University of Erlangen-Nuremberg

Multimedia Communications and Signal Processing

Prof. Dr.-Ing. André Kaup

Master thesis

Modeling and learning of optimal strategies – a case study on catching balls

by Boris Belousov

January 2016

Supervisors:

Prof. Jan Peters, Prof. Gerhard Neumann, TU Darmstadt;
Prof. Walter Kellermann, FAU Erlangen

Erklärung

Ich versichere, dass ich die vorliegende Arbeit ohne fremde Hilfe und ohne Benutzung anderer als der angegebenen Quellen angefertigt habe, und dass die Arbeit in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegen hat und von dieser als Teil einer Prüfungsleistung angenommen wurde. Alle Ausführungen, die wörtlich oder sinngemäß übernommen wurden, sind als solche gekennzeichnet.

Ort, Datum

Unterschrift

Contents

Abstract

How do humans run to catch a ball? Two explanations have been proposed in the literature. One view holds that people predict the trajectory of the ball ahead of time and plan their future actions accordingly; thus, "at some subconscious level, something functionally equivalent to the mathematical calculations is going on" [5]. This is the *model-based* approach, and it suggests that people use *predictive* catching strategies. The competing *heuristic* approach, on the contrary, prescribes purely *reactive* strategies; it says that people "ignore all causal variables necessary to compute the trajectory of the ball – the initial distance, velocity, angle, air resistance, speed and direction of wind, and spin . . . by paying attention to only one variable" [7].

In this work, we demonstrate in computer simulation that if the model-based view is correct and humans indeed have an internal (inherently uncertain) model of the world, and assuming they act optimally (maximizing the probability of catching the ball) under biological constraints (limited maximum velocity, finite field of view), the resultant interception trajectories obey heuristics. Thus, *both theories* are shown to be *compatible* with each other; moreover, the *model-based* view turns out to be *more general* because it can explain catches to which heuristics do not apply. Lastly, we show that both *reactive* and *predictive* behaviors naturally emerge from the *same computation*, with the switch between them being defined by two task specific parameters: model to observation noise ratio and reaction time to task duration ratio.

Mathematical notation

x, y, z	Coordinates
$\dot{x}$, $\ddot{x}$	First and second time derivatives
g, π	Standard gravity and ratio of a circle's circumference to its diameter
F_c, F, θ	2D vector of force, its module, and direction w.r.t. the x-axis
d, d_{xy}	Gaze direction (3D unit vector) and its projection onto the xy-plane
ϕ, ψ, ω_ϕ, ω_ψ	Angles that define d and corresponding angular velocities
μ	Damping coefficient in the model of the catcher
$\mathrm{f}(x(t),\, u(t))$	Continuous dynamics
$f(x_k, u_k)$	Discrete dynamics
T, τ, N	Task duration, discretization time step, and number of steps
t, k	Continuous time and discrete time
z_k, $h(x_k)$	Observation at time k and observation function
ϵ, Q; δ, R	System and observation noises with their covariances
$\mathcal{N}(\mu,\, \Sigma)$	Normal distribution with mean μ and covariance Σ
σ^2	Observation variance of the ball position
r; Ω	Vector between the catcher and the ball; angle between d and r
b	Belief or its representation through sufficient statistics
J, w_0, $w_1\, w_2$	Cost function and adjustable weights
I	Identity matrix

Chapter 1

Introduction

In the late 1950s, while studying decision-making, Nobel laureate Herbert A. Simon, who coined the term *bounded rationality*, pointed out that oftentimes "people seek solutions or accept choices or judgments that are 'good enough' for their purposes" [18], rather than seeking optimal solutions, simply because real-world problems are too complex to consider all possible outcomes and make truly rational decisions. This idea was further developed by Amos Tversky and Daniel Kahneman in the 1970s in their study of heuristics and biases in human decision making [20]. In the late 1990s, Gerd Gigerenzer went even further to propose that heuristics are superior to optimization and "less information, computation, and time can in fact improve accuracy" [7]; thus, in the course of years, heuristics transcended from being a mere consequence of bounded rationality to being the main cause of human actions. Although this view is not universally accepted [7], it dominates in certain particular cases, such as catching a flying ball, for example, where an alternative optimization based explanation has not been offered.

Several heuristics have been proposed to explain how an outfielder catches a baseball [4][12][14]. Despite being different, they all have been experimentally confirmed, which suggests that there is a common unifying cause. A peculiar property of heuristics is that they are model-free, meaning that they do not rely on the prediction of the ball trajectory. There is, however, strong evidence that humans do have a model of gravity [13] and they do predict future ball positions when catching a tennis ball [8]. How can that be then that catching a tennis ball, humans use predictive models, and when catching a baseball, they rely on heuristics? Is there anything special about catching a baseball?

We show that there is no paradox here: heuristics indeed hold in baseball catching, but they are a consequence of model-based optimal behavior. Our contribution is in framing the problem of ball catching as a constrained optimization problem under uncertainty; the catcher tries to intercept the ball by continuously predicting the ball trajectory and adjusting his actions accordingly, while for an external observer he may seem to be following heuristics. Current work continues research started in [10]. This view explains why there can exist many ball catching heuristics, which are just attributes of a catch, not the cause; and it shows that there is no fundamental difference between catching a tennis ball and a baseball, which are instances of the same problem with slightly different ball models.

Chapter 2

Computational model

This chapter introduces a mathematical model of the system consisting of a ball and a catcher which is meant to reflect the human behavior when catching a baseball. First, we develop a noise-free dynamical model capturing relevant physical properties of the ball and the catcher, which is then extended with a noise model reflecting the imperfections of human sensory system and predictive capabilities; second, we formulate the problem in the belief space to enable planning under uncertainty; and third, we describe an optimization based control scheme that is meant to mirror the thought process of the human when running to catch the ball.

2.1 Model specification

2.1.1 Model of the ball

The ball is modeled as a point mass in a uniform gravitational field. Aerodynamic forces are neglected, as they have only a small percentage effect on a trajectory of a baseball, according to [4]. State of the ball is completely defined by its coordinate and velocity:

$$\mathbf{x}_b = \begin{bmatrix} x_b & y_b & z_b & \dot{x}_b & \dot{y}_b & \dot{z}_b \end{bmatrix}^T . \tag{2.1}$$

Equations of motion read:

$$\begin{aligned} \ddot{x}_b &= 0, \\ \ddot{y}_b &= 0, \\ \ddot{z}_b &= -g, \end{aligned} \tag{2.2}$$

where $g \approx 9.81\,\mathrm{m/s^2}$ is the standard gravity. Trajectories of the ball are parabolic, unless the initial velocity is collinear with the z-axis. Figure 2.1 shows an example trjectory.

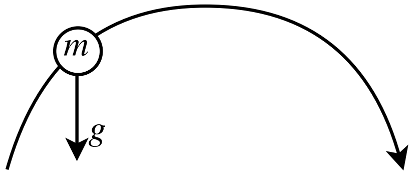


Figure 2.1: A simple kinematic ball model.

2.1.2 Model of the catcher

The catcher moves in xy-plane by applying force to himself. In figure 2.2, the force is denoted by F_c and the projection of the gaze direction d onto xy-plane by d_{xy}.

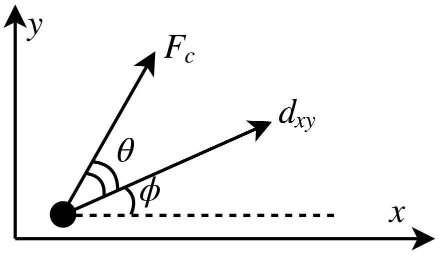


Figure 2.2: The catcher is a 2D point mass with a 3D gaze vector attached to it.

The catcher controls the module of the force $F = |F_c|$, direction θ in which it is applied, and gaze direction d (through two angles ϕ and ψ schematically shown in figure 2.3). The state and control vectors are defined as follows:

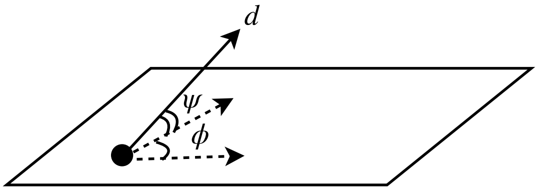


Figure 2.3: The gaze direction d is defined by the angles ϕ and ψ.

$$\mathbf{x}_c = \begin{bmatrix} x_c & y_c & \dot{x}_c & \dot{y}_c & \phi & \psi \end{bmatrix}^T, \tag{2.3}$$

$$\mathbf{u} = \begin{bmatrix} F & \theta & \omega_\phi & \omega_\psi \end{bmatrix}^T. \tag{2.4}$$

Equations of motion are given by:

$$\begin{aligned} \ddot{x}_c &= F\cos(\phi+\theta) - \mu\dot{x}_c, \\ \ddot{y}_c &= F\sin(\phi+\theta) - \mu\dot{y}_c, \\ \dot{\phi} &= \omega_\phi, \\ \dot{\psi} &= \omega_\psi. \end{aligned} \tag{2.5}$$

Why this model?

This particular model with friction μ was chosen because it limits maximum velocity. To fit the model parameters F and μ, we used data from Usain Bolt's record sprint at the 2008 Olympic Games in Beijing [6]; there, during the 100 m dash, he reaches velocity of 12 m/s in 3 s and maintains it till the end. Only the ratio F/μ is physically important because we are free to choose the mass of the catcher. In all experiments, $F_{\max} = 10\,\mathrm{N/kg}$ and $\mu = 10/12\,\mathrm{s}^{-1}$, so that the maximum velocity is limited by 12 m/s. We chose to control angles defining the gaze direction through the first derivatives because it simplifies computations and allows modeling of quick turns with relatively large discretization time steps while also reflecting the physical fact that turning the head is faster and easier than gaining momentum.

Box constraints

As already pointed out, in order to stay realistic, one has to bound the maximum applicable force. Other controls should be limited as well:

$$\begin{array}{rcccl} -\pi & \leq & \theta & \leq & \pi, \\ -2\pi & \leq & \omega_\phi & \leq & 2\pi, \\ -2\pi & \leq & \omega_\psi & \leq & 2\pi. \end{array} \tag{2.6}$$

Constraints of this type are called *box constraints*. The first inequality constrains the direction of the force θ to lie within an interval covering 2π, which does not really restrict the solution, but sufficiently simplifies optimization. The latter two inequalities forbid the catcher to turn too quickly.

Not only controls, but also states need to be restricted. Namely, the catcher is not allowed to look directly above him to avoid singularity, and looking below the horizon is excluded as irrelevant for ball catching. So, the gaze vector is required to lie on the upper hemisphere with a little hole in the middle:

$$0 \leq \psi < \pi/2. \tag{2.7}$$

In experiments, the upper bound is set a little bit smaller than $\pi/2$ to facilitate efficient optimization.

Running forward is faster than backwards

Humans run forward faster than backwards. We express it in the model by making the maximum force state-dependent:

$$0 \leq F \leq F_1 + F_2 \cos\theta. \tag{2.8}$$

The right-hand side smoothly interpolates the maximum force between $F_1 - F_2$ and $F_1 + F_2$ for any angle θ, so that running sideways is also allowed. Note that any other suitable function of θ could have been used instead of $\cos\theta$. This particular choice of interpolating function is motivated by the fact that small deviations in the angle θ

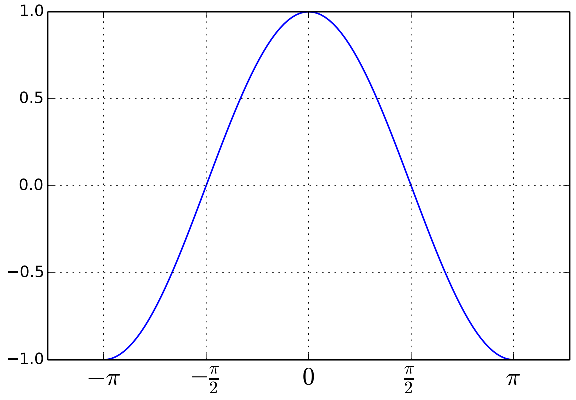


Figure 2.4: Graph of $\cos\theta$ for $|\theta| \leq \pi$.

about 0 and $\pm\pi$ should have small effects on the maximum force, because turning the head a little bit when running forward does not affect the velocity too much. Since $\cos\theta$ is smooth and quadratic in those regions, it serves the purpose well. Figure 2.4 shows the interpolating function. Running forward corresponds to $\theta = 0$, and running backwards corresponds to $|\theta| = \pi$.

One question still remains open; namely, "How to choose F_1 and F_2?" We set the maximum force $F_1 + F_2$ to match the maximum attainable *forward running velocity*, as before. The lower bound $F_1 - F_2$ is set, in its turn, to match the maximum attainable *backward running velocity*, which we get from the book of backward running records [17]. This choice yields numerical values $F_1 = 7.5\,\mathrm{N/kg}$ and $F_2 = 2.5\,\mathrm{N/kg}$.

2.1.3 Discrete dynamics

What we have developed so far is a dynamical model of a ball and a catcher, which can be represented by the first order differential equation:

$$\dot{x}(t) = \mathrm{f}(x(t), u(t)), \tag{2.9}$$

where vector $x = [\mathbf{x}_b; \mathbf{x}_c]$ holds the state of the ball and the catcher, and the nonlinear function f represents the right-hand side of the dynamical equations (2.2) and (2.5). As the next step, we *discretize* the continuous dynamics (2.9):

1. Discrete time. Let the task duration T be fixed, and let the discretization time step be $\tau = T/N$, where N is the number of time intervals. Discretization enables us to work with vectors x_k, $k = 0, 1, \ldots, N$, instead of functions $x(t)$, $t \in [0, T]$.

2. Discrete controls. We approximate $u(t)$ by a piecewise constant function that equals u_k on the k-th time interval, as is commonly done within the multiple shooting framework.

3. Discrete states. Under above stated assumptions, a discrete equivalent of the continuous dynamics (2.9) can be written as:

   $$x_{k+1} = f(x_k, u_k), \qquad k = 0, 1, \ldots, N-1, \tag{2.10}$$

   with f being an integrator function (we use different styles to discern continuous dynamics f from discrete dynamics f, which should not lead to confusion, since only discrete dynamics is used in the following). We use the 4-th order explicit Runge-Kutta method with 10 intermediate points to integrate (2.9) on each time interval. Simpler integration schemes diverge for big time steps (simulations run with $\tau = 0.1\,\mathrm{s}$), while more complicated integrators, such as CVODES, require a lot of computation without much gain in accuracy on the relatively simple integration problem at hand. Therefore, the intermediate solution has been favored.

2.1.4 Observation model

This section introduces an observation model that glues the models of the ball and catcher together by allowing the catcher to “look” at the ball.

An observation z_k at time $k \in \{0, 1, \ldots, N\}$ is a function of state $z_k = h(x_k)$. We assume that the catcher gets to observe the position of the ball, his own position and the gaze direction:

$$z = \begin{bmatrix} x_b & y_b & z_b & x_c & y_c & \phi & \psi \end{bmatrix} \tag{2.11}$$

As can be easily seen, this observation model allows the catcher to “see” the ball even if it is behind him, which is, of course, unacceptable. This problem is fixed in the next session, where the field of view is implicitly modeled through state-dependent noise.

2.1.5 Noise model

The notion of *uncertainty* is crucial for describing human ball catching behavior. The catcher has a model of the world which lets him predict future ball positions given his current knowledge. But neither the model nor the observations are perfect, which means predictions get worse, the further into the future the catcher tries to predict. To model these effects, we introduce uncertainty in both dynamics and observations:

$$x_k = f(x_{k-1}, u_{k-1}) + \epsilon_k, \qquad \epsilon_k \sim \mathcal{N}(0, Q), \tag{2.12}$$

$$z_k = h(x_k) + \delta_k, \qquad \delta_k \sim \mathcal{N}(0, \boldsymbol{R(x_k)}). \tag{2.13}$$

The essential point here is that the *observation noise is state-dependent* (marked in bold). By shaping the matrix-valued function $R(x_k)$, we can model the field of view: if the ball is in the center of the catcher's field of view, which coincides with the gaze direction, he gets very good observations; on the other hand, if the ball is outside of the field of view, he gets no observations at all, or very noisy observations. We also assume that the noise is distance-dependent: the closer the ball, the better the observations. Variance in observations of the ball position is then given by:

$$\sigma^2 = |r|(\sigma^2_{\text{max}}(1 - \cos\Omega) + \sigma^2_{\text{min}}), \tag{2.14}$$

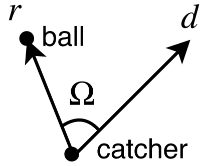


Figure 2.5: The catcher is looking in the direction d, while the ball is $|r|$ m away from him. The amount of uncertainty in observations depends on the angle Ω and the distance $|r|$ to the ball.

where r is the vector from the catcher to the ball, Ω is the angle between the gaze direction and vector r, and $\{\sigma_{min}, \sigma_{\max}\}$ are two adjustable parameters. If the catcher is looking at the ball ($\cos\Omega = 1$), the noise is minimal. If the ball is exactly behind the catcher ($\cos\Omega = -1$), the noise is maximal. In our experiments $\sigma^2_{\min} = 0.01$ and $\sigma^2_{\max} = 10$. The observation covariance matrix is $R = \operatorname{diag}\{\sigma^2, \sigma^2, \sigma^2, 0, 0, 0, 0\}$. The system noise covariance matrix is defined as a block-diagonal matrix with isotropic blocks for the ball and the catcher $Q = \operatorname{diag}\{\sigma_b^2 \mathbf{1}_6, \sigma_c^2 \mathbf{1}_6\}$ ($\mathbf{1}_n$ is a vector of n ones), $\sigma_b^2 = 1 \times 10^{-3}$ and $\sigma_c^2 = 1 \times 10^{-5}$.

Model vs observations

As we shall see, behavior of the catcher greatly depends on the system to observation noise ratio σ_b^2/σ^2. This ratio defines what the catcher believes more: his model of the world or observations. When the ratio is small, the catcher trusts his model and reluctantly follows the observations. When the ratio is big, the catcher abandons the model and relies on the observations.

2.2 Belief space formulation

Now, after introducing uncertainty into the model, we have to revise our definition of state. It is, indeed, no longer enough to only specify the position of the ball, but one has to additionally specify the uncertainty about the ball position.

2.2.1 Belief dynamics

Belief is a distribution over states. We consider Gaussian beliefs, which can be represented by a mean vector and a covariance matrix, $b_k = (\mu_k, \Sigma_k)$. Thus, the mean is associated with the most likely state of the system, and the covariance matrix gives the uncertainty about that state.

Due to nonlinearity of the dynamics and observation function, propagation of the belief is intractable and does not preserve Gaussianity; therefore, we are forced to resort to approximate filtering techniques. We chose to use the extended Kalman filter because it gives the belief propagation step in closed form which can be automatically differentiated with respect to the covariance matrix; the update rule of the Kalman filter $(\mu_{k-1}, \Sigma_{k-1}, u_{k-1}, z_k) \to (\mu_k, \Sigma_k)$ is given by [19]:

$$\begin{array}{lrcl} & \bar{\mu}_k & = & f(\mu_{k-1}, u_{k-1}), \\ \text{[prediction]} & \bar{\Sigma}_k & = & A_{k-1}\Sigma_{k-1}A_{k-1}^T + Q_k, \\ & K_k & = & \bar{\Sigma}_k C_k^T (C_k \bar{\Sigma}_k C_k^T + R_k)^{-1}, \\ \text{[correction]} & \mu_k & = & \bar{\mu}_k + K_k(\boldsymbol{z_k} - h(\bar{\mu}_k)), \\ & \Sigma_k & = & (I - K_k C_k)\bar{\Sigma}_k, \end{array} \tag{2.15}$$

where $A_{k-1} = \left.\frac{\partial f}{\partial x}\right|_{\mu_{k-1}}$, $C_k = \left.\frac{\partial h}{\partial x}\right|_{\bar{\mu}_k}$, and I is the identity matrix. System noise is small and fixed $Q_k \equiv Q$, while the observation noise is state-dependent $R_k = R(\bar{\mu}_k)$; notice that $R(\bar{\mu}_k) \neq R(x_k)$, which means we can only use predicted observation noise for planning because x_k is not directly accessible. Equations (2.15) define the belief dynamics, which is stochastic $\mu_k = \bar{\mu}_k + \xi_k$ due to the dependence on the observation z_k (marked in bold); the innovation term is Gaussian distributed $\xi_k \sim \mathcal{N}(0, K_k C_k \bar{\Sigma}_k)$.

2.2.2 Three types of uncertainty

Once we have the belief dynamics, we can predict future states of the system by forward simulating the dynamics. When reasoning about a future belief state of the system, three types of uncertainty should be discerned [3]:

1. *System uncertainty* is the uncertainty about the state, when observations are not available in principle.

2. *A posteriori* uncertainty is the uncertainty after observations will be incorporated; it is what the Kalman filter outputs.

3. *A priori* uncertainty is the uncertainty before the observations are incorporated; it is different from the system uncertainty: in order to predict a future belief,

it integrates out the observations which have not yet been acquired instead of ignoring them, as was the case with the system uncertainty.

A more detailed analysis of uncertainties is presented in Appendix (A.1), where, among other things, it is shown that *system uncertainty* $=$ *a priori* $+$ *a posteriori*.

2.2.3 Cost function

The goal of the catcher is to catch the ball; more precisely, to maximize the probability of catching it, because not every attempt is guaranteed to be successful. We assume the catch is successful if the ball is less than 0.5 m away from the catcher at the final time; we do not model the last phase of the catch during which the catcher relies on hand-eye coordination.

In Appendix (A.2), we show that maximization of the probability of catching the ball can be encoded by the following cost function, which is to be minimized:

$$\begin{aligned} J = w_0 \|\mu_{bN} - \mu_{cN}\|_2^2 \qquad & \text{[final position]} \qquad (2.16a)\\ + w_1 \operatorname{tr} \Sigma_N \qquad & \text{[final uncertainty]} \qquad (2.16b)\\ + \tau \sum_{k=0}^{N-1} u_k^T R u_k \qquad & \text{[total energy]} \qquad (2.16c)\\ + \tau w_2 \sum_{k=0}^{N-1} \operatorname{tr} \Sigma_k \qquad & \text{[running uncertainty]} \qquad (2.16d) \end{aligned}$$

Here, w_0, w_1, w_2 and R are adjustable weights; μ_{bN} and μ_{cN} are the belief positions of the ball and the catcher at the final time; τ is the discretization time step. Numerical values are $\tau = 0.1$, $w_0 = w_1 = 1000$, $w_2 = 100$, $R = \operatorname{diag}\{10, 1, 1, 0.1\}$. In words, this cost functions says, "Be at the interception point with minimal uncertainty about the system and spend as little energy as possible."

2.3 Belief space planning

In this section, we finally describe how the catcher chooses his actions. At every time step he predicts the future based on his current knowledge and decides what course of actions to take in order to intercept the ball; then he executes the first action, gets a new observation, and repeats the cycle. The key problem here is planning, and we solve by trajectory optimization [15].

2.3.1 Model predictive control

In order to perform planning under stochastic dynamics (2.15), we assume that future observations are *maximum likelihood observations*, i.e. $z_k = h(\bar{\mu}_k)$. This assumption makes dynamics deterministic [16]:

$$\begin{aligned} \mu_k &= f(\mu_{k-1}, u_{k-1}), \\ \Sigma_k &= (I - K_k C_k)\bar{\Sigma}_k, \end{aligned} \tag{2.17}$$

enabling the use of standard deterministic planning and control tools.

Dynamics (2.17), cost function (2.16), and constraints (2.6-2.8) fully specify an optimization problem, solution of which yields an open-loop sequence of controls $u_{0:N-1}$. The catcher executes the first action from that sequence, incorporates the new observation and repeats the cycle. This strategy – planning for N time steps, executing the first action, and replanning again – is known as *model predictive control* [11].

Reaction time

Amount of time it takes a sensory stimulus to trigger a behavioral response is called *reaction time*. For college-age individuals, it takes on average 200 ms to react to a visual stimulus [9]. We incorporate this reaction time into our model by delaying incoming observations. The catcher has to compensate the delay by forward simulating his internal dynamical model.

2.3.2 Covariance-free trajectory optimization

When solving the above specified optimization problem, standard methods have to be modified due to inclusion of covariance matrices into the state of the system. Three formulations of the optimization problem are possible:

1. Shooting (optimize over controls only). Every future state depends on the initial state and applied sequence of controls. Thus, everything depends on the controls only. This formulation contains the least number of optimization variables, but, unfortunately, suffers from sensitivity problems.

2. Partial collocation (optimize over controls and mean states). This is the method used in this work. It treats mean states as optimization variables, while imposing dynamics as a constraint. Covariance matrices enter the problem only through the cost function.

3. Full collocation (optimize over controls, mean states, and covariances). This is a straightforward extension of the standard collocation method to the case when state includes a covariance matrix. This method has several drawbacks. First, dimensionality of the optimization problem dramatically increases because of additional $O(n^2)$ optimization variables, where n is the dimensionality of the mean state μ. Second, special care must be taken to preserve positive semi-definiteness of covariance matrices in the course of optimization. Third, nonlinear matrix dynamics (2.17), involving, among other things, matrix inversion, has to be imposed as a constraint.

The first two approaches are called *covariance-free trajectory optimization* methods, because they treat covariance matrices as dependent variables and not as free parameters, thus keeping the computational burden low. The partial collocation method has been recently shown to be the fastest among the three and to deliver state-of-the-art performance [15], therefore it was chosen to be used in this work.

2.3.3 Tools for covariance-free trajectory optimization

We rely on two key advances in numerical optimal control – (i) automatic differentiation to accurately compute gradients of the nonlinear objective, and (ii) state-of-the-art interior point optimizers to efficiently solve large-scale optimization problems.

Automatic differentiation

For the partial collocation method, the objective $J = J(\mu_{0:N}, \Sigma_0, u_{0:N-1})$ is a function of the initial uncertainty Σ_0, sequence of mean states $\mu_{0:N}$ and sequence of controls $u_{0:N-1}$. The explicit representation (2.16), however, also depends on the future uncertainties $\Sigma_{1:N}$, which greatly complicates the computation of the gradient, because these uncertainties are recursive functions of the initial uncertainty. The gradient, nevertheless, can be computed by *automatic differentiation*, a technique for evaluating derivatives of computer represented functions that can deliver directional derivatives up to machine precision. Note that automatic differentiation is different from symbolic differentiation, which directly operates on functions represented in a special-purpose symbolic language. Several computational tools have been developed to facilitate automatic differentiation; we use CasADi [1] because it supports matrix-valued atomic operations and tightly integrates state-of-the-art nonlinear optimizers.

Interior point optimizer (IPOPT)

We compute gradients using automatic differentiation and pass them to IPOPT [2], which implements a primal-dual interior point method. We use the Broyden-Fletcher-Goldfarb-Shanno update within IPOPT in order to avoid prohibitively expensive computation of the exact Hessian. Due to nonlinear nature of the objective function, good initialization plays a crucial role in optimization; therefore, at every time step, we first solve a simpler problem in the state space which does not include covariance matrices but directly incurs a small cost for not looking at the ball, and then use this solution as initialization for the full problem in the belief space.

Chapter 3

Results

In this chapter, we present results of simulation experiments. We show that the proposed model generates behaviors which agree with heuristics when they are applicable (the ball is in the field of view of the catcher), and produces plausible behaviors even when heuristics are not applicable (the ball is outside of the field of view of the catcher). In the latter case, the catcher has to turn away from the ball for a short period of time in order to gain speed; he compensates for the absence of observations by forward simulating the internal model.

We explore effects of varying model parameters and demonstrate that qualitatively different behaviors emerge depending on how much the catcher relies on the internal model and how long is the reaction time compared to the task duration. Results can be best illustrated on extreme cases. In baseball, the catcher has quite a good model of the ball, and reaction delay (about 200 ms) is small compared to the task duration (about 3 s); therefore, the catcher can get many observations before he catches the ball, which means, he uses a *feedback strategy*. Even if the actual trajectory of the ball deviates from the initial prediction – due to wind change or effects of the spin – the catcher is able to successfully intercept the ball because he continuously gets observations and can adjust his prediction on the go. Another extreme case is table tennis. The ball flies so fast that players have to predict where it will hit the table even before the opponent hits the ball in order to be able to intercept it. Professionals take into account pose of the opponent and current game situation to perform such estimation; reaction delay in this case is big compared to the task duration, which forces players to use *open-loop strategies* and rely on predictive models.

We conclude that the view that humans rely on internal models and act optimally under natural constraints can explain how humans catch balls in games such as baseball. Heuristics may be regarded as necessary conditions of optimality for a restricted class of problems, when the catcher has enough time to continuously tracks the ball. The optimization-based approach is more general than the heuristic approach because it can be uniformly applied to any problem by describing the model and specifying the objective, while heuristics have to be independently discovered for every particular case. So, hitting a ball in table tennis appears to be a completely unrelated problem to catching a ball in cricket from the point of view of heuristics. Within the optimization framework, however, they are both related and differ only in the objective and the model. This generality may be key to understanding how humans can perform many different tasks using the same kind of computation.

3.1 Comparison against heuristics

In this section, three catching scenarios are presented – (i) successful catch, in which case all heuristics hold; (ii) failed catch, in which case heuristics do not hold; and (iii) unexplainable by heuristics successful catch, in which case heuristics do not hold, but the catcher nevertheless successfully intercepts the ball.

We consider four heuristics that have been proposed in the literature to explain how a fielder catches a baseball. They can be divided into two groups: point mass based and gaze based. Point mass based heuristics consider the catcher as a point mass, while gaze based heuristics augment the model by explicitly including the catcher's gaze and analyzing how the ball is positioned within the catcher's field of view. Optic acceleration cancellation (OAC) [4] and constant bearing angle (CBA) [4] heuristics belong to the first category, while generalized optic acceleration cancellation (GOAC) [14] and linear optical trajectory (LOT) [12] belong to the second category. GOAC, in fact, consists of OAC and one additional gaze-based heuristic, so when we refer to GOAC, we refer to that extra heuristic. This convention simplifies comparison with other heuristics.

Not all heuristics are independent [14]. OAC, for example, prescribes only how the catcher should move when he is in the plane spanned by the ball trajectory (it assumes a parabolic trajectory). It is not sufficient on its own, but has to be complemented by a heuristic that would tell the catcher how to move in the horizontal plane. Two alternatives have been proposed in the literature: CBA and GOAC. So, there are two possible combinations: OAC+CBA and OAC+GOAC (more correctly would be to say simply GOAC, as it already includes OAC). LOT, in contrast, is sufficient on its own.

It should be pointed out that since all heuristics are constructively derived from geometric considerations, they present sufficient conditions for a successful catch; thus, if a catcher follows OAC+CBA, OAC+GOAC, or LOT, he is guaranteed to be at the interception point at the right time. All these combinations have been reported to hold in human experiments.

3.1.1 Successful catch

We reproduce the settings from the experimental study [14], which allows us to compare simulation results with real human data. In that study, the ball was thrown from the origin and landed after 3 s at the distance of 15 m away from the origin. Four different initial positions of the catcher were reported, all of which lie within a circle of radius 6 m from the interception point. In the present work, we double the distances in order to enhance graphical representation of the results. All conclusions hold without restriction in the original settings from [14] as well. Along with the simulation results, we also report initial interception plans of the catcher in order demonstrate effects of the system and observation noises on interception trajectories.

Initial plan

Figure 3.1 shows a typical catch, where the catcher has enough time to get to the interception point without rushing. He plans to continuously track the ball in order to reduce the final uncertainty about the system. All four heuristics hold for the plan as can be seen in figure 3.2.

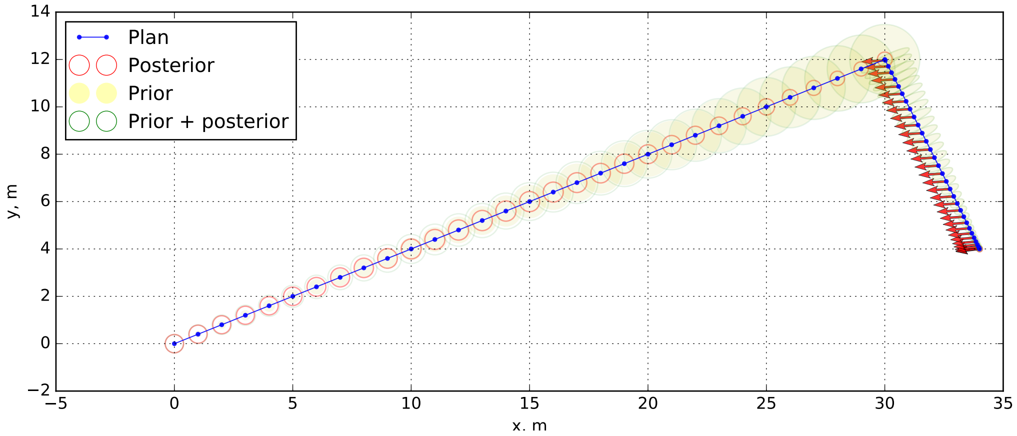


Figure 3.1: Initial interception plan of the catcher. The ball is thrown from the origin with initial uncertainty indicated by the red circle around the origin. The ball lands around the point (30, 12) after about 3 s in the air. The catcher's initial gaze is directed towards the origin. His goal is to be at the interception point at the right time and have the smallest possible posterior uncertainty about the system.

In figure 3.1 the catcher runs forward and to the right. The same analysis can be carried out for running forward and to the left, and backwards to the left and right. Heuristics hold in all four cases.

Simulation

Heuristics hold in simulation as well, with the only difference that the plots are noisier, as can be seen in figure 3.4. That is due to the fact that the catcher has to incorporate new observations, which introduces stochasticity into the system. It is worth noting that the catcher does not necessarily follow a straight trajectory in simulation, contrary to the plan (compare figures 3.3 and 3.1). Human catchers, indeed, rarely run in a straight line.

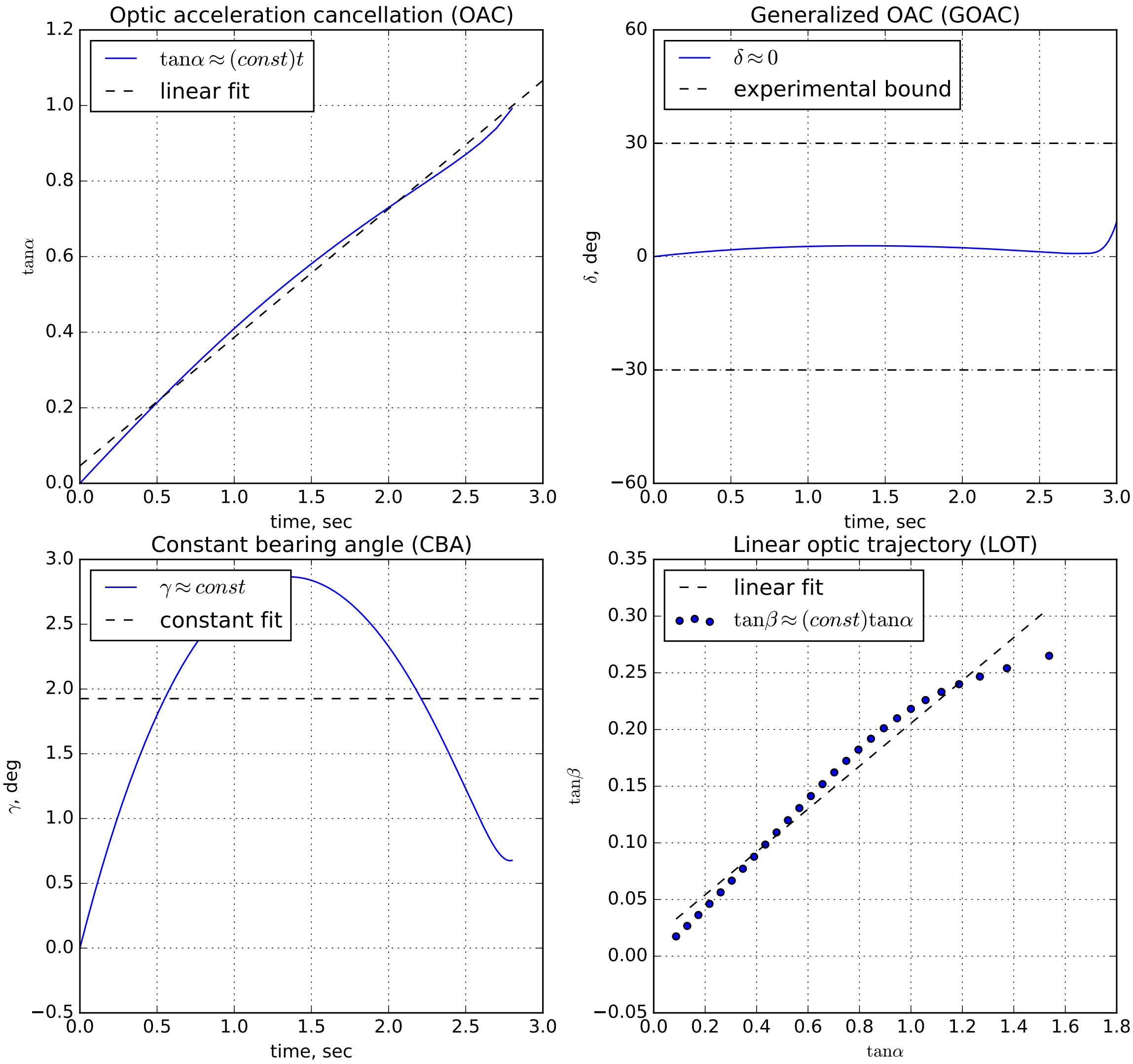


Figure 3.2: Heuristics for the initial interception plan. All four heuristics hold within the precision of human experiments. Although CBA demonstrates seemingly nonlinear behavior, absolute variation is less than 3 deg, which is below the measurement precision of human experiments. Last instants in all graphs should be ignored because heuristics do not apply to the final phase of the catch, where the catcher is assumed to rely on hand-eye coordination instead.

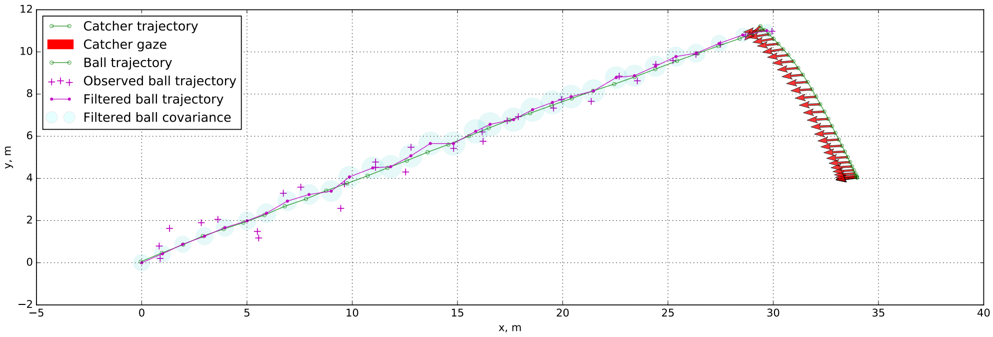


Figure 3.3: Successful catch that obeys heuristics. The catcher continuously tracks the ball and adjusts his interception strategy based on incoming observations. The catcher's trajectory is not necessarily a straight line.

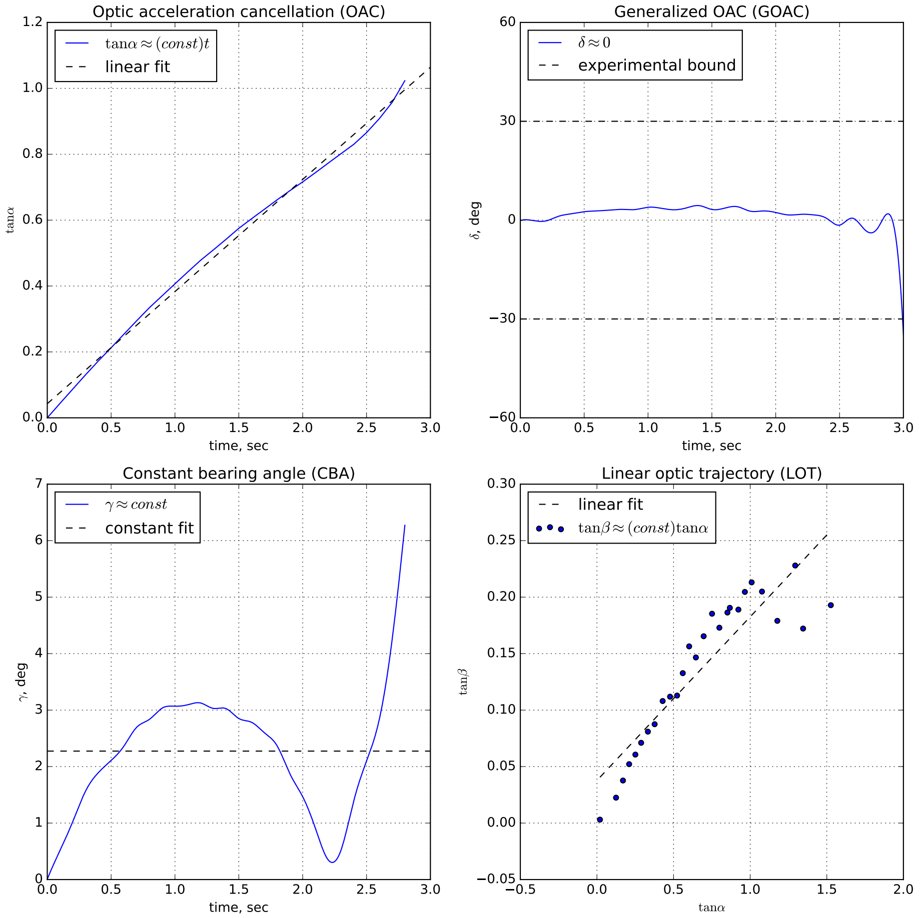


Figure 3.4: Heuristics hold in simulation. CBA stays within a 10 deg margin.

3.1.2 Failed catch

If the catcher fails to intercept the ball, heuristics do not hold, as expected. Figure 3.5 shows a typical failed catch, where the catcher is too slow to intercept the ball. Figure 3.6 depicts the heuristics.

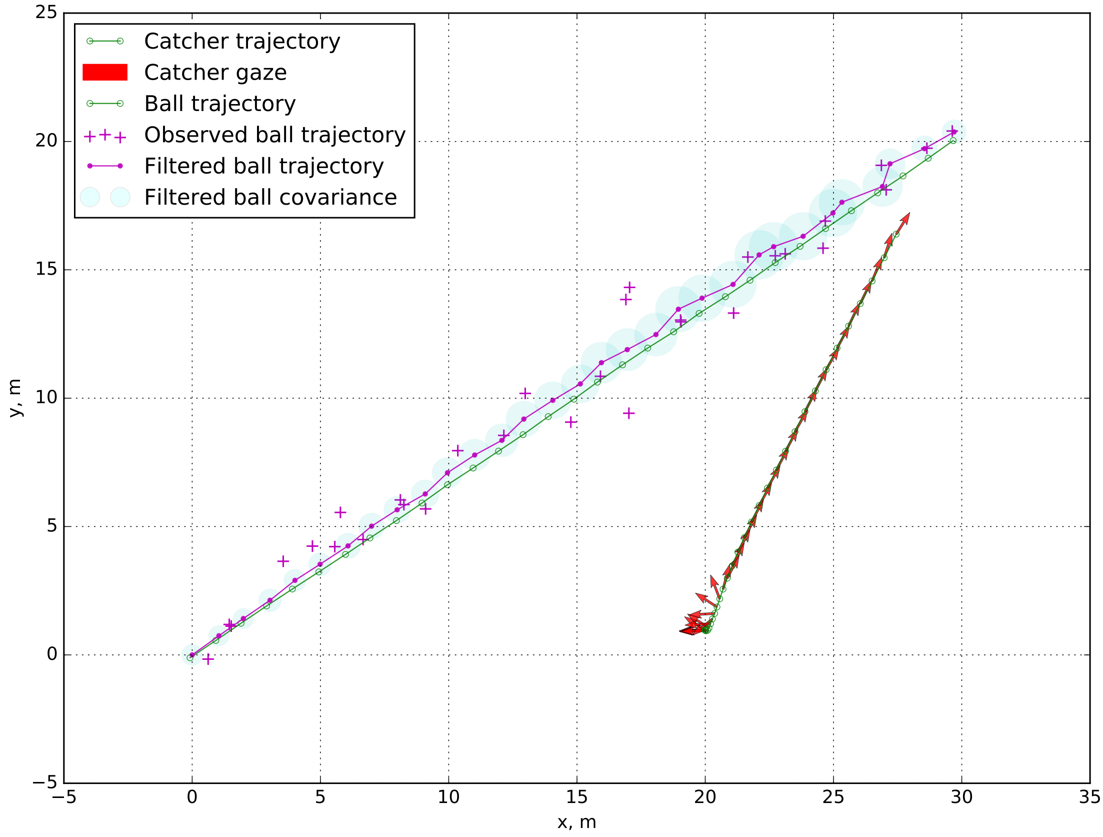


Figure 3.5: Failed catch. None of the heuristics hold in this case, as expected. The ball falls too far from the catcher's initial position, so that the catcher fails to intercept it even if he runs at maximum velocity.

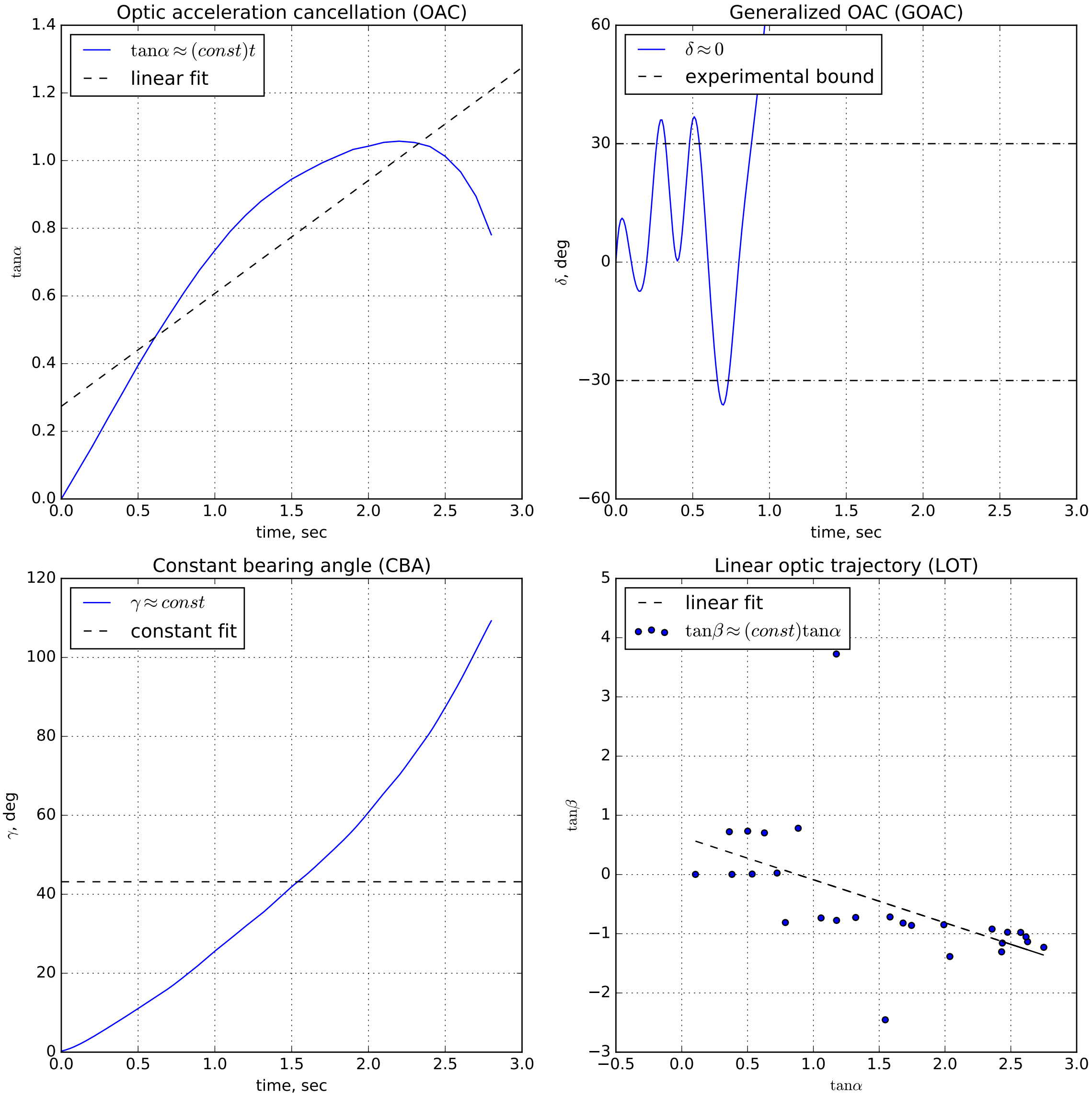


Figure 3.6: Heuristics do not hold if the catcher does not intercept the ball.

3.1.3 Unexplainable (by heuristics) catch

Figure 3.7 gives an example of a successful catch for which none of the heuristics hold, even for the planned trajectory. Heuristics assume that the catcher keeps the ball in the field of view all the time; this assumption is violated if the ball flies over the catcher's head, forcing him to turn around. Figure 3.8 shows heuristics in this case.

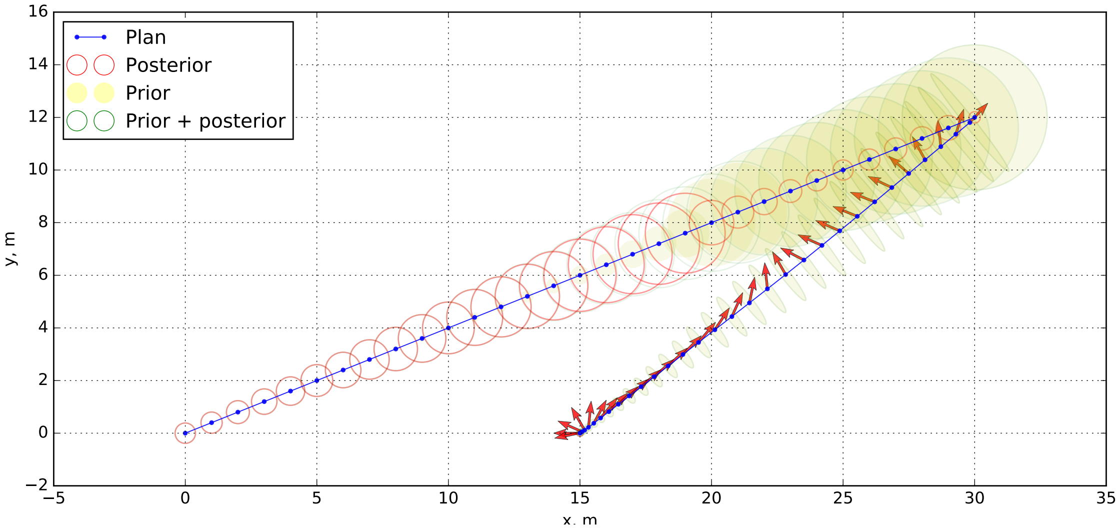


Figure 3.7: If the catcher sees that the ball will fly over his head and fall at a considerable distance from his initial position, he decides to turn away from the ball in the beginning for a short period of time in order to speed up. This behavior follows from the fact that the catcher can gain higher velocity when running forward. Heuristics do not apply in this case, because they all assume that the ball is in the field of view of the catcher all the time.

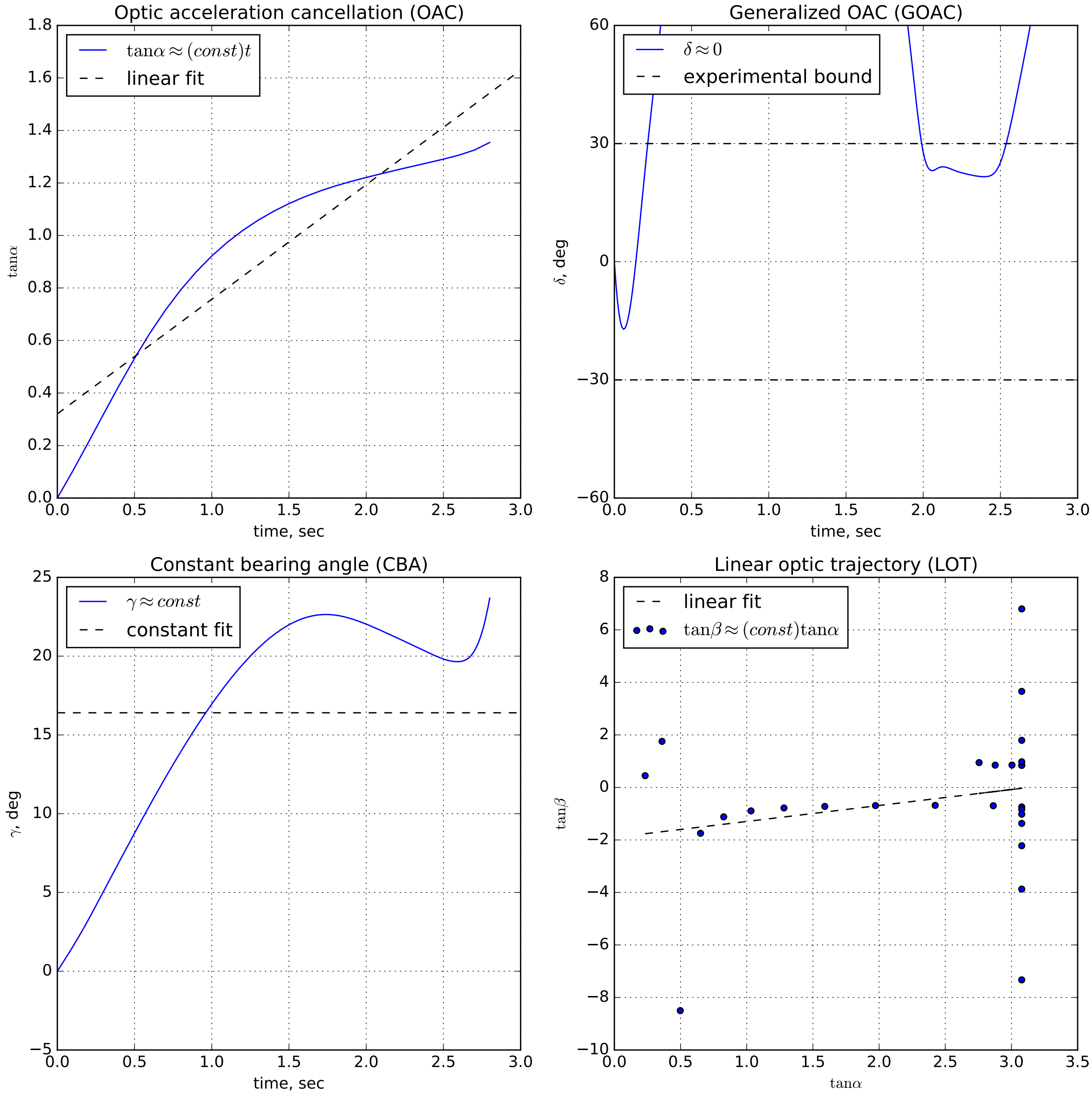


Figure 3.8: Unexplainable catch, heuristics. None of the heuristics hold despite the fact that the catcher successfully intercepts the ball.

3.2 Effects of noise and reaction time

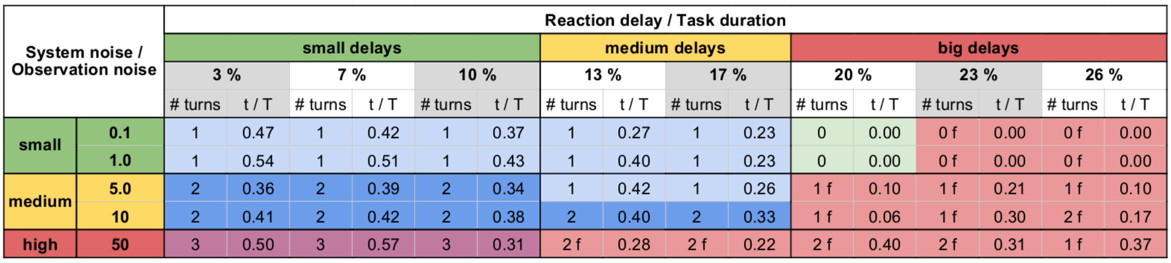
In this section, we analyze effects of varying model parameters on the catcher's strategy. In particular, we focus our attention on system to observation noise ratio (denoted α) and reaction time to task duration ratio (denoted β). These two parameters determine whether the catcher uses open-loop or feedback control policy.

When α is small, the catcher gives more weight to his internal model and relies less on observations. In this case, the catcher chooses between using open-loop and feedback control strategy based on how big is the reaction delay compared to the task duration. If β is small, the catcher is not restricted in the choice of the strategy. In most cases the catcher prefers a mixture of both, predicting for a moderate time into the future and adjusting his prediction based on incoming observations. When β grows, the catcher switches to a predictive strategy because he does not have enough time to get more observations.

When α is big, the catcher believes that he has an unreliable model; therefore, he puts more weight on observations. In this case, for small β, the strategy is purely reactive; the catcher compensates for the absence of a good model by extensively relying on observations. With growing β, the catcher is forced to switch to a predictive strategy, which usually means he is going to perform poorly due to the lack of a good model.

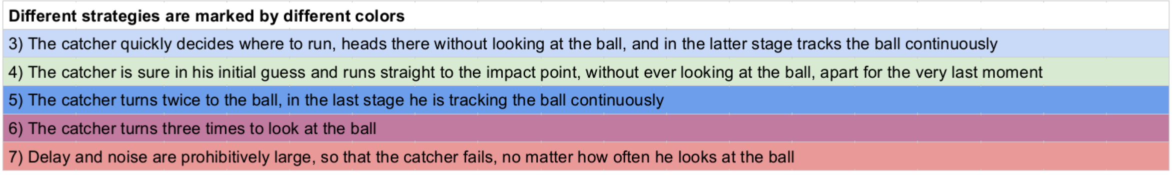
3.2.1 Experimental setup

The ball is thrown from the origin and lands at the point (30, 15, 0) after 3 seconds. The catcher starts at rest at point (20, 5) looking towards the origin. All parameters apart from the reaction delay and system noise are kept fixed. In particular, task duration and observation noise are kept fixed. For every reaction delay we document two quantities: number of turns the catcher makes and the fraction of time the catcher is looking at the ball. Failed attempts are marked with letter 'f'. Results are averaged over 10 trials. Figure 3.9 shows the averaged data and figure 3.10 gives a description of different catching strategies that emerge under different choices of parameters.



System noise / Observation noise		Reaction delay / Task duration																	
		small delays						medium delays				big delays							
		3 %		7 %		10 %		13 %		17 %		20 %		23 %		26 %			
		# turns	t / T	# turns	t / T	# turns	t / T	# turns	t / T	# turns	t / T	# turns	t / T	# turns	t / T	# turns	t / T		
small	0.1	1	0.47	1	0.42	1	0.37	1	0.27	1	0.23	0	0.00	0 f	0.00	0 f	0.00		
	1.0	1	0.54	1	0.51	1	0.43	1	0.40	1	0.23	0	0.00	0 f	0.00	0 f	0.00		
medium	5.0	2	0.36	2	0.39	2	0.34	1	0.42	1	0.26	1 f	0.10	1 f	0.21	1 f	0.10		
	10	2	0.41	2	0.42	2	0.38	2	0.40	2	0.33	1 f	0.06	1 f	0.30	2 f	0.17		
high	50	3	0.50	3	0.57	3	0.31	2 f	0.28	2 f	0.22	2 f	0.40	2 f	0.31	1 f	0.37		

Figure 3.9: Effects of noise and reaction time on the catcher's interception strategy. Top right: purely open-loop strategies. Bottom left: mostly feedback strategies. Otherwise, the strategy is mixed. Bottom right: failed catches.



Different strategies are marked by different colors

3) The catcher quickly decides where to run, heads there without looking at the ball, and in the latter stage tracks the ball continuously

4) The catcher is sure in his initial guess and runs straight to the impact point, without ever looking at the ball, apart for the very last moment

5) The catcher turns twice to the ball, in the last stage he is tracking the ball continuously

6) The catcher turns three times to look at the ball

7) Delay and noise are prohibitively large, so that the catcher fails, no matter how often he looks at the ball

Figure 3.10: Legend for figure 3.9. Different strategies are color coded.

Chapter 4

Conclusions

In this work, we have shown that observed human behavior in tasks such as catching a baseball can be explained by assuming that the catcher has an *internal model* of the environment and himself and acts *optimally* (maximizing the probability of catching the ball) under biological *constraints* (finite maximum velocity, running backwards is slow) and *imperfect knowledge* of the world (internal models only approximate reality). We have demonstrated in simulation experiments that the optimization-based approach is consistent with the heuristic view, a competing theory of ball catching, when heuristics are applicable (the catcher keeps the ball in the field of view all the time). The optimization-based approach has been shown to be more general than the heuristic view, as it can explain catches to which heuristics do not apply (e.g., when the ball flies over the catcher's head). Experiments with human subjects are required to confirm that such strategies are indeed used by real players.

Lastly, we have shown that the belief-space-based modeling approach allows one to explain how humans switch between *reactive and predictive catching strategies.* The choice of a strategy is defined by two parameters: the quality of the internal model and the task duration. If the model is excellent and the task duration is short, the catcher uses a predictive strategy; if the model is bad and the task duration is long, the catcher uses a reactive strategy; if the model is good and the task duration is long, the catcher uses a mixture strategy.

Appendix A

Derivations

A.1 Three types of uncertainty

In this section, we analyze uncertainties associated with predicting a future state of a stochastic system. We first derive the classical Kalman filter, revealing the meaning of *system uncertainty* and *a posteriori* uncertainty, and then show how to obtain the *a priori uncertainty* by integrating out future observations. Finally, we prove that the sum of the a priory and a posteriori uncertainties equals the system uncertainty.

A.1.1 Kalman filter

Let us begin by linearizing the dynamics and observation function:

$$x_k = A_{k-1} x_{k-1} + B_{k-1} u_{k-1} + \epsilon_k, \tag{A.1}$$

$$z_k = C_k x_k + \delta_k. \tag{A.2}$$

Figure A.1 shows the corresponding graphical model.

Assume the previous state is Gaussian distributed

$$p(x_{k-1}) = \mathcal{N}(x_{k-1} \,|\, \hat{x}_{k-1},\, \Sigma_{k-1}). \tag{A.3}$$

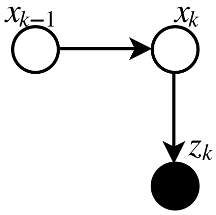


Figure A.1: One step of the Kalman filter.

Dynamics (A.1) can be rewritten as the conditional distribution

$$p(x_k \,|\, x_{k-1}) = \mathcal{N}(x_k \,|\, A_{k-1}x_{k-1} + B_{k-1}u_{k-1},\ Q_{k-1}). \tag{A.4}$$

From (A.3) and (A.4) we can integrate x_{k-1} out to obtain

$$p(x_k) = \mathcal{N}(x_k \,|\, A_{k-1}\hat{x}_{k-1} + B_{k-1}u_{k-1},\ A_{k-1}\Sigma_{k-1}A_{k-1}^T + Q_{k-1}). \tag{A.5}$$

Distribution (A.5) corresponds to the prediction step of the Kalman filter. We introduce shortcut notation for the predicted mean and covariance $p(x_k) = \mathcal{N}(x_k \,|\, \bar{x}_k,\ \bar{\Sigma}_k)$. Covariance matrix $\bar{\Sigma}_k$ gives the *system uncertainty*; it does not depend on observations. The next step is to incorporate observation z_k. Equation (A.2) is equivalent to the conditional distribution

$$p(z_k \,|\, x_k) = \mathcal{N}(z_k \,|\, C_k x_k,\ R_k). \tag{A.6}$$

Using it together with A.5 we can invert the conditioning:

$$p(x_k \,|\, z_k) = \mathcal{N}(x_k \,|\, \bar{x}_k + L_k(z_k - C_k\bar{x}_k),\ \bar{\Sigma}_k - L_k C_k \bar{\Sigma}_k), \tag{A.7}$$

where

$$L_k = \bar{\Sigma}_k C_k^T S_k^{-1}, \tag{A.8}$$

$$S_k = C_k \bar{\Sigma}_k C_k^T + R_k \tag{A.9}$$

Conditional distribution (A.6) corresponds to the correction step of the Kalman filter. In shortcut notation $p(x_k \,|\, z_k) = \mathcal{N}(x_k \,|\, \hat{x}_k,\ \Sigma_k)$. Covariance matrix Σ_k gives the *a posteriori* uncertainty; it is the uncertainty after observing z_k.

A.1.2 A priori uncertainty

During planning, future observations are not available, which increases the uncertainty about the future states. Therefore, future observations should not be considered fixed, as was done in (A.6), but should rather be treated as random variables. The corresponding graphical model is shown in figure A.2. For the Kalman filter, $\hat{x}_k$ is a number, because z_k is considered to be a fixed value; If, however, z_k is a random variable, then $\hat{x}_k$ also becomes a random variable, a Gaussian with a mean and a variance, which can be thought of as hyperparameters of the mean state.

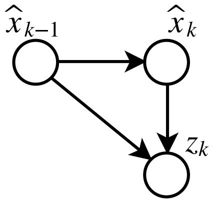


Figure A.2: Mean states $\hat{x}_k$ become random variables.

Let the previous mean be Gaussian distributed:

$$p(\hat{x}_{k-1}) = \mathcal{N}(\hat{x}_{k-1} \,|\, \mu_{k-1},\, \Lambda_{k-1}). \tag{A.10}$$

The conditional $p(\hat{x}_k \,|\, \hat{x}_{k-1})$ can be found by integrating out the observation z_k:

$$p(\hat{x}_k \,|\, \hat{x}_{k-1}) = \int p(\hat{x}_k \,|\, \hat{x}_{k-1},\, z_k)\, p(z_k \,|\, \hat{x}_{k-1})\, dz_k, \tag{A.11}$$

where

$$p(\hat{x}_k \,|\, \hat{x}_{k-1},\, z_k) = \delta(\hat{x}_k - (I - L_k C_k)\hat{x}_k - L_k z_k), \tag{A.12}$$

$$p(z_k \,|\, \hat{x}_{k-1}) = \mathcal{N}(z_k \,|\, C_k \bar{x}_k,\, S_k). \tag{A.13}$$

Thus, we get the dynamics of the mean state:

$$p(\hat{x}_k \,|\, \hat{x}_{k-1}) = \mathcal{N}(\hat{x}_k \,|\, \bar{x}_k,\, L_k C_k \bar{\Sigma}_k). \tag{A.14}$$

Finally, we integrate $\hat{x}_{k-1}$ out to obtain the update equations for the hyperparameters:

$$p(\hat{x}_k) = \mathcal{N}(\hat{x}_k \,|\, A_{k-1}\mu_{k-1} + B_{k-1}u_{k-1},\; A_{k-1}\Lambda_{k-1}A_{k-1}^T + L_k C_k \bar{\Sigma}_k), \tag{A.15}$$

which, in short, is $p(\hat{x}_k) = \mathcal{N}(\hat{x}_k \,|\, \mu_k,\, \Lambda_k)$. Covariance matrix Λ_k gives the *a priori* uncertainty; it is computed before the observations arrive by integrating them out.

A.1.3 System uncertainty decomposition

Now, we can combine results of the two previous subsections to show that the system uncertainty is the sum of the a priori and a posteriori uncertainties. Figure A.3 shows the graphical model where instead of observations we have hyperparameters.

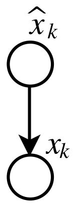


Figure A.3: Observations get absorbed into hyperparameters.

The conditional and marginal distributions are given respectively by:

$$p(x_k \,|\, \hat{x}_k) = \mathcal{N}(x_k \,|\, \bar{x}_k,\, \Sigma_k), \tag{A.16}$$

$$p(x_k) = \mathcal{N}(x_k \,|\, \mu_k,\, \Sigma_k + \Lambda_k). \tag{A.17}$$

Recall the equations for Σ_k and Λ_k:

$$\Sigma_k = \bar{\Sigma}_k - L_k C_k \bar{\Sigma}_k, \tag{A.18}$$

$$\Lambda_k = A_{k-1}\Lambda_{k-1}A_{k-1}^T + L_k C_k \bar{\Sigma}_k. \tag{A.19}$$

Their sum gives:

$$\Sigma_k + \Lambda_k = \bar{\Sigma}_k + A_{k-1}\Lambda_{k-1}A_{k-1}^T. \tag{A.20}$$

During planning, initial mean is fixed, i.e., $\Lambda_0 = 0$, leading to $\Sigma_1 + \Lambda_1 = \bar{\Sigma}_1$. If it holds for $k-1$, then it also holds for k:

$$\begin{aligned}
\Sigma_k + \Lambda_k &= \bar{\Sigma}_k + A_{k-1}\Lambda_{k-1}A_{k-1}^T && \text{(A.21)}\\
&= A_{k-1}\Sigma_{k-1}A_{k-1}^T + Q_{k-1} + A_{k-1}\Lambda_{k-1}A_{k-1}^T && \text{(A.22)}\\
&= A_{k-1}\Sigma_{k-1}A_{k-1}^T + Q_{k-1} + A_{k-1}(\bar{\Sigma}_{k-1} - \Sigma_{k-1})A_{k-1}^T && \text{(A.23)}\\
&= Q_{k-1} + A_{k-1}\bar{\Sigma}_{k-1}A_{k-1}^T && \text{(A.24)}\\
&= \bar{\Sigma}_k. && \text{(A.25)}
\end{aligned}$$

Thus, it is shown by induction that $\Sigma_k + \Lambda_k = \bar{\Sigma}_k$ for any k.

A.2 Probability of success as the objective function

In this section, we define an event *successful catch*, analyze its probability and show how to maximize it. We first consider 1D case, then generalize it to higher dimensions, and, finally, discuss how maximization of probability can be encoded within the model predictive control framework. Notation in this section slightly differs from the rest of the thesis to simplify derivations.

A.2.1 1D case

Let's consider the simplest case of a 1D ball and a 1D catcher and focus our attention on the final time-step. The only stochastic variable is the position b of the ball:

$$p(b) = \mathcal{N}(b \,|\, \mu, \Sigma). \tag{A.26}$$

Let the position c of the catcher be known with certainty. A catch is *successful*, if distance $r = b - c$ between the ball and the catcher is smaller than threshold α. The distance r is normally distributed $p(r) = \mathcal{N}(r \,|\, \mu - c,\, \Sigma)$, which allows us to find the *probability of success* $\Pr(|r| \leq \alpha)$ by integrating the probability density:

$$J(c,\, \Sigma) = \Pr(|r| \leq \alpha) = \int\limits_{-\alpha}^{\alpha} \mathcal{N}(r \,|\, \mu - c,\, \Sigma)\, dr. \tag{A.27}$$

Optimal catcher position

The goal of the catcher is to maximize the probability of success $J(c, \Sigma)$. We differentiate (A.27) w.r.t. parameters in order to find the maximum:

$$\begin{aligned}
\frac{\partial J}{\partial c} &= \int\limits_{-\alpha}^{\alpha} \frac{1}{\sqrt{2\pi\Sigma}}\, \mathrm{e}^{-\frac{1}{2}\frac{(r-(\mu-c))^2}{\Sigma}} \left(-\frac{1}{2} \cdot 2 \cdot \frac{r - (\mu - c)}{\Sigma} \right) dr \\
&= \frac{1}{\sqrt{2\pi\Sigma}} \int\limits_{-\alpha-(\mu-c)}^{\alpha-(\mu-c)} \mathrm{e}^{-\frac{1}{2}\frac{\xi^2}{\Sigma}}\, d\left(-\frac{1}{2}\frac{\xi^2}{\Sigma} \right) \\
&= \frac{1}{\sqrt{2\pi\Sigma}} \left[\mathrm{e}^{-\frac{1}{2}\frac{(\alpha-(\mu-c))^2}{\Sigma}} - \mathrm{e}^{-\frac{1}{2}\frac{(\alpha+(\mu-c))^2}{\Sigma}} \right].
\end{aligned} \tag{A.28}$$

The derivative vanishes when the expression in brackets in (A.28) equals zero, which happens if and only if $\mu = c$. The objective function (A.27) is maximized if the catcher happens to be at the most likely ball landing point, i.e., $c = \mu$.

Optimal level of uncertainty

Let us now find the optimal variance. We use the fact that $\mu = c$ at the optimum when substituting integration limits in the following formulas:

$$\begin{aligned}\frac{\partial J}{\partial \Sigma} &= \int\limits_{-\alpha}^{\alpha} \frac{1}{\sqrt{2\pi\Sigma}} \mathrm{e}^{-\frac{1}{2}\frac{\xi^2}{\Sigma}} \frac{1}{2}\left(\Sigma - \xi^2\right)\, d\xi \\ &= \frac{\alpha\sqrt{\Sigma}}{\sqrt{2\pi}} \mathrm{e}^{-\frac{1}{2}\frac{\alpha^2}{\Sigma}}. \end{aligned} \tag{A.29}$$

When $\Sigma \to 0+$, the derivative (A.29) approaches zero. The cost function (A.27) is maximized when uncertainty about the ball position is minimized, i.e., $\Sigma \to 0+$.

A.2.2 2D case

We consider the problem in the coordinate system attached to the catcher $c = 0$; the catcher controls μ indirectly by moving himself together with the coordinate system; we align the x-axis with vector μ to ease analysis. The function to be maximized is given by:

$$J(\mu, \Sigma) = \iint\limits_{\Omega} \frac{1}{\sqrt{(2\pi)^2 \det \Sigma}} \mathrm{e}^{-\frac{1}{2}(b-\mu)^T \Sigma^{-1}(b-\mu)}\, dx\, dy, \tag{A.30}$$

where $\Omega = \{b \mid |b| \leq \alpha\}$ is a disc of radius α centered at the origin.

In order to simplify analysis, we assume the ball coordinate b to be isotropic Gaussian, i.e., $\Sigma = \mathrm{diag}\{\sigma^2, \sigma^2\}$. It allows us to rewrite (A.30) in more detail:

$$J(\mu, \sigma^2) = \iint\limits_{\Omega} \frac{1}{2\pi\sigma^2} \mathrm{e}^{-\frac{(x-\mu)^2+y^2}{2\sigma^2}}\, dx\, dy \tag{A.31}$$

Tail integrals over circular areas of the Gaussian distribution cannot be computed in closed form; they are given by the Marcum Q-function. However, if we replace disk Ω

by square $G = \{b \mid |b_i| \leq \alpha, i = x, y\}$, integral (A.31) becomes separable:

$$J(\mu, \sigma^2) = \int_{-\alpha}^{\alpha} \frac{1}{\sqrt{2\pi\sigma^2}} \mathrm{e}^{-\frac{(x-\mu)^2}{2\sigma^2}}\, dx \int_{-\alpha}^{\alpha} \frac{1}{\sqrt{2\pi\sigma^2}} \mathrm{e}^{-\frac{y^2}{2\sigma^2}}\, dy \tag{A.32}$$

This reduces the 2D problem to two 1D problems which we have solved already; higher-dimensional cases can be treated in the same way.

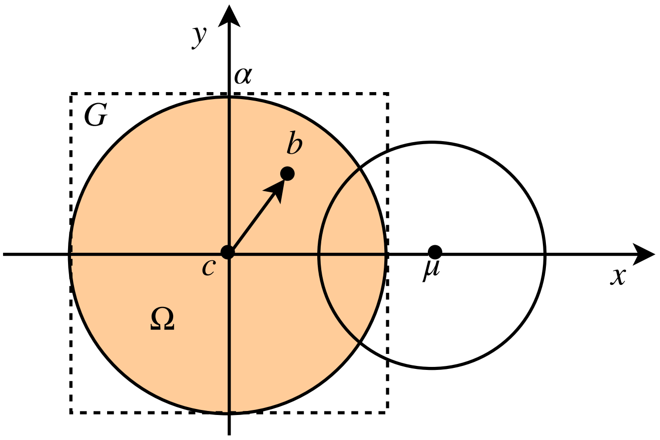


Figure A.4: The isotropic Gaussian probability density function centered at μ is to be integrated over the orange disc Ω of radius α. In order to make the integral tractable, we integrate over the square G instead; it gives an upper bound for the true probability because we increase the integration area.

A.2.3 Model predictive control objective

We have shown that the probability of success is maximized if the uncertainty of the ball position is minimized and the catcher is located at the mean predicted ball position at the final time. The catcher has to minimize the system uncertainty, because it is what eventually defines the spread of the ball positions. The system uncertainty can only grow with time; therefore, the only factor that can shrink the final system uncertainty is the initial system uncertainty, which, in its turn, equals the a posteriori uncertainty from the previous planning iteration. Thus, the a posteriori uncertainty has to be reduced in every iteration in order to reduce the final system uncertainty overall. This reasoning leads to the cost function (2.16), which penalizes the a posteriori uncertainty.

About the author

Prior to studying Communications and Multimedia Engineering at the University of Erlangen-Nuremberg, Boris Belousov graduated with distinctions from the Moscow Institute of Physics and Technology in year 2013 with a bachelor's degree in applied mathematics and physics. Studies in Germany have been generously supported by the German Academic Exchange Service (DAAD).

List of Figures

Bibliography

[1] J. Andersson, J. Akesson, and M. Diehl. Dynamic optimization with casadi. *Proceedings of the IEEE Conference on Decision and Control*, pages 681–686, 2012.

[2] L. T. Biegler and V. M. Zavala. Large-scale nonlinear programming using ipopt: An integrating framework for enterprise-wide dynamic optimization. *Computers and Chemical Engineering*, 33(3):575–582, 2009.

[3] A. Bry and N. Roy. Rapidly-exploring random belief trees for motion planning under uncertainty. *Proceedings - IEEE International Conference on Robotics and Automation*, pages 723–730, 2011.

[4] S. Chapman. Catching a baseball. *American Journal of Physics*, 36(10):868, 1968.

[5] R. Dawkins. *The Selfish Gene*. Oxford University Press, 1976.

[6] D. Genetics. World's fastest man, 2013.

[7] G. Gigerenzer and H. Brighton. Homo heuristicus: Why biased minds make better inferences. *Topics in Cognitive Science*, 1(1):107–143, 2009.

[8] M. M. Hayhoe, N. Mennie, K. Gorgos, J. Semrau, and B. Sullivan. The role of prediction in catching balls. *Journal of Vision*, 4(8):156–156, 2004.

[9] R. J. Kosinski. A literature review on reaction time, 2010.

[10] A. Mattmann, J. Peters, and G. Neumann. *Modeling How To Catch Flying Objects: Optimality Vs. Heuristics*. PhD thesis, 2014.

[11] D. Mayne, J. Rawlings, C. Rao, and P. Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, 36(6):789–814, 2000.

[12] M. McBeath, D. Shaffer, and M. Kaiser. How baseball outfielders determine where to run to catch fly balls. *Science*, 268(5210):569–573, 1995.

[13] J. McIntyre, M. Zago, a. Berthoz, and F. Lacquaniti. Does the brain model newton's laws? *Nature neuroscience*, 4(7):693–694, 2001.

[14] P. McLeod, N. Reed, and Z. Dienes. The generalized optic acceleration cancellation theory of catching. *Journal of experimental psychology. Human perception and performance*, 32(1):139–48, 2006.

[15] S. Patil, G. Kahn, M. Laskey, and J. Schulman. Scaling up gaussian belief space planning through covariance-free trajectory optimization and automatic differentiation. *Algorithmic Foundations of Robotics XI*, pages 515–533, 2015.

[16] R. Platt, R. Tedrake, L. Kaelbling, and T. Lozano-Perez. Belief space planning assuming maximum likelihood observations. *Robotics: Science and Systems*, 2010.

[17] R.-K. Saxonia. World records for backwards running, 2015.

[18] H. A. Simon. *Models of Man: Social and Rational.* 1957.

[19] S. Thrun. Probabilistic robotics. *Communications of the ACM*, 45(3):1999–2000, 2002.

[20] A. Tversky and D. Kahneman. Judgment under uncertainty: Heuristics and biases. *Science (New York, N.Y.)*, 185(4157):1124–1131, 1974.