

Dynamic Manipulation: Nonprehensile Ball Catching

Georg Bätz¹, Arhan Yaqub¹, Haiyan Wu¹, Kolja Kühnlenz^{1,2}, Dirk Wollherr¹ and Martin Buss¹

¹Institute of Automatic Control Engineering, ²Institute for Advanced Study

Technische Universität München

D-80290 München, Germany

{georg.baetz, arhan.yaqub, haiyan.wu, koku, dw, mb}@tum.de

Abstract—Most industrial robots nowadays still employ strategies that neglect or minimize the effects of task dynamics. Some tasks, however, are intrinsically dynamic and can only be accomplished by considering their dynamic aspects. We address ball catching as a prominent and widely studied example for such a task. The paper follows a special approach to accomplish the task: the nonprehensile catching, which means catching without a form- or force-closure grasp. Depending on the tracked ball velocity, two different catching methods are proposed: First, catching of the ball during the initial contact. Second, catching the ball after an initial rebound during the subsequent contact. For both approaches, the ball trajectory is predicted with a recursive least squares algorithm. The dynamic manipulability measure is used for the contact point selection. Once a permanent contact between ball and end effector is established, a balancing control based on force/torque feedback is applied. Both methods are experimentally validated using a six DoF industrial robot.

Keywords - Dexterous Manipulation, Reactive and Sensor-Based Planning, Real-time Control, Robotics

I. INTRODUCTION

In accordance with [1], we refer the term *dynamic manipulation* to methods which *actively use* the task dynamics instead of merely tolerating them. Dynamic manipulation offers two potential benefits: First, the execution time of tasks can be reduced. Second, the dexterity of the robotic system can be increased since numerous manipulation tasks are intrinsically dynamic and can only be accomplished by considering the dynamic aspects. For conventional manipulation with form- and/or form-closure grasp, task/object specific grippers are necessary. Here, the nonprehensile approach offers two advantages: First, the development of a gripper is dispensable [2]. Second, a generic end effector design (e.g. a plate) allows the handling of bulky objects. The increase in dexterity and the cost efficiency motivated the investigation of nonprehensile object catching.

A. State of the Art

The task of robotic ball catching with a form- or force-closure grasp has been studied by various researchers: Andersson realized two-dimensional catching of ping-pong balls up to a speed of 1.3 m/s with a PUMA robot arm [3]. Hove and Slotine presented three-dimensional catching with a four DoF manipulator [4]. Bühler, Koditschek and coworkers successfully applied a mirror law for catching a puck constrained to lie on an inclined plane [5]. As control input they used

a bar which was actuated around a revolute joint. Burrige *et al.* followed up this work and described a mirror law for catching a ball with a three DoF robot and a planar paddle as end effector [6]. Frese *et al.* presented ball catching with a seven DoF robot [7]. In their setup, a basket served as end effector to catch the ball. Riley and Atkeson investigated ball catching with a humanoid robot using a baseball glove to catch [8]. Namiki *et al.* developed a high-speed vision system with a sampling rate of 1 kHz and a resolution of 128 x 128 pixels. They combined the system with a multi-fingered robotic hand with eight DoF and demonstrated catching of a rubber ball [9]. The hand was mounted stationary on a table and vertical ball trajectories were investigated. Smith *et al.* studied teleoperated ball catching in [10].

Lynch and Mason discussed controllability, motion planning and implementation of planar dynamic nonprehensile manipulation [2]. In addition, the concept has been previously considered in various applications: Arai and Khatib investigated the task of parts reorienting using dynamic forces [11]. Aboaf *et al.* applied learning approaches to improve the robot's ability to juggle a ball [12]. Robotic juggling and batting tasks were also investigated by Schaal *et al.* [13], Bühler *et al.* [14] and Andersson [15].

B. Contribution

Previous works on robotic ball catching focused mainly on catching with either a force- or a form-closure grasp. An exception is the work of Burrige *et al.* which discussed the nonprehensile catching of a ball with a three DoF robot [6]. In our previous works, other dynamic manipulation skills have been investigated [16]. This paper presents a control design for two nonprehensile catching strategies based on combined visual and force/torque feedback. The approach considers relevant strategies for realizing flexible object handling. To this end, the catching task is investigated in three dimensional space. Also, specific challenges of the catching task are discussed: accurate tracking of the ball, model-based trajectory prediction and exact planning and execution of the robot motions. The remainder of the paper is organized as follows: 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.

II. SETUP

This section 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 six DoF force/torque sensor at the end effector. For the catching task, a circular plate with radius 0.17 m serves as 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. For further details on the open control architecture see [16].

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. 1. It has three main modules: ball tracking, ball trajectory prediction and robot trajectory generation. In the first module, ball tracking is performed based on visual sensor information. In the second module, the ball trajectory is predicted based on the provided sensor feedback. In the third module, the end effector trajectory is generated based on the predicted ball trajectory and the measured contact forces. The three modules will be described in the following subsections. For the motion control of the robot, a computed torque feed forward control in combination with a decentralized PD-controller is used. For the control design, the following assumptions are made:

- A1 Air resistance and rotational ball velocity are negligible.
- A2 Impacts between ball and end effector are instantaneous inelastic collisions described by the coefficient of restitution c_r and with the angle of incidence equal to the angle of reflexion.

A. Ball Tracking Module

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)

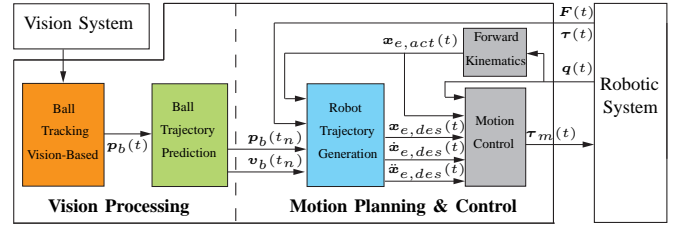


Fig. 1. Overall control structure

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. 2. 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.

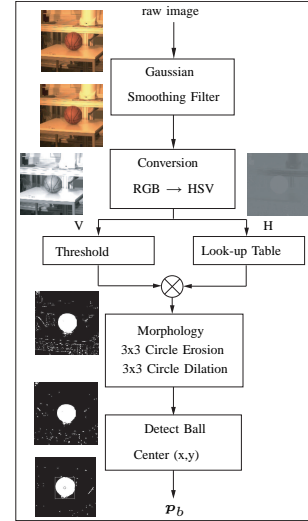


Fig. 2. Block diagram of the image processing

B. Ball Trajectory Prediction Module

The input for the module is the ball position $p_b(t)$ tracked by the camera system. The triggering of a throw can be realized in various ways: One approach is to check whether the vertical acceleration of the ball matches the gravitational acceleration for a certain period of time. If that is the case, one can assume that the ball is in free flight. The drawback of this method is that the acceleration $a_b(t)$ of the ball has to be determined. If only position information is available, imprecise/noisy acceleration estimates impede this approach. Hence, a different approach was used for the experiments: a throw is triggered when the vertical ball position exceeds a height constraint. The point in time when this occurs is denoted t_a and the ball is assumed to be in free flight afterwards. In order to find the suitable catching strategy and to generate the corresponding robot trajectory, the prediction of the ball trajectory is necessary. The prediction is realized with a recursive least squares fitting method for a sample

period T_s . The trajectory for free flight is given by

$$\tilde{\mathbf{p}}_{[k]} = \begin{bmatrix} p_{x,[k]} \\ p_{y,[k]} \\ p_{z,[k]} - 0.5gt_{[k]}^2 \end{bmatrix} = \begin{bmatrix} p_{x,0} & v_{x,0} \\ p_{y,0} & v_{y,0} \\ p_{z,0} & v_{z,0} \end{bmatrix} \begin{bmatrix} 1 \\ t_{[k]} \end{bmatrix} \quad (1)$$

where g denotes the gravitational acceleration. This results in the recursive least squares estimates for the parameters

$$\hat{\mathbf{m}}_{i[k+1]} = \hat{\mathbf{m}}_{i[k]} + \mathbf{h}_{[k]} \left(\bar{p}_{i[k+1]} - \mathbf{R}_{[k+1]}^T \hat{\mathbf{m}}_{i[k]} \right) \quad (2)$$

where $\bar{p}_{i[k+1]}$ is the measured i -coordinate of the ball and

$$\begin{aligned} \hat{\mathbf{m}}_{i[k]} &= [p_{i,0[k]} \quad v_{i,0[k]}]^T \quad i \in \{x, y, z\}, \\ \mathbf{R}_{[k+1]} &= [1 \quad t_{[k+1]}]^T, \quad \mathbf{h}_{[k]} = \frac{\mathbf{\Pi}_{[k]} \mathbf{R}_{[k+1]}}{1 + \mathbf{R}_{[k+1]}^T \mathbf{\Pi}_{[k]} \mathbf{R}_{[k+1]}}, \\ \mathbf{\Pi}_{[k+1]} &= \mathbf{\Pi}_{[k]} - \mathbf{h}_{[k]} \mathbf{R}_{[k+1]}^T \mathbf{\Pi}_{[k]}. \end{aligned} \quad (3)$$

With the estimates for the initial state of the ball, the future ball trajectory is predicted which serves as input for the robot trajectory generation module.

C. Robot Trajectory Generation Module

The trajectory generation is initiated at time $t_b = t_a + T_s$, when the final prediction of the ball trajectory becomes available. The inputs of the module are the predicted ball trajectory $f(\mathbf{p}_b(t_b), \mathbf{v}_b(t_b), t)$, the actual robot pose $\mathbf{x}_{act}(t)$ and the measured forces $\mathbf{F}(t)$ and torques $\boldsymbol{\tau}(t)$ at the end effector. The outputs are the desired end effector pose $\mathbf{x}_{e,des}(t) = [\mathbf{p}_{e,des}(t)^T, \mathbf{o}_{e,des}(t)^T]^T$, velocity $\dot{\mathbf{x}}_{e,des}(t)$ and acceleration $\ddot{\mathbf{x}}_{e,des}(t)$. For the end effector orientation, the angle/axis notation is used. The module performs the following subtasks: First, it applies a selection criteria to evaluate possible *catching points* (which refers to the point of initial contact). Second, it determines a catching strategy and creates a feasible catching trajectory for the end effector. Third, the module decelerates and stabilizes the ball on the plate once the contact between ball and end effector has been established.

1) Catching Point Selection Criteria:

For an intrinsically dynamic task, the robot motion planning has to be performed on-line based on the provided sensor information. Suitable trajectories have to be generated in limited time and in general there is no unique solution for the task. In this work, we choose catching point candidates based on two criteria which will be detailed in the following: the distance from mechanical joint limits and the dynamic manipulability measure. The creation of look-up tables then allows on-line evaluation of these selection criteria.

a) Distance from joint limits:

An intuitive selection criteria is the distance from mechanical joint limits [17],

$$w_j(\mathbf{q}) = \prod_{i=1}^n w_{j,i}(q_i, q_{i,min}, q_{i,max}) \quad (4)$$

where $w_{j,i}$ corresponds to the joint limit measure for the i^{th} joint. The value of $w_{j,i}$ equals 1 if joint i is within 75% of the admissible range and gradually decreases from 1 to 0 if the joint is within 75% to 100% of its admissible range.

b) Dynamic manipulability measure:

A global measure for the manipulation ability has been proposed by Yoshikawa in [18]. The measure quantifies the ability for arbitrarily changing position and orientation of the end effector in a given posture. The main drawback of this concept is the fact that it is a kinematic measure ignoring the arm dynamics. Hence it is not suitable for precise high-speed motion control. This shortcoming is addressed by the dynamic manipulability measure w_d introduced in [19]. The scalar value is defined as

$$w_d(\mathbf{q}) = \frac{1}{w_{d,max}} \sqrt{\det \left(\mathbf{J}(\mathbf{q}) \left(\mathbf{B}^T(\mathbf{q}) \mathbf{B}(\mathbf{q}) \right)^{-1} \mathbf{J}^T(\mathbf{q}) \right)} \quad (5)$$

where $\mathbf{q} \in \mathbb{R}^n$ is the vector of joint angles, $\mathbf{J}(\mathbf{q}) \in \mathbb{R}^{m \times n}$ is the manipulator Jacobian and $\mathbf{B}(\mathbf{q}) \in \mathbb{R}^{n \times n}$ is the inertia matrix of the manipulator. The scaling factor $w_{d,max}$ is used to normalize the measure. Fig. 3 shows the dynamic manipulability measure of the utilized robot for different yz -planes. The reference coordinate system is located in the robot base with the z -direction pointing vertically upwards.

2) Catching Trajectory:

In order to find possible catching points, the module determines where the predicted ball trajectory intersects the robot workspace. For the intersecting points, the values of w_j and w_d are determined by using look-up tables which have been created off-line. The ten points with the highest combined value $w_j + w_d$ are selected as a catching point candidates and the resulting trajectories are checked for dynamic feasibility. Based on this strategy, three different cases can occur: *no catch*, *direct catch* and *indirect catch*. The first case occurs if all catching points violate dynamical and/or workspace constraints. The latter two cases are detailed in the following two paragraphs.

a) Direct Catch:

For a direct catch, the end effector trajectory at time t_c of the initial contact has to fulfill the following constraints:

$$\begin{aligned} \mathbf{p}_e(t_c) &= \mathbf{p}_b(t_c) + \frac{r_b}{|\mathbf{v}_b(t_c)|} \mathbf{v}_b(t_c), \quad \mathbf{v}_e(t_c) = \mathbf{v}_b(t_c), \\ \mathbf{a}_e(t_c) &= \mathbf{a}_b(t_c), \quad \mathbf{n}_e(t_c) \parallel -\mathbf{v}_b(t_c). \end{aligned} \quad (6)$$

Here, r_b denotes the radius of the ball and $\mathbf{n}_e(t_c)$ the normal vector on the end effector plate. Once contact between ball and plate is established, the end effector is decelerated in the direction of $\mathbf{v}_e(t_c)$ with $|\mathbf{a}_e| = 30 \text{ m/s}^2$. Hence, the time t_d when the translational velocity of the end effector reaches zero is given by

$$t_d = t_c + |\mathbf{v}_b(t_c)| / |\mathbf{a}_e|. \quad (7)$$

The end effector constraints at time t_d for a direct catch are

$$\begin{aligned} \mathbf{p}_e(t_d) &= \mathbf{p}_e(t_c) + \mathbf{v}_e(t_c)(t_d - t_c) + 0.5\mathbf{a}_e(t_d - t_c)^2, \\ \mathbf{v}_e(t_d) &= \mathbf{0}, \quad \mathbf{a}_e(t_d) = \mathbf{0} \end{aligned} \quad (8)$$

The end effector orientation during the deceleration phase is determined by the balancing control, see subsection III-C.3. The trajectories for the first ($t_b < t < t_c$) and the second

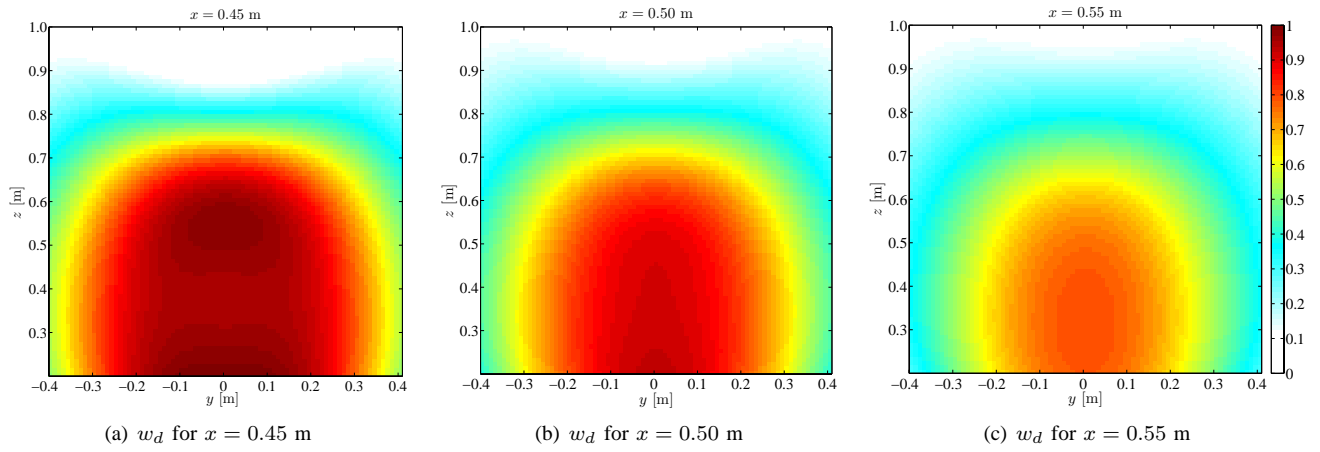


Fig. 3. Normalized dynamic manipulability measure w_d for different yz -planes: (a) $x = 0.45$ m, (b) $x = 0.50$ m, (c) $x = 0.55$ m. The measure is used for contact point selection based on the predicted ball trajectory.

($t_c < t < t_d$) catching phase are generated using fifth order polynomials

$$\mathbf{p}_e(t) = \mathbf{k}_5 t^5 + \mathbf{k}_4 t^4 + \mathbf{k}_3 t^3 + \mathbf{k}_2 t^2 + \mathbf{k}_1 t + \mathbf{k}_0 \quad (9)$$

where the coefficients $\mathbf{k}_j \in \mathbb{R}^3$ are determined by the initial and final constraints. The resulting trajectories of the catching point candidates are checked for workspace and acceleration constraints violations. A direct catch can be performed if one or more feasible solutions are found.

b) Indirect Catch:

If the direct catch algorithm does not obtain a feasible solution, the module checks whether an indirect catch is possible. Here, the end effector trajectory has to fulfill the following constraints at the initial contact time t_r :

$$\begin{aligned} \mathbf{p}_e(t_r) &= \mathbf{p}_b(t_r) - r_b \mathbf{n}_e(t_r), & \mathbf{v}_e(t_r) &= \mathbf{0}, & \mathbf{a}_e(t_r) &= \mathbf{0}, \\ \mathbf{n}_e(t_r) &= -\mathbf{s}/|\mathbf{s}|, \end{aligned} \quad (10)$$

where $\mathbf{s} = \mathbf{v}_b(t_r)/|\mathbf{v}_b(t_r)| + \mathbf{g}/|\mathbf{g}|$. With the assumptions A1 and A2, the last constraint ensures that the ball rebounds in vertical direction. For the indirect catch, there is an additional phase where the ball is again in free flight. The time t_c of the second contact at $\mathbf{p}_e(t_c) = \mathbf{p}_b(t_c)$ is calculated based on the ball velocity $\mathbf{v}_b(t_r)$ and the coefficient of restitution c_r . The constraints at t_c are identical to the ones of a direct catch with sole vertical velocity. The trajectories for the three phases of an indirect catch are also generated using fifth order polynomials and checked for feasibility. An indirect catch can be performed if one or more feasible solutions are found.

3) Balancing Control:

A balancing controller is used to keep the ball on the plate during the last catching phase ($t_c < t < t_d$). This control is activated at time t_c when the catching point is reached and a continuous contact between ball and robot is established. The position of the ball on the plate is estimated by measuring the forces and torques at the end effector. This eliminates the risk of occlusion that exists for vision-based ball tracking during contact with the end effector. For further details on the balancing control see [20].

IV. EXPERIMENT & RESULTS

In order to evaluate the control strategy, the following experiment was conducted: A basketball was thrown by a human operator into the robot's workspace. The ball trajectory was tracked by the vision system. Based on the predicted ball trajectory, an end effector trajectory for direct or indirect catching is created. In the following, experimental results for the ball trajectory prediction and the two nonprehensile catching methods are presented. To facilitate the presentation, the horizontal ball velocity is aligned with the y -axis.

A. Ball Trajectory Prediction

The results of the ball trajectory prediction for the vertical z - and horizontal y -direction for two different sample periods $T_s = t_b - t_a$ are illustrated in Fig. 4. The prediction is based on the recursive least squares estimate presented in Sec. III-B. In the experiments, a throw is triggered when $p_{b,z}$ exceeds 0.5 m. For the y -direction, both sample periods lead to accurate results: for a prediction horizon of $\Delta t = 0.5$ s, the deviations from the tracked are typically less than 0.05 m ($T_s = 50$ ms) respectively 0.02 m ($T_s = 100$ ms). For the z -direction, however, the accuracy improves significantly when using the longer sample period: for $\Delta t = 0.5$ s, the deviations from the tracked position reduce from 0.15 m ($T_s = 50$ ms) to 0.04 m ($T_s = 100$ ms).

B. Direct Catch

A sequence of snapshots for a direct catch is shown in Fig. 6: For $t_b < t < t_c$, the robot adjusts its position and velocity to match the constraints at t_c (snapshots 3-6). Then, for $t_c < t < t_d$, the ball is decelerated and the balancing scheme controls the end effector orientation based on the force/torque measurements (snapshots 7-15). A catch is considered successful if the ball comes to rest on the plate.

C. Indirect Catch

The ball and effector trajectory for an indirect catch is illustrated in Fig. 5. For the initial contact, the end effector adjusts its orientation according to (10) to let the ball rebound vertically. The ball is then caught with the second

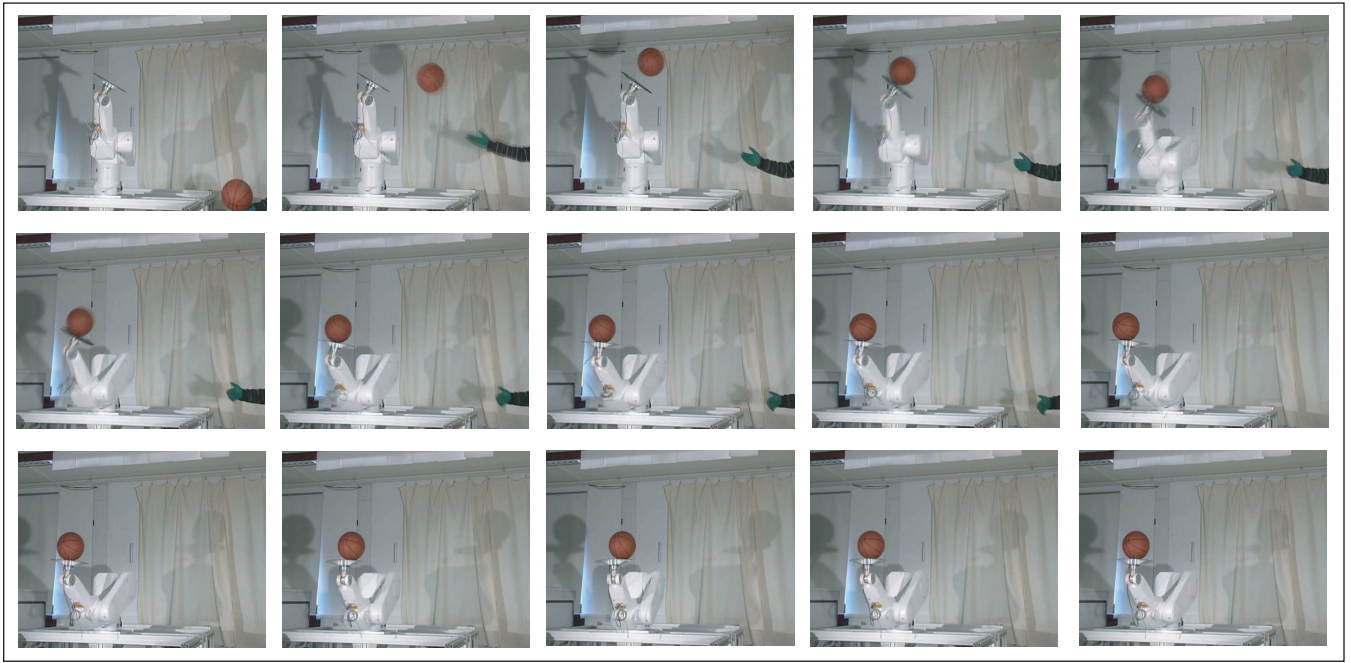
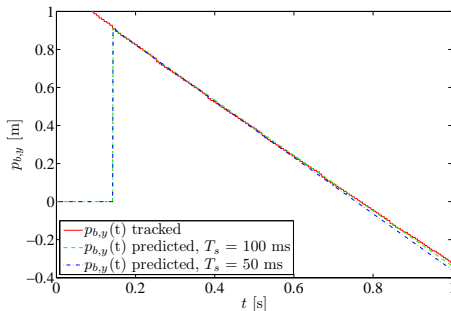


Fig. 6. Direct catch - experimental snapshots: initialization by human operator, catching and balancing (sequence from 0 s to 3 s)

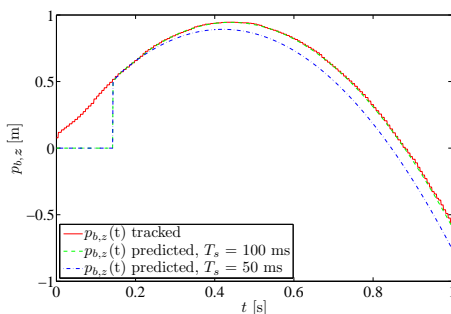
contact at t_c . Fig. 7 shows a sequence of snapshots for an indirect catch: After the free flight phase (snapshots 2-4), the initial contact occurs at t_r (snapshot 5), the ball bounces vertically (snapshots 6-7) and a continuous contact is established at t_c (snapshot 8). Then, the ball is decelerated and balanced on the plate.

V. CONCLUSION

Two strategies for nonprehensile ball catching with a six DoF industrial robot based on visual and force/torque sensor information were presented. The ball was tracked by a stereo camera system with appr. 150 Hz. A recursive least square algorithm was applied to predict the ball trajectory. The

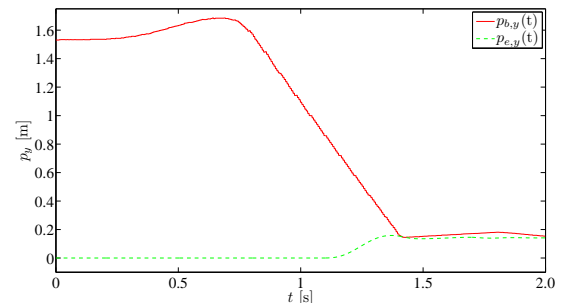


(a) Prediction for y -direction

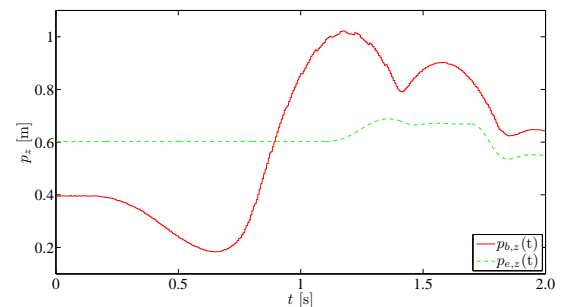


(b) Prediction for z -direction

Fig. 4. Actual and predicted ball trajectories for two different sample intervals T_s : (a) y -direction, (b) z -direction



(a) Ball and end effector trajectories in y -direction



(b) Ball and end effector trajectories in z -direction

Fig. 5. Ball and end effector trajectories for an indirect catch: (a) horizontal y -direction, (b) vertical z -direction

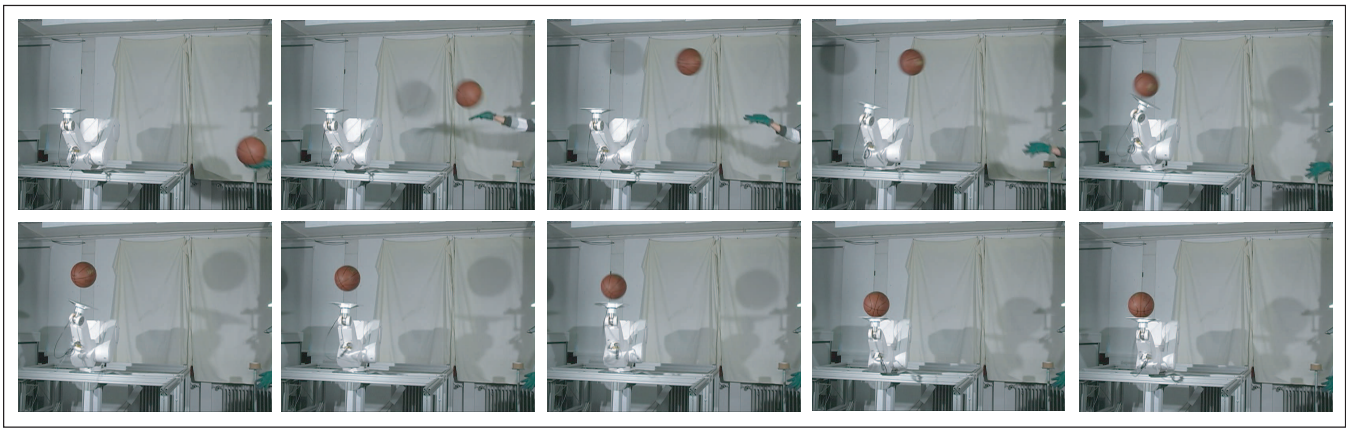


Fig. 7. Indirect catch - experimental snapshots: initialization by human operator, initial rebound and catching (sequence from 0 s to 2 s)

contact point was selected based on two criteria: the distance from mechanical joint limits and the dynamic manipulability measure. Both the direct and indirect catching method were experimentally evaluated. For the success rate, which is currently appr. 35%, there are two limiting factors: First, inaccurate ball trajectory predictions which occur if the sample period for the recursive least square estimations is too short. Second, trajectory tracking errors which occur if the planned trajectory is close to the hardware limits of the manipulator. For the indirect catch, the model-based prediction of the ball trajectory after the first impact based on the assumptions (A1, A2) was found to be an additional limiting factor for the success rate. This issue can be addressed by using the visual feedback to update the ball trajectory prediction after the first impact. The presented work considered the nonprehensile catching of spherical objects. Here, the object orientation does not affect the geometry of the contact surfaces. For other object geometries, this is generally not the case. Hence, the presented approach will be extended to track and predict the complete pose of the object. Then, a third selection criteria can be applied which evaluates the object orientation along the trajectory. In the future, experiments will be conducted evaluating the approach for objects with different shapes. Whether an indirect catch is feasible will depend on the physical properties of the particular object. In addition, a mechanical spring will be used to introduce an elastic coupling between robot and end effector. Such a design will help to relax the dynamic requirements for the robot and to increase the success rate.

VI. ACKNOWLEDGMENTS

The authors would like to thank Kwang-Kyu Lee, Xihua Lu and Andreas Achhammer for their valuable contributions. The first author gratefully thanks the German National Academic Foundation for their support.

REFERENCES

- [1] M. Mason and K. Lynch, "Dynamic manipulation," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems '93 IROS '93*, vol. 1, pp. 152–159, 26–30 July 1993.
- [2] K. Lynch and M. Mason, "Dynamic nonprehensile manipulation: Controllability, planning, and experiments," *International Journal of Robotics Research*, vol. 18, pp. 64–92, January 1999.
- [3] K. Anderson, "Real time intelligent visual control of a robot," in *IEEE Workshop on Intelligent Control*, 1985.
- [4] B. Hove and S. J.-J., "Experiments in robotic catching," in *Proceedings of the 1991 American Control Conference*, 1991.
- [5] M. Bühler, D. Koditschek, and P. Kindlmann, "Planning and control of robotic juggling and catching tasks," *The International Journal of Robotics Research*, vol. 13, no. 2, pp. 101 – 118, 1994.
- [6] R. Burrigge, A. Rizzi, and D. Koditschek, "Toward a dynamical pick and place," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems 95. Human Robot Interaction and Cooperative Robots*, vol. 2, pp. 292–297, 5–9 Aug. 1995.
- [7] U. Frese, B. Bauml, S. Haidacher, G. Schreiber, I. Schäfer, M. Hähne, and G. Hirzinger, "Off-the-shelf vision for a robotic ball catcher," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 3, pp. 1623–1629, 29 Oct.–3 Nov. 2001.
- [8] M. Riley and C. Atkeson, "Robot catching: Towards engaging human-humanoid interaction," *Autonomous Robots*, vol. 12, pp. 119–128, January 2002.
- [9] A. Namiki, Y. Imai, M. Ishikawa, and M. Kaneko, "Development of a high-speed multifingered hand system and its application to catching," in *Proceedings of the 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2003.
- [10] C. Smith, M. Bratt, and H. I. Christensen, "Teleoperation for a ballcatching task with significant dynamics," *Neural Networks, Special Issue on Robotics and Neuroscience*, vol. 24, pp. 604–620, May 2008.
- [11] H. Arai and O. Khatib, "Experiments with dynamic skills," in *Symposium on Flexible Automation*, 1994.
- [12] E. Aboaf, S. Drucker, and C. Atkeson, "Task-level robot learning: Juggling a tennis ball more accurately," in *Proc. of IEEE Int. Conf. on Robotics and Automation*, pp. 1290 – 1295, 1989.
- [13] 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.
- [14] M. Bühler, 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.
- [15] R. Andersson, "Dynamic sensing in a ping-pong playing robot," *Transactions on Robotics and Automation*, vol. 5, no. 6, 1989.
- [16] G. Bätz, K.-K. Lee, D. Wollherr, and M. Buss, "Robot basketball: A comparison of ball dribbling with visual and force/torque feedback," in *IEEE International Conference on Robotics and Automation*, 2009.
- [17] B. Siciliano, L. Sciacivico, L. Villani, and G. Oriolo, *Robotics - Modelling, Planning and Control*. Springer, 2008.
- [18] T. Yoshikawa, "Analysis and control of robot manipulators with redundancy," in *Proceedings of the 1st International Symposium on Robotics Research* (M. Press, ed.), pp. 735–747, 1983.
- [19] T. Yoshikawa, "Dynamic manipulability of robot manipulators," in *Robotics and Automation. Proceedings. 1985 IEEE International Conference on*, vol. 2, pp. 1033–1038, Mar 1985.
- [20] 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.