

Soft Robotic Gait Optimization Using Deep Learning

Joel Valley

November 16, 2025

Contents

1	Outline	2
1.1	Background	2
2	Offline Synthetic Data Optimization	4
2.1	Introduction	4
2.2	Paraboloid Dataset	5
2.2.1	Dataset Generation	5
2.2.2	Model Architecture	5
2.3	Spherical Dataset	6
2.3.1	Dataset Generation	6

Chapter 1

Outline

1.1 Background

The development of an untethered soft robot that uses electromagnetic leg actuation to drive locamotive action requires specific input parameters to create a desired gait. The desired optimized variable for this project is the robot's forward velocity. The goal is to not only predict the maximum velocity that the robot can move, but also the input parameters controlling the gait to achieve this desired maximum.

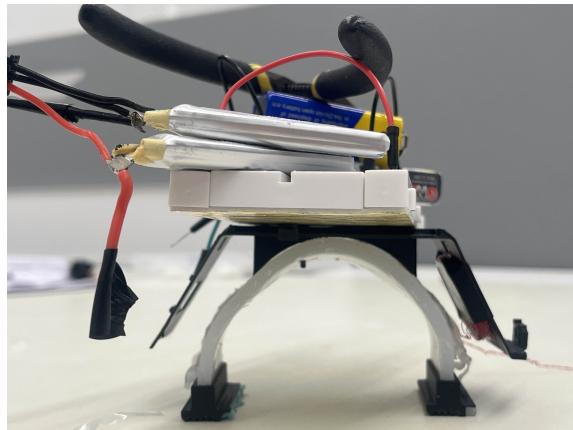


Figure 1.1: Soft Robot Prototype

As shown in Figure 1.1, the soft robot consists of a rigid plastic shell with two electromagnetic solenoids connected on either side. The soft portion of the robot is its silicone legs which are embedded with permanent magnets. At the ends of the

silicone legs are plastic feet with silicone embedded on only half of each foot, which allows for directional friction to ensure the robot is propelled forward. By pulsing the electromagnetic solenoids on each leg, the robot builds oscillatory momentum, and this rocking motion is what propels the prototype forward.

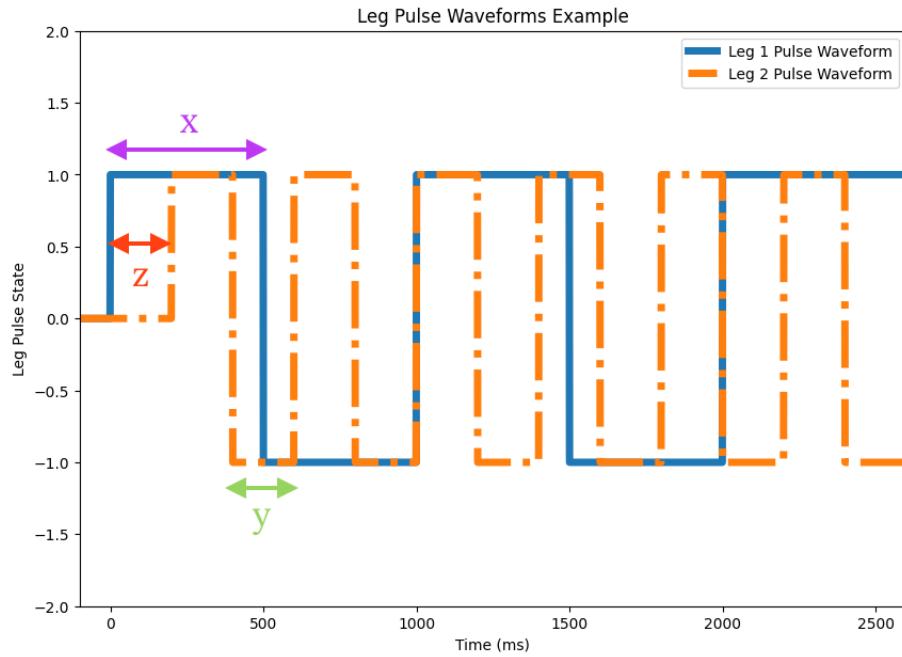


Figure 1.2: Leg Pulse Waveform Example Plot

The current hypothesis is that three main parameters contribute the most to the output velocity: first leg pulse width, second leg pulse width, and the offset between the waveforms. As seen in Figure 1.2, the parameters from here onward denoted as x , y and z will be the two leg pulse durations, and the pulse offset, respectively. Each leg has its own square waveform where the current passes through the solenoid one way, then after the set pulse duration, the motor drivers will be used to change the direction of the current, thus switching the electromagnet's polarity. The offset, z , is the time at which one leg's Pulse waveform lags the other. By varying these parameters, the forward velocity of the robot should change, and there should be some value for each of these parameters that will maximize that velocity. It is important to note that Figure 1.2 is only for illustrative purposes.

Chapter 2

Offline Synthetic Data Optimization

2.1 Introduction

To familiarize ourselves with the concepts of what is to be implemented on the prototype, the first phase of the gait optimization is performing the optimization offline using synthetic data. A 3-dimensional function is used to generate a dataset of speed values (w) using random inputs (x, y, z). This dataset is then passed to a multiple regression model that is to learn how to map the inputs to the outputs. This can then be taken a step further by optimizing an initial guess of the input parameters and performing gradient ascent on the model's predictions to find the maximum value of the model's predictions to locate the maximum predicted output. Once the model's predictions are maximized, the remaining optimized input represents the model's estimate of the parameters that produce the maximum output. This estimated maximum can then be compared to the true maximum of the underlying function (if known), allowing the error and accuracy of the model to be evaluated.

2.2 Paraboloid Dataset

2.2.1 Dataset Generation

The first synthetic dataset used to perform multiple regression is a simple concave-down 4D paraboloid with some shifting to have the vertex at a known, non-zero point.

$$w = -((x - 200)^2 + (y - 150)^2 + (z - 25)^2) + 20 + \epsilon \quad (2.1)$$

where:

w = speed of the robot

x = first leg pulse width

y = second leg pulse width

z = leg pulses waveforms offset

ϵ = random noise $[-0.1, 0.1]$

From the mathematical model in (2.1), it is evident that the maximum value of the paraboloid is $w \in [20 - |\epsilon_{max}|, 20 + |\epsilon_{max}|]$, and occurs at the vertex $(200, 150, 25)$.

Five-hundred sample points were generated using a normal distribution with a mean at the vertex of the paraboloid, and a standard deviation of 70 for each input. These values were chosen so that more datapoints near the maximum of the paraboloid would be generated, allowing the model to more accurately predict the maximum. A medium value for the standard deviation was chosen so that the model will still learn a general pattern for the data and not solely values near the peak.

Before passing the dataset to the neural netowork, it was normalized centered at zero, was split into a 80:20 train/test split, and the data was batchified for better model training.

2.2.2 Model Architecture

The architecture of the neural network used to learn this dataset is a multiple regression neural network, with four linear layers and three GELU layers. The GELU activation function was chosen through experimentation, and was used instead of ReLU to avoid dying ReLU, since our data is normalized and centered at zero, meaning we can have negative values.

2.3 Spherical Dataset

2.3.1 Dataset Generation