Robot Motion Planning in Dynamic, Uncertain Environments

Noel E. Du Toit, *Member, IEEE*, and Joel W. Burdick, *Member, IEEE*

***Abstract*—This paper presents a strategy for planning robot motions in dynamic, uncertain environments (DUEs). Successful and efficient robot operation in such environments requires reasoning about the future evolution and uncertainties of the states of the moving agents and obstacles. A novel procedure to account for future information gathering (and the quality of that information) in the planning process is presented. To approximately solve the stochastic dynamic programming problem that is associated with DUE planning, we present a partially closed-loop receding horizon control algorithm whose solution integrates prediction, estimation, and planning while also accounting for chance constraints that arise from the uncertain locations of the robot and obstacles. Simulation results in simple static and dynamic scenarios illustrate the benefit of the algorithm over classical approaches. The approach is also applied to more complicated scenarios, including agents with complex, multimodal behaviors, basic robot–agent interaction, and agent information gathering.**

***Index Terms*—Anticipated measurements, dynamic, information gathering, interaction, motion planning, partially closed-loop, receding horizon control (RHC), uncertain.**

I. Introduction

THIS STUDY is concerned with motion planning in *dynamic*, *uncertain environments* (DUEs). In such environments, robots must work in close proximity with many other moving agents, whose future actions and reactions are difficult to predict accurately. Moreover, only noisy measurements of the robot's own state and those of obstacles and moving agents are available for planning purposes. An example of a DUE 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. The human trajectories cannot be predicted with any certainty, and the behaviors of the individuals may differ, complicating the planning problem. However, some prior knowledge about preferred paths and behaviors may be available and should be integrated when possible. This paper presents a framework and initial algorithmic and simulation results that we hope will provide a foundation for future DUE motion planners.

Manuscript received December 21, 2010; revised May 23, 2011; accepted August 22, 2011. Date of publication September 12, 2011; date of current version February 9, 2012. This paper was recommended for publication by Associate Editor S. Carpin and Editor J.-P. Laumond upon evaluation of the reviewers' comments.

The authors are with the Department of Mechanical Engineering, California Institute of Technology, Pasadena, CA 91125 USA (e-mail: ndutoit@robotics.caltech.edu; jwb@robotics.caltech.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TRO.2011.2166435

Robot motion planning in dynamic environments has recently received substantial attention because of the Defense Advanced Research Project Agency (DARPA) Urban Challenge [1] and growing interest in service and assistive robots (see, e.g., [2] and [3]). In urban environments, traffic rules define the expected behaviors of the dynamic agents and constrain expected future locations of moving objects. In other applications, agent behaviors are less well defined, and the prediction of their future trajectories is more uncertain.

Previously proposed motion planning frameworks handle only specific subsets of the DUE problem. Classical motion planning algorithms [4] mostly ignore uncertainty when planning. When the future locations of moving agents are known, the two common approaches are to add a time dimension to the configuration space, or to separate the spatial and temporal planning problems [4]. When the future locations are unknown, the planning problem is solved locally [5]–[7] (via reactive planners) or in conjunction with a global planner that guides the robot toward a goal [4], [7]–[9]. The probabilistic velocity obstacle approach [10] extends the local planner to uncertain environments, but it is not clear how the method can be extended to capture more complicated agent behaviors (a constant velocity agent model is used).

The DUE problem is stochastic. Planning algorithms that account for stochastic uncertainty have been applied largely in static environments. Two types of stochastic systems are distinguished: nondeterministic (the uncertainties lie in a bounded set [11], [12]), and probabilistic (the uncertainties are described using probability distributions) [4]. This work uses a probabilistic formulation. One of the first stochastic robotic planning approaches was preimage backchaining [13]. Since then, discrete search strategies have also been extended to plan in belief space (see, e.g., [14]–[17]). The belief roadmap method [18] and the stochastic motion roadmap [19] builds a connected, collision-free graph in the static uncertain configuration space during a learning phase and then queries the (static) graph during execution. The benefit of this approach is reduced in dynamic environments since the graph needs to be reconstructed at each planning cycle. Alternatively, the problem can be posed as a stochastic dynamic program (SDP) [20]. When the system's dynamic equation is time invariant and the stage cost is constant (which is not the case for the problem considered here), SDP can be solved using partially observable Markov decision process (POMDP) methods [20], [21]. Otherwise, the SDP can be approximately solved with a Rollout algorithm (a limited lookahead policy) or a restricted information approach [20]. A special case of these approaches is the receding horizon control (RHC) framework which has been extended to a stochastic

RHC (SRHC) formulation in the case of robot localization uncertainty (see, e.g., [22]–[24]). For the related problem of probabilistic dynamic target tracking, a promising forward-search approach is proposed that makes efficient use of the linearity of the system, a lack of collision constraints, and the independent, Gaussian-distributed noise terms [25].

Previous work on the integration of interactive robot–agent models into the planning process is limited. Kluge and Prassler [26] introduced reflective navigation: The agent will maximize some utility function. This approach assumes knowledge of that utility function and dynamic capabilities of each agent. Berg *et al.* [27] introduce the reciprocal velocity obstacle to approximate the effect of agent deliberation: 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, but does not account for uncertainty and is limited to very simple agent behaviors.

Better prediction of future system states can result in greater efficiency of robot action in DUEs. Short-term predictors evolve the future state of the dynamic agents using a simple model (e.g., a constant velocity model [28], [29]). Longer term predictions can be improved by learning and using the dynamic agents' preferred paths to predict future states (see, e.g., [30] and [31]) or by inferring environment structure to inform the prediction process (see, e.g., [32] and [33]). Henry *et al.* [34] learn planning strategies from example pedestrian trajectories.

While individual components of the DUE problem have been previously considered, a comprehensive framework that integrates planning, prediction, and estimation is missing. This paper represents a formal effort to integrate these activities, while also incorporating the effect of anticipated future measurements in the motion planning process.[1] As shown by example, the proper inclusion of these effects can improve robot performance in dynamic environments. Because the exact DUE solution is intractable, we introduce the stochastic DP and RHC frameworks (see Section III) and present the partially closed-loop RHC (PCLRHC) approach in Section IV. Key results on probabilistic constraint satisfaction are described in Section V (see [36] for details). Computational considerations for the PCLRHC and the general DUE problem are presented in Section VI. Simulation results for a robot navigating in simple static and dynamic scenarios are presented in Section VII-A to illustrate some of the characteristics of this method. The approach is next applied to more complicated scenarios: agents with complicated, multimodal behavioral models (see Section VII-B) and basic robot–agent interactions (see Section VII-C), including information gathering.

II. Problem Statement

This section develops a standard constrained stochastic optimal control problem statement which encompasses the aspects of the DUE problem that is developed in the remainder of this paper. Our technical contribution is the subsequent analysis of this problem.

Let $x_i \in \mathbb{X}$ denote the system state (e.g., robot and agent positions and velocities) at time t_i, where the *state space* $\mathbb{X} \subseteq \mathbb{R}^{n_x}$ has dimension n_x. The control action u_i lies in the *action space* $\mathbb{U}$: $u_i \in \mathbb{U} \subseteq \mathbb{R}^{n_u}$. The disturbance, $\omega_i \in \mathbb{W} \subseteq \mathbb{R}^{n_\omega}$, models uncertainty in the objects' governing dynamic model.[2] This disturbance may be parameterized in terms of the system state and control, and is described by a conditional distribution: $\omega_i(x_i, u_i) \sim p(\omega_i|x_i, u_i)$. The disturbance is assumed to be mutually independent of previous disturbances, which are conditioned on x_i and u_i. The system (which consists of robot and dynamic agents) evolves in discrete time intervals (enumerated stages), starting at the current stage k. The evolution is governed by a discrete-time dynamic equation

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

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

The measurement y_i is an element of the *measurement space* $\mathbb{Y}(x_i)$, $y_i \in \mathbb{Y}(x_i) \subseteq \mathbb{R}^{n_y}$, and is corrupted by noise, $\nu_i \in \mathbb{V} \subseteq \mathbb{R}^{n_\nu}$, where $\nu_i(x_i) \sim p(\nu_i|x_i)$ may be parameterized in terms of the system state. The C^2 sensor mapping $h : \mathbb{X} \times \mathbb{V} \to \mathbb{Y}$ maps states to measurements

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

Our goal is to calculate a *feedback control policy* (e.g., $u_i = -Kx_i$) which defines a control action for every reachable system state from the current stage k to the N_{th} stage. Since the system states are uncertain, the control law at some future stage i is defined in terms of the *information state* (I-state), I_i. This I-state captures all the information that is available to controller, including the measurement y_i (see Section III-A). The control policies over the planning horizon are denoted by $\Pi = \{\pi_k(I_k), \ldots, \pi_{N-1}(I_{N-1})\}$ and can be thought of as output feedback policies. The set of admissible policies are defined as $\widetilde{\Pi}$.

In order to evaluate different possible trajectories for the purpose of planning, a stage-additive cost function, which captures planning and missions goals (e.g., $l_i(x_i, \pi_i(I_i)) = x_i^T x_i$ will draw the robot to the origin), is assumed:

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

Additionally, the controls and the states of the system may be constrained by nonlinear inequalities $g(x_i, u_{i-1}) \leq 0 \; \forall i = k \ldots N$ (e.g., collision avoidance). Since the system states are uncertain, these constraints are imposed as *chance constraints* of the form: $P(g(x_i, u_{i-1}) \leq 0) \geq \alpha$, where α is the level of confidence (see Section V).

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 while satisfying the constraints

[1] Initial results, which are presented in [35], are augmented here through a complete analysis of the PCLRHC approximations and computational requirements. Simulation results for cluttered, complicated dynamic scenarios, interactive robot–agent examples, and agent information gathering are presented.

[2] Object refers to either the robot, dynamic agents, or static obstacles.

$$\Pi^{(*)} = \arg\min_{\Pi \in \widetilde{\Pi}} E\left[L(x_{k:N}, \Pi, \omega_{k:N-1})\right]$$
$$\begin{aligned} s.t. \quad & x_{i+1} = f(x_i, u_i, \omega_i) \\ & y_i = h(x_i, \nu_i) \qquad \forall i = k \ldots N \\ & P(g(x_i, u_{i-1}) \le 0) \ge \alpha. \end{aligned} \tag{4}$$

This feedback control policy is generally difficult to obtain. To gain insight, we first consider the unconstrained version of this problem, which can be reformulated as an SDP problem, before the constraints are incorporated in the SRHC framework.

III. Stochastic Dynamic Program Approximations

The problem of Section II, without the state and control constraints, can be solved using the SDP formulation [20], [37] by converting the problem to belief space. The SDP approach constructs an optimal *feedback control policy* (which is defined for every reachable belief state), which makes the approach computationally intensive. The SRHC framework is an approximate solution to the SDP problem that additionally incorporates constraints. The SRHC approach recursively solves for a *sequence of control actions* at every planning cycle. The control is selected only for the states that the system is predicted to visit. It is useful to write the problem in terms of belief states to gain intuition about the use of anticipated information and draw parallels to standard robot motion planning formulations.

A. Planning in Belief Space

I-states summarize the information that is available to the planner [4]. Two I-states of interest here are the *history I-state* and the *belief state*. Let the *history of measurements* at the ith stage be $y_{0:i} \triangleq \{y_0, \ldots, y_i\}$ and the *history of controls* be $u_{0:i} \triangleq \{u_0, \ldots, u_i\}$. The *history I-state* I_i is defined as

$$I_i = \{I_0, u_{0:i-1}, y_{0:i}\} \tag{5}$$

where I_0 is the initial history I-state (i.e., *a priori* information). I-states evolve according to a dynamic transition function. The propagated I-state is unpredictable because next the measurement y_{i+1} is unknown. *The measurement y_{i+1} plays the role of a process noise in the history I-space* [20].

Since the history I-state can be unwieldy (its dimension grows over time), it is often useful to work with the simpler *belief state*, which is derived from the history I-state. The transformation between these spaces is exact if it is assumed that a Markov model governs the system's evolution [4] (the current state and control is the best predictor of future states [21]). The belief space is just another state space for which the states are defined as $b_i \triangleq p(x_i|I_i)$, and the state transition function $b_{i+1} = f_b(b_i, u_i, y_{i+1})$ is obtained from Bayes' rule [20], [21]. The cost function is converted into an equivalent function of the belief states (using a slight abuse of notation to highlight similarities to the optimal control problem for deterministic systems), noting that the control is to be selected and is, thus, not a random variable

$$c_i(b_i, \pi_i) \triangleq E\left[l_i(x_i, \pi_i, \omega_i)|I_i\right] \tag{6}$$

$$c_N(b_N) \triangleq E\left[l_N(x_N)|I_N\right]. \tag{7}$$

The expected cost is written in terms of the measurements when the problem is converted to the belief space[3]

$$E_{y_{k:N}}\left[c_N(b_N) + \sum_{i=k}^{N-1} c_i(b_i, \pi_i)\right]. \tag{8}$$

The DP algorithm solves this optimization problem with the backward recursion of the cost-to-go function $J_i(b_i)$ [20]:

$$J_N(b_N) = c_N(b_N) \tag{9}$$
$$J_i(b_i) = \min_{\pi_i} \; c_i(b_i, \pi_i) + E_{y_{i+1}}\left[J_{i+1}(b_{i+1})|I_i\right]. \tag{10}$$

The SDP algorithm constructs a feedback law on belief space. However, the set of possible future measurements is infinite in a probabilistic setting. Only a few SDP problems yield closed-form solutions (e.g., linear systems with quadratic cost, Gaussian noise terms, and no constraints) [20], [25]. One must, therefore, seek approximate solutions to the generally intractable SDP.

B. Approximations to Stochastic Dynamic Program

The SDP problem is approximated by 1) recursively solving a simplified problem for a *control sequence* instead of a control policy (e.g., the open-loop control strategy); 2) solving for a control policy over a limited horizon and then approximating the cost-to-go function beyond this horizon (limited lookahead policies); or 3) POMDP methods (which are not applicable here since a constant stage cost is not assumed) [20]. Strategy 1 uses a restricted information set[4] [20] when approximately solving the problem: Future measurements are ignored. The restricted information set at future stage i (based on actual measurements up to the current stage, k) is denoted the "open-loop" I-state:

$$I_i^{OL} = (y_1, \ldots, y_k, u_0, \ldots, u_{i-1}), \; i \ge k. \tag{11}$$

The resulting future belief states are the open-loop predicted distributions $b_i^{OL} = p(x_i|I_i^{OL})$, which can be updated using $b_{i+1}^{OL} = f_b^{OL}(b_i^{OL}, u_i)$ and then used in the SDP algorithm. As a result, the solutions that are obtained with this approximation tend to be conservative. However, the restricted information set defines the future belief states completely for a given control sequence (the problem becomes deterministic in terms of these belief states) and the expectation in (10) can be dropped:

$$J_i(b_i^{OL}) = \min_{u_i} \; c_i(b_i^{OL}, u_i) + J_{i+1}(f_b^{OL}(b_i^{OL}, u_i)). \tag{12}$$

The results of Sections III-A and B can now be used to formulate the SRHC problem to incorporate the constraints.

C. Stochastic Receding Horizon Control in Belief Space

RHC is a suboptimal control scheme that can explicitly incorporate state and control constraints into the planning problem [38], [39]. A simplified version of the problem in Section II

[3]The law of total expectation is used [4]. The Markov assumption is not required for this result.

[4]A restricted information set is a subset of the history I-state that is used to construct a more tractable approximating solution.

is solved over a finite horizon to stage M ($M \leq N$) to obtain a *sequence of control actions*. A portion of the plan is executed before new measurements are obtained, the system states are updated, and the problem is resolved. Feedback is moved to the planning phase, instead of the execution phase (i.e., it is an *outer-loop feedback mechanism*).

RHC was originally developed for deterministic systems. While the best approach to extend the RHC formulation to stochastic systems is still up for debate [22], [24], [40]–[42], it is convenient to convert the problem into the belief space (as per Section III-A):

$$\begin{aligned} \min_{u_{k:M}} \quad & E_{y_{k:M}}\left[c_M(b_M) + \sum_{i=k}^{M-1} c_i(b_i, u_i)\right] \\ s.t. \quad & b_{i+1} = f_b(b_i, u_i, y_{i+1}) \\ & P(g_b(b_i, u_{i-1}) \leq 0) \geq \alpha \qquad \forall i = k, \ldots, M \end{aligned} \tag{13}$$

where $g_b(b_i, u_{i-1})$ is the equivalent nonlinear constraint function that is written in terms of the belief state (see Section V). The dynamics and noise properties of the original system are encoded in the belief state transition function.

This problem formulation is still cumbersome since it is necessary to reason over the complete set of possible future measurements $y_{k:M}$. In a probabilistic system, this is an infinite set. Similar to Section III-B, this problem is commonly approximated by restricting the information set that is used by the planner: Measurements beyond the current stage are ignored, resulting in the open-loop receding horizon control (OLRHC) approach [22], [24], [40]–[42]. The resulting belief states are the objects' open-loop predicted distributions. For this reason, the OLRHC approach tends to be conservative, leading Yan and Bitmead [22] to introduce a "closed-loop covariance" (which is fixed at the one-step open-loop prediction value). This crudely accounts for the anticipated future information which will become available during plan execution. One of our technical contributions is the formal inclusion of future anticipated measurements into the SRHC framework with the PCLRHC approach.

IV. Partially Closed-Loop Receding Horizon Control

To account for anticipated future information, we define an alternative restricted information set, which forms the basis for our PCLRHC approach. To motivate the underlying assumptions, consider a linear system with Gaussian noise, where the dynamic and measurement models are given in Appendixes A and B by (47) and (50), respectively. The belief state transition function for the system is solved by the Kalman filter, resulting in prediction and update steps. Let $\hat{x}_{i|j} \triangleq E[x_i|I_j]$ and $\Sigma_{i|j} \triangleq E[(x_i - \hat{x}_{i|j})(x_i - \hat{x}_{i|j})^T|I_j]$:

Prediction step:

$$\hat{x}_{i|i-1} = A\hat{x}_{i-1|i-1} + Bu_{i-1} \tag{14}$$

$$\Sigma_{i|i-1} = A\Sigma_{i-1|i-1}A^T + FWF^T. \tag{15}$$

Measurement update step:

$$\hat{x}_{i|i} = \hat{x}_{i|i-1} + K_i(y_i - C\hat{x}_{i|i-1}) \tag{16}$$

$$\Sigma_{i|i} = \Sigma_{i|i-1} - K_iC\Sigma_{i|i-1} \tag{17}$$

where the innovation covariance is

$$\Gamma_{i|i-1} = C\Sigma_{i|i-1}C^T + HVH^T \tag{18}$$

and the Kalman gain is

$$K_i = \Sigma_{i|i-1}C^T\Gamma_{i|i-1}^{-1}. \tag{19}$$

The key insight for this system is that incorporating future measurements in the update step has two effects on belief state: 1) The value of the measurement *shifts the center of the belief state* [see (16)], and 2) incorporating the measurement *reduces uncertainty* in the belief state and is independent of the value of the measurement. For the PCLRHC approach, the *most likely* measurement is assumed, and as a result, the center of the resulting belief state is not updated. However, *the fact that a measurement will be taken* is incorporated into the planning process (which updates the shape of the distribution), resulting in an accurate approximation of the true belief state.

A. Assumption and Formulation

In Section III-B, a restricted information set is used to obtain the open-loop approximation to the SDP problem. For the PCLRHC approach, an alternative restricted information set is used to obtain another approximation to the SDP problem, which can then be extended to handle constraints (SRHC). If the most likely measurement $\tilde{y}_i = E[y_i|I_{i-1}^{\mathrm{PCL}}]$ is assumed for the future measurements, the restricted information set is

$$I_i^{\mathrm{PCL}} = (y_1, \ldots, y_k, \tilde{y}_{k+1}, \ldots, \tilde{y}_i, u_k, \ldots, u_{i-1}). \tag{20}$$

The belief state associated with I^{PCL} is

$$b_i^{\mathrm{PCL}} = p(x_i|I_i^{\mathrm{PCL}}) = p(x_i|u_{0:i-1}, y_{1:k}, \tilde{y}_{k+1:i}) \tag{21}$$

and the state transition function is defined as

$$b_{i+1}^{\mathrm{PCL}} = f_b^{\mathrm{PCL}}(b_i^{\mathrm{PCL}}, u_i, \tilde{y}_{i+1}). \tag{22}$$

This belief state is completely defined in terms of the control sequence since the most likely measurement can be calculated, and as a result, the expectation in (13) can be dropped. The resulting optimization problem that is solved by PCLRHC is

$$\begin{aligned} \min_{u_{k:M}} \quad & c_M(b_M^{\mathrm{PCL}}) + \sum_{i=k}^{M-1} c_i(b_i^{\mathrm{PCL}}, u_i) \\ s.t. \quad & b_{i+1}^{\mathrm{PCL}} = f_b^{\mathrm{PCL}}(b_i^{\mathrm{PCL}}, u_i, \tilde{y}_{i+1}) \\ & P(g_b(b_i^{\mathrm{PCL}}, u_{i-1}) \leq 0) \geq \alpha \qquad \forall i = k, \ldots, M. \end{aligned} \tag{23}$$

B. Properties of the Partially Closed-Loop Receding Horizon Control Approximation

To obtain this approximate algorithm, an assumption about the anticipated information was necessary. One concern is that this assumption introduces artificial information into the problem. However, it can be shown for linear systems with Gaussian noise that the information gain during system propagation using the assumption (over ignoring all measurements) is not more than the information gained when actually executing the system (and the true measurements are incorporated as they become available).

Proposition 1: For linear systems with Gaussian-distributed noise, the information gained (which is defined in terms of the relative information entropy $\mathcal{H}$) for the most likely measurement assumption ($\mathcal{H}_{\mathrm{PCL}}$) is less or equal to the information gained when the system is executed and the true measurements are incorporated ($\mathcal{H}_{\mathrm{Ex}}$):

$$\mathcal{H}_{\mathrm{PCL}} \leq \mathcal{H}_{\mathrm{Ex}}. \tag{24}$$

Proof: The relative entropy (which is also known as the KL divergence [43]) is a measure of the information gained by a distribution over some baseline distribution. For Gaussian distributions, the relative entropy has a closed-form solution. Let the baseline distribution be $\mathcal{N}(\mu_0, \Sigma_0)$. The information gained over the baseline distribution by $\mathcal{N}(\mu_1, \Sigma_1)$ is

$$\mathcal{H} = \frac{1}{2}\Bigg(\ln\left(\frac{\det(\Sigma_1)}{\det(\Sigma_0)}\right) + \mathrm{tr}\left(\Sigma_1^{-1}\Sigma_0\right) + (\mu_1 - \mu_0)^T\, \Sigma_1^{-1}\,(\mu_1 - \mu_0) - n_x \Bigg) \tag{25}$$

where n_x is the dimension of the state space, and $\det(\cdot)$ and $\mathrm{tr}(\cdot)$ are the determinant and trace of a matrix, respectively. There are two controllable sources of information gain: The relative size of the covariances (i.e., first two terms) and the shift in means (i.e., third term).

Let the baseline distribution be the open-loop-predicted distribution (i.e., no measurements beyond the current stage k are incorporated) with mean $\mu_0 = \hat{x}_{i|k}$ and covariance $\Sigma_0 = \Sigma_{i|k}$. This baseline distribution is compared with two distributions at stage i: The distribution used in the partially closed-loop (PCL) approach, where the anticipated measurements are incorporated from stage k to i, and the distribution obtained when the *actual* measurements are incorporated (during system execution).

Consider first the distribution obtained from the PCL approximation. By assuming that the most likely measurement will occur, the center of the belief state is not updated [see (16)] so that $\mu_1 = \hat{x}_{i|k}$. However, the covariance is updated (which accounts for the fact that a measurement will occur) so that $\Sigma_1 = \Sigma_{i|i}$. Thus, only the covariance update contributes to the information gain

$$\mathcal{H}_{\mathrm{PCL}} = \frac{1}{2}\left(\ln\left(\frac{|\Sigma_{i|i}|}{|\Sigma_{i|k}|}\right) + Tr\left(\Sigma_{i|i}^{-1}\Sigma_{i|k}\right) - n_x \right). \tag{26}$$

Next, the baseline distribution is compared with the distribution obtained from the actual system execution (when the true measurements are incorporated): $\mu_1 = \hat{x}_{i|i}$, and $\Sigma_C = \Sigma_{i|i}$. The information gained is

$$\mathcal{H}_{\mathrm{Ex}} = \mathcal{H}_{\mathrm{PCL}} + \frac{1}{2}((\hat{x}_{i|i} - \hat{x}_{i|k})^T\, \Sigma_{i|i}^{-1}\,(\hat{x}_{i|i} - \hat{x}_{i|k})). \tag{27}$$

Since $\Sigma_{i|i}$ is positive semidefinite, the quadratic term is nonnegative, and thus, $\mathcal{H}_{\mathrm{PCL}} \leq \mathcal{H}_{Ex}$. If the actual measurement is different from the most likely value (which occurs with probability 1), then the quadratic term results in information gain. ■

Remark: Proposition 1 shows that the most likely measurement assumption is the least informative assumption about the value of the future measurement. From (16), any other assumed value will introduce a shift in the mean, resulting in an artificial information gain. Thus, this approximation optimally incorporates the effect of the measurement by updating the covariance but ignores the information from the value of the (unknown) measurement.

As illustrated in the ensuing examples, the robot can make more aggressive plans in the presence of uncertain static and dynamic obstacles by accounting for the fact that future measurements will be taken, although their values cannot yet be predicted. However, we first return to the probabilistic state constraints in the problem formulation.

V. Chance Constraints

It is often necessary to impose nonlinear inequality constraints of the form $g(x_i, u_i) \leq 0$ on the system states when solving a motion planning problem (e.g., for obstacle avoidance). However, when the system states are described by unbounded probability distributions (e.g., normal distributions), there is no guarantee that the constraint can be satisfied for all possible realizations of the states. It is instead necessary to introduce *chance constraints* on the states, which are of the form $P(g(x_i, u_i) \leq 0) \leq \delta$, where δ is the *level of confidence*. The constraints are specified as limits on the probability of constraint violation.

Two types of chance constraints are considered: 1) linear constraints (e.g., velocity constraints) and 2) collision constraints (e.g., between the robot and dynamic agents). Results are summarized here (see [36] and [44] for details).

A. Linear Constraints on Gaussian-Distributed States

Let the linear chance constraint be of the form $P(a^T x \leq b) \leq \delta$. Assume Gaussian-distributed state variables x. Then, the chance constraint is satisfied iff

$$a^T \hat{x} + F^{-1}(\delta) \times \sqrt{a^T \Sigma a} \leq b \tag{28}$$

where $\hat{x} \triangleq E[x]$, and $\Sigma \triangleq E[(x - \hat{x})(x - \hat{x})^T]$. $F^{-1}(\delta)$ is the inverse of the cumulative distribution function for a *standard* scalar Gaussian variable [36].

B. Probabilistic Collision Checking

Probabilistic collision checking between the robot and an agent can be formulated as a chance constraint of the form $P(C) \leq 1 - \alpha$, where C is the collision condition (defined later). Let $\mathbb{X}_R(x_R) \subset \mathbb{R}^{n_x}$ be the set of points occupied by the robot (centered at x_R) and $\mathbb{X}_A(x_A) \subset \mathbb{R}^{n_x}$ be the set of points that are occupied by the agent (centered at x_A). The *collision condition* is defined as $C(x_R, x_A) : \mathbb{X}_R(x_R) \cap \mathbb{X}_A(x_A) \neq \{\emptyset\}$. The probability of collision is defined in terms of the joint distribution of the robot and agent as

$$P(C) = \int_{x_R}\int_{x_A} I_C(x_A, x_R)\, p(x_R, x_A)\, dx_R\, dx_A \tag{29}$$

where I_C is the indicator function, which is defined as

$$I_C(x_A, x_R) = \begin{cases} 1, & \text{if } \mathbb{X}_R(x_R) \cap \mathbb{X}_A(x_A) \neq \{\emptyset\} \\ 0, & \text{otherwise.} \end{cases} \quad (30)$$

This formulation of probabilistic collision checking is investigated by Du Toit [36] and can be implemented using Monte Carlo simulation (MCS). Alternatively, a small-object assumption yields a closed-form solution to the probability of collision (assuming Gaussian distributions, $x_R \sim \mathcal{N}(\hat{x}_R, \Sigma_R)$, and $x_A \sim \mathcal{N}(\hat{x}_A, \Sigma_A)$):

$$P(C) \approx V_{\mathcal{R}} \times \frac{1}{\sqrt{\det(2\pi\Sigma_C)}} \exp\left[-\frac{1}{2}(\hat{x}_R - \hat{x}_A)^T \Sigma_C^{-1} (\hat{x}_R - \hat{x}_A)\right] \quad (31)$$

where $V_{\mathcal{R}}$ is the volume of the robot, and $\Sigma_C \stackrel{\triangle}{=} \Sigma_R + \Sigma_A$. This solution results in a quadratic constraint on the robot mean state in terms of the agent mean state

$$(\hat{x}_R - \hat{x}_A)^T \Sigma_C^{-1} (\hat{x}_R - \hat{x}_A) \geq \kappa(\Sigma_C, \delta, V_{\mathcal{D}}) \quad (32)$$

which defines an ellipse around the agent that the robot must avoid for the constraint to be satisfied (see [36] for details).

C. Probabilistic Safety of the System

No practical robot can react instantaneously to unforeseen changes in the environment because of the dynamics of the system. Consideration of additional future stages during the planning phase is required to guarantee system safety. The number of additional stages that needs to be considered is highly problem dependent. One approach is to avoid inevitable collision states (ICS) [45]: states from which collisions cannot be avoided. In ICS, the objective is to identify at least one control sequence (from a system state) that avoids collisions for all future times. ICS has recently been extended to stochastic systems [46], [47]. The problem that is considered here is different: The probabilistic safety associated with a specific control sequence must be evaluated. The proposed approach is to guarantee probabilistic safety over some horizon by appropriately *conditioning* the collision chance constraints: All probable disturbances and measurements are considered over this horizon. This, in combination with the recursive formulation of the problem, is *expected* to render the approach insensitive to measurement outliers (this has been verified in simulation only). This important topic cannot be sufficiently addressed here (see [44]).

VI. Computational Considerations

The recursive nature of the PCLRHC approach and the requirement to operate in an uncertain environment necessitates a real-time solution of the motion planning problem. However, the underlying optimization problem that must be solved is highly dependent on the objective function, constraints, etc., making a concise treatment of the algorithm's computational complexity difficult. Instead, the complexity for the OLRHC and PCLRHC approaches are compared, and computational burdens specific to the DUE applications are described.

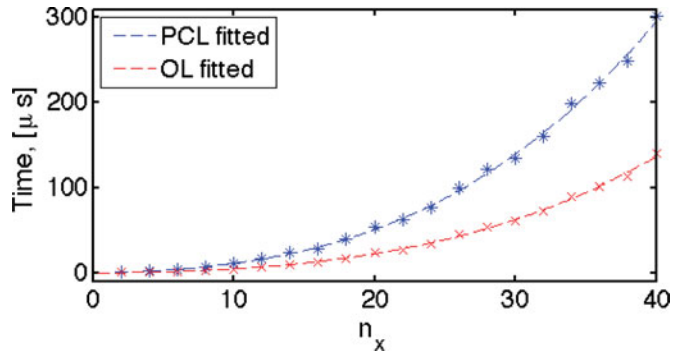

Fig. 1. Calculated processing times for the belief state transition function calculated for the PCLRHC and OLRHC approaches.

The following notation is used: Let n_x be the robot's C-space dimension, and let b_i^R and $b_i^{(j)}$ be the belief states associated with the robot and jth agent at stage i, respectively. μ_i^R and Σ_i^R is the mean and covariance of the robot belief state b_i^R. n_X is the dimension of the augmented state space, consisting of the robot's and agents' C-spaces (see, e.g., Section VII-C). Let $b_i = [b_i^R;\ b_i^{(1)}; \ldots b_i^{(n_A)}]$, where n_A is the number of agents.

A. Complexity of Open-Loop Receding Horizon Control Versus Partially Closed-Loop Receding Horizon Control

The main computational difference between the OLRHC and PCLRHC formulations lies in the propagation of the belief states. The belief state transition function is generally derived from Bayes' Rule. For linear systems with Gaussian noise, the belief state evolution is given by the Kalman Filter [see (14)–(17)]. For the OLRHC, only the prediction step of the filter is executed, which is known to scale as $\mathcal{O}(n_X^3)$. For the PCLRHC, the additional covariance update involves a matrix inversion and matrix multiplications, which also scales as $\mathcal{O}(n_X^3)$. Thus, the computational complexity of this step differs from that of the OLRHC approach by a constant factor. In order to evaluate this factor, the average processing times[5] for the two algorithms are plotted against the dimension of the state space in Fig. 1. A cubic function is fitted to the data to obtain a factor of 1.991.

When considering nonlinear systems or systems with non-Gaussian noise, no closed-form solution to the belief state transition function is available. In this case, a nonlinear filter (e.g., an extended Kalman filter or particle filter) must be used. The relative scaling of the calculations for these approaches has not been investigated.

B. Complexity Arising From the Dynamic, Uncertain Environment Problem Formulation

The presence of multiple obstacles in the DUE problem results in multiple local minima in the nonconvex optimization problem (e.g., passing to the left or right of an obstacle). Additionally, the behaviors of the moving objects may be coupled (because of interactions), and the noise models for the objects

[5] Averaged processing times for 500 000 function evaluations on a 2.66-GHz Intel Core 2 Duo processor with 4-GB RAM using the GSL CBLAS library.

may vary spatially and/or temporally (e.g., the uncertainty in the position measurements of the obstacles may be a function of the distance between the robot and the obstacle). The separation principle from control theory does not apply in general, and the control selection and estimation problems are coupled.

1) Multiple Obstacles: Since the general DUE problem is nonconvex, most optimization schemes cannot guarantee a global solution to this problem, and approaches that are insensitive to local minima should be considered. Sampling-based motion planners [4] may be particularly suited to these types of problems and additionally have proven to be computationally efficient. This benefit comes at the price of solving a discretized (approximate) optimization problem.

2) Uncertainty: Incorporating uncertainty in the problem structure impacts the complexity of the system state propagation and the evaluation of constraints in particular. In general, the belief state propagation (see Section VI-A) occurs inside the optimization loop since control selection and estimation are coupled. However, for linear Gaussian systems, covariance propagation is independent of system states and controls and can be computed outside the optimization loop.

The functional form of the constraints and noise distributions affect the computational complexity of constraint evaluation. For linear constraints and Gaussian-distributed uncertainty, constraint evaluation requires an additional matrix-vector multiplication [see, e.g., (28)], which scales as $\mathcal{O}(n_x^2)$. Greater complexity will arise for non-Gaussian noise or nonlinear constraints (e.g., collision chance constraints may require solution by MCS, see [35] for details).

3) Dependence of Dynamic Models: When modeling the interaction between the robot and agents, the augmented state space dimension $n_X = (n_A + 1) \times n_x$ implies a belief state transition calculation of order $\mathcal{O}(n_X^3) = \mathcal{O}((n_A + 1)^3 \times n_x^3)$. When the robot and obstacle models are independent, belief state propagation scales in $\mathcal{O}((n_A + 1) \times n_x^3)$. Since the robot motion does not affect the obstacle behavior in this latter case, the agents' belief state updates can be performed outside of the optimization loop.

Next, we show through simulation that the PCLRHC provides real benefit over the OLRHC approach.

VII. Simulations and Applications

The OLRHC (which is typical of current SRHC practice) and the PCLRHC approaches are compared in static and dynamic scenarios. While linear dynamical models (for both robot and agents) with Gaussian noise were used for simplicity, our approach is more generally applicable.[6] To demonstrate the flexibility of the PCLRHC approach to handle a wide variety of applications and situations, we extend and apply the method to several scenarios. First, the robot operates in a static environment, followed by a dynamic scenario with simple agent behaviors. Even these trivial examples show how anticipated future

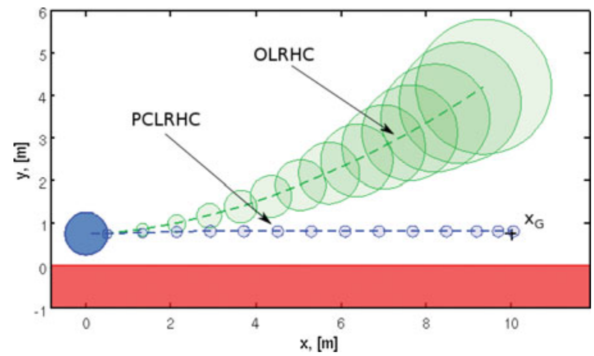

Fig. 2. Trajectories that are planned by the OLRHC (green, dashed) and PCLRHC (blue, dashed) methods, with associated $1-\sigma$ positional uncertainty ellipses. The static obstacle is red.

measurements and chance constraints (which are fundamental to the DUE problem) can affect the motion planning outcome. Next, more complicated agent behaviors are considered, including agents with multiple possible destinations or models and examples with robot–agent interaction. Finally, to highlight the PCLRHC approach's ability to incorporate the quality of anticipated future information, an information gathering example is presented.

All simulation results assume the following.

1) A disk robot (0.5-m radius) must navigate in a planar environment, possibly with disk agents (0.5-m radius).
2) The robot state consists of position and velocity components: $x_i^R = [\mathrm{p}_i^{Rx} \;\; \mathrm{p}_i^{Ry} \;\; \mathrm{v}_i^{Rx} \;\; \mathrm{v}_i^{Rx}]^T$. Similarly for the agent states. The initial state estimates for the objects are Gaussian distributed: $x_0^R \sim \mathcal{N}(\hat{x}_{0|0}^R, \Sigma_{0|0}^R)$ and $x_0^A \sim \mathcal{N}(\hat{x}_{0|0}^A, \Sigma_{0|0}^A)$ with $\hat{x}_{0|0}^R$ and $\hat{x}_{0|0}^A$ specified later, and $\Sigma_{0|0}^R = \Sigma_{0|0}^A = 0.01 \times \mathcal{I}_4$ (unless specified otherwise). $\mathcal{I}_4$ is the 4×4 identity matrix.
3) The dynamic and measurement models used are given in Appendixes A and B. Gaussian-distributed process and measurement noise terms with $W_R = W_A = V_R = V_A = 0.01 \times \mathcal{I}_2$ are used.
4) The objective is to minimize the *expected value* of the quadratic stage-additive cost function
$$\sum_{i=0}^{M-1} \left\{ \left(x_i^R - x_G\right)^T Q_i \left(x_i^R - x_G\right) + (u_i^R)^T R_i (u_i^R) \right\} + \left(x_M^R - x_G\right)^T Q_M \left(x_M^R - x_G\right) \tag{33}$$
where Q_i and R_i are defined later.
5) In the associated figures, circles indicate the robot (blue) and agents (red). The robot goal is shown with a black + and the agent destination with a black *. For the OLRHC approach, we distinguish between the *planned* solution (green, dashed) and *executed* trajectories (green, solid). The $1-\sigma$ positional uncertainty ellipses are overlayed along the planned trajectory (see, e.g., Fig. 2). Similarly blue trajectories depict the planned (dashed) and executed (solid) PCLRHC trajectories. Finally, the *predicted* (red,

[6] In Section IV, the PCLRHC approximation is motivated by the Kalman Filter since the algorithm is formulated in terms of belief states, and other non-linear filtering techniques can be utilized. The appropriateness of specific non-linear filters must be evaluated on a case-by-case basis.

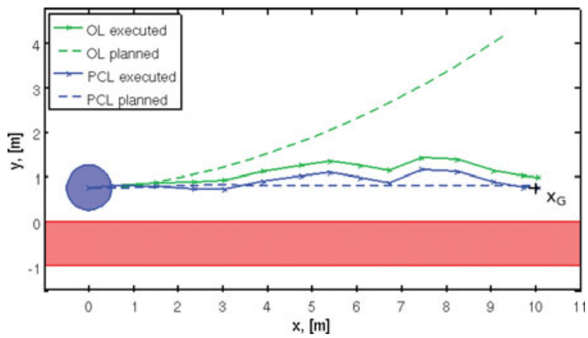

Fig. 3. Comparison of the planned and executed paths produced by the OLRHC and PCLRHC approaches.

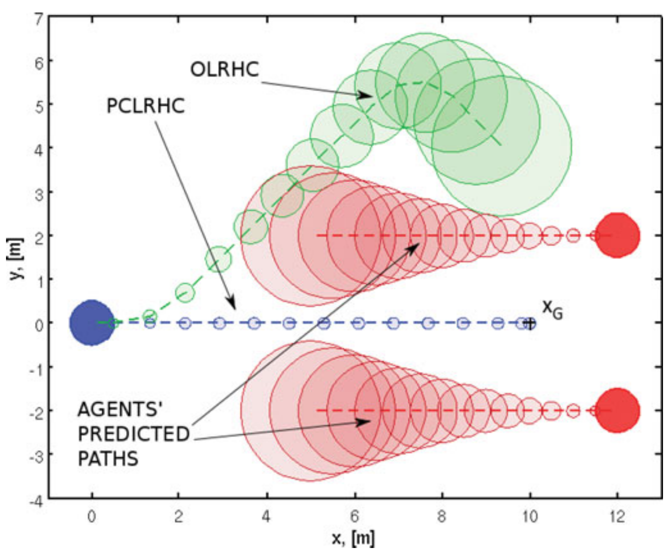

Fig. 4. Planned trajectories because of two oncoming agents.

dashed) and *actual* (red, solid) agent trajectories are indicated with $1-\sigma$ positional uncertainty ellipses (see, e.g., Fig. 4).

6) Consider the constraints (velocity and control constraints illustrate the ability to handle realistic constraints).
   a) Collision chance constraints (see Section V-B) are imposed at each stage with $\delta_{c,i} = 0.01$.
   b) Control magnitudes are less than unity at each stage.
   c) Each velocity component is limited to $[-2, 2]$ (except Example 8): $P(\mathrm{v}_i^{Rx} > 2|I_{i-2}) \leq \delta_{v,i}$ and $P(\mathrm{v}_i^{Rx} < -2|I_{i-2}) \leq \delta_{v,i}$ with $\delta_{v,i} = 0.01$ and similarly for v_i^{Ry}.

A. Simple Environments

1) Example 1 (Static Environment): The robot (whose motions are governed by the dynamic model of Appendix A-A) operates in the vicinity of a single rectangular obstacle. The robot's initial and goal locations are chosen so that the robot must skirt the obstacle to reach the goal. Measurements are governed by the model of Appendix B1. The robot initial mean state is $x_{0|0}^R = [0 \quad 0.75 \quad 1 \quad 0]^T$, and the goal lies at $x_G = [10 \quad 0.75 \quad 0 \quad 0]^T$. The collision chance constraint in this case is $P(x_i^{(2)} < 0|I_{i-2}) \leq \delta_{p,i}$, where $\delta_{p,i} = 0.01$. For the cost function parameters, $Q_M = \mathrm{diag}(10, 10, 0, 0)$, $Q_i = \mathrm{diag}(1, 1, 0, 0)$, and $R_i = \mathrm{diag}(1, 1, 0, 0)$, which are used for $i = 0, \ldots, M-1$.

The initial *planned* trajectories for each approach are shown in Fig. 2. The optimal solution is to travel in a straight line from the initial location to the goal. However, the planned open-loop path diverges from the straight line because of the chance constraints: Since future measurements are not considered during this plan, the growth in the predicted uncertainty forces the robot to move away from the obstacle. The obtained solution is very conservative, and the goal is not reached. Since the effects of the anticipated future measurements are incorporated in the PCLRHC plan, growth in uncertainty is bounded. The initially planned PCLRHC solution drives the robot directly to the goal.

The *executed* paths for the two approaches (see Fig. 3) are similar because of the RHC outer loop feedback mechanism: The problem is resolved at each planning cycle as new measurements are taken. However, the planned and executed OLRHC trajectories differ substantially, as the planner relies almost exclusively on the outer-loop feedback mechanism to execute a reasonable trajectory. On the other hand, the PCLRHC's 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. The PCLRHC approach efficiently uses the anticipated future information when solving the planning problem.

2) Example 2 (Oncoming Agents): Two dynamic obstacles move toward the robot. Their motions are independent of the robot's actions (i.e., the agents do not "react" to the robot's presence). However, the agent states enter the problem through the collision chance constraints. The robot's initial mean state is $\hat{x}_{0|0}^R = [0 \quad 0 \quad 10]^T$, and its goal lies at $x_G = [10 \quad 0 \quad 0 \quad 0]^T$. The dynamic model of Appendix A1 and linear position measurement model (see Appendix B1) govern each agent, with $\hat{x}_{0|0}^{A1} = [12 \quad 2 \quad 1 \quad 0]^T$ and $\hat{x}_{0|0}^{A2} = [12 \; -2 \; 1 \; 0]^T$. A collision constraint is applied to each agent (see Section V-B).

Fig. 4 shows the first stage *planned* trajectories for the OLRHC and PCLRHC approaches. The PCLRHC method obtains a significantly improved *planned and executed* trajectory as compared with the OLRHC approach: Because of the growth in uncertainty, the OLRHC approach cannot plan a path between the agents and must instead move around both agents. The PCLRHC approach can progress directly toward the goal and its executed trajectory is significantly shorter than the OLRHC path.

3) Example 3 (Monte Carlo Simulation of Crossing Agents): To confirm that the performance improvement is not specific to the chosen scenario, we carried out an MCS in which two dynamic obstacles cross the space between the robot and the goal. The same models, cost function, and constraints are used from the previous example. The simulation is repeated 200 times with randomized initial conditions of robot and agents (see Table I). The robot moves from left to right ($x_G = [12 \; 0 \; 0 \; 0]^T$), agent 1 moves from north to south, and agent 2 moves from south to north (crossing).

TABLE I
MC SIMULATION INITIAL CONDITION RANGES

	x	y	Heading	$\|v\|$ $[m/s]$
Robot	0	$[-2,\ 2]$	$[-22.5°,\ 22.5°]$	1.2
Agent 1	$[4,\ 8]$	6	$[-120°,\ -75°]$	1
Agent 2	$[4,\ 8]$	-6	$[75°,\ 120°]$	1

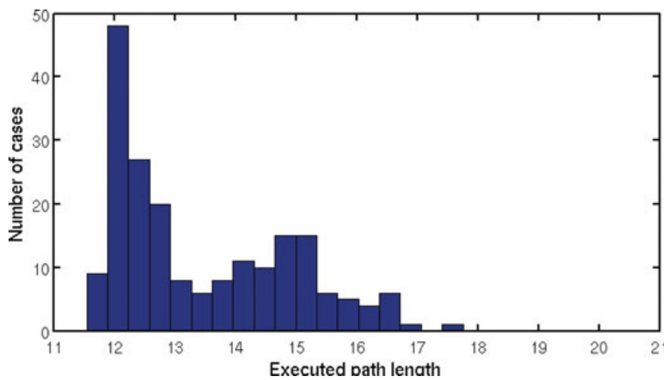

Fig. 5. Histogram of executed path lengths for the PCLRHC approach. The robot either reacts to the agents (peak around 15) or moves directly toward the goal (peak around 12.5).

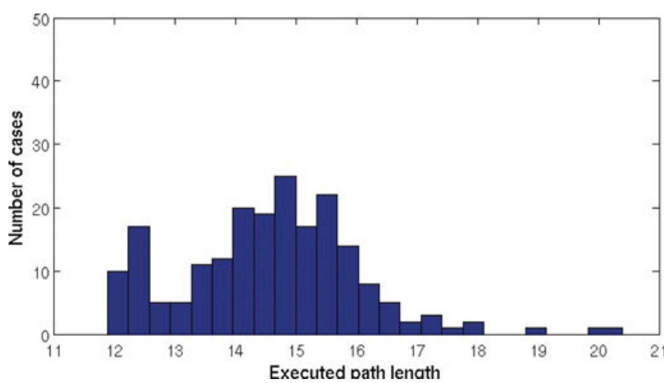

Fig. 6. Histogram of OLRHC executed path lengths (see Fig. 5).

The histograms of the *executed* path lengths show that the PCLRHC approach (see Fig. 5) more often finds direct paths to the goal (peak at 12.5 m) than the OLRHC approach (see Fig. 6). A larger second peak (at 15 m) in the OLRHC histogram suggests that the robot reacts to the agents more often, resulting in longer paths. The PCLRHC approach obtains shorter average executed paths. On a case-by-case comparison, the PCLRHC approach finds shorter paths in 72.0% of the trials, with at least a 10% path length improvement in 37.5% and at least a 20% improvement in 17.5% of the trials.

4) Example 4 (High Clutter Environment): One of the advantages of the PCLRHC approach over the OLRHC approach is the ability to handle high-clutter environments, as illustrated in Figs. 7 and 8. The robot is initialized at the origin and with the objective of reaching $x_G = [10 \quad 0 \quad 0 \quad 0]^T$. There are four static obstacles of radius 0.25 m in the environment (magenta). The measurement noise for the static obstacles has covariance $V_A = 0.01 \times \mathcal{I}_2$. The static obstacles are initialized at $[3 \quad 0]^T$, $[5 \quad 3]^T$, $[5 \quad -2]^T$, and $[8 \quad -1]^T$, respectively. In addition, three dynamic obstacles inhabit the robot's environment (red). The dynamic and measurement models of Section VII-A2 are employed. The dynamic obstacles are initialized at $\hat{x}^{A1}_{0|0} = [2 \quad 3 \quad 0.707 \quad -0.707]^T$, $\hat{x}^{A2}_{0|0} = [9 \quad 3 \quad -0.354 \quad -0.354]^T$, and $\hat{x}^{A3}_{0|0} = [9 \quad -3 \quad -1 \quad 0]^T$, respectively.

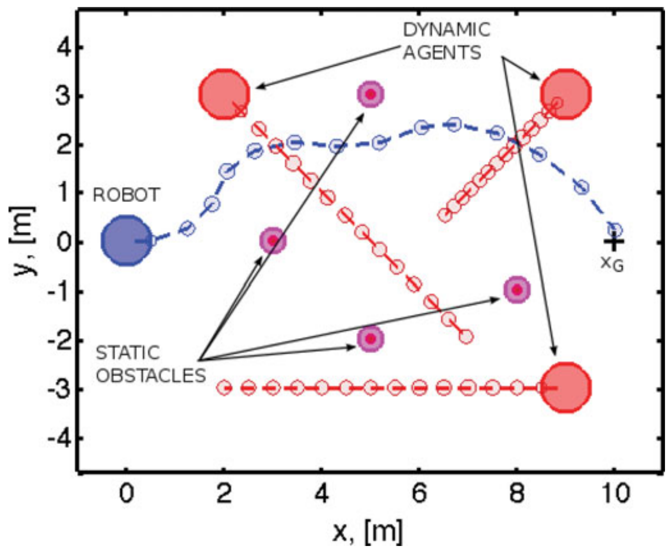

Fig. 7. Planned trajectory in a cluttered environment for the PCLRHC approach.

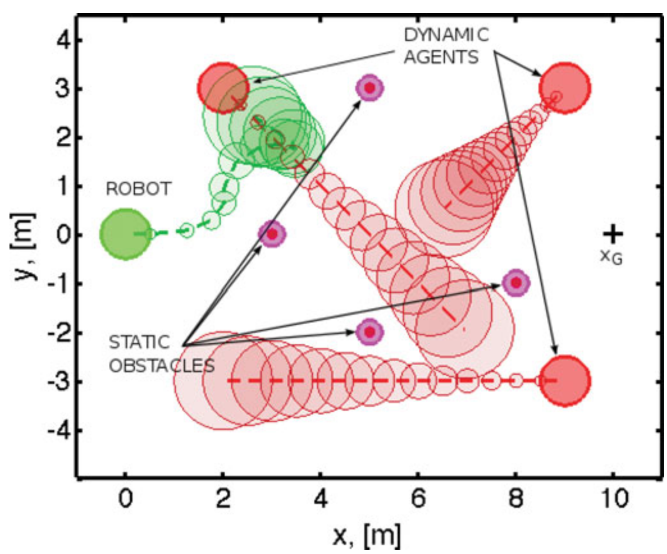

Fig. 8. Planned trajectory in a cluttered environment for the OLRHC approach.

As expected, the OLRHC approach fails to find a reasonable trajectory through this environment, because of the undesired growth in uncertainty and the associated conservatism. In contrast, the PCLRHC approach manages to steer the robot through the field of dynamic and static obstacles toward the goal.

B. Complicated, Independent Agent Behavior

The previous section showed the advantages that are obtained over the OLRHC approach when uncertainty growth is bounded by incorporating future anticipated measurements in the PCLRHC approach. This section considers scenarios where agents have more complicated behaviors (although still independent of the robot state), which results in greater uncertainty in future states of the system.

1) Example 5 (Agents With Multiple Destinations): In the cafeteria example, moving agents might be attracted to specific destinations (e.g., the salad bar or the cashier). This information can improve long-term agent position prediction. Agents with multiple possible destinations can be modeled by a system with discrete unknown parameters, with each parameter value modeling a different potential agent goal. The resulting probability distribution that describes the future agent location is multimodal.[7]

Consider the robotic system, cost function, and constraints of Section VII-A2. The goal is $x_G = [8\ 1\ 0\ 0]$. The initial robot state mean and covariance are $\hat{x}^R_{0|0} = [0\ 1\ 1\ 0]^T$ and $\Sigma^R_{0|0} = 0.1 \times I_4$. Assume that an agent is drawn to one of two possible destinations (see Appendix A2, with $k = 0.1$, $\theta^{(1)} = [10 \quad 4]$ and $\theta^{(2)} = [10 \ -4]$). The agent does not change destinations during the plan execution. A linear position measurement model (see Appendix B1) is assumed. The agent initial state has mean $\hat{x}^A_{0|0} = [2 \quad 0 \quad 1 \quad 0]^T$ and covariance $\Sigma^A_{0|0} = 0.1 \times I_4$. The destinations are initially approximately equally likely: $P(\theta^{(1)}|I_0) = 0.501$ and $P(\theta^{(2)}|I_0) = 0.499$. $\theta^{(2)}$ is the true destination.

Using the aforementioned models, the agent behavior is governed by

$$x^A_i = A(\theta)x^A_{i-1} + Bu^A_{i-1} + F\omega^A_{i-1} + f_\theta(\theta) \tag{34}$$

$$y^A_i = C(\theta)x^A_i + H\nu^A_i + h_\theta(\theta) \tag{35}$$

where the parameter vector $\theta \in \begin{bmatrix}\theta^{(1)} & \theta^{(2)} & \ldots & \theta^{(J)}\end{bmatrix}$ can assume one of J possible values, $\omega^A_{i-1} \sim \mathcal{N}(0, W)$ and $\nu^A_i \sim \mathcal{N}(0, V)$. It can be shown that the estimated agent state distribution is a weighted sum of Gaussian components [44]:

$$p(x^A_k|y^A_{1:k}, u^A_{0:k-1}, \{\theta^{(j)}\}) = \sum_{j=1}^{J} \mathrm{w}^{(j)}_k \mathcal{N}(\hat{x}^{A(j)}_{k|k}, \Sigma^{A(j)}_{k|k}) \tag{36}$$

where

$$\mathrm{w}^{(j)}_k \triangleq P(\theta^{(j)}|y^A_{1:k}, u^A_{0:k-1}) \tag{37}$$

is the probability that $\theta^{(j)}$ is the true parameter value. At each planning cycle, one filter is required to update each viable parameter value with the latest measurement.

Some flexibility exists in defining the "most likely measurement" for these multimodal distributions. In this paper, the evolution of the weights is not predicted since the effect of the robot's planned path on these probabilities is not modeled. The most likely measurement for each possible parameter value, which denoted the *locally* most likely measurement,[8] is used: $\tilde{y}^{A(j)}_i = E[y^A_i|y^A_{1:k}, \tilde{y}^{A(j)}_{k+1:i-1}, \theta^{(j)}]$. The predicted open-loop evolution of the agent's multimodal distribution is shown in Fig. 9. The multimodal probability distributions at different stages are overlaid. The spreads of the distributions increase because of the open-loop prediction of the future states.

[7]Recent work by He *et al.* [25] considered multimodal agent behavior in the tracking problem.

[8]Alternatively, a *globally* most likely measurement, can be used, generated from the current most probable parameter value, but this was found to introduce undesirable bias.

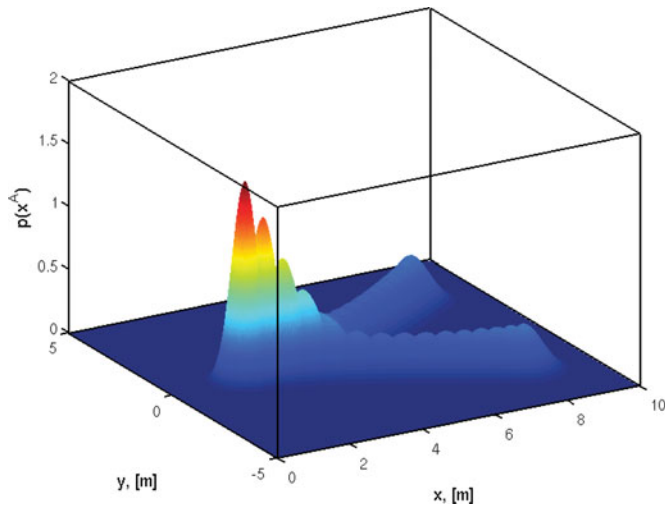

Fig. 9. Predicted open-loop evolution of multimodal distribution. Two components are clearly discernible, and the spread of the components increases.

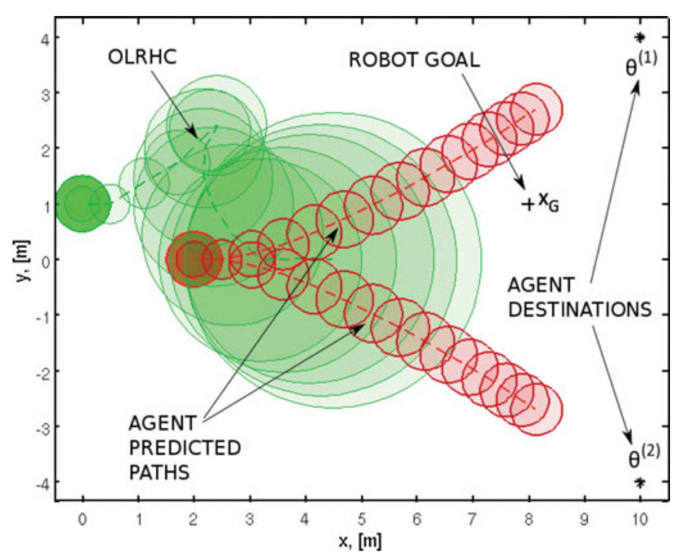

Fig. 10. OLRHC's Initial planned trajectory and the agent's predicted trajectories toward the two possible destinations

The OLRHC's initial planned trajectory is shown Fig. 10. Two agent trajectories (one associated with each destination) are plotted. The growth in agent uncertainty forces the robot to move away from both possible agent trajectories when the chance constraints are imposed, resulting in a very conservative plan. The PCLRHC's planned trajectory (see Fig. 11) moves the robot toward the goal while avoiding the agent.

The sequence of *executed* (solid) and the planned (dashed) trajectories are plotted in Fig. 12 for the PCLRHC (blue) and OLRHC (green) methods. Both possible agent trajectories are plotted (thicker lines indicate higher probability, as estimated by the robot, of being the true behavior). The OLRHC approach is practically unable to plan in the presence of multiple trajectories since the multimodal behavior effectively increases the clutter

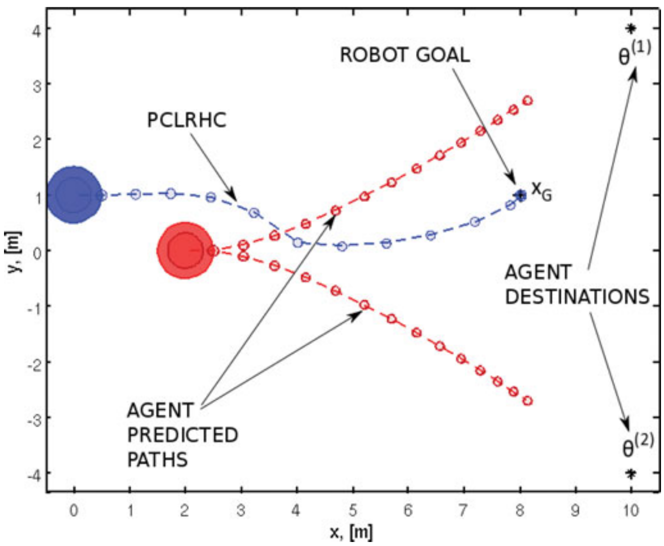

Fig. 11. Initially planned PCLRHC trajectory and the predicted agent trajectories towards the two possible destinations

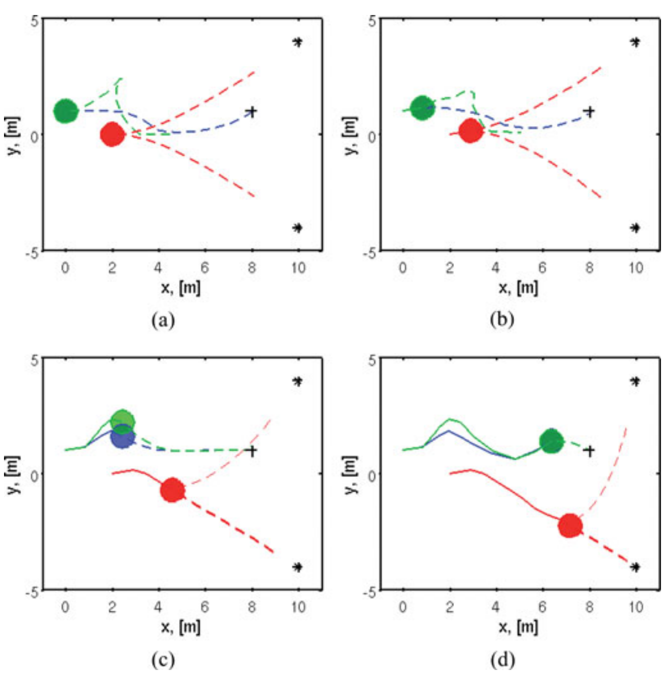

Fig. 12. Executed (solid) and planned or predicted (dashed) trajectories at stages 1, 2, 5, and 10 (thicker lines indicate more probable behaviors).

at future times. Since the actual destination is disambiguated as more information is obtained during system execution, the false agent trajectory is eventually ignored. The PCLRHC approach is better able to avoid the agent when both destinations are likely and ignores the false destination once the true destination becomes known.

2) Example 6 (Agents With Multiple Models): Systems with multiple model classes are of interest since complex behaviors can be obtained by combining simpler behavioral models. For example, the ability to distinguish between an agent moving at a constant velocity from one that is moving toward some goal location allows for better prediction and planning. Such scenarios are detailed in [44].

C. Interactive Robot–Agent Scenarios

This section considers motion planning when there are different kinds of coupling between the robot and agent models. To handle this interdependence, the problem is formulated in an augmented state space: $x_i = [(x_i^R)^T \; (x_i^{A1})^T \; \ldots \; (x_i^{An_A})^T]^T$. Similar augmented spaces are used for u_i, ω_i, y_i, and ν_i.

1) Example 7 (Agent Information Gathering): It is desirable to model the quality of information that can be obtained about the agents. In this example, the measurement quality is a function of agent distance from the robot: Better observations are obtained as the robot nears the agent. We use again the robotic system, cost, and constraints of Section VII-A2. Assume the agent dynamic model of Appendix A1, and a distance-dependent measurement model (see Appendix B2 with $d_{\max} = 5$, $n_\xi = 2$, and $\xi_i^{(p)} \sim \mathcal{N}(0,1)\ \forall p = 1, 2$). This *augmented system* has state-dependent noise (see, e.g., [48])

$$x_i = Ax_{i-1} + Bu_{i-1} + F\omega_{i-1} \tag{38}$$

$$y_i = Cx_i + H\nu_i + \sum_{p=1}^{n_\xi} \overline{H}^{(p)} \xi_i^{(p)} + \sum_{p=1}^{n_\xi} G^{(p)} x_i \xi_i^{(p)}. \tag{39}$$

Note that the multiplicative noise terms are the products of two Gaussian variables, which is non-Gaussian. Therefore, an optimal estimator is not generally available. An approximate estimator can be derived by assuming a Luenberger estimator:

$$\hat{x}_{i|i} = \hat{x}_{i|i-1} + K_i(y_i - \hat{y}_{i|i-1}). \tag{40}$$

The resulting two-step filter is given by [44]

Prediction step:

$$\hat{x}_{i|i-1} = A\hat{x}_{i-1|i-1} + Bu_{i-1} \tag{41}$$

$$\Sigma_{i|i-1} = A\Sigma_{i-1|i-1}A^T + FWF^T. \tag{42}$$

Measurement update step:

$$\hat{x}_{i|i} = \hat{x}_{i|i-1} + K_i(y_i - C\hat{x}_{i|i-1}) \tag{43}$$

$$\Sigma_{i|i} = (I - K_iC)\Sigma_{i|i-1} \tag{44}$$

where

$$\begin{aligned}\Gamma_{i|i-1} = {} & C\Sigma_{i|i-1}C^T + HVH^T + \sum_{l=1}^{n_\xi} \sigma_\xi^2 \overline{H}^{(l)} \overline{H}^{(l)T} \\ & + \sum_{l=1}^{n_\xi} \sigma_\xi^2 G^{(l)} \left(\Sigma_{i|i-1} + \hat{x}_{i|i-1}\hat{x}_{i|i-1}^T\right) G^{(l)T} \\ & + \sum_{l=1}^{n_\xi} \sigma_\xi^2 \left(\overline{H}^{(l)} \hat{x}_{i|i-1}^T G^{(l)T} + G^{(l)} \hat{x}_{i|i-1} \overline{H}^{(l)T}\right)\end{aligned} \tag{45}$$

$$K_i = \Sigma_{i|i-1}C^T\Gamma_{i|i-1}^{-1}. \tag{46}$$

The dependence between the robot and agent models must be accounted for in the chance constraints (see [36] and [44] for additional details).

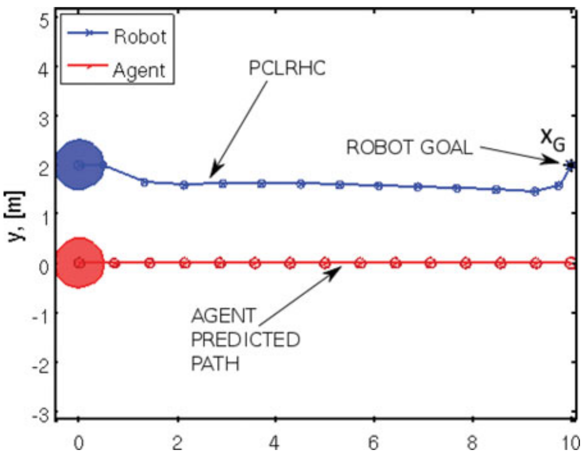

Fig. 13. PCLRHC's planned robot trajectory and predicted agent trajectory. The robot moves closer to the agent to improve information gathering quality.

In the simulations, the robot's goal is $x_G = [10\ 2\ 0\ 0]^T$. The mean initial states of robot and agent are $\hat{x}^R_{0|0} = [0\ 2\ 1\ 0]^T$ and $\hat{x}^A_{0|0} = [1\ 0\ 1.2\ 0]^T$. A quadratic cost function (33) of the *augmented* state and control is used. Unlike previous examples, the robot's plan affects the quality of agent measurement, and therefore, the resulting quality of future agent state estimates. As a result, the estimation and the planning processes are not separable, and the covariance terms in the cost function cannot be ignored. Here, $Q_N = \text{diag}(10, 10, 0, 0, 100, 100, 0, 0)$, $Q_i = \text{diag}(1, 1, 0, 0, 100, 100, 0, 0)$, and $R_i = \text{diag}(0.1, 0.1, 0, 0)$, for all $i = 0, \ldots, N-1$ are used in (33). The planned PCLRHC trajectory is shown in Fig. 13. The agent's position uncertainty is heavily penalized in the cost function in order to accentuate the active learning component of the plan. The plan moves the robot toward the agent to obtain more accurate measurements and, therefore, better information about the agent.

2) Example 8 (Adversarial Agent Model): This example considers an *adversarial* agent model which continually attempts to collide with the robot, causing the robot to actively avoid the agent. Again, the robotic system, cost, and constraints of Section VII-A2 are assumed. The robot's goal is $x_G = [10\ 0\ 0\ 0]$, and its mean initial state is $\hat{x}^R_{0|0} = [0\ 0\ 1\ 0]^T$. To complicate the planning task, the robot's velocity components are limited to $[-1.6\ 1.6]$ so that it cannot easily outrun the agent. For the agent, assume the adversarial model (see Appendix A3 with $k = 0.15$) and a linear position measurement model (see Appendix B1). The agent initial mean state is $\hat{x}^A_{0|0} = [5\ 3\ 0\ -1]^T$.

As shown in Fig. 14, the OLRHC method is initially unable to plan past the agent because of the growth in uncertainty. However, as new information is incorporated (by recursively resolving the problem), the robot can move toward the goal. The sequence of executed plans and planned trajectories is given in Fig. 15. The PCLRHC's planned trajectory is plotted in Fig. 16. The reduction in conservatism allows the PCLRHC approach to plan past the agent and move toward the goal. The robot's motion is more aggressive but still probabilistically safe.

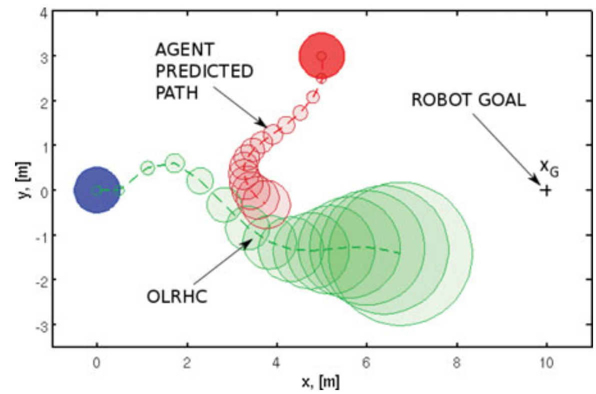

Fig. 14. OLRHC's planned trajectory for the robot (with robot velocity constraints $[-1.6\ 1.6]$) and the predicted trajectory for the adversarial agent.

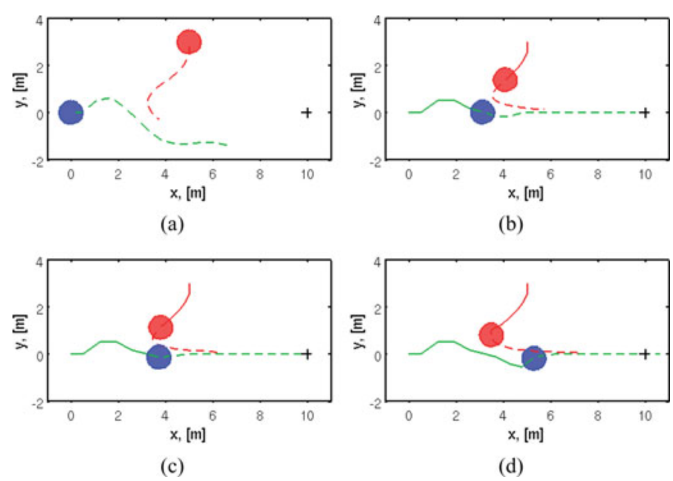

Fig. 15. OLRHC's executed and planned trajectories at stages 1, 6, 7, and 10 with an adversarial agent model (with robot velocity constraints $[-1.6\ 1.6]$).

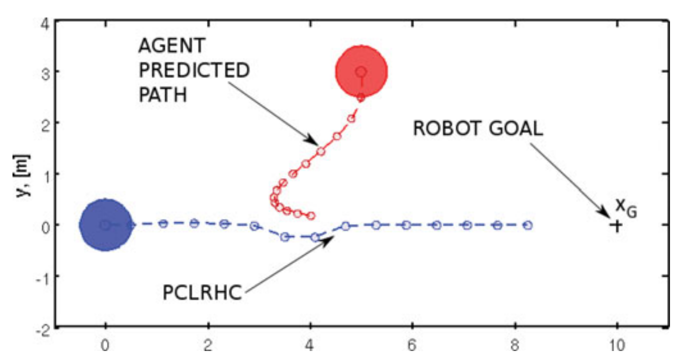

Fig. 16. PCLRHC's planned trajectories for the robot and the predicted trajectory for the adversarial agent (with robot velocity constraints $[-1.6\ 1.6]$).

What happens when the robot cannot physically avoid the dynamic agents? This case was briefly investigated by limiting the robot velocity components to $[-1.2\ 1.2]$. The OLRHC approach was unable to find a path toward the goal, as the adversarial agent can move sufficiently close to the robot to violate the chance constraints for all possible control actions. However, a feasible solution was found by the PCLRHC method. Snapshots from the sequence of executed and planned OLRHC trajectories are

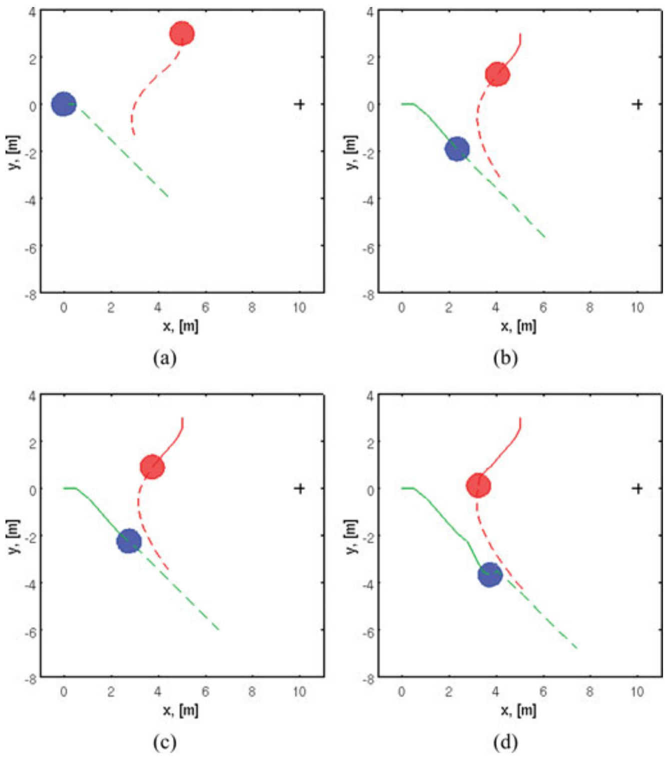

Fig. 17. OLRHC's executed and planned trajectories at stages 1, 6 7, and 10 with an adversarial agent model (with robot velocity constraints $[-1.2\ 1.2]$.) Note that the robot never reaches the goal.

given in Fig. 17. This is an example where the OLRHC approach cannot solve the problem, but the PCLRHC is still able to find a safe and efficient solution (not plotted). It should be noted that the PCLRHC approach will eventually fail when the robot velocity is sufficiently constrained.

3) Example 9 (Friendly Agent Model): It is easy to postulate problems where the robot must rely on the agent cooperation in order to successfully complete a mission. Such scenarios are detailed in [44].

VIII. Conclusion and Future Work

A complete strategy to solve motion planning problems in DUEs has been lacking. 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. The ultimate solution must integrate estimation, prediction, planning, and complex models of agent behavior. This paper took some initial steps toward building such a framework. Because the classical SDP does not readily incorporate collision constraints and exact SDP solutions are often intractable, we developed a PCLRHC strategy for this problem. Our approach was motivated by the desire to account for anticipated future information in the planning process. This way, we are better able to manage the growth of system uncertainty in the prediction component of the DUE solution, as the anticipated future information reduces the uncertainty that is associated with future belief states. Previous planning approaches are hampered by the growth in uncertainty associated with future belief states, leading to potentially constrained and conservative plans. Since the future belief states better represent the belief states realized during execution, the planned and executed PCLRHC paths are much closer, 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 of a robot navigating in static and simple dynamic environments highlighted the improvement in plan quality for the partially closed-loop approach, compared with an open-loop approach where all future information is ignored.

We showed by way of several examples that the PCLRHC approach can also be applied in more complex dynamic scenarios that include multiple possible agent destinations and/or multiple simple agent models. The PCLRHC framework allows for integrated parameters estimation and model selection of these complex multimodal distributions, which is key to efficiently incorporating complex agent behavior into the planning and execution process. Simulations demonstrated the potential benefit of the PCLRHC approach over standard approaches in these dynamic scenarios. These simulation results also demonstrated that the robot can adjust its plan to obtain better information about the dynamic agents. The PCLRHC approach is also better able to avoid adversarial agents, even with reduced dynamic capabilities, than the OLRHC approach.

The effects of system models, objective functions, uncertainty, and clutter in the environment on the computational complexity of the DUE problem are discussed. It is shown that the increase in the computational burden from the OLRHC to the PCLRHC approach is small. DUE problems are nonconvex, resulting in local minima in the solution space. An inherently parallelizable sampling-based approach (e.g., the expansive space tree [4]) may offer a practical implementation since it is less prone to local minima and have been successfully applied in high-dimensional spaces and will be investigated in future work.

Appendix A
Dynamic Models

1. Random Walk Model

The object's dynamics are governed by

$$x_i = Ax_{i-1} + Bu_{i-1} + F\omega_{i-1} \tag{47}$$

where state x consists of the planar position and velocity. Let $\mathcal{I}_2$ be as earlier and $\mathcal{O}_2$ be a 2×2 zero matrix. Then

$$A = \begin{bmatrix} \mathcal{I}_2 & \Delta t\, \mathcal{I}_2 \\ \mathcal{O}_2 & \mathcal{I}_2 \end{bmatrix},\ B = \begin{bmatrix} \mathcal{O}_2 \\ \mathcal{I}_2 \end{bmatrix},\ \text{and } F = \begin{bmatrix} \mathcal{O}_2 \\ \mathcal{I}_2 \end{bmatrix}$$

and $\Delta t = 0.5$ s. The process noise is independent with a Gaussian distribution, $\omega_{i-1} \sim \mathcal{N}(0, W)$.

2. Known Destination Model

The agent is attracted to a destination via a spring–mass–damper potential function, which has unit mass and is critically damped. Let k be the spring stiffness:

$$x_i^A = A x_{i-1}^A + N\theta + F\omega_{i-1}^A \tag{48}$$

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

$$A = \begin{bmatrix} \mathcal{I}_2 & \Delta t\, \mathcal{I}_2 \\ -k\Delta t\, \mathcal{I}_2 & (1 - 2\sqrt{k}\Delta t)\, \mathcal{I}_2 \end{bmatrix}, \; N = \begin{bmatrix} \mathcal{O}_2 \\ k\Delta t\, \mathcal{I}_2 \end{bmatrix}.$$

F is the same as for the random walk model (see Appendix A1).

3. Adversarial Agent Model

The agent is drawn to the robot's current position, which is based on a critically damped spring–mass–damper system with unit mass. The resulting dynamic equation has the form

$$x_i^A = A x_{i-1}^A + A_{RA} x_{i-1}^R + F\omega_{i-1}^A \tag{49}$$

where $\omega_{i-1}^A \sim \mathcal{N}(0, W_A)$ is white, Gaussian noise, A and F are as in Appendix A2 (with spring stiffness k), and

$$A_{RA} = \begin{bmatrix} \mathcal{O}_2 & \mathcal{O}_2 \\ k\Delta t\, \mathcal{I}_2 & \mathcal{O}_2 \end{bmatrix}.$$

Appendix B
Measurement Models

1. Linear Position Measurement Model

Assume that the measurement is a scaled, noisy subset of the system state:

$$y_i = C x_i + H\nu_i \tag{50}$$

where $C = [\mathcal{I}_2\ \mathcal{O}_2]$, $H = \mathcal{I}_2$, and with independent, Gaussian-distributed measurement noise, $\nu_i \sim \mathcal{N}(0, V)$.

2. Distance-Dependent Measurement Quality Model

Assume a measurement model with state-dependent noise:

$$y_i^A = C_A x_i^A + H_A \nu_i^A + \sum_{l=1}^{n_\xi} \Phi_l g(x_i^A, x_i^R)\xi_i^{(l)} \tag{51}$$

where $\nu_i^A \sim \mathcal{N}(0, V_A)$, and $\xi_i^{(l)} \sim \mathcal{N}(0, \sigma_\xi^2)\ \forall l = 1, \ldots, n_\xi$. $g(x_i^A, x_i^R)$ is some nonlinear function of the robot and agent states (e.g., distance between the objects). Φ_l is a matrix that aligns the lth state-dependent noise term with the appropriate component of the measurement.

Assume $g(x_i^A, x_i^R) = \|x_{i|i-1}^R - x_{i|i-1}^A\|_2^2$ so that the measurement quality decreases as the distance between the objects increase. Linearize this function around the conditional means of the robot and agent states [44]. Let $\tilde{x}_i \triangleq (\hat{x}_{i|i-1}^R - \hat{x}_{i|i-1}^A)$:

$$y_i^A = C_A x_i^A + H_A \nu_i^A + \sum_{l=1}^{n_\xi} \overline{H}_A^{(l)} \xi_i^{(l)} + \sum_{l=1}^{n_\xi} G_A^{(l)} x_i^A \xi_i^{(l)} + \sum_{l=1}^{n_\xi} G_R^{(l)} x_i^R \xi_i^{(l)}. \tag{52}$$

where

$$\overline{H}_A^{(l)} = \begin{cases} -\Phi_l \tilde{x}_i^T \tilde{x}_i, & \text{if } E\left[g(x_i^R, x_i^A)|I_{i-1}^{OL}\right] \le d_{\max}^2 \\ \Phi_l d_{\max}^2, & \text{otherwise} \end{cases}$$

$$G_A^{(l)} = \begin{cases} -2\Phi_l \tilde{x}_i^T, & \text{if } E\left[g(x_i^R, x_i^A)|I_{i-1}^{OL}\right] \le d_{\max}^2 \\ 0, & \text{otherwise} \end{cases}$$

$$G_R^{(l)} = \begin{cases} 2\Phi_l \tilde{x}_i^T, & \text{if } E\left[g(x_i^R, x_i^A)|I_{i-1}^{OL}\right] \le d_{\max}^2 \\ 0, & \text{otherwise.} \end{cases}$$

References

[1] Defense Advanced Res. Projects Agency, Urban Challenge. (2011). [Online]. Available: http://archive.darpa.mil/grandchallenge/

[2] S. Tadokoro, M. Hayashi, Y. Manabe, Y. Nakami, and T. Takamori, "On motion planning of mobile robots which coexist and cooperate with humans," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 1995, vol. 2, pp. 518–523.

[3] N. Roy, G. Gordon, and S. Thrun, "Planning under uncertainty for reliable health care robotics," in *Proc. Int. Conf. Field Serv. Robot.*, 2003, pp. 417–426.

[4] S. M. LaValle, *Planning Algorithms*. Cambridge, U.K.: Cambridge Univ. Press, 2006.

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

[6] P. Fiorini and Z. Shiller, "Motion planning in dynamic environments using velocity obstacles," *Int. J. Robot. Res.*, vol. 17, no. 7, pp. 760–772, 1998.

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

[8] A. Clodic, V. Montreuil, R. Alami, and R. Chatila, "A decisional framework for autonomous robots interacting with humans," in *Proc. IEEE Int. Workshop Robot Human Interact. Commun.*, 2005, pp. 543–548.

[9] B. Xu, D. Stilwell, and A. Kurdila, "A receding horizon controller for motion planning in the presence of moving obstacles," in *Prof. IEEE Int. Conf. Robot. Autom.*, 2010, pp. 974–980.

[10] C. Fulgenzi, A. Spalanzani, and C. Laugier, "Dynamic obstacle avoidance in uncertain environment combining PVOs and occupancy grid," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2007, pp. 1610–1616.

[11] Y. Kuwata, T. Schouwenaars, A. Richards, and J. How, "Robust constrained receding horizon control for trajectory planning," in *Proc. AIAA Guid., Navigat., Control Conf.*, 2005, pp. 1–12.

[12] J. Carson, "Robust model predictive control with a reactive safety mode," Ph.D. dissertation, Controls Dynamical Syst., Calif. Inst. Technol., Pasadena, 2008.

[13] T. Lozano-Perez, M. Mason, and R. H. Taylor, "Automatic synthesis of fine-motion strategies for robots," *Int. J. Robot. Res.*, vol. 3, no. 1, pp. 3–24, 1984.

[14] A. Lambert and D. Gruyer, "Safe path planning in an uncertain-configuration space," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2003, vol. 3, pp. 4185–4190.

[15] R. Pepy and A. Lambert, "Safe path planning in an uncertain-configuration space using RRT," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2006, pp. 5376–5381.

[16] J. Gonzalez and A. Stentz, "Planning with uncertainty in position using high-resolution maps," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2007, pp. 1015–1022.

[17] A. Censi, D. Calisi, A. De Luca, and G. Oriolo, "A Bayesian framework for optimal motion planning with uncertainty," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2008, pp. 1798–1805.

[18] S. Prentice and N. Roy, "The belief roadmap: Efficient planning in belief space by factoring the covariance," *Int. J. Robot. Res.*, vol. 28, no. 11–12, pp. 1448–1465, Nov./Dec. 2009.

[19] R. Alterovitz and T. Simon, "The stochastic motion roadmap: A sampling framework for planning with Markov motion uncertainty," in *Proc. Int. Conf. Robot. Sci. Syst.*, 2007, pp. 246–253.

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

[21] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. Cambridge, MA: MIT Press, 2005.

[22] 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, 2005.
[23] D. van Hessem and O. Bosgra, "Closed-loop stochastic model predictive control in a receding horizon implementation on a continuous polymerization reactor example," in *Proc. Amer. Control Conf.*, 2004, vol. 1, pp. 914–919.
[24] L. Blackmore, "Robust path planning and feedback design under stochastic uncertainty," *AIAA Guid., Navigat. Control Conf. Exhib.*, Honolulu, HI, 2008.
[25] R. He, A. Bachrach, and N. Roy, "Efficient planning under uncertainty for a target-tracking micro-aerial vehicle," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2010, pp. 1–8.
[26] B. Kluge and E. Prassler, "Reflective navigation: individual behaviors and group behaviors," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2004, vol. 4, pp. 4172–4177.
[27] J. van den Berg, M. Lin, and D. Manocha, "Reciprocal velocity obstacles for real-time multi-agent navigation," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2008, pp. 1928–1935.
[28] F. Large, D. Vasquez, T. Fraichard, and C. Laugier, "Avoiding cars and pedestrians using velocity obstacles and motion prediction," in *IEEE Intell. Veh. Symp.*, 2004, pp. 375–379.
[29] Q. Zhu, "Hidden Markov model for dynamic obstacle avoidance of mobile robot navigation," *IEEE Trans. Robot. Autom.*, vol. 7, no. 3, pp. 390–397, Jun. 1991.
[30] H. Helble and S. Cameron, "3-D path planning and target trajectory prediction for the Oxford aerial tracking system," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2007, pp. 1042–1048.
[31] M. Bennewitz, W. Burgard, G. Cielniak, and S. Thrun, "Learning motion patterns of people for compliant robot motion," *Int. J. Robot. Res.*, vol. 24, pp. 31–48, 2005.
[32] S. Thompson, T. Horiuchi, and S. Kagami, "A probabilistic model of human motion and navigation intent for mobile robot path planning," in *Proc. Int. Conf. Auton. Robots Agents*, 2009, pp. 663–668.
[33] A. Foka and P. Trahanias, "Predictive control of robot velocity to avoid obstacles in dynamic environments," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2003, vol. 1, pp. 370–375.
[34] P. Henry, C. Vollmer, and D. Fox, "Learning to navigate through crowded environments," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2010, pp. 981–986.
[35] N. E. Du Toit and J. W. Burdick, "Robotic motion planning in dynamic, cluttered, uncertain environments," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2010, pp. 966–973.
[36] N. E. Du Toit and J. W. Burdick, "Probabilistic collision checking with chance constraints," *IEEE Trans. Robot*, vol. 27, no. 4, pp. 809–815, Aug. 2011.
[37] D. P. Bertsekas, *Dynamic Programming and Optimal Control*, vol. 2, 3rd ed. Belmont, MA: Athena Scientific, 2005.
[38] D. Mayne, J. Rawlings, C. Rao, and P. Scokaert, "Constrained model predictive control: stability and optimality," *Automatica*, vol. 36, no. 26, pp. 789–814, 2000.
[39] M. Morari and J. H. Lee, "Model predictive control: Past, present and future," *Comput. Chem. Eng.*, vol. 23, pp. 667–682, 1999.
[40] A. T. Schwarm and M. Nikolaou, "Chance-constrained model predictive control," *AIChE J.*, vol. 45, no. 8, pp. 1743–1752, 1999.
[41] D. Li, F. Qian, and P. Fu, "Variance minimization approach for a class of dual control problems," *Automat. Control*, vol. 47, no. 12, pp. 2010–2020, 2002.
[42] 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," in *Proc. IEEE Conf. Decis. Control*, 2003, vol. 1, pp. 929–934.
[43] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, 2nd ed. New York: Wiley, 2006.
[44] N. E. Du Toit, "Robot motion planning in dynamic, cluttered, uncertain environments," Ph.D. dissertation, Dept. Mech. Eng., California Inst. Technol., Pasadena, 2010.
[45] T. Fraichard and H. Asama, "Inevitable collision states: A step towards safer robots?" *Adv. Robot.*, vol. 18, no. 10, p. 1001, 2004.
[46] A. Bautin, L. Martinez-Gomez, and T. Fraichard, "Inevitable collision states: A probabilistic perspective," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2010, pp. 4022–4027.
[47] D. Althoff, M. Althoff, D. Wollherr, and M. Buss, "Probabilistic collision state checker for crowded environments," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2010, pp. 1492–1498.
[48] E. Todorov, "Stochastic optimal control and estimation methods adapted to the noise characteristics of the sensorimotor system," *Neural Comput.*, vol. 17, no. 5, pp. 1084–1108, 2005.

Noel du Toit (M'10) received the B.Eng. degree in mechanical engineering (*cum laude*) from the University of Stellenbosch, South Africa, in 2001, the M.S. degree in aeronautical engineering from the Massachusetts Institute of Technology, Cambridge, in 2005, and the Ph.D. degree in mechanical engineering from the California Institute of Technology, Pasadena, in 2010.

His research is focused on high-level motion planning problems, including motion planning for information, planning with uncertainty, optimal sensing, and situational reasoning. His interests span applications in ground, air, surface, and underwater robotics.

Joel W. Burdick (M'05), photograph and biography not available at the time of publication.