

第 3 章

脚配置計画アルゴリズム

3.1 はじめに

本章では，前章で定式化した問題を解くためのアルゴリズムについて説明する．反復法を用いた計算アルゴリズムによって計算を行い，脚配置を計算する．即ち，定式化した二次計画問題式 (2.6) を逐次解き，入力列 U_k を更新して解に近づけていくという計算を行う．

全体の計画のアルゴリズムを Fig. 3.1 に示す．各要素について述べていく．

3.2 初期ステップ数決定と入力列初期化

まずはじめに，計画開始時に用いる初期ステップ数 k_{init} を設定する．平面上と高さ方向それぞれについて，目標位置までの距離を 1 ステップあたりの最大の直進距離 $d_{xy \max}$ ， $d_{z \max} = 2 z_{\max}$ で割ることで所要ステップ数を求め（平面上の最大直進距離は，事前に直線上を進む計画問題を解いておき，得られた解の直進距離を用いる），二つのうち大きい方を採用する．さらに，これに実用上 1 を加えたものを k_{init} として設定する．これは，先の計算で求めたものはあくまでも直進で目標に向かうことができる場合の最小値であり，地形や初期・目標状態の姿勢を考慮すると実際には求まったステップ数では到達できないと考えられ

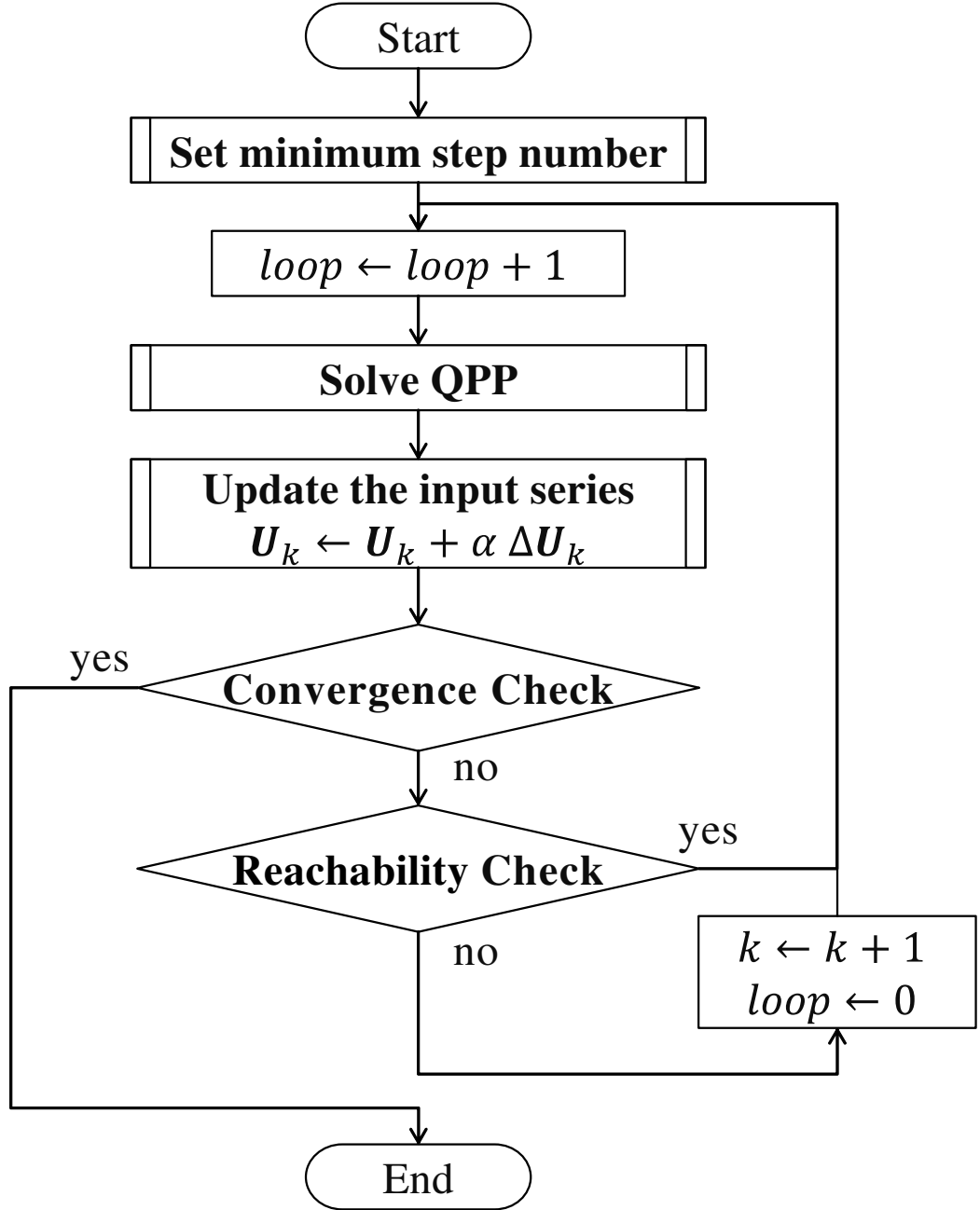


Fig. 3.1 Footstep planning algorithm

るからである.

$$\begin{aligned}
 k_{\text{init}} &= \lceil \max(k_{xy\text{max}}, k_{z\text{max}}) \rceil + 1 \\
 k_{xy\text{max}} &= \frac{\|(x_{\text{goal}}, y_{\text{goal}}) - (x^0, y^0)\|}{d_{xy\text{max}}} \\
 k_{z\text{max}} &= \max\left(\frac{\|z_{l\text{goal}} - z_l^0\|}{d_{z\text{max}}}, \frac{\|z_{r\text{goal}} - z_r^0\|}{d_{z\text{max}}}\right)
 \end{aligned} \tag{3.1}$$

また、入力列は零ベクトルで初期化する。

3.3 二次計画問題の求解と入力列更新

3.3.1 Goldfarb-Idnani 法による二次計画問題の求解

“Solve QPP” では、各反復における計画問題式 (2.9) を解き ΔU_k を求める。解法としては、式 (2.9) が凸二次計画問題であることから、有効制約法的一种である Goldfarb-Idnani (GI) 法 [18] を用いる。

線形近似により実行不可能な問題となってしまった場合は、若干の制約違反を許す緩和問題を設定して解く。文献 [19] で示されている無限大ノルムを用いた緩和問題式 (3.2) を設定する。

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2} \Delta U_k^T \Phi \Delta U_k + \phi^T \Delta U_k + \rho \xi \\ \text{subject to} \quad & \Omega \Delta U_k + \omega + \xi \mathbf{1} \geq \mathbf{0}, \quad \xi \geq 0 \end{aligned} \quad (3.2)$$

$\rho > 0$ はペナルティパラメータ、 ξ はスラック変数、 $\mathbf{1}$ は全要素が1のベクトルである。この問題はすでに二次計画問題ではなくなっているため、式 (3.3) および式 (3.3) に示すように二次計画問題へ近似する。

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2} \Delta U_k^T \Phi' \Delta U_k + \phi'^T \Delta U_k \\ \text{subject to} \quad & \Omega' \Delta U_k + \omega' \geq \mathbf{0} \end{aligned} \quad (3.3)$$

$$\begin{aligned} \Phi' &= \begin{pmatrix} \Phi & \mathbf{0} \\ \mathbf{0} & \varepsilon_{\text{relax}} \end{pmatrix}, & \phi' &= \begin{pmatrix} \phi \\ \rho \end{pmatrix} \\ \Omega' &= \begin{pmatrix} \Omega & \mathbf{1} \\ \mathbf{0} & 1 \end{pmatrix}, & \omega' &= \begin{pmatrix} \omega \\ 0 \end{pmatrix} \end{aligned} \quad (3.4)$$

$\varepsilon_{\text{relax}}$ は十分に小さい定数とする。

3.3.2 数値計算安定化のための各種パラメータ設定方法

問題で用いる係数行列およびベクトルを次のように設定することで、反復計算の安定性および収束性能を向上させる。

- K_p , K_u , K_{u_z} の各成分を、係るベクトルの各要素をそれぞれ最大値で正規化するように設定する

$$\begin{aligned}
 K_p &= \text{diag} \{k_x, k_y, k_\theta, k_l, k_{\theta_{fl}}, k_{\theta_{fr}}\} \\
 k_x &= k_y = (\max\{|x_{\text{goal}} - x^0|, |y_{\text{goal}} - y^0|\})^{-2} \\
 k_\theta &= (2\pi)^{-2} \\
 k_l &= (Y_{\max} - Y_{\min})^{-2} \\
 k_{\theta_{fl}} &= k_{\theta_{fr}} = (2\pi)^{-2}
 \end{aligned} \tag{3.5}$$

$$\begin{aligned}
 K_u &= \text{diag} \{\dots, k_{u_A}, k_{u_B}, k_{u_{l1}}, k_{u_{l2}}, k_{u_{fl}}, k_{u_{fr}}, \dots\} \\
 k_{u_A} &= k_{u_B} = (2 \tan^{-1}(X_{\max}/Y_{\min}) + \theta_{2\max})^{-2} \\
 k_{u_{l1}} &= k_{u_{l2}} = (Y_{\max} - Y_{\min})^{-2} \\
 k_{u_{fl}} &= k_{u_{fr}} = (\theta_{2\max} - \theta_{2\min})^{-2}
 \end{aligned} \tag{3.6}$$

$$\begin{aligned}
 K_{u_z} &= \text{diag} \{\dots, k_{u_{zl}}, k_{u_{zr}}, \dots\} \\
 k_{u_{zl}} &= k_{u_{zr}} = z_{\max}^{-2}
 \end{aligned} \tag{3.7}$$

- K_u , K_{u_z} は、前項の設定に加えて Sugihara の手法 [20] を適用し、評価関数式 (2.7) の値を用いて重み付けする (I は単位行列, Δk_u はスカラー定数)

$$\begin{aligned}
 K_u &\leftarrow V_{\text{now}} K_u + \Delta k_u I = \frac{1}{2} \|\epsilon_{\text{now}}\|_{K_p}^2 K_u + \Delta k_u I \\
 K_{u_z} &\leftarrow V_{\text{now}} K_{u_z} + \Delta k_{u_z} I = \frac{1}{2} \|\epsilon_{\text{now}}\|_{K_p}^2 K_{u_z} + \Delta k_{u_z} I
 \end{aligned} \tag{3.8}$$

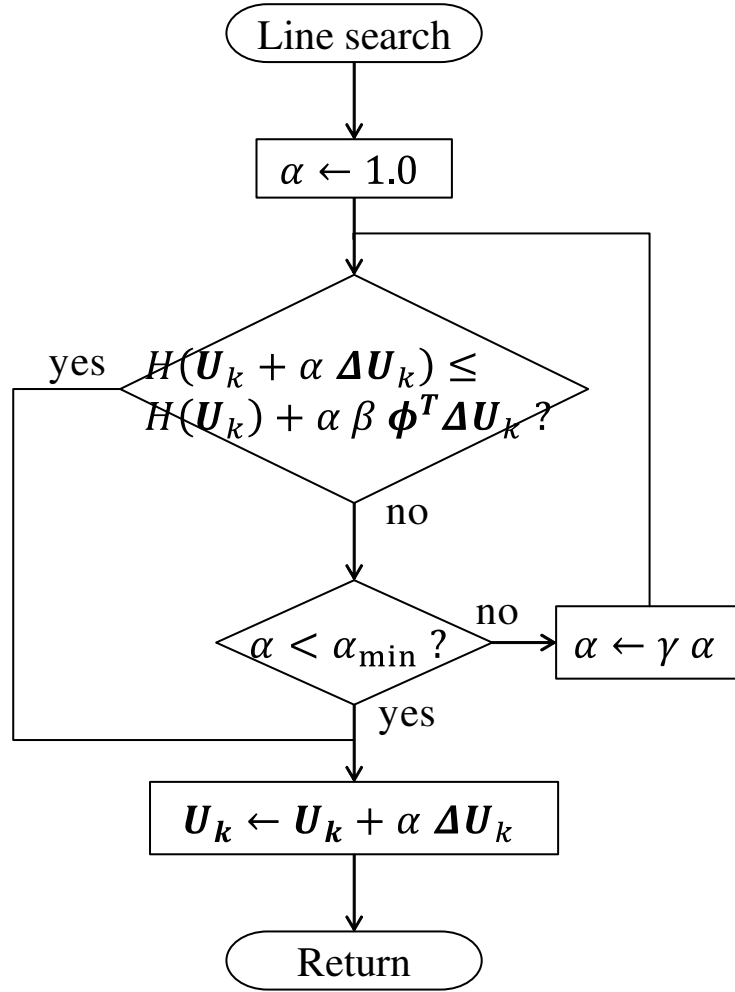


Fig. 3.2 Line search

- 一つ一つの制約条件 $\omega_i(\mathbf{q}^0, \mathbf{U}_k) \geq 0$ を $\|\nabla \omega_i\|$ で正規化することにより，GI 法内部での反復計算を安定化させ，また緩和問題での各制約条件の比重を揃える [21]
- 緩和問題を解く際，評価関数全体を $\|\boldsymbol{\phi}\|$ で正規化し，元の評価関数部分の値とスラック変数の値の比重を揃える ($\rho = 1$ とできる)

そして，得られた解を用いて Armijo の基準に基づく直線探索を行い，入力列を更新する．

3.3.3 直線探索

直線探索では，次式を満たすステップ幅 α を求める．

$$\begin{aligned} H(\mathbf{q}^0, \mathbf{U}_k + \alpha \Delta \mathbf{U}_k) &\leq H(\mathbf{q}^0, \mathbf{U}_k) + \beta \alpha \nabla V(\mathbf{q}^0, \mathbf{U}_k)^T \Delta \mathbf{U}_k \\ &= H(\mathbf{q}^0, \mathbf{U}_k) + \beta \alpha \boldsymbol{\phi}^T \Delta \mathbf{U}_k \end{aligned} \quad (3.9)$$

β は， $0 < \beta < 1$ を満たす定数である．今回， α は Fig. 3.2 に示す単純な探索により求める．
なお， γ は $0 < \gamma < 1$ を満たす減衰係数である．

3.4 終了判定

更新後，終了判定を行う．次の項目をすべて満たすことを終了条件とする．

- $\boldsymbol{\varepsilon}$ が収束（各要素が許容誤差 $\boldsymbol{\varepsilon}_{\text{goal}}$ の各要素よりも小さい）
- $\Delta \mathbf{U}_k$ が収束（各要素が許容誤差 $\boldsymbol{\varepsilon}_{\text{input}}$ の各要素よりも小さい）
- $\omega_i(\mathbf{q}^0, \mathbf{U}_k) \geq 0$ ($i = 0, \dots, 20k - 1$) をすべて満たす

これらの条件を全て満たしていれば，その時点での \mathbf{U}_k を解とし，計画を終了する．満たしていない場合は次節の処理へ移る．

3.5 目標到達可否判定

初期ステップ数 k_{init} は直線距離から算出するため，現在の k では歩数不足の場合がある．そこで，反復毎に目標状態へ収束する見込みがあるかどうかを判定し，見込みがない場合に k を増やす処理を行う．ここでは，次の項目のいずれかを満たせば到達不可と判定することとする．

- “ $\Delta \mathbf{U}_k$ 収束時に $\boldsymbol{\varepsilon}$ が未収束” という状況が $n_{\text{reachable1}}$ 回繰り返される

- 現在の k における反復回数 loop が $n_{\text{reachable2}}$ に達した段階で、 ϵ が大きい値をとる
(各要素が到達見込みの閾値 $\epsilon_{\text{reachable}}$ の各要素よりも大きい)

到達不可と判定された場合には $k \leftarrow k + 1$ とし、また \mathbf{U}_k はその内容を引き継ぎ $\mathbf{U}_k \leftarrow (\mathbf{U}_k^T, \mathbf{0}^T)^T$ として再度反復を開始する．そうでなければ、“Solve QPP” へ戻り反復を継続する．

3.6 地形情報の構築

前述の計画アルゴリズムを開始する前に、地形適応制約で用いる地形情報 $z_{\text{field}}(x, y)$ を構築する必要がある．深度カメラや測域センサから得られる点群データを用いて地形情報を構築することを想定し、次の手順で点群データからの構築を行った．

1. xy 平面を等間隔に幅 D_{grid} 四方の正方形グリッドで分割する．
2. 各グリッドの領域に含まれるサンプルデータ群の z 軸成分の平均値を、グリッド中央における高さとする．
3. x, y 方向それぞれについて、グリッド中心の点列をサンプルとする 3 次スプライン補間関数を計算．
4. $z_{\text{field}}(x, y), \partial z_{\text{field}}/\partial x, \partial z_{\text{field}}/\partial y$ は、与えられた点 (x, y) に対して直近の x 軸方向 2 本、 y 軸方向 2 本の補間関数上の値の距離重み付け平均で算出する．

なお、シミュレーションにおいて関数からサンプリングを行って点群データを得る場合、 D_{sample} 間隔でサンプリングを行う．

3.7 本章のまとめ

本章では、二次計画法を用いた反復法による脚配置計画問題の求解アルゴリズムの要素として、必要歩数の推定から毎回の二次計画問題の求解と緩和問題の設定、収束判定、収束見込み判定とステップ数追加について説明した．また、地形の違いなどにより問題ごとに大き

く変化してしまうことを可能な限り回避するため，反復法における数値計算の性能を安定させる重み行列の設定方法などについても述べた．

次章以降は，提案する脚配置計画手法の有用性および性能を検証するための計画シミュレーションおよび実機実験について述べていく．