

— 卒業論文報告書 —

作業空間の非干渉化に基づいた ビジュアルサーボ系の新しい設計法

平成14年2月13日提出

指導教官： 堀 洋一 教授

東京大学工学部電気工学科

00530 高橋 啓

内容梗概

ビジュアルサーボ系は、目標物体と、手先にカメラが取り付けられたロボットで構成される。目標物体はロボットの作業空間内を動き回り、ロボットは手先に取り付けられたカメラからの画像情報を用いてカメラ位置が目標物体位置と一致するように目標物体を追従する。

本研究では、制御対象であるロボットを作業空間座標系で非干渉化することにより可能となる、ビジュアルサーボ系の新しい設計法を提案する。

作業空間の非干渉化制御により、システム全体は線形化され、極配置法に基づいた状態フィードバックが可能となる。提案する設計手法においては、カメラ位置の過渡応答が任意に設定できる。また、設計の見通しが極めて良くなることが特長である。

また、シミュレーションと2軸DDロボットを用いた実験によって、提案する設計手法の有効性を示す。

目次

第1章	序論	1
1.1	研究の背景	1
1.2	研究の目的	2
第2章	ビジュアルサーボシステムの基礎的事項	3
2.1	ビジュアルサーボ系の構成	3
2.1.1	システム全体の構成	3
2.1.2	カメラのモデルと特徴量	3
2.2	ビジュアルサーボ系に関する研究の分類	4
2.2.1	目標物体の運動モデルに関する分類	4
2.2.2	ビジュアルサーボ系におけるむだ時間と予測に関する分類	5
2.2.3	画像ヤコビアンを用いた従来手法	6
2.3	本研究の立場	6
第3章	作業空間の非干渉化によるビジュアルサーボ系の線形化	8
3.1	作業空間の非干渉化	8
3.1.1	多自由度ロボットの運動学	8
3.1.2	作業空間での動力学方程式と外乱モデル	8
3.1.3	外乱オブザーバに基づいた作業空間の非干渉化	9
3.1.4	ロバストな位置制御系の実現	11
3.2	カメラの逆モデルによるビジュアルサーボ系の線形化	11
第4章	制御器の設計	13
4.1	極配置に基づいた状態フィードバック	13
4.1.1	状態フィードバックの設計	13
4.1.2	位置誤差収束の保証	14
4.2	極の位置による応答の変化	15
第5章	実験による比較・検討	18
5.1	実験装置の概要	18
5.1.1	ソフトウェア構成	18
5.1.2	ハードウェア構成	19
5.2	実験結果及び考察	20
5.2.1	作業空間位置制御系の実現と、作業空間位置指令に対する応答	21
5.2.2	目標物体を静止させた場合	22
5.2.3	目標物体を円運動させた場合	23
第6章	結論	25
	謝辞	26
	参考文献	27
	発表文献	28

第1章 序論

1.1 研究の背景

現在、様々な分野においてロボットが使用されているが、近年益々自律動作、あるいは未知の環境下での動作が要求されている。従来のロボットは、エンコーダ等内界センサからの信号のみに基づいて制御されているため、ロボットの機構モデルや外界環境が変化する、あるいは未知であるとそれに対応できず、タスクの実行に障害が生じることがあった。したがってロボットの自律的な制御のためには、タスクの実行状況をフィードバックするための外界センサが必要となる。つまり産業ロボット等、整備された環境で与えられた作業を繰り返す場合には、内界センサのみで十分であった。しかし、災害地区や宇宙空間などで作業を行う極限ロボットや、病院や家庭で活躍するロボットにおいては、全くの未知な環境で、自律的に作業を行う必要があるため、外界の情報を検出し、制御に用いるための機構が必要となる。そこで実際に対象物との接触を伴うような作業では、外界センサとして力センサが用いられる。しかし対象物位置の正確な把握が不可能な場合には、非接触のセンサが必要となる。また、非接触センサの中でも視覚センサは最も身近、かつ汎用的であるから、外界センサとして利用価値はかなり高い。しかし視覚情報を用いてロボットを操作する際に、視覚制御の実現を最も困難にしていたのは、画像解析に要するむだ時間が長いことであった。そのため初期の研究では、ロボットは静止した状態で撮像し、修正目標に向かって動作するという Look-and-Move(“見てから動く”)を繰り返す方法が用いられていた [1]。これは 画像処理と 3次元情報の計算、それに基づく軌道計画には膨大な計算が必要なため、オフライン処理が必要であるとの発想に基づいており、静的フィードバックと位置付けられる。その結果、視覚が観測される度にロボットの動きが中断され、動的応答が遅くなる。結果として、動く目標を追跡するなどのリアルタイムタスクは困難となる。

近年、主として計算機性能の向上に伴って、アームを動作させながら連続的に視覚情報を獲得し、フィードバックするビジュアルサーボが研究されるようになった。近年の視覚フィードバックの研究では、動く物体の実時間トラッキングを目標とした視覚サーボ法が盛んである。これは作業環境を理解してからおもむろに動き出す“見てから動く”方法とは異なり、作業対象物とアームの間の状態を観測しながら反射的に反応する“見ながら動く”方法であり、センサ情報の処理、軌道計画、モータ制御がすべて並列して動くという特徴を持つ [2]。

1.2 研究の目的

ビジュアルサーボシステムでは、画像ヤコビアンと呼ばれる、関節変位の変分から特徴量の変分への関係を用いて位置指令が生成される手法が一般的である。しかしながら、画像ヤコビアンが非線形であるために系全体が非線形となり、系の解析的な設計が困難になってしまう欠点があった。

そこで本研究ではこの問題点を解決すべく、マニピュレータに対する入力に関節変位を用いることなく、作業空間での目標軌道を採用することにより、系全体のシンプルかつ解析的な記述が可能となることを示す。また、プラントであるロボットを作業空間で非干渉化することによって、極配置法に基づいた状態フィードバックなどの解析的な設計手法が適用でき、それによりカメラ位置の過渡応答が調整可能で、設計の見通しがよくなることを実験によって示す。

第2章 ビジュアルサーボシステムの基礎的事項

2.1 ビジュアルサーボ系の構成

2.1.1 システム全体の構成

本研究では、ビジュアルサーボ系として図 2.1 に示すような、先端にカメラが上向きに取り付けられた 2 軸 DD ロボットを用いた系を考える。

本研究で対象とするビジュアルサーボ系は二自由度であるため、目標物体とカメラの z 方向の距離は不変かつ既知とする。制御目的はカメラ位置を目標物体位置に一致させ、リアルタイムで追従することである。ここで、カメラの位置を s_{cam} 、物体の位置を s_{obj} とする。

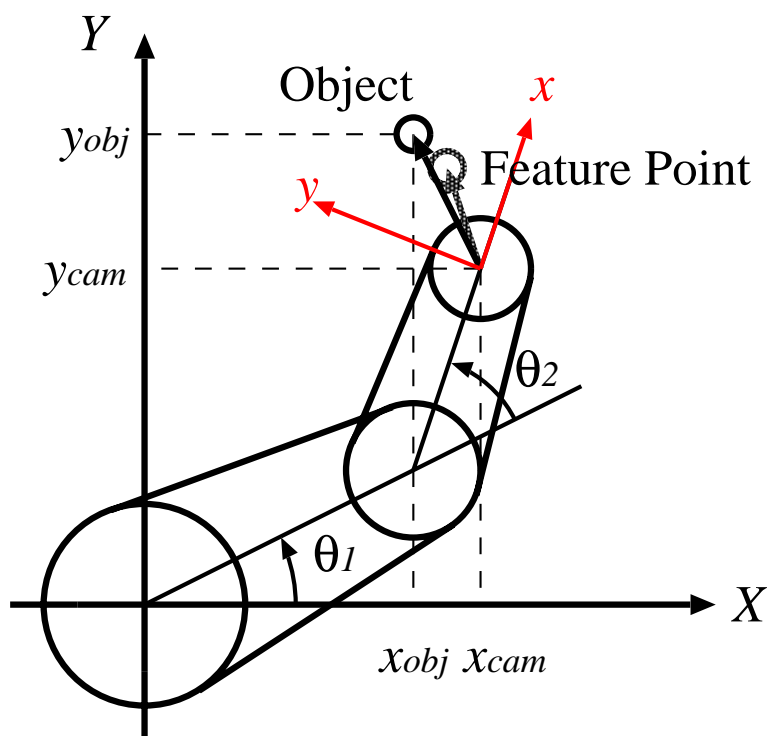


図 2.1: 2 軸 DD ロボットを用いたビジュアルサーボシステム

2.1.2 カメラのモデルと特徴量

特徴量は、カメラ中心を原点にした物体の位置 $(X, Y)^T$ を画像平面に写像したものであり、

$$\begin{bmatrix} \xi_x \\ \xi_y \end{bmatrix} = \frac{f}{z} \begin{bmatrix} X \\ Y \end{bmatrix} \stackrel{\text{def}}{=} \mathcal{L}^+(s_{cam}, s_{obj}) \quad (2.1)$$

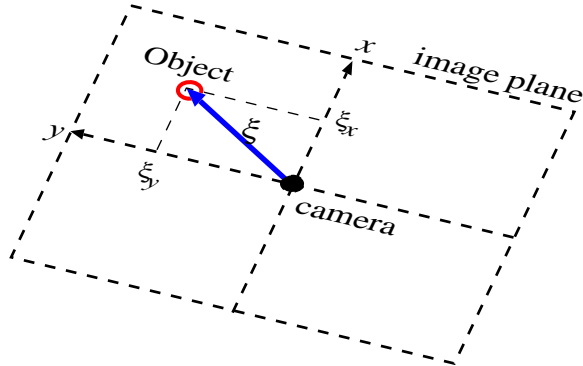


図 2.2: 画像空間上の特徴量ベクトル

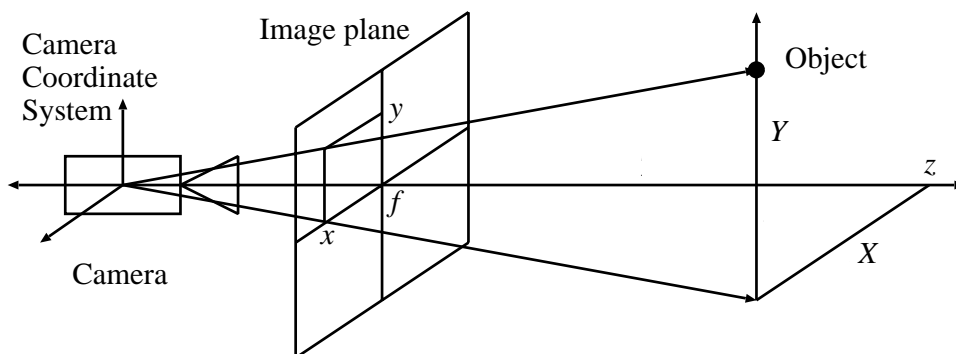


図 2.3: 目標物体の撮像モデル

と表される。 f はカメラの焦点距離、 z はカメラと目標物体間の z 軸方向の距離である。

ι^+ は s_{cam}, s_{obj} からの写像であり、この写像によって表される $(x, y)^T$ を特徴点とする。本研究では特徴量は単一の特徴点で表されるとし、このベクトルを特徴量ベクトル ξ とする。画像空間で見た特徴量ベクトル ξ を図 2.2 に示す。

目標物体位置 s_{obj} カメラ位置 s_{cam} から特徴量ベクトル ξ への写像 ι^+ は一般的にはある関数で記述されるが、本研究で対象とするシステムは一つのカメラで構成されるので、本研究のシステムにおける特徴量は、実座標に対するカメラ座標の回転角だけ $s_{obj} - s_{cam}$ を回転・拡大(縮小)した形になり、 $s_{obj} - s_{cam}$ に対して (2.3) 式で表されるような行列 $\iota(\theta)$ をかけたもので表すことができる。これを、本研究におけるカメラのモデルと呼ぶ。

$$\xi = \begin{bmatrix} \xi_x \\ \xi_y \end{bmatrix} = \iota(\theta) \begin{bmatrix} x_{obj} - x_{cam} \\ y_{obj} - y_{cam} \end{bmatrix} \quad (2.2)$$

$$\iota(\theta) = \frac{f}{z} \begin{bmatrix} \cos(\theta_1 + \theta_2) & \sin(\theta_1 + \theta_2) \\ -\sin(\theta_1 + \theta_2) & \cos(\theta_1 + \theta_2) \end{bmatrix} \quad (2.3)$$

2.2 ビジュアルサーボ系に関する研究の分類

2.2.1 目標物体の運動モデルに関する分類

ビジュアルサーボ系に関する研究を目標物体の運動の取扱いに関して分類すると、以下のようなになる。

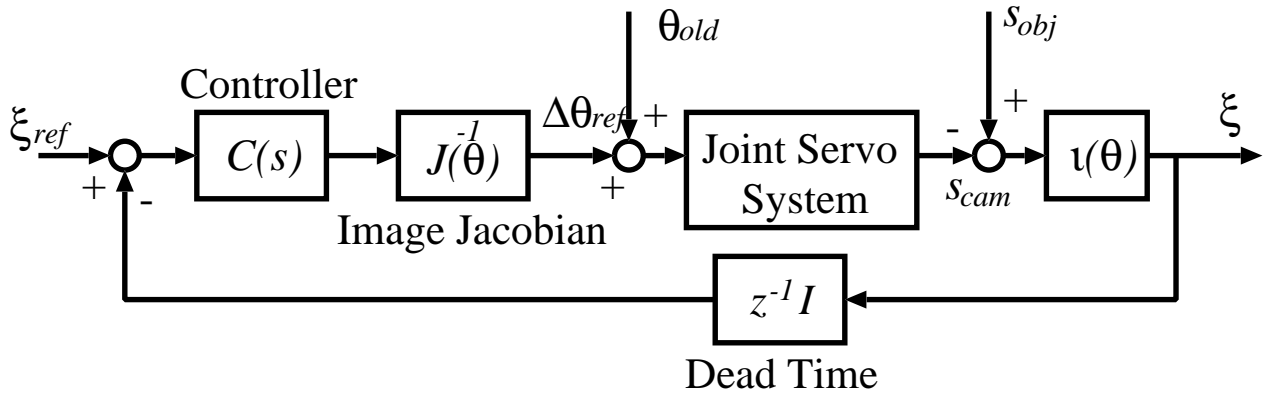


図 2.4: 画像ヤコビアンを用いた制御手法

1. 目標物体が静止している場合
2. 目標物体が運動している場合
 - (a) 目標物体の運動が既知である場合
 - (b) 目標物体の運動が未知である場合

目標物体が静止している場合は、目標物体の運動を考慮する必要はなく、画像処理のむだ時間や制御入力の調整などが問題となる。むだ時間に対して直接的にアプローチした研究としては、岡崎ら [3] のものなどが挙げられる。また、橋本、木村らはむだ時間そのものにはアプローチせず、系の安定性を向上させる手法 [4][5] を提案している。

目標物体が運動し、その軌道が既知である場合の研究では、予め与えられた目標軌道とリアルタイムの画像情報を用いて、より正確に制御を行うことが目的となる。遠藤、田中らは、目標軌道全体の情報を利用した最適予測制御を用いることで、滑らかかつ正確な軌道追従を実現した [6]。また、全目標軌道が既知であれば、むだ時間に対する対処をとらなくとも望ましい制御が可能となる。

また、一般に目標物体の運動が未知である場合は、物体の運動に関する情報はリアルタイムでしか得られないため、物体の運動が既知である場合と同等な精度で制御を行うことは難しい。任意の軌道を前提とした研究としては、非線形オブザーバを用いた手法 [7] などがある。

2.2.2 ビジュアルサーボ系におけるむだ時間と予測に関する分類

視覚フィードバック制御系では、特に目標物体が運動する場合において、視覚センサ及び画像処理による大きなむだ時間が大きく制御性能を劣化させ、これが問題となる場合が多い。むだ時間に対して直接的にアプローチする方法としては、物体の運動予測を行う方法 [7][8]、むだ時間による位相遅れを直接的に補償する方法 [3] がある。

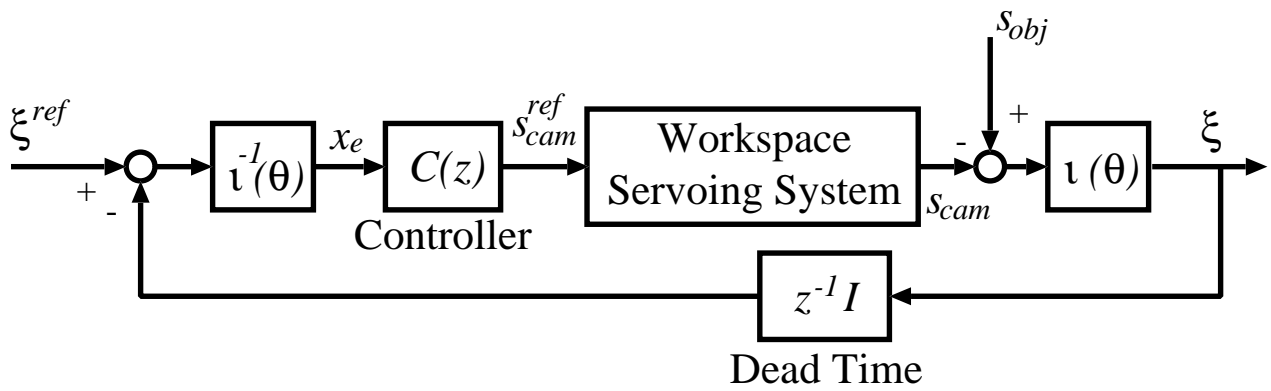


図 2.5: 作業空間に基づく制御手法

2.2.3 画像ヤコビアンを用いた従来手法

これまでに提案されている視覚フィードバック制御法のほとんどは、画像ヤコビアンの逆行列に基づくものである。画像ヤコビアンは (2.5) 式のように表され、画像特徴量の差分とマニピュレータの関節角の差分との関係である。図 2.4 のように、画像ヤコビアンの逆行列を用いて、関節角変位の目標値を生成することができる。ただし、図中の ξ^{ref} は目標特徴量であり、特徴点が画像の中心にくることが制御目的であることから、

$$\xi^{ref} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (2.4)$$

と表される。

$$\begin{aligned} J(\theta, s_{cam}) &= \frac{\partial \xi}{\partial s_{cam}} \frac{\partial s_{cam}}{\partial \theta} \\ &= \begin{bmatrix} \xi_y - \frac{f}{z} L_1 \sin(\theta_2) & \xi_y \\ -\xi_x - \frac{f}{z} L_1 \cos(\theta_2) - \frac{f}{z} L_2 & -\xi_x - \frac{f}{z} L_2 \end{bmatrix} \end{aligned} \quad (2.5)$$

本研究で対象とする系 (図 2.1) では画像ヤコビアンは (2.5) 式となる。画像特徴量 ξ とマニピュレータの関節角 θ は観測可能なため画像ヤコビアンの実現は簡単であるが、画像ヤコビアンは非線形であるため、コントローラ的设计を解析的に行うのは困難である。また画像ヤコビアンは、画像空間上の特徴量から関節空間上の関節角指令を直接的に生成するため、一般的にタスクが記述される作業空間上で記述することはできない。

2.3 本研究の立場

画像ヤコビアンを用いる手法には上記のような問題点があるため、本研究では画像ヤコビアンを用いず、作業空間上の変数のみを用いて記述する制御手法を適用する。作業空間上でタスクを記述するためには、制御対象であるマニピュレータを作業空間で非干渉化することが必要である。その上で、図 2.5 のようにカメラモデルの逆行列である $i^{-1}(\theta)$ を挿入することで、線形化が達成でき、併せてコントローラ $C(z)$ の解析的な設計が可能となる。

さらに、ビジュアルサーボ系は図 2.5 をとらえ直し、図 2.6 のように考えることができる。図 2.6 は、ビジュアルサーボ系への入力目標物体位置であるとする立場であり、本研究ではビジュアルサーボ系をこのようにとらえる。

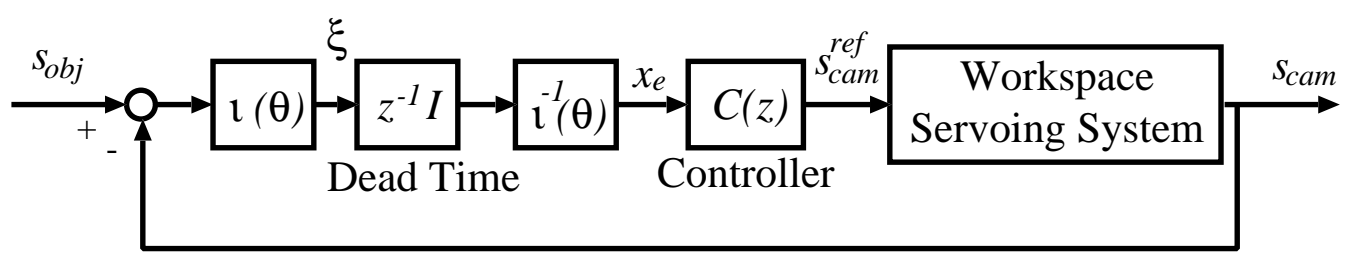


図 2.6: 目標物体位置を入力としてとらえた作業空間に基づく制御手法

本研究においては、目標物体の運動は未知であるものとする。また、本研究の目的は制御器の解析的な設計であり、線形予測などの簡単な運動予測では逆に系の安定性を損なうことがあるため、むだ時間に対する対処として 2.2.2 節で述べた目標物体の運動予測は、本研究では用いないものとする。

第3章 作業空間の非干渉化による ビジュアルサーボ系の線形化

3.1 作業空間の非干渉化

マニピュレータ作業空間の各軸に対して独立に位置指令を与えるためには、作業空間において非干渉化する必要がある。本研究では、作業空間での外乱オブザーバに基づいて作業空間の非干渉化を実現する [9]。

3.1.1 多自由度ロボットの運動学

多自由度ロボットのモーションコントローラを実現する際には、コントローラを構成する座標系として二種類考えられる。ひとつは、ロボットを駆動するアクチュエータの関節の変位を表す関節空間である。これはロボットに最終的な指令を与えるのに必要である。もうひとつは、ロボットの先端の運動を絶対座標で表す作業空間である。これは、ユーザにとって認識しやすい空間である。通常、ロボットの制御を行なう際には、ユーザにとって認識しやすい作業空間上で動作指令を与えることが望ましい。

そこで座標間の変換を行う必要があるが、座標変換の手法としてよく用いられるのは、逆運動学と呼ばれる、マニピュレータの手先位置からそれに対応する各関節変位を計算する問題を解き、作業空間の位置指令から関節空間の位置指令に変換する方法である。

このような座標変換はロボットの運動学的な問題であり、実際に高精度な動作を実現するためにはロボットの動力学をも考慮して制御系を構成する必要がある。またロボットの各関節には干渉力を含めた様々な外乱トルクが作用するため、これらの外乱トルクを積極的に抑圧する制御系が望ましい。このような観点から本研究では、逆運動学による変換を用いず、外乱オブザーバを用いた作業空間非干渉制御を実現する。

3.1.2 作業空間での動力学方程式と外乱モデル

関節空間におけるロボットの動力学方程式は (3.1) 式で表される。

$$J_n \ddot{\theta} = K_{tn} I_a^{ref} - T_{dis} = \tau_n - T_{dis} \quad (3.1)$$

θ は各関節の関節変位、 I_a^{ref} 、 τ_n は各関節への入力電流、入力トルクである。また J_n 、 K_{tn} は対角要素に各関節のノミナルなイナーシャおよびトルク係数をもつ対角行列であり、 T_{dis} には干渉トルクや各関節にとって外力である反作用トルク、摩擦トルクなどのほかに、プラントの変動も含まれている。

次に作業空間での動力学方程式を考える。作業空間の位置ベクトル x が、関節空間の位置ベクトル θ を用いて (3.2) 式のように表されるとすると、

$$x = H(\theta) \quad (3.2)$$

$$\dot{x} = \frac{\partial H(\theta)}{\partial \theta} \dot{\theta} = J_{aco} \dot{\theta} \quad (3.3)$$

$$\ddot{x} = J_{aco} \ddot{\theta} + \dot{J}_{aco} \dot{\theta} \quad (3.4)$$

となる。ただし、 J_{aco} はヤコビ行列を表す。また作業空間における力ベクトル F_n と関節空間でのトルクベクトル τ_n との関係は次式となる。

$$\tau_n = J_{aco}^T F_n \quad (3.5)$$

ここで作業空間の等価質量行列を定義する。外乱トルクの影響を無視すると、関節空間の運動エネルギーはノミナルな慣性行列 J_n を用いて (3.6) 式のように表される。

$$K_{joint} = \frac{1}{2} \dot{\theta}^T J_n \dot{\theta} \quad (3.6)$$

また、作業空間上における質量行列を M とすると、同空間における運動エネルギーも同様に (3.7) 式のようになる。

$$K_{work} = \frac{1}{2} \dot{x}^T M \dot{x} \quad (3.7)$$

関節空間と作業空間における運動エネルギーは等しいと考えることができ、

$$M = J_{aco}^{-T} J_n J_{aco}^{-1} \quad (3.8)$$

となる。(3.4), (3.5), (3.8) 式を関節空間の動力学方程式 (3.1) に代入すると、次式のような作業空間の動力学方程式 (3.9) が得られる。

$$M_n \ddot{x} = F_n - F_{dis} \quad (3.9)$$

ただし、 M_n はロボットの初期姿勢により設計されるノミナル質量行列、 F_{dis} は質量行列の変動によるトルク変動も含めた外乱トルクである。

(3.1), (3.9) 式を比較すると、ロボットの動力学方程式は、関節空間と作業空間で同じ形をとっている。ただし、定義される外乱の内容は異なる。

3.1.3 外乱オブザーバに基づいた作業空間の非干渉化

関節空間での外乱オブザーバに基づいた作業空間の非干渉化

関節空間で外乱オブザーバを構成することにより、慣性行列 J は任意の行列 J_n に置き換えることができる。これにより作業空間における等価質量行列も次式のように再定義される。

$$M = J_{aco}^{-T} J_n J_{aco}^{-1} \quad (3.10)$$

$$J_v = J_{aco}^T M_{vn} J_{aco} \quad (3.11)$$

(3.11) 式のように J_v を決めることで、 M は M_{vn} に固定化される。これを用いて作業空間での動力学方程式を書き直すと、

$$M_{vn} \ddot{x} = F_n + M_{vn} \dot{J}_{aco}(\theta) \quad (3.12)$$

$$- M_{vn} J_{aco}(\theta) J_n^{-1} G_s(s) T_{dis} \quad (3.13)$$

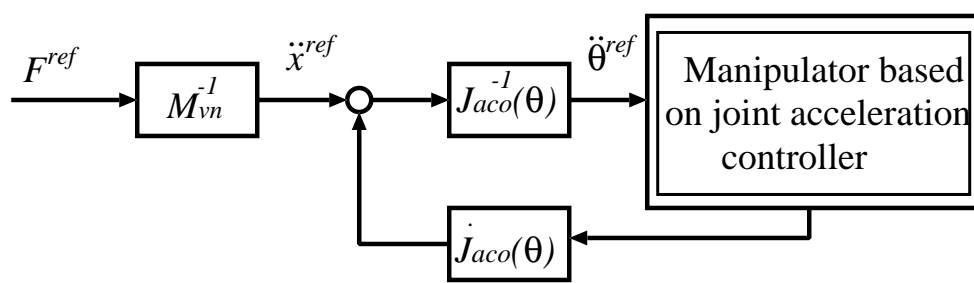


図 3.1: 関節空間オブザーバに基づいた作業空間での力コントローラ

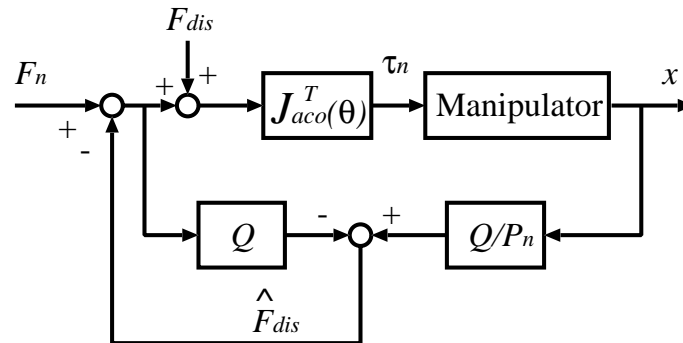


図 3.2: 作業空間オブザーバ

となる。ここで $G_s(s)$ は感度関数であり、オブザーバによって十分小さくなっているため、(3.13) 式の第三項は無視できる。このように、関節空間オブザーバに基づいた作業空間における力コントローラは図 3.1 のように表される。

図 3.1 では、作業空間の力ベクトルを等価質量行列により加速度ベクトルに変換し、更にヤコビの逆行列介して各関節への角加速度指令を生成している。重要なことは、等価質量行列 M_{vn} がマニピュレータの姿勢 (あるいは初期姿勢) に関係なく任意に設定できることである。 M_{vn} を対角行列にすることで作業空間の各軸方向の力応答は完全に非干渉化できる。

作業空間での外乱オブザーバに基づいた、作業空間の非干渉化

一方、作業空間オブザーバは (3.9) 式に基づいて構成される。基本的な方程式の形は関節空間と同一であることから、作業空間でのオブザーバ構成も関節空間オブザーバと同一になる。したがって、作業空間オブザーバの基本ブロック線図は図 3.2 のようになる。ここで、 Q および P_n は次のように定義される、ローパスフィルタ、プラントのノミナルモデルである。

$$Q = \begin{bmatrix} \frac{g}{s+g} & & 0 \\ & \ddots & \\ 0 & & \frac{g}{s+g} \end{bmatrix}, P_n = \begin{bmatrix} \frac{1}{M_{nx}s^2} & & 0 \\ & \ddots & \\ 0 & & \frac{1}{M_{ny}s^2} \end{bmatrix} \quad (3.14)$$

ただし M_{nx}, M_{ny} はそれぞれ x 方向, y 方向の等価質量行列である。ローパスフィルタ Q のカットオフ周波数を十分大きくとることによって、作業空間での外乱 F_{dis} の影響は遮断

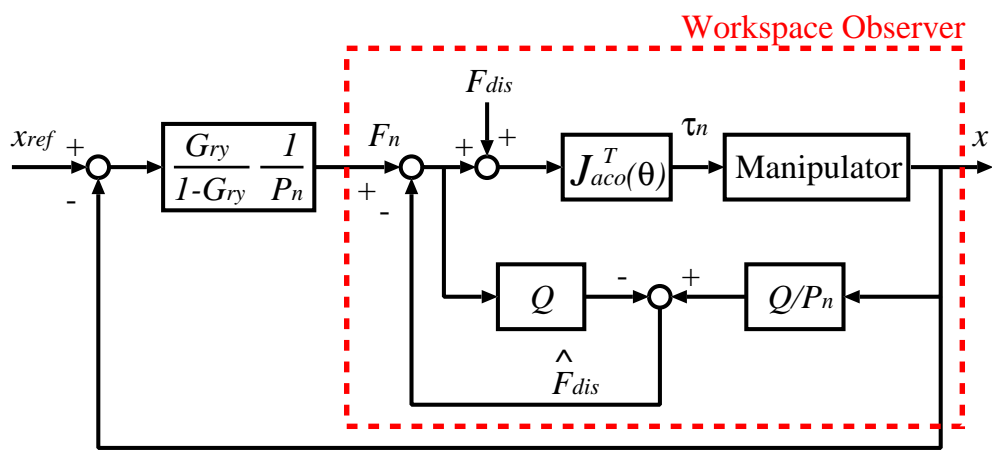


図 3.3: 外乱オブザーバを用いたロバスト位置制御系

される。したがって、作業空間オブザーバに基づいたロボットでは、作業空間での等価質量行列が外乱 F_{dis} の影響を受けずに固定化される。また P_n を対角行列とすることにより、作業空間において非干渉化が実現できる。

本節で述べたように、関節空間と作業空間のどちらで外乱オブザーバを構成しても、作業空間における非干渉化は達成できる。しかし、関節空間オブザーバに基づく非干渉化の際には座標変換としてヤコビの逆行列を用いるが、作業空間オブザーバに基づく非干渉化ではヤコビの転置行列を用いるため、座標変換の計算量の観点からは作業空間オブザーバを用いた手法が好ましいと言える [9]。

従って本研究では、作業空間オブザーバを用いて作業空間の非干渉化を行なう。

3.1.4 ロバストな位置制御系の実現

以上でプラントの、作業空間における非干渉化が達成され、作業空間各軸の力応答は P_n にノミナル化される。さらにプラントをノミナル化し、ロバストな位置制御系とするために、位置フィードバックコントローラとして図 3.3 に示すように、

$$\frac{G_{ry}}{1 - G_{ry}} \frac{1}{P_n} \quad (3.15)$$

を用いる。これにより、目標値応答が G_{ry} (希望目標値応答と呼ぶ) となる。このように外乱オブザーバを用いた制御系は、目標値追従特性と外乱抑圧特性が独立に設計できる、ロバストな位置制御系が実現される [10]。

3.2 カメラの逆モデルによるビジュアルサーボ系の線形化

視覚センサにより、画像特長量ベクトル ξ は (2.3) 式に示したように、 $\theta_1 + \theta_2$ だけ回転し、拡大 (縮小) した形となる。制御系内にこの非線形な要素が、解析的な制御系設計にとってのもう一つの問題点である。

本研究では、この非線形要素を打ち消すために、カメラモデル ((2.3) 式) の逆行列である $\iota^{-1}(\theta)$ をフィードバックループに挿入する (図 2.6)。マニピュレータの関節角である θ はエ

ンコーダによりリアルタイムで測定可能であり、 f/z は実験によって事前に測定できるので、 $\iota^{-1}(\theta)$ の実現は簡単である。これにより、系の非線形要素は全て線形化される。

第4章 制御器の設計

4.1 極配置に基づいた状態フィードバック

4.1.1 状態フィードバックの設計

作業空間で非干渉化された位置制御系の指令値に対する x, y 各軸の特性は G_{ry} であるため、位置制御系を制御対象としてビジョンループの制御周期で離散化する。その結果得られた、離散時間制御対象を (4.1), (4.2) 式とする。

$$\mathbf{x}[i+1] = \mathbf{A}\mathbf{x}[i] + \mathbf{B}u[i] \quad (4.1)$$

$$y[i] = \mathbf{C}\mathbf{x}[i] \quad (4.2)$$

本研究では図 4.1 のような状態フィードバックにより、制御系設計を行う。制御系への入力为目标物体位置 s_{obj} とし、出力をカメラ位置 s_{cam} としてとらえる。また、この場合の制御系は目標物体位置を位置指令とするサーボ系であり、ステップ状の目標値を仮定して、閉ループ内に内部モデルを持つように、 $\frac{1}{z-1}$ を挿入する。図 4.1 から、

$$\begin{aligned} z[i+1] &= z[i] + e[i-1] \\ &= z[i] + y_r[i-1] - \mathbf{C}\mathbf{x}[i-1] \end{aligned} \quad (4.3)$$

であるので、(4.1) 式と (4.3) 式をあわせた拡大系の状態方程式は

$$\begin{bmatrix} \mathbf{x}[i] \\ \mathbf{x}[i+1] \\ z[i+1] \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{A} & \mathbf{0} \\ -\mathbf{C} & \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} \mathbf{x}[i-1] \\ \mathbf{x}[i] \\ z[i] \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{B} \\ 0 \end{bmatrix} u[i] + \begin{bmatrix} \mathbf{0} \\ \mathbf{d} \\ y_r[i-1] \end{bmatrix} \quad (4.4)$$

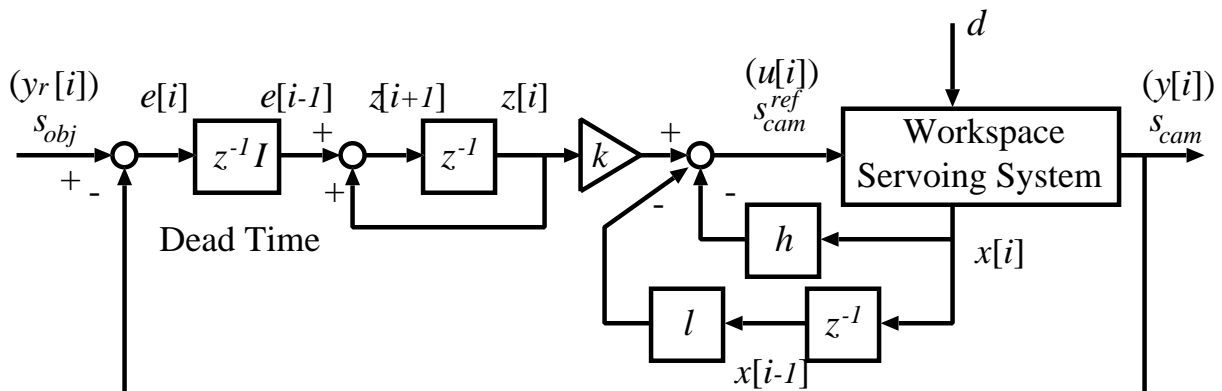


図 4.1: 状態フィードバック法

と表される。このとき、

$$u[i] = \begin{bmatrix} -l & -h & k \end{bmatrix} \begin{bmatrix} x[i-1] \\ x[i] \\ z[i] \end{bmatrix} \quad (4.5)$$

の状態フィードバックを施すと、系全体は、

$$\begin{bmatrix} x[i] \\ x[i+1] \\ z[i+1] \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \mathbf{I} & \mathbf{0} \\ -\mathbf{Bl} & \mathbf{A} - \mathbf{Bh} & \mathbf{Bk} \\ -\mathbf{C} & \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} x[i-1] \\ x[i] \\ z[i] \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{d} \\ y_r[i-1] \end{bmatrix} \quad (4.6)$$

となる。ただし h, l は、

$$h = \begin{bmatrix} h_1 & h_2 \end{bmatrix}, l = \begin{bmatrix} l_1 & l_2 \end{bmatrix} \quad (4.7)$$

で表されるゲイン行列である。また、

$$\hat{\mathbf{A}} = \begin{bmatrix} \mathbf{0} & \mathbf{I} & \mathbf{0} \\ -\mathbf{Bl} & \mathbf{A} - \mathbf{Bh} & \mathbf{Bk} \\ -\mathbf{C} & \mathbf{0} & 1 \end{bmatrix} \quad (4.8)$$

をシステム行列と呼ぶ。拡大系の状態方程式 (4.4) は可制御ではなく、システム行列の全ての固有値をフィードバックゲインである k, h, l によって自由に操作できるわけではない。しかし操作できない固有値は $z = 0$ にあるため、このシステムは可安定であるといえる。すなわち、極配置に基づいた状態フィードバックによってシステム行列の極を指定することにより、指令値である目標物体位置からの応答が自由に調節できる。また、極を全て零に設計することにより、有限制定義サボ系が実現される。

4.1.2 位置誤差収束の保証

以上で拡大系の状態が安定であることが示されたが、本節では、この制御系が定常誤差を生じないことを示す。偏差を

$$\begin{aligned} e[i] &= y_r[i] - y[i] \\ &= y_r[i] + \begin{bmatrix} -\mathbf{C} & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} x[i-1] \\ x[i] \\ z[i] \end{bmatrix} \end{aligned} \quad (4.9)$$

と定義する。また (4.6) 式から、状態 x と z の定常値は、

$$\begin{bmatrix} x[\infty] \\ x[\infty] \\ z[\infty] \end{bmatrix} = \begin{bmatrix} \mathbf{I} & -\mathbf{I} & \mathbf{0} \\ \mathbf{Bl} & \mathbf{I} - \mathbf{A} + \mathbf{Bh} & -\mathbf{Bk} \\ \mathbf{C} & \mathbf{0} & 0 \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{0} \\ \mathbf{d} \\ y_r[\infty] \end{bmatrix} \quad (4.10)$$

と表される。これを (4.9) 式に代入すると、

$$e[\infty] = y_r[i] + \begin{bmatrix} -\mathbf{C} & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{I} & -\mathbf{I} & \mathbf{0} \\ \mathbf{Bl} & \mathbf{I} - \mathbf{A} + \mathbf{Bh} & -\mathbf{Bk} \\ \mathbf{C} & \mathbf{0} & 0 \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{0} \\ \mathbf{d} \\ y_r[\infty] \end{bmatrix} \quad (4.11)$$

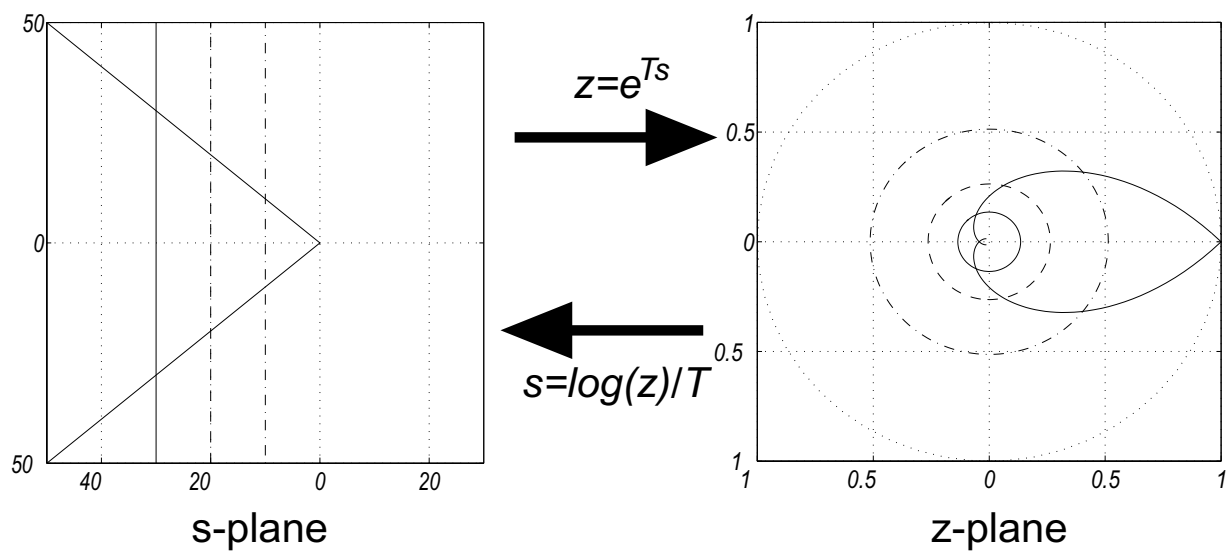


図 4.2: s 平面と z 平面との対応

を得る。ここで、

$$\begin{bmatrix} C & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} I & -I & 0 \\ Bl & I - A + Bh & -Bk \\ C & 0 & 0 \end{bmatrix} \quad (4.12)$$

の関係があることに注目すると、(4.11) 式は、

$$e[\infty] = y_r[\infty] - \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ d \\ y_r[\infty] \end{bmatrix} = 0$$

となる。すなわち、任意の定値外乱 d とステップ状の目標値入力 y_r に対して定常偏差が 0 になることが確認できる。加えて、この導出がパラメータ A, B, C, h, l, k の値に依存しないことに注意すると、定常ロバスト性の条件、

- 持続的な外乱入力が存在しても、制御量が与えられた目標入力 (目標値) に定常偏差なく追従する

も満たされていることがわかる。以上より、図 4.1 の制御系はフィードバックゲイン h, l, k を適当に選んで拡大系を安定化しさえすれば、1 型のロバストサーボ系になることが示された。

4.2 極の位置による応答の変化

4.1.1 節に述べた状態フィードバックにより、ビジュアルサーボシステムの極を自在に配置できるようになる。本節では、離散系のシステム行列の固有値の配置により、応答がどのように変化するかを調べ、単純なモデルによるシミュレーションにより検討する。

連続時間における s 平面と離散時間における z 平面との関係は、サンプリング周期 T を用いて

$$z = e^{Ts} \quad (4.13)$$

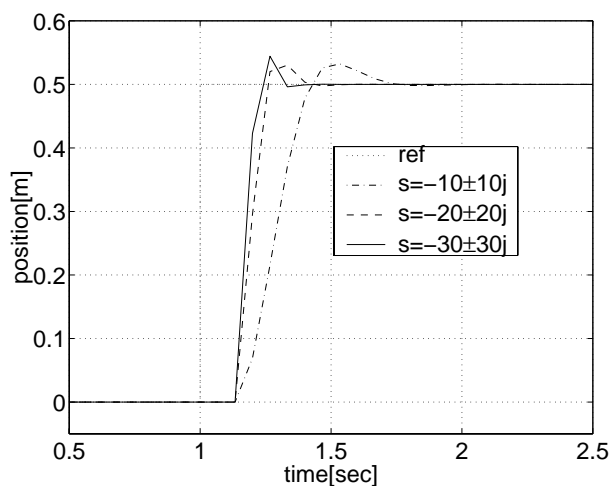


図 4.3: $s = -10 \pm 10j, -20 \pm 20j, -30 \pm 30j$ に極を配置した場合のステップ応答

と表される。(4.13) 式より、 s 平面と z 平面との対応は、図 4.2 のようになる。図 4.2 は、 s 平面上の直線

$$s = \alpha \pm \alpha j \quad (4.14)$$

$$s = -10 \pm \beta j \quad (4.15)$$

$$s = -20 \pm \beta j \quad (4.16)$$

$$s = -30 \pm \beta j \quad (4.17)$$

を z 平面に写像したものである。このように、 s 平面の虚軸に平行な直線は、 z 平面では $z = 0$ を中心とする円に対応する。また、 s 平面において原点を始点とし、負の実軸から $\pm 45[\text{deg}]$ の傾きの半直線 $s = \alpha \pm \alpha j$ の z 変換は、 $z = 1$ を始点とし、 $z = 0$ へ収束する曲線となる。

極配置に基づいた状態フィードバックによって理論的にはどのような応答になるのかを確かめるために、 $T = 66.6[\text{msec}]$ として、一次元の位置制御系モデル

$$G_{ry} = \frac{\omega_n^2}{s^2 + 2\omega_n s + \omega_n^2} \quad (4.18)$$

をプラントとするシミュレーションを行った。ここではプラントは一次元モデルであるので、カメラモデル $\iota(\theta)$ による $s_{obj} - s_{cam}$ から特徴量 ξ への変換、及び逆カメラモデル $\iota^{-1}(\theta)$ による特徴量 ξ から $s_{obj} - s_{cam}$ への変換は行わない。ここで、 $\omega_n = 15[\text{rad/sec}]$ である。ただし、システム行列 \hat{A} の五つの極のうち操作できない極 ($z = 0$) 以外の四つの極を、 s 平面上で

$$s = -10 \pm 10j, -10 \pm 10j \quad (4.19)$$

$$s = -20 \pm 20j, -20 \pm 20j \quad (4.20)$$

$$s = -30 \pm 30j, -30 \pm 30j \quad (4.21)$$

のように配置し、これを (4.13) 式によって z 平面の極に変換した。ステップ位置指令に対する応答の変化を図 4.3 に示す。 s 平面状での応答と同様に、適度なオーバーシュートを持ち、極の位置によって応答速度が変化していることが理解される。

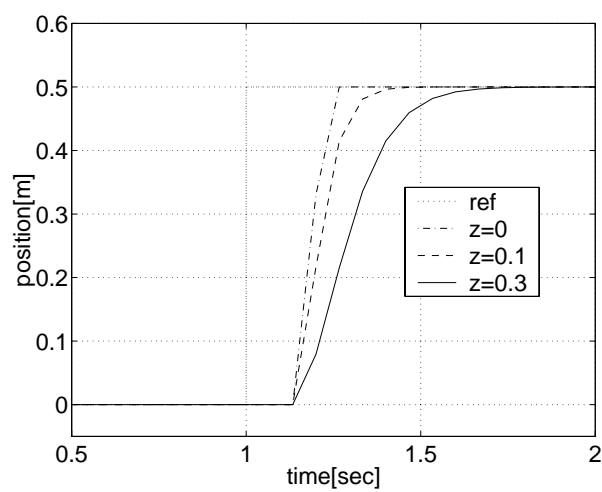


図 4.4: z 平面実軸上 (正方向) に極を配置した場合のステップ応答

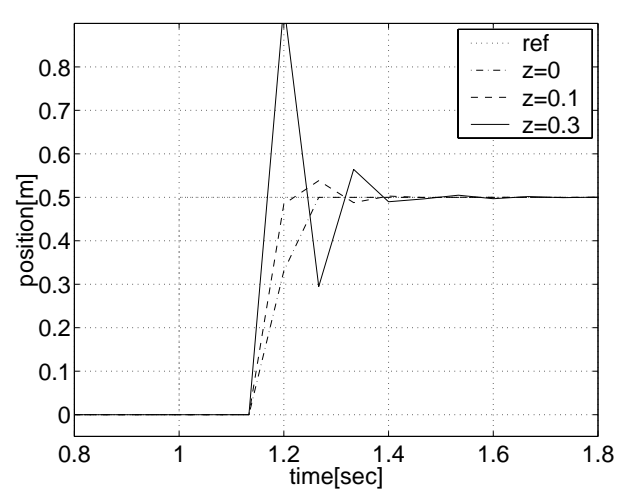


図 4.5: z 平面実軸上 (負方向) に極を配置した場合のステップ応答

さらに図 4.4 に、極を 4 重根とし、 $z = 0$ (有限整定), $z = 0.1, 0.3$ と変化させた場合のステップ応答を示す。 z 平面で単位円内正方向の実軸は、 s 平面での左半平面の実軸に対応するので、応答はオーバーシュートのない、臨界制動的なものとなっている。

次に、同じく 4 重根を、 $z = 0$ (有限整定), $z = -0.1, -0.3$ と変化させた場合のステップ応答を示す。 z 平面で単位円内負方向の実軸は、単位円内であるので安定ではあるものの、収束の速度は $z = 0$ から同じ距離にある正方向の極と等しいが、収束が振動的になっていることがわかる。

第5章 実験による比較・検討

5.1 実験装置の概要

提案する設計手法の有効性を検証するため、2軸DD(ダイレクト・ドライブ)ロボットを用いた実験を行った。2軸DDロボットの写真を図5.1に示す。

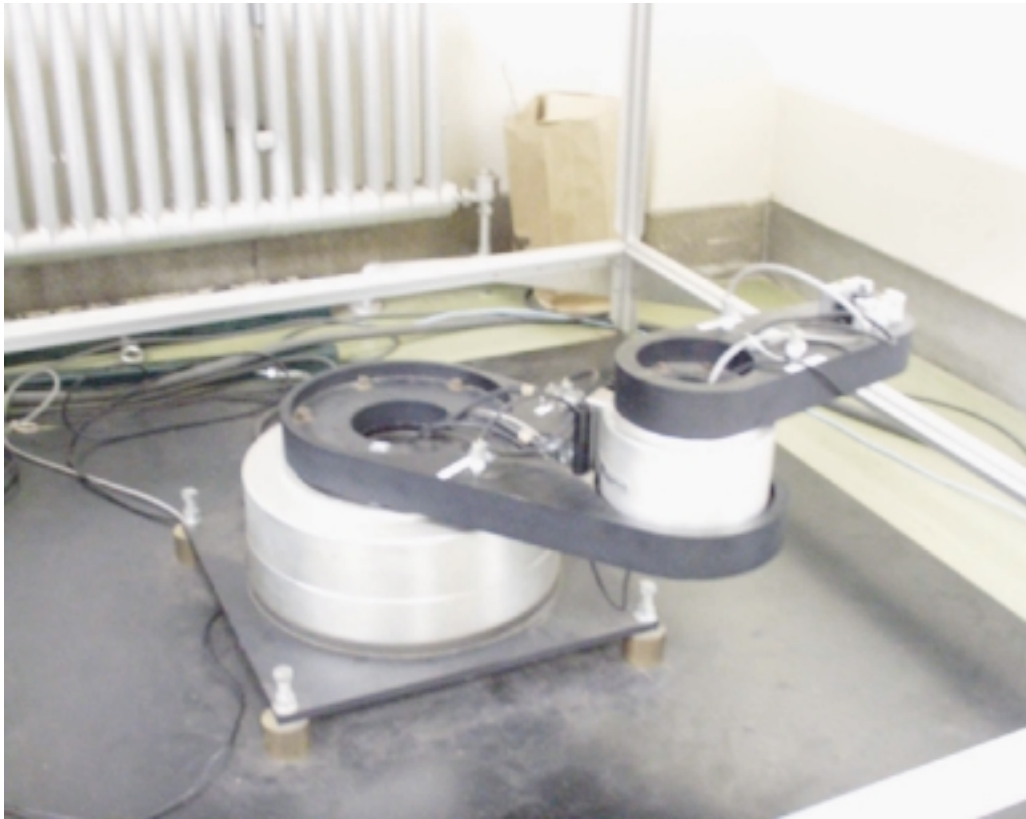


図 5.1: 実験に用いた 2 軸 DD ロボット

5.1.1 ソフトウェア構成

本実験では、一定のサンプリング周期で計算を行うことや、カメラが得た画像情報を PC にリアルタイムで受け渡すことが必要である。しかし Windows を始め、多くの OS ではタスクの処理実行権が規定の時間で回ってくることが保証されていない。そのため特に PC に負荷がかかった際に、別のタスクの処理の影響でリアルタイムの動作ができない恐れがある。そこで、本実験ではこのようなリアルタイム性を満たす OS として、RT-Linux(リアルタイム Linux)を用いた。

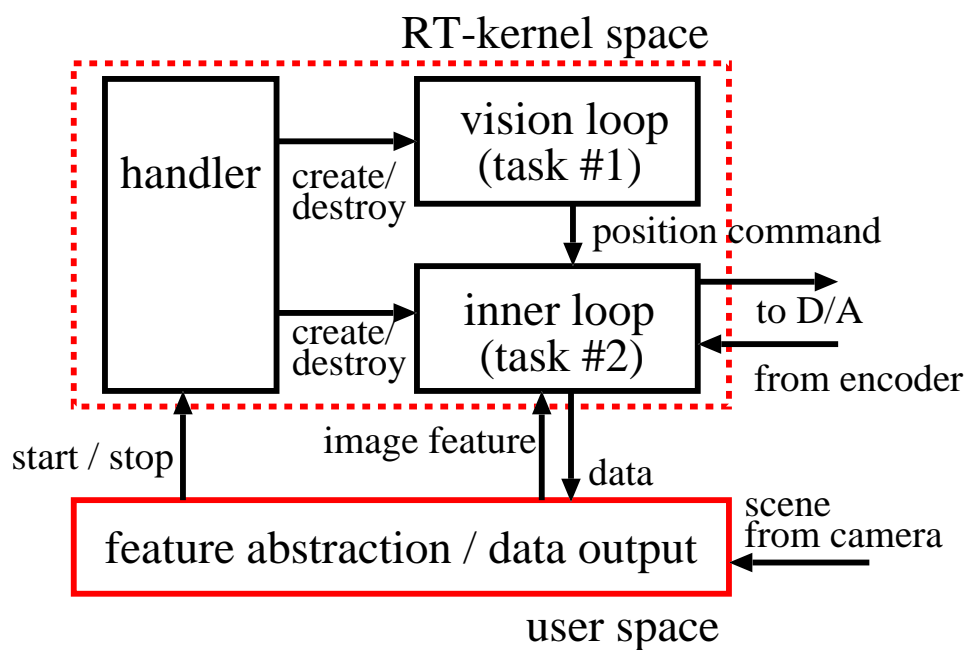


図 5.2: 実験システムのソフトウェア構成

RT-Linux は、Linux カーネルの上位にリアルタイムカーネルを付加する形により、ミリ秒以下のマルチタスク・リアルタイムシステムの構築を実現した OS である。従来の Linux カーネルはユーザ空間となり、最も優先度の低いリアルタイムタスクの一つになる。具体的には、リアルタイム性の優先度が高い処理は RT-カーネル上で行ない、必ずしも優先度が高くない処理はユーザ空間で行なえばよい。

実験システムのソフトウェア構成の概念図を図 5.2 に示す。本実験では、インナーループである作業空間位置制御系及びビジョンループの制御則の計算はカーネル空間上で行なった。ただし、カメラ画像のキャプチャ、画像処理や特徴量の算出は、ユーザ空間上で行なった。これは、カーネル上の制御則の処理を規定時間 (サンプリング時間) で終わらせるためには、リアルタイムカーネルでの演算を少なくする必要があるからである。

ユーザ空間上で算出された特徴量は、サンプリング時間ごとに RT-カーネルに受渡す。特徴量の計算のループは、33[msec] ごとに出力される CCD カメラからの画像信号に同期して行なわれるため、特徴量の算出はリアルタイムで行なわれているとしてよい。

5.1.2 ハードウェア構成

システムのハードウェア構成を図 5.3 に示す。またシステム全体の写真を図 5.4 に示す。

PC への外部からの入力、カメラからの画像とエンコーダからのパルス波である。カメラからの画像情報は 33[msec] ごとにビデオキャプチャボードに取り込まれ、画像処理及び特徴量の算出に用いられる。

またエンコーダはロボットの各関節に取り付けられ、関節の変位角に応じた数のパルス波を出力する。パルス波はカウンタボードによって計数され、パルス波数が関節角変位に変換される。

PC はこれらの情報をもとにプログラム内で制御則を計算し、D/A ボードに電圧指令情報を出力する。

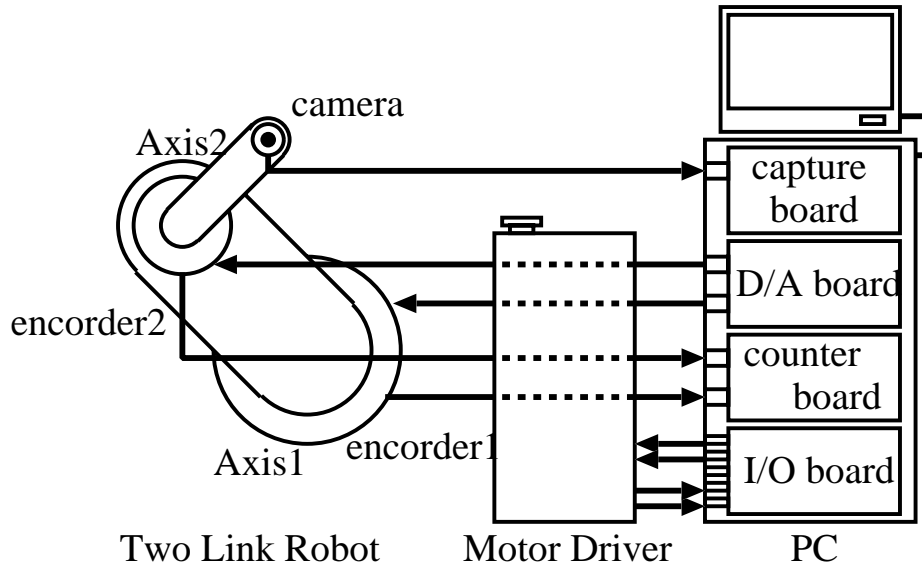


図 5.3: 実験システムの構成

パラレルI/O ボードはモータドライバの電源の ON/OFF や指令電圧のリミッタスイッチ・緊急停止スイッチの作動、ドライバの初期化など、ロボットの動作の根幹となる情報をドライバとやりとりする。

目標物体は、ステッピングモータを用いて円盤を回転させ、その円盤の端部に印を付けてそれを追従させる構成とした。円盤は、2軸 DD ロボットの上に組まれたやぐらに取り付けた。ステッピングモータの写真を図 5.5 に、CCD カメラから撮影した目標物体の写真を図 5.6 に示す。

5.2 実験結果及び考察

5.1 節に述べたような実験システムを用いて実験を行なった。

ロボットの各パラメータは第1軸の長さが $L_1 = 0.359[\text{m}]$ 、第2軸の長さが $L_2 = 0.24[\text{m}]$ 、 $F = f/z = 5.5$ である。また、目標物体は半径 $5[\text{cm}]$ の円周軌道上を角速度 $0.98\pi[\text{rad/sec}]$ で等速円運動させるか、あるいは静止させるものとした。作業空間位置制御系の希望目標値応答 G_{ry} は、 x, y 各軸で

$$G_{ry} = \frac{\omega_n^2}{s^2 + 2\omega_n s + \omega_n^2} \quad (5.1)$$

とし、 $\omega_n = 15[\text{rad/sec}]$ とした。また、 Q のゲインは $g = 150[\text{rad/sec}]$ とし、作業空間で非干渉化するサンプリング周期は $1[\text{msec}]$ とした。ビジョンループのサンプリング周期は $66.6[\text{msec}]$ とし、特長量の算出までにはちょうど一周期のむだ時間を持たせた。

状態フィードバックにあたっては、 h, l, k という五つのパラメータによって実際に操作できるのは四つの極であるので、必要なパラメータの数は四つである。したがって、パラメータのうち一つは極配置には必要ない。実験では作業空間での速度を求める際に、インナーループのサンプリング周期で後退差分法により第一軸、第二軸の各軸角速度を求め、ついで (3.3) 式によるロボットのヤコビアンを用いた速度生成法を利用している。後退差分法で角速度を求める際に速度情報が量子化を起し、情報が劣化している。したがって



図 5.4: 実験システム全景

本研究では極配置の際に、 $l_2 = 0$ としてその他四つのパラメータを一意に決定した。 l_2 は $x[i-1]$ の二つ目の要素、つまり X, Y 各軸に対しての一ステップ前の速度である。

また、特徴量の算出にあたっては、 120×90 [pixels] で撮影した画像から特徴量を算出し、特徴量を x 方向は $-1 < \xi_x < 1$ で、 y 方向は $-4/3 < \xi_y < 4/3$ で正規化した。 x 方向と y 方向で正規化の大きさが異なるのは撮像した画像の縦横比を保存するためである。

5.2.1 作業空間位置制御系の実現と、作業空間位置指令に対する応答

ビジュアルサーボ系において制御対象となる作業空間位置制御系は、3.1.4 節に述べたように、二自由度制御系によって構成し、ロバストな制御系となっていなければならない。

本節では実際に構成した作業空間位置制御系が、希望目標値応答 G_{ry} にどれだけ近いものとなっているか検討する。

シミュレーションにより生成された希望目標値応答 G_{ry} と、実際に構成された作業空間位置制御系のステップ応答の比較を行うために、シミュレーションのモデル G_{ry} と実際の作業空間位置制御系に対して、周期 2.0 [sec]、振幅 0.05 [m] の方形波位置指令を与えた。

図 5.7 に x 方向のステップ応答を、図 5.8 に y 方向のステップ応答を示す。

x 方向のステップ位置指令に対する応答は、シミュレーションによる G_{ry} の応答と実験に用いた位置制御系の応答がほぼ重なっており、 x 方向に関しては、ビジュアルサーボ系のプラントである位置制御系の特性は、希望目標値応答と一致しているとみなしてよい。

y 方向のステップ応答は、立上りは理想的な G_{ry} に比べて速い。しかし指令値の 90% 程度のところで一旦立上りが止まり、再び動き出すといった応答を示している。しかし整定ま

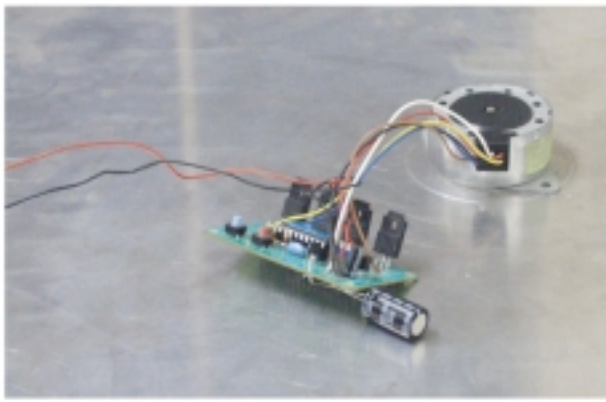


図 5.5: 目標物体を回すステッピングモータ

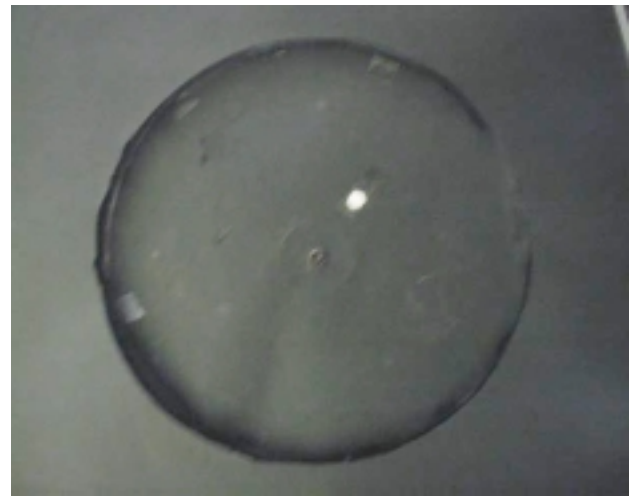


図 5.6: カメラから見た目標物体

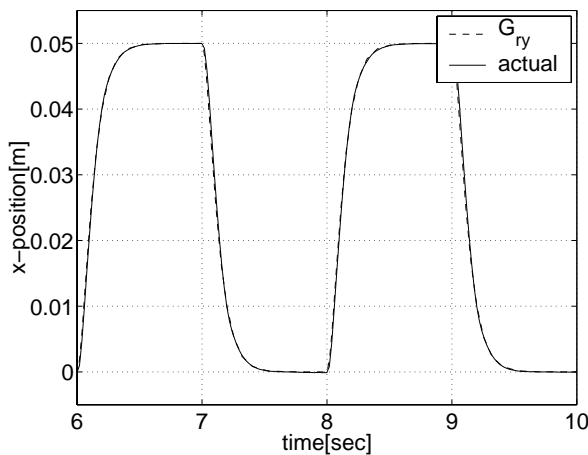


図 5.7: 理想的な G_{ry} と実際の作業空間位置制御系の、 x 方向ステップ応答の比較

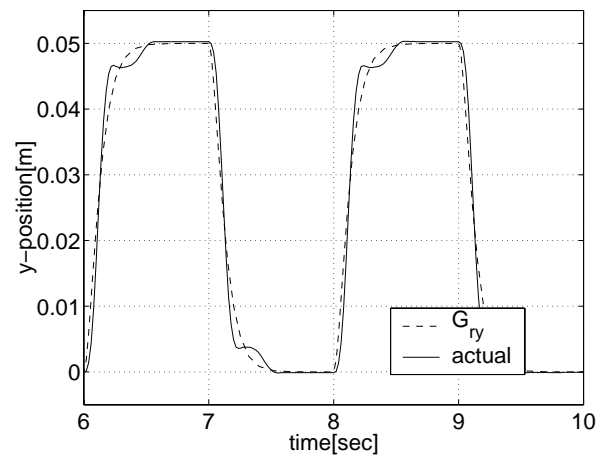


図 5.8: 理想的な G_{ry} と実際の作業空間位置制御系の、 y 方向ステップ応答の比較

での時間はシミュレーションによる G_{ry} の応答とほぼ等しく、この程度のモデルとのずれは、制御系の安定性にはほとんど影響しないと考えてよいと思われる。

5.2.2 目標物体を静止させた場合

目標物体を静止させた状態で実験を行い、初期位置からのステップ応答を見る。四つの極を実軸上に4重根として配置した場合の X 方向カメラ軌道を図 5.9 に、 Y 方向カメラ軌道を図 5.10 に示す。また、特徴量の時間変化を図 5.11, 5.12 に示す。ただし凡例における数値は4重極を示す。

極を 0.1, 0.3, 0.4 と $z = 0$ から遠ざけるにつれ、応答のオーバーシュートは小さくなり、ゆっくりとした応答となっている。逆に極を $z = 0$ に近付けるにつれて、速い応答になっていることがわかる。

しかし、極を全て零とした有限整定サーボ系では、オーバーシュートが大きくなりすぎ、特に Y 方向について振動的な応答となっている。

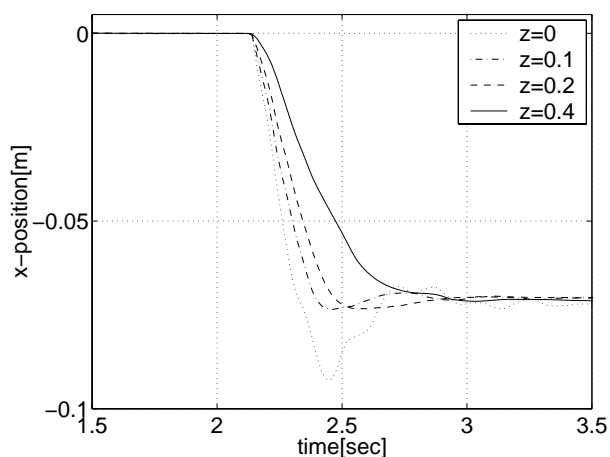


図 5.9: 目標物体が静止している場合のカメラ位置 X の時系列

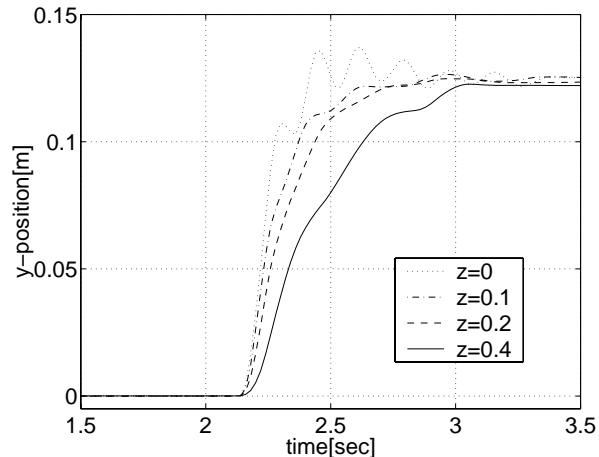


図 5.10: 目標物体が静止している場合のカメラ位置 Y の時系列

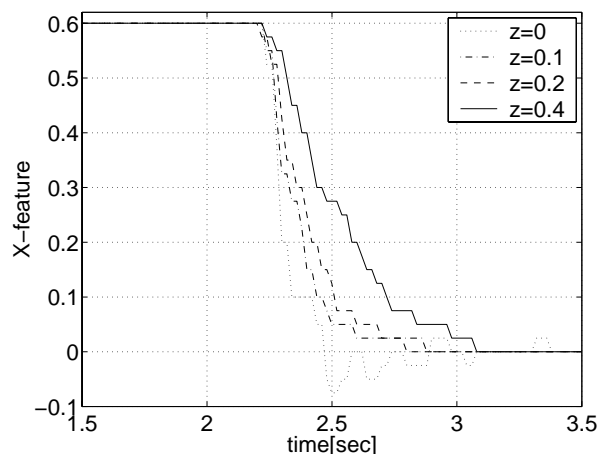


図 5.11: 目標物体が静止している場合の特徴量 ξ_x の時系列

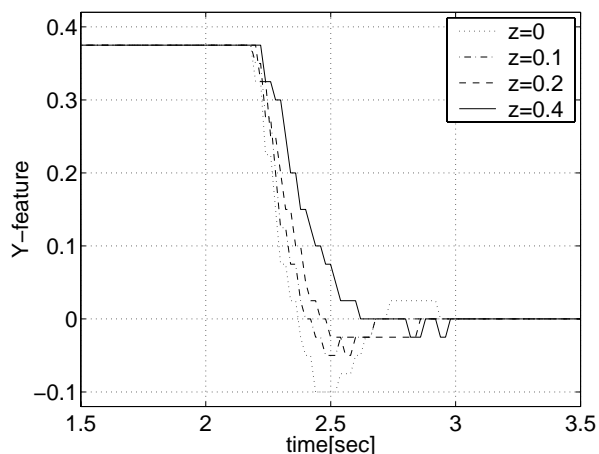


図 5.12: 目標物体が静止している場合の特徴量 ξ_y の時系列

Y 方向での応答がこのような振動的になっているのは、5.2.1 節で示したように、 Y 方向での作業空間位置制御系のノミナル化が完全でないことが影響していると考えられる。

また、極を負の実軸上においたときには、系は安定性を失った。これは、極が z 平面の負方向の実軸上にあるときには、4.2 節に述べたように、収束が振動的となり、不安定になりやすいためと思われる。

5.2.3 目標物体を円運動させた場合

図 5.13 に目標物体を等速円運動させた場合の X 方向の位置、図 5.15 に画像特徴量の時間変化を示す。ただし目標とする画像特徴量は原点であるので、特徴量が小さいほど追従誤差は小さい。4 重根の極の場合、その位置が正方向の z 軸上で単位円内ならば、安定性に問題はないことがわかる。また、図 5.9 においてオーバーシュートが少なく、かつ速い応答を示した、 $z = 0.2$ に 4 重根を置いた場合に最も追従誤差が小さいという結果が得られた。

しかし、 Y 方向の軌道は若干の振動性を持っている。これもまた目標物体が静止している場合と同様に、作業空間位置制御系が Y 方向に関して完全にノミナル化されていないこ

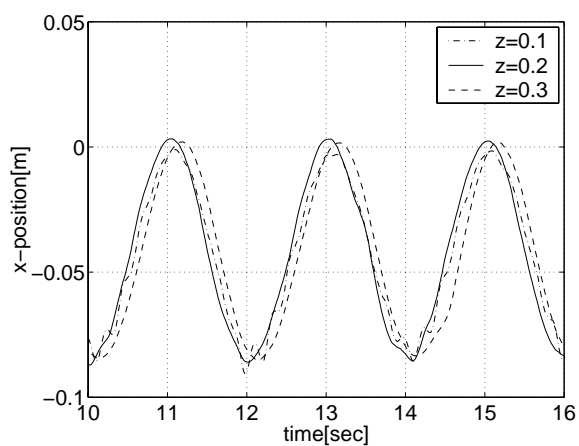


図 5.13: 目標物体が円運動している場合のカメラ位置 X の時系列

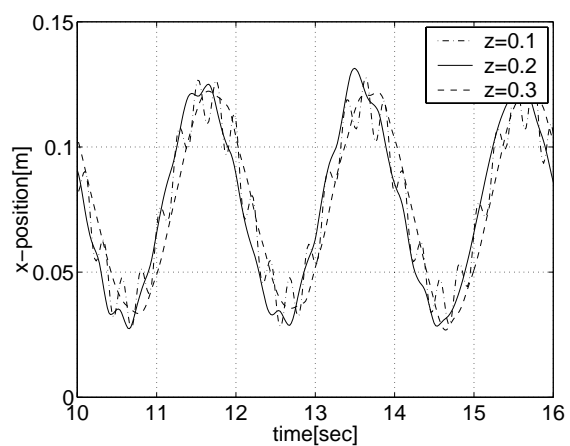


図 5.14: 目標物体が円運動している場合のカメラ位置 Y の時系列

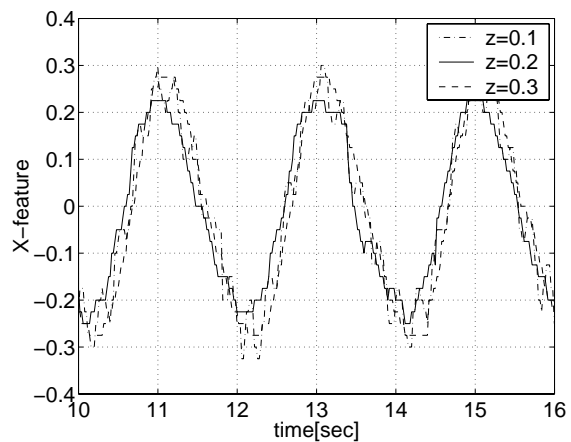


図 5.15: 目標物体が円運動している場合の画像特徴量 ξ_x の時系列

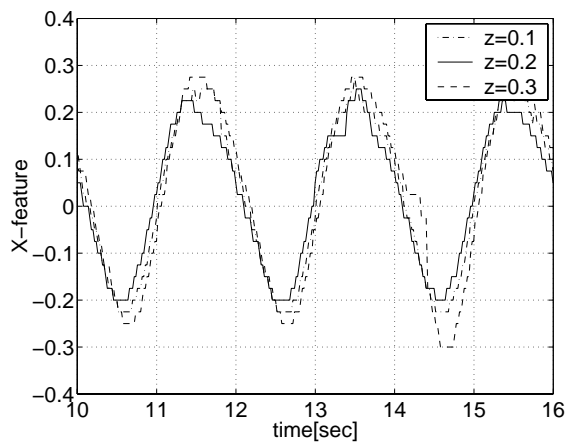


図 5.16: 目標物体が円運動している場合の画像特徴量 ξ_y の時系列

とが原因であると思われる。

第6章 結論

本論文では画像ヤコビアンを用いないビジュアルサーボ系の設計法として、プラントを作業空間において非干渉化し、極配置に基づいた状態フィードバック法を用いた設計法を提案した。提案する設計手法の特長は以下の三点である。

1. 制御対象を非干渉化された位置制御系とすることにより、画像ヤコビアンや、カメラモデルといったビジュアルサーボ系特有の非線形要素を線形化できる。
2. 線形化により、極配置法などの解析的な制御系設計手法を適用でき、見通しのよい制御系設計が可能となる。
3. 画像空間から関節空間へ直接変換を行なう画像ヤコビアンを用いないため、作業空間でのタスクの記述が可能となる。

さらに、極の配置のしかたで過渡応答を自由に変化させられることを、単純化したモデルを用いたシミュレーションによって示した。

最後に、実験によってこの設計法の有効性を検証した。実験により、実際のビジュアルサーボ系においても、位置指令として見なした目標物体に対する応答の立上りの速さ、オーバーシュートなどの過渡応答特性を、自由に変化させられることが示された。

今後の課題としては、物体の運動予測の方法と、運動予測を行なった際のコントローラの設計法や、安定性に関する考察が挙げられる。

謝 辞

私が堀研究室に在籍していたのはわずか一年でしたが、中身の濃い大変有意義な一年間でした。多くの御指導、御助言を頂き、学部生生活の四年間のなかで最も成長できた年であったと思います。

研究においても、多くの方々に支えて頂いたからこそ成し得たのであり、私一人の力では到底できなかったであろうと思います。本当に、感謝しても感謝しきれません。重ねて、素晴らしい人々のいる堀研究室で研究することができ、大変に幸運であったと思います。

堀洋一教授には、毎日大変お忙しい中、懇切丁寧、かつ適切なご指導を頂きました。研究についてのことのみならず、工学者としてのありかたや、心構えなどもご指導くださいました。ここに深く感謝致します。

内田利之技官には、実験機製作のみならず、一般の電子機器や機械の扱い方まで、未熟者の私が全く知らないことを教えて下さいました。ここに深く感謝致します。

研究員のチャンダン・チャクラボラティさん、宋申民さんには、研究テーマについて、またご家族や故郷のことなど、興味深いお話を聞かせて頂きました。博士課程3年の及部七朗斎さんには、研究室の全般に渡って本当に様々なことに関してお世話になりました。またディスプレイを長期間に渡って貸して下さいました。ここに深く感謝致します。

博士課程2年の関弘和さんには、RT-linux やロボットの動かし方、カメラの画像処理についてのみならず、机が隣だったこともあり、ちょっとしたことで質問させて頂きました、またそのたびに嫌な顔一つせず親切に教えて下さいました。ここに深く感謝致します。

博士課程1年の呉世訓さんには、一緒に過ごしたのは2ヵ月程でしたが、Matlab の使い方などを御指導頂きました。同じく博士課程1年の馬澄斌さんには、実験室で寒い中に手伝って下さいました。ここに深く感謝致します。

修士課程2年の坂東信尚さんには、研究全般に渡って様々なことをご指導頂きました。また、不適切な部分が多い私の論文を丁寧に読んでくださり、適切に助言して下さいました。同じく修士課程2年の畠直輝さんには、PCに関することの他に制御理論に関しても適切なご指導を頂きました。ここに深く感謝致します。

修士課程1年の井上友子さんは、初歩的な質問にも親身に教えて下さいました。同じく修士課程1年の呉俊徳さんは、研究テーマが近いこともあり、共に議論をして下さいました。同じく修士課程1年の岡野隆宏さんは、研究の進め方のコツなどを教えて下さいました。同じく修士課程1年の戴建華さんには、計算機について、特にUNIX の使い方についてご指導頂きました。同じく修士課程1年の張炳勳さんには、研究についての議論を親身にして下さいました。ここに深く感謝致します。

卒論生の磯将人君、大埜健君、佐藤仁彦君、長瀬啓明君には、夜中まで一緒に作業をしたり、実験を手伝ってくれたり、研究についての議論をただけでなく、楽しい学生生活をするうえで非常に大きな支えとなってくれました。ここに深く感謝致します。

最後になりましたが、秘書の木村麻里子さんには、研究室をなごやかにして下さい、お茶の時間にはお菓子やコーヒーを用意して下さいました。ここに深く感謝致します。

参考文献

- [1] 石川正俊, "ビジュアルサーボイングの現状と将来", 計測と制御, Vol.40, No.9, pp.615-617, 2001.
- [2] 橋本浩一, "視覚フィードバック制御 -静から動へ", システム/情報/制御, Vol.38, No.12, pp.21-27, 1994.
- [3] 岡崎俊太郎, 藤本博志, 堀洋一, "むだ時間を持つ視覚フィードバック制御系における高速でロバストな制御器の一提案", 電気学会産業計測制御研究会 IIC-00-18, pp.103-108, 2001.
- [4] 橋本浩一, 海老根巧, 木村英紀, "ビジュアルサーボイング-最適制御によるアプローチ-", 日本ロボット学会誌, Vol.12, No.5, pp.773-778, 1994.
- [5] 橋本浩一, 井上貴博, 木村英紀, "ビジュアルサーボイング-非線形制御アプローチ-", 日本ロボット学会誌, Vol.13, No.2, pp.263-269, 1995.
- [6] 遠藤公誉, 田中弘一, 荒川賢一, 武川直樹, "予見制御による視覚軌道追従制御法", 日本ロボット学会誌, Vol.15, No.4, pp.85-92, 1997.
- [7] 橋本浩一, 木村英紀, "ビジュアルサーボイング-非線形オブザーバアプローチ-", 日本ロボット学会誌, Vol.13, No.7, pp.92-99, 1995.
- [8] 猿田訓彦, 藤本博志, 堀洋一, "運動推定オブザーバによる運動項推定を用いたビジュアルサーボ", 電気学会産業計測制御研究会 IIC-01-13, 2001.
- [9] 村上俊之, 大西公平, "ロバスト制御に基づいた多自由度ロボットの安定性及び作業空間での非干渉制御に関する一考察", 電気学会論文誌 D, Vol.113, No.5, pp.639-646, 1993.
- [10] 梅野孝治, 堀洋一, "2自由度ロバストサーボ系の構成とマニピュレータの運動制御への応用", 電気学会論文誌 D, Vol.110, No.11, pp.1163-1170, 1990.

発表文献

[1] 高橋啓, 坂東信尚, 堀洋一, ”作業空間の非干渉化に基づいたビジュアルサーボ系の新しい設計法”, 電気学会産業計測制御研究会, IIC-02-21, 2002(発表予定).