

DISSERTATION

Bio-inspired Transport by Throwing System

An Analysis of Analytical and Bio-Inspired Approaches

Submitted at the Faculty of Electrical Engineering and Information Technology, TU
Wien in partial fulfillment of the requirements for the degree of
Doktor der technischen Wissenschaften (equals Ph.D.)

under supervision of

Prof. Dr. Dietmar Dietrich
Institute of Computer Technology
Technische Universität Wien

and

Prof. Dr. Uwe Janoske
Lehrstuhl Strömungsmechanik
Fakultät für Maschinenbau und Sicherheitstechnik
Bergische Universität Wuppertal

by

DI Martin Pongratz
Matr.Nr. 0326131
Sebastian Kneipp Gasse 9/20, 1020 Wien

Kurzfassung

Im Bereich der Produktionstechnik steigen die Anforderungen hinsichtlich Flexibilität. Diesem Trend müssen neben den eigentlichen Produktionsmaschinen auch die Transportsysteme Rechnung tragen. Ein möglicher Ansatz für flexiblere Produktionssysteme ist der Wurftransport-Ansatz, bei dem Objekte durch automatisiertes Werfen und Fangen transportiert werden. Für erfolgreiche Fangvorgänge ist eine präzise und schnelle Prognose der Flugbahn notwendig, um die Fangvorrichtung zeitgerecht zu positionieren.

Inhalt hier ist ein bionisches Wurftransportsystem, das aus einer Wurfvorrichtung, einem Kamerasystem, einem Prognosesystem und einem Roboter mit Steuerungsalgorithmen besteht. Für alle Teilsysteme, ausschließlich der Wurfvorrichtung, werden bionische Ansätze für die Informationsverarbeitung und Steuerung vorgestellt und diskutiert. Beim Auslesen der Kameradaten werden, ähnlich den Fixationen und Sakkaden beim Menschen, dynamisch verschiedene Teilbereiche des Bildes ausgelesen, um die zu verarbeitende Datenmenge auf die relevanten Daten zu beschränken. Das Prognosesystem basiert auf einem Erfahrungsschatz. Hier werden zwei Verfahren zur Repräsentation und Prognose von Flugbahnen eingeführt, die einzig auf bereits erfahrenen Flugbahnen die Prognose der aktuellen Flugbahn zulassen. Erfahrene Flugbahnen können hier sowohl simulierte Flugbahnen als auch aufgezeichnete Flugbahnen sein. Das Fangen wird mit einem, vom Menschenarm inspirierten, Roboter durchgeführt. Auch hier ist die Natur Vorbild für die Steuerungsalgorithmen und ermöglicht dadurch das weiche Fangen von geworfenen Objekten.

Das Gesamtsystem erreichte eine Fangrate von 86 % beim weichen Fangen. Durch das weiche Fangen ist es möglich, die Kräfte auf das geworfene Objekt beim Fangen deutlich zu reduzieren und damit auch den Wurftransport für eine größere Menge von Objekten anzuwenden. Der Rechenaufwand für die Prognose und Steuerung des Roboters kann von jeweils einem herkömmlichen PC zeitgerecht bewältigt werden.

Mithilfe der vorgestellten bionischen Ansätze ist es in jenen Bereichen, in denen die menschliche Einschätzung den maschinellen Vorhersagen überlegen ist, möglich präzisere Prognosen durchzuführen. Eine Weiterentwicklung der Auswertung der Kameradaten mit direkter Integration auf dem Sensor ermöglicht die Steigerung der Effizienz der Bildübertragung vor allem in Domänen, wo einzig Veränderungen oder Bewegungen relevant sind. Auch bei der erfahrungsschatzbasierten Prognose können zum Beispiel durch eine Glättung bzw. Filterung noch Steigerungen der Prognosegenauigkeit erreicht werden.

Abstract

Current and future production facilities are confronted with rising demands for flexibility. Besides the impact on the production machinery also the transportation systems have to accommodate this. A possible solution for more flexible transportation systems is transport-by-throwing, where objects are relocated based on automatic throwing and catching. Successful catching requires an accurate and timely prediction of the flight trajectory in order to place the catching device on time.

The content of this work is a bionic transport-by-throwing system, which is formed out of a throwing device, a camera system, a prediction system and a robot with the related controller system. For all subsystems, except the throwing device, bionic approaches for information processing and control are introduced and discussed. The image processing is based on fixations and saccades, similarly to the human, where the read-out areas of the cameras are dynamically adapted. This reduced the amount of data which has to be processed on the relevant data. The prediction system is working on the base of an experience database. Two procedures to represent and predict trajectories are introduced, which work only on previously experienced/recorded flight and allow the prediction of the current flight. The used experience can contain real as well as simulated flights. Catching the thrown object is done with a robotic arm that has a kinematic chain similar to the human arm. Soft catching of human is the archetype for the control algorithms here and allows soft catching, with low differential velocity, of the object.

The whole system achieves a success rate of 86 % for soft catching. Soft catching allows to reduce the forces on the object during catching significantly. This opens the transport-by-throwing approach to a wider range of application. The calculation demands for prediction and control of the robot is done by two personal computers within the timing constraints of the overall system. The introduced prediction algorithm allows to predict the flight of a ball with an accuracy on par with the state of the art prediction approaches. Future research targeting smoothing/filtering of the acquired/stored trajectories is one of the possibilities to enhance this approach further.

Acknowledgements

During the time working on this thesis and more general my whole live I've enjoyed the accompany of many people due to my interests in sports, technology and science. At this stage I would like to say thanks to all of the for the experience, I made together with them no matter if it was charming or conflicting at first sight. Finally, the interactions with others give each individual feedback and allow to adapt and grow. The two main persons responsible for my growth are my parents, and I thank them for the environment they created for me that gave certainty but also raised some challenges for me. I'd also want to thank my sister for her support and open ear throughout my life. During our joint time of this work my wife has helped me with an open ear and spent motivation on me to support me finishing.

This thesis is only possible because of Prof. Dietmar Dietrich's work that raised my interest in the Institute of Computer Technology at the TU Wien. I thank him for the discussions and remarks he gave throughout this work even as an emeritus. I would also like to thank Prof. Uwe Janoske for the cooperation and discussion in the field of this work. Also, my colleagues here at the Institute deserve thanks and especially the discussions with Friederich Bauer and Prof. Axel Jantsch, even if some of them were off-topic, were inspiring and helped me a lot to finish this thesis.

Last but not least I would like to thank the students here at the TU Wien, that cooperated on subtopics, especially Manuel Huber, Maximilian Götzinger and Bálint Toth regarding the practical experiment.

TABLE OF CONTENTS

1	Introduction	1
1.1	Motivation	1
1.2	Problem Statement	3
1.2.1	Transport by Throwing	3
1.2.2	Scientific Challenge	5
1.3	Proposed Methodology	7
2	State of the Art and Related Work	9
2.1	Material Transfer in Production Facilities	10
2.2	Research on Robotic Catching	15
2.2.1	Catching Emerging from Grasping	15
2.2.2	Larger-Scale Catching	19
2.3	Biological Archetype for Catching	38
2.3.1	Analysis of the Visual System During Catching	39
2.3.2	Bio-Inspired Information Processing	39
2.3.3	Analysis of the Catching Movement	41
2.4	Promising Solutions	44
3	Proposal	46
3.1	Image Acquisition	47
3.2	Bionic Associative Memory	49
3.2.1	Holistic Associative Memory	50
3.2.2	Progressive Associative Memory	53
3.3	Catching Movement Planning	59
4	Simulation and Implementation	67
4.1	Environment and Nomenclature Definition	68
4.2	Benchmark Process	70
4.2.1	Physical Description	70
4.2.2	Analysis of the used Equipment	72
4.3	Soft Catching	89
4.3.1	Constraints	90
4.3.2	Interception Point Determination	91
4.3.3	Velocity Maximization/Synchronization	93
4.4	Simulation	96

4.4.1	Trajectory Prediction	96
4.4.2	Catching Movement Simulation	99
4.5	Implementation	102
5	Results	106
5.1	Simulation	106
5.1.1	Physics-based Prediction Results	107
5.1.2	Bio-inspired Prediction Results	109
5.1.3	Discussion of the Simulation Results	121
5.2	Implementation	123
6	Conclusion and Future Work	126
6.1	Conclusion	126
6.2	Future Work and Outlook	128
A	Appendix	132
	Literature	135
	Internet References	144

ABBREVIATIONS

2D	2-dimension(al)
3D	3-dimension(al)
AoI	Area of Interest
DoF	Degrees of Freedom
KRC	Kuka Robotic Controller
LWR	Leight Weight Robot
RANSAC	Random Sample Consensus
RTS	Rauch-Tung-Striebel
RWCS	Robot World Coordinate System
SCCS	Stereo Camera Coordinate System
TCP	Tool Center Point
UKF	Unscented Kalman Filter

1 INTRODUCTION

The processing power of modern computer system has increased tremendously during the last decades. In spite of the possibility to process huge amounts of data within seconds with computers, humans are still required to execute certain tasks where pure calculation power is not the absolute requirement. Most of the related tasks for humans include abstraction, generalization, and prediction of data. One example task is video surveillance where humans identify abnormal or dangerous behavior based on their experience. Humans ability to identify regularities and irregularities in "data streams" and abstract a rule for regular or normal sequences is used here.

Besides the raw processing power of modern computer architectures and computer systems, the ability to perform these tasks by artificial systems is extremely limited and a current research topic in a number of domains (by example video surveillance and elderly support to name two of them). While processing power does not seem to limit the application of computer systems for these tasks, the basic principles for data acquisition [Vel07], data representation [Vel07], abstraction [Vel07] are limit factors. Prediction is one example that can improve the human-robot interaction [KS11, pg. 1]. In these domains, the application of human or biological information-processing principles can lead to more sophisticated systems.

Here the principles of human abstraction, memory and prediction are applied to the research field of automated material transport by throwing and catching. A trajectory prediction system based solely on previously experienced trajectories is introduced and the prediction performance (prediction accuracy over time) is compared to state of the art physical model based prediction systems. The Evaluation of the prediction accuracy under different environment states is done to identify advantages and drawbacks of this information processing principle.

1.1 Motivation

Biological systems like the human body use a number of sophisticated mechanisms to deal with the enormous amount of sensory data from all over the body. The main principle here is the abstraction that is used for every sense in the human body. An example for the abstraction taking place for sight is the comparison of the number of cones and rods inside the eye (overall more than 120 million) with the number of nerves (1.2 million) that connect the eye to the brain for further information processing [9]. According to [Foe93, pg. 42ff] [Foe99, pg. 37f] the layered structure of the neurons inside the eye allow to combine information of neighboring photoreceptor

cells to obtain a higher level abstraction of the visual data containing objects like edges, areas, and similar ones.

Another versatile mechanism of the human information processing is the association of previously acquired experience with the current sensory input. Association has a huge number of applications for humans ranging from the recognition of persons or personal objects to the prediction of future events or behavior. The first two examples are mainly relevant for sight and deal with an association of lower complexity. Only the special attributes (also called features) of an object are used to find similarities and thus also the associated memory of the object. For face recognition by example, humans evaluate a number of points in a face where they look for a certain time (fixations) [FHH+00]. The other example for an association, where also the future is predicted, is a more complicated task. Examples here range from predicting the reaction of a person to a sensible information to the prediction of a balls trajectory for an interception in sports. While the prior case demands in-depth knowledge of the person and its personality, the ball-catching task demands experience of a number of similar throws and a good association with historical trajectories with the actual one. The number of repetitions for achieving a certain quality of a task also depends on the intensity of the experience during practice for humans [Col08, p. 65ff].

The previously mentioned mechanisms and principles are mainly used in combination. By example, a child that is able to catch a green ball thrown at it is also able to catch a red ball thrown with the same velocity and trajectory. Here the combination of abstraction (either the red as well as the green ball is a ball) and the association of previously experienced trajectories of balls is combined and used to predict the interception position and time that allows successfully catching the ball. In addition research by Sloman suggests that the role of association, prediction and online-perception for vision and the resulting (re-)action is ignored by nearly all vision researchers [10]. Humans continuously use (short-time-)predictions or expectations to check their sensory input for plausibility [Hoh14]. In case, a bottle is put over the edge of a table the expectation is that the bottle falls to the ground and breaks there in case it is made of glass. Another example is a ball that's thrown between two people. Consider an object is blocking a part of the view on the flight for an observer. The observer still will expect the ball to enter his field of view based on the initial trajectory of the ball and his previous experience of thrown ball's trajectories. His assumption for the location where the ball will enter his field of view again might even be exceptionally accurate.

Regarding visual information, different technical solutions for abstraction have already been proposed and implemented. Solutions for abstraction of different degrees is possible. Examples are edge-detectors that allow extracting contours from images, feature-detectors that identify distinctive areas of images and allow to find corresponding areas in other images or object libraries and segmentation algorithms that spatially divide images into related subsections. Regarding technical maturity, these algorithms might not be on par with the methods applied in biological systems, but the basic functionality is existing and working. In general, it is possible to find a ball inside an image, detect the contours of the ball, which is an abstraction, and derive two-dimensional information about the center of the ball, which is another abstraction. This might not be possible in every case considering lighting, background/foreground differences and ball distance/size but it is possible under well-adjusted conditions. Analyzing human information processing and applying it to the problem of abstraction still has a lot of potential for future research but this will not be the main topic here.

In the field of association of experience with current sensory input the state-of-the-art is less advanced. Similar to abstraction, the application of biological principles or the mimicry of the

functions of biological systems for the task of experience creation and association have the potential to substantially level the abilities of technical systems. Establishing a functional model for experience and association inspired by the human archetype will be the core area here. Due to the given task of trajectory prediction for transport-by-throwing the requirements for this data processing model are a fast association of related experience with the current event, efficient management of the emerging database of movements and adequate abstraction and generalization of information.

The research question in this work is whether a system, solely based on abstracted information of previous flight trajectories, can outperform a state-of-the-art model-based prediction system for the task of trajectory prediction with respect to computation cost and accuracy of prediction. In addition to the two mentioned aspects also, the influence of undocumented impact factors like air streams or deviations of the object's flight properties will be discussed.

Such a system will allow mimicing biological information processing for this task. This means that the flight trajectory will not be predicted based on the consideration of gravity, aerodynamic forces or other physical aspects but only be based on the "knowledge" of the prediction system that is following the human way of flight trajectory prediction. No baseball player would consider calculating the flight trajectory of a baseball based on differential equations or other equations, his experience and record of flying balls will enable him to catch the ball with a high rate of success. The proposed processing model is derived from this behavior.

1.2 Problem Statement

Material transport, in general, deals with the relocation of mass. Properties like efficiency, variability and applicability are main attributes to transport systems. State of the art production faces the challenge of the conflicting requirements of automation and flexibility [PEG05] [GRF06, p. 2]. Traditional transportation system shows disadvantages on these attributes under some circumstances. Especially at junctions of a transport system more flexible approaches to transportation may show great potential.

1.2.1 Transport by Throwing

Transport-by-throwing has been proposed by Frank [FWWH⁺06]. The basic principle of automatized throwing and catching significantly increases the flexibility due to the lapse of additional infrastructure for material handling like conveyor belts. Regarding the three main shapes of flexibility, namely layout-flexibility, throughput-flexibility and the flexibility of the transported material proposed by Günthner [GW02, p. 1f], the transport-by-throwing approach has excellent properties regarding the layout flexibility and the throughput flexibility. In case, the production line is composed of modular production units the layout of the whole line is only restricted by the range of throwing. In the case of longer transportation paths routing via other production units is possible (similar to multi-hop routing in communication networks). All other (modular) production units can be used for transporting in case they are within the reach of the origin unit. In addition to the transportation on the same height-level also the transfer to different levels is possible without additional infrastructure. Regarding the flexibility of transported material, research has only dealt with highly symmetrical (point-symmetrical, axial-symmetrical) objects like tennis balls, cylinders or similar objects [FBM⁺08, BFK08, FMS09, BFPK09, PKFB10]. On the

other hand, challenges of the transport principle lie in the trajectory and orientation prediction and soft and harmless catching at the destination site. The task of successfully transporting an object between two spatial different locations is divided into four subtasks [FWWH⁺06, 92]. The four subtasks can be seen in Figure 1.1.

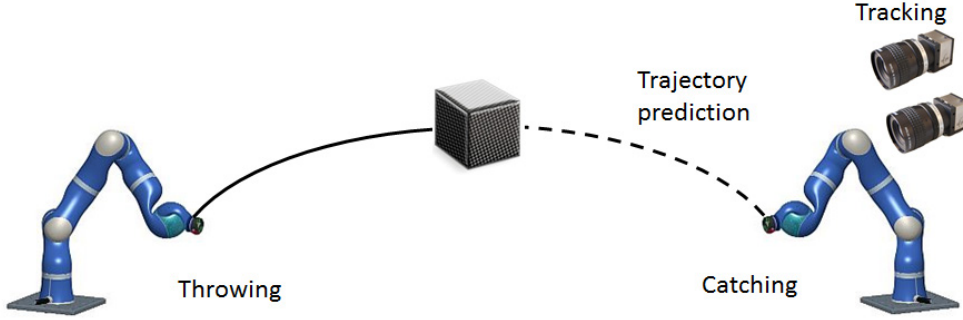


Figure 1.1: Overview of transport-by-throwing approach

Throwing

For the throwing site, the main task of the throwing unit is to accelerate the object to a suitable launching velocity and direction to reach the catching area of the catching site. Object's aerodynamic properties determine the suitable flight trajectory and thus the required launching parameters. Variances in the launching parameters can lead to a trajectory that does not allow the object to be caught at the destination site, thus, the requirement for accurate throwing in dependency to the catching area of the catching system exists [FBM⁺08, p. 1]. To minimize the stress on the thrown object the forces during acceleration have to be minimized.

Trajectory Prediction

Accurate flight trajectory prediction is necessary in order to position the catching device on time at the interception position. An object's flight trajectory depends largely on the launching parameters and the object's aerodynamic properties. Variances in the trajectory emerge due to small deviations in the launching parameters as well as deviations in the medium's properties (e.g. air streams, air temperature) [FWWH⁺06, p. 92] [FBM⁺08, p. 978]. An illustration of this problem is given in [FWWH⁺06, pg. 94]. A more detailed analysis with the determination of deviation distances for different objects is given in [BFK08, pg. 894f]. Most of these influencing factors can not be measured and considered during the flight time of the object. Consider measuring the magnitude and direction of the air streams along the object's trajectory - this is technically not feasible. Thus in addition to the information about the launching parameters of the object, the current object's motion state is used to update the current prediction for the flight trajectory.

Tracking

As mentioned in the previous paragraph additional information is necessary to predict the flight trajectory of an object with the required accuracy to catch the object at the destination site. Due to the object's motion visual sensors (e.g. cameras) are used mainly for the task of tracking

[KMK⁺10, PKFB10, Bar10, LRÅJ09]. Mainly stereo vision setups are used for this task but also research work based on single cameras exists [Bar10]. One approach is dealing with different numbers of cameras and also considers unsynchronized images from multiple cameras is presented in [LRÅJ10].

Catching

The information from the tracking and trajectory prediction system is finally used to position the catching device on time at the interception position and catch the object out of the air. While the forces on the object during the free flight are negligible, the catching-event is putting external forces on the object. In contrast to throwing, where the object is under full control of the throwing device during the acceleration phase, this is not valid for the catching site, especially at the instant of impact. Most of the current implementations primary focus on positioning the gripper at the predicted interception position [BSW⁺11] [Bar10]. High accuracy of the trajectory prediction in the early phase of the object's flight allows to adjust the velocity of the catching device to the velocity of the object and thus minimize the forces on the object during catching. This approach is covered only in a small number of approaches [HS91] (compare first paragraph in Section 2.2.2.1) and most recently with a cylindrical thrown object that has a stable flight [Fra12] (compare fourth paragraph in Section 2.2.2.1).

1.2.2 Scientific Challenge

For using transport by throwing on in a real production environment challenges in all four previously introduced areas have to be mastered. Especially in the areas of trajectory prediction, tracking and catching current approaches are not practical for general usage. For example work by the national aeronautics and space research center of the Federal Republic of Germany uses a cluster with 32 CPU cores for the planning of the first optimal movement path to the catching position and stops the search for the optimal solution after 60 ms [BWH10, pg. 2]. For updating the movement planning after every frame one of three cores of a 2.2 GHz Quad-Core-Xeon CPU is used in a round-robin scheme with a worst case running time of 60 ms per core. Considering such a setup for each transportation link renders transport by throwing to computational expensive for general usage. The combination of image acquisition, processing and object detection, trajectory prediction and catching motion planning needs to be more efficient regarding computational costs.

Still, while using such a potent system, the achieved success rate is $\approx 80\%$. Here the main question is why the analytical/physical approaches only achieve this success rate. Reasons, given in the literature, are the accuracy of the tracking system, the prediction accuracy and the challenges of the robotic actuator.

Visual Input

Experiments on the influence of image resolution and frame rate on the accuracy of the predicted trajectory [Pon09, pg. 62ff] show that both a high resolution and a high frame rate improve accuracy. On the other hand, an increase in either of them increases the amount of data to be processed within a given time as well. This yields in a conflicting state of interests. Methods to increase frame rate and resolution while keeping the processing requirements within a given envelope are required.

Experience Based Prediction

Calculating the flight trajectory based on computational fluid dynamics is the most accurate technical solution for predicting an objects flight path. This approach, on the other hand, is extremely expensive regarding computational power that also results in significant energy consumption. Simplified motion models for simple thrown objects have been introduced by [Bar10, pp. 115ff] [Pon09, pp. 38ff] [LRÅJ09, p. 3] that consider point-symmetrical objects like a tennis ball. These approaches show disadvantages in case the motion model does not consider all forces influencing the flight trajectory (e. g. spin of the tennis ball) [Pon09, p. 66]. More recent work has also dealt with axial-symmetrical objects [FMS09]. Experiments examining and using the aerodynamic effect of shoulder stabilization as well as a rotation along the symmetry axis are presented.

Here a bionic experience based prediction system will be introduced and implemented in a simulation environment as well as for practical experiments. The challenges for this system are:

- What is a suitable representation of the data?
- How to keep the memory consumption of the "experience" low? (depends largely on the representation)
- How to associate current processes or events with previous events?
- How can the general application of the data to current trajectories be achieved?

The solution for these challenges shall be found based on human- or bio-inspired information processing because:

- Humans are able to generalize the experience they have.
- A human associates historical events, a human recognizes people, a human is able to find orderliness/principles.
- Human build abstract models (physics, chemistry, mathematics).
- Humans accurately, efficiently and nearly instantaneously associate experience (e.g. a balls trajectory in tennis).

Biological trajectory prediction systems show all the aspects that are necessary for the task of transport-by-throwing and thus deal as an archetype for the experience based prediction system. Primary functions of the human trajectory prediction will be identified, and a hypothesis for the information processing model will be tested in a simulation environment and a practical experiment.

Based on the attributes of biological trajectory prediction systems the derived model for trajectory prediction is expected to

- Have advantages in terms of accuracy/computation power over state-of-the-art trajectory prediction models/systems
- Allow to use less sophisticated camera systems (lower resolution, lower frame rate thus lower cost)
- Provide feedback to verify if the derived model is suitable to describe the trajectory prediction in biological systems

Catching Movement

Another relevant aspect is the motion planning of the catching device. Usually, a catching device is mounted on a robot with varying number of Degrees of Freedom (DoF) ranging from a gantry robot [Bar10, pg. 11f] with 2 DoF to state of the art robots with 7 DoF [BWH10, pg. 1]. The used robot defines the possible movements (in the case of the gantry robot only positioning within a plane is possible) and, on the other hand, raises requirements for the movement planning. For each of the joint, the related position has to be calculated which raises the computational demands and, on the other hand, also the demands for the algorithm determining the joint's position. Additionally, the aspect of robotic mastering (setting the robot's parameters in the best possible way to have accurate Cartesian positioning based on the seven rotational joints possible) is more challenging for systems with higher DoF. By example, a 7 DoF System like the KUKA LWR 4+ used for this work has more DoF than needed to reach a particular position (3 DoF) with a certain orientation (3 DoF). This additional DoF can be used to reach the same position and orientation with different internal positions of the robot or a zero space movement while staying in the same place.

Also, more complex systems (systems with higher DoF) allow to align the trajectory of the catching device with the trajectory of the thrown object and thus enabling to catch the object in a soft way. This catching reduces the forces acting on the object during catching. Here it will be referred as soft catching. A benefit of soft catching is that the transport by throwing approach can be applied to a higher number of objects. For implementing soft catching efficient catching movement planning and timely synchronization of the trajectories is necessary.

1.3 Proposed Methodology

When approaching the challenges of transport by throwing the environment for the application have to be analyzed initially. This requirements analysis allows narrowing down the problems and difficulties that have to be targeted throughout the process of developing a transport by throwing system. The current state of the art transportation systems will be analyzed, and the benefits and possible disadvantages of a transport by throwing system will be derived.

An investigation of current research work will be used to identify related problems and their solutions. Open issues will be specified and basic information and principles to master these problems will be collected, this concludes the part of investigations. An analysis of the state of the art trajectory prediction and especially modeling will be done. Problems in this field will be identified.

Proposal

Based on the current research work a system architecture for a bionic transport by throwing system will be proposed. This system includes bionic aspects regarding image acquisition, image processing, trajectory prediction and catching movement planning. Using biological archetypes a more efficient way to calculate an object's position based on visual information from a set of cameras is presented. For the aspect of trajectory prediction, two approaches for experience based prediction will be proposed and discussed. The foundations are in the field of studies targeting the human information processing. While most of the studies cannot be validated or proven the approach here is also used to test the hypothesis and provide feedback on the related models.

Their respective advantages and disadvantages compared to each other and considering the state of the art physics-based prediction approaches will be discussed. Furthermore, an efficient method to determine a robot's catching movement will be proposed and reviewed.

Evaluation

The proposed approach for trajectory prediction will be evaluated in a simulation and real world experiments. To allow a realistic evaluation of the algorithms the analysis of a stereo camera system's accuracy regarding spatial position detection will be done. This data is required input for the simulation to model the error of the position detection introduced by the camera systems. Due to the fact that any real world system is only able to process this measured data, including the errors, the simulation has to consider this errors as well for position detection. The advantage of using a simulation, where the ground truth is known, will be used to determine the impact of the position detection error on the final prediction error, thus enabling to give statements about required position detection accuracies.

The process of experience build-up and the influence of the number of datasets for the resulting prediction accuracy will be examined. The possibility to expand the data set for the real world experiments with simulated data will be discussed and reviewed.

The additional steps to apply the proposed prediction methods to real world experiments with a linear-throwing machine, a set of two cameras, a standard PC including general purpose graphics processing units for image processing and trajectory prediction, as well as a KUKA LWR 4+ industrial robot, will be examined. Access to the robot is possible because this work is embedded in the KOROS initiative at the TU Wien. In this initiative, nine institutes of 4 faculties at the TU Wien cooperate in the field of robotics and human-robot interaction. The hardware infrastructure, consisting of 2 KUKA LWR 4+ robotic arms, one Aldebaran Romeo, and additional equipment was funded by the WWTF in 2010¹. Here included is the application of the proposed bionic concepts to the vision system, the prediction based on experience including the built-up process with real world data and the application of the soft catching movement to the robot. For the latter task, the challenge of synchronizing the status and command data of the robot with the image acquisition process of the cameras has to be mastered.

An evaluation based on a comparison of the bionic approaches with state of the art physics-based prediction algorithms will be done. This evaluation will be made twofold - in simulation, and the real world experiments will be used to test the most suitable developed prediction algorithm.

Reasoning

The results from both approaches and both test environments will be analyzed and discussed giving the foundation for the identification of future work and possible future improvements as well as a wider field of application.

¹KOROS proposal for UIP 2010 WWTF, granted on 2. Feb. 2011 (TU-Kooperationszentrum)

2 STATE OF THE ART AND RELATED WORK

In the following sections, the current state of the art for the three main parts of the transport-by-throwing approach will be analyzed and discussed. The first section deals with transportation systems and material flow in current production facilities and outlines advantages and disadvantages. Trends in production and transportation systems, as well as projections, are discussed. Furthermore potential benefits of the transport-by-throwing approach over current implementations are outlined.

Based on these realizations in the initial section of this chapter research work in the areas of robotic catching and biological archetypes for catching will be discussed. Furthermore promising solutions to the challenge of robotic catching will be discussed. The content of the sections in detail:

The second section, robotic catching, will deal with scientific work on grasping and catching with robotic systems. Approaches targeting catching of objects thrown from distances ranging from a few centimeter up to some meters will be discussed in two separate subsections. In the first subsection mainly approaches that do not require trajectory prediction due to the small throwing distance will be discussed. The second subsection will deal with throwing and catching on a larger scale (up to several meters) where trajectory prediction is required.

For the later part of research work tracking systems and trajectory prediction approaches and algorithms will separately be discussed in two subsections. Mainly analytical methods for determining the interception position of the objects are presented and discussed. Work targeting point-symmetrical and more general objects will be reviewed and also the aspect of orientation prediction will be considered. Current approaches in different fields are introduced and discussed. Their primary attributes will be outlined, and the suitability for the transport-by-throwing approach will be examined.

Biological archetypes for catching are the topic of the third section. Research on human catching will be discussed, and the impacts of these findings on technical systems will be evaluated. This section is closed with a discussion of bio-inspired information processing approaches and their suitability for the task of catching objects.

The closing section of this chapter is dedicated to the most promising approach for transport-by-throwing based on the first three sections of related work. Open problems will be discussed and the base for the following chapter, proposed solution for a bio-inspired prediction system for transport by throwing, will be outlined.

2.1 Material Transfer in Production Facilities

The requirements on production facilities have shifted to a very great extent during the previous years. While mass production has been a primary attribute since the Industrial Age a more complex requirement, the "Mass Customization" [CA98, GW03], is challenging current production facilities. While the same, or even higher, output in terms of pieces is required, the variety of the produced objects has increased significantly. Figure 2.1 shows some examples of goods where the variety has increased tremendously between the early 70s and the late 90s [CA98, p. 6].

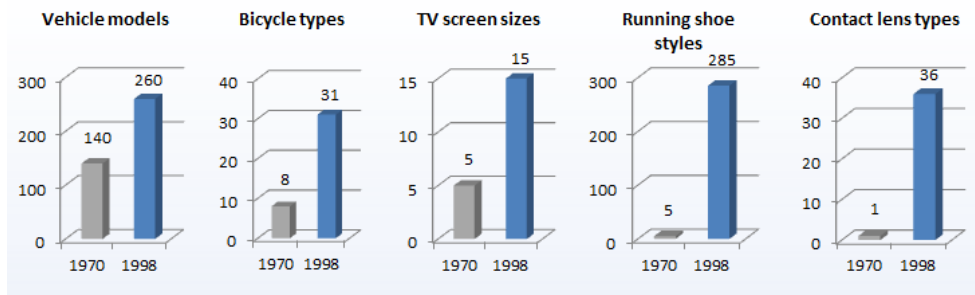


Figure 2.1: Examples of increased variety of goods between 1970 and 1998 [CA98, p. 6]

The increase of variations of goods produced raises the requirement on the production facilities. Flexibility is of main interest while cost efficiency and reliability have to be kept at the an unchanged high level. Flexible production systems also require flexible transportation systems [GW02, p. 1]. The overall increased complexity of the production process (both production and transportation system) cause an increased control- and material transfer complexity [Pil01, p. 93f].

Because of significantly increased production- and transportation costs, the production of small lot sizes still raises high costs. Automatized solutions for transportation are only suitable for larger lot sizes [GH03, p. 2]. The illustration used to point out this relation is presented in Figure 2.2. The significant increase in costs per piece with increased degree of automation in combination with flexibility (keyword "Mass Customization") [GW02, p. 1] is clearly illustrated here. State of the art production and transportation systems undoubtedly hamper the penetration of "Mass Customization".

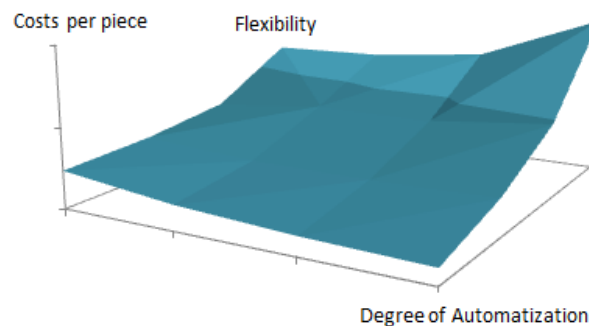


Figure 2.2: Schematic relation between degree of automation, flexibility and costs per piece [GH03, p. 2]

Comparison of different transportation systems and their suitability for flexible production systems requires explicit assessment criteria. Günthner suggests three main flexibility attributes of transportation systems [GW02, pp. 1f]. These are the flexibility of

- **Goods**

Targeting the variety of goods to be transported. Maximum flexibility here means that goods of different shape, size, mass, mechanical properties, surface properties, texture and other properties can be handled with the transportation system.

- **Layout**

Describing the connection of transfer stations. Given examples are fixed layout, flexible layout along a straight line, flexible layout along an arbitrary line and unrestricted layout [GH06, p. 2].

- **Throughput**

Classifying the possible directions of transportation. Presented examples [GH06, p. 2] suggest a line bound transportation system as a base that causes a sequential transportation along different stations. A fully meshed topology can also be realized in particular environments.

and states that a transportation system that fulfills these three requirements is convertible but that the costs for such a system are extraordinarily high, and even that such a system might not be possible to realize [GW02, p. 2]. This statement is not specified more accurately which raises some questions: what are the domains (in terms of goods), are there any? It is obvious that a transportation system flexible enough to deal with fully assembled cars and small plugs for electronic components is highly flexible, but the question is whether such a system is necessary at all. Hence, the statement leaves room for criticism.

In addition to the three main attributes of flexibility two more characteristics are suggested:

- **Extendability**

Allowing to extend the transportation system with new transfer station within the system's area.

- **Integrability**

Being a measure of the complexity of the system's integration into the managing system and bonding with other processes in the production system.

In addition, later work suggests another additional requirement: flexibility of automation's grade [GHW06, p. 3] but the attribute is not specified furthermore. The interpretation of the title is that either fully automatized stations as well as manually (by human) operated stations can be feed (loaded/unloaded) with the transportation system.

Another approach rates the modularity as the most important attribute of convertibility [Wie05, p. 26]. Modules are considered as autonomous working units with compatibility and interchangeability that are provided with material-, information- and energy-flows to perform the production task. In comparison to the requirements by Günthner [GW02, pp. 1f] here, a more revolutionary approach and definition of convertibility is used. The key to flexible manufacturing is seen in the realization of the production facility with independent modules that can be replaced or extended. This includes the attributes integrability, extendability, throughput, layout and goods but doesn't allow to rate transportation systems in different domains. Still the modularity can be seen as

one step further away from conventional transportation systems: Imagine a complete production facility realized out of a number of modular production unit connected by a self-configuring transportation systems. Such a system is comparable to the "emerging intelligence" of an ant colony.

Conventional transportations systems will now be discussed in the light of the current and especially future requirements based on the previously established knowledge and rating attributes.

Conveyors

Production facilities with conveyors for transportation task have been assigned the attributes of high operations efficiency and productivity [Miy06, p. 5]. Starting from 1995 a shift in the Japanese electronics industry from conveyor line production to cell production has been investigated. This trend has continuously been identified in studies 2004. The downsides of conveyor based systems cause costly and time-consuming reconfiguration of the production line for new models and extreme difficult layout reconfiguration. [Miy06, p. 3]

When considering the attributes established by Günthner, it is obvious that, while offering a good flexibility in terms of goods (not considering loading/unloading that can be a challenging task for a significant variation of products), the layout- and throughput-flexibility of conveyor-based transportation systems are limited due to their massive and bulky construction (compare Figure 2.3).



Figure 2.3: Typical conveyor in an industrial production environment ©CanStockPhoto

Automatic Guided Vehicle

In comparison to conveyors, automatic guided vehicles do not allocate the whole transportation line's space during the operation. This results in an added flexibility of the system as the area can be used otherwise when the vehicle is loaded/unloaded or is not passing the respective area in near future. Still such vehicles heavily rely on preset markers like wires or optical sensors [Akh11,



Figure 2.4: Example automatic guided vehicle with markers for route execution on the ground
©CanStockPhoto

p. 14] (compare Figure 2.4). This limits flexibility especially when fixed installed wires are used to guide the route of the vehicles.

In terms of the attributes by Günthner flexibility in terms of goods is limited due to the support fixture. The examples presented by Akhter by example show pallets for two of the three presented vehicles [Akh11, p. 14]. Loading and unloading these pallets results in additional manufacturing steps and thus increased time and costs. In terms of layout-flexibility, the routes of the automated guided vehicles could be altered or even be dynamically planned by a higher level control system. In the latter case also the throughput flexibility is very high.

The non-continuous flow of goods is seen as a disadvantage, as process inventory is accumulated before transporting which results in higher process inventory. Furthermore, the high utilization of space is disadvantageous. Usage of automatic guided vehicles for transportation of goods between work cells raises the assumption that they are the successor to conveyors based on the trend from conveyor line production to cell production [Miy06, p. 3] but they rather complement conveyors in applications where they are better suited.

Rotary Transfer Machines

In contrast, to conveyors mostly linear approach to transport rotary transfer machines gather a number of production process steps around a rotating table that fixes the goods and allows to move them from one processing unit to the next one. Such machines represent a combination of a small transportation system with the related processing steps and can be used to build a production cell. Loading and unloading are usually done with a robot like shown in Figure 2.5. Such work cells can be connected by automatic guided or conveyors to form larger systems.

In terms of flexibility such machines are very limited. The layout is predetermined by the round rotating table, usually the size is based on the numbers of processing steps which results in close

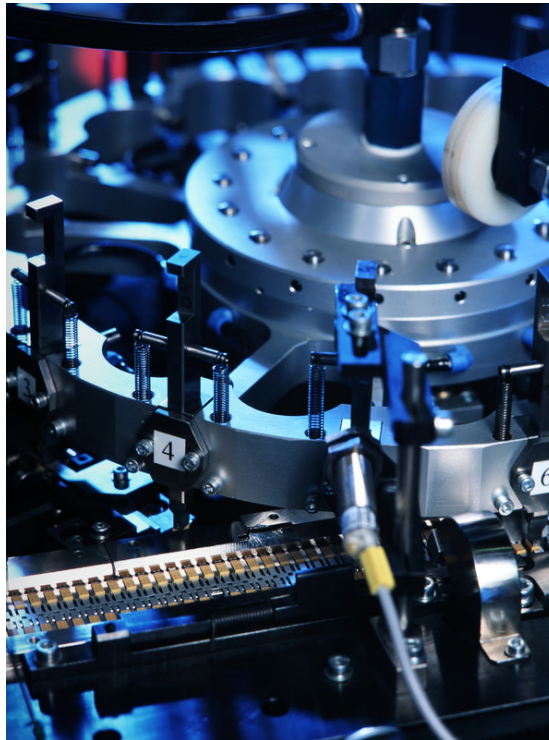


Figure 2.5: Example rotary transfer machine with an robotic arm for loading and unloading
©CanStockPhoto

to impossible extendability, throughput is clearly orientation and flexibility of goods is limited due to the fixtures for the goods on the table.

Throwing or Shooting

Simple realizations of throwing or shooting are used in production facilities. Examples are the sled for weaving, slotted casting cores in casting houses and deficient product dropout based on blow-processes. A more complex realization of throwing is used to stack sanitary products with a rate of 800 pieces per minute. A throwing distance of a lower number of centimeters is achieved by accelerating the objects with two belts. [Fra12, p. 7]

Transport by Automatized Throwing and Catching

The insufficient usability of the state of the art transportation systems for current flexibility requirements opens the door for innovation. Considering the trend towards work cell-based production systems [Miy06, p. 3] and a possible implementation of a work cell similar to Figure 2.5 with an industrial robot the approach of transport by throwing shows potential regarding the flexibility requirements. The robot used to load/unload the production cell could also be used to transport goods via throwing and catching to the next production cell. Limitations in terms of transportation distance exist based on the limited amount of energy for throwing that has to be consumed at the destination site without damaging the object. Still longer transportation distances could be realized by multi-hop throwing and catching. Here it has to be noted that a robot busy with the processing of an object can not be used for transportation of any other good.

Thus, the requirement for additional robots dedicated to transportation or redundant robots at a manufacturing cell might arise. In the latter case, the second robot could also be used for increasing the throughput at times when no other transport has to be processed. Flexibility here leads to increased complexity of planning and scheduling [Pil01, p. 93f].

In terms of flexibility the layout- and throughput-flexibility are extraordinarily high. Even three-dimensional production lines based on modular production cells could be realized where new cells with new processing abilities or for load balancing could be added any time. In terms of goods-flexibility the mass and aerodynamic properties of the object limit the application. Naturally high mass objects raise unmanageable requirements on the catching device.

2.2 Research on Robotic Catching

Robotic catching has been a research topic for more than 20 years now. The challenge of catching a thrown object are often seen as a test for the potential of current robotic hardware and control algorithms. This causes a wide range of theoretical and practical approaches to the topic. In this chapter, the related work is divided into four subsections.

The first subsection is dealing with robotic catching research where the origins of the task settings can be traced back to grasping. Due to a small free-flying distance of the object, the prediction of the object's trajectory is usually not necessary in these cases. Small distances are covered with these approaches and usually direct control algorithms, omitting prediction are used.

The second group, discussed in the second subsection, consists of research work where a larger distance is covered (several meters of free flying distance) and thus the requirement for trajectory prediction is existing. The larger distance also causes bigger deviation in the trajectories of the thrown objects and thus also catching devices, that can be moved over a wider range of spatial motion, have to be used. The bigger range of movement also requires that the movement towards the catching point is initiated as early as possible to minimize the mechanical stress on the catching device and limit the used energy. Due to the high number of approaches they are discussed in general initially and in specific subsections the aspects of the tracking systems, trajectory prediction, and the catching movements are compared and discussed.

2.2.1 Catching Emerging from Grasping

Robotic catching can be seen as a more sophisticated grasping problem where the object's state (position and orientation) changes over time in contrast to grasping a static object. Thus, the requirement of timely synchronization arises. In addition, to the transport-related application of throwing and catching also the usage of this application for demonstrations of the abilities of robots has been done.

Catching emerging from grasping has been surveyed at the University of Tokyo [NNII99]. The system presented is mainly built to show the capabilities of a high-speed vision system (presented 1992 [IMT92]) based on parallel data processing [NNII99, pg. 3200]. A 1 ms processing time is achieved by using seven DSPs (Digital Signal Processors) and the task of catching is enabled without the need for prediction based on this processing time [NNII99, pg. 3197]. An advantage of the system over more common approaches using cameras and prediction (explicitly mentioned is [HS91]) is seen in the avoidance of prediction. Considering the requirement for active vision due

to the limited resolution of the optical system (256 pixel overall) and the sophisticated hardware with dedicated tasks for each DSP the usage as a base for a general purpose tracking or catching system is limited. This algorithm is not comparable to other object tracking algorithms in machine vision because it was purposely built for this application [INI96].

Improved work with an updated sensor is presented in [NI03]. The resolution of a vision system is increased to 16384 (128 by 128) pixels [NITM00]. The focus is mainly on the processing improvements. The data, acquired with 1 *ms* cycle time, from the photodetectors is processed in column-parallel configuration instead of the completely parallel architecture for the previous 128 (16 by 16) pixel sensor. Further improvements are the usage of a stereo configuration with 3D self-windowing [NII02]. The main statement of the contribution is that the increasing speed of mechanical actuators (e. g. robotic arms) in combination with high-speed sensors allows to avoid prediction for tasks like catching. The disadvantages of a slower visual perception (like humans have) are the requirement of high-level prediction that requires hard learning with a long learning rate. Direct mapping from the sensor data to motor commands for the robotic arm is used. It is interesting that the rate of successful catches is not given in the publication.

This argumentation apparently leaves other aspects open. Even if the speed of the actuators is further increased the amount of power consumed for a movement is still depending and increased by the mass, that has to be moved, and also increases as the time of execution is decreasing. Under the aspect of energy efficiency, the statement clearly can be opposed by this fact.

The used gripper is introduced in another article [NIIK03]. The three-fingered hand is presented and the experimental results for catching a falling foam ball with a radius of 4 *cm* at about $4 \frac{m}{s}$ are given. The sensor for this task is the same sensor system as in [NI03] that is presented in [NITM00] in a stereo configuration with the same processing algorithms as in [NII02]. Consequently, prediction is not needed. The success rate of the gripper grasping the ball with a deviation of ± 2 *cm* from the center of the palm is more than 90 %.

This contribution shows potential for small scale catching. Still for the small deviation of only ± 2 *cm* the catching rate should even be higher than 90 %. Especially with a catching friendly foam ball. Further optimizations are given, and one aspect is the increase of actuator speed that is a brute force approach. The combination of the small scale catching system with a robotic arm, as presented in [NI03], looks very promising, but the lack of a stated success rate raises the thought that the real-time mapping of the sensor data to the movement of the robotic arm is not accurate enough. Reasons for this are seen in the delay of the robotic controller and the mechanical actuators to execute the movement commands.

The next step in the line of research is the catching of a more complex object, in this case, a cylinder with 5 *cm* diameter and 10 *cm* length [INHI04]. The vision system is similar to the one used in [NIIK03] and [NI03]. Again the lack of a stated success rate raises the thought that a successful catch was a rather rare event. For the overall system again the limitations of the catching area and the positive impact of foam (soft) object for catching are valid.

In 2005, the dribbling of a ball on a flat surface was presented [SNI05]. This raises the demands for the dynamic stability of the manipulation. A fundamental change was done in the area of the vision sensor as an integrated vision chip with 4096(64 by 64) pixels was used [KIY03]. The processing times for the momentum of the binary image could be reduced to 17.5 μs while the determination of the area could be achieved within 1.3 μs . The whole vision system was built out of two of these vision chips in a stereo configuration. The determination of the center of the spherical object was done based on the calculation of the moments of each image. The

actuator/gripper, the multi-fingered hand presented in [NIK03], was used to dribble a ball with a diameter of 30 mm with an amplitude of 0.15 m. This factors caused a period of 100 ms for the dribbling motion. Here prediction was used for the movement in order to reach a target position where stable dribbling is possible.

The process was stable in the simulation but not in the experiment. The limited resolution of the vision system with an assumed 3D position error of 10 mm was considered to be responsible for that. Also, the calculation of the center of the ball based on the (simple) determination of the momentum might also be a major source for the error. More advanced algorithms like the Hough Transform [P.V62] might reduce this error significantly.

In 2003 also research targeting the control architecture of a combined robotic manipulator (the multi-fingered hand and a robotic arm) was presented by the Japanese research team [NHI03]. The architecture was derived from the human archetype. The highly sophisticated human ability for dynamic tasks with visual feedback like tennis, basketball, baseball or even day-to-day activities like hitting a cockroach with a folded newspaper based on visual feedback [NHI03, pg. 873] inspired the research to develop a hierarchical control architecture. In comparison to earlier work based on simple visual feedback more sophisticated feedback was necessary for multi-objective, flexible and high-speed visual servoing [NHI03, pg. 873]. The hierarchical structures with efferent and afferent¹ signals that interact on each level in biological systems was discussed and a model for technical systems established [NHI03, pg. 874]. A main attribute here is that the sensory information (efferent signals) is fed to all the layers of control architecture. This is reasoned by the requirement of real-time task selection in each layer for complex tasks [NHI03, pg. 875]. Additionally also the relation between the gaze behavior and the movement of the manipulated object is discussed based on [JWBF01]. Conducting the manipulator to the object is seen as the main function here. This is opposed to other research that is discussed in Section 2.3. Also the attributes of the visual system are discussed and an engineering implementation using saccades is established [NHI03, pg. 876]. Interestingly the usage of position-based control, defined as control based on extracted positions, compared to image-based control, defined as control based on target images, is used in the model. While the argumentation of the authors favors image-based control due to the higher stability, and even argues that this might be the reason for the high stability of the human manipulation skills, this is clearly opposed by other research in Section 2.3. Another point of criticism is that the structure's property that all sensory information is fed to all layers without further processing is opposed by newer findings of the human process of perception (compare Section 2.3). The proposed control architecture is tested in an experiment with a 2 degrees of freedom (DoF) active vision consisting of the previously discussed SPE-256 (16x16) visual sensors, a 7 DoF arm and a 14 DoF four-fingered hand. A number of assumptions are used to simplify the overall task like the assumption that the object is only moving within one plane allowing the usage of a single camera and the experiment uses low dynamics with a ball mounted on a stick. Highly dynamic tasks (like catching a thrown ball), that are the main goal of the proposed architecture are not tested.

One year later, in 2004, research work dealing with the topic of batting a ball with a robotic arm is published [SNI04]. Batting here is a dynamic contact with the object and is in many aspects similar to catching. The main difference is that for catching the impact should be minimized while the impact is necessary for batting and should lead to an intended movement of the object. The article states that batting does not require the application of a human processing architecture, the human mechanisms. The way humans process information is seen as clear results of the "low throughput" visual system [SNI04, pg. 1192]. Thus, also the use of an efference copy to compensate

¹Efferent is defined here as sensory input while afferent is defined as (motion) command

the delay in visual processing of humans is not required by a technical implementation that has a high speed and low delay vision system. It is noted again that more recent research here clearly shows a different point of view (compare Section 2.3). In regards of the vision system the commonly used system with 1 *ms* cycle time and 16384 (128 *by* 128) *pixels* is used. Successful hits of the ball are achieved in with 90 % rate. The batting point is initially calculated based on a simple prediction that is updated during the movement of the arm based on the high-speed vision setup.

In the following year an update to the gripper was introduced [NI05]. The multi-fingered hand was enhanced to a three-finger hand with a design for maximum output for a short period of time [NI05, pg. 2655]. With the common stereo vision system [NITM00] a rate of 90 % success catches could be achieved. Similar to earlier work this number is based on a balls deviation equal to ± 2 *cm* from the center of the ball. The main focus of the paper is the analysis of catching. The dynamics of the catching movement are discussed, and a major result is that the impedance of the object and the control have to be matched. More precisely the mechanical impedance and the feedback control have to be balanced for stable catching [NI05, pg. 2660].

The updated actuator was subsequently used for dynamic re-grasping [FNTI06]. Different objects were manipulated without the need for multiple actuators. A single three-fingered hand was used for the quick operation. Compared to putting the object down and re-grasping it from there, this approach for object manipulation is a significant speed-up. The actuator's joint speed reached $1800 \frac{^\circ}{s}$ and the common vision system in a stereo configuration was used for the research work. The vision system was used to extract the position and the orientation of the object and more complex objects than the point symmetrical sphere were used. The most complex task achieved was to re-grasp a cylinder from power position (fingers around the object) to precision grip (cylinder axis normal to palm). For achieving this, a parabolic orbit for the center of mass was used, considering the mass of the object and gravity. In combination with the hang time and the initial velocity the orientation when falling was calculated [FNTI06, pg. 184]. The success rate for this experiment was at 35 %. This relatively low rate was explained by timing problems. The velocity of $2 \frac{m}{s}$ resulted in a 2 *ms* timing tolerance for the whole system in order to stay within the limits for successful catching [FNTI06, pg. 186]. Other reasons for unsuccessful re-grasping trials were seen in the inaccuracy of the target's/object's physical model and the physical model for throwing and catching. The tolerances of the system were ± 2 *cm* position error and ± 0.3 *rad* orientation error. All three reasons, the researchers identified to be responsible for the low success rate, are based on the fundamental principles of the approach. While arguing that only low throughput vision systems require prediction to achieve high dynamic tasks the results of the experiments clearly show the opposite. The required timely accuracy is simply too high. Using information from a number of recently acquired samples of position and orientation a more accurate determination of the current (or even future) object's state could be done. The other two aspects regarding the inaccuracies of the models can be targeted with simple learning algorithms where, at least, some parameters are updated based on the acquired data.

More recent research has focused on knotting and object handling in hand. Common attributes of these topics are the small range the object moves before it is caught and the high-speed position detection. These two restrictions allow catching based on the actual measured position of the object, omitting the need of an interception time, position, and orientation prediction system.

2.2.2 Larger-Scale Catching

The big variety of approaches to large-scale catching is shown here, and the theoretical and practical research works are discussed here. The first subsection aims to give an overview over the approaches dealing with ball catching (passive and active catching devices) as well as with catching of other objects and is closed by a paragraph about previous work by the research team at the FH Heilbronn and the Institute for Computer Technology at the Vienna University for Technology.

In the two further subsections are devoted to the specific aspects of the tracking systems and the prediction systems and algorithms. Common and differentiating aspects will be the main topic here. This structure allows to give a good overview over the wide range of approaches.

2.2.2.1 Discussion/Overview of/over Large-Scale Catching Approaches

Large scale catching approaches are divided into three main categories here. The first category is dealing with catching balls with passive catching devices. Passive here means that no part of the catching device is moving in order to obtain a without form- or force-closure grasp. Simple catching devices like baseball gloves, boxes with holes or cups are used here. The second category deals with catching darts with passive catching devices.

The third category is devoted to any objects with active grippers where the gripper is actuated during the catching phase. These approaches have a higher degree of complexity as the additional control requirements on the gripper are existing. In the final subsection, the previous own contributions to large-scale catching from the research team are discussed.

Catching Balls with Passive Catching Devices

One of the first approaches to robotic catching was done by Hove in 1991 [HS91]. The system designed makes use of a 30 Hz stereo vision system for the tracking of the thrown ball. This sensor system is the only input to the prediction and robot control as no other knowledge (like starting position or velocity are used) [HS91, pg. 380]. The reason for this is not stated in the report. The prediction of the trajectory is done within 0.5 s, that are a part of the flight time of the thrown object [HS91, pg. 380]. The robot motion starts as early as possible, and the goal for the motion is to minimize impact bouncing and slippage. In addition, the idea to mimic a good human catcher is stated. Details about the robot control are not given, but it is assumed that the robot is controlled in Cartesian space. This is based on Figure 4 of the publication [HS91, pg. 383] where a control loop for the whole system is given. This method avoids to use the joints of the robot in an efficient way and is an obstacle of the objective to mimic a good human catcher (compare 2.3.3). The avoidance of backtracking² is done to achieve a more graceful movement. Instead of an active gripper or similar catching tool, the robot end effector is moved along the trajectory of the ball with the same velocity [HS91, pg. 380]. Following on this velocity synchronization period a deceleration along the ball's former trajectory is executed. Workspace limitations of the robot are handled by using the best possible movement instead. The prediction is based on trajectory matching using a parabola to predict the ball's motion and a second order filter $\ddot{x} = 2 * F * (\dot{x}_{des} - \dot{x}) + F^2 * (x_{des} - x)$ [HS91, pg. 381]. The parameter F is used to change the

²here used to describe the circumstance that the end effector of the robotic arm initially moves towards the object and changes its direction afterward to follow the object

response of the whole system when the ball is closer to the robot. For a distance of less than 0.2 m the a smaller value is used to reduce jitter, for higher distances F has a higher value. The exact values are not stated. For the whole system, a success rate of $70-80\%$ is stated. A successful case is here that the convergence of the end effector and the ball to $0-5\text{ cm}$ is achieved. The reason for the unsuccessful trials is seen in the inaccuracies of the visual system causing the trajectory data to contain bigger measurement errors. Interestingly later work by Riley [RA02, pg. 122] considers the usage of trajectory matching as not suitable for a robot with a more limited workspace.

In contrast to the previously discussed work, where no actual actuator was used, a research team published findings on nonprehensile manipulation in 1998 [LM98]. Nonprehensile here means that no gripper is used for the manipulation of the object. Only a very simple actuator (without form- or force-closure grasp) with 2 DoF is used [LM98, pg. 27]. Based on the usage of gravitational, centrifugal and Coriolis forces the object's pose is manipulated. A typical example of human's object manipulation based on these forces is given with the throw of a basketball. Potential benefits include the extension to new robot primitives (option for too large and too heavy objects), a simple manipulator, flexibility and general usability of the "gripper", increased workspace size based on object throwing and higher DoF of the object's manipulation compared to the DoF of the manipulator (underactuated manipulation) [LM98, pg. 2]. These possible advantages come at the cost of an increased complexity of planning and control and the dependence on the (accurate) dynamic model and the requirement for an object with at least one partially planar surface. The complexity is transferred from the gripper, joints and actuator to planning and control but the fact that also higher complexity systems require higher complexity control algorithms has to be considered. This reduces the penalty of the increased requirements in these regards. In the practical experiments, a simple robot is used to control an object to a full-dimensional subset of the six-dimensional state space by slipping and rolling. The robot is used in related work for feeding tasks [LM98, pg. 3]. The movement of the robot for four types of motion are simulated before implementation using a fourth-order Runge-Kutta method to find the optimized solution. For snatching an object (cuboid), special friction and velocity constraints are existing and the optimized solution for the movement is found after 200 iterations which took 120 seconds. The movement has 9 knot points and is experimentally robust. For throwing the same object with reorientation (comparable to dynamic regrasping [FNTI06] but without form- or force-closure grasp) 98 iterations in 13 seconds needed for the optimization. 7 knot points and a special "double pump" behavior to increase the friction are attributes of the resulting movement. For the included catching the problem that the object tilts out of the plane arose for the experiments that were robust. The roll motion (similar to throwing with rotation but without the loss of contact) used only 1 DoF of the manipulator and was optimized with 55 iterations in 32 seconds. Without a limit on impact velocity, this motion was stable but after limiting the impact velocity to zero, the solution is not experimentally robust. The last motion is a roll-throw with the 1 DoF manipulator. The attributes of the movement, found after 54 iterations in 35 seconds, are 9 knot points and that it is fairly repeatable but has a lower accuracy than the normal throw. Overall the work shows the increased possibilities of dynamic manipulation (like rolling or throwing) but also the disadvantages of the limitation to partially planar objects are obvious. In addition, the dynamic modeling of such simple (cuboid) objects, and the challenges already examined there, show how hard it is to establish an accurate model.

In 2000, a humanoid robot was used for catching experiments [RA00]. The robot had 30 DoF and was used for dancing and catching experiments. A child's baseball glove was used for catching. A color stereo system was used to detect the position of the ball in space. The movement in x and y dimension is considered as linear while the movement in z dimension (height) is described

using a parabola. This description is transformed to a state representation, and a first prediction is done after five measurements of the camera system. The motion of the robot is planned using a programmable pattern generator [SS98]. This results in a bell shaped point-to-point movement to the interception point. The interception point is specified by a certain z dimension (height). The results of the experiments were that problems with the depth measurements arose and that experiments on adding a Kalman Filter were conducted [RA00, pg. 40]. This initial work was continued, and a more recent article was presented 2002 [RA02].

The updated work targeted human-humanoid interaction in the form of playing catch [RA02]. The aim was to do ball prediction and robot trajectory generation with human-like characteristics. The suggested bang-bang or bang-coast-bang trajectories from previous work was found to result in a movement that was not smooth enough and excited too much vibration on the humanoid robot. The vision system used as sensors was the same as in the earlier work [RA00] and supplied the centroids of pixels of a particular color at a frame rate of 60 *fps*. Also, the baseball glove used in the prior work was kept for catching. The vision system was located beside the human and humanoid so that the baseline of the stereo system was parallel to the plane of the thrown [RA02, pg. 122]. Regarding the prediction of the trajectory and the determination of the interception point the approach from the work presented in 2000 was kept. The parabola for the movement (considering gravity but ignoring drag) was formulated with $z_i = a * t_i^2 + b * t_i + c$ while the movement in x and y was assumed to be linear. The trajectory prediction started with five frames of the vision system and was updated with each new frame. The preferred catch height was specified and used to derive the interception position. Due to the limitations in the robot's speed-up and slow down in the moving range the usage of velocity matching during catching was avoided. A follow through movement is used to keep the ball in the baseball glove. The researchers use human inspired jerk free movements that result in bell-shaped velocity profiles. These velocity profiles reduce the vibrations and are desirable for the humanoid. The overall motion, determined with the programmable pattern generator [SS98] is based on the average velocity and the target position. Based on an earlier work [TS00] the inverse kinematic is used, and trajectory planning can be updated after every new measurement from the vision system. The required peak hand velocities for paths $< 1\text{ m}$ are $< 2.5\frac{\text{m}}{\text{s}}$ for a ball caught in a height of $0.32 - 0.4\text{ m}$. The catching movement takes between 350 ms and 850 ms . A specific number for the success rate is not stated, but a number of reasons for missed catches are given. Overall it is stated that the reasons are difficult to quantify. Two reasons are based on the working envelope of the robot. These are balls thrown outside the workspace of the robot and balls thrown too close to the torso. In some cases, the robot bumps the ball with the glove. This is caused because the direction of reaching into the interception positions is not specified and thus the glove hit the ball on the way to the interception position. The strategy to move the hand from the opposed direction of the ball's trajectory to the interception position is given as a promising solution to this problem even if the impact energy is increased due to the added difference in velocity. Another reason for unsuccessful catches is that the ball bounces out of the glove. Here the adding of a grasping mechanism or lessening the relative hand-ball velocity to reduce the abruptness of impact is considered as a solution strategy that is conflicting with the prior mentioned strategy to avoid bumps. In some cases, the trajectories are not executed well. Regarding the vision setup, the noisy position information is also considered as a reason for missed catches but still considered better than the implementation of a Kalman filter, that requires models for the sensor and process errors and noise. Furthermore, the trajectories show overshoot behavior that is a trade-off between speed and accuracy thus balancing the reachable balls and overshoot. The main improvements for the system are given in the reduction of the relative velocity and the usage of a template from human motion capture in

the form of a movement library, possibly with an interpolation scheme. The presented work has interesting aspects regarding (jerk free) motion planning, mimicry human motion behavior and the sources for the missed catches in combination with the suggested strategies for solving these. The statement versus the usage of a Kalman filter is nearly unique in the field of robotic catching.

A catching system using only commercial off the shelf components (from scratch) was presented in 2007 by Smith [SC07]. The system was designed for a throwing distance of 5 m with a flight time of 0.8 s. The velocity should be in the range of $6 \frac{m}{s}$ when arriving in the area of the robot. The catching area was specified at 0.6 by 0.6 m. A standard vision system with 50 fps was used and based on the findings of Frese [FBH⁺01, pg. 1628] the reduced accuracy of the cameras might lead to the necessity that up to 10 measurements have to be used for an accurate prediction [SC07, pg. 4057]. Based on the consequential assumption that 200 ms are used for prediction, 100 ms can be used for an optional human in the loop a maximum movement time of 500 ms was used for the determination of the mechanical components. In addition, the position accuracy of 1 cm and the movement of 0.9 m within 0.5 s are requirements. Furthermore, the requirement to have a closed solution for the kinematics to fulfill real-time requirements is raised. On the actuator side, the constructed robot uses three 1.5 kW motors and weights 10 kg. The link configuration is found based on an exhaustive search with a simulation of the arm [SC07, pg. 4058]. A computer with the real-time extension RTAI (Real-time Application Interface for Linux) is responsible for dynamic and kinematic calculations and trajectory generation. The performance of this computer is considered as good in the report. The control loop is implemented in soft-real-time with a latency of less than 1 ms. The result of the design is a robot that has a repeatability that is better than ± 1 mm. For the robotic catching experiments, the robot is equipped with a 14 cm diameter cardboard basket with damping material. The used throwing device is a mechanical launcher with a precision of ± 10 cm at 5 m distance. On the sensor side, a stereo set with a baseline of 60 cm is used in combination with a color segmentation based on the HSV (hue saturation value) color space and an extended Kalman filter to track the trajectory of the ball. The determination of the ball's center is based on an algorithm that calculates the center of the mass of pixels that have at least three neighbors with a sufficient similarity in the HSV color space. In order to reduce the computational expenses, only a subwindow is processed after the initial search for the ball. The size of this subwindow is several times the standard deviation in position. The result is a 4 ms long processing time per stereo-pair. The catching position is determined by intersecting the robot's workspace/plane with the prediction of the thrown ball's positions. In the conducted experiment 25 out of 32 balls are caught which equals a success rate of ≈ 78 %. The overall cost's for the robot are given at 50,000 EUR. The fact that the robot's price is at this value and the very high power demands (3 1.5 kW motors) rather limit the range of application of the robot. Robots developed by bigger manufacturers achieve similar or better performance in combination with lower energy consumption and similar or lower costs.

Catching Darts with Passive Catching Devices

In contrast to all the previous work, where balls were thrown and caught, Linderöth presents research work dealing with the "catching" of a dart [LRÅJ09]. To "catch" the dart, a dartboard is mounted to the robot. The task is to move the dartboard in order to hit a bull's eye with a dart thrown towards the robot. For tracking of the dart, two color cameras with a resolution of 656 by 490 pixel are used. The placement of the cameras is on the catching side with the argumentation that higher accuracy is achieved when the dart is closer to the dart board [LRÅJ09, pg. 884]. The detection of the darts in the images is based on the color in HSV color space (similar

to [SC07]) and the criteria for classification is found empirically. The center detection was based on the number of pixels with the adequate color and to determine this efficiently an integral image was used. The rather simple rule for the center determination is based on two successive main criteria: 7 out of 9 (3 by 3) pixels have to be of the right color (this filters outliers) and furthermore the highest number of pixels with the right color in a box of 5 by 5 pixels is considered as the center of the dart. If more than one square has the same number of neighbors, the center of gravity of these pixels is used. Interestingly the window size is not adapted to the distance of the dart to the vision system but fixed. The data of the detection system is fed to a Kalman filter where the prediction is done. The dart is therefore modeled as a particle without orientation. The state vector contains information about the positions and velocities. Regarding influencing forces only gravity is considered, and air drag is neglected. Multiple iterations of the Kalman filter are used to predict the interception position. The criteria for the dart is to reach the plane, in that the dart board is moved. Two approaches for dealing with the input data from the vision setup (two cameras) are presented. The first, widely used approach, is to do stereo triangulation and use this 3D data for the Kalman filter. The second approach is to use individual constraints for the Kalman filter. This allows that each dataset is used independently and that more cameras can be added. In terms of computation time, the achieved results are 10 *ms* time to process the images and do the prediction. That means that the prediction is readily available for each frame of the 50 *fps* vision system. Overall all frames are processed until the motion command is executed, which takes between 40 – 50 *ms* [LRÅJ09, pg. 887]. The author states that problems with objects of the same color (e. g. another dart with the same color) lead to detection and thus prediction errors. The accuracy of the whole system is given at 1 – 2 *cm*. For wobbling darts, the accuracy is decreased to ≈ 5 *cm*. This is a system design problem as the object for tracking is the tail of the dart but this point clearly differs from the center of mass, and an even bigger spatial distance is existing to the point of interest, namely the tip of the dart. As the orientation of the dart is not considered at all (modeling as a particle) this problem prevails.

Successive work presented in 2010 [LRÅJ10] changes the application to catching a ball with a robot. Here the problem with the orientation and the mismatching of the point of interest and the point that is tracked is avoided. A box with a round hole is used for catching instead of the previously used dart board. The scientific background is sensor data fusion for visual sensors in combination with Kalman filters. The focus is lying on a concept that is robust to errors in image acquisition and processing [LRÅJ10, pg. 4225]. The usage of view lines, described as intersections of two planes, as constraints for the Kalman filter allows an acquisition error tolerant processing of the sensor data. The prediction is based on a Kalman filter where gravity is modeled, but air drag is neglected. The visual sensor system is based on two cameras with 50 *fps* and a resolution of 656 by 480 *pixels*. In contrast to the previous work, stereo triangulation is completely avoided, but an approach with the two unsynchronized cameras (alternate frames) is presented. This approach shows good convergence after only 4 *frames* (2 for each camera). Even using only one camera is possible and examined but problems with rather big errors exist [LRÅJ10, pg. 4229]. The approach easily handles any number of cameras, and the accuracy is significantly improved for two or more of them. The performance of the ball catcher with a 5 *cm* diameter ball and a 6 *cm* diameter hole in the box is ≈ 50 %. The problem behind this rather low success rate is seen in the low velocity/acceleration of the robot ($0.5 \frac{m}{s}$ / $13 \frac{m}{s^2}$). From a distant position also the usage of a Kalman filter, that is designed for measuring/predicting linear processes and the used model, neglecting drag by example, are also possible reasons for the relatively low success rate.

In 2010, Bätz presented a new approach to the nonprehensile manipulation of a basketball [BYW⁺10]. This work uses a standard industrial robot compared to the simple 1 DoF to 2

DoF manipulator used in [LM98]. Similar to the earlier work the advantages are seen in the omission of a gripper and the usage of a generic end effector that allows handling [BYW+10, pg. 365]. The industrial robot is a Stäubli RX90B with a circular plane with a 17 *cm* diameter plate used as an end effector. For the whole work the assumptions are that the air resistance and rotational velocity of the basketball are negligible and that the impact is an inelastic collision. The control algorithms are implemented on a standard PC with Matlab/Simulink and RTAI (Real-time Application Interface for Linux). The sensor system is built out of two high-speed cameras with frame grabbers in a stereo configuration with 2 *m* baseline and a resolution of 1280 *by* 1024 *pixels*. Using a window search with a window size of 180 *by* 180 *pixels* with linear position prediction a frame rate of 150 *fps* is achieved. The tracking of the basketball is based on the color in HSV color space and two opening operations (erosion and dilation) are used. The first order moment of the classified pixels is used for the center ball. The prediction model considers only gravity and (similarly to most other approaches) neglects air drag. A recursive least squares estimation for the parameters of the trajectory is done. Two types of catches are examined: direct catches where the ball is kept in contact³ from the very first touch of the ball on the circular plane and indirect catches where the ball is rebounding on purpose to be caught afterward. Furthermore, the balancing of the ball on the plane after catching is done. Regarding the prediction error, the prediction with a time horizon of 0.5 *s* is stated with less than 0.05 *m*. The success rate for direct catches is 35 %. Limitation factors are seen in the inaccurate ball trajectory prediction and trajectory tracking errors when the path is too close to the hardware limits of the manipulator. For the indirect catches the two assumptions made are considered as inadequate as the visual feedback shows different behavior during rebound than the model based on the prediction trajectory. An idea for optimization is to use a spring between the robot and the end effector in order to relax the dynamic requirements for the robot and possibly increase the success rate. The results clearly show that the assumptions established for the work are problematic. The air resistance for the rather heavy basketball still has a noteworthy influence on the trajectory that can be derived from the indirect catching experiments. The challenges of modeling a process accurately once more are shown here.

Catching with Active Catching Devices

In 1995 Hong and Slotine were able to show the realization of the system introduced in 1991 based on a set of more advanced cameras [HES95] compared to [HS91] (first paragraph in Section 2.2.2.1). The vision system, identified as one main reason for unsuccessful trials, is updated with a system capable of 60 *fps* instead of the 30 *fps* of the previous system. The prediction is still based on a parabolic function that is fitted with least squares to the measured data of the vision system. The closest point of the trajectory to the robot's stand is used as catching position allowing acceleration of the robot within its workspace to reach the interception position. During the long time of 330 *ms* to close the gripper no updates on the robot's movement are done. The achieved success rate is again 70 – 80 %. The requirement here for a successful trial is that the hand is catching the ball. The hand's tolerance is said to be 12.5 *mm*. Another statement is made about the tolerance of timing errors of the prediction. A value of 5 *ms* is stated here. This is an interesting statement as this time combined with a throwing distance of 1.5 – 2.5 *m* and a flight time of ≈ 0.7 *s* results in a spatial distance of 10 – 17 *mm*. The partition into temporal and spatial errors here is questionable as there is a clear relationship between time and position given by the (vectorial) velocity. The majority of the failures is attributed to the noisy data from the

³or very close to contact

vision system [HES95, pg. 8]. In addition, the latency of the vision system is hard to compensate as the output is only available after computing the ball's position based on the image data. In addition to the balls also foam balls and paper airplanes are thrown, but only general information is given about these experiments.

Two years later the usage of a humanoid robot was used to show human dexterity, especially the catching of a thrown object [NIN⁺97, pg. 94]. In contrast, to the assumed human catching behavior based on the balls location, mapping of arm movement and catching based on visual feedback the last aspect is not considered in the research work. Only the prediction of the flight trajectory and the movement of the catching device (a cooking basket with 120 mm diameter) is done. The visual sensor acquiring the information for the trajectory prediction is a unsynchronized stereo set. For these set two strategies are introduced and used: for a falling ball the distance of the cameras to the ball is assumed to be constant and for a thrown ball the distance of the ball to the camera is derived from the size of the ball in the image. The measurement error for this procedure is given at 13 mm on a range of 300 – 1200 mm distance. A neuronal network is used to learn the inverse kinematic of the 5 DoF arm/hand. No reason for the usage is stated, but it is assumed that this was an easy solution with a suitable performance. The higher number of DoF for the arm/hand than the 3 DoF for the ball cause the problem to be ill-posed. By using two predefined conditions, this challenge is dealt with. The average error of the neural network based inverse kinematic calculation is given at 5 mm for theoretical data while only 52 mm are achieved by using the real robot with the visual sensor. The backlash of the gears and the offsets in the origins of the joint's angles is considered to cause this rather big deviation. Compared to many other approaches the challenge of the robotic mastering (determining the origins of the joint's angles) is considered and targeted here. The catching strategies depend on the task: for a falling ball, the hand is moved under the measured ball position and the 7 cm ball is caught; for the thrown ball a parabola in the sagittal plane is fit with least squares method. The first fit is done based on 5 data points, and each new data point is used for a new determination of the parabola's parameters. In addition, weights for the impact of each of the five measured points are used to raise the influence of the later acquired positions of the ball. The equations used to describe the movement are $x = a_0 + a_1 * z$ for the horizontal plane and $y = b_0 + b_1 * z + b_2 * z^2$ for the sagittal plane. Interestingly, while a parabola is stated in the text for the sagittal plane both equations have the variable z while the time is not used. Here an error due to the decrease of the velocity over time due to the air drag is introduced. Using additional visual feedback for the hand position during the final catching phase is a future topic. No final success rate is given, but the hint on the visual feedback suggests that the achieved success rate is rather low.

The national aeronautics and space research center of the Federal Republic of Germany has a wide selection of articles targeting robotic catching. The initial work is done in 2001 with a light weight robot version 2 (LWR II) and is devoted to the development and evaluation of an off-the-shelf vision system for a robotic ball catcher [FBH⁺01]. The configuration of the experiment is a throwing distance of 5 m and a flight time of 0.8 – 1 s of the foam ball used. The vision system used is a set of two cameras in a stereo configuration. The reason for this design decision is given in the fact that the prediction of a ball with one camera is an extremely ill-conditioned problem [FBH⁺01, pg. 1623]. The cameras are synchronized, have a baseline of 1 m, are aligned vertically and mounted on the throwing side in order to get more accurate measurements in the initial phase [FBH⁺01, pg. 2]. The relation between the robot and the camera is determined based on ≈ 10 measurements of the robot's position with the cameras in the field of view of the cameras. The data is evaluated manually. The cameras output image data in phase alternating line (PAL) format at a frame rate of 25 fps or 50 $\frac{\text{fields}}{\text{s}}$. Each of the fields is considered as a full

frame thus an error of 0.5 pixels is systematically introduced. The segmentation of the image and detection of the ball is based on background subtraction with a reference image, thresholding and shape detection based on fitting of the shape of the tracked object (here a 2:1 ellipse). A region of interest (RoI) of $2 - 40 \%$ of the image is used for the difference image in order to reduce the computational expenses. The precision achieved based on this tracking system is 3 cm [FBH⁺01, pg. 1624]. The position data from the vision system is fed to an extended Kalman filter (EKF) for state tracking in 3D. The motion model considers gravity and air drag. A Reynolds number (compare Section 4.2.1) of 20,000 is stated for a tennis ball or foam ball which results in a drag coefficient of $c_d = 0.45$ [FBH⁺01, pg. 1626]. A new prediction is done for each new field from the cameras. The overall system latency including exposure, image transfer, frame-grabber driver, vision algorithm, prediction and robot control is lower than 75 ms . For the determination of the catching position limitations in the workspace of the robot are considered, and an approach based on linearization is used where the position and velocity for catching are determined. [FBH⁺01, pg.1627]. The system achieves a success rate of $2/3$ with a net used for catching. The majority of the faults is ascribed to the camera limitations [FBH⁺01, pg. 1628]. A high accuracy of the prediction ($< 10 \text{ cm}$) in the early flight phase is a special attribute of the system. Problems with the illumination/lighting system are also stated to exist [FBH⁺01, pg. 1625].

Nine years later a new unified approach for catch point selection, catch configuration computation and path generation for better performance was presented by the same research institution [BWH10, pg. 2593]. Also, the hardware platform was updated to a 13 DoF system with the successor of the robotic arm (LWR III) and a newly developed gripper, called DLR Hand II. The vision setup was taken over from earlier research (similar setup to [FBH⁺01]). Tracking of the ball with an extended Kalman filter (EKF) was successful for $> 80 \%$ of the fields (line interlaced frames of the camera). Path planning for the robotic arm was done in an optimizer after the initial planning solution was computed on a 32 CPU core cluster. The calculation of the initial solution was stopped after 60 ms and yielded approximately 100 optimized solutions that were considered to contain the globally optimal movement. For each new frame, the planning algorithm was updated. This computation was done on a 2.2 GHz CPU and took 60 ms to compute. Three cores were used in a round-robin scheme in order to compute the new prediction immediately after the acquisition of the new image from the cameras. The precision of the trajectory prediction reached $< 2 \text{ cm}$ only 100 ms before the catch. Another CPU core was used for the control law of the hand that required fast closing and vibration suppression. The overall latency from the shutter of the cameras to the execution of the robot movement is stated at 90 ms . The robot communication is based on the fast research interface (FRI, compare [KW10]). The combination of high joint acceleration and comparable low maximum joint velocity of the robot results in a movement that is 80% at the maximum velocity of the joints and only 0.1 s is taken for acceleration. Three different catching strategies are presented that change the amount the robot moves and the position the catching is done. All of the three strategies catch the ball without synchronization of the robot's velocity to the ball's velocity. The success rate of the experiments is at 80% for two of the three strategies and lower for the third one. Errors are attributed to lost tracks of the ball, failures in the vision system and prediction errors (too low prediction accuracy) while the robot path planning and motion execution is considered as highly reliable. Also, the hand's orientation is important for successful catching [BWH10, pg. 2594]. The impressive system used very high computational power (1 PC for image processing, 1 32 core cluster for path planning, 1 PC for robot trajectory generation) and is rather a brute force and effective than an efficient solution. Still it achieves the highest catching rate to date.

The approach was applied to another robot platform, the "Rollin' Justin", that is a 53 DoF

mobile robotic system (2x7 DoF Arms + 2x12 DoF Hands and 5 DoF torso+head). The same success rate was achieved as for the individual robotic arm (80 %). Another application for the robot is preparing coffee, showing the versatile usability of the platform [BSW⁺11]. More details on this approach are presented in [BBW⁺11]. The challenges for catching on this platform are the high complexity (53 DoF) of the robot where the movements of the onboard sensors have to be tracked and the synchronization without a global clock [BBW⁺11, pg. 514]. The robot is therefore adapted for this task with minimal impact on other applications. All sensors used for catching are onboard of the robotic platform, but the computation of the data is mainly done on an off-board cluster that is WLAN coupled. The main sensor for the ball prediction and catching is a stereo-system with a small baseline (inside the robot's head) running at 25 *fps*. The balls used for the experiments are 8.5 *cm* diameter balls with 50 *g* that have a > 20 *cm* shorter flight trajectory than a purely ballistic trajectory. The challenges in detail are:

- low latency (< 1 *s* flight time, instantaneous reaction)
- precision in time and space (2 *cm* position, 5 *ms* time accuracy)
- moving camera system (inside the head)
- vibrations in the system [BBW⁺11, pg. 515]
- observation of kinematic state
- limited on-board resources and communication bandwidth
- no global clock and communication latency

The arms of the robot are LWR II arms and the gripper used is the DLR Hand II (similar to the more simple approach presented in [BWH10]). Controlling of the torso and arms is done in joint space. The visual system is composed of two cameras with 1616 *by* 1220 *pixels* running at 25 *fps* with an exposure time of 1.5 *ms* and a baseline of 20 *cm*. The choice of the vision setup is based on the statement that only high resolution is required, and that high frame rate is unnecessary since ball flight is well predictable. This is in contrast to the frame rate/resolution experiments presented in [Pon09, pg. 62ff]. Image data is processed with a circle detector and fed to a Multiple Hypothesis Tracker, that is handling multiple unscented Kalman filters (UKFs). The motion model used in the UKFs is considering gravity and (nonlinear) air drag. Prediction is done on propagating position and velocity. The visual tracking data equals 100 $\frac{MB}{s}$ (*Megabytes/s*) and is processed onboard of the robotic platform. Also, the control system for the movements is running on board due to the high control rate (1 *kHz*) while the path planner (responsible for the prediction of the ball's trajectory and the determination of the commanded joint paths) for the whole platform is running off-board. The path planner is the same as in [BWH10] and outputs soft motion for minimal joint accelerations, thus computing smooth and optimal joint paths [BBW⁺11, pg. 518]. The head movements are obtained by an inertial measurement unit (IMU) which raises the challenges of drifts in the measurements. The spatial calibration of the cameras, arms and IMU is done with a checkerboard and the temporal calibration is based on manually matching predicted ball positions. For synchronization of the sensor's measurements, the modules' timestamps and clocks are synchronized to 100 μs within minutes after system startup with a dedicated time synchronization demon [BBW⁺11, pg. 516]. Furthermore, the subsystems are de-jittered, and the latencies of the subsystems are calibrated in order to compensate them. Feedforward tracking terms are used to improve the overall accuracy and allow to consider the dynamic coupling between torso and arms. Regarding the calculation time the following numbers

are given: circle detection for an image pair takes 25 *ms*, tracking takes 5 – 10 *ms*, prediction and path planning (similar to [BWH10]) takes 60 *ms* for planning 7 DoF. This rather low time is considered as important, as the prediction changes significantly over time [BBW⁺11, pg. 518]. This statement is rather interesting as it is in contrast to the previously mentioned statement about the well predictable flight trajectory. Most probably, this problem lies in the tracking system. Experiments on different frame rate/resolution combinations would be interesting here. The results achieved is that "Rolin' Justin" is able to catch two balls simultaneously with a success rate of 80 % [BBW⁺11, pg. 513]. In contrast to the expected challenges the kinematic subchain head-hand improves the accuracy and releases the requirements for compensation of torso-movements. Moving from a kinematic⁴ to a dynamic⁵ movement planner is stated as future work, but the hard timing constraints make this a challenging tasks.

An approach on robotic catching focusing on learning is presented by Kim et. al. [KSB14]. The basics are already presented in an article published 2010 [KGB10]. A time-independent version of dynamic motor primitives is presented and used to represent the learning data from the motion of forty human catches [KGB10, pg. 108]. Also, the object's motion dynamics are learned based on a low number of 5 to 6 throws [KGB10, pg. 109]. The results are used to control a 7 DoF robotic arm. The arms are controlled by using Cartesian control. No further information about the trajectory tracking is given. The catching movement is limited to position the end-effector to minimize the movement towards the intersection point. The results presented are that the system is able to catch a flying ball successfully [KGB10, pg. 109] while no further details on a success rate or similar numbers are given. In addition, to this statement image-series of the humanoid robot are shown where a ball with a reduced motion space (rolling on a table or hanging on a wire from the ceiling) is caught. The successional work presented in [KSB14] extends to the point where objects with uneven shapes are caught by a 7 DoF robotic arm [KSB14, pg. 1]. The approach deals with all the aspects of catching a thrown object, namely trajectory prediction, prediction of the catching configuration and palling of the arm motion [KSB14, pg. 1]. Similar to the previously discussed research work the dynamic models for the thrown objects (a hammer, a tennis racket, an empty bottle, a partially filled bottle, and a cardboard box) is based on support vector regression [KSB14, pg. 4]. The information about possible grasping configurations is stored based on a set of training grasps/presented possible positions [KSB14, pg. 5]. This training set is the base to calculate a Gaussian mixture model (GMMM) for the reachable space [KSB14, pg. 5]. Furthermore the reachable space of the used robot, a KUKA LWR 4+ robot, is calculated (another GMM). In order to obtain the catching position both GMMs are multiplied resulting in a probability distribution. During the time when the object is within the resulting area, the likelihood of the respective catching configuration is calculated. These calculations are temporally separated by time slices. The configuration with the maximum likelihood is considered as the optimal configuration. For each new input of the object tracking system these steps are repeated [KSB14, pg. 7]. The motion capturing system uses markers and works with a sample rate of 240 *Hz*. The Pose information is filtered with a 25 *Hz* Butterworth filter. Both velocity and acceleration is determined based on cubic spline interpolation [KSB14, pg. 10].

The conducted experiments are based on 20 train trials [KSB14, pg. 13] feature an average flying time of 5 *s* \pm 0.5 *s* for the object [KSB14, pg. 13] and have a success rate of 73.3 % [KSB14, pg. 13]. The whole prediction is running on a 2.7 *GHz* quad-core PC. Whenever possible closed

⁴kinematic here is referred as based on geometrical and time-based properties of the motion (e.g. joint angles and/or Cartesian positions and their timely derivatives)[WGJ95, pg. 461]

⁵dynamic here is refereed as based on the forces and torques required to achieve a movement and considers the properties of the actuator (mass, inertia, stiffness) [WGJ95, pg. 461]

solutions were used to calculate the best catching configuration. In comparison, the research of the national aeronautics and space research center of the Federal Republic of Germany the computational power used by this approach is exceptionally low. As further details on the used tracking system are not given⁶. The failures of the system are classified as [KSB14, pg. 13]:

- too low robot velocity (12 out of 19)
- finger hitting the object (4 out of 19)
- robot's torque limit exceeded (3 out of 19)

A rather dubious comparison with humans is done, that shows a success rate for catching the respective objects between 10 % and 70 % [KSB14, pg. 14]. Even for the range of the presented objects, the success rate of humans for catching should be significantly above 10 %. Future work is stated in the areas of compliant catching, collision detection/avoidance and to model the robot's dynamics [KSB14, pg. 16] as well.

In 2000 Mehrandezh et. al. presented a method to intercept fast maneuvering objects. They defined the task of interception as "approaching a moving object while matching its location and velocity in the shortest possible time" [MSFB00, pg. 238]. Their hybrid approach consists of two phases. The first part is modified from navigation and guidance (missile interception) and the second part is tracking. The publication lacks any practical results but shows simulation data where the algorithm is compared to an algorithm that aims to minimize the angle between the trajectory of the object and the line of sight. In this comparison, the algorithm shows a shorter chasing time by 15 % to 30 % for the two most challenging setups. Here the two phases of the algorithm are interesting because research in the field of human catching also suggests these two phases (compare Section 2.3.3).

Aiming for a contactless interaction of a human with a robot Kober et. al. present work on robotic catching an juggling [KGM12]. A special robot with 39 *DoF* is used. The robot's actuators are mainly hydraulic driven. For the catching and juggling task 7 *DoF* of the arm and the 5 *DoF* of the fingers are used [KGM12, pg. 875]. The robot features a joint position controller with a command/sampling rate of 30 *Hz* [KGM12, pg. 876]. The maximum hand velocity is $\approx 1.5 \frac{m}{s}$. Both the control cycle time and the maximum achievable velocity of the are rather low in this setup compared to other approaches. The hand is modified for a more cup-like shape [KGM12, pg. 876]. This results in a tolerance of up to 5 *cm* variance [KGM12, pg. 878] for the catching position while still executing a successful catch. The sensor system used is a Xtion Pro, which uses active stereo at 30 *Hz* for a depth image and also delivers a color image with 640 by 480 *pixel*. The position of the ball and the human peer are determined based on the information from this sensor system. The flight trajectory prediction is based on a second order model (parabola). The initial parameters of the trajectory are determined with a Kalman filter with 5 *cm* standard deviation of the measurements and 5 *mm* of process noise [KGM12, pg. 877]. Reasons for choosing this arbitrary values is not given. The magnitude of the measurement noise gives a rough estimation of the tracking system's accuracy. A ball with a mass of 100 *g* and 7 *cm* diameter is used. Modeling the drag for the calculation of the flight trajectory is considered not required. This is in clear opposition to [FBH⁺01, pg. 1626] where the influence of drag on the flight distance of different balls is discussed. The usage of a significantly heavier (+100 %) ball with a slightly bigger diameter (between +2 % and +5 %) reduces the effect but still a reduction of ≈ 2 % is resulting.

⁶Additional information on the technical background of the tracking system was not available on the website of the vendor either

On a distance of 2.5 *m* the error of the trajectory model has the same size as the tolerance of the catching device (5 *cm*). The catching position is obtained by intersecting a predefined plane with the predicted trajectory [KGM12, pg. 877]. The area of possible catches is determined considering the limitations of the robot’s arm kinetics [KGM12, pg. 878]. For the success rate, two numbers are given. 75 % of the catches in the reachable area are successfully caught while the overall success rate is 47 %. The following reasons for the failures are given [KGM12, pg. 880f]:

- low control frequency
- low maximal velocity and acceleration
- delays and low frame rate of vision system
- inaccurate initial prediction
- errors due to linear model for inverse kinematics

For future work the extension of the catching movement to more joints of the robot and an extension of the catching area from a single plane are given [KGM12, pg. 881]. Both extensions result in more complex motion planning.

Own Contributions to Catching Objects

Within the research team, consisting of members at the Fachhochschule Heilbronn, the Bergische Universität Wuppertal and the Vienna University of Technology, the first idea of using automatized throwing and catching for object transportation was done by Frank et. al. in 2006 [FWWH⁺06]. In order to do an early examination of the feasibility of the concept an accuracy determination of a 64 *by* 48 *pixel* depth camera, delivering 50 *fps* was done. This experiment showed the technical feasibility.

In 2008 Barteit et. al. published a numerical analysis of prediction errors. The system examined consisted of a single camera for object tracking and a light barrier for the determination of the initial ball velocity. The setup allowed to use the measured velocity for determining the distance of the ball to the camera, thus delivering depth information [BFK08, pg. 893]. The image of the camera was processed with background subtraction (adjacent frame differencing), thresholding, filtering and Sobel filter based Hough transformation [BFK08, pg. 895] (more details in [Bar10, pg. 65ff]). For the trajectory prediction the motion is assumed to be within one plane [BFK08, pg. 897] and a model considering quadratic drag is used for the motion along both axes of the plane (the x-axis is aligned with the gravity vector) (details again in [Bar10, pg. 97ff]). The final error analysis of the prediction shows the following results: Considering 45 throws 29 of them had a horizontal prediction error of less than 12.5 *mm* and the respective value for the vertical prediction error is 28 throws. A prediction error bigger than 20 *mm* was shown in less than 10 % of the throws. This prediction error is significantly smaller than the tolerance of the catching device used in [KGM12] by example and in the same magnitude as the required prediction error determined by Bäuml et. al. [BBW⁺11, pg. 512]. Details of the measurement system for measuring an object’s position within a plane is discussed in [BFPK09]. Two variants for gaining the position information are given. The first is based on a grid of light barriers that trigger a camera. Here the influence of the distance between the light barriers and the achieved accuracy is given. The second variant is based on a touch screen and thus invasive. For both methods, the achievable measurement accuracy is within 2 – 3 *mm*.

Complementary to the research work targeting the prediction of the ball also the development of a control scheme for a Cartesian robot was done in 2008 [FBM⁺08]. The Cartesian robot planned to be used for catching an object has a maximum acceleration of $a = 50 \frac{m}{s^2}$ for the two dimensions. The control algorithm is based on a square with a shrinking size based on the increasing prediction accuracy. The robot is positioned in one corner of the square and smoothly moves during while the predicted impact position is updated. Due to the limitation of the robot to 2 DoF, the catching is exposing the object to high forces. The interaction with the vision system is not finished at this stage, so no results for the success rate of the whole system is given. This information is available later in [Bar10] with a modified system where the light barrier is made obsolete. Instead of measuring the initial velocity of the ball and using this information for the calculation of the depth (the distance from the ball to the camera), the size of the known ball in the image is used for determining the depth information [Bar10, pg. 101ff]. The prediction is done based on an analytical description of the flight trajectory with two simplifications: the trajectory is assumed to be within a plane, and quadratic air drag is considered in an iterative model. The second approach is an improvement over [BFK08] due to the fact that the relation between the drag for the both dimensions is incorporated. The resulting success rates for the whole systems are 80 % for 35 mechanically thrown tennis balls and 60 % for 70 human thrown tennis balls (thrown from 4 different positions).

In 2009 parallel to the work on point symmetrical objects (tennis balls) also, the examination of the flight behavior of more complex objects was started. The objects under investigation were cylinders, where the stabilization behavior was examined [FMS09]. The continuance and more elaborate work targeting soft catching of cylindrical objects with and without sensor input are given in [Fra12]. On main aspect here is the accurate throwing of the cylinders. This requires an advanced model of the flight behavior. The previously mentioned self-stabilization property of the cylinders is used. This self-stabilization occurs when cylinders are thrown approximately along their axis of symmetry. The extent of this behavior depends on the geometrical properties of the cylinders. Frank examines the possibility to throw and (softly) catch two cylinders, with different degree of self-stabilization) [Fra12, pg. 35ff]. Two different experimental setups are used: the first has a one DoF catching arm with a simple gripper in fixed position, for the second setup the gripper can be rotated around an axis parallel to the main axis of the arm to compensate rotation of the cylinder during flight. The additional sensors for the determination of the flight attitude are two distance sensors. The success rates for the catching experiments are: 98 % for the more compliant flying cylinder with the fixed gripper, 64 % for the less compliant flying cylinder with the fixed gripper and for the later case with the usage of the gripper rotation a catching rate of 85 % is achieved.

Akhter presented work on the Pose⁷ in 2011 [Akh11, pg. 69]. The approach is using a combination of feature tracking and homography to increase the efficiency and accuracy of the overall tracking procedure. The work is focusing tracking and uses standardized image series including the ground truth data for evaluation. One main requirement for this approach is that the object has, at least, a partially planar surface [Akh11, pg. 69].

Earlier work of the author is targeting only the prediction without the use of a catching robot or catching tool [PKFB10] [Pon09]. The focus of this work lies in the comparison of different analytical approaches for trajectory prediction. Three models are derived and applied to the task of prediction for a flying tennis ball. The first model is a second order polynomial model with independent movements for each direction (compare Equation 2.1).

⁷Pose here is the combination of position and orientation

$$\mathbf{p}(t) = \mathbf{p}_0 + \mathbf{v} * t + \mathbf{a} * t^2 \quad (2.1)$$

One axis of the reference coordinate system is aligned with the direction of gravity. The second model is a spatially separated model where the physics of the ball are considered (gravity, air drag). Each spatial direction is considered independently. This results in a closed solution for the description of the trajectory and allows least square fitting to the measured positions of the ball (compare Equation 2.2, Equation 2.3, Equation 2.4 and Equation 2.5).

$$x(t) = x_0 + \frac{1}{k} * \ln(1 + k * t * v_{x0}) \quad (2.2)$$

$$y(t) = y_0 + \frac{1}{k} * \ln\left(\frac{\cosh(\sqrt{g * k} * (t - t_0))}{\cosh(\sqrt{g * k} * t_0)}\right) \quad (2.3)$$

$$z(t) = z_0 + \frac{1}{k} * \ln(1 + k * t * v_z(0)) \quad (2.4)$$

$$k = c_W * A * \frac{\rho}{2 * m} \quad (2.5)$$

The third model is based on the flight physics (gravity, air drag) and due to the non-linearity of the air drag, a closed solution is not possible. The trajectory has to be calculated iteratively. Due to this property, a Monte Carlo simulation is done to fit the model to the measured points. The related equation for the third model is given in Equation 2.6, Equation 2.7 and Equation 2.8.

$$\mathbf{v}(t + \Delta t) = \mathbf{v}(t) + (\mathbf{a}_{drag}(t) + g) * \Delta t \quad (2.6)$$

$$\mathbf{p}(t + \Delta t) = \mathbf{p}(t) + \frac{\mathbf{v}(t) + \mathbf{v}(t + \Delta t)}{2} * \Delta t \quad (2.7)$$

$$\mathbf{a}_{drag}(t) = -\frac{c_{drag} * \rho * A * v(t) * \mathbf{v}(t)}{2 * m} \quad (2.8)$$

The experimental evaluation with a stereo camera system and the invasive position determination system presented in [BFPK09, pg. 684] shows that the spatially separated model is the most stable model. This is especially true for a position detection system with larger errors in the early flight phase. An additionally evaluated aspect is the influence of the temporal and image resolution of the camera system on the prediction accuracy. Here a constant bandwidth limitation (camera interface) limits the product of image resolution and frame rate. The result is that higher temporal resolution leads to a more increased prediction accuracy than increasing the image resolution.

2.2.2.2 Tracking Systems

Due to the big amount and variety of the approaches to robotic catching the subject of tracking systems will be briefly discussed in this subsection again. For tracking systems, a more wide definition is used than by example the definition of Lampert [LP10]. Lampert differentiates between per-frame detection (object detection in a still image) and tracking (region of interest based on the object's motion). Here both approaches are considered for the tracking system with the argumentation that both approaches lead to the information of the object's trajectory in space

within their field of application. Two parts of the tracking systems are elaborated on the sensor delivering the input the further processing stages and the processing stages to deal with this data. The output of the tracking system in the sense of the author is the position of the tracked object over time.

Sensor

Recapitulating the variety of approaches discussed in the previous section in regards of the tracking system the number of sensors and sensor configurations is rather limited. The main categories are given here:

- Special sensor [IMT92] [NI03] [KYH13] [Fra12]
- Single camera [LRÅJ10] (also dual camera) [BFK08][Bar10]
- Dual camera but no stereo [NIN⁺97] [LRÅJ09] (also stereo) [LRÅJ10] (also single camera)
- Stereo vision system [HS91] [HES95] [SC07] [LRÅJ09] (also no stereo) [FBH⁺01] [BWH10] [BBW⁺11] [BFB11] active stereo: [KGM12]

The special sensors are either high-speed black/white sensors specifically designed for the domain of real-time control in the case of [IMT92] [NI03] [KYH13] or the combination of two distance sensors used by [Fra12]. While the earlier have the advantage of delivering high resolution (spatial and temporal) information, the downside is that they are very specific hardware and the application of these sensors in the (more general) context of transport by throwing or automation causes rather high costs. The combination of the two distance sensors is suitable for the detection of the pose of an object within a certain plane [Fra12]. Tracking the object throughout the flight is not possible with these sensors in general.

Using cameras for the task of object tracking in the highly dynamic domain of robotic catching is very common. Still different approaches exist. The usage of an individual camera by Linderoth et. al. is possible but shows rather big measurement uncertainties [LRÅJ10, pg. 4529]. In comparison, Barteit uses either the size of the object or an additional sensor for the velocity measurement of the object to determine the depth (distance from the camera's focal point to the object) [BFK08] [Bar10]. Here the accuracy of the whole system is sufficient to allow successful catching. Still the measurement error for "optimal" circumstances [Bar10, pg. 104f] and for tacking of a thrown ball [Bar10, pg. 106f] are rather big, especially for larger distances of the ball to the focal point of the camera.

The approaches of dual camera setups used differ significantly. Nishiwaki et. al. [NIN⁺97] use a similar approach to Barteit [BFK08] [Bar10], where the size of the object is used for depth information. More details on the combination of the information from both cameras is not given. Still both cameras of the humanoid robot are used. Besides synchronized stereo Linderoth et. al. also use a Kalman filter with measurements from both cameras. Due to the different viewing angles of the cameras on to object the resulting state of the Kalman filter incorporates both pieces of information and allows to track the thrown object [LRÅJ09] [LRÅJ10].

Using a stereo vision system is by far the most common input sensor for the task of robotic catching [HS91] [HES95] [SC07] [LRÅJ09] [FBH⁺01] [BWH10] [BBW⁺11] [BFB11] [KGM12]. While the implementation details (camera type, interface, resolution, frame rate) differ largely

gaining depth information, the main advantage of the stereo system is warmly welcome to in the largest proportion of the experiments. Passive stereo cameras, consisting of two cameras, are the main part but also one approach with active stereo using the Xtion Pro (similar to the Microsoft Kinect) exists [KGM12]. Here the rather low frame rate of 30 *fps* is the main disadvantage. Other stereo camera systems operate in the range of 50 *fps*. The main advantage of a stereo camera set is the big range of available cameras and the expertise available on object detection.

Processing

When acquiring information via sensors, usually a first preprocessing step of the data is done in order to suppress outliers or measured data with a high error probability. Regarding the first processing steps in the research work discussed in the previous section two main variants are distinguishable. The first approach uses no processing of the data and the input information of the sensor is the base for the prediction/control algorithms ([HS91], [HES95], [NIN⁺97], [RA00] and [RA02]).

The second option is to use a Kalman filter. Here different variants of the filter are used. The initial version, designed for linear systems, is used by [LRÅJ09] [LRÅJ10] but clearly seen as a first step with the option to use an extended Kalman filter in future. This variant of the recursive filter was designed for nonlinear systems and is using a linearization step. The application of this variant is common ([SC07], [FBH⁺01] and [BWH10]). The most recent variant of the Kalman filter, the unscented Kalman filter uses a sample based technique and shows improvements for highly nonlinear systems. Due to the recent development this variant is only used by one implementation discussed in the previous section ([BBW⁺11])

While most approaches give only a rough sketch on the implementation stages of the processing like mentioning the algorithms use (e. g. Hough transform for ball/circle detection) the reports about the vision system by Birbach et. al. [BFB11] is outstanding in the level of details. The publication elaborates on the vision system used developed for the experiments of the national aeronautics and space research center of the Federal Republic of Germany, especially on the stage when the mobile platform "Rollin' Justin" is used. The stereo vision system for this platform has a low baseline and for compensation a high resolution is used [BFB11, pg. 5955f]. The system allows usage in an environment with an undefined background due to the robustness of the used multi-hypothesis tracker. The multi-hypothesis tracker features unscented Kalman filters for the ball's trajectory and environmental factors like gravity, air drag and the aerodynamic properties of the object (e. g. diameter) are learned. Being an improvement of previous work presented in [BF09] the circle detectors output are likelihoods with a small number of false positives. This output is used for the multi-hypothesis tracker that filters the false positives based on the most probable trajectories in the tracker. Here the question arises whether the opposed approach would not be more efficient. Using the multi-hypothesis tracker to predefine the search space for the circle detection might be of higher efficiency. Still a new way to detect new balls or their trajectories would be required. The circle detection is optimized as well. An improve Hough transform with the following features is introduced:

- Usage of a contrast normalized Sobel filter
- Improving the circle response of Hough-transform by replacing the hard thresholds
- Searching for circles in a multiscale pyramid to detect circles on a coarse level and refining the results

- Efficient coding and usage of SIMD SSE3
- Other measures

The result is a circle detection system where no hard thresholds are used; the detection of the false positives is outsourced to the multi-hypothesis tracker, and a desktop PC is faster than needed by a factor of 2.5. The usage of a ball wrapped into retro-reflective foil results in better optical properties and help to improve the performance of the vision detection algorithms. The overall latency of all processing step from image acquisition to data in a pot for the planning algorithm is 75 *ms*. In the final part of the report, the lessons learned are given. One of them is dealing real-time behavior versus performance. The operating system used was Linux and on one affected machine time-slices in the order of 50 *ms*, were lost. This behavior could only be avoided when activating several hardware components. Another problem was the lack of a global clock. Especially for sensor data fusion, a common clock would have been beneficial. In terms of sources of the prediction errors a wide variety, including sensor noise, calibration errors and timing problems, were given.

Further improvement of the system discussed above is presented in [BF11]. Instead of the multi-hypothesis tracker a Gaussian Mixture Probability Hypothesis Density filter is used. Using the same hardware platform with two cameras and an inertial measurement unit, the same tracking performance is achieved. The benefits are a reduced number of code lines (619 versus 3262 [BF11, pg. 3432]) and only 50 % worst case calculation time [BF11, pg. 3432].

Color based tracking with a movable, small-baseline stereo camera system is discussed by Kao et. al. [KYH13]. The movement of the stereo camera system is based on the position of the ball. The advantages of using a Kalman filter for this are discussed [KYH13, pg. 374f]. Using the Kalman filter, a smoother trajectory is extracted, and the compensation of delays is possible resulting in a successful real-time tracking system [KYH13, pg. 376].

2.2.2.3 Trajectory Prediction

The following subsections are devoted to the trajectory prediction aspect of the approaches discussed in Section 2.2.2.1 and additional aspects. Trajectory prediction is divided into two parts: the first where a physical model is the foundation for the prediction and the second where (machine) learning is used to predict the future trajectory. Most related work is focusing on the physics-based prediction. In addition to the prediction for catching systems also, the prediction for batting systems is discussed. Robotic catching using machine learning is limited to mainly one research group. This work is very recent on the other hand.

Physics based Trajectory Prediction

Trajectory prediction based on a physical model of the flight's dynamic is very common. Except the work from two research groups, discussed in the second part of the following paragraph, all approaches to object catching use models considering gravity and possibly air drag with differing levels of modeling accuracy. In addition also research work, mainly focusing on batting (ping-pong), is discussed as the task of trajectory prediction is required here as well.

Prediction in Catching Experiments Applying automatized throwing and catching on the task of transportation requires a distance in the range of 2–4 m between the origin and destination. Due to this distance the variances of the object’s trajectory at the destination site vary with a larger extend [Bar10, pg. 8]. Due to this behavior, the requirement to position the catching device timely arises. Trajectory prediction is required.

The successful realization of transport by throwing and catching heavily relies on an accurate prediction of the object’s position over time (trajectory) [Bar10, pg. 9ff] in the case of point symmetrical objects. In addition, the prediction of the orientation over time is required for more general objects with a lower degree of symmetry. The trajectory of the object is influenced mainly by gravity and the aerodynamic forces during the flight. Simple models for the calculation of an object’s trajectory exist but when more accurate and detailed calculation of aerodynamic behavior is required a shift towards computational fluid dynamics occurs. Due to the high computational expenses of computational fluid dynamics a range of motion models, describing the trajectory of a thrown object, have been developed. A variation of the considered impact factors and forces exists. Different approaches and their advantages and disadvantages will be discussed here.

One main group of models for trajectory prediction relies on the trajectory prediction with curve (least squares) fitting [HS91] [HES95] [NIN+97] [RA00] [RA02] [Pon09]. The approaches within this group differ by the abstraction level of the trajectory modeling. The most simple way is to consider only gravity and neglect all other aspects (air drag, Magnus force..). The solution for the trajectory then can be found in a parabolic equation (compare Equation 2.1). This solution is used in the following work: [HS91] [HES95] [NIN+97] [RA00] [RA02]. Another approach of finding a closed solution for the description of the flight trajectory is presented in [Pon09]. There the influence of air drag (quadratic model) is considered but, in order to find a closed solution, the movement of the object is considered as independent for the three spatial directions.

A quite individual approach is done by Barteit [Bar10, pg. 113ff]. The estimation of the launching parameters is done based on the current measured position of the thrown ball, the initial position of the ball and a parabolic model. After this step of estimation, the predicted trajectory is calculated based on an iterative model considering air drag (quadratic model, similar to Equation 2.7. In contrast to the following group, where a version of the Kalman filter is used for the estimation of the launching parameters, the Kalman filter’s initialization is considered as limiting this approach to the usage.

The group of using a Kalman filter for parameter estimation and an iterative model for the calculation/prediction of the flight trajectory can be split up into three subgroups. This partitioning is possible based on the variant of the Kalman filter used. Linderöth et. al. used a Kalman filter with a model where only gravity is considered [LRÅJ09] [LRÅJ10]. On one hand this might be related to the usage of darts as thrown objects, which have a lower degree of air drag due to their shape, but on the other hand also experiments with a thrown ball are presented [LRÅJ10]. Interestingly is the fact that the orientation is considered to be important for the prediction task for the dart’s due to the distance between the tracked tail of the dart and the relevant tip. This consideration in the model for the Kalman filter is not done. The updated version of the Kalman filter for observation and prediction of nonlinear dynamics, the Extended Kalman filter, is used by [SC07] [FBH+01] [BWH10] in combination with the iterative model for the calculation of the future flight trajectory. The filter uses a local linearization in order to be applicable to nonlinear problems. A better estimation of the initial parameters should be possible due to this, but a distinct comparison is not done. The third group of approaches used the most recent variant of the filter, the Unscented Kalman filter. This update uses a unscented transform to accommodate

the nonlinear dynamics. The only implementation using this variant for parameter estimation and an iterative model for trajectory prediction is done by [BBW⁺11].

Another individual approach to flight trajectory prediction is done by Frank et. al. [Fra12, pg. 27ff]. There a computational fluid dynamics simulation is used to determine the required launching parameters in order to throw to a certain position beforehand. This can be considered as accurate throwing. Depending on the flight dynamics of the thrown cylinder an additional sensor is used to determine the orientation of the cylinder in the catching area and align the catching tool [Fra12, pg. 17ff, pg. 87ff].

Prediction in Batting Experiments Work targeting the task of robotic ping-pong is done by Sun et. al. [SLW⁺09]. The examination of the dynamic model of a ping-pong ball is done. This model is validated due to experiments and shows a accuracy better than 1 mm [SLW⁺09, pg. 2381]. The statement on the effect of the initial velocity's deviation on the determined collision point shows the high sensitivity of the trajectory to deviations in the launching parameters. A difference of 1 % in initial velocity gives 0.5 cm change in collision point. An improvement over the state of the art tracking algorithms is shown in experiments.

A German research group also used a physical model, that considered gravity and linearized air drag, for prediction [MKP10]. The estimation of the initial parameters was done with an Extended Kalman filter. Specific for the approach is that the human hitting was used as an archetype. This resulted in the usage of four distinct stages for the ball hitting: the awaiting stage, the preparation stage, the hitting stage and the finishing stage. The simulation of this approach was successful, but the real experiments showed problems because of the neglected spin and inaccuracies of the vision system.

Besides modeling the flight behavior the work presented in [SFXT13] also models the rebound of a ping-pong ball on the basis of physics. The model is considering gravity and air drag. The simplification done is that the movements in the three spatial directions are considered as independent [SFXT13, pg. 2894]. This is done regardless of the introduced error. Additional fuzzy filtering before prediction allows to reduce the influence of measurement noise. On contrast to all other approaches, where a variant of the Kalman filter is used for this task, this renders the approach unique. This approach shows improved real-time performance of prediction compared to State-of-the-Art algorithms.

Machine Learning for Trajectory Prediction

The number of research papers devoted to the prediction of a flying ball's trajectory is very limited. Kim et. al. do an extensive test of various machine learning algorithms for trajectory prediction of unsymmetrical objects. The learning methods compared are Support Vector Regression with Radial Basis Function kernel (SVR), SVR with a polynomial kernel, Gaussian Mixture Regression, Echo State Network, Genetic Programming and Locally Weighted Projection Regression [KB12, pg. 1108]. For the evaluation, five different objects are used: a ball, a fully-filled bottle, a half-filled bottle, a hammer and a ping-pong racket. In order to track the objects for the practical evaluation a 120 Hz motion tracking system is used. The tracking is done with additional marker on the objects. Each object is thrown 20 times [KB12, pg. 1112] for learning purposes. The data of the motion tracking system is filtered with a 25 Hz Butterworth filter. The acceleration and velocity are calculated based on cubic spline interpolation. Support Vector Regression with Radial Basis Function kernel is found to be best [KB12, pg. 1120]. The advantage of this approach is seen in the

omission of the need of background knowledge about the object's properties. Only a few samples are needed for the prediction. These samples, on the other hand, have to be representative. The work is assuming that the position and orientation extraction is possible in real-time which is about to be tested in future work.

This future work is the content of another publication from the year 2014 [KSB14]. The catching of the objects mentioned above is the main objective. A minor change is done to the set of objects as the ping-pong racket is exchanged with a tennis racket. Prediction of the trajectory of the thrown object, as well as the catching configuration in combination with the planning of the arm motion, is realized. The grasping configuration is stored, and the dynamic model of the thrown objects is learned with the Support Vector Regression discussed in the previous work [KB12]. A separate SVR model is used for each of the objects. The dynamics of the thrown object is modeled by the following equation where ξ is the combination of position and orientation.

$$\dot{\xi} = f(\xi, \dot{\xi})$$

Similar to the findings in [KB12] an Extended Kalman filter is used for robustness reasons [KSB14, pg. 5].

2.2.2.4 Catching Movement

The catching movement implemented in the variety of research work discussed in Section 2.2.2.1 can be divided into two groups. The first is "hard" catching where the catching device is moved to the predicted interception position. In this case, the velocity difference between the object and the catching device is the full object velocity causing mechanical stress on the object and the catching device. This approach is used by [HES95], [NIN+97], [NI03], [SC07], [LRÅJ09], [LRÅJ10], [Bar10], [BWH10] and [KGM12]. Advantages of this approach lie in the released timing constraints (if the catching device is positioned right no further movement is necessary, and the ball will fall into the device) and the more simple determination of the interception position and path planning.

The significantly small group is implementing "soft" catching where, besides the catching position, also the object's velocity is considered. Depending on the dynamic properties of the catching device it is moved with the full velocity of the object or a proportion of it in order to minimize impact energy. This approach is used by [HS91] and [Fra12]. The advantages here are the reduced mechanical stress on the object and the catching device, also enabling catching of more fragile objects, while the whole process, including catching time determination and interception trajectory planning, is more complex.

Park et. al. presents ball catching based on captured data from human catching and a learning approach with evolutionary algorithms [PKK+09]. The catching is done in the hard way where simply the position is used for the control of the robotic arm. The survivor selection of the candidates is based on the minimal torques without violating the joint limits. [PKK+09, pg. 811]

2.3 Biological Archetype for Catching

One common attribute of the research works discussed in Section 2.2.2.1 is that the success rate is in the region of 80 % or below with one exception, the work of Frank [Fra12] where specific cylinders are caught with 95 % success rate. When comparing this attribute to humans, even

children, the success rate is remarkably low. Skilled human catchers are able to catch balls with more than $100 \frac{km}{h}$ [5]. This fact puts the human or similar skilled biological systems into the role of archetypes. The national aeronautics and space research center of the Federal Republic of Germany mentioned for their extensive research on robotic catching in the previous section is currently investigating on the topic of human-like ball catching [3].

Biomimicry can happen on three different levels. These are form, process, and ecosystems [1]. In this case, the process of object catching is the relevant information. In the following subsection research work investigating human catching will be discussed. Similarly to the discussion of the technical approaches in Section 2.2.2.2, Section 2.2.2.3 and Section 2.2.2.4 the topic will be divided into three topics: the visual system, the information processing (prediction) and the catching movement. Still in case of a human catching a clear isolation of these different functions is hard to do as the research discussed shows how tightly these aspects are coupled.

2.3.1 Analysis of the Visual System During Catching

Studies on the movements of head and eyes during catching are rather rare. This is caused by the high dynamics of this task and the limited availability of suitable tracking systems. Only in the most recent years surveys on this topic have been made. Mann et. al. is elaborating on the observation of extremely skilled cricket batters during trajectory following and batting [MSA13]. Instead of moving the eyes during the phase of trajectory following they used the head to keep the ball straight in front of their head. The eye movements were used for moving to the bounce point with a predictive saccade⁸. Another predictive saccade was used to position the eyes on the batting point. This allowed to watch the ball during batting [MSA13, pg. 10]. A similar behavior is observed by the absolute elite of tennis players [BEC09, pg. 38f]

Another survey by Uchida et. al. shows that baseball batters have eyes that can move faster than reference persons [UKH⁺13]. A relation to baseball training is suggested here. Interestingly this is in contradiction to the research work mentioned before where the very familiar task of tracking a cricket ball is solved mainly by the movement of the head.

An earlier work on investigating the human head, eye and arm movements during playing catch is presented by Hayhoe et. al. [MNBK05]. The dynamics of this task are significantly lower than by example during baseball, cricket or tennis due to the cooperative throwing and smaller distances. In this investigation, the role of the internal model for the task of catching a bouncing ball is done. During a series of throws/bounces/catches within a circle of test persons, the ball is secretly exchanged by a ball with different bouncing dynamics. For the first few catches, the tracking errors of the eyes are increasing and thus the direct impact of an internal model of the ball's dynamic properties is proved. A relation between the positioning of the eyes, dynamics of the object and expected behavior is obvious. After approximately three catches the internal model is updated, and the tracking accuracy is close to the initial tracking accuracy [MNBK05, pg. 79f].

2.3.2 Bio-Inspired Information Processing

Visual information is the main input for human catching. Research on the properties of the visual information pick-up by Förster shows that the structure of neurons in the early stages of

⁸this saccade is used to position the eyes in the bounce point prior to the ball reaching there allowing to observe the complete bounce

the information processing of visual information causes the calculation of abstract properties like edges, corners and similar objects [Foe93, pg. 42ff] [Foe99, pg. 37f]. This happens already on the first few layers of neurons, and thus, the information for the following layers are abstract information. By using similar structures again, even a greater degree of abstraction is possible (e. g. ball, cup or similar objects). Förster suggests recursive calculations based on the neuronal nets and calls this "Metaprogramm" or "Meta-Metaprogramm".

The interaction between knowledge and perception is discussed by Blakemore et. al. in [BC70]. Based on an experiment with a cat, that is not able to perceive horizontal lines and thus crashes into horizontal sticks the assumption is put up that biological systems like cats and humans are only able to perceive what they know already [BC70, pg. 477f]. A sensitive phase at the age of three weeks to three months is given for kittens. Especially the age of 28 days is given as a peak sensitive phase where only one hour of exposing to a vertical bar on that day develops nearly all neurons of the cortical area to prefer vertical edges [2]. According to Pratl this is also valid for other senses like the perception of acoustic information. The example given is understanding a song's lyrics [Pra06, p. 16]. Also, in this case, the existing knowledge is used and fitted to the heard sound. The result might be a Mondegreen, a misinterpretation of the perceived phrase. This clearly shows that perception is a construction process [Pra06, p. 16] and that the linkage between the sensors and the experience/knowledge is very tight. Pratl thus also postulates a symbolic (abstract) representation for physical values in the brain [Pra06, p. 35f]. In terms of visual perception, this means that instead of using metric measures to classify velocity, symbolic representations like "before" or "after" are used. Velik et. al. develops this symbolization further for multi-modal perception [Vel08]

The tight interaction between knowledge and perceptions requires to have a look at this aspect as well. The human memory can be divided into short-term and long-term memory [Sol02, pg. 143]. The long-term memory is further divided into three sub-memories by Schacter et. al. These memories are the semantic memory, procedural memory (perceptuomotor or ideomotor skills) and episodic memory. Deutsch et. al. shows that the usage of episodic memory has a positive impact on the lifespan of agents in a virtual world [DGLV08]. Regarding the task of ball catching the episodic memory (for the trajectory perception and prediction) and procedural memory (for catching) are required. Most probably all memories are involved for the highly complex task of ball catching.

A study done in zero gravity shows that also gravity is modeled in our brains [MZBL01] [4]. The task of ball catching in zero gravity is used to examine the internal model of gravity and the adaption to a new environment. The analysis of the arm's movement during catching shows that besides continuous update still a general model for the flight trajectory is used. In comparison to the fast adaption during approximately 3 throws in the case of the experiments by Hayhoe [MNBK05], discussed in the previous section, here the adaption takes about 15 days. During these time, the model kept persisting [MZBL01]. Two aspects here are significant: The model of the brain is much more persistent than the model of the ball's dynamics in the previous section. A reason for this might be the permanent experience of gravity during our life in comparison to a wide variety of ball dynamics that might be perceived. On the other hand, the usage of a model might be too limited. Certainly there is a representation of the permanent experience of gravity in the brain but how this representation is stored is unclear. This might be a model, an exhaustive set of experience or another representation. Obviously, it's there and can only be altered after a relatively long time.

Recent work regarding the time perception in the human brain shows a relation between the visual

perception and the encoding of time [SMKB13]. The very early stages of visual processing are involved in the time encoding. These stages might be the same stages that construct abstraction according to Förster [Foe93, pg. 42ff] [Foe99, pg. 37f]. The research work does not rule out a distributed or centralized temporal mechanism but clearly shows the involvement of the primary visual cortex and extrastriate areas (areas responsible for general perception) in this mechanism. This indicates a tight coupling of sensory input (visual, tactile or other senses) and the perception of time.

Succeeding studies by Gavazzi et. al. show that the type of the movement impacts the perception of the duration of the movement [GBP13]. In case, the movement matches a movement produced by human kinetics the accuracy and precision of time reproduction as well as movement reproduction are significantly higher than when the movement is artificial. The research work thus suggests a calibration of the time perception based on the models of action and the visual representation of the human motor repertoire [GBP13, pg. 1169]. This is considered especially for internal models of action. This opens the possibility that the brain is calibrating time-based on the environment (Newtonian mechanics) [GBP13, pg. 1169]. These findings are in line with the idea that the individual motor repertoire of the observer is a reference frame for visual events. This information might be crucial for the temporal processing.

In terms of time perception, a remarkable statement from Buetti is a good final thought for the section about human brain functions like perception or other processing. He stated that giving a relation between brain area and the mechanisms for functions like time encoding is suboptimal and that a more mechanistically oriented approach, targeting the underlying mechanisms, should be used [Bue11, pg. 2].

A more technical approach is given by Namiki et. al. [NHI03]. An architecture mimicking the human movement structure is established. This is a hierarchical/layered system connected with efferent/afferent information lines with five layers. Level 4 and 5 are for task switching, level 1-3 for the execution of selected task. An interesting aspect is that the information flow from the higher levels to the lower levels (commands or efferent information lines) are processed in each layer before leading to a final movement. On the other hand, the sensory inputs (or afferent signals) are only processed once an each layer receives the same information. Regarding the work of Förster the aspect of abstraction of input information is not considered at all while, on the other hand, the specialization of the movement information in each layer is considered.

2.3.3 Analysis of the Catching Movement

The studies on human catching movement are targeted at distinguishable parts. One is the catching movement of the whole body (relocation) in order to catch a ball similar to the task of a baseball fielder. This movement is not relevant and thus not discussed in this section. The second aspect is the movement of the hand in order to catch a flying object/ball. The latter task is also required in the first one, so this task can be seen as a specialization or simplification of the catching problem.

Early work of Peper et. al. is examining the types of information that are used for catching a ball. The tested variants are the ball's size and the angle of approach. The ball's size could be used for velocity and distance estimation based on the knowledge of the object. A variety of experiments is done and the results clearly indicate that the human is not predicting the trajectory and executing the catch but continuously updating the information. This also shows a continuous

coupling between perception and action [PBMB94, pg. 610]. This is in line with research by Dessing [DPBB05]. The experiments examine the usage of predictive⁹ or prospective¹⁰ strategies and the result is that a prospective strategy with continuous update is used [DPBB05, pg. 668].

In terms of soft catching by human (that means minimizing the contact force) Kajikawa et. al. conduct an experiment where an object has to be caught from a "conveyor belt" [KSOI99, pg. 699]. This task can be interpreted as 2D catching. The velocity of the conveyor belt is varied from $0.2 \frac{m}{s}$ via $0.4 \frac{m}{s}$ to $0.6 \frac{m}{s}$. Two tasks are set for the test subjects: grasp the object from the conveyor and grasp it with additionally minimizing contact force [KSOI99, pg. 699]. The research team divides the catching movement into four phases [KSOI99, pg. 701]

- approaching the object
- turn motion
- accelerate hand to reduce velocity error
- grasping

The second phase is rather a single time instant than a phase between phase 1 and phase 3 [KSOI99, pg. 701]. Here the general usage of this turning point is questionable as another starting position for the hand can lead to the omission of this point and thus the existence of this point is clearly related to the initial position of the hand.

Another approach considering experiments on catching in two dimensions is done by Flash and Hogan [FH85]. This study shows that the jerk (the timely derivate of acceleration, or third timely derivate of position) is minimized for human movements in Cartesian space. Interestingly a relation to joint space cannot be done [FH85, pg. 1698]. The question how the human motion control is considering Cartesian jerk cannot be answered.

A further approach considering the mechanics of the movement is done by Nakano et. al. [NIO⁺99]. The following four models for human movement planning are suggested [NIO⁺99, pg. 2142f]:

- the minimum hand jerk model in an extrinsic-kinematic space
- the minimum angle jerk model in an intrinsic-kinematic space
- the minimum torque change model in an intrinsic-dynamic-mechanical space
- the minimum commanded torque change model in an intrinsic-dynamic-neural space

⁹Definition for predictive: *"Predictive strategies assume that the catcher computes an a priori estimate of where the ball will go, selects an interception point along the ball's trajectory and then plans and executes a movement of the arm so as to place the hand at the selected interception point at the right time. The predicted interception point may be updated over time as new information about the target's trajectory is acquired, but the predictive strategy basically assumes that the movement has a defined endpoint that is computed some finite time before the arrival of the ball or the hand at that interception point."*[FMVDS12]

¹⁰Definition for prospective: *"Prospective strategies are, on the other hand, based on real-time calculations of how to direct the hand in such a way that it should intercept the ball at some time in the future, without any advance estimate of where the interception will occur."*[FMVDS12]

The used space definitions extrinsic is related to an external reference (e. g. external Cartesian coordinate system) while the internal space is the joints motion space of the human subject. The result of the research is that the internal space is used for movements which means that the muscle dynamics are considered for the movement planning. For trained movements the minimum torque change model describes the movement best. As the torque (for rotations) is related to a force (for translations) the minimum torque change model is related to the minimum force change model or minimum jerk model (compare [FH85] in the previous paragraph). Here the two studies agree but in the case of the reference frame/space (e. g. joint space or Cartesian space) the clearly differ.

The assumption that humans use movements according to the minimum jerk model is used by Bratt et. al. for compensation of the delay in teleoperation [BSC07]. The set task for a human is to catch a thrown ball via a teleoperation setup. Based on this movement model the prediction of the catching position $\approx 200\text{ ms}$ before catching is possible. This is a considerable part of the system's overall lag. Still the human reaction time is a limiting factor for successful catches [BSC07, pg. 2716].

Fligge et. al. examines human catching in three dimensions [FMVDS12]. The whole movement is seen as an initial fast movement and slow fine adjustments [FMVDS12, pg. 581]. The models, that result in similar motion patterns, are discussed. These are [FMVDS12, pg. 581f]:

- minimum torque change model (joint space)
- minimum variance model
- minimum jerk model (Cartesian space)

Due to the similar prediction based on these three models and the low computational expenses for the last model the minimum jerk model is used to model and predict human movements. The experiment is using 3D catching with a catapult, that throws from random positions with random velocities [FMVDS12, pg. 582]. A standing position with the right hand on the leg is used as starting position and no movement (walking) or bending of knees is allowed. The ball is caught when the velocity peak occurs. The ball velocity is matched only partially. [FMVDS12, pg. 583]. The minimum jerk model is able to explain 74 % of the trajectories. The point of minimum velocity is seen as a very characteristic attribute of the movement and is considered to divide the movement into two parts. Regarding the classification of these movements as predictive or prospective, the result of the research work is that the first (large) movement is predictive and that the following small movement is either prospective or predictive. The overall strategy could thus be to move to the point of minimum velocity as fast as possible and, starting from there, try to match the velocity of the ball or do the interception point prediction initially and update this information during the flight of the ball.

Complementary to the previous paragraph Okuda states that even the small movement is composed of a rough and a precise control state [OTI⁺09, pg. 3058]. Findings by Novak et. al. show that the primary movement and the submovement might overlap [NMH02, pg. 363]. Cesqui et. al. [CdPL12, pg. 9f] show that the number and timely overlapping of submovements vary even when the same individual catches variously thrown balls. This composition of submovements is seen to be necessary in order to fulfill the task of ball catching.

In experiments by Mazyn et al. [MSML07] the influence of lightning on human catching is analyzed. Test subjects have to press a button until they initiate the catching movement. After a number of trial runs, all lights are switched off within 3 ms after the subject released the button.

These experiments show that the additional submovements usually observed do not exist under this circumstance [MSML07, pg. 67]. One has to note that subjects delayed catching movement in order to gather more information about the flight thus leading towards significant faster catching movements with different velocity profiles.

2.4 Promising Solutions

Recapitulating the technical and biological research work the foundation for the work here is built. The promising solutions in the areas of the vision system, trajectory prediction, and catching movement are discussed in the following paragraphs.

Vision System

In the related technical work, the majority of implementation use a stereo vision system. This is based on the relatively common usability and the extraction of 3D data with thoroughly tested algorithms for image segmentation and shape recognition. The also holds true for the biological archetypes where most predators like leopards or eagles are equipped with two eyes and thus use a stereoscopic view. In terms of technical systems, the depth information is acquired based on stereo triangulation while a variety of cues for obtaining the depth information might be used for biological archetypes. A technical system with unsynchronized stereo and more extensive post processing of the data has been shown by Linderoth [LRÅJ10] but the technical implementation shows a number of disadvantages. For the research work in this thesis the usage of a stereo vision system is the most promising foundation for the further processing steps.

Regarding the first processing steps of the visual information most related work use specific algorithms to gain higher level information of the acquired scene (by example Hough transformation for circle detection when catching balls). This principle is in line with the research work on human processing where an abstraction in the very first layers of neuronal processing is assumed. Further processing is based on this abstracted information.

Trajectory Prediction

Regarding the field of trajectory prediction, nearly only analytical approaches for the prediction of the thrown object's flight trajectory are used. Usually, these approaches are based on a variant of the Kalman filter for parameter estimation followed by the calculation of the flight trajectory based on an iterative model considering the flight dynamics (usually gravity and air drag in a nonlinear way). The highest achieved catching rate of these approaches is in the range of 80 % and in a significant part of the research work the accuracy of the trajectory prediction is one of the reasons given for catching failures. Regarding the biological archetype and the processes involved in human catching only a little knowledge, relevant for a technical implementation, is available. The tight coupling of visual perception and time is proven. Additionally direct feedback from motor commands to the vision system is assumed to be existing. These attributes of the biological system can only be modeled with a large degree of assumptions because the research work in this area is not aiming for technical systems and thus delivers only marginal amounts of information.

For the further work here the existence of internal representations of previously perceived trajectories is assumed to be existing. This experience is used to predict the path of any thrown object. The disadvantage of dealing with entirely new circumstances like shown in the cat experiment (in Section 2.3.2) will thus be accepted. Strategies for dealing with this disadvantage will be given in the following chapter.

Catching Movement

Most of the research work targeting technical implementations for object catching simply position the catching device in the predicted interception point. This is a simple option that puts mechanical stress on both the catching device and the thrown object. The two approaches dealing with soft catching either use (limited) Cartesian control or accuracy throwing in order to avoid the requirement of prediction. Here a general approach to soft catching is the goal. In terms of research on human motion/catching, two phases are present. The first phase is a predictive movement with a shape explainable by the minimum jerk model. The second movement is not clearly classified but requires the movement of the field of view of the vision sensor towards or even into the catching position. Research work here shows that good catchers/batters are able to keep the ball in constant eye position based on the head movements. In terms of technical systems, this requires a movable vision system and increases the complexity of the position detection due to the additional movement and the required calibration.

Here a soft catching movement based on the kinematics of the robot's joints will be used. In terms of movement, the predictive movement to an initial point will be used either but due to the dynamic limitations of the robot the length of this movement might be of limited extend. The aspect of object tracking with an active vision system puts significant constraints on the camera's positioning and increases the computation demands and thus will be used optionally.

3 PROPOSAL

Transportation of objects based on throwing and catching has mainly been approached with modeling of the physical event and applying the mathematical solutions to the prediction problem. Equations for flight trajectories have been derived and algorithms for image processing developed. While the later cannot be evaded due to the optical sensors and their data representation a more bio-inspired approach to target the challenges will be proposed here. The approach models biological data acquisition and processing on a particular level of abstraction, the different platform (binary logic, personal computer/workstation compared to the archetype human, enormous neural network) causes this when realizing the proposed system.

A thrown tennis ball's trajectory is the result of a highly dynamic process that has a large number of influencing factors like the actual air density (temperature related), air streams in the surroundings, variation of initial launching parameters of the ball, differences in tennis ball's mass and diameter, differences in a tennis ball's felt and even more. It is obvious that predicting a tennis balls trajectory is a challenging task that demands detailed information about these factors. On contrary to technical systems that are based on physical models of the flight, the human archetype masters the task of predicting a tennis ball's trajectory extremely well. This prediction does not require any information about the physical background and is only based on the experience of similar situation previously. The result is that a human has to experience an adequate number of throws with high dedication for obtaining this capability at expert level [EKTR93] [Col08]. Still, basically, each child is able to catch a moderately fast thrown tennis ball. Extraordinary skills in this domain are performed by professional tennis players that return tennis balls with a velocity of up to 250 km/h [6]. An enormous amount of practice is necessary to master this challenge.

Overall System

For solving the challenge of catching a thrown object processes of data acquisition (mainly visual), trajectory prediction (experience based) and actuator control (for the catching arm) is necessary. Having a look at the way information is handled in the biological archetype (compare Section 2.3.2) offers a number of improvements over the state of the art (compare Section 2) approaches.

Consistent catching of a thrown object depends largely on the accuracy of the trajectory prediction. In comparison to a static positioning of the gripper in the interception point, where only the spatial information is relevant, soft catching of the object with matched trajectories (alignment of the object's and the catching device's movement at the instant of contact) requires timely accurate prediction as well. The biological approach is not completely clear but work from Pratl [Pra06,

pg. 35] suggests that the movement of objects is experienced as a symbol related to the attributes of the movement (velocity in the example, but more general: velocity, direction). Experiencing more different movements leads to the more accurate prediction of future movements. This is in line with learning theory for sports or musicians (compare [Col08, pg. 58f]) where the absolute number of hours spent on training the activity in combination with a high degree of focus on practice is made responsible for the actually accomplished performance.

Here two different ways to achieve trajectory prediction based on previously experienced flight paths are introduced in the following sections. Due to the focus on this area and the more detailed description only a short introduction is given here.

Based on the sequence of position acquired via the camera system and the processing of the image data a holistic trajectory prediction is proposed. This prediction considers the whole sequence of positions (sequence of positions as one movement-symbol) and matches similar movement-symbols of previously experienced throws to predict the future trajectory. The matching of similar movement-symbols can either be just the matching of the most similar individual symbol or the combination of a set of most similar symbols¹. This approach is introduced in Section 3.2.1), influences of parameters are analyzed in Section 5.1.2.1 and achievable prediction results are compared in Section 5.1.3).

An option here is to limit the number of considered positions of the measured trajectory for the matching (usage of the "recent history" of the movement) and match the most similar symbols to this dynamic changing movement-symbol. This approach demands less calculation power compared to the previously described approach, offers advantages regarding changing external influences on the trajectory (e. g. air streams) but the prediction accuracy has to be evaluated as less information is used to predict the future flight path.

The second approach uses the position change or velocity as the fundamental information for predicting. Advantages are seen in the independence from the absolute position, and thus, the data should be more versatile and allow more accurate prediction. Search complexity is a topic here and a strategy to reduce it is discussed and the achievable prediction accuracy is examined in Section 5.1.2.2 and Section 5.1.2.3.

In addition to the bionic approaches to image acquisition and trajectory prediction also, the task of movement planning for catching is done on this base. The movement of a human for catching an object softly is used to derive a strategy for catching an object softly automatically. The basic strategy will be presented in Section 3.3 while the practical implementation will follow in the next chapter (Section 4.5).

Efficient handling of the experience dataset is necessary for both approaches in order to fulfill the timely requirements of the overall system. The methods used to achieve this for the practical experiments, and the implementation are presented in Section 4.5.

3.1 Image Acquisition

The human eye has an enormous number of sensors for retrieving visual information (color and grayscale) with a variable resolution depending on the field of view [KO04, pg. 2]. Another attribute of the human eye is that the visual information retrieval is only possible when the

¹The term "most similar symbol" will be defined in the related section.

eye does not move (fixations) and that visual information cannot be retrieved during movements (saccades) [KO04, pg. 2]. During tracking of a moving object fixations and saccades are alternately used to follow the object's trajectory, keeping the object in the visual area with the highest density of light-sensitive cells [KO04, pg. 2]. This results in an optimized usage of the visual attributes of the eye.

For resembling this for technical systems, different approaches can be used. The system that resembles the biological archetype most accurate is a system of moving cameras with a resolution that is depending on the distance from the optical axis of the camera or with a small field of view. The downside of such a system is complex control of the movement and the need for consideration of the movement for the further processing of the image information.

Another possibility to resemble the visual sensor system is to use a camera with a field of view covering the complete scene and retrieving only the information from the relevant area of the image. State of the art cameras supporting AoI (Areas of Interest) provide this functionality. Specification of the relevant area can be done during runtime but depending on the camera's properties changing the area is not executed immediately on the next acquired image but might take more time depending on the camera's frame rate. In general, a short time prediction for the object's movement is necessary to follow the object. A system actively adapting the AoI's size and position depending on the motion history of the tracked object can be considered a good resemble of the biological archetype.

Overall the system allows to improve the prediction accuracy while keeping the computational costs at a reasonable level. The influence of a vision system's resolution and frame rate on the prediction accuracy has been shown in [Pon09, pg. 62ff]. A desirable high frame rate in combination with a high resolution of the vision system comes in hand with a high bandwidth camera interface and thus high cost. In order to keep the costs for the prediction system low an AoI for each camera is used. Based on the information about an initial launching position or the predicted flight trajectory only a small area of the whole image can be read out at the camera resulting in an increased frame rate. The reduced amount of information is beneficial also for the following step of object detection and pose estimation. A main requirement for this approach is a reliable prediction of the object's trajectory and the limitation to a single object that shall be tracked. If multiple objects are tracked based on this approach current cameras also support multi-AoI or the workaround, that area covering all AoI has to be read out and transferred from the camera.

For the biological archetype, this results in a focus of attention on the main object that shall be observed. The environment within the visual field of view is still perceived, but the importance of changes in this outer area in the field of view is reduced. Still significant changes in the area with reduced focus can lead to a shift in the focus of attention. In terms of a technical system based on the AoI, this is not possible. The reason for the downside is the technical realization where either pixels are read out or not. In comparison the human eye offers to change the resolution depending on the area within the field of view [KO04, pg. 2]. For a more accurate resembling of this function image sensors with an adjustable resolution for different areas (similar to the available binning feature for the whole transmitted image) are required. A sensor with a comparable capability has been proposed and implemented [OEG13, pg. 327] where the main focus is to reduce the overall bandwidth by compressing the image on the sensor already with minimal quality reduction. The same principle could be used for a chip, with a set AoI, for the area outside the AoI in order to allow to perceive the important data without any compression or loss in quality but still receive the surrounding area with lower quality. Especially environments where a change in the image

(e. g. surveillance applications or other security related tasks) could largely benefit from such an image sensor.

Determining the and adapting the optimal size of AoI as well as adapting the position of the area is another challenging task. In the environment of transport by throwing the initial position (origin of the object) can be considered as known. This allows to set up the initial position of the AoI to the origin position of the object. The size of the AoI in this area depends on the physical size of the object and the distance from the camera. The equation to derive the size of the AoI in pixels (l_{AoI}) depending on the object's size l_{obj} , the focal length of the camera in pixels f and the distance of the object to the cameras principal point d_{camobj} is given in Equation 3.1.

$$l_{AoI} = \frac{l_{obj}}{d_{camobj}} * f \quad (3.1)$$

This equation limits the size of the AoI exactly to (expected) size of the object in the image and thus does not have any tolerances for deviations in prediction or other error sources. Thus an additional factor k_{AoI} is introduced and the equation is modified to Equation 3.2. The magnitude of k depends on the maximum error of the (short term) prediction. Another factor is the reduced amount of visual information, required to be processed, based on the reduction of image size. The stress field of fault tolerance, the amount of information to be neglected and the increase in the frame rate of the vision system

$$l_{AoI} = k * \frac{l_{obj}}{d_{camobj}} * f \quad (3.2)$$

The open issue of adapting the AoI's position requires prediction of the object's movement. The movement in space is projected to a relocation in the camera's frame. In order to predict the relocation in the camera frame, the movement in space has to be predicted first and then projected to the camera's frame.

The first step of motion prediction (in 2D image space) is done with a simple linear model due to the high frame rate and the low computational expenses of this method. Assuming that no knowledge about the movement is known except the initial position, it is obvious that an initial value for the movement of the AoI is needed. These values are set based on manual measurements.

Limitations of the cameras in terms of hardware support for this feature lead to a more simple approach for the experiments where the size of the AoI is fixed at *768 by 2048 pixels* and further the image is cut down in software to *300 by 300 pixels* which corresponds to the maximum size needed. The big AoI is needed to cover the whole flight of the object and because the camera's driver does not assure that changes of the AoI-position are already considered for the next acquired image. Reducing the (software-)AoI's size further would lead to reduced computational demands but further increasing the frame rate of the camera system is not possible due to a fixed data acquisition rate.

3.2 Bionic Associative Memory

Work from Pratl [Pra06, pg. 35] suggests that the experience of movement is based on movement symbols and that prediction of movements is only possible if similar movements have already

been experienced. Work from Colvin [Col08, pg. 58f] suggests that the quality a human being is able to perform a task mainly depends on the hours of deliberate practice. This is in line with brain research [Pra06, pg. 12] that shows that perception is a mixture of sensory information and knowledge. Unknown objects have to be processed in detail. Also, research from language perception approves the assumption that the process of perception is constantly accompanied by association of knowledge [Col08, pg. 87ff] [Pra06, pg. 18].

Transferring this assumption to the task of catching an object raises the demand for a fast accessible experience of trajectories. Furthermore, a mechanism to associate these trajectories to a currently ongoing movement is required. The fact that absolutely similar movements never happen stands to reason that at least a certain level of generalization is happening. Thus, a continuous scale of similarity is suggested instead of a binary measure categorizing two properties as similar or different. In addition, the capability of humans to combine experience and use this to solve current problems suggests that the internal processing makes use of more than just the most similar experience. In coordination-training (e. g. in sports) the opinion that a wider range of experience allows to adapt to new circumstances faster is standard [Ph.13, pg. 210] [DGAB03].

An important factor here mentioned by Colvin [Col08, pg. 65ff] is the intensity the human being is working on the topic. The word "*deliberate*" is used here to express a higher degree of (intrinsic) motivation. This higher degree is seen as a continuous process of feedback and self-evaluation of the learner. Thus, this aspect is also required in a technical implementation that mimics this human processing model.

The basic idea of k-nearest neighbors (compare Section 2.2.2.3) fulfills the prior mentioned attributes only partially. Especially the combination and application to new circumstances is limited. Furthermore, the evaluation and feedback process is not designated.

In the following two subsections, two modifications of k-nearest neighbors to incorporate the main functions of the human movement experience, generalization, and prediction processing are introduced and discussed. The first variant is introducing a working-set of experience and a feedback and evaluation functionality to the classical approach, consequently targeting the reduction of association time. The second variant is extending this furthermore towards a system that allows to generalize and combine experience on a higher level. Thus allowing to extend the field of application even towards new, previously unexperienced, processes.

3.2.1 Holistic Associative Memory

Nearest neighbor algorithm (compare Section 2.2.2.3) is simple, efficient and effective in the fields of classification and pattern or object recognition. The main challenges of this algorithm are the computational complexity and memory requirements [BV10]. Here these challenges are considered with respect to the application of this approach to the task of transport by throwing, more accurately to the task of trajectory prediction. Biologically inspired approaches solving these challenges are introduced and discussed.

Figure 3.1 outlines a dataset with a database with three trajectories (green, blue, orange) and a current trajectory (red). For the task of trajectory prediction the future ball positions (circles) of the red trajectory have to be predicted. A number of different approaches exist to target the problem of big datasets. Here the approach mimicry human association is presented.

In term of similarity measures, the Euclidean distance (further on meant by the word distance) is the chosen unit as the trajectories are represented in 3D data Furthermore the combination

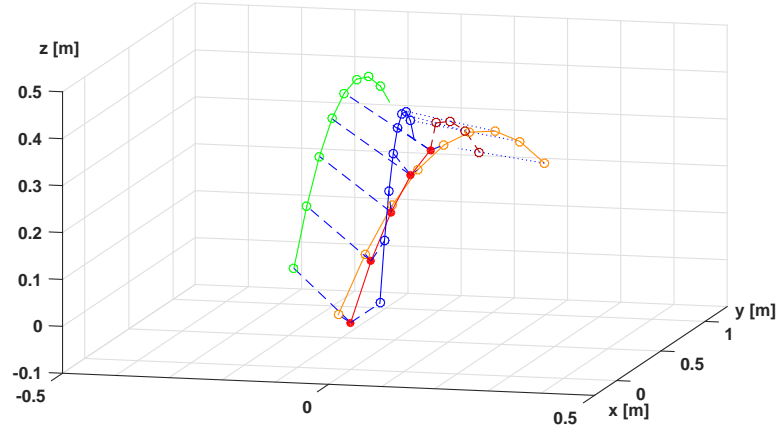


Figure 3.1: Basics of k-nearest neighbor, 3 throws from database (green, blue, orange) and a new one (red); solid lines for illustration, data points are circles, dashed lines connect related points for similarity determinations

of multiple similar datasets is used as humans tend to combine information from the memory as well. Thus, k-nearest neighbors are used instead of only the nearest neighbor. The strategy for the combination of these similar trajectories based on the similarity scale (weighted nearest neighbor) is considered suboptimal in environments with infinite sample case [BJ78]. Still the advantages of this approach are seen regarding the release of the limitation of equal weights in k-nearest neighbors, the transformation to a global algorithm and the wider range of used samples [BV10, pg. 303].

Memory Managements Principles

The combination of a wider range of samples raises the computational expenses. This is solved by the introduction of a small working set that is a part of the complete experience database. A illustration is given in Figure 3.2.

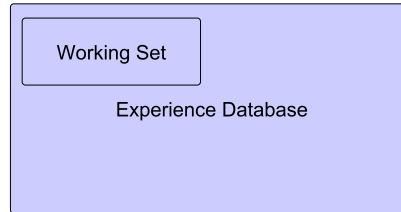


Figure 3.2: Whole experience database and small working set;

The small working set is used to improve the performance of the nearest-neighbor search in terms of search time in case it is necessary.

For the task of prediction, only the small working set is considered. An illustration of the prediction process is given in Figure 3.3. The number of acquired positions from a current velocity is compared with the related positions of the trajectories in the working set. Based on the k most similar trajectories (sum of distances) the future path is predicted. The inverse of the sum of distances is used to weight the k most similar trajectories in order to derive the prediction.

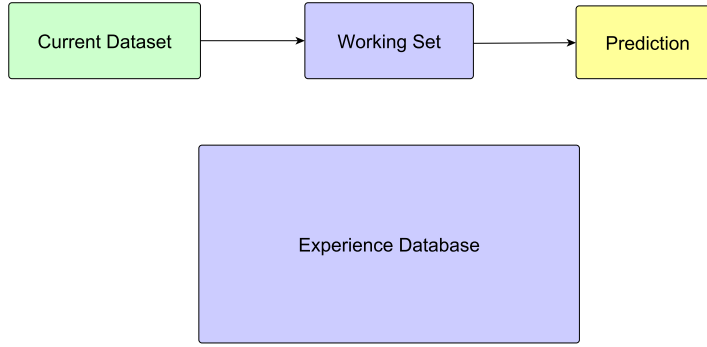


Figure 3.3: Prediction of the current trajectory based on the working set;

Due to the performance requirements the working set contains only a limited number of trajectories out of the whole experience database.

The process illustrated in Figure 3.3 is executed each time a new measurement from the position determination system is acquired. After the whole trajectory is acquired the current working set is updated based on an exhaustive search for similar paths in the whole experience database (compare Figure 3.4). Thus, during the time when the timing requirements are loosened up (between two throwing transactions, the time-consuming evaluation of the whole experience database is done. The reason for this behavior is the assumption that the most recent trajectory is a good representative of the current state of the environment. The size of the working set is logically adapted to the performance requirements and the used hardware for predicting.

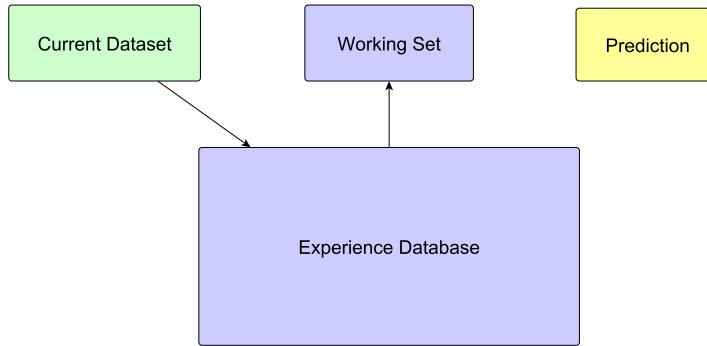


Figure 3.4: Whole experience database and small working set;

The small working set is used to improve the performance of the nearest-neighbor search in terms of search time.

The approach presented above can, in addition, be extended to an approach where the size of the working set is managed more dynamically. Instead of using a fixed size of the working set, the trajectories in the experience database obtain an additional parameter that specifies how frequently this trajectory was used as the nearest neighbor during the recent history. The recent history is equal to a number of throws before the current one. By sorting the experience database during the time between two throws and accessing it in a descending usage order the most similar trajectories to the most recent throws will be accessed first. Again the assumption that the most current paths represent the most probable environmental circumstances is used. Furthermore, because the most relevant trajectories are compared foremost, this approach can be used to stop the search for similar trajectories based on the timing constraints of the whole system.

Further Challenges

A remaining challenge of the trajectory prediction based on the above-mentioned approach is the prediction in the final phase of the flight and the dependency on the initial position of a trajectory. The first challenge can be illustrated by considering that the catching movement of the robot is interfering with the image acquisition during the final flight phase. In case this happens all trajectories would end right before the objects enters the catching area, and thus the prediction of the object's path in the relevant area could not be done at all.

The second challenge of the dependency on the initial position can be illustrated by reconsidering Figure 3.1. Even if the green trajectory was the perfect resemble of the red trajectory, the fact that the initial point (or starting point) of the green trajectory has a big distance to the starting point of the red trajectory would hinder the usage for prediction. The problem behind this is that the absolute position is considered here. A more general application of acquired knowledge to the current task of prediction is desirable.

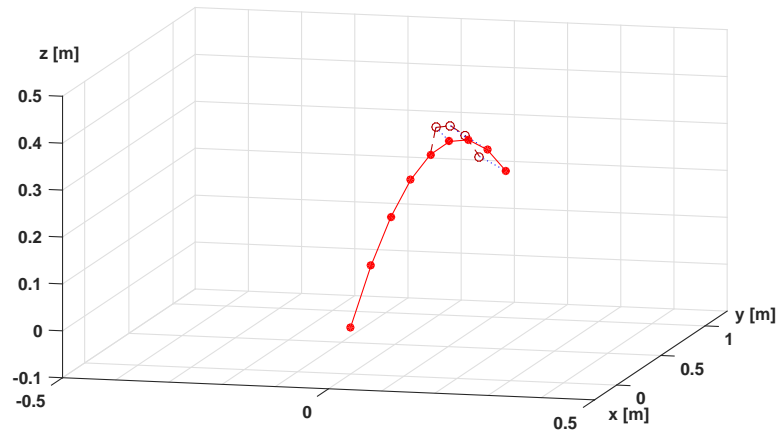


Figure 3.5: Result and prediction error of k-nearest neighbor prediction; real flight light red, predicted flight dark red

Finally, also the deviation in the final part of the flight path is a problem as the position information is considered for the prediction. Even using more trajectories (k) for the prediction shows this problem as it is illustrated in Figure 3.5.

3.2.2 Progressive Associative Memory

Targeting the open challenges from the holistic associative memory the progressive associative memory puts the task of prediction into another perspective. While positions are considered as the main information in the previous subsection here, the velocity is the main information for prediction. The trajectory prediction is done based on the current (measured) position of the object and the previously acquired velocity information from the experience database. Figure 3.6 gives an overview over the basic principle. The fundamental problem is presented in Figure 3.6(a). For a new trajectory (red) the future positions of the object shall be predicted based on a set of available trajectories (orange, green, blue). Instead of processing the prediction step right on this information the trajectories are transformed into velocities (Figure 3.6(b)). These velocities are the base for the similarity comparison. Based on the last acquired position of the object and

the most similar trajectory in terms of velocities (Figure 3.6(c)) the future trajectory is predicted (Figure 3.6(d)).

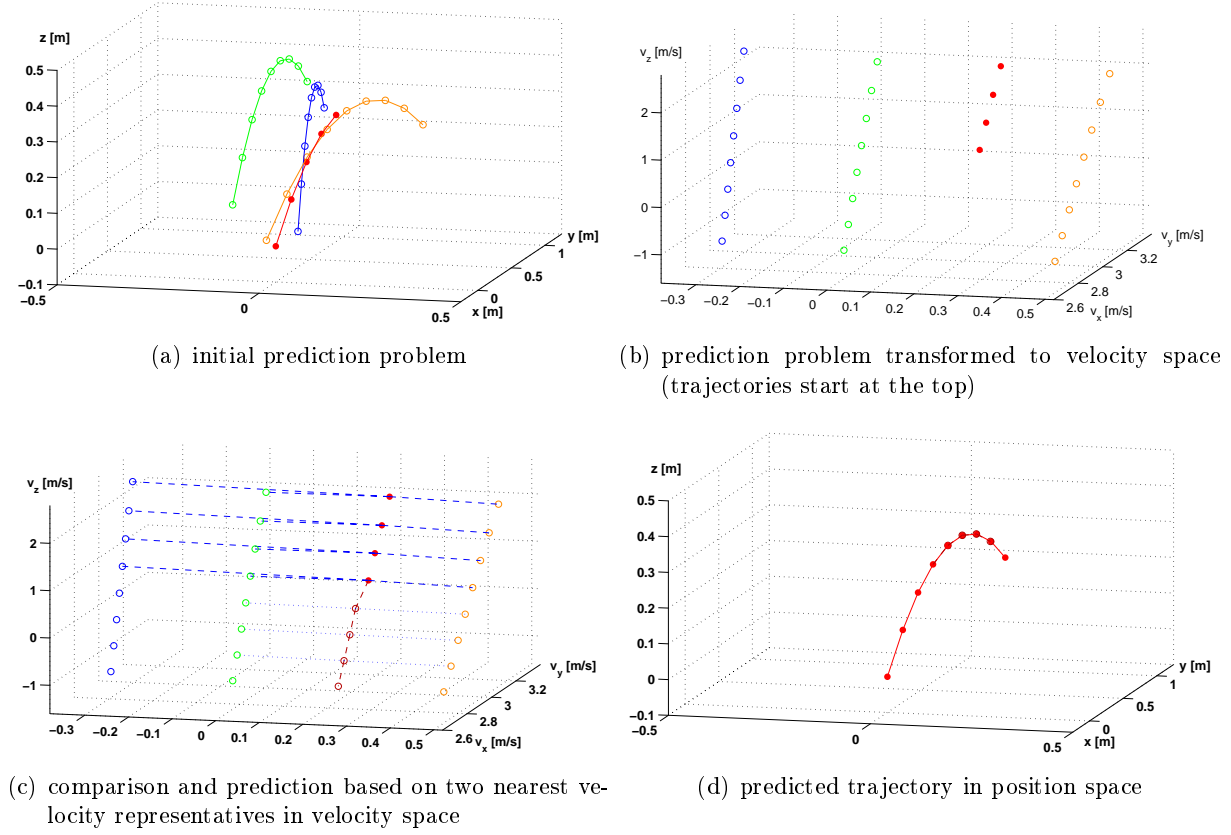


Figure 3.6: Basic principle of progressive prediction with the transformation to velocity space, association and prediction in velocity space and transformation back to the position space

In comparison to the prediction based on the position information (discussed in the prior subsection), the dependency on the initial position is avoided. In the illustration in Figure 3.6 the green trajectory is used for prediction despite being further away from the red trajectory in terms of positions. Concerning velocity this trajectory is very similar to the red path that can be seen in Figure 3.6b. Thus, any similar trajectory (in terms of velocity) can be used for prediction allowing to apply an overall smaller dataset to a wider range of prediction tasks.

Spatial Discretization

In a general case where different trajectories are stored in the experience database, the second quarter of a previously experienced flight path could be a good match the final quarter of the current trajectory. This causes an increased search complexity (both for the holistic as well as the progressive). Thus, the current velocities have to be compared to all previously experienced ones. In order to allow a generalized prediction (prediction of the current path based on any phase² of other trajectories), the current velocity (or a number of recent velocity measurements) have to be compared to the whole experience database. This raises the computational demands.

²Phase here in the means of time since start.

A representation of the velocity information that allows to access similar velocities efficiently is desirable in this regards.

In order to tackle this challenge a spatial discretization is used to minimize the search space and allow efficient comparison of trajectories (velocities of trajectories). An illustration of the process is given in Figure 3.7(a). Each velocity sample is a representative of a voxel, velocity class or movement symbol. The related figure in the continuous space is Figure 3.6(b). In contrast to quantization, where analog signals are converted to digital representations, the spatial discretization uses the already quantized data and decreases the resolution (increases the quantization error due to the increased distance between the values). This can be compared to moving from float point representation to fixed point representation of a number. The uncertainty introduced due to discretization rises for the sake of a faster processing time. The increased size in discretization steps is most obviously shown in the blue dataset. The x-component of the velocity is decreasing continuously from the top of the diagram to the bottom. In continuous space (Figure 3.6(b)) this is resembled well while the discretized representation in Figure 3.7(a) shows a jump between the third and the fourth measured velocity from the top of the dataset. This jump is caused by x-component of the velocity falling into the next lower discretization interval.

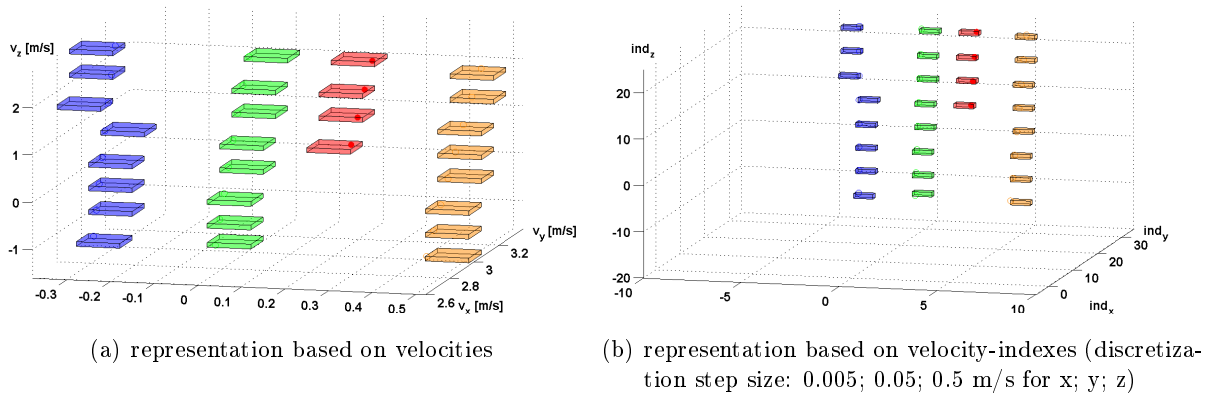


Figure 3.7: Spatial discretized velocities;

The blue, green, red and orange boxes are representative categories of the related velocities of the trajectories.

The continuous velocities are shown with circles. The dataset is the same as used in Figure 3.7(b)

The process of spatial discretization is similar to what is done in digitalizing audio data or for mapping for autonomous robots. In terms of robotic mapping either occupancy grid maps [TBF05, pg. 281ff] or maps based on normal distribution transform [BS03] are illustrative examples for this approach. Basically, an error due to discretization is caused by this procedure, but the advantages overbalance the intentionally introduced error. The size of the discretization steps is balancing the overall accuracy versus the size of search space. Smaller steps result in more accurate position representation but by example half the step size in each dimension causes a search space with eight times size. A relation between the position detection's accuracy (unavoidable error caused by the vision system and image processing algorithms) and the optimal size of the discretization steps is assumed to exist. Section 4.2.2.1 deals with the determination of the tracking system's accuracy and in Section 5.1.2.3 the simulation results for determination of the discretization step size's influence on the prediction result is given.

Searching in a database containing this spatially discretized velocity information for the same velocity is based only on the calculation of the indexes of the current velocity. Table 3.1 gives

the related indexes for the velocities given in Figure 3.7(b). If, for example, a similar velocity to the last measured velocity of the red trajectory is searched for, the database containing the blue, green and orange trajectory would not yield any trajectory with the same velocity. Extending the search to the neighbor velocity-representatives (all data fields with indexes ± 1) would yield the second-last velocity of the red trajectory (compare the indexes of the 4th and 3rd sample of red trajectory in Table 3.1). Opening up the search to data field with indexes ± 2 delivers the three final velocities of the red trajectory (compare the indexes of the 4th and 3rd/2nd sample of red trajectory in Table 3.1) and five velocities (compare the indexes of the 4th sample of the red trajectory with the second to sixth sample of the orange trajectory in Table 3.1) of the orange one. Thus, the first association with an existing trajectory can be made. Finding the most similar velocity is done based on the comparison of the indexes, to be more precise: picking the velocity of a foreign trajectory with the least distance. Subsequently, the 4th sample of the orange trajectory is the most similar to the 4th sample of the red trajectory with a distance of two discretization steps in the x-direction.

sample	blue			green			orange			red		
	<i>x</i>	<i>y</i>	<i>z</i>	<i>y</i>	<i>x</i>	<i>z</i>	<i>y</i>	<i>x</i>	<i>z</i>	<i>y</i>	<i>x</i>	<i>z</i>
1	-4	64	4	0	63	4	5	63	4	2	63	4
2	-4	63	3	0	62	3	5	62	3	2	62	3
3	-4	63	2	0	61	2	5	61	2	2	62	2
4	-3	62	1	0	60	1	5	60	1	2	61	1
5	-3	60	0	0	60	0	5	59	0	-	-	-
6	-3	60	-1	0	59	-1	5	59	-1	-	-	-
7	-3	59	-2	0	58	-2	5	58	-2	-	-	-
8	-3	59	-3	0	57	-2	5	57	-3	-	-	-

Table 3.1: Overview of the index representation of the trajectories given in Figure 3.7(b)

In order to keep the link between the individual velocities of a single trajectory the storage of the velocities has to keep the information about the chronologically prior measured velocity. In addition, for prediction, the information about the chronologically following velocity is required as well. Storing the trajectories as a doubly linked list of velocities allow to access the whole information about the path in both directions.

Reducing Quantization Influence on Reconstruction/Prediction

The spatial discretization of the data can be done in the regime of velocities as outlined in Figure 3.7(a) or on the position data shown in Figure 3.6(a) and afterward being transformed to the velocity regime. The benefits and downsides of each processing sequence will be discussed here briefly.

As mentioned before discretization introduces an error. To keep this error source's influence minimal the logical approach is to do the discretization as late as possible in the processing

steps in order to keep the minimum errors as long as possible³. This argumentation clearly is preferring to calculate the velocities prior to discretization (Figure 3.7). Figure 3.8 shows a comparison between both approaches when using the similar related discretization step size⁴ for both processing possibilities.

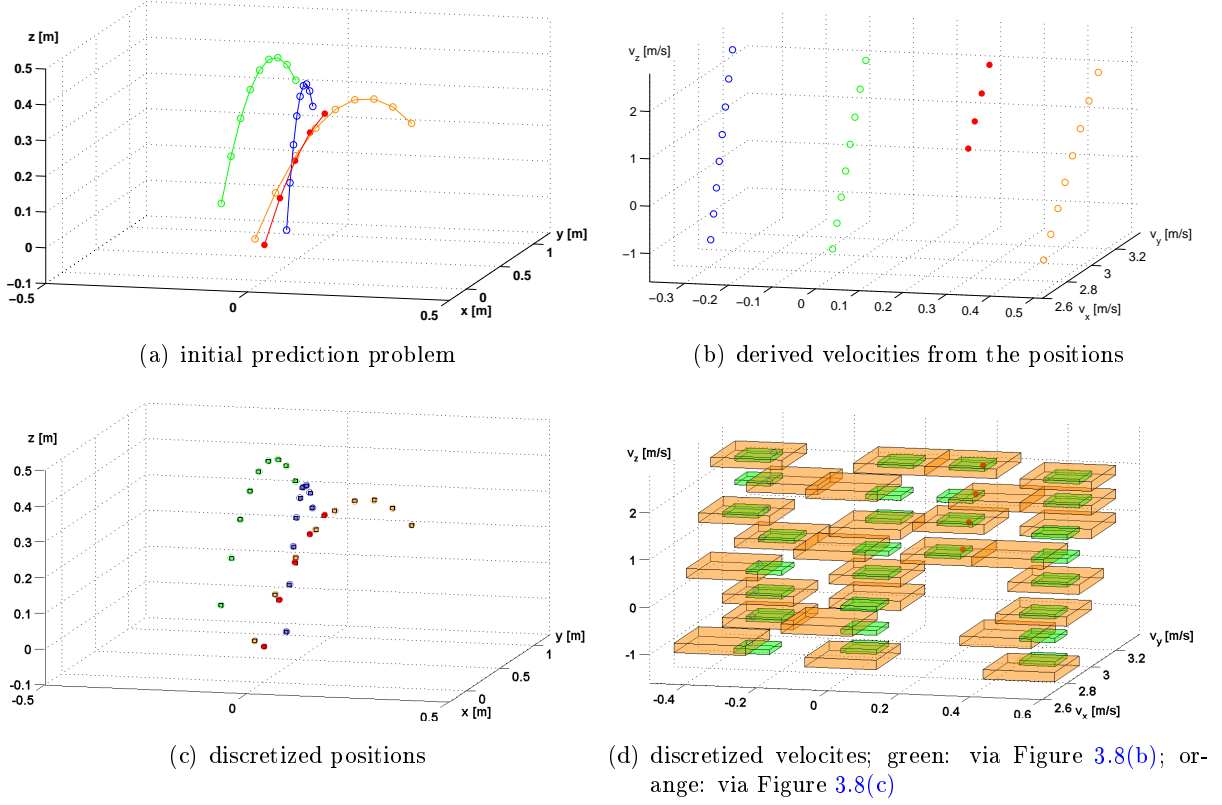


Figure 3.8: Comparison of initial velocity calculation and subsequent discretization and immediate discretization and subsequent velocity calculation;
Especially for reconstruction/prediction of the trajectory the immediate discretization offers advantages reducing the effects of error accumulation

The results of both processing orders yields in two different data sets in the spatially discretized velocity database (illustrated in Figure 3.8(d)). This is caused by the influences of the discretization errors. While the discretization error is the upper and lower limit for each of the discretized velocities in Figure 3.8(b) and can be accumulated during the reconstruction/prediction process, the same limits apply to the velocities based on the discretized positions even for any combination of the information. Thus, the seconds approach, to immediately do the discretization and calculate the velocities based on this information is preferable and used for the prediction.

The different size of the velocity representing cubes in Figure 3.8(d) originate from the different processing sequences. The smaller cubes are derived from immediate velocity calculation followed

³Mind the difference between quantization and the additional discretization here. Quantization is the sampling of analog signals and translation into digital signals. The discretization is the reduction of the resolution of the digital signal. Doing quantization as early as possible in the processing steps, similar to what is done in current telecommunication engineering, is advantageous also here

⁴similar related discretization step size means that the discretization size is transformed from position space to velocity space in the same manner as positions are transformed to velocities, thus offering the same discretization step size

by discretization while the bigger ones occur in the other way. The double size is caused by the difference of two discrete values, where the real value can be up to one discretization step bigger or smaller. The resulting size is two steps. In the other processing sequence, the size is limited to half a step larger or smaller resulting in a size of one discretization step.

One might argue that this causes the comparison of apples and oranges but the relevant information, the density or granularity of the velocity information database, is the same for both processing sequences. This can clearly be seen by half the cube length steps between the bigger cubes in Figure 3.8(d): While the representation size is larger, the distance between individual data points is still the same. Presentation of the combined information in Figure 3.8(d) in two individual figures is done in Figure 3.9.

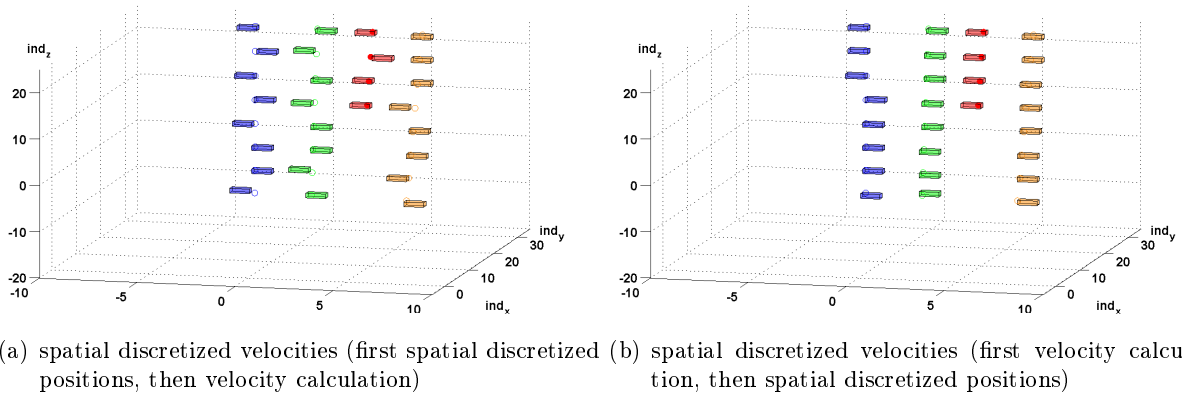


Figure 3.9: Comparison of initial velocity calculation and subsequent discretization and immediate discretization and subsequent velocity calculation;

Especially for reconstruction/prediction of the trajectory the immediate discretization (Figure 3.9(a)) offers advantages reducing the effects of error accumulation; for comparison the continues value velocities are also shown in the diagrams as circles

Using the prior discretized data for velocity calculation allows combining the advantages of generalization offered by the usage of velocity information but still keeping the (integrative position) information with a deterministic and limited error margin.

Recent History Consideration

Moving the starting point of the search from the starting point of the trajectory to the last acquired velocity allows to define a number of recent velocity-samples that are used for similarity calculations. By example only the last three velocities of the red trajectories (samples 2 to 4; the second to fourth measured position from the top of the red dataset in Figure 3.8(b) or Figure 3.8(d)) could be used. This allows relatively fast adoption to changes in the environment and the general usage of any recorded trajectory for prediction tasks of the current one (consider the discussion about the prediction of the final flight phase in the previous subsection). An elaboration on the suitable length of the recent history of prediction and the influence on the prediction accuracy is done in Section 5.1.2.3.

Real Time Performance Requirement

The usage of the velocity-based prediction allows to generalize trajectories to apply them for predicting current ones. By immediately discretizing the position information still the integrative information content of the trajectory is kept. This circumstance can be used to optimize the prediction by changing the number of points for comparison. Instead of comparing all velocity measurements in the recent history with the previously experienced ones, only a subset of the velocity measurements in the window for comparison can be used as the velocities are steady (results here are given in Section 5.1.2.3). This reduces the calculation expenses for comparison and allows a to compare the current trajectory to a wider range of path within the same timely envelope, possibly improving the prediction accuracy.

In order to fulfill the real-time requirement for the trajectory prediction, a strategy to deal with unsuccessful compares is needed. Here the chosen strategy is fall back to the previously associated trajectory (with an additional shift of the considered samples due to the timely progress) is used. A wide range of other strategies could be implemented, but they are the objective of further research.

3.3 Catching Movement Planning

Besides image acquisition and prediction (described in the previous sections of this chapter), the movement planning of the robot for successfully catching of the thrown object is the final part of the whole systems (neglecting the implementation details covered in Chapter 4). The task for the catching movement planning subsystem is to determine a suitable action for intercepting the currently thrown object based on predicted trajectory supplied by the prediction system. Main requirements for this movement are:

- Minimization of the impact energy and impact force during catching
- Adaptability to more recent and thus probably more accurate trajectory prediction data
- Mimicry human catching discussed in Section 2.3.3

The third requirement might seem unspecific but it can be seen as a result of the first two requirements. Human catching is adaptable (and thus reliable) and the impact force during catching is minimized in order to avoid damage (to the caught object and the hand/arm). Thus human catching is a very suitable archetype for fulfilling the two prior mentioned requirements, validate and optimize the system. These aspects will be discussed in the following paragraphs.

Minimization of Impact Energy

A main aspect of a successful transport by throwing and the generalization of the approach to mechanical sensitive objects is the reduction of the forces during throwing and catching. The mechanical strain on the thrown object during launching is limited because of the common initial velocity of the throwing device and the object and the inertia of the throwing device. On the other hand, while catching nearly all approaches discussed in Section 2.2.2 simply move the catching device into a position in line with the predicted trajectory of the object and thus expose the thrown object to a significant impact force.

Research from Hove [HS91] and Frank [Fra12] differs from most other approaches since the movement alignment of the catching device with the thrown object is of main interest. In the prior work the Cartesian position of the predicted ball's trajectory and the current data from the camera system is fed to the robot controller directly. Remarks about a special usage of the kinematic of the robot are not given. But considering the robot's maximum velocity in Cartesian mode of $2 \frac{m}{s}$ [HS91, pg. 382] it is assumed that the Cartesian control mode of the robot is limiting the maximum velocity of the catching device. The same is valid for the KUKA LWR 4+ robot used for the implementation of the transport-by-throwing system. The maximum Cartesian velocity equals $2 \frac{m}{s}$ [KUK11, pg. 277]. On the other hand, when using the joint position control mode the achievable Cartesian velocity of the end effector or catching device is significantly higher and in the same order of magnitude as the velocity of the thrown object.

In order to minimize the relative velocity between thrown object and catching device the following measures are taken:

- Launching parameters are chosen to cause the lowest magnitude of velocity in the catching area
- Kinematic of the robot is used to add up velocities of different joints in order to maximize end effector velocity

Optimal Launching The work of Frank [Fra12] shows how different launching parameters can yield in approaching the same circular trajectory of a possible catching device. The figure is

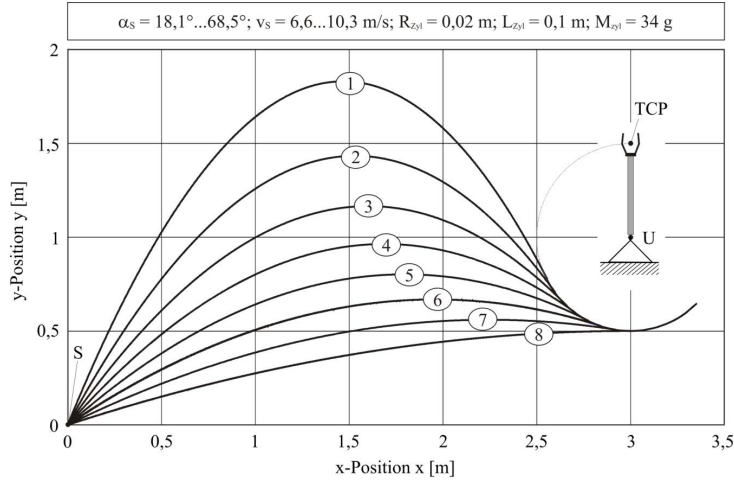


Figure 3.10: Trajectories of a thrown cylinder with a point tangential to the circular trajectory of the catching robot [Fra12, pg. 35]; S is the starting position, TCP the tool center point of the catching device;

Eight different combinations of velocity and launching angle are given to reach the circular trajectory of the TCP around the rotation center U

given here again in Figure 3.10 and illustrates that the circular motion path of the catching robot can be reached with different combinations of launching angles and velocities. An aspect that has not been covered is the magnitude of the velocity at the point where the object's trajectory is on the circular path of the catching device. This calculation will be done here. In order to minimize impact force/energy the flight trajectory with the minimum magnitude of velocity in the

interception area is preferable. In Figure 3.11 the simulation of a transport-by-throwing system is shown. The throwing device (TD) is located at the position $P_{TD} = (-2.5 / -1 / 0) \text{ m}$ and the

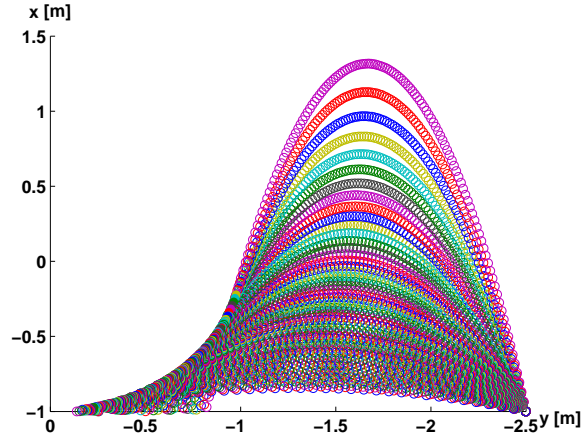


Figure 3.11: Visualization of different throwing trajectories to reach a point on a circular interception trajectory of the catching device

This simulation is based on the environment given above (position of throwing device $P_{TD} = (-2.5 / -1 / 0) \text{ m}$, main joint of catching device $P_{CD,J1} = (0 / 0 / 0) \text{ m}$, arm length $l_{arm} = 1 \text{ m}$, gravitational acceleration $g = 9.81 \frac{\text{m}}{\text{s}^2}$, mass of the tennis ball $m = 0.0577 \text{ kg}$, diameter of the tennis ball $d = 0.067 \text{ m}$, drag coefficient $C_D = 0.6$ and air density $\rho = 1.2041 \frac{\text{kg}}{\text{m}^3}$)

main joint (J1) of the catching device's arm is in $P_{CD,J1} = (0 / 0 / 0) \text{ m}$ with an arm length of 1 m . Trajectories with launching angles between 15° and 80° differing by 1° are shown that all have a point (interception point) where the trajectory of the ball is aligned with the trajectory of the throwing device. The required launching velocity is calculated based on an iterative model (compare Section 4.2.1) for the flight with the following parameters: gravitational acceleration $g = 9.81 \frac{\text{m}}{\text{s}^2}$, mass of the tennis ball $m = 0.0577 \text{ kg}$, diameter of the tennis ball $d = 0.067 \text{ m}$, drag coefficient $C_D = 0.6$ and air density $\rho = 1.2041 \frac{\text{kg}}{\text{m}^3}$. For each throwing angle the velocity is varied between $2 \frac{\text{m}}{\text{s}}$ and $8 \frac{\text{m}}{\text{s}}$ with an increment of $0.005 \frac{\text{m}}{\text{s}}$ and the best suitable velocity is taken. The criteria for this selection is the deviation of the trajectory in the catching area from the (circular) path of the catching device. For all tested throwing angles with the related velocity the distance between the circular trajectory of the throwing device and the (calculated) trajectory of the ball is smaller than 1.7 mm , thus, all paths pass within this distance to the circular trajectory of the catching device. While Figure 3.11 shows selected trajectories the essential information derived from this simulation, adapted to the geometrics of the final catching experiments is given in Figure 3.12. The magnitude of the velocities in the interception point is shown for different launching angles. The velocities decrease up to the point where the launching angle is equal to 51° and increase afterward. For the specific configuration of throwing device position and catching device location the minimum velocity in the interception area equals $4.55 \frac{\text{m}}{\text{s}}$. This configuration is also used for the simulation (compare Section 4.4) and implementation (compare Section 4.5).

Kinematic Consideration Here the work of Frank [Fra12] is used as a base. The approach from Hove [HS91] is not suitable since the usage of Cartesian control mode results in outsourcing of the motion planning to the robot controller causing the reduced maximum velocity. Frank, on the other hand, uses a simple kinematic with just one degree of freedom to realize "soft" catching. The

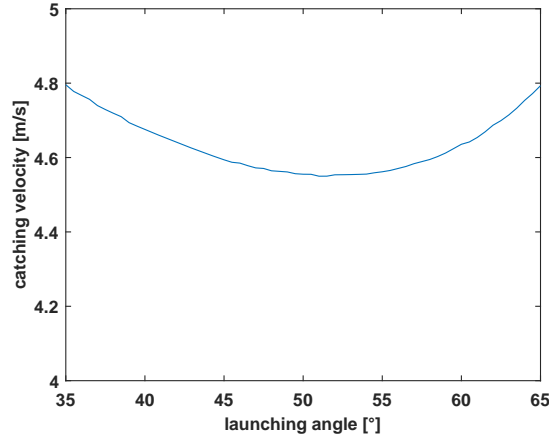


Figure 3.12: Visualization of the velocities in the interception point depending on the launching angle (compare Figure 3.11).

This simulation is based on the environment of the final catching experiments. The optimum throwing angle for a minimal velocity of the ball in the catching area is 51° and in the range of 46 to 56° only slightly higher velocities have to be handled

usage of such a simple kinematic is possible because of the compliant flight behavior of cylinders with small deviations to the right or left [Fra12, pg. 53].

Here the basic idea of this strategy will be generalized to more complex kinematics that also allow to catch objects with more variable flight trajectories or thrown with more deviations in the launching parameters. In the case used by Frank (compare Figure 3.10) the alignment of the thrown object's trajectory with the trajectory of the catching device is only possible at one point. This position is defined by same (3D) tangents on both trajectories. In case of a simple 1 DoF (Degree of Freedom) robot the path of the catching device is circular and the tangents are always 90° on the connection between the center of the joint (U in Figure 3.10) and the effective tool center point (TCP in Figure 3.10).

In case the model is generalized to a more complex kinematic like a 2 DoF main joint and an adjustable-length arm (length between U and TCP is variable) the point where the trajectory of the thrown object and the trajectory of the catching device (or TCP) is aligned is where the distance between the path and the center of the joint (U) is minimal⁵. This correspondence to a tangent of the catching device's trajectory that is in line with the thrown object's trajectory. An 3D illustration of this generalized model is given in Figure 3.13. The blue circles are a tennis ball's trajectory sampled with a simulated camera at 100 Hz , the red dot is the center of the 2 DoF main joint, the green dot is the interception position and the blue line between the red and the green dot is the arm with adjustable length. Mind that due to the scaling the angle between the trajectory and the arm does not look like 90° but this requirement is fulfilled.

A relatively simple system (3 DoF) is sufficient to allow soft catching with trajectory alignment in case the velocity of the thrown object can be reached. Additional DoF might not be used at all, might be used to realize one of the three required DoF (by example the adjustable-length arm) or might be used to increase the velocity of the catching device in case the main joint with 2 DoF is not able to realize the required velocity. In addition small and fast adjustment movements might

⁵One remark here: this distance must not be bigger than the maximum length of the adjustable-length arm. Otherwise, the object cannot be reached with the catching device.

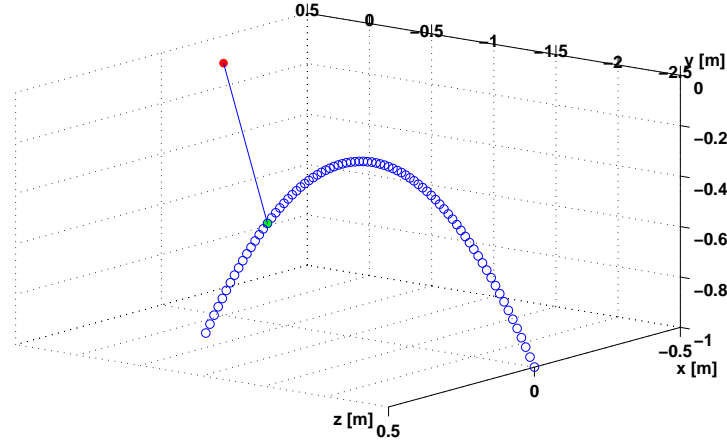


Figure 3.13: Visualization a simplified robotic arm for catching; red dot: robot origin position, green dot: interception position, blue line: robotic arm, blue circles: calculated ball positions

be realized with the additional joints. Whether this is reasonable depends on the relation between the sum of the mass moment inertia affected by the link and the maximum torque and must be evaluated separately for each robot/joint. The basic methodology for doing this is here while the particular application to the KUKA LWR 4+ robot is given in Section 4.5.

The calculation of the moment of inertia for simplified models of robot links and typical joint configurations is summarized in Table 3.2. The symbols used are for the solid cylinder: the moment of inertia J , the mass of the whole cylinder m , the radius of the cylinder r or length of the stab l ; These formulas will be used in Section 4.5 for determining maximum acceleration parameters of the KUKA LWR 4+.

Configuration (shape, rotation axis)	Equation for moment of inertia
solid cylinder, rotation along axis of symmetry	$J = \frac{1}{2} * m * r^2$
solid sphere, rotation along axis through center	$J = \frac{2}{5} * m * r^2$
solid stab, rotating along an axis in the flat end	$J = \frac{1}{3} * m * l^2$

Table 3.2: Formulas for calculation of mass of inertia of simplified robot links [8]; an example for joint 1 is the modeling of link 1 as a solid cylinder rotating along the axis of symmetry, link 2 to link 5 as cylinders rotating along an axis in the flat end and link 6 as a solid sphere rotating along an axis through the center (compare Figure 4.3 and further details in Section 4.3.3)

The simplified shapes are chosen to represent links of the KUKA LWR 4+. All links are modeled as solid cylinders. Depending on the relation between the considered joint and the maximum possible distance (resulting in the maximum possible moment of inertia) the formulas given in Table 3.2 allow to estimate the maximum angular acceleration for each joint based on the maximum torque specified. In addition, the parallel axis theorem (given in Equation 3.3) is used to consider links in distance to the joint. The used symbols are the moment of inertia J , the initial moment of inertia when rotating along an axis through the center of mass J_{cm} , the mass m and the distance between the center of mass and the new rotation axis r .

$$J = J_{cm} + m * r^2 \quad (3.3)$$

The relation between the maximum joint acceleration α_{max} , the maximum joint torque M_{max} and the moment of inertia J is given in Equation 3.4.

$$\alpha_{max} = \frac{M_{max}}{J} \quad (3.4)$$

Keeping the robot within the specified limits for torques is essential to keep it in operation since safety circuits disable the robot in case of a torque that is exceeding the specified maximum. The calculation of the maximum possible acceleration also allows to calculate the time required for each joint to reach the specified maximum velocity. The relation between maximum acceleration α_{max} , maximum velocity ω_{max} and the acceleration time to maximum velocity t_{vmax} is given in Equation 3.5.

$$t_{vmax} = \frac{\omega_{max}}{\alpha_{max}} \quad (3.5)$$

The individual acceleration time to maximum velocity for each joint is a good measure for the usage of the links. Slower joints can be used for a rough movement while the faster joints can be used for error correction and fine tuning of the catching movement (compare Section 2.3.3).

Online Adaptability of Movement

For a trajectory prediction system with position measurements of the thrown object as input data, the prediction accuracy increases with the number of positions measured. The higher number of input measurements that also allow to suppress big measurement errors (outliers) or decrease the influence of unavoidable small measurement error in the position acquisition system and the increasing accuracy of the position detection system as the object is moving towards the cameras are responsible here. This suggests starting the catching movement of the robot as late as possible in order to use the most accurate prediction. This, on the other hand, requires a very fast movement of the catching system that increases wearing of the robot and, in addition, is limited by the maximum allowed torque and velocity values of the robot.

A possibility to make use of the very latest prediction while still staying well within the specification of the robot is to use an online trajectory generation algorithm. The Reflexx Motion Library [KW10] [7] allows to adapt an already executed movement within one communication cycle of the robot. In combination with the Fast Research Interface (FRI) of the KUKA LWR 4+ this allows to adapt the catching movement of the robot on the fly.

In addition to the online adaptability also, the specification of maximum velocity, acceleration and jerk for the trajectory generation is possible since revision *IV* of the library. As discussed in Section 2.3.3 humans use movements with minimum jerk. While the Reflexx Motion Library does not allow to use minimum jerk as an optimization criteria for the movement, the specification of a jerk limit is possible and will be used. In addition, the Reflexx Motion Library also determines the execution time to reach a certain movement state (position, velocity) from the current state. This is a required feature to synchronize the movement of the catching device with the movement of the thrown object. Further details regarding this aspect are discussed in the next subsection and Section 4.5.

Mimicry Human Archetype

The two prior paragraphs dealt with deriving an optimal interception position for catching with minimum impact force and the online adaptability of the movement during catching. Here the

preparation to reach the catching position and synchronize the velocity to the approaching object is described. Using more recent, and thus also more accurate, predictions for the final catching movement is important to reach the interception position accurately.

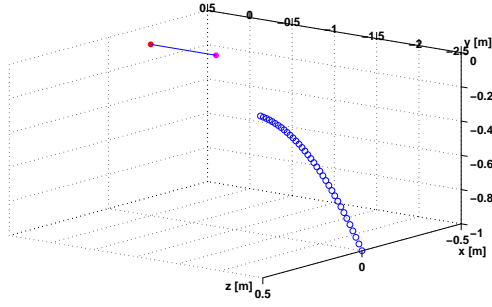
Based on the Section 2.3.3 especially the work of Kajikawa [KSOI99] and Fligge [FMVDS12] is usable here. The sequence of moving towards the object, reaching a turning point, reducing the velocity difference and grasping [KSOI99] the object is used for catching. This process is extended with the obvious but still missing phase of deceleration of the object/catching device. In order to achieve this sequence three positions are defined:

- the waiting position
- the interception position
- the stopping position

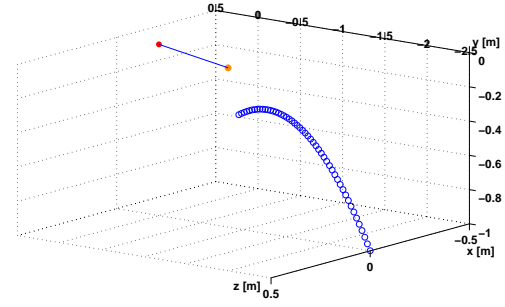
The basis for the waiting and the stopping position is the interception position. The interception position is derived like described in the first paragraph of this section. The waiting and stopping positions are used to allow acceleration to the interception position and the deceleration phase. The transition between the acceleration and the deceleration phase is smooth resulting in a limited jerk movement throughout the whole catching process.

The movement for the initial towards the waiting position is initiated based on the first reliable prediction of the flight trajectory and updated throughout the movement. This is in line with the research by Fligge [FMVDS12] where a predictive fast big movement is observed in combination with a smaller and finer adjustment movement. The second part of motion after a point of minimum velocity is considered as prospective where the current information is always used to refine the movement. The research suggests that during the initial movement all new information is discarded based on the evidence that the movement follows the minimum jerk criteria of point-to-point movements. Still the question arises whether the continues update of an action following the minimum jerk criteria could be differentiated from a non-updated movement following the same criteria. Experiments by Mazyn [MSML07] are carried out with a reduced information intake based on changing light conditions (the light is switched off after the hand's movement is initiated), but the research focuses on a different question and does not allow to clarify this circumstance. The resulting overall movement is illustrated in Figure 3.14.

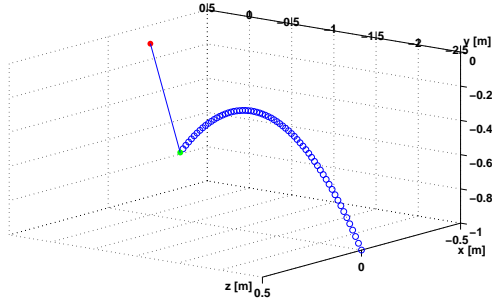
After the arrival of the initial trajectory prediction, the catching device is moved towards the waiting position. This movement is recurrently updated with new prediction data. The criteria for starting the acceleration movement towards the interception point is based on the predicted remaining flight time of the object to the (predicted) interception position and the movement execution time of the acceleration movement. Even during the acceleration movement more recent prediction data is used to update the movement but due to the hard real-time constraints of the system and the limited torques, the magnitude of refinement during this movement is limited. After reaching the interception position, the robot is deceleration until reaching the stopping position. Details for the implementation of the whole process are given in Section 4.5.



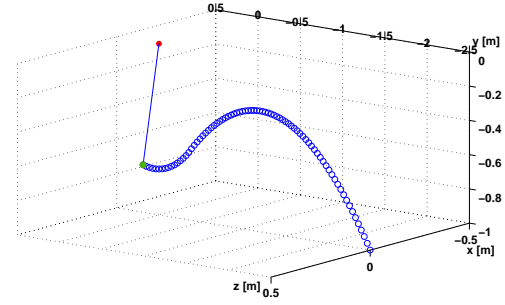
(a) Catching device (purple dot) in initial position; ball flying in early flight phase; red dot: robot origin position



(b) Catching device (orange dot) in waiting position, continuous update of waiting position for optimal acceleration to interception position; ball flying; red dot: robot origin position



(c) Moment of interception; catching device and ball in same/similar position (light green dot); red dot: robot origin position



(d) Catching device in stopping position (green dot); ball's trajectory changed from ballistic trajectory to circular in interception point; red dot: robot origin position

Figure 3.14: Visualization of the overall catching process: a) initial position, b) waiting position, c) catching position, d) stopping position; phases between the positions are a)-b) movement to waiting position, continuous update, b)-c) acceleration movement, c)-d) deceleration movement; overall a complete trajectory of a thrown ball (blue circles) and the related catching movement of the (simplified) robotic arm is shown

4 SIMULATION AND IMPLEMENTATION

The theoretical framework has been established in the previous chapter (Chapter 3). Now the focus is put on examining the capabilities of the proposed solution to problems in synthetic (e. g. a simulation environment) and real environments (e. g. a practical experiment with a robot). The process of a tennis ball's throw and the prediction of the flight trajectory will be examined as this will be the benchmark process. The application of the established information processing principle and prediction is used for predicting the future trajectory based on a set of previously acquired trajectories.

For calculation of the ball's trajectory in the simulation framework, a physical model of the process is required. This model consists of a model for the flight of the object and a model for the position acquisition error based on the used setup of cameras. While the common model in related research work [LRÅJ09], [LRÅJ10], [SC07], [FBH⁺01], [BWH10] and [BSW⁺11] is used for the flight model, the variation in the throwing process and the model for the position acquisition error is based on own work and practical experiments will be detailed in this Section.

The built-up process of the knowledge base will be the topic of the second section. This process allows to increase artificially the size of the experience database for the practical experiments and to determine the prediction accuracy's dependency on the size of the database.

The specific implementation of the soft catching strategy (discussed in Section 3.3) for the KUKA LWR 4+ will be outlined and discussed in the next section of this chapter. The limitations of the individual joint's mechanical properties (e. g. maximum torque, the range of movement) on the catching process will be outlined and influence of these parameters on the time for catching will be discussed.

A section devoted to the whole simulation process will follow. The simulation is used to improve the prediction system and deliver essential feedback for the real world implementation. All previously established models for measurement errors and process properties are incorporated into this section.

The closing section of the chapter covers the implementation of the prediction system in real world experiments with an industrial robot, a stereo camera setup and a PC. Performance related implementation details will be presented and discussed.

Results from simulation and real world experiments will be discussed in the following chapter (Chapter 5).

4.1 Environment and Nomenclature Definition

For all the following chapters, sections and subsection a common nomenclature is used. This terminology is defined here, and Figure 4.1 gives an overview of the components and a rough relation of the coordinate systems. Main aspects are the relations between the coordinate system of the robot and the camera system, the direction of the gravitational force and the origin of the thrown object.

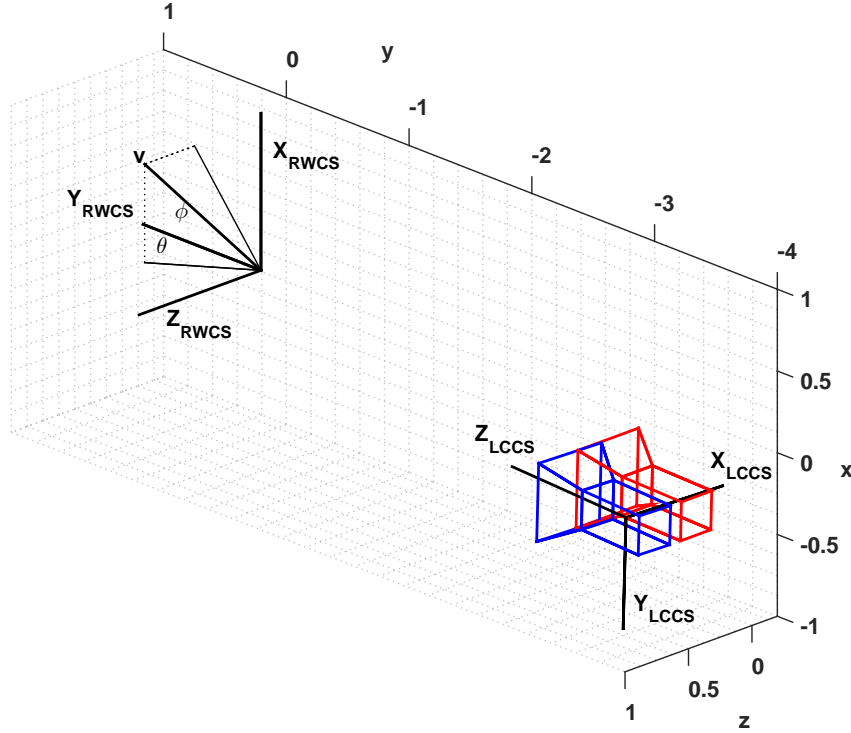


Figure 4.1: Definition of the two main coordinate system;

The robot- or world-coordinate system ($X_{RWCS}, Y_{RWCS}, Z_{RWCS}$) and the camera-coordinate system of the left camera ($X_{LCCS}, Y_{LCCS}, Z_{LCCS}$) are presented. The position of the ball when it leaves the throwing device is approximately $(-0.54; -2.90; 0.38)$ (compare Table 4.4). The definition of the throwing angles for the analysis of the throwing device's deviation uses θ as the angle in relation with the ascending component of the velocity and ϕ is the sideways angle. The blue camera illustration shows the position of the left and the red illustration the position of the right camera

An image from the laboratory, used for the experiments, is given in Figure 4.2. A ball is just launched and the cameras are hidden in the lower right corner of the image (compare Figure 4.1).

In addition to the general definition of the coordinate systems also, the nomenclature to express certain information is defined here. This data included 2D- and 3D positions, joint angles, positions of joints, distances between points or joints, Cartesian components of distances, masses of inertia, masses, and links.

- Positions are denoted by the capital letter P with the name indexed and the triple of coordinates in round brackets (example: Tool Center Point (TCP) $P_{TCP} = (0.7; 0.5; 0.3) m$. The meaning of this formalism is that the coordinate of the point TCP is $x = 0.7 m$, $y = 0.5 m$



Figure 4.2: An image of the laboratory used for the experiments; the cameras are hidden in the right lower part of the image; a ball is just launched (bottom right as well) (compare Figure 4.1)

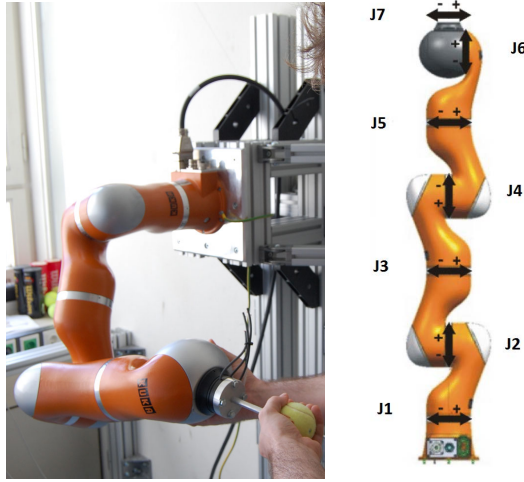


Figure 4.3: a) The KUKA LWR 4+ mounted to the wall, b) The KUKA LWR 4+ with 7 joints (J1 to J7, numbers increasing from the base of the robot to the flange)

and $z = 0.3 \text{ m}$ in terms of the related coordinate system. If no coordinate system is specified the world coordinate system is used.

- ϕ_{Jx} is the deflection of the joint x from the neutral position. The robot with all joints in the neutral position is shown in Figure 4.3. Also, the direction of positive and negative deflection is given in this figure.
- P_{Jx} is the position of the intersection of a joint and the joint that is closer to the base (or has the lower index). Only P_{J2} , P_{J4} and P_{J6} are used as these are the corners of the robot¹.
- In addition to the joint position also the position of the tool center point (TCP) of the catching device P_{TCP} , the interception position P_I and the origin of the coordinate system

¹ P_{J2} is comparable to the human shoulder joint position, P_{J4} to the human elbow joint position and P_{J6} to the wrist position. P_{J2} by example does not change when ϕ_{J1} is changing

P_0 is used

- $d_{J_x-J_y}$ is the distance between joint x and joint y
- P_{I_x} is the x-component of the interception position
- J_{J_x} is the mass of inertia relevant for joint x
- m_{L_x} is the mass of the link x between joint x and joint x1
- link x is the link directly following to joint x (between joint x and joint x+1)
- Δ is the position detection error additional indexes like Δ_r , Δ_x specify the error further e. g. error in the x-coordinate, overall error
- δ is the prediction error
- ι is the index for the discretized progressive prediction velocity voxel/class/movement symbol additional indexes like ι_x , ι_y specify the index's direction

4.2 Benchmark Process

The process chosen to test the practicability of the proposed algorithms (compare Chapter 3) is the prediction of a thrown tennis ball's trajectory. This highly dynamic process has a large number of influencing factors like the actual air density or temperature, air streams in the surroundings, variation of initial launching parameters of the ball, differences in tennis ball's mass and diameter, differences in a tennis ball's felt and the orientation of the seams [MAS08, pg. 10] and many more that affect the trajectory. It is obvious that predicting a tennis ball's trajectory is a challenging task that demands detailed information about these factors. Reconsidering the information discussed in Chapter 2, the maximum success rate for catching balls over more than 1 m achieved is 80 %. A number of researchers from that domain account the inaccurate modeling of the physical process for this.

In the following paragraphs, a physical model for the flight trajectory of a ball is discussed, this is the base for the simulation process. In order to consider the measurement errors of the tracking system in the simulation, the accuracy of the tracking system for moving spherical objects is analyzed. The usage of the determined information is twofold: It is a required information for the simulation of the whole process from the image acquisition until the planning of the robot's catching position, and the information is used to determine the required vision setup (2/4/6 cameras) for a reliable catching system for the implementation.

4.2.1 Physical Description

In the introduction of this section, it is already mentioned that the flight trajectory of a thrown tennis ball is sensitive to high a number of impact factors. Main influencing forces are the gravity and aerodynamic forces (Magnus force, lift force, drag force). For lower velocities even the orientation of the tennis ball's seam has an influence on the drag coefficient [MP01, pg. 10]

Models for Aerodynamic Forces

Aerodynamic forces on a moving object depend on the type of flow surrounding the object. Depending on the dominant flow characteristic by example the drag coefficient C_D varies to a large extent. The relations of inertia and viscosity forces can be described by the Reynolds number (Re). Relatively low Reynolds numbers characterize laminar flow while high Reynolds numbers characterize turbulent flow, the transition from laminar to turbulent flow is called critical region. In order to characterize the flow occurring when a tennis ball is thrown at velocities of $3 \frac{m}{s}$, $5 \frac{m}{s}$ and $10 \frac{m}{s}$ the Reynolds number for these three velocities are calculated. The nomenclature is based on the work by Mehta [MP01] (velocity of the ball U , diameter of the ball d and kinematic viscosity ν ; $\nu = 1.5 * 10^{-5} \frac{m^2}{s}$ for air at $20^\circ C$) and a ball diameter of $0.067 m$ is used (approved ball's diameter of Type 1 and 2 balls must be between $0.06541 m$ and $0.06858 m$ and the average equals $0.0670 m$):

$$Re = \frac{U * d}{\nu} \quad (4.1)$$

$$Re_3 = \frac{3 * 0.067}{1.5 * 10^{-5}} \approx 13400 \quad (4.2)$$

$$Re_5 = \frac{5 * 0.067 * 10}{1.5 * 10^{-5}} \approx 22300 \quad (4.3)$$

$$Re_{10} = \frac{10 * 0.067 * 10}{1.5 * 10^{-5}} \approx 44700 \quad (4.4)$$

The characterization of the resulting flow can be based on the critical Reynolds number for tennis balls. This number can be calculated for a smooth sphere (400.000 [MP01, pg. 188]) but the determination for tennis balls has to be done based on experiments. Experiments by Mehta [MP01, pg. 189] with Reynolds numbers between 100.000 and 284.000 (equaling velocities from $\approx 22 \frac{m}{s}$ to $\approx 64 \frac{m}{s}$) do not cover the critical Reynolds number for either a new or a used tennis ball. The interpretation that the critical Reynolds number of a used tennis ball is 100.000 [MP01, pg. 194] with the statement that it is presumed that transition had already occurred which means that the flow for all velocities is considered to be transcritical.

Experiments by Dunlop [Dun13] cover a lower range of Reynolds numbers ranging from 30.000 to ≈ 85.000 (equaling velocities from $\approx 6.7 \frac{m}{s}$ to $\approx 19.0 \frac{m}{s}$). These experiments cover the upper range of the relevant throwing velocities for transport-by-throwing based on tennis balls. The drag coefficient C_D determined here is close to constant in the range of $Re = 30.000$ to 40.000 with a value of 0.7 decreasing to 0.6 at $Re = 50.000$. The same value can also be extrapolated from the experiments by Metha.

The drag coefficient determined with the experiments is needed to calculate the resulting drag force F_D based on the air density ρ , object velocity v , drag coefficient C_d and the reference area A . The relation is given in Equation 4.5

$$F_D = \frac{\rho * v^2 * C_d * A}{2} \quad (4.5)$$

The direction of the resulting drag force is opposing the motion's direction between the object and the medium. In case the medium is moving, the resulting drag force is in general not aligned with the opposed direction of the object's movement.

Practical Experiments on Aerodynamic Forces

Initiatives of the International Tennis Federation to specify adequate tennis balls for different age groups cause a number scientific initiatives to quantify the aerodynamic behavior of modern tennis balls. Experiments at different velocities with varying spin of the balls enable to extrapolate the aerodynamic behavior for the presented task. This data is used for the simulation environment to improve the similarity between the simulated process and real experiments. A highly accurate simulation could improve the prediction performance based on simulated experience.

An early study by Chadwick [SGC98] determines the drag coefficient of a tennis balls in an interval between 0.35 for a worn ball up to 0.6 for a new ball. The velocity of the air stream for this results is either $20 \frac{m}{s}$ or $26 \frac{m}{s}$ which equals a range for the Reynolds number of 85.000 to 117.000. Even for maximum rough spheres, the limit of the drag coefficient is seen in the range of 0.8.

More recent work from Alam [ATW⁺07] focuses on the influence of spin on the drag coefficient and other parameters. The study uses a wider range of velocities ranging from $40 \frac{km}{h}$ to $80 \frac{km}{h}$. Especially the lower end of this range is interesting for the transport by throwing approach. The drag coefficient determined in this range for a spin rate of $500 \frac{1}{min}$ equals 0.7 for standard tennis balls. The tendency is that the drag coefficient is increasing with decreasing velocity.

Om 2008 a newer publication was presented showing a wide range of measurements for the drag coefficient for a number of used and new tennis balls [MAS08]. Balls with a different level of wearing, taken from the US Open tennis tournament, were compared and the measured data from wind tunnel experiments show that depending on the level of wearing the Reynolds number at $45 mph$ ($72.4 \frac{km}{h}$ or $20.1 \frac{m}{s}$) varies between 0.45 and 0.68. Results from an earlier study even show Reynolds numbers of up to 0.95 at lower velocities.

Chosen Parameters for the Simulation

There is a wide variation of the drag coefficient in the experiments conducted in related work. The dependency on the wearing level, where the drag coefficient initially rises (presumably due to the increase in roughness of the felt) and then decreases again is common to all experiments. Based on the wear level of the used balls the value of 0.6 is considered as an initial value for the simulation. This bandwidth of measured values also shows the complexity of the accurate physical description/modeling of the presumably simple process of a tennis ball's aerodynamic behavior or flight trajectory. The final part of the following section will elaborate on the topic of parameter selection.

4.2.2 Analysis of the used Equipment

For a simulation with relevant results for the practical experiment, the analysis of the used equipment and algorithms is of high importance. This analysis is done in this section. Most of the parameters that have to be determined for the simulation (properties of the position detection system (Subsection 4.2.2.1) or the throwing device (Subsection 4.2.2.2) cannot be measured directly due to the lack of the measurement system or the required ground truth for an error analysis. Due to this fact the analysis is based on a set of acquired throws with a velocity of $4 \frac{m}{s}$, $4.25 \frac{m}{s}$, $4.5 \frac{m}{s}$, $4.75 \frac{m}{s}$ and $5 \frac{m}{s}$ initial velocity. This data is used to determine the deviation of the throwing device and the accuracy of the position detection system. An analysis of the robot, which is used for catching, is not done because the specification gives information about the position accuracy

of the robot and deriving a realistic dynamical model (which is not provided by the manufacturer) would be a topic that equals another research project. The position detection system includes processing steps where different algorithms can be chosen along the processing chain, for example, the usage of background subtraction or not and Hough or RANSAC circle detection. In addition to this parameters also the way to estimate the real positions based on a larger number of datasets from an acquired trajectory can be varied (polynomial estimation, Rauch-Tung-Striebel smoother, iterative estimation) and thus the matrix of parameter combinations is very high. In order to cope with this circumstance the information from both steps was combined and thus the analysis of the position detection system in Subsection 4.2.2.1 is using the polynomial position estimation (which is a result of Subsection 4.2.2.2) and the parameters of the position detection algorithms are varied. The determination of the throwing device's deviations, on the other hand, is based only on the most suitable combination of position detection algorithms (in this case the RANSAC algorithm with background subtraction that is the result of Subsection 4.2.2.1) and the approaches for the position estimation are varied.

4.2.2.1 Position Detection System

The second main aspect of the simulation that has to be modeled is the accuracy of the tracking system. The influence here is essential as the algorithms for prediction also have to deal with the position detection errors of the system. Furthermore, the relationship between the setup and the number of the cameras and the accuracy of the position detection achieved by the visual sensor system is of main interest here. In the following paragraphs, the setup for the determination of the vision system's accuracy and the results of the measurements will be presented and discussed.

Related Work

In terms of the analysis of a stereo camera system's accuracy a publication from 2006 [LZL06] offers an error model that is verified by an experiment. For the theoretical deduction of the error, the researchers choose a combination of parameters to allow the generalization of the approach to other environments. While the parameter α , used for the angle between the connections of the cameras focal point with the object's position, is a logical and general one, the usage of the parameter h , used for the height of the camera's focal point above the object, cannot be understood (compare Figure 4.4). h is only depending on the definition of the coordinate system in the object's position O and a rotation of the coordinate system obviously changes the parameter h significantly. In case the researchers identify the position detection error individually for each spatial direction this would make a difference but as only the magnitude of the position measurement error is finally determined this definition is questionable. Using α and d in contrast would allow to use the research results more general as a rotation of the coordinate system in O has no impact on the magnitude of the positioning error. Still the basic method and the approach to the experiment is done very well. Only the small number of measurements to verify the theoretical approach is disadvantageous.

The practical setup uses a robotic arm with a 6 mm diameter sphere mounted to its flange. This sphere is moved into 20 areas that are defined based on the parameters α and h . In these areas 15 positions in a grid of 50 mm are used for the accuracy determination and additional positions are used for calibration. The position accuracy of the robot is given at ± 0.013 mm. Whether this parameter is the repeatability accuracy or the overall Cartesian accuracy is not stated explicitly.

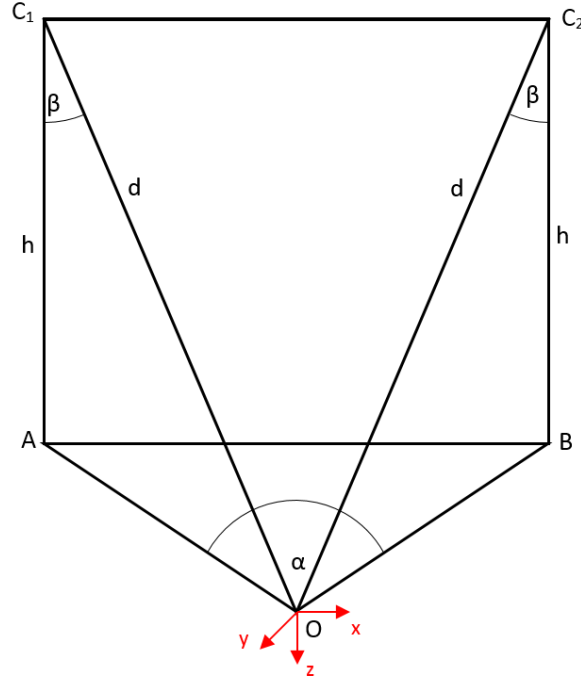


Figure 4.4: Camera setup, distances and angles between object O , camera 1 C_1 , camera 2 C_2 and the basis points A and B according to [LZL06, pg. 2];

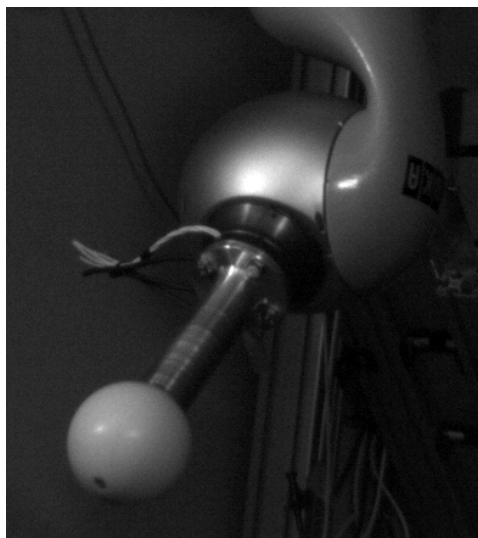
The usage of parameter h is questionable as the distance between the cameras C_1 and the object O is depending on h . Increasing h while keeping d constant is equal to a rotation around the x -axis of the object and should lead to the same euclidean error with a change of the error components in y - and z -axis

No remarks about the mastering of the robot are given. The image data contains measurements with a still standing robot thus no dynamic effects are considered. Due to the questionable parameter selection the results are not relevant here but the setup for the experiment is used as a base for the experiments here.

In addition to the position also, the resolution of the cameras in the stereo setup influences the accuracy of the position detection system ([PKFB10, pp. 5f] and more extensively [Pon09]). For a static object, the spatial resolution is relevant while for a moving object also the temporal resolution has to be considered. In many applications, the available bandwidth of the camera's interface is limited and thus the product of spatial and temporal resolution. Finding the optimum balance between these two parameters is a main goal here. The usage of an Area of Interest (AoI) allows to avoid running into this limitation in high-speed high-accuracy environments.

Experiment Setup for Static Objects

The availability of the KUKA LWR 4 robotic arm, which has both joint position and Cartesian position control mode, allows to use the robot as a reference system to put an object into different positions in the field of view of the vision setup. The setup consists of the robot with a spherical object mounted via an adapter to the mounting flange. An illustration of the test object is given in Figure 4.5(a).



(a) KUKA LWR 4 with the spherical tool on the flange



(b) Stands for the sphere used for the camera accuracy experiments; three stands with different heights (80 *mm*, 280 *mm* and 480 *mm*) for experiment with the external illumination, stand with triangle-shaped base with self illuminating sphere.

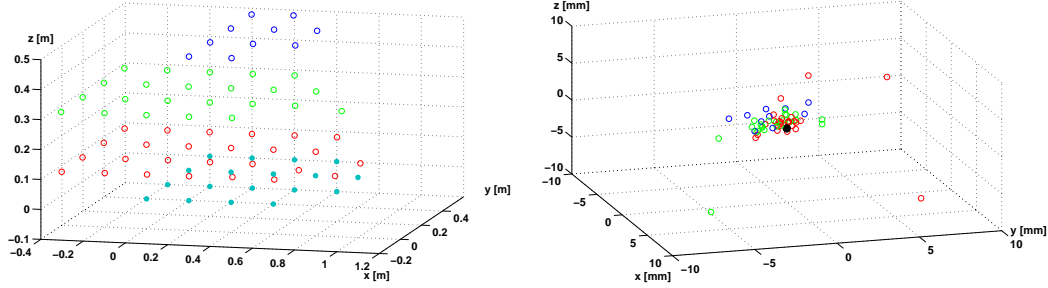
Figure 4.5: Tools used for the determination of the camera system's accuracy

During the setup of the experiments the problematic aspect of mastering² got evident. The standard mastering of the robot based on vernier scale was not accurate enough to allow a precise movement in Cartesian mode. This got obvious during a simple test where the robot's flange was fixed in a position, and the position of the robot's links was changed to the maximum extent. This is possible due to the 7 DoF of the KUKA LWR 4 robotic arm. Reading out the Cartesian position during this experiment brought up errors of up to 5 *mm* which are assumed to be in the range of the camera system's accuracy and thus the robot cannot be used as a reference system. Not that the Cartesian position accuracy does not correspond to the repeated positions accuracy as the links are moved to the same position in the second case.

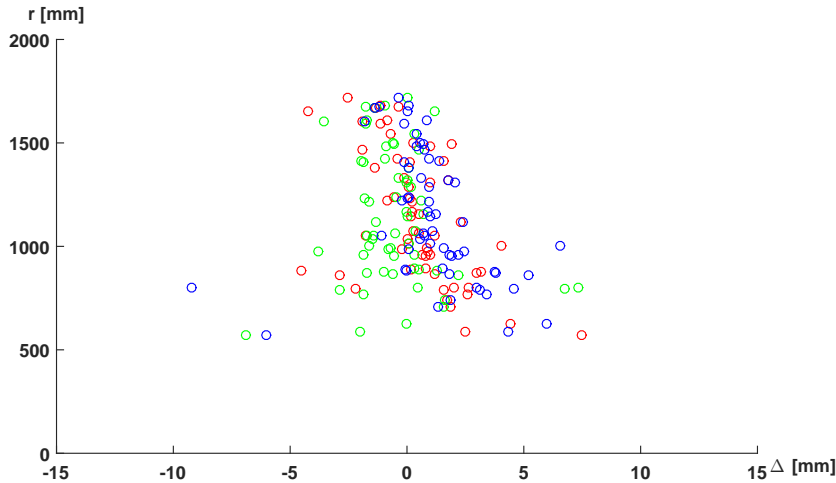
Experiment with external illumination The identified problem of mastering will be discussed more extensively in Section 4.5. For the determination of the vision system's accuracy, a different test setup was designed. This setup consists of a glass plate and 3 differently sized stands (small: 80 *mm*, medium: 280 *mm* and large: 480 *mm*) for a sphere (compare Figure 4.5(b)). A millimeter paper is applied onto the glass plate giving a 2D coordinate system on a plane. The stands with the sphere were positioned in a grid of 200 *mm by* 200 *mm* on the plate, and the positions of the spheres acquired with the vision system. Thus, a 3D grid of the center positions of the sphere could be extracted from the acquired images. For reference, the base points of the stands were extracted manually from the images. This could be achieved by equipping the stands with positions markings. These reference points were used to determine the accuracy of the vision setup without considering the influence of the Hough transform used to obtain the center position

²Mastering here means the adjustment of the zero positions of each of the KUKA LWR 4's joints. This information is essential for translating positions in joint space into Cartesian space

of a circle or spherical object. By comparing these reference base-points with the real distance on the millimeter paper, an alignment of the measured data with the plane was achieved (least square fit of measured data to a plane and projection to this plane).



(a) Points on ground plane: turquoise; small stand: red, medium stand: green, big stand: blue; compare Figure 4.5(b) (b) Error-component analysis of position detection system (small stand: red, medium stand: green, big stand: blue); measurements have a mean offset of (0.43; -0.48; 1.03) mm; this are the coordinates of the point cloud's center



(c) Errors component illustration of position detection system (red: x, green: y, blue: z)

Figure 4.6: Measurement data from camera accuracy determination: a) raw measurement data with reference measurements from ground plane points, b) error-component-analysis, c) spatial error over distance from camera system

The results of the measurements are presented in Figure 4.6. The main finding of these measurements is that a constant offset from the measured points of (0.43; -0.48; 1.03) mm, that can also be seen in Figure 4.6(b) as the center of the point cloud. The close to constant measurement error with decreasing distance to the cameras (but also decreasing the distance to the flood lights) shown in Figure 4.6(b) is remarkable. The increased magnitude for the closest measurements is due to errors of the object detection caused by the inhomogeneous illumination (small distance to flood lights). In order to clarify this circumstance, an updated experiment with a self-illuminating sphere in dark conditions was made. In comparison to the previous experiment only data with one mounting stand height was acquired (see the lower right stand in Figure 4.5(b)) but the influence of illumination could be clarified.

Experiment with self-illuminating sphere To avoid the influence of the inhomogeneous light, a self-lightning sphere was used for another experiment to determine the accuracy of the object detection algorithms isolated. The stand used is shown in Figure 4.5(b). The procedure for the accuracy determination is the same as in the previous case. The sphere is put at predefined positions on the millimeter paper, and these positions are also determined through the images of the stereo camera system. For a baseline comparison again manually identified positions on the millimeter paper are used. The information obtained with this experiment is again the random error of the position determination via the stereo camera system. The related figures to Figure 4.6 are shown in Figure 4.7.

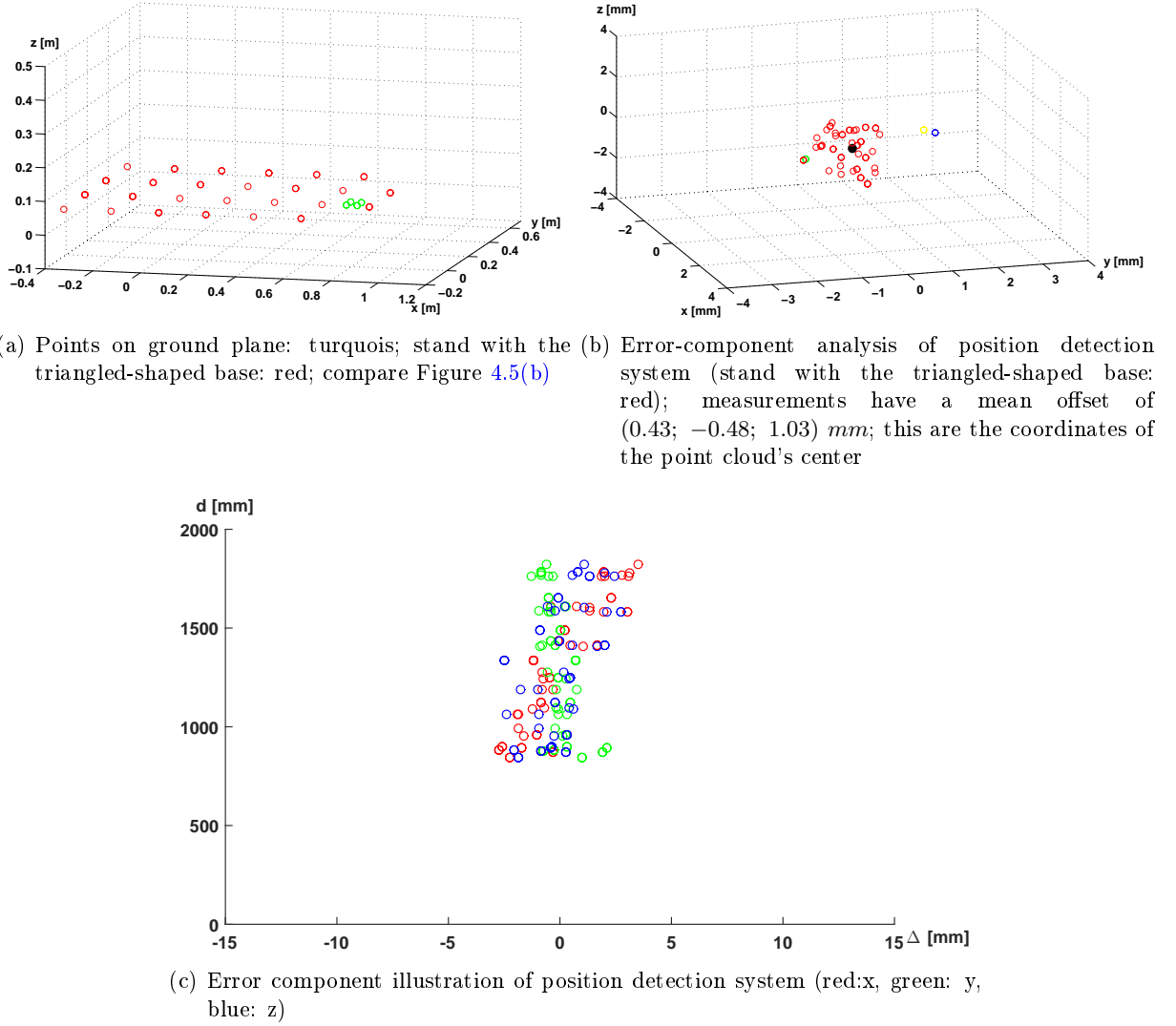


Figure 4.7: Measurement data from camera accuracy determination: a) raw measurement data with reference measurements from ground plane points, b) error depending on size of stand; measurements have a mean offset of $(-1.49; 1.28; -0.02) \text{ mm}$, c) spatial error over distance from camera system

Results and Interpretation The accuracy of the camera system is a main factor for the overall system. In addition, the accuracy determination is needed to model the vision system's

error accurately. The measurement error is given in a coordinate system called stereo camera coordinate system (SCCS). The origin of this coordinate system is determined based on the average position of both cameras; this means the origin is in the middle of the cameras. Determining the orientation of the SCCS is more challenging. The used orientation is based on the optical axes of the individual cameras determined with the camera calibration. The connection of the SCCS origin with the intersection of both camera's optical axes is the base for the orientation determination. As the two individual optical axes, in general, do not intersect in a point the center point of the shortest connection between the two optical axes is used. This allows a general description of a stereo camera setup and a general usage of the determined attributes (like accuracy in this case). A graphical illustration of the SCCS origin position and orientation is shown in Figure 4.8 while the more general definitions are given in Figure 4.1 in Section 4.1.

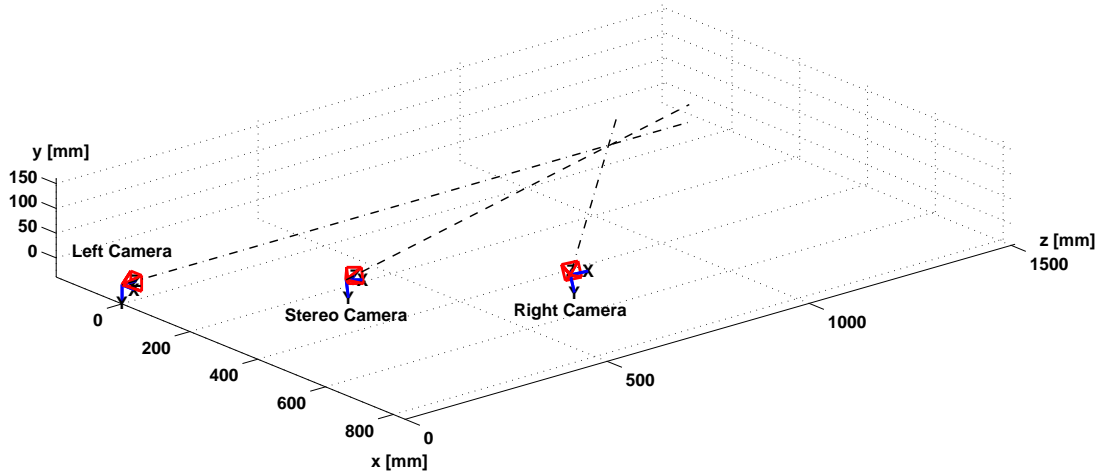


Figure 4.8: Illustration of the stereo camera coordinate system (SCCS) in relation of the coordinate system of the right and left camera; dash-dotted lines are the optical axes of the individual cameras, the dashed line is the hypothetical optical axis of the stereo camera system, the coordinate system's definition is shown in blue; all optical axes are in line with the related z-axis of the camera coordinate system

In the experiments with external illumination the mean position detection errors were (0.43; -0.48; 1.03) *mm*. The unknown distance between the reference corner of the small triangular ground plane of the stand and the center position of the self-lightning sphere prohibits to determine this error in the later case. The values are given in the SCCS. An overall overview of the error components (mean errors, error's standard deviation) is given in Table 4.1.

	ext. light			int. light		
	<i>x</i>	<i>y</i>	<i>z</i>	<i>y</i>	<i>x</i>	<i>z</i>
mean errors	2.12	-5.88	2.26	-	-	-
standard deviations	2.24	1.92	2.27	1.76	0.77	1.34

Table 4.1: Numerical representation of the numbers shown in Figure 4.6 and Figure 4.7; ext. light is the case with external illumination, int. light respective with the self illuminating sphere; *x*, *y*, *z* are the components in the CSSC (compare Figure 4.8; all measures in mm)

When comparing the errors, the root mean squares of the individual standard deviations are

used. This value is 2.15 mm for the external illumination and 1.35 mm for the self-lightning sphere. Obviously, the lighting conditions and illumination setup has an influence on the random error. As a quintessence for the practical experiment, the setup of the illumination has to be done with caution. Especially close to high power light the error of the position measuring system is increasing. The random error seems to be unaffected by the lighting conditions for a large range but in the case of very short distances to the dominating light sources the random error will increase.

Experiment Setup for Moving Objects

The information of the static object's detection accuracy gives valuable information about the algorithms inherent accuracy. The step to (fast) moving objects, in addition, gives information about the influence of the motion blur, caused by the movement of the object during exposure, on the accuracy in addition. Finally, the overall accuracy of the position detection system is the relevant information for the simulation, and thus, this accuracy is estimated here. The problem with this estimation is that the ground truth is not available and thus the trajectory estimation, used in the previous Subsection 4.2.2.2, is used here as the baseline for the position detection error determinations. Out of the three variants (polynomial fitting, Rauch-Tung-Striebel smoother and iterative fitting) the polynomial fitting is used because of the stability and the accuracy of trajectories' description (compare Subsection 4.2.2.2). The position detection system is the same as used in the experiments (compare Section 4.5). Here variants for the implementation are compared in addition. These variants are differing by the following aspects (compare Section 4.5 for further details):

- Usage of (static) background subtraction
- Usage of Hough algorithm or Random Sample Consensus (RANSAC) algorithm for circle detection

The combinations of these parameters lead to four variants of algorithms. Due to the fact that the Hough algorithm is deterministic, only one run of the Hough algorithm per trajectory is required. For the non-deterministic RANSAC algorithm, a set of 1000 runs is done. A statistical analysis of the set of runs with similar velocity and all velocities is done. The error analysis includes the following aspects:

- Ground truth
 - for the positions determined via the Hough algorithm the polynomial estimation for the whole trajectory is used as the ground truth
 - for the positions determined via the RANSAC algorithm to variants to determine the ground truth are used. These are the polynomial estimation of the
 - * mean of the positions derived based on 1000 RANSAC runs
 - * median of the positions derived based on 1000 RANSAC runs
- The spatial components of the distance in the stereo camera coordinate system (SCCS, compare Figure 4.8). This information is the relevant input for the simulation as a dependency of the error on the position of the ball in relation to the center of the stereo camera system is assumed to exist.

- The measure r as the distance of the detected position to the estimated position (independent of the coordinate system)

These error measures are analyzed with the distance of the object's center to the stereo camera system's origin as baseline due to the obvious relation between distance and the absolute resolution of the image information. In order to allow sound interpretation, the errors of the distance measurement are binned in bins with a size of 0.1 m at a range of 0.6 m to 3.0 m and within these bins a statistical analysis is done. For the spatial components (x, y, z) the error is modeled with a normal distribution and the measures for μ and the 99 % confidence band ($\mu \pm 2.576\sigma$) are presented in the three top rows of the related figures (Figure 4.9 and Figure 4.10). In the bottom row, the overall error is presented. Due to the fact that this error is limited to values above 0, the error is modeled with a Weibull distribution and also, in this case, the mean and the 99 % confidence band is given. Figure 4.10 illustrates the three processing algorithms based on an input

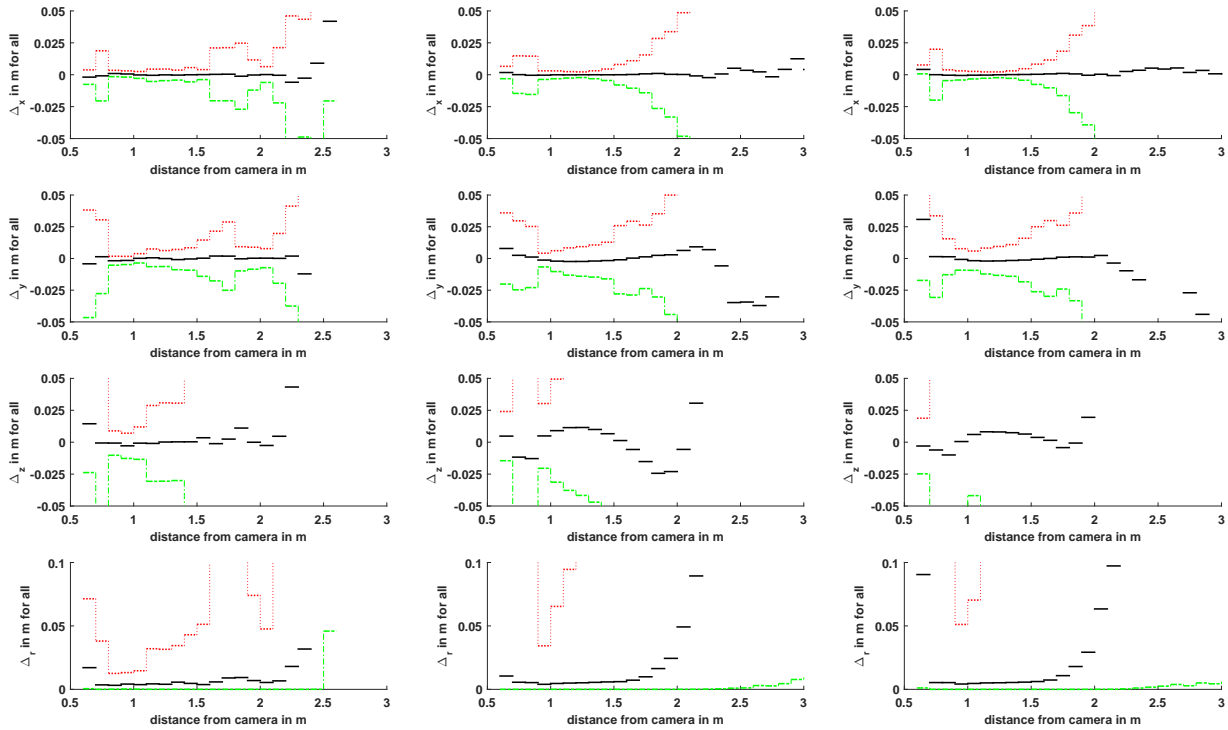


Figure 4.9: Error analysis of ball's position detection accuracy without background subtraction. 99 % confidence bands of distributions ($x/y/z$: Normal; r : Weibull); left column Hough, center column RANSAC with mean positions of 1000 runs as ground truth, right column RANSAC with median positions of 1000 runs as ground truth;
for all three cases polynomial functions are used to estimate the ground truth data from the complete dataset of the trajectory

image without background subtraction. The related figure determined with active background subtraction is presented in Figure 4.10. Having a look at both figures the usage of background subtraction improves the accuracy in many cases, especially in a distance of more than 1 m from the stereo camera system's origin. When comparing the different algorithms the Hough algorithm delivers the most accurate results. In the case of no background subtraction, this is clearly visible when considering the bottom row of Figure 4.9, in the case of usage of the background subtraction (bottom row of Figure 4.10) the difference between the Hough algorithm (first column)

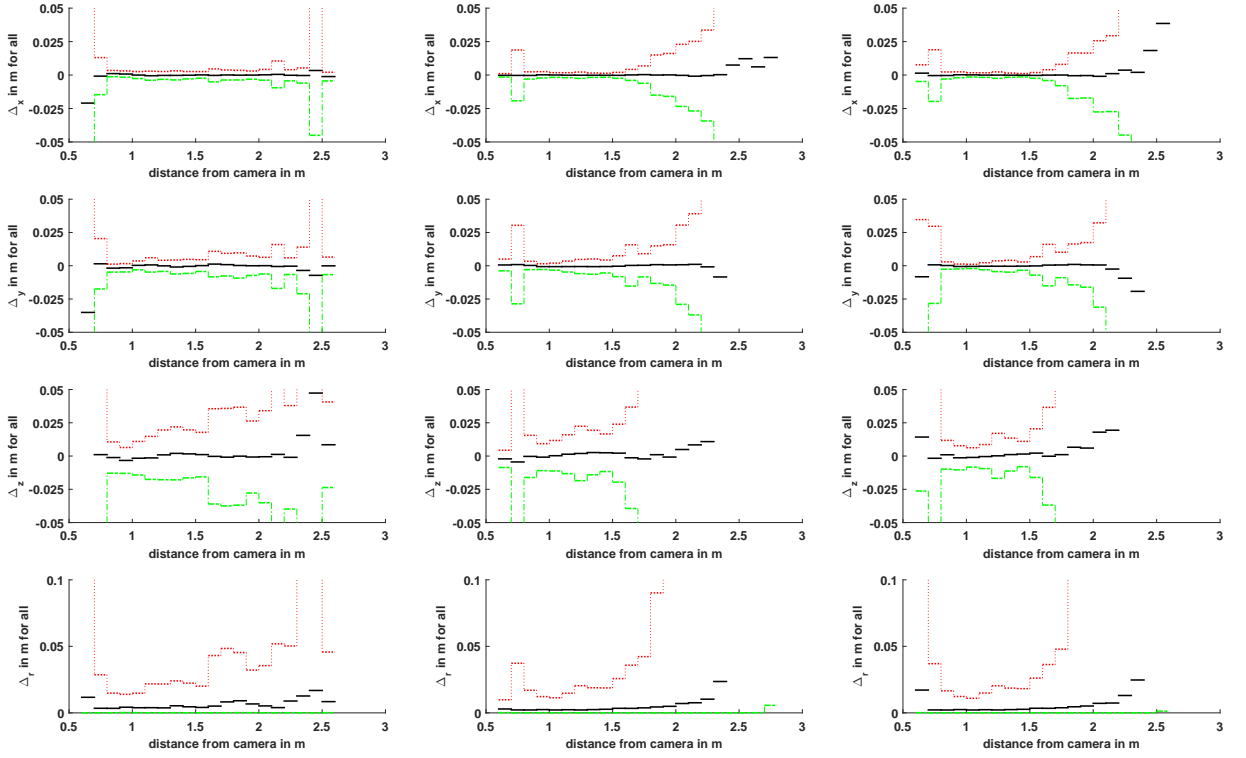


Figure 4.10: Error analysis of ball's position detection accuracy with background subtraction. 99 % confidence bands of distributions (x/y/z: Normal; r: Weibull); left column Hough, center column RANSAC with mean positions of 1000 runs as ground truth, right column RANSAC with median positions of 1000 runs as ground truth;

For all three cases polynomial functions are used to estimate the ground truth data from the complete dataset of the trajectory

and the RANSAC algorithm with a mean ground truth is less significant. Overall both the Hough algorithm with background subtraction and the RANSAC algorithm with background subtraction are viable options. One advantage of the RANSAC algorithm is the lower computation time by approximately 33 % (compare [G15, pg. 76ff]). For comparison issues, all calculations with the RANSAC algorithm (which use the mean or median of 1000 runs here as ground truth) were also done by using the each individual runs polynomial estimation as ground truth. This is a realistic scenario for the practical usage. The related statistical numbers and figures are very similar to the ones presented here. In below 0.7 m and above 1.5 m distance the errors are a little bit lower but the main result is the same (compare Figure A.3 and Figure A.4). The statistical data derived from these experiments is the basis for modeling the positions detection system's accuracy in the simulation (compare Section 4.4 and the related results in Section 5.1).

Optimization

The position detection based on the RANSAC algorithm allows one more step to improve the prediction accuracy. If the allowed radii for the circle search are limited, the detection error can be decreased. Limiting the allowed radius of the circles can be done based on the currently detected position of the ball for the next frame. The equations are given in Section 3.1 and the results achieved based on limiting the circle's radius in the successive image into a range of 92.5 %

to 107.5 % of the ball's calculated radius in the current frame with an additional fixed margin of -2 pixels and $+2 \text{ pixels}$ for the lower and upper limit improved the position detection accuracy significantly over the whole range of the ball's motion. The results are presented in Figure 4.11.

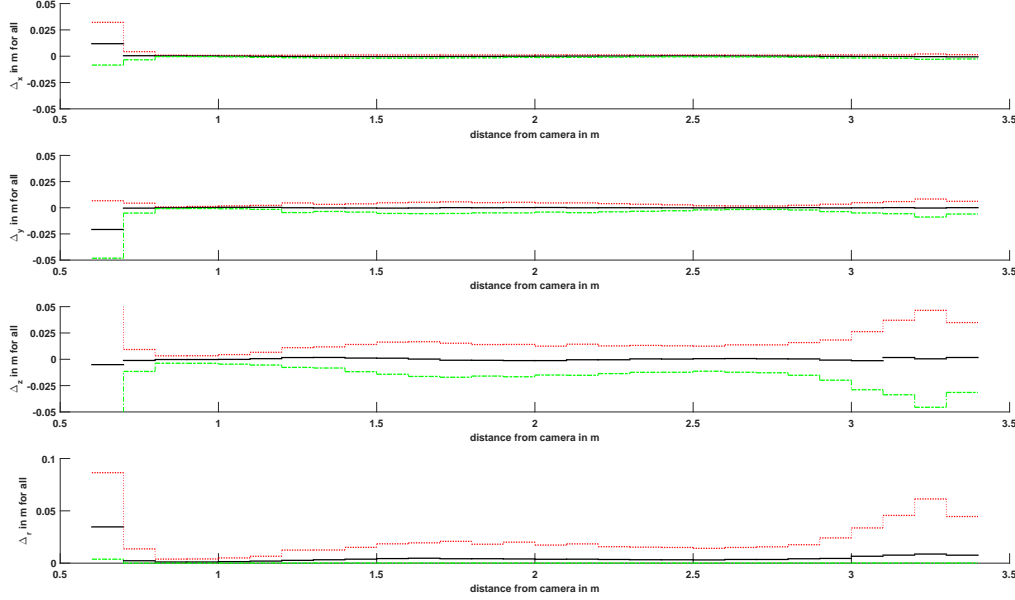


Figure 4.11: Error analysis of ball's position detection accuracy with background subtraction and additional radius prediction/limitation. 99 % confidence bands of distributions (x/y/z: Normal; r: Weibull);

polynomial functions are used to estimate the ground truth data from the complete dataset of the trajectory

The improvement is significant, and the result shows that the tracking error over a distance of 0.7 m to 2.9 m is close to constant. As expected the error for the depth component (z in the SCCS) is higher than the sideways or upwards error. The errors in the initial part of the flight path are existing due to the ball's size being similar to the size of the AoI, and thus, a small miss prediction of the AoI movement can cause cutting of the ball in the AoI. The error is limited to the very first frame, and thus, the first frame of the detected positions is not used for the prediction of the flight. This can lead to an increased position detection error. The constant error over such a long range also hints that the influence of the camera's resolution on this error is negligible in this area as the error should increase otherwise with rising distance from the cameras. Regarding the usage of more cameras, the constant error in this area also limits the benefits of using more cameras and switching to the closest ones for improved tracking accuracy. The reasons for the tracking error are more likely in the algorithm for detecting the edges of the ball. This also allows to use more cost-efficient cameras for the tracking task. The concept of increasing the frame rate based on the AoI and the fact that the position detection accuracy is not mainly influenced by the resolution should be examined further in future work.

4.2.2.2 Throwing Device

The throwing device used for the practical part of the work is a self-built throwing device with a linear acceleration direction (shown in Figure 4.12). The device can be controlled remotely from

a PC and allows to set the tension force of the linkage connected to the throwing cup. The initial velocity is measured based on two light barriers at a distance of 8 cm at the very end of the throwing device. The second light barrier is also used to start the triggering of the cameras (with a rate of 110 *fps*).

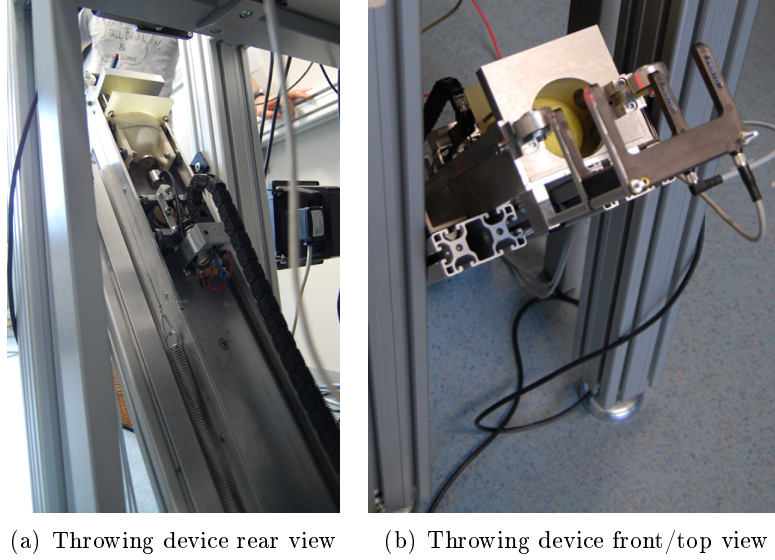


Figure 4.12: Linear throwing device used for the catching experiments; acceleration based on two tensioned springs with drag ropes; setting of a throwing force and measurement of throwing velocity based on two light barriers is possible via a control unit

The variation of the throwing device is important information for the simulation of the whole system. A sample dataset of trajectories at different target velocities ($4 \frac{m}{s}$, $4.25 \frac{m}{s}$, $4.5 \frac{m}{s}$, $4.75 \frac{m}{s}$, $5 \frac{m}{s}$; 21 trajectories each) was acquired. The setup for the observation is shown in Section 4.5 and the same system is used for the determination of the position detection system's accuracy as well (see Section 4.2.2.1). The setup for the acquisition of the variance of the throwing device is similar to the one used for the determination of the accuracy of the position detection system (compare Subsection 4.2.2.1) as well as the final practical experiments (compare Section 5.2). The cameras are placed on the throwing side in order to allow a precise acquisition of the flight's initial phase. The estimation of the initial parameters velocity \mathbf{v} (thus also magnitude of the velocity v , the initial throwing angles for the ascending component θ and the sideways angle ϕ (compare definition in Figure 4.1)) and the initial position are done with three estimation methods which will be compared and discussed in the following paragraphs.

Polynomial fitting based estimation

The first variant of parameter estimation is based on least squares polynomial function fitting. The order of the functions are 3^{rd} , 2^{nd} and 2^{nd} for the spatial directions x , y and z (compare definitions of the whole system's world coordinate system in Figure 4.1). Outliers are suppressed based on the following criteria: If the difference between the initially fitted value and the measured value is bigger than two times the standard deviation of this distance for all the data points the related point is suppressed for the second (and final) step of least squares fitting.

Rauch-Tung-Striebel smoother based estimation

The Unscented Kalman Filter based Rauch-Tung-Striebel smoother is the second variant of the parameter estimation. The filter/smoothener is configured to consider gravity and (an approximation of quadratic) air drag. The initial values of the state vector including position and velocity are based on the initial estimations for the polynomial estimation. The matrices for the covariance of the state, the covariance of the estimation noise and the covariance of the process noise are chosen manually as no better approach could be identified. The smoother processes the data forth and back, thus, the smoothened output for all data points is based on the consideration of all relevant measured points. The result of the smoother, and especially the initial parameters of the trajectory, showed a big dependency on the initial parameters and the matrix of the covariance of the state. In order to achieve stable results, the smoother was executed 12 times where the smoothened result of the prior processing was used as input for the following. The number of 12 repetition is based on the minimal chances of the estimated parameters after the 8th to 10th repetition.

Iterative model based estimation

The final variant of parameter estimation is based on the iterative flight model discussed in Section 4.2.1. Due to the non-linearity of the air drag, this model has to be calculated iteratively. In order to find the best suitable trajectory for the related measurements a three repetition Monte Carlo simulation with (by a factor of 3) decreasing variances are used with 5000 trajectories for each repetition. Initially, the following parameters were (normally distributed) varied: velocity \mathbf{v} (thus also magnitude of the velocity v), initial throwing angles for the ascending component θ and sideways angle ϕ (compare definition in Figure 4.1) and the initial position. This resulted in average distance of 3.0 cm per position between the fits and the measured positions. Including the drag coefficient c_d into the variations kept the estimation error at 3.0 cm. Varying the mass of the tennis ball m in addition is not meaningful as the quotient of c_d and m is used to determine the drag force. This means that by varying c_d implicitly also m is varied. The estimation error is 1.9 times the error compared to using the RTS estimation (1.58 cm) and 2.1 times the error of the polynomial model (1.43 cm). The challenge of fitting this model to the measured data is already discussed in [Pon09, pg. 59ff] and is partly related to the way the determined positions are fit to the measured data. Still, even using 5000 trajectories and three steps of improvement, the results could not be improved.

Results

Velocity The results of the estimation process of the three methods are briefly presented and discussed. Regarding the estimation of the velocity, the information about the measured value (from the light barriers of the throwing device) could be incorporated into the analysis. Thus, it is possible to compare the three estimation methods with the measured value and identify the quality of the estimation. The histograms for all four velocity measures is given in the top row Figure 4.13. Also normal distribution fits are presented with the Gaussian bell curve. Due to the fact that the particular relation between the values for a single trajectory is lost, additionally, the scatter plot is given in the bottom row. The main results for all velocity-setpoints are the same. The polynomial estimation showed very similar histograms and fitted Gaussian bell curves as the measured values. The iterative estimation is similar but has a slight tendency to underestimate the initial velocity and shows similar variations overall. The Rauch-Tung-Striebel smoother based

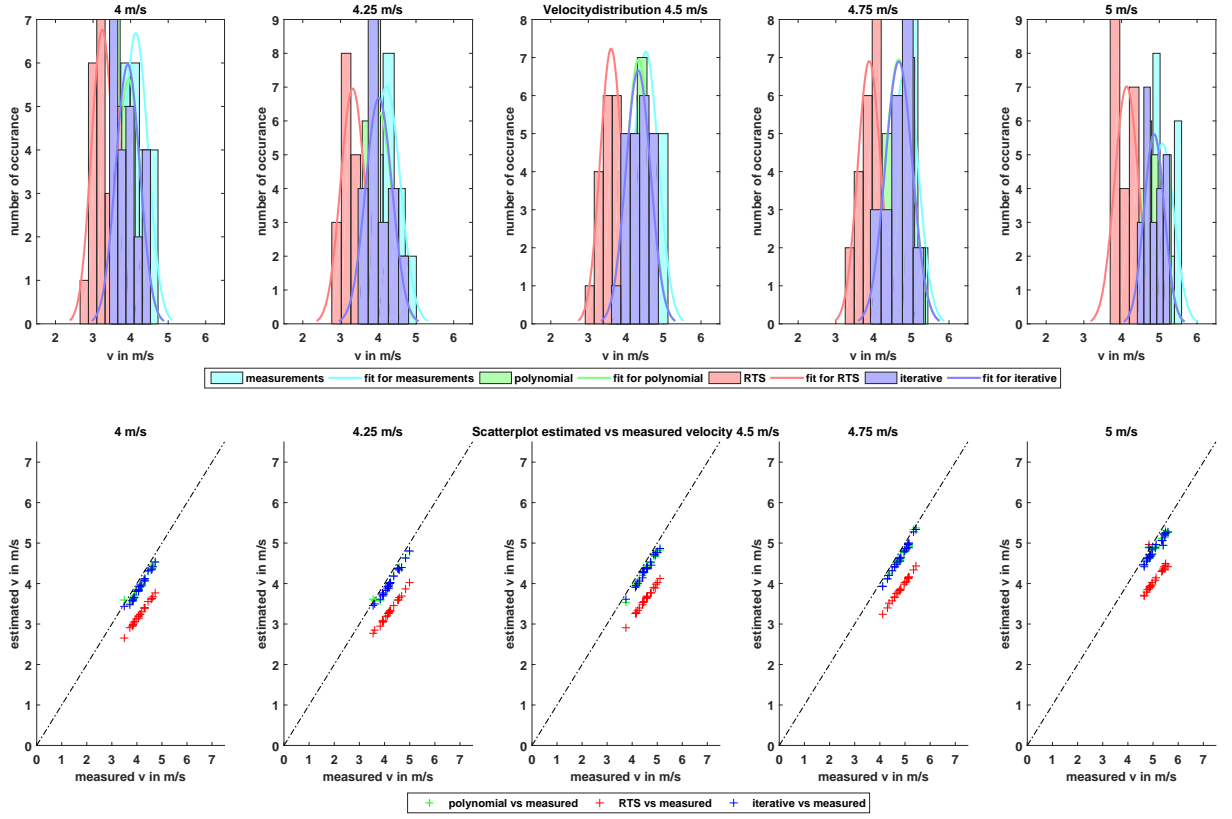


Figure 4.13: Graphical illustration of the velocity estimation results in comparison to (light barrier based) measurement data;

For the five velocities ($4 \frac{m}{s}$, $4.25 \frac{m}{s}$, $4.5 \frac{m}{s}$, $4.75 \frac{m}{s}$, $5 \frac{m}{s}$) a histogram and a Gaussian bell curve fit are presented in the top row. The bottom row shows scatter plot that outline the relation of corresponding measurements for the measured and estimated velocities. The estimations based on the polynomial method describe the measured data best, next best is the iterative method, and the RTS method has clearly the highest systematic difference. The numbers for the Gaussian bell curve fits are given in Table 4.2

estimation shows the biggest difference in the measured velocities. The scatter plots underline this behavior. If all points are on a single line which can be extended through the point (0, 0) the relation would be perfect (the dash-dotted line illustrates the optimum). Both the polynomial method and the iterative method show this behavior with some errors, but the Rauch-Tung-Striebel smoother method does clearly fail in fulfilling the second criteria. Overall this renders the polynomial estimation as the most suitable one for the parameter estimation especially in the first phase of the flight path. The comparison of the distribution fittings results is presented in Table 4.2.

Ascending and sideways angles In the case of the analysis of the initial angles' values, only the three methods for estimation are available as measuring these angles in the experiment setup with the virtual coordinate system located within the robot's enclosure is not accessible. The analysis here includes graphs for all trajectories of each velocity group. The ascending angles θ estimates are presented in the top row of Figure 4.14 while the bottom row illustrates the sideways angle ϕ .

Similar to the estimation of the initial velocity in the previous paragraph here the polynomial

set v	measured		polynomial		RTS		iterative	
all in $\frac{m}{s}$	μ	σ	μ	σ	μ	σ	μ	σ
4.00	4.14	0.33	3.95	0.31	3.25	0.29	3.93	0.32
4.25	4.22	0.37	4.03	0.34	3.32	0.32	4.01	0.35
4.50	4.53	0.34	4.34	0.33	3.61	0.29	4.33	0.33
4.75	4.84	0.34	4.67	0.36	3.89	0.30	4.67	0.36
5.00	5.08	0.31	4.86	0.27	4.13	0.32	4.86	0.27

Table 4.2: Numerical analysis of the velocity estimation visualized in Figure 4.13;

The (light barrier based) measured velocity and the estimates of the three estimation methods are compared in terms of average (μ) and standard deviation (σ). All values are in $\frac{m}{s}$

and the iterative methods show stable and reasonable results in term of the ascending angle θ . The throwing angle basically should be the same for all trajectories thus the standard deviation is a good measure of the quality of the estimation. Here the polynomial method, similarly to the estimation of the velocity, and the RTS smoothing have the lowest variations and (compare Table 4.3). The parameters estimated with the RTS smoother show a systematically lower ascending angle than both other models. Both the velocity and the ascending angle rise during the first part of the estimation here even though 12 runs of smoothing are done and the results of the prior runs are used as initial value for the current one. These values of the polynomial and the iterative method are in line with each other. Regarding the sideways angle ϕ all the estimates

set v		polynomial		RTS		iterative	
$[\frac{m}{s}]$	$[^\circ]$	μ	σ	μ	σ	μ	σ
4.00	θ	54.702	2.452	53.802	1.473	55.011	2.006
	ϕ	-2.177	0.666	-2.189	0.700	-1.658	1.211
4.25	θ	54.669	2.508	53.673	1.645	55.054	1.724
	ϕ	-2.302	0.309	-2.335	0.359	-1.923	0.744
4.50	θ	55.677	1.312	54.532	1.171	56.017	0.999
	ϕ	-2.051	0.407	-2.054	0.392	-1.681	1.070
4.75	θ	56.737	2.196	55.200	0.999	56.549	1.072
	ϕ	-1.745	0.875	-1.965	0.384	-1.575	0.807
5.00	θ	57.030	1.896	54.748	5.364	56.747	0.947
	ϕ	-1.914	0.500	-2.402	1.264	-1.521	1.002
All	θ	55.763	2.304	54.391	2.697	55.875	1.572
	ϕ	-2.038	0.608	-2.189	0.716	-1.671	0.973

Table 4.3: Numerical analysis of the ascending (θ) and sideways angle (ϕ) visualized in Figure 4.14;

The three estimation methods are compared in terms of average (μ) and standard deviation (σ). All values are in $^\circ$

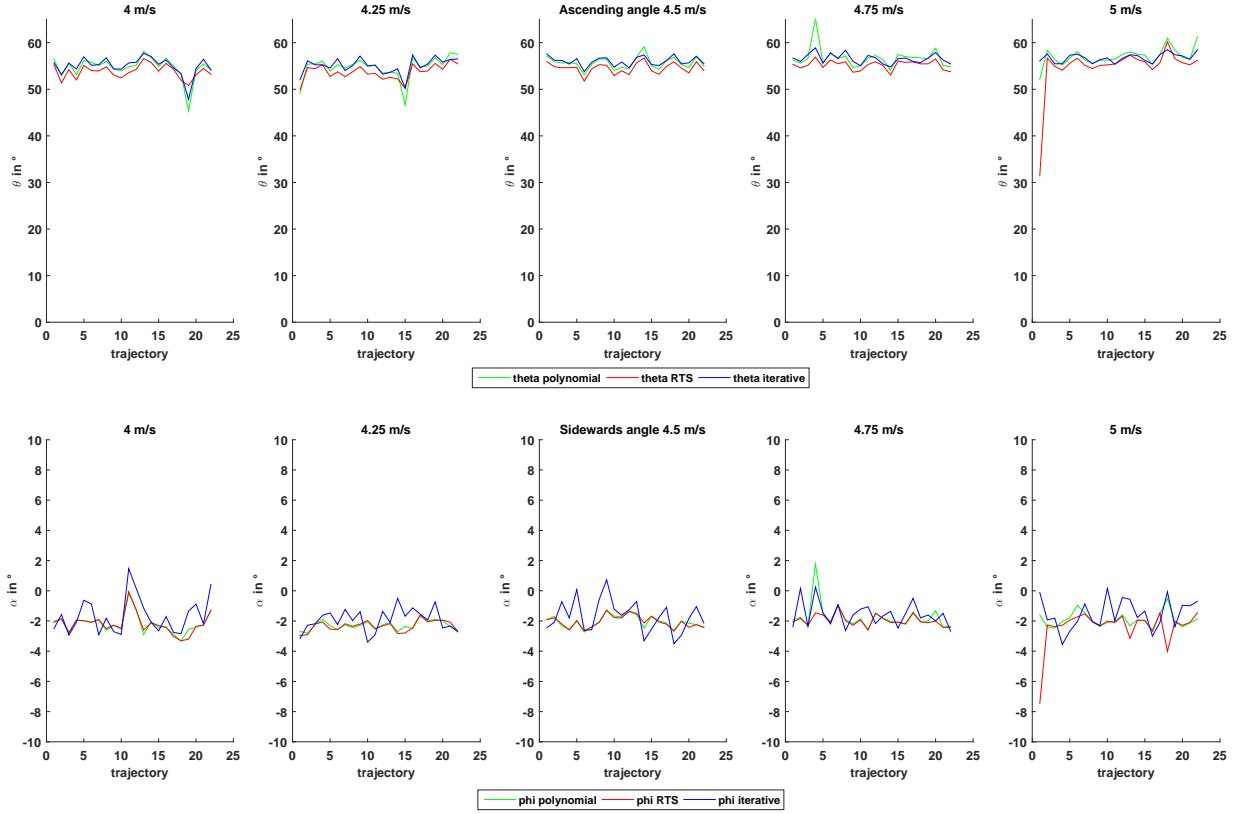


Figure 4.14: Graphical illustration of the initial angles estimation results in comparison;

A line showing the ascending angle (ϕ) of the trajectories related to the five velocities ($4 \frac{m}{s}$, $4.25 \frac{m}{s}$, $4.5 \frac{m}{s}$, $4.75 \frac{m}{s}$, $5 \frac{m}{s}$) is presented in the top row. The bottom row shows the sideways angle (α). The estimations based on the polynomial method describe the measured data best, next best is the iterative method, and the RTS method has clearly the highest systematic difference. The numbers for the statistical analysis are given in Table 4.3 for θ and ϕ

show similar mean values, but their standard deviations differ significantly (compare Table 4.3). The polynomial method again shows lowest variations and thus best stability all methods show similar standard deviations for the ascending angle θ . Regarding the sideways angle ϕ the iterative models shows significantly higher deviations. This has to be seen in context with the further parameter estimation discussed in the next but one paragraph. Overall the polynomial model shows the best behavior also for estimation of the launching angles.

Initial position In addition to the initial velocity parameters of the throw tennis ball also, the starting position has to be estimated. The initial position in the robot world coordinate system (compare Figure 4.1) is the information extracted. Again the estimation is done in the groups of the trajectories with the same target velocity and again for all the trajectories as this initial position should be the same for all of the velocities (it is independent of the velocity as it's acquisition is triggered straight when the ball leaves the second light barrier of the throwing device). The statistical analysis is given in Table 4.4, a graphical illustration is not given here for space reasons but put into the appendix as Figure A.1. Overall again the polynomial method is the one with the lowest deviation and, due to the limited alternative information available, the best method to estimate the initial position. In terms of mean position the iterative method's

set v		polynomial			RTS			iterative		
$\left[\frac{m}{s}\right]$	$[m]$	x	y	z	x	y	z	x	y	z
4.00	μ	-0.535	-2.901	0.384	-0.495	-2.869	0.383	-0.527	-2.903	0.380
	σ	0.002	0.011	0.001	0.006	0.001	0.002	0.009	0.034	0.011
4.25	μ	-0.536	-2.900	0.384	-0.494	-2.868	0.383	-0.528	-2.900	0.385
	σ	0.003	0.008	0.001	0.007	0.002	0.001	0.015	0.019	0.017
4.50	μ	-0.534	-2.899	0.384	-0.489	-2.868	0.383	-0.524	-2.894	0.380
	σ	0.002	0.003	0.001	0.006	0.002	0.001	0.015	0.010	0.016
4.75	μ	-0.534	-2.897	0.384	-0.484	-2.867	0.384	-0.532	-2.898	0.383
	σ	0.005	0.008	0.002	0.006	0.001	0.001	0.015	0.021	0.012
5.00	μ	-0.533	-2.900	0.384	-0.479	-2.877	0.387	-0.515	-2.898	0.379
	σ	0.007	0.023	0.001	0.010	0.057	0.013	0.027	0.022	0.021
All	μ	-0.535	-2.899	0.384	-0.488	-2.870	0.384	-0.525	-2.899	0.382
	σ	0.004	0.012	0.001	0.010	0.026	0.006	0.018	0.022	0.015

Table 4.4: Numerical analysis of the initial position's estimations;

The three estimation methods are compared in terms of average (μ) and standard deviation (σ) of each spatial component. All values are in m

estimation is similar to the polynomial method's again while the Rauch-Tung-Striebel smoother method shows a significantly different position/estimation (again).

Variances of the iterative method's model parameter A remarkable thing occurred during the realization of the iterative method. The quality of the model (in average distance from the estimated positions to the measured positions) could not be reduced below twice the position detection error of the other models. Allowing the method to vary the model's parameters drag coefficient c_d (and implicitly the mass of the tennis ball m) did not improved the quality of the model. The detailed statistical analysis of the drag coefficient after the 3rd repetition is given in Table 4.5, a graphical illustration is not given here for space reasons but in appendix (compare Figure A.2 for the drag coefficient and the overall estimation error). The statistical analysis shows unexpected high variations in the parameters c_d . The drag coefficient shows a significant difference between the largest and the smallest value in relation to the mean value for a target velocity and also the range expressed in the empirical work presented in Section 4.2.1.

Findings

The comparison of the estimated and the measured velocity (and thus also the positions) shows the problem of parameter estimation. The non-physics-based polynomial method shows advantages over both physics-based methods. In terms of parameter estimation based on the Kalman filter (or the derived usage of the Rauch-Tung-Striebel smoother) the initial estimation has larger deviations from the real values as the initial state of the object is highly impacting these initial estimation (compare [Bar10, pg. 116ff]). The iterative estimation method with Monte Carlo simulation shows

set v	c_d	
	$\frac{m}{s}$	
4.00	μ	0.557
	σ	0.136
4.25	μ	0.589
	σ	0.108
4.50	μ	0.503
	σ	0.142
4.75	μ	0.517
	σ	0.117
5.00	μ	0.481
	σ	0.186
All	μ	0.529
	σ	0.143

Table 4.5: The variances of the iterative method’s model parameter drag coefficient c_d

bigger errors by a factor of two compared to the other approaches, even when varying the drag coefficient c_d (and thus implicitly the mass of the tennis ball m). This also shows a problem of predicting with the iterative model based on predefined parameters and renders experience based prediction as more useful. For the Kalman filter based the iterative model is used in most related work, and there fixed parameters are used.

Based on the estimated parameters the optimal launching velocity range at the estimated position with the estimated throwing angles can be calculated. This is done in Section 4.4.2 where the catching movement is planned. Additionally, the information about the estimated trajectory (and thus ball position) can be used to determine the accuracy of the tracking algorithm for tracking the moving objects. This is done in Section 4.2.2.1. Another aspect of this estimation is that the experience database for the practical experiment can be extended with a suitable set of simulated throws in order to speed-up the process of learning. As this artificial learning is based on the iterative model, the variations of the parameters have to be considered, and the content of the database has to replicate these deviations (or even use bigger ones) to allow accurate prediction.

4.3 Soft Catching

The determination of the interception position and the joint positions of the robot for reaching the interception position is determined according to reach the goal of a maximum catching device velocity. The robot’s position maximizes the use of the joints to reach a maximum velocity of the catching tool mounted to the end effector of the robot. When comparing the robot to the model used in Section 3.3 the joints J1 and J2 are used to implement the 2 DoF (Degrees of Freedom) main joint while other joints are used to implement the variable lever length. In addition, the available joints are used to increase the velocity of the catching device in the interception point

and/or ensure that the tangents of the trajectories in the interception point are aligned. Reconsider Figure 3.13 in Section 3.3 for illustration of this simplified model. This simplified catching strategy is the foundation of the implemented catching strategy detailed here.

4.3.1 Constraints

The KUKA LWR 4+ kinematic is inspired by a human arm and offers a similar flexibility for movements. The development of the robot has been done by the German Center for Aerospace and Avionics [HSAS⁺02]. In order to maximize the velocity of the catching device at the interception position and using human inspired catching movement, the robot is mounted on a wall. This mounting and the kinematic allows to mimic the movement and alignment of a human being when catching an object and maximizing the effective lever length for the first joint, thus maximizing the obtainable velocity. In the image of the robot mounted to the wall is presented in Figure 4.3. In order to apply the strategy to the 7 DoF robotic kinematic additional steps are required. The sequential steps are detailed here:

1. Assign Joint3 position to 90 ° to superimpose movements of Joint1, Joint4, and Joint6 for maximum catching velocity
2. Determination of the interception position (normal on trajectory through Joint1 axis)
3. Assign Joint1 position (links between J2/J4 and J4/J6/TCP in one line); compare Figure 4.15
4. Calculation of the distance interception position / Joint1 axis
5. Determine Joint4 position accordingly
6. Calculate Joint2 position to reach interception point

The base for all calculations is the interception position. After this position is determined the position of J1 and J4 can be based on trigonometric calculations. Only after J4's position is known the position of J2 can be calculated. All calculations are executed with each new prediction update, and thus, the interception position and the catching movement is refined. Details for all steps of the process are given in the following paragraphs.

In addition Figure 4.3 gives an overview over the 7 joints of the LWR 4+ and illustrates the names of the joints. The nomenclature used to describe positions and dimensions of the robot and the calculations for masses of inertia or accelerations is given in Section 4.1. The angular velocity limitations of the robot's joints (presented in Table 4.6) restrict the maximum velocity of the end effector of the robot. When using a catching device with a tool center point 0.25 m from the flange and the joints J1, J4 and J6 (compare Figure 4.3) a maximum velocity of 4.25 $\frac{m}{s}$ can be achieved. This is of the same magnitude as the object's velocity at the interception point for optimal throwing at a distance of 2.5 m (compare Section 3.3). In general, the velocity difference of the robot and catching tool has to be minimized but still a differential velocity has to exist, otherwise, the robot will move synchronized away from the ball as the trajectories only match in the catching position. The bigger the velocity's difference, the less accurate the timing has to be and vice versa. As the timely synchronization of the overall system is a challenging task (compare Section 4.5) a reasonable fixed velocity limit is used to ensure soft catching at a high catching rate.

Joint	maximum angular velocity
J1	$112.5 \frac{^\circ}{s}$
J2	$112.5 \frac{^\circ}{s}$
J3	$112.5 \frac{^\circ}{s}$
J4	$112.5 \frac{^\circ}{s}$
J5	$180 \frac{^\circ}{s}$
J6	$112.5 \frac{^\circ}{s}$
J7	$112.5 \frac{^\circ}{s}$

Table 4.6: Specification of the KUKA LWR 4+ maximum joint velocity [KUK10, p. 8]

4.3.2 Interception Point Determination

In order to align the movement vector's direction of the object and the robot the interception position is determined based on the rules discussed in Section 3.3. The interception position is chosen based on the shortest distance between the predicted trajectory of the object and P_{J2} . A visualization showing the robots origin (red dot), P_{J2} (cyan dot), the interception position or P_{TCP} (green dot) and links of a simplified robot model (blue lines) are shown in Figure 4.15.

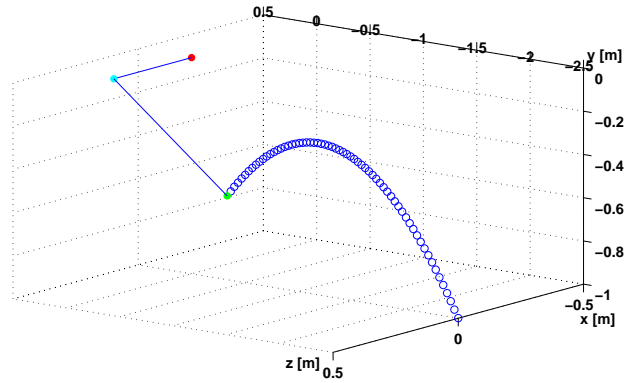


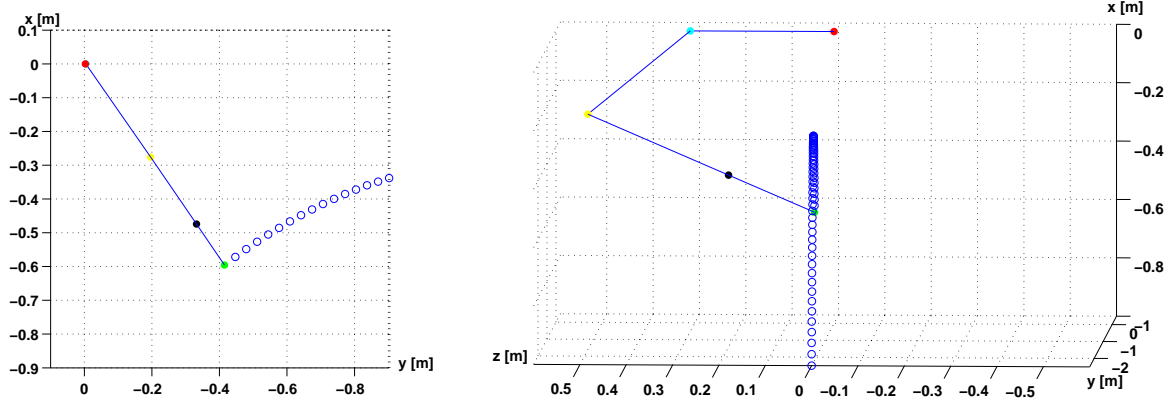
Figure 4.15: Visualization of the robot's catching position for soft catching; red dot: robot origin position, pink dot: P_{J2} , P_I green dot

Joint 1, Joint 3 and Joint 4 Position Determination

Based on the simplified model shown in Figure 4.15 and the known interception position the deflection of two of the seven joints can be determined. For J1, only the x- and y-component of the interception position are required. An illustration is given in Figure 4.16(a).

The equation for calculating ϕ_{J1} is given in Equation 4.6.

$$\phi_{J1} = 180^\circ + \text{atan}\left(\frac{P_{Iy}}{P_{Ix}}\right) \quad (4.6)$$



(a) Visualization of the robot's catching position for soft catching; detailed side view (b) Visualization of the robot's catching position for soft catching; detailed view along throwing direction

Figure 4.16: Robot in catching position with all links considered; robot origin position: red dot, P_{J2} cyan dot, P_{J4} yellow dot, P_{J6} black dot, P_{TCP}/P_I green dot

The offset of 180° is necessary to keep the robot well within the working envelope of $\pm 170^\circ$. In order to reach into the required quadrant the deflection of J2 is must be in the range of 0° to 180° . The detailed determination of J2's deflection is in the final paragraph of this section.

For the computation of ϕ_{J4} the distance d_{J2-I} between the intersection of J1 and J2's axis P_{J2} and the interception position P_I is the relevant parameter. An illustration is given in Figure 4.16(b) where the simplified model prior used is extended with additional joints and links. In order to reach a position as given in Figure 4.16(b) ϕ_{J3} needs to be in it's neutral position (Equation 4.7), otherwise the interception position will not be reached.

$$\phi_{J3} = 0^\circ \quad (4.7)$$

The resulting triangle between the intersection of J1 and J2's axes (P_{J2} , cyan dot), the interception position (P_I , green dot) and the intersection of J3 and J4's axis (P_{J4} , yellow dot) is within one plane and the length of the edges is given by the robot's dimension and the distance between the flange of the robot and the tool center point (TCP) of the catching device. For a catching device with 25 cm distance as used in the implementation (compare Section 4.5) the dimensions of the edges are: $d_{J2-J4} = 0.40$ m for cyan dot - yellow dot; $d_{J4-I} = 0.728$ m for yellow dot - green dot; distance from J1 and J2's axis intersection to interception position (d_{J2-IP}) variable and depending on trajectory for cyan dot - green dot. Based on these known dimensions the calculation of the deflection of J4 is possible based on the rule of cosine (Equation 4.8).

$$\phi_{J4} = \arccos\left(\frac{d_{J2-J4}^2 + d_{J4-TCP}^2 - d_{J2-TCP}^2}{2 * d_{J2-J4} * d_{J4-TCP}}\right) \quad (4.8)$$

Joint 2 Position Determination

With reference to Section 3.3 and the simplified model with a 2 DoF base joint and the variable length arm the position of joint 4 is used to adjust the lever length (distance between P_{J2} and P_I) and the position of joint 2 (in cooperation with joint 1) determines the direction of the lever.

Calculation of ϕ_{J2} is based on two steps with two triangles and the application of the rule of cosine for each. The first triangle (illustrated in Figure 4.16(b)) is the triangle between the robots origin (red dot), the interception position P_I (green dot) and the intersection of J1 and J2's axes (P_{J2} , cyan dot). The angle between the solid line and the broken line is one component of ϕ_{J2} , here named ϕ_{J2a} . The determination is based on the rule of cosine and the equation is given in Equation 4.9

$$\phi_{J2a} = \arccos\left(\frac{d_{J2-TCP}^2 + d_{J2-0}^2 - d_{TCP-0}^2}{2 * d_{J2-TCP} * d_{J2-0}}\right) \quad (4.9)$$

The second triangle is given by the intersection of J1 and J2's axes (P_{J2} , cyan dot), the intersection of J3 and J4's axes (P_{J4} , yellow dot) and the interception position P_I (green dot). Again the rule of cosine is used to determine the second part of ϕ_{J2} , ϕ_{J2b} (Equation 4.10). ϕ_{J2} is the supplementary angle to the sum of ϕ_{J2a} and ϕ_{J2b} (Equation 4.11).

$$\phi_{J2a} = \arccos\left(\frac{d_{J2-TCP}^2 + d_{J2-0}^2 - d_{TCP-0}^2}{2 * d_{J2-TCP} * d_{J2-0}}\right) \quad (4.10)$$

$$\phi_{J2} = \angle 180 - (\phi_{J2a} + \phi_{J2b}) \quad (4.11)$$

Joint 5, Joint 6 and Joint 7 Position Determination

The remaining three joints of the KUKA LWR 4+ have been considered to remain in their neutral position up to now. With this assumption, the interception position is reached by the catching device (or TCP). For reaching the interception position a movement for these three joints is not required.

4.3.3 Velocity Maximization/Synchronization

The resulting interception position of the robot presented in Figure 4.17 allows to use additional joints to joint 1 to increase the velocity of the catching device. These joints are joint 3, joint 4

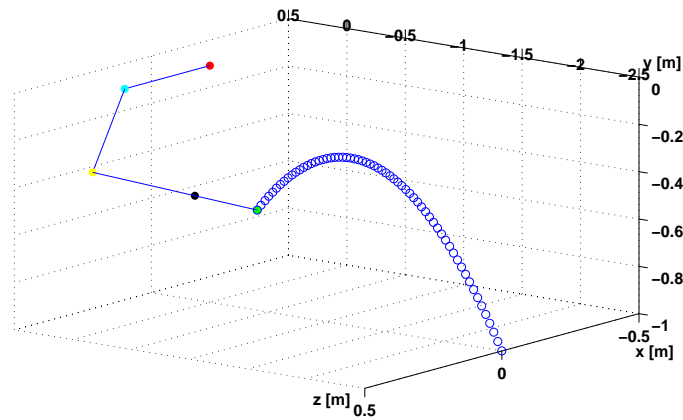


Figure 4.17: Visualization of the robot's catching position for soft catching; robot origin position: red dot, P_{J2} cyan dot, P_{J4} yellow dot, P_{J6} black dot, P_{TCP}/P_I green dot

and joint 6. The proportion of each joint for the overall catching device velocity depends on the robot's position when reaching the interception points. With objects caught close to the robot's joint one axis the proportion of joint three increases (compare human shoulder internal rotation) while the proportion of joint 4 decreases (compare human elbow extension). Regarding the joints 5, joint 6 and joint seven the usage to increase the velocity of the catching device is based on a movement of joint 6. In order to align the tangent of the circular movement with the trajectory of the thrown object at the interception point, joint 5 has to be deflected by 90° or -90° . Depending on the chosen value the movement direction of joint six changes. Joint 7 can be used to rotate the catching device, balance mounting of the catching device or keep the object in the catching device throughout the deceleration movement.

Based on the three positions (waiting position, interception position and stopping position) and the joints used for soft catching the worst case acceleration time to maximum joint velocity for each joint can be calculated (compare Section 3.3). The assumptions for each joint are based on a linear deviation of mass throughout the modules between each joint of the KUKA LWR 4+ and worst case scenario (robot position for maximum moment of inertia for each joint and a catching device with a mass of 0.5 kg and 0.16 m distance between the center of gravity³ and the mounting flange).

As calculation complexity increases from joint 7 to joint one due to the increased number of elements to consider, the estimation starts with joint 6 (joint 7 is not used for velocity synchronization). The mass moved by joint 6 is the catching device (modeled as a cylinder with the length 0.16 m and the mass 0.5 kg) with a displacement from the axis of 0.088 m and the spherical link between joint 6 and joint 7 which is modeled as a solid sphere with (diameter 0.088 m , mass $\frac{16}{7} \text{ kg}$). The resulting mass of inertia J_{J6} is given in Equation 4.12.

$$\begin{aligned} J_{J6} &= \frac{2 * m_{L6} * r_{L6}^2}{5} + \left(\frac{m_{CD} * l_{CD}^2}{3} + m_{CD} * (r_{L6} + 0.01 \text{ m})^2 \right) \\ &= 0.019 \text{ kg} * m^2 \end{aligned} \quad (4.12)$$

The first part corresponds to the spherical link (m_{67}) of the robot between joint 6 and 7, the second part of the catching device and the last part is the application of the Huygens-Steiner theorem⁴ to the catching device due to the distance between the axis of joint 6 and the mounting plane (mounting flange).

Joint 5 is (similar to joint 7) not used for velocity synchronization. For joint four the mass of 3 links (m_{L4} , m_{L5} and m_{L6}) and the catching device have to be moved. The worst case scenario here is that joint 6 is in the neutral position (joint 5 and joint seven only contribute to a longitudinal rotation that has no impact on the mass of inertia). The overall mass of inertia is resulting from the link 4, link 5 (modeled as one long stab), link 6 displaced by the sum of link 4's and link 5's length and the catching device displaced by the sum of the length of links 4, 5 and 6 (Equation 4.13).

$$\begin{aligned} J_{J4} &= \frac{m_{L4+L5} * l_{L4+L5}^2}{3} + \left(\frac{2 * m_{L6} * r_{L6}^2}{5} + m_{L6} * l_{L4+L5}^2 \right) \\ &\quad + \left(\frac{m_{CD} * l_{CD}^2}{3} + m_{CD} * (r_{L6} + l_{L4+L5} + 0.01 \text{ m})^2 \right) \\ &= 0.710 \text{ kg} * m^2 \end{aligned} \quad (4.13)$$

³for sake of simplicity a point-mass is used to model the catching device

⁴also known as parallel axis theorem

In analogy for joint three, the mass of 4 links (m_{L3} , m_{L4} , m_{L5} and m_{L6}) and the catching device have to be moved. The worst case scenario here is that joint 4 is bend by 90 ° and joint 7 is in the neutral position (joint 5 and joint seven only contribute to a longitudinal rotation that has no impact on the mass of inertia). The overall mass of inertia is resulting from the link 3, link 4, link 5 (modeled as one long stab), link 6 displaced by the sum of link 4's and link 5's length and the catching device displaced by the sum of the length of links 4, 5 and 6 (Equation 4.14).

$$J_{J3} = \frac{m_{L3} * r_{L3}^2}{2} + \frac{m_{L4+L5} * l_{L4+L5}^2}{3} + (\frac{2 * m_{L6} * r_{L6}^2}{5} + *m_{L6} * l_{L4+L5}^2) + (\frac{m_{CD} * l_{CD}^2}{3} + m_{CD} * (r_{L6} + l_{L4+L5} + 0.01 m)^2) = 0.726 kg * m^2 \quad (4.14)$$

Finally, also the mass of inertia for joint 1 is determined (J2 has the same maximal torque but a smaller mass of inertia due to the lacking influence of m_{L1}). The worst case scenario is that joint 2 is bend by 90 ° and all other joints are in their neutral position. The overall mass of inertia is resulting from the link 1, link 2, link 3, link 4, link 5 (link 2 to link 5 modeled as one long stab), link 6 displaced by the sum of link 2's to link 5's length and the catching device displaced by the sum of the length of links 2 to 6 (Equation 4.15).

$$J_{J1} = \frac{m_{L1} * r_{L1}^2}{2} + \frac{m_{L2+L3+L4+L5} * l_{L2+L3+L4+L5}^2}{3} + (\frac{2 * m_{L6} * r_{L6}^2}{5} + *m_{L6} * l_{L2+L3+L4+L5}^2) + (\frac{m_{CD} * l_{CD}^2}{3} + m_{CD} * (r_{L6} + l_{L2+L3+L4+L5} + 0.01 m)^2) = 3,747 kg * m^2 \quad (4.15)$$

After the calculation of the masses of inertia for the joints, relevant for the catching movement, the calculation of the maximum acceleration and thus furthermore the maximum acceleration time to the maximum velocity can be done. Using Equation 3.4 and Equation 3.5 on the maximum specified joint torques and joints velocities [KUK10] the maximum acceleration time is calculated. The results are given in Table 4.7. In addition, also the required distance in terms of the rotation angle is given in this table.

joint	J ($kg * m^2$)	M_{max} (Nm)	α_{max} (s^{-2})	ω_{max} ($\frac{1}{s}$)	t_{vmax} (ms)	α_{acc} °
1	3,75	200	53,4	1.96	36.8	4.14
3	0.73	100	137.7	1.96	14.3	1.60
4	0.71	100	140.9	1,96	13.9	1.57
6	0.02	30	1511	1.96	1.3	0.15

Table 4.7: Overview over the hardware restrictions in terms of acceleration time. Based on the mass of inertia the first joint needs the longest time to accelerate to the maximum joint velocity. The required time is 36.8 ms. All other relevant joints are faster by a factor of at least 2

As stated above, the joints 2, 5 and 7 are not used for synchronizing the velocity and decreasing the impact force during catching. In terms of maximum acceleration time, they usage can be calculated in the same way or, by using the related joint with a one digit smaller index the acceleration time can be bounded below this acceleration time. The joint characteristics are the same, but the mass of inertia is smaller due to the lapse of one link.

Based on this strategy the robot's position for reaching interception position can be calculated based on trigonometric functions from the interception position. Simulation and implementation-specific attributes of the catching movement are outlined in the following two sections.

4.4 Simulation

Prior to the implementation with the target platform (described in the following Section 4.5) the proposed algorithms for trajectory prediction and catching movement execution are tested in a simulation environment. This allows the identification of possible problems prior to executing the algorithms on the target platform and tuning the algorithms for better performance.

Due to the real-time connected nature of the vision system with the adaptive Area of Interest (AoI, compare Section 3.1) the vision system is not simulated but only implemented and tuned for the implementation (see details in Section 4.5). The measurement errors of the position detection system (derived in Section 4.2.2.1) are considered throughout the simulation based on an error modeled with standard deviation and the properties derived in the related section in terms of the stereo camera system's coordinate system.

For the other two parts of the whole system (trajectory prediction and catching movement planning and execution), Matlab was used to do the simulation. The level of detail for both parts is described in the following subsections.

4.4.1 Trajectory Prediction

The trajectory prediction based on the proposed algorithms described in Section 3.2 and an additional reference prediction system, based on a physical model, requires the modeling and the basic process, the flight of a tennis ball. The physical description of this process is discussed in Section 4.2.1 and the relevant parameters are extracted in Subsection 4.2.2.2. The database for the simulation and experiment is based on the parameters derived, also considering the variations of the environment parameters necessary to fit the iteratively calculated trajectories accurately.

The level of detail here covers the influences of the gravitational force, the (quadratic) air drag, the modeled deviation of the launching device and the determined acquisition accuracy of the position detection system (compare Subsection 4.2.2.1). Besides the detailed description of the effects influencing the trajectory as well as the position acquisition, the representation of the tennis ball's position is done in Cartesian coordinates in the simulation environment. For visualization reasons circular representations are used.

Overall five different approaches to predict the tennis ball's trajectory are simulated. Two of them are for reference and deal as the base for comparison. These models are the spatial separated model derived in previous work [Pon09, pg. 44ff] and the combination of parameter estimation with a variant of the Kalman filter (in this work here the unscented Kalman filter (UKF) is used due

to the benefits for nonlinear problems) which is used in a number of related works [LRÅJ09] [LRÅJ10] [SC07] [FBH⁺01] [BWH10] [BBW⁺11].

In the following three paragraphs, the specific attributes of the five prediction approaches in the simulation environment will be described in detail. The first discussed model will be the holistic associative memory prediction (compare Subsection 3.2.1) followed by the progressive associative memory prediction (compare Subsection 3.2.2). It is noteworthy that the prediction database was created without adding the error due to the position detection system, determined in Section 4.2.2.1. This is equivalent to smoothing the trajectories before storing them in the database. Also, all the positions available for the current flight are processed with the RTS smoothing algorithm. This had an insignificant influence on the prediction accuracy of the holistic model, minor influence on the progressive model without additional discretization but significant influence on the progressive prediction with additional discretization. The prediction results got significantly more stable with the usage of filtering. Finally the two reference model, the analytical model-based prediction [Pon09, pg. 44] and the UKF-iterative model will be discussed.

Holistic Associative Memory Prediction

The base for the holistic associative memory prediction is the experience dataset. This dataset contains the position information of each previously measured trajectories. In the case of the simulation, this database is initialized with 2048 trajectories with the parameters given in the preface of this subsection. The pure information containing the positions over time of each trajectory are fed to the prediction algorithm. A sequential search compares the stored position of each trajectory to the currently measured positions and calculates the individual distances' sum for each trajectory. In order to cope with minor deviations of the timing, a shift of up to 5 positions in both directions is possible. The number of 5 was chosen based on simulations runs where no shift was the most frequent used option and shifts between -2 and 2 rarely occurred.

For the simulation, no real time constraints are considered. Still the computation times of all the prediction models are acquired and briefly discussed in Section 5.1. Here it has to be noted that the computation times in the simulation environment (Matlab) are not comparable with an optimized implementation in C/C++. Still the can be used as a rough guideline for the computational demands.

Variations of the model here occur in terms of the size of the database, where random subsets of 1024, 512 and 256 trajectories from the 2048 trajectories of the main database are chosen. These samples are stable for a set of predictions. Another variation parameter is the number of recent positions/position-changes that are compared. The numbers here are all, 64, 32, 16 and 8.

Progressive Associative Memory Prediction

In addition to the straight comparison of the current positions with the positions of the trajectories stored in the database also, the comparison of the change between two position acquisitions is used. Here the dependency on the absolute position is avoided. The search algorithm is similar to the one used for the absolute positions.

As an extension of the mentioned prediction, the progressive prediction with additional discretization requires preprocessing of the position data from the trajectories of the experience dataset. Main aspects are described in Subsection 3.2.2. The main steps are discretization of the position data, determination of the velocities, transformation to indexes for more efficient searching,

storage as a double linked list for keeping the information about the connection of the individual velocities to build up a whole trajectory. Memory management for this kind of storage is also required as the deviation of the velocities might lead to some indexes and thus data fields that significantly more trajectories have than others. Keeping track of the usage of the allocated memory (or the maximum size of an array) is required. Similarly, to the holistic associative memory prediction the prediction is based on the most similar trajectory from the experience dataset. The most recent measured position of the current trajectory is combined with the velocity data from the experience dataset to acquire the predicted position information.

Within the prediction process, the same steps have to be done for the current velocity as for the whole dataset (discretization, velocities, indexes, linked list). For the most current velocity, the indexes are used to access all trajectories with that used the same velocity. No special search is required, only the determination of the indexes leads to the related velocities in the database. The storage as a linked list allows to compare the recent history with variable size of the previously acquired trajectories from the database to the current trajectory.

Due to the fact that the integral information is kept (compare Subsection 3.2.2 a reduction of the numbers of positions for the comparison with the recent history can be done (thus, the computation expenses are reduced). The impact of various dense considerations (e. g. for 32 recent sample the first, the last and the 30, 15, 8, 4, 2, 1, 0 samples within are considered). In addition to this variation also the discretization resolution is varied with two fundamental ideas: the first is to have the same discretization resolution for all three directions while the second considers the accuracy of the position detection system. In order to have a good comparison, the volume of one voxel is normalized, and additionally the dimensions are varied in the steps of $\frac{1}{4}$, 1, and 4 giving a good overview of the discretization interval's impact on the prediction accuracy and the computation time.

The search is initiated at the first and last discretized velocity. All trajectories containing the same velocity are compared then to the current trajectory, and the sum of the Euclidean distance is used to find the best fitting trajectory. Because also the first and the last velocity can differ from the currently measured one the next step is to compare all velocities with neighboring velocities. This search is stopped when the distance at the first/last velocity is bigger than two times the average error per considered position of the best-found trajectory. Figure 4.18 illustrates the accumulated sequence of neighbors that are considered.

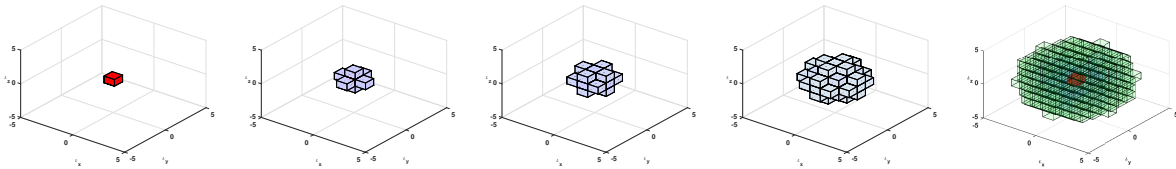


Figure 4.18: Visualization of the discretized progressive prediction neighborhood search

Analytical Model-based Prediction

The first reference model for the evaluation of the experience based prediction is the analytical model derived in [Pon09, pg. 44ff]. The model is based on the physics of a throw including the influences of gravity and air drag. The separated consideration of the movements in the three

spatial directions allows to obtain closed solutions to describe the movement. This approach was found to be the most accurate and most stable one in previous work [Pon09, pg. 59ff].

Each of the measurements can be weighted individually for the least squares fitting of the analytical solutions. A relation between the accuracy of the vision system and the weighting factors for the fit is logical. Based on the results of the camera system's accuracy determination (compare Section 4.2.2.1) the accuracy of the vision system is close to constant in the relevant region of the object's distance thus, the usage of equal weights for the individually measured positions is used.

The second reference model is based on a UKF parameter estimation and the prediction of the future trajectory based on the iterative model for the tennis ball's flight. This approach is discussed in detail in the related work section (compare Section 2.2.2).

4.4.2 Catching Movement Simulation

The strategy for soft catching discussed in Section 4.3 uses the robot's kinematic in order to achieve a small velocity difference between the arriving ball and the robot's catching device. Based on the dimensions of the links and the joint restrictions of the robot (compare Table 4.8) the catching area and the corresponding achievable velocity of the throwing device in the catching point can be calculated. The calculation of the velocity is based on the movement of the joints J1, J3 and J6

Joint	angular range
J1	$\pm 170^\circ$
J2	$\pm 120^\circ$
J3	$\pm 170^\circ$
J4	$\pm 120^\circ$
J5	$\pm 170^\circ$
J6	$\pm 120^\circ$
J7	$\pm 170^\circ$

Table 4.8: Specification of the KUKA LWR 4+ maximum joint movement range [KUK10, p. 8]

(compare Subsection 4.3.3). A visualization of the achievable velocities in the relevant part of the robot's workspace is given in Figure 4.21. The overall volume of the relevant catching area of the robot is 0.95 m^3 . The distance of the tool center point (TCP) to the mounting flange of the robot is based on the dimensions of the catching tool used for the experiments (compare Section 4.5) and has the value of 0.2 m . This dimension is relevant for the achievable catching velocity as it is directly influencing the length of the effective lever for the rotational movements. Figure 4.20 is giving a good overview over the achievable velocities in more detail. Only the catching velocities above a certain threshold ($1.5 \frac{\text{m}}{\text{s}}$, $2.5 \frac{\text{m}}{\text{s}}$, $3 \frac{\text{m}}{\text{s}}$ and $3.25 \frac{\text{m}}{\text{s}}$) are illustrated, thus showing the catching area's dependency on the velocity threshold. The area for catching velocities higher than $3 \frac{\text{m}}{\text{s}}$ with 0.35 m^3 is rather small compared to the unconstrained catching area of 0.95 m^3 . On the other hand catching velocity of $2.5 \frac{\text{m}}{\text{s}}$ reduces the impact energy of a ball arriving with $4 \frac{\text{m}}{\text{s}}$ to $\approx 14 \%$ and the related catching area has the size of 0.76 m^3 , thus the benefit of soft catching is usable in a relatively large area.

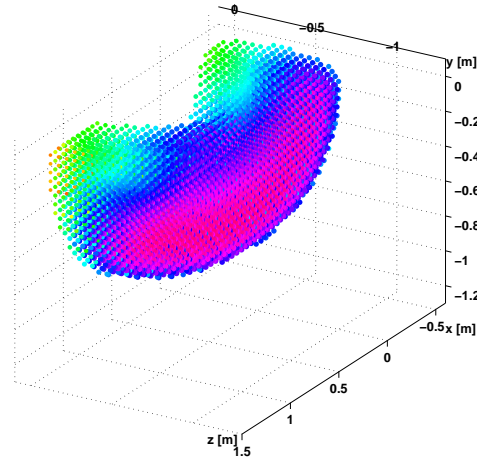


Figure 4.19: Visualization of the robot's catching area with indication of the achieved velocity (1 to 3,5 $\frac{m}{s}$ increasing from yellow via green and blue to red); the overall catching area covers an area of 0.95 m^3

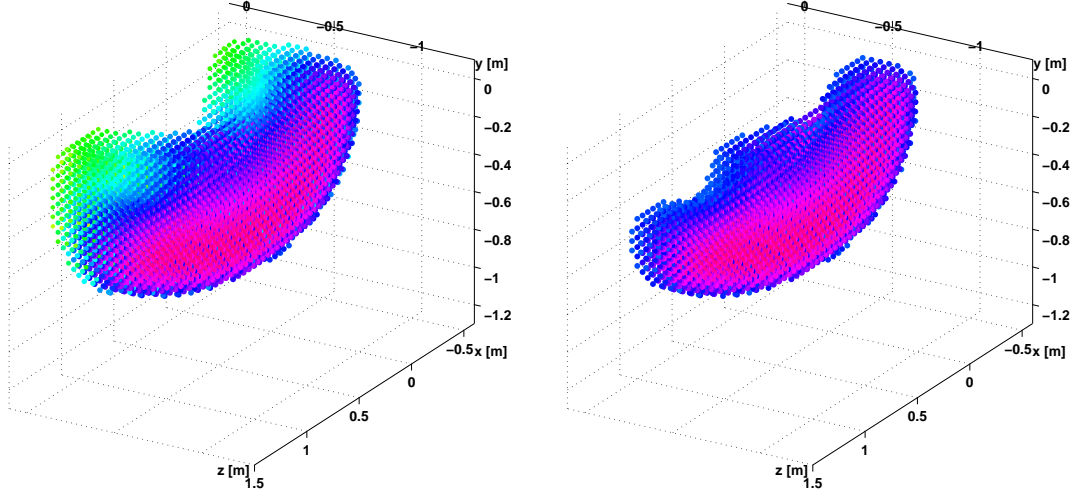
Optimal Initial Velocity

The knowledge about the catching area of the robot derived in the previous paragraphs and the knowledge about the throwing device's deviation allows determining the optimal average initial velocity of the ball to maximize the catching rate (based on the reachability of the robot). The information used for this determination is based on the polynomial parameter estimation done in Subsection 4.2.2.2. The required additional information for this calculation is the relation between the camera's coordinate systems (CCS) and the robot's world coordinate system (RWCS). This information is acquired based on calibration sheets mounted to the robot's mounting flange and the usage of them for the camera calibration. Based on this information the rotation and translation matrices between each of the CCS and the RWCS can be calculated. These matrices are given below:

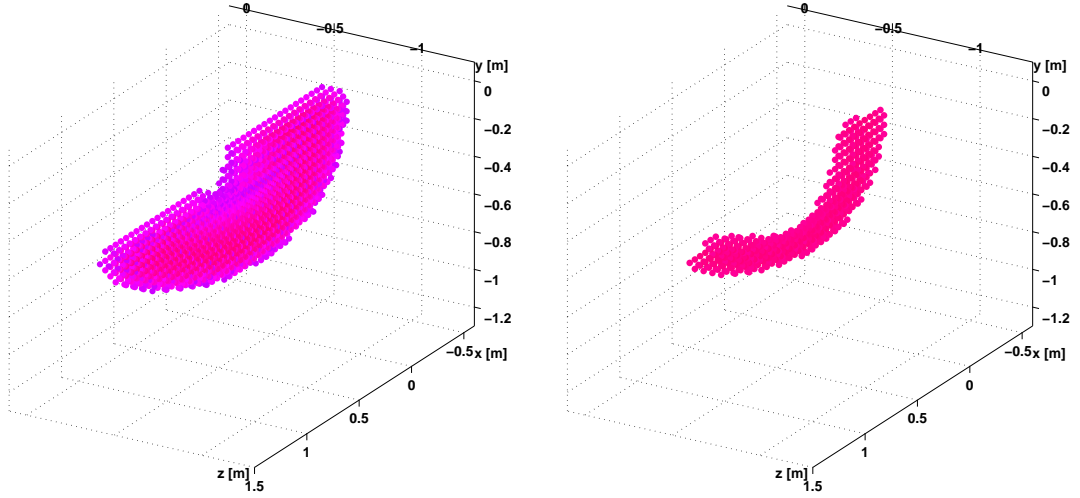
$$R = \begin{Bmatrix} 0.9999 & 0.0109 & 0.0012 \\ -0.0015 & 0.0230 & 0.9997 \\ 0.0109 & -0.9996 & 0.0231 \end{Bmatrix}$$

$$T = \begin{Bmatrix} -327.59 \\ -3106.83 \\ 428.25 \end{Bmatrix}$$

The additional parameters for the simulation are found in Subsection 4.2.2.2 and based on the polynomial estimation. The following parameters and their deviations are considered: initial position \mathbf{p} , the initial velocity \mathbf{v} (this parameter's mean is varied on purpose), the throwing angles θ and ϕ , the tennis ball's mass m and the drag coefficient c_d . 2000 throws at velocities between 3.8 $\frac{m}{s}$ to 4.8 $\frac{m}{s}$ with a step size of 0.02 $\frac{m}{s}$ are calculated and the achievable catching rate, when considering all the variances in the previously mentioned parameters, are derived based on the soft catching strategy. The resulting catching rates in relation to the initial velocity are shown in Figure 4.21. The graph presented clearly shows that the deviation of the throwing device is too big to achieve even a theoretical catching rate of 100 %. The maximum catching rate is achieved with a mean initial velocity of 5.0575 $\frac{m}{s}$ and equals 91.56 %. Only in these cases, the robot's catching area is traveled by the ball in a way that the soft catching strategy can be executed.



(a) an illustration of catching points where a catching velocity of at least $1.5 \frac{m}{s}$ can be achieved (b) an illustration of catching points where a catching velocity of at least $2.5 \frac{m}{s}$ can be achieved



(c) an illustration of catching points where a catching velocity of at least $3 \frac{m}{s}$ can be achieved (d) an illustration of catching points where a catching velocity of at least $3.25 \frac{m}{s}$ can be achieved

Figure 4.20: Visualization of the robot's catching area with indication of the achieved velocity (1 to $3.5 \frac{m}{s}$ increasing from yellow via green and blue to red);

The shape similar to a hollow cylinder is caused by the combination of the robot's link configuration and the catching algorithm. The catching area covers an area for $1.5 \frac{m}{s}$, $2.5 \frac{m}{s}$, $3 \frac{m}{s}$ and $3.25 \frac{m}{s}$ are $0.95 m^3$, $0.76 m^3$, $0.35 m^3$ and $0.07 m^3$. Remarkably the catching area for velocities higher than $2.5 \frac{m}{s}$ is 80 % of the whole catching area.

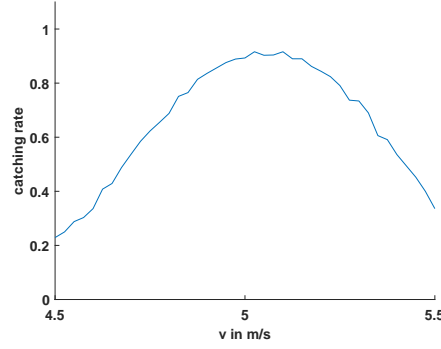


Figure 4.21: Graphical illustration of the theoretically achievable catching rate's dependency on the mean velocity of the ball thrown by the throwing device. The maximum is achieved at a velocity of $5.06 \frac{m}{s}$ with 91.6 %.

4.5 Implementation

This section is devoted to the experiment related implementation details and thus covers aspects that extending the relevant aspects of the simulation. The covered aspects are the high-speed image acquisition, the image processing, trajectory prediction, mastering of the robot, the catching devices for the robot, velocity synchronization of the catching movement and the timely synchronization of the whole system's components. Figure 4.22 shows an architectural overview of the

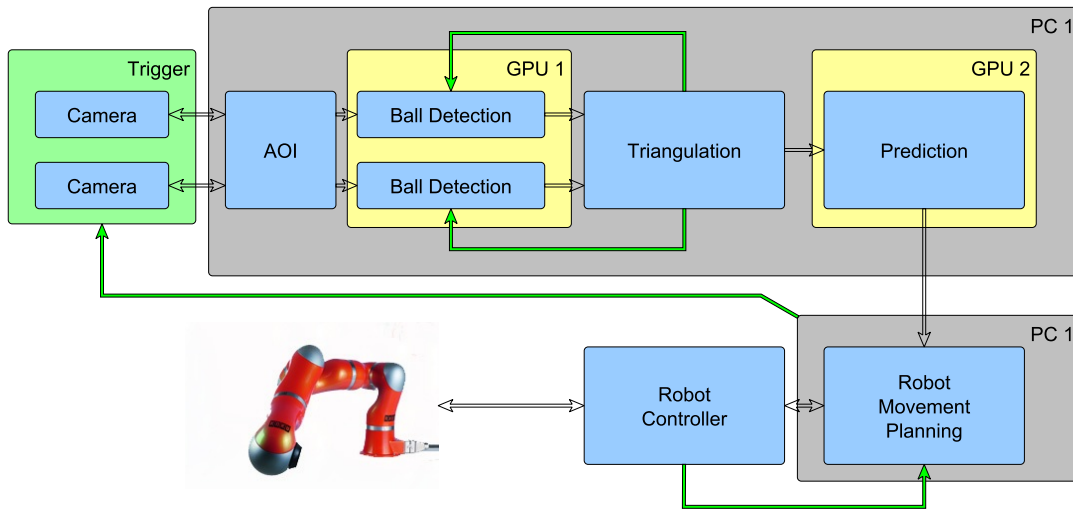


Figure 4.22: System architecture for the catching experiments

whole system. It consists of the two cameras interfaced with one PC via USB3.0. The position detection algorithms run on the CPU and GPU. The CPU takes care of low-level prediction and AoI shift while the GPU algorithms extract the center of the ball more accurately in the images based on the RANSAC algorithm. This information is used for the triangulation in the forward processing and as feedback for the expected ball diameter in the next frame for the ball detection (green arrows). For each new frame, a new prediction is done on the second GPU of PC1, and the result is sent via a network connection to the second PC that is a Linux real-time system with Xenomai. This systems handles the real-time communication with the robot controller and also

triggers the cameras based on the (deterministic) time-triggered communication with the robot. This allows to synchronize the camera's images with the time base of the robot. All aspects are elaborated in the following paragraphs.

Image Acquisition

The first part of the processing chain of the whole system is the image acquisition. The implemented system aims to incorporate the proposed aspects discussed in Section 3.1 with a moving Area of Interest (AoI) following the flying ball. The proposal also considers the size of the AoI to be adapted based on the distance of the object to the camera. Due to technical limitations of the used cameras (IDS Imaging UI-3370CP-M-GL) some of the proposed ideas could not be implemented, or software solutions instead of low-level hardware solutions had to be used. The main goal, to increase the frame rate of the acquired image information while reducing the amount of image information to extract the object's position was still kept as the main goal.

In terms of the AoI movement and size change the functionality of the camera and driver did not allow to change the AoI between two frames with reliable results. In some cases still the image information from the old position was transferred but presented at the more recent position. This (documented) behavior would lead to big measurement errors in the position detection, and thus, another way to implement the AoI movement had to be found. The solution implemented is to use the cameras in rotated positions where the lines of the cameras are approximately in line with the gravitational direction. This gives benefits regarding the achievable frame rate of the camera as this is largely dependent on the number of lines read out. An area of 2048 *by* 800 *pixels* is used for each of the two cameras. Within this sub-image, a smaller software-AoI is moved with a size of 300 *by* 300 *pixels*. This software-AoI is moved based on an initial preset movement and a linear prediction of the movement between two consecutive frames for the next AoI's position. The overall system's functionality is close to the proposed functionality in Section 3.1 but the achievable frame rate is limited to 110 *fps* in comparison to an estimated frame rate of ≈ 250 *fps*. The future hardware implementation of the AoI movement depends on changes in the camera driver. Based on information from the camera's manufacturer the time frame for this changes cannot be foreseen now.

Image Processing

The main parts of the image processing are done on a general purpose graphics processing unit. Roughly this contains the following steps (optional) background subtraction with a static background model, noise filtering, Canny edge detection and Hough circle transform or Random Sample Consensus (RANSAC) circle detection. The implementation of these main functions and additional steps like image conversion or similar things is covered in detail in a related master thesis [G15]. The additional step of triangulation is done on the central processing unit (CPU) because of the lack of performance improvement on a parallel computation platform like a GPGPU. The implementation of the image processing is combined with the software for image acquisition. The 300 *by* 300 *pixel* images and the additional information about the location of this AoI within the whole image are transferred to the GPGPU, and the processing steps are executed. In terms of software, the CUDA programming language is used, and the hardware platform is a Geforce 970 (which is an update compared to the GPGPU used in [G15]). This master thesis deals with the performance optimization of the algorithms for the timely requirements. The used combination of image preprocessing (background subtraction) and circle detection (Hough or RANSAC) is done

based on the findings on the accuracy of the system (compare Subsection 4.2.2.1) and thus the background subtraction with RANSAC circle detection is used for stability reasons and calculation time advantages.

Trajectory Prediction

Six variants for the trajectory prediction are proposed (compare Subsection 3.2) where the most suitable one, based on the simulation results is implemented, using the advantages of GPGPU's highly parallel architecture. The foundations of these prediction algorithms are given in related work [G15]. The initial variants given there are extended with the variants using only a window of recent positions/velocities for the comparison with the experience database. Additional details to the parameters of these trajectory prediction variants are given in the related results Section 5.2.

Robot Mastering

The aspect of robot mastering was already covered in Subsection 4.2.2.1 when dealing with the accuracy of the position detection system. The problems mentioned there with the inaccurate mastering causing a distance between the calculated (Cartesian) position and the real (Cartesian) position are also existing during catching. In terms of the position detection accuracy analysis, the single digit *mm* errors of the robot's position execution are in the same order of magnitude as the expected (and in Section 4.2.2.1 determined) errors of the position detection system. In terms of robotic catching, the errors of the prediction are considered to be one order of magnitude above these errors, additionally, the general position of the robot is similar for different catching positions, and thus, the impact of these errors is considered to be minimal. Additionally, the calibration of the robot to the camera system uses positions that are similar to the catching positions in order to minimize this error. Improving the mastering of the robot above the best effort level available would require extensive investigations on the mechanical characteristics of the robot and the comparison with the specification and thus is not done here.

Soft Catching and Velocity Synchronization

In order to enable soft catching with the KUKA LWR 4+ industrial robot, the determination of the optimal interception position is necessary. The implementation details of the basic strategy, developed in Section 3.3 and applied to the KUKA LWR 4+, are given here. Based on the mechanical estimation done in Subsection 4.3.3 the Reflexx Motion Library's (RML) (introduced in Section 3.3) parameter of the catching movement are set to keep the robot's velocity, acceleration, and overall stress within 90 % of the specified values. During the flight of the ball the robot's waiting position is continuously updated based on updated interception positions, and thus, the robot is moving, until the time for executing the final movement for soft catching is equal to the predicted remaining time for the ball to fly. This final catching movement is extended with a movement to decelerate the robot and the catching device while keeping the ball within the catching device. In order to gain additional information about the precision of the prediction in the experiment, three different sizes of catching devices are used. These catching devices are presented in Figure 4.23.

While the distance between the catching device's ring's center is 0.2 *m* for all of them, the inner diameter of the ring is 80 *mm*, 100 *mm* and 110 *mm*. Using these different catching devices allows



Figure 4.23: Catching devices with the same distance of the center of the ring to the mounting flange and different ring diameters (80 mm, 100 mm and 110 mm)

gaining additional information about the deviation of the prediction because the success rate of catching depends on the tolerances of the catching device.

Timely Synchronization

The whole system's function is within the real time domain which means that besides the result of certain calculations also the time the results is available is critical. A simple illustration is that the interception position has to be known prior (even a certain time before) the ball actually passes it. Otherwise, the information is worthless, and the ball will not be caught. This timely requirement also raises the need for a relation between the individual system's time to each other or timely synchronization. The robot controller is communicating with the remote host in a configurable cycle time periodically. Due to the absence of any external clock output of the Kuka Robot Controller (KRC), this communication is used to establish a global clock. The reception of a UDP packet from the KRC, which is configured to happen each millisecond, is connected to a kernel module in the Linux real-time kernel. A statistic module is keeping track of the reception and a clock signal for triggering the cameras is derived (compare the master thesis [Hubss]). This clock signal is, on the other hand, triggered based on the second light barrier of the throwing device in order to limit the triggering of the cameras to times when a ball is thrown. The times when the cameras are triggered are stored and thus, when information derived from these acquired images are transferred to the remote host, the time when this information was valid can be restored. This allows to move the robot to the catching position timely accurately. A statistical analysis of this trigger behavior is given in [Hubss]. Based on a predefined movement where the robot passes a light barrier the timely behavior of this architecture is examined. The latency of the processing delay is compensated but a jitter in the magnitude of $\approx 300\mu s$ still exists. This jitter can be neglected in terms of the camera's exposure time of 1 ms and the time of 9 ms between two trigger events.

5 RESULTS

To test the proposed algorithms for the trajectory prediction both a simulation and a practical experiment with an industrial robot is done. The simulation considers two state of the art models for reference, the model based on unscented Kalman filter (UKF) parameter estimation and iterative flight path calculation (compare Section 4.4.1 and Section 2.2.2) and the spatially separated model based on least square fitting of a simplified physical mode (compare Section 4.4.1 and previous work [Pon09, pg. 44ff]). In addition, the results of the holistic associative prediction are discussed in detail with two variants: one based on position association while the other based on position change association. Here the influence of the database's size and the consideration of a recent history (a certain number of position prior to the current measurement) and the influences on the prediction accuracy are examined. Furthermore, the progressive associative predictions results are presented and discussed. Again the influence of the main parameters (database size, recent history size, discretization interval shape, and size) are presented and reviewed. For all the simulation results a set of 20 throws acquired with the position detection system is used, and the main information for the comparison is the 99 % bound of the catching position prediction error. Whenever the values of the error are discussed without any special note, this is the meaning of the numbers discussed. The best suitable setup is then used for the practical experiments with the KUKA LWR 4+ robot. Here series of catching experiments of 20 throws are done with three differently sized catching devices (compare Section 2.2.2.1) and thus, besides the binary information about the catching success rate, also a dependency on the diameter of the catching device can be used to quantify the accuracy of the prediction system.

5.1 Simulation

For the simulation results, the real process is modeled accurately in order to give valid results for the practical experiment. The influences considered/modeled are: the accuracy of the camera system (compare Section 3.1), the variation of the throwing device and the properties of the prediction algorithm. In order to have a good measure of the accuracy which can be easily interpreted the (optimal) catching position of the simulated flight is compared with the catching position of the predicted flight. Here a problem arises due to the discrete positions with deviations from a smooth trajectory due to the acquisition with the position detection system. This is valid for all prediction methods (the two reference models, the holistic and the progressive prediction model): A small deviation in the predicted trajectory can lead to the selection of an earlier or later position in the trajectory for catching. This immediately leads to an error in the magnitude

of the position change within one frame acquisition interval of the (simulated) camera system. For the spatially separated reference prediction models this is only valid for the reference catching position, but the final results are the same. Figure 5.1 illustrates one of these occurrences. The

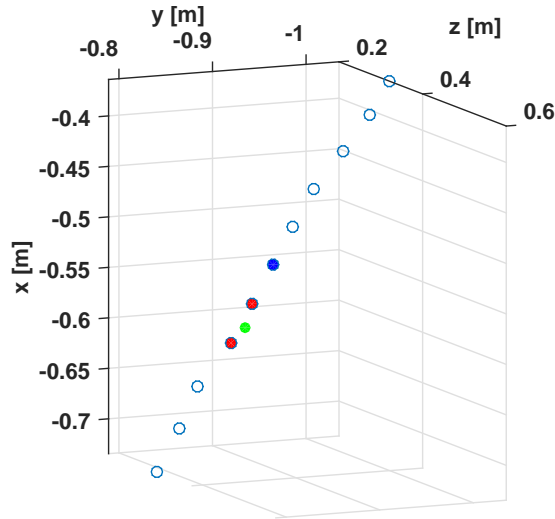


Figure 5.1: Illustration of the wrongly detected error

The reference catching position (green) is shifted in space and time, and thus the relevant error for the catch of the ball is smaller than the calculated distance between this positions and the blue point (predicted catching position). If, instead, the two closest position (red) are used and the normal distance to the connection of these points is used as error measure the quantitative information on the prediction error is better

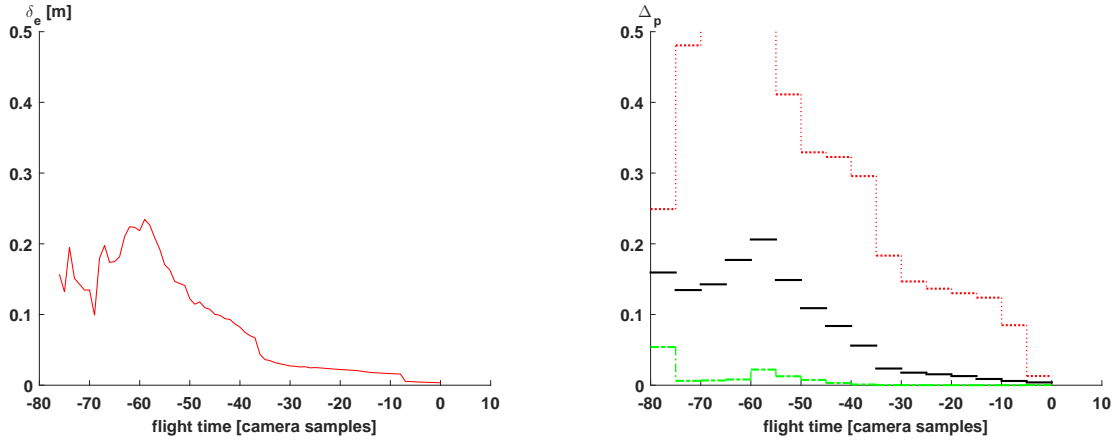
"wrong" position still lies on the flight trajectory and based on the related change of the catching time the position is well suitable again. In order to neglect this false error from the statistical evaluation the neighborhood of the predicted catching position (up to 5 positions prior and past) are compared to the reference catching position and the error is take from the two with the lowest distance. The normal distance to the connection of these two points is calculated and used as a measure for the prediction error. The important information for the practical experiment is how accurately the catching position is predicted over time. The main diagrams in the following sections indicate this relation. The magnitude of the error is used for the statistical analysis, where the 99 % bound of the Rayleigh distribution modeled error is given as main information. For each of these diagrams the catching time is shifted to the normalized position detection index 86 (thus, the detected position at this time is the catching position).

5.1.1 Physics-based Prediction Results

Both reference models are used in the simulation to deal as comparison for the proposed prediction algorithms. Here the mean catching position prediction error is given, and the statistical analysis is illustrated as well, showing the lower and upper bound of the 99 % error band. The variation of model specific parameters is not done in this case. The UKF-iterative model's initial parameters are set based on the mean settings of multiple optimization runs.

5.1.1.1 Spatial Separated Model Results

The spatially separated model is based on the consideration of independent movements along the three dimensions. For each movement a closed solution is found (compare [Pon09, pg. 44ff]) and depending on the dimension three or four (for the dimension where gravity is influencing the movement) parameters have to be estimated in order to describe the movement. This estimation is done based on least square fitting of the functions. The prediction results of this approach are shown in Figure 5.2.



(a) Mean error of the spatially separated model prediction for the catching position of 20 throws over time; Catching time is normalized to the time 0
 (b) 99 % confidence bounds (Weibull distribution) of the mean catching position prediction error; Catching time is normalized to the time 0; red upper bound, green lower bound, black mean

Figure 5.2: Illustration of the spatial separated prediction error over time

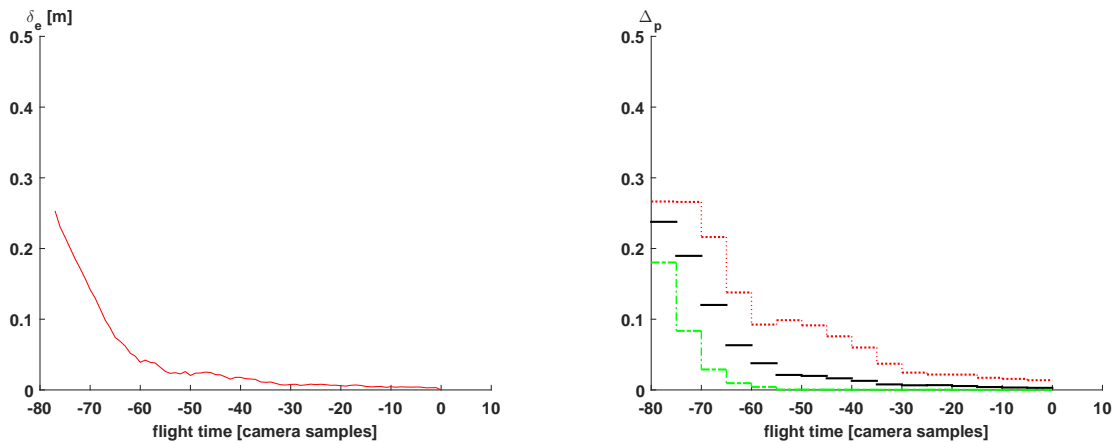
Catching time is normalized to the 86 time instant. Figure 5.2(a) shows the development of the mean error for 20 throws over time, Figure 5.2(b) shows the (Weibull) distribution with the 99 % confidence bounds; overall the prediction error's band is relatively big more than 35 frames before the catching time, after that the significantly decreases and is stable with an upper bound of approximately 5 cm

Figure 5.2(a) shows unexpected behavior in between 80 and 40 frames before catching where the mean prediction error is decreasing, increasing and decreasing again. This behavior is due to the low numbers of the successful fits/predictions in the early flight phase. The spatially separated prediction is getting stable only in the area of 45 frames before the catching time. Also, the step 10 frames prior to catching is caused by one flight that can only be predicted accurately in the very final phase.

5.1.1.2 Unscented Kalman Filter Estimated & Iterative Model Results

The combination of the (unscented) Kalman filter estimation and the iterative model trajectory calculation for prediction is the most frequently used approach for research work on robotic catching (compare Section 2.2.2). The filter estimates the measurement errors of the position observation system and the state of the flying ball, based on a physical model. The prediction errors for the 20 throws are shown in Figure 5.3.

Figure 5.3(a) illustrates the mean error over time. The smooth line decreases significantly until 60 frames before the catch (after that the error is below 5 cm). Then the mean error continuously



(a) Mean catching position prediction error of the EKF-iterative model over time; Catching time is normalized to the time 0 (b) Mean catching position prediction error of the EKF-iterative model over time; Catching time is normalized to the time 0

Figure 5.3: Illustration of the spatial separated prediction error over time

Catching time is normalized to the time 0. Figure 5.2(a) shows the development of the mean error for 20 throws over time, Figure 5.2(b) shows the (Weibull) distribution with the 99 % confidence bounds

decreases furthermore from there to 0 cm. In terms of the error band the upper band (shown in Figure 5.3(b)) of the errors is below 5 cm from the bin of 31 to 35 time steps before the catching time and below 2 cm from the bin of 11 to 15 time steps before.

5.1.2 Bio-inspired Prediction Results

The bio-inspired prediction is tested with the same set of 20 flights as the analytical reference models in the simulation. For both approaches, a number of parameters exist. The influences of variations in these parameters (e. g. number of flights in the database) are investigated based on the variation of these parameters. In order to provide a good overview of the influences, the individual impacts are carved out based on the prediction errors (similar to the analysis of the reference algorithms). The database with the flights used for the prediction is calculated based on the determined launching parameters of 82 flights where an iteratively calculated trajectory is fitted based on Monte Carlo Simulation with 4000 samples and 4 iterations, each using the best previously found trajectory as a base and reducing the variations to one-third.

5.1.2.1 Holistic Associative Prediction Results

In the holistic prediction, the stored positions of flight are compared with the current flight and the future trajectory of the current throw is predicted based on a similar trajectory in terms of positions. The approaches' prediction results are discussed in the following paragraphs.

One main parameter of the position-based prediction is the size of the database for prediction. This parameter has an influence on the prediction accuracy and the time necessary for calculating the prediction (even in the case of calculating it on a parallel architecture unless all trajectories can be processed in parallel). The first analysis of the prediction results is targeting this parameter. In addition also, the influence of the number of positions considered for the comparison is

analyzed. Again this parameter has an influence on the processing time (as more positions have to be compared) and it is assumed that also the prediction accuracy depends on it (as individual measurement errors of the position detection system are less important).

Database Size The illustration of the holistic prediction's results with variable database size is compared in Figure 5.5. Each of the first five diagrams is devoted to one recent history size

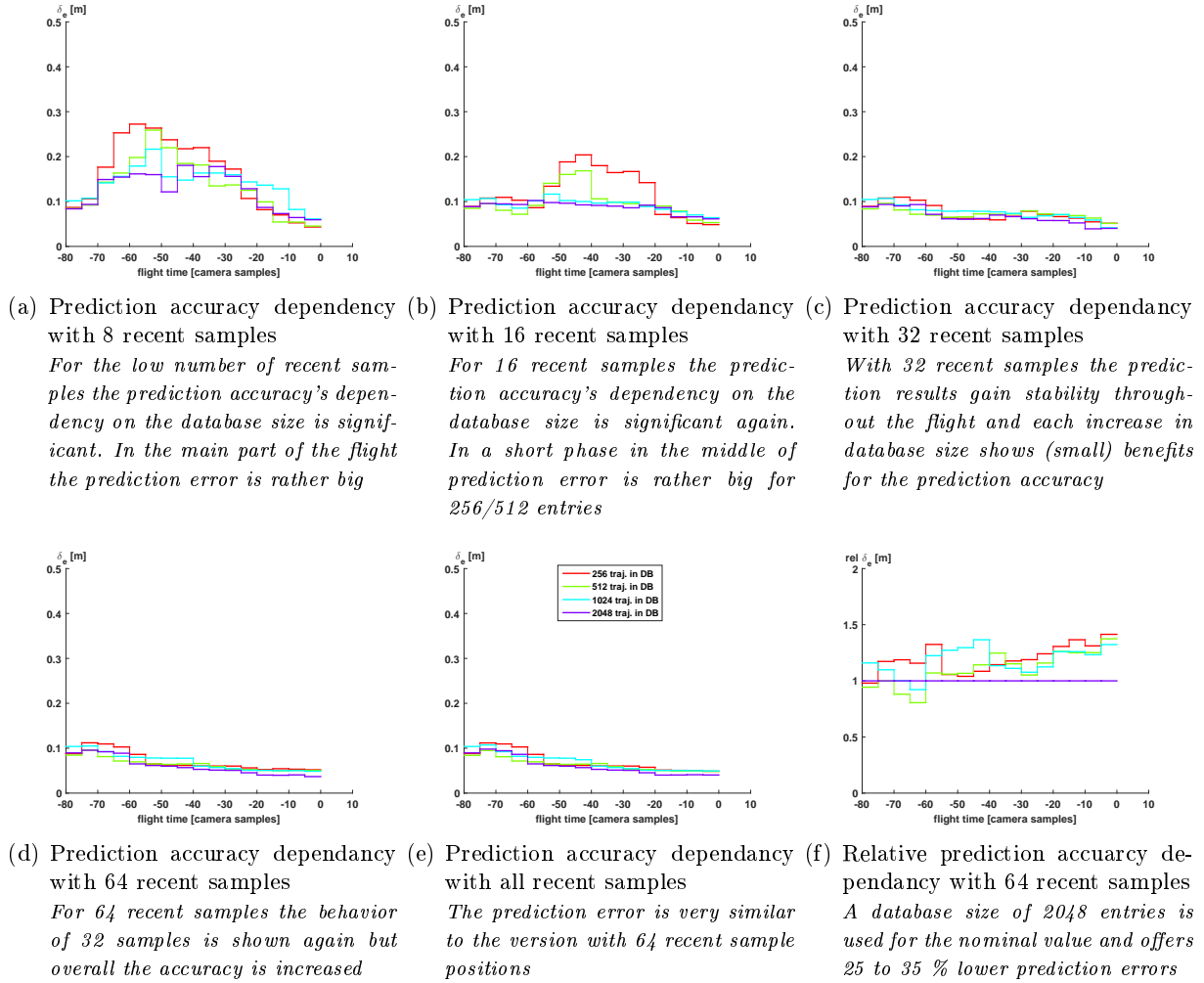


Figure 5.4: Illustration of the holistic prediction error over time

Catching time is normalized to the time 0. The recent history is fixed for each individual diagram. For 32 recent samples and more 1024/512/256 entries show $\approx 20/30/35\%$ higher prediction errors in the final phase than 2048 entries

(number of positions considered for the similarity determination; 8, 16, 32, 64 and 99/all as options) and the sixth is showing the relative error in case of 64 recent positions. The size of the recent history is the maximum of the positions considered. This has to be noted especially for the initial flight phase where the number of determined positions is (can be) smaller than the number of the recent samples. Also, the version with 32 recent position is similar the one with 64 recent positions for the first 32 positions. For short recent histories (8 and 16 positions) the influence of the database's size is minor due to instability because of the low number of samples

and the sensitivity to measurement noise, especially in the middle phase of the flight when the ball's vertical movement is slow. Here shifts in the prediction can occur and cause prediction errors. For the bigger recent history sizes increased database sizes show an increased prediction accuracy for the main (especially the last two-thirds) part of the flight. Overall the result is: the bigger the database's size, the better the prediction accuracy.

Number of Recent Samples In comparison to the previous illustration Figure 5.5 shows different recent history sizes at a fixed database size per diagram.

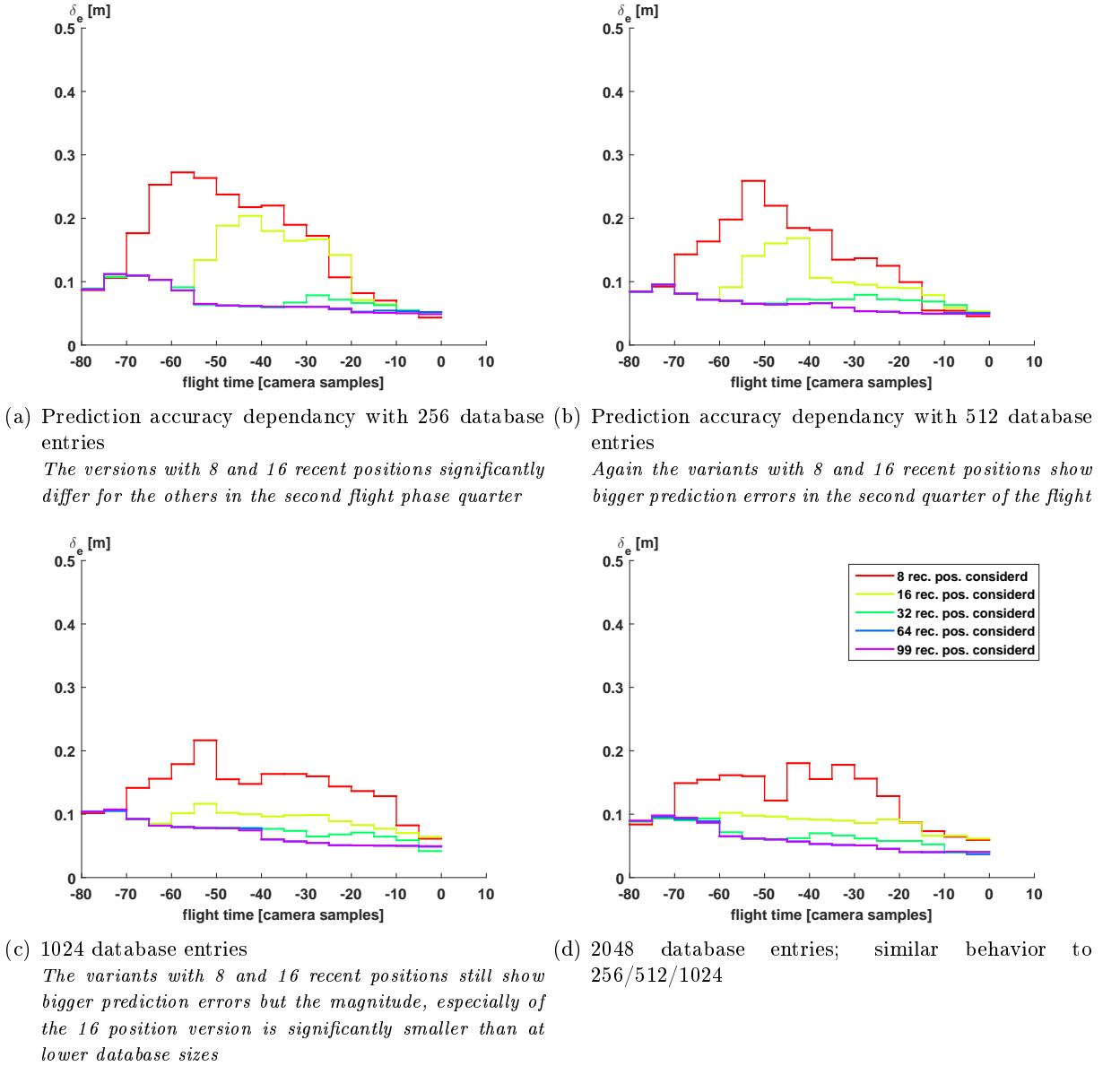


Figure 5.5: Illustration of the holistic prediction error over time

Catching time is normalized to the time 0. The database size is fixed for each individual diagram

The main behavior for all trajectory sizes is similar: 8 and 16 recent positions considered leads to bigger prediction errors through main parts of the flight. 64 recent positions and more show

similar errors for all trajectory sizes. In the very final part of the throw, the small recent history sizes manage to deliver better prediction results than the bigger ones due to the ability to adapt to the current situation better.

Comparison and Discussion For all combinations, a recent history size of 8 or 16 positions leads to instability in the prediction and causes bigger errors, especially in the middle flight phase. For bigger recent histories only small benefits exist when moving from 32 to 64 positions and minimal benefits when moving from 64 to all/99 positions. In initial phase all (from 32 on) are same because of the limited number of recent window size (number of positions detected is the maximum number of positions compared). This lead to the conclusion that a size of 32 recent position is a good trade of between calculation demands and prediction errors. In case additional positions can be considered without performance implications, a bigger recent history size still improves the prediction accuracy.

The influence of the database's size is more significant when using recent windows sizes of more than 32 positions that are the recommendation based on the analysis in the previous paragraph. For 32 recent positions considered both, 1024 and 2048 entries, deliver similar prediction results with small advantages for the higher database size. For the bigger recent history sizes the bigger database deliver more than 20 % better prediction results than a database size of 1024 entries (compare Figure 5.4(f)). For the implementation doubling the database's size results in double the processing power required to consider all trajectories. Based on this fact the ideal combination of parameters for the implementation of the holistic associative prediction is 1024 trajectories in the database and a recent sample window of 64 positions. For the database size, an increase still leads to a more accurate prediction, but the trade-off for processing requirements has to be considered.

Overall the significant prediction error in the final flight phase shows a major problem of the position-based prediction: A very close fit to the current trajectory has to exist in order to allow an accurate prediction. The dependency on the absolute positions is a big restriction on the usability of predictions in the database for prediction.

5.1.2.2 Progressive Associative Prediction Results

The progressive prediction based on the position change has the same parameters as the variant based on the positions: database size and recent windows size. Again the influence of varying these parameters on the prediction accuracy (upper bound of the 99 % error band) is done. A main difference is that the relative movement of the ball is considered which allows to use a the prediction database's entries more general which should lead to more accurate prediction, especially in the final flight phase.

Database Size In Figure 5.6 the influence of changing the database size while keeping the recent windows size constant is illustrated. For all the recent history sizes the dependency of the prediction accuracy on the database's size is insignificant. The suggested behavior, that fewer entries in the database still allow an accurate prediction (raised in Section 3.2.2) is confirmed by the simulation.

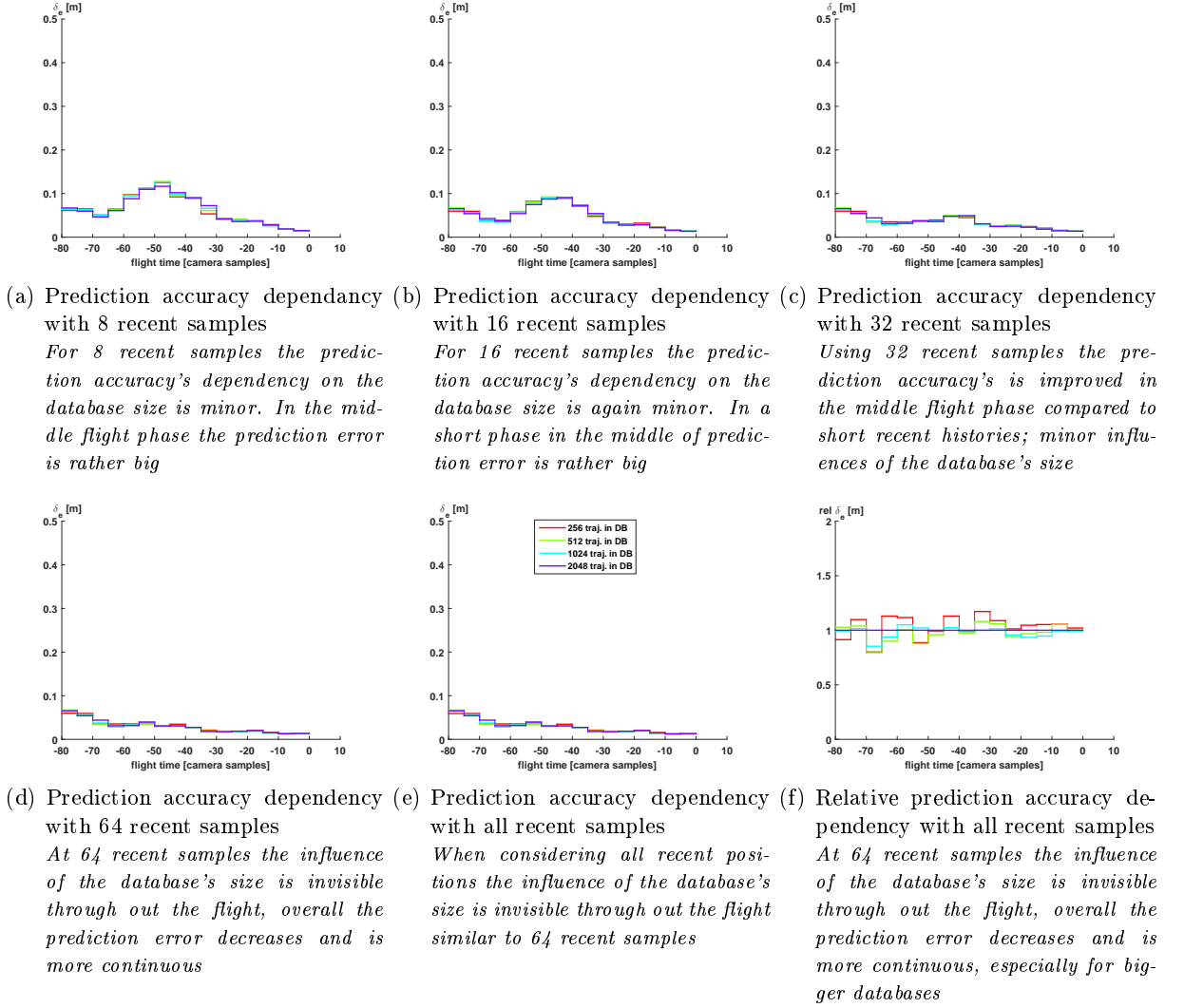


Figure 5.6: Illustration of the progressive prediction error over time

Catching time is normalized to the time 0. The recent history is fixed for each individual diagram

Number of Recent Samples The second part of the progressive prediction error analysis is targeting the recent history windows size's influence on the prediction accuracy. Figure 5.7 shows the prediction errors with fixed database sizes while varying the size of the recent window in the individual diagrams. The main behavior for all trajectory sizes is similar: bigger recent window sizes lead to more accurate prediction results, especially in the middle and last part of the flight. For the database sizes of 1024 and 2048, the recent histories of 64 and 99/all show very similar prediction accuracy. Using 32 recent positions instead leads to a bigger error in the middle part of the flight path prediction. The variants with 8 or 16 recent positions deliver substantially worse prediction accuracy.

Comparison and Discussion For all database sizes the bigger recent history windows result in a prediction error that decreases earlier, and lower prediction errors are achieved overall as well. Longer recent windows than 64 positions do not yield in any further benefit.

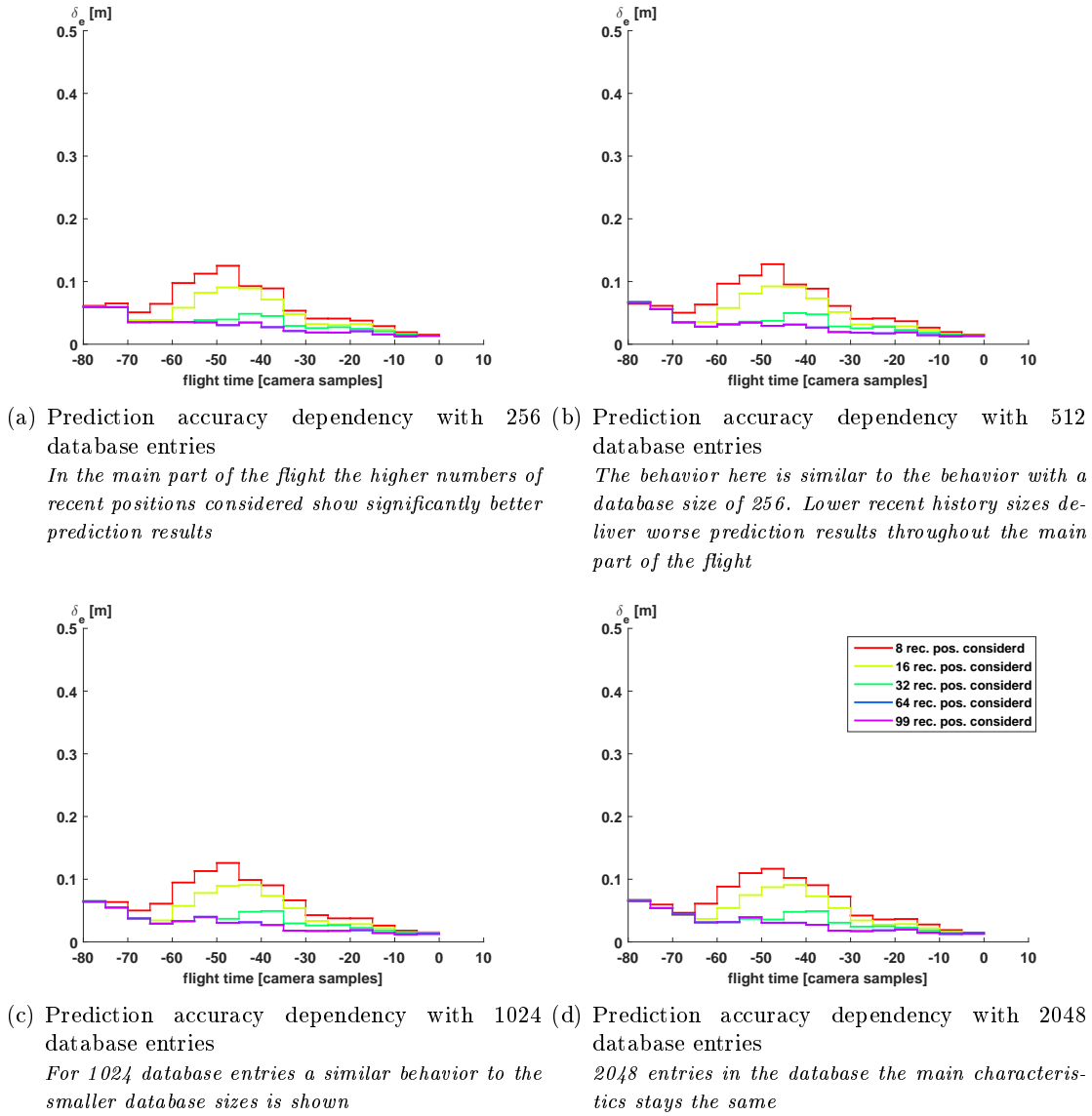
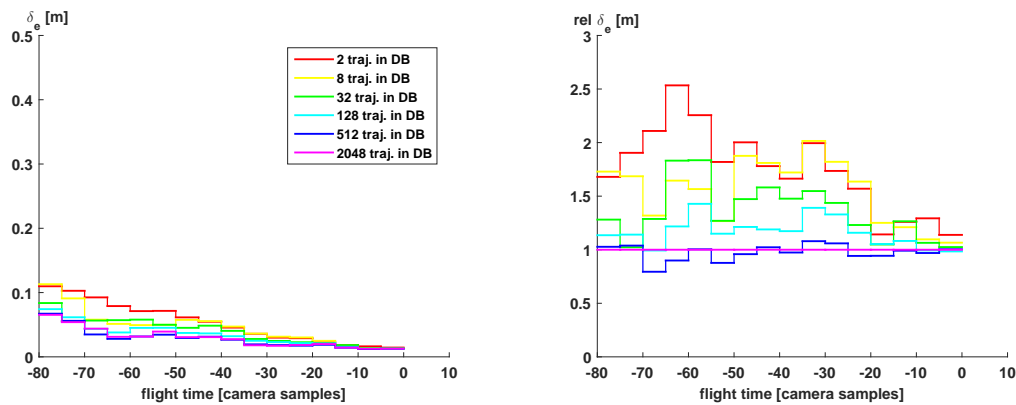


Figure 5.7: Illustration of the progressive prediction error over time

Catching time is normalized to the time 0. The database size is fixed for each individual diagram

In terms of the database's size, the influence of this parameter is interestingly small. This proves the assumption raised in Section 3.2.2 that moving away from the absolute position to changes causes a generalization of the data, and thus smaller databases lead to better prediction result and the benefit of moving to bigger ones is small to non-existing. To prove this assumption additional simulation runs with a further reduced size of the database down to only two trajectories were made. The results are shown in Figure 5.8 for a recent sample window of 64 positions. Overall the prediction error in the final phase is small as this prediction model is able to use the information more widely than by example the position-based prediction.

Even the tiny database of 2 trajectories allows to predict the flight with reasonable accuracy. More occupied databases lead to better results but still the general application of the small database's content is remarkable. This is valid especially in the final phase of the flight. While the relative



(a) Prediction accuracy dependency with 1024 database entries
For 1024 database entries a similar behavior to the smaller database sizes is shown

(b) Prediction accuracy dependency with 2048 database entries
2048 entries in the database the main characteristics stays the same

Figure 5.8: Prediction accuracy dependency with 64 recent samples; further reduced database size down to 2 trajectories

error shown in Figure 5.8(b) is significantly bigger for smaller databases the absolute error of the progressive prediction is small compared to the holistic prediction by example. This means that the overall error (compare Figure 5.8(a)) is still on a low level.

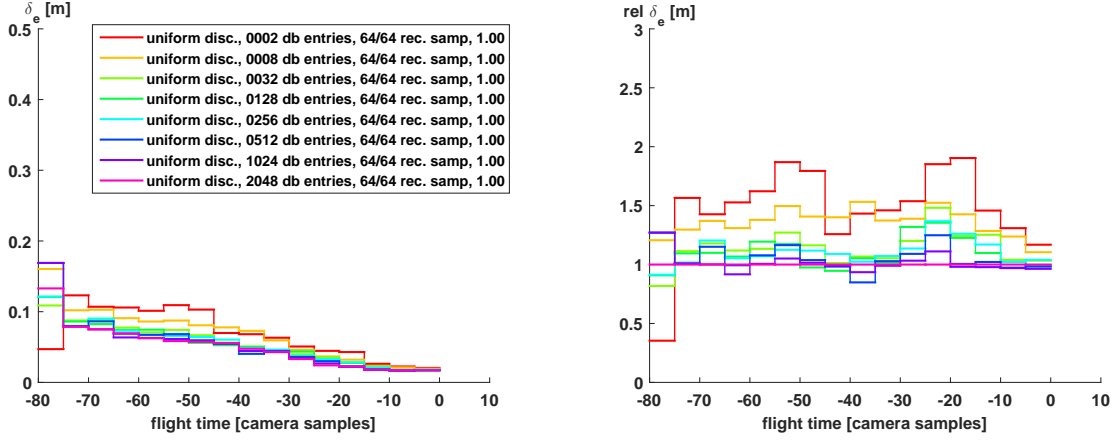
5.1.2.3 Progressive Associative Prediction Results with additional Discretization

The progressive associative prediction is examined in a similar way to the previously discussed prediction approaches. The upper bound of the 99 % error band is the measure discussed for the catching position prediction. The following diagrams will again show this parameter over time. For this approach, the number of parameters is significantly bigger than for the previous approaches. Both, the database size and the number of recent samples considered, have been parameters of the holistic associative prediction and in addition, the number of points considered in the recent sample window and the discretization interval are additional parameters. The influences of these parameters will be discussed with some examples in the following paragraphs.

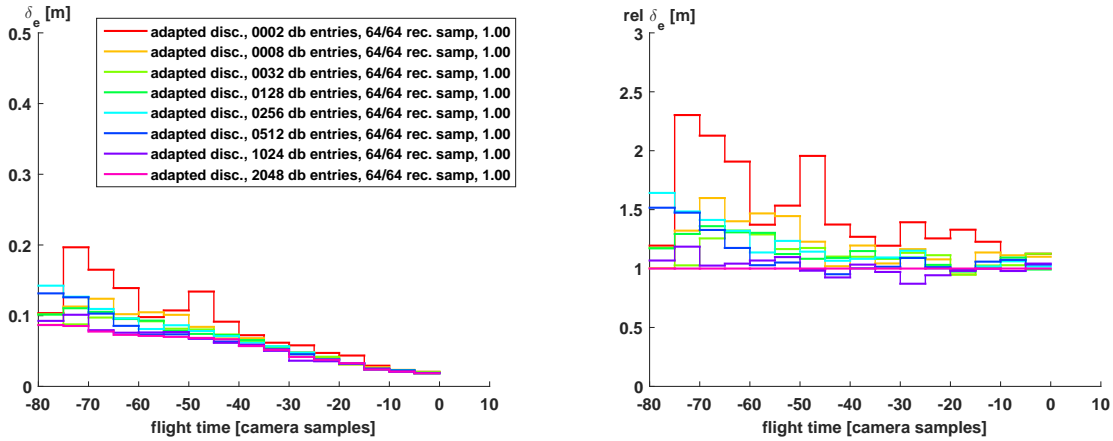
Results of Parameter Variation

Overall the number of different parameter combination is 840 (4 database sizes (256, 512, 1024 and 2048 trajectories), 5 different recent windows sizes (8, 16, 32, 64 and 99/all), on average 7 different numbers of supporting points (5 for 8 recent samples, 6 for 16 recent samples, 7 for 32 recent samples, 8 for 64 recent samples and 9 for 99 recent samples) and 6 different discretization intervals (either uniform or camera adapted relations between x/y/z-discretization steps and additionally the following factors: $\frac{1}{4}$, 1 and 4). Similarly to the holistic associative prediction the database size's and the number of recent samples' influence will be discussed first followed by the number of supporting points and the discretization interval. Because of the relation to the progressive prediction without additional discretization, the findings of the previous section are the starting point for the analysis here.

Database Size In order to provide a good overview of the database's size influence on the prediction accuracy a number of parameter combination of the other parameters are chosen and the related diagrams are shown in Figure 5.9. For the diagrams shown in Figure 5.9(a) and



(a) Prediction accuracy dependency with 64 recent samples down to a database size of 2 trajectories (b) Relative prediction accuracy dependency with 64 recent samples down to a database size of 2 trajectories



(c) Prediction accuracy dependency with 64 recent samples down to a database size of 2 trajectories (d) Relative prediction accuracy dependency with 64 recent samples down to a database size of 2 trajectories

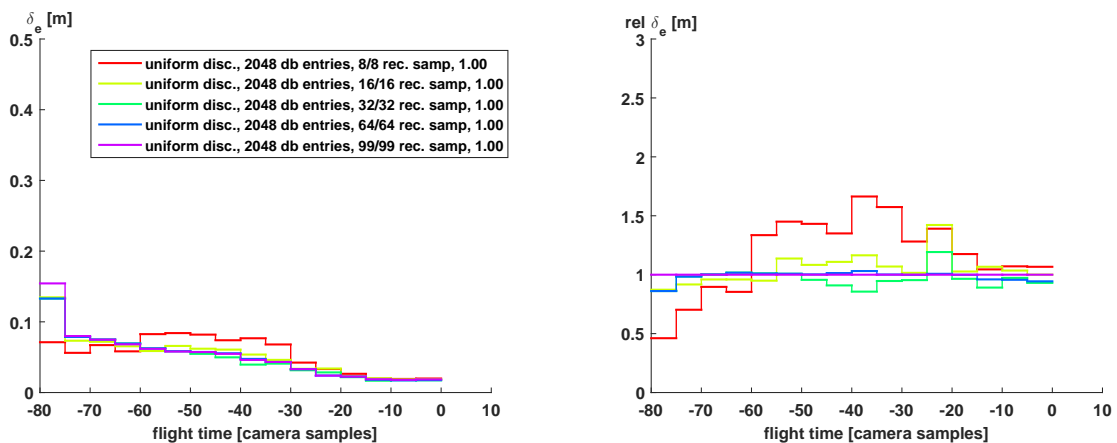
Figure 5.9: Illustration of the progressive prediction error over time; Catching time is normalized to the time 0; uniform discretization is used for the top diagrams and RANSAC-adapted for the bottom two

The influence of the databases size is relatively small and even tiny databases allow an accurate prediction. The RANSAC-adapted discretization shows bigger prediction errors which will be analyzed in Figure 5.13 and Figure 5.14 in more detail

Figure 5.9(b) the discretization interval is uniform. The same relation between prediction accuracy and database size as for the progressive prediction is shown. The influence of the database's size on the prediction accuracy is minor for the whole final flight. Also the extension of the database's size variation down to 2 entries is shown and again tiny databases allow to predict the flight with reasonable accuracy. This is comparable with the behavior of the progressive prediction where only extremely small databases led to significant larger prediction errors. The same is valid for the same parameter combinations but a discretization interval adapted to the camera system's accuracy. The error diagrams are shown in Figure 5.9(c) and Figure 5.9(d) and the interpretation is the same.

The conclusion here is that the database size has a minor impact on the prediction accuracy down to very small databases. As the database size has a clear impact on the processing time for the calculation or memory requirements the size can be chosen adequately to fit the timing/memory constraints of the prediction system. The database's size also has a direct influence on the numbers of velocity measurements within one voxel/velocity class (compare Figure 4.18 in Section 4.4.1). The occupancy of the individual voxels for the parameters database size and discretization interval are examined in the paragraph related to the discretization interval of this section.

Number of Recent Samples The number of recent samples considered for the similarity determination has an impact on the computation time as this proportionally depends on the number of database accesses. For each additional position considered the linked list has to be followed one more step and the distance calculation also has to be done to the additional position. Due to this behavior, a low number of recent samples is of advantage. Another positive aspect of a low recent history size is that the reaction to possible changes in the environment (sudden side wide for example) is faster as this change is relatively more relevant to a shorter recent history. On the other hand, a longer recent history is assumed to deliver a more accurate prediction because more positions are compared and thus the similarity search is profound. The results of the examination are shown in Figure 5.10 for uniform discretization. The recent history's

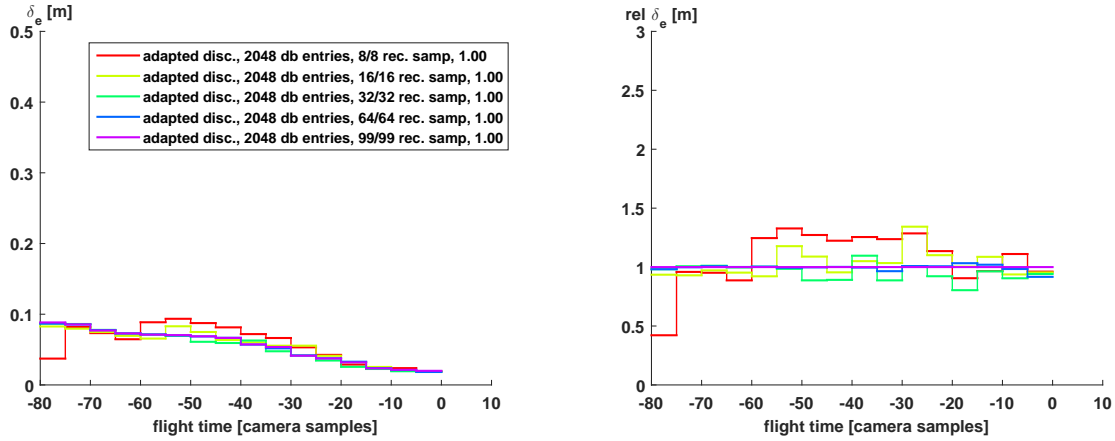


(a) Prediction accuracy dependency with a database size of 2048 and variable recent samples (b) Relative prediction accuracy dependency with a database size of 2048 and variable recent samples

Figure 5.10: Illustration of the progressive prediction error over time; Catching time is normalized to the time 0; uniform discretization is used

The influence of the recent history size is relatively small above 32; 16 and especially 8 recent samples show significantly bigger errors in the middle flight phase

influence on the prediction is significant for the lower recent history sizes of 8 and 16 positions where the prediction error, especially in the middle part of the flight phase is up to 50 % bigger than for higher recent sample numbers. The differences between the higher numbers of positions considered are insignificant. Figure 5.11 shows the same examination for the RANSAC-adapted discretization. In general, the same behavior as for the uniform discretization interval is shown. The extend of the influence on the prediction error is smaller in this case. The relative error for predicting with only eight recent samples is, at all times, smaller than 150 % of the error in the case when predicting with 99 recent samples.



(a) Prediction accuracy dependency with a database size of 2048 and variable recent samples (b) Relative prediction accuracy dependency with a database size of 2048 and variable recent samples

Figure 5.11: Illustration of the progressive prediction error over time; Catching time is normalized to the time 0; RANSAC-adapted for the bottom two

The influence of the recent history size is relatively small for above 32

Number of Supporting Points Depending on the size of the recent history window (number of recent samples considered) the number of supporting can be varied. In order to do this variation systematically each step of reduction approximately halves the number of points considered. The detailed number of points considered for the similarity determination for the variants of recent samples are given in Table 5.1. At the beginning of the flight, the number of acquired positions is

recent samples	variants of additional supporting points
8	0 1 2 3 6
16	0 1 2 4 7 14
32	0 1 2 4 8 15 30
64	0 1 2 4 8 16 31 62
99	0 1 2 3 6 12 24 49 97

Table 5.1: Additional supporting point variants in dependency on the number of recent samples; first and last velocity measurement are always considered

limiting the number of recent samples and thus also the number of supporting points is adapted accordingly. The prediction error's dependency on the number of supporting points is shown in Figure 5.12(a) and Figure 5.12(b). For the uniform discretization, the influence of the supporting points considered is significant. The fewer points considered the bigger the prediction error for most of the flight path's parts. Here the assumption that more points for the similarity calculation deliver a better quality measure. Figure 5.12(c) and Figure 5.12(d) show the same illustration for the RANSAC-adapted discretization. Interestingly the relative error's dependency on the number of supporting points is less pronounced here. Overall the error of the RANSAC-adapted discretization is $\approx 3\%$ bigger than for the uniform discretization. This aspect will be discussed in more detail in the following section.

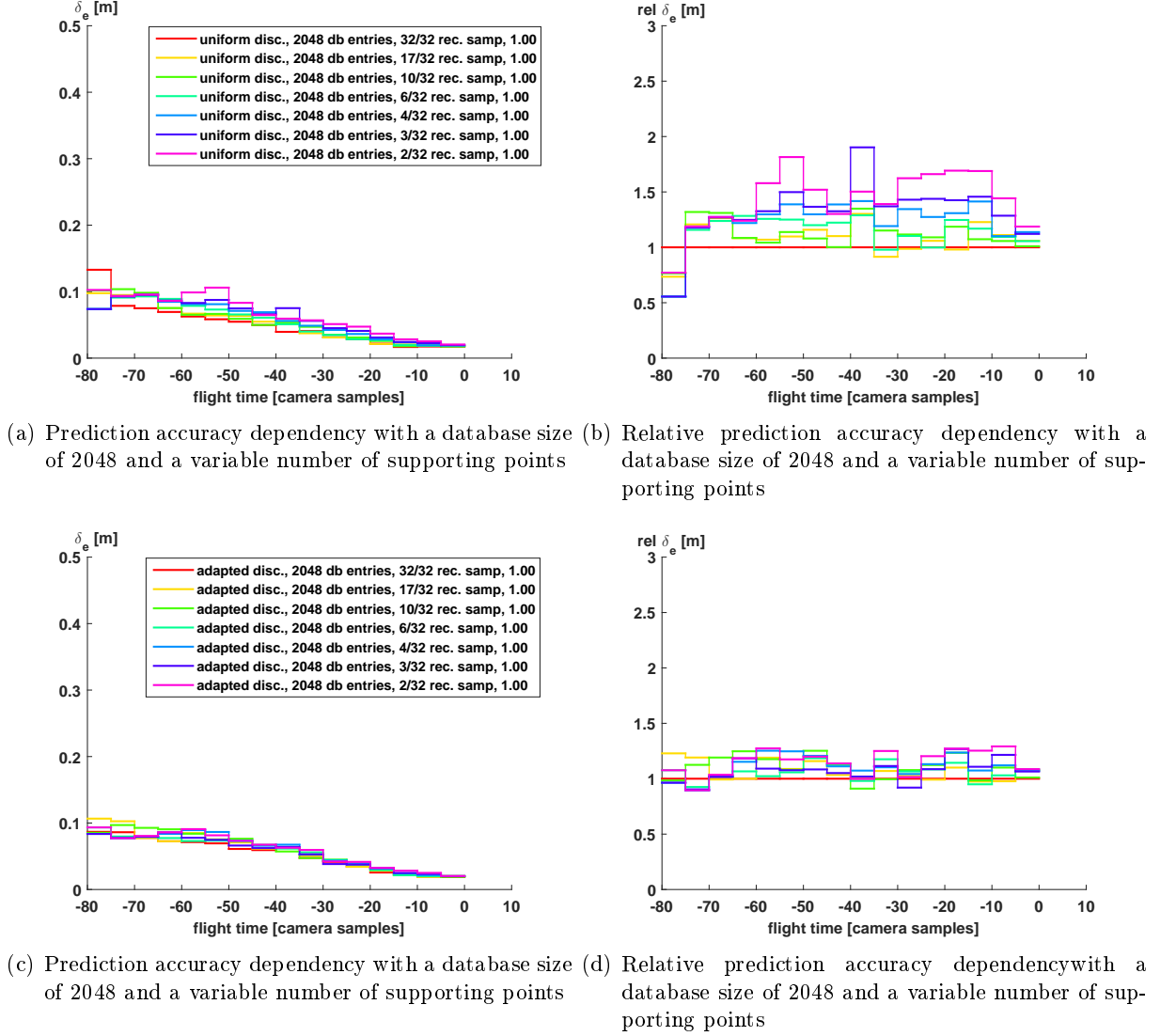


Figure 5.12: Illustration of the progressive prediction error over time; Catching time is normalized to the time 0; uniform discretization is used for the top diagrams and RANSAC-adapted for the bottom two

The prediction error is increasing as the number of supporting points decreases for the uniform discretization

Discretization Interval For the variation of the discretization interval, two main aspects are predominant. One aspect the volume of one discretization voxel in space and the second is the shape of the discretization voxel. The larger this voxel is, the larger the prediction uncertainty due to the discretization is as well. On the other hand very small voxels cause the search time to rise significantly because the population of the voxels with entries of velocities is sparse. The standard voxel used for the simulation has a uniform side length of $\frac{1 \text{ mm}}{\text{frame}} = 0.1 \frac{\text{m}}{\text{s}}$. This voxel's size is scaled up and down with the following factors: $\frac{1}{4}$, 1 and 4 resulting in a $\frac{1}{64}$ to 64 times the volume of a voxel. In addition, the hypothesis that considering the position detection system's accuracy for the determination of the voxel's shape is beneficial is tested. For this, the accuracy of the position detection system in the three directions is used as the proportion for the voxel's

shape. Again the scaling factors for $\frac{1}{4}$ to 4 are used. The influence of the used scaling factor on the prediction error for the uniform voxel is shown in Figure 5.13 for uniform discretization in the top row and RANSAC-adapted discretization in the bottom row.

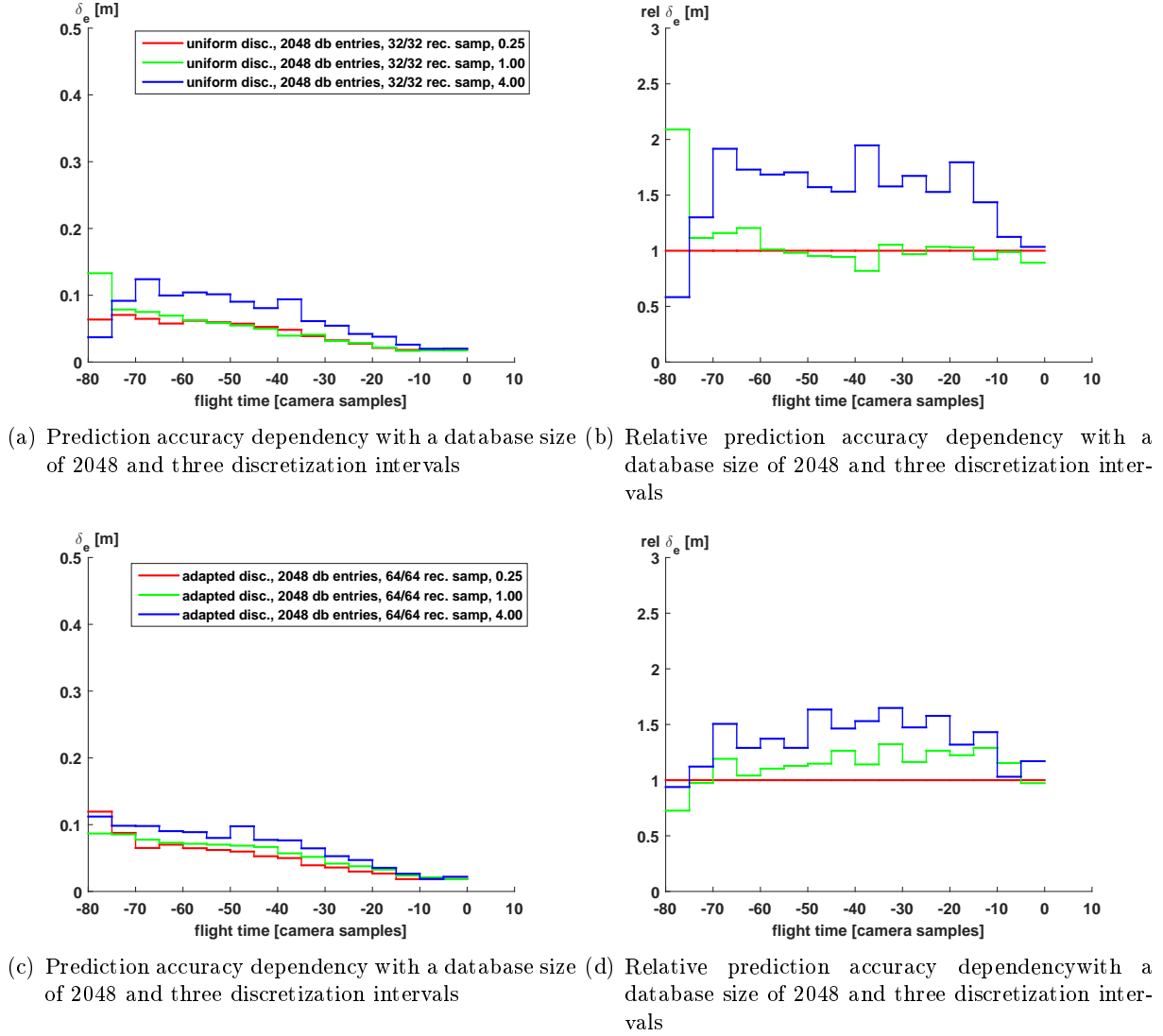


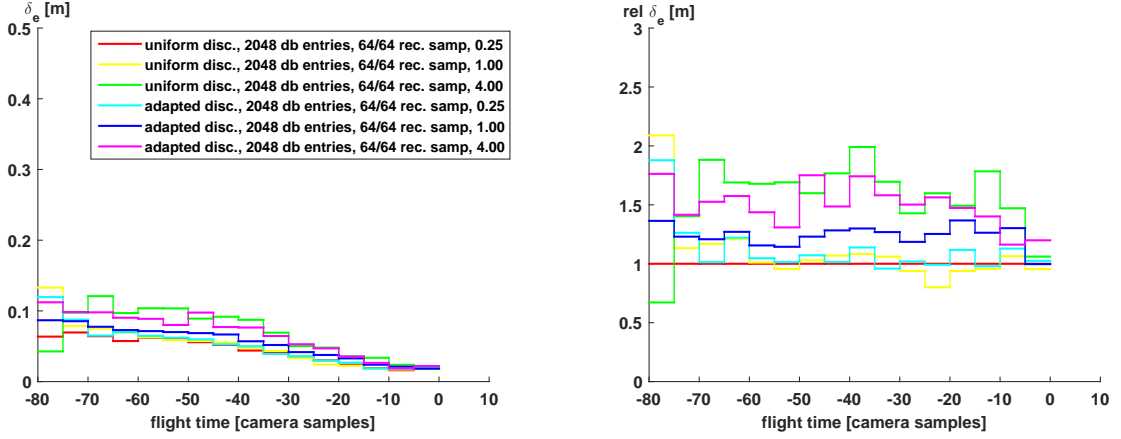
Figure 5.13: Illustration of the progressive prediction error over time; Catching time is normalized to the time 0; uniform discretization is used for the top diagrams and RANSAC-adapted for the bottom two

The prediction error is increasing as the discretization interval increases for the uniform discretization significantly when moving to four times the interval. For the RANSAC-adapted discretization the influence is less pronounced but also visible for the decreased interval size which leads to more accurate prediction.

The influence of the discretization interval's scaling is stronger at the uniform discretization but only when moving to four times the interval's size. Using a smaller interval does not show benefits there (compare Figure 5.13(a) and Figure 5.13(b) for 32 recent samples with all points used for comparison). Figure 5.13(c) and Figure 5.13(d) show the error diagrams for the 64 recent samples but for the adapted discretization interval as the basic unit for the scaling. The behavior here is the similar to the case with the uniform discretization interval but less significant and moving to

a smaller prediction interval shows benefits in this case.

Figure 5.14 shows a prediction error's comparison for uniform and RANSAC-adapted discretization intervals for 2048 databases and 64 recent samples with all recent samples considered. The results presented show that the uniform prediction at a discretization interval of $\frac{1}{4}$ and 1 are on par with the RANSAC-adapted at $\frac{1}{4}$. The RANSAC-adapted prediction then shows lower increases in the prediction error as the discretization interval is increased which causes the uniform interval with a factor of 4 performing worst. One might argue that the smoothing of the trajectory renders



(a) Prediction accuracy dependency with a database size of 2048, 64 recent samples and variable discretization intervals (b) Relative prediction accuracy dependency with a database size of 2048, 64 recent samples and variable discretization intervals

Figure 5.14: Illustration of the progressive prediction error over time; Catching time is normalized to the time 0;

The uniform discretization leads to the most accurate prediction at factors of $\frac{1}{4}$ and 1 and the least accurate at 4. The RANSAC-adapted discretization is on par at $\frac{1}{4}$ and within at the two remaining factors.

the tracking error of the position detection system insignificant. This argumentation is clearly comprehensible but in this special case, the largest uncertainty in the tracking system aligns with the main motion direction of the tracked object as the ball is thrown away from the cameras. Adapting the discretization interval's size to the prevailing velocity is done here implicitly which would also be an interesting topic to elaborate further on. On the other hand, if the system was to be used for more general movements in any directions this makes this optimization impossible due to the general application.

5.1.3 Discussion of the Simulation Results

While the previous sections were devoted to finding the optimal parameters within the individual proposes approaches here, the approaches are compared with the reference approach, the UKF-iterative approach. Figure 5.15 shows the absolute and relative (to the UKF-iterative approach) prediction error. Again the upper bound of the 99 % error band is used for comparison. All the prediction models introduced show a better prediction accuracy for the first half of the flight. In the later part, only the progressive prediction variant with 64 recent positions considered bests the reference approach with the progressive prediction with discretization showing a little bigger prediction error in the magnitude of less than 30 %. The errors at the time when the reference

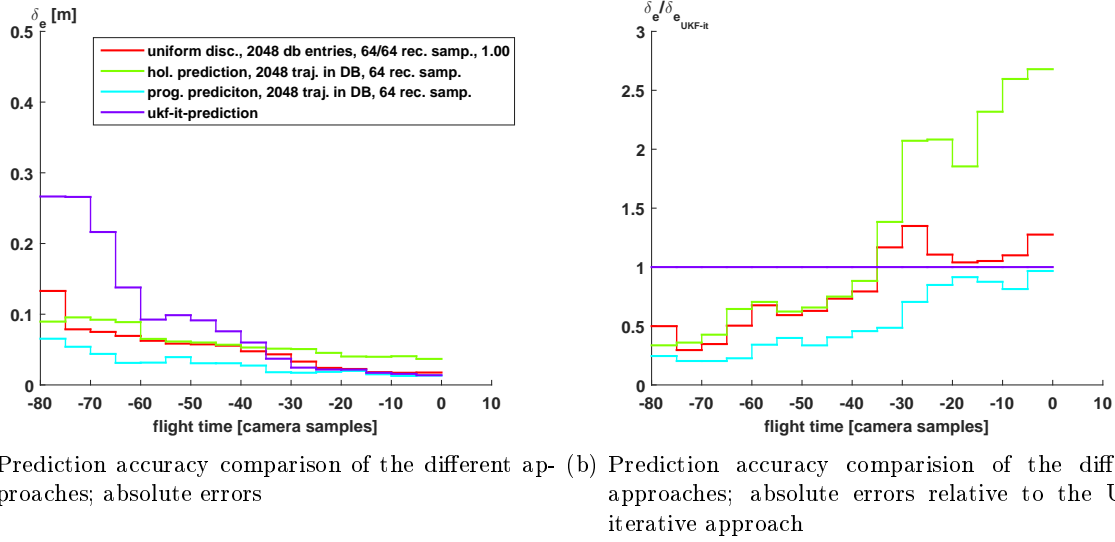


Figure 5.15: The comparison of the prediction error in Figure 5.15(a) shows that all own contributions show better prediction accuracy in the first third of the throw. In the second third, the benefits of the holistic prediction with 32 recent positions are also visible. The relative comparison in Figure 5.15(b) illustrates the differences in the final part better. In this area the UKF-iterative reference

model bests the progressive prediction with discretization are already at ≈ 4 cm. The Table 5.2 shows when the individual approaches manage to predict the catching position within different thresholds. The table shows a clear advantage for the progressive prediction model when having a

prediction variant	frames before catching when prediction error is below			
	5	4	3	2 cm
progressive w. discretization	40	30	25	15
holistic	25	5	-	-
progressive	80	75	40	35
UKF-iterative	35	35	35	15

Table 5.2: Number of supporting point variants in dependency on the number of recent samples

look at the time when the accuracy is below 2 cm which is the accuracy specified as the requirement for successful catching in related work [BBW⁺11, pg. 514]. Reaching this accuracy earlier allows to prepare the movement of the robot on time and gives headroom for the usage of the soft catching approach proposed in Section 4.3. The progressive prediction with discretization reaches this threshold at the same time as the UKF-iterative reference model and the holistic prediction never reaches this threshold. This result renders the progressive prediction without discretization as the most suitable prediction for the practical experiment and shows the possibilities of the progressive prediction with discretization for more general problems.

5.2 Implementation

Additional evaluation of the prediction results of the simulation presented in the previous subsection is done in a series of practical throws. The experiment setup is described in Section 4.5 and the environment, including the modeling of the environment, is described in Subsection 4.2.2, the used throwing device especially in Subsection 4.2.2.2. The catch rate for the three catching devices with a diameter of 80 mm, 100 mm and 110 mm are determined for the soft catching strategy introduced in Section 4.3.3. The differently sized catching devices allow putting the success rate in relation to the prediction accuracy determined in the previous section. Table 5.3 shows the catching rates achieved in experiments with 50 throws and the three catching devices. An image

$d_{cat-dev}$	80	100	110	mm
50 throws	20	42	43	successful catches
	40	84	86	success rate in %

Table 5.3: Catching rates of the robot with soft catching strategy and three differently sizes catching devices (compare Figure 4.23)

series, containing the 1st, 37th, 82nd, 85th, 88th and 94th image acquired by the cameras of a successful catch is shown in Figure 5.16. The left column is showing the first three image pairs of the series while the right column is showing last three image pairs. The soft catching movement is

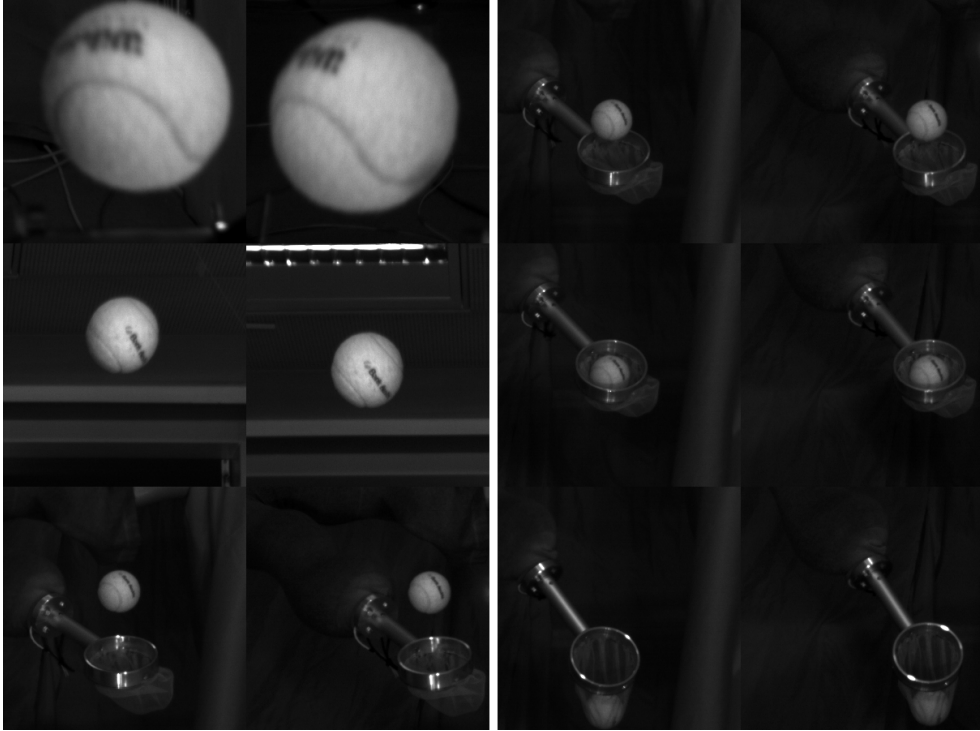


Figure 5.16: Image series of a successful catch; top row left images; bottom row right images; 1st, 37th, 82nd, 85th, 88th and 94th

illustrated in the images of the 82nd to the 94th frame. The robot is moving along the path of the ball during catching. The catching device used in Figure 5.16 is the biggest catching device with

110 *mm* diameter. Additionally images of the same throw acquired with two external cameras, that are synchronized with the main cameras, are shown in Figure 5.17 for the overview in Figure 5.16 and a series illustrating the details of the soft catching from the side is given in Figure 5.18. The series in Figure 5.18 shows the adaption of the robot's position during the flight of the ball.



Figure 5.17: Image series of a successful catch; images are synchronized with the ones shown in Figure 5.16

The robot's positions in the first three images differ significantly and show the adaptation to the prediction during this time. Another image series, showing the sensitivity of catching process,

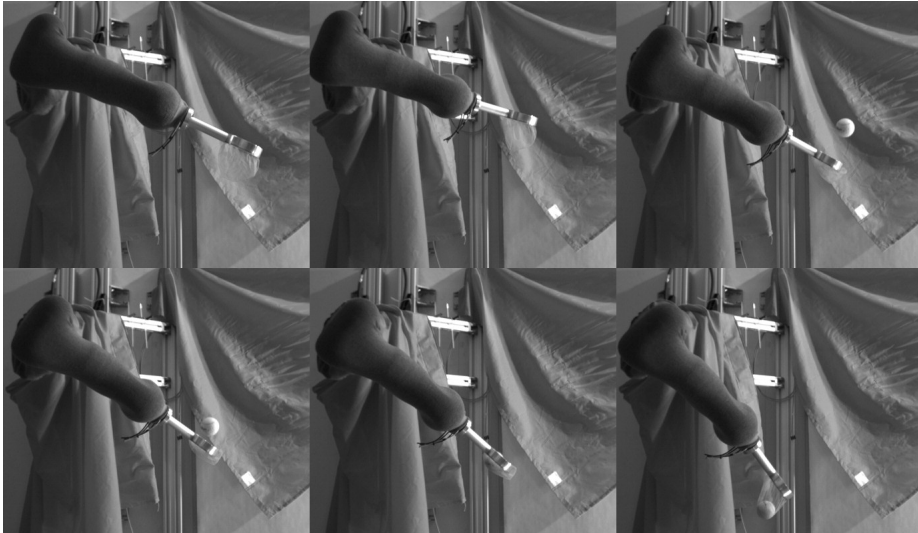


Figure 5.18: Image series of a successful catch; images are synchronized with the ones shown in Figure 5.16

is given in Figure 5.19. The catching device used in this case is the smallest one with 80 *mm* diameter. A sequence of 15 images is given (84th to 98th frame of the throw). The catch is not successful as the ball bounces four times on the corners of the catching device, and because the trajectory of the robot

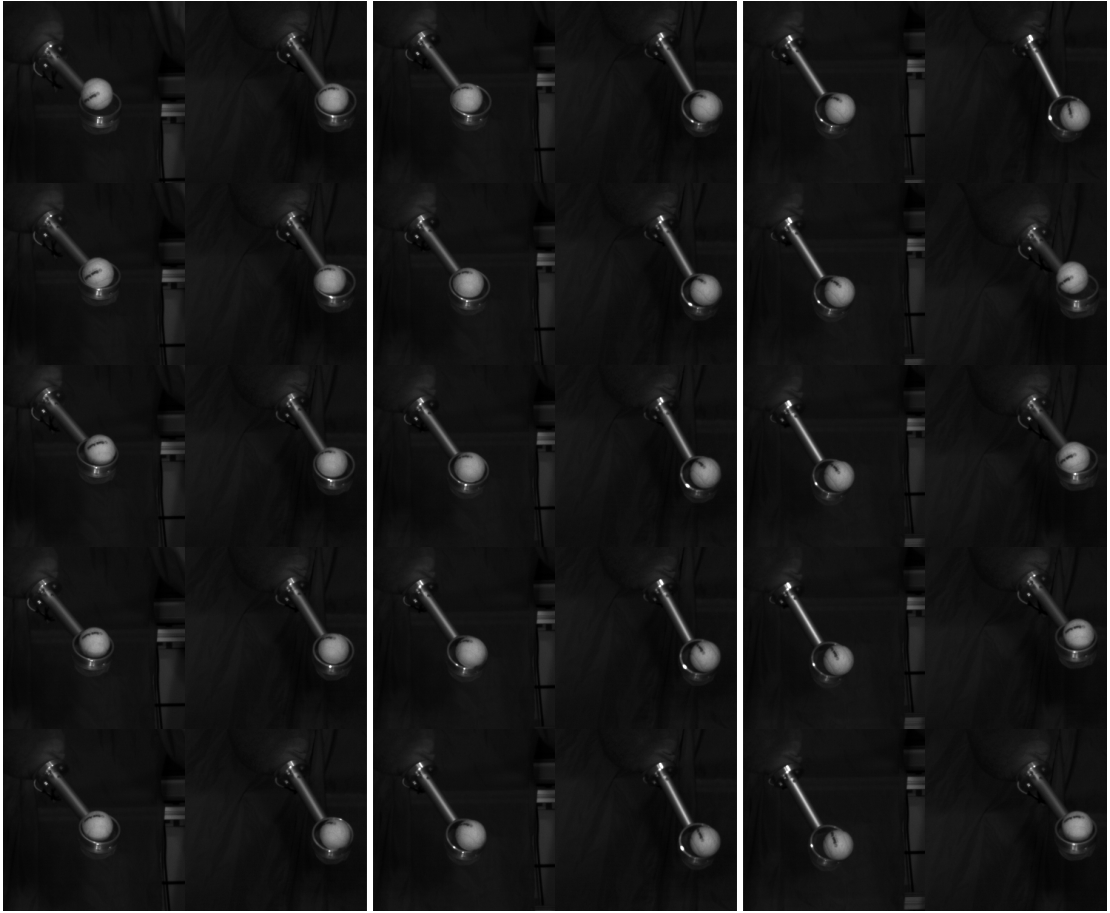


Figure 5.19: Image series of an unsuccessful catch; left and right images side by side; first column frame 84th to 88th; second column frame 89th to 93th; third column frame 94th to 98th

is differing from the ball due to the robot's constraints and the used soft catching algorithm, the ball is not caught. An additional evaluation of the system's timing aspect is given in the [Hubss].

6 CONCLUSION AND FUTURE WORK

In the introduction to this work, a number of questions have been raised, and the answers to this questions can be given based on the results achieved here. It is important to note that the work presented here is initial work while, for example, the reference prediction algorithm, more explicatively the unscented Kalman filter, is an algorithm that has been developed for 55 years now with the iterations of the original Kalman filter, the extended Kalman filter and finally the unscented Kalman filter. For the future, the prediction approach, presented here, has a lot of potential for improvement, and some ideas are outlined in the second section of this chapter.

6.1 Conclusion

While the motivation of this work is based on the transport-by-throwing approach, I would like to start this section by citing the conclusion of related work.

"To make a virtue out of necessity: One contribution of robotic sport activities is to operate robots in a regime that reveals problems which remain unnoticed in more forgiving applications. This is actually similar to sports training, where the goal is to come closer and closer to the level of performance the human body can do in principle." [BFB11, pg. 5962]

During the work on the presented topics, especially on the experiment, the limits of the state of the art technologies and algorithms has been visible many times. This is valid for all three subunits (tracking system, prediction system, and robot control system). Improvements in all the three regimes will lead to more successful object catching.

Raised Questions

In the introduction, the following questions have been raised, and the related answers will be given here:

Why are analytical/physical approaches not working perfectly? The answer to this question can be given based on the results presented in Section 4.2.2.1 and Section 4.2.2.2. On one hand even an optimized vision system (with feedback from the further processing steps) only achieves a tracking accuracy of $\approx 2 \text{ cm}$ (99.5 % error band bound) for the main parts of the observed flight trajectory. This is in the same magnitude as the required prediction accuracy in

for catching (compare Section 5.2 and [BBW⁺11]). If basic data's error is in the same order of magnitude as the required accuracy of the prediction even the best algorithms, that also model and consider the measurement errors, like the unscented Kalman filter cannot fulfill the task. On the other hand, the modeling of the ball's trajectory with the iterative model also showed that even with additional parameter variation of the drag coefficient (and implicitly the mass of the tennis ball) the estimation errors were two times the errors of the other methods. Here multiple factors have influences: A spin on the thrown ball can cause a change in the drag coefficient. Variations of the balls (felt and mass) thrown can cause variation of the fitted parameters. Variations of the temperature can cause variations in the drag coefficient. Summing up: there are a number of physical effects that are not modeled accurately and even if they were modeled accurately the acquisition of the required information and processing of the data would be a big challenge. An example for a successful accurate modeling of more complex flight parameters is the work presented in [Fra12], where self-stabilization of cylindrical objects is used for successfully catching 98 % of the thrown objects.

What is a suitable representation of the data? This question is answered by the prediction results, shown in Section 5.1, especially Subsection 5.1.3. Moving from positions to position changes or velocities has the advantage that the stored data can be used more general and widely for the prediction. This causes a better prediction accuracy. When additional steps (here discretization) are used to reduce the search complexity and improve the search speed basically similar results can be achieved, but the additional discretization can cause additional errors. These errors are depending on the discretization interval and as shown in Section 5.1.3 these errors can be insignificant if the interval is small enough. This means that the errors are below the errors of the tracking system, and thus the prediction accuracy is on par with the standard variant considering the position change.

How to keep the memory consumption of the experience low? In a general context, even the largest database used for the experience-based prediction has a size of only 5 *Megabyte*. For current PC systems, this database even fits into the cache of a CPU. Hence, the memory consumption is generally rather low here. In terms of the used GPU (Graphics processing unit) also the onboard RAM has approximately 1000 times the size. Still for the general context the usage of further discretization (as discussed in Section 3.2.2) allows to reduce the overall required amount of data, especially for big datasets where the initial overhead is relatively small. A second aspect to keep the memory consumption low is shown in Section 5.1.2.2. When the data representation is suitable (here position change), even tiny datasets (tested with down to 2 trajectories) can lead to relatively good prediction results. This is valid for both the progressive prediction without and with additional discretization. Here the introduced approach for prediction shows the potential for embedded applications, where power, area, and performance are the main limiting requirements.

How to associate current processes or events with previous events? The association of the current flight with the stored trajectories in the database can be done on currently available parallel processing units like GPUs with brute force. These architectures allow to process a significant amount of data in short time as long as the processing steps have reduced complexity and have to be applied to multiple data. This is exactly the case when comparing one (current) trajectory with multiple (stored) ones. In addition also more efficient algorithms for finding similar

velocities within a bigger dataset can be used. One approach is introduced and discussed in Section 3.2.2 and evaluated in Section 5.1.2.3. Additionally, to these approaches, further combination of similar trajectories could be done, but this is left for future work at this stage.

How can general application of the data to current trajectories be achieved? Predicting the trajectory of mechanically thrown ball is a rather constrained problem compared to predicting a ball thrown by a human from random positions. For the second case, efficient data storage and association are higher on the priority list than in the first case. The results here show that the first challenge can be addressed well with brute force approaches using the position change data. For a more general prediction, the usage of the additional discretization will have benefits in terms of the association complexity and thus prediction performance in terms of time. The error introduced due to the discretization has to be considered in relation to the required prediction accuracy. Using the discretization is basically the same procedure as using a symbol for the movement - this it is comparable to the current understanding of the processing in biological systems which, in general, is expected to be efficient and effective.

General Remarks

Additionally to the questions raised in the first chapter more aspects of the results obtained are worth mentioning.

Vision System Using the feedback from the triangulation for limiting the allowed radii in the next circle detection step increased the position detection system's accuracy significantly. This feedback on one hand allows this better accuracy but in a case of false detection, it can also lead to misdetections in the following frame and possible all consecutive ones. Here again, the biological system show higher stability despite using multiple feedbacks of information. The mechanism behind this property would be highly useful for technical systems as well.

Human Catching The work presented here and in related work shows how sophisticated the biological archetype, the human, is in terms of object catching. Despite using slow flying balls, cameras with a high resolution, state of the art image processing algorithms with additional feedback and either state of the art or self-proposed prediction algorithms the prediction accuracy is in the range of $\approx 2\text{ cm}$. Especially in sports, for well trained humans, the capabilities of the technical systems are comparable restricted. Here elaborating research on the mechanism in biological systems can lead to better results for the technical systems as well.

The open issues in the regime of robotic sports will require additional research work in many aspects until the performance of humans will be reached or even bettered.

6.2 Future Work and Outlook

The work presented here contains contributions in three main aspects. The future work is structured in this three main aspects in the following paragraphs. Some of the topics raised are relevant in more than one aspects so the related ideas will be mentioned in each section.

Vision System

For the vision system, the tracking accuracy improvement is the main aspect. Further improving the tracking accuracy will lead to more accurate prediction results for all subsequent prediction algorithms, either physics based or experience based. This improvement can be done based on the object detection algorithm.

Object Detection The RANSAC algorithm used here is based on the edge detection of the ball on the background. The processing power of current systems allows to use more algorithms for object detection in parallel, and the combination of this information might lead to better tracking accuracy. In the images processed here, the difference between the detected edge and the real outline of the ball was noticeable in some cases. Calculating the ball's position in space with stereo triangulation causes small deviations in the ball's position in the images to result in bigger deviations in the ball's distance from the camera system.

Temporal Resolution Another possibility to improve the tracking accuracy is in terms of the temporal resolution. The usage of areas of interests (AOIs) of the whole image for the further processing only allows to increase the frame rate in case of a bandwidth limit of the interface as it is the case here. Additionally, the calculation demands are decreased as only a subset of the pixels has to be processed. In this work this idea was used but the implementation was limited due to the camera driver's support. Based on an experiment a frame rate of $> 350 \text{ fps}$ could be achieved. This additional information can be used to reduce the mean tracking error further.

Multi-Cameras The usage of more cameras is another aspect that can lead to increased tracking accuracy. The magnitude of this method's success has to be questioned, evaluated and tested. On the other hand, as the tracking error for the optimized tracking system shown in Section 4.2.2.1 is close to constant for a distance of 0.5 m to 3 m . This allows the assumption that the error caused by the spatial quantization of the camera's pixels is not in the same order of magnitude as the overall error, and another aspect (e. g. the edge detection and RANSAC circle detection) is the main reasons for the error. Still this approach should be examined and evaluated.

Cost-efficient Cameras It is mentioned in the previous paragraph already that the tracking error of the camera system does not seem to be bound by the spatial resolution of the camera systems (it has close to constant for distances of 0.5 m to 3 m , compare Figure 4.11). This circumstance opens the field for usage of cameras with lower resolution and thus also lower cost. Additionally, if the hardware support is improved, the usage of the AoI of the used cameras should allow increasing the frame rate up to the range of 350 fps for the used cameras. This means that cameras with better hardware support and lower (spatial and temporal) resolution should enable a similar tracking accuracy with lower demands on the interface bandwidth, image processing and, economically relevant, cost.

Prediction System

As mentioned in the introduction of this chapter the bio-inspired approach for object catching is here in a very early stage. Experience based prediction can still be improved in many areas.

Further knowledge about the information processing of the human can be incorporated and can lead to improvements in the prediction accuracy. Here some ideas for the next steps to improve the prediction system are discussed.

Usage of Abstracted Information One step further then the preprocessing, done based on the unscented Kalman filter and Rauch-Tung-Striebel smoother, is to store more abstract information about the trajectories in the database. The first thought here is to store information as the initial launching parameters (position, velocity) and estimated parameters of the environment (gravity vector, drag coefficient, the mass of the ball) and use this information for comparison and prediction of the current flight. This method requires the calculation of the flight trajectory based on a physical model and again raised the problem of the incomplete modeling of the trajectory with simple models (compare Section 4.2.2.1). This leads to other ways of abstraction needed to solve this challenge.

General Objects The extension of this approach to more general objects is another logical step. Here the discretized progressive prediction has potential to show the benefits of the reduced search time more significant if the object is, for example, non-pointsymmetrical and additionally to the velocity change the orientation(-change) information has to be compared. For unstructured search strategies, this problem will lead to too long search times even for smaller databases. For the discretized progressive prediction the number of dimensions can be extended for each additional information (3 dimensions for orientation, one dimension for different objects for example). Still the neighborhood search is possible and similar object states can be associated with a short time.

Catching Hand-Thrown Objects Catching objects that are thrown by the throwing device used for the experiments have a limited variation (compare Section 4.2.2.2) and limit the requirements of the prediction as well. On the other hand, even the deviation of the throwing device leads to uncatchable balls for the robot. Still, catching hand thrown balls requires a more general content of the database (more variation in the stored trajectories) and thus, the database size is of significant importance. Here the findings in Section 5.1.2.2 and Section 5.1.2.3 show that the right representation of the data can cause that only small databases are required for a certain prediction accuracy. Additionally, the search can be based on the algorithms proposed in Section 3.2.2 and thus be faster and more efficient for this more general problem. The combination of the previously mentioned topics, especially the trajectory preprocessing, with this aspect would be highly recommendable for additional improvements of the prediction accuracy.

Learning Learning is currently not used in the approach. Basically adding learning to the approach means that the current trajectory information has to be smoothed to remove the errors of the tracking system and added to the database. This implies additional aspects that are worth mentioning. Firstly the database's size is growing if not additional measures are used. This causes the search time to increase and will ultimately lead to too slow predictions for the standard search algorithms. For the discretized progressive prediction this problem is less pronounced, and thus, this approach is more suitable for learning. The second aspect here is that each new trajectory only contains information until right before the catching instant as the ball is then hidden in the catching device and the position tracking cannot be done successfully. This causes the trajectories to be incomplete. Additional steps to prolong the trajectory throughout the catching area are required in case the trajectory shall be usable in a future prediction. Here different approaches can

be evaluated: prolonging the trajectory based on a number of similar trajectories in the database (off-line and thus more complex possible), using feedback from the visual system and the robot (Was it a successful catch? In which relation to the catching device's center was the ball?) to pick the right extension of the trajectory.

Transport-by-Throwing

Closing the loop the initial motivation, the transport-by-throwing approach, the usage in industrial environments requires additional efforts in all areas. The industrial environment with a changing background raises demands for more complex image processing algorithms, the more complex shape of the processed goods raises the demands for the detection system, also requiring orientation information and many other factors limit the implementation in this environment. Additionally, the success rate of one individual transport hop has to be very close to 100 % in order to have an economically feasible transportation system. This rate is still a seemingly unachievable goal, especially if the complexity of the goods is rising.

A APPENDIX

Figure showing the deviations of the launching position determined by the parameter estimation in Section 4.2.2.1

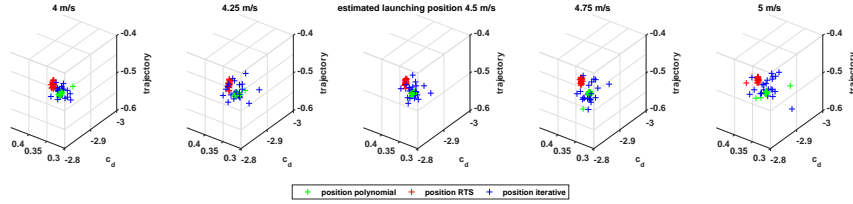


Figure A.1: Graphical illustration of the estimated launching position for the three estimation methods; polynomial: green; Rauch-Tung-Striebel smoother: red; iterative: blue

Figure showing the deviations of the drag coefficient and mean position error determined by the parameter estimation in Section 4.2.2.1

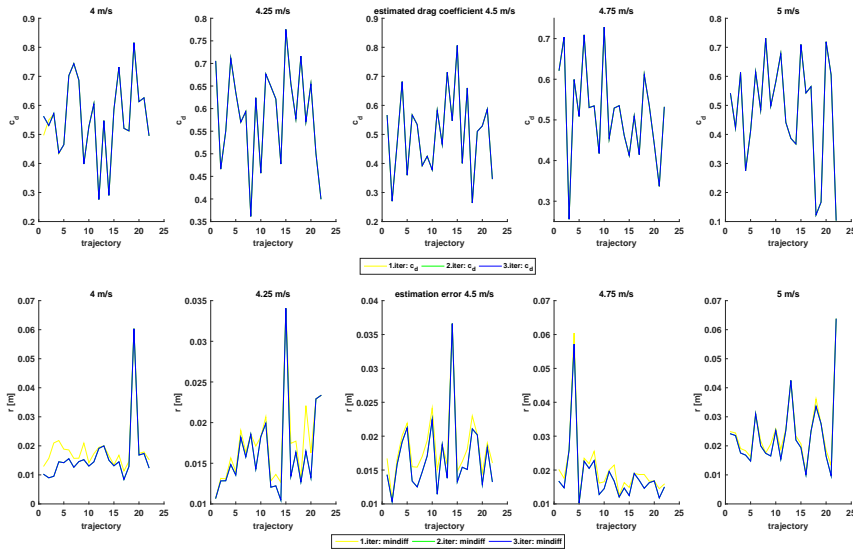


Figure A.2: Graphical illustration of the estimated drag coefficient (top row) and the average estimation error in regards of the ball's position (bottom row) for the iterative estimation model per examined trajectory

Figure A.3 is showing the position detection error based on fitted polynomial functions to each individual flight (ground truth) for Hough and without background subtraction.

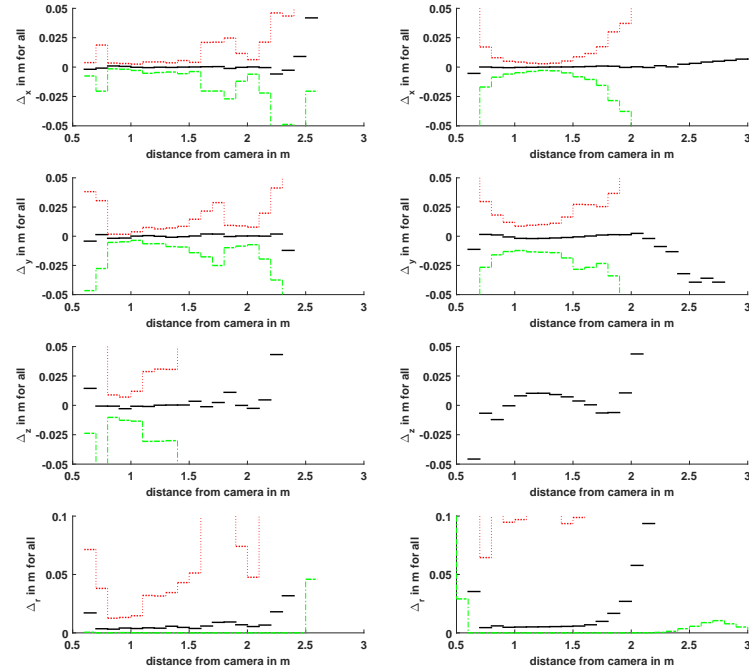


Figure A.3: Error analysis of ball's position detection accuracy without background subtraction. 99 % confidence bands of distributions (x/y/z: Normal; r: Weibull); left column Hough, right column RANSAC with individual ground truth, for all two cases polynomial functions are used to estimate the ground truth data from the complete dataset of the trajectory

Figure A.4 is showing the position detection error based on fitted polynomial functions to each individual flight (ground truth) for Hough and with background subtraction.

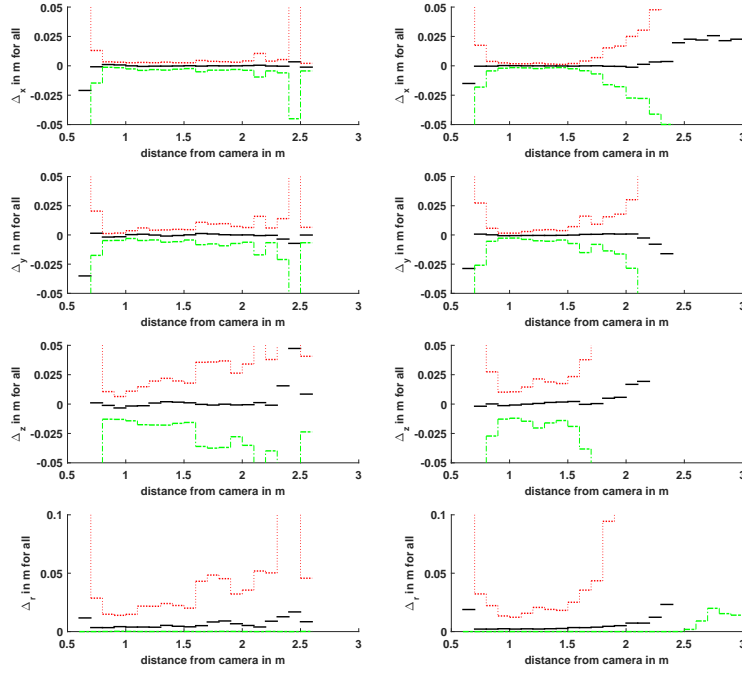


Figure A.4: Error analysis of ball's position detection accuracy with background subtraction. 99 % confidence bands of distributions (x/y/z: Normal; r: Weibull); left column Hough, right column RANSAC with individual ground truth, for all two cases polynomial functions are used to estimate the ground truth data from the complete dataset of the trajectory

LITERATURE

- [Akh11] AKHTER, Naeem: *Visual tracking of mechanically thrown objects with planar surfaces*, Vienna University of Technology, Department for Electrical Engineering and Information Technology, Institute for Computer Technology, PhD Dissertation, 2011
- [ATW⁺07] ALAM, F. ; TIO, W. ; WATKINS, S. ; SUBIC, A. ; NASER, J.: Effects of Spin on Tennis Ball Aerodynamics: An Experimental and Computational Study. (2007)
- [Bar10] BARTEIT, Dennis F.: *Tracking of thrown objects*, Vienna University of Technology, Department for Electrical Engineering and Information Technology, Institute for Computer Technology, PhD Dissertation, 2010
- [BBW⁺11] BAUML, B. ; BIRBACH, O. ; WIMBOCK, T. ; FRESE, U. ; DIETRICH, A. ; HIRZINGER, G.: Catching flying balls with a mobile humanoid: System overview and design considerations. In: *Humanoid Robots (Humanoids), 2011 11th IEEE-RAS International Conference on*, 2011. – ISSN 2164–0572, S. 513–520
- [BC70] BLAKEMORE, Gavin ; COOPER, Grahame F.: Development of the Brain depends on the Visual Environment. In: *Nature* 228 (1970), Oct, Nr. 5270, S. 477–478
- [BEC09] BRUCE ELLIOTT, Machar R. ; CRESPO, Miguel: New ITF Publication - Technique Development In Tennis Stroke Production. In: *Coaching & Sport Science Review* 49 (2009), 12, S. 26–26. ISBN 18122302
- [BF09] BIRBACH, Oliver ; FRESE, Udo: A Multiple Hypothesis Approach for a Ball Tracking System. In: FRITZ, Mario (Hrsg.) ; SCHIELE, Bernt (Hrsg.) ; PIATER, JustusH. (Hrsg.): *Computer Vision Systems* Bd. 5815. Springer Berlin Heidelberg, 2009. – ISBN 978–3–642–04666–7, S. 435–444
- [BF11] BIRBACH, O. ; FRESE, U.: Estimation and prediction of multiple flying balls using Probability Hypothesis Density filtering. In: *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, 2011. – ISSN 2153–0858, S. 3426–3433
- [BFB11] BIRBACH, O. ; FRESE, U. ; BAUML, B.: Realtime perception for catching a flying ball with a mobile humanoid. In: *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, 2011. – ISSN 1050–4729, S. 5955–5962
- [BFK08] BARTEIT, D. ; FRANK, H. ; KUPZOG, F.: Accurate prediction of interception positions for catching thrown objects in production systems. In: *Industrial Informatics, 2008. INDIN 2008. 6th IEEE International Conference on*, 2008. – ISSN 1935–4576, S. 893 –898
- [BFPK09] BARTEIT, D. ; FRANK, H. ; PONGRATZ, M. ; KUPZOG, F.: Measuring the intersection of a thrown object with a vertical plane. In: *Industrial Informatics, 2009.*

- INDIN 2009. *7th IEEE International Conference on*, 2009. – ISSN 1935–4576, S. 680–685
- [BJ78] BAILEY, T. ; JAIN, A. K.: A Note on Distance-Weighted k-Nearest Neighbor Rules. In: *Systems, Man and Cybernetics, IEEE Transactions on* 8 (1978), April, Nr. 4, S. 311–313. – ISSN 0018–9472
- [BS03] BIBER, P. ; STRASSER, W.: The normal distributions transform: a new approach to laser scan matching. In: *Intelligent Robots and Systems, 2003. (IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on* Bd. 3, 2003, S. 2743–2748 vol.3
- [BSC07] BRATT, M. ; SMITH, C. ; CHRISTENSEN, H.I.: Minimum jerk based prediction of user actions for a ball catching task. In: *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, 2007, S. 2710–2716
- [BSW⁺11] BAUML, B. ; SCHMIDT, F. ; WIMBOCK, T. ; BIRBACH, O. ; DIETRICH, A. ; FUCHS, M. ; FRIEDL, W. ; FRESE, U. ; BORST, C. ; GREBENSTEIN, M. ; EIBERGER, O. ; HIRZINGER, G.: Catching flying balls and preparing coffee: Humanoid Rollin’Justin performs dynamic and sensitive tasks. In: *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, 2011. – ISSN 1050–4729, S. 3443–3444
- [Bue11] BUETI, Domenica: The sensory representation of time. In: *Frontiers in Integrative Neuroscience* 5 (2011), Nr. 34. – ISSN 1662–5145
- [BV10] BHATIA, Nitin ; VANDANA: Survey of Nearest Neighbor Techniques. In: *CoRR* abs/1007.0085 (2010)
- [BWH10] BAUML, B. ; WIMBOCK, T. ; HIRZINGER, G.: Kinematically optimal catching a flying ball with a hand-arm-system. In: *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, 2010. – ISSN 2153–0858, S. 2592–2599
- [BYW⁺10] BATZ, G. ; YAQUB, A. ; WU, Haiyan ; KUHNLENZ, K. ; WOLLHERR, D. ; BUSS, M.: Dynamic manipulation: Nonprehensile ball catching. In: *Control Automation (MED), 2010 18th Mediterranean Conference on*, 2010, S. 365–370
- [CA98] COX, W. M. ; ALM, Richard: The right stuff: America’s move to mass customization. In: *Annual Report* (1998), S. 3–26
- [CdPL12] CESQUI, Benedetta ; D’AVELLA, Andrea ; PORTONE, Alessandro ; LACQUANITI, Francesco: Catching a Ball at the Right Time and Place: Individual Factors Matter. In: *PLoS ONE* 7 (2012), 02, Nr. 2, S. e31770
- [Col08] COLVIN, Geoff: *Talent is overrated: Confronting the unexpected facts about innate abilities*. 1. Penguin Group, 2008
- [DGAB03] DAVIDS, Keith ; GLAZIER, Paul ; ARAËJO, Duarte ; BARTLETT, Roger: Movement Systems as Dynamical Systems. In: *Sports Medicine* 33 (2003), Nr. 4, S. 245–260. – ISSN 0112–1642
- [DGLV08] DEUTSCH, T. ; GRUBER, A. ; LANG, R. ; VELIK, R.: Episodic memory for autonomous agents. In: *Human System Interactions, 2008 Conference on*, 2008, S. 621–626
- [DPBB05] DESSING, Joost C. ; PEPPER, C. (Lieke) E. ; BULLOCK, Daniel ; BEEK, Peter J.: How Position, Velocity, and Temporal Information Combine in the Prospective Control of Catching: Data and Model. (2005)
- [Dun13] DUNLOP, J.: Free Flight Aerodynamics of Sports balls. (2013)
- [EKTR93] ERICSSON, Anders ; KRAMPE, Ralf T. ; TESCH-ROMER, Clemens: The Role of Deliberate Practice in the Acquisition of Expert Performance. In: *Psychological Review* 100 (1993), S. 363–406

- [FBH⁺01] FRESE, U. ; BAUML, B. ; HAIDACHER, S. ; SCHREIBER, G. ; SCHAEFER, I. ; HAHNLE, M. ; HIRZINGER, G.: Off-the-shelf vision for a robotic ball catcher. In: *Intelligent Robots and Systems, 2001. Proceedings. 2001 IEEE/RSJ International Conference on* Bd. 3, 2001, S. 1623–1629 vol.3
- [FBM⁺08] FRANK, H. ; BARTEIT, D. ; MEYER, M. ; MITTNACHT, A. ; NOVAK, G. ; MAHLKNECHT, S.: Optimized Control Methods for Capturing Flying Objects with a Cartesian Robot. In: *Robotics, Automation and Mechatronics, 2008 IEEE Conference on*, 2008, S. 160 –165
- [FH85] FLASH, Tamar ; HOGANS, Neville: The Coordination of Arm Movements: An Experimentally Confirmed Mathematical Model. In: *Journal of neuroscience* 5 (1985), S. 1688–1703
- [FHH⁺00] FALK, R. J. ; HENDERSON, J. M. ; HOLLINGWORTH, A. ; MAHADEVAN, S. ; DYER, F. C.: Eye movements in human face learning and recognition. In: *Annual Meeting of the Cognitive Science Society*, 2000. – ISSN 2164–0572, S. 513–520
- [FMS09] FRANK, H. ; MITTNACHT, A. ; SCHEIERMANN, J.: Throwing of cylinder-shaped objects. In: *Advanced Intelligent Mechatronics, 2009. AIM 2009. IEEE/ASME International Conference on*, 2009, S. 59 –64
- [FMVDS12] FLIGGE, N. ; MCINTYRE, J. ; VAN DER SMAGT, P.: Minimum jerk for human catching movements in 3D. In: *Biomedical Robotics and Biomechatronics (BioRob), 2012 4th IEEE RAS EMBS International Conference on*, 2012. – ISSN 2155–1774, S. 581–586
- [FNTI06] FURUKAWA, N. ; NAMIKI, A. ; TAKU, S. ; ISHIKAWA, M.: Dynamic regrasping using a high-speed multifingered hand and a high-speed vision system. In: *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, 2006. – ISSN 1050–4729, S. 181–187
- [Foe93] VON FOERSTER, Heinz ; SCHMIDT, Siegfried J. (Hrsg.): *Wissen und Gewissen: Versuch einer Brücke*. 7. Suhrkamp Verlag, 1993. – ISBN 3518284762
- [Foe99] VON. FOERSTER, Heinz ; VON. FOERSTER, Heinz (Hrsg.): *Sicht und Einsicht - Versuche zu einer operativen Erkenntnistheorie*. 1. Heidelberg : Carl-Auer-Systeme Verlag, 1999
- [Fra12] FRANK, Thorsten: *Zur Automatisierung des sanften Werfens und Fangens von Objekten zum Zwecke des innerbetrieblichen Transports*, Bergischen Universität Wuppertal, Fachbereich D: Architektur, Bauingenieurwesen, Maschinenbau, Sicherheitstechnik, PhD Dissertation, 2012
- [FWWH⁺06] FRANK, H. ; WELLERDICK-WOJTASIK, N. ; HAGEBEUKER, B. ; NOVAK, G. ; MAHLKNECHT, S.: Throwing Objects – A bio-inspired Approach for the Transportation of Parts. In: *Robotics and Biomimetics, 2006. ROBIO '06. IEEE International Conference on*, 2006, S. 91 –96
- [Gĩ5] GÖTZINGER, Maximilian: *Object Detection and Flightpath Prediction*, TU Wien, Department for Electrical Engineering and Information Technology, Institute for Computer Technology, Diplomarbeit, 2015
- [GBP13] GAVAZZI, Gioele ; BISIO, Ambra ; POZZO, Thierry: Time perception of visual motion is tuned by the motor representation of human actions. (2013)
- [GH03] GÜNTNER, W.A. ; HEINECKER, M.: *Modulare Materialflusssysteme für wandelbare Fabrikstrukturen - Bewertungs- und Gestaltungsrichtlinien für wandelbare Materialflusssysteme*. (2003)
- [GH06] GÜNTNER, W.A. ; HEINECKER, M.: *Modulare Materialflusssysteme für wandelbare Fabrikstrukturen - Bewertungs- und Gestaltungsrichtlinien für wandelbare*

- Materialflusssysteme. (2006)
- [GHW06] GÜNTNER, W.A. ; HEINECKER, M. ; WILKE, M.: Materialflusssysteme für wandelbare Fabrikstrukturen. (2006)
- [GRF06] GÜNTNER, W.A. ; R., Chisu ; F., Kuzmany: Trends und Perspektiven in der Intralogistik. In: *15. Deutscher Materialfluss-Kongress "Intralogistik - Heute -Morgen - Übermorgen"*, VDI Verlag GmbH, 2006, S. 279 –289
- [GW02] GÜNTNER, W.A. ; WILKE, M.: Anforderungen an automatisierte Materialflusssysteme für wandelbare Logistikstrukturen. In: *Tagungsband Wissenschaftssymposium Logistik der BVL 2002*, 2002, S. 335 –345
- [GW03] GÜNTNER, W.A. ; WILKE, M.: Materialflusstechnologie - Anforderungen und Konzepte für wandelbare Materialflusssysteme. (2003)
- [HES95] HONG, Won ; JACQUES E. SLOTINE, Jean: Experiments in Hand-Eye Coordination Using Active Vision. In: *LECTURE NOTES IN CONTROL AND INFORMATION SCIENCES*, Springer-Verlag, 1995, S. 130–139
- [Hoh14] HOHWY, Jakob: *The Predictive Mind*. Oxford University Press, 2014. – ISBN 0199686734
- [HS91] HOVE, Barbara ; SLOTINE, Jean-Jacques E.: Experiments in Robotic Catching. In: *American Control Conference, 1991* (1991), S. 380–386
- [HSAS⁺02] HIRZINGER, G. ; SPORER, N. ; ALBU-SCHAFER, A. ; HAHNLE, M. ; KRENN, R. ; PASCUCCI, A. ; SCHEDL, M.: DLR's torque-controlled light weight robot III-are we reaching the technological limits now? In: *Robotics and Automation, 2002. Proceedings. ICRA '02. IEEE International Conference on* Bd. 2, 2002, S. 1710–1716 vol.2
- [Hubss] HUBER, Manuel: *Real-time Communication Architecture and Optimization for Robotic Catching*, TU Wien, Department for Electrical Engineering and Information Technology, Institute for Computer Technology, Diplomarbeit, in progress
- [IMT92] ISHIKAWA, M. ; MORITA, A. ; TAKAYANAGI, N.: High Speed Vision System Using Massively Parallel Processing. In: *Intelligent Robots and Systems, 1992., Proceedings of the 1992 IEEE/RSJ International Conference on* Bd. 1, 1992. – ISSN 1, S. 373–377
- [INHI04] IMAI, Y. ; NAMIKI, A. ; HASHIMOTO, K. ; ISHIKAWA, M.: Dynamic active catching using a high-speed multifingered hand and a high-speed vision system. In: *Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on* Bd. 2, 2004. – ISSN 1050–4729, S. 1849–1854 Vol.2
- [INI96] ISHII, I. ; NAKABO, Y. ; ISHIKAWA, M.: Target tracking algorithm for 1 ms visual feedback system using massively parallel processing. In: *Robotics and Automation, 1996. Proceedings., 1996 IEEE International Conference on* Bd. 3, 1996. – ISSN 1050–4729, S. 2309–2314 vol.3
- [JWBF01] JOHANSSON, Roland S. ; WESTLING, Göran ; BÄCKSTRÖM, Anders ; FLANAGAN, J. R.: Eye-hand coordination in object manipulation. In: *JOURNAL OF NEUROSCIENCE* 21 (2001), Nr. 17, S. 6917–6932
- [KB12] KIM, Seungsu ; BILLARD, Aude: Estimating the non-linear dynamics of free-flying objects. In: *Robotics and Autonomous Systems* 60 (2012), Nr. 9, S. 1108 – 1122. – ISSN 0921–8890
- [KGB10] KIM, Seungsu ; GRIBOVSKAYA, E. ; BILLARD, A.: Learning motion dynamics to catch a moving object. In: *Humanoid Robots (Humanoids), 2010 10th IEEE-RAS International Conference on*, 2010, S. 106–111
- [KGM12] KOBER, Jens ; GLISSON, Matthew ; MISTRY, Michael: Playing catch and juggling

- with a humanoid robot. In: *2012 12th IEEE-RAS International Conference on Humanoid Robots (Humanoids 2012)*. Osaka, Japan : IEEE, 2012, S. 875 – 881
- [KIIY03] KOMURO, T. ; ISHII, I. ; ISHIKAWA, M. ; YOSHIDA, A.: A digital vision chip specialized for high-speed target tracking. In: *Electron Devices, IEEE Transactions on* 50 (2003), Nr. 1, S. 191–199. – ISSN 0018–9383
- [KMK⁺10] KOBER, J. ; MÜLLING, K. ; KRÖMER, O. ; LAMPERT, C.H. ; SCHÖLKOPF, B. ; PETERS, J.: Movement templates for learning of hitting and batting. In: *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, 2010. – ISSN 1050–4729, S. 853 –858
- [KO04] KASPROWSKI, Pawel ; OBER, Józef: Eye Movements in Biometrics. In: MALTONI, Davide (Hrsg.) ; JAIN, AnilK. (Hrsg.): *Biometric Authentication* Bd. 3087. Springer Berlin Heidelberg, 2004. – ISBN 978–3–540–22499–0, S. 248–258
- [KS11] KWON, Woo Y. ; SUH, Il-Hong: Towards proactive assistant robots for human assembly tasks. In: *Human-Robot Interaction (HRI), 2011 6th ACM/IEEE International Conference on*, 2011. – ISSN 2167–2121, S. 175–176
- [KSB14] KIM, S. ; SHUKLA, A. ; BILLARD, A.: Catching Objects in Flight. In: *Robotics, IEEE Transactions on* PP (2014), Nr. 99, S. 1–17. – ISSN 1552–3098
- [KSOI99] KAJIKAWA, S. ; SAITO, M. ; OHBA, K. ; INOOKA, H.: Analysis of human arm movement for catching a moving object. In: *Systems, Man, and Cybernetics, 1999. IEEE SMC '99 Conference Proceedings. 1999 IEEE International Conference on* Bd. 2, 1999. – ISSN 1062–922X, S. 698–703 vol.2
- [KUK10] KUKA, GmbH R.: *Leichtbauroboter 4+ Spezifikation*. Version: Spez LBR 4+ V1 de. Zugspitzstr. 140, 86165 Augsburg: KUKA Robotics, 04 2010
- [KUK11] KUKA, GmbH R.: *KUKA System Software 5.6 lr*. Version: KSS 5.6 lr SI V5 de. Zugspitzstr. 140, 86165 Augsburg: KUKA Robotics, 12 2011
- [KW10] KROGER, T. ; WAHL, F.M.: Online Trajectory Generation: Basic Concepts for Instantaneous Reactions to Unforeseen Events. In: *Robotics, IEEE Transactions on* 26 (2010), feb., Nr. 1, S. 94 –111. – ISSN 1552–3098
- [KYH13] KAO, Sho-Tsung ; YANG, Zong-Yu ; HO, Ming-Tzu: Design and implementation of a color-based visual tracking control system. In: *Automatic Control Conference (CACS), 2013 CACS International*, 2013, S. 371–376
- [LM98] LYNCH, Kevin M. ; MASON, Matthew T.: Dynamic Nonprehensile Manipulation: Controllability, Planning, and Experiments. In: *International Journal of Robotics Research* 18 (1998), S. 64–92
- [LP10] LAMPERT, Christoph ; PETERS, Jan: Real-time detection of colored objects in multiple camera streams with off-the-shelf hardware components. In: *Journal of Real-Time Image Processing* (2010), S. 1–11. – ISSN 1861–8200
- [LRÅJ09] LINDEROTH, M. ; ROBERTSSON, A. ; ÅSTRÖM, K. ; JOHANSSON, R.: Vision Based Tracker for Dart-Catching Robot. (2009), S. 883–888
- [LRÅJ10] LINDEROTH, M. ; ROBERTSSON, A. ; ÅSTRÖM, K. ; JOHANSSON, R.: Object tracking with measurements from single or multiple cameras. In: *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, 2010. – ISSN 1050–4729, S. 4525–4530
- [LZL06] LIU, Junchuan ; ZHANG, Yuru ; LI, Zhen: Selection of Cameras Setup Geometry Parameters in Binocular Stereovision. In: *Robotics, Automation and Mechatronics, 2006 IEEE Conference on*, 2006, S. 1 –6
- [MAS08] MEHTA, Rabindra ; ALAM, Firoz ; SUBIC, Aleksandar: Review of tennis ball aerodynamics. In: *Sports Technology* 1 (2008), Nr. 1, S. 7–16. – ISSN 1934–6190

- [Miy06] MIYAKE, Dario I.: The Shift from Belt Conveyor Line to Work-cell Based Assembly Systems to Cope with Increasing Demand Variation and Fluctuation in The Japanese Electronics Industries. (2006)
- [MKP10] MULLING, K. ; KOBER, J. ; PETERS, J.: A biomimetic approach to robot table tennis. In: *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, 2010. – ISSN 2153–0858, S. 1921–1926
- [MNBK05] M., Hayhoe ; N., Mennie ; B., Sullivan ; K., Gorgos: The role of internal models and prediction in catching balls. (2005)
- [MP01] MEHTA, Rabindm D. ; PALLIS, Jani M.: Sports Ball Aerodynamics: Effects of Velocity, Spin and Surface Roughness. In: *Materials and Scienec in Sports* (2001), S. 185–197
- [MSA13] MANN, David L. ; SPRATFORD, Wayne ; ABERNETHY, Bruce: The Head Tracks and Gaze Predicts: How the World’s Best Batters Hit a Ball. In: *PLoS ONE* 8 (2013), 03, Nr. 3, S. e58289
- [MSFB00] MEHRANDEZH, M. ; SELA, N.M. ; FENTON, R.G. ; BENHABIB, B.: Robotic interception of moving objects using an augmented ideal proportional navigation guidance technique. In: *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on* 30 (2000), May, Nr. 3, S. 238–250. – ISSN 1083–4427
- [MSML07] MAZYN, Liesbeth I. ; SAVELSBERGH, Geert J. ; MONTAGNE, Gilles ; LENOIR, Matthieu: Planning and on-line control of catching as a function of perceptual-motor constraints. In: *Acta Psychologica* 126 (2007), Nr. 1, S. 59 – 78. – ISSN 0001–6918
- [MZBL01] MCINTYRE, J. ; ZAGO, M. ; BERTHOZ, A. ; LACQUANITI, F.: Does the brain model Newton’s laws? In: *Nature Neuroscience* 4 (2001), jul, Nr. 7, S. 693–694
- [NHI03] NAMIKI, Akio ; HASHIMOTO, Koichi ; ISHIKAWA, Masatoshi: A Hierarchical Control Architecture for High-Speed Visual Servoing. In: *The International Journal of Robotics Research* 22 (2003), Nr. 10-11, S. 873–888
- [NI03] NAMIKI, A. ; ISHIKAWA, M.: Robotic catching using a direct mapping from visual information to motor command. In: *Robotics and Automation, 2003. Proceedings. ICRA ’03. IEEE International Conference on* Bd. 2, 2003. – ISSN 1050–4729, S. 2400–2405 vol.2
- [NI05] NAMIKI, A. ; ISHIKAWA, M.: The Analysis of High-speed Catching with a Multi-fingered Robot Hand. In: *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, 2005, S. 2655–2660
- [NII02] NAKABO, Y. ; ISHI, I. ; ISHIKAWA, M.: 3D tracking using two high-speed vision systems. In: *Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on* Bd. 1, 2002, S. 360–365 vol.1
- [NIIK03] NAMIKI, A. ; IMAI, Y. ; ISHIKAWA, M. ; KANEKO, M.: Development of a high-speed multifingered hand system and its application to catching. In: *Intelligent Robots and Systems, 2003. (IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on* Bd. 3, 2003, S. 2666–2671 vol.3
- [NIN+97] NISHIWAKI, K. ; IONNO, A. ; NAGASHIMA, K. ; INABA, M. ; INOUE, H.: The humanoid Saika that catches a thrown ball. In: *Robot and Human Communication, 1997. RO-MAN ’97. Proceedings., 6th IEEE International Workshop on*, 1997, S. 94–99
- [NIO+99] NAKANO, Eri ; IMAMIZU, Hiroshi ; OSU, Rieko ; UNO, Yoji ; GOMI, Hiroaki ; YOSHIOKA, Toshinori ; KAWATO, Mitsuo: Quantitative Examinations of Internal Representations for Arm Trajectory Planning: Minimum Commanded Torque

- Change Model. In: *Journal of Neurophysiology* 81 (1999), Nr. 5, S. 2140–2155. – ISBN 1522–1598
- [NITM00] NAKABO, Y. ; ISHIKAWA, M. ; TOYODA, H. ; MIZUNO, S.: 1 ms column parallel vision system and its application of high speed target tracking. In: *Robotics and Automation, 2000. Proceedings. ICRA '00. IEEE International Conference on* Bd. 1, 2000. – ISSN 1050–4729, S. 650–655 vol.1
- [NMH02] NOVAK, K.E. ; MILLER, L.E. ; HOUK, J.C.: The use of overlapping submovements in the control of rapid hand movements. In: *Experimental Brain Research* 144 (2002), Nr. 3, S. 351–364. – ISSN 0014–4819
- [NNII99] NAMIKI, A. ; NAKABO, Y. ; ISHII, I. ; ISHIKAWA, M.: High speed grasping using visual and force feedback. In: *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on* Bd. 4, 1999. – ISSN 1050–4729, S. 3195–3200 vol.4
- [OEG13] OIKE, Y. ; EL GAMAL, A.: CMOS Image Sensor With Per-Column Sum Delta ADC and Programmable Compressed Sensing. In: *Solid-State Circuits, IEEE Journal of* 48 (2013), Nr. 1, S. 318–328. – ISSN 0018–9200
- [OTI+09] OKUDA, H. ; TAKEUCHI, H. ; INAGAKI, S. ; SUZUKI, T. ; HAYAKAWA, S.: Understanding of positioning skill based on feedforward / feedback switched dynamical model. In: *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, 2009, S. 3057–3062
- [PBMB94] PEPPER, L. ; BOOTSMA, R. J. ; MESTRE, D. R. ; BAKKER, F. C.: Catching balls: how to get the hand to the right place at the right time. In: *J Exp Psychol Hum Percept Perform* 20 (1994), Jun, Nr. 3, S. 591–612
- [PEG05] PFOHL, Hans-Christian ; ELBERT, Ralf ; GOMM, Moritz: *Auf Augenhöhe mit dem Top-Management? : Wert schaffen statt nur Kosten senken?* 1. Hamburg : Dt. Verkehrs-Verlag, Frank Straube, Januar 2005
- [Ph.13] PH.D., Jeffrey C. I.: *Motor Behavior: Connecting Mind and Body for Optimal Performance*. LWW, 2013. – ISBN 1451175892
- [Pil01] PILLER, F.T.: *Mass Customization.: Ein wettbewerbsstrategisches Konzept im Informationszeitalter*. 1. Deutscher Universitätsvlg, 2001 (Markt- und Unternehmensentwicklung). – ISBN 9783824474769
- [PKFB10] PONGRATZ, M. ; KUPZOG, F. ; FRANK, H. ; BARTEIT, D.: Transport by throwing - A bio-inspired approach. In: *Industrial Informatics (INDIN), 2010 8th IEEE International Conference on*, 2010, S. 685 –689
- [PKK+09] PARK, Ga-Ram ; KIM, KangGeon ; KIM, Chang H. ; JEONG, Mun-Ho ; YOU, Bum-Jae ; RA, Syungkwon: Human-like catching motion of humanoid using Evolutionary Algorithm(EA)-based imitation learning. In: *Robot and Human Interactive Communication, 2009. RO-MAN 2009. The 18th IEEE International Symposium on*, 2009. – ISSN 1944–9445, S. 809–815
- [Pon09] PONGRATZ, Martin: *Object Touchdown Position Prediction*, Vienna University of Technology, Department for Electrical Engineering and Information Technology, Institute for Computer Technology, Diplomarbeit, 2009
- [Pra06] PRATL, Gerhard: *Processing and symbolization of ambient sensor data*, Vienna University of Technology, Department for Electrical Engineering and Information Technology, Institute for Computer Technology, PhD Dissertation, 2006
- [P.V62] P.V.C., Hough: *Method and means for recognizing complex patterns*. December 1962. – US Patent 3,069,654
- [RA00] RILEY, Marcia ; ATKESON, Christopher G.: Methods for motion generation and

- interaction with a humanoid robot: Case studies of dancing and catching. In: *Proc. 2000 Workshop on Interactive Robotics and Entertainment, Robotics Inst., Carnegie Mellon Univ*, 2000, S. 35–42
- [RA02] RILEY, Marcia ; ATKESON, ChristopherG.: Robot Catching: Towards Engaging Human-Humanoid Interaction. In: *Autonomous Robots* 12 (2002), Nr. 1, S. 119–128. – ISSN 0929–5593
- [SC07] SMITH, C. ; CHRISTENSEN, H.I.: Using COTS to Construct a High Performance Robot Arm. In: *Robotics and Automation, 2007 IEEE International Conference on*, 2007. – ISSN 1050–4729, S. 4056–4063
- [SFXT13] SU, Hu ; FANG, Zaojun ; XU, De ; TAN, Min: Trajectory Prediction of Spinning Ball Based on Fuzzy Filtering and Local Modeling for Robotic Ping-Pong Player. In: *Instrumentation and Measurement, IEEE Transactions on* 62 (2013), Nov, Nr. 11, S. 2890–2900. – ISSN 0018–9456
- [SGC98] STEVE G. CHADWICK, Steve J. H.: Determination of the coefficient of drag of three different tennis balls using a wind tunnel. (1998)
- [SLW⁺09] SUN, Lei ; LIU, Jingtai ; WANG, Yingshi ; ZHOU, Lu ; YANG, Qi ; HE, Shan: Ball's flight trajectory prediction for table-tennis game by humanoid robot. In: *Robotics and Biomimetics (ROBIO), 2009 IEEE International Conference on*, 2009, S. 2379–2384
- [SMKB13] SALVIONI, Paolo ; MURRAY, Micah M. ; KALMBACH, Lysiann ; BUETI, Domenica: How the Visual Brain Encodes and Keeps Track of Time. In: *The Journal of Neuroscience* 33 (2013), Nr. 30, S. 12423–12429
- [SNI04] SENOO, T. ; NAMIKI, A. ; ISHIKAWA, M.: High-speed batting using a multi-jointed manipulator. In: *Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on* Bd. 2, 2004. – ISSN 1050–4729, S. 1191–1196 Vol.2
- [SNI05] SHIOKATA, D. ; NAMIKI, A. ; ISHIKAWA, M.: Robot dribbling using a high-speed multifingered hand and a high-speed vision system. In: *Intelligent Robots and Systems, 2005. (IROS 2005). 2005 IEEE/RSJ International Conference on*, 2005, S. 2097–2102
- [Sol02] SOLMS, O.: *The brain and the inner world: An introduction to the neuroscience of subjective experience*. Rev 2003. London : Karnac, 2002
- [SS98] SCHAAL, Stefan ; STERNAD, Dagmar: Programmable pattern generators. In: *3rd International Conference on Computational Intelligence in Neuroscience* Citeseer, 1998, S. 48–51
- [TBF05] THRUN, S. ; BURGARD, W. ; FOX, D.: *Probabilistic Robotics*. MIT Press, 2005 (Intelligent robotics and autonomous agents). – ISBN 9780262201629
- [TS00] TEVATIA, G. ; SCHAAL, S.: Inverse kinematics for humanoid robots. In: *Robotics and Automation, 2000. Proceedings. ICRA '00. IEEE International Conference on* Bd. 1, 2000. – ISSN 1050–4729, S. 294–299 vol.1
- [UKH⁺13] UCHIDA, Yusuke ; KUDOH, Daisuke ; HIGUCHI, Takatoshi ; HONDA, Masaaki ; KAZUYUKI, Kanosue: Dynamic Visual Acuity in Baseball Players Is Due to Superior Tracking Abilities. (2013)
- [Vel07] VELIK, R.: A model for multimodal humanlike perception based on modular hierarchical symbolic information processing, knowledge integration, and learning. In: *Bio-Inspired Models of Network, Information and Computing Systems, 2007. Bionetics 2007. 2nd*, 2007, S. 168 –175
- [Vel08] VELIK, Rosemarie: *A Bionic Model for Human-like Machine Perception*.

- Saarbrücken, Germany, Germany : VDM Verlag, 2008. – ISBN 3838100506, 9783838100500
- [WGJ95] WOLPERT, Daniel ; GHARAMANI, Zoubin ; JORDAN, Michael I. *Are arm trajectories planned in kinematic or dynamic coordinates? An adaptation study.* 1995
- [Wie05] WIENDAHL, H.P.: *Planung modularer Fabriken.* 1. Hanser, 2005. – ISBN 9783446400450

INTERNET REFERENCES

- [1] Bidl about-aims. <http://bidl.tongji.edu.cn/AboutBiomimicry.html>.
- [2] Colin blakemore does terrible things to kitteh.z. for science! <https://www.youtube.com/watch?v=QzkMo45pcUo>.
- [3] Dlr - institut für robotik und mechatronik - human like ball catching. http://www.dlr.de/rm/desktopdefault.aspx/tabid-3818/6241_read-9007/.
- [4] Gravity in the brain - nasa science. http://science.nasa.gov/science-news/science-at-nasa/2002/18mar_playingcatch/.
- [5] mlbtv colmia did you catch that vine. <https://www.youtube.com/watch?v=QohWU1-rc7g>.
- [6] AP, editor. *Ivo Karlovic Sets World Fastest Serve Record At Davis Cup (VIDEO)*. The Huffington Post, 2013. http://www.huffingtonpost.com/2011/03/06/ivo-karlovic-fastest-serve-record_n_831980.html.
- [7] D.-I. T. K. (CEO), editor. *Reflexxes - Online Trajectory Generation - Robotics - Home*. Reflexxes GmbH, 2012. <http://www.reflexxes.com/>.
- [8] W. F. Inc., editor. *Trägheitsmoment - Wikipedia*, 2014. <http://de.wikipedia.org/wiki/Trägheitsmoment>.
- [9] MedizInfo. July 2011. <http://www.medizininfo.de/augenheilkunde/netzhaut/anatomie.htm>.
- [10] Sloman. July 2011. <http://www.cs.bham.ac.uk/~axs/misc/talks/sloman-beyond-gibson.pdf>.