

# Death on Two Legs

Tyler Christensen, William Lewis, Addison Powell, Jared Smith

## Overview

Our project is modelling how to move a biped across terrain. Given a point of connection and 2 legs acting as double-pendulums, we attempt to minimize the distance between the robots final state and a given point using the angles of each leg as control.

Prior work in optimal control which inspired this project includes Koch et. al's work with walking gait control [KRL16], Friston's work with motor functions [Fri11], and Weng et. al's work with musculoskeletal models [WHA22].

## The Problem

To model the robot's movement, we are treating the legs as double pendulums while in air but inverted double pendulums while on the ground. A  $\theta$  parameter represents the angles of the "thigh" pendulums, and a  $\phi$  parameter represents the angle of the "calf" pendulums. Our state  $\vec{x}$ , represents the position of the top of the legs where they connect.

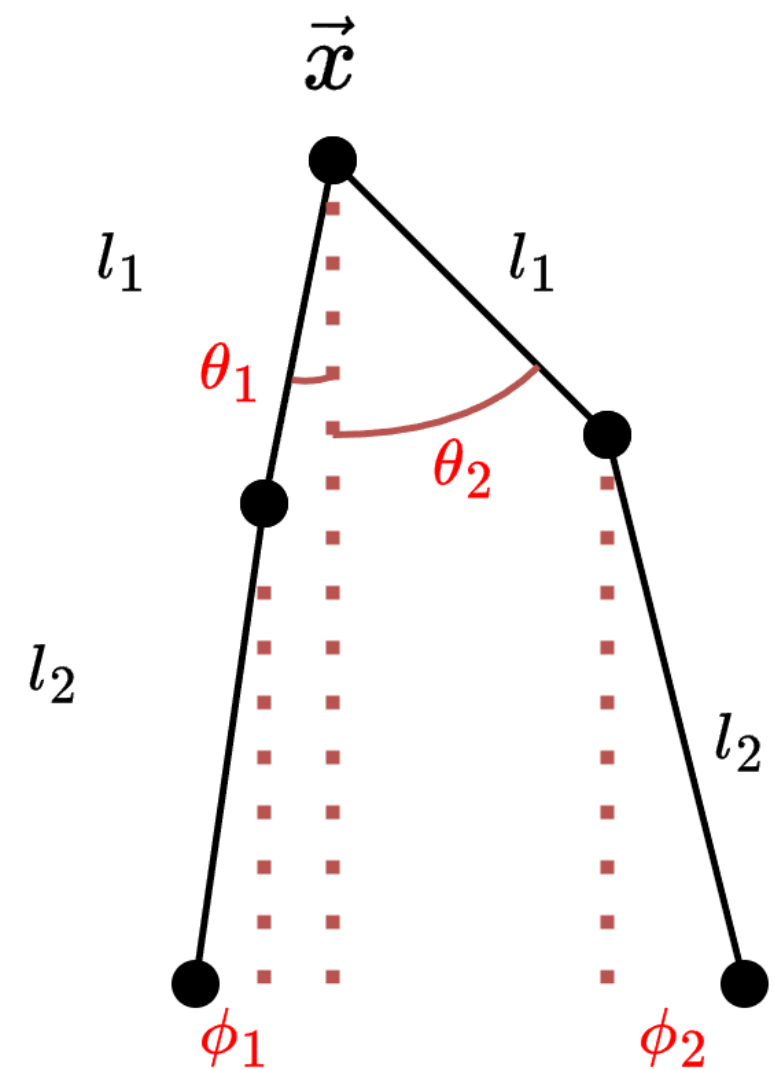


Figure 1: A figure of the system that represents the robot

Our model tries to minimize the amount the legs move, and the time spent moving for  $\vec{x}$  to reach a final state. We are also constraining the possible angles the legs can bend to keep them similar to the standard human range of motion:

$$\begin{aligned} -10^\circ &\leq \theta_i \leq 90^\circ \\ -90^\circ + \theta_i &\leq \phi_i \leq 0^\circ + \theta_i \end{aligned}$$

In the interest of generating a problem we can solve, we decide to abstract it from real world physics, seeing if we can encourage similar behavior mathematically. In other words, we do not consider things such as friction or mass but instead have rules such as the foot being on the ground fixes it in place or neither foot on the ground causes a negative velocity in the y direction.

## Plan of Attack

Given the constraints in the previous section, we have devised the following cost functional, with our desired final position as  $y^*$ .

$$\int_0^{t_f} (\alpha_3 u_1)^2 + (\alpha_3 u_2)^2 + (\alpha_4 u_3)^2 + (\alpha_4 u_4)^2 + (\alpha(y - y^*)^4) dt.$$

Since our control is defined on a compact set one might think the optimal control would end up being bang-bang but at any discrete timestep we want to avoid extreme motion in the legs. To achieve this we normalize the control by the desired maximum amount of motion at a timestep (such as 5 degrees of change from one frame to another) and then square this term leading to the  $(\alpha_3 u_1)^2$  and similar terms in the cost function.

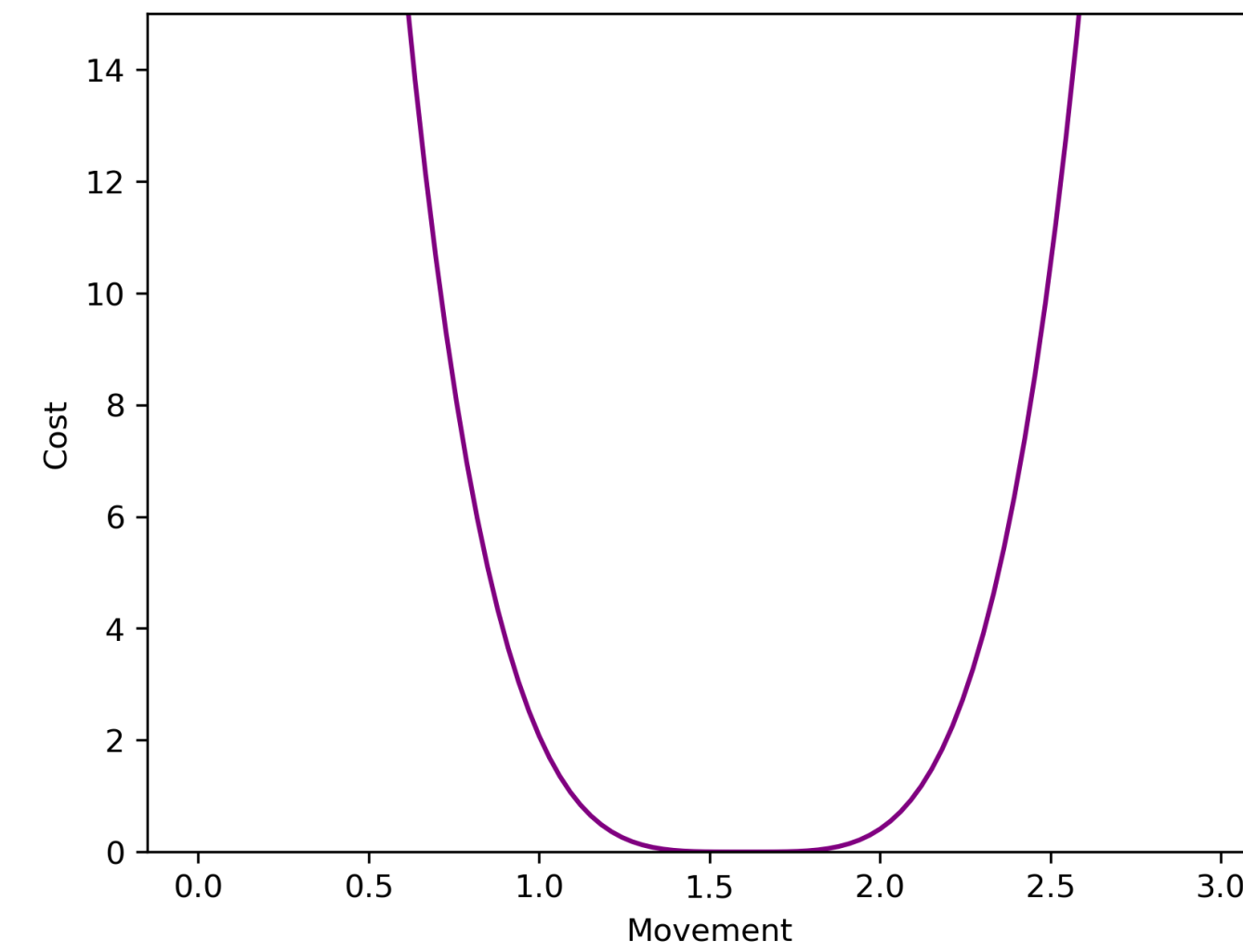


Figure 1: The cost of y being far from the desired value

We then call the following our state:

$$\begin{aligned} x'_i &= l_2(\phi'_i) \cos \phi_i + l_1(\theta'_i) \cos \theta_i \\ y'_i &= l_2(\phi'_i) \sin \phi_i + l_1(\theta'_i) \sin \theta_i. \end{aligned}$$

This comes from finding the position of the top point of a single double pendulum and then taking the derivative with respect to time to try and find the force it exerts on the point  $\vec{x}$ . Also note the derivative terms, we handle these by adding them to our control, making it 8-dimensional.

Then define the following condition:

$$c_i = y - (l_1 \cos(\theta_i) + l_2 \cos(\phi_i)) \leq 0.$$

Therefore, our state equation comes out to

$$\begin{aligned} \vec{x}' &= \begin{bmatrix} x' \\ y' \end{bmatrix} \\ x' &= \begin{cases} x'_1 + x'_2 & \text{if } c_1 \wedge c_2 \\ x'_1 & \text{if } c_1 \\ x'_2 & \text{if } c_2 \\ 0 & \text{else} \end{cases} \quad y' = \begin{cases} y'_1 + y'_2 & \text{if } c_1 \wedge c_2 \\ y'_1 & \text{if } c_1 \\ y'_2 & \text{if } c_2 \\ -\alpha_2 & \text{else} \end{cases} \end{aligned}$$

Given these equations, we can then define the modified Hamiltonian with

$$\begin{aligned} \mathcal{L} &= p_1 \cdot x' + p_2 \cdot y' - (\alpha_3 u_1)^2 - (\alpha_3 u_2)^2 - (\alpha_4 u_3)^2 - (\alpha_4 u_4)^2 - (\alpha(y - y^*)^4 \\ &\quad - \mu_1(\phi_1 - \theta_1) - \mu_2(-\frac{\pi}{2} + \theta_1 - \phi_1) - \mu_3(-\frac{\pi}{18} - \theta_1) - \mu_4(-\frac{\pi}{2} + \theta_1) \\ &\quad - \mu_5(\phi_2 - \theta_2) - \mu_6(-\frac{\pi}{2} + \theta_2 - \phi_2) - \mu_7(-\frac{\pi}{18} - \theta_2) - \mu_8(-\frac{\pi}{2} + \theta_2), \end{aligned}$$

$$\vec{p}' = \begin{bmatrix} -\frac{DL}{Dx} \\ -\frac{DL}{Dy} \end{bmatrix} = \begin{bmatrix} 0 \\ 4(x_1(y - y^*)^3) \end{bmatrix}, \quad \vec{p}'(t_f) = -\frac{\Delta \phi}{\Delta \vec{x}(t_f)} = \vec{0}.$$

Once we have these equations, solving for  $D\mathcal{L} = \vec{0}$ , we find

$$\begin{aligned} \tilde{\theta}'_1 &= \alpha_3 \begin{cases} 1, & \text{if } p_1(l_1 \cos \theta_1) + p_2(l_2 \sin \theta_1) < 0 \\ -1, & \text{if } p_1(l_1 \cos \theta_1) + p_2(l_2 \sin \theta_1) > 0, \end{cases} \\ \tilde{\theta}'_2 &= \alpha_3 \begin{cases} 1, & \text{if } p_1(l_1 \cos \theta_2) + p_2(l_2 \sin \theta_2) < 0 \\ -1, & \text{if } p_1(l_1 \cos \theta_2) + p_2(l_2 \sin \theta_2) > 0, \end{cases} \\ \tilde{\phi}'_1 &= \alpha_4 \begin{cases} 1, & \text{if } p_1(l_1 \cos \phi_1) + p_2(l_2 \sin \phi_1) < 0 \\ -1, & \text{if } p_1(l_1 \cos \phi_1) + p_2(l_2 \sin \phi_1) > 0, \end{cases} \\ \tilde{\phi}'_2 &= \alpha_4 \begin{cases} 1, & \text{if } p_1(l_1 \cos \phi_2) + p_2(l_2 \sin \phi_2) < 0 \\ -1, & \text{if } p_1(l_1 \cos \phi_2) + p_2(l_2 \sin \phi_2) > 0. \end{cases} \end{aligned}$$

## Current State of Affairs

Due to the complicated nature of the problem, our model currently does not produce the desired result when applied via Python. A major problem is defining the evolution of the control when the foot is not on the ground.

With continual improvements to one of the many parameters of the problem, we foresee success with this project. For example, one improvement that we can make to this is changing the ODE that was derived from the cost functions and state equations by making minor changes to the behavior of the cost function or state. Another improvement would be to solve for one step and then feed the final conditions of taking one step into the initial conditions of the next step, thereby discretizing the walk.

From now, our current plan is to iterate on the equations we derived and continually update the code until we determine the optimal control for the bipedal robot. Given we can solve this issue, it can be the basis for more exciting and complicated problems in the future such as different terrain or fine tuning parameters to encourage jumping.

## What We Have Learned

The optimization of physical systems can become extremely difficult. Abstracting real world physics into mathematical rules motivated by cost is helpful in order to make this problem solvable in the context of our curriculum.

## References

[Fri11] Karl Friston, *What Is Optimal about Motor Control?*, Neuron **72** (2011), no. 3, 488–498.

[KRL16] M. W. Koch, M. Ringkamp, and S. Leyendecker, *Discrete Mechanics and Optimal Control of Walking Gaits*, Journal of Computational and Nonlinear Dynamics **12** (2016), no. 021006.

[WHA22] Jiacheng Weng, Ehsan Hashemi, and Arash Arami, *Adaptive Reference Inverse Optimal Control for Natural Walking With Musculoskeletal Models*, IEEE Transactions on Neural Systems and Rehabilitation Engineering **30** (2022), 1567–1575, Conference Name: IEEE Transactions on Neural Systems and Rehabilitation Engineering.