Project 2

Problem description

In this project the motion of a fox and a rabbit is being observed. Our task is to find out if the latter will be caught by the former. This is also known as the fox-rabbit chase problem. It is solved using pursuit curves derived from differential equations. Therefore, it has many formulations according to the paths and the speeds of the objects.

In our case, the fox starts from a point O, moves directly towards the opening G of a circular fence and then chases the rabbit depending on the fact if the prey is in sight or it is behind a straight line fence AE. If the former is true /as shown in Figure 1/, the chaser starts pursuing it and if it is not /as shown in Figure 2/ - it goes directly to A and once the view is not blocked anymore, the fox moves directly towards the target. The task is to find the time T, the coordinates of the predator and the distance traveled by it for the time T when either the rabbit escapes or is caught.

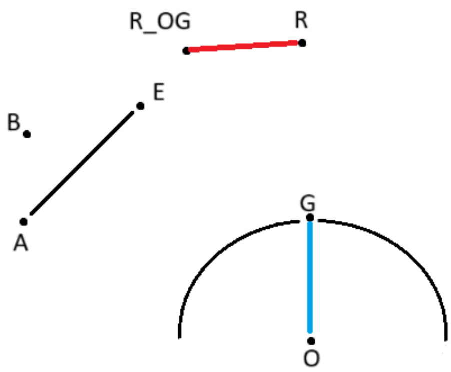

Figure 1: The view is not blocked.

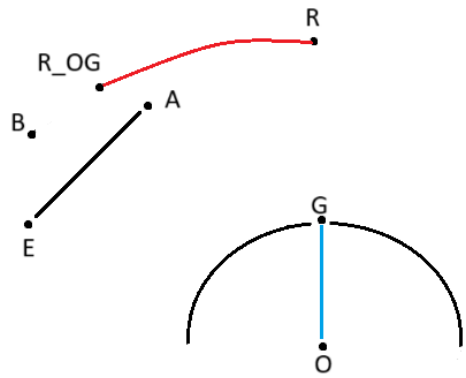

Figure 2: The view is blocked.

History

The history of the pursuit curves dates back in time and has many applications. ”Pursuit curves were considered in general by the French scientist Pierre Bouguer in 1732. However, the term ’pursuit curve’ was first defined by George Boole in his ’Treatise on differential equations’ in 1859. The curved path described by a fighter plane making an attack on a moving target while holding the proper aiming allowance is a pursuit curve, so such curves are relevant to current military research.” (Michael Lloyd, 2006-07, p.4)

Given information

In our problem we have the following information:

- The initial coordinates of the fox - O $[0, 0]$.
- The initial coordinates of the rabbit - R $[0, 800]$.

- The coordinates of the burrow $800[-\sin(\pi/3), \cos(\pi/3)] = [-692.8203, 400]$, the opening point G $[0, 300]$ and the fence AE ($A = [-350, 620]$, $E = [-500, 350]$).
- The rabbit follows a circular path towards the burrow.

Question 1

1) Problem solution

Now, let us discuss how this problem can be solved when we have constant speeds of $s_f = 17m/s$ for the fox and $s_r = 12m/s$ for the rabbit.

The main problem is to find several differential equations, which we would be able to apply in MATLAB later. First, we need to find an expression for the coordinates of the rabbit at any time t. Since it is moving on a circular path towards the burrow, they are:

$(x, y) = (-800 \times \sin(\theta(t)), 800 \times \cos(\theta(t)))$, where $\theta(t)$ describes the angle at time t. An equivalent expression is $(x, y) = (-800 \times \sin(s_r t/800), 800 \times \cos(s_r t/800))$, since $\theta = s_r t/\text{radius}$, where the radius in our case is 800. That makes our first differential equation 1:

$$\frac{d\theta}{dt} = \frac{s_r}{800} \tag{1}$$

Hence, we can do the differential equations 2 and 3 for the coordinates of the rabbit as well:

$$\frac{dx}{dt} = -800\ cos(\theta(t))\frac{d\theta}{dt} \tag{2}$$

$$\frac{dy}{dt} = -800\ sin(\theta(t))\frac{d\theta}{dt} \tag{3}$$

The last differential equation needed for the rabbit is the distance traveled, which is denoted by d_r. Since, $distance = speed \times time \Leftrightarrow d_r = s_r \times t$, equation 4 is created:

$$\frac{d(d_r)}{dt} = s_r \tag{4}$$

Now, let us define the coordinates of the fox at time t as $(z(1), z(2))$ and do the according differential equations. Since the velocity is "the derivative of the position with respect to time" (AMSI, ESA, 2019, "Instantaneous velocity and speed"), the fox's velocity should be found. We know that:

$$\text{velocity} = \frac{\text{displacement between two points}}{\text{time needed to make the displacement}}$$

(W. Moebs, S. J. Ling, J. Sanny, 2016, "Average Velocity")

The distance between the fox and the rabbit is calculated by: dist$=\sqrt{(x - z(1))^2 + (y - z(2))^2}$. So the time needed is dist$/s_f$. The displacement between x and $z(1)$ is $x - z(1)$ and between y and $z(2)$ is $y - z(2)$. Having all this information, equations 5 and 6 can be generated:

$$\frac{dz(1)}{dt} = \frac{s_f(x - z(1))}{\text{dist}} \tag{5}$$

$$\frac{dz(2)}{dt} = \frac{s_f(y - z(2))}{\text{dist}} \tag{6}$$

Finally, if d_f denotes the distance traveled by the fox, then $d_f = s_f t$. Thus equation 7 is the last differential equation needed for this problem.

$$\frac{d(d_f)}{dt} = s_f \tag{7}$$

2) Code implementation

First, we should assign different labels to all the unknowns. Let us use the notation shown in Table **??**:

z(1)	z(2)	z(3)	z(4)	z(5)	z(6)	z(7)
x-coordinate of the fox	y-coordinate of the fox	distance traveled by the fox	x-coordinate of the rabbit	y-coordinate of the rabbit	angle θ at time t of the position of the rabbit	distance traveled by the rabbit

Table 1: Notation of each unknown in the differential equations

In order to solve the ODE in MATLAB, all these values need to be stored in an initial vector $z0$. Since we know that the fox starts from $[0, 0]$, the rabbit starts from $[0, 800]$, the distances traveled at time $= 0$ are 0's and the angle is 0 degrees it follows that the vector is as defined on line 13 in Figure 9. Note also that in Figure 10, on lines 10 to 18 we use the same differential equations as the ones described in 1) Problem solution with the only difference that in equations 5 and 6 the $x-$ and $y-$ coordinates of the rabbit are substituted with these of point G. The reason is that the motion of the fox from O to G needs to be considered as well. Similarly, the same method is applied on lines 41 to 49 in Figure 10 - the coordinates of point A substitute these of the rabbit.

3) Results

After running the code we get the output shown in Figure 3.

```
1 te =
2    69.8132
3 ze =
4    1.0e+03 *
5    -0.6789    0.4210    1.1868   -0.6928    0.4000    0.0010    0.8378
```

Figure 3: Output for constant speeds

As we can observe, the rabbit manages to escape, since its coordinates /the 4-th and the 5-th value on line 5 in Figure 3/ are the same as the coordinates of the burrow. This conclusion can be made, because the event function in Figure 11 ensures that the motion of the rabbit stops once it reaches the burrow. The coordinates of the fox are $[-678.9, 421]$. The distance traveled by the fox is 1186.8 m. The time when the rabbit escapes is $T = 69.8132$ seconds.

The plot we get is shown in Figure 4.

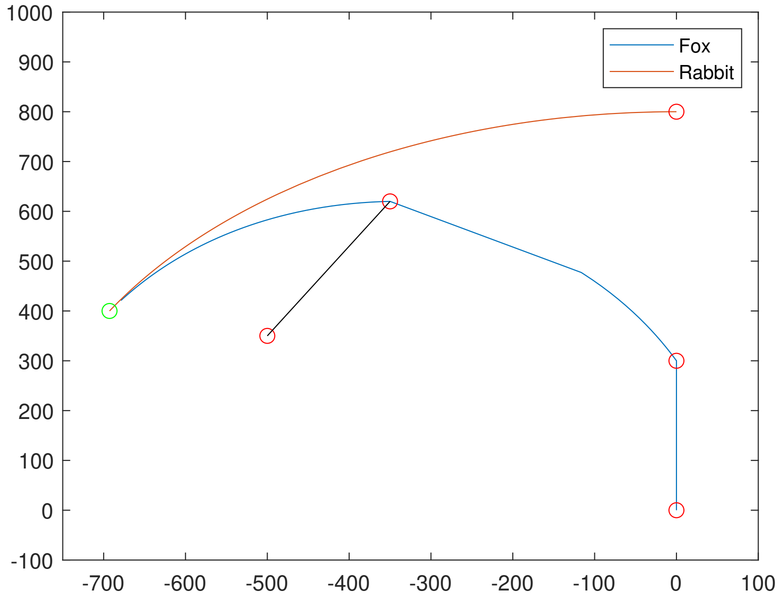

Figure 4: The plot of the motions of the rabbit and the fox when they have constant speeds

Question 2

1) Problem solution

In question 2, the speeds are diminishing. That means that they are decreasing over time, because, as in real life, both animals become more and more tired. They are given by:

$$s_f = s_{f0} \times e^{-\mu_f d_f(t)} \text{ and } s_r = s_{r0} \times e^{-\mu_r d_r(t)}$$

, where $\mu_f = 0.0002$ and $\mu_r = 0.0008$. Hence, they are functions of the traveled distances $z(3)$ and $z(7)$ we assigned in *2) Code implementation*.

2) Code implementation

In question 2 we use the same code as the one in question 1 with the only difference that lines 8 and 9 in Figure 9 are substituted with the code shown in Figure 5, where the distances traveled are 0's, since the diminishing speeds start from time $t = 0$.

```
sf0=17;
sr0=12;
mu_f=0.0002;
mu_r=0.0008;
s_f=sf0*exp(-mu_f*0);
s_r=sr0*exp(-mu_r*0);
```

Figure 5: MATLAB code for substituting the constant speeds with the diminishing speeds

and lines 5 and 6 in Figure 10 are substituted by the lines shown in Figure 6.

```
sf0=17;
sr0=12;
mu_f=0.0002;
mu_r=0.0008;
s_f=sf0*exp(-mu_f*z(3));
s_r=sr0*exp(-mu_r*z(7));
```

Figure 6: MATLAB code for substituting the constant speeds with the diminishing speeds

3) Results

After running the code from Figures 9, 10 and 11 with the changes shown in Figures 5 and 6, we get the output shown in Figure 7.

```
te =
   66.0497
ze =
   1.0e+03 *
   -0.5553   0.5759   1.0129  -0.5554   0.5758   0.0008   0.6138
```

Figure 7: MATLAB code for the output for diminishing speeds

Upon observation, the fox catches the rabbit in this scenario because the rabbit's coordinates /$[-555.4, 575.8]$/ have not yet reached the burrow's coordinates. The coordinates of the fox are $[-555.3, 575.9]$, the distance traveled by the predator is 1012.9 m. and the time needed for the rabbit to be caught is 66.0497 seconds.

The plot we get is shown in Figure 8.

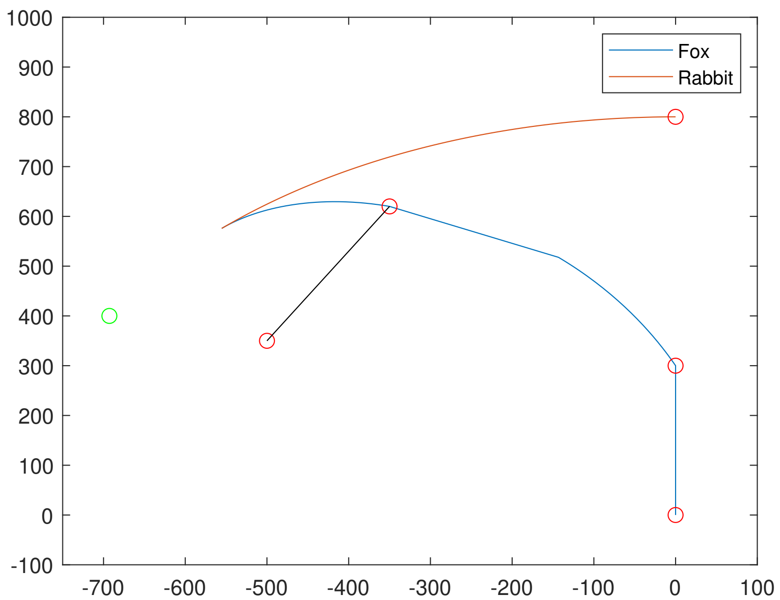

Figure 8: The plot of the motions of the rabbit and the fox when they have diminishing speeds

Appendix

Code Listings

[All the codes have been tested in MATLAB. Note that the codes in Figures 9, 10 and 11 should be run in a single m-file.]

```
A=[-350,620];%coordinates of A
G=[0,300];%coordinates of G
E = [-500 350];%coordinates of E
O = [0, 0];% coordinates of the starting point of the fox
R=[0,800];%coordinates of the starting point of the rabbit
x_burrow = 800*[-sin(pi/3)]; %x-coordinate of the burrow
y_burrow = 800*[cos(pi/3)]; %y-coordinate of the burrow
s_r = 12; %speed of the rabbit
s_f = 17; %speed of the fox
mindist = 0.1; %minimal distance for the fox to catch the rabbit
ts = [0 100]; %the time span needed for the ODE

z0 = [0 0 0 0 800 0 0]; %the inital vector for the ODE
options = odeset('RelTol', 1e-6, 'AbsTol', 1e-6, 'Events', @(t, z) foxirab(t, z, s_r, mindist, x_burrow));
[t, z, te, ze, zi] = ode45(@(t, z) fode(t, z, s_r, s_f), ts, z0, options);%solving the ODEs
te
ze
zi;

% Creating the plot
figure;
plot(z(:,1),z(:,2)) %plotting the motion of the fox
hold on;
plot(z(:,4),z(:,5)) %plotting the motion of the rabbit
hold on;
xlim([-750 100]) %limits of x-axis on the graph
ylim([-100 1000]) %limits of y-axis on the graph
plot(A(1), A(2), 'ro', 'MarkerSize', 7); %plotting point A
plot(O(1), O(2), 'ro', 'MarkerSize', 7); %plotting point O
plot(E(1), E(2), 'ro', 'MarkerSize', 7); %plotting point E
plot(G(1), G(2), 'ro', 'MarkerSize', 7); %plotting point G
plot(R(1), R(2), 'ro', 'MarkerSize', 7); %plotting point R
plot([A(1), E(1)], [A(2), E(2)], 'k-'); %plotting the fence AE
plot(x_burrow, y_burrow, 'go', 'MarkerSize', 7); %plotting the burrow
legend('Fox','Rabbit'), hold off %legend for the fox and the rabbit
```

Figure 9: MATLAB code for the solution and the plot

```matlab
%Creating the ODE function with all the differential equations
function dzdt = fode(t, z, s_r, s_f)
A=[-350,620];
G=[0,300];
s_r = 12;
s_f = 17;
z(4) = -800 * sin(z(6)); %x-coordinate of the rabbit at time t
z(5) = 800 * cos(z(6)); %y-coordinate of the rabbit at time t
if z(2)>=0 && z(2)<=300 %if the fox is at the starting point it moves directly
    towards the opening
    dist = sqrt((G(1) - z(1))^2 + (G(2) - z(2))^2); %distance between G and fox
    dzdt = zeros(7, 1); %creating a vector which will store 7 values
    dzdt(1) = s_f * (G(1) - z(1)) / dist;
    dzdt(2) = s_f * (G(2) - z(2)) / dist;
    dzdt(3) = s_f;
    dzdt(4) = -800 * cos(z(6)) * s_r / 800;
    dzdt(5) = -800 * sin(z(6)) * s_r / 800;
    dzdt(6) = s_r / 800;
    dzdt(7) = s_r;
elseif z(4)>A(1)% if the rabbit is not behind the fence the fox goes after it
    dist = sqrt((z(4) - z(1))^2 + (z(5) - z(2))^2); %distance between fox and rabbit
    dzdt = zeros(7, 1);
    dzdt(1) = s_f * (z(4) - z(1)) / dist;
    dzdt(2) = s_f * (z(5) - z(2)) / dist;
    dzdt(3) = s_f;
    dzdt(4) = -800 * cos(z(6)) * s_r / 800;
    dzdt(5) = -800 * sin(z(6)) * s_r / 800;
    dzdt(6) = s_r / 800;
    dzdt(7) = s_r;
else % if the rabbit is behind the fence the fox goes straight to A
    dist = sqrt((A(1) - z(1))^2 + (A(2) - z(2))^2); %distance between the fox and A
    dzdt = zeros(7, 1);
    dzdt(1) = s_f * (A(1) - z(1)) / dist;
    dzdt(2) = s_f * (A(2) - z(2)) / dist;
    dzdt(3) = s_f;
    dzdt(4) = -800 * cos(z(6)) * s_r / 800;
    dzdt(5) = -800 * sin(z(6)) * s_r / 800;
    dzdt(6) = s_r / 800;
    dzdt(7) = s_r;
end
    if z(1)-A(1)<=0.01 % if the fox reaches A it starts chasing the rabbit again
    dist = sqrt((z(4) - z(1))^2 + (z(5) - z(2))^2); %distance between fox and rabbit
    dzdt = zeros(7, 1);
    dzdt(1) = s_f * (z(4) - z(1)) / dist;
    dzdt(2) = s_f * (z(5) - z(2)) / dist;
    dzdt(3) = s_f;
    dzdt(4) = -800 * cos(z(6)) * s_r / 800;
    dzdt(5) = -800 * sin(z(6)) * s_r / 800;
    dzdt(6) = s_r / 800;
    dzdt(7) = s_r;
    end
end
```

Figure 10: MATLAB code for the differential equations

```
%Creating the event function
function [value, isterminal, direction] = foxirab(t, z, s_r, mindist, x_burrow)
    z(4) = -800 * sin(z(6));
    z(5) = 800 * cos(z(6));
    dist = sqrt((z(4) - z(1))^2 + (z(5) - z(2))^2);%distance between fox and rabbit
    x_burrow = 800 * [-sin(pi/3)];
    % Define the event conditions
    value(1) = dist - mindist; % Event occurs when the distance between
    %the rabbit and the fox is equal to mindist
    isterminal(1) = 1;
    direction(1) = -1;

    value(2)=z(4)-x_burrow; % Event occurs when the rabbit reaches the burrow
    isterminal(2)=1;
    direction(2)=-1;
end
```

Figure 11: MATLAB code for the events

References

Michael Lloyd, 2006-07 - "Pursuit Curves", "History"; p. 4, last accessed 20 November 2023, < https://pdf4pro.com/view/pursuit-curves-maa-sections-23f77d.html >.

AMSI, ESA, 2019 - "Instantaneous velocity and speed"; last accessed 20 November 2023, < https://amsi.org.au/ESA_Senior_Years/SeniorTopic3/3i/3i_2content_5.html >.

William Moebs, Samuel J. Ling, Jeff Sanny, Sep 19, 2016, "University Physics Volume 1", "Average Velocity", last accessed 20 November 2023, < https://openstax.org/books/university-physics-volume-1/pages/3-1-position-displacement-and-average-velocity >.