Model Predictive Control for Breathing Sine-Wave Tracking in the Cartpole System

Varun Rayamajhi

Abstract

This report presents a Model Predictive Control (MPC) formulation for breathing sine-wave tracking in the cartpole system. The objective is to make the cart follow a sinusoidal position reference with a slowly varying amplitude envelope while keeping the pole upright and satisfying the bounded input constraint $u \in [-1, 1]$. This behavior extends standard fixed-amplitude sine-wave tracking by requiring the controller to track both the fast oscillation and the slower amplitude modulation. The tracking task is encoded through an MPC cost that penalizes cart position error, cart velocity error, pole-angle deviation, pole angular velocity, and control effort. A ramp-in reference is introduced to reduce the initial velocity mismatch and prevent large startup control spikes. Experimental results show that the controller achieves stable and accurate tracking for moderate breathing frequencies, while higher-frequency references lead to larger phase lag, increased tracking error, and control inputs closer to saturation. Overall, the results show that MPC can generate the desired breathing motion, but tracking performance is limited by the underactuated cartpole dynamics and bounded control authority.

1 Introduction

We selected Model Predictive Control (MPC) as the control method and designed a breathing sine-wave tracking behavior for the cartpole system. In this behavior, the cart tracks a sinusoidal position reference whose amplitude changes slowly over time.

The resulting control problem is more difficult than constant-amplitude sine-wave tracking. The controller must follow the fast oscillations of the sine wave while also adapting to the slower variation in the reference amplitude. At the same time, the MPC controller must keep the pole upright and respect the bounded control input constraint $u \in [-1, 1]$.

2 Cartpole System and Problem Setup

The cartpole system consists of a cart moving horizontally on a frictionless track with a pole attached through an unactuated revolute joint. The system is underactuated because a single horizontal control input must control both the cart position and the pole angle. The state is defined as

$$\mathbf{x} = \begin{bmatrix} x & \dot{x} & \theta & \dot{\theta} \end{bmatrix}^{\mathsf{T}}, \tag{1}$$

where x is the cart position, $\dot{x}$ is the cart velocity, θ is the pole angle measured from the upright vertical position, and $\dot{\theta}$ is the pole angular velocity. The control input is the horizontal force applied to the cart. In our MPC implementation, the input is continuous and bounded as

$$u \in [-1, 1]. \tag{2}$$

The objective is to make the cart track a desired reference trajectory while keeping the pole upright. Therefore, the desired pole angle and angular velocity are

$$\theta_{\mathrm{ref}}(t) = 0, \qquad \dot{\theta}_{\mathrm{ref}}(t) = 0. \tag{3}$$

At the same time, the controller must respect the input constraint and avoid episode termination due to excessive cart displacement or pole angle.

Let m_c denote the cart mass, m_p denote the pole mass, and L denote the pole length. The total mass of the system is

$$M = m_c + m_p. \tag{4}$$

Using the parameters from the cartpole environment, we use

$$m_c = 1.0 \text{ kg}, \qquad m_p = 0.1 \text{ kg}, \qquad L = 0.5 \text{ m}. \tag{5}$$

The nonlinear continuous-time dynamics are written using the intermediate variable

$$D = \frac{F + m_p L \dot{\theta}^2 \sin\theta}{M}, \tag{6}$$

where F is the horizontal force applied to the cart. The pole angular acceleration and cart acceleration are then given by

$$\ddot{\theta} = \frac{g \sin\theta - \cos\theta\, D}{L\left(\frac{4}{3} - \frac{m_p \cos^2\theta}{M}\right)}, \tag{7}$$

and

$$\ddot{x} = D - \frac{m_p L \ddot{\theta} \cos\theta}{M}. \tag{8}$$

The system is simulated in discrete time using Euler integration with time step $\tau = 0.02$. The discrete-time state update is

$$\begin{aligned}
x_{t+1} &= x_t + \tau \dot{x}_t, && (9)\\
\dot{x}_{t+1} &= \mathrm{clip}\left(\dot{x}_t + \tau \ddot{x}_t, -\dot{x}_{\max}, \dot{x}_{\max}\right), && (10)\\
\theta_{t+1} &= \theta_t + \tau \dot{\theta}_t, && (11)\\
\dot{\theta}_{t+1} &= \mathrm{clip}\left(\dot{\theta}_t + \tau \ddot{\theta}_t, -\dot{\theta}_{\max}, \dot{\theta}_{\max}\right). && (12)
\end{aligned}$$

For the breathing sine-wave tracking task, the control problem is not only to stabilize the upright equilibrium, but to track a time-varying cart reference while maintaining pole stability. The reference state used by the controller is

$$\mathbf{x}_{\mathrm{ref}}(t) = \begin{bmatrix} x_{\mathrm{ref}}(t) & \dot{x}_{\mathrm{ref}}(t) & 0 & 0 \end{bmatrix}^{\mathsf{T}}. \tag{13}$$

Thus, the MPC controller must balance cart trajectory tracking against pole stabilization. This tradeoff is important because aggressive cart motion requires larger control forces, but the same force also affects the pole dynamics. As a result, perfect tracking may not be achievable when the desired reference requires accelerations that conflict with the upright-pole objective or exceed the bounded input range.

In our experiments, we use the termination thresholds

$$|x| \le x_{\max}, \qquad |\theta| \le \theta_{\max}, \tag{14}$$

with $x_{\max} = 4.0\,\mathrm{m}$ and $\theta_{\max} = 25^\circ$ for the custom tracking task.

3 Reference Trajectory Design and MPC Formulation

We define the reference trajectory as

$$x_{\text{ref}}(t) = r(t)\,A(t)\sin(\omega t), \tag{15}$$

where

$$A(t) = A_0\big(1 + \beta\sin(\omega_a t)\big), \qquad \omega_a < \omega. \tag{16}$$

A_0 is the base amplitude, $\beta \in (0,1)$ is the modulation depth, $\omega = 2\pi f$ is the main sine frequency, and $\omega_a = 2\pi f_{\text{amp}}$ is the (slow) envelope frequency. We also use a ramp-in function $r(t)$ during the first T_{ramp} seconds to avoid a large startup transient [1]. We use the following piecewise ramp-in function:

$$r(t) = \begin{cases} \frac{1}{2}\left(1 - \cos\left(\frac{\pi t}{T_{\text{ramp}}}\right)\right), & 0 \le t \le T_{\text{ramp}}, \\ 1, & t > T_{\text{ramp}}. \end{cases} \tag{17}$$

Tracking only $x_{\text{ref}}(t)$ can produce trajectories with noticeable phase error in the veocity profile. To reduce this phase/derivative error and improve tracking, we therefore track both position and velocity by using the reference state $\mathbf{x}_{\text{ref}}(t) = \big[x_{\text{ref}}(t),\, \dot{x}_{\text{ref}}(t),\, 0,\, 0\big]^{\mathsf{T}}$. We obtain $\dot{x}_{\text{ref}}(t)$ by differentiating Eq. (15):

$$\dot{x}_{\text{ref}}(t) = r(t)\left(\dot{A}(t)\sin(\omega t) + A(t)\omega\cos(\omega t)\right) + \dot{r}(t)\,A(t)\sin(\omega t). \tag{18}$$

We note that $\theta_{\text{ref}}(t)$ and $\dot{\theta}_{\text{ref}}(t)$ are set to 0 because we want to keeep the pole upright. Finally, we derived our tracking cost for MPC as follows:

$$L(\mathbf{x}_t, \mathbf{u}_t) = (\mathbf{x}_t - \mathbf{x}_{\text{ref},t})^\top Q(\mathbf{x}_t - \mathbf{x}_{\text{ref},t}) + \mathbf{u}_t^\top R\mathbf{u}_t, \qquad \Phi(\mathbf{x}_T) = (\mathbf{x}_T - \mathbf{x}_{\text{ref},T})^\top Q_T(\mathbf{x}_T - \mathbf{x}_{\text{ref},T}), \tag{19}$$

where $Q = \text{diag}(q_x, q_{\dot{x}}, q_\theta, q_{\dot{\theta}})$,$R > 0$, and $Q_T = mQ$ for some constant m (in our case, m=1 produced good results).

3.1 Controller Tuning and Experimental Procedure

3.1.1 Constant Sine Wave Tracking

We first started with the standard trajectory tracking behavior, i.e. track a constant sine wave

$$x_{\text{ref}}(t) = A_0\,\sin(\omega t), \qquad \dot{x}_{\text{ref}}(t) = A_0\,\omega\cos(\omega t). \tag{20}$$

while keeping the pole upright. In our initial runs, we saw a large control spike at the start. The reason is that even when $x_{\text{ref}}(0) = 0$, a pure sine wave has a non-zero desired initial velocity $\dot{x}_{\text{ref}}(0) = \omega A_0$. The environment typically resets near $(x, \dot{x}) \approx (0,0)$. Therefore, according to our reference trajectory, the controller tries to instantly create a non-zero cart velocity at $t = 0$. This can can saturate the control input and cause the pole to fall. To fix this, we introduced the ramp-in function (Eq. (17)) and used the following reference trajectory

$$x_{\text{ref}}(t) = A_0\,r(t)\sin(\omega t), \qquad \dot{x}_{\text{ref}}(t) = A_0(\dot{r}(t)\sin(\omega t) + r(t)\,\omega\cos(\omega t))\,. \tag{21}$$

so that $\dot{x}_{\text{ref}}(0) = 0$. This gave more stable tracking on the constant sine wave tracking behavior (Figure 1).

3.1.2 Breathing Sine Wave Tracking

After we achieved stable sine wave tracking (with constant amplitude), we increased the difficulty by modulating the amplitude with a slow envelope (breathing). We used the same ramp-in function and defined

$$x_{\text{ref}}(t) = r(t)\, A(t) \sin(\omega t), \qquad A(t) = A_0\big(1 + \beta \sin(\omega_a t)\big). \tag{22}$$

The corresponding reference cart velocity is

$$\dot{x}_{\text{ref}}(t) = r(t) \left(\dot{A}(t) \sin(\omega t) + A(t)\omega \cos(\omega t)\right) + \dot{r}(t)\, A(t) \sin(\omega t), \tag{23}$$

where $\dot{A}(t) = \beta\, \omega_a A_0 \cos(\omega_a t)$.

3.1.3 Hyperparameter Tuning and Key Decisions

We performed over 200 experiments/iterations to achieve our chosen behavior, and we only report the major decisions/strategies we used in hyperparameter tuning. We tuned parameters using a simple loop: run an episode rollout, inspect the plots and metrics, and adjust a small set of parameters. Our main strategies were:

- **Avoid saturation first.** When tracking was too aggressive (too high q_x and $q_{\dot{x}}$ or R too small), the optimizer pushed u toward ± 1 for long periods. This often led to larger pole oscillations and caused termination via pole-angle limit. We fixed such cases by increasing R, which helped reduced saturation and produced smoother control. Increasing q_θ and $q_{\dot{\theta}}$ also helped because it prevented the controller from sacrificing pole stabilization for tracking.

- **Reduce phase lag.** When tracking was stable but lagged behind the reference, increasing $q_{\dot{x}}$ helped us in matching the reference velocity profile. Increasing q_x helped us in matching the reference position profile. We also observed that increasing H and `max_iters` helped the controller better handle future reference changes, but it increased runtime. Therefore, owing to our computational capacity, we restricted increasing the horizon beyond 120 and `max_iters` beyond 5.

- **Increase difficulty gradually.** We tuned on the constant sine wave tracking behavior first. Even with this behavior, we increased the difficulty gradually: we first kept the frequency constant and tuned on the amplitudes from $A \in \{0.2, 0.4, 0.6, 0.8, 1.0\}$. Afterwards, we increased the frequency from 0.05Hz to 0.25Hz with step sizes of 0.05Hz. This process of experimentation showed us that feasibility was limited by the bounded control inputs. We observed that higher (A, f) combinations push required acceleration beyond what $|u| \leq 1$ can supply. This led to unavoidable saturation and loss of stability. When we mvoed the difficulty to breathing sine wave, we followed the same philosophy: we kept the main (A_0, f) feasible and introduced a slow envelope so that difficulty increases without instantly crossing the actuation limits. We finally tuned on the tracking behavior with a higher envelope frequency.

- **Line search.** In our experiments, we realized that line search was crucial for stability as we increased the tracking difficulty. It prevents overly aggressive updates that can cause saturation. Therefore, for this task, iLQR updates controls in the forward pass with line search as $u_t^{\text{new}} = u_t + \alpha k_t + K_t(x_t^{\text{new}} - x_t)$, with $\alpha \in \{1.0, 0.5, 0.25, 0.1, 0.05, 0.02\}$.

4 Results and Discussion

4.1 Constant sine wave

In Figure 1, we track a constant sine wave reference with amplitude $A_0 = 0.4\,\text{m}$ and frequency $f = 0.2\,\text{Hz}$. We use MPC with hyperparameters $Q = \text{diag}(120, 75, 2500, 250)$, $R = 0.05$, $H = 100$, and `max_iters=5`. The cart follows the reference position and velocity with RMSE $0.052\,\text{m}$ and $0.079\,\text{m/s}$ respectively.

From panels (a) and (b), we observe that the tracking is good overall, but the cart shows a small delay (phase lag) and its peaks are slightly smaller than the reference, especially near the maximum and minimum points. Panel (c) shows that the pole stays close to upright (with peak angle about 2.77°), and panel (d) shows that the control input stays within the limit $u \in [-1, 1]$. Finally, panel (e) shows a nearly closed loop in the $(x, \dot{x})$ phase plot, which is consistent with steady periodic wave tracking[1].

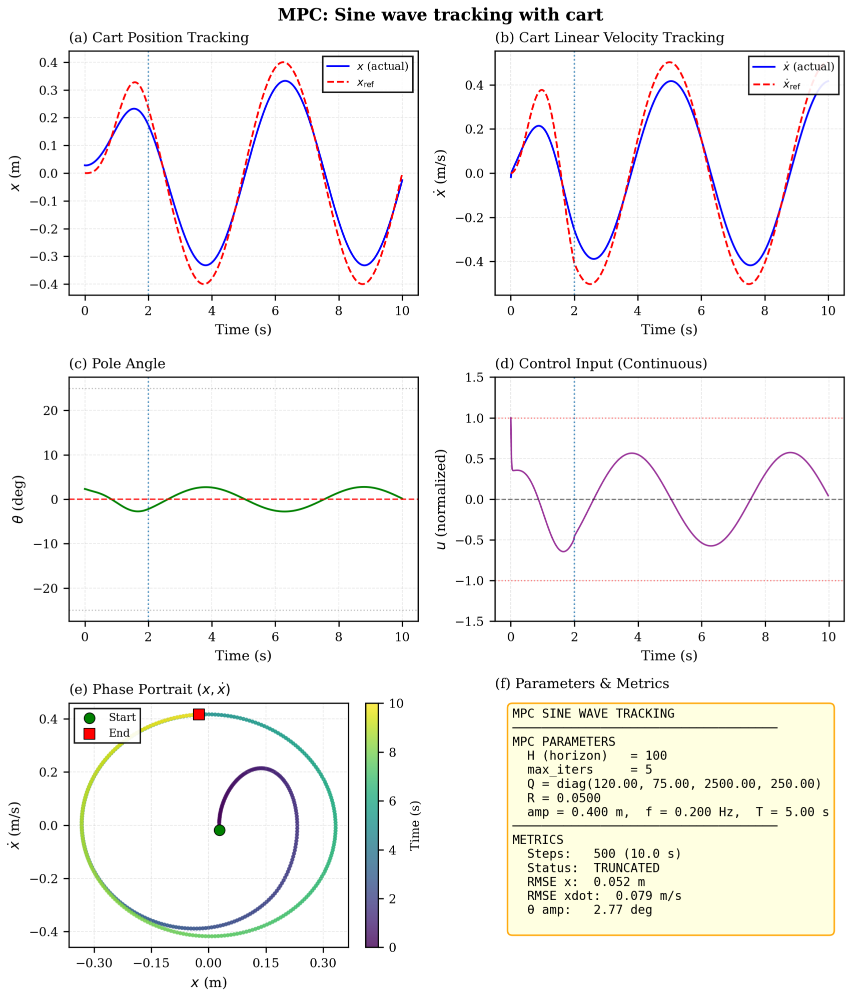

Figure 1: Trajectory Tracking with MPC: Constant Sine Wave

4.1.1 Breathing sine wave (low envelope frequency)

In Figure 2, we track a breathing sine wave reference with a slowly varying amplitude envelope (low envelope frequency $f_{\text{amp}} = 0.05\,\text{Hz}$) and main sine frequency $f = 0.15\,\text{Hz}$. We've visualized

[1] Since our main goal was breathing sine wave tracking, we didn't attempt perfect tracking of a constant sine wave and proceeded after obtaining satisfactory results.

the envelope by plotting the bounds $\pm r(t)A(t)$ around the reference.

Overall, we see that the tracking is strong: the cart position and velocity closely follow the reference in panels (a) and (b). There's only a small delay near the peaks, but the errors are low (RMSE $x = 0.024\,\text{m}$ and RMSE $\dot{x} = 0.030\,\text{m/s}$). The low envelope frequency makes the tracking easier because the amplitude changes slowly, so over short time windows ($T = 500$ steps or 10s) the reference looks close to a constant-amplitude sine wave. Panel (c) shows that the pole remains close to upright (peak angle about 2.44°). Similarly, panel (d) shows that the control input stays within the limits $u \in [-1, 1]$, with a smoother control. Finally, the $(x, \dot{x})$ phase plot in panel (e) is not perfectly closed. This is expected because the motion is not strictly periodic when the amplitude is changing over time.

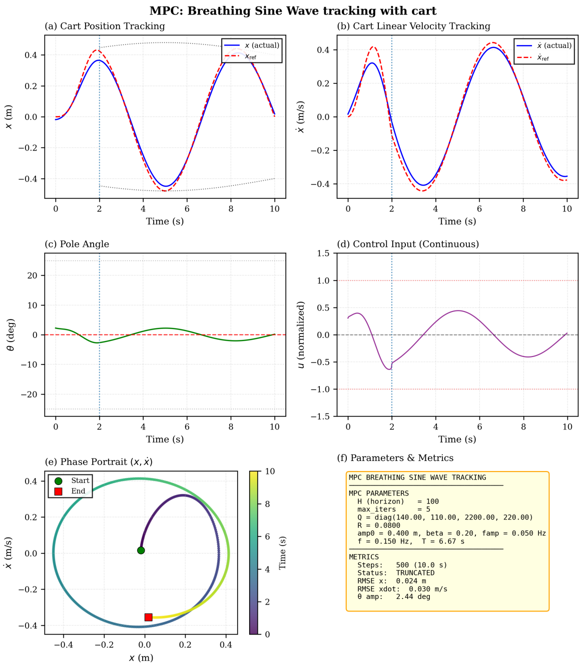

Figure 2: Trajectory Tracking with MPC: Breathing Sine Wave (low envelope frequency)

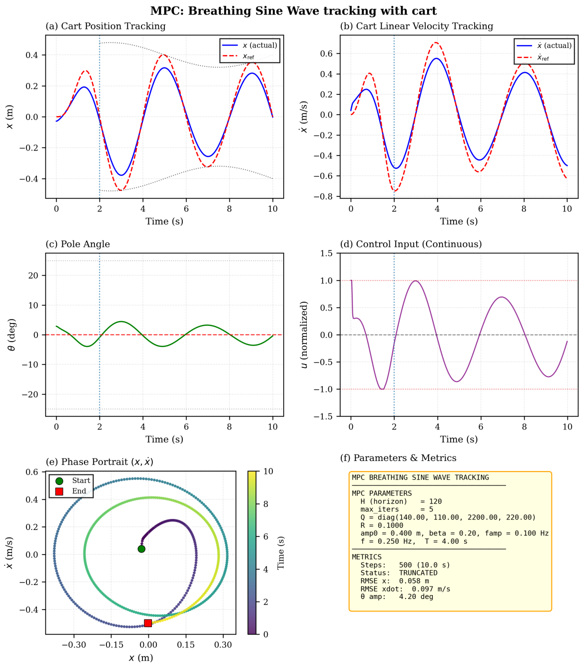

Figure 3: Trajectory Tracking with MPC: Breathing Sine Wave (high envelope frequency)

4.1.2 Breathing sine wave (high envelope frequency)

In Figure 3, we increase the difficulty by using a faster breathing reference ($f = 0.25\,\text{Hz}$ and a higher envelope frequency $f_{\text{amp}} = 0.10\,\text{Hz}$). We also use a longer horizon $H = 120$ (other parameters are shown in panel (f)).

Compared to the low-frequency case in Figure 2, the tracking is noticeably worse: panels (a) and (b) show larger mismatches near the peaks. This lag is reflected in the higher errors (RMSE $x = 0.058\,\text{m}$ and RMSE $\dot{x} = 0.097\,\text{m/s}$). We also observe that the controller works harder: panel (d) shows the input coming close to the limits $u = \pm 1$. Panel (c) indicates the pole is still kept near upright, but with a larger oscillation (peak angle about 4.20°). Finally, the $(x, \dot{x})$ phase portrait in panel (e) does not form a single closed loop. This is again expected because the amplitude is changing more quickly. We can clearly see that at around 6-8 seconds, the loop in panel (e) is smaller and the corresponding position and velocity in panels (a) and (b) are also smaller. The same pattern is seen for bigger loop around 2-4 seconds. This behavior clearly demonstrates that the cart is performing the breathing behavior.

5 Conclusion

Overall, our results show a clear tradeoff: increasing the reference difficulty (higher f and f_{amp}) demands larger and faster control actions, so the controller operates closer to the input limits $u \in [-1, 1]$ while still prioritizing pole stabilization. As a result, tracking errors increase mainly near the peaks and during rapid envelope changes. This is expected because perfect tracking would require control effort that violates the actuator bounds or would destabilize the pole. Finally, we argue why our breathing sine-wave tracking isn't perfect in Appendix 7.1. Likewse, we also provide the videos of 5 episode rollouts in Appendix 7.2

References

[1] K. M. Lynch and F. C. Park, *Modern robotics.* Cambridge University Press, 2017.

[2] D. Perkins, O. J. Escobar, and L. Green, "Dqn performance with epsilon greedy policies and prioritized experience replay," *arXiv preprint arXiv:2511.03670*, 2025.

[3] E. Even-Dar and Y. Mansour, "Learning rates for q-learning," *Journal of machine learning Research*, vol. 5, no. Dec, pp. 1–25, 2003.

[4] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. Scokaert, "Constrained model predictive control: Stability and optimality," *Automatica*, vol. 36, no. 6, pp. 789–814, 2000.

[5] J. Chen, W. Zhan, and M. Tomizuka, "Constrained iterative lqr for on-road autonomous driving motion planning," in *2017 IEEE 20th International conference on intelligent transportation systems (ITSC).* IEEE, 2017, pp. 1–7.

[6] V. Makoviychuk, L. Wawrzyniak, Y. Guo, M. Lu, K. Storey, M. Macklin, D. Hoeller, N. Rudin, A. Allshire, A. Handa *et al.*, "Isaac gym: High performance gpu-based physics simulation for robot learning," *arXiv preprint arXiv:2108.10470*, 2021.

Appendix

7.1 Why is our breathing sine-wave tracking imperfect?

We argue why our chosen tracking behavior is not perfect. We note that the perfect tracking of our chosen cart trajectory would require $x(t) = x_{\text{ref}}(t)$ (and typically $\dot{x}(t) = \dot{x}_{\text{ref}}(t)$ and $\ddot{x}(t) = \ddot{x}_{\text{ref}}(t)$ at all times). However, we've kept the cartpole underactuated intentionally for the challenge. Because of the under-actuation, a single input force F (with $u \in [-1, 1]$) must simultaneously regulate the cart motion and keep the pole upright. In the cartpole dynamics, the accelerations are coupled via the intermediate variable

$$D = \frac{F + m_p L \dot{\theta}^2 \sin\theta}{M}, \qquad \ddot{\theta} = \frac{g \sin\theta - \cos\theta\, D}{L\left(\frac{4}{3} - \frac{m_p \cos^2\theta}{M}\right)}, \qquad \ddot{x} = D - \frac{m_p L \ddot{\theta} \cos\theta}{M}. \tag{24}$$

Therefore, choosing F to match $\ddot{x}_{\text{ref}}$ ultimately also changes $\ddot{\theta}$ (and vice versa). For small angles ($\sin\theta \approx \theta$, $\cos\theta \approx 1$) and moderate $\dot{\theta}$, we note that $D \approx F/M$ and

$$\ddot{\theta} \approx \frac{g\theta - \frac{F}{M}}{L\left(\frac{4}{3} - \frac{m_p}{M}\right)}. \tag{25}$$

This shows that any nonzero F produces an angular acceleration unless θ moves away from zero to compensate. Therefore, when we demand aggressive cart accelerations, it conflicts with the requirement to keep $\theta \approx 0$. Finally, bounded actuation $u \in [-1, 1]$ (and the environment's clipping of velocities in the discrete-time update) limits the achievable acceleration envelope. Consequently, if a fast-changing reference requires more acceleration than the bounded input can provide (still while keeping the pole stable), the controller cannot track it perfectly and will show phase lag and smaller peaks.

7.2 Video of 5 episode rollouts

1. Breathing Sine Wave with high frequency (sped up by 3x): YouTube Video Link
2. Breathing Sine Wave with low frequency (sped up by 2.5x): YouTube Video Link
3. Constant Sine Wave (sped up by 2.5x): YouTube Video Link