The Chase Problem (Part 2)

David C. Arney

Introduction

In the previous section, entitled The Chase Problem (Part 1), we discussed a discrete model for a chasing scenario where one thing chases another. Some of the applications of this kind of chasing were given in the examples of the previous section: missiles intercepting other missiles, anti-tank round seeking a tank, and torpedo tracking an enemy ship. In this section, we extend and refine this first model, build a continuous model for this problem, and build more effectiveness and sophistication into our chase algorithm.

The Problem

Our problem is to determine the movement path for the chaser, given that we know the location (and sometimes more information) of the target. We start with the assumption that the chaser knows the target's position exactly. The chaser's position is represented in two-dimensional Cartesian coordinates by $(x_0(t), y_0(t))$. We also assume that the chaser moves at a constant speed (given by s) and the target's position in two dimensions is given by the parametric relationship $(x_1(t), y_1(t))$. We start with the technique that the chaser moves directly towards the target. Later we'll allow for the chaser to "lead" the target. As the location of the target changes, the chaser continually adjusts its path to continue to move directly toward the target.

The Model

We can model this procedure with a system of differential equations, one for each space dimension we model. Since we'll perform our modeling in two space dimensions (x and y), we'll build systems of two differential equations. Our modeling process for a continuous time changing event is to set up relationships that express the derivatives of the changing variables $(x'_0(t), y'_0(t))$ in terms of functions of variables.

Let's draw a diagram of what happens during a time interval Δt. We will take the limit as $\Delta t \to 0$ in order to get continuous feedback and continuous movement for the chaser. Figure 1 shows the locations of the chaser and the target at some time t. The movement made by the chaser over that interval is indicated. The final position of the chaser is given by $(x_0(t+\Delta t), y_0(t+\Delta t))$.

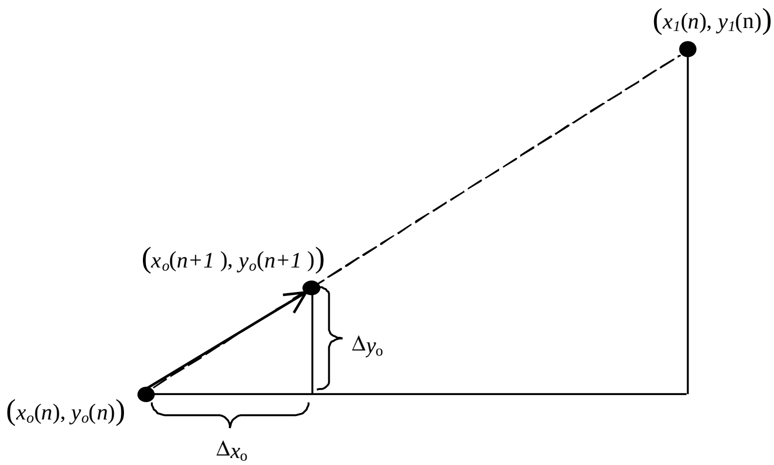

Figure 1: Movement by the chaser during the time interval from step *n* to step *n+1* .

We see two similar triangles in Figure 1. Like the discrete case developed in Part 1, the relation between these similar right triangles enables us to write our model. Recall that we can set up proportional equations relating the sides of the triangles with the hypotenuse of the triangles. First let's determine formulas for each of the sides of our two triangles. The larger triangle in Figure 1 has its horizontal side of length $(x_1(t)- x_0(t))$. The vertical side has length $(y_1(t)- y_0(t))$. Therefore, by the Pythagorean Theorem the hypotenuse has length $\sqrt{(x_1(t)-x_0(t))^2+(y_1(t)-y_0(t))^2}$. The smaller triangle has sides $x_0(t+\Delta t)-x_0(t)=\Delta x_0$ and $y_0(t+\Delta t)-y_0(t)=\Delta y_0$. We need to determine the length of the hypotenuse in terms of values other than Δx_0 and Δy_0 . We also know that the chaser moves at speed *s.* Therefore, the length of the hypotenuse can be approximated by the distance traveled during the time interval or $s\Delta t$. We write out the equations relating the sides of the triangles with the hypotenuse of the triangles. First the horizontal side and hypotenuse of both triangles produce the relationship:

$$\frac{x_0(t+\Delta t)-x_0(t)}{s\Delta t}=\frac{x_1(t)-x_0(t)}{\sqrt{(x_1(t)-x_0(t))^2+(y_1(t)-y_0(t))^2}} \qquad (1)$$

The vertical sides and hypotenuse produce:

$$\frac{y_0(n+1)-y_0(n)}{s\Delta t}=\frac{y_1(n)-y_0(n)}{\sqrt{(x_1(n)-x_0(n))^2+(y_1(n)-y_0(n))^2}} \qquad (2)$$

We now have difference quotients on the left sides of Equations (1) and (2). We take the limit as $\Delta t \to 0$ of (1) and (2) to produce the differential equations

$$\lim_{t \to 0} \frac{x_0(t+\Delta t) - x_0(t)}{\Delta t} = \frac{dx_0(t)}{dt} = \frac{s(x_1(t) - x_0(t))}{\sqrt{(x_1(t)-x_0(t))^2 + (y_1(t)-y_0(t))^2}} \tag{3}$$

$$\lim_{t \to 0} \frac{y_0(t+\Delta t) - y_0(t)}{\Delta t} = \frac{dy_0(t)}{dt} = \frac{s(y_1(t) - y_0(t))}{\sqrt{(x_1(t)-x_0(t))^2 + (y_1(t)-y_0(t))^2}} \tag{4}$$

This is our chase/movement model (equations (3) and (4)), which will provide a means of determining the movement of the chaser, when we know the movement of the target. This system of differential equations is nonlinear and must be solved numerically. Therefore, we will need a numerical solver on a computer or calculator to determine the path of the chase. Many software packages that use Euler's method or the Runge-Kutta method are available. It is also possible to implement these algorithms by converting the differential equations to difference equations and implementing iteration on a spread sheet. Remember our assumptions: the chaser moves at a constant speed and the chaser always sees the target. When we solve our differential equations with a numerical method, we actually model the chaser moving toward the target for a set time interval, Δt, used in the numerical scheme. We usually set the time interval to be very small to assure accuracy of the solution and to approximate accurately the continuous movement of the chaser.

We need to determine when to stop our calculations. In the previous section, we discussed several of the factors involved in this decision. There is no need to continue after the chaser has caught the target. We need a stopping criteria that reflects "catching" the target. We will assume that "catching" the target means just being "close enough" or within the tolerance of the stopping criteria denoted by e. We have choices for determining this tolerance value. It could be a fixed value or a function of the speed *s* and time interval Δt. We implement stopping criteria in our model by determining the distance, denoted *d(t)*, between the chaser and target after each iteration. The value of *d(t)* is determined by the distance formula between two points,

$$d(t) = \sqrt{(x_0(t)-x_1(t))^2 + (y_0(t)-y_1(t))^2} \tag{5}$$

We stop when $d(t) < e$ or after a specified amount of time has expired ($t > M$). Let's look at an example to see how this works.

Example 1: Anti-tank Round

A soldier located at point *(0,3)* launches a tracking anti-tank round with speed 3.5 at a tank at time $t = 0$ following a elliptical course given by the following parametric equations:

$$x_1(t) = 8 - 3\cos t \quad \text{and} \quad y_1(t) = 4\sin t \tag{6}$$

We will determine the round's path based on its seeker using our tracking model of moving directly toward the tank. If the kill radius of the round is *0.25* units, our stopping criteria is $e = 0.25$. Substituting the known values and functions into our model, Equations (3) and (4), produces

$$\frac{dx_0(t)}{dt} = \frac{3.5(8 - 3\cos t - x_0(t))}{\sqrt{(8 - 3\cos t - x_0(t))^2 + (4\sin t - y_0(t))^2}} \tag{7}$$

$$\frac{dy_0(t)}{dt} = \frac{3.5\,(4\sin t - y_0(t))}{\sqrt{(8 - 3\cos t - x_0(t))^2 + (4\sin t - y_0(t))^2}} \tag{8}$$

Starting with our initial condition, $x_0(0) = 0$ *and* $y_0(0) = 3$, we use a Runge-Kutta solver with $\Delta t = 0.06$ until we achieve our stopping criteria of $d(t) < e = 0.25$. This produces the solution for the path of the round. The graphs of the paths for both the round and the tank, until their impact at a time slightly great than *5* seconds, are given in Figure 2. Notice how the round curves around to follow the tank and eventually catches it.

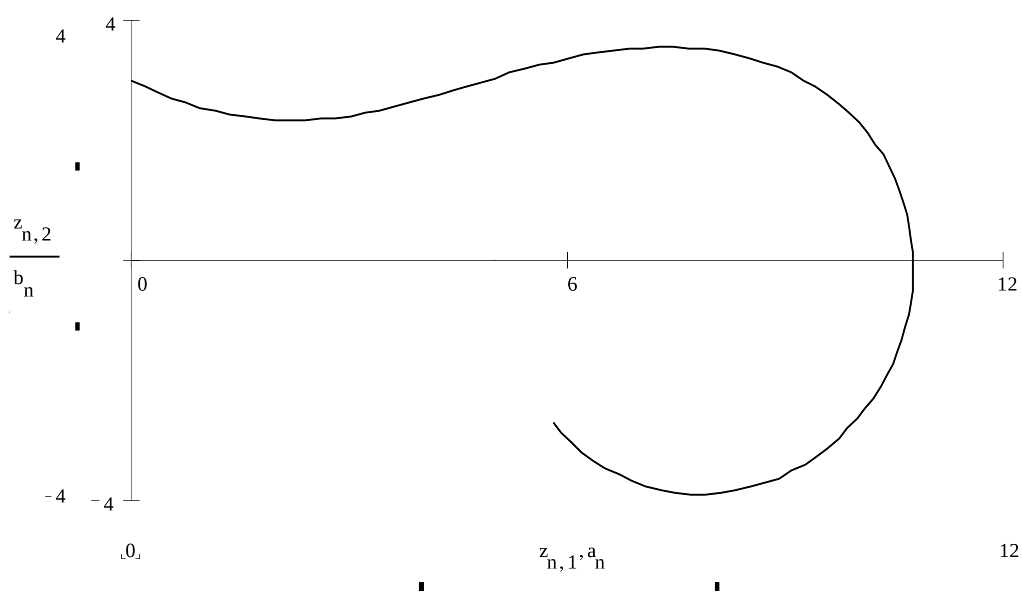

Figure 2. Graph of the paths of the round (solid curve) and the tank (dotted curve), from launch (*t=0*) to impact.

Does our solution make sense? Does the chaser move in an efficient path toward the target? Does the chaser stop when the stopping criteria is achieved? In general the answers to these questions are “yes”. It appears that we have a good model, but it may not be the best. It could help if the round was able to “lead” the tank so it could catch it faster. We’ll try implementing a “lead” algorithm for this chase problem later in this section.

The Modeling Process

Let's review our modeling process for this problem. Our behavior of interest, the movement of the chaser, is continuous in nature.   We modeled this movement with a continuous differential equation. Our solution method for this model, the Runge-Kutta numerical method, is discrete and gives an approximate solution to the continuous model. We then converted the discrete sequence of locations of the path to a continuous path by connecting the points in the graphs of Figure 2. We show this interplay between discrete and continuous representations in our modeling process in Figure 3.

Behavior	Model	Solution Method	Verification Method
(movement of chaser)	(differential equations)	(Runge-Kutta numerical scheme)	(graph in Figure 2)

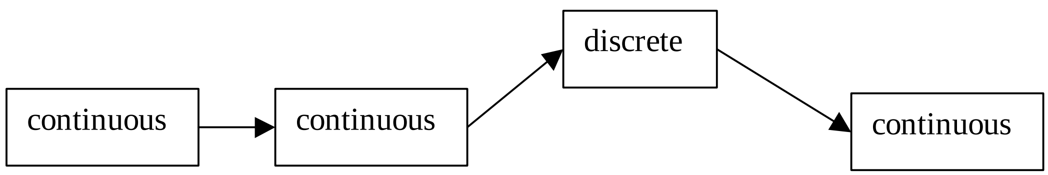

Figure 3. Interplay between discrete and continuous in the modeling process of the differential equation chase model.

The "lead" algorithm

How do we get the chaser to "lead" the target? We need to take into account both the speed and the velocity of the target, then use that information to predict where the target will be when the chaser catches the target. We use the Taylor polynomial to do this. The Taylor polynomial is an approximation to a function. For the function *f(x)* and using the start point as *x*=*a*, we write the Taylor polynomial of degree *n* as

$$f(x) \approx f(a) + f'(a)(x-a) + \frac{f''(a)}{2!}(x-a)^2 + \frac{f'''(a)}{3!}(x-a)^3 + \ldots + \frac{f^{(n)}(a)}{n!}(x-a)^n \quad (9)$$

We can use this approximation for the two functions $x_1(t)$ and $y_1(t)$ representing the two dimensions of the target's path. First, let's use the 1st-degree polynomial approximations, which take into account the location and the velocity (but not the acceleration). We must be expeditious in our selection of *a* and *x* in Equation (9), and set *n=1*. To get our approximations in the proper form, we use $x=t+\Delta t$ and $a=t$, and therefore, $(x-a)=\Delta t$. Then, we can write

$$x_1(t+\Delta t) = x_1(t) + x_1'(t)\Delta t \quad \text{and} \quad y_1(t+\Delta t) = y_1(t) + y_1'(t)\Delta t \ . \qquad (10)$$

The value of $\Delta\ t$ is the value of the time advance to the location where the target is predicted to be, in order to have a proper lead. The "phantom" location to aim for is simply the point $(x_1(t+\Delta t), y_1(t+\Delta t))$. Therefore, we modify the model in Equations (3) and (4), using this "phantom" lead point in place of *$(x_1(t), y_1(t))$* and the formulas in Equation (10) to obtain

$$\frac{dx_0(t)}{dt} = \frac{s(x_1(t) + x_1'(t)\Delta t - x_0(t))}{\sqrt{(x_1(t) + x_1'(t)\Delta t - x_0(t))^2 + (y_1(t) + y_1'(t)\Delta t - y_0(t))^2}} \tag{11}$$

$$\frac{dy_0(t)}{dt} = \frac{s(y_1(t) + y_1(t)\Delta t - y_0(t))}{\sqrt{(x_1(t) + x_1'(t)\Delta t - x_0(t))^2 + (y_1(t) + y_1'(t)\Delta t - y_0(t))^2}} \tag{12}$$

The geometry of this lead algorithm is shown in Figure 4.

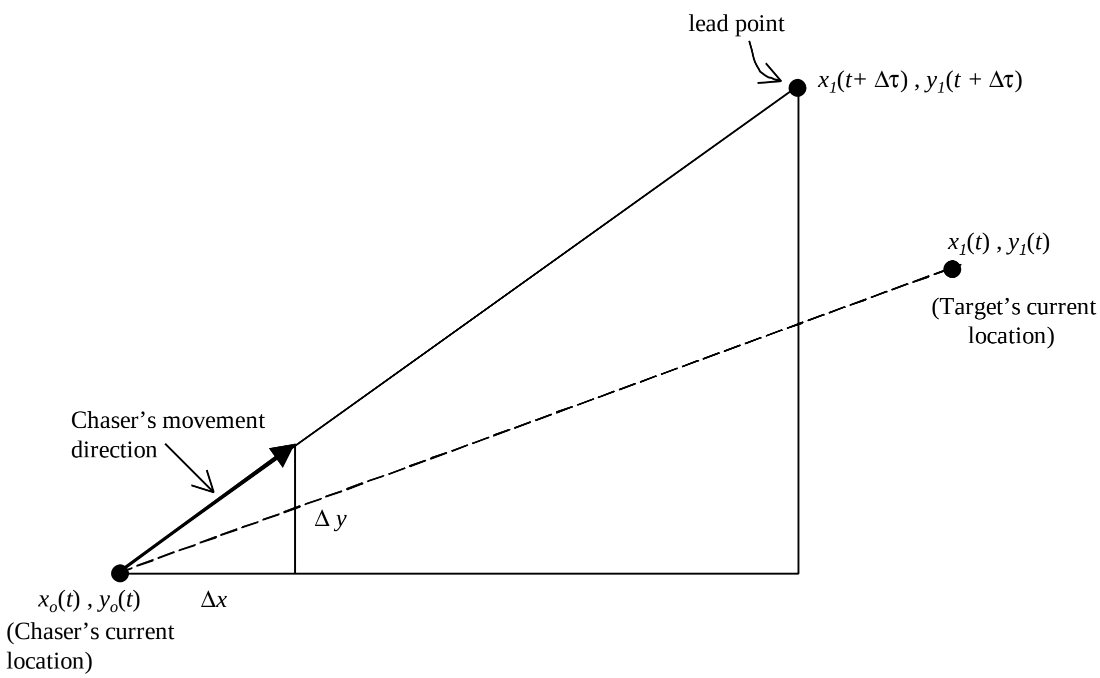

Figure 4: Path of chaser when heading for the "lead" point.

We need an algorithm to determine the value for Δt. How much should we lead? We could do this several ways. One way is to think of $\Delta\ t$ as the time needed to catch the target. We will approximate this "catch" time by using the time for the chaser to reach the target's current location. Therefore, the formula for $\Delta\ t$ is simply the current distance between the chaser and the target given in Equation (5) divided by the speed *s*. We write this as

$$\Delta t = \frac{\sqrt{(x_0(t) - x_1(t))^2 + (y_0(t) - y_1(t))^2}}{s} \tag{13}$$

Then our new “lead” model is formed by substituting Equation (13) into Equations (11) and (12):

$$\frac{dx_0(t)}{dt}=\frac{s\left(x_1(t)+x_1'(t)\frac{\sqrt{(x_0(t)-x_1(t))^2+(y_0(t)-y_1(t))^2}}{s}-x_0(t)\right)}{\sqrt{\left(x_1(t)+x_1'(t)\frac{\sqrt{(x_0(t)-x_1(t))^2+(y_0(t)-y_1(t))^2}}{s}-x_0(t)\right)^2+\left(y_1(t)+y_1'(t)\frac{\sqrt{(x_0(t)-x_1(t))^2+(y_0(t)-y_1(t))^2}}{s}-y_0(t)\right)^2}}$$

(14)

$$\frac{dy_0(t)}{dt}=\frac{s\left(y_1(t)+y_1'(t)\frac{\sqrt{(x_0(t)-x_1(t))^2+(y_0(t)-y_1(t))^2}}{s}-y_0(t)\right)}{\sqrt{\left(x_1(t)+x_1'(t)\frac{\sqrt{(x_0(t)-x_1(t))^2+(y_0(t)-y_1(t))^2}}{s}-x_0(t)\right)^2+\left(y_1(t)+y_1'(t)\frac{\sqrt{(x_0(t)-x_1(t))^2+(y_0(t)-y_1(t))^2}}{s}-y_0(t)\right)^2}}$$

(15)

Let’s try this model in our previous scenario of an anti-tank round.

Example 2: Anti-tank Round (revisited)

A soldier located at *(0,3)* launches a tracking anti-tank round with speed 3.5 at a tank at time $t = 0$, following a elliptical course given by:

$$x_1(t) = 8 - 3\cos t \qquad \text{and} \qquad y_1(t) = 4\sin t \ . \qquad (16)$$

We determine the round’s path based on its seeker using our new tracking model of “leading” the tank. We use $e = 0.25$ and substitute the known values and functions into Equations (14) and (15). Starting with the initial condition, $x_0(0) = 0$ *and* $y_0(0) = 3$, we use a Runge-Kutta solver with $\Delta t = 0.06$ until we achieve our stopping criteria of $d(t) < e = 0.25$. This produces the solution for the path of the round. The graphs of the paths for both the round and the tank until their impact are given in Figure 5. By comparing this graph with that of Figure 2, we see that the lead algorithm saves time and chase distance by turning sooner and more sharply to catch the tank.

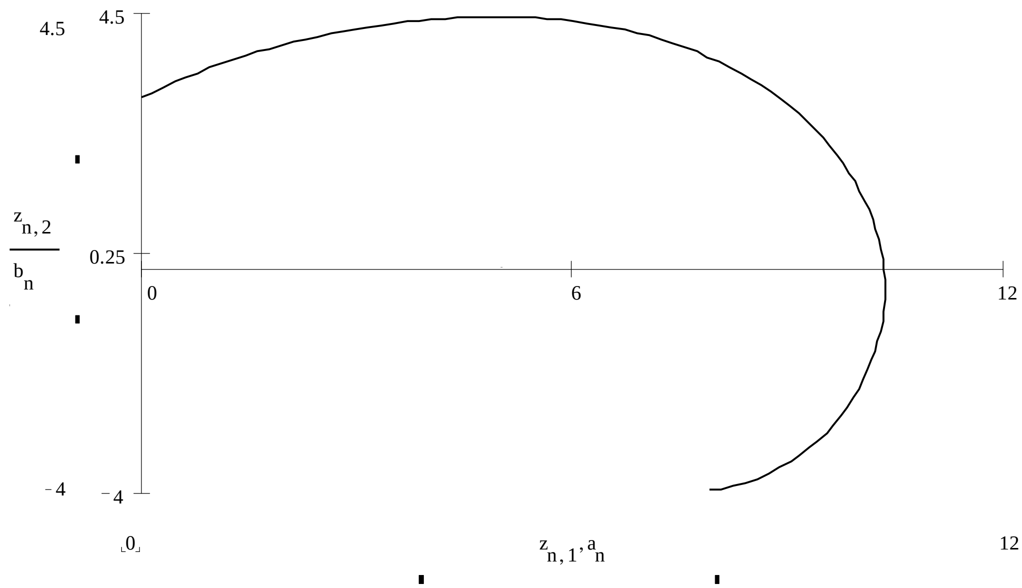

Figure 4. Graph of the paths of the round (solid curve) and the tank (dotted curve), from launch (*t=0*) to impact.

It appears that we have developed a better model by leading the target. Is it the best we can do? Are there better lead models? Of course, these are leading questions. There probably are better ways to compute Δt. And we could keep more than just one term of the Taylor Polynomial Equation (9). If we keep two terms, *n=2*, we would get new approximations for our "lead" location of the target. We would now be taking into account the acceleration of the tank, as well as its velocity. The new model is written as follows:

$x_1(t+\Delta t) = x_1(t) + x_1'(t)\Delta t + x_1''(t)\Delta t^2$ and $y_1(t+\Delta t) = y_1(t) + y_1'(t)\Delta t + y_1''(t)\Delta t^2$. (17)

Substitution of these formulas into Equations (3) and (4), along with using Equation (13) for Δt, creates a model with very large, messy differential equations. We won't try to show them here, but we'll show an example which compares the various models we have discussed.

Example 3: To Lead or not to Lead

This time the soldier firing the anti-tank round is located at the origin, *(0,0).* He launches a tracking round with speed 5 at a tank at time *t = 0.* The tank follows an oscillating course given by:

$$x_1(t) = 3 + 3t \qquad \text{and} \qquad y_1(t) = 2\sin(3t) \qquad (18)$$

We determine the round's path based on its seeker using three different tracking models: 1) moving directly toward the tank, 2) leading the tank by using the velocity (one derivative term in the Taylor polynomial as in Equations 14 and

15), and 3) leading the tank by using velocity and acceleration (including 2 terms in the Taylor Polynomial shown in Equation 17). Our stopping criteria is $e=0.25$. The path of the round tracking directly for the tank is given in Figure 5. The round catches the tank at $t=3.9$ seconds. The path of the round when leading the target using the tank's velocity is shown in Figure 6. This path is more direct and catches the tank in 2.8 seconds. Finally, the new lead algorithm using both velocity and acceleration produces the graph in Figure 7. This model produces a catch at 2.7 seconds. This last method is not much faster than the velocity only model. Sometimes it doesn't help or may even hinder to lead the target, but, in general, the more information you use the quicker you can catch your target.

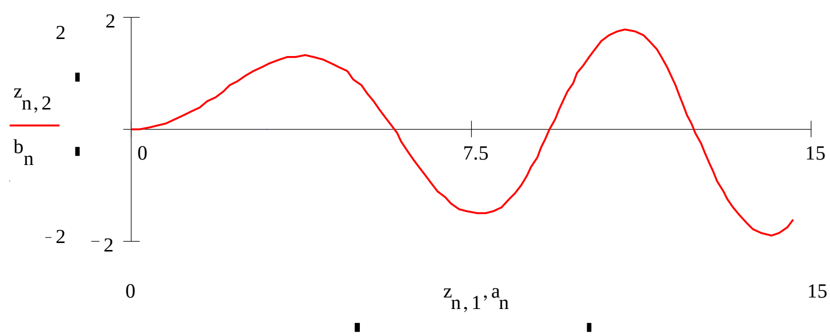

Figure 5: Paths of target (dotted curve) and chaser (solid curve) using the model of moving directly toward the target.

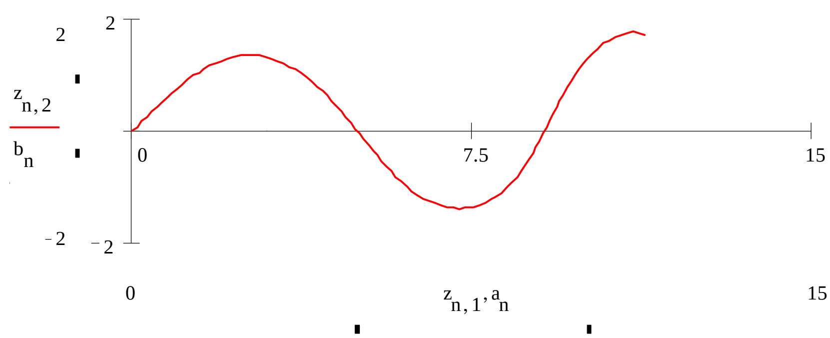

Figure 6: Paths of target (dotted curve) and chaser (solid curve) using the model of leading the tank by using the velocity.

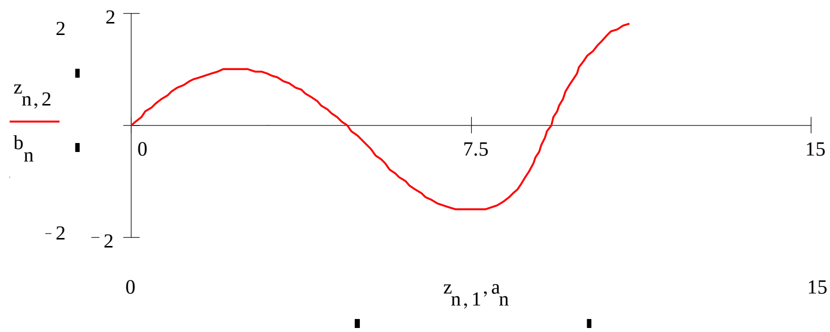

Figure 7: Paths of target (dotted curve) and chaser (solid curve) using the model of leading the tank by using both the velocity and acceleration.

In this section, we have studied and solved a challenging problem with many applications. Our model and its solutions have performed well in the examples we have solved. We now know what happens when we lead the target, instead of moving directly toward it. There are still many questions we have not addressed. What about the maneuverability of the chaser? Can it always turn fast enough to make the necessary moves of the algorithm? How should the target move to evade the chaser? These are difficult questions that merit further study and more sophisticated mathematical models. Good luck to those who study this important problem with numerous military applications.

Exercises

1. An enemy tank, currently at location (12,0), is moving in a zigzag pattern away from your location with parametric equations: $x_1(t) = 12 + 3t$ and $y_1(t) = 2\sin(2t)$. You launch a tank tracking round moving at a speed of 12 from your location at (0,0). The guidance system of the radar-controlled round always moves directly toward the tank target.
a) Write a system of differential equations that models the movement of the round towards the target.
b) Use a numerical scheme to solve the equations and plot the solution for *0<t<2*.
c) What is the distance between the tank and the round at *t =2*? Is the round closer at *t = 2* or *t = 0* ?

2. Using the same general scenario as exercise 1 for an enemy tank starting at (12,0) and moving with equations: $x_1(t) = 12 + 3t$ and $y_1(t) = 2\sin(2t)$, you launch a tracking round moving at speed of 12 from your location at (0,0). This new improved round has a guidance system that leads the tank by considering its velocity.
a) Write a system of differential equations by substituting into Equations (15) and (16) that models the movement of the round towards the target.
b) Use a numerical scheme to solve the equations and plot the solution for *0<t<2*.

c) What is the distance between the tank and the round at *t =2*? Is the round closer at *t = 2* or *t = 0* ?

3. A ship located at (20,15) detects a torpedo at (15,6) and begins the evasive maneuver of moving directly away from the torpedo at a constant speed of 8.
a) What are the parametric equations, using time *t* as the parameter, for the path of the ship with *t=0* representing the start time of this path?
b) If the torpedo follows the ship with a speed of 10, what are the differential equations that govern the motion of the torpedo?

4. A ship located at (20,15) detects a torpedo at (15,6) and begins an evasive maneuver defined by the equations $x_1(t) = 20 + 3t$ and $y_1(t) = 15 + 4t + 3\sin(t)$.
a) If the torpedo follows the ship with a speed of 10, what are the differential equations that govern the motion of the torpedo using the 3 chase algorithms described in this section (direct intercept, lead using velocity, lead using velocity and acceleration)?
b) Solve the 3 models in part (a) and determine which algorithm guides the torpedo closest to the ship after 3 seconds.

5. Discuss the dichotomy of discrete and continuous mathematics. Include in your discussion examples of behaviors and functions that are naturally discrete and behaviors and functions that are naturally continuous.

References

Dunbar, Steven R., "Minimodule: Chase Problems," *The UMAP Journal*, vol. 15, no. 4, 1994, pp. 351-356.

Fellman, Bruce, "Guess Who's Coming to Dinner: Mechanisms which Help Insects Escape Bats," National Wildlife, vol.31, Feb 1993, pp. 42-45.

Johnston, Elgin and Mathews, Jerold, "Project Based Calculus at Iowa State," *Computer Algebra Systems in Education Newsletter*, no. 11, September 1991, pp. 1-7.

Simmons, George, *Differential Equations with with Applications and Historical Notes*, 2nd edition, New York: McGraw-Hill, 1991.

Yates, Robert C., *Differential Equations*, New York: McGraw-Hill, 1952 (pp. 109-111).

Yavin, Y. and Pachter, M. P. (editors), "Pursuit-Evasion Differential Games III," *Computers & Mathematics with Applications*, vol. 26, no. 6, Sep 93, pp. 1-152 (entire volume dedicated to this subject).