摘要

近年，生産の効率化や医療現場での支援などへの需要が高まっているという背景から，人間の作業を代替する二足ロボットが注目されており，多方面での活躍が期待されている．しかし，二足ロボットは自由度が高く制御が複雑である上，四足ロボットや車輪型ロボットなどと比較すると不安定であるため，歩行を継続しつつも転倒を防ぐような高い踏破能力が求められている．現在，多くの歩行制御では簡略化のために着地位置のみの計画や数歩先程度の計画しか行わないといった制約が大きい限定的な手法が多く，計算時間の観点からも歩行中の環境変化に対して十分に対処出来ないという問題がある．本研究では，開始地点から目標地点へと移動する動力学的な安定性を考慮した歩行軌道を，実時間で計画することを目的とする．具体的には，capturability の枠組みで事前に計算した状態遷移をグラフとして表現することで，最短経路計画問題としてゴールに至る最短経路を探索する．提案手法の有効性は，２地点間を直線的に移動する歩行の軌道計画を通じて評価する．

目 次

1　緒言

1.1　研究背景

近年，経済の発展や少子高齢化の進行に伴い，生産の効率化や医療現場における多様な支援が求められている．また，災害時には人が立ち入ると危険な環境下での作業を余儀なくされ，人命救助や捜索活動，復旧作業などが難航する．これらの背景から，人間の作業を代替するロボットへの需要が高まっている．数多くのロボットの中でも二足ロボットは，人間の活動する環境および人間の使用する道具をそのまま適用しやすいという特徴を持ち，さまざまな場面での活躍が期待されている．しかし四足ロボットや車輪型ロボットなどと比較すると，二足ロボットは重心が高く接地面積が小さいため，歩行などの動作中に転倒しやすいという問題点がある．転倒するとロボット自身が壊れるだけでなく，人間に危害を加えてしまうリスクがあるため，歩行を継続しつつも転倒を防ぐような高い踏破能力が求められている．

1.2　従来研究

実環境での二足ロボットを使ったタスクの完遂に向けては，目標地点まで移動する「歩行」と環境変化にも耐えうる「転倒回避」の 2 つの能力が重要となる．

歩行軌道の計画は一般的に計算コストが高いため，まず着地位置を計画し，決定された着地位置から重心軌道を計算するという方法がとられている．Karkowski ら [1] は，凹凸路面上の複数歩にわたる着地点列を 50 ms 程度の時間で決定する手法を提案している．しかし，突然の環境変化や外乱に対しては重心の動きによって安定化できる着地位置が変化するため，予め着地位置を決定する方法では安定な軌道が得られず転倒してしまう恐れがある．

また，着地位置と重心軌道を 1 つの最適化問題として定式化し，軌道を求める手法も存在する [2] が，計画には数十秒から数分程度要するため，環境変化に対してリアルタイムで軌道を修正することは困難である．

歩行だけでなく，転倒回避に関する研究も多くなされている．Stephens[3] は，二足ロボットが未知の外乱を受けた時の転倒回避戦略を大きく三つに分類した．比較的小さな外乱に対しては，足首関節や股関節を屈曲・伸展させて転倒を回避する．大きな外乱に対しては，足を踏み出して転倒を回避するが，踏み出す量は運動学的拘束や外乱の大きさに応じて異なり，重心の動きに応じた踏み出し位置 (特に圧力中心の位置) を求める必要がある．

Pratt ら [4] は，二足ロボットの重心が一定の高さを保ちながら水平方向にのみ運動すると仮定し，ロボットの運動を支持点の真上で停止するための踏み出し位置である instantaneous capture point を提案した．さらに Koolen ら [5] は，複数歩を用いた着地位置の計画をするために capturability という概念を提唱し，実際に capturability に基づいた計画が有用であることを示した．これらの枠組みを用いれば，歩行中に外乱を受けたロボットは踏み出し位置と重心軌道を修正して転倒を回避することが可能となる．

山本ら [6][7] は，capturability に基づき，計画を上位の胴体軌道と下位の着地点列

に分けることで高速に着地位置決定を行う手法を提案している．計画を分割したことで計算時間は 10 ms とリアルタイム性に優れているが，着地点列の変化は重心軌道，すなわち上位ループの胴体軌道に影響を及ぼすため歩行の実行可能性が低くなる恐れがある．また，次に踏み出すべき 1 歩のみを計画しており，2 歩以上を必要とするような外乱への対処が困難である．

Kim ら [8] は，capturability の枠組みで事前に計算した capture region を利用して，複数歩を要するような外乱に対しても高速に軌道計画が可能な手法を提案した．しかし歩行の効率性の観点は無視されており，外乱を受けた後に転倒を回避して停止するという限定された動作にとどまっている．

1.3 研究目的

本研究では，開始地点から目標地点へと移動する動力学的な安定性を考慮した歩行軌道を，実時間で計画することを目的とする．計画において，重心軌道と着地点列は同時に決定されるため，外乱や環境変化に対しても柔軟に適応することが可能となる．具体的には，capturability の枠組みで事前に計算した状態遷移をグラフとして表現することで，ゴールに至る最短経路を探索する．提案手法の有効性は，2 地点間を直線的に移動する歩行の軌道計画を通じて評価する．

第 2 章で，二足ロボットを簡略化したモデルとして用いる三次元線形倒立振子モデルと，倒立振子の挙動を決定する軌道エネルギおよび capturability について説明する．第 3 章で，capturability 解析から得られたデータベースを利用して，重心軌道を含めた歩行軌道を生成する軌道計画器について説明する．第 4 章で，水平路面上に設定された 2 地点間を直線的に移動する軌道計画を行い，想定する制御周期内で計算が可能かどうかを検証する．第 5 章で，まとめと今後の展望について述べる．

2　線形倒立振子モデルと capturability

2.1　三次元線形倒立振子モデル

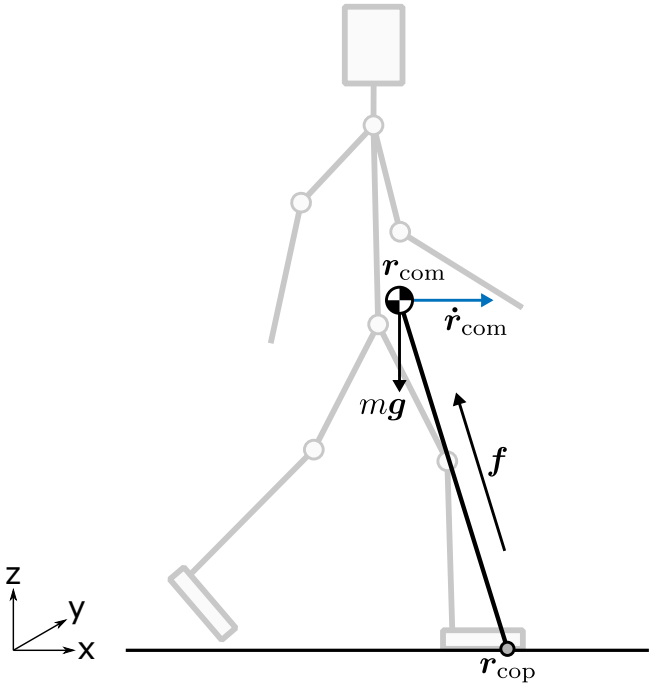


Fig. 1: Linear inverted pendulum model

二足ロボットは自由度が高く制御が複雑であるため，ロボットを簡略化したモデルとして Fig. 1 に示す三次元線形倒立振子モデル [9] を用いる．なお，ロボットの前後方向を x 軸，側面方向を y 軸，鉛直方向を z 軸とする．三次元線形倒立振子モデルでは，質量 m の二足ロボットを床反力の圧力中心 (Center of Mass; CoP) と重心を結ぶ仮想的な倒立振子で表現する．この倒立振子は支持点から重心の方向への床反力 $\boldsymbol{f}$ と重力 $m\boldsymbol{g}$ のみが作用し，支持点まわりにトルクは発生しない．ここで，$\boldsymbol{f} = [f_{\mathrm{x}}\ f_{\mathrm{y}}\ f_{\mathrm{z}}]^{\mathrm{T}}$，$\boldsymbol{g} = [0\ 0\ -g]^{\mathrm{T}}$ である．重心の位置を $\boldsymbol{r}_{\mathrm{com}} = [r_{\mathrm{com,x}}\ r_{\mathrm{com,y}}\ r_{\mathrm{com,z}}]^{\mathrm{T}}$，CoP の位置を $\boldsymbol{r}_{\mathrm{cop}} = [r_{\mathrm{cop,x}}\ r_{\mathrm{cop,y}}\ r_{\mathrm{cop,z}}]^{\mathrm{T}}$ とおけば，倒立振子モデルの重心の運動方程式は次のようになる．

$$m \begin{bmatrix} \ddot{r}_{\mathrm{com,x}} \\ \ddot{r}_{\mathrm{com,y}} \\ \ddot{r}_{\mathrm{com,z}} \end{bmatrix} = \begin{bmatrix} f_{\mathrm{x}} \\ f_{\mathrm{y}} \\ f_{\mathrm{z}} \end{bmatrix} + m \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix} \tag{1}$$

重心が高さ z_0 を保ちながら水平運動する場合，z 方向の床反力 f_{z} は重力とつり合うので，

$$f_{\mathrm{z}} = mg \tag{2}$$

となる．このとき，床反力 $\boldsymbol{f}$ の他の成分は式 (1) と式 (2) より計算できる．

$$f_{\mathrm{x}} = \frac{mg}{z_0}(r_{\mathrm{com,x}} - r_{\mathrm{cop,x}}) \tag{3}$$

$$f_{\mathrm{y}} = \frac{mg}{z_0}(r_{\mathrm{com,y}} - r_{\mathrm{cop,y}}) \tag{4}$$

以上より，式 (1) の運動方程式は次のように書き換えられる．

$$\begin{bmatrix} \ddot{r}_{\mathrm{com,x}} \\ \ddot{r}_{\mathrm{com,y}} \\ \ddot{r}_{\mathrm{com,z}} \end{bmatrix} = \omega_0^2 \begin{bmatrix} r_{\mathrm{com,x}} - r_{\mathrm{cop,x}} \\ r_{\mathrm{com,y}} - r_{\mathrm{cop,y}} \\ 0 \end{bmatrix} \tag{5}$$

ただし，

$$\omega_0 = \sqrt{\frac{g}{z_0}} \tag{6}$$

である．

2.2 軌道エネルギと instantaneous capture point

CoP が時間によらず空間上に固定されている線形倒立振子モデルの運動で保存されるエネルギとして，軌道エネルギ [10] が知られている．ここでは導出法をまとめ，転倒回避や歩行との関連性について述べる．

まず，x 軸方向について考える．式 (5) の $\ddot{r}_{\mathrm{com,x}}$ に対して，両辺に $\dot{r}_{\mathrm{com,x}}$ を乗ずる．

$$\ddot{r}_{\mathrm{com,x}}\dot{r}_{\mathrm{com,x}} = \omega_0^2\dot{r}_{\mathrm{com,x}}(r_{\mathrm{com,x}} - r_{\mathrm{cop,x}}) \tag{7}$$

$$\frac{d}{dt}\left\{\frac{1}{2}\dot{r}_{\mathrm{com,x}}^2 - \frac{1}{2}\omega_0^2(r_{\mathrm{com,x}} - r_{\mathrm{cop,x}})^2\right\} = 0 \tag{8}$$

両辺を時間で積分すると，

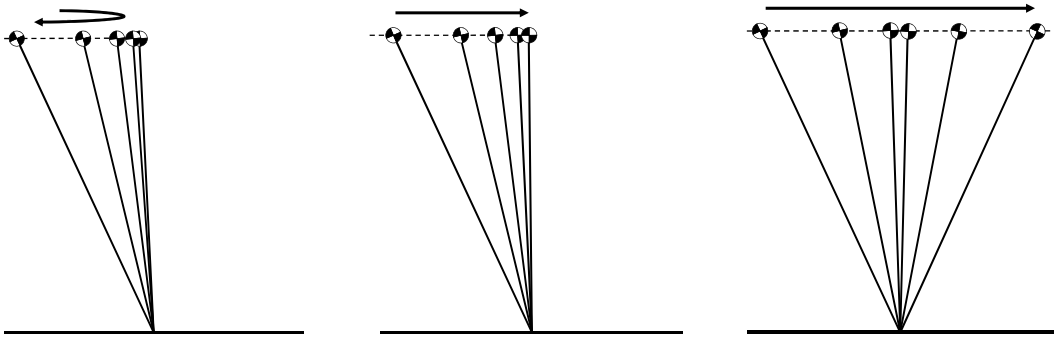
$$\frac{1}{2}\dot{r}_{\mathrm{com,x}}^2 - \frac{1}{2}\omega_0^2(r_{\mathrm{com,x}} - r_{\mathrm{cop,x}})^2 = E \tag{9}$$

を得る．式 (9) の右辺の E が軌道エネルギと呼ばれ，倒立振子の運動では支持点が変わらない限り一定に保たれる．左辺の第一項は単位質量当たりの運動エネルギ，第二項は単位質量当たりのポテンシャルエネルギにそれぞれ対応しており，軌道エネルギはその 2 つのエネルギを足し合わせたものになっている．y 方向も同様に計算すると，次のようになる．

$$E_{\mathrm{LIP,x}} = \frac{1}{2}\dot{r}_{\mathrm{com,x}}^2 - \frac{1}{2}\omega_0^2(r_{\mathrm{com,x}} - r_{\mathrm{cop,x}})^2 \tag{10}$$

$$E_{\mathrm{LIP,y}} = \frac{1}{2}\dot{r}_{\mathrm{com,y}}^2 - \frac{1}{2}\omega_0^2(r_{\mathrm{com,y}} - r_{\mathrm{cop,y}})^2 \tag{11}$$

倒立振子の運動は軌道エネルギー E_{LIP} の符号によって Fig.2 に示すように分類される．軌道エネルギーが負の場合，重心が接地点の真上に達するまでに水平方向の



(a) $E_{\mathrm{LIP}} < 0$ (b) $E_{\mathrm{LIP}} = 0$ (c) $E_{\mathrm{LIP}} > 0$

Fig. 2: Relationship between orbital energy and inverted pendulum motion

重心速度が 0 となり，その後倒立振子は逆方向に運動する．軌道エネルギーが正の場合，接地点の真上を重心が通過するときの重心の水平方向の速度は正となり，その後も倒立振子は同方向に運動し続ける．軌道エネルギーが 0 の場合，接地点の真上で重心の水平方向の速度が 0 となり，倒立振子の運動は停止する．この軌道エネルギを 0 にするような足の着地位置が instantaneous capture point(ICP) となる [11].

ICP を $\boldsymbol{r}_{\mathrm{icp}} = [r_{\mathrm{icp,x}}\ r_{\mathrm{icp,y}}]^{\mathrm{T}} \in \mathbb{R}^2$ とおけば，次式のように表される．

$$r_{\mathrm{icp,x}} = r_{\mathrm{com,x}} + \frac{\dot{r}_{\mathrm{com,x}}}{\omega_0} \tag{12}$$

$$r_{\mathrm{icp,y}} = r_{\mathrm{com,y}} + \frac{\dot{r}_{\mathrm{com,y}}}{\omega_0} \tag{13}$$

式 (12)(13) の両辺を時間で微分すれば次式が得られる．

$$\begin{bmatrix} \dot{r}_{\mathrm{icp,x}} \\ \dot{r}_{\mathrm{icp,y}} \end{bmatrix} = \begin{bmatrix} \dot{r}_{\mathrm{com,x}} \\ \dot{r}_{\mathrm{com,y}} \end{bmatrix} + \frac{1}{\omega_0} \begin{bmatrix} \ddot{r}_{\mathrm{com,x}} \\ \ddot{r}_{\mathrm{com,y}} \end{bmatrix} \tag{14}$$

ここに式 (5) を代入して整理すると，ICP の微分方程式が得られる．

$$\begin{bmatrix} \dot{r}_{\mathrm{icp,x}} \\ \dot{r}_{\mathrm{icp,y}} \end{bmatrix} = \omega_0 \begin{bmatrix} r_{\mathrm{icp,x}} \\ r_{\mathrm{icp,y}} \end{bmatrix} - \omega_0 \begin{bmatrix} r_{\mathrm{cop,x}} \\ r_{\mathrm{cop,y}} \end{bmatrix} \tag{15}$$

式 (15) の微分方程式を解けば，ICP が時間の関数として求まる．

$$\begin{bmatrix} r_{\mathrm{icp,x}}(t) \\ r_{\mathrm{icp,y}}(t) \end{bmatrix} = \begin{bmatrix} r_{\mathrm{icp,x}}(0) - r_{\mathrm{cop,x}} \\ r_{\mathrm{icp,y}}(0) - r_{\mathrm{cop,y}} \end{bmatrix} \mathrm{e}^{\omega_0 t} + \begin{bmatrix} r_{\mathrm{cop,x}} \\ r_{\mathrm{cop,y}} \end{bmatrix} \tag{16}$$

以上より，xy 平面における時刻 t での重心と ICP の関係は Fig. 3 のようになる．ICP は重心の位置から重心速度の方向に伸ばした延長線上にあり，CoP から指数関数的に発散する．

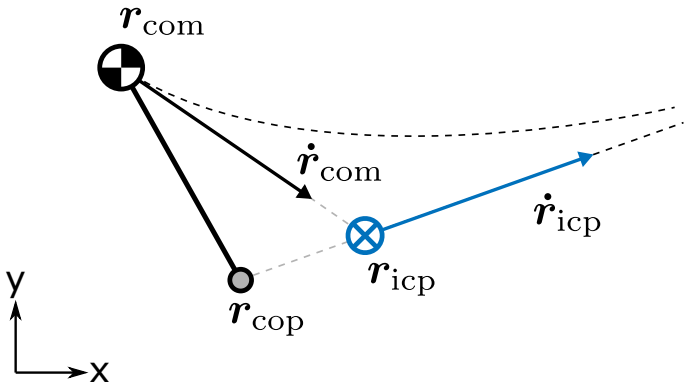


Fig. 3: Top view of the 3D-LIPM with point foot for a given initial state at time t

2.3 capturability

Koolen ら [5] はロボットの安定性を判別する枠組みとして，capturability という概念を提唱した．N-step capturability とは，ロボットが N 歩以下で安定な状態になることができることを表している．どのような動作をしても不安定になり最終的に転倒してしまう状態の集合を $\boldsymbol{X}_{\mathrm{failed}}$ としたとき，N-step capturable および N-step cpture region が次のように定義されている．

N-step capturable

ある状態 $\boldsymbol{x}_0$ が N-step capturable ($N \in \mathbb{N}$)
⇔ 状態 $\boldsymbol{x}_0$ から N 歩以下の踏み出しで $\boldsymbol{X}_{\mathrm{failed}}$ に到達しない状態遷移が
少なくともひとつ存在する

N-step capture point

ある平面上の点 $\boldsymbol{r}$ が N-step capture point ($N > 0$)
⇔ 状態 $\boldsymbol{x}_0$ から 1 歩の踏み出しで $\boldsymbol{X}_{\mathrm{failed}}$ に到達せず，
かつ $(N-1)$-step capturable になる状態遷移が少なくともひとつ存在する

N-step capture region

N-step capture point の集合

これらの関係を図示すると Fig. 4 のようになる．

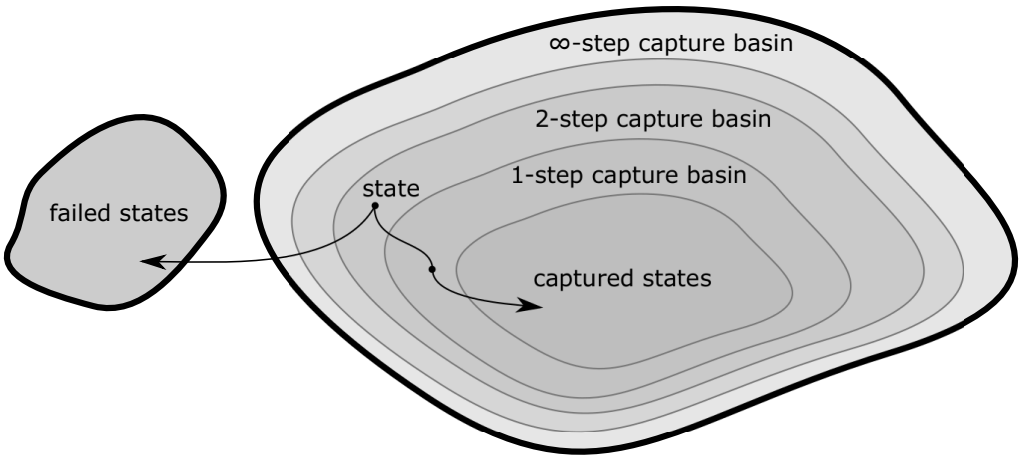


Fig. 4: Capturability framework

3　capturability 解析と歩行計画

3.1　軌道計画器の概要

本研究では，実時間で歩行軌道を計画するために 2 段階に分けた計画を行う．提案手法の概念図を Fig. 5 に示す．

1 段階目の計画は capturability 解析である．まず，十分な摩擦があり，障害物のない水平面上を想定して capturability 解析を行い，capture region を求める．解析では，N-step capturable な状態に対して，入力となる次の踏み出し位置に足を踏み出した際に $(N-1)$-step capturable な状態になるか，不安定な状態になるかを判別する．$(N-1)$-step capturable な状態になる入力が capture region であるので，データベースとして保存しておく．

2 段階目の計算では解析結果を基に着地点列を決定する．はじめに，実環境からの情報を基に，capture region のデータベースにフィルタをかける．一般的には capture region を候補点として着地点の選択を行うが，障害物がある場合は着地できないということになるので，候補点から除外する．そして，残った候補点の中から評価値が最小となるような踏み出し位置を決定する．

それぞれの計算のタイミングであるが，1 段階目の計算はオフラインで処理し，2 段階目の計算はオフラインで処理する．両者をオンラインで計算することも考えられるが，一般に 1 段階目の計算コストは大きいため，実時間での軌道計画が困難となる．以上の方法により，リアルタイムでの軌道計画が可能となる．また，制御周期以内に複数歩にわたる軌道計画を実現できる．

KIM ら [8] はロボットの足を円に近似し，支持足を基準とした離散化して解析を行った．本研究はその手法を足が多角形の場合にも適用できるように拡張したものであり，より現実のロボットに近い条件で解析・計画を行う．以降では，制御対象の状態と入力について説明した後,「解析」と「計画」の方法を詳細に述べる．

3.2　状態空間モデル

解析に用いる足の形状を Fig. 6 に示す．Fig. 6 の足は多角形で左右対称となっており，中心は足首位置である．また，単脚支持期および両脚支持期での支持多角形を Fig. 7 に示す．単脚支持期では片方の足の多角形の凸包となり，両脚支持期では

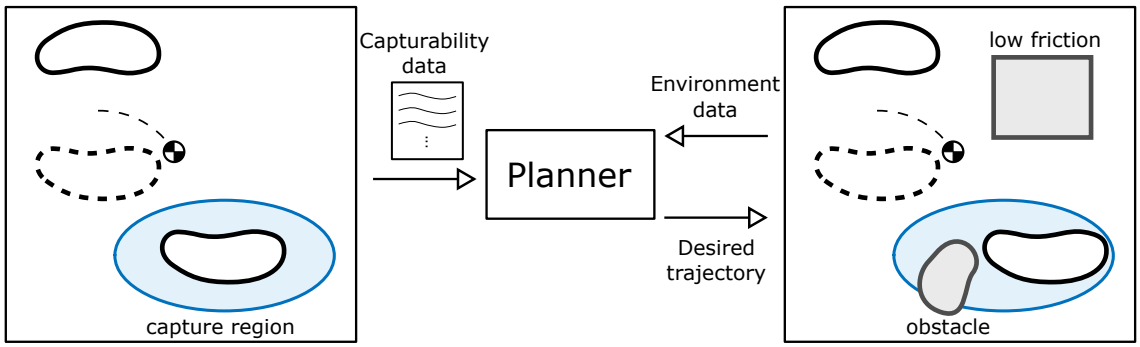


Fig. 5: Conceptual diagram of the proposed method

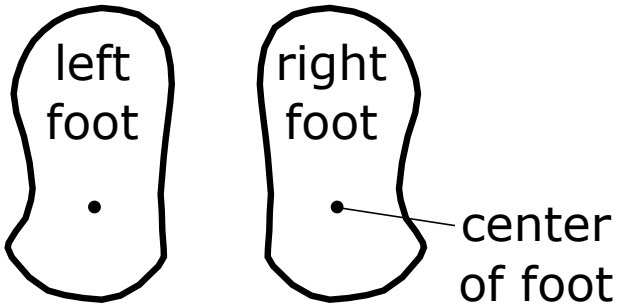
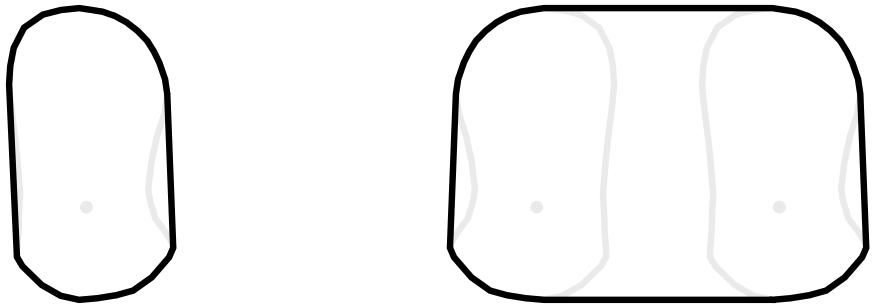


Fig. 6: Foot model



(a) single support phase (b) double support phase

Fig. 7: Support region

両方の足の多角形を内包する凸包となる．

本研究での単脚支持，両脚支持の扱いは次のようになっている．歩行中は単脚支持期のみとし，支持脚交換の際に CoP が瞬時に切り替わるものとする．単脚支持期開始時における CoP の位置を Fig. 8 に示す．ICP が支持多角形 R_{sup} の外部にある場合は，CoP は ICP に対する支持多角形上の最近点とする (Fig. 8a)．支持多角形の内部にある場合は，CoP は足の中心とする (Fig. 8b)．特に ICP が足の中心にある場合は，CoP と ICP が一致するため，ICP は初期位置から動かないことになる (Fig. 8c)．一方，歩行は両脚支持期で終了するものとする．両脚支持期開始時における CoP の位置を Fig. 9 に示す．先ほどと同様に，ICP が支持多角形 R_{sup} の外部にある場合は，CoP は ICP に対する支持多角形上の最近点とする (Fig. 9a)．支持多角形の内部にある場合は，CoP は ICP に一致するよう設定する (Fig. 9b)．

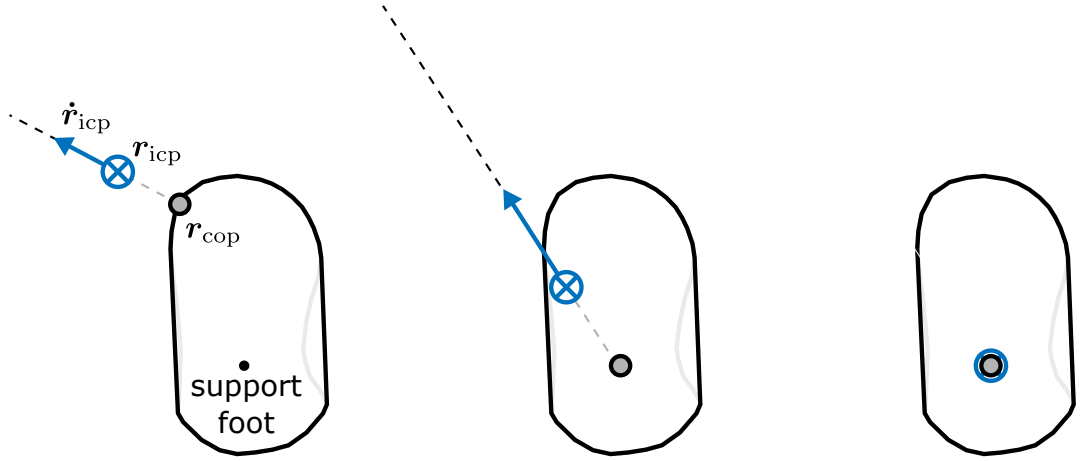
k 歩目における状態変数を次式で定義する．

$$\boldsymbol{x}_{\mathrm{k}} = \begin{bmatrix} \boldsymbol{d}_{\mathrm{icp,k}} \\ \boldsymbol{d}_{\mathrm{sw,k}} \end{bmatrix} \in \mathbb{R}^4 \tag{17}$$

ここで，$\boldsymbol{d}_{\mathrm{icp,k}} = [d_{\mathrm{icp,k,x}}\ d_{\mathrm{icp,k,y}}]^{\mathrm{T}}$ および $\boldsymbol{d}_{\mathrm{sw,k}} = [d_{\mathrm{sw,k,x}}\ d_{\mathrm{sw,k,y}}]^{\mathrm{T}}$ は，それぞれ支持足を基準とした ICP の位置，遊脚の足首位置である．併せて，制御入力を次のように定義する．

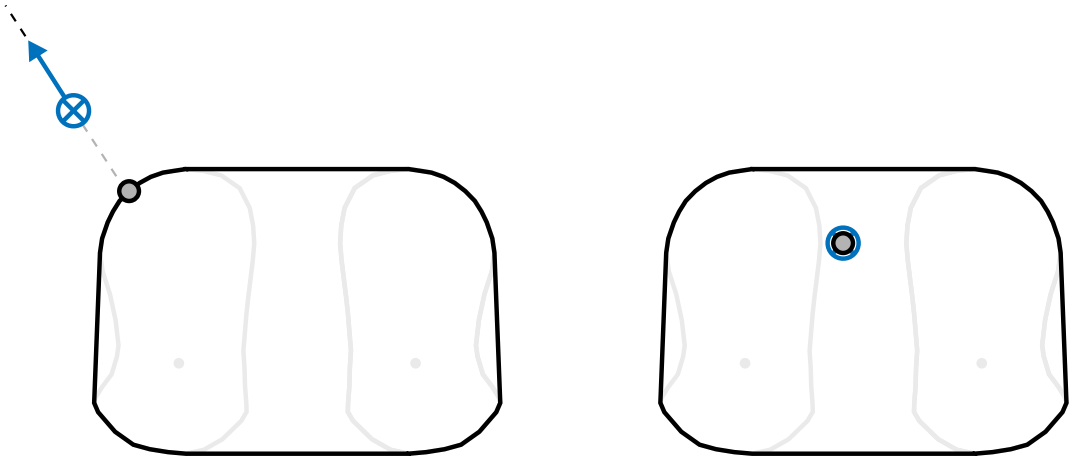
$$\boldsymbol{u}_{\mathrm{k}} = \begin{bmatrix} u_{\mathrm{k,x}} \\ u_{\mathrm{k,y}} \end{bmatrix} \in \mathbb{R}^2 \tag{18}$$

制御入力は，次の着地位置である．



(a) $\boldsymbol{r}_{\mathrm{icp}} \notin R_{\mathrm{sup}}$ (b) $\boldsymbol{r}_{\mathrm{icp}} \in R_{\mathrm{sup}}$ (c) $\boldsymbol{r}_{\mathrm{icp}} = \boldsymbol{r}_{\mathrm{cop}}$

Fig. 8: CoP approximation while walking



(a) $\boldsymbol{r}_{\mathrm{icp}} \notin R_{\mathrm{sup}}$ (b) $\boldsymbol{r}_{\mathrm{icp}} \in R_{\mathrm{sup}}$

Fig. 9: CoP approximation when stop walking

状態と入力の関係を図示すると Fig. 10 となる．支持足を基準とした直交座標系で，進行方向を+x 軸方向，側面方向のうち遊脚が存在する側を+y 軸方向にとる．Fig. 10 は右足が支持足の場合であるが，左足が支持足の場合も同様に座標系をとれば Fig. 10 のようになる．

また，遊脚は Fig. 11 の軌道で動くとする．Fig. 11 は，開始点から足を真上に上げ，等速 (移動可能な最高速度) で次の着地点の真上へと移動した後，足を真下に下ろすという動作を表している．足の上げ下ろしにかかる時間を Δt_{min}，足の速さの最大値を v_{max} とすれば，遊脚の移動にかかる時間 τ が次式で求まる．

$$\tau = \frac{\|\boldsymbol{u}_{\mathrm{k}} - \boldsymbol{d}_{\mathrm{sw,k}}\|}{v_{\mathrm{max}}} + \Delta t_{\mathrm{min}} \tag{19}$$

ここまで「状態」と「入力」について説明したが，capturability 解析をするにあたり支持脚交換時の状態の切り替わりを導出する必要がある．現在の支持足を基準

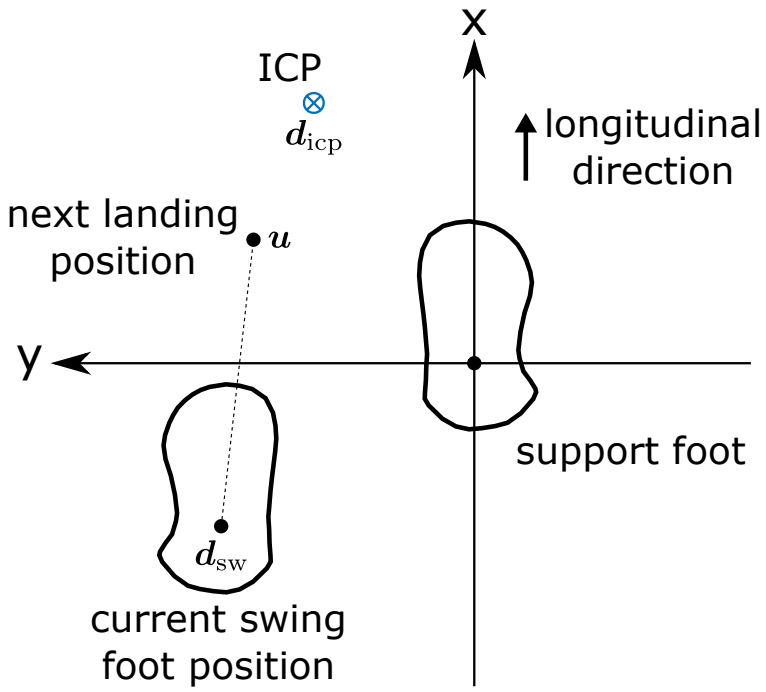


Fig. 10: State space model

とした踏み出し後の状態を

$$\hat{\boldsymbol{x}}_{\mathrm{k}} = \begin{bmatrix} \hat{\boldsymbol{d}}_{\mathrm{icp,k}} \\ \hat{\boldsymbol{d}}_{\mathrm{sw,k}} \end{bmatrix} \in \mathbb{R}^4 \tag{20}$$

とおけば，式 (16) および式 (17)-(19) から求まる．

$$\hat{d}_{\mathrm{icp,k,x}} = (r_{\mathrm{icp,k,x}} - r_{\mathrm{cop,k,x}})\mathrm{e}^{\omega_0 \tau} + r_{\mathrm{cop,k,x}} \tag{21}$$
$$\hat{d}_{\mathrm{icp,k,y}} = (r_{\mathrm{icp,k,y}} - r_{\mathrm{cop,k,y}})\mathrm{e}^{\omega_0 \tau} + r_{\mathrm{cop,k,y}} \tag{22}$$
$$\hat{d}_{\mathrm{sw,k,x}} = u_{\mathrm{k,x}} \tag{23}$$
$$\hat{d}_{\mathrm{sw,k,y}} = u_{\mathrm{k,y}} \tag{24}$$

次の支持足を基準とした踏み出し後の状態は，Fig. 12 より幾何学的に求めることができる．

$$d_{\mathrm{icp,k+1,x}} = \hat{d}_{\mathrm{icp,k,x}} - \hat{d}_{\mathrm{sw,k,x}} \tag{25}$$
$$d_{\mathrm{icp,k+1,y}} = -\hat{d}_{\mathrm{icp,k,y}} + \hat{d}_{\mathrm{sw,k,y}} \tag{26}$$
$$d_{\mathrm{sw,k+1,x}} = -\hat{d}_{\mathrm{sw,k,x}} \tag{27}$$
$$d_{\mathrm{sw,k+1,y}} = \hat{d}_{\mathrm{sw,k,y}} \tag{28}$$

以降では，上記の写像を $f()$ と表記する．

$$\boldsymbol{x}_{\mathrm{k+1}} = f(\boldsymbol{x}_{\mathrm{k}}, \boldsymbol{u}_{\mathrm{k}}) \tag{29}$$

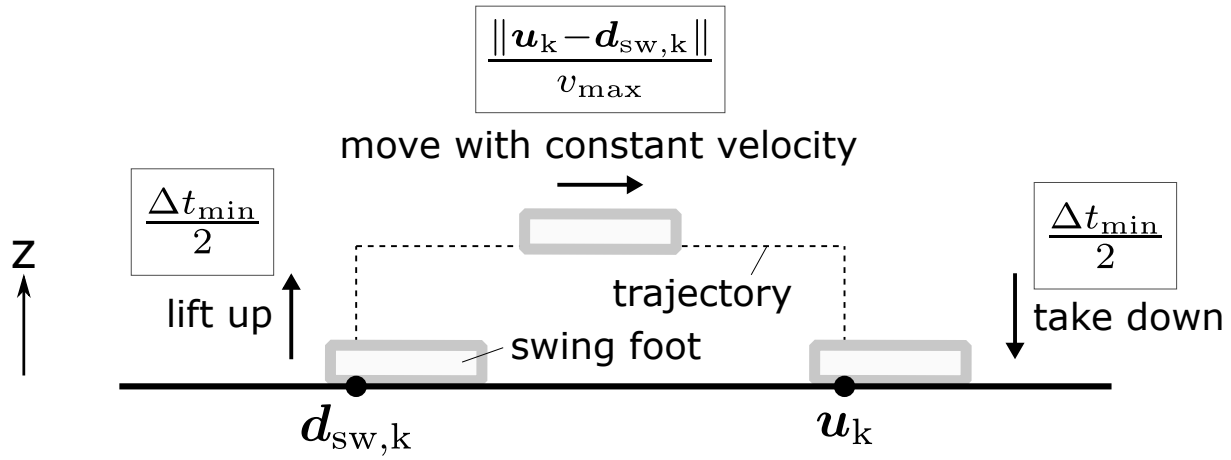


Fig. 11: Trajectory of swing foot

3.3 capturability 解析

外乱等により不安定になったロボットが転倒を回避するには，ICP が支持多角形内に存在しなければならないため，ロボットは何歩か踏み出す必要がある．しかし，遊脚の速度などの制約があるので，対応可能な ICP には限界がある．この限界は ∞-step viable capture basin と呼ばれ，ここでは $\mathcal{P}_\infty$ と表記する．N-step viable capture basin は N が大きくなるにつれて ∞-step viable capture basin に収束する．具体的には，以下のように計算される．

ICP が支持多角形内に存在する場合は，CoP を ICP に一致させるだけで踏み出すことなく安定な状態になる．そのため，0-step viable capture basin $\mathcal{P}_0$ は次のようになる．

$$\mathcal{P}_0 = \{\boldsymbol{x} \,|\, r_{\mathrm{icp}} \in R_{\mathrm{sup}}\} \tag{30}$$

適当な入力によって $\mathcal{P}_0$ になる状態の集合を $\mathcal{P}_1$ とすれば，$\mathcal{P}_0$ から $\mathcal{P}_1$ を計算することができる．

$$\mathcal{P}_1 = \{\boldsymbol{x} \,|\, \exists \boldsymbol{u} \in U \text{ s.t. } f(\boldsymbol{x}, \boldsymbol{u}) \in P_0\} \tag{31}$$

再帰的に，N-step viable capture basin $\mathcal{P}_{\mathrm{N}}$ が次式で計算される．

$$\mathcal{P}_{\mathrm{N}} = \{\boldsymbol{x} \,|\, \exists \boldsymbol{u} \in U \text{ s.t. } f(\boldsymbol{x}, \boldsymbol{u}) \in P_{\mathrm{N}-1}\} \tag{32}$$

N-step viable capture region は，状態 $\boldsymbol{x}$ から $\mathcal{P}_{\mathrm{N}-1}$ に遷移させる踏み出し位置の集合である．ここでは入力 $\boldsymbol{u}$ が踏み出し位置であるので，N-step viable capture region は実行可能な入力の集合として表現される．

$$\mathcal{U}_{\mathrm{N}}(\boldsymbol{x}) = \{\boldsymbol{u} \,|\, \boldsymbol{u} \in U,\, f(\boldsymbol{x}, \boldsymbol{u}) \in \mathcal{P}_{\mathrm{N}-1}\} \tag{33}$$

N-step viable capture region の計算は，KIM ら [8] が提案したアルゴリズムを用いる．極座標系と直交座標系という点で異なるが，状態を下限と上限の間で離散化すれば同様に計算可能である．まず，状態と入力を Table 1 に示すように離散化する．ただし，状態の遊脚位置と入力の着地位置は同じ最小値・最大値・分割数とするので，Fig. 13 のようになる．

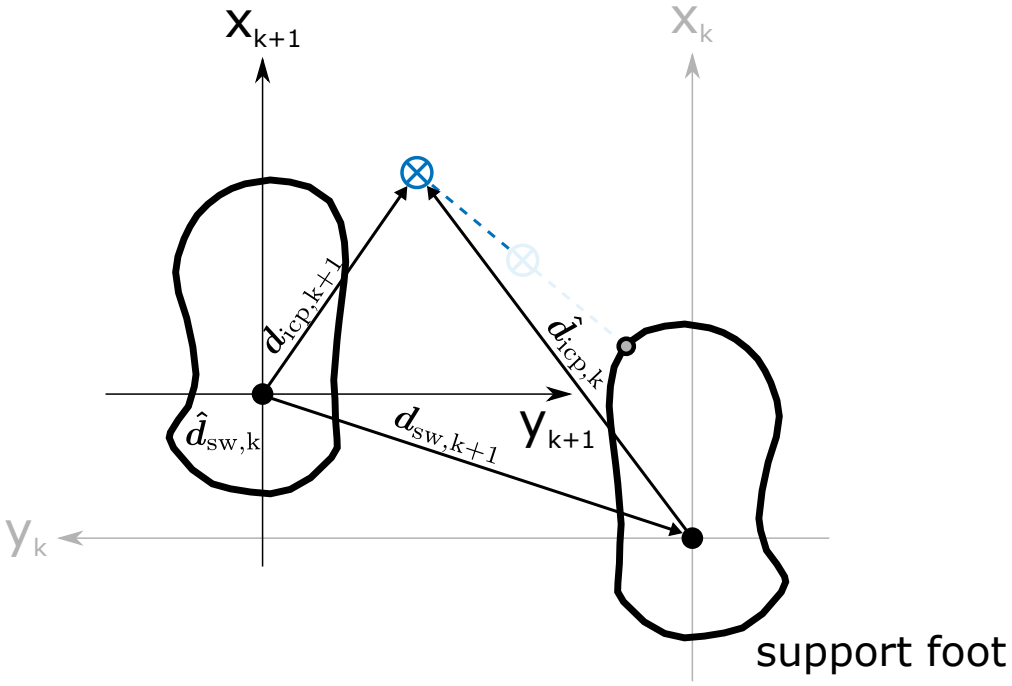


Fig. 12: Conversion of state due to the support leg exchange

Table 1: Discretization of the state space and the input space

	Variable	Min	Max	Resolution
State	$d_{\mathrm{icp,x}}$	$d_{\mathrm{icp,x,max}}$	$d_{\mathrm{icp,x,min}}$	$\Delta d_{\mathrm{icp,x}}$
	$d_{\mathrm{icp,y}}$	$d_{\mathrm{icp,y,max}}$	$d_{\mathrm{icp,y,min}}$	$\Delta d_{\mathrm{icp,y}}$
	$d_{\mathrm{sw,x}}$	$d_{\mathrm{sw,x,max}}$	$d_{\mathrm{sw,x,min}}$	$\Delta d_{\mathrm{sw,x}}$
	$d_{\mathrm{sw,y}}$	$d_{\mathrm{sw,y,max}}$	$d_{\mathrm{sw,y,min}}$	$\Delta d_{\mathrm{sw,y}}$
Input	u_{x}	$u_{\mathrm{x,max}}$	$u_{\mathrm{x,min}}$	Δu_{x}
	u_{y}	$u_{\mathrm{y,max}}$	$u_{\mathrm{y,min}}$	Δu_{y}

3.4 軌道計画器

この節では，capturability 解析の結果を経路探索問題として扱うための方法を説明した後，探索法の詳細について述べる．

capturability 解析の結果として得られるのは，状態 $\boldsymbol{x}$ と入力 $\boldsymbol{u}$ の組である．その結果と軌道計画との関連を，具体例を挙げて説明する．capturability 解析によって得られた結果を模式的に表した図が，Fig. 14 である．初めの状態 $\boldsymbol{x}_0$ が 2-step capturable で，位置 p_0 に支持足がある場合を考える．状態 $\boldsymbol{x}_0$ から 1-step capturable な状態 $\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3$ に遷移する踏み出し位置をそれぞれ $\boldsymbol{p}_1, \boldsymbol{p}_2, \boldsymbol{p}_3$(capture region 内の踏み出し位置) とする．ここで $\boldsymbol{p}_2$ に踏み出した場合は状態が $\boldsymbol{x}_2$ に遷移し，さらに安定な状態 $\boldsymbol{x}_5$ に遷移する踏み出し位置 $\boldsymbol{p}_4$ が存在する．一方，$\boldsymbol{p}_3$ に踏み出した場合は状態が $\boldsymbol{x}_3$ に遷移し，さらに $\boldsymbol{p}_5$ に踏み出すことで状態 $\boldsymbol{x}_4$ に遷移する．状態 $\boldsymbol{x}_4$ から $\boldsymbol{p}_6$ に踏み出した時に再び状態 $\boldsymbol{x}_3$ に戻ったとすると，2 つの状態間を交互に遷移し続ける入力が存在することになる．この場合が定常歩行に相当し，適当な踏み出し位置を選

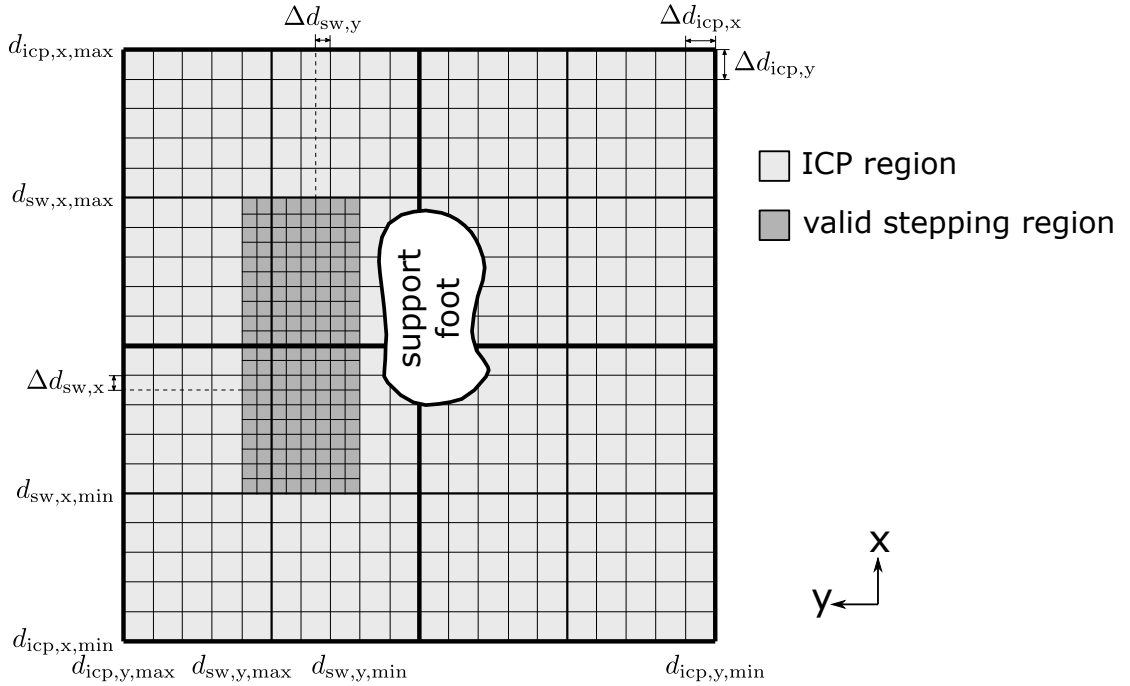


Fig. 13: Discretization of the state space and the input space

べば前に進み続けることができる．

そこで，ノード $\boldsymbol{n}$ を次式で定義する．

$$\boldsymbol{n} = \{\boldsymbol{x}, \boldsymbol{p}, p_{\mathrm{sw}}\} \tag{34}$$

ただし，$\boldsymbol{p}$ はグローバル座標系での着地位置，p_{sw} は遊脚の左右である．

$$p_{\mathrm{sw}} = \begin{cases} \text{left} & (\text{if “swing foot” = “left foot”}) \\ \text{right} & (\text{if “swing foot” = “right foot”}) \end{cases} \tag{35}$$

ノード間はエッジで結合され，Fig. 15 で表されるものとする．エッジは踏み出しによる状態遷移を表し，必ず一方向のみの結合となる．以上のようにノードおよびエッジを定義すれば，ノードの遷移を表すグラフは Fig. 16 となる．軌道計画においてはスタート位置が始点ノード $\boldsymbol{n}_0$，ゴール位置が終点ノード $\boldsymbol{n}_{\mathrm{l}}, \boldsymbol{n}_{\mathrm{r}}$ になる．ここで，$\boldsymbol{n}_{\mathrm{l}}$ は左足のゴール位置，$\boldsymbol{n}_{\mathrm{r}}$ は右足のゴール位置である．capturability 解析の結果を使った軌道計画が，グラフの経路探索問題として扱えることになる．以降では，始点ノードから終点ノードに至るまでの最短経路を探索する方法について説明する．

グラフの最短経路を探索するためのアルゴリズムは数多く研究されてきたが，ここでは評価関数に従って次に探索する最も望ましいノードを選択する探索アルゴリズムである A*アルゴリズム [12] を用いる．A*アルゴリズムでは，評価関数 $f(n)$ を始点ノードからノード $\boldsymbol{n}_i$ までのコスト $g(n)$ とノード $\boldsymbol{n}_i$ から終点ノードまでのコストの推定値 $h(\boldsymbol{n}_i)$ の和として表す (Fig. 17)．

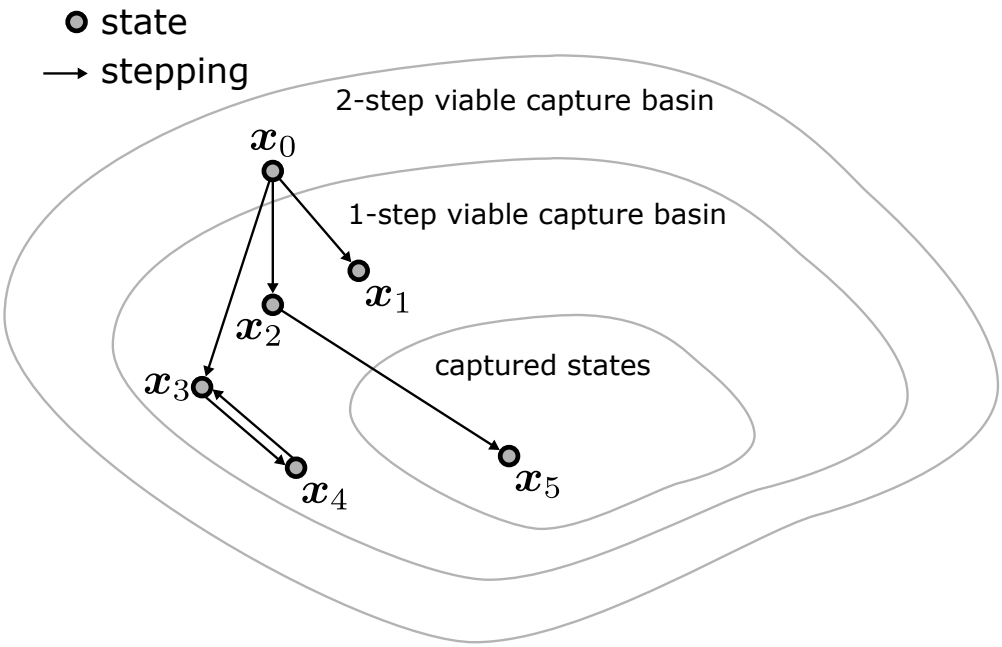
$$f(\boldsymbol{n}_i) = g(\boldsymbol{n}_i) + h(\boldsymbol{n}_i) \tag{36}$$

$g(\boldsymbol{n}_i)$ と $h(\boldsymbol{n}_i)$ についてはユークリッド距離を用いる．

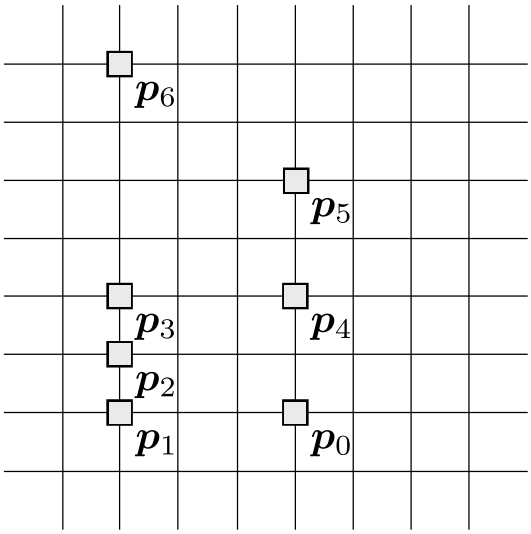
$$g(\boldsymbol{n}_i) = \sum_{j=1}^{i} \|\boldsymbol{p}_j - \boldsymbol{p}_{j-1}\| \tag{37}$$

$$h(\boldsymbol{n}_i) = \begin{cases} a \cdot \|\boldsymbol{p}_{\mathrm{l}} - \boldsymbol{p}_i\| & (\text{if } p_{\mathrm{sw}} = \text{left}) \\ a \cdot \|\boldsymbol{p}_{\mathrm{r}} - \boldsymbol{p}_i\| & (\text{if } p_{\mathrm{sw}} = \text{right}) \end{cases} \tag{38}$$

ここで，$\boldsymbol{p}_i$ はノード $\boldsymbol{n}_i$ の成分であるグローバル座標系での着地位置，$\boldsymbol{p}_{\mathrm{l}}, \boldsymbol{p}_{\mathrm{r}}$ は，左足および右足の終点位置，$a \in \mathbb{R}$ はヒューリスティックコストのスケールを調整する定数である．$h(n)$ が実コスト $h^*(n)$ より小さければ解の最適性が保証されるが，大きければ解の最適性が保証されない．



(a) state space



(b) landing position

Fig. 14: The interpretation of the analysis results

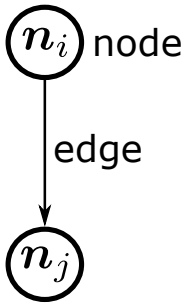


Fig. 15: Node and edge

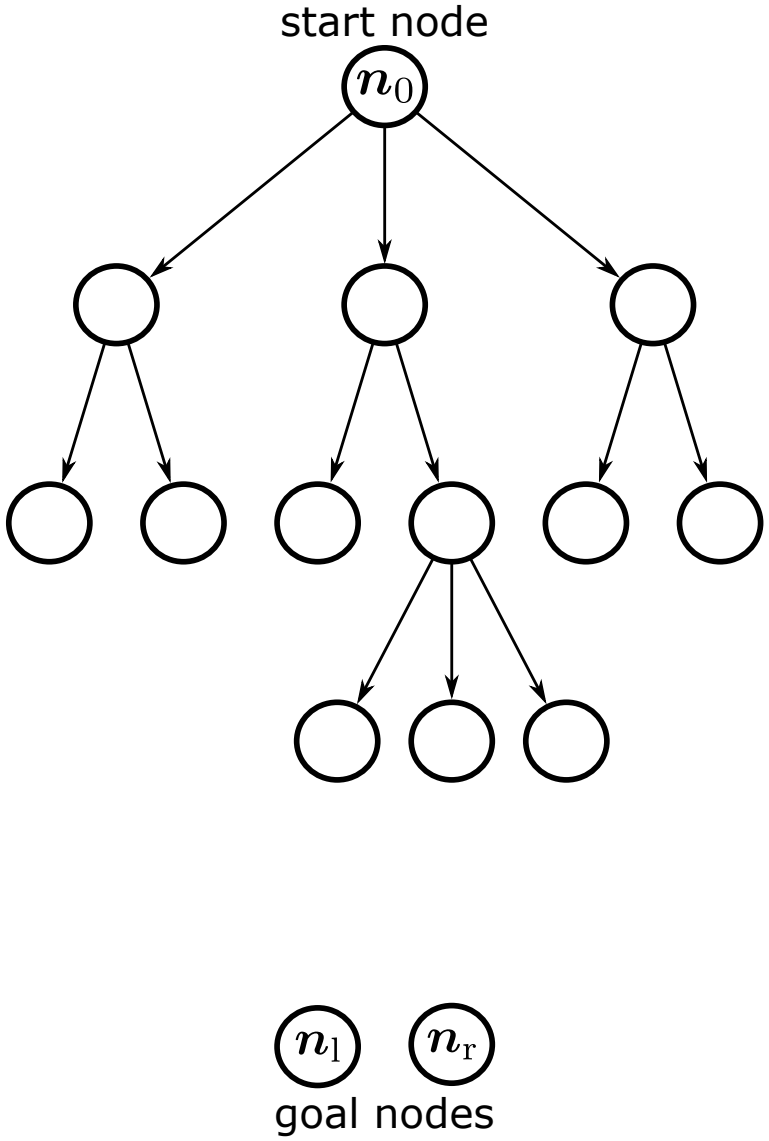


Fig. 16: Graph showing node transitions

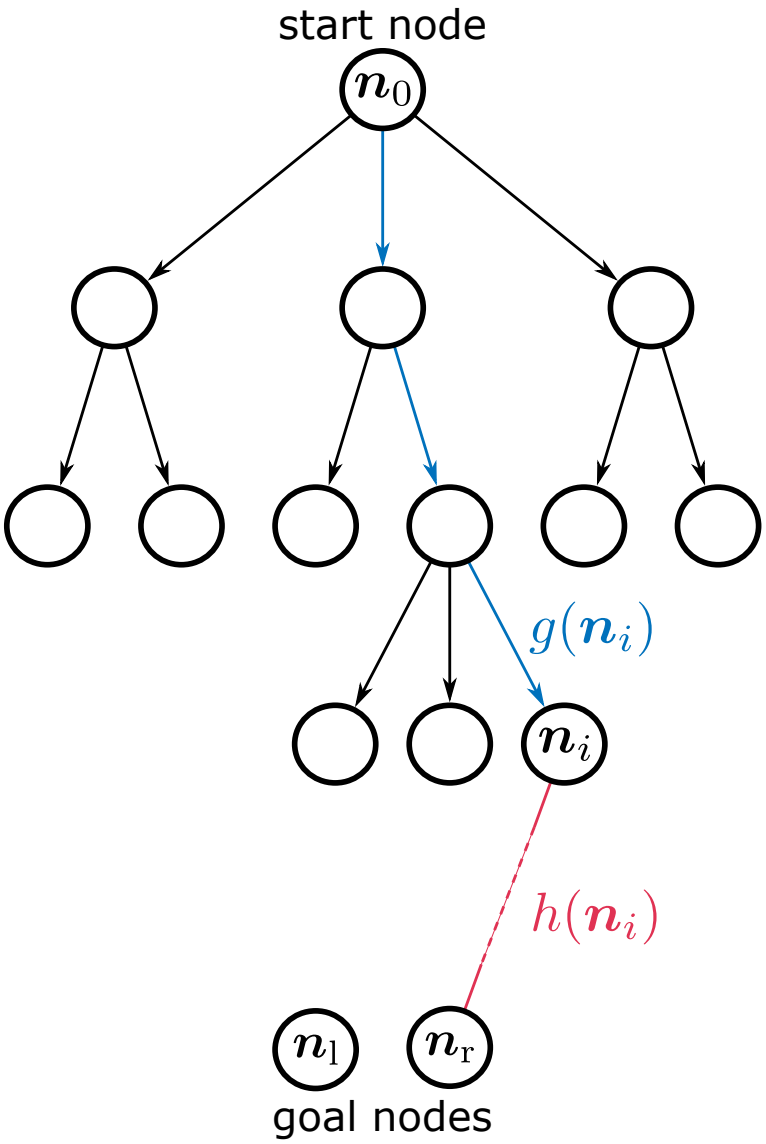


Fig. 17: Moving cost and heuristics cost

4　数値計算による有効性の検証

4.1　計算に用いるパラメータ

提案手法の有効性を検証するために，水平路面に対してスタート地点とゴール地点を設定し，軌道計画に要した時間を比較する．

まず，解析に用いるロボットのパラメータを以下に示す．すべてのパラメータは，Aldebaran Robotics 社製のヒューマノイドロボット NAO を参考に決定した．ロボットの足は Fig. 18 に示すような形状となっており，足の輪郭の内側に足形状を近似した十角形を作成し，それを解析に用いる．足の多角形の各頂点の座標は Table 2 に示すとおりである．併せて，ICP の存在範囲，遊脚の到達可能領域を Table 3 および Fig. 19 に示す．重心の高さや遊脚の速度などのその他の解析に用いるパラメータについては，Table 4 にまとめる．

次に，軌道計画を行うスタート地点とゴール地点を Fig .20 に示すように設定する．スタート地点及びゴール地点の右側が右足に，左側が左足に対応する．グローバル座標系の原点はスタート地点での両足の中点とし，前方を x 軸方向，左側面を y 軸方向にとる．足幅間隔 w は 0.12 m とする．この条件の下で，前方への移動距離 d を 0.1 m から 0.5 m まで 0.1 m 刻みで設定し，移動距離と計算時間の相関を調べる．右足・左足のどちらかがゴール地点に到達すれば探索を終了する．また，移動距離と併せてヒューリスティック関数のスケールを表す変数 a の値を変更し，計算時間に及ぼす影響を調査する．

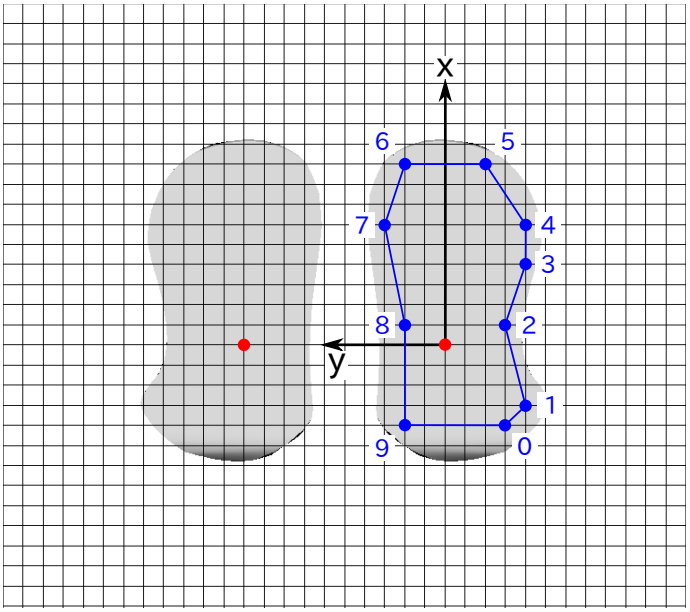


Fig. 18: The geometric structure of foot

Table 2: The geometry parameter of foot

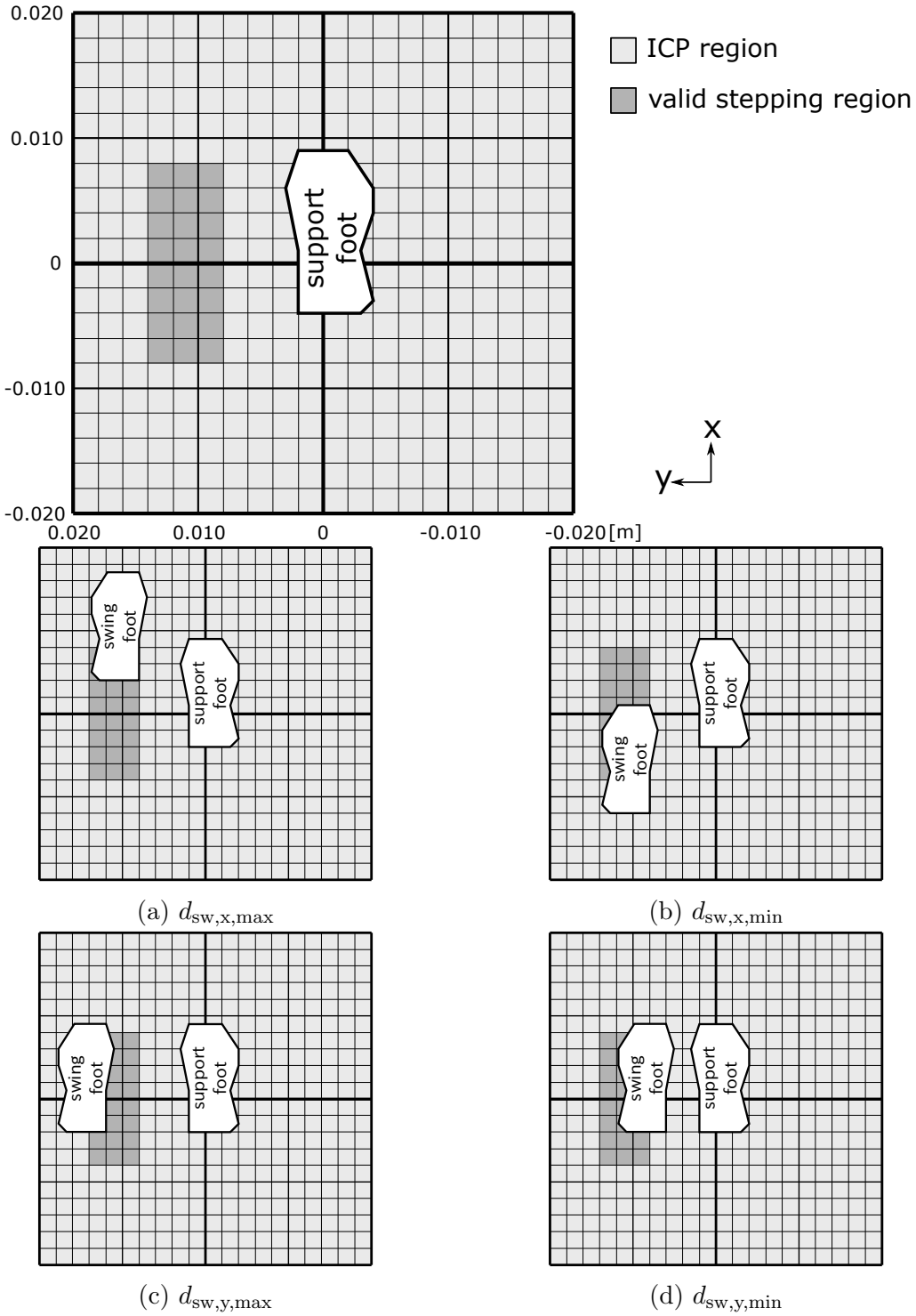
Point No.	Foot origin coordinate x	y
0	-0.04	-0.03
1	-0.03	-0.04
2	0.01	-0.03
3	0.04	-0.04
4	0.06	-0.04
5	0.09	-0.02
6	0.09	0.02
7	0.06	0.03
8	0.01	0.02
9	-0.04	0.02

Table 3: The value of the discrete state

Name	Min	Max	Resolution
$d_{\mathrm{icp,x}}$	-0.200	0.200	0.020
$d_{\mathrm{icp,y}}$	-0.200	0.200	0.020
$d_{\mathrm{sw,x}}$	-0.080	0.080	0.020
$d_{\mathrm{sw,y}}$	0.080	0.140	0.020

Table 4: Parameters used for capturability computation

Variable	Explanation	Value
g	gravitational acceleration [m/s^2]	9.81
z_0	the height of the center of mass [m]	0.3
$v_{\max}$	swing foot velocity [m/s]	0.5
Δt	minimum step duration [s]	0.1



(a) $d_{\mathrm{sw,x,max}}$ (b) $d_{\mathrm{sw,x,min}}$

(c) $d_{\mathrm{sw,y,max}}$ (d) $d_{\mathrm{sw,y,min}}$

Fig. 19: ICP region and valid stepping region used for the calculation

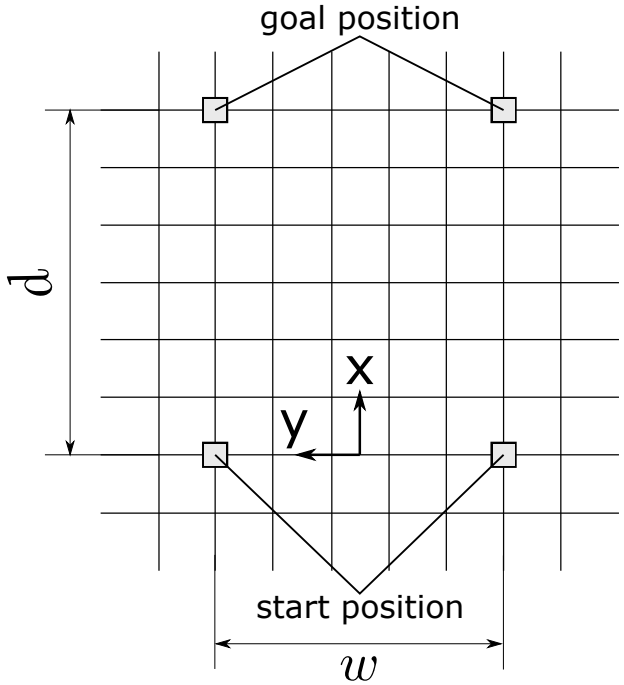


Fig. 20: The start position and the goal position

4.2 計算結果

移動距離と計算時間の相関を Fig. 21 に示す．ただし，$a = 1, 2$ で $d = 0.3, 0.4, 0.5$ の時には 1 分以上計算しても解が得られなかったため，グラフから除外している．また，提案手法は制御周期 10 ms 程度のロボットに実装することを想定しているため，10 ms の線を黒破線で表している．全体的に，距離が大きくなればなるほど計算時間も増える傾向にあることが分かった．10 ms 以内に計算できたのは，0.3 m 以下の時のみで，0.3 m 以上では 10 ms を大きく超える場合も存在した．$a = 1$ から $a = 4$ とヒューリスティック関数の重みを大きくすると計算時間は短くなるが，$a = 4$ から $a = 5$ にした際には，逆に計算時間が増えてしまった．この原因は，ゴールに近いノードから探索しすぎたためである．一般的な踏み出し可能な領域が常に一定の領域に固定されている計画問題ではこのような問題は起こらないが，提案手法では踏み出し可能な領域が均一でないために起こったと考えられる．

次に，Fig. 22 に展開されたノード数と計算時間の関係を示す．ノード数が増えるほど計算時間も大きくなり，ノード数が 10^4 を超えたあたりからは線形な関係がみられる．計算時間を 10 ms 以下にするためには，$10^{3.2} \simeq 1600$ 程度のノード数で解を見つけなければならないことが分かった．少ないノード数で探索するためには，状態の分割幅を大きくすることが考えられるが，単に粗い探索をしただけでは解が見つからない場合も存在する恐れがある．粗い探索と細かい探索を組み合わせるなどの探索方法を改善することが今後の課題である．

また，得られた結果のうち，代表的な軌道を xy 座標系で可視化したものが Fig. 23 から Fig. 25 である．Fig. 23 から Fig. 25 は，それぞれ $d = 0.1, 0.2, 0.3$ としたときの軌道であり，左図が着地点の軌跡，右図がノードの広がりである．比較するために，同じ移動距離で計算時間が最大のもの (上図) と最小のもの (下図) を示している．

ヒューリスティックコストのスケール a の変更による着地点の軌跡の変化は見られなかったが，探索時間が短い場合ほど空間的なノードの広がりも小さくなった．障害物等が存在して着地できない位置が存在する場合は，ノードの広がりが計算時間を左右する重要な要因となる．なるべく空間的にノードを広げないことも計算の高速化には有効であると考えられる．

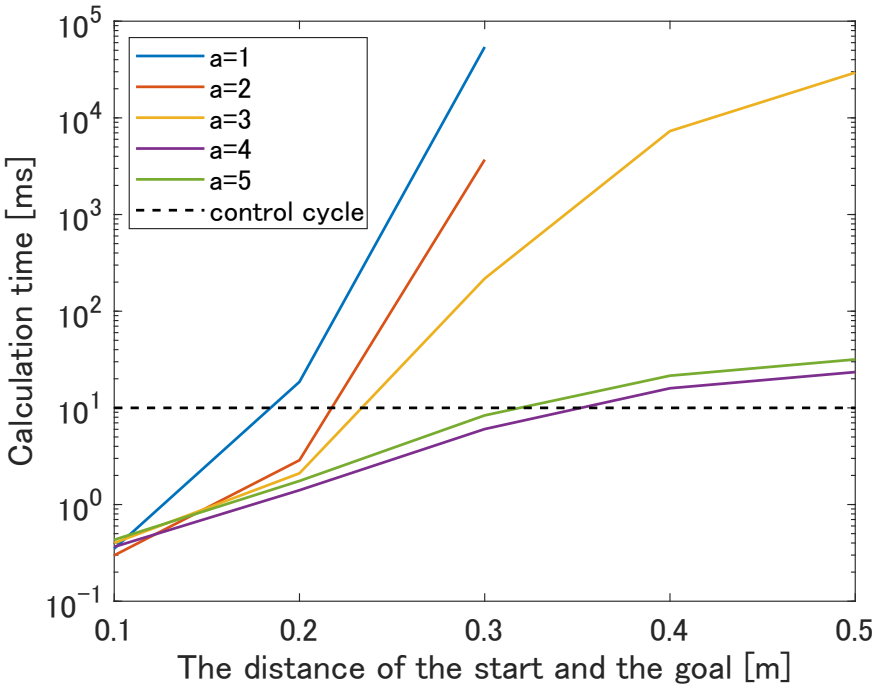


Fig. 21: The relationship between moving distance and calculation time

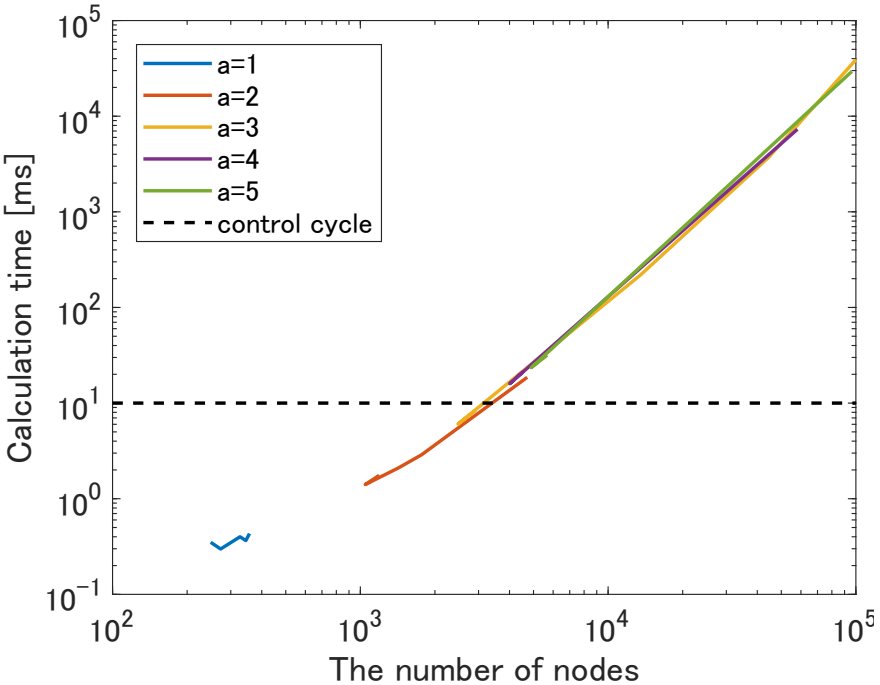
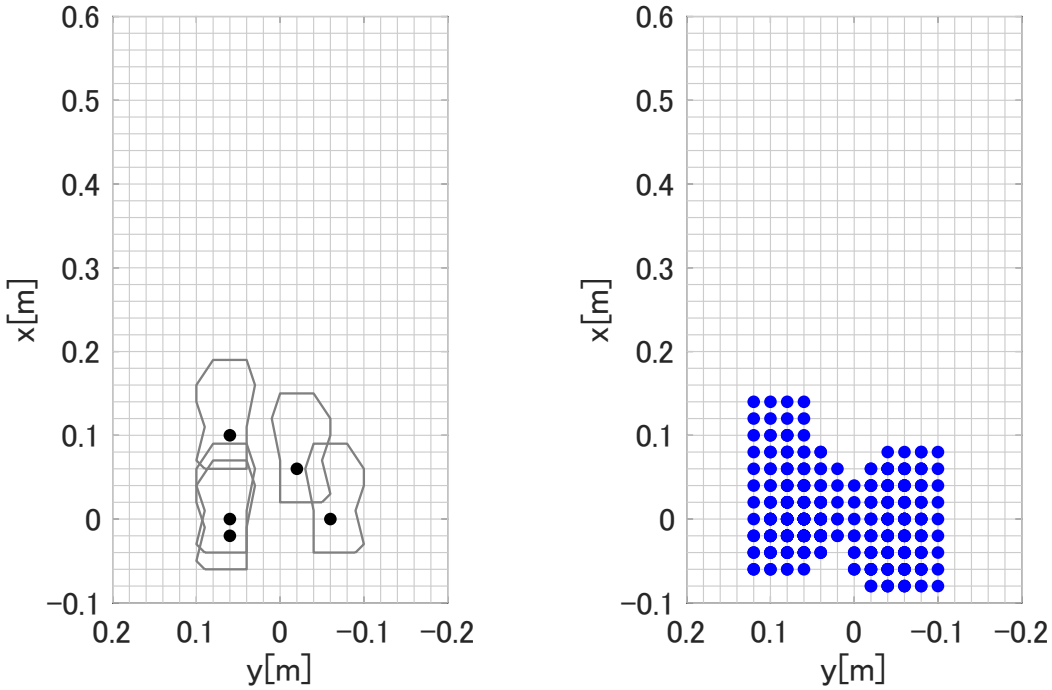
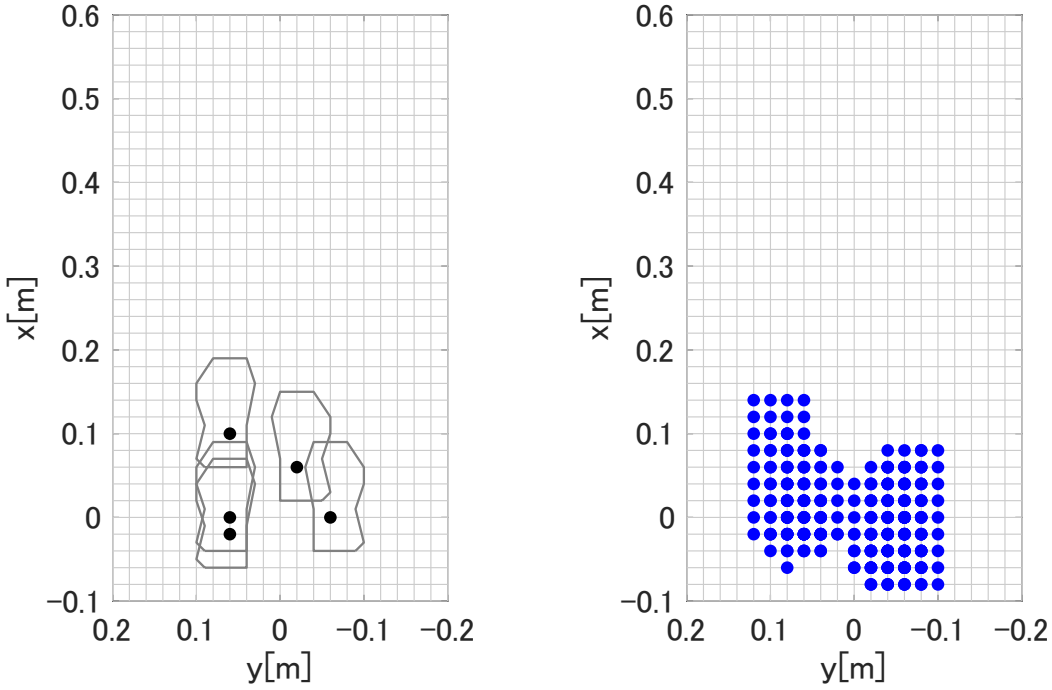


Fig. 22: The relationship between the number of nodes and calculation time



(a) footstep($a = 5$)

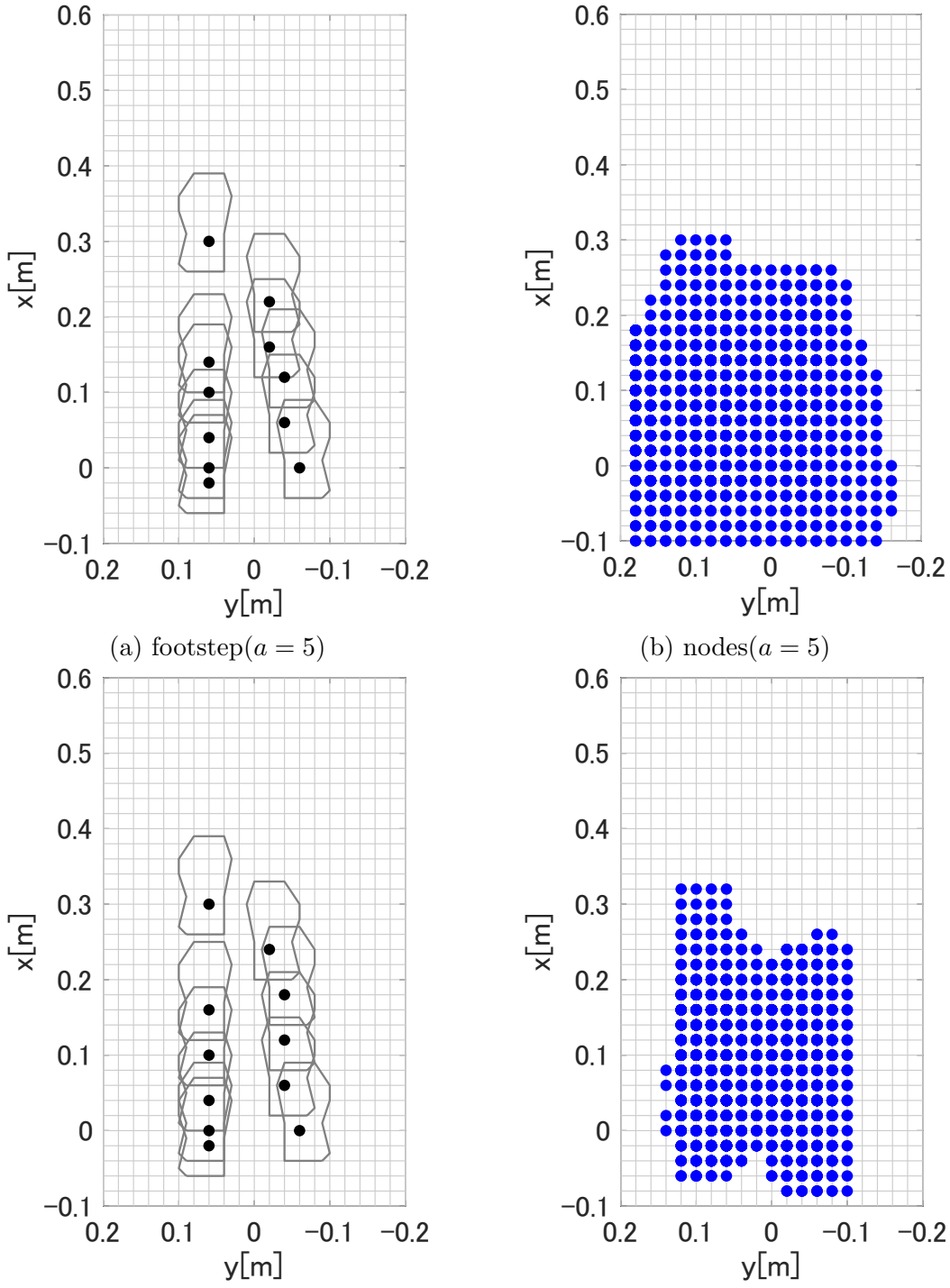
(b) nodes($a = 5$)



(c) footstep($a = 2$)

(d) nodes($a = 2$)

Fig. 23: Moving distance $d = 0.1$



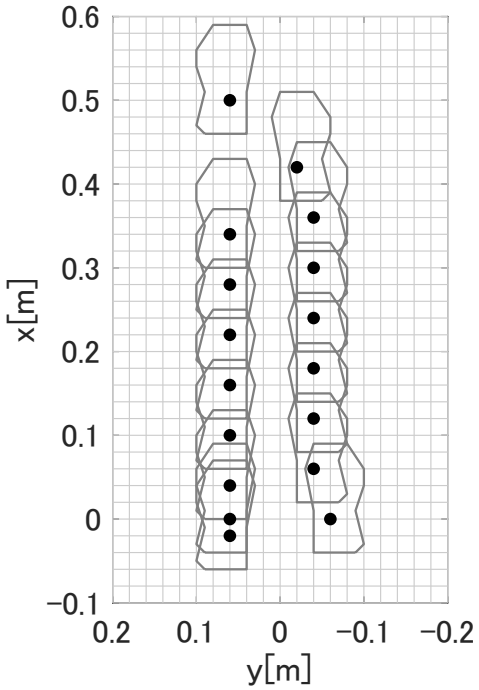
(a) footstep($a = 5$)

(b) nodes($a = 5$)

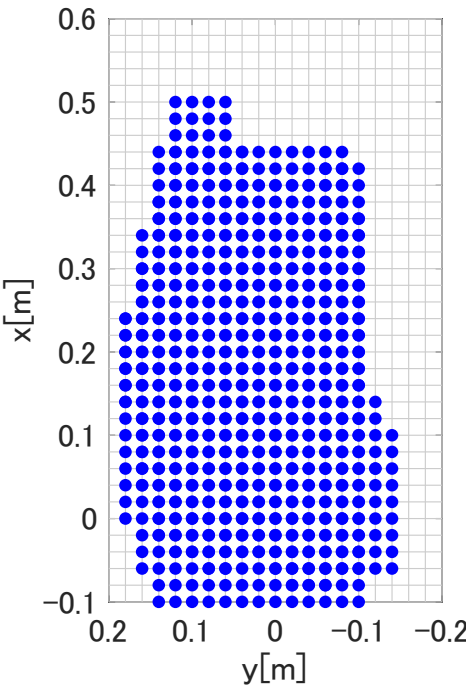
(c) footstep($a = 2$)

(d) nodes($a = 2$)

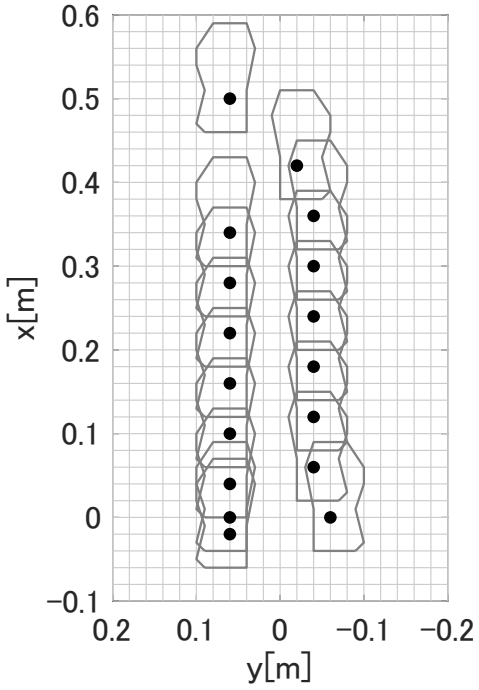
Fig. 24: Moving distance $d = 0.2$



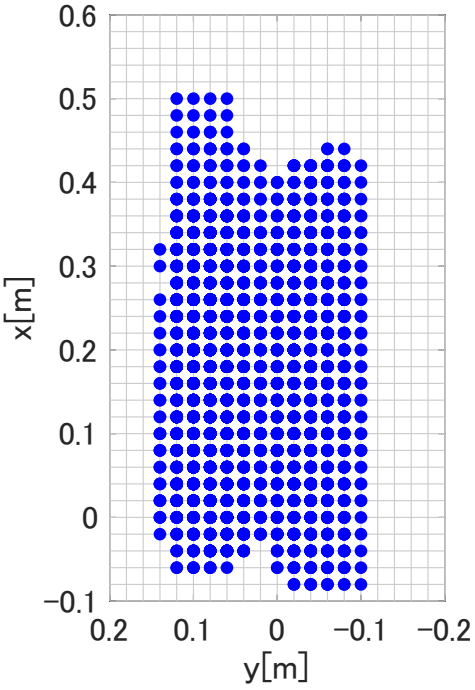
(a) footstep($a = 5$)



(b) nodes($a = 5$)



(c) footstep($a = 2$)



(d) nodes($a = 2$)

Fig. 25: Moving distance $d = 0.3$

5　結言

5.1　まとめ

本研究では，歩行軌道の実時間計画を目的として，歩行計画を capturability の枠組みで事前に計算した状態遷移を用いた経路探索問題として定式化した．その結果，開始地点から目標地点へと移動する動力学的な安定性を考慮した歩行軌道が，実時間で計画可能であることが示された．計画に要する時間は，スタート地点からゴール地点までの距離およびヒューリスティックコストのスケールに大きく依存することが分かった．

5.2　今後の展望

今後の課題として，直線移動以外の様々な環境設定に対しても実時間で軌道が生成できるように，アルゴリズムを改良することが挙げられる．併せて，障害物の存在や外乱の印加に対して提案手法の歩行計画が有効であるかを検証する予定である．また，数値シミュレーションおよび実機を用いた実験を行い，提案する計画手法が実際のロボットに対して有効であることを示す必要がある．

参考文献

[1] P. Karkowski, S. Oßwald, and M. Bennewitz: “Real-time footstep planning in 3D environments”, 2016 IEEE-RAS 16th International Conference on Humanoid Robots, pp. 69-74, 2016.

[2] R. Deits and R. Tedrake: “Footstep Planning on Uneven Terrain with Mixed-Integer Convex Optimization”, 2014 IEEE-RAS 14th International Conference on Humanoid Robots, pp. 279-286, 2014.

[3] B. Stephens: “Humanoid Push Recovery”, 2007 IEEE-RAS 7th International Conference on Humanoid Robots, pp. 589-595, 2007.

[4] J. Pratt, J. Carff, S. Drakunov, and A. Goswami: “Capture point: A step toward humanoid push recovery”, 2006 IEEE-RAS 6th International Conference on Humanoid Robots, pp. 200-207, 2006.

[5] T. Koolen, T. de Boer, J. Rebula, A. Goswami, and J. Pratt: “Capturability-Based Analysis and Control of Legged Locomotion, Part 1: Theory and Application to Three Simple Gait Models”, The International Journal of Robotics Research, Vol.31, pp.1094-1113, 2012.

[6] 山本 孝信, 杉原 知道：“地形・運動学的制約・Capturability を考慮した二脚ロボットの即時的着地位置決定”, 日本機械学会ロボティクス・メカトロニクス講演会講演概要集, 2A2-H07, 2018.

[7] 山本 孝信, 杉原 知道：“外界や目的地の変動に低感度な二脚ロボットの移動経路計画”, 第 36 回日本ロボット学会学術講演会予稿集, 1J2-06, 2018.

[8] G. Kim, H. Kuribayashi, Y. Tazaki, and Y. Yokokohji: “Omni-Directional Fall Avoidance of Bipedal Robots with Variable Stride Length and Step Duration”, 2018 IEEE-RAS 18th International Conference on Humanoid Robots, pp. 718-724, 2018.

[9] S. Kajita, F. Kanehiro, K. Kaneko, K. Yokoi, and H. Hirukawa: “The 3D Linear Inverted Pendulum Mode: A simple modeling for a biped walking pattern generation”, 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 239-246, 2001.

[10] 梶田 秀司: “ヒューマノイドロボット”, オーム社, 2005.

[11] J. Pratt, J. Carff, S. Drakunov, and A. Goswami: “Capture point: A step toward humanoid push recovery”, 2006 IEEE-RAS 6th International Conference on Humanoid Robots, pp. 200-207, 2006.

[12] P. E. Hart, N. J. Nilsson, and B. Raphael: “A Formal Basis for the Heuristic Determination of Minimum Cost Path”, IEEE Transactions on System Science and Cybernetics, Vol. 4, No. 2, pp. 100-107, 1968.