Proceedings of the
47th IEEE Conference on Decision and Control
Cancun, Mexico, Dec. 9-11, 2008

Iterative Risk Allocation: A New Approach to Robust Model Predictive Control with a Joint Chance Constraint

Masahiro Ono and Brian C. Williams

Abstract— **This paper proposes a novel 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 an objective function while limiting the probability of violating any constraints over the planning horizon (joint chance constraint). Prior arts include ellipsoidal relaxation approach [1] and Particle Control [2], but the former yields very conservative result and the latter is computationally intensive. Our 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 convex, but the objective function is not always differentiable. We developed a fast descent algorithm for the upper-stage called Iterative Risk Allocation (IRA), which yield much smaller suboptimality than ellipsoidal relaxation method while achieving a substantial speedup compared to and Particle Control.**

I. Introduction and Problem Statement

Model Predictive Control has drawn the attention of researchers in a wide range of fields from chemical plant control and financial engineering to unmanned aerial vehicle path planning. Robustness against uncertainty is an important issue when it is applied to real-world robotic systems, which are subject to exogenous disturbance, actuation error, and state estimation error.

There is a considerable body of work on robust Model Predictive Control (RMPC), which assumes a bounded disturbance [3][4][5][6]. However, in many practical cases, disturbance is often stochastic and unbounded. This paper focuses on RMPC under Gaussian distributed disturbance, which is a good approximation of stochastic disturbances for many problems.

When the disturbance is unbounded, it is impossible to guarantee that state constraints are satisfied since there is always a finite probability of having a disturbance that is large enough to push the state out of the feasible region. Therefore, RMPC with unbounded disturbance constrains the *probability* of violating constraints. This constraint is called as *chance constraint*.

RMPC problem usually has multiple state constraints; if it has N constraints in a single time step and T time steps in its planning horizon, there are NT constraints in a problem. Given multiple state constraints, there are two kinds of chance constraints; *individual* chance constraint and *joint* chance constraint[7]. Individual chance constraints limit the probability of violating each single constraint, while joint chance constraint limits the probability of violating any constraints in a problem. For example, individual chance constraints in a racing car path planning problem mean the limit of the probability of having the race car crash into a wall in each time instant. On the other hand, a joint chance constraint means the limit of the probability of crash from start to goal. Although individual chance constraints are easy to solve, a joint chance constraint is more natural and intuitive for the system operator.

This paper sets its focus on RMPC with Gaussian disturbances and a *joint* chance constraint, which is formally described in the following subsection.

A. Formal Problem Statement

Notations: The Following notations are used throughout this paper.

$\boldsymbol{x}_k$: State vector at time k(A random variable).
$\boldsymbol{u}_k$: Control input at time k.
$\boldsymbol{w}_k$: Disturbance at time k(A random variable).
$\bar{\boldsymbol{x}}_k := E[\boldsymbol{x}_k]$: Nominal state at time k.

$$\boldsymbol{X} := \begin{bmatrix} \boldsymbol{x}_0 \\ \vdots \\ \boldsymbol{x}_T \end{bmatrix} \quad \boldsymbol{U} := \begin{bmatrix} \boldsymbol{u}_0 \\ \vdots \\ \boldsymbol{u}_{T-1} \end{bmatrix} \quad \bar{\boldsymbol{X}} := \begin{bmatrix} \bar{\boldsymbol{x}}_0 \\ \vdots \\ \bar{\boldsymbol{x}}_T \end{bmatrix}$$

Problem 1: RMPC with a joint chance constraint

$$\min_{\boldsymbol{U}} \quad E[J(\boldsymbol{X}, \boldsymbol{U})] \tag{1}$$

$$\text{s.t.} \quad \boldsymbol{x}_{k+1} = A\boldsymbol{x}_k + B\boldsymbol{u}_k + \boldsymbol{w}_k \tag{2}$$

$$\boldsymbol{u}_{\min} \leq \boldsymbol{u}_k \leq \boldsymbol{u}_{\max} \tag{3}$$

$$\boldsymbol{w}_k \sim \mathcal{N}(\boldsymbol{0}, \Sigma_w) \tag{4}$$

$$\boldsymbol{x}_0 \sim \mathcal{N}(\bar{\boldsymbol{x}}_0, \Sigma_{x,0}) \tag{5}$$

$$\Pr\left[\bigwedge_{k=0}^{T}\bigwedge_{i=1}^{N} \boldsymbol{h}_k^{iT}\boldsymbol{x}_k \leq g_k^i\right] \geq 1-\Delta \tag{6}$$

We assume a discrete-time linear time invariant (LTI) system with disturbance over a time horizon T. Exogenous disturbance and actuation error are represented by $\boldsymbol{w}$, and state estimation error is represented by $\boldsymbol{x}_0$. Both random variables have a Gaussian distribution with variance Σ_w and $\Sigma_{x,0}$, respectively. The joint chance constraint is described as (6), where Δ is the upper bound of the probability of violating any of N constraints during the planning horizon $0 \leq k \leq T$.

This research is funded by The Boeing Company grant MIT-BA- GTA-1
Masahiro Ono is a PhD student, MIT. hiro_ono@mit.edu
Brian C. Williams is a professor, MIT. williams@mit.edu

B. Related Works and Proposed Approach

Problem 1 is hard to solve since the computation of the left hand side of (6) involves the multidimensional Gaussian integral. There are two algorithms that are previously proposed for solving RMPC with a *joint* chance constraint (Problem 1). One turns a stochastic RMPC problem to a deterministic problem using a very conservative ellipsoidal relaxation [1]. Although this algorithm is computationally efficient, its result is highly suboptimal since the ellipsoidal relaxation produces a very conservative bound. The other is a sampling-based method called Particle Control [2]. It can directly optimize the control sequence without using a conservative bound, such as ellipsoidal relaxation. However, it is slow since the dimension of the decision vector grows proportionally to the number of samples. Another important issue with Particle Control is that, although there is a guarantee that it converges 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 [1][8]. Blackmore et. al. proposed an elegant method in [9] where they decomposed a *joint* chance constraint into *individual* chance constraints by using Boole's inequality, so that Problem 1 can be solved in the same manner as RMPC with *individual* chance constraints. However, the method has non-negligible conservatism since it fixes each individual risk bound to an uniform value. Our new approach exploits this point to achieve further optimality by using a novel concept called *risk allocation* [10]; we decompose a joint chance constraint efficiently by flexibly allocate risk bounds to individual chance constraints. The resulting algorithm consists of two stages, with its upper-stage optimizing risk allocation, while the lower-stage solving RMPC with *individual* chance constraints. The upper-stage optimization problem is convex but not always differentiable. A standard optimization algorithm for such a problem is the subgradient method, but the convergence is slow. To solve the upper-stage optimization problem efficiently, we developed a descent algorithm called Iterative Risk Allocation (IRA). IRA algorithm finds the descent direction by exploiting the problem structure, instead of computing subgradient. It converges very quickly compared to Particle Control [2], and the resulting suboptimality is much smaller than ellipsoidal approximation approach [1].

The concept of the risk allocation and IRA algorithm is initially developed on discrete/continuous hybrid domain and presented in the authors' previous work [10], but it lacked the discussion about the optimality. This paper first proves the convexity of the upper-stage optimization by limiting the focus on linear systems with Gaussian distribution. Then the suboptimality and the convergence speed of IRA algorithm is discussed by comparing with the subgradient algorithm, which is proved to converge to the global optimum when the optimization problem is convex.

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 two-stage optimization approach and a proof of convexity of the upper-stage optimization problem. Section IV and Section V describe two algorithms for the upper-stage; the subgradient method and the newly developed Iterative Risk Allocation algorithm. The performance of these algorithms are demonstrated and compared on simulations in Section VI.

II. Review of RMPC with individual chance constraints

RMPC with *individual* chance constraints is stated as follows.

Problem 2: RMPC with individual chance constraints

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

Problem 2 can be solved efficiently by turning the stochastic problem into a deterministic one.

First, the variance of $\boldsymbol{x}_k$ is computed as follows, using the variance of $\boldsymbol{w}$ and $\boldsymbol{x}_0$.

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

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

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

Although the derivation of $\bar{J}$ is not always trivial, it is rather simple for some specific forms of J that are widely used in practical cases. When J is linear in $\boldsymbol{X}$, $\bar{J}(\cdot) = J(\cdot)$; see [8] for the case of a quadratic objective function; when J is only the function of $\boldsymbol{U}$ (independent of $\boldsymbol{X}$), $\bar{J}(\cdot) = J(\cdot)$.

Finally, the individual chance constraints (7) are turned into deterministic constraints on the nominal state using constraint tightening[1][8] as (13). Now Problem 2 is equivalent to the following deterministic MPC problem (Problem 3).

Problem 3: Deterministic MPC on nominal states (Lower-stage)

$$\min_{\boldsymbol{U}} \quad \bar{J}(\bar{\boldsymbol{X}},\boldsymbol{U}) \tag{10}$$

$$\text{s.t.} \quad \bar{\boldsymbol{x}}_{k+1} = A\bar{\boldsymbol{x}}_k + B\boldsymbol{u}_k \tag{11}$$

$$\boldsymbol{u}_{\min} \le \boldsymbol{u}_k \le \boldsymbol{u}_{\max} \tag{12}$$

$$\boldsymbol{h}_k^{iT}\bar{\boldsymbol{x}}_k \le g_k^i - m_k^i(\delta_k^i) \tag{13}$$

where $-m_k^i(\cdot)$ is the inverse of cumulative distribution function of one-dimensional Gaussian distribution with variance $\boldsymbol{h}_k^{iT}\Sigma_{x,k}\boldsymbol{h}_k^i$. Note the negative sign.

$$m_k^i(\delta_k^i) = -\sqrt{2\boldsymbol{h}_k^{iT}\Sigma_{x,k}\boldsymbol{h}_k^i}\ \mathrm{erf}^{-1}(2\delta_k^i - 1) \tag{14}$$

where erf^{-1} is the inverse of the Gauss error function. See Fig. 1 for the graphical interpretation of constraint tightening (13).

III. Two-stage Optimization Approach

A. Risk Allocation

Problem 1 can also be solved efficiently if it is reduced to Problem 2 (or Problem 3, equivalently). The only difference between Problem 1 and Problem 2 is chance constraints (6) and (7). Observe that, using the 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 (15), implies the *joint* chance constraint (6) [9].

$$\sum_{k=0}^{T} \sum_{i=1}^{N} \delta_k^i \leq \Delta \tag{15}$$

For later convenience, a vector $\boldsymbol{\delta}$ is defined as follows;

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

The key observation is that Problem 1 is reduced to Problem 2, once the upper bounds of the probability of violating individual constraints $\boldsymbol{\delta}$ are fixed so that (15) is satisfied. Then a question arises; *how to fix* $\boldsymbol{\delta}$? This problem can be viewed as a resource allocation problem; the goal is to find the optimal resource allocation $\boldsymbol{\delta}$ that maximizes the expected utility while the total amount of resource is limited to Δ. Thus we call $\boldsymbol{\delta}$ a "*risk allocation*".

B. Two-stage Optimization and Convexity

The previous observation naturally lead to a two-stage optimization approach; the upper-stage optimizes risk allocation $\boldsymbol{\delta}$ while lower-stage optimizes $\boldsymbol{U}$ given a risk allocation $\boldsymbol{\delta}$ (i.e. solving Problem 3). The upper-stage optimization problem is formally stated as follows.

Problem 4: Risk Allocation Optimization (Upper-stage)

$$\min_{\boldsymbol{\delta}} \quad \bar{J}^\star(\boldsymbol{\delta}) \tag{17}$$

$$\text{s.t.} \quad \sum_{k=0}^{T} \sum_{i=1}^{N} \delta_k^i \leq \Delta \tag{18}$$

$$\delta_k^i \geq 0 \tag{19}$$

$$\boldsymbol{\delta} \in \{\boldsymbol{\delta} \mid^{\exists} \boldsymbol{U}, \bar{\boldsymbol{X}} \text{ that satisfies } (11)-(13)\} \tag{20}$$

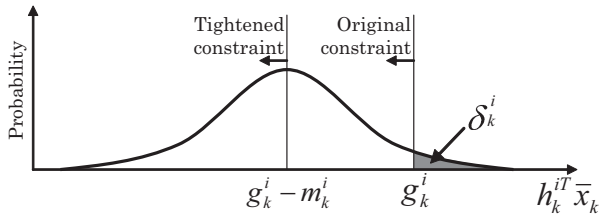

Fig. 1. Graphical interpretation of constraint tightening (13). The solid curve represents the probability distribution of $\boldsymbol{h}_k^{iT}\boldsymbol{x}_k$ when $\boldsymbol{h}_k^{iT}\bar{\boldsymbol{x}}_k = g_k^i - m_k^i(\delta_k^i)$. m_k^i is a safety margin imposed on the the center of distribution (i.e. nominal state $\bar{\boldsymbol{x}}$), so that the probability of violating the ith constraint at time k is less than the upper bound δ_k^i.

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

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

The original RMPC with joint chance constraint (Problem 1) is solved by solving Problem 4 (**upper-stage**) together with Problem 3 (**lower-stage**). An important fact is that Problem 4 is convex under a practical assumption.

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

The assumption $\Delta \leq 0.5$ is reasonable because allowable probability of failure is much less than 50% in most cases.

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 first show the convexity of the feasible region of $\boldsymbol{\delta}$. It is obvious that the region defined by (18) and (19) is convex, so we will show that the region defined by (20) is also convex. Since $(\boldsymbol{X}_1^\star, \boldsymbol{U}_1^\star)$ and $(\boldsymbol{X}_2^\star, \boldsymbol{U}_2^\star)$ are feasible solutions for Problem 3, $(\lambda\boldsymbol{X}_1^\star + (1-\lambda)\boldsymbol{X}_2^\star, \ \lambda\boldsymbol{U}_1^\star + (1-\lambda)\boldsymbol{U}_2^\star)$ satisfies (11) and (12) for linearity for all $0 \leq \lambda \leq 1$. It also satisfies (13), since

$$\begin{aligned} &\boldsymbol{h}^{iT}\{\lambda\bar{\boldsymbol{x}}_{k,1} + (1-\lambda)\bar{\boldsymbol{x}}_{k,2}\} \\ &\quad \leq g_k^i - \lambda m_k^i(\delta_{k,1}^i) - (1-\lambda)m_k^i(\delta_{k,2}^i) \\ &\quad \leq g_k^i - \lambda m_k^i(\lambda\delta_{k,1}^i + (1-\lambda)\delta_{k,2}^i) \end{aligned} \tag{22}$$

The second inequality holds because $m_k^i(\delta_k^i)$ is convex under a condition $0 \leq \delta_k^i \leq 0.5$, which is implied by the assumption $\Delta \leq 0.5$ and (19). The convexity of m_k^i immediately follows from the fact that $\mathrm{erf}(x)$ is convex for $x \leq 0$. Therefore, $(\lambda\boldsymbol{X}_1^\star + (1-\lambda)\boldsymbol{X}_2^\star, \lambda\boldsymbol{U}_1^\star + (1-\lambda)\boldsymbol{U}_2^\star)$ is a feasible (but not necessarily optimal) solution of Problem 3 for $\lambda\boldsymbol{\delta}_1 + (1-\lambda)\boldsymbol{\delta}_2$. Thus, the feasible region of Problem 4 is convex.

Next we prove the convexity of the objective function of Problem 4. Since Problem 3 is feasible for $\lambda\boldsymbol{\delta}_1 + (1-\lambda)\boldsymbol{\delta}_2$ for all $0 \leq \lambda \leq 1$, there is an optimum solution $(\boldsymbol{X}_\lambda^\star, \boldsymbol{U}_\lambda^\star)$ that gives a smaller or equal value of the objective function than $(\lambda\boldsymbol{X}_1^\star + (1-\lambda)\boldsymbol{X}_2^\star, \lambda\boldsymbol{U}_1^\star + (1-\lambda)\boldsymbol{U}_2^\star)$. Note that $\bar{J}(\boldsymbol{X}_\lambda^\star, \boldsymbol{U}_\lambda^\star)$ is equivalent to the objective function of Problem 4, $\bar{J}^\star(\lambda\boldsymbol{\delta}_1 + (1-\lambda)\boldsymbol{\delta}_2)$. Using convexity of $\bar{J}(\boldsymbol{X}, \boldsymbol{U})$,

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

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

IV. Subgradient Method

The upper-stage optimization (Problem 4) is convex, but its objective function is not always differentiable. The subgradient method is a standard optimization algorithm for such problems. We first derive the gradient and subgradient of the objective function, $\bar{J}^\star(\boldsymbol{\delta})$, which is necessary for the subgradient method.

A. Gradient

The derivation of the gradient of $\bar{J}$ is not trivial. For a feasible risk allocation $\boldsymbol{\delta}$ at which the objective function is differentiable,

$$\frac{\partial \bar{J}^\star}{\partial \delta_k^i} = \frac{\partial \bar{J}^\star}{\partial m_k^i}\frac{dm_k^i}{d\delta_k^i} \tag{24}$$

The second differential is obtained in a closed form as follows;

$$\frac{dm_k^i}{d\delta_k^i} = \frac{-1}{p_k^i(m_k^i(\delta_k^i))} \tag{25}$$

where $p_k^i(\cdot)$ is the probability distribution function of zero-mean Gaussian distribution with variance $\boldsymbol{h}_k^{iT}\Sigma_{x,k}\boldsymbol{h}_k^i$. Note the negative sign.

On the other hand, the first differential, $\frac{\partial \bar{J}^\star}{\partial m_k^i}$, is harder to obtain. Even in the simplest case where $\bar{J}$ is linear, it requires the following complicated procedure including matrix inversion.

First, Problem 3 is reformulated in a simple form as follows, by eliminating $\bar{\boldsymbol{X}}$ using (11);

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

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

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

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

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

where H_A is a n_U by n_U full rank matrix. The objective function $\bar{J}^\star(\boldsymbol{\delta})$ is differentiated by $\boldsymbol{m}$ as follows;

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

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

When there are more than n_U active independent constraints, $\bar{J}^\star(\boldsymbol{\delta})$ is not differentiable, hence we need to derive the subgradient.

B. Subgradient

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. Any "gradient" vector obtained from (30) using H'_A in place of H_A is included in the subgradient.

C. Subgradient Method

We used the projected subgradient method with a constant step size a and a non-summable diminishing step size $a/\sqrt{i}$. The constant step size achieves faster convergence, but only the diminishing step size can guarantee the convergence to the optimum [11][12]. The gradient have to be bounded in order to guarantee the convergence, but $\frac{\partial \bar{J}^\star}{\partial \delta_k^i} \to \infty$ when $\delta_k^i \to 0$. Therefore we used the following constraints in place of (19).

$$\delta_k^i \ge \epsilon \tag{32}$$

where $\epsilon > 0$ is a small number.

Since it is difficult to calculate the projection on the space defined by (18)-(20), we use the projection on the space defined by (18)-(19) instead. Therefore the subgradient method may fail to find a feasible solution even though the original problem is feasible.

V. Iterative Risk Allocation

As shown in the previous section, computation of gradient and subgradient is not trivial even in the simplest case where $\bar{J}$ is linear, since it involves n_U by n_U matrix inversion. Moreover, its convergence is often unstable for a large step size since the subgradient method is not a descent algorithm. Using small step size leads to a slow convergence.

We developed a descent algorithm called Iterative Risk Allocation (IRA) to solve Problem 4. Instead of using subgradient that is hard to compute, IRA exploits the fact that $\bar{J}^\star$ is a decreasing function of δ_k^i to find a descent direction. We first prove that $\bar{J}^\star$ is indeed a decreasing function.

A. Monotonicity of $\bar{J}^\star$

Theorem 2: Monotonicity of $\bar{J}^\star$

$$\forall k, i \quad \frac{\partial \bar{J}^\star}{\partial \delta_k^i} \le 0$$

Proof: Let $\boldsymbol{\delta}$ and $\boldsymbol{\delta}'$ be risk assignments, and $\mathcal{R}(\boldsymbol{\delta})$ and $\mathcal{R}(\boldsymbol{\delta}')$ be the corresponding feasible region of $(\bar{\boldsymbol{X}}, \boldsymbol{U})$ in Problem 3 defined by (11)-(13). If $\delta_k^i \le \delta_k^{i\prime}$ for all (k,i), then $\mathcal{R}(\boldsymbol{\delta}) \subseteq \mathcal{R}(\boldsymbol{\delta}')$ since m_k^i is a decreasing function of δ_k^i. Therefore $\bar{J}^\star(\boldsymbol{\delta}) \ge \bar{J}^\star(\boldsymbol{\delta}')$ and thus Theorem 2 holds since $\bar{J}^\star(\boldsymbol{\delta})$ is the minimum of $\bar{J}(\boldsymbol{\delta})$ in $\mathcal{R}(\boldsymbol{\delta})$. ■

When Problem 3 is a linear programming problem, the optimal solution is always on the intersection of active constraints. Therefore $\bar{J}^\star$ is a strictly decreasing function of δ_k^i when the ith constraint at time k is active, as is stated in the following lemma.

Lemma 1: Strict monotonicity of $\bar{J}^\star$ when Problem 3 is LP

$$\frac{\partial \bar{J}^\star}{\partial \delta_k^i} < 0 \quad \text{if } \boldsymbol{h}_k^{iT}\bar{\boldsymbol{x}}_k^\star = g_k^i - m_k^i(\delta_k^i) \text{ and } \bar{J} \text{ is linear.}$$

B. A Descent Algorithm

Given a feasible risk assignment $\boldsymbol{\delta}_{(0)}$, it is able to construct a sequence of feasible risk assignments $(\boldsymbol{\delta}_{(0)}, \boldsymbol{\delta}_{(1)}, \boldsymbol{\delta}_{(2)}, \cdots \boldsymbol{\delta}_{(n)})$ such that $\bar{J}^\star(\boldsymbol{\delta}_{(0)}) \geq \bar{J}^\star(\boldsymbol{\delta}_{(1)}) \geq \cdots \geq \bar{J}^\star(\boldsymbol{\delta}_{(n)})$ by using Theorem 2 as follows.

Assume that $\boldsymbol{\delta}_{(n)}$ is a feasible risk allocation that satisfies (18)-(20). We first construct $\boldsymbol{\delta}'_{(n)}$ by tightening the inactive constraints. For all (i,k) where ith constraint at time k is active, set $\delta^{i\,\prime}_{k(n)} = \delta^i_{k(n)}$. For all (i,k) where ith constraint at time k is inactive, choose $\delta^{i\,\prime}_{k(n)} < \delta^i_{k(n)}$ so that

$$\boldsymbol{h}_k^{iT}\bar{\boldsymbol{x}}^\star_k < g^i_k - m^i_k(\delta^{i\,\prime}_{k(n)}) < g^i_k - m^i_k(\delta^i_{k(n)}). \tag{33}$$

where $\bar{\boldsymbol{x}}^\star_k$ is the optimal nominal state at k given $\boldsymbol{\delta}_{(n)}$. The constraint (33) ensures that the optimal solution for $\boldsymbol{\delta}_{(n)}$ is feasible for $\boldsymbol{\delta}'_{(n)}$. It follows from $\delta^{i\,\prime}_{k(n)} < \delta^i_{k(n)}$ that $\mathcal{R}(\boldsymbol{\delta}') \subseteq \mathcal{R}(\boldsymbol{\delta})$ where $\mathcal{R}(\boldsymbol{\delta})$ is a feasible region of $(\bar{\boldsymbol{X}}, \boldsymbol{U})$ for $\boldsymbol{\delta}$. Therefore the optimal solution for $\boldsymbol{\delta}_{(n)}$ is also the optimal solution for $\boldsymbol{\delta}'_{(n)}$, and thus $\bar{J}^\star(\boldsymbol{\delta}_{(n)}) = \bar{J}^\star(\boldsymbol{\delta}'_{(n)})$. Finally, $\boldsymbol{\delta}'_{(n)}$ is feasible, because; (i) (18) is satisfied since $\sum_{k=0}^T \sum_{i=1}^N \delta^{i\,\prime}_{k(n)} < \sum_{k=0}^T \sum_{i=1}^N \delta^i_{k(n)}$; (ii) (19) follows from (33) and the fact that $m^k_i(\delta) \to \infty$ when $\delta \to 0$; (iii) (20) is satisfied since all tightened constraints are inactive for $\boldsymbol{\delta}'_{(n)}$.

Next we construct $\boldsymbol{\delta}_{(n+1)}$ from $\boldsymbol{\delta}'_{(n)}$ by loosening the active constraints. For all (i,k) where ith constraint at time k is inactive, set $\delta^i_{k(n+1)} = \delta^{i\,\prime}_{k(n)}$. For all (i,k) where ith constraint at time k is active, choose $\delta^i_{k(n+1)} > \delta^{i\,\prime}_{k(n)}$ so that (18) is satisfied. It is straightforward to show that $\boldsymbol{\delta}_{(n+1)}$ is feasible. Theorem 2 implies that $\bar{J}^\star(\boldsymbol{\delta}'_{(n)}) \geq \bar{J}^\star(\boldsymbol{\delta}_{(n+1)})$. Therefore,

$$\bar{J}^\star(\boldsymbol{\delta}_{(n)}) = \bar{J}^\star(\boldsymbol{\delta}'_{(n)}) \geq \bar{J}^\star(\boldsymbol{\delta}_{(n+1)}). \tag{34}$$

By constructing the sequence $(\boldsymbol{\delta}_{(0)}, \boldsymbol{\delta}_{(1)} \cdots \boldsymbol{\delta}_{(n)})$ recursively in this manner, the objective function $\bar{J}^\star$ monotonically decreases. When $\bar{J}$ is linear, a stronger bound (35) can be obtained by using Lemma 1;

$$\bar{J}^\star(\boldsymbol{\delta}_{(n)}) = \bar{J}^\star(\boldsymbol{\delta}'_{(n)}) > \bar{J}^\star(\boldsymbol{\delta}_{(n+1)}). \tag{35}$$

Observe that if all constraints are active or all constraints are inactive for $\boldsymbol{\delta}_{(n)}$, it is impossible to construct $\boldsymbol{\delta}_{(n+1)}$ using the procedure described in the previous subsection. Actually having all constraints *inactive* is a sufficient condition for the optimality of Problem 4. On the other hand, having all constraints *active* is not a sufficient condition.

C. Iterative Risk Allocation Algorithm

The discussions in the previous subsections lead to a simple yet very powerful descent algorithm called Iterative Risk Allocation (IRA), which is described in Algorithm 1. It is initialized by a uniform risk allocation (Line 1). The lower-stage optimization problem is solved in Line 4 to compute the optimal solution for current risk allocation $\boldsymbol{\delta}$. Line 6 terminates the algorithm when all constraints are active or inactive. Line 10 tightens inactive constraints according to (33) with a parameter $0 < \alpha < 1$, while Line 14 loosens active constraints. In Line 10, $1 - \mathrm{cdf}^i_k(g^i_k - \boldsymbol{h}_k^{iT}\bar{\boldsymbol{x}}^\star_k)$ is the probability of violating a constraint at (k,i), where $\mathrm{cdf}^i_k(\cdot)$ is the cumulative distribution function of the Gaussian distribution with variance $\boldsymbol{h}_k^{iT}\Sigma_{x,k}\boldsymbol{h}^i_k$, and $\bar{\boldsymbol{x}}^\star_k$ is the optimized nominal state for $\boldsymbol{\delta}$.

It follows from the discussion in the previous subsection that the IRA algorithm generates a sequence of feasible risk assignments $(\boldsymbol{\delta}_{(0)}, \boldsymbol{\delta}_{(1)}, \cdots \boldsymbol{\delta}_{(n)})$ that monotonically decreases the objective function $\bar{J}^\star(\boldsymbol{\delta})$.

In the next section the performance of IRA algorithm is compared with the subgradient method as well as two prior arts; the ellipsoidal relaxation approach and the Particle Control, using simulations.

VI. Simulation

The performance comparison of five algorithms are compared on simulation. The five algorithms are;
-**IRA:** Two-stage optimization with IRA for the upper-stage
-**SM(d):** Two-stage optimization with the subgradient method for the upper-stage (diminishing step size)
-**SM(c):** Two-stage optimization with the subgradient method for the upper-stage (constant step size)
-**ER:** Ellipsoidal relaxation approach [1]
-**PC:** Particle Control [2]
The subgradient method is not a descent algorithm and hence the stop condition is hard to define; therefore we computed a fixed number (300) of iterations.

Following parameters are used;

$$J(\boldsymbol{X},\boldsymbol{U}) = \sum_{k=1}^T |u_k|, \quad \Delta = 0.05, \quad T = 10, \quad N = 2$$
$$A = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}, \quad B = \begin{pmatrix} 0 \\ 0.033 \end{pmatrix}$$
$$u_{\max} = 0.2, u_{\min} = -0.2, \boldsymbol{h}_k^{1T} = (1,0), \boldsymbol{h}_k^{2T} = (-1,0)$$

Algorithm 1 Iterative Risk Allocation

1: $\forall(k,i) \quad \delta^i_k \leftarrow \Delta/(T \cdot N)$
2: **while** $|\bar{J}^\star - \bar{J}^\star_{prev}| < \epsilon$ **do**
3: $\quad \bar{J}^\star_{prev} \leftarrow \bar{J}^\star$
4: $\quad$ Solve Problem 3 with $\boldsymbol{\delta}$.
5: $\quad N_{active} \leftarrow$ number of steps where constraint is active
6: $\quad$ **if** $N_{active} = 0$ or $N_{active} = T \cdot N$ **then**
7: $\quad\quad$ **break**;
8: $\quad$ **end if**
9: $\quad$ **for all** (k,i) such that ith constraint at kth time step is inactive **do**
10: $\quad\quad \delta^i_k \leftarrow \alpha\delta^i_k + (1-\alpha)\left\{1 - \mathrm{cdf}^i_k(g^i_k - \boldsymbol{h}_k^{iT}\bar{\boldsymbol{x}}^\star_k)\right\}$
11: $\quad$ **end for**
12: $\quad \delta_{residual} \leftarrow \Delta - \sum_{k=0}^T \sum_{i=1}^N \delta^i_k$
13: $\quad$ **for all** (k,i) such that ith constraint at kth time step is active **do**
14: $\quad\quad \delta^i_k \leftarrow \delta^i_k + \delta_{residual}/N_{active}$
15: $\quad$ **end for**
16: **end while**

$$x_0 = \begin{pmatrix} 0.01 \\ 0 \end{pmatrix}, \quad \Sigma_{x,0} = \Sigma_w = \begin{pmatrix} 0.001 & 0 \\ 0 & 0 \end{pmatrix}$$

The bounds, g_k^1 and g_k^2, are randomly generated. We used $\alpha = 0.7 \cdot 0.98^n$ (with n being iteration index) for IRA and $a = 0.001$ for the subgradient methods. These are the largest step sizes that can achieve stable convergence. We set $\epsilon = 10^{-8}$ for (32).

The performance of the five algorithms is compared in Table I. The numbers in the table are the average of 237 randomly generated problems. All three two-stage optimization algorithms with risk allocation (IRA, SM(d), and SM(c)) has much less suboptimality than the ellipsoidal relaxation approach while achieving a significant speed up compared to Particle Control.

The *probability of failure* in Table I is defined as follows;

$$\text{Probability of failure} := 1 - \Pr\left[\bigwedge_{k=0}^{T}\bigwedge_{i=1}^{N} \boldsymbol{h}_k^{iT}\boldsymbol{x}_k \le g_k^i\right]$$

We used Monte Carlo simulation with 10,000 samples to evaluate the probability of failure. Difference between the probability of failure and $\Delta = 0.05$ is the measure of conservatism. The ellipsoidal relaxation has strong conservatism, which leads to a large suboptimality.

Figure 2 compares the convergence speed of IRA and the subgradient methods on a typical problem. The convergence of IRA is significantly faster than the subgradient methods. The weakness of IRA is the lack of the theoretical guarantee of convergence to the optimal. However, the empirical result shows that the suboptimality is considerably small. Table I shows that IRA yields even better solution than the subgradient methods after 300 iterations on average. Figure 3 is the histogram of $\bar{J}^\star_{\mathrm{IRA}} - \min(\bar{J}^\star_{\mathrm{SM(d)}}, \bar{J}^\star_{\mathrm{SM(c)}})$. The objective function value of IRA is smaller or equal to the objective function value of both subgradient methods in most cases; IRA yields worse solution in several cases, but the difference is less than 0.01 in those cases; on the other hand, the subgradient methods may be worse than IRA by up to 0.08.

VII. Conclusion

A novel two-stage optimization method for robust Model Predictive Control (RMPC) with Gaussian disturbance is presented. We proved that its upper-stage is a convex optimization, but the objective function is not always differentiable. We developed a descent algorithm for the upper-stage called Iterative Risk Allocation, which showed fast convergence and a small suboptimality in simulations.

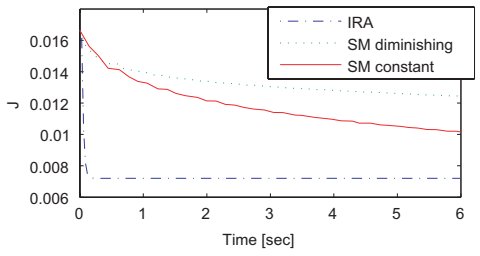

Fig. 2. Convergence of IRA and the subgradient methods

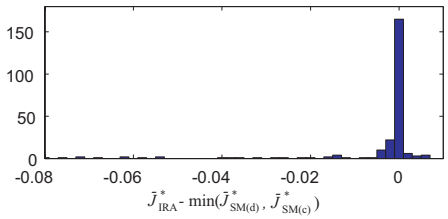

Fig. 3. Histogram of the difference in the objective function value of IRA and both subgradient methods.

Acknowledgments

Thanks to Michael Kerstetter, Scott Smith, Ronald Provine and the Boeing Company for their support. Lars Blackmore and Yoshiaki Kuwata provided valuable feedback.

References

[1] 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.
[2] Lars Blackmore. A probabilistic particle control approach to optimal, robust predictive control. In *Proceedings of the AIAA Guidance, Navigation and 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.
[5] Johan Löfberg. *Minimax Approaches to Robust Model Predictive Control*. PhD thesis, Linköping Studies in Science and Technology, 2003.
[6] Yoshiaki Kuwata, Arthur Richards, and Jonathan How. Robust receding horizon control using generalized constraint tightening. *Proceedings of American Control Conference*, 2007.
[7] Pu Li, Moritz Wendt, and Günter Wozny. A probabilistically constrained model predictive controller. *Automatica*, 38:1171–1176, 2002.
[8] Jun Yan and Robert R. Bitmead. Incorporating state estimation into model predictive control and its application to network traffic control. *Automatica*, 41:595–604, 2005.
[9] Lars Blackmore, Hui Li, and Brian C. Williams. A probabilistic approach to optimal robust path planning with obstacles. In *Proceedings of American Control Conference*, 2006.
[10] Masahiro Ono and Brian C. Williams. An efficient motion planning algorithm for stochastic dynamic systems with constraints on probability of failure. In *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence*, 2008.
[11] N. Z. Shor. *Minimization Methods for Non-Differentiable Functions*. Springer-Berlag, 1985.
[12] Dimitri P. Bertsekas. *Nonlinear Programming*. Athena Scientific, 1999. Dewey Library - Reserve Stacks — T57.8.B47 1999.

TABLE I

Performance of two-stage optimization methods with risk allocation and prior arts; values are the average of 237 randomly generated problems.

Algorithm	IRA	SM (d)	SM (c)	ER	PC
$J^\star$	0.0906	0.0978	0.0957	0.3502	0.0959
Comp. time [sec]	0.33	26.4	30.7	0.05	212.2
Prob. of failure	0.0378	0.0183	0.306	$< 10^{-5}$	0.0281