

Analytical Walking Models + RL towards Less Painful Biped Walking Policies

Noah Fisher and Daniel Levine



Training walking policies is hard and opaque for model free.

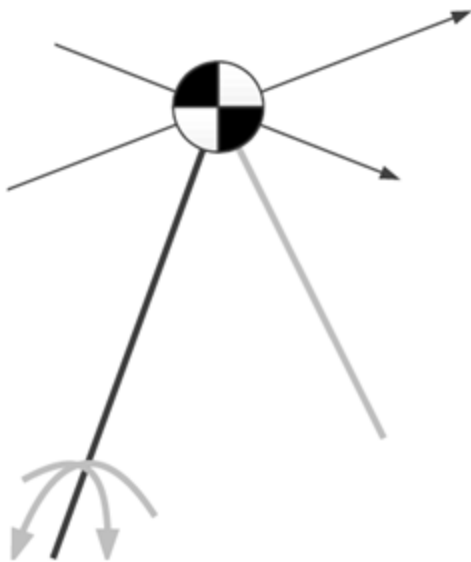
Creating walking controllers is tedious and often fragile using classical methods.

Can we create a classical foundation for baseline ability, and use RL to provide robustness?

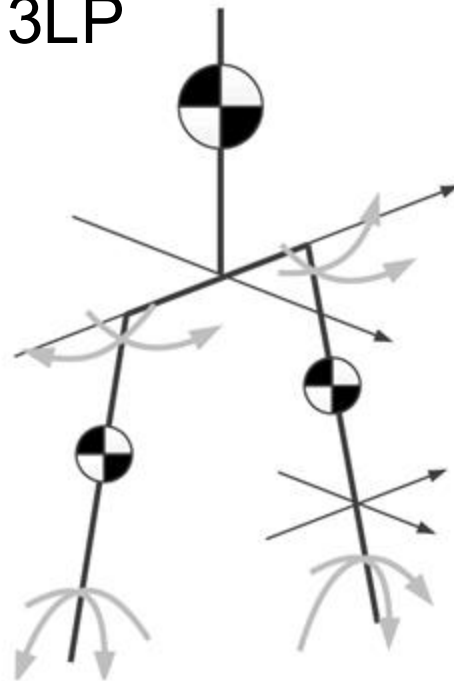
To Answer This Question:

- Linear Dynamics Models for Walking
- Quadratic Cost
- Closed Form Expressions → Derive LQR Baselines
- Introduce Unmodelled Noise
- Train (Ideally) Using Structure but with perturbations
- Compare Baseline Analytic Controllers to RL Policies (in progress)

Our Linear Template Models: LIP, 3LP



LIP, Kajita 1991



3LP, Faraji 2015

3LP is a three linearized pendulum extension for swing dynamics.

3D LIP

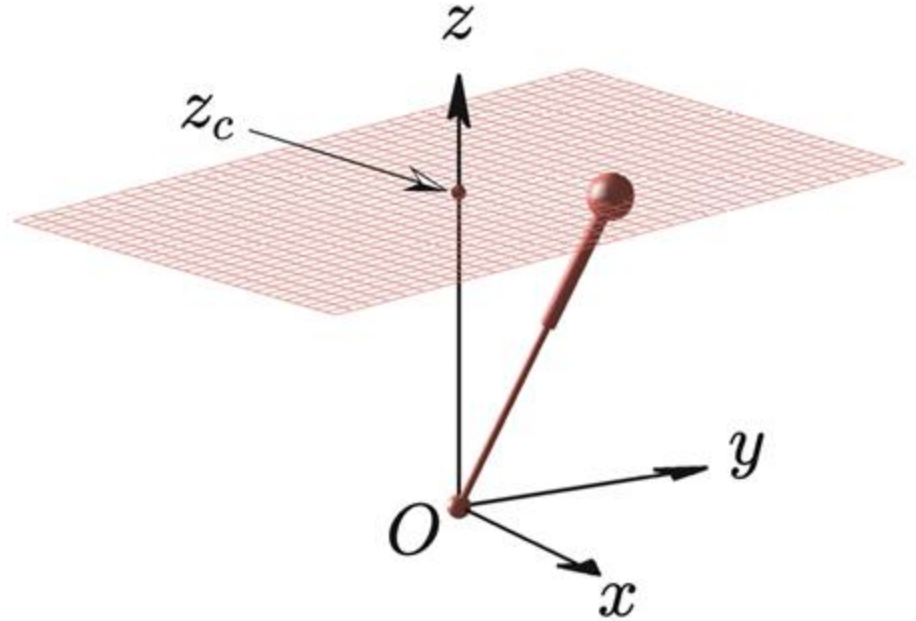
LIP Dynamics

$$\ddot{x} = \frac{g}{z_c} x + \frac{1}{m z_c} u_p$$

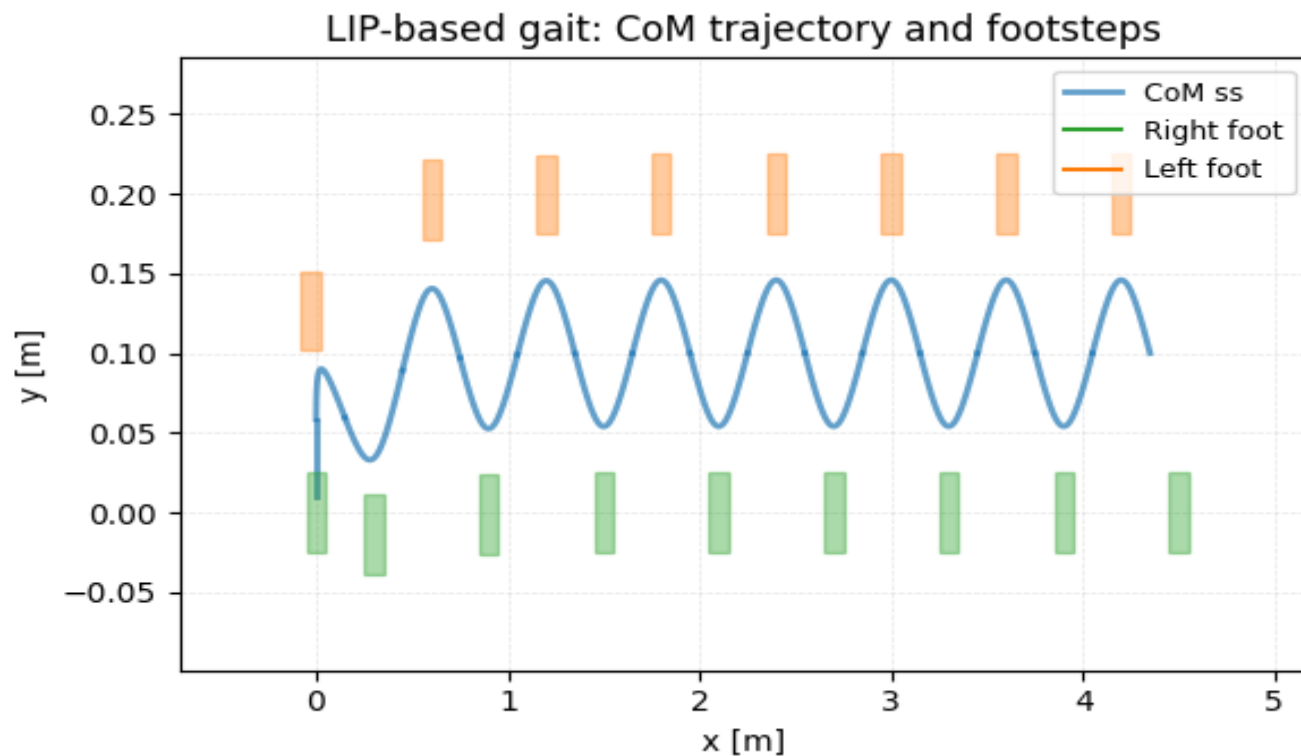
$$\ddot{y} = \frac{g}{z_c} y - \frac{1}{m z_c} u_r$$

$$u_r = f(\tau_r, x, y)$$

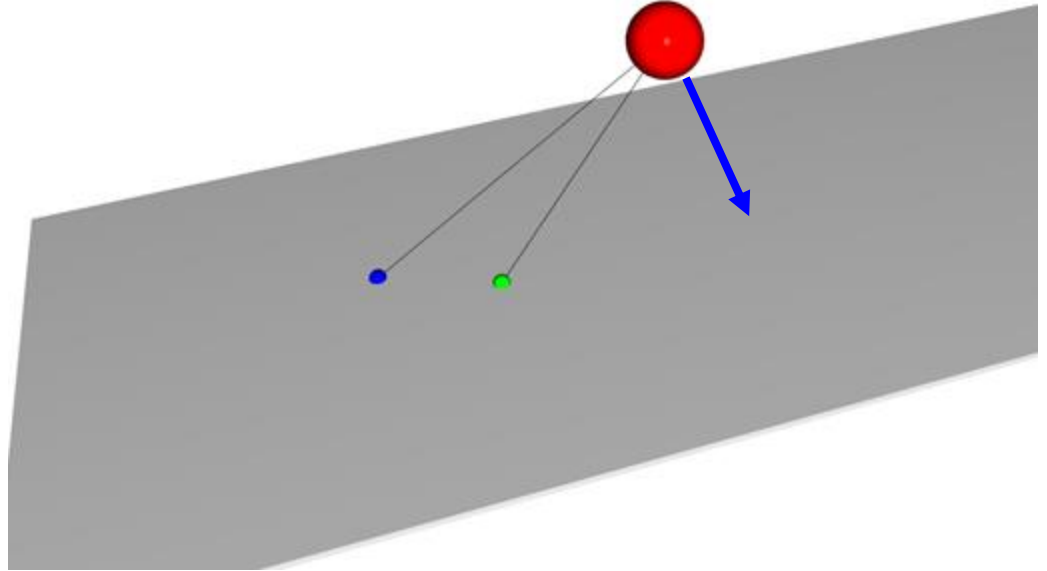
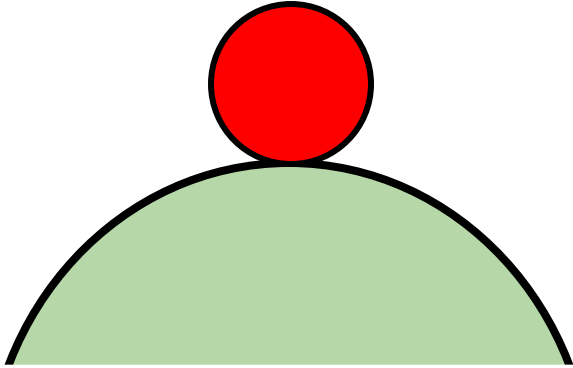
$$u_p = f(\tau_p, x, y)$$



Generating a Walking Pattern

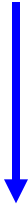


However, the Trajectories are Unstable Equilibriums



Closing the Loop Part 1: Discretization (ZOH)

$$\dot{s} = As + Bu$$



$$s_{k+1} = A_d s_k + B_d u_k$$

$$u = \begin{bmatrix} u_p \\ u_r \end{bmatrix}$$

$$s = \begin{bmatrix} x \\ \dot{x} \\ y \\ \dot{y} \end{bmatrix}$$

$$A_d = e^{A\Delta T}$$

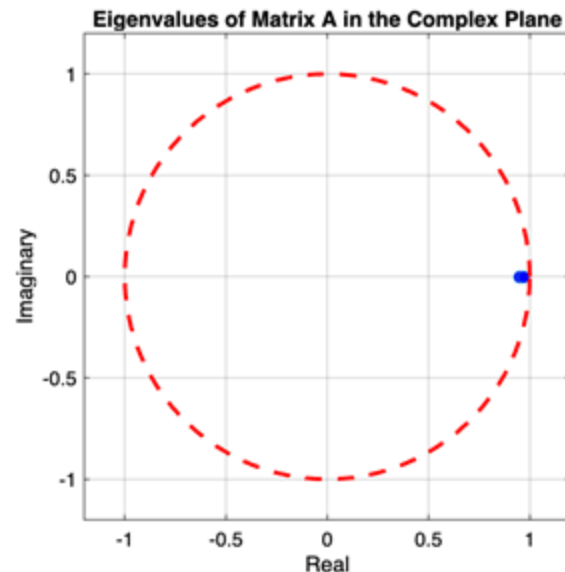
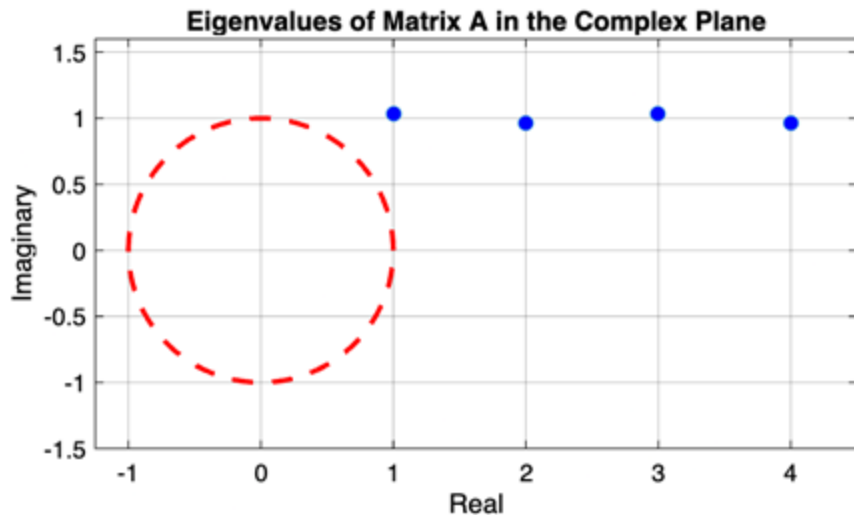
$$B_d = \int_0^{\Delta t} e^{A\tau} d\tau$$

Closing the Loop Part 2: LQR and the Riccati

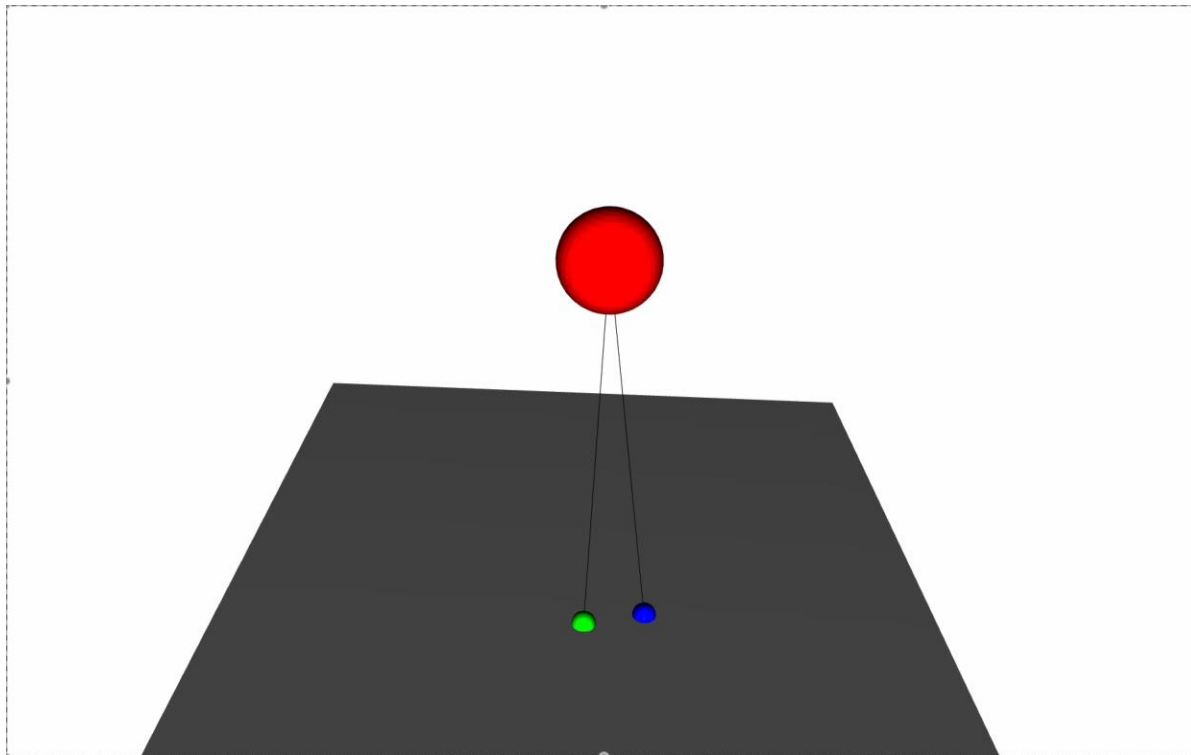
$$P = A_d^T (P - P B_d (R + B_d^T P B_d)^{-1} B_d^T P) A_d + Q$$

$$K = (R + B_d^T P B_d)^{-1} B_d^T P A_d$$

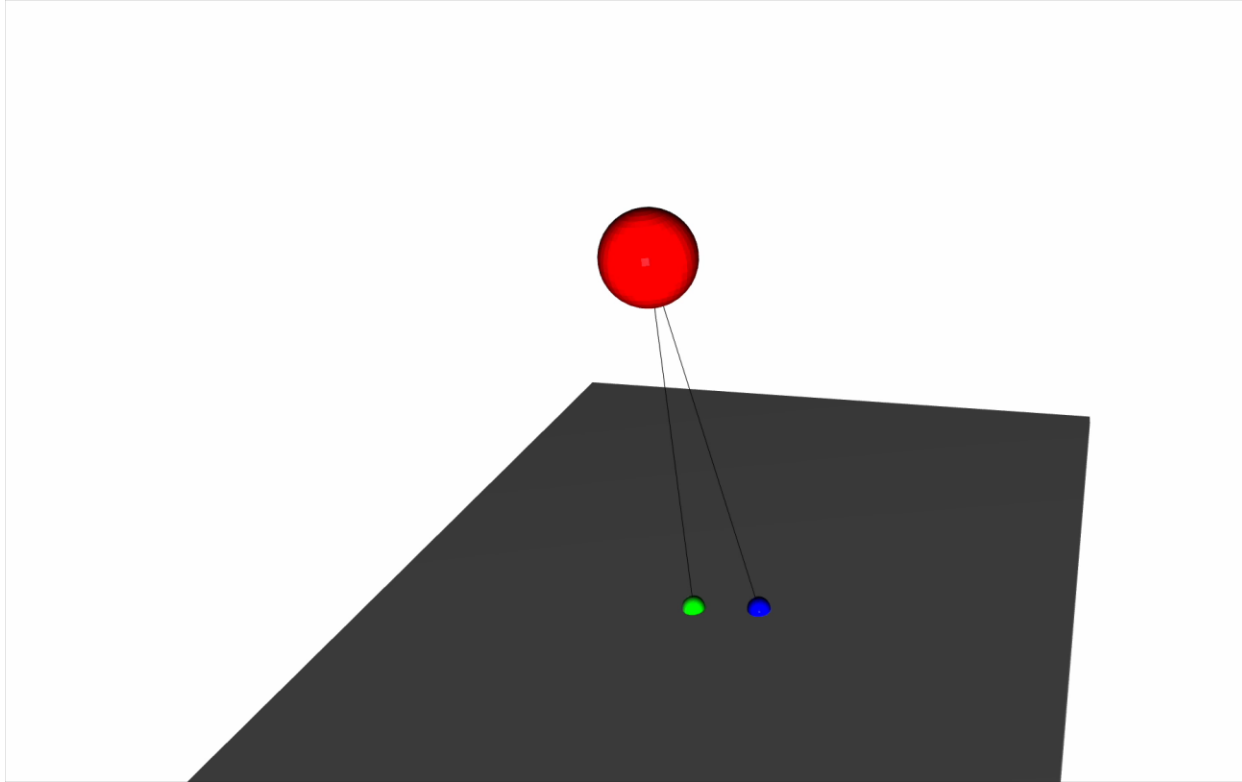
$$u_k = -K(s_k - s_k^{ref})$$



Our Gait is Stable for Random Disturbances!



But NOT Stable for Noisy Footsteps :(



Problem: We don't observe footstep error, and it's difficult to model

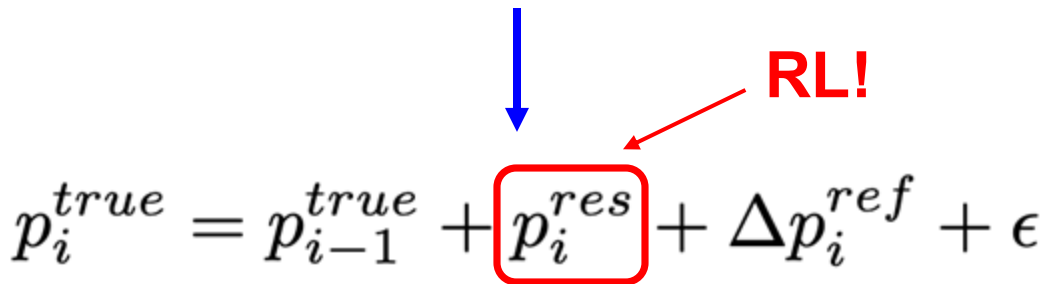
Solution: Use RL to compute a residual for foot placement!

$$p_i^{true} = p_{i-1}^{true} + \Delta p_i^{ref} + \epsilon$$

$$i \in 1, 2, \dots, num_steps$$

$$\Delta p_i^{ref} = p_i^{ref} - p_{i-1}^{ref}$$

$$\epsilon \sim \mathcal{N}(0, 0.02)$$


$$p_i^{true} = p_{i-1}^{true} + \boxed{p_i^{res}} + \Delta p_i^{ref} + \epsilon$$

RL!

LIP RL Setup

Goal: Robustly track reference trajectory even noisy steps

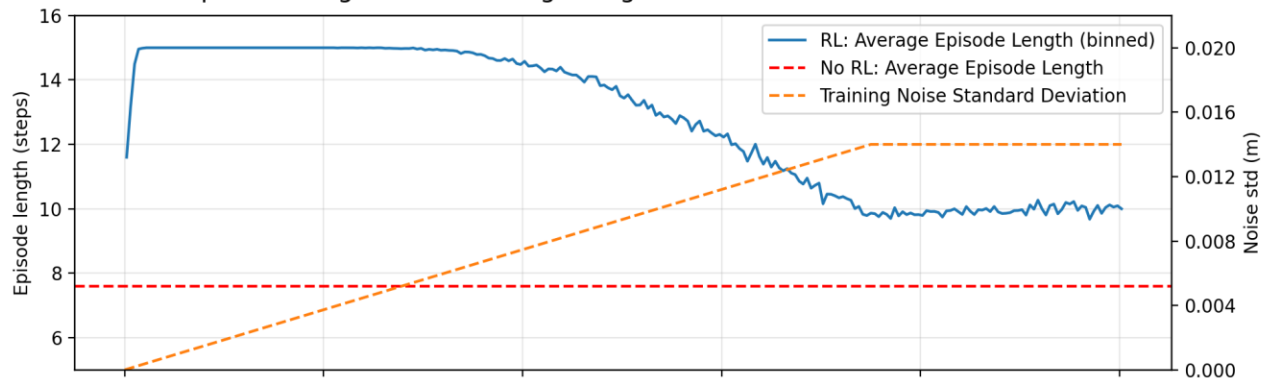
State: $s = [x_{rel}^{belief} \quad \dot{x} \quad y_{rel}^{belief} \quad \dot{y} \quad x_{rel}^{ref} \quad \dot{x}^{ref} \quad y_{rel}^{ref} \quad \dot{y}^{ref} \quad \bar{e}_x \quad \bar{e}_y]^T \in \mathbb{R}^{10}$

Action: $a = [p_x^{res} \quad p_y^{res}]^T \in \mathbb{R}^2$

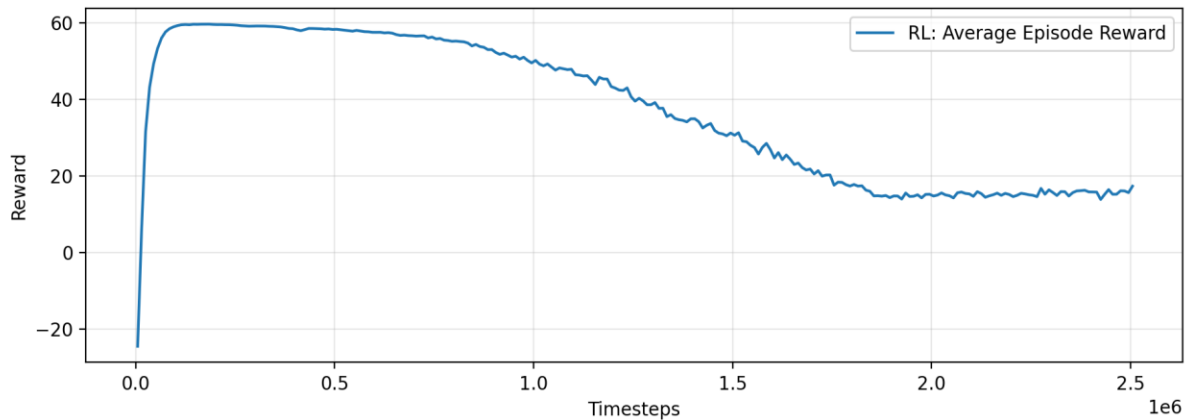
reward Defined using error coordinates from ref gait

Learning Curves

Episode Length Over Training Alongside Baseline and Noise Schedule



Rewards Over Training



3LP

3LP Dynamics

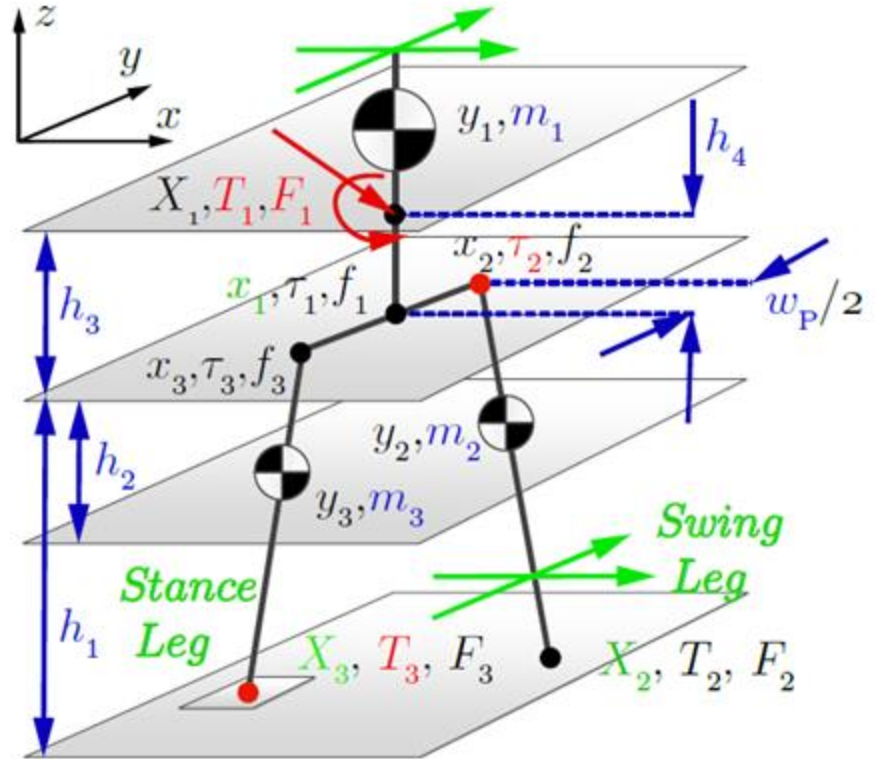
Model

- **Linear! (but ugly)**

$$\ddot{X}(t) = C_X X(t) + C_U U + C_V V(t) + C_W W$$

- **3 Linear Pendulums**
- Constant and ramped ankle and hip torques
- **Closed form solution for gait gen and control**

$$Q(t) = A(t)Q(0) + B(t)R$$



Two Phases: Hybrid Model → Total Stride Map

Total Stride Map (closed form dynamics)

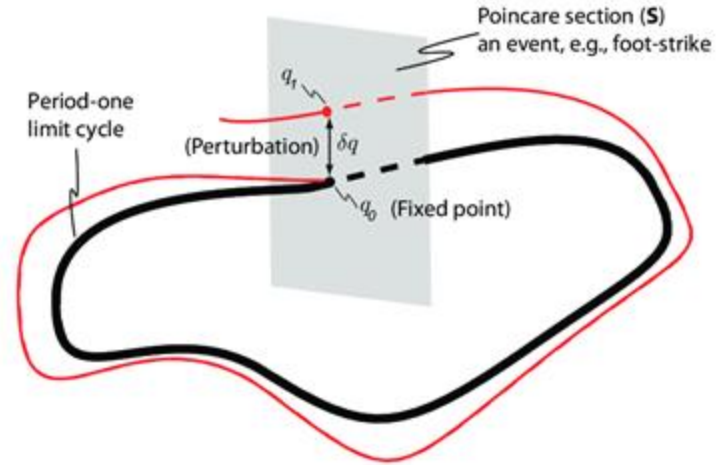
$$Q(t) = A(t)Q(0) + B(t)R$$

$$A(t) = \begin{cases} A^{ds}(t) & \text{if } t \leq t_{ds} \\ A^{ss}(t - t_{ds})A^{ds}(t_{ds}) & \text{if } 0 < t - t_{ds} \leq t_{ss} \end{cases}$$

$$B(t) = \begin{cases} B^{ds}(t) & \text{if } t \leq t_{ds} \\ A^{ss}(t - t_{ds})B^{ds}(t_{ds}) + B^{ss}(t - t_{ds}) & \text{if } 0 < t - t_{ds} \leq t_{ss} \end{cases}$$

Apply a Foot Swap and Centering “Reset” matrix to make every cycle appear the same math-wise.

$$Q_{next_stride} = CS(A_{stride}Q_k + B_{stride}R)$$



↑
If stable,
always look the
same

Use the Stride Map + Nullspace Conditions for Reference Gait

For Symmetric, Periodic Gait, Solve for Limit Cycle over 2 Full Strides (L,R,L)

1. Write out the double stride map

Step 1 Map ($L \rightarrow R$):

$$\mathbf{A}_1 = \mathbf{C}\mathbf{S}\mathbf{A}_{stride}, \quad \mathbf{B}_1 = \mathbf{C}\mathbf{S}\mathbf{B}_{stride}$$

$$\mathbf{Q}_1 = \mathbf{A}_1\mathbf{Q}_0 + \mathbf{B}_1\mathbf{R}$$

Step 2 Map ($R \rightarrow L$):

$$\mathbf{A}_2 = \mathbf{C}\mathbf{S}\mathbf{A}_{stride}, \quad \mathbf{B}_2 = \mathbf{C}\mathbf{S}\mathbf{B}_{stride}\mathbf{J}$$

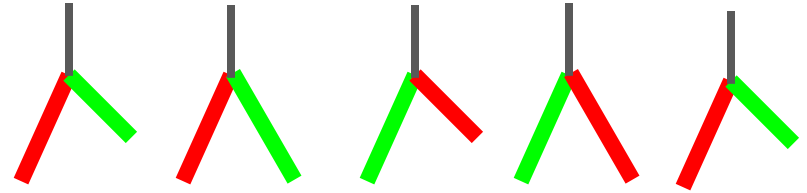
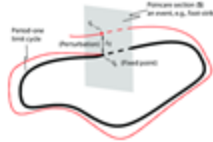
$$\mathbf{Q}_2 = \mathbf{A}_2\mathbf{Q}_1 + \mathbf{B}_2\mathbf{R}$$

The Combined 2-Stride Map:

Substituting \mathbf{Q}_1 into \mathbf{Q}_2 :

$$\mathbf{Q}_2 = \mathbf{A}_2(\mathbf{A}_1\mathbf{Q}_0 + \mathbf{B}_1\mathbf{R}) + \mathbf{B}_2\mathbf{R}$$

$$\mathbf{Q}_2 = \underbrace{(\mathbf{A}_2\mathbf{A}_1)}_{\mathbf{A}_{2stride}}\mathbf{Q}_0 + \underbrace{(\mathbf{A}_2\mathbf{B}_1 + \mathbf{B}_2)}_{\mathbf{B}_{2stride}}\mathbf{R}$$



2. Solve limit cycle torques for double stride map via null space equations:

Periodicity - Rel Pos of limbs are the same at boundaries

$$\mathbf{M}(\mathbf{Q}_0 - \mathbf{Q}_2) = 0$$

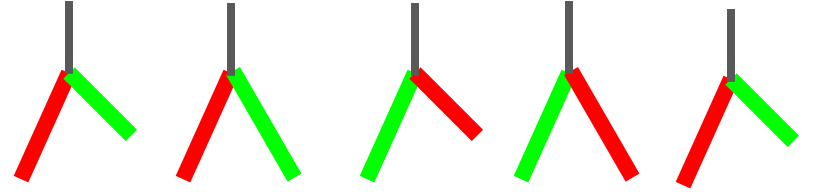
Symmetry - Swing foot has zero vel relative to stance foot at impact

$$\mathbf{C}_{sw}\mathbf{M}\mathbf{Q}_2 = 0$$

No Slip - Stance foot must have zero vel

$$\mathbf{C}_{stance}\mathbf{Q}_2 = 0$$

Use the Stride Map + Nullspace Conditions for Reference Gait



Null Space Solve:

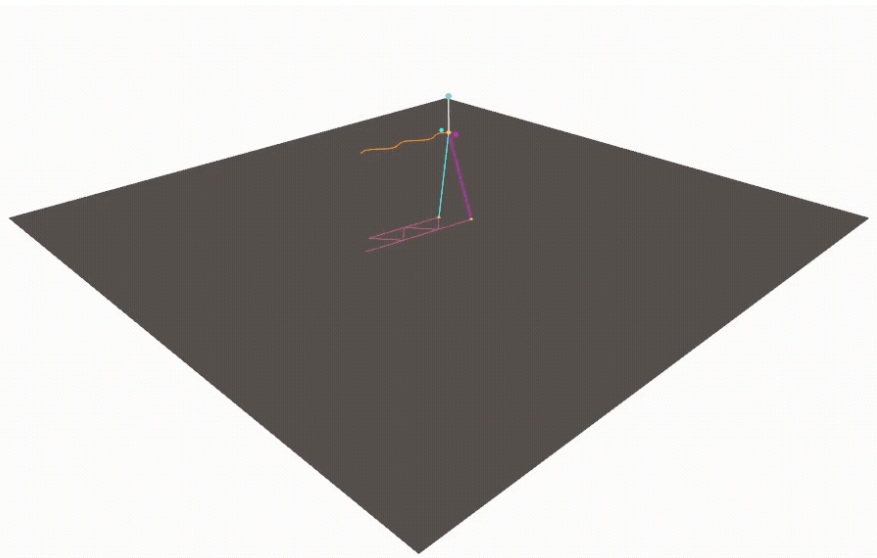
$$\begin{bmatrix} \mathbf{M}(\mathbf{I} - \mathbf{A}_{2stride}) & -\mathbf{M}\mathbf{B}_{2stride} \\ \mathbf{C}_{sw}\mathbf{M}\mathbf{A}_{2stride} & \mathbf{C}_{sw}\mathbf{M}\mathbf{B}_{2stride} \\ \mathbf{C}_{stance}\mathbf{A}_{2stride} & \mathbf{C}_{stance}\mathbf{B}_{2stride} \end{bmatrix} \begin{bmatrix} \mathbf{Q}_0 \\ \mathbf{R} \end{bmatrix} = \mathbf{0}$$

And then we solve a QP to select the minimum energy gait to achieve the target speed, ds and ss parameters.

Reference Gait Generation From Selected Nullspace Criteria

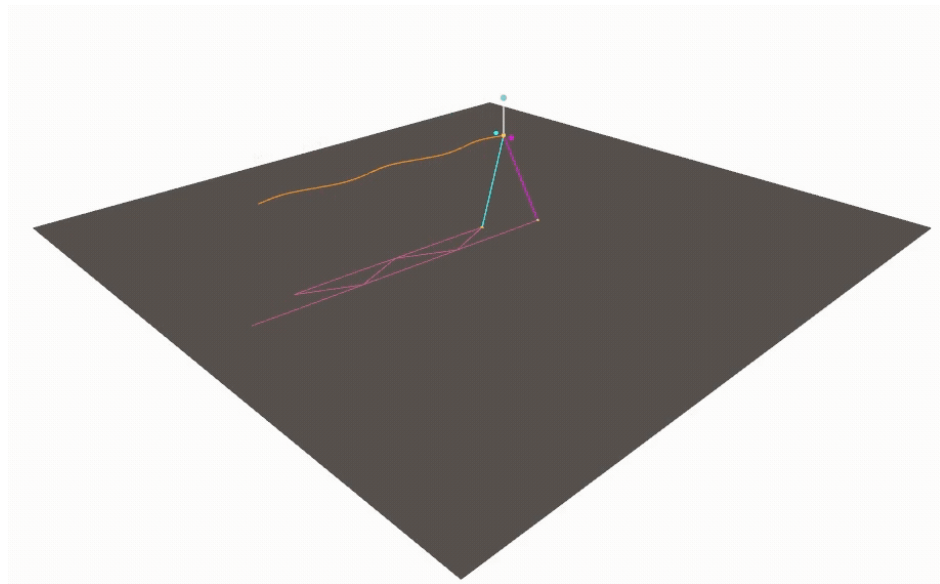
Speed: 0.4 m/s

Step Time: 0.1s ds, 0.6s ss



Speed: 1.2 m/s

Step Time: 0.1s ds, 0.6s ss



DLQR and Time Projection Controller

DLQR - Correction Once Per Stride

Standard discrete LQR procedure, using the **stride map dynamics** (reduced state to moving pelvis and foot via M).

$$z_{raw} = MQ$$

$$z_{k+1} = A_{red}z_k + B_{red}u_k$$

$$J = \sum_{k=0}^{\infty} (z_k^T Q z_k + u_k^T R u_k)$$

$$P = A^T P A - A^T P B (R + B^T P B)^{-1} B^T P A + Q$$

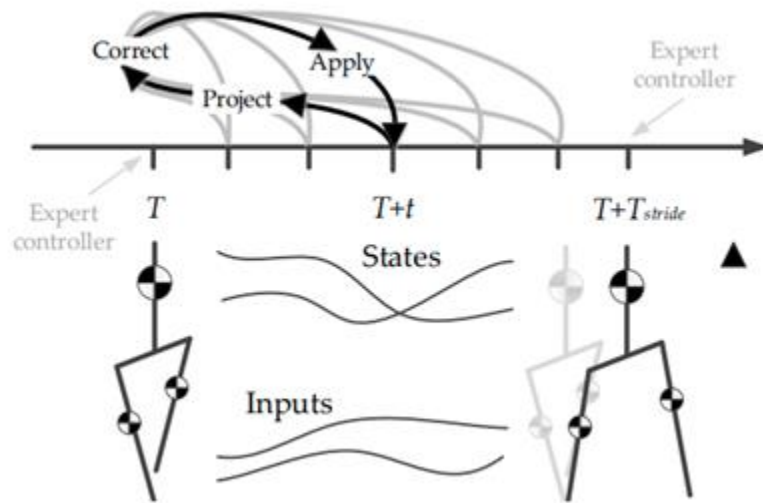
$$u_k = -K z_k$$

Solved via fixed point iteration due to repeated eigenvalues.
Decomposition methods numerically fail.

As we are stabilizing the poincare map, this is once per stride.

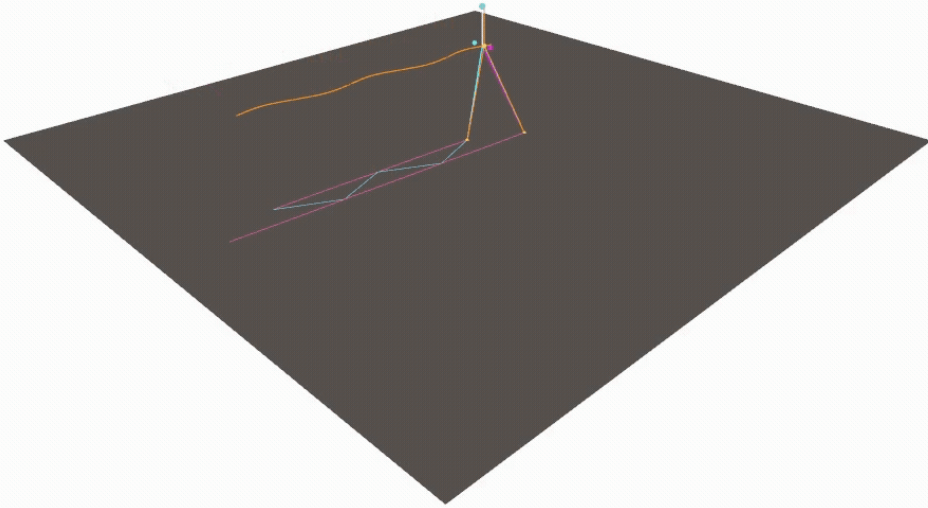
DLQR + Time Projection – Run it back!

Extension of LQR for continuous control by projection of error to boundary for DLQR (Faraji 2017)

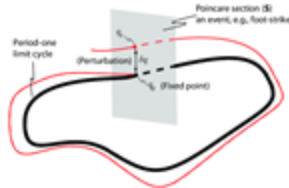
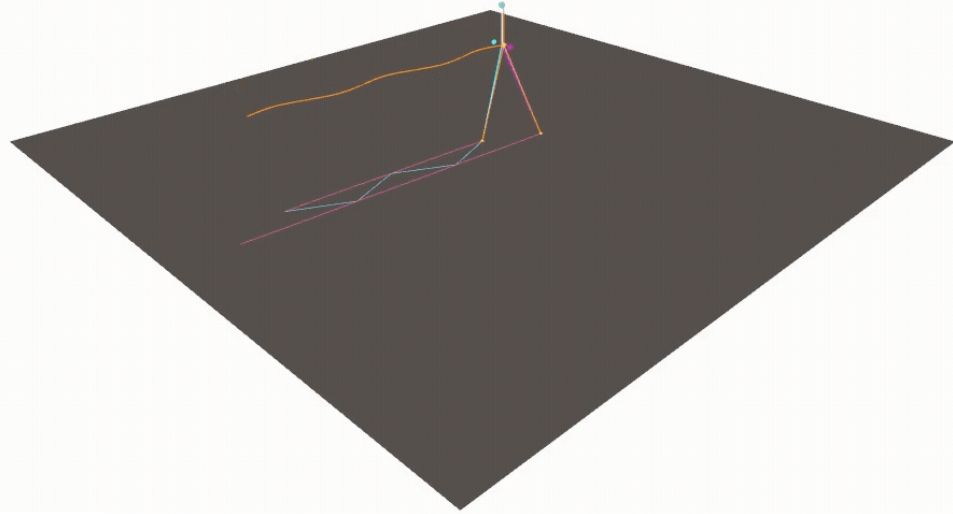


Open Loop vs. DLQR Time Proj. 5 cm Perturbation

Open Loop (1.2m/s)



Time Projection DLQR (1.2m/s)



OK! Stabilizing 3LP is not hopeless.

3LP RL Setting

On Policy Stochastic Actor Critic, TD(0)

3LP RL Setting

(Designed for Actor Critic)

State:

$$s_k = \begin{bmatrix} x_k \\ \phi_k \\ v_{\text{cmd}} \end{bmatrix} \in \mathbb{R}^{10}$$

$x_k \in \mathbb{R}^8$: reduced relative state at control tick k (canonical frame),

$\phi_k \in [0, 1)$: normalized stride phase at tick k (0 = start of stride, 1 = next foot impact),

v_{cmd} : desired forward walking speed (scalar).

Actions: $a_k = \delta p_k \in \mathbb{R}^8 \quad a_{\min} \leq a_k \leq a_{\max}$

Small change torques applied to hip and ankle

Transition Dynamics:

$$Q(t + \Delta t) = A_{\sigma, \Delta t}(t) Q(t) + B_{\sigma, \Delta t}(t) R$$

$$\delta p_k = \begin{bmatrix} \delta U_{h,y} \\ \delta U_{h,x} \\ \delta U_{a,y} \\ \delta U_{a,x} \\ \delta V_{h,y} \\ \delta V_{h,x} \\ \delta V_{a,y} \\ \delta V_{a,x} \end{bmatrix} \in \mathbb{R}^8$$

Parameterization for Actor Critic

Originally Designed for Stride Level Control – But using linear controllers could not stabilize using DLQR alone → aimed to use policy w/more frequent actions

State Error Coordinates Relative to Reference Trajectory:

$$e_k = x_k - x^{\text{ref}}(\phi_k; v_{\text{cmd}}) \in \mathbb{R}^8 \quad \phi = t_{\text{stride}}/T_{\text{stride}}$$

Actor Parameterization: Linear Combination of Error Coordinates → Linear state feedback policy

$$z_k = \begin{bmatrix} e_k \\ \phi_k - 0.5 \\ v_{\text{cmd}} - v_{\text{nom}} \end{bmatrix} \in \mathbb{R}^{10} \quad \phi_a(s_k) = z_k \in \mathbb{R}^{10}$$

$$a_k = \delta p_k = W_a \phi_a(s_k)$$

Critic Parameterization: Linear Combination of Quad Monomials

Critic Parameterization: Linear Combination of Quadratic Monomials

Using error coordinates z_k as for actor parameterization:

Constant term:

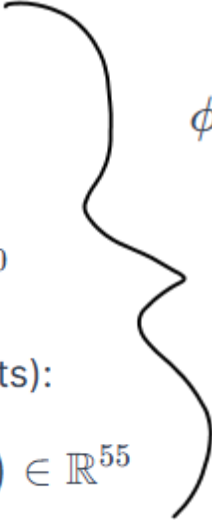
$$\phi_c^{(0)} = 1$$

Linear terms:

$$\phi_c^{(1)} = z_k \in \mathbb{R}^{10}$$

Quadratic terms (unique upper-triangular products):

$$\phi_c^{(2)} = \text{vec}_{\text{upper}}(z_k z_k^\top) \in \mathbb{R}^{55}$$


$$\phi_c(s_k) = \begin{bmatrix} 1 \\ z_k \\ \text{vec}_{\text{upper}}(z_k z_k^\top) \end{bmatrix} \in \mathbb{R}^{66}$$

$$V_\theta(s_k) = \theta^\top \phi_c(s_k)$$

Reward (Cost) Function and Termination

Reward: Formulate like negative LQR Cost

$$c_k = e_k^\top Q_e e_k + q_v e_v^2 + u_k^{\text{corr}\top} R_u u_k^{\text{corr}}$$

$$\tilde{c}_k = \Delta t c_k$$

$$r_k = -\tilde{c}_k$$

State error $e_k \in \mathbb{R}^8$

Speed error $e_v = v_k - v_{\text{cmd}}$

Torque correction

$$u_k^{\text{corr}} = M(\theta_k) a_k \in \mathbb{R}^4$$

Termination: "Fall"

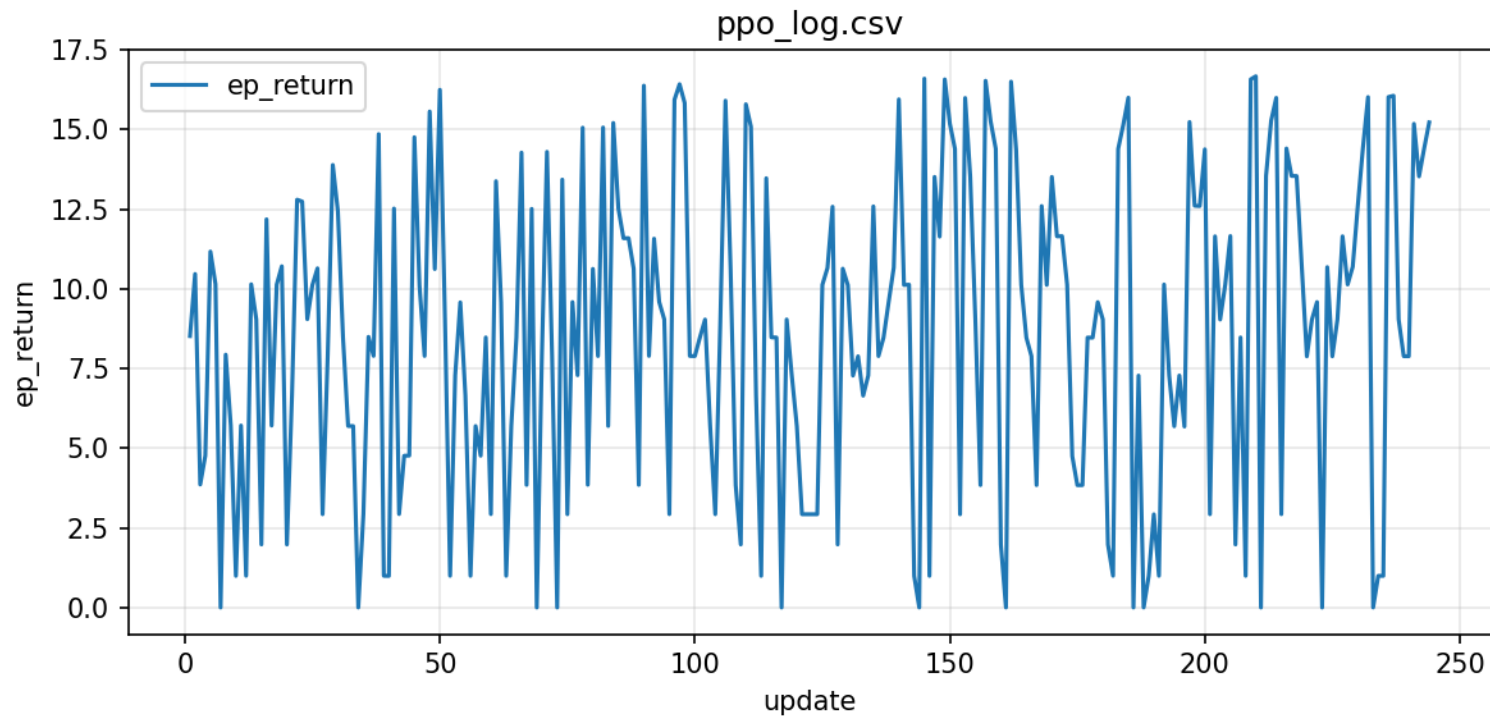
- Pelvis moves too far from stance foot

- Pelvis velocity exceeds a threshold

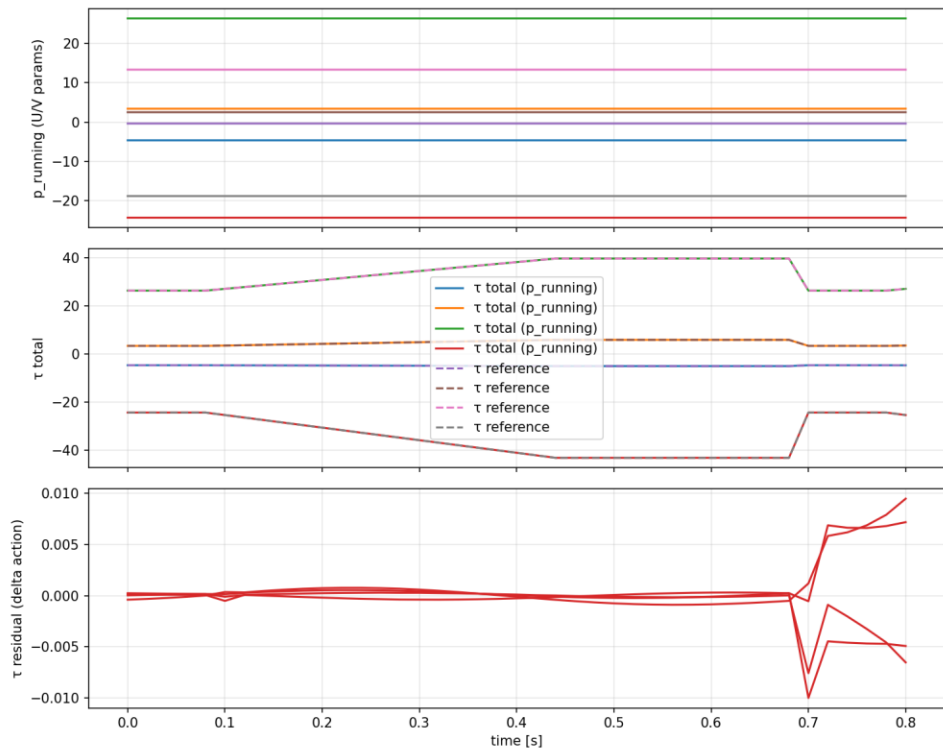
- Large discrete negative reward

In practice – AC stalled,
PPO

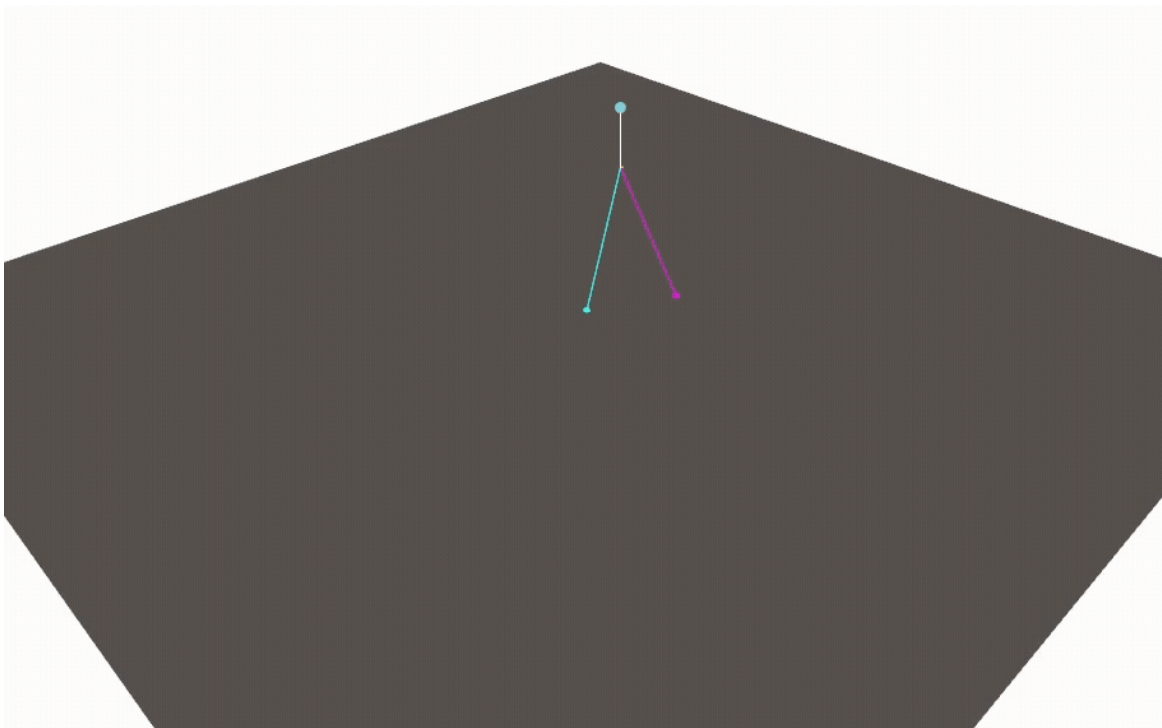
In progress – but PPO also stalled



Residuals and Torques Looks Reasonable, But Policy is Poor



Residuals and Torques Looks Reasonable, But Policy is Poor



Conclusions

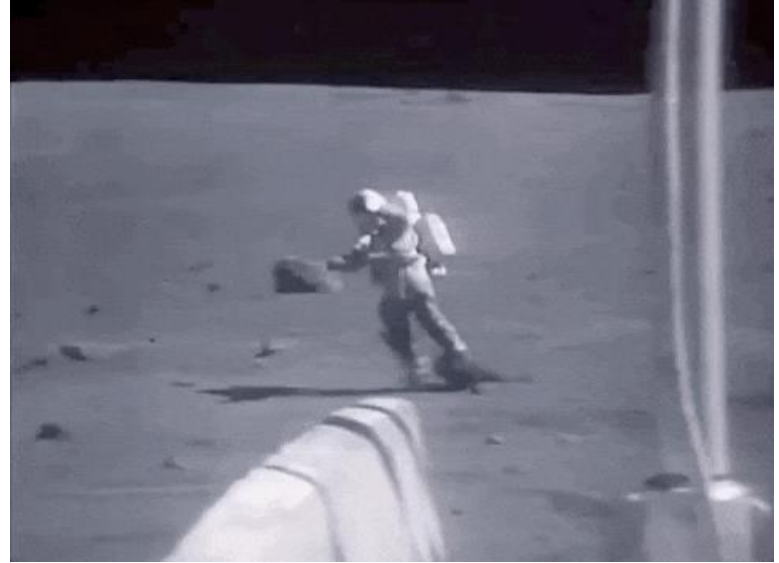
Insights

- Even with all the structure we built up, for walking in this precise manner – learning residuals was still not easy. The unstable system wants to be unstable. This was surprising.
- LQR, even for a difficult Poincare-map system, works decently (though also a bit fragile)
- If for a linear system LQR doesn't work – likely RL won't either. (RL is LQR!)
- For all the cleverness with linear combinations – MLP's and PPO seem just as good.
- We want the RL part to really work for the report!

Lastly,

We had fun and learned a lot!

Thanks all!



Sources

- Kajita, S. (2017). Linear inverted pendulum-based gait. In *Humanoid robotics: A reference* (pp. 1-18). Springer, Dordrecht.
- Faraji, S., & Ijspeert, A. J. (2017). 3LP: A linear 3D-walking model including torso and swing dynamics. *the international journal of robotics research*, 36(4), 436-455.
- Faraji, S., & Ijspeert, A. J. (2016). A new time-projecting controller based on 3LP model to recover intermittent pushes. *arXiv preprint arXiv:1605.03039*.