Experimental validation of an MPC-POMDP model of ball catching

Experimentelle Validierung eines MPC-POMDP Ballfangmodells

Bachelor-Thesis von Lennart Ebeling aus Darmstadt

Februar 2019

TECHNISCHE
UNIVERSITÄT
DARMSTADT

Experimental validation of an MPC-POMDP model of ball catching
Experimentelle Validierung eines MPC-POMDP Ballfangmodells

Vorgelegte Bachelor-Thesis von Lennart Ebeling aus Darmstadt

1. Gutachten: Prof. Dr. Jan Peters
2. Gutachten: Boris Belousov

Tag der Einreichung:

Erklärung zur Abschlussarbeit gemäß § 23 Abs. 7 APB der TU Darmstadt

Hiermit versichere ich, Lennart Ebeling, die vorliegende Bachelor-Thesis ohne Hilfe Dritter und nur mit den angegebenen Quellen und Hilfsmitteln angefertigt zu haben. Alle Stellen, die Quellen entnommen wurden, sind als solche kenntlich gemacht worden. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Mir ist bekannt, dass im Fall eines Plagiats (§ 38 Abs. 2 APB) ein Täuschungsversuch vorliegt, der dazu führt, dass die Arbeit mit 5,0 bewertet und damit ein Prüfungsversuch verbraucht wird. Abschlussarbeiten dürfen nur einmal wiederholt werden.

Bei der abgegebenen Thesis stimmen die schriftliche und die zur Archivierung eingereichte elektronische Fassung überein.

Bei einer Thesis des Fachbereichs Architektur entspricht die eingereichte elektronische Fassung dem vorgestellten Modell und den vorgelegten Plänen.

Datum / Date: Unterschrift / Signature:

_____ _____

Abstract

The problem of ball catching, also known as the outfielder problem, has been used for a long time to illustrate the divide between predictive and reactive behaviors of humans and animals. Recently developed computational model [1], that captures essential characteristics of the outfielder problem, has been shown capable of reproducing previously obtained experimental results and unify a number of heuristic-based approaches. Importantly, this computational model predicts a distinctive result in a particular experimental scenario that has not been studied before; namely, for high overhead balls, the catcher must turn away from the ball for a moment in order to speed up to the interception point. In this thesis, we propose an actionable experimental setting for testing the prediction of the model; crucial variables and experimental settings are identified and extensively studied in simulation to provide a solid base for experimental validation. In addition, simulation environment is significantly improved and extended with on the one hand a more accurate, realistic ball model and on the other hand a robust, singularity-free angle parameterization. These improvements result in more stable optimization. To align simulated results with experimental data, a system identification procedure is outlined that is to be performed jointly with catching experiments.

Zusammenfassung

Das Ballfangproblem, auch bekannt als Outfielder-Problem, wird seit langem genutzt, um die Diskrepanz zwischen vorausschauendem und reaktivem Verhalten von Mensch und Tier zu veranschaulichen. Es wurde gezeigt, dass ein kürzlich entwickeltes Berechnungsmodell aus [1] die wesentlichen Merkmale des Outfielder-Problems erfasst. Das Berechnungsmodell ist in der Lage, zuvor erhaltene experimentelle Ergebnisse zu reproduzieren und eine Reihe von heuristisch basierten Ansätzen zu vereinen. Wichtig ist, dass dieses Berechnungsmodell ein unverwechselbares Ergebnis in einem bestimmten experimentellen Szenario voraussagt, das bisher noch nicht untersucht wurde. Nämlich, dass sich bei langen Überkopf-Bällen der Fänger für einen Moment vom Ball abwenden muss, um zu beschleunigen und zu dem Punkt zu gelangen, an dem der Ball gefangen werden kann. In dieser Arbeit wird ein geeignetes Versuchsumfeld für die Prüfung der Vorhersage des Modells vorgeschlagen. Entscheidende Variablen und Einstellungen werden identifiziert und umfassend in der Simulation untersucht, um eine solide Grundlage für die experimentelle Validierung zu geben. Darüber hinaus wird die Simulationsumgebung deutlich verbessert und erweitert mit einerseits einem genaueren, realistischeren Ballmodell und andererseits einer robusten, singularitätsfreien Winkelparametrisierung. Diese Verbesserungen führen zu einer stabileren Optimierung der Simulation. Um simulierte Ergebnisse mit experimentellen Daten abzugleichen, wird ein Systemidentifizierungsverfahren skizziert, das gemeinsam mit den Fangexperimenten durchgeführt werden soll.

Contents

1. Introduction **2**
- 1.1. Motivation . . . 2
- 1.2. Related work . . . 2
- 1.3. Overview . . . 3

2. Baseline mathematical model of ball catching **4**
- 2.1. Model of the catcher . . . 4
- 2.2. Simple Model of the ball . . . 5
- 2.3. Optimal control model of ball catching . . . 6
- 2.4. Limitations to the baseline model . . . 8

3. Extended and improved simulation environment **9**
- 3.1. Removing singularities by using unitary angle representation . . . 9
- 3.2. Using a realistic ball model obtained from baseball data . . . 10
- 3.3. Evaluation of different covariance matrix representations . . . 11

4. Results of the simulation **12**
- 4.1. Simulation results with the extended ball model . . . 12
- 4.2. Effects of noise . . . 16
- 4.3. Model mismatch: planning with a 'wrong' model . . . 17

5. A proposal for experimental validation of the ball catching model **19**
- 5.1. General description of the experimental setting . . . 19
- 5.2. Specification of control variables . . . 20
- 5.3. System identification for the model . . . 23

6. Conclusion and outlook **25**
- 6.1. Conclusion . . . 25
- 6.2. Outlook . . . 25

Bibliography **27**

A. Some appendix results **29**
- A.1. Simulation results for experimental setups . . . 29

Figures and Tables

List of Figures

2.1. Schematic view of the catcher from above. Vector d_{xy} represents the projection of the gaze on the $xy-plane$ and vector a_c represents the controllable forece applied to the catcher. Angles φ and θ are counted relative to the positive direction of the x-axis. . . . 4
2.2. Schematic view of catcher from the side. Representation ψ as angle between the gaze d and d_xy on the xy-plane. The angle ψ counted with respect to d_{xy} and is limited by $0 \leq \psi < \pi/2$. . . . 4
2.3. Baseline ball model that only accounts for the gravitational force. . . . 6

3.1. Schematic look at catcher from above. Representation of d_{xy} and a_c and the variables $\cos\varphi$,$\sin\varphi$ $\cos\theta$ and $\sin\theta$ assosiated with them. . . . 9
3.2. Schematic representation of the extended ball model with the gravitational force, drag force and lift force acting to it. The ball has a backspin rotation and a velocity. The drag force acts in $-\vec{v}$ direction, the lift force in $\vec{\omega} \times \vec{v}$ direction and the gravitational force acts downwards. . . . 11

4.1. Comparison of the trajectory of the ball with the different models. . . . 13
4.2. Ball and catcher with gaze direction d_{xy} from above. Mean over 10 MPC simulation runs. Catchable side ball where heuristics hold. The corresponding plots for the heuristics are shown in Figure 4.3. . . . 13
4.3. Plot the heuristics OAC, CBA, GOAC and LOT corresponding to Figure 4.2. It shows the average heuristics over 10 MPC simulation runs. . . . 14
4.4. Ball and Catcher with gaze direction d_{xy} from above. Mean over 10 MPC simulation runs. The interception point of the ball is too far for the catcher to run backwards. Because of this the catcher turns around to the predicted interception points, for fast acceleration. Later he looks back to the ball. The corresponding plots for the heuristics are shown in Figure 4.5. . . . 15
4.5. Plot the heuristics OAC, CBA, GOAC and LOT corresponding to Figure 4.4. It shows the average heuristics over 10 MPC simulation runs. None of the heuristics holds although the catcher catches the ball. . . . 15
4.6. Plot the heuristics OAC, CBA, GOAC and LOT corresponding to Figure 4.7. It shows the average heuristics over 10 MPC simulation runs. . . . 16
4.7. Ball and Catcher with gaze direction d_{xy} from above. Mean over 10 MPC simulation runs. The initial noise was significantly increased. The corresponding plots for the heuristics are shown in Figure 4.6. . . . 17
4.8. Ball and Catcher with gaze direction d_{xy} from above. Mean over 10 MPC simulation runs. The simulation uses the extended ball model while for belief space planning the simple ball model is used. The first planning calculation plan towards the point $(15, 0)^T$. The simulated ball with noise lands at approximately $(14, 0)^T$. The heuristics of these simulation runs are shown in Figure 4.9. . . . 17
4.9. Plot the heuristics OAC, CBA, GOAC, and LOT corresponding the plots of the catcher and the ball in Figure 4.8. It shows the average heuristics over 10 MPC simulation runs. . . . 18

5.1. Schematic illustration of the experiment field from above. . . . 20

A.1. Result of the simulation for the start position (7,4) for the catcher averaged over three MPC-simulation runs. The heuristics corresponding to this result is shown in Figure A.2 . . . 29
A.2. Heuristic (OAC, CBA, GOAC, and LOT) plots for the simulation, when the catcher start at (7,4). The corresponding result plot is shown in Figure A.1 . . . 29
A.3. Result of the simulation for the start position (7,1) for the catcher averaged over three MPC-simulation runs. The heuristics corresponding to this result is shown in Figure A.4 . . . 30
A.4. Heuristic (OAC, CBA, GOAC, and LOT) plots for the simulation, when the catcher start at 7,1). The corresponding result plot is shown in Figure A.3 . . . 30
A.5. Result of the simulation for the start position (9,4) for the catcher averaged over three MPC-simulation runs. The heuristics corresponding to this result is shown in Figure A.6 . . . 31
A.6. Heuristic (OAC, CBA, GOAC, and LOT) plots for the simulation, when the catcher start at (9,4). The corresponding result plot is shown in Figure A.5 . . . 31

A.7. Result of the simulation for the start position (9,1) for the catcher averaged over three MPC-simulation runs. The heuristics corresponding to this result is shown in Figure A.8 32
A.8. Heuristic (OAC, CBA, GOAC, and LOT) plots for the simulation, when the catcher start at 9,1). The corresponding result plot is shown in Figure A.7 32
A.9. Result of the simulation for the start position (11,4) for the catcher averaged over three MPC-simulation runs. The heuristics corresponding to this result is shown in Figure A.10 33
A.10. Heuristic (OAC, CBA, GOAC, and LOT) plots for the simulation, when the catcher start at (11,4). The corresponding result plot is shown in Figure A.9 33
A.11. Result of the simulation for the start position (15,4) for the catcher averaged over three MPC-simulation runs. The heuristics corresponding to this result is shown in Figure A.12 34
A.12. Heuristic (OAC, CBA, GOAC, and LOT) plots for the simulation, when the catcher start at (15,4). The corresponding result plot is shown in Figure A.11 34

List of Tables

5.1. Proposed start positions of the catcher when ball starts at $(0, 0)$ and the interception point is $(14, 0)$. Classification of the ball and whether the catcher has to turn his gaze towards the interception point are shown. Simulation results of these initial states are shown in the appendix. The ID is for referring to the inital states and is consecutively. 22
5.2. Proposed start positions of the catcher classification are shown for the system identification. Simulation results of these initial states are shown in the appendix. The ID is for referring to the inital states and is consecutively from Table 5.1. 24

Abbreviations, Symbols and Operators

List of Abbreviations

Notation	Description
CBA	Constant bearing angle
EKF	Extended kalman filter
GOAC	Generalized optic acceleration cancellation
LOT	Linear optic trajectory
MDP	Markov decision process
MPC	Model predictive control
OAC	Optic acceleration cancellation
POMDP	Partial observable markov decision process
RK4	4-th order explicit Runge-Kutta

List of Symbols

Notation	Description
$\mathbf{x}, \mathbf{y}, \mathbf{z}$	Coordinates [m]
$\dot{\mathbf{x}}, \ddot{\mathbf{x}}$	First and second derivative
$\mathbf{v}, \mathbf{a}$	Velocity [m/s] and acceleration [m/s^2]
$\mathbf{F_c}, \mathbf{a_c}$	Controllable Force [N] and acceleration [m/s^2] of the catcher
$\mathbf{m}, \boldsymbol{\Theta}$	Mass [kg] and moment of inertia
$\mathbf{d}, \mathbf{d_{xy}}$	Gaze direction (3D unit vector) and its projection onto the xy-plane
$\boldsymbol{\varphi}, \boldsymbol{\psi}, \boldsymbol{\omega_\varphi}, \boldsymbol{\omega_\psi}$	Angles that define d and corresponding angular velocities
$\boldsymbol{\theta}$	Angle between d_{xy} and F_c
$\mathbf{F_D}, \mathbf{F_L}$	Drag force [N] and Lift force [N]
$\mathbf{c_D}, \mathbf{c_L}$	Coefficients for drag and lift
$\boldsymbol{\omega_b}$	Rotational speed of the ball in rounds per minute (rpm)
$\vec{\boldsymbol{\omega}}_\mathbf{b}$	Rotation axis of the ball
$\boldsymbol{\lambda}$	Damping coefficient of the catcher [1/s]
$\mathbf{t}, \mathbf{k}$	Continuous time and discrete time
$\mathbf{T}, \boldsymbol{\tau}, \mathbf{N}$	Task duration, discretization time step and number of steps
$\mathbf{J}, \boldsymbol{\omega_1}, \boldsymbol{\omega_2}, \boldsymbol{\omega_3}$	Cost function and adjustable weights

$\mathbf{F_1}, \mathbf{F_2}$	Adjustable parameters for the catcher's acceleration [m/s^2]
$\mathbf{g}, \pi$	Standard gravity and ratio of a circle's circumference to its diameter
$\mathbf{I}$	Identity matrix
$\mathbf{f(x(t), u(t))}$	Continuous dynamics
$\mathbf{g(x_k, u_k)}$	Discrete dynamics
$\mathbf{h(x_k)}, \mathbf{z_k}$	Observation function and observation at time k
$\mathbf{\Sigma}$	Covariance matrix
μ	Mean
$\mathbf{N(\mu, \Sigma)}$	Normal distribution with mean μ and covariance Σ

List of Operators

Notation	Description	Operator
norm	Euclidean norm	$\|\| \bullet \|\|_2$
tr	Trace of a matrix	$\mathrm{tr}(\bullet)$

1 Introduction

This chapter starts with a motivation for this thesis, followed by the related work and an overview of the chapters of this thesis.

1.1 Motivation

The problem of ball catching, also known as the outfielder problem, is a very simple, clear scenario for figuring out how humans approach motor control problems. Heuristics can be used to explain human ball catching strategies [2],[3]. Experiments in which humans had to catch balls, showed that these heuristics could be applied [4] [5] [6]. Heuristics in general belong to an approach called specialist approach in opposite the approach is called the generalist approach [7]. In the specialist approach humans choose their actions reactively based on visual feedback whereas in the generalist approach a catcher predicts the ball trajectory to plan optimal behaviour.
In simulations it was shown that optimal control can be used to describe the catcher's behaviour [8]. A specific model of the generalist approach was developed by [1]. This model uses belief space planning in combination with model predicted control (MPC) to calculate the actions of the catcher. It can even explain those catches that cannot be explained by heuristics [9]. In the case when the catcher cannot reach the ball while always looking in the direction of the ball. So he has to turn his gaze away from the ball towards the predicted interception point, accelerate fast and then look back to the ball to catch it. This works because it is assumed that running forwards is faster than running backwards.
The aim of this thesis is on the one hand to propose a clear and detailed description of human experiments that need to be performed to verify a the MPC-POMDP model of ball catching and on the other hand to create a robust, reliable, fast simulation environment that allows for extensive experimentation and evaluation of various settings to compare simulated data with experimental results. The proposal for the experiments include a general description of the experimental setting and the specification of the control variables for the experiments as well as a validation of the model. Valuable improvements have been added to the simulation. These improvements include a two-parameter unitrary angles representation as well as a realistic ball model from baseball data that includes drag and lift forces.

1.2 Related work

There are different heuristics that explain human ball catching. The focus is on three popular theories: Chapman's theory [2], the generalized optic acceleration cancellation (GOAC) theory [6] , and the (LOT) theory [3]. Chapman proposed two heuristics: the optic acceleration cancellation (OAC) heuristic and the constant bearing angle (CBA) heuristic. All of these four heuristics (OAC, CBA, GOAC and LOT) are well-supported by experiments [4, 5, 6]. There are different types of experiments for ball catching: virtual-reality [10] where a virtual ball has to be caught, real world experiments [11] and simulated experiments [12]. In the real-world experiments there are those where humans are supposed to catch balls and those where animals are supposed to intercept objects. It can be shown that dragonflies and fish apply to CBA [13, 14]. It has also been shown that dogs use LOT when catching frisbees [15].
A different approach than using heuristics is to use optimal control. In this approach it is assumed that the catcher has an internal model to predict the ball's trajectory. This approach is called the generalist appoach [7]. It was shown that optimal control can be used to model the ball catching of a human [8]. In [1] and [9] an optimal control model was described, which uses a belief space planning method based on model predictive control. It can be shown that this method agrees with the catching heuristics. Furthermore this model can explain even successful catches that violate heuristics. Another advantage of this model is that it can deal with high Gaussian noise whereas the performance is barely affected [16].

1.3 Overview

This section provides an overview of the structure of the thesis and gives a brief summary of the topics discussed in each chapter. The thesis is structured as follows:

Chapter 2 gives an introduction to the baseline model of ball catching. The model of the catcher and the ball is presented. The following is a brief introduction to the optimal control model of ball catching, including an introduction to partially observable markov decision processes (POMDP), the transition and observation model, approximate gaussian belief dynamics and the objective function. The chapter closes with showing the system limitations of the model.

Chapter 3 explains the changes and improvements in the implemented simulation. It is shown how the angle representation is changed to a two-parameter unitrary angle representation and realistic ball model from baseball data is introduced.

Chapter 4 presents the results of the implemented simulation. The results of the simulation with the extended ball model are presented. More results that consider the influence of noise and model are shown.

Chapter 5 discusses a proposal for carrying out the experiments to validate the MPC-POMDP model of ball catching. It is examined how the control variables for the experiment can be specified. It is also explained how to identify and validate system parameters of the model with the experiments.

Chapter 6 concludes the thesis and gives an outlook.

The Appendix contains some evaluations of the simulation for the proposed experiments in Chapter 5.

2 Baseline mathematical model of ball catching

The baseline mathematical model of ball catching from [1, 9] is introduced in this chapter. First the model of the catcher and the ball are described. Then optimal control with POMDPs and belief states will be introduced. Finally, the limitations of this model are stated.

2.1 Model of the catcher

The catcher is modeled as point mass with a gaze vector. The movement of the agent is restricted to th xy-plane. The gaze can be represented by two angles. These angles are called φ and ψ. The angle φ represents the yaw, ψ the pitch. In figure 2.1 the angles φ is visualised and in figure 2.2 the angle ψ is visualized.

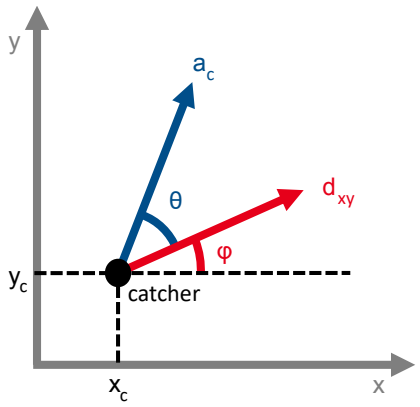

Figure 2.1.: Schematic view of the catcher from above. Vector d_{xy} represents the projection of the gaze on the $xy-plane$ and vector a_c represents the controllable forece applied to the catcher. Angles φ and θ are counted relative to the positive direction of the x-axis.

The catcher can be controlled by applying forces to him. The catcher himself can control the acceleration force F_c and its direction by θ. The catcher actually controls the accerelartion $a_c = F_c/m$ where m is the mass of the catcher. This is visualized in figure 2.1. Furthermore the angles φ and θ can be controlled by the angular speed ω_φ and ω_θ. The control vector of the catcher is

$$u = [F_c, \theta, \omega_\varphi, \omega_\psi]^T. \tag{2.1}$$

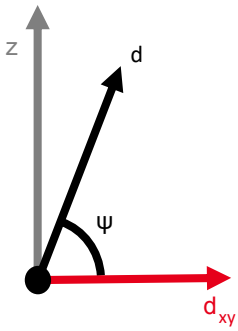

Figure 2.2.: Schematic view of catcher from the side. Representation ψ as angle between the gaze d and d_xy on the xy-plane. The angle ψ counted with respect to d_{xy} and is limited by $0 \leq \psi < \pi/2$.

The state vector of the catcher is given by

$$x_c = [x, y, \dot{x}, \dot{y}, \varphi, \psi]^T. \tag{2.2}$$

The equations of motion are given by

$$m\ddot{x} = F_c \cos(\varphi + \theta) - \lambda_m \dot{x}, \tag{2.3}$$
$$m\ddot{y} = F_c \sin(\varphi + \theta) - \lambda_m \dot{y}, \tag{2.4}$$
$$\Theta\dot{\varphi} = \Theta\omega_\varphi, \tag{2.5}$$
$$\Theta\dot{\psi} = \Theta\omega_\psi, \tag{2.6}$$

where Θ is the moment of inertia and λ is a friction coefficient for the catcher. The damping force is described by $\lambda_m v_c$ where $v_c = [\dot{x}, \dot{y}]^T$ is the velocity of the catcher. The terms $\Theta\omega_\varphi$ and $\Theta\omega_\psi$ are angular momenta. In the equations 2.5 and 2.6 moment of inertia Θ cancels out. As λ and the control F_c can be chosen arbitrarily, they can be divided by the mass of the catcher m. With $a_c = F_c/m$ and $\lambda = \lambda_m/m$ the mass on the left hand side of equation (2.3a and 2.3b) is crossed out. This results in the following simplified equations of motion

$$\ddot{x}_c = a_c \cos(\varphi + \theta) - \lambda\dot{x}, \tag{2.7}$$
$$\ddot{y}_c = a_c \sin(\varphi + \theta) - \lambda\dot{y}, \tag{2.8}$$
$$\dot{\varphi} = \omega_\varphi, \tag{2.9}$$
$$\dot{\psi} = \omega_\psi. \tag{2.10}$$

In order to roughly reflect reality, limitations must be specified for the individual parameters. When a parameter is bounded from both the top and bottom, it is called a box constraint.

$$-2\pi \leq \omega_\varphi \leq 2\pi \tag{2.11}$$
$$-2\pi \leq \omega_\psi \leq 2\pi \tag{2.12}$$

The box constraints 2.11 and 2.12 ensure that the catcher does not turn too fast. In order that the view of the catcher only remains within the permitted space will be ensured by

$$0 \leq \psi < \frac{\pi}{2}. \tag{2.13}$$

Because the ball is only above the xy-plane, ψ must not be negative. To avoid the singularity of ψ in $\pi/2$, ψ must be smaller than $\pi/2$. In simulation, the upper bound is set a little smaller than $\pi/2$ to faciliate optimization. The model reflects that while running forward the catcher can accelerate faster than running backwards. Because of this the bounds of the acceleration a_c cannot be constant. The limits for a_c are given by

$$0 \leq a_c \leq F_1 + F_2 \cos\theta \tag{2.14}$$

with F_1=7.5 m/s^2 (= 7.5 N/kg) and F_2=2.5 m/s^2. In literature, the two constants are called F_1 and F_2 [1]. The notation is kept in this thesis.

2.2 Simple Model of the ball

The simple ball model ignores all kinds of aerodynamic effects. Because of that there is only the gravitational force $F_g = -m_b g$ acting on the ball, with the mass m_b of the ball and g the standard gravitation $g = 9.81 m/s^2$. The gravitational force only acts in z-direction. Both other directions do not get any force, so the velocity in x- and y-direction stays constant over the entire flight. The equations of motion are defined by

$$m_b\ddot{x}_b = 0, \tag{2.15}$$
$$m_b\ddot{y}_b = 0, \tag{2.16}$$
$$m_b\ddot{z}_b = F_g = -m_b g. \tag{2.17}$$

When dividing by the mass of the ball, the mass is crossed out, to $\ddot{z}_b = -g$, which means that the mass of the ball is not needed in this ball model. In figure 2.3 the ball is illustrated schematically. It has a 3D vector of velocity and only one force acts on the ball. The trajectory for this model is a parabolic curve.

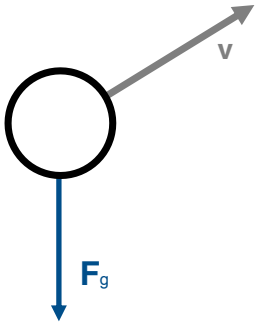

Figure 2.3.: Baseline ball model that only accounts for the gravitational force.

2.3 Optimal control model of ball catching

It follows the definitions of a Markov decision process (MDP) and a partially observable Markov decision process (POMDP). The transition and observation model, approximate gaussian belief dynamics, and the objective function are introduced.

2.3.1 Definitions of MDPs and POMDPs

A Markov decision process (MDP) is a mathematical model of decision making under uncertainty [17]. The state transition function of an MDP satisfies the Markov property. The Markov property means in words that the future state only depends on the current state and action and not on the past states.
Formally, an MDP is defined by a tuple $(S, A, T, R, \gamma, s_0)$ [17] where

- S is a set of discrete states which describe the environment,
- A ia a set of discrete actions that can be performed by the agent,
- T is a transistent function which gives the probability of a new state,
- $R(s,a)$ a reward function that gives a reward for taking an action a in a state s,
- $\gamma \in (0,1)$ is the discount factor and
- s_0 is the initial state.

A Partially observable markov decision process (POMDP) is an extension of MDP to cases when the system state is not fully observed [17]. A POMDP can be represented by a tuple of $(S, A, T, \gamma, R, \Omega, O, b_0)$, which adds to an MDP

- Ω is a set of observations,
- O is a observation model and
- b_0 is the initial belief.

2.3.2 Transition and observation model

Not all parts of the state vector can be observed. Only the catcher's and the ball's positions and catcher's gaze direction can be observed. The observation z is then described by

$$z = [x_b, y_b, z_b, x_c, y_c, \varphi, \psi]^T. \tag{2.18}$$

The function that maps the state to the observation is defined as $z_k = h(x_k)$ at discrete time step k. This model would allow the catcher to see the ball no matter in which direction he looks. In order to prevent this the noise is modelled state dependent. There is also an uncertainty of the system. Both uncertainties are described by:

$$x_k = g(\mu_{k-1}, u_{k-1}) + \epsilon \qquad \epsilon = N(0, Q) \tag{2.19}$$

$$z_k = h(x_k) + \delta_k \qquad \delta = N(0, R(x_k)) \tag{2.20}$$

where $N(\mu, \Sigma)$ is the normal distribution with mean and covariance.

2.3.3 Approximate gaussian belief dynamics

In order to integrate the uncertainties in the model used, the positions are represented by belief states. Gaussian beliefs are considered [18]. These gaussian beliefs can be represented by the mean and a covariance matrix $b_k = (\mu_k, \Sigma_k)$. The mean is associated with the most likely state of the system and the covariance matrix specifies the uncertainty about this state. To calculate the next belief state $(\mu_{k-1}, \Sigma_{k-1}, u_{k-1}, z_k) \rightarrow (\mu_k, \Sigma_k)$ the extended Kalman filter is used, which has the following update rule [19, 20]:

$$\bar{\mu}_k = g(\mu_{k-1}, u_{k-1}) \tag{2.21a}$$
$$\bar{\Sigma}_k = A_{k-1}\Sigma_{k-1}A_{k-1}^T + Q_k \tag{2.21b}$$
$$K_k = \bar{\Sigma}C_k^T(C_k\bar{\Sigma}_k C_k^T + R_k)^{-1} \tag{2.21c}$$
$$\mu_k = K_k(z_k - h(\bar{\mu}_k)) \tag{2.21d}$$
$$\Sigma_k = (I - K_k C_k)\bar{\Sigma}_k \tag{2.21e}$$

where $g(x,u)$ is the discretized function,

$A_{k-1} = \frac{\partial g(x,u)}{\partial x}\Big|_{\mu_{k-1}}$ is the linearisation of the discrete function at the current belief,

$C_{k-1} = \frac{\partial h(x)}{\partial x}\Big|_{\bar{\mu}_k}$ is the linearisation of the observation function,

I the identity matrix, Q_k the system noise and R_k the observation noise. The system noise is fixed $Q_k = Q$ while the observation noise is state-dependent $R_k = R(\bar{\mu}_k)$

2.3.4 Approximate uncertainty-aware objective function

The objective function used in this model can be divided into different parts. The catcher has to get at least near the ball until the end of the simulation with a preferably small uncertainty. The 'final' part of the objective function is

$$J_{\text{final}} = \omega_0||\mu_b - \mu_c||_2^2 + \omega_1 \text{tr}(\Sigma_N) \tag{2.22}$$

where μ_c is the catcher's final predicted position, μ_b the ball's final predicted function and Σ_N the covariance in the final step. ω_0 and ω_1 are weights of different parts of the cost function. For the catcher to be energy-efficient a part for control cost is added.

$$J_{\text{energy}} = \tau \sum_{k=0}^{N-1} u_k^T R u_k \tag{2.23}$$

with τ the time between two discrete time steps, which is constant, and $R = [10, 1, 1, 0.1]^T$ the weight for the different controls.
The third part of the objective function deals with the running uncertainty. This part is described by

$$J_{\text{running}} = \tau\omega_2 \sum_{k=0}^{N-1} \text{tr}(\Sigma_k). \tag{2.24}$$

where ω_2 is a weight, and Σ_k the uncertainty matrix at time step k. The complete objective function in this model of catching a ball is given by a sum of 2.22, 2.23 and 2.24

$$J = J_{\text{final}} + J_{\text{energy}} + J_{\text{running}} = \omega_0||\mu_b - \mu_c||_2^2 + \omega_1 \text{tr}(\Sigma_N) + \tau \sum_{k=0}^{N-1} u_k^T R u_k + \tau\omega_2 \sum_{k=0}^{N-1} \text{tr}(\Sigma_k) \tag{2.25}$$

Model predictive control (MPC) is a method to solve process control problems dealing with constraints. MPC formulation combines optimal control, stochastical control, control of processes with delay, multivariable control and future references [21]. There are two models for calculation in MPC: one for the simulation and one for the planner. The planner calculates his action till the prediction horizon in each iteration. And they propagate only the next step by taking the first action of the plan. In MPC usually a horizon for the actions is given called the control horizon. It specifies how many future actions will be planned. In this model the prediction horizon and the control horizon are both until the end of the flight of the ball. So the planner always have to plan the next $N - k - 1$ actions with N the number of discrete time steps and k the current discrete time step. The planner tries to minimize a given objective function. For this model it is equation 2.25.

2.4 Limitations to the baseline model

The baseline model uses a naive representation of angles φ and θ by limiting the range of values to either $[\pi, \pi]$ or $[0, 2\pi]$. However, such representation obviously leads to discontinuities at the boundaries. To fix this issue a two-parameter unitrary angles representation is introduced in section 3.1. The simple ball model has a strong simplification of the trajectory. Only the initial velocity influences the ball's trajectory. Therefore Section 3.2 introduces a model that is closer to reality obtained from baseball data. Another simplification can be found in the catcher model. When we say 'running forwards' it means that the direction of the gaze has the same direction as the acceleration a_c. 'Running backwards' means that the gaze is in the opposite direction as the acceleration a_c. In this model the rotation of the body is not modelled which means that it cannot be used to define 'running forwards' or 'running backwards'. In special cases that makes a difference. For example an agent goes backwards looking backwards, then the direction of a_c would be the same as the gaze d_{xy}. So the simulation considers this to be 'forward running'. In connection with ball catching it is sufficient to model only one angle. In addition, another angle would make the optimization much more difficult.
When $\theta = 0$ the catcher runs forwards and when $\theta = \pi$ it runs backwards, see equation 2.14. In between it is interpolated with the cosine function. So it assumed that running backwards is the slowest. In reality that is not the case. Moving sidewards is slower than running backwards in reality. The cosine function could be replaced, but that could be challenging for the optimizer.

3 Extended and improved simulation environment

Although the baseline model captures the main features of the ball catching problem, it suffers from several computational limitations. First, a naive angle representation which leads to discontinuities at the boundaries. Second, a simplified ball model that does not correspond to the measurable trajectory of baseball balls [22]. Therefore, we propose several modifications and improvements that are aimed at making the model fast, robust, and reliable. Although there already is an existing implementation from [1] in Python version 2.7, in the context of this thesis the simulation has been rewritten from scratch. For the new implementation Python version 3.7 was used. The covariance-free multiple shooting method [23] for trajectory optimization [24, 25] in the belief space was used. Derivatives of the cost function are computed using CasADI [26, 27] the latest version 3.4.5. Non-linear optimization is done by Ipopt [28]. L-BFGS and warm-starts are used [9]. The improvements that will be introduced in this chapter are a different representation of the angles, a more realistic ball model and investigations regarding the covariance matrix.

3.1 Removing singularities by using unitary angle representation

The baseline model used a naive representation of angles by limiting the range of values to either $[-\pi, \pi]$ or $[0, 2\pi]$. However, such representation obviously leads to discontinuities at the boundaries. Instead, we propose a singularity-free two-parameter unitary representation $(\cos\varphi, \sin\varphi)$. Although this representation requires two parameters instead of one and introduces an additional normalization constraint $\cos^2 + \sin^2 = 1$, it allows for smooth 360 rotations of the catcher which were not possible in the baseline model.

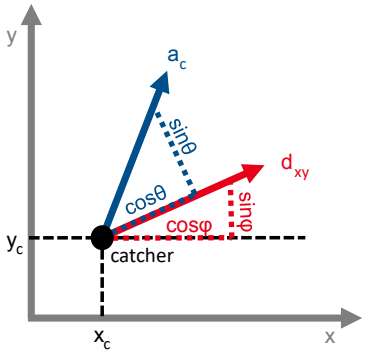

Figure 3.1.: Schematic look at catcher from above. Representation of d_{xy} and a_c and the variables $\cos\varphi$,$\sin\varphi$ $\cos\theta$ and $\sin\theta$ assosiated with them.

Because of the two-parameter representation the state vector 2.2 and the action vector 2.1 are updated to

$$u = (a_c, \cos\theta, \sin\theta, \omega_\varphi, \omega_\psi)^T, \tag{3.1}$$

$$x_c = (x, y, \dot{x}, \dot{y}, \cos\varphi, \sin\varphi, \psi)^T. \tag{3.2}$$

Thus part of the equations of the equations of motion must be adapted. The equation of motion for the angle φ of the catcher 2.9 has to be replaced by

$$\mathrm{d}\cos\varphi/\mathrm{d}t = -\sin\varphi\ \omega_\varphi, \tag{3.3}$$

$$\mathrm{d}\sin\varphi/\mathrm{d}t = \sin\varphi\ \omega_\varphi. \tag{3.4}$$

For angle θ additional constraints have to be defined because it is not in the state vector but in the control vector.

$$-1 \leq \cos\theta \leq 1 \tag{3.5}$$

$$-1 \leq \sin\theta \leq 1 \tag{3.6}$$

The relation between these two variables is shown by the following

$$(\cos\theta)^2 + (\sin\theta)^2 = 1. \tag{3.7}$$

For angle φ the same relation has to be true.

$$(\cos\varphi)^2 + (\sin\varphi)^2 = 1 \tag{3.8}$$

For the normalization constraint $(\cos\varphi)^2+(\sin\varphi)^2 = 1$ different integrators were compared. So the explicit euler did not remain close to the 1 even for small time steps τ. Instead it diverged. The integrator RK4, on the other hand, remains close to 1 in all experiments. Over the total time period of one simulation the deviation was smaller than 0.01:

$$0.99 \leq (\cos\varphi)^2 + (\sin\varphi)^2 \leq 1.01 \tag{3.9}$$

Because of this change in the representation of the angles balls can be caught in both directions. Different parts of the equations of motion were updated to avoid having to calculate the explicit angles φ and θ for each step. For

$$cos(\alpha+\beta) = \cos(\alpha)\cos(\beta) - \sin(\alpha)\sin(\beta) \tag{3.10}$$

$$sin(\alpha+\beta) = \sin(\alpha)\cos(\beta) + \cos(\alpha)\sin(\beta) \tag{3.11}$$

with arbitrary angles α and β, lead to an update of the equations of motion of the catcher 2.3.

$$\ddot{x} = a_c(\cos\varphi\cos\theta - \sin\varphi\sin\theta) - \lambda v_x \tag{3.12}$$

$$\ddot{y} = a_c(\cos\varphi\sin\theta + \sin\varphi\cos\theta) - \lambda v_y \tag{3.13}$$

The observation changes from equation 2.18 to

$$z = [x_b, y_b, z_b, x_c, y_c, \cos\varphi, \sin\varphi, \psi]^T \tag{3.14}$$

3.2 Using a realistic ball model obtained from baseball data

The extended ball model includes aerodynamic effects. There are two other important forces acting on the ball in addition to the gravitational force: the drag force and the magnus force [29, 30]. The magnus force is also called the lift force. The drag force is caused by air resistance of the ball. The direction of the force is the opposite direction of the velocity. It can be described by the following equation [31, 32]

$$F_D = \frac{1}{2} C_D \rho A ||\vec{v}||_2^2 \tag{3.15}$$

where C_D is the drag coefficient, $\rho = 1.23\text{kg/m}^3$ is the air density and A is the cross sectional area of the ball. For the drag coefficient the calculation from [22] is used. These coefficients were calculated based on flight balls in the american baseball league. [22]

$$C_D = c_d(1 + c_{d,spin} S^2) \tag{3.16}$$

with $c_d = 0.4$ and $c_{d,spin} = 0.2$. In addition to its velocity the ball has a rotation. This causes the magnus force on the ball. The magnus force F_m acts in the $\vec{\omega}_b \times \vec{v}_b$ direction. $\vec{\omega}_b$ is the rotation axis vector and $\vec{v}_b = (\dot{x_b}, \dot{y_b}, \dot{z_b})^T$ the velocity of the ball. We assume in our model that the ball has only backspin rotation. The other spins directions (sidespin and drall) are negligibly small in contrast to the backspin, so that they can be ignored. As a result of having backspin and no topspin the magnus force in our model produce a lift force. It can be described with [33]

$$F_D = \frac{1}{2} C_L \rho A ||\vec{v}||_2^2 \tag{3.17}$$

where C_L is the lift coefficient. Further on the coefficients C_L can be defined as [33]

$$C_L = 2C_D S \left[1 + \frac{v}{2C_D}\frac{dC_d}{dv}\right] \tag{3.18}$$

$$S = r\omega_b / v \tag{3.19}$$

where S is the spin coefficient, r the radius of the ball and ω_b the roations in rpm (rounds per minute). In this model, the mass of the ball can no longer be neglected. The mass of the ball is assumed to be 145.3 g (5 1/8 ounces). This is the mass of a standard baseball. Wind was not considered in our model. The equations of motions for the ball will result in

$$m\vec{a}_b = F_L \frac{\vec{\omega}_b \times \vec{v}_b}{||\vec{\omega}_b \times \vec{v}_b||_2^2} + F_D \frac{-\vec{v}_b}{||v_b||_2^2} + F_g \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \tag{3.20}$$

where $\vec{a}_b = (\ddot{x}_b, \ddot{y}_b, \ddot{z}_b)^T$ and $\vec{v}_b = (\dot{x}_b, \dot{y}_b, \dot{z}_b)^T$.

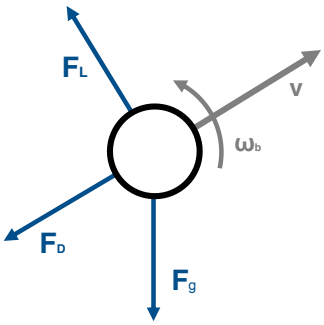

Figure 3.2.: Schematic representation of the extended ball model with the gravitational force, drag force and lift force acting to it. The ball has a backspin rotation and a velocity. The drag force acts in $-\vec{v}$ direction, the lift force in $\vec{\omega} \times \vec{v}$ direction and the gravitational force acts downwards.

3.3 Evaluation of different covariance matrix representations

A significant amount of time in the calculation of the belief space planner depends on the Extended kalman filter 2.21a-e. In step 2.21c the inverse of the matrix multiplication has to be calculated. Moreover during the calculation of the objective function only the trace of the matrix is needed. Computation with a full covariance matrix is expensive. It turns out that approximating it by a diagonal matrix does not lead to a significant degradation of performance in simulation; on the other hand, the computational efficiency raises noticeably.
There are situations where this simplification leads to wrong behaviour of the belief space planner. This could be the case when entries in the matrix become negative. Due to this simplification, the simulation runs much faster. However, in some cases the results are not as expected. For this reason, the variant with the full matrix is preferred. During the investigations of the required CPU times the various functions were also tested. There exist functions in CasADI. Its runtime was compared with that of Python. It turned out that the embedded CasADI functions are much faster. Therefore CasADI functions are still used in most places in the simulation.

4 Results of the simulation

The results presented here were obtained by simulation with the extended ball model and the full covariance matrix. All simulations calculate with a time delay of 0.2s. Due to reaction delay the catcher starts running, when the ball is already in the air.

4.0.1 Heuristics

When results of the simulation are presented in this chapter, the results of the ball catching heuristics are also evaluated. A number of heuristics have been proposed to explain how humans catch balls [6] [34]. We focus on three theories: Chapmans theory [2], the generalized optic acceleration cancellation (GOAC) theory [6], and the linear optical trajectory (LOT) theory [3].
The angle of elevation of gaze is α. Angle γ is the bearing angle with respect to the direction from the initial positions of the catcher to the ball. Chapman considered a simple kinematic problem where the ball follows a parabolic trajectory and the catcher's gaze is always directed towards the ball. Only the positions are relevant. Chapman demonstrated mathematically that a fielder standing in the plane of flight of a ball on a parabolic trajectory would intercept it if he ran at a constant speed that kept the acceleration of the tangent of alpha at zero [6]. This heuristic is explained by the following equation $\mathrm{d}\tan\alpha/\mathrm{d}t = \text{const}$. In other word, the tangent of alpha should have a almost linear curve. This heuristic is called the Optic acceleration cancellation (OAC) The other heuristics proposed by Chapman is the Constant bearing angle (CBA) heuristic. This requires the bearing angle $\gamma = \text{const}$. In both of Chapman's heuristics the observation direction is not needed, it is only positions based. It can be shown that CBA is also used by animals like dragonflies or fish [13, 14].
The two other heuristics include the observation direction. In [6] chapman's theory was extended by a field of view. The catcher's running velocity is coupled to the location of the ball in the visual field. The angle of rotation of gaze is called δ. The GOAC heuristic indicate that the angles delta and gamma should approximately be equal. If the ball is in another area of the field of view, the catcher must react accordingly. This tracking heuristic allows reactions to uncertain observations.
Both in chapman's theory and in GOAC theory the elevation angle α and the bearing angle γ are controlled independently. In [3] the linear optical trajectory (LOT) heuristics was proposed, which controls both angles jointly. The horizontal optical angle is β. LOT means mathematically that $\tan\alpha/\tan\beta = \text{const}$. LOT is also used by animals for object interception [15]

4.1 Simulation results with the extended ball model

It can be shown that the model still works with the extended ball model. A comparison of the trajectories of the simple and the extended ball model is shown. After that two scenarios where the catcher can catch the ball are shown. First, one where the ball catching heuristics hold. Second, a successful catch, where the heuristics do not hold.

4.1.1 Comparison of the two ball models in simulation

The different ball trajectories are shown in picture 4.1. Obviously the models have different trajectories. The differences are not so big with these initial velocities. The difference is only about two meters between the simple ball model and the extended ball model. Every marker on the trajectory stands for one time step $\tau = 0.1\,\mathrm{s}$. The extended model with drag and lift, shown by the red line, has a higher flight time than the other model. The green line shows the trajectory of the simple ball model and the blue line shows the trajectory beeing calculated with drag but not with the lift force. The flight time of the extended model with drag and lift is about 0.1 s longer. This can also be seen on the amount of markers. The belief state planner still works with the extended ball model. Also the MPC has no problems running the extended model what will be shown in the next subsections (Subsection 4.1.2 and Subsection 4.1.3).

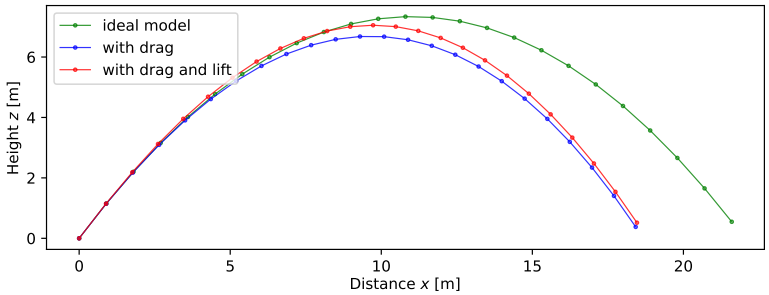

Figure 4.1.: Comparison of the trajectory of the ball with the different models.

4.1.2 Successful catch where heuristics hold

The ball starts at about $(0,0,0)^T$ with an initial velocity of $(8,0,10)^T$. The catcher starts at $(15,4)^T$, standing still and looking towards the ball. There is variance at the initial positions. In Figure 4.2 the curves of the ball and the catcher can be seen from above. It shows the result averaged over 10 MPC simulation runs. The mean values and the variance is plotted of the ball's and the catcher's position. For the catcher's gaze only the mean direction is plotted. The ball is the green line. Each marker stands for one time step $\tau = 0.1\,\text{s}$. The catcher's positions are blue, the catcher's gaze direction is shown as red arrows. It can be seen that the catcher's look al the time towards the ball. He runs in almost straight line towards the interception points. The average distance between the catcher and the ball in the end of the simulation was 0.1 meters. The maximal distance was 0.11 meters.
In Figure 4.3 the heuristics to this setting are plotted. In the upper left the OAC heuristic is plotted. In addition to the average simulation result and its variance, the linear fit was also plotted. The variance in the OAC line can hardly be seen. In the lower left the CBA heuristic is plotted. The curve is only roughly linear. After all, the deviation to the linear fit is small. Further back in the next section, Figure 4.5 shows how much the curve deviates if the heuristic is not maintained. In the upper right GOAC is plotted. It can be seen that the curves of the roation angle δ and the bearing angle γ corresponds. The deviations of both curves in the beggining at 0.2 seconds is related to the time delay of the catcher. In the lower right figure LOT shown. Therefore the mean of all simulation runs is plotted in green and its linear fit. Two explicit simulation results are plotted in blue and orange. Not all 10 simulation runs were plotted for clarity.

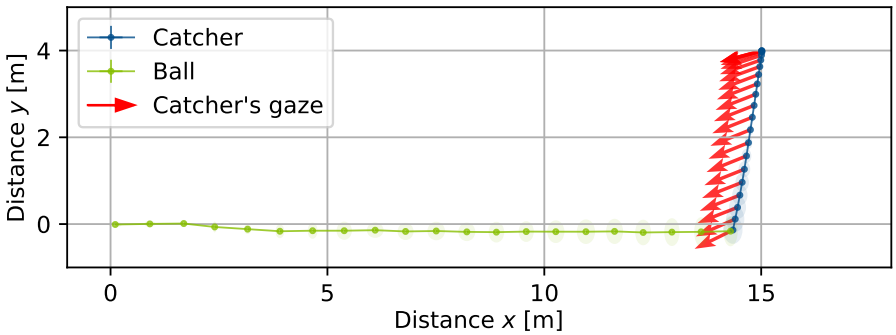

Figure 4.2.: Ball and catcher with gaze direction d_{xy} from above. Mean over 10 MPC simulation runs. Catchable side ball where heuristics hold. The corresponding plots for the heuristics are shown in Figure 4.3.

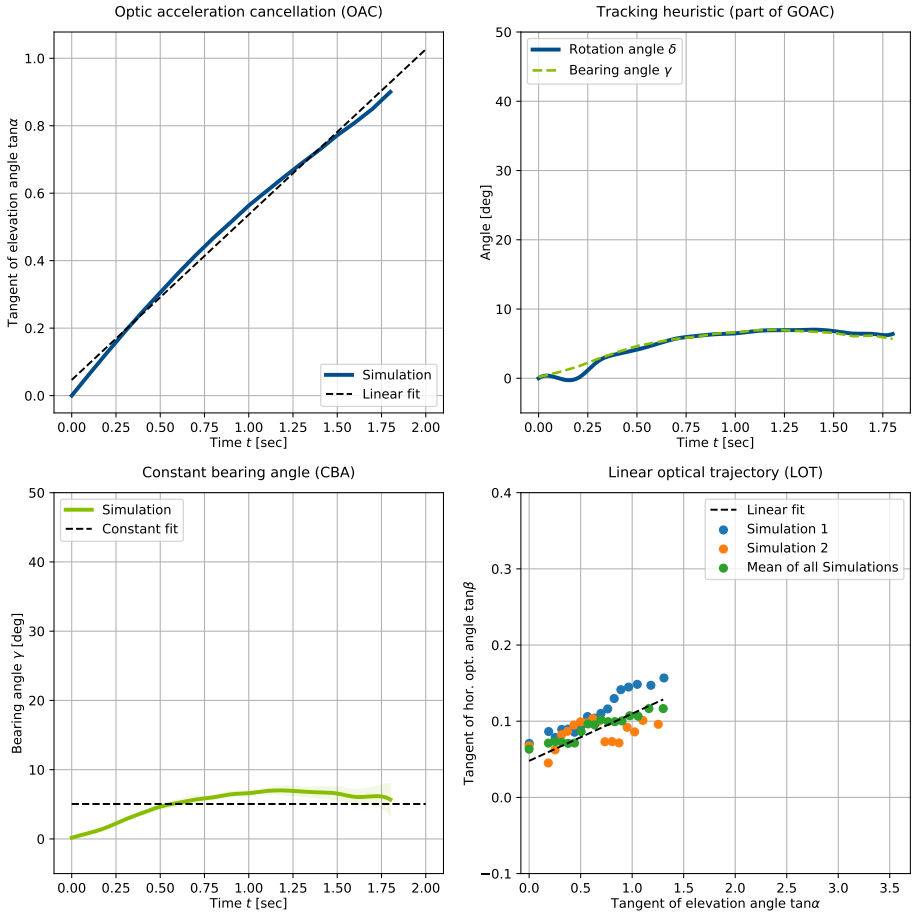

Figure 4.3.: Plot the heuristics OAC, CBA, GOAC and LOT corresponding to Figure 4.2. It shows the average heuristics over 10 MPC simulation runs.

4.1.3 Succesful catch, but heuristics do not hold

As already mentioned, this model can explain succesful catches which heuristics do not hold. In this setting the ball has the same initial conditions as in the previous section.
This time the catcher starts further away from the interception point of the ball. The distance is too long to run sideways, so the catcher turns to the interception point. Later he looks towards the ball again. In this scenario, the heuristics no longer hold, but the catcher can catch the ball nevertheless. In Figure 4.4 the situation is plotted from above. It shows the ball in green, the catcher in blue and the catcher's gaze in red. It is the average over 10 simulation runs. The behaviour that the catcher turns his gaze away from the ball can be seen easily in the figure. The average distance of the ball and the catcher in the end time step was: 0.22 m. The maximal distance was 0.36 m.
In Figure 4.5 the heuristics for this setting are plotted. Again in the upper left the OAC is plotted. It is no longer as linear as in the previous plot of the heuristics. In the lower left the CBA heuristic is shown. It is not nearly linear. In the upper right GOAC is shown. The rotation angle δ and the bearing angle γ do not correspond. Last but not least in the lower right LOT is plotted. Also here is in green the mean over all 10 simulation runs shown. Two explicit simulations are plotted in blue and orange. Not all 10 simulation runs were plotted for clarity, too.

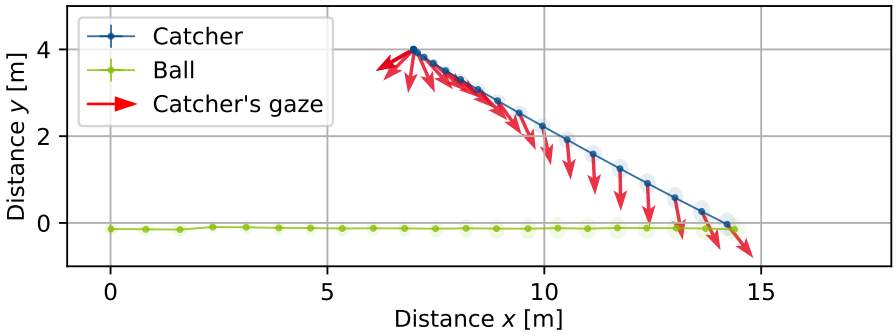

Figure 4.4.: Ball and Catcher with gaze direction d_{xy} from above. Mean over 10 MPC simulation runs. The interception point of the ball is too far for the catcher to run backwards. Because of this the catcher turns around to the predicted interception points, for fast acceleration. Later he looks back to the ball. The corresponding plots for the heuristics are shown in Figure 4.5.

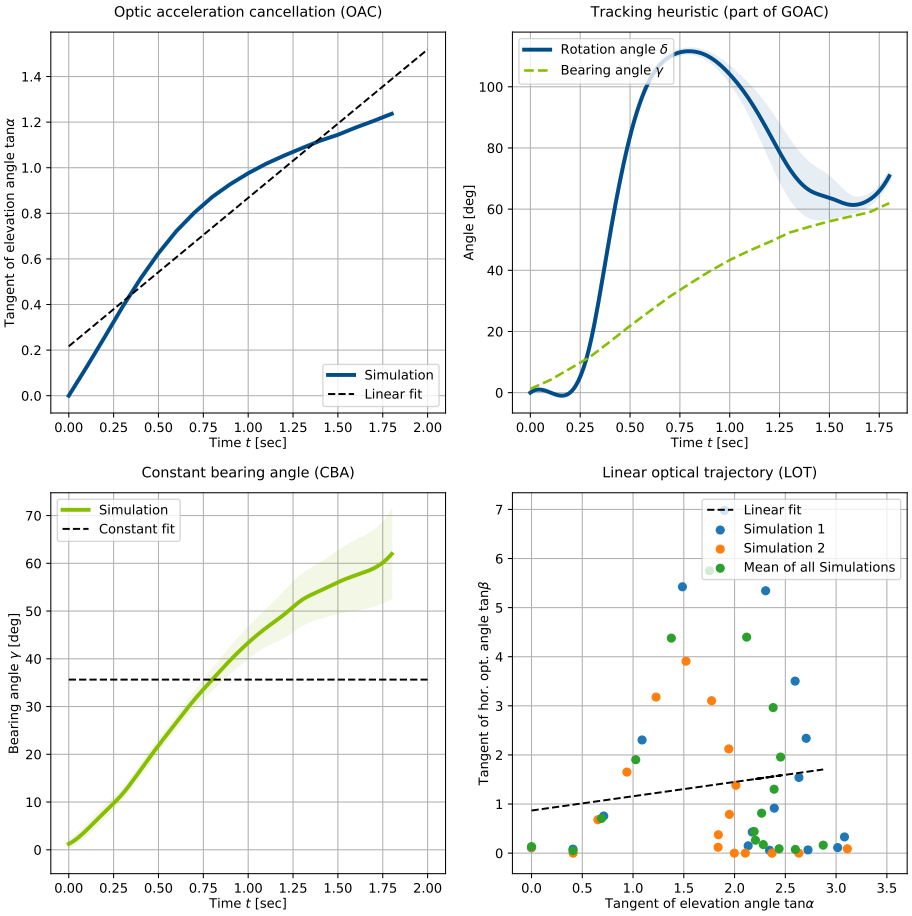

Figure 4.5.: Plot the heuristics OAC, CBA, GOAC and LOT corresponding to Figure 4.4. It shows the average heuristics over 10 MPC simulation runs. None of the heuristics holds although the catcher catches the ball.

4.2 Effects of noise

To investigate the effect of noise the noise of the initial state was increased. This makes it possible to investigate how stable this system is against small or large deviations in the start positions. Here, too, the result was averaged over 10 simulations. The initial noise was significantly increased and the heuristics hold. The catcher can still catch the ball reliably. The average distance of the catcher and the ball in the end time step was 0.13 m. The maximal distance for the 10 simulation runs was 0.29 m. In Figure 4.7 the results of the simulation runs are plotted. It can be seen that the noise is significantly higher than in Figure 4.2. In Figure 4.6 the heuristics are plotted. The left upper figure shows the OAC, which is almost linear from simulation. The lower left figure shows the CBA heuristic, which shows small deviations to the constant fit. It can be recognized that the higher initial noise influences the deviation in CBA. The deviation of the CBA is higher as there is not that high noise, see Figure 4.2. The upper right figure shows that the rotation angle δ and the bearing angle γ are about the same. The lower right shows the LOT. The green dots represent the mean over all 10 simulation runs. Two explicit simulation results are plotted in blue and orange. Not all 10 simulation runs were plotted for clarity.

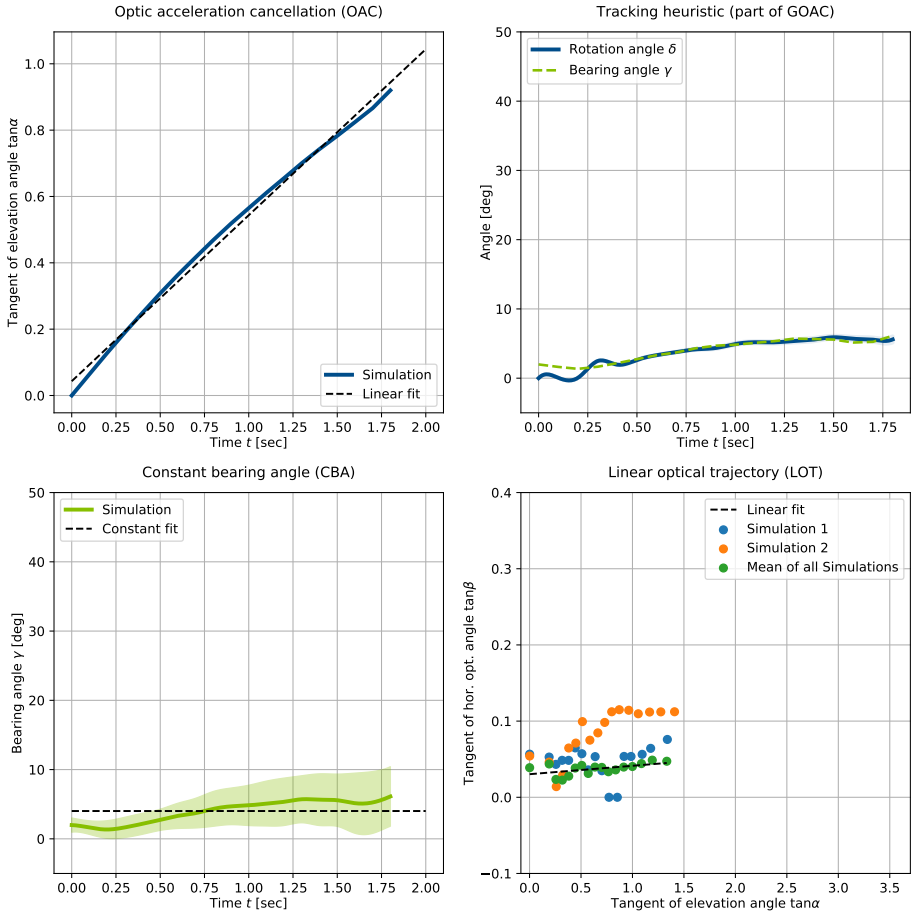

Figure 4.6.: Plot the heuristics OAC, CBA, GOAC and LOT corresponding to Figure 4.7. It shows the average heuristics over 10 MPC simulation runs.

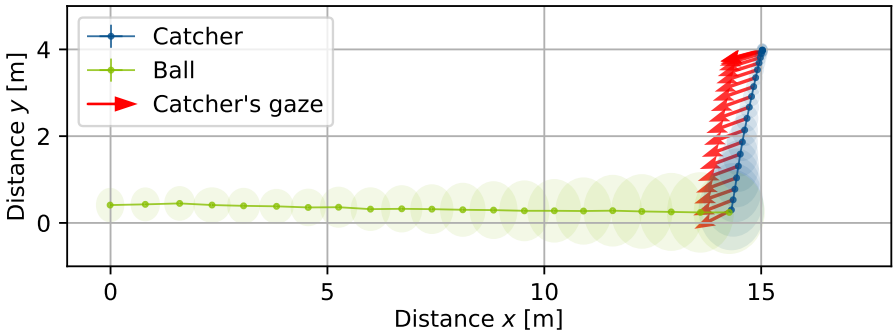

Figure 4.7.: Ball and Catcher with gaze direction d_{xy} from above. Mean over 10 MPC simulation runs. The initial noise was significantly increased. The corresponding plots for the heuristics are shown in Figure 4.6.

4.3 Model mismatch: planning with a 'wrong' model

In the previous scenarios it was assumed that the model used by the planner corresponds to the model used for the simulation. In [16] it was examined how different algorithms behave when the planner calculates with a ball model without drag while the simulation calculates with a ball model with drag. The algorithm used here in this thesis was also investigated. Here we investigate how the catcher behaves when the simulation calculates with the extended ball model (introduced in Section 3.2), but the planner plans his actions with the simple ball model (from Section 2.2). Figure 4.8 shows the result. The first planning calculation plans towards the point $(15, 0)^T$. The simulated ball with noise lands at approximately $(14, 0)^T$. Despite the wrong mathematical model, the catcher can still catch the ball, because the deviations of the two models are not too big, in this scenario. The average distance of the catcher and the ball in the end time step was 0.15 m, the maximal one was .17 m.
The heuristics for this are shown in Figure 4.9. The left upper figure shows the OAC, which is almost linear from simulation. The lower left figure shows the CBA heuristic, which shows small deviations to the constant fit. The upper right figure shows that the rotation angle δ and the bearing angle γ are about the same. The lower right shows the LOT. The green dots represent the mean over all 10 simulation runs. Two explicit simulation results are plotted in blue and orange. Not all 10 simulation runs were plotted for clarity. So all heuristics hold for this settings.

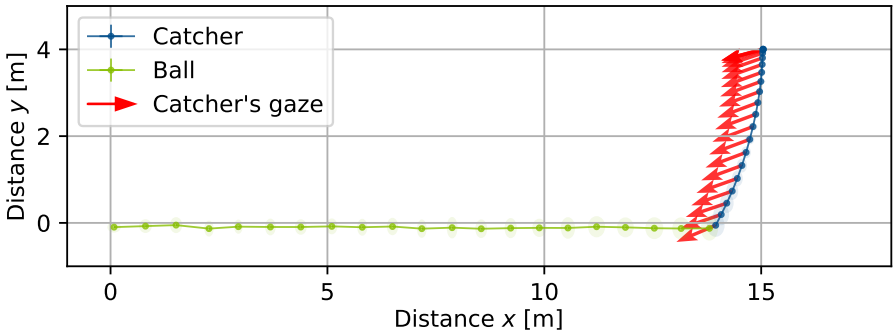

Figure 4.8.: Ball and Catcher with gaze direction d_{xy} from above. Mean over 10 MPC simulation runs. The simulation uses the extended ball model while for belief space planning the simple ball model is used. The first planning calculation plan towards the point $(15, 0)^T$. The simulated ball with noise lands at approximately $(14, 0)^T$. The heuristics of these simulation runs are shown in Figure 4.9.

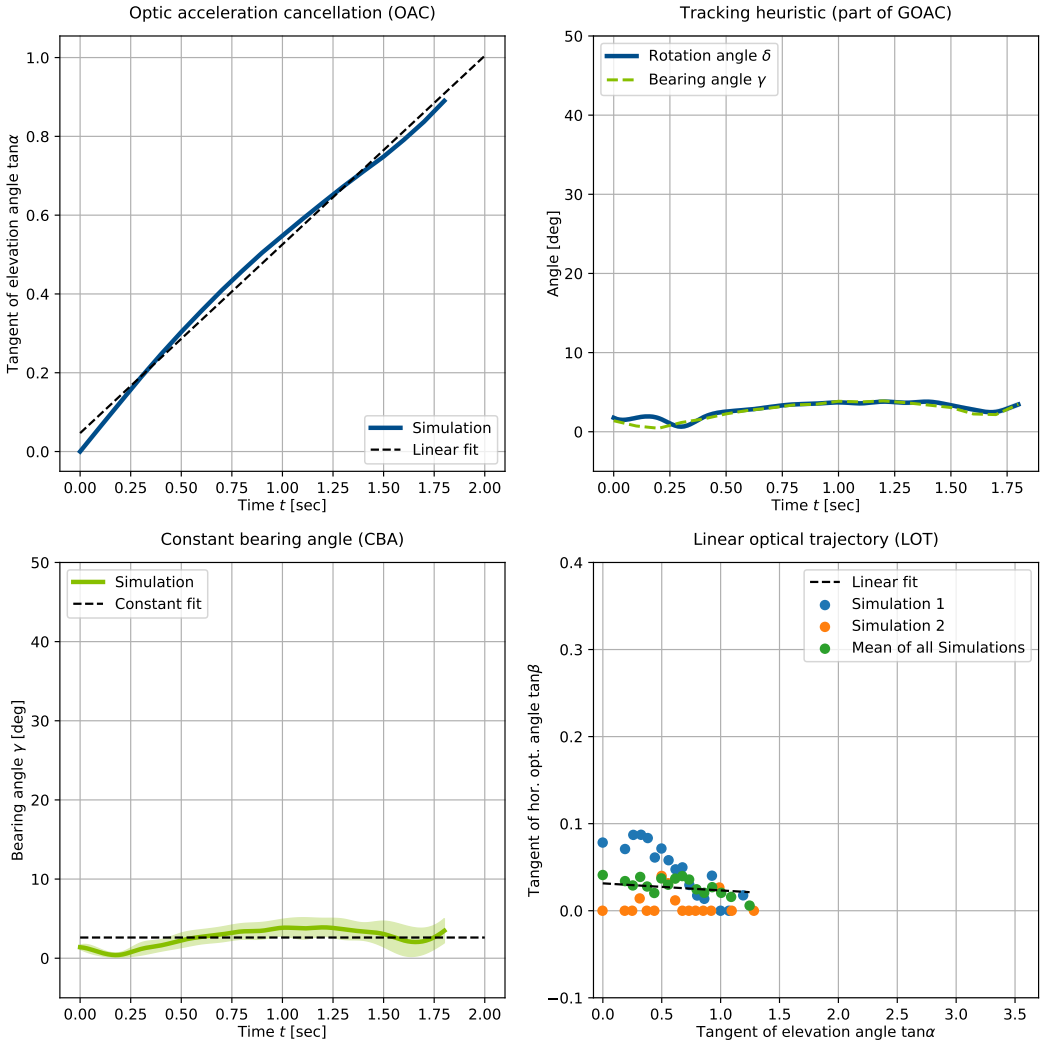

Figure 4.9.: Plot the heuristics OAC, CBA, GOAC, and LOT corresponding the plots of the catcher and the ball in Figure 4.8. It shows the average heuristics over 10 MPC simulation runs.

5 A proposal for experimental validation of the ball catching model

The aim of the experiments is to show if the model will work for human ball catching or not. The main goal is to show whether the behaviour shows in figur 4.4 can be shown in in human ball catching. In this chapter some general information will be given. Then different control variables and their values are discussed. Finally, system identification for the ball and catcher model will be covered.

5.1 General description of the experimental setting

Some general information about the participants, the field and materials as well as the procedure are showns.

5.1.1 Participants

All participants should participate voluntarily. The only requirements for them would be not to be unathletic. It would be good if they had some experience in ball games, like how to catch a ball. All participants are free to choose how to catch the ball, especially which hand they use to catch the ball. The participants will not be introduced in the theories of the experiments. All participants have to do some warming up exercises prior the experiments, because we want them to sprint to get the ball and we do not want anyone to get hurt. The participants should not have an increased reaction time due to additional factors, such as alcohol or drugs.

5.1.2 Materials

For tracking positions of the catcher, the ball and the catcher's gaze, it is planned to use visual tracking by cameras. The cameras are placed around the field so that they can detect the positions in a certain area. All positions have be in the range of this area during the attempt. The plan is to have enough cameras to cover an area of 15 meter length (x-coordinate), 4 meter width (y-coordinate) and 5 meters height (z-coordinate). The length describes the x-axis, the width the y-axis and the height the z-axis. Maybe it is possible to enlargen the area with more cameras. Then the maximal range would be 20 meters length, 5 meters width and 7 height. To start with the smaller numbers, the constraints look like:

$$0\,\mathrm{m} \leq x \leq 15\,\mathrm{m}$$

$$0\,\mathrm{m} \leq y \leq 4\,\mathrm{m}$$

$$0\,\mathrm{m} \leq z \leq 5\,\mathrm{m}$$

All positions should be in this range at all times. This has to be considered when choosing start positions of the ball and catcher and the initial velocity of the ball. Another condition would be that the catcher has to looks towards the ball-machine when starting the experiment.

5.1.3 Procedure

The catcher starts at a given initial position. The ball machine is set to a certain initial velocity at a given position. Another person is needed to put the ball into the ball-machine to start the experiment. When the ball-machine throws the ball into the given area, the catcher should run to catch the ball. It is noted if the catch was successful. As successful catch is defined when the catcher at least touches the ball. As we do not model the end-sequence of catching it is not that important if it is a safe catch. This information will not be told to the participants, so that they still try to catch the ball safely. After it was noted if the catch was succesful the next given initial positions are set. Several runs can be performed by the same participant one after the other.

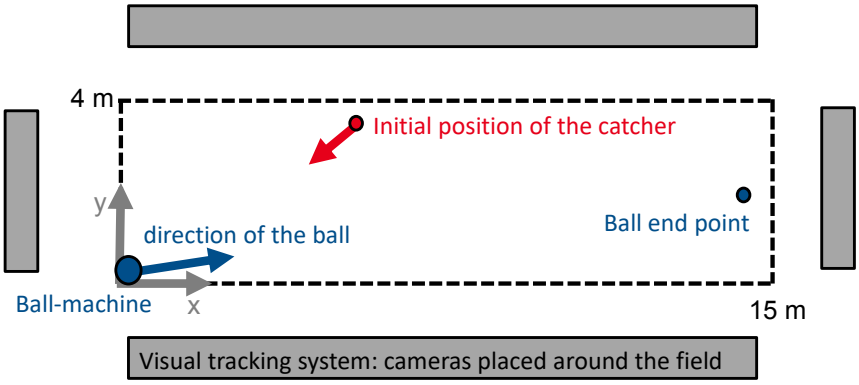

Figure 5.1.: Schematic illustration of the experiment field from above.

5.2 Specification of control variables

There are different control variables that can be changed during the experiments. An overview of the control variables:

- Start position of the ball machine,
- Initial velocity of the ball which results in the ball trajectory,
- Start position of the catcher,
- Reaction delay of the catcher,
- Observation noise of the catcher.

The different catching strategies depending on reaction delay, task duration, the system noise, and the observation noise are shown in [9] in figure 6. The system noise cannot be manipulated and the task duration is given by the initial velocity of the ball. The observation noise and the reaction delay can be manipulated by a small amount only.

5.2.1 Ball initial state

The ball-throwing-machine will be placed at the smaller line of the field. Because of this the initial position of the ball will be limited to $\vec{x}_{b0} = (0, y, 0)^T$. The vertical starting speed is selected so that the maximum permissible height of 5 metres is not exceeded, but slightly below. Because of that the ball should stay in the air nearly the same amount of time. The inital horizontal velocity can be calculated from the ball interception point. So it is sufficient to give the interception point of the ball for the experiments.

5.2.2 Start position of the catcher

For the behaviour of the catcher the flight duration, the interception point, and the orientation of the catcher to the interception point are important. While the first two are set by the ball initial state, the third is influenced by the initial position of the catcher. The orientation of the catcher to the interception point can be roughly divided into four areas:

- Near balls,
- Long balls,
- Side balls,
- Short ball.

In this section the different start positions for the catcher are described and an explanation is given why they are helpful.

Near ball, shoot at catcher's initial position

The simplest setting is the following: shooting the ball at or at least near the catcher's initial position. With this experiment we can test the accuracy of the cameras and the noise and the reliability of the image recognition for the case that the catcher stays near to the initial position. This has only to be done at the beginning of the experiments. This is of course an experiment that we do not need to repeat often. So not all participants have to catch this ball. We might repeat this setting on a new experiment day (if the experiments take more than one day) because camera positions or lighting conditions might change what can lead to a change of the noise of the image recognition. These catches can be used for calibrations of the cameras and image recognition.

Short ball

When the catcher has to reduce the distance to the origin of the ball, this is called a short ball. The catcher has to run forward. Two different scenarios are possible here:

- a ball that can be caught
- a ball that is too far away to be caught (but still not that far, that the catcher does not try to catch it)

With this experiment the forward catching can be analysed and compared with the simulation. In the first case it is expected that the heuristics hold. We expect that the catcher tries to reach the ball even in the second case. With experiments of the second case we can measure the maximal acceleration and speed of the catcher. From these results we may have to adjust the parameters of the simulation. Especially the parameters F_1 and F_2 are effected (a more detailed description can be found in Section 5.3.1). It might be the case that the catcher thinks he cannot reach this ball (which is correct) and does not even start to run. In this case the run could be repeated with less distance between impact point of the ball and the catcher's initial position. This means the initial speed of the ball has to be increased.

Side ball

A fly ball for which the catcher only has to mainly run left or right is called a side ball. In this case the catcher does not reduce his distance to the origin of the ball significantly, nor increase it. With the results of our simulation we can classify side balls in three categories:

- catchable ball, where the catcher always looks towards the ball,
- catchable ball, where the catcher turns his gaze away from the ball,
- uncatchable ball.

For catchable side balls, where the catcher turns his gaze towards the predicted interception points, will be a challenge of measuring the turning of the catcher's gaze because the head movement will be small. In addition to this there are constraints of the field because the area is only four meters wide. Because we have the constraints of only four or five meters of the field we could not test catchable side balls, where the catcher turn his gaze, and uncatchable side balls. Catchable side ball, where the catcher always looks towards the ball, can be run in the experiments. Catchable side balls with no-gaze turning have a similar expectation as catchable short balls. They should fit the heuristics. In addition to the 'catchable short ball' we could measure with the 'easy catchable side ball' the difference between the right and the left side of the catcher.

Long ball

A ball, for which the catcher has to increase the distance to the origin of the ball, is called a 'long ball'. There are two strategies how the catcher can get a long ball. The first one is to run backwards. This has a limited range because of flight duration. If the catcher cannot get to the impact point in time, he might use another strategy: turn towards the predicted impact point, accelerate fast and then turn the head back to the ball. This strategy can work because running forwards is faster than running backwards. If the catcher still cannot get the ball even with turning, the ball is unreachable. This will result in three catogories of long balls:

- catchable long ball, catcher can look towards the ball all time
- catchable long ball, catcher has to turn away his gaze from the ball
- unreachable long ball

For a long ball without gaze-turning we expect that heuristics will hold, similar to catchable side balls with no gaze-turning and catchable short balls. A long ball with gaze-turning is too far away to be caught by running backwards. In this case the catcher has to turn around, run forward and then look back towards the ball. This behaviour is shown in figure 4.4. This is the main behaviour we want to test with this experiment. These balls are the most important ones for the experiment. Because of that we have to test a higher amount of long balls with gaze-turning than the other ones. Although these balls are too far away for running backwards, some participants nevertheless will try to catch it by running backwards and fail catching it. Because of this we might not see the expected behaviour in some trials. The angle between the catcher's straight line to the impact points and the ball's xy-projection line can be varied.
Unreachable long balls cannot be caught even with the strategy of turning around and running forwards.

Choosing the start positions for the experiments

It could be challenging to set the ball machine to the exact same position again. So we would have high variance on the ball's initial velocity. So it is better to change the initial position position of the catcher than the ball-machine settings to get a higher repeatability. As a result as many trials as possible should have the same initial settings for the ball. It is proposed to repeat each initial scenario at least 5 times per participant. So it would be at least 30 catches per participant (plus those for the system identification and the reaction delay). In 5.1 the proposed initial positions for the catcher are shown, when the interception point is $(14, 0)$. It is assumed that the ball starts at $(0, 0)$. Each position is classified as a ball that was described in this section.

Table 5.1.: Proposed start positions of the catcher when ball starts at $(0, 0)$ and the interception point is $(14, 0)$. Classification of the ball and whether the catcher has to turn his gaze towards the interception point are shown. Simulation results of these initial states are shown in the appendix. The ID is for referring to the inital states and is consecutively.

ID	Initial position of the catcher	Classified ball as	Catcher has to turn his gaze
1	(7|4)	hardly catchable long ball	gaze-turning
2	(7,1)	long ball	gaze-turning
3	(9,4)	long ball	gaze-turning
4	(9,1)	long ball	no gaze-turning
5	(11,4)	side long ball	no gaze-turning
6	(15,4)	side ball	no gaze-turning

5.2.3 Observation noise

We want to modify the observation noise of the catcher. One possibility is to reduce the contrast between the ball and the background, so that the ball is more difficult to observe. Therefore the weather of the day and the field background are important. It is likely that not all of the experiments can be done in one day. It would be nice if the sky on these two or three days would be different (blue, starting sunset, partial cloudy, cloudy but not raining). So if we have a cloudy experiment day and we take a grey ball it would be more difficult to see the ball for the catcher than if we take a greenish ball. For varying the background we could also change the start direction to the opposite narrow field side.
During the experiments it is expected that each participant gets more routine in catching and evaluating the situation, doing better predictions and decreasing their internal model system uncertainty. For this the training with 'easy balls' (catchable short, easy side ball and no-gaze-turning long ball) are helpful. Another approach of varying the observation noise and to confuse the internal model is to take different type of balls such as a tennis ball. They are a bit smaller and made of a different material and therefore fly a bit different. In case that our ball-machine only works with one type of balls, this may not be feasible. In summary, the proposed scenarios for the observation noise are:

- Doing the experiments with normal balls and
- Color the ball to modify the observation noise.

5.2.4 Reaction delay

To modify the reaction delay over one whole trial is a challenging task. It could be something to affect the catcher's vestibular system. To influence this additional equipment would be needed. There are many more possibilities in influencing the reaction time of humans [35] (e.g. degree of arousal, alcohol, coffein,...). This influence is a more theoretical thought and not really practicable. Instead of trying to have impact on the reaction delay over the whole trial we could try to influence the initial reaction delay. One way to slightly increase the reaction delay the delay is to let the catcher starts with eyes closed. The one who starts the ball throwing process with the ball machine, will make a sound when starting. It can be a word like 'go', 'start', 'hey' or something else. This sound has to be loud enough so that the participant can hear it. When the participant hears the sound he should open his eyes immediately and try to catch the ball. This will result in a slightly increased initial reaction delay. To get a longer reaction delay the catcher could start with his head turned away from the ball. By this we get an initial reaction delay of the time needed to turn the head. To get an even longer initial reaction delay the catcher could start with his body and his gaze turned away from the ball. As soon as he hears the sound he has to turn his body and try to catch the ball. In this scenario it is expected that the reaction delay will be too high. In summary, the proposed scenarios to influence the initial reaction delay are those, when the catcher starts:

- Turned to the ball with his gaze also towards the ball,
- Turned to the ball but with his gaze turned away of the ball,
- Turned away from the ball and also his gaze turned away from the ball.

It is proposed to do the special scenarios with more reaction delay with the initial states of setting number 3 from Table 5.1.

5.3 System identification for the model

The model and its parameters can be improved by the experiments.

5.3.1 Validating the model of the catcher

With the experiments the parameters of the model can be checked. For the catcher, this involves the parameters F_1, F_2 and λ. The ration between $(F_1 + F_2)/\lambda$ is influenced directly the maximal possible velocity:

$$v_{\text{max,forwards}} = \frac{F_1 + F_2}{\lambda}, \tag{5.1}$$

and the acceleration when start running forwards

$$a_{\text{forwards}} = F_1 + F_2 - \lambda v. \tag{5.2}$$

with v_{max} the maximal possible velocity of the catcher. This can be derived from 2.7, 2.8 and 2.14, when the second derivatives of x and y are zero. This is the case at maximum speed in this model with damping.
The sum $F_1 + F_2$ indicates how fast the catcher can accelerate by running forwards. The difference $F_1 - F_2$ indicates how fast the catcher can accelerate backwards. More important than testing the difference $F_1 - F_2$ is testing the ratio $(F_1 - F_2)/\lambda$. It indicates the maximal speed when running backwards

$$v_{\text{max,backwards}} = \frac{F_1 - F_2}{\lambda}, \tag{5.3}$$

and the acceleration when start running backwards

$$a_{\text{backwards}} = F_1 - F_2 - \lambda v. \tag{5.4}$$

The maximal velocity and acceleration forwards and backwards can be measured in the experiments. For this measurement two additional experiment settings are proposed, shown in Table 5.2. For the setting 8 participants are told to only run backwards. They must not turn the gaze, because we want to measure the velocity and acceleration of backwards running when trying to catch a ball.

Table 5.2.: Proposed start positions of the catcher classification are shown for the system identification. Simulation results of these initial states are shown in the appendix. The ID is for referring to the inital states and is consecutively from Table 5.1.

ID	Initial position of the catcher	Interception point	Classification and specifics
7	(15, 0)	(≤7, 0)	hardly catchable short ball
8	(≤7, 0)	(14, 0)	hardly catchable long ball - only backwards running allowed

5.3.2 Validating the model of the ball

For the ball model the drag and lift coefficients probably have to be adjusted to fit the real situation of our experiments. That is because we use different balls in simulation and probably in the experiments. In the model baseballs are used to calculate the ball's trajectory. In the experiments probably softer balls are used. That could influence the trajectory. Moreover we will use a ball machine. The model uses pitched and batted balls. The parameter C_D and C_L in the equations 3.16 and 3.18 should be adapted. Maybe it is sufficient to adjust only the rotation speed ω, the mass m_b and the radius r of the ball.
To identify the parameters of the ball model some balls with different starting velocities should be shot. There should also be some fly balls with a launch angle of more than 70°, if the ball machine allows us such angles. It can be used to try to achieve the behaviour from [29] Figure 3.6 on page 48 where the ball does a loop when the launch angle is more than 80°. These balls for the parameter identification do not need to be catched.

5.3.3 Adapt the coefficient of the cost function

There are also the coefficients $\omega_0, \omega_1, \omega_2,$ and R of the cost function that could be modify. They cannot be calculated analytically, so they have to be tuned manually or by inverse Reinforment learning.

6 Conclusion and outlook

6.1 Conclusion

In this thesis, a proposal for an experimental validation of the described MPC-POMDP model of ball catching was presented. It was discussed which initial states are to be selected, in particular the chatcher's position and the initial velocity of the ball. The initial position influences the catcher's behaviour. So this model of ball catching can not only explain successfully caught balls, which can be explained by heuristics, but also those balls when the catcher has to turn his gaze away from it. In the experiments focus will be on this situation. We want to see in the experiments the special behaviour of this model. Further it was discussed how the observation noise and the reaction delay could be influenced. Additionally our model can be improved with the experiments. In particular, the acceleration and damping coefficients of the catcher and the drag and lift coefficients of the ball are of interest. Unfortunately, the experiments themselves could not be carried out during this thesis because of necessary equipment not having arrived in time.

In the context of this thesis a simulation of the MPC-POMDP model of ball catching was rewritten from scratch in the latest python version 3.7 using also the latest version of CasADI 3.4.5. Important changes and improvements have been implemented, as described in chapter 3. The angle representation has been changed to a two-parameter unitary representation. This causes more variables to be calculated, but has the advantage that optimizations are more stable because there are less local minima and maxima. Another important improvement in the simulation is that a more realistic model of the ball has been implemented. In contrast to the simple model, this one does not only use the gravitational force that acts on the ball but also the drag force and the lift force act on the ball. The drag force is caused by the air resistance and the lift force by the backspin rotation of the ball.
The effect of the extended ball model on the ball catching process was investigated. The effect of the noise was also tested. Additionally the effect of the MPC algorithm's planner having a wrong model was investigated. Specifically, the behavior was checked for situations when the planner uses the ideal ball model while the extended model is actually used for calculation.

6.2 Outlook

The extended simulation model and the proposed experimental validation setting constitute a foundation on which thorough evaluation of the proposed MPC-POMDP model is to be carried out. It is of interest to see whether the same model can explain behavior of animals too, e.g., dragonflies (known to follow CBA heuristic) and fish.
The computational approach used in this thesis is quite universal and one can in theory tackle any continuous POMDP with it. However, practically speaking, planning horizons beyond 15 time steps are completely infeasible in real time. Thus, finding more efficient computation procedures for POMDPs is of fundamental importance. One possibility is to reduce the number of time steps for belief space planning. It has been shown that despite small differences in the planner, the catcher can still catch the ball. So it could be tried to let the planner run with another number of time steps. The discretization time step τ between the time steps is then increased.
It would also be interesting to let a robot catch balls with this method, on a smaller scale. Additionally this model is not limited to ball interception. So it could also be interesting to hit balls by specifying a final velocity which could be realized by an additional part in the cost function.

Bibliography

[1] B. Belousov, “Modeling and learning of optimal strategies a case study on catching balls”, Master thesis. 2016.

[2] S. Chapman, “Catching a baseball”, *American Journal of Physics*, 1968.

[3] M. K. McBeath, D. M. Shaffer, and M. K. Kaiser, “How baseball outfielders determine where to run to catch fly balls”, *Science.*, 1995.

[4] P. W. Fink, P. S. Foo, and W. H. Warren, “Catching fly balls in virtual reality: A critical test of the outfielder problem”, 2009.

[5] A. Chohan, P. M. V. Kampen, M. H. G. Verheul, and G. J. Savelsbergh, “Children’s use of the bearing angle in interceptive actions”, 2008.

[6] P. McLeod, N. Reed, and Z. Dienes, “The generalized optic acceleration cancellation theory of catching”, 2006.

[7] S. Höfer, “On decomposability in robot reinforcement learning”, PhD thesis. 2017.

[8] A. Mattmann, “Modeling how to catch flying objects: Optimality vs. heuristics”, Bachelor thesis. 2014.

[9] B. Belousov, G. Neumann, C. A. Rothkopf, and J. Peters, “Catching heuristics are optimal control policies”, 2016.

[10] F. T. J. M. Zaal and C. F. Michaels, “The information for catching fly balls: Judging and intercepting virtual balls in a cave”, *Journal of Experimental Psychology*, 2003.

[11] L. I. N. Mazyn, M. Lenoir, G. Montagne, C. Delaey, and G. J. P. Savelsbergh, “Stereo vision enhances the learning of a catching skill”, 2006.

[12] D. A. Kistemaker, H. Faber, and P. J. Beek, “Catching fly balls: A simulation study of the chapman strategy”, *Human Movement Science.*, 2009.

[13] G. J. Diaz, F. Phillips, and B. R. Fajen, “Intercepting moving targets: a little foresight helps a lot”, Research Article. Springer, 2009.

[14] D. M. Shaffer, R. S. Marken, I. Dolgov, and A. B. Maynor, “Chasin choppers: using unpredictable trajectories to test theories of object interception”, Springer, 2009.

[15] D. M. Shaffer, S. M. Krauchunas, M. Eddy, and M. K. McBeath, “How dogs navigate to catch frisbees”, *American Psychological Society*, Research article, 2004.

[16] S. Höfer, J. Raisch, M. Toussaint, and O. Brock, “No free lunch in ball catching: A comparison of cartesian and angular representations for control”, 2018.

[17] H. Chinaei and B. Chaib-draa, “Building dialogue pomdps from expert dialogues”, Springer, 2016.

[18] J. van den Berg, S. Patil, and R. Alterovitz, “Motion planning under uncertainty using iterative local optimization in belief space.”, *The International Journal of Robotics Research*, 2012.

[19] R. P. Jr., R. Tedrake, L. Kaelbling, and T. Lozano-Perez, “Belief space planning assuming maximum likelihood observations”, 2010.

[20] S. Thrun, W. Burgard, and D. Fox, “Probabilistic robotics”, 2005.

[21] E. F. Camacho and C. Bordons, “Model predictive control”, Springer, 2007.

[22] D. Kagan and A. M. Nathan, “Statcast and the baseball trajectory calculator”, *American Assosiation of Physics Teachers*, 2017.

[23] M. L. S. Patil, G. Kahn and J. Schulman, “Scaling up gaussian belief space planning through covariance-free trajectory optimization and automatic differentiation”, *Algorithmic Foundations of Robotics XI*, 2015.

[24] M. Diehl, H. G. Bock, H. Diedam, and P.-B. Wieber, “Fast direct multiple shooting algorithms for optimal robot control.”, In: Fast Motions in Biomechanics and Robotics, 2005.

[25] J. T. Betts, “Survey of numerical methods for trajectory optimization”, *Journal of Guidance, Control, and Dynamics*, 1998.

[26] J. Andersson, “A general-purpose software framework for dynamic optimization”, 2013.

[27] J. Andersson, J. Akesson, and M. Diehl, “Casadi: A symbolic package for automatic differentiation and optimal control”, Springer, 2012.

[28] A. Wachter and L. T. Biegler, “On the implementation of a primal-dual interior point filter line search algorithm for large-scale nonlinear programming”, 2006.

[29] R. Cross, “Physics of baseball and softball”, Springer, 2011.

[30] R. K. Adair, “The physics of baseball”, 1995.

[31] D. Kagan and A. M. Nathan, “Simplified model for the drag coefficient of a pitched baseball”, *American Assosiation of Physics Teachers*, 2014.

[32] NASA, “Drag on a baseball”, 2015, https://www.grc.nasa.gov/WWW/k-12/airplane/balldrag.html.

[33] A. M. Nathan, “The effect of spin on a flight of a baseball”, *American Assosiation of Physics Teachers*, 2008.

[34] M. Zago, J. McIntyre, P. Senot, and F. Lacquaniti, “Visuo-motor coordination and internal models for object interception”, 2009.

[35] R. J. Kosinski, “A literature review on reaction time”, 2013.

A Some appendix results

A.1 Simulation results for experimental setups

The initial setting can found in table 5.1. All results are the mean over three simulation runs. The standard parameters are used for all kind of noise. Everey second picture shows the corsponding heuristics plots.

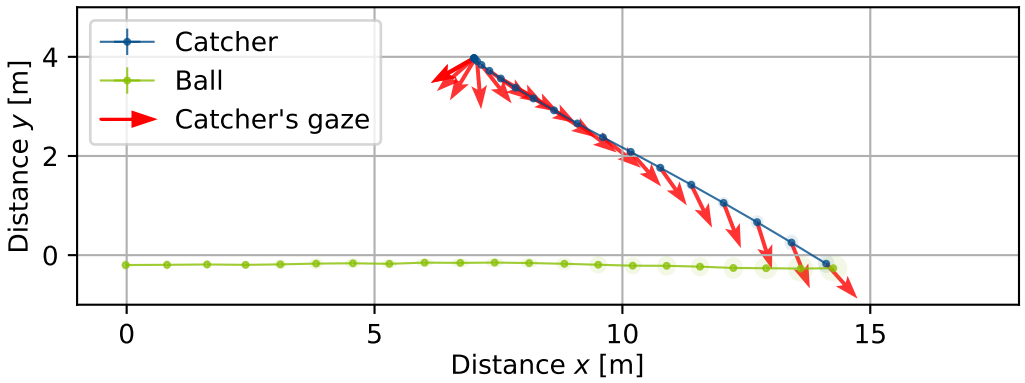

Figure A.1.: Result of the simulation for the start position (7,4) for the catcher averaged over three MPC-simulation runs. The heuristics corresponding to this result is shown in Figure A.2

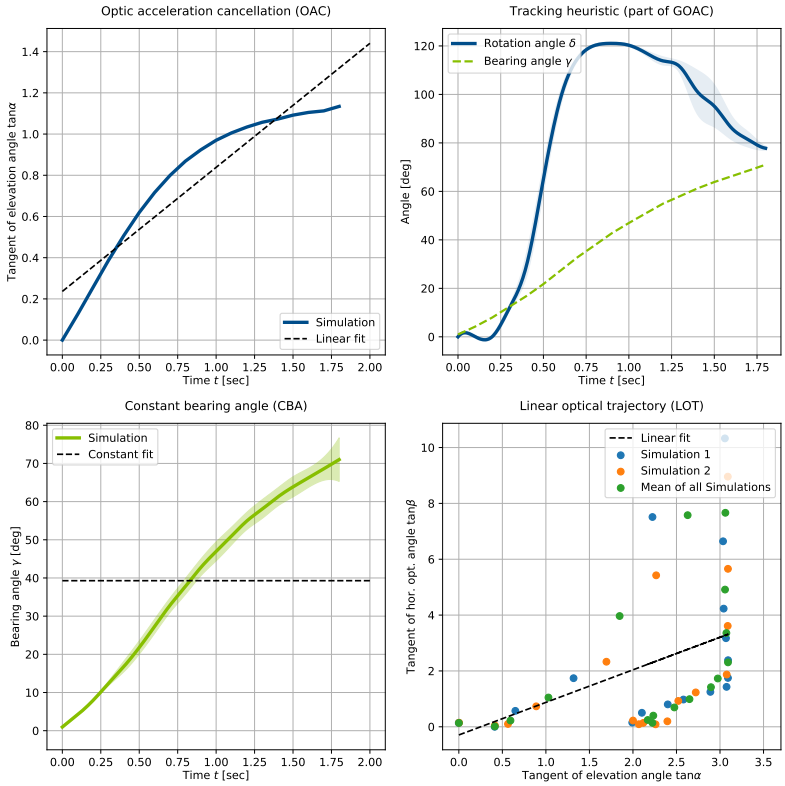

Figure A.2.: Heuristic (OAC, CBA, GOAC, and LOT) plots for the simulation, when the catcher start at (7,4). The corresponding result plot is shown in Figure A.1

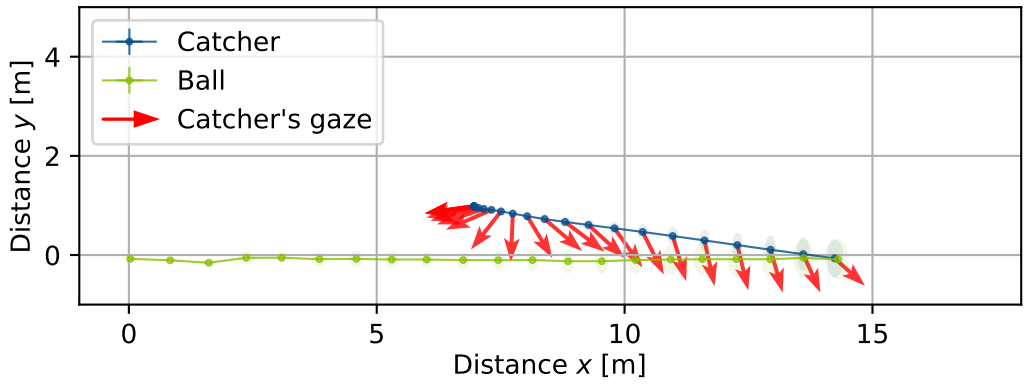

Figure A.3.: Result of the simulation for the start position (7,1) for the catcher averaged over three MPC-simulation runs. The heuristics corresponding to this result is shown in Figure A.4

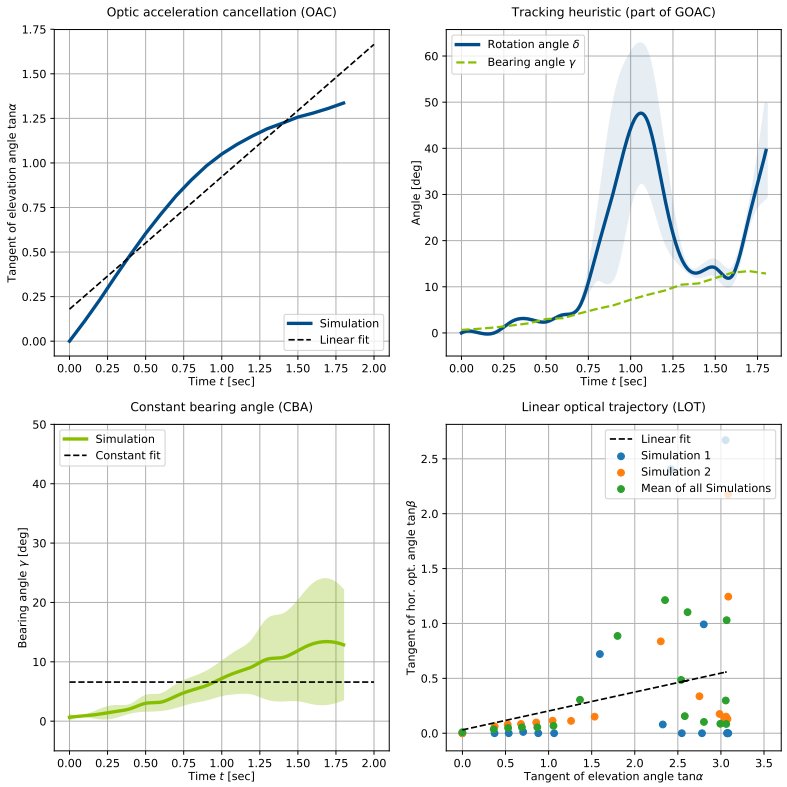

Figure A.4.: Heuristic (OAC, CBA, GOAC, and LOT) plots for the simulation, when the catcher start at 7,1). The corresponding result plot is shown in Figure A.3

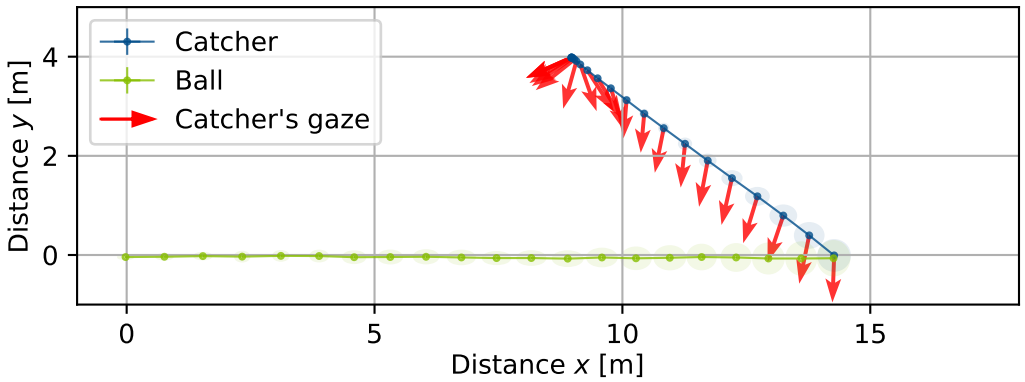

Figure A.5.: Result of the simulation for the start position (9,4) for the catcher averaged over three MPC-simulation runs. The heuristics corresponding to this result is shown in Figure A.6

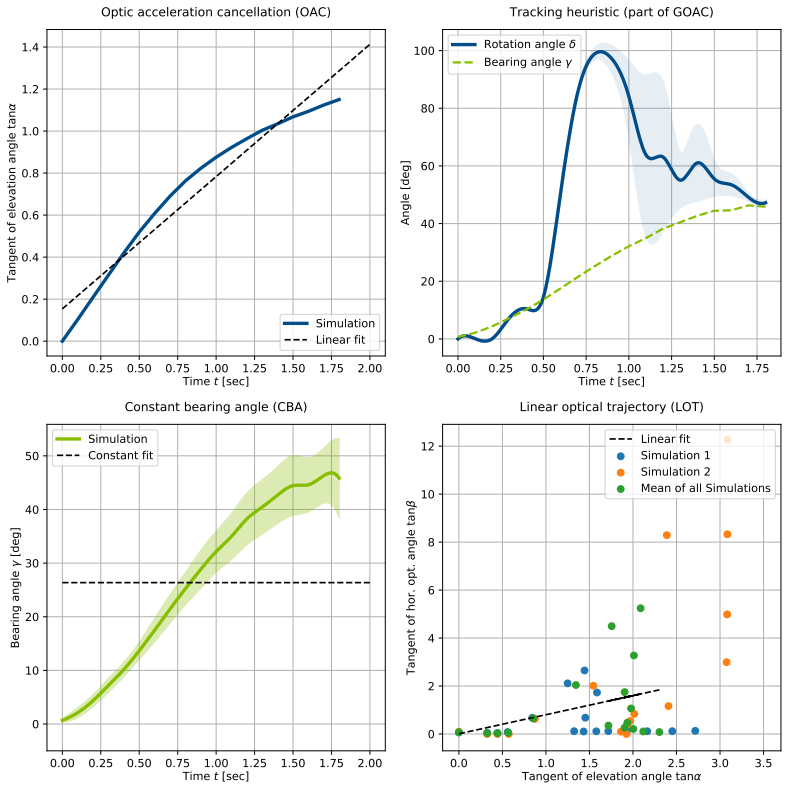

Figure A.6.: Heuristic (OAC, CBA, GOAC, and LOT) plots for the simulation, when the catcher start at (9,4). The corresponding result plot is shown in Figure A.5

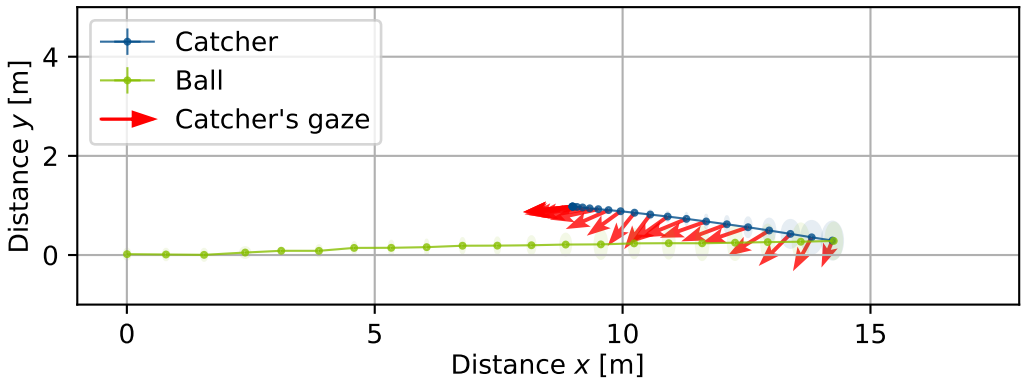

Figure A.7.: Result of the simulation for the start position (9,1) for the catcher averaged over three MPC-simulation runs. The heuristics corresponding to this result is shown in Figure A.8

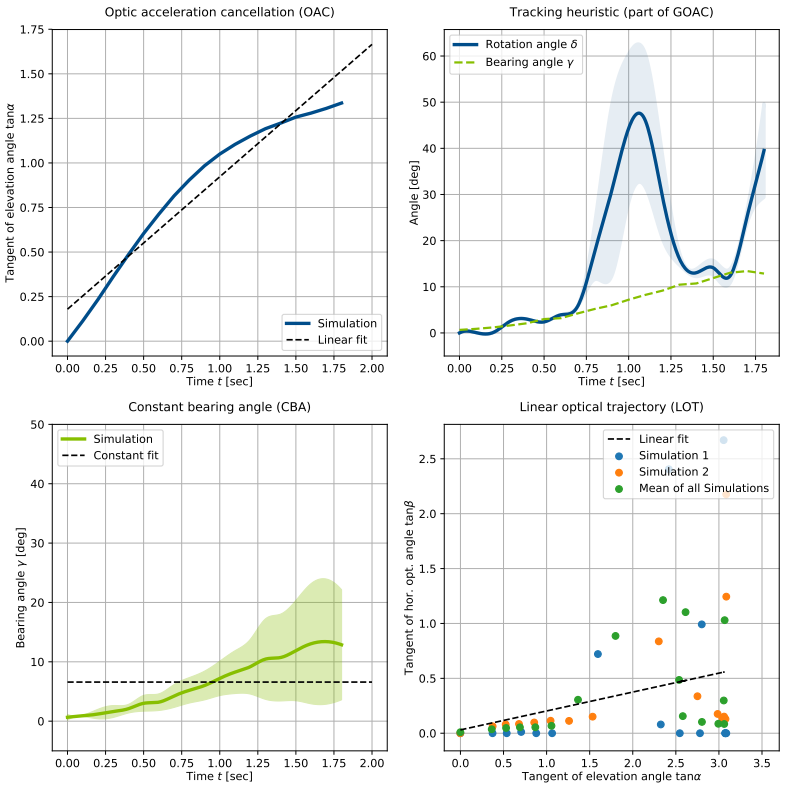

Figure A.8.: Heuristic (OAC, CBA, GOAC, and LOT) plots for the simulation, when the catcher start at 9,1). The corresponding result plot is shown in Figure A.7

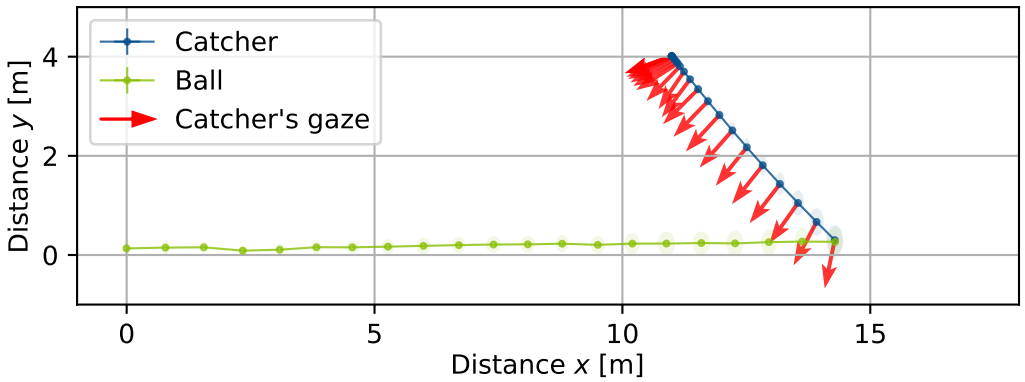

Figure A.9.: Result of the simulation for the start position (11,4) for the catcher averaged over three MPC-simulation runs. The heuristics corresponding to this result is shown in Figure A.10

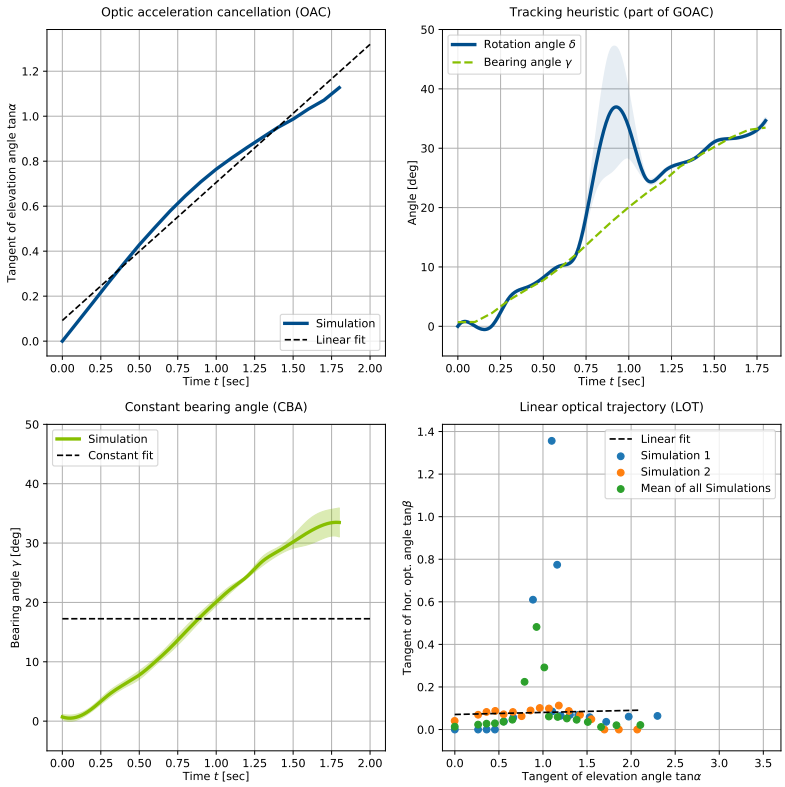

Figure A.10.: Heuristic (OAC, CBA, GOAC, and LOT) plots for the simulation, when the catcher start at (11,4). The corresponding result plot is shown in Figure A.9

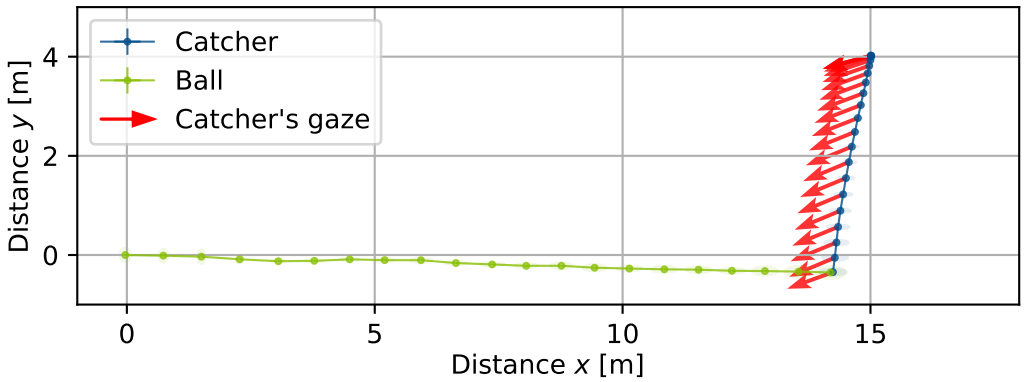

Figure A.11.: Result of the simulation for the start position (15,4) for the catcher averaged over three MPC-simulation runs. The heuristics corresponding to this result is shown in Figure A.12

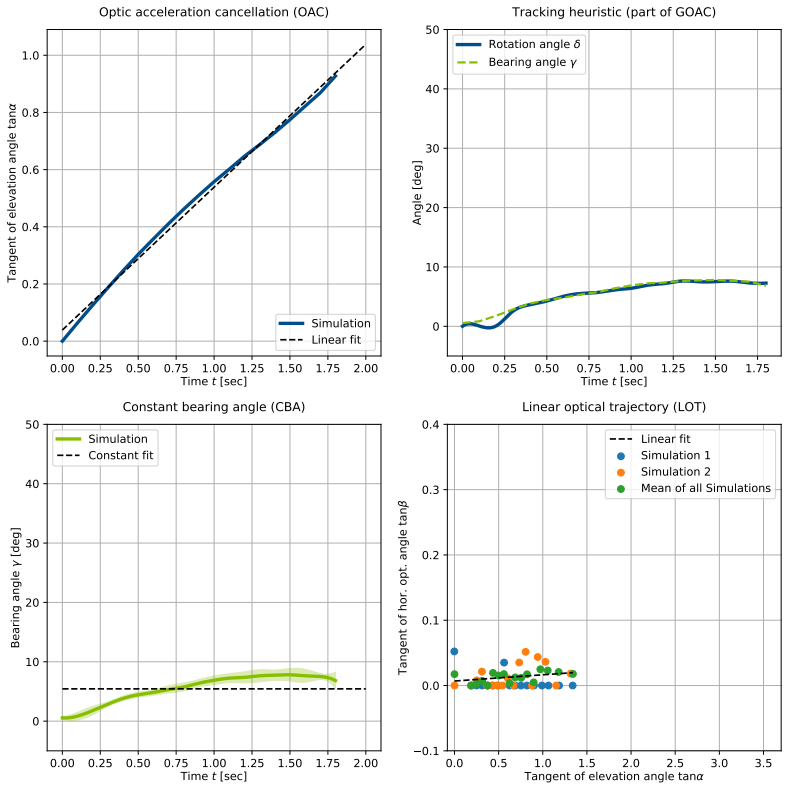

Figure A.12.: Heuristic (OAC, CBA, GOAC, and LOT) plots for the simulation, when the catcher start at (15,4). The corresponding result plot is shown in Figure A.11