三次元地形上における
二足歩行ロボットの
モデルベース脚配置計画

平成 27 年度

大阪市立大学大学院工学研究科
電子情報系専攻　前期博士課程

小　林　　大　気

概要

- My hovercraft is full of eels.
- A légpárnás hajóm tele van angolnákkal.
- 私のホバークラフトは鰻でいっぱいです。
- 小なり <; 等号 =; 大なり >;

二足歩行ロボットの自律移動には，目的地までの経路選択や軌道計画のみならず，局所的範囲での機構制約や地形に適応した，実時間での脚配置計画が重要となる．脚配置計画手法としてよく研究されている固定パターンの組み合わせ問題としてグラフ探索手法で解く探索ベース手法では，地形適応などが容易に行えるが，厳密な最適化はできず，探索における計算コストも高い．一方で，脚先遷移を運動学モデルで表現して逆運動学問題として解くモデルベースの手法も提案されており，原理的により厳密に最適化でき計算コストも低く抑えられるため，局所的な脚配置計画としてはより適していると考えられる．平面上では実時間で計画可能なモデルベース手法が既に提案されているが，三次元地形上での計画は実現されていなかった．

そこで，従来の水平方向移動のみを考慮した歩行モデルに独立した垂直方向の移動動作を加え，地形の形状情報に基づき，遊脚の接地点を支持脚基準の着地ワークスペース範囲内へ制限する制約を定式化し加えることで，実時間性を維持しつつ，三次元地形に適応可能なモデルベースの脚配置計画手法を提案した．二足歩行ロボット HRP-2 および NAO を想定した計画シミュレーションにより，段差など不連続な変化のある地形形状では最適化計算の不安定性に関して改善の余地があるが，連続的な形状の地形では安定して最適化された脚配置

が得られており，またいずれにおいても実時間で計画できていることを示した.

目次

図目次

表目次

第 1 章

緒言

1.1 研究背景

カメラ画像を利用して人物を追従する様々な研究が行われており，監視・警備などのセキュリティや災害などの救助活動，スポーツ戦略分析などの応用が期待される．西川らは，グラフ最適化アルゴリズムに基づく複数のカメラを使用した多人数トラッキングシステムを開発し，人物の追跡制度や計算時間を評価している [**?**, **?**].

その中でも，カメラ自身が撮影対象に応じて移動することによって，一定の画角に縛られることなく，オクルージョン等で撮影対象を見失う可能性が減少し，人物の追従精度の向上が考えられる．主に移動カメラのプラットフォームとして利用されていたものとして，カメラを搭載した UAV が挙げられる．Teuliere らは，画像の色のヒストグラムの類似度を利用して，地上の撮影対象を 1 台の UAV で追従し，撮影対象がオクルージョンにより見えなくなった場合でも，パーティクルフィルタを用いて撮影対象を見失わないようにしている [**?**]. Bethke らは，地上の撮影対象に対して複数の UAV を使用して追従し，それぞれの UAV の位置情報と UAV が撮影した画像から，撮影対象の正確な位置と速度を推定している [**?**]. Naseer らは，2 台のカメラを取り付けた UAV を使用して人物の追従をしており，正面を向けたカメラで人物とジェスチャーの認識，天井に向けたカメラで AR マーカによる UAV の

位置推定を行っている [**?**]．Price らは，MAV に搭載された PC 上でリアルタイムで動作し，かつ信頼性がある人物認識のためのニューラルネットワークを実現し，そのニューラルネットワークの認識結果を利用し，複数の MAV で人物追従を行っている [**?**]．Wang らは，複数の UAV を用いたビジョンベースによる捜索・救助システムを提案しており，UAV 間の通信を維持しながら捜索領域をより大きくするための UAV の経路を計画している [**?**].

カメラ以外を利用した人物追従に関する研究も存在する，Bajracharya らは，地上を移動するロボットにレーザ測域センサ（LiDAR）を搭載し，LiDAR から取得した点群情報から歩行者の認識・追従を行う [**?**]．また古川らは，海上で遭難した人物を UAV を利用して救助活動を行うことを想定し，再帰的ベイズ推定で対象人物の移動の推定を行い，推定された位置に UAV を移動させることによって人物の追従を行っている [**?**, **?**]．複数の移動物体を複数の UAV で追従する研究に関しては，Pack らが，追従対象である移動物体が断続的に信号を発信し続けて，その取得した信号に基づいて複数の UAV を制御している [**?**].

そこで我々は，Fig. 1.1 のように，地上に設置された LiDAR と UAV に搭載されたカメラを協調させることによって，地上で活動している複数人物をモニタリングするシステムを提案している．提案するシステムでは，LiDAR で得られる点群情報から人物の位置検出を行い，UAV に搭載されたカメラ（移動カメラ）から得られる画像データを解析することで，人物の着用するビブスの背番号から個人の特定を行う．さらに，LiDAR からはオクルージョンや自由に撮影位置を変更ができないといった理由により追従することのできない人物を，移動カメラの画像情報から補完し，正確な移動軌跡の導出を行う．

そこで我々は，Fig. 1.1 のように，地上に設置された LiDAR と UAV に搭載されたカメラを協調させることによって，地上で活動している複数人物をモニタリングするシステムを提案している．提案するシステムでは，LiDAR で得られる点群情報から人物の位置検出を行い，UAV に搭載されたカメラ（移動カメラ）から得られる画像データを解析することで，人物の着用するビブスの背番号から個人の特定を行う．さらに，LiDAR からはオクルージョンや自由に撮影位置を変更ができないといった理由により追従することのできない人物を，移動カメラの画像情報から補完し，正確な移動軌跡の導出を行う．

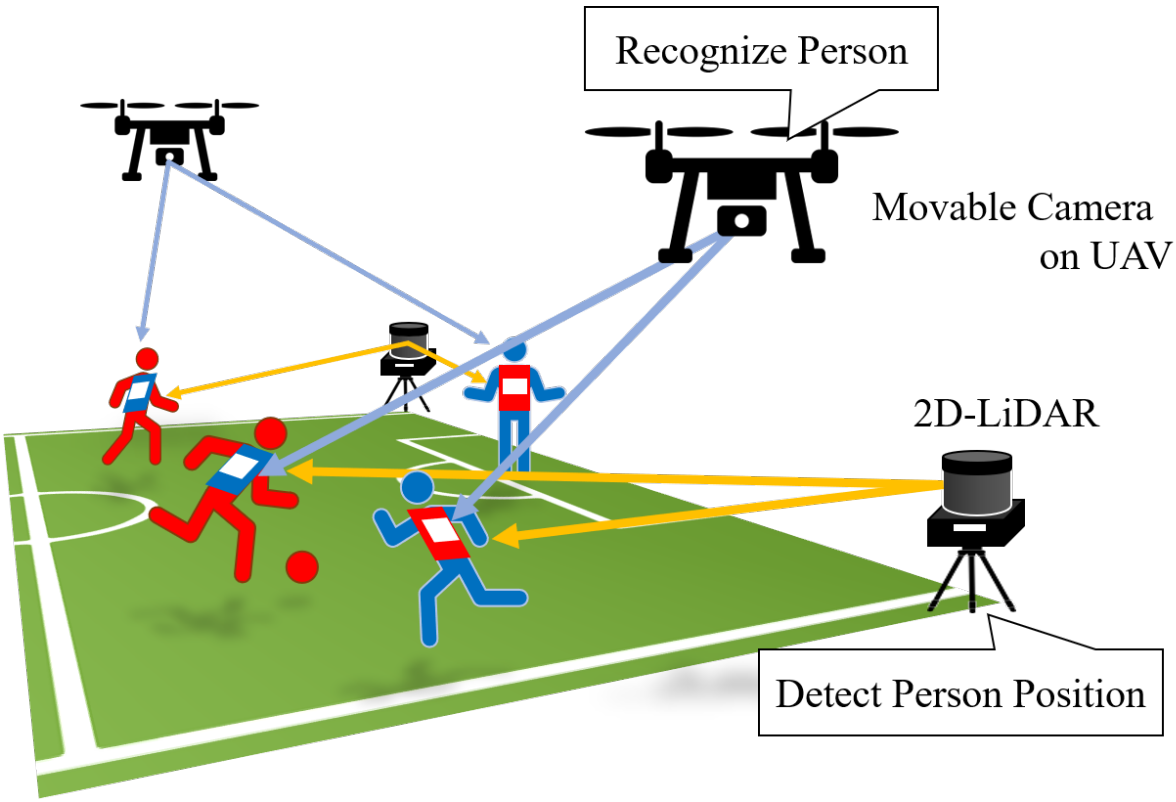

Fig. 1.1　Proposed Monitoring System

1.2　本論文の構成

本論文の構成は，次のようになっている．まず本章において，本研究の背景・目的などを述べた．2 章では，昨年度まで行っていた先行研究についての概要とその問題点，課題点などについて述べる．3 章では，LiDAR の位置と角度の校正手法について述べる．4 章では，屋外での実機実験の実装とその実験の結果について述べる．最後に 5 章では，本研究の結果などについてまとめ，今後の課題と展望について述べる．

第 2 章

地上固定 LiDAR と複数 UAV を用いた撮影計画の概要

2.1 はじめに

本章では，昨年度まで佐々木らが行っていた地上固定 LiDAR と複数 UAV を用いた撮影計画の概要について述べる．[**?**] またその研究における問題点，課題点などについて述べる．

2.2 先行研究における実装

このモデルの状態変数は $\boldsymbol{q}^i = (x^i, y^i, \theta^i, l^i, \theta^i_{fl}, \theta^i_{fr})$, 入力変数は $\boldsymbol{u}^i = (u^i_A, u^i_B, u^i_{l1}, u^i_{l2}, u^i_{fl}, u^i_{fr})$ と定義される．また，状態方程式**??**は $\boldsymbol{q}^i$，$\boldsymbol{u}^i$ を用いて式 (2.1) で表される．

$$\boldsymbol{q}^{i+1} = \boldsymbol{G}\left(\boldsymbol{q}^i, \boldsymbol{u}^i\right) \tag{2.1}$$

2.3　脚先の高さの取り扱い

前節で述べた脚配置モデル**??**は平面上の遷移についてのものであり，$\boldsymbol{q}^i$，$\boldsymbol{u}^i$ は要素に脚先高さに関するものを含まない．これは，提案する運動学モデルが，脚の垂直方向の移動を平面上の動作と独立して行えるという仮定のもとで構築されていることに由来する．地面への接地を考えると，脚先の z 座標は xy 平面上の位置における地形形状に合わせることになる．左右の脚先の xy 平面上の座標それぞれを $(x_l^i,\, y_l^i) = (x^i,\, y^i)$，$(x_r^i,\, y_r^i) = (x^i + l^i \cos\theta^i,\, y^i + l^i \sin\theta^i)$ とすれば，z_l^i，z_r^i は**??**に示すように式 (2.2) で表すことができる．

$$\begin{aligned} z_l^i &= z_{\text{field}}\left(x_l^i,\, y_l^i\right) \\ z_r^i &= z_{\text{field}}\left(x_r^i,\, y_r^i\right) \end{aligned} \tag{2.2}$$

地形形状 $z_{\text{field}}(x,\, y)$ が既知であれば，z_l^i，z_r^i は式 (2.2) を用いて $\boldsymbol{q}^i$ で表現できる．これを用いて地形との接触制約を課すことにより，平面上の脚配置モデルでありながら地形を考慮した三次元の脚配置計画を行うことが可能となる．具体的な制約式については 2.6 節で示す．

2.4　計画問題の定式化と解法

??より，k ステップ後の状態 $\boldsymbol{q}^k$ は初期状態 $\boldsymbol{q}^0$ と入力列 $\boldsymbol{U}_k = (\boldsymbol{u}^{0^T}, \ldots, \boldsymbol{u}^{k-1^T})^T$ を用いて $\boldsymbol{q}^k = \boldsymbol{G}_k(\boldsymbol{q}^0,\, \boldsymbol{U}_k)$ と表すことができる．目標状態 $\boldsymbol{q}_{\text{goal}}$ へ k ステップで到達するとしたときの脚配置計画問題は，式 (2.3) の非線形代数方程式の解 $\boldsymbol{U}_k$ を求める逆運動学問題となる．

$$\boldsymbol{q}_{\text{goal}} = \boldsymbol{G}_k\left(\boldsymbol{q}^0,\, \boldsymbol{U}_k\right) \tag{2.3}$$

この問題式 (2.3) を $\boldsymbol{U}_k$ について解けば脚配置計画が求まる．

本論文では，目標追従を評価関数で表し，制約条件下での最適化問題として解くことを考える．$\boldsymbol{U}_k$，$\boldsymbol{q}^0$ を変数とする評価関数を $V = H(\boldsymbol{q}^0,\, \boldsymbol{U}_k)$，制約条件を $\boldsymbol{\omega}(\boldsymbol{q}^0,\, \boldsymbol{U}_k) \geq \boldsymbol{0}$ とし

て，式 (2.4) の最適化問題を定義する．

$$\begin{aligned} &\text{minimize} \quad V = H\left(\boldsymbol{q}^0, \boldsymbol{U}_k\right) \\ &\text{subject to} \quad \boldsymbol{\omega}\left(\boldsymbol{q}^0, \boldsymbol{U}_k\right) \geq \boldsymbol{0} \end{aligned} \tag{2.4}$$

この式 (2.4) の非線形計画問題を，反復法を用いて解くことを考える．反復のある段階での入力列 $\boldsymbol{U}_k$ に微小変化 $\boldsymbol{\Delta U}_k$ を加えて $\boldsymbol{U}_k \leftarrow \boldsymbol{U}_k + \boldsymbol{\Delta U}_k$ として更新するとき，更新前の評価値 V_{now} は，更新後の評価値 V_{next} と変化量 ΔV を用いて $V_{\text{next}} = V_{\text{now}} + \Delta V$ と表すことができる．線形近似を用いて V_{next} を $\boldsymbol{\Delta U}_k$ に関する二次形式，制約条件を一次式に近似し，更新量 $\boldsymbol{\Delta U}_k$ を求める問題を式 (2.5) の制約条件付きの二次計画問題に定式化する．

$$\begin{aligned} &\text{minimize} \quad \Delta V = \frac{1}{2}\boldsymbol{\Delta U}_k^T \boldsymbol{\Phi}\, \boldsymbol{\Delta U}_k + \boldsymbol{\phi}^T \boldsymbol{\Delta U}_k \\ &\text{subject to} \quad \boldsymbol{\Omega}\, \boldsymbol{\Delta U}_k + \boldsymbol{\omega} \geq \boldsymbol{0} \end{aligned} \tag{2.5}$$

この問題式 (2.5) を逐次解いて $\boldsymbol{U}_k$ を更新していくことで，計画問題の解が得られる．

なお, 以降では入力列更新前と後の変数をそれぞれ $(\text{variable\_name})_{\text{now}}$, $(\text{variable\_name})_{\text{next}}$ の形で表すこととする．

2.5 評価関数

本論文では，目標状態到達を計画の指標とし，最適化する評価関数には目標追従二乗誤差を用いる．

$$V = \frac{1}{2}\|\boldsymbol{\varepsilon}\|^2 = \frac{1}{2}\left\|\boldsymbol{q}_{\text{goal}} - \boldsymbol{q}^k\right\|_{K_p}^2 \tag{2.6}$$

K_p は重み行列である．これを式 (2.4) の問題の評価関数として設定する．なお，目標追従以外の指標（ [**?**, **?**] における運動エネルギなど）も評価関数に加える事で最適化可能である．

問題式 (2.5) へ適用するため，$\boldsymbol{\Delta U}_k$ によって更新された後の状態 $\boldsymbol{q}_{\text{next}}^k$ を，$\boldsymbol{q}_{\text{now}}^k$ の $\boldsymbol{U}_k$ に

ついてのヤコビ行列 $J_k = \partial \boldsymbol{G}_k / \partial \boldsymbol{U_k}$ を用いて線形近似する．

$$\boldsymbol{q}^k_{\text{next}} = \boldsymbol{q}^k_{\text{now}} + J_k \, \boldsymbol{\Delta U}_k \tag{2.7}$$

式 (2.7) を用いて二次計画問題式 (2.5) の評価関数を式 (2.8) で定義する．

$$\begin{aligned} V_{\text{next}} &= \frac{1}{2} \left\| \varepsilon_{\text{next}} \right\|^2_{K_p} + \frac{1}{2} \left\| \boldsymbol{\Delta U}_k \right\|^2_{K_u} + \frac{1}{2} \left\| \boldsymbol{\Delta U}_{zk} \right\|^2_{K_{u_z}} \\ &= \frac{1}{2} \left(\varepsilon_{\text{now}} - J_k \, \boldsymbol{\Delta U}_k \right)^T K_p \left(\varepsilon_{\text{now}} - J_k \, \boldsymbol{\Delta U}_k \right) \\ &\quad + \frac{1}{2} \, \boldsymbol{\Delta U}_k^T K_u \, \boldsymbol{\Delta U}_k \\ &\quad + \frac{1}{2} \left(J_{zk} \, \boldsymbol{\Delta U}_k \right)^T K_{u_z} \left(J_{zk} \, \boldsymbol{\Delta U}_k \right) \\ &= \frac{1}{2} \, \boldsymbol{\Delta U}_k^T \Phi \, \boldsymbol{\Delta U}_k + \phi^T \, \boldsymbol{\Delta U}_k + \text{const.} \\ \Phi &= J_k^T \, K_p J_k + K_u + J_{zk}^T \, K_{u_z} J_{zk} \\ \phi &= -J_k^T \, K_p \, \varepsilon_{\text{now}} \end{aligned} \tag{2.8}$$

K_p，K_u，K_{u_z} は評価関数の各項を重み付ける対角行列である（対角成分の設定方法は**??**に示す）．$\phi \in R^{6k}$ はベクトル，$\Phi \in R^{6k \times 6k}$ は実対称の正定値行列である．

式 (2.8) の式変形前の V_{next} の定義において，第一項は前述の目標追従二乗誤差の評価項，第二項は入力列の更新量 $\boldsymbol{\Delta U}_k$ 最小化の評価項である．第三項は，高さ変位を最小化するための項である．脚配置モデルが状態として z_l^i，z_r^i を持つと仮定すると，式 (2.9) のように次ステップへ遷移すると考えられる．

$$\begin{aligned} z_l^{i+1} &= z_l^i + u_{zl}^i \\ z_r^{i+1} &= z_r^i + u_{zr}^i \end{aligned} \tag{2.9}$$

高さに対する仮想的な入力 u_{zl}^i，u_{zr}^i は式 (2.2) を用いて $\boldsymbol{q}^i$，$\boldsymbol{q}^{i+1}$，$z_{\text{field}}(x, y)$ から計算でき，高さに対する仮想入力列 $\boldsymbol{U}_{zk} = (u_{zl}^0, u_{zr}^0, \ldots, u_{zl}^{k-1}, u_{zr}^{k-1})^T$ は $\boldsymbol{q}^0$，$\boldsymbol{U}_k$ で表すことができる．入力列更新時の $\boldsymbol{U}_{zk}$ の変化 $\boldsymbol{\Delta U}_{zk}$ を最小化すれば，高さ方向の変位を最小化する計画が得

られる．なお，式 (2.8) 中の J_{zk} は，$\boldsymbol{U}_{zk}$ の $\boldsymbol{U}_k$ についてのヤコビ行列 $J_{zk} = \partial \boldsymbol{U}_{zk} / \partial \boldsymbol{U}_k$ である．

2.6 制約条件

制約条件は，三次元上の脚配置を実現するために必要となる地形との接触についての制約と，適用するロボットに合わせて定式化されるハードウェア上の制約からなる．

2.6.1 地形との接触制約

平面上で定義されたモデルを用いて三次元の脚配置を実現するため，地形との接触について制約を課す必要がある．脚の垂直方向の移動が平面動作から独立していること，機構的に踏み出し時の高さには限界があることを踏まえ，式 (2.2) を用いて両脚間の高低差について制約を課す．

$$\begin{aligned} -z_{\max} \leq z_l^{i+1} - z_r^i \leq z_{\max} \\ -z_{\max} \leq z_r^{i+1} - z_l^{i+1} \leq z_{\max} \end{aligned} \tag{2.10}$$

$z_{\max}$ は高低差の絶対値上限である．この制約を課すと，歩行時の垂直方向の移動が上下限値内に収まるように歩幅を抑えた脚配置が求まることになり，地形を考慮した脚配置計画が実現される．地形形状に関しては，平面位置に対応する高さ $z_{\text{field}}(x, y)$，線形近似時に必要となる x，y 方向それぞれの傾き $\partial z_{\text{field}}/\partial x$, $\partial z_{\text{field}}/\partial y$ が既知であればよい．上限値 $z_{\max}$ は，定数としても関数としてもよい．即ち，踏み出しの高さによって遊脚の水平方向の可動範囲が変化するような場合には，$\boldsymbol{q}^i$，$\boldsymbol{u}^i$ および z_{field} を用いて表現される関数とするなど，適用するロボットに合わせて定式化を行えばよい．

なお，現在は地形の傾斜に対する脚先の姿勢を考慮していない．今後，脚裏形状や制御における許容範囲，地形形状情報から定式化できると考えている．

2.6.2 遊脚の配置制約

制約条件を定式化するために，**??**，**??**および**??**中に示している，次の三つのパラメータを定義する．

- 支持脚と遊脚の相対距離 L^i
- 支持脚の y 軸から見た直線（L^i）の傾き θ_1^i
- 支持脚を中心としたときの両脚先間の相対角度 θ_2^i

L^i，θ_1^i，θ_2^i はそれぞれ，右脚支持の場合に $L^i = l^i + u_{l1}^i$，$\theta_1^i = u_A^i - \theta^i + \theta_{fr}^i$，$\theta_2^i = \theta_{fl}^{i+1} - \theta_{fr}^i$，左脚支持の場合に $L^i = l^{i+1}$，$\theta_1^i = \theta^{i+1} - \theta_{fl}^{i+1}$，$\theta_2^i = \theta_{fl}^{i+1} - \theta_{fr}^{i+1}$ となる．

まず，支持脚に対する遊脚の相対角度 θ_2^i に制約を課す．これは式 (2.11) で定式化される（**??**参照）．

$$\theta_{2\min} \leq \theta_2^i \leq \theta_{2\max} \tag{2.11}$$

$\theta_{2\max}$，$\theta_{2\min}$ は相対角度の上下限値である．

次に，支持脚に対する遊脚の可動範囲を**??**に示す領域内に制限する制約を課す．支持脚に対する遊脚の相対位置座標 $(x_{sw}^i, y_{sw}^i) = (L^i \sin\theta_1^i, L^i \cos\theta_1^i)$ は，$x_{sw}^i \geq 0$，$x_{sw}^i < 0$ それぞれの範囲において，$(0, Y_{\min})$ を中心とする楕円形の領域に制限されることとなる．この楕円の径については，$x_{sw}^i \geq 0$，$x_{sw}^i < 0$ それぞれの領域において，長軸は共通だが短軸は異なる．

$$\begin{gathered} y_{sw}^i \geq Y_{\min} \\ \left(\frac{x_{sw}^i}{X}\right)^2 + \left(\frac{y_{sw}^i - Y_{\min}}{Y_{\max} - Y_{\min}}\right)^2 \leq 1.0 \\ \left(X = \begin{cases} X_{\max} & \left(x_{sw}^i \geq 0\right) \\ X_{\min} & \left(x_{sw}^i < 0\right) \end{cases} \right) \end{gathered} \tag{2.12}$$

$Y_{\max}$，$Y_{\min}$ は Σ_{sp} の y 軸上の両脚間隔の上限値及び下限値．また，$X_{\max}$，$X_{\min}$ は，脚間距

離が最小のときの前進・後進それぞれについての Σ_{sp} の x 軸方向の歩幅上限値.

2.6.3 脚先の干渉防止制約

さらに，脚先が重なることを防ぐための制約を課す．**??**に示している，支持脚座標系 Σ_{sp} から見た遊脚の頂点 $\boldsymbol{R}_A^i = (x_{R_A}^i, y_{R_A}^i)$ と $\boldsymbol{R}_B^i = (x_{R_B}^i, y_{R_B}^i)$，また遊脚座標系 Σ_{sw} から見た左脚の頂点 $\boldsymbol{L}_A^i = (x_{L_A}^i, y_{L_A}^i)$ と $\boldsymbol{L}_B^i = (x_{L_B}^i, y_{L_B}^i)$ は式 (2.13) で表される．

$$
\begin{aligned}
x_{R_A}^i &= L^i \sin\theta_1^i + O_{xf} \cos\theta_2^i + O_{yi} \sin\theta_2^i \\
y_{R_A}^i &= L^i \cos\theta_1^i + O_{xf} \sin\theta_2^i - O_{yi} \cos\theta_2^i \\
x_{R_B}^i &= L^i \sin\theta_1^i - O_{xb} \cos\theta_2^i + O_{yi} \sin\theta_2^i \\
y_{R_B}^i &= L^i \cos\theta_1^i - O_{xb} \sin\theta_2^i -^i O_{yi} \cos\theta_2^i \\
x_{L_A}^i &= L^i \sin\left(-\theta_1^i - \theta_2^i\right) + O_{xf} \cos\theta_2^i + O_{yi} \sin\theta_2^i \\
y_{L_A}^i &= L^i \cos\left(-\theta_1^i - \theta_2^i\right) + O_{xf} \sin\theta_2^i - O_{yi} \cos\theta_2^i \\
x_{L_B}^i &= L^i \sin\left(-\theta_1^i - \theta_2^i\right) - O_{xb} \cos\theta_2^i + O_{yi} \sin\theta_2^i \\
y_{L_B}^i &= L^i \cos\left(-\theta_1^i - \theta_2^i\right) - O_{xb} \sin\theta_2^i - O_{yi} \cos\theta_2^i
\end{aligned}
\tag{2.13}
$$

提案するモデルにおける脚先形状は実際のロボットの脚先を包含できる最小の長方形としている．$O_{yi}O_{yo}$ はそれぞれ脚先中心から内側および外側の辺までの長さ，O_{xf}，O_{xb} はそれぞれ脚先中心から前端，後端までの長さである．また，入力列の更新前後で制約式が不連続に変化するのを防ぐために，**??**に示すような左脚座標系から見た左脚内側，右脚座標系から見た右脚内側の形状を表現する関数 $f_f(x)$ を導入する．脚先座標系の y 軸の正の方向に凸で，パラメータによって矩形に近づけられる関数が望ましく，本論文では三つのパラメータ a，b，c をもつ x の 10 乗の関数とした．

$$
f_f(x) = a\,(x-b)^{10} + c \tag{2.14}
$$

パラメータは Levenberg-Marquardt 法による非線形最小二乗推定によりフィッティングを行った．これを用いて，制約条件を式 (2.15) で表現する．

$$\begin{aligned} f_f\left(x^i_{R_A}\right) \leq y^i_{R_A} \quad , \quad f_f\left(x^i_{R_B}\right) \leq y^i_{R_B}, \\ f_f\left(x^i_{L_A}\right) \leq y^i_{L_A} \quad , \quad f_f\left(x^i_{L_B}\right) \leq y^i_{L_B} \end{aligned} \tag{2.15}$$

2.6.4 二次計画問題への適用

以上の制約条件 $\omega_i(\boldsymbol{q}^0, \boldsymbol{U}_k) \geq 0$ $(i = 0, \ldots, 20k-1)$ は，まとめて $\boldsymbol{\omega}(\boldsymbol{q}^0, \boldsymbol{U}_k) = (\omega_0, \ldots, \omega_{20k-1})^T \geq \boldsymbol{0}$ $(\boldsymbol{\omega} \in R^{20k})$ と表される．これを問題式 (2.5) に適用するため，更新後の制約条件について線形近似を行う．

$$\boldsymbol{\omega}_{\text{next}} = \boldsymbol{\omega}_{\text{now}} + \Omega\, \boldsymbol{\Delta U_k} \geq \boldsymbol{0} \tag{2.16}$$

$\Omega \in R^{20k \times 6k}$ は $\boldsymbol{\omega}_{\text{now}}$ の $\boldsymbol{U}_k$ についてのヤコビ行列 $\Omega = \partial \boldsymbol{\omega}_{\text{now}} / \partial \boldsymbol{U}_k$ である．

2.7 本章のまとめ

本章では，三次元適応のための歩行モデルの拡張および追加の制約条件について説明した．従来の水平方向の脚配置計画のためのモデルに，独立した垂直方向の動作を追加し，脚配置遷移モデルの状態および入力についての次元の拡張を行うことなく，事前に与えられた地形情報に基いて垂直方向の移動量を制限する制約を定式化し追加することで，計算コスト増加を抑えつつ三次元地形への適応を可能にした．次章では，本章で定義した計画問題の解法について，全体のアルゴリズムおよび求解性能の安定化のためのいくつかの手法について述べる．

第 3 章

複数 LiDAR の位置校正方法とオクルージョン範囲の導出

3.1 はじめに

本章では，LiDAR の位置を取得するために使用した手法を説明する．また，決定した LiDAR の位置とそれが認識した人物の位置からオクルージョンが起こっている範囲を導出し，その範囲を UAV が撮影できるように目標位置を設定する手法を説明する．

3.2 複数 LiDAR の位置決定方法

まず始めに，複数 LiDAR の位置の取得方法について述べる．先行研究では 1 台の LiDAR の位置を UAV に搭載している GPS センサと同じもの使用してを取得していた．しかし，UAV に搭載されている GPS センサから得られる緯度経度情報は数 m から十数 m の誤差がある．複数の LiDAR を使用する際，位置の誤差が大きいと，1 つの物体を 2 つの物体と誤認してしまう恐れがある．そのため，２つの LiDAR の正確な相対位置を取得するために

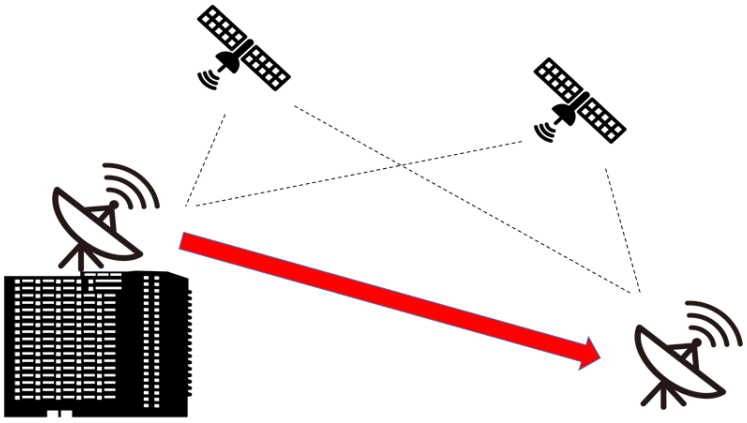

Fig. 3.1　Image of RTK-GPS Positioning System

RTK-GPS 測位という手法を利用して位置を取得する．

3.2.1　RTK-GPS 測位

RTK-GPS(Real-Time Kinematic GPS) 測位とは，位置が分かっていて移動しない基地局 (Base) と位置情報を取得しようとしている観測点である移動局 (Rover) で同時に GPS 観測を行い，基準局で観測したデータを移動局へリアルタイムに送信し，基準局の位置に基づいて移動局の位置を求める手法である．さらにネットワークを利用して基地局と移動局のデータ送信を行うことで，基地局と移動局が長距離で離れていても精度の高い演算ができる．

3.2.2　RTK-GPS 測位と単独測位の比較

単独測位では数 m から十数 m の誤差が発生するのに対して，RTK-GPS 測位では数 cm の誤差が発生するといわれている．ここで，実際に計測したデータを比較して測位の性能の差を述べる．計測位置は大阪市立大学 F 棟 507 号室のベランダであり，Fig. 3.2 は 30 秒の

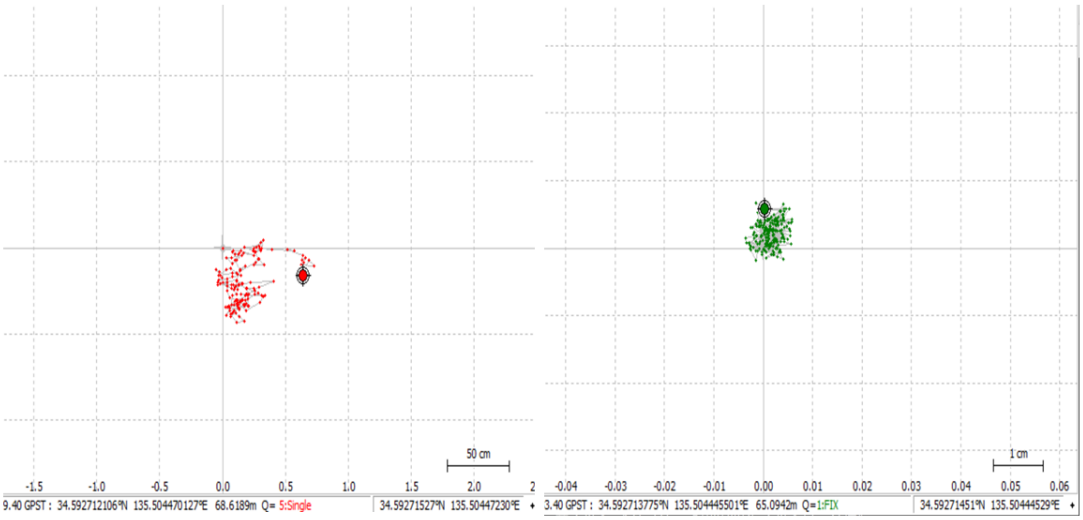

Fig. 3.2　Single GPS Positioning (right) and RTK-GPS Positioning (left)

計測データをグラフ化したものである．Fig. 3.2 左のグラフは単独測位の結果であり，グラフは 1 マス 50cm である．同図右のグラフは RTK-GPS 測位の結果であり，グラフは 1 マス 1cm である．グラフはから見て取れるように，単独測位は 50cm から 1m の誤差があり，RTK-GPS 測位の誤差は 1cm から 2cm 以内に収まっている．

3.3　GPS 座標から map 座標への変換

本研究では，東を x 座標正方向，北を y 座標正方向とする map 座標系を用いる．また座標原点は LiDAR1 の座標をもとに算出する．ここでは，GPS センサから得た LiDAR1 の緯度経度と座標から原点の緯度経度の値の算出方法，原点の緯度経度の値と LiDAR2 の緯度経度の値から LiDAR2 の map 座標の算出方法を述べる．

LiDAR1 の GPS 座標を $lon_{lidar1}, lat_{lidar1}$,map 座標を x_{lidar1}, y_{lidar1} とすると求める map 座標原点の緯度経度の値 $lon_{origin}, lat_{origin}$ は以下の式 (3.1) で算出される．なお式中の R は地球の赤道半径である．

$$
\begin{aligned}
lat_{origin} &= \frac{y_{lidar1}}{R} \times \frac{180}{\pi} + lat_{lidar1} \\
lon_{origin} &= \frac{x_{lidar1}}{R} \times \frac{180}{\pi} \times \frac{1}{cos(lat_{origin}\frac{180}{\pi})} + lon_{origin}
\end{aligned}
\tag{3.1}
$$

式 (3.1) より得られた map 座標原点の緯度経度の値を用いて，LiDAR2 の map 座標 x_{lidar2}, y_{lidar2} を式 (3.2) 求めることができる．LiDAR2 の GPS 座標を $lon_{lidar2}, lat_{lidar2}$ とする．

$$
\begin{aligned}
x_{lidar2} &= R(lon_{lidar2} - lon_{lidar2})\frac{\pi}{180}cos((lat_{lidar2} - lat_{origin})\frac{\pi}{180}) \\
y_{lidar2} &= R(lat_{llidar2} - lat_{origin})\frac{\pi}{180}
\end{aligned}
\tag{3.2}
$$

3.4 オクルージョンが発生した場合の UAV の撮影位置の決定

前節まででは，LiDAR の位置を取得するための手法を述べた．人物行動範囲内に複数人の人物が存在する場合，オクルージョンが発生し．地上設置 LiDAR では人物行動範囲内の全ての人物をとらえきれない場合が存在する．この節では，オクルージョンが発生した場合の LiDAR の撮影できない領域と，その領域を考慮した UAV の目標位置の導出方法について述べる．

.

3.4.1 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

$$
\begin{aligned}
K_p &= \mathrm{diag}\,\{k_x, k_y, k_\theta, k_l, k_{\theta_{fl}}, k_{\theta_{fr}}\} \\
k_x &= k_y = (\max\{|x_{\mathrm{goal}} - x^0|, |y_{\mathrm{goal}} - y^0|\})^{-2} \\
k_\theta &= (2\pi)^{-2} \\
k_l &= (Y_{\max} - Y_{\min})^{-2} \\
k_{\theta_{fl}} &= k_{\theta_{fr}} = (2\pi)^{-2}
\end{aligned}
\tag{3.3}
$$

$$
\begin{aligned}
K_u &= \mathrm{diag}\,\{\ldots, k_{u_A}, k_{u_B}, k_{u_{l1}}, k_{u_{l2}}, k_{u_{fl}}, k_{u_{fr}}, \ldots\} \\
k_{u_A} &= k_{u_B} = (2\tan^{-1}(X_{\max}/Y_{\min}) + \theta_{2\max})^{-2} \\
k_{u_{l1}} &= k_{u_{l2}} = (Y_{\max} - Y_{\min})^{-2} \\
k_{u_{fl}} &= k_{u_{fr}} = (\theta_{2\max} - \theta_{2\min})^{-2}
\end{aligned}
\tag{3.4}
$$

$$
\begin{aligned}
K_u &= \mathrm{diag}\,\{\ldots, k_{u_{zl}}, k_{u_{zr}}, \ldots\} \\
k_{u_{zl}} &= k_{u_{zr}} = z_{\max}^{-2}
\end{aligned}
\tag{3.5}
$$

- K_u，K_{u_z} は，前項の設定に加えて Sugihara の手法 [**?**] を適用し，評価関数式 (2.6) 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）

$$
\begin{aligned}
K_u &\leftarrow V_{\mathrm{now}}\, K_u + \Delta k_u\, I = \frac{1}{2}\|\varepsilon_{\mathrm{now}}\|^2_{K_p}\, K_u + \Delta k_u\, I \\
K_{u_z} &\leftarrow V_{\mathrm{now}}\, K_{u_z} + \Delta k_{u_z}\, I = \frac{1}{2}\|\varepsilon_{\mathrm{now}}\|^2_{K_p}\, K_{u_z} + \Delta k_{u_z}\, I
\end{aligned}
\tag{3.6}
$$

- 一つ一つの制約条件 $\omega_i(\boldsymbol{q}^0, \boldsymbol{U}_k) \geq 0$ 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 [**?**]
- 緩和問題を解く際，評価関数全体を $\|\phi\|$ 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．

3.4.2 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．

$$\begin{aligned} H\left(\boldsymbol{q}^0, \boldsymbol{U}_k + \alpha\boldsymbol{\Delta U}_k\right) &\leq H\left(\boldsymbol{q}^0, \boldsymbol{U}_k\right) + \beta\alpha\nabla V\left(\boldsymbol{q}^0, \boldsymbol{U}_k\right)^T \boldsymbol{\Delta U}_k \\ &= H\left(\boldsymbol{q}^0, \boldsymbol{U}_k\right) + \beta\alpha\boldsymbol{\phi}^T \boldsymbol{\Delta U}_k \end{aligned} \tag{3.7}$$

β は，$0 < \beta < 1$ を満たす定数である．今回，α は**??**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．

3.5 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）
- $\boldsymbol{\Delta U}_k$ 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）
- $\omega_i(\boldsymbol{q}^0, \boldsymbol{U}_k) \geq 0$ $(i = 0, \ldots, 20k - 1)$ 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 $\boldsymbol{U}_k$ 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．

3.6 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．

- “$\boldsymbol{\Delta U}_k$ 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 $\varepsilon_{\text{reachable}}$ 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 $\boldsymbol{U}_k$ はその内容を引き継ぎ $\boldsymbol{U}_k \leftarrow (\boldsymbol{U}_k^T, \boldsymbol{0}^T)^T$ 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．

3.7 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．

3.8 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．

第 4 章

実環境におけるシステム構築

4.1 はじめに

2 章および 3 章で説明した複数 LiDAR を用いた人物追従システムと UAV 撮影計画を実環境において構築する．本章では，今回構築したシステムに利用した開発プラットフォームおよび機器について説明する．

4.2 Robot OperationSystem の概要

本研究では，ロボット用のソフトウェアプラットフォームである Robot OperationSystem(ROS) を利用している．ROS は Ubuntu や Linux Mint などの Linux 系 OS でサポートされているミドルウェアやソフトウェアフレームワークの一種である．またロボットを動作させるためのプロセス間の通信，パッケージ管理，ソフトウェア開発に必要なツールやライブラリを提供している．ROS でサポートされている言語は主に C++ と Python であり Java や Lisp などの言語も使用できる．つづいて ROS 通信について説明する．ROS ではプログラムをノードと呼ばれる比較的小さなプログラムに細分化して実装し，そのノード間で情報をやりとりしてロボットなどの制御を行う．ノード間の通信は主に 3 種類あり，単方向非同

期通信方式のトピック通信，双方向同期通信方式のサービス通信，双方向非同期通信方式のアクション通信を利用できる．本研究ではトピック通信のみを用いているので，サービス通信，アクション通信の説明は省略する．図にトピック通信のイメージ図を示す．トピックとはノード間でやりとりするメッセージの名前であり，トピックを一定周期で配信するノードを配信者ノードと配信されたトピックを受ける購読者ノードによって通信システムは構築される．ノード実行時に ROS マスターにトピック名とメッセージの形式が登録され，購読者ノードは ROS マスター内で受信したいトピックを配信しているノードを探索し，その配信者ノードと通信を行う．本実験では，撮影計画を実行する PC（ホスト，LiDAR に接続している PC，UAV を直接制御しているタブレットに ROS を導入し，地上から UAV の制御を行う．

4.3 ROS による UAV 制御

本研究では，撮影計画を実装する UAV として，民生用ドローンおよび関連機器製造会社である DJI 社の Mavic Mini を複数台使用する．以下，MavicMini の性能の概要と制御を行うための通信モデルについて説明する．

MavicMini は GPS センサを搭載しており，フライトコントローラで自身の位置情報を管理している．昨年度まで使用していた同社が販売している Matrice600 とは異なり，ROS と直接通信ができる DJI Onboard SDK が使用できない．そのため MavicMini の制御は DJI Mobile SDK を用いて行う．DJI Mobile SDK は Android 端末，IOS 端末のアプリによりフライトコントローラと ROS の通信を可能にする API を提供する．今回の実験では Android 端末で MavicMini を制御するアプリを開発し，無線 LAN によって ROS マスターと通信を行う．図に通信モデルを示す．

4.4　地上設置 LiDAR と GPS センサ

LiDAR はレーザ機器やセンサなどの開発を行う北陽電機株式会社の UTM-30LX-EW，同社の UST-30LX の 2 台を使用する．性能はどちらも検出範囲は 270[deg]，角度分解能は 0.25[deg] であり，LAN ケーブルから点群情報を送信する．

本研究では 3 章で示した通り, LiDAR の位置情報を GPS センサから取得し, map 座標を得

Source：https://www.dji.com/jp/mavic-mini

Fig. 4.1　Mavic Mini

る．今回の実験では GPS センサは ublox 社のマルチバンド GNSS アンテナの ANN-MB-01 を使用する．

昨年度の研究では LiDAR の情報を有線 LAN ケーブルでホスト PC に送信していたが，LiDAR を複数台使用する場合，有線 LAN ケーブルで接続するのは非常に不便であり効率が悪い．そこで，LiDAR の点群情報と GPS センサからの位置情報を無線でホスト PC に送信するシステムを構築する，LiDAR や GPS センサから直接データを無線通信することができないので，一旦ホスト PC とは別の PC に接続してその PC からホスト PC へ無線接続する．LiDAR と接続する PC（サブ PC) には Raspberry Pi4 modelB を採用した．Raspberry Pi4 は Ubuntu server OS に対応しており ROS をインストールすることができる．LiDAR をサブ PC に有線 LAN ケーブルで接続し，サブ PC 内で点群情報を ROS トピックとして配信するノードを立ち上げることでホスト PC との無線接続を可能とする．GPS 情報はサブ PC にデータを取り込んだ後，サブ PC とホスト PC で TCP 通信を行う．ホスト PC で，受け取った GPS 情報をトピックに変更するノードを立ち上げることにより，LiDAR の位置情報

Source：https://www.hokuyo-aut.co.jp/search/single.php?serial=146

Fig. 4.2　UTM-30LX-EW

を UAV 撮影計画の ROS プログラム内で使用することができる．

4.5　本章のまとめ

本章では，HRP-2 に基づいたパラメータを用いて，連続的変化のある地形と不連続な地形とでいくつかの問題設定で計画を行った．そして，不連続な地形上では計算が不安定化してしまう問題が残るが，連続的な地形では安定して最適化された脚配置計画を得ることができ，またいずれにおいても計算時間は両脚支持期間よりも短く，実時間性を有していることが示された．

Source：https://www.hokuyo-aut.co.jp/search/single.php?serial=195

Fig. 4.3　UST-30LX

Source：https://www.hokuyo-aut.co.jp/search/single.php?serial=146

Fig. 4.4　Multi-band GNSS Antenna ANN-MB-01

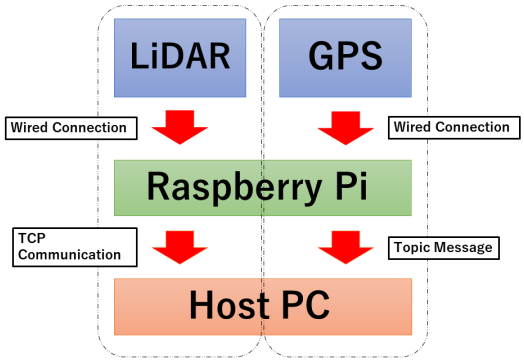

Fig. 4.5　Wireless Communication Model for LiDAR and GPS antenna

第 5 章

NAO における

脚配置計画シミュレーション

本章では，Aldebaran Robotics のヒューマノイドロボット NAO [**?**] を用いた計画シミュレーションおよび実機実験，また kinect v2 [**?**] で計測した地形情報を用いた計画シミュレーションについて記述する．

5.1　踏破性能とパラメータ設定

パラメータとしては，NAO に標準搭載されているシステム NAOqi が提供する脚配置指定による歩行指示の API を用いることを前提に，それらを利用する際に定められている限界値を各制約条件のパラメータとして設定する [**?**]．NAO の歩行制御については，標準の制御用 API としては二次元平面上の歩行を前提としたものしか提供されておらず，床面の上下方向の誤差に対するロバスト性を持った二次元平面上における歩行制御が行われている [**?**]．そのため，本章におけるシミュレーションおよび実験では，提案手法における垂直方向移動の上下限値には，脚先の上下方向の誤差に対するロバスト性において耐えうると [**?**] で示さ

Table 5.1 Model parameters for NAO

parameters	value
$z_{\max}$	0.5 cm
$d_{xy\,\max}$	16.73 cm
$d_{z\,\max}$	1.0 cm
walking period (whole)	420–600 ms
$\theta_{2\max}$	30°
$\theta_{2\min}$	−30°
$X_{\max}$	6.0 cm
$X_{\min}$	−4.0 cm
$Y_{\max}$	16.0 cm
$Y_{\min}$	8.8 cm
O_{xf}	10.69 cm
O_{xb}	5.61 cm
O_{yi}	3.95 cm
O_{yo}	5.23 cm
a	-1.18245×10^{-10}
b	2.54
c	4.95

Table 5.2 Algorithm parameters for NAO

parameters	value
Δk_u	10^{-5}
Δk_{u_z}	10^{-5}
$\varepsilon_{\text{relax}}$	10^{-15}
$\alpha_{\min}$	0.15
β	0.5
γ	0.95
$\varepsilon_{\text{goal}}$	$(0.05, 0.05, 0.05, 0.05, 0.05, 0.05)^T$ (cm, deg)
$\varepsilon_{\text{input}}$	$(\ldots, 10^{-6}, 10^{-6}, 10^{-6}, 10^{-6}, 10^{-6}, 10^{-6}, \ldots)^T$ (cm, deg)
$\varepsilon_{\text{reachable}}$	$(2.5, 2.5, 0.5, 1.0, 0.5, 0.5)^T$ (cm, deg)
$n_{\text{reachable1}}$	10
$n_{\text{reachable2}}$	$3k$

Table 5.3 Map parameters for NAO

parameters	value
D_{grid}	5.0 cm
D_{sample}	1.0 cm

Table 5.4　Result of planning (slope)

θ_{slope} (deg)	k_{init}	k	iteration	time(ms)
0.0	11	11	10	10.84
4.5	11	12	52	65.20

れている誤差の許容値を設定する．検証内容としては，三次元の地形形状を持つフィールドにおいて，ロバスト性の範囲内で踏破可能となる，目標状態までの最適化された脚配置を計画できることを検証するものとする．

パラメータは Table 5.1，Table 5.2 および Table 5.3 のとおりである．

5.2　スロープ型の地形でのシミュレーション

初期状態 $\boldsymbol{q}^0 = (0.0, 0.0, 0.0, 10.0, 0.0, 0.0)^T$ (cm, deg) から目標状態 $\boldsymbol{q}_{\text{goal}} = (50.0, 150.0, 0.0, 10.0, 0.0, 0.0)^T$ へと向かう計画を行う．地形としては，y 軸方向において平面と傾斜角 θ_{slope} の斜面が $y = 50\,\text{cm}$ の位置で接続しているスロープ型の地形を用いる．

$$z_{\text{field}}(x,\, y) = \begin{cases} 0.0 & (y < 50.0) \\ (y - 50.0)\tan\theta_{\text{slope}} & (50.0 \leq y) \end{cases} \tag{5.1}$$

$\theta = 0.0°$，$4.5°$ の 2 パターンで計画を行う．

??，Table 5.4 に計画結果を示す．$4.5°$ の場合，平面上での歩幅に対して斜面上では歩幅を小さく取り，高低差制約を守りながら進む計画になっているのが**??**からも分かる．歩幅が狭いためステップ数は初期値から 1 増やす必要があり，また反復回数，計算時間は $0.0°$ の場合に比べていずれも増加している．

5.3 実地形データを用いた地形情報構築と計画

実際のロボットが自律移動を行う場合，環境の設計図などから事前に得られた地形形状を用いるのではなく，環境中で測域センサなどを用いて計測したデータから適宜地形情報を構築し，脚配置計画を行う必要がある．

ここでは，Kinect v2 を用いて実際に取得した点群データから地形情報を構築し，その地形データを用いて計画シミュレーションを行う．5.2 節と同様のスロープ型地形を用意し，ワールド座標系基準で (40.0, −113.0, 88.5) の位置にチルト角 15.4° で見下ろす姿勢で設置した Kinect v2 で距離画像を取得する．そして，取得した画像をワールド座標系基準の点群データへ変換し，それを用いて 3.7 節に示す手順で $z_{\mathrm{field}}(x, y)$ を構築する．問題としても 5.2 節と同様，初期状態 $\boldsymbol{q}^0 = (0.0, 0.0, 0.0, 10.0, 0.0, 0.0)^T$ (cm, deg) から目標状態 $\boldsymbol{q}_{\mathrm{goal}} = (50.0, 150.0, 0.0, 10.0, 0.0, 0.0)^T$ へと向かう計画を行う．

??および**??**に結果を示す．ワンスキャンでの計測であること，およびノイズを考慮した地形情報構築ができていないことなどから，マップに歪みが含まれている．そのため，初期状態付近で地形適応制約を守るために小さい歩幅で歩行する必要があり，理想関数上での計画結果とは異なり 13 ステップでの計画となっている．計算時間は十分歩行周期以内に収まっており，実時間制は保たれている．

5.4 実機実験

ここからは，実機を用いた実験について述べる．

NAO への歩行指令では，まず脚配置計画を行い，得られた脚配置の系列から踏み出し量の系列を求める．次に，NAO の制御システムである NAOqi OS が標準で提供している，踏み出し量と時間を指定して踏み出しを行わせる API を歩数分だけ呼び出して歩行を指令する．この API 呼び出しに対して，NAOqi OS 内の歩行制御モジュールによって軌道生成が行われ，指示した脚配置計画を実現する歩行制御が行われる．なお，この API は平面上の歩

行のためのものであり，平面上での位置と姿勢を指令するもので，内部で行われる歩行制御も平面歩行のためのものである．また，内蔵カメラの画像を用いた自己位置および姿勢の推定は行われておらず，外部のカメラによる計測や補正も行っていない．

5.4.1 計画の実機適用と動作確認

まず，提案手法によって得られる脚配置計画が実際に実機に適用可能であることを確認する実験を行った．5.2 節におけるシミュレーションの傾斜角 4.5° の場合の計画を NAO に適用した．**??**にその結果を示す．

平面上の歩行を仮定した歩行制御を用いて斜面を歩かせているため，歩行が不安定になり姿勢が計画からずれたものとなった．しかし，地形制約により上下方向の移動が NAO の許容値である 0.5 cm 以下に抑えられた計画がなされているため，制御のロバスト性で姿勢を維持でき，目標状態付近まで到達することができている．これにより，提案手法で実機に適用可能な脚配置計画が得られていることが確認できた．

5.4.2 目標到達精度および歩行の安定性の評価

次に，終了時の位置・姿勢と，歩行中の足裏圧力センサの値を計測し，目標状態への到達精度と，歩行の安定性について評価する実験を行った．いずれの評価においても，5.2 節における傾斜角 4.5° の場合の地形を対象とし，提案手法による地形形状を考慮した計画と，従来手法による平面上での計画の結果それぞれを用いて歩行させ，データを計測した．なお，提案手法を用いない場合は 11 ステップの計画となっており，提案手法における計画結果の 12 ステップより少ない．

まず，目標状態への到達精度の評価について述べる．歩行終了時の左脚の位置 x, y および姿勢 θ を計測し，計画で設定した目標状態 ($x = 50\,\mathrm{cm}$, $y = 150\,\mathrm{cm}$, $\theta = 0°$) に対する誤差を評価した．なお，初期状態は ($x = 0\,\mathrm{cm}$, $y = 0\,\mathrm{cm}$, $\theta = 0°$)．提案手法，従来手法それぞれで五回ずつ実験を行った．なお，NAO による歩行精度の参考として，提案手法での計画を用いて平面上を歩かせた場合の歩行についても同様に計測を行った．Table 5.5 に，提案手

Table 5.5 Results of evaluation experiment of goal reaching error

mathod	x (cm)	y (cm)	θ (deg)
proposal	-24.4 ± 2.5	5.3 ± 1.0	21.0 ± 2.7
nonproposal	-46.5 ± 10.3	5.9 ± 3.3	39.1 ± 7.4
proposal (plane)	2.3 ± 3.7	6.7 ± 1.6	5.4 ± 2.8

法および従来手法での斜面歩行における誤差および提案手法での平面歩行における誤差の，五回の試行での平均と標準偏差を示す．まず評価の前提として，提案手法で平面を歩行させた場合の結果から NAO による歩行では平面上においても誤差が生じてしまうこと，また提案手法での斜面歩行の結果と比較すると x，θ について誤差が増加しており斜面歩行では更に精度が落ちることがわかる．これを踏まえても，提案手法と従来手法とによる斜面歩行の結果を比較すると，いずれの要素についても明らかに誤差が改善されており，特に x，θ においては大きな改善が見られる．標準偏差を比較しても，いずれの要素においても提案手法のほうが小さく抑えられている．これらの結果から，従来手法に対して，提案手法によって目標到達の精度が改善されていることが分かる．

次に，片脚支持期の ZMP 値を用いた歩行の安定性の評価について述べる．NAO の左右の足裏それぞれ四ヶ所（(7.025, 2.310)，(7.025, –2.990)，(–3.025, 1.910)，(–2.965, –2.990)(cm)）に配置されている圧力センサの値を 25 ms ごとに計測し，脚先座標系 Σ_{fl} および Σ_{fr} 基準の片脚支持期の ZMP 応答値を算出した．

提案手法と従来手法とで 12 s 以降の左脚の ZMP 応答を比較する．x 軸方向**??**及び y 軸方向**??**いずれについても，従来手法においては，本来 12 s 以前の応答と同様にすぐに 0 に戻るはずであるが，0 から変化したあとに戻るのが明らかに遅く，歩行が不安定になっていることが分かる．提案手法ではこのような ZMP の応答は見られず，それ以前と同様の応答を示しており，斜面上においても安定して歩行ができていることが分かる．

最後に，床反力の評価について述べる．ZMP 評価と同様に三回目の実験データを用い，左右それぞれについて圧力センサの示す値の合計値を比較した．一つの圧力センサで最大 25 N まで計測でき，片脚全体の床反力は最大 100 N である．**??**に提案手法および従来手法それぞれにおける左右の足裏の床反力の遷移を示す．

提案手法を用いない場合の結果において，左脚の床反力値が 12 s 以降に高い値をとっている．これはバランスを崩したことによって左脚が強く着地してしまったことを表しており，歩行が不安定になっていたと考えられる．これに対して，提案手法では床反力は最後まで周期的に変化しており，歩行の安定性が改善されていると言える．

以上の実験データおよび評価から，目標到達精度，および ZMP 応答および床反力で評価した歩行の安定性ついて提案手法の有効性が確認された．

5.5 本章のまとめ

本章では，二足歩行ロボット NAO において，標準の平面歩行制御における垂直方向のロバスト性を利用した三次元地形での計画として，まず，計画シミュレーションによりスロープ形状の地形に適応した脚配置計画が得られることを確認した．また，Kinect v2 センサを用いて実際に計測したスロープに対しても同様に計画ができることを示した．さらに，実機実験により実際に前述のロバスト性を利用してスロープを歩行できる計画が得られていることや，目標到達の精度や歩行の安定性について有効性があることを示した．

第 6 章

結論

6.1 まとめ

本論文では，二足歩行ロボットにおける地形形状を考慮した三次元脚配置計画の手法を提案した．従来の水平方向の脚配置計画のためのモデルに，独立した垂直方向の動作を追加し，脚配置遷移モデルの状態および入力についての次元の拡張を行うことなく，事前に与えられた地形情報に基いて垂直方向の移動量を制限する制約を定式化し追加することで，計算コスト増加を抑えつつ三次元地形への適応を可能にした．ヒューマノイドロボット HRP-2 を想定した計画シミュレーションを行い，連続的変化の三次元地形に対しては実時間で脚配置計画の最適化が行えることを示した．不連続な変化のある地形では計算が不安定化してしまうといった課題が残るが，地形情報の構築方法や計算手法などの見直しで改善できると考えられる．また，ヒューマノイドロボット NAO を想定した計画シミュレーションも行い，HRP-2 想定の時と同様に実時間で地形適応を実現したかいが得られることを示し，また実機実験により，実際に二足歩行ロボットに適用可能な脚配置が得られていることを確認した．

6.2 課題と展望

残された課題として，まず計画シミュレーションにおいても示された不連続な地形変化に対する求解性能の低下を改善する必要がある．現在の二次計画法を利用した反復法による最適化は，[?] でモデル予測制御に使われていることなどから実用上は十分なものであるが，その収束性については厳密には保証されていない [?]．また，脚先の三次元的な姿勢および形状を考慮できていないという問題がある．これらを考慮して制約条件を追加し，地形との接触に関して傾きの大きすぎる場所への接地を避けるような計画を行う必要がある．これに関連して，地形情報の構築方法についても，現在の簡易的なものではなく，誤差を考慮した制約設定を可能にする，実際の形状からの誤差範囲を保証できるような方法の検討が必要である．また，三次元歩行における消費エネルギを評価に含めて最適化を行う計画手法について検討を進めていく．これは，現在の近似前の評価関数は目標状態との誤差のみを評価しており，ロボットの移動において重要な指標となる消費エネルギを最適化できていないためである．

将来的な展望として，モデルベースと探索手法の住み分けについての検討が必要であると考えている．提案手法の目標距離の限界は原理上存在しないが，実際の制御では脚配置に誤差が必ず発生するため長距離に渡る最適化を行うのは実用上の意味があまりないと考えられる．そこで，離散的な粗い探索による長距離の歩行計画と，提案手法であるモデルベースの脚配置最適化を組み合わせた手法が必要になると考えており，組み合わせ方や住み分けの仕方について調査・検討を進めていく．また，現状の歩容制御と切り離した上での脚配置計画ではなく，歩行制御計画と脚配置計画とを統合した，歩容全体の計画を行う手法の検討を進めていく．

謝辞

本論文を作成するにあたり，終始懇切なるご指導，御鞭撻を賜りました本学電子情報系専攻の田窪朋仁教授に厚く御礼申し上げます．また，研究を進める上で様々なご助言をいただきました，上野敦志講師に感謝の意を表します．最後になりましたが，本研究室の皆様にも日頃からご協力頂いたことに感謝いたします．

参考文献

[1] Yuri Nishikawa, Hitoshi Sato, and Jun Ozawa, “Multiple sports player tracking system based on graph optimization using low-cost cameras”, 2018 IEEE International Conference on Consumer Electronics, pp.1–4, 2018.

[2] 西川由理, 佐藤仁, 小澤順, “グラフ最適化を用いた多人数追跡手法におけるグリッドマップ生成の並列化”, 人工知能学会全国大会論文集 2018 年度人工知能学会全国大会（第 32 回）論文集, pp.2D101–2D101, 2018.

[3] Teuliere C, Eck L, and Marchand E, “Chasing a moving target from a flying uav”, Conference on Intelligent Robots and Systems (IROS), pp.4929–4934, 2011.

[4] Bethke B, Valenti M, and How J, “Cooperative vision based estimation and tracking using multiple UAVs”, Advances in cooperative control and optimization, pp.179–189, 2007.

[5] T Nasser, J Sturm, and D Cremers, “FollwMe: Person following and gesture recognition with a quadrocopter”, Proceedings in IEEE/RSJ IROS, pp.624–630, 2013.

[6] E Price, G Lawless, H H Bulthoff, M Black, and A Ahmad, “Deep Neural Network-based Cooperative Visual Tracking through Multiple Micro Aerial Vehicles”, arXiv preprint, arXiv:1820.01346, 2018.

[7] J Wang, W B Chen, and V Temu, “Multi-Vehicle Motion Planning for Search and Tracking.”, IEEE Conference on Multimedia Information Processing and Retrieval (MIPR), pp.352–355, 2018.

[8] M Bajracharya, B Moghaddam, A Howard, S Brennan, and L H Matthies, “A Fast Stereo-

Based System for Detecting and Tracking Pedestrians from a Moving Vehicle.”, The Int’l J. Robotics Research, Vol.28,pp.1466–1485, 2009.

[9] T Furukawa, F Bourgault, B Lavis, and H F Durrant-Whyte, “Recursive Bayesian search-and-tracking using coordinated UAVs for lost targets.”, Proceedings of IEEE International Conference on Robotics and Automation, pp.2521–2526, 2006.

[10] T Furukawa, L C Mak, H Durrant-Whyte, and R Madhavan, “Autonomous bayesian search and tracking. and its experimental validation.’, Advanced Robotics, 26(5-6):pp.461–485, 2012.

[11] Pack D J, P Delima, and G J Toussaint, “Cooperative control of UAVs for localization of intermittently emitting mobile targets”, IEEE Transactions on Systems,Man,and Cybernetics, vol.39,no.4,pp.959–970, 2009.

[12] 佐々木徹 “地上設置 LiDAR と複数 UAV を用いた人物追従システムに関する研究”, 大阪市立大学　修士論文，2019

研究業績

国内発表（査読なし）

- 小林大気, 田窪朋仁, 上野敦志, “離散時間運動学モデルに基づく二足歩行ロボットの 3 次元脚配置計画”, ロボティクス・メカトロニクス講演会 2013 (ROBOMEC2013), 1A1-P11, May 2013.

国際発表（査読あり）

- **Daiki Kobayashi**, Tomohito Takubo, Atsushi Ueno, “3D Gait Planning Based on Discrete-time Kinematic Model of Biped Walking”, The 25th 2014 International Symposium on Micro-Nano Mechatronics and Human Science (MHS2014), Nagoya, Japan, Nov. 2014.

論文誌

- **Daiki Kobayashi**, Tomohito Takubo, Atsushi Ueno, “Model-Based Footstep Planning Method for Biped Walking on 3D Field”, Journal of Robotics and Mechatronics, vol. 27, no. 2, pp. 156–166, Apr. 2015.