No Free Lunch in Ball Catching: A Comparison of Cartesian and Angular Representations for Control
Supplementary Material (S1 Text)
Proofs: Angular Representations Violate Markov Property

Contents

1 Introduction 1
2 Non-Markov κθ 2
3 Non-Markov Derivative of κθ 3
4 Non-Markov Second Derivative of κθ 3

1 Introduction

In the following, we will prove that the representations used by angular controllers for ball catching are not Markov. To that end, we will prove the following theorem:

Theorem 1.1 (Angular representations are not Markov). Given \( x \in \{ \theta, \dot{\theta}, \ddot{\theta} \} \) with dynamics \( x_A(t) \) governed by initial conditions \( A = \{ \nu_A, \phi_A \} \) and agent trajectory \( a(t) \), there exist initial conditions \( B = \{ \nu_B, \phi_B \} \), \( A \neq B \), time steps \( t_A, t_B \), and an agent trajectory \( a(t) \) such that two following conditions are fulfilled:

(i) \( x_A(t_A) = x_B(t_B) \),

(ii) \( \dot{x}_A(t_A) \neq \dot{x}_B(t_B) \).

Our ansatz to proof Theorem 1.1 is by finding counterexamples showing that the angular representations \( \{ \theta, \dot{\theta}, \ddot{\theta} \} \) violate the Markov property.

We know a quantity \( x(t) \) fulfills the Markov property if for \( t > t_n > \ldots > t_1 \):

\[
P(x(t) \mid x(t_n)) = P(x(t) \mid x(t_n), \ldots, x(t_1)).
\] (1)

Intuitively, this means that it is sufficient to rely solely on \( x(t_n) \) for predicting any future state \( x(t) \); adding more information about the past of \( x(t) \) does not make the prediction better. If \( x \) is deterministic, Eq. (1) is equivalent to the statement that there exists exactly one \( x(t) \) for which
We begin with the simplest case, \( \theta \), and prove the following lemma:

**Lemma 2.1** (\( \theta \) is not Markov). For \( \theta_A(t) \) governed by initial conditions \( A = \{ \nu_A, \phi_A \} \) and agent trajectory \( a(t) \), there exist initial conditions \( B = \{ \nu_B, \phi_B \} \), \( A \neq B \), time steps \( t_A, t_B \), and an agent trajectory \( a(t) \) such that (i) \( \theta_A(t_A) = \theta_B(t_B) \) but (ii) \( \dot{\theta}_A(t_A) \neq \dot{\theta}_B(t_B) \).

**Proof.** We know that \( \theta \) is linear if the agent is waiting at the landing point (\( D = 0 \)), and that the trajectory of \( \theta \) has slope \( \dot{\theta} \). Therefore, we consider this special case, and search for two different sets of initial parameters \( A = \{ \nu_A, \phi_A \} \), \( B = \{ \nu_B, \phi_B \} \) and time steps \( t_A \) and \( t_B \) such that (i) \( \theta_A(t_A) = \theta_B(t_B) \) and (ii) \( \dot{\theta}_A(t_A) \neq \dot{\theta}_B(t_B) \).

We start by computing the values of \( \theta \) and its derivatives for the special case \( D = 0 \):

\[
\theta = \frac{gt}{2\nu \cos(\phi)}, \quad (2)
\]

\[
\dot{\theta} = \frac{g}{2\nu \cos(\phi)}, \quad (3)
\]

The equations show that the slope \( \dot{\theta} \) is constant and solely depends on the initial parameters (and gravity). Therefore, the only way to make them different, and thus fulfill condition (ii), is to choose different initial parameters. Without loss of generality, we assume \( \nu_A \neq \nu_B \) and \( \phi_A = \phi_B \). We now need to find time steps \( t_A \) and \( t_B \) such that equation (i) is fulfilled, \( \theta_A(t_A) = \theta_B(t_B) \). By solving this equation for \( t_A \), we obtain:

\[
t_A = \frac{\nu_A t_B}{\nu_B}, \quad (4)
\]
Therefore, (i) \( \theta_A(t_A) = \theta_B(\nu_A \nu_B^{-1} t_B) = \theta_B(t_B) \), but by construction (ii) \( \dot{\theta}_A(t_A) \neq \dot{\theta}_B(t_B) \). This shows that \( \theta \) is not Markov.

\[ \square \]

3 Non-Markov Derivative of \( \theta \)

Next, we show that \( \dot{\theta} \) is not Markov either:

Lemma 3.1 (\( \dot{\theta} \) is not Markov). For \( \dot{\theta}_A(t) \) governed by initial conditions \( A = \{ \nu_A, \phi_A \} \) and agent trajectory \( a(t) \), there exist initial conditions \( B = \{ \nu_B, \phi_B \} \), \( A \neq B \), time steps \( t_A, t_B \), and an agent trajectory \( a(t) \) such that (i) \( \dot{\theta}_A(t_A) = \dot{\theta}_B(t_B) \) but (ii) \( \ddot{\theta}_A(t_A) \neq \ddot{\theta}_B(t_B) \).

Proof. This case is a bit more involved as we cannot assume that the agent waits at landing point, that is we cannot assume \( a(t) = R, D = 0 \). The reason is that for this case \( \dot{\theta} \) is constant and thus \( \dot{\theta}_A \) is zero, which makes the representation Markov in a trivial way.

Our ansatz is to create an explicit dependency between \( \nu_A \) and \( \nu_B \) as well as \( t_A \) and \( t_B \), and then solve for the parameters such that they fulfill (i) and (ii). Without loss of generality, we will keep \( \phi_A = \phi_B \), and define \( \nu_B = \nu_A + \varepsilon \) by some \( \varepsilon \neq 0 \), and define \( t_B = t_A + \delta \) for some \( \delta \neq 0 \) and solve for \( \delta \).

Additionally, we need to ensure that \( D \neq 0 \) (and \( D \neq -R \); it must not coincide with the ball starting position because then \( \theta \) is linear, too). Since we are only interested in a counterexample, we can set \( D \) to any suitable value and thus use \( D = -R_1 \). This means the agent is standing exactly in the middle between the ball’s starting and landing point.

By applying all of the aforementioned constraints, we can compute \( \theta(t) \) and its first and second derivatives:

\[
\theta(t) = -\nu \sin(\phi) \frac{\nu_A \sin(\phi) - \frac{1}{2} gt}{\nu \sin(\phi) - gt \cos(\phi)}, \quad (5)
\]

\[
\dot{\theta}(t) = \frac{g \left( \nu \sin(\phi) - \frac{1}{2} gt \right) + (-\nu \sin(\phi) + gt)^2 \nu (-\nu \sin(\phi) + gt)^2 \cos(\phi)}{\nu \sin(\phi) + gt^2 \cos(\phi)}, \quad (6)
\]

\[
\ddot{\theta}(t) = \frac{\nu g^2 \sin^2(\phi)}{(\nu \sin(\phi) - gt)^3 \cos(\phi)}. \quad (7)
\]

By solving \( \ddot{\theta}_A(t_A) = \ddot{\theta}_B(t_B) \) for \( \delta \), we obtain two conjugate solutions, from which we choose the positive one. As \( \delta \) results in a complex formula, we omit writing them explicitly here. The solution can be found in the online material referenced at the beginning of this section.

This shows that any two sets of initial conditions \( A \) and \( B \) will have identical values for \( \dot{\theta} \) but different values for \( \ddot{\theta} \) at \( t_A \) and \( t_B = t_A + \delta \), respectively. Moreover, it is easy to find sets of initial conditions for which \( \delta \neq 0 \), and choose a value for \( \varepsilon \neq 0 \). This shows that \( \dot{\theta} \) is not Markov.

\[ \square \]

4 Non-Markov Second Derivative of \( \theta \)

Finally, we show that \( \ddot{\theta} \) is not Markov, either:

Lemma 4.1 (\( \ddot{\theta} \) is not Markov). For \( \ddot{\theta}_A(t) \) governed by initial conditions \( A = \{ \nu_A, \phi_A \} \) and agent trajectory \( a(t) \), there exist initial conditions \( B = \{ \nu_B, \phi_B \} \), \( A \neq B \), time steps \( t_A, t_B \), and an agent trajectory \( a(t) \) such that (i) \( \ddot{\theta}_A(t_A) = \ddot{\theta}_B(t_B) \) but (ii) \( \dddot{\theta}_A(t_A) \neq \dddot{\theta}_B(t_B) \).

3
Proof. The proof is carried out equivalently as for \( \dot{\theta} \). We begin by computing the third derivative of \( \theta \):

\[
\ddot{\theta}(t) = \frac{3\nu g^3 \sin^2(\phi)}{(\nu \sin(\phi) - gt)^4 \cos(\phi)}.
\]

(8)

We can then find an expression for \( \delta \) by solving \( \ddot{\theta}_A(t_A) = \ddot{\theta}_B(t_B) \) for \( \delta \). Again, we omit writing out \( \delta \) explicitly and state that it is possible to find sets of initial conditions for which \( \delta \neq 0 \) and choose \( \varepsilon \neq 0 \). Thus \( \theta \) is not Markov.

Proof. Taken together, the three Lemmas 2.1[4.1] prove Theorem 1.1.

References

1. Meurer A, Smith CP, Paprocki M, Žertík O, Kirpichev SB, Rocklin M, et al. SymPy: symbolic computing in Python. PeerJ Computer Science. 2017;3:e103.