

# インテリジェントロボットモーション レポート

## 軌道計画と軌道生成

Path planning and Trajectory Generation

2023 年 7 月 19 日 提出

千葉工業大学 先進工学部 未来ロボティクス学科

20C1117 望月悠矢



# 概要

本レポートでは、インテリジェントロボットモーション第 11 回「軌道計画と軌道生成」の解説を行う。マニピュレータロボットや移動型ロボットなどがある地点から目標地点まで動かすときにニュートン・オイラー法やラグランジュ法を用いて導出した運動学に基づいてマニピュレータが出すトルクや力を計算するが、このように計算された力は各時刻の力を表すものであり、ロボットは軌道を描くことはできず、ロボットの通るべき経路は考慮されない。したがって、ロボットの通るべき「軌道」を「計画」し、その軌道上をロボットが追従するように出すべき力を連続な力で表し、「軌道」を「生成」する必要がある。はじめに軌道計画（経路計画）と軌道計画の違いと概念について解説し、次に軌道計画の主要なアルゴリズムの基礎的な理論について簡単に紹介する。最後に bang-bang 制御による軌道生成の解説を行う。

キーワード: ロボット, 経路計画, 軌道生成

# 目次

第 1 章	経路計画と軌道生成	1
1.1	第 1 章の導入 . . . . .	1
1.1.1	経路と軌道の違い . . . . .	3
1.1.2	経路計画とは . . . . .	4
1.1.3	軌道生成とは . . . . .	5
1.2	第 1 章のまとめ . . . . .	7
第 2 章	経路計画のアルゴリズム	8
2.1	第 2 章の導入 . . . . .	8
2.1.1	グラフ探索 . . . . .	9
2.1.2	ポテンシャル法 . . . . .	10
2.1.3	RRM ( Random Road Map ) . . . . .	11
2.2	第 2 章のまとめ . . . . .	12
第 3 章	軌道生成の例題	13
3.1	第 3 章の導入 . . . . .	13
3.1.1	連続性のない制御 . . . . .	14
3.1.2	bang-bang 制御 . . . . .	16
3.2	第 3 章のまとめ . . . . .	21
参考文献		22

# 目次

1.1	Configuration space . . . . .	2
1.2	Path planning . . . . .	4
1.3	Trajectory Generation . . . . .	5
2.1	Connected graph . . . . .	9
2.2	Potential field [1] . . . . .	10
2.3	RRM . . . . .	11
3.1	No continuity . . . . .	14
3.2	bang-bang control . . . . .	16
3.3	bang-bang control with additional velocity constraints . . . . .	19

# 第 1 章

## 経路計画と軌道生成

### 1.1 第 1 章の導入

運動学を導出し適用するだけでは，ロボットを適切に動作させることができない．制御をするには人間が意図したとおりに動いてほしい．そのために用いられる方法が軌道計画（経路計画）と軌道生成である．運動学だけでは適切に動作させることができないというのは，運動学はロボットが出す各時刻の力しか計算できないからである．今回扱う軌道生成では各時刻の力ではなく，連続した時間の力を計算することができる．これによりロボットに連続的な動作をさせることができる．その上で，ロボットが障害物などを回避し，ある地点から目的の地点までの軌道を決定するのが軌道計画（経路計画）である．

軌道計画（経路計画）と軌道生成には大きな違いがある．その違いについて，使用される英単語の違いから見ていく．その後に経路計画と軌道生成についての解説を行う．

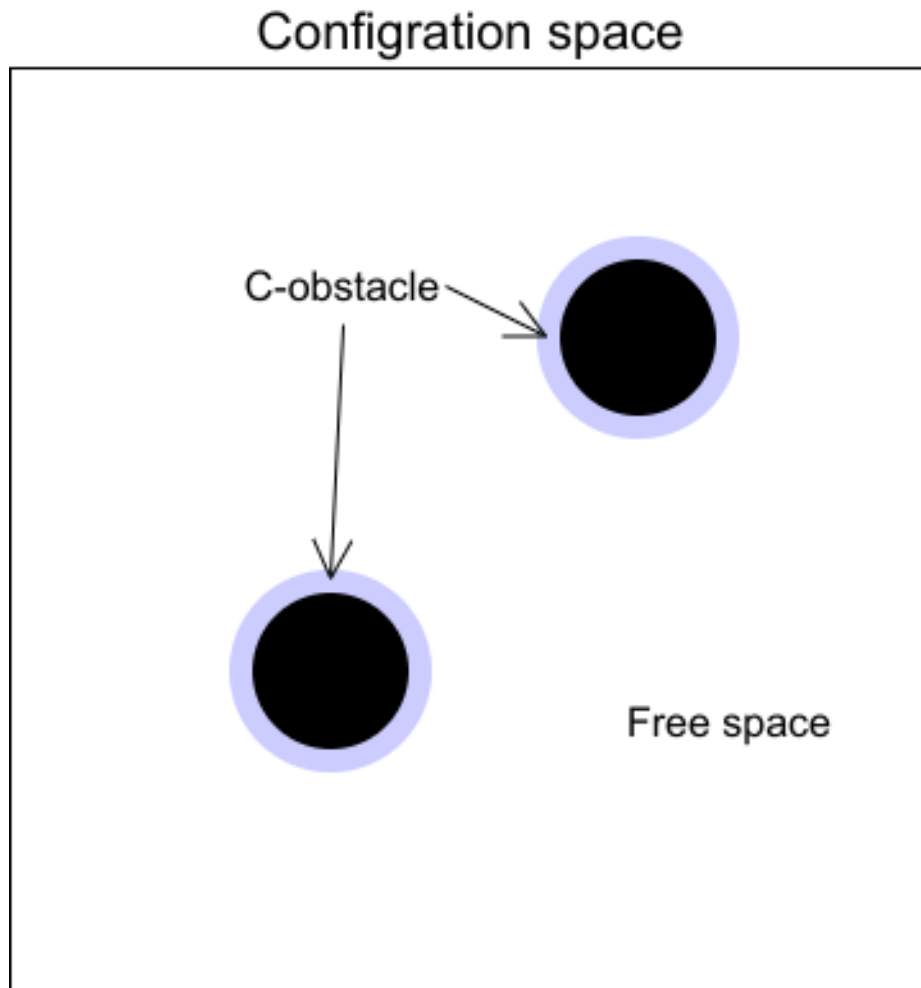


Fig. 1.1 Configuration space

経路計画では基本的にコンフィグレーション空間を使用する．コンフィグレーション空間内にある障害物を C-obstacle と呼ぶ．C-obstacle は実際の空間の障害物に対応しており，障害物のギリギリに経路が計画されるのを防ぐため，Fig.1.1 の水色の円のように実際の大きさよりも膨張させて用いられることが多い．

### 1.1.1 経路と軌道の違い

軌道に対応する英単語として”Path”と”Trajectory”という単語が挙げられる．この 2 つの英単語は運動の計画において違う意味を持ち，その意味は以下ようになる．

- Path：ロボットがどこを通るか
- Trajectory：ロボットがどのような運動で通るか

このように，”Path”と”Trajectory”には違いがあり，これらを日本語で再定義すると，”Path”は「経路」，”Trajectory”は「軌道」とすることができる．そして，ロボットがどこを通るかを計画することを”Path planning”，ロボットがどのような運動で通るかを決めることを”Trajectory generation”と呼ぶ．以降，区別を明確にするために”Path planning”を「経路計画」，”Trajectory generation”を「軌道生成」と呼ぶことにする．



## 1.1.2 経路計画とは

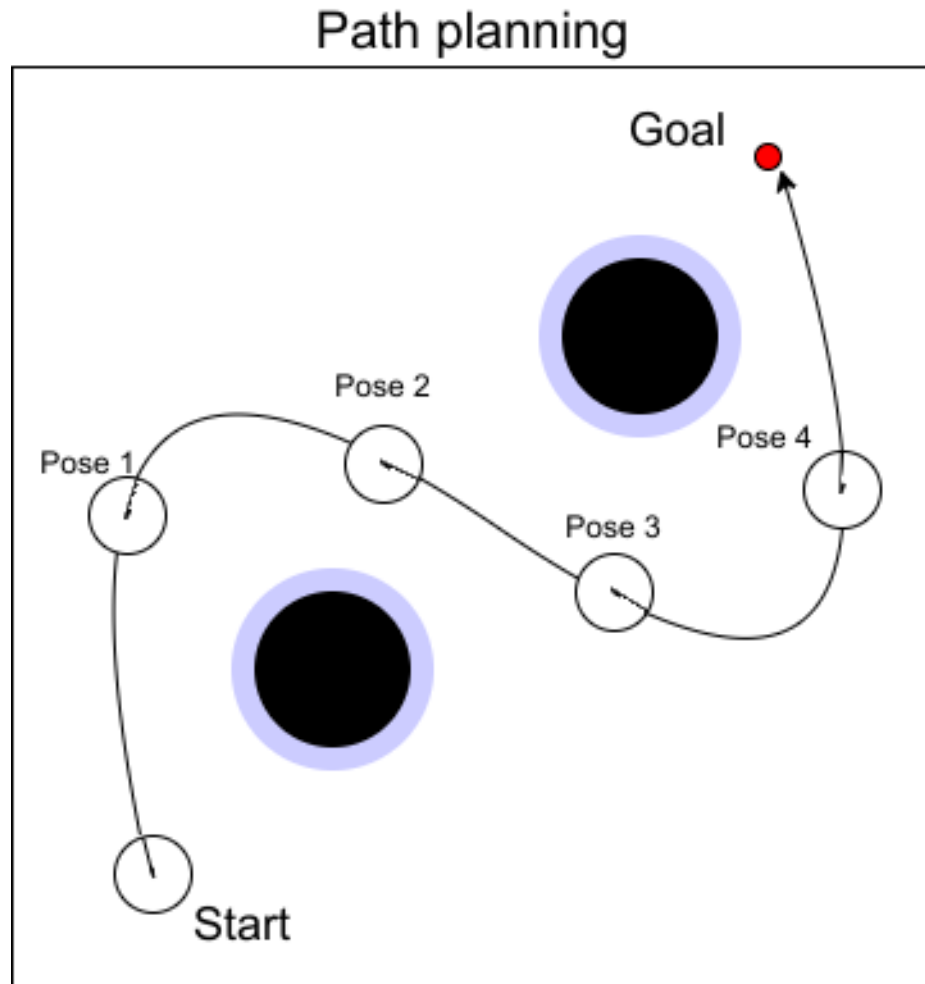


Fig. 1.2 Path planning

経路計画は前述したように、ロボットがどのように通るかを計画することである。経路計画では、ロボットの姿勢と位置、スタートからゴールまでにある障害物が考慮されたスタート地点からゴールまでの最短な経路が計画される。計画された経路はロボットが通るべき位置、そのときに取るべき姿勢とその地点を通る順番の 3 つの条件からなる複数の点の集合であると言える。

Fig.1.2 は経路計画のイメージ図である。黒と青の円で表されているのが障害物であり、円と線分で表されているのがロボットである。線分の向きがロボットの姿勢を表している。Fig.1.2 のように経路計画では、ロボットの姿勢と位置が順番に指定される。

## 1.1.3 軌道生成とは

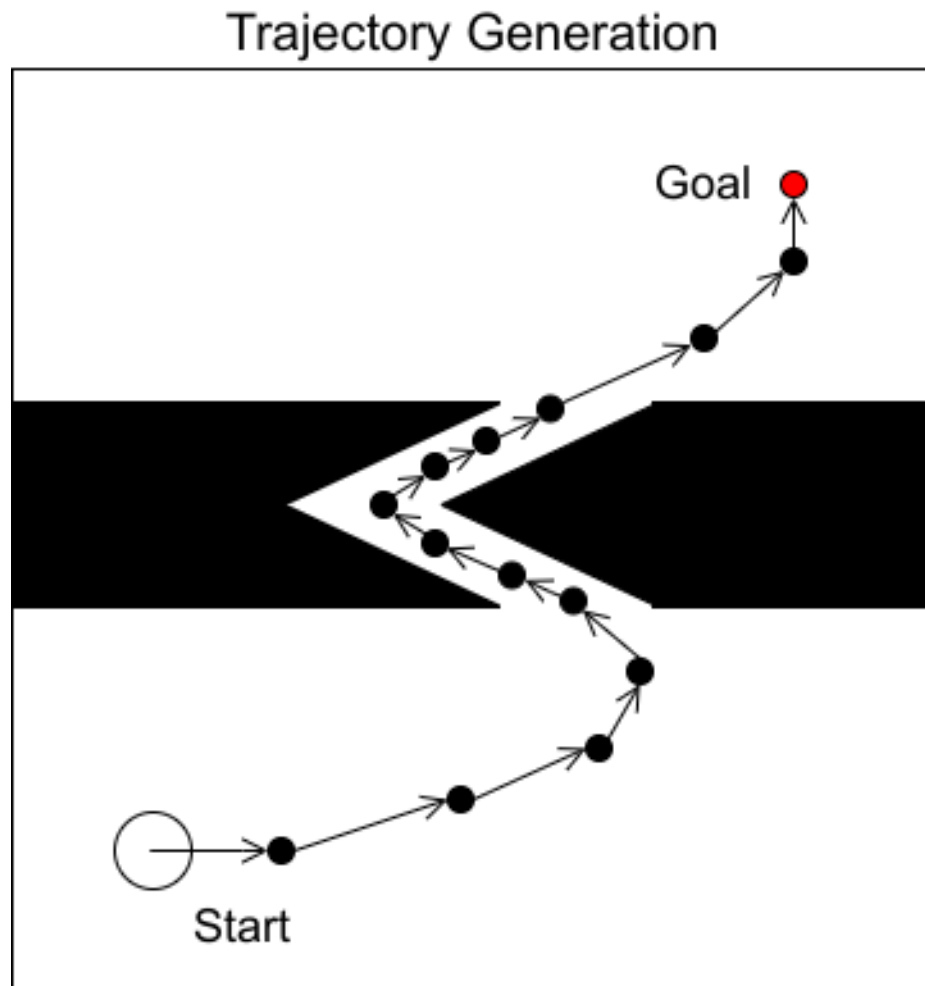


Fig. 1.3 Trajectory Generation

軌道生成は、「1.1.1 経路と軌道の違い」で述べたように、ロボットがどのような運動で通るかを決めることである。多くの経路計画ではロボットの運動学やメカニズムを無視しているため、ロボットの機構や特性で動作が難しい経路が計画されることがある。その例を以下に示す。

- アッカーマンリンクの自動車がその場で  $90^\circ$  回転するような経路
- 対向二輪のロボットが姿勢を変えずに車輪に対して垂直に移動するような経路
- 等速で狭い場所も広い場所を通るような経路

そこで，ロボットが実際に動作及び制御可能な範囲で経路を追従させるために運動学やメカニズムを考慮した「軌道生成」を用いてロボットの運動を決める．軌道生成を行うことで狭い道で速度を遅くしたり，広い道では速度を速くするといった動作や計画された経路の通過地点に対してロボットの機構や特性を考慮した軌道を生成することができる．

Fig.1.3 は軌道生成のイメージである．黒い点がロボットの位置であり，矢印がロボットの速度，中央にある黒い部分が障害物である．軌道生成ではこのようなロボットの軌道を生成することができる．

## 1.2 第 1 章のまとめ

第一章では, "Path" と "Trajectory" の違い, 経路計画と軌道生成について解説した. "Path" はロボットどのように通るか, "Trajectory" はロボットがどのように運動するかという意味であった. 経路計画は障害物と位置と姿勢を考慮したロボットが通る経路を計画するものであった. また, 軌道生成はロボットの運動学, 機構や特性を考慮した軌道を生成することであった.

## 第 2 章

# 経路計画のアルゴリズム

### 2.1 第 2 章の導入

経路計画には様々なアルゴリズムがあり，ベースとなる理論も様々である．それらの中で代表的なアルゴリズムの基礎理論である，ダイクストラ法や A\*法に使用されるグラフ探索，人工ポテンシャルを使用するポテンシャル法，RRT などに使用されるランダムに配置した点を元に経路を決める RRM，以上の 3 種類のアルゴリズムの基礎的な理論について簡単に紹介する．

## 2.1.1 グラフ探索

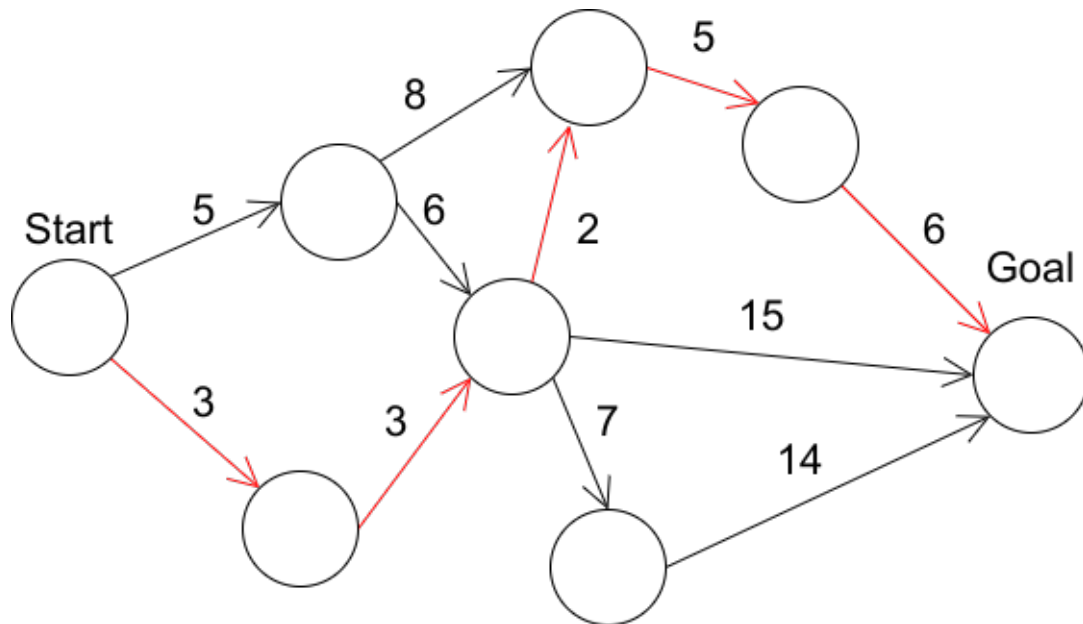


Fig. 2.1 Connected graph

グラフ探索では、経由する点と点を結び最短な経路を探索する。ここで点を「頂点」、点と点を結んで線にしたものを「パス」という。グラフ探索ベースの経路計画では、スタートの距離、すなわち重みを0として頂点間のパスに指定された重みに応じて頂点の重みが加算されていく。最終的にゴールまでの重みが小さい経路が選択されそれが最短の経路であるという理論である。Fig.2.1がそのイメージ図である。円が頂点、矢印がパスである。現実的には、点がロボットが経由する地点で、矢印がロボットが移動する経路で重みがロボットが移動する距離である。赤い矢印の経路が最も重みが小さいため、最短経路ということになる。

しかし、このアルゴリズムにはパスがロボットの運動として適切かどうかという問題がある。すなわち、前述したロボットの機構や特性を無視したパスが生成される問題である。それらの問題があってもスタートからゴールまでを最短距離で移動するには有用なアルゴリズムであり、ダイクストラ法やA\*法で使用されている。これらのアルゴリズムは大域経路計画としても使用されている。

グラフ探索の別の応用先としてはカーナビゲーションが挙げられる。

### 2.1.2 ポテンシャル法

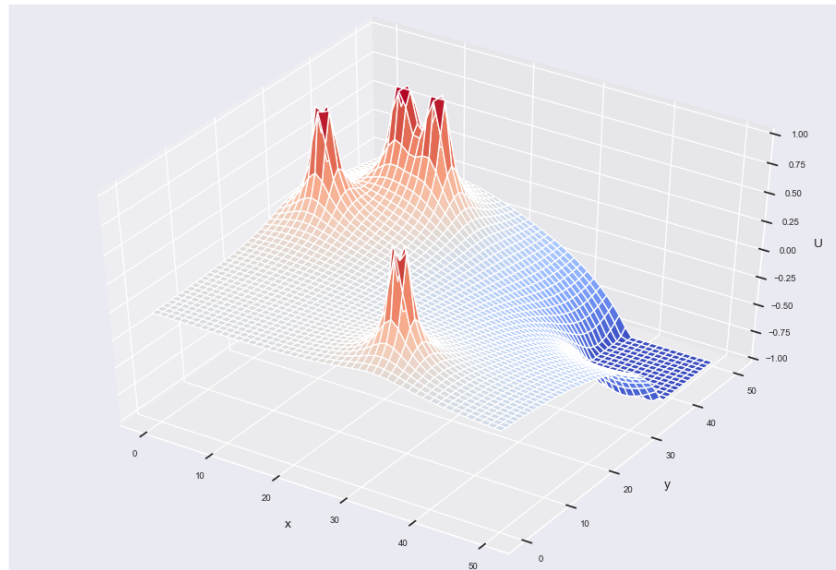


Fig. 2.2 Potential field [1]

ポテンシャル法は人工的に作り出したポテンシャル場を使用して、スタートからゴールまでボールが転がる方向にロボットを移動させる方法である。ポテンシャル場はゴールが谷底の斜面になっており、障害物は山になる。スタートからボールを転がすと山と山の間の谷をコロコロと谷底のゴールに向かって転がっていくというのがポテンシャル法のイメージである。

問題点としては、局所解であるローカルミニマムに陥ってしまうことが挙げられる。これはゴールではない部分に谷底がもう一つできてしまう現象である。このローカルミニマムを回避する方法は右手法など様々に研究されている。

## 2.1.3 RRM ( Random Road Map )

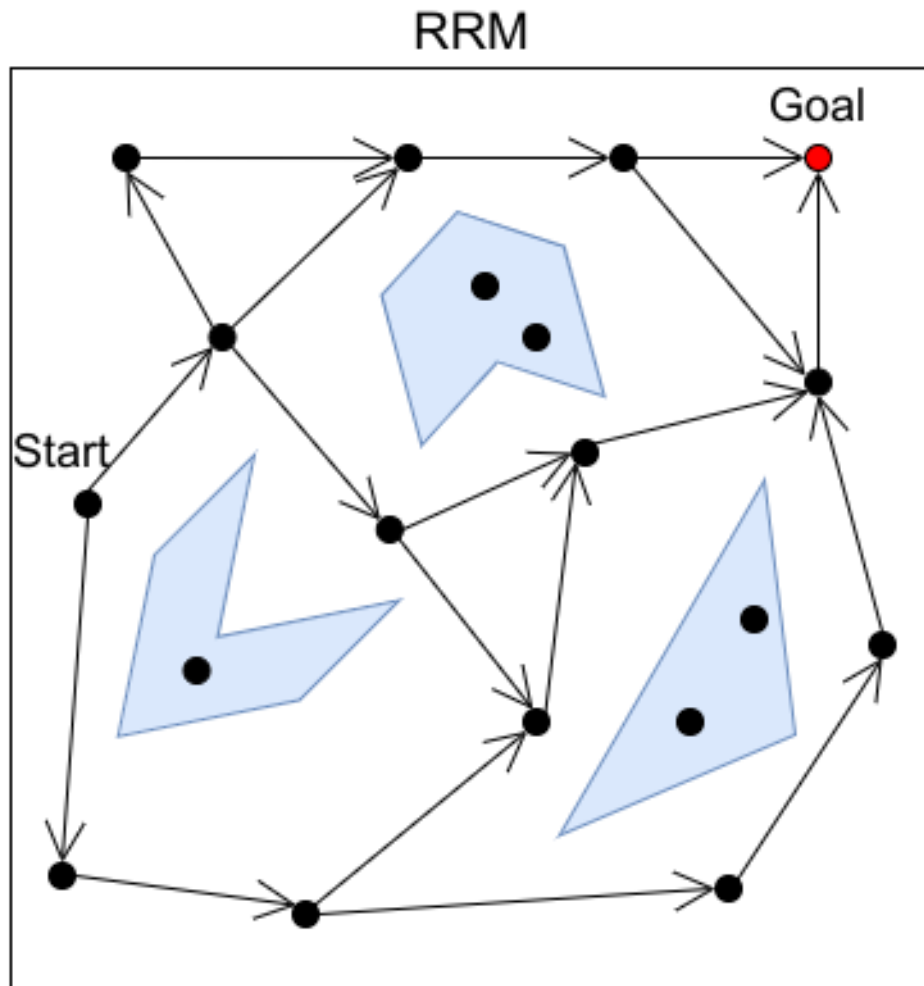


Fig. 2.3 RRM

RRM はランダムに Seeds と呼ばれる点をばらまき，障害物の内部の Seeds は考えずにその他の点を結んでグラフを作り経路を計画する手法である．Fig.2.3 は RRM のイメージ図である．黒い点が Seeds，水色の図形が障害物，矢印が経路である．このようにランダムに配置された Seeds の内，障害物内のものを除いて接続していき経路を生成する．

問題点としては，計算量が多いことと，通る場所が狭いときに更に計算時間がかかることである．この問題点を解決するために RRT などの改良版のアルゴリズムが存在する．



## 2.2 第 2 章のまとめ

第 2 章では経路計画のアルゴリズムの基礎について簡単に紹介した。グラフ探索ベースの経路計画は、頂点とパスの重みが考慮され、最も重みが少ない経路が選択される。ポテンシャル法ではスタートから谷底のゴールに向かってボールが転がる方向にロボットを移動させる。RRM はランダムに点を配置し、障害物内部以外の点を結び経路を計画する。経路計画では、実際に選択されたパスは通れるのか、という問題が存在する。その問題を解決するために、狭い部分では速度を落とし、広い部分では速度を上げるといった、運動を生成するために軌道生成が用いられる。

## 第 3 章

# 軌道生成の例題

### 3.1 第 3 章の導入

経路計画には実際に選択されたパスは通れるのかという問題が残った．これを解決するために，軌道生成を行う．今回の軌道生成では，関節空間を用いて位置  $x$  の 1 自由度の対向二輪ロボットで行う．はじめに，連続性の必要性について説明し，次に bang-bang 制御を用いた軌道生成を解説する．最後に，台形速度曲線の制御を解説する．

軌道生成で用いられるグラフは時間をパラメータとした位置・姿勢などの変化のグラフである．

## 3.1.1 連続性のない制御

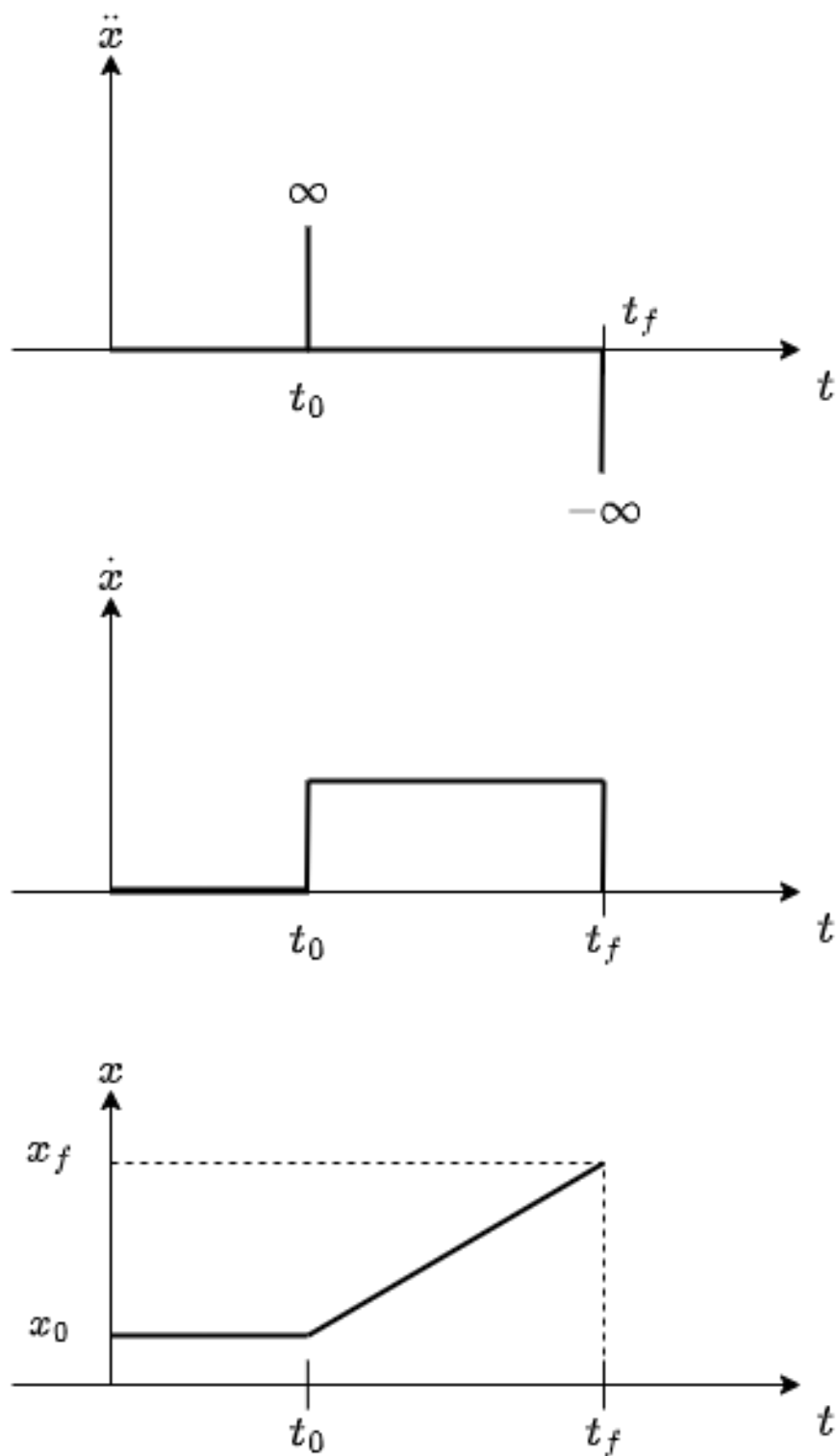


Fig. 3.1 No continuity

ロボットの位置を変化させるには速度を出す必要がある．初速度  $0\text{m/s}$  から目標速度までステップ状に変化させると時間に対する位置と加速度のグラフは Fig3.1 のようになる．図を見ると加速度が無限大のインパルスが必要としていることがわかり，これは現実的ではない．加速度が無限大のインパルスが必要としているのは，速度の変化がステップ状になっているからである．すなわち，速度の変化量が無限になるからである．ここからわかることは，速度は連続性が必要であるということである．また，速度は位置の変化量であるため，位置も連続性が必要である．

このように速度に連続性のない制御は現実的ではないことがわかる．逆に，加速度の連続性は必須ではないが加速度に  $1\text{ms}$  ぐらいの制御周期で連続性をもたせると，なめらかな力制御を実現できる．

## 3.1.2 bang-bang 制御

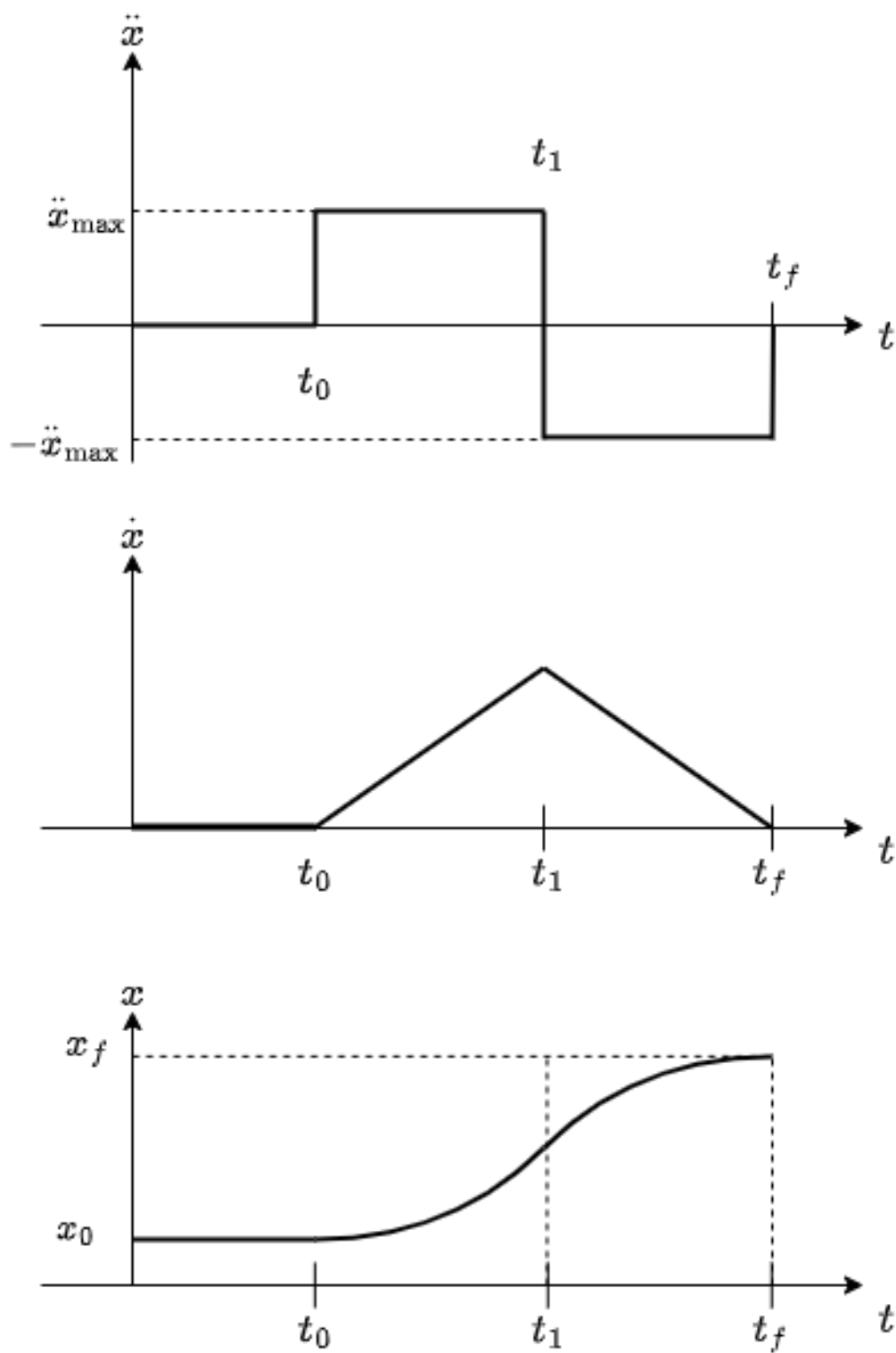


Fig. 3.2 bang-bang control

速度に連続性をもたせた制御方式に bang-bang 制御があり，最短時間軌道を達成できる．最短時間軌道は，最速で目標地点に到達する際に有用な方法である．bang-bang 制御では Fig.3.2 の加速度のグラフからわかるように，最大加速と最小加速の2つの加速度をかけているため，最短時間で目標地点に到達することができる．これは出発時にアクセルを全開にして，一気にブレーキをかけて，止まるというイメージである．

軌道生成及び制御を行うには初期値，目標値が必要である．今回は，位置  $x$  を変数とするため，初期値である初期位置を  $x_0$ ，目標値である目標位置を  $x_f$ ，また，初速度  $\dot{x}_0 = 0$ ，目標地点での速度  $\dot{x}_f = 0$  とする．また，bang-bang 制御は最大加速度  $\ddot{a}_{max}$  を必要とする．この bang-bang 制御を式で表すと以下ようになる．

$$\ddot{x} = \begin{cases} a & (t_0 < t \leq t_1) \\ -a & (t_1 < t \leq t_f) \end{cases} \quad (3.1)$$

$$\dot{x} = \begin{cases} \dot{x}_0 + a(t - t_0) & (t_0 < t \leq t_1) \\ \dot{x}_1 + a(t - t_1) & (t_1 < t \leq t_f) \end{cases} \quad (3.2)$$

$$\text{where } \dot{x}_1 = \dot{x}_0 + a(t_1 - t_0), \dot{x}_0 = 0$$

$$x = \begin{cases} x_0 + \frac{1}{2}a(t - t_0)^2 & (t_0 < t \leq t_1) \\ x_1 + \dot{x}_1(t - t_1) - \frac{1}{2}a(t - t_1)^2 & (t_1 < t \leq t_f) \end{cases} \quad (3.3)$$

$$\text{where } x_1 = x_0 + \frac{1}{2}a \left\{ \frac{(t_1 - t_0)}{2} \right\}^2$$

拘束条件を導出する．以上の3つの式を一般性を失わずに  $t_0 = 0$  とすると以下のようになる．

$$\ddot{x} = \begin{cases} a & (0 < t \leq t_1) \\ -a & (t_1 < t \leq t_f) \end{cases} \quad (3.4)$$

$$\text{where } t_1 = \frac{t_f}{2}$$

$$\dot{x} = \begin{cases} at & (t_0 < t \leq t_1) \\ \dot{x}_1 - a(t - \frac{t_f}{2}) & (t_1 < t \leq t_f) \end{cases} \quad (3.5)$$

$$\text{where } \dot{x}_1 = a \frac{t_f}{2}$$

$$x = \begin{cases} x_0 + \frac{1}{2}at^2 & (t_0 < t \leq t_1) \\ x_1 + \dot{x}_1 t - \frac{1}{2}at^2 & (t_1 < t \leq t_f) \end{cases} \quad (3.6)$$

$$\text{where } x_1 = x_0 + \frac{1}{2}a\left(\frac{t_f}{2}\right)^2$$

したがって、目標位置は

$$x_f = x_1 + \dot{x}_1 t_f + a \frac{(t_f)^2}{2} \quad (3.7)$$

となり、式 (3.7) に  $\dot{x}_1$  ,  $x_1$  を代入すると以下ようになる。

$$x_f = x_0 + \frac{1}{2}a\left(\frac{t_f}{2}\right)^2 + a \frac{t_f}{2} t_f + a \frac{(t_f)^2}{2} \quad (3.8)$$

式 (3.8) の  $x_f$  ,  $x_0$  , は定数であり、変数は、目標地点への到達時間  $t_f$  と加速度  $a$  である。最終到達時間  $t_f$  が決まると加速度  $a$  を求めることができ、すべての経路を決めることができる。このときの加速度  $a$  は最大加速度  $\ddot{x}_{max}$  となる。

ここで、与える条件をまとめると以下ようになる。

目標

$$x_0, x_f$$

拘束

$$\dot{x}_0 = 0, \dot{x}_f = 0, t_0 = 0, t_f$$

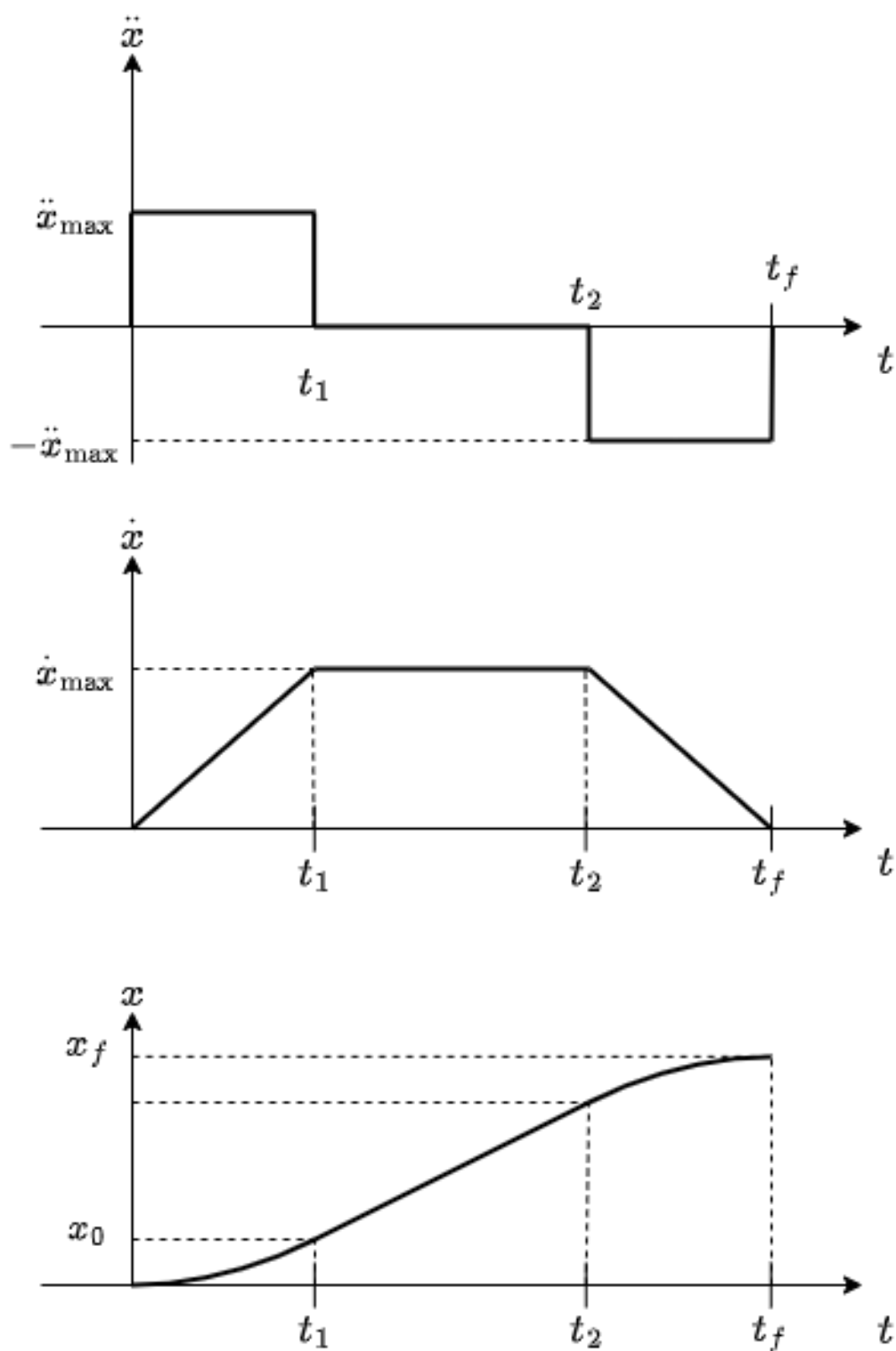


Fig. 3.3 bang-bang control with additional velocity constraints



bang-bang 制御は加速度の変化に対応していないが、速度を変えることはできる。拘束条件として、 $\dot{x}_{max}$  を追加すると、Fig.3.3 のようになる。ここで  $\dot{x}_{max}$  は  $\dot{x}_{max} = at_1$  と求めることができる。最短時間で最大速度を出すとき  $\dot{x}_{max} = \ddot{x}_{max}t_1$  となる。目標地点到達時間  $t_f$  は  $t_f = t_1 + t_2$  となり、 $t_1$  と  $t_2$  がわかれば、求めることができる。

$t_1$  は  $\dot{x}_{max} = \ddot{x}_{max}t_1$  より

$$t_1 = \frac{\dot{x}_{max}}{\ddot{x}_{max}} \quad (3.9)$$

となる。

$t_0 < t \leq t_1$  のときの速度面積と  $t_2 < t \leq t_f$  のときの速度面積は同じであるため、 $t_2$  での距離は  $t_0$  から  $t_2$  までの速度が作り出す長方形の速度面積と同じになる。したがって、 $t_2$  は

$$t_2 = \frac{x_f - x_0}{\dot{x}_{max}} \quad (3.10)$$

となる。

これにより、 $\dot{x}_{max}$  と  $\ddot{x}_{max}$  を求めることができる。

## 3.2 第 3 章のまとめ

第 3 章では bang-bang 制御による軌道制御の解説を行った．速度と位置には連続性が必要であり，拘束条件と目標を定める必要があった．bang-bang 制御では最大加速度を求める必要があった．また，速度が台形の場合の計算を行った．この場合は，拘束条件に最大速度を追加することで，なだらかな速度変化を行えることを紹介した．

## 参考文献

- [1] Qiita — 【python】python でポテンシャル場の計算. [https://qiita.com/koichi\\_baseball/items/32e590f4d6151ca2072a](https://qiita.com/koichi_baseball/items/32e590f4d6151ca2072a), 6 2020. (Accessed on 7/13/2023).
- [2] 日本機械学会 — 軌道計画 [jsme mechanical engineering dictionary]. <https://www.jsme.or.jp/jsme-medwiki/doku.php?id=14:1002683>, 7 2017. (Accessed on 7/13/2023).
- [3] 日本機械学会 — 経路計画 [jsme mechanical engineering dictionary]. <https://www.jsme.or.jp/jsme-medwiki/doku.php?id=14:1003539>, 2 2023. (Accessed on 7/13/2023).
- [4] kikakurui.com — jis b 0186:2003 移動ロボット-用語. <https://kikakurui.com/b0/B0186-2020-01.html>. (Accessed on 7/13/2023).
- [5] Tokoro's tech-note — [techaday:0007] 軌道と経路の違い. <https://blog.tokor.org/2015/12/01/%E8%BB%8C%E9%81%93%E3%81%A8%E7%B5%8C%E8%B7%AF%E3%81%AE%E9%81%95%E3%81%84/>, 12 2015. (Accessed on 7/13/2023).
- [6] 工業大学生ももやまのうさぎ塾 — うさぎでもわかる離散数学(グラフ理論) 第 14 羽  
ダイクストラ法による最短経路の求め方. <https://www.momoyama-usagi.com/entry/math-risan14>, 10 2019. (Accessed on 7/13/2023).
- [7] 岩崎 聡 山内 由章 中田 昌宏 彌城 祐亮. ポテンシャル法によるロボット製品の障害物回避技術の開発. 三菱重工技報. (2014), vol.51, no.1, p.40-45.
- [8] 諸岡健一. 九州大学 — コンフィギュレーション空間. <http://fortune.ait.kyushu-u.ac.jp/~morooka/lecture/Lecture03.pdf>. (Accessed on 7/13/2023).
- [9] アールティヒューマノイドロボットブログ — ロボットアーム研修 (まえけん編) その  
10 関節空間上の軌道生成. <https://rt-net.jp/humanoid/archives/1939#toc3>, 7

2020. (Accessed on 7/13/2023).

[10] ひつじ tips — ロボットの人工ポテンシャル法をざっくり解説する.

<https://mu-777.hatenablog.com/entry/2022/01/08/152647#%E4%BA%BA%E5%B7%A5%E3%83%9D%E3%83%86%E3%83%B3%E3%82%B7%E3%83%A3%E3%83%AB%E6%B3%95%E3%81%AF%E4%BD%95%E3%82%92%E3%81%99%E3%82%8B%E3%82%82%E3%81%AE%E3%81%8B>, 1

2022. (Accessed on 7/13/2023).

本レポートはインテリジェントロボットモーション第 11 回「軌道計画と軌道生成」の講義を元に制作した。