

# Accurate Object Throwing by an Industrial Robot Manipulator

Wilhelm August, Steffen Waeldele, Bjoern Hein and Heinz Woern

Karlsruhe Institute of Technology, Karlsruhe, Germany

wilhelm.august@kit.edu, steffen.waeldele@student.kit.edu, bjoern.hein@kit.edu, heinz.woern@kit.edu

## Abstract

In this work we have investigated the processes required for throwing objects at given targets with a KUKA KR-16 industrial robot manipulator. For this purpose a path planning algorithm and a throwing strategy were determined. A simple two-finger gripper which can be opened or closed pneumatically is used to hold the object during the acceleration phase. The position of the target is tracked by an optical tracking system. At first the experimental setup, the physical basics, the mathematical model, and the path planning algorithm are explained. Experimental results conclude the paper.

## 1 Introduction

In the manufacturing process it could make sense to throw workpieces from one position to another to save time and energy. Transportation of goods in this manner supports a serialized supply of production machines, in comparison to e.g. a pallet system where a large number of workpieces arrive simultaneously. By using throwing machines the production process may be designed more flexible. If the workpiece is needed in another place of the production hall merely the throwing parameters have to be computed and adjusted.

The idea of throwing goods as a transporting mechanism is not a new concept. There are several publications which investigate this kind of transporting. Basically three different approaches can be found: human like modeled throwing motions, mostly used in the humanoid robotics; human like link arms strengthened by springs; and the vast majority are spring shooting devices. Almost all of them also consider the catching process which is of course important for the further processing.

The human body is a formidable construct. World class track and field athletes are able to throw a two kilogram discus (woman size: 1 kg) about 75 m and a

800 g heavy javelin over 100 m long. Kim et al. [2008] used this fact as reference and starting point to model and mimic this whole body movement. They generated the overarm throwing motion of a 55 degree of freedom biped human model from multibody dynamics and optimization and considered some different throwing strategies based on this model. Riley et al. [2000; 2002] investigated throwing and catching processes with a 30 degree of freedom humanoid robot. They used a stereo color vision system to make juggling motions which included throwing and catching the ball. A similar work was presented by the working group of Nishiwaki [2008] in which catching processes by a humanoid robot were studied.

Long throw strategies were investigated by Ichinose et al. [2008] using a two-link arm with a spring strengthened passive joint. The model of their two degree of freedom robot arm is much easier in comparison to the humanoid robots mentioned above. They reached a horizontal distance of about 2.5 m. The construction of the throwing arm allows only one throwing strategy.

An overview of throwing objects in industrial applications is given in the publication of Frank et al. [2008a]. In this paper two different kinds of throwing and shooting devices were presented, the mechanical spring throwing device and the belt driven actuator. Furthermore a Cartesian robot was presented which is equipped with an optical tracking system. It was designed to catch objects thrown into its working space. Another publication of the same author has dealt with the differences of targeted throwing and capturing where the launching platform or the capturing device can be designed flexibly [Frank, 2008b]. Aside from considering flight trajectories of cylinder-shaped, i.e. non-spherical objects thrown with a pneumatic throwing device [Frank et al., 2009], Frank et al. presented in [2007] the concept of “flexible manufacturing systems” (FMS), considered some convenient capturing devices and in [2008c] also dealt with control methods for capturing flying objects. A highly sophisticated gripper, a multi-fingered high-speed hand

and its application to catching is presented by Namiki et al. [2003]. They dealt with different kinds of grasping techniques supported by a visual feedback control system.

Sameshima et al. [2008] made a further proposal for a new carrier system which contains a throwing and a catching robot on a linear slider. They used only the dynamic information (no optical tracking) of both robots to calculate the flight trajectory. In their work Sameshima et al. presented a prototype of a one-dimensional throwing-catching-system.

Beside some interesting YouTube videos no scientific publication on classical industrial robot manipulators could be found.

## 2 Experimental Setup

The core of the experimental setup is the industrial robot manipulator KUKA KR-16 (see **Figure 1**). It is originally designed for handling loads up to 16 kg. With its six-joint construction it is extremely versatile and broadly applicable. In the industry this universal robot is used in a variety of tasks including welding, drilling, grinding or transporting small loads.

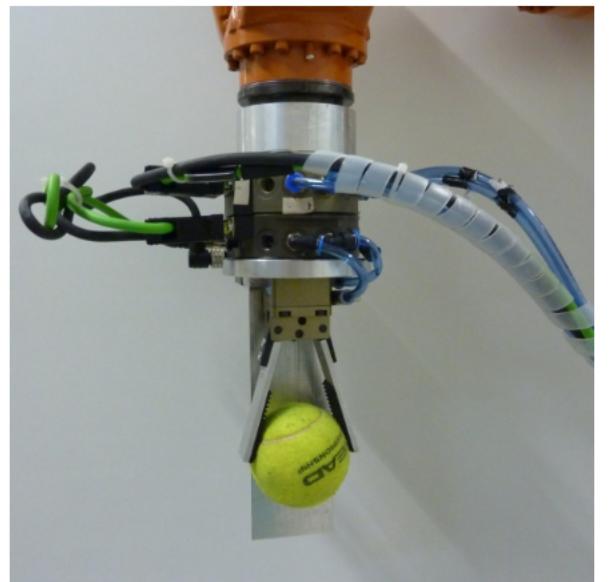
The gripper which is attached to the tool center point (TCP) of the robot manipulator is a pneumatic two-finger-angular gripper Schunk PWG 40S (see **Figure 2**) with an opening angle of  $20^\circ$  per finger and a recommended work piece weight of 0.9 kg. The closing and opening time is 0.01 s and the gripping force amounts to 230 N at 6 bar. Behind the gripper and the tennis ball there is a fixed aluminium bar. It is used to stabilize and to drive the ball for a short time interval after the gripper has opened.

The target is an aluminium bar on a common camera tripod with a tracking body on the side (see **Figure 3**). The height of the tripod can be varied from 0.6 m to 1.53 m. The position of the target is tracked by an optical tracking system (ART GmbH) which comprises six active IR-cameras and a server. Therefor a tracking body is attached to the side of the bar. The active cameras beam IR-light in the cell around the robot and detect the reflections of the tracking body consisting of five reflecting spheres. With the stereo information of the position of the reflecting spheres it is possible to calculate the position and the orientation of the tracking body. The position of the target is the input to the path planning algorithm which is explained in section 3.2.

As described in the following section 3.1 the initial velocity determines the degree of precision of the throw. It is important to come as close as possible to the computed absolute value and direction of the initial velocity. That means that the TCP of the robot has to follow the computed acceleration path and comply with every desired point in time. To achieve this strong requirements a



**Figure 1:** *KUKA KR-16 robot manipulator with a pneumatic two-finger gripper. In the upper left corner (orange circle) of the picture there is one of the six tracking cameras of the ART tracking system.*



**Figure 2:** *Pneumatic two-finger gripper Schunk PWG 40S holding a tennis ball. Behind the ball a fixed aluminium bar is mounted. It is used to stabilize the gripper opening process and to drive the ball for a short time after the gripper has opened.*

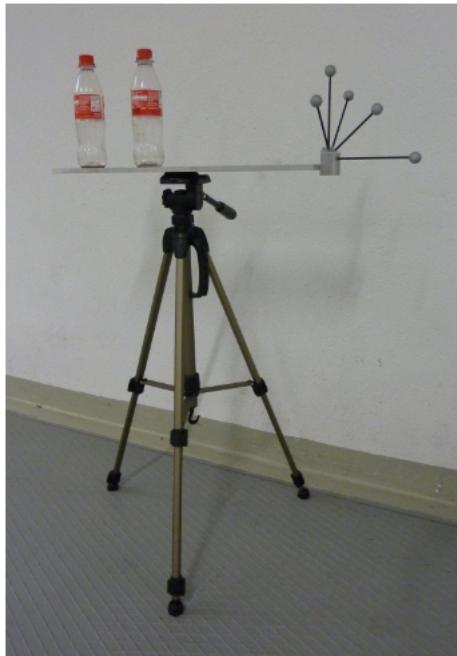


Figure 3: Target: an aluminium bar on a tripod with two plastic bottles on it. A tracking body with five reflecting spheres which is attached to the bar is used to track the target by the ART tracking system.

fast connection between the robot control server and the robot-PC is needed. Therefore the KUKA.Ethernet RSI XML technology is used (see **Figure 4**). It allows an exchange of XML packages containing the desired position generated by the robot control server which is running on the Linux-PC and current position data is sent by the Robot-PC every 12 ms. The internal KUKA robot manipulator control tries to reach the desired position given by the control server. If the jerk is too large it interrupts the connection and the motion stops immediately.

### 3 Theoretical Considerations

In this section the theoretical background of the throwing process and strategy is explained. At first the physical basics and the mathematical model are derived and in the second part of this section the path planning algorithm and the throwing strategy are commented.

#### 3.1 Physical Basics and Mathematical Model

The dynamics of a mass point that is thrown in a vacuum under gravitation is well known. The trajectory is a parabola and the whole flight is determined by the tangent angle and the absolute value of the initial velocity. With the zero acceleration in  $x$ - and  $-g$  in  $y$ -direction following time dependences result

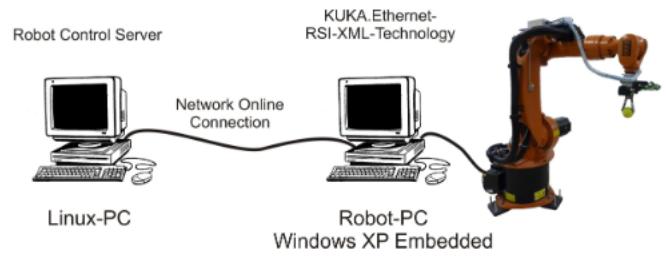


Figure 4: Robot manipulator control: a robot control server runs on the Linux-PC (left side) and maintains an online connection to the Robot-PC (Windows XP Embedded, right side). The KUKA.Ethernet RSI XML technology allows an exchange of XML packages within 12 ms with current (robot side) and desired position (server side).

$$x = v_{0x}t + x_0, \quad (1)$$

$$y = -\frac{1}{2}gt^2 + v_{0y}t + y_0. \quad (2)$$

Without loss of generality the starting position may be chosen to  $(x_0 = 0, y_0 = 0)$ . Combining the equations (1) and (2) yields the following parabola equation

$$y = -\frac{1}{2} \frac{g}{v_{0x}^2} x^2 + \frac{v_{0y}}{v_{0x}} x. \quad (3)$$

With  $v_{0x} = v_0 \cos \beta$  and  $v_{0y} = v_0 \sin \beta$  (see **Figure 3**) (3) can be expressed with  $v_0$  and  $\beta$

$$y = -\frac{1}{2} \frac{g}{v_0^2 \cos^2 \beta} x^2 + \tan \beta x. \quad (4)$$

As already mentioned the flight trajectory is determined by the initial velocity  $\vec{v}_0$  which comprises the absolute value  $v_0$  and the angle  $\beta$ . Since  $v_0$  and  $\beta$  can be varied in a certain interval, equation (4) has an infinite number of solutions. How to find convenient parameters  $v_0$  and  $\beta$  is described in the following section.

There are some interesting physical effects on flying objects in air like air friction (Magnus effect), imbalances and vibrations. All this effects are neglected in the mathematical model. A point mass flying along an ideal parabola is considered.

#### 3.2 Path Planning Algorithm

After the physical basics of the thrown mass point in vacuum have been discussed in section 3.1 the path planning algorithm will be explained in this section. Since all effects concerning air friction are neglected the initial velocity vector determines the trajectory and consequently the accuracy of the strike. An overview of the path planning process is given in **Figure 6**. The description of the path planning algorithm starts with the

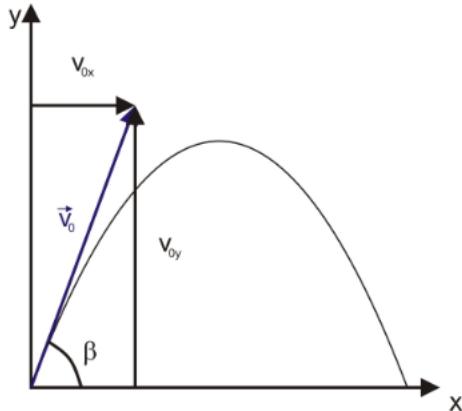


Figure 5: A draft of an ideal throw in vacuum. A point mass is shot out of the origin of the coordinate system with the initial velocity  $\vec{v}_0$ . The trajectory is a parabola.

acceleration, goes ahead with the parabola following and ends with the deceleration phase.

Equation (4) defines a family of parabola equations with the parameters  $v_0$  and  $\beta$  in a plane defined by the  $x$ - and  $y$ -axis. For a fixed target point (which is given by the tracking system in the experiment) there is an infinite number of solutions in a plane. The solution space gets a further dimension by rotating the plane over the gravity vector through the target point. A part of this solution space is located in the working space of the robot, i.e. can be reached by the kinematic chain. To find a convenient initial vector the three-dimensional Cartesian working space of the robot is discretized in concentrical cylinders over the axis A1 of the robot. The resolution of the discretisation determines the size of the search space and thereby the computation time. The optimal throwing angle in Cartesian space is the one with the smallest absolute value. It amounts to  $45^\circ$  to horizontal if the height of the target is the same as the release point. But not all of these points can be reached by the manipulator with the required velocity. Therefore all these points have to be tested in joint space (six axes - six dimensions). Here begins the planning of the acceleration path (see **Figure 6**, left part).

At first a release position is tested for the reachability in joint space. Therefore the Cartesian position is transformed by the geometrical inverse transformation into joint space. Secondly the initial velocity vector is transformed by the inverted Jacobian matrix

$$\dot{q} = J^{-1}(q)v, \quad (5)$$

with  $\dot{q}$  the six-dimensional velocity vector in joint space. With this information velocity limits can be checked and the corresponding starting and end point can be calculated in joint space. Afterwards the reachability of the

starting and the end point is checked and if their position is above the ground. The position and velocity limits can be found in the robot manipulator data sheet [KUKA]. The velocity limits are  $\frac{150^\circ}{s}$  for the axis A1, A2, A3,  $\frac{330^\circ}{s}$  for the axis A4, A5 and  $\frac{615^\circ}{s}$  for the last axis A6. Now the motion of the robot manipulator can be planned for each axis in joint space directly. Therefore trapezium acceleration ramps are used with a maximum slope value limited to  $\frac{120^\circ}{s^2}$  (determined by experiment). Since the path for each axis is planned independently in joint space, the profiles have to be synchronized to reach the release position at the same time.

The most suitable velocity vector  $\dot{q}$  in joint space can be chosen due to the first weighting criterion

$$j_1 = \min(\dot{q}^T(\beta)\mathbf{W}\dot{q}(\beta)), \quad (6)$$

with  $\beta$  the initial angle of the velocity vector to the horizontal and  $\mathbf{W}$  the weighting matrix

$$\mathbf{W} = \begin{bmatrix} w_{1,1} & & & & & 0 \\ & w_{2,2} & & & & \\ 0 & & \ddots & & & \\ & & & w_{6,6} & & \end{bmatrix}, \quad (7)$$

with  $w_{i,i} \geq 0$ ,  $i \in \{1, 2, \dots, 6\}$ , the weighting factors for each axis. The simplest weighting matrix is the identity matrix  $\mathbb{1}$ .

A further boundary condition demands that the orientation of the TCP has to be perpendicular to the velocity vector and its first time derivation additionally has to be zero. The angle around the initial velocity vector is arbitrary, but it should be chosen in a way that the flight trajectory of the ball and the deceleration path do not cross. The most convenient angle was determined by experiment. For most throwing processes it is useful to grab the ball from above.

To improve the accuracy of the throw and to compensate the imprecision of the gripper which does not open instantaneously an additional path planning step is done. In this step the robot manipulator tries to follow the ballistic trajectory of the object and the gripper opens during this phase of motion (see **Figure 6**, middle part). Following the flight parabola is very tricky. Because of the acceleration limits of the robot only a couple of points along the parabola can be followed. To increase this number, positions in the direction of throwing but also backwards are tested in time steps of 12 ms. The search stops if neither a next nor a foregoing point can be found. Every new point at the beginning of the parabola has to be tested whether reachable from the acceleration phase and every parabola added at the end of the parabola following phase has to be checked whether suitable as starting point for the deceleration phase.

# Path planning algorithm

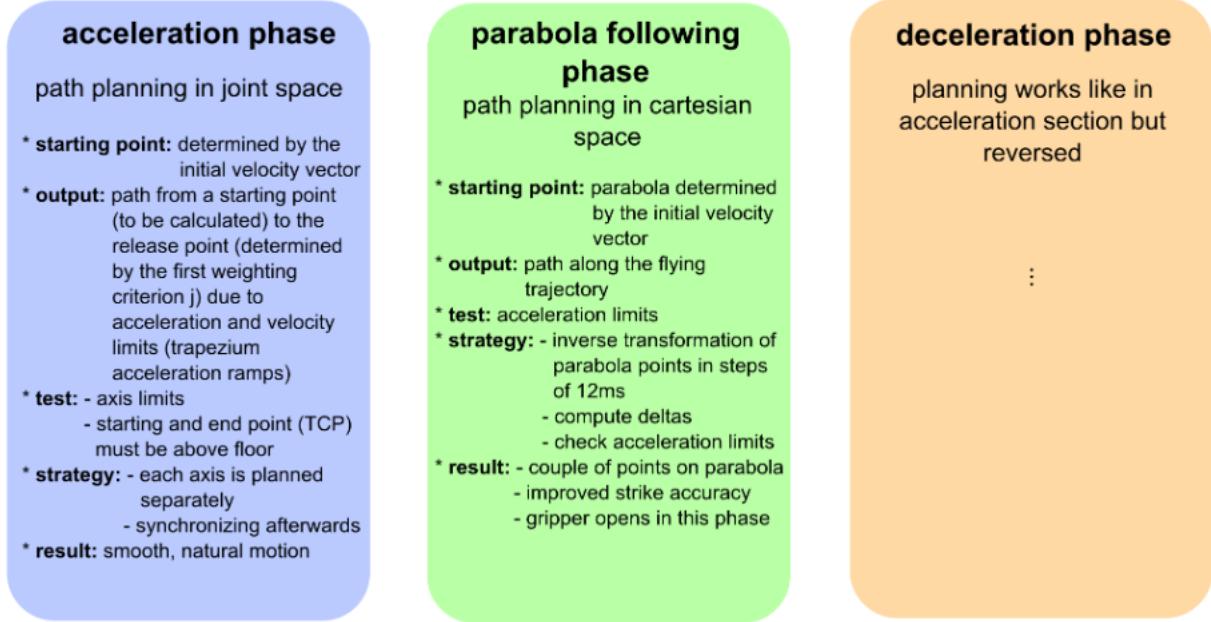


Figure 6: Scheme of the path planning process. It comprises three phases: acceleration, parabola following and deceleration phase.

The last phase of the path planning process is the already mentioned deceleration phase (see **Figure 6**, right part). It should be possible to slow down the robot with control after the TCP has left the flight trajectory. The planning is the same as in the acceleration phase but reversed.

At the end of the path planning process a second weighting criterion

$$j_2 = \min \left( \left( \begin{array}{c} j_1 \\ e^{-n^2} \end{array} \right)^T \mathbf{G} \left( \begin{array}{c} j_1 \\ e^{-n^2} \end{array} \right) \right) \quad (8)$$

is applied.

It takes two factors into account: the result of the first weighting criterion  $j_1$  and the number of points  $n$  which can be followed on the flight parabola. Again, the simplest weighting matrix is the identity  $\mathbb{1}$ .

## 4 Experiments and Results

In this section the results of the path planning algorithm and a throwing experiment are shown.

**Figure 7** shows the complete path in Cartesian space. The colors of the different phases are corresponding to the ones in **Figure 6**. The acceleration phase is marked blue. This part starts in the first blue point in the background of the picture and ends in the first point of the flight parabola. Here starts the second phase, the

parabola following phase, which is marked green and the parabola itself is marked dark blue in this picture. In this phase the gripper opens while following the ballistic trajectory of the object. In this seldom case, which is illustrated in **Figure 7**, 12 points of the flight trajectory can be followed. In most cases only a few points can be followed because of the restrictive acceleration and velocity limits. After releasing the object on the parabola, the TCP leaves the flight trajectory and slows down. This part of the path is marked by orange dots.

In **Figure 8** the desired path  $q$  (green) and the first two derivatives, the velocity  $\dot{q}$  (red) and the acceleration  $\ddot{q}$  (blue) of the axis A1 of a further experiment are shown. The three phases of the path planning are marked with blue, green and orange according to the acceleration, the parabola following and the deceleration phase respectively. As already mentioned in the previous section the acceleration of the axes is predefined as a trapezium ramp. The first and the last acceleration phases are planned in the joint space and all physical limits can be considered. In the parabola following phase the robot is forced to follow the parabola which is defined in the Cartesian space. This confinement leads to enormous leaps of the acceleration in the joint space and that is the reason why only a couple of parabola points can be followed by the robot manipulator. The biggest leap down to about  $-\frac{600^\circ}{s^2}$  is cut up in the diagram to make

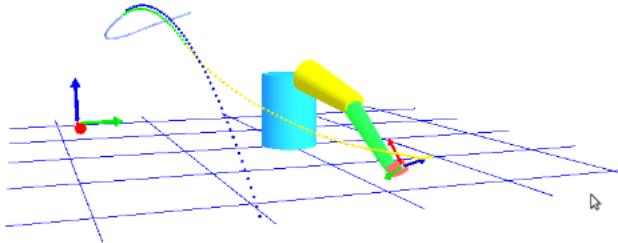


Figure 7: A draft of the complete path in Cartesian space. The acceleration phase is marked blue, the parabola following phase - green (the flight parabola - dark blue) and the deceleration phase - orange.

the interesting range viewable.

To compare the accuracy of the robot following the commanded values, the actual values are drawn as a black fine line in the same diagram. There is an apparent time delay of about 0.1 s between the set and the actual values. It results from the connection time delay and the inertia of the robot and merely means a shift of motion in time. The imprecision of following the desired path is mostly apparent visible in the actual values of the acceleration  $\ddot{q}$ . Two different effects of the internal robot manipulator control are visible in this diagram: a noise in the acceleration and deceleration phases and a smoothing of the acceleration peaks in the parabola following phase. Both effects affect the accuracy in a negative way. The desired initial velocity vector can not be reached exactly.

**Figure 9** shows a series of images of a successful throw. The starting point is close to the floor to exhaust the whole height from the ground to the point when the gripper opens for the acceleration phase. The gripper holds the tennis ball sidewise. That ensures that the gripper opens and does not cross the trajectory of the flying ball. Unfortunately the parabola following phase is short and therefore cannot be seen in this image series. In this experiment only four points can be followed on the flight parabola. The target is a plastic bottle on the tracked aluminium bar on the camera tripod. The thrown tennis ball strikes the bottle in the center of mass and knocks the bottle down from the bar. The time intervals between the pictures are not the same. The idea behind the choice of the pictures is to give an impression of the whole throwing process which lasts, similar as shown in **Figure 8**, about 3 s. The length of the distance between the target and the robot manipulator basis is about 2.5 m.

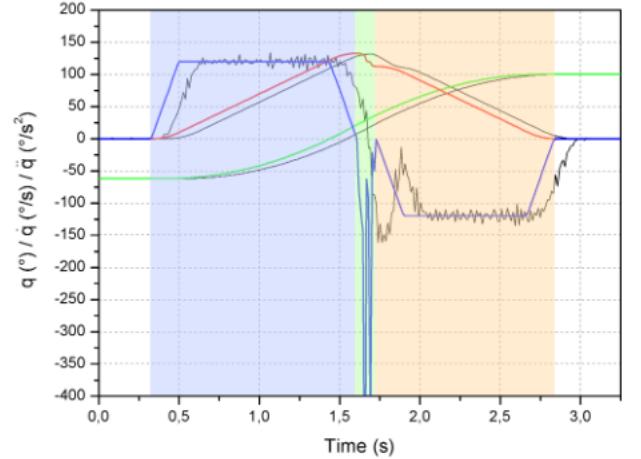


Figure 8: A comparison of commanded and actual values of the path  $q$  (green), the velocity  $\dot{q}$  (red) and the acceleration  $\ddot{q}$  (blue) for the axis A1. The commanded values are colored, the corresponding actual values are drawn as a black fine line. The phases of the path planning are marked accordingly: blue for acceleration, green - parabola following and orange - deceleration.

## 5 Summary and Conclusion

This paper presents a successful path planning strategy for object throwing processes with an industrial robot manipulator. The presented algorithm consists of three planning phases: the acceleration, the parabola following and the deceleration phase. The middle phase, the parabola following phase, helps to improve the accuracy of the strike and compensates the imprecision caused by the inaccuracy due to the jitter when opening the pneumatic gripper. A further improvement in the precision of the throwing process is the chosen orientation while opening the gripper. The following motion of the end effector will not cross the flight trajectory of the ball.

The characteristics and the advantages of the the presented algorithm are:

- It is easily transferable to other six-joint kinematics
- The acceleration and the deceleration phases are planed in joint space independently, therefore only four CPU-intensive transformations are needed
- The parabola following phase improves the accuracy of the throw
- The size of the search space and thereby the computation time depend on the resolution of the discretization

## References

- [Frank *et al.*, 2006] Heinz Frank, Norbert Wellerdick-Wojtasik, Bianca Hagebeuker, Gregor Novak, and Ste-



Figure 9: *Image series of the throwing process.*

fan Mahlknecht. *Throwing Objects - A bio-inspired Approach for the Transportation of Parts*. IEEE International Conference on Robotics and Biomimetics, pages 91-96, Kunming, China, December 2006.

[Frank *et al.*, 2007] Heinz Frank, Dennis Barteit, Norbert Wellerdick-Wojtasik, and Thorsten Frank. *Autonomous Mechanical Controlled Grippers for Capturing Flying Objects*. 5th IEEE International Conference on Industrial Informatics, pages 431-436, Vienna, Austria, November 2007.

[Frank, 2008a] Heinz Frank. *Design and Simulation of a Numerical Controlled Throwing Device*. AICMS 08, Second Asia International Conference on Modelling and Simulation, pages 777-782, Kuala Lumpur, Malaysia, May 2008.

[Frank, 2008b] Heinz Frank. *Determination of Launching Parameters for Throwing Objects in Logistic Processes with Direct Hits*. EFTA, IEEE International

Conference on Emerging Technologies and Factory Automation, pages 58-61, Hamburg, Germany, October 2008.

[Frank *et al.*, 2008c] Heinz Frank, Dennis Barteit, Marcus Meyer, Gregor Novak, and Stefan Mahlknecht. *Optimized Control Methods for Capturing Flying Objects with a Cartesian Robot*. IEEE Conference on Robotics, Automation and Mechatronics, pages 160-165, Chengdu, China, September 2008.

[Frank *et al.*, 2009] Heinz Frank, Anton Mittnacht, and Johann Scheiermann. *Throwing of Cylinder-Shaped Objects*. AIM, IEEE/ASME International Conference on Advanced Intelligent Mechatronics, pages 59-64, Singapore, July 2009.

[Ichinose *et al.*, 2008] Shigenori Ichinose, Shunsuke Katsumata, Shigeki Nakaura, and Mitsuji Sampei. *Throwing Motion Control Experiment utilizing 2-Link Arm with Passive Joint*. SICE Annual Conference, pages

3256-3261, The University Electro-Communications, Tokyo, Japan, August 2008.

[Kim *et al.*, 2008] Joo H. Kim, Yujiang Xiang, Rajankumar Bhatt, Jingzhou (James) Yang, Jasbir S. Arora, and Karim Abdel-Malek. *Throwing Motion Generation of a Biped Human Model*. BioRob, 2nd Biennial IEEE/RAS-EMBS International Conference on Biomedical Robotics and Biomechatronics, pages 587-592, Scottsdale, AZ, USA, October 2008.

[KUKA] KUKA ROBOTER GMBH. *KUKA Data Sheet KR16-2, The masterful movers in the low payload category*. Global Sales Center, Hery-Park 3000, Gersthofen, Germany <http://www.kuka robotics.com>, 2010.

[Namiki *et al.*, 2003] Akio Namiki, Yoshiro Imai, Masatoshi Ishikawa, and Makoto Kaneko. *Development of a High-speed Multifingered Hand System and Its Application to Catching*. IEEE/RSJ International Conference on Intelligent Robots and Systems, Vol.3, pages 266-2671, Las Vegas, Nevada, USA, October 2003.

[Nishiwaki *et al.*, 2008] Koichi Nishiwaki, Atsushi Konno, Koichi Nagashima, Masayuki Inaba, and Hirochika Inoue. *The Humanoid Saika that Catches a Thrown Ball*. 6th IEEE International Workshop on Robot and Human Communication, pages 94-99, Sendai, Japan, Sept./Oct. 1997.

[Riley *et al.*, 2000] Marcia Riley, Ales Ude, and Christopher G. Atkeson. *Methods for Motion Generation and Interaction with a Humanoid Robot: Case Studies of Dancing and Catching*. Workshop on Interactive Robotics and Entertainment, pages 35-42, Pittsburgh, Pennsylvania, USA, April/May 2000.

[Riley and Atkeson, 2002] Marcia Riley and Christopher G. Atkeson. *Robot Catching: Towards Engaging Human-Humanoid Interaction*. Kluwer Academic Publishers, In Autonomous Robots, Vol.12, pages 119-128, Netherlands, January 2002.

[Sameshima and Oguro, 2008] Yutaka Sameshima and Ryuichi Oguro. *Developing a New Carrier System with Throwing and Catching Robot*. ICCAS, International Conference on Control, Automation and Systems, pages 152-155, Seoul, Korea, October 2008.