Non-Trivial Evasion

Taylor Gledhill, Joseph Wilkes, Raelynn Wonnacott, Nathan Nelson

April 2021

Abstract

To expand upon our exploration of pursuit and evasion dynamics from last semester we utilized optimal control to experiment with improved evasion techniques for a rabbit being chased by a fox. First, we employed an LQR approach which for initial values of t appeared to give intuitive results. However, as we examined greater values of t we saw that the LQR approach was insufficient potentially due to an infinite time horizon setup and that linearization lead to decreasing time steps in the solution. We then utilized Pontraygin's Maximum principle both with fixed end time and free end time. We also included a non-integral component in the objective function to incentive the rabbit to reach the rabbit hole at the origin. With specific hyper-parameters the infinite time scenario of the maximum principle yielded satisfactory results.

1 Background

Last semester, we explored pursuit and evasion dynamics with varying situations such as when the evader has a goal destination, when the pursuer predicts the path of the evader, and various evasion techniques of the evader.

The idea to model pursuit curves is nothing new, with published works describing the concept dating as early as the eighteenth century [1], but we seek to put a new spin on the problem by focusing on Optimal Control so we can apply concepts we learn this semester in order to choose the optimal path for the rabbit.

2 Mathematical Representation

We let $\boldsymbol{r}$ and $\boldsymbol{f}$ be the two-dimensional position of the rabbit and fox, respectively. Acknowledging that these are both part of the state of our problem, we let our state $\boldsymbol{x}$ be given by $\boldsymbol{x} = \begin{bmatrix}\boldsymbol{r} & \boldsymbol{f}\end{bmatrix}^T = \begin{bmatrix}x_{\boldsymbol{r}} & y_{\boldsymbol{r}} & x_{\boldsymbol{f}} & y_{\boldsymbol{f}}\end{bmatrix}^T$.

Because the rabbit travels at a constant speed, the only thing we have control over is the angle θ at which it should travel, and so we let our control variable $\boldsymbol{u}$ be given by $\boldsymbol{u} = \begin{bmatrix}\cos\theta & \sin\theta\end{bmatrix}^T$.

We seek to minimize the cost functional

$$J[\boldsymbol{u}] = \int_0^{t_f} \left[s_{\boldsymbol{r}} + \|\boldsymbol{r}\|^2 - \beta \|\boldsymbol{r} - \boldsymbol{f}\|^2\right] dt, \tag{1}$$

where t_f is sometimes free and sometimes fixed, β is a constant, the $s_{\boldsymbol{r}}$ term exists in order to penalize longer paths to the goal, the $\|\boldsymbol{r}\|^2$ term exists in order to encourage the rabbit to return to its home (located at the origin), and the $\beta \|\boldsymbol{r} - \boldsymbol{f}\|^2$ term exists in order to penalize the rabbit getting to close to the fox.

The associated state equation is given by

$$\boldsymbol{x'} = \begin{bmatrix} s_{\boldsymbol{r}} \cos\theta \\ s_{\boldsymbol{r}} \sin\theta \\ s_{\boldsymbol{f}} \frac{x_{\boldsymbol{r}} - x_{\boldsymbol{f}}}{\|\boldsymbol{f} - \boldsymbol{r}\|_2} \\ s_{\boldsymbol{f}} \frac{y_{\boldsymbol{r}} - y_{\boldsymbol{f}}}{\|\boldsymbol{f} - \boldsymbol{r}\|_2} \end{bmatrix}, \tag{2}$$

where $s_{\boldsymbol{r}}$ and $s_{\boldsymbol{f}}$ are constants that determine the speed of the rabbit and fox, respectively. The first two terms represent the rabbit traveling in the direction determined by the control variable, and the last two terms represent the fox traveling toward the rabbit's current position.

The boundary conditions for our problem are given by

$$\boldsymbol{r}(0) = r_0, \quad \boldsymbol{f}(0) = f_0, \quad \boldsymbol{r}(t_f) = \boldsymbol{0}. \tag{3}$$

3 Solution

3.1 LQR

Note we formatted our initial solution as an LQR problem. The cost functional was able to be written in LQR format as shown below.

$$J[\theta] = \int_0^{t_f} \left(\begin{bmatrix} x_{\boldsymbol{r}} \\ y_{\boldsymbol{r}} \\ x_{\boldsymbol{f}} \\ y_{\boldsymbol{f}} \end{bmatrix}^T \begin{bmatrix} 1-\beta & 0 & \beta & 0 \\ 0 & 1-\beta & 0 & \beta \\ \beta & 0 & -\beta & 0 \\ 0 & \beta & 0 & -\beta \end{bmatrix} \begin{bmatrix} x_{\boldsymbol{r}} \\ y_{\boldsymbol{r}} \\ x_{\boldsymbol{f}} \\ y_{\boldsymbol{f}} \end{bmatrix} + \begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix}^T \begin{bmatrix} s_{\boldsymbol{r}} & 0 \\ 0 & s_{\boldsymbol{r}} \end{bmatrix} \begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix} \right) dt$$

With β chosen to ensure that the matrix is positive semi-definite.

However, in order to ensure our state equation was linear we found it necessary to remove the normalizing terms. This enabled us to get a solution. Our linear state equation is shown below, with $\boldsymbol{x}' = A\boldsymbol{x} + B\boldsymbol{u}$.

$$\boldsymbol{x}' = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ s_{\boldsymbol{f}} & 0 & -s_{\boldsymbol{f}} & 0 \\ 0 & s_{\boldsymbol{f}} & 0 & -s_{\boldsymbol{f}} \end{bmatrix} \begin{bmatrix} x_{\boldsymbol{r}} \\ y_{\boldsymbol{r}} \\ x_{\boldsymbol{f}} \\ y_{\boldsymbol{f}} \end{bmatrix} + \begin{bmatrix} s_{\boldsymbol{r}} & 0 \\ 0 & s_{\boldsymbol{r}} \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix} = \begin{bmatrix} s_{\boldsymbol{r}} \cos\theta \\ s_{\boldsymbol{r}} \sin\theta \\ s_{\boldsymbol{f}}(x_{\boldsymbol{r}} - x_{\boldsymbol{f}}) \\ s_{\boldsymbol{f}}(y_{\boldsymbol{r}} - y_{\boldsymbol{f}}) \end{bmatrix}$$

In order to solve our LQR problem we used the algebraic Riccati equation.

3.2 Pontryagin's Maximum Principle: Fixed End Time

In order to enable us to include our normalizing constants in our state equations as well as have increased control over final time conditions and penalties, we decided to try another approach using Pontryagin's maximum principle.

The cost function and conditions are as follows:

$$\min_u J[u] = \int_0^{t_f} \left[-\beta(x_r - x_f)^2 - \beta(y_r - y_f)^2 + sr + x_r^2 + y_r^2\right] dt$$

$$\boldsymbol{x}' = \begin{bmatrix} x_r \\ y_r \\ x_f \\ y_f \end{bmatrix}' = \begin{bmatrix} s_r \cos(u) \\ s_r \sin(u) \\ s_f \frac{x_r - x_f}{\sqrt{(x_r - x_f)^2 + (y_r - y_f)^2}} \\ s_f \frac{y_r - y_f}{\sqrt{(x_r - x_f)^2 + (y_r - y_f)^2}} \end{bmatrix}, \qquad \boldsymbol{x}(0) = \boldsymbol{x}_0.$$

We formulated the Hamiltonian (with a four-dimensional costate) and took appropriate partial derivatives to get the following equations.

$$\begin{aligned} H &= -L + \boldsymbol{p} \cdot \boldsymbol{f} \\ &= \beta(x_r - x_f)^2 + \beta(y_r - y_f)^2 - sr - x_r^2 - y_r^2 + p_1 s_r \cos(u) + p_2 s_r \sin(u) \\ &\quad + p_3 s_f \frac{x_r - x_f}{\sqrt{(x_r - x_f)^2 + (y_r - y_f)^2}} + p_4 s_f \frac{y_r - y_f}{\sqrt{(x_r - x_f)^2 + (y_r - y_f)^2}}. \end{aligned}$$

$$\begin{aligned} p_1' = -\frac{DH}{dx_r} &= -\beta\left(-2x_f + 2x_r\right) + 2x_r - \frac{p_3 s_f}{\left((x_r - x_f)^2 + (y_r - y_f)^2\right)^{0.5}} \\ &\quad - \frac{p_3 s_f\,(x_r - x_f)\,(x_f - x_r) + p_4 s_f\,(x_f - x_r)\,(y_r - y_f)}{\left((x_r - x_f)^2 + (y_r - y_f)^2\right)^{1.5}}, \end{aligned}$$

$$p_2' = -\frac{DH}{dy_r} = -\beta\left(-2y_f + 2y_r\right) + 2y_r - \frac{p_4 s_f}{\left(\left(x_r - x_f\right)^2 + \left(y_r - y_f\right)^2\right)^{0.5}}$$
$$- \frac{p_3 s_f \left(x_r - x_f\right)\left(y_f - y_r\right) + p_4 s_f \left(y_r - y_f\right)\left(y_f - y_r\right)}{\left(\left(x_r - x_f\right)^2 + \left(y_r - y_f\right)^2\right)^{1.5}},$$

$$p_3' = -\frac{DH}{dx_f} = -\beta\left(2x_f - 2x_r\right) + \frac{p_3 s_f}{\left(\left(x_r - x_f\right)^2 + \left(y_r - y_f\right)^2\right)^{0.5}}$$
$$- \frac{p_3 s_f \left(x_r - x_f\right)\left(x_r - x_f\right) + p_4 s_f \left(x_r - x_f\right)\left(y_r - y_f\right)}{\left(\left(x_r - x_f\right)^2 + \left(y_r - y_f\right)^2\right)^{1.5}},$$

$$p_4' = -\frac{DH}{dy_f} = -\beta\left(2y_f - 2y_r\right) + \frac{p_4 s_f}{\left(\left(x_r - x_f\right)^2 + \left(y_r - y_f\right)^2\right)^{0.5}}$$
$$- \frac{p_3 s_f \left(x_r - x_f\right)\left(y_r - y_f\right) + p_4 s_f \left(y_r - y_f\right)\left(y_r - y_f\right)}{\left(\left(x_r - x_f\right)^2 + \left(y_r - y_f\right)^2\right)^{1.5}}.$$

With the boundary constraints of $p_i(t_f) = 0$ for all $i \in \{1, 2, 3, 4\}$. However, this did not result in the rabbit ending at its burrow, so the endpoint constraints were updated to $p_1(t_f) = 2mx_r(t_f)$, $p_2(t_f) = 2my_r(t_f)$, and $p_3(t_f) = p_4(t_f) = 0$, where m was a variable that allowed us to alter the cost of returning to the burrow. This altered the original cost function by adding $\phi(t_f) = m(x_r(t_f)^2 + y_r(t_f)^2)$ to $J[u]$.

We then solved for the optimal control solution $\tilde{u}$, which gave us two potential solutions

$$\tilde{u} \in \left\{-2\arctan\left(\frac{p_1 - \sqrt{p_1^2 + p_2^2}}{p_2}\right), -2\arctan\left(\frac{p_1 + \sqrt{p_1^2 + p_2^2}}{p_2}\right)\right\}.$$

After testing both options, we found that the second option occasionally provided better results, as can be seen in Figure 1.

3.3 Pontryagin's Maximum Principle: Free End Time

This problem is more realistically modeled with a free end time, because the rabbit is interested in getting to its hole in the safest way possible, not just in getting to its hole at a specific time.

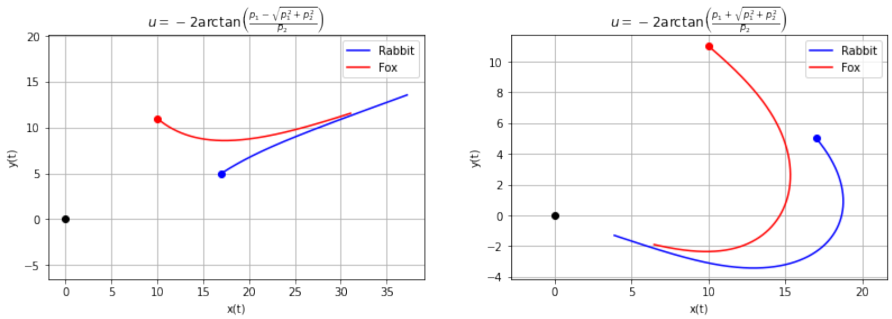

Figure 1: The two plots above show the difference between the two values of $\tilde{u}$. The plot to the left is the $\tilde{u}$ listed first, while the plot on the right is the second value of $\tilde{u}$. This provides the solution we would expect to see.

Letting t_f be free introduces another endpoint constraint that $H(t_f) = 0$, i.e.

$$\begin{aligned} 0 = H(t_f) = \beta(x_r(t_f) - x_f(t_f))^2 &+ \beta(y_r(t_f) - y_f(t_f))^2 - sr \\ &- x_r(t_f)^2 - y_r(t_f)^2 + p_1(t_f)s_r\cos(u(t_f)) + p_2(t_f)s_r\sin(u(t_f)) \\ &+ p_3(t_f)s_f\frac{x_r(t_f) - x_f(t_f)}{\sqrt{(x_r(t_f) - x_f(t_f))^2 + (y_r(t_f) - y_f(t_f))^2}} \\ &+ p_4(t_f)s_f\frac{y_r(t_f) - y_f(t_f)}{\sqrt{(x_r(t_f) - x_f(t_f))^2 + (y_r(t_f) - y_f(t_f))^2}}. \end{aligned}$$

In order to ensure that our optimal control solution would work for this formulation, we tried solutions for both values of $\tilde{u}$ (see Figure 2) and found that $\tilde{u}$ did not need to be changed from the fixed time formulation.

Using this constraint and the techniques discussed in class, we were able to attempt to numerically solve for optimal end time in different situations.

4 Interpretation

4.1 LQR

In general the LQR model gave us very reasonable solutions. Figure 3 shows the initial solution calculated using the LQR approach. This figure was calculated with $s_r = s_f = .5, \beta = .9$ and $t_f = 100$ We evaluated our

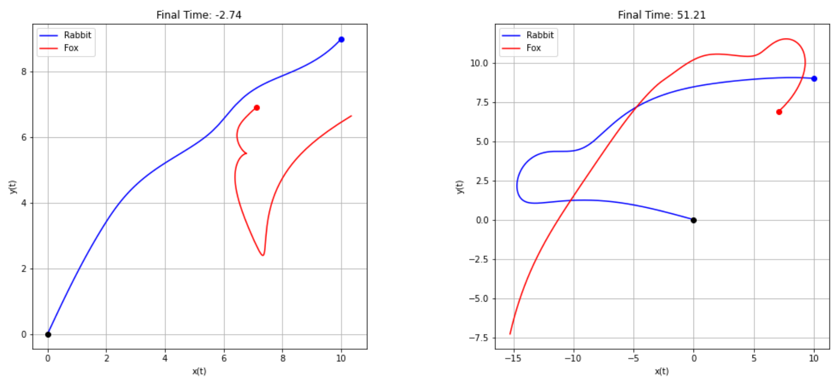

Figure 2: We can see the unfavorable solution in the left visual in that it yields a negative time (listed as the title of the subplot). Note that $\beta = 0$ for both of these cases.

solution at 500 time steps. For initial values of t, the rabbit was successful at evading the fox and the fox moved towards the rabbit.

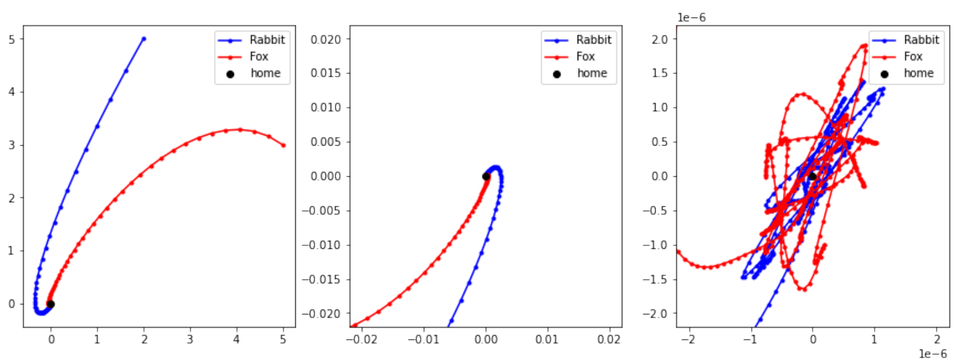

Figure 3: Time steps for initial LQR solution.

However, there are two main problems with this model. First, we can see that the time steps get smaller and eventually restrict the movement of both the rabbit and the fox. The removal of normalizing terms in our state equation is a potential cause. Another possible explanation is that our LQR solution is infinite time horizon. A possible fix for this would be to penalize a large end time. If the rabbit has an infinite amount of time to reach home our model may not care if the rabbit is moving very slowly as long as movement is in the correct direction.

The second problem that we see in this model again occurs near the end time. Here we see erratic behavior from both the rabbit and the fox. Because of these issues near our final time, we put constraints on the end time in our next model.

4.2 Pontryagin's Maximum Principle: Fixed End Time

In this scenario we included the following term in the cost functional: $\phi = m\left(x_r(t_f)^2 + y_r(t_f)^2\right)$ where m is a parameter that incentivizes the rabbit to reach the origin (the rabbit hole) and final destination where it would be safe from the fox. With a fixed end time we found that the rabbit took longer paths than necessary because the rabbit wants to reach it's burrow as close to t_f as possible, but not before, as seen in Figure 4. We saw that for small values of m that the rabbit was more focused on fleeing from the fox rather than trying to reach the rabbit hole. When we dramatically increased m then the rabbit would take more risks of being near the fox in order to reach it's burrow.

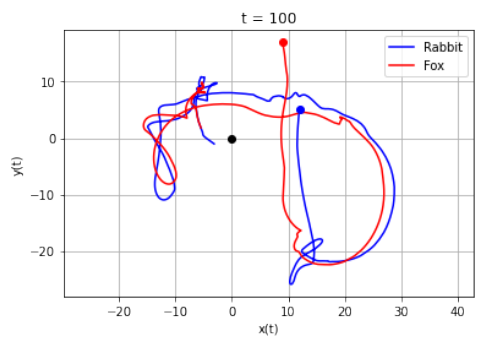

Figure 4: Here we have $m = 50$, which should be strong enough to force the rabbit to focus on getting to the burrow. However, because $t_f = 100$, it takes a winding path to postpone reaching the origin until that time.

The solutions in this scenario suffer from instability. As seen in Figure 5, the numerical solutions are highly sensitive to changes in the end time.

These issues could be addressed by adjusting our original equations or allowing the rabbit to stop moving once it reaches it's burrow, but for the purposes of this project we have decided to focus on other methods such as those discussed in the next section.

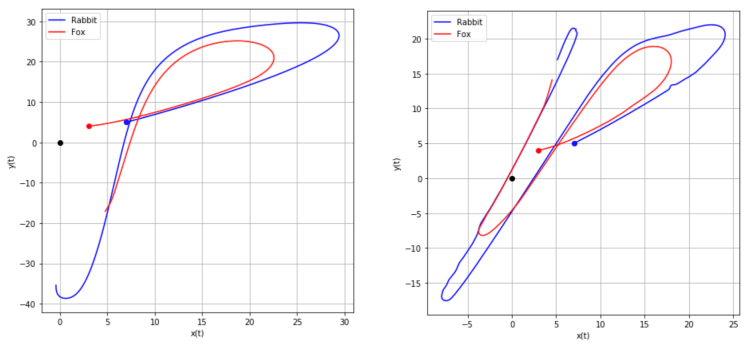

Figure 5: Fixed-time solutions are highly sensitive to perturbations in the end time. On the left, $t_f = 80$, and on the right, $t_f = 80 + 3 \times 10^{-5}$

4.3 Pontryagin's Maximum Principle: Free End Time

We would expect that the free end time condition would produce better solutions than fixed end times, because the rabbit has more freedom in choosing what path to take to its hole. Unfortunately, however, in this situation, the numerical solver often failed to converge to a solution. Additionally the rabbit and the fox would take long looping paths or head off in the wrong direction if the parameters weren't tuned precisely as seen in Figure 6.

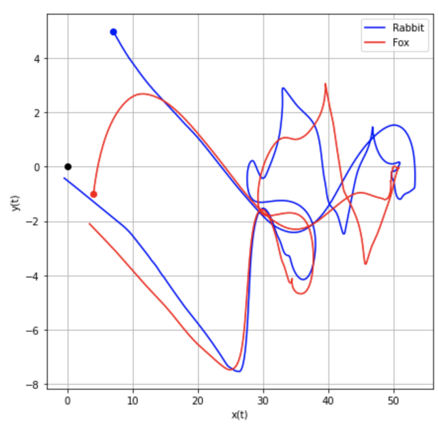

Figure 6: With $m = 0$ we see that the rabbit is attempting to avoid the fox at all costs even though it's postponing the goal of reaching the rabbit hole.

We found that for sufficiently large m the rabbit would reach the final destination, but the behavior of the fox was less stable under this construction. In some cases the fox would even disregard the rabbit and follow an alternative path. To fix this problem, we tried various values of β and m.

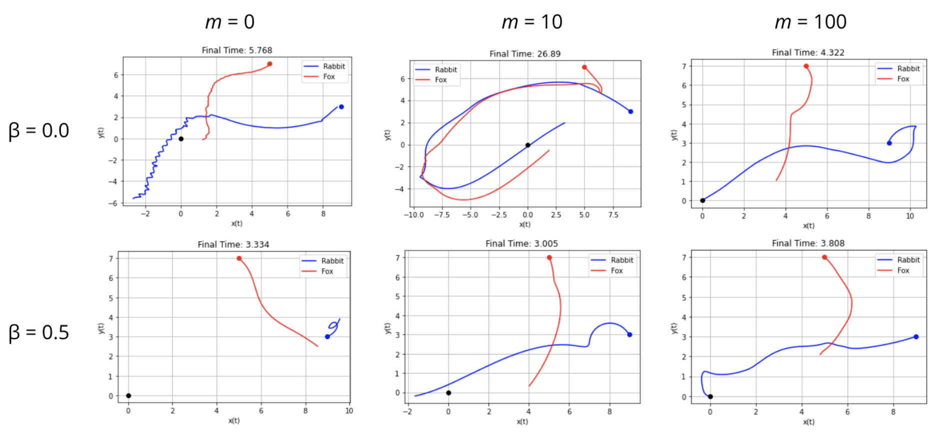

Figure 7: This shows how varying the β and m values changes our solution, even when all other parameters are held constant.

After fine tuning the parameters, as seen in Figure 7, we were able to determine values for β and m that resulted in a satisfactory solution as seen in Figure 8.

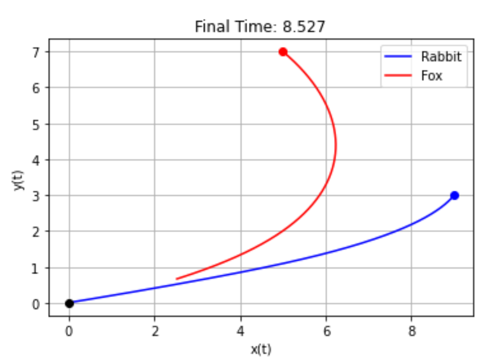

Figure 8: Here we have $\beta = 0.5$ and $m = 1000$. This is the solution that we would expect to see given the starting position of the fox and the rabbit.

5 Conclusion

We were able to successfully model the movement of the rabbit and the fox. Our LQR solution performed well but had issues near the final time. This is because we left our model with a free final end time and had to modify our state equation to be suitable for LQR. In order to incorporate more constraints on the end time and use our correct state equation we turned to Pontryagin's maximum principle with a fixed end time. This model was reasonable but as discussed earlier the rabbit often had to take unnecessarily long paths in order to reach home at the specified end time. This model also experienced instability. Next, we tried Pontryagin's maximum principle with a free end time. Leaving the end time free we often got long looping paths from the fox and the rabbit. Incentivizing the rabbit to reach the origin improved performance of the model. This solution gave us our best models, but required fine tuning of the β and m values for each starting placement.

Given our approach, we would not recommend that this be implemented in a practical application. It is possible that with further investigation and improved methods for determining the most favorable parameters, this could produce feasible results on a consistent basis.

References

[1] https://www.hsu.edu/uploads/pages/2006-7afpursuit.pdf