Project 1: Gradient-based Algorithms and Differentiable Programming

1. Introduction

Consider a simple formulation of rocket landing where the rocket state x(t) is represented by its distance to the ground d(t) and its velocity v(t), i.e., x(t)=[d(t),v(t)]T, where t specifies time. The control input of the rocket is its acceleration a(t). The discrete-time dynamics follows

$$d(t+1) = d(t) + v(t)\Delta t,$$
$$v(t+1) = v(t) + a(t)\Delta t,$$

where Δt is a time interval. Further, let the closed-loop controller be

$$a(t) = f_\theta(x(t))$$

where fθ(·) is a neural network with parameters θ, which are to be determined through optimization.

For each time step, we assign a loss as a function of the control input and the state: l(x(t),a(t)). In this example, we will simply set l(x(t),a(t))=0 for all t=1,...,T−1, where T is the final time step, and l(x(T),a(T))=||x(T)||2=d(T)2+v(T)2. This loss encourages the rocket to reach d(T)=0 and v(T)=0, which are proper landing conditions.

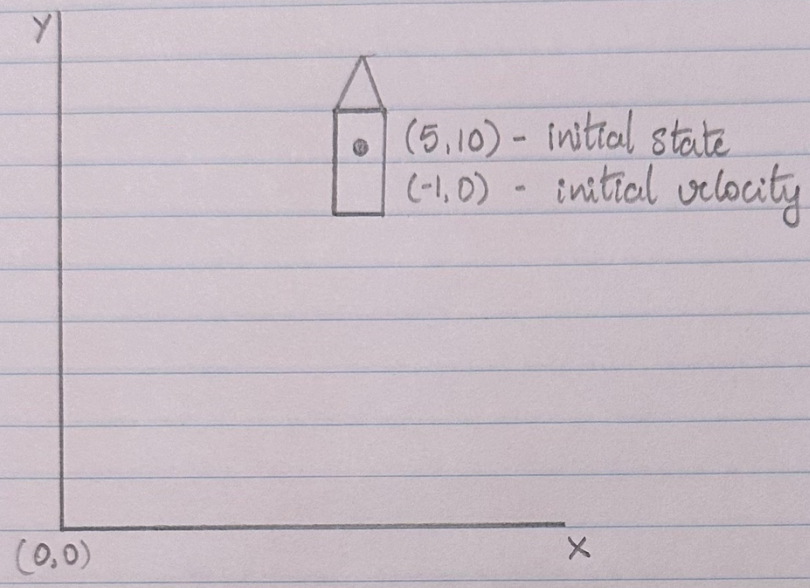
The optimization problem is now formulated as

$$\begin{aligned} \min_\theta \quad & ||x(T)||^2 \\ & d(t+1) = d(t) + v(t)\Delta t, \\ & v(t+1) = v(t) + a(t)\Delta t, \\ & a(t) = f_\theta(x(t)),\ \forall t = 1, \ldots, T-1 \end{aligned}$$

While this problem is constrained, it is easy to see that the objective function can be expressed as a function of x(T−1)and a(T−1), where x(T−1) as a function of x(T−2) and a(T−2), and so on. Thus it is essentially an unconstrained problem with respect to θ.

In the following, we code this problem up with PyTorch, which allows us to only build the forward pass of the loss (i.e., how we move from x(1) to x(2) and all the way to x(T)) and automatically get the gradient ∇θ*l*(x(T),a(T)).

Problem Definition.



The objective of this project is to load the rocket as close to the target landing site at zero velocity.

Objective Function.

$$\min_{\theta} \| s(T) \|^2$$

where, $s(T) = (Y(T), V_y(T), X(T), V_x(T))$

Variables

s : state
t : Time
Y : y coordinates
V_y : Velocity in y direction.
X : x coordinates.
V_x : Velocity in x direction.
a_y : Acceleration in y direction
a_x : Acceleration in x direction

Constraints

$$Y(t+1) = Y(t) + V_y(t)\Delta t.$$

$$V_y(t+1) = V_y(t) + a_y(t)\Delta t - a_y drag(t)\Delta t$$

$$\cancel{x(t)}\ x(t+1) = x(t) + V_x(t)\Delta t$$

$$V_x(t+1) = V_x(t) + a_x(t)\Delta t$$

$$a_y(t), a_x(t) = f_\theta(x(t)), \quad \forall\ t = 1, 2, \ldots T-1$$

Problem formulation

Changing the given environment parameters:

acceleration due to gravity $g = 9.81$

Thrust constant $= 20$

Calculating the drag
I will be make few assumptions for drag effect

Drag force : $D = C_d * A * .0.5 * \gamma * V^2$

$C_d = 0.25$

$A = P_i * R^2 = 34\,m^2$

$\gamma = 850\ Kg/m^3$

$V =$ velocity $=$ Delta state

$m = 549054$ kg.

$D = 0.25 * 34 * 0.5 * 850 *$ delta_state^2

Applying force balance

$m *$ drag deceleration $= D$

Drag deceleration $= 0.00658 *$ delta_state^2

Net Acceleration = Boost Acceleration - Drag Deceleration.

After Re-calculating new velocity
~~delta~~ delta state = Net acceleration * frame time *
t.tensor([0., -1., 0., 0.])

```
import sys
import os
import math
import numpy
import numpy as np
import numpy.linalg
import numpy.linalg as la
from scipy import stats
from scipy.stats import *
from scipy.stats.distributions import *

image = raw_input("enter image: ")
print "Loading image %s..." % image
```

```
# 1-dimensional parameters
name = "p1"
# the name of this experiment
# the name of the output file to write to (without the .mat or .h5 suffix)
MODEL_DIM = 1
# the dimensionality of the model (number of parameters)
# (including hyperparameters, if sampling hyperparameters)
# (including noise parameter, if using a noise parameter)
# (not including the latent function, if using a latent function)
```

```
# MCMC SAMPLER SETTINGS:
# (ignored if using exact inference)
# the MCMC method to use (one of "slice", "hmc", "elliptical")
# (ignored if using exact inference)
MCMC_METHOD = "slice"
# the number of MCMC samples to collect
# (ignored if using exact inference)
NUM_SAMPLES = 1000
# the number of MCMC samples to discard as burn-in
# (ignored if using exact inference)
# (ignored if using a fixed number of MCMC samples)
# (ignored if using slice sampling or elliptical slice sampling)
# (ignored if using HMC with dual-averaging step size adaptation)
# (ignored if using HMC with Riemann manifold adaptation)
# (ignored if using HMC with Gaussian mass matrix adaptation)
# (ignored if using HMC with fixed mass matrix)
# (ignored if using HMC with fixed step size)
# (ignored if using HMC with Riemann manifold HMC)
# (ignored if using HMC with Gaussian mass matrix HMC)
# (ignored if using HMC with fixed mass matrix HMC)
# (ignored if using HMC with fixed step size HMC)
# (ignored if using HMC with Riemann manifold HMC with fixed step size)
# (ignored if using HMC with Gaussian mass matrix HMC with fixed step size)
# (ignored if using HMC with fixed mass matrix HMC with fixed step size)
BURN_IN = 0
# the number of initial MCMC samples to discard
# (ignored if using exact inference)
# (ignored if using a fixed number of MCMC samples)
# (ignored if using slice sampling or elliptical slice sampling)
# (ignored if using HMC with dual-averaging step size adaptation)
# (ignored if using HMC with Riemann manifold adaptation)
# (ignored if using HMC with Gaussian mass matrix adaptation)
# (ignored if using HMC with fixed mass matrix)
# (ignored if using HMC with fixed step size)
# (ignored if using HMC with Riemann manifold HMC)
# (ignored if using HMC with Gaussian mass matrix HMC)
# (ignored if using HMC with fixed mass matrix HMC)
# (ignored if using HMC with fixed step size HMC)
# (ignored if using HMC with Riemann manifold HMC with fixed step size)
# (ignored if using HMC with Gaussian mass matrix HMC with fixed step size)
# (ignored if using HMC with fixed mass matrix HMC with fixed step size)
THIN = 1
# the number of MCMC samples to keep after burn-in
# (ignored if using exact inference)
# (ignored if using a fixed number of MCMC samples)
# (ignored if using slice sampling or elliptical slice sampling)
# (ignored if using HMC with dual-averaging step size adaptation)
# (ignored if using HMC with Riemann manifold adaptation)
# (ignored if using HMC with Gaussian mass matrix adaptation)
# (ignored if using HMC with fixed mass matrix)
# (ignored if using HMC with fixed step size)
# (ignored if using HMC with Riemann manifold HMC)
# (ignored if using HMC with Gaussian mass matrix HMC)
# (ignored if using HMC with fixed mass matrix HMC)
# (ignored if using HMC with fixed step size HMC)
# (ignored if using HMC with Riemann manifold HMC with fixed step size)
# (ignored if using HMC with Gaussian mass matrix HMC with fixed step size)
# (ignored if using HMC with fixed mass matrix HMC with fixed step size)
```

```
# 2-dimensional parameters
name = "p2"
# the name of this experiment
# the name of the output file to write to (without the .mat or .h5 suffix)
MODEL_DIM = 2
# the dimensionality of the model (number of parameters)
# (including hyperparameters, if sampling hyperparameters)
# (including noise parameter, if using a noise parameter)
# (not including the latent function, if using a latent function)
```

```
# MCMC SAMPLER SETTINGS:
# (ignored if using exact inference)
# the MCMC method to use (one of "slice", "hmc", "elliptical")
# (ignored if using exact inference)
MCMC_METHOD = "slice"
# the number of MCMC samples to collect
# (ignored if using exact inference)
NUM_SAMPLES = 1000
# the number of MCMC samples to discard as burn-in
# (ignored if using exact inference)
# (ignored if using a fixed number of MCMC samples)
# (ignored if using slice sampling or elliptical slice sampling)
# (ignored if using HMC with dual-averaging step size adaptation)
# (ignored if using HMC with Riemann manifold adaptation)
# (ignored if using HMC with Gaussian mass matrix adaptation)
# (ignored if using HMC with fixed mass matrix)
# (ignored if using HMC with fixed step size)
# (ignored if using HMC with Riemann manifold HMC)
# (ignored if using HMC with Gaussian mass matrix HMC)
# (ignored if using HMC with fixed mass matrix HMC)
# (ignored if using HMC with fixed step size HMC)
# (ignored if using HMC with Riemann manifold HMC with fixed step size)
# (ignored if using HMC with Gaussian mass matrix HMC with fixed step size)
# (ignored if using HMC with fixed mass matrix HMC with fixed step size)
BURN_IN = 0
# the number of initial MCMC samples to discard
# (ignored if using exact inference)
# (ignored if using a fixed number of MCMC samples)
# (ignored if using slice sampling or elliptical slice sampling)
# (ignored if using HMC with dual-averaging step size adaptation)
# (ignored if using HMC with Riemann manifold adaptation)
# (ignored if using HMC with Gaussian mass matrix adaptation)
# (ignored if using HMC with fixed mass matrix)
# (ignored if using HMC with fixed step size)
# (ignored if using HMC with Riemann manifold HMC)
# (ignored if using HMC with Gaussian mass matrix HMC)
# (ignored if using HMC with fixed mass matrix HMC)
# (ignored if using HMC with fixed step size HMC)
# (ignored if using HMC with Riemann manifold HMC with fixed step size)
# (ignored if using HMC with Gaussian mass matrix HMC with fixed step size)
# (ignored if using HMC with fixed mass matrix HMC with fixed step size)
THIN = 1
# the number of MCMC samples to keep after burn-in
# (ignored if using exact inference)
# (ignored if using a fixed number of MCMC samples)
# (ignored if using slice sampling or elliptical slice sampling)
# (ignored if using HMC with dual-averaging step size adaptation)
# (ignored if using HMC with Riemann manifold adaptation)
# (ignored if using HMC with Gaussian mass matrix adaptation)
# (ignored if using HMC with fixed mass matrix)
# (ignored if using HMC with fixed step size)
# (ignored if using HMC with Riemann manifold HMC)
# (ignored if using HMC with Gaussian mass matrix HMC)
# (ignored if using HMC with fixed mass matrix HMC)
# (ignored if using HMC with fixed step size HMC)
# (ignored if using HMC with Riemann manifold HMC with fixed step size)
# (ignored if using HMC with Gaussian mass matrix HMC with fixed step size)
# (ignored if using HMC with fixed mass matrix HMC with fixed step size)
```

```
# 3-dimensional parameters
name = "p3"
# the name of this experiment
# the name of the output file to write to (without the .mat or .h5 suffix)
MODEL_DIM = 3
# the dimensionality of the model (number of parameters)
# (including hyperparameters, if sampling hyperparameters)
# (including noise parameter, if using a noise parameter)
# (not including the latent function, if using a latent function)
```

```
# MCMC SAMPLER SETTINGS:
# (ignored if using exact inference)
# the MCMC method to use (one of "slice", "hmc", "elliptical")
# (ignored if using exact inference)
MCMC_METHOD = "slice"
# the number of MCMC samples to collect
# (ignored if using exact inference)
NUM_SAMPLES = 1000
# the number of MCMC samples to discard as burn-in
# (ignored if using exact inference)
# (ignored if using a fixed number of MCMC samples)
# (ignored if using slice sampling or elliptical slice sampling)
# (ignored if using HMC with dual-averaging step size adaptation)
# (ignored if using HMC with Riemann manifold adaptation)
# (ignored if using HMC with Gaussian mass matrix adaptation)
# (ignored if using HMC with fixed mass matrix)
# (ignored if using HMC with fixed step size)
# (ignored if using HMC with Riemann manifold HMC)
# (ignored if using HMC with Gaussian mass matrix HMC)
# (ignored if using HMC with fixed mass matrix HMC)
# (ignored if using HMC with fixed step size HMC)
# (ignored if using HMC with Riemann manifold HMC with fixed step size)
# (ignored if using HMC with Gaussian mass matrix HMC with fixed step size)
# (ignored if using HMC with fixed mass matrix HMC with fixed step size)
BURN_IN = 0
# the number of initial MCMC samples to discard
# (ignored if using exact inference)
# (ignored if using a fixed number of MCMC samples)
# (ignored if using slice sampling or elliptical slice sampling)
# (ignored if using HMC with dual-averaging step size adaptation)
# (ignored if using HMC with Riemann manifold adaptation)
# (ignored if using HMC with Gaussian mass matrix adaptation)
# (ignored if using HMC with fixed mass matrix)
# (ignored if using HMC with fixed step size)
# (ignored if using HMC with Riemann manifold HMC)
# (ignored if using HMC with Gaussian mass matrix HMC)
# (ignored if using HMC with fixed mass matrix HMC)
# (ignored if using HMC with fixed step size HMC)
# (ignored if using HMC with Riemann manifold HMC with fixed step size)
# (ignored if using HMC with Gaussian mass matrix HMC with fixed step size)
# (ignored if using HMC with fixed mass matrix HMC with fixed step size)
THIN = 1
# the number of MCMC samples to keep after burn-in
# (ignored if using exact inference)
# (ignored if using a fixed number of MCMC samples)
# (ignored if using slice sampling or elliptical slice sampling)
# (ignored if using HMC with dual-averaging step size adaptation)
# (ignored if using HMC with Riemann manifold adaptation)
# (ignored if using HMC with Gaussian mass matrix adaptation)
# (ignored if using HMC with fixed mass matrix)
# (ignored if using HMC with fixed step size)
# (ignored if using HMC with Riemann manifold HMC)
# (ignored if using HMC with Gaussian mass matrix HMC)
# (ignored if using HMC with fixed mass matrix HMC)
# (ignored if using HMC with fixed step size HMC)
# (ignored if using HMC with Riemann manifold HMC with fixed step size)
# (ignored if using HMC with Gaussian mass matrix HMC with fixed step size)
# (ignored if using HMC with fixed mass matrix HMC with fixed step size)
```

```
# 4-dimensional parameters
name = "p4"
# the name of this experiment
# the name of the output file to write to (without the .mat or .h5 suffix)
MODEL_DIM = 4
# the dimensionality of the model (number of parameters)
# (including hyperparameters, if sampling hyperparameters)
# (including noise parameter, if using a noise parameter)
# (not including the latent function, if using a latent function)
```

```
# MCMC SAMPLER SETTINGS:
# (ignored if using exact inference)
# the MCMC method to use (one of "slice", "hmc", "elliptical")
# (ignored if using exact inference)
MCMC_METHOD = "slice"
# the number of MCMC samples to collect
# (ignored if using exact inference)
NUM_SAMPLES = 1000
# the number of MCMC samples to discard as burn-in
# (ignored if using exact inference)
# (ignored if using a fixed number of MCMC samples)
# (ignored if using slice sampling or elliptical slice sampling)
# (ignored if using HMC with dual-averaging step size adaptation)
# (ignored if using HMC with Riemann manifold adaptation)
# (ignored if using HMC with Gaussian mass matrix adaptation)
# (ignored if using HMC with fixed mass matrix)
# (ignored if using HMC with fixed step size)
# (ignored if using HMC with Riemann manifold HMC)
# (ignored if using HMC with Gaussian mass matrix HMC)
# (ignored if using HMC with fixed mass matrix HMC)
# (ignored if using HMC with fixed step size HMC)
# (ignored if using HMC with Riemann manifold HMC with fixed step size)
# (ignored if using HMC with Gaussian mass matrix HMC with fixed step size)
# (ignored if using HMC with fixed mass matrix HMC with fixed step size)
BURN_IN = 0
# the number of initial MCMC samples to discard
# (ignored if using exact inference)
# (ignored if using a fixed number of MCMC samples)
# (ignored if using slice sampling or elliptical slice sampling)
# (ignored if using HMC with dual-averaging step size adaptation)
# (ignored if using HMC with Riemann manifold adaptation)
# (ignored if using HMC with Gaussian mass matrix adaptation)
# (ignored if using HMC with fixed mass matrix)
# (ignored if using HMC with fixed step size)
# (ignored if using HMC with Riemann manifold HMC)
# (ignored if using HMC with Gaussian mass matrix HMC)
# (ignored if using HMC with fixed mass matrix HMC)
# (ignored if using HMC with fixed step size HMC)
# (ignored if using HMC with Riemann manifold HMC with fixed step size)
# (ignored if using HMC with Gaussian mass matrix HMC with fixed step size)
# (ignored if using HMC with fixed mass matrix HMC with fixed step size)
THIN = 1
# the number of MCMC samples to keep after burn-in
# (ignored if using exact inference)
# (ignored if using a fixed number of MCMC samples)
# (ignored if using slice sampling or elliptical slice sampling)
# (ignored if using HMC with dual-averaging step size adaptation)
# (ignored if using HMC with Riemann manifold adaptation)
# (ignored if using HMC with Gaussian mass matrix adaptation)
# (ignored if using HMC with fixed mass matrix)
# (ignored if using HMC with fixed step size)
# (ignored if using HMC with Riemann manifold HMC)
# (ignored if using HMC with Gaussian mass matrix HMC)
# (ignored if using HMC with fixed mass matrix HMC)
# (ignored if using HMC with fixed step size HMC)
# (ignored if using HMC with Riemann manifold HMC with fixed step size)
# (ignored if using HMC with Gaussian mass matrix HMC with fixed step size)
# (ignored if using HMC with fixed mass matrix HMC with fixed step size)
```

```
# 5-dimensional parameters
name = "p5"
# the name of this experiment
# the name of the output file to write to (without the .mat or .h5 suffix)
MODEL_DIM = 5
# the dimensionality of the model (number of parameters)
# (including hyperparameters, if sampling hyperparameters)
# (including noise parameter, if using a noise parameter)
# (not including the latent function, if using a latent function)
```

```
# MCMC SAMPLER SETTINGS:
# (ignored if using exact inference)
# the MCMC method to use (one of "slice", "hmc", "elliptical")
# (ignored if using exact inference)
MCMC_METHOD = "slice"
# the number of MCMC samples to collect
# (ignored if using exact inference)
NUM_SAMPLES = 1000
# the number of MCMC samples to discard as burn-in
# (ignored if using exact inference)
# (ignored if using a fixed number of MCMC samples)
# (ignored if using slice sampling or elliptical slice sampling)
# (ignored if using HMC with dual-averaging step size adaptation)
# (ignored if using HMC with Riemann manifold adaptation)
# (ignored if using HMC with Gaussian mass matrix adaptation)
# (ignored if using HMC with fixed mass matrix)
# (ignored if using HMC with fixed step size)
# (ignored if using HMC with Riemann manifold HMC)
# (ignored if using HMC with Gaussian mass matrix HMC)
# (ignored if using HMC with fixed mass matrix HMC)
# (ignored if using HMC with fixed step size HMC)
# (ignored if using HMC with Riemann manifold HMC with fixed step size)
# (ignored if using HMC with Gaussian mass matrix HMC with fixed step size)
# (ignored if using HMC with fixed mass matrix HMC with fixed step size)
BURN_IN = 0
# the number of initial MCMC samples to discard
# (ignored if using exact inference)
# (ignored if using a fixed number of MCMC samples)
# (ignored if using slice sampling or elliptical slice sampling)
# (ignored if using HMC with dual-averaging step size adaptation)
# (ignored if using HMC with Riemann manifold adaptation)
# (ignored if using HMC with Gaussian mass matrix adaptation)
# (ignored if using HMC with fixed mass matrix)
# (ignored if using HMC with fixed step size)
# (ignored if using HMC with Riemann manifold HMC)
# (ignored if using HMC with Gaussian mass matrix HMC)
# (ignored if using HMC with fixed mass matrix HMC)
# (ignored if using HMC with fixed step size HMC)
# (ignored if using HMC with Riemann manifold HMC with fixed step size)
# (ignored if using HMC with Gaussian mass matrix HMC with fixed step size)
# (ignored if using HMC with fixed mass matrix HMC with fixed step size)
THIN = 1
# the number of MCMC samples to keep after burn-in
# (ignored if using exact inference)
# (ignored if using a fixed number of MCMC samples)
# (ignored if using slice sampling or elliptical slice sampling)
# (ignored if using HMC with dual-averaging step size adaptation)
# (ignored if using HMC with Riemann manifold adaptation)
# (ignored if using HMC with Gaussian mass matrix adaptation)
# (ignored if using HMC with fixed mass matrix)
# (ignored if using HMC with fixed step size)
# (ignored if using HMC with Riemann manifold HMC)
# (ignored if using HMC with Gaussian mass matrix HMC)
# (ignored if using HMC with fixed mass matrix HMC)
# (ignored if using HMC with fixed step size HMC)
# (ignored if using HMC with Riemann manifold HMC with fixed step size)
# (ignored if using HMC with Gaussian mass matrix HMC with fixed step size)
# (ignored if using HMC with fixed mass matrix HMC with fixed step size)
```

```
# 6-dimensional parameters
name = "p6"
# the name of this experiment
# the name of the output file to write to (without the .mat or .h5 suffix)
MODEL_DIM = 6
# the dimensionality of the model (number of parameters)
# (including hyperparameters, if sampling hyperparameters)
# (including noise parameter, if using a noise parameter)
# (not including the latent function, if using a latent function)
```

```
# MCMC SAMPLER SETTINGS:
# (ignored if using exact inference)
# the MCMC method to use (one of "slice", "hmc", "elliptical")
# (ignored if using exact inference)
MCMC_METHOD = "slice"
# the number of MCMC samples to collect
# (ignored if using exact inference)
NUM_SAMPLES = 1000
# the number of MCMC samples to discard as burn-in
# (ignored if using exact inference)
# (ignored if using a fixed number of MCMC samples)
# (ignored if using slice sampling or elliptical slice sampling)
# (ignored if using HMC with dual-averaging step size adaptation)
# (ignored if using HMC with Riemann manifold adaptation)
# (ignored if using HMC with Gaussian mass matrix adaptation)
# (ignored if using HMC with fixed mass matrix)
# (ignored if using HMC with fixed step size)
# (ignored if using HMC with Riemann manifold HMC)
# (ignored if using HMC with Gaussian mass matrix HMC)
# (ignored if using HMC with fixed mass matrix HMC)
# (ignored if using HMC with fixed step size HMC)
# (ignored if using HMC with Riemann manifold HMC with fixed step size)
# (ignored if using HMC with Gaussian mass matrix HMC with fixed step size)
# (ignored if using HMC with fixed mass matrix HMC with fixed step size)
BURN_IN = 0
# the number of initial MCMC samples to discard
# (ignored if using exact inference)
# (ignored if using a fixed number of MCMC samples)
# (ignored if using slice sampling or elliptical slice sampling)
# (ignored if using HMC with dual-averaging step size adaptation)
# (ignored if using HMC with Riemann manifold adaptation)
# (ignored if using HMC with Gaussian mass matrix adaptation)
# (ignored if using HMC with fixed mass matrix)
# (ignored if using HMC with fixed step size)
# (ignored if using HMC with Riemann manifold HMC)
# (ignored if using HMC with Gaussian mass matrix HMC)
# (ignored if using HMC with fixed mass matrix HMC)
# (ignored if using HMC with fixed step size HMC)
# (ignored if using HMC with Riemann manifold HMC with fixed step size)
# (ignored if using HMC with Gaussian mass matrix HMC with fixed step size)
# (ignored if using HMC with fixed mass matrix HMC with fixed step size)
THIN = 1
# the number of MCMC samples to keep after burn-in
# (ignored if using exact inference)
# (ignored if using a fixed number of MCMC samples)
# (ignored if using slice sampling or elliptical slice sampling)
# (ignored if using HMC with dual-averaging step size adaptation)
# (ignored if using HMC with Riemann manifold adaptation)
# (ignored if using HMC with Gaussian mass matrix adaptation)
# (ignored if using HMC with fixed mass matrix)
# (ignored if using HMC with fixed step size)
# (ignored if using HMC with Riemann manifold HMC)
# (ignored if using HMC with Gaussian mass matrix HMC)
# (ignored if using HMC with fixed mass matrix HMC)
# (ignored if using HMC with fixed step size HMC)
# (ignored if using HMC with Riemann manifold HMC with fixed step size)
# (ignored if using HMC with Gaussian mass matrix HMC with fixed step size)
# (ignored if using HMC with fixed mass matrix HMC with fixed step size)
```

```
# 7-dimensional parameters
name = "p7"
# the name of this experiment
# the name of the output file to write to (without the .mat or .h5 suffix)
MODEL_DIM = 7
# the dimensionality of the model (number of parameters)
# (including hyperparameters, if sampling hyperparameters)
# (including noise parameter, if using a noise parameter)
# (not including the latent function, if using a latent function)
```

```
# MCMC SAMPLER SETTINGS:
# (ignored if using exact inference)
# the MCMC method to use (one of "slice", "hmc", "elliptical")
# (ignored if using exact inference)
MCMC_METHOD = "slice"
# the number of MCMC samples to collect
# (ignored if using exact inference)
NUM_SAMPLES = 1000
# the number of MCMC samples to discard as burn-in
# (ignored if using exact inference)
# (ignored if using a fixed number of MCMC samples)
# (ignored if using slice sampling or elliptical slice sampling)
# (ignored if using HMC with dual-averaging step size adaptation)
# (ignored if using HMC with Riemann manifold adaptation)
# (ignored if using HMC with Gaussian mass matrix adaptation)
# (ignored if using HMC with fixed mass matrix)
# (ignored if using HMC with fixed step size)
# (ignored if using HMC with Riemann manifold HMC)
# (ignored if using HMC with Gaussian mass matrix HMC)
# (ignored if using HMC with fixed mass matrix HMC)
# (ignored if using HMC with fixed step size HMC)
# (ignored if using HMC with Riemann manifold HMC with fixed step size)
# (ignored if using HMC with Gaussian mass matrix HMC with fixed step size)
# (ignored if using HMC with fixed mass matrix HMC with fixed step size)
BURN_IN = 0
# the number of initial MCMC samples to discard
# (ignored if using exact inference)
# (ignored if using a fixed number of MCMC samples)
# (ignored if using slice sampling or elliptical slice sampling)
# (ignored if using HMC with dual-averaging step size adaptation)
# (ignored if using HMC with Riemann manifold adaptation)
# (ignored if using HMC with Gaussian mass matrix adaptation)
# (ignored if using HMC with fixed mass matrix)
# (ignored if using HMC with fixed step size)
# (ignored if using HMC with Riemann manifold HMC)
# (ignored if using HMC with Gaussian mass matrix HMC)
# (ignored if using HMC with fixed mass matrix HMC)
# (ignored if using HMC with fixed step size HMC)
# (ignored if using HMC with Riemann manifold HMC with fixed step size)
# (ignored if using HMC with Gaussian mass matrix HMC with fixed step size)
# (ignored if using HMC with fixed mass matrix HMC with fixed step size)
THIN = 1
# the number of MCMC samples to keep after burn-in
# (ignored if using exact inference)
# (ignored if using a fixed number of MCMC samples)
# (ignored if using slice sampling or elliptical slice sampling)
# (ignored if using HMC with dual-averaging step size adaptation)
# (ignored if using HMC with Riemann manifold adaptation)
# (ignored if using HMC with Gaussian mass matrix adaptation)
# (ignored if using HMC with fixed mass matrix)
# (ignored if using HMC with fixed step size)
# (ignored if using HMC with Riemann manifold HMC)
# (ignored if using HMC with Gaussian mass matrix HMC)
# (ignored if using HMC with fixed mass matrix HMC)
# (ignored if using HMC with fixed step size HMC)
# (ignored if using HMC with Riemann manifold HMC with fixed step size)
# (ignored if using HMC with Gaussian mass matrix HMC with fixed step size)
# (ignored if using HMC with fixed mass matrix HMC with fixed step size)
```

```
# 8-dimensional parameters
name = "p8"
# the name of this experiment
# the name of the output file to write to (without the .mat or .h5 suffix)
MODEL_DIM = 8
# the dimensionality of the model (number of parameters)
# (including hyperparameters, if sampling hyperparameters)
# (including noise parameter, if using a noise parameter)
# (not including the latent function, if using a latent function)
```

```
# MCMC SAMPLER SETTINGS:
# (ignored if using exact inference)
# the MCMC method to use (one of "slice", "hmc", "elliptical")
# (ignored if using exact inference)
MCMC_METHOD = "slice"
# the number of MCMC samples to collect
# (ignored if using exact inference)
NUM_SAMPLES = 1000
# the number of MCMC samples to discard as burn-in
# (ignored if using exact inference)
# (ignored if using a fixed number of MCMC samples)
# (ignored if using slice sampling or elliptical slice sampling)
# (ignored if using HMC with dual-averaging step size adaptation)
# (ignored if using HMC with Riemann manifold adaptation)
# (ignored if using HMC with Gaussian mass matrix adaptation)
# (ignored if using HMC with fixed mass matrix)
# (ignored if using HMC with fixed step size)
# (ignored if using HMC with Riemann manifold HMC)
# (ignored if using HMC with Gaussian mass matrix HMC)
# (ignored if using HMC with fixed mass matrix HMC)
# (ignored if using HMC with fixed step size HMC)
# (ignored if using HMC with Riemann manifold HMC with fixed step size)
# (ignored if using HMC with Gaussian mass matrix HMC with fixed step size)
# (ignored if using HMC with fixed mass matrix HMC with fixed step size)
BURN_IN = 0
# the number of initial MCMC samples to discard
# (ignored if using exact inference)
# (ignored if using a fixed number of MCMC samples)
# (ignored if using slice sampling or elliptical slice sampling)
# (ignored if using HMC with dual-averaging step size adaptation)
# (ignored if using HMC with Riemann manifold adaptation)
# (ignored if using HMC with Gaussian mass matrix adaptation)
# (ignored if using HMC with fixed mass matrix)
# (ignored if using HMC with fixed step size)
# (ignored if using HMC with Riemann manifold HMC)
# (ignored if using HMC with Gaussian mass matrix HMC)
# (ignored if using HMC with fixed mass matrix HMC)
# (ignored if using HMC with fixed step size HMC)
# (ignored if using HMC with Riemann manifold HMC with fixed step size)
# (ignored if using HMC with Gaussian mass matrix HMC with fixed step size)
# (ignored if using HMC with fixed mass matrix HMC with fixed step size)
THIN = 1
# the number of MCMC samples to keep after burn-in
# (ignored if using exact inference)
# (ignored if using a fixed number of MCMC samples)
# (ignored if using slice sampling or elliptical slice sampling)
# (ignored if using HMC with dual-averaging step size adaptation)
# (ignored if using HMC with Riemann manifold adaptation)
# (ignored if using HMC with Gaussian mass matrix adaptation)
# (ignored if using HMC with fixed mass matrix)
# (ignored if using HMC with fixed step size)
# (ignored if using HMC with Riemann manifold HMC)
# (ignored if using HMC with Gaussian mass matrix HMC)
# (ignored if using HMC with fixed mass matrix HMC)
# (ignored if using HMC with fixed step size HMC)
# (ignored if using HMC with Riemann manifold HMC with fixed step size)
# (ignored if using HMC with Gaussian mass matrix HMC with fixed step size)
# (ignored if using HMC with fixed mass matrix HMC with fixed step size)
```

```
# 9-dimensional parameters
name = "p9"
# the name of this experiment
# the name of the output file to write to (without the .mat or .h5 suffix)
MODEL_DIM = 9
# the dimensionality of the model (number of parameters)
# (including hyperparameters, if sampling hyperparameters)
# (including noise parameter, if using a noise parameter)
# (not including the latent function, if using a latent function)
```

```
# MCMC SAMPLER SETTINGS:
# (ignored if using exact inference)
# the MCMC method to use (one of "slice", "hmc", "elliptical")
# (ignored if using exact inference)
MCMC_METHOD = "slice"
# the number of MCMC samples to collect
# (ignored if using exact inference)
NUM_SAMPLES = 1000
# the number of MCMC samples to discard as burn-in
# (ignored if using exact inference)
# (ignored if using a fixed number of MCMC samples)
# (ignored if using slice sampling or elliptical slice sampling)
# (ignored if using HMC with dual-averaging step size adaptation)
# (ignored if using HMC with Riemann manifold adaptation)
# (ignored if using HMC with Gaussian mass matrix adaptation)
# (ignored if using HMC with fixed mass matrix)
# (ignored if using HMC with fixed step size)
# (ignored if using HMC with Riemann manifold HMC)
# (ignored if using HMC with Gaussian mass matrix HMC)
# (ignored if using HMC with fixed mass matrix HMC)
# (ignored if using HMC with fixed step size HMC)
# (ignored if using HMC with Riemann manifold HMC with fixed step size)
# (ignored if using HMC with Gaussian mass matrix HMC with fixed step size)
# (ignored if using HMC with fixed mass matrix HMC with fixed step size)
BURN_IN = 0
# the number of initial MCMC samples to discard
# (ignored if using exact inference)
# (ignored if using a fixed number of MCMC samples)
# (ignored if using slice sampling or elliptical slice sampling)
# (ignored if using HMC with dual-averaging step size adaptation)
# (ignored if using HMC with Riemann manifold adaptation)
# (ignored if using HMC with Gaussian mass matrix adaptation)
# (ignored if using HMC with fixed mass matrix)
# (ignored if using HMC with fixed step size)
# (ignored if using HMC with Riemann manifold HMC)
# (ignored if using HMC with Gaussian mass matrix HMC)
# (ignored if using HMC with fixed mass matrix HMC)
# (ignored if using HMC with fixed step size HMC)
# (ignored if using HMC with Riemann manifold HMC with fixed step size)
# (ignored if using HMC with Gaussian mass matrix HMC with fixed step size)
# (ignored if using HMC with fixed mass matrix HMC with fixed step size)
THIN = 1
# the number of MCMC samples to keep after burn-in
# (ignored if using exact inference)
# (ignored if using a fixed number of MCMC samples)
# (ignored if using slice sampling or elliptical slice sampling)
# (ignored if using HMC with dual-averaging step size adaptation)
# (ignored if using HMC with Riemann manifold adaptation)
# (ignored if using HMC with Gaussian mass matrix adaptation)
# (ignored if using HMC with fixed mass matrix)
# (ignored if using HMC with fixed step size)
# (ignored if using HMC with Riemann manifold HMC)
# (ignored if using HMC with Gaussian mass matrix HMC)
# (ignored if using HMC with fixed mass matrix HMC)
# (ignored if using HMC with fixed step size HMC)
# (ignored if using HMC with Riemann manifold HMC with fixed step size)
# (ignored if using HMC with Gaussian mass matrix HMC with fixed step size)
# (ignored if using HMC with fixed mass matrix HMC with fixed step size)
```

```
# 10-dimensional parameters
name = "p10"
# the name of this experiment
# the name of the output file to write to (without the .mat or .h5 suffix)
MODEL_DIM = 10
# the dimensionality of the model (number of parameters)
# (including hyperparameters, if sampling hyperparameters)
# (including noise parameter, if using a noise parameter)
# (not including the latent function, if using a latent function)
```

```
# MCMC SAMPLER SETTINGS:
# (ignored if using exact inference)
# the MCMC method to use (one of "slice", "hmc", "elliptical")
# (ignored if using exact inference)
MCMC_METHOD = "slice"
# the number of MCMC samples to collect
# (ignored if using exact inference)
NUM_SAMPLES = 1000
# the number of MCMC samples to discard as burn-in
# (ignored if using exact inference)
# (ignored if using a fixed number of MCMC samples)
# (ignored if using slice sampling or elliptical slice sampling)
# (ignored if using HMC with dual-averaging step size adaptation)
# (ignored if using HMC with Riemann manifold adaptation)
# (ignored if using HMC with Gaussian mass matrix adaptation)
# (ignored if using HMC with fixed mass matrix)
# (ignored if using HMC with fixed step size)
# (ignored if using HMC with Riemann manifold HMC)
# (ignored if using HMC with Gaussian mass matrix HMC)
# (ignored if using HMC with fixed mass matrix HMC)
# (ignored if using HMC with fixed step size HMC)
# (ignored if using HMC with Riemann manifold HMC with fixed step size)
# (ignored if using HMC with Gaussian mass matrix HMC with fixed step size)
# (ignored if using HMC with fixed mass matrix HMC with fixed step size)
BURN_IN = 0
# the number of initial MCMC samples to discard
# (ignored if using exact inference)
# (ignored if using a fixed number of MCMC samples)
# (ignored if using slice sampling or elliptical slice sampling)
# (ignored if using HMC with dual-averaging step size adaptation)
# (ignored if using HMC with Riemann manifold adaptation)
# (ignored if using HMC with Gaussian mass matrix adaptation)
# (ignored if using HMC with fixed mass matrix)
# (ignored if using HMC with fixed step size)
# (ignored if using HMC with Riemann manifold HMC)
# (ignored if using HMC with Gaussian mass matrix HMC)
# (ignored if using HMC with fixed mass matrix HMC)
# (ignored if using HMC with fixed step size HMC)
# (ignored if using HMC with Riemann manifold HMC with fixed step size)
# (ignored if using HMC with Gaussian mass matrix HMC with fixed step size)
# (ignored if using HMC with fixed mass matrix HMC with fixed step size)
THIN = 1
# the number of MCMC samples to keep after burn-in
# (ignored if using exact inference)
# (ignored if using a fixed number of MCMC samples)
# (ignored if using slice sampling or elliptical slice sampling)
# (ignored if using HMC with dual-averaging step size adaptation)
# (ignored if using HMC with Riemann manifold adaptation)
# (ignored if using HMC with Gaussian mass matrix adaptation)
# (ignored if using HMC with fixed mass matrix)
# (ignored if using HMC with fixed step size)
# (ignored if using HMC with Riemann manifold HMC)
# (ignored if using HMC with Gaussian mass matrix HMC)
# (ignored if using HMC with fixed mass matrix HMC)
# (ignored if using HMC with fixed step size HMC)
# (ignored if using HMC with Riemann manifold HMC with fixed step size)
# (ignored if using HMC with Gaussian mass matrix HMC with fixed step size)
# (ignored if using HMC with fixed mass matrix HMC with fixed step size)
```

Analysis of the Results:

After each iteration, the Distance Vs. Velocity plot for the rocket landing is displayed along with the matching loss function value for both the X and Y directions.

The program is executed for 60 iterations and the neural network achieves the expected result in the 57th iteration. The rocket is optimized to the (0,0) coordinate and the velocity of the rocket is optimized to 0,0 in both x and y direction.

In the first iteration the loss is 76851.016 and the loss gets reduced over every iteration.

For the first 10 iterations the loss gets reduced drastically and then it goes down gradually minimizing to 0 on the 57th iteration.

By 40th iteration the loss is 0.05, the rocket is almost at the correct position but has some velocity. It takes about 17 iterations to bring it to complete 0.

At 51st iteration and onwards, we have achieved loss = 0 and thus, our objective function has been properly minimized and the target landing conditions are satisfied. Thus, we can say that with the currently formulated neural network and dynamic model, we are able to achieve convergence and minimize our objective function in around 50 iterations each with a time step of 0.1 seconds. The "V vs distance" plots for both X and Y shows the change in velocity with distance when looked from the right to the left.