Computer Science and Artificial Intelligence Laboratory
Technical Report

MIT-CSAIL-TR-2008-013 March 6, 2008

Efficient Motion Planning Algorithm for Stochastic Dynamic Systems with Constraints on Probability of Failure

Masahiro Ono and Brian C. Williams

massachusetts institute of technology, cambridge, ma 02139 usa — www.csail.mit.edu

Efficient Motion Planning Algorithm for Stochastic Dynamic Systems with Constraints on Probability of Failure [*]

Masahiro Ono and **Brian C. Williams**
Massachusetts Institute of Technology
77 Massachusetts Avenue
Cambridge, MA 02139
hiro_ono@mit.edu, williams@mit.edu

Abstract

When controlling dynamic systems such as mobile robots in uncertain environments, there is a trade off between risk and reward. For example, a race car can turn a corner faster by taking a more challenging path. This paper proposes a new approach to planning a control sequence with guaranteed risk bound. Given a stochastic dynamic model, the problem is to find a control sequence that optimizes a performance metric, while satisfying chance constraints i.e. constraints on the upper bound of the probability of failure. We propose a two-stage optimization approach, with the upper stage optimizing the risk allocation and the lower stage calculating the optimal control sequence that maximizes the reward. In general, upper-stage is a non-convex optimization problem, which is hard to solve. We develop a new iterative algorithm for this stage that efficiently computes the risk allocation with a small penalty to optimality. The algorithm is implemented and tested on the autonomous underwater vehicle (AUV) depth planning problem, which demonstrates the substantial improvement in computation cost and suboptimality compared to the prior arts.

Introduction

Physically grounded AI systems typically interact with their environment through a hybrid of discrete and continuous actions. Two important capabilities for such systems are kinodynamic motion planning and plan execution on a hybrid discrete/continuous plant. For example, our application is an autonomous underwater vehicle (AUV) shown in Figure 1, which conducts a bathymetric mapping mission for up to 20 hours without human supervision. This system should ideally navigate itself to areas of scientific interest according to a game plan provided by scientists. Since the AUV's maneuverability is limited, it needs to plan its path with taking vehicle dynamics into account, in order to avoid collisions with the seafloor. A model-based executive, called Sulu (Léauté 2005), implemented these two capabilities in deterministic environment for AUVs.

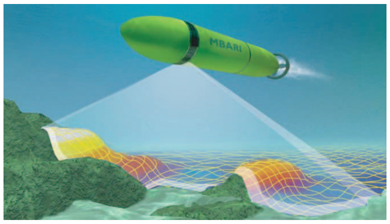


Figure 1: Autonomous underwater vehicle *Dorado* of Monterey Bay Aquarium Research Institute, conducting bathymetric mapping mission

Real-world systems, however, are exposed to stochastic disturbances. Stochastic systems typically have a risk of failure due to unexpected events, such as unpredictable tides and currents that affect the AUV's motion. To reduce the risk of failure, the AUV needs to stay away from the failure states, such as the seafloor. This has the consequence of reducing mission performance, since it prohibits high resolution observation of the seafloor. Thus operators of stochastic systems need to trade-off risk and performance.

A common approach to trading-off risk and performance is to define a positive reward for mission achievement and a negative reward for failure, and then optimize the expected reward using a Markov Decision Process (MDP) encoding. However, in many practical cases, only an arbitrary definition of reward is possible. For example, it is hard to define the value of scientific discovery compared to the cost of losing the AUV.

Another approach to trading off the risk and performance is to limit the probability of failure (*chance constraint*) and maximize the performance under this constraint. For example, an AUV minimizes the average altitude from the seafloor while limiting the probability of collision to 0.1%.

There is a considerable body of work on this approach in Robust Model Predictive Control (RMPC) community. If the distribution of disturbance is bounded, zero failure probability can be achieved by sparing the safety margin between the failure states and the nominal states (Kuwata, Richards, & How 2007). If the distribution is unbounded, which is the case in many practical applications, the chance constraint needs to be considered. When only the probability of fail-

[*] This research is funded by The Boeing Company grant MIT-BA-GTA-1

ure of each individual time step is constrained (*chance constraints at individual time steps)*, the stochastic problem can be easily reduced to a deterministic problem by constraint tightening(Yan & Bitmead 2005)(van Hessem 2004).

A challenge arises when the probability of failure of entire mission is constrained (*chance constraint over the mission*). This is the case in many practical applications; for example, an AUV operator would like to limit the probability of losing it in a mission, rather than in each time instant. The chance constraint over mission can be decomposed into chance constraints at individual time steps using an ellipsoidal relaxation technique (van Hessem 2004). However, the relaxation is very conservative, hence the result is significantly suboptimal.

A sample based algorithm called Particle Control (Blackmore 2006) uses Mixed Integer Linear Programming (MILP) to directly optimize the control sequence. The algorithm can handle the probability of failure over the mission directly without using the conservative bound. However, it is slow when it is applied to the problem like goal-directed execution of temporally flexible plans (Léauté & Williams 2005), due to the large dimension of the decision vector. Another important issue with Particle Control is that, although there is a converging guarantee to the true optimum when the number of the samples goes to infinity, there is no guarantee that the original chance constraint is satisfied with finite number of samples.

We propose a new fast algorithm called Bi-stage Robust Motion Planning (BRMP), which plans the control sequence with small suboptimality and strict guarantee of satisfying a chance constraint over a mission. There are two key contributions regarding to the BRMP algorithm; the first is the introduction of a bi-stage optimization approach, with the upper stage optimizing the risk allocation and the lower stage optimizing the control sequence. The second is the development of a risk allocation algorithm for the upper stage, called Iterative Risk Allocation (IRA). Although IRA does not offer a guarantee of the convergence to the global optima, it does have the guarantee of monotonic increase of the objective function over iterations. Simulation results on our implementation demonstrates a substantial improvement in suboptimality compared to the ellipsoidal relaxation approach, while achieving a significant speed up compared to the Particle Control.

The rest of paper is outlined as follows. The next section introduces the notion of risk allocation, followed by the formal problem statement. Then the two key ideas in BRMP, the bi-stage optimization approach and the Iterative Risk Allocation algorithm, are presented. The BRMP algorithm is implemented on the case with linear dynamics and Gaussian distribution, and applied to AUV navigation problem, on which the performance of BRMP is compared with ellipsoidal relaxation approach and Particle Control.

Risk Allocation

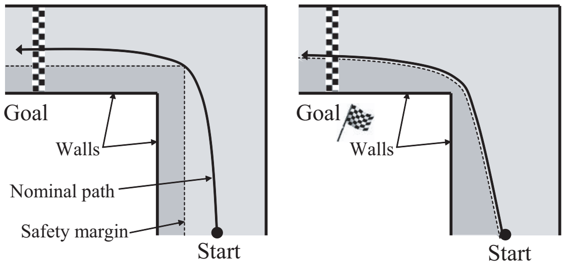
Racing Car Example Imagine the racing car example shown in Figure 2. The task is to plan a control sequence of wheel and acceleration that minimizes the time to reach a goal, with the guarantee that the probability of crashing into a wall during the race is less than a certain probability, say, 0.1% (chance constraint over mission). Planning the control sequence is equivalent to planning the nominal path, which is shown as the solid lines in the Figure 2. We assume that the dynamics of the vehicle is stochastic and the distribution of uncertainty is unbounded. To limit the probability of crashing into the wall, a good driver would set the safety margin, which is colored in dark gray in Figure 2, and then plan the nominal path that does not penetrate the safety margin. In other words, the driver *tightens* the original constraints (the walls) and set new constraints on the nominal path, which is shown as the dotted line.

The driver wants to set the safety margin as small as possible to make the path shorter. However, since the probability of crash during the race is bounded, there is a certain lower bound on the total size of the safety margin. We assume here that the total area of the safety margin is lower-bounded. Given this constraint, there are different strategies of setting a safety margin; in Figure 2(a) the width of the margin is uniform; in Figure 2(b) the safety margin is narrow around the corner, and wide at the other places.

An intelligent driver would take the strategy of (b), since he knows that going closer to the wall at the corner is effective to make the path shorter while doing so at the straight line is not. A key observation here is that taking a risk (i.e. setting narrow safety margin) at the corner results in a greater reward (i.e. time saving) than taking the same risk at the straight line. This gives rise to the notion of *risk allocation*. The good risk allocation strategy is to save risk when the reward is small while taking it when the reward is great.

Another important observation is, once risk is allocated and the safety margin is fixed (i.e. chance constraint *over the mission* is decomposed into chance constraints at *individual time steps*), the stochastic performance optimization problem with chance constraint over the mission has been reduced to a deterministic nominal path planning problem with tightened constraints. This can be solved quickly with existing deterministic path planning algorithms.

These two observations naturally lead to bi-stage optimization approach (Figure 3), in which its upper stage allocates risk to each time step while its lower stage tightens constraints according to the risk allocation and solves the resulting deterministic problem.



(a) Uniform risk allocation (b) Optimal risk allocation

Figure 2: Risk allocation strategies on the racing car example

The next section formally states the problem, and the subsequent section formally describes the bi-stage optimization algorithm, called Bi-stage Robust Motion Planning.

Formal Problem Statement

Our goal is to develop a method that can generalize to planning over either continuous or discrete state spaces, such as kinodynamic path planning and PDDL planning.

Let $\boldsymbol{x}_t \in \mathcal{X}$, $\boldsymbol{u}_t \in \mathcal{U}$, and $\boldsymbol{w}_t \in \mathcal{W}$ denote the state vector, control input (action) vector, and disturbance vector at time step t, respectively. For example, in AUV navigation case, $\boldsymbol{x}$ is position and velocity of the vehicle, $\boldsymbol{u}$ is ladder angle and throttle position, and $\boldsymbol{w}$ is the uncertainty in position and velocity. The domains $\mathcal{X}$, $\mathcal{U}$ and $\mathcal{W}$ may be a continuous state space, discrete state space, or a hybrid of both. The uncertainty model of $\boldsymbol{w}_t$ is given as a probability distribution function $f : \mathcal{W} \rightarrow [0, 1]$.

$$\boldsymbol{w}_t \sim f(\boldsymbol{w}) \tag{1}$$

The stochastic dynamics model for a continuous space or the state transition model for a discrete space is defined as follows

$$\boldsymbol{x}_{t+1} = g(\boldsymbol{x}_t, \boldsymbol{u}_t, \boldsymbol{w}_t) \tag{2}$$

where $g : \mathcal{X} \times \mathcal{U} \times \mathcal{W} \rightarrow \mathcal{X}$ is the state transition function. Note that $\boldsymbol{x}$ is a random variable while $\boldsymbol{u}$ is deterministic.

Assuming that the initial state $\boldsymbol{x}_0$ is known deterministically, the *nominal states* $\bar{\boldsymbol{x}}_t \in \mathcal{X}$ are defined as the sequence of deterministic states evolved from $\boldsymbol{x}_0$ along Eq. (2) without disturbances, such that

$$\bar{\boldsymbol{x}}_{t+1} = g(\bar{\boldsymbol{x}}_t, \boldsymbol{u}_t, 0). \tag{3}$$

Let $\mathcal{R}_t \subset \mathcal{X}$ denote the *feasible region* at time step t. In AUV navigation case, $\mathcal{R}$ is the ocean above the seafloor. A mission is *failed* when $\boldsymbol{x}_t$ is out of this region at any time step in the mission duration $t \in [0, T]$. The *probability of failure* over the mission P_{Fail} is defined as follows

$$P_{Fail} = \Pr[(\boldsymbol{x}_1 \notin \mathcal{R}_1) \vee (\boldsymbol{x}_2 \notin \mathcal{R}_2) \vee \cdots \vee (\boldsymbol{x}_T \notin \mathcal{R}_T)]. \tag{4}$$

The *chance constraint over the mission* is the upper bound of the probability of failure over the mission

$$P_{Fail} \leq \delta. \tag{5}$$

Finally, the objective function (i.e. reward) J is given as a function $h : \mathcal{X}^T \times \mathcal{U}^T \rightarrow \mathbb{R}$ that is defined on the sequence of nominal states and control inputs;

$$J = h(\bar{\boldsymbol{x}}_{1:T}, \boldsymbol{u}_{1:T}). \tag{6}$$

The problem is formulated as an optimization of control (action) sequence $\boldsymbol{u}_{1:T}$ that maximizes the objective function Eq.(6) given the state transition model, uncertainty model, and the chance constraint.

Problem 1: Control Sequence Optimization with Chance Constraint

$$\begin{array}{ll} \text{Maximize} & J = h(\bar{\boldsymbol{x}}_{1:T}, \boldsymbol{u}_{1:T}) \\ s.t. & \text{Eq.}(1), (2), \text{and}(5). \end{array}$$

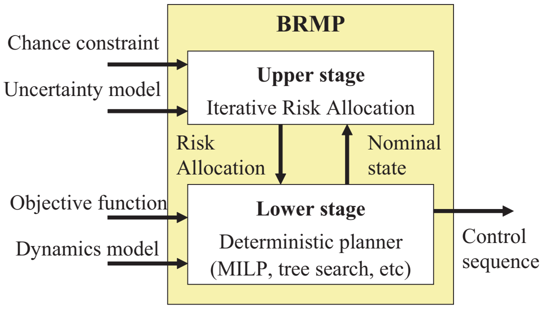


Figure 3: Architecture of Bi-stage Robust Motion Planning

Bi-stage Robust Motion Planning algorithm

Our approach to solving Problem 1 is the Bi-stage Robust Motion Planning (BRMP) algorithm (Figure 3). As described in the previous sections, the chance constraint *over the mission* is decomposed into chance constraints *at individual time steps* by risk allocation. It results in the bi-stage optimization approach, which is the first important contribution in this paper.

Decomposition of chance constraint over the mission

The probability of failure at time step t is defined as follows:

$$P_{Fail,t} = \Pr[\boldsymbol{x}_t \notin \mathcal{R}_t]. \tag{7}$$

Using the union bound or Boole's inequality ($Pr[A \cup B] \leq \Pr[A] + \Pr[B]$), it can be easily shown that the original chance constraint Eq.(5) is implied by the following conjunction (Blackmore & Williams 2006)

$$\bigwedge_{t=1}^{T} P_{fail,t} \leq \delta_t \tag{8}$$

$$\wedge \quad \sum_{t=1}^{T} \delta_t \leq \delta \tag{9}$$

where Eq.(8) refers to the *chance constraints at individual time steps*. *Risk allocation* means assigning values to $(\delta_1, \delta_2 \cdots, \delta_T)$. Once the risk is allocated so that Eq.(9) is satisfied, the original chance constraint over the mission Eq.(5) is replaced by a set of chance constraints at individual time steps Eq.(8).

Thus the original optimization problem (Problem 1) can be decomposed into risk allocation optimization (upper-stage) and control sequence optimization with chance constraints at individual time steps (lower-stage), which is described in the next subsection.

Lower-stage optimization

The stochastic optimization problem with the chance constraints at individual time steps (Eq.(8)) is reduced to the deterministic planning problem of the nominal states $\bar{\boldsymbol{x}}$ by constraint tightening (i.e. setting a safety margin) (Yan &

Bitmead 2005)(van Hessem 2004). Safety margin at t (denoted by $\mathcal{M}_t$) is calculated so that the following conditional probability is bounded by the given risk assignment δ_t.

$$Pr[\boldsymbol{x}_t \notin \mathcal{R}_t \mid \bar{\boldsymbol{x}}_t \in (\mathcal{R}_t - \mathcal{M}_t)] \le \delta_t \tag{10}$$

The distribution of $\boldsymbol{x}_t$ can be calculated a priori from Eq.(1) and Eq.(2). Given the safety margin $\mathcal{M}_t$, the chance constraints at individual time steps t (Eq.(8)) are implied by the following tightened constraints on the nominal states, which are deterministic.

$$[(\bar{\boldsymbol{x}}_1 \in (\mathcal{R}_1 - \mathcal{M}_1)] \wedge \cdots \wedge [(\bar{\boldsymbol{x}}_T \in (\mathcal{R}_T - \mathcal{M}_T)] \tag{11}$$

The lower stage optimization problem is to find the control sequence $\boldsymbol{u}_{1:T}$ which maximizes the objective function Eq.(6) given the tightened constraints Eq.(11).

Problem 2: Lower-stage Optimization

$$\begin{aligned} &\text{Maximize} J = h(\bar{\boldsymbol{x}}_{1:T}, \boldsymbol{u}_{1:T}) \\ &s.t. \qquad \text{Eq.}(3) \text{ and } (11) \end{aligned}$$

No random variables are involved in this optimization problem. It can be solved by existing deterministic planning methods. For hybrid state space with linear dynamics(Eq.(2)), Mixed-integer Linear Programing (MILP) (Richards *et al.* 2002) is widely used. For discrete state space, standard tree search algorithms can be used.

For later use, this optimization process is expressed as a function of the risk allocation as follows;

$$LSO(\delta_1 \cdots \delta_T) = \max_{\boldsymbol{u}_{1:T}} J \quad s.t. \quad \text{Eq.}(3), (10), \text{and}(11). \tag{12}$$

Upper-stage Optimization

The upper-stage optimizes the risk allocation $\delta_1 \cdots \delta_T$ according to the constraint Eq.(9).

Problem 3: Upper-stage Optimization

$$\begin{aligned} &\text{Maximize} \quad LSO(\delta_1 \cdots \delta_T) \\ &s.t. \qquad \text{Eq.}(9) \end{aligned}$$

The question is how to optimize Problem 3. In general it is non-convex optimization problem, which is very hard to solve. The next section introduces the second important contribution of this paper, a risk allocation algorithm for the upper stage called Iterative Risk Allocation.

Iterative Risk Allocation Algorithm

The Iterative Risk Allocation (IRA) algorithm (Algorithm 1) solves Problem 3 with iterations. It has a parameter $0 < \alpha < 1$. In Line 4, the lower-stage optimization function LSO (Eq.(12)) is modified so that it also outputs the resulting nominal state sequence $\bar{\boldsymbol{x}}_{1:T}$. A constraint is active at time t iff the nominal state $\bar{\boldsymbol{x}}$ is on the boundary of $(\mathcal{R}_t - \mathcal{M}_t)$. The graphical interpretation is that the constraint is active when the nominal path touches the safety margin (Figure 2 and 5). In Line 7, $\Pr(\boldsymbol{x}_t \notin \mathcal{R}_t \mid \bar{\boldsymbol{x}}_t)$ is the actual probability of failure at time t given the nominal state $\bar{\boldsymbol{x}}_t$. It is equal to δ_t only when the constraint is active, and otherwise it is less than δ_t.

Algorithm 1 Iterative Risk Allocation

```
 1: $\forall t \quad \delta_t \leftarrow \delta/T$
 2: while $J - J_{prev} > \epsilon$ do
 3:     $J_{prev} \leftarrow J$
 4:     $[J, \bar{\boldsymbol{x}}_{1:T}] \leftarrow LSO(\delta_1 \cdots \delta_T)$
 5:
 6:     for all $t$ such that constraint is inactive at $t$ th step do
 7:         $\delta_t \leftarrow \alpha\delta_t + (1-\alpha)\Pr(\boldsymbol{x}_t \notin \mathcal{R}_t \mid \bar{\boldsymbol{x}}_t)$
 8:     end for
 9:     $\delta_{res} \leftarrow \delta - \sum_{t=1}^{T} \delta_t$
10:     $N_{active} \leftarrow$ number of steps where constraint is active
11:     for all $t$ such that constraint is active at $t$ th step do
12:         $\delta_t \leftarrow \delta_t + \delta_{res}/N_{active}$
13:     end for
14: end while
```

In every loop of the algorithm the nominal path is planned using the lower-stage optimization given the current risk allocation (Line 4). Risk assignment is decreased when the constraint is inactive (Line 7), and it is increased when the constraint is active (Line 12). Line 9 and 12 ensure that $\sum_{t=1}^{T} \delta_t = \delta$ so that the suboptimality due to the union bound is minimized.

There are two important features of this algorithm, which are described in the following theorems.

Theorem 1 The objective function J monotonically increases over the iteration of Algorithm 1.

Proof. If there are inactive constraints, they keep being inactive after Line 7 since $\alpha < 1$. Thus, the objective function J does not change at this point. Then in Line 12, the active constraints are relaxed, so J increases. If there are no inactive constraints, then $\delta_{res} = 0$ and thus risk assignments of active constraints do not change. So consequently the objective function does not change as well. □

The proof of Theorem 1 also implies another important theorem.

Theorem 2 Algorithm 1 converges if and only if all constraints are active.

Note that Algorithm 1 has no convergence guarantee to the global optima. However, Theorem 1 ensure that if δ_t is initialized with the risk allocation obtained from ellipsoidal relaxation approach, the result of Algorithm 1 is no worse than that of the ellipsoidal relaxation approach. Our empirical results demonstrate that Algorithm 1 yields much less conservative result when started from the simple uniform risk allocation $\delta_t = \delta/T$ ($t = 1 \cdots T$) (Line 1 of Algorithm 1).

One additional note is that an interesting property of the parameter α is observed in the simulation; as α becomes large, convergence becomes faster but suboptimality gets larger, as shown in Figure 4. This property enables a trade-off between suboptimality and computational cost.

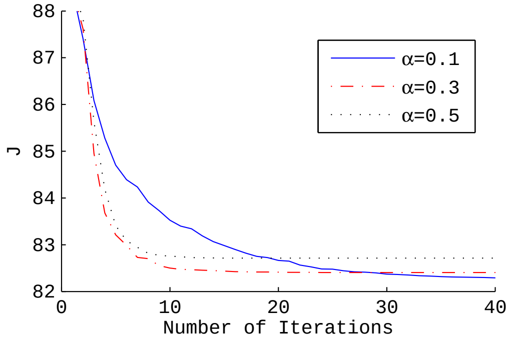


Figure 4: Convergence of the iterative risk allocation algorithm with different settings of α

Linear Time Invariant System with Gaussian Disturbance

In many practical applications, a continuous system can often be approximated as a linear time-invariant (LTI) system with Gaussian disturbances. The general form of the BRMP algorithm derived in the previous sections are deployed to the linear Gaussian case in this section.

The state and action domain is continuous $\mathcal{X} = \mathbb{R}^{n_x}$ and $\mathcal{U} = \mathbb{R}^{n_u}$. The deterministic constraints such as actuator saturation is addressed by adding linear constraints on $\boldsymbol{u}$, rather than limiting its domain $\mathcal{U}$. The state transition model (Eq.(2)) is linear as follows;

$$\boldsymbol{x}_{t+1} = \boldsymbol{A}\boldsymbol{x}_t + \boldsymbol{B}\boldsymbol{u}_t + \boldsymbol{w}_t. \tag{13}$$

The distribution of $\boldsymbol{w}$ (Eq.(1)) is zero-mean Gaussian with covariance matrix Σ_w.

$$\boldsymbol{w} \sim \mathcal{N}(\boldsymbol{0}, \Sigma_w) \tag{14}$$

Then the distribution of $\boldsymbol{x}_t$ is also Gaussian with the covariance matrix given as

$$\Sigma_{x,t} = \sum_{k=0}^{t-1} \boldsymbol{A}^k \Sigma_w (\boldsymbol{A}^k)^T. \tag{15}$$

The feasible region is defined by the conjunction of N_t linear constraints;

$$\mathcal{R}_t = \left\{ \boldsymbol{x}_t \in \mathcal{X} \ : \bigwedge_{i=1}^{N_t} \boldsymbol{h}_t^{iT}\boldsymbol{x}_t \le g_t^i \right\}. \tag{16}$$

Thus the chance constraint of individual time steps (Eq.(7)(8)) is described as follows

$$\Pr\left[\bigvee_{i=1}^{N_t} \boldsymbol{h}_t^{iT}\boldsymbol{x}_t > g_t^i\right] \le \delta_t. \tag{17}$$

This joint chance constraint can be again decomposed by risk allocation. The decomposition results in the set of chance constraints on the probability of violation of individual constraints. Thus Eq.(8) and (9) is replaced by the following;

$$\bigwedge_{t=1}^{T}\bigwedge_{i=1}^{N_t}\left(\Pr[\boldsymbol{h}_t^{iT}\boldsymbol{x}_t > g_t^i] \le \delta_t^i\right) \wedge \sum_{t=1}^{T}\sum_{i=1}^{N_t}\delta_t^i \le \delta. \tag{18}$$

The risk allocation problem of δ_t^i can be solved by the iterative risk allocation algorithm (Algorithm 1).

The constraint tightening $\mathcal{R} - \mathcal{M}$ in Eq.(11) is equivalent to reducing the upper bounds g_t^i of Eq.(16). The nominal states are bounded by the tightened constraints such that

$$\boldsymbol{h}_t^{iT}\bar{\boldsymbol{x}}_t \le g_t^i - m_t^i \tag{19}$$

$$m_t^i = \sqrt{2\boldsymbol{h}_t^{iT}\Sigma_{x,t}\boldsymbol{h}_t^i}\, erf^{-1}(1 - 2\delta_t^i) \tag{20}$$

where erf^{-1} is the inverse of the Gauss error function. The conditional probability of failure at time t in the Algorithm 1, Line 7 is replaced by the probability of violating the i th constraint at time t, which is equal to the cumulative distribution function.

$$\Pr(\boldsymbol{x}_t > g_t^i \mid \bar{\boldsymbol{x}}_t) = \frac{1}{2}\left(1 + erf\,\frac{\boldsymbol{h}_t^{iT}(\boldsymbol{x}_t - \bar{\boldsymbol{x}})}{\sqrt{2\boldsymbol{h}_t^{iT}\Sigma_{x,t}\boldsymbol{h}_t^i}}\right) \tag{21}$$

If the objective function (Eq.(6)) is also linear, the lower-stage optimization can be solved by Linear Programming. If it is quadratic, Quadratic Programming can be used.

Simulation: AUV Depth Planning

Problem Setting We assumed the case where an autonomous underwater vehicle (AUV) plans a path to minimize the average altitude from the sea floor, while limiting the probability of crashing into it. Tides and currents gives the AUV disturbance. The linear dynamics is discretized with interval $\Delta t = 5$. The AUV's horizontal speed is constant at 3.0 knots, so only the vertical position needs to be planned. The dynamics model is taken from the actual AUV developed by Monterey Bay Aquarium Research Institute (Figure 1), and the actual bathymetric data of the Monterey Bay is used. The deterministic planning algorithm used in the lower-stage has been demonstrated in the actual AUV mission.

The AUV has six real-value states and takes one real-value control input, thus $\mathcal{X} = \mathbb{R}^6$ and $\mathcal{U} = \mathbb{R}$. Disturbance w with $\sigma_w = 10$ [m] acts only on the third component of $\boldsymbol{x}$, which refers to the depth of the vehicle. The AUV's elevator angle and pitch rate are deterministically constrained.

The depth of the AUV is required to be less than the seafloor depth for the entire mission ($1 \le t \le 20$) with probability $\delta = 0.05$. The objective is to **minimize** (not maximize) the average of AUV's nominal altitude from the sea floor.

Algorithms tested The following three algorithms are implemented in Matlab and run on a machine with Pentium 4 2.80 GHz processor and 1.00 GB of RAM. The planning horizon is 100 seconds (20 time steps with 5 second time interval).

(a) Ellipsoidal relaxation approach (van Hessem 2004)
(b) Bi-stage Robust Motion Planning ($\alpha = 0.3$)
(c) Particle Control (20 particles) (Blackmore 2006)

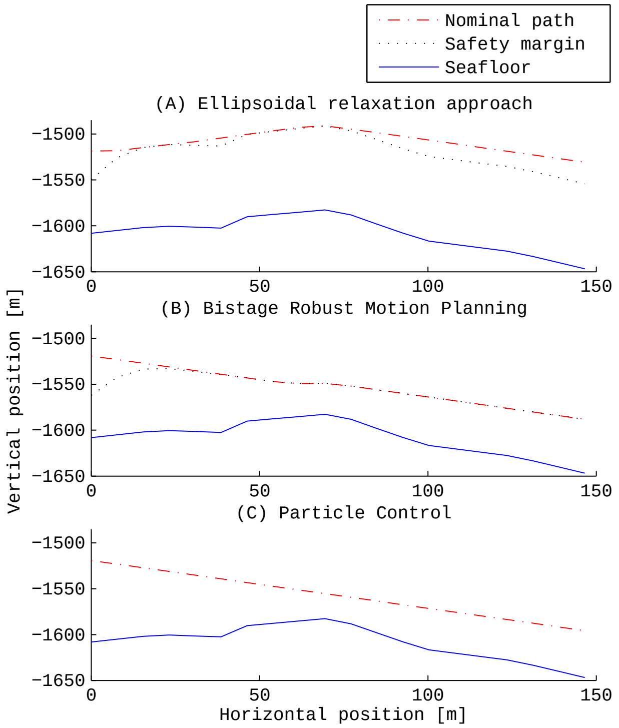


Figure 5: Nominal path of AUV and safety margin planned by three algorithms

Result Figure 5 shows the nominal paths and safety margins planned by three algorithms. Safety margin is not shown in (c) since Particle Control does not explicitly compute it. Ellipsoid relaxation yields a large safety margin, which touches the nominal path (i.e. constraint is active) only at a few points. This is because ellipsoid relaxation uniformly allocates risk to each step. On the other hand, Bi-stage Robust Motion Planning algorithm gives the safety margin that almost corresponds with the nominal path. This fact implies that risk is allocated efficiently such that a large portion of risk is allocated to the critical points, such as the top of the seamount.

The performance of the three algorithms is compared in Table 1. The resulting probability of failure P_{Fail} is evaluated by Monte Carlo simulation with 100,000 samples. The plan generated by the ellipsoidal relaxation approach ((a) ER) results in nearly zero probability of failure although the bound is $P_{Fail} \leq 0.05$, which shows its strong conservatism. Bi-stage Robust Motion Planning ((b) BRMP) is also conservative, but much less so than (a). On the other hand, the probability of failure of Particle Control ((c) PC) is higher than the bound, which means the violation of the chance constraint. This is because the Particle Control is a sample based stochastic algorithm, and the chance constraint is just approximately satisfied.

The value of objective function (reward) J is the measure of optimality. Note that this is a minimization problem, so smaller J means better performance. The true optimal value of J lies between (b) BRMP and (c) Particle Control, since the former is suboptimal and the latter is ”overoptimal” in the sense that it does not satisfy the chance constraint. Thus the suboptimality of the BRMP is less than 10%. On the other hand, ellipsoidal relaxation yields very large J, which reflects its large suboptimality.

Table 1: Performance comparison on the AUV depth planning problem with chance constraint $P_{Fail} \leq 0.05$.
(a) ER: Ellipsoid relaxation approach, (b) BRMP: Bi-stage Robust Motion Planning, PC: Particle Control

Algorithm used	(a) ER	(b) BRMP	(c) PC
Resulting P_{Fail}	$< 10^{-5}$	0.037	0.085
Objective function J	99.3	55.2	51.1
Computation time [sec]	1.9	4.1	481.2

The computation time of Particle Control is longer than the planning horizon (100 sec). Although BRMP is slightly slower than the ellipsoidal relaxation approach, it achieved a substantial speed up from Particle Control.

Conclusion

In this paper, we have developed a new algorithm called Bi-stage Robust Motion Planning (BRMP). It computes the control sequence that maximizes an objective function while satisfying a chance constraint over the mission. It consists of two stages; upper stage that allocates risk to each time step, and lower stage that tightens constraints according to the risk allocation and solves the resulting deterministic problem. Risk allocation in the upper stage can be efficiently computed by Iterative Risk Allocation algorithm. The BRMP algorithm is implemented on the case with linear dynamics and Gaussian distribution, and applied to AUV navigation problem. It is demonstrated that BRMP achieves substantial speed up compared to Particle Control, with much less suboptimality compared to a ellipsoidal relaxation approach.

Acknowledgments

Thanks to Michael Kerstetter, Scott Smith and the Boeing Company for their support. Kanna Rajan, David Caress, Conor McGann, and Frederic Py at Monterey Bay Aquarium Research Institute for their collaboration. Lars Blackmore provided valuable feedback.

References

Blackmore, L., and Williams, B. C. 2006. Optimal manipulator path planning with obstacles using disjunctive programming. In *Proceedings of the American Control Conference*.

Blackmore, L. 2006. A probabilistic particle control approach to optimal, robust predictive control. In *Proceedings of the AIAA Guidance, Navigation and Control Conference*.

Kuwata, Y.; Richards, A.; and How, J. 2007. Robust receding horizon control using generalized constraint tightening. *Proceedings of American Control Conference*.

Léauté, T., and Williams, B. C. 2005. Coordinating agile systems through the model-based execution of temporal plans. In *Proceedings of the Twentieth National Conference on Artificial Intelligence (AAAI)*.

Léauté, T. 2005. Coordinating agile systems through the model-based execution of temporal plans. Master's thesis, Massachusetts Institute of Technology.

Richards, A.; Schouwenaars, T.; How, J. P.; and Feron, E. 2002. Spacecraft trajectory planning with avoidance constraints using mixed-integer linear programming. *AIAA Journal of Guidance, Control, and Dynamics*.

van Hessem, D. H. 2004. *Stochastic inequality constrained closed-loop model predictive control with application to chemical process operation*. Ph.D. Dissertation, Delft University of Technology.

Yan, J., and Bitmead, R. R. 2005. Incorporating state estimation into model predictive control and its application to networktraffic control. *Automatica*.