2010 IEEE International Conference on Robotics and Automation
Anchorage Convention District
May 3-8, 2010, Anchorage, Alaska, USA

Robotic Motion Planning in Dynamic, Cluttered, Uncertain Environments

Noel E. Du Toit (ndutoit@caltech.edu) and Joel W. Burdick (jwb@robotics.caltech.edu)

Abstract— **This paper presents a strategy for planning robot motions in dynamic, cluttered, and uncertain environments. Successful and efficient operation in such environments requires reasoning about the future system evolution and the uncertainty associated with obstacles and moving agents in the environment. This paper presents a novel procedure to account for future information gathering (and the quality of that information) in the planning process. After first presenting a formal Dynamic Programming (DP) formulation, we present a Partially Closed-loop Receding Horizon Control algorithm whose approximation to the DP solution integrates prediction, estimation, and planning while also accounting for chance constraints that arise from the uncertain location of the robot and other moving agents. Simulation results in simple static and dynamic scenarios illustrate the benefit of the algorithm over classical approaches.**

I. INTRODUCTION

This paper is concerned with motion planning in *Dynamic*, *Cluttered*, and *Uncertain Environments* (DCUEs). In such environments, 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 move 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. In this paper we present a framework and some initial algorithms and simulation results that we hope will provide a foundation for future DCUE motion planners.

Robot motion planning in dynamic environments has recently received substantial attention due to the DARPA Urban Challenge [1] and growing interest in service and assistive robots (*e.g.*, [2], [3]). In urban environments, traffic rules influence 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.

N.E. Du Toit and J.W. Burdick are with the Dep. of Mechanical Engineering, California Institute of Technology, Pasadena, CA 91125, USA

To date, various proposed motion planning algorithms and frameworks handle only specific subsets of the DCUE problem. Of course, classical motion planning algorithms [4] 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) [4]. When the future locations are unknown, the planning problem is either solved locally [5]–[7] (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 [4], [7], [8]. One attempt to extend the local planner to uncertain environments is the Probabilistic Velocity Obstacle approach [9].

Planning algorithms for stochastic systems have been developed, but largely for static environments. Two types of stochastic systems are distinguished: non-deterministic (where the uncertainties lie in a bounded set, *e.g.*, [10], [11]), and probabilistic [4]. Probabilistic systems are the main focus of this work. The first stochastic planning approach was pre-image back-chaining [12]. Discrete search strategies have also been extended to probabilistic systems where the problem is solved in an extended state space (pose $\times$ covariance) (*e.g.*, [13]–[15]). Alternatively, the problem can be posed as a stochastic dynamic program (SDP) [16]. When the system's dynamic equation is time-invariant and the stage cost is constant, the stochastic problem can be solved using POMDP (Partially Observable Markov Decision Process) methods [16], [17]. Otherwise, the SDP problem can be approximately solved with a Rollout algorithm (a limited lookahead policy) or a restricted information approach [16]. Finally, the Receding Horizon Control (RHC) framework for deterministic systems has been extended to a stochastic RHC formulation in the particular case of robot localization uncertainty (*e.g.*, [18]–[20]).

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.*, [21], [22]). Longer term predictions can be constrained by learning the dynamic agents' preferred paths and using these paths to predict future states (e.g., [23], [24]), or by inferring structure in the environment which can inform the prediction process (*e.g.*, [25], [26]).

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 paper represents the first formal effort to incorporate the effect of anticipated future measurements in the motion planning process. As shown by example, the proper inclusion of these effects can improve robot performance in the presence of uncertain agent behavior. We also introduce a novel analysis of the chance constraints that model the probability of collisions between moving objects whose positions are uncertain (Section V-B). Because the exact DCUE solution is rather intractable, we introduce a stochastic RHC framework and present the Partially Closed-loop RHC solution approach in Section IV. This approach is motivated from a dynamic programming point of view in Section III. Results for a robot navigating in a static (Section VI-A) and a dynamic environment (Section VI-B) are presented to illustrate some of the characteristics of this method. For implementation simplicity, examples of linear systems with Gaussian noise terms are considered, even though the approach is more general.

II. PROBLEM STATEMENT

This section develops an abstract constrained stochastic optimal control problem statement which encompasses all of the aspects of the DCUE problem developed in the remainder of the paper. The formulation presented here is standard. Our subsequent analysis of this problem represents our technical contribution.

Let the system state, which could represent the robot state as well as the states of other moving agents, be a member of a *state space* $\mathbb{X}$: $x_i \in \mathbb{X} \subseteq \mathbb{R}^{n_x}$. The control $u_i(x_i)$ (i.e., the input commands to the robot) is an element of the *action space* $\mathbb{U}(x_i)$: $u_i(x_i) \in \mathbb{U}(x_i) \subseteq \mathbb{R}^{n_u}$. The disturbance, $\omega_i(x_i, u_i) \in \mathbb{W}(x_i, u_i) \subseteq \mathbb{R}^{n_\omega}$, models the uncertainty in the robot's and agents' governing dynamic models and is described by the conditional distribution $\omega_i(x_i, u_i) \sim p(\omega_i|x_i, u_i)$. The disturbance is assumed to be independent of previous disturbances, $\omega_{0:i-1}$. The system state is governed by the following discrete-time dynamic system:

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

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

In general, the system state is not directly measurable, but must be inferred from noisy measurements. The measurement, y_i, is an element of the *measurement space* $\mathbb{Y}(x_i)$: $y_i \in \mathbb{Y}(x_i) \subseteq \mathbb{R}^{n_y}$. The measurement is corrupted by measurement noise, $\nu_i(x_i) \in \mathbb{V}(x_i) \subseteq \mathbb{R}^{n_\nu}$, which is described by the conditional distribution $\nu_i(x_i) \sim p(\nu_i|x_i)$. The C^2 sensor mapping, $h : \mathbb{X} \times \mathbb{V} \to \mathbb{Y}$ maps every state into a measurement:

$$y_i = h(x_i, \nu_i). \tag{2}$$

The *information state*[1], I_i, captures all the information available to controller at some stage i, and is discussed in Section III. Consider a class of N-stage *control policies* from the current stage, k, $\Pi = \{\pi_k(I_k), \ldots, \pi_{N-1}(I_{N-1})\}$, that maps the I-state to a control, $u_i = \pi_i(I_i)$. A stage-additive cost function that captures the essence of the robot task is assumed:

$$L(x_{k:N}, \Pi, \omega_{k:N-1}) = l_N(x_N) + \sum_{i=k}^{N-1} l_i(x_i, \pi_i(I_i), \omega_i). \tag{3}$$

The expected cost incurred by the robot as it carries out its control policy is thus:

$$C_\Pi = E\left[l_N(x_N) + \sum_{i=k}^{N-1} l_i(x_i, \pi_i(I_i), \omega_i)\right] \tag{4}$$

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

Additionally, the controls may be constrained by nonlinear inequality functions $c(u_{k:N-1}) \leq 0$ and the system states are chance constrained: $P(x \notin \mathbb{X}_{free}) \leq \delta$. This latter constraint will model obstacle avoidance under uncertainty.

The optimal policy, $\Pi^{(*)} = \{\pi_k^{(*)}(I_k), \ldots, \pi_{N-1}^{(*)}(I_{N-1})\}$, minimizes the expected cost over the set of admissible policies, $\widetilde{\Pi}$, while satisfying the constraints:

$$\begin{aligned}
\Pi^{(*)} \quad &= \quad \arg\min_{\Pi \in \widetilde{\Pi}} C_\Pi \\
s.t. \quad & \quad x_{i+1} = f\left(x_i, u_i, \omega_i\right) \\
& \quad y_i = h(x_i, \nu_i) \\
& \quad c(u_{k:N-1}) \leq 0 \\
& \quad P(x \notin \mathbb{X}_{free}) \leq \delta.
\end{aligned}$$

This problem is very difficult to solve in general. To gain insight, we will first consider the unconstrained version of this problem, which can be reformulated as a stochastic dynamic programming problem. The constraints will be incorporated in a later section.

III. STOCHASTIC DYNAMIC PROGRAMMING

The problem of Section II, without the state and control constraints, can be solved using the stochastic dynamic programming (SDP) formulation. The SDP approach constructs the optimal feedback control: for every reachable future state, the optimal control action is defined. In the *perfect state information* problem, the system state is known or measurable. The SDP problem is an *imperfect state information* problem since the state is not directly measurable. Two features of the SDP approach are of interest here: converting the problem into the belief space, and approximating this problem by restricting the information available to the planner.

A. Converting to Belief Space

I-states summarize the information available to the planner or controller. Two I-states of interest here are the *history I-state*, and the *belief state*. Let the *history of measurements* at the i^{th} stage to be $y_{0:i} \triangleq \{y_0, y_1, \ldots, y_i\}$ and the *history of controls* be $u_{0:i} \triangleq \{u_0, u_1, \ldots, u_i\}$. The *history information state*, η_i, is defined as:

$$\eta_i = \{\eta_0, u_{0:i-1}, y_{0:i}\}. \tag{5}$$

[1]For brevity, information state and space will be abbreviated as I-state and I-space, respectively.

where η_0 is the initial history I-state. I-states evolve according to a dynamic transition functions. The I-state at the next stage is unpredictable because the next measurement y_{i+1} is unknown. *This measurement, y_{i+1}, plays the role of a process noise in the history I-space* [16].

The history I-state summarizes all of the information that can be known about the stochastic system up to the current time. However, because the history I-state can be unwieldy in practice, it is often useful to work with the simpler belief state. The belief state is derived from the history I-state with a sufficient information mapping [4]. We assume that a Markov probabilistic model governs the system's evolution: the current system state is only a function of the state at the previous time step and the controls and disturbances experienced by the system at the lat time step. Thus, the current state is the best predictor of the future states [17]. The belief state is defined as:

$$\zeta_i \triangleq p(x_i|\eta_i). \tag{6}$$

and the belief state transition function, $f_\zeta(\eta_{i-1}, u_{i-1}, y_i)$ is obtained from Bayes' rule [16], [17].

To proceed with the analysis of this problem, the cost function, (3), must be converted into an equivalent cost function in terms of the belief states. The cost at each stage is the expected value of $l_i(x_i, \pi_i, \omega_i)$, conditioned on all the information available at that stage:

$$\bar{l}_i(\zeta_i, \pi_i, y_{i+1}) \quad = \quad E\left[l_i(x_i, \pi_i, \omega_i)|\eta_i\right] = \bar{l}_i(\zeta_i, \pi_i).$$

The expectation is taken with respect to $p(x_i, \omega_i|\eta_i)$. The terminal cost is:

$$\bar{l}_N(\zeta_N) = E\left[l_N(x_N)|\eta_N\right].$$

The expectation is taken with respect to $p(x_N|\eta_N)$. The cost function becomes:

$$\bar{L}(\zeta_{k:N}, \pi_{k:N-1}, y_{1:N}) = \bar{l}_N(\zeta_N) + \sum_{i=k}^{N-1} \bar{l}_i(\zeta_i, \pi_i). \tag{7}$$

The expected cost in the belief space is written in terms of the 'disturbances' (measurements):

$$\bar{C}(\zeta_k) = E_{y_{1:N}}\left[\bar{l}_N(\zeta_N) + \sum_{i=k}^{N-1} \bar{l}_i(\zeta_i, \pi_i)\right]. \tag{8}$$

The dynamic programming algorithm is used to solve this optimization problem by using the backwards recursion [16]:

$$J_N(\zeta_N) = \bar{l}_N(\zeta_N) \tag{9}$$

$$\begin{aligned} J_i(\zeta_i) = \min_{\pi_i}\ & \bar{l}_i(\zeta_i, \pi_i) \\ & + E_{y_{i+1}}\left[J_{i+1}(f_\zeta(\zeta_i, \pi_i, y_{i+1}))|\eta_i\right]. \end{aligned} \tag{10}$$

The SDP algorithm is a feedback algorithm on belief space: *the optimal control action is defined for every possible combination of control and measurement sequences*. But the set of possible measurements that can be obtained is infinite. Only a few SDP problems allow for closed form solution (*e.g.*, linear systems with quadratic cost, Gaussian noise terms, and no constraints) [16]. Instead, one must resort to approximate solutions.

B. Approximations to the SDP

The SDP problem is approximated by either (i) recursively solving a simplified problem for a control sequence instead of a control policy (*e.g.*, the Certainty Equivalent Control strategy and the Open-loop Control strategy), or (ii) solving for a control policy over a limited horizon and then approximating the cost-to-go function beyond this horizon (*e.g.*, Limited Lookahead Policies) [16]. The Open-loop Control strategy is of interest here. This control strategy uses a restricted information set[2] [16] when approximately solving the problem: measurements beyond the current stage are ignored. The restricted information set is: $\bar{\eta}_i = (y_1, \ldots, y_k, u_0, \ldots, u_{i-1})$, $i \geq k$. The belief states associated with this restricted information set, $\bar{\zeta}_i = p(x_i|\bar{\eta}_i)$ are the open-loop predicted distributions, and are used in the SDP algorithm. The effect of using the restricted information set is that these future belief states are completely defined for a given control sequence. Thus, the problem becomes deterministic and (10) becomes:

$$J_i(\bar{\zeta}_i) = \min_{u_i}\ \bar{l}_i(\bar{\zeta}_i, u_i) + J_{i+1}(f_\zeta(\bar{\zeta}_i, u_i, y_{i+1})). \tag{11}$$

This result can now be used to formulate the stochastic receding horizon control problem to incorporate the constraints.

IV. PARTIALLY CLOSED-LOOP RECEDING HORIZON CONTROL

While the best approach to extend the RHC formulation to stochastic systems is still up for debate, it is convenient to convert the problem into the belief space and then use a restricted information set to make the problem deterministic in terms of the belief states. A sequence of control actions over a finite horizon, $M \leq N$, is obtained instead of a feedback control law for all reachable belief states. The feedback mechanism is moved outside of the planning loop: the control sequence is executed over a short interval (typically one stage), and the planning process is repeated with new measurements obtained after executing the control sequence. by recursively solving the problem, new information is incorporated:

$$\begin{aligned} \min_{u_{k:M}} \quad & \bar{l}_M(\bar{\zeta}_M) + \sum_{i=k}^{M-1} \bar{l}_i(\bar{\zeta}_i, u_i) \\ s.t. \quad & \zeta_{i+1} = f_\zeta(\eta_i, u_i, y_{i+1}) \\ & P(x_{k:M} \notin X_{free}) \leq \delta \\ & c(u_{k:M-1}) \leq 0. \end{aligned} \tag{12}$$

Planning and control approaches proposed to date have used a restricted information set corresponding to the Open-loop Control approximation to the SDP problem: measurements beyond the current stage are ignored during the planning process (since the future measurements are obviously

[2]A restricted information set is a subset of the history I-state or belief state that is used constuct a more tractable approximating solution.

unknown at present). This restriction results in the Open-loop Receding Horizon Control (OLRHC) approach [18], [20], [27]–[29]. Since no future measurements are considered, the resulting belief states are the open-loop predicted distributions for the robot and agents. Yan and Bitmead [18] noted that the covariances of the future belief states grow when using the open-loop RHC formulation. They introduced a 'closed-loop covariance' where the covariance for all future belief states are fixed at the one-step ahead open-loop prediction value. This crudely accounts for the anticipated future information during the planning problem. One of our main technical contributions is the formal inclusion of the future anticipated measurements into the stochastic RHC framework.

In an attempt to better account for anticipated future information, we define an alternative restricted information set which is the basis for the *Partially Closed-loop Receding Horizon* (PCLRHC) control approach that we now introduce. The key observation is that the future measurements have two effects on the future belief states (for linear systems with Gaussian noise terms): the *value of the measurement changes only the center of the belief state (not the spread)* and *the measurement reduces the uncertainty in the belief state.* The PCLRHC 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 future belief states are predictable for a given control sequence. The restricted information set used is $\tilde{\eta}_i = (y_1, \ldots, y_k, \tilde{y}_{k+1}, \ldots, \tilde{y}_i, u_k, \ldots, u_{i-1})$, where $\tilde{y}_j = E[y_j|\tilde{\eta}_{j-1}]$ is the most probable measurement. The belief state associated with this restricted information set is $\tilde{\zeta}_i = p(x_i|\tilde{\eta}_i) = p(x_i|u_{0:i-1}, y_{1:k}, \tilde{y}_{k+1:i})$. The problem of (12) is solved in terms of this new belief state.

By assuming the most probable measurement, *the fact that a future measurement is going to occur is used to reduce the uncertainty of the predicted belief state, but the center of the belief state is not updated.* It can be shown, using the relative information entropy, that this assumption does not introduce artificial information into the problem (at least for a linear system with Gaussian noise terms) [30]. Additionally, the most-likely measurement assumption is the least informative assumption about the value of the measurement possible: any other value will introduce information and bias is the problem.

As we shall see in the ensuing examples, by taking into account the fact that the robot will take future measurements, the robot can make more aggressive plans in the presence of chance constraints that define collisions between uncertain moving objects.

V. 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 *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 $\mathbb{X}_{free} = \{x : c(x) \leq 0\}$. In the stochastic setting, the constraints are specified as limits on the probability of constraint violation.

From the SDP algorithm, when the control at stage i is selected, the only unknown quantity is the next measurement, y_{i+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 $P(x_{i+1} \notin X_{free}|\eta_i) \leq \delta_i$.

Two types of chance constraints are considered: 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). For the former, the only unknown parameter is the state, and standard results exist. For the latter, the robot and object states are unknown. We present a novel analysis of this problem below.

A. Linear Chance Constraints with Gaussian Variables

For Gaussian state variables, linear chance constraints are simple to evaluate. Let the linear chance constraint be of the form $P(a^T x_{i+1} > b|\eta_i) \leq \delta$. Let $\hat{x}_{i+1|i} \triangleq E[x_{i+1}|\eta_i]$ and $\Sigma_{i+1|i} \triangleq E[(x_{i+1} - \hat{x}_{i+1|i})(x_{i+1} - \hat{x}_{i+1|i})^T|\eta_i]$.

Lemma 1: The chance constraint $P(a^T x_{i+1} > b|\eta_i) \leq \delta$ is satisfied if

$$a^T \hat{x}_{i+1|i} + F(\delta) \times \sqrt{a^T \Sigma_{i+1|i} a} \leq b \tag{13}$$

where $F(\delta)$ is the inverse of the cummulative distribution function for a standard Gaussian variable.

Proof: Refer to [19], [31]. ■

To enforce the chance constraint with $\delta = 0.01$, the value for the $F(\delta) = 2.3263$.

B. Collision Chance Constraints

Probabilistic obstacle avoidance can be formulated as a chance constraint. Blackmore [32] assumed known, static, non-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 1) for each obstacle. Obstacle uncertainty is ignored in this framework. 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.

C. Probability of Collision

For simplicity, assume a disk robot (radius ε) and a point obstacle[3]. Let x^R be the robot position state, and x^A be the agent position state. 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:

$$P(C) = \int_{x^R} \int_{x^A} I_C(x^A, x^R) p(x^R, x^A) dx^R dx^A \tag{14}$$

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

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. (14) can be written as:

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

This integral function is difficult to evaluate for general robot and obstacle geometries. To gain some intuition, assume that the robot radius is small. 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 is therefore:

$$P(C) \approx V_{\mathcal{B}} \times \int_{x^R} p(x^A = x^R|x^R) p(x^R) dx^R. \qquad (15)$$

D. Collision Chance Constraints for Systems with Gaussian Variables

If it is now assumed that the robot and the obstacle (objects) position uncertainties can be described by independent Gaussian distributions, then the integral in Eq. (15) can be evaluated in closed form [30]. Let $x_R \sim N(x_R; \hat{x}_R, \Sigma_R)$ and $x_A \sim N(x_A; \hat{x}_A, \Sigma_A)$, then

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

where $\Sigma_C \triangleq \Sigma_R + \Sigma_A$ is the combined position covariance. The constraint $P(C) \leq \delta$ is converted into a constraint on the mean states of the robot and obstacle:

$$(\hat{x}^R - \hat{x}^A)^T \Sigma_C^{-1} (\hat{x}^R - \hat{x}^A) \geq \kappa \qquad (16)$$

where κ is a function of the level of certainty, the size of the robot (and obstacle), and the combined covariance. A lookup table was created to capture the dependence of κ on the Σ_C. 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, $\lambda \triangleq \min(eig(\Sigma_C))$[4]. The true probability of collision was estimated for different covariances at different locations using a Monte Carlo simulation and κ was chosen to enforce the constraint. Refer to Figure 1. The resulting values for κ are documented in Table I.

[4]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.

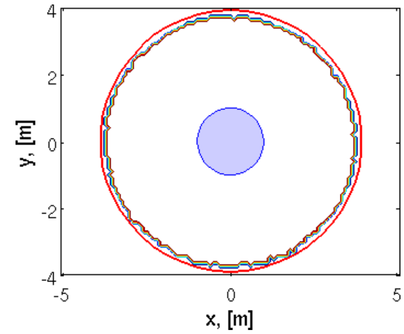

Fig. 1. Lookup table generation: approximate the probability of collision between objects of radius $0.5\ m$ using a Monte Carlo simulation for $\lambda = 1$. The robot location is fixed at the origin and the obstacle location is varied. Evaluate and plot the collision condition (contour plot) and find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

TABLE I

COLLISION CONSTRAINT PARAMETER, κ, FOR DIFFERENT VALUES OF λ WITH $\alpha = 0.99$ AND DISK 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

VI. RESULTS

The OLRHC (which is representative of current stochastic RHC practice) and the PCLRHC approaches are compared in static and dynamic scenarios. Even these trivial examples show how anticipated future measurements and chance constraints (which are fundamental to the DCUE problem) can affect the motion planning outcome. Linear models for the robot and agents (objects) with Gaussian noise terms were used for simplicity, but the approach presented in this paper is more generally applicable.

In the examples, consider planar disk objects (robot and agents) of radius 0.5 m whose dynamics are governed by:

$$x_i = Ax_{i-1} + Bu_{i-1} + F\omega_{i-1} \qquad (17)$$

where

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

where $\Delta t = 0.5$ s. The system state consists of the planar positions and velocities. The process noise is independent and normally distributed with zero mean and $\Sigma_\omega = 0.01 \times I_2$ (I_2 is the identity matrix in 2 dimensions).

A linear position measurement model is assumed with normally distributed measurement noise (zero mean and $\Sigma_\nu = 0.01 \times I_2$). The measurement equation is:

$$y_i = Cx_i + H\nu_i \qquad (18)$$

where

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

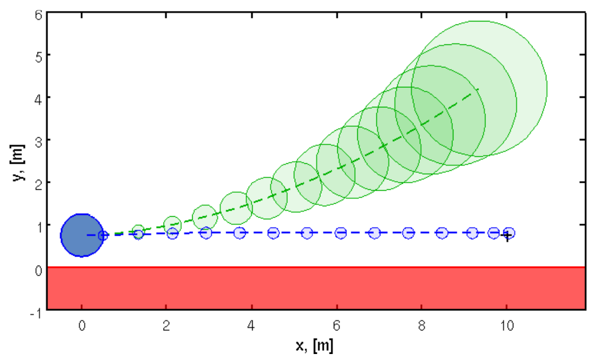

Fig. 2. Planned trajectories with the OLRHC (green, dashed) and PCLRHC (blue, dashed) approaches, with the associated 1-σ uncertainty ellipses. The static obstacle is plotted in red.

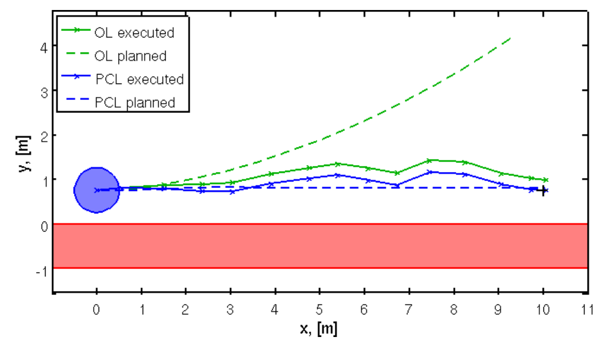

Fig. 3. Comparison of the planned and executed paths produced by the OLRHC and PCLRHC approaches

A. Static Environment Example

The robot must navigate in a static environment with a single rectangular static obstacle (known geometry and position). 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 $\hat{x}_{k|k} = [0\ 0.75\ 1\ 0]^T$ and $\Sigma_{k|k} = 0.01 \times I_4$. 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:

$$L(x_0) = (x_M - x_G)^T Q_M (x_M - x_G) + \sum_{i=0}^{M-1} \left\{ (x_i - x_G)^T Q_i (x_i - x_G) + u_i^T R_i u_i \right\} \quad (19)$$

where $Q_M = \mathrm{diag}(10, 10, 0, 0)$, $Q_i = \mathrm{diag}(1, 1, 0, 0)$, and $R_i = \mathrm{diag}(1, 1, 0, 0)$, for all $i = 0, \ldots, M-1$.

In order to fully test the framework, each control input is constrained to be less than unity at each stage: $u_i^{(1)} \leq 1$ and $u_i^{(2)} \leq 1$, $i = 0, \ldots, M-1$. The collision 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 $[-2, 2]$:

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

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

The *planned* paths at the initial stage for the OLRHC (green, dashed) and PCLRHC (blue, dashed) approaches are shown in Figure 2. The static obstacle is illustrated in red. The 1-σ error ellipses along the trajectories represent the uncertainty associated with the robot position. 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 planned OL path diverges from the straight line due to the chance constraints that are being enforced. Since future measurements are not considered during this plan, the robot's uncertainty grows into the future, and therefore the robot must 'back-off' from the position constraint. The obtained solution is very conservative, 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. The initially planned solution from the PCLRHC approach drives the robot straight to the goal.

Fig. 3 shows that the *executed* paths are similar due to the Receding Horizon Control "outer loop feedback" mechanism (the problem is re-solved as new measurements are taken at each planning cycle). However, the planned and executed trajectories for the OLRHC approach differ substantially, as the planner relies almost exclusively on the outer-loop feedback mechanism to execute a reasonable trajectory. 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.

B. Dynamic Environment: Oncoming Agents

Consider again the same robotic system of Section VI-A, with the same dynamic and measurement models, and the same cost function. Two dynamic obstacles move towards the robot. We assume that the agent models are independent of the robot state (i.e., the agent doesn't "react" to the presence of the robot). However, the agent states enter the problem through the collision chance constraints. The goal state is $x_G = [10\ 0\ 0\ 0]$. The robot initial state is $x_0 \sim \mathcal{N}(\hat{x}^R_{0|0}, \Sigma^R_{0|0})$, where $\hat{x}^R_{k|k} = [0\ 0\ 1\ 0]^T$ and $\Sigma^R_{k|k} = 0.01 \times I_4$. The first agent's initial state is $x_0^{A1} \sim \mathcal{N}(\hat{x}^{A1}_{0|0}, \Sigma^{A1}_{0|0})$, where $\hat{x}^{A1}_{k|k} = [12\ 2\ 1\ 0]^T$ and $\Sigma^{A1}_{k|k} = 0.01 \times I_4$. The second agent's initial state is $x_0^{A2} \sim \mathcal{N}(\hat{x}^{A2}_{0|0}, \Sigma^{A2}_{0|0})$, where $\hat{x}^{A2}_{k|k} = [12\ -2\ 1\ 0]^T$ and $\Sigma^{A2}_{k|k} = 0.01 \times I_4$. Finally, the collision chance constraints are imposed for each agent and at each stage with $\delta_c = 0.01$.

Fig. 4 shows the first stage *planned* and predicted trajectories for the OLRHC (green, dashed) and PCLRHC (blue, dashed) approaches and the agents (red, dashed). The 1-σ uncertainty ellipses are plotted along these trajectories. In this example, the PCLRHC approach obtains a significantly improved *planned and executed* trajectory. The OLRHC approach is unable to plan between the agents, and must instead move around both agents. This is due to the growth in

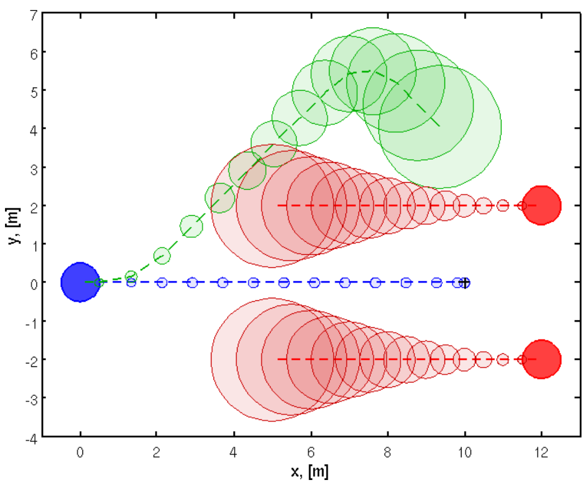

Fig. 4. Planned trajectories with the OLRHC (green, dashed) and PCLRHC (blue, dashed) approaches and the predicted agent trajectories (red, dashed) with the 1-σ uncertainty ellipses

uncertainty. The PCLRHC approach is able to make progress directly towards the goal. It is clear that the PCLRHC approach is significantly better than the OLRHC approach. A Monte Carlo simulation is used to quantify this improvement for this specific scenario. To quantify this improvement more generally, a Monte-Carlo simulation is used in the following example.

C. Monte-Carlo Simulation: Crossing Agents

Two dynamic obstacles cross the space between the robot and the goal. The same models, cost function, and constraints are used as in Section VI-B. The goal state is $x_G = [12\ 0\ 0\ 0]$. The robot initial state is $x_0^R \sim \mathcal{N}(\hat{x}_{0|0}^R, \Sigma_{0|0}^A)$, where $\hat{x}_{0|0}^R$ 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 for each agent is imposed at each stage with $\delta_c = 0.01$.

TABLE II

MONTE-CARLO SIMULATION INITIAL CONDITION RANGES

	x	y	Heading
Robot	0	$[-2,\ 2]$	$[-22.5^\circ,\ 22.5^\circ]$
Agent 1	$[4,\ 8]$	6	$[-120^\circ,\ -75^\circ]$
Agent 2	$[4,\ 8]$	-6	$[75^\circ,\ 120^\circ]$

The simulation is repeated 200 times with randomized initial conditions. The ranges of the variable are given in Table II. The robot initial velocity is 1.2 m/s and the initial velocity is fixed at 1 m/s for both agents. The robot moves from left to right, agent 1 moves from north to south, and agent 2 from south to north (crossing).

The executed trajectories from the PCLRHC approach are compared to the OLRHC approach results. From the histograms of the executed path lengths for the OLRHC (Figure 5) and PCLRHC (Figure 6) approaches, the PCLRHC

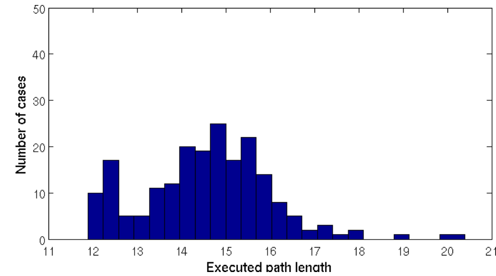

Fig. 5. Histogram of executed path lengths for the OLRHC approach. The first peak (centered around 12.5) corresponds to cases where the robot is able to move directly to the goal. The second peak (centered around 15) corresponds to the cases where the robot has to maneuver around the agents.

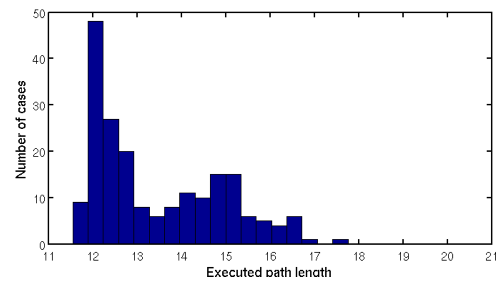

Fig. 6. Histogram of executed path lengths for the PCLRHC approach. The first peak (centered around 12.5) corresponds to cases where the robot is able to move directly to the goal. The second peak (centered around 15) corresponds to the cases where the robot has to maneuver around the agents.

approach is more often able to find direct paths to the goal (from the height of the peak centered around 12.5 m) than the OLRHC approach. A larger second peak (at 15 m) for the OLRHC approach indicates that the approach must react to the agents more often, resulting in longer paths. 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, with at least a 10% improvement in 37.5% and at least a 20% improvement in 17.5% of the cases.

VII. CONCLUSIONS AND FUTURE WORK

A complete strategy for solving motion planning problems in Dynamic, Cluttered, and Uncertain Environments (DCUEs) has been lacking. In this paper we took some initial steps toward building such a framework. These environments are characterized by uncertainty in the positions of the robot and moving agents, as well as uncertainty about the future trajectories of the agents. Because stochastic dynamic programming (SDP) does not readily incorporate collision constraints, and because exact SDP solutions are often intractable, this paper developed the Partially Closed-Loop Receding Horizon Control (PCLRHC) strategy for this problem. Our approach was motivated by the desire to account for anticipated future information in the planning process. In this way, we are better able to manage the growth of system uncertainty in the prediction component of the DCUE solution, as the anticipated future information reduces the uncertainty associated with future belief states.

Previous planning approaches were hampered by the growth in uncertainty associated with future belief states, leading to highly constrained and conservative plans. The planned and executed paths are much closer for the PCLRHC approach, indicating that the planner effectively uses the anticipated future information during the planning process. These plans are more aggressive because they take into account the fact that updates of the world's state will be available. Simulation results for a robot navigating in a static and dynamic environments highlighted the improvement in plan quality for the partially closed-loop approach, compared to the open-loop approach where all future information is ignored.

Collision constraints in the case of uncertain robot and agent position are naturally formulated in terms of chance constraints. We developed a new analysis of collision chance constraints. Previous results assumed perfect knowledge of the obstacle states and geometries.

The partially closed-loop approach effectively manages the uncertainty growth during belief state propagation. This potentially allows for more complicated agent behaviors to be modeled and included in the planning process. E.g., one could model agents as "cooperative," "neutral," or "combative." Open-loop approaches are unable to plan in these scenarios since the complicated agent behaviors result in more uncertainty, and the problem becomes even more constrained. Future work will focus on planning among agents with more complicated behaviors. For example, agents with multi-modal behavior models (such as different possible destinations), and agents with multiple possible models are of interest. Furthermore, agent models that are dependent on the robot states are of interest, since these models will allow the robot to actively gather information about the agents (*i.e.*, adjust the plan to improve the quality of the information).

REFERENCES

[1] DARPA Urban Challenge, "www.darpa.mil/grandchallenge," DARPA, November 2007.

[2] 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*, vol. 2, p. 2518, 1995.

[3] N. Roy, G. Gordon, and S. Thrun, "Planning under uncertainty for reliable health care robotics," in *International Conference on Field and Service Robotics*, 2003. [Online]. Available: citeseer.ist.psu.edu/roy03planning.html

[4] S. M. LaValle, *Planning Algorithms*. Cambridge University Press, 2006.

[5] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," *IEEE Robotics and Automation Magazine*, vol. 4, no. 1, pp. 23–33, 1997.

[6] P. Fiorini and Z. Shiller, "Motion planning in dynamic environments using velocity obstacles," *International Journal of Robotics Research*, vol. 17, no. 7, pp. 760–772, July 1998. [Online]. Available: citeseer.ist.psu.edu/fiorini98motion.html

[7] H. Choset, K. M. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L. E. Kavraki, and S. Thrun, *Principles of Robot Motion*. MIT Press, 2007.

[8] 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*, August 2005, pp. 543–548.

[9] 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. [Online]. Available: http://emotion.inrialpes.fr/bibemotion/2007/FSL07

[10] 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.

[11] J. Carson, "Robust model predictive control with a reactive safety mode," Ph.D. dissertation, California Institute of Technology, 2008.

[12] T. Lozano-Perez, M. Mason, and R. H. Taylor, "Automatic synthesis of fine-motion strategies for robots," *International Journal of Robotics Research*, vol. 3, no. 1, 1984.

[13] 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*, October 2006, pp. 5376–5381.

[14] J. Gonzalez and A. Stentz, "Planning with uncertainty in position using high-resolution maps," *IEEE International Conference on Robotics and Automation*, pp. 1015–1022, April 2007.

[15] 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*, May 2008, pp. 1798–1805.

[16] D. P. Bertsekas, *Dynamic Programming and Optimal Control*, 3rd ed. Athena Scientific, 2005, vol. 1.

[17] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. MIT Press, 2005.

[18] J. Yan and R. R. Bitmead, "Incorporating state estimation into model predictive control and its application to network traffic control," *Automatica*, vol. 41, no. 4, pp. 595–604, April 2005.

[19] D. van Hessem and O. Bosgra, "Closed-loop stochastic model predictive control in a receding horizon implementation on a continuous polymerization reactor example," *American Control Conference*, vol. 1, pp. 914–919, June 2004.

[20] L. Blackmore, "Robust path planning and feedback design under stochastic uncertainty," in *AIAA Guidance, Navigation and Control Conference*, no. AIAA-2008-6304, August 2008.

[21] F. Large, D. Vasquez, T. Fraichard, and C. Laugier, "Avoiding cars and pedestrians using velocity obstacles and motion prediction," in *IEEE Intelligent Vehicles Symposium*, June 2004, pp. 375–379.

[22] Q. Zhu, "Hidden markov model for dynamic obstacle avoidance of mobile robot navigation," *IEEE Transactions on Robotics and Automation*, vol. 7, no. 3, pp. 390–397, June 1991.

[23] 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*, April 2007, pp. 1042–1048.

[24] M. Bennewitz, W. Burgard, G. Cielniak, and S. Thrun, "Learning motion patterns of people for compliant robot motion," *International Journal of Robotics Research*, vol. 24, pp. 31–48, 2005.

[25] 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*, February 2009, pp. 663–668.

[26] A. Foka and P. Trahanias, "Predictive control of robot velocity to avoid obstacles in dynamic environments," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 1, October 2003, pp. 370–375.

[27] A. T. Schwarm and M. Nikolaou, "Chance-constrained model predictive control," *AIChE Journal*, vol. 45, no. 8, pp. 1743–1752, 1999. [Online]. Available: http://dx.doi.org/10.1002/aic.690450811

[28] D. Li, F. Qian, and P. Fu, "Variance minimization approach for a class of dual control problems," *Automatic Control*, vol. 47, no. 12, pp. 2010–2020, December 2002.

[29] D. van Hessem and O. 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*, vol. 1, pp. 929–934, December 2003.

[30] N. Du Toit, "Robot motion planning in dynamic, cluttered, uncertain environments," Ph.D. dissertation, California Institute of Technology, 2010.

[31] M. Lobo and S. Boyd, "Policies for simultaneous estimation and optimization," in *American Control Conference*, vol. 2, June 1999, pp. 958–964.

[32] L. Blackmore, "A probabilistic particle control approach to optimal, robust predictive control," Massachusetts Institute of Technology, Tech. Rep., 2006.