Robust Execution of Qualitative State Plans for Agile Systems using Probabilistic Particles

Lars Blackmore
Massachusetts Institute of Technology
77 Massachusetts Avenue
Cambridge, MA 02139
larsb@mit.edu

Brian Williams
Massachusetts Institute of Technology
77 Massachusetts Avenue
Cambridge, MA 02139
williams@mit.edu

Abstract

Agile autonomous systems such as Autonomous Underwater Vehicles (AUVs) have great potential in fields such as science exploration and surveillance. These continuous dynamic systems must perform complex, time-critical missions, while being robust to uncertainty in their operating environment. Discrete and continuous sources of uncertainty include disturbances, uncertain localization, modeling uncertainty, and component failures.

Previous work used execution of *temporal state plans* to achieve control of dynamic systems. This paper extends this work to robust execution of state plans under stochastic uncertainty. Given a temporal state plan and dynamic system model our new executive plans a near-optimal control sequence subject to the probability of plan failure being less than a specified threshold. This threshold permits the user to trade robustness against performance.

To make the optimal, robust execution problem tractable, we approximate the system state distribution using a finite number of probabilistic particles. We then optimally plan the future distribution of these particles, with the property that as the number of particles tends to infinity, the approximation becomes exact. This gives an any-time approach to robust execution under stochastic uncertainty. We demonstrate the performance of the algorithm in simulation using an AUV scenario.

Introduction

Control of autonomous dynamic systems, such as Unmanned Air Vehicles (UAVs) and Autonomous Underwater Vehicles (AUVs) has received a great deal of attention in recent years. Such systems have enormous potential in fields such as ocean science, space exploration. Control of these dynamic systems is challenging for a number of reasons. First, mission success involves achieving task plans comprised of goals that constrain both discrete and continuous state, and are related by timing constraints. Second, such systems are underactuated, meaning that not all state variables are directly controllable. Third, there is significant uncertainty in the operating environment. Sources of uncertainty include disturbances, uncertain localization, modeling error, and component failures. This uncertainty is stochastic, and can cause the true system state to deviate significantly from the plan.

In this paper we provide a *model-based executive* that addresses these challenges. The executive takes as its input a temporal qualitative state plan, whose activities constrain the allowed continuous states of the system, and a mixed discrete-continuous linear model of system dynamics. Since states are typically not directly controllable, the executive issues optimal low-level commands, while taking into account uncertainty, to ensure *robust* satisfaction of the state plan.

Building upon prior work on plan dispatchability (Morris, Muscettola, & Tsamardinos 1998; Tsamardinos, Pollack, & Ramakrishnan 2003), (Vidal & Ghallab 1996) introduced a framework for temporal plan execution that models uncertainty with *simple temporal networks under uncertainty*, and guarantees success for these modeled disturbances. This research line focuses on discrete, directly controllable systems.

(Williams *et al.* 2003) introduced the Titan model-based executive, which controls discrete-event, underactuated systems according to a *qualitative state plan* comprised of a sequence of constraints on goal states. Combining model-based and temporal plan execution, (Leaute & Williams 2005) introduced the Sulu executive, which controls continuous, underactuated systems by executing a state plan that involves continuous goal states and temporal constraints.

In a similar spirit to prior work in on-board planning and execution(Ambros-Ingerson & Steel 1988; Wilkins & Myers 1995; Chien *et al.* 2000), the *Sulu* executive uses a Model Predictive Control (MPC) approach(Richalet *et al.* 1976) to plan optimal control sequences over a long horizon, while executing a shorter sequence of control actions. Through continual replanning, the executive is able to adapt to disturbances. However, Sulu does not model or plan for uncertainty. In many cases, optimal plans that do not take into account uncertainty are brittle; a fuel-optimal path through an obstacle field will only just miss obstacles, and wind disturbances are likely to cause failure. Robust execution of temporal state plans, therefore, requires explicit modeling and planning for uncertainty.

This paper extends the work of (Leaute & Williams 2005), producing an executive that is robust to stochastic uncertainty. Previous work(Li, Wendt, & Wozny 2000; Hessem 2004) used a *chance-constrained* formulation for robustness under stochastic uncertainty. Chance constraints ensure that the probability of failure is below a user-specified threshold, denoted δ. Specifying this threshold enables the user to trade robustness against performance; a plan with a low δ typically requires more fuel, or more time, to complete. Execution under stochastic uncertainty is well known to be challenging because it involves online planning in the much larger space of future state *distributions* rather than future states. Prior work approximates the optimal, chance-constrained planning problem using a finite set of samples, or *particles*, to approximate the state distribution(Blackmore 2006; Blackmore *et al.* 2007). The key idea is to plan the evolution of the particle distribution optimally, while satisfying chance constraints, with the property that the approximation becomes exact as the number of particles tends to infinity. This gives an any-time approach to planning under stochastic uncertainty.

We extend this 'particle control' approach to robust execution of qualitative state plans with temporal constraints. The resulting executive enables the user to impose chance constraints on groups of state activities within the plan; for example, requiring that the goal state is reached in time with a certain probability, while ensuring that collision with obstacles occurs with a different probability. We show that for a class of hybrid discrete-continuous dynamic systems known as Jump Markov Linear Systems (JMLS), the resulting optimization problem can be solved to global optimality using Mixed Integer Linear Programming (MILP). JMLS are powerful modeling tools for many dynamic autonomous systems; uncertainty in the continuous dynamics is used to model disturbances and localization uncertainty, for example, while stochastic transitions in the discrete dynamics are used to model component failures(Costa, Fragoso, & Marques 2005). Our executive uses a receding-horizon approach to plan near-optimal controls while being robust to all of these forms of uncertainty. Furthermore, while related to prior work on conformant and probabilistic planning(Majercik & Littman 1998; Smith & Weld 1998; Domshlak & Hoffmann 2006), our approach is different in being able to plan in *continuous* decision spaces, as well planning with temporally flexible goals.

Problem Statement

Given a dynamic system (or *plant*) described by a *Jump Markov Linear System*, and a *Chance-Constrained Qualitative State Plan*, specifying the desired evolution of the plant state along with reliability constraints on the probability of failure, the *Continuous Robust Execution Problem* consists of designing a control sequence that is consistent with the state plan. In this section we present a formal definition of this problem.

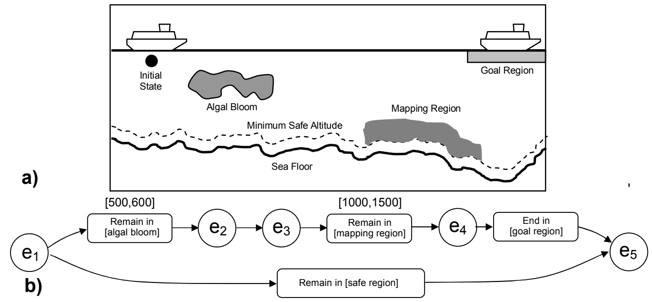

Figure 1. a) Autonomous Underwater Vehicle science mission and b) Qualitative State Plan for the AUV mission

AUV Planning Example We use an AUV depth planning example to demonstrate the approach. In this example, the plant is an AUV performing an ocean science mission that involves collecting data from within an algal bloom and performing mapping of the sea floor. The mission is depicted in Fig. 1. An informal description of the mission's state plan is:

> The AUV must reach the algal bloom and remain there for between $500s$ and $600s$. Afterwards, the AUV must go to the mapping depth and remain there for between $1000s$ and $1500s$. At the end of the mission the AUV must arrive at the goal region to rendezvous with the surface vessel. At all times the AUV must remain at a safe altitude above the sea floor, and must stay within its operating limits.

Definition of a JMLS A Jump Markov Linear System (JMLS) $\mathcal{M} = \langle \mathbf{x}_c, \mathcal{X}_c, \mathbf{x}_d, \mathcal{X}_d, \mathbf{u}, \Omega, \mathcal{D}, V \rangle$ consists of a vector $\mathbf{x}_c$ of *continuous state variables*, taking on values from the *continuous state space* $\mathcal{X}_c \subset \Re^n$, the *discrete mode* $\mathbf{x}_d$, taking on values from the *discrete state space* $\mathcal{X}_d$, a vector $\mathbf{u}$ of *input variables*, taking on values from the *input space* $\Omega \subset \Re^m$, a set $\mathcal{D}$ of discrete-time *state equations*, and a *transition distribution* V. The state equations $\mathcal{D}$ define the evolution of the continuous state $\mathbf{x}_c$ and take the form:

$$\mathbf{x}_{c,\tau+1} = A(\mathbf{x}_{d,\tau})\mathbf{x}_{c,\tau} + B(\mathbf{x}_{d,\tau})\mathbf{u}_\tau + \omega_\tau. \quad (1)$$

The evolution of the discrete state $\mathbf{x}_d$ is defined by:

$$p\big(\mathbf{x}_{d,\tau+1} = j | \mathbf{x}_{d,\tau} = i\big) = V(i,j). \quad (2)$$

The hybrid discrete-continuous state $\langle \mathbf{x}_c, \mathbf{x}_d \rangle$ is denoted $\mathbf{x}$. We use the subscript form $\mathbf{x}_{c,\tau}$ to denote the continuous state $\mathbf{x}_c$ at time t_τ. The sequence $\langle \mathbf{x}_{c,0}, \dots, \mathbf{x}_{c,N} \rangle$ is denoted $\mathbf{x}_{c,0:N}$. The sampling interval in (1) is denoted Δt. We use $\mathbf{x}_c$ to denote the random variable and use X_c to denote its realization.

In our AUV planning example, we consider the motion of the AUV in the vertical plane. We use the linearized, discrete-time closed-loop longitudinal dynamics derived by (McEwen & Streitlien 2001) to model the operation of the AUV in mode $\mathbf{x}_d = 1$ ('nominal'); the mode $\mathbf{x}_d = 2$ ('faulty') dynamics are identical except that the elevator actuator has no effect. The control inputs $\mathbf{u}$ are depth waypoints given to the AUV.

Definition of a Chance Constrained Qualitative State Plan A Chance Constrained Qualitative State Plan $P = \langle \mathcal{E}, \mathcal{C}, \mathcal{A}, \mathcal{G}_c, \mathcal{G}_m, F \rangle$ specifies a desired evolution of the plant state over time, and is defined by a set $\mathcal{E}$ of discrete events, a set $\mathcal{A}$ of *activities*, imposing constraints on the plant state evolution, a set $\mathcal{C}$ of *temporal constraints* between events, a set $\mathcal{G}_c$ of *chance constraints* that specify reliability constraints on the success of activities in the plan, a set $\mathcal{G}_m$ of *expected state constraints* that guarantee the success of activities for the most likely system state, and an *objective function* F, which must be minimized.

An *activity* $a = \langle e_S, e_E, c_S \rangle$ has an associated real-valued *start event* e_S and an *end event* e_E. c_S is called a *state constraint* on the variable $\mathbf{x}$ and can take one of the following forms, where R_S, R_E, $R_\forall$ and $R_\exists$ are regions of the state space S, and T is a schedule for P:

1. *Start in state region* R_S: $\mathbf{x}_\tau \in R_S \quad t_\tau = T(e_S)$;
2. *End in state region* R_E: $\mathbf{x}_\tau \in R_E \quad t_\tau = T(e_E)$;
3. *Remain in state region* $R_\forall$: $\mathbf{x}_\tau \in R_\forall \quad \forall t_\tau \in [T(e_S), T(e_E)]$;
4. *Go through state region* $R_\exists$: $\mathbf{x}_\tau \in R_\exists \quad \exists t_\tau \in [T(e_S), T(e_E)]$.

Since *start in* and *go through* activities are easily derivable from the primitives *remain in* and *end in*, we consider only *remain in* and *end in* activities from here.

A *temporal constraint* $\langle e_S, e_E, \Delta T^{min}_{e_S \to e_E}, \Delta T^{max}_{e_S \to e_E} \rangle$ is a constraint, specifying that the duration from a start event e_S to an end event e_E be in the real-valued interval $[\Delta T^{min}_{e_S \to e_E}, \Delta T^{max}_{e_S \to e_E}] \subseteq [0, +\infty]$.

For a stochastic system, the predicted state is a random variable. The success of a set of activities $\mathcal{A}$ is hence a random event. A *chance constraint* denoted c_c specifies that the set of activities $\mathcal{A}(c_c)$ must fail with probability at most $\delta(c_c)$. An *expected state constraint*, denoted c_m, requires the set of activities $\mathcal{A}(c_m)$ to succeed for the expected system state $E[\mathbf{x}]$ and schedule T. We define the indicator function $s(\cdot)$ as follows:

$$s(\mathcal{A}(c_c), X_{1:N}, T) = \begin{cases} 1 & \text{any activity in } \mathcal{A}(c_c) \text{ fails} \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

We illustrate a Qualitative State Plan diagrammatically by an acyclic directed graph in which events are represented by nodes, temporal constraints by arcs, labeled by their corresponding time bounds, and activities by arcs labeled with associated state constraints. The Qualitative State Plan for the AUV mission is shown in Fig. 1b). This plan has two chance constraints:

1. The probability that either `Remain in [safe region]` or `End in [goal region]` (the safety activities) fails is at most 10^{-6}.
2. The probability that any of the other activities (the science activities) fails is at most 0.02.

Note that the plan requires a lower probability of failure for activities that are essential for safety of the AUV than for activities upon which the science return of the mission depends.

Definition of Robust Execution Problem A Chance Constrained Qualitative State Plan $P = \langle \mathcal{E}, \mathcal{C}, \mathcal{A}, \mathcal{G}_c, \mathcal{G}_m, F \rangle$ is *satisfied* by a random state sequence $\mathbf{x}_{0:f}$, an input sequence $\mathbf{u}_{0:f}$ and a schedule T if and only if the schedule T is temporally consistent, that is T satisfies all of the constraints in $\mathcal{C}$, all of the chance constraints in $\mathcal{G}_c$ are satisfied, and all of the expected state constraints in $\mathcal{G}_m$ are satisfied. $\mathbf{u}_{0:f}$ is *optimal* if it satisfies P while minimizing the *objective function* $F(\mathbf{u}_{0:f}, \mathbf{x}_{0:f}, T)$. A common objective is to minimize the scheduled time $T(e_E)$ for the end event e_E of P.

Given an initial state distribution $p(\mathbf{x}_0)$, a plant model $\mathcal{M}$ and a state plan P, the *robust execution problem* consists of incrementally generating, for every time step t_τ, a control action $\mathbf{u}_\tau$ given a sequence of observations $\mathbf{y}_{0:\tau}$. The final resulting control sequence $\mathbf{u}_{0:f}$, the resulting state trajectory $\mathbf{x}_{c,0:f}$ and schedule T must satisfy the JMLS $\mathcal{M}$ and the Qualitative State Plan P.

Summary of Approach

The robust execution problem is intractable for two key reasons. First, in the case of long-duration missions, a full plan is too long to be generated in a single step. Second, finding an optimal control sequence that satisfies the Qualitative State plan exactly is intractable; constraints on the probability of activity failure are particularly problematic, since this value cannot be evaluated in closed form in the general case. In order to make the robust execution problem tractable we introduce two approximations. First, we solve a *receding horizon* approximation to the problem. Second, we solve a *particle control* approximation to the robust execution problem; this uses a finite set of particles to approximate the planned state distribution. We show that this approximation enables the planning problem to be solved to global optimality using MILP.

Receding Horizon Execution

In a similar manner to (Leaute & Williams 2005) we use a *receding horizon*, or MPC approach to make the robust execution problem tractable. We solve the robust execution up to a limited *planning horizon*, and re-solves it when it reaches a shorter *execution horizon*. This approaches generates control sequences that are optimal over the planning horizon, and are globally near-optimal.

We define the Receding-Horizon Robust Execution Problem as follows: Given an initial state distribution $\mathbf{x}_0$, a plant model $\mathcal{M}$ and a state plan P, the *single-stage, limited horizon* robust execution problem consists of generating an optimal control sequence $\mathbf{u}_{0:N_P}$ that satisfies P, where N_P is the length of the planning horizon. The *receding horizon robust execution problem* consists of iteratively solving single-stage limited horizon robust execution problems for successive initial states $\mathbf{x}_{i\cdot N_E}$ with $i = 0, 1, \ldots$ where $N_E \leq N_P$ is the execution horizon.

Particle Control Approximation

The key technical challenge is to make the single-stage limited horizon robust execution problem tractable for online solution. We accomplish this by approximating the problem using a finite set of samples or particles. Prior work(Blackmore 2006; Blackmore *et al.* 2007) used this approach to perform control of dynamic systems within temporally-static feasible regions. We now extend this to execution of temporally-flexible plans.

The key observation behind the new method is that, by approximating all probabilistic distributions using particles, an intractable stochastic optimization problem can be approximated as a tractable deterministic optimization problem. By solving this deterministic problem we obtain an approximate solution to the original stochastic problem, with the additional property that as the number of particles used tends to infinity, the approximation becomes exact.

In outlining the method, note that for the JMLS the future continuous states $\mathbf{x}_{c,\tau}$ for $\tau = 1, \ldots, N_p$ are functions of the control inputs $\mathbf{u}_{0:N_p-1}$, the initial state $\mathbf{x}_0$, disturbances $\nu_{0:N_p-1}$, and the mode sequence $\mathbf{x}_{d,0:N_p}$:

$$\mathbf{x}_{c,\tau} = f_\tau(\mathbf{x}_{c,0}, \mathbf{x}_{d,0:\tau-1}, \mathbf{u}_{0:\tau-1}, \nu_{0:\tau-1}). \qquad (4)$$

The initial state, disturbances and mode sequence are uncertain, but are random variables with known distributions. Hence the future continuous states are also random variables, whose distributions depend on the control inputs.[1] The chance constrained particle control method for Qualitative State Plans is given in Table 1. The method is illustrated in Fig. 2. This approach solves an approximation of the single-stage, limited horizon robust execution problem. Due to the strong law of large numbers, as the number of particles converges to infinity, the approximated probability of activity failure (7) converges to the true probability of failure. Furthermore the approximated cost function and expected state converge to the true values, again due to the strong law of large numbers. We therefore have convergence of the approximated optimization problem to the exact single-stage robust execution problem as the number of particles tends to infinity.

The general formulation presented in this section encompasses a very general set of dynamic systems, including ones with nonlinear dynamics and hybrid state; the key restriction being that the distributions of the

[1]Modeling errors can be modeled as an additional stochastic disturbance process(Ljung 1999). For notational simplicity we assume for the rest of the development a single disturbance process; however the method applies equally to multiple disturbance processes, and hence to modeling errors as well as external disturbances.

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

$$\mathbf{x}_{1:N_p}{}^{(i)} := \langle \mathbf{x}_1{}^{(i)} \mathbf{x}_2{}^{(i)} \ldots \mathbf{x}_{N_p}{}^{(i)} \rangle$$
$$\mathbf{x}_\tau{}^{(i)} = f_\tau(\mathbf{x}_0{}^{(i)}, \mathbf{u}_{0:\tau-1}, \nu^{(i)}_{0:\tau-1}), \qquad (5)$$

where $\mathbf{x}_0{}^{(i)}$, $\nu^{(i)}_{0:\tau-1}$ and $\mathbf{x}_{d,0:N_p-1}{}^{(i)}$ are *known values* sampled from random variables, whereas $\mathbf{u}_{0:\tau-1}$ are decision variables over which to optimize.
3) *Approximate the chance constraints in terms of the generated particles.* The probability of any activity in the set $\mathcal{A}(c_c)$ failing is given by:

$$p\Big(\mathcal{A}(c_c) \text{ fails}\Big) = \int_{\mathbf{x}_{1:N_p}} s\Big(\mathcal{A}(c_c), \mathbf{x}_{1:N_p}, T\Big) d\mathbf{x}_{1:N_p}, \quad (6)$$

where $s(\cdot)$ is the indicator function defined in (3). Given the generated particles, this probability can be approximated as:

$$p\Big(\mathcal{A}(c_c) \text{ fails}\Big) \approx \frac{1}{N} \sum_{i=1}^{N} s\Big(\mathcal{A}(c_c), \mathbf{x}_{1:N_p}{}^{(i)}, T\Big). \qquad (7)$$

The chance constraint c_c is then approximated as follows:

$$\frac{1}{N} \sum_{i=1}^{N} s\Big(\mathcal{A}(c_c), \mathbf{x}_{1:N_p}{}^{(i)}, T\Big) \leq \delta.$$

In other words, for no more than δ of the particles can any activity in $\mathcal{A}(c_c)$ fail. Note that a particle represents a state *trajectory* over the entire planning horizon.
4) *Approximate the constraints on the expected state.* A constraint on the expected state can be written:

$$s\Big(\mathcal{A}(c_m), E[\mathbf{x}_{1:N_p}], T\Big) = 1. \qquad (8)$$

Using the sample mean approximation to the full expectation we have the approximated constraint:

$$s\Big(\mathcal{A}(c_m), \frac{1}{N} \sum_{i=1}^{N} \mathbf{x}_{1:N_p}{}^{(i)}, T\Big) = 1. \qquad (9)$$

5) *Approximate the cost function in terms of particles*

$$\hat{F}(\mathbf{u}_{0:N_p-1}, \mathbf{x}_{1:N_p}{}^{(1)}, \cdots, \mathbf{x}_{1:N_p}{}^{(N)}, T)$$
$$\approx F(\mathbf{u}_{0:N_p-1}, \mathbf{x}_{1:N_p}, T) \qquad (10)$$

6) *Solve deterministic constrained optimization problem*: Minimize $\hat{F}(\mathbf{u}_{0:N_p-1}, \mathbf{x}_{1:N_p}{}^{(1)}, \cdots, \mathbf{x}_{1:N_p}{}^{(N)}, T)$ over $\mathbf{u}_{0:N_p-1}$ and T, subject to:

$$\frac{1}{N} \sum_{i=1}^{N} s\Big(\mathcal{A}(c_c), \mathbf{x}_{1:N_p}{}^{(i)}, T\Big) \leq \delta(c_c), \qquad (11)$$

for all chance constraints c_c, and

$$s\Big(\mathcal{A}(c_m), \frac{1}{N} \sum_{i=1}^{N} \mathbf{x}_{1:N_p}{}^{(i)}, T\Big) = 1, \qquad (12)$$

Table 1. Chance Constrained Particle Control Algorithm

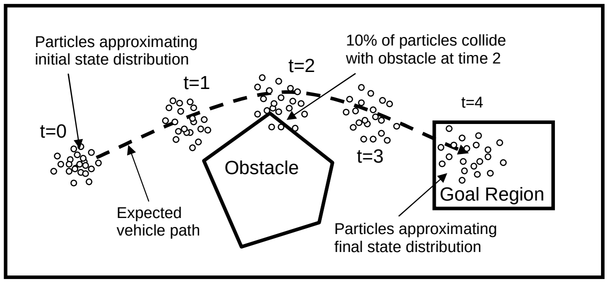

Figure 2. Illustration of new particle control method for robust limited-horizon execution. The Qualitative State Plan P has two activities. The first requires the system state to end in the goal region, and the second requires the state to remain outside of the obstacles at all time steps. A single chance constraint requires failure of either activity to occur with probability at most 10%. The particle control method approximates this so that at most 10% of the particles fail.

uncertain variables must be independent of the control inputs and system state. It is not necessarily true, however, that the optimization problem that results from this formulation is tractable. We now show that for Jump Markov Linear Systems, a piecewise linear cost function and a polygonal feasible region, the optimization can be solved using efficient MILP methods.

Mixed Integer Linear Programming Encoding of Particle Control Problem

We now describe the encoding of the particle control problem for Qualitative State Plans as a MILP.

Approximate Chance Constraints

The particle control algorithm in Table 1 requires that we constrain the number of particles for which an activity fails (Equation 11). We encode this constraint by introducing binary variables $z_i(c_c)$ for each chance constraint c_c that indicate whether a set of activities fails or succeeds for a given particle, such that:

$$z_i(c_c) = 0 \implies s\big(\mathcal{A}(c_c), \mathbf{x}_{1:N_p}{}^{(i)}, T\big) = 1. \tag{13}$$

Constraining the sum of these binary variables then encodes the constraint (11):

$$\frac{1}{N}\sum_{i=1}^{N} z_i(c_c) \le \delta(c_c). \tag{14}$$

The challenge is now to encode the implication (13). For all remain-in activities in $\mathcal{A}(c_c)$ we must encode the implication:

$$z_i(c_c) = 0 \implies \mathbf{x}_\tau{}^{(i)} \in R_\forall(a) \quad \forall \tau \in [T(e_S), T(e_E)], \tag{15}$$

while for end-in activities we must encode the following implication:

$$\begin{aligned} & z_i(c_c) = 0 \implies \left(T(e_E) \le 0 - \frac{\Delta t}{2}\right) \\ & \vee \left(T(e_E) \ge t_{N_p} + \frac{\Delta t}{2}\right) \\ & \vee \left(\exists \tau \quad \mathbf{x}_\tau \in R_E \quad t_\tau \in \left[T(e_E) - \frac{\Delta t}{2}, T(e_E) + \frac{\Delta t}{2}\right]\right), \end{aligned} \tag{16}$$

where we have performed the same time-discretization as (Leaute & Williams 2005); either $T(e_E)$ is scheduled within the planning horizon, in which case $\mathbf{x}_\tau \in R_E$ at the time step closest to $T(e_E)$, or $T(e_E)$ is scheduled outside the planning horizon.

We encode the implications (15) and (16) by introducing additional binary variables, as we now describe. For each time step τ, for each remain-in activity a we define binary variables $b_{after}(\tau, e_S)$, $b_{before}(\tau, e_E)$ and $b_{during}(\tau, e_S, e_E)$, for all $\tau \in \{0, \ldots, N_p\}$. Here, e_S is the start event of a and e_E is the end event of a. Using the following encoding for all $\tau \in \{0, \ldots, N_p\}$, for all events e_S and e_E:

$$\begin{aligned} t_\tau - T(e_S) &< M \cdot b_{after}(\tau, e_S) \\ T(e_E) - t_\tau &\le M \cdot b_{before}(\tau, e_E) \\ b_{during}(\tau, e_S, e_E) &\le M(2 - b_{after} - b_{before}), \end{aligned} \tag{17}$$

we ensure that:

$$t_\tau \in [T(e_S), T(e_E)] \implies b_{during}(\tau, e_S, e_E) = 0. \tag{18}$$

We define binary variables $b_{at}(\tau, e_E)$ for each time step $\tau \in \{0, \ldots, N_p\}$ and each event e_E. We also define binary variables $b_{before}(e_E)$ and $b_{after}(e_E)$ for each event e_E. We impose the following constraints for all $\tau \in \{0, \ldots, N_p\}$, for all events e_S and e_E:

$$\begin{aligned} t_\tau - \frac{\Delta t}{2} - T(e_E) &\le M \cdot b_{at}(\tau, e_E) \\ T(e_E) - t_\tau - \frac{\Delta t}{2} &\le M \cdot b_{at}(t, e_E) \\ T(e_E) &\le M \cdot b_{before}(e_E) \\ \frac{\Delta t}{2} + t_{N_p} - T(e_E) &\le M \cdot b_{after}(e_E) \\ b_{before}(e_E) + b_{after}(e_E) + \sum_{\tau=0}^{N_p} b_{at}(\tau, e_E) &= N_p + 2. \end{aligned} \tag{19}$$

These constraints ensure that either some time step τ occurs within $\frac{\Delta t}{2}$ of $T(e_E)$, and that $b_{at}(\tau, e_E) = 0$ for that τ, or $T(e_E)$ is scheduled outside of the planning horizon. We now use the binaries $b_{at}, b_{before}, b_{after}$ to encode (15) and (16) first for convex, then for nonconvex feasible regions.

Convex Feasible Regions We define a convex polygonal feasible region $R(a)$ associated with activity a as a conjunction of linear constraints $\mathbf{a}_l^T(a)\mathbf{x} \leq b_l(a)$ for $l = 1, \dots, N_a$, where $\mathbf{a}_l(a)$ is defined as pointing outwards from the polygonal region. Then $\mathbf{x}$ lies within $R(a)$ if and only if all of the constraints are satisfied:

$$\mathbf{x} \in R(a) \iff \bigwedge_{l=1,\cdots,N_a} \mathbf{a}_l^T(a)\mathbf{x} \leq b_l(a). \tag{20}$$

For all remain-in activities $a \in \mathcal{A}_\forall(c_c)$ we now impose the following constraint for all $\tau \in \{0, \dots, N_p\}$, for all l and for all $a \in \mathcal{A}_\forall(c_c)$:

$$\mathbf{a}_l^T(a)\mathbf{x}_\tau{}^{(i)} - b_l(a) \leq M\Big(z_i(c_c) + b_{during}(\tau, e_S, e_E)\Big), \tag{21}$$

where M is a large positive constant, and e_S and e_E are the start and end events for activity a. If $z_i(c_c) = 0$ and $b_{during}(\tau, e_S, e_E) = 0$ then every constraint is satisfied for particle i, otherwise (for large enough M), particle i is unconstrained. For convex regions $R(a)$, we therefore have, for all $t_\tau \in [T(e_S), T(e_E)]$, for all $a \in \mathcal{A}_\forall(c_c)$:

$$z_i(c_c) = 0 \implies \mathbf{x}_\tau{}^{(i)} \in R(a), \tag{22}$$

as required. For all end-in activities $a \in \mathcal{A}_E$ we impose the following constraint for all $\tau \in \{0, \dots, N_p\}$, for all l, for all $a \in \mathcal{A}_E(c_c)$:

$$\mathbf{a}_l^T(a)\mathbf{x}_\tau{}^{(i)} - b_l(a) \leq M\Big(z_i(c_c) + b_{at}(\tau, e_E)\Big), \tag{23}$$

where M is a large positive constant and e_E is the end event of activity a. If $z_i(c_c) = 0$ and $b_{at}(\tau, e_E) = 0$ then every constraint is satisfied for particle i, otherwise particle i is unconstrained. For convex regions $R_E(a)$ we have therefore encoded implication (16) for all $a \in \mathcal{A}_E(c_c)$, as required.

Non-convex Feasible Regions A polytopic non-convex feasible region can be described as the complement of a number of polytopic infeasible regions, or obstacles. In other words, at time t_τ the state $\mathbf{x}_\tau$ is in the region if and only if all obstacles are avoided for all time steps. For each activity a with non-convex feasible region $R(a)$, we define a set $O(a) = \Big\langle O(a)_1, \dots, O(a)_{M_a} \Big\rangle$ of obstacles that collectively form the complement of $R(a)$. As noted by (Leaute & Williams 2005) avoidance of a polygonal obstacle can be expressed in terms of a disjunction of linear constraints. That is, the system state at time t_τ, $\mathbf{x}_\tau$, avoids the obstacle $O_j(a)$ if and only if:

$$\bigvee_{l=1,\dots,N_{aj}} \mathbf{a}_{jl}^T(a)\mathbf{x}_\tau \geq b_{jl}(a). \tag{24}$$

We now introduce binary variables $d_{ij\tau l}(a) \in \{0, 1\}$ that indicate whether a given constraint l for a given obstacle $O_j(a)$ is satisfied by a given particle i at a given time step τ. We impose the following for all i, j, l, for all end-in activities $a \in \mathcal{A}_E(c_c)$ and for all $\tau \in \{0, \dots, N_p\}$:

$$\mathbf{a}_{jl}^T(a)\mathbf{x}_\tau{}^{(i)} - \mathbf{b}_{jl}(a) + Md_{i\tau jl}(a) \geq 0 \tag{25}$$

$$\mathbf{a}_{jl}^T(a)\mathbf{x}_\tau{}^{(i)} - \mathbf{b}_{jl}(a) + Md_{i\tau jl}(a) \geq 0, \tag{26}$$

and:

$$\sum_{l=1}^{N_{aj}} d_{i\tau jl}(a) - (N_{aj} - 1) \leq Me_{i\tau j}(a) \tag{27}$$

$$\sum_{l=1}^{N_{aj}} d_{i\tau jl}(a) - (N_{aj} - 1) \leq Me_{i\tau j}(a). \tag{28}$$

These ensure that $e_{i\tau j}(a) = 0$ implies that at least one constraint in obstacle $O_j(a)$ is satisfied by particle i at time step τ. This in turn implies that obstacle $O_j(a)$ is avoided by particle i at time step τ. We now impose, for all remain-in activities, for all $\tau \in \{0, \dots, N_p\}$, for all $a \in \mathcal{A}_\forall(c_c)$ and for all i:

$$\sum_{j=1}^{M_a} e_{i\tau j}(a) \leq M \cdot \Big(z_i(c_c) + b_{during}(\tau, e_S, e_E)\Big), \tag{29}$$

which ensures that, for non-convex feasible regions $R_\forall(a)$, for all $a \in \mathcal{A}_\forall(c_c)$ and for all i:

$$z_i(c_c) = 0 \implies \mathbf{x}_\tau{}^{(i)} \in R_\forall(a) \quad \forall\tau \in [T(e_S), T(e_E)], \tag{30}$$

as required. For all end-in activities $a \in \mathcal{A}_E(c_c)$, we impose the following constraint, for all $\tau \in \{0, \dots, N_p\}$, and for all i:

$$\sum_{j=1}^{M_a} e_{i\tau j}(a) \leq M \cdot \Big(z_i(c_c) + b_{at}(\tau, e_S, e_E)\Big), \tag{31}$$

which ensures that the implication (16) holds for non-convex regions $R_E(a)$, for all $a \in \mathcal{A}_E(c_c)$ as required.

In this section we encoded the implication (13) using linear constraints on continuous and binary decision variables. This means that using the linear constraint (14), we have expressed the approximated chance constraint (11) as a constraint suitable for MILP optimization. Furthermore, as the number of particles converges to infinity, we have convergence of the approximated constraint to the exact chance constraint.

Approximated Expected State Constraints

In accordance with (12), for each expected state constraint $c_m \in \mathcal{G}_m$ we must ensure that for the expected state, approximated using the sample mean of the particle set, all activities succeed in the set $\mathcal{A}(c_m)$. We use $\mathbf{x}_t{}^{(m)}$ to denote the sample mean:

$$\mathbf{x}_\tau{}^{(m)} \triangleq \sum_{i=1}^{N} \mathbf{x}_\tau{}^{(i)} = \sum_{i=1}^{N} \left(\sum_{j=0}^{\tau-1} A^{\tau-j-1}B(\mathbf{u}_j + \nu_j^{(i)}) + A^\tau \mathbf{x}_0{}^{(i)} \right). \tag{32}$$

Expected state constraints can then be handled in a similar manner to chance constraints, except that expected state constraints *require* the corresponding activities to succeed. As the number of particles converges to infinity, the approximated constraint (32) converges to the exact expected state constraint.

Temporal Constraints

Temporal constraints have the form $c = \langle e_S, e_E, \Delta T^{min}_{e_S \to e_E}, \Delta T^{max}_{e_S \to e_E} \rangle$. These are encoded, for each temporal constraint c, as follows:

$$\begin{aligned} T(e_E) - T(e_S) &\geq \Delta T^{min}_{e_S \to e_E} \\ T(e_E) - T(e_S) &\leq \Delta T^{max}_{e_S \to e_E}. \end{aligned} \tag{33}$$

To ensure determinism, we fix the the time of events that have occurred in the past.

Cost Function

The cost function F can be a function of the control inputs $\mathbf{u}_{0:N_p-1}$, the schedule T, and the system state trajectory $\mathbf{x}_{1:N_p}$. Since the system state is uncertain, however, this cost function will typically be an expectation over the system state. In this case we approximate the expectation using the sample mean of the cost function. This is evaluated using the particle population as follows. The true expectation is given by:

$$E[F] = \int F(\mathbf{u}_{0:N_p-1}, \mathbf{x}_{1:N_p}, T) p(\mathbf{x}_{1:N_p}) d\mathbf{x}_{1:N_p} \tag{34}$$

Since $p(\mathbf{x}_{1:N_p})$ can be an arbitrary distribution, this integral is intractable in most cases. The approximated expectation is given by:

$$\hat{F} = \frac{1}{N} \sum_{i=1}^{N} F(\mathbf{u}_{0:N_p-1}, \mathbf{x}_{1:N_p}{}^{(i)}, T), \tag{35}$$

and this can be evaluated without integration. As the number of particles tends to infinity, we have $\hat{F} \longrightarrow E[F]$. Furthermore, since we assume that F is a piecewise linear function of the state and control inputs, the expression for $\hat{F}$ in (35) is also piecewise linear.

Guidance Heuristic

As noted by (Leaute & Williams 2005), end-in activities scheduled outside of the planning horizon are challenging for receding horizon approaches to hybrid execution. Consider the case where an end-in activity with region R_E that is scheduled to start during the planning horizon, i.e. $T(e_S) \in [t_0, t_{N_p}]$, but is scheduled to end after the end of the planning horizon, i.e. $T(e_E) > t_{N_p}$. Intuitively, the system state should be making progress towards R_E during the time period $[T(e_S), t_{N_p}]$. We therefore use a *guidance heuristic* to steer the system state towards R_E when $T(e_S)$ is outside of the planning horizon. We use an identical approach to (Leaute & Williams 2005) except that we minimize the distance between the *expected* system state and the end-in region R_E. The objective in the constrained optimization is:

$$\text{Minimize } \hat{F} + H(\mathbf{x}_T{}^{(m)}), \tag{36}$$

where $H(\cdot)$ is the cost-to-go estimate. (Leaute & Williams 2005) discuss in detail the forms of $H(\cdot)$ that are suitable for MILP optimization.

Analytic Particles

From the definition of JMLS in (1) we use (5) to obtain the following expression for each particle:

$$\begin{aligned} \mathbf{x}_{c,\tau}{}^{(i)} = &\sum_{j=0}^{\tau-1} \Big(\prod_{l=1}^{\tau-j-1} A(\mathbf{x}_{d,l}{}^{(i)}) \Big) \Big(B(\mathbf{x}_{d,j}{}^{(i)}) \mathbf{u}_j + \nu_j^{(i)} \Big) \\ &+ \Big(\prod_{l=1}^{\tau} A(\mathbf{x}_{d,l}{}^{(i)}) \Big) \mathbf{x}_{c,0}{}^{(i)}. \end{aligned} \tag{37}$$

Note that this is a linear function of the control inputs $\mathbf{u}_{0:\tau-1}$, and that $\mathbf{x}_{c,0}{}^{(i)}$, $\nu_j^{(i)}$ and $\mathbf{x}_{d,l}{}^{(i)}$ are all known values. Hence each particle $\mathbf{x}_{c,1:N_p}{}^{(i)}$ is linear in $\mathbf{u}_{0:\tau-1}$.

Summary

We have shown that the deterministic optimization problem posed in Table 1 can be encoded as a MILP. The program involves continuous decision variables; the control sequence, the state trajectory of each particle and the schedule; and binary decision variables; these indicate success of different sets of activities for individual particles. Extremely efficient commercially-available solvers exist that guarantee finding the global optimum to MILPs in finite time. By restricting the size of the particle set, we ensure that this optimization is tractable for real-time computation. Furthermore, as the number of particles converges to infinity, the approximated limited-horizon robust execution problem converges to the exact problem.

Simulation Results

We implemented the receding-horizon particle control approach on a PC with a 2.80GHz Pentium 4 processor. We use the AUV example illustrated in Fig. 1 with $\Delta t = 10s$, a planning horizon of $200s$, and an execution horizon of $10s$. Fig. 3 shows a typical solution to the single-stage limited-horizon robust execution at the beginning of the mission using 50 particles. The solution ensures that the approximated probability of the AUV missing the bloom region is at most 0.02. The AUV makes use of its full pitch angle capability (15°) while guaranteeing that the approximated probability of violating its operational limits is less than 10^{-6}. Fig. 4 shows the MILP solution time taken to solve the single-stage execution problem, averaged over each iteration in the AUV mission. With 20 particles or fewer, the optimization is fast enough to be solved in real-time (in the average case) with an execution horizon of $10s$. Above this value, the execution horizon must be increased, or the optimization must be terminated before a global optimum has been found.

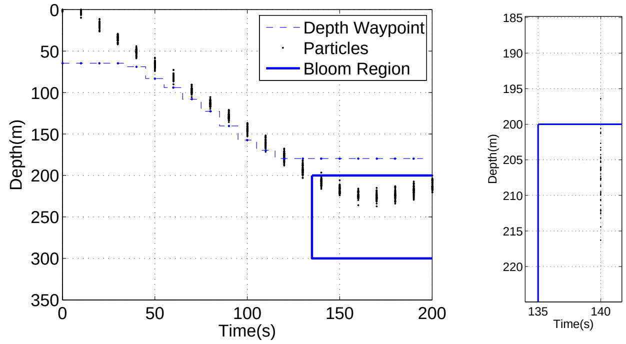

Figure 3. Left: Typical solution to single-stage limited-horizon robust execution problem. **Right:** Plan ensures that at most 2% of the particles fail to satisfy `Remain in [bloom region]`

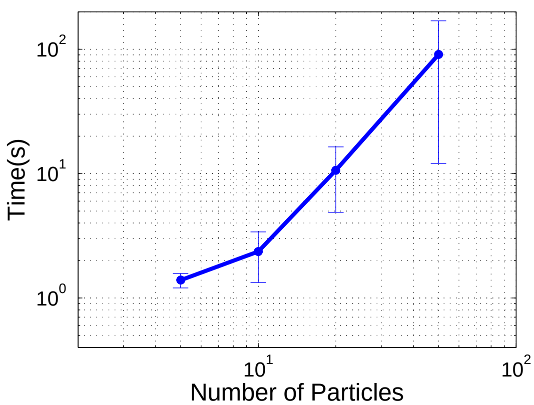

Figure 4. Average MILP solution time against size of particle set. Error bars represent one standard deviation.

Conclusion

We have presented a model-based executive for stochastic hybrid discrete-continuous systems that plans explicitly for uncertainty, ensuring that constraints on the probability of failure are satisfied. We use a particle approximation to make the planning problem tractable. As the number of particles tends to infinity, the approximation becomes exact. This gives an any-time approach to planning under stochastic uncertainty.

References

Ambros-Ingerson, J. A., and Steel, S. 1988. Integrating planning, execution and monitoring. In *Proc. Seventh National Conf. on AI*. AAAI Press.

Blackmore, L.; Bektassov, A.; Ono, M.; and Williams, B. C. 2007. Robust, optimal predictive control of jump markov linear systems using particles. In Bemporad, A.; Bicchi, A.; and Buttazzo, G., eds., *Hybrid Systems: Computation and Control*, Lecture Notes in Computer Science, 4416. Springer-Verlag. 104–117.

Blackmore, L. 2006. A probabilistic particle control approach to optimal, robust predictive control. In *Proc. AIAA Guidance, Navigation and Control Conf.*

Chien, S.; Knight, R.; Stechert, A.; Sherwood, R.; and Rabideau, G. 2000. Using iterative repair to improve responsiveness of planning and scheduling. In *Proceedings of the Fifth International Conference on Artificial Intelligence Planning and Scheduling.*

Costa, O. L. V.; Fragoso, M. D.; and Marques, R. P. 2005. *Discrete-Time Markov Jump Linear Systems.* Springer.

Domshlak, C., and Hoffmann, J. 2006. Fast probabilistic planning through weighted model counting. In *Proceedings of the 16th International Conference on Automated Planning and Scheduling (ICAPS-06).*

Hessem, D. H. V. 2004. *Stochastic Inequality Constrained Closed-loop Model Predictive Control.* Ph.D. Dissertation, TU Delft, Delft, The Netherlands.

Leaute, T., and Williams, B. C. 2005. Coordinating agile systems through the model-based execution of temporal plans. In *Proc. 20th National Conference on Artificial Intelligence.*

Li, P.; Wendt, M.; and Wozny, G. 2000. Robust model predictive control under chance constraints. *Computers and Chemical Engineering* 24:829–834.

Ljung, L. 1999. *System Identification: Theory for the user.* New York: Prentice Hall.

Majercik, S. M., and Littman, M. L. 1998. MAXPLAN: A new approach to probabilistic planning. *Artificial Intelligence Planning Systems* 86–93.

McEwen, R., and Streitlien, K. 2001. Modeling and control of a variable-length auv. In *Proc. Unmanned Untethered Submersible Technology Conference.*

Morris, P.; Muscettola, N.; and Tsamardinos, I. 1998. Reformulating temporal plans for efficient execution. In *Proc. International Conference on Principles of Knowledge Representation and Reasoning*, 444–452.

Richalet, J.; Rault, A.; Testud, J.; and Papon, J. 1976. Algorithmic control of industrial processes. In *Proceedings of the 4th IFAC Symposium on Identification and System Parameter Estimation*, 1119–1167.

Smith, D. E., and Weld, D. S. 1998. Conformant graphplan. In *Proc. 15th National Conf. AI.*

Tsamardinos, I.; Pollack, M. E.; and Ramakrishnan, S. 2003. Assessing the probability of legal execution of plans with temporal uncertainty. In *Proc. ICAPS.*

Vidal, T., and Ghallab, M. 1996. Dealing with uncertain durations in temporal constraint networks dedicated to planning. In *Proc. ECAI*, 48–52.

Wilkins, D. E., and Myers, K. L. 1995. A common knowledge representation for plan generation and reactive execution. *J. Logic Computation* 5(6):731–761.

Williams, B. C.; Ingham, M.; Chung, S. H.; and Elliott, P. H. 2003. Model-based programming of intelligent embedded systems and robotic space explorers. *Proceedings of the IEEE* 91(1):212–237.