FlappyBird OptimalControl

Jeff Hansen, Eliza Powell, Gwen Johnson, Benj McMullin

Overview

We utilize optimal control techniques to create a solution to the popular mobile game Flappy Bird. Our obstacle avoidance model enables the bird to safely navigate through a series of pipes by adding spikes of acceleration to contend with the constant pull of gravity. We formulate this problem with a cost functional comprising of a reward and cost component. We reward the control when it successfully guides the bird in between the pipes, and we penalize the control when its shape deviates from a tapping motion.

Cost Functional and Boundary Conditions

$$J[u] = \int_0^{t_f} W\left[u(t) - \beta \max\{0, \sin(\alpha t)\}\right]^2 dt - \exp\left(\frac{-\left(y(t_f) - \frac{(P_l + P_u)}{2}\right)^2}{\sigma^2}\right)$$

$$\mathbf{x}'(t) = \begin{bmatrix} y(t) \\ y'(t) \end{bmatrix}' = \begin{bmatrix} y'(t) \\ u(t) - 9.8 \end{bmatrix}, \quad \mathbf{x}(0) = \begin{bmatrix} y_0 \\ 0 \end{bmatrix}, \quad \mathbf{x}(t_f) = \begin{bmatrix} \text{free} \\ 0 \end{bmatrix}$$

Methods and Approach

We model the tapping of the user through our control $u(t)$, which impacts the vertical acceleration of the bird. The function $\beta \max\{0, \sin(\alpha t)\}$ is a continuous function which only recognizes the positive parts of a sine wave. By manipulating the amplitude (β) and wavelength (α) of the sine component, the function results in a continuous series of sharp spikes that simulate discrete taps. Our functional guides the $u(t)$ to look like discrete taps by penalizing the squared difference between $u(t)$ and this sine function by the weight W.

In order for the bird to thread the gap between the two pipe obstacles, we add an endpoint incentive cost similar to a horizontal Gaussian shape (see Figure 2). This pushes the bird's path between the two pipes without forcing it to end exactly at a certain point; thus, the model is able to adjust and account for circumstances in which slightly different endpoints are needed without violating the laws of the model.

In Flappy Bird, the screen is moving forward at a constant rate, so we only include the y position and velocity in our state equation. The acceleration is represented by $u(t) - 9.8$, which accounts for a change in acceleration, velocity, and position when a tap occurs.

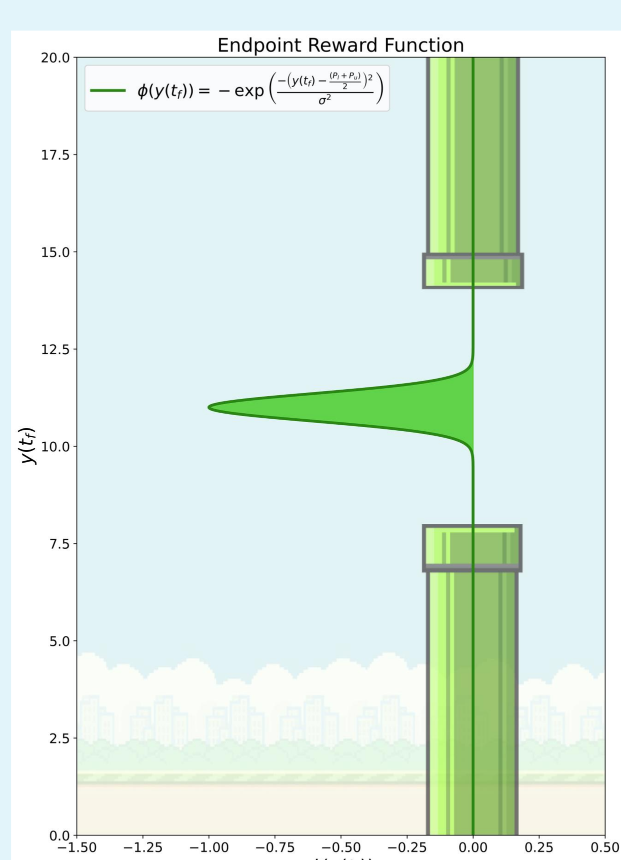


Figure 2 (above): This depicts the endpoint reward function that guides the bird through the pipes. We use a Gaussian with mean μ set to the midpoint between the upper and lower pipes and variance σ^2 set to a small value (0.1^2).

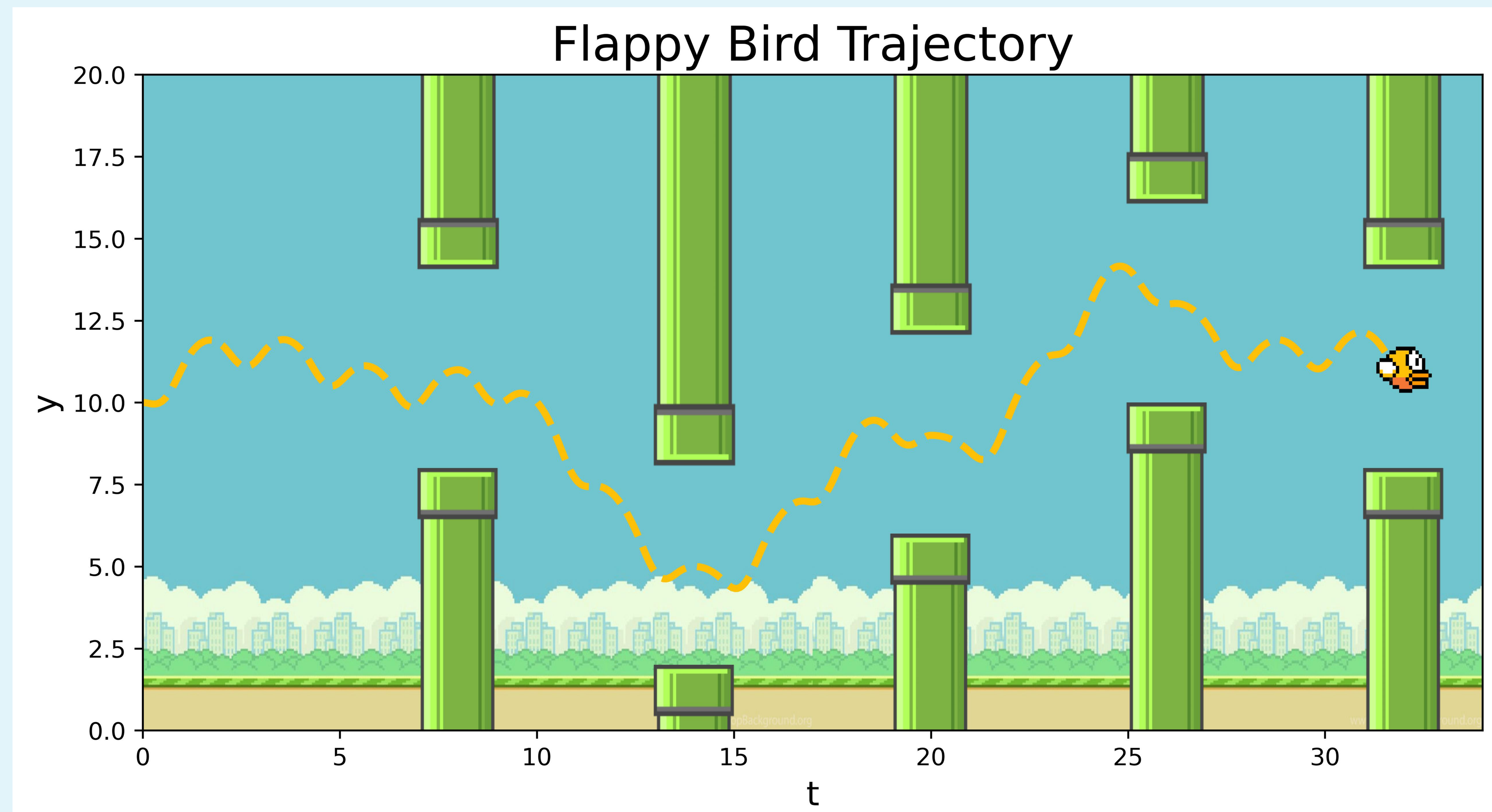
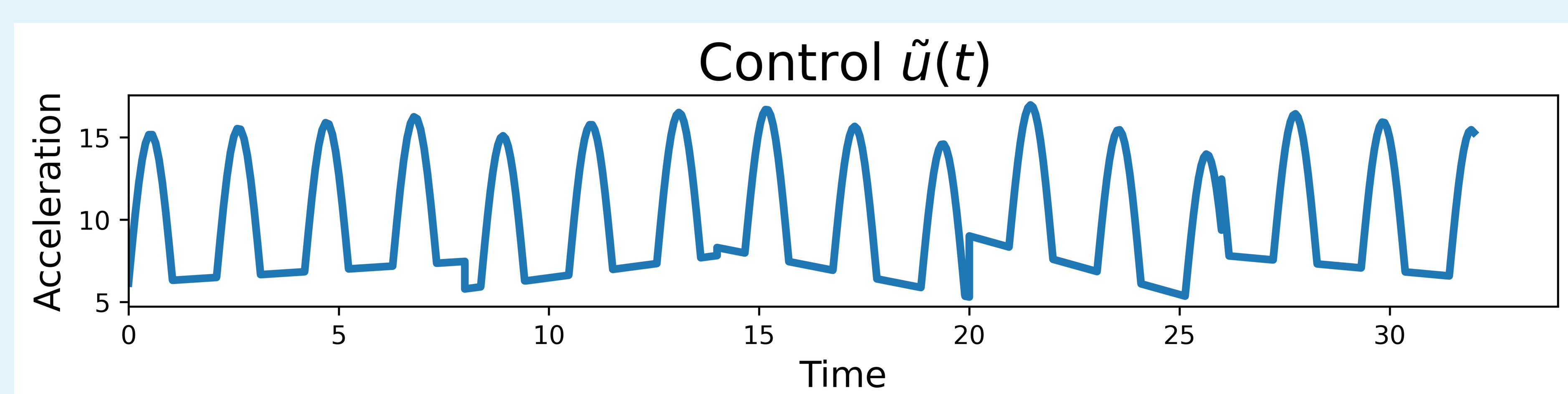
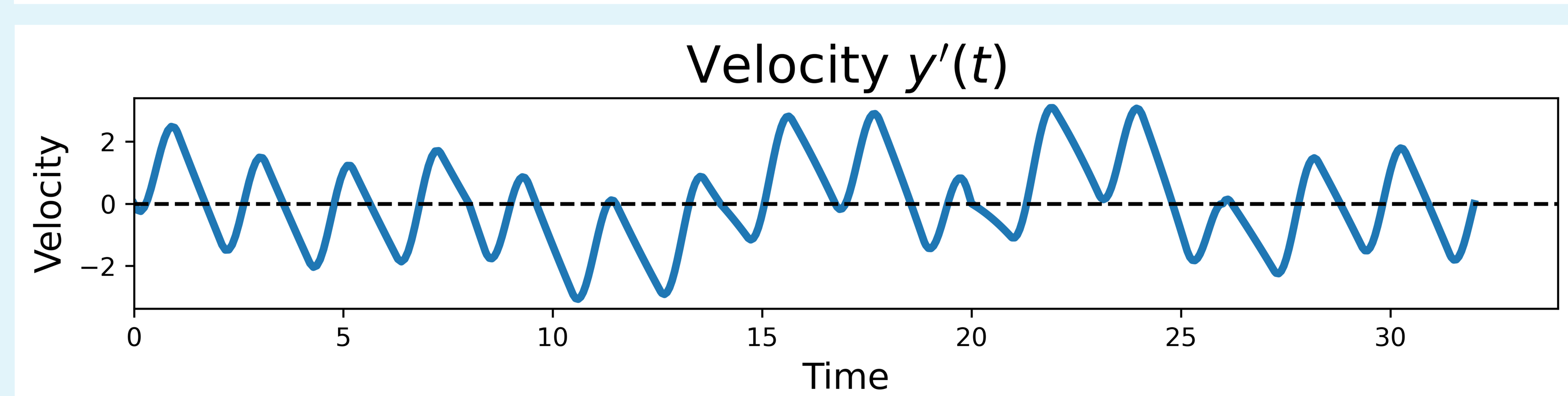


Figure 1 (above): This showcases the bird's trajectory through a series of pipes. To generate this, we break the pipe system into sub-problems (start-pipe1, pipe1-pipe2, etc.). We then use solve_bvp on each subsection where the current section's initial conditions are set to the final state of the previous solution. Finally, we concatenate all of the data from these solutions into one large plot, and add the pipes in the appropriate locations. Since each sub-problem solution represents the optimal path from one pipe to the next, by Bellman's Optimality Theorem, combining these solutions results in the optimal full path.





Figures 3 and 4 (above): This showcases how our control emulates distinct periodic taps. The control adds spikes of acceleration that fight against the constant pull of gravity. Because of this, the vertical velocity alternates between going up after a tap, and down after the tap's affect wears off.

Derivation

We begin by defining our Hamiltonian as follows:

$$H = \mathbf{p} \cdot \mathbf{f} - L = p_0 y'(t) + p_1(u(t) - 9.8) - W\left[u(t) - \beta \max\{0, \sin(\alpha t)\}\right]^2$$

Using the Pontryagin Maximum Principle, we solve for our costate equations:

$$p_0' = -\frac{\partial H}{\partial y} = 0$$
$$p_1' = -\frac{\partial H}{\partial y'} = -p_0$$

Since our $y(t_f)$ is free, our costate endpoint conditions are:

$$p_0(t_f) = -\frac{\partial \phi}{\partial y(t_f)} = \left(\frac{-(2y(t_f) - (P_l + P_u))}{\sigma^2}\right) \exp\left(\frac{-\left(y(t_f) - \frac{(P_l+P_u)}{2}\right)^2}{\sigma^2}\right)$$

$$p_1(t_f) = \text{free}$$

Finally, we compute the optimal control input $\tilde{u}$ by maximizing the Hamiltonian with respect to u:

$$\frac{\partial H}{\partial u} = 0 = p_1 + 2W\left[u(t) - \beta \max\{0, \sin(\alpha t)\}\right]$$
$$\implies \tilde{u} = \beta \max\{0, \sin(\alpha t)\} - \frac{p_1}{2W}$$

The optimal control input $\tilde{u}$ is determined by balancing the cost associated with deviations from the desired trajectory $\beta \max\{0, \sin(\alpha t)\}$ against the benefits of achieving desired state transitions.

Conclusion

This project seeks to find the most efficient path for a Flappy Bird to navigate through a series of pipe obstacles. In developing the early stages of our model, we encountered several hurdles in solving our functional, including singular Jacobian matrices. Upon restructuring our state space to solely include the vertical velocity and acceleration, we identified the constant horizontal velocity as the root of these errors. Leveraging Pontryagin's maximum principle as well as addressing these initial "obstacles", we successfully create an algorithm to beat the Flappy bird game. We achieve this through simulating a discrete tapping motion using a sine wave, as well as implementing a Gaussian endpoint reward. While this initial approach centers on penalizing deviations of the control and rewarding successful clearance of the pipes, our future work includes varying this approach by incorporating the pipe obstacles into our cost functional and changing the frequency of taps.

References

R. Gan, "Flappy Bird: Optimization of Deep Q-Network by Genetic Algorithm," 2022 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA), Dalian, China, 2022, pp. 703-707, doi: 10.1109/ICAICA54878.2022.9844595.