sqrt{a^T \Sigma a}} \\ &\Longleftrightarrow \quad \xi_n \leq b_n \end{aligned}$$

where $b_n \triangleq \frac{b - a^T \hat{x}}{\sqrt{a^T \Sigma a}}$. But $P(\xi_n \leq b_n) = \mathrm{cdf}(b_n)$, where $\mathrm{cdf}(b_n)$ is the cumulative distribution function for a standard Gaussian random variable. For the chance constraint to be satisfied:

$$\begin{aligned} P(a^T x \leq b) \geq \alpha \quad &\Longleftrightarrow \quad \mathrm{cdf}(b_n) \geq \alpha \\ &\Longleftrightarrow \quad b_n \geq \mathrm{icdf}(\alpha) \end{aligned}$$

Table 2.2: Level of certainty versus inverse cumulative distribution function values

α	0.900	0.950	0.990	0.999
$F^{-1}(\alpha)$	1.2816	1.6449	2.3263	3.0902

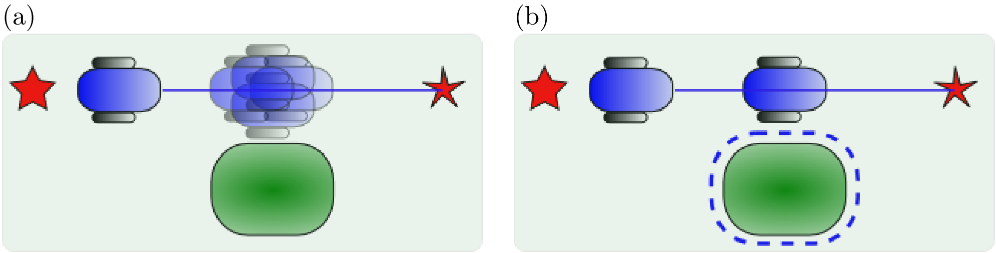


Figure 2.1: (a) The robot (blue) is propagated, resulting in an uncertain state (represented by multiple transparent robots). The static obstacle (green) is to be avoided. (b) The constraint is tightened (dashed blue) by some amount that accounts for the level of confidence and the predicted state covariance.

where icdf(α) is the inverse of the cumulative distribution function. Substituting for b_n, the result is obtained. ■

The chance constraint is converted to a constraint of the mean of the robot and is a function of the covariance of the robot state. Effectively, the constraint is tightened by some amount that is a function of the level of confidence and covariance. This is illustrated in Figure 2.1. Refer to Table 2.3.1 for example values of icdf(α) for different values of α. Note that it is necessary for $\alpha \geq \frac{1}{2}$ in the above analysis for icdf$(\alpha) \geq 0$ (see [56])

2.3.2 Sets of Linear Chance Constraints

Consider next the problem of imposing a set of m chance constraints on the states. When formulating the problem, the designer can either impose the individual constraints: $P(a_i^T x \leq b_i) \geq \alpha_i \ \forall i = 1, \ldots, m$ (the certainty level for each constraint is specified), or impose the constraints jointly as a set: $P(a_i^T x \leq b_i \ \forall i = 1, \ldots, m) \geq \alpha$ (the total certainty level is specified) [56]. Both formulations are useful in robotics motion planning. For example, consider a single state system with the constraint that the state should not exceed a given value at any stage. The designer can either enforce the constraint at each planning cycle with certainty level α_{ind}, or for the whole trajectory, α_{traj}. For the former, an individual chance constraint is enforced, and the probability of constraint violation over the trajectory is the sum of the probability of collision at each planning cycle, $\alpha_{tot} = N\alpha_{ind}$. For the latter, a joint chance constraint is imposed, with the state redefined as the trajectory, $x_{0:N}$, $m = N$, and $\alpha_{tot} = \alpha_{joint}$.

To impose the set of constraints, it is convenient to formulate the problem in terms of the probability of not being in free space: $P(x \notin \mathbb{X}_{free}) = 1 - P(x \in \mathbb{X}_{free}) \leq 1 - \alpha = \delta$. A part of δ is normally assigned to each of the individual constraints that are to be satisfied [**?**, 10, 59]. Define the event $a_i^T x \leq b_i \triangleq A_i$, then $x \in \mathbb{X}_{free} \Leftrightarrow \bigcap_{i=1}^m A_i$. The negation of this expression (represented by the overbar) is given by De Morgan's law:

$$\overline{\bigcap_{i=1}^{m} A_i} = \bigcup_{i=1}^{m} \overline{A_i}.$$

From Boole's inequality:

$$P(\bigcup_{i=1}^{m} \overline{A_i}) \leq \sum_{i=1}^{m} P(\overline{A_i}).$$

Thus, if $P(\overline{A_i}) \leq \epsilon_i$ for each i, then:

$$\sum_{i=1}^{m} \epsilon_i = \delta \implies P(a_i^T x \leq b_i \; \forall i = 1, \ldots, m) \geq \alpha.$$

It now becomes necessary to specify the level of certainty, ϵ_i, for each of the individual constraints. The first approach is to assign a constant value, $\epsilon_i = \frac{\delta}{m}$, but this has been shown to be very conservative [**?**]. An alternative approach is to do 'risk allocation': the certainty level for a specific constraint is chosen based on how relevant that constraint is to the overall chance constraint satisfaction. Ono and Williams [44] solved the path planning problem with risk allocation iteratively, and significantly less conservative solutions are obtainable. Blackmore [**?**] extended this approach by formulating the risk allocation problem as part of the path planning problem.

2.4 Recursive Estimation Theory

In robot systems, not all the states and parameters are known or directly measurable. These states and parameters often need to be inferred from the measurements, which are noisy realizations of the internal state. Additionally, process noise makes the outcome of the dynamic evolution of the system uncertain. Estimation theory deals with maintaining an estimate of the states and the uncertainty associated with the states, given the system dynamic and measurement equations and the properties of the noise processes.

Definition 2.9: *State Estimation*

Let the state, $x_k \in \mathbb{X}$, be an element of the state space, *$\mathbb{X} \subseteq \mathbb{R}^{n_x}$. The control $u_k(x_k) \in \mathbb{U}(x_k)$, is an element of the* action space, *$\mathbb{U}(x_k) \subseteq \mathbb{R}^{n_u}$. The disturbance, $\omega_k(x_k, u_k) \in \mathbb{W}(x_k, u_k)$, is described by the conditional distribution $\omega_k(x_k, u_k) \sim p(\omega_k | x_k, u_k)$ and is an element of the set $\mathbb{W}(x_k, u_k) \in \mathbb{R}^{n_\omega}$. The disturbance does not depend on the previous disturbances, $\omega_{0:k-1}$. Consider* *the following discrete-time dynamic system:*

$$x_{k+1} = f\left(x_k, u_k, \omega_k\right) \tag{2.24}$$

where $f : \mathbb{X} \times \mathbb{U} \times \mathbb{W} \rightarrow \mathbb{X}$ *is the* state transition function *which is assumed to be* C^2*.*

The measurement, $y_k \in \mathbb{Y}(x_k)$*, is defined to be in the* measurement space, $\mathbb{Y}(x_k) \in \mathbb{R}^{n_y}$*. The measurement noise,* $\nu_k(x_k) \in \mathbb{V}(x_k)$*, is described by the conditional distribution* $\nu_k(x_k) \sim p(\nu_k|x_k)$*. A sensor mapping,* $h : \mathbb{X} \times \mathbb{V} \rightarrow \mathbb{Y}$*, is* C^2 *and maps every state into a measurement through the measurement equation:*

$$y_k = h(x_k, \nu_k). \tag{2.25}$$

Denote the history of measurements by $y_{1:k}$*, and the history of controls by* $u_{0:k-1}$*.*

The distribution of the state conditioned on the measurement history, $p(x_k|y_{1:k}, u_{0:k-1})$*, is the desired quantity in the estimation process.*

It is assumed that the state is *complete* (Markov assumption): the current state is the best predictor of the future states of the system. In other words, past states, measurements, and controls contain no additional information to help predict the future states of the system [53]. This assumption allows for the estimation process to be defined recursively. The Bayes filter is the most general form of the recursive estimator. Alternatively, the recursive estimation problem can be formulated as a variance minimization problem.

Remark 2.2: *Some notation is necessary to distinguish the expectations and covariances associated with the different distributions of interest in recursive state estimation. Let* $\hat{x}_{m|n} \triangleq E\left[x_m|y_{1:n}\right]$ *be the expectation of the state at stage* m *conditioned on the history of measurements up to stage* n*. Similarly, let* $\Sigma_{m|n} \triangleq E\left[(x_m - \hat{x}_{m|n})(x_m - \hat{x}_{m|n})^T|y_{1:n}\right]$ *be the covariance of the state at stage* m*, conditioned on the history of measurements up to stage* n*.*

2.4.1 Bayes Filter

The objective is to calculate the posterior probability distribution, $p(x_k|y_{1:k}, u_{0:k-1})$. Using Bayes' rule, this can be written as:

$$p(x_k|y_{1:k}, u_{0:k-1}) = \frac{p(y_k|x_k, y_{1:k-1}, u_{0:k-1})p(x_k|y_{1:k-1}, u_{0:k-1})}{p(y_k|y_{1:k-1}, u_{0:k-1})}.$$

$p(y_k|x_k, y_{1:k-1}, u_{0:k-1})$ is known as the *measurement probability* and it is a function of the measurement equation, Eq. (2.25). The state is assumed to be complete, and this distribution can be written as $p(y_k|x_k, y_{1:k-1}, u_{0:k-1}) = p(y_k|x_k)$ since the past measurements contain no additional information that is not already contained in the state (which is the best predictor). To write this in recursive form, use is made of the marginalization theorem:

$$p(x_k|y_{1:k-1}, u_{0:k-1}) = \int_{x_{k-1}} p(x_k|x_{k-1}, y_{1:k-1}, u_{0:k-1}) p(x_{k-1}|y_{1:k-1}, u_{0:k-1}) dx_{k-1}$$

where $p(x_k|x_{k-1}, y_{1:k-1}, u_{0:k-1})$ is known as the *state transition probability* and is a function of the system dynamic equation, Eq. (2.24). Due to the Markov assumption, $p(x_k|x_{k-1}, y_{1:k-1}, u_{0:k-1}) = p(x_k|x_{k-1}, u_{k-1})$. $p(x_{k-1}|y_{1:k-1}, u_{0:k-1})$ is the state estimate at stage $k-1$, which is assumed to be known. Note also that

$$\begin{aligned} p(y_k|y_{1:k-1}, u_{0:k-1}) &= \int_{x_k} p(y_k|x_k, y_{1:k-1}, u_{0:k-1}) p(x_k|y_{1:k-1}, u_{0:k-1}) dx_k \\ &= \int_{x_k} p(y_k|x_k) p(x_k|y_{1:k-1}, u_{0:k-1}) dx_k \\ &\triangleq \beta \end{aligned}$$

since this distribution is independent of the state and is just a normalizing factor.

The filter has two parts:

1. The *prediction step*, where the state transition probability is calculated (this is a prediction of the outcome of the dynamic evolution of the system without the new measurement):

$$p(x_k|y_{1:k-1}, u_{0:k-1}) = \int_{x_{k-1}} p(x_k|x_{k-1}, u_{k-1}) p(x_{k-1}|y_{1:k-1}, u_{0:k-1}) dx_{k-1}.$$

2. The *measurement update step* incorporates the latest measurement:

$$p(x_k|y_{1:k}, u_{0:k-1}) = \beta \, p(y_k|x_k) p(x_k|y_{1:k-1}, u_{0:k-1}).$$

2.4.2 Variance Minimization

Kalman noted that the optimal estimate of a state is the expected value of that state, conditioned on the measurements up to that time, $E[x_k|y_{1:k}]$. This estimate is obtained by minimizing the least squares error problem: given the past measurements, find the estimate that satisfies the system equations and minimizes the expectation of the square of the estimation error. Let the $\bar{x}_k$ be the estimate, then the estimation error is defined as $e_k \triangleq x_k - \bar{x}_k$.

Lemma 2.2: *The estimate that minimizes the the expected value of the square of the estimation error is the expectation of the state conditioned on the information in the system.*

PROOF: The objective is to find the estimate that minimizes $J = E[e_k^T e_k|y_{1:k}]$. Using the definition of the the estimation error, the cost function is written as:

$$\begin{aligned} J &= E\left[e_k^T e_k | y_{1:k}\right] \\ &= E\left[(x_k - \bar{x}_k)^T (x_k - \bar{x}_k) | y_{1:k}\right] \\ &= E\left[x_k^T x_k | y_{1:k}\right] - E\left[x_k^T | y_{1:k}\right] \bar{x}_k - \bar{x}_k^T E\left[x_k | y_{1:k}\right] + \bar{x}_k^T \bar{x}_k. \end{aligned}$$

The first derivative with respect to the estimate is:

$$\frac{\partial J}{\partial \bar{x}_k} = -2E\left[x_k | y_{1:k}\right] + 2\bar{x}_k$$

and the second derivative is

$$\frac{\partial^2 J}{\partial \bar{x}_k^2} = 2.$$

Setting the first derivative equal to zero, the optimal estimate is obtained: $\bar{x}_k^{(*)} = E\left[x_k | y_{1:k}\right]$. ■

Using the properties of the trace of a matrix, $E[e_k^T e_k | y_{1:k}] = Tr(E[e_k e_k^T | y_{1:k}]) = Tr(\Sigma_{k|k})$, where Tr is the trace of the matrix. Thus, the optimal estimation problem can be thought of as selecting the estimate that minimizes the covariance of the estimation error. This general result is independent of the form of the system under investigation (i.e., nonlinear versus linear systems).

For systems with linear dynamics, a linear measurement equation, and with Gaussian noise, a closed-form solution of the conditional expectation and covariance is available since the variables are jointly Gaussian (see for example [13], p. 128). This is not generally the case for systems with non-linear equations and/or non-Gaussian noise.

2.4.3 Linear Gaussian Systems: Kalman Filter

Assume linear system dynamics and a linear measurement equation of the form:

$$x_k = A_{k-1} x_{k-1} + B_{k-1} u_{k-1} + F_{k-1} \omega_{k-1} \tag{2.26}$$

$$y_k = C_k x_k + H_k \nu_k \tag{2.27}$$

where $\omega_{k-1} \sim N(0, W)$ and $\nu_k \sim N(0, V)$ are independent Gaussian white noise terms. A_k, B_k, F_k, C_k, and H_k are known matrices. Assume that the posterior distribution at time $k-1$ is normal and known, $p(x_{k-1}|y_{1:k-1}) = N(x_{k-1}; \hat{x}_{k-1|k-1}, \Sigma_{k-1|k-1})$. Then, $p(x_k|y_{1:k}) = N(x_k; \hat{x}_{k|k}, \Sigma_{k|k})$ is also a normal distribution obtained from the Kalman Filter.

Definition 2.10: *Kalman Filter*

The prediction step of the Kalman Filter is given by:

$$\hat{x}_{k|k-1} = A_{k-1}\hat{x}_{k-1|k-1} + B_{k-1}u_{k-1} \tag{2.28}$$

$$\Sigma_{k|k-1} = A_{k-1}\Sigma_{k-1|k-1}A_{k-1}^T + F_{k-1}WF_{k-1}^T. \tag{2.29}$$

The measurement update step:

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k(y_k - \hat{y}_{k|k-1}) \tag{2.30}$$

$$\Sigma_{k|k} = (I - K_kC_k)\Sigma_{k|k-1} \tag{2.31}$$

where

$$\hat{y}_{k|k-1} = C_k\hat{x}_{k|k-1} \tag{2.32}$$

$$\Gamma_{k|k-1} = C_k\Sigma_{k|k-1}C_k^T + H_kVH_k^T \tag{2.33}$$

$$K_k = \Sigma_{k|k-1}C_k^T\Gamma_{k|k-1}^{-1}. \tag{2.34}$$

2.4.4 Nonlinear Gaussian Systems: Extended Kalman Filter

For systems with nonlinear model for the objects (robot or agents) of the form:

$$\begin{aligned} x_i &= f(x_{i-1}, u_{i-1}, \omega_{i-1}) \\ y_i &= h(x_i, \nu_i) \end{aligned}$$

where $\omega_{i-1} \sim N(0, W)$ is the white Gaussian disturbance that is independent of the previous process noise terms, $\omega_{0:i-2}$. $\nu_i \sim N(0, V)$ is the white Gaussian measurement noise that is independent of the previous noise terms, $\nu_{1:i-1}$.

The consequence of the nonlinear system equations is that the state and measurements are not jointly Gaussian after propagation through the system dynamics. Closed form solutions for the conditional expectation and covariance are not generally obtainable. Some approximate estimators have been developed, of which the Extended Kalman Filter is the most common nonlinear estimator. The system dynamic and measurement equations are linearized, and a Kalman Filter is derived for the linearized system. This approximates the resulting distributions as Gaussian.

The nonlinear dynamic equation is approximated with a first-order Taylor series expansion about the means $\hat{x}_{i-1|i-1}$ and $\hat{\omega}_{i-1} \triangleq E[\omega_{i-1}|x_{i-1}, u_{i-1}]$ (for non-zero mean process noise):

$$\begin{aligned} f(x_{i-1}, u_{i-1}, \omega_{i-1}) \quad &\approx \quad f(\hat{x}_{i-1|i-1}, u_{i-1}, \hat{\omega}_{i-1}) + \frac{\partial f}{\partial x_{i-1}} \,|_{\hat{x}_{i-1|i-1}, u_{i-1}, \hat{\omega}_{i-1}} (x_{i-1} - \hat{x}_{i-1|i-1}) \\ & \quad + \frac{\partial f}{\partial \omega_{i-1}} \,|_{\hat{x}_{i-1|i-1}, u_{i-1}, \hat{\omega}_{i-1}} (\omega_{i-1} - \hat{\omega}_{i-1}). \end{aligned}$$

The nonlinear measurement equation is approximated with a first-order Taylor series expansion about the means $\hat{x}_{i|i-1}$ and $\hat{\nu}_i \triangleq E[\nu_i|x_i]$:

$$h(x_i, \nu_i) \quad \approx \quad h(\hat{x}_{i|i-1}, \hat{\nu}_i) + \frac{\partial h}{\partial x_i} \,|_{\hat{x}_{i|i-1}, \hat{\nu}_i} (x_i - \hat{x}_{i|i-1}) + \frac{\partial h}{\partial \nu_i} \,|_{\hat{x}_{i|i-1}, \hat{\nu}_i} (\nu_i - \hat{\nu}_i).$$

Let

$$\begin{aligned} \widetilde{A}_{i-1} \quad &\triangleq \quad \frac{\partial f}{\partial x_{i-1}} \,|_{\hat{x}_{i-1|i-1}, u_{i-1}, \hat{\omega}_i} \quad \text{and} \quad \widetilde{\mathrm{F}}_{\mathrm{i-1}} \triangleq \frac{\partial \mathrm{f}}{\partial \omega_{\mathrm{i-1}}} \,|_{\hat{\mathrm{x}}_{\mathrm{i-1|i-1}}, \mathrm{u}_{\mathrm{i-1}}, \hat{\omega}_{\mathrm{i}}} \\ \widetilde{C}_i \quad &\triangleq \quad \frac{\partial h}{\partial x_i} \,|_{\hat{x}_{i|i-1}, \hat{\nu}_i} \quad \text{and} \quad \widetilde{H}_i \triangleq \frac{\partial h}{\partial \nu_i} \,|_{\hat{x}_{i|i-1}, \hat{\nu}_i} . \end{aligned}$$

Definition 2.11: *Extended Kalman Filter*

The prediction step of the Extended Kalman Filter is given by:

$$\begin{aligned} \hat{x}_{i|i-1} \quad &= \quad f(\hat{x}_{i-1|i-1}, u_{i-1}, \hat{\omega}_{i-1}) \\ \Sigma_{i|i-1} \quad &= \quad \widetilde{A}_{i-1} \Sigma_{i-1|i-1} \widetilde{A}_{i-1}^T + \widetilde{F}_{i-1} W \widetilde{F}_{i-1}^T. \end{aligned}$$

Measurement update step:

$$\begin{aligned} \hat{x}_{i|i} \quad &= \quad \hat{x}_{i|i-1} + K_i(y_i - h(\hat{x}_{i|i-1}, \hat{\nu}_i)) \\ \Sigma_{i|i} \quad &= \quad (I - K_i \widetilde{C}_i) \Sigma_{i|i-1} \end{aligned}$$

where

$$\begin{aligned} \Gamma_{i|i-1} \quad &= \quad \widetilde{C}_i \Sigma_{i|i-1} \widetilde{C}_i^T + \widetilde{H}_i V \widetilde{H}_i^T \\ K_i \quad &= \quad \Sigma_{i|i-1} \widetilde{C}_i^T \Gamma_{i|i-1}^{-1}. \end{aligned}$$

Chapter 3

Partially Closed-Loop Receding Horizon Controller

Planning in uncertain, dynamic environments requires the simultaneous solution of the planning and state estimation problems. Since the future system states are unknown, they must be predicted so that the quality of the proposed plan can be evaluated during the planning and execution process. The growth in state uncertainty during this prediction must be managed. Furthermore, there is a need to incorporate more complicated robot and agent behavioral models: models where the nature of the robot's planned path affects the quality of information gathered, and where other moving agents may have complex, path-dependent interactions with the robot.

A dynamic programming framework can address these problems, but is very cumbersome computationally and does not readily handle state, collision, and control constraints. In this chapter, the solution to the Dynamic Programming (DP) problem with Imperfect State Information (ISI) is outlined. Closed form solutions to this problem are not normally obtainable, and approximations to the DP ISI problem are presented. The Partially Closed-Loop Controller (PCLC) is introduced in this chapter as one approximate solution to the DP ISI problem. This approximation motivates the definition of the Partially Closed-Loop Receding Horizon Control (PCLRHC) formulation, which allows for constraints to be incorporated.

3.1 Constrained Feedback Control with Imperfect State Information

The problem of interest in this chapter is the feedback control problem with imperfect state information of Definition 2.3. Additionally, it is often necessary to impose constraints on the states and the controls of the system. In general, these constraints are described as non-linear inequality functions, $c(x_{k:N}, u_{k:N-1}) \leq 0$. Practical systems are stochastic due to disturbances and the states must be estimated from noisy measurements. Two types of stochastic systems are distinguished: systems where the noise terms are described in terms of set-membership, and probabilistic systems (noise terms are described by distributions). Probabilistic systems will be pursued exclusively in this work. Constraints on the states in the form $c(x_{k:N}, u_{k:N-1}) \leq 0$ cannot be evaluated exactly since there is no guarantee that the constraints can be satisfied for all possible realizations of the states. It is instead necessary to enforce *chance constraints*, or stochastic state constraints (Section 2.3).

Definition 3.1: *Constrained Feedback Control with Imperfect State Information*
Consider the feedback control problem with imperfect state information over a horizon of N stages, defined in Section 2.3. The dynamics of the system are governed by Eq. (2.2), where the disturbance is described by Eq. (2.1). The measurement equation is given in Eq. (2.10), with measurement noise described by Eq. (2.9).

A stage-additive cost function is defined in Eq. (2.11). A class of control policies that maps at each stage the information state to a control, $u_i = \pi_i(\eta_i)$, is used: $\Pi = \{\pi_k(\eta_k), \ldots, \pi_{N-1}(\eta_{N-1})\}$. The expected cost associated with a control policy Π is:

$$C_\Pi = E\left[l_N(x_N) + \sum_{i=k}^{N-1} l_i(x_i, \pi_i(I_i), \omega_i)\right] \tag{3.1}$$

where the expectation is taken with respect to the joint distribution: $p(x_k, \omega_{k:N-1}, \nu_{k:N})$.

Additionally, the controls are constrained at each stage by the set of n_j nonlinear inequality constraints $c_j(u_i) \leq 0\ \forall j = 1, \ldots, n_c$, where $c_j : \mathbb{U} \to \mathbb{R}$ is C^2 and the system states are constrained by the chance constraints $P(x_i \notin \mathbb{X}_{free}) \leq \delta$. $\mathbb{X}_{free} \subseteq \mathbb{R}^{n_x}$ is the subset of the state space where no collisions occur (refer to Section 2.3).

An optimal policy, $\Pi^{(*)} = \{\pi_k^{(*)}(\eta_k), \ldots, \pi_{N-1}^{(*)}(\eta_{N-1})\}$, minimizes C_Π over the set of admissible policies, $\widetilde{\Pi}$, while satisfying the state chance constraints and the control constraints:

$$\begin{aligned}\Pi^{(*)} = \arg\min_{\Pi \in \widetilde{\Pi}} \quad & E_{\omega_{0:N-1}}\left[l_N(x_N) + \sum_{i=k}^{N-1} l_i(x_i, \pi_i(\eta_i), \omega_i)\right] \\ s.t. \quad & c_j(u_i) \leq 0 \quad \forall i = k, \ldots, M-1,\ \forall j = 1, \ldots, n_c \\ & P(x_i \notin \mathbb{X}_{free}) \leq \delta \quad \forall i = k, \ldots, M.\end{aligned}$$

The optimal cost is given by

$$C^{(*)}(x_0) \quad = \quad E_{\omega_{0:N-1}}\left[l_N(x_N) + \sum_{k=0}^{N-1} l_k(x_k, \pi_k^{(*)}, \omega_k)\right]. \tag{3.2}$$

Section 2.1.4 showed that the unconstrained version of the above problem can be solved using dynamic programming. The solution and approximation approaches to the DP ISI problem are presented next and are the basis for Stochastic Receding Horizon Control, which allows for the constraints to be incorporated in the problem.

3.2 Dynamic Programming with Imperfect State Information

Dynamic programming is a general, powerful approach to the optimal feedback control problem. One shortcoming of the approach is its inability to directly handle constraints on the states and controls. It is very useful to understand the solution and approximations to the DP ISI problem, presented in Section 2.1.5. Consider the problem in Definition 2.1.4 (Definition 3.1 without the constraints).

3.2.1 Solutions to DP ISI

Though it is sometimes possible to obtain closed form solutions to the dynamic programming problem, such solutions tend to be an exception rather than the rule ([6] p. 282). The *curse of dimensionality* is a major culprit in problems with finite, discrete state-spaces, finite control action spaces, and finite measurement spaces. There is an exponential growth in the computational requirements as the sizes of these spaces are increased.

Because the DP ISI problem is based on probabilistic systems, the algorithm gives a feedback on belief space: the optimal control action is defined for every reachable belief state. The executed control sequence and realized measurement sequence determine the future belief states. With the DP ISI problem, *the optimal control action is defined for every possible combination of control sequence and measurement sequence*. For probabilistic systems, the set of possible measurements that can be obtained is infinite, resulting in a very complicated problem. Only a few problems allow closed form solution for the DP ISI problem, for example linear systems with quadratic cost and Gaussian noise terms [1, 6]. The solution in this case is detailed in Appendix A to gain insight into the problem formulation in the belief space. A key property of this specific system is that the control sequence does not affect the quality of estimation (separation principle in control theory). These results cannot generally be extended to non-linear systems or systems with non-Gaussian noise. Instead, one must resort to solving the problem approximately, and a number of approximate methods have been proposed. A selection of these methods are described next.

3.2.2 Approximation Methods

Since the DP ISI approach attempts to solve the feedback control problem (i.e., the control is defined for all possible belief states), it is generally very difficult to find solutions. Instead, the problem is approximated in one of two ways: (i) recursively solve a simplified problem for a control sequence

Algorithm 3.1 Certainty Equivalent Control Strategy

1. Compute the typical values $\bar{x}_k(\zeta_k)$, $\bar{\omega}_k(\bar{x}_k, u_k)$.
2. Find the control sequence, $\{u_k, \ldots, u_{N-1}\}$, that minimizes the deterministic cost function with the uncertain parameters fixed at their typical values:

$$\min_{u_{k:N-1}} \bar{l}_N(\bar{x}_N) + \sum_{i=k}^{N-1} l_i(\bar{x}_i, u_i, \bar{\omega}_i(\bar{x}_i, u_i)).$$

3. Apply the first control action, obtain a new measurement, and repeat the process.

(instead of a control policy) at each planning cycle, or (ii) solve for a control policy over a limited horizon and then approximate the cost-to-go function beyond this horizon. The feedback mechanism due to recursively re-solving the planning problem is referred to in this work as the *outerloop feedback mechanism.*

Methods that recursively solve a simplified problem for a control sequence are of interest in this work. Two common recursive approximations are the *Certainty Equivalent Control* strategy and the *Open-loop Control* strategy. These approaches are summarized here, and the reader is referred to Bertsekas [6] Chapter 6 for a more complete treatment of approximations for the DP ISI problem. The *Limited Lookahead Policies* strategy is an example of approach (ii) above (see [6], p. 304).

3.2.2.1 Certainty Equivalent Control Strategy

Certainty Equivalent Control is a suboptimal control scheme where the problem is solved in the regular state space (see Definition 2.1.1) and not in the belief space. The uncertain variables are fixed at some typical value (e.g., the expected values). The uncertainty in the variables is ignored and the problem becomes deterministic. All future information is ignored.

Let $\bar{x}_k$ be a typical value for the state, and assume that for every state-control pair a typical value for the process noise has been selected: $\bar{\omega}_k(\bar{x}_k, u_k)$. For example, the conditional expectations of the uncertain parameters can serve as typical values:

$$\begin{aligned} \bar{x}_k &= E[x_k|\eta_k] \\ \bar{\omega}_k &= E[\omega_k|\bar{x}_k, u_k]. \end{aligned}$$

The suboptimal control applied at the current time is generated recursively from the open-loop problem. Refer to Algorithm 3.1.

Algorithm 3.2 Open-Loop Control Strategy

1. Calculate the belief state at the current time: $\zeta_k = p(x_k|\eta_k)$.
2. Obtain the optimal control sequence, $\{u_k, \ldots, u_{N-1}\}$, from the following dynamic programming problem:

$$\bar{J}_i(\bar{\zeta}_i) = \min_{u_i} l_i^{ISI}(\bar{\zeta}_i, u_i) + \bar{J}_{i+1}(f_{\bar{\zeta}}(\bar{\zeta}_i, u_i)) \quad k \leq i \leq N-1$$

$$\bar{J}_N(\bar{\zeta}_N) = l_N^{ISI}(\bar{\zeta}_N)$$

   where $l_N^{ISI}(\bar{\zeta}_N)$ and $l_i^{ISI}(\bar{\zeta}_i, u_i)$ are defined in Section 2.1.4.4.

3. Apply the control input u_k, obtain a new measurements, and repeat this process.

3.2.2.2 Open-Loop Control Strategy

The Open-Loop Control (OLC) strategy uses the information available at the current stage to determine the belief state, ζ_k, but then ignores the possible influence of any new information when determining the optimal control sequence. It is convenient to think of the approach as solving the DP ISI problem with a restricted information set ([6], p. 376).

Definition 3.2: *Restricted Information State*

The restricted information set at stage i is a subset of the true information set. This restricted information set is used to define the restricted information state, $\bar{\eta}_i$.

The restricted information state used in the OLC strategy consists of the measurements up to the current stage, k, and controls up to stage $i > k$: $\bar{\eta}_i = (y_1, \ldots, y_k, u_0, \ldots, u_{i-1})$. The corresponding belief state is $\bar{\zeta}_i = p(x_i|\bar{\eta}_i)$. The future belief states are completely defined for a given control sequence, $u_{0:i-1}$, and the problem becomes deterministic (in terms of the resulting belief states). From Eq. (2.19), the expected cost in terms of the belief state is calculated over all possible disturbances and the disturbance is the next measurement (see Remark 2.1). Since the future measurements are ignored, this expectation can be dropped. Refer to Algorithm 3.2.

Remark 3.1: *The open-loop control problem is easier to solve than the feedback control problem because the restricted information set limits the set of reachable belief states in the problem. The disadvantage of the approach is that the uncertainty associated with the future belief states grows since the influence of the future information is ignored.*

3.3 Partially Closed-Loop Control Strategy

A novel approximation to the DP ISI problem, called the *Partially Closed-Loop Control* (PCLC) strategy, is presented here. The DP ISI problem is again approximated by recursively solving the planning problem on a restricted set of information, $\bar{\eta}_i$. The difference between the OLC strategy and the PCLC strategy is in the definition of the restricted information set. The key observation supporting this method is that the future measurements have two effects on the belief states: (i) *the value of the measurement changes only the mean of the belief state (not the spread), and (ii) the uncertainty associated with the measurement determines the uncertainty in the belief state, as well as the mean.* This can be understood in the context of the Kalman Filter (Section 2.4.3): the posterior mean, $\hat{x}_{i|i}$ (Eq. (2.30)), is a function of the actual value of the measurement, y_i. The posterior covariance, $\Sigma_{i|i}$ (Eq. (2.31)), on the other hand, is a function only of the prior state covariance and the innovation covariance, and not the value of the measurement. The effect of the measurement can be captured without unnecessarily biasing the future belief states by an appropriate assumed value for th future measurement, as described below.

3.3.1 Approximation: Most Probable Measurement

The PCLC strategy assumes that the most probable future measurement will occur, instead of considering the set of all possible measurements. Since a single measurement is assumed at every future stage, the set of reachable belief states is defined by the set of admissible control policies and the problem has similar complexity to the OLC strategy. By assuming that the most probable measurement value is obtained, the mean of the predicted belief state is not changed, but the uncertainty of this belief state is updated to reflect the fact that measurements are going to be taken at future times.

The restricted information set that is used is $\bar{\eta}_i = (y_1, \ldots, y_k, \tilde{y}_{k+1}, \ldots, \tilde{y}_i, u, \ldots, u_{i-1})$, where $\tilde{y}_j = E[y_j|\bar{\eta}_{j-1}]$ is the most probable value of y_j at stage j. The belief state associated with this restricted information set is $\bar{\zeta}_i = p(x_i|\bar{\eta}_i) = p(x_i|u_{0:i-1}, y_{1:k}, \tilde{y}_{k+1:i})$.

3.3.2 Algorithm

In Eq. (2.23), the expected cost is calculated over the restricted set of disturbances, $\bar{y}_{i+1}$. However, since $\bar{y}_{i+1} = \tilde{y}_{i+1}$, and not the set of all possible measurements, the outcome of the process becomes deterministic, and the expectation can be dropped. Refer to Algorithm 3.3.

Note that the future belief states are completely specified for a given control sequence since a single measurement at each future state is assumed. The problem is changed from having to calculate a feedback policy for all reachable belief states (optimal feedback control), to calculating the controls for a set of belief states that are fixed for a given control sequence. The approach relies on the outerloop feedback mechanism to correct for the actual measurements that are obtained.

Algorithm 3.3 Partially Closed-Loop Control Strategy

1. Calculate the belief state at the current time: $\zeta_k = p(x_k|\eta_k)$.
2. Find a control sequence, $\{u_k, \dots, u_{N-1}\}$, from the following dynamic programming problem:
$$\bar{J}_i(\bar{\zeta}_i) = \min_{u_i \in U} l_i^{ISI}(\bar{\zeta}_i, u_i) + \bar{J}_{i+1}(f_\zeta(\bar{\zeta}_i, u_i, \hat{y}_{i+1})) \quad k+1 \leq i \leq N-1 \tag{3.3}$$
$$\bar{J}_N(\bar{\zeta}_N) = l_N^{ISI}(\bar{\zeta}_N) \tag{3.4}$$
where $l_N^{ISI}(\bar{\zeta}_N)$ and $l_i^{ISI}(\bar{\zeta}_i, u_i)$ are defined in Section 2.1.4.4.
3. Apply the control input u_k, obtain a new measurements, and repeat this process.

3.3.3 Properties of PCLC Approach

A key requirement for the PCLC approach is that information is not artificially introduced into the problem by the most probable value assumption. This property can be shown, for the special case of Gaussian distributions, using the relative entropy of the system state distributions obtained at some future stage, i.

Proposition 3.1: *For Gaussian distributed system states, the information gain from the most probable measurement value assumption is less or equal to the information gain in the executed system (and the true measurements are obtained):*

$$D_{KL}^{exec} \geq D_{KL}^{PCLC}.$$

Proof: The relative entropy is a measure of the information gain from some base-line (BL) state distribution, $p_{BL}(x)$, to some comparitive (C) distribution, $p_C(x)$, where $x \in \mathbb{R}^{n_x}$. The relative entropy, also known as the Kullback-Leibler divergence (or KL-divergence) [19], is given by:

$$D_{KL}(p_C||p_{BL}) = \int_x p_C(x) \ln \frac{p_C(x)}{p_{BL}(x)} dx$$

For the specific case of normal distributions, $p_{BL}(x) = N_{BL}(\mu_{BL}, \Sigma_{BL})$ and $p_C(x) = N_C(\mu_C, \Sigma_C)$, the KL divergence is given by:

$$D_{KL}(p_C||p_{BL}) = \frac{1}{2}\left(\ln\left(\frac{|\Sigma_C|}{|\Sigma_{BL}|}\right) + Tr\left(\Sigma_C^{-1}\Sigma_{BL}\right) + \left(\mu_C - \mu_{BL}\right)^T \Sigma_C^{-1}\left(\mu_C - \mu_{BL}\right) - n_x\right).$$

In this analysis, the base-line distribution is the open-loop distribution (all future information is ignored): $p_{BL}(x_i|y_{1:k}) = N_{BL}(\mu_{BL}, \Sigma_{BL})$, where $\mu_{BL} = \hat{x}_{i|k}$ and $\Sigma_{BL} = \Sigma_{i|k}$. This is compared first to the distribution obtained from the PCLC approximation: $p_C(x_i|y_{1:k}, \tilde{y}_{k+1:i}) = N_C(\mu_C, \Sigma_C)$, where $\mu_C = \tilde{x}_{i|k} \triangleq E[x_i|y_{1:k}, \tilde{y}_{k+1:i}]$ and $\Sigma_C = \Sigma_{i|i}$. Note that with this particular choice of future measurement, the mean of the distribution is not updated, so that $\mu_C = \hat{x}_{i|k} = \mu_{BL}$. Thus, the information gain under the PCLC feedback approach is:

$$\begin{aligned} D_{KL}^{PCLC} &= \frac{1}{2}\left(\ln\left(\frac{|\Sigma_{i|i}|}{|\Sigma_{i|k}|}\right) + Tr\left(\Sigma_{i|i}^{-1}\Sigma_{i|k}\right) + \left(\hat{x}_{i|k} - \hat{x}_{i|k}\right)^T \Sigma_{i|i}^{-1}\left(\hat{x}_{i|k} - \hat{x}_{i|k}\right) - n_x\right) \\ &= \frac{1}{2}\left(\ln\left(\frac{|\Sigma_{i|i}|}{|\Sigma_{i|k}|}\right) + Tr\left(\Sigma_{i|i}^{-1}\Sigma_{i|k}\right) - n_x\right). \end{aligned}$$

Second, the base-line distribution is compared to the best information case: the distribution that can be obtained from the execution of the system (when the true measurements have been obtained): $p_C(x_i|y_{1:i}) = N_C(\mu_C, \Sigma_C)$, where $\mu_C = \hat{x}_{i|i}$ and $\Sigma_C = \Sigma_{i|i}$. The information gain in this best-case is:

$$D_{KL}^{exec} = \frac{1}{2}\left(\ln\left(\frac{|\Sigma_{i|i}|}{|\Sigma_{i|k}|}\right) + Tr\left(\Sigma_{i|i}^{-1}\Sigma_{i|k}\right) + \left(\hat{x}_{i|i} - \hat{x}_{i|k}\right)^T \Sigma_{i|i}^{-1}\left(\hat{x}_{i|i} - \hat{x}_{i|k}\right) - n_x\right).$$

The effect of the value of the measurement is to shift the mean of the distribution, so the quadratic term in terms of the means is non-negative. Since the covariances for the executed and PCLC approaches are the same, the desired result is obtained. ■

The information gain for the executed system is equal to or larger than the information gain with the PCLC approximation. Equality is achieved when the obtained sequence of measurements coincides with the most probable measurements (which is a very low probability scenario).

Remark 3.2: *The most probable measurement is the least informative assumption about the value of the measurement that is possible: any other assumption will introduce bias in the system, which will result in an information gain. Thus, this approximation optimally uses the effect of the measurement by updating the covariance, but then ignores the information gain from the value of the measurement. This is desirable since the value of the measurement cannot be known a priori, but the effect of the measurement can be modeled.*

3.4 Stochastic Receding Horizon Control

The receding horizon control (RHC) approach was motivated by the need to incorporate constraints and handle nonlinear dynamics in the optimal control problem (see Section 2.2). Attempts have been made to extend the basic RHC approach to stochastic systems. As with the FC ISI problem (Definition 2.3, without the constraints), it is convenient to convert the problem into the belief space (see Section 2.1.4.4). In this space, the measurement at the next stage is the only uncertain parameter (see Remark 2.1). As for the basic RHC problem, a sequence of control actions is obtained (as opposed to a feedback control law) over some finite horizon, $M \leq N$, and the problem is re-solved at each stage. Note the similarity to the approximate methods for the DP ISI problem, Section 3.2.2, which also used the outerloop feedback mechanism.

Definition 3.3: *Stochastic Receding Horizon Control*

Consider the constrained feedback control problem with imperfect state information (Definition 3.1). The problem is converted to the belief space (see Section 2.1.4.4), and the stochastic receding horizon control *(SRHC) problem is formulated in terms of the belief state, ζ_i (Definition 2.6). The objective is to find the sequence of control actions, $u_{k:M-1}$, over the horizon M, that minimizes the expected cost, satisfies the system dynamics, and satisfies the problem constraints:*

$$\begin{aligned} \min_{u_{k:M}} \quad & E_{y_{k+1:M}}\left[l_M^{ISI}(\zeta_M) + \sum_{i=k}^{M-1} l_i^{ISI}(\zeta_i, u_i) \right] \\ s.t. \quad & c_j(u_i) \leq 0 \quad \forall i = k, \ldots, M-1, \ \forall j = 1, \ldots, n_c \\ & P(x_i \notin \mathbb{X}_{free} | \eta_\tau) \leq \delta \quad \forall i = k, \ldots, M-1. \end{aligned} \tag{3.5}$$

The conditioning of the chance constraint will be discussed below.

3.4.1 Properties of the SRHC Formulation

The stability of solutions to the receding horizon control problem have received substantial attention, but progress on issues related to the robustness of the solutions in the presence of disturbances has been slower. The purpose of this section is to give a brief overview of the standard approaches to show stability and robustness, and to high light possible shortcomings for the the problem of planning in dynamic, uncertain environments. For a detailed discription, the interested reader is referred to review papers such as Mayne et al. [42] and Morari and Lee [43].

3.4.1.1 Stability: Deterministic System

Rigorous stability analysis is limited to deterministic problems. Two approaches to prove the stability of solutions have emerged [42], both of which employ the value function, $J_0^N(x)$, as a control Lyapunov function. The first approach, constrains the terminal cost, $l_N(x_N)$ and the terminal constraint set, $x_N \in \mathbb{X}_F$, and assumes a stabilizing local controller, $\kappa_F(x)$, to show stability. The second approach uses a sequence of monotonically decreasing value functions, $J_0^j(x)$, for $j = 1, \ldots, N$, to ensure stability.

For the problem at hand, the robot additionally operates in a dynamic environment and assumptions about the dynamic capability of the robot compared to the moving agents in the environment is necessary. No matter how good the planner, if the robot cannot out-maneuver the other agents, there can be no guarantees about the stability of the system (the ability to reliably reach some goal set). Agent can always push the robot away from the goal. Thus, to make headway on the stability analysis, it will be necessary to assume that the robot can react sufficiently fast to it's dynamic environment so as to make progress towards the goal in order to monotonically decrease the value function. This requires further investigation and is beyond the scope of this thesis.

3.4.1.2 Robustness: Bounded State Covariance

When uncertainty is introduced into the system, questions about the stability and performance of the planning approach is raised. Most studies on the robustness of the SRHC formulation assume the unconstrained problem [42] and a set-membership description of the noise terms. In the presence of state constraints, it is additionally necessary to ensure that the constraints are not violated due to the uncertainties.

The uncertainties are most often described in terms of bounded sets and the stability is shown for some worst-case disturbance (using the stability results from before). In the case of probabilistic disturbances, any disturbance is theoretically possible (though not very probable). As a result, stability in the deterministic sense can never be guaranteed. Instead, a weaker stability condition needs to be defined: stability with some probability. This important and interesting problem is outside the scope of this thesis.

When constraints are considered, one necessary condition for any notion of stability is to have bounded predicted state uncertainty. If the predicted state distribution covariance grows in an unbounded manner, and constraints are imposed in the form of chance constraints, then any state constraint will eventually force the robot out of the terminal (goal) set. It is informative at this point to consider the conditions under which the predicted state distributions have bounded covariance. For this, consider the special case of a linear system with Gaussian noise (the resulting optimal estimator is the Kalman Filter, Section 2.4.3). Using the system of Eq. (2.26) and Eq. (2.27), the distribution is propagated according to Eq. (2.28) to Eq. (2.34).

First, if all future information is ignored, then only the prediction step of the filter is executed, resulting in the following sequence for the predicted state covariances (from stage k to $k+3$):

$$\begin{aligned}
\Sigma_{k+1|k} &= A\Sigma_{k|k}A^T + FWF^T \\
\Sigma_{k+2|k} &= AA\Sigma_{k|k}A^TA^T + AFWF^TA^T + FWF^T \\
\Sigma_{k+3|k} &= AAA\Sigma_{k|k}A^TA^TA^T + AAFWF^TA^TA^T + AFWF^TA^T + FWF^T.
\end{aligned}$$

If the open-loop system has stable dynamics, then the effect of past disturbances diminish over time, and the covariance remains bounded, but one cannot generally assume that stable system dynamics. One approach to bound the covariance and account for future information is to use an assumed closed-loop system: the future system states are fed back with some constant stabilizing controller so that the closed-loop system becomes stable. Let $u_i \triangleq u_i^R + Kx_i$, where u_i^R is some reference input, and K is a constant-gain controller, then the closed-loop system has the form:

$$x_i = A_{cl}x_{i-1} + B_{i-1}u^R_{i-1} + F_{i-1}\omega_{i-1}$$

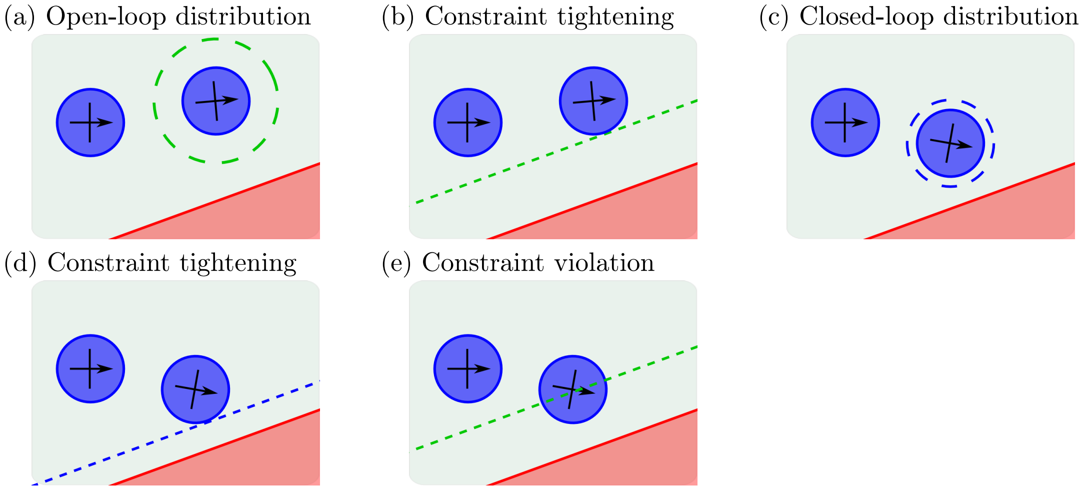
where A_{cl} is stable and as a result the covariance growth is bounded. The disadvantage is that the future state covariance is a function of some pre-selected controller gain, and the resulting predicted distributions might not be accurate representations of the system state distribution evolution during execution since the disturbance and measurement noise properties are not accounted for. A second alternative is to feed back the effects of the anticipated measurements. In this case, the measurement update step of the Kalman Filter is used to update the covariance of the predicted states. The posterior covariance (after measurement update) is given in non-standard form by:

$$\Sigma_{i|i} = \left(\Sigma^{-1}_{i|i-1} + C^T \left(HVH^T\right)^{-1} C\right)^{-1}$$

and is independent of the measurement value. Two cases are considered: (i) the measurement noise covariance, V, is small relative to the current state distribution covariance, $\Sigma_{k|k}$, and the process noise covariance, W, and (ii) the measurement noise covariance, V, is large relative to $\Sigma_{k|k}$ and W. For the first case, $\|\Sigma^{-1}_{i|i-1}\| \ll \|\left(HVH^T\right)^{-1}\|$ and $\|\Sigma_{i|i}\| \sim \|V\|$ as $i \to \infty$ (the predicted covariance is bounded by some function of the measurement noise covariance). For the second case, $\|\Sigma_{i|i}\| \sim \|\Sigma_{i|i-1}\|$ initially, but eventually this covariance will dominate the measurement noise and again $\|\Sigma_{i|i}\| \sim \|V\|$ as $i \to \infty$. Thus, when we incorporate the measurements, the predicted covariance will be bounded as some function of the measurement noise, as desired. The resulting predicted distribution covariances will also reflect the obtained future distribution covariances.

3.4.1.3 Chance Constraint Conditioning

The control action, u_k, affects the next robot state (via the dynamics) and the chance constraint at stage $k+1$ must be considered when evaluating the effect of control action u_k. In the DP ISI framework, the role of the disturbance in the belief space is played by the future measurements (see Remark 2.1). If the chance constraint is imposed while accounting for the all the possible measurements (disturbances) at the next stage, the 1-step open-loop predicted distribution is used and the robot's safety is probabilistically guaranteed (for all possible measurements). This is illustrated in Figure 3.1 (a). From Section 2.3 the effect of the chance constraint is to tighten the constraint by some amount that is a function of the desired level of confidence and the predicted state covariance, as illustrated in Figure 3.1 (b). The imposed chance constraint becomes: $P(x_{k+1} \notin X_{free}|\eta_k) \leq \delta$ where measurements up the stage k are considered. Assuming for now that anticipated future measurements are available (this will be substantiated in the sections to follow), the closed-loop predicted



(a) Open-loop distribution

(b) Constraint tightening

(c) Closed-loop distribution

(d) Constraint tightening

(e) Constraint violation

Figure 3.1: The effect of conditioning in chance constraints for a robot (blue) at stage k (left) and $k+1$ (right). The static constraint (red) is imposed using chance constraints. (a) The 1-step open-loop predicted distribution is indicated with a 1-σ uncertainty ellipse (green, dashed). (b) The resulting constraint tightening with the open-loop predicted distribution. (c) The closed-loop predicted distribution uncertainty (blue, dashed). (d) The resulting constraint tightening with the closed-loop predicted distribution. (e) Some disturbances can result in constraint violation.

distribution (for the anticipated measurments) can alternatively be used during constraint imposition, as illustrated in Figure 3.1 (c). The resulting constraint tightening is illustrated in Figure 3.1 (d) for the constraint: $P(x_{k+1} \notin X_{free}|\eta_{k+1}) \leq \delta$. However, some true measurements are possible that will result in constraint violation (see Figure 3.1 (e)), and the former constraint should be used.

In the previous discussion, an implicit assumption is that the robot can react to the environment at the next stage $(k+1)$. In practical applications, the robot cannot react instantaneously to its environment due to the robot dynamics and inertia effects. In dynamic environments, the robot must additionally react to the moving agents. The robot's dynamic capability, relative to the agents' dynamic capabilities, determines how quickly the robot can react to the changes in the environment. When stochasticity is added to the problem, the (chance) constraints grow proportionally to the growth in predicted state uncertainty, and the robot must be able to out-maneuver this artificial dynamic effect. All of these factors can be captured by the required *reaction time*, t_r, for the robot. The reaction time will be application specific and must be determined on a case-by-case basis. This reaction time plays an important role in guaranteeing the safety of the approximate SRHC algorithms to follow.

When evaluating a candidate control sequence, the chance constraints at some future stage $i \leq k + t_r$ must hold for all possible measurements during an interval of duration t_r to guarantee (probabilistically) the safety of the system. Consider the following thought experiment: assume a system with a reaction time of $t_r = 2$ stages. Starting at stage k, if the chance constraint is imposed while accounting for the all the possible measurements at the next stage, the 1-step open-loop predicted distribution is used. This results in the conditioning of the chance constraint: $P(x_{k+1} \notin X_{free}|\eta_k) \leq \delta$. However, the robot requires $t_r = 2$ stages to react to the environment and consequently there exists disturbances that do not allow for the robot to satisfy the chance constraints at stage $k + 2$ if $P(x_{k+2} \notin X_{free}|\eta_{k+1}) \leq \delta$ is used, since this constraint expression implicitly assumes that the robot has a reaction time of $t_r = 1$. If the control sequence $u_{k:k+1}$ is chosen so that $P(x_{k+1} \notin X_{free}|\eta_k) \leq \delta$ and $P(x_{k+2} \notin X_{free}|\eta_k) \leq \delta$ (using the 2-step open-loop predicted distribution), then the system state will be probabilistically safe for all disturbances for the duration of the reaction time. At the future states, the anticipated measurements up to that stage is assumed and the chance constraint must hold for all possible measurements (disturbances in belief space) for the duration of the reaction time: $P(x_{i+1} \notin X_{free}|\eta_i) \leq \delta$ and $P(x_{i+2} \notin X_{free}|\eta_i) \leq \delta$. In that case, the robot can recover (with some probability) from any sequence of disturbances.

In general, the appropriate conditioning of the chance constraint to guarantee probabilistic safety is:

$$P(x_i \notin X_{free}|\eta_{i-t_r}) \leq \delta.$$

Refer to Sections 3.5.2 and 3.5.3 for simulation results that illustrate the necessity for this conditioning.

3.4.2 Approximations to the SRHC Problem

Solving the SRHC problem is not trivial for the same reasons the DP ISI problem is difficult to solve; the set of reachable belief states where the cost needs to be evaluated is large (infinite in the continuous case). The SRHC problem has been approximated in similar ways to the DP ISI problem: the problem is approximately formulated as the deterministic counterpart (Certainty-Equivalent RHC), or the problem is defined on a restricted information set (Open-Loop RHC).

3.4.2.1 Certainty-Equivalent Receding Horizon Control

Typical values (such as the expected values) are assumed for the uncertain parameters, and the uncertainty is ignored so that the problem becomes deterministic ([6] p. 366) in the Certainty Equivalent Receding Horizon Control (CERHC) approach. The RHC algorithm of Definition 2.8 is used with the typical values of the states replacing the stochastic values. The CERHC algorithm is given in Algorithm 3.4.

3.4.2.2 Open-Loop Receding Horizon Control

As with the OLC approximation to the DP ISI problem, measurements beyond the current stage are ignored in the Open-Loop Receding Horizon Control (OLRHC) approach [9, 10, 12, 36, 48, 57, 59, 58, 61]. Since it is assumed that no future measurements will be available, the future belief states

Algorithm 3.4 Certainty Equivalent Receding Horizon Control

1. Compute the typical values $\bar{x}_k(\zeta_k)$, $\bar{\omega}_k(\bar{x}_k, u_k)$.
2. Find the control sequence, $\{u_k, \ldots, u_{N-1}\}$, that minimizes the deterministic cost function with the uncertain parameters fixed at their typical values:

$$\begin{aligned} \min_{u_{k:N-1}} \quad & \bar{l}_N(\bar{x}_N) + \sum_{i=k}^{N-1} l_i(\bar{x}_i, u_i, \bar{\omega}_i(\bar{x}_i, u_i)) \\ s.t. \quad & c(u_i) \le 0 \\ & \bar{x}_{k:M} \in \mathbb{X}_{free}. \end{aligned}$$

3. Apply the first control action, obtain a new measurement, and repeat the process.

Algorithm 3.5 Open-Loop Receding Horizon Control

1. Calculate the belief state at the current stage.
2. Define a control sequence, $u_{k:M-1}$, and calculate the future belief states in terms of this control sequence, $\{\zeta_k, \bar{\zeta}_{k+1}(u_k), \ldots, \bar{\zeta}_{k+M}(u_{k+M})\}$.
3. The optimal control sequence is obtained from the optimization problem:

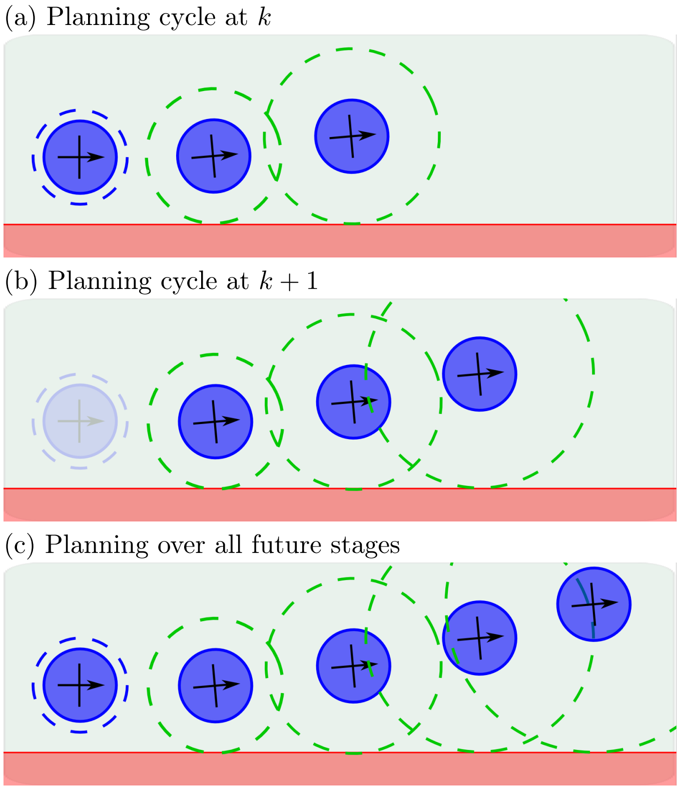
$$\begin{aligned} \min_{u_{k:M} \in U} \quad & l_k^{ISI}(\zeta_k, u_k) + \sum_{i=k+1}^{M-1} l_i^{ISI}(\bar{\zeta}_i, u_i) + l_M^{ISI}(\bar{\zeta}_M) \\ s.t. \quad & c(u_i) \le 0 \\ & P(x_{i+1} \notin X_{free} | \bar{\eta}_i) \le \delta_i \end{aligned} \tag{3.6}$$

   where the constraints are imposed for all $i = 1, \ldots, M$. $l_M^{ISI}(\bar{\zeta}_M)$ and $l_i^{ISI}(\bar{\zeta}_i, u_i)$ are defined in Section 2.1.4.4.

4. Execute the first control action, obtain a new measurement, and repeat the process.

are known for a given control sequence. Let $\{\zeta_k, \bar{\zeta}_{k+1}(u_k), \ldots, \bar{\zeta}_{k+M}(u_{k+M})\}$ be those belief states, where $\bar{\zeta}_i = p(x_i|\bar{\eta}_i) = p(x_i|u_{0:i-1}, y_{1:k})$. Since the future measurements are fixed (they are ignored), the expectation in the stage cost term, Eq. (3.5), can be dropped. The OLRHC algorithm is given in Algorithm 3.5.

Yan and Bitmead [61] noted that the solutions obtained from the OLRHC approach tends to be very conservative due to the chance constraints that are imposed. To see this, note that the conditioning for the chance constraints become $P(x_i \notin X_{free}|\eta_k) \le \delta$ for $i \ge k$ since all future measurements are ignored. The effect is that the uncertainty grows (see Section 3.4.1.2). Combined with the chance constraint formulation, the constraints are tightened proportionally to the covariance of the predicted state distribution (Section 2.3), resulting in very conservative solutions. This is illustrated for a system with reaction time $t_r = 2$ in Figure 3.2 where the green, dashed ellipses



(a) Planning cycle at k

(b) Planning cycle at $k + 1$

(c) Planning over all future stages

Figure 3.2: The open-loop approach is conservative when the robot (blue) moves along a static obstacle (red). The safety buffer around the robot due to chance constraint tightening is given in green, dashed circles. (a) The open-loop prediction for the next 2 stages is used (reaction time, $t_r = 2$ stages). (b) At the following stage, the anticipated measurements are ignored and the predicted state distribution is used. (c) For all future stages, the uncertainty grows, which results in conservative solutions when the chance constraints are imposed.

Algorithm 3.6 Approximately Closed-Loop Receding Horizon Control

1. Calculate the belief state at the current stage.
2. Define a control sequence, $u_{k:M-1}$, and calculate the future belief states in terms of this control sequence, $\{\zeta_k, \bar{\zeta}_{k+1}(u_k), \ldots, \bar{\zeta}_{k+M}(u_{k+M})\}$.
3. Replace the covariances of the future belief states at the covariance at the second stage, $\Sigma_{k+1|k}$, to obtain a new set of belief states, $\{\zeta_k, \tilde{\zeta}_{k+1}(u_k), \ldots, \tilde{\zeta}_{k+M}(u_{k+M})\}$.
4. The optimal control sequence is obtained from the optimization problem:
$$\begin{aligned} \min_{u_{k:M} \in U} \quad & l_k^{ISI}(\zeta_k, u_k) + \sum_{i=k+1}^{M-1} l_i^{ISI}(\tilde{\zeta}_i, u_i) + l_M^{ISI}(\tilde{\zeta}_M) \\ s.t. \quad & c(u_i) \leq 0 \\ & P(x_{i+1} \notin X_{free} | \bar{\eta}_i) \leq \delta_i \end{aligned} \tag{3.7}$$
where the constraints are imposed for all $i = 1, \ldots, M$. $l_M^{ISI}(\bar{\zeta}_M)$ and $l_i^{ISI}(\bar{\zeta}_i, u_i)$ are defined in Section 2.1.4.4.
5. Execute the first control action, obtain a new measurement, and repeat the process.

indicate the effective size of the robot at future states due to constraint tightening.

3.4.2.3 Approximately Closed-Loop Receding Horizon Control

Yan and Bitmead [61] noted that the covariance of the future belief states grow unbounded when using the OLRHC approach. To counter this effect, the 'closed-loop covariance' is introduced in the Approximately Closed-Loop Receding Horizon Control (ACLRHC) approach. The covariance for all future states is fixed at the one-step ahead open-loop prediction value. This is a crude attempt to account for the anticipated future information during the planning problem. For a linear system, with white Gaussian noise, the future belief states are normally distributed, with the covariance of fixed beyond the second stage in the future: $\bar{\zeta}_i = N(\hat{x}_{i|k}, \Sigma_{k+1|k}) \; \forall \geq k+1$. The ACLRHC approach is given in Algorithm 3.6.

This approximation addresses the conservatism due to the OLRHC approach, but by assuming a fixed covariance for all future states, the planner is unable to reason about the quality of information (the estimation process may be a function of the control sequence, as is the case with active learning). This is a major disadvantage for a general approach.

3.4.3 Partially Closed-Loop Receding Horizon Control

The goal here is to extend the new PCLC strategy of Section 3.3 to include chance constraints.

Algorithm 3.7 Partially Closed-Loop Receding Horizon Control

1. Calculate the belief state at the current stage.
2. Define a control sequence, $u_{k:M-1}$, and calculate the future belief states in terms of this control sequence, $\{\zeta_k, \bar{\zeta}_{k+1}(u_k), \ldots, \bar{\zeta}_{k+M}(u_{k+M})\}$.
3. The optimal control sequence is obtained from the optimization problem:

$$\begin{aligned} \min_{u_{k:M}\in U} \quad & l_k^{ISI}(\zeta_k, u_k) + \sum_{i=k+1}^{M-1} l_i^{ISI}(\bar{\zeta}_i, u_i) + l_M^{ISI}(\bar{\zeta}_M) \\ s.t. \quad & c(u_i) \leq 0 \\ & P(x_{i+1} \notin X_{free}|\bar{\eta}_i) \leq \delta_i \end{aligned} \tag{3.8}$$

   where the constraints are imposed for all $i = 1, \ldots, M$. $l_M^{ISI}(\bar{\zeta}_M)$ and $l_i^{ISI}(\bar{\zeta}_i, u_i)$ are defined in Section 2.1.4.4.

4. Execute the first control action, obtain a new measurement, and repeat the process.

3.4.3.1 Approximation

As with the PCLC approach, the most probable future measurement is assumed. The restricted information set used in this approach is $\bar{\eta}_i = (y_1, \ldots, y_k, \tilde{y}_{k+1}, \ldots, \tilde{y}_i, u, \ldots, u_{i-1})$, where $\tilde{y}_j = E[y_j|\bar{\eta}_{j-1}]$ is the most probable value of y_j at stage j. The problem is solved for the resulting set of belief states, instead of the belief states that result from all possible future measurements. Since the values for the future measurements are assumed, the future belief states are known for a given control sequence (similar to the OLRHC and ACLRHC approaches). Let $\{\zeta_k, \bar{\zeta}_{k+1}(u_k), \ldots, \bar{\zeta}_{k+M}(u_{k+M})\}$ be those belief states, where $\bar{\zeta}_i = p(x_i|\bar{\eta}_i) = p(x_i|u_{0:i-1}, y_{1:k}, \tilde{y}_{k+1:i})$.

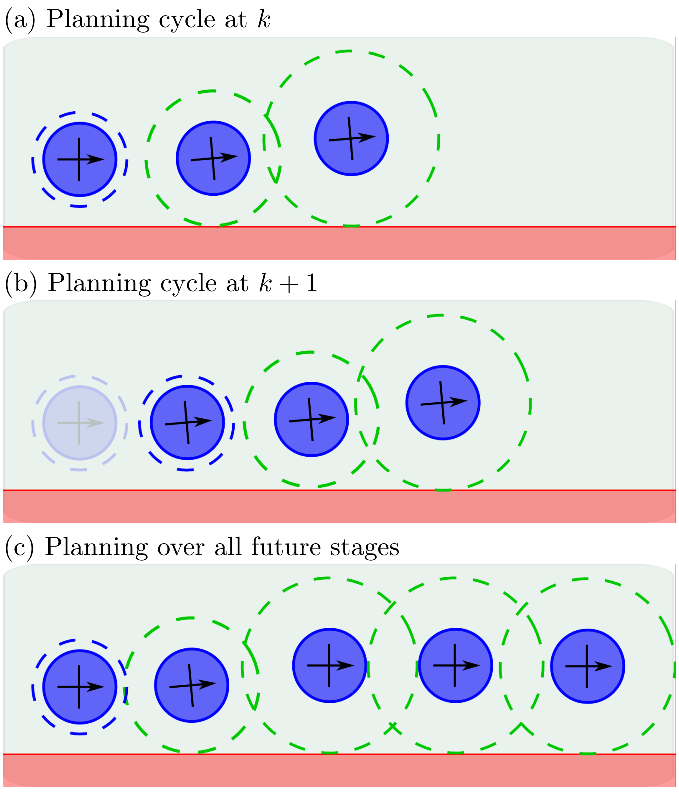
3.4.3.2 Algorithm

Since the future measurements are fixed at the most probable value, the expectation in the stage cost term, Eq. (3.5), can be dropped since the 'disturbances' can only assume a single value. The PCLRHC approach is given in Algorithm 3.7.

3.4.3.3 Properties of the PCLRHC Approach

The assumption about the most probable measurement does not introduce information in the planning process since the best-information case (after execution) always contains at least as much information (Section 3.3.3). Also, the assumed value of the measurement contains the least amount of information. As a result, the effect of the measurement is optimally incorporated, but the information contained in the value of the (unknown) future measurement is ignored.

As noted in Section 3.4.1.3, the chance constraints that are imposed are conditioned on measurements up to t_r previous stages, where t_r is the reaction time of the system. For the chance


(a) Planning cycle at k

(b) Planning cycle at $k+1$

(c) Planning over all future stages

Figure 3.3: The robot (blue) moves along a static obstacle (red). The effective size of the robot due to chance constraint tightening is given in green, dashed circles. (a) The open-loop prediction for the next 2 stages is used (reaction time, $t_r = 2$ stages). (b) At the following stage, the anticipated measurements is incorporated and the closed-loop distribution is used. (c) For all future stages, at worst the 2-step open-loop prediction is used to impose the constraints.

constraints, the t_r-step open-loop predicted state distribution is used, but since the future measurements are accounted for, this distribution has bounded covariance (as opposed to the possibly unbounded covariance in the open-loop case) as illustrated in Figure 3.3 for a reaction time of $t_r = 2$. As a result, the constraint tightening is bounded and less conservative solutions can be obtained, while maintaining the safety of the system.

3.5 Simulation Results

The OLRHC (which is representative of current stochastic RHC practice) and the PCLRHC approaches are compared in a simple scenarios using linear chance constraint results. First, the approaches are applied in a static environment where the robot must skirt a static obstacle. Next, the approaches are applied to a dynamic environment where the robot must follow a moving agent with a prescribed minimum separation distance. This example is used to highlight the reaction time (dynamic capability) and constraint conditioning results above.

3.5.1 Example 1: Static Environment

Even this trivial example shows how anticipated future measurements and chance constraints (which are fundamental to the DCUE problem) can affect the motion planning outcome. The robot must navigate in a static environment with a single rectangular static obstacle with known geometry and position (its boundary lies on the x-axis).

Consider a point robot with the random walk dynamic model (Appendix B.1.3) and let $\Delta t = 0.5$ s and $W = 0.01 \times I_2$ (I_2 is the identity matrix in 2 dimensions). A linear position measurement model (Appendix B.2.2) is assumed with $V = 0.01 \times I_2$. The robot's initial position and goal location are chosen so that the robot must skirt the obstacle. The system is initialized with: $x_0 \sim \mathcal{N}(\hat{x}_{0|0}, \Sigma_{0|0})$, where $\hat{x}_{0|0} = [0\ 0.75\ 1\ 0]^T$ and $\Sigma_{0|0} = 0.01 \times I_4$. The goal state is $x_G = [10\ 0.75\ 0\ 0]$ and the objective is to minimize the expected value of the stage-additive cost function:

$$L(x_0) = (x_N - x_G)^T Q_N (x_N - x_G) + \sum_{i=0}^{N-1} \left\{ (x_i - x_G)^T Q_i (x_i - x_G) + u_i^T R_i u_i \right\}$$

where $Q_N = \text{diag}(10, 10, 0, 0)$, $Q_i = \text{diag}(1, 1, 0, 0)$, and $R_i = \text{diag}(1, 1, 0, 0)$, for all $i = 0, \ldots, N-1$.

In order to fully test the framework, each control input is constrained at each stage: $|u_i^{(1)}| \leq 1$ and $|u_i^{(2)}| \leq 1$, $i = 0, \ldots, N-1$. The state chance constraint takes the form: $P(x_i^{(2)} < 0|\eta_{i-1}) \leq \delta_{p,i}$ where $\delta_{p,i} = 0.01$. Furthermore, each velocity component is constrained to the range of $[-2, 2]$:

$$\begin{aligned} &P(x_i^{(3)} > 2|\eta_{i-1}) \leq \delta_{v,i} \text{ and } P(x_i^{(3)} < -2|\eta_{i-1}) \leq \delta_{v,i} \\ &P(x_i^{(4)} > 2|\eta_{i-1}) \leq \delta_{v,i} \text{ and } P(x_i^{(4)} < -2|\eta_{i-1}) \leq \delta_{v,i} \end{aligned}$$

where $\delta_{v,i} = 0.01$ for $i = 0, \ldots, M$.

For the OLRHC approach, the belief state is normally distributed, $\bar{\zeta}_i = p(x_i|\bar{\eta}_i) = N(\hat{x}_{i|k}, \Sigma_{i|k})$, where $\hat{x}_{i|k} = E[x_i|y_{1:k}, u_{0:i-1}]$ and $\Sigma_{i|k} = E[(x_i - \hat{x}_{i|k})(x_i - \hat{x}_{i|k})^T|y_{1:k}, u_{0:i-1}]$ are obtained from the Kalman Filter prediction step. The stage additive cost function in terms of the belief state is:

$$L^{ISI}(\zeta_0) \quad = \textstyle\sum_{i=0}^{M-1} \quad \left\{ Tr(Q_i \Sigma_{i|k}) + \hat{x}_{i|k}^T Q_i \hat{x}_{i|k} + u_i^T R_i u_i \right\} + Tr(Q_M \Sigma_{M|k}) + \hat{x}_{M|k}^T Q_M \hat{x}_{M|k}.$$

Note that the trace terms in the cost function cannot be influenced by the choice of controls for this problem, and these terms can be dropped from the objective function, so that it reduces to:

$$C \quad = \quad \min_{u_{k:M-1}} \sum_{i=k}^{M-1} \{ \hat{x}_{i|k}^T Q_i \hat{x}_{i|k} + u_i^T R_i u_i \} + \hat{x}_{M|k}^T Q_M \hat{x}_{M|k}.$$

The chance constraints are converted to constraints on the mean value of the belief states for normal distributions, as discussed in Section 2.3.1. From Lemma 2.1, the linear chance constraints are satisfied for all stages, $i = 1, \ldots, N$, if:

$$\begin{array}{llll}
P(x_i^{(2)} & \leq & 0|\bar{\eta}_{i-1}) \leq \delta_{v,i} & \iff \quad a^T\hat{x}_{i|k} + \text{icdf}(1-\delta_{p,i})\sqrt{a^T\Sigma_{i|k}a} \leq 0 \text{ with a} = [0\ -1\ 0\ 0]^{\mathrm{T}} \\
P(x_i^{(3)} & \geq & 2|\bar{\eta}_{i-1}) \leq \delta_{v,i} & \iff \quad a^T\hat{x}_{i|k} + \text{icdf}(1-\delta_{\nu,i})\sqrt{a^T\Sigma_{i|k}a} \leq 2 \text{ with a} = [0\ 0\ 1\ 0]^{\mathrm{T}} \\
P(x_i^{(3)} & \leq & -2|\bar{\eta}_{i-1}) \leq \delta_{v,i} & \iff \quad a^T\hat{x}_{i|k} + \text{icdf}(1-\delta_{\nu,i})\sqrt{a^T\Sigma_{i|k}a} \leq 2 \text{ with a} = [0\ 0\ -1\ 0]^{\mathrm{T}} \\
P(x_i^{(4)} & \geq & 2|\bar{\eta}_{i-1}) \leq \delta_{v,i} & \iff \quad a^T\hat{x}_{i|k} + \text{icdf}(1-\delta_{\nu,i})\sqrt{a^T\Sigma_{i|k}a} \leq 2 \text{ with a} = [0\ 0\ 0\ 1]^{\mathrm{T}} \\
P(x_i^{(4)} & \geq & -2|\bar{\eta}_{i-1}) \geq \alpha_{v,i} & \iff \quad a^T\hat{x}_{i|k} + \text{icdf}(1-\delta_{\nu,i})\sqrt{a^T\Sigma_{i|k}a} \leq 2 \text{ with a} = [0\ 0\ 0\ -1]^{\mathrm{T}}
\end{array}$$

where $\text{icdf}(1-\delta) = 3.0902$ for $\delta = 0.01$ (from Table 2.3.1).

For the PCLRHC approach, the belief state is normally distributed, $\bar{\zeta}_i = p(x_i|u_{0:i-1}, y_{1:k}, \tilde{y}_{k+1:i}) = N(\hat{x}_{i|i}, \Sigma_{i|i})$, where $\tilde{y}_j = E[y_j|\bar{\eta}_{j-1}]$ is the most probable measurement. $\hat{x}_{i|i} = E[x_i|y_{1:k}, \tilde{y}_{k+1:i}, u_{0:i-1}]$ and $\Sigma_{i|i} = E[(x_i - \hat{x}_{i|i})(x_i - \hat{x}_{i|i})^T|y_{1:k}, \tilde{y}_{k+1:i}, u_{0:i-1}]$ are obtained by recursively executing the Kalman Filter. Using the most probable measurement, the posterior mean in Eq. (2.30) becomes $\hat{x}_{i|i} = \hat{x}_{i|i-1}$. The stage additive cost terms, as well as the constraints, are the same as for the OLRHC formulation, but with the mean and covariances defined above.

The *planned* paths at the initial stage for the OLRHC (green) and PCLRHC (blue) approaches are shown in Figure 3.4 with 1-σ error ellipses representing the uncertainty. The robot (blue) and the static obstacle (red) are plotted. The optimal solution to this problem is trivial: the robot should travel in a straight line from the initial location to the goal location. However, the OLRHC solution diverges from the straight line due to the chance constraints that are being enforced. Since future measurements are not considered during this plan, the uncertainty in the predicted robot states grow and the robot must ‘back-off’ from the position constraint. The obtained solution is very cautious, and the goal is not reached. For the PCLRHC approach, the growth in uncertainty is bounded since the effects of the anticipated future measurements are incorporated in the plan and the solution drives the robot straight to the goal.

Figure 3.5 shows that the *executed* paths are similar due to the outerloop feedback mechanism. However, the PCLRHC's executed trajectory follows the optimal trajectory. Repeating the problem 30 times with different random noise samples (but using the same samples for both algorithms) the average deviation from the straight line for the PCLRHC approach is $0.174 \pm\ 0.058$ m. This is compared to 0.328 ± 0.088 m for the OLRHC problem. The maximum deviation for the PCLRHC approach is 0.420 ± 0.134 m and 0.657 ± 0.140 m for the OLRHC approach. Figure 3.5 also includes the planned trajectories calculated during the first planning cycle (dashed curves) to compare against the executed trajectories. The planned and executed trajectories for the OLRHC approach differ

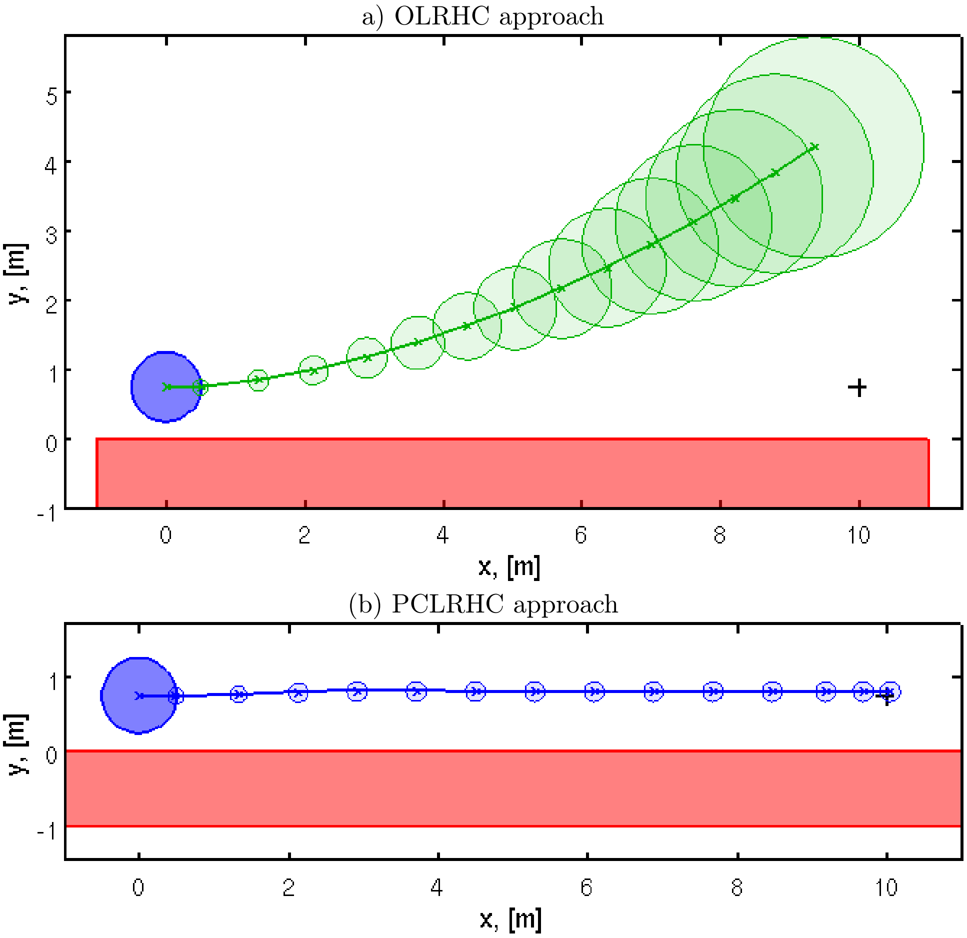


Figure 3.4: The planned trajectory with the (a) OLRHC (green) and (b) PCLRHC (blue) approach. The static obstacle is plotted in red.

substantially, as the planner relies almost exclusively on the outer-loop feedback mechanism to execute a reasonable trajectory. Clearly, the OLRHC is not efficiently using all available information. For the PCLRHC approach on the other hand, the planned and executed trajectories are very similar, and the outer loop feedback mechanism is used to correct for the actual measurements and noise encountered along the trajectory. This implies that the planner is efficiently using the anticipated future information when solving the planning problem.

An unwanted artifact of the growth of uncertainty for the OLRHC approach is that the robot cannot remain at the goal location, even if the goal can be reached before the end of the planning horizon (even in static environments). This is again due to the growth of the uncertainty of the state, and not due to a real threat to the robot (such as an approaching moving agent). Refer to Figure 3.6 for the solution at stage 10 as an example of this effect. The PCLRHC approach does not exhibit this effect.

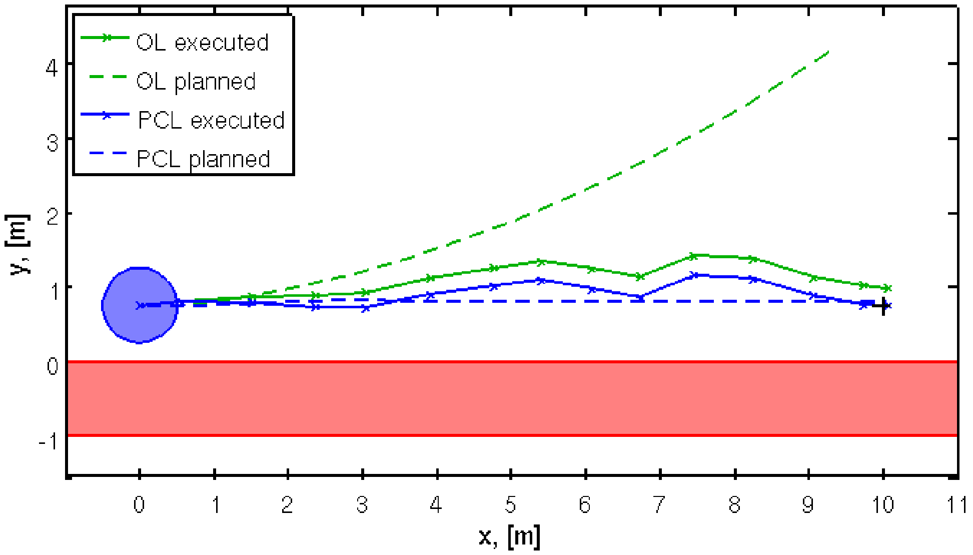


Figure 3.5: The planned (dashed) and executed (solid) paths for the OLRHC (green) and PCLRHC (blue) approaches. The executed trajectories are very similar, but the executed and planned paths for the PCLRHC approach are much closer than for the OLRHC counterparts.

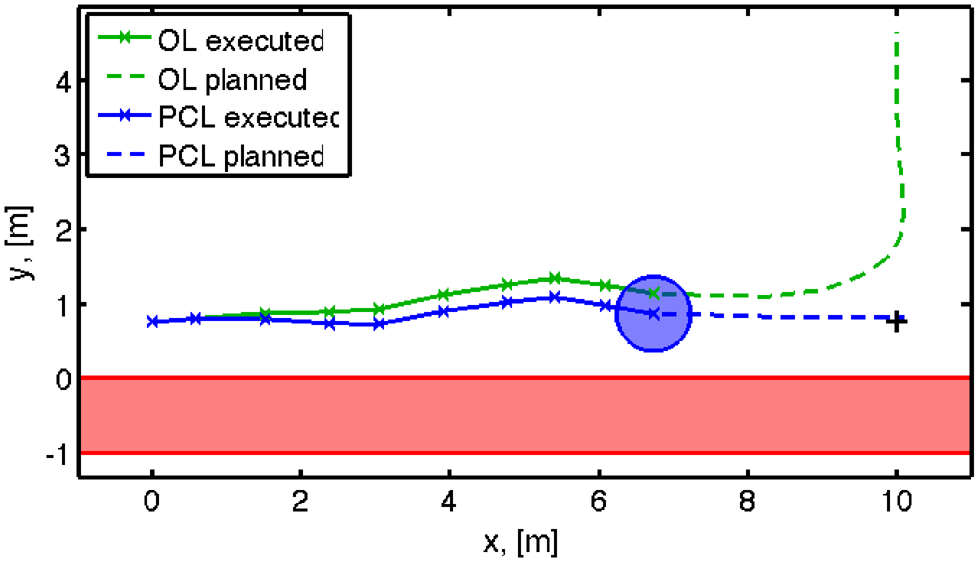


Figure 3.6: The planned (dashed) and executed (solid) paths for the OLRHC (green) and PCLRHC (blue) approaches. The OLRHC approach is unable to keep the robot at the goal.

3.5.2 Example 2: 1-D Car Following (Position)

The purpose of this example is to highlight to importance of the robot's ability to react to the changing environment (sufficient dynamic capability). This failure mode is not due to the approximation used in this research, but is due to the dynamic and stochastic nature of the problem considered here. The examples are chosen to illustrate failures.

Consider a point robot moving under a 1-D random walk dynamic model (position only, Appendix B.1.1) and let $\Delta t = 0.5$ s and $W = 0.01$. The control action changes the instantaneous velocity. A 1-D linear position measurement model (Appendix B.2.1) is assumed with $V = 0.01$.

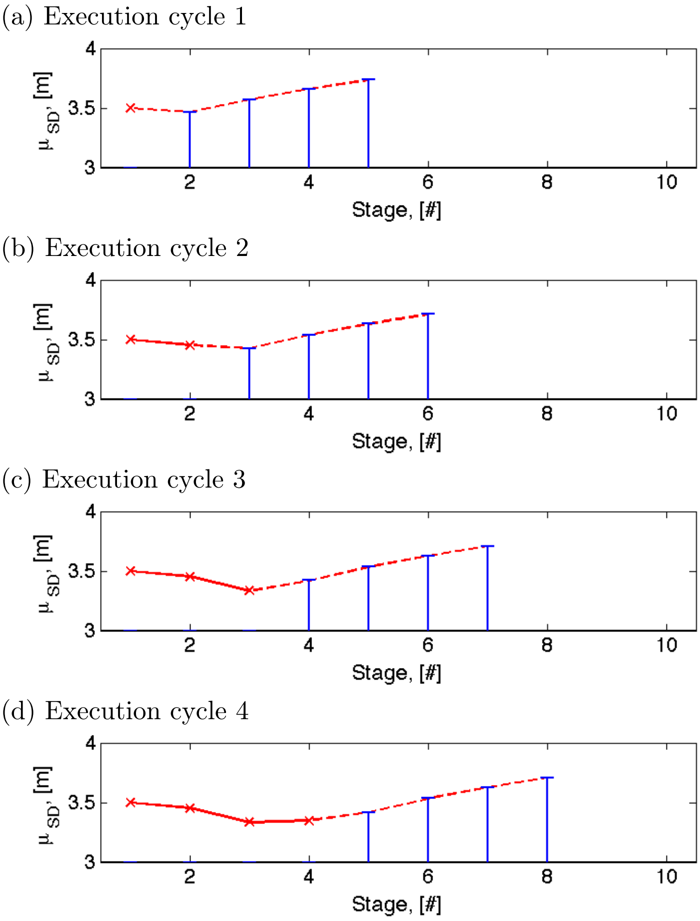
Similarly, an agent governed by a 1-D random walk dynamic model (position only, Appendix B.1.1) is assumed with $W = 0.01$. The agent travels at a nominally steady velocity ($v_A = 1$ m/s) which is randomly perturbed. A 1-D linear position measurement model (Appendix B.2.1) is assumed with $V = 0.01$.

The goal state is $x_G = 10$ and the objective is the same as in Section 3.5.1, with $Q_N = 10$, $Q_i = 1$, and $R_i = 0.1$, for all $i = 0, \ldots, N-1$. The change in control input is constrained at each stage: $|u_i - u_{i+1}| \leq \Delta v_{max}$, where Δv_{max} is specified below and $|u_i| \leq 1$. The separation distance between the robot and the agent results in a linear chance constraint of the form: $P(x_i^A - x_i^R \geq \rho | \eta_{i-1}) \geq \alpha_p$ where $\delta_p = 0.01$ and $\rho = 3$ m. The system is initialized with $x_0 \sim \mathcal{N}(\hat{x}_{0|0}, \Sigma_{0|0})$, where $\hat{x}_{0|0}$ is specified below and $\Sigma_{0|0} = 0.01$. Since the system is linear, with Gaussian noise and a normal initial distribution, the future states are normal as well, and the distance constraint can be written as:

$$\hat{x}_{i|i-1}^R - \hat{x}_{i|i-1}^A + \text{icdf}(\alpha)\sqrt{\Sigma_{i|i-1}^R + \Sigma_{i|i-1}^R} + \rho \leq 0.$$

3.5.2.1 Case 1: Sufficient Dynamics

The dynamic capability of the robot $\Delta v_{max} = 1$ m/s and the initial states $\hat{x}_{0|0}^R = 0$ and $\hat{x}_{0|0}^A = 3.5$, and the robot velocity capability allows it to react to the changes in the environment within 1 stage ($t_r = 1$). The initial velocity is specified with $u_{-1} = 1.5$ m/s. The planned separation distance (red, dashed) over the next 5 cycles are plotted for 4 executions for the OLRHC approach in Figure 3.7 (a) - (d). The separation distance from the executed plans (red, solid) are also shown. The effective constraints (due to constraint tightening) over the planning horizon are indicated with a blue 'T'-bar. A constraint violation occurs when the predicted (red, dashed) distances cross a blue 'T'-bar. In this example, not constraint violations occur. Similar results are given for the PCLRHC approach in Figure 3.8 (a) - (d). The probabilistic safety of the robot is guaranteed (see Section 3.4.1.3).



(a) Execution cycle 1

(b) Execution cycle 2

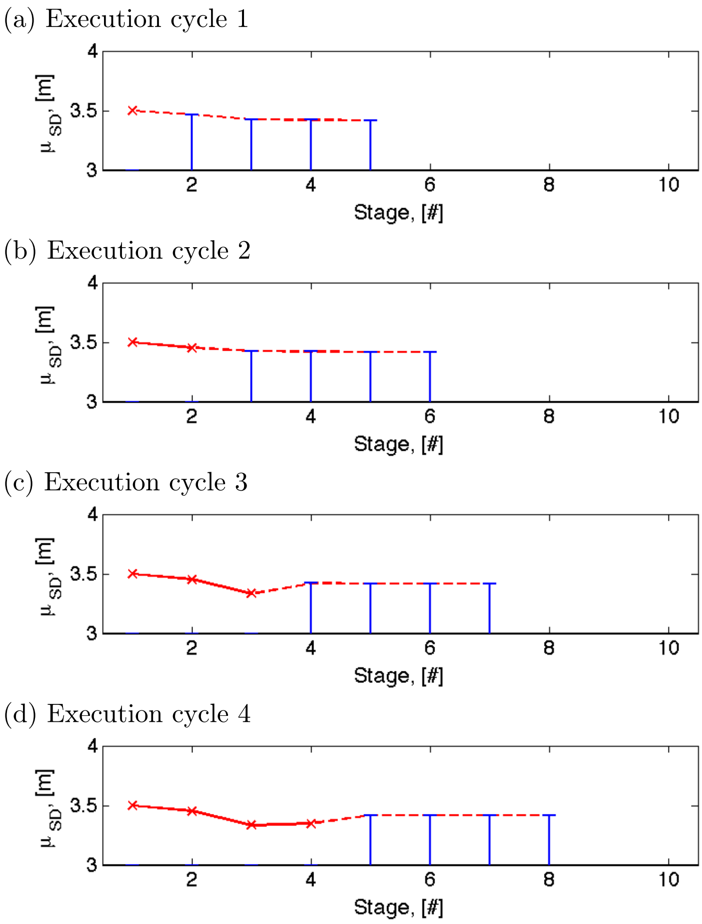
(c) Execution cycle 3

(d) Execution cycle 4

Figure 3.7: The separation distances (between the robot and moving agent) predicted from the OLRHC solution (red, dashed) and the obtained separation distances from the execution of the solutions (red, solid) for stages 1 through 4. The effective constraints (due to constraint tightening) that are imposed over the planning horizon are indicated with a blue 'T'-bar.

3.5.2.2 Case 2: Insufficient Dynamics due to Input Constraints

The dynamic capability of the robot is kept the same as in case 1. When the initial states are chosen as $\hat{x}^R_{0|0} = 0$ and $\hat{x}^A_{0|0} = 4$, and the initial velocity is set to $u_{-1} = 2$ m/s, the robot cannot react fast enough to the agent's movement in order to maintain the safety of the system. Effectively, the robot reaction time is higher ($t_r > 1$). Both the OLRHC and PCLRHC approaches fail to find a feasible solution at execution cycle 3 (the constraint at stage 2 is violated). See Figure 3.9. The probabilistic safety of the robot is not guaranteed, but this is not an artifact of the proposed PCLRHC approach. In fact, due to the uncertainty growth in the OLRHC approach (and the associated constraint tightening), the PCLRHC approach is expected to do somewhat better in these cases.



(a) Execution cycle 1

(b) Execution cycle 2

(c) Execution cycle 3

(d) Execution cycle 4

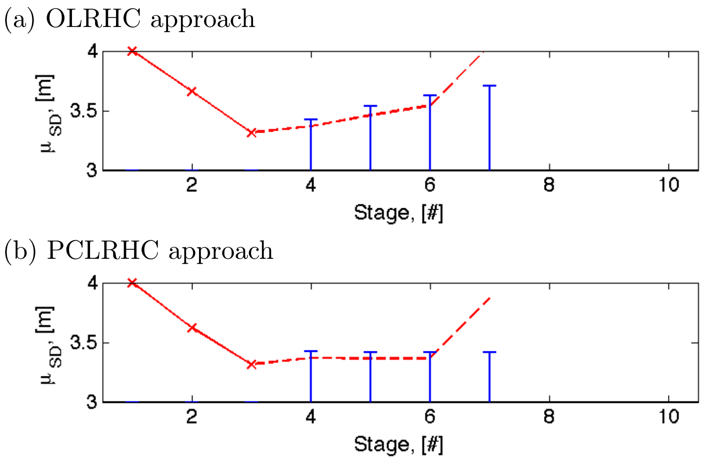
Figure 3.8: The separation distances predicted from the PCLRHC solution (red, dashed) and the obtained separation distances from the execution of the solutions (red, solid) for stages 1 through 4. The constraint tightening is plotted at the next 4 stages (at each planning cycle) in blue.

3.5.2.3 Case 3: Insufficient Dynamics due to Uncertainty Growth

The initial states and initial velocity of case 1 is used. The dynamic capability of the robot is reduced to $\Delta v_{max} = 0.5$ m/s so the the robot cannot react fast enough to growth in prediction uncertainty. Both the OLRHC and PCLRHC approaches fail to find a feasible solution (the constraint at stage 2 is violated) in this example, but it is expected that the PCLRHC approach will be less succeptable to this phenomena since the open-loop prediction is used over a shorter period during constraint evaluation than the OLRHC approach. See Figure 3.10. The probabilistic safety of the robot is not guaranteed, but again this is not an artifact of the proposed PCLRHC approach.

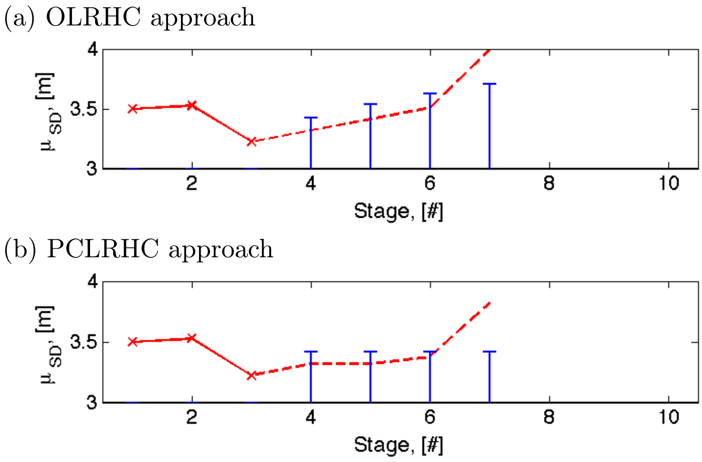
3.5.3 Example 3: 1-D Car Following (Position and Velocity)

The purpose of this example is to highlight to importance of the robot reaction time on the safety of the PCLRHC approach, presented in Section 3.4.1.3. The examples are chosen to illustrate failures.


(a) OLRHC approach

(b) PCLRHC approach

Figure 3.9: The required separation distance cannot be guaranteed at stage 2 of execution cycle 3 since the robot is unable to react to the changes in the environment. A constraint violation occurs when a predicted (red, dashed) distance crosses a blue 'T'-bar at some stage.


(a) OLRHC approach

(b) PCLRHC approach

Figure 3.10: The required separation distance cannot be guaranteed at stage 2 of execution cycle 3 since the robot is unable to react to the constraint tightening due to the uncertainty in the states

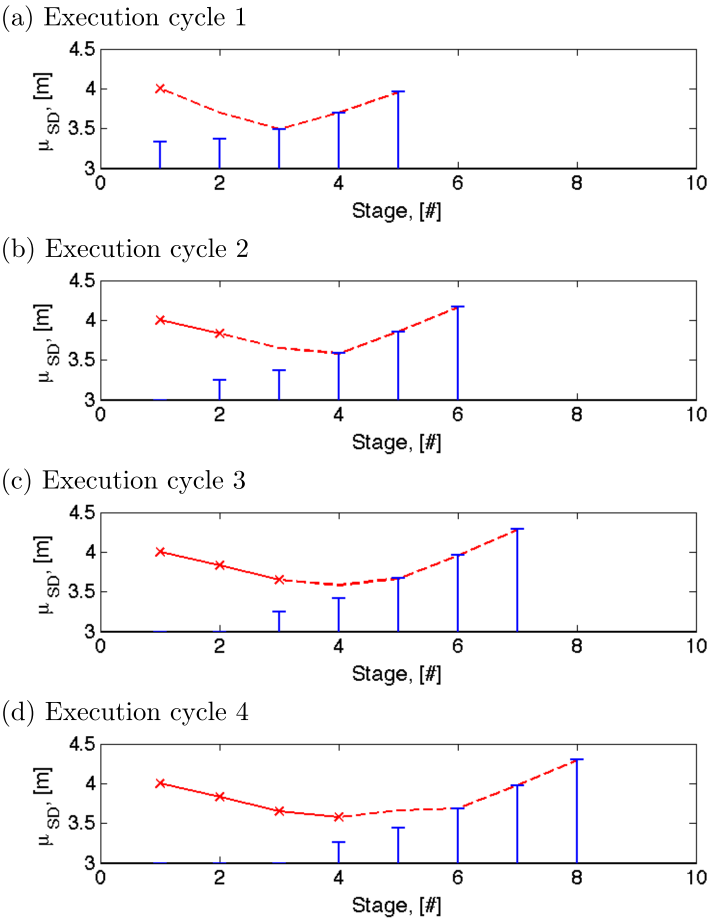
Consider a point robot under a 1-D random walk dynamic model (position and velocity, Appendix B.1.2) and let $\Delta t = 0.5$ s and $W = 0.01 \times I_2$. From the dynamics, the reaction time of the system is at least $t_r = 2$ stages. A 1-D linear position measurement model (Appendix B.2.1) is assumed with $V = 0.01$.

Similarly, an agent governed by a 1-D random walk dynamic model (position and velocity, Appendix B.1.1) is assumed with $W = 0.01 \times I_2$. The agent travels at a nominally steady velocity ($v_A = 1$ m/s) which is randomly perturbed. A 1-D linear position measurement model (Appendix B.2.1) is assumed with $V = 0.01$.

The robot is initialized to $x_0^R \sim \mathcal{N}(\hat{x}_{0|0}^R, \Sigma_{0|0}^R)$, where $\hat{x}_{0|0}^R = [0\ 1.6]^T$ and $\Sigma_{0|0}^R = 0.01 \times I_2$ and the agent is initialized to $x_0^A \sim \mathcal{N}(\hat{x}_{0|0}^A, \Sigma_{0|0}^A)$, where $\hat{x}_{0|0}^A = [4\ 1]^T$ and $\Sigma_{0|0}^A = 0.01 \times I_2$. The goal state is $x_G = \begin{bmatrix} 10 & 0 \end{bmatrix}$ and the objective and velocity constraints of Section 3.5.1 are used, with $Q_N = \text{diag}(10, 0)$, $Q_i = \text{diag}(1, 0)$, and $R_i = 0.1$, for all $i = 0, \ldots, N-1$. The separation distance constraints of Section 3.5.2 are assumed. Additionally, the control is constrained to $|u_i| \leq 1$. Note that these initial conditions should allow the robot sufficient capability to react to the environment.

3.5.3.1 Case 1: Correctly Assumed Reaction Time

A robot reaction time of $t_r = 2$ is correctly assumed, and the chance constraints are imposed accordingly (see Section 3.4.1.3). Using a similar color scheme to Section 3.5.2, the predicted and executed separation distances are plotted for the OLRHC and PCLRHC approaches in Figures 3.11 and 3.12 for the first 4 planning (and execution) cycles. The safety of the system is probabilistically guaranteed since the correct conditioning is used when evaluating the chance constraints. Again, the growth in uncertainty results in conservative plans for the OLRHC approach. As expected, the constraints tightening for the PCLRHC approach remains bounded.

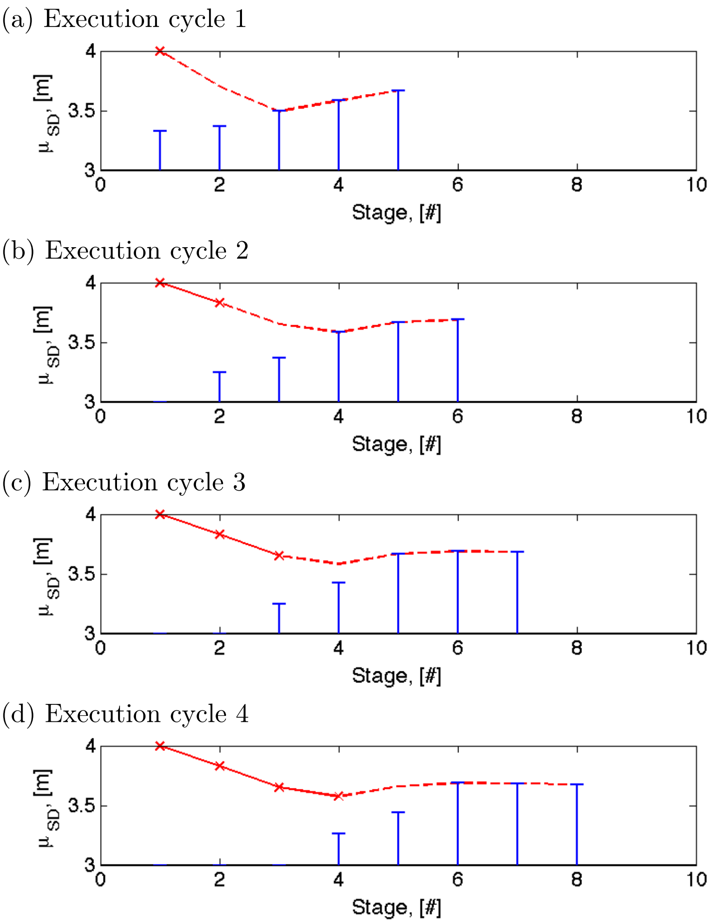

(a) Execution cycle 1

(b) Execution cycle 2

(c) Execution cycle 3

(d) Execution cycle 4

Figure 3.11: The separation distances predicted from the OLRHC solution (red, dashed) and the obtained separation distances from the execution of the solutions (red, solid) for cycles 1 through 4. The constraint tightening is plotted at the next 4 stages (at each planning cycle) in blue.



(a) Execution cycle 1

(b) Execution cycle 2

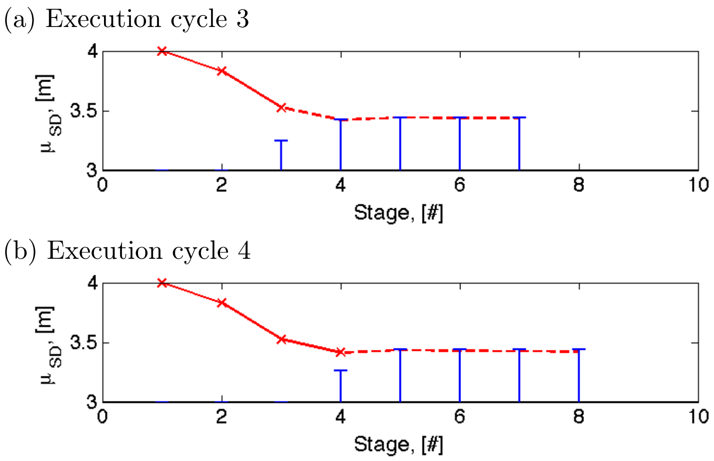
(c) Execution cycle 3

(d) Execution cycle 4

Figure 3.12: The separation distances predicted from the PCLRHC solution (red, dashed) and the obtained separation distances from the execution of the solutions (red, solid) for cycles 1 through 4. The constraint tightening is plotted at the next 4 stages (at each planning cycle) in blue.

3.5.3.2 Case 2: Incorrectly Assumed Reaction Time

A robot reaction time of $t_r = 1$ is incorrectly assumed, and the chance constraints are imposed accordingly (see Section 3.4.1.3). Since all future measurements are ignored in the OLRHC approach, this has no bearing on the results. For the PCLRHC approach, this is a significant failure mode since the approach assumes future measurements and plans accordingly. If a shorter reaction time is assumed, then the approach is overly confident and can end up in situations where future constraints cannot sufficiently be imposed. This is demonstrated for the above example in Figure 3.13, where the results from the first two planning (execution) cycles are presented. The constraint at cycle 2, stage 2 cannot be satisfied since the robot was overly aggressive. As a result, the robot's reaction time plays a cardinal role in guaranteeing the safety of the system.


(a) Execution cycle 3

(b) Execution cycle 4

Figure 3.13: The required separation distance constraint is violated at stage 2 of execution cycle 4. The robot is overly aggressive with the wrongly assumed robot reaction time and is unable to react to some of the disturbances in the system.

3.6 Summary

In this chapter, the Partially Closed-Loop Receding Horizon Control approach is derived from the dynamic programming algorithm for problems with imperfect state information. It is shown that the approximation does not introduce artificial information into the planning problem and that the least informative value of the anticipated measurements is used. The properties of the SRHC approach is presented, with emphasis on the conditioning of the chance constraints to ensure safe operation. The open-loop approximation to the problem results in the growth of the predicted state uncertainty, causing the problem to become overly constrained. This results in an infeasible problem for the OLRHC approach and the planned trajectory varies greatly from the executed trajectory. The PCLRHC approach utilizes the effect of the anticipated measurement on the uncertainty of the system, but discards the actual (unknown) value of the measurement. This allows the planner to manage the uncertainty growth as the system states are predicted. The similarity between the planned and executed trajectories is much higher than for the OLRHC approach. An important possible failure mode for the PCLRHC approach was presented.

Chapter 4

Systems with Simple, Independent Robot-Agent Models

The growth in the uncertainty of the predicted robot state during the planning problem causes open-loop approaches to obtain conservative solutions, even in static environments. This occurs because the anticipated information to be gained from future measurements is not accounted for during the planning process. The Partially Closed-Loop Receding Horizon Control (PCLRHC) approach was presented in Section 3.4.3. This approach uses the anticipated information to limit the uncertainty growth, resulting in less conservative solutions. The benefit of this algorithm over the Open-Loop Receding Horizon Control (OLRHC) approach for static environments was established in Section 3.5.1.

In this chapter, the PCLRHC approach is applied to the robot motion planning problem in dynamic environments. The focus is on agent models with simple behaviors whose actions are independent of the robot dynamics. The robot and agent states can be propagated separately (due to the independence). The agent states enter the problem through the collision chance constraints. Two planning problems are considered: systems with unknown states, and systems with unknown states and unknown, continuously varying parameters. For each of these cases, estimation results and approaches for imposing collision chance constraints are presented. Specifically, a novel approach for probabilistic collision avoidance is presented in Section 4.1.3. Simulation results are presented to illustrate the benefit of the PCLRHC approach for these types of problems.

4.1 Systems with Unknown State

The objective is to investigate the effect of taking future information into account in the planning in a dynamic setting. From Section 3.5.1 it is known that the OLRHC approach is very cautious due to the growth in uncertainty in the predicted state of the moving objects, even in static environments. In this section, the effect of using the PCLRHC approach is investigated for dynamic environments.

The robot and agent motion models are assumed to be independent.

Let x_i^R, u_i^R, and ω_i^R, respectively, denote the robot state, control, and process noise, as defined in Section 2.1.1. Let the process noise be distributed as $p(\omega_i^R|x_i^R, u_i^R) = p(\omega_i^R)$. Let y_i^R and ν_i^R be the robot measurement and measurement noise, as defined in Section 2.1.4. The measurement noise is distributed as $p(\nu_i^R|x_i^R) = p(\nu_i^R)$. The robot dynamic and measurement equations are described by the nonlinear functions:

$$\begin{aligned} x_i^R &= f^R(x_{i-1}^R, u_{i-1}^R, \omega_{i-1}^R) \\ y_i^R &= h^R(x_i^R, \nu_i^R). \end{aligned}$$

Similarly, for the j^{th} agent: let x_i^{Aj} and $\omega_i^{Aj} \sim p(\omega_i^{Aj}|x_i^{Aj}, u_i^{Aj}) = p(\omega_i^{Aj})$ be the agent state and process noise. Let y_i^{Aj} and $\nu_i^{Aj} \sim p(\nu_i^{Aj}|x_i^{Aj}) = p(\nu_i^{Aj})$ be the robot measurement and measurement noise. Let n_A be the total number of dynamic obstacles (agents) in the vicinity of the robot. The independent agent dynamic and measurement equations are described by the nonlinear functions:

$$\begin{aligned} x_i^{Aj} &= f^{Aj}(x_{i-1}^{Aj}, u_{i-1}^{Aj}, \omega_{i-1}^{Aj}) \\ y_i^{Aj} &= h^{Aj}(x_i^{Aj}, \nu_i^{Aj}). \end{aligned}$$

Estimation and chance constraints results are presented for the special cases of linear and non-linear systems with white Gaussian noise. Simulation results are presented that highlight the benefit of the PCLRHC approach over the OLRHC approach for moving objects governed by a random walk model.

4.1.1 Estimation

The probability distribution of interest during the estimation process is the joint distribution of the robot and agent states, given the measurements:

$$p(x_i^R, x_i^{A1}, \ldots, x_i^{An_A}|y_{1:i}^R, y_{1:i}^{A1}, \ldots, y_{1:i}^{An_A}, u_{0:i-1}^R) = p(x_i^R|y_{1:i}^R, u_{0:i-1}^R)p(x_i^{A1}|y_{1:i}^{A1}) \ldots p(x_i^{An_A}|y_{1:i}^{An_A}).$$

Use was made of the independence assumption of the robot and agent models to factor the distribution, and the robot and the agent states can independently be estimated. The agent states enter the planning problem when the collision chance constraints are enforced.

For systems (either the robot or the moving agents) with linear system equations and white, Gaussian noise, the optimal estimator for this system is the Kalman Filter (Section 2.4.3). Alternatively, the Information Filter [53] can be used. For systems with nonlinear system equations and white Gaussian noise, the optimal estimator for this system is the Extended Kalman Filter (Section 2.4.4). Alternatively, the Extended Information Filter and the Unscented Kalman Filter, or non-parametric filters, such as the Histogram Filter or Particle Filter, can be used [53]. When the noise terms are non-Gaussian, two approaches are used: approximate the noise as Gaussian, or use a non-parametric estimation algorithm. For the former, the extended Kalman Filter or Unscented Kalman Filter can be used. For the latter, the histogram filter or the particle filter can be applied [53].

4.1.2 Chance Constraints

To ensure collision avoidance between objects whose position is uncertain, it is necessary to introduce *chance constraints*: $P(x \notin \mathbb{X}_{free}) \leq \delta$. The positive scalar δ is the collision avoidance *level of confidence* and $\mathbb{X}_{free}$ denotes the free space where the constraints are not violated (when uncertainty effects are not considered). Often, the free space can be defined by $\mathbb{X}_{free} = \{x : c(x) \leq 0\}$, where $c(x)$ is a piecewise continuous function. In the stochastic setting, the constraints are specified as limits on the probability of constraint violation.

In the DP ISI formulation, when the control at stage i is selected, the only unknown quantity is the next measurement, y_{i+1} (see Remark 2.1). When selecting the control, u_i, the chance constraint must hold for all possible values of y_{i+1} and the chance constraint to be imposed is $P(x_{i+1} \notin X_{free}|\eta_i) \leq \delta_i$, where η_i is the information set at stage i (see Section 3.4.1.3).

When evaluating chance constraints, it is common to assume Gaussian distributions on system uncertainties. Two types of chance constraints are considered in this chapter: linear constraints of the form: $P(Ax > b) \leq \delta$ (e.g., velocity constraints) and collision constraints, $P(C) \leq \delta$, (e.g., between the robot and other agents). The collision condition, C, is defined below. For the former, the only unknown parameter is assumed to be the robot state, and standard results exist (presented in Section 2.3.1). For the latter, both the robot and object states are unknown. A novel analysis of this problem is presented below.

4.1.3 Probabilistic Collision Avoidance

Probabilistic obstacle avoidance can be formulated as a chance constraint. Blackmore [8] assumed known, static, convex, polyhedral obstacles and a Gaussian distribution for robot positional uncertainty. With these assumptions, collision avoidance can be formulated as a set of linear chance constraints (of the type described by Lemma 2.1) for each obstacle. Obstacle uncertainty is ignored in the framework if [8]. We wish to evaluate collision chance constraints when *both* the obstacle and robot locations are uncertain. It is assumed that the geometry of the robot and obstacles are known.

4.1.3.1 Probability of Collision

For simplicity's sake, let us initially assume a disc robot (radius ε) and a point obstacle[1]. The collision condition is defined as $C : x^A \in \mathcal{B}(x^R, \varepsilon)$, where $\mathcal{B}(z, r)$ is a ball of radius r centered at z. Let $V_{\mathcal{B}}$ be the volume of this ball. The probability of collision is defined as:

$$P(C) = \int_{x^R} \int_{x^A} I_C(x^A, x^R) p(x^R, x^A) dx^R dx^A \tag{4.1}$$

where I_C is the indicator function, defined as:

$$I_C(x^A, x^R) = \begin{cases} 1 & \text{if } x^A \in \mathcal{B}(x^R, \varepsilon) \\ 0 & \text{otherwise.} \end{cases}$$

Using the indicator function and the definition of the joint distribution, Eq. (4.1) can be written as:

$$P(C) = \int_{x^R} \left[\int_{x^A \in \mathcal{B}(x^R, \varepsilon)} p(x^A | x^R) dx^A \right] p(x^R) dx^R. \tag{4.2}$$

This integral function is difficult to evaluate for general robot and obstacle geometries. To gain some intuition, assume that the robot occupies a small volume and the obstacle is a point. The inner integral can be approximated with a constant value of the conditional distribution of the obstacle evaluated at the robot location, multiplied by the volume, $V_{\mathcal{B}}$, occupied by the robot:

$$\int_{x^A \in \mathcal{B}(x^R, \varepsilon)} p(x^A | x^R) dx^A \approx V_{\mathcal{B}} \times p(x^A = x^R | x^R).$$

The approximate probability of collision for small objects is therefore:

$$P(C) \approx V_{\mathcal{B}} \times \int_{x^R} p(x^A = x^R | x^R) p(x^R) dx^R. \tag{4.3}$$

4.1.3.2 Collision Chance Constraints for Systems with Gaussian Variables

If it is assumed that the robot and the obstacle (objects) are small and their position uncertainties can be described by independent Gaussian distributions, then the integral in Eq. (4.3) can be evaluated in closed form (Appendix C.1). Let $x_R \sim N(x_R; \hat{x}_R, \Sigma_R)$ be the normally distributed state representing the position of the robot, and $x_A \sim N(x_A; \hat{x}_A, \Sigma_A)$ be the independent position state of a point object. Then,

$$\int_{x^R} p(x^A = x^R | x^R) p(x^R) dx^R = \frac{1}{\sqrt{\det(2\pi\Sigma_C)}} \exp\left[-\frac{1}{2} \left(\hat{x}^R - \hat{x}^A\right)^T \Sigma_C^{-1} \left(\hat{x}^R - \hat{x}^A\right) \right]$$

[1] This analysis can readily be extended to the case where both the robot and obstacle have disc geometries.

where $\Sigma_C \triangleq \Sigma_R + \Sigma_A$ is the combined position covariance. This analysis additionally gives intuition about the appropriate combination of the covariances of the two distributions. The probability of collision is given by:

$$P(C) = \frac{1}{\sqrt{\det(2\pi\Sigma_C)}} \exp\left[-\frac{1}{2}\left(\hat{x}^R - \hat{x}^A\right)^T \Sigma_C^{-1} \left(\hat{x}^R - \hat{x}^A\right)\right] \times V_{\mathcal{B}}. \tag{4.4}$$

The objective is to convert the constraint $P(C) \leq 1 - \alpha$ into a constraint on the mean states of the robot and obstacle. From Eq. (4.4), the equivalent constraint is:

$$\begin{aligned} (\hat{x}^R - \hat{x}^A)^T \Sigma_C^{-1} (\hat{x}^R - \hat{x}^A) &\geq -2\ln\left(\sqrt{\det(2\pi\Sigma_C)}\frac{1-\alpha}{V_{\mathcal{B}}}\right) \\ &\triangleq \kappa \end{aligned}$$

where κ is a function of the level of certainty, the sizes of the robot and obstacle, and the combined covariance of the position states. This constraint is in the form of an ellipse around the obstacle that the robot has to avoid to ensure that the collision constraint is satisfied. A lookup table was created to account for the geometry of the objects. To simplify the current implementation, the lookup table was created specifically for disc objects of radius 0.5 m and a certainty level of $\alpha = 0.99$. Furthermore, κ, was catalogued according to the smallest eigenvalue of the combined position covariance, $\lambda \triangleq \min(eig(\Sigma_C))^2$. The true probability of collision was estimated for different covariances using a Monte Carlo simulation to obtain the corresponding values for κ (refer to Figure 4.1 for an illustration of the process for $\lambda = 0.1$ and $\lambda = 1$, respectively). The resulting values for κ are documented in Table 4.1.

Table 4.1: Collision constraint parameter, κ, for different values of λ, with $\alpha = 0.99$ and disc objects of radius $0.5m$

λ	0.005	0.01	0.05	0.1	0.5	1	5	10
$\kappa(\lambda)$	195	105	34.0	21.5	10.5	7.70	3.60	2.00

4.1.4 Simulation Results

Consider again the robotic system of Section 3.5.1. The first example (single dynamic obstacle) is used to gain intuition about the OL and PCL approaches in simple dynamic environments. The second example illustrates the potential benefit of the PCLRHC approach over the OLRHC approach for a specific scenario by navigating between two oncoming agents. In the third example, the robot

[2] This is a reasonable parameterization in this work since most of the position distributions had very similar eigenvalues (the uncertainty ellipses were roughly circular) and using the smaller eigenvalue results in a slightly conservative collision constraint.

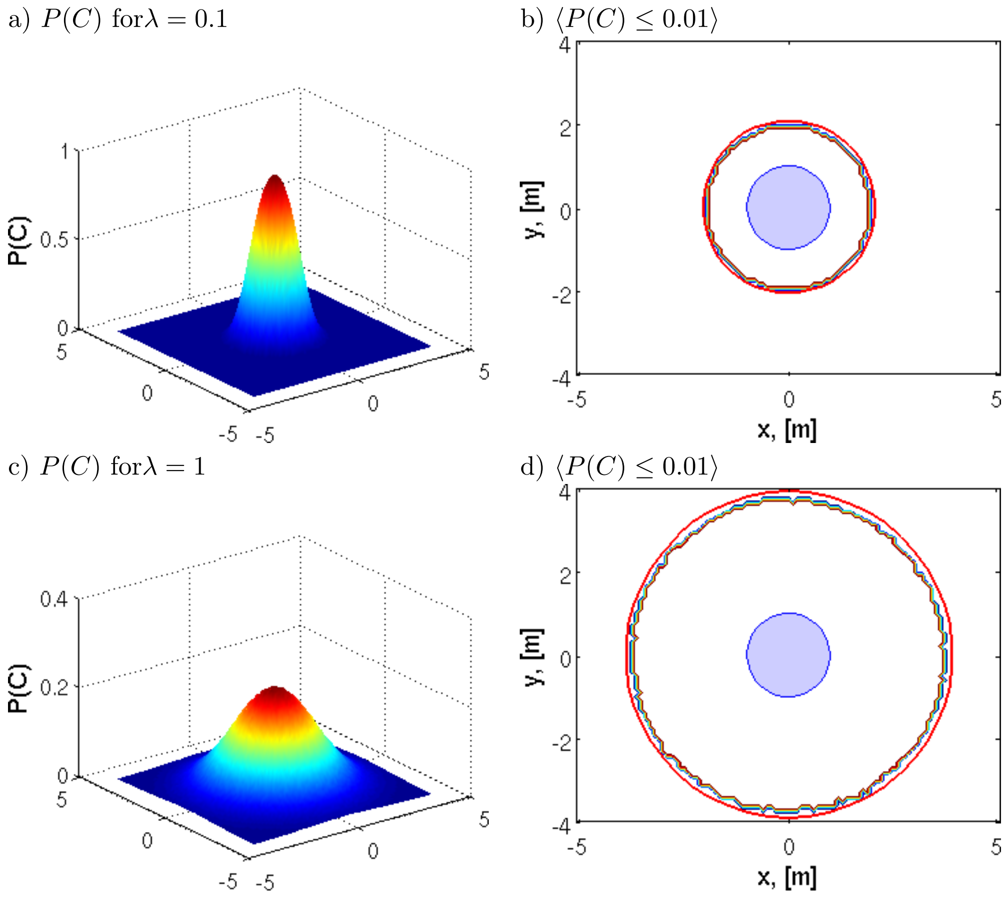


Figure 4.1: Lookup table generation: (a) estimate the probability of collision between objects of radius 0.5 m using a Monte Carlo simulation for $\lambda = 0.1$. The robot location is fixed at the origin and the obstacle location is varied. (b) Evaluate and plot the collision condition (contour plot) and find the ellipse that encloses the constraint violation area (red ellipse). The robot is plotted in blue. (c) and (d) correspond to $\lambda = 1$.

has to navigate among two agents that cross paths with the robot. A Monte-Carlo simulation is used to justify the use of the PCLRHC approach in the following chapters.

4.1.4.1 Example 1: Single Dynamic Obstacle

A single dynamic obstacle moves with a random walk model (Section B.1.3) with $W_A = 0.01 \times I_2$. A linear position measurement model (Section B.2.2) is used with $V_A = 0.01 \times I_2$. The goal state is $x_G = [10\ 1.5\ 0\ 0]$. The cost function and constraints of Section 3.5.1 are used. The robot initial state is $x_0^R \sim N(\hat{x}_{0|0}^R, \Sigma_{0|0}^A)$, where $\hat{x}_{0|0}^R = [0\ 1.5\ 1\ 0]^T$ and $\Sigma_{0|0}^R = 0.01 \times I_4$. The agent initial state is $x_0^A \sim N(\hat{x}_{0|0}^A, \Sigma_{0|0}^A)$, where $\hat{x}_{0|0}^A = [1\ 0\ 1.2\ 0]^T$ and $\Sigma_{0|0}^A = 0.01 \times I_4$. Finally, the collision chance constraint is imposed at each stage with $\delta_c = 0.01$.

Figure 4.2 shows the first stage *planned* trajectory for the OLRHC approach, with the robot in blue and the agent in red. The robot's planned trajectory (green) and the agent's predicted

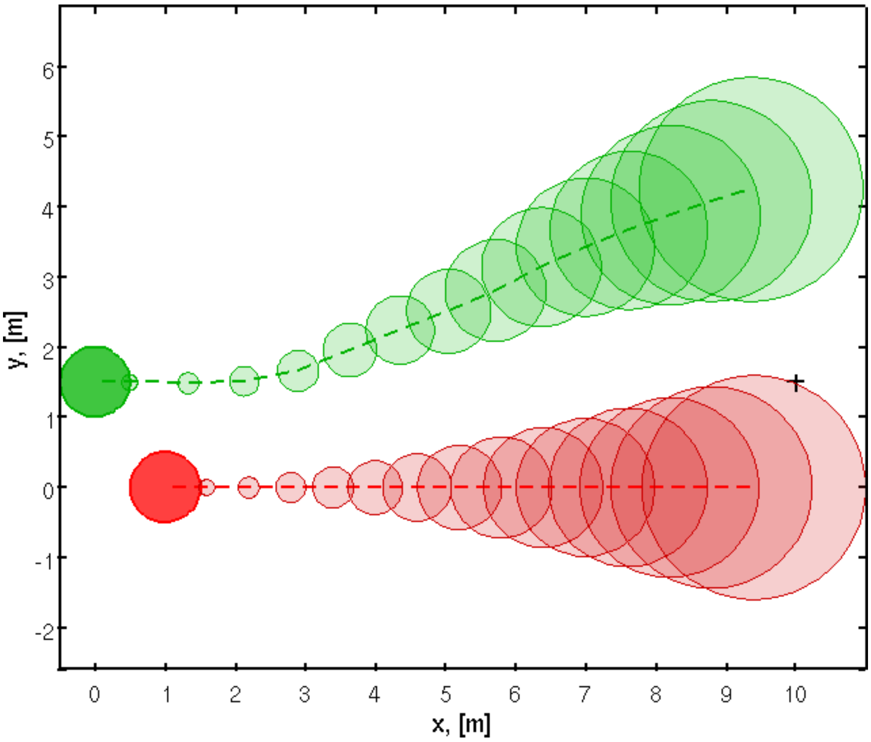


Figure 4.2: The planned trajectory with the OLRHC approach (green) and the predicted trajectory for the agent (red) with uncertainty ellipses. The growth in uncertainty forces the robot away from the agent, and the plan does not reach the goal.

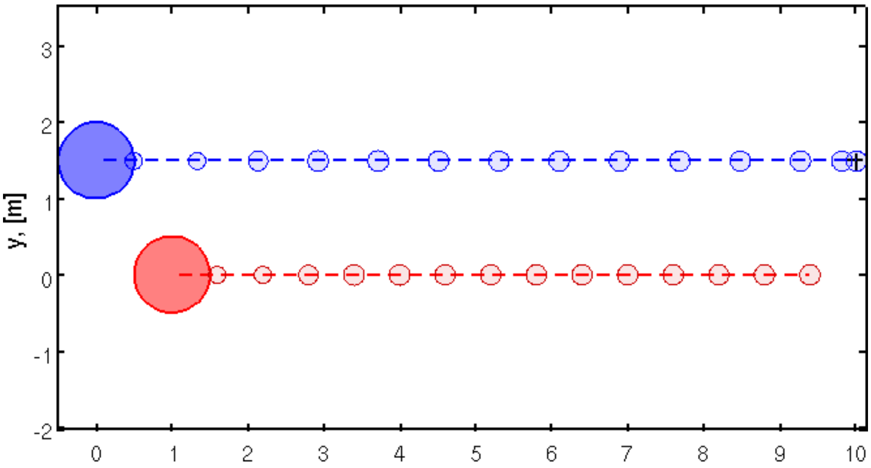


Figure 4.3: The planned trajectory with the PCLRHC approach (blue) and the predicted agent trajectory (red)

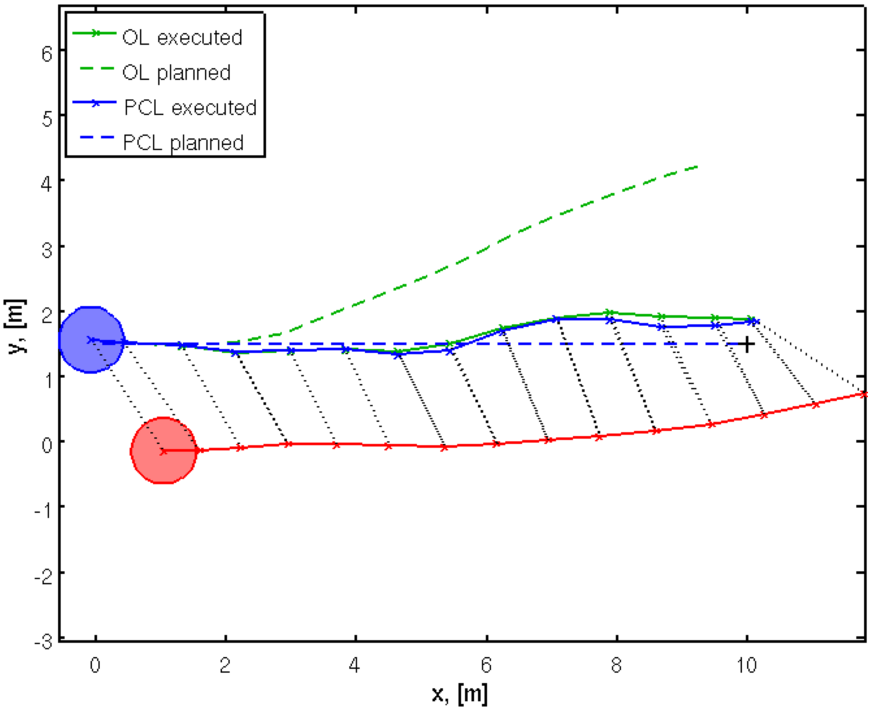


Figure 4.4: The planned (dashed) and executed (solid) paths for the OLRHC (green) and PCLRHC (blue) approaches and the predicted agent trajectorry (red). The executed trajectories are very similar, but the OLRHC approach has to rely on the outerloop feedback to obtain a reasonable executed trajectory. The executed and planned paths for the PCLRHC approach are much closer.

trajectory (red) are plotted with 1-σ uncertainty ellipses. The location uncertainty of both the agent and robot grows since future information is ignored. The problem becomes highly constrained and the resulting solution is very conservative. The trajectory does not reach the goal.

The initially planned trajectory for the PCLRHC approach is shown in Figure 4.3, using the same coloring scheme as above. The robot and agent state uncertainties remain bounded since the effect of the future measurements are taken into account during this first stage plan.

The *executed* trajectories (solid) and the initial planned or predicted trajectories (dashed) are plotted in Figure 4.4. The time-correlated locations of the robot and agent (for the executed trajectories) are indicated by the dotted lines. While the two approaches produce similar executed trajectories, the initially planned and subsequently executed trajectories for the OLRHC approach differ greatly. This discrepancy implies that the OLRHC approach relies on the outer-loop feedback mechanism to obtain a reasonable executed trajectory. The PCLRHC solution is much closer to the executed trajectory, and again, the PCLRHC approach deviates less from the preferred plan during execution. One might say that the PCLRHC plan is more aggressive, as it does not shy away from keeping close contact with the agent, as the planner knows it will be taking future measurements that help the robot avoid collision. Like the previous example, the outer-loop feedback mechanism allows the PCLRHC approach to correct for the actual realization of the noise and measurements, but the algorithm does not only rely on this to obtain a reasonable executed trajectory.

4.1.4.2 Example 2: 2 Oncoming Dynamic Obstacles

Two dynamic obstacles move towards the robot with random walk models (Section B.1.3) with $W_A = 0.01 \times I_2$. A linear position measurement model (Section B.2.2) is used for the agent with $V_A = 0.01 \times I_2$. The goal state is $x_G = [10\ 0\ 0\ 0]$. The cost function and constraints of Section 3.5.1 are used. The robot initial state is $x_0^R \sim \mathcal{N}(\hat{x}_{0|0}^R, \Sigma_{0|0}^A)$, where $\hat{x}_{0|0}^R = [0\ 0\ 1\ 0]^T$ and $\Sigma_{0|0}^R = 0.01 \times I_4$. The first agent initial state is $x_0^{A1} \sim \mathcal{N}(\hat{x}_{0|0}^{A1}, \Sigma_{0|0}^{A1})$, where $\hat{x}_{0|0}^{A1} = [12\ 2\ -1\ 0]^T$ and $\Sigma_{0|0}^{A1} = 0.01 \times I_4$ and the second agent initial state is $x_0^{A2} \sim \mathcal{N}(\hat{x}_{0|0}^{A2}, \Sigma_{0|0}^{A2})$, where $\hat{x}_{0|0}^{A2} = [12\ -2\ -1\ 0]^T$ and $\Sigma_{0|0}^{A2} = 0.01 \times I_4$. Finally, the collision chance constraint is imposed at each stage with $\delta_c = 0.01$.

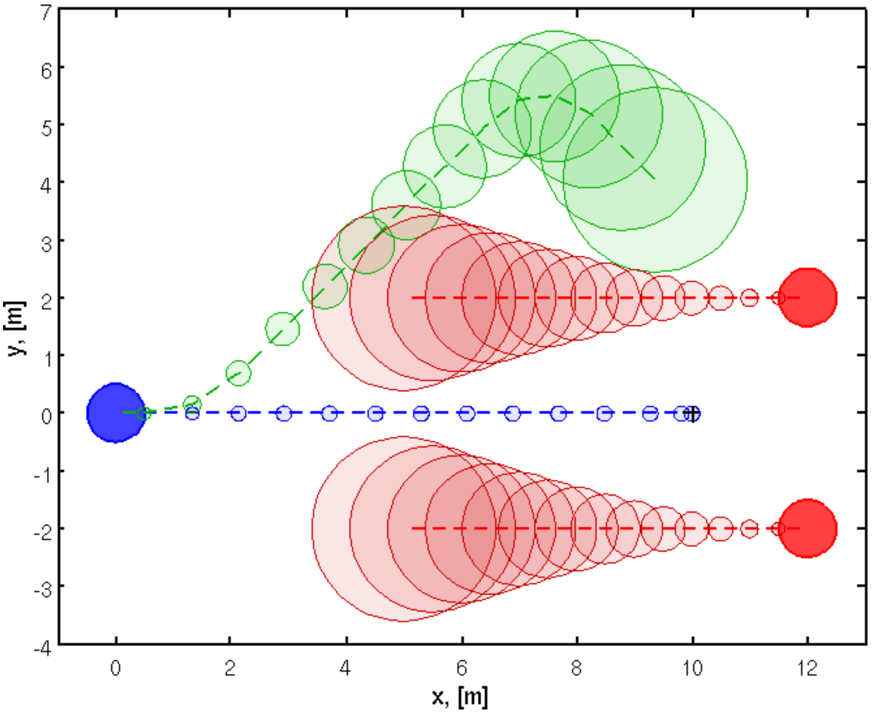


Figure 4.5: The planned trajectories of the robot with the OLRHC (green) and PCLRHC (blue) approaches with uncertainty ellipses are plotted. The predicted trajectory for the agent (red) with the OL uncertainty ellipses is plotted. The robot and agents are indicated with blue and red circles, respectively.

Figure 4.5 shows the first stage *planned* trajectory for the robot for the OLRHC approach (green), with the 1-σ uncertainty ellipses. The trajectory does not reach the goal and the robot cannot pass between the agents because of the growth in uncertainty. The robot plans around both agents. The solution from the PCLRHC approach (blue), also shown in Figure 4.5, is less conservative and moves the robot between the two agents. This is an example where the solutions from the two approaches are qualitatively very different.

The sequence of *executed* trajectories (solid) and the initial planned (dashed) are plotted in Figure

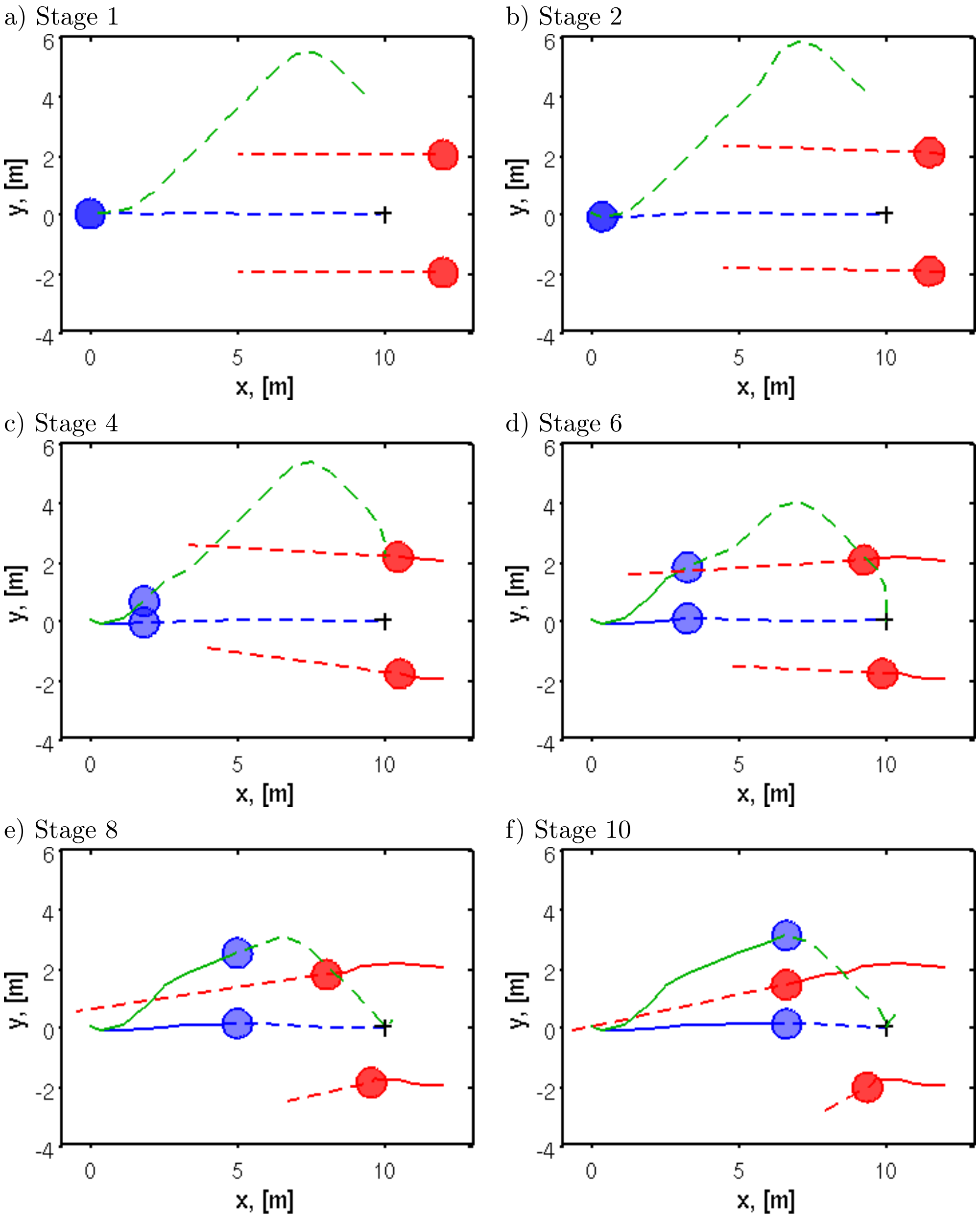


Figure 4.6: Executed (solid) and planned or predicted (dashed) trajectories at stages 1, 2, 4, 6, 8, and 10 using the OLRHC (green) and PCLRHC (blue) approaches. With the OLRHC approach, the robot is forced to move around the agents (red), whereas the robot moves between the agents with the PCLRHC approach.

4.6. The PCLRHC approach takes the robot between the agents, while the OLRHC approach takes the robot around both agents. The OLRHC executed path is significantly longer than the PCLRHC trajectory, illustrating the superiority of the PCLRHC approach over the OLRHC approach in multi-agent scenarios.

4.1.4.3 Example 3: Monte-Carlo Simulation of Robot with 2 Crossing Agents

Two dynamic obstacles cross the space between the robot and the goal. The robot and agents have random walk models (Section B.1.3) with $W_R = 0.01 \times I_2$ and $W_A = 0.01 \times I_2$, respectively. The linear position measurement model (Section B.2.2) is used for the robot and agent with $V_R = 0.01 \times I_2$ and $V_A = 0.01 \times I_2$, respectively. The goal state is $x_G = [12\ 0\ 0\ 0]$. The cost function and constraints of Section 3.5.1 are used. The robot initial state is $x_0^R \sim \mathcal{N}(\hat{x}_{0|0}^R, \Sigma_{0|0}^A)$, where $\hat{x}_{0|0}^R$ is defined below and $\Sigma_{0|0}^R = 0.01 \times I_4$. The first agent initial state is $x_0^{A1} \sim \mathcal{N}(\hat{x}_{0|0}^{A1}, \Sigma_{0|0}^{A1})$, where $\hat{x}_{0|0}^{A1}$ is defined below and $\Sigma_{0|0}^{A1} = 0.01 \times I_4$, and the second agent initial state is $x_0^{A2} \sim \mathcal{N}(\hat{x}_{0|0}^{A2}, \Sigma_{0|0}^{A2})$, where $\hat{x}_{0|0}^{A2}$ is defined below and $\Sigma_{0|0}^{A2} = 0.01 \times I_4$. Finally, the collision chance constraint is imposed at each stage with $\delta_c = 0.01$.

The simulation is repeated 200 times with randomized initial conditions: the robot x-location is fixed at the origin, and the y-location is sampled (from a uniform distribution) from $[-2,\ 2]$. The initial velocity is fixed at 1.2 m/s and the initial heading is uniformly sampled from $[-22.5°,\ 22.5°]$ (the robot moves roughly horizontally from left to right). For agent 1, the y-location is fixed at 6, and the x-location is sampled (from a uniform distribution) from $[4,\ 8]$. The initial velocity is fixed at 1 m/s and the initial heading is uniformly sampled from $[-120°,\ -75°]$ (agent 1 moves roughly from north to south). For agent 2, the y-location is fixed at -6, and the x-location is sampled (from a uniform distribution) from $[4,\ 8]$. The initial velocity is fixed at 1 m/s and the initial heading is uniformly sampled from $[75°,\ 120°]$ (agent 2 moves roughly from south to north). A typical executed scenario is given in Figure 4.7.

In order to evaluate the practical benefit of the PCLRHC approach[3], the executed trajectories are compared to the OLRHC approach results. From the histograms of the executed path lengths for the OLRHC (Figure 4.8) and PCLRHC (Figure 4.9) approaches, the PCLRHC approach is more often able to find direct paths to the goal (from the height of the peak centered around 12.5) than the OLRHC approach. A larger second peak for the OLRHC approach indicates that the approach must react to the agents more often, resulting in longer paths. In fact, the mean of the obtained path lengths for the PCLRHC approach is 13.47 ± 1.46 m, compared to 14.58 ± 1.43 m for the OLRHC approach. On average, the PCLRHC approach obtains shorter executed paths.

On a case-by-case comparison, the PCLRHC approach finds shorter paths in 72.0% of the cases,

[3] In the previous examples it was already shown that the PCLRHC approach is safe if an appropriate reaction time is assumed, the planned solutions are less conservative than the OL approach, and the planned and executed trajectories are more similar in the PCLRHC approach.

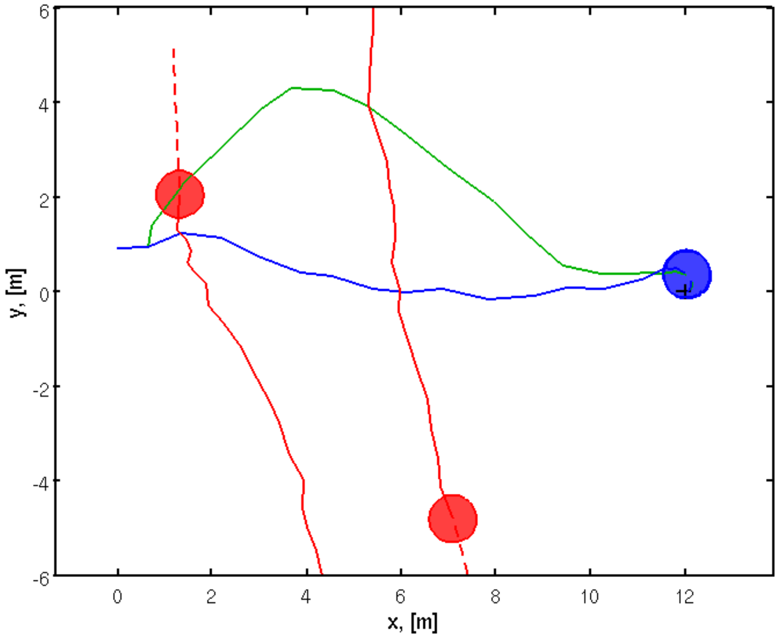


Figure 4.7: Crossing scenario: the robot (blue) moves towards the goal (left to right). Two agents (red) cross the space between the robot and the goal. Executed trajectories (solid) and planned/predicted trajectories (dashed) are plotted for the OLRHC (green) and PCLRHC (blue) approaches. The initial conditions of the robot and agents are randomized.

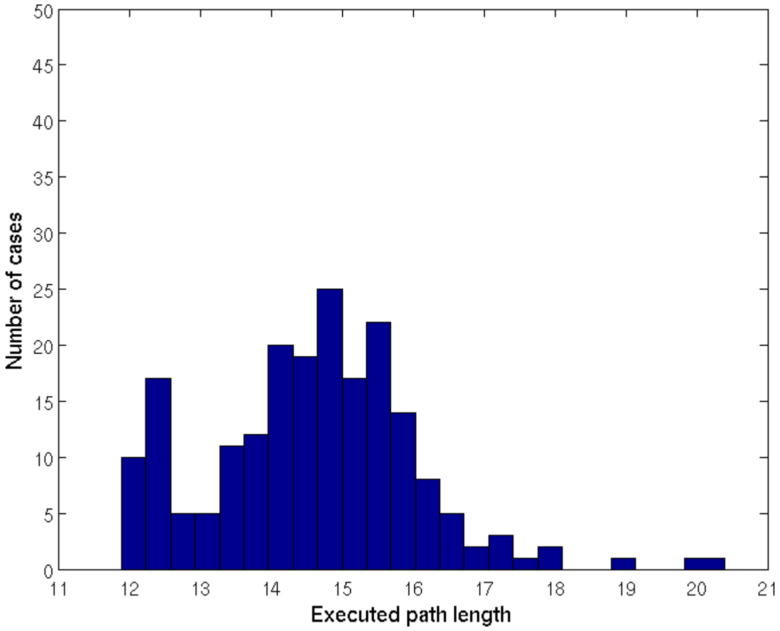


Figure 4.8: Histogram of executed path lengths for the OLRHC approach. The first peak (centered around 12.5) corresponds to cases where the robot is able to move directly to the goal. The second peak (centered around 15) corresponds to the cases where the robot has to maneuver around the agents.

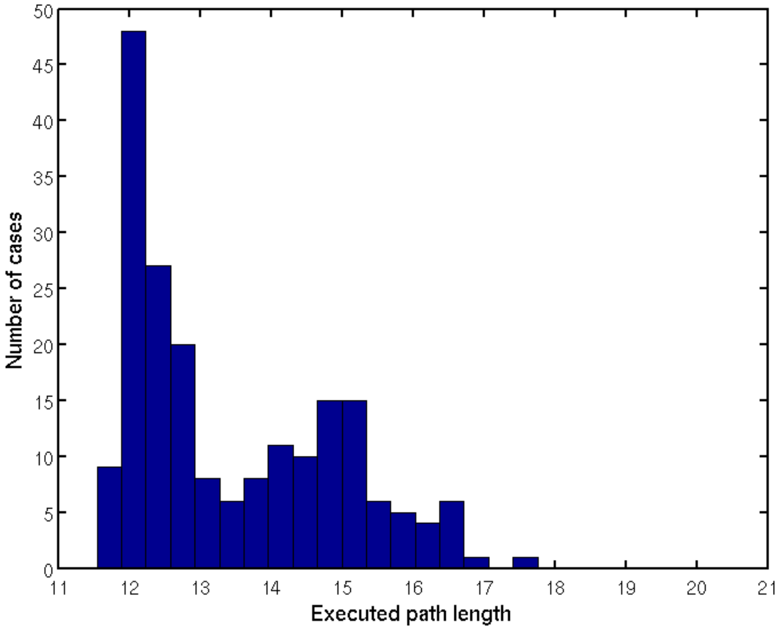


Figure 4.9: Histogram of executed path lengths for the PCLRHC approach. The first peak (centered around 12.5) corresponds to cases where the robot is able to move directly to the goal. The second peak (centered around 15) corresponds to the cases where the robot has to maneuver around the agents.

with at least a 10% improvement in 37.5% and at least a 20% improvement in 17.5% of the cases. Figure 4.7 is an example of a significant improvement where the PCLRHC approach is able to make direct progress towards the goal and the OLRHC approach plans around the obstacles (due to the conservatism).

In the current implementation, the constrained nonlinear optimization package that is used is very succeptible to local minima. At planning cycle $i+1$, the optimal solution from cycle i is used to seed the optimizer. It is possible for the planner to get stuck in a local minimum, resulting in a sub-optimal executed trajectory. This is true for both the PCLRHC and OLRHC approaches. An example of this case is given in Figure 4.10 where the PCLRHC approach is unable to choose a possibly more efficient qualitative solution. An implementation that handles local minima more readily is desired, especially since these minima are very common in scenarios with multiple dynamic agents.

In conclusion, there is an overall benefit to using the PCLRHC approach over the OLRHC approach (as established from the Monte-Carlo simulation). In some cases the benefit is very substantial (>20% improvement). In the sections to follow, the simulation results will focus on these cases of significant benefit.

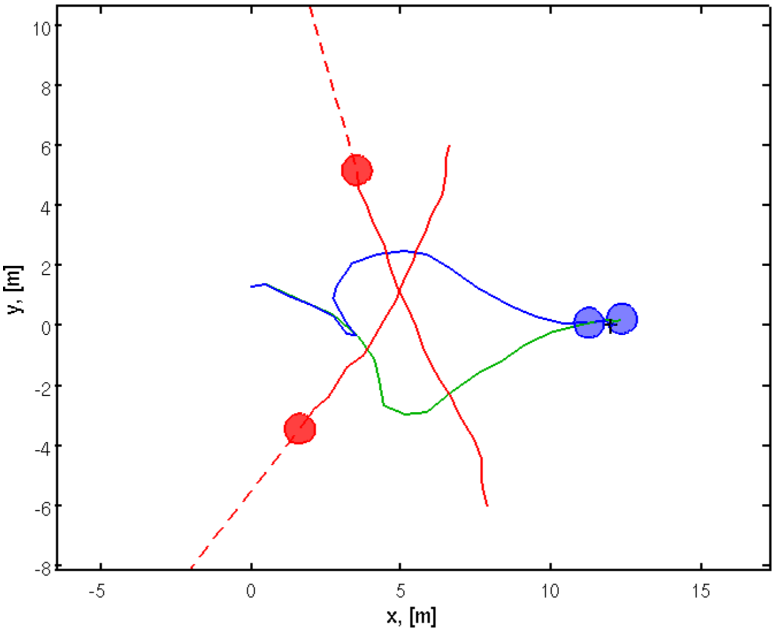


Figure 4.10: The PCLRHC approach gets stuck in a local minima (an artifact of the implementation, and not the PCLRHC approach), resulting in suboptimal executed path

4.2 Systems with Unknown States and Unknown, Continuous Parameters

In most practical applications, the system models are not perfectly known. Process or measurement noise is one way to capture the uncertainty in the system. However, in many cases it is possible to derive parametric models, where some of these parameters are unknown. For example, physical parameters effecting the dynamics of the system might be unknown (such as spring stiffnesses) or the effectiveness of the control inputs are only approximately known.

Assume the nonlinear dynamic and measurements model for an object (either the robot or an agent) of the form:

$$x_{i+1} = f(x_i, \theta_i, u_i, \omega_i) \tag{4.5}$$

$$y_i = h(x_i, \theta_i, \nu_i) \tag{4.6}$$

where θ_i is the set of n_θ unknown parameters. The disturbances are distributed according to $p(\omega_i|x_i, \theta_i, u_i)$ and the measurement noise is distributed according to $p(\nu_i|x_i, \theta_i)$.

Continuously varying uncertain parameters are considered here and discrete-valued parameters are considered in Section 5.1. Two problems are of interest here: (i) accounting for the parameter uncertainty in the models but not estimating their values, and (ii) estimating the parameters with the rest of the unknown states of the system. Necessary extensions to standard estimation results for these cases are presented in addition to incorporating these uncertainties in the planning process. No extensions to the chance constraints results are necessary.

4.2.1 Estimation: Accounting for Uncertain Parameters

If it is assumed that the unknown parameters have a known, fixed distribution, then the uncertainty associated with the parameters can be accounted for during the planning process. Assume that $p(\theta_i|x_i, u_i) = N(\hat{\theta}, \Lambda)$ is the constant, known, fixed distribution of the uncertain parameters. The uncertain parameters either enter the equation as a product of the state, or this term is independent of the states. Since the uncertain parameters are not to be estimated with the state, these parameters are treated as noise terms.

4.2.1.1 Additive Uncertain Parameters

When the terms in the system equations with the uncertain parameters are independent of the system state (additive), these terms are treated as non-zero mean Gaussian noise. Assume the following dynamic and measurement equations:

$$\begin{aligned} x_{i+1} &= Ax_i + Bu_i + F_\omega \omega_i + F_\theta \theta_i \\ y_i &= Cx_i + H_\nu \nu_i + H_\theta \theta_i \end{aligned} \tag{4.7}$$

where $\omega_i \sim N(0, W)$ and $\nu_i \sim N(0, V)$ and are independent. The parameters are assumed to be independent of the states and noise terms. Since the parameter distributions are assumed to be known, they can be converted to zero-mean variables by writing them as the sum of the mean and another zero-mean random variable: $\theta_i = \tilde{\theta}_i + \hat{\theta}$, where $\tilde{\theta}_i \sim N(0, \Lambda)$ and $\hat{\theta}$ is known. The zero-mean, independent noise terms can be combined into a single variable:

$$\bar{\omega}_i = F_\omega \omega_i + F_\theta \tilde{\theta}_i \sim N(0, F_\omega W F_\omega^T + F_\theta \Lambda F_\theta^T) \tag{4.8}$$

and

$$\bar{\nu}_i = H_\nu \nu_i + H_\theta \tilde{\theta}_i \sim N(0, H_\nu V H_\nu^T + H_\theta \Lambda H_\theta^T).$$

Define the known terms $c_i \triangleq F_\theta \hat{\theta}_i$ and $d_i \triangleq H_\theta \hat{\theta}_i$.

$$\begin{aligned} x_{i+1} &= Ax_i + Bu_i + \bar{\omega}_i + c_i \\ y_i &= Cx_i + \bar{\nu}_i + d_i. \end{aligned}$$

The Kalman Filter is used to estimate this system (Section 2.4.3).

4.2.1.2 Multiplicative Uncertain Parameters

In many practical situations the uncertain parameters are expected to be multiplied with the unknown state. For example, the spring stiffness of a suspension system is multiplied by the displacement. Assume the following dynamic and measurement equations:

$$x_{i+1} = Ax_i + Bu_i + F\omega_i + \sum_{k=1}^{n_\theta} M^{(k)}\theta_i^{(k)} x_i \tag{4.9}$$

$$y_i = Cx_i + H\nu_i + \sum_{k=1}^{n_\theta} G^{(k)}\theta_i^{(k)} x_i \tag{4.10}$$

where $p(\omega_i|x_i, u_i) = N(0, W)$ and $p(\nu_i|x_i) = N(0, V)$ and are independent. Each component of the parameter is assumed to be normally distributed with zero-mean[4]: $p(\theta_{i-1}^{(k)}|x_{i-1}, u_{i-1}) = N(0, \sigma_\theta^2)$ $\forall k = 1, \ldots, n_\theta$. The product of two Gaussian variables is non-Gaussian and an optimal estimator is not generally available. One common approach to obtain an approximate estimator is to assume that the estimator is in the form of a Luenberger estimator (e.g., [54]):

$$\hat{x}_{i|i} = \hat{x}_{i|i-1} + K_i(y_i - \hat{y}_{i|i-1}).$$

It can then be shown (see Appendix D.1) that the resulting estimator has the following form:
Prediction step:

$$\hat{x}_{i|i-1} = A\hat{x}_{i-1|i-1} + Bu_{i-1} \tag{4.11}$$

$$\Sigma_{i|i-1} = A\Sigma_{i-1|i-1}A^T + FWF^T + \sum_{k=1}^{n_\theta} \sigma_\theta^2 M^{(k)} \left(\Sigma_{i-1|i-1} + \hat{x}_{i-1|i-1}\hat{x}_{i-1|i-1}^T\right) M^{(k)T}. \tag{4.12}$$

Measurement update step:

$$\hat{x}_{i|i} = \hat{x}_{i|i-1} + K_i(y_i - C\hat{x}_{i|i-1}) \tag{4.13}$$

$$\Sigma_{i|i} = (I - K_iC)\Sigma_{i|i-1} \tag{4.14}$$

[4] In general the parameters will not have zero mean. However, the system can be converted to a zero-mean parameter system since the parameter distributions are assumed to be known and fixed. Each component of the parameter can be written as, $\theta^{(k)} = \tilde{\theta}^{(k)} + \hat{\theta}^{(k)}$, where $\tilde{\theta}^{(k)} \sim N(0, \sigma_{\theta,k}^2)$ and $\hat{\theta}^{(k)}$ is known. Thus, the system can be written as

$$x_{i+1} = \bar{A}x_i + Bu_i + F_\omega\omega_i + \sum_{k=1}^{n_\theta} F_\theta^{(k)}\tilde{\theta}_i^{(k)} x_i$$

$$y_i = \bar{C}x_i + H_\nu\nu_i + \sum_{k=1}^{n_\theta} H_\theta^{(k)}\tilde{\theta}_i^{(k)} x_i$$

where $\bar{A} = A + \sum_{k=1}^{n_\theta} F_\theta^{(k)}\hat{\theta}_i^{(k)}$ and $\bar{C} = C + \sum_{k=1}^{n_\theta} H_\theta^{(k)}\hat{\theta}_i^{(k)}$.

where

$$\begin{aligned}\Gamma_{i|i-1} &= C\Sigma_{i|i-1}C^T + HVH^T + \sum_{k=1}^{n_\theta} \sigma_\theta^2 G^{(k)} \left(\Sigma_{i|i-1} + \hat{x}_{i|i-1}\hat{x}_{i|i-1}^T\right) G^{(k)T} \\ K_i &= \Sigma_{i|i-1}C^T\Gamma_{i|i-1}^{-1}.\end{aligned}$$

4.2.2 Joint Estimation of Uncertain, Continuous Parameters and States

The *dual control* community is interested in controlling a system with unknown parameters and states. The controls must be chosen to optimally gather information about the model parameters, and drive the system towards the goal. Fel'dbaum first introduced the *dual effect* of control actions: actions affect the information gathered about the system, as well as progress the system towards the goal. The reader is referred to the survey by Filatov and Unbehauen [21] and the paper by Londoff et al. [39].

Consider the following system:

$$\begin{aligned} x_{i+1} &= f_x(x_i, \theta_i, u_i, \omega_i^x) \\ y_i^x &= h_x(x_i, \theta_i, \nu_i^x). \end{aligned}$$

The white, Gaussian noise terms are distributed according to $\omega_i^x \sim N(0, \Sigma_\omega)$ and $\nu_i^x \sim N(0, \Sigma_\nu)$ and are independent of the other noise, state, and parameters. The unknown parameters are to be estimated along with the unknown state. Define the augmented state as: $z_i = [x_i\ \theta_i]^T$. A dynamic update model for the parameters must be assumed:

$$\theta_{i+1} = f_\theta(\theta_i, \omega_i^\theta)$$

where the disturbance, $\omega_i^\theta \sim N(0, \Omega_\omega)$, is added for numerical stability of the estimation process. The augmented system has a dynamic equation of the form:

$$z_{i+1} = \begin{bmatrix} x_{i+1} \\ \theta_{i+1} \end{bmatrix} = \begin{bmatrix} f_x(x_i, \theta_i, u_i, \omega_i^x) \\ f_\theta(\theta_i, \xi_i) \end{bmatrix} \triangleq f(z_i, u_i, \omega_i)$$

where

$$\omega_i \triangleq \begin{bmatrix} \omega_i \\ \xi_i \end{bmatrix} \sim N\left(0, \begin{bmatrix} \Sigma_\omega & 0 \\ 0 & \Omega_\omega \end{bmatrix}\right) = N\left(0, W\right).$$

A measurement model for the parameters must be assumed (in most cases the parameters will not be directly measureable):

$$y_i^\theta = h^\theta(\theta_i, \nu_i^\theta)$$

where the measurement noise, $\nu_i^\theta \sim N(0, \Omega_\nu)$, is assumed to be independent of the other noise terms, states, and parameters. The augmented measurement is defined as:

$$y_i = \begin{bmatrix} y_i^x \\ y_i^\theta \end{bmatrix} = \begin{bmatrix} h^x(x_i, \theta_i, \nu_i^x) \\ h^\theta(\theta_i, \nu_i^x) \end{bmatrix} \triangleq h(z_i, \nu_i)$$

where

$$\nu_i \triangleq \begin{bmatrix} \nu_i^x \\ \nu_i^\theta \end{bmatrix} \sim N\left(0, \begin{bmatrix} \Sigma_\nu & 0 \\ 0 & \Omega_\nu \end{bmatrix}\right) = N\left(0, V\right).$$

The resulting system is either linear or nonlinear with white, Gaussian noise. The methods developed in Section 4.1 can be directly applied.

Two key issues, that are not addressed here, are observability and persistence of excitation. Observability is a property of the system equations, specifically the A-C matrix pair. The reader is referred to standard texts on Control Theory for a discussion on observability (e.g., [25, 31]). Persistence of excitation requires a sufficiently rich excitation for the parameter estimates converge to their true values. This is the topic of Adaptive Control [50].

4.2.3 Simulation Results

The unknown parameters increase the uncertainty in the system. It is expected that the OLRHC approach will be even more conservative due to the growth of the open-loop covariance associated with the unknown parameters. The hope is that the PCLRHC approach will be able to obtain reasonable solutions for this increased-uncertainty case as well. In this example, the parameter uncertainty is accounted for, but the parameter values are not to be estimated with the states.

Assume a random walk dynamic model (Section B.1.3) for the robot with $W_R = 0.01 \times I_2$. A linear position measurement model (Section B.2.2) is used, with $V_R = 0.01 \times I_2$. The agent is modeled as moving towards a destination point (Appendix B.1.4) with $W_A = 0.01 \times I_2$, $k_x = 0.1$, $k_y = 0.1$, and $\Delta t = 0.5$. A linear position measurement model is used for the agent, with $V_A = 0.01 \times I_2$. The initial states, goal state, cost function, and constraints of Section 4.1.4.1 are used.

The destination is the uncertain parameter in this example ($\theta_i = t_i$), and has a fixed distribution $t_i \sim N(\hat{t}, \Sigma_t)$, with $\hat{t} = [\ 10 \quad 0\ ]^T$ and $\Sigma^t = I_2$. The additional uncertainty enters the system additively through the $N = F^\theta$ matrix. The elements in the matrix are small ($k_x \Delta t = 0.05$) and the covariance of the process noise is increased from $tr(\Sigma_\omega) = 0.02$ to $tr(F_\omega \Sigma_\omega F_\omega^T + N \Sigma_t N^T) = 0.025$ (according to Eq. (4.8)).

Figure 4.11 shows the first stage *planned trajectory* for the OLRHC approach. The robot's planned trajectory (green, dashed) and the agent's predicted trajectory (red, dashed) are plotted with 1-σ uncertainty ellipses. The uncertainty of the agent destination is represented by the cyan

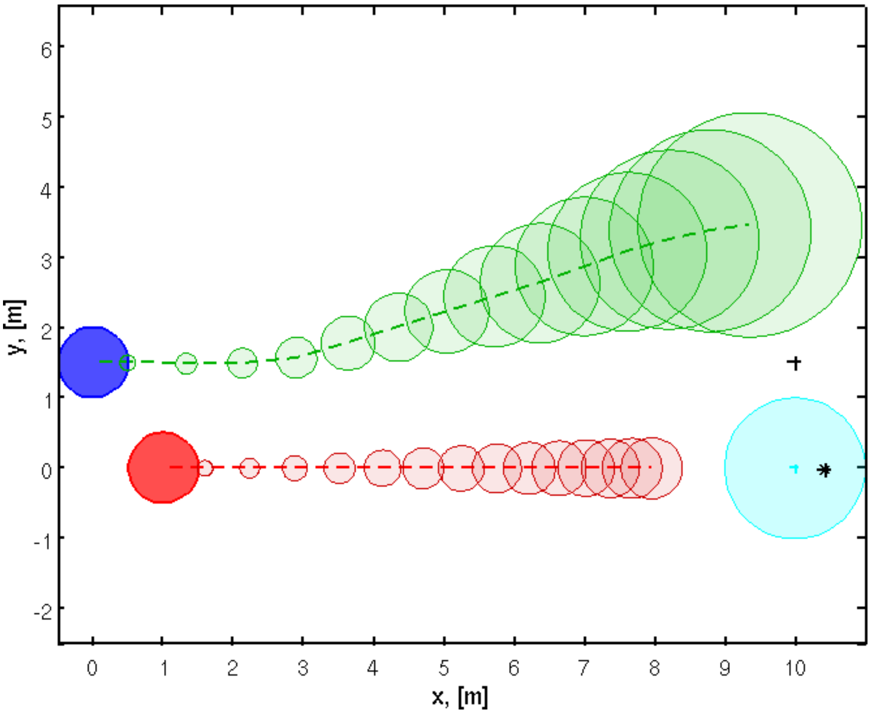


Figure 4.11: The planned trajectory with the OLRHC approach (green) and the predicted trajectory for the agent (red) with uncertainty ellipses. The uncertainty of the agent destination is represented by the cyan ellipse. The growth in uncertainty forces the robot away from the agent, and the plan does not reach the goal.

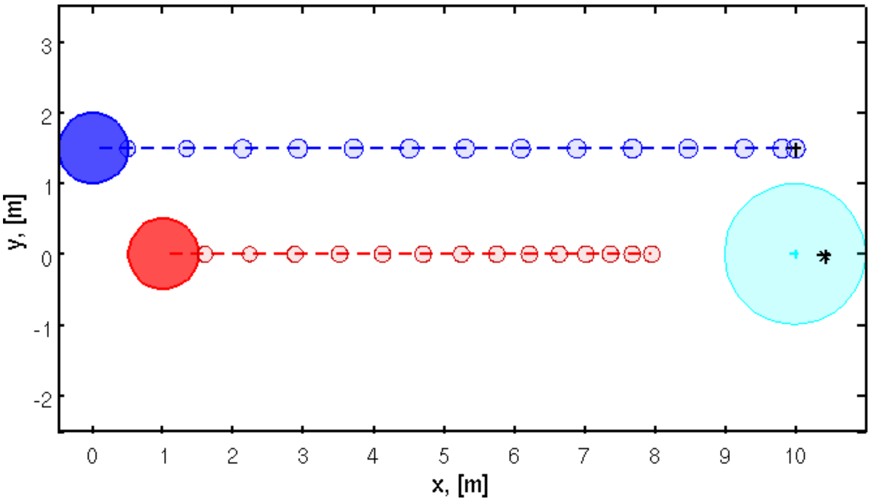


Figure 4.12: The planned trajectory with the PCLRHC approach (blue) and the predicted agent trajectory (red). The uncertainty due to the unknown agent destination is represented by the cyan ellipse.

ellipse. The location uncertainty of both the agent and robot grows since future information is ignored. The problem becomes highly constrained and the resulting solution is very conservative. The trajectory does not reach the goal.

The initially planned trajectory for the PCLRHC approach is shown in Figure 4.12. The robot's planned trajectory (blue, dashed) and the agent's predicted trajectory (red, dashed) are plotted with 1-σ uncertainty ellipses. The robot and agent state uncertainties remain bounded since the effect of the future measurements are taken into account during this first stage plan.

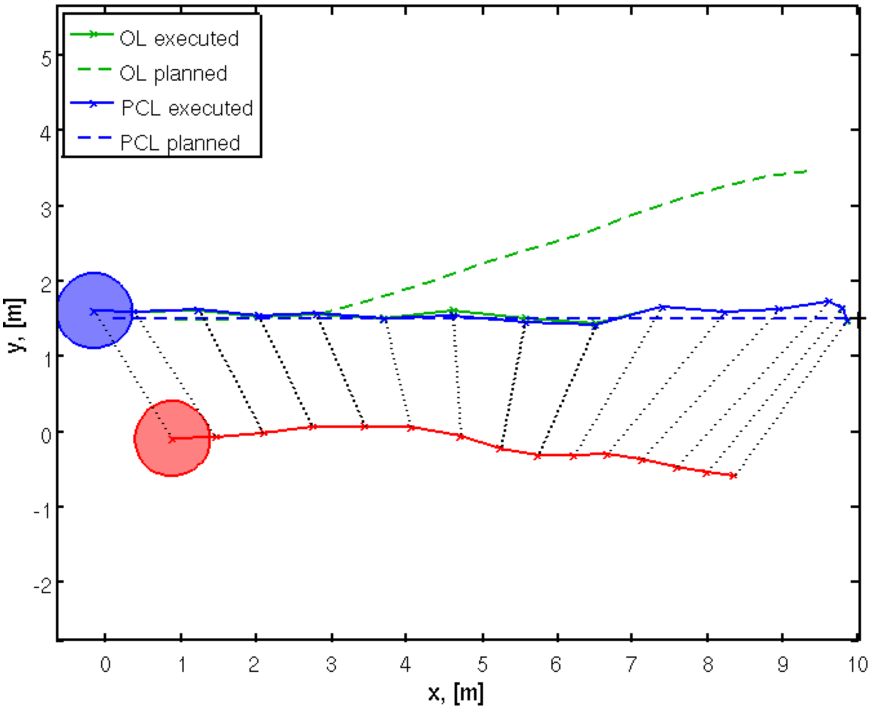


Figure 4.13: The planned (dashed) and executed (solid) paths for the OLRHC (green) and PCLRHC (blue) approaches and the predicted agent trajectory (red). The executed trajectories are very similar. The executed and planned paths for the PCLRHC approach are much closer than for the OLRHC approach.

The *executed* trajectories (solid) and the initial planned or predicted trajectories (dashed) are plotted in Figure 4.13. The time-correlated locations of the robot and agent (for the executed trajectories) are indicated by the dotted lines. Again, the two approaches produce similar executed trajectories, the initially planned and subsequently executed trajectories for the OLRHC approach differ greatly. The PCLRHC solution is much closer to the executed trajectory, and again, the PCLRHC approach deviates less from the preferred plan during execution.

4.3 Summary

In this chapter, the PCLRHC approach is applied to dynamic scenarios. The focus is on standard, independent models for the objects (robot and agents). The independence of the models result in the factorization of the joint posterior distribution of the object variables, allowing for the estimation process for each object to be implemented separately. First, the problem of systems with unknown states are considered. The estimation processes for various system classes are presented, and the chance constraints are formulated.

Simulation results are presented for linear systems. First, a scenario with a single dynamic agent is considered and the PCLRHC approach is able to obtain less conservative solutions than the OLRHC approach. The executed trajectories for the OLRHC and PCLRHC approaches are very similar due to the outerloop feedback mechanism, motivating the scenario with two oncoming dynamic agents. The planned and executed trajectories from the PCLRHC approach are significantly shorter than the cautious planned and executed trajectories from the OLRHC approach. A Monte-Carlo simulation is used to establish the practical benefit of the PCLRHC approach over the OLRHC approach by considering a randomized scenario where two agents move across the path of the robot. The mean executed path length for the PCLRHC approach is shorter. The PCLRHC approach obtains shorter executed paths in 72% of the cases, with a significant improvement (>20%) in 17.5% of the cases.

Next, joint parameter and state estimation is presented, focussing on continuous unknown parameters. Simulation results for a linear system with Gaussian noise, where the unknown Gaussian parameters enter the dynamics additively, are presented.

Chapter 5

Systems with Complex, Independent Robot-Agent Models

In the previous chapter, the PCLRHC approach is applied to motion planning in dynamic, uncertain environments. The advantage of this algorithm over the OLRHC approach is in bounding the uncertainty growth of the propagated system by accounting for future information. This allows for a less conservative solution from the PCLRHC approach that is much closer to the executed trajectory than is accomplished by the OLRHC approach.

In this chapter, agents with more complicated behaviors are considered. The models are still independent of the robot state. The complicated behaviors result in more uncertainty about the evolution of the system. The models considered in this chapter include systems with muliple possible destinations, and systems with multiple possible models.

Systems with multiple destinations can be posed as a system with discrete unknown parameters: the parameters that model the agent's goal state can assume one of a finite number of discrete values. The resulting state probability distribution is multimodal: for the system assumed in this work a sum-of-Gaussians distribution is obtained. The estimation process and chance constraints for this distribution is considered. Additionally, it is necessary to estimate the probability of that specific destination is the true destination of the agent.

Systems with multiple model classes are of interest since complex behaviors can be obtained from simpler behavioral models. Again, the resulting distribution is multimodal. Finally, it is necessary to estimate relative plausibility of the different models and to use this estimate during the planning process.

5.1 Systems with Unknown States and Unknown, Discrete Parameters

Consider a system of the form:

$$x_i = A(\theta)x_{i-1} + Bu_{i-1} + F\omega_{i-1} + f_\theta(\theta) \tag{5.1}$$

$$y_i = C(\theta)x_i + H_\nu \nu_i + h_\theta(\theta) \tag{5.2}$$

where $\theta \in \left[\begin{array}{cccc} \theta^{(1)} & \theta^{(2)} & \dots & \theta^{(J)} \end{array} \right]$ is a parameter vector that can assume one of J possible values. The disturbances are distributed according to $\omega_{i-1} \sim N(0, W)$ and the measurement noise is distributed according to $\nu_i \sim N(0, V)$.

The dynamic programming problem for a linear system with Gaussian noise, quadratic cost, and uncertain, discrete parameters has been attempted [36, 37]. This approach attempts to obtain an optimal feedback law: given any measurement and control combination, the optimal control law is defined. The problem is solved iteratively: the weights are fixed and the control law for the resulting multimodal distribution is chosen, then the weights are updated according to the new control law. This approach has the same drawback as before in that hard constraints cannot be incorporated and the cost function is limited to quadratic functions, which make imposing constraints as cost difficult.

In this section, the agents have multiple possible destinations, represented as discrete uncertain parameters and resulting in a multimodal (sum-of-Gaussians) distribution for the states. The estimation process has two parts: updating the weights (probability that a certain value of the parameter is the true value) and updating the state estimate conditioned on a specific value for the unknown parameter. In the PCLRHC approach (see Section 3.4.3), the most likely future measurements are assumed. There is some flexibility in implementing this algorithm through the choice of 'most likely' measurement.

5.1.1 Estimation: Sum-of-Gaussians Distributions

The objective is to recursively estimate the distribution of the system state, given the relative plausibility of each possible value of the discrete parameters and the probability distribution of the state at the previous cycle (i.e., $p(x_{k-1}|y_{1:k-1}, u_{1:k-2}, \theta^{(j)})$ and $P(\theta^{(j)}|y_{1:k-1}, u_{0:k-2})$ are given for all $j = 1, \dots, J$). The posterior distribution is of interest:

$$p(x_k|y_{1:k}, u_{0:k-1}, \{\theta^{(j)}\})$$

where $\{\theta^{(j)}\}$ is the finite set of possible values of θ. Using the law of total probability, this distribution can be written as:

$$p(x_k|y_{1:k}, u_{0:k-1}, \{\theta^{(j)}\}) = \sum_{j=1}^{J} p(x_k|y_{1:k}, u_{0:k-1}, \theta^{(j)}) P(\theta^{(j)}|y_{1:k}, u_{0:k-1}). \tag{5.3}$$

The terms in the summation can be calculated separately. The probability of the j^{th} instantiation of the parameter vector is independent of the system state and can be viewed as a time-dependent weight:

$$w_k^{(j)} \triangleq P(\theta^{(j)}|y_{1:k}, u_{0:k-1})$$

where $\sum_{j=1}^{J} w_k^{(j)} = 1$. This weight is recursively updated, using Bayes' Law:

$$P(\theta^{(j)}|y_{1:k}, u_{0:k-1}) = \frac{p(y_k|\theta^{(j)}, y_{1:k-1}, u_{0:k-1}) P(\theta^{(j)}|y_{1:k}, u_{0:k-1})}{p(y_k|y_{1:k-1}, u_{0:k-1})}.$$

Note that the measurement equation (Eq. (5.2)) is linear with white Gaussian noise when the parameter value is known (through conditioning) and $p(y_k|\theta^{(j)}, y_{1:k-1}, u_{0:k-1})$ is a normal distribution. Let

$$p(y_k|\theta^{(j)}, y_{1:k-1}, u_{0:k-1}) = N(\hat{y}_{k|k-1}^{(j)}, \Gamma_{k|k-1}^{(j)})$$

and note that

$$p(y_k|y_{1:k-1}, u_{0:k-1}) = \sum_{j=1}^{J} p(y_k|\theta^{(j)}, y_{1:k-1}, u_{0:k-1}) P(\theta^{(j)}|y_{1:k}, u_{0:k-1}),$$

then

$$w_k^{(j)} = \frac{N(\hat{y}_{k|k-1}^{(j)}, \Gamma_{k|k-1}^{(j)}) w_{k-1}^{(j)}}{\sum_{j=1}^{J} N(\hat{y}_{k|k-1}^{(j)}, \Gamma_{k|k-1}^{(j)}) w_{k-1}^{(j)}}.$$

The distribution of the state, conditioned on j^{th} value of the parameter in Eq. (5.3) has a normal distribution since the dynamic equation, Eq. (5.1), is linear with white Gaussian noise when the parameter value is given. Thus, the standard Kalman Filter of Section 2.4.3 can be used to estimate the distribution

$$p(x_k|y_{1:k-1}, u_{0:k-1}, \theta^{(j)}) = N(\hat{x}_{k|k}^{(j)}, \Sigma_{k|k}^{(j)}).$$

The estimated quantity then becomes:

$$p(x_k|y_{1:k}, u_{0:k-1}, \{\theta^{(j)}\}) = \sum_{j=1}^{J} w_k^{(j)} N(\hat{x}_{k|k}^{(j)}, \Sigma_{k|k}^{(j)}).$$

One filter is necessary for each of the possible values of the discrete variables. The resulting distribution is a weighted sum-of-Gaussians. In this work, the filters for all possible values of the discrete parameters are maintained, even when the likelihood of that parameter is small. These unlikely values are naturally ignored in the planning process with the use of the weights. Alternatively, hypothesis testing can be used to discard some of the parameter values and to discontinue the filter for those specific parameter values (see, for example, [15, 29]). This will result in a computation speed-up for the approach, especially in the case where the set of possible values for the discrete parameters is large. This extension will be investigated in future work.

5.1.2 Collision Chance Constraints: Sum-of-Gaussians Distribution

From Eq. (4.3), the probability of collision between two point objects at x_R and x_A, respectively, is given by:

$$P(C) \approx V_{\mathcal{B}} \times \int_{x^R} p(x^A = x^R | x^R) p(x^R) dx^R.$$

Let $x_R \sim N(x_R; \hat{x}_R, \Sigma_R)$ be the normally distributed state representing the position of a small disc robot, and $x_A \sim \sum_{j=1}^{J} w_k^{(j)} N(x_A^{(j)}; \hat{x}_A^{(j)}, \Sigma_A^{(j)})$ be the independent position state of a point obstacle whose uncertainty is distributed according to a sum-of-Gaussians with J components. The weights, $w_k^{(j)}$, are independent of the obstacle state. It is known that the product of independent Gaussian distributions is a weighted Gaussian distribution (see Appendix C.1) and the probability of collision for this case is:

$$P(C) \quad \approx \quad V_{\mathcal{B}} \times \sum_{j=1}^{J} w_k^{(j)} \frac{1}{\sqrt{\det\left(2\pi\Sigma_C^{(j)}\right)}} \exp\left[-\frac{1}{2}\left(\hat{x}_R - \hat{x}_A^{(j)}\right)^T \left(\Sigma_C^{(j)}\right)^{-1} \left(\hat{x}_R - \hat{x}_A^{(j)}\right)\right]$$

where $\Sigma_C^{(j)} \triangleq \Sigma_R + \Sigma_A^{(j)}$ is the combined position covariance for each component. The objective is to constrain the mean robot state $\hat{x}_R$ to guarantee that the collision chance constraint is satisfied. To account for the multiple weighted components of the agent distribution, each component is allocated a fraction of the total probability of collision, $\delta_j = \frac{1-\alpha}{J}$, so that $\sum_{j=1}^{J} \delta_j = 1 - \alpha$ and the overall collision constraint is satisfied. As a result, a constraint of the following form is obtained for each component:

$$P_j(C) \quad \approx \quad V_{\mathcal{B}} \times w_k^{(j)} \frac{1}{\sqrt{\det\left(2\pi\Sigma_C^{(j)}\right)}} \exp\left[-\frac{1}{2}\left(\hat{x}_R - \hat{x}_A^{(j)}\right)^T \left(\Sigma_C^{(j)}\right)^{-1} \left(\hat{x}_R - \hat{x}_A^{(j)}\right)\right].$$

Similar to Section 4.1.3.2, a lookup table can be constructed to obtain a constraint for each weighted component of the form:

$$\begin{aligned}(\hat{x}_R - \hat{x}_A)^T \left(\Sigma_C^{(j)}\right)^{-1} (\hat{x}_R - \hat{x}_A^{(j)}) &\geq -2\ln\left(\sqrt{\det\left(2\pi\Sigma_C^{(j)}\right)}\frac{\delta_j}{w_k^{(j)} \times V_{\mathcal{B}}}\right) \\ &\triangleq \kappa_j\end{aligned}$$

where κ is now a function of the effective level of certainty, δ_j, the size of the objects, the combined covariance, and each component weight, $w_k^{(j)}$. Disc objects of radius 0.5m are assumed with a total level of confidence of 0.99. It is further assumed that the agents will have two modes ($J = 2$), so that the lookup table is generated for $\delta_1 = \delta_2 = 0.005$. The constraining ellipse is parameterized by κ and was catalogued according to the smallest eigenvalue of the combined position covariance, $\lambda \triangleq \min(eig(\Sigma_C))$, in Table 5.1.

Table 5.1: Collision constraint parameter, κ, for different values of λ and weights with $\delta_j = 0.005$ and disc objects of radius 0.5 m

$\kappa(\lambda, w_k^{(j)})$	λ	0.005	0.01	0.05	0.1	0.5	1	5	10
$w_k^{(j)}$									
0.005		40.0	10.0	0	0	0	0	0	0
0.01		115	56	11	5	0	0	0	0
0.03		155	78	20	11.5	2.7	0.8	0	0
0.05		157	88	23	13.5	4.1	2.2	0	0
0.07		160	89	25.5	14.5	5.2	3.0	0	0
0.09		160	98	26.5	16.0	5.8	3.6	0	0
0.1		160	98	27	16.5	6.0	3.9	0.04	0
0.15		175	98	29	18	7.2	4.8	0.9	0
0.2		175	104	29	19.5	7.9	5.4	1.6	0.08
0.25		185	104	32	20.5	8.4	6.0	2.1	0.55
0.3		190	104	32	20.5	9.0	6.5	2.4	0.9
0.4		195	104	33.5	21.5	9.7	7.2	3.0	1.5
0.5		195	104	33.5	23	10.3	7.7	3.5	2.0
0.6		195	104	35.5	24	10.8	8.1	3.9	2.45
0.7		195	115	35.5	24	11.1	8.5	4.3	2.8
0.8		195	115	35.5	24.5	11.4	8.8	4.6	3.05
0.9		195	115	35.5	25	11.7	9.0	4.8	3.2
1.0		195	105	35.5	25.5	11.9	9.2	5.0	3.35

5.1.3 PCLRHC Implementations

In the PCLRHC approach, it is assumed that future measurements will be taken. Multiple implementations are possible by assuming different 'most likely' measurements since the agent state distribution has multiple peaks. Two implementations are investigated here: (i) assume that the globally most likely measurement occurs and use this measurement to update the weights and the probability distributions for the states conditioned on the parameter values, and (ii) update the state distributions with the locally (for each peak) most likely measurement and do not update the weights since the effect of the robot's planned trajectory on the information about the discrete parameter values is not modeled. Alternatively, the effect of planned robot trajectory on the ability of the robot to distinguish the agent parameter values should be modeled, in which case the weights and the state estimates can be updated without introducing a heavy bias towards the current most probable parameter value. This was not investigated in this work, but is of interest in future investigations.

5.1.3.1 Globally Most Likely Measurement

Assume that the most probable measurement occurs (the highest peak in the multimodal agent state distribution) and use this measurement to update *both* the weights and the state estimates: $\tilde{y}_i = E[y_i|y_{1:k}, \tilde{y}_{k+1:i-1}, \{\theta^{(j)}\}]$. This strategy, denoted by the *globally most likely measurement* (GMLM), assumes that the currently dominating component of the sum-of-Gaussian distribution gives the true parameter value. All future measurements are biased to reflect this belief. The expected effect is that the weight (probability) of this parameter value will increase, and the weights for the other parameter values will be reduced. Thus, the propagated system will be heavily biased towards this current most probable parameter value.

5.1.3.2 Locally Most Likely Measurement

The probability of each parameter value is not updated since the effect of the robot's planned path on these probabilities are not modeled. For the conditional state distributions, the measurement that is most likely for each distinct value of the parameters, denoted the *locally most likely measurement* (LMLM), is used: $\tilde{y}_i^{(j)} = E[y_i|y_{1:k}, \tilde{y}_{k+1:i-1}^{(j)}, \theta^{(j)}]$. The probabilities of each possible parameter value are maintained at the current level, and the state estimators for each of the possible parameter values account for the future measurements.

5.1.4 Simulation Results

In this example, a random walk model (Appendix B.1.3) with $W^R = 0.01 \times I_2$ and a linear position measurement model (Appendix B.2.2) with $V^R = 0.01 \times I_2$ are assumed for the robot. The cost

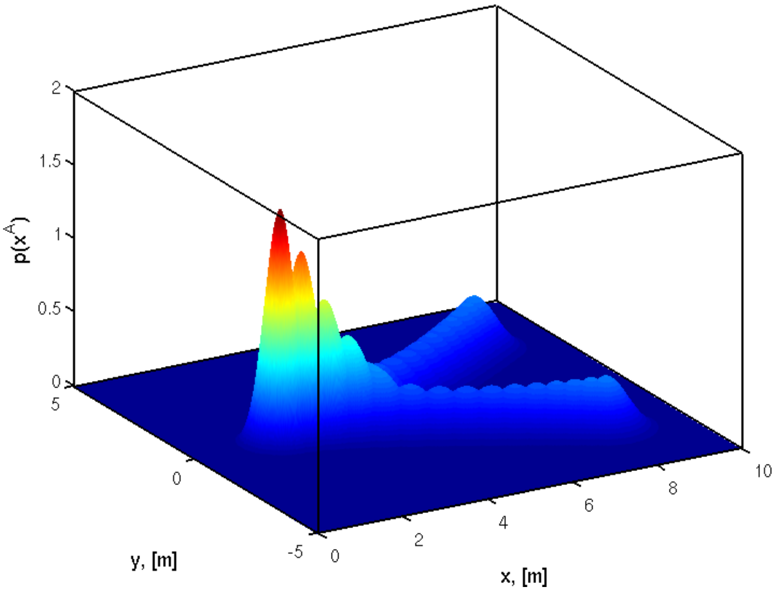


Figure 5.1: Predicted open-loop evolution of multimodal distribution. Two components are clearly discernable, and the spread of the components increases.

function and constraints of Section 4.1.4.1 are used. The goal state is $x_G = [8\ 1\ 0\ 0]$. The robot initial state is $x_0^R \sim \mathcal{N}(\hat{x}_{0|0}^R, \Sigma_{0|0}^A)$, where $\hat{x}_{0|0}^R = [0\ 1\ 1\ 0]^T$ and $\Sigma_{0|0}^R = 0.1 \times I_4$.

The agent is drawn to a destination point (Appendix B.1.4) with stiffnesses $k_x = k_x = 0.1$ and $W^A = 0.01 \times I_2$. Two possible destinations are assumed:

$$\theta^{(1)} = \begin{bmatrix} 10 \\ -4 \end{bmatrix} \quad \text{and} \quad \theta^{(2)} = \begin{bmatrix} 10 \\ 4 \end{bmatrix}.$$

The destination locations are perfectly known, but it is not known which destination the agent is traveling to. It is assumed that the agent does not change destinations during the execution of the plan. A linear position measurement model (Appendix B.2.2) is assumed, with $V^A = 0.01 \times I_2$. The agent initial state is $x_0^A \sim \mathcal{N}(\hat{x}_{0|0}^A, \Sigma_{0|0}^A)$, where $\hat{x}_{0|0}^A = [2\ 0\ 1\ 0]^T$ and $\Sigma_{0|0}^A = 0.1 \times I_4$. These destinations are approximately equally likely when the algorithm is initialized: $P(\theta^{(1)}|\eta_0) = 0.501$ and $P(\theta^{(2)}|\eta_0) = 0.499$. The second destination is the true destination.

In the OLRHC approach, all future measurements are ignored and the uncertainty of the robot and the agent grows unbounded. The predicted open-loop evolution of the multimodal distribution for the agent state is shown in Figure 5.1. The multimodal probability distributions at different stages are overlaid. The probability of each destination stays constant since no new information is available. The components of the distribution for the two possible destinations are clearly discernable. The spread of the distributions increase due to the open-loop prediction of the future states. The initial planned trajectory for the robot using the OLRHC (green) approach and the

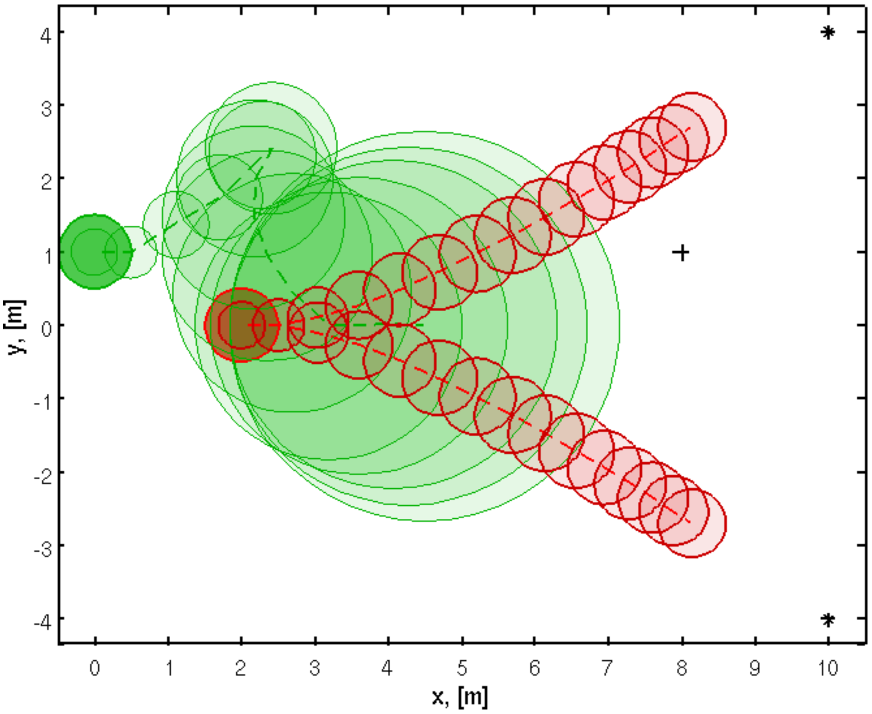


Figure 5.2: Initial planned trajectory with the OLRHC (green) approach and the agent predicted trajectories (red) towards the two possible destinations are plotted with uncertainty ellipses

predicted trajectory for the agent (red) with uncertainty ellipses are shown in Figure 5.2. For the agent, the two trajectories (one associated with each destination) is plotted, and the thickness of the line indicates the probability of that destination (thicker ⇒ higher probability). The uncertainty growth causes the robot to move away from both possible trajectories for the agent when the chance constraints are imposed, resulting in a very conservative plan.

For the PCLRHC with GMLM approach, a single anticipated measurement for the agent at each stage is used. The predicted evolution of the multimodal distribution for the agent state is shown in Figure 5.3. Again, the multimodal probability distributions at different stages are overlaid. First, note that the spread of the distributions are smaller since the future measurements are taken into account. The predicted evolution of the weights (probability of each destination being the true destination) is given in Figure 5.4. The weight associated with destination 1 is plotted in blue. These globally most probable measurements reinforce the belief that the currently most probable destination is the true destination, and the weight goes to unity. The second component of the multimodal distribution disappears. The solution is biased towards the currently most likely destination, and the other possibilities are abandoned. However, the effect of the robot plan on the disambiguation of the destination of the agent is not modelled, and this bias is unjustified. The obtained plan for the PCLRHC with GMLM (blue) approach and the predicted trajectory for the agent (red) with uncertainty ellipses are shown in Figure 5.5. The predicted trajectory associated

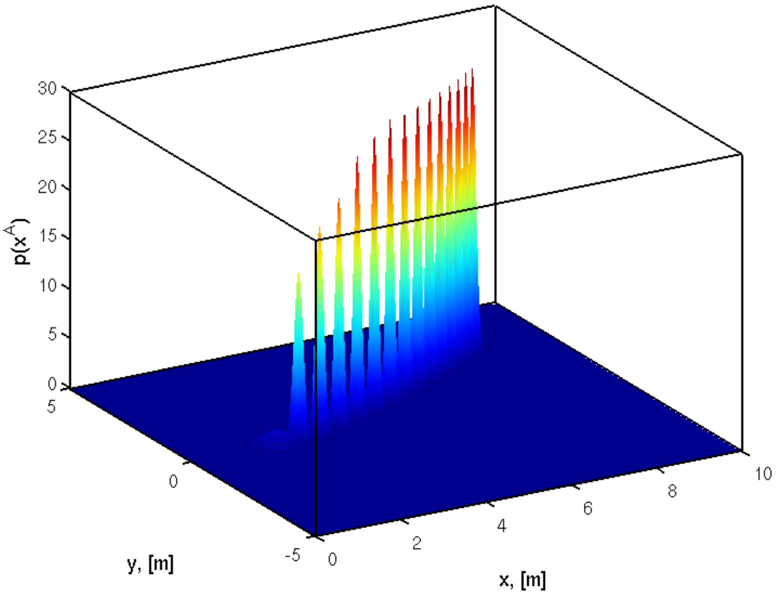


Figure 5.3: Predicted closed loop evolution of multimodal distribution with GMLM. The component of the multimodal distribution associated with the currently less likely destination disappears.

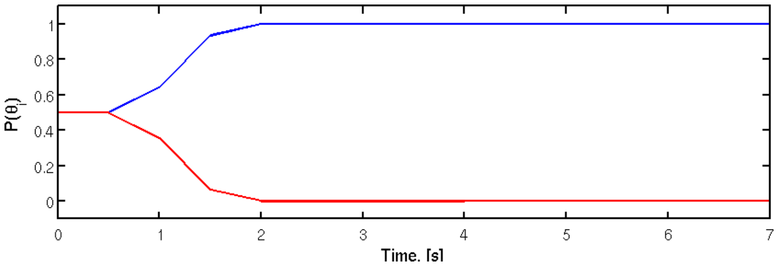


Figure 5.4: Predicted weights with GMLM. The weight associated with the true destination (red) is wrongfully predicted to decrease, resulting in an undesirable bias towards the currently most probable destination.

with the less likely destination is artificially pulled towards the more likely destination, which is an unwanted effect.

Lastly, consider the PCLRHC with LMLM approach. The weights are not updated since the effect of the robot motion on the estimation of the true destination of the agent is not modeled. The predicted evolution of the multimodal distribution for the agent state is shown in Figure 5.6. Both components of the multimodal distribution are maintained and the spread of the distributions are smaller since the future measurements are taken into account. The obtained plan for the PCLRHC with LMLM approach is given in Figure 5.7 using the same color scheme as in the previous paragraph. The planned trajectory moves the robot towards the goal while avoiding the agent. The predicted agent state evolution still accounts for both possible destinations, which is the desired behavior. It is thus concluded that the LMLM implementation is more appropriate than the GMLM implementation for the current application.

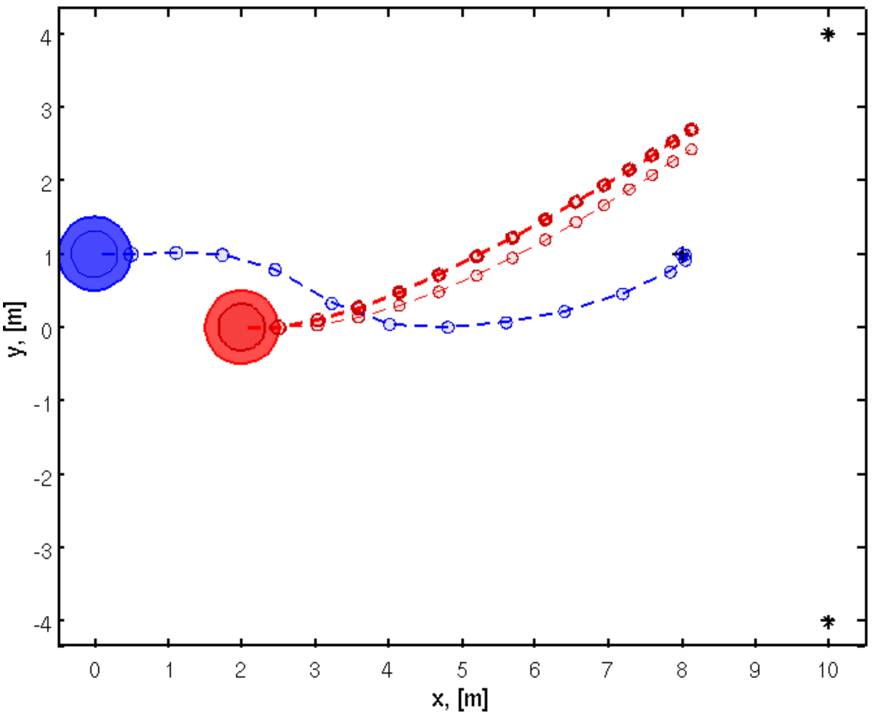


Figure 5.5: The initially planned trajectory with PCLRHC with GMLM approach (blue) and the predicted agent trajectories (red). The predicted trajectory associated with the less likely destination is artificially pulled towards the more likely destination, which is an unwanted effect.

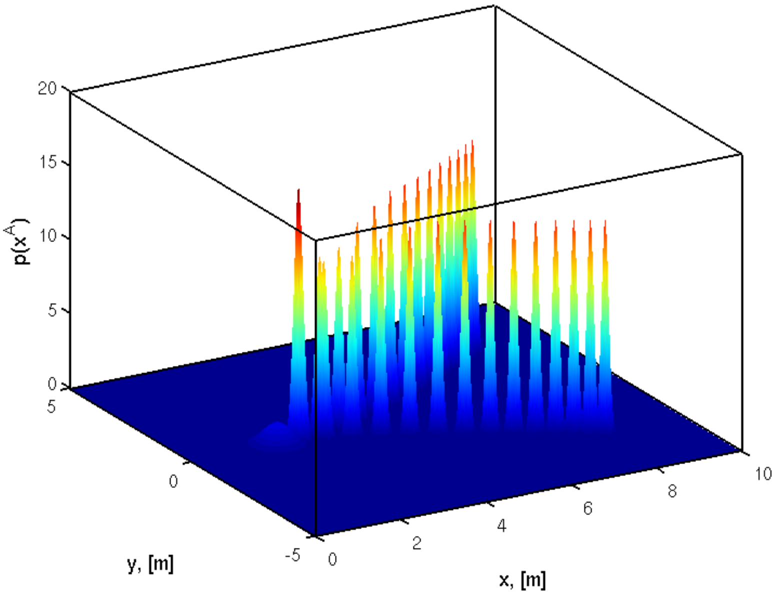


Figure 5.6: Predicted evolution of multimodal distribution with PCL approach with LMLM. Both components of the multimodal distribution associated are maintained.

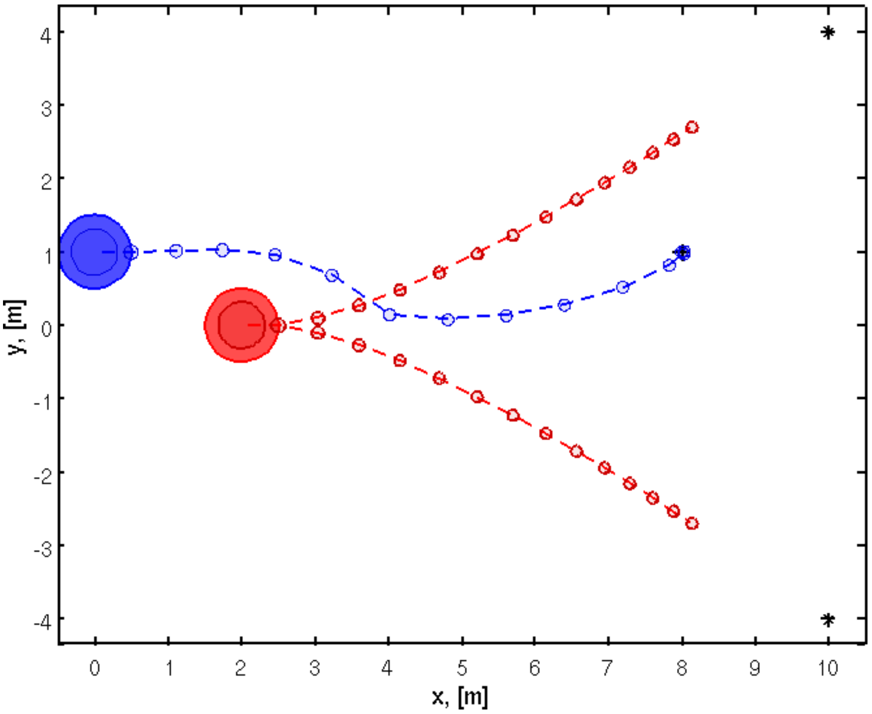


Figure 5.7: The initially planned trajectory with PCLRHC approach with LMLM (blue) and the predicted agent trajectories (red)

The sequence of *executed* trajectories (solid) and the planned (dashed) are plotted in Figure 5.8 for the PCLRHC approach with LMLM (blue) and the OLRHC approach (green). Both destinations are accounted for in the planning process. The OLRHC approach is practically unable to plan with both destinations due to the uncertainty growth. As the actual destination is disambiguated, the agent trajectory due to the false destination is ignored in the plan and a more reasonable trajectory is obtained. The robot cannot stay at the goal location. The PCLRHC approach is able to avoid the agent for both destinations and ignores the false destination once the true destination becomes known. The weights for the executed plan with the PCLRHC with LMLM approach is given in Figure 5.9, with the true destination indicated in red.

5.2 Systems with Multiple Models

Until now, all the systems evaluated consisted of a single parametric model, possibly with multiple parameter values. In this section, systems with multiple stochastic models are considered. To make this explicit, the stochastic system model is defined.

Definition 5.1: *Stochastic System Model*

Let $\mathcal{M}(\theta)$ be a predictive parametric model of the evolution of a system with parameters θ. The

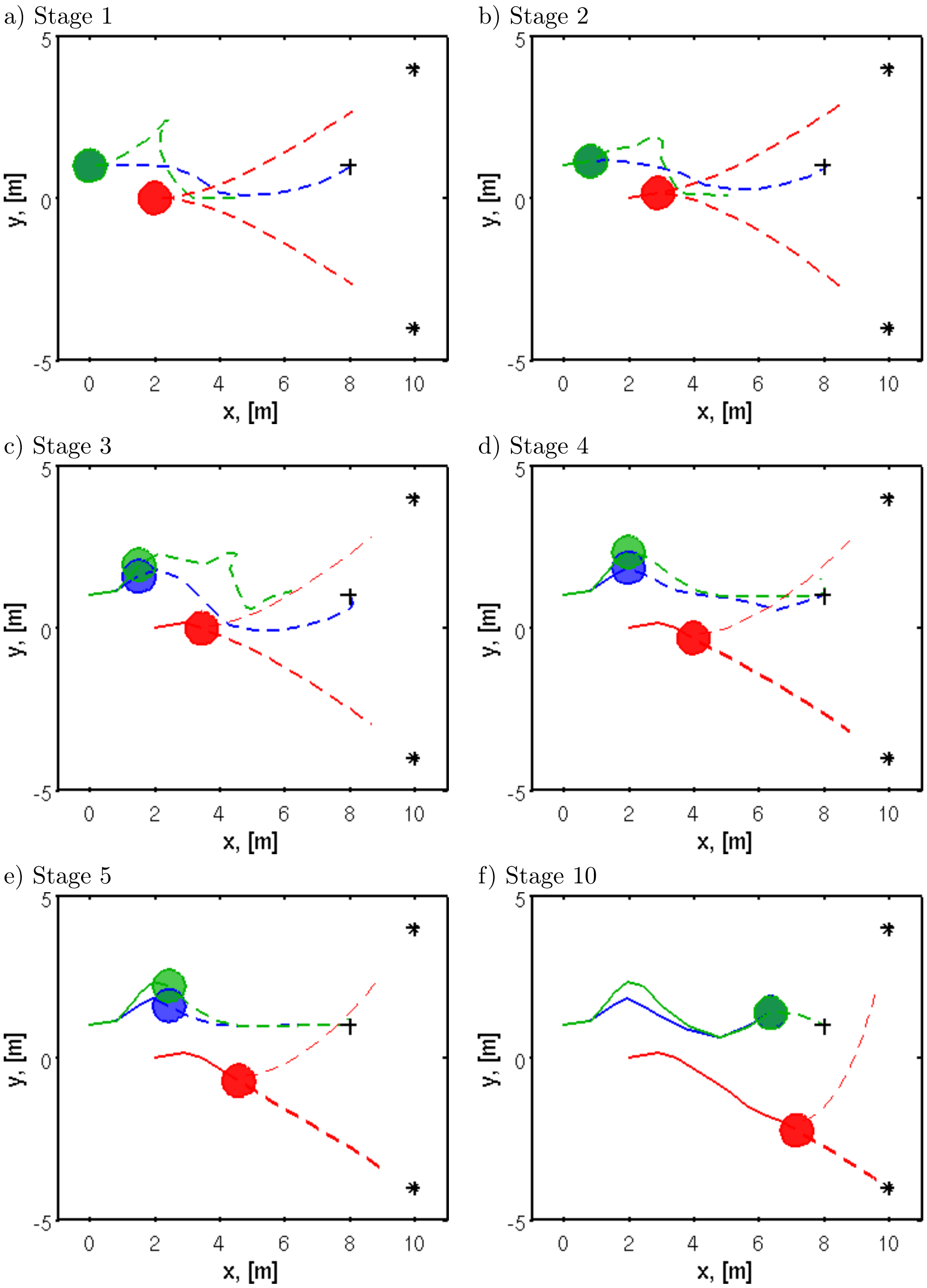


Figure 5.8: Executed (solid) and planned or predicted (dashed) trajectories at stages 1, 2, 3, 4, 5, and 10 using the OLRHC (green) and PCLRHC with LMLM (blue) approaches. The OLRHC approach is practically unable to handle the two agent behaviors (red, with thicker lines indicating more probable behaviors), whereas the robot moves between the agents with the PCLRHC with LMLM approach.

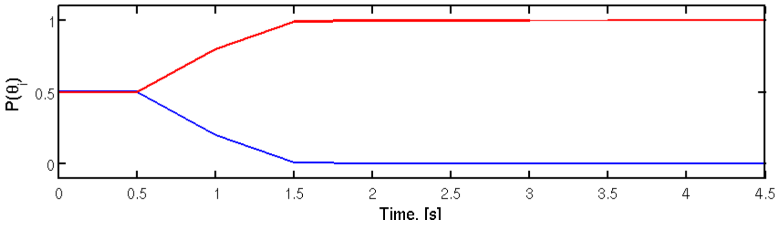


Figure 5.9: The actual evolution of weights with LMLM. As more information is gathered, the probability of the true destination (red) increases. Initially the destinations cannot be distinguished.

model defines the stochastic outcome of the system for a specific instantiation of the parameters, $\tilde{\theta}$:

$$p(x_k|y_{1:k}, u_{0:k-1}, \tilde{\theta}, \mathcal{M}).$$

The parameters can assume a value from a set of possible parameter values, Θ. The stochastic model is defined as the set of predictive models for these possible values of the parameters:

$$\{p(x_k|y_{1:k}, u_{0:k-1}, \tilde{\theta}, \mathcal{M}) : \tilde{\theta} \in \Theta\}.$$

To complete the definition of the stochastic model, it is necessary to specify the relative plausibility of each of the values of the parameters:

$$p(\theta|y_{1:k}, u_{0:k-1}, \mathcal{M})\ \forall \theta \in \Theta.$$

Suppose a set $\bigcup_{l=1}^{L} \mathcal{M}^{(l)}$ of candidate parametric models is proposed. For example, $M^{(1)}$ is a neutral agent model, $M^{(2)}$ is a friendly agent model, and $M^{(3)}$ is an adversarial agent model. The motion planning problem must be solved while accounting for the different possible models and it is desirable to select the most appropriate model as new information is obtained (stochastic model identification).

Assume the following dynamic and measurement equation for the l^{th} model:

$$x_i = A^{(l)}(\theta_l)x_{i-1} + B^{(l)}u_{i-1} + F^{(l)}\omega_{i-1}^{(l)} + f_\theta^{(l)}(\theta_l) \tag{5.4}$$

$$y_i = C^{(l)}(\theta_l)x_i + H^{(l)}\nu_i + h_\theta^{(l)}(\theta_l) \tag{5.5}$$

where $\theta_l \in \left[\begin{array}{cccc} \theta_l^{(1)} & \theta_l^{(2)} & \ldots & \theta_l^{(J_l)} \end{array}\right]$ are the parameters that can assume one of J_l possible values. The disturbances are distributed according to $\omega_{i-1}^{(l)} \sim N(0, W^{(l)})$ and the measurement noise is distributed according to $\nu_i^{(l)} \sim N(0, V^{(l)})$.

5.2.1 Estimation: Multiple Stochastic Models

The objective is to recursively estimate the distribution of the state of the system, given all the possible models and the finite instantiations of the discrete parameters for each model. The initial relative plausibility of each of the models must be specified, as well as the relative plausibility of the parameters for each of the models, and the initial distribution of the states. Let θ_l be the parameter vector associated with model l, and $\theta_l^{(j)} \in \Theta_l$ be the j^{th} instantiation of those parameters. Then, $p(x_{k-1}|y_{1:k-1}, u_{1:k-2}, \theta_l^{(j)}, \mathcal{M}^{(l)})$ and $P(\theta_l^{(j)}|y_{1:k-1}, u_{0:k-2}, \mathcal{M}^{(l)})$ are given for all $j = 1, \ldots, J_l$ and for each model, $\mathcal{M}^{(l)}$, and $P(\mathcal{M}^{(l)}|y_{1:k-1}, u_{0:k-2})$ is given $\forall l = 1, \ldots, L$.

The posterior distribution of interest is:

$$p(x_k|y_{1:k}, u_{0:k-1}, \{\mathcal{M}^{(l)}\}).$$

Using the law of total probability, this can be written as:

$$p(x_k|y_{1:k}, u_{0:k-1}, \{\mathcal{M}^{(l)}\}) = \sum_{l=1}^{L} p(x_k|y_{1:k}, u_{0:k-1}, \mathcal{M}^{(l)}) P(\mathcal{M}^{(l)}|y_{1:k}, u_{0:k-1}). \tag{5.6}$$

The terms in the summation of Eq. (5.6) can be calculated separately. The probability of the l^{th} model is assumed to be independent of the state of the system, and can be thought of as a time-dependent weight:

$$v_k^{(l)} \triangleq P(\mathcal{M}^{(l)}|y_{1:k}, u_{0:k-1})$$

where $\sum_{l=1}^{L} v_k^{(l)} = 1$. This weight is recursively updated, using Bayes' Law:

$$v_k^{(l)} = P(\mathcal{M}^{(l)}|y_{1:k}, u_{0:k-1}) = \frac{p(y_k|y_{1:k-1}, u_{0:k-1}, \mathcal{M}^{(l)}) P(\mathcal{M}^{(l)}|y_{1:k-1}, u_{0:k-1})}{p(y_k|y_{1:k-1}, u_{0:k-1})}. \tag{5.7}$$

The second term in the numerator of Eq. (5.7) is the weight at the previous stage, $v_{k-1}^{(l)}$. The first term in the numerator of Eq. (5.7) is calculated using the law of total probability:

$$p(y_k|y_{1:k-1}, u_{0:k-1}, \mathcal{M}^{(l)}) = \sum_{j=1}^{J_l} p(y_k|y_{1:k-1}, u_{0:k-1}, \theta_l^{(j)}, \mathcal{M}^{(l)}) P(\theta_l^{(j)}|y_{1:k-1}, u_{0:k-1}, \mathcal{M}^{(l)}). \tag{5.8}$$

The first term in the sum of Eq. (5.8) is calculated from the measurement equation for model $\mathcal{M}^{(l)}$. The measurement equation (Eq. (5.5)) is linear with Gaussian noise when the parameter values are known (through conditioning) and $p(y_k|y_{1:k-1}, u_{0:k-1}, \theta_l^{(j)}, \mathcal{M}^{(l)})= N(\hat{y}_{k|k-1}^{(l,j)}, \Gamma_{k|k-1}^{(l,j)})$ is a normal distribution. The second term in the sum of Eq. (5.8) is the probability that the current parameter instantiation is the true value, and is independent of the state. It can be considered as a time-dependent weight, $w_k^{(l,j)} \triangleq P(\theta_l^{(j)}|y_{1:k}, u_{0:k-1}, \mathcal{M}^{(l)})$, where $\sum_{j=1}^{J_l} w_k^{(l,j)} = 1$. This weight is recursively updated, using Bayes' Law:

$$w_k^{(l,j)} = \frac{p(y_k|y_{1:k-1}, u_{0:k-1}, \theta_l^{(j)}, \mathcal{M}^{(l)})P(\theta_l^{(j)}|y_{1:k-1}, u_{0:k-1}, \mathcal{M}^{(l)})}{p(y_k|y_{1:k-1}, u_{0:k-1}, \mathcal{M}^{(l)})}. \tag{5.9}$$

The first term in the numerator of Eq. (5.9) was calculated above. The second term in the numerator is the weight of the parameter instantiation at the previous stage: $w_{k-1}^{(l,j)}$. The denominator of Eq. (5.9) is obtained using the law of total probability:

$$p(y_k|y_{1:k-1}, u_{0:k-1}, \mathcal{M}^{(l)}) = \sum_{j=1}^{J_l} p(y_k|y_{1:k-1}, u_{0:k-1}, \theta_l^{(j)}, \mathcal{M}^{(l)})P(\theta_l^{(j)}|y_{1:k}, u_{0:k-1}, \mathcal{M}^{(l)}),$$

so that Eq. (5.9) becomes

$$w_k^{(l,j)} = \frac{N(\hat{y}_{k|k-1}^{(l,j)}, \Gamma_{k|k-1}^{(l,j)})w_{k-1}^{(l,j)}}{\sum_{j=1}^{J_l} N(\hat{y}_{k|k-1}^{(l,j)}, \Gamma_{k|k-1}^{(l,j)})w_{k-1}^{(l,j)}} \tag{5.10}$$

and Eq. (5.8) becomes

$$p(y_k|y_{1:k-1}, u_{0:k-1}, \mathcal{M}^{(l)}) = \sum_{j=1}^{J_l} w_k^{(l,j)} N(\hat{y}_{k|k-1}^{(l,j)}, \Gamma_{k|k-1}^{(l,j)}).$$

The numerator of Eq. (5.7) is

$$p(y_k|y_{1:k-1}, u_{0:k-1}, \mathcal{M}^{(l)})P(\mathcal{M}^{(l)}|y_{1:k-1}, u_{0:k-1}) = v_{k-1}^{(l)} \sum_{j=1}^{J_l} w_k^{(l,j)} N(\hat{y}_{k|k-1}^{(l,j)}, \Gamma_{k|k-1}^{(l,j)})$$

and the denominator is given by

$$p(y_k|y_{1:k-1}, u_{0:k-1}) = \sum_{l=1}^{L} \left(v_{k-1}^{(l)} \sum_{j=1}^{J_l} \left(w_k^{(l,j)} N(\hat{y}_{k|k-1}^{(l,j)}, \Gamma_{k|k-1}^{(l,j)}) \right) \right).$$

The weight is given by:

$$v_k^{(l)} = \frac{v_{k-1}^{(l)} \sum_{j=1}^{J_l} w_k^{(l,j)} N(\hat{y}_{k|k-1}^{(l,j)}, \Gamma_{k|k-1}^{(l,j)})}{\sum_{l=1}^{L} \left(v_{k-1}^{(l)} \sum_{j=1}^{J_l} \left(w_k^{(l,j)} N(\hat{y}_{k|k-1}^{(l,j)}, \Gamma_{k|k-1}^{(l,j)}) \right) \right)}. \tag{5.11}$$

It remains to calculate the first term in the sum in Eq. (5.6):

$$p(x_k|y_{1:k}, u_{0:k-1}, \mathcal{M}^{(l)}) = \sum_{j=1}^{J_l} p(x_k|y_{1:k}, u_{0:k-1}, \theta_l^{(j)}, \mathcal{M}^{(l)})P(\theta_l^{(j)}|y_{1:k}, u_{0:k-1}, \mathcal{M}^{(l)}). \tag{5.12}$$

The distribution of the state for model l and conditioned on j^{th} value of the parameter in Eq. (5.12) has a normal distribution since the dynamic equation, Eq. (5.4), is linear with Gaussian noise when the parameter value is given (through conditioning). Thus, the standard Kalman Filter of Section 2.4.3 can be used to estimate the distribution:

$$p(x_k|y_{1:k-1}, u_{0:k-1}, \theta_l^{(j)}, \mathcal{M}^{(l)}) = N(\hat{x}_{k|k}^{(l,j)}, \Sigma_{k|k}^{(l,j)}).$$

The second term in the sum of Eq. (5.12) is the weight, $w_k^{(l,j)}$. The distribution of the state for model $\mathcal{M}^{(l)}$ is:

$$p(x_k|y_{1:k}, u_{0:k-1}, \mathcal{M}^{(l)}) = \sum_{j=1}^{J_l} w_k^{(l,j)} N(\hat{x}_{k|k}^{(l,j)}, \Sigma_{k|k}^{(l,j)}). \tag{5.13}$$

The state distribution, given the set of models, is a sum-of-Gaussians distribution, given by:

$$p(x_k|y_{1:k}, u_{0:k-1}, \{\mathcal{M}^{(l)}\}) = \sum_{l=1}^{L} v_k^{(l)} \left(\sum_{j=1}^{J_l} \left(w_k^{(l,j)} N(\hat{x}_{k|k}^{(l,j)}, \Sigma_{k|k}^{(l,j)}) \right) \right). \tag{5.14}$$

In the special case where the models contain no uncertain parameters, the above equations simplify to:

$$v_k^{(l)} = \frac{v_{k-1}^{(l)} N(\hat{y}_{k|k-1}^{(l)}, \Gamma_{k|k-1}^{(l)})}{\sum_{l=1}^{L} v_{k-1}^{(l)} N(\hat{y}_{k|k-1}^{(l)}, \Gamma_{k|k-1}^{(l)})} \tag{5.15}$$

and

$$p(x_k|y_{1:k}, u_{0:k-1}, \{\mathcal{M}^{(l)}\}) = \sum_{l=1}^{L} v_k^{(l)} N(\hat{x}_{k|k}^{(l)}, \Sigma_{k|k}^{(l)}). \tag{5.16}$$

5.2.2 Chance Constraints: Sums of Gaussians

The resulting distribution for the multiple models systems of the form of Eq. (5.4) and Eq. (5.5) is a sum-of-Gaussians distribution (Eq. (5.14)). The chance constraints for a sum-of-Gaussians distribution was presented in Section 5.1.2 and can directly be applied here.

5.2.3 PCLRHC Implementations

As discussed in Section 5.1.3, two implementations were investigated: (i) assume that the globally most likely measurement occurs to update the weights and state estimates, and (ii) update the state distributions with the locally most likely measurement and do not update the weights. It was shown that the implementation using the locally most likely measurement resulted in the desirable behavior, and is used exclusively in this section.

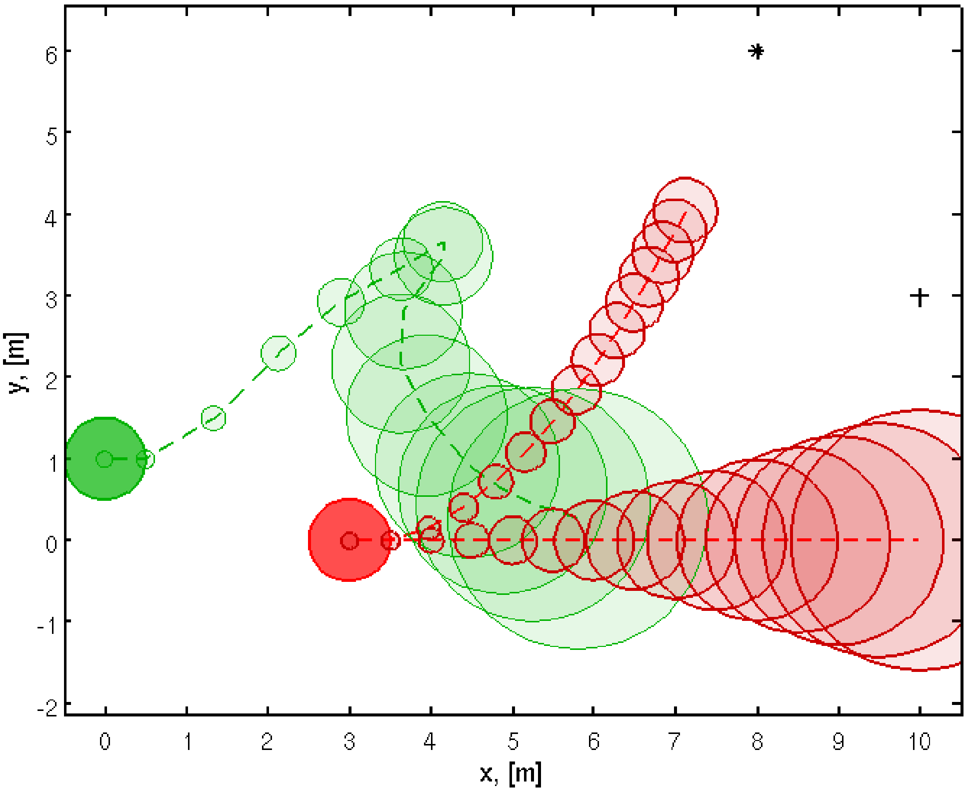


Figure 5.10: Initial planned trajectory with the OLRHC (green) approach and the agent predicted trajectories (red) with the two possible models are plotted with uncertainty ellipses.

5.2.4 Simulation Results

In this example, a random walk model (Appendix B.1.3) with $W^R = 0.01 \times I_2$ and a linear position measurement model (Appendix B.2.2) with $V^R = 0.01 \times I_2$ are assumed for the robot. The cost function and constraints of Section 4.1.4.1 are used. The goal state is $x_G = [10\ 3\ 0\ 0]$. The robot initial state is $x_0^R \sim \mathcal{N}(\hat{x}_{0|0}^R, \Sigma_{0|0}^A)$, where $\hat{x}_{0|0}^R = [0\ 1\ 1\ 0]^T$ and $\Sigma_{0|0}^R = 0.01 \times I_4$.

An agent with two possible dynamic models is assumed. The first model draws the agent to a single destination point (Appendix B.1.4) with stiffnesses $k_x = k_x = 0.1$, destination $t = [10\ 4]^T$, and $W^A = 0.01 \times I_2$. The second model is a random walk model (Appendix B.1.3) with $W^A = 0.01 \times I_2$. A linear position measurement model (Appendix B.2.2) is assumed, with $V^A = 0.01 \times I_2$. The models are perfectly known (the special case in Section 5.2.1). It is assumed that the agent does not change models during the execution of the plan. The agent initial state is $x_0^A \sim \mathcal{N}(\hat{x}_{0|0}^A, \Sigma_{0|0}^A)$, where $\hat{x}_{0|0}^A = [0\ 3\ 1\ 0]^T$ and $\Sigma_{0|0}^A = 0.01 \times I_4$. The relative plausibility of the models when the algorithm is initialized are $P(\mathcal{M}^{(1)}|\eta_0) = 0.501$ and $P(\mathcal{M}^{(2)}|\eta_0) = 0.499$. The true model for the agent is the random walk model.

The initial planned trajectory for the robot using the OLRHC (green) approach and the predicted trajectory for the agent (red) with uncertainty ellipses are shown in Figure 5.10. For the agent, the

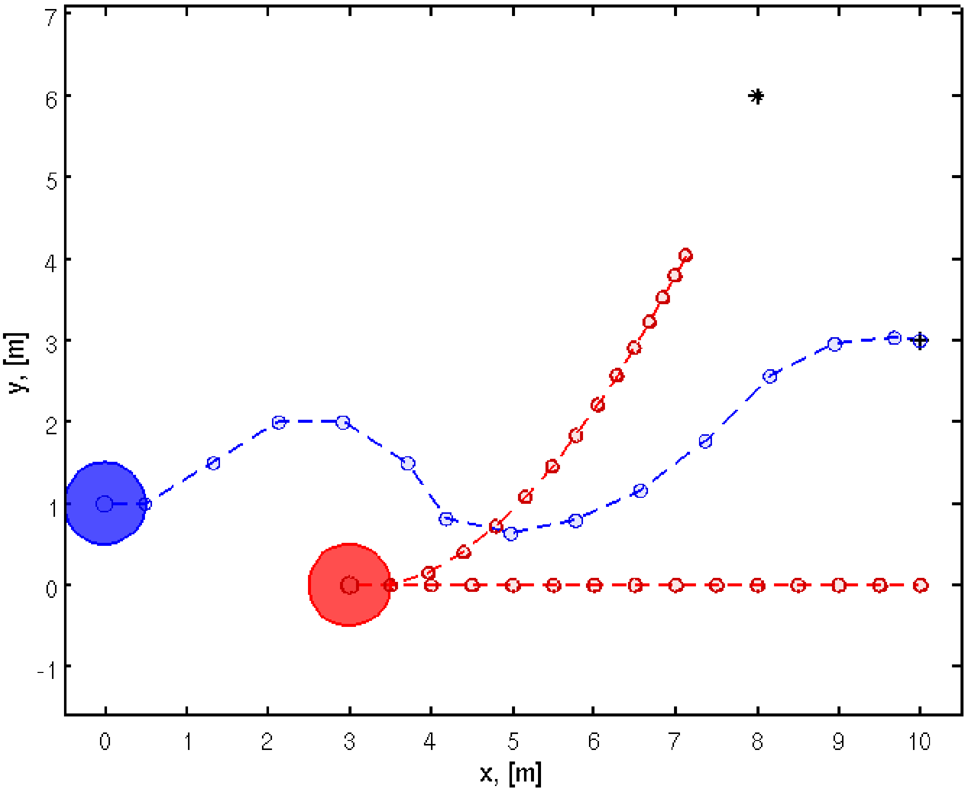
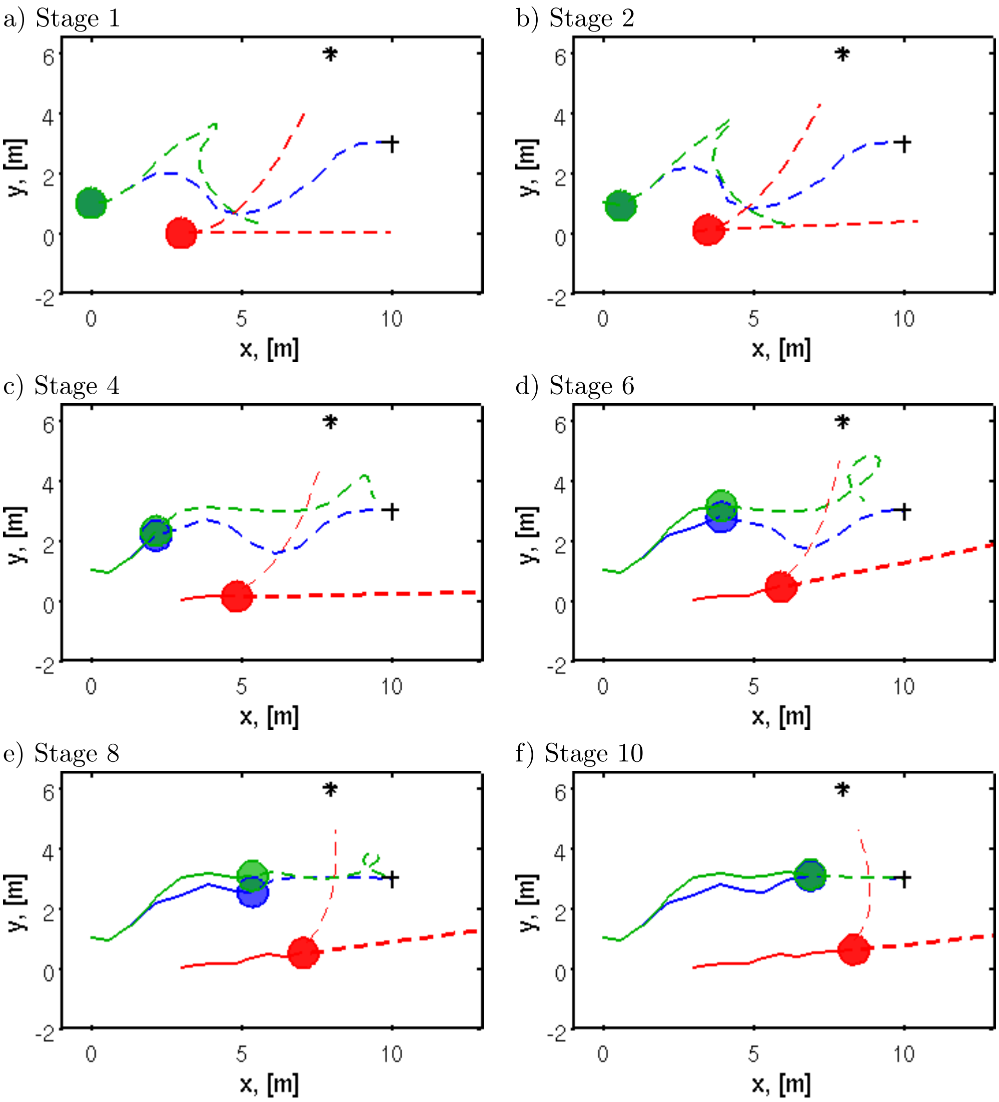


Figure 5.11: The initially planned trajectory with PCLRHC approach with LMLM (blue) and the predicted agent trajectories (red)

two trajectories (one associated with each model) are plotted, and the thickness of the line indicates the probability of that model (thicker ⇒ higher probability). The uncertainty growth causes the robot to move away from both possible trajectories for the agent when the chance constraints are imposed, resulting in a very conservative plan.

The obtained plan for the PCLRHC with LMLM (blue) approach and the predicted trajectory for the agent (red) with uncertainty ellipses are shown in Figure 5.11. The planned trajectory moves the robot towards the goal while avoiding the agent. The predicted agent state evolution still accounts for both possible destinations, which is the desired behavior.

The sequence of *executed* trajectories (solid) and the planned (dashed) are plotted in Figure 5.12 for the PCLRHC with LMLM approach (blue) and the OLRHC approach (green). Both models are accounted for in the planning process. The OLRHC approach is practically unable to plan with both models due to the uncertainty growth. As the actual model becomes known, the agent trajectory due to the false model is ignored in the plan, but the growth in uncertainty still affects the plan. The PCLRHC approach is able to avoid the agent for both models and cannot move directly towards the goal. The plan ignores the false destination once the true destination becomes known. The weights for the executed plan with the PCLRHC with LMLM approach is given in Figure 5.13, with the


a) Stage 1

b) Stage 2

c) Stage 4

d) Stage 6

e) Stage 8

f) Stage 10

Figure 5.12: Executed (solid) and planned or predicted (dashed) trajectories at stages 1, 2, 4, 6, 8, and 10 using the OLRHC (green) and PCLRHC with LMLM (blue) approaches. The OLRHC approach is practically unable to handle the two agent behaviors (red, with thicker lines indicating more probable behaviors).

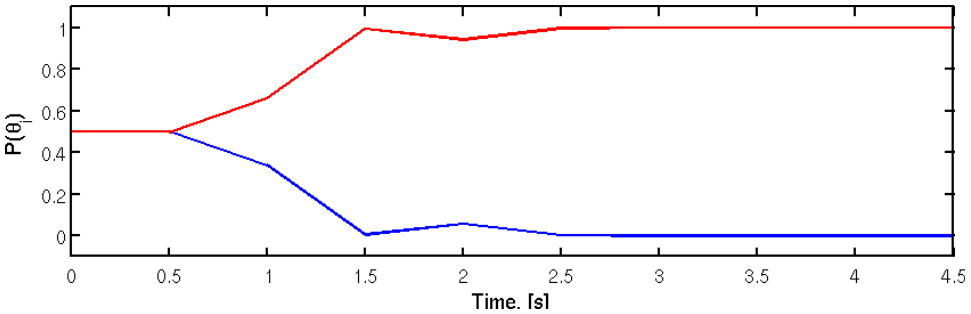


Figure 5.13: The actual evolution of weights with LMLM. As more information is gathered, the probability of the true model (red) increases. Initially the true agent model cannot be distinguished.

true destination indicated in red.

5.3 Summary

This chapter considered motion planning in the case where the agent models possess more complicated behaviors. The agent models are still independent of the robot state. The extensions presented in this chapter are motivated by agent models with multiple destinations and agents with multiple possible models. The former resulted in systems with uncertain discrete valued parameters in the models, and the estimation of the resulting distribution, a sum-of-Gaussians distribution, is presented. The collision chance constraints for this multimodal distribution is presented. An example of an agent with two possible destinations is presented to illustrate the benefit of the PCLRHC approach over the OLRHC approach.

Next, agents with multiple possible parametric models were considered. The estimation process for these systems was presented, resulting again in a sum-of-Gaussians distribution for the assumed dynamic and measurement equations. An example of an agent with two possible models is presented. Both models are accounted for in the motion planning problem. Again, the solution obtained from the PCLRHC approach is less conservative than the solution from the OLRHC approach.

Chapter 6

Systems with Dependent Robot-Agent Models

To date, models for the agents that are independent of the robot states have been used exclusively. In this chapter, the motion planning problem for systems with robot-state dependent agent models are considered. The following problems are considered: systems where the noise terms in the agent models are robot-state dependent, and systems where the agent dynamics is a function of the robot states and possibly of the other agent states. The former is motivated by the desire to model the quality of information obtainable about the agent behaviors: the robot can observe the agent better when closer to the agent. The latter problem is motivated by the need to model the interaction between the robot and the agents.

Let x_i^R, u_i^R, and ω_i^R be the robot state, control, and process noise, as defined in Section 2.1.1. The process noise is distributed as $p(\omega_i^R|x_i^R, u_i^R)$. Let y_i^R and ν_i^R be the robot measurement and measurement noise, as defined in Section 2.1.4. The measurement noise is distributed as $p(\nu_i^R|x_i^R)$. The robot dynamic and measurement equations are generally described by the nonlinear functions. Let n_A be the total number of dynamic obstacles (agents) in the vicinity of the robot. For the j^{th} agent: let x_i^{Aj}, u_i^{Aj}, and $\omega_i^{Aj} \sim p(\omega_i^{Aj}|x_i^{Aj}, u_i^{Aj})$ be the agent state, control, and process noise. Let y_i^{Aj} and $\nu_i^{Aj} \sim p(\nu_i^{Aj}|x_i^{Aj}, x_i^R)$ be the agent measurement and measurement noise. The agent dynamic and measurement equations are generally described by the nonlinear functions, with the model being an explicit function of the robot state. In order to handle the interdependence of the agent and robot models, the problem is formulated in the augmented state space of robot and agent dynamics. Define the augmented state of the system as

$$x_i = \left[\begin{array}{cccc} \left(x_i^R\right)^T & \left(x_i^{A1}\right)^T & \dots & \left(x_i^{An_A}\right)^T \end{array} \right]^T \tag{6.1}$$

and similarly for the the augmented controls, u_i, the augmented process noise, ω_i, the augmented measurements, y_i, and the augmented measurement noise, ν_i.

6.1 Agents with Robot-State Dependent Measurement Noise

Consider the robot model of the form:

$$\begin{aligned} x_i^R &= A_R x_{i-1}^R + B_R u_{i-1}^R + F_R \omega_{i-1}^R \\ y_i^R &= C_R x_i^R + H_R \nu_i^R \end{aligned}$$

with independent white Gaussian noise terms $\omega_{i-1}^R \sim N(0, W_R)$ and $\nu_i^R \sim N(0, V_R)$. This model is independent of the agent state.

The agent model considered here has a measurement model where the quality of the measurement of the agent is a function of the distance between the robot and the agent. Consider an agent dynamic model for the j^{th} agent of the form:

$$x_i^{Aj} = A_{Aj} x_{i-1}^{Aj} + B_{Aj} u_{i-1}^{Aj} + F_{Aj} \omega_{i-1}^{Aj}$$

where $\omega_{i-1}^{Aj} \sim N(0, W_{Aj})$. A general measurement model with state-dependent noise has the form:

$$y_i^A = C_A x_i^A + H_A \nu_i^A + \sum_{l=1}^{n_\xi} \Phi_l g(x_i^A, x_i^R) \xi_i^{(l)}$$

where $\nu_i^A \sim N(0, V_A)$, and $\xi_i^{(l)} \sim N(0, \sigma_\xi^2)$ $\forall l = 1, \ldots, n_\xi$. $g(x_i^A, x_i^R)$ is some nonlinear function of the robot and agent states. Φ_l is a matrix that aligns the l^{th} state dependent noise term with the appropriate component of the measurement. In this work, the system is restricted to the linear version of this equation:

$$y_i^A = C_A x_i^A + H_A \nu_i^A + \sum_{l=1}^{n_\xi} \overline{H}_A^{(l)} \xi_i^{(l)} + \sum_{l=1}^{n_\xi} G_A^{(l)} x_i^A \xi_i^{(l)} + \sum_{l=1}^{n_\xi} G_R^{(l)} x_i^R \xi_i^{(l)}.$$

This measurement model has multiplicative noise and is a function of both the robot and the agent state. This model is motivated by using the linearized distance-square function for $g(x_i^A, x_i^R)$ (see Appendix B.2.3).

6.1.1 Estimation: State-Dependent (Multiplicative) Noise

The probability distribution of interest during the estimation process for the current problem is the joint distribution of the robot and agent states, given the measurements:

$$p(x_i^R, x_i^{A1}, \ldots, x_i^{An_A} | y_{1:i}^R, y_{1:i}^{A1}, \ldots, y_{1:i}^{An_A}, u_{0:i-1}^R).$$

In the previous chapters, the assumption that the robot and agent models are all independent have the effect that this joint distribution can be written as the product of the state distribution for each object. In this chapter, this simplification is not possible, and the problem must be formulated in the augmented state space for the general case. Assume that the *augmented system* has the form:

$$\begin{aligned} x_i &= Ax_{i-1} + Bu_{i-1} + F\omega_{i-1} && (6.2)\\ y_i &= Cx_i + H\nu_i + \sum_{l=1}^{n_\xi} \overline{H}^{(l)}\xi_i^{(l)} + \sum_{l=1}^{n_\xi} G^{(l)}x_i\xi_i^{(l)} && (6.3) \end{aligned}$$

with independent white Gaussian noise terms. If it is assumed that x_{i-1} is normally distributed, then the multiplicative noise terms are the products of two Gaussian variables, which are not normally distributed. Thus, an optimal estimator is not generally available. As in Section 4.2.1.2, an approximate estimator can be derived by assuming that the estimator is of the form of a Luenberger estimator:

$$\hat{x}_{i|i} = \hat{x}_{i|i-1} + K_i(y_i - \hat{y}_{i|i-1}).$$

Then it can be shown (Appendix D.1) that the resulting estimator is of the form:
Prediction step:

$$\begin{aligned} \hat{x}_{i|i-1} &= A\hat{x}_{i-1|i-1} + Bu_{i-1} && (6.4)\\ \Sigma_{i|i-1} &= A\Sigma_{i-1|i-1}A^T + FWF^T. && (6.5) \end{aligned}$$

Measurement update step:

$$\begin{aligned} \hat{x}_{i|i} &= \hat{x}_{i|i-1} + K_i(y_i - C\hat{x}_{i|i-1}) && (6.6)\\ \Sigma_{i|i} &= (I - K_iC)\Sigma_{i|i-1} && (6.7) \end{aligned}$$

where

$$\begin{aligned} \Gamma_{i|i-1} &= C\Sigma_{i|i-1}C^T + HVH^T + \sum_{l=1}^{n_\xi} \sigma_\xi^2 G^{(l)}\left(\Sigma_{i|i-1} + \hat{x}_{i|i-1}\hat{x}_{i|i-1}^T\right)G^{(l)T} + \\ &\quad \sum_{l=1}^{n_\xi} \sigma_\xi^2 \overline{H}^{(l)}\overline{H}^{(l)T} + \sum_{l=1}^{n_\xi} \sigma_\xi^2\left(\overline{H}^{(l)}\hat{x}_{i|i-1}^T G^{(l)T} + G^{(l)}\hat{x}_{i|i-1}\overline{H}^{(l)T}\right) && (6.8)\\ K_i &= \Sigma_{i|i-1}C^T\Gamma_{i|i-1}^{-1}. && (6.9) \end{aligned}$$

Using this filter, the resulting distribution of the state is approximated with a Gaussian distribution.

6.1.2 Chance Constraints: Jointly Gaussian Distributions

As in Section 4.1.3, the collision chance constraint between a robot and obstacle must be imposed. From Eq. (4.3), the probability of collision between two point objects at x_R and x_A, respectively, is given by:

$$P(C) \approx V_{\mathcal{B}} \times \int_{x^R} p(x^A = x^R | x^R) p(x^R) dx^R.$$

In the previous sections, the agent state distribution was independent of the robot state. Now, the agent model is a function of the robot state. Assuming that the robot and agent states are jointly Gaussian, it can be shown (Appendix C.2) that:

$$\int_{x^R} p(x^A = x^R | x^R) p(x^R) dx^R \quad = \quad \frac{1}{\sqrt{\det(2\pi\Sigma_C)}} \exp\left[-\frac{1}{2} \left(\hat{x}^R - \hat{x}^A \right)^T \Sigma_C^{-1} \left(\hat{x}^R - \hat{x}^A \right) \right] . (6.10)$$

where $\Sigma_C \triangleq \Sigma_R + \Sigma_A - \Sigma_M - \Sigma_M^T$ is the combined position covariance. This analysis provides additional intuition about the appropriate combination of the covariances of the two distributions. Following the procedure of Section 4.1.3, Eq. 6.10 constrains the mean of the robot state to ensure that the collision chance constraint is enforced. The chance constraint becomes:

$$(\hat{x}^R - \hat{x}^A)^T \Sigma_C^{-1} (\hat{x}^R - \hat{x}^A) \quad \geq \quad \kappa$$

where κ is obtained from a lookup table (Table 4.1) as parameterized by the smallest eigenvalue of the combined covariance. Note that in this work the eigenvalues of the position covariances were similar (resulting in close to circular uncertainty ellipses), making the parameterization of κ by the smallest eigenvalue reasonable. In the case where the uncertainty ellipse is not circular, the use of the smaller eigenvalue ensures slightly conservative constraints. More accurate parameterization will be pursued in the future.

6.1.3 Simulation Results

In this example, a system with a simple dependency between the agent and the robot is considered. The robot model is independent of the agent states. The agent dynamics are independent of the robot state, but the measurement quality improves as the robot gets closer to the agent. The system contains a single dynamic agent.

The robot has a random walk dynamic model (see Appendix B.1.3) with $W^R = 0.01 \times I_2$ and a linear position measurement model (see Appendix B.2.2) with $V^R = 0.01 \times I_2$. The agent also has a random walk dynamic model (independent of the robot state) with $W^A = 0.01 \times I_2$. The linearized distance-dependent position measurement model (Appendix B.2.3) is used for the agent, with $V^R = 0.01 \times I_2$. Let $d_{max} = 5$, $n_\xi = 2$, $\xi_i^{(k)} \sim N(0,1)\ \forall k = 1, 2$.

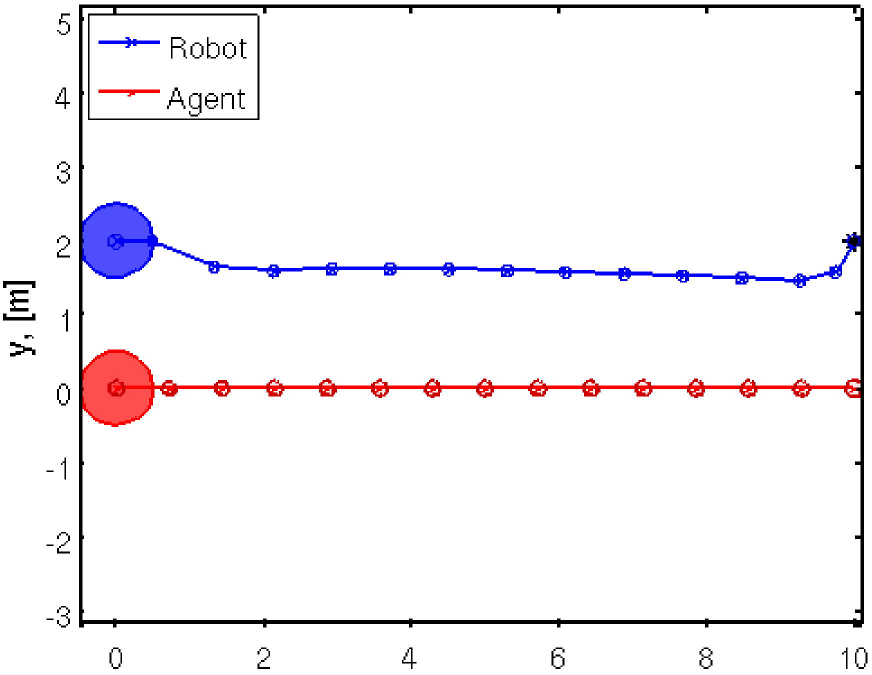


Figure 6.1: The planned trajectory for the robot (blue) and the predicted trajectory for the agent (red) with uncertainty ellipses. The agent has a robot-state dependent measurement noise model, allowing the robot to move towards the agent to improve the quality of information.

The goal state is $x_G = [10\ 2\ 0\ 0]^T$. The robot initial state is $x_0 \sim \mathcal{N}(\hat{x}_{0|0}, \Sigma_{0|0})$, where $\hat{x}_{0|0} = [0\ 2\ 1\ 0]^T$ and $\Sigma_{0|0} = 0.01 \times I_4$. The agent initial state is $x_0^A \sim \mathcal{N}(\hat{x}_{0|0}^A, \Sigma_{0|0}^A)$, where $\hat{x}_{0|0}^A = [1\ 0\ 1.2\ 0]^T$ and $\Sigma_{0|0} = 0.01 \times I_4$. A quadratic cost function is used, similar to Section 4.1.4.1. However, the robot can influence the quality of the agent measurement, and the future estimates of the agent state become a function of the planned trajectory of the robot up to that stage. This means that the estimation and the planning processes are not separable. Since the actions of the robot now have an effect on the estimation of the system, the covariance terms in the cost function for the imperfect state information problem cannot be ignored:

$$\begin{aligned} l_M^{ISI}(\zeta_M) &= Tr(Q_M \Sigma_{M|M}) + (\hat{x}_{M|M} - x_G)^T Q_M (\hat{x}_{M|M} - x_G) \\ l_i^{ISI}(\zeta_i, u_i) &= Tr(Q_i \Sigma_{i|i}) + (\hat{x}_{i|i} - x_G)^T Q_i (\hat{x}_{i|i} - x_G) + u_i^T R_i u_i \end{aligned}$$

where $Q_N = \text{diag}(10, 10, 0, 0, 100, 100, 0, 0)$, $Q_i = \text{diag}(1, 1, 0, 0, 100, 100, 0, 0)$, and $R_i = \text{diag}(0.1, 0.1, 0, 0)$, for all $i = 0, \ldots, N-1$. The same chance constraints as in Section 4.1.4.1 are imposed. Since the estimation for the agent state is dependent on the robot state through the distance-dependent noise model, the collision chance constraints of Section 6.1.2 must be used.

The planned trajectory for the example specified above is given in Figure 6.1 for the PCLRHC approach. The planned trajectory for the robot (blue) is plotted, along with the uncertainty ellipses

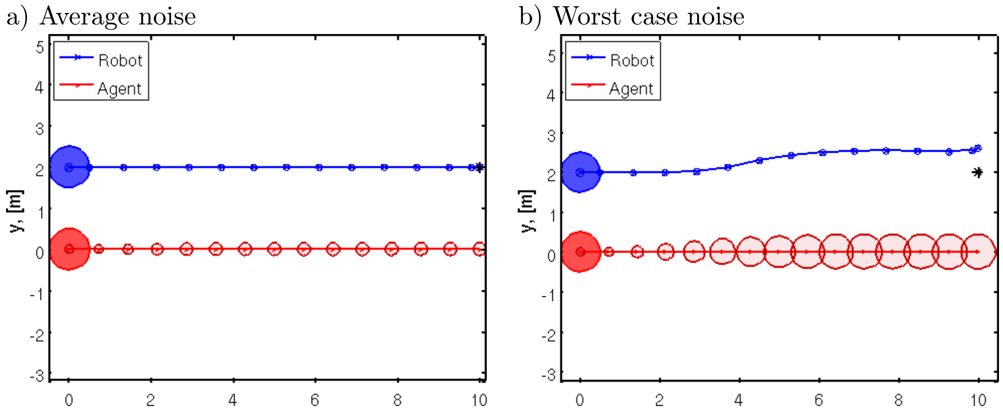


Figure 6.2: The planned trajectory for the robot (blue) and the predicted trajectory for the agent (red) with measurement noise terms that are independent of the robot state, but of comparable magnitude

for the position variables. The agent (red) position and predicted trajectory is plotted. Again, the position uncertainty along the predicted path is represented by the uncertainty ellipses. The position uncertainty is heavily penalized in the cost function specified above to accentuate the active learning component. The robot chooses actions to minimize this uncertainty, resulting in a solution where the robot chooses a longer path that moves towards the agent to get more accurate measurements. Thus, the robot adjusts its actions to get better information.

In order to compare this result to a system without state-dependent noise, an equivalent measurement noise model must be defined. Two options are compared: a measurement model with some average noise properties, and a measurement model with worst-case noise properties. Both are obtained by fixing the measurement noise at some level, so that the measurement noise becomes state-independent:

$$y_i \quad = \quad Cx_i + H_\nu \nu_i + \sum_{k=1}^{n_\xi} \Phi^{(k)} d^2 \xi_i^{(k)}.$$

For the average model, $d = 2.5$, and for the worst case model, $d = 5$. The planned trajectories for these cases are given in Figure 6.2.

The robot cannot influence the uncertainty evolution of the system since the noise is not state dependent, and it has to rely on passive learning from the PCL approach. Even with the PCLRHC approach, the measurement uncertainty for the agent is big enough for the worst-case model that the collision chance constraints push the robot away from the agent.

The uncertainty associated with the position of the agent (using the trace of the covariance matrix) is plotted for the three cases above in Figure 6.3. The uncertainty is kept to a minimum when the state dependent noise is incorporated into the model. When compared to the average noise

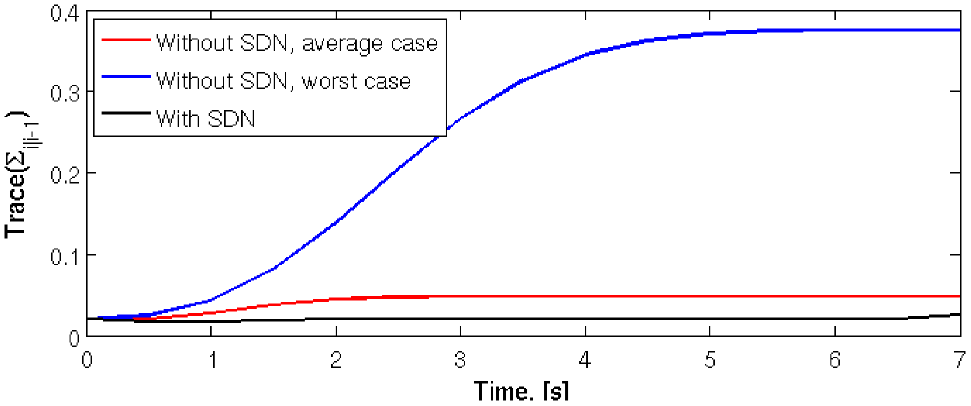


Figure 6.3: The evolution of the covariance of the system state with the different agent measurement noise models. The overall uncertainty is much reduced by using the state-dependent noise model.

model above, the uncertainty is lower, even though the distance threshold is higher for the system with the state dependent noise. The worst-case model above has the same threshold as the system with state dependent noise, but the robot is allowed to be more aggressive when the state dependent measurement model is used since the quality of the information is higher.

6.2 Interactive Models

Previous works on interactive robot-agent models is very limited. Kluge and Prassler [30] introduced the notion of reflective navigation where the agents are assumed to make decisions in a similar way to the robot. It is assumed that the agent will maximize some utility function, and requires knowledge of that utility function and the dynamic capabilities of each agent. The geometric and velocity uncertainty is accounted for by using probabilistic velocity obstacles for short-term prediction. Different levels of reflection are considered. In a similar manner, Van Den Berg et al. [55] introduce the reciprocal velocity obstacle to approximate the effect of agent deliberation. The key observation is that if all the movers take appropriate action to avoid collisions, then the correct behavior can be obtained. This approach is applied to a large number of movers, and a noted limitation is the ability to anticipate interactions with other agents and the effect on the robot-agent interaction. As a result, collisions can still occur due to agent occlusion. This approach does not account for the stochasticity of the system, and it is not clear how the method can be extended to obtain more intelligent agent behaviors.

In the current work, three classes of agent models are distinguished: friendly agents, neutral agents, and adversarial agents. The agent models used to date are neutral since the agent behavior is not adjusted for the robot actions. In this section, friendly and adversarial agents are considered. The friendly agent model brings the agent to a complete stop when the distance between the objects is below some threshold. For an adversarial model, the agent attempts to reach the robot. For both interactive agent models, knowledge of the robot state is assumed.

6.2.1 Friendly Agent

The agent comes to a stop when the distance between the agent and the robot is below some threshold, and uses a random walk model (Appendix B.1.3) otherwise. The agent is assumed to have a linear position measurement model (Appendix B.2.2).

6.2.1.1 Agent Model

The dynamic equation for the j^{th} agent is given by:

$$x_i^{Aj} = A_{Aj} x_{i-1}^{Aj} + B_{Aj} u_{i-1}^{Aj} + F_{Aj} \omega_{i-1}^{Aj}$$

where $\omega_{i-1}^{Aj} \sim N(0, W_{Aj})$ is white, Gaussian noise and

$$B_{Aj} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad F_{Aj} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

The stopping behavior is obtained by switching between different A_{Aj} according to the distance between the agent and the robot and the agent velocity components. Let x_i be the augmented state (as defined in Eq. (6.1)), then the distance between the robot and agent can be written as $x_i^T M x_i$, and the expected distance is $d = E\left(x_i^T M x_i | \eta_i\right) = \hat{x}_{i|i}^T M \hat{x}_{|ii} + Tr(M\Sigma_{i|i})$. Let v_x and v_y be the components of the agent velocity vector, then the dynamic model is given by Table 6.1 for the case where the distance between the robot and agent are below the threshold, δ. Let $\Delta v_{x,max}$, $\Delta v_{x,min}$, $\Delta v_{y,max}$, and $\Delta v_{y,min}$ define the maximum and minimum incremental changes in agent velocity for the velocity components. The following matrices are used:

$$A_{RW} = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad A_{S1} = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$A_{S2} = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad A_{S3} = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

Table 6.1: Friendly agent model when distance is below threshold

v_x	v_y	A_{Aj}	u
$v_x \geq \Delta v_{x,max}$	$v_y \leq \Delta v_{y,max}$	A_{RW}	$\begin{bmatrix} -\Delta v_{x,max} & -\Delta v_{y,max} \end{bmatrix}^T$
$v_x \geq \Delta v_{x,max}$	$\Delta v_{y,min} < v_y < \Delta v_{y,max}$	A_{S1}	$\begin{bmatrix} -\Delta v_{x,max} & 0 \end{bmatrix}^T$
$v_x \geq \Delta v_{x,max}$	$v_y \leq \Delta v_{y,min}$	A_{RW}	$\begin{bmatrix} -\Delta v_{x,max} & -\Delta v_{y,min} \end{bmatrix}^T$
$\Delta v_{x,min} < v_x < \Delta v_{x,max}$	$v_y \leq \Delta v_{y,max}$	A_{S2}	$\begin{bmatrix} 0 & -\Delta v_{y,max} \end{bmatrix}^T$
$\Delta v_{x,min} < v_x < \Delta v_{x,max}$	$\Delta v_{y,min} < v_y < \Delta v_{y,max}$	A_{S3}	$\begin{bmatrix} 0 & 0 \end{bmatrix}^T$
$\Delta v_{x,min} < v_x < \Delta v_{x,max}$	$v_y \leq \Delta v_{y,min}$	A_{S2}	$\begin{bmatrix} 0 & -\Delta v_{y,min} \end{bmatrix}^T$
$v_x \leq \Delta v_{x,min}$	$v_y \leq \Delta v_{y,max}$	A_{RW}	$\begin{bmatrix} -\Delta v_{x,min} & -\Delta v_{y,max} \end{bmatrix}^T$
$v_x \leq \Delta v_{x,min}$	$\Delta v_{y,min} < v_y < \Delta v_{y,max}$	A_{S1}	$\begin{bmatrix} -\Delta v_{x,min} & 0 \end{bmatrix}^T$
$v_x \leq \Delta v_{x,min}$	$v_y \leq \Delta v_{y,min}$	A_{RW}	$\begin{bmatrix} -\Delta v_{x,min} & -\Delta v_{y,min} \end{bmatrix}^T$

6.2.1.2 Simulation Results

A random walk model (Appendix B.1.3) with $W^R = 0.01 \times I_2$ and a linear position measurement model (Appendix B.2.2) with $V^R = 0.01 \times I_2$ are assumed for the robot. The friendly model presented in the previous section is assumed, with $W^A = 0.01 \times I_2$ and $V^A = 0.01 \times I_2$. The cost function of Section 6.1.3 is used. The goal state is $x_G = [10\ 0\ 0\ 0]$. The robot initial state is $x_0^R \sim \mathcal{N}(\hat{x}_{0|0}^R, \Sigma_{0|0}^A)$, where $\hat{x}_{0|0}^R = [0\ 0\ 1\ 0]^T$ and $\Sigma_{0|0}^R = 0.01 \times I_4$. The agent initial state is $x_0^A \sim \mathcal{N}(\hat{x}_{0|0}^A, \Sigma_{0|0}^A)$, where $\hat{x}_{0|0}^A = [5\ 5\ 0\ -1]^T$ and $\Sigma_{0|0}^A = 0.01 \times I_4$. Finally, the collision chance constraint is imposed at each stage with $\delta_c = 0.01$. Furthermore, $\delta = 5$ m, $\Delta v_{x,max} = \Delta v_{x,min} = \Delta v_{y,max} = \Delta v_{y,min} = 0.2$ m/s are assumed.

Two cases are considered: (i) the agent has a friendly model (Figures 6.4 to 6.6), and (ii) the agent has a neutral model (Figure 6.7). The initially planned trajectories for the friendly agent, obtained from the OLRHC (green) and PCLRHC (blue) approaches, are plotted in Figure 6.4 (with the uncertainty ellipses). The dynamic obstacle predicted trajectory is plotted in red (with the uncertainty ellipses for the open-loop case). The solution from the PCLRHC approach is less conservative than the solution from the OLRHC approach. The PCLRHC approach is able to move in front of the agent since the solution is less conservative. The sequence of executed plans and planned trajectories are given in Figures 6.5 and 6.6 with the PCLRHC and OLRHC approaches, respectively.

The initially planned trajectories for the neutral agent, obtained from the OLRHC (green) and PCLRHC (blue) approaches, are plotted in Figure 6.7 (with the uncertainty ellipses). The dynamic obstacle predicted trajectory is plotted in red (with the uncertainty ellipses for the open-loop case). For both the PCLRHC and OLRHC approaches, the obtained solution is to pass behind the agent. When comparing Figures 6.4 and 6.7, the effect of the agent cooperation is limited in the case of OLRHC due to the conservatism in the solution. The PCLRHC approach is able to make use of the anticipated cooperation to obtain a significantly shorter trajectory. It is easy to postulate problems

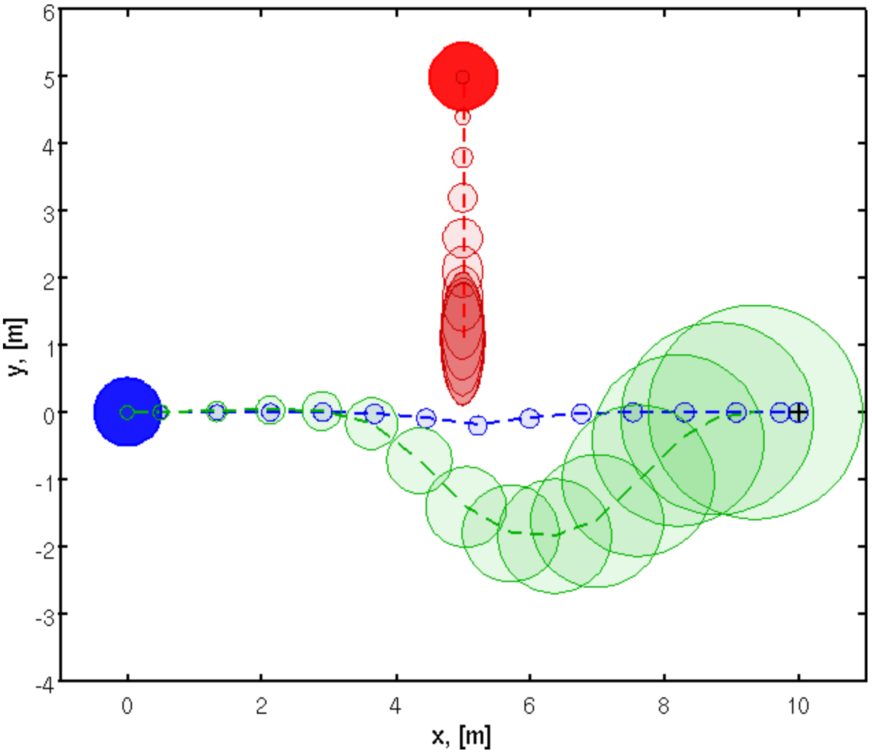


Figure 6.4: Planned trajectories for the robot and the *friendly* agent, using the OLRHC approach (green) and PCLRHC approach (blue), with 1-σ uncertainty ellipses along the planned trajectories. The robot and agent initial positions are indicated with the solid blue and solid red circles, respectively. The predicted agent trajectory, with the 1-σ uncertainty ellipses (OL case) is given in red.

where the robot has to rely on the cooperation of the agents in order to successfully reach the goal location (the *invisible robot* motion planning problem is intuitively harder than the *visible robot* motion planning problem). The reduction of conservatism with the PCLRHC approach is expected to assist in these situations as well.

A third case is of interest: the robot expects the agent to come to a stop, when in fact the agent does not react to the robot. The agent model that the robot assumes is wrong. This case is very hard to evaluate since the robot behavior and reaction to the misinformation is very dependent on the specific scenario, the models, and the dynamics of the system. Generally, this type of model uncertainty can be captured with the process noise term (unmodeled dynamics), and is not considered further in this work.

6.2.2 Adversarial Model: Attracted to Robot

The adversarial model is based on a critically damped spring-mass-damper system where the agent is drawn towards the robot. The agent attempts to occupy the same space as the robot, and the robot has to actively avoid the agent.

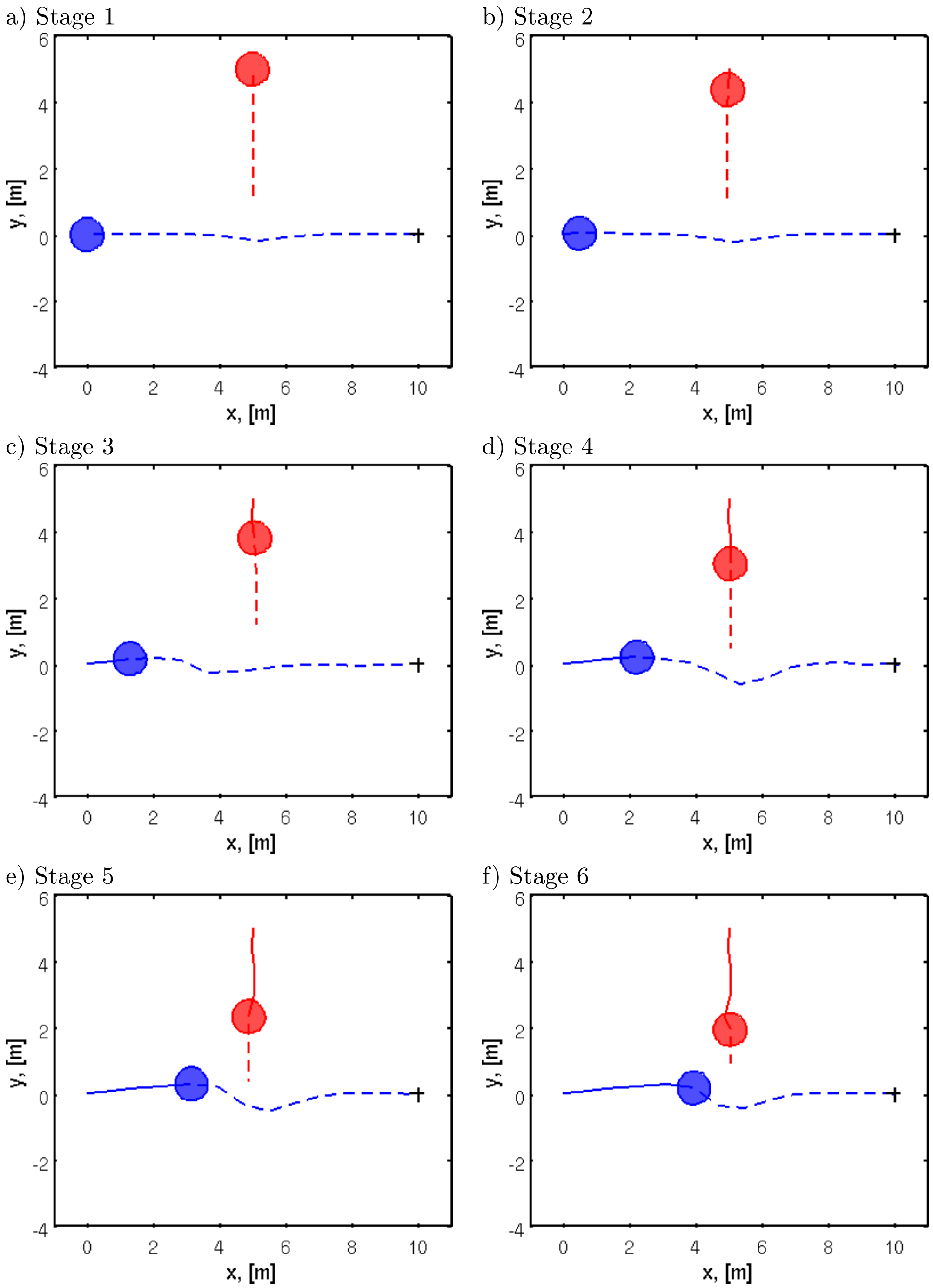


Figure 6.5: Executed (solid) and planned (dashed) trajectories at stages 1, 2, 3, 4, 5, and 6 with a friendly agent model using the PCLRHC approach. The robot (blue) is able to pass in front of the agent (red), resulting in a shorter executed path than the OL case.

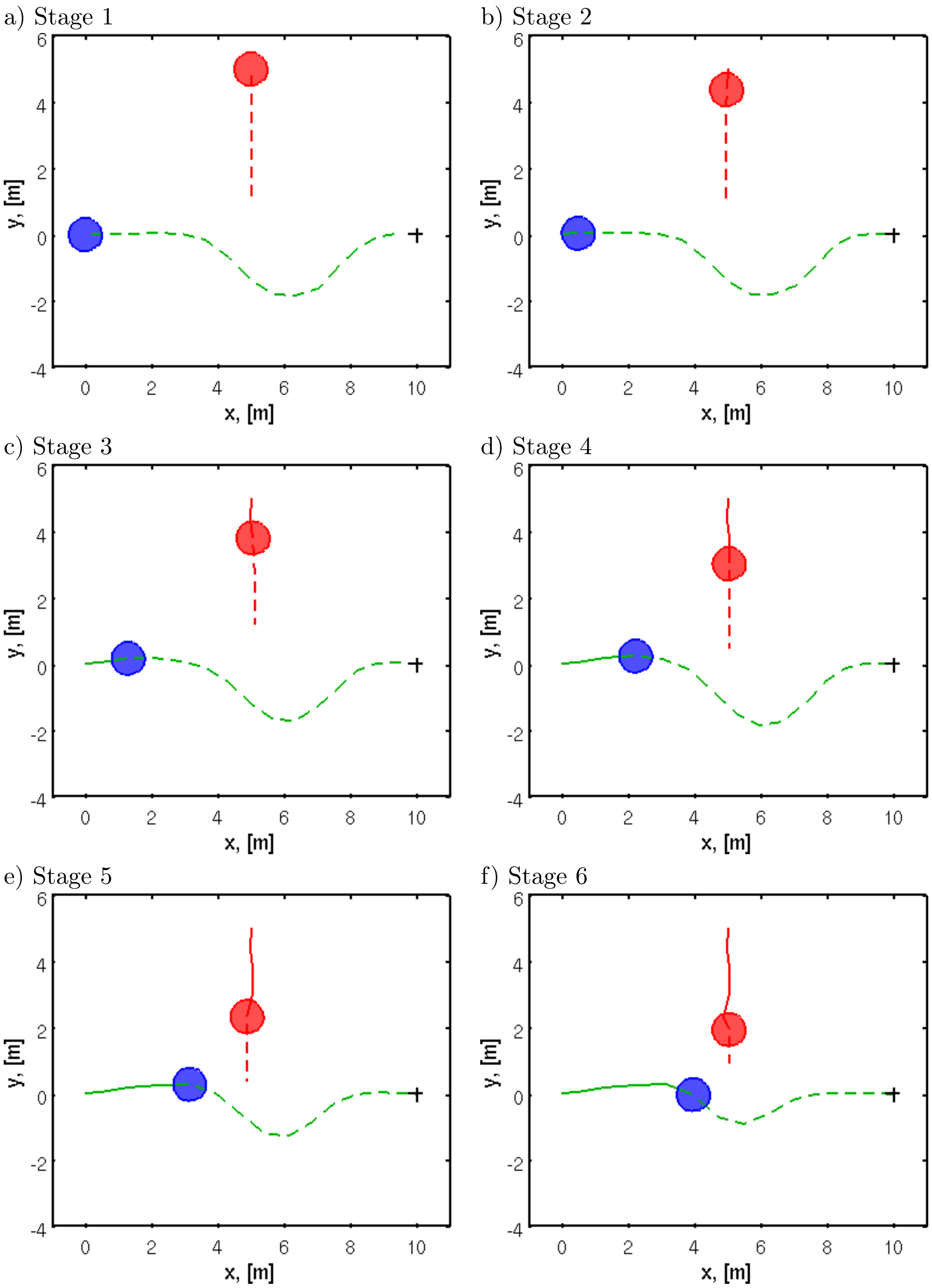


Figure 6.6: Executed (solid) and planned (dashed) trajectories at stages 1, 2, 3, 4, 5, and 6 with a friendly agent model using the OLRHC approach. The robot (green) is forced to pass behind the agent (red), resulting in a longer executed path than the PCL case.

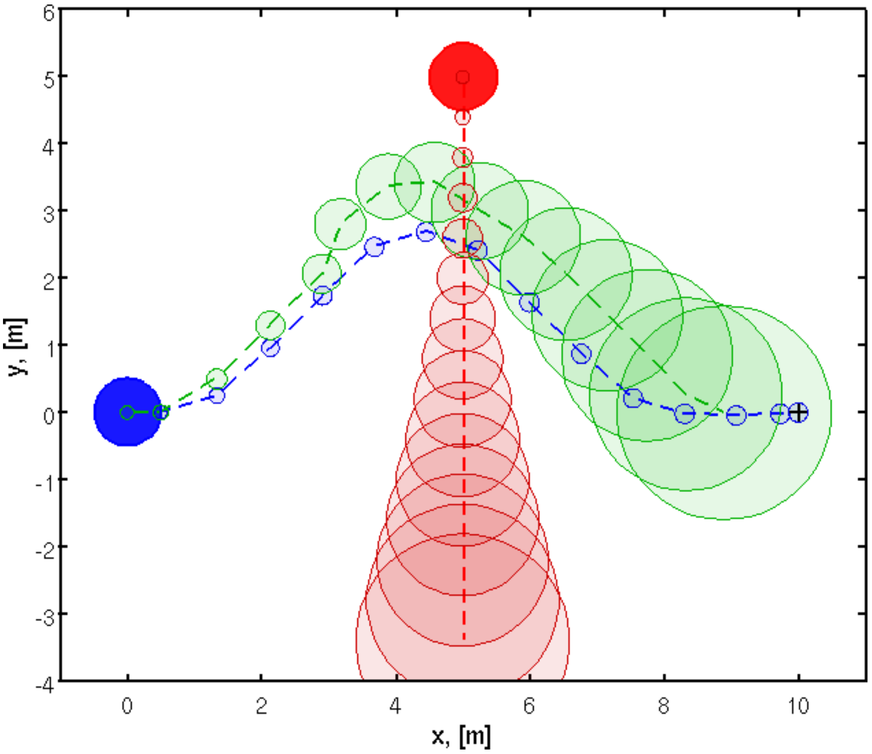


Figure 6.7: Planned trajectories for the robot and the *neutral* agent model, using the OLRHC approach (green) and PCLRHC approach (blue), with 1-σ uncertainty ellipses along the planned trajectories. The robot and agent initial positions are indicated with the solid blue and solid red circles, respectively. The predicted agent trajectory, with the 1-σ uncertainty ellipses (OL case) is given in red.

6.2.2.1 Agent Model

The agent dynamic equation is of the form:

$$x_i^{Aj} \quad = \quad A_{Aj}x_{i-1}^{Aj} + A_{RA}x_{i-1}^{R} + F_{Aj}\omega_{i-1}^{Aj} \qquad (6.11)$$

where $\omega_{i-1}^{Aj} \sim N(0, W_{Aj})$ is white, Gaussian noise and

$$A_{Aj} = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ -k_x\Delta t & 0 & 1-2\sqrt{k_x}\Delta t & 0 \\ 0 & -k_y\Delta t & 0 & 1-2\sqrt{k_y}\Delta t \end{bmatrix}$$

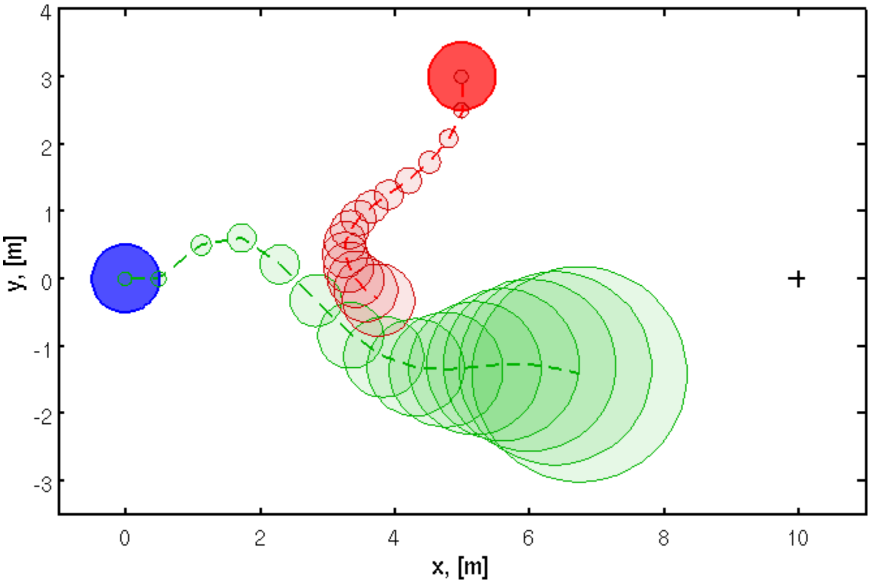


Figure 6.8: Planned trajectories for the robot and the adversarial agent, using the OLRHC approach (green) with 1-σ uncertainty ellipses along the planned trajectories. The robot and agent initial positions are indicated with the solid blue and solid red circles, respectively. The predicted agent trajectory, with the 1-σ uncertainty ellipses, is given in red.

$$A_{RA} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ k_x\Delta t & 0 & 0 & 0 \\ 0 & k_y\Delta t & 0 & 0 \end{bmatrix} \quad F_{Aj} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

k_x and k_y are the spring stiffnesses.

6.2.2.2 Simulation Results

A random walk model (Appendix B.1.3) with $W^R = 0.01 \times I_2$ and a linear position measurement model (Appendix B.2.2) with $V^R = 0.01 \times I_2$ are assumed for the robot. The friendly model presented in the previous section is assumed, with $W^A = 0.01 \times I_2$ and $V^A = 0.01 \times I_2$. The cost function of Section 6.1.3 is used. The goal state is $x_G = [10\ 0\ 0\ 0]$. The robot initial state is $x_0^R \sim \mathcal{N}(\hat{x}_{0|0}^R, \Sigma_{0|0}^A)$, where $\hat{x}_{0|0}^R = [0\ 0\ 1\ 0]^T$ and $\Sigma_{0|0}^R = 0.01 \times I_4$. The agent initial state is $x_0^A \sim \mathcal{N}(\hat{x}_{0|0}^A, \Sigma_{0|0}^A)$, where $\hat{x}_{0|0}^A = [5\ 3\ 0\ -1]^T$ and $\Sigma_{0|0}^A = 0.01 \times I_4$. The collision chance constraint is imposed at each stage with $\delta_c = 0.01$. $k_x = k_x = 0.15$ are assumed. Furthermore, each velocity component of the robot is constrained to the range $[-1.6\ 1.6]$.

The planned trajectory for the friendly agent, obtained from the OLRHC approach is plotted with the uncertainty ellipses in Figure 6.8. The dynamic obstacle predicted trajectory is plotted in red (with the uncertainty ellipses). The OLRHC approach is initially unable to get past the agent due to the growth in uncertainty. However, the robot is fast enough to move away from the agent,

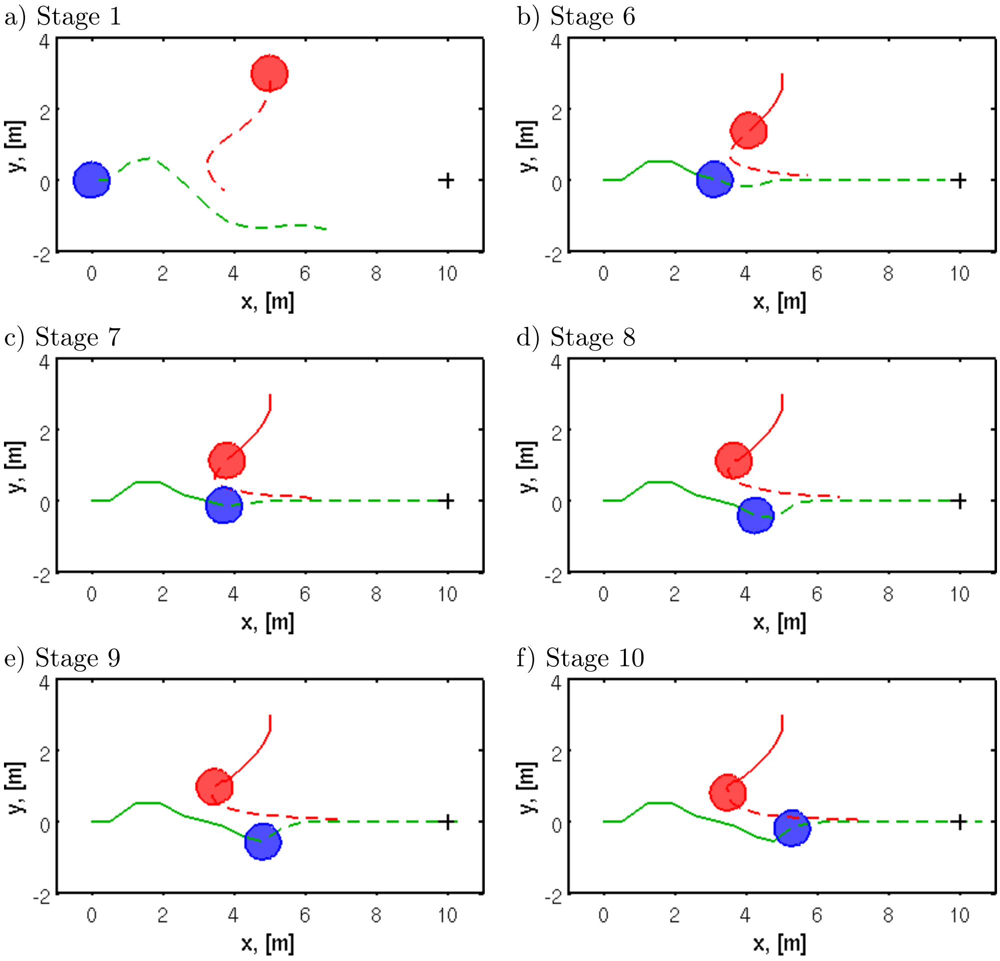


Figure 6.9: Executed (solid) and planned (dashed) trajectories at stages 1, 6, 7, 8, 9, and 10 with an adversarial agent model using the OLRHC approach. The robot (green) is forced to move away from the agent (red), but is able to move towards the goal due to the outerloop feedback mechanism.

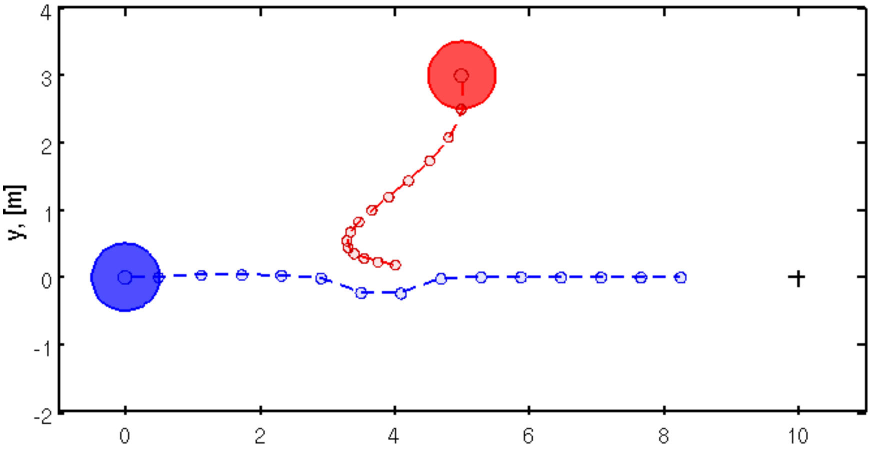


Figure 6.10: Planned trajectories for the robot and the adversarial agent, using the PCLRHC approach (blue) with 1-σ uncertainty ellipses along the planned trajectories. The robot and agent initial positions are indicated with the solid blue and solid red circles, respectively. The predicted agent trajectory, with the 1-σ uncertainty ellipses, is given in red.

and with the help of the outerloop feedback (by recursively solving the problem) is able to move towards the goal. The sequence of executed plans and planned trajectories are given in Figure 6.9. Another artifact of the OLRHC approach is that the robot cannot stay at the goal due to the growth in robot and agent uncertainty.

The planned trajectory for the adversarial agent, obtained from the PCLRHC approach is plotted with the uncertainty ellipses in Figure 6.10. The dynamic obstacle predicted trajectory is plotted in red (with the uncertainty ellipses). The reduction in conservatism allows the PCLRHC approach to plan past the agent and move towards the goal. The sequence of executed plans and planned trajectories are given in Figure 6.11.

In these interactive scenarios, an interesting question is: what happens when the robot cannot avoid the dynamic agents. This case was briefly investigated by limiting the robot velocity components to the range $[-1.2\ 1.2]$. At this level, the OLRHC approach was unable to find a reasonable path towards the goal. In fact, the problem became infeasible since the agent was able to move sufficiently close to the robot to violate the chance constraints for all possible control actions. Since the predicted positional uncertainty for the robot and agent is reduced with the PCLRHC approach, a feasible solution is still obtainable for this case. The sequence of executed plans and planned trajectories for the OLRHC approach is given in Figure 6.12. The obtained solution for the PCLRHC approach is very similar to the result shown in Figure 6.11. This is example where the OLRHC approach cannot solve the problem, but the PCLRHC is still able to find a safe and efficient solution. It should be noted that the PCLRHC approach will also eventually fail when the robot velocity is sufficiently constrained.

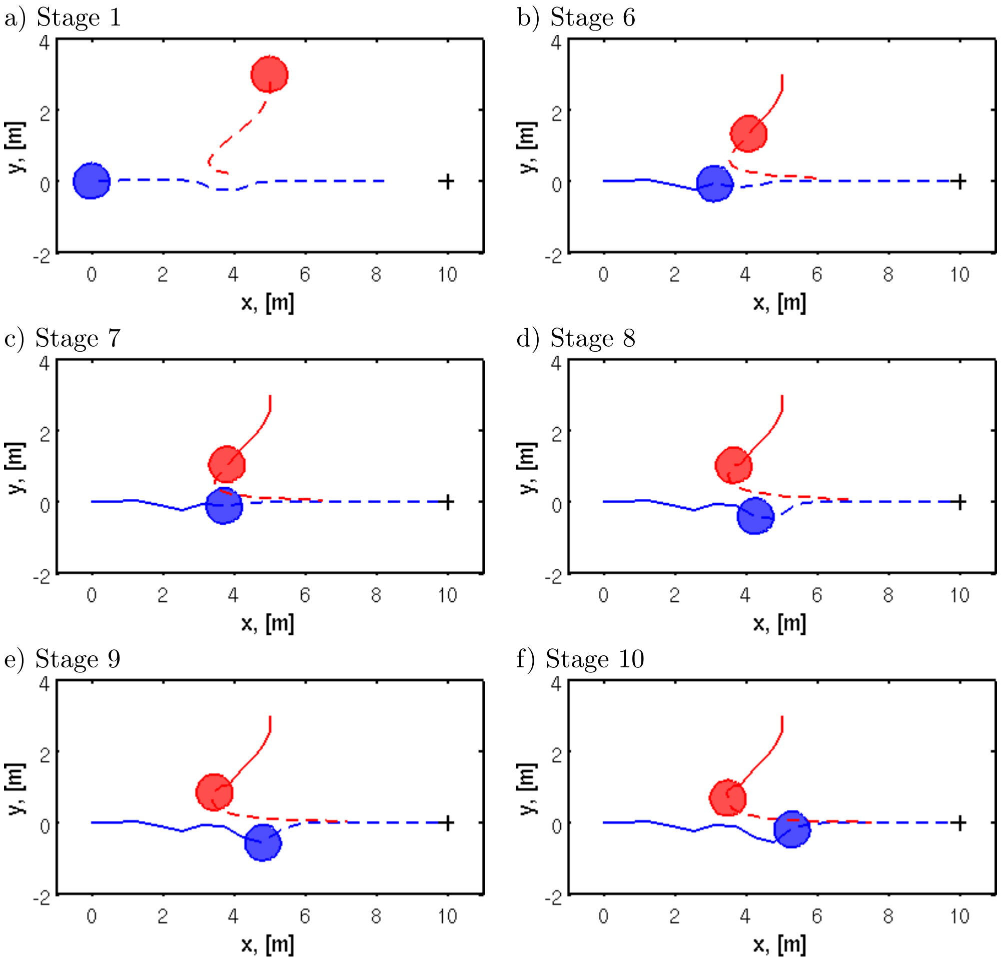


Figure 6.11: Executed (solid) and planned (dashed) trajectories at stages 1, 6, 7, 8, 9, and 10 with an adversarial agent model using the PCLRHC approach. The robot (blue) locally avoids the agent (red) to make progress towards the goal.

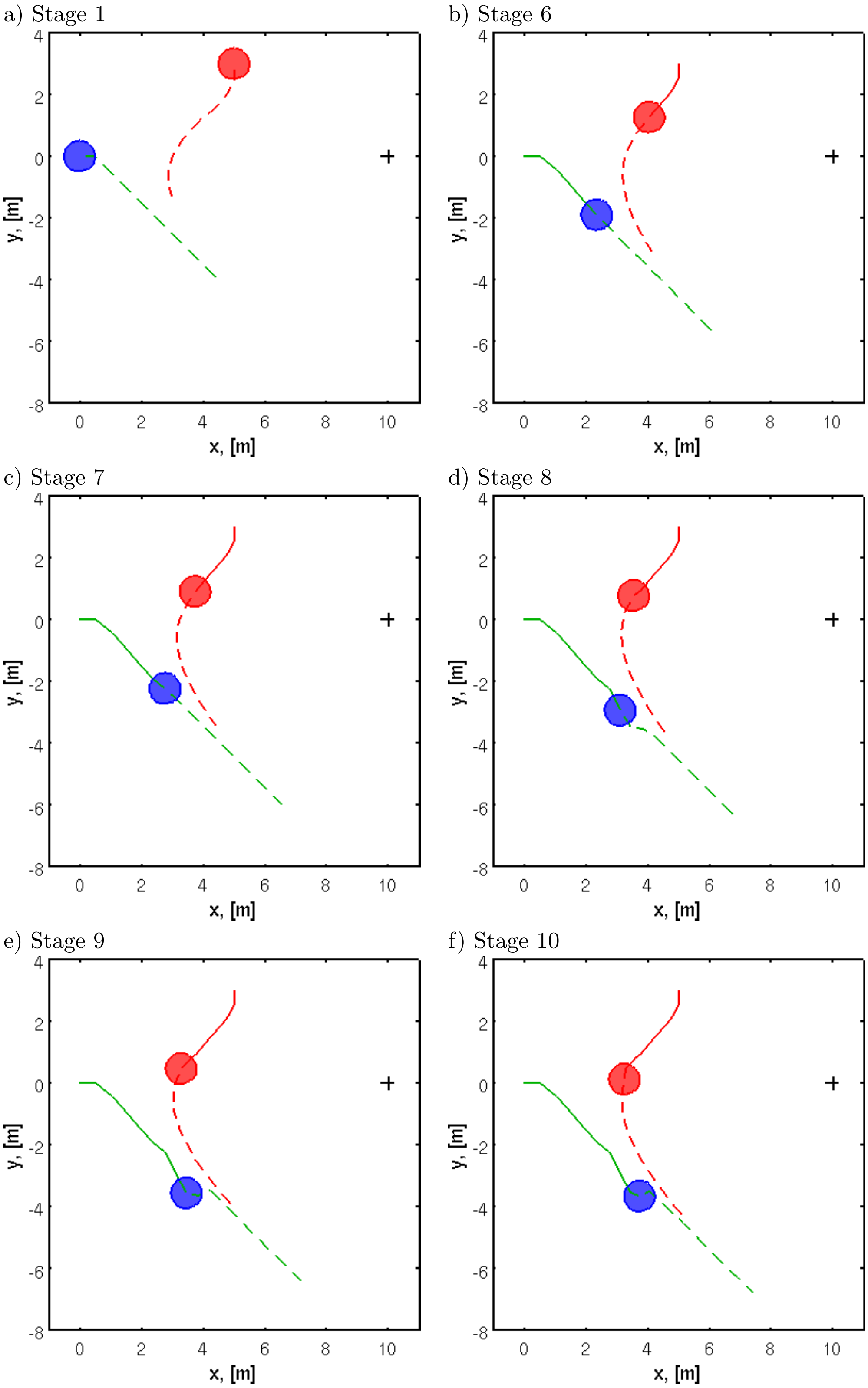


Figure 6.12: Executed (solid) and planned (dashed) trajectories at stages 1, 6, 7, 8, 9, and 10 with an adversarial agent model using the OLRHC approach. The robot (green) is forced to move away from the agent (red) and is unable to move towards the goal.

6.3 Summary

In this chapter, agent models that are dependent on the robot state are considered. To facilitate this dependence, the problem is posed in terms of the augmented state. The robot and agent state estimation problems are not separable anymore, but standard estimation results are still valid for the augmented system. Extensions to collision chance constraint results are presented to handle the dependence of the models.

In an attempt to model the effect of the robot trajectory on the quality of information about the agents, a robot-state dependent agent measurement model is introduced. This model allows for the robot to actively learn about the agents. Simulation results are presented to this effect.

Interactive agent models are considered: neutral agents, friendly agents, and adversarial agents. By accounting for the cooperation of the friendly agents, more efficient solutions are obtainable from the PCLRHC approach than with a neutral agent. The OLRHC approach is unable to leverage this cooperation due to the conservatism in the solutions.

The reduction in prediction uncertainty with the PCLRHC approach is leveraged when operating in environments with adversarial agents. The robot is unable to reach the goal location using the OLRHC approach, and must rely on the outerloop feedback mechanism to execute a reasonable path. The PCLRHC is also able to find solutions when the kinematic constraints limit the robot's ability to avoid the adversarial agent, where the OLRHC approach fails.

Chapter 7

Conclusions and Future Work

7.1 Summary of Thesis Contributions

The main contribution of this thesis is the incorporation of anticipated future information in the planning process for dynamic environments. While individual components of the motion planning problem in dynamic, cluttered, and uncertain environments (DCUE) have been previously considered, a comprehensive framework that integrates planning, prediction, and estimation as needed to solve the DCUE planning problem has been missing. This thesis represents the first formal effort to incorporate the effect of anticipated future measurements in the motion planning process in dynamic environments. As shown by example, the proper inclusion of these effects can improve robot performance in the presence of uncertain agent behavior. Additionally, a novel analysis of the chance constraints that model the probability of collisions between moving objects whose positions are uncertain is presented.

Approximations to the stochastic dynamic programming problem are used in Chapter 3 to derive the Partially Closed-Loop Receding Horizon Control (PCLRHC) algorithm. This algorithm accounts for the anticipated future measurements in the system. It is shown that approximation does not introduce artificial information into the planning problem, and that the least informative value for the anticipated measurement is used. As a result, the effect of the measurement on the obtained belief states is captured, but all information associated with the actual value of the measurement is ignored. The correct conditioning of the chance constraints are crucial to guarantee the safety of the system in the presence of uncertainty. The benefit of this algorithm over state-of-the-art algorithms is demonstrated for a simple example in a static environment: obtained planned solutions are less conservative and planned and executed paths are more similar. The safety of the system and related issues is illustrated in a car-following example. The PCLRHC approach successfully manages the growth in uncertainty as the system is propagated, which allows the algorithm to find feasible solutions in environments with agents with more complex behaviors than was previously possible.

In Chapter 4 the PCLRHC algorithm is applied to dynamic scenarios where the agent behavior is simple and independent of the robot actions. The benefit of the PCLRHC approach over the OLRHC approach is demonstrated with a Monte-Carlo simulation. On average, the PCLRHC approach obtains shorter planned and executed paths, with a significant (>20%) improvement in executed path length in 17.5% of the cases. A novel approach to probabilistic obstacle avoidance is also presented that accounts for the uncertainty associated with the agent location and geometry.

More complex dynamic example scenarios are considered in Chapter 5. Complicated agent behaviors are induced by incorporating environment structure in the agent models such as possible destinations, or by using multiple simple models. Extensions to the estimation results and the probabilistic obstacle avoidance approach to handle the resulting multimodal distributions are presented. Simulation results demonstrate the benefit of the PCLRHC approach over standard approaches in these dynamic scenarios.

In Chapter 6 the PCLRHC algorithm is applied to dynamic scenarios with interaction between the robot and the agent. The extension of the probabilistic obstacle avoidance approach to capture the dependence of the agent and robot models is presented. Simulation results demonstrate the ability of the robot to adjust its plan to obtain better information about the dynamic agents. The PCLRHC approach is also better able to leverage the cooperation of friendly agents and avoid adversarial agents, even with reduced dynamic capabilities, than the OLRHC approach.

7.2 Future Directions

The current implementation of the PCLRHC approach is limited by the solution of the underlying constrained nonlinear optimization problem. Currently, a commercial unconstrained nonlinear solver is used to obtain a solution, giving good results in the simple scenarios considered here. However, when applied to cluttered, dynamic environments, the problem is inherently strife with local minima, which is a known weakness for these optimization schemes. Additionally, these optimization schemes are notoriously computationally intensive. This is not an artifact of the PCLRHC approach developed here, but rather the specific implementation. A sampling-based planner was briefly investigated as a more efficient implementation which is less prone to local minima than the continuous optimization schemes. Standard sampling-based planning results cannot be applied directly since the belief space is not easily partitioned. However, initial results are positive and the indicate that the problem is very parallelizable. This line of research will be continued in the future.

Additionally, the approximate ellipsoidal enforcement of the probabilistic obstacle avoidance problem presented here is conservative (consistent with standard approaches). The conservatism can be reduced by using another approximation of the probability of collision (perhaps a heuristic approach), or a Monte-Carlo simulation approach. The latter also has the advantage of being able to handle the multimodal distributions that result from the complicated agent behavioral models.

Furthermore, this method is very parallelizable. Reducing the conservatism associated with these probabilistically enforced constraints will allow for even more complex behaviors to be considered.

Finally, the PCLRHC approach is a general suboptimal control approach which can be applied to problems beyond the traditional robotic motion planning problem considered here. The approach allows for explicit reasoning about the use of the anticipated future measurements in the system, and these notions might be extendable to other applications. Specifically, the robot grasping problem is of interest. The objects to be grasped have uncertain locations, geometries, and surface characteristics. New information about the object becomes available from measurements, and the above framework can potentially be applied to take this anticipated information into account.

Appendix A

Linear System, Quadratic Cost, Gaussian Noise

The DP ISI palgorithm gives a feedback control policy on belief space: the optimal control action is defined for every reachable belief state. The executed control sequence and realized measurement sequence determine the future belief states. Linear systems with quadratic cost and Gaussian noise terms is one of the few problems that allow closed-form solution [1, 6].

The problem of unconstrained feedback control problem with imperfect state information of Definition 2.3 is considered here. The problem is solved from the current stage, k, over a horizon, $N-k$. Assume a linear system with the following dynamic and measurement equations:

$$\begin{aligned} x_i &= Ax_{i-1} + Bu_{i-1} + F\omega_{i-1} \\ y_i &= Cx_i + H\nu_i. \end{aligned}$$

$\omega_{i-1} \sim N(0, W)$ and $\nu_i \sim N(0, V)$ are the Gaussian noise terms, independent of the past noise terms. A quadratic stage-additive cost function is assumed:

$$L(x_k) = x_N^T Q_N x_N + \sum_{i=k}^{N-1} \left\{ x_i^T Q_i x_i + u_i^T R_i u_i \right\}.$$

From Section 2.2, the strategy to solve this problem is first to convert it into belief space and then to use the DP ISI algorithm. The belief state is defined in Section 2.1.4.3 as:

$$\zeta_k \triangleq p(x_k|\eta_k) = f_\zeta(\zeta_{k-1}, u_{k-1}, y_k)$$

and the cost terms are converted to the belief space (Section 2.1.4.4):

$$L^{ISI}(\zeta_{0:N}, u_{0:N-1}, y_{1:N}) = l_N^{ISI}(\zeta_N) + \sum_{i=k}^{N-1} l_i^{ISI}(\zeta_i, u_i)$$

where

$$l_i^{ISI}(\zeta_i, u_i) \quad = \quad E\left[x_i^T Q_i x_i | \eta_i\right] + u_i^T R_i u_i$$

and

$$l_N^{ISI}(\zeta_N) = E\left[x_N^T Q_N x_N | \eta_N\right].$$

From Definition 2.7, the cost-to-go terms can be explicitly calculated. Starting at the final stage, N:

$$\begin{aligned} J_N(\zeta_N) &= l_N^{ISI}(\zeta_N) \\ &= E\left[x_N^T Q_N x_N | \eta_N\right]. \end{aligned}$$

At stage $N-1$, it is assumed that measurements $y_{0:N-1}$ have been obtained, so that ζ_{N-1} is known and thus it is necessary to select the optimal control action, u_{N-1}^*:

$$\begin{aligned} J_{N-1}(\zeta_{N-1}) &= \min_{u_{N-1}} \; l_{N-1}^{ISI}(\zeta_{N-1}, u_{N-1}) + E_{y_N}\left[J_N(f_\zeta(\zeta_{N-1}, u_{N-1}, y_N))|\eta_{N-1}\right] \\ &= \min_{u_{N-1}} \; E\left[x_{N-1}^T Q_{N-1} x_{N-1} | \eta_{N-1}\right] + u_{N-1}^T R_{N-1} u_{N-1} + E\left[E\left[x_N^T Q_N x_N | \eta_N\right] | \eta_{N-1}\right] \\ &= E\left[x_{N-1}^T Q_{N-1} x_{N-1} | \eta_{N-1}\right] + \min_{u_{N-1}} \left\{u_{N-1}^T R_{N-1} u_{N-1} + \right. \\ &\quad E\left[\left(A x_{N-1} + B u_{N-1} + F\omega_{N-1}\right)^T Q_N \left(A x_{N-1} + B u_{N-1} + F\omega_{N-1}\right) | \eta_{N-1}\right]\Big\}. \end{aligned}$$

Since $E\left[\omega_{N-1}|\eta_{N-1}\right] = 0$, this expression can be written as:

$$\begin{aligned} J_{N-1}(\zeta_{N-1}) &= E_{x_{N-1}}\left[x_{N-1}^T \left(A^T Q_N A + Q_{N-1}\right) x_{N-1} | \eta_{N-1}\right] + \\ &\quad E_{\omega_{N-1}}\left[\omega_{N-1}^T Q_N \omega_{N-1} | \eta_{N-1}\right] + \\ &\quad \min_{u_{N-1}} \left\{u_{N-1}^T \left(B^T Q_N B + R_{N-1}\right) u_{N-1} + \right. \\ &\quad \left. 2E\left[x_{N-1}^T | \eta_{N-1}\right] A^T Q_N B u_{N-1}\right\}. \end{aligned}$$

The optimal cost can be selected by evaluating the first and second derivatives of the cost, to obtain:

$$\begin{aligned} u_{N-1}^* &= -\left(B^T Q_N B + R_{N-1}\right)^{-1} B^T Q_N A E\left[x_{N-1}|\eta_{N-1}\right] \\ &\triangleq \pi_{N-1}^*\left(\zeta_{N-1}\right). \end{aligned}$$

Upon substitution, and introducing the estimation error $e_{N-1} \triangleq x_{N-1} - E\left[x_{N-1}^T|\eta_{N-1}\right]$, the cost-to-go at stage $N-1$ is obtained:

$$\begin{aligned} J_{N-1}(\zeta_{N-1}) &= E\left[x_{N-1}^T K_{N-1} x_{N-1}|\eta_{N-1}\right] + E\left[e_{N-1}^T P_{N-1} e_{N-1}|\eta_{N-1}\right] + \\ & E\left[\omega_{N-1}^T Q_N \omega_{N-1}|\eta_{N-1}\right] \end{aligned}$$

where the matrices are given by:

$$\begin{aligned} P_{N-1} &= A^T Q_N B \left(B^T Q_N B + R_{N-1}\right)^{-1} B^T Q_N A \\ K_{N-1} &= A^T Q_N A - P_{N-1} + Q_{N-1}. \end{aligned}$$

Next, consider stage $N-2$. The cost-to-go is given by:

$$\begin{aligned} J_{N-2}(\zeta_{N-2}) &= \min_{u_{N-2}} l_{N-2}^{ISI}(\zeta_{N-2}, u_{N-2}) + E_{y_{N-1}}\left[J_{N-1}(f_\zeta(\zeta_{N-2}, u_{N-2}, y_{N-1}))|\eta_{N-2}\right] \\ &= E\left[x_{N-2}^T Q_{N-2} x_{N-2}|\eta_{N-2}\right] + E\left[E\left[e_{N-1}^T P_{N-1} e_{N-1}|\eta_{N-1}\right]|\eta_{N-2}\right] + \\ & E\left[E\left[\omega_{N-1}^T Q_N \omega_{N-1}|\eta_{N-1}\right]|\eta_{N-2}\right] + \\ & \min_{u_{N-2}} \left\{u_{N-2}^T R_{N-2} u_{N-2} + E\left[E\left[x_{N-1}^T K_{N-1} x_{N-1}|\eta_{N-1}\right]|\eta_{N-2}\right]\right\} \end{aligned}$$

where it is important to note the the term associated with the estimation error is not included in the minimization. The reason for this is that for linear systems this error is independent of the control sequence (see [6]). In other words, for this system, the choice of control actions have no effect on the quality of estimation. Substituting for $x_{N-1} = Ax_{N-2} + Bu_{N-2} + \omega_{N-2}$, the optimal control action is obtained:

$$\begin{aligned} u_{N-2}^* &= -\left(B^T K_{N-1} B + R_{N-2}\right)^{-1} B^T K_{N-1} A E\left[x_{N-2}|\eta_{N-2}\right] \\ &\triangleq \pi_{N-2}^*\left(\zeta_{N-2}\right). \end{aligned}$$

Using induction, a recursion for the optimal control policy can be obtained for stage i

$$\pi_i^*\left(\zeta_i\right) = L_i E\left[x_i|\eta_i\right]$$

where

$$L_i = -\left(B^T K_{i+1} B + R_i\right)^{-1} B^T K_{i+1} A$$

and K_i given recursively by the Riccati equation

$$\begin{aligned}
K_N &= Q_N \\
P_i &= A^T K_{i+1} B \left(B^T K_{i+1} B + R_i\right)^{-1} B^T K_{i+1} A \\
K_i &= A^T K_{i+1} A - P_i + Q_i.
\end{aligned}$$

The key observation for this derivation is noting that the estimation quality is independent of the sequence of control actions. This allows the estimation and control processes to be separated (separation principle in control theory).

These results cannot be extended to non-linear systems or systems with non-Gaussian noise (including systems with state-dependent noise) since closed form solutions of the optimal estimator cannot be obtained. Some results have been obtained for systems with multiplicative noise by assuming a simple form for the estimator [54].

Appendix B

System Models for Objects

The models that are used throughout this work are described here. A discrete-time state transition equation describes the evolution of the system:

$$x_i = f_{i-1}\left(x_{i-1}, u_{i-1}, \omega_{i-1}\right), \qquad i = k+1, \ldots, N-1 \tag{B.1}$$

where x_{i-1} is the state (which is not known or directly measurable), $u_{i-1}$is the control, and $\omega_{i-1}(x_{i-1}, u_{i-1})$ is the disturbance. The disturbance is described by the conditional distribution $p(\omega_{i-1}|x_{i-1}, u_{i-1})$ and is independent of the previous disturbances, $\omega_{0:i-2}$. The state of the system is defined as (unless stated otherswise):

$$x_i = \begin{bmatrix} x_i^{(1)} \\ x_i^{(2)} \\ x_i^{(3)} \\ x_i^{(4)} \end{bmatrix} = \begin{bmatrix} x - \text{position of object} \\ y - \text{position of object} \\ x - \text{velocity of object} \\ y - \text{velocity of object} \end{bmatrix}. \tag{B.2}$$

The controls consist of two components:

$$u_i = \begin{bmatrix} u_i^{(1)} \\ u_i^{(2)} \end{bmatrix}. \tag{B.3}$$

A sensor mapping, $h_i(x_i,\nu_i)$ maps every state into a measurement y_i with measurement noise, $\nu_i(x_i)$:

$$y_i = h_i(x_i, \nu_i). \tag{B.4}$$

The measurement noise is described by a conditional distribution $p(\nu_i|x_i)$.

B.1 Dynamic Models

B.1.1 1-D Random Walk Dynamic Model (Position Only)

The position of the object, x_i, is the state and the instantaneous velocity of the object is controlled, resulting in a 1-D dynamic equation with independent white Gaussian noise terms. The dynamic equation is of the form:

$$x_i = x_{i-1} + \Delta t \; u_{i-1} + \omega_{i-1}$$

where $\omega_{i-1} \sim N(0, W)$ is the Gaussian disturbance that is independent of the previous noise terms, $\omega_{0:i-2}$.

B.1.2 1-D Random Walk Dynamic Model (Position and Velocity)

The state is the 1-D position and velocity of the object:

$$x_i = \begin{bmatrix} x_i^{(1)} \\ x_i^{(2)} \end{bmatrix} = \begin{bmatrix} \text{position} \\ \text{velocity} \end{bmatrix},$$

resulting in a 2-D dynamic equation with independent white Gaussian noise terms. The dynamic equation is of the form:

$$x_i = Ax_{i-1} + Bu_{i-1} + F\omega_{i-1}$$

where $\omega_{i-1} \sim N(0, W)$ is the Gaussian disturbance that is independent of the previous noise terms, $\omega_{0:i-2}$. The parameter matrices are given by:

$$A = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \qquad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \qquad F = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

B.1.3 2-D Random Walk Dynamic Model

The random walk model has a 4-D linear dynamic equation with independent white Gaussian noise terms. The dynamic equation is of the form:

$$x_i = Ax_{i-1} + Bu_{i-1} + F\omega_{i-1} \tag{B.5}$$

where $\omega_{i-1} \sim N(0, W)$ is the Gaussian disturbance that is independent of the previous noise terms, $\omega_{0:i-2}$. The parameter matrices are given by:

$$A = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \quad F = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}. \tag{B.6}$$

The controls and noise enter through the velocity components of the system.

B.1.4 2-D Agent Drawn to Destination

Assume a model where the agent is drawn towards a target location: $t_i = [t_i^x \; t_i^y]^T$. The model is based on a spring-mass-damper system analogy, which has unit mass and is critically damped. Let k_x be the spring stiffness in the x-direction, and k_y be the spring stiffness in the y-direction. The dynamic equation has the form:

$$x_i = Ax_{i-1} + Nt_{i-1} + Bu_{i-1} + F\omega_{i-1} \tag{B.7}$$

where $\omega_{i-1} \sim N(0, W)$ is the white Gaussian disturbance that is independent of the previous noise terms, $\omega_{0:i-2}$. The parameter matrices are given by:

$$A = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ -k_x \Delta t & 0 & 1 - 2\sqrt{k_x}\Delta t & 0 \\ 0 & -k_y \Delta t & 0 & 1 - 2\sqrt{k_y}\Delta t \end{bmatrix} \quad N = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ k_x \Delta t & 0 \\ 0 & k_y \Delta t \end{bmatrix} \tag{B.8}$$

and

$$B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \quad F = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

B.2 Measurement Models

B.2.1 1-D Linear Position Measurement Model

Consider the 1-D linear measurement model where the position of the object is measured:

$$y_i = x_i + \nu_i$$

where $\nu_i \sim N(0, V)$ is the white Gaussian measurement noise that is independent of the previous noise terms, $\nu_{1:i-1}$.

B.2.2 2-D Linear Position Measurement Model

Consider a linear measurement equation of the form:

$$y_i = Cx_i + H\nu_i \tag{B.9}$$

where $\nu_i \sim N(0, V)$ is the white Gaussian measurement noise that is independent of the previous noise terms, $\nu_{1:i-1}$. The parameter matrices are given by:

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \qquad H = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}. \tag{B.10}$$

B.2.3 Linearized Distance-Dependent Position Measurement Model

An agent measurement model is desired with larger measurement noise when the robot and the agent are far apart and with an improvement in measurement quality when the two are in close proximity. It is necessary to distinguish between the robot and agent states here: let x_i^R be the robot state. The agent variables are annotated with a superscript A. Assume a measurement equation of the form:

$$y_i^A = C_A x_i^A + H_A \nu_i^A + \sum_{l=1}^{n_\xi} \Phi_l g(x_i^A, x_i^R)\xi_i^{(l)}$$

where $\nu_i^A \sim N(0, V_A)$, and $\xi_i^{(l)} \sim N(0, \sigma_\xi^2)$ $\forall l = 1, \ldots, n_\xi$ are independent white Gaussian noise terms. The parameter matrices are given by Eq. (B.10). The term $g(x_i^A, x_i^R)$ is some nonlinear function of the robot and agent states. The variable Φ_l is a matrix that aligns the l^{th} state dependent noise term with the appropriate component of the measurement. The linearized distance-squared function, $g(x_i^A, x_i^R) = (x_i^R - x_i^A)^T(x_i^R - x_i^A)$, is used. This function is linearized about the conditional mean of the robot and agent states ($\hat{x}_{i|i-1}^R$ and $\hat{x}_{i|i-1}^A$):

$$g(x_i^R, x_i^A) \approx g(\hat{x}_{i|i-1}^R, \hat{x}_{i|i-1}^A) + \frac{\partial g}{\partial x_i^R}|_{\hat{x}_{i|i-1}^R, \hat{x}_{i|i-1}^A} \left(x_i^R - \hat{x}_{i|i-1}^R\right) + \frac{\partial g}{\partial x_i^A}|_{\hat{x}_{i|i-1}^R, \hat{x}_{i|i-1}^A} \left(x_i^A - \hat{x}_{i|i-1}^A\right).$$

The resulting linear measurement model with state dependent measurement noise has the form:

$$y_i^A = C_A x_i^A + H_A \nu_i^A + \sum_{l=1}^{n_\xi} \overline{H}_A^{(l)} \xi_i^{(l)} + \sum_{l=1}^{n_\xi} G_A^{(l)} x_i^A \xi_i^{(l)} + \sum_{l=1}^{n_\xi} G_R^{(l)} x_i^R \xi_i^{(l)} \tag{B.11}$$

where

$$\overline{H}_A^{(l)} = \begin{cases} -\Phi_l (\hat{x}_{i|i-1}^R - \hat{x}_{i|i-1}^A)^T (\hat{x}_{i|i-1}^R - \hat{x}_{i|i-1}^A) & \text{if } E\left[(x_i^R - x_i^A)^T (x_i^R - x_i^A) | \bar{\zeta}_{i-1}\right] \leq d_{max}^2 \\ \Phi_l d_{max}^2 & \text{otherwise} \end{cases}$$

$$G_A^{(l)} = \begin{cases} -2\Phi_l (\hat{x}_{i|i-1}^R - \hat{x}_{i|i-1}^A)^T & \text{if } E\left[(x_i^R - x_i^A)^T (x_i^R - x_i^A) | \bar{\zeta}_{i-1}\right] \leq d_{max}^2 \\ 0 & \text{otherwise} \end{cases}$$

$$G_R^{(l)} = \begin{cases} 2\Phi_l (\hat{x}_{i|i-1}^R - \hat{x}_{i|i-1}^A)^T & \text{if } E\left[(x_i^R - x_i^A)^T (x_i^R - x_i^A) | \bar{\zeta}_{i-1}\right] \leq d_{max}^2 \\ 0 & \text{otherwise} \end{cases} .$$

Appendix C

Probability of Collision

C.1 Independent, Gaussian Objects

Assume that $p(x_R) = N_{x_R}(\hat{x}_R, \Sigma_R)$ and $p(x_A) = N_{x_A}(\hat{x}_A, \Sigma_A)$ are independent. From Eq. (4.3), the probability of collision involves the integral:

$$\int_{x_R} p(x_A = x_R | x_R) p(x_R) dx_R.$$

Using the independence assumption:

$$\int_{x_R} p(x_A = x_R | x_R) p(x_R) dx_R \quad = \quad \int_{x_R} N_{x_R}(\hat{x}_A, \Sigma_A) N_{x_R}(\hat{x}_R, \Sigma_R) dx_R.$$

It is known that the product of independent Gaussian distributions is a weighted Gaussian (e.g., [46]):

$$N_{x_R}(\hat{x}_A, \Sigma_A) N_{x_R}(\hat{x}_R, \Sigma_R) = N_{\hat{x}_R}(\hat{x}_A, \Sigma_R + \Sigma_A) N_{x_R}(m_c, \Sigma_c)$$

where

$$m_c = \Sigma_c \left(\Sigma_R^{-1} \hat{x}_R + \Sigma_A^{-1} \hat{x}_A \right)$$

and

$$\Sigma_c = \left(\Sigma_R^{-1} + \Sigma_A^{-1} \right)^{-1}.$$

Evaluating the integral, we obtain:

$$\int_{x_R} p(x_A = x_R | x_R) p(x_R) dx_R \quad = \quad N_{\hat{x}_R}(\hat{x}_A, \Sigma_R + \Sigma_A) \int_{x_R} N_{x_R}(m_c, \Sigma_c) dx_R$$

and the integral evaluates to one, giving the desired result:

$$P(C) = \frac{1}{\sqrt{\det\left(2\pi\Sigma_C\right)}} \exp\left[-\frac{1}{2} \left(\hat{x}^R - \hat{x}^A \right)^T \Sigma_C^{-1} \left(\hat{x}^R - \hat{x}^A \right) \right] \times V_{\mathcal{B}}. \tag{C.1}$$

C.2 Jointly Gaussian Point Objects

From Eq. (4.3), the probability of collision involves the integral:

$$\int_{x_R} p(x_A = x_R|x_R)p(x_R)dx_R$$

Assume that the robot and agent state (augmented state) are jointly Gaussian (the time index is dropped for brevity):

$$p(x) = N\left(\left[\begin{array}{c} \hat{x}_R \\ \hat{x}_A \end{array}\right], \left[\begin{array}{cc} \Sigma_R & \Sigma_M \\ \Sigma_M^T & \Sigma_A \end{array}\right]\right).$$

The dependence condition results in $\Sigma_M \neq 0$. Then, from the marginal distribution of the robot state:

$$p(x_R) = N(\hat{x}_R, \Sigma_R),$$

and from the conditional distribution of the agent state, x_A:

$$p(x_A|x_R) = N\left(\bar{x}_A(x_R), \overline{\Sigma}_A\right)$$

where

$$\bar{x}_A(x_R) = \hat{x}_A + \Sigma_M^T \Sigma_R^{-1}(x_R - \hat{x}_R)$$

and

$$\overline{\Sigma}_A = \Sigma_A - \Sigma_M^T \Sigma_R^{-1} \Sigma_M.$$

The product of the two Gaussian distributions have the form:

$$\begin{aligned} p(x^A = x^R|x^R)p(x^R)dx^R &= N_{x_A = x_R}\left(\bar{x}_A(x_R), \overline{\Sigma}_A\right) \times N_{x_R}(\hat{x}_R, \Sigma_R) \\ &= \alpha \exp\left(-L(x_R, \hat{x}_R, \hat{x}_A)\right) \end{aligned}$$

where

$$\alpha = \frac{1}{\sqrt{\det\left(2\pi\Sigma_R\right)}} \frac{1}{\sqrt{\det\left(2\pi\overline{\Sigma}_A\right)}}$$

and

$$L(x_R, \hat{x}_R, \hat{x}_A) = \frac{1}{2}\left(x_R - \hat{x}_R\right)^T \Sigma_R^{-1}\left(x_R - \hat{x}_R\right) + \frac{1}{2}\left(x_A - \bar{x}_A(x_R)\right)^T \overline{\Sigma}_A^{-1}\left(x_A - \bar{x}_A(x_R)\right).$$

The function $L(x_R, \hat{x}_R, \hat{x}_A)$ is quadratic in x_R and the next step is to partition this function as $L(x_R, \hat{x}_R, \hat{x}_A) = L_1(x_R, \hat{x}_R, \hat{x}_A) + L_2(\hat{x}_R, \hat{x}_A)$ so that the integral can be evaluated efficiently. With this partition we obtain:

$$\int_{x_R} p(x_A = x_R|x_R)p(x_R)dx_R \quad = \quad \alpha\left[\int_{x_R} \exp(-L_1(x_R, \hat{x}_R, \hat{x}_A)dx_R\right]\exp(-L_2(\hat{x}_R, \hat{x}_A)). \quad (C.2)$$

By evaluating the first and second partial derivatives[1] with respect to x_R, $L_1(x_R, \hat{x}_R, \hat{x}_A)$ can be written as a normal distribution in terms of x_R:

$$\frac{\partial L(x_R, \hat{x}_R, \hat{x}_A)}{\partial x_R} = 0 \quad \Rightarrow \quad \tilde{x}_R = \Psi \bar{x}_R$$

where

$$\bar{x}_R \quad = \quad \left(\Sigma_R^{-1} - \overline{\Sigma}_A^{-1}\Sigma_M^T\Sigma_R^{-1} + \Sigma_R^{-1}\Sigma_M\overline{\Sigma}_A^{-1}\Sigma_M^T\Sigma_R^{-1}\right)\hat{x}_R + \left(\overline{\Sigma}_A^{-1} - \Sigma_R^{-1}\Sigma_M\overline{\Sigma}_A^{-1}\right)\hat{x}_A$$

and

$$\begin{aligned}\Psi \quad &\triangleq \quad \left(\frac{\partial^2 L(x_R, \hat{x}_R, \hat{x}_A)}{\partial x_R^2}\right)^{-1} \\ &= \quad \Sigma_R - (\Sigma_R - \Sigma_M)\left(\Sigma_R + \Sigma_A - \Sigma_M - \Sigma_M^T\right)\left(\Sigma_R - \Sigma_M^T\right).\end{aligned}$$

Define $L_1(x_R, \hat{x}_R, \hat{x}_A) \triangleq \frac{1}{2}(x_R - \bar{x}_R)^T\Psi^{-1}(x_R - \bar{x}_R)$, and noting that

$$\int_{x_R} \exp\left(-L_1(x_R, \hat{x}_R, \hat{x}_A)\right)dx_R = \sqrt{\det(2\pi\Psi)},$$

then Eq. (C.2) becomes

$$\int_{x_R} p(x_A = x_R|x_R)p(x_R)dx_R \quad = \quad \alpha\sqrt{\det(2\pi\Psi)}\exp(-L_2(\hat{x}_R, \hat{x}_A)). \quad (C.3)$$

Recall that $L_2(\hat{x}_R, \hat{x}_A) = L(x_R, \hat{x}_R, \hat{x}_A) - L_1(x_R, \hat{x}_R, \hat{x}_A)$, which is quadratic in $\hat{x}_R$ and $\hat{x}_A$. A

[1] For a generic normal distribution of random variable z, the mean, μ, and the covariance, Γ, of the distribution can be recovered from the first and second derivatives of the function $L(z)$ ([53], Section 3.2.4, p. 46):

$$\begin{aligned}\mathbb{N}(z;\mu,\Gamma) \quad &= \quad \frac{1}{\sqrt{\det(2\pi\Gamma)}}\exp(-L(z)) \\ L(z) \quad &= \quad \frac{1}{2}(z-\mu)^T\Gamma^{-1}(z-\mu)\end{aligned}$$

then

$$\begin{aligned}\frac{\partial L(z)}{\partial z} \quad &= \quad \Gamma^{-1}(z-\mu) \\ \frac{\partial^2 L(z)}{\partial z^2} \quad &= \quad \Gamma^{-1}\end{aligned}$$

and setting the first derivative equal to zero:

$$\frac{\partial L(z)}{\partial z} = 0 \quad \Rightarrow \quad \bar{z} = \mu.$$

similar strategy is used to show that Eq. (C.3) is in the form of a normal distribution, $N_{\hat{x}_R}(m, \Phi)$. Taking the partial derivatives with respect to $\hat{x}_R$:

$$\frac{\partial L_2(\hat{x}_R, \hat{x}_A)}{\partial \hat{x}_R} = 0 \quad \Rightarrow \quad m = \hat{x}_A,$$

$$\begin{aligned}\Phi &\triangleq \left(\frac{\partial^2 L_2(\hat{x}_R, \hat{x}_A)}{\partial x_R^2}\right)^{-1} \\ &= \Sigma_R + \Sigma_A - \Sigma_M - \Sigma_M^T,\end{aligned}$$

and

$$\begin{aligned}\alpha\sqrt{\det(2\pi\Psi)} &= \frac{\sqrt{\det(2\pi\Psi)}}{\sqrt{\det\left(2\pi\Sigma_R\right)\det\left(2\pi\overline{\Sigma}_A\right)}} \\ &= \frac{1}{\sqrt{\det\left(2\pi\Phi\right)}}.\end{aligned}$$

Now,

$$\begin{aligned}L_2(\hat{x}_R, \hat{x}_A) &= \frac{1}{2}\left(\hat{x}_R - m\right)^T \Phi^{-1}\left(\hat{x}_R - m\right) \\ &= L(x_R, \hat{x}_R, \hat{x}_A) - L_1(x_R, \hat{x}_R, \hat{x}_A)\end{aligned}$$

and, as desired, Eq. (C.3) becomes:

$$\int_{x_R} p(x_A = x_R | x_R) p(x_R) dx_R = N_{\hat{x}_R}\left(\hat{x}_A, \Sigma_R + \Sigma_A - \Sigma_M - \Sigma_M^T\right). \tag{C.4}$$

Appendix D

Estimation for Systems with Multiplicative Noise

Systems with multiplicative noise are of interest in this research in two scenarios: (i) the parametric model of the system contains uncertain, unknown parameters that get multiplied with the uncertain system state (Section 4.2.1.2), and (ii) an agent measurement model is desired where the quality of the measurement is a function of the distance between the robot and agent (Section 6.1.1). The former is presented in detail below, and the latter is a special case where the additional dependence between the noise terms needs to be accounted for.

D.1 Systems with Multiplicative Noise

Assume a system of the form:

$$\begin{aligned} x_{i+1} &= Ax_i + Bu_i + F\omega_i + \sum_{k=1}^{n_\xi} M^{(k)}\xi_i^{(k)}x_i \\ y_i &= Cx_i + H\nu_i + \sum_{k=1}^{n_\xi} G^{(k)}\xi_i^{(k)}x_i \end{aligned}$$

where $\omega_i \sim N(0, W)$ and $\nu_i \sim N(0, V)$ and are independent white Gaussian noise terms. The multiplicative noise terms are individually white Gaussian, $\xi_{i-1}^{(k)} \sim N(0, \sigma_\xi^2)\ \forall k = 1, \ldots, n_\xi$ and are assumed to be independent of the state and other noise terms. The product of two Gaussian variables is non-Gaussian and an optimal estimator is not generally available. An approximate estimator can be derived by approximating the optimal estimator in the form of a Luenberger estimator. Assume $\hat{x}_{i|i} = E\left[x_i|y_{1:i}\right] \approx \hat{x}_{i|i-1} + K_i(y_i - \hat{y}_{i|i-1})$. The estimation error is defined as $e_i \triangleq x_i - \hat{x}_{i|i}$ and the prediction error as $\varepsilon_i \triangleq x_i - \hat{x}_{i|i-1}$. The expected value of the state prior to incorporating the measurement, using the independence of $\xi_i^{(k)}$ and the state, is:

$$\hat{x}_{i|i-1} \quad = \quad A\hat{x}_{i-1|i-1} + Bu_{i-1}.$$

The prior covariance of the state is equal to the covariance of the prediction error, $\Sigma_{i|i-1} = E\left[\varepsilon_i \varepsilon_i^T | y_{1:i-1}\right]$ and is given by:

$$\Sigma_{i|i-1} \quad = \quad A\Sigma_{i-1|i-1}A^T + FWF^T + \sum_{k=1}^{n_\xi} \sigma_\xi^2 M^{(k)} \left(\Sigma_{i-1|i-1} + \hat{x}_{i-1|i-1}\hat{x}_{i-1|i-1}^T\right) M^{(k)T}.$$

The estimation gain, K_i, is chosen to minimize the posterior covariance of the state, $\Sigma_{i|i} = E\left[e_i e_i^T | y_{1:i}\right]$. The estimation error is written in terms of the prediction error and the innovation, $\tilde{y}_i \triangleq y_i - \hat{y}_{i|i-1}$, where

$$\hat{y}_{i|i-1} = E\left[y_i | y_{1:i-1}\right] = C\hat{x}_{i|i-1}$$

and

$$\begin{aligned} \Gamma_{i|i-1} \quad &\triangleq \quad E\left[\tilde{y}_i \tilde{y}_i^T | y_{1:i-1}\right] \\ &= \quad C\Sigma_{i|i-1}C^T + HVH^T + \sum_{k=1}^{n_\xi} \sigma_\xi^2 G^{(k)} \left(\Sigma_{i|i-1} + \hat{x}_{i|i-1}\hat{x}_{i|i-1}^T\right) G^{(k)T}. \end{aligned}$$

Note that the prediction error and innovation are per definition independent of the new measurement, so that $E\left[\varepsilon_i \varepsilon_i^T | y_{1:i}\right] = E\left[\varepsilon_i \varepsilon_i^T | y_{1:i-1}\right]$ and $E\left[\tilde{y}_i \tilde{y}_i^T | y_{1:i}\right] = E\left[\tilde{y}_i \tilde{y}_i^T | y_{1:i-1}\right]$. The gain that minimizes the estimation covariance is:

$$K_i \quad = \quad \Sigma_{i|i-1} C^T \Gamma_{i|i-1}^{-1}$$

resulting in the optimal posterior covariance

$$\Sigma_{i|i} \quad = \quad (I - K_i C)\Sigma_{i|i-1}.$$

D.2 Systems with Distance-Dependent Measurement Models

In Section 6.1, an agent model is introduced that models the effect of distance between the robot and the agent on the quality of the measurement obtained. The resulting augmented system has a measurement equation of the form:

$$y_i \quad = \quad Cx_i + H\nu_i + \sum_{l=1}^{n_\xi} \overline{H}^{(l)} \xi_i^{(l)} + \sum_{l=1}^{n_\xi} G^{(l)} x_i \xi_i^{(l)}.$$

The last two terms contain the same random variable, and this needs to be accounted for during the estimation process. Using the same approach as in Section D.1, the only difference is in calculating the covariance of the innovation, $\tilde{y}_i \triangleq y_i - \hat{y}_{i|i-1}$, where $\hat{y}_{i|i-1}$ is as before, and

$$\begin{aligned}
\Gamma_{i|i-1} \quad &= \quad C\Sigma_{i|i-1}C^T + HVH^T + \sum_{l=1}^{n_\xi} \sigma_\xi^2 G^{(l)} \left(\Sigma_{i|i-1} + \hat{x}_{i|i-1}\hat{x}_{i|i-1}^T \right) G^{(l)T} + \\
&\quad \sum_{l=1}^{n_\xi} \sigma_\xi^2 \overline{H}^{(l)} \overline{H}^{(l)T} + \sum_{l=1}^{n_\xi} \sigma_\xi^2 \left(\overline{H}^{(l)} \hat{x}_{i|i-1}^T G^{(l)T} + G^{(l)} \hat{x}_{i|i-1} \overline{H}^{(l)T} \right).
\end{aligned}$$

The rest of the filter is the same as in Section D.1.

Bibliography

[1] Y. Bar-Shalom. Stochastic dynamic programming: Caution and probing. *IEEE Transactions on Automatic Control*, 26(5):1184–1195, October 1981.

[2] Y. Bar-Shalom and E. Tse. Dual effect, certainty equivalence, and separation in stochasticcontrol. *IEEE Transactions on Automatic Control*, 19(5):494–500, October 1974.

[3] Y. Bar-Shalom and E. Tse. Caution, probing, and the value of information in the control of uncertain systems. *Annals of Economic and Social Measurement*, 5(3):60–74, 1976.

[4] A. Bemporad. Reducing conservativeness in predictive control of constrained systems with disturbances. In *37th IEEE Conference on Decision and Control*, volume 2, pages 1384–1389, December 1998.

[5] M. Bennewitz, W. Burgard, G. Cielniak, and S. Thrun. Learning motion patterns of people for compliant robot motion. *International Journal of Robotics Research*, 24:31–48, 2005.

[6] D. P. Bertsekas. *Dynamic Programming and Optimal Control*, volume 1. Athena Scientific, 3rd edition, 2005.

[7] D. P. Bertsekas. *Dynamic Programming and Optimal Control*, volume 2. Athena Scientific, 3rd edition, 2005.

[8] L. Blackmore. A probabilistic particle control approach to optimal, robust predictive control. Technical report, Massachusetts Institute of Technology, 2006.

[9] L. Blackmore. Robust path planning and feedback design under stochastic uncertainty. In *AIAA Guidance, Navigation and Control Conference*, number AIAA-2008-6304, August 2008.

[10] L. Blackmore, H. Li, and B. Williams. A probabilistic approach to optimal robust path planning with obstacles. In *American Control Conference*, June 2006.

[11] L. Blackmore and M. Ono. Convex chance constrained predictive control without sampling. In *AIAA Guidance, Navigation and Control Conference*, 2009.

[12] L. Blackmore and B. C. Williams. Optimal, robust predictive control of nonlinear systems under probabilistic uncertainty using particles. In *American Control Conference*, pages 1759–1761, July 2007.

[13] O. Cappe, E. Moulines, and T. Ryden. *Inference in Hidden Markov Models.* Springer, 2005.

[14] J.M. Carson. *Robust model predictive control with a reactive safety mode.* PhD thesis, California Institute of Technology, 2008.

[15] D. A. Castanon, B. C. Levy, and A. S. Willsky. Algorithms for the incorporation of predictive information in surveillance theory. *International Journal of Systems Science*, 16(3):367–382, 1985.

[16] A. Censi, D. Calisi, A. De Luca, and G. Oriolo. A bayesian framework for optimal motion planning with uncertainty. In *IEEE International Conference on Robotics and Automation*, pages 1798–1805, May 2008.

[17] H. Choset, K. M. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L. E. Kavraki, and S. Thrun. *Principles of Robot Motion.* MIT Press, 2007.

[18] A. Clodic, V. Montreuil, R. Alami, and R. Chatila. A decisional framework for autonomous robots interacting with humans. In *IEEE International Workshop on Robot and Human Interactive Communication*, pages 543–548, August 2005.

[19] T. M. Cover and J. A. Thomas. *Elements of Information Theory.* Wiley, 2nd edition, 2006.

[20] DARPA Urban Challenge. www.darpa.mil/grandchallenge, November 2007.

[21] N.M. Filatov and H. Unbehauen. Survey of adaptive dual control methods. *IEEE Proceedings - Control Theory and Applications*, 147(1):118–128, 2000.

[22] P. Fiorini and Z. Shiller. Motion planning in dynamic environments using velocity obstacles. *International Journal of Robotics Research*, 17(7):760–772, July 1998.

[23] A.F. Foka and P.E. Trahanias. Predictive control of robot velocity to avoid obstacles in dynamic environments. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 1, pages 370–375, October 2003.

[24] D. Fox, W. Burgard, and S. Thrun. The dynamic window approach to collision avoidance. *IEEE Robotics and Automation Magazine*, 4(1):23–33, 1997.

[25] B. Friedland. *Control System Design: An Introduction to State-Space Methods.* Dover Publications, Inc., 2005.

[26] C. Fulgenzi, A. Spalanzani, and C. Laugier. Dynamic obstacle avoidance in uncertain environment combining pvos and occupancy grid. In *IEEE International Conference on Robotics and Automation*, April 2007.

[27] J.P. Gonzalez and A.T. Stentz. Planning with uncertainty in position using high-resolution maps. *IEEE International Conference on Robotics and Automation*, pages 1015–1022, April 2007.

[28] H. Helble and S. Cameron. 3-d path planning and target trajectory prediction for the oxford aerial tracking system. In *IEEE International Conference on Robotics and Automation*, pages 1042–1048, April 2007.

[29] R. C. Hill, W. E. Griffiths, and G. C. Lim. *Principles of Econometrics.* John Wiley & Sons, Inc., 3rd edition, 2008.

[30] B. Kluge and E. Prassler. Reflective navigation: individual behaviors and group behaviors. In *IEEE International Conference on Robotics and Automation*, volume 4, pages 4172–4177, May 2004.

[31] B.C. Kuo. *Automatic Control Systems.* John Wiley and Sons, Inc., 7th edition, 1995.

[32] Y. Kuwata, T. Schouwenaars, A. Richards, and J. How. Robust constrained receding horizon control for trajectory planning. In *AIAA Guidance, Navigation, and Control Conference*, San Francisco, California, August 2005.

[33] F. Large, D. Vasquez, T. Fraichard, and C. Laugier. Avoiding cars and pedestrians using velocity obstacles and motion prediction. In *IEEE Intelligent Vehicles Symposium*, pages 375–379, June 2004.

[34] J.-C. Latombe. *Robot Motion Planning.* Kluwer Academic Press, 1991.

[35] S. M. LaValle. *Planning Algorithms.* Cambridge University Press, 2006.

[36] D. Li, F. Qian, and P. Fu. Variance minimization approach for a class of dual control problems. *Automatic Control*, 47(12):2010–2020, December 2002.

[37] D. Li, F. Qian, and P. Fu. Optimal nominal dual control for discrete-time linear-quadratic gaussian problems with unknown parameters. *Automatica*, 44(1):119–127, 2008.

[38] P. Li, M. Wendt, and G. Wozny. A probabilistically constrained model predictive next term controller. *Automatica*, 38(7):1171–1176, July 2002.

[39] B. Lindoff, J. Holst, and B. Wittenmark. Analysis of approximations of dual control. *International Journal of Adaptive Control and Signal Processing*, 13(7):593–620, 1999.

[40] M.S. Lobo and S. Boyd. Policies for simultaneous estimation and optimization. In *American Control Conference*, volume 2, pages 958–964, June 1999.

[41] T. Lozano-Perez, M. Mason, and R. H. Taylor. Automatic synthesis of fine-motion strategies for robots. *International Journal of Robotics Research*, 3(1), 1984.

[42] D.Q. Mayne, J.B. Rawlings, C.V. Rao, and P.O.M. Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, 36:789–814(26), June 2000.

[43] M. Morari and J. H. Lee. Model predictive control: past, present and future. *Computers & Chemical Engineering*, 23:667–682, 1999.

[44] M. Ono and B.C. Williams. Efficient motion planning algorithm for stochastic dynamic systems with constraints on probability of failure. MIT CSAIL TR-2008-013, Massachusetts Institute of Technology, March 2008.

[45] R. Pepy and A. Lambert. Safe path planning in an uncertain-configuration space using rrt. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5376–5381, October 2006.

[46] K. B. Petersen and M. S. Pedersen. Matrix cookbook. www.matrixcookbook.com, November 2008.

[47] N. Roy, G. Gordon, and S. Thrun. Planning under uncertainty for reliable health care robotics. In *International Conference on Field and Service Robotics*, 2003.

[48] A. T. Schwarm and M. Nikolaou. Chance-constrained model predictive control. *AIChE Journal*, 45(8):1743–1752, 1999.

[49] A. Simpkins, R. de Callafon, and E. Todorov. Optimal trade-off between exploration and exploitation. In *American Control Conference*, pages 33–38, June 2008.

[50] J.-J. E. Slotine and W. Li. *Applied Nonlinear Control.* Prentice Hall, 1991.

[51] S. Tadokoro, M. Hayashi, Y. Manabe, Y. Nakami, and T. Takamori. On motion planning of mobile robots which coexist and cooperate with humans. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2:2518, 1995.

[52] S. Thompson, T. Horiuchi, and S. Kagami. A probabilistic model of human motion and navigation intent for mobile robot path planning. In *International Conference on Autonomous Robots and Agents*, pages 663–668, February 2009.

[53] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics.* MIT Press, 2005.

[54] E. Todorov. Stochastic optimal control and estimation methods adapted to the noise characteristics of the sensorimotor system. *Neural Computation*, 17(5):1084–1108, 2005.

[55] J. van den Berg, Ming Lin, and D. Manocha. Reciprocal velocity obstacles for real-time multi-agent navigation. In *IEEE International Conference on Robotics and Automation*, pages 1928–1935, May 2008.

[56] D.H. van Hessem. *Stochastic inequality constrained closed-loop model predictive control with application to chemical process operation*. PhD thesis, Delft University, 2004.

[57] D.H. van Hessem and O.H. Bosgra. A conic reformulation of model predictive control including bounded and stochastic disturbances under state and input constraints. *IEEE Conference on Decision and Control*, 4:4643–4648, December 2002.

[58] D.H. van Hessem and O.H. Bosgra. A full solution to the constrained stochastic closed-loop mpc problem via state and innovations feedback and its receding horizon implementation. *IEEE Conference on Decision and Control*, 1:929–934, December 2003.

[59] D.H. van Hessem and O.H. Bosgra. Towards a separation principle in closed-loop predictive control. *American Control Conference*, 5:4299–4304, June 2003.

[60] D.H. van Hessem and O.H. Bosgra. Closed-loop stochastic model predictive control in a receding horizon implementation on a continuous polymerization reactor example. *American Control Conference*, 1:914–919, June 2004.

[61] J. Yan and R. R. Bitmead. Incorporating state estimation into model predictive control and its application to network traffic control. *Automatica*, 41(4):595–604, April 2005.

[62] Q. Zhu. Hidden markov model for dynamic obstacle avoidance of mobile robot navigation. *IEEE Transactions on Robotics and Automation*, 7(3):390–397, June 1991.