ISSN 2433-2224(Online)
JAXA-RM-23-003

宇宙航空研究開発機構研究開発資料

JAXA Research and Development Memorandum

統計的最適化におけるモンテカルロ・シミュレーション評価回数更新法の見直し

Modification for Updating Number of Monte Carlo Simulations in Stochastic Parameter Optimization

元田 敏和

MOTODA Toshikazu

2024年1月

宇宙航空研究開発機構

Japan Aerospace Exploration Agency

目次

統計的最適化における
モンテカルロ・シミュレーション評価回数更新法の見直し

元田 敏和[*1]

Modification for Updating Number of Monte Carlo Simulations in Stochastic Parameter Optimization

MOTODA Toshikazu[*1]

ABSTRACT

Stochastic Parameter Optimization (SPO) method utilizes Monte Carlo Simulation (MCS) to obtain the probability of failure that should be minimized. Since MCS incorporates various uncertain parameters into a nonlinear system, the optimized system can become robust against the uncertainties. On the other hand, SPO requires great amounts of computational resources because MCS must be repeatedly performed during the optimization process. In order to alleviate the computational burden, the number of simulations for each MCS is increased as the SPO proceeds and the MCS result improves. The objective of the number update is to reduce the computational cost when the probability of failure is relatively high. In the current algorithm, the number of MCS sometimes becomes unexpectedly large, and as a result more computational time is required. It might be inconvenient in the actual development process where design term is often limited. So, the algorithm for increasing the number of MCS is modified to evaluate the system appropriately at any time during the optimization process. The effects of the modification are confirmed using a real experimental flight model that was used in the past flight test.

Keywords: Stochastic Parameter Optimization, Monte Carlo simulation, Confidence Interval, Robust System

概　要

統計的最適化(SPO)では，モンテカルロ・シミュレーション(MCS)を利用して評価関数となる失敗確率を得て，それを最小化する．MCS では様々な不確定パラメータを同時に非線形システムに組み込むため，最適化されたシステムはこれらの不確定パラメータに対してロバストとなる．その一方で，SPO では繰り返し MCS 計算を実行する必要があるため，計算負荷が極めて大きくなる．この計算負荷を緩和するため，MCS におけるシミュレーションの評価回数を最適化の過程で変化させており，計算が進行して MCS 結果が改善されるにつれて，評価回数を増加させている．本稿の目的は，最適化の過程におけるシミュレーション評価回数を，より合理的な方法で抑制し，計算負荷を軽減することである．現状の

[*] 2023 年 10 月 24 日受付 (Received October 24, 2023)
[*1] 航空技術部門 航空環境適合イノベーションハブ (Aviation Environmental Sustainability Innovation Hub, Aviation Technology Directorate)

アルゴリズムでは，評価回数が突然大きく増加することがあり，その結果として全体の計算負荷が増大する事象が発生している．開発期間に制約がある実際の設計現場などにおいては，この状況は望ましいとはいえない．そこでシミュレーション評価回数の増加ロジックをより合理的な方法として見直し，計算負荷の抑制を試みた．またロジック修正の効果を，過去の飛行実験で用いられた現実的なシミュレーション・モデルを用いて検証した．

記号

$\boldsymbol{d}$: 設計ベクトル
H : 高度
J : 評価関数
K_D: 微分ゲイン
K_F: フィードフォワード･ゲイン
K_I: 積分ゲイン
K_P: 比例ゲイン
k: 設計パラメータ
N: MCS 評価回数
N_{TOTAL}: 最適化の全シミュレーション回数
n: 設計パラメータ数
P_F: 真の失敗確率
P_{fail}: サンプルの失敗確率
P_U: 失敗確率の信頼区間･上限値
p_0: N回の試行で一度も失敗しない確率
V_g: 対地速度
X: 滑走路方向位置
Y: 滑走路中心線からの垂直方向位置
$\dot{Z}$: 沈下率
α: 危険率
β_g: 対地横滑り角
γ: 経路角
Θ: ピッチ角
ρ: 相関係数
Φ: ロール角

添字

best: 最良値
F: 失敗ケース
hi: SIMPLEX 上の最大(悪)値
hold: 一時保存値
new: 発生させた新たな値
ref: 基準軌道

1. はじめに

様々な不確定要因を考慮したシステムに対する設計パラメータの最適化において, モンテカルロ・シミュレーション(MCS)による評価を組み込んだ最適化法がある [1-3]．その手法はいくつか提案されており, 遺伝的アルゴリズムを用いる方法 [1], 計算負荷を大幅に低下させる方法 [2]，Downhill-SIMPLEX 法 [4]と焼鈍し法 [5]を組み合わせた方法 [3] などである. いずれの方法も様々な不確かさを含んだ非線形システムを MCS で評価し，失敗確率を最小とする設計パラメータを探索する．MCS 結果を直接最適化できるため, 様々な不確定パラメータに対してロバストな設計結果が得られる利点がある．その一方で，最適化計算の過程で MCS を多数回実行する必要があるため, 計算負荷が高く, 結果を得るまでに時間を要するという問題がある．この問題は並列計算を利用することで，ある程度は緩和できる [6].

当機構では計算負荷と最適化性能の双方を勘案し，実用上の観点から文献[3]の手法に基づいて並列計算システムを構築した [7]. 最適化法の概要とその詳細なアルゴリズムは，文献[8]および[9]にそれぞれ記してある．本アルゴリズムでは計算負荷緩和策の一環として，MCS 評価回数を最適化計算の過程で変化させている. 最適化結果の良さだけを追求するのであれば，MCS の評価回数Nは最初から大きく設定したほうが良い．なぜなら MCS の出力は失敗確率という統計量であり，Nが大きいほど大数の法則により MCS 結果の信頼性は高くなるからである. しかしそれでは過大な計算負荷から, 実際の設計現場での利用に

おいて障害となる.

大きな計算負荷を緩和するためのソフトウェア的な方策として, 最適化計算の初めは比較的小さいNで MCS 評価を実行し, 探索が進んで MCS 結果が改善すると共に, つまり失敗確率が低下すると共に, Nを増加させる. このようにすれば一定程度, 計算負荷を軽減することが可能となる. 但しこの場合, MCS 結果の精度をある程度犠牲にしている面があり, 最適化性能は一定程度劣化する可能性はある. 実用性の観点から初期のNを抑制するものであり, 許容できる計算負荷であるなら, Nはできるだけ大きな値とするほうが最適化性能の追求には望ましい. つまり最良値は得られないとしても, 限られた時間内に一定の改善された結果が得られることが実用上の利点である. それでも, あらゆる不確定性を考慮したシステムの性能改善が期待できることに変わりはない. これが本最適化法の大きなメリットである.

本稿では文献[9]で提案された MCS の評価回数Nの更新方法とその問題点を示し, その後に改善策について述べる. 以下では, これまでの信頼区間を利用したN更新の方法を示す. 次にその問題点に言及し, 修正案を提示する. 最後に修正した最適化法を具体的な飛行システムに適用した例を示し, 修正前後のN増加の様子と計算負荷の違いを比較して考察する.

2. N更新の考え方

MCS の評価回数N は, 失敗確率が比較的大きいときにはそれほど大きくする必要はなく, 最適化計算が進み改善されるのに応じて大きくしていく必要がある. この理由を図 1 に示す. 失敗確率をP_{fail}としたとき, $P_{fail} = 0.1$のシステムでは平均的に 10 回の試行で 1 回は失敗ケースが検出できる. ところがP_{fail}が小さくなり, 例えば$P_{fail} = 0.01$のシステムでは平均的に 100 回の試行でようやく 1 回の失敗ケースが検出できる. つまり$P_{fail} = 0.01$のシステムを 10 回の試行で評価

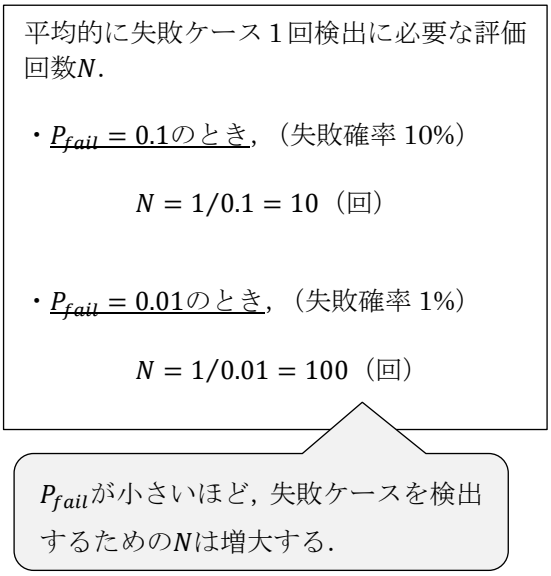


図 1　失敗確率と評価回数

しても失敗ケースが検出されない確率が高くなる. つまり, システムを適切に評価できなくなる. 以上より, 失敗確率が小さいシステムほど, より大きなNを必要とすることになる.

但しP_{fail}の値は MCS 実行前には不明であるため, 信頼区間 [10]を考慮した過不足のないNを事前に算出することは不可能である. よって, 失敗ケースをほぼ確実に検出できるようなNを, 以下の考え方で求めている.

システムの真の失敗確率をP_Fとする. いまN回評価したときに, x回失敗が現れる確率$p(x)$は, 二項分布 [10,11]の次式より得られる.

$$p(x) = {}_N C_x \cdot {P_F}^x \cdot (1 - P_F)^{(N-x)} \qquad (1)$$

よって, 一度も失敗が現れない確率$p(0) = p_0$は,

$$p_0 = (1 - P_F)^N \qquad (2)$$

式(2)を変形すると,

$$N = \ln(p_0)/\ln(1 - P_F) \qquad (3)$$

p_0はN回の評価で一度も失敗ケースが現れない確率であるから, 少なくとも一回は失敗ケースが

現れる確率は$(1-p_0)$である. よって式(3)で表されるNよりも大きな評価回数であれば，$(1-p_0)$の確率で少なくとも一回は失敗ケースが現れることになる. 例えば$p_0=0.01$と設定すれば, 式(3)で表されるN回の試行で，99%［=1-0.01］の確率で少なくとも一回は失敗が現れる.

式(3)に基づくP_FとNの関係を，縦軸・横軸共に対数スケールとして図 2 に示す. P_Fが小さくなるにつれて，失敗ケースを検出するために必要なNは, このスケール上で直線的に増加していくことがわかる. またp_0を小さくすれば，つまりより確実に（高い確率で）失敗ケースを検出するためには，必要とされるNは大きくなる.

従って，基本的に式(3)で表される関係を用いて，最適化計算の中で評価回数Nを更新する．ここでP_Fは, 真の失敗確率であるので未知数である. この値として, 最適化計算の過程でそれまでに出現した失敗確率の最小値を用いた. 一方p_0は失敗ケースが現れない確率であり, 設計者が設定する調整パラメータとした.

3. これまでのN更新方法

Nの更新には式(3)を用いるが，P_Fの設定についてはそれほど単純ではない. これは各設計点が最適化の過程で，異なるNにより MCS 評価されることによる．以下ではまず文献[9]で用いた最良値の判定方法について述べ，その後に式(3)における具体的なP_Fの設定方法とその問題点を明らかにする.

3.1. 設計ベクトルの最良値

基本的には, 失敗確率が最小となる設計ベクトルを最良とする. ただし最適化計算の進行と共にNを増加させていくため，各設計点は同じ評価回数Nで MCS 評価されているわけではない．大数の法則よりNが大きいほど結果の信頼性は高いため，単純に失敗確率同士を比較し，小さい方が優れていると判定することは適切ではない. このため信頼区間 [10]を考慮し, その上限値P_U, つまり

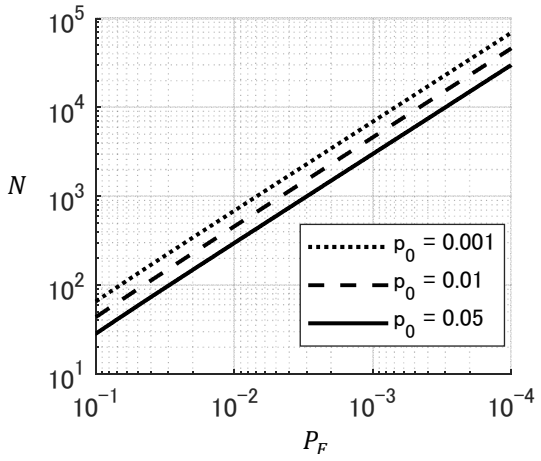


図 2 P_F, p_0と試行回数N

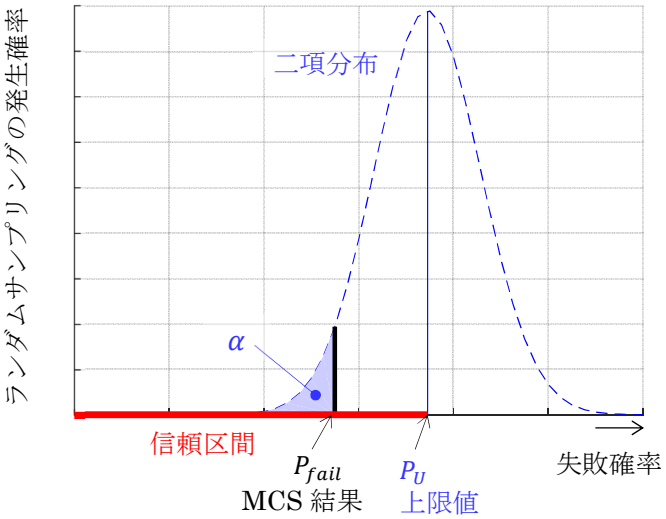


図 3 失敗確率の上限値

最悪値を比較し，P_Uが最も小さい設計ベクトルをその時点での最良とする.

失敗確率P_{fail}と上限値P_Uの関係を, 図 3 に示す [12]. MCS 結果から得られる失敗確率がP_{fail}である．いま仮に真の失敗確率がP_Uであるとすると，実際に得られた結果であるP_{fail}以下となる確率がαであることを表す. αは危険率であり, 事前に設定するパラメータである．一般的には$\alpha=0.05$が用いられることが多い. つまりP_Uは，間違える確率αを受け入れることを前提として，失敗確率の最大（最悪）値を表す．またNが大きいほど信頼区間幅は小さくなり，P_UはP_{fail}に近づく．これはサンプルによるばらつきが小さくなり，P_{fail}の

精度が上がることを意味する.

このため小さなNで MCS 評価された$P_{fail\cdot1}$が, 大きなNで評価された$P_{fail\cdot2}$よりも小さいとしても, それらの上限値P_Uは逆転することもある. この例を図 4 に示す. 設計点 1 は$N=200$で評価され$P_{fail\cdot1}=0.025$であり, 一方設計点 2 は$N=400$で評価され$P_{fail\cdot1}=0.03$であるとする. このとき失敗確率は$P_{fail\cdot1}<P_{fail\cdot2}$であり, 設計点 1 のほうが優れている. ところが信頼区間の上限値では, 設計点 1 の$P_{U\cdot1}=0.0518$に対して設計点 2 は$P_{U\cdot2}=0.0482$であり, $P_{U\cdot1}>P_{U\cdot2}$となって優劣が逆転する.

以上より本最適化計算では, 各設計点においてP_Uを算出し, その時点で最も小さなP_U値をもつ設計点を最良値としている.

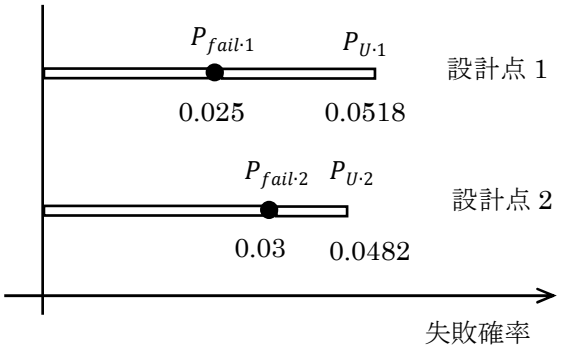
3.2. N更新の問題

基本的には, 前項で述べたP_U最小となる設計点が最良値という考え方である. しかし単純に最良値における失敗確率P_Fを用い, 式(3)により新たな評価回数Nを求めると, 不都合が生じることがある. その事例を, 以下に述べる.

最適化における新たな設計点の発生ロジックとして Downhill-SIMPLEX 法(DS 法)[4]を利用している. DS 法では, 設計点がn次元ベクトル(設計パラメータがn個)であるとき, $(n+1)$個のベクトルで構成される SIMPLEX と呼ばれる多面体を変形しながら, 最適解を探索する. 例えば設計パラメータがk_1, k_2の 2 個であるときは, 設計ベクトル$\boldsymbol{d}_i=[k_{1\cdot i}\quad k_{2\cdot i}]^T$として, 図 5 に示すように 3 角形を構成する. 各$\boldsymbol{d}_i$を MCS 評価することで, それぞれに対応するP_Uが求められる. DS 法ではこの SIMPLEX の頂点のうち, 最もP_Uが大きな設計ベクトル$\boldsymbol{d}_{hi}$を移動させていく. つまり SIMPLEX 上のP_Uの差を利用して, 悪い値をもつ設計ベクトルを移動させ, 最適値を探索していく. 探索が進むにつれて, SIMPLEX 多面体は小さくなり, 極値に収束していく.

ところが SIMPLEX 多面体は十分に収束していないにも関わらず, 全ての頂点でP_Uの値が一致することがある. これは評価関数の多峰性に起因することもあるし, またNの大きさが十分でなくP_{fail}の精度不足による場合もある. いずれにしてもこのときには, SIMPLEX 上のP_Uに差が現れるまで, Nを増加させる必要がある. ところが, 式(3)を単純に適用してもNが増加しない事象が発生することがある.

この事例を表 1 に示す. その時点における最良値(最小値)$P_{U\cdot best}=0.0184$を得たときの評価回数, 失敗数をそれぞれ$N_{best}=500$, $N_{F\cdot best}=2$と



設計点	N	N_F	P_{fail}	P_U
1	200	5	0.025	0.0518
2	400	12	0.030	0.0482

図 4　失敗確率の上限値例

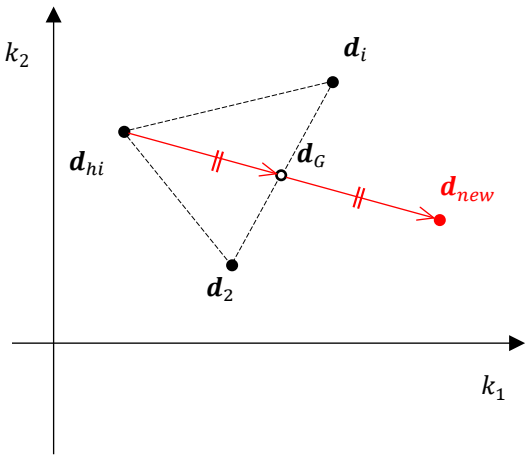


図 5　DS 法による設計点の更新

表 1 N更新の数値例

	N	N_F	P_U	N_{new}
最良値	500	2	0.0184	1149
	↓	↓	↓	
現在値	1149	11	0.0197	479
	1149	**2**	−	**2644**

する．このとき式(3)から算出される新たな評価回数は$N_{new} = 1149$となる．この評価回数において，現在のすべての SIMPLEX 上の失敗数が$N_F = 11$となり，P_Uに差がなくなったとする．このとき差が現れるまでNを増やす必要があるが，最良値に式(3)を適用した結果が$N_{new} = 1149$であり，これ以上増やしようがない．このような状態に陥ると，SIMPLEX 上のP_Uに差がないため，これ以上探索を進めることができなくなる．何らかの方法でP_Uに差が現れるまで評価回数Nを増やす必要がある．

上記の問題を解消するため，式(3)適用の際のP_F算出に用いる評価回数を，最良値に対応するN_{best}(=500)ではなく，現時点のNの値(=1149)を使用することとしている．表 1 の例では$N = 1149$, $N_F = 2$として，式(3)を適用することとした．この結果$N_{new} = 2644$となり，評価回数を増加させることができるようになる．

但し，式(3)適用時に現時点の$N = 1149$を用いるが，失敗数だけは最良値に対応する$N_F = 2$を採用することに，確率的な合理性はない．便宜上N_{new}を増加させるための方策として用いているだけである．この結果，Nが急激に増加して計算負荷が増大する事象が発生する．

4. N更新方法の修正

前項のNの更新法では失敗数だけ$P_{U\cdot best}$に対応する回数$N_{F\cdot best}$を用いるものであるが，式(3)から得られるN_{new}は急激に大きくなる傾向がある．何とかNを増やす必要があるケースが発生する

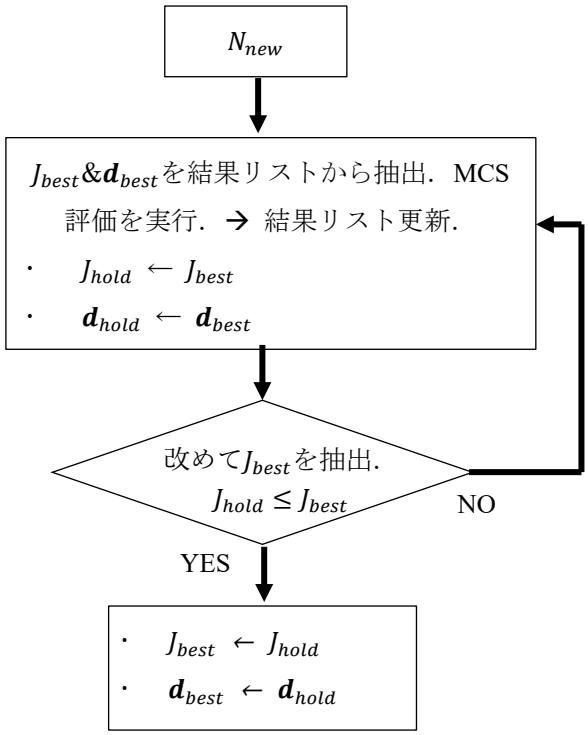


図 6 最良値の更新アルゴリズム

とはいえ，失敗数だけ$N_{F\cdot best}$を用い分母のNは現在値を用いることは，確率的な議論として合理性に欠けることに加え，Nが過大になることで最適化計算全体での計算負荷も大きくなる．そもそも最適化計算の過程でNを増加させるのは，計算負荷を抑制するためである．

以上を考慮し，より合理的なNの増加ロジックとして，「最良値は，常にその時点の評価回数Nで MCS 評価される」ように修正する．つまり式(3)で用いる最良値$P_{F\cdot best}$は，常に現時点の評価回数Nを用いて算出された値とする．表 1 の例では，最良値は$N = 500$でしか評価されていないが，この設計ベクトルも$N = 1149$で評価する．

これを実現するためのアルゴリズムを図 6 に示す．この処理はNを更新する度に実行する．実際の最適化計算では，評価関数をJとしており，P_Uの常用対数として次式で表される．

$$J = \log_{10} P_U \tag{4}$$

新旧の評価関数を比較するロジックの都合でP_Uそのものではなく，その対数であるJを用いているが，実質的にはP_Uと同じものであり，その大小関係に変化はない．

図 6 において評価回数がN_{new}に更新されたとき，最良値J_{best}に対応する設計ベクトル$\boldsymbol{d}_{best}$を結果リストより取り出す．次に$\boldsymbol{d}_{best}$を評価回数N_{new}で MCS 評価し直し，J_{best}，$\boldsymbol{d}_{best}$を更新する．そして結果リストの情報も更新する．但し，このとき更新されたJ_{best}が最良（最小）であるとは限らないため，その結果をJ_{hold}，$\boldsymbol{d}_{hold}$として一時保存しておく．その後改めて最良となるJ_{best}を，過去の MCS 評価結果のリストより取り出す．

ここで$J_{hold} \leq J_{best}$であれば，一時保存していた値が最良値であるため，改めて$J_{hold} \rightarrow J_{best}$，$\boldsymbol{d}_{hold} \rightarrow \boldsymbol{d}_{best}$と戻して最良値を確定させる．一方で，$J_{hold} > J_{best}$となれば，保持していた値は最良値ではないため，改めて最良値を取り出すところから，上記の手順を繰り返す．そして評価回数N_{new}における最良値を確定させる．以上の手順により，最良値は常にN_{new}で MCS 評価された状態となる．なお以上の修正アルゴリズムは，当機構が保有する MCS 並列計算システム [7]にも反映した．

5. 適用例

実際の飛行実験に用いられた飛行シミュレーション・プログラムを用いて，最適化計算における評価回数N更新の効果を確認する．ここで取り上げるのは，小型自動着陸実験(ALFLEX)[13-16]であり，この誘導パラメータの最適化を実行する．以下では，はじめに ALFLEX の概要について述べ，その後に最適化結果を示してN更新の効果を確認する．

5.1. ALFLEX 飛行実験

本飛行実験は将来の宇宙往還機を想定した自動着陸技術の獲得を目的としたもので，1996 年にオーストラリアのウーメラ飛行場にて実施された．13 回の着陸実験が試みられ，全て成功している．実験プロファイルを図 7 に，実験機の 3 面図を図 8 に示す．実験機はヘリコプターから 1 本のケーブルで懸吊されて飛行し，滑走路手前の高度 1500m で分離される．実験機は推力を持たず滑空し，搭載計算機によって制御され，滑走路

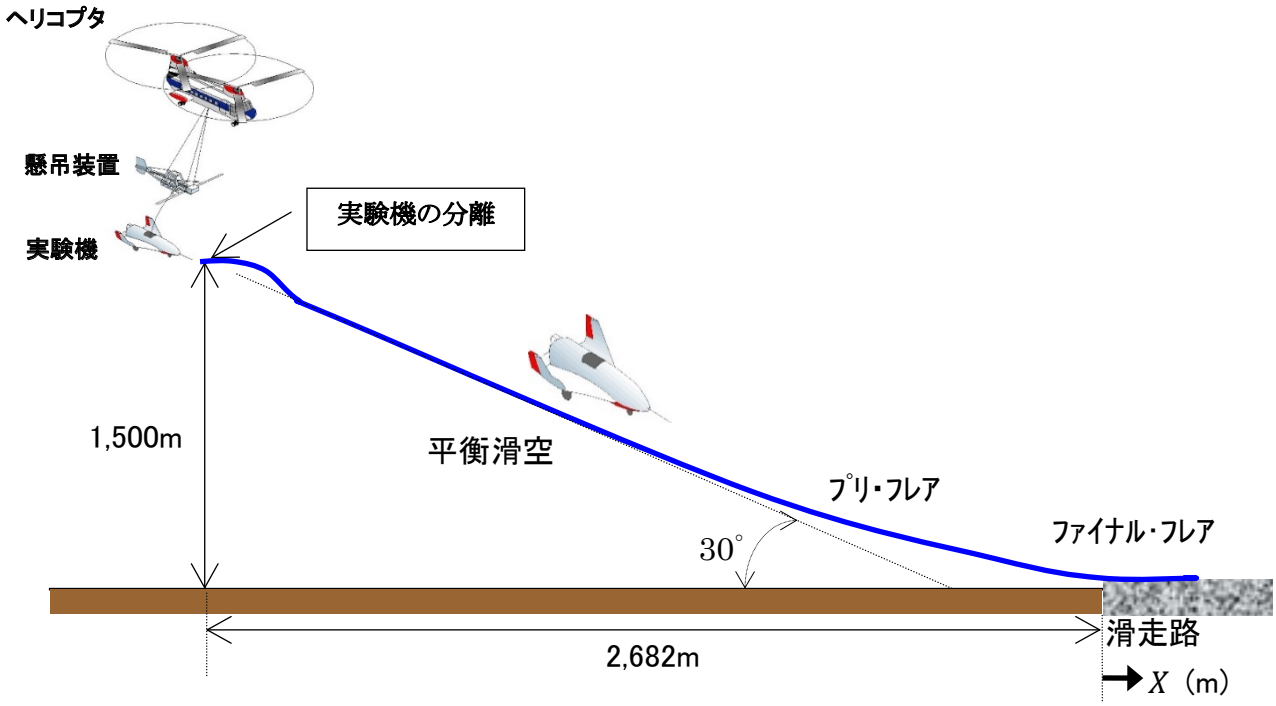


図 7　飛行実験プロファイル

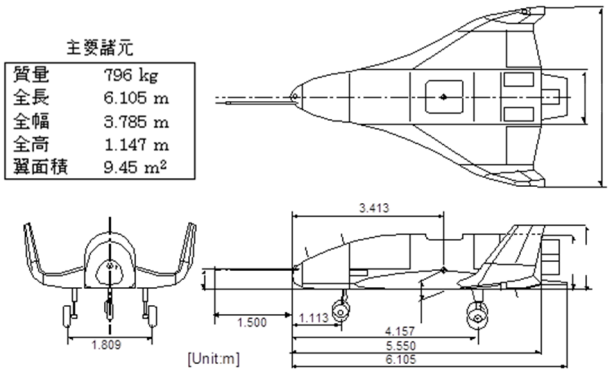


図 8 ALFLEX3面図と主要諸元

機体モデル誤差

- 慣性特性誤差
- 空力モデル誤差
- アクチュエータ・モデル誤差

センサ計測誤差

- 慣性センサ(IMU)
- 対気センサ(ADS)
- マイクロ波着陸システム(MLS)
- レーザ高度計(RA)

環境条件の変動

- 定常風（強さ&方向）
- 重力加速度
- 大気温度
- 大気圧

初期条件の変動

- 機体状態(位置, 速度, 姿勢角, 角速度)の変動
- 航法誤差

ランダム時系列変動

- 乱気流
- センサ出力(加速度)

図 9 不確定パラメータ

に自動着陸する. 分離後に実験機は引起しフェーズを経て, 動圧一定の安定した平衡滑空フェーズに入る．このときの飛行経路角は-30°である．そして滑走路にある程度近づいたところで, 引起しを始め経路角が 0°付近となるまで誘導される．これがプリフレア・フェーズである．着陸直前になると, 沈下率を主として制御するファイナルフレア・フェーズに移行して, 滑走路に軟着陸する．飛行実験前には MCS による評価も実施された．そのときに考慮された不確定パラメータは 100 以上であり，図 9 に示すカテゴリーの項目を含む．

5.2. 最適化の条件

本飛行実験の評価基準は, 飛行中に不安定とならないこと, および接地時の状態量が要求を満足すること，である．接地状態の評価基準を表 2 に示す. これらの評価基準を全て満足すれば“成功”, 1 項目でも要求を満足できなければ“失敗”と判定する．

これらの接地条件を満足するために重要となるのがプリフレア・フェーズである．この前段階の平衡滑空フェーズでは, 直線で表される基準軌道上を飛行させるために, フィードバック制御を主として, 様々な外乱や不確かさに対するロバスト性を十分に確保している．ところがプリフレア・フェーズでは曲線で表される基準軌道に追従させるため, フィードフォワード制御が主要な役割を果たす. フィードバック制御も含んでいるが, 平衡滑空フェーズに比べるとロバスト性の低下は避けられない. このフェーズで可能な限りロバスト性を高めることが, 成功確率の向上に大きく寄与する．ここでプリフレア・フェーズの縦の誘導則の構造を図 10 に示す．最適化により調整するパラメータは 4 つの誘導パラメータ，つまり K_F，K_P，K_I，K_Dである．

表 2 ALFLEX 評価基準

	接地時 パラメータ	許容範囲	単位
縦運動	X位置	$0 \le X \le 1000$	(m)
	沈下率$\dot{Z}$	$\dot{Z} \le 3.0$	(m/s)
	対地速度V_g	$V_g \le 62.0$	(m/s)
	ピッチ角Θ	$\Theta \le 23.0$	(deg)
横運動	Y位置	$\lvert Y \rvert \le 18.0$	(m)
	ロール角Φ	$\lvert \Phi \rvert \le 10.0$	(deg)
	対地横滑り角β_g	$\lvert \beta_g \rvert \le 8.0$	(deg)

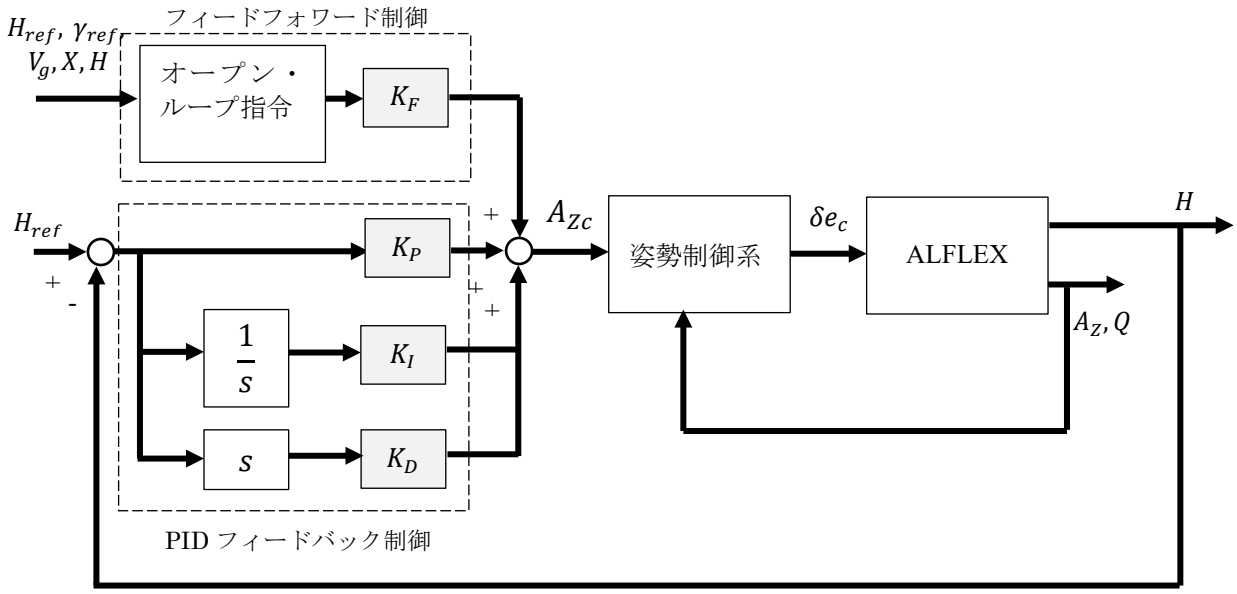


図 10 プリフレア・フェーズの縦誘導則

5.3. N更新ロジックの効果

本最適化計算の結果は，MCS の入力として用いるランダムサンプルの違いに影響を受ける. このため 4 つの調整パラメータの初期値は固定し，MCS のランダム入力を変更して，同じ最適化計算を 20 回繰返して実行し評価した．以下では計算負荷の変化を確認し, 次に最適化性能の変化との関わりについて考察する.

5.3.1. 計算負荷の比較

最適化計算が終了するまでの全シミュレーション回数を比較したものを図 11 に示す. Nの増加ロジック更新前後の結果を示した. 更新前の平均は 42.5 万回であるのに対して，ロジック更新によって 27.8 万回程度まで大幅に低下したことが確認できる. 今回のロジック更新により，平均的に 35%程度計算負荷が軽減されたことになる.

また図 12 には，最適化計算における探索ステップ数（MCS 評価した数）と，累積シミュレーション回数の履歴を示す. 青線がロジック更新前, 赤線が今回のロジック更新後の，それぞれ 20 回の最適化計算の結果である．ロジック更新前後で，計算終了までの探索ステップ数にはそれほど大きな違いはないが，縦軸の累積シミュレーション回数は異なることがわかる. このことは, ロジック更新によっても解の探索はそれまでと同等に実行されていることを示す．その一方で各設計ベクトルの MCS 評価回数が抑制されているため，最終的な累積シミュレーション回数N_{TOTAL}も軽減されている.

5.3.2. 最適化性能との関係

次に最適化性能との関係を確認する．最終的に求められた設計ベクトル（最適解）を，公平かつ精度良く評価するために，最適化計算に使用したものとは独立した乱数を用いて, それぞれ 10 万回の MCS 評価を実行した．この結果，各最適解に対応する MCS 評価による失敗確率が得られる．この失敗確率の信頼区間上限値P_Uと，最適化計算に要した全シミュレーション回数N_{TOTAL}を示したのが図 13 である. 横軸が全シミュレーション回数, 縦軸が失敗確率上限値であり，赤丸がロジック更新後の結果を示す.

既に確認したように横軸の全シミュレーション回数は，今回のロジック更新により平均的に 35%程度計算負荷が軽減されている. これに対して縦軸の失敗確率上限値（最適化性能）は，平均的にロジック更新前の 0.242%から，0.253%に若干悪化している. 悪化の割合はロジック更新前に対して 4.3%程度である.

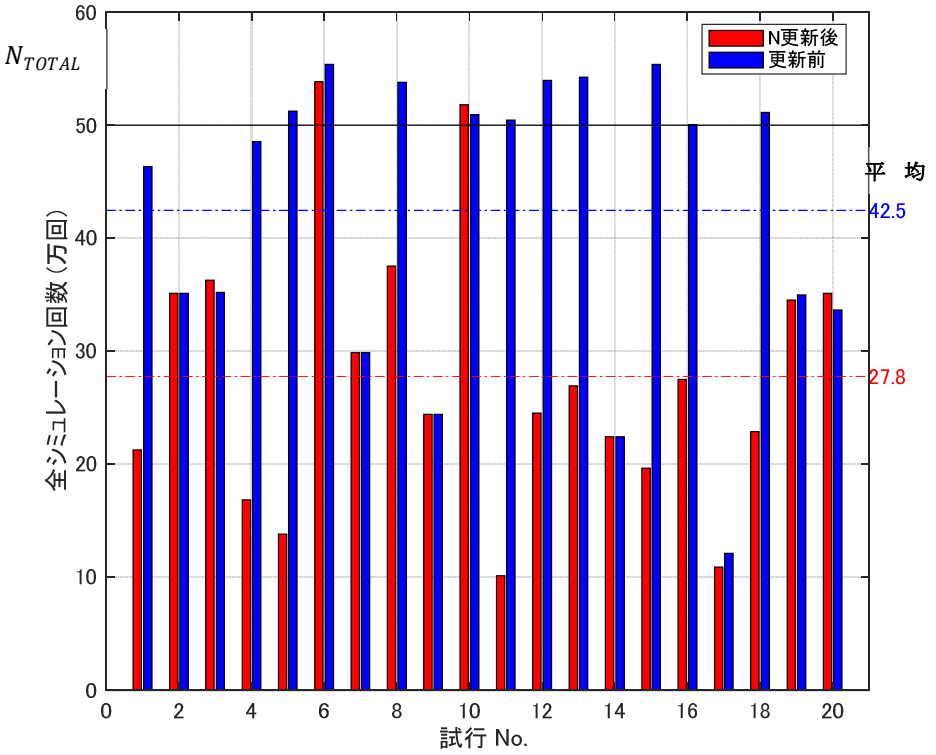


図 11 N更新前後の全シミュレーション回数の比較

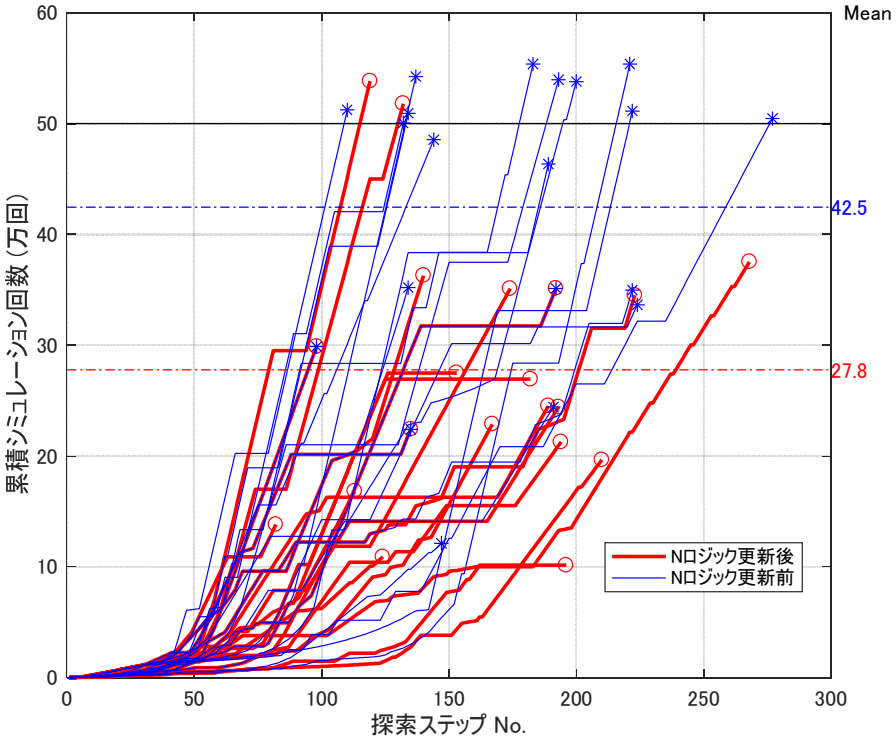


図 12 累積シミュレーション回数履歴の比較

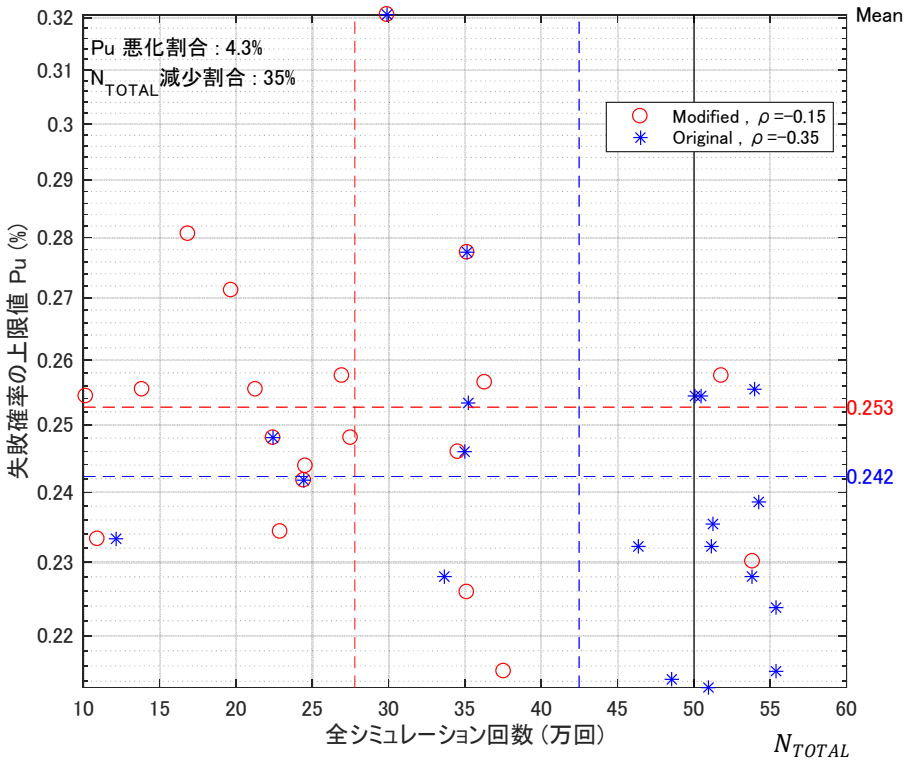


図 13　最適化性能と全シミュレーション回数の関係

一般に試行回数Nが大きいほど，MCS 結果の精度は高くなる．このためロジック更新によりNが抑制されると MCS の精度がそれだけ劣化するため，最適化計算の過程で発生させた設計ベクトルの評価関数も精度が悪化する．このことは，ある程度事前に予想できたものではある．今回の例ではロジック修正に伴う計算負荷の軽減 35%程度に対して，性能の劣化は 4.3%程度であった．

また参考情報として，相関係数ρを示した．N更新前後でいずれも負の相関となっており，N_{TOTAL}が増加するほどP_Uは改善する傾向となっている．但し，N更新前は-0.35 に対してN更新後は-0.15 であり，それほど強い相関ではない．

最適化結果は以上であるが，この結果は「計算負荷を軽減する代わりに，性能を一定程度犠牲にしなければならない」ということを意味するものではない．今回の修正は，計算過程で現れるその時点における最良値を，より合理的に評価するための改良を施した結果であり，意図しない急激なN増加を避けるロジックとしては理論的に妥当なものである．もし計算負荷に余裕があり，できるだけ最適化性能を追求したいという状況であれば，N更新の式(3)の調整パラメータp_0をより小さく調整することで，Nを増加させ，最適化性能の向上を狙うことは可能である．

5.3.3. p_0調整の効果

次にp_0の効果を確認する．これまでの計算例では，$p_0 = 0.01$を用いたが，これを$p_0 = 0.001$と小さくしてNを増加させた状態で，同じ最適化計算を実行した．この結果を図 14～図 16 に示す．これらは図 11～図 13 に対応する結果である．図 14 より，p_0を 1/10 とすることで全シミュレーション回数は平均的に 27.8 万回から 41.9 万回に大きく増加している．図 15 はNの増加履歴を比較したものであるが，探索ステップ数が増加しているわけではなく，各ステップにおけるNが増加する傾向がみられる．全シミュレーション回数と共に最適化性能を比較したものが図 16 である．

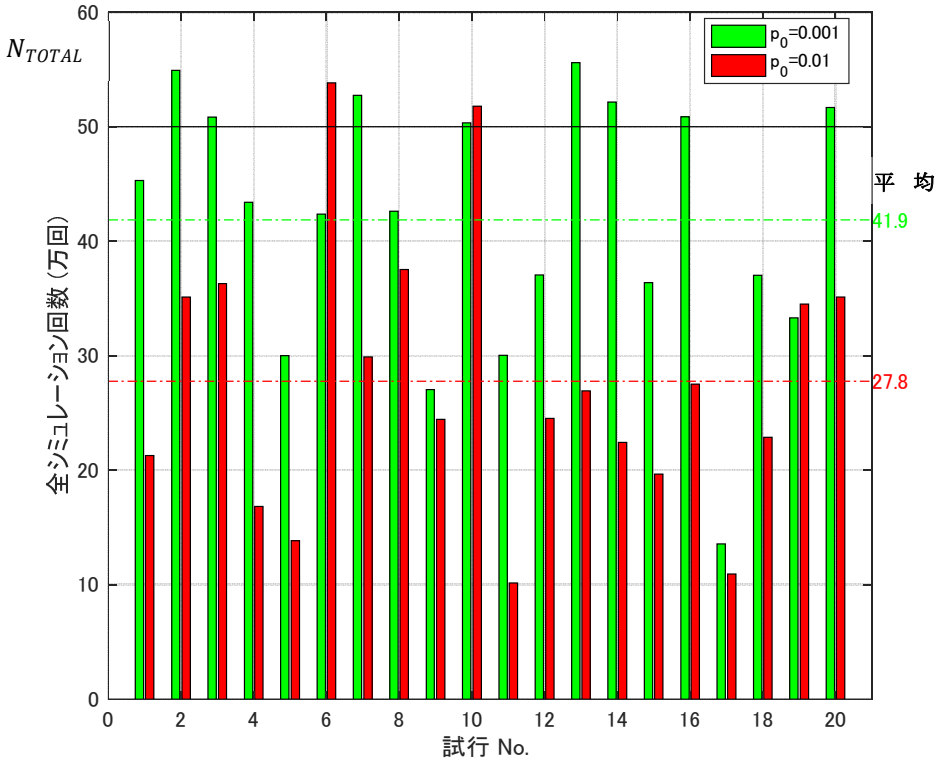


図 14 p_0変更前後の全シミュレーション回数の比較

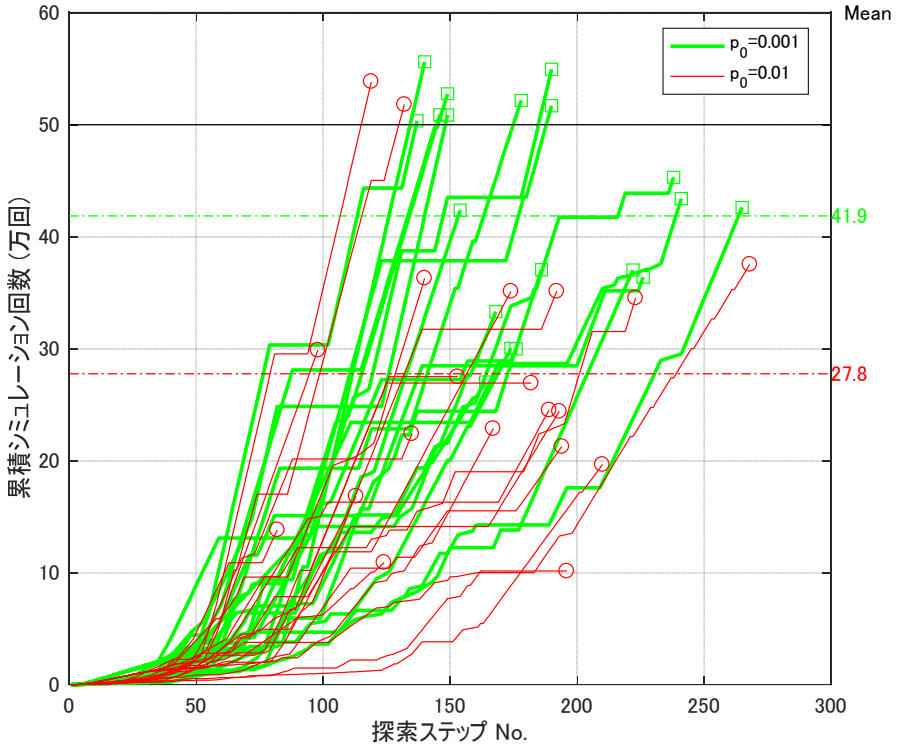


図 15 p_0変更による累積シミュレーション回数履歴の比較

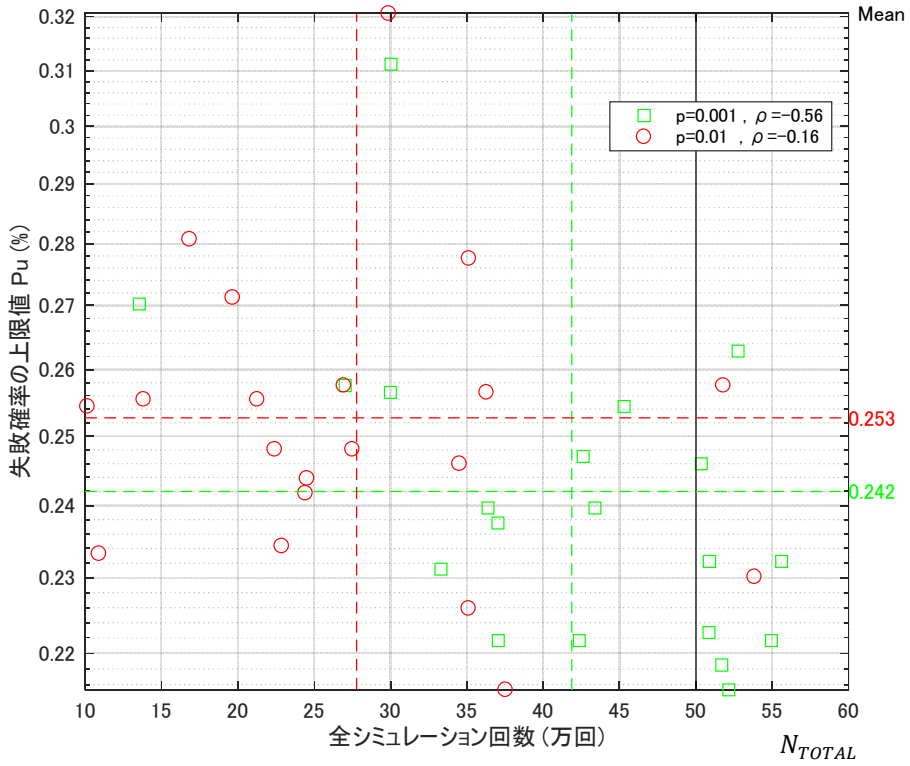


図 16 p_0変更による最適化性能と全シミュレーション回数

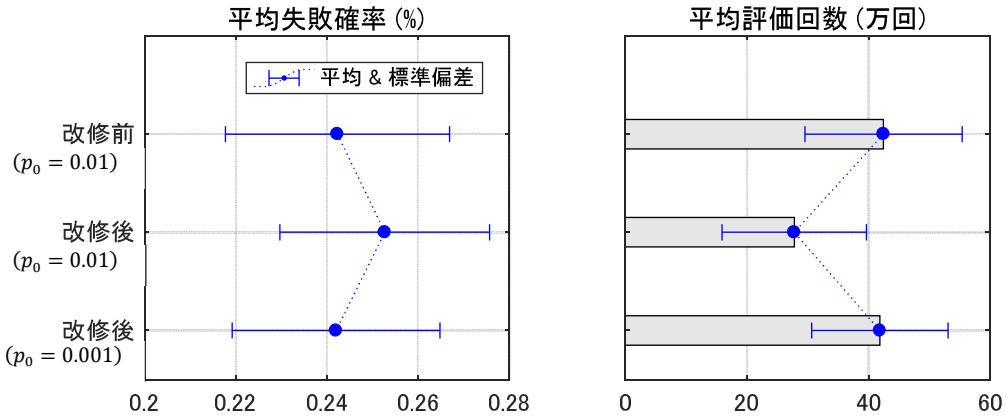


図 17 ロジック修正およびp_0変更による影響比較

p_0変更により失敗確率は平均的に，0.253(%)から 0.242(%)に改善している．またN_{TOTAL}とP_Uに負の相関がみられることは，図 13 とも同様の傾向である．

5.3.4. 試行結果のまとめ

今回実行した最適化計算の試行結果をまとめたものが図 17 である．N更新ロジックの改修前$(p_0 = 0.01)$と改修後$(p_0 = 0.01\ \&\ 0.001))$について，それぞれ入力サンプルを変更して 20 回の最適化計算を実行した．その結果得られた失敗確率と全評価回数の平均値と標準偏差を示している．今回の改修によりNの急激な増加が抑制されたことで，全シミュレーション回数N_{TOTAL}は抑えられ，意図したとおり計算負荷は低下した．一方で最適化性能はやや悪化する結果となった．次にN更新のパラメータp_0を調整してシミュレーション回数を増

加させた最適化計算を実行したところ，N_{TOTAL}は増加し，最適化性能は改善した．平均的にはN_{TOTAL}，最適化性能とも改修前の結果とほぼ同等となった．また標準偏差を比較すると，改修前と改修後($p_0 = 0.001$)では平均的な結果は同等だが，改修後のほうが失敗確率・評価回数共に，ばらつきはやや小さくなっている．なお今回の検証において改修前の結果と，改修後に$p_0 = 0.001$に変更した結果がほぼ同等となっているが，これは偶然であり結果が同等となるようにp_0を調整したものではない．

今回の結果から，計算負荷と最適化性能はある程度トレードオフの関係にありそうなことが確認できる．これはNを増加させれば，より精度の高い評価関数Jが算出され，最適化計算の過程で探索の方向がより正しくなることに起因するためと推測される．よって計算負荷（計算時間）の制約が許すのであれば，p_0を小さく設定することで，いくらかの最適化性能の向上は期待できる．

一般に失敗確率が改善される（小さくなる）ほど，そのシステムを評価するNは大きくする必要がある．今回例としたシステムでは，最終的に失敗確率が 0.25%程度まで改善された．この程度まで改善できるシステムを評価するためには，まだNを増加させれば性能改善の余地は多少ありそうなことは確認できた．一方で，計算負荷の軽減35%程度に対して，性能の劣化は3.4%であった．この例では，わずかな性能向上に対して，計算負荷の増大はかなり大きくなっている．わずかであっても性能向上を追求したいのか，又はより短時間で結果を出すことを優先したいのか，これは個々の設計対象と要求性能により事情は異なるため，それらを勘案して調整パラメータp_0を設定する必要がある．

6. まとめ

最適化計算における計算負荷，つまりシミュレーション評価回数，を抑制する目的で導入している評価回数Nの増加ロジックを修正した．これまでは最適化計算の過程でNが急激に増加する事象が発生することがあり，計算負荷抑制の観点からは好ましくない現象であったことから，確率的により合理的なロジックとして見直した．修正の効果は，MCS の入力となる乱数を変更しながら 20 回の最適化計算を繰り返すことにより確認した．この結果N増加ロジックの更新により，計算負荷がかなり抑制されることを確認できた．一方で最適化結果には，若干の悪化が見られた．これは大きなNを用いるほうが評価関数はより正確になることに起因するものであり，最適化性能と計算負荷はある程度トレードオフの関係になっているためと推測される．但し，修正後のロジックにおいても必ずしも最適化性能を犠牲にしなければならないわけではなく，別の調整パラメータを用いることで，より大きなNを算出させることは可能であり，その性能改善の可能性も示した．

計算負荷と最適化性能をそれぞれどの程度優先させるかは，最適化計算を利用する場面での時間と要求性能を考慮して決定する必要がある．実際の設計現場においては時間的な制約が厳しいことも多く，今回の修正により意図しない計算負荷の増大が回避できるようになったことは，実用性の観点から望ましいと考える．本最適化法は MCS 結果を直接最適化するものであり，非線形システムのロバスト性を直接高めることができる．本最適化法が航空宇宙システムの設計・開発に寄与できれば幸甚である．

参考文献

1) Marrison, C. I. and Stengel, R. F., “Design of Robust Control Systems for Hypersonic Aircraft,” *Journal of Guidance, Control and Dynamics*, Vol. 21, No. 1, 1998, pp. 58-63, https://doi.org/10.2514/2.4197.

2) Miyazawa Y. and Motoda T., “Stochastic Parameter Tuning Applied to Space Vehicle Flight Control Design,” *Journal of Guidance, Control,*

and Dynamics, Vol.24, No.3, 2001, pp.597-604, https://doi.org/10.2514/2.4751.

3) Motoda, T., Stengel, R. F. and Miyazawa, Y. “Robust Control System Design Using Simulated Annealing,” *Journal of Guidance, Control and Dynamics*., 25, 2 (2002), pp. 267-274, https://doi.org/10.2514/2.4878.
4) Press, W. H., Teukolsky, S. A., Vetterling, W. T. and Flannery, B. P., *Numerical Recipes in C*, 2nd ed., Cambridge University Press, New York, 1993, pp. 408-412.
5) Aarts, E. and Korst, J., *Simulated Annealing and Boltzmann Machines*, John Wiley & Son Ltd., New York, 1989, pp. 13-31.
6) Schubert, W. M., and Stengel, R. F., “Parallel Synthesis of Robust Control Systems”, *IEEE Transactions on Control Systems Technology*, Vol. 6, No. 6, 1998, pp. 701-706, https://doi.org/10.1109/87.726531.
7) 元田敏和，中川英治，渡邉篤，「MCS 並列計算システムの開発」，特集 MCS 並列計算システムの開発と応用 第 1 回，日本航空宇宙学会誌，第 67 巻，第 5 号，2019 年，pp. 167−173.
8) 元田敏和，「独自機能（２）：MCS を利用した設計パラメータのロバスト最適化」，特集 MCS 並列計算システムの開発と応用 第 5 回，日本航空宇宙学会誌，第 67 巻，第 11 号，2019 年，pp. 375−381.
9) 元田敏和，「モンテカルロ評価を利用した設計パラメータのロバスト最適化」，JAXA-RR-22-002, 2022 年 8 月 . http://doi.org/10.20637/00048696.
10) Conover, W. J., *Practical Nonparametric Statistics*, Third edition, John Wiley & Sons, Inc., New York, 1999, pp.131-133, and 188-190. ISBN 9780471160687.
11) 東京大学教養部統計学教室編，「統計学入門」，東京大学出版会，第 12 刷，1997, pp.111-113.
12) 元田敏和，「モンテカルロ法によるシステムの統計的評価」, JAXA-RR-07-005, 2007 年 12 月 . https://jaxa.repo.nii.ac.jp/records/2176.
13) Motoda, T., Miyazawa, Y., Ishikawa, K. and Izumi, T., “Automatic Landing Flight Experiment Flight Simulation Analysis and Flight Testing,” *Journal of Spacecraft and Rockets*, Vol. 36, No. 4, 1999, pp. 554-560. https://doi.org/10.2514/3.27199.
14) Miyazawa, Y., Nagayasu, M., and Nakayasu, H., “Flight Testing of ALFLEX Guidance, Navigation and Control System,” International Council of the Aeronautical Sciences, Paper 98-1.1.3, Sept. 1998.
15) Miyazawa Y., Motoda T., Izumi, T. and Hata, T., “Longitudinal Landing Control Law for an Autonomous reentry vehicle,” *Journal of Guidance, Control, and Dynamics*, Vol. 22, No. 6, 1999, pp. 791-800. https://doi.org/10.2514/2.4480.
16) Yanagihara, M., Shigemi, M. and Suito, T., “Estimating aerodynamic characteristics of automatic landing flight experiment vehicle using flight data,” *Journal of Aircraft*, Vol. 36, No. 6, 1999, pp. 926-933. https://doi.org/10.2514/2.2553.

宇宙航空研究開発機構研究開発資料　JAXA-RM-23-003
JAXA Research and Development Memorandum

統計的最適化におけるモンテカルロ・シミュレーション評価回数更新法の見直し
Modification for Updating Number of Monte Carlo Simulations in Stochastic Parameter Optimization

発　　行　　国立研究開発法人宇宙航空研究開発機構（JAXA）
〒182-8522　東京都調布市深大寺東町7-44-1
URL: https://www.jaxa.jp/
発　行　日　　2024年1月19日
電子出版制作　　松枝印刷株式会社