Spatial Navigation Algorithms: Applications to Mobile Robotics

Thomas Sugar, Michael McBeath

Mechanical and Aerospace Engineering, Psychology
Arizona State University
Tempe, AZ 85283

Abstract

Human navigational principles for catching are researched and applied to mobile robotic applications. Two principles, constant optical rate and angular constancy, can be used by a fielder or mobile robot to define a trajectory for interception of a projectile without knowledge of the ball's or fielder's world coordinates. Two robust, viewer-centered algorithms are modeled which allow interception of three-dimensional, projectile trajectories.

1 Introduction

Interception of a projectile like a baseball is an interesting task for both humans and robots. Human behavior patterns may provide insight into new robotic control algorithms. Different spatial perceptual strategies for intercepting a projectile have been debated. We are focusing on perceptual strategies used for the initial phase of catching. Namely, how to determine a path so that the fielder converges to the destination of the projectile. The strategies assume that humans have a low level, unconscious, visual-action control algorithm that is reliable and robust [1].

One strategy, Optical Acceleration Cancellation (OAC) [2], describes a baseball fielder catching a fly ball by selecting a running path to achieve optical acceleration cancellation of the ball in the image plane or *constant optical rate in the camera's image plane.* For this strategy, fielders run along a path that is laterally aligned with the ball and maintain a constant change in the tangent of the vertical optical angle ($\tan(\alpha)$). See Figure 1. Thus, in order to catch the ball, a fielder needs to select a running path that maintains the change in $\tan(\alpha)$ equal to a constant ($\frac{d}{dt}(\tan(\alpha)) = c$). A drawback of this strategy is that the fielder must be perfectly aligned with the ball and spatial derivatives need to be calculated.

Another perceptual strategy proposed by McBeath et. al. [3] is to maintain a Linear Optical Trajectory (LOT). McBeath et. al. proposed that rather than alignment, the fielder maintains a monotonically increasing linear optical trajectory (LOT) of the ball relative to the home-plate and background scenery. The LOT model maintains the *optical projection angle, ψ, to be constant* so that there exists an imaginary linear line on the background scenery due to the projection image of the ball from the perspective of the fielder. See Figure 2. By doing so, the image of the ball always rises above the fielder guaranteeing that the fielder travels to the correct destination. The models and more details of these two strategies are discussed later in this paper. Advantages to this strategy are two-fold. First, the strategy describes a path for a projectile that can move in three dimensions. Second, the strategy uses only optical geometry to describe the fielder's path, thus converting the spatial-temporal problem into a simple geometric one.

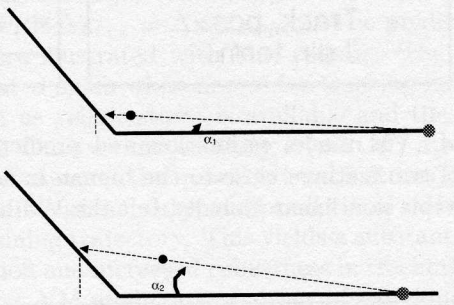

Figure 1: Fielders adjust their position so that the image of the ball rises at a constant rate. For parabolic ball trajectories, $\frac{d}{dt}\tan(\alpha)$ equals a constant.

In the paper, the OAC and LOT models are discussed followed by simulations to show the validity of the mathematical models of the two strategies. The simulations are based on either a constant optical rate in the image (OAC), or a constant angle (LOT). In this paper simulations for three dimensional trajectories are described.

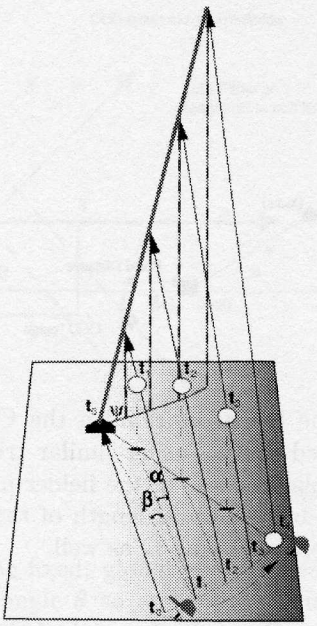

Figure 2: Fielders adjust their path to keep the image of the ball on the projection plane monotonically rising along the straight-line trajectory described by the angle, ψ.

2 Literature Review

Mathematically, the OAC model stipulates that the fielder selects a running path that is laterally aligned with the ball. He adjusts the approach velocity so that the change in the vertical optical angle from the ground plane is constant with respect to time, $(\frac{d}{dt}(\tan(\alpha)) = c)$ [2]. If the rate of change of $\tan(\alpha)$ increases, the fielder has run too far forward (or too slowly backwards), and the ball will land behind the fielder. Similarly, if the rate of change of $\tan(\alpha)$ decreases, the fielder has run too far backward (or forward too slowly) and the ball will fall in front. The rule to catch the ball is simple. In order to catch the ball, the fielder needs merely to select a running path so that the $\frac{d}{dt}(\tan(\alpha))$ is equal to a constant achieving an optical acceleration cancellation algorithm (OAC). The algorithm nulls any errors by keeping the image of the ball always constantly rising in the image plane.

According to McBeath et. al., the OAC model is elegant but has several weaknesses. First, the acceleration to be canceled is $\frac{d}{dt}(\tan(\alpha))$ not the vertical optical angle α itself. If $\frac{d}{dt}(\tan(\alpha)) = c$, then as the angle α increases, the derivative, $\frac{d}{dt}(\tan(\alpha))$, decreases. Therefore, the OAC model requires maintenance of an optical rate that is not constant, but based on the tangent function [3]. Second, OAC solutions require a precise ability to discriminate acceleration, and typical outfielders lack sufficient acceleration resolution to account for their accuracy in catching a fly ball.

McBeath et. al. proposed the Linear Optical Trajectory, or LOT model. The projection of the ball is analyzed as a unified two-dimensional (2-D) image in which two angles, α (vertical optical angle) and β (lateral optical angle), are introduced to define a linear optical trajectory, with a constant projection angle, ψ. In this model, the fielder selects a running path that maintains a monotonically rising linear optical trajectory (LOT) with the ball relative to home plate and the background scenery, in order to catch the ball. See Figure 2. The magnitude of the angle ψ is related to the angles α and β by the relationship $\tan\psi = \frac{tan\alpha}{tan\beta}$ when using projection plane geometry.

Edward Aboufadel, a mathematician, has proposed two models, a Strong LOT and a Regular LOT model [4]. The Strong LOT model stipulates that a fielder's running path is determined by the location of the ball (x, y, z) and he chooses image variables p and q, $(p = \frac{y_{image}}{x_{image}})$ and $(q = \frac{z_{image}}{x_{image}})$ to describe the trajectory. In the Strong LOT model, a fielder has to maintain p and q equal to respective constants [4]. For this model, there is only one unique path of the fielder for every ball trajectory. The Regular LOT model allows the image plane (or projection plane) to rotate and p and q can vary in time as long as $\tan\psi$ is maintained constant. Obviously, there is no longer a unique path, but the fielder still keeps the $\tan\psi$ equal to a constant in the moving image plane, and it can be shown that the fielder still arrives at the destination. In both LOT strategies, the linear projection angle of the ball to the background scenery has to be maintained. If the image of the ball in the image plane starts to decelerate vertically, the ball curves down toward the fielder's front horizon. If the ball starts to accelerate vertically, the ball curves up toward the zenith and arches past the fielder toward his back horizon. Thus, the LOT strategy promotes optical speed constancy cancellation by controlling the geometry of the projection of the ball.

Relevant research in robotics includes reactive based mobile robotic systems. In robotics, reactive systems or behavior-based models by Arkin and others [5] typically rely on sensory data and react to the environment to avoid collision or follow other mobile robots. Also in robotics, [6], Burgstadt and Ferrier have demonstrated a mobile robot system based on the OAC model. In their implementation, they collected acceleration data from the image, which is very noisy. In contrast, we have adapted the problem to use control variables in the visual plane simplifying the problem [7]. Others have been interested in vi-

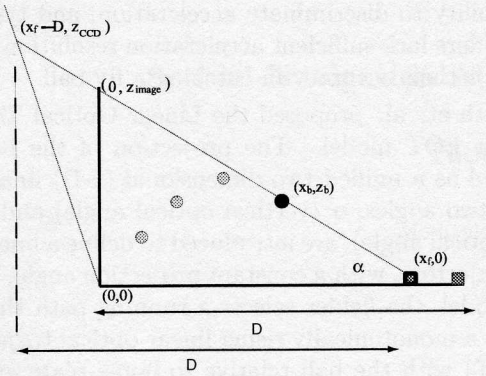

Figure 3: The desired height of the ball from the ground should rise at a constant rate. The actual height of the ball, z_{image}, varies depending on the robot's position. With a camera, the desired height of the ball, z_{ccd}, should rise at a constant rate. (Proof: Use a fixed base length, D.)

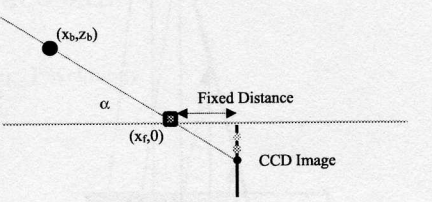

Figure 4: The ball's position on the CCD image is directly related to z_{ccd} using similar triangles. This CCD image plane moves as the fielder moves forward or backward, but the focal length of the camera is a fixed distance, thus fixing D as well.

sual control in the image plane such as Zhang and Ostrowski. [8].

Missile interception research is also relevant, but we are focusing on understanding the human motor-visual system. Sastry et al. are focusing on the mathematics of pursuit [9].

Along with reactive algorithms such as the OAC and LOT theories, map building algorithms are studied in both the perceptual and robotic literature. Typically, map building methods are based on a world-centered coordinate system, while reactive methods are based on a viewer-centered coordinate system. A path could be determined to pursue a target once the map of the environment is built. In these systems, a world map of the robot is known, and the robot is directed to the target location determined by curve fitting past image data. At MIT, a ball catching robot has been built, but it is not mobile and relies on fitting a curve to the past image data to determine the future location of the projectile [10]. Other researchers have suggested modeling the ball trajectory and determining the destination of the ball in world coordinates. Physicists note that complex iterative models are required to capture the influence of everyday variables like temperature, humidity, friction and projectile spin, thus making this strategy very difficult [11].

3 Modeling and Simulation

3.1 OAC Model

In contrast to previous work in robotics, our calculations are done in the image plane. See Figures 3 and 4. We develop the OAC algorithm for a mobile robot using a vision system. In the original OAC algorithm proposed by perceptual psychology, the ball must rise according to the $\tan(\alpha)$.

$$\frac{d}{dt}(\tan\alpha) = \frac{d}{dt}\left(\frac{z_b}{(x_f - x_b)}\right) = C \tag{1}$$

where $(x_f, 0)$ and (x_b, z_b) are the world coordinates of the fielder and ball positions. It can easily be seen that with perfect knowledge,

$$\tan\alpha = Ct \tag{2}$$

$$x_f = x_b + \frac{z_b}{Ct} \tag{3}$$

where

$$C = \frac{\dot{z}_b(0)}{D} \tag{4}$$

$$D = x_f(0) - x_b(0). \tag{5}$$

The equations for the mobile robot are defined in the CCD-image plane which moves forward and backward with the fielder's movement. In this *moving* plane, the height of the ball must rise at a constant rate.

$$z_{image} = x_f \tan(\alpha) = x_f Ct \tag{6}$$

$$z_{ccd} = D\tan(\alpha) = DCt. \tag{7}$$

In the *moving* image plane, the ball height will equal $D\tan(\alpha)$ and the change in the height of the ball in the image plane will equal $DC = \dot{z}_b(0)$. The fielder cannot know the initial upward velocity of the ball, but can estimate it after a fraction of a second. In our simulations, a period of 0.3s is used to estimate the upward velocity of the ball.

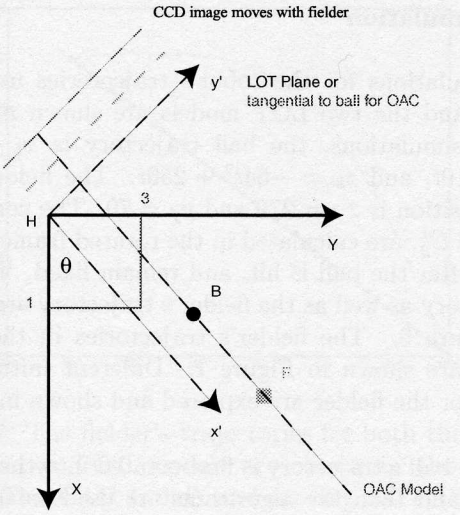

Figure 5: The fixed, global coordinate system is rotated by an angle θ to make the calculations much simpler. In the the new (x', y') coordinate system, the x' axis is aligned with the ball and the robot. Therefore, the 2D OAC model will be used in the x' direction. The robot will align itself with the ball in the y' direction.

The variable, a, will be the estimate of the initial upward velocity of the ball. This estimated variable is used in our experimental work with planar trajectories [7]. Note, it is much easier for a controller to keep the height of the ball moving upward by an amount DC in the CCD-image, than keeping $\frac{d}{dt}(\tan(\alpha))$ equal to a constant.

3.2 OAC Model, 3 Dimensions

In the original mathematical analysis by Aboufadel [4], the first base line and third base line defined the x and y axes for the fixed global frame. We propose to use a different fixed global frame. The new fixed, global coordinate system is rotated about the home plate to simplify the calculations. See Figure 5.

In the initial phase, the line connecting the fielder (or robot) and the point described by the projection of the ball onto the ground plane will define the direction of the x' axis. The original coordinate system will then be rotated by an amount

$$\theta = \tan^{-1}\left(\frac{(y_{fi} - y_{bi})}{(x_{fi} - x_{bi})}\right). \tag{8}$$

The fielder or the robot do not define a fixed global system when carrying out the task, but a coordinate system must be defined for simulation purposes in order to compare algorithms and develop control algorithms. The fielder will maintain either of two principles constant optical rate, or angular constancy.

In the new coordinate system, the robot's position will be adjusted in order to keep the height of the ball in the CCD image constantly rising. It is important to note that experimentally, the ball will rise at a constant rate in the CCD image plane, but the fielder will not calculate or know the exact position of the ball. The fixed coordinate system is used for simulation and analysis only. The 3D OAC algorithm is defined for simulation purposes:

$$\tan\alpha = C't \tag{9}$$

$$x'_f = x'_b + \frac{z'_b}{C't} \tag{10}$$

$$y'_f = y'_b. \tag{11}$$

As in the planar analysis, the height of the ball can be calculated in the plane that passes through the y' axis ($z_{imageOAC}$) or in the plane that moves (forward or backward) with the fielder (z_{ccdOAC}). The moving plane is orthogonal to the x' axis. See Figure 5.

$$z_{imageOAC} = x'_f C't \tag{12}$$

$$z_{ccdOAC} = x'_f(0)C't = A(constant)t \tag{13}$$

$$C' = \frac{z'_b(0.3s)}{0.3(x'_f(0.3) - x'_b(0.3))} \tag{14}$$

As stated earlier, the constants are determined after the ball is hit, and we have chosen a delay of 0.3 seconds corresponding to a camera delay.

The mobile robot algorithm is defined in the image that moves forward and backward with the robot. The robot will keep the ball vertically rising as well as centered laterally in the image plane. The robot controller is based on the position of the ball defined by (y_{ccd}, z_{ccd}) in the camera image.

$$\dot{x}'_f = K_p(z_{ccd} - at) \tag{15}$$

$$\dot{y}'_f = K_p(y_{ccd} - 0) \tag{16}$$

3.3 LOT Model

3.3.1 Fixed LOT Image plane

A mathematical description of the LOT model in world coordinates was developed by Aboufadel [4], but it is very complicated because a rotated coordinate system is not used. In the rotated coordinate frame, the LOT image plane (vertical plane) is aligned with the y' axis. See Figure 5. The robot adjusts its position to keep the position of the ball monotonically rising along the vector described by the angle, ψ, which

lies in the LOT image plane. In this analysis the LOT image plane does not move forward or backward. It is fixed.

$$\tan\psi = C_2' \quad (17)$$

$$x_f' = \frac{C_2' x_b'}{C_2' - \frac{z_b'}{y_b'}} \quad (18)$$

$$y_f' = y_b' \quad (19)$$

$$C_2' = \frac{z_b'(0.3)x_f'(0.3)}{y_b'(0.3)(x_f'(0.3) - x_b'(0.3))} \quad (20)$$

$$z_{imageLOT} = x_f' \tan(\alpha) \quad (21)$$

$$z_{ccdLOT} = x_f'(0)\tan(\alpha) \quad (22)$$

It is important to note that time does not appear in the equations for the LOT model. Also, if the ball is hit directly at the fielder, the LOT model breaks down due to a singularity.

The height of the ball in the fixed image plane is given by $z_{imageLOT}$. The height of the ball in the CCD-image plane which moves with the fielder is given by z_{ccdLOT}.

3.3.2 Moving LOT Image plane

The LOT algorithm is different if the fielder keeps the ball moving in a straight line in the moving image plane. Again, as the fielder moves forward or backward, the moving image plane remains a fixed distance, D from the fielder. The constraint is due to the focal distance, d, of the camera. See Figure 5. The new algorithm is much simpler.

$$\tan\psi = C_2' \quad (23)$$

$$x_f' = x_b' + \frac{D z_b'}{C_2' y_b'} \quad (24)$$

$$y_f' = y_b' \quad (25)$$

$$D = x_f'(0) \quad (26)$$

$$z_{movingccdLOT} = D\tan(\alpha) \quad (27)$$

The control algorithm for the robot will be calculated in the image plane that moves forward and backward with the robot, and does not rely on a fixed coordinate frame. The robot will keep the ball vertically rising proportional to the lateral distance in the image plane. The robot controller is based on the position of the ball defined by (y_{ccd}, z_{ccd}) in the camera.

$$\dot{x}_f' = K_p(z_{ccd} - C_2' y_{ccd}) \quad (28)$$

$$\dot{y}_f' = K_p(y_{ccd} - 0) \quad (29)$$

4 Simulation

Simulations for the robot's trajectories using the OAC and the two LOT models are shown next. In these simulations, the ball trajectory is $x_b = 75t$, $y_b = 10t$, and $z_b = -64t^2 + 256t$. The fielder's initial position is $x_f = 270$ and $y_f = 70$. The constants, C' and C_2', are calculated in the rotated frame at $0.3s$ right after the ball is hit, and remain fixed. The ball trajectory as well as the fielder's trajectory are shown in Figure 6. The fielder's trajectories in the (x, y) plane are shown in Figure 7. Different initial positions for the fielder are explored and shown in Figure 8.

The ball's trajectory is first rotated into the (x', y') frame and then the algorithms are used to calculate the fielder's trajectory. All data is then rotated back to the original (x, y) frame for presentation.

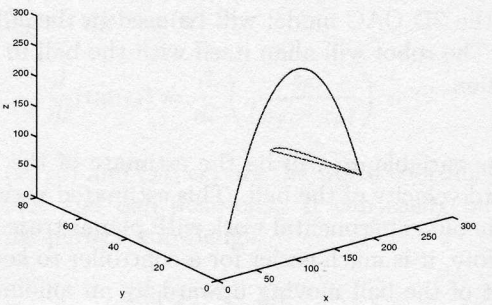

Figure 6: The ball and the fielder's trajectories are shown in the (x, y, z) frame. The OAC and moving LOT algorithms describe a straight path to the destination while the LOT algorithm describes a slightly curved path. The ball's trajectory is first rotated into the (x', y') frame and then the algorithms are used to calculate the fielder's trajectory. All data is then rotated back to the original (x, y) frame for presentation.

All three algorithms are used to calculate trajectories that allow a robot to intercept the projectile at the destination even with different initial conditions. These curves are smooth because noise has not been added to either the projectile or the fielder's position. The OAC and the moving LOT algorithm describe straight constant paths while the fixed LOT algorithm describes curved paths.

The projectile path was altered by wind which causes the ball to shift sideways and drag which causes the ball to land quicker. See Figure 9. The fielder takes an initial look at the ball after $0.3s$ and the values for C' and C_2' are calculated. The constant initial

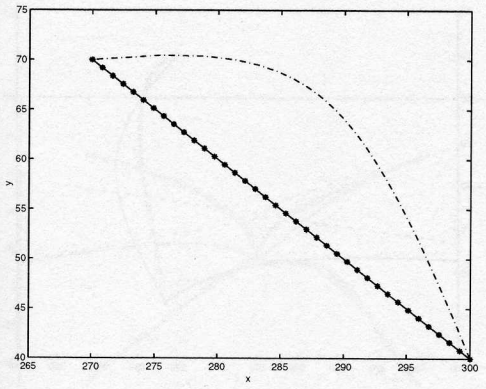

Figure 7: The fielder's trajectories for both the OAC and the two LOT models are shown in the (x, y) plane. The OAC (solid) and moving LOT (*'s) algorithms describe a straight, shortest distance path. The fixed LOT (dashed) algorithm describes a curved path. Both LOT algorithms have the advantage of being independent of time.

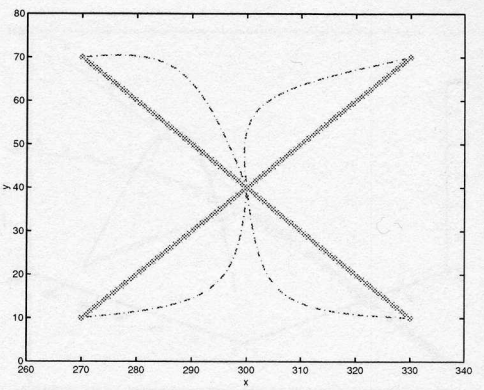

Figure 8: The fielder's trajectories for both the OAC and the two LOT models are shown in the (x, y) plane for different initial conditions. The OAC (solid) and moving LOT (*'s) algorithms describe a straight, shortest distance path. The fixed LOT (dashed) algorithm describes curved paths.

values for C' and C_2' are used throughout the simulation even though the path of the projectile changes. The perceptual algorithms describe a model where a constant rate or a constant angle is used during the interception task. Even with constant values, the fielder is able to intercept the projectile which takes a slightly different path.

The fielder's paths in the (x, y) plane are shown in Figure 10 and different initial conditions are shown in Figure 11.

4.1 Image Plane

It is interesting to understand the projection of the ball in the image plane which passes through the y' axis. The paths that were generated by the algorithms in the previous section are just desired paths. The robot will actually move depending on the data in the image plane.

The initial conditions are the same $x_b = 75t$, $y_b = 10t$, and $z_b = -64t^2 + 256t$ and wind and drag are eliminated. The fielder's initial position is again $x_f = 270$ and $y_f = 70$. If the ball follows a parabolic trajectory, both LOT algorithms define a straight line in the image plane (almost straight for the OAC), and all of the lines are very close. See Figure 12.

If the position of the ball is calculated in the moving image plane, then the OAC algorithm, z_{ccdOAC}, matches exactly with the original LOT curves, $z_{imageLOT}$ and $z_{movingccdLOT}$. In this simulation, these three lines exactly match.

In the second case, wind alters the trajectory of the ball, but all algorithms describe paths to intercept the projectile. In the image plane, the ball's projection is very different using the two algorithms. The OAC algorithm describes a curved path, but $\frac{d}{dt}(\tan(\alpha)) = C$ only controls the height of the ball insuring that it rises at a constant rate in the CCD-image. On the other hand, the LOT algorithm forces the curve to be a straight line. See Figure 13.

5 Applications to Mobile Robots

Experiments with a mobile robot pursuing a ball with a planar trajectory have been started [7]. In these experiments, the upward velocity, a, is calculated using an average of the upward movement of the ball in the first four frames. This average, a, is calculated in pixels/frame, and then used throughout the experiment. The control algorithm has been shown to be robust to large errors in the value of a [7]. Subsequent movements of the robot forward and backward are based on keeping this experimental value constant.

The simple control algorithm for the robot is based on image data only. The error between the actual height of the ball and the desired height of the ball in the image plane is calculated.

$$\dot{x}_f = K_p \left(z_{ccd} - at \right) \tag{30}$$

In one experiment, the actual pixel centroid of the ball moves linearly upward in the image plane during three quarters of the experiment, and the robot is able to maintain a constant optical rate of the height of the ball's image. At the end of the experiment, the difference between the actual centroid and the desired

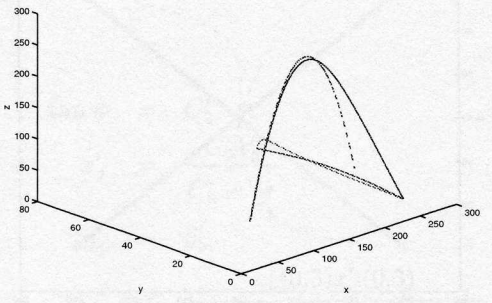

Figure 9: The ball and the fielder's trajectories are shown in the (x, y, z) frame. In this case, the ball lands at a different location than expected because wind pushes the ball to the side and drag causes the ball to land shorter than expected. All three algorithms describe trajectories that allow the fielder to intercept the ball.

centroid in the image plane increases, and hence, the robot accelerates quickly. See Figure 14.

In the figure, the camera frame is along the horizontal axis. In the top figure, the actual robot velocity and the desired robot velocity is measured in inches/s. In the bottom figure, the centroid of the ball is along the vertical axis and is measured in pixels.

6 Discussion

We are investigating principles such as constant optical rate, and angular constancy in the image plane and applying these principles to navigational pursuit. With these principles, control algorithms for visual-servoing using image data will allow mobile robots to intercept projectiles with complicated, varying trajectories. We developed algorithms for both the OAC and the LOT models proposed by psychologists *in three dimensions* and in the future controllers will be implemented on experimental hardware. The LOT algorithm calculated in the moving image plane performed very well in parabolic and windy ball trajectories. We believe the multi-disciplinary research between perceptual psychology and robotics will help us develop navigational algorithms that will be robust, powerful, and simple.

Acknowledgments

Support from ASU is gratefully acknowledged.

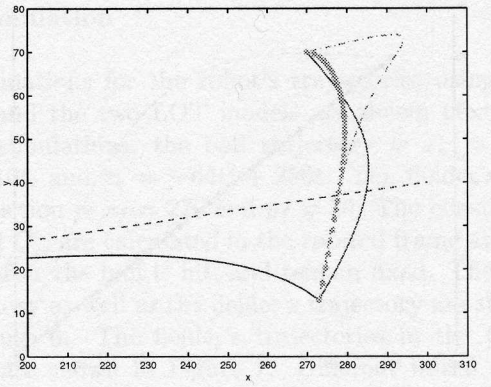

Figure 10: The fielder's trajectories for the OAC and the two LOT models are shown in the (x, y) plane. The dashed line describes the ball's original trajectory if drag and wind are not present. The solid line describes the ball's trajectory. The OAC (solid), moving LOT (*'s), and the fixed LOT (dashed) algorithms describe paths that are curved and intercept the ball at approximately $(270, 10)$ and start at $(270, 70)$.

References

[1] A. Milner and M. Goodale, *The Visual Brain in Action.* Oxford Psychology Series 27, Oxford University Press, 1995.

[2] P. McLeod and Z. Dienes, "Running to catch the ball," *Nature*, vol. 362, no. 6415, 1993.

[3] M. K. McBeath, D. M. Shaffer, and M. K. Kaiser, "How baseball outfielders determine where to run to catch fly balls," *Science*, vol. 268, no. 5210, pp. 569–573, 1995.

[4] E. Aboufadel, "A mathematician catches a baseball," *American Mathematics Monitor*, vol. 103, pp. 870–878, 1996.

[5] R. C. Arkin, *Handbook of Brain Theory and Neural Networks*, ch. Reactive Robotic System, pp. 793–796. MIT Press, 1995.

[6] J. A. Borgstadt and N. J. Ferrier, "Interception of a projectile using a human vision-based strategy," in *IEEE International Conference on Robotics and Automation*, 2000.

[7] A. Suluh, T. Sugar, and M. McBeath, "Spatial navigational principles: Applications to mobile robotics," in *IEEE International Conference on Robotics and Automation*, 2001.

[8] H. Zhang and J. P. Ostrowski, "Visual servoing with dynamics: Control of an unmanned blimp," in *1999 International Conference of Robotics and Automation*, 1999.

[9] R. Vidal, S. Rashid, C. Sharp, O. Shakernia, and S. S. Sastry, "A testbed for multi-agent pursuit-evasion games with unmanned ground and aerial vehicles," in *IEEE International Conference on Robotics and Automation*, 2001.

[10] W. Hong, "Robotic catching and manipulation using active vision," Master's thesis, MIT, 1995.

[11] R. G. Watts and R. T. Bahill, *Keeping Your Eye on the Ball.* NY: Freeman and Co., 1990.

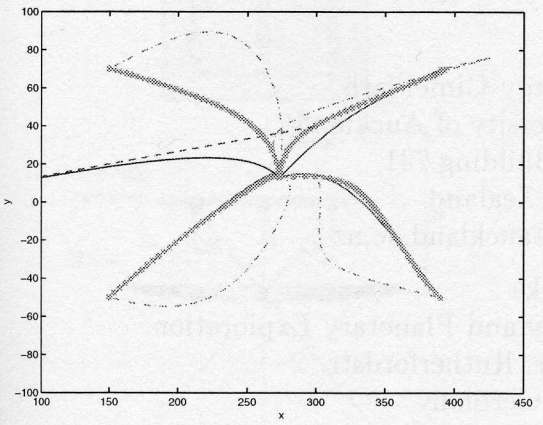

Figure 11: The fielder's trajectories for the OAC (solid), the moving LOT (*'s) and the fixed LOT (dashed) are shown in the (x, y) plane for different initial conditions. The more sweeping and curved paths are calculated using the fixed LOT algorithm.

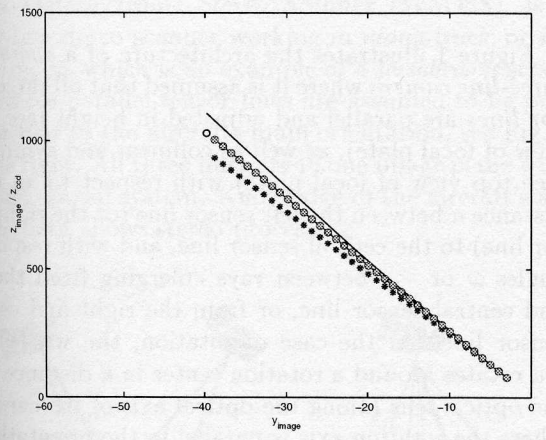

Figure 12: The position of the ball in the projection plane is shown for the OAC (solid black = $z_{imageOAC}$, black o's = z_{ccdOAC}) and LOT (solid blue = $z_{imageLOT}$, blue *'s = z_{ccdLOT} and green *'s = $z_{movingccdLOT}$) algorithms. If the ball follows a parabolic trajectory, then the three curves, $z_{imageLOT}$, $z_{movingccdLOT}$, and z_{ccdOAC} are straight and exactly match. All curves are very close.

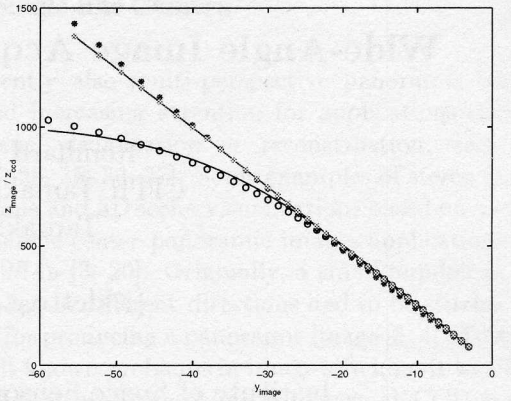

Figure 13: The position of the ball in the projection plane is shown for the OAC (solid black = $z_{imageOAC}$, black o's = z_{ccdOAC}) and LOT (solid blue = $z_{imageLOT}$, blue *'s = z_{ccdLOT} and green *'s = $z_{movingccdLOT}$) algorithms. The ball's path is altered by wind and drag. In this case, the trajectories of the fielder are changed, but the ball is still intercepted. The projections $z_{imageOAC}$ and z_{ccdOAC} of the ball using the OAC algorithm are curved and significantly altered. In the LOT case, the projections of the ball $z_{imageLOT}$ and $z_{movingccdLOT}$ are forced to remain on a straight line and exactly match. The projection, z_{ccdLOT} is nearly straight but has a slight curve.

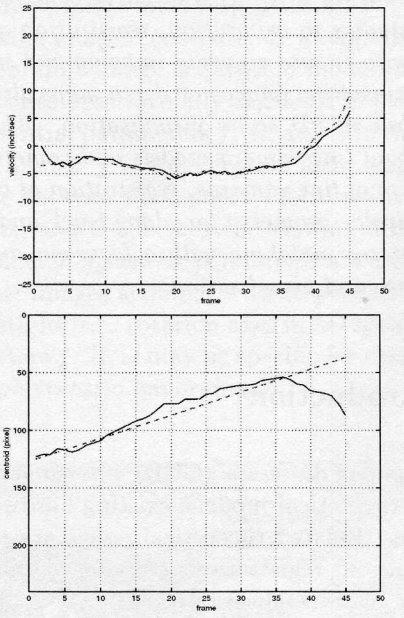

Figure 14: In the top figure, the desired (dashed) and actual velocity of the robot is given for a ball which lands in front of the robot. In the bottom figure, the desired and actual center of mass of the object is shown in pixels.