

A Biomimetic Approach to Robot Table Tennis

Katharina Mülling^{1,2}, Jens Kober^{1,2}, Jan Peters^{1,2}
email: *{muelling,kober,jan.peters}@tuebingen.mpg.de*

All authors are affiliated both with:

(1) Max Planck Institute for Intelligent Systems
Spemanstr. 38, 72076 Tübingen, Germany

and

(2) Technische Universität Darmstadt
FG Intelligente Autonome Systeme
Hochschulstr. 10, 64289 Darmstadt, Germany

July 1, 2011

Abstract

Playing table tennis is a difficult motor task that requires fast movements, accurate control and adaptation to task parameters. Although human beings see and move slower than most robot systems, they significantly outperform all table tennis robots. One important reason for this higher performance is the human movement generation. In this paper, we study human movements during table tennis and present a robot system that mimics human striking behavior. Our focus lies on generating hitting motions capable of adapting to variations in environmental conditions, such as changes in ball speed and position. Therefore, we model the human movements involved in hitting a table tennis ball using discrete movement stages and the virtual hitting point hypothesis. The resulting model was evaluated both in a physically realistic simulation and on a real anthropomorphic seven degrees of freedom Barrett WAMTM robot arm.

keywords biomimetic robotics, motor control, biologically inspired systems

1 INTRODUCTION

Humans perform complex tasks relying on little feedback with long latencies and have strong limits on their movement execution. Additionally, they suffer from inaccurate sensory information in largely unmodeled environments. Interception tasks, as table tennis, require fast and accurate movements that are precisely coordinated with the visual information of the ball and the opponent. However, the human central nervous system has little time to process feedback about the environment and has to rely largely on feed-forward components (Wolpert et al., 1998) such as accurate task models as well as predictions about the opponent and the ball. Nevertheless, human beings outperform table tennis and baseball robots, which are tailored for these tasks. This performance gap is to a significant part due to the way in which humans perform robust movements.

Computational models of motor control and learning that describe human motion generation can be useful for neuroscientists to verify hypotheses on human motor control. These models can also be useful in robotics to create robot systems that are able to perform a wide variety of movements robustly and adapt these movements to unexpected environmental conditions and new requirements.

Table tennis has long fascinated roboticists as a particularly difficult task. Significant research on robot table tennis started around 1983 (Billingsley), and was followed by a period of enthusiastic research until 1993 (Andersson, 1988; Knight & Lowery, 1986; Hartley, 1987; Hashimoto et al., 1987; Fässler et al., 1990). A few groups continue to work on this problem (Miyazaki et al., 2005; Matsushima et al., 2005; Acosta et al., 2003), as detailed in Section 1.1. These early approaches used hardware engineering solutions to avoid inherent problems in high speed anthropomorphic movement generation and vision in a human inhabited environment.

In contrast to previous approaches, we use an anthropomorphic robot arm with seven degrees of freedom (DoFs) and concentrate on generating smooth movements that properly distribute the forces over the different DoFs. To cope with the resulting challenges of this approach, we present a biomimetic approach for trajectory generation and movement adaptation based on theories pertaining to human motor control in table tennis. Our goal¹ is to generate human-like striking movements on an anthropomorphic robot arm with seven DoFs. We investigate the problem of determining joint angles and joint velocities for a redundant robot arm at the interception point. Furthermore, we study the planing of arm trajectories for returning an incoming ball towards a desired point on the opponent's court. The resulting trajectories can be adapted to new environmental conditions. The presented robot table tennis player is able to return balls served by a ball cannon on an International Table Tennis Federation (ITTF) standard sized table. The system works well both in simulation and on a real Barrett WAMTM.

1.1 Related Work

Research on robot table tennis started with robot ping pong competitions initiated by Billingsley (1983). Billingsley developed a special set of rules for the competition. In contrast to human table tennis, the table is only 0.5 m wide and 2 m long and the net has a height of 0.25 m. Wire frames were attached to each end of the table and the net. For a shot to be valid, the ball has to pass through all three frames. Thus, the maximum ball speed is limited to 10 m/s. Several early systems were presented by Knight & Lowery (1986), Hartley (1987), Hashimoto et al. (1987) and others. For this early work, the major bottleneck was the lack of fast real-time vision.

An important breakthrough was achieved when Andersson (1988) presented the first robot ping pong player capable of playing against humans and machines. Andersson and his team also employed the simplified robot table tennis rules suggested by Billingsley. He used a high-speed video system and a six DoF PUMA 260 arm with a 0.45 m long stick mounted between table tennis racket and robot. Andersson implemented an expert system-based controller that chooses the strategy as a response to the incoming ping pong ball. The system was later reproduced by Harrison et al. (2005).

In 1993, the last robot table tennis competition took place and was won by Fässler and his team of the Swiss Federal Institute of Technology (Fässler et al., 1990). Although the competitions ceded to exist, the problem was by no means solved, but the current limits were met in terms of robot hardware, algorithms, and vision equipment. The focus in that period was mainly on designing better hardware rather than finding robust algorithms that can compensate for lower-quality hardware.

¹This article extends preliminary work presented at the Conference on the Simulation of Adaptive Behavior (Mülling et al., 2010)

Author	Vision System	Robot System	Comments
Andersson (1988)	Four high speed cameras, 60 Hz	Puma 260, 6 degree of freedom (DoF), 45 cm stick between robot and racket.	First robot ping pong player able to play against humans and robots.
Fässler et al. (1990)	Two CCD cameras, 50 Hz.	Light weight aluminum structures, 6 DoF (3 linear, 3 rotation).	Won the last robot ping pong competition in 1993.
Acosta et al. (2003)	Single CCD video camera, 40 Hz.	Lightweight robot with two paddles, 5 DoF (2 prismatic, 3 revolution).	Engineered for table tennis, just reflecting the ball at the correct angle on a small table.
Miyazaki et al. (2005)	Quick MAG System 3, 60 Hz.	Robot mounted on the table, 4 DoF (2 DoF linear + 2 DoF pan-tilt unit).	Learned input-output maps to predict the impact time, ball position and velocity.
Angel et al. (2005)	Single Sony XC_HC 50 camera, placed on the end-effector.	Parallel robot, 4 DoF, maximum end-effector speed 4 m/s.	Focus on visual control of the robot.

Table 1: This table shows a few robot table tennis systems as examples. Note that most systems include linear axes to achieve the necessary speed or have an additional stick mounted between racket and the robot’s palm.

Nevertheless, interest in robot table tennis did not wane completely and groups continued to work on it. Acosta et al. (2003) constructed a low-cost robot showing that a setup with two paddles is sufficient for playing if the ball is just reflected at the correct angle by a stationary paddle. Miyazaki et al. (2005) were able to show that a slow four DoF robot system consisting of two linear axes and a two DoF pan-tilt unit suffices if the right predictive mappings are learned. See Table 1 for an overview of the major contributions in robot table tennis so far.

All systems were tailored for the table tennis task and relied heavily on high-gain feedback, over-powered motors (no saturation), linear axes (easy to control), and light-weight structures (no torque saturation, little moving inertia). They were engineered in such a way that they could execute any straight movement towards the ball at a rapid pace with the right approach angle. The important problems of generating smooth movements that properly distribute the forces over the arm’s different DoFs was usually avoided.

1.2 Contributions

Our setup differs in three ways from the work existing up to now. First, instead of using a hardware approach that simplifies the motion generation, we use an anthropomorphic redundant seven DoF robot arm. The robot has nonlinear revolute joints, and large inertia, for example, the wrist alone has a mass of 2.5 kg weight at the elbow. Thus, we have strong constraints on the joint velocities and accelerations. Second, our vision system operates in a semi-structured and human-inhabited environment. The third aspect is the application of a biomimetic approach. Humans still outperform robots that are specifically designed for the specific task of robot table tennis. We must therefore investigate how humans control their movements and why their motions are more efficient. Hence, we derive a computational model for human motor control in table tennis and implement this model on a robot. We employ known hypotheses on the movement structure in table tennis, the identification of a virtual target, the timing of interception tasks and the resolution of redundancy.

In this paper, we will proceed as follows. In Section 2, we present knowledge on modeling a table tennis stroke based on biological hypotheses such that we are able to obtain a trajectory of a table tennis stroke in Section 3. In Section 4, we present the evaluation of our system in simulation as well as on a real Barrett WAM™ showing that our setup is able to return incoming volleys to desired points on the opponent’s court. Finally, we summarize our work as well as our results.

2 Modeling Human Striking Movements in Table Tennis

To create an anthropomorphic table tennis playing robot, we rely on hypotheses on human motor control in table tennis. Hence, in this section, we present background information on modeling table tennis from a racket sports' perspective. In particular, we focus on movement generation, movement initiation, the selection of important visual information, and the movement stages in table tennis. Finally, we will outline computational concepts that arise from the biological hypotheses.

2.1 Striking Movement Generation

To perform complex movements, the human brain has to determine a proper sequence of muscle activation, the integration of sensory information, and the generation of motor commands to produce coherent motor behavior. There exist many theories on how humans control coordinated movements, which describe characteristics observed in human motor control as the speed-accuracy trade-off (Woodworth, 1899; Fitts, 1954), the ability to reach goals reliably and repeatedly while still exhibiting variability in the movement of the individual degrees of freedom (Todorov & Jordan, 2002), goal-directed corrections (Elliott et al., 1999), and biomechanical redundancy (Bernstein, 1967). In order to find generative principles underlying movement generation, neuroscientists often employ concepts from optimal control and, thus, use cost functions to evaluate the performance of the system (Bryson & Ho, 1975; Hogan, 1984; Harris & Wolpert, 1998; Todorov & Jordan, 2002). Another important theory is the so-called motor program, a memory-based representation of movement patterns (Henry & Rogers, 1960; Keele, 1968; Schmidt, 1975).

2.1.1 Cost Functions for Resolving the Redundancy in Striking Movements

The human body employs approximately 650 major muscles to control its joints (Scott & Fong, 2004). However, although human beings have many DoF, only a few DoF are actually required for a movement. There exist an infinite number of configurations of joints and muscles in the arm that all accomplish the same task. Such biomechanical redundancy makes the system flexible, but also difficult to control.

One possible generative principle underlying movement generation is the concept of optimal control (Bryson & Ho, 1975). Here, it is postulated that movement patterns for the task are chosen such that a specific cost J

$$\min_{\mathbf{w}} J = \frac{1}{2} \int_0^T c(\mathbf{w}) dt,$$

is minimized, where T is the movement duration and \mathbf{w} the parameters describing the movement. The cost function c measures the cost of the movement with respect to an undesirable feature, such as jerk or the amount of metabolic energy used for the movement.

Most studies on cost functions focus primarily on reaching and pointing movements (Uno et al., 1989; Hogan, 1984; Flash & Hogan, 1985; Harris & Wolpert, 1998) where one can observe a bell-shaped velocity curve (Morasso, 1981), as well as a clear relationship between movement duration and amplitude (Flash & Hogan, 1985; Roitman et al., 2004). However, these relationships do not hold in striking sports as shown by the work of Bootsma & Wieringen (1990). Energy-optimality (Alexaner, 1997; Kuo, 2005) does not appear to explain striking movements, as it would result in movements that are awkward both for human beings and robots. Cruse et al. (1986; 1990) suggested that the comfort of the posture may play a major role in arm movement generation and, hence, it may be part of the cost function of striking movements. According to Cruse (1986), the cost is induced by proximity to a fixed comfort posture in joint-space. This implies that the cost will be minimal if the joint angles are the same as for the comfort posture, and increases with the distance from this posture. Mathematically interpreted this cost function can be understood as minimizing $\sum_i^N c_i(\|q_i^* - q_i\|)$, where q_i^* and q_i are the optimal and current joint values respectively, $c_i(\cdot)$ is a sensitivity function that is attached to this joint and N is the number of DoF. We therefore employ this cost function and resolve the redundancy of the arm accordingly (see Section 3.3).

2.1.2 Motor Programs for Striking Movements

Humans are likely to use a set of pre-structured movement commands, often referred to as motor programs (Henry & Rogers, 1960; Keele, 1968; Schmidt, 1975). Motor programs determine the order and timing of muscle contractions and, hence, define the shape of the action. Sensory information can modify motor programs to generate rapid corrections for environmental changes, as found in table tennis by Bootsma & Wieringen (1990).

Schmidt (1988, 2003) proposed the concept of generalized motor programs (GMP), which are stored in memory and contain elementary movements that may be used as actions. Such actions contain several movements that have similar invariant features such as relative timing and the sequence of the components. These invariant features are consistent when performing the action. A specific movement can be adapted to environmental conditions by altering the parameters of the GMP, such as movement time, amplitude and the goal position of the movement.

As a possible theory on how GMPs could generate coordinated movements, Schmidt (1975, 2003) suggested the motor schema theory. A schema is a set of rules that is developed by extracting information from related experience. The schema provides the parameters for the GMP and, thus, allows it to adapt to new situations and environmental context. According to the theory of Schmidt (1975, 2003), performing an action involves first selecting the appropriate motor program from the memory and, subsequently, setting the adjustable parameters according to the rules defined by the motor response schema.

The validity of this framework for explaining human striking behaviors in table tennis is supported by experimental evidence of Tyldesley & Whiting (1975). Their experiments demonstrated consistent spatial and temporal movement patterns for expert ping pong players. Tyldesley & Whiting (1975) concluded that a professional player chooses a movement program for which the execution time is known from their repertoire, and then decides when to initiate the drive. This observation is known as *operational timing hypothesis*. For more information about the timing of movements see Section 2.2.

For the table tennis robot, we represent the movement program for each degree of freedom involved in striking a table tennis ball as 5th order splines.

2.2 Initiation of Hitting Movements

While spatial planning of the trajectory is necessary for any movement, striking a moving object relies critically on accurate and precise timing to achieve the desired interception. An important component of the visual information, which is used to control the timing in an interception task, is the *time-to-contact* (Hecht, 2004). Time-to-contact is the time until the object reaches the observer or a point in space where it can be intercepted by the observer. Therefore, two essential questions need to be addressed. First, how to specify the time-to-contact and, second, how to coordinate the timing of the action accordingly. Several hypotheses have been suggested to address these questions and the most common ones are discussed below.

2.2.1 Tau Hypothesis

A possible time-to-contact estimation strategy employs the speed and distance of the approaching object. However, this approach cannot explain why humans are able to estimate the time to contact for unknown objects. Lee & Young (1985) hence suggested that the temporal information on the ball's arrival is defined by an optic variable τ . This variable is specified as the relative inverse rate of dilation of the object image on the retina.

The critical value that triggers a specific action is called the *tau margin*. Lee & Young (1985) suggested that humans gear their actions towards the tau margin rather than to the real time to contact. Bootsma & Wieringen (1988) presented experimental evidence that the initiation of the drive in table tennis may indeed be based on the tau-margin.

2.2.2 Stroke Timing Strategies

Hubbard & Seng (1954) concluded from their experience with professional baseball players that the duration of the swing was constant and independent of the speed of the pitched ball. This consistency in swing duration indicates that the batters adjusted the initiation of their swings according to the tau-margin. Such behavior, hence, always results in the same timespan required for each swing, irrespective of the ball's speed.

Tyldesley & Whiting (1975) proposed the operational timing hypothesis independently of the work of Hubbard & Seng (1954). Tildesley and Whiting conducted an experiment with amateur, intermediate and expert table tennis players. The subjects were asked to perform a forehand drive to a designated target area on the table. While expert players and, to a lesser extent, intermediate players achieved a high degree of spatial consistency, novice players failed to do so. The results provided support for the concept of consistent motor programs and lead to the assumption that these motor programs just need to be initiated at the right time. The operational timing hypothesis hence states that expert players reduce the interception task to the problem of determining when to initiate the action.

Bootsma & Wieringen (1990) investigated the validity of the operational timing hypothesis using five expert table tennis players. As expected, they found that the subjects performed consistent movement patterns. However, in contrast to Tildesley and Whiting, they concluded that the small variations in the temporal movement pattern are functional, and not the result of noise. They measured the standard deviation in the timing of the stroke initiation and the ball-racket contact. The initial time accuracy was defined as the variability of the tau-margin. The contact timing accuracy was defined as the variability of the travel direction of the racket at the hitting point. According to the operational hypothesis, the variations at the stroke initiation should be less than those at the hitting point. Instead they found a standard deviation in timing of 2 – 5 ms for the moment of contact while having 5 – 21 ms at initiation. However, their experiments showed a decrease in variability. This finding, therefore, contradicts the operational timing hypothesis.

In later work, Bootsma and Wiering suggested a continuous perception-action model with a funnel-like type of control, in which the participant has to initiate his swing within a fixed spatio-temporal bandwidth (Williams & Starkes, 2002). The drive can be aligned according to perceptual information available during the swing, which would cause an increasing accuracy during the moment of initiation and the moment of ball-racket contact.

In the robot table tennis setup, we will therefore initiate the hitting movement when the time-to-contact is below a threshold value.

2.3 Extracting Essential Context Parameters

To return an approaching ball successfully to the opponent's court, humans have to extract the necessary context parameters from observations of their environment. Knowledge of the gaze behavior can yield information about which parts of the ball flight are important for planning the striking movement.

Studies by Ripoll & Fleurance (1988) and Rodrigues et al. (2002) revealed that expert table tennis players track the ball only at the beginning of its trajectory. The duration of the ball tracking depends on the characteristics of the stroke. For example, balls moving to the player's body are tracked for a longer period than balls which moving lateral to the players body (Ripoll & Fleurance, 1988) due to the difference in measurement accuracy. In contrast to beginners, skilled players exhibit an earlier ball tracking onset (Rodrigues et al., 2002). The final part of the ball trajectory starts with the last bounce of the ball, and ends when the ball and racket make contact. In this time period, head and eyes are fixed on the estimated area of ball-racket contact (Ripoll & Fleurance, 1988). Ripoll and Fleurance concluded from their studies that it is not necessary for expert players to track the ball throughout the entire trajectory. In contrast to the experts, beginners track the ball over the whole trajectory.

The observation of the ball during the first parts of the ball flight allows the player to obtain necessary information on the location of the bounce and for planning the striking motion. The eye-head stabilization in the period before ball-racket contact increases the sensitivity of the moving ball's peripheral image (Ripoll & Fleurance, 1988; Rodrigues et al., 2002). This uncertainty reduction may be crucial for an accurate stroke generation.

2.4 Movement Stages of a Stroke

Table tennis exhibits a regular, modular structure that has been studied by Ramanantsoa & Durey (1994). They analyzed a top player and proposed four movement stages with respect to certain ball events, that is, bouncing, net crossing, and striking. According to their hypothesis, the following four stages can be distinguished during a match of expert player. For clarity, we have labeled them according to their functionality.

Awaiting Stage. The ball moves towards the opponent who returns it towards the net. The player's own racket is moving downwards. The player decides to use either a forehand or backhand stroke. At the end of this

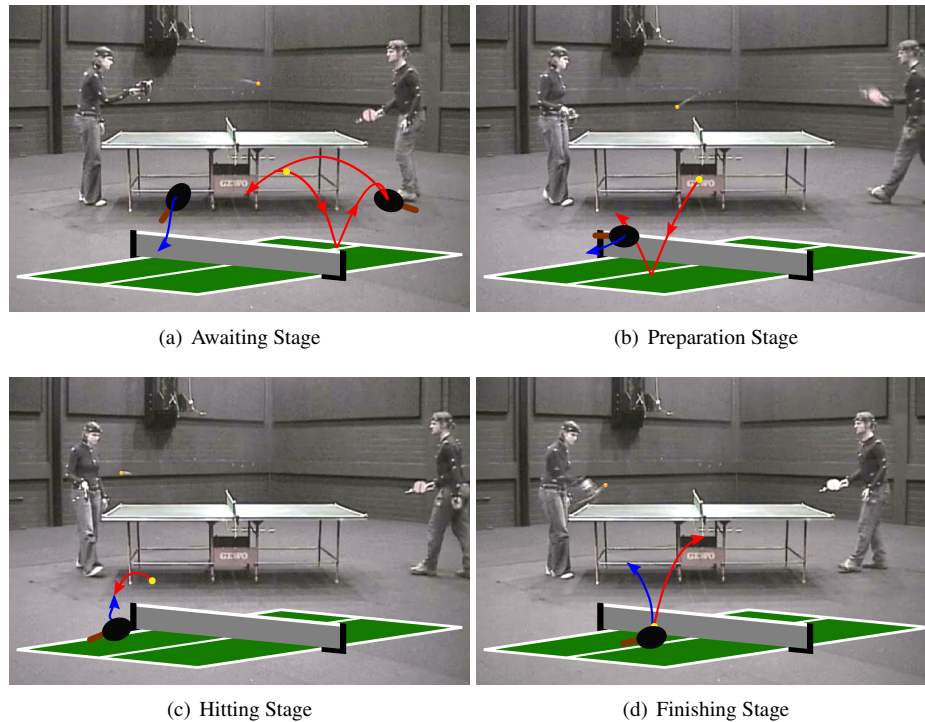


Figure 1: This figure illustrates the four movement stages of Ramanantsoa & Durey (1994) recorded in a VICON™ motion capture system where (a) shows the Awaiting Stage in which the opponent is observed, (b) the Preparation Stage in which the stroke is prepared, (c) the Hitting Stage in which the ball is intercepted, and (d) the Finishing Stage. The red and blue arrow show the movement of the ball and the racket, respectively, in each stage.

stage, the racket will be in a plane parallel to the table's surface.

Preparation Stage. The ball moves towards the player, has already passed the net, and will bounce off the table during this stage. The racket is moving backwards in order to prepare the stroke. For forehand strokes, the racket remains in the same plane as in the awaiting phase. The amplitude of the lateral displacements in the preparation stage depends on the ball's flight direction relative to the body's position when the ball crosses the net. The player chooses a point where he plans to intercept the ball. Ramanantsoa & Durey (1994) refer to this point as the virtual hitting point.

Hitting Stage. The ball moves towards the virtual hitting point where the player intercepts it. In a first substage, final adjustments are made. The precision of the stroke depends on the time available for the execution of this substage. During the second substage, the racket moves towards the virtual hitting point until it hits the ball. For expert players, the temporal duration of this phase appears to be constant and lasts approximately 80 ms.

Finishing Stage. After having been hit, the ball moves towards the opponent while the racket is moving upwards to a stopping position. This stage ends with the ball crossing the net and the velocity of the racket tending to zero.

We have verified the stages suggested by Ramanantsoa & Durey (1994) in a VICON™ motion capture setup with two naive players where each of the stages can be observed distinctly (see Figure 1). Color coded trajectories

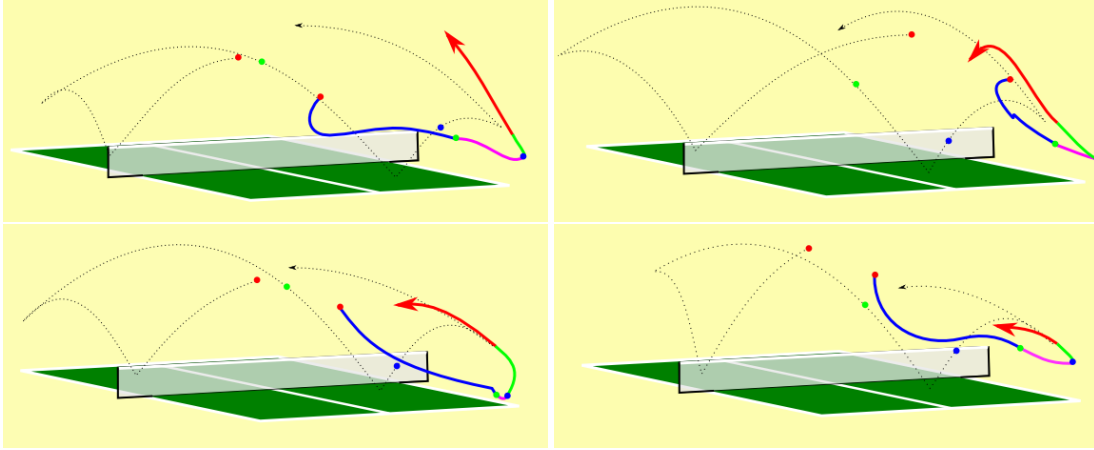


Figure 2: This figure shows different trajectories of intermediate table tennis players for one hitting motion. The trajectories are color coded according to the stages suggested by Ramanantsoa and Durey. The awaiting stage is colored in blue, the preparation stage in magenta, the hitting stage in green and the follow through stage in red. Colored circles show the corresponding position on the ball and arm trajectory respectively.

of the racket for one hitting motion starting with the awaiting stage can be seen in Figure 2. Note that for expert table tennis players the temporal and spacial timing would be more pronounced.

From a computational point of view, this model corresponds to a finite state automaton. By encoding such a sequence in the internal states of the policy, similar movements can be generated artificially.

3 A Biologically-Inspired Trajectory Generator for Table Tennis Strokes

To evaluate and use the observation-based model presented in Section 2, we have developed a computational realization that is suitable for real-time execution on a robot. We proceed as follows: First, we discuss the system’s key elements in an overview. Subsequently, we explain how to determine the goal parameters of table tennis, that is, the time-to-contact and the virtual hitting point. Furthermore, we describe the realization of the movement generation for the striking motion, which includes the resolution of redundancy by minimizing the discomfort as described in Section 2.1.1.

3.1 Overview of the Biomimetic Player

We adopt the movement stages of the model by Ramanantsoa & Durey (1994) as outlined in Section 2.4, and use a finite state automaton to represent this model. Subsequently, we have to plan the arm’s movement for each of the four stages.

We decided to plan the trajectories in the robot’s joint space. Planning in joint space has a variety of advantages over planning in task space. A joint-space trajectory can be directly controlled and does not require an intermediate inverse kinematics mapping. Thus, we avoid an additional nonlinear component in the control loop. In order to realize the movement program, we rely on a spline-based representation to encode the trajectory for each stage and each DoF. More details are given in Section 3.6.

To generate the arm’s trajectories, we have to determine constraints for the movements of each DoF. While desired final joint configurations suffice for the awaiting, preparation and finishing stages, the hitting stage requires a well-chosen movement goal. To determine the goal parameters, that is, the position, velocity and orientation of the end-effector, we rely on the virtual hitting point hypothesis (Ramanantsoa & Durey, 1994). The goal parameters are chosen before the hitting motion begins. Therefore, the system has to first choose a point $\mathbf{x}_{\text{table}}$ on the opponent’s court where the ball will be returned to. The choice of $\mathbf{x}_{\text{table}}$ is part of a higher level strategy. Second, the system needs to determine the interception point of the ball and racket trajectories, which specifies the virtual hitting point

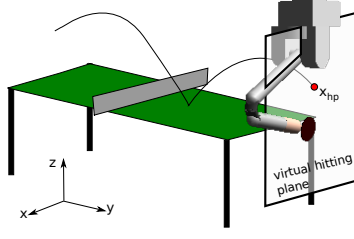


Figure 3: This figure illustrates the virtual hitting plane in the table tennis setup. The intersection point of the ball trajectory with the virtual hitting plane defines the virtual hitting point \mathbf{x}_{hp} . The x direction of the world coordinate system is parallel to the net, the y direction is parallel to the long side of the table and the z direction goes upwards.

\mathbf{x}_{hp} . The hitting point is the intersection point of the ball with the virtual hitting plane of the robot. The fixed virtual hitting plane used in our setup is illustrated in Figure 3. Given \mathbf{x}_{table} and \mathbf{x}_{hp} , we can compute the batting position and velocity of the racket. Unfortunately, there are still infinitely many arm configurations that correspond to the desired racket position and velocity. To resolve this ambiguity problem, the system minimizes the distance to a comfort posture in joint space as proposed by Cruse et al. (1990) for human motor control. Doing so, we are able to find the orientation of the end-effector at the hitting point closest to the desired posture (see Section 3.3). The corresponding joint configuration for the virtual hitting point can then be computed using a quaternion-based inverse kinematics (see Section 3.4).

The hitting movement is initiated when the time t_{hp} to the intersection of the ball with the virtual hitting plane is below a threshold. This step requires the system to predict when the ball is going to reach the virtual hitting plane. The expected hitting time can be estimated by predicting the trajectory of the ball using a model of the ball's dynamics, which is described in Section 3.2. Following the suggestion of Bootsma & Wieringen (1988) that some online adaptation of the movement can be performed, the virtual hitting point is updated during the execution of the movement. As a result, the movement generation for the hitting and finishing stages is adapted to the new hitting point up to 100 ms before the predicted interception. This time period corresponds approximately to the human visuomotor delay (Abbs & Cole, 1987).

In order to realize these four stages, the system needs to detect the ball and determine its position \mathbf{x}_b . The vision system consists of two stereo cameras with Prosilica GE640C Gigabit Ethernet cameras and a GPU-based 60 Hz blob detection. Due to noise on the sensor level, the visual information is filtered using an extended Kalman filter (EKF) (Sorenson, 1985). The system's equations used for the EKF are given in Equation (1). The resulting algorithm is described in Algorithm 1.

3.2 Dynamics Model of the Table Tennis Ball

To predict the position and velocity of the ball at time t_{j+1} based on the state at time t_j , we have to model the aerodynamics of the ball and the physics of the ball's bounces on the table. The ballistic flight model needs to incorporate air drag, gravity, and spin. As the latter is hard to observe from visual measurements, our model neglects spin. For table tennis balls, we can assume that the air drag is proportional to the square of the velocity of the ball. Using symplectic Euler integration, we can model the ball movement in discrete time form by

$$\ddot{\mathbf{x}}_b^{j+1} = \mathbf{g} - C \|\dot{\mathbf{x}}_b^j\| \dot{\mathbf{x}}_b^j, \quad \dot{\mathbf{x}}_b^{j+1} = \dot{\mathbf{x}}_b^j + \ddot{\mathbf{x}}_b^{j+1} \Delta t, \quad \mathbf{x}_b^{j+1} = \mathbf{x}_b^j + \dot{\mathbf{x}}_b^{j+1} \Delta t, \quad (1)$$

where $\ddot{\mathbf{x}}_b$, $\dot{\mathbf{x}}_b$, and \mathbf{x}_b denote the acceleration, velocity, and position vector of the ball respectively, \mathbf{g} the gravity vector and Δt the time difference. The air resistance scale factor $C = c_w \rho A / (2m)$ is determined by the drag coefficient c_w , the density of the air ρ , the size of the ball's cross-sectional area A and the mass of the table tennis ball m . For the bouncing behavior of the ball on the table, we assume velocity changes in all three directions. This change in velocity $\dot{\mathbf{x}}_b^{j+1} = \epsilon_T \dot{\mathbf{x}}_b^j$ is determined by the coefficient $\epsilon_T = [\epsilon_{Tx}, \epsilon_{Ty}, -\epsilon_{Tz}]^T$ where ϵ_{Tx} is the coefficient of friction on the table and ϵ_{Tz} is the coefficient of restitution.

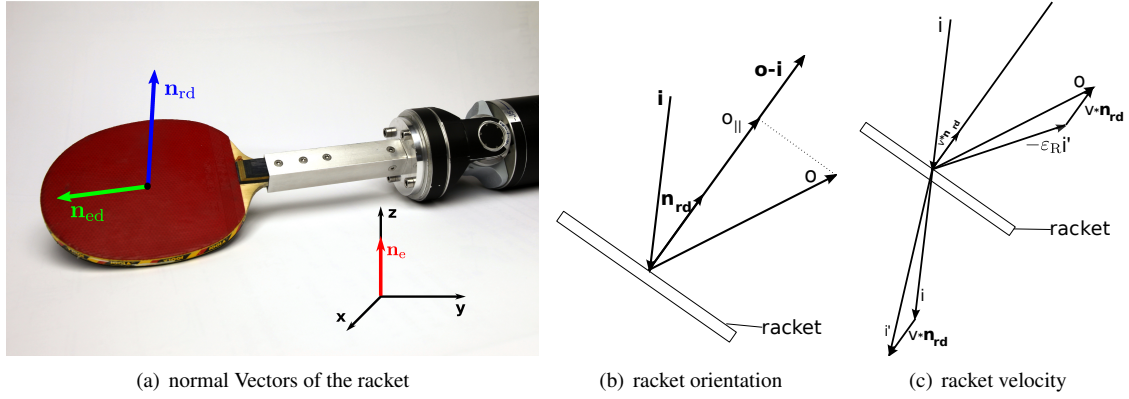


Figure 4: Computation of the desired normal vector \mathbf{n}_{ed} of the end-effector and the velocity based on the orientation of the racket \mathbf{n}_{rd} , the velocity \mathbf{o} and \mathbf{i} of the ball after and before the impact respectively. Figure (a) illustrates the normal vector \mathbf{n}_e , the desired orientation of the racket \mathbf{n}_{rd} and the resulting desired orientation \mathbf{n}_{ed} . The normal of the end-effector \mathbf{n}_{ed} is perpendicular to \mathbf{n}_{rd} . Figure (b) shows the relationship between \mathbf{n}_{rd} , \mathbf{o} and \mathbf{i} . Assuming the absence of spin and a speed change only in the $\mathbf{o} - \mathbf{i}$ direction, \mathbf{n}_{rd} is given by the normalized difference vector of \mathbf{o} and \mathbf{i} . Figure (c) illustrates the computation of the velocity \mathbf{v} of the racket based on the relation of \mathbf{v} , \mathbf{n}_{rd} , \mathbf{o} and \mathbf{i} .

3.3 Determining the Goal Parameters

After estimating the virtual hitting point, we need to determine the orientation and velocity of the end-effector. Therefore, the system chooses the height z_{net} at which the returning ball should pass over the net, as well as the position \mathbf{x}_{table} where the ball should bounce on the opponent's court. The choice of these variables belongs to the strategy of the system, which is out of the scope of this paper. For the purpose of demonstration, we will choose z_{net} and \mathbf{x}_{table} independently from sets of possible values according to a uniform distribution. To determine the goal parameters, we have to first compute the desired outgoing velocity \mathbf{o} of the ball that is, the velocity of the ball at the moment after the ball-racket impact. Directly from it, we can determine the required velocity and orientation of the racket.

Desired Outgoing Velocity. To compute the velocity \mathbf{o} of the ball after the ball-racket impact at \mathbf{x}_{hp} , we need to know which requirements the resulting trajectory has to fulfill. First, the ball has to pass the net at a certain height z_{net} . Second, the ball has to bounce at the desired target \mathbf{x}_{table} on the opponent's court. These two constraints lead to the following nonlinear equations

$$\begin{aligned} x_{net} &= x_{hp} + o_x t_{net} - 0.5Cv o_x t_{net}^2 \\ z_{net} &= z_{hp} + o_z t_{net} - 0.5Cv o_z t_{net}^2 \\ \mathbf{x}_{table} &= \mathbf{x}_{hp} + \mathbf{o} t_{tab} - 0.5Cv \mathbf{o} t_{tab}^2 \end{aligned}$$

where $\mathbf{x}_{net} = [x_{net}, z_{net}]^T$, $\mathbf{x}_{hp} = [x_{hp}, y_{hp}, z_{hp}]^T$, $\mathbf{x}_{table} = [x_t, y_t, z_t]^T$, $\mathbf{o} = [o_x, o_y, o_z]^T$, x_{net} is the x -position of the net, z_t is the height of the table, and $v = \sqrt{o_x^2 + o_y^2 + o_z^2}$. The time variables t_{net} and t_{tab} correspond to the time at which the ball passes over the net and reaches the target of the opponent court respectively. Since these equations are nonlinear in the variables of interests, we have to solve the problem numerically. Therefore, we use a globally convergent solver for nonlinear equation systems, which combines the Newton-Raphson update with a modification for global convergence (Dennis & Schnabel, 1983).

Racket Orientation. The orientation of the end-effector is specified as a rotation that transforms the normal vector of the end-effector \mathbf{n}_e to the desired normal vector \mathbf{n}_{ed} . To define \mathbf{n}_{ed} , we have to compute the desired normal direction of the racket \mathbf{n}_{rd} that results in the desired outgoing vector \mathbf{o} of the ball given the velocity vector \mathbf{i} at the hitting point before impact (see Figure 4).

If we assume only a velocity change $\mathbf{o} - \mathbf{i}$ in the normal direction of the racket \mathbf{n}_{rd} , we obtain $\mathbf{o} - \mathbf{i} = \mathbf{n}_{rd}(o_{||} - i_{||})$, where $o_{||}$ and $i_{||}$ denotes the components of \mathbf{o} and \mathbf{i} , respectively, which are parallel to the desired racket normal. Note, that if we assume the absence of spin, all changes in velocity occur in this component. Hence, $\|\mathbf{o} - \mathbf{i}\| = o_{||} - i_{||}$. Thus, we can determine the desired racket normal by

$$\mathbf{n}_{rd} = \frac{\mathbf{o} - \mathbf{i}}{\|\mathbf{o} - \mathbf{i}\|}. \quad (2)$$

The rotation of \mathbf{n}_e to \mathbf{n}_{ed} , which defines the orientation, is represented in terms of quaternion $q_{ed} = q_{rd}q_{re}$ where q_{re} is the quaternion that describes the rotation from the racket to the end-effector and $q_{rd} = (\cos(\gamma/2), \mathbf{u} \sin(\gamma/2))$, with $\gamma = \mathbf{n}_e^T \mathbf{n}_{rd} / (\|\mathbf{n}_e\| \|\mathbf{n}_{rd}\|)$ and $\mathbf{u} = \mathbf{n}_e \times \mathbf{n}_{rd} / \|\mathbf{n}_e \times \mathbf{n}_{rd}\|$, defines the transformation of the normal of the end-effector \mathbf{n}_e to the desired racket normal \mathbf{n}_{rd} . See Figure 4 for an illustration of the computation of the racket normal used to define the orientation of the end-effector.

As there exist infinitely many racket orientations that have the same racket normal, we need to determine the final orientation depending on a preferred joint configuration. The resulting quaternion of the end-effector q_{ed} is determined by a rotation about \mathbf{n}_{rd} . The corresponding joint values and velocities are then computed using inverse kinematics (see Section 3.4). The orientation whose corresponding joint configuration θ_{hp} yields the minimum distance to the comfort posture θ_{com} is used as the desired racket orientation. Therefore, we weight each DoF according to its contribution to the cost function. The comfort position θ_{com} is a fixed joint configuration for each of the different striking motions (see Section 3.5). We choose this configuration such that the distance to the joint limits is as large as possible.

Required Racket Velocity. After computing the orientation of the end-effector, we have to calculate the velocity of the end-effector at the time of the ball's interception. We can describe the relation between the components of the incoming and outgoing velocity parallel to the racket norm as $o_{||} - v = \epsilon_R(-i_{||} + v)$, where ϵ_R denotes the coefficient of restitution of the racket, v the speed of the racket along its normal. This equation can be solved for v which yields the desired racket velocity

$$v = \frac{o_{||} + \epsilon_R i_{||}}{1 + \epsilon_R}. \quad (3)$$

Given the velocity and orientation of the racket, we now have all necessary information to specify the virtual hitting point.

3.4 Translating Virtual Hitting Points into Configurations

From the virtual hitting point, we have computed the position, orientation and velocity of the end-effector in task space (i.e., Cartesian positions and rotations in the form of quaternions). As we plan the motions in joint space (as discussed in Section 3.1), we have to compute the corresponding joint state θ consisting of the joint angles θ_i , joint velocities $\dot{\theta}_i$ and joint accelerations $\ddot{\theta}_i$ for each joint i of the arm. The transformation of joint values into Cartesian positions and velocities \mathbf{x} can be determined using the forward kinematics $\mathbf{x} = f(\theta)$. Hence, to realize the transformation from Cartesian space into the joint space, we require the inverse kinematics, that is, $\theta = f^{-1}(\mathbf{x})$. Solving the inverse kinematics problem by analytically inverting the forward kinematics is not feasible, as there may exist infinitely many solutions due to the redundancy. To yield human-like motions, we follow the suggestion of Cruse et al. (1990) for resolving the biomechanical redundancy in humans. Hence, the inverse kinematics problem for the redundant DoFs can be solved numerically by minimizing the distance to the comfort posture in joint space, while finding a racket position and orientation that coincides with the virtual hitting point \mathbf{x}_{hp} . The cost function of the inverse kinematics problem is given by

$$\min_{\Delta \theta_{0:T}} F = \sum_{t=0}^T \frac{1}{2} (\Delta \theta_t + \theta_t - \theta_R)^T \mathbf{W} (\Delta \theta_t + \theta_t - \theta_R) \quad (4)$$

$$\text{s.t. } \Delta \mathbf{x}_t = \mathbf{J}(\theta_t) \Delta \theta_t, \quad (5)$$

where θ_R is the optimization posture, the change in the joint state $\Delta\theta = \dot{\theta}\delta t$, θ_t is the current joint configuration of the robot and \mathbf{W} is a symmetric, positive definite weight matrix. Thus, we have the Lagrangian

$$L = \sum_{t=0}^T \frac{1}{2} (\Delta\theta_t + \theta_t - \theta_R)^T \mathbf{W} (\Delta\theta_t + \theta_t - \theta_R) + \lambda_t^T (\Delta\mathbf{x} - \mathbf{J}(\theta_t) (\Delta\theta_t + \theta_t - \theta_R)). \quad (6)$$

Minimizing L with respect to λ and $\Delta\theta$ yields

$$\theta_{t+1} = \theta_t + \mathbf{J}_w^\dagger \Delta\mathbf{x}_t + (\mathbf{I} - \mathbf{J}_w^\dagger \mathbf{J}) (\theta_t - \theta_R), \quad (7)$$

where \mathbf{I} denotes the identity matrix, $\mathbf{J}_w^\dagger = \mathbf{J}_w^T (\mathbf{J}_w \mathbf{J}_w^T)^{-1}$ denotes a weighted pseudo inverse, and $\mathbf{J}_w = \mathbf{J} \mathbf{W}^{-1}$ denotes the weighted Jacobian. The final joint configuration for the desired hitting point can be computed by iteratively running the algorithm until the error is below a certain error threshold. Note that this method may have numerical problems at singularities where the rank reduction makes the Jacobian inversion impossible and an additional ridge term is needed. Nevertheless, this method enables a fast computation of the inverse kinematic function that provides control over the arm configurations and is used for the transformation of the Cartesian position and orientation into joint space.

The step length in task space $\Delta\mathbf{x} = [\Delta\mathbf{x}_p; \Delta\mathbf{x}_o]$ determines the accuracy of the cost and, hence, is essential for computing a short path. We define the position part of the Cartesian difference vector $\Delta\mathbf{x}$ as

$$\Delta\mathbf{x}_p = g_p (\mathbf{x}_{hp} - \mathbf{x}_e), \quad (8)$$

where g_p is a scaling factor and \mathbf{x}_e and \mathbf{x}_{hp} are the actual and desired Cartesian states of the end-effector, respectively. For orientation control, we use the quaternion error q^E between the actual and desired orientation for the second part of the step length

$$q^E = q_{ed} q_e^*, \quad (9)$$

$$\Delta\mathbf{x}_o = g_o [q_1^E, q_2^E, q_3^E]^T, \quad (10)$$

where q_{ed} is the quaternion describing the desired end-effector orientation, q_e^* denotes the complex conjugate of the actual quaternion of the end-effector orientation and g_o defines a scaling factor. The scaling factors g_p and g_o determine the convergence speed and robustness of the method; large values result in a higher convergence speed, but the algorithm may never find a solution if the values are set too high.

When the algorithm cannot find an appropriate joint configuration, the system decides not to return the incoming ball.

3.5 Movement Parameters

To return an incoming ball, the movement is generated for each of the four stages. We determine the start and end position, velocity and acceleration for each of the four stages as described in Section 3.1. The start and end positions for all stages are fixed except the hitting point which defines the end configuration of the hitting stage and the start configuration of the finishing stage. The fixed start and end positions are chosen to produce hitting movements similar to those exhibited by humans. The corresponding joint velocities and accelerations are set to zero. The configuration at the hitting point is computed for each stroke individually as described above. The duration of each stage (i.e., t_{as} , t_{ps} , t_{hs} , t_{fs}) is chosen such that the robot is able to execute the movement. The durations of the hitting stage t_{hs} is equal to the estimated time to hit t_{hp} but is at least 250 ms and at most 350 ms. If the estimated time to hit is less than 250 ms and the hitting stage is not yet initiated, the system decides not to execute the stroke, since this would require infeasible accelerations for the robot's joints.

Similar to humans who distinguish between several striking movements in table tennis, we have to implement different strokes to adapt to different hitting points in the workspace of the robot. Therefore, we divide the virtual hitting plane into three areas. A stroke movement is assigned to each of these areas, resulting in two forehand and one backhand strokes. Each stroke is defined by an individual start and end position for the individual stages and joints. When a ball moving towards the robot is detected, the virtual hitting point is estimated. Based on the area in which the predicted hitting point is located, the corresponding stroke type is chosen.

Algorithm 1 Table Tennis Algorithm

Initialize: switch to *AwaitingStage*

repeat

 Extract ball position \mathbf{x}_b

 EK-Filter: $\mathbf{x}_b \rightarrow \mathbf{x}_t, \dot{\mathbf{x}}_t$

 EK-Prediction: $\mathbf{x}_t, \dot{\mathbf{x}}_t \rightarrow t_{hp}, \mathbf{x}_{hp}, \dot{\mathbf{x}}_{hp}$

 —————Switch Stage—————

if *FinishingStage* **and** *MovementEnds*

 Switch to *AwaitingStage*

 Compute $\alpha_k = \mathbf{M}^{-1}(0, t_{as}) \mathbf{b}_k^{as}$ for each DoF k

else if *AwaitingStage* **and** $t_{hp} \leq t_{as} + t_{hs}$

 Switch to *PreparationStage*

 Compute $\alpha_k = \mathbf{M}^{-1}(0, t_{ps}) \mathbf{b}_k^{ps}$ for each DoF k

else if *PreparationStage* **and** $t_{hp} \leq t_{hs}$

 Switch to *HittingStage*

else if *HittingStage* **and** *BallHit*

 Switch to *FinishingStage*

 Compute $\alpha_k = \mathbf{M}^{-1}(0, t_{fs}) \mathbf{b}_k^{fs}$ for each DoF k

end if

 —————Update Striking Motion—————

if *HittingStage*

 Solve with Newton-Raphson for \mathbf{o} using

$$f(\mathbf{o}, \mathbf{x}_{hp}, t_{net}) = \mathbf{x}_{net}$$

$$f(\mathbf{o}, \mathbf{x}_{hp}, t_{table}) = \mathbf{x}_{table}$$

 Determine joint configuration at hitting point

$$v = o_{||} + \epsilon_R i_{||} / (1 + \epsilon_R)$$

$$\mathbf{n}_{rd} = \mathbf{o} - \mathbf{i} / (\|\mathbf{o} - \mathbf{i}\|)$$

$$q_{ed} = q_{rd} q_{yrot}$$

 Determine optimal rotation about \mathbf{n}_{rd} by

$$\theta_{opt} = \arg \min_{\theta_{hp}} \|\theta_{com} - \theta_{hp}\|$$

 with Inverse Kinematics: $\mathbf{x}_{hp}, q_{ed}, v \rightarrow \theta_{hp}$

 Compute $\alpha_k = \mathbf{M}^{-1}(0, t_{hs}) \mathbf{b}_k^{hs}$ for each DoF k

end if

 —————Executing Movement—————

for each DoF k **do**

$$\theta_k = \sum_{l=1}^5 \alpha_{kl} t^l$$

end for

 Execute $(\theta, \dot{\theta}, \ddot{\theta})$ with Inverse Dynamics Control.

until user stops program

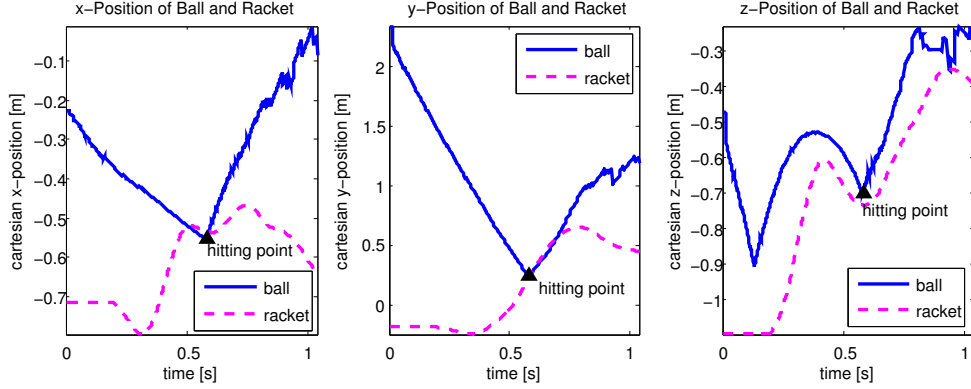


Figure 5: This figure shows the movement of the racket and the ball on the real Barrett WAM™ for a successful striking movement. The ball (solid blue line) moves towards the robot until it is hit by the racket (dashed magenta line) at the virtual hitting point (black triangle). The y-axis is aligned with the long side of the table tennis table, the x-axis is aligned with the width of the table and the z-axis goes upwards.

3.6 Movement Generation

The trajectory needs to be planned in joint space where high velocity movements can be executed more reliably. For the execution of the movements, we need a representation to obtain position $\theta(t)$, velocity $\dot{\theta}(t)$ and accelerations $\ddot{\theta}(t)$ of the joints of the manipulator at each point in time t such that it can be executed with an inverse dynamics-based controller (Spong et al., 2006). Fifth order polynomials represent the trajectory at all stages (see Appendix). Such polynomials are the minimal sufficient representation to generate smooth trajectories, and can be evaluated quickly (Craig, 1989). Mimicking the four stage model of Ramanantsoa & Durey (1994), we select four different splines that interpolate between the initial and final configurations. As the trajectory of the hitting and finishing stages depends on the hitting point, trajectories have to be calculated jointly at the beginning of the hitting stage and have to be adjusted every time the virtual hitting point is updated.

The coefficients and final acceleration of the hitting stage are chosen such that the maximal acceleration during this stage is minimized (see Appendix). In addition to the minimization of the acceleration, this solution results in a decreased position overshoot before the hitting point. Thus, it reduces the risk of running into joint limits.

The pseudo-code summarizing the table tennis setup described in this section is shown in Algorithm 1.

4 Evaluations

In this section, we demonstrate that robot table tennis is feasible with the proposed biomimetic player. To evaluate the performance of the player, we examined the accuracy in striking an incoming ball. We also analyzed the accuracy of the prediction of the virtual hitting point, the time of flight, and the accuracy of returning the ball to a desired position on the opponent’s court. Furthermore, we analyzed the performance of the dynamics model of the ball and compared the biomimetic player’s movement generation to a human player. For the experimental evaluations, we used a Barrett WAM™ arm, as well as a physically realistic simulated version of the complete setup.

4.1 Evaluation against a Ball Launcher

We assume a setup consisting of a Barrett WAM™, a table, a racket, four cameras and a ball launcher. The Barrett WAM™ is an anthropomorphic seven DoF arm that is capable of high speed motion (e.g., for the end-effector we measured velocities of 6.5 m/s and accelerations of 234.6 m/s²). The robot is mounted in a hanging position from the ceiling. A standard racket, with a 16 cm diameter, is attached to the end-effector. The setup includes a standard sized table and a table tennis ball according to the ITTF rules. The ball is served randomly by a ball launcher to

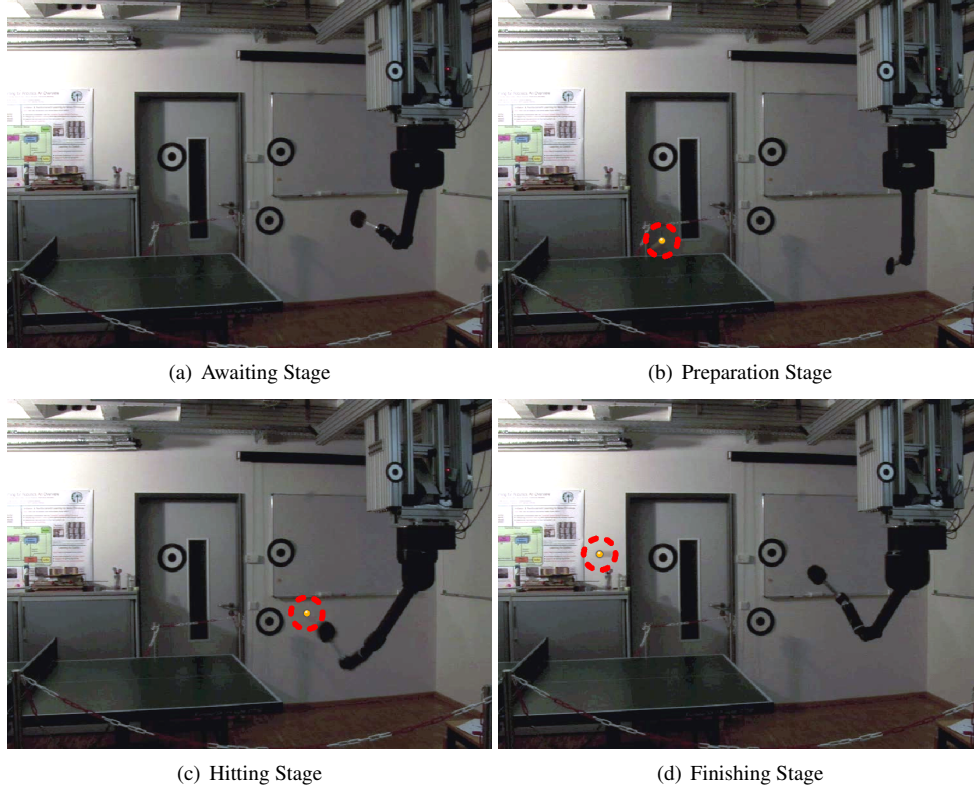


Figure 6: The figure shows the different stages, matching those in Figure 1, but performed by the real robot. At the beginning the robot is in the rest posture waiting for an incoming ball (Subfigure a). As a ball moves towards the robot, the arm performs a backswing motion to prepare the striking movement (Subfigure b). Based on the prediction of the ball the system chooses a hitting point. When the estimated time to contact reaches a critical value the arm moves towards the hitting point and hits the ball back to the opponent (Subfigure c). The robot arm moves upwards with decreasing velocity until the velocity is zero (Subfigure d).

the work space of the robot. As a result, the ball passes the robot's end of the table in an area of approximately $2\text{ m} \times 0.75\text{ m}$. This area serves as the virtual hitting plane. The target point on the opponent's court is chosen randomly over the whole area.

Evaluation in a Simulated Environment. The table tennis system is capable of returning an incoming volley to the opponent's court which is served by a ping pong ball launcher at random times and to arbitrarily chosen positions in the workspace of the robot. In simulation, when the ball was served 1,000 times to random positions in the work-space of the robot, the system was able to return 95.5% of the balls. In 85% of the trials the ball was returned successfully to the opponent's court. In 4.5% of the trials the system refused to perform a hitting motion due to joint and acceleration limits. The mean deviation from the target point of the opponent's court was approximately 13 cm. The mean deviation of the position of the racket's mid-point to the ball at the moment of contact was 3 cm. The estimated hitting point was corrected by 2.5 cm from the initial estimate. The estimated hitting time changed by approximately 20 ms. When the system gets the true position and velocity of the ball, the ball was returned to the opponent's court in 96% of the trials. The simulation of the Barrett WAMTM robot arm was created using the SL framework (Schaal, 2009). To build the table tennis environment, we used a model of the flight and bouncing behavior of the ball as discussed in Section 3.2. The coefficients of restitution of both racket-ball and ball-table interactions were determined experimentally ($\epsilon_R = 0.78$, $\epsilon_{Tx} = 0.73$, $\epsilon_{Tz} = 0.92$). As the parameters for the air drag, we estimated $c_w = 0.47$, $m = 0.0027\text{ kg}$, $\rho = 1.293\text{ kg/m}^3$ and $A = 0.0013$ from real

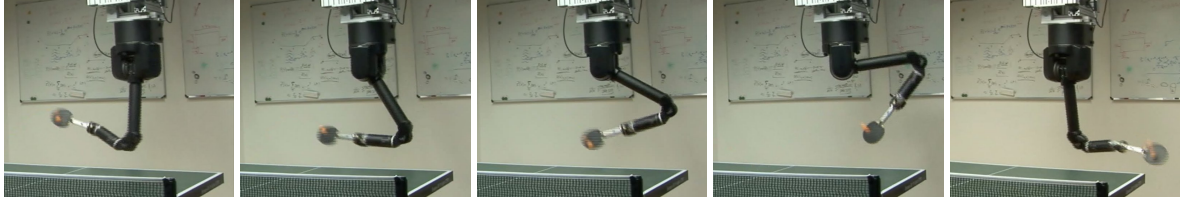


Figure 7: Various target of the biomimetic table tennis setup. The plane of possible hitting points has a length of 1.8 m.

table tennis trajectories.

Evaluation on the Real System. Subsequently, we successfully transferred the setup onto a real Barrett WAMTM robot equipped with two partially overlapping stereo camera pairs. We used the same biomimetic player as in the simulated setup. The extended Kalman filter, based on the ballistic flight of a point mass with estimated restitution factors, tracks the ball well (for an extensive discussion see Section 4.2). However, the prediction of the virtual hitting point and time is less accurate in the real world than in simulation due to the neglected spin and the inaccuracies of the vision system. As ground truth data is not available, this can only be assessed qualitatively. Furthermore, all balls served by the ball launcher possess a large amount of topspin. Hence, these predictions needed to be updated more frequently and the trajectory generation adapts accordingly. A ball launcher served 150 balls to the fore- and backhand area of the robot. 99 % of these balls were returned by the system. The improved performance compared to the simulated system is mainly due to the limited workspace of the ball launcher. In simulation the ball cannon is able to serve balls to each point of the workspace of the robot with variable velocities. See Figure 6 for snapshots of a stroke and Figure 5 for trajectories of the racket and the ball of the real system. In the evaluations the robot covered a hitting plane with a length of 1.8 m illustrated in Figure 7. A video showing the performance of the system can be found at <http://www.robot-learning.de/Research/BiomimeticPlayer>.

4.2 Accuracy of the ball dynamics model

We analyzed the accuracy of the ball’s dynamics model (see Section 3.2) on various ball trajectories. To assess the model’s quality, we recorded ball paths from a competitive and a cooperative human table tennis game. Furthermore, we recorded trajectories with extreme top-, side- and underspin to study the robustness of the proposed biomimetic player in returning these balls.

In order to investigate how well the dynamics model corresponds to the real system, we computed the deviation of the internal dynamics model from the trajectories captured by the vision system. Therefore, we first estimated the velocity of the ball at the bouncing point using the EKF. The trajectory of the ball was then precomputed for 200 ms using the dynamics model. The mean deviation of the ball position 200 ms after the bounce was estimated by computing the root mean squared error (RMSE) between both trajectories. In a cooperative game of two naive players (including the ball trajectories of 200 volleys) the average amount of deviation was $5.9 \text{ cm} \pm 2.6 \text{ cm}$. The deviation in each direction was smaller than $3.6 \text{ cm} \pm 2.4 \text{ cm}$. In the competitive game the player used different kinds of spin as well as smash strokes such that it was difficult for the opponent to return the ball. The mean deviation of the used dynamics model from the vision information was $11.59 \text{ cm} \pm 6.8 \text{ cm}$. The main difference to the cooperative game was the average amount of topspin and therefore the deviation in the y direction which increased from $3.6 \text{ cm} \pm 2.4$ to $10.54 \text{ cm} \pm 7.6 \text{ cm}$.

Unlike a human being, a ball launcher can create trajectories with only top-, side- or underspin. Hence, we analyzed the biomimetic player using trajectories produced by a ball launcher with regard to sensitivity to the different types of spin. Despite that the mean deviation of the topspin trajectory was $39.16 \text{ cm} \pm 4.5 \text{ cm}$ (with a deviation of $38.58 \text{ cm} \pm 4.8 \text{ cm}$ in y direction), the biomimetic player was still able to adapt to these changes and returned 61% of the balls successfully to the opponent’s court. The underspin trajectories had a mean deviation of $11.05 \text{ cm} \pm 3 \text{ cm}$. Here, the biomimetic player was able to return 85% of the served balls successfully to the opponent’s court. However, its performance decreased when facing sidespin as the biomimetic player was not able

	Error		
	x	y	z
Human game (cooperative)	03.43 ± 2.05	03.59 ± 2.40	3.19 ± 1.79
Human game (competitive)	04.03 ± 2.85	10.54 ± 3.51	2.58 ± 2.44
extreme Topspin	02.01 ± 1.57	38.58 ± 4.77	6.22 ± 3.06
extreme Sidespin	17.79 ± 1.46	04.63 ± 2.61	2.79 ± 1.44
extreme Underspin	01.17 ± 0.77	09.84 ± 3.53	4.90 ± 1.97

Table 2: Root Square Mean Error in centimeters of the deviation of the applied dynamics model of the ball and the vision information of the ball 200 ms after the bounce. The velocity of the model was set to the estimate of the EKF before bouncing. The y direction corresponds to the side direction of the table, x direction corresponds to the long side of the table and z to the height (see Figure 3).

to adapt to these trajectories. The mean deviation from the observed trajectories was $18.6 \text{ cm} \pm 1.1 \text{ cm}$. Most prediction error was due to the differences in the x direction, that is, the direction along the net. Table 2 shows the deviations of the different recordings from the vision information 200 ms after the bounce for all three Cartesian directions. Note that the spin produced by the ball launcher was much higher than the spin that we found in games of naive human players.

4.3 Comparison to Human Behavior and Performance

The robot is able to return balls over the whole width of the table. However, compared to a human player, the robot lacks the acceleration abilities. Therefore, the human can adapt to unexpected ball trajectories after the ball bounces of the table much better than the robot. While the human can perform the hitting motion in 80 ms the robot needs 350 ms due to lower accelerations. Furthermore, the human is able to switch from for- to backhand while observing the ball moving towards him. The Barrett WAMTM may needs up to 500 ms for this action to avoid violating its acceleration and torque limits. The robot is not just limited by its acceleration abilities but also by the missing degrees of freedom; as it has no floating base, it cannot move sideways as well as back and forwards. Thus, to compare the performance of our biomimetic player to a human, we have to compare it to a human that is standing at a fixed location. When the ball was served to the small forehand area of a stationary human, the performance of returning a ball successfully to an certain area of the table decreased to 50%. Without a given target on the table, the human was able to return 97 % of the balls successfully to the opponent’s court.

We also compared the hitting movements of the robot qualitatively to the strikes of a naive human player. Similar to humans, the robot hits the ball in a circular hitting movement in an upward motion. The impact point of the racket and the ball is on an axis parallel to the table going through the base of the human and robot respectively.

5 Discussion and Conclusion

In this section, we discuss the inclusion of strategy, optimization possibilities for the movement generation and spin estimation. Finally, we summarize the contributions of this paper.

Including a Strategy. The presented evaluation is based on a strategy where the player does not attempt to beat the opponent. Thus, the player does not consider the game history and chooses the target on the opponent’s court randomly. This randomized policy enables the player to return the ball successfully. However, this policy would not suffice to win against an intermediate player. Therefore, we have to choose the goal parameters according to the position and actions of the opponent. To realize this aim, a model of the opponent’s strategy is necessary. From the observed game play, the robot player may be able to learn opponent’s preferences and how to prepare for the opponent’s return.

Optimizing the Movement Generation. Since the robot is limited in its accelerations abilities, switching between fore- and backhand while playing on the real system is often not possible. Predicting the expected hitting area based on the movements of the opponent would give the real system additional time to adjust its motion generation and to switch from fore- to backhand.

Estimating Spin. As discussed in the analysis of accuracy of the ball dynamics model, the inclusion of a spin estimate (especially sidespin) could improve the performance of the robot. However, spin cannot be determined from the few pixels in which the ball is observed by the camera. Even worse, the number of frames is insufficient to estimate spin based on the ball's trajectory. Hence, using only vision information of the ball would not allow us to receive a good model of the present spin. Opponent modeling could enable the system to observe the opponent's arm and racket movements and to use this data to predict the expected spin.

Conclusion. In this article, we have presented a novel table tennis setup which is based on biological assumptions on human movement generation. Our setup, with an anthropomorphic seven DoF robot arm and a human-inhabited environment, is significantly more challenging than the tailored ones of previous robot table tennis players. The movement structure and initiation is based on a hypothesis of the movement structure of expert table tennis players by Ramanantsoa & Durey (1994). The resulting biomimetic player structures the table tennis stroke into four stages and uses a virtual hitting point and pre-shaping of the orientation to parameterize the goal. The redundancy of the arm is solved by minimizing the distance to a defined comfortable hitting posture. This concept has proven to be successful in operation and to produce human-like hitting motions. We demonstrated that the biomimetic player can be used as an explicit policy for returning incoming table tennis balls to a desired point on the opponent's court in a physically realistic simulation as well as on the real Barrett WAM™ robot. A video can be found at <http://www.robot-learning.de/Research/BiomimeticPlayer>.

6 Acknowledgments

The project receives funding from the European Community's Seventh Framework Programme under grant agreement no. ICT-270327 CompLACS.

Appendix

Fifth order Polynomials

A fifth order polynomial is given by

$$\theta_k = \sum_{l=0}^5 \alpha_{kl} t^l, \quad (11)$$

where $\alpha_k = [\alpha_{k0}, \alpha_{k1}, \alpha_{k2}, \alpha_{k3}, \alpha_{k4}, \alpha_{k5}]^T$ are adjustable parameters and k denotes the DoF. The boundary conditions for the joint positions, velocities, and accelerations at the time points t_i and t_f are given by

$$\theta_k(t_i) = p_i, \quad \dot{\theta}_k(t_i) = v_i, \quad \ddot{\theta}_k(t_i) = a_i, \quad (12)$$

$$\theta_k(t_f) = p_f, \quad \dot{\theta}_k(t_f) = v_f, \quad \ddot{\theta}_k(t_f) = a_f, \quad (13)$$

where p_i, v_i, a_i, p_f, v_f and a_f are the joint angles, velocities, and accelerations at the time points t_i and t_f , respectively. With the linear equation system $\mathbf{M}\alpha = \mathbf{b}$ given by

$$\underbrace{\begin{bmatrix} 1 & t_i & t_i^2 & t_i^3 & t_i^4 & t_i^5 \\ 0 & 1 & 2t_i & 3t_i^2 & 4t_i^3 & 5t_i^4 \\ 0 & 0 & 2 & 6t_i & 12t_i^2 & 20t_i^3 \\ 1 & t_f & t_f^2 & t_f^3 & t_f^4 & t_f^5 \\ 0 & 1 & 2t_f & 3t_f^2 & 4t_f^3 & 5t_f^4 \\ 0 & 0 & 2 & 6t_f & 12t_f^2 & 20t_f^3 \end{bmatrix}}_{\mathbf{M}(t_i, t_f)} \underbrace{\begin{bmatrix} \alpha_{k0} \\ \alpha_{k1} \\ \alpha_{k2} \\ \alpha_{k3} \\ \alpha_{k4} \\ \alpha_{k5} \end{bmatrix}}_{\alpha_k} = \underbrace{\begin{bmatrix} p_i \\ v_i \\ a_i \\ p_f \\ v_f \\ a_f \end{bmatrix}}_{\mathbf{b}}, \quad (14)$$

we can efficiently solve for α using Gauss-Seidel elimination (Press et al., 2007).

Acceleration-minimizing Fifth Order Polynomials

For table tennis, the trajectories resulting from the 5th order polynomials can result in high accelerations that exceed the acceleration limits of the robot. Hence, we would like to find a final acceleration value a_f for the hitting stage such that the maximal acceleration is minimized. Therefore, we apply the following constraints to the trajectory

$$\theta_k(t_i) = p_i, \quad \dot{\theta}_k(t_i) = v_i, \quad \ddot{\theta}_k(t_i) = a_i, \quad (15)$$

$$\theta_k(t_f) = p_f, \quad \dot{\theta}_k(t_f) = v_f, \quad \ddot{\theta}_k(t_j) = 0, \quad (16)$$

where $\ddot{\theta}_k(t_j)$ is the jerk and t_j is an arbitrary time point with $t_i < t_j < t_f$ at which the acceleration is maximal. Given these constraints we can compute the coefficients α_{k0} to α_{k5} . Evaluating $\partial \ddot{\theta}(t) / \partial t_j = 0$, yields the solutions

$$t_j^1 = -\frac{(\sqrt{6}-4)t_i + (\sqrt{6}-6)t_f}{10}, \quad (17)$$

$$t_j^2 = \frac{(\sqrt{6}+4)t_i + (6-\sqrt{6})t_f}{10}. \quad (18)$$

The expression for t_j^1 corresponds to the maximum near to the goal while t_j^2 corresponds to the maximum near the start position. The first solution for t_j is the one of interest, since updating the trajectory at the end of the stage can cause high acceleration peaks at this point. The resulting acceleration that minimizes the acceleration at $t_j^1 = -0.1((\sqrt{6}-4)t_i + (\sqrt{6}-6)t_f)$ is given by

$$a_f = \frac{-(2\sqrt{6}-3)(t_i - t_f)v_i + ((4\sqrt{6}-1)(t_i - t_f)v_f + (4-6^{\frac{3}{2}})(p_i - p_f))}{(\sqrt{6}+1)(t_i - t_f)^2}. \quad (19)$$

This final acceleration does not just minimize the maximal acceleration but also the position overshoot at the end of the trajectory.

References

- Abbs, J., & Cole, K. (1987). Neural mechanisms of motor equivalence and goal achievement. In S. Wise (Ed.), *Higher Brain Functions: Recent Explorations of the Brain's Emergent Properties* (pp. 15–43). Hoboken, NY: John Wiley & Sons.
- Acosta, L., Rodrigo, J., Mendez, J., Marchial, G., & Sigut, M. (2003). Ping-pong player prototype. *IEEE Robotics and Automation Magazine*, 10(4), 44–52.
- Alexander, R. (1997). A minimum energy cost hypothesis for human arm trajectories. *Biological Cybernetics*, 76(2), 97–105.
- Andersson, R. (1988). *A robot ping-pong player: experiment in real-time intelligent control*. Cambridge, MA, USA: MIT Press.
- Angel, L., Sebastian, J., Saltaren, R., Aracil, R., & Gutierrez, R. (2005). Robotenis: Design, dynamic modeling and preliminary control. In *Proc. Int. Conf. Advanced Intelligent Mechatronics* (pp. 747–752).
- Bernstein, N. (1967). *The coordination and regulation of movements*. Pergamon, Oxford.
- Billingsley, J. (1983). Machineroe joins new title fight. *Practical Computing*, May/June, 14–16.
- Bootsma, R., & Wieringen, P. van. (1988). Visual control of an attacking forehand drive in table tennis. In *Complex movement behaviour: The motor-action controversy* (pp. 189–199). North-Holland.
- Bootsma, R., & Wieringen, P. van. (1990). Timing an attacking forehand drive in table tennis. *Journal of Experimental Psychology: Human Perception and Performance*, 16(1), 21–29.
- Bryson, A., & Ho, Y. (1975). *Applied optimal control*. Wiley, New York.
- Craig, J. (1989). *Introduction to robotics. mechanisms and control*. Reading, Massachusetts: Addison-Wesley Publishing Company, Inc.
- Cruse, H. (1986). Constraints for joint angle control of the human arm. *Biological Cybernetics*, 54(2), 125–132.
- Cruse, H., Brüwer, M., Brockfeld, P., & Dress, A. (1990). On the cost functions for the control of the human arm movement. *Biological Cybernetics*, 62, 519–528.
- Dennis, J., & Schnabel, R. (1983). *Numerical methods for unconstrained optimization and nonlinear equations*. Englewood Cliffs, NJ: Prentice-Hall.
- Elliott, D., Binsted, G., & Heath, M. (1999). The control of goal-directed limb movements. *Human Movement Science*, 18.
- Fässler, H., Beyer, H. A., & Wen, J. T. (1990). A robot ping pong player: optimized mechanics, high performance 3d vision, and intelligent sensor control. *Robotersysteme*, 6(3), 161–170.
- Fitts, P. (1954). The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology*, 47(6), 381–391.
- Flash, T., & Hogan, N. (1985). The coordination of arm movements: an experimentally confirmed mathematical model. *Journal of Neurosciences*, 5(7), 1688–1703.
- Harris, C., & Wolpert, D. (1998). Signal-dependent noise determines motor planning. *Nature*, 394, 780–784.
- Harrison, R., Mackenzie, J., Morris, B., & Springett, J. (2005). *Design and build of a robotic ping pong player* (Tech. Rep.). The University of Adelaide.
- Hartley, J. (1987). Toshiba progress towards sensory control in real time. *The Industrial robot*, 14(1), 50–52.

- Hashimoto, H., Ozaki, F., Asano, K., & Osuka, K. (1987). Development of a ping pong robot system using 7 degrees of freedom direct drive. In *Industrial applications of robotics and machine vision (IECON)* (pp. 608–615).
- Hecht, H. (2004). *Time-to-contact*. Amsterdam, The Netherlands: Elsevier Science.
- Henry, F., & Rogers, D. (1960). Increased response latency for complicated movements and the memory drum theory of neuromotor reaction. *Research Quarterly*, 31, 448–458.
- Hogan, N. (1984). An organizing principle for a class of voluntary movement. *Journal of Neuroscience*, 4(11), 2745–2754.
- Hubbard, A., & Seng, C. (1954). Visual movements of batters. *Research Quarterly*, 25, 42–57.
- Keele, S. (1968). Movement control in skilled motor performance. *Psychological Bulletin*, 70(6), 387–403.
- Knight, J., & Lowery, D. (1986). Pingpong-playing robot controlled by a microcomputer. *Microprocessors and Microsystems*, 10(6), 332–335.
- Kuo, A. (2005). Harvesting energy by improving the economy of human walking. *Science*, 309(5741), 1686–1687.
- Lee, D., & Young, D. (1985). Visual timing of interceptive action. In D. Ingle, M. Jeannerod, & D. Lee (Eds.), *Brain mechanisms and spatial vision* (pp. 1–30). Dordrecht, Netherlands: Martinus Nijhoff.
- Matsushima, M., Hashimoto, T., Takeuchi, M., & Miyazaki, F. (2005). A learning approach to robotic table tennis. *IEEE Trans. on Robotics*, 21, 767–771.
- Miyazaki, F., Matsushima, M., & Takeuchi, M. (2005). Learning to dynamically manipulate: A table tennis robot controls a ball and rallies with a human being. In *Advances in robot control* (pp. 3137–341). Springer.
- Morasso, P. (1981). Spatial control of arm movements. *Exp Brain Res*, 42(2), 223–227.
- Mülling, K., Kober, J., & Peters, J. (2010). Simulating human table tennis with a biomimetic robot setup. In *From Animals to Animats 11: Eleventh International Conference on the Simulation of Adaptive Behavior* (pp. 273–282).
- Press, W., Teukolsky, S., Vetterling, W., & Flannery, B. (2007). *Numerical recipes: The art of scientific computing* (3th ed.). Cambridge University Press.
- Ramanantsoa, M., & Durey, A. (1994). Towards a stroke construction model. *Int. Journal of Table Tennis Science*, 2(2), 97–114.
- Ripoll, H., & Fleurance, P. (1988). What does keeping one's eye on the ball mean? *Ergonomics*, 31, 1647–1654.
- Rodrigues, S., Vickers, J., & A., W. (2002). Head, eye and arm coordination in table tennis. *Journal of Sport Sciences*, 20, 187–200.
- Roitman, A., Massaquoi, S. G., Takahashi, K., & Ebner, T. (2004). Kinematic analysis of manual tracking in monkeys: characterization of movement intermittencies during a circular tracking task. *Journal of Neurophysiology*, 91(2), 901–911.
- Schaal, S. (2009). *The SL simulation and real-time control software package* (Tech. Rep.). University of Southern California.
- Schmidt, R. (1975). A schema theory of discrete motor skill learning theory. *Psychological Review*, 82(4), 225–260.
- Schmidt, R. (1988). *Motor and action perspectives on motor behavior* (O. Meijer & K. Roth, Eds.). Amsterdam: Elsevier.

- Schmidt, R. (2003). Motor schema theory after 27 years: Reflections and implications for a new theory. *Research Quarterly for Exercise and Sport*, 74(4), 366–379.
- Scott, A., & Fong, E. (2004). *Body structures and functions* (10th ed.). Clifton Park, NY: Delmar Learning.
- Sorenson, H. W. (1985). *Kalman filtering: theory and application*. Los Alamitos, CA: IEEE Press.
- Spong, M., Hutchinson, S., & Vidyasagar, M. (2006). *Robot modeling and control*. Hoboken, NY: John Wiley & Sons, Inc.
- Todorov, E., & Jordan, M. (2002). Optimal feedback control as a theory of motor coordination. *Nature Neuroscience*, 5(11), 1226–1235.
- Tyldesley, D., & Whiting, H. (1975). Operational timing. *Journal of Human Movement Studies*, 1(4), 172–177.
- Uno, Y., Kawato, M., & Suzuki, R. (1989). Formation and control of optimal trajectory in human multijoint movement – minimum torque-change model. *Biological Cybernetics*, 61(2), 89–101.
- Williams, A., & Starkes, J. (2002). Cognitive expertise and performance in interceptive actions. In *Interceptive actions in sport: Information and movement*. New York, NY: Routledge Chapman & Hall.
- Wolpert, D., Miall, C., & Kawato, M. (1998). Internal models in the cerebellum. *Trends in Cognitive Science*, 2, 338–347.
- Woodworth, R. (1899). The accuracy of voluntary movement. *Psychological Review*, 3(13), 1–106.