A Biomimetic Approach to Robot Table Tennis

Katharina Mülling, Jens Kober, Jan Peters

***Abstract*— Although human beings see and move slower than table tennis or baseball robots, they manage to outperform such robot systems. One important aspect of this better performance is the human movement generation. In this paper, we study trajectory generation for table tennis from a biomimetic point of view. Our focus lies on generating efficient stroke movements capable of mastering variations in the environmental conditions, such as changing ball speed, spin and position. We study table tennis from a human motor control point of view. To make headway towards this goal, we construct a trajectory generator for a single stroke using the discrete movement stages hypothesis and the virtual hitting point hypothesis to create a model that produces a human-like stroke movement. We verify the functionality of the trajectory generator for a single forehand stroke both in a simulation and using a real Barrett WAM™.**

I. INTRODUCTION

Table tennis has long fascinated roboticists as a particularly difficult task. The main work on robot table tennis started in 1983 [1] and ended in 1993 [2]–[6], but single groups continued work until today [7]–[9] (see Section II for a more detailed review). These early approaches used smart engineering to overcome inherent problems like movement generation, orientation of the racket and vision in an human inhabited environment. 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.

Humans perform complex skills relying on little feedback with long latencies, have strong limits on their possible execution, and have chronically inaccurate sensory information on largely unmodeled environments. Table tennis requires fast and accurate movements to achieve a decent playing performance. Understanding how humans perform a complex task such as table tennis can yield essential knowledge for skill execution and learning in robotics.

In this project, we construct a trajectory generator for table tennis striking movements based on known hypotheses on human motor control in table tennis. Our goal is to get a step closer to understanding which basic building blocks are needed for generic robot skill execution systems. We describe the construction of a robot ping pong player 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 stage model [10]. We end up with a method that works sufficiently well in simulation and on a real Barrett WAM™.

In this paper, we will proceed as follows. In Section III, we present all relevant 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 IV. In Section V, we present the results of the simulation and of our real setup.

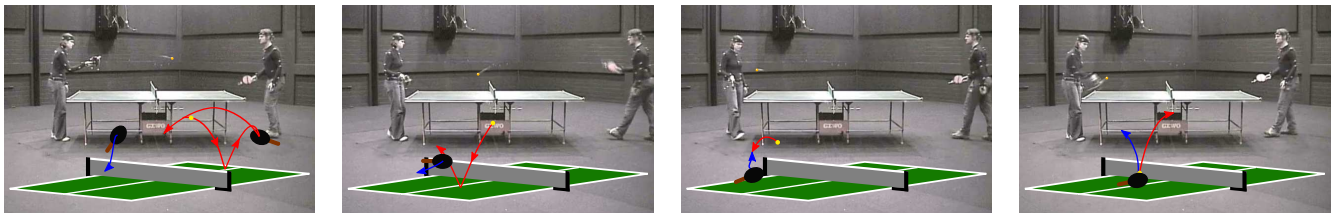
All authors are with the Department of Empirical Inference, Max Planck Institute for Biological Cybernetics, Spemannstr. 38, 72076 Tübingen, Germany, email: firstname.lastname@tuebingen.mpg.de

II. LITERATURE REVIEW

Work on robot table tennis started with the robot table tennis competitions initiated by Billingsley in 1983 [1]. Several early systems were presented by Knight & Lowery [3], Hartley [4], Hashimoto [5] and others. For this early work, the major bottleneck was fast real-time vision. An important breakthrough was achieved in 1988 by Andersson at AT&T Bell Laboratories who presented the first robot ping pong player capable of playing against humans and machines [2]. Andersson and his team employed the simplified robot table tennis rules suggested by Billingsley[1]. Andersson used a high-speed video system and a six degree of freedom (DoF) PUMA 260 arm with a $0.45\,\mathrm{m}$ long stick mounted on it. In 1993, the last robot table tennis competitions took place and was won by Fässler et al. [6] of the Swiss Federal Institute of Technology. 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.

Nevertheless, interest in robot table tennis did not wane and a series of groups continued work on robot table tennis. Acosta et al. [9] constructed a low-cost robot showing that a setup with two paddles can already suffice for playing if the ball is just reflected at the correct angle by a stationary paddle. Miyazaki et al. [7], [8] 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.

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 problem of generating smooth movements that properly distribute the forces over the different DoFs of the arm was often avoided. In our setup, the task is more difficult as the robot does not have linear axes but has to deal with large inertia; the wrist adds roughly $2.5\,\mathrm{kg}$ weight at the elbow. Thus, we have severe constraints



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

Fig. 1. This figure illustrates the four movement stages of Ramanantsoa et al. [10] recorded in a VICON^{TM} motion capture system for verification during our study. The red arrow shows the movement of the ball in the phase and the blue arrow indicates the movement of the racket.

in the joint velocities (the maximal velocity is approximately $10\,\mathrm{rad/s}$) and accelerations. Furthermore, our vision system operates in a cluttered environment.

III. Modeling Human Striking Movement in Table Tennis

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

A. Movement Stages during a Stroke

Table tennis exhibits a very regular, modular structure studied by Ramanantsoa and Durey [10], [11]. They analyzed a top player and proposed a spatial adjustment with respect to certain ball events, i.e., bouncing, net crossing and stroke. According to the hypothesis of Ramanantsoa, the following four stages can be distinguished during expert players' play and, to make them more understandable, we labeled them according to their function:

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 heads towards the player, has already passed the net and will hit the table during this stage. The racket moves backwards in order to prepare the stroke. The player chooses a hitting point where he plans to hit the ball to which we refer as the virtual hitting point.

Hitting Stage. The ball moves towards the virtual hitting point where the player intercept it. The racket moves towards the virtual hitting point until it hits the ball in a circular movement. For expert players the duration of this stage is constant and lasts approximately $80\,\mathrm{ms}$.

Finishing Stage. After having been hit, the ball is on the return path to the opponent while the racket moves upwards to an end position.

[1]In contrast to human ping pong rules, the table is only $0.5\,\mathrm{m}$ wide and $2\,\mathrm{m}$ long. The net has a height of $0.25\,\mathrm{m}$. Wire frames were attached at each end of the table and the net. For a valid shot, the ball has to pass all frames. As a result, the maximum ball speed is slowed down to $10\,\mathrm{m/s}$.

We have tested and verified the stages suggested by Ramanantsoa et al. [10] in a VICON^{TM} motion capture setup for two intermediate players where each of the stages can be observed distinctively (see Figure 1).

B. Movement Selection and Goal Determination

As humans appear to rely on motor programs [12], it is also plausible that pre-structured movement commands are employed for each of the four stages. These motor programs need to be chosen based upon 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 modify motor programs to generate rapid corrections in the case of changing environmental demands as found in table tennis by Bootsma and van Wiering [13]. This observation is strengthened by the supportive evidence of Tildesley and Whiting [14], who showed that expert table tennis players exhibit a consistent spatial and temporal movement pattern. They concluded that a professional player chooses a movement program for which the execution time is known from his movement repertoire and decides when to initiate the drive. This hypothesis is known as operational timing hypothesis.

The problem of which information is used in order to decide when to initiate the movement has not yet been solved. It is likely that humans use the so-called *time to contact*, which is the time until an object reaches the observer, to control the timing. Hence, the operational timing hypothesis implies that humans have to initiate the chosen movement program when the time to contact reaches a critical value.

In our biomimetic player, we represent movement programs using splines. The hitting point is adapted according to the incoming ball and the desired return. All other start and end positions, velocities and accelerations of the stages and the duration of the movements are fixed.

C. Movement Generation

Next, we need to discuss how the different strokes are generated. There are infinitely many ways to generate racket trajectories and, due to the redundancy of the arm, there are also numerous different arm postures to realize the same task-space trajectory in joint-space. In order to find generative principles underlying the movement generation,

neuroscientists often turn to optimal control [15]. One approach is the use of cost functions which allow the computation of trajectory formation for arm movements. Most studies focus primarily on reaching and pointing movements where one can observe a bell-shape velocity curve and a clear relationship between movement duration and amplitude. However, this relationship does not hold for striking sports. Cruse et al. [16] 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 correspondingly to 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 IV-C for details).

IV. A BIOLOGICALLY-INSPIRED TRAJECTORY GENERATOR FOR TABLE TENNIS STROKES

To evaluate and use the behavioral model presented in Section III, we replace the data-driven observations by a computational realization suitable for real-time execution on a robot. To achieve this aim, we proceed as follows: firstly, we discuss all required components in an overview. Subsequently, we discuss the details of the dynamics model for table tennis, the computation of the goal parameters, the movement generation, the vision system and the filtering of the vision information.

A. General Assumptions

As outlined in Section III-A, we assume the movement stages of the model by Ramanantsoa et al. [10] 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 $\mathbf{x}_b$. Due to noise in the vision processing, the system needs to filter this information (see Section IV-F).

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 $\mathbf{x}_{\mathrm{table}}$ 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 $\mathbf{x}_{\mathrm{hp}}$. The hitting point is determined by the location where the ball trajectory intersects a virtual hitting plane in the hitting 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 involved computations are given in Section IV-C.

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

Movement initiation follows the presented movement stages and is triggered when the time t_{hp} of the predicted ball intersecting the virtual hitting plane is less than a threshold. 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 dynamics model of the ball described in Section IV-B. Following the suggestion in [17] that some online adaptation of the movement can take place, we update the virtual hitting point and subsequently the movement generation for the hitting and finishing stage. For the determination of the movement program, we rely upon a spline-based representation for encoding the trajectory. More details are given in Section IV-E. An overview of the resulting algorithm can be found in Algorithm 1.

B. Dynamics Model of the Table Tennis Ball

To predict the position and velocity of the ball at time t_{j+1} based on the ones at time t_j, we have to model the aerodynamics of the ball, and the physics of a ball's bounce off the table. To model the ballistic flight of the ball we have to consider air drag, gravity and spin. As the latter is hard to observe from data, our model currently neglects the spin. For the 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

$$\begin{aligned} \mathbf{a}_{j+1} &= \mathbf{g} - C\|\mathbf{v}_j\|\mathbf{v}_j, \\ \mathbf{v}_{j+1} &= \mathbf{v}_j + \mathbf{a}_{j+1}\Delta t, \\ \mathbf{p}_{j+1} &= \mathbf{p}_j + \mathbf{v}_{j+1}\Delta t, \end{aligned} \tag{1}$$

where $\mathbf{a}$ denotes the acceleration vector of the ball, $\mathbf{v}$ denotes the velocity of the ball, $\mathbf{p}$ denotes the position of the ball, $\mathbf{g}$ is the gravity vector, Δt is the time difference, $C = c_w \rho A/(2m)$, 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 on the table, 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.

C. 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 as we do not attempt to model strategies based on an opponent), we instead draw them from a distribution of plausible values. To determine the goal parameters, we have to first compute the desired outgoing velocity vector $\mathbf{o}$ of the ball which corresponds to the velocity of the ball after the impact with the racket. Directly from this vector, we can determine the required velocity and orientation of the racket.

a) Desired Outgoing Vector: Based on the dynamics model derived in Section IV-B, we obtain 5 nonlinear 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)

$$f(\mathbf{o}, \mathbf{x}_{\mathrm{hp}}, t_{\mathrm{net}}) \quad = \mathbf{x}_{\mathrm{net}}, \tag{2}$$

$$f(\mathbf{o}, \mathbf{x}_{\mathrm{hp}}, t_{\mathrm{tab}}) \quad = \mathbf{x}_{\mathrm{table}}, \tag{3}$$

where $\mathbf{x}_{\mathrm{net}} = [x_{\mathrm{net}}, z_{\mathrm{net}}]^T$, $\mathbf{x}_{\mathrm{table}} = [x_b, y_b, z_b]^T$, x_{net} is the x-position for the net and z_b is the height of the table. 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 [18].

b) Racket Orientation: The orientation of the endeffector is specified as a rotation that transforms the normal vector $\mathbf{n}_{\mathrm{e}}$ to the desired normal vector $\mathbf{n}_{\mathrm{ed}}$. To define $\mathbf{n}_{\mathrm{ed}}$, we have to compute the normal direction of the racket $\mathbf{n}_{\mathrm{rd}}$ that results in the desired outgoing vector $\mathbf{o}$ for the predicted velocity of the ball $\mathbf{i}$ at the hitting point

$$\mathbf{n}_{\mathrm{rd}} = \frac{\mathbf{o} - \mathbf{i}}{\|\mathbf{o} - \mathbf{i}\|}. \tag{4}$$

Here $\mathbf{i}$ is the velocity vector of the incoming ball at the virtual hitting point before impact. Note that we assume only a speed change $\mathbf{o} - \mathbf{i}$ in the normal direction $\mathbf{n}_{\mathrm{rd}}$. The rotation that transforms $\mathbf{n}_{\mathrm{e}}$ to $\mathbf{n}_{\mathrm{rd}}$ is defined in terms of quaternions by

$$q_{\mathrm{ed}} = q_{\mathrm{rd}} q_{\mathrm{yrot}}, \tag{5}$$

where q_{yrot} is the quaternion that describes the rotation from the racket to the endeffector and $q_{\mathrm{rd}} = (\cos(\gamma/2), \mathbf{u}\sin(\gamma/2))$, with $\gamma = \mathbf{n}_{\mathrm{e}}^T \mathbf{n}_{\mathrm{rd}} / (\|\mathbf{n}_{\mathrm{e}}\| \|\mathbf{n}_{\mathrm{rd}}\|)$ and $\mathbf{u} = \mathbf{n}_{\mathrm{e}} \times \mathbf{n}_{\mathrm{rd}} / \|\mathbf{n}_{\mathrm{e}} \times \mathbf{n}_{\mathrm{rd}}\|$, is the quaternion that defines the transformation of the normal of the end-effector $\mathbf{n}_{\mathrm{e}}$ to the desired racket normal $\mathbf{n}_{\mathrm{rd}}$.

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 $\mathbf{n}_{\mathrm{rd}}$. For this purpose, the orientation of the end-effector q_{ed} is rotated about the normal $\mathbf{n}_{\mathrm{rd}}$ of the racket. The corresponding join values, velocities and accelerations are then computed using inverse kinematics. The inverse kinematics problem for the redundant DoFs is solved numerically by minimizing the distance to the comfort posture in joint space while finding the racket position and orientation which coincides with the virtual hitting point $\mathbf{x}_{\mathrm{hp}}$. The orientation whose corresponding joint configuration $\boldsymbol{\theta}_{\mathrm{hp}}$ yields the minimum distance to the comfort position $\boldsymbol{\theta}_{\mathrm{com}}$ is used as a desired racket orientation.

c) 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 outgoing velocity vector parallel to the racket norm using

$$o_{||} - v = \varepsilon_R(-i_{||} + v), \tag{6}$$

where ε_{R} denotes the coefficient of restitution of the racket, v the speed of the racket along its normal and $o_{||}$ and $i_{||}$ denotes the components of $\mathbf{o}$ and $\mathbf{i}$, respectively, which are parallel to the racket normal. This equation can be solved for v which yields the desired racket velocity.

D. Movement Parameters

To perform a hitting movement to return an incoming ball, we have to generate the movement for each of the four stages. As stated in Section IV-A, we determine the start and end position, velocity and acceleration for each of the four stages. The start and end position for the awaiting and preparation stage as well as the start and end configuration of the finishing and hitting stage, respectively, are fixed and chosen to produce a hitting movement similar as exhibited by humans. The corresponding joint velocities and accelerations are set to zero. The start and end position and velocity of the finishing and hitting stage, respectively, are determined by the joint configuration of the hitting point described above and is determined for each stroke individually. The duration of each stage (i.e., t_{as}, t_{ps}, t_{hs}, t_{fs}) are chosen such that the robot is able to execute the movement. The duration of the hitting stage t_{hs} is equal to the estimated time to hit t_{hp}.

E. Movement Generation

We plan our trajectory in joint space, where high velocity movements can be executed more reliably than in workspace. For the execution of the movements, we need a representation to obtain position $\boldsymbol{\theta}(t)$, velocity $\dot{\boldsymbol{\theta}}(t)$ and accelerations $\ddot{\boldsymbol{\theta}}(t)$ 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 $\theta_k = \sum_{l=0}^{5} \alpha_{kl} t^l$, where $\boldsymbol{\alpha}_k = [\alpha_{k0}, \alpha_{k1}, \alpha_{k2}, \alpha_{k3}, \alpha_{k4}, \alpha_{k5}]^T$ are adjustable parameters and k denotes the DoF, to represent the trajectory of all stages. Such polynomials are the minimal sufficient representation, generate smooth trajectories and can be evaluated quickly as well as easily. Applying the four stage model of Ramanantsoa et al. [10], we can determine four different splines interpolating between the 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.

The boundary conditions for the joint positions, velocities and accelerations at the time point t_i and t_f are given by

$$\theta_k(t_i) = p_i \qquad \dot{\theta}_k(t_i) = v_i \qquad \ddot{\theta}_k(t_i) = a_i \tag{7}$$

$$\theta_k(t_f) = p_f \qquad \dot{\theta}_k(t_f) = v_f \qquad \ddot{\theta}_k(t_f) = a_f \tag{8}$$

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.

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: $\mathbf{x}_t, \dot{\mathbf{x}}_t \rightarrow t_{\text{hp}}, \mathbf{x}_{\text{hp}}, \dot{\mathbf{x}}_{\text{hp}}$

  {*Switch Stage*}
  if *FinishingStage* **and** *MovementEnds*
    Switch to *AwaitingStage*
    Compute $\boldsymbol{\alpha}_k = \mathbf{M}^{-1}(0, t_{\text{as}})\mathbf{b}_k^{\text{as}}$ for each DoF k
  else if *AwaitingStage* **and** $t_{\text{hp}} \leq t_{\text{as}} + t_{\text{hs}}$
    Switch to *PreparationStage*
    Compute $\boldsymbol{\alpha}_k = \mathbf{M}^{-1}(0, t_{\text{ps}})\mathbf{b}_k^{\text{ps}}$ for each DoF k
  else if *PreparationStage* **and** $t_{\text{hp}} \leq t_{\text{hs}}$
    Switch to *HittingStage*
  else if *HittingStage* **and** *BallHit*
    Switch to *FinishingStage*
    Compute $\boldsymbol{\alpha}_k = \mathbf{M}^{-1}(0, t_{\text{fs}})\mathbf{b}_k^{\text{fs}}$ for each DoF k
  end if

  {*Update Striking Motion*}
  if *HittingStage*
    Solve with Newton-Raphson for $\mathbf{o}$ using
      $f(\mathbf{o}, \mathbf{x}_{\text{hp}}, t_{\text{net}}) = \mathbf{x}_{\text{net}}$
      $f(\mathbf{o}, \mathbf{x}_{\text{hp}}, t_{\text{table}}) = \mathbf{x}_{\text{table}}$
    Determine joint configuration at hitting point
      $v = o_{||} + \varepsilon_R i_{||}/(1 + \varepsilon_R)$
      $\mathbf{n}_{\text{rd}} = \mathbf{o} - \mathbf{i}/(\|\mathbf{o} - \mathbf{i}\|)$
      $q_{\text{ed}} = q_{\text{rd}} q_{\text{yrot}}$
      Determine optimal rotation about $\mathbf{n}_{\text{rd}}$ by
        $\boldsymbol{\theta}_{\text{opt}} = \arg\min_{\boldsymbol{\theta}_{\text{hp}}} \|\boldsymbol{\theta}_{\text{com}} - \boldsymbol{\theta}_{\text{hp}}\|$
        with Inverse Kinematics: $\mathbf{x}_{\text{hp}}, q_{\text{ed}}, v \rightarrow \boldsymbol{\theta}_{\text{hp}}$
    Compute $\boldsymbol{\alpha}_k = \mathbf{M}^{-1}(0, t_{\text{hs}})\mathbf{b}_k^{\text{hs}}$ 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 ($\boldsymbol{\theta}$, $\dot{\boldsymbol{\theta}}$, $\ddot{\boldsymbol{\theta}}$) with Inverse Dynamics Control.
until user stops program

With the linear equation system $\mathbf{M}\boldsymbol{\alpha} = \mathbf{b}$ given by

$$\underbrace{\begin{bmatrix} 1 & t_i & t_i^2 & t_i^3 & t_i^4 & t_i^5 \\ 0 & 1 & 2t_i & 3t_i^2 & 4t_i^3 & 5t_i^4 \\ 0 & 0 & 2 & 6t_i & 12t_i^2 & 20t_i^3 \\ 1 & t_f & t_f^2 & t_f^3 & t_f^4 & t_f^5 \\ 0 & 1 & 2t_f & 3t_f^2 & 4t_f^3 & 5t_f^4 \\ 0 & 0 & 2 & 6t_f & 12t_f^2 & 20t_f^3 \end{bmatrix}}_{\mathbf{M}(t_i, t_f)} \underbrace{\begin{bmatrix} \alpha_{k0} \\ \alpha_{k1} \\ \alpha_{k2} \\ \alpha_{k3} \\ \alpha_{k4} \\ \alpha_{k5} \end{bmatrix}}_{\boldsymbol{\alpha}_k} = \underbrace{\begin{bmatrix} p_i \\ v_i \\ a_i \\ p_f \\ v_f \\ a_f \end{bmatrix}}_{\mathbf{b}}, \quad (9)$$

we can solve for $\boldsymbol{\alpha}$ by Gauss-Seidel elimination efficiently.

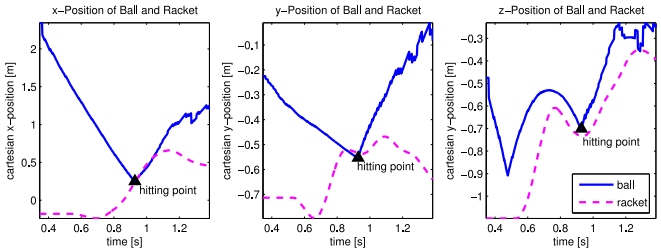


Fig. 2. This figure shows the movement of the racket and the ball on the real robot for a successful striking movement. The hitting point is indicated by the black triangle.

F. Filtering the Vision Information

The vision system consists of a stereo camera setup with two Prosilica GE640C Gigabit Ethernet cameras and a GPU-based 60 Hz blob detection. The vision information $\mathbf{x}_b$ contains the 3D position coordinates of the detected blob. To filter $\mathbf{x}_b$ and to track the table tennis ball, we use an extended Kalman filter (EKF) [19]. The system equations used for the EKF are given in Equation (1).

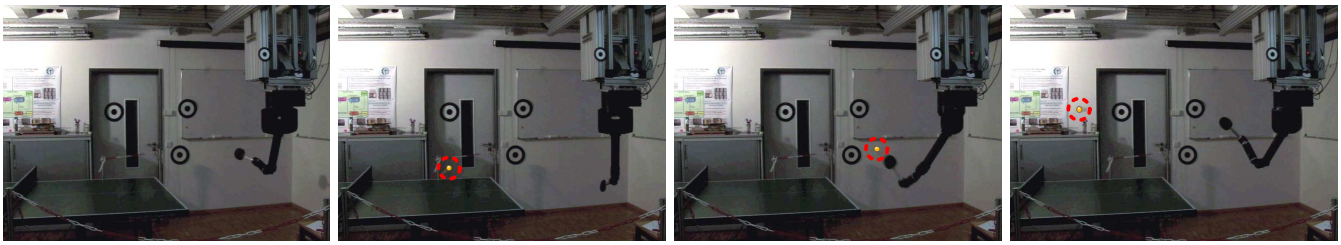
V. Evaluations

In this section, we demonstrate that the biomimetic trajectory generator model can be used effectively for robot table tennis in a ball cannon setup. For this purpose, we will first examine the resulting setup in a simulation of the robot table tennis setup. We discuss the accuracy of the system in striking a ball such that it hits a desired point. As second evaluation, we implement the model on a real robot and demonstrate that it can successfully return balls. We employ a Barrett WAM™ arm with seven DoFs that is capable of high speed motion. A racket with 16 cm in diameter 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 forehand of the robot. As a result, the balls passes the robot's end of the table in an area of approximately 1 m^2. This area serves as the virtual plane.

A. Evaluation in Simulation

We employed the SL framework [20] to create a simulation of an anthropomorphic robot arm. To create the environment, we used a simplified model of the flight and bouncing behavior of the ball as discussed in Section IV-B. We model the noise and delay of the vision system but have not yet included spin. The coefficients of restitution of both racket-ball and ball-table interactions were determined experimentally.

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 arbitrarily chosen positions. In simulations where a ball cannon served the ball 10,000 times to a random position in the work-space of the robot, the system was able to return 98.5% of the balls. In 85% of the trials the ball was returned successfully to the opponent court. The mean deviation of the position of the



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

Fig. 3. The figure shows the different stages, matching those in Figure 1, but performed by the real robot.

racket mid point to the ball at the moment of contact is 1.8 cm. This result could be further improved by optimizing the determination of the outgoing vector and the trajectory generation in joint space.

B. Application on a Barrett WAM^{TM}

Subsequently, we successfully transferred the setup onto a real Barrett WAM robot equipped with two partially overlapping stereo camera pairs. We use the exact same biomimetic trajectory generator as in the simulated setup. As the arising differences were small, we will only focus on these in this discussion. The extended Kalman filter, based on the ballistic flight of a point mass with estimated restitution factors, tracks the ball well. However, the prediction of the virtual hitting point and time is less accurate than in simulation due to the neglected spin and inaccuracies in the vision system. Hence, these predictions need to be updated frequently and the trajectory generation is adapted. As a result, the robot manages to hit the ball. See Figure 3 for snapshots of the movement and Figure 2 for the trajectories of the racket and the ball of the real system.

VI. Conclusion

Using all knowledge on human table tennis available to us, we have formed a trajectory generator for striking movements. This model is realized in a computational form using analytical counterparts. We show that the resulting model can be used as an explicit policy for returning incoming table tennis balls to a desired point of the opponent's court in simulation as well as on the real of a redundant seven DoF Barrett WAM^{TM} robot. 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 table tennis 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 achieving a higher precision in returning the ball to a desired point on the table. Furthermore, we plan to replace the spline based trajectory for movement generation by using motor primitives [21] for each of the four stages suggested by Ramanantsoa.

References

[1] J. Billingsley, "Robot ping pong," *Practical Computing*, 1983.

[2] R. Andersson, *A robot ping-pong player: experiment in real-time intelligent control*. Cambridge, MA, USA: MIT Press, 1988.

[3] J. Knight and D. Lowery, "Pingpong-playing robot controlled by a microcomputer," *Microprocessors and Microsystems*, vol. 10, no. 6, pp. 332 –335, 1986.

[4] J. Hartley, "Toshiba porgress towards sensory control in real time," *The Industrial robot*, vol. 14-1, pp. 50–52, 1987.

[5] H. Hashimoto, F. Ozaki, K. Asano, and K. Osuka, "Development of a ping pong robot system using 7 degrees of freedom direct drive," in *Industrial applications of Robotics and machine vision (IECON)*, 1987, pp. 608–615.

[6] H. Fässler, H. A. Beyer, and J. T. Wen, "A robot ping pong player: optimized mechanics, high performance 3d vision, and intelligent sensor control," *Robotersysteme*, vol. 6, pp. 161–170, 1990.

[7] F. Miyazaki, M. Matsushima, and M. Takeuchi, "Learning to dynamically manipulate: A table tennis robot controls a ball and rallies with a human being," in *Advances in Robot Control*. Springer, 2005, pp. 3137–341.

[8] M. Matsushima, T. Hashimoto, M. Takeuchi, and F. Miyazaki, "A learning approach to robotic table tennis," *IEEE Trans. on Robotics*, vol. 21, pp. 767 – 771, 2005.

[9] L. Acosta, J. Rodrigo, J. Mendez, G. Marchial, and M. Sigut, "Ping-pong player prototype," *Robotics and Automation magazine*, vol. 10, pp. 44–52, 2003.

[10] M. Ramanantsoa and A. Durey, "Towards a stroke construction model," *International Journal of Table Tennis Science*, vol. 2, pp. 97–114, 1994.

[11] A. Durey and F. Orfeuil, "Spins and trajectories in table tennis," in *Table Tennis Scientist's Conference*, 1989.

[12] R. Schmidt and C. Wrisberg, *Motor Learning and Performance*, 2nd ed. Human Kinetics, 2000.

[13] R. Bootsma and P. van Wieringen, "Timing an attacking forehand drive in table tennis," *Journal of Experimental Psychology: Human Perception and Performance*, vol. 16, pp. 21–29, 1990.

[14] D. Tyldesley and H. Whiting, "Operational timing," *Journal of Human Movement Studies*, vol. 1, pp. 172–177, 1975.

[15] E. Todorov, "Optimality principles in sensorimotor control," *Nature Neuroscience*, vol. 7, pp. 907–915, 2004.

[16] H. Cruse, M. Brüwer, P. Brockfeld, and A. Dress, "On the cost functions for the control of the human arm movement," *Biological Cybernetics*, vol. 62, pp. 519–528, 1990.

[17] R. Bootsma and P. van Wieringen, "Visual control of an attacking forehand drive in table tennis," in *Complex Movement Behaviour: The Motor-Action Controversy*. North-Holland, 1988, pp. 189–199.

[18] J. Dennis and R. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Englewood Cliffs, NJ: Prentice-Hall, 1983.

[19] H. W. Sorenson, *Kalman filtering: theory and application*. IEEE Press, 1985.

[20] S. Schaal, "The SL simulation and real-time control software package," University of Southern California, Tech. Rep., 2009.

[21] A. J. Ijspeert, J. Nakanishi, and S. Schaal, "Learning attractor landscapes for learning motor primitives," in *Advances in Neural Information Processing Systems 16 (NIPS)*, vol. 15. Cambridge, MA: MIT Press, 2003, pp. 1547–1554.