

2023（令和5）年度 修士学位論文

水平駆動形平行メカニズムを用いた  
収穫用マニピュレータの制御

Harvesting Manipulator using Horizontal Drive  
Parallel Mechanism

2024年2月13日

高知工科大学大学院 工学研究科基盤工学専攻  
知能機械工学コース

1265003 上野 勇希

指導教員 岡 宏一

# 目次

第 1 章	序論	1
1.1	研究背景	1
1.2	研究目的	3
1.3	パラレルメカニズム	3
1.4	パラレルメカニズムの分類	4
第 2 章	提案するマニピュレータと運動学	5
2.1	提案するリンクメカニズム	5
2.2	リンク比による負荷の違い	5
2.3	運動学	7
2.3.1	パラレルメカニズムの逆運動学	7
2.3.2	パラレルメカニズムの順運動学	9
第 3 章	実験機の製作	11
3.1	製作した実験機	11
3.2	使用機材	12
3.3	モータの制御方法	13
第 4 章	実験機の精度検証	14
4.1	検証方法	14
4.2	シミュレーションの可動範囲	15
4.3	実験機の可動範囲	16
第 5 章	実験準備	18
5.1	実験環境	18
5.2	画像処理	18
5.2.1	中心位置検出	19
5.2.2	距離検出	20
5.3	制御方法	23
5.3.1	局所位置制御	23
5.3.2	角速度指令	24
5.3.3	軌道制御	26
5.4	シミュレーション	27
5.4.1	角速度指令のシミュレーション	27
5.4.2	軌道制御のシミュレーション	28
第 6 章	実験結果	30
6.1	局所位置制御	30
6.2	角速度指令	32
6.3	軌道制御	34
6.4	結果の比較	36

6.5	シミュレーションとの比較 .....	37
6.6	角速度指令ゲインの再検証 .....	39
6.6.1	ゲイン減少法 .....	39
6.6.2	ゲイン増加法 .....	41
6.7	制御の組み合わせ .....	43
第7章	結論 .....	46
7.1	研究のまとめ .....	46
7.2	今後の課題 .....	46
	謝辞 .....	47
	参考文献 .....	48

# 第1章 序論

## 1.1 研究背景

農林水産省による 2018 年から 2023 年の農業人口と 65 歳以上の人口のデータ<sup>[1]</sup> を図 1.1 に示す。図 1.1 によると、2018 年では農業人口が 145.1 万人であったが、年々減少の傾向があり、2023 年では 116.4 万人となっている。また、65 歳以上の高齢者の割合は 2018 年では約 68%であったが、2023 年では約 70.7%と上昇している。このデータにより、農業従事者の高齢化や労働力不足が課題となっていることがわかる。現在、農林水産省は課題を解決するためにスマート農業<sup>[2]</sup>を推進し、ICT やロボットの先端技術を積極的に取り入れることで自動化を実現しようとしている。自動化によって作業者がいない時間帯でも収穫作業が可能のため、負担削減や作業効率の向上が期待できる。これまでの開発事例としては、図 1.2 に示す円筒座標系型、図 1.3 に示す垂直多関節型などのマニピュレータが挙げられる。

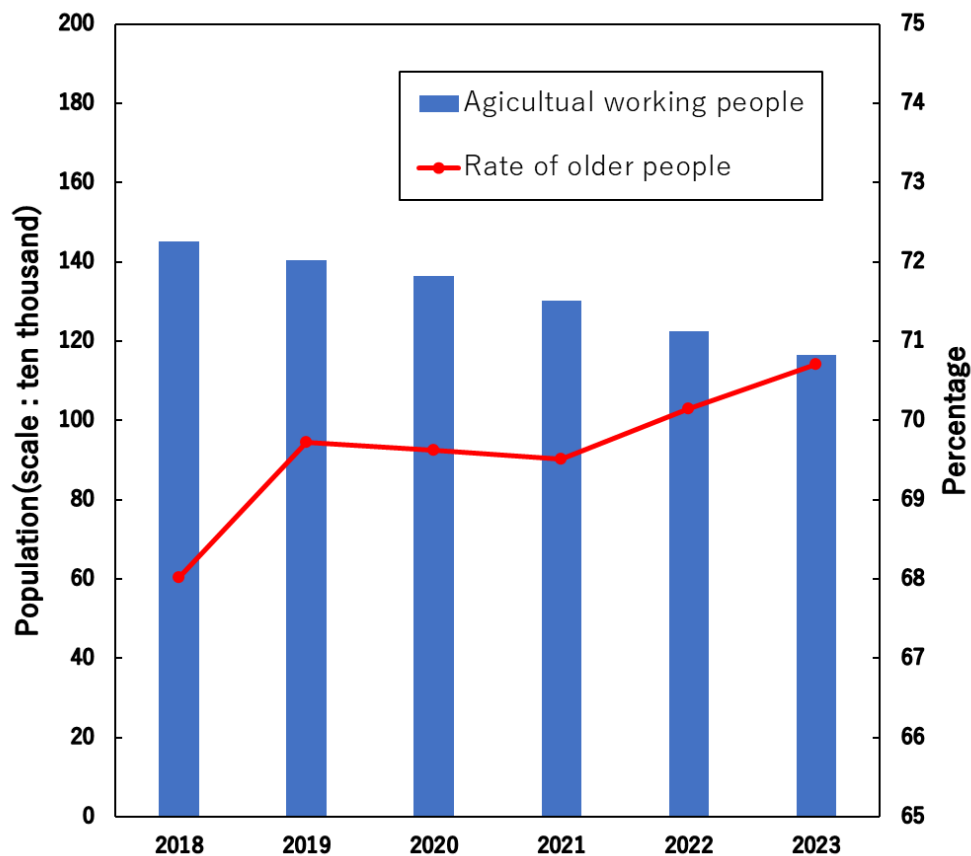


Fig.1.1 Transition of Agricultural working population<sup>[1]</sup>



Fig1.2 Cylindrical coordinate system manipulator<sup>[3]</sup>



Fig1.3 Vertical articulated manipulator<sup>[4]</sup>

図 1.2 のロボットは AGRIST 株式会社が開発した吊り下げ式ピーマン収穫ロボット「L」<sup>[3]</sup>でワイヤーを伝って移動，ピーマンを探索し円筒座標系型のマニピュレータで収穫を行う．図 1.3 のロボットはヤンマーホールディングス株式会社が開発した大玉トマト収穫ロボット<sup>[4]</sup>で個体ごとに形状の異なり，扱いが難しい大玉トマトを垂直多関節型のマニピュレータで収穫を行う．以上に採用されているシリアルリンク型のマニピュレータは各リンク間にアクチュエータを配置する必要があり，その重量によって可動部の質量が大きくなるため，高速化は難しい．同様の理由で根元のずれが出力先で大きな誤差になってしまうという問題がある．特に，ロボットを活用した農作業においては，収穫の効率化が大きな目的となっているため，高速動作が可能なマニピュレータの需要は高い．

## 1.2 研究目的

本研究では高知県で生産量の多いピーマンを対象とし、高速、高精度を目指した収穫用マニピュレータの開発を目的としている。このため、高速化が容易なパラレルメカニズムを採用し、収穫動作に対応させるため水平方向に駆動する実験機を製作する。本稿では水平駆動の課題を解決するためにリンク比の検討を行う。そして、製作した実験機を用いて複数の制御方法について実験を行った。

## 1.3 パラレルメカニズム

シリアルメカニズムとパラレルメカニズムについて図 1.4 に示す。シリアルメカニズムは、アクチュエータを配置した各ジョイントを介してリンクが直列に連結されている機構である。高い自由度を持ち、人間のよう動作ができることが利点であり、多くの工場で使用されている。しかし、前述したように高速化が難しく、誤差が累積されやすいという課題がある。これに対してパラレルメカニズムは、アクチュエータを並列に配置している機構であり、各リンクを同時に動作させている。特徴として各リンクに引張・圧縮のどちらかの力のみが作用する機構とできるため剛性を高めやすいこと、アクチュエータをベースに配置することで出力先が軽量となり、高速化が容易であること、各リンクの誤差が平均化されることで出力先の誤差は小さくなり高精度となることが挙げられる<sup>[9]</sup>。これらの特徴は精度が高いことで果実を傷つけにくい点、高速な動作で効率を良くできる点で効果的である。

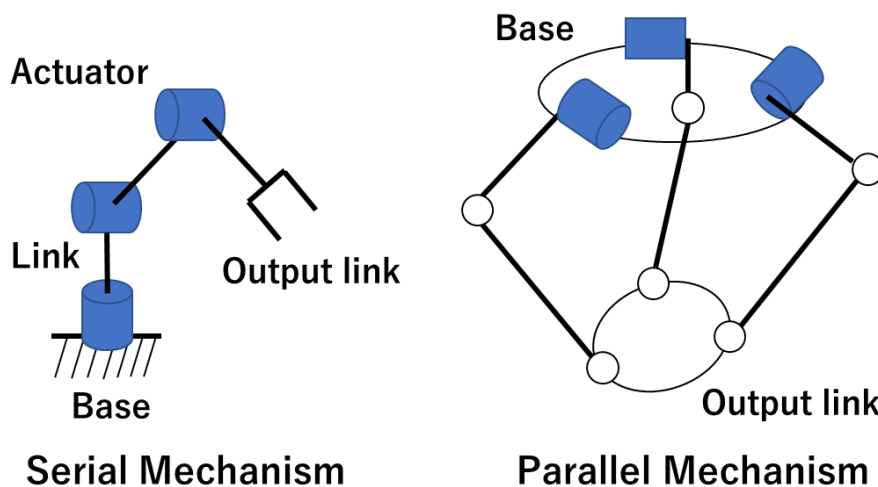


Fig1.4 Outline of the mechanism

#### 1.4 パラレルメカニズムの分類

図 1.5 に駆動方式の違いから伸縮型，回転型，直動型と大きく 3 つに分類<sup>[6]</sup>したパラレルメカニズムを示す。

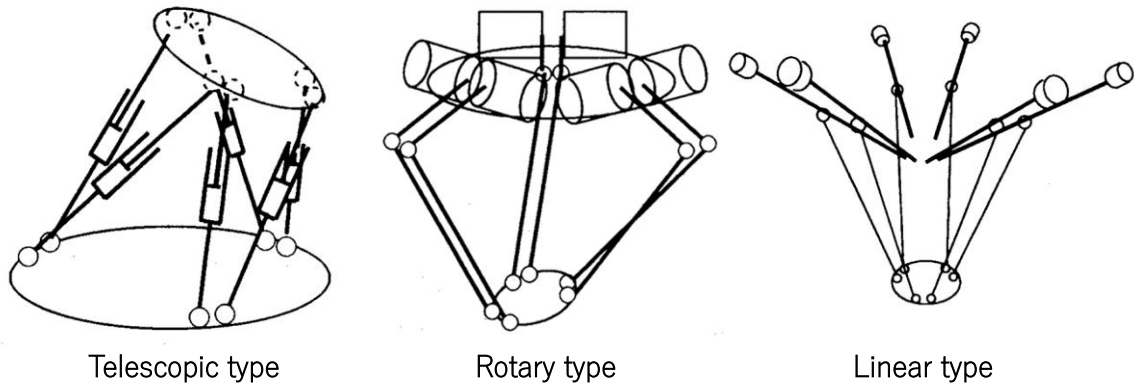


Fig1.5 Classification of parallel mechanism<sup>[6]</sup>

駆動方式についてそれぞれ説明する．伸縮型はシリンダタイプのアクチュエータによってリンクを伸縮させる方式であり，機構がシンプルで運動学計算も比較的簡単である．しかし，アクチュエータ自体が動くため，機構の大型化や可動範囲が狭いという課題が挙げられる．回転型はモータなどの回転機構によってリンクを動作させる方式であり，アクチュエータが固定されているため高速化が容易で可動範囲も広い．しかし，アクチュエータ側のリンクに曲げモーメントが発生するため重量物の保持には向いていない．直動型は直動アクチュエータを用いてリンクを制御し，出力先を動作させる方式である．アクチュエータと共にベースに固定できるため高出力が期待できるが，ベース部に大きなスペースが必要であるという課題がある．本研究では，収穫用マニピュレータとして軽量の農作物を対象としていること，移動機構に乗せた運用を想定しているため機構の大型化は避けたいことの 2 点から，回転型のパラレルメカニズムを採用した。

## 第2章 提案するマニピュレータと運動学

### 2.1 提案するリンクメカニズム

マニピュレータとして使用する平行リンクロボットのモデル図を図 2.1 に示す。ベース上に配置した3つのアクチュエータによる入力で3自由度の並進運動の位置決めを行う。リンクをつなぐジョイントと出力リンク上のジョイントは2自由度である。従来の平行リンクロボットは、主に鉛直方向で使用されているが、収穫においては果実の上部にある葉や茎が邪魔をして鉛直駆動では難しい場合が多いため、水平駆動への変更を提案する。その場合、図 2.1 の赤矢印のように手先にかかる荷重の向きが鉛直方向である Z 軸方向となり、各リンクにかかる曲げ応力やそれに伴う変位が大きくなる。このため対応策としてリンク長さの比を変更する。

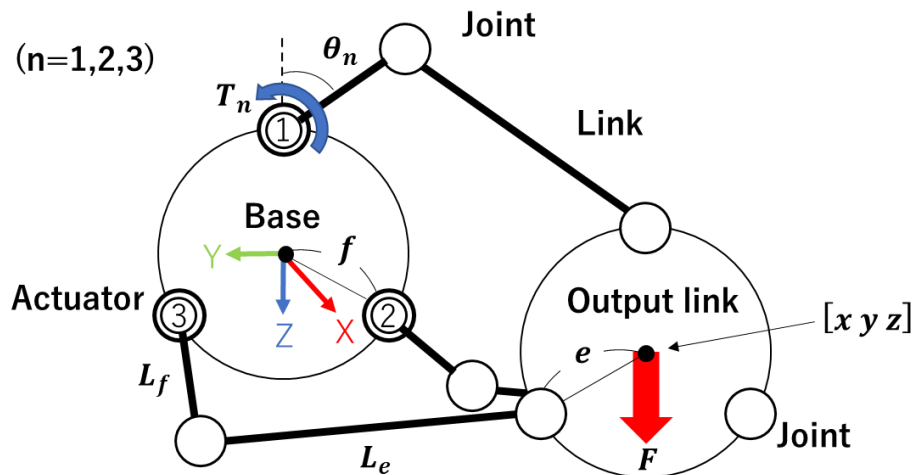


Fig2.1 Model of parallel link robot

### 2.2 リンク比による負荷の違い

図 2.1 について XZ 平面から見た上部のリンクを図 2.2 に示す。一般に、可動範囲を広くとるために図 2.2 左部のようにベース側のリンク長さ  $L_f$  は出力リンク側のリンク長さ  $L_e$  より短くしているが、今回は負荷を軽減するために図 2.2 右部のように比を変更していく。具体的には、リンク長さの比を 1:3, 1:2, 1:1, 2:1 と変更していき、SolidWorks を用いて応力解析を行った。その結果を図 2.3 に、最大曲げ応力、最大変位についての比較を表 2.1 に示す。

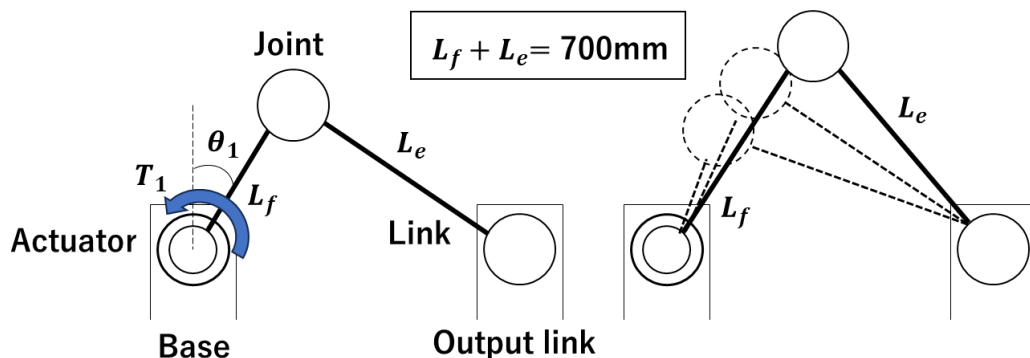


Fig2.2 Model diagram of upper link



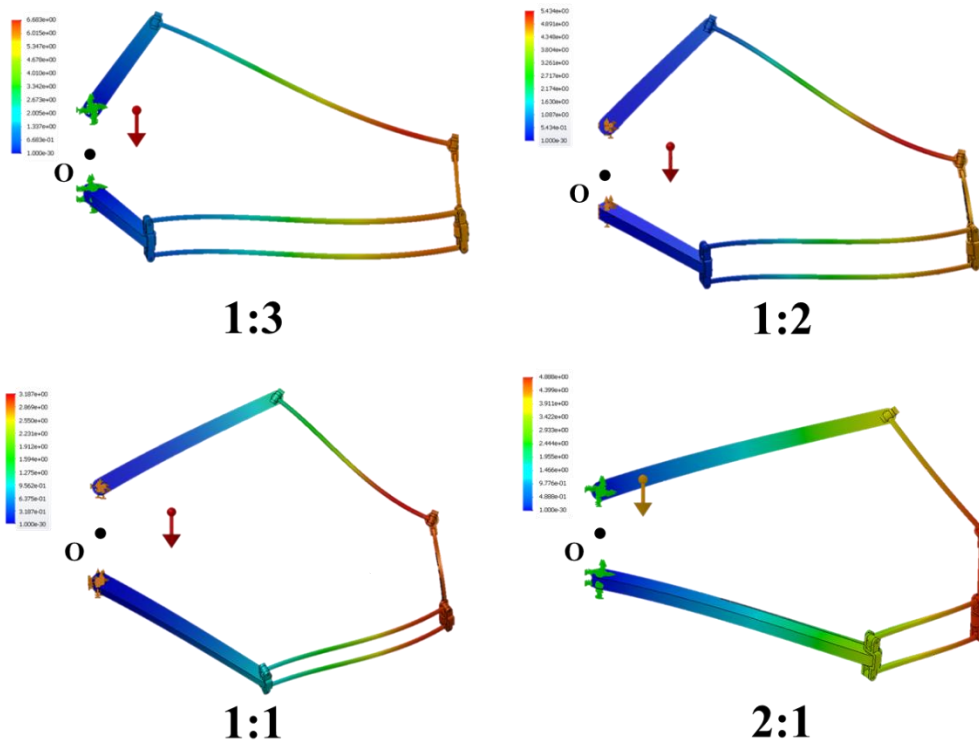


Fig2.3 Stress analysis result for different link length ratios

Table2.1 Comparison of maximum bending stress and maximum displacement

$L_f:L_e$	Maximum bending stress[MPa]	Maximum displacement[mm]
1:3	1.1501	6.683
1:2	0.6531	5.434
1:1	0.3189	3.187
2:1	0.8072	4.888

図 2.3 より、長いほうのリンクに負荷が大きくかかることがわかる。そのため、表 2.1 のようにリンク比が 1:1 のとき、リンクの長さが等しいことで負荷が均等化され、最大曲げ応力とそれに伴う変位が小さくなることがわかった。リンク比を 1:1 にする別の利点として、図 2.3 に示す原点 O 付近も駆動可能なことが挙げられる。これにより収穫後に原点付近のベースに戻り、収穫物を回収する動作が可能となる。以上のことより水平駆動形パラレルメカニズムの実験機はリンク比を 1:1 に変更する。

## 2.3 運動学

マニピュレータをはじめとするロボットの制御では、機構の入力と出力される運動との関係である運動学解析が必要である。パラレルリンクロボットにおいては、入力角度に対して手先の現在位置を予想する順運動学、目標とする手先位置に対して入力する角度を求める逆運動学が必要であり、動作実験を行うために前項で定めたリンク比 1:1 のパラレルメカニズムにおける運動学を計算する。また、運動学計算で用いる記号を以下のように定義する。

$r$  : ベース半径, 出力リンク半径

$l$  : ベース側リンク長さ, 出力リンク側リンク長さ

$\theta_n$  : 入力角度

$E_0$  : 出力位置 $[x\ y\ z]$

$F_n$  : ベース上にある 1 自由度の回転ジョイント

$E_n$  : 出力リンク上にある 2 自由度のユニバーサルジョイント

$E'_n$  :  $E_{ny} = 0$ となる $E_n$ から遷移させた位置

$J_n$  : リンクをつなぐ 2 自由度のユニバーサルジョイント

$J'_n$  : 出力リンク上で中心になるように $J_n$ を遷移させた位置 $[x_n\ y_n\ z_n]$

添え字 $n$ は時計周りに上部から 1, 2, 3 の順番となっている。また、リンク比が 1:1 であるのでベース半径と出力リンク半径, ベース側リンク長さと出力リンク側リンク長さを等しくしている。

### 2.3.1 パラレルメカニズムの逆運動学

パラレルメカニズムの逆運動学を計算して出力位置 $[x\ y\ z]$  から入力角度 $[\theta_1\ \theta_2\ \theta_3]$ を算出する<sup>[7]</sup>。図 2.4 の左図に全体図, 右図に XZ 平面上のベースを示す。

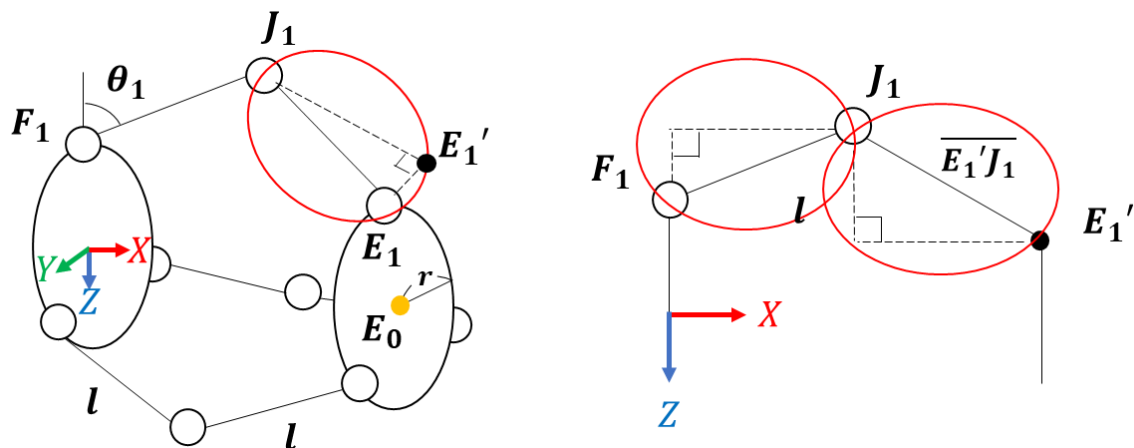


Fig2.4 Model diagram in inverse kinematics

まず  $n=1$  の場合について考える.  $OF_1$ ,  $OE_1$ ,  $OE'_1$  は以下となる.

$$OF_1 = (0, 0, -r) \quad (2.3.1)$$

$$OE_1 = (x, y, z - r) \quad (2.3.2)$$

$$OE'_1 = (x, 0, z - r) \quad (2.3.3)$$

図 2.4 より赤線で囲った直角三角形が 3 つあるので以下の式が得られる.

$$|E'_1 J_1| = \sqrt{|E_1 J_1|^2 - |E_1 E'_1|^2} = \sqrt{l^2 - y^2} \quad (2.3.4)$$

$$(J_{1z} - F_{1z})^2 + (J_{1x} - F_{1x})^2 = l^2 \quad (2.3.5)$$

$$(J_{1z} - E'_{1z})^2 + (J_{1x} - E'_{1x})^2 = l^2 - y^2 \quad (2.3.6)$$

式(2.3.5)と式(2.3.6)に式(2.3.1)から式(2.3.3)の値を代入して以下の式が得られる.

$$(J_{1z} - (-r))^2 + J_{1x}^2 = l^2 \quad (2.3.7)$$

$$(J_{1z} - (z - r))^2 + (J_{1x} - x)^2 = l^2 - y^2 \quad (2.3.8)$$

$J_{1z}$  と  $J_{1x}$  について解くと

$$J_{1z} = \frac{(r - ab - \sqrt{c})}{b^2 + 1} \quad (2.3.9)$$

$$J_{1x} = a + bJ_{1z} \quad (2.3.10)$$

ただし

$$a = \frac{1}{2x} \{x^2 + y^2 + z^2 - 2zr\} \quad (2.3.11)$$

$$b = \frac{1}{x} \{-z\} \quad (2.3.12)$$

$$c = -(a - r)^2 + l^2(b^2 + 1) \quad (2.3.13)$$

であり, 求める入力角度  $\theta_1$  は以下となる.

$$\theta_1 = \tan^{-1} \left( \frac{J_{1x}}{-r - J_{1z}} \right) \quad (2.3.14)$$

それぞれの変数は出力変位  $[x \ y \ z]$  が既知であるとき一意に定まるので  $\theta_1$  も一意に求められる.

$n=2,3$  については, 以下の式のように X 軸周りに 120 度, -120 度回転させて考える.

$$y_2 = y \cos 120 - z \sin 120 \quad (2.3.15)$$

$$z_2 = y \cos 120 + z \sin 120 \quad (2.3.16)$$

$$y_3 = y \cos(-120) - z \sin(-120) \quad (2.3.17)$$

$$z_3 = y \cos(-120) + z \sin(-120) \quad (2.3.18)$$

それぞれ添え字を変更して同様の計算を行うことで  $\theta_2$ ,  $\theta_3$  が求められる.

以上より, 出力位置  $[x \ y \ z]$  から入力角度  $[\theta_1 \ \theta_2 \ \theta_3]$  を算出できた.

### 2.3.2 パラレルメカニズムの順運動学

パラレルメカニズムの順運動学を計算して入力角度 $[\theta_1 \theta_2 \theta_3]$ から出力位置 $[x y z]$ を算出する<sup>[7]</sup>.  
 図 2.5 の左図に全体図, 右図に YZ 平面上のベースを示す.

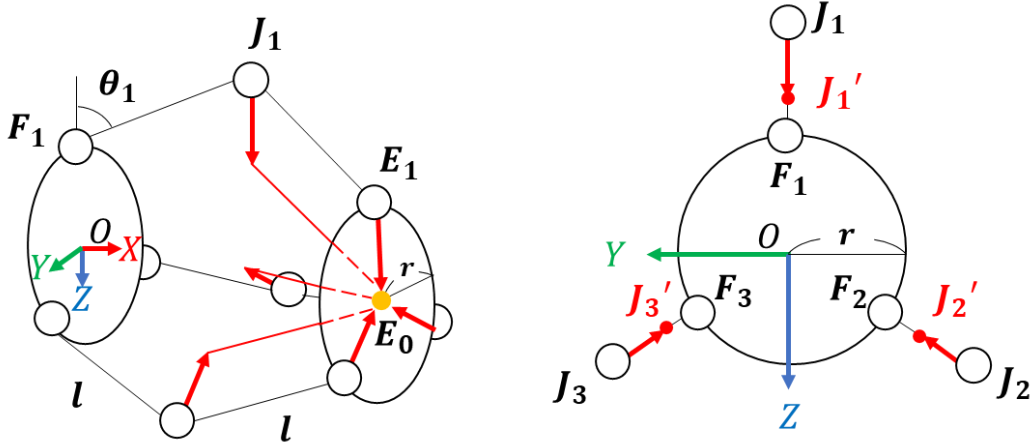


Fig2.5 Model diagram in forward kinematics

図 2.5 から以下の式が得られる.

$$|OF_1| = |OF_2| = |OF_3| = |J_1J'_1| = |J_2J'_2| = |J_3J'_3| = r \quad (2.3.19)$$

$$F_1J_1 = (l \sin \theta_1, 0, -l \cos \theta_1) \quad (2.3.20)$$

$$F_2J_2 = (l \sin \theta_2, -l \cos \theta_2 \cos 30^\circ, l \cos \theta_2 \sin 30^\circ) \quad (2.3.21)$$

$$F_3J_3 = (l \sin \theta_3, l \cos \theta_3 \cos 30^\circ, l \cos \theta_3 \sin 30^\circ) \quad (2.3.22)$$

また,  $OJ'_n = OF_n + F_nJ_n + J_nJ'_n$ であるので $OJ'_n$ は $F_nJ_n$ と等しいので $OJ'_n$ はそれぞれ以下となる.

$$OJ'_1 = (x_1, y_1, z_1) = (l \sin \theta_1, 0, -l \cos \theta_1) \quad (2.3.23)$$

$$OJ'_2 = (x_2, y_2, z_2) = (l \sin \theta_2, -l \cos \theta_2 \cos 30^\circ, l \cos \theta_2 \sin 30^\circ) \quad (2.3.24)$$

$$OJ'_3 = (x_3, y_3, z_3) = (l \sin \theta_3, l \cos \theta_3 \cos 30^\circ, l \cos \theta_3 \sin 30^\circ) \quad (2.3.25)$$

求めたい出力位置は $J'_n$ を中心とする半径 $l$ の 3 つの球が交わる点である. 3 つの球の方程式は以下となる.

$$(x - x_1)^2 + y^2 + (z - z_1)^2 = l^2 \quad (2.3.26)$$

$$(x - x_2)^2 + (y - y_2)^2 + (z - z_2)^2 = l^2 \quad (2.3.27)$$

$$(x - x_3)^2 + (y - y_3)^2 + (z - z_3)^2 = l^2 \quad (2.3.28)$$

$w_n = x_n^2 + y_n^2 + z_n^2$ として, 式(2.3.26), 式(2.3.27), 式(2.3.28)について連立させると以下となる.

$$(x_1 - x_2)x - y_2y + (z_1 - z_2)z = \frac{w_1 - w_2}{2} \quad (2.3.29)$$

$$(x_1 - x_3)x - y_3y + (z_1 - z_3)z = \frac{w_1 - w_3}{2} \quad (2.3.30)$$

$$(x_2 - x_3)x + (y_2 - y_3)y + (z_2 - z_3)z = \frac{w_2 - w_3}{2} \quad (2.3.31)$$

整理して

$$y = a_1x + b_1 \quad (2.3.32)$$

$$z = a_2x + b_2 \quad (2.3.33)$$

ただし

$$a_1 = \frac{1}{d}\{(x_2 - x_1)(z_3 - z_1) - (x_3 - x_1)(z_2 - z_1)\} \quad (2.3.34)$$

$$b_1 = -\frac{1}{2d}\{(w_2 - w_1)(z_3 - z_1) - (w_3 - w_1)(z_2 - z_1)\} \quad (2.3.35)$$

$$a_2 = \frac{1}{d}\{(x_2 - x_1)y_3 - (x_3 - x_1)y_2\} \quad (2.3.36)$$

$$b_2 = \frac{1}{2d}\{(w_2 - w_1)y_3 - (w_3 - w_1)y_2\} \quad (2.3.37)$$

$$d = (z_2 - z_1)y_3 - (z_3 - z_1)y_2 \quad (2.3.38)$$

式(2.3.32), 式(2.3.33)を式(2.3.26)に代入して

$$(a_1^2 + a_2^2 + 1)x^2 + 2\{a_1b_1 + a_2(b_2 - z_1) - x_1\}x + \{b_1^2 + (b_2 - z_1)^2 + x_1^2 - l^2\} = 0 \quad (2.3.39)$$

簡単にするため係数をそれぞれ以下のおく.

$$A = a_1^2 + a_2^2 + 1 \quad (2.3.40)$$

$$B = 2\{a_1b_1 + a_2(b_2 - z_1) - x_1\} \quad (2.3.41)$$

$$C = \{b_1^2 + (b_2 - z_1)^2 + x_1^2 - l^2\} \quad (2.3.42)$$

式(2.3.39)は2次方程式 $Ax^2 + Bx + C = 0$ となるので $x$ について解くと

$$x = \frac{-B \pm \sqrt{B^2 - 4AC}}{2A} \quad (2.3.43)$$

X 軸の方向を考えて正を選択すると以下となる.

$$x = \frac{-B + \sqrt{B^2 - 4AC}}{2A} \quad (2.3.44)$$

これを式(2.3.32), 式(2.3.33)に代入して  $y, z$  を求められる. それぞれの変数は 3 つの入力角度 $\theta_n$  が既知であるとき一意に定まるので, 出力位置も一意に求められる.

以上より, 入力角度 $[\theta_1 \theta_2 \theta_3]$ から出力位置 $[x y z]$ を算出できた.

### 第3章 実験機の製作

#### 3.1 製作した実験機

図 3.1 に製作した実験機を示す。

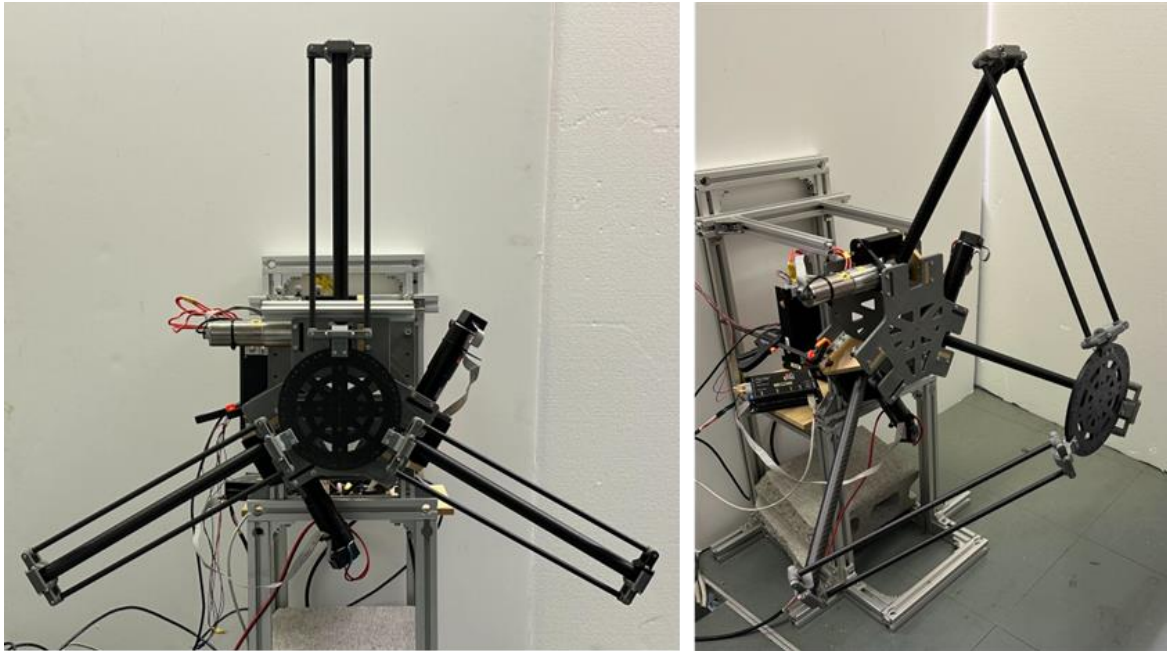


Fig3.1 Experimental manipulator

DC モータとモータドライバをそれぞれ 3 つずつ使用し、各リンクには軽くて丈夫なカーボン棒を採用している。ベースフレームと出力フレーム、その他の小さな部品については 3D プリンタで製作した。用いたフィラメント素材は Z-ASA pro で密度  $1.176\text{g/cm}^3$ 、曲げ強度  $45.60\text{MPa}$  である。

### 3.2 使用機材

使用している機材を以下に示す。図 3.2 は DC モータ DCX 35L 36V であり、図 3.3 はギアヘッド GPX37 A26:1、図 3.4 はエンコーダ ENX 16 EASY、図 3.5 はモータドライバ EPOS2 50/5 である。また、各パラメータを表 3.1 に示す。



Fig3.2 DC Motor



Fig3.3 Gear Head



Fig3.4 Encoder



Fig3.5 Motor Driver

Table3.1 Manipulator parts

Figure	Name	Model number	Remark
Fig3.2	DC Motor	DCX35L 36V	Maximum RPM : 12300 rpm
Fig3.3	Gear Head	GPX37 A26:1	Maximum Torque : 6.8 Nm
Fig3.4	Encoder	ENX 16 EASY	Pulse Number : 1024 pulse
Fig3.5	Motor Driver	EPOS 2 50/5	Position control and Angular velocity control

モータについては以下の条件を満たすように選定した。

- i) 最大許容トルクは、手先に取り付けるハンド機構や認識機構によって 1kg の負荷がかかると仮定した場合に加わる最大のトルク 4.56Nm 以上であること。
- ii) 高速化を目指すために 0.1 秒で 0 度から 90 度までの回転が可能となるギア比を含めた回転速度が 150rpm 以上であること。
- iii) 現在の角度を取得できるエンコーダが取り付け可能なこと。

### 3.3 モータの制御方法

モータの制御は、PC を用いて MATLAB とモータドライバのシリアル通信によって行った。システムの概略を図 3.6 に示す。エンコーダで角度の値をフィードバックしながらモータドライバによって位置指令や角速度指令を行う。

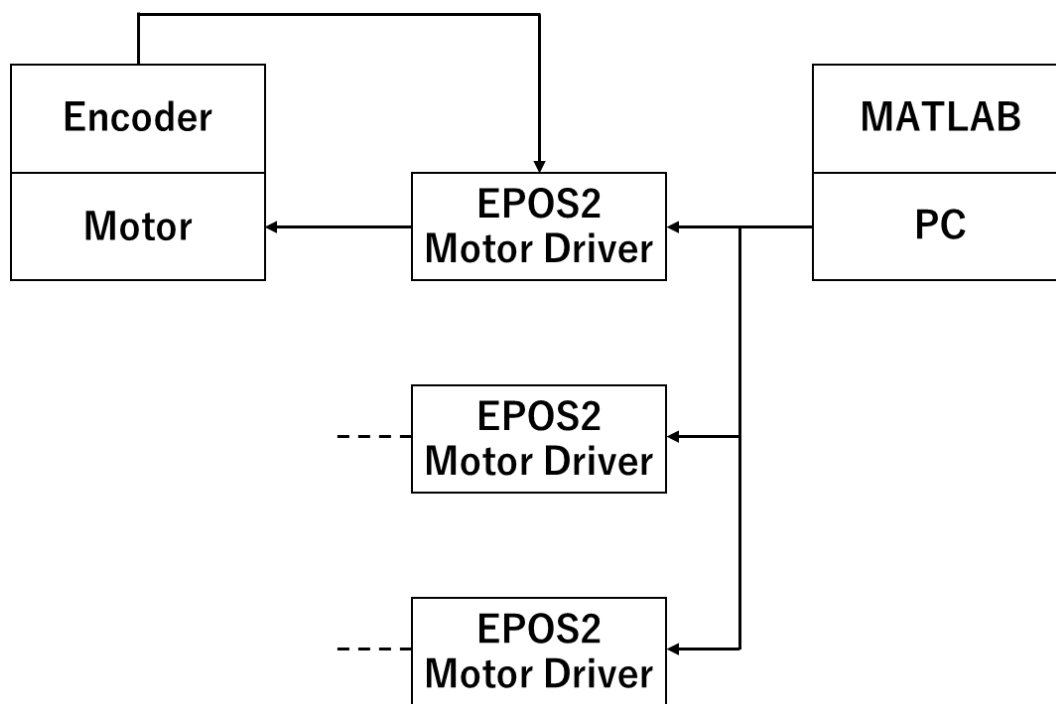


Fig3.6 Control system



## 第4章 実験機の精度検証

### 4.1 検証方法

実験機の精度を確認するため、シミュレーションでの可動範囲が製作した実験機で動作可能か検証した。実験環境を以下の図 4.1 に示す。

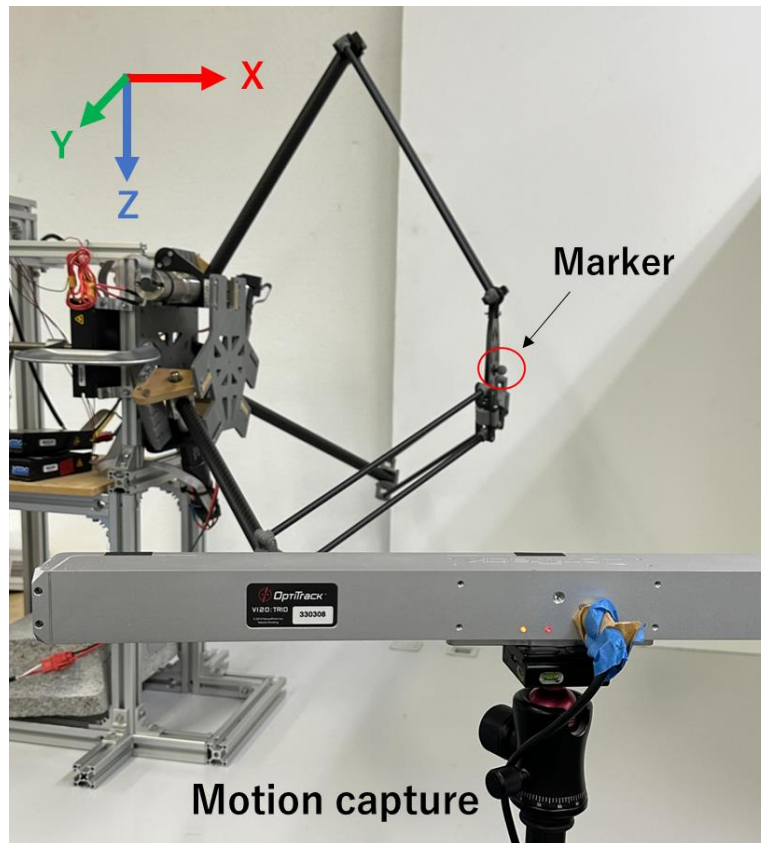


Fig4.1 Experimental environment

実験ではモーションキャプチャを用いて手先の位置情報を取得する。図 4.2 に使用するモーションキャプチャ Opti Track V120:Trio を示す。手先に反射マーカを取り付け、撮影することで並進 3 自由度の位置データを取得できる。



Fig4.2 Motion capture

## 4.2 シミュレーションの可動範囲

MATLAB を用いて 3 つの入力角度を 0 度から 85 度までそれぞれ変化させ、出力位置の範囲をシミュレーションした。これを XY 平面と XZ 平面に分けて図 4.3 に示す。青線が範囲の最大値であり、 $x=300\sim 500\text{mm}$  の範囲で  $y, z$  方向に広く可動できていることがわかる。実験予定のハウスのピーマン畑では畝間の中心から  $400\sim 600\text{mm}$  の範囲に果実は分布している。手先にハンドを取り付け、畝間を走行する移動機構に乗せての運用を想定しているため、シミュレーションで得られた可動範囲は十分であると考えられる。

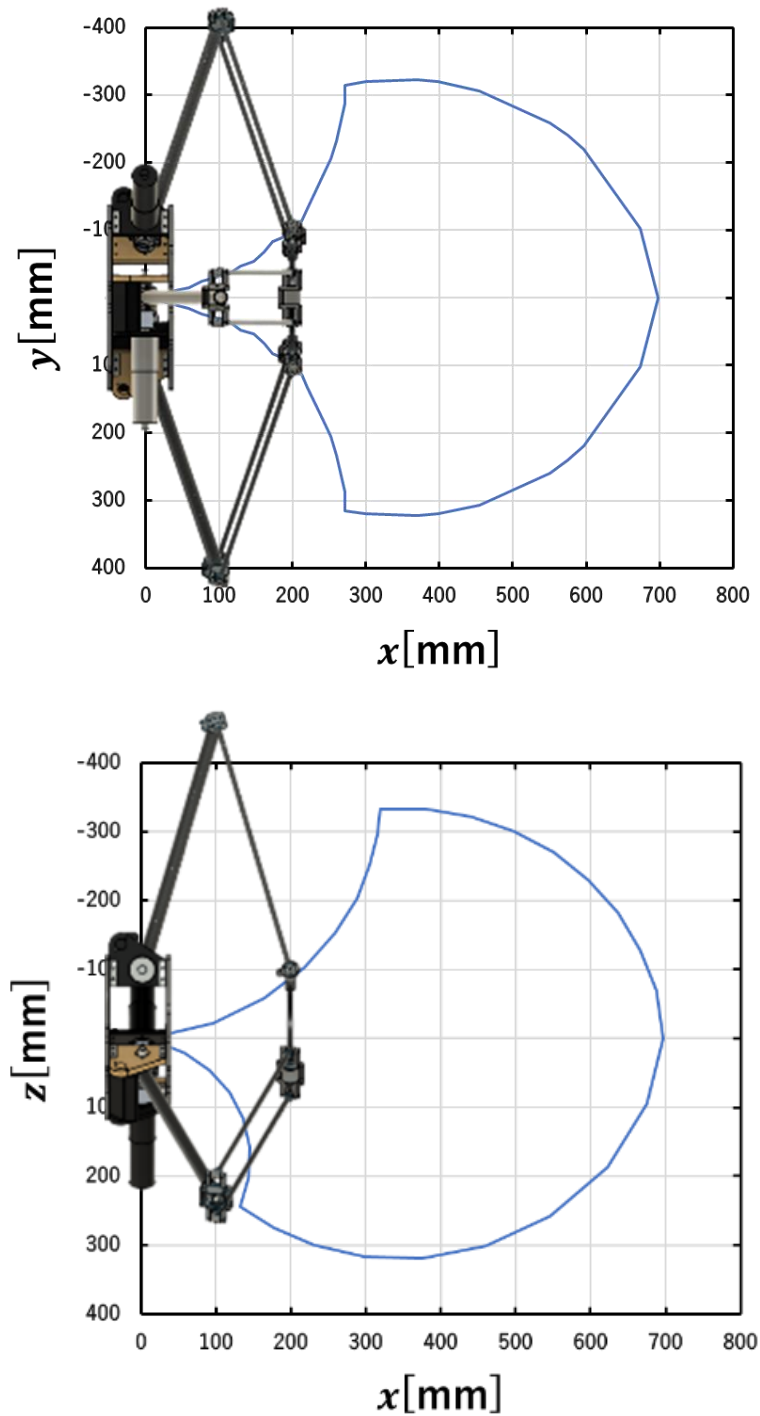


Fig4.3 Workspace for simulation

### 4.3 実験機の可動範囲

シミュレーションで得られた位置を実験機で動作させる実験を2つ行った。具体的にはXY平面とXZ平面の位置情報それぞれについて、前章で導出したパラレルメカニズムの逆運動学により角度に変換し、モータに入力した。モーションキャプチャによって取得した手先の位置情報を図4.3に重ね、その結果を図4.4に示す。青線がシミュレーションによる結果で赤線が実験によって得た手先の軌跡である。

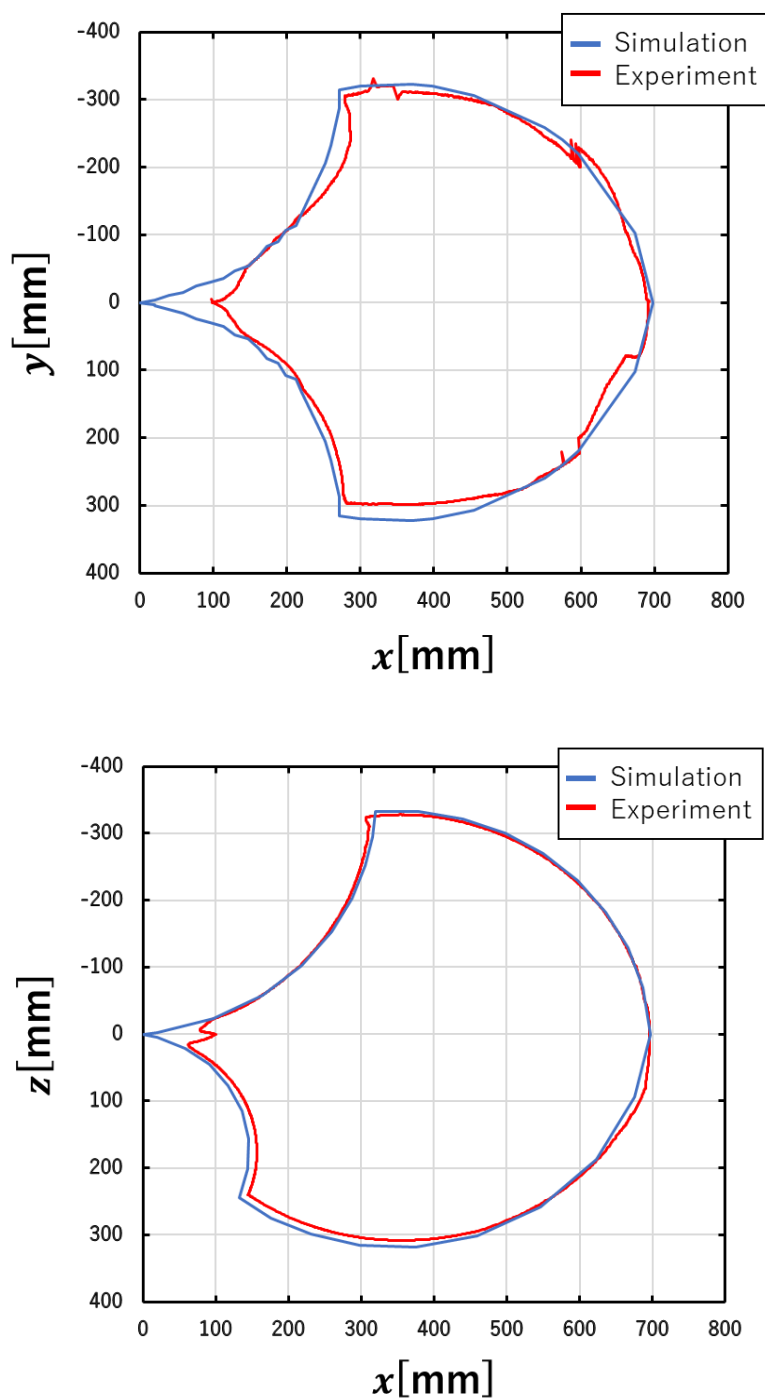


Fig4.4 Comparison of simulation and experimental workspaces

図 4.3 において初期段階で誤差が生じているのは、シミュレーションは初期位置が[000]であるが、実験機では動作できない位置であることから、実験の初期位置を[10000]としているためである。その後は概ね追従できており、水平駆動によって重力の影響を受ける Z 方向については、その影響を考慮した入力としているため、大きな誤差は見られなかった。この結果から実験機はシミュレーション通りの可動範囲であり、水平方向の駆動でも比較的精度は良いことがわかった。

## 第5章 実験準備

### 5.1 実験環境

実験環境を図 5.1 に示す。製作した実験機の手先に RGB 情報と深度情報を取得できるカメラを取り付け、手先とピーマンの相対距離をフィードバックしながら対象に向かい、正面 200mm で停止する実験を 3 通りの制御方法で行った。前章と同様に、モーションキャプチャを用いて手先の位置情報を取得する。カメラによる画像処理、各制御方法の詳細は次節以降で述べる。



Fig5.1 Experimental environment with camera

### 5.2 画像処理

使用するカメラ Intel® RealSense™ Depth Camera d435i を図 5.2 に示す。深度カメラが 2 つ、RGB カメラが 1 つ搭載されている。また、表 5.1 に深度、RGB それぞれの仕様を示す<sup>[8]</sup>。



Fig5.2 Depth and RGB camera<sup>[8]</sup>

Table5.1 Camera Specifications

	Depth	RGB
view angle (Horizontal×Vertical×Diagonal)	85.2×58×94 deg	69.4×42.5×77 deg
Maximum Frame rate	90 fps	60 fps
Maximum Frame Resolution	1280×720	1920×1080
Distance	0.1 ~ 10m	—
Size	90×25×25 mm	

深度カメラと RGB カメラでは画角が異なるため、深度カメラの画角を RGB カメラに揃えて使用した。また、解像度は  $640 \times 480$  として最大のフレームレート 60 fps を出せるようにした。

例として  $640 \times 480$  で取得したピーマンの深度画像を図 5.3 に RGB 画像を図 5.4 に示す。座標軸の向きは図 5.1 と同様であり、図 5.4 と図 5.5 における紙面の奥側に垂直な向きが X 軸、水平方向右向きが Y 軸、鉛直下向きが Z 軸である。次項より、この 2 つの画像を用いてピーマンまでの距離を計算する。ただし、本研究では画像処理を簡単とするため白い背景を使っている。

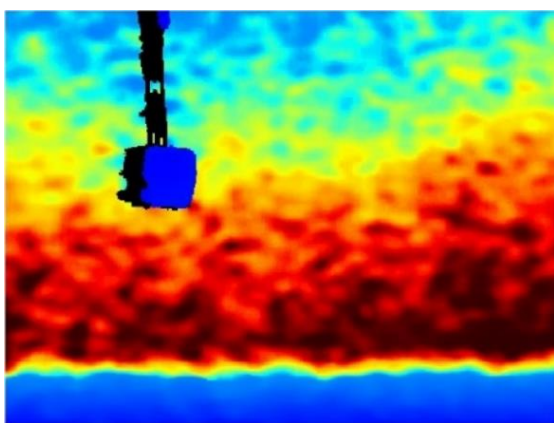


Fig5.3 Depth image



Fig5.4 RGB image

### 5.2.1 中心位置検出

図 5.4 の RGB 画像を用いて以下の手順でピーマンの中心位置を求める。

- ① RGB 画像をグレースケール化
- ② 色のしきい値を用いて 2 値化
- ③ エッジからの穴の塗りつぶし
- ④ イメージを縮小し、小さいオブジェクトを削除

処理の手順①から④における画像を図 5.5 から図 5.8 に示す。

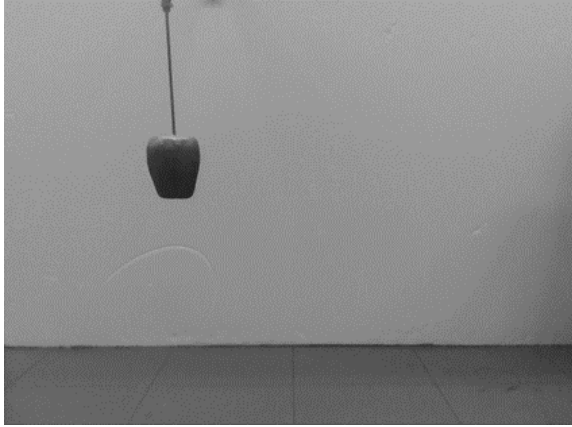


Fig5.5 ①Gray scaling

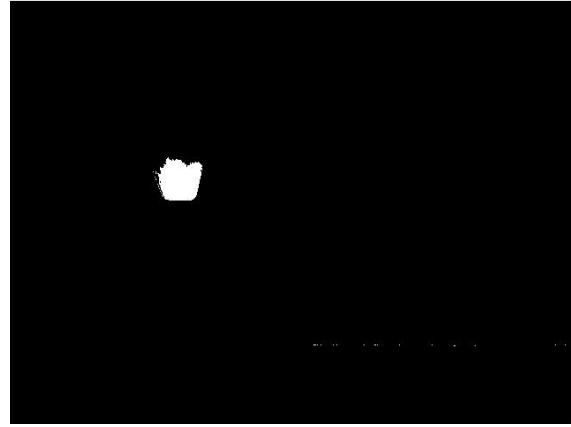


Fig5.6 ②Binarization

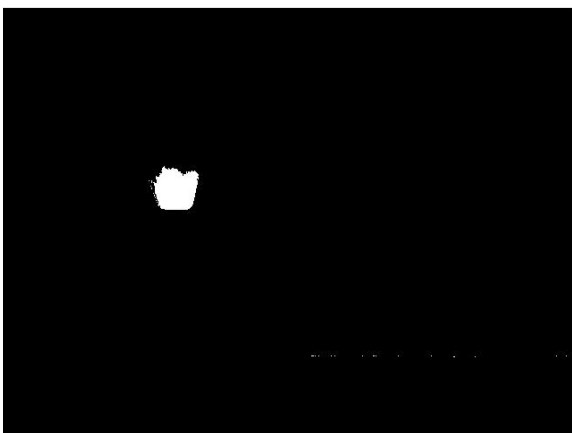


Fig5.7 ③Hole Fill

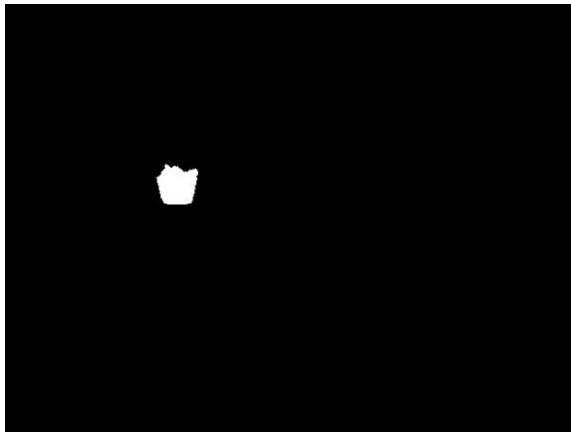


Fig5.8 ④Reduce image size and  
Remove small objects

今回は図 5.6 から図 5.7 に変化がほとんど見られないが、仮に図 5.6 におけるピーマンの内側に黒い部分があった場合に塗りつぶしを行う。この結果よりピーマンの中心位置の 2 次元のピクセル数を得られる。ピクセル数は左上が[0 0]であり、右下が[640 480]である。

### 5.2.2 距離検出

図 5.3 の深度画像にピクセル数を当てはめてカメラとピーマンの中心の相対距離を計算する。X 軸方向、つまり単純な奥行きの情報にピクセル数を当てはめるだけで値が得られる。これを用いて Y 軸、Z 軸方向の距離を計算する。以降では中心位置のピクセル数を $[p_y p_z]$ 、カメラで取得するピーマンまでの 3 軸の距離を $[c_x c_y c_z]$ とおく。

まず、Y 軸について考える。図 5.9 に XY 平面上のカメラとピーマンの位置関係のイメージを示す。

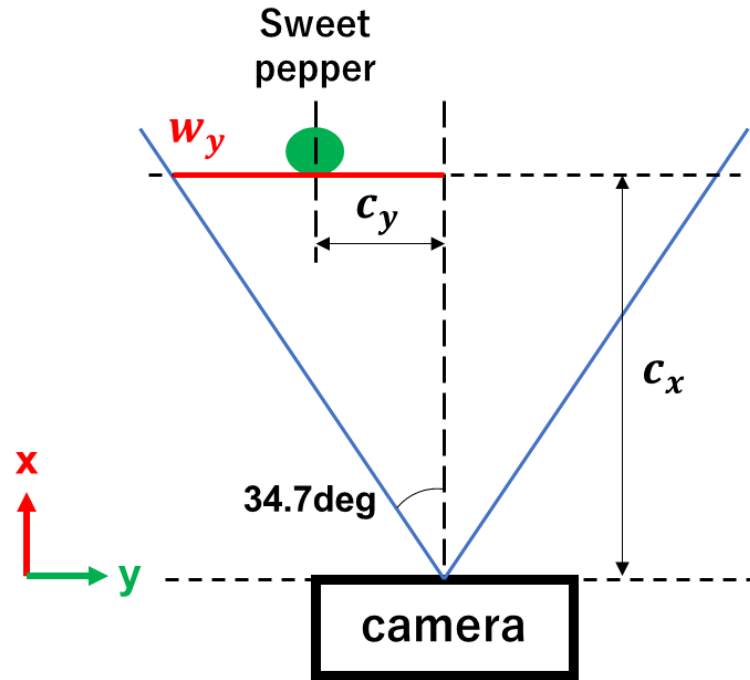


Fig5.9 Positional image of the camera and sweet pepper in XY plane

図 5.9 における青線はカメラの画角，赤線 $w_y$ は画像の中心から端までの長さ，緑図はピーマンを表しており，表 5.1 より水平方向の画角を 69.4 度としている． $w_y$ は X 軸方向の相対距離 $c_x$ と画角を用いて以下の式のように求められる．

$$w_y = c_x \tan 34.7 = 0.6924c_x \quad (5.2.1)$$

ピーマンの中心のピクセル数 $p_y$ は 0 から 640 の範囲であることを考慮して，Y 軸方向の相対距離 $c_y$ は以下の式で求められる．

$$c_y = w_y \left( \frac{p_y}{320} - 1 \right) = 0.6924c_x \left( \frac{p_y}{320} - 1 \right) \quad (5.2.2)$$

次に，Z 軸方向について考える．図 5.10 に XZ 平面上のカメラとピーマンの位置関係のイメージを示す．Y 軸の場合と同様に考え，鉛直方向の画角は 42.5 度であることから， $h_z$ は以下の式となる．

$$h_z = c_x \tan 21.25 = 0.3889c_x \quad (5.2.3)$$

Z 軸方向のピクセル数 $p_z$ は 0 から 480 の範囲であるため，相対距離 $c_z$ は以下の式で求められる．

$$c_z = h_z \left( \frac{p_z}{240} - 1 \right) = 0.3889c_x \left( \frac{p_z}{240} - 1 \right) \quad (5.2.4)$$

以上より，RGB 画像と深度画像を用いてカメラとピーマンの相対距離を求められることがわかった．



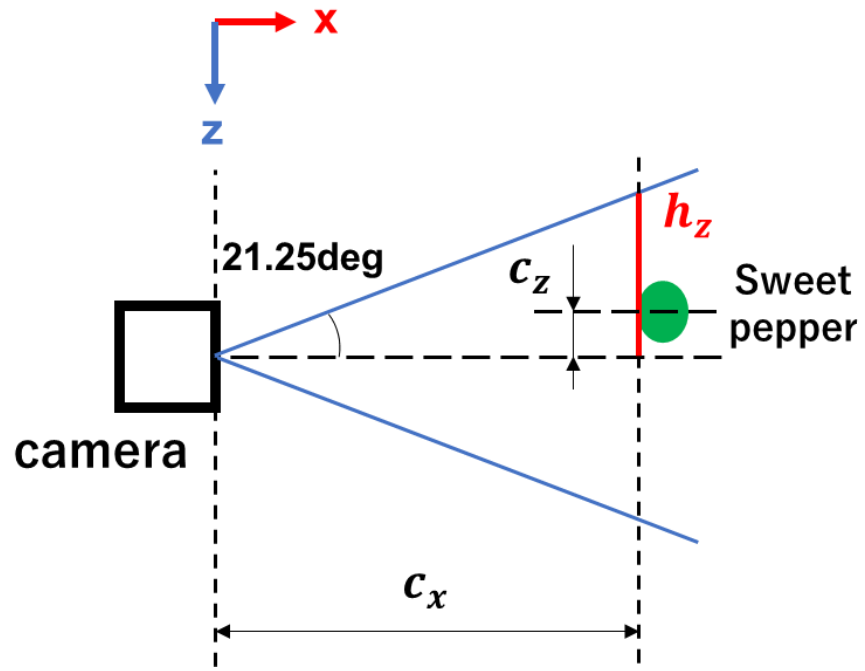


Fig5.10 Positional image of the camera and sweet pepper in XZ plane

### 5.3 制御方法

ブロック線図で使用する記号を以下のように定義する.

- $y$  : 手先位置  $[x \ y \ z]$   
エンコーダによって取得したリンクの角度から順運動学によって推測する.
- $r$  : 相対距離の目標位置  $[r_x \ r_y \ r_z]$   
ピーマンの正面 200mm で停止するように  $[200 \ 0 \ 0]$  とした.
- $r_v$  : 手先の目標速度  $[r_{vx} \ r_{vy} \ r_{vz}]$
- $u$  : モータ入力値  $[u_1 \ u_2 \ u_3]$  (角度, 角速度)
- $u_p$  : モータ入力値  $[u_{px} \ u_{py} \ u_{pz}]$  (位置)
- $c$  : 前節の方法で, カメラによって取得した手先位置とピーマンの相対距離  $[c_x \ c_y \ c_z]$
- $e$  : 位置偏差  $[e_x \ e_y \ e_z]$   
目標位置とカメラで取得した距離の差.
- $d$  : 重力による外乱  $[0 \ 0 \ d_z]$

#### 5.3.1 局所位置制御

局所位置制御のブロック線図を図 5.11 に示す.  $Z^{-1}$ ブロックは前回値の保持, inverse kinematics ブロックはパラレルメカニズムの逆運動学を表す.

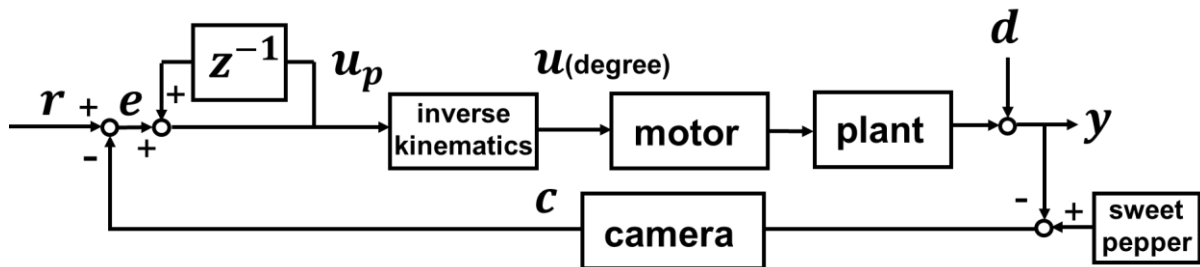


Fig5.11 Block diagram for Position control

局所位置制御では, 以下の式(5.3.1)に示すようにひとつ前の入力値に現在の位置偏差を加算し, 逆運動学によって 3 軸の並進位置を 3 つの角度に変換して角度入力を決している. 式(5.3.1)における  $T^{-1}$  は逆運動学を表す. モータは入力した角度に到達するまで可能な限り最大の角速度で動作させる.

$$\begin{bmatrix} u_{1_i} \\ u_{2_i} \\ u_{3_i} \end{bmatrix} = T^{-1} \begin{bmatrix} u_{px_{i-1}} + e_{x_i} \\ u_{py_{i-1}} + e_{y_i} \\ u_{pz_{i-1}} + e_{z_i} \end{bmatrix} \quad (5.3.1)$$

### 5.3.2 角速度指令

角速度指令のブロック線図を図 5.12 に示す。  $K_{px}$ ,  $K_{py}$ ,  $K_{pz}$  ブロックは 3 軸それぞれの比例ゲインを表す。

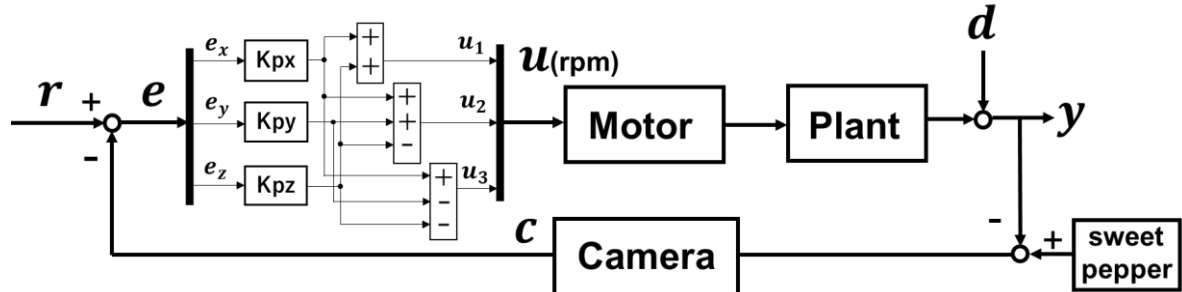


Fig5.12 Block diagram for Angular velocity control

角速度指令では、パラレルメカニズムにおいて並進 3 軸方向を考慮したとき、それぞれのリンクの動作方向が手先の方向で決定されることを考慮する。例として、図 5.13 に  $[x \ y \ z] = [100 \ 0 \ 0]$  から  $[400 \ 0 \ 0]$  と  $x$  のみ動かした場合、図 5.14 に  $[x \ y \ z] = [400 \ -200 \ 0]$  から  $[400 \ 200 \ 0]$  と  $y$  のみ動かした場合、図 5.15 に  $[x \ y \ z] = [400 \ 0 \ -200]$  から  $[400 \ 0 \ 200]$  と  $z$  のみ動かした場合について 3 つのリンク角度をそれぞれ示す。  $xyz$  それぞれの軸方向と各リンクの番号については図 2.2 のモデル図と同様である。

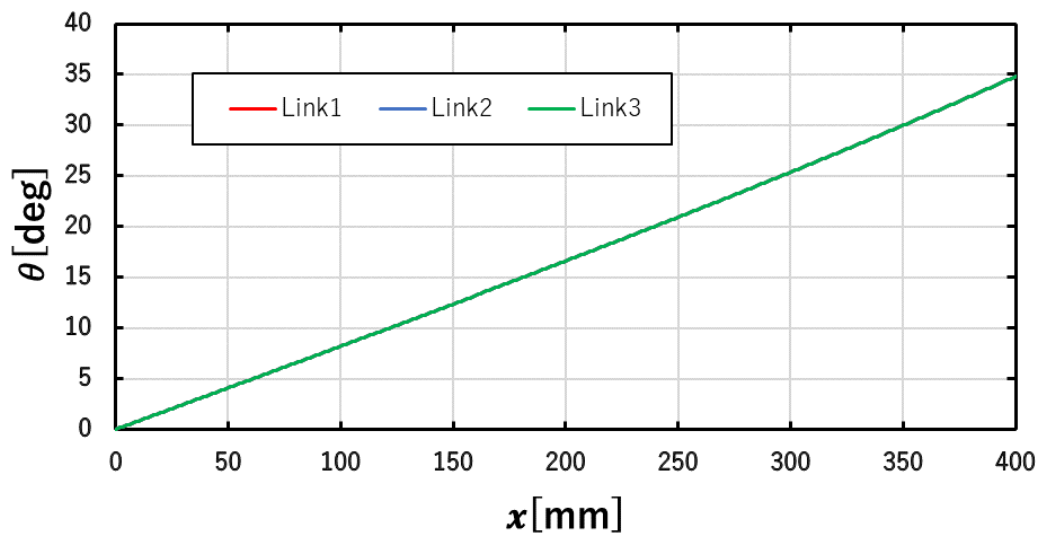


Fig5.13 Link angle vs  $x$

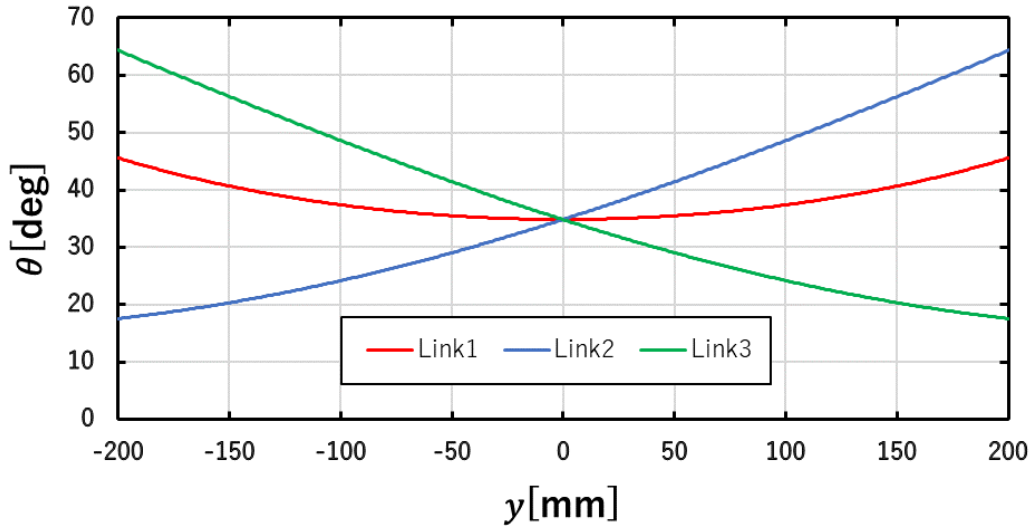


Fig5.14 Link angle vs  $y$

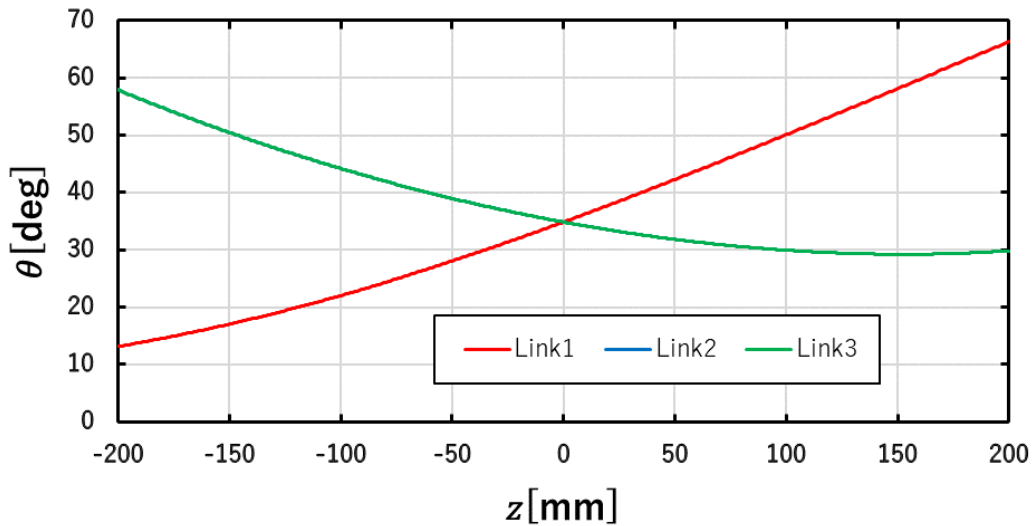


Fig5.15 Link angle vs  $z$

図 5.13 ではリンクはすべて等しい角度であるため、線は重なっている。また、 $y = z = 0\text{mm}$  で  $x$ のみ変位している場合、リンクは線形となることがわかる。図 5.14 では $y$ の位置変位に対してリンク 1 の角度変位は小さく、リンク 2 とリンク 3 は $y = 0\text{mm}$  を中心に線対称であることがわかる。つまり、 $y$ が正方向に動く場合、リンク 2 の角度は正方向であるがリンク 3 の角度は負方向に動く。図 5.15 では $z$ の変位に対してリンク 2 とリンク 3 の角度が等しいため、線は重なっている。 $z$ が正方向に動く場合、リンク 1 の角度は正方向となるがリンク 2 とリンク 3 の角度は負方向に動く。以上のことから、以下の式(5.3.2)から式(5.3.4)のように 3 軸方向の偏差にゲインを乗算して、それらの足し引きによって角速度入力を計算する式を定めた。この関係は図 5.12 のブロック線図にも表している。ただし、式(5.3.2)においては、 $y$ の位置変位に対してリンク 1 の角度変位は小さいため、偏差 $e_y$ を無視している。

$$u_1 = K_{px}e_x + K_{pz}e_z \quad (5.3.2)$$

$$u_2 = K_{px}e_x + K_{py}e_y - K_{pz}e_z \quad (5.3.3)$$

$$u_3 = K_{px}e_x - K_{py}e_y - K_{pz}e_z \quad (5.3.4)$$

各ゲインについてはシミュレーションによって決定した。詳細は次節で述べる。

### 5.3.3 軌道制御

軌道制御のブロック線図を図 5.16 に示す。Jacobi ブロックはヤコビ行列を表す。

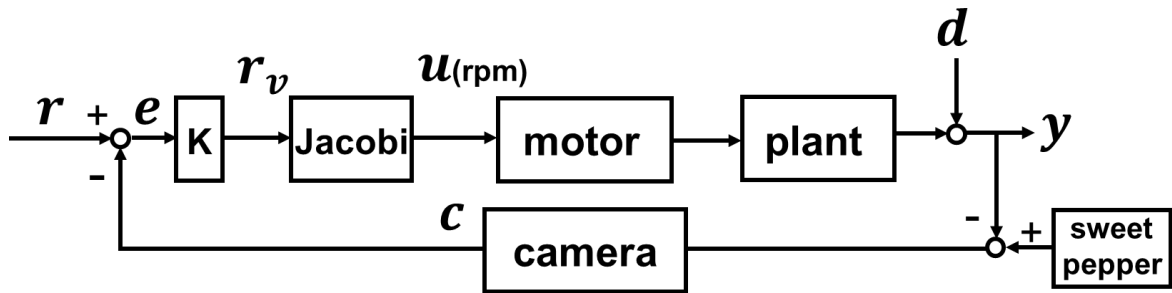


Fig5.16 Block diagram for Routing control

軌道制御ではヤコビ行列を用いるため、パラレルメカニズムのヤコビ行列を以下のように変形していく。

パラレルメカニズムの逆運動学 $T^{-1}$ により手先位置とリンク角度の関係式は以下となる。

$$\begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \end{bmatrix} = T^{-1} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad (5.3.5)$$

ヤコビ行列 $J$ は位置の微小変位と角度の微小変位との関係を表すので以下となる。

$$J = \begin{bmatrix} \frac{\partial \theta_1}{\partial x} & \frac{\partial \theta_1}{\partial y} & \frac{\partial \theta_1}{\partial z} \\ \frac{\partial \theta_2}{\partial x} & \frac{\partial \theta_2}{\partial y} & \frac{\partial \theta_2}{\partial z} \\ \frac{\partial \theta_3}{\partial x} & \frac{\partial \theta_3}{\partial y} & \frac{\partial \theta_3}{\partial z} \end{bmatrix} \quad (5.3.6)$$

式(5.3.6)を以下のように変形する。

$$\begin{bmatrix} d\theta_1 \\ d\theta_2 \\ d\theta_3 \end{bmatrix} = J \begin{bmatrix} dx \\ dy \\ dz \end{bmatrix} \quad (5.3.7)$$

式(5.3.7)を時間 $t$ で微分すると以下となる。

$$\begin{bmatrix} \frac{d\theta_1}{dt} \\ \frac{d\theta_2}{dt} \\ \frac{d\theta_3}{dt} \end{bmatrix} = J \begin{bmatrix} \frac{dx}{dt} \\ \frac{dy}{dt} \\ \frac{dz}{dt} \end{bmatrix} \quad (5.3.8)$$

式(5.3.8)は手先速度に対するリンクの角速度を表すため、要求する手先の速度を出すために必要なリンクの角速度はヤコビ行列を用いることで求められる。

以上を用いて、図 5.16 の軌道制御は、ゲイン K ブロックによって位置偏差を要求する手先の速度である目標速度  $r_v$  に変換し、現在位置に基づいたヤコビ行列によって目標速度から各リンクの角速度の目標値に変換する制御方法となっている。

## 5.4 シミュレーション

角速度指令、軌道制御のそれぞれのゲインについて、初期位置からのシミュレーションを行い、その結果による試行錯誤によって決定し、実験に使用するゲインの基準とした。シミュレーションには Simulink を使用した。

### 5.4.1 角速度指令のシミュレーション

Simulink 上で使用するブロック線図を図 5.17 に示す。

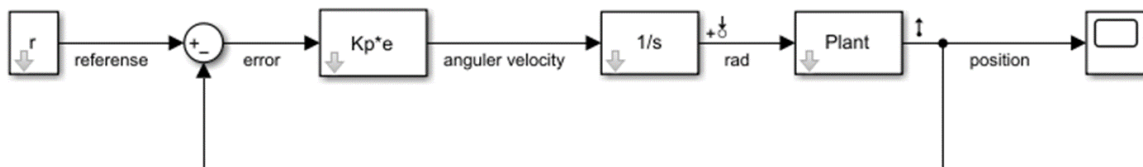


Fig5.16 Block diagram for Angular velocity control on Simulink

シミュレーションでは、カメラによるフィードバックはできないため、手先の目標位置を[400 -200 -150]と定めて初期位置[100 0 0]から動作させ、手先の位置情報をフィードバックした。リンク角度はモータ入力値である角速度を積分して求めた。サンプリング時間については 1 ステップ 0.045s とした。これはカメラによる 60fps での画像取得と画像処理の時間である。最終的に決定したゲインにおけるシミュレーション結果を図 5.17 に示す。重力による外乱を考慮しているため、最初 Z 軸方向の変位が遅れている。収束時間は約 2.3 秒となっており、これよりゲインを上げると収束時間は短くなるがオーバーシュートが目立つようになるため図 5.17 を結果とするゲインを実験の基準として決定した。

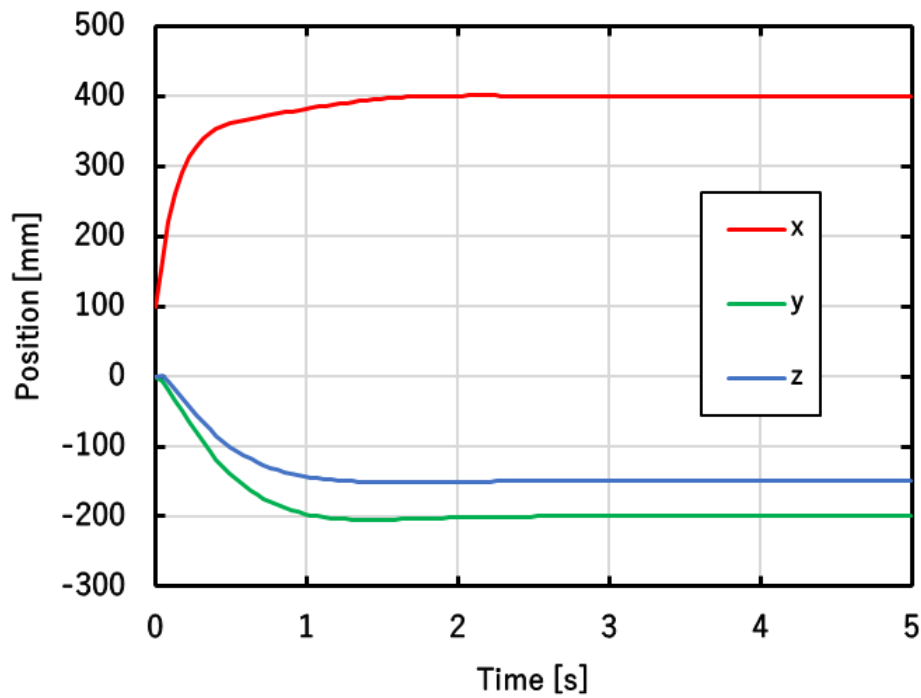


Fig5.17 Simulation result of Angular velocity control

#### 5.4.2 軌道制御のシミュレーション

Simulink 上で使用するブロック線図を図 5.18 に示す。

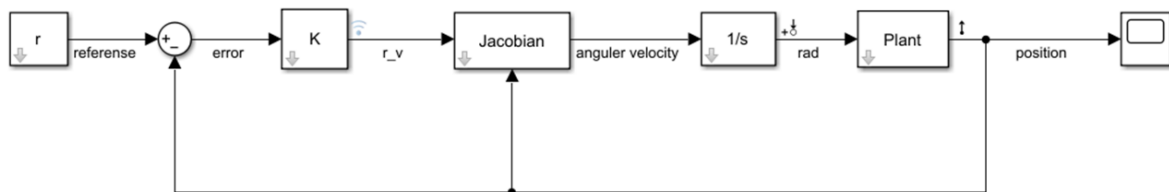


Fig5.18 Block diagram for Routing control on Simulink

目標位置や初期位置などのパラメータは角速度指令のシミュレーションと同様である。手先の目標速度と現在位置を用いて Jacobian ブロックで角速度の計算をしている。ヤコビ行列の計算時間を考慮し、サンプリング時間については 1 ステップ 0.2s とした。最終的に基準として決定したゲインにおけるシミュレーション結果を図 5.19 に示す。角速度指令の際に見られた Z 軸方向の変位の遅れはなかった。収束時間は約 2.2 秒となった。

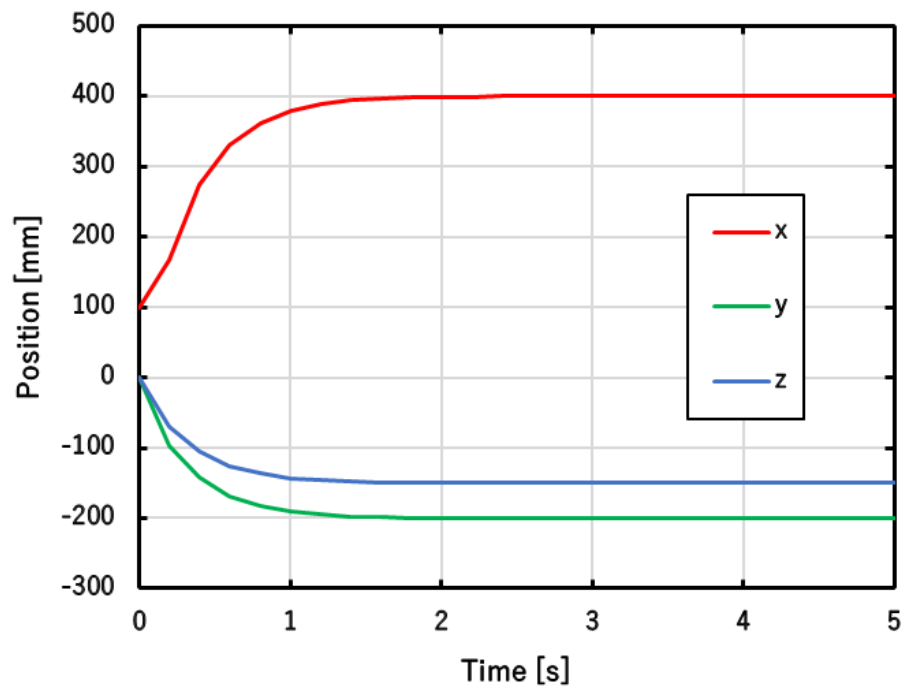


Fig5.19 Simulation result of Routing control



## 第6章 実験結果

### 6.1 局所位置制御

局所位置制御の実験結果について図 6.1 にモーションキャプチャを用いて取得した時間と手先位置の関係, 図 6.2 に手先の軌跡, 図 6.3 に手先の速度を示す. 図 6.2 における緑点はピーマンの位置[610.28 -139.11 -104.0]mm である. ピーマンの正面から 200mm の位置が目標値であるため, 収束する理想の位置は[410.28 -139.11 -104.0]mm となる.

動作開始直後は Y, Z への変位が小さく, X 軸に沿って直線的に動作している. これは 3 つのモータに同じ最大角速度を与えているからである. ピーマンの正面をとらえるまで計 3 回の動作を行っており, Y 軸方向では大きなオーバーシュートが見られる. 手先の最大速度は 2258.51mm/s となっており, 収束時間は 2.32s, 最終的な誤差 $[e_x \ e_y \ e_z]$ は[-4.946 3.372 0.268]mm である. 入力位置に到達するまで待つ時間が生じるために, 修正が遅くなるので最大速度に対して収束時間が長くなってしまう. 局所位置制御は高速な動作が可能であるが, 収束時間や精度が悪いことがわかった.

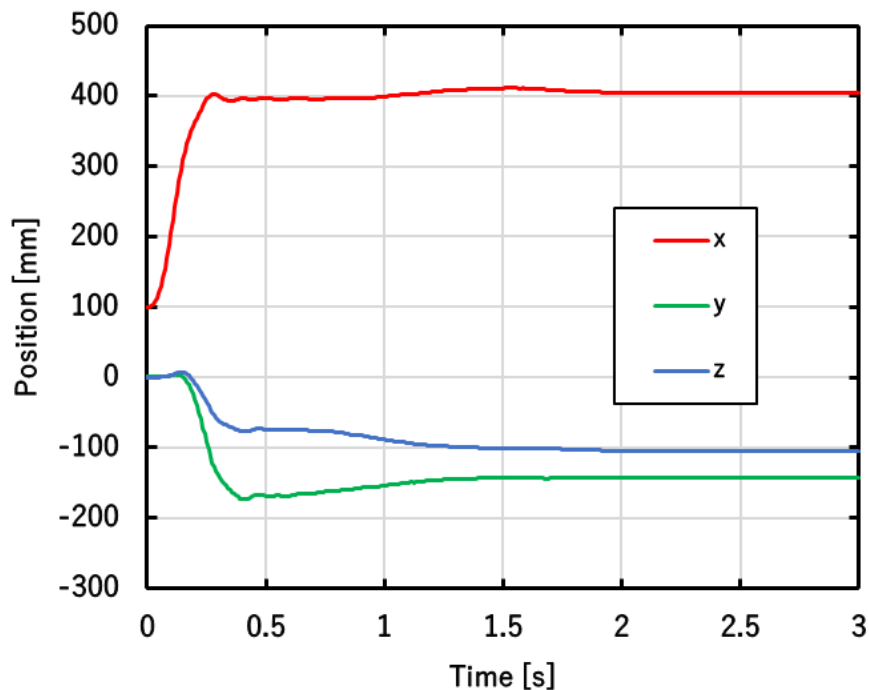


Fig6.1 Experimental results Position vs Time of Position control

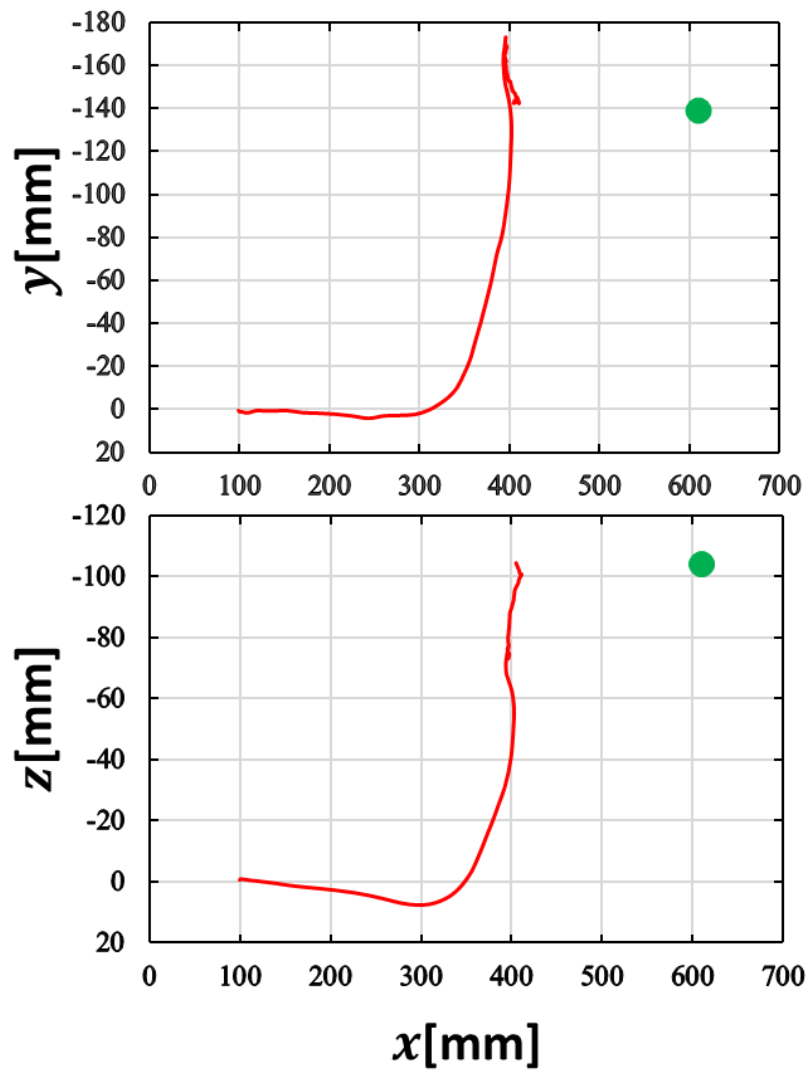


Fig6.2 Experimental results Trajectory of Position control

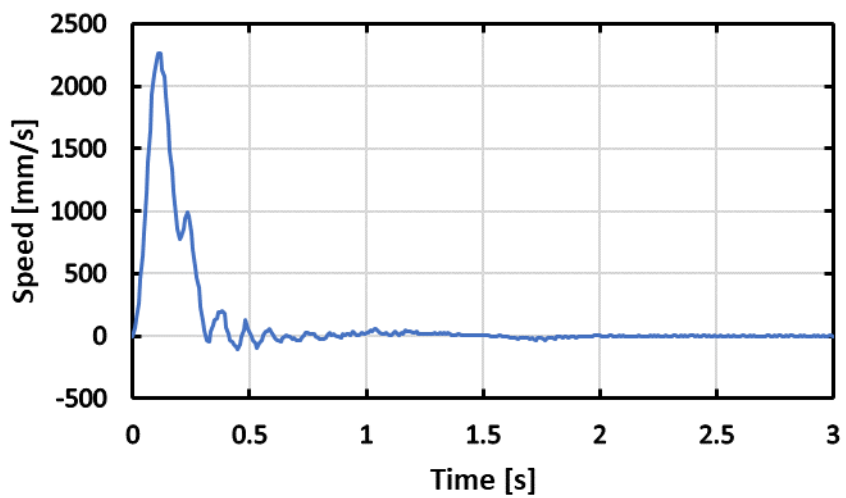


Fig6.3 Experimental results Speed vs Time of Position control

## 6.2 角速度指令

角速度指令の実験結果について図 6.4 にモーションキャプチャを用いて取得した時間と手先位置の関係, 図 6.5 に手先の軌跡, 図 6.6 に手先の速度を示す.

Y 軸方向にオーバーシュートがみられ, 動作開始直後から Y 軸と Z 軸方向に動作していることがわかる. 角速度指令は局所位置制御の場合と異なり, ゲインによって定めた異なる角速度を 3 つのモータに入力しているからである. 手先の最大速度は 1648.61mm/s となっており, 収束時間は 1.42s, 最終的な誤差 $[e_x \ e_y \ e_z]$ は $[0.548 \ 0.220 \ 0.310]$ mm である. サンプル時間は 1 ステップ 0.04s から 0.05s 程度となっているため, 素早い修正が可能となり, 収束は早くなっている. 角速度指令はある程度的高速な動作が可能であり, 収束が早く精度が高いことがわかった.

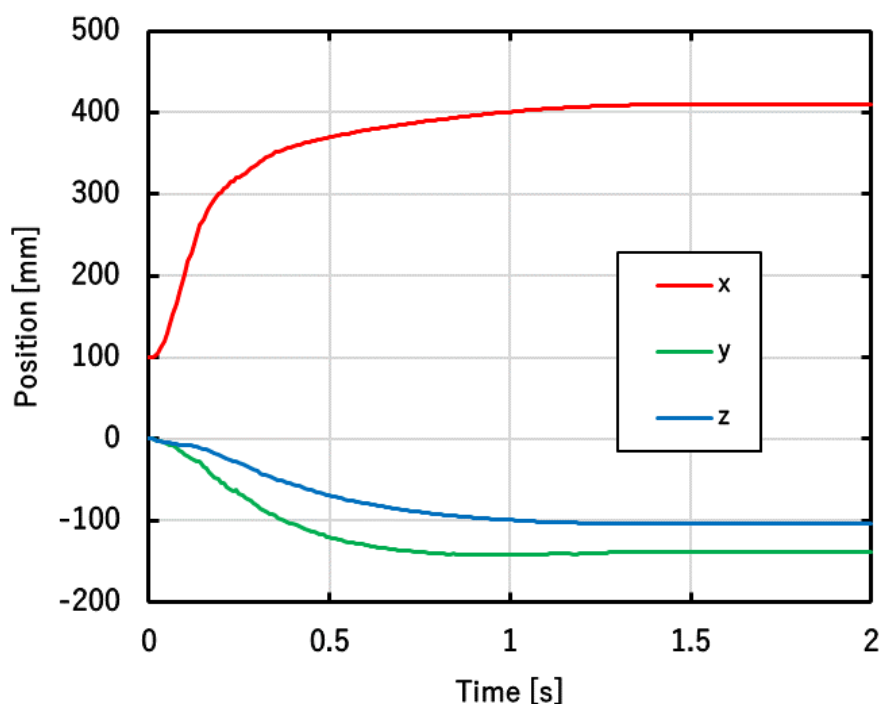


Fig6.4 Experimental results Position vs Time of Angular velocity control

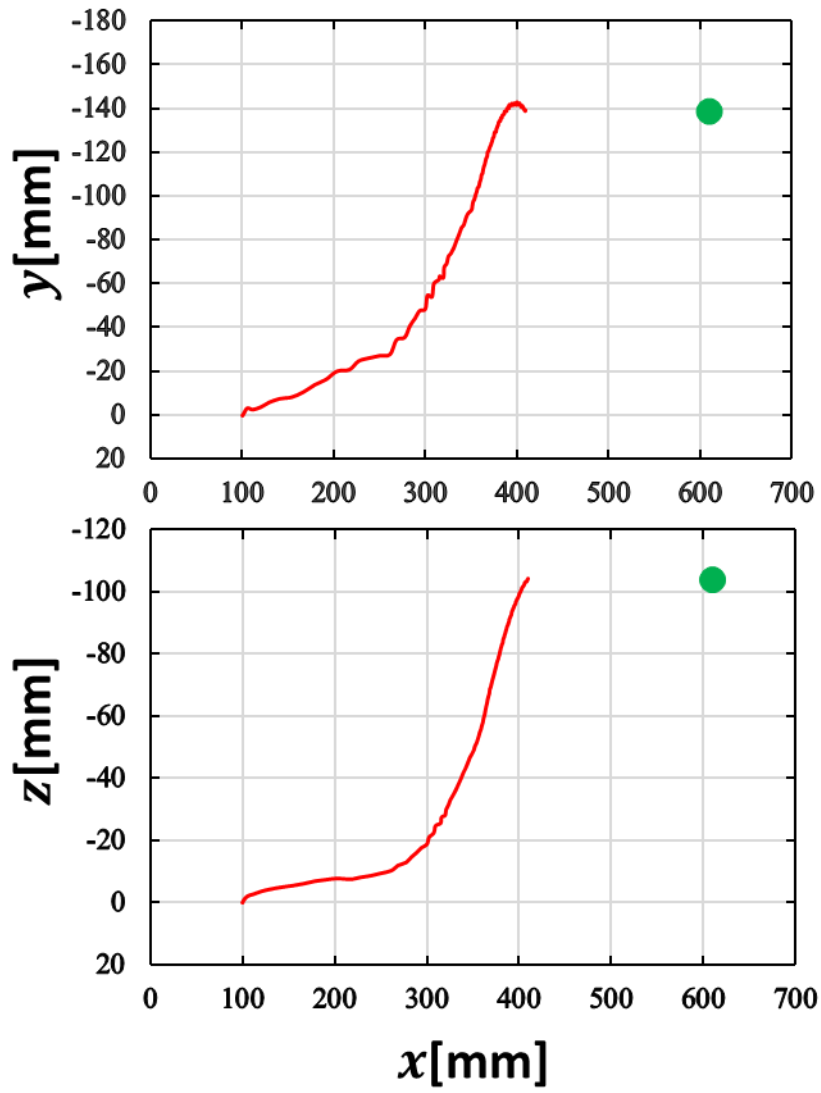


Fig6.5 Experimental results Trajectory of Angular velocity control

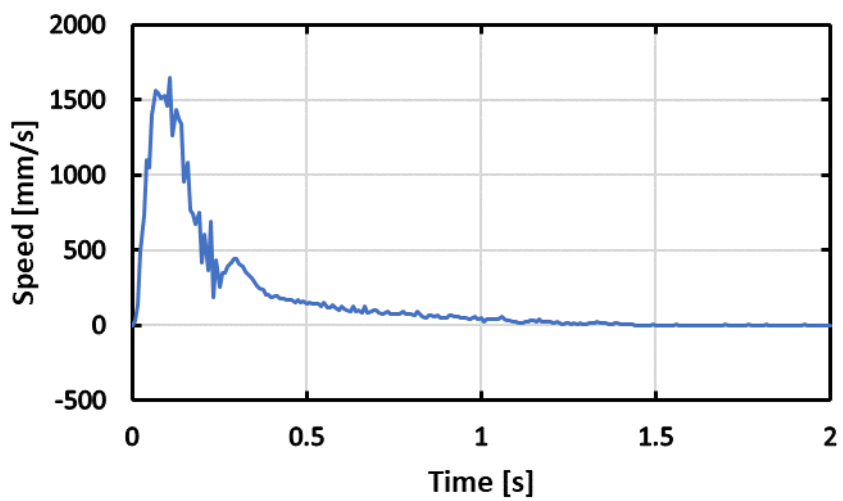


Fig6.6 Experimental results Speed vs Time of Angular velocity control

### 6.3 軌道制御

軌道制御の実験結果について図 6.7 にモーションキャプチャを用いて取得した時間と手先位置の関係, 図 6.8 に手先の軌跡, 図 6.9 に手先の速度を示す.

X 軸方向にオーバーシュートが見られ, 角速度指令と同様の理由で動作開始から Y 軸と Z 軸方向に変位している. 手先の最大速度は 678.74mm/s となっており, 収束時間は 3.65s, 最終的な誤差 $[e_x \ e_y \ e_z]$ は $[0.801 \ -0.480 \ 0.131]$ mm である. ヤコビ行列の複雑な計算時間によってサンプリング時間は 1 ステップ 0.2s から 0.3s 程度となっており, 角速度指令よりも遅くなっているため, 収束時間が長くなってしまった. 軌道制御は低速な動作となり, 収束が遅いが精度は高いことがわかった.

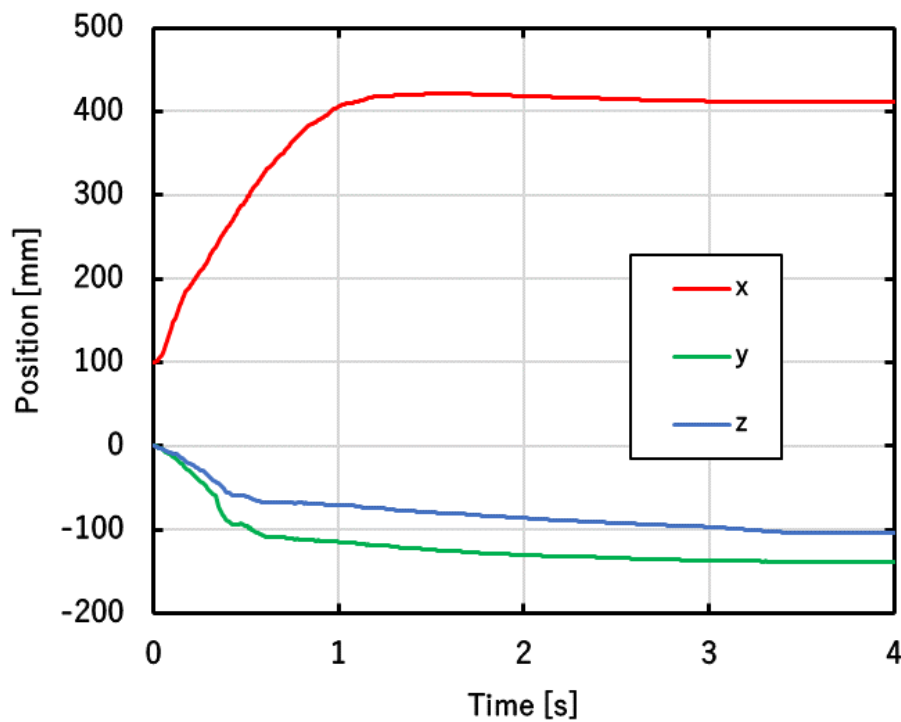


Fig6.7 Experimental results Position vs Time of Routing control

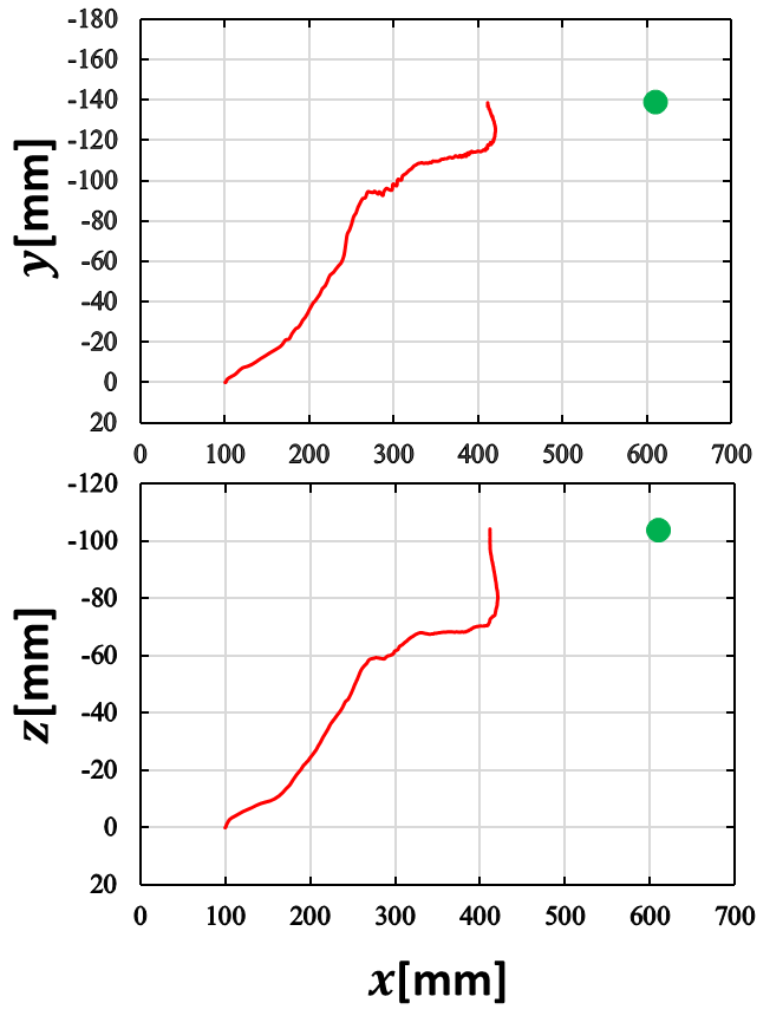


Fig6.8 Experimental results Trajectory of Routing control

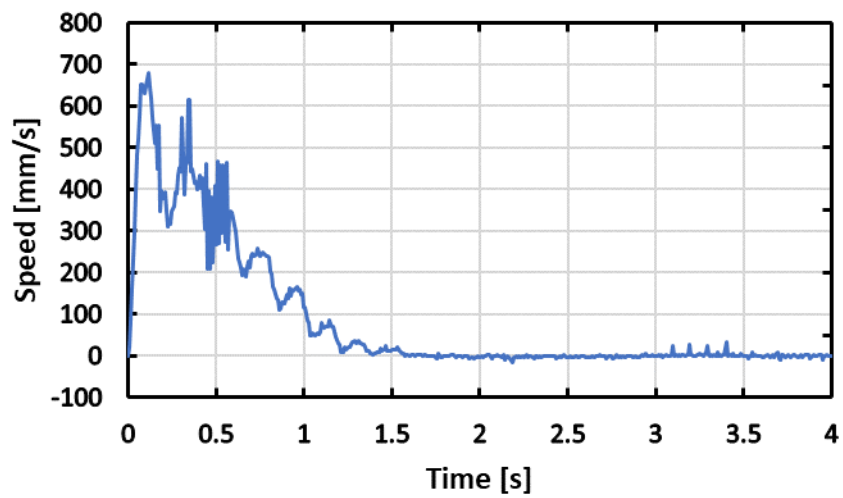


Fig6.9 Experimental results Speed vs Time of Routing control

## 6.4 結果の比較

3つの制御方法について、手先の軌跡である図 6.2, 図 6.5, 図 6.8 を重ね、それを図 6.10 に示す。赤線, 青線, 黄色線の順で局所位置制御, 角速度指令, 軌道制御である。また, 最大速度, 収束時間, 最終的な誤差について比較した結果を表 6.1 に示す。

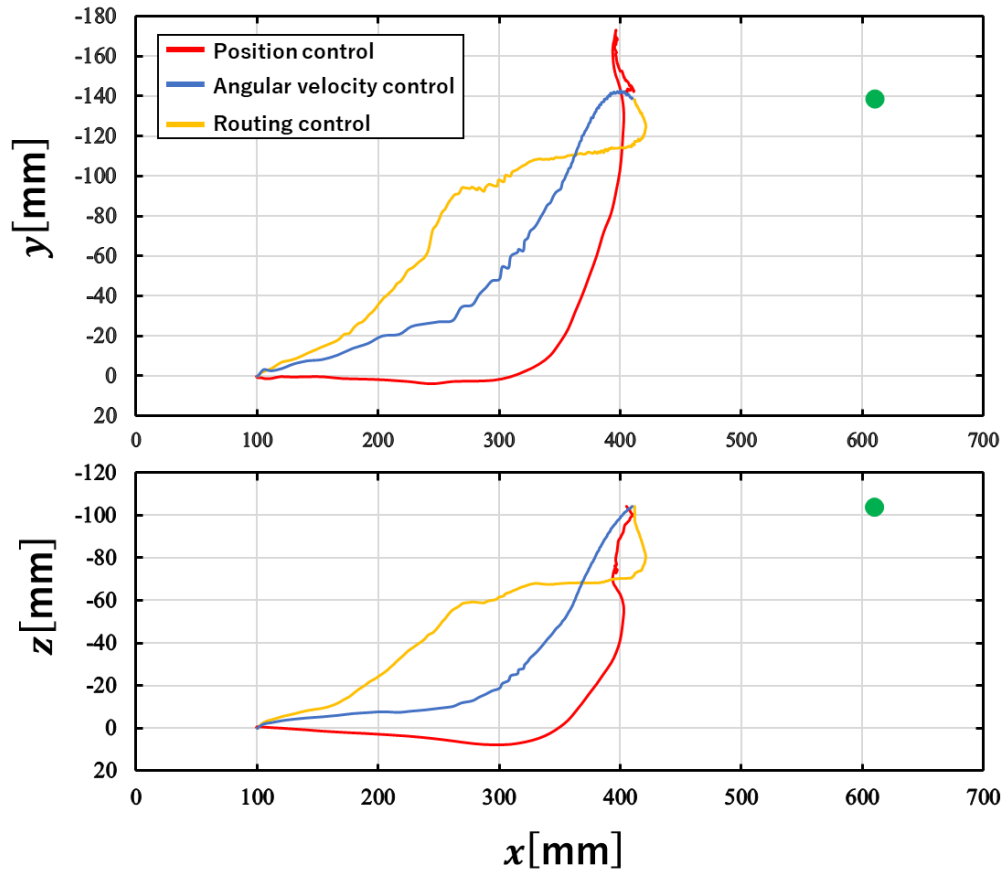


Fig6.10 Comparison of Trajectories

Table6.1 Comparison Experimental results

	Max speed [mm/s]	Convergence Time [s]	Error [mm]		
			$e_x$	$e_y$	$e_z$
Position control	2258.51	2.32	-4.946	3.372	0.268
Angular velocity control	1648.61	1.42	0.548	0.220	0.310
Routing control	678.74	3.65	0.801	-0.480	0.131

図 6.10 より制御方法によって軌跡が大きく異なることがわかる。特に、軌道制御はヤコビ行列の特性から最も素早く Y 軸と Z 軸方向へ動作しており、直線的にピーマンに向かっており、葉やほかの果実などの障害物を避けるための経路設計を行う際に、軌道制御は優位に働くと考える。また、表 6.1 からわかるように、最大速度は局所位置制御が、収束時間と最終的な誤差については角速度指令が最も良い結果となった。軌道制御の最大速度が遅い原因としては、ヤコビ行列の特性から最初リンク 2 が負方向へ動作するため、ゲインを上げすぎると実験機が許容できる角度-10 度を超えてしまうことから、一定のゲインに抑える必要があるためである。以上の結果から、単純に高速で動作させたい場合は局所位置制御を、収束時間を早めたい場合は角速度指令を、経路設計を行いたい場合は軌道制御を使用すればよいことがわかった。

### 6.5 シミュレーションとの比較

最も結果が良かった角速度指令についてシミュレーションと比較する。5.4.1 項の方法で、目標値とサンプリング時間を実験と同様にした場合についてシミュレーションを行った。この結果を図 6.4 に重ね、それを図 6.11 に示す。3 つの入力と 3 つの出力は非線形な関係であることから、目標値が違うため図 5.17 のシミュレーション結果とは軌跡や収束時間は異なっている。

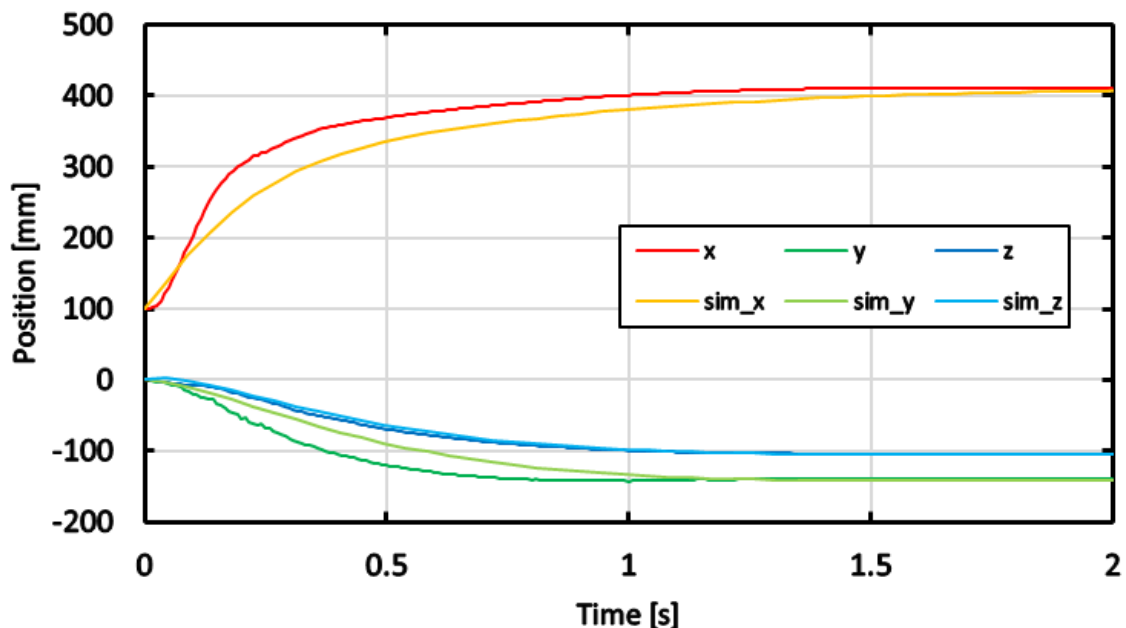


Fig6.11 Comparison of experiment and simulation

シミュレーションよりも実験のほうが収束は早い結果となった。この結果について考察する。



カメラによる画像処理，モータへの角速度の入力，モータの動作という手順とその時間の関係を図 6.12 に示す．

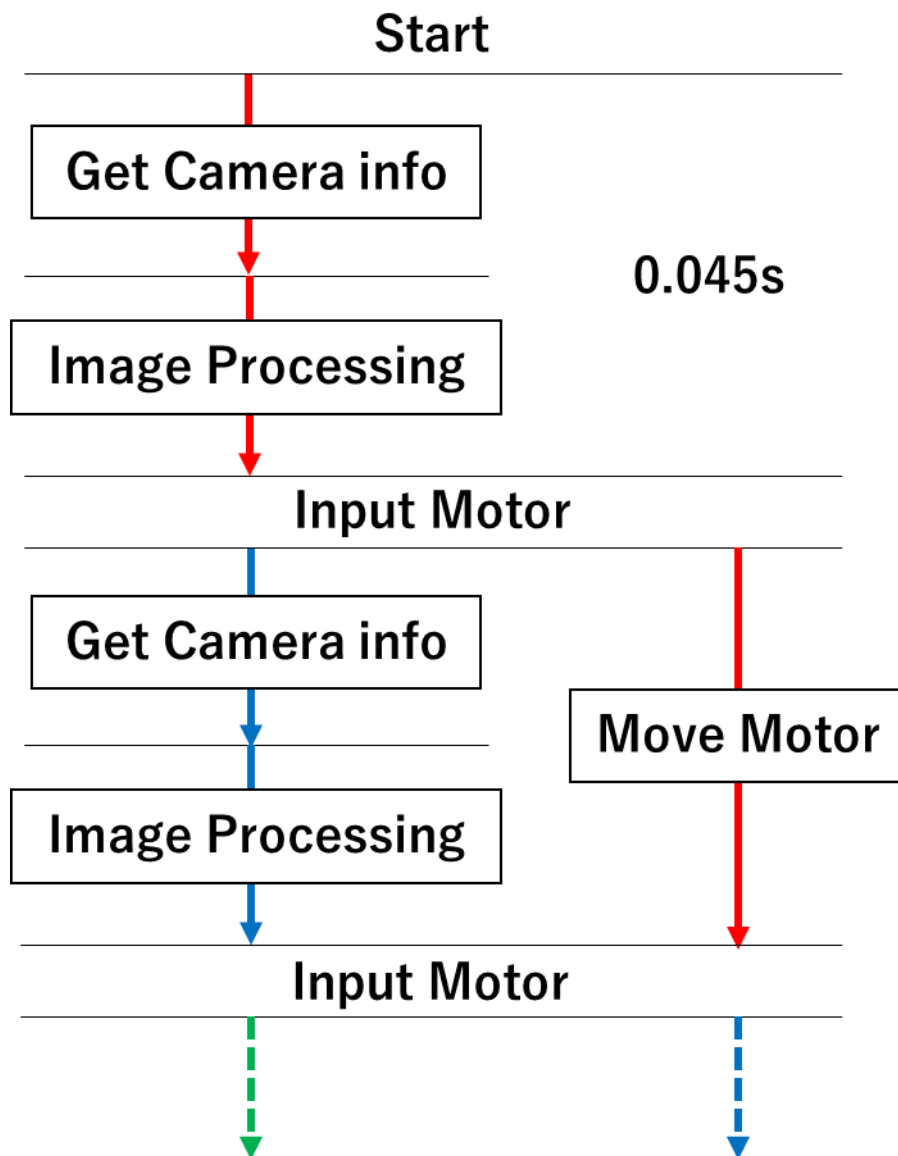


Fig6.12 Processing and Time Relationships

図 6.12 では 1 回目の処理を赤線，2 回目を青線，3 回目を緑線で示している．モータは入力をした後のカメラ情報の取得と画像処理にかかる時間の計 0.045s の間動作する．そのため，2 回目以降の入力値に使用するカメラの情報は 1 つ前の情報を使用することになる．要は 2 回目と 1 回目の入力値は等しくなり，3 回目以降は 1 つ前の偏差からの入力値となってしまう．これは，モータの動作時間を固定して処理時間はモータを停止させることで解決ができる．しかし，毎回停止することによって振動が大きくなることや 1 周期の時間が長くなってしまいう課題が生じるため，実験機をこのままにし，シミュレーションを変更してこの課題に対応させる．

2回目以降は1つ前の偏差を使用するシミュレーションを行った結果を図6.4に重ね、それを図6.13に示す。

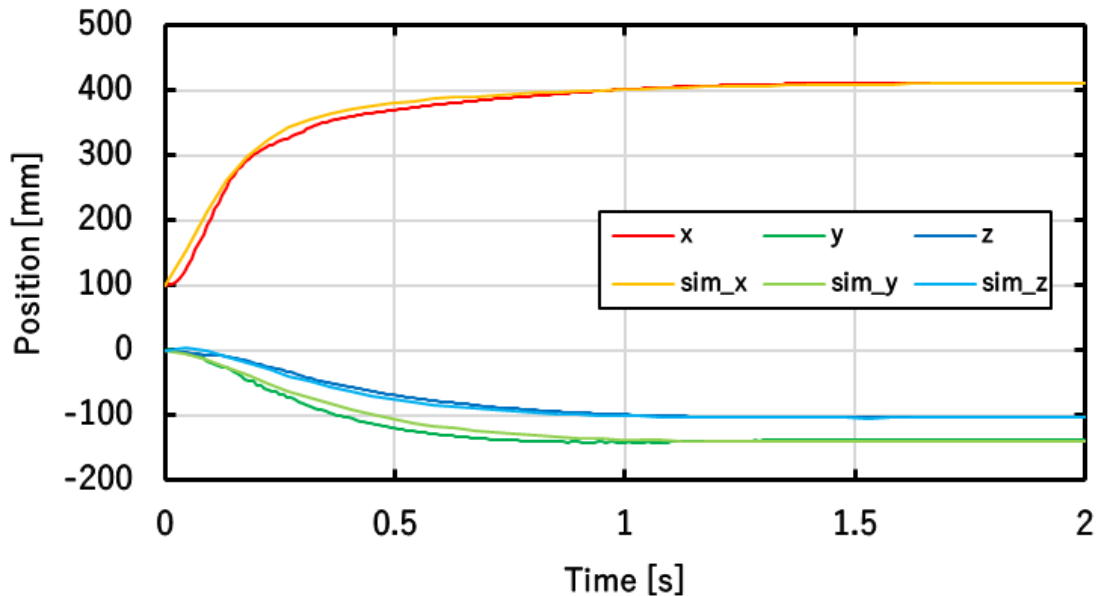


Fig6.13 Comparison of experiment and simulation after the change method

シミュレーションを実験の環境に対応させることで、実験とシミュレーションでほとんど同様の結果が得られた。この結果から、シミュレーションベースでのゲインの検証も可能となった。

## 6.6 角速度指令ゲインの再検証

角速度指令について、深度情報によってゲインを線形に変更していく方法を2通り検討し、実験を行った。ゲイン $K$ の計算式は以下となる。

$$K_i = aR_i + b \quad (6.6.1)$$

$$a = \frac{K_0 - K_n}{R_0 - 200} \quad (6.6.2)$$

$$b = K_n - 200a \quad (6.6.3)$$

ただし、 $R_i$ は任意の位置からの手先からピーマンまでの深度情報であり、 $R_0$ は初期位置での深度情報である。また、 $K_0$ は動作直後のゲイン、 $K_n$ は目標値でのゲインであり、 $K_0$ と $K_n$ を指定してゲインを決定する。以下では、深度情報をもとにゲインを小さくしていく方法をゲイン減少法、大きくしていく方法をゲイン増加法とする。

### 6.6.1 ゲイン減少法

この方法は、深度情報が大きい初期位置のときに、定数ゲインでは設定できなかった大きさのゲインをとることが狙いである。式(6.6.2)、式(6.6.3)において $K_0 > K_n$ である。実験結果について図6.14にモーションキャプチャを用いて取得した時間と手先位置の関係、図6.15に手先の軌跡、図6.16に手先の速度を示す。

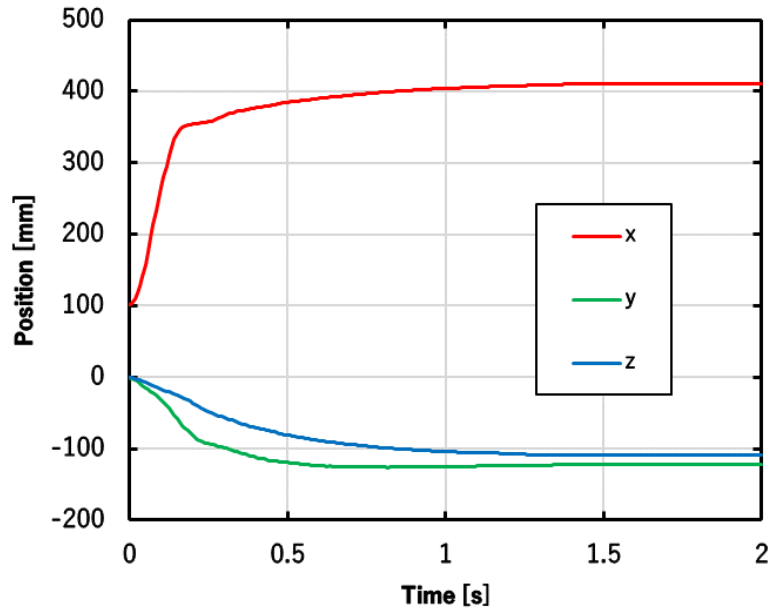


Fig6.14 Experimental results Position vs Time of Gain decrease method

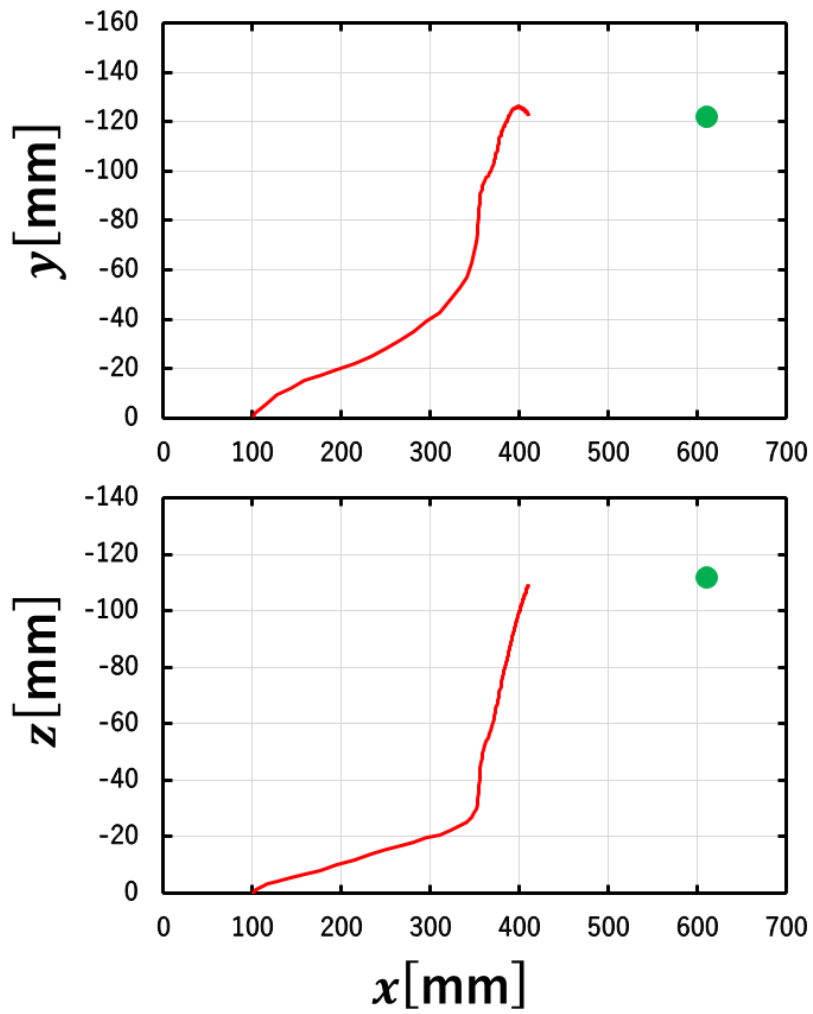


Fig6.15 Experimental results Trajectory of Gain decrease method

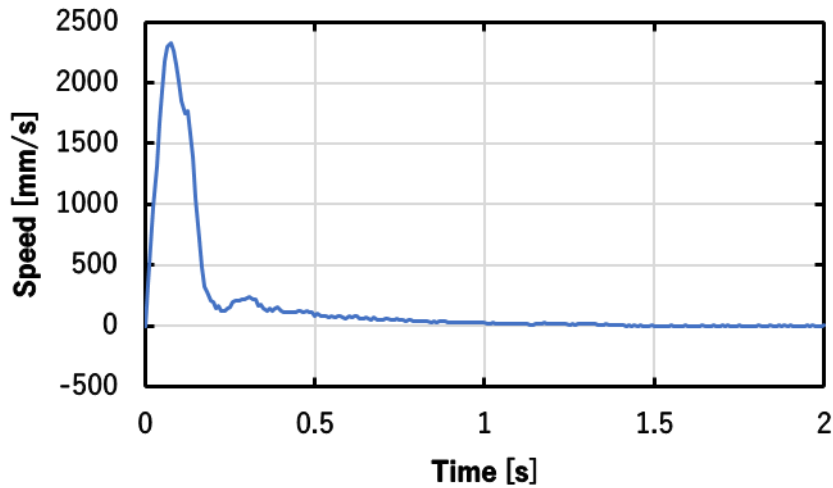


Fig6.16 Experimental results Speed vs Time of Gain decrease method

手先の最大速度は 2332.23mm/s となっており, 収束時間は 1.36s, 最終的な誤差 $[e_x \ e_y \ e_z]$ は $[-0.400 \ 0.729 \ 0.991]$ mm である. 少しだけ環境は異なるが 6.2 節で行った実験と比較すると, 最大速度は約 682mm/s 大きくなっており, 収束時間はほとんど変化がなかった. この方法では, 定数ゲインでは設定できなかった大きさのゲインを使用できるため速度を向上させられるが, 収束時間については大きな影響を及ぼせないことがわかった.

### 6.6.2 ゲイン増加法

この方法は, 深度情報が小さい目標に近づいた状態のときに, ゲインを大きくとり, 収束を早くすることが狙いである. 式(6.6.2), 式(6.6.3)において $K_0 < K_n$ である. 実験結果について図 6.17 にモーションキャプチャを用いて取得した時間と手先位置の関係, 図 6.18 に手先の軌跡, 図 6.19 に手先の速度を示す.

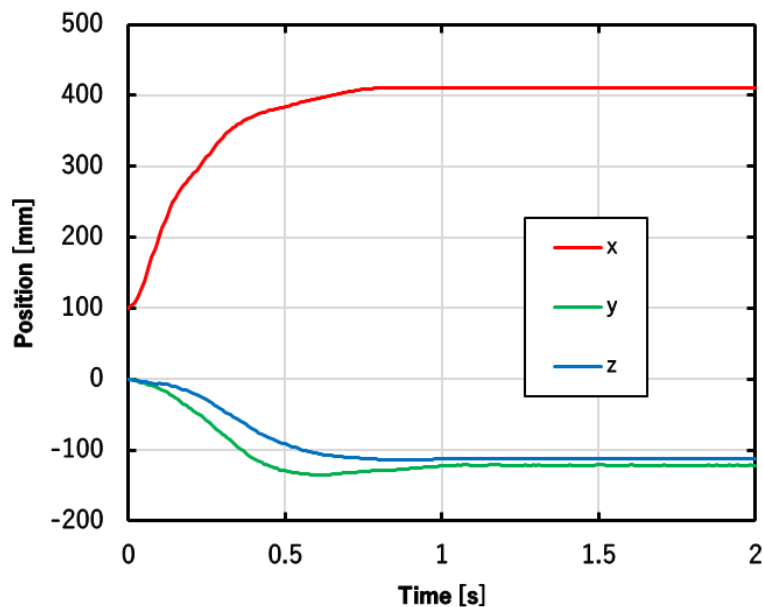


Fig6.17 Experimental results Position vs Time of Gain increase method

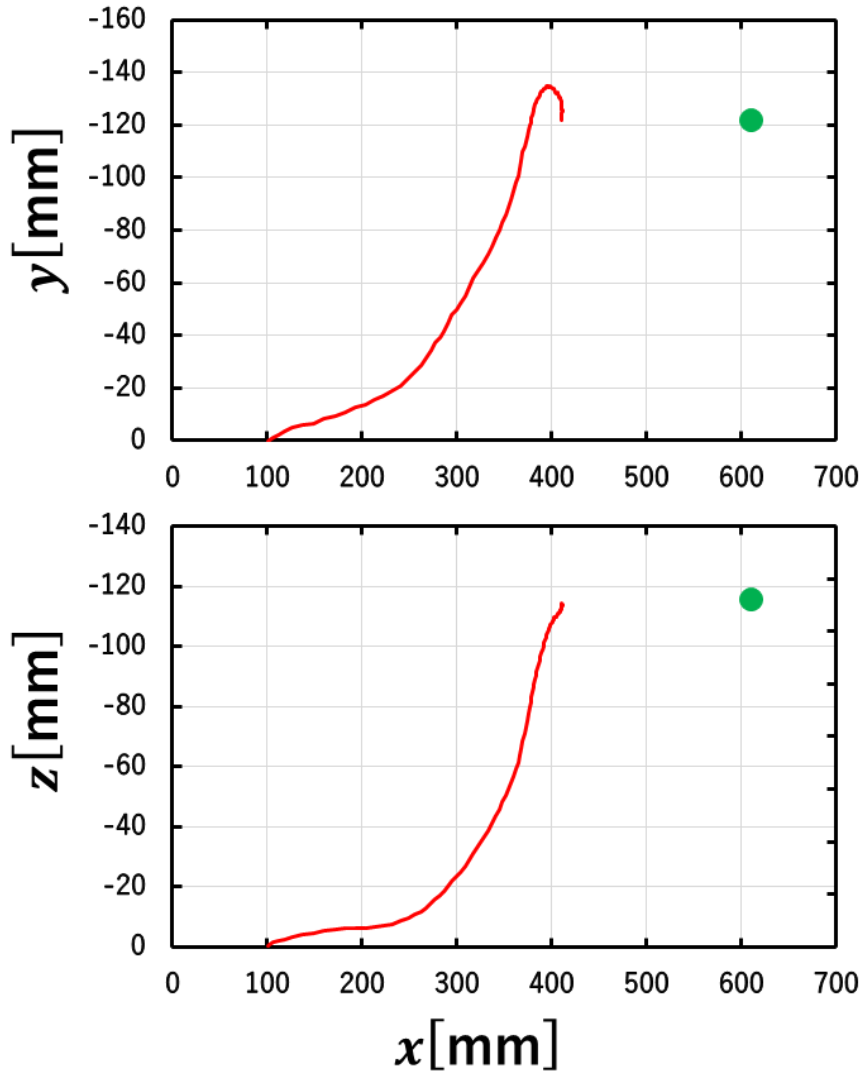


Fig6.18 Experimental results Trajectory of Gain increase method

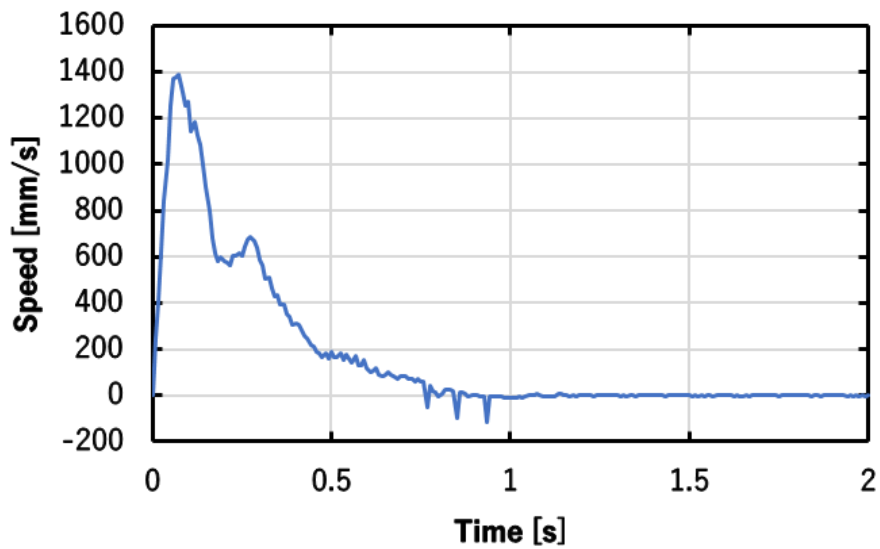


Fig6.19 Experimental results Speed vs Time of Gain increase method

手先の最大速度は 1391.58mm/s となっており, 収束時間は 1.025s, 最終的な誤差 $[e_x \ e_y \ e_z]$ は $[0.630 \ 0.863 \ 0.442]$ mm である. 6.2 節で行った実験と比較すると, 最大速度は約 257mm/s 小さくなっており, 収束時間は約 0.4 秒短く, Y 軸方向のオーバーシュートが大きくなっている. 図 6.19 の速度グラフを前項の結果である図 6.16 と比較すると, 手先が速度を持つ時間が長いことがわかる. 目標値付近でもある程度の速度を持つことができるため, 収束時間が短くなったと考える. この方法では, 目標に近づくにつれて, ゲインが大きくなるため, 収束時間を早められるが, オーバーシュートが大きくなってしまったことがわかった.

今回は, ゲインを単純に線形に変化させる 2 つの方法について実験を行ったが, 他の方法についても検討し, 常に最善の性能を実現させていきたい.

### 6.7 制御の組み合わせ

最大速度は局所位置制御, 収束時間と精度は角速度指令が優れていることがわかったため, 局所位置制御と角速度指令を組み合わせる実験を行う. フローチャートを図 6.20 に示す.

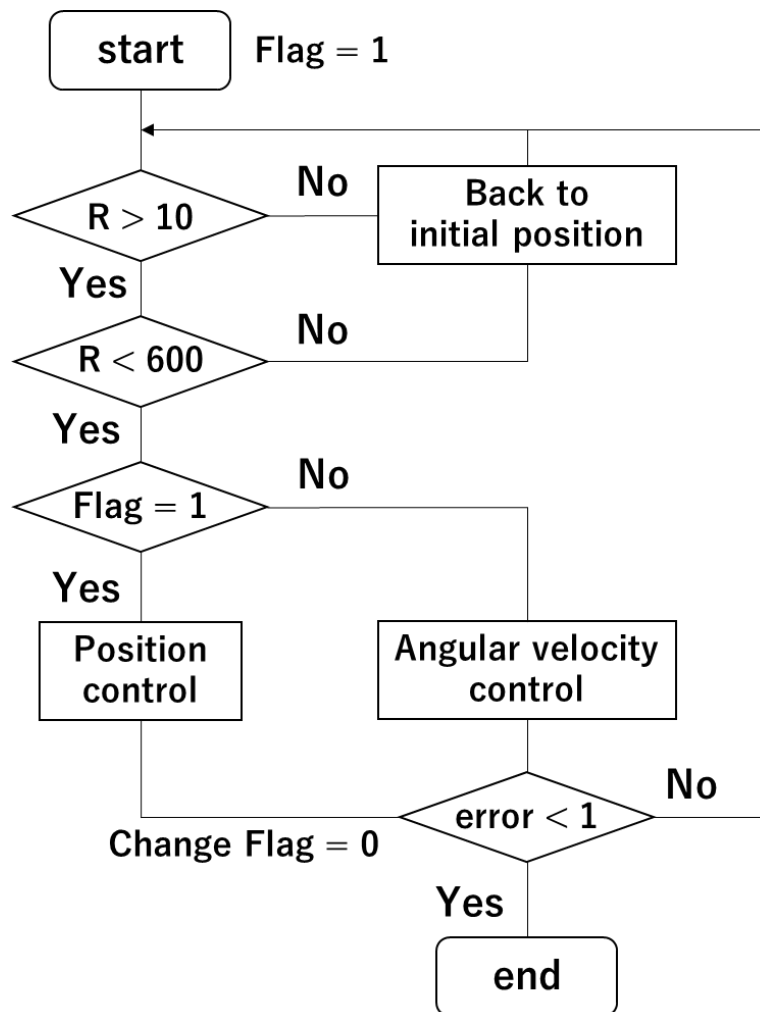


Fig6.20 Flowchart

図 6.20 に示すように、実験機が動作可能な範囲である深度情報が 10 から 600mm のときに、Flag という変数を用いて制御方法を 2 つ切り替える方法である。Flag が 1 の場合に局所位置制御を行い、Flag が 0 の場合に角速度指令を行う。動作開始後は Flag =1 の状態で始まるため、最初は局所位置制御によって高速でピーマンに近づき、その後の修正を角速度指令によって素早く、精度よく行うことが狙いである。実験結果について図 6.21 にモーションキャプチャを用いて取得した時間と手先位置の関係、図 6.22 に手先の軌跡、図 6.23 に手先の速度を示す。

手先の最大速度は 2361.95mm/s となっており、収束時間は 1.38s、最終的な誤差 $[e_x \ e_y \ e_z]$ は $[0.848 \ 0.732 \ 0.382]$ mm である。6.1 節で行った局所位置制御のみの場合と比べると、最大速度は同じような値となっており、収束時間と誤差ともに小さくできている。また、6.2 節で行った速度指令のみの場合と比べると、最大速度が大きくなっており、これによって収束時間も少し短くできている。2 つの制御方法を組み合わせることによってそれぞれの利点を活かし、欠点を改善することができた。ただし、局所位置制御の欠点である Y 軸方向の大きなオーバーシュートは改善ができていないため、対応策を考案する必要がある。

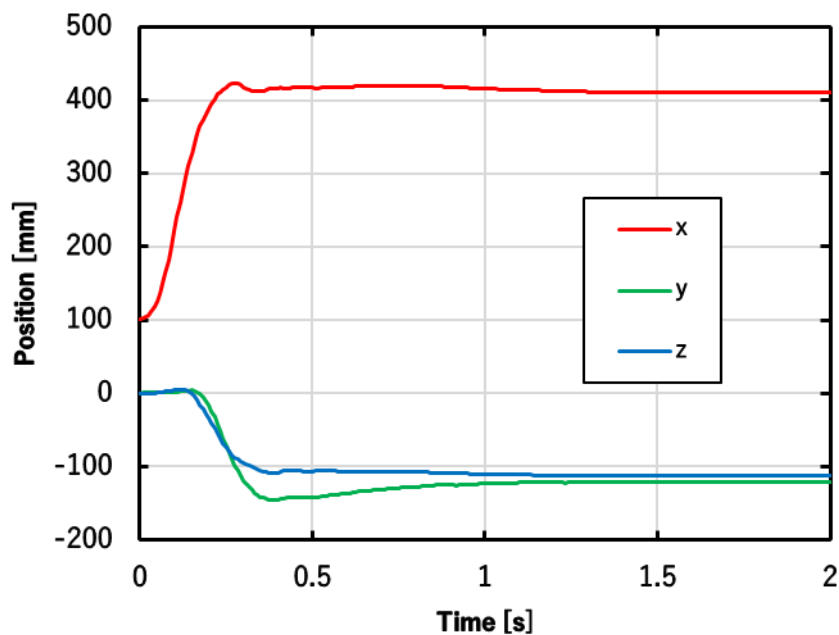


Fig6.21 Experimental results Position vs Time of Combination for Position and Angular velocity control

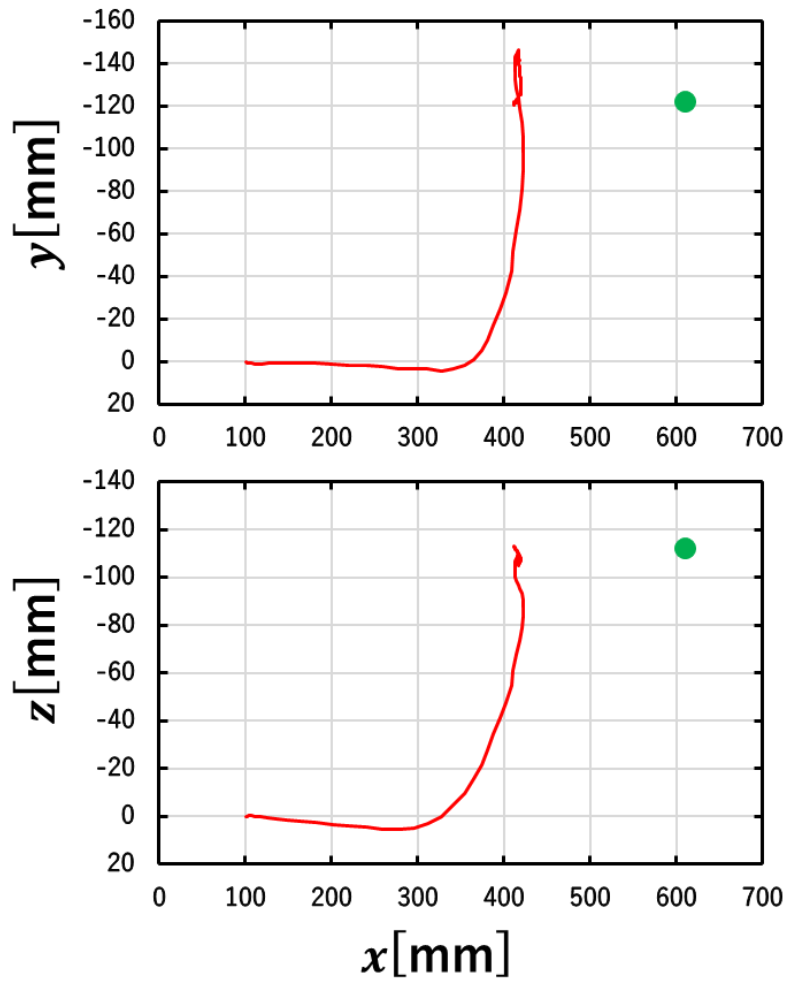


Fig6.22 Experimental results Trajectory of Combination for Position and Angular velocity control

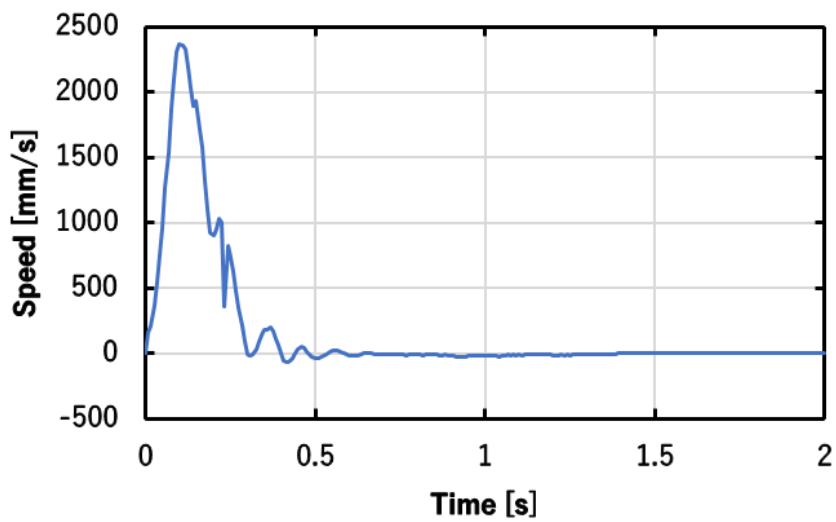


Fig6.23 Experimental results Speed vs Time of Combination for Position and Angular velocity control



## 第7章 結論

### 7.1 研究のまとめ

本稿では高速で高精度という特徴をもつパラレルメカニズムを使用した収穫用マニピュレータを提案し、収穫動作に対応させるため水平方向への駆動に変更した。水平駆動の課題を解決するためにリンク比の変更を提案し、ベース側と出力側のリンクを等しくすることでリンクに加わる曲げ応力を軽減できることがわかった。カーボン棒を使用した実験機を製作し、可動範囲は問題ないことを確認した。実験機の手先にカメラを取り付けて、シミュレーションから得られたゲインを参考に3つの制御方法について動作実験を行い、比較することで各制御方法の利点と欠点がわかった。

### 7.2 今後の課題

今後は、実験で最も良い結果となった角速度指令についてさらに最適なゲインを検証していく。今回はマニピュレータと簡単な認識機構のみの内容となったため、今後はハンド機構、移動機構などとの組み合わせを検証し、実際の農地での実験を行っていく。

## 謝辞

本研究を行うにあたって、懇切丁寧にご指導いただきました岡宏一教授に感謝申し上げます。  
相談に乗ってくださった研究室のみなさまに感謝申し上げます。

本研究は高知県の「IoP が導く Next 次世代型施設園芸農業への進化プロジェクト」の助成を受けております。この場をお借りして感謝申し上げます。

## 参考文献

- [1] 農林水産省ホームページ：農業労働力に関する統計  
<https://www.maff.go.jp/j/tokei/sihyo/data/08.html>  
(参照日 2024年2月8日)
- [2] 農林水産省ホームページ：スマート農業  
<https://www.maff.go.jp/j/tokei/sihyo/data/08.html>  
(参照日 2024年2月8日)
- [3] AGRIST 株式会社ホームページ  
<https://agrist.com/products/robot>  
(参照日 2024年2月8日)
- [4] ヤンマーホールディングス株式会社ホームページ  
<https://www.yanmar.com/jp/news/2022/03/01/104681.html>  
(参照日 2024年2月8日)
- [5] 立矢宏，“パラレルメカニズム”，森北出版株式会社，2019.
- [6] 増田峰知，“直動型パラレルメカニズムの解析と応用”，大阪大学大学院学位論文(2003)
- [7] Delta robot kinematics-Tutorials  
<https://hypertriangle.com/~alex/delta-robot-tutorial/>  
(参照日 2024年2月8日)
- [8] Depth Camera D435i - Intel RealSense  
<https://www.intelrealsense.com/depth-camera-d435i/>  
(参照日 2024年2月8日)