Racket Control and Spinning Ball Measurement for Table Tennis Robot

Dissertation

Chunfang LIU

Department of Mechanical Science and Engineering

Graduate School of Engineering

Nagoya University

Nagoya, Japan

2013

Contents

1 Introduction **1**
- 1.1 Background of the Research on Table Tennis Robot . . . 1
- 1.2 The State of the Art on Table Tennis Robot . . . 1
  - 1.2.1 System configuration . . . 2
  - 1.2.2 Vision system . . . 4
  - 1.2.3 Control system . . . 6
- 1.3 Demands for Table Tennis Robot . . . 9
- 1.4 The Purpose of the Thesis . . . 10
- 1.5 Organization of the Thesis . . . 13

2 Online Measuring the Ball's State with High Speed Camera **15**
- 2.1 Preliminary . . . 16
- 2.2 Estimation of the Translational Velocity . . . 18
- 2.3 Estimation of the Rotational Velocity (Method 1) . . . 21
- 2.4 Estimation of the Rotational Velocity (Method 2) . . . 24
  - 2.4.1 Construction of the estimated $(k+1)$th frame of image data . . . 25
  - 2.4.2 Estimation of the most plausible rotational velocity . . . 27
- 2.5 Experimental Results . . . 28
  - 2.5.1 Experiments with a rotation machine . . . 29
  - 2.5.2 Experiments with a catapult machine . . . 31
- 2.6 Summary . . . 34

3 Physical Models for Predicting the Motion of the Ball **37**
- 3.1 Preliminary . . . 37
- 3.2 Aerodynamics Model . . . 38
- 3.3 Table Rebound Model . . . 39
- 3.4 Racket Rebound Model . . . 42
- 3.5 Experimental Verification of the Three Models . . . 45
  - 3.5.1 The case of the aerodynamics model . . . 45
  - 3.5.2 The case of the table rebound model . . . 49
  - 3.5.3 The case of the racket rebound model . . . 50
- 3.6 Summary . . . 51

4 Racket Control Method **55**
- 4.1 Preliminary . . . 56
- 4.2 Properties of the Physical Models . . . 57
  - 4.2.1 Solution to the racket rebound model . . . 57
  - 4.2.2 Two-point boundary value problem for the aerodynamics model . . . 60

4.3 Racket's Striking Posture and Velocity . . . 62
4.4 Numerical Simulations . . . 65
4.5 Experimental Results . . . 66
4.5.1 Experimental settings . . . 66
4.5.2 Some results . . . 70
4.6 Summary . . . 72

5 Conclusions **79**
5.1 Summary . . . 79
5.2 Further Research . . . 81

A Table Tennis System **83**
A.1 Table, Ball, Racket . . . 83
A.2 High and Middle Speed Cameras . . . 84
A.2.1 High speed camera . . . 84
A.2.2 Middle speed camera . . . 85
A.3 7-DOF Manipulator . . . 86
A.4 Others . . . 86
A.4.1 Automatic catapult machine . . . 86
A.4.2 PCI memolink . . . 87

B Robot Kinematics and Cameras' Calibrations **89**
B.1 Robot Kinematics . . . 89
B.1.1 Frame arrangement of the manipulator . . . 89
B.1.2 Forward kinematics . . . 89
B.1.3 Jacobian matrix . . . 92
B.2 Cameras' Calibrations . . . 96
B.2.1 Pinhole camera model . . . 96
B.2.2 Calibration of high speed camera . . . 97
B.2.3 Calculation of the 3D coordinate from 2D image coordinates . . . 98
B.2.4 Calibration of middle speed camera . . . 99

Bibliography **101**

Acknowledgements **107**

Published Papers **109**

Chapter 1

Introduction

1.1 Background of the Research on Table Tennis Robot

The science technology rapidly develops over the recent thirty years, especially in computer vision, artificial intelligence, robot hardware which greatly promote the development of robotics. Now, the conception of "robot" is not limited to industrial robot which can complete only fixed tasks in a structured environment. Various kinds of robots that can work in dynamic or unstructured environments have been studied such as the space robot, underwater robot, which work in dangerous environments and home service robot, entertainment robot, which appear in human's daily life. The robot working in a dynamic environment has to *online* acquire and perceive the information of the surrounding environment by using external sensors e.g. force sensor, tactile sensor, vision sensor. Moreover, it has to predict the future development trend of the object and then quickly makes a decision for planning the motion of the robot's mechanical body.

"Table tennis robot" is a representative of them, which means that the robot can play table tennis. In order to do that, the robot has to sense a coming ball's trajectory; make a reasonable judgment about how to return the ball and realize a flexible striking, which is a typical *real-time* and *intelligent* process.

The research about "table tennis robot" already has nearly thirty years' history since Prof. John Billingsley from Portsmouth University in England initiated the table tennis robot competition in 1983. After this year, the table tennis robot research has gradually become a research hotpot by reason of its difficulty and challenge in both real-time and intelligent design. In the year of 1988, there was an important breakthrough that Russell Anderson at AT&T Bell Laboratory presented the first table tennis robot (a 6 DOF PUMA 260 arm) which was capable to play against humans [1, 3]. Follows include the presented table tennis systems in Japan [14], Swiss [11], Spain and China, which show that the robot has already been able to play continuously longer time with a human opponent who is beginner skill level. However, the robot still can not control a ball returned by the human opponent with fast translational speed or high rotational speed. In other words, "triumph over humans" in a competition is still a long way to go for the research of table tennis robot.

1.2 The State of the Art on Table Tennis Robot

Imagine that if you are playing table tennis, what will be done in turn: seeing by eyes where the ball is coming and what kind of ball it is; thinking by the brain how to return the ball; moving

the hand with a racket to strike the ball. In order to accomplish the task of playing table tennis, the robot should be provided with systems which are similar to our hand-eye coordination system. Summarizing the previous research on table tennis robot, it consists of three basic subsystems: vision system, control system and mechanical system, which correspond to the human eye, brain and hand, respectively. Their relation can be expressed by using Figure 1.1.

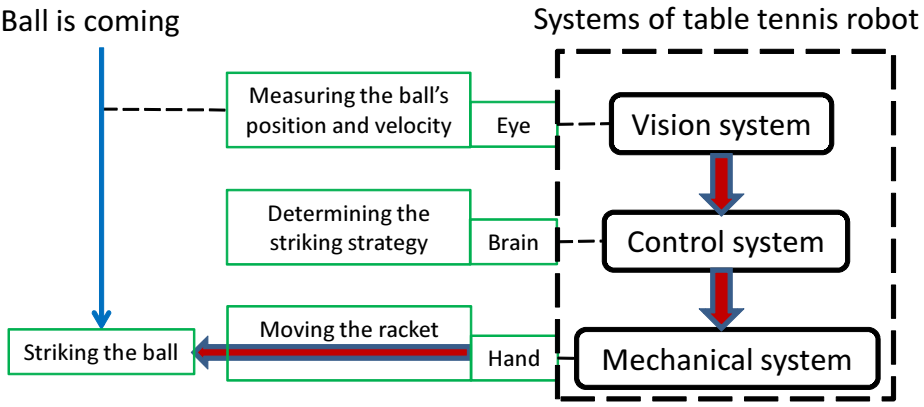


Figure 1.1: Construction of table tennis system

1.2.1 System configuration

There have been many system configurations developed for playing table tennis. As for the vision system, it may include one, two or four cameras and an image processor. The control system usually is a PC. The mechanical system consists of a manipulator of which the end-effector is the racket which is similar to human hand holding a racket. However, there is not a uniform structure for the mechanical system. Let us see some typical table tennis systems presented.

Actually, from 1985 to 1988, the contests for table tennis robot had been held four times in Europe. Several table tennis systems were proposed at that time. However, due to the under-developed real-time vision system and motion servo technology, the early systems complied with the 20 rules made by Prof. John Billingsley for improving the success rate of the striking. For example, the size of the table was limit in 0.5 m $\times$ 2 m. Figure 1.2 shows a representative system in the eighties of last century, which is developed by Andersson et al. in USA. The robot is a 6 DOF PUMA 260 arm with a stick and four cameras are used to detect the incoming ball's state [1, 2, 3, 4]. Under the robot table tennis rules suggested by Prof. John Billingsley, this system realized playing against humans for the first time.

In 2003, L. Acosta [6] from Spain constructed a low-cost robot which used two paddles as shown in Figure 1.3. The robot totally has 5 degrees of freedom and only one single camera is applied to measure the ball's 3D position with the geometrical relationship between the ball and the shadow. It showed that the robot could play well if the speed of the ball was small ($<$ 5 m/s). However, the table size was only half of the standard.

Figure 1.4 is the system configuration developed by Miyazaki et al. [27, 28, 30], which has 4 degrees of freedom that 2 DOF are able to move in the horizontal plane and 2 DOF are able to determine the attitude of the racket. A stereo vision system (Quick MAG) measures the ball's position from the sensed images in every 1/60 sec. This robot system can return the ball to a desired position on the table with a specified flying duration.

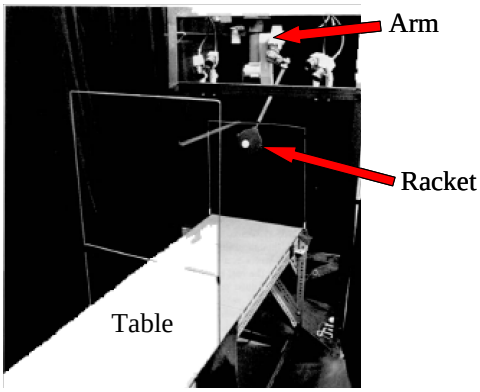


Figure 1.2: 6-DOF structure (Andersson et al. in 1988)

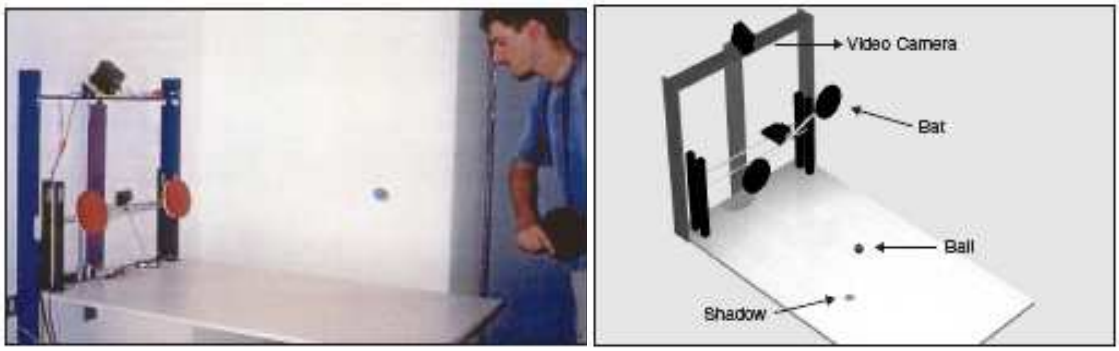


Figure 1.3: 5-DOF structure (Acosta et al. in 2003)

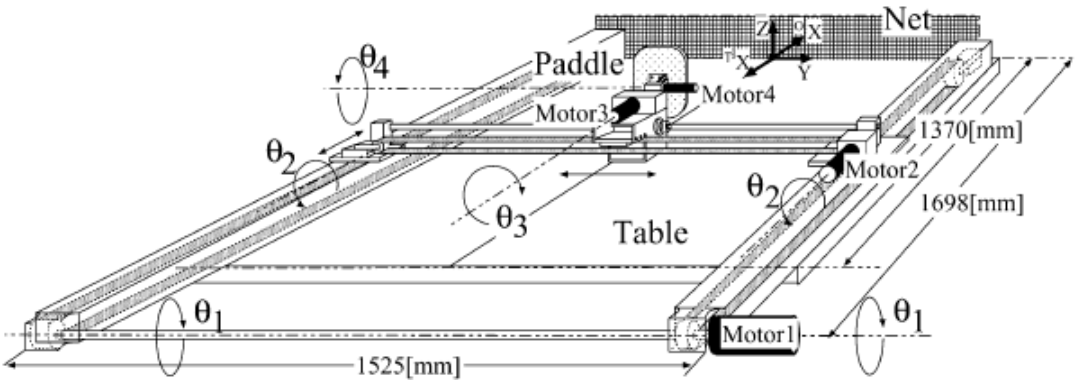


Figure 1.4: 4-DOF structure (Miyazaki et al. in 2005)

The system in Figure 1.5 is based on a selective compliant robot arm (SCARA) which has 4 degrees of freedom, i.e., three degrees for moving in x, y and z axes and one degree for performing the wrist rotation [48]. Two 60 Hz cameras are applied to sense the ball's images and Intel's Integrated Performance Primitives (IPPs) are used as the basis for the image-processing software.

Figure 1.6 demonstrates a 5-DOF table tennis robot developed by Institute of Automation, Chinese Academy of Sciences in China which can move the racket in x-y-z axis as shown in (a) and rotate the racket around the pitch and yaw angles as shown in (b).

In fact, 5-DOF mechanical system is required for robot playing table tennis (3 degrees of freedom for moving the racket's position in x-y-z axis and 2 degrees of freedom for rotating the

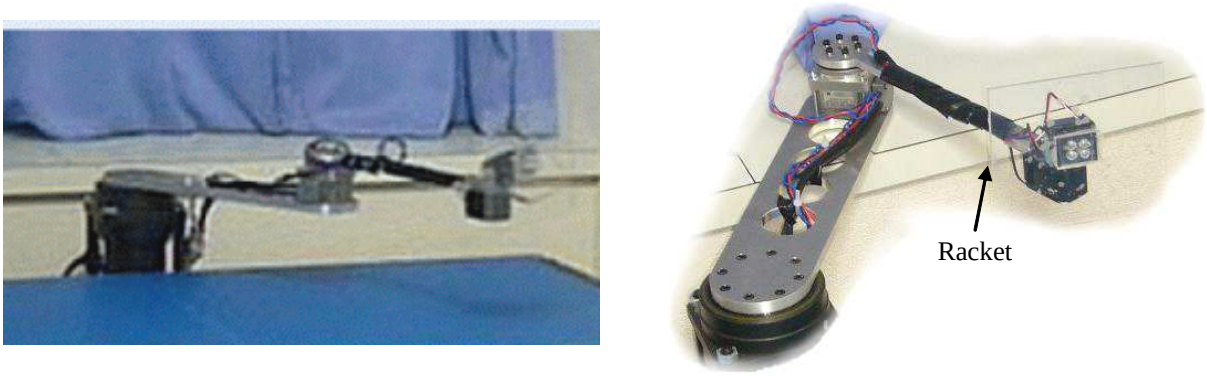
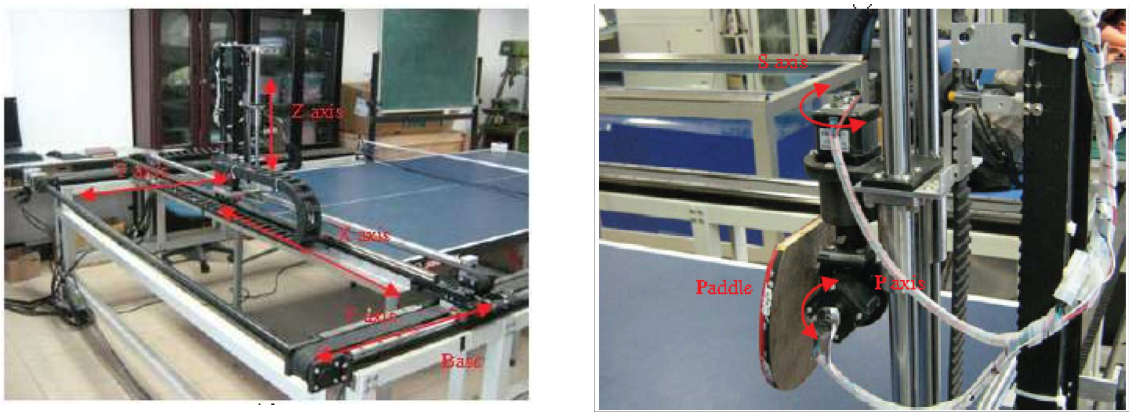


Figure 1.5: 4-DOF structure (Liu et al. in 2008)



(a) Translational X, Y, Z axis (b) Rotational S, P axis

Figure 1.6: 5-DOF structure (Yang et al. in China in 2010)

racket's directions (the yaw and the pitch angles)). However, redundant degrees of freedom can increase the flexibility of the arm's striking motion, which are capable to avoid singularity in the motion planning of the manipulator [43, 53]. Therefore, in this thesis, a 7-DOF manipulator is used for the table tennis game. This thesis focuses on the vision system that how to measure the ball's position and translational/ rotational velocities and the control system that how to control the racket for returning the ball with a desired locus. Therefore, in the next two sections, the research states about the vision system and the control system will be discussed in detail.

1.2.2 Vision system

The vision system seems like human eye, which is used for detecting the flying ball's 3D position and velocity in some frames. The robot uses this information for predicting the ball's flying trajectory and controlling the racket for striking the ball with a desired manner. Therefore, the task of the vision system has a considerable significance since it directly decides whether or not the following predicting task and controlling task can be performed well.

The vision system consists of cameras and image processor (PC or special processor). Table 1.1 shows the representative developments of the vision system of table tennis robot from 1988.

The number of camera can be one, two (stereoscopic vision) or more. In the single camera system, only one image needs to be transmitted and processed, which simplifies the procedures and lowers the cost. However, it needs some auxiliaries such as light, shadow, which has a strict requirement on the working environment. More cameras system (Andersson used 4 video

cameras) can enlarge the range of the visual field and improve the detection accuracy. However, the image processing becomes complicated which increases the computing time and also there are problems of cameras' calibration and synchronous of data acquisition. Therefore, most of the presented vision systems use stereoscopic vision (two cameras) system for detection.

Table 1.1: The development of the vision system for the table tennis robot

Time (year)	Researcher	Number of camera, frame rate, processor
1988	Anderson [4]	Four video cameras, 60Hz, MC68020 processor
1993	Fassler [11]	Two cameras, 50Hz, MC68020, MC68000 processor
2003	L. Acosta [6]	One camera, 40Hz, PC
2005	Miyazaki [27]	Two cameras, 60Hz, Quick MAG
2005	K.P. Modi [29]	A USB communicative camera, 15Hz, work station
2005	Y. Zhang [58]	One camera, 60-89Hz, PC
2006	Y. Zhang [58]	Two cameras, 60-89Hz, PC
2007	Quanta-View Inc. [48]	Two cameras (1394 bus), 60Hz, Intel Xeon processors
2008	Y. Zhang [58]	Two cameras, 60-89Hz, PC
2010	Z. Zhang [59]	Two cameras (DSP, FPGA), 250Hz, PC

The ball's state includes three basic factors: position, translational velocity and rotational velocity, and both the translational velocity and rotational velocity of the ball play important roles in winning a table tennis match [49]. However, until now, almost all the presented vision systems for the table tennis robot can only detect the incoming ball's position and translational velocity. The ball's rotational velocity is ignored. Zhang et al. [59] shows a method of tracking a flying ball in which the area of the ball image in every frame of image data is detected by means of the frame difference and the search region method. However, their research is under the assumption that the ball's rotational velocity is in the range of 0$\sim$20 rad/s. ***One important reason is that there is no suitable vision sensor and algorithm of measuring the ball's rotational velocity for using in the high real-time demand situation of table tennis***. When a human being plays a table tennis, the translational velocity of the ball is in the range of 4$\sim$30 m/s, nevertheless, in the robot case, the ball's translational velocity generally is smaller than 7 m/s since if it is too fast, the response time left for the robot is too short to be used for hitting the ball. Actually, if the translational velocity is 5$\sim$7 m/s, the ball's flying duration from the position being detected to the striking position is about 400$\sim$600 ms and the robot usually needs about 300$\sim$500 ms for moving the manipulator from the waiting position to the striking position. Therefore, the vision system must complete the detection task in just a few milliseconds (the time from sensing the image to obtaining the measuring results should be smaller than 80 ms), which is a great challenge for both the hardware of the vision sensor and the detection algorithm.

However, it is generally well known that since the table tennis ball is "small" and "light", the rotation of the ball is an essential factor for various skills and tactics of table tennis such as chop, drive, loop, push and so on [45, 50]. This means that the robot can never compete with a man who has high skill level if the robot can not control a spinning ball. Moreover, for a spinning ball, the rotation is an important factor that influences the ball's flying trajectory, which has been shown in [37] by using the aerodynamics model and some experimental data and in [18] by comparing the observed real trajectory with the theoretical spin-free ball trajectory of drive and push shots. Therefore, it is necessary to consider the influence of the rotational velocity for a spinning ball.

In a word, in order to compete with a human player in a table tennis game, the robot must

be able to master how to detect the incoming ball's rotational velocity and how to control the returning ball's rotational velocity.

There has been some research about measuring the rotation velocity of a ball. Watanabe et al. [51] propose a multi-target tracking algorithm for a high speed vision chip and as one of its applications, they show some experimental results on measuring a rotational velocity of a ball in real-time; the used algorithm is that by the proposed method, multi-feature on the ball's surface is tracked in two-dimensional (2D) image space and also the feature matching between the kth frame and the $(k+1)$th frame is carried out. Then three-dimensional (3D) position of each feature's centroid in each frame is reconstructed to estimate the rotational velocity by using the consecutive features' 3D positions. The results are excellent; however, unfortunately the experimental results show that the measurable range of the rotational speed is almost less than 1200 rpm, which is not satisfactory because the rotational speed reaches about 3000 rpm in the case of table tennis balls (The rotational velocity of the ball varies with different table tennis techniques: 40∼85 c/s (2400∼5100 rpm, back spin), 30∼80 c/s (1800∼4800 rpm, smash), 50∼75 c/s (3000∼4500 rpm, top spin serve), 90∼160 c/s (5400∼9600 rpm, high loop), 100∼160 c/s (6000∼9600 rpm, forward loop) [16]).

As the image registration method [9, 19, 57] without feature matching, the following method is used in [44] and [46]; $(k+1)$th frame of image data are estimated by using a candidate of rotational velocity and the sensed kth frame of image data. Then the most plausible rotational velocity is searched by comparing the estimated image data with the sensed image data in every frame.

Reference [44] shows the measurement of a baseball's spin in *off-line* by using camera video, where image processing is focused on how to separate the ball image from the background and how to eliminate the effect of the outdoor light condition. In [46], the rotational velocity of table tennis ball is measured. However, the frame rate used in the experiment is 500 Hz which is not fast enough to realize the table tennis playing robot. In fact, when the rotational speed is 3000 rpm, the ball rotates 36 deg in the interval of the frame rate, which means the consecutive frames of image data are too coarse for the image registration method to estimate the rotational velocity. In addition, unfortunately the estimated results in [46] do not show clearly how well the proposed method can work.

1.2.3 Control system

When playing table tennis, the control system works like player's brain which has to analyze the coming ball's trajectory, has to think strategies for returning the ball and has to manipulate the robot's arm. Hence, it is the core system for reflecting the robot's intelligence. Precisely, "the task of the robot brain" can be divided as the following ones in turn:

a) predicting the incoming ball's trajectory;

b) determining a suitable posture and velocity for the racket at the striking time;

c) planning a reference trajectory of the manipulator with the racket such that the returned ball would reach a given destination on the opponent table;

d) controlling the robot according to the motion planning in c).

This section will mainly focus on the tasks a)–c) as shown in Figure 1.7.

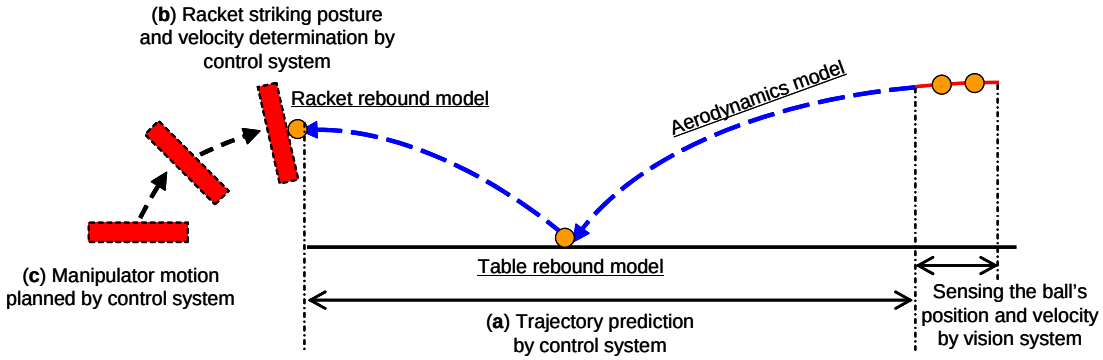


Figure 1.7: Tasks of the control system

a) predicting the incoming ball's trajectory

After acquiring the detected incoming ball's information from the vision system, the control system begins to predict the ball's trajectory. There are two purposes of the trajectory prediction: 1) for determining a striking position and a striking time; 2) for obtaining the ball's state just before striking. Usually, the striking position will be firstly decided under some required conditions on the predicted trajectory. For example, if the ball's predicted position in x-axis or z-axis satisfies a required value, or if the predicted translational velocity of the ball in z-axis is near to 0 after rebounding from the table, or if both the ball's predicted position and velocity accord with a condition.

Let us imagine the ball's motion before striking: after shot by the opponent, the ball flies in the air, rebounds from the table and flies in the air again. Therefore, in order to plan the ball's flying trajectory, it is necessary to simulate the ball's flight in the air and the ball's rebound from the table (See Figure 1.7). There have been many aerodynamics models and table rebound models presented by the previous table tennis research [6, 58, 59]. However, those models only consider the translational velocity of the ball. The rotational velocity is totally ignored. Hence, those models are not suitable for a spinning ball.

b) determining the racket's striking state for returning the ball to a reference

Consider what the human player will do next after he decides the special point to strike the ball back. Surely, he will think how to strike the ball with the racket in order for the returned ball to achieve a desired position on the opponent court. Maybe he also wishes the returned ball can reach a desired velocity at that position. Similarly, the robot player has to determine the racket's velocity and posture at the striking time for returning the ball to a reference position. This is the racket control problem.

Generally, the problem can be described with Figure 1.8, which determines the racket's striking velocity and posture through the racket rebound model and the aerodynamics model under the two informations: the ball's velocity just before the striking and the ball's desired landing state.

Until now, there have been many researches on controlling the racket for table tennis robot returning balls. Acosta et al. [6] determined the posture of the racket under the assumption that when the racket struck the ball, the reflection angle of the ball was same as the incidence angle and also there was about 30% energy loss after the rebound. The trajectory of the ball was estimated by the parabolic throw formula which considered the friction of the ball in the air. The ball was returned to a target point on the table. Miyazaki et al. [30] proposed a

racket control method by using three input-output maps which corresponded to racket rebound model and aerodynamics model. The maps were constructed based on experimental data, not on physical models. Their method could return the ball to a desired position on the table with specified flight duration. Yang et al. [54] determined the racket's striking velocity by using the translational velocity of the incoming ball and the expected translational velocity of the returned ball at the striking time in which the energy loss was also considered. As the racket's striking posture, they fixed the racket's pitch angle and decided the racket's yaw angle by using the direction of the coming ball and the rule of the specular reflection.

However, all of the above studies have not considered exact physical models of either racket rebound or aerodynamics. Moreover, the rebound ball's rotational velocity has not been treated as one of control variables.

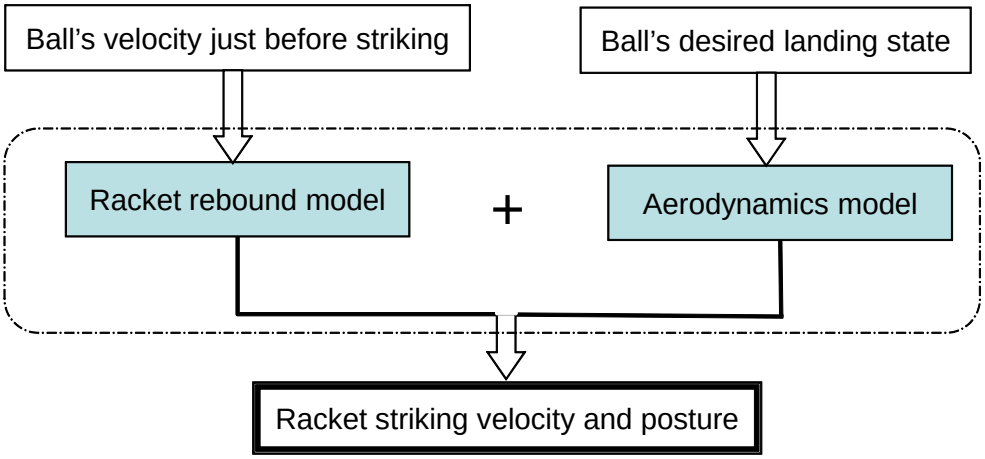


Figure 1.8: The general racket control problem

c) planning a reference trajectory for the manipulator with the racket

The third task of the control system is to plan a reference motion trajectory for the manipulator with the racket. It should be noted that the manipulator motion planning for the table tennis is different from that for the industrial robot arm based on the time optimization. In the table tennis case, in order to successfully return the ball to a desired position with a desired velocity, the racket must achieve a proper state *just at the striking time*. Therefore, the planning is to optimize the manipulator's motion under the constraint of the time. Here, the racket's proper striking state includes three terms: *position*, *velocity* and *posture* which are determined at the above two tasks.

The racket motion is generally planned as a loop which includes four steps: approaching from the waiting position to the striking position; striking with a proper state; reducing the speed to zero; returning to the waiting position (Figure 1.9). And the most popular motion trajectory planning method used for table tennis arm is the fifth degree polynomial method that from the start time to the end time the position trajectory is expressed by using a fifth order polynomial.

However, the fifth order polynomial method does not consider the capacity of the driver. In order to increase the usage of the motor and the success rate of the planning, [20] proposed a fixed-time trapezoidal acceleration trajectory planning method for table tennis and [58] presented a polygonal line speed planning method for their second generation table tennis robot.

In [31, 32], they studied human movements when playing table tennis and presented a robot system which mimicked human striking behavior.

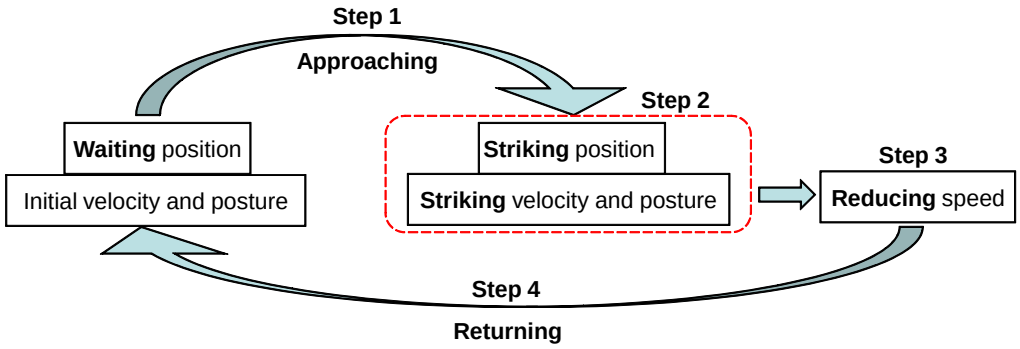


Figure 1.9: Motion planning strategy

Another point is about the equipment of the control system. In some presented table tennis robots [6, 29], both the vision system and the control system share the same CPU which reduces the transmission time of the detected ball's state data and lowers the cost of the robot. However, most of table tennis robots [30, 54] use two or more processers by considering the complexity of the tasks. In this thesis, the developed table tennis system uses one PC for the vision system and one PC for the control system to decrease the computation.

1.3 Demands for Table Tennis Robot

It has been mentioned in Section 1.1 that "robot playing table tennis is a typical *real-time* and *intelligent* process". Therefore, *real-time* and *intelligence* are two important demands for the table tennis robot. Besides them, for achieving a successful striking, the robot also has to satisfy the high requirement on the *accuracy*. In this section, the three basic requirements for the table tennis robot are discussed in detail.

Demand 1: Real Time

"Real time" means the ability of a software or hardware system that can complete a task in a limit time or quickly response to an external event, which is a measurement of the system's processing speed and reaction time. It is the most important requirement for the table tennis system since the speed of the table tennis ball is very high (4∼30 m/s when humans play), which means only a very short response period leaves for the sensors, processors and manipulator. Especially, for the vision system, measuring the ball's position, translational and rotational velocities in a few milliseconds (the time from sensing the image to obtaining the measuring results should be smaller than 80 ms) are really difficult for both the vision sensors and the image processing technology.

Demand 2: Accuracy

The *accuracy* requirement for the table tennis robot has two meanings: 1) the racket is able to strike the ball; 2) after struck by the racket, the ball is able to return to a destination on the opponent court with desired velocities. As for the first meaning, it is a demand on the prediction accuracy of the striking position and the striking time. The prediction error of the striking position should be smaller than " the radius of the racket - the radius of the ball ". On

the other hand, the second meaning is a demand on the determination accuracy of the racket's striking velocity and posture. The accuracy requirement depends on the opponent player that if he is a beginner, the robot just needs to return the ball to an easy position on the table, i.e. the center of the opponent court. However, if he is a player with high skill level, in order to win him, the robot needs to return the ball to a difficult position with high accuracy.

Demand 3: Intelligence

As we all know, table tennis is a "high technique" sport since the ball is so light that it can have high rotational speed and various kinds of spins, e.g. top spin, back spin, side spin. When playing table tennis, the player has to think different strategies to return the balls with different kinds of speeds and spins. Similarly, when the robot plays table tennis, it also has to quickly determine a striking method to overcome these uncertain incoming balls. For example, for each incoming ball, the robot has to decide a new striking position and striking time; and it has to determine a new striking posture and velocity for the racket and plan a new motion trajectory for the manipulator. Therefore, it must be able to adjust its motion and striking method according to the random coming ball, which is quite different from the conventional industrial robot that can only repeat a same task in all the working time. Hence, it is necessary for the table tennis robot to have a high intelligent brain.

1.4 The Purpose of the Thesis

The purpose of this thesis is to realize a table tennis robot which can control a spinning table tennis ball. To achieve this purpose, **the robot must be able to detect and control the rotational velocity of the table tennis ball in a real-time situation.** According to these requirements, in this thesis, we develop a system of table tennis robot as shown in Figure 1.10 and Figure 1.11 which consists of the following three subsystems:

(1) the vision system: two high speed cameras (900 Hz) and a PC;

(2) the control system: a PC;

(3) the mechanical system: a 7-DOF manipulator.

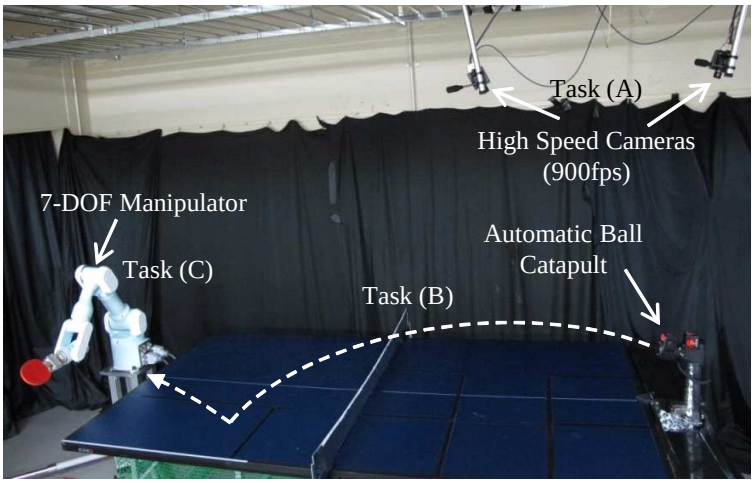


Figure 1.10: A robot playing table tennis

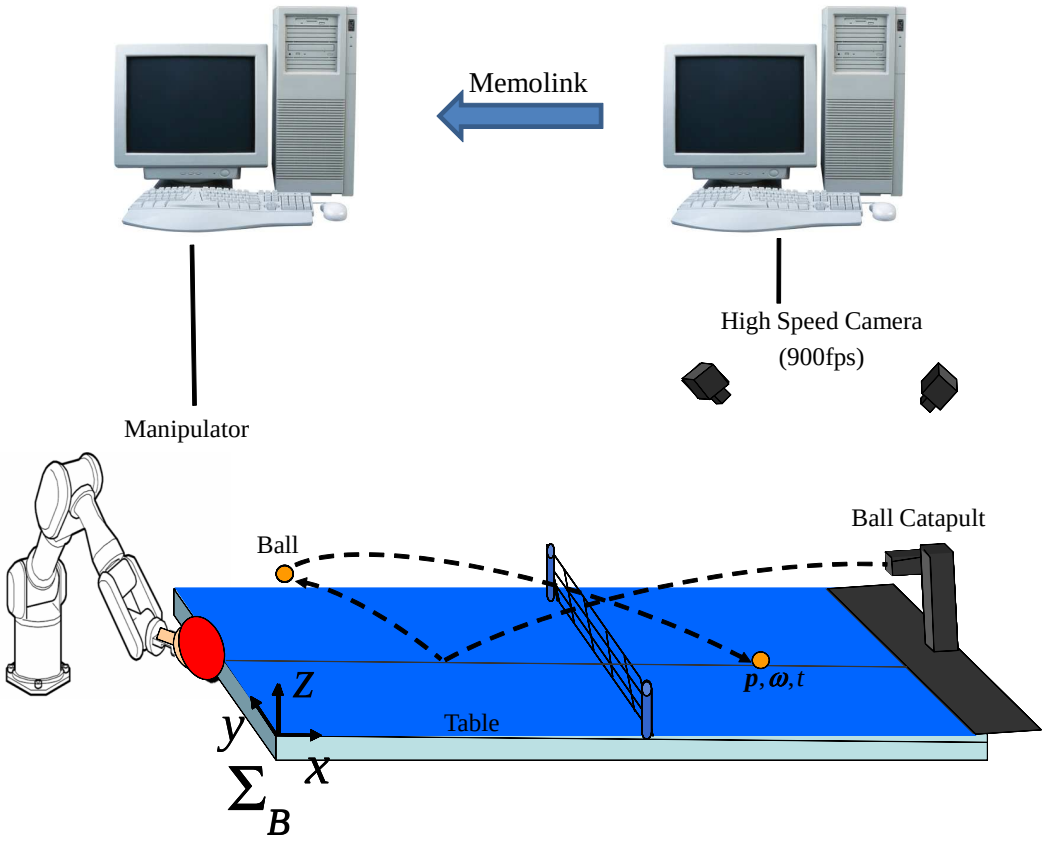


Figure 1.11: The table tennis system

The detail information of these equipments can be found in Appendix A. As the summaries of Section 1.2 and 1.3, in order to reach the goal of controlling the spin of the ball, the above subsystems must be able to accomplish the following three tasks (See Figure 1.10):

(A) Detecting a ball's state (position, translational and rotational velocities) by using vision sensors immediately after the opponent player hit the ball,

(B) Predicting the ball's trajectory by using the ball's state detected in (A), aerodynamic model and collision models with the table as well as the racket, and

(C) Determining the racket's posture and velocity at the striking time such that the returned ball would reach a given destination on the opponent table with a desired landing state and planning a reference trajectory of the robot manipulator with the racket by the ball's trajectory predicted in (B).

Corresponding to the above three tasks, in this thesis, first of all, a real-time method is proposed for measuring the ball's position, translational and rotational velocities by applying two high speed cameras (900 Hz) [22, 23] (Task A). Then, in order to solve the racket control problem, three basic physical models built by our previous research [36, 37, 38] are introduced, which include the aerodynamics model, the table rebound model and the racket rebound model (Task B). Finally, a racket control method is proposed which can control the spin of the returned ball (Task C).

This thesis mainly concentrates on Task (A) and Task (C).

Detect the ball's translational and rotational velocities by using two high speed cameras

For the first task, this thesis proposes an on-line method for estimating both translational and rotational velocities of a table tennis ball by using only a few consecutive frames of image data which are sensed by two high speed cameras. In order to estimate the translational velocity, three-dimensional (3D) position of the ball's center at each instant of camera frame is obtained, where the on-line method of reconstructing the 3D position from the two-dimensional (2D) image data of two cameras is proposed without the pattern matching process. The proposed method of estimating the rotational velocity belongs to the image registration methods, where in order to avoid the pattern matching process too, a rotation model of the ball is used to make an estimated image data from an image data sensed at the previous instant of camera frame and then the estimated image data are compared with the image data sensed at the next instant of camera frame to obtain the most plausible rotational velocity by using the least square and the conjugate gradient method. It is similar to the one shown in [46], however, this thesis clarifies the parts which are vague and/or unexplained [46]; e.g., a process of "image resampling" [57], i.e. the interpolation, and how to search the most plausible rotational velocity from the least square problem. The effectiveness of the proposed method is shown by some experimental results in the case of a ball rotated by a rotation machine as well as in the case of a flying ball shot from a catapult machine.

Determine the racket's state for controlling the returning ball with a desired landing state

For the third task, a racket control method is proposed for returning the ball to a desired position on the opponent court with a desired rotational velocity at a desired landing time. The method determines the racket's state, i.e., the racket's striking posture (yaw and pitch angles) and translational velocity based on two physical models: the racket rebound model and the aerodynamics model. Firstly, in order to better understand the essence of the racket control problem, this thesis divides it as two subproblems (See Figure 1.12):

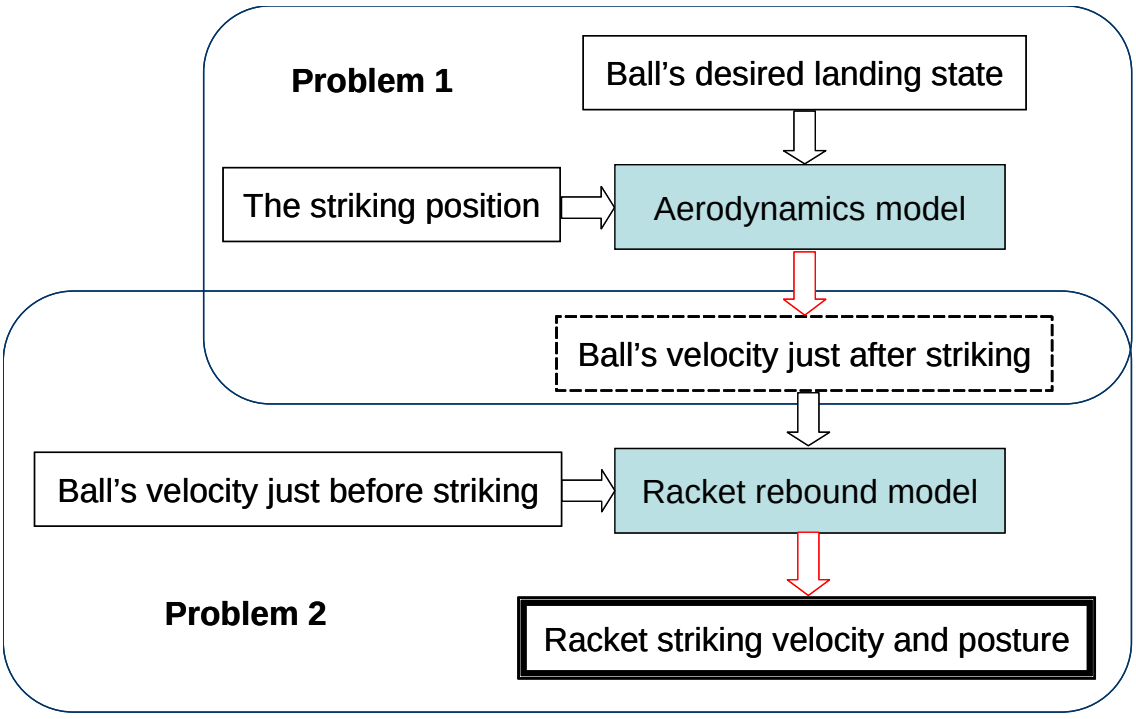


Figure 1.12: Racket control problem

1) supposing the striking position and the ball's desired landing state are given, solve the aerodynamics problem for obtaining the ball's velocity just after rebounding from the racket;

2) supposing the ball's velocities just before and after rebounding are known, solve the racket rebound problem for obtaining the racket's striking velocity and posture.

Notice that in Figure 1.12, the solid line square, the dash line square and the double solid line square express the given parameter, the intermediate solved parameter and the finial solved parameter, respectively. Then, the properties of these two problems (the problem of the aerodynamics model and the problem of the racket rebound model) are clarified from a theoretical viewpoint. In fact, the first problem is a typical two-point boundary value problem about the aerodynamics model that given the boundary conditions, i.e., the striking position and the landing position, solves the differential equation of the aerodynamics model. As for the second problem, basically, for a ball's velocity just before striking, the racket's velocity and posture at the striking time determine the ball's velocity just after striking, which means that the racket's striking velocity and posture determine the changing of ball's velocities just before and after the striking. The relation between them is named as the racket rebound model. Therefore, the second problem, in nature, is solving an inverse problem of the racket rebound model for obtaining the racket's velocity and posture at the striking time.

However, it has been shown that solving the problem of the aerodynamics model needs too much computation time, which is not suitable for a real-time situation. Therefore, a modified algorithm is proposed which can be used for a real-time process by introducing a simple aerodynamics model. The effectiveness of the proposed racket control method is verified by numerical simulations as well as experiments.

1.5 Organization of the Thesis

The followings show the organization of the thesis in detail.

In Chapter 2, an *on-line* algorithm is proposed for measuring the translational and rotational velocities of a flying table tennis ball. As for estimating the translational velocity, the 3D position of the ball's center at each instant of camera frame is obtained by searching points on the contour of ball image and finding perpendicular bisectors between those points. As for the rotational velocity, two methods (Method 1 and Method 2) are demonstrated: Method 1 is an algorithm presented in our previous paper [35]; Method 2 is the proposed method in this thesis which estimates (k+1)th frame of image data by using a candidate of rotational velocity and the sensed kth frame of image data. Then the most plausible rotational velocity is searched by comparing the estimated image data with the sensed image data in every frame. The effectiveness of the proposed on-line algorithm has been verified by comparing with Method 1 with some experimental results: one experiment was carried out by a rotational machine and another experiment was carried out by a catapult machine.

In Chapter 3, firstly, three physical models about ball motion are introduced: aerodynamics model, table rebound model and racket rebound model which consider both the ball's translational velocity and rotational velocity. The properties and the construction processes of the three models are described in detail. Then, some experimental results are shown to verify the effectiveness of these physical models.

In Chapter 4, we propose a racket control method for returning the ball to a desired position with a desired rotational velocity at a desired landing time. Firstly, by regarding the racket rebound model as a set of nonlinear equations with respect to the racket's state, an existence condition for real solutions of the set of nonlinear equations is shown and moreover the solutions are expressed in the closed form. On the other hand, associated with the aerodynamics model, when considering the racket control, it is shown that two-point boundary value problem plays important role. Then, an on-line control method for the racket's state is proposed. Finally, some numerical simulations and experimental results are displayed to verify the effectiveness of the proposed method.

Chapter 5 is the conclusions and future work.

Chapter 2

Online Measuring the Ball's State with High Speed Camera

As discussed in Chapter 1, performing the measuring task of the ball's state has important meanings for predicting the ball's trajectory and controlling the racket. Therefore, it has higher requirements on the *accuracy* and *real-time*. In this Chapter, we propose an *on-line* algorithm for measuring the ball's position, translational and rotational velocities by using about 6 consecutive frames of image data sensed by two high speed cameras (900 Hz) [21, 22, 23].

Actually, there have been a lot of researches on the measurement of the ball's velocity with vision sensors. Zhang et al. [59] shows a method of tracking a flying ball in which the area of the ball image in every frame of image data is detected by means of the frame difference and the search region method. Zhang [58] proposes a method of tracking the ball by using an extended Kalman filter. In these researches, the translational velocity of the ball is measured. However, the rotational velocity of the ball is not measured. It is shown [37] by using the aerodynamic model and some experimental data that the ball's rotation is an important factor that influences the trajectory of flying table tennis ball. Watanabe et al. [51] propose a multi-target tracking algorithm for a high speed vision chip and as one of its applications, they show some experimental results on measuring the rotational velocity of a ball in real-time. The used algorithm is that by the proposed method, multi-feature on the ball's surface is tracked in two-dimensional (2D) image space and also the feature matching between kth frame and $(k+1)$th frame is carried out. Then three-dimensional (3D) position of each feature's centroid in each frame is reconstructed to estimate the rotational velocity by using the consecutive features' 3D positions. The results are excellent; however, unfortunately the experimental results show that the measurable range of the rotational speed is almost less than 1200 rpm, which is not satisfactory because the rotational speed reaches about 3000 rpm in the case of table tennis balls.

As the image registration method [57] without feature matching, the following method is used in [44] and [46]; $(k+1)$th frame of image data are estimated by using a candidate of rotational velocity and the sensed kth frame of image data. Then the most plausible rotational velocity is searched by comparing the estimated image data with the sensed image data in every frame.

Reference [44] shows the measurement of a baseball's spin in *off-line* situation. In [46], the rotational velocity of a table tennis ball is measured. However, the frame rate used in the experiment is 500 Hz which is not fast enough to realize the table tennis playing robot. In addition, unfortunately the estimated results in [46] do not show clearly how well the proposed method can work.

In the previous research [35], an *on-line* method for measuring translational and rotational velocities of a flying table tennis ball has been proposed. Though this method does not belong to the image registration method mentioned above, the method has the same property that the computation cost is low because feature matching is not needed. Unfortunately, it is pointed out in [21] that the proposed method in [35] does not measure the rotational velocity with high accuracy because the method does not take into the consideration of the rotational axis of the ball which causes a rotation of image data in the image coordinate.

In this chapter, an algorithm is proposed to measure the translational and rotational velocities of a table tennis ball in real-time and some experimental results are used to verify the effectiveness of the proposed on-line algorithm; one experiment was carried out by a rotation machine and another experiment was carried out by a catapult machine.

Section 2.1 is the arrangement of the measurement system. In order to measure the 3D position of the ball with the sensed 2D image data, it is necessary to build exact coordinate systems for the equipments. In this section, four 3D and two 2D coordinate systems are introduced and also the conversions between the coordinates are described.

In Section 2.2, a method for estimating the translational velocity of the ball is presented. In order to do that, 3D position of the ball's center at each instant of camera frame is obtained by searching points on the contour of ball image and finding perpendicular bisectors between those points.

In Section 2.3, the previous proposed method [35] (Method 1) is introduced for comparing with the proposed method (Method 2) in the next section.

In Section 2.4, an image registration algorithm (Method 2) for estimating the rotational velocity of the ball is proposed. Firstly, a nearest point method is presented to estimate the intensities of the pixels on the later frame by using the information of the previous frame and then the conjugate gradient method is used to seek the optimum of the rotational velocity.

Section 2.5 is the experimental results with both a ball rotated by a rotational machine and a flying ball shot from a catapult machine which sufficiently demonstrate the effectiveness of the proposed method.

2.1 Preliminary

Figure 2.1 shows a setup of measurement system which consists of a table, a catapult machine and two high speed cameras. The table is an international standard one with $1525(W) \times 760(H) \times 2740(D)$. The high speed cameras (Hamamatsu Photonics), Camera 1 and Camera 2, both are monochrome with frame rate 900 Hz, resolution 232×232, focal length of the lens $f = 3.5 \times 10^{-2}$ m, and pixel size $\varepsilon = 2.0 \times 10^{-5}$ m/pixel. More detailed parameters of them are displayed in Appendix A. In this chapter, four 3D coordinate systems and two 2D coordinate systems are utilized in the measurement system as shown in Figure 2.1.

With respect to the 3D coordinate systems, Σ_B is the base coordinate system which is fixed at a corner of the table. Σ_b is the ball coordinate system fixed at the center of the ball, and hence, Σ_b translates and rotates subject to the ball's motion. The camera coordinate system $\Sigma_{C_n}(n = 1, 2)$ is fixed at Camera 1 and 2, respectively.

With respect to the 2D coordinate systems, the image coordinate system $\Sigma_{I_n}(n = 1, 2)$ is fixed on the image plane of Camera n, where a position on the image plane is denoted by (u, v). The sampling point of the image data in the image plane, i.e., "pixel index", is denoted by (i, j). Notice that (i, j) is a couple of integers and (u, v) is a couple of real numbers. In the measuring method, in order to distinguish explicitly between (i, j) and (u, v), (i, j) is called

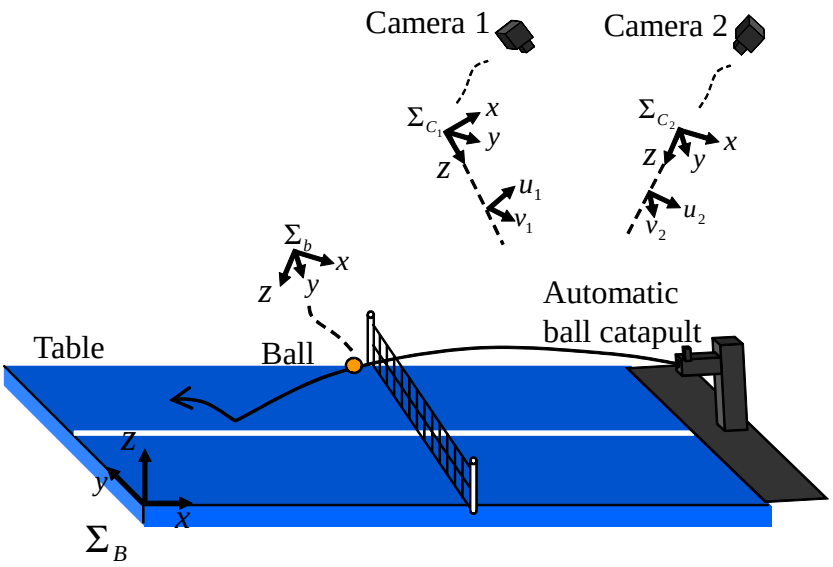


Figure 2.1: Setup of table tennis ball system

a pixel coordinate and (u, v) is called a physical coordinate. Note that every pixel coordinate (i, j) has the corresponding physical coordinate (u, v), i.e., $(u, v) = \varepsilon(i, j)$, but not vice versa.

Figure 2.2 shows the coordinate system Σ_{I_n} in more detail. Because the image plane is finite, some notations for minimum and maximum values of both the pixel and the physical coordinates are prepared as follows.

$$Z_I := \left\{ (i,j)^T \in \mathbb{Z}^2 \;\middle|\; i_{min} \le i \le i_{max}, j_{min} \le j \le j_{max} \right\} \tag{2.1}$$

$$R_I := \left\{ (u,v)^T \in \mathbb{R}^2 \;\middle|\; u_{min} \le u \le u_{max}, v_{min} \le v \le v_{max} \right\} \tag{2.2}$$

where $(u_{min}, v_{min}) := \varepsilon(i_{min} - 0.5, j_{min} - 0.5)$ and $(u_{max}, v_{max}) := \varepsilon(i_{max} + 0.5, j_{max} + 0.5)$. Notice that $i_{max} - i_{min} = j_{max} - j_{min} = 231$.

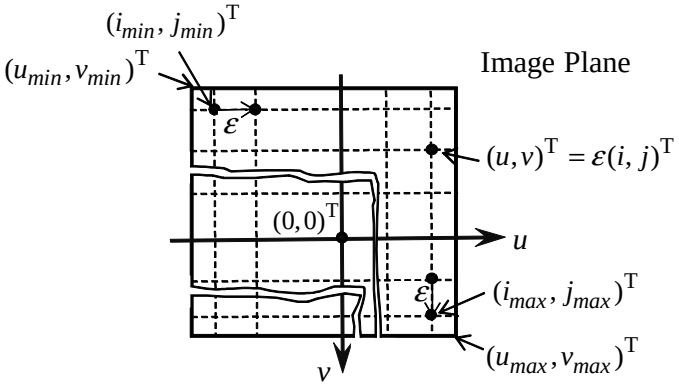


Figure 2.2: Coordinate conversion

Figure 2.3 shows an example of image data we can use to estimate the velocities, which is composed of only 6 frames. This is due to that the high speed camera has a narrow field of vision and also the ball's translational speed is very high. The area of ball image is not so large, compared with the frame size. Note that the ball image has its inscribed quadrangle (the black square) with size about 50×50 in the pixel coordinate.

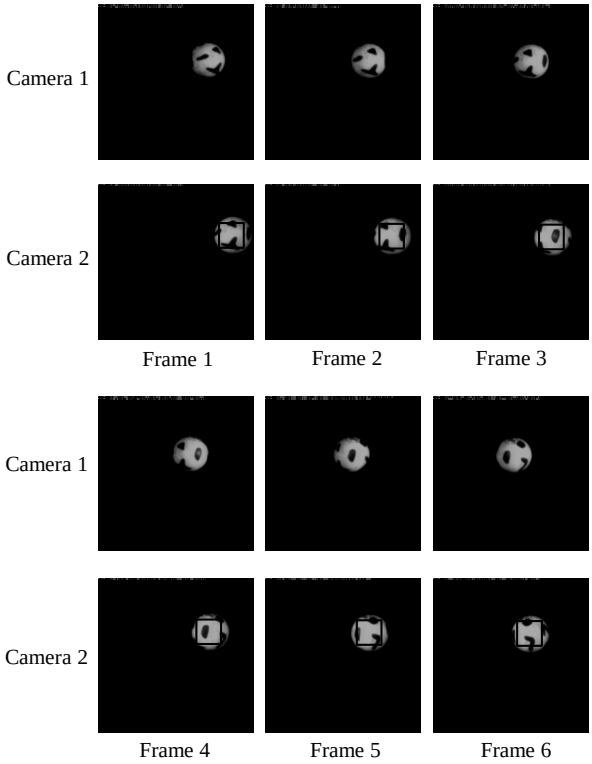


Figure 2.3: Image data obtained by two cameras

Suppose that a 3D position is expressed as ${}^{B}\boldsymbol{P}$ and ${}^{C_n}\boldsymbol{P}$ in the base coordinate system Σ_B and the camera coordinate system $\Sigma_{C_n}(n = 1, 2)$ respectively. Then it is well known that

$$ {}^{C_n}\boldsymbol{P} \;=\; {}^{C_n}\boldsymbol{P}_B + {}^{C_n}\boldsymbol{R}_B\,{}^{B}\boldsymbol{P} \tag{2.3} $$

where ${}^{C_n}\boldsymbol{P}_B \in \mathbb{R}^3$ is a representation vector which describes the origin of Σ_B in Σ_{C_n} and ${}^{C_n}\boldsymbol{R}_B \in \mathbb{R}^{3\times 3}$ is a rotation matrix of Σ_B with respect to Σ_{C_n}. Note that ${}^{C_n}\boldsymbol{P}_B$ and ${}^{C_n}\boldsymbol{R}_B$ are determined and fixed by camera calibration process (See Appendix B).

It is assumed in this thesis that the relation between the camera coordinate system Σ_{C_n} and the image coordinate system Σ_{I_n} is expressed by the pinhole camera model (See Appendix B). Therefore, suppose that a 3D position ${}^{C_n}\boldsymbol{P} = (X_{C_n}, Y_{C_n}, Z_{C_n})^T$ in Σ_{C_n} is projected into a 2D position $\boldsymbol{p}_n = (u_n, v_n)^T$ in Σ_{I_n}, then the perspective projection holds as follows

$$ (u_n, v_n) \;=\; \left(f\frac{X_{Cn}}{Z_{Cn}}, f\frac{Y_{Cn}}{Z_{Cn}} \right) \tag{2.4} $$

where f is the focal length of lens.

2.2 Estimation of the Translational Velocity

Suppose that K frames of image data can be used to estimate the velocity and each frame is sensed at time t^k $(k = 1, \cdots, K)$ where $t^{k+1} - t^k = \Delta t := 1/900$ s. Notice that as mentioned in Preliminary, the number K is less than 10, which corresponds to about 10 ms, a very short time interval. Therefore we assume that the ball's translational velocity during this interval is constant even though the ball flies under the gravity.

The proposed method of estimating a translational velocity (speed and direction) uses both K frames of image data sensed at Camera 1 and Camera 2.

Let ${}^B\boldsymbol{P}_b^k \in \mathbb{R}^3$ denote a position of ball's center at time t^k. Once all ${}^B\boldsymbol{P}_b^k$'s $(k = 1, \cdots, K)$ are estimated, it is easy to see that the most plausible translational velocity ${}^B\boldsymbol{v}_b$ is given by solving the following least square problem.

$$\min_{{}^B\boldsymbol{v}_b, {}^B\bar{\boldsymbol{P}}_b} \sum_{k=1}^{K} \left\{ {}^B\hat{\boldsymbol{P}}_b^k \left({}^B\boldsymbol{v}_b, {}^B\bar{\boldsymbol{P}}_b \right) - {}^B\boldsymbol{P}_b^k \right\}^{\mathrm{T}} \left\{ {}^B\hat{\boldsymbol{P}}_b^k \left({}^B\boldsymbol{v}_b, {}^B\bar{\boldsymbol{P}}_b \right) - {}^B\boldsymbol{P}_b^k \right\}$$

where ${}^B\hat{\boldsymbol{P}}_b^k \left({}^B\boldsymbol{v}_b, {}^B\bar{\boldsymbol{P}}_b \right) := {}^B\boldsymbol{v}_b t^k + {}^B\bar{\boldsymbol{P}}_b$, and ${}^B\bar{\boldsymbol{P}}_b$ is a position of the ball's center at the time $t = 0$.

Therefore it turns out that the original estimation problem is converted into a problem of how to estimate ${}^B\boldsymbol{P}_b^k$ from the kth frame of image data sensed at two cameras. Suppose that $\left({}^n u_b^k, {}^n v_b^k \right)$ $(n = 1, 2)$ is a physical coordinate of the ball's center in the image coordinate system Σ_{I_n}. It is well known that ${}^B\boldsymbol{P}_b^k$ can be reconstructed from both $\left({}^n u_b^k, {}^n v_b^k \right)$'s $(n = 1, 2)$ by using (2.3) and (2.4).

Now an proposed algorithm of estimating $\left({}^n u_b^k, {}^n v_b^k \right)$ from the kth frame of image data in Σ_{I_n} will be shown. Hereafter the proposed image processing is described without the indices k and n since it is same for any time and both cameras.

1) Detect four points on the contour of ball in the image data:

   Figure 2.4 (a) shows a frame of image data, where relatively bright parts correspond to the area of ball image. From left to right and from top to down, scan the image data line by line as shown in Figure 2.4 (a) and search the first pixel coordinate $\boldsymbol{\xi}_1 = (i_1, j_1)$, the image intensity of which is larger than the prespecified threshold. This pixel coordinate $\boldsymbol{\xi}_1$ is regarded as a point on the contour of ball. $\boldsymbol{\xi}_\ell = (i_\ell, j_\ell)$ $(\ell = 2, 3, 4)$ is detected in the same way as $\boldsymbol{\xi}_1 = (i_1, j_1)$. Scanning and searching for $\boldsymbol{\xi}_2, \boldsymbol{\xi}_3$ and $\boldsymbol{\xi}_4$ start from the upper right, the lower right and the lower left respectively as shown in Figure 2.4 (b), (c) and (d).

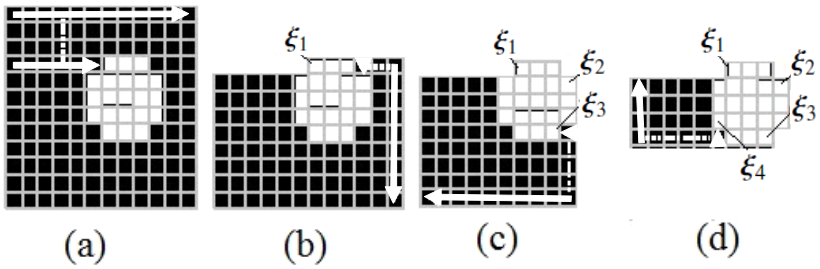


Figure 2.4: Image data and edge points detection

2) Calculate the center of the ball image:

   As shown in Figure 2.6, the center of the ball image is presented as the nearest point to the four perpendicular bisectors between $\boldsymbol{\xi}_\ell$ and $\boldsymbol{\xi}_{\ell+1}$ $(\ell = 1, 2, 3, 4)$ (See Figure 2.5). Note that $\boldsymbol{\xi}_5$ is regarded as $\boldsymbol{\xi}_1$. It is easy to see that the perpendicular bisector between $\boldsymbol{\xi}_\ell$ and $\boldsymbol{\xi}_{\ell+1}$ is given by

$$y - \frac{j_\ell + j_{\ell+1}}{2} = \frac{i_\ell - i_{\ell+1}}{j_{\ell+1} - j_\ell} \left(x - \frac{i_\ell + i_{\ell+1}}{2} \right).$$

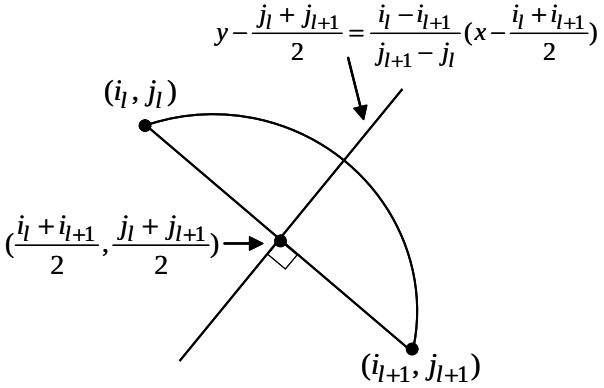


Figure 2.5: The perpendicular bisector between points $\boldsymbol{\xi}_l$ and $\boldsymbol{\xi}_{l+1}$

Therefore the nearest point $\boldsymbol{\xi}_b = (x_b, y_b)$ can be obtained by solving the least square problem as follows.

$$(x_b, y_b) \quad = \quad \arg \min_{x,y} \sum_{\ell=1}^{4} (y - a_\ell x - b_\ell)^2, \tag{2.5}$$

where

$$a_\ell = \frac{i_\ell - i_{\ell+1}}{j_{\ell+1} - j_\ell}, \quad b_\ell = \frac{{i_{\ell+1}}^2 - {i_\ell}^2 + {j_{\ell+1}}^2 - {j_\ell}^2}{2(j_{\ell+1} - j_\ell)} \ \ (l = 1, 2, 3, 4) \tag{2.6}$$

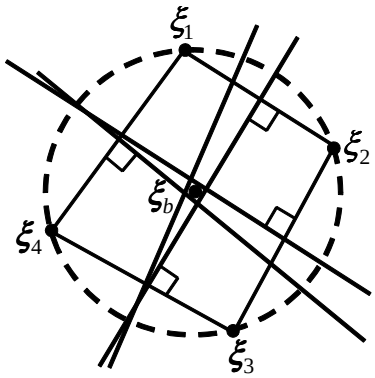


Figure 2.6: Calculation of the ball center on the image

Define $\boldsymbol{A}$, $\boldsymbol{b}$ and $\boldsymbol{x}$ as

$$\boldsymbol{A} = \begin{bmatrix} -a_1 & 1 \\ -a_2 & 1 \\ -a_3 & 1 \\ -a_4 & 1 \end{bmatrix}, \boldsymbol{b} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix}, \boldsymbol{x} = \begin{bmatrix} x \\ y \end{bmatrix}. \tag{2.7}$$

and noting that

$$\sum_{\ell=1}^{4} (y - a_\ell x - b_\ell)^2 = (\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b})^T (\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}). \tag{2.8}$$

Then, it is easy to see that for $\boldsymbol{x}^* = [x_b, y_b]^T$,

$$\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{x}^* - \boldsymbol{A}^T\boldsymbol{b} = 0. \tag{2.9}$$

Therefore,

$$\boldsymbol{x}^* = (\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T\boldsymbol{b}. \tag{2.10}$$

Note that the physical coordinate (u_b, v_b) of the center of the ball image is given as $(u_b, v_b) = \varepsilon(x_b, y_b)$.

2.3 Estimation of the Rotational Velocity (Method 1)

An *on-line* method for measuring the rotational velocity of a flying table tennis ball with the sensed images from two cameras has been proposed by our previous research [35]. In this section, the method is summarized briefly for comparing with the method proposed in the later section. The following notations will be used for estimating the rotational velocity.

${}^B\boldsymbol{P}_b^k$	:	the center of ball at time t^k in Σ_B
${}^B\boldsymbol{P}_m^k$	:	the centroid of the mth feature area at time t^k in $\Sigma_B, m \in \{1, 2, ..., M\}$
${}^b\boldsymbol{P}_m^k$	:	the centroid of the mth feature area at time t^k in Σ_b
${}^n\boldsymbol{\xi}_c^k$	:	the center of ball on the image at time t^k in $\Sigma_{I_n} (n = 1, 2)$
${}^n\boldsymbol{\xi}_m^k$	:	the image coordinate corresponding to ${}^B\boldsymbol{P}_m^k$

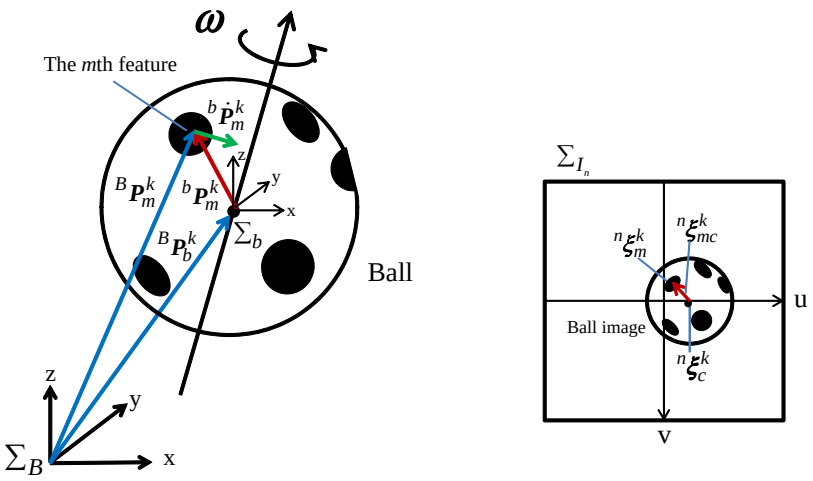


Figure 2.7: The position of the feature's centroid on the ball

In Figure 2.7, the left one shows the 3D position of the centroid of the feature area on the ball and the right one is the corresponding image coordinate of that feature area's centroid. Notice that ${}^b\boldsymbol{P}_m^k = {}^B\boldsymbol{P}_m^k - {}^B\boldsymbol{P}_b^k$ and its corresponding vector on the image is ${}^n\boldsymbol{\xi}_{mc}^k = {}^n\boldsymbol{\xi}_m^k - {}^n\boldsymbol{\xi}_c^k$ where ${}^n\boldsymbol{\xi}_m^k = ({}^nu_m^k, {}^nv_m^k)^T$.

A basic relation between ${}^b\boldsymbol{P}_m^k$ and $\boldsymbol{\omega} \in \mathbb{R}^3$ is well-known as

$${}^b\dot{\boldsymbol{P}}_m^k = \boldsymbol{\omega} \times {}^b\boldsymbol{P}_m^k = -\boldsymbol{S}({}^b\boldsymbol{P}_m^k)\boldsymbol{\omega} \tag{2.11}$$

where $\boldsymbol{S}({}^{b}\boldsymbol{P}_m^k) \in \mathbb{R}^{3\times 3}$ is a skew-symmetric matrix corresponding to the cross product of ${}^{b}\boldsymbol{P}_m^k$.

On the other hand, by substituting (2.3) into (2.4), the kinematics between the centroid of the feature area ${}^{B}\boldsymbol{P}_m^k$ and the corresponding 2D image coordinate $\boldsymbol{\xi}_m^k = \left[{}^{1}u_m^k, {}^{1}v_m^k, {}^{2}u_m^k, {}^{2}v_m^k\right]^T \in \mathbb{R}^4$ is given as

$$
\begin{aligned}
{}^{n}u_m^k &= \frac{f}{\varepsilon}\frac{r_{11}{}^{B}P_{xm}^k + r_{12}{}^{B}P_{ym}^k + r_{13}{}^{B}P_{zm}^k + {}^{C_n}P_x}{r_{31}{}^{B}P_{xm}^k + r_{32}{}^{B}P_{ym}^k + r_{33}{}^{B}P_{zm}^k + {}^{C_n}P_z}, \\
{}^{n}v_m^k &= \frac{f}{\varepsilon}\frac{r_{21}{}^{B}P_{xm}^k + r_{22}{}^{B}P_{ym}^k + r_{23}{}^{B}P_{zm}^k + {}^{C_n}P_y}{r_{31}{}^{B}P_{xm}^k + r_{32}{}^{B}P_{ym}^k + r_{33}{}^{B}P_{zm}^k + {}^{C_n}P_z},
\end{aligned}
\tag{2.12}
$$

where r_{ij} is the element of ${}^{C_n}\boldsymbol{R}_B$. Differentiating (2.12) under $n = 1, 2$ leads to

$$
\dot{\boldsymbol{\xi}}_m^k = \boldsymbol{J}({}^{B}\boldsymbol{P}_m^k){}^{B}\dot{\boldsymbol{P}}_m^k \tag{2.13}
$$

in which

$$
\boldsymbol{J}({}^{B}\boldsymbol{P}_m^k) = \frac{\partial \boldsymbol{\xi}_m^k}{\partial {}^{B}\boldsymbol{P}_m^k} = \begin{bmatrix} \frac{\partial {}^{1}u_m^k}{\partial {}^{B}P_{xm}^k} & \frac{\partial {}^{1}u_m^k}{\partial {}^{B}P_{ym}^k} & \frac{\partial {}^{1}u_m^k}{\partial {}^{B}P_{zm}^k} \\ \frac{\partial {}^{1}v_m^k}{\partial {}^{B}P_{xm}^k} & \frac{\partial {}^{1}v_m^k}{\partial {}^{B}P_{ym}^k} & \frac{\partial {}^{1}v_m^k}{\partial {}^{B}P_{zm}^k} \\ \frac{\partial {}^{2}u_m^k}{\partial {}^{B}P_{xm}^k} & \frac{\partial {}^{2}u_m^k}{\partial {}^{B}P_{ym}^k} & \frac{\partial {}^{2}u_m^k}{\partial {}^{B}P_{zm}^k} \\ \frac{\partial {}^{2}v_m^k}{\partial {}^{B}P_{xm}^k} & \frac{\partial {}^{2}v_m^k}{\partial {}^{B}P_{ym}^k} & \frac{\partial {}^{2}v_m^k}{\partial {}^{B}P_{zm}^k} \end{bmatrix} \in \mathbb{R}^{4\times 3} \tag{2.14}
$$

is the image jacobian. (2.13) is equivalent to

$$
\dot{\boldsymbol{\xi}}_{mc}^k + \dot{\boldsymbol{\xi}}_c^k = \boldsymbol{J}({}^{B}\boldsymbol{P}_m^k)({}^{b}\dot{\boldsymbol{P}}_m^k + {}^{B}\dot{\boldsymbol{P}}_b^k). \tag{2.15}
$$

Note that $\boldsymbol{\xi}_c^k$ and ${}^{B}\boldsymbol{P}_b^k$ are related by

$$
\dot{\boldsymbol{\xi}}_c^k = \boldsymbol{J}({}^{B}\boldsymbol{P}_m^k){}^{B}\dot{\boldsymbol{P}}_b^k. \tag{2.16}
$$

Substituting (2.16) into (2.15) and combining the result equation with (2.11) lead to

$$
\dot{\boldsymbol{\xi}}_{mc}^k = -\boldsymbol{J}({}^{B}\boldsymbol{P}_m^k)\boldsymbol{S}({}^{b}\boldsymbol{P}_m^k)\boldsymbol{\omega}, \tag{2.17}
$$

which can be discretized as

$$
\Delta\boldsymbol{\xi}_{mc}^k = -\boldsymbol{J}({}^{B}\boldsymbol{P}_m^k)\boldsymbol{S}({}^{b}\boldsymbol{P}_m^k)\boldsymbol{\omega}\Delta t, \tag{2.18}
$$

where $\Delta\boldsymbol{\xi}_{mc}^k = \boldsymbol{\xi}_{mc}^k - \boldsymbol{\xi}_{mc}^{k-1}$.

Finally, the rotational velocity is derived as,

$$
\boldsymbol{\omega} = -\{\boldsymbol{J}({}^{B}\boldsymbol{P}_m^k)\boldsymbol{S}({}^{b}\boldsymbol{P}_m^k)\}^{+}\frac{\Delta\boldsymbol{\xi}_{mc}^k}{\Delta t}. \tag{2.19}
$$

Notice that in (2.19), ${}^{b}\boldsymbol{P}_m^k$ can be solved by the relation ${}^{b}\boldsymbol{P}_m^k = {}^{B}\boldsymbol{P}_m^k - {}^{B}\boldsymbol{P}_b^k$. However, it is impossible to obtain ${}^{B}\boldsymbol{P}_m^k$ and $\Delta\boldsymbol{\xi}_{mc}^k$ since $\boldsymbol{\xi}_{mc}^k$ can not find its correspondence in the previous frame. Here, an on-line method of estimating the rotational velocity is proposed where instead of considering (2.18) for each feature, (2.18) is regarded as a relation of the average values $\Delta\boldsymbol{\xi}_{\bar{m}c}^k$, ${}^{B}\boldsymbol{P}_{\bar{m}}^k$ and ${}^{b}\boldsymbol{P}_{\bar{m}}^k$ of $\{\Delta\boldsymbol{\xi}_{mc}^k, {}^{B}\boldsymbol{P}_m^k, {}^{b}\boldsymbol{P}_m^k | m = 1, 2, ..., M\}$, i.e.,

$$
\Delta\boldsymbol{\xi}_{\bar{m}c}^k = -\boldsymbol{J}({}^{B}\boldsymbol{P}_{\bar{m}}^k)\boldsymbol{S}({}^{b}\boldsymbol{P}_{\bar{m}}^k)\boldsymbol{\omega}\Delta t. \tag{2.20}
$$

(1) How to get $\Delta\boldsymbol{\xi}^k_{\bar{m}c}$.

In Figure 2.8, ${}^nG^k(s,t) \in \mathbb{N}^{232\times232}$ is defined as the *k*th frame of image data in Camera n (n = 1, 2) and ${}^ng^k(s,t) \in \mathbb{N}^{N_{gn}\times N_{gn}}$ is the segmented 2D image data of the inscribed quadrangle of the ball at Camera n. Note that the values of ${}^nG^k$ and ${}^ng^k$ are binarized to (0, 1) with a suitable threshold. Method 1 estimates the correlation between the *k*th and (*k*+1)th images by minimizing the following performance function:

$$V_n({}^nh, {}^nl) := \frac{1}{(N_{gn}-{}^nh)(N_{gn}-{}^nl)} \times \sum_{s=1}^{N_{gn}-{}^nh} \sum_{t=1}^{N_{gn}-{}^nl} \left({}^ng^{k+1}(s,t) - {}^ng^k(s+{}^nl, t+{}^nh)\right)^2. \tag{2.21}$$

This operation can be illustrated by using the right of Figure 2.8, where $({}^nh_{min}, {}^nl_{min})$ is the

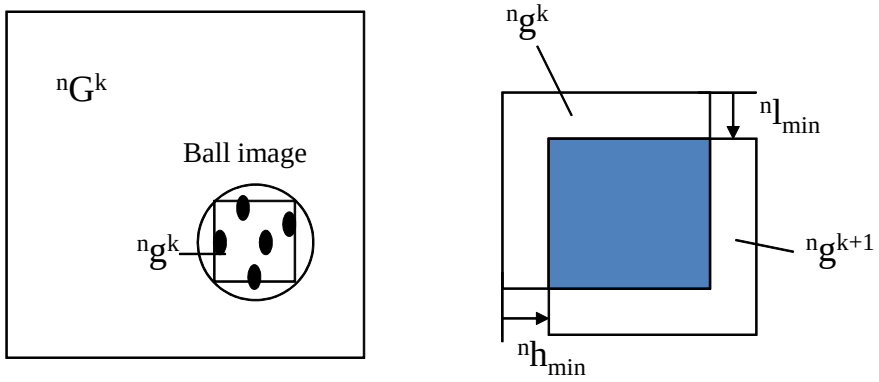


Figure 2.8: The segmented 2D image data and the correlation between the two images

solution of the correlation. In fact, $({}^nh_{min}, {}^nl_{min})$ is the best horizontal and vertical shifts that the sensed $(k+1)$th frame of image data matches the sensed kth frame of image data of Camera n. Then, the representative image displacement $\Delta\boldsymbol{\xi}^k_{\bar{m}c}$ is expressed as:

$$\Delta\boldsymbol{\xi}^k_{\bar{m}c} = [{}^1h^k_{min}, {}^1l^k_{min}, {}^2h^k_{min}, {}^2l^k_{min}]^T. \tag{2.22}$$

(2) How to get ${}^B\boldsymbol{P}^k_{\bar{m}}$ and ${}^b\boldsymbol{P}^k_{\bar{m}}$.

Firstly, as shown in Figure 2.9, the centroids of the feature areas on the 2D images are calculated by

$${}^n\boldsymbol{\xi}^k_f = \frac{\Sigma_{s,t\in D}{}^nG^k(s,t)[s,t]^T}{\Sigma_{s,t\in D}{}^nG^k(s,t)} - {}^n\boldsymbol{\xi}^k_c, \tag{2.23}$$

where

$$D := \left\{(s,t)\middle|\left[{}^n\xi^k_c - \frac{N_{gn}}{2}\right] \le s,t \le \left[{}^n\xi^k_c + \frac{N_{gn}}{2}\right]\right\}$$

and $[.]$ stands for the integer as an argument rounded off.

Secondly, the representative 3D position ${}^B\boldsymbol{P}^k_{\bar{m}}$ is calculated by using the centroids ${}^1\boldsymbol{\xi}^k_f$ and ${}^2\boldsymbol{\xi}^k_f$.

Thirdly, ${}^B\boldsymbol{P}^k_{\bar{m}}$ is modified to a position on the surface of the ball as shown in [35] and ${}^b\boldsymbol{P}^k_{\bar{m}} = {}^B\boldsymbol{P}^k_{\bar{m}} - {}^B\boldsymbol{P}^k_b$.

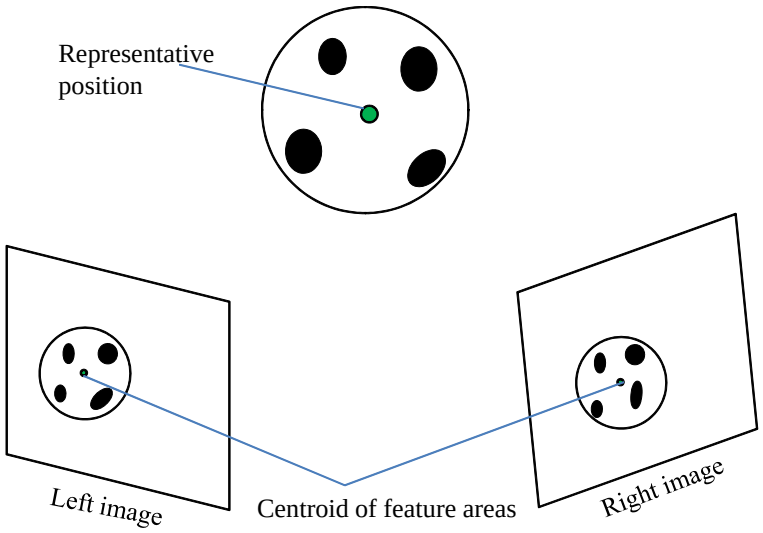


Figure 2.9: The representative position of the feature areas

The advantage of this method is that the computation cost is low since feature matching is not needed. However, the method does not measure the rotational velocity with high accuracy [21] *because the method does not take into the consideration of the rotational axis of the ball which causes a rotation of image data in the image coordinate.*

2.4 Estimation of the Rotational Velocity (Method 2)

In this section, we also propose an algorithm of estimating the rotational velocity by using the K frames of image data under the condition that the image processing mentioned in Section 2.2 has been already done, i.e., all the values below have been known and so they can be used to estimate the rotational velocity.

$$
\begin{aligned}
{}^{B}\boldsymbol{P}_b^k &: \text{the center of ball at time } t^k \text{ in } \Sigma_B \\
{}^{C_n}\boldsymbol{P}_b^k &: \text{the center of ball at time } t^k \text{ in } \Sigma_{C_n} \\
{}^{B}\boldsymbol{v}_b &: \text{the estimated translational velocity in } \Sigma_B \\
{}^{C_n}\boldsymbol{v}_b &: \text{the estimated translational velocity in } \Sigma_{C_n}
\end{aligned}
$$

The proposed method belongs to the image registration method. Our method uses the K frames of image data sensed by one camera, say Camera 2, under the above condition. In addition, hereafter almost all things are described in the camera coordinate system Σ_{C_2}. Therefore, in this section, we use simple notations $\boldsymbol{P}^k$, $\boldsymbol{P}_b^k$ and $\boldsymbol{v}_b$ instead of ${}^{C_2}\boldsymbol{P}^k$, ${}^{C_2}\boldsymbol{P}_b^k$ and ${}^{C_2}\boldsymbol{v}_b$ respectively.

Suppose that as shown in Figure 2.10, an specified point on the ball's surface, which is a fixed point in the ball coordinate system Σ_b, is represented by $\boldsymbol{P}^k$ at time t^k in the camera coordinate system Σ_{C_2}. Then it is well known that

$$
\boldsymbol{P}^{k+1} - \boldsymbol{P}_b^{k+1} = \boldsymbol{R}_{\boldsymbol{\omega}} \left(\boldsymbol{P}^k - \boldsymbol{P}_b^k \right) \tag{2.24}
$$

$$
\boldsymbol{R}_{\boldsymbol{\omega}} = \exp \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} \Delta t \tag{2.25}
$$

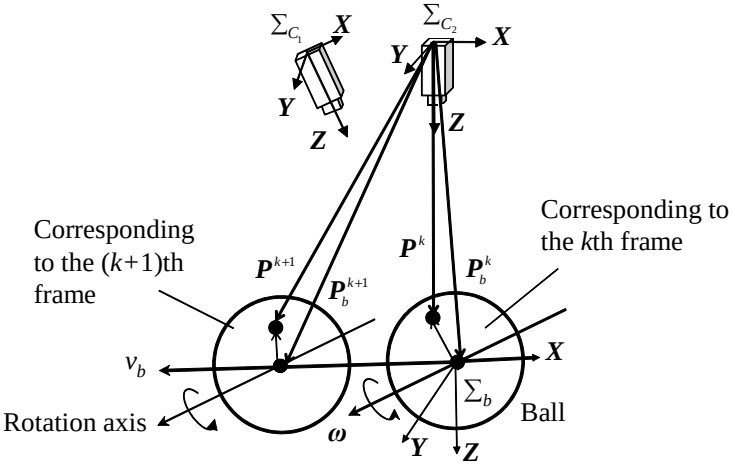


Figure 2.10: Ball's rotation with translation in the camera coordinate system

where $\boldsymbol{\omega} = (\omega_x, \omega_y, \omega_z)^T$ is the rotational velocity.

The proposed method consists of two sorts of image processings; the first one is to construct the estimated $(k+1)$th frame of image data by using the sensed kth frame of image data and a candidate of rotational velocity $\boldsymbol{\omega}$. The second one is to search the most plausible rotational velocity $\boldsymbol{\omega}^*$ by solving a least square problem with an objective function given as a difference between the estimated image data and the sensed image data all over the frames. Hereafter those two processings will be explained in detail.

2.4.1 Construction of the estimated $(k+1)$th frame of image data

Let $\boldsymbol{P} \in \mathbb{R}^3$ denote a point on the ball's surface in the camera coordinate system Σ_{C_2} and this point $\boldsymbol{P}$ is assumed to be projected into the physical coordinate $\boldsymbol{p} \in \mathbb{R}^2$ in the image coordinate system Σ_{I_2}.

Then the mapping $\boldsymbol{P} \rightarrow \boldsymbol{p}$ is just the perspective projection given as (2.4). In fact that $\boldsymbol{P} = (X_{C_n}, Y_{C_n}, Z_{C_n})^T$ and $\boldsymbol{p} = (u_n, v_n)^T$ in (2.4). Moreover, it is straightforward to see that the inverse mapping $\boldsymbol{p} \rightarrow \boldsymbol{P}$ exists because $\boldsymbol{P}$ can be solved in (2.4) with the condition that $\boldsymbol{P}$ locates on the ball's surface, i.e., $\|\boldsymbol{P} - \boldsymbol{P}_b\| = r$ where $\boldsymbol{P}_b$ is the ball's center.

Figure 2.11 displays a flow of constructing the estimated $(k+1)$th frame of image data.

Physical coordinate transformation from the kth frame to the $(k+1)$th frame

By sensing the kth frame of image data, we obtain the image intensities of image data. The image intensity at the pixel coordinate $(i,j)^T \in Z_I$ is denoted by $I^k_{(i,j)}$.

When the estimated $(k+1)$th frame of image data are constructed by using the kth frame of image data and a candidate of rotational velocity, in order to express interim image data, the image intensity at the physical coordinate $\boldsymbol{p} := (u,v)^T \in R_I$, which is described by $I^{k+1}(\boldsymbol{p})$, is needed.

Given the kth frame of image data, we pay attention to the image data in the inscribed quadrangle of the ball image which is shown as a black quadrangle in Figure 2.11. The set of all pixel coordinates inside the black quadrangle is denoted by

$$Z^k_{Iq} := \left\{ (i,j)^T \in Z_I \;\middle|\; i^k_{minq} \leq i \leq i^k_{maxq},\ j^k_{minq} \leq j \leq j^k_{maxq} \right\}.$$

Associated with a pixel coordinate $(i,j) \in Z_{Iq}^k$, the 3D point on the ball's surface which is projected into (i,j) is denoted by $\boldsymbol{P}_{(i,j)}^k$ (See Step 1 in Figure 2.11).

In (2.24) and (2.25) with $\boldsymbol{P}^k$ replaced by $\boldsymbol{P}_{(i,j)}^k$ and a candidate of rotational velocity $\boldsymbol{\omega}$, we can obtain $\boldsymbol{P}^{k+1}$ as a function of (i,j) and $\boldsymbol{\omega}$, which is denoted by $\boldsymbol{P}_{(i,j)}^{k+1}(\boldsymbol{\omega})$. Thus our estimation is that the point $\boldsymbol{P}_{(i,j)}^k$ moves to the point $\boldsymbol{P}_{(i,j)}^{k+1}(\boldsymbol{\omega})$ at time t^{k+1} (See Step 2 in Figure 2.11).

According to (2.4), the 3D point $\boldsymbol{P}_{(i,j)}^{k+1}(\boldsymbol{\omega})$ is projected into a 2D point with the physical coordinate $(u,v)^T$. This physical coordinate is denoted by $\boldsymbol{p}_{(i,j)}(\boldsymbol{\omega})$ because it is a function of (i,j) and $\boldsymbol{\omega}$ (See Step 3 in Figure 2.11).

From the above observation, we conclude that under the rotational velocity $\boldsymbol{\omega}$, the point, which is projected into the pixel coordinate (i,j) in the kth frame, is projected into the physical coordinate $\boldsymbol{p}_{(i,j)}(\boldsymbol{\omega})$ in the $(k+1)$th frame. Therefore, it is reasonable that the image intensity of the $(k+1)$th frame is estimated as follows.

$$I^{k+1}\left(\boldsymbol{p}_{(i,j)}(\boldsymbol{\omega})\right) \;=\; I_{(i,j)}^k \quad \text{for } \forall (i,j) \in Z_{Iq}^k \tag{2.26}$$

Recall that the pixel coordinate (i,j) is discrete and the physical coordinate (u,v) is continuous. Therefore, notice that in general, $\boldsymbol{p}_{(i,j)}(\boldsymbol{\omega})$ has no corresponding pixel coordinate.

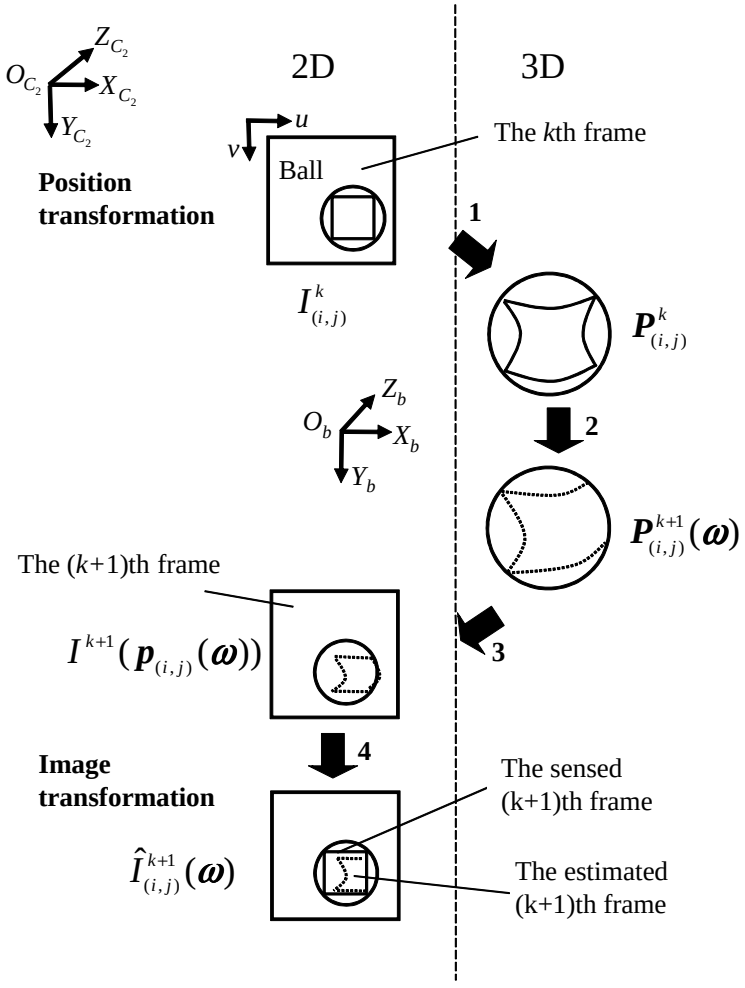


Figure 2.11: Flow of constructing the estimated $(k+1)$th frame of image data

Image resampling for the estimated $(k+1)$ th frame of image data

As mentioned above, the constructed $(k+1)$th frame of image data given by (2.26) can not be compared with the sensed $(k+1)$th frame of image data because the sensed frame of image data have the intensities at the pixel coordinates. On the other hand, as shown in Figure 2.12, almost all $\boldsymbol{p}_{(i,j)}(\boldsymbol{\omega})$'s (denoted by blue points) do not correspond to any the pixel coordinates (i,j)'s (denoted by red points).

Therefore the estimated $(k+1)$th frame of image data which have intensities at the pixel coordinate must be constructed from $I^{k+1}\left(\boldsymbol{p}_{(i,j)}(\boldsymbol{\omega})\right)$'s (see Step 4 in Figure 2.11), which is a sort of interpolation. The constructed intensity at the pixel coordinate (i,j) is denoted by $\hat{I}^{k+1}_{(i,j)}(\boldsymbol{\omega})$.

There are many ways of interpolations [21] and here the nearest point method is employed.

Associated with the pixel coordinate (i,j) in the image coordinate system Σ_{I_2}, define its ε neighborhood $\boldsymbol{N}_{(i,j)}$as

$$\boldsymbol{N}_{(i,j)} \;:=\; \left\{ (u,v)^T \in \boldsymbol{R}_I \;\middle|\; \left\| (u,v)^T - \varepsilon(i,j)^T \right\| < \varepsilon \right\}. \tag{2.27}$$

And also define the set of all $\boldsymbol{p}_{(i,j)}$'s as

$$\boldsymbol{S}^{k+1}(\boldsymbol{\omega}) \;:=\; \left\{ \boldsymbol{p}_{(i,j)}(\boldsymbol{\omega}) \;\middle|\; (i,j) \in \boldsymbol{Z}^k_{Iq} \right\} \tag{2.28}$$

and then define

$$\boldsymbol{NS}^{k+1}_{(i,j)}(\boldsymbol{\omega}) \;:=\; \boldsymbol{N}_{(i,j)} \cap \boldsymbol{S}^{k+1}(\boldsymbol{\omega}). \tag{2.29}$$

The nearest point method assigns the estimated image intensity $\hat{I}^{k+1}_{(i,j)}(\boldsymbol{\omega})$ as

$$\hat{I}^{k+1}_{(i,j)}(\boldsymbol{\omega}) \;=\; I^{k+1}\left(\boldsymbol{p}_{(\ell^*,m^*)}(\boldsymbol{\omega})\right) \tag{2.30}$$

where $\boldsymbol{p}_{(\ell^*,m^*)}(\boldsymbol{\omega}) \in \boldsymbol{NS}^{k+1}_{(i,j)}(\boldsymbol{\omega})$ is the nearest point to the pixel coordinate (i,j), i.e.,

$$\boldsymbol{p}_{(\ell^*,m^*)}(\boldsymbol{\omega}) \;=\; \arg \min_{\boldsymbol{p} \in \boldsymbol{NS}^{k+1}_{(i,j)}(\boldsymbol{\omega})} \left\| \boldsymbol{p} - \varepsilon(i,j)^T \right\|. \tag{2.31}$$

Note that if $\boldsymbol{NS}^{k+1}_{(i,j)}(\boldsymbol{\omega}) = \phi$, then $\hat{I}^{k+1}_{(i,j)}(\boldsymbol{\omega})$ is not defined.

2.4.2 Estimation of the most plausible rotational velocity

In order to search the most plausible rotational velocity $\boldsymbol{\omega}^*$, a least square problem is formulated as $\min_{\boldsymbol{\omega}} E(\boldsymbol{\omega})$, where the objective function is given by

$$E(\boldsymbol{\omega}) \;:=\; \frac{1}{K-1} \sum_{k=1}^{K-1} \frac{1}{|\boldsymbol{IJ}^{k+1}|} \sum_{(i,j) \in \boldsymbol{IJ}^{k+1}} \left\{ \hat{I}^{k+1}_{(i,j)}(\boldsymbol{\omega}) - I^{k+1}_{(i,j)} \right\}^2 \tag{2.32}$$

with $\boldsymbol{IJ}^{k+1} := \left\{ (i,j) \;\middle|\; \boldsymbol{NS}^{k+1}_{(i,j)}(\boldsymbol{\omega}) \neq \phi \right\}$.

The conjugate gradient method is applied to solve the above least square problem because the conjugate gradient method does not need to calculate the Jacobian at every iteration, which saves storages and computation time.

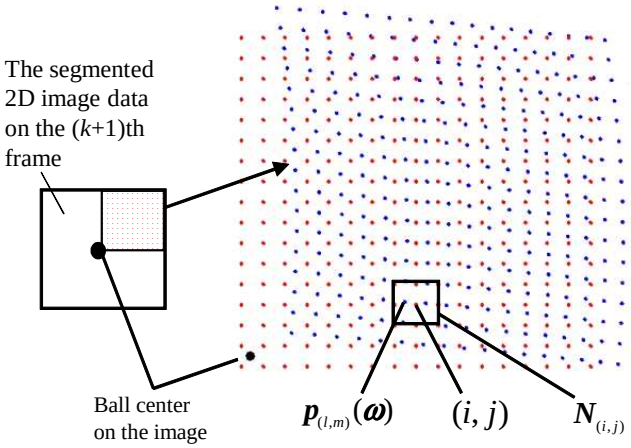


Figure 2.12: Projection of $\boldsymbol{P}_{(i,j)}^{k+1}(\boldsymbol{\omega})$ to $\boldsymbol{p}_{(i,j)}(\boldsymbol{\omega})$ on the later frame

The line search algorithm updates the estimated rotational velocity $\boldsymbol{\omega}_h^q$ as follows.

$$\begin{cases} \boldsymbol{\omega}_h^q = \boldsymbol{\omega}_h + \alpha_h^q \boldsymbol{d}_h \\ \boldsymbol{\omega}_{h+1} = \boldsymbol{\omega}_h + \alpha_h^{q_h} \boldsymbol{d}_h \\ \alpha_{h+1}^0 = \alpha_h^{q_h} \end{cases} \tag{2.33}$$

where $h \in \mathbb{N}$ and $\boldsymbol{d}_h \in \mathbb{R}^3$ is the descent direction of $E(\boldsymbol{\omega})$, i.e.,

$$\boldsymbol{d}_h = \begin{cases} -\boldsymbol{g}_h & \text{for } h = 0 \\ -\boldsymbol{g}_h + \beta_h \boldsymbol{d}_{h-1} & \text{for } h \geq 1 \end{cases} \tag{2.34}$$

and $\boldsymbol{g}_h = \frac{\partial E(\boldsymbol{\omega})}{\partial \boldsymbol{\omega}}|_{\boldsymbol{\omega}_h}$.

β_h is defined by the Polak-Ribiere-Polyak method [56]. Note that β_h is limited in [0,1].

$\alpha_h^q > 0$ represents the step size, which starts from an initial step size α_0^0. While searching along the direction $\boldsymbol{d}_h$, α_h^q is updated as α_h^{q+1} with $q \in \mathbb{N}$ and this process terminates at $q = q_h$ where $\alpha_h^{q_h}$ satisfies the weak Wolfe-Powell rule [56]. Notice that the initial values $\boldsymbol{\omega}_0$, α_0^0 are given.

The iteration number IN is defined as the total sum of the step size searching numbers in all the descend directions.

$$IN = \sum_h q_h \tag{2.35}$$

In the step size searching process, either if the step size $\alpha_h^q < \alpha_{min}$ or if $E(\boldsymbol{\omega}) < E_{min}$ and $\|\boldsymbol{g}_h\| < g_{max}$, the iteration terminates. In addition, we set $\alpha_0^0 = 20$, $q_h < q_{max}$ and $h < h_{max}$ to avoid too much process time.

2.5 Experimental Results

This section demonstrates two kinds of experimental results; one is carried out with a rotation machine and another is carried out with a catapult machine.

As you can see from Figures 2.3, 2.13, and 2.19, each black blob on the ball surface is painted in different shape each other, e.g., thick line, rectangular, ellipse, etc. It is paid attention in the experiments that the balls have almost the same patterns on their surfaces [*].

[*]The estimation precision would change when the ball with different shapes of blobs is used. The optimal pattern of shapes for estimating ball's velocity is not known and it is a future problem.

The light condition is also important in the experiments when the high speed cameras are used. In the case of experiments with the rotation machine, one halogen lamp (300 W) is located about 0.5 m far from the sensed ball. In the case of experiments with the catapult machine, two halogen lamps (300 W) are used, one of them is located about 0.5 m far from the sensed ball and the other is located about 1 m far from the sensed ball. Those arrangements are so decided that there is a shade as few as possible around the contour of the ball images *. The background is set in black color.

In all the cases, six frames of image data are used to estimate the translational and rotational velocities, i.e. $K = 6$, because it is realized that if $K \leq 4$, the proposed method sometimes provides unstable estimated velocities.

In the conjugate gradient method used for estimating the rotational velocity, the initial value of rotational velocity is set as $\boldsymbol{\omega}_0 = 1000\left(\frac{1}{\sqrt{3}}, -\frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}\right)$ rpm. In addition, the parameters with which the iteration is terminated are set as follows; $\alpha_{min} = 0.5$, $E_{min} = 1000$, $g_{max} = 3$, $q_{max} = 4$ and h_{max} = 6.

2.5.1 Experiments with a rotation machine

Figure 2.13 shows a setup of experiment using a rotation machine. A motor which drives a rotational axis of the machine is controlled by PC through a Motion Controller and the rotation machine can give a rotation to a table tennis ball with any rotational velocity as you like.

In the experiment with the rotation machine, the ball has no translational velocity. Therefore this experiment aims exclusively to verify how well the proposed method of estimating the rotational velocity works. However, the *on-line* algorithm which includes the image processing of both translational and rotational velocities is used even in this experiment.

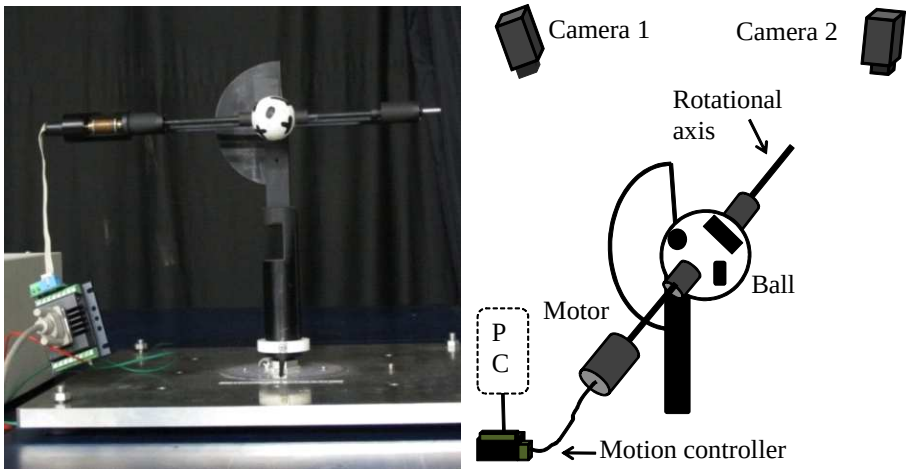


Figure 2.13: The rotation machine

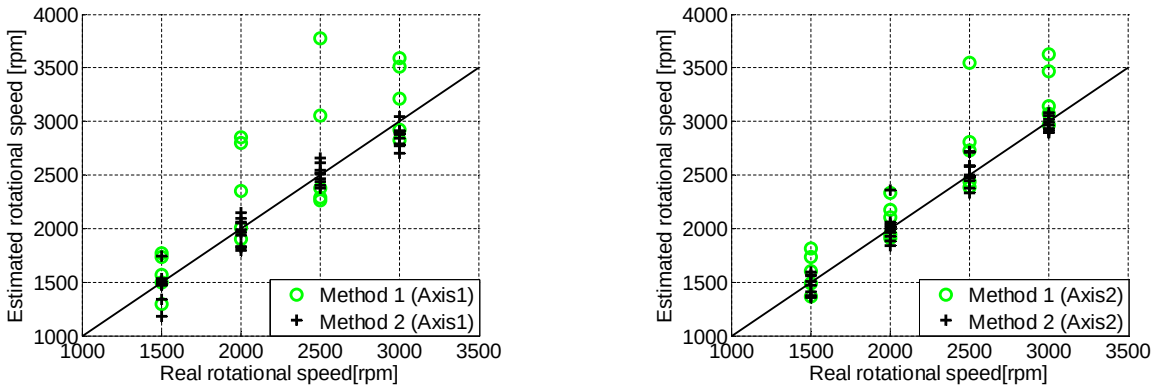
In the experiments, the rotational axis of the ball is set as two cases; Axis 1: $\boldsymbol{\omega}/\|\boldsymbol{\omega}\| = (-0.038, 0.999, -0.006)$ and Axis 2: $\boldsymbol{\omega}/\|\boldsymbol{\omega}\| = (-0.557, 0.814, 0.165)$. In both cases, the rotational speed $\|\boldsymbol{\omega}\|$ is set as four cases; from 1500 rpm to 3000 rpm with the increment of 500 rpm.

Before showing the experimental results, we will discuss the upper bound of the rotational speed that can be accurately estimated by the setup of the paper. The frame rate of the high

*The shade causes some errors for detecting the four points on the contour of the ball in the image data (See the algorithm in Section 2.2), which makes the precision of measuring the ball's velocity worse.

speed camera is 900 Hz and one side of the inscribed quadrangle in the ball image (See Figure 2.3) seems to be about a quarter of the ball's great circle. Therefore if the rotational speed is over than 13500 rpm (=900×60×$\frac{1}{4}$), $(k+1)$th frame of image data cannot overlap kth frame of image data, which means that the proposed method cannot work. It is easy to imagine that the greater the overlap of two consecutive frame of image data, a better measurement precision is obtained. If the overlap should be more than two thirds or three quarters of the inscribed quadrangle, the upper bound of the rotational speed that can be estimated by the setup of the paper is about 4500 rpm or 3400 rpm, respectively.

Figure 2.14 shows the experimental results on the estimated rotational speed, where Figure 2.14 (a) and (b) are respectively the cases of axis 1 and 2, and also the black pluses are given by the proposed method (Method 2); the green circles are given by the previous method (Method 1) [35], respectively. It shows that 89% of the estimation errors are smaller than 200 rpm irrespective of both rotational axis and speed. In addition, the proposed method (Method 2) achieves better performances than the previous method (Method 1) [35].



(a) Case of Axis 1

(b) Case of Axis 2

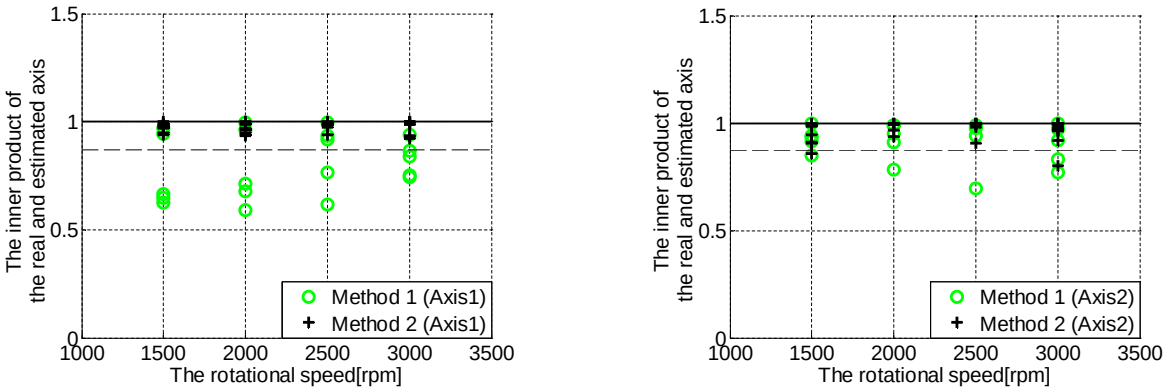
Figure 2.14: The estimated rotational speed

The experimental results on the estimated rotational axis are shown in Figure 2.15, where Figure 2.15 (a) and (b) are respectively the cases of axis 1 and 2. The results in those figures are illustrated by the inner product between the real axis and the estimated axis; suppose that $\boldsymbol{\omega}_{real}$ and $\boldsymbol{\omega}_{est}$ are respectively the real and the estimated rotational velocities, then the results show the inner product $\langle \boldsymbol{\omega}_{real}/\|\boldsymbol{\omega}_{real}\|, \boldsymbol{\omega}_{est}/\|\boldsymbol{\omega}_{est}\| \rangle$. Note that it is in a range of [-1, 1] and the inner product 1 means the estimated axis is completely equal to the real one. As shown in Figure 2.15, the black pluses are given by Method 2; the green circles are given by Method 1 [35]. The results display that 75% of the inner products in the proposed method are greater than 0.97, which means the estimation errors are less than 15 deg. Those results are also much better than those of Method 1 [35].

The processing time with PC (Operating System: Windows XP sp2; CPU: Intel(R) Xeon(R) E5430, 2.66GHz; Physical Memory: 2.00GB RAM) is in the range [30, 65] ms, which depends on the iteration number IN defined in (2.35).

We assume that a table tennis ball reaches the robot in 500∼700 ms after the opponent player hits the ball. As mentioned in Chapter 1, besides of the proposed image processing, i.e. the estimation of the translational and rotational velocities of the ball, the following two processes are needed.

- Predict the ball's trajectory by using the data of estimated ball's velocities, the aerody-



(a) Case of Axis 1 (b) Case of Axis 2

Figure 2.15: The estimated rotational axis

namic model and collision models. (It is known by our research group that this process takes about 20 ms.)

- Plan the robot motion by the data of ball's trajectory and a given destination of returned ball and then control the robot to the hit position at a specified time with a specified posture of racket. (It is known by our research group that this process takes about 400 ms.)

Therefore the total processing time would be less than 500 ms, thus the proposed algorithm of estimating the ball's velocities is fast enough for the real-time situation.

2.5.2 Experiments with a catapult machine

In this experiment, a table tennis ball is shot from a ball catapult machine (Figure 2.16). This machine has 10 speed scales marked with the number from 1 to 10 and the ball catapults out with higher speeds of both translation and rotation when the scale is set bigger. And also the machine can control the rotational axis of the ball such as topspin, backspin, etc.

The experiment measures the translational and rotational velocities of the ball immediately after the ball catapults out from the machine. The machine's speed scale is set as three cases of the marks 3, 4 and 5 which correspond to almost same speeds as human players hit with. The rotational axis is set as topspin.

Tables 2.1, 2.2 and 2.3 show the results of both translational and rotational velocities estimated by the proposed method in the case of the speed scale 3, 4 and 5, respectively, where two trials, Trial 1 and Trial 2, are shown in Table 2.2 of the speed scale 4. All the estimated velocities seem reasonable and the processing times are short enough. Note that Trial 1 and Trial 2 in Table 2.2 have almost same results but those are not exactly same. This also seems to be reasonable, because the catapult machine set at the same scale shoots balls with almost same translational and rotational velocities, but those velocities are not exactly same.

Figure 2.17 shows the iteration processes of the objective function $E(\boldsymbol{\omega})$ defined in (2.32) and we can see that the objective functions decrease greatly in about 10 iterations and converge to constants less than about 900 in all the cases. Note that the objective function with 900 corresponds to that the residual image, $(I^{k+1} - \hat{I}^{k+1})$, has the absolute value of 30 every pixel on the average. Figure 2.18 describes the iteration processes of the estimated rotational speed, from which we can say that the estimated speeds also converge to the stable values in about 10 iterations in all the cases.

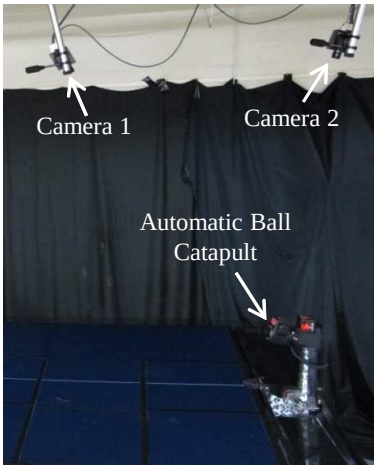


Figure 2.16: Experimental system with catapult machine

Table 2.1: The estimated results on speed scale 3

Translational velocity $\|\boldsymbol{v}_b\| \times \frac{\boldsymbol{v}_b}{\|\boldsymbol{v}_b\|}$ [m/s]	5.57 × (-0.97, 0.12, 0.19)
Rotational velocity $\|\boldsymbol{\omega}\| \times \frac{\boldsymbol{\omega}}{\|\boldsymbol{\omega}\|}$ [rpm]	2737 × (0.13, 0.99, 0.06)
Processing time [ms]	32
Iteration number *IN*	8

Table 2.2: The estimated results on speed scale 4

(Trial 1)

Translational velocity $\|\boldsymbol{v}_b\| \times \frac{\boldsymbol{v}_b}{\|\boldsymbol{v}_b\|}$ [m/s]	6.37 × (-0.98, 0.15, 0.15)
Rotational velocity $\|\boldsymbol{\omega}\| \times \frac{\boldsymbol{\omega}}{\|\boldsymbol{\omega}\|}$ [rpm]	3188 × (0.10, 0.99, -0.09)
Processing time [ms]	47
Iteration number *IN*	10

(Trial 2)

Translational velocity $\|\boldsymbol{v}_b\| \times \frac{\boldsymbol{v}_b}{\|\boldsymbol{v}_b\|}$ [m/s]	6.25 × (-0.99, 0.08, 0.06)
Rotational velocity $\|\boldsymbol{\omega}\| \times \frac{\boldsymbol{\omega}}{\|\boldsymbol{\omega}\|}$ [rpm]	3253 × (0.01, 0.99, 0.04)
Processing time [ms]	47
Iteration number *IN*	10

Table 2.3: The estimated results on speed scale 5

Translational velocity $\|\boldsymbol{v}_b\| \times \frac{\boldsymbol{v}_b}{\|\boldsymbol{v}_b\|}$ [m/s]	7.29 × (-0.98, 0.14, 0.17)
Rotational velocity $\|\boldsymbol{\omega}\| \times \frac{\boldsymbol{\omega}}{\|\boldsymbol{\omega}\|}$ [rpm]	3630 × (0.14, 0.99, 0.02)
Processing time [ms]	47
Iteration number *IN*	12

Compared Trial 1 with Trial 2 in the case of the speed scale 4, Figure 2.17 shows that the final value of the objective function is bigger in Trial 2 than in Trial 1, but Figure 2.18 shows that the estimated rotational speeds converge to plausible values in both cases. Now we will compare the image data obtained by the proposed method for Trial 1 and Trial 2.

Figures 2.19 and 2.20 demonstrate the comparison of an estimated image data with a sensed one in Trial 1 and Trial 2, respectively in the case of the speed scale 4.

In each figure, (a) represents the sensed kth frame of image data and (b) is the image

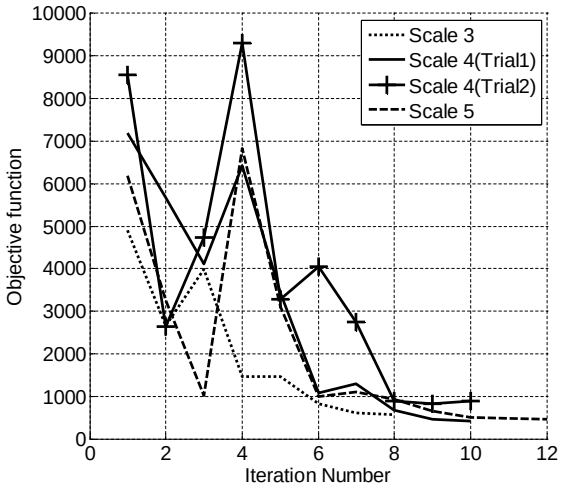


Figure 2.17: The iteration process of the objective function

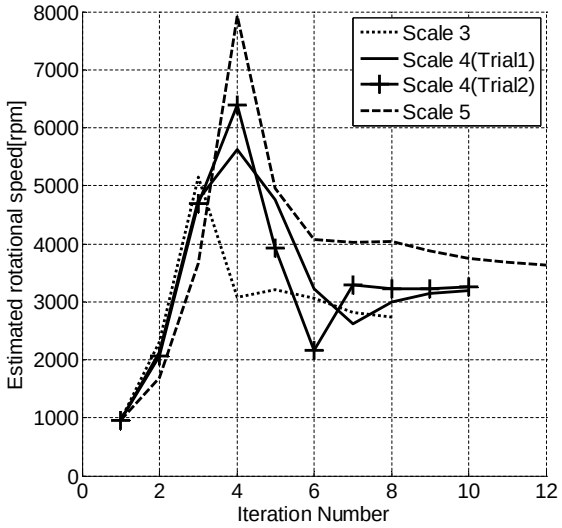


Figure 2.18: The iteration process of the estimated rotational speed

intensities, I^k, in the inscribed quadrangle of the ball image (a). (c) is the estimated image intensities, $\hat{I}^{k+1}$, of $(k+1)$th frame by using (b) and the estimated rotational velocity in Table 2.2. On the other hand, (d) represents the sensed $(k+1)$th frame of image data and (e) is the image intensities, I^{k+1}, in the inscribed quadrangle of the ball image (d). (f) is the residual image obtained by $(\hat{I}^{k+1} - I^{k+1})$. We can see that most of the value in (f) are very small, which means the proposed method works well. Compared Figure 2.20's (f) with Figure 2.19's (f), the values are larger in Figure 2.20 than in Figure 2.19. (Recall that the final value of the objective function in Trial 2 is greater than one in Trial 1.) This may be caused by some shades on the ball surface, which you can see as shades in the area of warm color in (b) and (e) of Figure 2.20.

From the above observations, it may be possible to say that the shade on the ball surface affects the final values of the objective functions, but the proposed least square method suppresses that effect to estimate the rotational velocity.

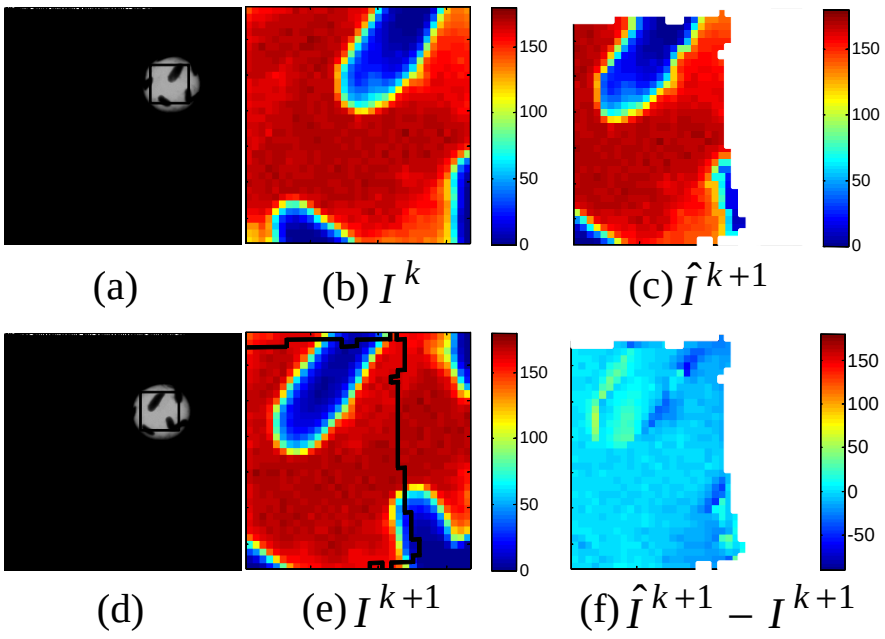


Figure 2.19: Image transformation result on Trial 1, the speed scale 4

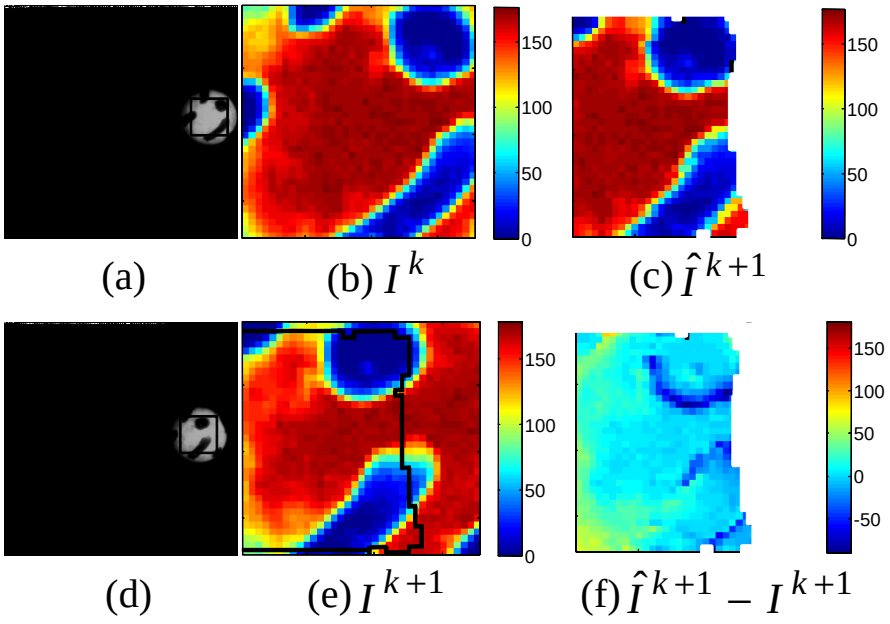


Figure 2.20: Image transformation result on Trial 2, the speed scale 4

2.6 Summary

In this chapter, we proposed the *on-line* algorithm for measuring the translational and rotational velocities of a table tennis ball. Figure 2.21 displays the estimation process in detail.

With respect to estimating the translational velocity, one of the important processes is how to estimate the center of the ball image. The proposed method estimates the ball's center on the image by using the nearest point to the perpendicular bisectors between four points on the contour of the ball image.

The rotational velocity is estimated by the image registration method, where the method of

estimating $(k+1)$th frame of image data is proposed in detail by using the sensed kth frame of image data and the candidate of rotational velocity. And also in order to search the most plausible rotational velocity, the conjugate gradient method is applied to minimize the intensity residuals between the estimated and the sensed frames.

The experimental results with the rotation machine show that 89% of the estimation errors of the rotational speeds are smaller than 200 rpm and 75% of the estimation errors of the rotational axes are less than 15 deg. The estimation error may be caused by two reasons: (1) the intensity difference of the same feature between the previous and the later frames; (2) the estimation accuracy of the ball's center. Moreover, the experimental results with the catapult machine demonstrate that both the translational and rotational velocities have been estimated well and the processing time is in the range of [30, 65] ms. Therefore, the proposed method is accurate and fast enough to realize a table tennis playing robot.

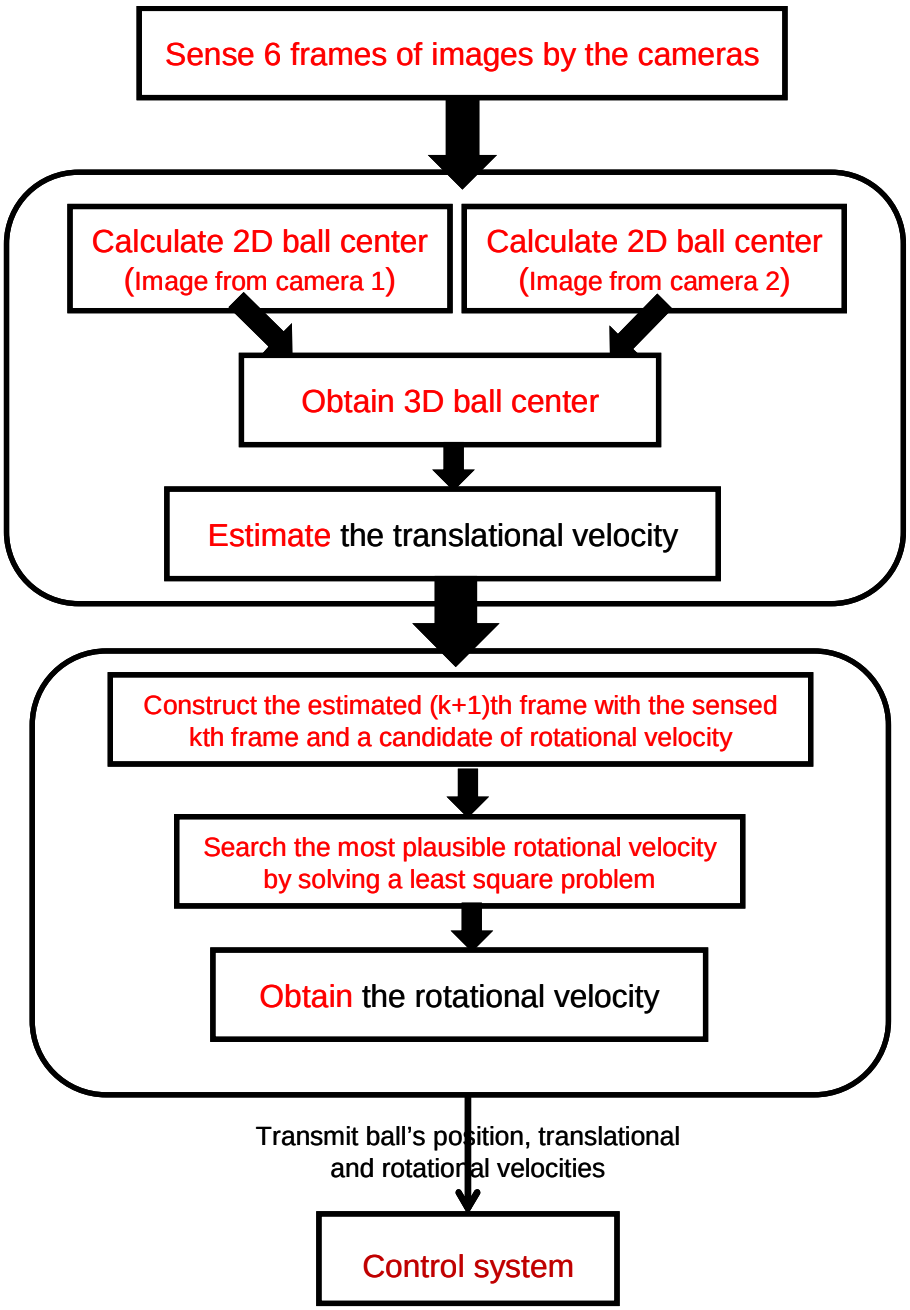


Figure 2.21: The process of estimating the translational and rotational velocities

Chapter 3

Physical Models for Predicting the Motion of the Ball

In this chapter, three physical models of the ball motion presented by our research group [36, 38] will be introduced (aerodynamics model, table rebound model and racket rebound model) and some experimental data will be displayed to verify the effectiveness of the proposed models. These three models are built by considering both the ball's *translational velocity* and *rotational velocity*, which are the basis of determining the racket striking velocity and posture for controlling the returning ball's rotational velocity (Topic of Chapter 4).

3.1 Preliminary

Predicting the ball's motion trajectory is the first essential task for the control system since the robot has to determine a striking position, striking time and the ball's velocities just before striking. Figure 3.1 shows the process of the trajectory prediction from the point the opponent shots out the ball to the point the racket strikes the ball (the blue dash line).

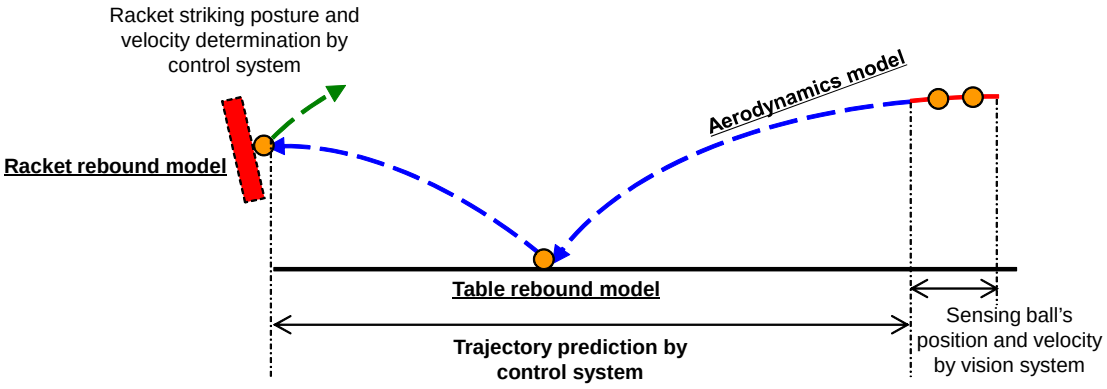


Figure 3.1: Planning the ball's motion trajectory

It is easy to see that there are two basic models needed for planning the ball's motion trajectory before striking. One is for the flight of the ball in the air and another is for the collision between the ball and the table, which are named as the aerodynamics model and the table rebound model, respectively (See Figure 3.1). Furthermore, at the striking point, in order to determine the racket's velocity and posture at the striking time for returning the ball to a

desired position, it is necessary to build a relation between the racket's striking state and the ball's states just before and after the striking, which is named as a racket rebound model. These three physical models have been presented in our previous research [36, 37, 38].

The presented models have one important improvement comparing with the previous research on table tennis robot [6, 58, 60] that they [36, 37, 38] are exact analytical models for the ball motion since not only the ball's translational velocity but also the rotational velocity are included in the models. It is well-known that "table tennis ball is much more difficult than other ball playing games such as tennis ball, base ball, et.al.". Actually, this difficulty just comes from the ball's various kinds of rotational velocities. Therefore, in order to develop a table tennis robot which has high playing techniques, the rotational velocity of the ball should not be ignored in the models.

The followings are some important coordinates and variables which will be used in this chapter for describing the rebounding phenomena.

Σ_B : on a corner of the table (Same as Chapter 2);
Σ_R : on the center of the racket and the z axis is vertical to the racket′s plane;
Σ_b : on the center of the ball (Same as Chapter 2).

In the aerodynamics model, $\boldsymbol{p}_b \in \mathbb{R}^3$ is the position of the ball's center in Σ_B. In the rebound models (table and racket), $(\boldsymbol{v}_b, \boldsymbol{\omega}_b)$ and $(\boldsymbol{v}_b', \boldsymbol{\omega}_b')$ express the ball's translational and rotational velocities just before and after rebounding. However, in the table rebound model, the variables are under Σ_B; in the racket rebound model, the variables are under Σ_R, respectively.

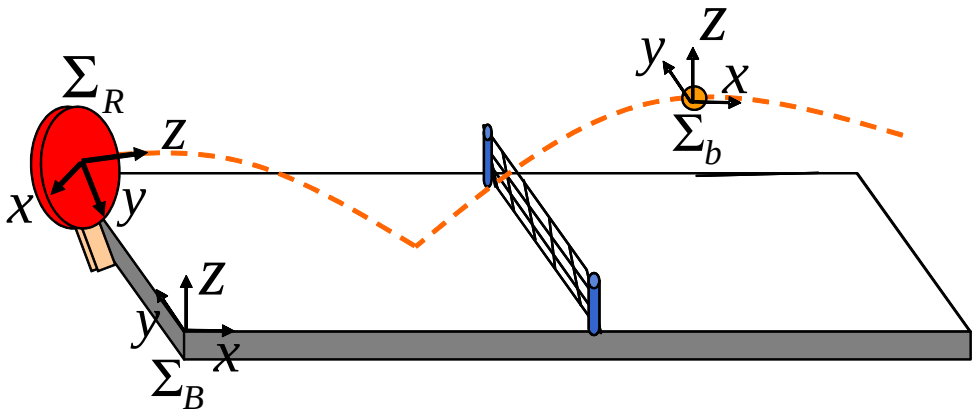


Figure 3.2: Coordinate systems of table, racket and ball

It should be noted that the rebound models (table, racket) are constructed based on the assumption that the table tennis ball does not deform when rebounding. In the table rebound model, the friction determines the type of the contact during the impact. In the racket rebound model, the elastic has the dominant effect and there is the assumption that the kinetic energy of the contact velocity is stored as the potential energy because of the elasticity.

3.2 Aerodynamics Model

When a table tennis ball flies in the air with translational and rotational velocities, the ball's motion is mainly affected by three forces: gravity, drag force and Magnus force, which can be

given as follows.

$$\ddot{\boldsymbol{p}}_b(t) = -\boldsymbol{g} - C_D(t)\frac{\rho}{m}S_b\|\dot{\boldsymbol{p}}_b(t)\|\dot{\boldsymbol{p}}_b(t) + C_M(t)\frac{\rho}{m}V_b\boldsymbol{\omega}\times\dot{\boldsymbol{p}}_b(t) \tag{3.1}$$

where $\boldsymbol{p}_b(t) \in \mathbb{R}^3$ is the ball's position at the time t, $\boldsymbol{g} = [0, 0, g]^T$ with the acceleration of gravity $g = 9.8$ m/s^2, $S_b = \frac{1}{2}\pi r^2$, and $V_b = \frac{4}{3}\pi r^3$. Notice that $\boldsymbol{\omega} = [\omega_x, \omega_y, \omega_z]^T \in \mathbb{R}^3$ is the ball's rotational velocity which is here assumed constant when the ball flies in the air. The parameters m and ρ are the ball's mass and the air density; $m = 2.7 \times 10^{-3}$ kg and $\rho = 1.184$ kg/m^3(25°C).

The second and third terms in the right of (3.1) are the drag force and Magnus force (See Figure 3.3) in which $C_D(t)$ and $C_M(t)$ are the coefficients of them respectively, which are given by

$$C_D(t) = a_D + b_D h(\dot{\boldsymbol{p}}_b(t), \boldsymbol{\omega}), \quad C_M(t) = a_M + b_M h(\dot{\boldsymbol{p}}_b(t), \boldsymbol{\omega}) \tag{3.2}$$

where

$$h(\dot{\boldsymbol{p}}_b(t), \boldsymbol{\omega}) = \frac{\dot{p}_{bx}\omega_y - \dot{p}_{by}\omega_x}{\sqrt{(\dot{p}_{bx}\omega_y - \dot{p}_{by}\omega_x)^2 + (\dot{p}_{bx}^2 + \dot{p}_{by}^2)\omega_z^2}} \tag{3.3}$$

and $a_D = 0.505, b_D = 0.065, a_M = 0.094$ and $b_M = -0.026$.

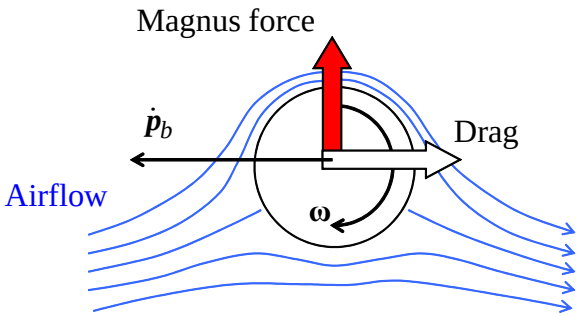


Figure 3.3: Aerodynamic model

3.3 Table Rebound Model

Figure 3.4 shows a ball collides with and rebounds from the table in which $(\boldsymbol{v}_b, \boldsymbol{\omega}_b)$ and $(\boldsymbol{v}_b', \boldsymbol{\omega}_b')$ represent the translational and rotational velocities of the ball's center just before and after the rebound in Σ_B.

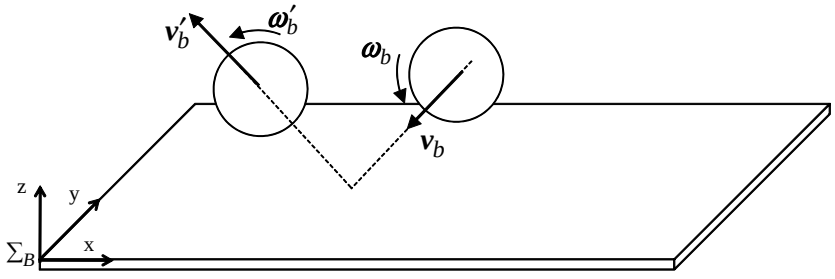


Figure 3.4: Ball's velocities just before and after rebounding from the table

Actually, $\boldsymbol{v}_b'$ just after rebounding changes due to the friction and the rotational velocity $\boldsymbol{\omega}_b$ as shown in Figure 3.5. The yellow and the red circles represent the cases of the top and back spins respectively and the ball velocities just before the rebound are same. It is easily confirmed that the velocity v_{bz}' in the case of the back spin is greater than in the case of the top spin. In addition, the rotational velocity $\boldsymbol{\omega}_b'$ also changes due to the friction. Therefore, in order to predict the ball trajectory after the rebound from the table, it is necessary to consider the friction. In fact, during the infinitesimal interval of this collision, rolling, friction and restitution, all of these happen simultaneously. Hence, $\boldsymbol{v}_b'$ and $\boldsymbol{\omega}_b'$ are determined by a complicated function of $\boldsymbol{v}_b$ and $\boldsymbol{\omega}_b$.

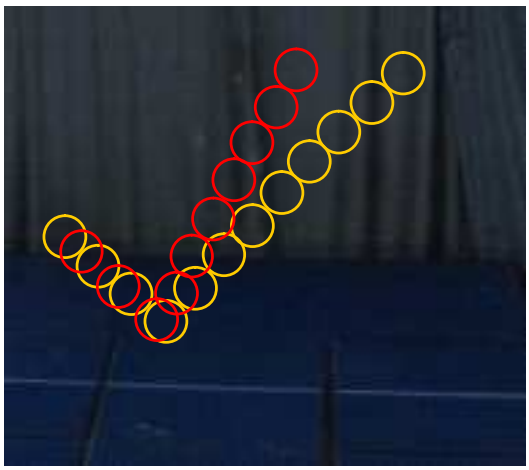


Figure 3.5: Difference of $\boldsymbol{v}_b'$ when $\boldsymbol{\omega}_b$ is the top or back spin

Now let the contact velocity $\boldsymbol{v}_{bT}$ be defined by

$$\boldsymbol{v}_{bT} := \begin{bmatrix} v_{bx} & v_{by} & 0 \end{bmatrix}^{\mathrm{T}} + \boldsymbol{\omega}_b \times \boldsymbol{r} = \begin{bmatrix} v_{bx} - r\omega_{by} \\ v_{by} + r\omega_{bx} \\ 0 \end{bmatrix} \tag{3.4}$$

where $\boldsymbol{r} = [0 \; 0 \; -r]^{\mathrm{T}} \in \mathbb{R}^3$ and $r \in \mathbb{R}_+$ is the radius of the ball. For the modeling, the following assumptions are made.

Assumption 1: During the impact of the rebound, the ball is in *a point contact* with the table. This means that any moment around z-axis does not affect on the ball during the impact.

Assumption 2: The differences between translational and angular momentums before and after the rebound equal the impulses at the rebound.

$$m\boldsymbol{v}_b' - m\boldsymbol{v}_b = \boldsymbol{P} \tag{3.5}$$

$$I\boldsymbol{\omega}_b' - I\boldsymbol{\omega}_b = \boldsymbol{r} \times \boldsymbol{P} \tag{3.6}$$

where m is the ball's mass and $I = \frac{2}{3}mr^2$ is the moment of inertia. $\boldsymbol{P} \in \mathbb{R}^3$ is the impulse in the translational direction and $\boldsymbol{r} \times \boldsymbol{P}$ is given as the impulse of the rotation.

Assumption 3: In the z direction, there holds the following simple relationship:

$$v_{bz}' = -e_t v_{bz} \tag{3.7}$$

where e_t is the coefficient of restitution between the ball and the table.

Assumption 4: The impulse in the x and y directions $\boldsymbol{P}_{xy} = [P_x\ P_y\ 0]^{\mathrm{T}} \in \mathbb{R}^3$ is given by

$$\boldsymbol{P}_{xy} = -\lambda \frac{\boldsymbol{v}_{bT}}{\|\boldsymbol{v}_{bT}\|},\ 0 \leq \lambda \leq \mu |P_z|, \tag{3.8}$$

where μ is the dynamic coefficient of friction between the ball and the table.

Assumption 5: The contact velocities $\boldsymbol{v}_{bT}$ and $\boldsymbol{v}'_{bT}$ just before and after the rebound are in the same direction. That is, the following relation holds,

$$\boldsymbol{v}'_{bT} = \nu \boldsymbol{v}_{bT},\ \nu \geq 0, \tag{3.9}$$

where either ($\nu \neq 0, \lambda = \mu |P_z|$) or ($\nu = 0, 0 \leq \lambda < \mu |P_z|$) happens.

Assumption 4 and Assumption 5 are necessary to express that $\boldsymbol{v}'_{bT}$ may be reduced in its magnitude from $\boldsymbol{v}_{bT}$, but the direction cannot be changed. Assumption 4 means that the impulse in the x and y directions is related to the one in the z direction, where the friction force in the x and y directions is proportional to f_z (the normal force) with μ and its direction is the opposite one of $\boldsymbol{v}_{bT}$ as shown in Figure 3.6(a). In (3.8), λ is the magnitude of $\boldsymbol{P}_{xy}$, which is equal to $\mu |P_z|$ since $\boldsymbol{v}'_{bT}$ cannot become zero. If $\boldsymbol{v}'_{bT}$ could reach to the opposite direction of $\boldsymbol{v}_{bT}$ under $\lambda = \mu |P_z|$, then $\boldsymbol{v}'_{bT}$ should be zero and λ should be among $[0, \mu |P_z|)$.

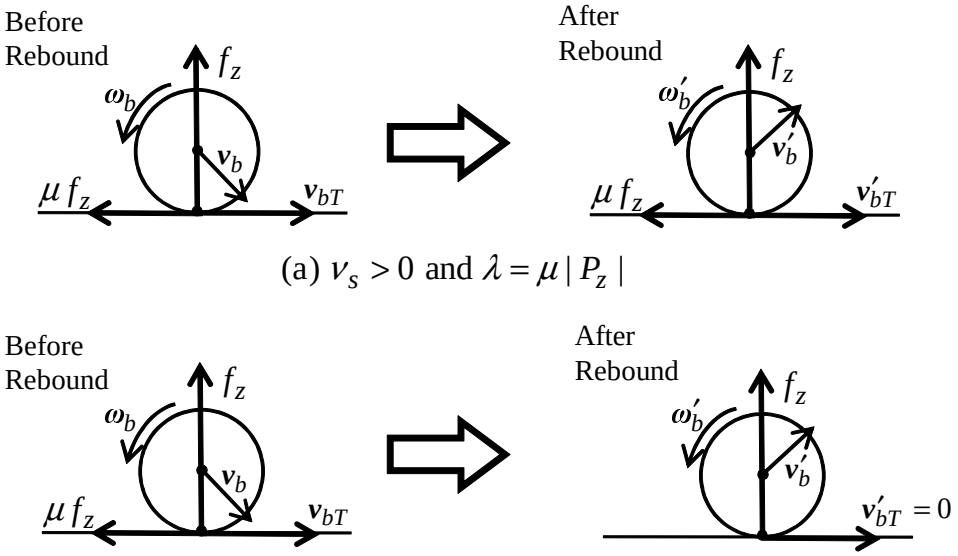


Figure 3.6: Phenomena during an infinitesimal interval of the rebound

The impulse in the z direction can be derived by combining (3.7) and (3.5):

$$P_z = -m(1 + e_t) v_{bz}. \tag{3.10}$$

In the x and y directions, the contact velocity $\boldsymbol{v}'_{bT}$ after the rebound is derived by (3.4) with (3.5), (3.6) and Assumption 4:

$$\boldsymbol{v}'_{bT} = -\lambda \left(\frac{1}{m} + \frac{r^2}{I}\right) \frac{\boldsymbol{v}_{bT}}{\|\boldsymbol{v}_{bT}\|} + \boldsymbol{v}_{bT} = \nu(\lambda) \boldsymbol{v}_{bT}, \tag{3.11}$$

where

$$\nu(\lambda) = -\frac{\lambda}{\|\boldsymbol{v}_{bT}\|} \left(\frac{1}{m} + \frac{r^2}{I}\right) + 1. \tag{3.12}$$

Combining (3.10), $I = \frac{2}{3}mr^2$ and (3.12), let ν_s be defined as

$$\nu_s = \nu(\mu|P_z|) = 1 - \frac{5}{2}\mu(1+e_t)\frac{|v_{bz}|}{\|\boldsymbol{v}_{bT}\|}. \tag{3.13}$$

It follows that (A) if $\nu_s > 0$, $\nu = \nu_s$ and $\lambda = \mu|P_z|$; (B) if $\nu_s \leq 0$, $\nu = 0$ and

$$\lambda = ||\boldsymbol{v}_{bT}||/(\frac{1}{m} + \frac{r^2}{I}) = \frac{2}{5}m||\boldsymbol{v}_{bT}||. \tag{3.14}$$

For the case (A), i.e., $\nu_s > 0$, substituting (3.8) with $\lambda = \mu|P_z|$ and (3.10) into (3.5) and (3.6), we can obtain the relations of the translational and rotational velocities before and after the rebound,

$$\boldsymbol{v}_b' = \boldsymbol{A}_{vv}\boldsymbol{v}_b + \boldsymbol{A}_{v\omega}\boldsymbol{\omega}_b, \tag{3.15}$$

$$\boldsymbol{\omega}_b' = \boldsymbol{A}_{\omega v}\boldsymbol{v}_b + \boldsymbol{A}_{\omega\omega}\boldsymbol{\omega}_b, \tag{3.16}$$

where

$$\boldsymbol{A}_{vv} := \begin{bmatrix} 1-\alpha & 0 & 0 \\ 0 & 1-\alpha & 0 \\ 0 & 0 & -e_t \end{bmatrix}, \boldsymbol{A}_{v\omega} := \begin{bmatrix} 0 & \alpha r & 0 \\ -\alpha r & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix},$$

$$\boldsymbol{A}_{\omega v} := \begin{bmatrix} 0 & -\frac{3\alpha}{2r} & 0 \\ \frac{3\alpha}{2r} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \boldsymbol{A}_{\omega\omega} := \begin{bmatrix} 1-\frac{3\alpha}{2} & 0 & 0 \\ 0 & 1-\frac{3\alpha}{2} & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

$$\alpha := \mu(1+e_t)\frac{|v_{bz}|}{\|\boldsymbol{v}_{bT}\|}. \tag{3.17}$$

For the case (B), i.e., $\nu_s \leq 0$, by using $\lambda = \frac{2}{5}m\|\boldsymbol{v}_{bT}\|$ and the similar calculation of the case (A), we can get the coefficient matrices of (3.15) and (3.16) as

$$\boldsymbol{A}_{vv} := \begin{bmatrix} \frac{3}{5} & 0 & 0 \\ 0 & \frac{3}{5} & 0 \\ 0 & 0 & -e_t \end{bmatrix}, \boldsymbol{A}_{v\omega} := \begin{bmatrix} 0 & \frac{2r}{5} & 0 \\ -\frac{2r}{5} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix},$$

$$\boldsymbol{A}_{\omega v} := \begin{bmatrix} 0 & -\frac{3}{5r} & 0 \\ \frac{3}{5r} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \boldsymbol{A}_{\omega\omega} := \begin{bmatrix} \frac{2}{5} & 0 & 0 \\ 0 & \frac{2}{5} & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{3.18}$$

Notice here the parameters of the ball are $m = 2.7 \times 10^{-3}$ kg and $r = 2.0 \times 10^{-2}$ m. The dynamic coefficient of friction is $\mu = 0.25$ and the coefficient of restitution between the ball and the table is $e_t = 0.93$, which are obtained by the experimental data [36].

3.4 Racket Rebound Model

Figure 3.7 shows an example of the ball rebounding from the racket rubber. The green circles represent the same point on the ball. It is confirmed that the rotational velocity about the axis normal to the image plane changes to the inverse direction after the rebound. This implies that unlike the collision between the ball and the table, $\boldsymbol{v}_{bT}'$ can change the direction from $\boldsymbol{v}_{bT}$ in the collision between the ball and the racket. It can not be explained by only using the friction and the same of the table rebound model. Actually, the rubber stores elastic energy

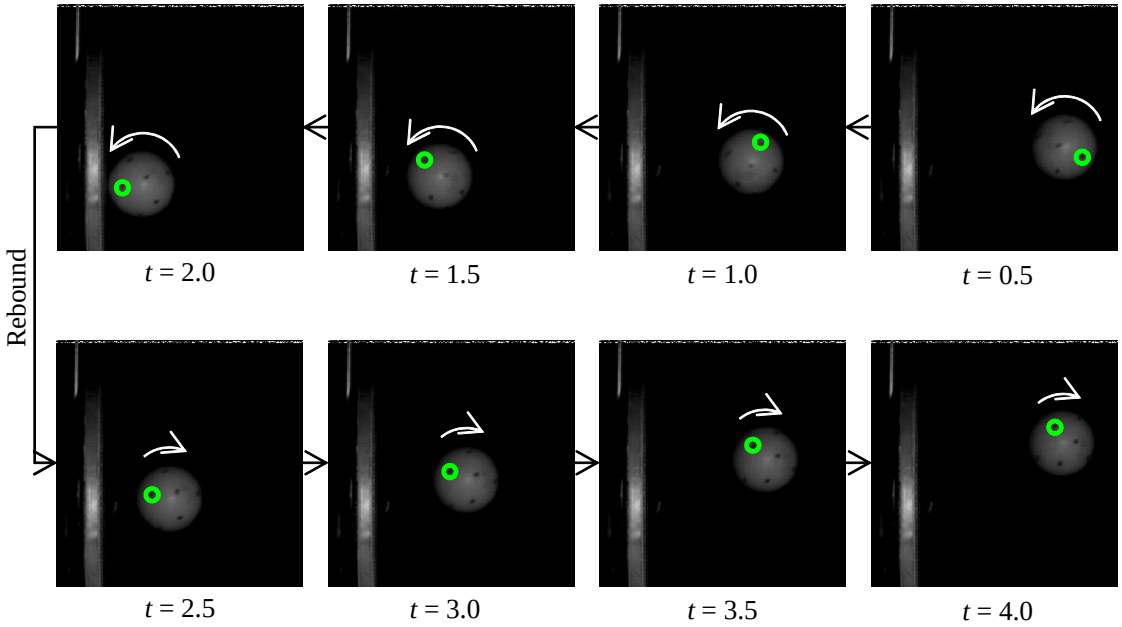


Figure 3.7: Rebound of the ball from the racket rubber

at the collision process. In order to achieve desired ball trajectory after the rebound from the racket, it is necessary to consider this phenomenon.

Figure 3.8 shows the rebound between the ball and the racket rubber in which $(\boldsymbol{v}_b, \boldsymbol{\omega}_b)$ and $(\boldsymbol{v}_b', \boldsymbol{\omega}_b')$ represent the translational and rotational velocities of the ball's center just before and after the striking with respect to the frame Σ_R.

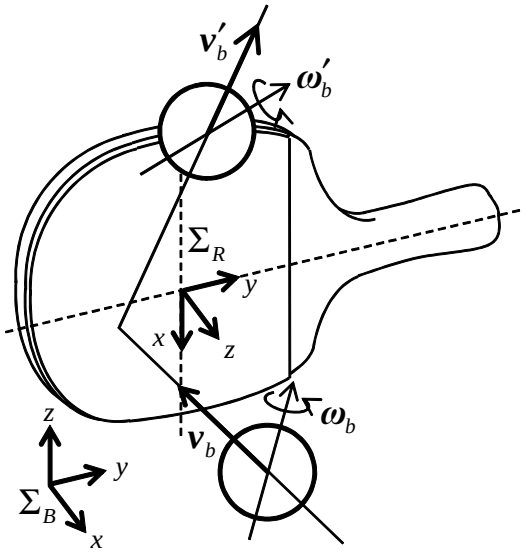


Figure 3.8: Racket rebound model

Assumptions 1-3 are also made here.

Instead of assumptions 4 and 5, in order to express the effect of the elasticity parallel to the surface, we model the tangent motion of the racket rubber as the motion of the virtual mass m_α with the spring k_s and the displacement $\boldsymbol{s}(t) \in \mathbb{R}^3$ as shown in Figure 3.9. Notice that m_α is the equivalent mass consisting of the mass of the ball and the deformed area of the rubber. For the model, we make the following assumption.

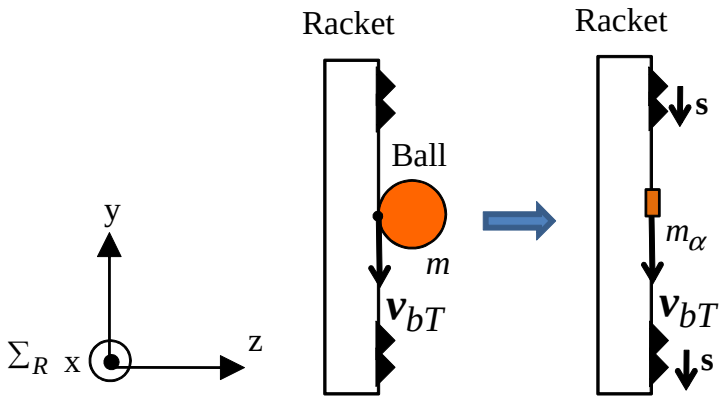


Figure 3.9: Visual spring model parallel to the surface

Assumption 4': The impulse $\boldsymbol{P}_{xy} \in \mathbb{R}^3$ in the x and y directions is given as

$$\boldsymbol{P}_{xy} = -k_p \boldsymbol{v}_{bT}, \tag{3.19}$$

where k_p is a constant fixed from the experimental data.

This assumption is comming from the following consideration.

Imagine that the kinetic energy of the visual mass is conserved as the elasticity energy without any dissipation. Suppose that $\boldsymbol{s}_{max} = [s_{xmax}, s_{ymax}, 0]^T \in \mathbb{R}^3$ is the maximum displacement of the virtual mass with the same direction of $\boldsymbol{v}_{bT}$ and the rubber's elasticity is uniform in any direction with its stiffness k_s. Let the infinitesimal interval of the collision be T, if the displacement $\boldsymbol{s}(t)$ has a constant velocity $\boldsymbol{v}_s = \frac{\boldsymbol{s}_{max}}{T}$, then it is easy to see that $\boldsymbol{s}(t) = \boldsymbol{v}_s t (0 \leq t \leq T)$ and

$$\boldsymbol{P}_{xy} = \int_0^T -k_s \boldsymbol{s}(t) dt = -k_s \boldsymbol{v}_s \int_0^T t dt = -\frac{1}{2} k_s T \boldsymbol{v}_s^2 = -\frac{1}{2} k_s T \boldsymbol{s}_{max}. \tag{3.20}$$

On the other hand, from

$$\frac{1}{2} m_\alpha \|\boldsymbol{v}_{bT}\|^2 = \frac{1}{2} k_s \|\boldsymbol{s}_{max}\|^2, \frac{\boldsymbol{v}_{bT}}{\|\boldsymbol{v}_{bT}\|} = \frac{\boldsymbol{s}_{max}}{\|\boldsymbol{s}_{max}\|}, \tag{3.21}$$

we get

$$\boldsymbol{s}_{max} = \sqrt{\frac{m_\alpha}{k_s}} \boldsymbol{v}_{bT}. \tag{3.22}$$

Therefore,

$$\boldsymbol{P}_{xy} = -\frac{1}{2} k_s T \sqrt{\frac{m_\alpha}{k_s}} \boldsymbol{v}_{bT} = -k_p \boldsymbol{v}_{bT}, \tag{3.23}$$

where

$$k_p = \frac{1}{2} T \sqrt{m_\alpha k_s}. \tag{3.24}$$

From assumptions 1-3, i.e., (3.5)-(3.7) and assumption 4', i.e., (3.20), we obtain the Racket Rebound Model in the frame Σ_R as

$$\boldsymbol{v}_b' = \boldsymbol{A}_{vv} \boldsymbol{v}_b + \boldsymbol{A}_{v\omega} \boldsymbol{\omega}_b, \tag{3.25}$$

$$\boldsymbol{\omega}_b' = \boldsymbol{A}_{\omega v} \boldsymbol{v}_b + \boldsymbol{A}_{\omega\omega} \boldsymbol{\omega}_b, \tag{3.26}$$

where

$$\boldsymbol{A}_{vv} := \begin{bmatrix} 1-k_{pv} & 0 & 0 \\ 0 & 1-k_{pv} & 0 \\ 0 & 0 & -e_r \end{bmatrix}, \boldsymbol{A}_{v\omega} := k_{pv} \begin{bmatrix} 0 & r & 0 \\ -r & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix},$$

$$\boldsymbol{A}_{\omega v} := k_{p\omega} \begin{bmatrix} 0 & -r & 0 \\ r & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \boldsymbol{A}_{\omega\omega} := \begin{bmatrix} 1-k_{p\omega}r^2 & 0 & 0 \\ 0 & 1-k_{p\omega}r^2 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

and

$$k_{pv} := \frac{k_p}{m}, k_{p\omega} := \frac{k_p}{I}. \tag{3.27}$$

The parameters e_r = 0.73, $k_{pv} = 0.615$ and $k_{p\omega} = 2570$ are obtained from experiments [36] .

Now, we will consider the model (3.25) and (3.26) in the frame Σ_B. Suppose the racket's striking posture and velocity are given by (α, β) and ${}^B\boldsymbol{V}$ in Σ_B, respectively. Then, the racket rebound model can be written as

$$\begin{bmatrix} \boldsymbol{R}_R^{\mathrm{T}} & 0 \\ 0 & \boldsymbol{R}_R^{\mathrm{T}} \end{bmatrix} \begin{bmatrix} {}^B\boldsymbol{v}_b' - {}^B\boldsymbol{V} \\ {}^B\boldsymbol{\omega}_b' \end{bmatrix} = \begin{bmatrix} \boldsymbol{A}_{vv} & \boldsymbol{A}_{v\omega} \\ \boldsymbol{A}_{\omega v} & \boldsymbol{A}_{\omega\omega} \end{bmatrix} \begin{bmatrix} \boldsymbol{R}_R^{\mathrm{T}} & 0 \\ 0 & \boldsymbol{R}_R^{\mathrm{T}} \end{bmatrix} \begin{bmatrix} {}^B\boldsymbol{v}_b - {}^B\boldsymbol{V} \\ {}^B\boldsymbol{\omega}_b \end{bmatrix} \tag{3.28}$$

where ${}^B\boldsymbol{v}_b$ and ${}^B\boldsymbol{v}_b'$ are the ball's translational velocities in Σ_B just before and after the striking, and ${}^B\boldsymbol{\omega}_b$ and ${}^B\boldsymbol{\omega}_b'$ are the ball's rotational velocities in Σ_B just before and after the striking. $\boldsymbol{R}_R$ is the rotational matrix from Σ_B to Σ_R, which is given by

$$R_R = \begin{bmatrix} \cos\beta & \sin\beta\sin\alpha & \sin\beta\cos\alpha \\ 0 & \cos\alpha & -\sin\alpha \\ -\sin\beta & \cos\beta\sin\alpha & \cos\beta\cos\alpha \end{bmatrix}.$$

3.5 Experimental Verification of the Three Models

3.5.1 The case of the aerodynamics model

In order to identify the coefficients of the drag force and Magnus force, (C_D, C_M) in the aerodynamics model (3.1), the ball's rotational velocity $\boldsymbol{\omega}$ is assumed to be constant during the flying. This assumption has been verified in [37]. The ball position $\boldsymbol{p}_b \in \mathbb{R}^3$ is measured widely by two middle speed cameras (150 Hz) of Radish System (Library, Co.). The coefficients are identified in [37, 39] by minimizing the difference of the trajectories of the measured ball and the numerical solution of the aerodynamics model (See Figure 3.10) with the following objective function:

$$\min_{\boldsymbol{C}} \boldsymbol{V}(\boldsymbol{C}), \tag{3.29}$$

where $\boldsymbol{C} := [C_D, C_M]^T$ and

$$\boldsymbol{V}(\boldsymbol{C}) := \sum_{j=1}^{N_t} \frac{\boldsymbol{V}_j(\boldsymbol{C})}{N_t}, \quad \boldsymbol{V}_j(\boldsymbol{C}) := \sum_{i=1}^{N_j} \frac{1}{N_j} \| \boldsymbol{p}_{b_j}(t_i) - \hat{\boldsymbol{p}}_{b_j}(t_i, \boldsymbol{C}) \|^2$$

in which N_t is the number of all the experimental trials and N_j is the number of the measured data at the *j*th trial. The sampled time t_i is defined as $t_i := i\Delta t$, $\Delta t = 1/150$ s. $\boldsymbol{p}_{b_j}$ and $\hat{\boldsymbol{p}}_{b_j}$

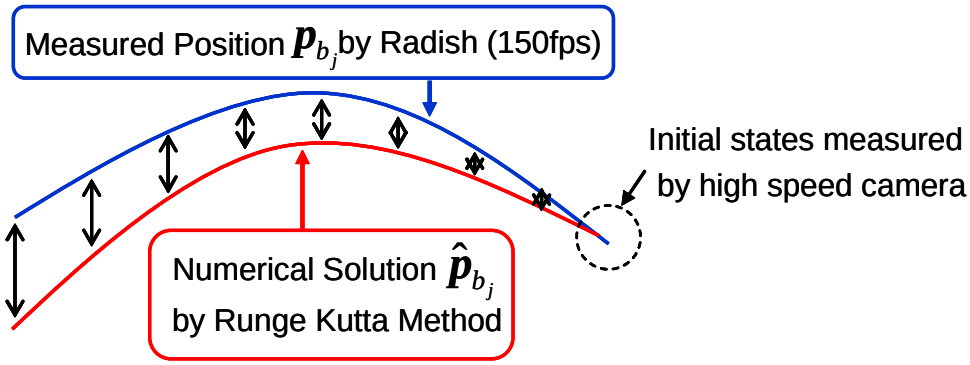
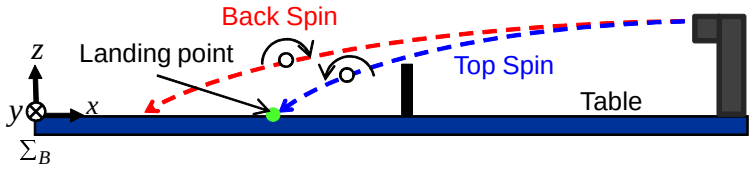


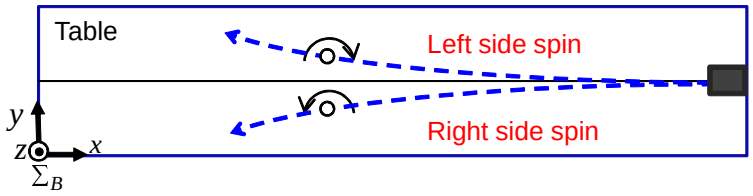
Figure 3.10: Coefficients identification

are the measured and simulated positions. $\hat{p}_{b_j}$ is solved in the interval $[t_1, t_{N_j}]$ by the fourth-order Runge-Kutta method with the initial states of $(\hat{p}_{b_j}, \dot{\hat{p}}_{b_j}, \hat{\omega}_j)$ measured by the high speed cameras around the ball catapult.

The aerodynamics model is verified by 4 cases [37, 39]: (a) Top spin, (b) Back spin, (c) Left side spin and (d) Right side spin. Note here that the experimental data for Case (a) and Case (b) are pure top spin ($\omega_y < 0$) and pure back spin ($\omega_y > 0$) with $\dot{p}_{by}, \omega_x, \omega_z \simeq 0$ and the experimental data for Case (c) and Case (d) are pure left side spin ($\omega_z < 0$) and pure right side spin ($\omega_z > 0$) with $\omega_x, \omega_y \simeq 0$. Figure 3.11 displays the flying trajectories of the ball under the above 4 cases. In the verification experiments, for each spin case, the balls were shot from the automatic catapult machine with three speed scales (3, 4, 5).



(a) Ball trajectories of the top and back spins



(b) Ball trajectories of the left and right side spins

Figure 3.11: The flying trajectories of the ball

Table 3.1 is the initial states of the top spin balls. Since the ball's flying distance in x-axis is much longer than that in y and z axis, the differences of the landing points in x-axis between the measured ones and the predicted ones by the aerodynamics model are shown in Table 3.2. Figure 3.12 demonstrates the ball's flying distance in x-axis from the position detected by the high speed cameras to the position landing on the table. The horizontal and vertical axes express the ball's flying distances obtained by models and measured by the middle speed cameras, respectively. The solid line represents the case where these values are same. The markers of the pink squares, the blue circles and the green stars ("3t", "4t", "5t") represent top spin balls which are shot in Speed Scale 3, 4 and 5, respectively. These results are calculated

with the aerodynamics model. The black markers are obtained with the free falling model. It can be seen easily that most of the markers obtained by the aerodynamics model are near to the solid line, which are much better than the ones obtained by the free falling model.

Table 3.1: Initial states of the balls for top spin verification

Speed Scale	$\dot{p}_{bx}$ [m/s]	$\dot{p}_{bz}$ [m/s]	ω_y [rad/s]
3 top	-4.99±0.09	2.53±0.082	-300±10
4 top	-6.22±0.11	1.07±0.15	-360±23
5 top	-7.46±0.076	1.68±0.087	-400±3.5

Table 3.2: The difference of the landing points in x-axis between the measured ones and the predicted ones [m] (Verification of the top spin)

Speed Scale	Aerodynamics model
3 top	0.086± 0.046
4 top	0.11± 0.059
5 top	0.075± 0.082

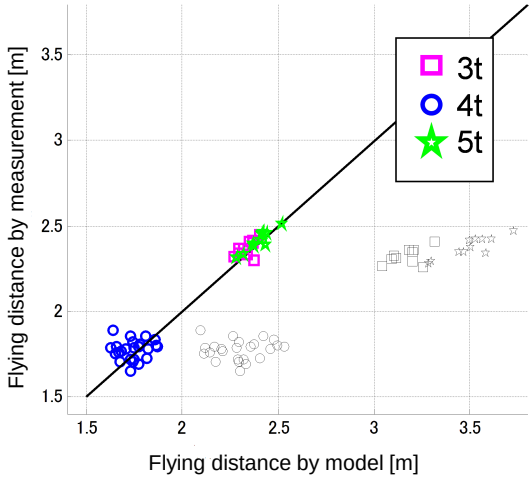


Figure 3.12: Verification of the top spin

Table 3.3 shows the initial states of the back spin balls. $\boldsymbol{V}(\boldsymbol{C})$ in (3.29) is used to verify the effectiveness of the aerodynamics model as shown in Table 3.4. Figure 3.13 demonstrates an example of the trajectory comparison in z-axis, where the lines of the blue, red, magenta and green represent the cases of the measured data, top spin, back spin and no-air resistance. In fact, this is a back spin ball. Top spin and back spin mean the coefficients (C_D, C_M) for top spin and back spin are used in the aerodynamics model. No-air resistance is the free falling situation. It is found that the magenta line is very close to the blue line and the other lines are under the blue line.

Table 3.5 is for the verification of the side spin cases, where the norm of the difference of the flying distances at about 1.8 m between the measured ones and the predicted ones by the aerodynamics model are shown. Since the side spin balls have bigger velocity in y-axis than

Table 3.3: Initial states of the balls for back spin verification

Speed Scale	$\dot{p}_{bx}$ [m/s]	$\dot{p}_{bz}$ [m/s]	ω_y [rad/s]
3 back	-5.1±0.077	1.47±0.10	270±21
4 back	-6.7±0.048	0.41±0.11	323±37
5 back	-7.4±0.051	-0.17±0.14	361±21

Table 3.4: $\boldsymbol{V}(\boldsymbol{C})$ $[m^2/N_j]$ (Verification of the back spin)

Speed Scale	Aerodynamics model
3 back	$0.52\times10^{-3}\pm 0.39\times10^{-3}$
4 back	$0.19\times10^{-3}\pm 0.38\times10^{-3}$
5 back	$0.14\times10^{-3}\pm 0.20\times10^{-3}$

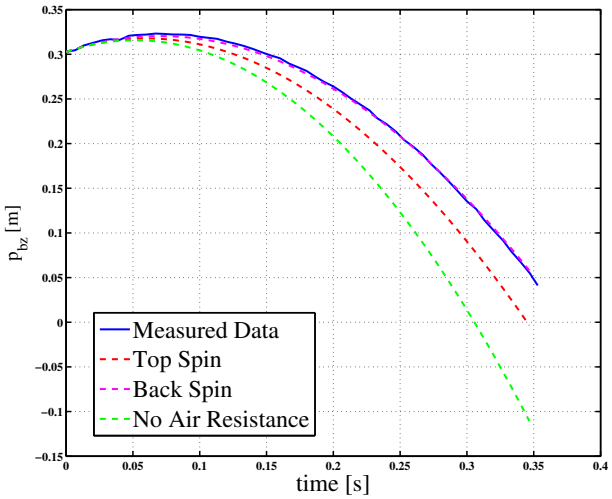


Figure 3.13: Trajectory comparison for the verification of the back spin

the pure top or pure back spins (See Figure 3.11), the differences of the flying distances in both x and y axis are displayed in Figure 3.14, where the triangles and the circles express the left and right side spin, respectively. The red color markers mean that the difference is bigger than 3.7 cm. It can be seen that most of experimental results have small differences from the measured data. Therefore, the aerodynamics model is validate for predicting the trajectory of a ball with spinning.

Table 3.5: The norm of the difference of the flying distances at about 1.8 m between the measured ones and the predicted ones [m] (Verification of the side spin)

Speed Scale	Aerodynamics model
3 left side	0.017± 0.006
4 left side	0.023± 0.010
5 left side	0.033± 0.004
3 right side	0.014± 0.008
4 right side	0.013± 0.003
5 right side	0.020± 0.008

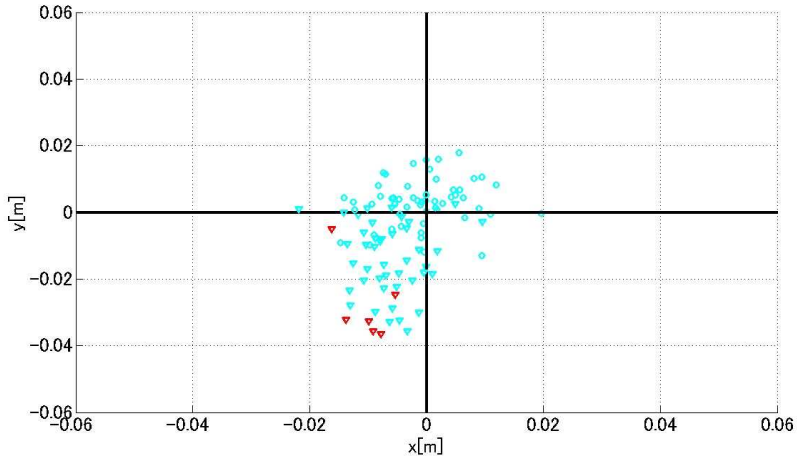


Figure 3.14: The difference of the flying distances between the measured ones and the predicted ones (Verification of the side spin)

3.5.2 The case of the table rebound model

The table rebound model is verified by 4 cases: (a) Top spin, (b) Back spin, (c) Side-top spin and (d) Side-back spin as illustrated in Figure 3.15.

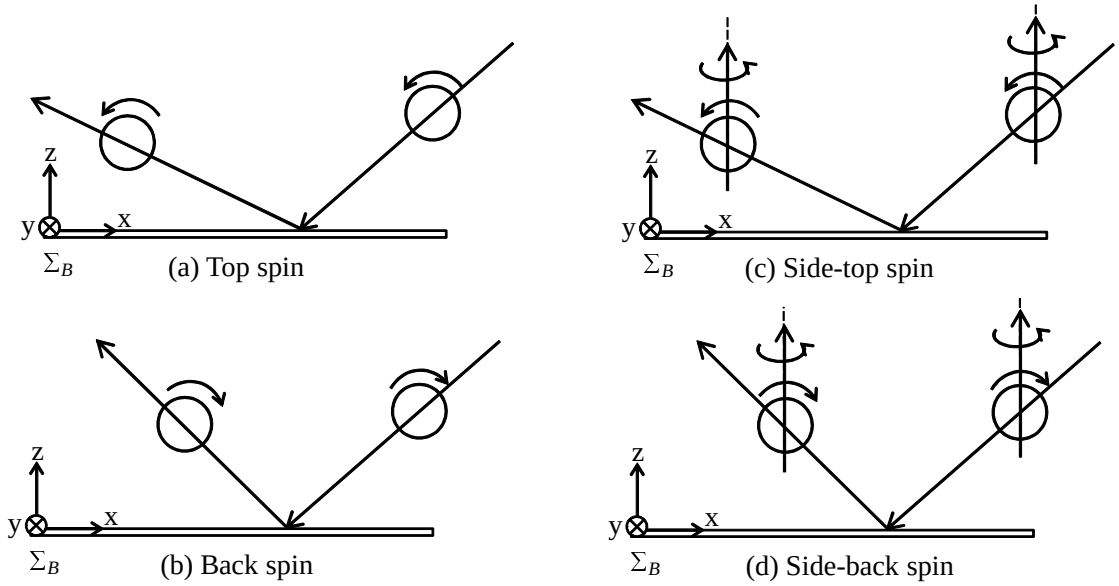


Figure 3.15: Situations of the verification of the table rebound model

Table 3.6 shows the verification of the cases of the top and back spins. Since the experimental data are pure top spin ($\omega_{by} < 0$) and pure back spin ($\omega_{by} > 0$) with $v_{by}, \omega_{bx}, \omega_{bz} \simeq 0$. Hence, only (v_{bx}, v_{bz}) m/s and ω_{by} rad/s are shown in Table 3.6. The number 1) and 2) are the cases of the top spin and the number 3)-6) are the cases of the back spin. The errors of the velocity and the rotation are $e_v = 1.0 - 11.1\%$ and $e_\omega = 1.5 - 10.5\%$, respectively.

Figure 3.16 is also for the verification of the cases of the top and back spins which are represented as the circles and the squares, respectively. The horizontal and vertical axes are the experimental values and the calculated values from the model of the translational and rotational velocities after the rebound. The solid lines represent the cases where these values are same.

Table 3.6: Verification of the table rebound model

No.	$(v_{bx}, v_{bz}, \omega_{by})$	$(v'_{bx}, v'_{bz}, \omega'_{by})$	Model
1)	(-2.8, -3.3, 279)	(-3.8, 3.0, 196)	(-4.0, 3.1, 199)
2)	(-3.6, -2.1, 216)	(-3.8, 2.0, 209)	(-3.8, 2.0, 194)
3)	(-4.1, -2.2, -286)	(-3.1, 2.0, -224)	(-3.0, 2.1, -206)
4)	(-3.8, -1.8, -216)	(-3.0, 1.7, -161)	(-2.9, 1.7, -150)
5)	(-2.7, -2.2, -170)	(-1.9, 1.9, -100)	(-1.7, 1.9, -90.6)
6)	(-2.6, -2.2, -165)	(-1.7, 2.0, -93.7)	(-1.5, 2.1, -83.9)

Figure 3.17 is used for the verification of the side-top spin case (the circles) and side-back spin case (the squares). Note that the side-top spin means $\omega_{by} < 0, \omega_{bz} > 0$ and the side-back spin means $\omega_{by} > 0, \omega_{bz} > 0$. In Figure 3.17, multi-rotational axes are displayed. It is confirmed that most of the data in Figures 3.16 and 3.17 are close to the solid lines. The few data which are a little far from the solid lines maybe caused by the quantization errors of the image data [36]. Hence, the experimental data show the validation of the rebound model on the table, even though the model is constructed under Assumption 1-5 in Subsec. 3.3.

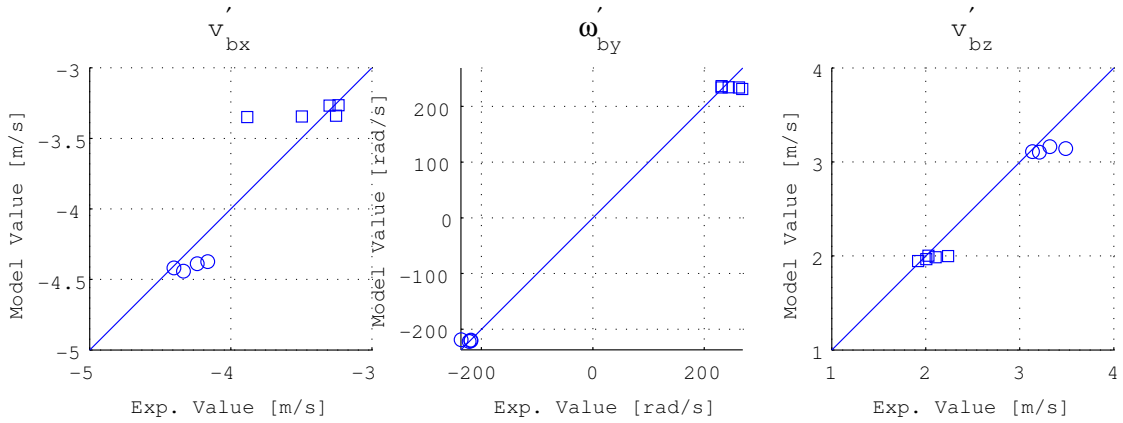


Figure 3.16: Verification of the top spin and back spin

3.5.3 The case of the racket rebound model

The rebound model on the racket is verified by 4 cases [36] of (a) Top spin, (b) Back spin, (c) Side-top spin ($v_{by} > 0$) and Side-top spin ($v_{by} < 0$) as illustrated in Figure 3.18.

Figure 3.19 shows the results of the pure top spin and pure back spin. Figure 3.20 shows the results of side-top spin of $v_{by} > 0$ and $v_{by} < 0$. The red data are used for the identification of the parameters and the blue data are used for the verification. The circles and the squares represent the top spin and the back spin in Figure 3.19 and the side-top spin of $v_{by} > 0$ and the side-top spin of $v_{by} < 0$ in Figure 3.20. It is confirmed that the red and blue data are close to the solid lines with the errors due to the quantization errors of the image data.

Table 3.7 shows some of the experimental data of the contact velocity before and after the rebound and the ones after the rebound calculated by the model. It is confirmed that the calculated velocities are in the opposite directions of the velocities before the rebound and close to the velocities after the rebound. Therefore, the proposed model can represent the specified effect of the rubber.

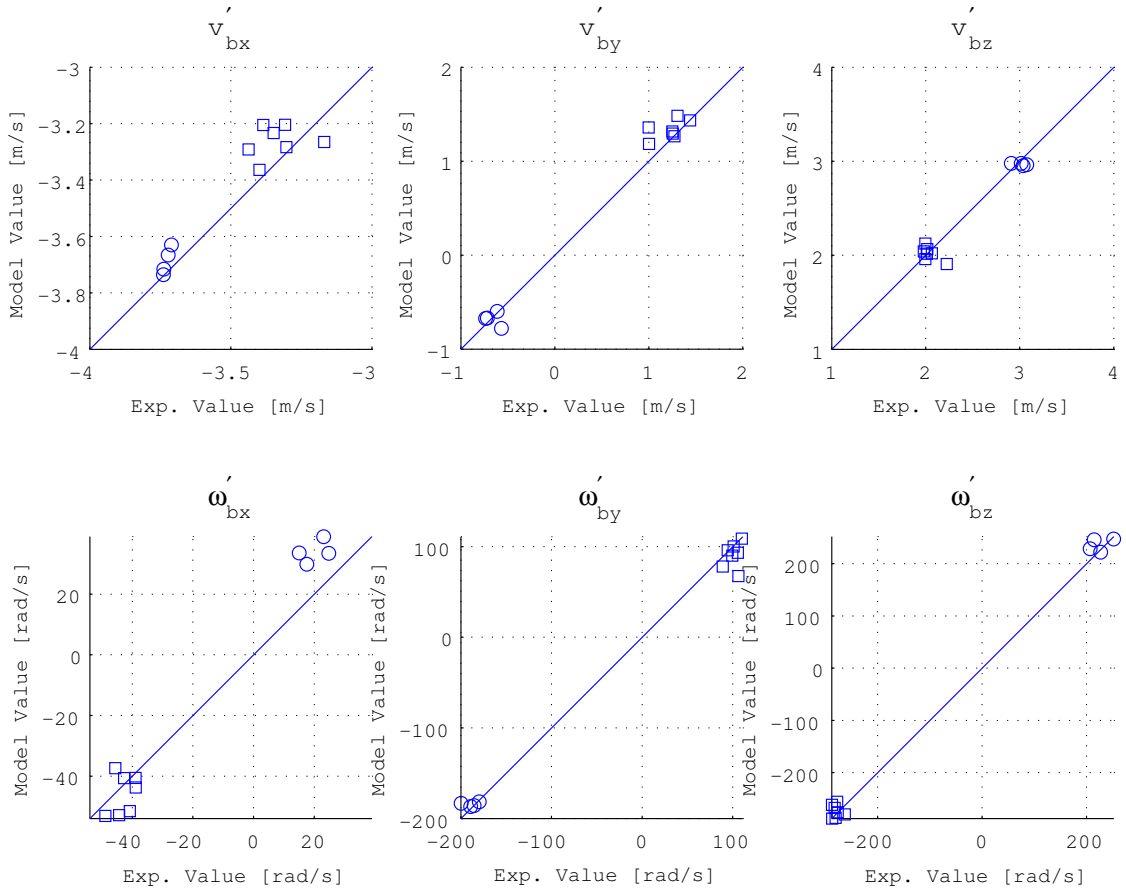


Figure 3.17: Verification of the side spin ($\omega_{by} > 0$ and $\omega_{by} < 0$)

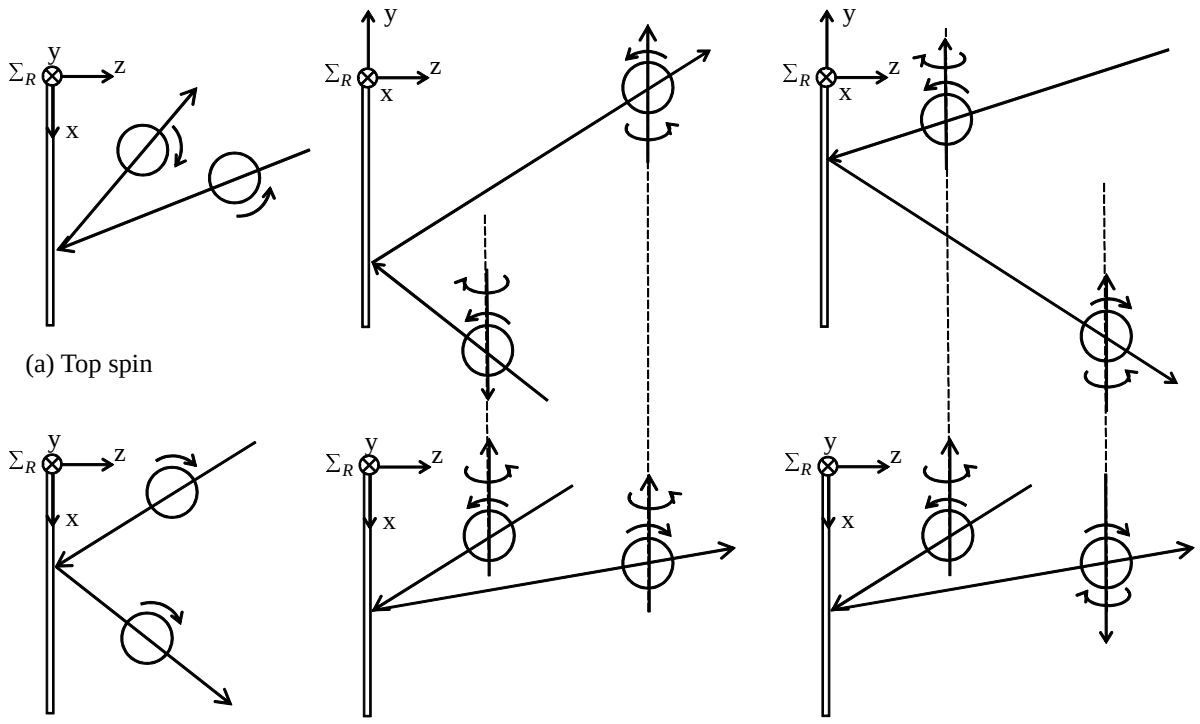
Table 3.7: Verification of the tangent velocity [m/s]

No.	Before the rebound	After the rebound	Calculated by Model
1	-3.79	2.35	2.21
2	-3.87	2.12	2.26
3	-3.77	2.04	2.20
4	-3.92	2.03	2.29
5	-3.68	2.02	2.15
6	-3.74	2.37	2.18

3.6 Summary

In this chapter, three physical models which have been presented by our research group are demonstrated in detail, which include the aerodynamics model, table rebound model and racket rebound model. The three models are built based on the ball's translational and rotational velocities in which the aerodynamics model includes the Magnus force caused by the rotational velocity of the ball; the rebound models characterize the variation of the ball's translational and rotational velocities before and after rebounding.

Some experimental data are shown to verify the effectiveness of these models, which include top spin balls, back spin balls and side spin balls. The experiments for two rebound models demonstrate that the estimated ball's velocities just after rebounding coincide well with



(a) Top spin

(b) Back spin

(c) Side-top spin ($v_{by} > 0$)

(d) Side-top spin ($v_{by} < 0$)

Figure 3.18: Situations of the verification of the racket

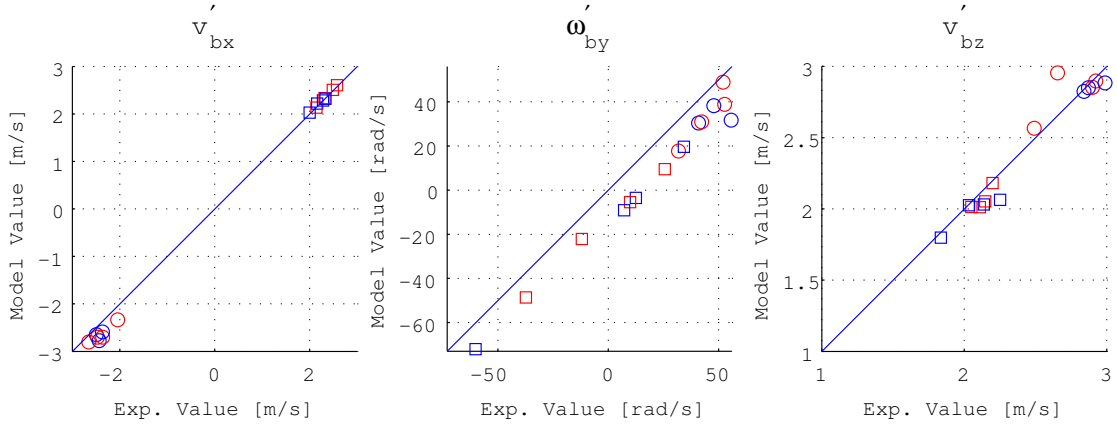


Figure 3.19: Verification of the top spin and back spin

the real velocities after rebounding and the experiments for the aerodynamics model display that the trajectory prediction is accurate enough for the racket to strike the balls, when the table rebound model is used together.

In fact, it has important meanings for building these three models since it makes the following two works become possible: (1) The robot can successfully strike the coming ball with various kinds of rotational velocities; (2) The robot can control the returned ball's rotational velocity. Therefore, the three models are the basis of controlling a spinning table tennis ball.

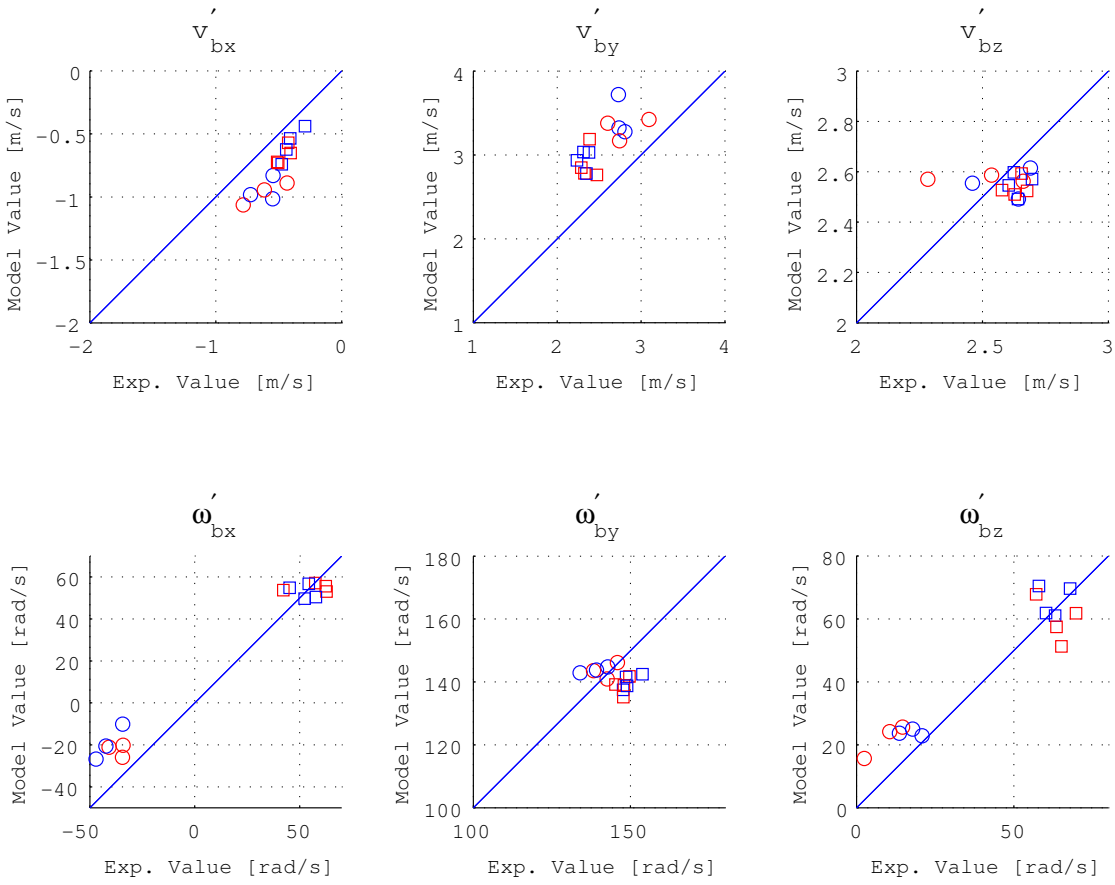


Figure 3.20: Verification of the side spin

Chapter 4

Racket Control Method

Until now, the target of most of the developed table tennis robots is to return the ball to *a desired position* on the opponent's table. If the ball's landing velocity or time can be controlled by the robot? In the table tennis ball game, it is a desired condition that the robot can control the ball's landing state which includes totally 9 variables: position, translational/rotational velocities and time. However, as shown in Figure 4.1, the input (racket's striking state, i.e. posture and velocity) only has 5 variables. Therefore, it is easy to see that all of the 9 variables of the output (the ball's state at the landing point) cannot be independently controlled by the racket's 5 striking variables. According to this observation, we have to consider the following two problems:

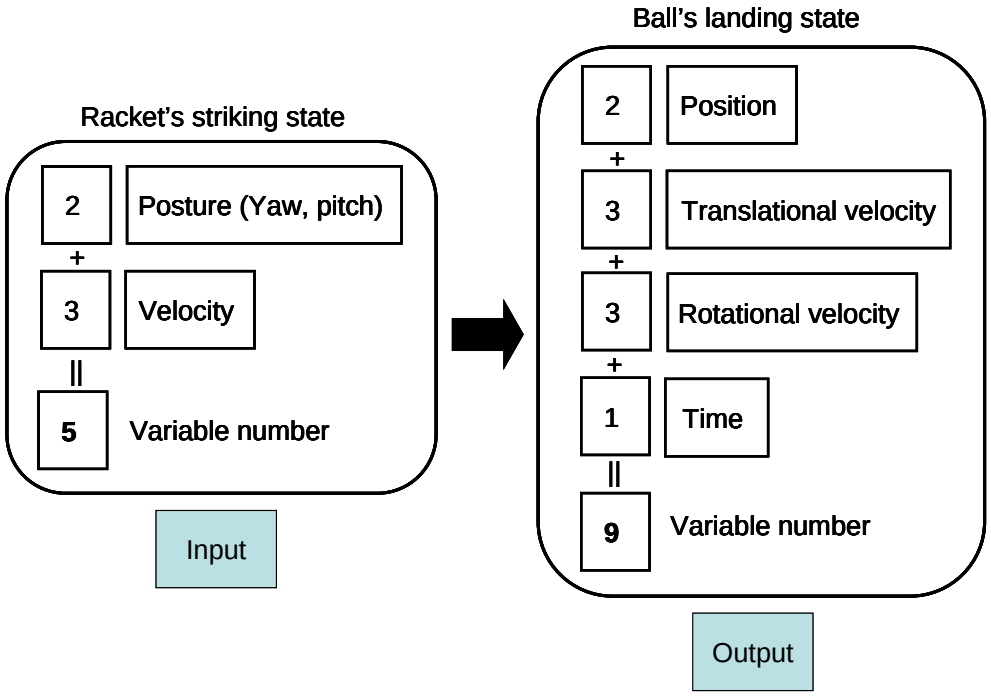


Figure 4.1: Racket control problem

1) Which variables of the ball's state at the landing point can be controlled by the racket's striking velocity and posture?

2) If a reference is given for the independently controlled variables shown in the first problem, how can the racket's velocity and posture be determined to achieve the reference?

In order to answer the above two problems, this chapter analyzes the properties of two physical models: the racket rebound model (RRM) and the aerodynamics model (ADM). In Section 4.2, by regarding RRM as a set of nonlinear equations with respect to the racket's state, an existence condition for real solutions of the set of nonlinear equations is shown and moreover the solutions are expressed in the closed form. On the other hand, associated with ADM, when considering the racket control, it is shown that a two-point boundary value problem plays important role. In Section 4.3, an on-line control method for the racket's state is proposed which can return the ball to a desired landing position with a desired rotational velocity at a desired landing time. Section 4.4 and 4.5 show numerical simulations and experimental results which verify the effectiveness of the proposed method [24, 25, 26].

4.1 Preliminary

The racket rebound model (RRM) and the aerodynamics model (ADM), which have been introduced in Chapter 3 will be applied in this chapter for solving the racket control problem. Therefore, in this section, we briefly review these two models for discussing their properties in the later section.

Racket Rebound Model (RRM)

The racket rebound model (RRM) expresses a relation between ball's velocities at the moment just before and after a racket strikes the ball. In Chapter 3, it is denoted that the racket strikes the ball with the racket's translational velocity ${}^B\boldsymbol{V} \in \mathbb{R}^3$ and the posture of the yaw angle $\alpha \in [-\frac{\pi}{2}, \frac{\pi}{2}]$ and the pitch angle $\beta \in [0, \pi]$. The ball's translational and rotational velocities just before and after the racket strikes the ball are defined as ${}^B\boldsymbol{v}_b \in \mathbb{R}^3$, ${}^B\boldsymbol{\omega}_b \in \mathbb{R}^3$ and ${}^B\boldsymbol{v}'_b \in \mathbb{R}^3$, ${}^B\boldsymbol{\omega}'_b \in \mathbb{R}^3$, respectively. In this section, since almost all the things are described in Σ_B, the simple notations $\boldsymbol{v}_0, \boldsymbol{v}_1, \boldsymbol{\omega}_0, \boldsymbol{\omega}_1$ and $\boldsymbol{V}$ are used instead of ${}^B\boldsymbol{v}_b, {}^B\boldsymbol{v}'_b, {}^B\boldsymbol{\omega}_b, {}^B\boldsymbol{\omega}'_b$ and ${}^B\boldsymbol{V}$. Then the relation between $(\boldsymbol{v}_0, \boldsymbol{\omega}_0)$ and $(\boldsymbol{v}_1, \boldsymbol{\omega}_1)$ can be shown as follows.

$$\begin{bmatrix} \boldsymbol{v}_1 - \boldsymbol{V} \\ \boldsymbol{\omega}_1 \end{bmatrix} = R_{RRM}(\alpha, \beta) \begin{bmatrix} \boldsymbol{v}_0 - \boldsymbol{V} \\ \boldsymbol{\omega}_0 \end{bmatrix}, \tag{4.1}$$

where

$$R_{RRM}(\alpha, \beta) = \begin{bmatrix} R_R & 0 \\ 0 & R_R \end{bmatrix} \begin{bmatrix} A_{vv} & A_{v\omega} \\ A_{\omega v} & A_{\omega\omega} \end{bmatrix} \begin{bmatrix} R_R & 0 \\ 0 & R_R \end{bmatrix}^T,$$

$$R_R = \begin{bmatrix} \cos\beta & \sin\beta\sin\alpha & \sin\beta\cos\alpha \\ 0 & \cos\alpha & -\sin\alpha \\ -\sin\beta & \cos\beta\sin\alpha & \cos\beta\cos\alpha \end{bmatrix},$$

$$A_{vv} = \mathrm{diag}(1 - k_v, 1 - k_v, -e_r), A_{v\omega} = k_v r S_{12}, \tag{4.2a}$$

$$A_{\omega v} = -k_\omega r S_{12}, A_{\omega\omega} = \mathrm{diag}(1 - k_\omega r^2, 1 - k_\omega r^2, 1), \tag{4.2b}$$

and

$$S_{12} = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

Note that R_R is a rotational matrix from Σ_B (the base coordinate) to Σ_R (the racket coordinate) with the yaw α and the pitch β. The parameter $r = 2 \times 10^{-2}$ m is the radius of the ball, and the parameters $k_v = 6.15 \times 10^{-1}$, $k_\omega = 2.57 \times 10^3$ and $e_r = 7.3 \times 10^{-1}$ are obtained by experimental data [36, 38].

Aerodynamics Model(ADM)

The aerodynamic model (ADM) is a differential equation which shows the motion of the ball flying in the air with the rotational velocity:

$$\ddot{\boldsymbol{p}}(t) = -\boldsymbol{g} - C_D(t)\frac{\rho}{m}S_b\|\dot{\boldsymbol{p}}(t)\|\dot{\boldsymbol{p}}(t) + C_M(t)\frac{\rho}{m}V_b\boldsymbol{\omega}\times\dot{\boldsymbol{p}}(t), \tag{4.3}$$

where for simplicity, in this chapter, $\boldsymbol{p}(t)\in\mathbb{R}^3$ is used instead of $\boldsymbol{p}_b(t)$ to express the position of the ball's center at the time t, $\boldsymbol{g}=[0,0,g]^T$, $S_b=\frac{1}{2}\pi r^2$, and $V_b=\frac{4}{3}\pi r^3$. Notice that the ball's rotational velocity $\boldsymbol{\omega}=[\omega_x,\omega_y,\omega_z]^T\in\mathbb{R}^3$ is assumed as a constant value.

Table 4.1: The parameters in the aerodynamics model

Parameter	Value
g	9.8 m/s^2
r	2 $\times 10^{-2}$m
m	2.7 $\times 10^{-3}$kg
ρ	1.184 kg/m^3(25°C)

The drag coefficient $C_D(t)$ and Magnus coefficient $C_M(t)$ vary with $(\dot{\boldsymbol{p}}(t),\boldsymbol{\omega})$, which are given by

$$C_D(t)=a_D+b_Dh(\dot{\boldsymbol{p}}(t),\boldsymbol{\omega}),\quad C_M(t)=a_M+b_Mh(\dot{\boldsymbol{p}}(t),\boldsymbol{\omega}) \tag{4.4}$$

where

$$h(\dot{\boldsymbol{p}}(t),\boldsymbol{\omega})=\frac{\dot{p}_x\omega_y-\dot{p}_y\omega_x}{\sqrt{(\dot{p}_x\omega_y-\dot{p}_y\omega_x)^2+(\dot{p}_x^2+\dot{p}_y^2)\omega_z^2}} \tag{4.5}$$

and $a_D=0.505, b_D=0.065, a_M=0.094$ and $b_M=-0.026$.

4.2 Properties of the Physical Models

4.2.1 Solution to the racket rebound model

Concerning RRM described by (4.1), we will consider the following problem: given $\boldsymbol{v}_0,\boldsymbol{\omega}_0,\boldsymbol{v}_1$ and $\boldsymbol{\omega}_1$, find $\boldsymbol{V},\alpha,\beta$ which satisfy (4.1).

Note that (4.1) is a set of 6 nonlinear equations with respect to $\boldsymbol{V},\alpha,\beta$. The upper three equations in (4.1) are

$$\boldsymbol{v}_1-\boldsymbol{V}=R_RA_{vv}R_R^T(\boldsymbol{v}_0-\boldsymbol{V})+R_RA_{v\omega}R_R^T\boldsymbol{\omega}_0. \tag{4.6}$$

In order to solve $\boldsymbol{V}$, the above equation is rewritten as

$$\begin{aligned}(\boldsymbol{I}-R_RA_{vv}R_R^T)\boldsymbol{V} &= \boldsymbol{v}_1-R_RA_{vv}R_R^T\boldsymbol{v}_0-R_RA_{v\omega}R_R^T\boldsymbol{\omega}_0\\ &= (\boldsymbol{v}_1-\boldsymbol{v}_0+\boldsymbol{v}_0)-R_RA_{vv}R_R^T\boldsymbol{v}_0-R_RA_{v\omega}R_R^T\boldsymbol{\omega}_0\\ &= (\boldsymbol{I}-R_RA_{vv}R_R^T)\boldsymbol{v}_0+(\boldsymbol{v}_1-\boldsymbol{v}_0)-R_RA_{v\omega}R_R^T\boldsymbol{\omega}_0.\end{aligned} \tag{4.7}$$

Consequently, $\boldsymbol{V}$ can be expressed as

$$\begin{aligned}\boldsymbol{V} &= \boldsymbol{v}_0+(\boldsymbol{I}-R_RA_{vv}R_R^T)^{-1}\left\{(\boldsymbol{v}_1-\boldsymbol{v}_0)-R_RA_{v\omega}R_R^T\boldsymbol{\omega}_0\right\}\\ &= \boldsymbol{v}_0+R_R\left(I-A_{vv}\right)^{-1}\left\{R_R^T(\boldsymbol{v}_1-\boldsymbol{v}_0)-A_{v\omega}R_R^T\boldsymbol{\omega}_0\right\}.\end{aligned} \tag{4.8}$$

On the other hand, the last three equations in (4.1) are

$$\boldsymbol{\omega}_1 = R_R A_{\omega v} R_R^T(\boldsymbol{v}_0 - \boldsymbol{V}) + R_R A_{\omega\omega} R_R^T \boldsymbol{\omega}_0. \tag{4.9}$$

Substituting (4.8) into (4.9), leads to

$$\begin{aligned} \boldsymbol{\omega}_1 &= -R_R A_{\omega v}(I - A_{vv})^{-1}\left\{R_R^T(\boldsymbol{v}_1 - \boldsymbol{v}_0) - A_{v\omega} R_R^T \boldsymbol{\omega}_0\right\} + R_R A_{\omega\omega} R_R^T \boldsymbol{\omega}_0 \\ &= -R_R A_{\omega v}(I - A_{vv})^{-1} R_R^T(\boldsymbol{v}_1 - \boldsymbol{v}_0) + R_R\left\{A_{\omega v}(I - A_{vv})^{-1} A_{v\omega} + A_{\omega\omega}\right\} R_R^T \omega_0. \end{aligned} \tag{4.10}$$

Note that by using (4.2a) and (4.2b),

$$S := -R_R A_{\omega v}(I - A_{vv})^{-1} R_R^T = \frac{k_\omega r}{k_v} S_s(\alpha, \beta), \tag{4.11}$$

where

$$S_s(\alpha, \beta) = \begin{bmatrix} 0 & \cos\alpha\cos\beta & \sin\alpha \\ -\cos\alpha\cos\beta & 0 & \cos\alpha\sin\beta \\ -\sin\alpha & -\cos\alpha\sin\beta & 0 \end{bmatrix} \tag{4.12}$$

is a skew symmetric matrix. Moreover, by using (4.2a), (4.2b), it is easy to see that

$$A_{\omega v}(I - A_{vv})^{-1} A_{v\omega} + A_{\omega\omega} = I_3. \tag{4.13}$$

Therefore, (4.10) is rewritten as

$$\boldsymbol{\omega}_1 = S(\boldsymbol{v}_1 - \boldsymbol{v}_0) + \boldsymbol{\omega}_0. \tag{4.14}$$

By introducing new variables $\boldsymbol{\xi}$ and $\boldsymbol{\eta}$ as $\boldsymbol{\xi} = \frac{k_\omega r}{k_v}(\boldsymbol{v}_1 - \boldsymbol{v}_0)$ and $\boldsymbol{\eta} = \boldsymbol{\omega}_1 - \boldsymbol{\omega}_0$, then (4.14) is equivalent to

$$\boldsymbol{\eta} = S_s(\alpha, \beta)\boldsymbol{\xi}. \tag{4.15}$$

From the above observation, it is easy to see that if RRM described by (4.1) has a solution of $(\boldsymbol{V}, \alpha, \beta)$, then it is necessary that $\boldsymbol{\xi}^T\boldsymbol{\eta} = \boldsymbol{\xi}^T S_s(\alpha, \beta)\boldsymbol{\xi} = 0$ since $S_s(\alpha, \beta)$ is a skew symmetric matrix, which is equivalent to

$$(\boldsymbol{v}_1 - \boldsymbol{v}_0)^T(\boldsymbol{\omega}_1 - \boldsymbol{\omega}_0) = 0. \tag{4.16}$$

Now we will solve (4.15) with respect to α and β under the condition that $\boldsymbol{\xi}^T\boldsymbol{\eta} = 0$ holds. By using the notations $\boldsymbol{\eta} = [\eta_x, \eta_y, \eta_z]^T$ and $\boldsymbol{\xi} = [\xi_x, \xi_y, \xi_z]^T$, (4.15) is rewritten as

$$\eta_x = \xi_y \cos\alpha\cos\beta + \xi_z \sin\alpha, \tag{4.17}$$
$$\eta_y = -\xi_x \cos\alpha\cos\beta + \xi_z \cos\alpha\sin\beta, \tag{4.18}$$
$$\eta_z = -\xi_x \sin\alpha - \xi_y \cos\alpha\sin\beta. \tag{4.19}$$

Firstly, for solving α, (4.17) and (4.19) are written as

$$\xi_y \cos\alpha\cos\beta = \eta_x - \xi_z \sin\alpha, \tag{4.20}$$
$$\xi_y \cos\alpha\sin\beta = -\eta_z - \xi_x \sin\alpha. \tag{4.21}$$

(4.20) and (4.21) lead to

$$\xi_y^2 \cos^2\alpha = (\eta_x - \xi_z \sin\alpha)^2 + (-\eta_z - \xi_x \sin\alpha)^2,$$

which is equivalent to the following quadratic equation with respect to $\sin\alpha$,

$$a\sin^2\alpha + 2b\sin\alpha + c = 0, \tag{4.22}$$

where

$$a = \xi_x^2 + \xi_y^2 + \xi_z^2, \quad b = \xi_x\eta_z - \xi_z\eta_x, c = \eta_x^2 + \eta_z^2 - \xi_y^2.$$

Now we will derive a necessary and sufficient condition for (4.22) to have real roots of $\sin\alpha$. Notice that the quadratic equation (4.22) has real roots if and only if $b^2 - ac \geq 0$.

$$\begin{aligned} b^2 - ac &= (\xi_x\eta_z - \xi_z\eta_x)^2 - (\xi_x^2 + \xi_y^2 + \xi_z^2)(\eta_x^2 + \eta_z^2 - \xi_y^2) \\ &= (\xi_x^2 + \xi_y^2 + \xi_z^2 - \eta_x^2 - \eta_z^2)\xi_y^2 - (\xi_x^2\eta_x^2 + \xi_z^2\eta_z^2 + 2\xi_x\xi_z\eta_x\eta_z). \end{aligned} \tag{4.23}$$

According to (4.16),

$$(\boldsymbol{v}_1 - \boldsymbol{v}_0)^T(\boldsymbol{\omega}_1 - \boldsymbol{\omega}_0) = \xi_x\eta_x + \xi_y\eta_y + \xi_z\eta_z = 0. \tag{4.24}$$

Hence, $-\xi_y\eta_y = \xi_x\eta_x + \xi_z\eta_z$, which leads to

$$\xi_y^2\eta_y^2 = (\xi_x\eta_x + \xi_z\eta_z)^2 = \xi_x^2\eta_x^2 + \xi_z^2\eta_z^2 + 2\xi_x\eta_x\xi_z\eta_z. \tag{4.25}$$

Substituting (4.25) into (4.23) yields

$$b^2 - ac = (\xi_x^2 + \xi_y^2 + \xi_z^2 - \eta_x^2 - \eta_y^2 - \eta_z^2)\xi_y^2. \tag{4.26}$$

Therefore, $b^2 - ac \geq 0$ can be shown to be equivalent to

$$\|\boldsymbol{\eta}\|^2 := \eta_x^2 + \eta_y^2 + \eta_z^2 \leq \xi_x^2 + \xi_y^2 + \xi_z^2 =: \|\boldsymbol{\xi}\|^2. \tag{4.27}$$

Thus α is given by

$$\sin\alpha = \frac{-b \pm \sqrt{b^2 - ac}}{a}. \tag{4.28}$$

Now we will consider a necessary and sufficient condition for (4.28) to have real roots of α. Notice here that the condition (4.27) implies

$$-1 \leq \frac{-b \pm \sqrt{b^2 - ac}}{a} \leq 1, \tag{4.29}$$

which means that (4.28) always have real roots of α.

In fact, it is easy to see that $a(\eta_x^2+\eta_y^2+\eta_z^2) \leq a^2$ under (4.27). Moreover, $a(\eta_x^2+\eta_y^2+\eta_z^2) \geq b^2$ since

$$\begin{aligned} a(\eta_x^2 + \eta_y^2 + \eta_z^2) - b^2 &= (\xi_x^2 + \xi_y^2 + \xi_z^2)(\eta_x^2 + \eta_y^2 + \eta_z^2) - (\xi_x\eta_z - \xi_z\eta_x)^2 \\ &= (\xi_x\eta_x + \xi_z\eta_z)^2 + (\xi_x\eta_y)^2 + (\xi_y\eta_x)^2 + (\xi_y\eta_y)^2 + (\xi_y\eta_z)^2 + (\xi_z\eta_y)^2 \\ &\geq 0. \end{aligned} \tag{4.30}$$

Therefore, $b^2 \leq a(\eta_x^2+\eta_y^2+\eta_z^2) \leq a^2$, which means $|b| \leq a$, i.e., both $a-b \geq 0$ and $a+b \geq 0$ hold. On the other hand, it is easy to see that $b^2-ac \leq (a+b)^2$ and $b^2-ac \leq (a-b)^2$ since

$$\begin{aligned}(a+b)^2-(b^2-ac) &= a(a+2b+c)\\ &= (\xi_x^2+\xi_y^2+\xi_z^2)(\xi_x^2+\xi_y^2+\xi_z^2+2\xi_x\eta_z-2\xi_z\eta_x+\eta_x^2+\eta_z^2-\xi_y^2)\\ &= (\xi_x^2+\xi_y^2+\xi_z^2)\left[(\xi_x+\eta_z)^2+(\xi_z-\eta_x)^2\right]\\ &\geq 0, \end{aligned} \tag{4.31}$$

$$\begin{aligned}(a-b)^2-(b^2-ac) &= a(a-2b+c)\\ &= (\xi_x^2+\xi_y^2+\xi_z^2)(\xi_x^2+\xi_y^2+\xi_z^2-2\xi_x\eta_z+2\xi_z\eta_x+\eta_x^2+\eta_z^2-\xi_y^2)\\ &= (\xi_x^2+\xi_y^2+\xi_z^2)\left[(\xi_x-\eta_z)^2+(\xi_z+\eta_x)^2\right]\\ &\geq 0, \end{aligned} \tag{4.32}$$

which by noting that $a \pm b \geq 0$, means $\sqrt{b^2-ac} \leq a \pm b$. Now it is straightforward to prove that $\sqrt{b^2-ac} \leq a+b$ implies $\frac{-b+\sqrt{b^2-ac}}{a} \leq 1$ and $\sqrt{b^2-ac} \leq a-b$ implies $-1 \leq \frac{-b-\sqrt{b^2-ac}}{a}$, which are equivalent to (4.29).

Once α is determined by (4.28), β is obtained by (4.18) as

$$\xi_z \sin\beta - \xi_x \cos\beta = \frac{\eta_y}{\cos\alpha}, \tag{4.33}$$

which has a real solution β if and only if

$$|\eta_y| \leq \sqrt{\xi_x^2+\xi_z^2}\cos\alpha,$$

which is equivalent to

$$\frac{\eta_y^2}{\xi_x^2+\xi_z^2} \leq 1-\left(\frac{-b\pm\sqrt{b^2-ac}}{a}\right)^2. \tag{4.34}$$

By summarizing the above observation, we get the next proposition.

Proposition 1: RRM described by (4.1) has a solution of $(\boldsymbol{V}, \alpha, \beta)$ if and only if

(1) $\boldsymbol{\xi}^T\boldsymbol{\eta} = 0$,

(2) $\|\boldsymbol{\eta}\| \leq \|\boldsymbol{\xi}\|$,

(3) $\frac{\eta_y^2}{\xi_x^2+\xi_z^2} \leq 1-\left(\frac{-b\pm\sqrt{b^2-ac}}{a}\right)^2$.

When these conditions are satisfied, $\boldsymbol{V}$, α, and β are given by (4.8), (4.28) and (4.33).

4.2.2 Two-point boundary value problem for the aerodynamics model

Here we will consider the two-point boundary value problem (TPBVP) of ADM described by (4.3), i.e., given $\boldsymbol{p}(t_1)$ and $\boldsymbol{p}(t_2)$ with $t_1 < t_2$, find $\boldsymbol{p}(t)$ and $\dot{\boldsymbol{p}}(t)$ for all $t \in [t_1, t_2]$ which satisfy (4.3).

It is well known that unlike the initial value problem, the TPBVP of an ordinary differential equation does not always have a (unique) solution. Let us consider the following TPBVP [8]:

$$\begin{cases} \ddot{\boldsymbol{p}}(t) = \boldsymbol{f}(t, \boldsymbol{p}, \dot{\boldsymbol{p}}), \\ \boldsymbol{p}(t_1) = \boldsymbol{p}_1 \in \mathbb{R}^n,\ \boldsymbol{p}(t_2) = \boldsymbol{p}_2 \in \mathbb{R}^n,\ t_1 < t_2 \in \mathbb{R}. \end{cases} \tag{4.35}$$

The next theorem is known as one of sufficient conditions for existence of a solution of the TPBVP.

Theorem 2 [8] : Suppose that there exist constants $K, L > 0$ such that $\boldsymbol{f}(t, \boldsymbol{p}, \dot{\boldsymbol{p}})$ satisfies

$$\|\boldsymbol{f}(t, \boldsymbol{p}_a, \dot{\boldsymbol{p}}_a) - \boldsymbol{f}(t, \boldsymbol{p}_b, \dot{\boldsymbol{p}}_b)\| \leq K\|\boldsymbol{p}_a - \boldsymbol{p}_b\| + L\|\dot{\boldsymbol{p}}_a - \dot{\boldsymbol{p}}_b\|, \tag{4.36}$$

for any $\boldsymbol{p}_a, \boldsymbol{p}_b \in \mathbb{R}^n$ and any $t \in \mathbb{R}$. Then if t_1 and t_2 satisfy

$$K\frac{(t_2 - t_1)^2}{8} + L\frac{t_2 - t_1}{2} < 1, \tag{4.37}$$

the TPBVP described by (4.35) has a unique solution.

Proof: Define $\boldsymbol{u} \in B$: the Banach space of $C[[t_1, t_2], \mathbb{R}^n]$ with the norm

$$\|\boldsymbol{u}\|_B \triangleq \max_{t_1 \leq t \leq t_2} \left[K\|\boldsymbol{u}(t)\|\right] + L\|\dot{\boldsymbol{u}}(t)\|]\,. \tag{4.38}$$

The mapping $T : B \to B$ is defined by

$$T\boldsymbol{u}(t) \triangleq \int_{t_1}^{t_2} G(t, s)\boldsymbol{f}\left(s, \boldsymbol{u}(s), \dot{\boldsymbol{u}}(s)\right) ds + \boldsymbol{w}(t), \tag{4.39}$$

where $G(t, s)$ is the Green's function for the boundary value problem

$$G(t, s) = \begin{cases} (t_2 - t)(s - t_1)/(t_1 - t_2), & t_1 \leq s \leq t \leq t_2, \\ (t_2 - s)(t - t_1)/(t_1 - t_2), & t_1 \leq t \leq s \leq t_2. \end{cases} \tag{4.40}$$

and

$$\boldsymbol{w}(t) = \frac{\boldsymbol{p}_2 - \boldsymbol{p}_1}{t_2 - t_1}t + \frac{\boldsymbol{p}_1 t_2 - \boldsymbol{p}_2 t_1}{t_2 - t_1}. \tag{4.41}$$

Here, we introduce three properties of $G(t, s)$ for later use,

(1) $|G(t, s)| \leq \frac{(t_2 - t_1)}{4}$,

(2) $\int_{t_1}^{t_2} |G(t, s)|ds = \frac{(t_2 - t)(t - t_1)}{2}$ and so $\int_{t_1}^{t_2} |G(t, s)|ds \leq \frac{(t_2 - t_1)^2}{8}$,

(3) $\int_{t_1}^{t_2} \left|\frac{dG(t,s)}{dt}\right| ds = \frac{(t_2 - t)^2 + (t - t_1)^2}{2(t_2 - t_1)}$ and so $\int_{t_1}^{t_2} \left|\frac{dG(t,s)}{dt}\right| ds \leq \frac{t_2 - t_1}{2}$.

For $\boldsymbol{u}_1, \boldsymbol{u}_2 \in B$,

$$\begin{aligned} \|T\boldsymbol{u}_2(t) - T\boldsymbol{u}_1(t)\| &= \left\| \int_{t_1}^{t_2} G(t, s) \left\{\boldsymbol{f}\left(s, \boldsymbol{u}_2(s), \dot{\boldsymbol{u}}_2(s)\right) - \boldsymbol{f}\left(s, \boldsymbol{u}_1(s), \dot{\boldsymbol{u}}_1(s)\right)\right\} ds \right\| \\ &\leq \int_{t_1}^{t_2} |G(t, s)| \left\|\boldsymbol{f}\left(s, \boldsymbol{u}_2(s), \dot{\boldsymbol{u}}_2(s)\right) - \boldsymbol{f}\left(s, \boldsymbol{u}_1(s), \dot{\boldsymbol{u}}_1(s)\right)\right\| ds \\ &\leq \int_{t_1}^{t_2} |G(t, s)| \left\{K\left\|\boldsymbol{u}_2(s) - \boldsymbol{u}_1(s)\right\| + L\left\|\dot{\boldsymbol{u}}_2(s) - \dot{\boldsymbol{u}}_1(s)\right\|\right\} ds \\ &\leq \left\|\boldsymbol{u}_2 - \boldsymbol{u}_1\right\|_B \int_{t_1}^{t_2} |G(t, s)|ds \\ &\leq \frac{(t_2 - t_1)^2}{8} \left\|\boldsymbol{u}_2 - \boldsymbol{u}_1\right\|_B. \end{aligned} \tag{4.42}$$

$$\begin{aligned}\left\|\frac{dT\boldsymbol{u}_2(t)}{dt}-\frac{dT\boldsymbol{u}_2(t)}{dt}\right\| &= \left\|\int_{t_1}^{t_2}\frac{dG(t,s)}{dt}\left\{\boldsymbol{f}\left(s,\boldsymbol{u}_2(s),\dot{\boldsymbol{u}}_2(s)\right)-\boldsymbol{f}\left(s,\boldsymbol{u}_1(s),\dot{\boldsymbol{u}}_1(s)\right)\right\}ds\right\| \\ &\leq \int_{t_1}^{t_2}\left|\frac{dG(t,s)}{dt}\right|\left\|\boldsymbol{f}\left(s,\boldsymbol{u}_2(s),\dot{\boldsymbol{u}}_2(s)\right)-\boldsymbol{f}\left(s,\boldsymbol{u}_1(s),\dot{\boldsymbol{u}}_1(s)\right)\right\|ds \\ &\leq \left\|\boldsymbol{u}_2-\boldsymbol{u}_1\right\|_B\int_{t_1}^{t_2}\left|\frac{dG(t,s)}{dt}\right|ds \\ &\leq \frac{t_2-t_1}{2}\left\|\boldsymbol{u}_2-\boldsymbol{u}_1\right\|_B. \end{aligned} \tag{4.43}$$

Therefore,

$$\left\|T\boldsymbol{u}_2-T\boldsymbol{u}_1\right\|_B\leq\left(K\frac{(t_2-t_1)^2}{8}+L\frac{t_2-t_1}{2}\right)\left\|\boldsymbol{u}_2-\boldsymbol{u}_1\right\|_B<\left\|\boldsymbol{u}_2-\boldsymbol{u}_1\right\|_B. \tag{4.44}$$

Thus, condition (4.37) implies that T is a contraction mapping and then it follows from the Banach Fixed-Point Theorem that T has a unique fixed point in $C[[t_1,t_2],\mathbb{R}^n]$, which is the solution of the boundary value problem (4.35) (Q.E.D).

In the case of ADM described by (4.3), note that

$$\boldsymbol{f}(t,\boldsymbol{p},\dot{\boldsymbol{p}})=-\boldsymbol{g}-C_D(t)\frac{\rho}{m}S_b\|\dot{\boldsymbol{p}}(t)\|\dot{\boldsymbol{p}}(t)+C_M(t)\frac{\rho}{m}V_b\boldsymbol{\omega}\times\dot{\boldsymbol{p}}(t),$$

where $C_D(t)$ and $C_M(t)$ are functions of $\dot{\boldsymbol{p}}$ and $\boldsymbol{\omega}$ as shown in (4.4) and (4.5). Therefore, $\boldsymbol{f}(t,\boldsymbol{p},\dot{\boldsymbol{p}})$ is a function of only $\dot{\boldsymbol{p}}$ and does not depend on $\boldsymbol{p}$, so $\boldsymbol{f}(t,\boldsymbol{p},\dot{\boldsymbol{p}})$ can be denoted by $\boldsymbol{f}(\dot{\boldsymbol{p}})$. By the mean-value theorem, there exists a $\dot{\boldsymbol{p}}$ such that

$$\boldsymbol{f}(\dot{\boldsymbol{p}}_a)-\boldsymbol{f}(\dot{\boldsymbol{p}}_b)=\frac{\partial\boldsymbol{f}(\dot{\boldsymbol{p}})}{\partial\dot{\boldsymbol{p}}}(\dot{\boldsymbol{p}}_a-\dot{\boldsymbol{p}}_b),$$

which implies that by supporting that $\|\dot{\boldsymbol{p}}(t)\|\leq 10$ m/s and $\|\boldsymbol{\omega}\|\leq 400$ rad/s, we can use Theorem 2 and it is easy to see that in (4.36), $K\to 0$ and

$$L=\max_{\dot{\boldsymbol{p}},\boldsymbol{\omega}}\left\|\frac{\partial\boldsymbol{f}(\dot{\boldsymbol{p}})}{\partial\dot{\boldsymbol{p}}}\right\|\approx 3.17,$$

where

$$\begin{aligned}\frac{\partial\boldsymbol{f}(\dot{\boldsymbol{p}})}{\partial\dot{\boldsymbol{p}}}=-\frac{\rho}{m}S_b\|\dot{\boldsymbol{p}}\|&\left\{b_D\dot{\boldsymbol{p}}\frac{\partial h}{\partial\dot{\boldsymbol{p}}}+C_D\left(I+\frac{\dot{\boldsymbol{p}}\dot{\boldsymbol{p}}^T}{\|\dot{\boldsymbol{p}}\|^2}\right)\right\} \\ &+\frac{\rho}{m}V_b\left\{b_M\boldsymbol{\omega}\times\dot{\boldsymbol{p}}\frac{\partial h}{\partial\dot{\boldsymbol{p}}}+C_M(\boldsymbol{\omega}\times)^T\right\}.\end{aligned}$$

Therefore it can be concluded that the TPBVP of ADM has a unique solution if $t_2-t_1<\frac{2}{L}\approx 0.63$ s.

4.3 Racket's Striking Posture and Velocity

Before discussing how to control the racket's striking posture and velocity, recall how the ball motion is generated by two physical models, RRM and ADM.

Let the time be $t = 0$ when the racket strikes the ball, and denote the time just before and after striking the ball by t_0 and t_1, respectively, i.e., $t_0 = 0_-$ and $t_1 = 0_+$.

Suppose the ball's state at the time t_0 is given as

$$\boldsymbol{p}(t_0) = \boldsymbol{p}_0,\ \dot{\boldsymbol{p}}(t_0) = \boldsymbol{v}_0,\ \boldsymbol{\omega}(t_0) = \boldsymbol{\omega}_0. \tag{4.45}$$

According to RRM described by (4.1), the racket's state $(\boldsymbol{V}, \alpha, \beta)$ determines

$$\boldsymbol{p}(t_1) = \boldsymbol{p}_1,\ \dot{\boldsymbol{p}}(t_1) = \boldsymbol{v}_1,\ \boldsymbol{\omega}(t_1) = \boldsymbol{\omega}_1, \tag{4.46}$$

where note that $\boldsymbol{p}_1 = \boldsymbol{p}_0$.

Then ADM described by (4.3), where the initial conditions $\boldsymbol{p}(t_1)$, $\dot{\boldsymbol{p}}(t_1)$ are given by (4.46), provides a ball motion $(\boldsymbol{p}(t), \dot{\boldsymbol{p}}(t))$ for $t \in [t_1, t_2]$ (as shown in Figure 4.2). Note that the time t_2 is a landing time when the ball lands in the opposite court, i.e.,

$$\boldsymbol{p}(t_2) = \boldsymbol{p}_2 = [p_{x2}, p_{y2}, 0]^T. \tag{4.47}$$

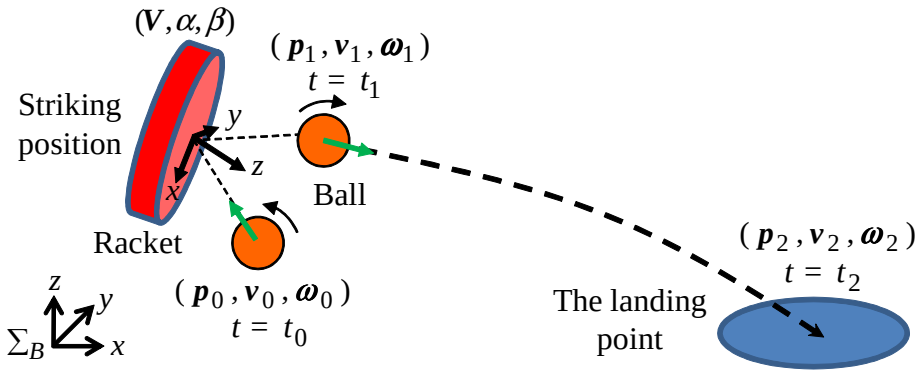


Figure 4.2: Rebound ball's motion

Now, in order for the table tennis playing robot to return a ball to a desired destination in the opposite court, we will propose how to control the racket's state $(\mathrm{V}, \alpha, \beta)$ in real-time process. Note that the ball's state just before striking, $\boldsymbol{p}_0, \boldsymbol{v}_0, \boldsymbol{\omega}_0$, are known. Suppose that the desired destination $\boldsymbol{p}_2$ and the landing time t_2 are requested. By using the proposed method of racket control, i.e., the racket's state $(\mathrm{V}, \alpha, \beta)$, it can be achieved not only that $\boldsymbol{p}(t_2)$ is equal to $\boldsymbol{p}_2$ but also that two elements of $\boldsymbol{\omega}(t) = \boldsymbol{\omega}_1 = \boldsymbol{\omega}_2$ are regulated as you like, say the y-element $\omega_y(t) = \omega_{y1} = \omega_{y2}$ and the z-element $\omega_z(t) = \omega_{z1} = \omega_{z2}$ are controlled at any desired values.

In fact, first of all, by considering (4.3) (ADM) with the boundary conditions $\boldsymbol{p}(t_1) = \boldsymbol{p}_1 = \boldsymbol{p}_0, \boldsymbol{p}(t_2) = \boldsymbol{p}_2$, we would use one of the well-known numerical methods of the TPBVP, e.g., the initial value methods, the finite difference methods, etc. [5] and also we can use a command "bvp4c" for the TPBVP in MATLAB. But those methods are not suitable for real-time process because they need a lot of computing time. Then we will propose a method of solving the TPBVP which gives an approximate solution in real-time process, i.e., with a reasonable computing time (< 30 ms). Instead of ADM described by (4.3), we propose to use a simple aerodynamics model (SAM)

$$\begin{cases} \ddot{p}_x(t) &= -D|\dot{p}_x(t)|\dot{p}_x(t), \\ \ddot{p}_y(t) &= -D|\dot{p}_y(t)|\dot{p}_y(t), \\ \ddot{p}_z(t) &= -g, \end{cases} \tag{4.48}$$

where $D = C_D \frac{\rho}{m} S_b$ is constant. Notice that (4.48) has a solution slightly different from the exact solutions of ADM described (4.3). But we can find out *analytical* solutions for SAM.

Let us solve $\dot{\boldsymbol{p}}_{x1}(t_1) = \boldsymbol{v}_1$ under the given conditions that $p_x(t_1) = p_{x1}, p_y(t_1) = p_{y1}, p_z(t_1) = p_{z1}$ and $p_x(t_2) = p_{x2}, p_y(t_2) = p_{y2}, p_z(t_2) = 0$ where $p_{x2} > p_{x1}$ and $p_{z1} > 0$ are assumed.

In x axis, according to the base coordinate, $\ddot{p}_x(t) = -D\dot{p}_x^2(t)$. Its integral equations (4.50), (4.51) can be obtained readily with the method of derivation of parameters as follows:

$$\int \frac{\ddot{p}_x(t)}{\dot{p}_x^2(t)} dt = -\int D dt. \tag{4.49}$$

$$\dot{p}_x = \frac{1}{Dt + C_{x1}}, \tag{4.50}$$

$$p_x = \frac{1}{D} \ln |Dt + C_{x1}| + C_{x2}, \tag{4.51}$$

where C_{x1} and C_{x2} are constant values and C_{x1} can be easily obtained by substituting $\dot{p}_x(t_1) = v_{x1}$ into (4.50),

$$C_{x1} = \frac{1}{v_{x1}} - Dt_1. \tag{4.52}$$

Then, by using (4.52), $p_x(t_1) = p_{x1}$ and $p_x(t_2) = p_{x2}$, it can be seen that

$$p_{x2} - p_{x1} = \frac{1}{D} \ln(Dv_{x1}(t_2 - t_1) + 1). \tag{4.53}$$

In y axis, actually, $\ddot{p}_y(t) = -\mathrm{sgn}(v_{y1}) D\dot{p}_y^2(t)$. With the same method as x axis, it yields

$$p_{y2} - p_{y1} = \mathrm{sgn}(v_{y1}) \frac{1}{D} \ln(D|v_{y1}|(t_2 - t_1) + 1). \tag{4.54}$$

In z axis, the integral solutions are

$$\dot{p}_z = \int \ddot{p}_z dt = -gt + C_{z1} \tag{4.55}$$

$$p_z = -g\frac{t^2}{2} + C_{z1}t + C_{z2}. \tag{4.56}$$

Substituting the boundary conditions, $\dot{p}_z(t_1) = v_{z1}, p_z(t_1) = p_{z1}$ and $p_z(t_2) = 0$ into (4.55) and (4.56) lead to

$$-p_{z1} = -\frac{1}{2} g(t_2 - t_1)^2 + v_{z1}(t_2 - t_1). \tag{4.57}$$

Therefore,

$$\begin{aligned} v_{x1} &= \tfrac{\exp\{D(p_{x2}-p_{x1})\}-1}{D(t_2-t_1)}, \\ v_{y1} &= \mathrm{sgn}(p_{y2} - p_{y1}) \tfrac{\exp\{D|p_{y2}-p_{y1}|\}-1}{D(t_2-t_1)}, \\ v_{z1} &= \tfrac{1}{2} g(t_2 - t_1) - \tfrac{p_{z1}}{t_2-t_1}, \end{aligned} \tag{4.58}$$

where $\mathrm{sgn}(p_{y2} - p_{y1}) = \mathrm{sgn} v_{y1}$ is used.

Second, $\boldsymbol{v}_0, \boldsymbol{v}_1, \boldsymbol{\omega}_0$ and the desired ω_{y1}, ω_{z1} determine ω_{x1} as shown in the condition (4.16). Finally, we can obtain the racket's state $(\mathrm{V}, \alpha, \beta)$ as shown in Subsec. 4.2.1. Figure 4.3 shows the flow chart of the solving process in which the parameters in the green square, blue square and red square are the given, intermediate solved and final solved parameters, respectively. All those processes take the computing time less than 30 ms.

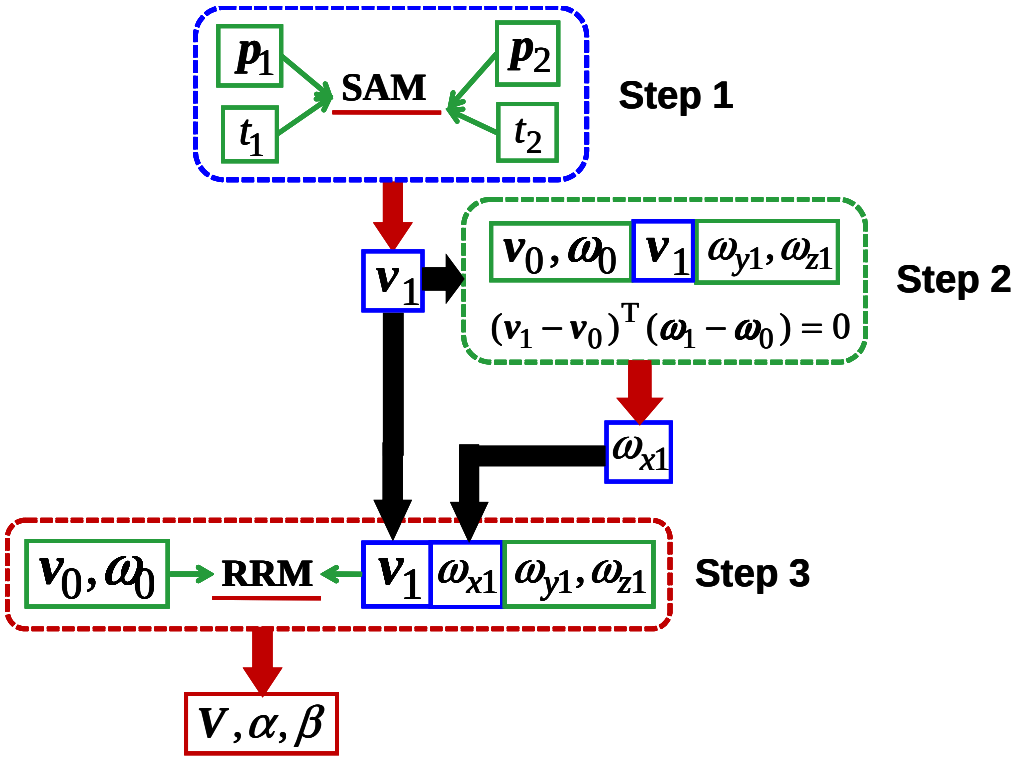


Figure 4.3: The process of solving the racket's striking posture and velocity

4.4 Numerical Simulations

In this section, it will be verified via numerical simulations how well the racket's state $(\boldsymbol{V}, \alpha, \beta)$ obtained by the proposed method can work. The ball's state $(\boldsymbol{p}_0, \boldsymbol{v}_0, \boldsymbol{\omega}_0)$ just before the racket strikes the ball is set in three cases; $\boldsymbol{p}_0 = [-0.15, 0.70, 0.25]$ m is common for the three cases, and $\boldsymbol{v}_0$, $\boldsymbol{\omega}_0$ are shown in Table 4.2(a). Furthermore, the desired $(p_{x2}, p_{y2}, \omega_{y1}, \omega_{z1}, t_2)$ are also shown in Table 4.2(b). From the base coordinate system Σ_B (see Figure 4.2), it is easy to see that the comming ball and the returning ball of the case 1) are in backspin, ones of the case 2) are in topspin, and ones of the case 3) are in the side spin, respectively. Note that the desired p_{x2}, p_{y2} m and t_2 s are set in common for the three cases.

Associated with the three cases shown in Table 4.2, the proposed method provides the racket's state as shown in Table 4.3; "the proposed" means the racket's state which is obtained by the proposed method, i.e., using SAM, and "the true" means the racket's state which is obtained by the command "bvp4c" in MATLAB and achieves the desired $(p_{x2}, p_{y2}, \omega_{y1}, \omega_{z1}, t_2)$ exactly. Table 4.4 shows results of $p_{x2}, p_{y2}, \omega_{y1}, \omega_{z1}, t_2$ when the racket's states via the proposed method are used. Comparing Table 4.4 with Table 4.2(b), the average errors on p_{x2}, p_{y2} and t_2 are 0.082 m, 0.045 m and 0.025 s, respectively. In addition, it can be seen that ω_{y1} and ω_{z1} can be achieved exactly. Those results are due to that RRM described (4.1) is exactly solved, but ADM described (4.3) is approximated as SAM described by (4.48) to be solved in real-time process.

Moreover, the returning ball's trajectory and time-history are shown in Figures 4.4-4.6 for the cases 1)-3), respectively. In each figure, the red line shows the result obtained by the proposed method, and the green dotted line shows the result which achieves the desired destination exactly.

Table 4.2: Ball's state and desired destination

(a) Ball's state just before struck by racket

	$[v_{x0}, v_{y0}, v_{z0}]$ [m/s]	$[\omega_{x0}, \omega_{y0}, \omega_{z0}]$ [rad/s]
1)	[-2.5, 0, 0.1]	[0, 150, 0]
2)	[-4.5, 0, 0.3]	[0, -150, 0]
3)	[-2.75, -0.8, -0.5]	[0, -50, 150]

(b) The desired $(p_{x2}, p_{y2}, \omega_{y1}, \omega_{z1}, t_2)$

	$[p_{x2}, p_{y2}]$ [m]	$[\omega_{y1}, \omega_{z1}]$ [rad/s]	t_2 [s]
1)	[2.055, 0.768]	[-100, 0]	0.6
2)	[2.055, 0.768]	[100, 0]	0.6
3)	[2.055, 0.768]	[0, -100]	0.6

Table 4.3: Racket' state via the proposed method and the true state

	$\boldsymbol{V}$ [m/s]	α [rad]	β [rad]
1) the proposed	[1.48, 0.095, 1.41]	-0.011	0.792
the true	[1.51, 0.11, 1.34]	-0.013	0.811
2) the proposed	[0.65, 0.064, 1.72]	-0.012	1.66
the true	[0.67, 0.075, 1.87]	-0.014	1.63
3) the proposed	[1.48, 0.23, 1.30]	0.31	1.25
the true	[1.50, 0.38, 1.33]	0.28	1.24

Table 4.4: The obtained $(p_{x2}, p_{y2}, \omega_{y1}, \omega_{z1}, t_2)$

	$[p_{x2}, p_{y2}]$ [m]	$[\omega_{y1}, \omega_{z1}]$ [rad/s]	t_2 [s]
1)	[2.09, 0.76]	[-100, 0]	0.620
2)	[1.92, 0.78]	[100, 0]	0.561
3)	[1.98, 0.66]	[0, -100]	0.582

4.5 Experimental Results

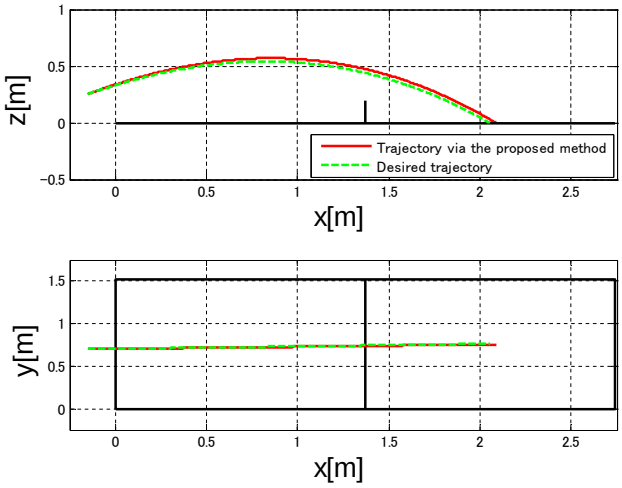
4.5.1 Experimental settings

Prediction of Ball's Motion Trajectory Before Struck by the Racket

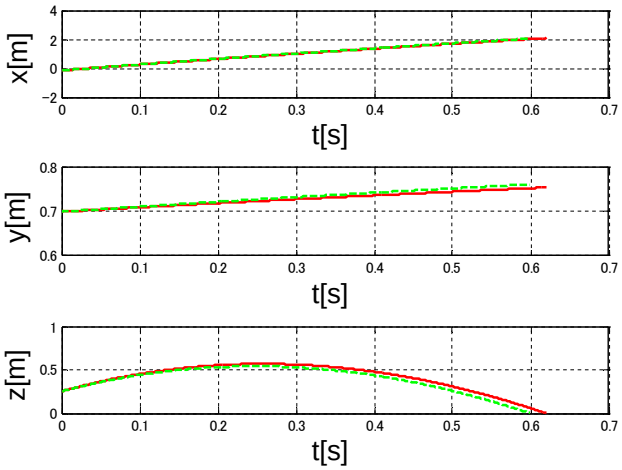
In the experiment, a table tennis ball is shot from a ball catapult machine. The ball's position, translational velocity and rotational velocity just after being shot by the catapult machine are measured by using two high speed monochrome cameras (900 Hz) [23]. Then, using these measured ball's information as the initial state $(\boldsymbol{p}_{init}, \boldsymbol{v}_{init}, \boldsymbol{\omega}_{init})$, the motion trajectory of the ball is predicted based on the aerodynamics model (ADM) and the table rebound model (TRM) (Chapter 3) to determine the striking position $\boldsymbol{p}_0$ and the striking time t_0 as well as the ball's state $(\boldsymbol{v}_0, \boldsymbol{\omega}_0)$ just before the striking.

The following shows the prediction process in detail:

Firstly, just after obtaining the ball's initial position and velocities measured with the high speed cameras, the motion trajectory of the ball is predicted by using an initial value method,



(a) Ball trajectory



(b) Ball time-history

Figure 4.4: The case 1) of backspin

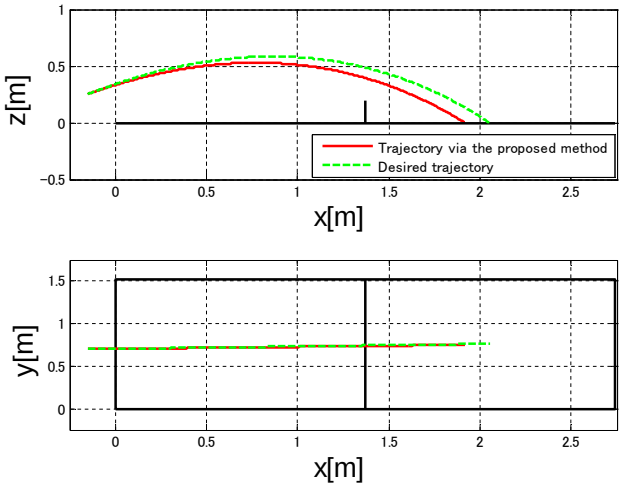
the fourth-order Runge-Kutta method, based on the differential equation of ADM, which includes the position prediction and velocity prediction:

Solve $\boldsymbol{y}$, $\dot{\boldsymbol{y}} = f(t, \boldsymbol{y})$ with initial condition $\boldsymbol{y}(t_0)$.

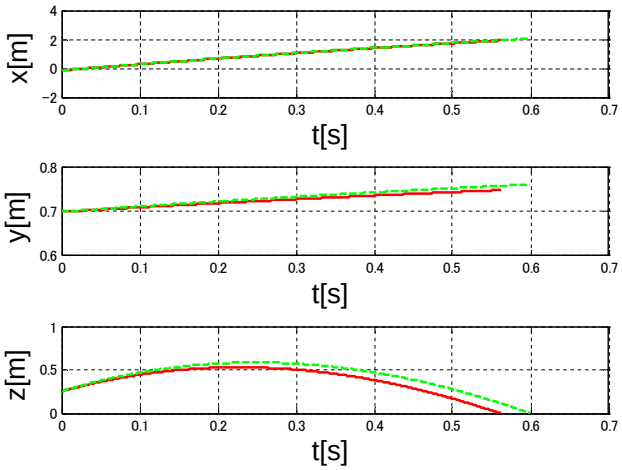
$$t_{i+1} = t_i + h$$

$$\begin{cases} k_1 = hf(t_i, \boldsymbol{y}_i) \\ k_2 = hf(t_i + h/2, \boldsymbol{y}_i + k_1/2) \\ k_3 = hf(t_i + h/2, \boldsymbol{y}_i + k_2/2) \\ k_4 = hf(t_i + h, \boldsymbol{y}_i + k_3) \end{cases}$$

$$\boldsymbol{y}_{i+1} = \boldsymbol{y}_i + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) \tag{4.59}$$



(a) Ball trajectory



(b) Ball time-history

Figure 4.5: The case 2) of topspin

Position Prediction:

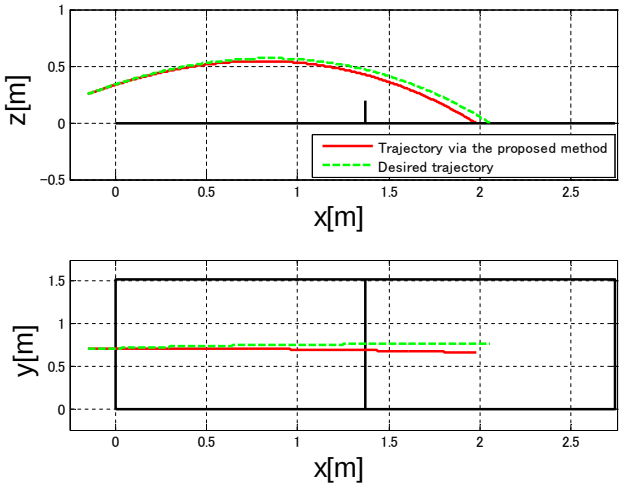
$$\begin{cases} \boldsymbol{y} = \boldsymbol{p} \\ \dot{\boldsymbol{y}} = f(t, \boldsymbol{y}) = \boldsymbol{v} \\ \boldsymbol{y}(t_0) = \boldsymbol{p}_{init} \end{cases} \tag{4.60}$$

Velocity Prediction:

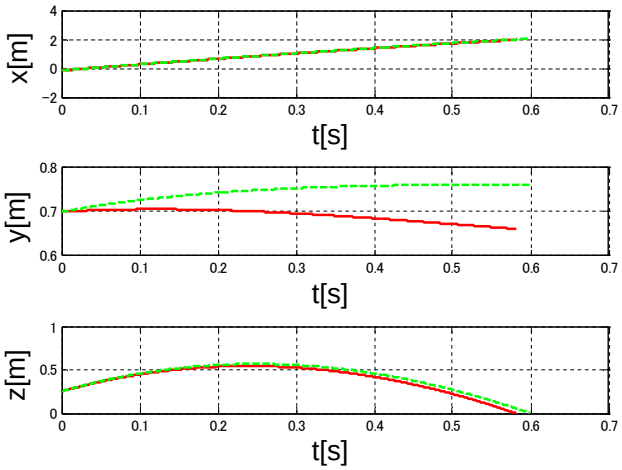
$$\begin{cases} \boldsymbol{y} = \boldsymbol{v} \\ \dot{\boldsymbol{y}} = f(t, \boldsymbol{y}) = -\boldsymbol{g} - C_D \frac{\rho}{m} S_b \|\boldsymbol{v}\| \boldsymbol{v} + C_M \frac{\rho}{m} V_b \boldsymbol{\omega} \times \boldsymbol{v} \\ \boldsymbol{y}(t_0) = \boldsymbol{v}_{init}, \boldsymbol{\omega}_{init} \end{cases} \tag{4.61}$$

where h is the iteration step size which is set as $\frac{1}{1200}$ in the experiment. Notice that the ball's rotational velocity $\boldsymbol{\omega}$ is considered as a constant value when the ball flies in the air.

Secondly, when $p_z \leq 0$ (the ball's position in z-axis) at the first time, it is confirmed that the ball collides with the table and the ball's state (position and velocities) at this time is seen



(a) Ball trajectory



(b) Ball time-history

Figure 4.6: The case 3) of sidespin

as the state just before rebounding from the table. Then, by using TRM, the ball's state just after the rebounding can be estimated.

Thirdly, setting the ball's state just after rebounding from the table as the initial value, predict the ball's motion trajectory after rebounding with the same method as in the first step. And the prediction stops if one of the following conditions is satisfied:

1) if $-0.3[m] \leq p_x \leq 0$ and $v_z \leq 0$,

2) if $p_x \leq -0.3[m]$.

Then, the stop time of the prediction is decided as the striking time. The ball's position and velocities at that time are determined as the ball's state just before striking with the racket.

Determination of 7 Joints' Postures of the Manipulator at the Striking Time

The robot is a robot manipulator with 7 DOF (PA10, Mitsubishi H.I.) and the racket is attached on the robot's end effector. The method of controlling the posture and velocity of the racket at the striking position is shown in Sec.4.3. Now we will show how to control the robot's joints' angles at the striking position.

Let $\boldsymbol{q} \in \mathbb{R}^7$ be the vector of joint angles with its *i*-th element q_i being the *i*-th joint angle. Then the kinematic equation is given as $[\boldsymbol{p}^T, \boldsymbol{\theta}^T]^T = \boldsymbol{F}(\boldsymbol{q})$ where $\boldsymbol{p} \in \mathbb{R}^3$ is a position of the racket's center and $\boldsymbol{\theta} = [\alpha, \beta, \gamma]^T \in \mathbb{R}^3$ is the racket's posture. Recall that the references of $\boldsymbol{p}$, α and β are given as shown in Sec.4.3, but γ is free. Therefore the kinematic equation is reduced to

$$\begin{bmatrix} \boldsymbol{p} \\ \boldsymbol{\theta}_{\bar{2}} \end{bmatrix} = \boldsymbol{F}_{\bar{5}}(\boldsymbol{q}), \tag{4.62}$$

where $\boldsymbol{\theta}_{\bar{2}} = [\alpha, \beta]^T \in \mathbb{R}^2$ and $\boldsymbol{F}_{\bar{5}}(\boldsymbol{q}) = [I_5\ 0_{5\times 1}]\boldsymbol{F}(\boldsymbol{q})$. Notice that given $[\boldsymbol{p}^T, \boldsymbol{\theta}_{\bar{2}}^T] \in \mathbb{R}^5$, the equation (4.62) is a set on nonlinear equations to $\boldsymbol{q} \in \mathbb{R}^7$ and so it could have infinite number of solutions if there exists a solution $\boldsymbol{q}$. Here, in order to obtain a solution $\boldsymbol{q}$ of (4.62), the Newton-Raphson method is used, i.e.,

$$\boldsymbol{q}^{(k+1)} = \boldsymbol{q}^{(k)} + \boldsymbol{J}_{\bar{5}}^{\dagger}(\boldsymbol{q}^{(k)}) \left([\boldsymbol{p}^T, \boldsymbol{\theta}_{\bar{2}}^T]^T - \boldsymbol{F}_{\bar{5}}(\boldsymbol{q}^{(k)}) \right), \tag{4.63}$$

where $J_{\bar{5}}^{\dagger}$ is a pseudo inverse of the Jacobian $J_{\bar{5}}$, i.e., $J_{\bar{5}} = \frac{\partial \boldsymbol{F}_{\bar{5}}}{\partial \boldsymbol{q}} \in \mathbb{R}^{5\times 7}$ and $J_{\bar{5}}^{\dagger} = J_{\bar{5}}^T \left(J_{\bar{5}} J_{\bar{5}}^T \right)^{-1} \in \mathbb{R}^{7\times 5}$. The iterative process (4.63) starts with $\boldsymbol{q}^{(0)}$ which is a waiting position for the comming ball and the iterative process stops when the stop condition is satisfied, i.e., $\| [\boldsymbol{p}^T, \boldsymbol{\theta}_{\bar{2}}^T]^T - \boldsymbol{F}_{\bar{5}}(\boldsymbol{q}^{(k)}) \| \leq e$ where $e > 0$ is a preassigned small number.

Now we will show how to achieve the racket's velocity $\boldsymbol{V}$.

Again the kinematic equation $[\boldsymbol{p}^T, \boldsymbol{\theta}^T]^T = \boldsymbol{F}(\boldsymbol{q})$ is reduced as $\boldsymbol{p} = \boldsymbol{F}_{\bar{3}}(\boldsymbol{q})$ where $\boldsymbol{F}_{\bar{3}}(\boldsymbol{q}) = [I_3\ 0_{3\times 3}]\boldsymbol{F}(\boldsymbol{q})$. Therefore it is derived directly that $\boldsymbol{V} = \boldsymbol{J}_{\bar{3}}(\boldsymbol{q})\dot{\boldsymbol{q}}$ where $\boldsymbol{J}_{\bar{3}} = \frac{\partial \boldsymbol{F}_{\bar{3}}}{\partial \boldsymbol{q}} \in \mathbb{R}^{3\times 7}$. Then $\dot{q}_7 = 0$ is always selected for simplicity, so by introducing $\dot{\boldsymbol{q}}_{\bar{6}} = [I_6\ 0_{6\times 1}]\dot{\boldsymbol{q}} \in \mathbb{R}^6$, the above equation is reduced as

$$\boldsymbol{V} = \bar{\boldsymbol{J}}_{\bar{3}}(\boldsymbol{q})\dot{\boldsymbol{q}}_{\bar{6}}, \tag{4.64}$$

where $\bar{\boldsymbol{J}}_{\bar{3}} = \boldsymbol{J}_{\bar{3}}[I_6\ 0_{6\times 1}]^T \in \mathbb{R}^{3\times 6}$. Therefore, $\dot{\boldsymbol{q}}_{\bar{6}}$ can be determined by

$$\dot{\boldsymbol{q}}_{\bar{6}} = \bar{\boldsymbol{J}}_{\bar{3}}^{\dagger}(\boldsymbol{q})\boldsymbol{V} \tag{4.65}$$

in which $\bar{\boldsymbol{J}}_{\bar{3}}^{\dagger} = (\bar{\boldsymbol{J}}_{\bar{3}}\boldsymbol{W}^{-1})^{\mathrm{T}}(\bar{\boldsymbol{J}}_{\bar{3}}\boldsymbol{W}^{-1}\bar{\boldsymbol{J}}_{\bar{3}}^T)^{-1}$. $\boldsymbol{W}^{-1}$ = diag $\{1,\ 0.6,\ 2, 2,\ 2\pi, 2\pi\}$ is a weight matrix with respect to the joints' limit velocity.

4.5.2 Some results

Here some experimental results are shown in the cases that the catapult machine shot a backspin ball and a topspin ball.

The striking position $\boldsymbol{p}_0 = \boldsymbol{p}_1$ and the ball's velocities $\boldsymbol{v}_0, \boldsymbol{\omega}_0$ were estimated real-timely by using the high speed cameras' data as in Table 4.5(a). The desired destination $\boldsymbol{p}_2$ in the opposite court, the landing time t_2 and the desired rotational velocities ω_{y1}, ω_{z1} are set as in Table 4.5(b). Then the racket's state obtained by the proposed method is shown in Table 4.5(c). In order to evaluate how well the hitting position is estimated and how well the ball is controlled by the racket to return to the desired destination, two middle speed colored cameras (Library Inc.) are used, whose frame rate is 150 Hz.

Table 4.5: Experimental results

(a) Ball's state just before struck by racket

	back spin	top spin
$[p_{x0}, p_{y0}, p_{z0}]$[m]	[-0.30, 0.66, 0.23]	[-0.14, 0.73, 0.26]
$[v_{x0}, v_{y0}, v_{z0}]$ [m/s]	[-2.97, -0.0077, 0.34]	[-4.17, -0.12, -0.0023]
$[\omega_{x0}, \omega_{y0}, \omega_{z0}]$ [rad/s]	[-28.02, 229.76, -28.58]	[7.53, -241.09, -6.28]

(b) The desired (p_{x2}, p_{y2}, ω_{y1}, ω_{z1}, t_2)

	$[p_{x2}, p_{y2}]$ [m]	$[\omega_{y1}, \omega_{z1}]$ [rad/s]	t_2 [s]
back spin	[2.055, 0.768]	[-125.66, 0]	0.6
top spin	[2.055, 0.768]	[104.72, 0]	0.5

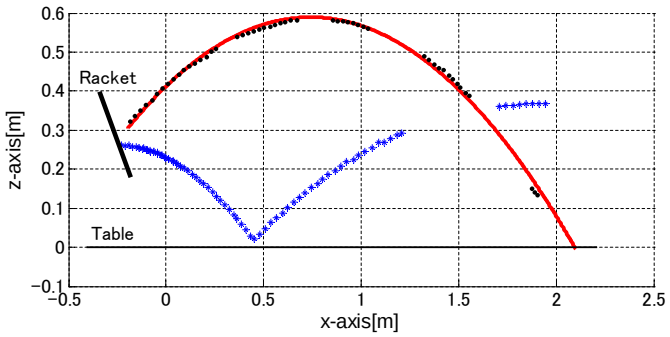
(c) Racket' state via the proposed method

	$\boldsymbol{V}$ [m/s]	α [rad]	β [rad]
back spin	[1.26, -0.18, 1.75]	-0.068	0.708
top spin	[1.15, 0.019, 0.64]	-0.028	1.814

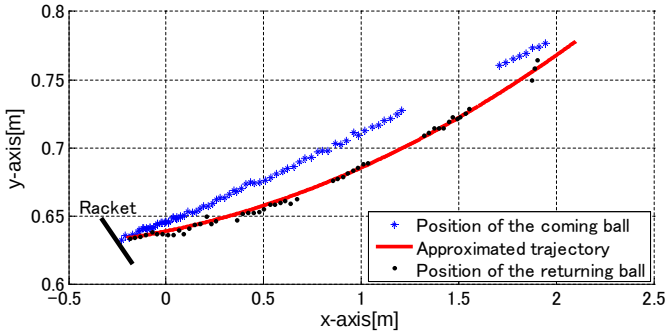
Figure 4.7 shows the ball's trajectories of x-z plane and x-y plane in the case of back spin ball. The blue * shows the comming ball's trajectory sensed by the middle speed camera system. The black dots show the returning ball's trajectory sensed by the middle speed camera system, and the red line is a smooth curve approximating the black dots. Figure 4.9 shows the time-history of the returning ball's position, where the black dots are data sensed by the middle speed camera system and the red curve is a smooth curve derived from the black dots. The horizontal solid line is the desired position. From Figures 4.7 and 4.9, it can be evaluated that the achieved destination $\boldsymbol{p}_2$ in the opposite court is $[p_{x2}, p_{y2}] = [2.10, 0.78]$ m and the landing time t_2 is $t_2 = 0.622$ s. In the case of top spin ball, Figure 4.8 shows the ball's trajectories of x-z plane and x-y plane and Figure 4.10 shows the time-history of the ball where the blue *, the black dots, the red curve and the horizontal solid line express the same things as in Figures 4.7 and 4.9. From those figures, in the case of top spin ball, it can be evaluated that the achieved destination $\boldsymbol{p}_2$ in the opposite court is $[p_{x2}, p_{y2}] = [1.90, 0.744]$ m and the landing time is $t_2 = 0.451$ s.

The experimental results above have almost same features as observed in the numerical simulations, i.e., the achieved p_{x2} and t_2 is larger than the desired ones in the case of back spin ball, while the achieved p_{x2} and t_2 are smaller than the desired ones in the case of top spin ball.

Associated with the requested racket's state ($\boldsymbol{V}, \alpha, \beta$), the requested joint angle and velocity of the robot manipulator at the striking instant, $\boldsymbol{q}_{ref}$ and $\dot{\boldsymbol{q}}_{ref}$, are determined as shown in (4.63) and (4.65). It is not a concern in this thesis how to control the manipulator to achieve $\boldsymbol{q}(0) = \boldsymbol{q}_{ref}$ and $\dot{\boldsymbol{q}}(0) = \dot{\boldsymbol{q}}_{ref}$, but it must be shown how near the attained $\boldsymbol{q}(0)$ and $\dot{\boldsymbol{q}}(0)$ are to the requested $\boldsymbol{q}_{ref}$ and $\dot{\boldsymbol{q}}_{ref}$. Figures 4.11 and 4.12 show the time-histories of the angles and velocities of the joints #1$\sim$ #7 in the case of back spin ball, where the blue lines show $\boldsymbol{q}_{ref}$ and $\dot{\boldsymbol{q}}_{ref}$ and the red dots show $\boldsymbol{q}(t)$ and $\dot{\boldsymbol{q}}(t)$ which is calculated by $\frac{\boldsymbol{q}(t+h)-\boldsymbol{q}(t)}{h}$ with a sampling period h. The vertical green line shows the striking instant, i.e., $t = 0$. Figures 4.13 and 4.14 also show the time-histories of the angles and velocities of the joints #1$\sim$ #7 in the case of top spin ball. From both figures, it can be seen that $\boldsymbol{q}(0)$ and $\dot{\boldsymbol{q}}(0)$ have been controlled well.

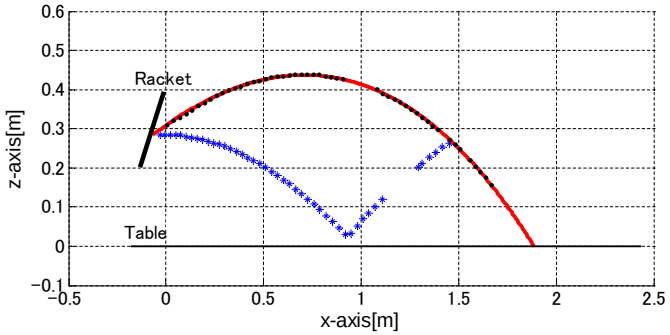


(a) x-z plane

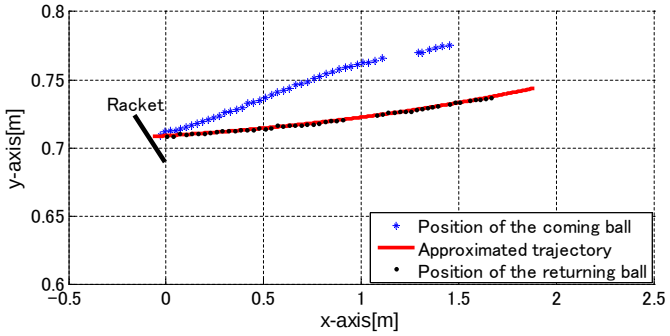


(b) x-y plane

Figure 4.7: Ball trajectory (Experiment): the case of backspin ball



(a) x-z plane



(b) x-y plane

Figure 4.8: Ball trajectory (Experiment): the case of topspin ball

4.6 Summary

In this chapter, firstly, the racket control planning is discussed rigorously from a theoretical viewpoint, i.e., concerning a table tennis robot, a two-point boundary value problem is

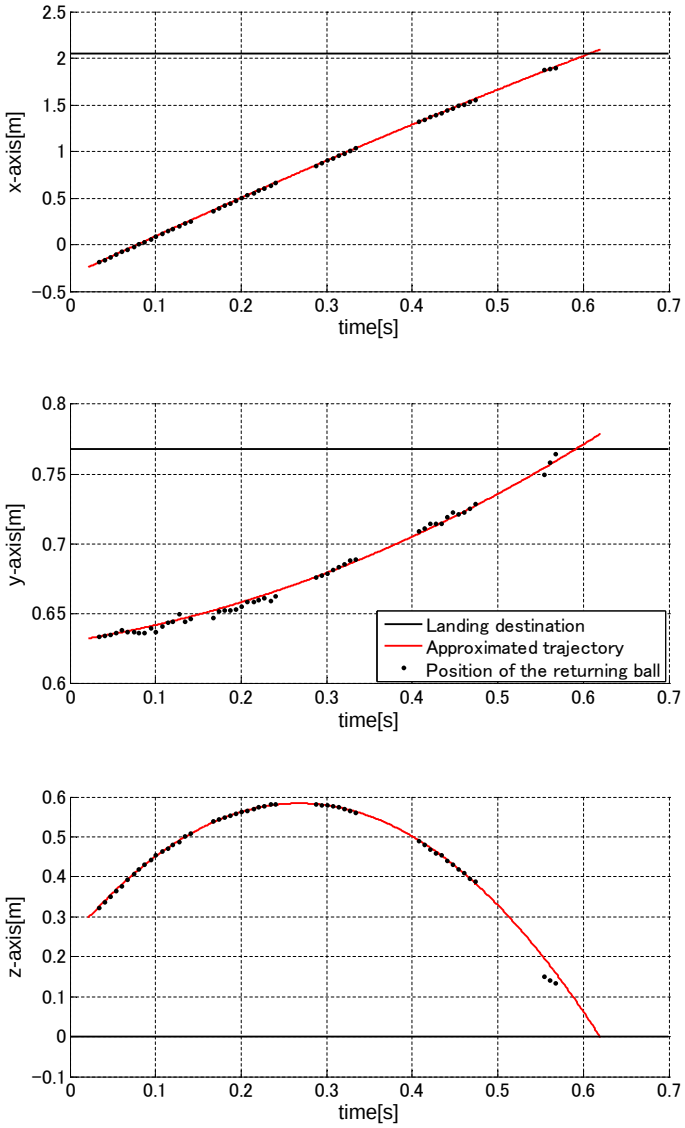


Figure 4.9: Ball time-history (Experiment): the case of backspin ball

discussed more rigorously. Then, an on-line racket control method is proposed for returning a table tennis ball to a desired position with a desired rotational velocity at a desired landing time. In detail, the algorithm of determining the racket's state (velocity and posture) is derived by performing the following two works: 1) solving nonlinear equations of the racket rebound model (RRM); 2) solving the two-point boundary value problem of the differential equation of the aerodynamics model (ADM). However, in fact, solving the work 2 needs a lot of computing time since the aerodynamics model is a complicated differential equation which can be solved by using numerical methods. Therefore, a simple aerodynamics model (SAM) which can directly obtain the analytical solution is proposed for using in the on-line situation.

Some numerical simulations and experimental results have been demonstrated to verify the effectiveness of the proposed method. The numerical simulations show that the average errors of the landing position in x-axis and y-axis (the length and width directions of the table) are 0.082 m and 0.045 m, respectively. And the average error of the landing time is 0.025 s. The desired rotational velocity can be achieved exactly. Those results are due to that RRM is exactly solved, but ADM is approximated as SAM to be solved in real-time process. The experiments

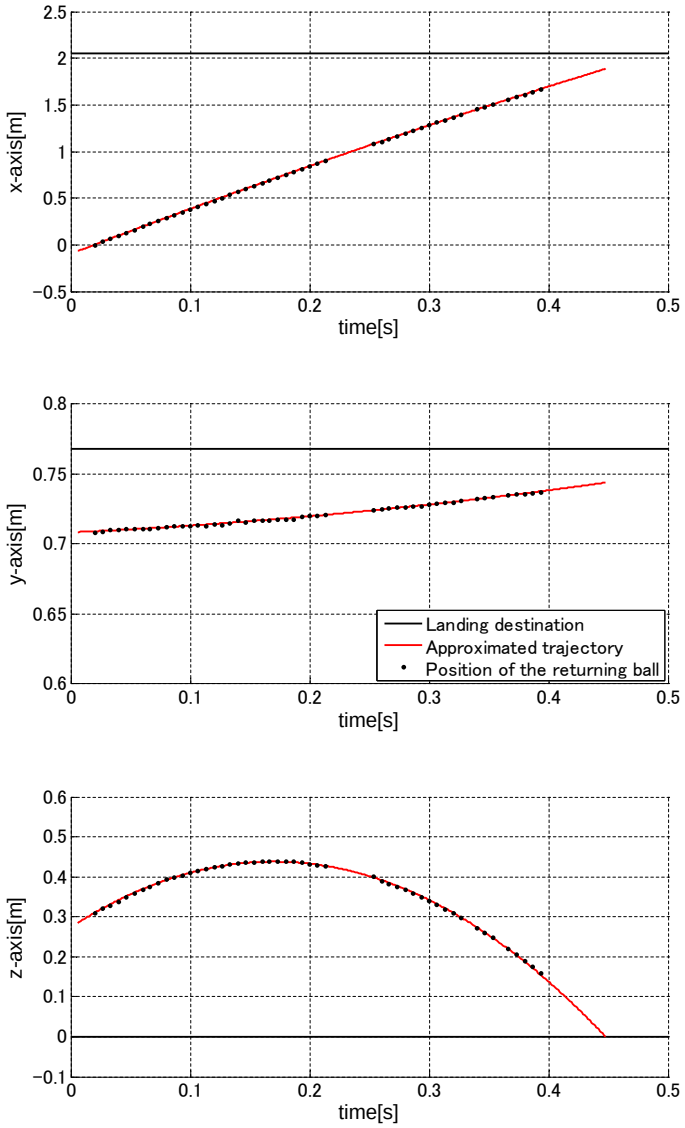
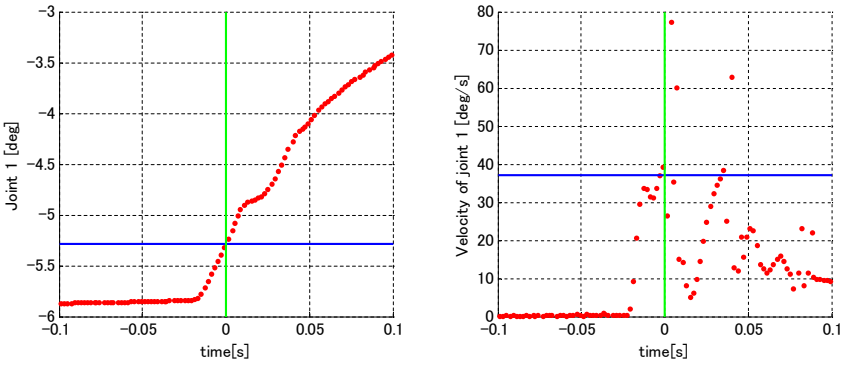


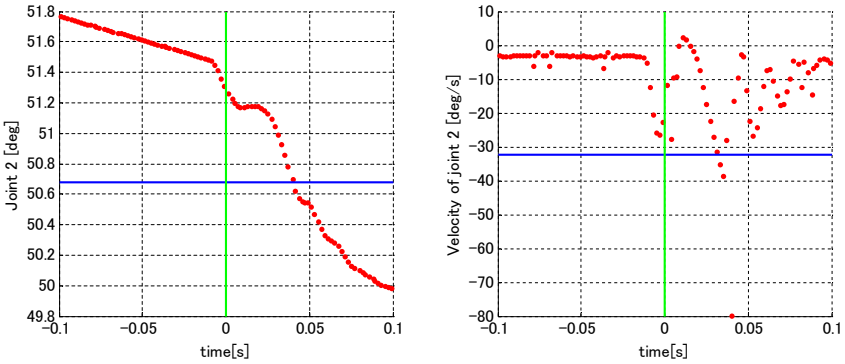
Figure 4.10: Ball time-history (Experiment): the case of topspin ball

got the similar results as the simulations.

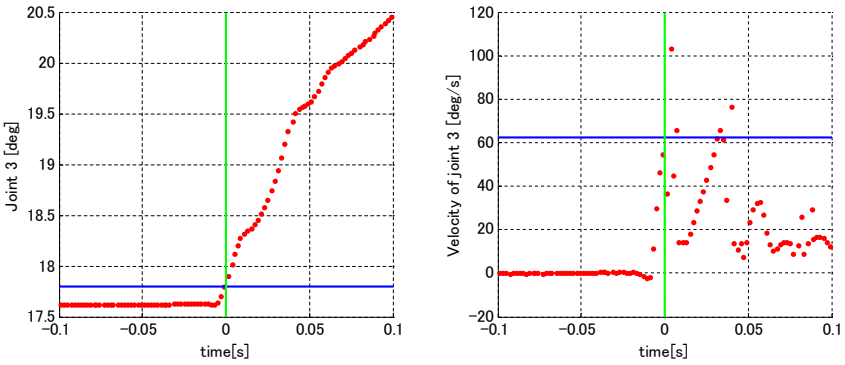
Furthermore, please notice that the method proposed in this chapter does not belong to a class of visual feedback control systems. In fact, the high speed cameras (900 Hz) have narrow fields so that the cameras can observe less than 10 frames of images just after the ball is shot from the catapult machine. After the cameras estimate the ball's translational and rotational velocities at this instant, it usually takes about 500 msec and more until the manipulator with the racket strikes the coming ball. During this interval, the proposed method controls the manipulator based on the physical models without any visual feedback. From this sense, the proposed method is not robust against the physical models' errors. One of the future researches is to integrate the proposed method with the visual feedback control.



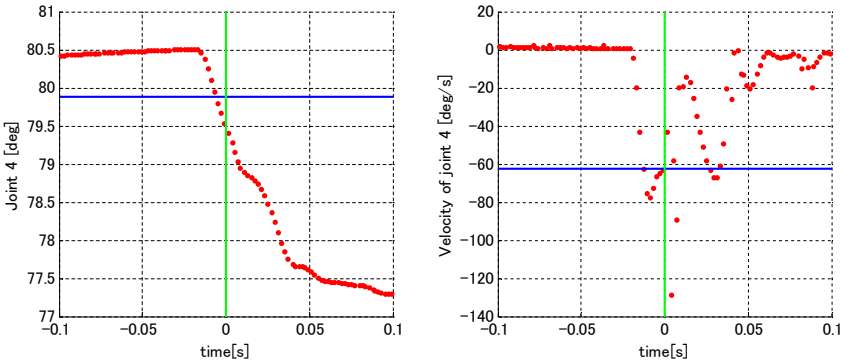
(a) The time-histories of q_1 (left) and $\dot{q}_1$ (right)



(b) The time-histories of q_2 (left) and $\dot{q}_2$ (right)

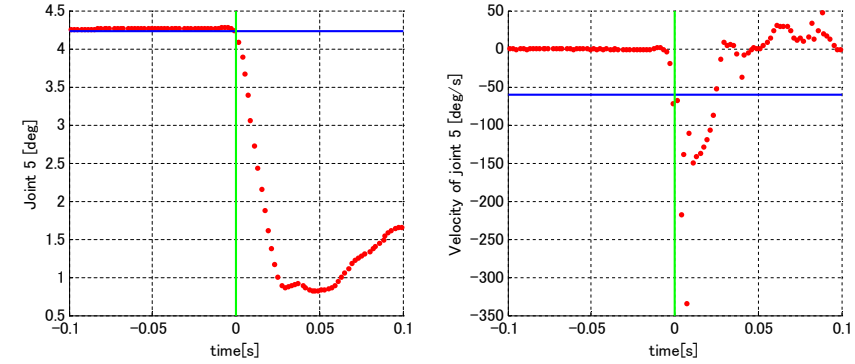


(c) The time-histories of q_3 (left) and $\dot{q}_3$ (right)

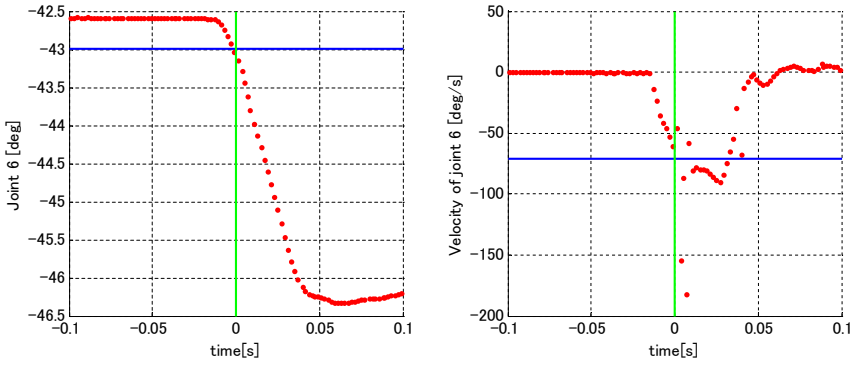


(d) The time-histories of q_4 (left) and $\dot{q}_4$ (right)

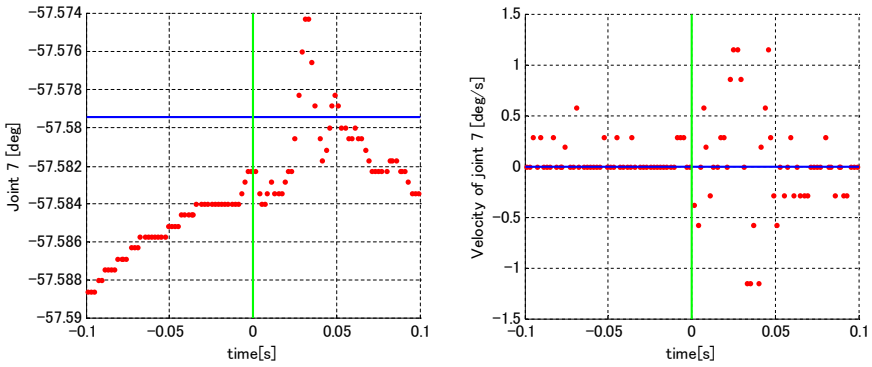
Figure 4.11: The 1, 2, 3, 4 joints' time-histories in the case of back spin ball



(a) The time-histories of q_5 (left) and $\dot{q}_5$ (right)

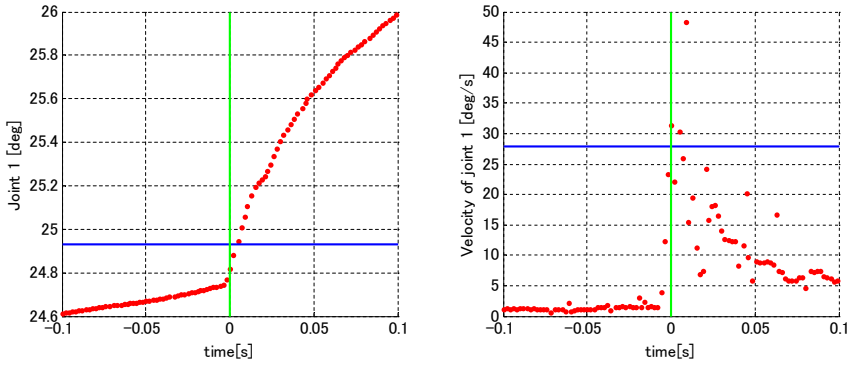


(b) The time-histories of q_6 (left) and $\dot{q}_6$ (right)

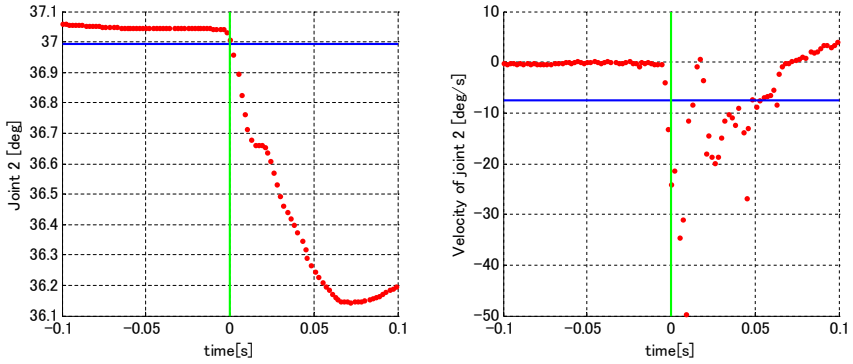


(c) The time-histories of q_7 (left) and $\dot{q}_7$ (right)

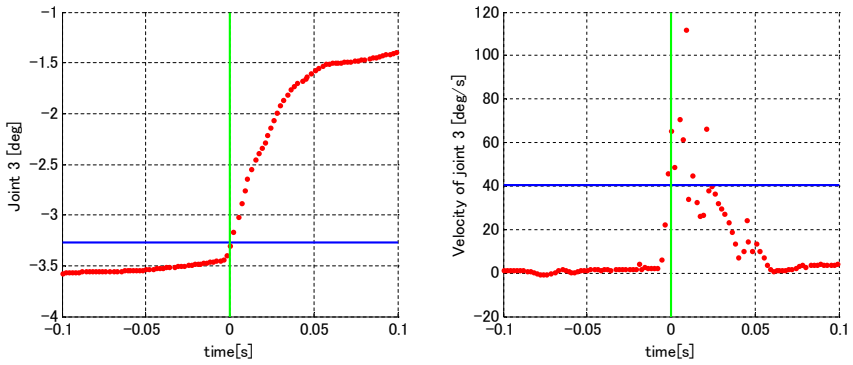
Figure 4.12: The 5, 6, 7 joints' time-histories in the case of back spin ball



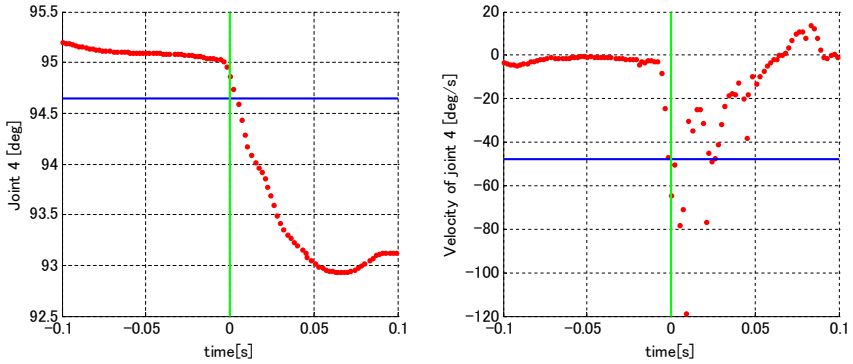
(a) The time-histories of q_1 (left) and $\dot{q}_1$ (right)



(b) The time-histories of q_2 (left) and $\dot{q}_2$ (right)

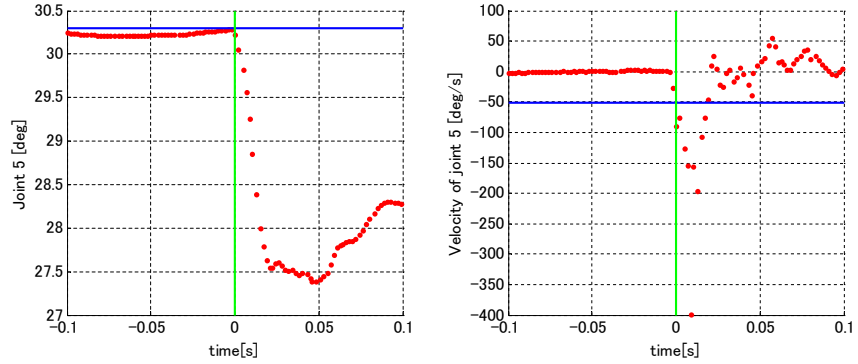


(c) The time-histories of q_3 (left) and $\dot{q}_3$ (right)

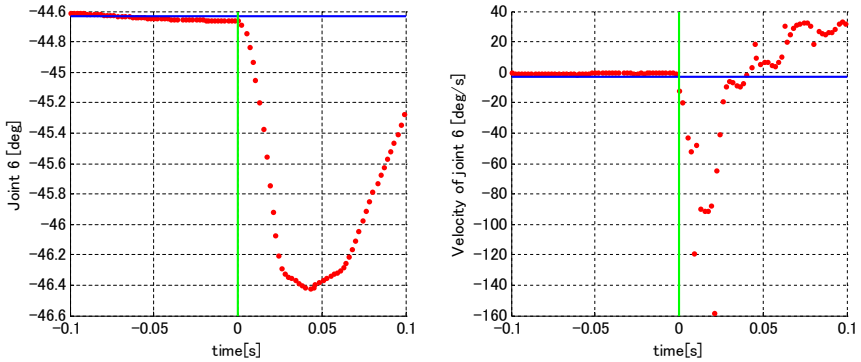


(d) The time-histories of q_4 (left) and $\dot{q}_4$ (right)

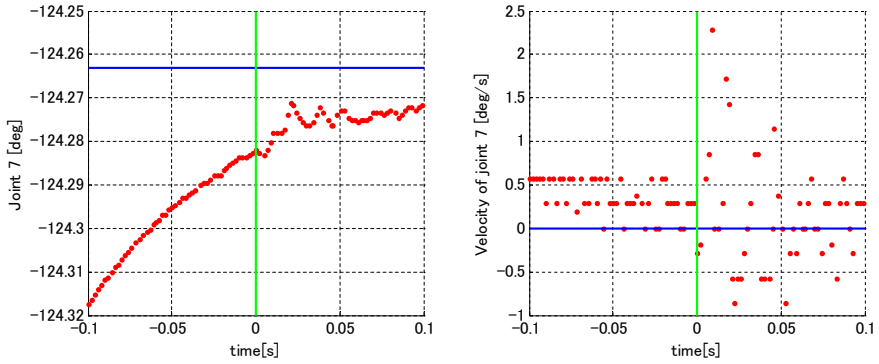
Figure 4.13: The 1, 2, 3, 4 joints' time-histories in the case of top spin ball



(a) The time-histories of q_5 (left) and $\dot{q}_5$ (right)



(b) The time-histories of q_6 (left) and $\dot{q}_6$ (right)



(c) The time-histories of q_7 (left) and $\dot{q}_7$ (right)

Figure 4.14: The 5, 6, 7 joints' time-histories in the case of top spin ball

Chapter 5

Conclusions

The conclusions are summarized in this chapter and some suggestions are shown for the further research.

5.1 Summary

Owing to the high requirements on *real-time*, *accuracy*, and *intelligence*, the research of table tennis robot has been a hot research topic in robotics since the year of 1983. Many different shapes of table tennis robots have been developed during these years such as Andersson's 6-DOF arm robot with 4 60Hz cameras, Acosta's 5-DOF low-cost two-paddle robot system with a single camera and Miyazaki's 4-DOF robot system with two cameras. These robot systems have been able to play against human opponents who are beginner level. However, "Robot beats humans" in a table tennis game is still a long way to go.

One common feature of the previous research is that the vision system can only measure the ball's position and translational velocity. The effect of the ball's rotational velocity is ignored because of the difficulty of measuring the ball's rotational velocity in an on-line situation. It is well known that table tennis is a "high technique" sport and the rotational velocity is an indispensable factor for the various kinds of techniques. Therefore, in order to realize a robot which can compete with humans in a game with high skills, it is necessary to consider the rotational velocity of the ball. This thesis tries to develop such a robot which can **detect** and **control** the ball's rotational velocity.

In Chapter 1, the systems of the presented table tennis robots are discussed in detail. Generally, it consists of three parts: the vision system (eye), the control system (brain) and the mechanical system (hand). With these systems, the table tennis robot has to accomplish three basic tasks:

1) Detecting a ball's state by using vision sensors,

2) Predicting the ball's trajectory by using the ball's state detected in task 1),

3) Determining the racket's posture and velocity at the striking time such that the returned ball would reach a given destination on the opponent table, planning a reference trajectory of the robot manipulator with the racket by the ball's trajectory predicted in task 2) and also controlling the robot according to the motion planning.

Corresponding to the above tasks, in order to develop a table tennis robot which can detect and control the rotational velocity of the ball, this thesis has done the following works:

1) Proposed a method for detecting a ball's *position, translational and rotational velocities* by using two high speed cameras (900 Hz) immediately after the opponent player hit the ball,

2) For predicting the ball's trajectory and performing work 3), introduced three physical models presented by our previous research: aerodynamics model, table rebound model and racket rebound model which considered both the ball's translational velocity and rotational velocity,

3) Proposed a racket control method which could return the ball to a desired position on the table with a desired rotational velocity at a desired landing time.

This thesis mainly focuses on **Task 1)** and **Task 3)**.

In Chapter 2, an *on-line* algorithm is proposed for measuring the translational and rotational velocities of a table tennis ball with only 6 consecutive frames of image data sensed by using two high speed cameras (900 Hz). With respect to estimating the translational velocity, one of the important processes is how to estimate the center of the ball image. The proposed method detects four points on the contour of the ball image and calculates the perpendicular bisectors between every two adjacent points of these contour points. Then, the ball's center on the image is estimated as the nearest point to these perpendicular bisectors. The rotational velocity is estimated by the image registration method, where the method of estimating $(k+1)$th frame of image data is proposed in detail by using the sensed kth frame of image data and the candidate of the rotational velocity. And also in order to search the most plausible rotational velocity, the conjugate gradient method is applied to minimize the intensity residuals between the estimated and the sensed frames. The experimental results with the rotation machine show that 89% of the estimation errors of the rotational speeds are smaller than 200 rpm and 75% of the estimation errors of the rotational axes are less than 15 deg. Moreover, the experimental results with the catapult machine demonstrate that both the translational and rotational velocities have been estimated well and the processing time is in the range of [30, 65] ms. Therefore, the proposed method is accurate and fast enough to realize a table tennis playing robot.

In Chapter 3, three physical models about ball motion are introduced, which have been presented by our research group: aerodynamics model (ADM), table rebound model (TRM) and racket rebound model (RRM). Comparing with the research of the previous presented table tennis robots, these three models have a significant improvement that the ball's rotational velocity is considered as one of the deciding factors of the ball motion. As for ADM, the ball's gravity and drag force as well as the Magnus force caused by the ball's rotational velocity are considered influencing the ball's trajectory. The rebound models (TRM and RRM) characterize the variation of the ball's translational and rotational velocities just before and after the rebounding. Some experimental data are shown to verify the effectiveness of these models, which include top spin balls, back spin balls and side spin balls. The experiments for two rebound models demonstrate that the estimated ball's velocities just after rebounding coincide well with the real velocities after rebounding and the experiments for the aerodynamics model display that the trajectory prediction is accurate enough for the racket to strike the balls, when

the table rebound model is used together. In fact, it has important meanings for building these three models since they are the basis for the robot to control a spinning table tennis ball.

In Chapter 4, a racket control method is proposed for returning the ball to a desired position on the opposite court with a desired rotational velocity at a desired landing time. Here, the determined racket state includes the translational velocity and the posture of the yaw and the pitch angles. And the determination method is based on the racket rebound model (RRM) and the aerodynamics model (ADM). The racket control problem consists of two subproblems; (a) the problem of solving a set of nonlinear equations which comes from RRM and (b) the two-point boundary value problem of the nonlinear differential equation which comes from ADM. This thesis clarifies the fundamental properties of those subproblems: for the first problem about RRM, an existence condition for real solutions of the set of nonlinear equations is shown and the solutions are expressed in the closed form; as for the second problem associated with ADM, it shows that the two-point boundary value problem of ADM needs too much computing time to be treated in real-time manner. Therefore, an on-line algorithm is proposed by using an approximate aerodynamic model (SAM). Some numerical simulations and experimental results have been demonstrated to verify the effectiveness of the proposed method. The numerical simulations show that the average errors of the landing position in x-axis and y-axis (the length and width directions of the table) are 0.082 m and 0.045 m, respectively. And the average error of the landing time is 0.025 s. The desired rotational velocity can be achieved exactly since the racket rebound model (RRM) is solved exactly. The experimental results are similar to the ones of the simulations.

Furthermore, it should be noted that the method proposed in Chapter 4 does not belong to a class of visual feedback control systems. In fact, the high speed cameras (900 Hz) have narrow fields so that the cameras can observe less than 10 frames of images just after the ball is shot from the catapult machine. After the cameras estimate the ball's translational and rotational velocities at this instant, it usually takes about 500 msec and more until the manipulator with the racket strikes the coming ball. During this interval, the proposed method controls the manipulator based on the physical models without any visual feedback. From this sense, the proposed method is not robust against the physical models' errors. One of the future researches is to integrate the proposed method with the visual feedback control.

5.2 Further Research

The followings are some suggestions for the further research.

In order to realize a robot which can **compete with human beings** in a table tennis game, this thesis proposes a real-time method for detecting the ball's translational/ rotational velocities and also presents a method for determining the racket's striking state, which is capable to control the rotational velocity of the returning ball. The proposed methods work well in the experiments, where the balls with different spins are shot from an automatic catapult machine. However, the developed robot system still can not play against a human opponent. The reasons mainly come from the problems of the hardware:

(1)*The reason caused by the vision system*: The high speed cameras (900 Hz) have two shortcomings which make them not suitable to be used for detecting the ball's state when the robot rallies with a human being. The first shortcoming is that the high speed cameras have a narrow field of view. The resolution of the camera is 232×232 and the field of view of the two cameras is only about 15 cm$\times$15 cm which is too small to be used for detecting the ball

which moves in all the range of the table. The second shortcoming is that a strict additional light condition is required for the experiment since the exposure time for one frame is too short. This also limits the flying range of the ball since the estimation accuracy of the ball's rotational velocity will be affected if the ball is too near or too far from the light.

Suggestion for the vision system: Use middle speed cameras for estimating the ball's position, translational and rotational velocities. In fact, our group has developed another method for estimating the ball's rotational velocity by using two middle speed cameras (150 Hz) of which the field of view is large enough to cover all the table and additional light condition is not needed.

(2)*The reason caused by the manipulator*: The mechanical system applied in this thesis is a 7-DOF manipulator. There are some limits for the 7-DOF manipulator. Firstly, as shown in Sec.4.5 of Chapter 4, since there are two redundant degrees of freedom, the kinematic equation for solving the striking postures of the 7 joints has infinite number of solutions. The methods for obtaining one solution always cost much computation time. Furthermore, the limit velocity of the racket attached on the end-effector is about 3 m/s, which should be faster for high speed coming balls. Finally, the workspace of the manipulator is not large enough.

Suggestion for the manipulator: In fact, for accomplishing the table tennis striking task, 5 degrees of freedom are enough (three degrees of freedom for determining the racket's position in X, Y, Z coordinates and two degrees of freedom for determining the posture of the racket at the striking time). And for a 5-DOF robot, there is no redundant degree of freedom. Then, solving the 5 joints' striking postures from the inverse kinematics is much easier, which will reduce the computing time greatly. Therefore, using a 5-DOF robot is suggested. Another point, it is better to use a lighter arm for moving faster. As for enlarging the work space, we suggest that the robot arm can move in a frame attached on the table. Actually, we have developed a new 5-DOF robot which has these good features.

Furthermore, I would like to give some comments on improving three models of ball motion, i.e., the aerodynamics model, the table rebound model, and the racket rebound model.

These models are very important in this research because the racket control has been carried based on those physical models. The models constructed and used in this research could be verified to be correct with enough precision and to work well by carrying out the experiments. However, if you wish to predict the coming ball's position and velocity with more precision as well as to return the ball to the desired position on the opposite court more precisely, you can improve those physical models. For instance, with respect to the aerodynamics model, the rotational velocity is assumed here to be constant when the ball flies, but in fact, the rotational velocity would be changing. About the table rebound model, it was constructed under Assumptions 1 to 5 in Chapter 3 and some parameters of the model were identified by the experimental data. Rigorously speaking, the contact between the ball and the table could not be a point contact, and the phenomena assumed in collision between the ball and the racket would not be real either. When you want to construct those models more realistically, you need to observe and analyze those phenomena more precisely by using super high speed cameras and other precision instruments.

Appendix A

Table Tennis System

Figure A.1 shows the configuration of the table tennis system and the coordinate systems on the table and the manipulator. The equipments for the system include the table, ball, racket, two high speed cameras, 7 DOF manipulator, automatic catapult machine and PCI memolink. In addition, two middle speed cameras have been used in the experiments (Chapter 4) to detect the returning ball's trajectory for evaluating how well the ball achieves to the desired destination. In this section, the parameters of these equipments are described in detail.

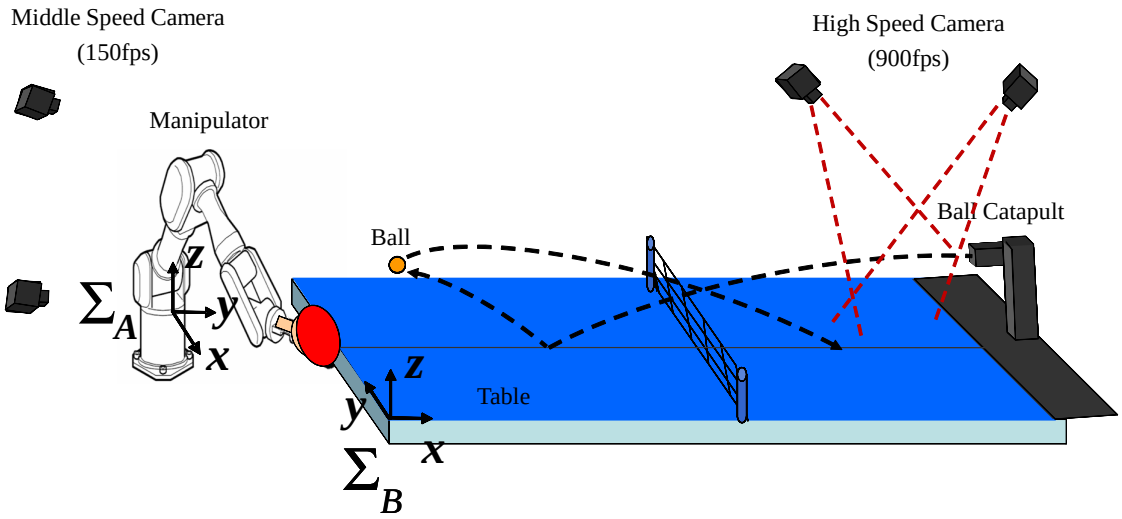


Figure A.1: Configuration of the table tennis system

A.1 Table, Ball, Racket

Figure A.2 displays the size of the table which is an international standard one. Table A.1 is the specification of the table. The table tennis ball used (Table A.2) is an standard one recognized by International Table Tennis Federation. Table A.3 shows the specification of the racket's rubber.

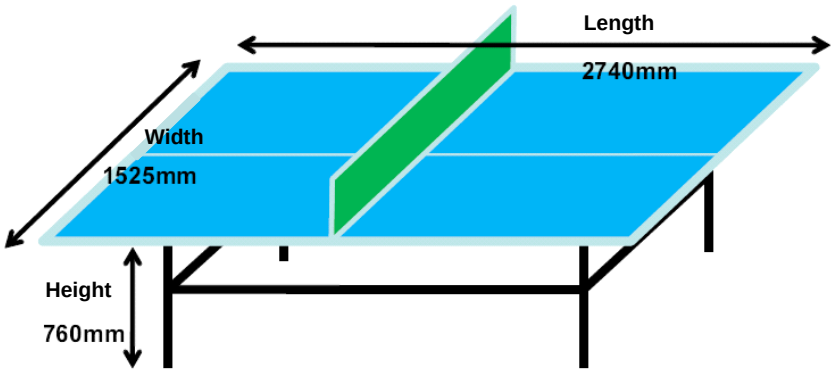


Figure A.2: Standard table size

Table A.1: The specification of table tennis table

Maker	IGNIO
Stock Number	IG-2PG0016
Playfield	MDF (Material), 18[mm](Thickness)
Frame	Steel 30[mm]
Size	1525(W)×760(H)×2740(D)
Weight	90kg

Table A.2: The specification of table tennis ball

Maker	Nittaku
Maker Type	3 Star(Premium), 40mm Hard ball
Diameter	40[mm]
Color	White
Weight	2.7[g]

Table A.3: The rubber of the racket

Maker	Japanese table tennis corporation
Maker Type	MORISTO DF
Thickness	3[mm]
Color	Black

A.2 High and Middle Speed Cameras

A.2.1 High speed camera

In the table tennis system, two high speed cameras (Table A.5) are used for detecting the ball's translational and rotational velocities. Table A.4 and Table A.6 are the specifications of the lens and the sensor, respectively.

Table A.4: The specification of the lens

Maker	FUJINON
Model	HF35HA-1B
Focal Length	35[mm]
Diaphragm	F1.6∼F22
Hyperfocal Distance	∞∼0.25[m]

Table A.5: The specification of high speed camera

Active Sensor Size	10.24mm×10.24mm
Array Size	512×512(random part readout ex.232×232)
Pixel Size	29μm
Aperture Ratio	45%
Frame Rate	250fps(1000fps for 232×232 array size)
Power Consumption	6W
Gradation	10bit
Others	Readout the original and processed image data Use Visual C++ interface library

Table A.6: The specification of the sensor

	Profile Sensor	Image Sensor
Process	0.6μm, 2-poly, 3-metal CMOS Process	
Chip Size	13.0×14.3mm	
Pixel Size	20μm	
Pixel Number	512×512	512×512
Readout Model	PPS	APS
Aperture Ratio	2% (×2)	44%
Highest Frame Rate	1620fps	248fps(512×512), 2421fps(128×128)
Power Voltage	5V(Analog), 3V(Digital)	5V
Power Consumption	75mW@1000fps	450mW@125fps
Sensitivity	0.8μV/elec.	10.7μV/elec.
Dark Output Voltage	300mV/sec@25°C	100mV/sec@25°C
Random Noise	12mVrms	0.7mVrms
Saturation Output Voltage	3.2V	1.4V
Number of Port	2(X,Y)	8
ADC	8bit ADC Internal Model(2 Number)	No(External Model)
Output Form	2 ports, digital output	8 ports, analog output

A.2.2 Middle speed camera

The middle speed camera is produced by Library co.,Ltd (Table A.7), which is capable to track high speed moving objects in real-time situation with an image processing software. Table A.8 is the specification of the lens for the middle speed camera.

Table A.7: Radish camera (middle speed camera)

Active Sensor Size	4.8mm×3.6mm
Array Size	640× 480
Pixel Size	7.4μm
Frame Rate	150fps
Power Consumption	1A
Gradation	Color 24bit

Table A.8: The specification of the lens (middle speed camera)

Maker	VS Technology
Product Name	SV-0814MP
Focal Length	8.3[mm]
Diaphragm	F1.4∼F16
Hyperfocal Distance	∞∼0.1[m]

A.3 7-DOF Manipulator

A 7-DOF manipulator (Mitsubishi) is applied as the mechanical system. Figure A.3 shows the general view of the manipulator. Tables A.9, A.10 and A.11 display the specifications of 3 shoulder joints, 2 elbow joints and 2 wrist joints, respectively. Table A.12 is the maximum speed of each joint.

Table A.9: The specification of the manipulator (shoulder 3 joints)

Joint Name		S1	S2	S3
Motor	Rated Torque [N·m]	4.64	4.64	2.00
	Rated Current [A]	10	10	10
	Rotation Direction	CCW	CCW	CCW
Arm	Working Area [deg]	-177∼177	-94∼94	-174∼174

Table A.10: The specification of the manipulator (elbow 2 joints)

Joint Name		E1	E2
Motor	Rated Torque [N·m]	2.00	0.29
	Rated Current [A]	10	10
	Rotation Direction	CCW	CCW
Arm	Working Area [deg]	-137∼137	-255∼255

A.4 Others

A.4.1 Automatic catapult machine

Table A.13 is the specification of the automatic catapult machine.

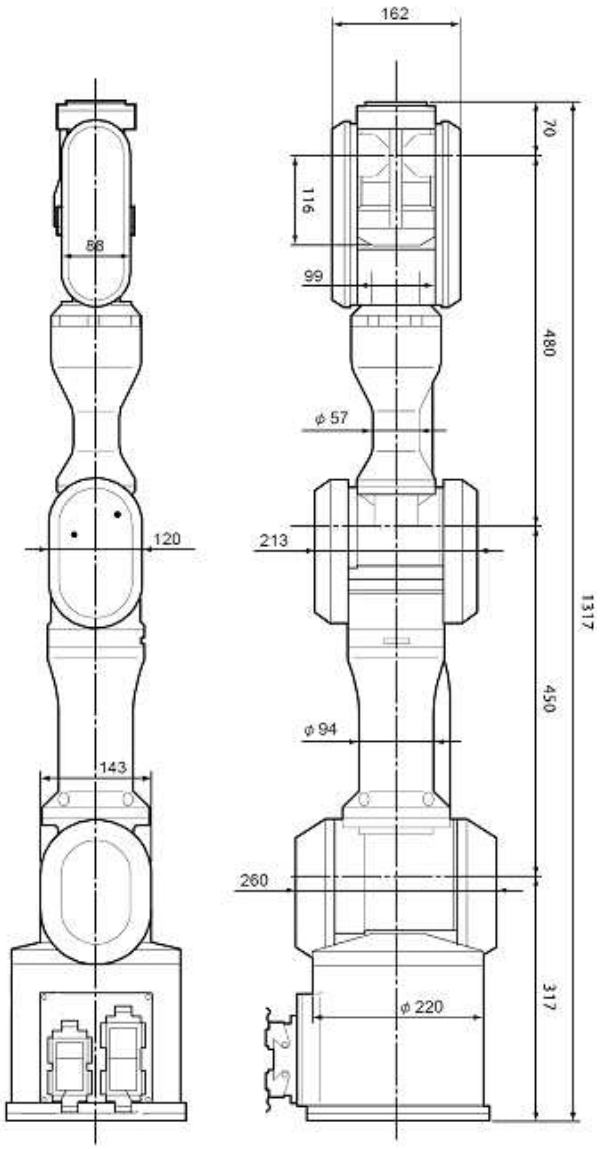


Figure A.3: The general view of the manipulator

Table A.11: The specification of the manipulator (wrist 2 joints)

Joint Name		W1	W2
Motor	Rated Torque [N·m]	0.29	0.29
	Rated Current [A]	10	10
	Rotation Direction	CCW	CCW
Arm	Working Area [deg]	-165∼165	-255∼255

A.4.2 PCI memolink

There are two PCs in the table tennis system that one is for the vision system and another one is for the control system. These two PCs are connected by a PCI Memolink for transmit-

Table A.12: Maximum moving velocity of each joint

Maximum Velocity [deg/sec]	1 joint	2 joint	3 joint	4 joint	5 joint	6 joint	7 joint
	57.3	57.3	114.6	114.6	360.0	360.0	360.0

Table A.13: The specification of automatic catapult machine

Maker	SAN-EI
Maker Type	Table Tennis Machine Robo-Pong 2040
Power	100[V]
Ball Speed	16∼120[km/h]
Ball Frequency	26∼94[number of ball/min]
Functions	Regulate the oscillator speed of the machine head. Position the head in the desired direction. Adjust the angle of the machine head (up and down) by hand. Put any type of spin on the ball(top spin, back spin, side spin and so on). Be capable of using either the 38[mm] or 40[mm] ball.

ting the measured ball's information (See Figure A.4). Table A.14 is the specification of the Memolink.

Table A.14: PCI Memolink

Maker	Interface	
Product Name	PCI4911	PCI4915
Memolink Type	Master	Terminator Slave
Connection Type	Master:Slave = 1:8	
Maximum Transfer Velocity	512KB/s[Byte],1MB/s[Byte]	
Maximum Transfer Length	30m	
Memory Size	256KB	
Data Protection	Parity check	
Bus	PCI local bus 32 bit, 33MHz	
Interrupt Facility	Have	
Power Consumption	DC+5V(±):1.4A(TYP)	DC+5V(±):0.3A(TYP)
I/O Port	8 Port	
Memory Size	16KB	
Number of Slot	1	

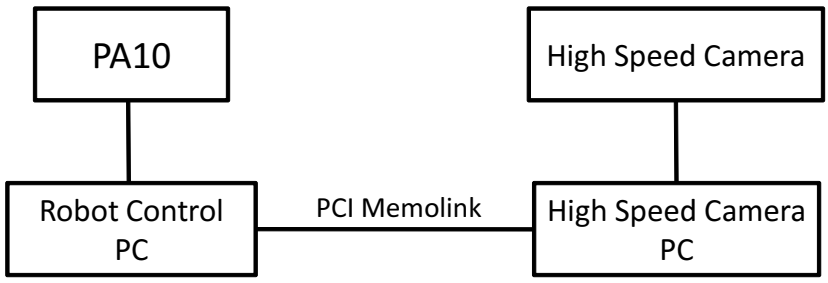


Figure A.4: Connection of PCs with a PCI Memolink

Appendix B

Robot Kinematics and Cameras' Calibrations

B.1 Robot Kinematics

In this thesis, the robot system is a 7 DOF manipulator controlled with PID. When the robot executes tasks, it is necessary to transform the control space since the manipulator is controlled in the work space while its actuators operate in the joint space [12]. Forward kinematics defines a function (kinematics equations) to compute the position and the posture of the end-effector from the specified joints' postures. Then, the velocity relations between them are determined by the Jacobian of this function. The inverse kinematics is the reverse process of this function, which computes the joints' postures for achieving a specified position and posture of the end-effector. In this section, we introduce the derivation of the forward kinematic equations and the corresponding Jacobian matrix.

B.1.1 Frame arrangement of the manipulator

Figure B.1 displays the frame arrangement of the manipulator.

B.1.2 Forward kinematics

Based on the frame of each joint, forward kinematics can be solved by using the homogenous transformation as follows. Notice that the angle of the *i*th joint is expressed as θ_i and $C_i := \cos\theta_i$, $S_i := \sin\theta_i$.

$$ {}^{0}T_1 = \begin{bmatrix} C_1 & -S_1 & 0 & 0 \\ S_1 & C_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{B.1} $$

$$ {}^{1}T_2 = \begin{bmatrix} C_2 & -S_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -S_2 & -C_2 & 0 & l_b \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{B.2} $$

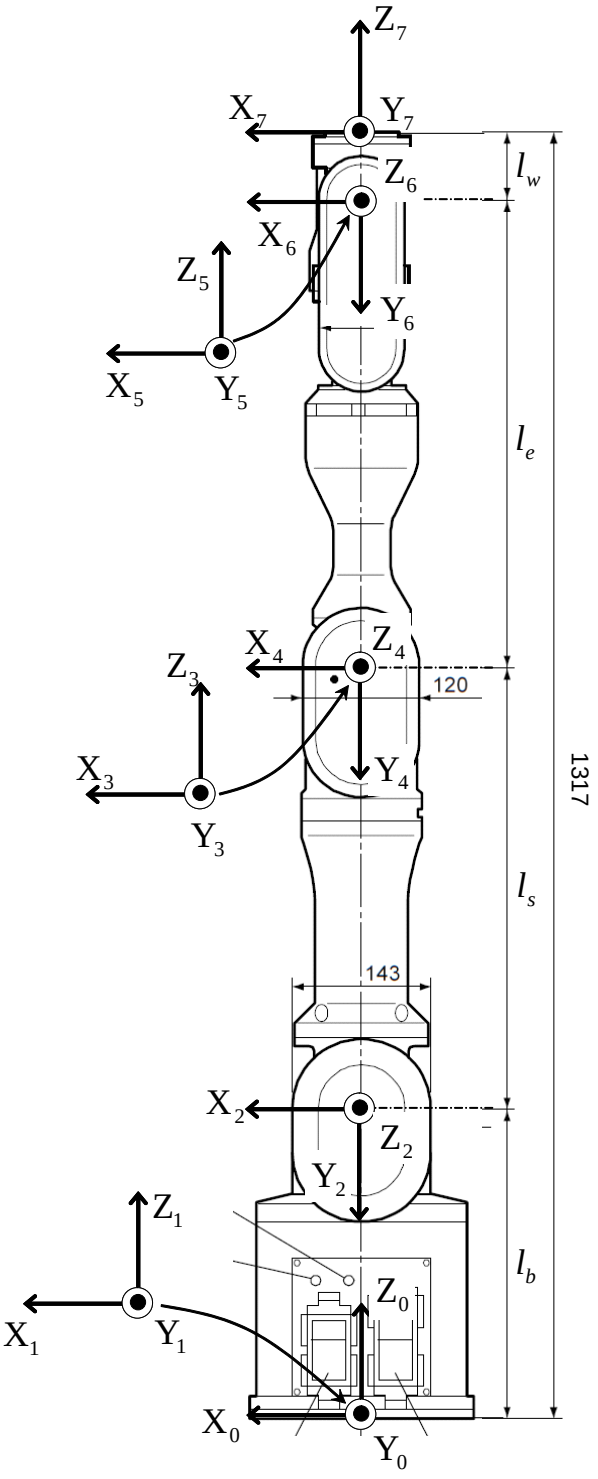


Figure B.1: Frame arrangement of the manipulator

$$ {}^{2}T_3 = \begin{bmatrix} C_3 & -S_3 & 0 & 0 \\ 0 & 0 & -1 & -l_s \\ S_3 & C_3 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{B.3} $$

$$ {}^{3}T_{4} = \begin{bmatrix} C_4 & -S_4 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -S_4 & -C_4 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{B.4} $$

$$ {}^{4}T_{5} = \begin{bmatrix} C_5 & -S_5 & 0 & 0 \\ 0 & 0 & -1 & -l_e \\ S_5 & C_5 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{B.5} $$

$$ {}^{5}T_{6} = \begin{bmatrix} C_6 & -S_6 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -S_6 & -C_6 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{B.6} $$

$$ {}^{6}T_{7} = \begin{bmatrix} C_7 & -S_7 & 0 & 0 \\ 0 & 0 & -1 & -l_w \\ S_7 & C_7 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{B.7} $$

where ${}^{0}T_{7} = {}^{0}T_{1}\ {}^{1}T_{2}\ {}^{2}T_{3}\ {}^{3}T_{4}\ {}^{4}T_{5}\ {}^{5}T_{6}\ {}^{6}T_{7}$ can be expressed as:

$$ {}^{0}T_{7} = \begin{bmatrix} t_{11} & t_{12} & t_{13} & t_{14} \\ t_{21} & t_{22} & t_{23} & t_{24} \\ t_{31} & t_{32} & t_{33} & t_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{B.8} $$

where

$$ \begin{aligned} t_{11} = {} & C_7C_6C_5C_4C_1C_2C_3 - C_7C_6C_5C_4S_1S_3 - C_7C_6C_5C_1S_2S_4 - C_7C_6S_5C_1C_2S_3 \\ & - C_7C_6S_5S_1C_3 - C_7S_6S_4C_1C_2C_3 + C_7S_6S_4S_1S_3 - C_7S_6C_1S_2C_4 - S_7S_5C_4C_1C_2C_3 \\ & + S_7S_5C_4S_1S_3 + S_7S_5C_1S_2S_4 - S_7C_5C_1C_2S_3 - S_7C_5S_1C_3 \end{aligned} \tag{B.9} $$

$$ \begin{aligned} t_{12} = {} & -S_7C_6C_5C_4C_1C_2C_3 + S_7C_6C_5C_4S_1S_3 + S_7C_6C_5C_1S_2S_4 + S_7C_6S_5C_1C_2S_3 \\ & + S_7C_6S_5S_1C_3 + S_7S_6S_4C_1C_2C_3 - S_7S_6S_4S_1S_3 + S_7S_6C_1S_2C_4 - C_7S_5C_4C_1C_2C_3 \\ & + C_7S_5C_4S_1S_3 + C_7S_5C_1S_2S_4 - C_7C_5C_1C_2S_3 - C_7C_5S_1C_3 \end{aligned} \tag{B.10} $$

$$ \begin{aligned} t_{13} = {} & S_6C_5C_4C_1C_2C_3 - S_6C_5C_4S_1S_3 - S_6C_5C_1S_2S_4 - S_6S_5C_1C_2S_3 - S_6S_5S_1C_3 \\ & + C_6S_4C_1C_2C_3 - C_6S_4S_1S_3 + C_6C_1S_2C_4 \end{aligned} \tag{B.11} $$

$$ \begin{aligned} t_{14} = {} & l_wS_6C_5C_4C_1C_2C_3 - l_wS_6C_5C_4S_1S_3 - l_wS_6C_5C_1S_2S_4 - l_wS_6S_5C_1C_2S_3 \\ & - l_wS_6S_5S_1C_3 + l_wC_6S_4C_1C_2C_3 - l_wC_6S_4S_1S_3 + l_wC_6C_1S_2C_4 + l_eS_4C_1C_2C_3 \\ & - l_eS_4S_1S_3 + l_eC_1S_2C_4 + C_1S_2l_s \end{aligned} \tag{B.12} $$

$$ \begin{aligned} t_{21} = {} & C_7C_6C_5C_4S_1C_2C_3 + C_7C_6C_5C_4C_1S_3 - C_7C_6C_5S_1S_2S_4 - C_7C_6S_5S_1C_2S_3 \\ & + C_7C_6S_5C_1C_3 - C_7S_6S_4S_1C_2C_3 - C_7S_6S_4C_1S_3 - C_7S_6S_1S_2C_4 - S_7S_5C_4S_1C_2C_3 \\ & - S_7S_5C_4C_1S_3 + S_7S_5S_1S_2S_4 - S_7C_5S_1C_2S_3 + S_7C_5C_1C_3 \end{aligned} \tag{B.13} $$

$$t_{22} = -S_7C_6C_5C_4S_1C_2C_3 - S_7C_6C_5C_4C_1S_3 + S_7C_6C_5S_1S_2S_4 + S_7C_6S_5S_1C_2S_3 \quad \text{(B.14)}$$
$$- S_7C_6S_5C_1C_3 + S_7S_6S_4S_1C_2C_3 + S_7S_6S_4C_1S_3 + S_7S_6S_1S_2C_4 - C_7S_5C_4S_1C_2C_3$$
$$- C_7S_5C_4C_1S_3 + C_7S_5S_1S_2S_4 - C_7C_5S_1C_2S_3 + C_7C_5C_1C_3$$

$$t_{23} = S_6C_5C_4S_1C_2C_3 + S_6C_5C_4C_1S_3 - S_6C_5S_1S_2S_4 - S_6S_5S_1C_2S_3 + S_6S_5C_1C_3 \quad \text{(B.15)}$$
$$+ C_6S_4S_1C_2C_3 + C_6S_4C_1S_3 + C_6S_1S_2C_4$$

$$t_{24} = l_wS_6C_5C_4S_1C_2C_3 + l_wS_6C_5C_4C_1S_3 - l_wS_6C_5S_1S_2S_4 - l_wS_6S_5S_1C_2S_3 \quad \text{(B.16)}$$
$$+ l_wS_6S_5C_1C_3 + l_wC_6S_4S_1C_2C_3 + l_wC_6S_4C_1S_3 + l_wC_6S_1S_2C_4 + l_eS_4S_1C_2C_3$$
$$+ l_eS_4C_1S_3 + l_eS_1S_2C_4 + S_1S_2l_s$$

$$t_{31} = -C_7C_6C_5S_2C_3C_4 - C_7C_6C_5C_2S_4 + C_7C_6S_2S_3S_5 + C_7S_6S_2C_3S_4 \quad \text{(B.17)}$$
$$- C_7S_6C_2C_4 + S_7S_5S_2C_3C_4 + S_7S_5C_2S_4 + S_7S_2S_3C_5$$

$$t_{32} = S_7C_6C_5S_2C_3C_4 + S_7C_6C_5C_2S_4 - S_7C_6S_2S_3S_5 - S_7S_6S_2C_3S_4 + S_7S_6C_2C_4 \quad \text{(B.18)}$$
$$+ C_7S_5S_2C_3C_4 + C_7S_5C_2S_4 + C_7S_2S_3C_5$$

$$t_{33} = -S_6C_5S_2C_3C_4 - S_6C_5C_2S_4 + S_6S_2S_3S_5 - C_6S_2C_3S_4 + C_6C_2C_4 \quad \text{(B.19)}$$

$$t_{34} = -l_wS_6C_5S_2C_3C_4 - l_wS_6C_5C_2S_4 + l_wS_6S_2S_3S_5 - l_wC_6S_2C_3S_4 + l_wC_6C_2C_4 \quad \text{(B.20)}$$
$$- l_eS_2C_3S_4 + l_eC_2C_4 + C_2l_s + l_b$$

B.1.3 Jacobian matrix

Equation (B.21) shows the velocity kinematics which relates the translational velocity and the angular velocity of the end-effector ($\dot{\boldsymbol{p}}, \boldsymbol{\omega}$) with the joints' velocity $\dot{\boldsymbol{\theta}}$ by the Jacobian matrix J. Note that $\dot{\theta}_i (i = 1, 2, \cdots, 7)$ is the velocity of the ith joint.

$$\begin{bmatrix} \dot{\boldsymbol{p}} \\ \boldsymbol{\omega} \end{bmatrix} = J\dot{\boldsymbol{\theta}} \quad \text{(B.21)}$$

$$J = \begin{bmatrix} j_{11} & j_{12} & j_{13} & j_{14} & j_{15} & j_{16} & j_{17} \\ j_{21} & j_{22} & j_{23} & j_{24} & j_{25} & j_{26} & j_{27} \\ j_{31} & j_{32} & j_{33} & j_{34} & j_{35} & j_{36} & j_{37} \\ j_{41} & j_{42} & j_{43} & j_{44} & j_{45} & j_{46} & j_{47} \\ j_{51} & j_{52} & j_{53} & j_{54} & j_{55} & j_{56} & j_{57} \\ j_{61} & j_{62} & j_{63} & j_{64} & j_{65} & j_{66} & j_{67} \end{bmatrix} \quad \text{(B.22)}$$

in which the element is displayed as follows:

$$j_{11} = (-(((S_1C_2C_3 + C_1S_3)C_4 - S_1S_2S_4)C_5 + (-S_1C_2S_3 + C_1C_3)S_5)S_6 \quad \text{(B.23)}$$
$$- ((S_1C_2C_3 + C_1S_3)S_4 + S_1S_2C_4)C_6)lw - ((S_1C_2C_3 + C_1S_3)S_4 + S_1S_2C_4)le$$
$$- S_1S_2ls$$

$$j_{12} = C_1(-(-((-S_2C_3C_4 - C_2S_4)C_5 + S_2S_3S_5)S_6 - (-S_2C_3S_4 + C_2C_4)C_6)lw + (-S_2C_3S_4 + C_2C_4)le + C_2ls) \tag{B.24}$$

$$j_{13} = S_1S_2(-(-((-S_2C_3C_4 - C_2S_4)C_5 + S_2S_3S_5)S_6 - (-S_2C_3S_4 + C_2C_4)C_6)lw + (-S_2C_3S_4 + C_2C_4)le + C_2ls) - C_2(-(-(((S_1C_2C_3 + C_1S_3)C_4 - S_1S_2S_4)C_5 + (-S_1C_2S_3 + C_1C_3)S_5)S_6 - ((S_1C_2C_3 + C_1S_3)S_4 + S_1S_2C_4)C_6)lw + ((S_1C_2C_3 + C_1S_3)S_4 + S_1S_2C_4)le + S_1S_2ls) \tag{B.25}$$

$$j_{14} = (-S_1C_2S_3 + C_1C_3)(-(-((-S_2C_3C_4 - C_2S_4)C_5 + S_2S_3S_5)S_6 - (-S_2C_3S_4 + C_2C_4)C_6)lw + (-S_2C_3S_4 + C_2C_4)le) - S_2S_3(-(-(((S_1C_2C_3 + C_1S_3)C_4 - S_1S_2S_4)C_5 + (-S_1C_2S_3 + C_1C_3)S_5)S_6 - ((S_1C_2C_3 + C_1S_3)S_4 + S_1S_2C_4)C_6)lw + ((S_1C_2C_3 + C_1S_3)S_4 + S_1S_2C_4)le) \tag{B.26}$$

$$j_{15} = ((S_1C_2C_3 + C_1S_3)S_4 + S_1S_2C_4)(-(-((-S_2C_3C_4 - C_2S_4)C_5 + S_2S_3S_5)S_6 - (-S_2C_3S_4 + C_2C_4)C_6)lw + (-S_2C_3S_4 + C_2C_4)le) - (-S_2C_3S_4 + C_2C_4)(-(-(((S_1C_2C_3 + C_1S_3)C_4 - S_1S_2S_4)C_5 + (-S_1C_2S_3 + C_1C_3)S_5)S_6 - ((S_1C_2C_3 + C_1S_3)S_4 + S_1S_2C_4)C_6)lw + ((S_1C_2C_3 + C_1S_3)S_4 + S_1S_2C_4)le) \tag{B.27}$$

$$j_{16} = -(-((S_1C_2C_3 + C_1S_3)C_4 - S_1S_2S_4)S_5 + (-S_1C_2S_3 + C_1C_3)C_5)(-((-S_2C_3C_4 - C_2S_4)C_5 + S_2S_3S_5)S_6 - (-S_2C_3S_4 + C_2C_4)C_6)lw + (-(-S_2C_3C_4 - C_2S_4)S_5 + S_2S_3C_5)(-(((S_1C_2C_3 + C_1S_3)C_4 - S_1S_2S_4)C_5 + (-S_1C_2S_3 + C_1C_3)S_5)S_6 - ((S_1C_2C_3 + C_1S_3)S_4 + S_1S_2C_4)C_6)lw \tag{B.28}$$

$$j_{17} = 0 \tag{B.29}$$

$$j_{21} = -(-(((C_1C_2C_3 - S_1S_3)C_4 - C_1S_2S_4)C_5 + (-C_1C_2S_3 - S_1C_3)S_5)S_6 - ((C_1C_2C_3 - S_1S_3)S_4 + C_1S_2C_4)C_6)lw + ((C_1C_2C_3 - S_1S_3)S_4 + C_1S_2C_4)le + C_1S_2ls \tag{B.30}$$

$$j_{22} = S_1(-(-((-S_2C_3C_4 - C_2S_4)C_5 + S_2S_3S_5)S_6 - (-S_2C_3S_4 + C_2C_4)C_6)lw + (-S_2C_3S_4 + C_2C_4)le + C_2ls) \tag{B.31}$$

$$j_{23} = C_2(-(-(((C_1C_2C_3 - S_1S_3)C_4 - C_1S_2S_4)C_5 + (-C_1C_2S_3 - S_1C_3)S_5)S_6 - ((C_1C_2C_3 - S_1S_3)S_4 + C_1S_2C_4)C_6)lw + ((C_1C_2C_3 - S_1S_3)S_4 + C_1S_2C_4)le + C_1S_2ls) - C_1S_2(-(-((-S_2C_3C_4 - C_2S_4)C_5 + S_2S_3S_5)S_6 - (-S_2C_3S_4 + C_2C_4)C_6)lw + (-S_2C_3S_4 + C_2C_4)le + C_2ls) \tag{B.32}$$

$$\begin{aligned} j_{24} = S_2S_3(-(-(((C_1C_2C_3 - S_1S_3)C_4 - C_1S_2S_4)C_5 + (-C_1C_2S_3 - S_1C_3)S_5)S_6 \\ - ((C_1C_2C_3 - S_1S_3)S_4 + C_1S_2C_4)C_6)lw + ((C_1C_2C_3 - S_1S_3)S_4 + C_1S_2C_4)le \\ - (-C_1C_2S_3 - S_1C_3)(-(-((-S_2C_3C_4 - C_2S_4)C_5 + S_2S_3S_5)S_6 \\ - (-S_2C_3S_4 + C_2C_4)C_6)lw + (-S_2C_3S_4 + C_2C_4)le) \end{aligned} \tag{B.33}$$

$$\begin{aligned} j_{25} = (-S_2C_3S_4 + C_2C_4)(-(-(((C_1C_2C_3 - S_1S_3)C_4 - C_1S_2S_4)C_5 \\ + (-C_1C_2S_3 - S_1C_3)S_5)S_6 - ((C_1C_2C_3 - S_1S_3)S_4 + C_1S_2C_4)C_6)lw \\ + ((C_1C_2C_3 - S_1S_3)S_4 + C_1S_2C_4)le) \\ - ((C_1C_2C_3 - S_1S_3)S_4 + C_1S_2C_4)(-(-((-S_2C_3C_4 - C_2S_4)C_5 + S_2S_3S_5)S_6 \\ - (-S_2C_3S_4 + C_2C_4)C_6)lw + (-S_2C_3S_4 + C_2C_4)le) \end{aligned} \tag{B.34}$$

$$\begin{aligned} j_{26} = -(-(-S_2C_3C_4 - C_2S_4)S_5 + S_2S_3C_5)(-(((C_1C_2C_3 - S_1S_3)C_4 - C_1S_2S_4)C_5 \\ + (-C_1C_2S_3 - S_1C_3)S_5)S_6 - ((C_1C_2C_3 - S_1S_3)S_4 + C_1S_2C_4)C_6)lw \\ + (-((C_1C_2C_3 - S_1S_3)C_4 - C_1S_2S_4)S_5 \\ + (-C_1C_2S_3 - S_1C_3)C_5)(-((-S_2C_3C_4 - C_2S_4)C_5 + S_2S_3S_5)S_6 - (-S_2C_3S_4 + C_2C_4)C_6)lw \end{aligned} \tag{B.35}$$

$$j_{27} = 0 \tag{B.36}$$

$$j_{31} = 0 \tag{B.37}$$

$$\begin{aligned} j_{32} = -S_1(-(-(((S_1C_2C_3 + C_1S_3)C_4 - S_1S_2S_4)C_5 + (-S_1C_2S_3 + C_1C_3)S_5)S_6 \\ - ((S_1C_2C_3 + C_1S_3)S_4 + S_1S_2C_4)C_6)lw + ((S_1C_2C_3 + C_1S_3)S_4 + S_1S_2C_4)le \\ + S_1S_2ls + S_1S_2L2) - C_1(-(-(((C_1C_2C_3 - S_1S_3)C_4 - C_1S_2S_4)C_5 \\ + (-C_1C_2S_3 - S_1C_3)S_5)S_6 - ((C_1C_2C_3 - S_1S_3)S_4 + C_1S_2C_4)C_6)lw \\ + ((C_1C_2C_3 - S_1S_3)S_4 + C_1S_2C_4)le + C_1S_2ls) \end{aligned} \tag{B.38}$$

$$\begin{aligned} j_{33} = C_1S_2(-(-(((S_1C_2C_3 + C_1S_3)C_4 - S_1S_2S_4)C_5 + (-S_1C_2S_3 + C_1C_3)S_5)S_6 \\ - ((S_1C_2C_3 + C_1S_3)S_4 + S_1S_2C_4)C_6)lw + ((S_1C_2C_3 + C_1S_3)S_4 + S_1S_2C_4)le \\ + S_1S_2ls) - S_1S_2(-(-(((C_1C_2C_3 - S_1S_3)C_4 - C_1S_2S_4)C_5 + (-C_1C_2S_3 - S_1C_3)S_5)S_6 \\ - ((C_1C_2C_3 - S_1S_3)S_4 + C_1S_2C_4)C_6)lw + ((C_1C_2C_3 - S_1S_3)S_4 + C_1S_2C_4)le + C_1S_2ls) \end{aligned} \tag{B.39}$$

$$\begin{aligned} j_{34} = (-C_1C_2S_3 - S_1C_3)(-(-(((S_1C_2C_3 + C_1S_3)C_4 - S_1S_2S_4)C_5 \\ + (-S_1C_2S_3 + C_1C_3)S_5)S_6 - ((S_1C_2C_3 + C_1S_3)S_4 + S_1S_2C_4)C_6)lw \\ + ((S_1C_2C_3 + C_1S_3)S_4 + S_1S_2C_4)le \\ - (-S_1C_2S_3 + C_1C_3)(-(-(((C_1C_2C_3 - S_1S_3)C_4 - C_1S_2S_4)C_5 + (-C_1C_2S_3 - S_1C_3)S_5)S_6 \\ - ((C_1C_2C_3 - S_1S_3)S_4 + C_1S_2C_4)C_6)lw + ((C_1C_2C_3 - S_1S_3)S_4 + C_1S_2C_4)le) \end{aligned} \tag{B.40}$$

$$\begin{aligned} j_{35} = &((C_1C_2C_3 - S_1S_3)S_4 + C_1S_2C_4)(-(-(((S_1C_2C_3 + C_1S_3)C_4 - S_1S_2S_4)C_5 \\ &+ (-S_1C_2S_3 + C_1C_3)S_5)S_6 - ((S_1C_2C_3 + C_1S_3)S_4 + S_1S_2C_4)C_6)lw \\ &+ ((S_1C_2C_3 + C_1S_3)S_4 + S_1S_2C_4)le) \\ &- ((S_1C_2C_3 + C_1S_3)S_4 + S_1S_2C_4)(-(-(((C_1C_2C_3 - S_1S_3)C_4 - C_1S_2S_4)C_5 \\ &+ (-C_1C_2S_3 - S_1C_3)S_5)S_6 - ((C_1C_2C_3 - S_1S_3)S_4 + C_1S_2C_4)C_6)lw \\ &+ ((C_1C_2C_3 - S_1S_3)S_4 + C_1S_2C_4)le) \end{aligned} \tag{B.41}$$

$$\begin{aligned} j_{36} = &-(-((C_1C_2C_3 - S_1S_3)C_4 - C_1S_2S_4)S_5 \\ &+ (-C_1C_2S_3 - S_1C_3)C_5)(-(((S_1C_2C_3 + C_1S_3)C_4 - S_1S_2S_4)C_5 \\ &+ (-S_1C_2S_3 + C_1C_3)S_5)S_6 - ((S_1C_2C_3 + C_1S_3)S_4 + S_1S_2C_4)C_6)lw \\ &+ (-((S_1C_2C_3 + C_1S_3)C_4 - S_1S_2S_4)S_5 \\ &+ (-S_1C_2S_3 + C_1C_3)C_5)(-(((C_1C_2C_3 - S_1S_3)C_4 - C_1S_2S_4)C_5 \\ &+ (-C_1C_2S_3 - S_1C_3)S_5)S_6 - ((C_1C_2C_3 - S_1S_3)S_4 + C_1S_2C_4)C_6)lw \end{aligned} \tag{B.42}$$

$$j_{37} = 0 \tag{B.43}$$

$$j_{41} = 0 \tag{B.44}$$

$$j_{42} = -S_1 \tag{B.45}$$

$$j_{43} = C_1S_2 \tag{B.46}$$

$$j_{44} = -C_1C_2S_3 - S_1C_3 \tag{B.47}$$

$$j_{45} = (C_1C_2C_3 - S_1S_3)\,S_4 + C_1S_2C_4 \tag{B.48}$$

$$j_{46} = ((-C_4C_2C_3 + S_4S_2)\,S_5 - C_5C_2S_3)\,C_1 + S_1\,(-C_3C_5 + S_5C_4S_3) \tag{B.49}$$

$$\begin{aligned} j_{47} = &((C_5C_4C_2C_3 - C_5S_4S_2 - S_5C_2S_3)\,S_6 + C_6\,(S_2C_4 + C_3C_2S_4))\,C_1 \\ &- S_1\,((C_3S_5 + C_5C_4S_3)\,S_6 + C_6S_4S_3) \end{aligned} \tag{B.50}$$

$$j_{51} = 0 \tag{B.51}$$

$$j_{52} = C_1 \tag{B.52}$$

$$j_{53} = S_1S_2 \tag{B.53}$$

$$j_{54} = -S_1 C_2 S_3 + C_1 C_3 \tag{B.54}$$

$$j_{55} = (S_1 C_2 C_3 + C_1 S_3)\, S_4 + S_1 S_2 C_4 \tag{B.55}$$

$$j_{56} = ((-C_4 C_2 C_3 + S_4 S_2)\, S_5 - C_5 C_2 S_3)\, S_1 - C_1\, (-C_3 C_5 + S_5 C_4 S_3) \tag{B.56}$$

$$\begin{aligned} j_{57} = {} & ((C_5 C_4 C_2 C_3 - C_5 S_4 S_2 - S_5 C_2 S_3)\, S_6 + C_6\, (S_2 C_4 + C_3 C_2 S_4))\, S_1 \\ & + C_1\, ((C_3 S_5 + C_5 C_4 S_3)\, S_6 + C_6 S_4 S_3) \end{aligned} \tag{B.57}$$

$$j_{61} = 1 \tag{B.58}$$

$$j_{62} = 0 \tag{B.59}$$

$$j_{63} = C_2 \tag{B.60}$$

$$j_{64} = S_2 S_3 \tag{B.61}$$

$$j_{65} = -S_2 C_3 S_4 + C_2 C_4 \tag{B.62}$$

$$j_{66} = (S_2 C_3 C_4 + C_2 S_4)\, S_5 + S_2 S_3 C_5 \tag{B.63}$$

$$j_{67} = ((-C_4 C_5 C_3 + S_5 S_3)\, S_6 - C_6 C_3 S_4)\, S_2 - C_2\, (C_5 S_4 S_6 - C_6 C_4) \tag{B.64}$$

B.2 Cameras' Calibrations

B.2.1 Pinhole camera model

In Figure B.2, the center of the lens is assumed as a pinhole. The light rayed from a 3D point $(X_C, Y_C, Z_C)^T$ enters the camera through this pinhole and projects on the real image plane in the inverse position $(u', v')^T$. In order to avoid image inversion, it is assumed that there is a visual image plane in front of the center of the projection and $(u, v)^T$ is the corresponding 2D image coordinate of $(X_C, Y_C, Z_C)^T$ in the visual image plane. Notice that Σ_C is set on the origin of the lens and the Z-axis of Σ_C which is perpendicular to the image plane is called the optical axis.

The pinhole camera model defines a geometric relationship as shown in (B.65) between the 3D point $(X_C, Y_C, Z_C)^T$ and its 2D corresponding projection $(u, v)^T$ on the image plane. This geometric mapping from 3D to 2D is called *perspective projection*.

$$(u, v) \;=\; \left(f\frac{X_C}{Z_C}, f\frac{Y_C}{Z_C} \right). \tag{B.65}$$

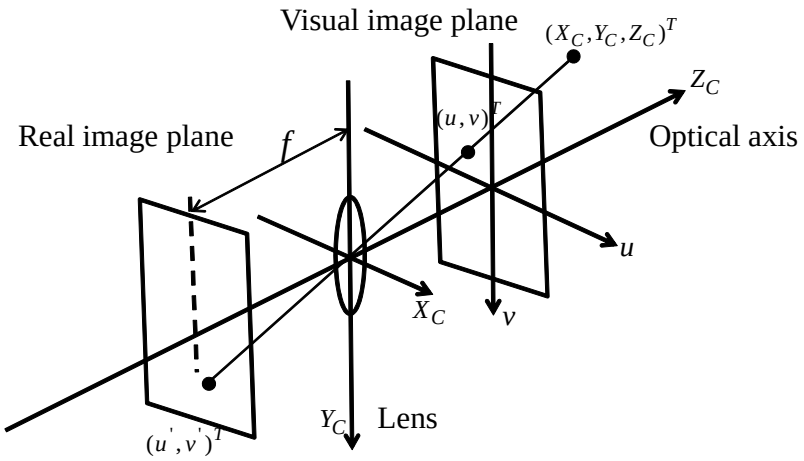


Figure B.2: Pinhole camera model

B.2.2 Calibration of high speed camera

Suppose that a 3D position is expressed as ${}^{B}\boldsymbol{P}$ and ${}^{C_n}\boldsymbol{P}$ in the base coordinate system Σ_B and in the camera coordinate system $\Sigma_{C_n}(n = 1, 2)$, respectively. Then it is well known that

$$ {}^{C_n}\boldsymbol{P} \;=\; {}^{C_n}\boldsymbol{P}_B + {}^{C_n}\boldsymbol{R}_B {}^{B}\boldsymbol{P}, \tag{B.66} $$

where ${}^{B}\boldsymbol{P} = (X_B, Y_B, Z_B)^T$, ${}^{C_n}\boldsymbol{P} = (X_{C_n}, Y_{C_n}, Z_{C_n})^T$.

In (B.66), ${}^{C_n}\boldsymbol{P}_B \in \mathbb{R}^3$ describes the origin of Σ_B in Σ_{C_n} and ${}^{C_n}\boldsymbol{R}_B \in \mathbb{R}^{3\times3}$ is a rotation matrix of Σ_B with respect to Σ_{C_n}. These two vectors are obtained by the camera calibration, which includes the following steps:

Firstly, define ${}^{C_n}\boldsymbol{P}_B$ and ${}^{C_n}\boldsymbol{R}_B$ as

$$ {}^{C_n}\boldsymbol{P}_B = \begin{bmatrix} X_{BC_n} \\ Y_{BC_n} \\ Z_{BC_n} \end{bmatrix}, \qquad {}^{C_n}\boldsymbol{R}_B = \begin{bmatrix} r_{11}^n & r_{12}^n & r_{13}^n \\ r_{21}^n & r_{22}^n & r_{23}^n \\ r_{31}^n & r_{32}^n & r_{33}^n \end{bmatrix}, \tag{B.67} $$

where $r_{ij}(i, j = 1 \sim 3)$ corresponds to the element in $\boldsymbol{R}$ derived from XYZ Euler angle with respect to $\theta_z, \theta_y, \theta_x$.

$$ \boldsymbol{R} = \begin{bmatrix} \cos\theta_z \cos\theta_y & \cos\theta_z \sin\theta_y \sin\theta_x - \sin\theta_z \cos\theta_x & \cos\theta_z \sin\theta_y \cos\theta_x + \sin\theta_z \sin\theta_x \\ \sin\theta_z \cos\theta_y & \sin\theta_z \sin\theta_y \sin\theta_x + \cos\theta_z \cos\theta_x & \sin\theta_z \sin\theta_y \cos\theta_x - \cos\theta_z \sin\theta_x \\ -\sin\theta_y & \cos\theta_y \sin\theta_x & \cos\theta_y \cos\theta_x \end{bmatrix}. \tag{B.68} $$

Secondly, substituting (B.67) into (B.66) and (B.65) in turn leads to (B.69) and (B.70).

$$ \begin{aligned} X_{C_n} &= r_{11}^n X_B + r_{12}^n Y_B + r_{13}^n Z_B + X_{BC_n}, \\ Y_{C_n} &= r_{21}^n X_B + r_{22}^n Y_B + r_{23}^n Z_B + Y_{BC_n}, \\ Z_{C_n} &= r_{31}^n X_B + r_{32}^n Y_B + r_{33}^n Z_B + Z_{BC_n}, \end{aligned} \tag{B.69} $$

$$ \begin{aligned} u_n &= f \frac{r_{11}^n X_B + r_{12}^n Y_B + r_{13}^n Z_B + X_{BC_n}}{r_{31}^n X_B + r_{32}^n Y_B + r_{33}^n Z_B + Z_{BC_n}}, \\ v_n &= f \frac{r_{21}^n X_B + r_{22}^n Y_B + r_{23}^n Z_B + Y_{BC_n}}{r_{31}^n X_B + r_{32}^n Y_B + r_{33}^n Z_B + Z_{BC_n}}. \end{aligned} \tag{B.70} $$

Thirdly, choose m points and measure the 3D coordinates of them in Σ_B. Then, calculate their 2D coordinates by using two methods: 1) the method shown in (B.70), obtaining (u_n, v_n) from their 3D coordinates; 2) the method of obtaining $(\tilde{u}_n, \tilde{v}_n)$ from their 2D coordinates on the image.

Finally, estimate $X_{BC_n}, Y_{BC_n}, Z_{BC_n}, \theta_z, \theta_y, \theta_x$ by minimizing the following objective function,

$$J = \sum_{j=1}^{m}((\tilde{u}_{nj} - u_{nj})^2 + (\tilde{v}_{nj} - v_{nj})^2). \tag{B.71}$$

B.2.3 Calculation of the 3D coordinate from 2D image coordinates

After obtaining ${}^{C_1}\boldsymbol{P}_B$, ${}^{C_1}\boldsymbol{R}_B$, ${}^{C_2}\boldsymbol{P}_B$, ${}^{C_2}\boldsymbol{R}_B$ through the camera calibration in the above section, the 3D coordinate ${}^{B}\boldsymbol{P} = (X_B, Y_B, Z_B)^T$ can be solved by using its 2D coordinates (u_1, v_1) and (u_2, v_2) in the two cameras' images from (B.70).

In fact, (B.70) is equal to the following equation:

$$\begin{aligned}
\frac{u_1}{f}(r^1_{31}X_B + r^1_{32}Y_B + r^1_{33}Z_B + Z_{BC_1}) &= r^1_{11}X_B + r^1_{12}Y_B + r^1_{13}Z_B + X_{BC_1},\\
\frac{v_1}{f}(r^1_{31}X_B + r^1_{32}Y_B + r^1_{33}Z_B + Z_{BC_1}) &= r^1_{21}X_B + r^1_{22}Y_B + r^1_{23}Z_B + Y_{BC_1},\\
\frac{u_2}{f}(r^2_{31}X_B + r^2_{32}Y_B + r^2_{33}Z_B + Z_{BC_2}) &= r^2_{11}X_B + r^2_{12}Y_B + r^2_{13}Z_B + X_{BC_2},\\
\frac{v_2}{f}(r^2_{31}X_B + r^2_{32}Y_B + r^2_{33}Z_B + Z_{BC_2}) &= r^2_{21}X_B + r^2_{22}Y_B + r^2_{23}Z_B + Y_{BC_2},
\end{aligned} \tag{B.72}$$

which can be rewritten as

$$\begin{bmatrix}
\frac{u_1}{f}r^1_{31} - r^1_{11} & \frac{u_1}{f}r^1_{32} - r^1_{12} & \frac{u_1}{f}r^1_{33} - r^1_{13}\\
\frac{v_1}{f}r^1_{31} - r^1_{21} & \frac{v_1}{f}r^1_{32} - r^1_{22} & \frac{v_1}{f}r^1_{33} - r^1_{23}\\
\frac{u_2}{f}r^2_{31} - r^2_{11} & \frac{u_2}{f}r^2_{32} - r^2_{12} & \frac{u_2}{f}r^2_{33} - r^2_{13}\\
\frac{v_2}{f}r^2_{31} - r^2_{21} & \frac{v_2}{f}r^2_{32} - r^2_{22} & \frac{v_2}{f}r^2_{33} - r^2_{23}
\end{bmatrix}
\begin{bmatrix} X_B \\ Y_B \\ Z_B \end{bmatrix} =
\begin{bmatrix}
X_{BC_1} - \frac{u_1}{f}Z_{BC_1}\\
Y_{BC_1} - \frac{v_1}{f}Z_{BC_1}\\
X_{BC_2} - \frac{u_2}{f}Z_{BC_2}\\
Y_{BC_2} - \frac{v_2}{f}Z_{BC_2}
\end{bmatrix}. \tag{B.73}$$

Then,

$$\begin{bmatrix} X_B \\ Y_B \\ Z_B \end{bmatrix} = A^{\dagger}B \tag{B.74}$$

in which

$$A = \begin{bmatrix}
\frac{u_1}{f}r^1_{31} - r^1_{11} & \frac{u_1}{f}r^1_{32} - r^1_{12} & \frac{u_1}{f}r^1_{33} - r^1_{13}\\
\frac{v_1}{f}r^1_{31} - r^1_{21} & \frac{v_1}{f}r^1_{32} - r^1_{22} & \frac{v_1}{f}r^1_{33} - r^1_{23}\\
\frac{u_2}{f}r^2_{31} - r^2_{11} & \frac{u_2}{f}r^2_{32} - r^2_{12} & \frac{u_2}{f}r^2_{33} - r^2_{13}\\
\frac{v_2}{f}r^2_{31} - r^2_{21} & \frac{v_2}{f}r^2_{32} - r^2_{22} & \frac{v_2}{f}r^2_{33} - r^2_{23}
\end{bmatrix}, \qquad B = \begin{bmatrix}
X_{BC_1} - \frac{u_1}{f}Z_{BC_1}\\
Y_{BC_1} - \frac{v_1}{f}Z_{BC_1}\\
X_{BC_2} - \frac{u_2}{f}Z_{BC_2}\\
Y_{BC_2} - \frac{v_2}{f}Z_{BC_2}
\end{bmatrix} \tag{B.75}$$

and A is obtained by

$$\boldsymbol{A}^{\dagger} = (\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T. \tag{B.76}$$

B.2.4 Calibration of middle speed camera

This section is for the calibration of the middle speed cameras. There is a difference between the calibrations of the high speed and the middle speed cameras that for the previous one, the rotational matrix from Σ_B to Σ_{C_n} used is nonlinear, but it is linear for the later one.

The calibration of the middle speed cameras includes the following steps:

Firstly, the basic transformation relation in (B.66) is also held here that

$$ {}^{C_n}\boldsymbol{P} \;=\; {}^{C_n}\boldsymbol{P}_B + {}^{C_n}\boldsymbol{R}_B{}^{B}\boldsymbol{P}, \tag{B.77} $$

where ${}^{B}\boldsymbol{P} = (X_B, Y_B, Z_B)^T$, ${}^{C_n}\boldsymbol{P} = (X_{C_n}, Y_{C_n}, Z_{C_n})^T$. ${}^{C_n}\boldsymbol{P}_B \in \mathbb{R}^3$ describes the origin of Σ_B in Σ_{C_n} and ${}^{C_n}\boldsymbol{R}_B \in \mathbb{R}^{3\times3}$ is a rotation matrix of Σ_B with respect to Σ_{C_n}. They are defined as

$$ {}^{C_n}\boldsymbol{P}_B = \begin{bmatrix} X_{BC_n} \\ Y_{BC_n} \\ Z_{BC_n} \end{bmatrix}, \qquad {}^{C_n}\boldsymbol{R}_B = \begin{bmatrix} r_{11}^n & r_{12}^n & r_{13}^n \\ r_{21}^n & r_{22}^n & r_{23}^n \\ r_{31}^n & r_{32}^n & r_{33}^n \end{bmatrix}. \tag{B.78} $$

Secondly, substituting (B.78) into (B.77) and (B.65) in turn leads to (B.79) and (B.80).

$$ \begin{aligned} X_{C_n} &= r_{11}^n X_B + r_{12}^n Y_B + r_{13}^n Z_B + X_{BC_n}, \\ Y_{C_n} &= r_{21}^n X_B + r_{22}^n Y_B + r_{23}^n Z_B + Y_{BC_n}, \\ Z_{C_n} &= r_{31}^n X_B + r_{32}^n Y_B + r_{33}^n Z_B + Z_{BC_n}, \end{aligned} \tag{B.79} $$

$$ \begin{aligned} u_n &= f\frac{r_{11}^n X_B + r_{12}^n Y_B + r_{13}^n Z_B + X_{BC_n}}{r_{31}^n X_B + r_{32}^n Y_B + r_{33}^n Z_B + Z_{BC_n}}, \\ v_n &= f\frac{r_{21}^n X_B + r_{22}^n Y_B + r_{23}^n Z_B + Y_{BC_n}}{r_{31}^n X_B + r_{32}^n Y_B + r_{33}^n Z_B + Z_{BC_n}}, \end{aligned} \tag{B.80} $$

which is expressed as

$$ \begin{aligned} u_n &= \frac{b_1^n X_B + b_2^n Y_B + b_3^n Z_B + b_4^n}{b_9^n X_B + b_{10}^n Y_B + b_{11}^n Z_B + 1}, \\ v_n &= \frac{b_5^n X_B + b_6^n Y_B + b_7^n Z_B + b_8^n}{b_9^n X_B + b_{10}^n Y_B + b_{11}^n Z_B + 1} \end{aligned} \tag{B.81} $$

with $b_i^n (i = 1, ..., 11)$. Then, rewrite (B.81) as

$$ \boldsymbol{A}^n \boldsymbol{b}^n = \boldsymbol{c}^n, \tag{B.82} $$

where

$$ \boldsymbol{A}^n = \begin{bmatrix} X_B & Y_B & Z_B & 1 & 0 & 0 & 0 & 0 & -u_n X_B & -u_n Y_B & -u_n Z_B \\ 0 & 0 & 0 & 0 & X_B & Y_B & Z_B & 1 & -v_n X_B & -v_n Y_B & -v_n Z_B \end{bmatrix}, \tag{B.83} $$

$$ \boldsymbol{b}^n = \begin{bmatrix} b_1^n & b_2^n & b_3^n & b_4^n & b_5^n & b_6^n & b_7^n & b_8^n & b_9^n & b_{10}^n & b_{11}^n \end{bmatrix}^T, \tag{B.84} $$

$$ \boldsymbol{c}^n = [u_n, v_n]^T. \tag{B.85} $$

Finally, solve $\boldsymbol{b}^n$ by using the 3D coordinates of m points and their corresponding 2D coordinates on the image with (B.86),

$$ \boldsymbol{A}_j^n \boldsymbol{b}^n = \boldsymbol{c}_j^n (j = 1, \cdots, m). \tag{B.86} $$

Actually, it is seen easily that if ${}^{B}\boldsymbol{P}, u_n, v_n$ are obtained, $\boldsymbol{A}^n, \boldsymbol{c}^n$ are known matrix and vector. Hence, $\boldsymbol{b}^n$ can be solved by using (B.87).

$$\begin{bmatrix} \boldsymbol{A}_1^n \\ \vdots \\ \boldsymbol{A}_m^n \end{bmatrix} \boldsymbol{b}^n = \begin{bmatrix} \boldsymbol{c}_1^n \\ \vdots \\ \boldsymbol{c}_m^n \end{bmatrix} \tag{B.87}$$

Bibliography

[1] R. L. Andersson: *A Robot Ping-Pong Player: Experiment in Real-Time Intelligent Control*, Chapter 1, Cambridge, MA:MIT Press, 1987.

[2] R. L. Andersson: Understanding and applying a robot ping-pong player's expert controller, *1989 IEEE International Conference on Robotics and Automation*, pp. 1284–1289, 1989.

[3] R. L. Andersson: Aggressive trajectory generator for a robot ping-pong player, *IEEE Control Systems Magazine*, Vol. 9, No. 2, pp. 15–21, 1989.

[4] R. L. Andersson: A low latency 60Hz stereo system for real-time visual control, *Proceedings of 5th IEEE International Symposium on Intelligent Control*, pp. 165–170, 1990.

[5] U. M. Ascher, R. M. M. Mattheij, and R. D. Russell: *Numerical Solution of Boundary Value Problems for Ordinary Differential Equations*, SIAM, 1995.

[6] L. Acosta, J. J. Rodrigo, J. A. Mendez, G. N. Marichal, and M. Sigut: Ping-pong player prototype, *IEEE Robotics Automation Magazine*, Vol. 10, No. 4, pp. 44–52, 2003.

[7] F. Alam, A. Subic, J. Naser, M. G. Rasul, and M. M. K. Khan, A study of spin effects on tennis ball aerodynamics, *Wseas Transactions on Fluid Mechanics*, Vol. 3, No. 3, pp. 271–278, 2008.

[8] S. R. Bernfeld, V. Lakshmikantham: *An Introduction to Nonlinear Boundary Value Problems*, Chapter 1, Academic Press,Inc., 1974.

[9] S. C. Cain, M. M. Hayat, and E. E. Armstrong: Projection-based image registration in the presence of fixed-pattern noise, *IEEE Transactions on Image Processing*, Vol. 10, No. 12, pp. 1860–1872, 2001.

[10] K. Chen, X. Yang, L. Liu, and D. Yang: *Robot Technology and Application*, Qinghua University Press, 2006, in Chinese.

[11] H. Fassler, H. A. Vasteras, and J. W. Zurich: A robot ping pong player: optimized mechanics, high performance 3D vision, and intelligent sensor control, *Robotersysteme*, Vol. 6, pp. 161–170, 1990.

[12] M. M. Fatch, H. Farhangfard: On the transforming of control space by manipulator jacobian, *International Journal of Control, Automation, and Systems*, Vol. 6, No. 1, pp. 101–108, 2008.

[13] L. Greengard: Spectral integration and two-point boundary value problems, *SIAM Journal on Numerical Analysis*, Vol. 28, No. 4, pp. 1071–1080, 1991.

[14] J. Hartley: Toshiba progresses towards sensory control in real-time, *Industrial Robot*, Vol. 14, No. 1, pp. 50–52, 1987.

[15] H. Hashimoto, F. Ozaki, K. Asano, and K. Osuka: Development of a pingpong robot system using 7 degrees of freedom direct drive arm, *1987 Int. Conf. on Industrial Electronics, Control, and Instrumentation*, pp. 608–615, 1987.

[16] T. Han: The dynamic characteristic of table tennis: relative principle between spin and speed, *Sports Science*, Vol. 14, No. 6, pp. 48–55, 1994, in Chinese.

[17] J. Knight, D. Lowery: Pingpong-playing robot controlled by a microcomputer, *Microprocessors and Microsystems*, Vol. 10, No. 6, pp. 332–335, 1986.

[18] S. Kusubori, K. Yoshida, and H. Sekiya: The functions of spin on shot trajectory in table tennis, *The 30th Annual Conference of Biomechanics in Sports*, pp. 245–248, 2012.

[19] B. D. Lucas, T. Kanade: An iterative image registration technique with an application to stereo vision, *Proceedings of Imaging Understanding Workshop*, pp. 121–130, 1981.

[20] X. Liu, Z. Xie, and Q. Zhang: A fixed-time trapezoidal acceleration trajectory planning method for ping-pang playing robots, *Machinery & Electronics*, No. 9, pp. 55–58, 2011, in Chinese.

[21] C. Liu, Y. Hayakawa, and A. Nakashima: On a modified method of measuring the rotational speed of a table tennis ball, *Proc. SICE 11th Control Division Conf.*, 18321, 2011.

[22] C. Liu, Y. Hayakawa, and A. Nakashima: A registration algorithm for on-line measuring the rotational velocity of a table tennis ball, *Proc. of the 2011 IEEE International Conference on Robotics and Biomimetics*, pp. 2270–2275, 2011.

[23] C. Liu, Y. Hayakawa, and A. Nakashima: An on-line algorithm for measuring the translational and rotational velocities of a table tennis ball, *SICE Journal of Control, Measurement, and System Integration*, Vol. 5, No. 4, pp. 233–241, 2012.

[24] C. Liu, Y. Hayakawa, and A. Nakashima: Racket control for robot playing table tennis ball, *International Conference on Control, Automation and Systems*, pp. 1427–1432, 2012.

[25] C. Liu, Y. Hayakawa, and A. Nakashima: Racket control and its experiments for robot playing table tennis ball, *Proc. of the 2012 IEEE International Conference on Robotics and Biomimetics*, pp. 241–246, 2012.

[26] C. Liu, Y. Hayakawa, and A. Nakashima: Racket control for a table tennis robot to return a ball, *SICE Journal of Control, Measurement, and System Integration*, Vol. 6, No. 4, 2013.

[27] F. Miyazaki, M. Takeuchi, M. Matsushima, T. Kusano, and T. Hashimoto: Realization of the table tennis task based on virtual targets, *Proceedings of the 2002 IEEE Int. Conference on Robotics & Automation*, pp. 3844–3849, 2002.

[28] F. Miyazaki, M. Matsushima, and M. Takeuchi: Learning to dynamically manipulate: a table tennis robot controls a ball and rallies with a human being, *Advances in Robot Control*, Springer, pp. 317–341, 2005.

[29] K. P. Modi: *Visual application of human robot interaction: development of a ping pong playing robotic arm* , M.S.dissertation, Rochester Institute of Technology, Rochester, New York, 2005.

[30] M. Matsushima, T. Hashimoto, M. Takeuchi, and F. Miyazaki: A learning approach to robotic table tennis, *IEEE Transactions on Robotics*, Vol. 21, No. 4, pp. 767–771, 2005.

[31] K. Mülling, J. Peters: A computational model of human table tennis for robot application, *Proc. of the 2009 Autonome Mobile System*, pp. 57–64, 2009.

[32] K. Mülling, J. Kober, and J. Peters: A biomimetic approach to robot table tennis, *Proceedings of the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems(IROS 2010)*, pp. 1921–1926, 2010.

[33] Y. Nakabo, I. Ishii, and M. Ishikawa: 3D tracking using two high-speed vision systems, *Proceedings of the 2002 IEEE/RSJ Int. Conference on Intelligent Robots and Systems*, pp. 360–365, 2002.

[34] A. Nakashima, Y. Kobayashi, Y. Ogawa, and Y. Hayakawa: Modeling of rebound phenomenon between ball and racket rubber with spinning effect, *ICROS-SICE International Joint Conference 2009*, pp. 2295–2300, 2009.

[35] A. Nakashima, Y. Tsuda, C. Liu, and Y. Hayakawa: A real-time measuring method of translational/rotational velocities of a flying ball, *2010 IFAC Symposium on Mechatronic Systems*, pp. 732–738, 2010.

[36] A. Nakashima, Y. Ogawa, Y. Kobayashi, and Y. Hayakawa: Modeling of rebound phenomenon of a rigid ball with friction and elastic effects, *Proc. of IEEE American Control Conference*, pp. 1410–1415, 2010.

[37] J. Nonomura, A. Nakashima, and Y. Hayakawa: Analysis of effects of rebounds and aerodynamics for trajectory of table tennis ball, *Proc. of SICE Annual Conference*, pp. 1567–1572, 2010.

[38] A. Nakashima, Y. Ogawa, C. Liu, and Y. Hayakawa: Robotic table tennis based on physical models of aerodynamics and rebounds, *Proc. of the 2011 IEEE International Conference on Robotics and Biomimetics*, pp. 2348–2354, 2011.

[39] A. Nakashima, J. Nonomura, C. Liu, and Y. Hayakawa: Hitting back-spin balls by robotic table tennis system based on physical models of ball motion, *10th International IFAC Symposium on Robot Control*, pp. 834–841, 2012.

[40] J. Nonomura: *Realization of a table tennis playing robot based on the physical models - a case of back spin balls*, M.S.dissertation, Department of Mechanical Science and Engineering, Nagoya University, Japan, 2012.

[41] S. Rusdorf, G. Brunnett: Real time tracking of high speed movements in the context of a table tennis application, *Proceedings of the ACM Symposium on Virtual Reality Software and Technology*, pp. 192–200, 2005.

[42] S. Rusdorf, G. Brunnett, M. Lorenz, and T. Winkler: Real-time interaction with a humanoid avatar in an immersive table tennis simulation, *IEEE Transactions on Visualization and Computer Graphics*, Vol. 13, No. 1, pp. 15–25, 2007.

[43] H. Seraji, B. Bon: Real-time collision avoidance for dexterous 7-DOF arms, *http://trs-new.jpl.nasa.gov/dspace/ bitstream/2014/22335/1/97-0812.pdf*.

[44] H. Shum, T. Komura: Tracking the translational and rotational movement of the ball using high-speed camera movies, *2005 IEEE Int. Conf. Image Processing*, pp. 1084–1087, 2005.

[45] H. Tang, M. Mizoguchi, and S. Toyoshima: Speed and spin characteristics of the 40mm table tennis ball, *International Journal of Table Tennis Sciences*, No. 4&5, pp. 278–284, 2002.

[46] T. Tamaki, T. Sugino, and M. Yamamoto: Measuring ball spin by image registration, *Proc.10th Frontiers of Computer Vision*, pp. 269–274, 2004.

[47] A. Traslosheros, J. M. Sebastián and L. Ángel, et.al.: Visual servoing for the robotenis system: a strategy for a 3 DOF parallel robot to hit a ping-pong ball, *The 50th Conference on Decision and Control and European Control Conference*, pp. 5695–5701, 2011.

[48] Vision-based robot/human ping-pong, www.quanta-view.com.

[49] H. Wu, Z. Qin, S. Xu, and E. Xi: Experimental research in table tennis spin, *International Journal of Table Tennis Sciences*, No. 1, pp. 73–78, 1992.

[50] J. Wang, T. Liu, and X. Wang: Research work and the development of rotational theory of table tennis in China, *International Journal of Table Tennis Sciences*, No. 3, 1996.

[51] Y. Watanabe, T. Komuro, S. Kagami, and M. Ishikawa: Multi-target tracking using a vision chip and its applications to real-time visual measurement, *Journal of Robotics and Mechatronnics*, Vol. 17, No. 2, pp. 121–129, 2005.

[52] Y. Wang: *Visual C++ Object Oriented Programming*, Qinghua University Press, 2006, in Chinese.

[53] J. Xiang, C. Zhong, and W. Wei: General-weighted least-norm control for redundant manipulators, *IEEE Transactions on Robotics*, Vol. 26, No. 4, pp. 660–669, 2010.

[54] P. Yang, D. Xu, H. Wang, and J. Zhang: Design and motion control of a ping pong robot, *Proc. of the 8th World Congress on Intelligent Control and Automation*, pp. 102–107, 2010.

[55] P. Yang, D. Xu, H. Wang, and Z. Zhang: Control system design for a 5-DOF table tennis robot, *The 11th Int. Conf. Control, Automation, Robotics and Vision*, pp. 1731–1735, 2010.

[56] G. Yuan, S. Lu, and Z. Wei: A line search algorithm for unconstrained optimization. *J. Software Eng. Appl.*, Vol. 3, No. 5, pp. 503–509, 2010.

[57] B. Zitova, J. Flusser: Image registration methods: a survey, *Image and Vision Computing*, Vol. 21, No. 11, pp. 977–1000, 2003.

[58] Y. Zhang: *Ping-pong robot calibration and trajectory tracking based on real-time vision*, Ph.D.dissertation, Zhe Jiang University, Zhe Jiang, China, 2009, in Chinese.

[59] Z. Zhang, D. XU: High-speed vision system based on smart camera and its target tracking algorithm, *Robot*, Vol. 31, No. 3, pp. 229–234, 2009, in Chinese.

[60] Z. Zhang: *Visual measurement and control for table tennis robot*, Ph.D.dissertation, Institude of Automation, Chinese Academic of Sciences, Pei Jing, China, 2010, in Chinese.

[61] G. Zhang, B. Li, H. Zheng, and H. Gong: Trajectory planning for a robotic ping pong player, Chinese Journal of Scientific Instrument, Vol. 32, No. 6(S1), pp. 87–91, 2011, in Chinese.

Acknowledgements

This thesis is submitted for Doctor degree in Mechanical Science and Engineering at Nagoya University, Japan. Dr. Yoshikazu HAYAKAWA, Professor of Nagoya University, was my supervisor. The research of this thesis was accomplished under my supervisor and many individuals' help. Hence, I would like to express my gratitude to all those who have helped me during the study and during the writing of this thesis.

First and foremost, I would like to show my deepest gratitude to my supervisor, Prof. Yoshikazu HAYAKAWA, who gave me valuable guidance in every stage of the research. Without his enlighten instructions and great encouragement, the completion of this research and the thesis would be impossible. Prof. HAYAKAWA is a respectable and responsible teacher. I do appreciate his patience and rigorous in the academic study, which influence my future study.

Secondly, I am extremely grateful to Dr. Yoji UNO, Professor of Nagoya University, Dr. Tatsuya SUZUKI, Professor of Nagoya University, and Dr. Takeshi FURUHASHI, Professor of Nagoya University for reviewing the thesis. Their valuable recommends improve me greatly not only on the research of the thesis but also on the writing skills.

Thirdly, on the completion of the thesis, I wish to thank Dr. Akira NAKASHIMA, assistant professor of Nagoya University, who gave me considerable help by means of suggestion, comments and criticism. My sincere thanks are also given to Dr. Kenji FUJIMOTO, Professor of Kyoto University. I benefited greatly from his valuable suggestions in my presentation.

Fourthly, I feel grateful to all the students in the laboratory, especially, the table tennis group, who helped me much in the experiments of the thesis and Dr. Hao, whose good advice in mathematics always helped me solving research problems.

Fifthly, I wish to thank China Scholarship Council, who provided me scholarships for the daily life. It made me concentrate on the study and had an easy life in Japan.

Last but not least, I would like to thank my parents and my boy friend, who constantly encouraged me when I felt frustrated in the three years' doctor research.

March 2013,

Chunfang Liu.

Published Papers

The following list is about the published journal and conference papers at the doctor stage.

Chapter 2

[35] A. Nakashima, Y. Tsuda, C. Liu and Y. Hayakawa: A real-time measuring method of translational/rotational velocities of a flying ball, *2010 IFAC Symposium on Mechatronic Systems*, pp. 732–738, 2010.

[21] C. Liu, Y. Hayakawa, A. Nakashima: On a modified method of measuring the rotational speed of a table tennis ball, *Proc. SICE 11th Control Division Conf.*, 18321, 2011.

[22] C. Liu, Y. Hayakawa, A. Nakashima: A registration algorithm for on-line measuring the rotational velocity of a table tennis ball, *Proc. of the 2011 IEEE International Conference on Robotics and Biomimetics*, pp. 2270–2275, 2011.

[23] C. Liu, Y. Hayakawa, A. Nakashima: An on-line algorithm for measuring the translational and rotational velocities of a table tennis ball, *SICE Journal of Control, Measurement, and System Integration*, Vol. 5, No. 4, pp. 233–241, 2012.

Chapter 3

[38] A. Nakashima, Y. Ogawa, C. Liu, and Y. Hayakawa: Robotic table tennis based on physical models of aerodynamics and rebounds, *Proc. of the 2011 IEEE International Conference on Robotics and Biomimetics*, pp. 2348–2354, 2011.

[39] A. Nakashima, J. Nonomura, C. Liu, and Y. Hayakawa: Hitting back-spin balls by robotic table tennis system based on physical models of ball motion, *10th International IFAC Symposium on Robot Control*, pp. 834–841, 2012.

Chapter 4

[24] C. Liu, Y. Hayakawa, A. Nakashima: Racket control for robot playing table tennis ball, *International Conference on Control, Automation and Systems*, pp. 1427–1432, 2012.

[25] C. Liu, Y. Hayakawa, A. Nakashima: Racket control and its experiments for robot playing table tennis ball, *Proc. of the 2012 IEEE International Conference on Robotics and Biomimetics*, pp. 241–246, 2012.

[26] C. Liu, Y. Hayakawa, A. Nakashima: Racket control for a table tennis robot to return a ball, *SICE Journal of Control, Measurement, and System Integration*, Vol. 6, No. 4, 2013.