Project 2

November 25, 2022

1 General introduction

This report aimed to show approaches used in solving the problem of chasing. The problem would be described and simply analyzed in the `Problem introduction` section, the detailed analysis and the equation deriving process along with the logic of solving this problem is in the `Problem solution` section. The Matlab codes used to solve this problem would be partly presented and explained in the `Code implication` section, and the result of this project would be shown in the `Result` section.

2 Problem introduction

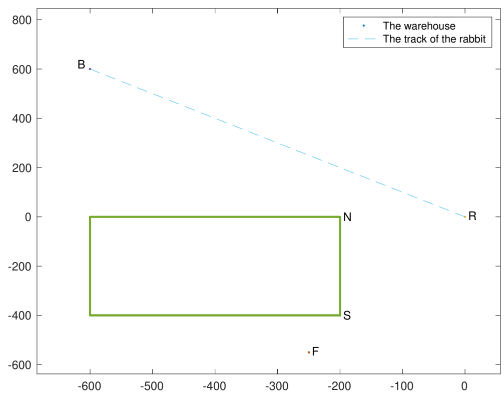

Figure 1: Initial Condition

As it is illustrated in figure 1, in the way of chasing the rabbit, the predator fox's sight might be blocked by a warehouse which is marked by green boxed lines in figure 1. The warehouse has two corners (S and N) on the right-hand side and the left-hand side of the warehouse is considered to be infinitely extended, coordinates of the two corners at the right-hand side are $S = [-200, -400]$ and $N = [-200, 0]$.
If the fox is blocked by corner S, the lower corner of the warehouse, the fox would run towards S until it can see the rabbit or the corner S is reached. If the fox's sight is blocked by N, the upper corner of the warehouse, the fox would run parallel to line N-S until it can see the rabbit. If the sight of the fox is not blocked, it would run straightly towards the rabbit.
At the same time, the rabbit is running straightly towards the burrow located at $[-600, 600]$, denoted by B, the track has been marked as the dashed line in figure 1, the chasing process would bring to an end

if the rabbit reaches the burrow before being caught by the fox, which means the rabbit has escaped. However, if during the chasing process, the distance between the rabbit and the fox is less than 0.1m, the rabbit is regarded as caught by the fox.
The initial position of fox and rabbit are $F = [-250, -550]$ and $R = [0, 0]$.
In question 1, the speed of the fox and the rabbit are constant, i.e. the acceleration is 0. The speed of fox is $s_f(0) = 16m/s$, and $s_r(0) = 13m/s$ for rabbit.
In question 2, the speed of the fox is dependent on the distance from the fox's initial position to its position at time t multiply a constant $\mu_f = 0.0002$. And this situation is the same for the rabbit while $\mu_r = 0.0008$. Generally, the speed of the fox and rabbit in question 2 would decrease as the distance they have travelled increases. This would be explained in the next section.
The required output includes the process end time T, the position of the fox when the process end as well as the distance travelled by the fox in time T. Additionally, plots showing the track of the fox and the rabbit should be presented.

3 Problem solution

In this section, the way to derive equations for required variables (the positions of the fox and the rabbit, and their speed for them) will be introduced first, and then the problem-solving logic would be explained in the second part. Note that the equations used to solve question 1 are almost the same as question 2(the only difference is the speed) and the logic of solving the two questions is identical.

3.1 Derive equations for variables

At the very beginning denotes the speed at time t as $s_f(t)$ for the fox and $s_r(t)$ for the rabbit, the distance that the fox has travelled as $d_f(t)$ and $d_r(t)$ for the rabbit. The position of the fox and rabbit at time t is denoted by $[x_f, y_f]$ and $[x_r, y_r]$.

3.1.1 The position of the fox at time t

By the definition of physics, the velocity is equal to the position that has changed to divide the time has changed. i.e. $V = \frac{dr}{dt}$ where r is the displacement contains the change in the x-axis and y-axis, i.e. $dr = (dx)^2 + (dy)^2$. Hence, the velocity can be rewrite as $\frac{dV}{dt} = \sqrt{(\frac{dx}{dt})^2 + (\frac{dy}{dt})^2}$ ①.
Then, note that if the fox is heading towards a thing, then the tangent line at the end of its trajectory at time t will point to the thing it is running towards. Based on this, suppose the thing that the fox running towards has coordinates $[a, b]$ at time t, we can come up with: $\begin{pmatrix} \frac{d(x_f)}{dt} \\ \frac{d(y_f)}{dt} \end{pmatrix} = s_f(t) * \begin{pmatrix} a - x_f \\ b - y_f \end{pmatrix}$,②.
By combining equation ① and ②, we can come up with equations to determine the coordinates of the fox at time t: $\begin{cases} \frac{d(x_f)}{dt} = \frac{(s_f(t)*(a-x_f))}{\sqrt{((a-x_f)^2+(b-y_f)^2)}}, \\ \frac{d(y_f)}{dt} = \frac{(s_f(t)*(b-y_f))}{\sqrt{((a-x_f)^2+(b-y_f)^2)}}, \end{cases}$ ③.

3.1.2 The position of the rabbit at time t

Next, equations to calculate coordinates at time t of the rabbit would be determined. First, note that the track of the rabbit is a straight line from $[0, 0]$ to $[-600, 600]$ if it is not caught by the fox. By the given information, the slope of the rabbit's track is $\frac{y_B - y_r}{x_b - x_r} = -1$, so the angle between the track line and the x-axis is 135°. Additionally, knowing that velocity is equal to the speed multiply the movement direction, and the direction can be calculated by $cos(135°)$ for the x-coordinate and $sin(135°)$ for the y-coordinate, we are able to write down the position calculating equation for the rabbit. The derivative of the current position along the x-axis and y-axis against time t is:

$$\begin{cases} \frac{d(x_r)}{dt} = -\sqrt{\frac{1}{2}} * s_r(t), \\ \frac{d(y_r)}{dt} = \sqrt{\frac{1}{2}} * s_r(t), \end{cases} \quad ④.$$

3.1.3 The speed of the fox and the rabbit

From the equations listed above, it can be found that in order to figure out the current position of the fox and rabbit, the speed of the fox and rabbit should be determined.

Clearly, changes in speed are continuous for question 2, since it depends on the distance that has been travelled.

According to the definition of physics, speed can be obtained by the derivative of distance against time, i.e. $\begin{cases} \frac{d(d_f(t))}{dt} = s_f(t) = s_f(0) * e^{-\mu_f * d_f(t)}, \\ \frac{d(d_r(t))}{dt} = s_r(t) = s_r(0) * e^{-\mu_r * d_r(t)}, \end{cases}$ ⑤ where $\mu_f = 0.0002$ and $\mu_r = 0.0008$.

And for question 1, the speed is constant, which means the speed is not dependent on the distance ever travelled. So $\begin{cases} s_f(t) = s_f(0), \\ s_r(t) = s_r(0), \end{cases}$ ⑥ .

3.2 The logic of solving this problem

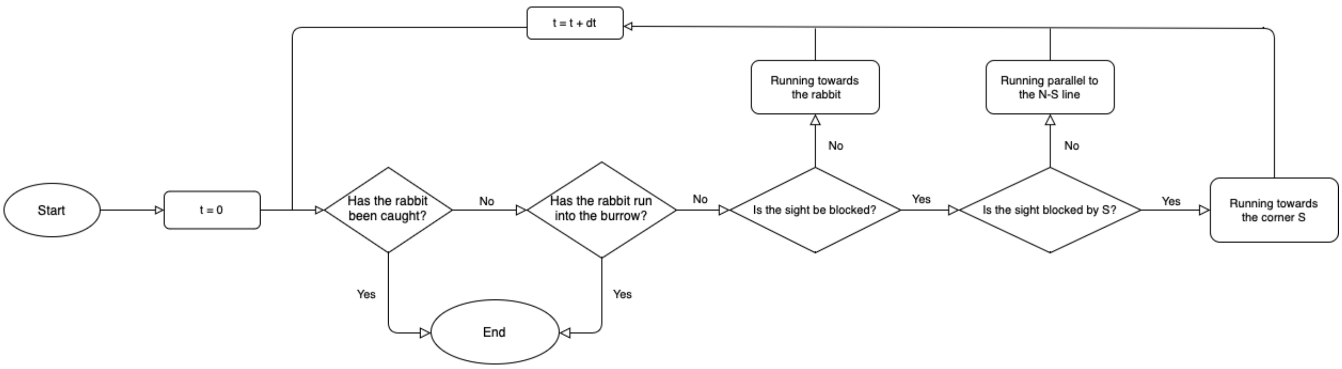

Figure 2: Flowchart

This flowchart showed the basic logic used in solving this question and will be explained in this section.

3.2.1 Process ending conditions

Figure 2 illustrated that under two conditions, the process would be ended: the rabbit is caught or the rabbit reaches the burrow. Next, I would like to convert these two conditions into mathematical language. For the first condition, the rabbit is caught means the distance between the rabbit and the fox is less than 0.1m during the chasing process, i.e. $\sqrt{(x_f - x_r)^2 + (y_f - y_r)^2} < 0.1$.

For the second condition, the rabbit reaches the burrow means the rabbit shares the same coordinate with burrow B, i.e. $x_r = x_B$ and $y_r = y_B$. Since the x and y coordinate of the rabbit is associated, either $x_r = x_B$ or $y_r = y_B$ can ensure the rabbit reaches the burrow.

3.2.2 Sight blocking conditions

In this section, two judgements: '`Is the sight be blocked?`' and '`Is the sight be blocked by S?`' in the flowchart(figure 2) would be explained.

Generally, during the chasing process, there are two basic cases: the rabbit is in sight and the rabbit is not in sight.

First, define the intersection point of the line FR which connects the Fox and the Rabbit with the line N-S or its extensions as p.

Note that, if the intersection point p is located at the extension line of line N-S, that means the rabbit can be seen by the fox since the sight line FR is not intersected with the N-S segment line. If p is located at the segment of line N-S, the sight is blocked. Hence, by checking where the y-coordinate of the line connecting the fox and the rabbit intersect with the line N-S or its extensions, we can find whether the sight has been blocked.

Knowing the coordinates of the fox and the rabbit are $[x_f, y_f]$ and $[x_r, y_r]$, the line FR that connects the fox and the rabbit can be determined: $FR = \frac{y_f - y_r}{x_f - x_r} * x + y_f - \frac{y_f - y_r}{x_f - x_r} * x_f$. And since the intersection point, p is located on the N-S line or its extension, p has the same x-coordinates as N and S, i.e. $x_p = x_S = x_N = -200$. Then the y-coordinates of p which is the main variable used to judge whether the sight is blocked can be calculated by: $y_p = \frac{y_f - y_r}{x_f - x_r} * (-200) + y_f - \frac{y_f - y_r}{x_f - x_r} * x_f$ 7.

After finishing the set a variable used to judge whether the rabbit is in sight, determining which corner blocked the sight is the next task, so we should subdivide the 'sight is blocked' case into two cases: the sight is blocked by S and the sight is blocked by N.

It is worth mentioning that the fox can only be blocked by one corner at any time. Before the fox is able to see corner N, the sight can only be blocked by corner S, after seeing corner N, corner S would never block the sight again. The reason is that at the beginning the rabbit runs from the right-hand side, so the fox should move to the right to catch the rabbit, i.e. x-coordinate increases, if the x-coordinate of the fox is larger than S, the blocked corner is regarded as passed, the fox can see all things on the right-hand side of line N-S, hence would not be affected by the corner S anymore, the only possible blocking corner is N now.

Defining a point q as the intersection point of line NF and $y = y_S = -400$. If $x_q < x_S$, the sight is blocked by S, otherwise the sight is blocked by N. The formula of line FN is that $y = \frac{y_f - N(2)}{x_f - N(1)} * x + y_f - \frac{y_f - N(2)}{xf - N(1)} * x_f$.

The x-coordinate of q is the most crucial value here, the formula for it is $x_q = \frac{y_q - y_f + \frac{y_f - N(2)}{x_f - N(1)} * x_f}{\frac{y_f - N(2)}{x_f - N(1)}}$, let y have the same y-coordinate as S, i.e. $y_q = y_S = -400$, the formula becomes $x_q = \frac{-400 - y_f + \frac{y_f - N(2)}{x_f - N(1)} * x_f}{\frac{y_f - N(2)}{x_f - N(1)}}$ 8.

It is possible to determine which of the three cases was entered, by calculating and comparing the values of the variables with the coordinate of S and N. All three possible cases are: the sight is blocked by S, the sight is blocked by N, and the sight is not blocked. Next, I would like to explain under what situation, each case would be triggered and the track of the fox in each case.

Sight blocked by S

If the y-coordinate of intersection point p is larger than S and smaller than N while the x-coordinate of intersection point q is smaller than S, i.e. $y_S < y_p < y_N$ and $x_q < x_S$ this fox is regarded as blocked by corner S. $y_S < y_p < y_N$ implies the sight is blocked, $x_q \le x_S$ implies the corner S is the blocking corner. In this case, the fox would run towards point S. According to 3 and S's coordinates, the position of the fox at time t is: $\begin{cases} \frac{d(x_f)}{dt} = \frac{(s_f(t) * (x_S - x_f))}{\sqrt{((x_S - x_f)^2 + (y_S - y_f)^2)}}, \\ \frac{d(y_f)}{dt} = \frac{(s_f(t) * (y_S - y_f))}{\sqrt{((x_S - x_f)^2 + (y_S - y_f)^2)}}, \end{cases}$.

Sight blocked by N

If the y-coordinate of intersection point p is larger than S and smaller than N while the x-coordinate of intersection point q is larger than S, i.e. $y_S < y_p < y_N$ and $x_q \geq x_S$ the fox is regarded as blocked by corner N. $y_S < y_p < y_N$ implies the sight is blocked, $x_q \geq x_S$ implies the corner N is the blocking corner. In this case, the fox would run parallel to line N-S. According to ③ and the information of line N-S ($x = -200$), the position of fox at time t is: $\begin{cases} \frac{d(x_f)}{dt} &= 0 , \\ \frac{d(y_f)}{dt} &= s_f(t), \end{cases}$.

Sight is not blocked

If the y-coordinate of intersection point p is smaller than S or larger than N, i.e. $y_p \leq y_S$ or $y_p \geq y_N$, the sight is regarded as not being blocked. In this case, the fox would run towards the rabbit. According to ③ and ④, the position of fox at time t is: $\begin{cases} \frac{d(x_f)}{dt} &= \frac{(s_f(t)*(x_r-x_f))}{\sqrt{((x_r-x_f)^2+(y_r-y_f)^2)}}, \\ \frac{d(y_f)}{dt} &= \frac{(s_f(t)*(y_r-y_f))}{\sqrt{((x_r-x_f)^2+(y_r-y_f)^2)}}, \end{cases}$.

3.2.3 General logic explaination

After explaining two blocks in the flowchart, it is time to make a general explanation for the flowchart(figure 2).

As figure 2 illustrated, the time t is set to be 0 at the beginning, once the four judgements explained in the last two sections are finished, we would enter one of the three cases listed in 3.1.3, and solve corresponding ODE functions for time t, after this, the round is over, we would process to the next stage and the time t would plus dt (a fairly small value), then redo the whole process until one of the ending condition satisfied.

4 Code implecation

At this stage, the logic of the solution has been completely explained. The codes listed below are for question 2 some interpretations would be made regarding codes. The way to solve question one would be explained in the last part of this section.

4.1 ODEs for Q2

Listing 1: ODEs for Q2

```
function dzdt = cases_Q2(t,z)
    xf=z(1);yf=z(2);yr=z(3);df=z(4);dr=z(5);
    sf0=16;sr0=13;muf=0.0002;mur = 0.0008;S=[-200,-400];N=[-200,0];
    k = (yr-yf)/(-yr-xf);%slope of the line connecting two animals
    b = yf-k*xf;%The intercept distance of the line connecting two animals
    k1 = (yf-N(2))/(xf-N(1));%slope of the line connecting the fox and N
    b1= yf-k1*xf;%The intercept distance of the line connecting the fox and N
if k*S(1)+b>=S(2) && k*N(1)+b<=N(2) && S(1)>(S(2)-b1)/k1 %Blocked by S
    dzdt(1,1)=(sf0*exp(-muf*df)*(S(1)-xf))/sqrt((S(1)-xf)^2+(S(2)-yf)^2);%The fox's x
    dzdt(2,1)=(sf0*exp(-muf*df)*(S(2)-yf))/sqrt((S(1)-xf)^2+(S(2)-yf)^2);%The fox's y
    dzdt(3,1)=sqrt(1/2)*sr0*exp(-mur*dr);%The rabbit's y
    dzdt(4,1)=sf0*exp(-muf*df);%Travelled distance of the fox
    dzdt(5,1)=sr0*exp(-mur*dr);%Travelled distance of the rabbit
```

```
elseif k*S(1)+b>=S(2) && k*N(1)+b<=N(2) && S(1)<=(S(2)-b1)/k1 %Blocked by N
    dzdt(1,1)=0;
    dzdt(2,1)=sf0*exp(-muf*df);
    dzdt(3,1)=sqrt(1/2)*sr0*exp(-mur*dr);
    dzdt(4,1)=sf0*exp(-muf*df);
    dzdt(5,1)=sr0*exp(-mur*dr);
else %Not blocked
    dzdt(1,1)=(sf0*exp(-muf*df)*(-yr-xf))/sqrt((-yr-xf)^2+(yr-yf)^2);
    dzdt(2,1)=(sf0*exp(-muf*df)*(yr-yf))/sqrt((-yr-xf)^2+(yr-yf)^2);
    dzdt(3,1)=sqrt(1/2)*sr0*exp(-mur*dr);
    dzdt(4,1)=sf0*exp(-muf*df);
    dzdt(5,1)=sr0*exp(-mur*dr);
end
```

This contains all the ode functions of this problem. The column vector z contains five variables, the xf and yf are the coordinates of the fox, yr is the y-coordinate of the rabbit, df and dr are the distance travelled by the fox and by the rabbit. Since the rabbit is running on the line $y = -1 * x$, the x-coordinate can be easily computed by multiplying -1 by the y-coordinate, so it is not listed here. All the ODEs have been discussed in part 3.1. Variable k and b are related to 7 and variables $k1$ and $b1$ are related to 8, the judgement containing k and b determines whether the sight is blocked, and the judgement containing $k1$ and $b1$ determines the sight is blocked by which corner, the detailed explanation is in 3.2.2.
Note that the last lines in each 'if statement' are formulae of speed which have been explained in section 3.1.3, the equation is 5.

4.2 Events

Listing 2: Events

```
1 function [value,isterminal,direction] = escapeorcaught(t,z)
2 value(1)=(z(1)+z(3))^2+(z(2)-z(3))^2-0.01;%The distance between two animals <=0.1
3 isterminal(1)= 1;
4 direction(1) = -1;
5 value(2)=z(3)-600;%The rabbit reaches the burrow
6 isterminal(2)= 1;
7 direction(2)= 1;
8 end
```

This is an event function containing two events that can bring the process to the end (discussed in section 3.2.1). $Value(1)$ calculates the distance between two animals minus 0.1 which is one of the stopping conditions, line 3 means if $Value(1) = 0$, i.e. the rabbit was caught, the solving of ode would end, and line 4 means capture the event 1 if $Value(1)$ is from the decreasing direction. $Value(2)$ would equal zero if the rabbit reached the burrow, line 7 means capture the event 2 if $Value(2)$ is from the increasing direction.

4.3 Main of Q2

Listing 3: Main of Q2

```
1 S = [-200,-400];N = [-200,0];
2 B = [-600,600];sf0 = 16;sr0 = 13;
3 xf=-250;yf=-550;df=0;yr=0;dr=0;t=0;
4 z0=[xf(end);yf(end);yr(end);df(end);dr(end)];
5 %Give initial value to ODE
6 tSpan=[t(end),100];
7 options = odeset('Events', @escapeorcaught,'MaxStep',1e-3);
8 %Stop the process when the Rabbit reach the hole.
9 [t,z,te,ze,yi] = ode45(@(t,z) cases_Q2(t,z),tSpan,z0,options);
10 if yi==2
11     disp(['The rabbit has escaped, the position of the fox is ','[',num2str(ze(1)),',',
        num2str(ze(2)),']',' when the time is ',num2str(te),'s',',',' the travelled
        distance is ',num2str(ze(4))])
12 elseif yi==1
13     disp(['The fox caught the rabbit at ','[',num2str(ze(1)),',',num2str(ze(2)),']','
        when the time is ',num2str(te),'s',', the travelled distance is ',num2str(ze(4))
        ])
14 end
15 x=[-200,-200,-600,-600,-200,-600];
16 y=[-400,0,0,-400,-400,-400];
17 plot(z(:,1),z(:,2),-z(:,3),z(:,3),x,y,'LineWidth',2),legend('The track of the Fox','The
    track of the Rabbit','The warehouse')
18 hold on
19 text(-200,-400,' S');
20 text(-200,0,' N');
```

The first three lines are just assigning values to variables. Line 4 is assigning initial values to 'ODEs for Q2', the order is the same as the order in variable z and line 6 sets the time span for the ODE function, the second value in the tSpan means that stop solving ODEs if $t = 100$, these two lines are important in solving the ODE function. Note that the ending time is not exactly the process ending time, the chosen time span is longer than the expected time for the result that the rabbit escaped. The choice of a longer time span would not affect the result, since the Event function contained in options in line 7 would stop the process at the proper time. And the $MaxStep$ in line 7 is used to adjust the precision of the ode45. Line 9 is about solving ODEs, t and z are containing all data from the beginning to the time when the process end, while te contains the time when the event trigged and ze contains values of variables in z when the event trigged. yi contains the order of the event that has been trigged. Line 10 to line 14 is an 'if statement' used to judge whether the rabbit is caught, if event 1 in the function `Events` is trigged, i.e. $yi = 1$, the rabbit is caught, if event 2 is trigged, i.e. $yi = 2$, the rabbit escaped. Additionally, this 'if statement' can output the required data. Lines from 15 to 20 are used to plot the trajectory map.

4.4 Modification for Question 1

To solve question 1, we just need to set $\mu_f = \mu_r = 0$ in 'ODE for Q2', then the speed of the fox and the rabbit would be the same as the equation ⑥. Rename the function as `case_Q1` and substitute `case_Q2` in the `Main of Q2`, line 9 to `case_Q1`, keeping the remaining part the same as before.

5 Result

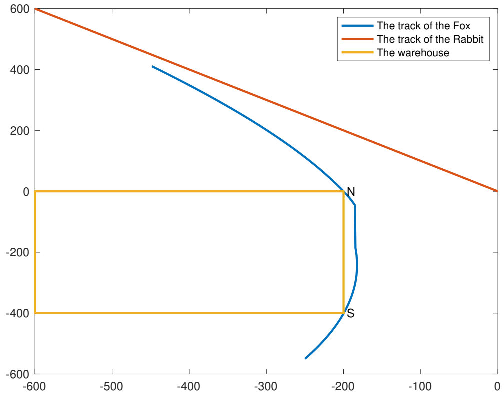

Figure 3: Trajectory map of Question 1

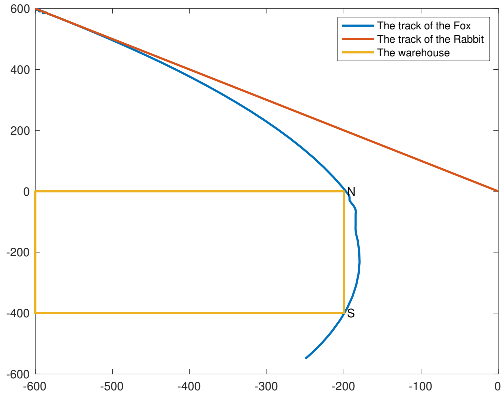

Figure 4: Trajectory map of Question 2

```
>> RabbitandFox_Q1
The rabbit has escaped, the position of the fox is [-448.3938,410.7335] when the time is 65.2714s, the travelled distance is 1044.3423
```

Figure 5: Result of Question 1

```
>> RabbitandFox_Q2
The fox caught the rabbit at [-586.0393,586.0393] when the time is 90.463s, the travelled distance is 1271.2008
```

Figure 6: Result of Question 2

Figure 3 and figure 5 illustrated that the rabbit has escaped for question 1, and in the way of chasing, the fox is only blocked by corner N.
Figure 4 and figure 6 illustrated that for question 2, the rabbit is caught, this may be because the speed of the rabbit decreases faster than the fox. Additionally, the fox is only blocked by the corner N.