Computer Science and Artificial Intelligence Laboratory
Technical Report

MIT-CSAIL-TR-2008-014 March 6, 2008

Two-stage Optimization Approach to Robust Model Predictive Control with a Joint Chance Constraint

Masahiro Ono and Brian C. Williams

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

Two-stage Optimization Approach to Robust Model Predictive Control with a Joint Chance Constraint *

Masahiro Ono †and Brian Williams‡

March 6, 2008

Abstract

This report proposes a new two-stage optimization method for robust Model Predictive Control (RMPC) with Gaussian disturbance and state estimation error. Since the disturbance is unbounded, it is impossible to achieve zero probability of violating constraints. Our goal is to optimize the expected value of a objective function while limiting the probability of violating any constraints over the planning horizon (joint chance constraint). Prior arts include constraint tightening with ellipsoidal relaxation [8] and Particle Control [1], but the former yields very conservative result and the latter is computationally intensive. Our new approach divide the optimization problem into two stages; the upper-stage that optimizes risk allocation, and the lower-stage that optimizes control sequence with tightened constraints. The lower-stage is a regular convex optimization, such as Linear Programming or Quadratic Programming. The upper-stage is also a convex optimization under practical assumption, but the objective function is not always differentiable, and computation of its gradient or subgradient is expensive for large-scale problem. A necessary condition for optimality, which does not explicitly use gradient and hence easy to compute, is discussed. A descent algorithm for the upper-stage called Iterative Risk Allocation (IRA), which does not require the computation of gradient, is proposed. Although the algorithm is not guaranteed to converge to the optimal, empirical results show that it converges quickly to a point that is close to the optimal. Suboptimality is much smaller than ellipsoidal relaxation method while achieving a substantial speedup compared to Particle Control.

1 Introduction and Problem Statement

Model Predictive Control have drawn attention of researchers in a wide range of field from chemical plant control, financial engineering to unmanned aerial vehicle path planning. When it is applied to the real-world robotic system, robustness against uncertainty, such as exogenous disturbance, actuation error, and state estimation error, is an important issue.

A lot of research has been done on robust Model Predictive Control (RMPC) that assumes bounded disturbance [3][4][5][6]. However, in many practical cases, disturbance is often stochastic and unbounded. This paper focuses on RMPC under Gaussian distribution, since it is often a good approximation of those stochastic disturbances. When the disturbance is unbounded, it is impossible to achieve zero probability of violating constraints, or conduct min-max optimization. In such case an effective strategy is to optimize the expected value of a objective function while limiting the probability of violating any constraints over the planning horizon (joint chance constraint). The optimization problem to solve is formally defined as follows.

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

†CSAIL Model-based Embedded and Robotic Systems Group. hiro_ono@mit.edu

‡CSAIL Model-based Embedded and Robotic Systems Group. williams@mit.edu

1.1 Formal Problem Statement

Notations: Following notations are used throughout the paper.

$$
\begin{array}{rcl}
\boldsymbol{x}_n & : & \text{State vector at time } n\text{(Randomvariable).} \\
\boldsymbol{u}_n & : & \text{Control input at time } n. \\
\boldsymbol{w}_n & : & \text{Disturbance at time } n\text{(Random variable).} \\
\bar{\boldsymbol{x}}_n & : & = E[\boldsymbol{x}_n] : \text{Nominal state.} \\
\boldsymbol{X} & := & (\boldsymbol{x}_0^T \ \boldsymbol{x}_1^T \cdots \boldsymbol{x}_N^T)^T \\
\boldsymbol{U} & := & (\boldsymbol{u}_0^T \ \boldsymbol{x}_1^T \cdots \boldsymbol{x}_{N-1}^T)^T \\
\bar{\boldsymbol{X}} & := & (\bar{\boldsymbol{x}}_0^T \ \bar{\boldsymbol{x}}_1^T \cdots \bar{\boldsymbol{x}}_N^T)^T
\end{array}
$$

Problem 1: RMPC with a joint chance constraint

$$
\begin{array}{lll}
\min\limits_{\boldsymbol{U}} & E[J(\boldsymbol{X}, \boldsymbol{U})] & (1) \\
\text{s.t.} & \boldsymbol{x}_{n+1} = A\boldsymbol{x}_n + B\boldsymbol{u}_n + \boldsymbol{w}_n & (2) \\
 & \boldsymbol{u}_{min} \le \boldsymbol{u}_n \le \boldsymbol{u}_{max} & (3) \\
 & \boldsymbol{w}_n \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_w) & (4) \\
 & \boldsymbol{x}_0 \sim \mathcal{N}(\bar{\boldsymbol{x}}_0, \boldsymbol{\Sigma}_{x,0}) & (5) \\
 & \Pr\left[\bigwedge_{n=0}^{N} \bigwedge_{i=1}^{I_n} \boldsymbol{h}^{iT}\boldsymbol{x}_n \le g_n^i\right] \ge 1 - \Delta & (6)
\end{array}
$$

We assume a discrete-time linear time invariant (LTI) system with disturbance over a time horizon N. Exogenous disturbance and actuation error are represented by $\boldsymbol{w}$, and state estimation error is represented by $\boldsymbol{x}_0$. Both random variables have Gaussian distribution with variance Σ_w and $\Sigma_{x,0}$, respectively. We refer (6) to as *joint* chance constraint, since it limits the probability of violating *any* of I_n constraints *throughout the time horizon* $0 < n < N$, instead of limiting the probability of violating *a individual* constraint in *a single time step* (individual chance constraints, introduced in (7)). The upper bound of the probability of failure over the time horizon is denoted by Δ, which is a free parameter defined by system operators. When Problem 1 is used as a open-loop planning problem, Δ means the probability of failure over a mission (from start to goal).

For example, in a path planning problem of an autonomous underwater vehicle (AUV) that conducts bathymetric mapping mission, an AUV operator wants to minimize the average altitude from the sea floor to obtain as high resolution data as possible with an upper-bound Δ of the probability of colliding with the sea floor throughout the mission.

1.2 Related Works and Approach

As far as the authors know, there are two algorithms that can solve RMPC with a *joint* chance constraint. One of them uses ellipsoidal relaxation technique [8] to derive the necessary condition of (6) and reduce the stochastic MPC to deterministic problem by constraint tightening. Although it is computationally efficient, ellipsoidal relaxation is a very conservative bound, and hence the result has significant suboptimality.

The other one is a sampling-based algorithm called Particle Control [1]. It can directly optimize the control sequence without using a conservative bound such as ellipsoidal relaxation. Its another advantage over [8] is that it can handle non-Gaussian distributions. However, it is slow when it is applied to a large-scale problem 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.

On the other hand, RMPC with *individual* chance constraints can be solved efficiently by constraint tightening[8][9]. Our new approach decomposes a joint chance constraint into individual chance constraints by a novel idea called *risk allocation*. The resulting algorithm consists of two stages, with its upper-stage optimizing risk allocation while the lower-stage corresponding to RMPC with *individual* chance constraints. The suboptimality of the proposed algorithm is much less than ellipsoidal relaxation method while computation is significantly faster than Particle Control.

The rest of paper is outlined as follows. We first briefly review RMPC with individual chance constraints and its solution, followed by an introduction of the new two-stage optimization approach . Convexity of the upper-stage optimization is proved and the necessary condition for optimality is discussed in Section 4. In the following section a fast descent algorithm for upper-stage optimization called Iterative Risk Allocation (IRA) is introduced.

2 Review of RMPC with individual chance constraints

RMPC with *individual* chance constraints is stated as follows.
Problem 2: RMPC with individual chance constraints

$$\begin{aligned}
\min_{\boldsymbol{U}} \quad & E[J(\boldsymbol{X},\boldsymbol{U})] \\
\text{s.t.} \quad & \boldsymbol{x}_{n+1} = A\boldsymbol{x}_n + B\boldsymbol{u}_n + \boldsymbol{w}_n \\
& \boldsymbol{u}_{min} \le \boldsymbol{u}_n \le \boldsymbol{u}_{max} \\
& \boldsymbol{w}_n \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_w) \\
& \boldsymbol{x}_0 \sim \mathcal{N}(\bar{\boldsymbol{x}}_0, \boldsymbol{\Sigma}_{x,0}) \\
& \Pr\left[\boldsymbol{h}_n^{iT}\boldsymbol{x}_n \le g_n^i\right] \ge 1-\delta_n^i \qquad (7)\\
& \quad (n = 0\cdots N, i = 0\cdots I_n)
\end{aligned}$$

This stochastic problem is turned into a deterministic problem. First, the distribution of $\boldsymbol{x}_n$ (hence, distribution of $\boldsymbol{X}$) is computed as follows using the distribution of $\boldsymbol{w}$ and $\boldsymbol{x}_0$.

$$\Sigma_{x,n} = \sum_{k=0}^{n-1} \boldsymbol{A}^k \Sigma_w (\boldsymbol{A}^k)^T + \Sigma_{x,0}. \tag{8}$$

Since the distribution of $\boldsymbol{X}$ is known, expectation of the objective function can be described as a function of nominal state, which is a deterministic variable.

$$E[J(\boldsymbol{X},\boldsymbol{U})] = \bar{J}(\bar{\boldsymbol{X}},\boldsymbol{U}) \tag{9}$$

When J is linear, $\bar{J}(\bar{\boldsymbol{X}},\boldsymbol{U}) = J(\bar{\boldsymbol{X}},\boldsymbol{U})$; see a reference [9] for the case of quadratic objective function.

The individual chance constraints (7) is turned into deterministic constraints on the nominal state using constraint tightening[8][9]. Thus Problem 2 is equivalent to the following deterministic MPC problem. When $\bar{J}$ is linear, it is Linear Programming.
Problem 3: Deterministic MPC on nominal states (Lower-stage)

$$\begin{aligned}
\min_{\boldsymbol{U}} \quad & \bar{J}(\bar{\boldsymbol{X}},\boldsymbol{U}) && (10)\\
\text{s.t.} \quad & \bar{\boldsymbol{x}}_{n+1} = A\bar{\boldsymbol{x}}_n + B\boldsymbol{u}_n && (11)\\
& \boldsymbol{u}_{min} \le \boldsymbol{u}_n \le \boldsymbol{u}_{max} && (12)\\
& \boldsymbol{h}_n^{iT}\bar{\boldsymbol{x}}_n \le g_n^i - m_n^i(\delta_n^i) && (13)
\end{aligned}$$

where $m_n^i(\cdot)$ is the inverse of cumulative distribution function of one-dimensional Gaussian distribution with variance $\boldsymbol{h}^{iT}\Sigma_{x,n}\boldsymbol{h}^i$. It is convex when $\delta_n^i \in [0\ 0.5]$.

$$m_n^i(\delta_n^i) = \sqrt{2\boldsymbol{h}_n^{iT}\Sigma_{x,n}\boldsymbol{h}_n^i}\ erf^{-1}(1-2\delta_n^i) \tag{14}$$
$$(n = 0\cdots N, i = 0\cdots I).$$

where erf^{-1} is the inverse of the Gauss error function. The interpretation is that the algorithm set a safety margin m_n^i to keep the nominal state away from infeasible region, so that the probability of violating ith constraint at time n is less than the upper bound δ_n^i.

3 Two-stage Optimization Approach

3.1 Risk Allocation

Observe that, using union bound or Boole's inequality ($Pr[A \cup B] \leq \Pr[A] + \Pr[B]$), a set of individual chance constraints (7) together with the following additional constraint implies the original joint chance constraint (6) [2].

$$\sum_{n=0}^{N}\sum_{i=1}^{I_n}\delta_n^i \leq \Delta \tag{15}$$

In other words, once the upper bounds of the probability of violating individual constraints δ_n^i are fixed according to (15), then RMPC with joint chance constraint (Problem 1) is turned into RMPC with individual chance constraints (Problem 2/3). For later convenience, a vector $\boldsymbol{\delta}$ is defined as follows;

$$\boldsymbol{\delta} = (\delta_0^1\ \delta_0^2\ \cdots\delta_N^{I-1}\ \delta_N^I)^T. \tag{16}$$

It can be viewed as a resource allocation problem; the total amount of resource Δ is upper-bounded, and the problem is to find the optimal resource allocation $\boldsymbol{\delta}$ to maximize the utility. Thus we call $\boldsymbol{\delta}$ as "*risk allocation*". The good risk allocation strategy is to save risk when the gain of taking risk is small, while taking risk when a great gain is expected by doing so. For example, when a racing car driver wants to reach the goal as quick as possible, he would take greater risk at corner by running inside track (i.e. greater probability of getting out of course) since it leads to time saving, while he would save risk by running middle of the course at straight line since running on the edge of the course does not leads to time saving.

3.2 Two-stage Optimization

By optimizing risk allocation $\boldsymbol{\delta}$ as well as control sequence $\boldsymbol{U}$ in Problem 3 with a constraint (15), the original Problem 1 is optimized. (Strictly speaking, it is suboptimal in general since the union bound is a conservative bound; however, the suboptimality is small when $\Delta << 1$.) However, since the constraints (13) are nonlinear in terms of $\boldsymbol{\delta}$, the resulting optimization problem is hard to solve.

Our approach is to separate the optimization of risk allocation $\boldsymbol{\delta}$ from Problem 3 to keep the linearity in constraints, so that efficient Linear Programming or Quadratic Programming solver can be used. The resulting algorithm is a two-stage optimization; upper-stage optimizes risk allocation $\boldsymbol{\delta}$ while lower-stage (Problem 3) optimizes $\boldsymbol{U}$. The upper-stage optimization problem is formally stated as follows.

Problem 4: Risk Allocation Optimization (Upper-stage)

$$\min_{\boldsymbol{\delta}} \quad \bar{J}^\star(\boldsymbol{\delta}) \tag{17}$$
$$\text{s.t.} \quad \sum_{n=0}^{N}\sum_{i=1}^{I}\delta_n^i \leq \Delta \tag{18}$$
$$\delta_n^i \geq 0 \tag{19}$$
$$\boldsymbol{\delta} \in \{\boldsymbol{\delta} \mid \text{Problem 3 is feasible}\} \tag{20}$$

where $\bar{J}^\star(\boldsymbol{\delta})$ is the optimum objective function of Problem 3 given $\boldsymbol{\delta}$.

$$\bar{J}^\star(\boldsymbol{\delta}) = \min_{\boldsymbol{U}} \bar{J}(\bar{\boldsymbol{X}}, \boldsymbol{U}) \text{ s.t. } (11) - (13) \tag{21}$$

By solving Problem 4 (**upper-stage**) together with Problem 3 (**lower-stage**), the original RMPC with joint chance constraint (Problem 1) is solved. The next question is, of course, how to solve Problem 4. To answer the question, the next section discuss its convexity as well as the necessary condition for optimality.

4 Convexity and Necessary Condition for Optimality

4.1 Convexity

Theorem 1: Problem 4 is a convex optimization if the objective function of Problem 3 $\bar{J}(\boldsymbol{X}, \boldsymbol{U})$ is convex and $\Delta \le 0.5$.

Proof Let $\boldsymbol{\delta}^1$ and $\boldsymbol{\delta}^2$ be feasible risk allocations that satisfy (18) - (20). Let $(\boldsymbol{X}_1^\star, \boldsymbol{U}_1^\star)$ and $(\boldsymbol{X}_2^\star, \boldsymbol{U}_2^\star)$ be the optimum solution of Problem 3 for $\boldsymbol{\delta}^1$ and $\boldsymbol{\delta}^2$ respectively. We will first show that, for $0 \le k \le 1$, $(k\boldsymbol{X}_1^\star + (1-k)\boldsymbol{X}_2^\star,\, k\boldsymbol{U}_1^\star + (1-k)\boldsymbol{U}_2^\star)$ is a feasible solution.

It satisfies (11) and (12) for linearity. It also satisfies (13), since

$$\begin{aligned} &\boldsymbol{h}^{iT}\{k\bar{\boldsymbol{x}}_{n,1} + (1-k)\bar{\boldsymbol{x}}_{n,2}\} \\ &\qquad \le g_n^i - k m_n^i(\delta_{n,1}^i) - (1-k) m_n^i(\delta_{n,2}^i) \\ &\qquad \le g_n^i - k m_n^i(k\delta_{n,1}^i + (1-k)\delta_{n,2}^i) \end{aligned} \tag{22}$$

The second inequality uses the fact that $m_n^i(\delta_n^i)$ is convex when $0 \le \delta_n^i \le \Delta \le 0.5$. The convexity of m can be immediately proved from the fact that $\frac{df}{dx} \le 0$ for $x \le 0$, where $f(\cdot)$ is the probability distribution function of Gaussian. Therefore, $(k\boldsymbol{X}_1^\star + (1-k)\boldsymbol{X}_2^\star, k\boldsymbol{U}_1^\star + (1-k)\boldsymbol{U}_2^\star)$ is a feasible solution of Problem 3 for $k\boldsymbol{\delta}_1 + (1-k)\boldsymbol{\delta}_2$. Thus, the feasible region of Problem 4 is convex.

Next we will show the convexity of the objective function of Problem 4. Since Problem 3 is feasible for $k\boldsymbol{\delta}_1 + (1-k)\boldsymbol{\delta}_2$, there is an optimum solution $(\boldsymbol{X}_k^\star, \boldsymbol{U}_k^\star)$ that gives less or equal value of objective function than $(k\boldsymbol{X}_1^\star + (1-k)\boldsymbol{X}_2^\star, k\boldsymbol{U}_1^\star + (1-k)\boldsymbol{U}_2^\star)$. Note that $\bar{J}(\boldsymbol{X}_k^\star, \boldsymbol{U}_k^\star)$ is equivalent to the objective function of Problem 4 $\bar{J}^\star(k\boldsymbol{\delta}_1 + (1-k)\boldsymbol{\delta}_2)$. Using convexity of $\bar{J}$,

$$\begin{aligned} &\bar{J}^\star(k\boldsymbol{\delta}_1 + (1-k)\boldsymbol{\delta}_2) \\ &\qquad = \bar{J}(\boldsymbol{X}_k^\star, \boldsymbol{U}_k^\star) \\ &\qquad \le \bar{J}\left(k\boldsymbol{X}_1^\star + (1-k)\boldsymbol{X}_2^\star, k\boldsymbol{U}_1^\star + (1-k)\boldsymbol{U}_2^\star\right) \\ &\qquad \le k\bar{J}(\boldsymbol{X}_1^\star, \boldsymbol{U}_1^\star) + (1-k)\bar{J}(\boldsymbol{X}_2^\star, \boldsymbol{U}_2^\star) \\ &\qquad = k\bar{J}^\star(\boldsymbol{\delta}_1) + (1-k)\bar{J}^\star((\boldsymbol{\delta}_2) \end{aligned} \tag{23}$$

Thus the objective function of Problem 4 $\bar{J}^\star(\boldsymbol{\delta})$ is convex.

4.2 Gradient

Computation of the gradient is of great interest for optimization. Although it is not a main issue of this paper, we will briefly describe how to obtain the gradient.

The objective function of Problem 4 $\bar{J}^\star(\boldsymbol{\delta})$ is not always differentiable. When it is differentiable,

$$\frac{\partial \bar{J}^\star}{\partial \delta_n^i} = \frac{\partial \bar{J}^\star}{\partial m_n^i}\frac{dm_n^i}{d\delta_n^i} \tag{24}$$

Since $m_n^i(\delta_n^i)$ is the inverse function of the cumulative distribution function of Gaussian,

$$\frac{dm_n^i}{d\delta_n^i} = \frac{1}{f(m_n^i(\delta_n^i))} \tag{25}$$

where $f(\cdot)$ is the probability distribution function of Gaussian.

On the other hand, $\frac{\partial \bar{J}^\star}{\partial m_n^i}$ is harder to obtain. It may be obtained as a resulting Lagrange multiplier in some cases; when Problem 3 is Linear Programming, it can be obtained by matrix operation.

We assume that $\bar{J}$ is linear and hence Problem 3 is Linear Programming. Problem 3 can be described in a simple form as follows by eliminating $\bar{\boldsymbol{X}}$ using (11);

$$\min_{\boldsymbol{U}} \quad \boldsymbol{f}^T \boldsymbol{U} \tag{26}$$

$$\text{s.t.} \quad H\boldsymbol{U} \leq \boldsymbol{g} - \boldsymbol{m}(\boldsymbol{\delta}) \tag{27}$$

Let $\boldsymbol{U}^\star$ be the optimized decision vector, and $\bar{J}^\star = \boldsymbol{f}^T \boldsymbol{U}^\star$ be the optimized objective function. $\bar{J}^\star(\boldsymbol{\delta})$ is differentiable iff there are exactly n_U active independent constraints in (27), where n_U is the number of dimensions of $\boldsymbol{U}$. We divide (27) into active and inactive constraints.

$$\text{Active} \quad : \quad H_A \boldsymbol{U}^\star = \boldsymbol{g}_A - \boldsymbol{m}_A(\boldsymbol{\delta}_A) \tag{28}$$

$$\text{Inactive} \quad : \quad H_I \boldsymbol{U}^\star < \boldsymbol{g}_I - \boldsymbol{m}_I(\boldsymbol{\delta}_I) \tag{29}$$

Note that H_A is a n_U by n_U full rank matrix when $\bar{J}^\star(\boldsymbol{\delta})$ is differentiable. In such case $\bar{J}^\star(\boldsymbol{\delta})$ is differentiated by $\boldsymbol{m}$ as follows;

$$\frac{\partial \bar{J}^\star}{\partial \boldsymbol{m}_A} = -f^T H_A^{-1} \tag{30}$$

$$\frac{\partial \bar{J}^\star}{\partial \boldsymbol{m}_I} = \mathbf{0}. \tag{31}$$

Additional note is that, in the case of $\bar{J}$ being linear, the Hessian is zero when it is differentiable.

4.3 Subgradient

When there is more than n_U active independent constraints, $\bar{J}^\star(\boldsymbol{\delta})$ is differentiable. Let n_H be the number of active constraints. Let H'_A be a n_U by n_U matrix constructed from H_A by removing $(n_H - n_U)$ rows from H_A. There are $\binom{n_H}{n_U}$ ways to construct H'_A. Let $\boldsymbol{v}$ be a "gradient" vector obtained from (30) using H'_A. There are $\binom{n_H}{n_U}$ ways as well to construct $\boldsymbol{v}$. Let $\mathcal{V}$ be the set of all $\binom{n_H}{n_U}$ "gradient" vectors $\boldsymbol{v}$. The subgradient $\partial \bar{J}^\star$ is the convex hull of $\mathcal{V}$.

In most cases $\bar{J}^\star(\boldsymbol{\delta})$ is indifferentiable at the optimal point. Thus, in general, the necessary and sufficient condition for optimality is $\mathbf{0} \in \partial \bar{J}^\star$.

This computation is generally expensive since it requires $\binom{n_H}{n_U}$ n_U by n_U matrix inversions.

4.4 Necessary Condition for optimality

When the dimension of decision vector is large, the computation of gradient and subgradient become expensive due to the matrix inversion. We found a necessary condition for optimality that does not explicitly use gradient.

Theorem 2: Necessary condition for optimality
($\boldsymbol{\delta}$ optimizes Problem 4) $\Rightarrow$ [(Constraints (13) are active for all $n = 0 \cdots N$ and $i = 1 \cdots I_n$ with $\boldsymbol{\delta}$) $\vee$ (They are inactive for all $n = 0 \cdots N$ and $i = 1 \cdots I_n$ with $\boldsymbol{\delta}$)]

In other words, Problem 4 cannot be optimum if only some of constraints in (13) are active.

Proof Assume that there are both active and inactive constraints in (13) for a risk assignment $\boldsymbol{\delta}$. Let $n = n_I, i = i_I$ be a time and an index of constraint where (13) is inactive. Its risk allocation $\delta_{n_I}^{i_I}$ can be decreased (i.e. the constraint at (n, i) is tightened) by a finite value to $\hat{\delta}_{n_I}^{i_I}$ without making any change to the objective function $J(\boldsymbol{X}, \boldsymbol{U})$ since the constraint is inactive. Then one can find an active constraint (n_A, i_A) where the objective function $J(\boldsymbol{X}, \boldsymbol{U})$ is decreased by increasing its risk allocation $\delta_{n_A}^{i_A}$ to $\hat{\delta}_{n_A}^{i_A}$ so that $(\delta_{n_A}^{i_A} + \delta_{n_I}^{i_I} - \hat{\delta}_{n_I}^{i_I})$ (i.e. the constraint at (n, i) is loosened). The new risk allocation $\hat{\boldsymbol{\delta}}$ satisfies (15). Thus a risk allocation including $\delta_{n_A}^{i_A}$ and $\delta_{n_I}^{i_I}$ is not optimal.

Note that the proof of Theorem 2 also provides a descent direction.

Lemma 1: Descent direction If there are both active and inactive constraints, $(\hat{\boldsymbol{\delta}} - \boldsymbol{\delta})$ in the proof of Theorem 2 above is a descent direction.

Another important note is that, although having all constraints active is *not* a sufficient condition for optimality, having all constraints *inactive* is a sufficient condition for optimality.

Lemma 2: Sufficient condition for optimality
(Constraints (13) are inactive for all $n = 0 \cdots N$ and $i = 1 \cdots I$ with $\boldsymbol{\delta}$)$\Rightarrow$ ($\boldsymbol{\delta}$ optimizes Problem 4)

Proof Assume that all constraints in (13) are inactive. One cannot decrease the objective function $J(\boldsymbol{X}, \boldsymbol{U})$ by changing any of δ_n^i, since all constraints are inactive. Thus it is optimal.

5 Iterative Risk Allocation Algorithm

From Theorem 2, Lemma 1 and Lemma 2, a simple yet very powerful descent algorithm called Iterative Risk Allocation (IRA) is derived. It is computationally efficient since it does not explicitly use gradient nor sub-gradient. Although it is not guaranteed to converge to the optimum, the suboptimality is small in practice. Given a feasible initial risk assignment, it generate the sequence of feasible risk assignment that monotonically decreases the objective function.

Algorithm 1 describes the IRA algorithm. It has a parameter $0 < \alpha < 1$, which corresponds to a step size. The algorithm can be started from any feasible risk assignment $\boldsymbol{\delta}$ that satisfies $\sum_{n=0}^{N} \sum_{i=1}^{I} \delta_n^i = \Delta$, but in most cases uniform risk assignment works well (Line 1).

In Line (10), $cdf(g_n^i - \boldsymbol{h}_n^{iT} \bar{\boldsymbol{x}}_n^\star)$ is the probability of violating a constraint at (n, i), where $cdf(\cdot)$ is the cumulative distribution function of zero-mean Gaussian distribution with variance $\boldsymbol{h}_n^{iT} \Sigma_{x,n} \boldsymbol{h}_n^i$, and $\bar{\boldsymbol{x}}_n^\star$ is the optimized nominal state for $\boldsymbol{\delta}$. A constraint is active at (n, i) when $\delta_n^i = cdf(g_n^i - \boldsymbol{h}_n^{iT} \bar{\boldsymbol{x}}_n^\star)$ and inactive otherwise. In practice, we judge that a constraint is active if

$$|\delta_n^i - cdf(g_n^i - \boldsymbol{h}_n^{iT} \bar{\boldsymbol{x}}_n^\star)| < \epsilon \tag{32}$$

where ϵ is a small positive real number.

The algorithm uses Lemma 2 to obtain descent direction. It reduce the risk allocation where the constraint is inactive (Line 10) and increase it where the constraint is active (Line 14). Line 12 and 14 ensure that $\sum_{n=0}^{N} \sum_{i=1}^{I_n} \delta_t^i = \Delta$ so that the suboptimality due to the union bound is minimized.

It can be easily seen from the proof of Theorem 2 that the objective function monotonically decreases over the iteration of Algorithm 1. It is also clear from the proof of Theorem 2 that Algorithm 1 converges only to a risk allocation that makes all constraint active or inactive. Line 5 stops the algorithm in those cases.

The performance of IRA algorithm is demonstrated by simulation in [7].

References

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

[2] Lars Blackmore and Brian C. Williams. Optimal manipulator path planning with obstacles using disjunctive programming. In *Proceedings of the American Control Conference*, 2006.

[3] Eric C. Kerrigan. *Robust Constraint Satisfaction: Invariant Sets and Predictive Control.* PhD thesis, University of Cambridge, 2000.

[4] Mayuresh V. Kothare, Venkataramanan Balakrishnan, and Manfred Morari. Robust constrained model predictive control using linear matrix inequalities. *Automatica*, 32(10):1361–1379, October 1996.

Algorithm 1 Iterative Risk Allocation

$\forall(n,i) \quad \delta_n^i \leftarrow \Delta/(N_{active} = \sum_{n=0}^{N} I_n)$
loop
  Solve Problem 3 with $\boldsymbol{\delta}$.
  $N_{active} \leftarrow$ number of steps where constraint is active
  if $N_{active} = 0$ or $N_{active} = \sum_{n=0}^{N} I_n$ **then**
    break;
  end if
  for all (n,i) such that ith constraint at nth time step is inactive **do**
    $\delta_n^i \leftarrow \alpha\delta_n^i + (1-\alpha) cdf_n^i(g_n^i - \boldsymbol{h}_n^{iT}\bar{\boldsymbol{x}}_n^{\star})$
  end for
  $\delta_{res} \leftarrow \Delta - \sum_{n=0}^{N}\sum_{i=1}^{I_n} \delta_n^i$
  for all (n,i) such that ith constraint at nth time step is inactive **do**
    $\delta_n^i \leftarrow \delta_n^i + \delta_{res}/N_{active}$
  end for
end loop

[5] Yoshiaki Kuwata, Arthur Richards, and Jonathan How. Robust receding horizon control using generalized constraint tightening. *Proceedings of American Control Conference*, 2007.

[6] Johan Löfberg. *Minimax Approaches to Robust Model Predictive Control*. PhD thesis, Linköping Studies in Science and Technology, 2003.

[7] Masahiro Ono and Brian C. Williams. Efficient motion planning algorithm for stochastic dynamic systems with constraints on probability of failure. Technical report, MIT Computer Science and Artificial Intelligence Laboratory, 2008.

[8] Dennis Harald van Hessem. *Stochastic inequality constrained closed-loop model predictive control with application to chemical process operation*. PhD thesis, Delft University of Technology, 2004.

[9] Jun Yan and Robert R. Bitmead. Incorporating state estimation into model predictive control and its application to networktraf fic control. *Automatica*, 2005.