An 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 a 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 reward. In general, the 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, and demonstrates a 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 a bathymetric mapping mission using Dorado-class autonomous underwater vehicle (AUV) (Figure 1) operated by the Monterey Bay Aquarium Research Institute (MBARI). Dorado-class mapping AUV is 6,000 m rated, and can operate for up to 20 hours without human supervision. This system should ideally navigate itself to areas of scientific interest, such as underwater canyons, according to a game plan provided by scientists. Since the AUV's maneuverability is limited, it needs to plan its path while 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 using deterministic model.

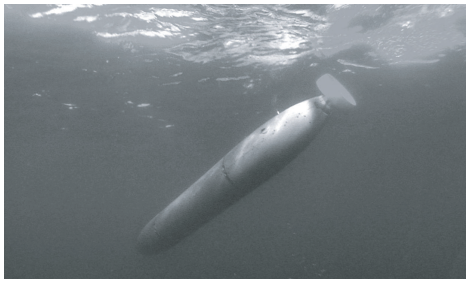


Figure 1: Dorado-class autonomous underwater vehicle of Monterey Bay Aquarium Research Institute

Real-world systems can be exposed to significant levels of stochastic disturbance. 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 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 with 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 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 within 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,

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

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 failure 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 the entire mission is constrained (*a 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 the vehicle throughout the mission, rather than during each time instant. The chance constraint over the entire 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) represents state distributions by samples and 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 a conservative bound. However, the algorithm is slow when it is applied to many problems such as 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 soundness. Although there is a convergence guarantee to the true optimum as the number of the samples goes to infinity, there is no guarantee that the original chance constraint is satisfied for a finite number of samples.

We propose a new fast algorithm called Bi-stage Robust Motion Planning (BRMP), which is only slightly suboptimal and offers strict guarantee of satisfying a chance constraint over a mission. The BRMP algorithm makes two key contributions; 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 successive 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 this paper is outlined as follows. The next section introduces the notion of risk allocation, followed by a formal problem statement. Next, the two key ideas in BRMP, the bi-stage optimization approach and the Iterative Risk Allocation algorithm, are presented. The BRMP algorithm is implemented for systems with linear dynamics and Gaussian distributed noise. It is applied to AUV navigation problem, for which the performance of BRMP is compared with the ellipsoidal relaxation approach and Particle Control.

Risk Allocation

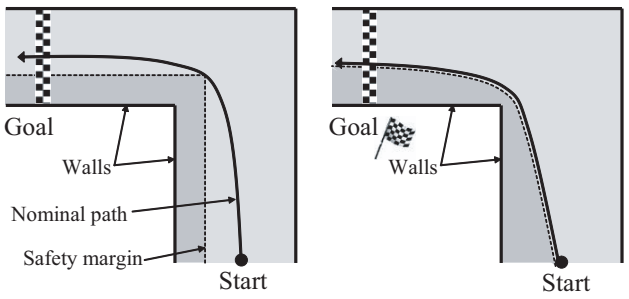
Racing Car Example Consider the race car example shown in Figure 2. The task is to plan a control sequence of steering and acceleration that minimizes the time to reach a goal, with the guarantee that the probability of crashing into a wall during the race is less than a certain probability, say, 0.1% (the chance constraint over the mission). Planning the control sequence is equivalent to planning the nominal path, which is shown by the solid lines in 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 a safety margin along the wall, and then plan a nominal path that does not penetrate the safety margin. (Safety margins are the areas colored in dark gray in Figure 2). In other words, the driver *tightens* the feasibility constraints.

The driver wants to set the safety margin as small as possible, in order 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 has a lower bound. 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 other places.

An intelligent driver would take the strategy of (b), since he knows that going closer to the wall at the corner is more effective at making the path shorter than at the straight line is not. A key observation here is that taking risk (i.e. setting a 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*. A 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. a chance constraint *over the mission* is decomposed into chance constraints at *individual time steps*), the stochastic optimization problem has been reduced to a deterministic nominal path planning problem with tightened constraints. This can be solved quickly with existing deterministic path planning algorithms.



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

Figure 2: Risk allocation strategies on the racing car example

These two observations naturally lead to a 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.

The next section formally states the problem, and the subsequent section 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, for AUV navigation, $\boldsymbol{x}$ denotes position and velocity of the vehicle, $\boldsymbol{u}$ denotes ladder angle and throttle position, and $\boldsymbol{w}$ denotes the uncertainty in position and velocity. The domains $\mathcal{X}$, $\mathcal{U}$ and $\mathcal{W}$ may be continuous (i.e. real-valued), discrete, 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 $\boldsymbol{x}_0$ according to Eq. (2) without disturbances, that is:

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

Let $\mathcal{R}_t \subset \mathcal{X}$ denote the *feasible region* at time step t. For example of AUV navigation, $\mathcal{R}$ corresponds to the ocean above the seafloor. A mission *fails* when $\boldsymbol{x}_t$ is out of this region at any time within 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 an upper bound on 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 over control (action) sequences $\boldsymbol{u}_{1:T}$ that maximizes the objective function Eq.(6), given the state transition model, uncertainty model, and the chance constraint.

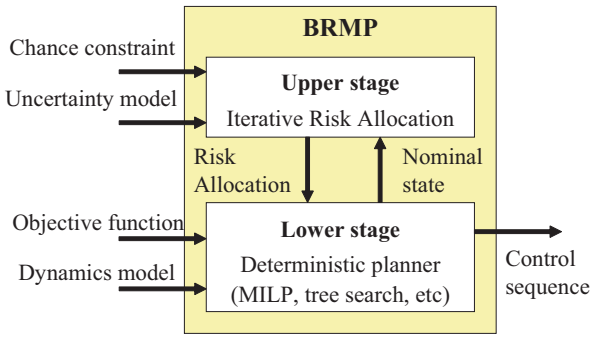


Figure 3: Architecture of Bi-stage Robust Motion Planning

Problem 1: Control Sequence Optimization with Chance Constraint

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

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 at 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):

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

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

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 (the upper-stage of BRMP) and control sequence optimization with chance constraints at individual time steps (the lower-stage). The lower-stage is described in the next subsection, followed by the upper-stage.

Lower-stage optimization

The stochastic optimization problem with chance constraints at individual time steps (Eq.(8)) is reduced to deterministic planning problem over nominal states $\bar{\boldsymbol{x}}$ using constraint tightening; constraint tightening corresponds intuitively to setting a safety margin (Yan & Bitmead 2005)(van Hessem 2004). The safety margin at t (denoted $\mathcal{M}_t$) is calculated so that the following conditional probability is bounded by the given risk assignment δ_t:

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

Since the distribution of $\boldsymbol{x}_t$ can be calculated a priori from Eq.(1) and Eq.(2), $\mathcal{M}_t$ can also be computed off-line and compiled in a table. For many common distributions such as Gaussian, $\mathcal{M}_t$ can be derived analytically.

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.

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

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

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

No random variables are involved in this optimization problem. It can be solved by existing deterministic planning methods. For a hybrid state space with a linear dynamics(Eq.(2)), Mixed-integer Linear Programing (MILP) is widely used. For a 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;

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

Upper-stage Optimization

The upper-stage optimizes the risk allocation $\delta_1 \cdots \delta_T$ according to 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 a 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.

Algorithm 1 Iterative Risk Allocation

```
1: ∀t   δ_t ← δ/T
2: while J − J_prev > ε do
3:     J_prev ← J
4:     [J, x̄_{1:T}] ← LSO(δ_1 ··· δ_T)
5:     N_active ← number of steps where constraint is active
6:     if N_active = 0 or N_active = T then
7:         break;
8:     end if
9:
10:    for all t such that constraint is inactive at t th step do
11:        δ_t ← (1 − α)δ_t + α Pr(x_t ∉ R_t | x̄_t)
12:    end for
13:    δ_res ← δ − Σ_{t=1}^{T} δ_t
14:    for all t such that constraint is active at t th step do
15:        δ_t ← δ_t + δ_res/N_active
16:    end for
17: end while
```

Iterative Risk Allocation Algorithm

The Iterative Risk Allocation (IRA) algorithm (Algorithm 1) solves Problem 3 through iterative improvement. It has a parameter (an interpolation coefficient) $0 < \alpha < 1$. The lower stage optimization is solved 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 11, $\Pr(\boldsymbol{x}_t \notin \mathcal{R}_t \mid \bar{\boldsymbol{x}}_t)$ 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.

For each iteration of the algorithm the nominal path is planned using the lower-stage optimization algorithm, given the current risk allocation (Line 4). Risk assignment is decreased when the constraint is inactive (Line 11), and it is increased when the constraint is active (Line 15). Line 13 and 15 ensure that $\sum_{t=1}^{T} \delta_t = \delta$ so that the suboptimality due to the union bound is minimized.

The interpolation coefficient α (Line 11 of Algorithm 1) can be interpreted as a "step size". Just like a fixed step size of descend algorithms, a large α leads to a fast descent but greater suboptimality, as shown in Figure 4. A good strategy is to use a large α (~ 0.7) at the beginning of the algorithm, and gradually discount it over iterations.

Properties

A very important property of the IRA algorithm is summarized by the following theorem.

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

$$J^i \leq J^{(i+1)} \quad (13)$$

where i is the iteration index.

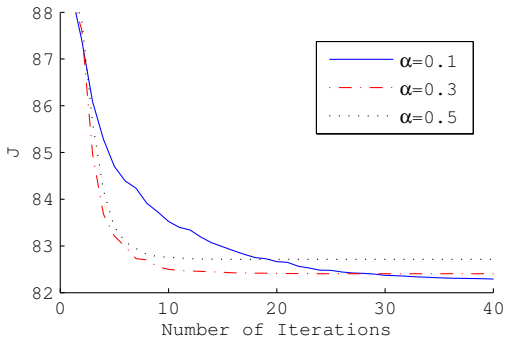


Figure 4: A typical convergence plot of the iterative risk allocation algorithm with different settings of α. Plots are obtained from AUV depth navigation example described in the Simulation section

Proof. If there are inactive constraints, they keep being inactive after Line 11, since $\alpha < 1$. Thus, the objective function J does not change at this point. Then in Line 15, the active constraints are relaxed, so J become equal or larger than in the previous iteration. If there are no inactive constraints, then $\delta_{res} = 0$ and thus the risk assignment of active constraints does not change. Consequently, the objective function does not change as well. □

In a special case where the lower stage optimization is a linear programming, the strict monotonicity $J^i < J^{(i+1)}$ can be obtained when there are both active and inactive constraints.

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 can yield a significantly less conservative result when started from the simple uniform risk allocation $\delta_t = \delta/T$ $(t = 1 \cdots T)$ (Line 1 of Algorithm 1).

The IRA algorithm is sound but not complete. If the initial risk allocation (Line 1 of Algorithm 1)is feasible, then the algorithm is guaranteed to find a feasible risk allocation that is equal or better than the initial risk allocation; however, if the initial risk allocation is infeasible the algorithm fails to find feasible solution. Therefore if the IRA algorithm is initialized with the solution of the ellipsoidal approximation approach, it can always find equal or better feasible solution than the ellipsoidal approximation. In practice, the simple uniform initial risk allocation as Line 1 of Algorithm 1 is less conservative than the ellipsoidal approximation in most cases (this may not hold only when the upper-bound of the risk δ is close to one; however, in practical cases, δ is very close to zero). Therefore, although both the ellipsoidal approximation and IRA is incomplete, IRA can find a feasible solution whenever the ellipsoidal approximation approach can find one. On the other hand, Particle Control algorithm is complete but not sound for a finite number of samples.

IRA algorithm can be complete if the set of feasible risk allocation has non-zero volume (indeed, it is often the case). In such case we can find a feasible initial risk allocation using branch and bound algorithm, but it may requires significant computation time in the worst case.

Linear Time Invariant System with a Gaussian Disturbance

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

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

$$\boldsymbol{x}_{t+1} = \boldsymbol{A}\boldsymbol{x}_t + \boldsymbol{B}\boldsymbol{u}_t + \boldsymbol{w}_t. \tag{14}$$

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

$$\boldsymbol{w} \sim \mathcal{N}(\boldsymbol{0}, \Sigma_w) \tag{15}$$

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

$$\Sigma_{x,t} = \sum_{k=0}^{t-1} \boldsymbol{A}^k \Sigma_w (\boldsymbol{A}^k)^T. \tag{16}$$

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

$$\mathcal{R}_t = \left\{ \boldsymbol{x}_t \in \mathcal{X} \ : \bigwedge_{i=1}^{N_t} \boldsymbol{h}_t^{iT} \boldsymbol{x}_t \le g_t^i \right\}. \tag{17}$$

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

$$\Pr\left[\bigvee_{i=1}^{N_t} \boldsymbol{h}_t^{iT} \boldsymbol{x}_t > g_t^i \right] \le \delta_t. \tag{18}$$

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

$$\bigwedge_{t=1}^{T} \bigwedge_{i=1}^{N_t} \left(\Pr[\boldsymbol{h}_t^{iT} \boldsymbol{x}_t > g_t^i] \le \delta_t^i \right) \ \wedge \ \sum_{t=1}^{T} \sum_{i=1}^{N_t} \delta_t^i \le \delta. \tag{19}$$

The risk allocation problem of δ_t^i is 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.(17). The nominal states are bounded by the tightened constraints such that

$$\boldsymbol{h}_t^{iT} \bar{\boldsymbol{x}}_t \ \le \ g_t^i - m_t^i \tag{20}$$

$$m_t^i \ = \ \sqrt{2 \boldsymbol{h}_t^{iT} \Sigma_{x,t} \boldsymbol{h}_t^i}\, erf^{-1}(1 - 2\delta_t^i), \tag{21}$$

where erf^{-1} is the inverse of the Gauss error function. The conditional probability of failure at time t in the Algorithm 1, Line 11 is replaced by the probability of violating the i th constraint at time t, which is equal to the cumulative distribution function,

$$\Pr(\boldsymbol{x}_t > g_t^i \mid \bar{\boldsymbol{x}}_t) = \frac{1}{2}\left(1 + erf\frac{\boldsymbol{h}_t^{iT}(\boldsymbol{x}_t - \bar{\boldsymbol{x}})}{\sqrt{2\boldsymbol{h}_t^{iT}\Sigma_{x,t}\boldsymbol{h}_t^i}}\right). \quad (22)$$

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 assume 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. The AUV is disturbed by tides and currents. We discretize the linear dynamics 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 Dorado-class AUV that is operated by Monterey Bay Aquarium Research Institute (Figure 1), and the bathymetric data is taken from Monterey Bay. The deterministic planning algorithm used in the lower-stage has been demonstrated in the actual AUV mission.

The AUV has six real-valued state and takes one real-valued 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 ($1 \leq t \leq 20$) with probability $\delta = 0.05$. The objective is to **minimize** the average of AUV's nominal altitude above the floor.

Algorithms tested Three algorithms are implemented in Matlab and run on a machine with a Pentium 4 2.80 GHz processor and 1.00 GB of RAM. The planning horizon is 100 seconds (20 time steps each with a 5 second time intervals). Three algorithms are:
(a) Ellipsoidal relaxation approach (van Hessem 2004),
(b) Bi-stage Robust Motion Planning ($\alpha = 0.3$), and
(c) Particle Control (20 particles) (Blackmore 2006)

Result The three algorithms are run on 50 cases with different segments of the Monterey Bay sea floor. Figure 5 shows a typical result of the three algorithms with $\delta = 0.05$. Ellipsoid relaxation yields a large safety margin, which touches the nominal path (i.e. constraint is active) only at a few points, just as Figure 2-(a). This is because ellipsoid relaxation uniformly allocates risk to each step. On the other hand, Bi-stage Robust Motion Planning algorithm generates a nominal path that touches the safety margin at most points, just as Figure 2-(b). This 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.

Table 1: Performance comparison on the AUV depth planning problem with chance constraint $P_{Fail} \leq 0.05$. Values are the average of 50 runs on different segments of the Monterey Bay sea floor. Planning horizon is 20 steps (100 seconds).
(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}	$< 10^{-5}$	0.023	0.297
Objective function J	88.6	64.1	56.2
Computation time [sec]	0.093	1.36	915.6

Figure 6 shows the optimal risk allocation computed by BRMP algorithm on the same case as Figure 5 with $\delta = 0.05$. A large portion of the risk is allocated to the time steps when AUV go above the sea mountain, just as taking a large risk at the corner in the race car example.

The performance of the three algorithms is compared in Table 1. Values in the table are the average of 50 cases. 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 violation of the chance constraint. This is because Particle Control is a sample based stochastic algorithm that does not have guarantee that the chance constraint is strictly satisfied (i.e. Particle Control is not sound).

The ellipsoidal relaxation approach fail to find a solution in one case out of 50, while BRMP find solutions in all 50 cases. This is because the strong conservatism of the ellipsoidal relaxation(i.e. large safety margin) makes the optimization problem infeasible.

The value of objective function J is a measure of optimality. Note that this problem is a minimization, hence 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. The suboptimality of the BRMP is 14% at most, while the ellipsoidal relaxation has 57% suboptimality at most.

The computation time of Particle Control is longer than the planning horizon (100 sec). Although BRMP is slower than the ellipsoidal relaxation approach, it is approximately one thousand times faster than Particle Control.

Conclusion

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

Acknowledgments

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

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 network traffic control. *Automatica* 41:595–604.

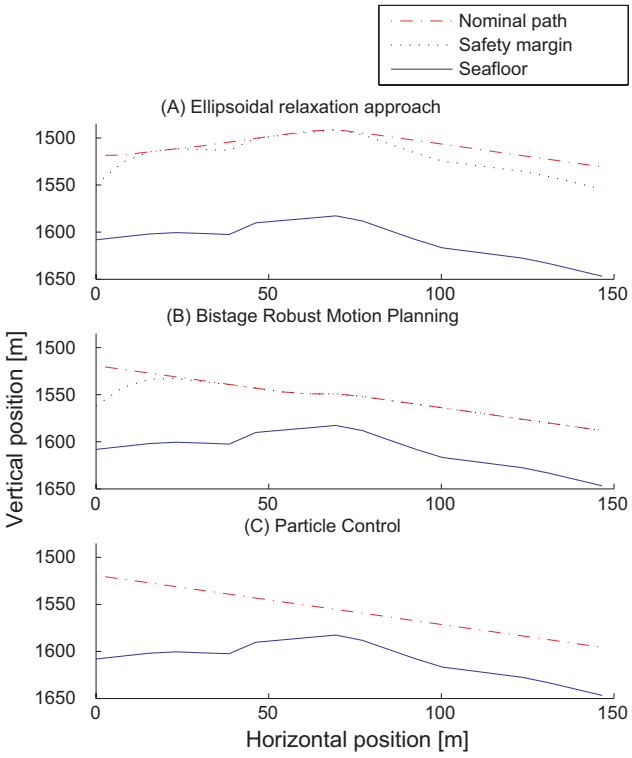


Figure 5: Nominal path of AUV and safety margin planned by three algorithms. Safety margin is not shown in (c) since Particle Control does not explicitly compute it.

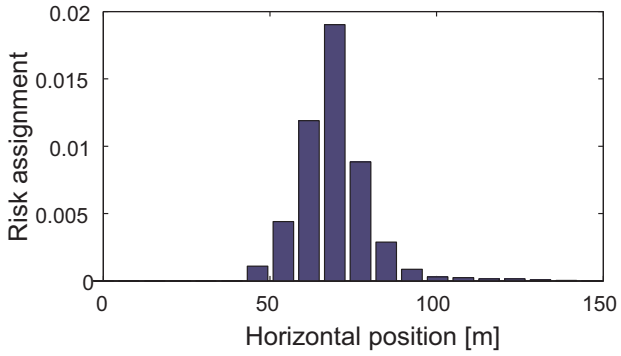


Figure 6: Risk allocation of Figure 5 (b)