MIT Libraries | DSpace@MIT

MIT Open Access Articles

A probabilistic particle-control approximation of chance-constrained stochastic predictive control

Massachusetts Institute of Technology

A Probabilistic Particle-Control Approximation of Chance-Constrained Stochastic Predictive Control

Lars Blackmore, Masahiro Ono, Askar Bektassov, and Brian C. Williams

Abstract—**Robotic systems need to be able to plan control actions that are robust to the inherent uncertainty in the real world. This uncertainty arises due to uncertain state estimation, disturbances, and modeling errors, as well as stochastic mode transitions such as component failures. Chance-constrained control takes into account uncertainty to ensure that the probability of failure, due to collision with obstacles, for example, is below a given threshold. In this paper, we present a novel method for chance-constrained predictive stochastic control of dynamic systems. The method approximates the distribution of the system state using a finite number of particles. By expressing these particles in terms of the control variables, we are able to approximate the original stochastic control problem as a deterministic one; furthermore, the approximation becomes exact as the number of particles tends to infinity. This method applies to arbitrary noise distributions, and for systems with linear or jump Markov linear dynamics, we show that the approximate problem can be solved using efficient mixed-integer linear-programming techniques. We also introduce an important weighting extension that enables the method to deal with low-probability mode transitions such as failures. We demonstrate in simulation that the new method is able to control an aircraft in turbulence and can control a ground vehicle while being robust to brake failures.**

Index Terms—**Chance constraints, hybrid discrete-continuous systems, nonholonomic motion planning, planning under stochastic uncertainty.**

I. INTRODUCTION

ROBUST control of robotic systems has received a great deal of attention in recent years [12], [36], [39], [40], [47], [53]. Robotic systems need to be able to plan control actions that are robust to the inherent uncertainty in the real world. This uncertainty arises due to uncertain state estimation, disturbances, and modeling errors, as well as stochastic mode transitions such

L. Blackmore is with the Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA 91109 USA (e-mail: lars@jpl.nasa.gov).

M. Ono is with the Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, MA 02139 USA (e-mail: hiro_ono@mit.edu).

A. Bektassov is with Eni E&P, 00144 Rome, Italy (e-mail: askar.bektassov@gmail.com).

B. C. Williams is with the Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, MA 02139 USA (e-mail: williams@mit.edu).

as component failures. Many authors have investigated control under set-bounded uncertainty, for example, [6], [18], [19], [31] and [44]. In this case, robust control ensures that failure is prevented under all possible uncertainties.

In many cases, for example, wind disturbances, uncertainty is best represented using a stochastic model, rather than a set-bounded one [4]. The problem of control under stochastic uncertainty has been researched extensively, for example, [7], [38], [41], and [54]. Early approaches such as linear quadratic Gaussian control [7] used the certainty equivalence principle [3]; this enables uncertain variables to be replaced with their expectations and control laws to be designed in terms of these expectations. For unconstrained problems with linear systems and Gaussian noise, controllers can be designed that are optimal with regard to the original stochastic control problem.

Alternative approaches expressed certain classes of stochastic control problem as a Markov decision process (MDP) [43]. For discrete-state spaces, value iteration can be used to find the control policy that maximizes the expected reward. Continuous-state spaces can be handled by discretization; however, the problem quickly becomes intractable as the number of discrete states grows [27]; hence, the application of this approach to dynamic systems with high-dimensional continuous-state spaces has been limited.

Predictive stochastic control takes into account probabilistic uncertainty in dynamic systems and aims to control the predicted distribution of the system state in some optimal manner over a finite planning horizon [28], [41], [43], [57]. This is a very challenging problem, since we must optimize over the space of possible future state distributions. Furthermore, with stochastic uncertainty, it is typically not possible to prevent failure in all possible cases. Instead, previous authors have proposed a *chance-constrained* formulation [41], [50]. Chance constraints specify that the probability of failure must be below a given threshold. Failure can be defined as collision either with an obstacle or failure to reach a goal region. This chance-constrained formulation is a powerful one as it enables the user to specify a desired level of conservatism, which can be traded against performance. In this paper, we are concerned with the problem of *chance-constrained* predictive control under stochastic uncertainty.

Recent work has considered chance-constrained predictive control of linear systems when all forms of uncertainty are additive and Gaussian, and the feasible region is convex. A number of authors suggested approaches that pose the stochastic control problem as an equivalent deterministic optimization problem [5], [12], [25], [32], [50].

For many problems, the assumptions of additive Gaussian noise and convex feasible regions do not apply. In this paper, we describe a new approach that relaxes these assumptions and applies to a more general class of system than previous approaches. The key idea behind the new approach is to approximate all probability distributions using a finite set of samples or *particles*. We then approximate the stochastic predictive control problem as a deterministic one, with the property that as the number of particles tends to infinity, the approximation tends to the original stochastic problem. This method can handle arbitrary, even multimodal distributions, and, in principle, can deal with nonlinear systems. However, we show that in the case of stochastic linear systems with uncertain parameters and jump Markov linear systems (JMLSs), the resulting optimization problem can be solved efficiently using mixed-integer linear programming (MILP) [26]. JMLSs are convenient tools for representing robotic systems subject to random component failures [13], [23], [56]. The new method designs optimal control inputs that ensure chance constraints are satisfied despite both component failures and continuous uncertainty, such as disturbances.

The idea of approximating stochastic control problems and stochastic optimization problems using sampling was previously proposed by a number of authors. These approaches use samples to approximate the expectation of a cost function, thus yielding a deterministic optimization problem that minimizes the approximated expectation. Samples are variously referred to as "particles" [21], [24] "simulations" [52], and "scenarios" [14], [36], [55]. In the Pegasus system, Ng and Jordan [36] propose the conversion of an MDP with stochastic transitions into an equivalent one, where all randomness is represented in the initial state. The method draws a finite number of samples, or scenarios, from the initial state and finds the policy that optimizes the sample mean of the reward for the scenarios. In a similar manner, Greenfield and Brockwell [24] proposed a method for predictive control that approximates the expected cost of a given control sequence using a finite number of samples. Andrieu *et al.* [2] used particles to approximate the value function and its gradient in an optimal stochastic control problem, and used a gradient search method to find a locally optimal solution. In the case of optimal sensor scheduling, Singh *et al.* [52] employed a similar approximation method, but in this case used stochastic approximation to minimize the cost function. This approach is guaranteed to converge to a local optimum under certain assumptions. Tarim *et al.* [55] introduced a scenario-based approach to stochastic constraint programming. This approach handles chance-constrained problems but is, however, limited to discrete decision spaces. Vidyasagar [58] considered sample-based approximations of robust controller synthesis problems, while Shapiro [51] proposed sample-based approximations of multistage stochastic optimization problems, where certain decisions can be made after the realizations of some uncertain variables are observed. In both [58] and [51], an approximation of the expected cost is minimized, and analytic results are given that describe how many samples are required to guarantee, in a probabilistic sense, a particular level of approximation error. Nesterov [35] considers single-stage

stochastic-optimization problems in which the goal is to minimize expected cost and proposes a method that draws samples at each step of an iterative optimization scheme. Again, analytic characterizations of the approximation error introduced by sampling are provided.

In this paper, we extend this research in four ways. First, we incorporate constraints on the probability of failure, enabling chance-constrained stochastic control with continuous decision variables. Control with constraints on the probability of events is a powerful capability, which is also a much more challenging problem than control with only constraints on expected values, for example. Second, we show that in the case of stochastic linear dynamic systems and JMLSs, the approximated chance-constrained control problem can be solved to *global* optimality using MILP in an efficient manner. Third, we use importance weighting to ensure that low-probability mode transitions such as failures are handled in an efficient manner. Finally, we can handle nonconvex feasible regions.

Our approach was first published in [9] and [11]. Concurrently with this, Calafiore and Campi [14] introduced a sampling approach to solve chance-constrained *feedback* control problems for convex feasible regions; they use a bounding approach to determine how many times the constraints must be sampled to ensure chance constraint satisfaction. This approach can be applied to chance-constrained *predictive* control problems; however, as noted by Campi and Calafiore [17], the bounding approach introduces conservatism, and is restricted to convex problems. Conservatism means that the true probability of constraint violation is below the allowable level specified by the chance constraint. Excessive conservatism can lead to unnecessarily high-cost solutions and infeasibility. In Section IX, we provide an empirical comparison of our approach with that of [14].

Another approach for chance-constrained control with non-Gaussian uncertainty was previously proposed by Lecchini-Visintini *et al.* [29], and uses a Monte–Carlo Markov chain (MCMC) framework [45] to find an approximately optimal control input through simulation-based optimization. This works by estimating the distribution of a random variable, which is constructed so that the peaks of the distribution coincide with the optimal decision value. The estimation process is then carried out by MCMC [45]. Analytic results on the convergence of MCMC are given by Lecchini-Visintini *et al.* [30] and Rosenthal [46]. Unlike our proposed approach based on MILP, MCMC does not rely on linear system dynamics, and can handle cases where the distribution of the random variables depends on the system state. However, for robotic control problems, the MCMC approach has the disadvantage of being sensitive to manually chosen parameters such as the burn-in period and the "peakedness" parameter [29]. In Section VIII, we provide an empirical comparison of our approach with that of Lecchini-Visintini *et al.* [29].

We demonstrate our new method in simulation using two scenarios. First, we consider control of an unmanned aircraft subject to turbulence and uncertain localization. Second, we consider control of a wheeled robotic vehicle with failure-prone brakes. The results show that the method is effective in solving

the approximated stochastic control problem and that for a large number of particles, the approximation error becomes small.

II. PROBLEM STATEMENT

In this paper, we consider a discrete-time dynamic system with state $\mathbf{x} \in \mathbb{R}^{n_x}$, control inputs $\mathbf{u} \in \mathbb{R}^{n_u}$, and model parameters $\theta \in \mathbb{R}^{n_\theta}$. Disturbances $\nu \in \mathbb{R}^{n_x}$ act on the system. The future states of the system are defined by the following functions:

$$\mathbf{x}_1 = f_1(\mathbf{x}_0, \mathbf{u}_0, \nu_0, \theta_0)$$
$$\mathbf{x}_2 = f_2(\mathbf{x}_0, \mathbf{u}_0, \mathbf{u}_1, \nu_0, \nu_1, \theta_0, \theta_1)$$
$$\vdots$$
$$\mathbf{x}_T = f_T(\mathbf{x}_0, \mathbf{u}_{0:T-1}, \nu_{0:T-1}, \theta_{0:T-1}). \qquad (1)$$

We use $\mathbf{x}_t$ to denote the value of variable $\mathbf{x}$ at time t, and use $\mathbf{x}_{1:T}$ to denote the sequence $\langle \mathbf{x}_1, \ldots, \mathbf{x}_T \rangle$.

We consider the case where the initial state, model parameters, and disturbances are uncertain, but are modeled as random variables. In this case, the future states are also random variables, whose distributions depend on the control inputs. We are concerned with the following optimal, chance-constrained stochastic control problem.

Problem 1 (Chance-constrained control problem):

Minimize:
$$E[h(\mathbf{u}_{0:T-1}, \mathbf{x}_{1:T})]$$

subject to:
$$p(\mathbf{x}_{1:T} \notin F) \leq \delta$$
$$E[\mathbf{x}_{1:T}] \in G$$
$$\mathbf{u}_{0:T-1} \in U$$

where $h(\cdot)$ is a cost function defined over the control inputs and system state trajectory, F is an operator-defined feasible region for the system state trajectory, G is an operator-defined feasible region for the *expected* state trajectory, and U is an operator-defined feasible region for the control inputs. In other words, the problem under consideration is to design a finite, optimal sequence of control inputs, taking into account probabilistic uncertainty, which ensures that the state of the system leaves a given feasible region F with probability at most δ, and keeps the expected state within another feasible region G. In the case of vehicle path planning, F can be defined so that the system state is in a goal region at the final time step and avoids a number of obstacles at all time steps. G can be defined so that the expected velocity is identically zero at the final time. Optimality can be defined in terms of either minimizing control effort or time to reach the goal, for example. Note that we impose a *joint* chance constraint, i.e., one over the multidimensional variable $\mathbf{x}_{1:T}$. This is in contrast to prior work that considered chance constraints over scalar variables, for example, [15] and [16].

We consider the following three sources of uncertainty.

1) The initial state is specified as a probabilistic distribution over possible states. This uncertainty arises because the

system state cannot be observed directly; hence, we must estimate a distribution over the system state from noisy observations.

2) Disturbances act on the system state. These are modeled as stochastic processes. In the case of aircraft path planning, this may represent wind or turbulence disturbances.

3) The system parameters are not known exactly. This may arise either due to approximate modeling (through linearization, for example) or because of random jumps. These jumps can model component failures, for example.

We assume that the distributions of the uncertainty mentioned here are known. We make *no assumptions* about the form the distributions take, except that the random variables are independent of the control inputs. We also assume that we can generate samples from the uncertain distributions. For notational simplicity, in the following, we do not consider uncertainty in the feasible region F. The extension to this form of uncertainty is straightforward.

The key idea behind solving this stochastic control problem is to approximate all distributions using samples, or particles, and then, solve the resulting deterministic problem. In Section III, we review some results relating to sampling from random variables. In Section IV, we then describe the new approach in detail.

III. SAMPLING FROM RANDOM VARIABLES

Previous work has shown that approximating the probability distribution of a random variable using samples, or *particles*, can lead to tractable algorithms for estimation and control [21]. Here, we review some properties of samples drawn from random variables.

Suppose that we are given two multivariate probability distributions $p(\mathbf{x})$ and $q(\mathbf{x})$. We would like to calculate properties of the *target distribution* $p(\mathbf{x})$ such as the expectation

$$E_X[f(X)] = \int f(\mathbf{x})p(\mathbf{x})d\mathbf{x}. \qquad (2)$$

In many cases, this integral cannot be evaluated in closed form. Instead, we approximate the value by drawing N independent, identically distributed random samples $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N)}$ from the *proposal distribution* $q(\mathbf{x})$, and calculate the weighted sample mean

$$\hat{E}_X[f(X)] = \frac{1}{N} \sum_{i=1}^{N} w_i f(\mathbf{x}^{(i)}), \qquad w_i = \frac{p(\mathbf{x}^{(i)})}{q(\mathbf{x}^{(i)})} \qquad (3)$$

where w_i is known as the *importance weight*. As long as $q(\mathbf{x}) > 0$ for all $\mathbf{x}$, such that $p(\mathbf{x}) > 0$, and under weak assumptions on the boundedness of $f(\cdot)$ and the moments of $p(\mathbf{x})$, from the strong law of large numbers [8], we have the convergence property, i.e.,

$$\hat{E}_X[f(X)] \longrightarrow E_X[f(X)] \text{ as } N \longrightarrow \infty. \qquad (4)$$

Hence, the expectation, which could not be evaluated exactly in closed form, can be approximated as a summation over a finite number of particles. In addition, note that we have approximated the expectation over the target distribution by drawing samples from the proposal distribution. In the simplest case, we can set

$p(\mathbf{x}) = q(\mathbf{x})$ and sample directly from the target distribution; this is known as *fair sampling*. However, prior work has shown that other choices of $q(\mathbf{x})$ can significantly improve algorithm performance. We elaborate on this in Section VI.

The convergence property (4) can also be used to approximate the probability of a certain event, such as the event $f(\mathbf{x}) \in A$. This is exactly given by

$$P_A = \int_{f(\mathbf{x}) \in A} p(\mathbf{x}) d\mathbf{x}. \tag{5}$$

This expression is equivalent to the expectation

$$P_A = E_X[g(\mathbf{x})] \quad g(\mathbf{x}) = \begin{cases} 0, & f(\mathbf{x}) \in A \\ 1, & f(\mathbf{x}) \notin A. \end{cases} \tag{6}$$

We can therefore approximate P_A as

$$\hat{P}_A = \frac{1}{N} \sum_{i=1}^{N} w_i g(\mathbf{x}^{(i)}). \tag{7}$$

Note that this expression is simply the weighted fraction of particles for which $f(\mathbf{x}^{(i)}) \in A$. We assume that the evaluation of $f(\cdot)$ and to check whether a given value is in A are both straightforward, and also is the calculation of $\hat{P}_A$; we simply need to count how many of the propagated particles $f(\mathbf{x}^{(i)})$ fall within A. By contrast, the evaluation of P_A as in (5) requires a finite integral over an arbitrary probability distribution, where even the calculation of the bounds on the integral may be intractable. Hence, the particle-based approximation is extremely useful, especially given the convergence property

$$\hat{P}_A \longrightarrow P_A \quad \text{as} \quad N \longrightarrow \infty \tag{8}$$

again, given assumptions on the boundedness of $f(\cdot)$ and the moments of $p(\mathbf{x})$ [8]. In Section IV, we use this property to approximate the stochastic control problem defined in Section II.

IV. OUTLINE OF CHANCE-CONSTRAINED PARTICLE CONTROL METHOD

In this section, we describe a new method for approximating the general chance-constrained stochastic control problem described in Section II. The new method works by approximating all probabilistic distributions with particles, thereby approximating an intractable stochastic optimization problem as a tractable deterministic optimization problem. By solving this deterministic problem, we obtain an approximate solution to the original stochastic problem (problem 1), with the additional property that as the number of particles used tends to infinity, the approximation becomes exact. The new chance-constrained particle control method is given in Table I and illustrated in Fig. 1.

This general formulation can encompass a very broad range of chance-constrained problems. In the algorithm outline, we have omitted two key parts: first, how to perform the resulting optimization efficiently and, second, how to choose the proposal distribution $q(\cdot)$. In Section V, we show that in the case of linear and JMLSs and polygonal feasible regions, the optimization can be solved using efficient MILP methods. In Section VI, we discuss the choice of proposal distribution and design a

TABLE I
CHANCE-CONSTRAINED PARTICLE CONTROL ALGORITHM

1) Generate N samples from the proposal distribution $q(\mathbf{x}_0, \nu_{0:T-1}, \theta_{0:T-1})$ defined over the initial state, disturbances and model parameters. Calculate the weight w_i for each sample according to (3).

2) Express the distribution of the future state trajectories approximately as a set of N particles. Each particle $\mathbf{x}_{1:T}^{(i)}$ corresponds to the state trajectory given a particular set of samples $\{\mathbf{x}_0^{(i)}, \nu_{0:T-1}^{(i)}, \nu_{0:T-1}^{(i)}\}$, and *depends explicitly on the control inputs* $\mathbf{u}_{0:T-1}$, which are yet to be generated.

$$\mathbf{x}_{1:T}^{(i)} = \begin{bmatrix} \mathbf{x}_1^{(i)} \\ \mathbf{x}_2^{(i)} \\ \vdots \\ \mathbf{x}_T^{(i)} \end{bmatrix}, \quad \mathbf{x}_t^{(i)} = f_t(\mathbf{x}_0^{(i)}, \mathbf{u}_{0:t-1}, \nu_{0:t-1}^{(i)}, \theta_{0:t-1}^{(i)}). \tag{9}$$

3) Approximate the chance constraints in terms of the generated particles. Using the result in (8) the probability of failure is approximated as follows:

$$p(\mathbf{x}_{1:T} \notin F) = \int_{\mathbf{x}_{1:T} \notin F} p(\mathbf{x}_{1:T}) d\mathbf{x}_{1:T} \approx \frac{1}{N} \sum_{i=1}^{N} w_i g(\mathbf{x}_{1:T}^{(i)}),$$

where $g(\cdot)$ is defined in (6). The approximated chance constraint is then:

$$\frac{1}{N} \sum_{i=1}^{N} w_i g(\mathbf{x}_{1:T}^{(i)}) \leq \delta. \tag{10}$$

In other words, a weighted fraction of no more than δ of the particles can fall outside of the feasible region. Note that a particle represents a state *trajectory* over the entire planning horizon.

4) Approximate the constraints on the expected state using the sample mean approximation. For example,

$$E[\mathbf{x}_{1:T}] \in G \quad \text{becomes} \quad \frac{1}{N} \sum_{i=1}^{N} w_i \mathbf{x}_{1:T}^{(i)} \in G. \tag{11}$$

5) Approximate the expected cost in terms of particles

$$\hat{h} \triangleq \frac{1}{N} \sum_{i=1}^{N} w_i h(\mathbf{u}_{0:T-1}, \mathbf{x}_{1:T}^{(i)}) \approx E[h(\mathbf{u}_{0:T-1}, \mathbf{x}_{1:T})]. \tag{12}$$

6) Solve deterministic constrained optimization problem:
Minimize $\hat{h}$ over control inputs $\mathbf{u}_{0:T-1} \in U$ *subject to*

$$\frac{1}{N} \sum_{i=1}^{N} w_i g(\mathbf{x}_{1:T}^{(i)}) \leq \delta \quad \frac{1}{N} \sum_{i=1}^{N} w_i \mathbf{x}_{1:T}^{(i)} \in G. \tag{13}$$

proposal to improve the performance with low-probability fault transitions.

V. MIXED-INTEGER LINEAR PROGRAMMING FORMULATION OF PARTICLE CONTROL PROBLEM

A. Linear Systems

We first restrict our attention to the case of uncertain linear system dynamics and polygonal feasible regions. Furthermore, we assume that the cost function h is piecewise linear. Previous
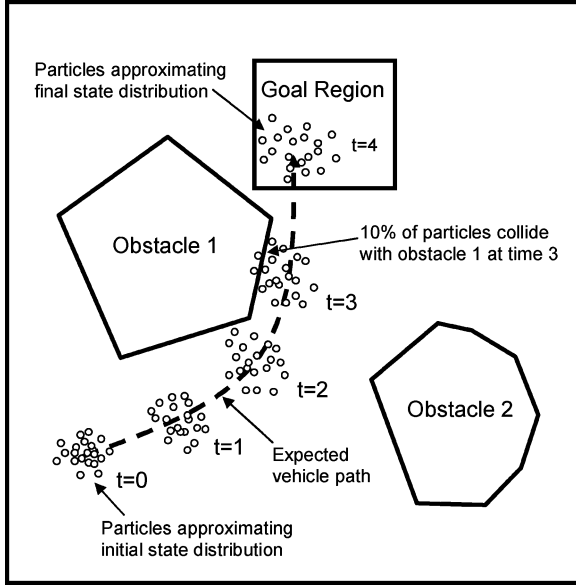
Fig. 1. Particle-control method in a vehicle path-planning scenario. The feasible region is defined so that the plan is successful if the vehicle avoids the obstacles at all time steps and is in the goal region at the final time step. The objective is to find the optimal sequence of control inputs so that the plan is successful with a probability of at least 0.9. The particle-control method approximates this so that at most, 10% of the particles fail. In this example, all weights w_i are 1.

work has shown that optimal path planning with obstacles for vehicles, such as aircraft or satellites, can be posed as predictive control design for linear systems in polygonal feasible regions [49], [44]. Optimality can be defined in terms of fuel use or time, for example. We extend this research by showing that the particle control method outlined in Section IV can be used to design control inputs for linear systems that ensure chance constraints are satisfied under probabilistic uncertainty in the initial state and disturbances.

We consider the linear discrete-time system model

$$\mathbf{x}_{t+1} = A(\theta_t)\mathbf{x}_t + B(\theta_t)\mathbf{u}_t + \nu_t \qquad (14)$$

where $A(\theta_t)$ indicates that the matrix A is a function of the parameters θ_t. Substituting this system model into (9), we obtain the following equation for $\mathbf{x}_t^{(i)}$:

$$\mathbf{x}_t^{(i)} = \sum_{j=0}^{t-1}\left(\prod_{l=2}^{t-j}A(\theta_{l-1}^{(i)})\right)\left(B(\theta_j^{(i)})\mathbf{u}_j + \nu_j^{(i)}\right) + \prod_{l=0}^{t-1}A(\theta_l^{(i)})\mathbf{x}_0^{(i)}.$$

$$(15)$$

Note that this is a linear function of the control inputs, and that $\mathbf{x}_0^{(i)}$, $\nu_j^{(i)}$, and $\theta_j^{(i)}$ are all known values, as generated in step 1 of Table I. Hence, each particle $\mathbf{x}_{1:T}^{(i)}$ is a known linear function of the control inputs. Furthermore, the sample mean of the particles is a known linear function of the control inputs.

To impose the approximate chance constraint (10), we need to constrain the weighted fraction of particles that fall outside of the feasible region. To do so, we first define a set of N binary variables $z_1,\ldots,z_N$, where $z_i \in \{0,1\}$. These binary variables are defined so that $z_i = 0$ implies that particle i falls inside the feasible region. We then constrain the weighted sum of these
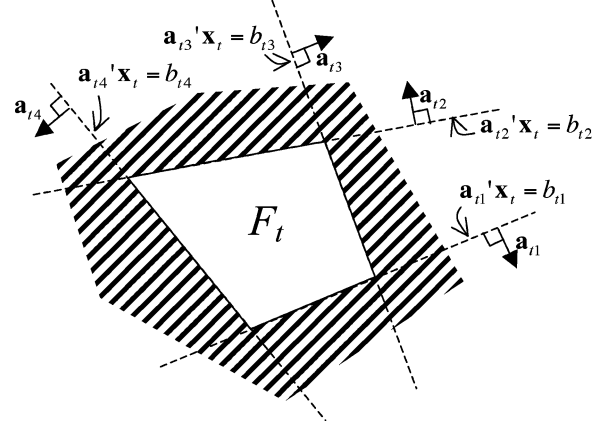

Fig. 2. Two-dimensional convex polygonal feasible region F_t defined for the state at time t. The vectors $\mathbf{a}_{t1},\ldots,\mathbf{a}_{tN_t}$ are the unit outward normals to the N_t line segments that define the feasible region.

binary variables, i.e.,

$$\frac{1}{N}\sum_{i=1}^{N}w_i z_i \leq \delta. \qquad (16)$$

This constraint ensures that the weighted fraction of particles falling outside of the feasible region is at most δ. We now describe how to impose constraints, such that

$$z_i = 0 \implies \mathbf{x}_{1:T}^{(i)} \in F \qquad (17)$$

first for convex polygonal feasible regions and then for nonconvex polygonal feasible regions.

1) Convex Feasible Regions: A convex polygonal feasible region F defined for the state trajectory $\mathbf{x}_{1:T}$ is defined as a conjunction of convex polygonal constraints F_t for each time step $1,\ldots,T$, such that

$$\mathbf{x}_{1:T} \in F \iff \bigwedge_{t=1,\ldots,T} \mathbf{x}_t \in F_t. \qquad (18)$$

In turn, the polygonal feasible region F_t is defined as a conjunction of linear constraints $\mathbf{a}'_{tl}\mathbf{x}_t \leq b_{tl}$ for $l = 1,\ldots,N_t$, where $\mathbf{a}_{tl}$ is defined as pointing outward from the polygonal region. Then, as illustrated in Fig. 2, $\mathbf{x}_t$ lies within F_t if and only if all of the constraints are satisfied, i.e.,

$$\mathbf{x}_t \in F_t \iff \bigwedge_{l=1,\ldots,N_t} \mathbf{a}'_{tl}\mathbf{x}_t \leq b_{tl}. \qquad (19)$$

We now use "Big M" techniques to ensure that the constraint (17) is satisfied. We impose the following constraints:

$$\mathbf{a}'_{tl}\mathbf{x}_t^{(i)} - b_{tl} \leq Mz_i \quad \forall t \ \forall l \qquad (20)$$

where M is a large positive constant. A value of $z_i = 0$ implies that, for particle i, every constraint is satisfied for every time step, whereas for large enough M, a value of $z_i = 1$ leaves particle i unconstrained. We therefore have

$$z_i = 0 \implies \{\mathbf{x}_t^{(i)} \in F_t \ \forall t\} \implies \mathbf{x}_{1:T}^{(i)} \in F \qquad (21)$$
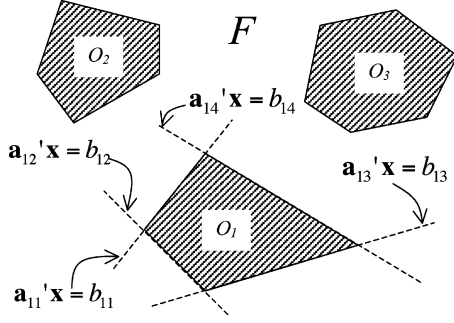
as required.

Fig. 3. Two-dimensional nonconvex polygonal feasible region F. The feasible region is the complement of several convex obstacles (shaded). Each obstacle O_j is defined by the N_j vector normals $\mathbf{a}_{j1}, \ldots, \mathbf{a}_{jN_j}$.

2) Nonconvex Feasible Regions: Predictive control within a nonconvex feasible region is a much more challenging problem than control within a convex feasible region [44]. However, as shown by Schouwenaars *et al.* [49], vehicle path planning with obstacles can be posed as such a problem; hence, it is of great interest.

A polygonal nonconvex feasible region can, in general, be defined as the complement of L polygonal infeasible regions, as shown in Fig. 3. In other words, the state trajectory $\mathbf{x}_{1:T}$ is in the feasible region if and only if all obstacles are avoided for all time steps. As noted by Schouwenaars *et al.* [49], avoidance of a polygonal obstacle can be expressed in terms of a disjunction of linear constraints, i.e., the system state at time t avoids obstacle O_j, defined as shown in Fig. 3, if and only if

$$\bigvee_{l=1,\ldots,N_j} \mathbf{a}'_{jl}\mathbf{x}_t \geq b_{jl}. \tag{22}$$

In a similar manner to [49], we introduce binary variables $d_{ijtl} \in \{0,1\}$ that indicate whether a given constraint l for a given obstacle O_j is satisfied by a given particle i at a given time step t. The constraint

$$\mathbf{a}'_{jl}\mathbf{x}_t^{(i)} - b_{jl} + Md_{ijtl} \geq 0 \tag{23}$$

means that $d_{ijtl} = 0$ implies that constraint l in obstacle O_j is satisfied by particle i at time step t. Again, M is a large positive constant. We now introduce binary variables $e_{ijt} \in \{0,1\}$ that indicate whether a given obstacle O_j is avoided by a given particle i at a given time step t. The constraint

$$\sum_{l=1}^{N_j} d_{ijtl} - (N_j - 1) \leq Me_{ijt} \tag{24}$$

ensures that $e_{ijt} = 0$ implies that at least one constraint in obstacle O_j is satisfied by particle i at time step t. This, in turn, implies that obstacle O_j is avoided by particle i at time step t.

Next, we introduce binary variables $g_{ij} \in \{0,1\}$ that indicate whether a given obstacle O_j is avoided by particle i at *all* time steps. The constraints

$$\sum_{t=1}^{T} e_{ijt} \leq Mg_{ij}, \qquad \sum_{j=1}^{L} g_{ij} \leq Mz_i \tag{25}$$

ensure that $g_{ij} = 0$ implies that obstacle j is avoided at all time steps by particle i, and that $z_i = 0$ implies that all obstacles are avoided at all time steps by particle i. Hence, for nonconvex feasible regions F, we have

$$z_i = 0 \implies \mathbf{x}_{1:T}^{(i)} \in F \tag{26}$$

as required.

B. Particle Control for Jump Markov Linear Systems

A special case of the linear dynamics (14) that is of particular interest for robotics is the JMLS [56], [23], [13], [20]. A JMLS is a system whose model parameters θ take on values from a finite set. In this case, θ_t is referred to as the discrete state, or *mode*, and $\mathbf{x}_t$ is referred to as the continuous state; hence, a JMLS is a form of a hybrid discrete-continuous system. Without loss of generality, we assume that θ_t is an integer in the range $1, \ldots, n_\theta$. The dynamics of a JMLS are defined as

$$\mathbf{x}_{t+1} = A(\theta_t)\mathbf{x}_t + B(\theta_t)\mathbf{u}_t + \nu_t \tag{27}$$

where θ_t is a Markov chain that evolves according to a transition matrix $T \in \mathbb{R}^{n_\theta \times n_\theta}$, such that

$$p(\theta_{t+1} = j | \theta_t = i) = T_{ij} \tag{28}$$

where T_{ij} denotes the (i,j)th element of matrix T. The variable ν_t is a disturbance process whose distribution can take any form but is assumed to be known.

In order to apply the particle control approach of Section IV to JMLS, we sample from discrete mode sequences, initial continuous state and continuous disturbances according to a proposal distribution $q(\mathbf{x}_0, \nu_{0:T-1}, \theta_{0:T-1})$. Given a discrete mode sequence and samples for all of the uncertain continuous variables, the future system state trajectory is a known deterministic function of the control inputs. Now, each particle provides a sample of the future *hybrid discrete-continuous* state trajectory as a function of the control inputs. Since JMLSs are a special case of the uncertain linear dynamics (14), each particle is a linear function of the control inputs.

C. Expected State Constraints

In step 4 of Table I, we approximate the expected state constraints as constraints on the sample mean of the particles. Since the sample mean is a linear function of the control inputs, for a polygonal feasible regions G, these constraints can be posed as mixed integer linear constraints using "Big M" techniques that follow a similar development to that in Section V-A. For the sake of brevity, we omit the details here, but see [10].

D. Control Constraints

Since the controls $\mathbf{u}_{0:T-1}$ are deterministic, the constraint $\mathbf{u}_{0:T-1} \in U$ can be handled without approximation using standard approaches for predictive control. For polygonal U, this results to mixed-integer linear constraints; for details, see [10].

E. Cost Function

The optimal, chance-constrained control problem requires that we minimize the expected cost $E[h]$. The true expectation is given by

$$E[h] = \int h(\mathbf{u}_{0:T-1}, \mathbf{x}_{1:T})p(\mathbf{x}_{1:T})d\mathbf{x}_{1:T}. \quad (29)$$

Since $p(\mathbf{x}_{1:T})$ can be an arbitrary distribution, this integral is intractable in most cases. We therefore approximate the expectation using the sample mean as in (3), i.e.,

$$\hat{h} = \frac{1}{N}\sum_{i=1}^{N} w_i h(\mathbf{u}_{0:T-1}, \mathbf{x}_{1:T}^{(i)}). \quad (30)$$

This expression can be evaluated without integration, and converges to the true expectation as the number of particles approaches infinity. Furthermore, since we assume that h is a piecewise linear function of the state and control inputs, the expression for $\hat{h}$ in (30) is also piecewise linear.

F. Summary of Mixed-Integer Linear-Programming Formulation

To summarize, the approximated stochastic predictive control problem defined in Section IV defines a constrained optimization problem. If the proposal distribution $q(\cdot)$ is chosen so that the weights w_i do not depend on the control inputs, and the system to control is either linear or jump Markov linear, we can express the approximated chance constraint (10) and the approximated expectation constraint (11) using linear constraints on the control inputs. These constraints involve integer variables that indicate whether a particular particle stays within the feasible region. Furthermore, the approximated cost function (30) is piecewise linear in the control inputs. Hence, the chance-constrained particle control problem can be posed as a mixed-integer linear program. The resulting optimization finds the best sequence of control inputs, such that, at most, a weighted fraction δ of the particles falls outside of the feasible region. This weighted fraction approximates the probability of the future state trajectory falling outside of the feasible region, and as the number of particles tends to infinity, the approximation becomes exact. The optimality criterion and constraints on the expected state are also approximated in terms of particles, and the approximation becomes exact as the number of particles tends to infinity. While MILPs are worst-case exponential, in the average case, they can be solved very quickly using commercially available software [26], as we demonstrate in Section VII.

VI. CHOOSING A PROPOSAL DISTRIBUTION

We now consider the problem of choosing a proposal distribution $q(\cdot)$. A considerable amount of work in estimation has shown that the "best" proposal distribution depends heavily on the particular problem being addressed, and a great deal of work has focussed on developing proposal distributions for specific applications, for example, [22], [23] and [34]. This applies equally to control problems.

A proposal distribution must meet the following criteria. First, we must ensure that $q(\cdot) > 0$ wherever $p(\cdot) > 0$. Second, for the optimization to be posed as an MILP, $p(\cdot)/q(\cdot)$ cannot be a function of the control inputs $\mathbf{u}$. Third, we must be able to sample from $q(\cdot)$.

These criteria are all met by the "fair" proposal distribution $q(\cdot) = p(\cdot)$. In Section VII-A and B, we show empirically that the fair proposal distribution is effective in two different scenarios that have only continuous dynamics. This is, however, not the case for hybrid systems with low-probability mode transitions such as failures. We demonstrate this empirically in Section VII-C. The key insight is that these discrete jumps in the system dynamics have a large effect on the optimal, chance-constrained control strategy, compared with the probability of such jumps occurring. In this section, we design a proposal distribution that addresses this problem.

We consider proposal distributions of the following factored form:

$$q(\mathbf{x}_0, \nu_{0:T-1}, \theta_{0:T-1}) = p(\mathbf{x}_0, \nu_{0:T-1}|\theta_{0:T-1})q(\theta_{0:T-1}). \quad (31)$$

In other words, we use the fair distribution over initial state and disturbances but use a proposal distribution $q(\theta_{0:T-1})$ defined over mode sequences. To sample from this distribution, we first generate samples of the mode sequence from $q(\theta_{0:T-1})$, and for each sample $\theta_{0:T-1}^{(i)}$, we generate samples of $\mathbf{x}_0$ and $\nu_{0:T-1}$ from their true joint distribution. With this factorization, the importance weight has a particularly simple form, i.e.,

$$w_i = \frac{p(\mathbf{x}_0^{(i)}, \nu_{0:T-1}^{(i)}|\theta_{0:T-1}^{(i)})p(\theta_{0:T-1}^{(i)})}{p(\mathbf{x}_0^{(i)}, \nu_{0:T-1}^{(i)}|\theta_{0:T-1}^{(i)})q(\theta_{0:T-1}^{(i)})} = \frac{p(\theta_{0:T-1}^{(i)})}{q(\theta_{0:T-1}^{(i)})}. \quad (32)$$

We must now choose $q(\theta_{0:T-1})$ in order to address the problem of low-probability mode sequences. In order to motivate our choice of $q(\theta_{0:T-1}^{(i)})$, we present two unsuitable candidates. First, consider a fair sampling approach:

$$q(\theta_{0:T-1}) = p(\theta_{0:T-1}). \quad (33)$$

With this proposal, low-probability mode sequences are rarely sampled. Such sequences include those where a particular component, such as the brakes on a car, has transitioned into a "failed" mode. In this case, the control sequence generated by the particle control approach is strongly dependent on whether a sequence with the "failed" mode is sampled. If no such brake failure is sampled, the algorithm will not design a control sequence that is robust to such failures. The proposal (33) is unsuitable, because it yields a high probability that no fault transitions will be sampled.

Next, consider a proposal equal to the pseudouniform distribution $q(\theta_{0:T-1}) = U(\theta_{0:T-1})$, where $U(\cdot)$ assigns an equal probability to each mode sequence for which $p(\theta_{0:T-1}) > 0$. More precisely

$$U(\theta_{0:T-1}) = \begin{cases} 1/n_p, & p(\theta_{0:T-1}) > 0 \\ 0, & p(\theta_{0:T-1}) = 0 \end{cases} \quad (34)$$

where n_p is the number of mode sequences for which $p(\theta_{0:T-1}) > 0$. The use of this proposal ensures that sequences involving faults are sampled with the same likelihood as the

mode sequence without failures (which, in reality, has much higher probability). This solves the problem of failure sequences being sampled too infrequently. We assume for now that there is only one mode sequence without failures, which we refer to as the *nominal* mode sequence $\theta_{0:T-1}^{\mathrm{nom}}$. The drawback in using this proposal is that there is a significant likelihood that the nominal mode sequence is not sampled. If this occurs, the deterministic optimization will typically be infeasible; achieving most control tasks requires a nonzero probability of the system components operating nominally.

To address the problem of low-probability sequences, we instead use the following "failure-robust" proposal:

$$q^*(\theta_{0:T-1}) = \begin{cases} P_{\mathrm{nom}}, & \theta_{0:T-1} = \theta_{0:T-1}^{\mathrm{nom}} \\ \frac{1-P_{\mathrm{nom}}}{n_p-1}, & \theta_{0:T-1} \neq \theta_{0:T-1}^{\mathrm{nom}} \end{cases} \quad (35)$$

where

$$P_{\mathrm{nom}} = 1 - (1-\lambda)^{1/N} \quad (36)$$

and N is the number of particles. This proposal ensures that the nominal mode sequence is sampled at least once with probability λ, and shares the remaining probability space evenly among the remaining mode sequences. This increases the probability of sampling a failure sequence. In Section VII, we give an empirical analysis that shows that this proposal distribution significantly outperforms the fair proposal distribution (33) when there are low-probability transitions such as failures. Note that, while we have assumed here a single nominal sequence, the extension to multiple nominal mode sequences is straightforward.

VII. RESULTS

In this section, we demonstrate the new particle control approach in simulation using three scenarios. In Section VII-A, the method is used to control an aircraft within a flight envelope in heavy turbulence, while in Section VII-B, the method is used to generate chance-constrained, optimal paths for an aircraft operating in an environment containing obstacles. These scenarios do not consider component failures, and therefore, in Section VII-C, the method is applied to the problem of ground vehicle control with failure-prone brakes.

In each of these examples, the system dynamics are stable; in the case of the aircraft examples, stability is provided by an inner loop controller, while for the ground vehicle, stability is provided by friction. As with all approaches that plan open-loop control actions for stochastic systems, it is important that either the system dynamics are stable or the planning horizon is short relative to the disturbance covariance. If neither of these hold, the system state covariance will typically grow to be too large for there exist a feasible solution that satisfies the chance constraints.

A. Chance-Constrained Predictive Control in a Convex Feasible Region

The new particle control method was used to generate chance-constrained predictive control inputs for an aircraft performing an altitude change maneuver in turbulence. In this scenario, successful execution means that the aircraft remains within a
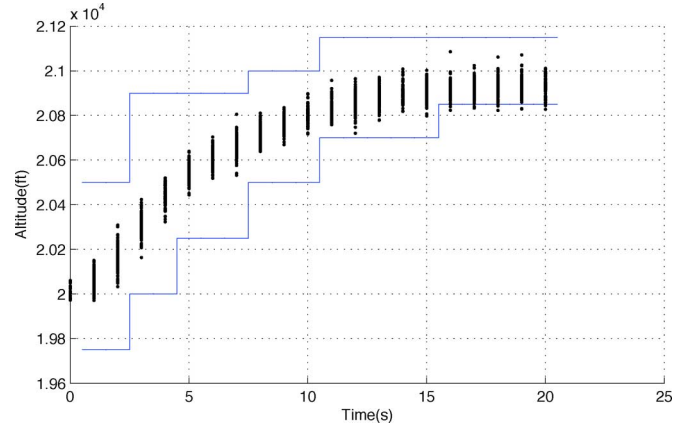

Fig. 4. Path of particles for typical solution to flight envelope scenario. A total of 100 particles were used, and the desired probability of failure is 0.1. Ten particles fall outside of the flight envelope.

defined flight envelope. An example is shown in Fig. 4. Note that the feasible region here is *convex* in the space of state trajectories, and there is a convex constraint on each vector $\mathbf{x}_1, \ldots, \mathbf{x}_T$; hence, the block vector $\mathbf{x}_{1:T}$ is constrained to be in a convex region.

For the aircraft model, we use the linearized longitudinal dynamics of a Boeing 747 traveling at Mach 0.8. Since the angle of attack of most aircraft is low in normal operation, linearizing the dynamics about the equilibrium state or *trim state* of the aircraft yields a good approximation to the true dynamics, which can be used to develop control algorithms [33]. We assume that there is an inner loop controller that issue elevator commands, which is an altitude-hold autopilot. This controller consists of a fixed-gain proportional full-state feedback controller that is defined so that

$$a_t = K(\mathbf{x}_t - \mathbf{x}_t^r), \quad \mathbf{x}_t^r = \begin{bmatrix} 0 & 0 & 0 & 0 & \mathbf{u}_t \end{bmatrix}' \quad (37)$$

where a_t is the elevator angle at time step t, and $\mathbf{u}_t \in \Re$ is the desired altitude setpoint at time step t. The aim of the chance-constrained predictive controller is to issue the altitude setpoint commands $\mathbf{u}_t$ to the autopilot in an optimal manner so that the aircraft breaks the flight envelope with a probability of at most δ. Using the controller in (37), the closed-loop dynamics of the aircraft can be expressed in the discrete-time form of (14) with

$$A = \begin{bmatrix} 0.99 & 0.024 & -1.20 & -27.1 & 4.4\mathrm{E}^{-3} \\ -0.16 & 0.32 & -10.0 & -31.4 & -0.029 \\ 2\mathrm{E}-6 & 1\mathrm{E}-6 & 0.013 & 0.011 & 1.5\mathrm{E}^{-5} \\ -1.8\mathrm{E}^{-4} & 1.7\mathrm{E}^{-4} & -4.4\mathrm{E}^{-3} & -0.156 & -1.3\mathrm{E}^{-4} \\ 0.032 & -1.04 & 20.3 & 1.02\mathrm{E}^3 & 0.9 \end{bmatrix} \quad (38)$$

and

$$B = \begin{bmatrix} -4.48\mathrm{E}^{-3} \\ 0.030 \\ -1.5\mathrm{E}^{-5} \\ 1.28\mathrm{E}^{-4} \\ 0.079 \end{bmatrix} \quad (39)$$

where time increments of 2 s were used in the discretization, and we have used $a\mathrm{E}^b$ to denote $a \times 10^b$. We use a planning horizon of 20 increments.

Disturbances due to turbulence have been studied extensively, and are modeled as stochastic noise processes that are far from Gaussian [4]. In this section, the process noise ν_t is drawn from the Dryden turbulence model described in Military Specification MIL-F-8785C [1]. We assume heavy turbulence, as defined in MIL-F-8785C, with a low-altitude wind velocity of 25 m/s.

Optimality is defined in terms of fuel consumption, and we assume the following relationship between fuel consumption h and elevator angle at time t, i.e., a_t,

$$h = \sum_{t=0}^{T-1} |a_t|. \qquad (40)$$

Since we assume that an inner loop controller issues elevator angle commands, a_t depends on the disturbances that act on the aircraft; for example, if a fixed altitude is commanded by the predictive controller, the autopilot will issue larger elevator commands in order to reject large disturbances than for smaller ones. Since the disturbances are stochastic, we cannot directly optimize the fuel consumption defined in (40). Therefore, we instead optimize the expected fuel consumption $E[h]$.

We impose a maximum elevator deflection of 0.5 rad, which models actuator saturation. Again, since elevator deflection depends on stochastic disturbances, we cannot prevent actuator saturation with absolute certainty. We instead define a chance constraint, which, approximated using the particle control method, ensures that actuator saturation occurs with at most a given probability. In the results shown here, we define this probability to be zero, thereby ensuring that saturation occurs with approximately zero probability. We do not explicitly model nonlinearities due to saturation, instead considering the plan to have failed if saturation occurs.

The initial state distribution was generated using a particle filter. The particle filter tracked the system state for ten time steps, leading up to time $t = 0$, while the aircraft held altitude. Observations of pitch rate and altitude were made subject to additive Rayleigh noise [42]. This non-Gaussian noise means that a particle filter will typically outperform a Kalman filter [21]. The number of particles used for estimation was the same as that used for control.

Fig. 4 shows a typical solution generated by the particle control algorithm for 100 particles. Here, we use the fair proposal distribution $q(\cdot) = p(\cdot)$, i.e., disturbance samples are generated using the MIL-F-8785C turbulence model [1]. The desired probability of failure is 0.1, and ten particles fall outside of the flight envelope.

The true probability of failure for a given plan was estimated using 10^6 random simulations. Since the generated plan depends on the values sampled from the various probability distributions, 40 plans were generated for each scenario. Fig. 5 shows the results for a desired probability of failure of 0.1. It can be seen that the mean probability of failure gets closer to the desired value as the number of particles increases and that the variance decreases. For 200 particles, the approximation is close; the mean is 0.1073, and the standard deviation is 0.0191. Hence,
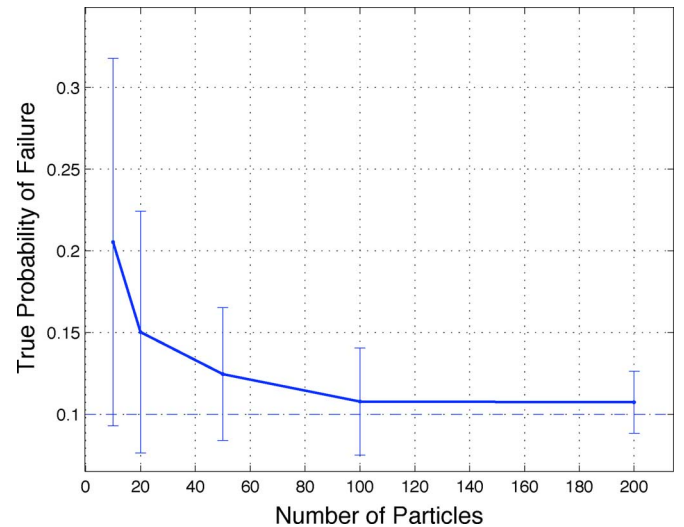

Fig. 5. True probability of failure against number of particles used to design control inputs. The desired probability of failure was 0.1, shown as the dashed line. The results shown are for 40 sets of designed control inputs, with the solid line denoting the mean and the error bars denoting the standard deviation of the probability of failure. As the number of particles increases, the mean probability of failure approaches the desired probability of failure, and the variance decreases.
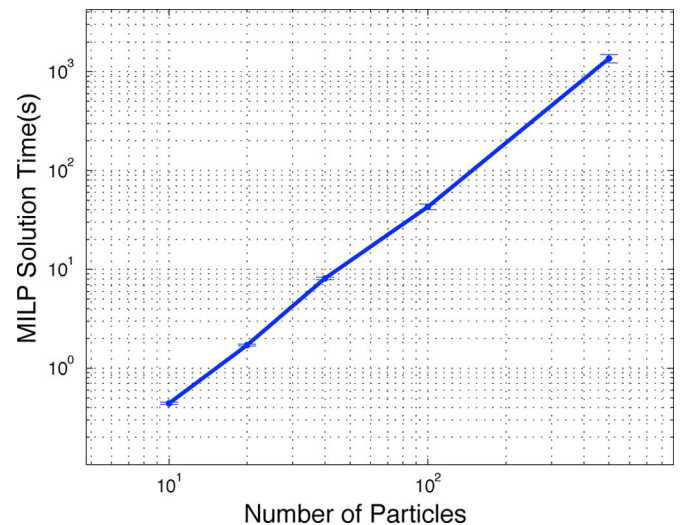

Fig. 6. MILP solution time for particle control with Boeing 747 example. The specified maximum probability of failure was 0.1.

the particle control algorithm can generate optimal solutions to problems that are close to the full stochastic control problem with relatively few particles. Fig. 6 shows the time taken to solve the mixed-integer linear program for this example as a function of the number of particles.

B. Chance-Constrained Vehicle Path Planning With Obstacles

The new particle control method was also applied to an unmanned aerial vehicle (UAV) path-planning scenario with obstacles, wind, and uncertain localization. In this scenario, successful execution means that the UAV is in the goal region at the end of the time horizon and that the UAV avoids all obstacles at all time steps within the horizon.

Previous work [49] has shown that, for the purposes of path planning, an aircraft operating in a 2-D environment can be modeled as a double integrator with velocity and acceleration constraints. This model is based on the assumption that an inner loop autopilot can be used to drive the UAV to a given waypoint, as long as the velocity does not go below a minimum value or above a maximum value, and maximum acceleration levels are not exceeded. Turn-rate constraints are handled conservatively using acceleration magnitude constraints. We use the same aircraft model and assume a maximum aircraft velocity of 50 m/s, time steps of 1 s, and a planning horizon of 10 s. We use a fixed-gain proportional full-state feedback controller for the inner loop autopilot, which is defined such that

$$\mathbf{a}_t = K\left(\mathbf{x}_t - \mathbf{x}_t^r\right), \qquad \mathbf{x}_t^r = \begin{bmatrix} u_{x,t} \\ 0 \\ 0 \\ u_{y,t} \\ 0 \\ 0 \end{bmatrix}, \qquad \mathbf{u}_t \triangleq \begin{bmatrix} u_{x,t} \\ u_{y,t} \end{bmatrix} \tag{41}$$

where $\mathbf{u}_t$ is the waypoint command at time step t, and $\mathbf{a}_t$ is the acceleration vector applied to the double integrator aircraft model at time step t. The aim of the chance-constrained predictive controller is to issue the waypoint commands $\mathbf{u}_t$ to the autopilot in an optimal manner so that the aircraft collides with an obstacle with a probability of at most δ. After time discretization, the closed-loop aircraft dynamics are given in the form of (14) by

$$A = \begin{bmatrix} 1.1 & -4.9 & -6.8 & 0 & 0 & 0 \\ 0.59 & -2.7 & -4.1 & 0 & 0 & 0 \\ -0.26 & 1.7 & 2.8 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1.1 & -4.9 & -6.8 \\ 0 & 0 & 0 & 0.59 & -2.7 & -4.1 \\ 0 & 0 & 0 & -0.26 & 1.7 & 2.8 \end{bmatrix} \tag{42}$$

and

$$B = \begin{bmatrix} 0.57 & 0 \\ 0.053 & 0 \\ -0.044 & 0 \\ 0 & 0.57 \\ 0 & 0.053 \\ 0 & -0.044 \end{bmatrix}. \tag{43}$$

Uncertain disturbances, due to wind, act on the UAV. We use the Dryden wind model with a low-altitude wind speed of 15 m/s and light turbulence, which is as defined in MIL-F-8785C. We assume an inner loop controller that acts to reject disturbances. As described in Section VII-A, uncertainty in localization leads to uncertainty in the initial position of the UAV. The obstacle map used is shown in Fig. 7. Optimality was again defined in terms of fuel consumption, which we assume is related to input acceleration as follows:

$$h = \sum_{t=0}^{T-1}\left\{|a_{x,t}| + |a_{y,t}|\right\} \qquad \mathbf{a}_t \triangleq \begin{bmatrix} a_{x,t} \\ a_{y,t} \end{bmatrix} \tag{44}$$

where $a_{x,t}$ and $a_{y,t}$ are the commanded accelerations at time t in the x- and y-directions, respectively. In order to reduce
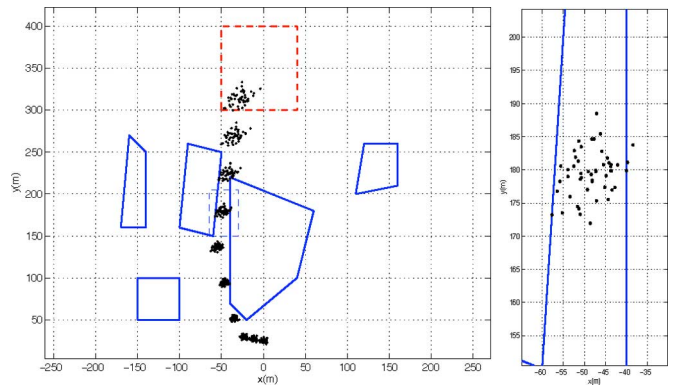

Fig. 7. Path of particles for typical solution to UAV path-planning problem for a probability of failure of 0.04. Obstacles are in blue, while the goal is in red and dashed. At this level of conservatism, the aircraft is able to pass through the narrow corridor. The particle control method ensures that at most two particles out of 50 collide with the obstacles (inset). This solution has a fuel cost of 73.98.
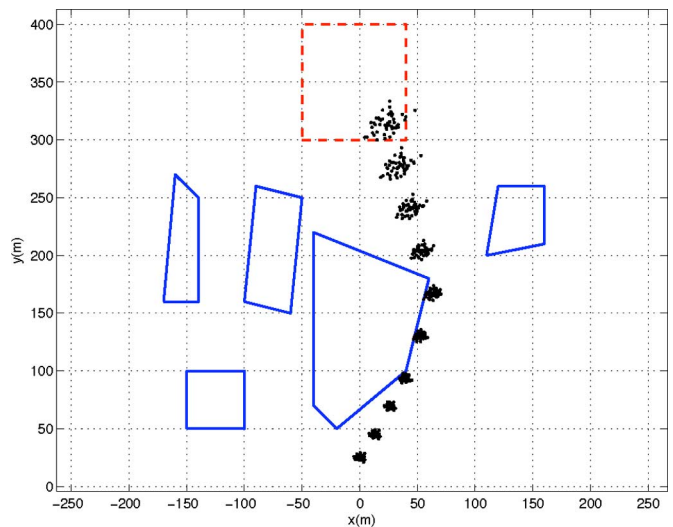

Fig. 8. Path of particles for typical solution to UAV path-planning problem for a probability of failure of 0.02. Obstacles are in blue, while the goal is in red and dashed. At this level of conservatism, the aircraft no longer passes through the narrow corridor but goes around the largest obstacle. This solution has a fuel cost of 74.72. Hence, the reduced probability of failure comes at the expense of fuel.

the solution time of the resulting MILP problem, we employed an iterative deepening technique, which is described in the Appendix. As in Section VII-A, we use a fair proposal distribution.

Results for the scenario are shown in Figs. 7 and 8. Fifty particles were used for these examples. Fig. 7 shows that if a probability of failure of 0.04 or above is acceptable, the planned path of the UAV can go through the narrow corridor at $(-50, 200)$. It can be seen that exactly two particles collide with the obstacles as expected. For a lower probability of failure, however, the planned path is more conservative, as shown in Fig. 8. This path avoids the narrow corridor at the expense of fuel efficiency.
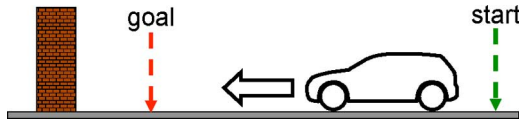
Fig. 9.   Ground-vehicle brake-failure scenario. The expected vehicle position must arrive at the goal in the minimum possible time while avoiding collision with the wall.

C. Chance-Constrained Control With Component Failures

The new particle-control method was applied to a ground-vehicle brake-failure scenario. In this scenario, the wheeled vehicle starts at rest at a distance of 8 m from its goal. The vehicle can apply acceleration and braking inputs. The brakes can fail, however, in which case braking has no effect. The vehicle's expected position, conditioned on nominal brake operation, must arrive at the goal in the minimum possible time. Overall failure of the plan is defined as collision of the vehicle with a wall, which is situated 4 m past the goal. The situation is illustrated in Fig. 9.

The vehicle is modeled as a JMLS with two operational modes, such that $\theta_t \in \{1, 2\}$. In mode 1, $\theta = 1$ and the brakes are functional, while in mode 2, $\theta = 2$ and braking has no effect, i.e.,

$$A(1) = A(2) = \begin{bmatrix} 0.9 & 0 \\ 1 & 1 \end{bmatrix} \qquad B(1) = \begin{bmatrix} 1 & -1 \\ 0 & 0 \end{bmatrix}$$

$$B(2) = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}. \tag{45}$$

The transition matrix, which is as defined in (28), is given by

$$T = \begin{bmatrix} 0.999 & 0.001 \\ 0.0 & 1.0 \end{bmatrix}. \tag{46}$$

The controls issued by the chance-constrained predictive controller are defined by

$$\mathbf{u}_t = \begin{bmatrix} a_{a,t} \\ a_{b,t} \end{bmatrix} \tag{47}$$

where $a_{a,t} \geq 0$ is the commanded acceleration, and $a_{b,t} \geq 0$ is the commanded deceleration due to braking. The switching dynamics in (45) and (46) mean that if the brakes are functional, at every time step, there is a probability of 0.001 that they become faulty. Once faulty, the brakes remain faulty. The brakes are initially functional, and the initial vehicle state is known exactly; however, random acceleration disturbances act on the vehicle. Also, frictional forces proportional to velocity act to decelerate the vehicle. A planning horizon of 20 time steps was used with time intervals of 1 s.

The particle controller must generate control inputs that are robust to both continuous disturbances and the possibility of brake failure. Intuitively, the optimal strategy heavily depends on the desired level of conservatism. A race car driver, who can tolerate a relatively high probability of failure, would accelerate as hard as possible and brake as hard as possible to achieve the minimum time solution. A bus driver, on the other hand, must achieve a probability of failure of close to zero and would
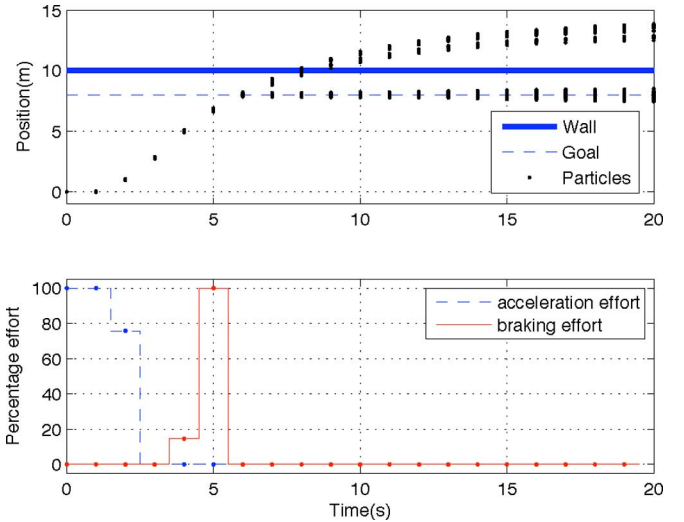

Fig. 10.   Typical solution for a maximum probability of failure of 0.01 with 100 particles. The vehicles arrives at the goal within 6 s but will collide with the wall if a brake failure occurs. This particular solution gives a true probability of failure of approximately 0.006.
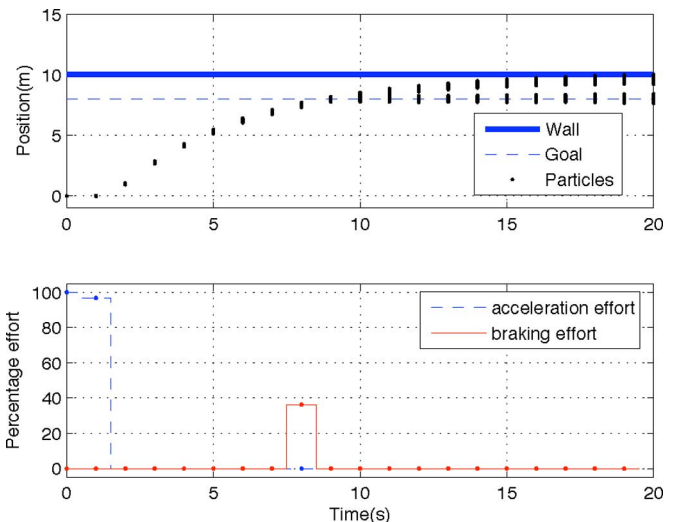

Fig. 11.   Typical solution for a maximum probability of failure of 10^{-6} with 100 particles. The vehicle travels more slowly and arrives within 9 s, which is later than with the more aggressive solution. In the case of brake failure, however, friction brings the vehicle to rest before collision with the wall. This solution is therefore robust to brake failure, giving a probability of failure of approximately 1.0×10^{-6}.

therefore accelerate more slowly and brake more gradually. Both of these strategies are generated by the particle control method. Fig. 10 shows a typical solution generated by the particle control approach for a maximum probability of failure of 0.01 using the proposal distribution (35). Fig. 11 shows a typical solution for a maximum probability of failure of 10^{-6}. The more conservative solution takes 9 s, while the more aggressive solution takes only 6 s.

We now demonstrate that the failure-robust proposal distribution (35) enables the particle controller to take into account the low-probability brake failures. Fig. 12 shows a typical solution generated by using a fair proposal distribution. In this
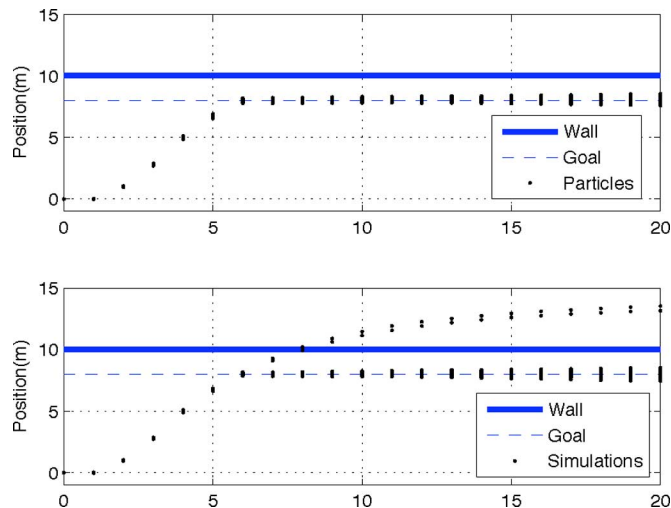
Fig. 12. Typical solution with fair proposal distribution for a maximum probability of failure of 10^{-6} with 100 particles. (Top) Planned particle distribution. Because no particles have sampled the brake failure, the controller plans aggressively. (Bottom) Monte–Carlo simulations of true-state trajectory. In reality, there is a probability of approximately 0.0050 that a brake failure occurs at or before $t = 5$ s, which causes the vehicle to collide with the wall.
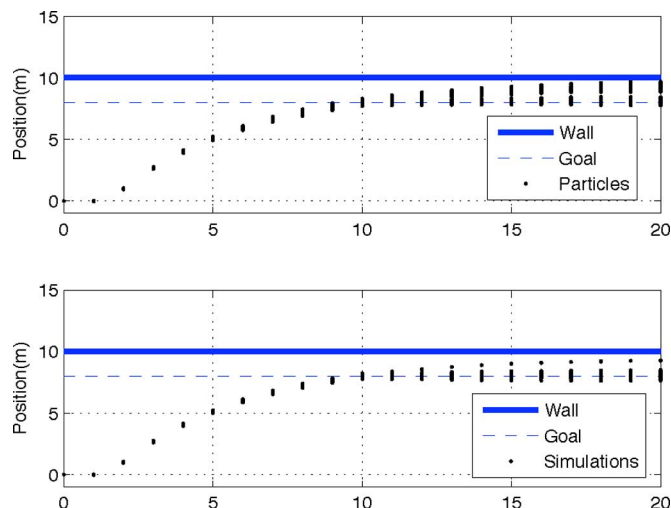

Fig. 13. Typical solution with failure-robust proposal distribution for a maximum probability of failure of 10^{-6}, with 100 particles. (Top) Planned particle distribution (importance weights not shown). Many particles have sampled brake failures, and hence, the controller plans to take brake failures into account. (Bottom) Monte–Carlo simulations of true-state trajectory. The vehicle collides with the wall with a probability of approximately 1.0×10^{-6}.

case, the algorithm did not sample any of the failure transitions and, therefore, has generated an inappropriately aggressive control policy that does not take into account the possibility of brake failure. Fig. 13 shows a typical solution generated by using the failure-robust proposal. By increasing the probability of sampling failure transitions, the importance weighting has taken into account brake failure, thus generating an appropriately conservative plan. Fig. 14 compares the fair proposal distribution approach against the failure-robust proposal in terms of the true probability of failure. In this example, the desired probability of failure was 10^{-6}. The failure-robust proposal achieves a true probability of failure dramatically closer to the desired value than the fair proposal. In addition, note that for larger particle
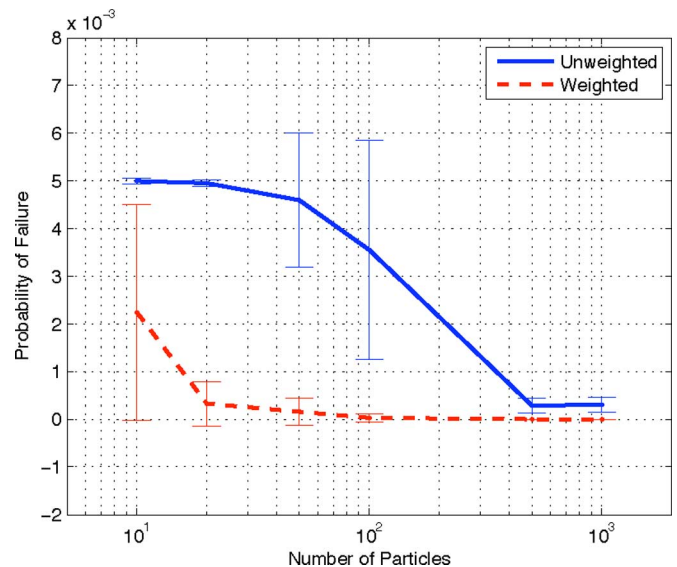

Fig. 14. True probability of failure against number of particles for fair proposal and failure-robust proposal. The desired probability of failure was 10^{-6}. The failure-robust proposal achieves a true probability of failure dramatically closer to the desired value than the fair sampling case. With a very small particle set, the effect of the failure-robust proposal is diminished since the probability of sampling the nominal sequence must be high in order to satisfy constraints on the probability of a feasible solution.
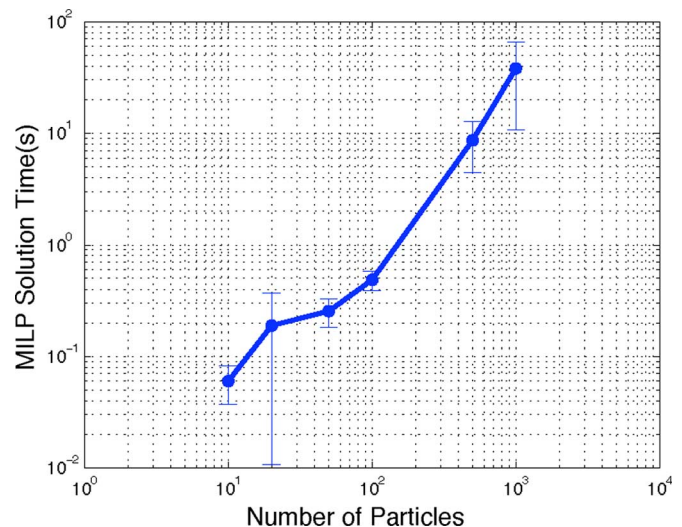

Fig. 15. MILP solution time for ground-vehicle problem. The specified maximum probability of failure was 0.01.

sets, the fair proposal approaches the failure robust one, except that the variance is much greater in the fair-proposal case. This is because on the rare occasion that brake failure transitions are sampled, the solution is very different from the average case. This variance is particularly undesirable for control. The MILP solution times for this example are shown in Fig. 15.

VIII. COMPARISON WITH MONTE–CARLO MARKOV CHAIN METHODS

One alternative approach to chance-constrained control with non-Gaussian uncertainty and continuous decision variables was

previously proposed by Lecchini-Visintini *et al.* [29]. This approach uses an MCMC framework [45] to find an approximately optimal control input through simulation-based optimization. This works by estimating the distribution of a random variable, which is constructed so that the peaks of the distribution coincide with the optimal decision value. The estimation process is then carried out by MCMC [45]. In this section, we first discuss differences between the two approaches and then provide a performance comparison.

MCMC has two main advantages over the new particle control approach. First, it is not restricted to linear system dynamics, and second, the distributions of the uncertain variables can be functions of the state and control inputs. However, there are two key disadvantages. The first is that the convergence of MCMC is sensitive to a number of parameters. These parameters must be tuned carefully by hand to achieve good performance, which is a process that was presented in [29] and takes several iterations. Second, MCMC is more computationally expensive than our MILP approach for the Boeing 747 altitude envelope problem described in Section VII-A, as we show in this section.

The MCMC optimization approach proposed in [29] converts a chance-constrained problem to an unconstrained stochastic optimization problem that penalizes constraint violation. Applying this approach to problem 1, the cost function is

$$\tilde{h}(\mathbf{u}_{0:T-1}, \mathbf{x}_{1:T}) = \begin{cases} e^{-h(\mathbf{u}_{0:T-1}, \mathbf{x}_{1:T})} + \Lambda, & \mathbf{x}_{1:T} \in F \\ 1, & \mathbf{x}_{1:T} \notin F \end{cases} \quad (48)$$

where Λ is a parameter used to reward constraint satisfaction. It is shown in [29] that the maximizer of (48) will satisfy the original chance constraint $p(\mathbf{x}_{1:T} \notin F) \le \delta$ if Λ is chosen, such that

$$\Lambda = \frac{1 - \hat{P}}{\delta - \hat{P}} \quad (49)$$

where $\hat{P}$ is the probability of constraint violation for some known feasible solution. To minimize suboptimality, $\hat{P}$ should be minimized.

The MCMC approach proposed in [29] was applied to the Boeing 747 problem described in Section VII-A with a maximum probability of failure of $\delta = 0.1$. We were able to find a feasible solution with $\hat{P} = 5 \times 10^{-5}$, which gives $\Lambda = 10.0045$. We used a uniform proposal distribution for the altitude setpoints $\mathbf{u}_{0:T-1}$, with minimum and maximum values of 19750 and 21750 ft, respectively. These values were chosen since they are the minimum and maximum values of the flight path envelope. The cost $h(\cdot)$ was as defined in (40), and the constraints were defined by the flight envelope shown in Fig. 4. The MCMC approach generates J independent samples of the random variables per iteration. A larger J is more computationally intensive, but concentrates the resulting distribution around the optimal solution. We varied J from 1 to 50 in our experiments. MCMC uses a "burn-in" period to allow the Markov chain to reach a stationary distribution. Samples obtained during this period are discarded. We used a "burn-in" period of one-tenth of the total number of iterations.

Fig. 16 shows the best solution obtained by using the MCMC approach either after 10 000 iterations or 13.4 h of computation.
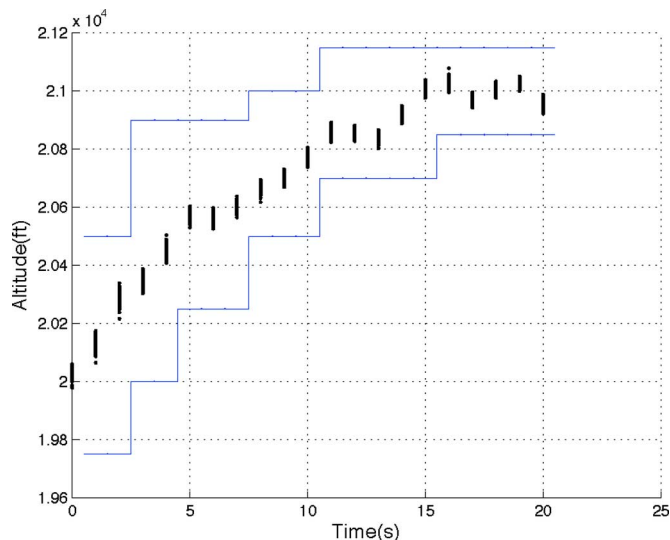

Fig. 16. Best MCMC solution after 10 000 iterations or 13.4 h of computation. The solution is feasible but has a cost of 16.3 compared with particle control's average cost of 1.84.

The best solution was obtained by using $J = 10$. After these iterations, the solution is feasible but has a very high cost of 16.3 compared with the particle control's average cost of 1.83. This high cost is due to the unnecessary altitude changes that can be seen in Fig. 16. The true probability of failure was estimated using 10^6 Monte–Carlo simulations to be approximately zero, which indicates a high degree of conservatism in the plan.

The particle control approach proposed in this paper solves this problem in seconds for reasonable levels of approximation error (see Figs. 5 and 6) and achieves a much lower cost. The particle control approach is therefore more computationally efficient that the MCMC approach of Lecchini-Visintini *et al.* [29]. We acknowledge that it might be possible to improve the performance of MCMC by further manual tuning of the optimization parameters; however, our attempts to do so did not yield any improvement. Furthermore, the necessity for manual tuning makes the MCMC approach less appropriate for autonomous robotic applications. For $J = 1$ and $J = 5$, for example, MCMC did not find a feasible solution after 10 000 iterations.

IX. COMPARISON WITH SCENARIO METHOD

Another approach to solve chance-constrained convex optimization problems was proposed by Calafiore and Campi [14]. This *scenario* approach can be applied to chance-constrained predictive control problems with non-Gaussian noise and linear system dynamics. Unlike the particle control approach presented in this paper, the scenario approach is limited to convex feasible regions. In this section, we compare the two approaches for a problem with a convex feasible region, and discuss key differences between the algorithms. The key idea behind the *scenario* method of [14] is to generate samples of the uncertain variables, which are referred to as scenarios, and to ensure that the constraints are satisfied for all scenarios. Calafiore and Campi [14] provide an elegant bounding result that specifies how many

scenarios are sufficient to ensure that the chance constraints are satisfied with a certain confidence. Letting the state sequence for each scenario be denoted by $\mathbf{x}_{1:T}^{(i)}$, the result then specifies that

$$\mathbf{x}_{1:T}^{(i)} \in F, \quad i = 1, \ldots, N_s \implies p\big(p(\mathbf{x}_{1:T} \notin F) \leq \delta\big) \geq \beta \tag{50}$$

where N_s is the number of scenarios specified by the analytic result of Calafiore and Campi [14], as a function of the confidence parameter β, the chance constraint value δ, and the dimension of the problem.

The scenario method and the particle control method take two different approaches to deal with the intractability of the full chance-constrained problem stated in Section II. Although particle control *approximates* the chance-constrained problem using samples, the scenario method *bounds* the chance-constrained problem using samples. Bounding and approximation are two common solutions to deal with intractable problems. Bounding techniques have the advantage of guaranteeing that constraints are satisfied, while approximation techniques often do not. However, if the bound used is too loose, the solution returned by the bounding approach can be highly conservative, thereby leading to excessive cost and even infeasibility [37]. We now show empirically that this is indeed the case when comparing the scenario approach of Calafiore and Campi [14] and the particle-control approach presented in this paper.

We used the scenario approach of Calafiore and Campi [14] to solve the aircraft altitude control problem described in Section VII-A. The maximum probability of failure was set to $\delta = 0.1$, and the confidence parameter was set to $\beta = 0.999$. For these parameters, the analytic result of Calafiore and Campi [14] specifies that $N_s = 1439$ scenarios are sufficient to ensure that the chance constraints are satisfied with high probability. Since the plan generated depends on the sampled values, we ran the scenario approach 40 times. The true probability of failure of the solution was estimated using 10^6 Monte–Carlo simulations. The mean probability of failure was 5.2×10^{-4}, and the variance was 5.4×10^{-4}. Hence, on average, for this example, the scenario approach gives a probability of failure almost 200 times less than the allowable value, which indicates a high degree of conservatism. As shown in Fig. 5, particle control with 200 particles has an average probability of failure of 0.107 and a variance of 0.019. Although the approximation technique of particle control does not provide guarantees that the chance constraints are satisfied, it avoids the conservatism of the bounds used in the scenario technique. This is further reflected in the cost of the solution, which for the scenario approach has an average of 2.07 and for particle control has an average of 1.84.

X. CONCLUSION

In this paper, we have presented a novel approach to optimal, chance-constrained stochastic control that takes into account probabilistic uncertainty due to disturbances, uncertain state estimation, modeling error, and stochastic mode transitions, so that the probability of failure is less than a defined threshold δ. Although we did not consider uncertainty in the feasible region,

the extension is straightforward; this can be used to model, for example, uncertainty in obstacle location. The new method approximates the original stochastic problem as a deterministic one that uses a number of particles. By controlling the trajectories of these particles in a manner optimal with regard to the approximated problem, the method generates approximately optimal solutions to the original chance-constrained problem. Furthermore, the approximation error tends to zero as the number of particles tends to infinity. By using a particle-based approach, the new particle control method is able to handle arbitrary probability distributions. We demonstrate the method in simulation and show that the true probability of failure tends to the desired probability of failure as the number of particles used increases. Furthermore, the time taken to find the globally optimal solution is significantly less than the existing MCMC approaches to chance-constrained control.

APPENDIX

Here, we describe the approach used to reduce the time required for the UAV obstacle avoidance problem in Section VII-B. This approach reduces the average solution time while still guaranteeing that the globally optimal solution is found, if one exists. The key observation is that, since the solution time for MILP is exponential in the size of the problem, solving many small MILPs is faster than solving one large one. Our approach uses this, i.e., to solve many small MILPs instead of one large one, while still guaranteeing that, eventually, the returned solution is the globally optimal, feasible solution to the full MILP. This is inspired by *iterative-deepening* approaches used in graph search [48]. We use `fullMILP` to denote the MILP encoding the entire chance-constrained particle-control problem. The approach proceeds as follows.

1) Initialize `subMILP` as having one randomly chosen particle from `fullMILP` and no obstacles, i.e., $k \leftarrow 0$.
2) Solve `subMILP` to global optimality to get the solution `Solution`(k).
3) Check whether all constraints in `fullMILP` are satisfied by `Solution`(k).
4) If yes, return `Solution`(k) and stop.
5) If no, check if `subMILP=fullMILP`. If yes, return `infeasible`. Otherwise, add to `subMILP`, the particle and the obstacle in `fullMILP` with the greatest constraint violation.
6) $k \leftarrow k + 1$. Go to Step 2.

Since constraints are only added to `subMILP` and never removed, we know that the cost of each successive solution `Solution`(k) cannot decrease. Therefore, once `Solution`(k) is a feasible solution to `fullMILP`, we know that `Solution`(k) is the globally optimal, feasible solution to `fullMILP`. Constraints and particles are added until a feasible solution to `fullMILP` is found or `subMILP=fullMILP`, in which case, we know no feasible solution to `fullMILP` exists. Hence, the iterative-deepening procedure described earlier guarantees to find the globally optimal solution if one exists. In practice, we found that the approach enables the solution to be found much more quickly than simply solving `fullMILP` directly.

REFERENCES

[1] U.S. military specification MIL-F-8785C, U.S. Dept. Defense, Nov. 1980.
[2] C. Andrieu, A. Doucet, S. S. Singh, and V. B. Tadic, "Particle methods for change detection, system identification, and control," in *Proc. IEEE*, vol. 92, no. 3, pp. 423–438, Mar. 2004.
[3] M. Aoki, *Optimization of Stochastic Systems*. New York: Academic, 1967.
[4] N. M. Barr, D. Gangsaas, and D. R. Schaeffer, "Wind models for flight simulator certification of landing and approach guidance and control systems," Boeing Commercial Airplane Co., Renton, WA, Paper FAA-RD-74-206, 1974.
[5] I. Batina, "Model predictive control for stochastic systems by randomized algorithms," Ph.D. dissertation, Tech. Univ. Eindhoven, Eindhoven, The Netherlands, 2004.
[6] A. Bemporad and M. Morari, *Robust Model Predictive Control: A Survey*. Berlin, Germany: Springer-Verlag, 1999, pp. 207–226.
[7] D. P. Bertsekas, *Dynamic Programming and Stochastic Control*. New York: Academic, 1976.
[8] D. P. Bertsekas and J. N. Tsitsiklis, *Introduction to Probability.*. Belmont, MA: Athena Scientific, 2002.
[9] L. Blackmore, "A probabilistic particle control approach to optimal, robust predictive control," in *Proc. AIAA Guid., Navigat. Control Conf.*, 2006, pp. 1–15.
[10] L. Blackmore, "Robust execution for stochastic hybrid systems," Ph.D. dissertation, Mass. Inst. Technol., Cambridge, MA, 2007.
[11] L. Blackmore, A. Bektassov, M. Ono, and B. C. Williams, "Robust, optimal predictive control of jump Markov linear systems using particles," in *Hybrid Systems: Computation and Control, HSCC* (Lecture Notes in Computer Science, 4416), A. Bemporad, A. Bicchi, and G. Buttazzo, Eds. New York: Springer-Verlag, 2007, pp. 104–117.
[12] L. Blackmore, H. X. Li, and B. C. Williams, "A probabilistic approach to optimal robust path planning with obstacles," in *Proc. Amer. Control Conf.*, Minneapolis, MN, 2006.
[13] L. Blackmore, S. Rajamanoharan, and B. C. Williams, "Active estimation for jump Markov linear systems," *IEEE Trans. Autom. Control*, vol. 53, no. 10, pp. 2223–2236, Nov. 2008.
[14] G. C. Calafiore and M. C. Campi, "The scenario approach to robust control design," *IEEE Trans. Autom. Control*, vol. 51, no. 5, pp. 742–753, May 2006.
[15] G. C. Calafiore and L. El Ghaoui, "Linear programming with probability constraints—Part 1," in *Proc. Amer. Control Conf.*, 2007, pp. 2636–2641.
[16] G. C. Calafiore and L. El Ghaoui, "Linear programming with probability constraints—Part 2," in *Proc. Amer. Control Conf.*, 2007, pp. 2642–2647.
[17] M. C. Campi and G. C. Calafiore, "New results on the scenario design approach," in *Proc. Conf. Decis. Control*, 2007, pp. 6184–6189.
[18] P. C. Chandrasekharan, *Robust Control of Linear Dynamical Systems*. New York: Academic, 1996.
[19] L. Chisci, P. Falugi, and G. Zappa, "Predictive control for constrained systems with polytopic uncertainty," in *Proc. Amer. Control Conf.*, Arlington, VA, 2001.
[20] O. L. V. Costa, M. D. Fragoso, and R. P. Marques, *Discrete-Time Markovian Jump Linear Systems*. New York: Springer-Verlag, 2005.
[21] A. Doucet, N. de Freitas, and N. J. Gordon, *Sequential Monte Carlo Methods in Practice*. New York: Springer-Verlag, 2001.
[22] A. Doucet, N. Gordon, and V. Krishnamurthy, "Particle filters for state estimation of jump Markov linear systems," Dept. Eng., Cambridge Univ., Cambridge, U.K., Tech. Rep. CUED/FINFENG/TR 359, 1999.
[23] N. Freitas, "Rao-Blackwellised particle filtering for fault diagnosis," in *Proc. IEEE Aerosp. Conf.*, 2001, pp. 4-1767–4-1772.
[24] A. Greenfield and A. Brockwell, "Adaptive control of nonlinear stochastic systems by particle filtering," in *Proc. Conf. Control Autom.*, 2003, pp. 887–890.
[25] D. H. Van and Hessem, "Stochastic inequality constrained closed-loop model predictive control," Ph.D. dissertation, Tech. Univ. Delft, Delft, The Netherlands, 2004.
[26] *ILOG CPLEX User's Guide*, IBM ILOG, 1999.
[27] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra, "Planning and acting in partially observable stochastic domains," Brown Univ., Providence, RI, Tech. Rep. CS-96-08, 1996.
[28] P. Kall and S. W. Wallace, *Stochastic Programming*. New York: Wiley, 1994.
[29] A. Lecchini-Visintini, W. Glover, J. Lygeros, and J. Maciejowski, "Monte Carlo optimisation for conflict resolution in air traffic control," *IEEE Trans. Intell. Transp. Syst.*, vol. 7, no. 4, pp. 470–482, Dec. 2006.
[30] A. Lecchini-Visintini, J. Lygeros, and J. M. Maciejowski, "On the approximate domain optimization of deterministic and expected value criteria," in *Proc. Conf. Decis. Control*, 2008, pp. 1–6.
[31] Y. I. Lee and B. Kouvaritakis, "A linear programming approach to constrained robust predictive control," in *Proc. Amer. Control Conf.*, 2000, pp. 1765–1770.
[32] P. Li, M. Wendt, and G. Wozny, "A probabilistically constrained model predictive controller," *Automatica*, vol. 38, pp. 1171–1176, 2002.
[33] D. McRuer, I. Ashkenas, and D. Graham, *Aircraft Dynamics and Automatic Control*. Princeton, NJ: Princeton Univ. Press, 1990.
[34] R. Morales-Menendez, N. de Freitas, and D. Poole, "Real-time monitoring of complex industrial processes with particle filters," in *Proc. Neural Inf. Process. Syst.*, 2002, pp. 1–8.
[35] Y. Nesterov, "Primal-dual subgradient methods for convex problems," Center Operations Res. Econometrics, Catholic Univ. Louvain, Louvain-la-Neuve, Belgium, Tech. Rep., 2005.
[36] A. Y. Ng and M. Jordan, "PEGASUS: A policy search method for large MDPs and POMDPs," in *Proc. 16th Conf. Uncertainty Artif. Intell.*, 2000, pp. 1–10.
[37] M. Ono and B. C. Williams, "An efficient motion planning algorithm for stochastic dynamic systems with constraints on probability of failure," in *Proc. Conf. Decis. Control*, 2008, pp. 1–9.
[38] H. Pham, "On some recent aspects of stochastic control and their applications," *Probability Surv.*, vol. 2, pp. 506–549, 2005.
[39] J. Pineau and G. Gordon, "POMDP planning for robust robot control," in *Proc. 12th Int. Symp. Robot. Res.*, 2005, pp. 69–82.
[40] J. Pineau, G. Gordon, and S. Thrun, "Policy-contingent abstraction for robust robot control," presented at the Conf. Uncertainty Artif. Intell., Acapulco, Mexico, 2003.
[41] A. Prekopa, *Stochastic Programming*. Norwell, MA: Kluwer, 1995.
[42] J. G. Proakis, *Digital Communications*. New York: McGraw-Hill, 1995.
[43] M. L. Puterman, *Markov Decision Processes*. New York: Wiley, 1994.
[44] A. Richards, J. How, T. Schouwenaars, and E. Feron, "Plume avoidance maneuver planning using mixed integer linear programming," in *Proc. AIAA Guid., Navigational Control Conf.*, 2001, pp. 1–10.
[45] C. Robert and G. Casella, *Monte Carlo Statistical Methods.*. New York: Springer-Verlag, 1999.
[46] J. S. Rosenthal, "Minorization conditions and convergence rates for Markov Chain Monte Carlo," *J. Amer. Stat. Assoc.*, vol. 90, no. 430, pp. 558–566, 1995.
[47] 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.
[48] S. J. Russell and P. Norvig, Eds., *Artificial Intelligence: A Modern Approach*. Englewood Cliffs, NJ: Prentice-Hall, 2003.
[49] T. Schouwenaars, B. De Moor, E. Feron, and J. How, "Mixed integer programming for multi-vehicle path planning," in *Proc. Eur. Control Conf.*, 2001, pp. 2603–2608.
[50] A. Schwarm and M. Nikolaou, "Chance-constrained model predictive control," *AIChE J.*, vol. 45, pp. 1743–1752, 1999.
[51] A. Shapiro, "Stochastic programming approach to optimization under uncertainty," *Math. Programming, Ser. B*, vol. 112, no. 1, pp. 183–220, 2008.
[52] S. S. Singh, N. Kantas, B. Vo, A. Doucet, and R. Evans, "Simulation-based optimal sensor scheduling with application to observer trajectory planning," *Automatica*, vol. 43, no. 5, pp. 817–830, 2007.
[53] J.-J. E. Slotine and D. R. Yoerger, "The robust control of robot manipulators," *Int. J. Robot. Res.*, vol. 4, pp. 49–64, 1985.
[54] R. S. Sutton, *On the Significance of Markov Decision Processes*. Berlin, Germany: Springer-Verlag, 2003, pp. 133–154.
[55] S. A. Tarim, S. Manandhar, and T. Walsh, "Stochastic constraint programming: A scenario-based approach," *Constraints*, vol. 11, pp. 53–80, 2006.

[56] J. Tugnait, "Detection and estimation for abruptly changing systems," *Automatica*, vol. 18, pp. 607–615, 1982.

[57] A. N. Vargas, W. Furloni, and J. B. R. do Val, "Constrained model predictive control of jump linear systems with noise and non-observed Markov state," in *Proc. Amer. Control Conf.*, 2006, pp. 929–934.

[58] M. Vidyasagar, "Randomized algorithms for robust controller synthesis using statistical learning theory," *Automatica*, vol. 37, pp. 1515–1528, 2000.

Lars Blackmore received the M.Eng. degree in electrical engineering from the University of Cambridge, Cambridge, U.K., and the Ph.D. degree in control and estimation from Massachusetts Institute of Technology, Cambridge, in 2007.

He is currently a Technologist with the Guidance and Control Analysis Group, Jet Propulsion Laboratory, California Institute of Technology, Pasadena. His research interests include guidance and control for planetary pinpoint and precision landing, guidance and control under stochastic uncertainty, and estimation for formation-flying spacecraft.

Masahiro Ono received the S.B. degree in aeronautics and astronautics from the University of Tokyo, Tokyo, Japan, in 2005 and the S.M. degree in aeronautics and astronautics in 2007 from the Massachusetts Institute of Technology (MIT), Cambridge, where he is currently working toward the Ph.D. degree in aeronautics and astronautics.

He is also an S.M. candidate in the Technology and Policy Program at MIT. His research interests include optimal control and planning under uncertainty and their application to robotics.
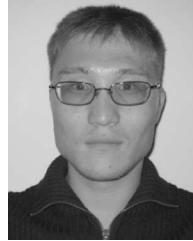
Askar Bektassov received the M.Sc. degree in information engineering from the Università degli Studi Roma Tre, Rome, Italy, and the MBA degree from Scuola Mattei, Rome.

He is currently with the Negotiations and Contracts Team, Eni E&P, Rome, where he deals with decision making in a stochastic environment.

Brian C. Williams received the Ph.D. degree from the Massachusetts Institute of Technology (MIT), Cambridge, in 1989.

He pioneered multiple-fault, model-based diagnosis at the Xerox Palo Alto Research Center. He was with the National Aeronautics and Space Administration (NASA) Ames Research Center, where he was engaged in the autonomous systems area. He coinvented the remote agent model-based autonomous control system, which flew on the NASA Deep Space One probe in 1999. He is currently a Professor with the Department of Aeronautics and Astronautics, MIT. He is also a member of the Advisory Council of the NASA Jet Propulsion Laboratory, California Institute of Technology, Pasadena. His research interests include model-based autonomy—the creation of long-lived autonomous systems that are able to explore, command, diagnose, and repair themselves using fast, common-sense reasoning.