Simulating Human Table Tennis with a Biomimetic Robot Setup

Katharina Mülling, Jens Kober, and Jan Peters

Max Planck Institute for Biological Cybernetics, Department of Empirical Inference
Spemannstr. 38, 72076 Tübingen, Germany
{katharina.muelling,jens.kober,jan.peters}@tuebingen.mpg.de

Abstract. Playing table tennis is a difficult motor task which requires fast movements, accurate control and adaptation to task parameters. Although human beings see and move slower than most robot systems they outperform all table tennis robots significantly. In this paper we study human table tennis and present a robot system that mimics human striking behavior. 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 is implemented on an anthropomorphic robot arm with 7 degrees of freedom using robotics methods. We verify the functionality of the model both in a physical realistic simulation of an anthropomorphic robot arm and on a real Barrett WAM™.

Key words: biomimetic table tennis, anthropomorphic robot arms

1 Introduction

Table tennis has long fascinated roboticists as a particularly difficult task. The main work on robot table tennis started in 1983 [3] with a robot ping pong competition and ended in 1993 [2, 12, 9, 10, 8] when the competition came to an end, but single groups continued work until today [15, 14, 1]. These early approaches used smart engineering to overcome inherent problems like movement generation, orientation of the racket and vision in an human inhabited environment. Furthermore, they used a much smaller table and modified table tennis rules [3]. In contrast to these approaches, we use an anthropomorphic robot arm with seven degrees of freedoms (DoFs) and concentrate on generating smooth movements that properly distribute the forces over the different DoFs. Therefore, we employ a biomimetic approach for trajectory generation and movement adaptation.

Table tennis requires fast and accurate movements to achieve high playing performance. However, for such quick and forceful movements, the human central nervous system has little time to process feedback about the environment and has to rely largely on feedforward components [21] such as accurate task models as well as predictions about the opponent and the ball. Understanding how humans perform so well in such a complex task as table tennis may yield essential knowledge for skill execution in robotics. In this project, it is our goal to construct a model of table tennis striking movements based on known hypotheses of human motor control in table tennis. We want to get a step closer to understanding which basic building blocks are required for generic robot skill execution systems. We describe the construction of a robot ping pong player, with seven DoFs, that is capable of returning a ball on an International Table Tennis Federation (ITTF) standard sized table served by a ball cannon. We focus particularly on modeling the arm trajectories in striking movements based on human table tennis data using a multi-stage model [16]. We end up with a method that successfully adapts the stroke according to the movement of the ball. The setup works sufficiently well in simulation and on a real Barrett WAM[1].

In this paper, we will proceed as follows. In Section 2, we present all relevant background on modeling a table tennis stroke based on biological hypotheses such that we are able to obtain a model of a table tennis stroke in Section 3. In Section 4, we present the results of our implementation and show that the proposed model works well in simulation and on the real robot.

2 Modeling Striking Movement in Human Table Tennis

In this section, we present background information on modeling table tennis from a racket sports perspective. In particular, we focus on movement stages, motion selection and parameterization, and movement generation. At the end of each of these sections, we will outline which computational concepts arise from the biological hypotheses.

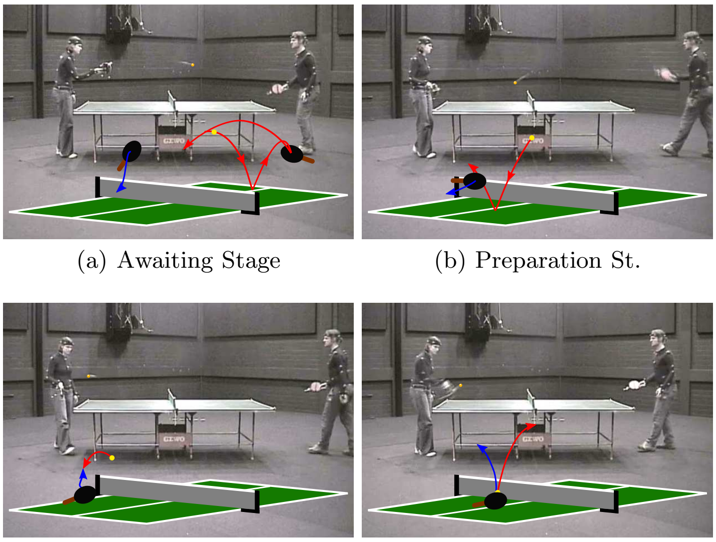
2.1 Movement Stages of a Stroke

Table tennis exhibits a regular, modular structure that has been studied by Ramanantsoa and Durey [16]. They analyzed a top player and proposed a spatial adjustment of four movement stages with respect to certain ball events, i.e., bouncing, net crossing and stroke. According to their hypothesis, the following four stages can be distinguished during playing of experts and, to make them more understandable, we have labeled them according to their functionality:

Awaiting Stage. The ball moves towards the opponent who hits it back towards the net. The racket is moving downwards. At the end of this stage the racket will be in a plane parallel to the table surface.

Preparation Stage. The ball comes 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 is in the same plane as it is in the awaiting phase. For backhand strokes the racket moves on a frontal plane nearly perpendicular to the plane in the awaiting stage. The player chooses a hitting point where he plans to hit the ball to which we refer as the virtual hitting point.

[1] Note, that a preliminary version with no real robot results and a simplified dynamical model has been presented at a German local conference.



(a) Awaiting Stage (b) Preparation St.

(c) Hitting Stage (d) Finishing Stage

Fig. 1. This figure illustrates the four movement stages of Ramanantsoa et al. [16] 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 arrow shows the movement of the ball in the phase and the blue arrow the movement of the racket.

Hitting Stage. The ball moves towards the virtual hitting point where the player intercepts it. In a first substage final adjustments are done. In the second substage the racket moves towards the virtual hitting point until it hits the ball in a circular movement. For expert players the duration of this phase appears to be constant and lasts approximately $80ms$. At the point of impact the lateral velocity (in the direction of the small table side) is zero and the velocity in direction of the long table side reaches its apex.

Finishing Stage. After having been hit, the ball is on the return path to 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 and Durey [16] in a VICON motion capture setup for two intermediate players where each of the stages can be observed distinctively (see Figure 1). From a computational point of view, this model corresponds to a finite state automaton.

2.2 Movement Selection and Goal Determination

As humans appear to rely on elementary motor programs [18], it is likely that pre-structured movement commands are employed for each of the four stages.

These motor programs are adapted to the environmental stimuli at the beginning of each stage. Motor programs determine the order and timing of the muscle contractions and, by doing so, define the shape of the action produced. Sensory information can further modify motor programs to generate rapid corrections in the case of changing environmental demands as found in table tennis [5]. The system is only altering the parameters of the movement such as duration, amplitude, and the final goal position of the movement [18]. This is supported by the experiments in [20], which demonstrated that expert players exhibit a consistent spatial and temporal movement pattern in table tennis. The authors of [20] concluded that a professional player chooses a movement program for which the execution time is known from their repertoire and decides when to initiate the drive. This observation is known as operational timing hypothesis.

The problem of what information is used for initiating the movement is not yet solved. Most likely humans use the so-called *time to contact*, i.e., is the time until an object reaches the observer, to control the timing of their stroke stages. Lee [13] suggested that humans determine the time to contact by an optical variable τ that is specified as the inverse of the relative rate of dilation of a retinal image of an object. Using the operational timing hypothesis, biomimetic system has to initiate the chosen movement program when τ reaches a critical value.

We represent one set of movement programs for a specific forehand as splines. The start and end position, velocity and acceleration of the stages as well as the durations of the movements are given by pre-defined values which are fixed while the end and start conditions of the hitting and finishing stage, respectively, can be selected freely. Here we use the hitting point which is adapted according to the incoming ball and the desired return.

2.3 Movement Generation

Assuming that movement stages, selection and initiation are known, we need to discuss how the different strokes are generated. There are infinitely many ways to generate racket trajectories and, due to redundancies in the arm, there are also numerous different arm posture to execute the same task-space trajectory in joint-space. In order to find generative principles underlying the movement generation, neuroscientists often turn to optimal control [19]. One approach is the use of cost functions which allow the computation of trajectory formation for arm movements. Most cost functions focus primarily on reaching and pointing movements where one can observe a bell-shape velocity curve as well as a clear relationship between movement duration and amplitude. However, this does not hold for striking sports. Cruse et al. [6] suggested a cost function for the control of the human arm movement based on the comfort of the posture. For each joint, the cost is induced by proximity to a comfort posture in joint-space, i.e., the cost is minimal if the joint angles are the same as for the comfort posture and increases with the distance between comfort posture and joint position. For movement generation, this cost is minimized. We employ this cost function to select a comfortable joint configuration at the hitting point (see Section 3.3).

3 A biologically-inspired Trajectory Generator for Table Tennis Strokes

In this section, we will discuss how the parts of the behavioral model presented in Section 2 can be implemented as a mathematical model suitable for real-time execution on a robot. For doing so, we proceed as follows: first, we present all required components in an overview. Subsequently, we discuss the details of the dynamics model for table tennis in Section 3.2, the computation of the goal parameters in Section 3.3 and the trajectory generation in Section 3.4.

3.1 General Assumptions

As outlined in Section 2.1, we assume the movement stages of the model by Ramanantsoa et al. [16] and use a finite state automaton to represent this model. In order to realize each of these four stages, the system has to detect the ball and determine its position $\mathbf{p}_b$. Due to noise in the vision processing, the system needs to filter this information.

To generate the arm trajectories, we have to determine the constraints for the movements of each joint of the arm in each stage. While desired final joint configurations suffice for the awaiting, preparation and finishing stages, the hitting stage requires a well-chosen movement goal which is the hardest to realize. The system has to first choose a point on the court of the opponent where the ball needs to be returned[2]. Secondly, we have to determine the intersection point of the ball and the racket, which specify the virtual hitting point $\mathbf{p}_e$. The hitting point is determined by the location where the ball trajectory intersects a virtual hitting plane in the forehand area of the robot. Based on the choice of these two points, the necessary batting position, orientation and velocity of the racket are chosen as goal parameters for the hitting movement. More details on the computations involved are given in Section 3.3.

Movement initiation is triggered in accordance with the movement stages and using the movement goals, i.e., when the time of the predicted ball intersecting the virtual hitting point $\boldsymbol{p}_\mathrm{e}$ is less than a threshold, the hitting movement is initiated. This step requires the system to predict when the ball is going to reach the virtual hitting plane. The current hitting time can be determined by predicting the trajectory of the ball using the physical model of the aerodynamic and bouncing behavior of the ball described in Section 3.2. Following the suggestion in [4] that some online adaptation of the movement can take place, we update the virtual hitting point if the estimates changes drastically. For the determination of the movement program, we rely upon a spline-based representation for encoding the trajectory. More details are given in Section 3.4.

[2] Humans choose this point as part of a higher level strategy. To date, we choose them in an ad-hoc fashion not conditioned on the opponent.

3.2 Dynamics Model

To predict the position and velocity of the ball at time t_1 based on the ones at time t_0, we have to model the aerodynamics of the ball and the physics of a ball's bounce off of a table. For modeling the ballistic flight of the ball we have to consider air drag, gravity and spin. As the latter is hard to determine, our model currently neglects the spin. For a table tennis ball we can assume that the air drag is proportional to the square of the velocity of the ball. Using symplectic Euler integration, we can implement the following model in discrete time form:

$$\boldsymbol{a_k} = \boldsymbol{g} - C\|\boldsymbol{v_k}\|\boldsymbol{v_k} \qquad \boldsymbol{v_{k+1}} = \boldsymbol{v_k} + \boldsymbol{a_k}\Delta t \qquad \boldsymbol{p_{k+1}} = \boldsymbol{p_k} + \boldsymbol{v_{k+1}}\Delta t, \tag{1}$$

where $\boldsymbol{p}$ denotes the position of the ball, $\boldsymbol{v}$ is the velocity, $\boldsymbol{a}$ denotes the acceleration, $\boldsymbol{g} = -9.81m/s^2[0,0,1]^T$ is the gravity, $C = c_w\rho A/(2m)$, c_w is the drag coefficient, ρ is the density of the air, A is the size of the ball surface and m is the mass of the table tennis ball.

For the bouncing behavior of the ball we assume a velocity change in z-direction only. This change in velocity $v_z = -\varepsilon_T v_z$ is determined by the coefficient of restitution ε_T.

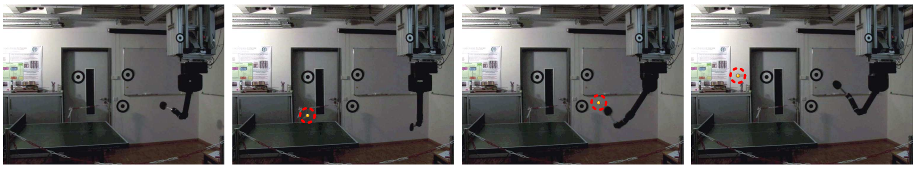
3.3 Determining the Goal Parameters

After determining the virtual hitting point, the system can freely choose the height z_{net} at which the returning ball passes the net as well as the positions x_{b}, y_{b} where the ball will bounce on the opponents courts. The y-axis is along the net and the x-axis is aligned with the long side of the table. The choice of these three variables belongs to the higher level functionality and is not covered in this model, we instead draw them from a distribution of plausible values. To determine the goal parameters, we have to first calculate the desired outgoing velocity vector $\boldsymbol{O}$ of the ball which corresponds to the desired velocity of the ball after the impact with the racket. Directly from it, we can also determine the required velocity and orientation of the racket.

Desired Outgoing Vector. Based on the dynamics model derived in Section 3.2, we obtain 5 non-linear equations with 5 unknowns, i.e., the time until the ball reaches the opponents court, the time until the ball reaches the net and the desired outgoing vector (3 components). Since these equations are non linear in the variables of interests, we have to solve the problem numerically. Therefore, we need to use a globally convergent solver for nonlinear equation systems, which combines the Newton-Raphson update with a modification for global convergence [7].

Goal Orientation. The orientation of the end-effector is specified as a rotation that transforms the normal vector $\boldsymbol{n}_e$ to the desired normal vector $\boldsymbol{n}_{\text{ed}}$ given by

$$\boldsymbol{n}_{\text{rd}} = \frac{\boldsymbol{O} - \boldsymbol{I}}{\|\boldsymbol{O} - \boldsymbol{I}\|}, \tag{2}$$



(a) Awaiting Stage (b) Preparation St. (c) Hitting Stage (d) Finishing Stage

Fig. 2. The figure shows the different phases on the real robot. Note that the black circles are part of our safety system and have nothing to do with the task.

where $\boldsymbol{O}$ is the velocity of the outgoing ball after the ball-racket impact and $\boldsymbol{I}$ is the velocity vector of the incoming ball at the virtual hitting point before impact. Note that we assume only a speed change $\boldsymbol{O}-\boldsymbol{I}$ in the normal direction $\boldsymbol{n}$. The rotation is defined in terms of quaternions by

$$q_{\mathrm{ed}'} = q_{\mathrm{rd}} q_{\mathrm{yrot}}, \tag{3}$$

where q_{yrot} is the quaternion that describes the rotation from the racket to the end-effector and $q_{\mathrm{rd}} = (\cos(\theta/2), \boldsymbol{u}\sin(\theta/2))$, with $\theta = \boldsymbol{n}_{\mathrm{e}}^T \boldsymbol{n}_{\mathrm{rd}}/(\|\boldsymbol{n}_{\mathrm{e}}\|\|\boldsymbol{n}_{\mathrm{rd}}\|)$ and $\boldsymbol{u} = \boldsymbol{n}_{\mathrm{e}} \times \boldsymbol{n}_{\mathrm{rd}}/\|\boldsymbol{n}_{\mathrm{e}} \times \boldsymbol{n}_{\mathrm{rd}}\|$, is the quaternion that defines the transformation of the normal of the end-effector $\boldsymbol{n}_e$ to the desired racket normal $\boldsymbol{n}_{\mathrm{rd}}$. As there exist infinitely many racket orientations that have the same racket normal, we need to determine the final orientation depending on a preferred end-effector position. The resulting quaternion of the end-effector q_{ed} is determined by the rotation about the normal of the racket. The orientation with the corresponding joint values is chosen to yield the minimum distance to the comfort position in joint space is used as a desired racket orientation.

Required Racket Velocity. Next we have to calculate the velocity vector for the end-effector at the time of the ball's interception. We can describe the relation between the components of the incoming and ingoing velocity vector parallel to the racket norm using

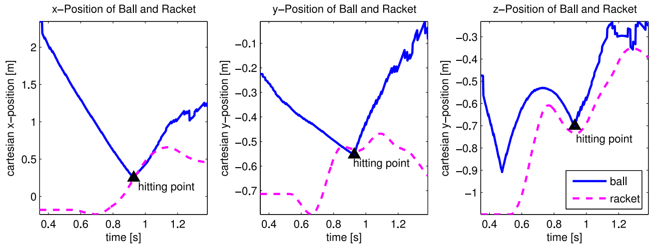
$$O_{||} - v = \varepsilon_R(-I_{||} + v), \tag{4}$$

where ε_{R} denotes the coefficient of restitution of the racket and v the speed of the racket along its normal. This equation can be solved for v yielding the desired racket velocity.

3.4 Trajectory Generation

For the execution of the movements, we need a representation to obtain position $\boldsymbol{q}(t)$, velocity $\dot{\boldsymbol{q}}(t)$ and accelerations $\ddot{\boldsymbol{q}}(t)$ of the joints of the manipulator at each point in time t so that it can be executed with an inverse dynamics based controller. We used fifth order polynomials $\boldsymbol{q} = \sum_{j=0}^{5} \boldsymbol{a}_j t^j$ to represent the trajectory of all stages. Such polynomials are the minimal sufficient representation,

Fig. 3. This figure shows the movement of the racket and the ball on the real robot for one stroke movement. The hitting point is indicated by the black triangle.



generate smooth trajectories and can be evaluated quickly as well as easily. Applying the four stage model of Ramanantsoa et al. [16], we can determine four different spline phases consisting of splines interpolating between fixed initial and final positions. As the trajectory of the hitting and finishing state depends on the hitting point, trajectories have to be calculated jointly at the beginning of the hitting stage and have to be recalculated every time the virtual hitting point is updated.

4 Evaluations

In this section, we demonstrate that the presented biomimetic robot table tennis model can be used effectively in a setup where the ball is served by a ball cannon. Firstly, we present the simulated setup for the table tennis task. Secondly, we implement the model on a real robot.

We employ a Barrett WAM arm with seven DoFs that is capable of high speed motion. A standard table tennis racket is attached to the end-effector. The robot arm interacts with a standard sized table and a table tennis ball according to the ITTF rules. The ball is served randomly by a ball cannon to the right half of the table. This range corresponds roughly to an area of $1m^2$. The virtual hitting point is determined as the intersection point of the ball and the virtual hitting plane discussed in Section 3 (it covers the whole $1m^2$). The ball is tracked using a stereo vision system with a sampling rate of 60 frames per second and the vision information is filtered using an extended Kalman Filter based on the dynamics model described in Section 3.2

4.1 Simulated Setup

We employed the SL framework [17] to create a simulation of an anthropomorphic robot arm. Subsequently, we used a model of the flight and the bouncing behavior of the ball as described in Section 3.2. We model the noise and delay of the vision system. The coefficients of restitution of both racket-ball and ball-table interactions were determined in a VICON setup.

The table tennis system is capable of returning an incoming volley to the opponents court which was served by a ball cannon at random times and to randomly selected positions. In an evaluation setup where the ball cannon served the ball 10,000 times to a random position in the work-space of the robot, the

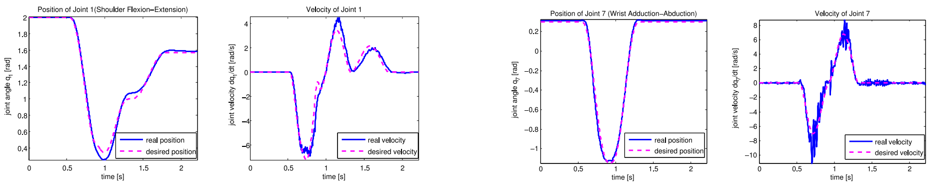


Fig. 4. This figure shows the trajectories for representative joint positions and velocities for one stroke movement. Note that the tracking errors are often due to low-gain control.

system was able to return 98% of the balls. In 75% of the trials the ball was returned to the opponent's court. The mean distance of the position of the racket mid point from the ball at the moment of contact is $1.8\,cm$. This result could be further improved by optimizing the trajectory generation in joint space.

4.2 Application on a Barrett WAM^{TM}

We have subsequently set up the same framework on a real robot using two partially overlapping stereo-setups for visual input.We are going to detail the arising differences here. An extended Kalman filter, based on a ballistic flight model with estimated restitution factors, tracks the ball well. However, the prediction of the virtual hitting point and time is less accurate due to unobserved spin and an underestimated initial velocity of the ball. These predictions are updated frequently and the trajectory generation is adapted. Nevertheless, the robot manages to hit the ball reliably. The main problem for missing balls and underestimating the velocity of the ball up to now is the limited field of view of the camera setup. See Figure 3 for the trajectories of the racket and the ball of the real system, Figure 4 for trajectories of individual joints and Figure 2 for snapshots of the movement.

5 Conclusion

Using knowledge on human table tennis, we have created a biomimetic model for striking movements. This model is realized in a computational form. We have shown that the resulting model can be used as an explicit policy for returning incoming table tennis balls to the opponent's court using a real seven DoF Barrett WAM. Our setup, with an anthropomorphic arm and a cluttered environment, is significantly more challenging than the tailored ones of previous robot table tennis players. The biomimetic model with its four stages of the stroke and the goal parameterization using virtual hitting points and pre-shaping of the orientation has proven to be successful in operation.

Our future work will concentrate on improving the precision in returning the ball to a desired point on the table and to improve the transition between fore- and backhand. Furthermore, we plan to replace the spline based trajectory for movement generation by dynamic systems motor primitives [11] for each of the four stages suggested by Ramanantsoa.

References

1. Acosta, L., Rodrigo, J., Mendez, J., Marchial, G., Sigut, M.: Ping-pong player prototype. Robotics and Automation magazine **10** (2003) 44–52
2. Andersson, R.: A robot ping-pong player: experiment in real-time intelligent control. MIT Press, Cambridge, MA, USA (1988)
3. Billingsley, J.: Robot ping pong. Practical Computing (1983)
4. Bootsma, R., van Wieringen, P.: Visual control of an attacking forehand drive in table tennis. In: Complex Movement Behaviour: The Motor-Action Controversy. North-Holland (1988) 189–199
5. Bootsma, R., van Wieringen, P.: Timing an attacking forehand drive in table tennis. Journal of Experimental Psychology: Human Perception and Performance **16** (1990) 21–29
6. Cruse, H., Brüwer, M., Brockfeld, P., Dress, A.: On the cost functions for the control of the human arm movement. Biological Cybernetics **62** (1990) 519–528
7. Dennis, J., Schnabel, R.: Numerical Methods for Unconstrained Optimization and Nonlinear Equations. Prentice-Hall, Englewood Cliffs, NJ (1983)
8. Fässler, H., Beyer, H.A., Wen, J.T.: A robot ping pong player: optimized mechanics, high performance 3d vision, and intelligent sensor control. Robotersysteme **6** (1990) 161–170
9. Hartley, J.: Toshiba porgress towards sensory control in real time. The Industrial robot **14-1** (1987) 50–52
10. Hashimoto, H., Ozaki, F., Asano, K., Osuka, K.: Development of a ping pong robot system using 7 degrees of freedom direct drive. In: Industrial applications of Robotics and machine vision (IECON). (1987) 608–615
11. Ijspeert, A.J., Nakanishi, J., Schaal, S.: Learning attractor landscapes for learning motor primitives. In: Advances in Neural Information Processing Systems 16 (NIPS). Volume 15., Cambridge, MA, MIT Press (2003) 1547–1554
12. Knight, J., Lowery, D.: Pingpong-playing robot controlled by a microcomputer. Microprocessors and Microsystems **10**(6) (1986) 332 –335
13. Lee, D., Young, D. In: Visual timing of interceptive action. Dordrecht, Netherlands: Martinus Nijhoff (1985) pp. 1–30
14. Matsushima, M., Hashimoto, T., Takeuchi, M., Miyazaki, F.: A learning approach to robotic table tennis. IEEE Trans. on Robotics **21** (2005) 767 – 771
15. Miyazaki, F., Matsushima, M., Takeuchi, M.: Learning to dynamically manipulate: A table tennis robot controls a ball and rallies with a human being. In: Advances in Robot Control. Springer (2005) 3137–341
16. Ramanantsoa, M., Durey, A.: Towards a stroke construction model. International Journal of Table Tennis Science **2** (1994) 97–114
17. Schaal, S.: The SL simulation and real-time control software package. Technical report, University of Southern California (2009)
18. Schmidt, R., Wrisberg, C.: Motor Learning and Performance. Second edn. Human Kinetics (2000)
19. Todorov, E.: Optimality principles in sensorimotor control. Nature Neuroscience **7** (2004) 907–915
20. Tyldesley, D., Whiting, H.: Operational timing. Journal of Human Movement Studies **1** (1975) 172–177
21. Wolpert, D., Miall, C., Kawato, M.: Internal models in the cerebellum. Trends in Cognitive Science **2** (1998) 338 – 347