2009 IEEE International Conference on Robotics and Automation
Kobe International Conference Center
Kobe, Japan, May 12-17, 2009

Robot Basketball: A Comparison of Ball Dribbling with Visual and Force/Torque Feedback

Georg Bätz, Kwang-Kyu Lee, Dirk Wollherr and Martin Buss
Institute of Automatic Control Engineering
Technische Universität München
D-80290 München, Germany
{georg.baetz,kk.lee,dw,mb}@tum.de

Abstract— **Ball dribbling is a central element of basketball and a main challenge for creating basketball robots is to achieve stability of the periodic dribbling task. In this paper two control designs for ball dribbling with an industrial robot are compared. For the two strategies, the ball position is determined either through force/torque or visual sensor feedback and the ball trajectory is predicted with a recursive least squares algorithm. The end effector trajectory for each dribbling cycle is generated based on the predicted ball position/velocity at the dribbling height and the estimated coefficient of restitution. For both tracking approaches, dribbling for multiple cycles is achieved. The vision-based approach performs better as compared to the force/torque-based approach, in particular for imprecise estimates of the coefficient of restitution.**

I. INTRODUCTION

A. Motivation

The goal of developing personal robots capable of interacting with humans in a natural way poses a number of challenges. One of these challenges is the development of advanced sensory and motor skills: For natural interaction, the robot has to act and react in time periods comparable to humans. Sports, in general, is one of the most challenging domains for robots as it requires fast and highly dexterous manipulation as well as capacious perception of the environment. On this account, several sports have already been investigated by various researchers: soccer as the most prominent example [1], but also dancing [2], table tennis [3] skiing and gymnastics [4] have been the subject of robotics research. To the best of our knowledge, robotic basketball has not been studied in depth so far. However, basketball is an excellent demonstration scenario, in particular for fast and interactive manipulation. It has a truly spatial character, is highly interactive and requires advanced sensory/motor skills as well as a high level of mobility.

In previous work, basic skills required for robotic basketball have been investigated [5]. In this work, two control designs for ball dribbling based on different sensor modalities are presented and evaluated. To equip a robot with the skill of ball dribbling in dynamic environments, both the vision-based and the force/torque-based approach are essential: The use of visual sensors leads to better performance as it allows direct observation of the ball's state. However, due to possible occlusions of the ball, the skill of dribbling based on force/torque information is also important.

B. State of the Art

The dribbling task is closely related to juggling tasks which have been studied by various researchers. Holmes investigated the motion of a ball bouncing vertically on a sinusoidally vibrating table [6]. Buehler *et al.* studied the stabilization of planar juggling tasks [7]. Rizzi *et al.* followed up this work and accomplished spatial juggling of two balls with a 3 DoF robot [8]. Schaal *et al.* discussed open loop stable control strategies [9] and the aspect of learning control [10] for robot juggling. Ronsse *et al.* studied juggling based on force respectively impact time feedback and for different types of actuation [11], [12].

The comparison of the dribbling task with the *classic* juggling task shows an important conceptual difference between the two tasks: in addition to the actuation impact, the dribbling task is characterized by a second, autonomous impact with the ground. This additional loss of energy has two effects which complicate the dribbling task: Firstly, dribbling exhibits a higher sensitivity towards parameter variations in comparison with the *classic* juggling task. Secondly, a return to the dribbling height is not guaranteed and dependent on the estimate of the coefficient of restitution c_r and the precise motion execution. Due to these differences, the dribbling task is an interesting example of a rhythmic task to be studied. An example of robot dribbling was presented by Shiokata *et al.* [13]. For the experiment, a ping-pong ball and a multi-fingered hand were used. The dribbling task was performed solely with the center finger of the device, giving the system four degrees of freedom. In the presented experiments, the dribbling could only be maintained for about 1 s.

C. Contribution

This paper presents two control designs for ball dribbling based on force/torque feedback and visual feedback, respectively. The dribbling task is investigated in three dimensional space. Also, specific challenges of the dribbling task are discussed; namely accurate tracking of the ball, trajectory prediction based on a detailed physical model and exact timing and execution of the robot motions. In Sec. II, the hardware setup, composed of robotic and vision system, is presented. A detailed description of the control design is given in Sec. III. Experimental results are shown in Sec. IV. Finally, a conclusion and outlook can be found in Sec. V.

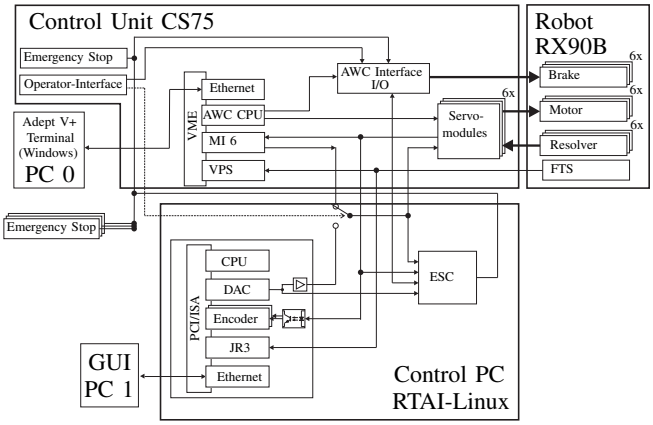

Fig. 1. Open control architecture with a Stäubli RX90B and a control PC

II. SETUP

This section briefly introduces the two main components of the experimental setup: the robotic system with an open control architecture and the vision system.

A. Robotic System

For the experiments, a Stäubli RX90B industrial robot with six revolute joints is used. The robot is equipped with a 6 DoF force/torque sensor at the end effector. For the dribbling task, a circular plate with radius 0.17 m is mounted on the end effector. In order to implement and evaluate own control concepts, a PC-based controller has been developed which works in parallel with the robot control unit. The configuration, calibration and supervising of the robot is performed with the original architecture. The control, however, is passed over to the additional PC, which runs Matlab/Simulink in a real-time Linux (RTAI) environment. A schematic of the setup is depicted in Fig. 1. For further details on the open control architecture see [14].

B. Vision System

The stereo vision system for ball tracking consists of two Mikrotron MC1311 high-speed cameras, two frame grabbers and a general purpose PC with two PCIe ports. The tracking algorithm obtains images from the frame grabber for image processing. The ball position is sent to the control PC via a TCP network connection. The two cameras are mounted with a baseline b = 2 m and converging axes. The distance between baseline and fixation point is 3 m, the distance between baseline and robot base is 3.5 m.

III. CONTROL DESIGN

The overall control structure of the system is depicted in Fig. 2. It has three main modules: ball tracking, ball trajectory prediction and robot trajectory generation. In the first module, ball tracking is performed based on either force/torque or visual sensor information. In the second module, the ball trajectory is predicted based on the provided sensor information. In the third module, the end effector trajectory is generated based on the predicted ball trajectory. The three modules will be described in the following subsections. As a reference, Fig. 3 shows the notations which

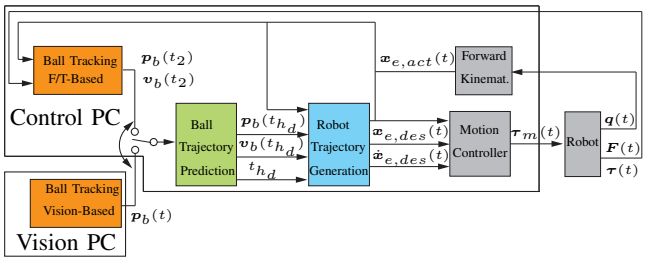

Fig. 2. Overall control structure with a control PC and a vision PC

are used throughout the paper. As depicted, the subscripts 1 to 4 denote the points in time directly before/after an impact with the end effector and the ground, respectively.

A. Ball Tracking Module

The design of the ball tracking module depends on whether tracking is performed force/torque- or vision-based. In the following, the two ball tracking modules are presented.

1) Force/Torque-based:

In this case, the ball position $\boldsymbol{p}_b(t)$ is tracked based on the measurements of the force/torque sensor. As the task is performed without gripping, useful measurements are only available during the impact between the ball and the plate at the end effector. These impacts have a duration of 15 to 25 ms and $t_{i,e}$ is defined by the point in time where the magnitude of the contact force in vertical direction $F_{z,con}(t)$ reaches its local maximum. After compensating the actual end effector acceleration $\boldsymbol{a}_{e,act}(t_{i,e})$ and considering the actual end effector orientation $\boldsymbol{o}_{e,act}(t_{i,e})$, the ball position

$$\boldsymbol{p}_b(t_{i,e}) = \boldsymbol{p}_e(t_{i,e}) + \begin{bmatrix} -M_{y,con}/F_{z,con} \\ +M_{x,con}/F_{z,con} \\ -r_b \end{bmatrix} (t_{i,e}) \qquad (1)$$

is calculated, where r_b denotes the radius of the ball, $\boldsymbol{p}_e(t_{i,e})$ the actual end effector position at the point of impact and the subscript *con* the contact forces and torques. For the calculation of the ball velocity $\boldsymbol{v}_b(t)$, the physical model of the collision is used. The coefficient of restitution c_r provides a quantitative measure for the loss of energy during the collision. It depends on the properties of the collision and the colliding objects, e.g. shape, stiffness and the velocity at which the impact occurs [15]. For a constant vertical end effector velocity $v_{z,e}$, the coefficient of restitution for the

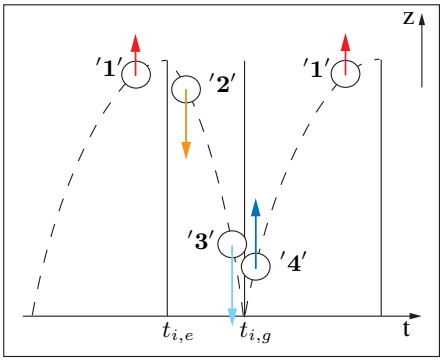

Fig. 3. Notation of the different points in time during a dribbling cycle

impact with the end effector is defined as

$$c_{r,e} = \frac{v_{z,e} - v_{z,b}(t_2)}{v_{z,b}(t_1) - v_{z,e}} \tag{2}$$

where $v_{z,b}(t_1)$ and $v_{z,b}(t_2)$ denote the vertical ball velocity before and after the collision. For the setup, $c_{r,e}$ was determined experimentally with the following assumptions: First, the ball properties are invariant as the ball is kept at constant pressure. Second, for the velocity in the experiments which ranges from 0 to ± 5 m/s, there is no noticeable dependence of $c_{r,e}$ on the velocity. In contrast, the decreasing stiffness of the end effector plate in radial direction from center to rim was considered. This results in

$$c_{r,e}(p, v, \boldsymbol{p}_b(t_{i,e})) \approx c_{r,e}(\boldsymbol{p}_b(t_{i,e})) = c_{r,e}. \tag{3}$$

At $t_{i,e}$, the relative ball velocity with respect to the end effector coordinate system is zero $v^e_{z,b}(t_{i,e}) = 0$ and Newton's second law can be written as

$$0 - v^e_{z,b}(t_1) = \frac{1}{m_b} \int_{t_1}^{t_{i,e}} F_{z,con}(t) \mathrm{d}t. \tag{4}$$

With (1)–(4), the relative velocity $v^e_{z,b}(t_2)$ after the impact is calculated. The ball velocity in horizontal direction is neglected, which means $p_{x,b}(t)$ and $p_{y,b}(t)$ are assumed to be constant throughout the next dribbling cycle. With $v^e_{z,b}(t_2)$ and the assumption of zero velocity in the horizontal plane, the absolute ball velocity immediately after the impact

$$\boldsymbol{v}_b(t_2) = \left[0, 0, v^e_{z,b}(t_2) + v_{z,e}\right]^T, \tag{5}$$

is determined and forms together with $\boldsymbol{p}_b(t_2)$ the output of the force/torque-based ball tracking module.

2) Vision-based:

The main part of the ball tracking algorithm with the vision system is color-based tracking. First, the ball is extracted according to its color information in the HSV space. Then, the speckle, not related edges and noise in the image are filtered out through an opening (1 x erosion, 1 x dilation) operation. Finally, by computing the first order moment of the image, the center of the ball is obtained. A block diagram of the image processing algorithm is depicted in Fig. 4. In order to increase the tracking frequency and to reduce the tracking delay, a window searching method is used: Once the ball is detected in the full size image (1280 x 1024 pixel), the search area is reduced to a 180 x 180 pixel window. The position of the window is determined by a linear position prediction and tracking is performed with 150 Hz.

B. Ball Trajectory Prediction Module

The algorithm for the prediction of the ball trajectory depends on whether the ball tracking is performed force/torque- or vision-based. In the following, the two algorithms for trajectory prediction are described in detail. Regardless of the used tracking modality, the output of the module is the predicted point in time t_{h_d} when the desired dribbling height h_d will be reached as well as the ball position $\boldsymbol{p}_b(t_{h_d})$ and

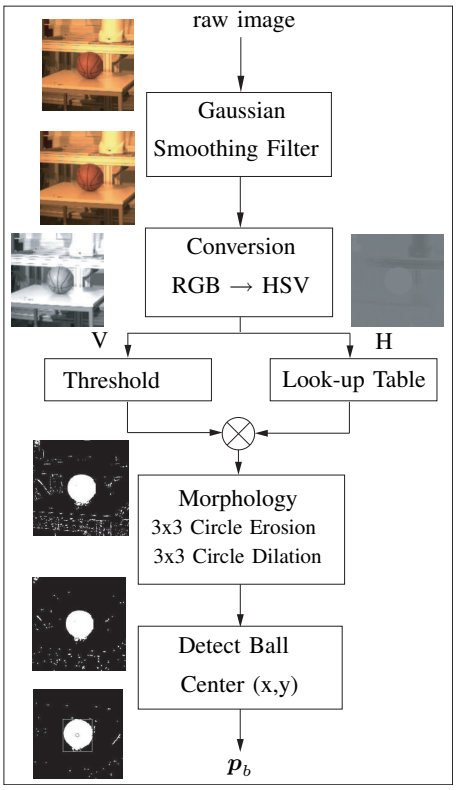

Fig. 4. Block diagram of the image processing

the predicted ball velocity $\boldsymbol{v}_b(t_{h_d})$ at the desired dribbling height. For the free flight phases, the model

$$\boldsymbol{p}_b(t) = \frac{1}{2} g \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} (t - t_k)^2 + \boldsymbol{v}_b(t_k)(t - t_k) + \boldsymbol{p}_b(t_k) \tag{6}$$

is used, where g denotes the gravitational acceleration and the index k labels drop ($k = 2$) or rise ($k = 4$) phase. For the ground contact, it is assumed that the angle of impact with respect to the vertical z-axis is smaller than $30°$. With this assumption, no sliding occurs at the contact point, see [15], [16]. Further it is assumed that the rotational velocity components of the ball before the impact are negligible. This approximates the horizontal ball velocity

$$\boldsymbol{v}_{h,b}(t_4) \approx \frac{m_b r_b^2}{I_b + m_b r_b^2} \boldsymbol{v}_{h,b}(t_3) \tag{7}$$

where I_b denotes the moment of inertia of the ball. Inserting the expression for a spherical shell, $I_b = \frac{2}{3} m_b r_b^2$, in (7) leads to a proportionality constant $c_f = 0.6$ for the horizontal velocities before and after the impact. This defines the ball velocity after the ground impact as

$$\boldsymbol{v}_b(t_4) = \left[c_f, c_f, -c_{r,g}\right]^T \boldsymbol{v}_b(t_3) \tag{8}$$

where $c_{r,g}$ is the coefficient of restitution for the ground impact. The models for the free flight and the bounce phase are used to calculate the future ball trajectory based on the input values of the module.

1) Force/Torque-based:
In this case, the input to the module is the ball position $\boldsymbol{p}_b(t_2)$ and the ball velocity $\boldsymbol{v}_b(t_2)$ at a known time t_2. Based on these specs and the physical model described above, the resulting position and velocity of the ball is determined and the values for t_{h_d}, $\boldsymbol{p}_b(t_{h_d})$ and $\boldsymbol{v}_b(t_{h_d})$ are calculated. When the next impact occurs, updated values for $\boldsymbol{p}_b(t_2)$, $\boldsymbol{v}_b(t_2)$ and t_2 are provided by the tracking module and the ball trajectory prediction is updated accordingly.

2) Vision-based:
In this case, the input for the ball trajectory prediction is the tracked ball position $\boldsymbol{p}_b(t)$. Based on the measurements and the underlying physical model, the parameters of the ball trajectory are determined with the following approach: If the ball position exceeds/falls short of a specified threshold value, the module determines the parameters of the equation of motion using (6) with a recursive least squares algorithm based on the measurements of the last 70 ms. As (6) is only valid in free flight, the threshold value $\boldsymbol{p}_{b,tr}$ for the ball position has to be chosen in such a way that the last 70 ms of the ball trajectory have been in free flight. With known equation of motion, the position $\boldsymbol{p}_b(t_{h_d})$ and velocity $\boldsymbol{v}_b(t_{h_d})$ of the ball and the point in time t_{h_d} when it reaches the desired dribbling height h_d are calculated.

The accuracy of the prediction depends on the number of measurements, the precision of the underlying model and the specified threshold value. The dependence on the specified threshold is due to the fact that the rotational movement of the ball is neglected. On this account, predictions after the ground impact are more precise, see Sec. IV.

C. Robot Trajectory Generation Module

The inputs for the robot trajectory generation are the predicted ball position in the xy-plane $x_b(t_{h_d})$ and $y_b(t_{h_d})$, the predicted ball velocity $\boldsymbol{v}_b(t_{h_d})$ at the dribbling height h_d and the point in time t_{h_d} when h_d is reached. For initialization, the algorithm for the robot trajectory generation depends on whether the ball tracking is performed force/torque- or vision-based. The differences for the two modalities will be described at the end of this section. Once the initialization is done, the trajectory generation is independent of the used sensor modality. During the impacts with the ball, the module allows only vertical end effector motions. Horizontal deviations from the desired position are compensated by adjusting the orientation of the end effector. The orientation is described by angle/axis representation, with axis $\boldsymbol{r}_a$ in the horizontal xy-plane and angle $\phi = 0°$ corresponding to an end effector plate facing the ground.

With known $c_{r,g}$ and dribbling height h_d, the vertical ball velocity after the impact required for a return to the dribbling height h_d is

$$v_{z,b}(t_2) = -\sqrt{2g\left(h_d - r_b\right)\left(c_{r,g}^2 - 1\right)/c_{r,g}^2}. \tag{9}$$

To determine the required horizontal ball velocity after the impact $v_{h,b}(t_2)$, the deviations e_x and e_y of the ball's x- and y-coordinate and the overall error e_h in the horizontal plane

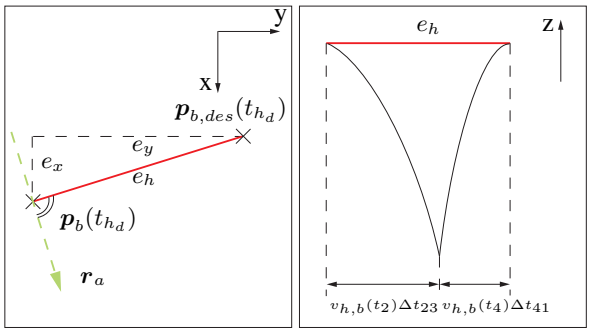

Fig. 5. Position error e_h and axis of rotation $\boldsymbol{r}_a$

at the specified dribbling height

$$\begin{aligned} e_x &= p_{x,b,des}(t_{h_d}) - p_{x,b}(t_{h_d}) \\ e_y &= p_{y,b,des}(t_{h_d}) - p_{y,b}(t_{h_d}) \\ e_h &= \sqrt{e_x^2 + e_y^2} \end{aligned} \tag{10}$$

are calculated, see Fig. 5. Based on the deviations e_x and e_y, the axis of rotation

$$\boldsymbol{r}_a = \frac{1}{e_h}\left[-e_y, -e_x, 0\right]^T \tag{11}$$

is defined as illustrated in Fig. 5. The horizontal motion of the ball during one cycle can be written as sum of the contributions of drop and rise phase

$$e_h = v_{h,b}(t_2)(\Delta t_{23} + c_f \Delta t_{41}), \tag{12}$$

which defines the required horizontal ball velocity $v_{h,b}(t_2)$. With the assumption of no sliding between the contact points during impact and the neglection of rotational velocity, the vertical and horizontal ball velocity after the impact are determined as a function of ϕ, $v_{z,e}$ and the known inputs. With small angle approximation, this leads to

$$\begin{aligned} v_{z,b}(t_2) \approx& -c_{r,e} v_{z,b}(t_1) + (1 + c_{r,e})\, v_{z,e} \\ &+ \phi\,(c_f - c_{r,e})\, v_{h,b}(t_1) \\ v_{h,b}(t_2) \approx& \phi(c_f + c_{r,e})\,(v_{z,b}(t_1) - v_{z,e}) + c_f v_{h,b}(t_1). \end{aligned} \tag{13}$$

With (2) and (9)-(14), the vertical end effector velocity $v_{z,e}$ and the rotation angle ϕ are determined. With the inputs and the calculated values, the end effector trajectory is created using 6th order polynomial interpolation. At the time at which the ball is predicted to reach the dribbling height h_d, the desired position, orientation and velocity of the end effector are

$$\begin{aligned} \boldsymbol{p}_{e,des}(t_{h_d}) &= \left[x_b(t_{h_d}), y_b(t_{h_d}), h_d + r_b\right]^T \\ \boldsymbol{o}_{e,des}(t_{h_d}) &= \left[\cos(\phi/2), \boldsymbol{r}_a \sin(\phi/2)\right]^T * \boldsymbol{o}_{e,i} \\ \boldsymbol{v}_{e,des}(t_{h_d}) &= \left[0, 0, v_{z,e}\right]^T. \end{aligned} \tag{14}$$

After the impact with the ball, the end effector decelerates and returns to its initial/rest height. During the dribbling cycle, the horizontal end effector position is adjusted to match the tracked horizontal ball position. For the motion control of the robot, a computed torque feedforward control in combination with a decentralized PD-controller is used.

1) Force/Torque-based:
In this case, the ball can only be tracked while in contact with the plate. Due to the short time period of the impacts and the inertia of the robot, suitable trajectories cannot be initiated and executed during the impacts. For that reason, an initial ball position and velocity has to be known. This is realized by starting the dribbling task at a pre-specified time with the ball initially at rest on the plate. The initialization movement consists of a simultaneous translational and rotational movement.

2) Vision-based:
In this case, the ball is tracked by the vision system. The dribbling task is initiated when the predicted ball position $\boldsymbol{p}_b(t_{h_d})$ and velocity $\boldsymbol{v}_b(t_{h_d})$ are within a pre-specified range. This means that the dribbling task is triggered by letting the ball drop in the workspace of the robot.

IV. EXPERIMENTAL RESULTS

In this section, the experimental results for the two ball tracking methods are presented. For both methods, a desired dribbling height of $h_d = 0.9$ m is specified.

A. Force/Torque-based tracking

With the force/torque-based approach, dribbling for 5 to 10 cycles is achieved. The performance is strongly dependent on a precise estimate of $c_{r,e}/c_{r,g}$. This estimate can be provided either through separate experiments or through execution of the vision-based approach. Due to the fact that the end effector itself is moving and vibrating during the impact, the measured relative forces/torques are noisy. This complicates the estimation of $\boldsymbol{v}_b(t_2)$, especially for impacts with smaller force magnitudes. Another decline in performance arises from the fact that the horizontal ball velocity is assumed to be zero. This means, the desired end effector orientation is calculated based on the xy-position of the last impact. Consequently, an increase of the position error during the current dribbling cycle is not compensated.

B. Vision-based tracking

With the vision-based approach, dribbling for multiple cycles (> 15) is achieved. The ball is held in a human's hand and then dropped into a specified area of the robot workspace to initialize the dribbling. Fig. 6 shows a sequence of snapshots taken during a dribbling task. The time duration between two consecutive snapshots is 0.16 s, the extracted sequence corresponds to the time span from 0.5 s to 2.9 s in the following figures. In Fig. 7, the ball position tracked with the vision system during a dribbling task is depicted. To create the desired end effector trajectory, the ball trajectory is predicted. As depicted in Fig. 8 and 9, the two different methods described in section III-B are evaluated: a prediction before and a prediction after the bounce on the ground. In Fig. 8, the value of the variable changes from 0 to h_d when the next prediction of t_{h_d} becomes available. Accordingly, a step down from h_d to 0 occurs once the predicted time $t_{h,d}$ is reached. This means, the wider the step, the earlier the prediction is available and for an ideal prediction, the step down will occur exactly when the ball reaches h_d. As depicted in Fig. 8, the accuracy of both approaches is almost identical while the first approach provides the predicted value approximately 0.15 s earlier. Fig. 9 shows that for the prediction of the x-coordinate of the ball at the dribbling height, the second approach is more accurate. This is also true for the y-coordinate, which is omitted for the sake of compact representation. Again, the drawback of the second method is that the time span until the ball reaches the dribbling height h_d is approximately 0.15 s smaller. Hence, the second method requires faster motion execution of the robot. In the presented experiment, the robot trajectory is generated based on the first prediction method. The paper includes a video attachment which illustrates the vision-based dribbling. For large position errors $e_h \geq 0.25$ m, the performance is currently limited: To avoid exceedance of the maximum joint velocities of the robot, the admissible robot workspace is restricted.

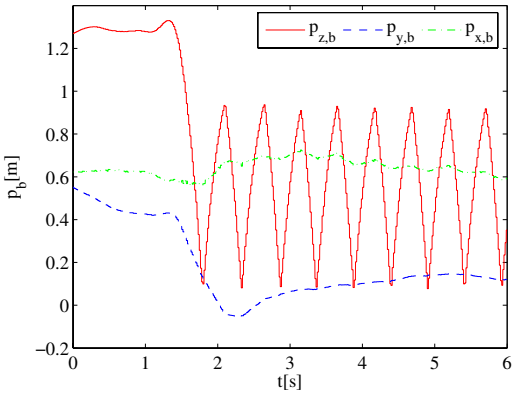

Fig. 7. Tracked ball position $\boldsymbol{p}_b(t)$ (vision-based)

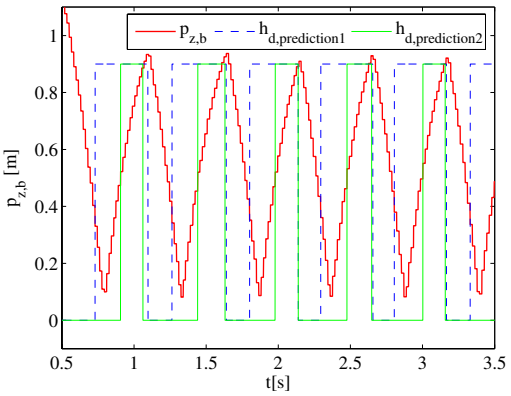

Fig. 8. Predicted point in time $t_{h,d}$ (vision-based) for reaching h_d

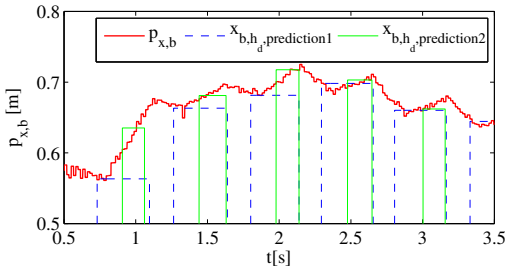

Fig. 9. Predicted x-coordinate $p_{x,b}(t_{h_d})$ (vision-based) of the ball at h_d

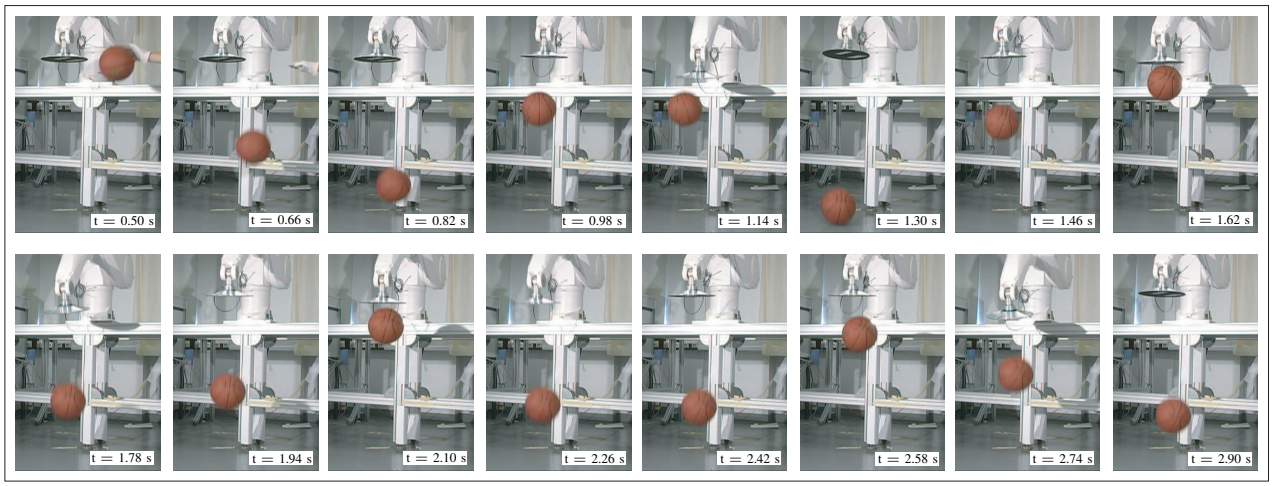

Fig. 6. Experimental snapshots of vision-based dribbling with initialization by human operator (sequence from 0.5 s to 2.9 s)

V. DISCUSSION

A. Conclusions/Results

Two control designs for ball dribbling with a 6 DoF industrial robot based on different sensor modalities were presented. For both approaches, dribbling for multiple cycles was achieved. The investigated methods for model-based trajectory prediction showed a high accuracy for the vertical direction. The performance in horizontal direction differed, due to the fact that the rotational ball velocity was neglected in this work. For the vision-based approach, the coefficient of restitution $c_{r,e}/c_{r,g}$ is adapted based on the tracked ball position $\boldsymbol{p}_b(t)$. For the force/torque-based approach, experimentally determined, fixed values for $c_{r,e}$ and $c_{r,g}$ were used. Hence, this approach is more sensitive to parameter variations. However, the force/torque-based controller can be used to maintain the dribbling task if the vision-based approach fails, e.g. when the ball is occluded. Also, one can switch to the force/torque-based controller once the values of $c_{r,e}/c_{r,g}$ have been found with the vision-based controller.

B. Future Work

When investigating the dribbling task based on different sensor modalities, the question arises to what extent a control design based on vision and force/torque will improve the performance of the robot. To this end, experiments to fuse the vision and force/torque sensor information are in preparation. For the goal of a robotic basketball player, ball dribbling has to be realized for changing environmental conditions. To this end, future work will extend the system model with a more detailed impact model including rotational ball velocity. In addition, experiments will be conducted evaluating dexterous dribbling tasks with varying height and varying horizontal position using the industrial robot as well as a dual-arm manipulator robot.

VI. ACKNOWLEDGMENTS

The first author gratefully thanks the German National Academic Foundation for their support.

REFERENCES

[1] H. Kitano, M. Asada, I. Noda, and H. Matsubara, "Robocup: robot world cup," *IEEE Robotics & Automation Magazine*, vol. 5, pp. 30–36, Sept. 1998.

[2] K. Kosuge, "Human-robot interaction - what we learned from robot helpers and dance partner robots," in *Proc. 17th IEEE International Symposium on Robot and Human Interactive Communication RO-MAN 2008*, 1–3 Aug. 2008.

[3] L. Acosta, J. Rodrigo, J. Mendez, G. Marichal, and M. Sigut, "Ping-pong player prototype," *IEEE Robotics & Automation Magazine*, vol. 10, pp. 44–52, Dec. 2003.

[4] K. Hasegawa, S. Shimizu, and M. Yoshizawa, "Robotics applied to sports engineering," *Advanced Robotics*, vol. 14, no. 5, pp. 377–379, 2000.

[5] K.-K. Lee, G. Bätz, and D. Wollherr, "Basketball robot: Ball-on-plate with pure haptic information," in *Proc. IEEE International Conference on Robotics and Automation ICRA 2008*, pp. 2410–2415, 2008.

[6] P. J. Holmes, "The dynamics of repeated impacts with a sinusoidally vibrating table," *Journal of Sound and Vibration*, vol. 84, no. 2, pp. 173 – 189, 1982.

[7] M. Buehler, D. Koditschek, and P. Kindlmann, "A one degree of freedom juggler in a two degree of freedom environment," in *Proc. IEEE International Workshop on Intelligent Robots*, pp. 91–97, Oct. 31 –Nov. 2, 1988.

[8] A. Rizzi and D. Koditschek, "Progress in spatial robot juggling," in *Proc. IEEE International Conference on Robotics and Automation*, pp. 775–780, 12–14 May 1992.

[9] S. Schaal and C. Atkeson, "Open loop stable control strategies for robot juggling," in *Proc. IEEE International Conference on Robotics and Automation*, pp. 913–918, 2–6 May 1993.

[10] S. Schaal and C. Atkeson, "Robot juggling: implementation of memory-based learning," *IEEE Control Systems Magazine*, vol. 14, pp. 57–71, Feb. 1994.

[11] R. Ronsse, P. Lefevre, and R. Sepulchre, "Rhythmic feedback control of a blind planar juggler," *IEEE Transactions on Robotics*, vol. 23, no. 4, pp. 790–802, 2007.

[12] R. Ronsse, P. Lefevre, and R. Sepulchre, "Sensorless stabilization of bounce juggling," *IEEE Transactions on Robotics*, vol. 22, no. 1, pp. 147–159, 2006.

[13] D. Shiokata, A. Namiki, and M. Ishikawa, "Robot dribbling using a high-speed multifingered hand and a high-speed vision system," in *Proc. IEEE International Conference on Intelligent Robots and Systems IROS 2005*, 2005.

[14] J. Hoogen, R. Riener, and G. Schmidt, "Control concepts for an industrial robot used as kinesthetic knee joint simulator," in *Proc. Third International Workshop on Robot Motion and Control RoMoCo '02*, pp. 21–26, 9–11 Nov. 2002.

[15] A. Domenech, "A classical experiment revisited: The bounce of balls and superballs in three dimensions," *American Journal of Physics*, vol. 1, pp. 28 – 36, 2005.

[16] R. Cross, "Grip-slip behavior of a bouncing ball," *American Journal of Physics*, vol. 11, pp. 1093 – 1102, 2002.