NTNU
Kunnskap for en bedre verden

IT3105 - Project 1

A general purpose JAX-based controller

Author :
Danial Bashir

February, 2024

Table of Contents

List of Tables

1 Description of parameters

Table 1 is a description of the system parameters.

Parameter	Description
consys_epochs	Number of training epochs
consys_timesteps	Number of time steps in the system
consys_learning_rate	Learning rate for the system
consys_disturbance_range	Range of disturbance in the system
controller_target	Target value for the controller
pid_initial_params	Initial parameters for PID controller: [k_p, k_d, k_i]
nn_initial_range_weights	Range of initial weights for NeuralNetController
nn_initial_range_biases	Range of initial biases for NeuralNetController
nn_controller_layers	Number of nodes in each layer of NeuralNetController
nn_controller_activation_functions	Activation functions for each non-input layer in NeuralNet-Controller
bathtub_init_height	Initial height of water in bathtub
bathtub_cross_sectional_area	Cross-sectional area of the bathtub
bathtub_drain_cross_sectional_area	Cross-sectional area of the bathtub drain
bathtub_gravity	Gravity constant
cournot_init_q1	Initial quantity for firm 1 in Cournot competition
cournot_init_q2	Initial quantity for firm 2 in Cournot competition
cournot_init_profit	Initial profit for firm 1 in Cournot competition
cournot_max_price	Maximum price in Cournot competition
cournot_marginal_cost	Marginal cost in Cournot competition
drone_init_velocity	Initial velocity of the drone
drone_mass	Mass of the drone
drone_drag_coefficient	Drag coefficient for the drone

Table 1: System Parameters with Descriptions

2 Plant 1 - Bathtub

Summary of the run - PID controller. The parameters used are shown in Table 2, and the learning progression is shown in Figure 1. The PID initial parameters and the learning rate were chosen by trial and error. The MSE quickly drops and stays fluctuating between 0.3 and 0.4. The water height after 50 timesteps is between 18.0 and 20.0 for every epoch.

Parameter	Value
consys_epochs	100
consys_timesteps	50
consys_learning_rate	0.04
consys_disturbance_range	[-0.01, 0.01]
controller_target	20.0
pid_initial_params	[0.1, 0.3, 0.5]
bathtub_init_height	20.0
bathtub_cross_sectional_area	10.0
bathtub_drain_cross_sectional_area	0.1
bathtub_gravity	9.8

Table 2: System Parameters for Bathtub with Classic PID Controller

Summary of the run - Controller: Neural net controller. The learning progression is shown in Figure 3. The PID initial parameters and the learning rate were chosen by trial and error and the plant parameters are the same as the previous run. The MSE quickly drops and stays around 10^{-6}. The water height for Epoch > 5 is stable between 19.99 and 20.01.

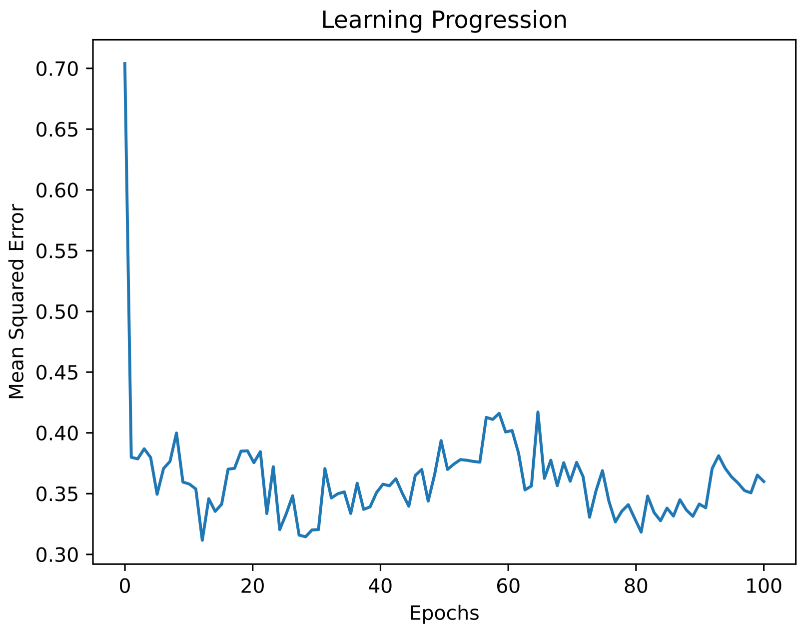

Figure 1: Learning progression, PID controller, Bathtub

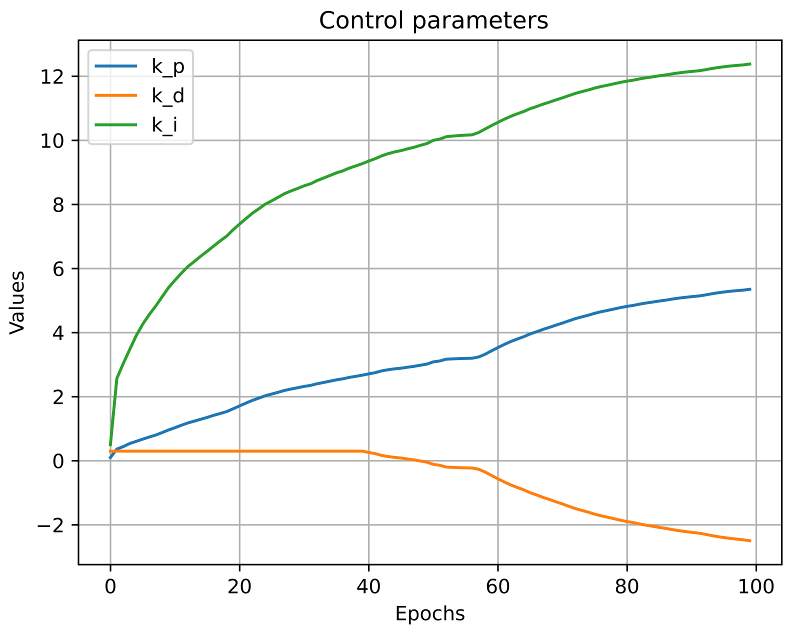

Figure 2: Control parameters, PID controller, Bathtub

Parameter	Value
consys_epochs	100
consys_timesteps	50
consys_learning_rate	0.04
consys_disturbance_range	[-0.01, 0.01]
controller_target	20.0
nn_initial_range_weights	[-0.1, 0.1]
nn_initial_range_biases	[-0.1, 0.1]
nn_controller_layers	[3, 10, 10, 10, 1]
nn_controller_activation_functions	["sigmoid", "sigmoid", "sigmoid", "linear"]
bathtub_init_height	20.0
bathtub_cross_sectional_area	10.0
bathtub_drain_cross_sectional_area	0.1
bathtub_gravity	9.8

Table 3: System Parameters for Bathtub with Neural Net Controller

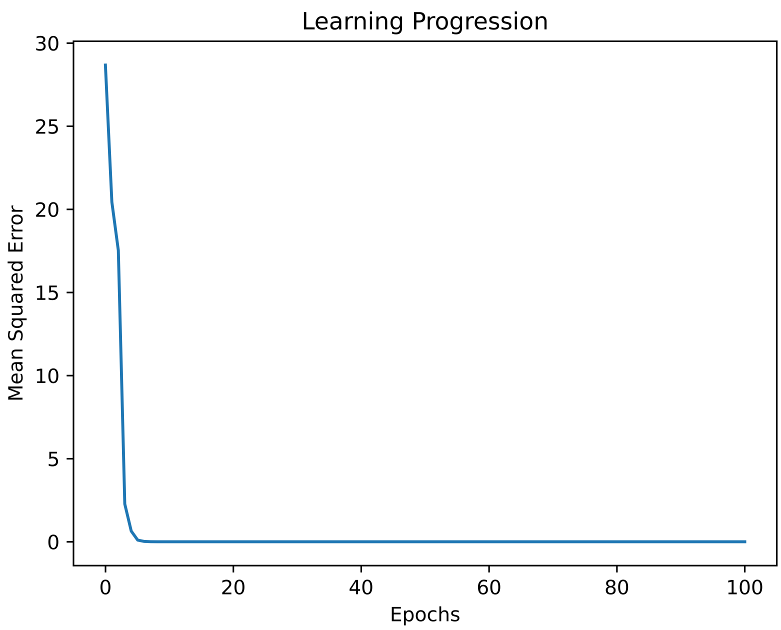

Figure 3: Learning progression, Neural Net controller, Bathtub

3 Plant 2 - Cournot Competition

Summary of the run - PID controller. The learning progression is shown in Figure 4 and the parameters are shown in Table 4. The PID initial parameters and the learning rate were chosen by trial and error. Note the initialization of the q_0, q_1, and initial profit parameters. The MSE quickly drops and stays fluctuating and descending, reaching a minimum of around 2.5 as shown in Figure 4. The output profit after 50 timesteps is between 49.1 and 50.9 for all epochs.

Parameter	**Value**
consys_epochs	100
consys_timesteps	50
consys_learning_rate	0.000002
consys_disturbance_range	[-0.01, 0.01]
controller_target	50.0
pid_initial_params	[-0.01, 0.01, 0.01]
cournot_init_q1	0.5
cournot_init_q2	0.5
cournot_init_profit	50.0
cournot_max_price	100.0
cournot_marginal_cost	1.0

Table 4: System Parameters for Cournot Competition with PID controller

Summary of the run - Controller: Neural net controller. The learning progression is shown in Figure 6 and the parameters are shown in Table 5. The learning rate was chosen by trial and error. Note the initialization of the q_0, q_1, and initial profit parameters. The MSE quickly drops and stays stable for Epoch > 5 with values around 0.001. The output profit after 50 timesteps is stable between 50.002 and 50.007 for Epoch > 11.

Parameter	**Value**
consys_epochs	100
consys_timesteps	50
consys_learning_rate	0.001
consys_disturbance_range	[-0.01, 0.01]
controller_target	50.0
nn_initial_range_weights	[-0.1, 0.1]
nn_initial_range_biases	[-0.1, 0.1]
nn_controller_layers	[3, 15, 15, 15, 1]
nn_controller_activation_functions	["tanh", "tanh", "tanh", "linear"]
cournot_init_q1	0.5
cournot_init_q2	0.5
cournot_init_profit	50.0
cournot_max_price	100.0
cournot_marginal_cost	1.0

Table 5: System Parameters for Cournot Competition with Neural Net Controller

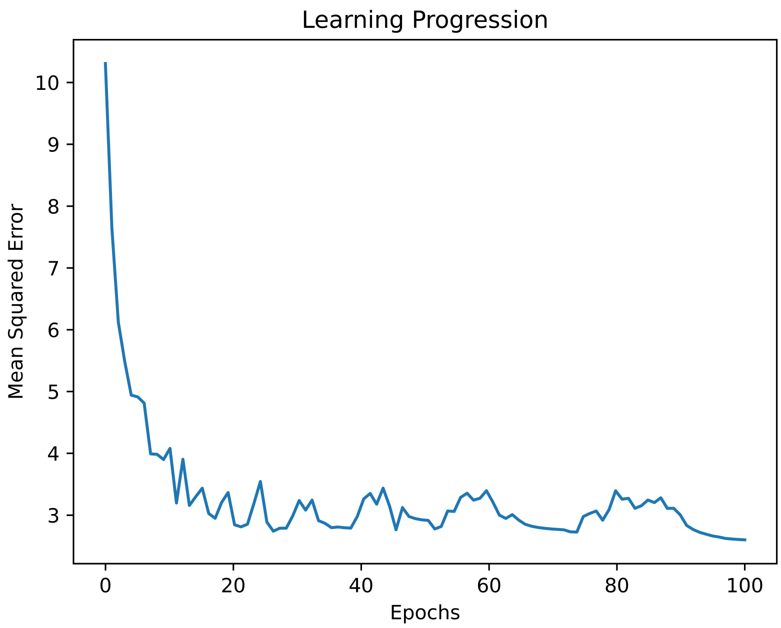

Figure 4: Learning progression, PID controller, Cournot Competition

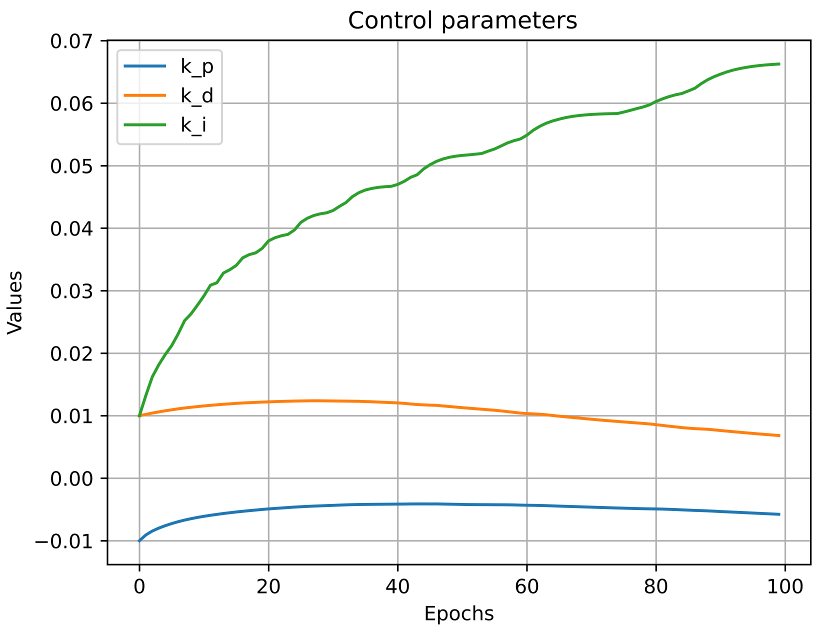

Figure 5: Control parameters, PID controller, Cournot Competition

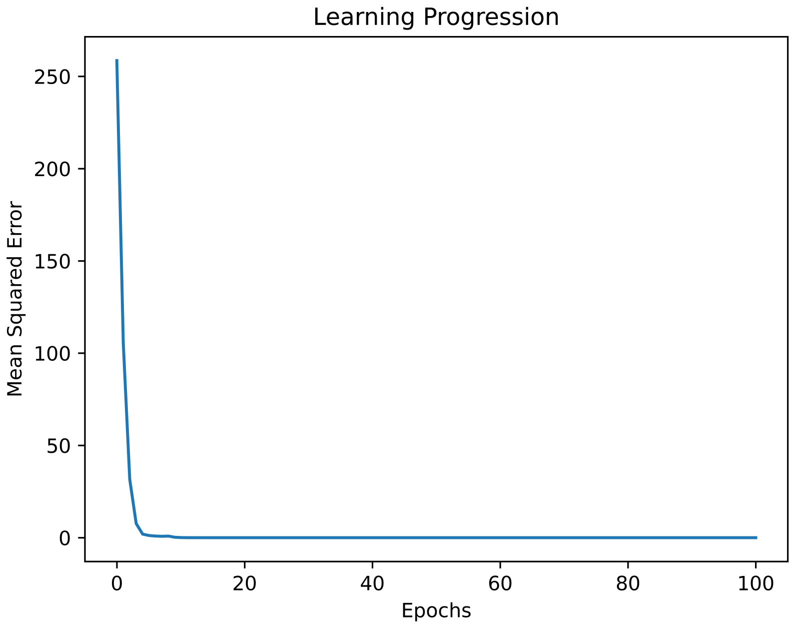

Figure 6: Learning progression, Neural Net controller, Cournot Competition

4 Plant 3 - Drone

Consider a drone control system where the goal is to maintain a constant velocity. The drone is affected by the drag force and by the output of its engines (U).

The relationship of the forces acting on the drone and its acceleration can be expressed as:

$$U - \text{Drag} = m \cdot a \tag{1}$$

where $\text{Drag} = c \cdot v(t)^2 \cdot D$.

The dynamics of the drone's velocity ($v(t)$) can be described by the following partial differential equation:

$$U - c \cdot v(t)^2 \cdot D = m \cdot \frac{\partial v(t)}{\partial t} \tag{2}$$

where:

$$\begin{aligned} m &\text{ is the mass of the drone,} \\ v(t) &\text{ is the velocity of the drone at time } t, \\ D &\text{ is a disturbance to the system,} \\ U &\text{ is the force output from the drone's engine,} \\ c &\text{ is the drag coefficient.} \end{aligned}$$

This partial differential equation captures the effect of the force from the drone's engine (U), the drag force ($c \cdot v(t)^2 \cdot D$), and the mass (m) on the change in velocity over time.

The drone receives two inputs on every time step:

- U, the force output from the drone's engine.
- D, a random disturbance that varies with each time step.

The control objective is to maintain a constant velocity despite the presence of disturbances and the effects of drag.

Summary of the run - PID controller. The learning progression is shown in Figure 7 and the parameters are shown in Table 6. The PID initial parameters and the learning rate were chosen by trial and error. The MSE quickly drops and stays fluctuating and descending, reaching a minimum of around 1.25 as shown in Figure 7. The output velocity after 50 timesteps is between 8.2 and 10.9 for all epochs. The results are significantly better when the range of valid disturbance values is smaller.

Parameter	**Value**
consys_epochs	100
consys_timesteps	50
consys_learning_rate	0.1
consys_disturbance_range	[0.9, 1.1]
controller_target	10.0
pid_initial_params	[0.0, 0.0, 2.0]
drone_init_velocity	10.0
drone_mass	20.0
drone_drag_coefficient	1.5

Table 6: System Parameters for Drone with PID Controller

Summary of the run - Controller: Neural net controller. The learning progression is shown in Figure 9 and the parameters are shown in Table 7. The learning rate was chosen by trial and

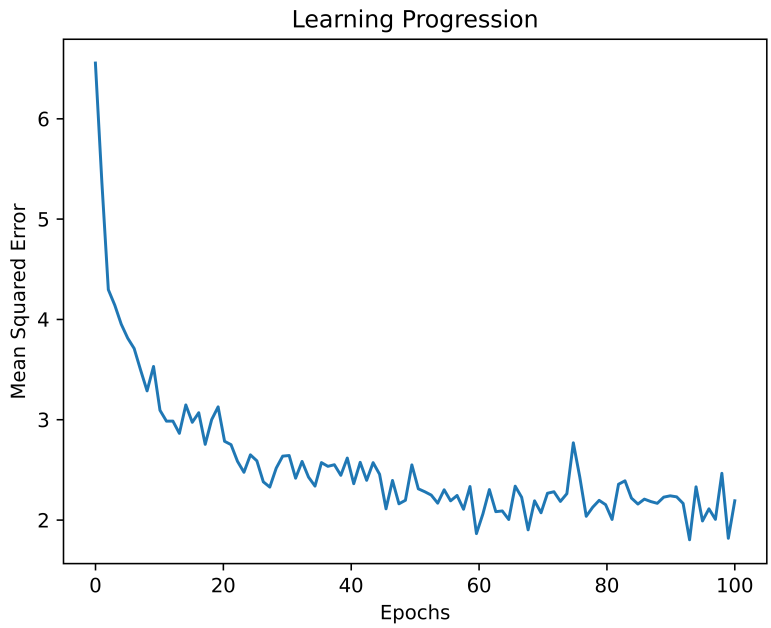

Figure 7: Learning progression, PID controller, Drone

error. The MSE quickly drops and stays stable for Epoch > 5 with values around 2.0, but with some outliers because of the big disturbance range. The output velocity after 50 timesteps is stable between 9.0 and 11.0 for Epoch > 6.

Parameter	**Value**
consys_epochs	100
consys_timesteps	50
consys_learning_rate	0.01
consys_disturbance_range	[0.9, 1.1]
controller_target	10.0
nn_initial_range_weights	[-0.1, 0.1]
nn_initial_range_biases	[-0.1, 0.1]
nn_controller_layers	[3, 15, 15, 15, 1]
nn_controller_activation_functions	["relu", "relu", "relu", "linear"]
drone_init_velocity	10.0
drone_mass	20.0
drone_drag_coefficient	1.5

Table 7: System Parameters for Drone with Neural Net Controller

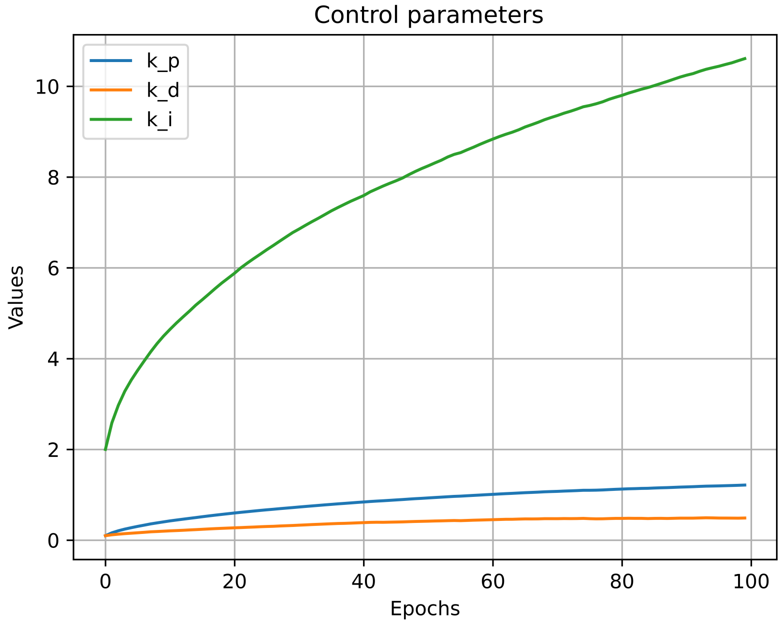

Figure 8: Control parameters, PID controller, Drone

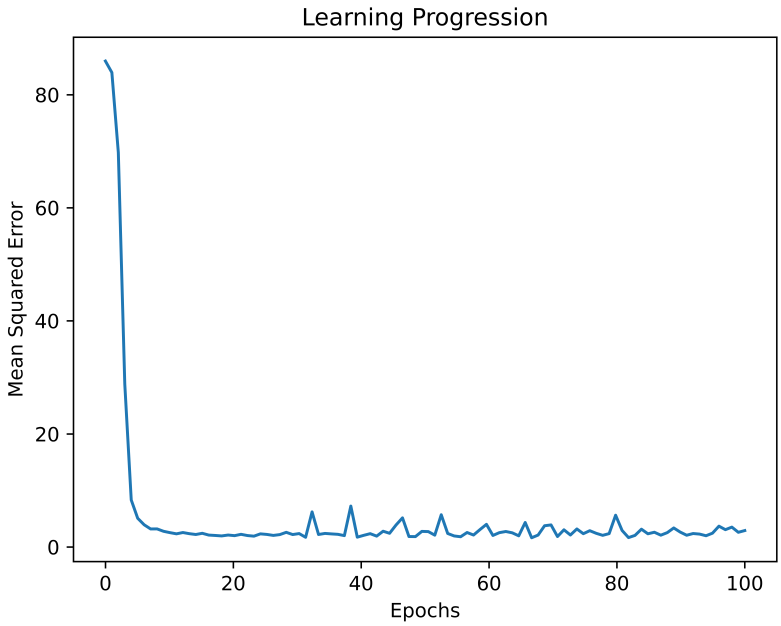

Figure 9: Learning progression, Neural Net controller, Drone