

Title	動物体を扱うロボットの運動計画と制御
Author(s)	阪口, 健
Citation	大阪大学, 1993, 博士論文
Version Type	VoR
URL	https://doi.org/10.11501/3065938
rights	
Note	

Osaka University Knowledge Archive : OUKA

<https://ir.library.osaka-u.ac.jp/>

Osaka University

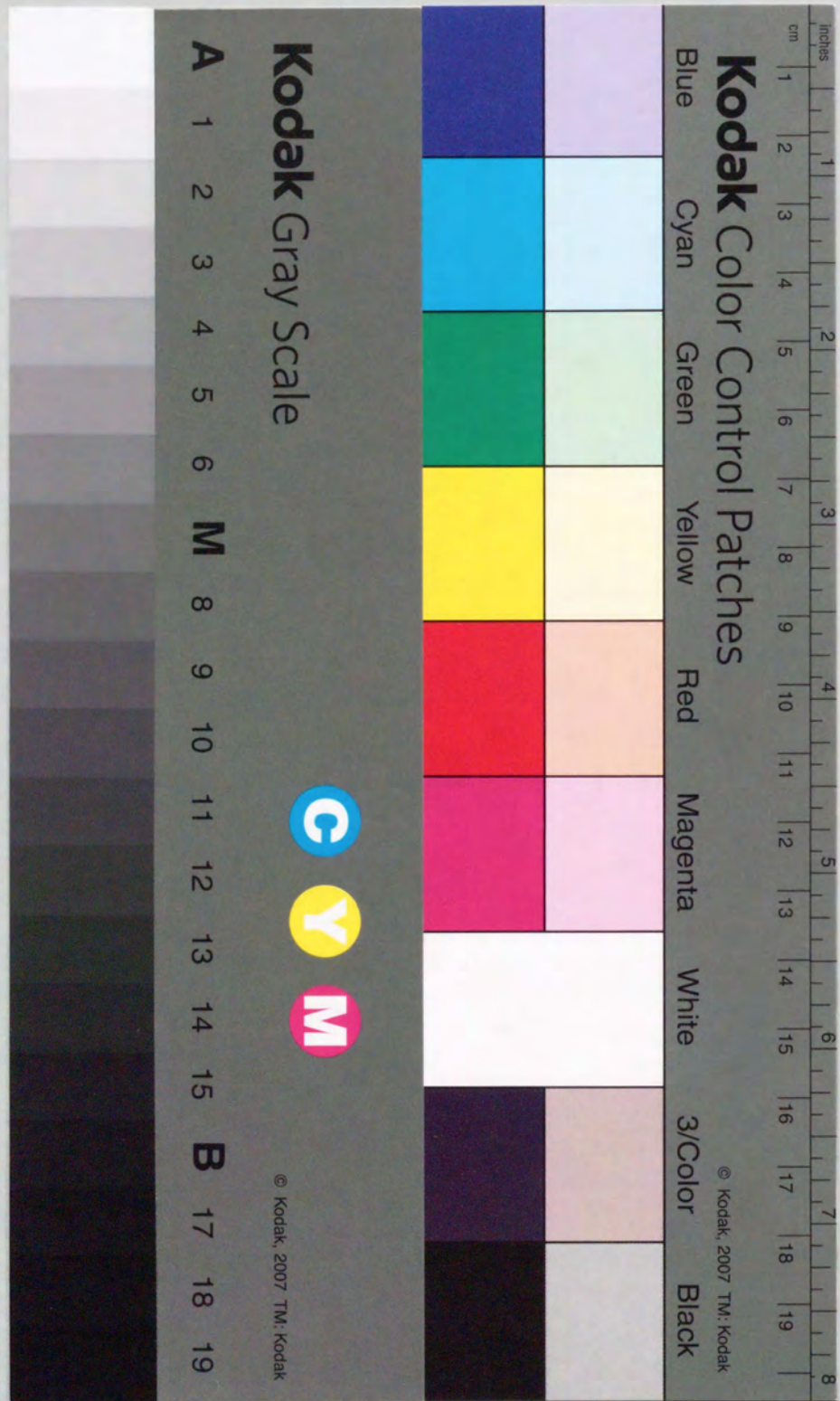
動物体を扱うロボットの運動計画と制御

Robot Motion Planning and Control
for Moving Objects

阪 口 健

大阪大学大学院 基礎工学研究科

1993 年 1 月



①

i

動物体を扱うロボットの運動計画と制御

**Robot Motion Planning and Control
for Moving Objects**

阪 口 健

大阪大学大学院 基礎工学研究科

1993 年 1 月

Abstract

This thesis describes a robot motion planning in a very dynamic environment, that is, a trajectory generation to attain a working purpose with redundancy free to use under the restriction of time and space. We examine the relation between this redundancy and the dynamic dexterity of robots. Especially we focus on "Juggling" and "Cup & Ball" tasks as the typical examples of such a working purpose(task). To deal with moving objects (balls) in the gravitational field, the motion of the robot is determined considering the dynamic property of balls. We will describe that a parametric representation of the robot motion makes it easy to generate an appropriate trajectory of the end-effector for "juggling" task and "cup & ball" task. Also we will describe that a hierarchical learning control strategy is effective to generate a desired motion pattern for juggling which is not given beforehand and then outline its implementation to a real robot. To juggle several balls repeatedly by a real robot we predict the location and time at which a ball is caught by using sensory feedback signals and modify the end-tip trajectory (We call this scheme a "reactive planning"). In "cup & ball" task, no a priori knowledge of the ball (mass & length of a string) is assumed to be known beforehand. To perform this task, we propose how to generate the motion pattern of the manipulator with a force sensing device attached to the end-tip. Experimental results obtained using 2 d.o.f. direct drive manipulator are also shown.

目次

1 序論	1
1.1 本研究の背景	1
1.2 従来の研究状況	2
1.3 本研究の立場	4
1.4 本論文の構成	5
2 ジャグリングロボット	9
2.1 はじめに	9
2.2 手先の計画軌道	10
2.2.1 楕円軌道の採用	10
2.2.2 楕円の表現法	11
2.2.3 人間の運動パターンの解析	11
2.2.4 手先軌道と弾道軌道	14
2.2.5 ジャグリング実現のための拘束条件	14
2.2.6 軌道パラメータの導出	17
2.2.7 ボールの反発係数を考慮しない場合	19
2.2.8 媒介変数 θ_t に関する拘束条件	19
2.2.9 ジャグリングにおけるパラメータ獲得	21
2.3 軌道の計画結果	21
2.4 実験的検討	29
2.4.1 システム構成	29
2.4.2 階層学習制御則	32
2.4.3 ビジュアルフィードバックによる手先軌道の微小修正	40
2.5 実験例	46
2.6 おわりに	52

3 剣玉ロボット	53
3.1 はじめに	53
3.2 軌道計画	54
3.2.1 振子運動のモデル	55
3.2.2 投げ上げから捕獲までの手先の計画軌道	56
3.2.3 手先軌道パラメータの決定方法	58
3.2.4 空間パラメータの決定	58
3.2.5 時間パラメータの決定	62
3.2.6 全パラメータの決定	63
3.2.7 剣玉におけるパラメータ獲得	63
3.3 軌道の計画結果	64
3.4 実験的検討	68
3.4.1 システム構成	68
3.4.2 物理パラメータの推定	70
3.4.3 振幅制御のアルゴリズム	71
3.4.4 振幅の推定	77
3.4.5 計画軌道と実軌道の mismatch 解消アルゴリズム	78
3.5 実験例	81
3.6 おわりに	84
4 結論	85
附 章	
A ロボットへの指令値	87
A.1 平行リンク型 2 自由度 DD マニピュレータの逆運動学	87
A.2 学習制御則	89
B 振子運動中のボールの運動方程式	91
C システムの仕様	93
C.1 ダイレクト・ドライブ・モータの仕様	93
C.1.1 サーボドライバの制御モード	93
C.2 ダイレクト・ドライブ・ロボットアームの仕様	94

C.3 力覚センサの仕様	96
謝辞	97
参考文献	99
関連論文	105

目次

2.1	Illustration of a parameter θ	11
2.2	Plot of the end-point positions of human juggling	12
2.3	Analysis of human juggling — a displacement of $(\theta, \dot{\theta})$	13
2.4	Elliptic trajectory of the end-effector	15
2.5	Example of a parameter θ_t	20
2.6	Flowchart of acquiring juggling parameters	22
2.7	Juggling trajectories (varying T , fixing $n = 2, k = 1.0, \theta_0 =$ $195[\text{deg}]$)	24
2.8	Juggling trajectories (varying k , fixing $n = 2, T = 0.6[\text{s}], \theta_0 =$ $195[\text{deg}]$)	25
2.9	Juggling trajectories (varying θ_0 , fixing $n = 2, T = 0.6[\text{s}], k = 1$) .	26
2.10	Juggling trajectories (varying n , fixing $T = 0.6[\text{s}], k = 1, \theta_0 =$ $195[\text{deg}]$)	27
2.11	Acceleration of the end-effector in the direction of a vertical line	28
2.12	System block diagram	29
2.13	Experimental apparatus	30
2.14	Position of the image area	31
2.15	Picture of DD robot arm	31
2.16	Hierarchical learning control algorithm	33
2.17	Pictures showing the hierarchical learning — part 1	36
2.18	Pictures showing the hierarchical learning — part 2	37
2.19	Performance in the hierarchical learning control	38
2.20	Endtip trajectory after learning	38
2.21	Specified time parameter θ_t	39
2.22	Conceptual illustration of “visual feedback”	40
2.23	Experimental result during one juggling cycle	42
2.24	Pictures showing one-ball juggling — part 1	43

2.25	Pictures showing one-ball juggling — part 2	44
2.26	Pictures showing one-ball juggling — part 3	45
2.27	Pictures showing two-balls Juggling — part 1	48
2.28	Pictures showing two-balls Juggling — part 2	49
2.29	Pictures showing two-balls Juggling — part 3	50
2.30	Pictures showing two-balls Juggling — part 4	51
3.1	“Cup & Ball” task	54
3.2	Forces acting on a ball in pendulum motion	56
3.3	Parameters of an elliptic trajectory	57
3.4	Ball motion during “cup & ball” task	59
3.5	Algorithm for acquiring space parameters	61
3.6	Algorithm for acquiring time parameter	63
3.7	Flowchart of acquiring cup & ball parameters	64
3.8	Process acquiring a trajectory (trajectory 1 on Table 1)	66
3.9	Process acquiring a trajectory (trajectory 2 on Table 1)	66
3.10	Process acquiring a trajectory (trajectory 3 on Table 1)	67
3.11	Process acquiring a trajectory (trajectory 4 on Table 1)	67
3.12	Schematic diagram of the system	68
3.13	Experimental apparatus	69
3.14	Picture of DD robot arm with mechanism to keep a cup horizontally	69
3.15	Swing control in pendulum motion	72
3.16	Specified displacement x_t	73
3.17	Force signals in pendulum motion	74
3.18	Tension in pendulum motion (increasing its swing)	75
3.19	Tension in pendulum motion (decreasing its swing)	75
3.20	Displacement x_t in swing control	76
3.21	Displacement y_t in swing control	76
3.22	Situations after pitching a ball	78
3.23	Force signal after pitching in case (a)	79
3.24	Force signal after pitching in case (b)	80
3.25	Force signal after pitching in case (c)	80
3.26	Pictures showing “cup & ball” — part 1	82

3.27	Pictures showing “cup & ball” — part 2	83
A.1	Model of DD robot arm for inverse kinematics	87
A.2	Block diagram of Learning control	90
C.1	Direct-drive robot arm	95
C.2	Position of a force sensor	96

表目次

3.1 Resolved ellipse parameters	64
3.2 Estimation for the mass of a ball	70
3.3 Estimation for the length of the thread	70
3.4 Estimation for the amplitude of the pendulum motion	77
C.1 Standard specifications of “Dynaserv(DMB1030)”	93
C.2 Each link’s parameter of DD robot	95
C.3 Standard specifications of a force sensor “KBL-301L”	96

第1章 序論

1.1 本研究の背景

ロボットのように自由度の多い機械システムでは、目的のタスクを実現する運動パターンをいかに生成するかが最も基本的な問題となる。これは通常、運動計画と呼ばれる静的な問題と、それに対して動的なサーボ問題(ロボットの運動制御問題)に分けて扱われる。従来、両者の問題はそれぞれ独立に考察されることが多かったが、近年タクトタイムを短縮するニーズの高まりと共に、互いに切り離して考えられなくなっている。例えば、スピードが変化するベルトコンベア上の物体をマニピュレータによって取り上げるタスクでは、視覚センサを用いて運動パターンを逐次修正する必要がある、全体が一つのサーボ問題になる。つまり、先のようなアプローチは、機械システムのおかれた環境自体が変化する場合(対象とする物体が動いているような場合)には適用し難い。

センサ情報と機械システムの運動をタスク内容に応じて有機的に対応付ける問題は、センサ・ベース制御問題として近年注目を浴びている [1]。中でもニューラル・ネットワークによるアプローチは、生体機能との類似性のみならず、その自己学習能力と並列処理化が容易であることから、極めて有望視されている [2]。しかし、ネットワークの設計指針を明確に与えることができない現段階では、複雑な特性を有する全システムをブラックボックス化して捉えることには、かなりの無理がある。むしろ、タスクを実現するために必要な以下の機能

1. 先見的知識や学習制御ストラテジを用いた理想運動パターンの獲得
2. センサ情報の前処理と環境変化の予測
3. 状況に応じた理想運動パターンの修正

を個々に具現化した後、それらを統合するアプローチが望ましい。

本研究では、これらの機能を駆使して実現される人間の器用なタスクとして「ジャグリング」や「剣玉」に注目する。地球の重力場において自由落下する対象物との相互作用を含むタスク、つまり、投げや捕獲といった一連の行動を必要とするタスクがジャグリングや剣玉である。これらのタスクでは、対象物の動特性をも考慮に入れてロボットの運動を決定しなければならず、軌道のプランニングが重要な問題となる。特に、ジャグリングでは、対象物の個数の増加と共にその実現は難易度を増す。これは、次に受け取るべき対象物がアームの可動範囲外に出る前に、投げや捕獲といった一連のジャグリングタスクを行わなければならない、それに要する時間も個数の増加と共に短くなるからである。その他、これらのタスクは、その上達が可視であること、センサ処理を含めたトータルなシステムの設計が必要であること、作業に対する自由度が大きいこと、などの特色を持った興味深いタスクである。

1.2 従来の研究状況

運動計画と運動制御の問題を互いに切り離さず、一つの問題として取り扱う研究が幾つか行われている。

Doubowsky や Shin らは、ロボットダイナミクスやアクチュエータの出力限界を考慮して時間最適な運動パターンを生成する方法を提案している [3],[4]。また、尾崎らは、ロボットの作業環境が動的に変化する場合に、さらに作業環境の動特性まで考慮に入れた運動計画を提案した [5]。Andersson がピンポンロボット用に開発した軌道生成法は、そのような問題を考慮した典型例と言える [9]。

また、人工知能の分野においても、従来のようにプランの生成器と実行器を切り離さずに、両者が一体化されたモジュールを基本単位とする構成に変わりつつある [10]。従来のプランニングでは、「アルゴリズムにより作成されたプランは、実行時にも最後まで正しく、しかもこれらはプラン通りに実行される。」と言う前提が暗黙の了解として存在していたが、これでは現実の環境変化に即応できないことから、時空間的な拘束条件の下で、限定された現実の状況に対応しながら、少数の動作選択肢の中から最適な動作を選択・実行していく **Reactive planning** の考え方が提案されている [11]。Agre らは、知覚から動作に移るまでの時間を短縮するために、センサより得た情報を索引として、直接的にある動作を呼び出す簡単なルーチンを用意し、環境変化や予期せぬ事態に対する即応性を向上させている [12]。

同様なアプローチが必要な「ジャグリング」に関して、複数の物体の投げ上げを繰り返す通常のジャグリング(お手玉)とは異なり、ボールをパドルで上方に連続して打つタスクが、既に幾人かの研究者によって議論されている。これらは、ボールのパドルに対する反発特性を利用したもので、制御概念を全く用いない Shannon [13] の興味深い結果も含んでいる。

Koditschek らは、このタスクに対して、ミラーアルゴリズムと呼ぶ一種の巧妙な非線形フィードバック則を提案した [15]。このアルゴリズムは、ロボットと環境との断続的な相互作用を含むタスクを離散的な動的システムとして表現することにより導かれている。実際に彼らは、1 自由度のシーソー型パドルを用い、傾斜した平面内で 2 個のバック(アイスホッケーで用いられるボール)のジャグリングを実現している。但し、ロボットはリアルタイムでバックの位置をモニタできることを前提としている。一方 Atkeson らは、練習によってそのタスクを学習するアプローチを採り、タスクレベルラーニングを提唱した [19]。この学習則は、練習中のタスクレベルでの実行誤差のモデル構築と、そのモデルを利用してタスクレベルでの指令値を洗練することから成り立っている。実際に彼らは、3 自由度のロボットに大きな卓球用ラケットを付けて、1 個のテニスボールを長時間ジャグリングさせることに成功している。

剣玉のような玩具の操作について考察した研究も過去に幾つか行われている。

黒須は、人間によるヨーヨー操作を取り上げ、ヨーヨーのダイナミクスを解析した後、人間の手先の上下パターンが最小時間問題の解と定性的に一致していることを示している [22]。竹中らは、ロボット化を念頭において剣玉を取り上げ、非線形補償によるロボットの軌道追従制御や視覚フィードバックを工夫して、実際にロボットによる剣玉を実現している [23]。但し、ロボットの手先軌道は、人間の操作パターン分析を基に関節の可動範囲やアクチュエータの出力限界等を考慮して計画しておき、それをサーボ系によって実現する方法を採っている。つまり、作業環境との相互作用を考慮して系統的に軌道計画されたものではない。その他、田島らによって 1 自由度ロボットによるコマ回しの軌道計画問題を扱った興味深い研究も行われている [24]。

1.3 本研究の立場

我々は、「ジャグリングタスク」や「剣玉タスク」は動物体を扱う上で器用さを必要とするタスクの典型例であると考えている。これらのタスクは、作業環境である飛翔物体の「投げ」と「捕獲」の基本動作から成り立っていることに注目すると、それ程本質的な違いはない。

従来の研究状況を見てもわかるように、投げと捕獲を同時に取り上げた研究は少ない。我々は、これらを実行する上で問題となる点を明らかにし、その解決を通してロボットによる実現の方法を導くと共に、人間がジャグリングや剣玉を実行する際に必要となるスキルの実体に取りたい。

動物体を扱うタスクにおける問題点

動物体を扱うタスクの代表的なものには、「投げる」・「打つ」・「かわす」・「捕る」等が挙げられるが[31]、対象とする物体の動きを見て、行動をおこすという点ではどれも変わりがない。そこで、本論文では「投げ」と「捕獲」に的を絞って考察することにする。

ロボットによる実現を念頭に、まずこれらの基本動作を整理し、その実現を計る場合の問題点を示す。

投げ

投げと捕獲の複合したタスクを滑らか(連続的)に実現するためには、対象物体の弾道軌道が毎回ほとんど同じ軌道を辿ることが望ましく、そうすれば次の捕獲が行ない易くなる。即ち、投げのタスクは、仮想的な標的を狙う能力によって実現される。

ロボットは人間と違い再現性はよい。つまり、同じ軌道を正確に何度でも辿ることができるわけである。しかしながら、いくらロボットが思い通りに動いても、対象物のダイナミクスの影響によって、対象物が常に同じ弾道軌道を辿るとは限らない。この誤差が大きいと、ロボットは正確な捕獲位置を予測できず、その結果、捕獲失敗の確率が高くなる。このように、「捕獲」と「投げ」のタスクは密接に関連しており、しかもモデル化しにくい要素を多分に含んでいる。

また、対象物をより速くあるいはより速くへ(高く)投げようとするときのように、ロボットが行なおうとするタスクが高速性を要求すればするほど、ロボット

自身のダイナミクスも無視できないものとなってくることも忘れてはならない。

捕獲

対象物体を人間が捕獲する場合、まず第一に、対象物体の位置・移動の速さや方向をできる限り早く読み取らなければならない。次に、到達予定地点及び時刻を予測すると共に、その情報に基づいて捕獲のための運動パターンを決定しなければならない。特に、決定がなされる直前に対象物体と違った方向に自分自身が動いている場合には、その運動も考慮しなければならない。

捕獲タスクに関するこれらの観察から、ロボットには、捕獲タスクを実現するために以下のような能力が必要と考えられる。

1. 視覚や力覚情報による対象物体の運動状態の把握
2. 1.に基づく捕獲地点の予測
3. 2.に基づくロボットの運動パターンの決定

これらのことを飛翔物体が空中にあるわずかな間に行なわなければならないので、またハードウェアの制約の下にアルゴリズムを組まなければならないので、捕獲を実現することはそれほど容易ではない。

さらに、これら「投げ」や「捕獲」のタスクを実現するロボットの動作パターンには多くの自由度が残されており、その中から適切なものを見出すことも必要である。

1.4 本論文の構成

本論文では、以上に述べた「投げ」や「捕獲」の基本動作を滑らかに連結して行なわれるジャグリングや剣玉タスクの実現を通して、先に述べた3機能をロボットによって実現するための方法論を示すと共に、これらのタスクを人間が行なう場合に必要となるスキルについても言及する。

第2章では、ジャグリングロボットにおける運動計画と制御について述べる。ジャグリングでは、投げと捕獲を連続して交互に繰り返し行わなければならない

ので、ロボットは絶えずボールの動きに合わせた運動をしなければならない。つまり、先に述べた3機能全てを用いなければ、本タスクは実現できない。

2.1節では、ジャグリングロボットの概要について述べ、2.2節で、手先の軌道計画を行う。ロボットの手先軌道として、「投げ」と「捕獲」の動作を滑らかに連結できる手先軌道を楕円軌道とし、作業環境とロボット自身の相互作用により生じる時空間的拘束を自由に設定できるパラメータ表現を導入することによって、様々な運動パターンの手先軌道を獲得できる方法を提案する。

2.3節では、提案した方法により、複数個の対象物に対して任意の頂点高さの弾道軌道でジャグリングタスクが実現可能なことをシミュレーションにより示す。

この方法を実際に行わせるロボットシステムを2.4.1節に示す。

ロボットが先の方法によって獲得された手先軌道を実行するとき、対象物やロボットアームのダイナミクスの影響により理想通りの運動が実現されるとは限らない。そこで2.4.2節において、計画された軌道を視覚情報を用いて修正することにより、実際に理想的なボールの弾道軌道を生成する手先軌道を学習的に獲得する方法として階層学習制御則を提案し、その有効性を示す。以上は1.1節で述べた第1機能に相当する。

計画された軌道は、再現性のない外乱に対処することはできない。ジャグリングを継続的に実行させるには、この再現性のない外乱にも対応できる第2、第3の機能が必要となる。この問題を克服する一つの方法は、視覚センサを用いて予測された捕獲地点と時刻によって計画軌道の時間履歴を調節することである。2.4.3節ではこの方法に基づき、軌道を微小修正するためのビジュアルフィードバック則について述べ、その有効性を示す。

2.5節で、平面2自由度ロボットマニピュレータによるジャグリングの実験結果を示し、2.6節において、本章のまとめを行う。

第3章では、剣玉ロボットにおける運動計画と制御について述べる。ここで取り上げる剣玉タスクは、振子運動を利用してボールを投げ上げ捕獲するタスクである。剣玉タスクは1回きりの試行であるため、そのタスクの性格上、3つの機能の中で第1機能の理想運動パターンの獲得が中心的課題となる。特に、動作パターンの自由度の利用を最適性の面から議論する。

3.1節では、剣玉ロボットの概要について述べ、3.2節で、手先の軌道計画を行う。ロボットの手先軌道として、ジャグリングと同様の楕円軌道を採用し、パラメータを空間的な挙動を表現するパラメータと時間的な挙動を表現するパラメー

タに分け、前者についてはボールが目標捕獲位置を通過するように、後者についてはボールに加えられる力学的エネルギーが最も大きくなるように最適性を考慮して決定する。

3.3節では、提案した方法により、任意の目標捕獲位置に対してボールを捕獲できる様々な手先軌道が生成可能であることを示す。

3.4節では、任意の振子運動をしているボールを、3.3節の方法で決定した手先軌道によって目標捕獲地点まで誘導するために必要な諸機能について説明する。

3.4.2~3.4.4節では、手先軌道の自律的プランニングに必要な物理パラメータの推定方法を示し、計画された手先軌道の初期条件に適合するボールの運動状態を実現するための振幅制御アルゴリズムを提案する。この際、タスクを自律的に行なわせるための振幅の推定アルゴリズムも併せて示す。また、3.4.5節では、力学モデルに基づいて計画されたボールの軌道と実際のそれとの差を解消するために、ボールの捕獲の成否を力覚センサーによって判別し、計画軌道を修正する一種の学習方法を示す。

3.5節では、剣玉の紐の長さやボールの質量を未知として行った平面2自由度ロボットマニピュレータによる剣玉の実験結果を示し、3.6節において、本章のまとめを行う。

最後に第4章で、本研究の主な結果をまとめ、今後の課題等について述べる。

第2章 ジャグリングロボット

2.1 はじめに

本章では、ジャグリングについて特に

1. ロボットの手先軌道の系統的な計画
2. 計画軌道と実軌道のミスマッチの解消
3. ビジュアルフィードバックによる実軌道のリアルタイム微小修正

の問題に焦点を絞って考察する。即ち、ロボットがタスク遂行能力を身に付け、繰り返し実行する方法を提案する。

なお、ここで取り上げるジャグリングは、1本の腕で複数個のボールを順に投げて捕る一般的なパターンを考える。

タスク遂行に必要なボールの運動状態の検出は、ロボットから離れた場所に設置した視覚センサにより行う。

1. の軌道計画では、まずロボットの手先軌道として、「投げ」と「捕獲」の動作を滑らかに連結できる手先軌道を楕円軌道とし、理論および実際の観点からそれが理想的な手先軌道であることを確認する。

次に、作業環境とロボット自身の相互作用により生じる時空間的拘束を自由に設定できるパラメータ表現を手先軌道に導入することによって、様々な運動パターンの手先軌道を獲得できる方法を提案する。

この際、複数個の対象物に対して任意の頂点高さの弾道軌道でジャグリングが可能なることをシミュレーションにより示す。

なお、提案した理論式によって理想的な軌道を決定しても、実際には対象物やロボットアームのダイナミクスの影響により理想通りの運動が実現されとは限らない。

2. に関しては、計画された軌道を視覚情報を用いて修正することにより、実際に理想的なボールの弾道軌道を生成する手先軌道を学習的に獲得する方法として階層学習制御則を提案する。

以上の問題は、我々が第1章で提案する第1機能の実現に当たる。

計画された軌道は学習制御則を用いることによって正確に達成されうるので、サーボ問題を陽に考慮する必要はない。また、再現性のある外乱をも考慮した軌道が学習される。但し、再現性のない外乱に対処することはできない。ジャグリングを継続的に実行させるには、モデル化が困難な再現性のない外乱にも対応できる第2、第3の機能が必要となる。これらの両機能は密接に関連しており、両者を切り離して考えることはできない。この問題を克服する一つの方法は、視覚センサを用いて予測された捕獲地点と時刻によって計画軌道の時間シーケンスを調節することである。

3. ではこの方法に基づき、ジャグリングタスクを周期的に連続して行うために必要な手先軌道の微小修正のためのビジュアルフィードバック則について述べる。

最後に、平面2自由度ロボットマニピュレータによるジャグリングの実験結果を示す。

2.2 手先の計画軌道

2.2.1 楕円軌道の採用

動物体を扱うタスクを実行するには、どのような手先運動(軌道)が適しているのだろうか?特にジャグリングのように、複数の基本運動(投げと捕獲)から成り立っているタスクなどでは、各運動をどのように結び付け、全運動をどのように構成すればよいであろうか?

人間の腕もロボットのマニピュレータも各関節の可動範囲や各アクチュエータの出力トルクの限界を持っているので、「ジャグリング」タスクを効果的に実行するためには、「投げ」と「捕獲」をできる限り滑らかに結合することが望ましい。

我々は、ロボットの手先軌道として、軌道の滑らかさ・軌道の周期性・広いクラス軌道パターンを少数のパラメータで表現できることを考慮し、楕円軌道を採用する。

2.2.2 楕円の表現法

例えば、楕円の幾何学式は、 l_a を楕円の長径、 l_b を短径としたとき、

$$\frac{x^2}{l_a^2} + \frac{y^2}{l_b^2} = 1$$

と表わすことができるが、円と同じようにして媒介変数 θ を用いることによって、次式のように書き直すことが出来る。

$$x = l_a \cos \theta, \quad y = l_b \sin \theta$$

Fig.2.1 は、楕円の媒介変数を図示したものである。

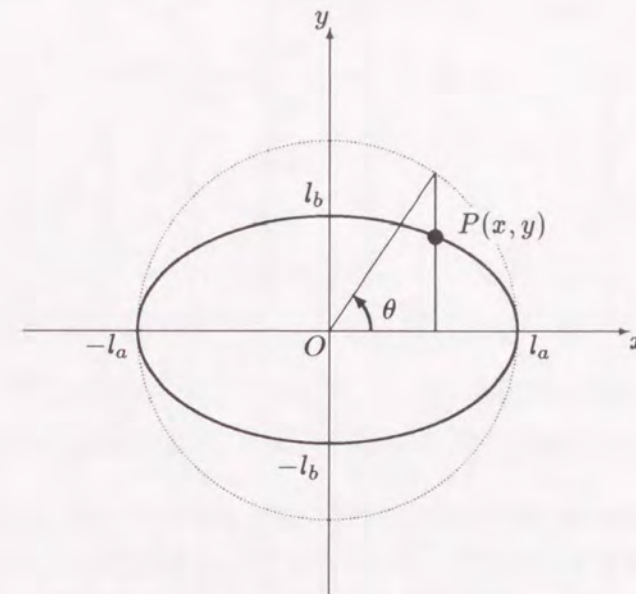


Fig.2.1: Illustration of a parameter θ

楕円が水平面に対して角度 β だけ傾いているとすると、楕円を表すパラメータは l_a, l_b, β, θ の4つになる。

2.2.3 人間の運動パターンの解析

我々は工学的な立場から楕円軌道を採用したが、その妥当性を見るために人間の動作パターンも解析し、その特徴を抽出してみた。

我々は、2人の被験者—A, B—に片手で2個のテニスボールをジャグリングするように指示した。その動きをビデオテープに記録し、彼らの手先が2次元平面内を動くものと仮定して、ビデオ画面から1コマずつ(1/30秒間隔)、彼らの手先位置をプロットしていった。Fig.2.2は、ジャグリング1周期の手先位置をプロットしたものの一例である。これらのデータから、いずれの手先軌道も楕円に近いことがわかる。また、楕円のパラメータには個人差があることも読み取れる。

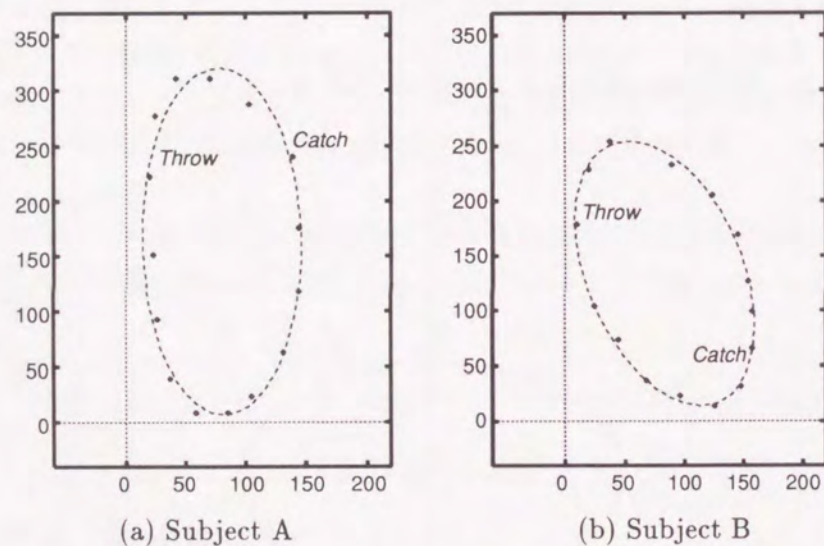
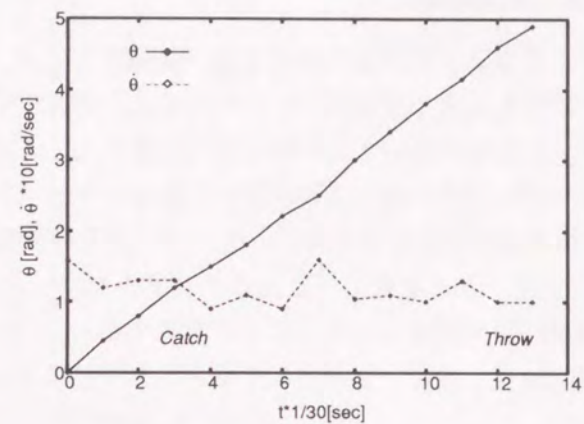


Fig.2.2: Plot of the end-point positions of human juggling

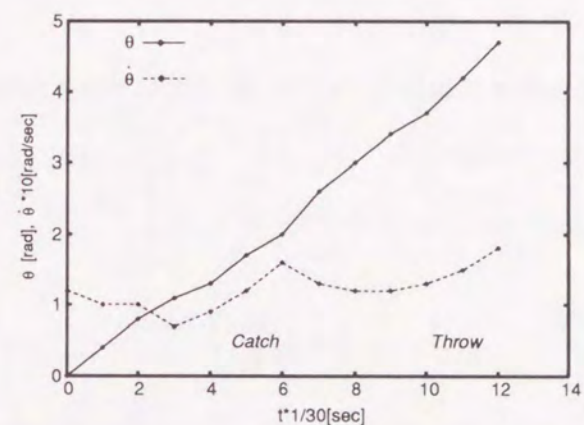
次に、軌道の時間的変化を見るために、Fig.2.2のデータから媒介変数の時間微分 $\dot{\theta}$ の時間履歴を計算した。これらのグラフをFig.2.3に示す。

これらのデータから、以下の事実がわかる。

1. 投げと捕獲のタスクを滑らかに結び付け、それらを周期的に実行するには、楕円軌道が適している。
2. 媒介変数 θ の変化は、ボールを捕獲する直前を除いてはほとんど周期的である。
3. θ は時間に関する増加関数である(ただし、1周期内)。
4. 媒介変数の時間微分 $\dot{\theta}$ は、ある一定範囲内で変化している。



(a) Subject A



(b) Subject B

Fig.2.3: Analysis of human juggling — a displacement of $(\theta, \dot{\theta})$

以上の解析結果から考えても、投げと捕獲を滑らかに途切れることなく連結し、しかも周期的に実現できる手先軌道として、楕円軌道は妥当と考えられる。

2.2.4 手先軌道と弾道軌道

Fig.2.4 に示すような x - y 座標系における楕円軌道を考える.

ジャグリングでは手先軌道と弾道軌道の2つの軌道を同時に考えなければならず, これらを区別するために時刻 t における手先位置を ${}^e\mathbf{x}_t = ({}^e x_t, {}^e y_t)^T$ と記述する. また, 手先は図に示すように初期位置から動き出すものとし, その時刻を $t=0$ として手先軌道の時間履歴を考える. ここで, 右下付き文字 t はその変数が t の時間関数であることを意味し, 左上付き文字 e は“end-effector”の頭文字を取り, その変数が手先軌道上にあることを意味する.

このとき, 手先の位置・速度・加速度はそれぞれ以下のように書ける.

$$\begin{aligned} {}^e\mathbf{x}_t &= \mathbf{R}\mathbf{A}_t\mathbf{l} = \mathbf{f}(\theta(t)) \\ {}^e\dot{\mathbf{x}}_t &= \mathbf{R}\dot{\mathbf{A}}_t\mathbf{l} \\ {}^e\ddot{\mathbf{x}}_t &= \mathbf{R}\ddot{\mathbf{A}}_t\mathbf{l} \end{aligned} \quad (2.1)$$

ここで, \mathbf{f} は媒介変数 θ の関数を意味し, \mathbf{R} は次式で表される回転行列であり

$$\mathbf{R} = \begin{pmatrix} \cos \beta & -\sin \beta \\ \sin \beta & \cos \beta \end{pmatrix}$$

\mathbf{A}_t と \mathbf{l} は

$$\mathbf{A}_t = \begin{pmatrix} \cos \theta_t & 0 \\ 0 & \sin \theta_t \end{pmatrix}, \quad \mathbf{l} = \begin{pmatrix} l_a \\ l_b \end{pmatrix}$$

である.

一方, 弾道方程式は以下のような式で書ける.

$$\begin{aligned} {}^b\mathbf{x}_t &= {}^e\mathbf{x}_{t_t} + t({}^e\dot{\mathbf{x}}_{t_t}) - \frac{1}{2}t^2\mathbf{g} \\ {}^b\dot{\mathbf{x}}_t &= {}^e\dot{\mathbf{x}}_{t_t} - t\mathbf{g} \\ {}^b\ddot{\mathbf{x}}_t &= -\mathbf{g} \end{aligned} \quad (2.2)$$

ここで, $\mathbf{g} = (0, g)^T$ は重力加速度ベクトルであり, 時刻 t はボールの投げ上げ時刻 t_t と標的とするボールの捕獲時刻 t_c の間の値をとる ($t_t \leq t \leq t_c$).

2.2.5 ジャグリング実現のための拘束条件

以下にジャグリングを実現するために必要な拘束条件を示す.

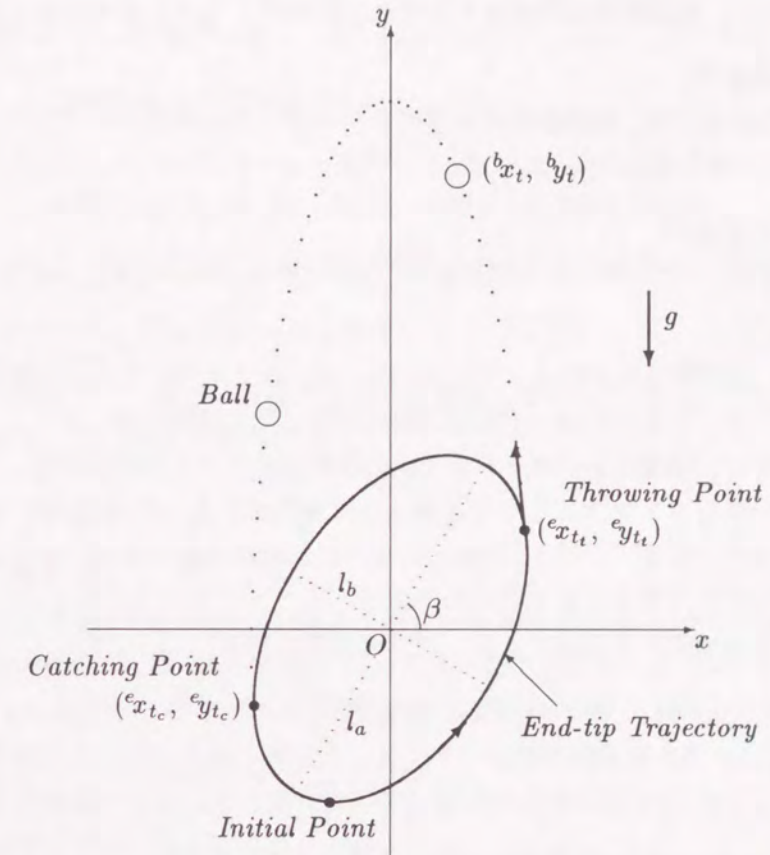


Fig.2.4: Elliptic trajectory of the end-effector

- l_a, l_b, β : length of semi-major and semi-minor axes and inclination of an ellipse
- θ_t : parameter of an ellipse $\theta_t = \theta(t)$
- ${}^b x_t, {}^b y_t$: position of a ball
- ${}^e x_t, {}^e y_t$: position of the end-effector
- t_t, t_c : time when a ball is thrown up and caught
- g : gravitational acceleration vector

1. [位置条件]

手先は、捕獲直前に標的とするボールに到達していなければならない。

2. [速度条件]

手先の速度は、捕獲直前には標的とするボールの速度とほとんど同じでなくてはならない。

3. [加速度条件]

y 方向におけるボールの加速度は、投げ上げ時において重力加速度 ($-g$) と等しい。

条件 1, 2. は捕獲条件である。特に 2. は、ボールと手先の反発を考慮した条件となっている。本研究では、手先効果器に特別な把持機構を持たせることを考えていないので、捕獲直前にボールと手先の相対速度をできるだけ小さくしておかないとボールをファンブルして受け損なう可能性がある。条件 3. は、ボールの投げ上げ条件である。これは、手先からボールに加わる抗力が零になったときに、ボールが手先から離れるとしたものである。

これらの拘束条件を考慮すると、以下の条件を得る。

$${}^e\mathbf{x}_{t_c} = {}^e\mathbf{x}_{t_t} + (t_c - t_t){}^e\dot{\mathbf{x}}_{t_t} - \frac{1}{2}(t_c - t_t)^2\mathbf{g} \quad (2.3)$$

$${}^e\dot{\mathbf{x}}_{t_c} = {}^e\dot{\mathbf{x}}_{t_t} - (t_c - t_t)\mathbf{g} \quad (2.4)$$

$${}^e\ddot{y}_{t_t} = -g \quad (2.5)$$

式 (2.3), (2.4) に式 (2.1) を代入して、次式を得る。

$$\mathbf{R}\mathbf{B}_1\mathbf{l} = -\frac{1}{2}(t_c - t_t)^2\mathbf{g} \quad (2.6)$$

$$\mathbf{R}\mathbf{B}_2\mathbf{l} = -(t_c - t_t)\mathbf{g} \quad (2.7)$$

ここで、

$$\mathbf{B}_1 = \mathbf{A}_{t_c} - \mathbf{A}_{t_t} - (t_c - t_t)\dot{\mathbf{A}}_{t_t}$$

$$\mathbf{B}_2 = \dot{\mathbf{A}}_{t_c} - \dot{\mathbf{A}}_{t_t}$$

である。 $\mathbf{B}_1, \mathbf{B}_2$ は対角行列であるので、式 (2.6), (2.7) は次のように書き直すことができる¹。

$$\mathbf{l} = -\frac{1}{2}(t_c - t_t)^2\mathbf{B}_1^{-1}\mathbf{R}^T\mathbf{g} \quad (2.8)$$

$$\mathbf{l} = -(t_c - t_t)\mathbf{B}_2^{-1}\mathbf{R}^T\mathbf{g} \quad (2.9)$$

¹回転行列の性質より $\mathbf{R}^{-1} = \mathbf{R}^T$ 。

これら 2 式より、

$$2\mathbf{B}_1 - (t_c - t_t)\mathbf{B}_2 = \mathbf{0} \quad (2.10)$$

を得る。式 (2.10) を成分で書き表せば、

$$\begin{cases} 2(\cos\theta_{t_c} - \cos\theta_{t_t}) + (t_c - t_t)(\dot{\theta}_{t_c}\sin\theta_{t_c} + \dot{\theta}_{t_t}\sin\theta_{t_t}) = 0 \\ 2(\sin\theta_{t_c} - \sin\theta_{t_t}) - (t_c - t_t)(\dot{\theta}_{t_c}\cos\theta_{t_c} + \dot{\theta}_{t_t}\cos\theta_{t_t}) = 0 \end{cases} \quad (2.11)$$

となる。これらから $(t_c - t_t)$ を消去すると、以下のような条件式を得る。

$$(1 - \cos(\theta_{t_c} - \theta_{t_t}))(\dot{\theta}_{t_c} - \dot{\theta}_{t_t}) = 0 \quad (2.12)$$

ボールの投げ上げ位置と捕獲位置は違うと仮定しているので、 $\theta_{t_c} - \theta_{t_t} \neq 0$ である。従って、以下のような条件式

$$\dot{\theta}_{t_c} = \dot{\theta}_{t_t} \quad (2.13)$$

が導き出される。

これは、手先が楕円軌道上を動くとき、もしも媒介変数 θ_t の時間微分が投げ上げ時と捕獲時で一致しなければ、ジャグリングが実現されないことを意味する。

式 (2.11) は投げ上げ時刻 t_t と捕獲時刻 t_c のみに依存する連立方程式とみなすこともできるので、これを解くことにより、拘束条件を満たす t_t, t_c を決定することができる。なおこの決定に関して、楕円形状を決定するパラメータ l_a, l_b, β には依存しないことに注意されたい。

2.2.6 軌道パラメータの導出

次に、式 (2.11) から求まる t_t, t_c を用いて楕円形状を決定するパラメータ l_a, l_b, β を表わそう。式 (2.9) に条件式 (2.13) を代入して、 $\sin\beta, \cos\beta$ について書き表わすと、

$$\begin{cases} \sin\beta = l_a p_a \\ \cos\beta = l_b p_b \end{cases} \quad (2.14)$$

$$p_a = \frac{\dot{\theta}_{t_t}(\sin\theta_{t_c} - \sin\theta_{t_t})}{g(t_c - t_t)}$$

$$p_b = \frac{\dot{\theta}_{t_t}(\cos\theta_{t_t} - \cos\theta_{t_c})}{g(t_c - t_t)}$$

となる。また、拘束条件式 (2.5) を変形すると、

$$l_a q_a \sin \beta + l_b q_b \cos \beta = 1 \quad (2.15)$$

$$q_a = \frac{\dot{\theta}_{t_i}^2 \cos \theta_{t_i} + \ddot{\theta}_{t_i} \sin \theta_{t_i}}{g}$$

$$q_b = \frac{\dot{\theta}_{t_i}^2 \sin \theta_{t_i} - \ddot{\theta}_{t_i} \cos \theta_{t_i}}{g}$$

となる。そこで、式 (2.14) を式 (2.15) に代入し、三角関数の性質を用いれば、以下のような連立方程式を得ることができる。

$$\begin{cases} p_a q_a l_a^2 + p_b q_b l_b^2 = 1 \\ p_a^2 l_a^2 + p_b^2 l_b^2 = 1 \end{cases} \quad (2.16)$$

これらの式を解くと、楕円形状のパラメータは次式のように与えられる。

$$l_a = \sqrt{\frac{p_b - q_b}{p_a(p_b q_a - p_a q_b)}} \quad (2.17)$$

$$l_b = \sqrt{\frac{q_a - p_a}{p_b(p_b q_a - p_a q_b)}} \quad (2.18)$$

$$\beta = \tan^{-1} \sqrt{\frac{p_a(p_b - q_b)}{p_b(q_a - p_a)}} \quad (2.19)$$

ここで、もし $\ddot{\theta}_{t_i} = 0$ であれば、

$$l_a = l_b = \frac{g(t_c - t_i)}{\dot{\theta}_{t_i} \sqrt{2(1 - \cos(\theta_{t_c} - \theta_{t_i}))}}$$

$$\beta = \tan^{-1} \frac{\sin \theta_{t_c} - \sin \theta_{t_i}}{\cos \theta_{t_c} - \cos \theta_{t_i}}$$

となる。つまり、楕円形状を決定するパラメータ l_a, l_b, β がそれぞれ投げ上げ時刻 t_i 、捕獲時刻 t_c 、媒介変数 θ_i によって表わされたことになる。しかしながら、この θ_i に未だ自由度が残っている。

2.2.7 ボールの反発係数を考慮しない場合

2.2.5 節の拘束条件において、条件 2. は必ずしも必要ではない。例えば、反発係数が無いに等しい物体でジャグリングを行なう場合や、把持機能を持った手先効果器を用いてジャグリングを行なう場合、捕獲位置さえ合っていればジャグリングは遂行できる。しかしながら、捕獲位置で手先軌道と弾道軌道が交わる形を取るよりも、接する形になった方がより確実である。

そこで、条件 2. の速度ベクトルが一致するという条件を、速度の方向が一致する条件に緩和する。すなわち、式 (2.4) を次式のように置き換える。

$${}^c \dot{x}_{t_c} = k {}^b \dot{x}_{t_c} \quad (k > 0)$$

これを成分で表わせれば、次のようになる。

$$\frac{{}^c \dot{y}_{t_c}}{{}^c \dot{x}_{t_c}} = \frac{{}^b \dot{y}_{t_c}}{{}^b \dot{x}_{t_c}} = \frac{{}^c \dot{y}_{t_i} - g(t_c - t_i)}{{}^c \dot{x}_{t_i}} \quad (2.20)$$

これと式 (2.3) に式 (2.1) を代入して、式 (2.11) に代わる方程式

$$2(1 - \cos(\theta_{t_c} - \theta_{t_i})) - (t_c - t_i) \dot{\theta}_{t_i} \sin(\theta_{t_c} - \theta_{t_i}) = 0 \quad (2.21)$$

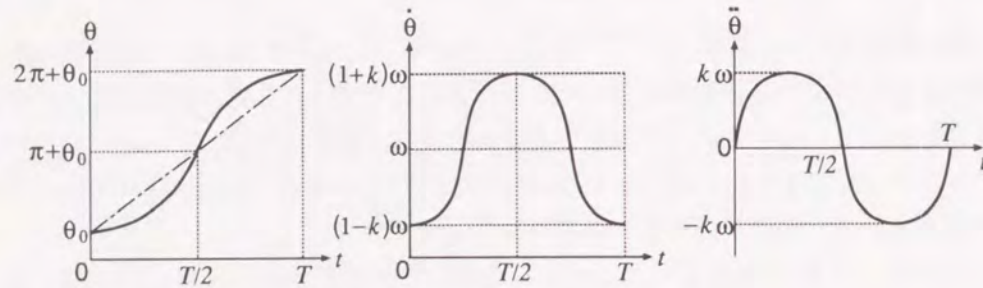
が求まる。

この場合、拘束式が 1 つ少なくなるので、上式に投げ上げ時刻 t_i か捕獲時刻 t_c のどちらか一方を予め与えてやる必要が出てくる。これにより求めた時刻を用いて残りのパラメータ l_a, l_b, β を決定する。決定に際しては、式 (2.5), (2.8) から前節と同様にして求めることができる。

2.2.8 媒介変数 θ_i に関する拘束条件

先に述べたように、ロボット手先の運動パターンは、媒介変数 θ_i の関数として決定される。残っている問題は、この θ_i の与え方である。ここでは、以下の条件によって特定化されたあるクラスの関数に θ_i を限定する。

1. θ_i は周期関数である。
2. θ_i は時間に関する増加関数である (ただし、1 周期内)。
3. θ_i は C^3 級である。

Fig.2.5: Example of a parameter θ_t

1. は θ が $0 \sim 2\pi$ の周期的な値を取らなければジャグリングのような連続した楕円軌道を描けないことから、2. は手先が逆行しないように、3. は加速度の急激な変化によってロボットアームに掛かる負担をできる限り押さえるようにするために必要な条件となっている。これらの条件は、人間のジャグリングの観察結果とも合致する。

さらに、幾つものボールを用いたジャグリングに対する手先軌道を簡単に表すために、次のような媒介変数 θ_t を採用する (Fig.2.5)。

$$\theta_t = \omega t - k \sin \omega t + \theta_0 \quad (2.22)$$

ここで、 ω は T をジャグリングの周期としたとき

$$\omega = \frac{2\pi}{T} \quad (2.23)$$

で与えられる。また、 k は正弦波の振幅であり、その範囲は条件 2 より、

$$-1 \leq k \leq 1 \quad (2.24)$$

であり、 θ_0 は θ_t の初期値である。

これらのパラメータ T, k, θ_0 を変えることにより、媒介変数の時間履歴を調整することができ、マニピュレータの手先軌道とボールの弾道軌道も調整できる。 T は弾道軌道の高さを変化させ、 k は楕円軌道の形に影響を与え、 θ_0 は弾道軌道の高さを固定しながら、手先軌道を変化させる。

先に述べた 3 条件を満たす媒介変数 θ_t であれば、式 (2.22) 以外の関数を探ることもできる。次章の剣玉ロボットでは、使用するロボットアーム (モータの出

カトルク) を考慮して、ボールの投げ上げ速度が最大になるようにその関数を決定している。このような最適化のアプローチについては、次章で詳しく述べる。

2.2.9 ジャグリングにおけるパラメータ獲得

Fig.2.6 は、ジャグリングにおける全パラメータを計算するための手順を示したものである。

予め与えるのは、ボールの個数 n と媒介変数 θ_t の形、及び Newton 法における投げ上げ時刻、捕獲時刻の初期値 (t_{t0}, t_{c0}) である。ここで、 t_{t0} は周期 T に対して $0 < t_{t0} < T$ の範囲内で、 t_{c0} はボールの個数により $(n-1)T < t_{c0} < nT$ の範囲内で与えなければならない。

この初期値をもとに、Newton 法を用いて式 (2.11) から投げ上げ時刻 t_t と捕獲時刻 t_c を導出する。そして、これらの値を式 (2.17)~(2.19) に代入すれば、楕円形状を決定するパラメータ (l_a, l_b, β) が算出される。

2.3 軌道の計画結果

前節で述べた方法により、様々な媒介変数 θ_t を用いてシミュレーションを行なった。本方法によって獲得された楕円軌道の幾つかを以下に示す。

Fig.2.7 に周期 T を変化させた例を示す。周期 T が大きくなるにつれて手先軌道は相似形で大きくなる。そのため、ボールの滞空時間も長くなり、弾道軌道の最高点も高くなる。

式 (2.22) の $\theta_t = \theta(t)$ は周期 T の周期関数であるから、 t_t, t_c は周期に比例する。また、式 (2.17), (2.18), (2.22) から l_a, l_b が周期の 2 乗に比例する関係が得られる。

Fig.2.8 に、 $\sin \omega t$ の係数 k を変化させた場合の例を示す。 k の変化によってボールが上がる高さはほとんど変わらないが、手先軌道の形状は大きく変化することがわかる。

Fig.2.9 に、初期角度 θ_0 を変化させた場合の例を示す。ボールの高さはほとんど変わらないが、 θ_0 の変化によって手先軌道の形状が多少変化することがわかる。

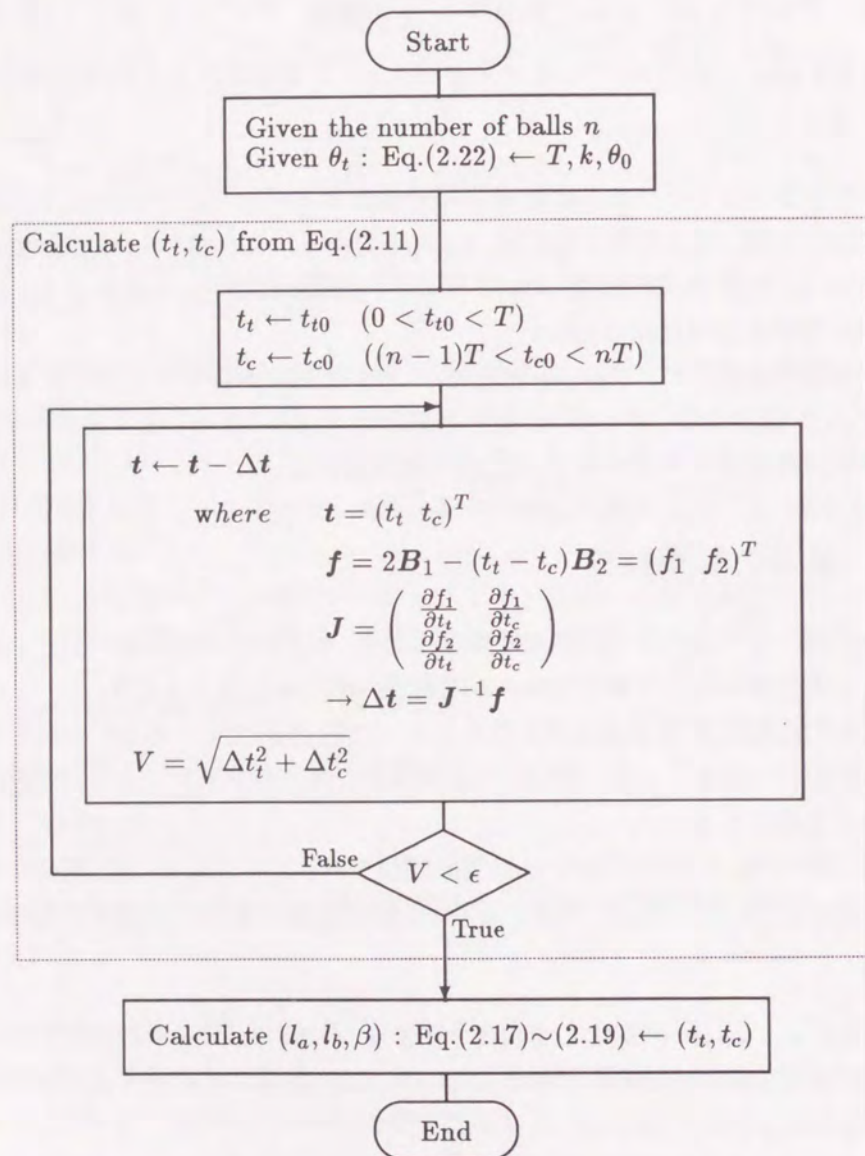


Fig.2.6: Flowchart of acquiring juggling parameters

t_t 及び t_c が θ_0 に依存しないことは、式 (2.11) から明かである。なぜならば、例えば θ_t の代わりに $\theta_t + \delta\theta$ を代入したとすると、式 (2.11) の第1式は

$$\begin{aligned}
 & 2\{\cos(\theta_{t_t} + \delta\theta) - \cos(\theta_{t_c} + \delta\theta)\} \\
 & + (t_t - t_c)\{\dot{\theta}_{t_t} \sin(\theta_{t_t} + \delta\theta) + \dot{\theta}_{t_c} \sin(\theta_{t_c} + \delta\theta)\} \\
 & = \cos \delta\theta \underbrace{\{2(\cos \theta_{t_t} - \cos \theta_{t_c}) + (t_t - t_c)(\dot{\theta}_{t_t} \sin \theta_{t_t} + \dot{\theta}_{t_c} \sin \theta_{t_c})\}}_{=0} \\
 & + \sin \delta\theta \underbrace{\{2(\sin \theta_{t_t} - \sin \theta_{t_c}) - (t_t - t_c)(\dot{\theta}_{t_t} \cos \theta_{t_t} + \dot{\theta}_{t_c} \cos \theta_{t_c})\}}_{=0} \\
 & = 0
 \end{aligned}$$

となるからである (第2式についても同様)。

これらのシミュレーション結果から、弾道軌道の頂点高さを変えるには周期 T を、手先軌道の形状を大きく変えるには k を、弾道軌道の頂点高さを一定に保ちながら手先軌道を変えるには θ_0 をそれぞれ調節すればよいことがわかる。

Fig.2.10 は、ボールの個数を変化させた場合の例である。ボールの個数は、我々が提案する方法の中では、明確な形で変数として扱ってはいない。ボールの個数に合わせた手先軌道を得るためには、ニュートン法に代入する t_c の初期値を適当に変えてやればよい。例えば、3個のボールをジャグリングする場合には、 $2T$ と $3T$ の間にその初期値を選べばよい。

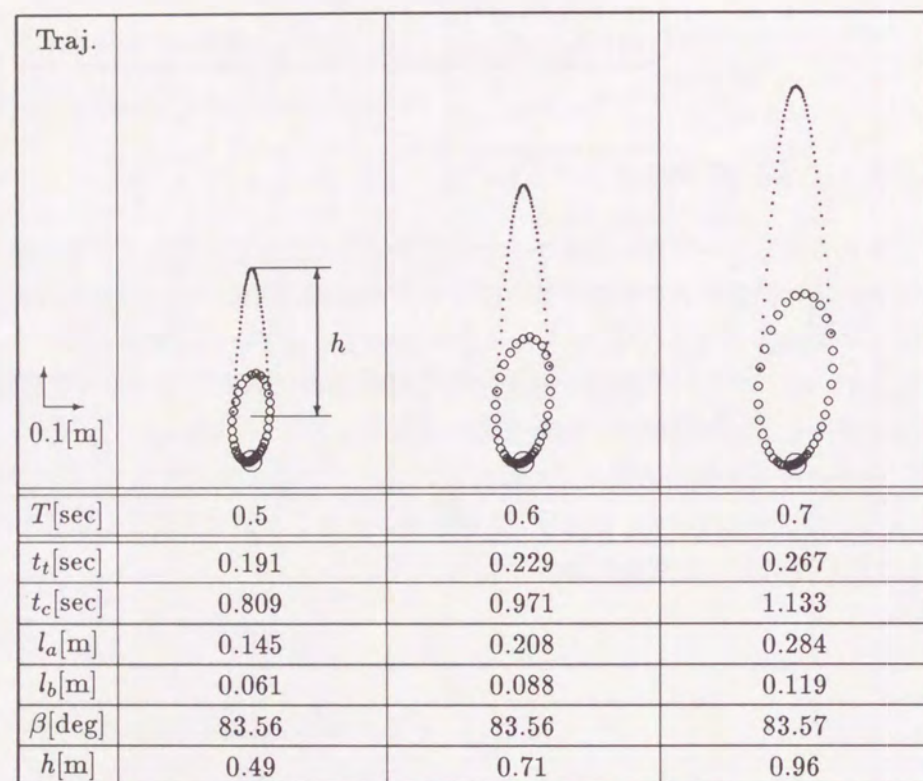


Fig.2.7: Juggling trajectories (varying T , fixing $n = 2, k = 1.0, \theta_0 = 195$ [deg])

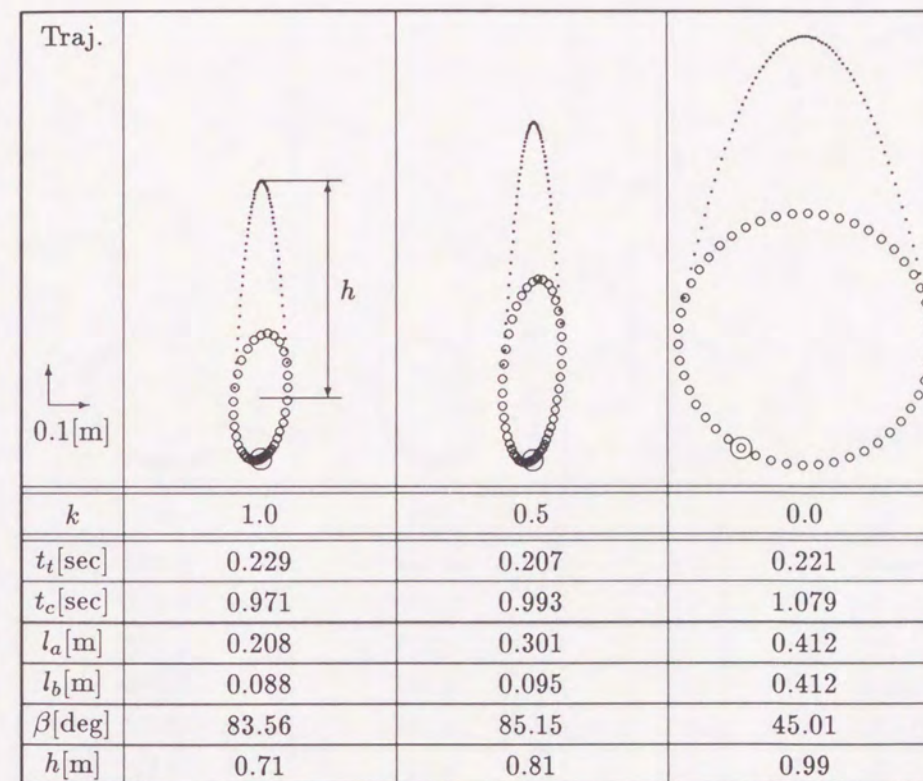


Fig.2.8: Juggling trajectories (varying k , fixing $n = 2, T = 0.6$ [s], $\theta_0 = 195$ [deg])

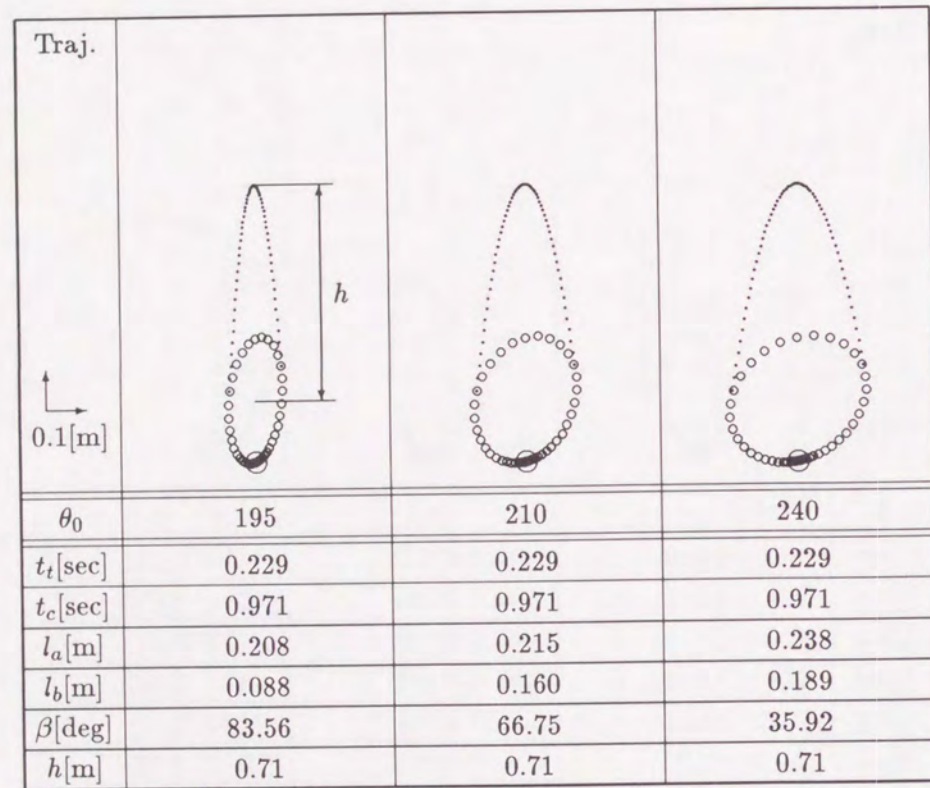


Fig.2.9: Juggling trajectories (varying θ_0 , fixing $n = 2, T = 0.6[s], k = 1$)

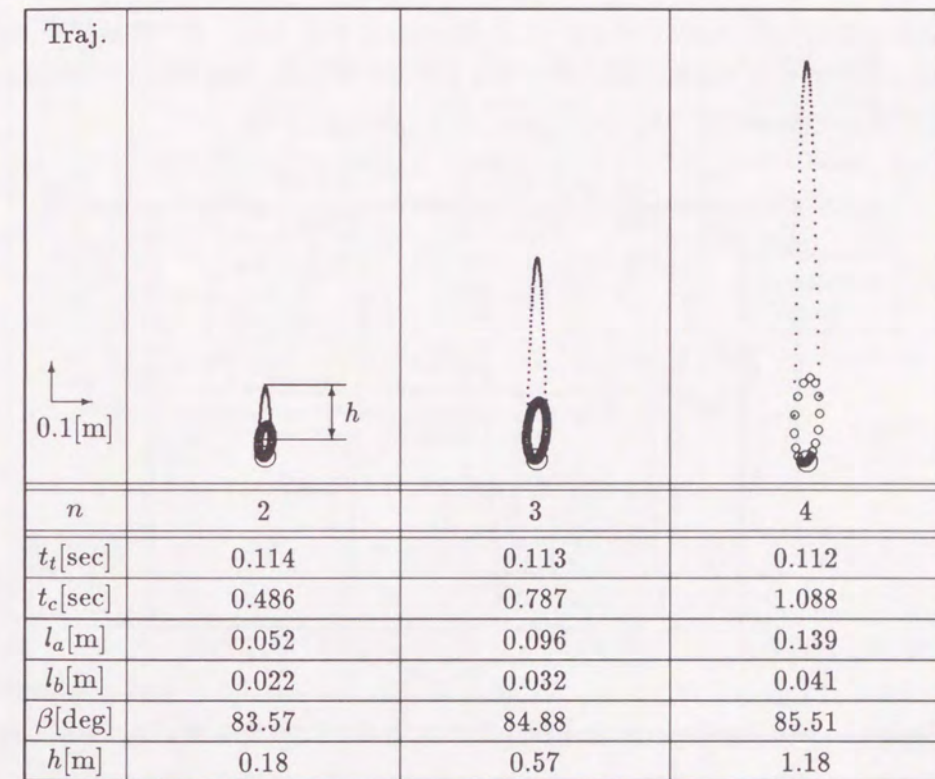


Fig.2.10: Juggling trajectories (varying n , fixing $T = 0.6[s], k = 1, \theta_0 = 195[\text{deg}]$)

今回のシミュレーションで採用した拘束条件の1つに、ボールが投げ出される瞬間の y 方向の加速度が $-g$ に等しいとした式 (2.5) があるが、厳密に言えば、

$$\begin{aligned} t < t_t &: \ddot{y}_{t_t} > -g \\ t = t_t &: \ddot{y}_{t_t} = -g \\ t > t_t &: \ddot{y}_{t_t} < -g \end{aligned} \quad (2.25)$$

でなければならない。例えば $t < t_t$ で $\ddot{y}_{t_t} < -g$ になると、ボールは投げ上げ時刻 t_t 以前に手先を離れ、宙に浮いてしまうからである。Fig.2.11 に手先の y 方向の加速度の一例を示す。

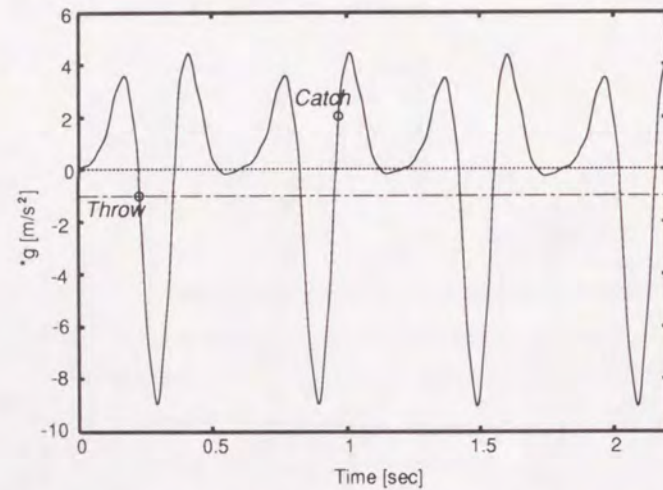


Fig.2.11: Acceleration of the end-effector in the direction of a vertical line

この図からもわかるように、 \ddot{y}_{t_t} はあらゆる時点において式 (2.25) を満たしている。なお、本節に示すシミュレーション結果はすべて、この条件を満足している。

なお、ジャグリングに関して Shannon が提案した理論 [13] によると、ボールの個数 (B)、手の数 (H)、各ボールが手に握られているときの時間 (D)、各ボールの飛行時間 (F)、それぞれの手が空の時間 (E) の間には、次の関係が成り立つ。

$$\frac{B}{H} = \frac{D+F}{D+E}$$

我々が提案した理論式によって導出された結果は全て、上式を満たしている。

2.4 実験的検討

2.4.1 システム構成

Fig.2.12は、ビジュアルフィードバックを含んだトータルなシステムの概略である。リアルタイムの信号処理のために、複数のトランスピュータ (T-800 20MHz) を用いた構成をとっている [32],[33]。

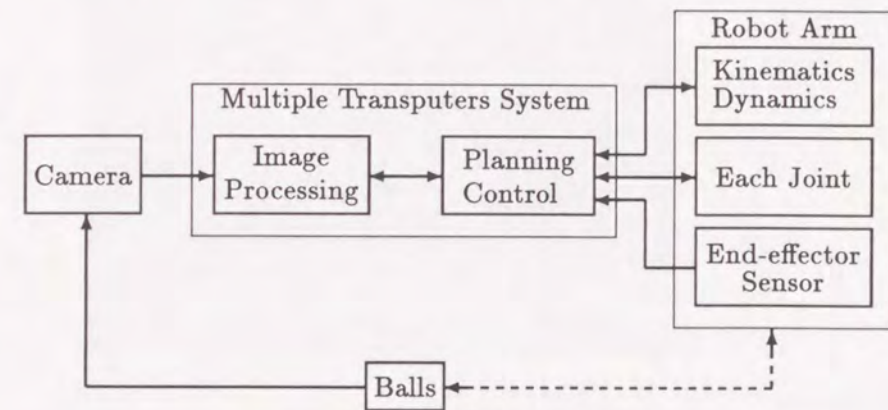


Fig.2.12: System block diagram

Fig.2.13 に本システムの全景を示す。ジャグリングを行なわせるためのマニピュレータとして、ダイレクトドライブ方式の鉛直平面2自由度平行リンクマニピュレータ (附章 C 参照) を使用する。なお、手先効果器には特別な把持機構や射出機構を持たず、漏斗状の受け皿を用いる。このような手先効果器を使用する利点は、ロボットの動力的な器用さを示せることであり、反面その欠点は、把持されないボールが受け皿の中で動き回る可能性があることである。また、画像の2値化処理 [34] を行い易くするため、ボールは白色にし、背景にはブラックボードを配置した。

また、環境認識のための視覚センサ (CCD カメラ) をマニピュレータから 4.76m 離れた場所に床から高さ 1.69m の位置に固定し、Fig.2.14 に示すようにマニピュレータの上方 0.64m×0.60m の範囲が撮像されるように設定した。この範囲のシーンが 256×240pixel の画像データに変換され、トランスピュータを用いた画像処理ボードによって処理された後、制御を司るトランスピュータに転送される。

マニピュレータへの指令は、パルス発信モジュールを介して位置制御モードで行い (附章 A.1, C 参照)、マニピュレータの関節の角変位は、DD モータ付属の

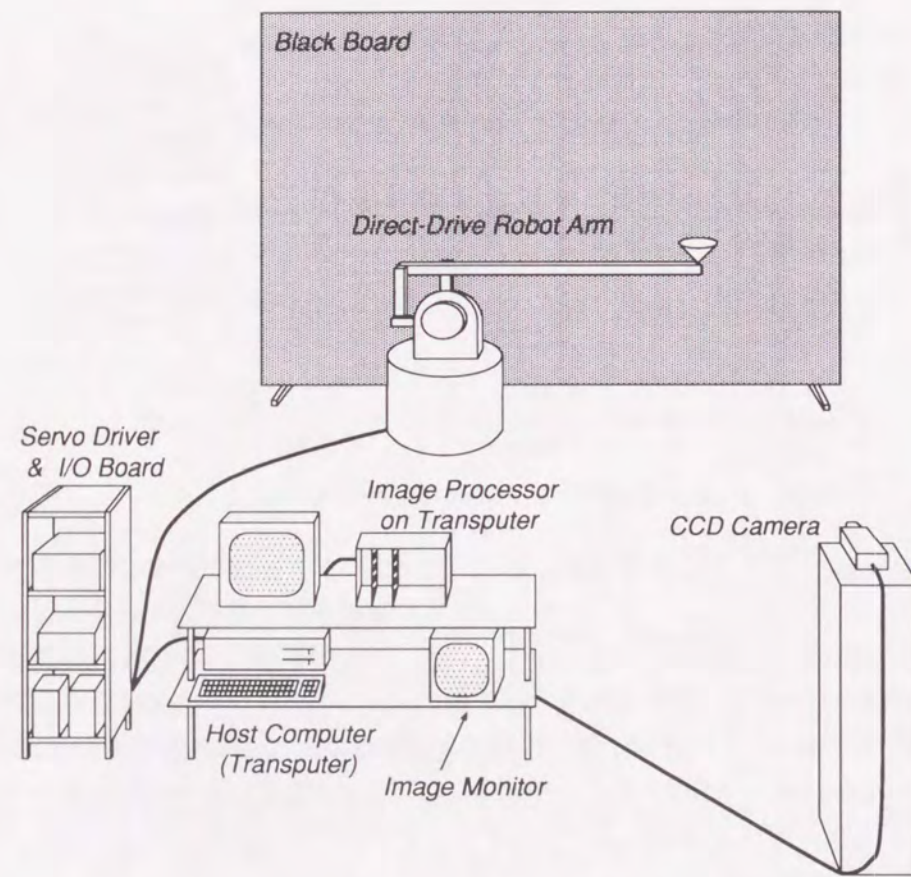


Fig.2.13: Experimental apparatus

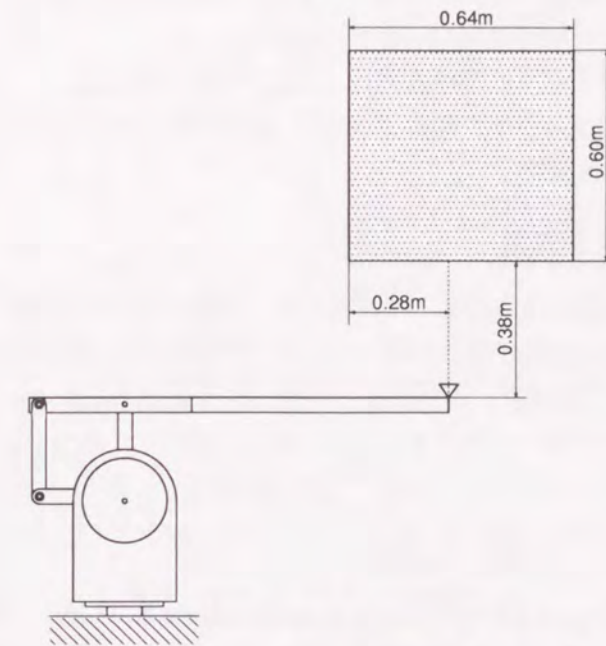


Fig.2.14: Position of the image area

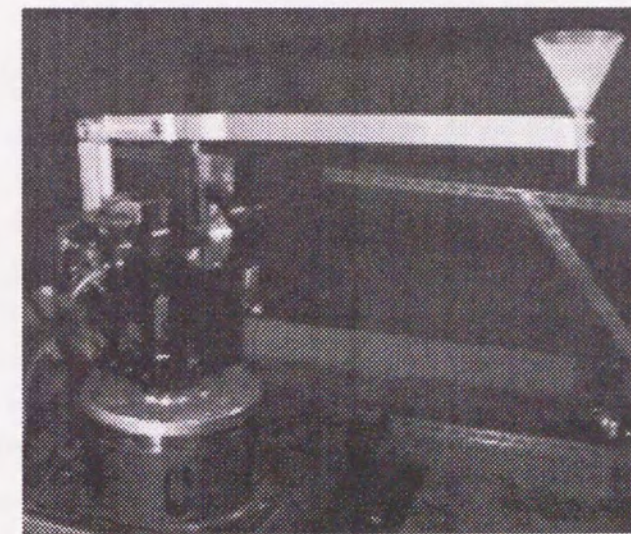


Fig.2.15: Picture of DD robot arm

エンコーダにより発生するパルス信号をカウンタモジュールでカウントして取り込む。サンプリング時間は 0.533msec である。

ボールは反発係数がほぼ零の直径約 0.05m, 質量 0.025kg の不定形物を用い, 手先効果器にはプラスチック製のカップで口の直径 0.15m, 高さ 0.13m のものを用いる (Fig.2.15参照)。

2.4.2 階層学習制御則

ここでは, 計画軌道を修正することにより, 実際に投げと捕獲を実現できる理想的なボールの弾道軌道を発生するロボットの手先軌道を学習的に獲得する方法として, Fig.2.16に示すような階層学習制御則を提案する。

弾道軌道を左右するのは投げ上げ時の速度である。この速度は, 手先の楕円軌道によって生じる。ボールの位置は, 楕円軌道式 (2.1) を用いて

$$\begin{aligned} {}^b\mathbf{x}_t &= \mathbf{R}(\mathbf{A}_{t_t} + t\dot{\mathbf{A}}_{t_t})\mathbf{l} - \frac{1}{2}t^2\mathbf{g} \\ &= \begin{pmatrix} l_a(\cos\theta_{t_t} - t\dot{\theta}_{t_t}\sin\theta_{t_t})\cos\beta - l_b(\sin\theta_{t_t} + t\dot{\theta}_{t_t}\cos\theta_{t_t})\sin\beta \\ l_a(\cos\theta_{t_t} - t\dot{\theta}_{t_t}\sin\theta_{t_t})\sin\beta + l_b(\sin\theta_{t_t} + t\dot{\theta}_{t_t}\cos\theta_{t_t})\cos\beta - \frac{t^2g}{2} \end{pmatrix} \end{aligned} \quad (2.26)$$

と表わせるので, 手先軌道を構成するパラメータ $(l_a, l_b, \beta, \theta_t)$ をそれぞれ変更することによってボールの投げ上げ速度, すなわち弾道軌道を修正することができる。

そこで, 弾道軌道の頂点を仮想的な入力と見なし, 目標投げ上げ速度を達成するための手先軌道の修正と目標弾道軌道を達成するための投げ上げ速度の修正を階層的に行うことを目的としたのが本アルゴリズムである。

具体的には, Fig.2.16 の内側のループにより, 変更された各パラメータによって再生成された楕円軌道を実現できるまで, 手先軌道を学習制御法により修正する (附章 A.2 参照)。ここで, 学習則を用いた最大の理由は, 投げと捕獲の動作を結合した軌道は加減速が激しく, 一般の制御則でその手先軌道を実現することがかなり困難であることによる。

再生成された楕円軌道が達成されたならば, 実際にジャグリングを行い, ボールの捕獲の成否を確認する。ボールの捕獲に失敗した場合, 外側のループによって目標弾道軌道を達成するようにボールの投げ上げ速度を修正する。投げ上げ速度を変更するためには, 前に述べたようにその速度を生み出すロボットの手先軌道を変更する必要がある。

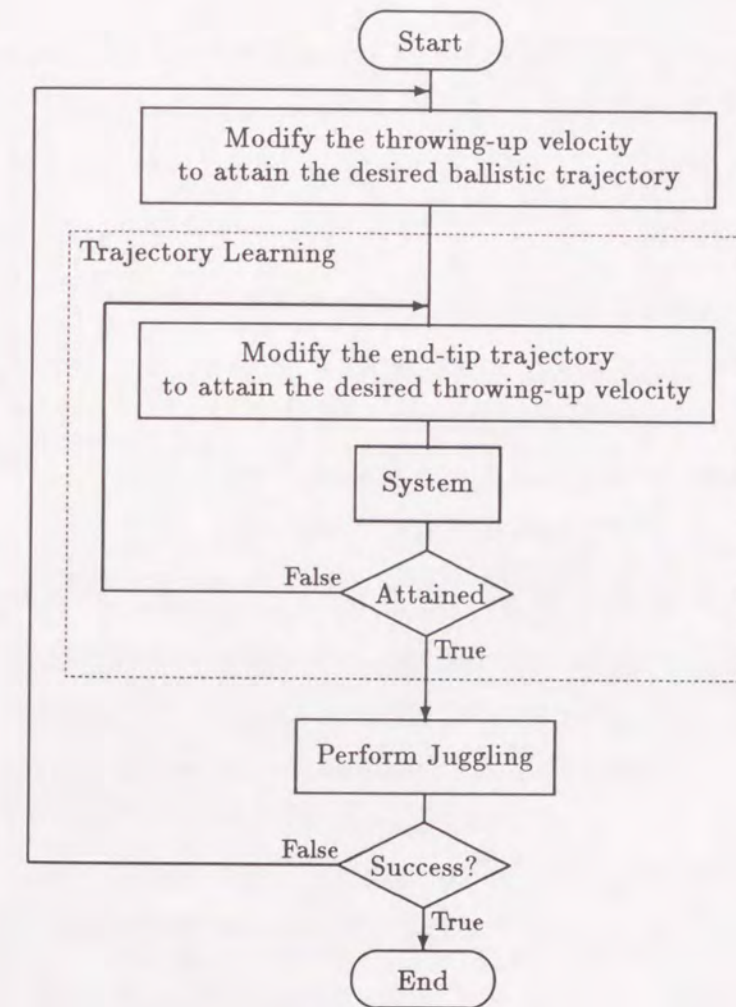


Fig.2.16: Hierarchical learning control algorithm

これは、手先の楕円軌道の各パラメータに対して次のようにして行う。画像から取り込んだボール位置より、実際の弾道軌道の最頂点 $({}^b x_{t_p}^*, {}^b y_{t_p}^*)$ を割り出し、これと弾道軌道モデルによる理論上の最頂点 $({}^b x_{t_p}, {}^b y_{t_p})$ を用いて、次の評価量 J を計算する。

$$J = w_1({}^b x_{t_p} - {}^b x_{t_p}^*)^2 + w_2({}^b y_{t_p} - {}^b y_{t_p}^*)^2 \quad (2.27)$$

ここで、 w_i ($i=1, 2$) は重み定数である。次に J を最小化するように、パラメータベクトル $\mathbf{P} = (l_a, l_b, \beta, T)^T$ の修正値 $\Delta\mathbf{P}$ を

$$\Delta\mathbf{P} = -\mathbf{K}\mathbf{P} \frac{\partial J}{\partial \mathbf{P}} \quad : \quad \mathbf{K}\mathbf{P} = (K_{l_a}, \dots, K_T) > 0 \quad (2.28)$$

より求める[36]~[38]。

式(2.26)よりボールの頂点位置は

$${}^b \mathbf{x}_{t_p} = \begin{pmatrix} l_a C_1 \cos \beta - l_b C_2 \sin \beta \\ l_a C_1 \sin \beta + l_b C_2 \cos \beta - \frac{t_p^2 g}{2} \end{pmatrix}$$

where $C_1 = \cos \theta_{t_t} - t_p \dot{\theta}_{t_t} \sin \theta_{t_t}$
 $C_2 = \sin \theta_{t_t} + t_p \dot{\theta}_{t_t} \cos \theta_{t_t}$

と表わせるので、修正値 $\Delta\mathbf{P}$ は、それぞれ以下のように書き表わせる。

$$\begin{aligned} \Delta l_a &= -2K_{l_a} \{w_1({}^b x_{t_p} - {}^b x_{t_p}^*)C_1 \cos \beta + w_2({}^b y_{t_p} - {}^b y_{t_p}^*)C_1 \sin \beta\} \\ \Delta l_b &= 2K_{l_b} \{w_1({}^b x_{t_p} - {}^b x_{t_p}^*)C_2 \sin \beta - w_2({}^b y_{t_p} - {}^b y_{t_p}^*)C_2 \cos \beta\} \\ \Delta \beta &= 2K_\beta \{w_1({}^b x_{t_p} - {}^b x_{t_p}^*)(l_a C_1 \sin \beta + l_b C_2 \cos \beta) \\ &\quad - w_2({}^b y_{t_p} - {}^b y_{t_p}^*)(l_a C_1 \cos \beta - l_b C_2 \sin \beta)\} \\ \Delta T &= -2K_T \{w_1({}^b x_{t_p} - {}^b x_{t_p}^*)(l_a C_3 \cos \beta - l_b C_4 \sin \beta) \\ &\quad + w_2({}^b y_{t_p} - {}^b y_{t_p}^*)(l_a C_3 \sin \beta + l_b C_4 \cos \beta)\} \end{aligned}$$

where $C_3 = -\sin \theta_{t_t} \frac{\partial \theta_{t_t}}{\partial T} - t_p \sin \theta_{t_t} \frac{\partial \theta_{t_t}}{\partial T} - t_p \dot{\theta}_{t_t} \cos \theta_{t_t} \frac{\partial \theta_{t_t}}{\partial T}$
 $C_4 = \cos \theta_{t_t} \frac{\partial \theta_{t_t}}{\partial T} + t_p \cos \theta_{t_t} \frac{\partial \theta_{t_t}}{\partial T} - t_p \dot{\theta}_{t_t} \sin \theta_{t_t} \frac{\partial \theta_{t_t}}{\partial T}$

ここで、媒介変数 θ_t の初期値 θ_0 は、投げ上げ時刻や捕獲時刻、並びに弾道軌道の頂点高さに影響を及ぼさないことがシミュレーションにより示されており、なおかつ、実験からもそれが明らかになったので、修正の対象としない。

このようにして式(2.28)により修正された各パラメータから楕円軌道を再生成し、内側のループに戻る。

この2つの学習ループを交互に実行することによって、最終的に目標弾道軌道を達成する手先軌道を得ることができる。

実験結果

提案したアルゴリズムでは、楕円軌道の長短径、その傾き、並びに周期の4つのパラメータが変更できる。この選定如何で本アルゴリズムの収束の仕方が変わって来る。

計画軌道で得られたパラメータ

- 2個のジャグリングの計画軌道パラメータ

$$(l_a, l_b, \beta, \theta_0, T) = (0.156[\text{m}], 0.094[\text{m}], 81.0[\text{deg}], 218.0[\text{deg}], 0.50[\text{sec}])$$

に対して、本アルゴリズムを以下のような条件

- 条件

$$\begin{aligned} \text{目標弾道頂点 } ({}^b x_{t_p}, {}^b y_{t_p}) &= (0.0, 0.67[\text{m}]) \\ (w_1, w_2) &= (0.0001, 0.00001) \\ (K_{l_a}, K_{l_b}, K_\beta, K_T) &= (50.0, 200.0, 0.0045, 0.000009) \end{aligned}$$

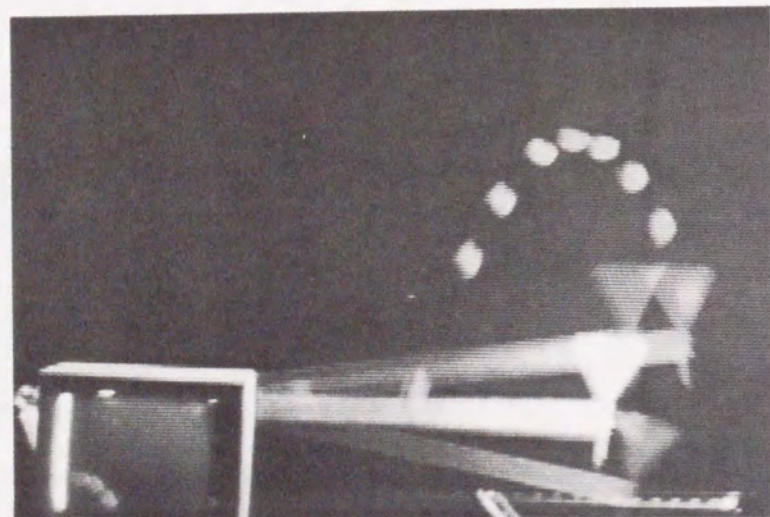
で適用した結果、手先の楕円軌道は

- 階層学習後の軌道パラメータ

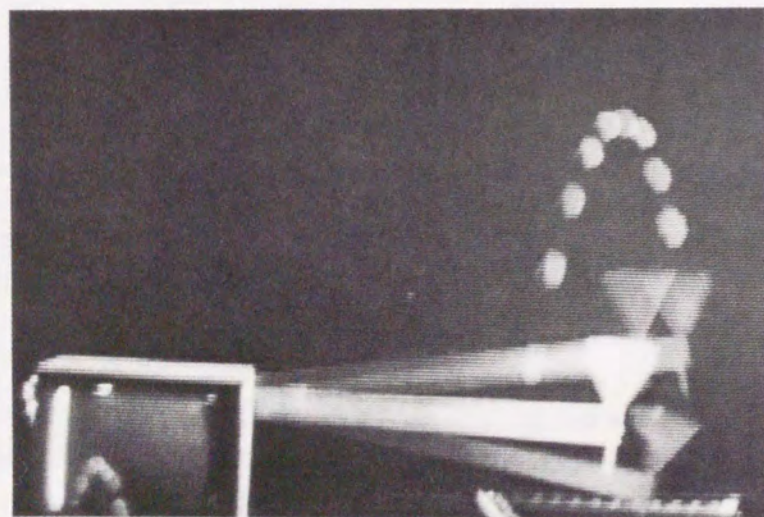
$$(l_a, l_b, \beta, T) = (0.143[\text{m}], 0.055[\text{m}], 66.5[\text{deg}], 0.38[\text{sec}])$$

に変更された。この結果から、ボールの投げ出し角度を大きく変えるには、楕円の傾きを変更することが最も有効であり、ボールの頂点高さを大きく変えるには、投げ出し速度を左右する周期を変更することが最も有効であることがわかる。

各試行毎のボールの弾動軌道の変化の様子を多重露光写真としてFig.2.17~Fig.2.18に示し、このときの評価量 J の変化の様子と最終的に得られた楕円軌道をそれぞれFig.2.19, Fig.2.20に示す。初期条件として計画軌道を与えているので、どのような手先軌道でもFig.2.19のようにほぼ10回で理想的な捕獲

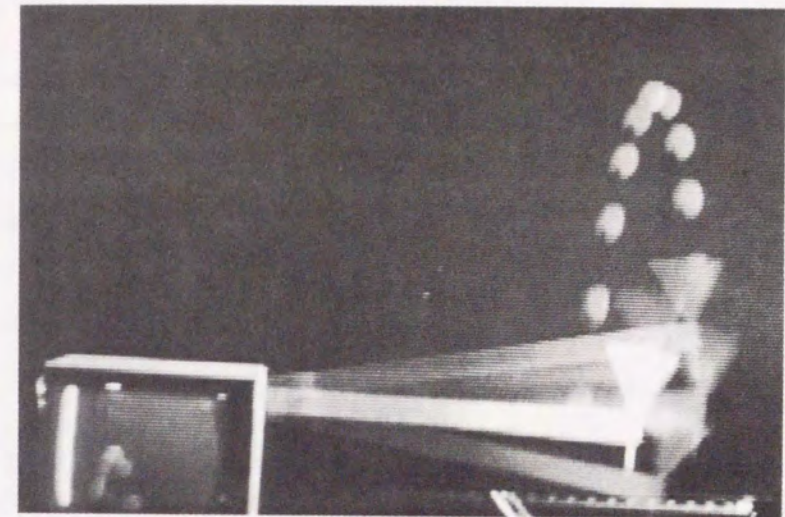


(a) Trial number 1st



(b) Trial number 2nd

Fig.2.17: Pictures showing the hierarchical learning — part 1



(c) Trial number 4th



(d) Trial number 9th

Fig.2.18: Pictures showing the hierarchical learning — part 2

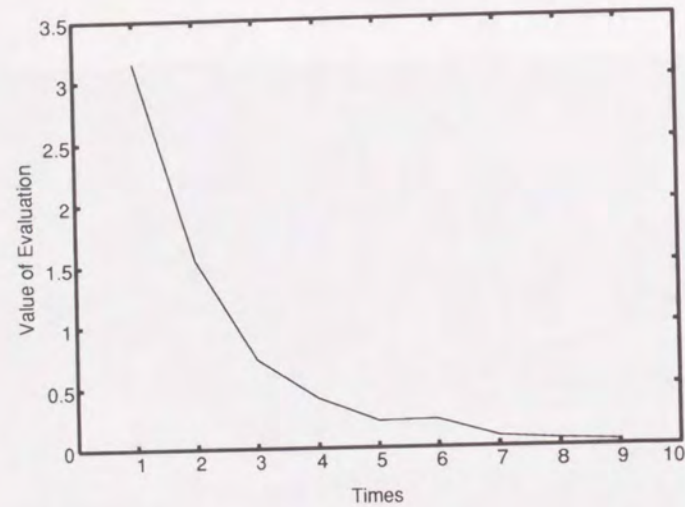


Fig. 2.19: Performance in the hierarchical learning control

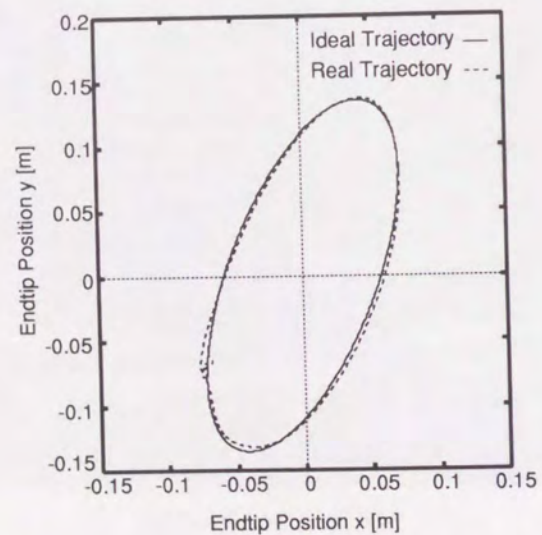


Fig. 2.20: Endtip trajectory after learning

が可能となる。なお、ボールより少し大きめのカップを手先効果器として用いているため、捕獲自体は8回前後で実現されている。また、Fig. 2.20のような手先軌道を学習するには約30回の試行回数が必要とした。

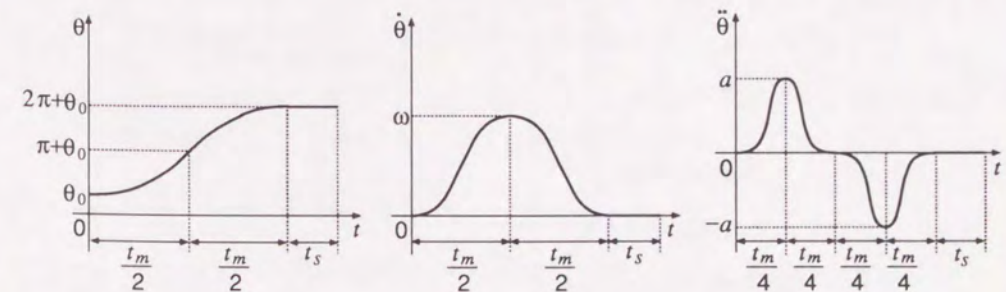
なお、附章 A.2 に示すように、軌道の学習則では初期化を前提としているので、1周の楕円軌道が終了して次の楕円軌道を再開するまでに、ロボットは初期位置に復帰しなければならない。そこで、新たに以下に示すような媒介変数を採用した (Fig. 2.21 参照)。

$$\theta_t = \begin{cases} a \left\{ \frac{1}{\omega^2} (\cos \omega t - 1) + \frac{1}{2} t^2 \right\} + \theta_0 & (0 \leq t \leq \frac{t_m}{2}) \\ -a \left\{ \frac{1}{\omega^2} (\cos \omega t - 1) + \frac{1}{2} t^2 \right\} + 2\omega t - 2\pi + \theta_0 & (\frac{t_m}{2} \leq t \leq t_m) \\ 2\pi + \theta_0 & (t_m \leq t \leq t_m + t_s) \end{cases}$$

$$\text{where } \omega = \frac{4\pi}{t_m}, \quad a = \frac{8\pi}{t_m^2}$$

この媒介変数によって得られたシミュレーション結果を初期条件として階層学習制御則を適用し、理想的な手先軌道を実験的に獲得した。

ところで、本アルゴリズムはボールのダイナミクスも含んだ学習則となっているが、ボールを特定しているために、他の質量や形などが大きく違うボールに対しても学習で獲得された手先軌道が望ましい弾道軌道を与えるとは限らない。2個以上のジャグリングをする場合、それぞれのボールの微妙な相違が、弾道軌道のズレや捕獲のタイミングのズレを引き起こす原因となると考えられる。

Fig. 2.21: Specified time parameter θ_t

2.4.3 ビジュアルフィードバックによる手先軌道の微小修正

階層学習制御則は、あくまで実現可能なジャグリング運動のフィードフォワード入力を生成するものであって、再現性のない外乱は補償できない。この種の外乱に対処するには、視覚によるボールの運動予測に基づいた手先軌道の微小修正が必要となる。

手先軌道の微小修正は、標的とするボールが空中にある間に行われなければならない。また、対象とするボールが増えるほど、それに費やすことのできる時間は短くなる。従って、ボールの軌道に合わせて捕獲のための手先軌道を再生成していたのでは、間に合わなくなる可能性が高い。しかも再生成する軌道は、すべて予め学習されていなければならない。学習則で獲得された軌道を利用して他の軌道を実現する方法も論じられているが[39]、その方法はDFTを用いて全軌道を再構成し直すため、やはり実時間レベルでの軌道生成には適用し難い。また、幾つかの手先軌道を学習してメモリに蓄えておき、ボールの軌道に応じて適宜手先軌道を変更する方法も考えられるが、どれだけ軌道を学習すればよいのかという点が明確でなく、メモリの制約もあることからあまり有効な方法とは言えない。

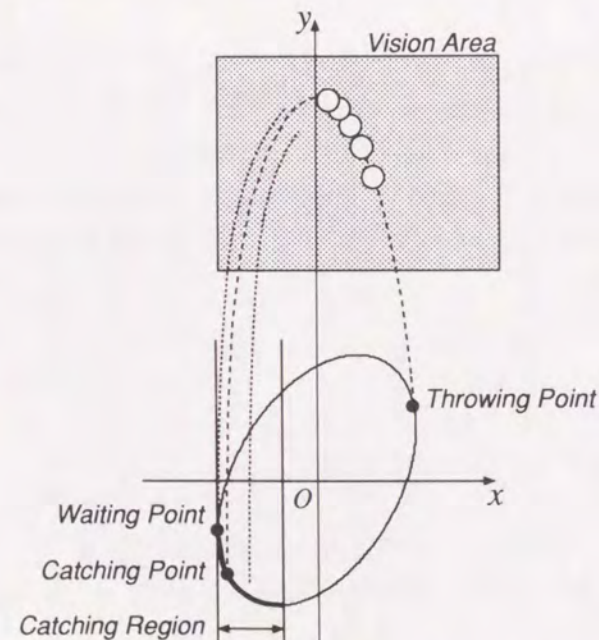


Fig.2.22: Conceptual illustration of "visual feedback"

そこで、

- ボールの弾道軌道は大きくずれない。
- ボールの弾道軌道と手先軌道は常に交差する。

という仮定の下に、捕獲のタイミングを調整する次のような方法を考えた。

具体的には、投げ上げ地点から捕獲地点の間に待機地点を設け、計画された捕獲地点と画像から予測された捕獲地点との時間的なズレ分だけ待機地点での待機時間を調整する。

手先軌道の修正は、以下のようなアルゴリズムにより行う。

[手先軌道を微小修正するアルゴリズム]

1. 捕獲する区間を Fig.2.22 のように限定し、その区間内の楕円軌道上の離散点と、待機地点からその点へ手先が移動するのに要する時間とを関連付けたテーブルを予め作成する。
この離散点の個数によって予測捕獲地点の精度が決定される。
2. 連続した画像数枚から得られたボールの位置情報(重心)より、最小2乗法[35]を用いてボールの弾道(放物)軌道を決定する。
3. ボールの弾道軌道と手先の楕円軌道との交点を2分割法[35]を用いて算出する。
4. 3.で得られた交点から、1.で作成したテーブルを用いて、待機地点から捕獲予定地点までのロボットアームの移動時間を割り出す。
ボールの捕獲予定時刻も先の交点計算にもなって算出されるので、この移動時間を考慮して待機時間が計算される。
5. この予測結果に基づいて、ロボットコントローラに与える待機時間を調整する。
6. 以後、2.~5.を繰り返す。

1.~4.までは、全て画像処理ボードに搭載されたトランスピュータによって計算し、待機時間を調整する微小時間だけを制御を司るトランスピュータに送信する。

この方法は、捕獲のタイミングを調整するだけなので、新たに手先軌道を計画し直さなくても学習した軌道をそのまま使うことができる。しかも、修正に要する計算時間が非常に少なくて済むので、リアルタイム制御に適したリアクティブ・プランニングの一方法と考えることもできる。

実験結果

手先が 32[msec] の間待機する状態を標準とし、ビジュアルフィードバック則によりこの待機時間を変更して落下してくるボールを捕獲する実験を行なった。1 個のジャグリングの場合の結果を Fig.2.23 に示す。

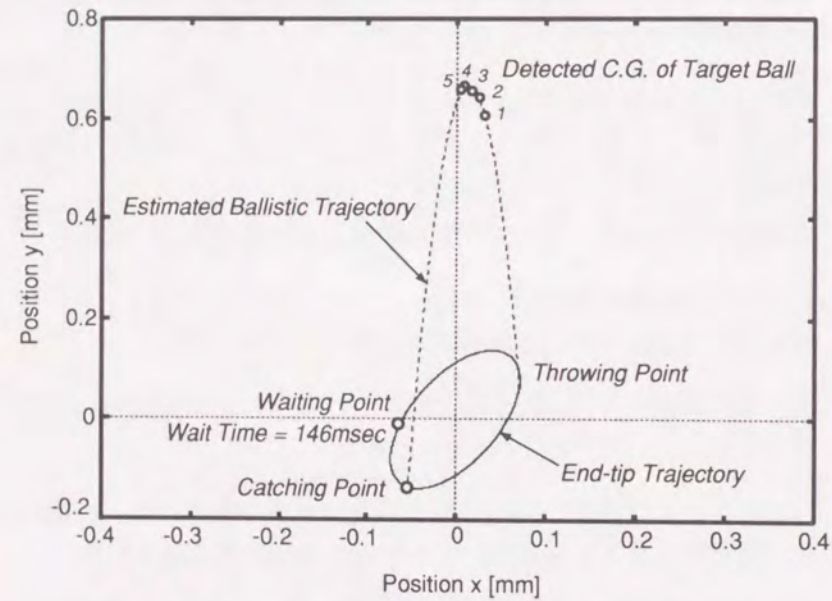


Fig.2.23: Experimental result during one juggling cycle

ボールが画像エリアに入らない限り、手先は待機した状態を保つようになっているので、人間が投げたボールが画像エリアを通り手先の楕円軌道を横切る場合にも本アルゴリズムは有効に働く。Fig.2.24~Fig.2.26 は、画像エリアに入るように人間によって投げられたボールに反応して捕獲動作をとり(1 回の空回り)、さらにそれを投げ上げて再び捕獲する実験結果を示した分解写真である。向かって右に人間が立ち、ボールを上空へ放り投げている。

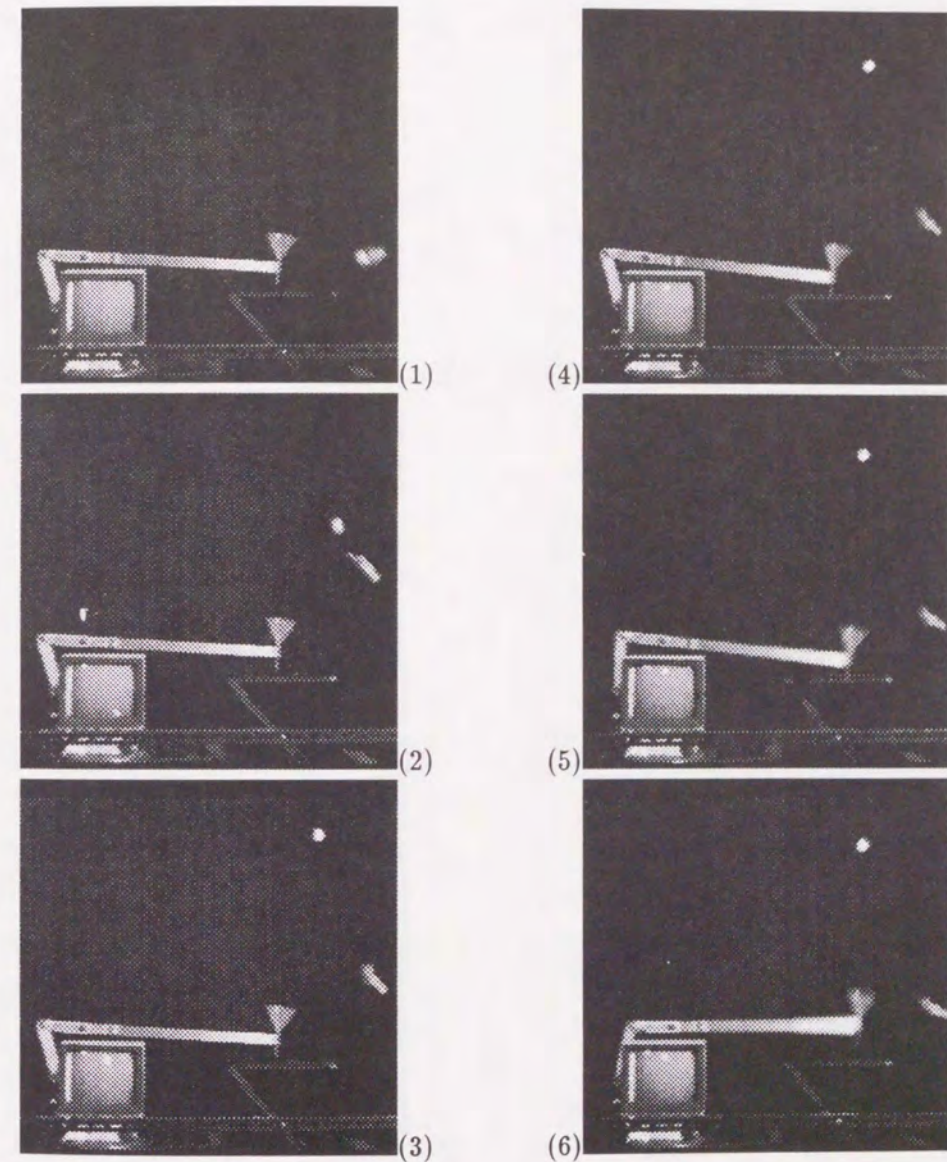


Fig.2.24: Pictures showing one-ball juggling — part 1

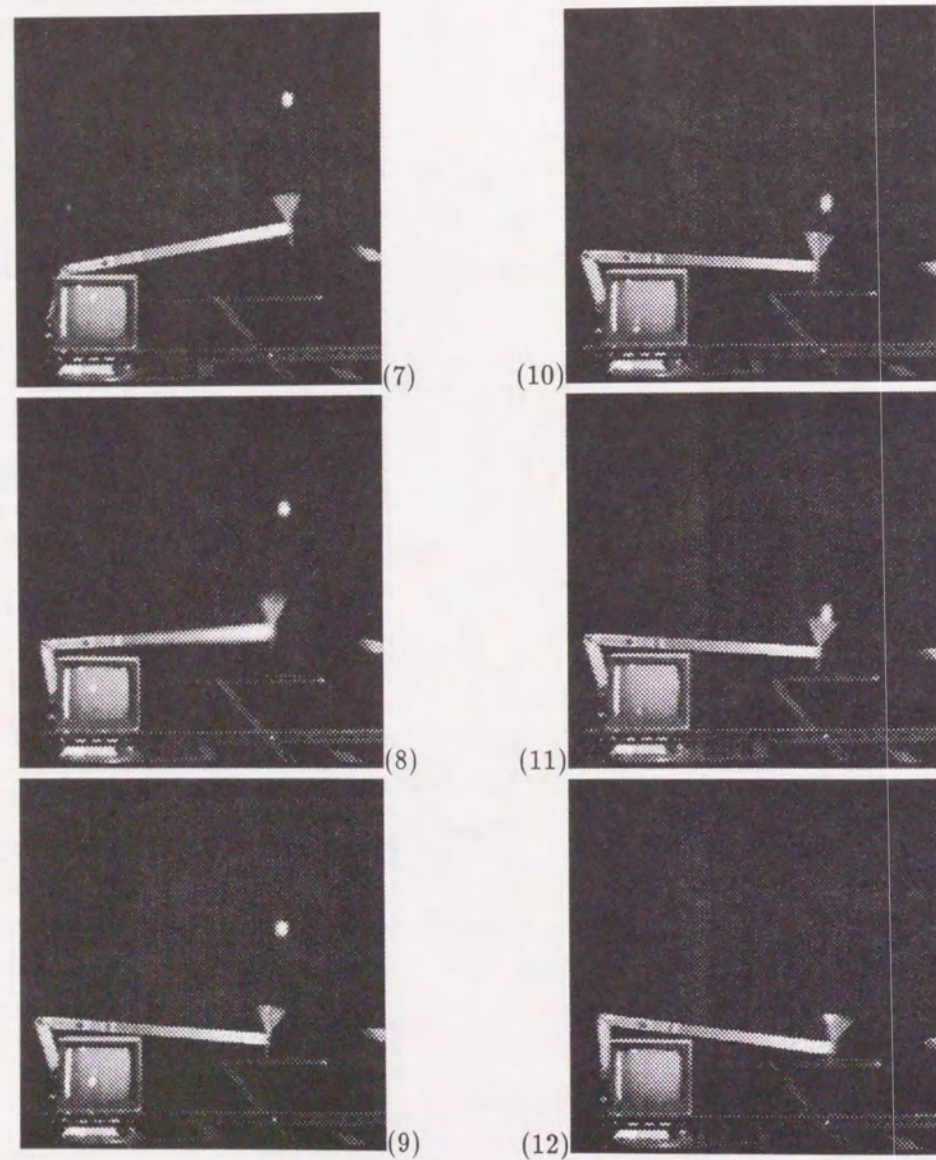


Fig.2.25: Pictures showing one-ball juggling — part 2

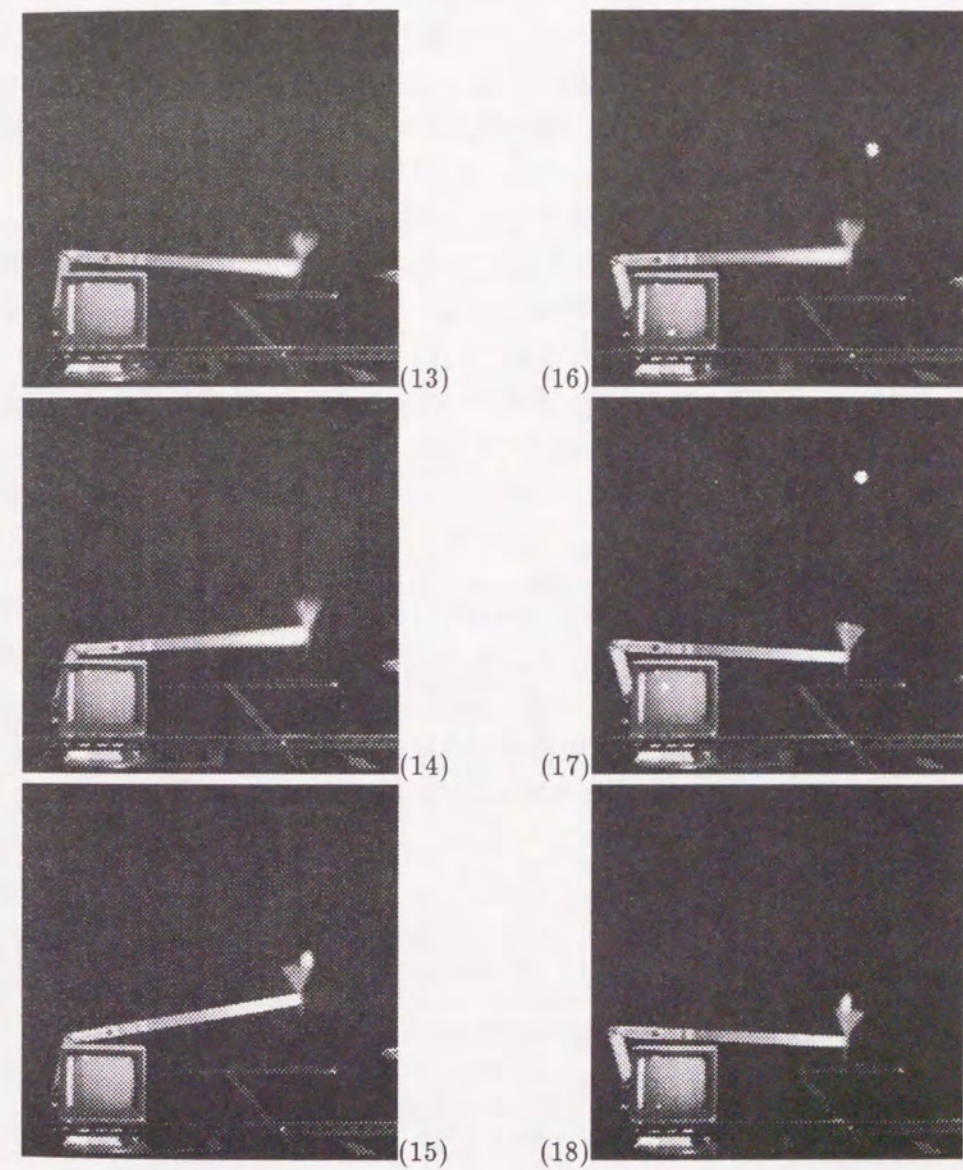


Fig.2.26: Pictures showing one-ball juggling — part 3

写真の左下に画像処理の様子を映し出すモニタが置かれている。

写真1~3は、投げられたボールが画像エリアに入り、弾道軌道の頂点に達するまでを示している。ボールがこの頂点に至るまでに5枚の画像が取り込まれ、2値化されたデータからボールの重心位置を割り出し、これらの5つの重心位置から最小2乗法によって弾道軌道を決定し、この軌道と手先軌道が交わる点から待機時間を計算するという一連の計算が行われる。1枚の画像の取り込み時間は1/30秒である。

写真4から、このボールを捕獲するための行動が始まる。ボール2個のジャグリングを考慮した実験となっているので、手先はまずボールを投げ上げる行動をとってから、標的とするボールを捕獲するようになっている。写真4~10までがその運動を表している。

写真10~11間でアームの初期位置復帰がなされ、写真12で捕獲のための引きの動作が始められている。

写真13以降は、捕獲したボールをロボットが自分自身で投げ上げて再び捕獲する動作を示している。

なお、ボールがアームの可動平面内からずれた位置に投入された場合、1つのカメラしか用いていない本システムでは対処のしようがない。このような場合、捕獲の成功は投げ上げられたボールがこの平面内に残ったまま弾道軌道を描くかどうかで決まる。幸い、カップの構造的な特性により、カップの口径分の軌道誤差までは許容できることが実験的にも確認されている。

2.5 実験例

ボール2個のジャグリングにおける実験結果の一部を以下に示す。

階層学習制御則で得られた軌道パラメータ

ボールの個数 $n = 2$ に対して
 $(l_a, l_b, \beta, \theta_0, T) = (0.143[\text{m}], 0.055[\text{m}], 66.5[\text{deg}], 218.0[\text{deg}], 0.38[\text{sec}])$

上記の軌道パラメータを用いて実験したときの様子の一部を分解写真として Fig.2.27~Fig.2.30 に示す。

まず最初に投げ上げるボールは、既にカップの中に収めてある。写真1の状態

からボールが投げ出され、アームが1周の楕円軌道を終えて再び動き出すまでの間に2個目のボールを人間が投げ入れる。楕円運動に要する時間が0.38[sec]で、アームの初期化に要する時間が50[msec]、そして捕獲のタイミング調整のための待機時間が32[msec]であるから、2個目を投入するタイミングもおろそかにできないが、ビジュアルフィードバックを用いているので、前節で述べたような問題が生じなければ多少のズレはカバーできる。

写真4からボールが弾道軌道の頂点に達するまで(写真6)の間に、空中にあるボールの画像5枚がメモリに蓄えられ、その後待機時間が計算される。

写真7~8にかけてボールは加速されて投げ上げられる。写真9では既に標的となるボールが落ちてきている。

写真10はアームが楕円軌道の一番高い所に位置したときの状態を表している。写真13, 20に見られるように、このアーム状態で2個のボールがほぼ真横に並んでいるときが最も安定したジャグリングとなっている(これは、人間のビデオ解析からも同様なことが言える)。つまり、このような状態になるようにビジュアルフィードバックが機能していると言える。

写真11~12は、落下してきたボールを捕獲するための行動を示している。

本システムで用いている画像処理ボードは、1枚の画像を取り込むのに1/30[sec]かかることは前節で既に述べたが、それが陽にわかるような写真は提示しなかった。写真13からわかるように、ボール2個が横に並んでいるにもかかわらず、モニタにはボールが尾を引くような形で1個しか写っていない。この1/30[sec]後の状態が写真14である。今度はモニタに2個のボールが写っている。また写真15は、投げ上げられて間のないボール状態を示している。この1/30[sec]後の状態が写真16であるが、あまりにも速いスピードでボールが上昇しているため、モニタにボールが3つ存在しているように写っている(実際には縦の2つはつながっている)。現在はボールの弾道軌道の頂点近傍をデータとして用いているので、なんら支障はきたしていないが、扱うボールの個数が増えれば、無視できない問題となるであろう。この場合、画像エリアの再選定、シャッタースピードの速いカメラの使用などでこれらの問題をクリアできると考えられる。

写真17以降は、2個のボールを両方とも捕獲してこのタスクを終了する過程を示したものである。

以上の結果は、ボール1個につき200回、つまり延べ回数にして400回のジャ

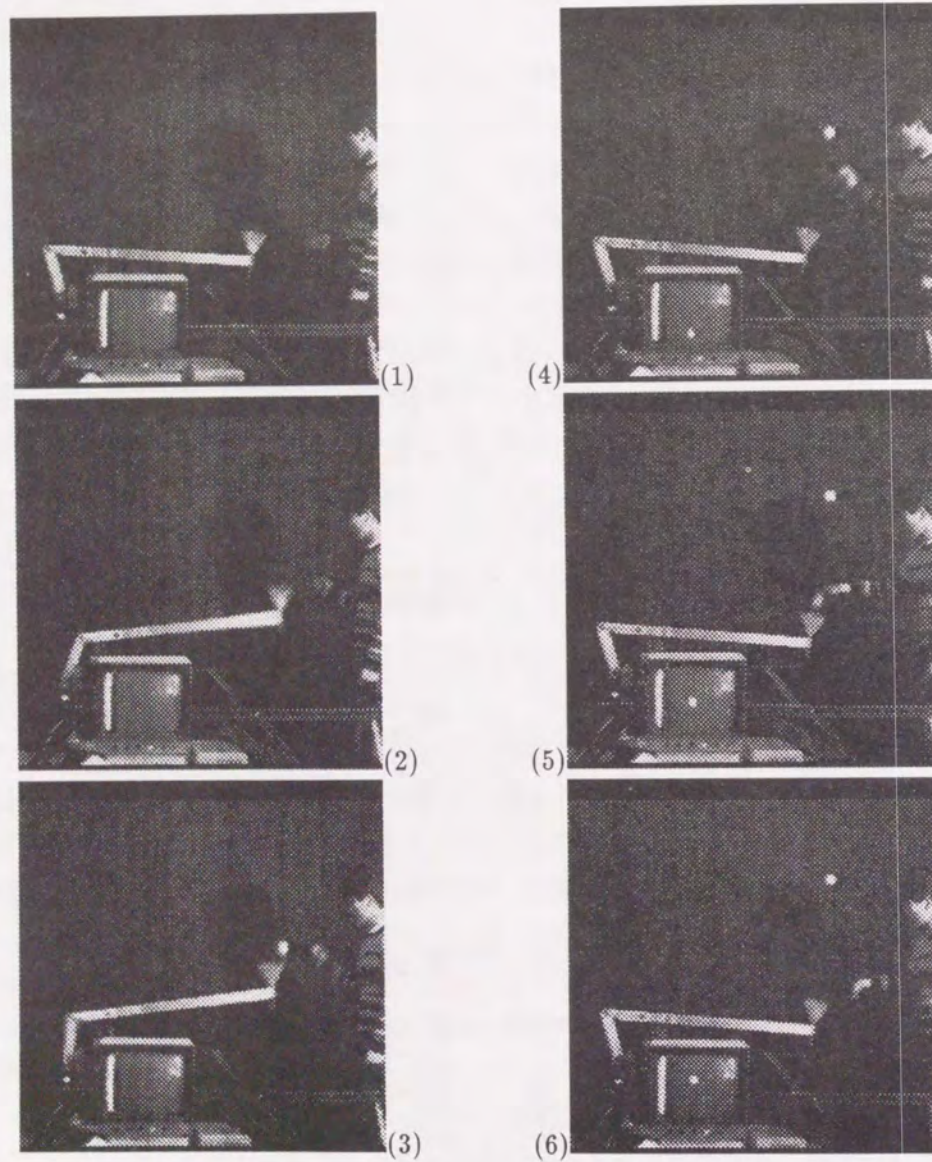


Fig.2.27: Pictures showing two-balls Juggling — part 1

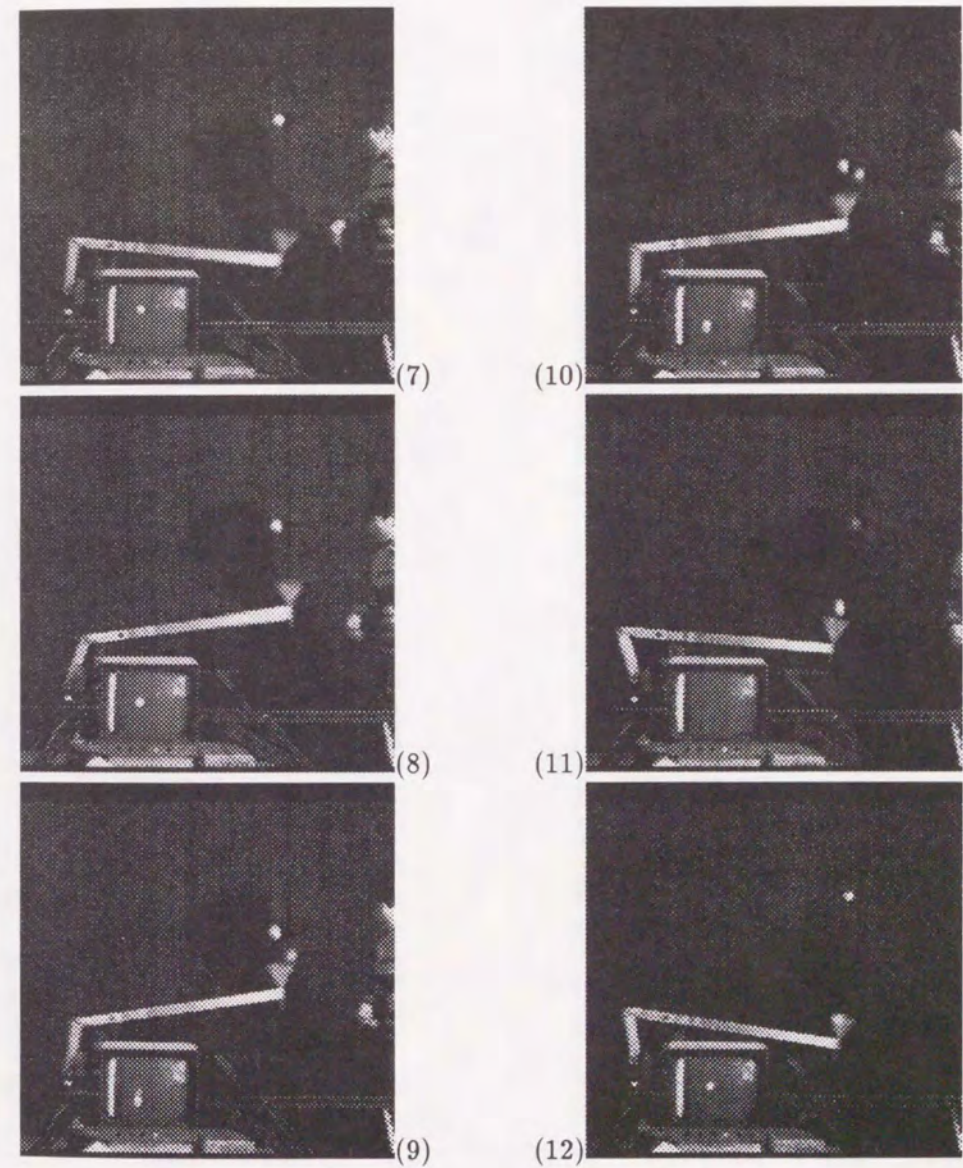


Fig.2.28: Pictures showing two-balls Juggling — part 2

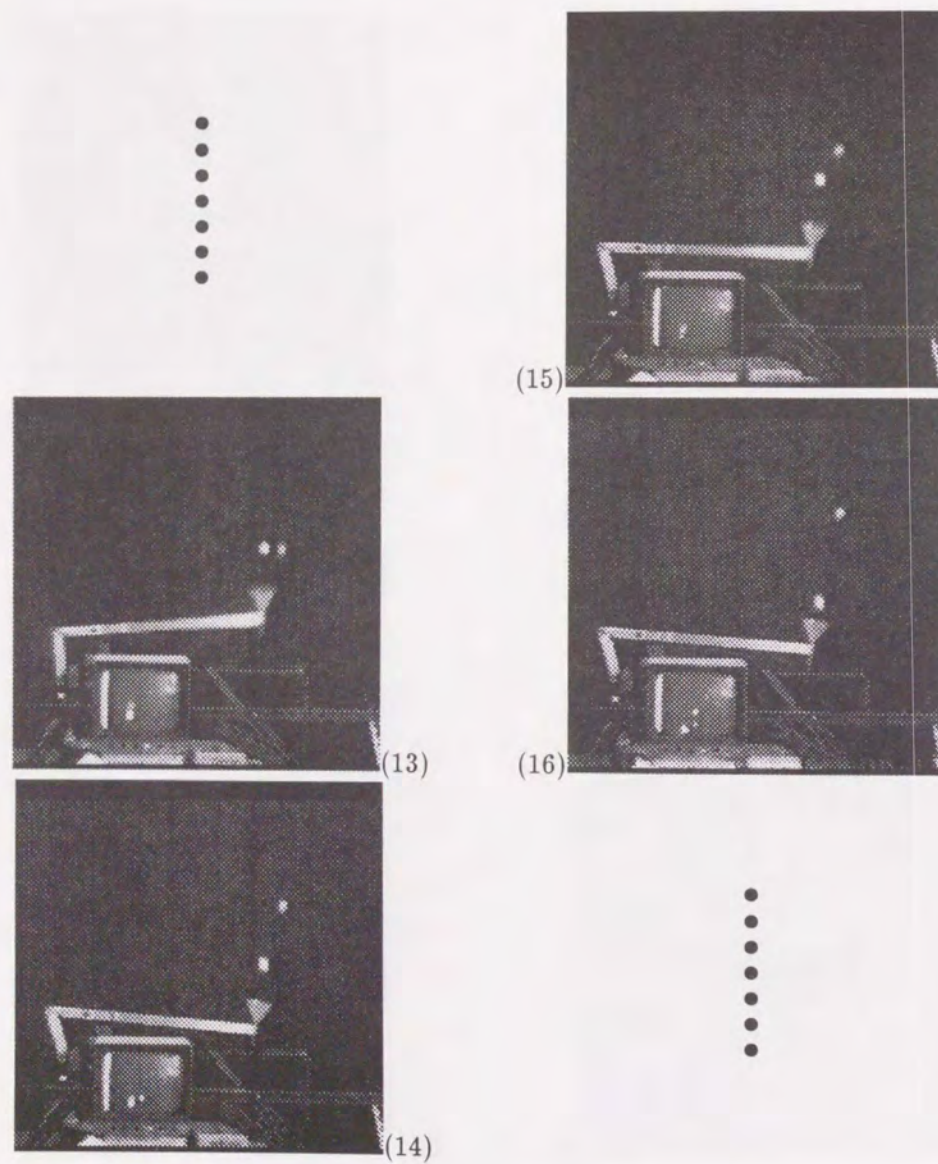


Fig.2.29: Pictures showing two-balls Juggling — part 3

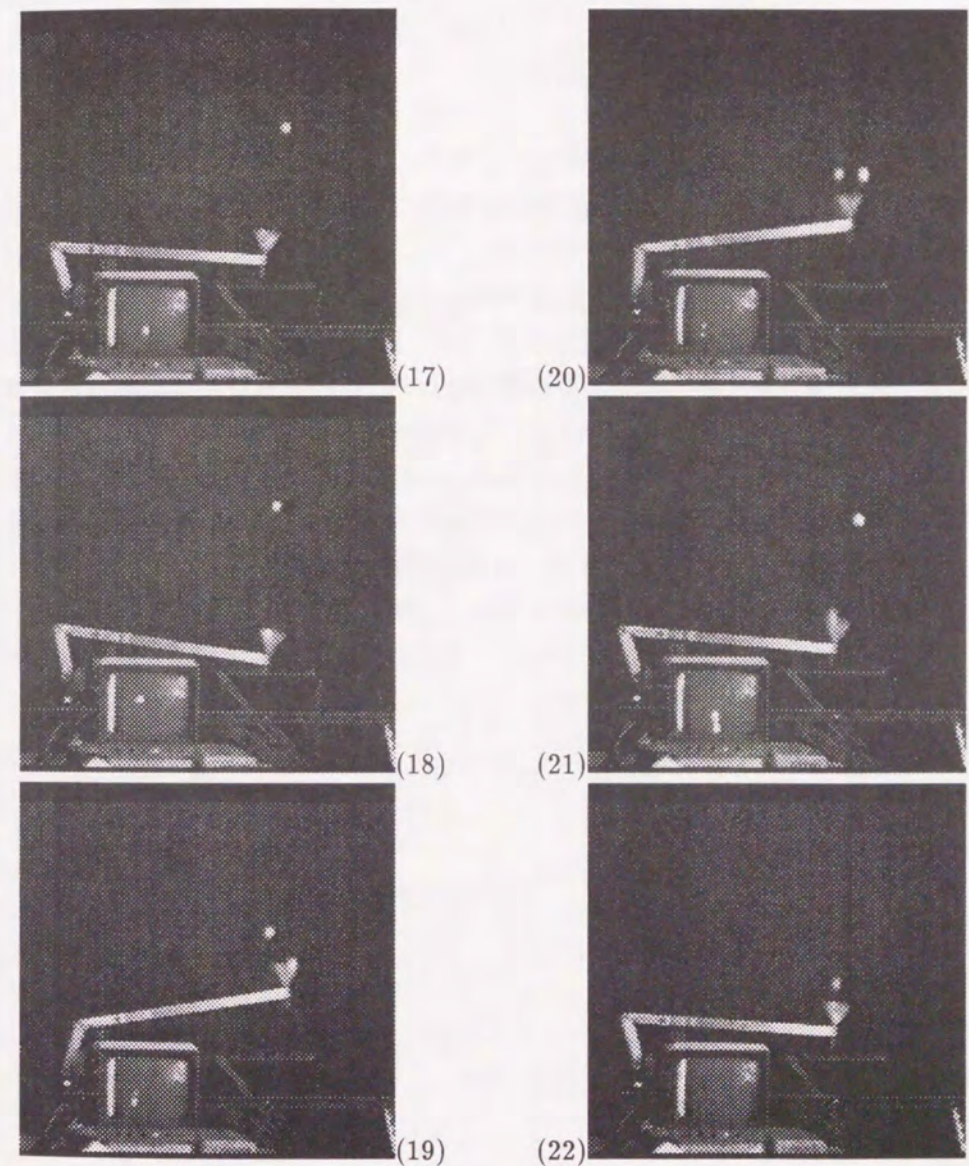


Fig.2.30: Pictures showing two-balls Juggling — part 4

グリングを実行させたときの一部を表したものであるが、400回目にボールを2個とも捕獲して終わるように指令したもので、これ以上の回数も可能である。

2.6 おわりに

本章では、人間の器用なタスクとしてジャグリングを取り上げ、それをロボットが実行する方法について考察した。

タスク実行前の手先軌道を計画する段階では、時空間的に拘束されてはいるが自由度の大きいジャグリングタスクに対して、ロボットの手先軌道をパラメトリックに決定できることを示した。言い換えれば、ジャグリングでは媒介変数や捕獲時間等の各パラメータを変化させることによって、様々な手先軌道を決定することができる。もちろん、この計画方法はロボットの形状に依存せず、タスクに対する「ノウハウ」として用いられるものである。

タスクの実行段階では、計画軌道と実軌道の誤差を解消する階層学習制御則を提案し、また繰り返し段階においては、ビジュアルフィードバックによる捕獲タイミングの微小修正法を示し、それらの有効性を実験的に検証した。

第3章 剣玉ロボット

剣玉のボールの質量や紐の長さなどの事前情報が全く与えられず、しかも手に伝わってくる力覚情報しか利用できない状況下で剣玉を実行するとしたらどのようなにすればよいだろうか？

人間がこのような状況下で剣玉を行なう場合、視覚情報が利用できないため文字どおり手探りでの実行となる。まずは、ボールをぶら下げて何らかの運動をさせ、その動特性の把握に努め、そこから得られた情報を基にしてボールを投げ上げ捕獲する手先軌道を考え、取りあえず実行してみることだろう。その間も手に伝わってくる力覚情報を頼りにボールの運動状態をモニタし続け、剣玉(捕獲)が失敗したと判断されれば、その時の情報を基に先ほどとは違う手先軌道を計画し直しながら成功するまで実行を繰り返す。

ここでは、ロボットが同じ状況下で環境を把握し、自らの力だけで(人間が関わることなく)剣玉を成功させる方策について述べる。

3.1 はじめに

本章では、剣玉について特に

1. ロボットの手先軌道の系統的な計画
2. 計画軌道と実軌道のミスマッチの解消

の問題に焦点を絞って考察する。即ち、ロボットが自律的にタスク遂行能力を身に付ける方法を提案する。特に1.では、前章では取り上げなかった最適化のアプローチにも言及する。

なお、剣玉にはいろいろな技があるが、できるだけ一般的な Fig.3.1 に示すパターン(ボールが振子運動している状態から投げ上げを行い、カップで捕獲する)を考える。タスク遂行に必要なボールの運動状態の検出は、ロボットの手先に装着した力覚センサにより行う。人間では、目を閉じて視覚情報を断ち切った

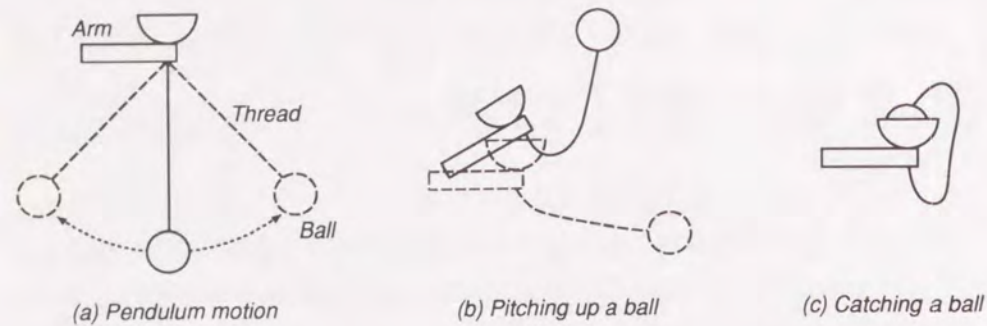


Fig.3.1: "Cup & Ball" task

状態に対応する。また、剣玉の紐の長さやボールの質量は事前にはわかっていないものとする。

1. の軌道計画では、まずロボットの手先軌道として、軌道の滑らかさ、軌道の周期性、広いクラスの軌道パターンを少数のパラメータで表現できることを考慮し、ジャグリングと同様に楕円軌道を採用する。さらに、パラメータを空間的な挙動を表現するパラメータと時間的な挙動を表現するパラメータに分け、前者についてはボールが目標捕獲位置を通過するように、後者についてはボールに加えられる力学的エネルギーが最も大きくなるように決定する。以上のプロセスは、ロボットが剣玉を実施する前の作業計画にあたる。実際に作業を行ってみると、様々な要因により作業目標が達成できないことも起こり得る。2. では、ボールの捕獲の成否を力覚センサーによって判別し、計画軌道を修正する一種の学習方法を示す。最後に、剣玉の紐の長さやボールの質量を未知として行った平面2自由度ロボットマニピュレータによる剣玉の実験結果を示す。

3.2 軌道計画

ロボットは静止し、ボールが紐で釣り下げられた状態を初期状態とする。ボールを振り運動させる動作を行ない、まず軌道計画時に必要となる物理パラメータを推定する(詳細は 3.4.2 節)。

振り運動しているボールをさらに加速して投げ上げ、カップで捕獲するタスク

3.2. 軌道計画

をロボットが実行するには、少なくとも次の2つのサブタスクを実現する動作パターンをロボット自ら計画する能力が必要となる。

第1 サブタスク

ロボットの機構やパワーの制約から、ボールを一気に投げ上げることが不可能な場合には、ボールに力学的エネルギーを付加するための動作が必要となる。このサブタスクの実現は、次の第2サブタスクに比べて容易である。そのアルゴリズムおよび実験結果は 3.4.3 節に示す。

第2 サブタスク

振り運動しているボールが十分な力学的エネルギーを持っていれば、次に必要となるのは適当なタイミングで投げ上げ、目標の位置で捕獲するタスクである。投げ上げ動作開始のタイミングは、ボールの振り運動の振幅から決定する(詳細は 3.4.4 節)。本節では、ある決まったタイミングで開始されるこの第2サブタスクを実現するロボットの手先軌道の計画法について詳しく述べる。

ロボットは、このようにして計画された手先軌道を実際に生成し、ボールの捕獲の成否を力覚情報から判断しながら軌道の再計画・再実行を繰り返す。

3.2.1 振り運動のモデル

軌道計画を行なう場合、ロボットの動作パターンがボールの運動に及ぼす影響を何らかの形でモデル化しておくことが必要である。ここでは、ロボットに事前に与えておく必要最小限の知識として、Fig.3.2 に示すモデルを考える。

図中の x, y 軸は、それぞれ水平、鉛直方向にとり、支点からボールの中心までの長さを l 、ボールの質量を m とする。また、ボールの鉛直からの振れ角を α_t 、その最大値、即ち振幅の大きさを α_s 、紐の張力は F とし、重力加速度を g で表す。手先の座標は楕円の場合と同じく (x_t, y_t) である。

紐により拘束を受けているときのボールには、図中に示すような力が働くから、ボールを質点と考えれば、円周方向、半径方向について、それぞれ以下の関係式が成り立つ。

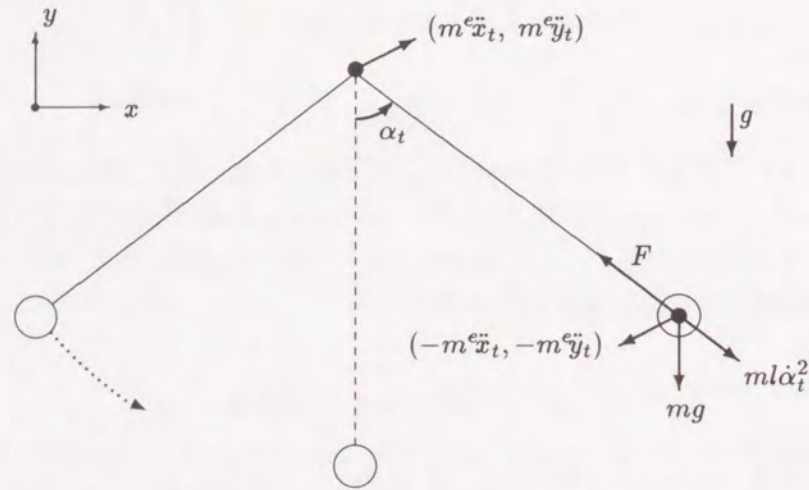


Fig.3.2: Forces acting on a ball in pendulum motion

- m : ball mass
- l : length of a thread
- α_t : swing angle of a thread
- F : tension of a thread
- g : gravitational acceleration

$$ml\ddot{\alpha}_t + mg \sin \alpha_t + m^e\ddot{x}_t \cos \alpha_t + m^e\ddot{y}_t \sin \alpha_t = 0 \quad (3.1)$$

$$F - mg \cos \alpha_t - ml\dot{\alpha}_t^2 = 0 \quad (3.2)$$

ボールが紐によって拘束されないときは、もちろん自然落下の運動モデルに従うものとする。なお、これらのモデルはあくまで近似モデルであるため、それに基づいて導かれる計画軌道も近似の域を出ないことに留意しておく。

3.2.2 投げ上げから捕獲までの手先の計画軌道

第2サブタスクにおいてロボットは、振子運動しているボールを最大の振れ位置にある状態から目標捕獲位置まで移動させるべく、その手先位置を変化させな

ければならない。その間、ボールは紐を通してロボットに拘束される状態からフリーの状態へと移り、ロボットの手先は目標捕獲位置でボールの到着を待つことになる。

このタスクにおけるボール運動の拘束条件はジャグリングのものとは違うが、ジャグリングに用いた楕円軌道がこの場合にも適用できる。

軌道パラメータの設定

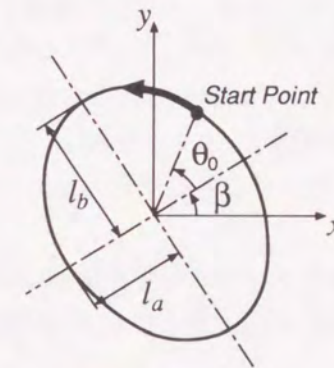


Fig.3.3: Parameters of an elliptic trajectory

式(2.1)と同様の楕円軌道

$${}^e\mathbf{x}_t = \mathbf{R}\mathbf{A}_t\mathbf{l}$$

$$\begin{pmatrix} {}^e x_t \\ {}^e y_t \end{pmatrix} = \begin{pmatrix} \cos \beta & -\sin \beta \\ \sin \beta & \cos \beta \end{pmatrix} \begin{pmatrix} \cos \theta_t & 0 \\ 0 & \sin \theta_t \end{pmatrix} \begin{pmatrix} l_a \\ l_b \end{pmatrix}$$

を採用し、媒介変数 θ_t の形を以下のようにする。

$$\theta_t = \pi \left(1 - \cos \frac{\pi t}{T}\right) + \theta_0 \quad (3.3)$$

ここで、 T は周期であり、 θ_0 は θ_t の初期値である。また、 t は時刻を表し $0 \leq t \leq T$ である。

楕円軌道を決定するパラメータは $(l_a, l_b, \beta, \theta_0, T)$ である。これら5つのパラメータのうち、 $(l_a, l_b, \beta, \theta_0)$ は軌道の空間的な特性に依存したパラメータ(以後

「空間パラメータ」と呼ぶ)であり, T は軌道の時間特性に依存したパラメータ (以後「空間パラメータ」と呼ぶ) である. これらをそれぞれ次のようにベクトル P_s, P_t で表わすことにする.

$$P_s = (l_a, l_b, \beta, \theta_0)$$

$$P_t = T$$

3.2.3 手先軌道パラメータの決定方法

目標捕獲位置 (x_c, y_c) を与え, そこでボールを捕獲できるような手先軌道パラメータ $(l_a, l_b, \beta, \theta_0, T)$ を決定する. ただし, 手先の楕円軌道のスタート時間は, (Fig.3.4) に示すようにボールが手前側の最大振れ角 $-\alpha_s$ にある時刻に等しいものとする. 実験における紐の長さ l , および振幅 α_s は容易に推定できる (3.4.2, 3.4.4 節で詳しく述べる) ので, ここでは予め与えられているものとして話を進める.

ボールは手先の運動によって Fig.3.4 に示すような軌跡を辿る. 張力が零になるまでに手先は 1 周期分の楕円運動を終えてしまうので, 近似的に次の関係が成り立つ.

手先が 1 周期分の楕円運動を終えてからボールの放出 (張力が零) が行なわれるまでの間, ボールの力学的エネルギー (運動エネルギー+位置エネルギー) は保存される.

また本手法では, 空間パラメータ P_s を決定する間は時間パラメータ P_t を固定し, 時間パラメータ P_t を決定する間は空間パラメータ P_s を固定して, 両パラメータの最適値を決定する.

3.2.4 空間パラメータの決定

まず, 空間パラメータ P_s を決定する手順を説明する. 最初に以下のような拘束条件を設定する.

手先が楕円軌道上を動いている間, 紐の張力は零にはならない.

Fig.3.4 のように, 紐の張力が零の瞬間 (以後, 放出時刻と呼ぶ) の振れ角 α_t を α_t とすると, この時点からボールの自由運動が始まる. 手先の運動が完了した直後にボールの自由運動が始まるので, 「剣玉」タスクにおける楕円軌道は, 目標捕獲位置 (x_c, y_c) に向けてボールを投げるのに重要な役割を果たすと言える.

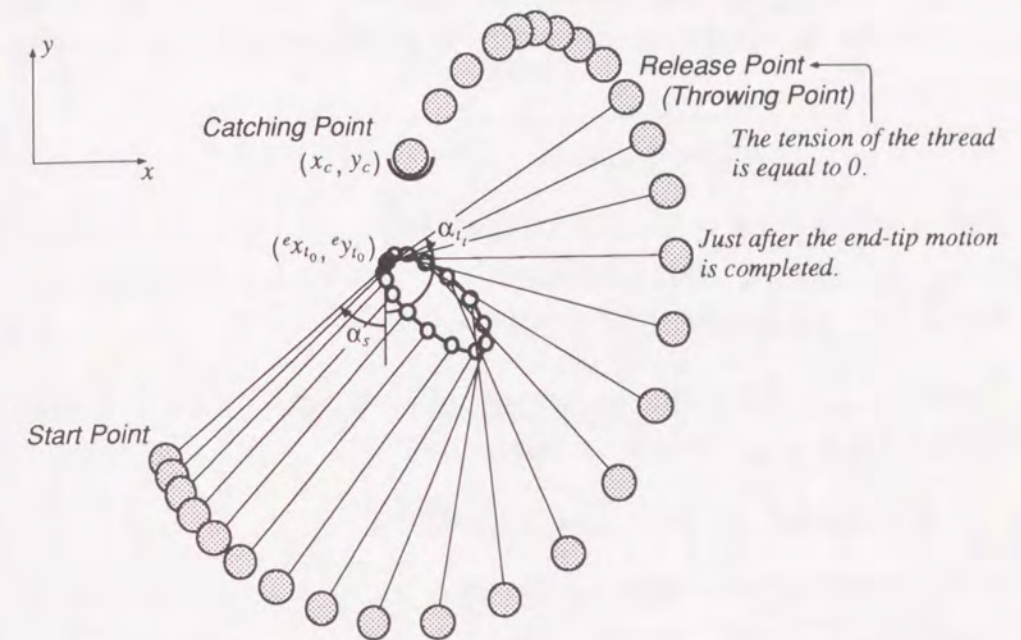


Fig.3.4: Ball motion during “cup & ball” task

$({}^e x_{t_t} + l \sin \alpha_{t_t}, {}^e y_{t_t} - l \cos \alpha_{t_t})$ がボールの弾道軌道の出発点となるので, ボールを捕獲するための条件は, ジャグリングと同じく以下のように表すことができる.

$$x_c = {}^e x_{t_t} + l \sin \alpha_{t_t} + l \dot{\alpha}_{t_t} \cos \alpha_{t_t} \cdot (t_c - t_t)$$

$$y_c = {}^e y_{t_t} - l \cos \alpha_{t_t} + l \dot{\alpha}_{t_t} \sin \alpha_{t_t} \cdot (t_c - t_t) - \frac{1}{2} g (t_c - t_t)^2$$

上式から $(t_c - t_t)$ を消去すると, 次式が得られる.

$$\dot{\alpha}_{t_t}^2 = \frac{A}{B} \quad (3.4)$$

$$A = g (x_c - {}^e x_{t_t} - l \sin \alpha_{t_t})^2$$

$$B = 2l^2 \{ (x_c - {}^e x_{t_t} - l \sin \alpha_{t_t}) \cos \alpha_{t_t} \sin \alpha_{t_t} - (y_c - {}^e y_{t_t} + l \cos \alpha_{t_t}) \cos^2 \alpha_{t_t} \}$$

ここで, ボールが投げ上げられた時点では, ロボットの手先は既に運動の出発点に戻って静止しているので, 投げ上げ座標 $({}^e x_{t_t}, {}^e y_{t_t})$ と手先運動の出発点 $({}^e x_{t_0}, {}^e y_{t_0})$ が同じ点であることを用いた.

一方放出前後で、ボールが紐に拘束された状態からフリーの状態に移るようになるためには、放出時刻において以下のような力の釣合条件が成り立たなければならない。

$$F = m(g \cos \alpha_{t_e} + l \dot{\alpha}_{t_e}^2) = 0 \quad (3.5)$$

これらの式 (3.4), (3.5) を用いると、目標捕獲位置 (x_c, y_c) を与えることによって $(\alpha_{t_e}, \dot{\alpha}_{t_e})$ が決まる。なお、目標捕獲位置は任意に与えることができるが、実際にはロボット手先の楕円軌道上の一点を選んでいる。

求まった $(\alpha_{t_e}, \dot{\alpha}_{t_e})$ を用い、放出時 (紐の張力が零の瞬間) におけるボールの力学的エネルギー E_d を次のように表現しておく。

$$E_d = \frac{1}{2} m l^2 \dot{\alpha}_{t_e}^2 + m g l (1 - \cos \alpha_{t_e}) = m E_d^* \quad (3.6)$$

ただし、位置エネルギーの基準を絶対座標系の $y = {}^c y_{t_0} - l$ とし、 E_d^* はボールの単位質量当りの力学的エネルギーである。

このとき、最初に設定した問題

目標捕獲位置 (x_c, y_c) で捕獲できるように空間パラメータ \mathbf{P}_s を決定する。

は、次のような問題に置き換えることができる。

放出時のボールの力学的エネルギーが E_d^* と等しくなるように、空間パラメータ \mathbf{P}_s を決定する。

力学的エネルギー保存の仮定より、更に次のように書き換えることができる。

手先が一周の楕円運動を終えた瞬間のボールの力学的エネルギーが E_d^* と等しくなるように、空間パラメータ \mathbf{P}_s を決定する。

$(\alpha_{t_e}, \dot{\alpha}_{t_e})$ を手先運動が完了した直後の $(\alpha_t, \dot{\alpha}_t)$ の値とし、評価関数 V_s として

$$V_s(\alpha_{t_e}, \dot{\alpha}_{t_e}) = \left\{ E_d^* - \frac{1}{2} l^2 \dot{\alpha}_{t_e}^2 - l g (1 - \cos \alpha_{t_e}) \right\}^2 \quad (3.7)$$

を定義すると、設定問題は次のようになる。

$V_s(\alpha_{t_e}, \dot{\alpha}_{t_e}) < \epsilon$ ($\epsilon \ll 1$) の条件を満足するように空間パラメータ \mathbf{P}_s を決定する。

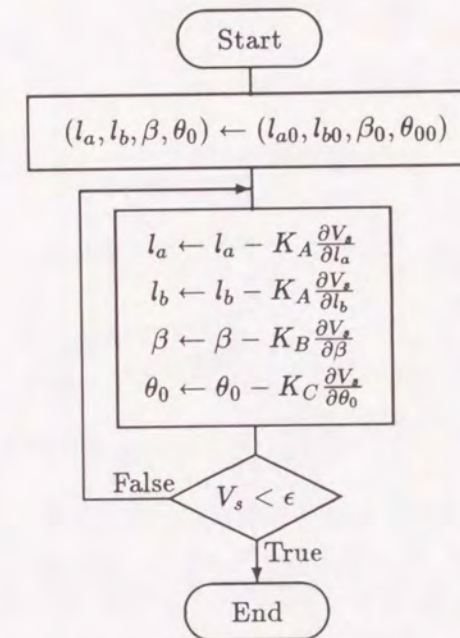


Fig.3.5: Algorithm for acquiring space parameters

この最適化問題を勾配法 [36] を適用して解く。Fig.3.5 にその手順を表すフローチャートを示す。

このアルゴリズムにおいて、空間パラメータ \mathbf{P}_s を修正するために、 $\frac{\partial V_s}{\partial \mathbf{P}_s}$ の値を求める必要がある。 V_s は $\alpha_{t_e}, \dot{\alpha}_{t_e}$ の関数なので、 $\frac{\partial V_s}{\partial \mathbf{P}_s}$ は次のように表すことができる。

$$\frac{\partial V_s}{\partial \mathbf{P}_s} = \frac{\partial V_s}{\partial \alpha_{t_e}} \frac{\partial \alpha_{t_e}}{\partial \mathbf{P}_s} + \frac{\partial V_s}{\partial \dot{\alpha}_{t_e}} \frac{\partial \dot{\alpha}_{t_e}}{\partial \mathbf{P}_s} \quad (3.8)$$

上式の右辺において、 $\frac{\partial V_s}{\partial \alpha_{t_e}}, \frac{\partial V_s}{\partial \dot{\alpha}_{t_e}}$ は式 (3.7) を $\alpha_{t_e}, \dot{\alpha}_{t_e}$ で偏微分し、

$$\frac{\partial V_s}{\partial \alpha_{t_e}} = 2 \left\{ E_d^* - \frac{1}{2} l^2 \dot{\alpha}_{t_e}^2 - l g (1 - \cos \alpha_{t_e}) \right\} \times (-l g \sin \alpha_{t_e})$$

$$\frac{\partial V_s}{\partial \dot{\alpha}_{t_e}} = 2 \left\{ E_d^* - \frac{1}{2} l^2 \dot{\alpha}_{t_e}^2 - l g (1 - \cos \alpha_{t_e}) \right\} \times (-l^2 \dot{\alpha}_{t_e})$$

より求めることができる。なお、 α_{t_e} , $\dot{\alpha}_{t_e}$ は、以下のようなボールの運動方程式から計算できる。

$$\ddot{\alpha}_t = -(g \sin \alpha_t + \ddot{x}_t \cos \alpha_t + \ddot{y}_t \sin \alpha_t)/l \quad (3.9)$$

$$\text{初期条件} \begin{cases} \alpha_t = -\alpha_s \\ \dot{\alpha}_t = 0 \end{cases}$$

また、式(3.8)の $\frac{\partial \alpha_{t_e}}{\partial P_s}$, $\frac{\partial \dot{\alpha}_{t_e}}{\partial P_s}$ は次のようにして求める。式(3.9)の微分方程式を P_s で偏微分すると、

$$l \frac{\partial \ddot{\alpha}}{\partial P_s} = -(g \cos \alpha_t - \ddot{x}_t \sin \alpha_t + \ddot{y}_t \cos \alpha_t) \frac{\partial \alpha_t}{\partial P_s} - \left(\frac{\partial \ddot{x}_t}{\partial P_s} \cos \alpha_t + \frac{\partial \ddot{y}_t}{\partial P_s} \sin \alpha_t \right) \quad (3.10)$$

となる。この式(3.8)を解くには初期条件が必要である。 $t=0$ では、パラメータ P_s の値に関わらず $\alpha_t = -\alpha_s$, $\dot{\alpha}_t = 0$ である。よって、軌道パラメータが異なる値であっても、 α_t , $\dot{\alpha}_t$ をそれぞれのパラメータについて1階微分したときの値は零である。それ故、初期条件は

$$\frac{\partial \alpha_t}{\partial P_s} = \frac{\partial \dot{\alpha}_t}{\partial P_s} = 0 \quad \text{at } t=0$$

のように表わせる。式(3.10)を上期の初期条件のもとで解くことにより、 $\frac{\partial \alpha_{t_e}}{\partial P_s}$, $\frac{\partial \dot{\alpha}_{t_e}}{\partial P_s}$ が求められる。

こうして、式(3.8)の右辺の値をすべて求めることができる。

3.2.5 時間パラメータの決定

次に、時間パラメータ P_t の決定手順を述べる。この手順は、基本的には空間パラメータに対する手順と同じである。

評価関数 V_t を

$$V_t(\alpha_{t_e}, \dot{\alpha}_{t_e}) = \frac{1}{2} l^2 \dot{\alpha}_{t_e}^2 + l g (1 - \cos \alpha_{t_e}) \quad (3.11)$$

のように与える。ここで、 α_{t_e} , $\dot{\alpha}_{t_e}$ は式(3.9)に見受けられるように時間パラメータ $P_t (= T)$ の関数であるので、 $V_t(\alpha_{t_e}, \dot{\alpha}_{t_e}) = V_t(T)$ とみなすことができる。従って、問題を次のように設定する。

$$V_t(T^*) = \max\{V_t(T)\} \text{ を満足する最適な周期 } T^* \text{ を計算する。}$$

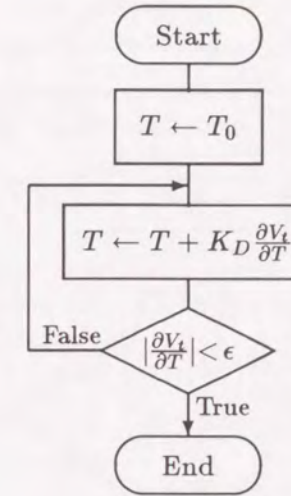


Fig.3.6: Algorithm for acquiring time parameter

即ち、ロボットによってボールに与えられる力学的エネルギーを最大化するような時間パラメータ P_t を決定する。

空間パラメータの決定の場合と同様、この最適化問題を勾配法を適用して解く。その手順を表すフローチャートをFig.3.6に示す。

このフローチャートにおける具体的な計算は、上記の空間パラメータを求めたときの手順と同様である。

3.2.6 全パラメータの決定

実際には、空間パラメータと時間パラメータの決定方法を併用する。まず、空間パラメータを固定して、最適な時間パラメータを計算した後、その時間パラメータを用いて空間パラメータを修正する。この空間パラメータの値を用いて、再び同様のプロセスを繰り返し、修正量が微小となる時点でこのプロセスを終了する。このとき、全軌道パラメータが定まる。

3.2.7 剣玉におけるパラメータ獲得

Fig.3.7は、剣玉タスクにおける全パラメータを計算するための手順を示したものである。

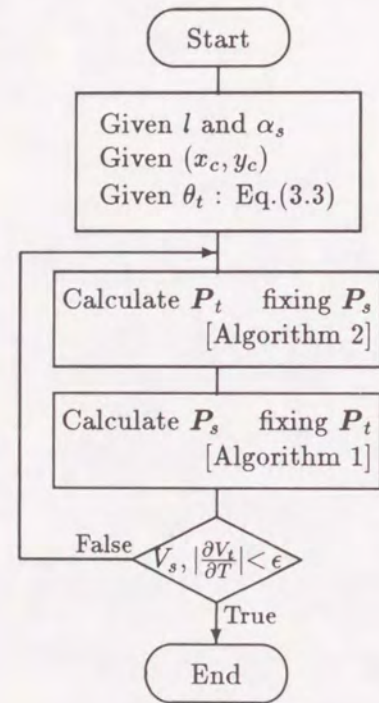


Fig.3.7: Flowchart of acquiring cup & ball parameters

このプロセスで予め与えなければならないのは、紐の長さ l と最大振れ角 α_s , 目標捕獲位置 (x_c, y_c) , 媒介変数の形 θ_t , 及び各最適化問題における空間パラメータ, 時間パラメータの初期値 $(l_{a0}, l_{b0}, \beta_0, \theta_{00}, T_0)$ である.

3.3 軌道の計画結果

Table 3.1 に, 以下の条件の下で本手法を用いて求めた手先軌道パラメータを示す.

- 条件 $K_A = 0.005, K_B = K_C = 1, K_D = 1$
 $\epsilon = 1.0 \times 10^{-6}$
 $({}^c x_{t_0}, {}^c y_{t_0}) = (x_c, y_c) = (0, 0)$

Table 3.1: Resolved ellipse parameters

Trajectory	1	2		
α_s [deg]	50	50		
l [m]	0.44	0.44		
	Initial value	Planned value	Initial value	Planned value
l_a [m]	0.0200	0.0414	0	0.1227
l_b [m]	0.0500	0.0961	0.0300	0.1123
β [deg]	45.0	51.4	0	27.0
θ_0 [deg]	90.0	105.5	30.0	100.3
T [sec]	0.63	0.69	0.70	0.58

Trajectory	3	4		
α_s [deg]	50	65		
l [m]	0.35	0.50		
	Initial value	Planned value	Initial value	Planned value
l_a [m]	0.0200	0.0299	0.0100	0.0384
l_b [m]	0.0500	0.0786	0.0500	0.1135
β [deg]	30.0	43.4	30	43.5
θ_0 [deg]	80.0	104.4	100.0	115.1
T [sec]	0.80	0.59	0.50	0.73

Table 3.1 の trajectory 1 と trajectory 2 を比較する. いずれも, 最大振れ角 α_s , 紐の長さ l の値は同じで, 軌道パラメータの初期値だけが異なる. 条件を満たす, つまりボールが目標捕獲位置を通るように効率よく投げ上げる手先軌道はユニークには決まらず, それぞれ異なる軌道が計画されていることがわかる. 人間の場合を考えると, 同じ剣玉の技に対し, その技を達成する手先の軌道は個人により異なる. この原因の 1 つは, シミュレーション結果にも見られるように, 技の達成に至るまでのプロセスにおいて, 最初の試行に用いた軌道, 即ち手先の初期軌道の違いが考えられる. 技を達成するまでの練習回数についても, 初期軌道の与え方によることが, シミュレーションから推測される.

trajectory 3 は trajectory 1, 2 と比べて紐の長さを短くしたときのものである。紐が短い分、手先はボールにエネルギーを与えやすく、そのために、前の2つの軌道に比べると、長径、短径とも小さい楕円で目標を達成している。

trajectory 4 は、紐は長い場合であるが、初期条件の α_s の値を大きくとったために、ボールに与えられる初期エネルギーが大きく、手先にエネルギーを与える負担が少なくなっている。

また、Fig.3.8, Fig.3.9, Fig.3.10, Fig.3.11に trajectory 1, 2, 3, 4 について、その軌道修正の様子の一部を載せる。なお、微分方程式はルンゲ・クッタ・ギル法を用い、サンプリングタイムを 2.5msec とした。

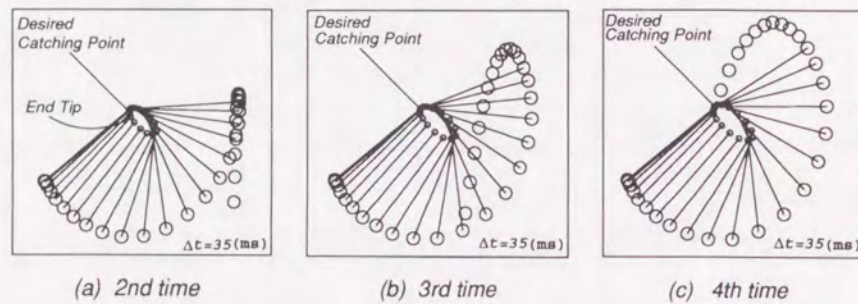


Fig.3.8: Process acquiring a trajectory (trajectory 1 on Table 1)

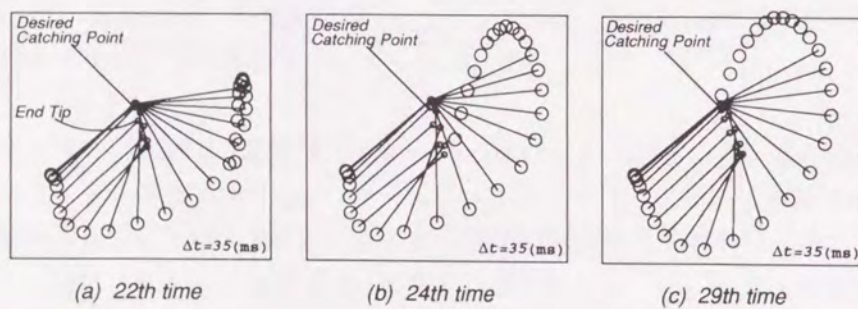


Fig.3.9: Process acquiring a trajectory (trajectory 2 on Table 1)

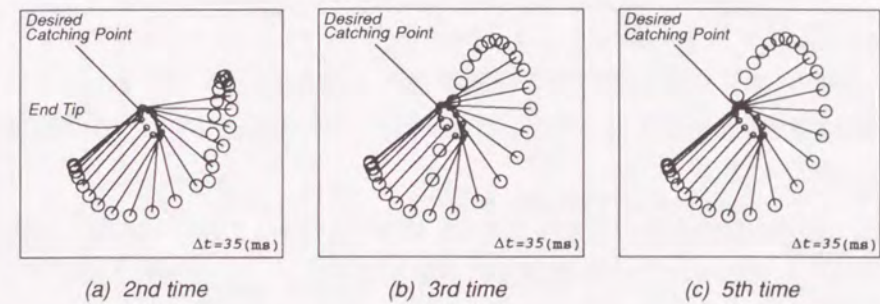


Fig.3.10: Process acquiring a trajectory (trajectory 3 on Table 1)

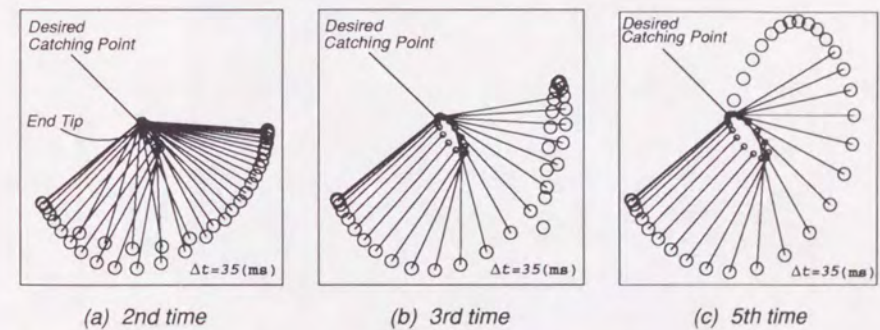


Fig.3.11: Process acquiring a trajectory (trajectory 4 on Table 1)

3.4 実験的検討

3.4.1 システム構成

Fig.3.12 にシステム構成を示す。剣玉を行なわせるためのマニピュレータとして、ダイレクトドライブ方式の鉛直平面 2 自由度平行リンクマニピュレータ (Fig.3.13 参照) を使用する。なお、手先のカップが常に水平を保つよう機構的な工夫を加えている。

また、環境認識のための力覚センサ (3 軸ロードセル) をマニピュレータの先端に取り付け、そのセンシング部に紐の一端を取り付け、紐の張力を検出する (附章 C 参照)。またそのセンサ信号は、ストレイン・アンプで増幅し、AD 変換モジュールにより、制御マシンに取り込む。

マニピュレータへの指令は、DA 変換モジュールを介して速度制御モードで行い、またマニピュレータの関節の角変位は、DD モータ付属のエンコーダにより発生するパルス信号をカウンタモジュールでカウントして取り込む。サンプリング時間は 1.5msec である。

ボールは市販されている剣玉のボール (木製) で直径 0.06m、質量 0.055kg、紐はたこ糸で長さ 0.41m、捕獲器はアルミニウム製のカップで口の直径 0.09m のものを用いる (Fig.3.14 参照)。

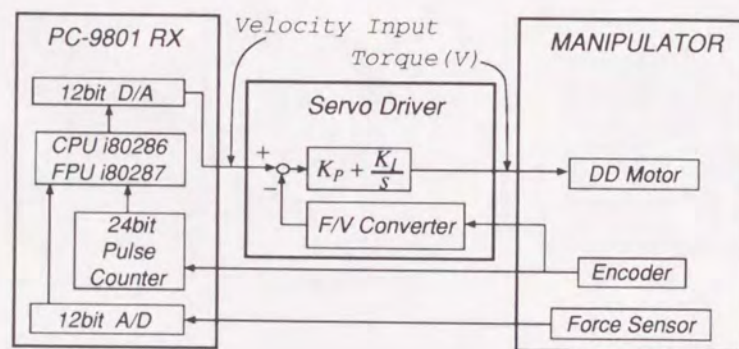


Fig.3.12: Schematic diagram of the system

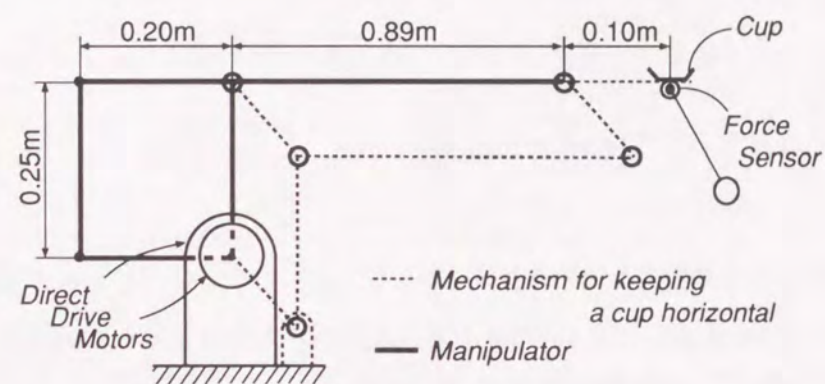


Fig.3.13: Experimental apparatus

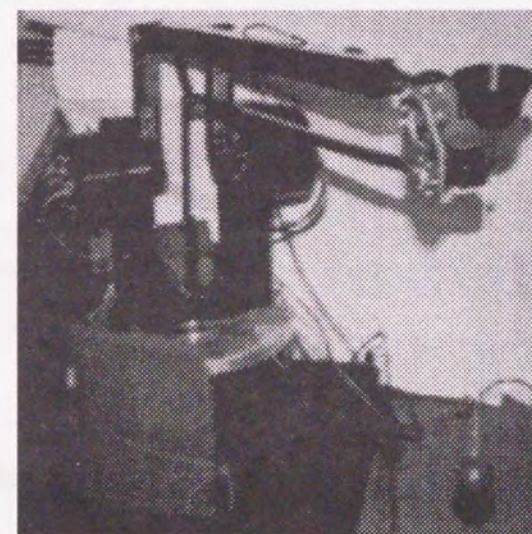


Fig.3.14: Picture of DD robot arm with mechanism to keep a cup horizontally

3.4.2 物理パラメータの推定

ボールの重さの推定

ボールを紐で釣り下げ静止した状態での力覚センサの値より推定する。

紐の長さの推定

この推定は、以下のようなアルゴリズムにより行う。

[紐の長さを推定するアルゴリズム]

1. ボールを紐で釣り下げ静止した状態から、マニピュレータの手先に適当な水平往復運動を行わせる。
2. 手先を止める (このときボールは振子運動している)。
3. 紐の張力を連続して測定することにより、そのときの周期 T_p と振幅 α_s を算出する。なお、 α_s の算出法については、次々節で述べる。
4. 次の関係式

$$l = \frac{T_p^2 g}{16 \left(\int_0^{\frac{\pi}{2}} \frac{d\phi}{\sqrt{1-k^2 \sin^2 \phi}} \right)^2} \quad k = \sin \frac{\alpha_s}{2} \quad (3.12)$$

を用いて、紐の長さ l を推定する。なお、上式の () の中の積分値は、文献 [40] の完全楕円積分のテーブルを利用するものとする。

本研究で考えている振子運動は、 α_s が大きいいため、線形化した微分方程式から得られる結果は使えない。そこで、紐の長さ l を求める式は、式 (B.10) を l について解いたものを用いている (附章 B 参照)。

実験結果

● ボールの重さの推定

$m=0.055[\text{kg}]$ のボールに対する 15 回の測定結果は Table 3.2 に示す通りである。この結果から、 $\pm 0.7\%$ の精度範囲で推定できていることがわかる。

Table 3.2: Estimation for the mass of a ball

Mass of a ball		$m[\text{kg}]$
True value		0.055
Estimated values	maximum	0.0554
	minimum	0.0547

● 紐の長さの推定

[紐の長さを推定するアルゴリズム] に従って、3 種類の長さの紐について、それぞれ 5 回の推定を行った。Table 3.3 にその結果を示す。どの紐の長さの場合も、 $\pm 1[\text{cm}]$ 以上の推定誤差を生じる場合があるので、実際にこの推定値を軌道計画に用いる場合は、より推定誤差を小さくするために、同推定を数回繰り返してその平均値を用いる。

Table 3.3: Estimation for the length of the thread

Length of the thread		$l[\text{m}]$		
True value		0.50	0.47	0.37
Estimated values	1st trial	0.498	0.469	0.358
	2nd trial	0.495	0.468	0.373
	3rd trial	0.518	0.460	0.365
	4th trial	0.496	0.464	0.365
	5th trial	0.495	0.473	0.358

3.4.3 振幅制御のアルゴリズム

第 1 サブタスク、即ちボールの力学的エネルギーを付加するための動作としては、Fig.3.15 のように手先を強制振動させる方法が最も簡便である。

手先の力覚センサで同期をとりながら、振子運動しているボールの振幅を制御するアルゴリズムを以下に示す。

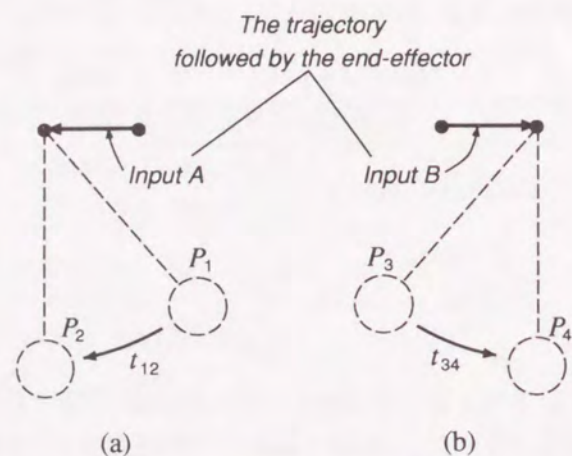


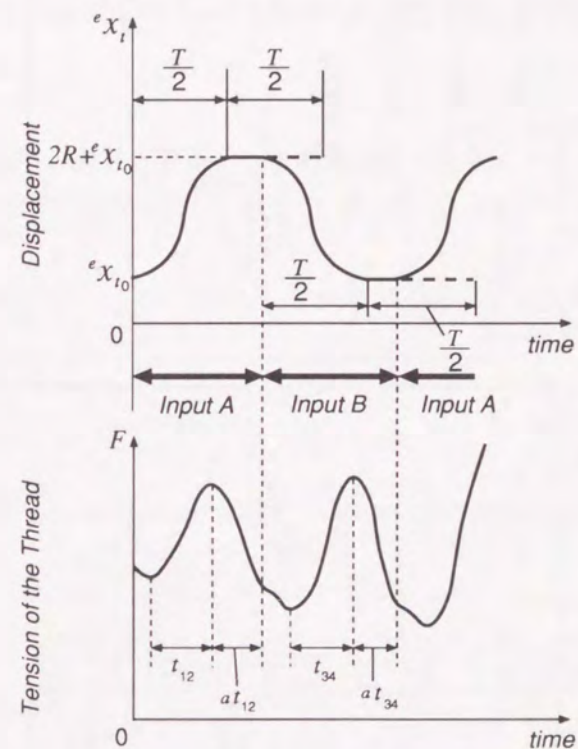
Fig.3.15: Swing control in pendulum motion

[振幅制御のアルゴリズム]

1. 手先にある往復運動をさせるような関節への指令値 (時系列の指令値データ) を求めて, その指令値を 2 つに分け, 前半の指令値を A, 後半の指令値を B とする.
2. まずボールが静止している状態から指令値 A の入力を開始する.
3. Fig.3.15(a) において $[P_1]$ (振幅片端) から $[P_2]$ (振幅中央) の所要時間を t_{12} 秒とし (状態 $[P_1]$, $[P_2]$ は力覚センサの値より判断する), $[P_2]$ を通過してから $a \cdot t_{12}$ ($0 \leq a \leq 1$) 秒後に指令値 A を指令値 B に変更する.
4. 力覚センサ値がある設定値を超えるまで, 3 を繰り返す.

なお, P_1 と P_3 の区別は力覚センサの水平成分を利用し, a を適当に与えることで振動の増減を制御する.

このアルゴリズムの 1. における指令値の軌道としては様々な軌道が考えられるが, 本研究では, 次式のような軌道を与える (Fig.3.16 参照).

Fig.3.16: Specified displacement e_x_t

• Input A

$$e_x_t = \begin{cases} e_x_{t_0} + C(1 - \cos \frac{2\pi t}{T_s}) & (0 \leq t \leq \frac{1}{2}T_s) \\ e_x_{t_0} + 2C & (\frac{1}{2}T_s \leq t \leq T_s) \end{cases} \quad (3.13)$$

$$e_y_t = e_y_{t_0} \quad (0 \leq t \leq T_s) \quad (3.14)$$

• Input B

$$e_x_t = \begin{cases} e_x_{t_0} + C(1 - \cos \frac{2\pi t}{T_s}) & (0 \leq t \leq \frac{1}{2}T_s) \\ e_x_{t_0} & (\frac{1}{2}T_s \leq t \leq T_s) \end{cases} \quad (3.15)$$

$$e_y_t = e_y_{t_0} \quad (0 \leq t \leq T_s) \quad (3.16)$$

ここで、 (x_{t_0}, y_{t_0}) は手先の初期位置を示し、 $T_s = 2\pi\sqrt{\frac{l}{g}}$ である。また、 C は定数であり、実験的に求めた値を用いることとする。

また目標の振幅よりも大きな振幅でボールが振り運動しているとき、その振幅を小さくしたい場合も [振幅制御のアルゴリズム] の係数 a の値を変えればよい。

実験結果

まず、マニピュレータを静止させたままボールのみを振り運動させた場合の紐の張力 (力覚センサの値) $f_x, f_y, F = \sqrt{f_x^2 + f_y^2}$ の変化の様子を **Fig.3.17** に示す。

この図から、滑らかな正弦波状の張力 F が得られていることがわかる。

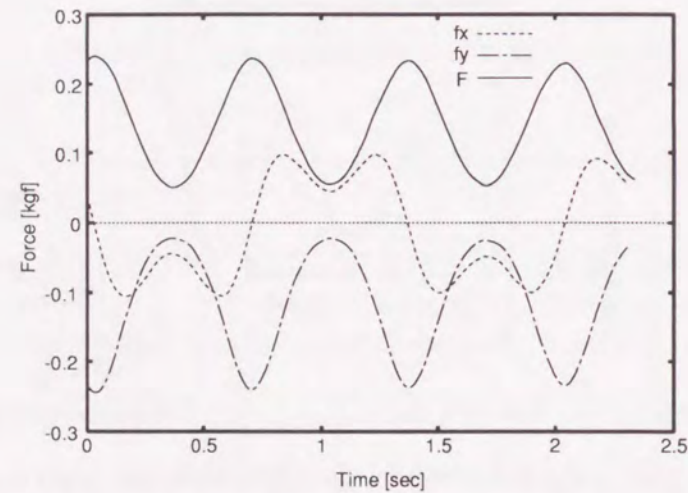


Fig.3.17: Force signals in pendulum motion

そこで、以下のような条件のもとに実験を行い、

- 条件 $R = 0.1[\text{m}], \quad T_s = 0.93[\text{sec}]$

吊り下げられ静止しているボールを振り運動させたり ($a = 2.0$)、振り運動しているボールの振幅を小さくすること ($a = 1.2$) が可能であることを確認した。**Fig.3.18**, **Fig.3.19** に振幅を増減させたときの力覚センサ値 F の変化の様子を示す。

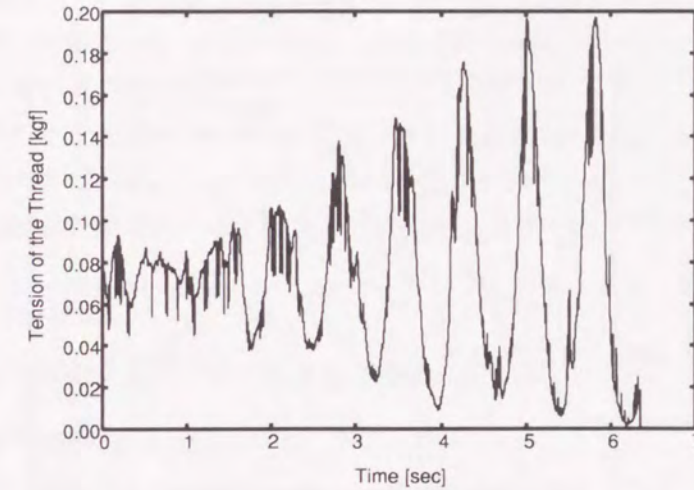


Fig.3.18: Tension in pendulum motion (increasing its swing)

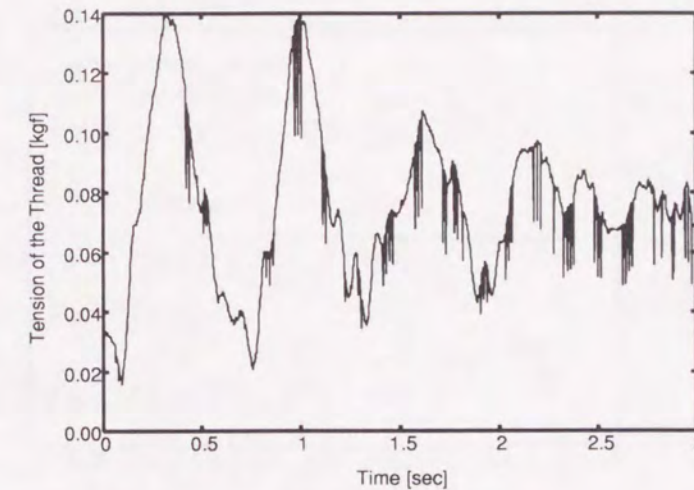


Fig.3.19: Tension in pendulum motion (decreasing its swing)

Fig.3.20, Fig.3.21 に、実験時にマニピュレータが描いた手先軌道も示しておく。

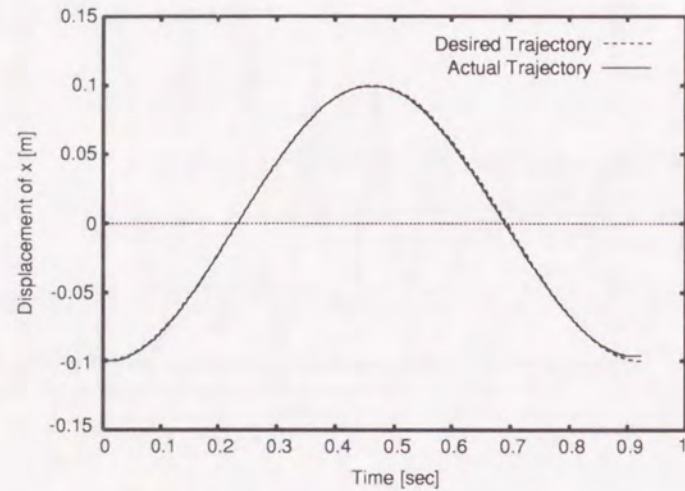


Fig.3.20: Displacement e_{x_t} in swing control

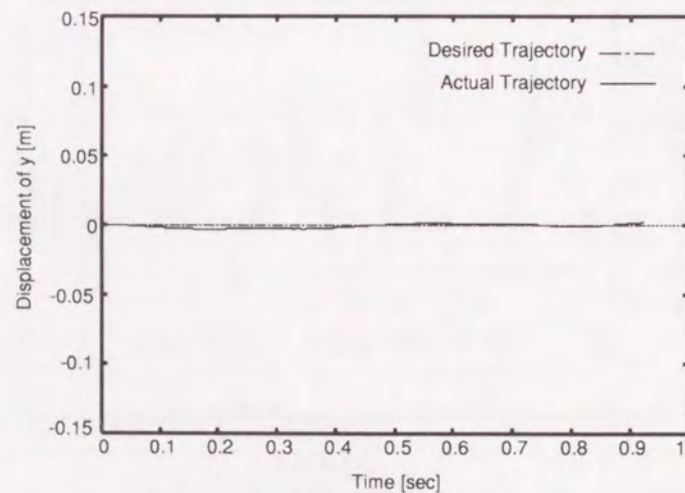


Fig.3.21: Displacement e_{y_t} in swing control

3.4.4 振幅の推定

第2サブタスクの開始は、振子運動しているボールが手前側の最大の振れ位置にある時刻とする。このようにすれば、そのときの振幅を用いて、以後の手先軌道が容易に決定できる(3.2節)。

さて、振幅の推定に振子運動中の周期を用いる方法も考えられるが、周期は振幅の違いにあまり影響されないので、振幅の推定には用いにくい。そこで、次のように紐の張力の最大値と最小値を用いて振幅を推定する。

張力の最小値 (F_{min})

張力が最小になるのは、ボールが最大振幅にあるときである。

$$F_{min} = mg \cos \alpha_s \quad (\alpha_s = \max \{ \alpha \}) \quad (3.17)$$

張力の最大値 (F_{maz})

張力が最大になるのは、ボールが振幅中央にあるときである。

$$F_{maz} = mg + ml\dot{\alpha}_c^2 \quad (\dot{\alpha}_c = \max \{ \dot{\alpha} \}) \quad (3.18)$$

ところで、ボールの力学的エネルギー(位置エネルギー+運動エネルギー)が保存されるとすると、次の関係式が成り立つ。

$$mgl(1 - \cos \alpha_s) = \frac{1}{2} ml^2 \dot{\alpha}_c^2 \quad (3.19)$$

上式を l について解き、式(3.18)に代入すると、

$$F_{maz} = mg(3 - 2 \cos \alpha_s) \quad (3.20)$$

と表せる。式(3.17), (3.20)より mg を消去すると、

$$\frac{F_{maz}}{F_{min}} = \frac{3 - 2 \cos \alpha_s}{\cos \alpha_s} \quad (3.21)$$

となる。上式を α_s について解くと、

$$\alpha_s = \cos^{-1} \left(\frac{3}{2 + \frac{F_{maz}}{F_{min}}} \right) \quad (3.22)$$

と表せる。式(3.22)を用いれば振幅を推定できる。この推定方法はボールの質量や紐の長さの推定値を用いなくてもよい点に特徴があり、実際にも高い精度(誤差 $\pm 1(\text{deg})$ 程度)で推定できた(Table 3.4 参照)。

実験結果

ボールを振り運動させたときの振幅の真値と、紐の張力の最大最小値、並びに式 (3.22) より算出した振幅の推定値を **Table 3.4** に示す。ここで、振幅の真値に関しては、紐の固定部（手先）に取り付けた分度器から目測で読み取った値であるので、若干の誤差が含まれている可能性がある。

Table 3.4: Estimation for the amplitude of the pendulum motion

True value[deg]	F_{max} [kg]	F_{min} [kg]	Estimated value[deg]
22	0.05717	0.04719	20.91
29	0.06421	0.04401	29.85
44	0.07857	0.03519	44.89

3.4.5 計画軌道と実軌道のミスマッチ解消アルゴリズム

3.2 節に述べた方法で計画された手先軌道が精度よく実現できたとしても、ボールに作用する空気抵抗をはじめ、モデル化していない要素によって計画された通り作業は実行されない。実際、実験では学習制御 [49]~[48] を用いて手先をほぼ計画軌道通り動作させたにもかかわらず、目標位置でボールの捕獲ができないケースが生じた。

そこで、モデル化されていない要素によるエネルギーロスを補う次のような方法を考えた。

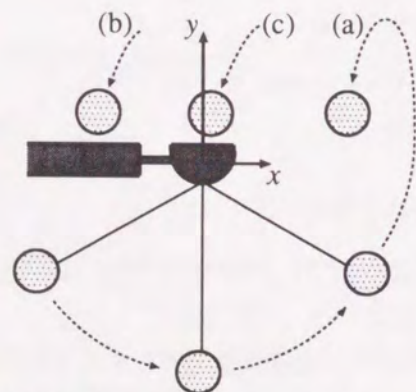


Fig.3.22: Situations after pitching a ball

[ミスマッチ解消アルゴリズム]

1. ボールの投げ上げ後も力覚センサ情報 f_x を連続してとり続けることにより、

- (a) カップよりも $+x$ 側にボールが落ちた場合
- (b) カップよりも $-x$ 側にボールが落ちた場合
- (c) カップにボールが入った場合

を見分ける (**Fig.3.22**参照).

2. 手先運動開始時の最大振れ角 α_s を、

- (a) の場合は $\alpha_s = \alpha_s + \delta$ [deg]
- (b) の場合は $\alpha_s = \alpha_s - \delta$ [deg]
- (c) の場合はそのまま

のように微小修正を行う (δ は微小修正量).

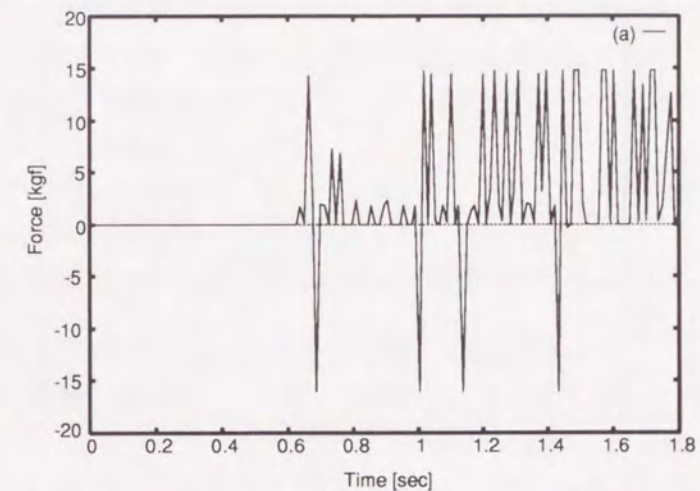


Fig.3.23: Force signal after pitching in case (a)

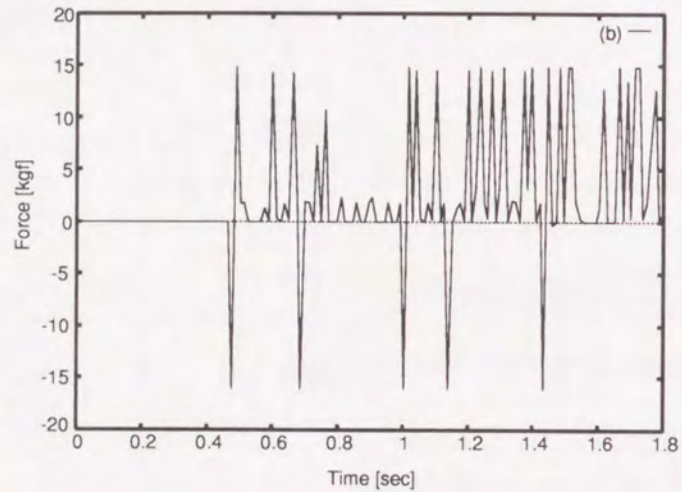


Fig.3.24: Force signal after pitching in case (b)

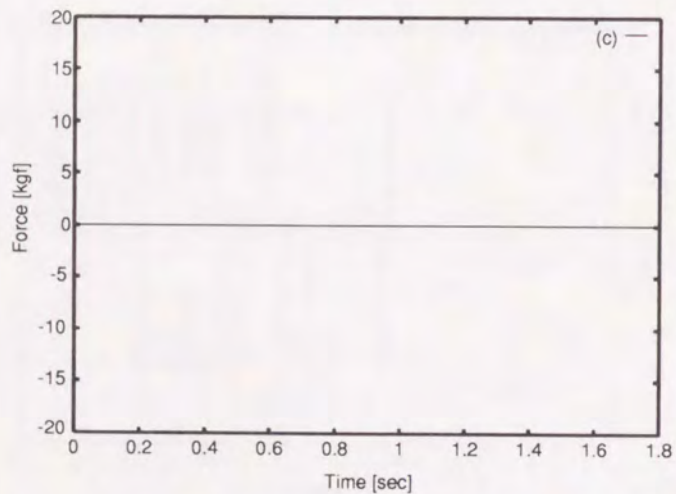


Fig.3.25: Force signal after pitching in case (c)

以上の方法により、手先軌道を変化させることなく、ボールの軌道を微小修正することが可能となった。

Fig.3.23, Fig.3.24, Fig.3.25 に (a)~(c) それぞれの場合の投げ上げ後にモニタされた力覚センサの値 f_x を示す。これより、最初のトリガ値がプラスの場合は (a) の場合、マイナスの場合は (b)、トリガが入らなかったときが (c) の場合と判別できる。

3.5 実験例

実験結果の一部を以下に示す。

軌道計画で得られた軌道パラメータ

- 軌道パラメータ

$$\left. \begin{array}{l} \text{始点} \quad ({}^c x_{t_0}, {}^c y_{t_0}) = (0.0, 0.0) \\ \text{目標捕獲位置} \quad (x_c, y_c) = (0.0, 0.07[\text{m}]) \\ \text{振幅} \quad \alpha_s = 50.0[\text{deg}] \end{array} \right\} \text{に対して}$$

$$(l_a, l_b, \beta, \theta_0, T) = (0.0547[\text{m}], 0.0890[\text{m}], 18.47[\text{deg}], 139.19[\text{deg}], 0.63[\text{sec}])$$

に対し、実際に剣玉が実現できたのは、 $\alpha_s = 58.0[\text{deg}]$ であった。

なお、本実験では $\delta = 1.0[\text{deg}]$ として微小修正を行った。また、目標捕獲位置が y 方向において $0.07[\text{m}]$ ずれているのは、紐の支持部とカップの位置がそれだけ離れているからである。

なお軌道パラメータを変えても、計画時の α_s に対してボールは常に目標捕獲地点の手前 ($+x$ 側) を通過する傾向があった。即ち、エネルギーロスによって生じる軌道誤差であり、やはり上述の軌道誤差解消方法によって解決し得た。

このときの様子を分解写真として Fig.3.26~Fig.3.27 に示す。まず、写真1~3のように振子運動を繰り返し、目標とする振幅になるように手先を制御する。手先に取り付けられた力覚センサの値を基に、振子運動の最大振幅が目標振幅になったとき (写真3) に手先は楕円運動を始める。写真4~7が、アームが楕円運動をしている様子を示したものである。写真7は手先の楕円運動が終了した瞬間である。その後も暫くは紐の張力は残っており、ボールは紐に拘束されたまま上昇を続け、写真8の位置で自由運動を始める。

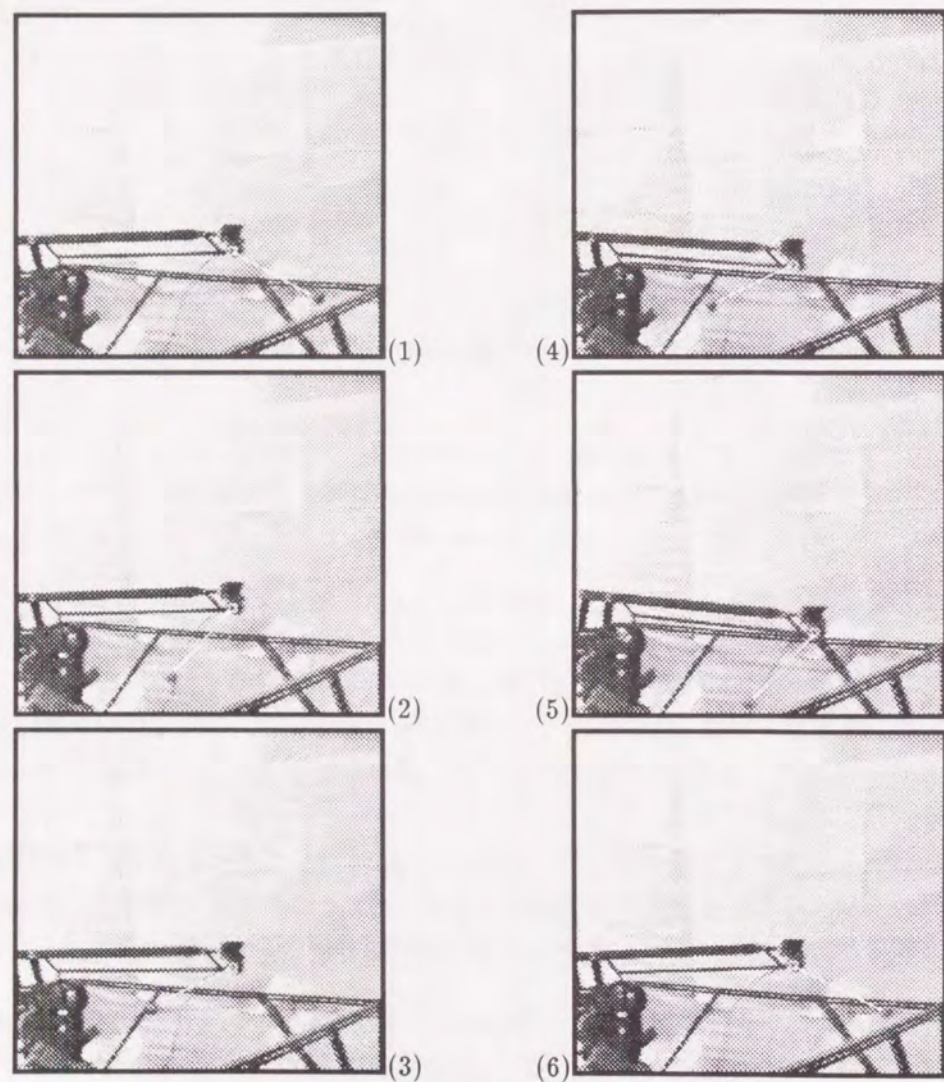


Fig.3.26: Pictures showing "cup & ball" — part 1

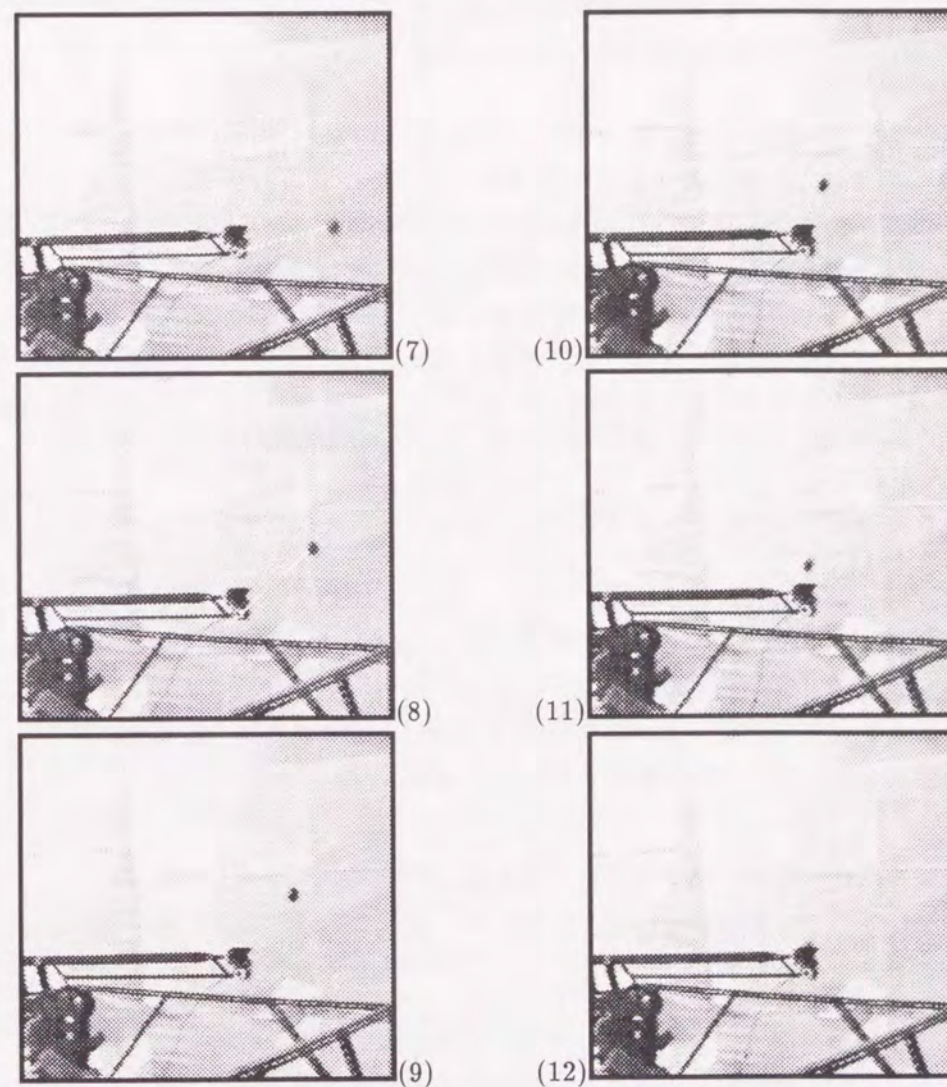


Fig.3.27: Pictures showing "cup & ball" — part 2

この間も力覚センサは張力をモニタし続けており、再び力が観測された場合、 f_x の値の正負によって mismatch アルゴリズムの (a), (b) を判断し、再び写真 1 の状態に戻る。 $f_x = 0$ の状態が続いたとき、剣玉タスクの終了となる。

写真 9 以降は、ボールが弾道軌道の頂点に達するような自由落下運動をして手先で捕獲される状態 (写真 12) を表わしたものである。

3.6 おわりに

本章では、作業環境との力学的相互干渉のある作業として剣玉を取り上げ、それをロボットがマスターする方法について考察した。

作業実行前の手先軌道を計画する段階では、最適な軌道を自らの力で獲得する方法を、作業の実行段階では、計画軌道と実軌道の誤差を自律的に解消する方法を提案し、それらの有効性を実験的に検証した。これらの方法は、人間が無意識に行なっている合理的な動作を獲得するための学習プロセスにも似ている。

第4章 結論

本研究では、近年注目されているセンサ・ベース制御に関する一アプローチとして、3つの基本機能に着目し、それぞれを具現化する方法について考察した。このアプローチは、ロボットが実行するにはかなり難度の高い「動物体を扱うタスク」に対する制御戦略を獲得するための一般性のある方法を目指したものである。以下に本研究の主な結果を示す。

- 動物体を扱うタスクの中で、「投げ」と「捕獲」に注目し、両タスクを同時に実現できるタスクとして、1本の腕で複数個のボールを順に投げて捕るジャグリングタスクと振子運動を利用してボールを投げ上げて捕獲する剣玉タスクを取り上げ、ロボットがタスク遂行能力を身に付け実行する方法を提案した。
- 軌道計画の段階では、まずロボットの手先軌道として「投げ」と「捕獲」の動作を滑らかに連結でき、周期性を持つ楕円軌道を採用し、作業環境とロボット自身の相互作用により生じる時空間的拘束を自由に設定できるパラメータ表現を導入することによって、様々な運動パターンの手先軌道を獲得する方法を提案した。

これにより、ジャグリングでは複数個の対象物に対して任意の頂点高さの弾道軌道を生成できることを、剣玉では任意の目標捕獲位置でボールを受け止めることが可能なことをシミュレーションで確認した。

- 提案した理論式によって理想的な軌道を決定しても、実際には対象物やロボットアームのダイナミクスの影響により理想通りの運動が実現されとは限らない。ジャグリングでは、視覚情報を用いてこの計画された軌道を修正し、理想的なボールの弾道軌道を得るための階層学習制御則を提案した。また剣玉に関しては、紐の張力情報を用いてモデル化されていない要素によるエネルギーロスを補う mismatch 解消アルゴリズムを提案し、それらの有効性を実験的に検証した。

- 剣玉タスクでは、力覚センサを活用して紐の長さやボールの質量などの物理パラメータを割り出し、振幅の大きさを算出して未知なパラメータを自ら獲得し、軌道計画まで至る一連のアルゴリズムを提案した。また、一定振幅の振子運動をロボット自身で実現するアルゴリズムを提案し、これらの有効性も確認した。
- ジャグリングでは、たとえ計画軌道が正確に実現されたとしても、モデル化できない動的な要因のため飛翔物体の軌道誤差が幾分か残ってしまい、タスクの継続が出来なくなることがある。この問題を克服する1つの方法として、視覚センサを用いて予測された捕獲地点に基づいて計画軌道の時間シーケンスを調節するビジュアルフィードバック則を提案し、実験によりその有効性を示した。

本論文で示した主な結果は、手先軌道を楕円に限定することによって得られたように見えるかもしれないが、本質的には運動のパラメトリックな表現から自然に導かれるものである。すなわち、適当な数のパラメータを用いて運動の時空間的パターンが表現できれば、手先軌道は楕円である必要はない。

しかし、「投げる」・「打つ」・「捕る」といった個々の動物体を扱うタスクを実現するロボットの手先軌道に、どのようなパラメータ表現が適しているのかという問題に対しては、明確な解答を与えてはいない。効率よく学習するには、ロボットにどれほどの事前知識を与えればよいのかなどの学習的観点からの考察も含め、これらの問題に関しては今後の課題としたい。

附章 A ロボットへの指令値

A.1 平行リンク型 2 自由度 DD マニピュレータの逆運動学

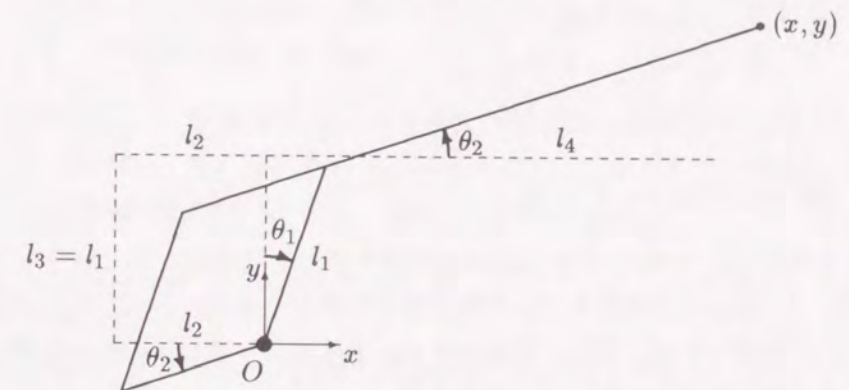


Fig.A.1: Model of DD robot arm for inverse kinematics

Fig.A.1 のように絶対座標系を設定し、ロボットアームの手先位置を (x, y) 、各関節角変位を図のように (θ_1, θ_2) で表すと、 (x, y) と (θ_1, θ_2) は、その幾何学的関係より以下のように記述できる。

$$\begin{cases} x = l_1 \sin \theta_1 + l_4 \cos \theta_2 \\ y = l_1 \cos \theta_1 + l_4 \sin \theta_2 \end{cases} \quad (\text{A.1})$$

式 (A.1) に三角関数の合成を用いると、

$$\begin{cases} \theta_1 = \cos^{-1} \left(\frac{l_1^2 - l_4^2 + x^2 + y^2}{2l_1 \sqrt{x^2 + y^2}} \right) + \tan^{-1} \frac{x}{y} \\ \theta_2 = \cos^{-1} \left(\frac{l_4^2 - l_1^2 + x^2 + y^2}{2l_4 \sqrt{x^2 + y^2}} \right) + \tan^{-1} \frac{y}{x} \end{cases} \quad (\text{A.2})$$

もしくは

$$\begin{cases} \theta_1 = \sin^{-1} \left(\frac{l_1^2 - l_4^2 + x^2 + y^2}{2l_1 \sqrt{x^2 + y^2}} \right) - \tan^{-1} \frac{y}{x} \\ \theta_2 = \sin^{-1} \left(\frac{l_4^2 - l_1^2 + x^2 + y^2}{2l_4 \sqrt{x^2 + y^2}} \right) - \tan^{-1} \frac{x}{y} \end{cases} \quad (\text{A.3})$$

が導かれる。本実験では、式 (A.3) を用いた。

各関節への指令値は、位置制御も速度制御も角速度で与えているので、式 (A.1) を微分して、

$$\begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} = \begin{pmatrix} l_1 \cos \theta_1 & -l_4 \sin \theta_2 \\ -l_1 \sin \theta_1 & l_4 \cos \theta_2 \end{pmatrix} \begin{pmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{pmatrix} \quad (\text{A.4})$$

となる (位置制御ではパルスボードが速度指令を位置指令に変換する)。ここで、右辺の行列

$$J = \begin{pmatrix} l_1 \cos \theta_1 & -l_4 \sin \theta_2 \\ -l_1 \sin \theta_1 & l_4 \cos \theta_2 \end{pmatrix} \quad (\text{A.5})$$

は、ヤコビ行列である。この行列式を求めると、

$$\det(J) = l_1 l_4 \cos(\theta_1 + \theta_2) \quad (\text{A.6})$$

となるので、 $\theta_1 + \theta_2 = \pm 90[\text{deg}]$ が特異点である。これは、本ロボットアームでは平行リンクがつぶれてアームが伸びきった状態に相当するが、手先軌道を生成する際に注意すれば、完全に回避できる。従って、ヤコビ行列の逆行列が存在して、各関節の角速度は次式のようになる。

$$\begin{pmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{pmatrix} = \frac{1}{l_1 l_4 \cos(\theta_1 + \theta_2)} \begin{pmatrix} l_4 \cos \theta_2 & l_4 \sin \theta_2 \\ l_1 \sin \theta_1 & l_4 \cos \theta_1 \end{pmatrix} \begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} \quad (\text{A.7})$$

各関節モータへの指令値 ($u_1(t), u_2(t)$) は、上式の ($\dot{\theta}_1, \dot{\theta}_2$) から

$$\begin{pmatrix} u_1(t) \\ u_2(t) \end{pmatrix} = C \begin{pmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{pmatrix} \quad (\text{A.8})$$

のように生成した (C は DD モータのサーボドライバにより決まる定数)。

A.2 学習制御則

学習制御則は従来の制御則と違い、繰り返し練習することによって、与えられた理想の運動フォーム (軌道) を自動的に獲得する方法である。本論文中でも軽く触れたが、学習には以下に示すような前提条件が公理系としてまとめられている [44]~[48]。

1. 1 回の運動 (軌道) は、ある短い有限時間 $T(> 0)$ で終わる。
2. その有限時間区間 $t \in [0, T]$ に渡って、理想の軌道 $y_d(t)$ (これを出力ともいう) が先験的に与えられている。
3. 初期化は常に一定である。
これから行なう試行回数を第 k 回目とすると、このときの初期状態 $x^k(0)$ は、運動の開始の際、常に同一状態に初期化される。

$$x^k(0) = x^0 \quad k = 1, 2, \dots \quad (\text{A.9})$$

4. 対象系のダイナミクスは、繰り返し練習中是不変である。
5. 出力 $y^k(t)$ は測定可能である。
つまり、任意の k 回目の試行の際の誤差

$$e^k(t) = y_d(t) - y^k(t) \quad (\text{A.10})$$

は常に計算可能である。

6. 次回のアクチュエータ入力 $u_{k+1}(t)$ は、簡単な再帰形式

$$u^{k+1}(t) = L(u^k(t), e^k(t)) \quad (\text{A.11})$$

で構成される ($L(*)$ は学習オペレータ)。

Fig.A.2 に学習則のブロック線図を示す。

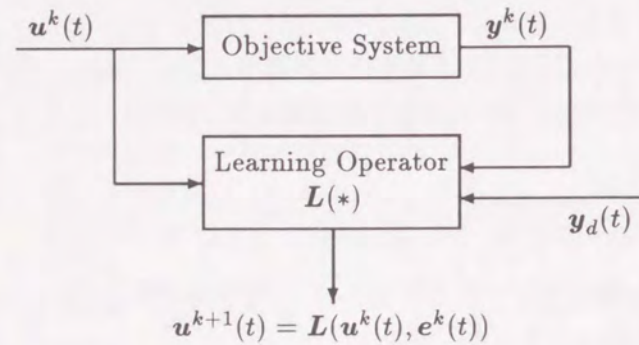


Fig.A.2: Block diagram of Learning control

本研究では、以上のカテゴリーに属する「時間進めの学習則」を用いている [49]. この学習則におけるアクチュエータ入力は以下のように与えられる.

$$\mathbf{u}^{k+1}(t) = \mathbf{u}^k(t) + \gamma \mathbf{e}^k(t+h) \quad (\text{A.12})$$

ここで、 γ は定数の学習ゲイン、 h は時間進めである.

本実験では、最初の入力を $\mathbf{u}^1(t) = (u_1(t), u_2(t))$ とし、理想軌道を $\mathbf{y}_d = (\theta_1, \theta_2)$ 、各試行毎の出力はモータの回転角度 (パルス数) とした.

附章 B 振子運動中のボールの運動方程式

マニピュレータが止まっているとき ($\ddot{x}_t = \ddot{y}_t = 0$), 振子運動しているボールの運動方程式は,

$$\ddot{\alpha}_t = -\frac{g}{l} \sin \alpha_t \quad (\text{B.1})$$

と表せる. この両辺に $\frac{d\alpha_t}{dt}$ を掛けて t について積分すると, エネルギー積分

$$\frac{1}{2} \dot{\alpha}_t^2 = E - \frac{g}{l} (1 - \cos \alpha_t) \quad (\text{B.2})$$

を得る. $\alpha_t = 0$ で位置エネルギーが零になるようにし, このときの角速度 $\frac{d\alpha_t}{dt}$ の値を $\dot{\alpha}_0$ とすれば, $E = \frac{1}{2} \dot{\alpha}_0^2 = 0$ である.

さて,

$$E = \frac{2g}{l} k^2 \quad (k > 0) \quad (\text{B.3})$$

とおき, $1 - \cos \alpha_t = 2 \sin^2 \frac{\alpha_t}{2}$ を考慮すれば,

$$\dot{\alpha}_t^2 = \frac{4g}{l} \left(k^2 - \sin^2 \frac{\alpha_t}{2} \right) \quad (\text{B.4})$$

を得る. ここで,

$$k = \sin \frac{\alpha_s}{2} \quad (0 < k < 1) \quad (\text{B.5})$$

とおくと, α_s は角速度が零になる角, すなわち α_t の振幅を表すことになる.

また,

$$\sin \frac{\alpha_t}{2} = kz \quad (k < 1) \quad (\text{B.6})$$

とおくと,

$$d\alpha_t = \frac{2k dz}{\cos \frac{\alpha_t}{2}} = \frac{2k dz}{\sqrt{1 - k^2 z^2}} \quad (\text{B.7})$$

であるから，式 (B.4) を書き換えて積分し，

$$\sqrt{\frac{g}{l}}t = \int_0^z \frac{dz}{\sqrt{(1-z^2)(1-k^2z^2)}} \quad (\text{B.8})$$

を得る．ただし，ボールが最低点を通る時刻を $t=0$ にとった．さらに， $z = \sin \phi$ とおくと，

$$\sqrt{\frac{g}{l}}t = \int_0^\phi \frac{d\phi}{\sqrt{1-k^2\sin^2\phi}} \quad (\text{B.9})$$

となる．

振れ角は $\phi = \frac{\pi}{2}$ で最大値に達するが，これは周期 T_p を用いると，時刻 $t = \frac{T_p}{4}$ のときであるから，上式は

$$T_p = 4\sqrt{\frac{l}{g}} \int_0^{\frac{\pi}{2}} \frac{d\phi}{\sqrt{1-k^2\sin^2\phi}} \quad (\text{B.10})$$

と書き直すことができる．つまり，振子運動中は上式で表されるような周期で運動することになる．

α_t が微小として $\sin \alpha_t \cong \alpha_t$ とすると，単振動の運動方程式を得ることができる．この近似を行うと，振動の周期は振幅によらず，

$$T_p = 2\pi\sqrt{\frac{l}{g}} \quad (\text{B.11})$$

で与えられる．

附章 C システムの仕様

C.1 ダイレクト・ドライブ・モータの仕様

ここでは，本研究に用いるモータの性能について説明する．使用するモータ（横河電機製のダイナサーブ DMB1030）は，外部ロータ方式を採用しており，中心軸部分は中空になっている．

Table C.1: Standard specifications of “Dynaserv(DMB1030)”

	Rated power	250	W
	Rated torque	20 (2.0)	N·m (kgf·m)
	Rated speed	2	rps
	Maximum torque	30 (3.0)	N·m (kgf·m)
	Maximum speed	2.4	rps
Positioning	Resolving capacity of encoder	655,360	p/rev
	Precision in positioning	30	sec
	Precision in repetition	± 2	sec
	Torque ripple	5	%
	Speed ripple	3	%
	Efficiency	70	%
	Weight	7	kg
	Size of an outward form(diameter×height)	φ160×118	mm×mm

C.1.1 サーボドライバの制御モード

モータ駆動用のサーボドライバには，以下に示す 4 つの制御モードがある．

1. 速度制御系

アナログ電圧 (±6V) を入力とした速度制御系である．

(a) PI 型速度制御 (高精度速度制御)

速度制御に積分・比例動作を用い、速度制御系の伝達特性が2次系になっている。ドライバー内に強い速度フィードバックループが構成されているので、外乱に強く、滑らかな速度制御が可能となっている。この方式は一般の DC/AC サーボモータの制御と同じである。

- i. P 型速度ループを帯域調整できる DC ゲインを最大イナーシャで最適に調整しておけば、イナーシャ変動がある場合に有効となる。
- ii. 速度ループ帯域のダンピングを調整できる AC ゲインがある。

(b) P 型速度制御

応答は速いがループゲインが低いので、外乱の影響を受け易い形になっている。DC ゲインのみ調整可能。

2. 位置制御系

シリアルパルス列を入力とした位置制御系である。

(a) I-PD 型位置制御 (高精度位置決め)

位置の積分フィードバックを用いているので、最終1パルスまでの位置決めが可能。また、イナーシャ変動に対しても安定した制御が可能。帯域を5から20Hzの範囲内で決めることができ、ソフトウェアサーボ演算中のデジタル積分器の出力にリミッタを設けてワインドアップ現象を防ぐスイッチが付いている。

(b) P 型位置制御 (コンプライアンス制御)

位置フィードバックに比例制御を用いて、位置決め精度は余りよくないが位置誤差に比例したトルク出力が得られる。従って、コンプライアンス制御が可能。I-PD 型と同じく帯域範囲決定・ワインドアップ防止の他に DC ゲインが調整できる。

本研究では、ジャグリングタスクの実験においては、I-PD 型位置制御を、剣玉タスクの実験においては、PI 型速度制御を用いた。

C.2 ダイレクト・ドライブ・ロボットアームの仕様

ここでは、本研究において設計試作した DD ロボット [50] を Fig.C.1 に示す。これは、平行リンク機構を持った2自由度ロボットアームである。

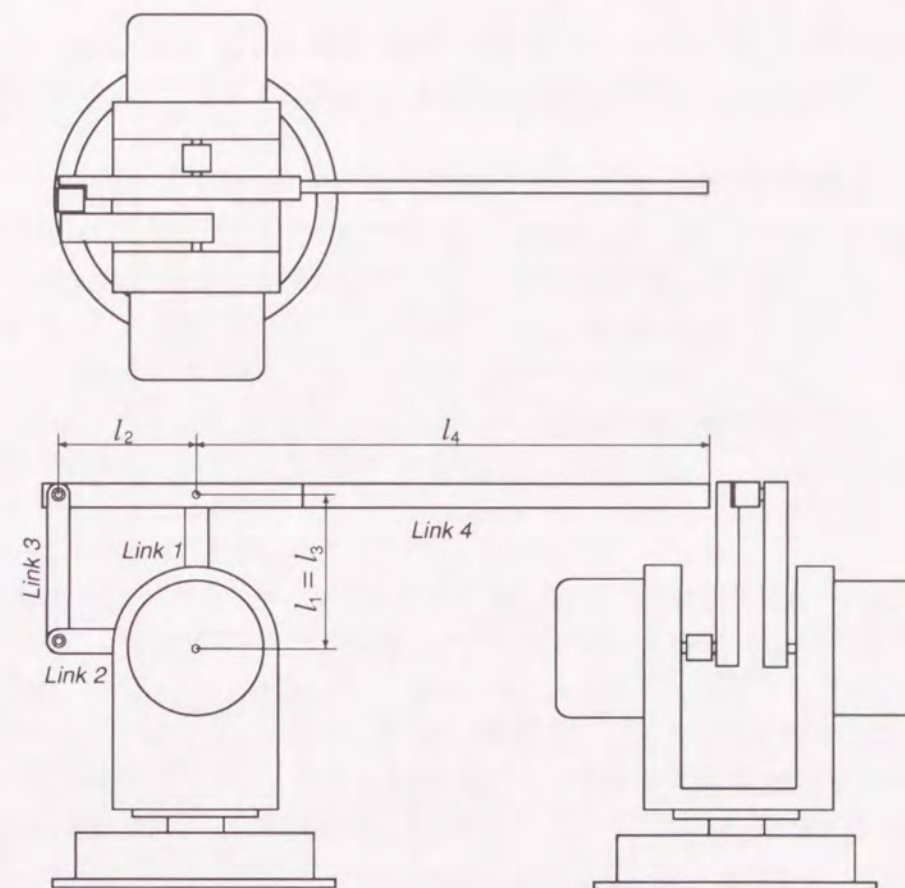


Fig.C.1: Direct-drive robot arm

Table C.2: Each link's parameter of DD robot

Length of arms [m]	l_1	l_2	l_3	l_4
		0.250	0.200	0.250
Weight [kg]	m_1	m_2	m_3	m_4
	1.090	0.750	1.160	2.050

where m_4 is a weight of link 4 ($l_2 + l_4$).

C.3 力覚センサの仕様

本実験に使用した力覚センサの仕様は **Table C.3** に示す通りである。また、力覚センサをどのような位置に付けたのかも、**Fig.C.2** に示しておく。

Table C.3: Standard specifications of a force sensor "KBL-301L"

Rated load	30	kgf
Limit of load	45	kgf
Distortional rate	$2000 \times 10^{-6} / 30$	/kg
Precision	0.5	%
Influence on temperature	below ± 0.02	%/ $^{\circ}\text{C}$
Weight	0.25	kg

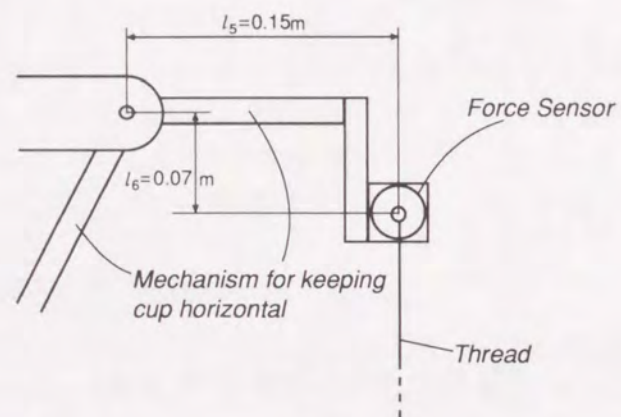


Fig.C.2: Position of a force sensor

謝辞

本研究を行うにあたり、終始一貫、御指導御鞭撻頂きました宮崎文夫教授、そして博士課程1年まで御指導頂きました有本卓教授(現 東京大学)に心より感謝致します。そして、論文作成にあたりまして、福岡秀和教授、並びに辻本良信教授から多くの御意見を賜りました。厚く御礼申し上げます。

さらに、理論的な議論に加わって頂きました升谷保博助手、実験におきまして貴重な御助言を頂きました林清重技官、また、実験用ロボットの設計製作に御尽力を頂きました三菱電機(株)の増田隆広氏、藪内賀義氏、武田宗久氏、坪井光明氏に深く感謝の意を表します。

現在に至るまで、ロボット立ち上げ時におきましては野村秀樹君(現 豊田中央研究所)、浪花智英君(現 東京大学助手)、人間のタスク解析における被験者として山本一之君(現 三菱電機(株))、新井良和君(大学院生)、ジャグリングシミュレーションでは児玉正幸君(現 マツダ自動車(株))、その初期実験では澤井恒義君(現 川崎重工業(株))、剣玉の実現に関しては佐藤守一君(現 豊田中央研究所)諸氏に御協力頂き、実際の実験では、藤田政宜君(大学院生)、渡辺洋君(大学院生)、吉井芳徳君(学部生)に御協力頂きました。ここに心より御礼申し上げます。また、本研究のために御協力頂いた研究室の諸氏に深く感謝致します。

最後に、物心両面で激励して下さいました家族や先輩後輩、友人達に心から感謝致します。

阪口 健

1993年1月

参考文献

- [1] W. Thomas Miller III: "Sensor-Based Control of Robotic Manipulators Using a General Learning Algorithm," *IEEE Journal of Robotics and Automation*, pp. 157-165, Vol.RA-3, No.2, April, 1987.
- [2] W. Thomas Miller III: "Real-Time Application of Neural Networks for Sensor-Based Control of Robots with Vision," *IEEE Transactions on System, Man, and Cybernetics*, pp. 825-831, Vol.19, No.4, 1989.
- [3] S. Doubowsky and Z. Shiller: "Optimal Dynamic Trajectories for Robotic Manipulators," *Proc. of RoMansy, 5th CISM-IFTOMM Symp.*, pp. 133-143, 1984.
- [4] K. G. Shin and N. D. McKay: "Minimum-Time Control of Robotic Manipulators with Geometric Path Constraints," *IEEE Trans. on Automatic Control AC-30-6.*, pp. 531-541, 1985.
- [5] 尾崎弘明, 秀田保, 山本元司, 毛利彰: "空間経路が指定されるマニピュレータ動作の時間短縮", 計測自動制御学会論文集, pages 1074-1080, 1986.
- [6] 狩野弘之, 高山訓治: "最小加速度規範によるマニピュレータの滑らかな軌道制御", 日本ロボット学会誌, pages 3-10, Vol.7, No.1, 1988.
- [7] 永井潤, 鈴木達也, 大熊繁: "運動エネルギーに着目したマニピュレータの実時間軌道計画法", 第9回日本ロボット学会学術講演会予稿集, pages 833-836, 1991.
- [8] M. H. Raibert: "Legged Robots That Balance," *The MIT Press*, 1986.
- [9] R. L. Andersson: "A Robot Ping-Pong Player: Experiment in Real-Time Intelligent Control," *The MIT Press*, 1987.

- [10] 榎木哲夫: “学習とリアクティブプランニング”, メカトロニクス研究会—インテリジェント・マシン・システム研究部会—報告書, 財団法人 京都高度技術研究所, 1992.
- [11] L. P. Kaelbling: “An Architecture for Intelligent Reactive Systems,” *Reasoning about Actions and Plans in M. Georgeff and A. Lansky eds.*, San Mateo, California, Morgan Kaufmann, pages 394–424, 1987.
- [12] P. Agre and Pengi D. Chapman: “An Implementation of a Theory of Activity,” *Proceedings of 6th AAAI-87*, pages 268–272, 1987.
- [13] Claude E. Shannon: “Unified Field Theory of Juggling” (in “PROFILE/CAREERS”), *IEEE SPECTRUM*, page 75 (72–76), April, 1992.
- [14] M. Bühler and D.E. Koditschek: “Robotics in an Intermittent Dynamical Environment: A Prelude to Juggling,” *A Talk Delivered at the 26th IEEE Conference on Decision and Control*, pages (paper available from authors—not in proceedings), Los Angeles, California, December 11, 1987.
- [15] M. Bühler, D. E. Koditschek, and P. J. Kindlmann: “Planning and Control of Robotic Juggling Tasks,” *Preprint of the 5th International Symposium of Robotics Research*, pages 270–281, Tokyo, Japan, August 28–31, 1989.
- [16] A. A. Rizzi and D. E. Koditschek: “Progress in Spatial Robot Juggling,” *Proceedings of the 1992 IEEE International Conference on Robotics and Automation*, pages 775–780, Nice, France, May, 1992.
- [17] Christopher G. Atkeson and Joseph McIntyre: “Robot Trajectory Learning Through Practice,” *1989 IEEE International Conference on Robotics and Automation*, pages 1737–1742, San Francisco, California, April 7–10, 1986.
- [18] Christopher G. Atkeson, Eric W. Aboaf, Joseph McIntyre, and David J. Reinkensmeyer: “Model-Based Robot Learning,” *Robotics Research 4*, pages 103–110, MIT Press, 1988.
- [19] Eric W. Aboaf, Steven M. Drucker, and Christopher G. Atkeson: “Task-Level Robot Learning: Juggling a Tennis Ball More Accurately,” *1989*

- IEEE International Conference on Robotics and Automation*, pages 1290–1295, Scottsdale, Arizona, May 14–19, 1989.
- [20] Christopher G. Atkeson: “Using Locally Weighted Regression for Robot Learning,” *1991 IEEE International Conference on Robotics and Automation*, pages 958–963, Sacramento, California, April, 1991.
- [21] Chae H. An, Christopher G. Atkeson, John M. Hollerbach 著, 松本治彌 訳: “モデルに基づくロボットマニピュレータの制御”, 実教出版株式会社, 1991.
- [22] 黒須茂: “人間によって操作される一力学系 (ヨーヨー操作) の運動解析”, 計測自動制御学会論文集, pages 277–282, Vol.14, No.3, 1978.
- [23] 竹中一起: “視覚をもつマニピュレータの動的制御 (ロボットによる「けんだま」の実現)”, 日本機械学会論文集 (C 編), pages 2046–2053, Vol.50, No.458, 1984.
- [24] 田島不二夫, 下山勲, 三浦宏文: “マニピュレータによるこまの運動制御”, 第 4 回日本ロボット学会学術講演会予稿集, pages 63–64, 1986.
- [25] 山口英郎, 藤井克彦: “ヒューマン・コントローラの学習過程とそのモデル化—Gradual Mode による学習制御過程—”, 第 10 回計測自動制御学会学術講演会予稿集, pages 32–40, 1971.
- [26] “Skill Learning and Sensory Optimization Help Robot Play Ball,” *Robotics and Machine Perception*, July 1992.
- [27] Gelfand et. al: “Acquisition of Automatic Activity Through Practice: Changes in Sensory Input”, *Proceedings. AAAI-92*, San Jose, July, 1992.
- [28] Jhon T. Feddema, Owen R. Mitchell: “Vision-Guided Servoing with Feature-Based Trajectory Generation,” *IEEE Transactions on Robotics and Automation*, pages 691–700, Vol.5, No.5, October, 1989.
- [29] Jhon Lloyd and Vincent Hayward: “Real-Time Trajectory Generation Using Blend Functions”, *1991 IEEE International Conference on Robotics and Automation*, pages 784–789, Sacramento, California, April, 1991.

- [30] Yu Wang: "Dynamics and Planning of Collisions in Robotic Manipulator", *1991 IEEE International Conference on Robotics and Automation*, pages 478-483, Sacramento, California, April, 1991.
- [31] 大築立志: "「たくみ」の科学", 朝倉書店, 1988
- [32] 山本正樹, 中井泰明, 村上安範: "トランスピュータ入門", 日刊工業新聞社, 1990.
- [33] 木村博昭, 福島實: "トランスピュータによる並列処理", 海文堂, 1990.
- [34] 谷内田正彦: "ロボットビジョン [人工知能シリーズ 11]", 昭晃堂, 1990.
- [35] 奥村晴彦: "C 言語による最新アルゴリズム事典", 技術評論社, 1991.
- [36] Peter Dyer and Stephen R. McPeaynolds: "The Computation and Theory of Optimal Control," *Academic Press*, 1965.
- [37] 嘉納秀明: "システムの最適理論と最適化"(コンピュータ制御機械システムシリーズ 3), コロナ社, 1987.
- [38] 加藤寛一郎: "工学的最適制御非線形へのアプローチ", 東京大学出版会, 1988.
- [39] T. Manabe, F. Miyazaki: "Learning Control based on Local Linearization by using DFT," *Proc. of the 4th IEEE/RSJ International Workshop on Intelligent Robots and Systems '91, Intelligence for Mechanical Systems*, pages 639-646, Osaka, Japan, November 3-5, 1991.
- [40] マルセル・ボル: "万能数値表", 白水社.
- [41] 森口繁一, 宇田川圭久, 一松信: "数学公式 III —特殊関数—", 岩波全書, 1960.
- [42] H. Asada: "Robot Analysis and Control", *The MIT Press*, 1987.
- [43] 美多勉, 大須賀公一: "ロボット制御工学入門", コロナ社, 1989.
- [44] 有本卓: "ロボットの力学と制御", 朝倉書店, 1991.

- [45] 川村貞夫, 宮崎文夫, 有本卓: "学習制御方式のシステム論的考察", 計測自動制御学会論文集, pages 445-450, Vol.21, No.5, 1985.
- [46] S. Kawamura, F. Miyazaki, and S. Arimoto: "Applications of learning method for dynamic control of robot manipulators," *Proc. 24th IEEE Conference on Decision and Control*, pages 1381-1386, Ft. Lauderdale, Florida, 1985.
- [47] S. Kawamura, F. Miyazaki, and S. Arimoto: "Realization of Robot Motion Based on a Learning," *IEEE Trans. on SMC*, pages 126-134, Vol.18, No.1, 1988.
- [48] 有本卓: "学習制御の理論", 日本機械学会誌, pages 2-8, Vol.93, No.856, 1990.
- [49] 内山勝: "試行による人工の手の高速運動パターンの形成", 計測自動制御学会論文集, pages 706-712, Vol.14, No.6, 1978.
- [50] 浅田春比古: "DD ロボット入門—開発史・機構・設計・制御技術", 工業調査会, 1987.

関連論文

(1) 学術論文集

1. T. Sakaguchi, Y. Masutani, F. Miyazaki: "A Study on Juggling Task," *Proc. of the 4th IEEE/RSJ International Workshop on Intelligent Robots and Systems '91, Intelligence for Mechanical Systems*, pages 1418-1423, November 3-5, 1991, Osaka, Japan, International House Osaka.
2. T. Sakaguchi, S. Sato, F. Miyazaki: "A Study on Tasks of Dealing with Moving Objects," *IMACS/SICE International Symposium on Robotics, Mechatronics and Manufacturing Systems '92 Kobe*, pages 1418-1423, September 16-20, 1992, Kobe, Japan.
3. T. Sakaguchi, S. Sato, F. Miyazaki: "Robot Motion Planning for Moving Objects," *IFAC 12th World Congress International Federation of Automatic Control '93 Sydney*, July 19-23, 1993 Sydney, Australia (submitted).
4. T. Sakaguchi, M. Fujita, H. Watanabe, F. Miyazaki: "Motion Planning and Control for a Robot Performer," *IEEE International Conference on Robotics and Automation '93*, May 2-7, 1993, Atlanta, America (submitted).
5. 佐藤 守一, 阪口 健, 升谷 保博, 宮崎 文夫: "作業環境との相互干渉を有するタスクの自律生成—ロボットによる剣玉の実現—", 日本機械学会論文集, 59巻 558号 C編, 平成5年2月.
6. 阪口 健, 藤田 政宜, 宮崎 文夫: "動物体を扱うロボットの運動計画と制御", (日本機械学会論文集投稿中).

(2) 学術研究集會会談録

1. 阪口健, 升谷保博, 宮崎文夫: “ロボットにおける動力学的な器用さに関する一考察”, 計測自動制御学会第2回ロボティクス・自動化システムシンポジウム講演論文集, pages 69-72, 平成2年4月10日,11日, 東京, 国立教育会館.
2. 阪口健, 児玉正幸, 升谷保博, 宮崎文夫: “ジャグリングタスクにおける運動パターンの生成”, 日本ロボット学会第1回ロボットシンポジウム予稿集, pages 133-138, 平成3年5月9日,10日,11日, 仙台, 東北大学青葉記念会館.
3. 佐藤守一, 阪口健, 升谷保博, 宮崎文夫: “忘却因子を付加した学習制御の実験的考察”, 第35回システム制御情報学会講演論文集, pages 137-138, 平成3年5月22日,23日,24日, 京都, 京都私学会館.
4. 阪口健, 児玉正幸, 升谷保博, 宮崎文夫: “ジャグリングタスクにおける手先軌道の獲得”, 日本機械学会ロボティクス・メカトロニクス講演会'91講演論文集, pages 25-28(Vol.A), 平成3年6月7日,8日, 名古屋, 名古屋国際会議場.
5. 阪口健, 升谷保博, 宮崎文夫: “ジャグリングタスクにおける運動プランニング”, 日本機械学会第69期全国大会講演会講演論文集, pages 618-620(Vol.C), 平成3年10月16日,17日,18日, 名古屋, 名古屋大学工学部.
6. 渡辺洋, 阪口健, 宮崎文夫: “作業環境との相互干渉を有するタスクの自律生成—ロボットによる剣玉の実現—”, 日本機械学会関西支部第251回講演会講演論文集, 平成4年11月28日,29日, 大阪, 大阪電気通信大学工学部.
7. 藤田政宜, 阪口健, 宮崎文夫: “動物体を扱うタスクに関する一考察—学習による軌道生成と視覚を用いた軌道修正—”, 日本機械学会関西支部第251回講演会講演論文集, 平成4年11月28日,29日, 大阪, 大阪電気通信大学工学部.

