Catching a Baseball: A Reinforcement Learning Perspective using a Neural Network

Rajarshi Das
Santa Fe Institute
1660 Old Pecos Trail, Suite A
Santa Fe, NM 87501

Sreerupa Das
Department of Computer Science
University of Colorado
Boulder, CO 80309-0430

Abstract

Moments after a baseball batter has hit a fly ball, an outfielder has to decide whether to run forward or backward to catch the ball. Judging a fly ball is a difficult task, especially when the fielder is in the plane of the ball's trajectory. There exists several alternative hypotheses in the literature which identify different perceptual features available to the fielder that may provide useful cues as to the location of the ball's landing point. A recent study in experimental psychology suggests that to intercept the ball, the fielder has to run such that the double derivative of $tan\phi$ with respect to time is close to zero (i.e. $d^2(tan\phi)/dt^2 \approx 0$), where ϕ is the elevation angle of the ball from the fielder's perspective (McLeod & Dlenes 1993). We investigate whether $d^2(tan\phi)/dt^2$ information is a useful cue to *learn* this task in the Adaptive Heuristic Critic ($\mathcal{AHC}$) reinforcement learning framework. Our results provide supporting evidence that $d^2(tan\phi)/dt^2$ information furnishes strong initial cue in determining the landing point of the ball and plays a key role in the learning process. However our simulations show that during later stages of the ball's flight, yet another perceptual feature, the perpendicular velocity of the ball (v_p) with respect to the fielder, provides stronger cues as to the location of the landing point. The trained network generalized to novel circumstances and also exhibited some of the behaviors recorded by experimental psychologists on human data. We believe that much can be gained by using reinforcement learning approaches to learn common physical tasks, and similarly motivated work could stimulate useful interdisciplinary research on the subject.

Introduction

Scientists have often wondered how an outfielder in the game of baseball or cricket can judge a fly ball by running either forward or backward and arriving at the right point at the right time to catch the ball (Bush 1967, Chapman 1969, Todd 1981). When the ball is coming directly at the fielder, the ball appears to rise or fall in a vertical plane, and thus the fielder has information about elevation angle of the ball and its rate of change. In the more typical case, when the ball is hit to the side, the fielder gets a perspective view of the trajectory of the ball

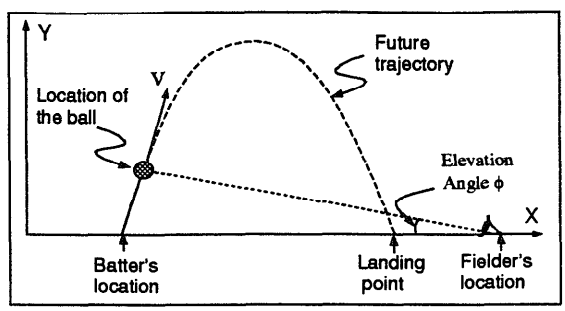

Figure 1: The fielder has to run and intercept the ball at the end of the ball's flight.

and there is additional information about azimuth angle and its rate of change. Hence, judging a fly ball is usually the most difficult when the fielder is in the plane of the ball's motion (Figure 1). Yet, moments after a batter hits the ball directly towards a fielder, the fielder has to decide if it is a short pop up in front, or a high fly ball over the fielder's head, and run accordingly. Thus, there is an important temporal credit assignment problem in judging a fly ball, since the success or failure signal is obtained long after the actions that lead to that signal are taken.

Considerable work in experimental psychology has focused on identifying the perceptual features that a fielder uses to judge a fly ball (Rosenberg 1988, Todd 1981). Several alternative hypothesis, as to the perceptual features that are important in making the decisions, have been postulated. In this paper, we explore the problem in detail using a reinforcement learning model. Our experimental results support one recent hypothesis that postulates the use of a specific trigonometric feature as an initial cue to determine the eventual landing point. However, in our reinforcement learning model this trigonometric feature by itself is not sufficient to learn the task successfully. We investigate other perceptual features which used in tandem with the trigonometric feature help the reinforcement learning system to successfully learn to catch fly balls. In trying to solve similar commonplace physical tasks

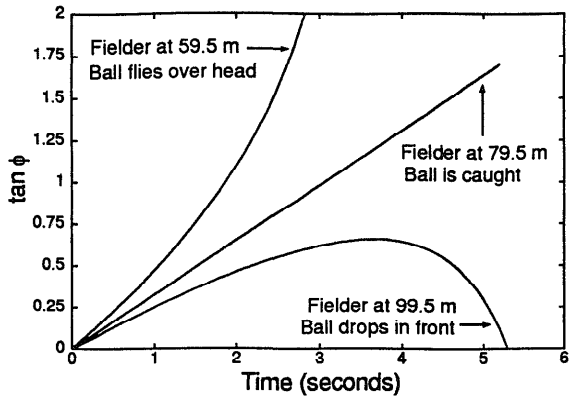

Figure 2: The figure shows the variation in $tan\phi$ as seen by three different fielders standing at 59.5 m, 79.5 m, and 99.5 m from the batter. The initial velocity of the ball is 30 m/s, directed at an angle 60° from the horizontal. Here the range of the trajectory of the ball is 79.5 m, and since this simulation ignores air resistance, $tan\phi$ increases at a constant rate only for the fielder standing at 79.5 m.

using reinforcement learning we not only learn more about the reinforcement learning models themselves but also understand the underlying complexities involved in a physical task.

The physics of judging a fly ball

The problem of trajectory interception was analyzed by Chapman using Newton's laws of motion (Chapman 1968). For a perfect parabolic trajectory, the tangent of the ball's elevation angle ϕ increases at a steady rate with time (i.e. $d(tan\phi)/dt = constant$) over the entire duration of flight, if the fielder stands stationary at the ball's landing point (Figure 2). This simple principle holds true for any initial velocity and launch angle of the ball over a finite range. If the ball is going to fall in front of the fielder, then $tan\phi$ grows at first and then decreases with $d^2(tan\phi)/dt^2 < 0$. On the other hand, if the ball is going to fly over the fielder's head, then $tan\phi$ grows at an increasing rate with $d^2(tan\phi)/dt^2 > 0$. Chapman suggested that if a fielder runs with a constant velocity so that $d(tan\phi)/dt$ is constant then the fielder can reach the proper spot to catch the ball just as it arrives.

However, Chapman neglected the effects of aerodynamic drag on the ball which significantly affects the ball's trajectory and range. When air resistance is taken into account, Brancazio claimed that the specific trigonometric feature cited by Chapman cannot provide useful cues to the fielder (Brancazio 1985). In addition, Chapman's hypothesis makes the unrealistic assumption that the fielder runs with a constant velocity while attempting to catch a fly ball. Brancazio went on to show that many of the other perceptual features available to a fielder (see Table 1) *cannot* provide significant initial cue to determine the landing point of the ball. After eliminating several possible candidate features, Brancazio hypothesized that the angular acceleration of the ball $d^2\phi/dt^2$ provides the strongest initial cue as to the location of the eventual landing point. He also conjectured that the angular acceleration of a fielder's head while the fielder tries to visually track a fly ball, might be detected by the vestibular system in the inner ear, which in turn might provide feedback to influence the judgement process of the fielder.

Brancazio's List of Perceptual Features Available to the Fielder	
Symbol	Feature
ϕ	Angle of Elevation
$d\phi/dt$	Rate of change of ϕ
$d^2\phi/dt^2$	Rate of change of $d\phi/dt$
D	Distance between ball and fielder
dD/dt	Rate of change of D (= $-v_r$, the radial velocity)
v_p	Velocity of ball perpendicular to fielder
dv_p/dt	Rate of change of v_p

Table 1: Brancazio showed that, with the possible exception of $d^2\phi/dt^2$, these features provide no significant initial cue as to the location of the ball's landing point. Note that D is inversely proportional to the apparent size of the ball. Other possible perceptual features include $tan\phi$, $d(tan\phi)/dt$, $d^2(tan\phi)/dt^2$.

Recent experimental results obtained by McLeod and Dlenes (McLeod & Dlenes 1993) however show that an experienced fielder runs such that $d^2(tan\phi)/dt^2$ is maintained close to zero until the end of the ball's flight. McLeod and Dlenes suggest that this is a very robust strategy for the real world, since the outcome is independent of the effects of aerodynamic drag on the ball's trajectory, or the ball following a parabolic path. However little is understood about how human beings *learn* to intercept a free falling ball (Rosenberg 1988) and exactly how $d^2(tan\phi)/dt^2$ information helps in the learning process.

In this paper, we provide supporting evidence that $d^2(tan\phi)/dt^2$ information furnishes strong initial cue as to the landing point of the ball and plays a key role in the learning process in a reinforcement learning framework. However, in the later stages of the ball's flight, $d^2(tan\phi)/dt^2$ provides conflicting cues and the reinforcement learning model has difficulty in intercepting fly balls. We delineate the cause of this problem and use an additional perceptual feature that helps in learning the task.

Using reinforcement learning to catch a baseball

We use Barto, Sutton and Anderson's Multilayer Adaptive Heuristic Critic ($\mathcal{AHC}$) model (Anderson 1986) to learn the task. The general framework of reinforcement learning is as follows: an agent seeks to control a discrete time stochastic dynamical system. At each time step, the agent observes the current environmental state x and executes action a. The agent receives a payoff (and/or pays a cost) which is a function of state x and action a, and the system makes a probabilistic transition to state y. The agent's goal is to determine a control policy that maximizes some objective function. $\mathcal{AHC}$ is a reinforcement algorithm for discovering an extended plan of actions which maximizes the cumulative long-term reward received by an agent as a result of its actions.

In the $\mathcal{AHC}$ framework, the model consists of two sub-modules (networks); one is the agent (action network), that tries to learn search heuristics in the form of a probabilistic mapping from the states to the actions in order to maximize the objective function. Typically the objective function is a cumulative measure of payoffs and costs over time. The other module is the critic (evaluation network) that tries to evaluate the agent's performance based on the reinforcement received from the environment as a result of the action just taken.

In our implementation of the $\mathcal{AHC}$ model, the action $a(t)$, taken by the agent (action network) corresponds to the instantaneous acceleration of the fielder at time t. The state, x, is assumed to be described by a set of inputs provided to the model at every time step. The action network generates real valued actions, $a(t)$, at every time step, similar to that described by Gullapalli (Gullapalli 1993). The output of the action network determines the mean, $\mu(t)$, and the output of the evaluation network determines the standard deviation, $\sigma(t)$ of the acceleration, $a(t)$, at a particular time.

$$\mu(t) = \text{output of action network},$$

$$\sigma(t) = max(r(t), 0.0)$$

where $r(t)$ is the output of the evaluation network. Assuming a Gaussian distribution Ψ, the action $a(t)$ is computed using $\mu(t)$ and $\sigma(t)$.

$$a(t) \sim \Psi(\mu(t), \sigma(t))$$

In the course of learning, both the evaluation and action networks are adjusted incrementally in order to perform credit assignment appropriately. The most popular and best-understood approach to a credit assignment problem is the *temporal difference* (TD) method (Sutton 1988), and the $\mathcal{AHC}$ is a TD based reinforcement learning approach (Anderson 1986).

Since the objective of learning is to maximize the agent's performance, a natural measure of performance is the *discounted cumulative reinforcement* (or for short, *utility*) (Barto et al. 1990):

$$r(t) = \sum_{k=0}^{\infty} \gamma^k f(t+k)$$

where $r(t)$ is the discounted cumulative reinforcement (utility) starting from time t throughout the future, $f(t)$ is the reinforcement received after the transition from time t to $t+1$, and $0 \leq \gamma \leq 1$ is a discount factor, which adjusts the importance of long term consequences of actions. Thus the utility, $r(t)$, of a state x is the immediate payoff plus the utility, $r(t+1)$, of the next state y, discounted by γ. Therefore the desired function must satisfy:

$$r(t) = f(t) + \gamma r(t+1)$$

Relating these ideas to the $\mathcal{AHC}$ model, the output of the evaluation network corresponds to $r(t)$. During learning, the evaluation network tries to generate the correct utility of a state. The difference between the actual utility of a state and its predicted utility (called the TD error) is used to adjust the weights of the evaluation network using backpropagation algorithm (Rumelhart et al. 1986). The action network is also adjusted according to the same TD error (Sutton 1988, Lin 1992). The objective function that determines the weight update rules is defined as:

$$Error = \begin{cases} f(t) + \gamma r(t+1) - r(t) & \bullet \text{ while the ball is in the air,} \\ f(t) - r(t) & \bullet \text{ if the ball has hit the ground.} \end{cases}$$

Simulation details

The perceptual features that are available to the fielder while judging a fly ball define the input variables of our system. At any time t, the inputs to the system include: ϕ, $d^2(tan\phi)/dt^2$, v_f—the velocity of the fielder, and a binary flag which indicates whether the ball is spatially in front of or behind the fielder. Thus the system receives no information about the absolute coordinates of the ball or the fielder at any point in time. Initially, the fielder is positioned at a random distance in front of or behind the ball's landing point. The initial velocity and the initial acceleration of the fielder are both set to zero. Once the ball is launched, the fielder's movement is controlled by the output $a(t)$ of the action network which determines the fielder's acceleration at time t. The simulation is continued (see Appendix for the equations) until the ball's trajectory is complete and a failure signal is generated. If the ball has hit the ground and the fielder has failed to intercept the ball, the failure signal $f(t)$ is proportional to the fielder's distance from the ball's landing point.

$$f(t) = \begin{cases} 0 & \text{while the ball is in the air,} \\ 0 & \text{if } D(final) \leq \mathcal{R} \text{ (Success!),} \\ -C\ |D(final)| & \text{if } D(final) > \mathcal{R} \text{ (Failure!).} \end{cases}$$

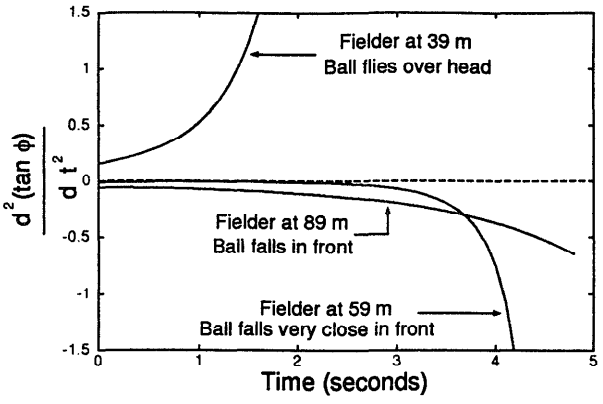

Figure 3: The variation of $d^2(tan\phi)/dt^2$ as seen from three different positions. Aerodynamic drag is taken into consideration in this simulation, and for the same initial parameters as in Figure 2, the range decreases to 57.5 m. The ball touches the ground at $t = 4.9$ second. Note that for the fielder stationed very close to the ball's landing point at 59 m, the $d^2(tan\phi)/dt^2$ is close to zero for most of the ball's flight, but it increases dramatically at the end.

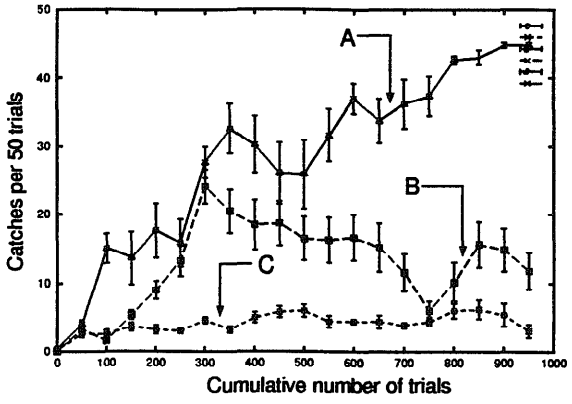

Figure 4: The plots show the number of successful catches every 50 trials as a function of total number of trials for three different sets of input features. The three sets of features are (A) both $d^2(tan\phi)/dt^2$ and v_p. (B) $d^2(tan\phi)/dt^2$ but not v_p, (C) $d^2(\phi)/dt^2$. (The other input features: ϕ, v_f and the binary flag were used in all three sets). The initial angle of the ball is chosen randomly between [50°, 70°]. The fielder's initial position is also chosen from a random distribution between [47.5m, 67.5m]. The initial velocity and initial acceleration of the fielder are both set to zero in every trial.

where $D(final)$ is the distance between the ball and the fielder when the ball hits the ground, $\mathcal{R}$ is the catching radius and C is a positive constant. In order to account for last moment adjustments made by the fielder (for example, making a final dive at the ball !), a catch is considered successful if the ball hits the ground within a region around the fielder's position defined by the catching radius, $\mathcal{R}$. In our simulations, the catching radius was set to 2 m. It may be noted here that the information–whether the ball fell in front of or behind the fielder–is not a part of the reinforcement signal. This information is provided as a part of the input signal and thus, all throughout the ball's trajectory, the fielder knows whether the ball is in front of or behind the fielder.

The inputs to the network are computed as follows. The raw inputs, as determined by the system dynamics (defined in the Appendix), are first clipped using the following lower and upper bounds (indicated by $\langle\rangle$): $\langle -10.0, 10.0\rangle m/s$ for the fielder's velocity, v_f; $\langle -5.0, 5.0\rangle m/s^2$ for the fielder's acceleration; $\langle 0°, 180°\rangle$ for ϕ; $\langle -25.0, 25.0\rangle m/s$ for v_p (referred to in the next section); $\langle -0.5, 0.5\rangle s^{-2}$ for $d^2(tan\phi)/dt^2$. The clipped inputs are then normalized between 0.0 and 1.0 and finally presented to the network. Nevertheless, while determining the system dynamics none of the values are either scaled or clipped. A sampling frequency of 10 Hz (i.e. $\Delta t = 0.1$s) is used during the simulation of the system.

Results

Our results, using the $\mathcal{AHC}$ learning approach, show that $d^2(tan\phi)/dt^2$ information by itself is *not* sufficient to learn the task at hand. After an initial learning period, the system surprisingly learns to move the fielder away from the ball's landing point instead of moving towards it. Figure 3 delineates the underlying problem. For a fielder standing at the ball's landing point, $d^2(tan\phi)/dt^2$ is always zero. However, if the fielder is only a small distance away from the ball's landing point, $d^2(tan\phi)/dt^2$ is close to zero for most of the ball's flight, until near the end when it increases dramatically. Thus large and small magnitudes of $d^2(tan\phi)/dt^2$ can be associated with both large and small values of negative failure signals providing conflicting cues to a learning system. We therefore investigate other perceptual features that might help in the learning process by removing the ambiguity.

Figure 4 plots the performance of the network when different sets of inputs (perceptual features) are used (in addition to ϕ, v_f, and the binary direction flag). In the figure, each learning curve is an average of 10 independent trials, where each curve corresponds to one of the three different sets of perceptual features (A) $d^2(tan\phi)/dt^2$ and v_p, where v_p is the perpendicular component of the ball's velocity as seen by the fielder, (B) $d^2(tan\phi)/dt^2$ (McLeod & Dlenes' hypothesis), and (C) $d^2(\phi)/dt^2$ (Brancazio's hypothesis). In the simulations each trial begins with the fielder at a random position in the range [47.5m, 67.5] in front of the ball and the ball is thrown with an initial angle randomly distributed in [50°, 70°]. The plots show that the network could not learn the task using only $d^2(tan\phi)/dt^2$ or using

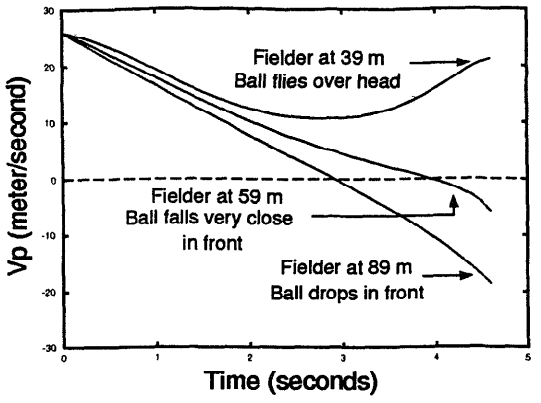

Figure 5: The variation of the perpendicular component of the ball's velocity as seen from three different positions. The initial parameters are the same as in Figure 2. The ball touches the ground at $t = 4.9$ seconds. Note that the three plots are very close to each other for the first three seconds, and diverge only at the end of the ball's flight.

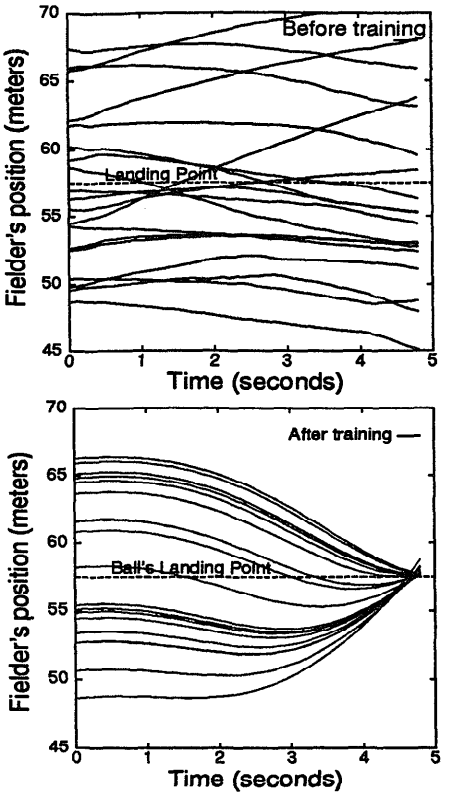

Figure 6: The two space-time plots show the fielder's distance from the batter in 20 trials, *before* (left) and *after* (right) training for 10000 trials. The initial parameters of the ball are the same as in Figure 2 and the fielder's initial position is chosen from a random distribution [47.5m, 67.5m]. The initial velocity and the initial acceleration of the fielder are both set to zero in every trial. The ball's range is 57.5 m which is reached at $t = 4.9$ second.

only $d^2(\phi)/dt^2$. Let us analyze why v_p could possibly help in learning (Brancazio 1985). Figure 5 plots the variation of v_p as seen by the fielder standing at three different positions. Initially, v_p provides little cue as to the balls landing point, but as the ball's flight comes to an end, v_p is significantly different for the fielders standing at different positions. Interestingly enough, the network is able to learn the task, since v_p information adds the necessary discriminating ability in judging fly balls during the latter stages of the ball's flight.

The above results suggests that in our reinforcement learning model both $d^2(tan\phi)/dt^2$ and v_p are necessary for learning the task of catching a ball. During the initial part of the ball's flight, the system learns to keep $d^2(tan\phi)/dt^2$ very small, and move in the correct direction. Towards the end of the ball's flight, when $d^2(tan\phi)/dt^2$ increases drastically, the system learns to use v_p to decide whether to run forward or backward.

Figure 6 shows space-time plots of the fielder's trajectories before and after training for 20 different trials (the initial positions of the fielder are set randomly, although the initial angle of the ball is identical in each trial). In their experiments with a skillful fielder, McLeod and Dlenes observed that the fielder does not automatically run to the point where the ball will fall and then wait for it, rather the fielder tracks the ball throughout its trajectory till it hits the ground. We see a similar behavior in Figure 6 after the system has learned to catch. More interestingly, Figure 6 shows that a fielder who is initially positioned slightly in front of the landing point of the ball, goes through a temporary phase when the fielder actually runs away from the eventual landing point of the ball. The data presented by McLeod and Dlenes shows surprisingly similar behavior among experienced fielders.

Our last set of simulations focus on the generalizational performance of a trained network. Figure 7 depicts the results. The network is first trained with trials where the initial angle of the ball is randomly set to a value in the range $[57^o, 62^o]$. After training, we test the network on trials where the initial angle is randomly selected from increasing ranges: from $[57^o, 62^o]$ to $[45^o, 75^o]$. As is evident from the plot, the network is able to generalize and perform reasonably well in situations which it had not experienced during the training phase. Note that the range of ball's trajectory during training is bounded between 56.1m (for 57^o) and 62.3m (for 62^o) which is much smaller than the range of the ball's trajectory during testing (which varies between 69.69m (for 45^o) and 34.36m (for 75^o)). These results in generalization performance indicate that the network is able to extract important rules from the perceptual features

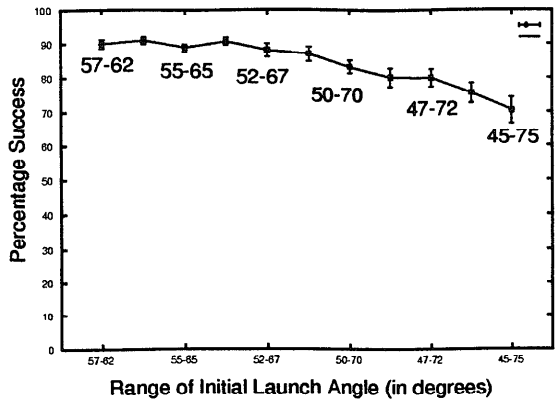

Figure 7: Average generalization performance of a trained network is shown. The network is trained on trajectories with the ball's initial angle ranging between $57° - 62°$. The trained network is then tested on trials where the initial angle ranged between $45° - 75°$. The simulations are averaged over 10 runs with the fielder's initial position chosen from a random distribution [47.5m, 67.5m].

rather than memorize the training data.

Conclusion and future work

The goal of this research is to determine if a reinforcement learning model can learn to catch fly balls using a specific trigonometric feature suggested in the experimental psychology literature. We have shown that for the reinforcement learning model discussed in this paper, $d^2(tan\phi)/dt^2$ and v_p information play a vital role in the learning the task. It is possible that in later stages of the ball's trajectory, an experienced fielder might use other perceptual features like stereoscopic vision as the guiding mechanism. We are currently investigating such a hypothesis. We believe much can be gained by using reinforcement learning approaches to learn common physical tasks, and we hope that this work would stimulate useful interdisciplinary research on the subject.

Acknowledgements

We thank C. W. Anderson, K. L. Markey, M.C. Mozer, S.J. Nowlan, and the anonymous reviewers of this paper for their valuable suggestions.

Appendix: The Equations of Motion

The equations of motion in two dimensions for a projectile can be expressed as:

$$x'' = -Kvv_x, \qquad y'' = -Kvv_y - g \qquad (1)$$

where x'' and y'' are the instantaneous horizontal and vertical accelerations, v_x and v_y are the horizontal and vertical components of the velocity of the ball v, g is the acceleration due to gravity and K is the aerodynamic drag force constant equal to $0.005249\ m^{-1}$ for a baseball (Brancazio 1985). Using a sampling time of Δt second, the above equations are numerically integrated using third derivatives as follows:

$$\Delta x = v_x\Delta t + 0.5x''(\Delta t)^2 + 0.1667x'''(\Delta t)^3, \quad (2)$$

$$\Delta y = v_y\Delta t + 0.5y''(\Delta t)^2 + 0.1667y'''(\Delta t)^3, \quad (3)$$

where $x''' = -K(v'v_x + vx'')$, $y''' = -K(v'v_y + vy'')$, and $v' = (v_xx'' + v_yy'')/v$. The velocity components are also updated as:

$$\Delta v_x = x''(\Delta t) + 0.5x'''(\Delta t)^2, \qquad (4)$$

$$\Delta v_y = x''(\Delta t) + 0.5y'''(\Delta t)^2 \qquad (5)$$

Given the current coordinates of the fielder and the ball, and their respective velocities, it is possible to calculate the variables associated with ϕ, and $tan\phi$ including their derivatives using trigonometric equations and calculus.

References

Anderson, C.W. 1986. Learning and Problem Solving with multilayer connectionist systems. Ph.D. diss., Computer Science, Univ. of Massachusetts, Amherst.

Barto, A.G., Sutton, R.S., & Watkins, C.J.C.H. 1990. "Learning and sequential decision making," In: M. Gabriel & J.W. Moores (Eds.), *Learning and computational neuroscience*, MIT Press.

Brancazio, P.J. 1985. "Looking into Chapman's homer: The physics of judging a fly ball," *American Journal of Physics*, Vol. 53, No. 9, pp. 849-855.

Bush, V. 1967. *Science is not enough*, Wm. Morrow Co., NY.

Chapman, C. 1968. "Catching a baseball," *American Journal of Physics*, Vol. 36, No. 10, pp. 868-870.

Gullapalli, V. 1990. "A stochastic reinforcement learning algorithm for learning real-valued functions," *Neural Networks*, Vol. 3, pp. 671-691.

Lin, L.J. 1992. "Self-improving reactive agents based on reinforcement learning, planning, and teaching," *Machine Learning*, 8, pp. 293-321.

McLeod, P. & Dienes, Z. 1993. "Running to catch the ball," *Nature*, Vol. 362, pp. 23.

Rosenberg, K.S. 1988. "Role of visual information in ball catching," *Journal of Motor Behavior*, Vol. 20, No. 2, pp. 150-164.

Rumelhart, D.E., Hinton, G.E., & William, R.J. 1986. "Learning internal representations by error propagation," *Parallel Distributed Processing: Explorations in the microstructure of cognition. Vol. 1.*, Bradford Books/MIT Press.

Sutton, R.S. 1988. "Learning to predict by the methods of temporal differences." *Machine Learning*, 3, pp. 9-44.

Todd, J.T. 1981. "Visual information about moving objects," *Journal of Experimental Psychology: Human Perception and Performance*, Vol. 7, No. 4, pp. 795-810.