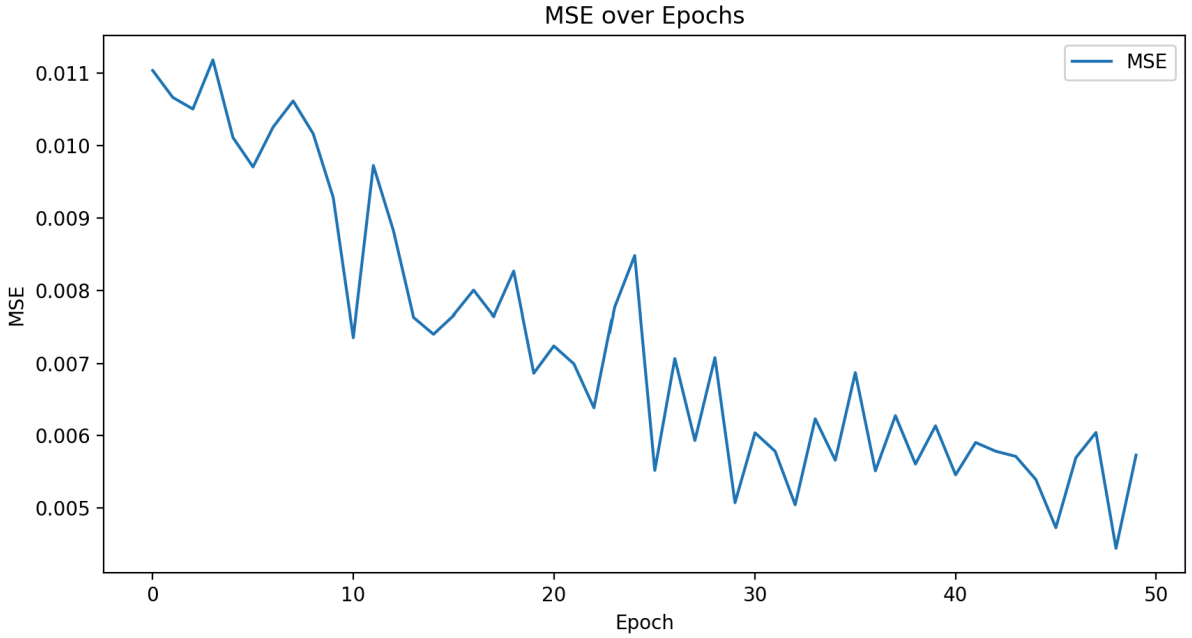
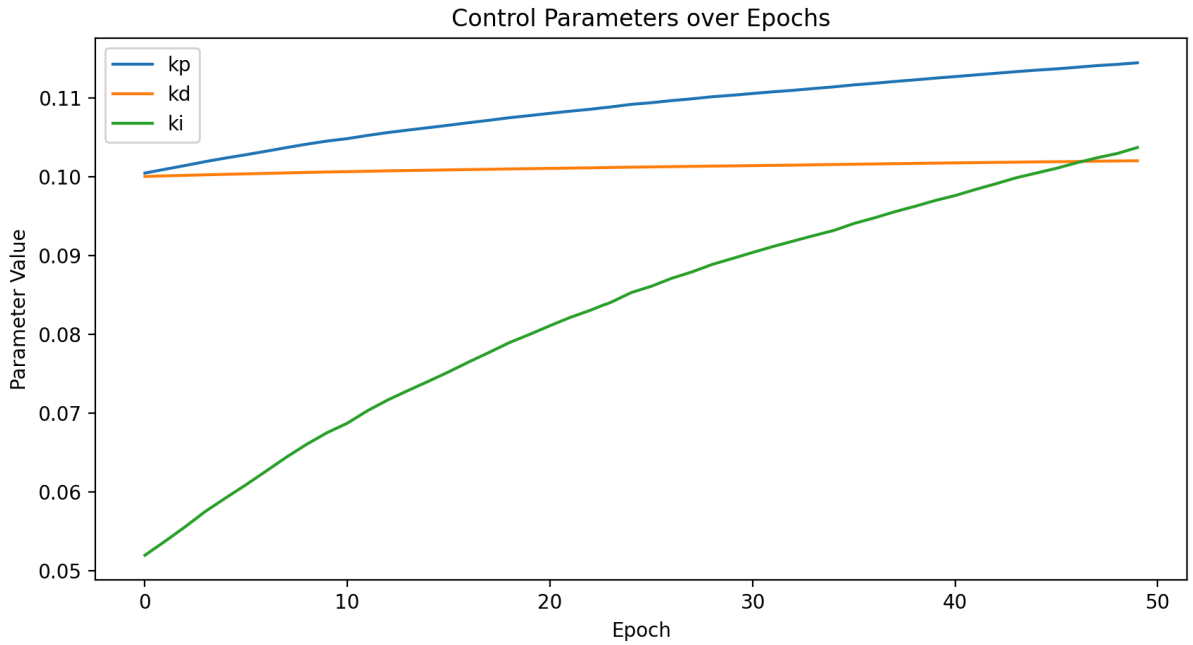
Jax-Controller Report

Run 1

Object	Parameter	Value
Consys	Controller	ControllerJax
Consys	Plant	plantBathtub
Consys	Epochs	50
Consys	Timesteps	10
Consys	learning_rate	0.01
Consys	D_range	[-0.01, 0.01]
ControllerJax	kp	0.1
ControllerJax	kd	0.10
ControllerJax	ki	0.05
plantBathtub	cross_sectional_area	1
plantBathtub	drain_area	0.01
plantBathtub	initial_height	1

Kommentar: Denne var den første, ikke så vakker mse, kan ha med høy D_range å gjøre. Eller litt få epoker.

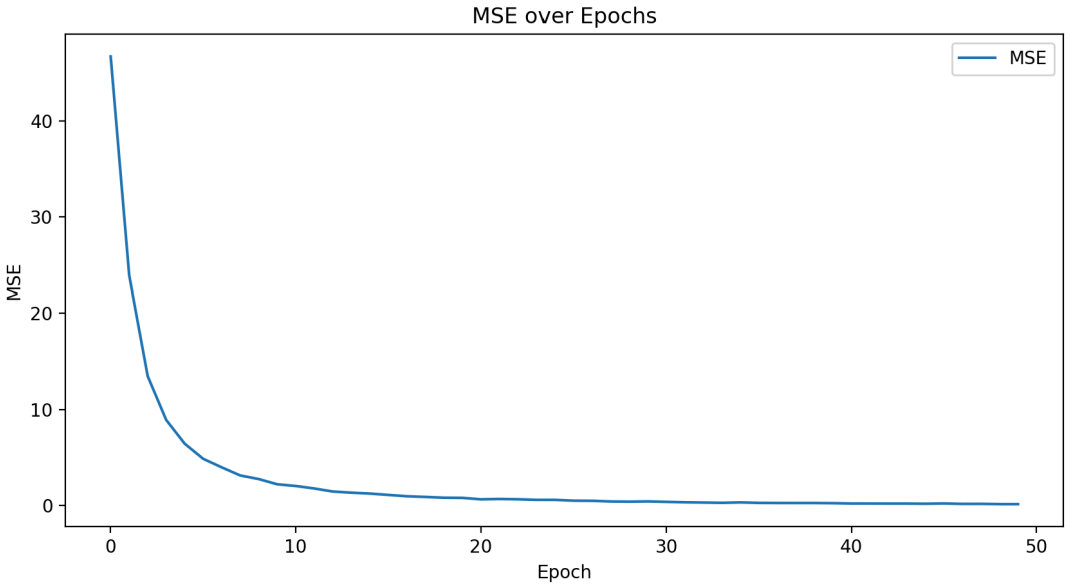


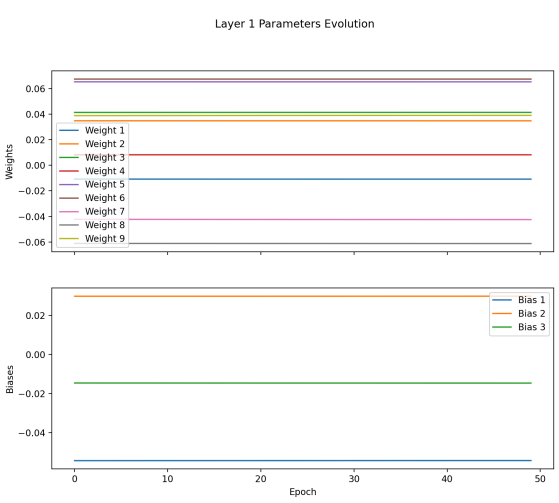


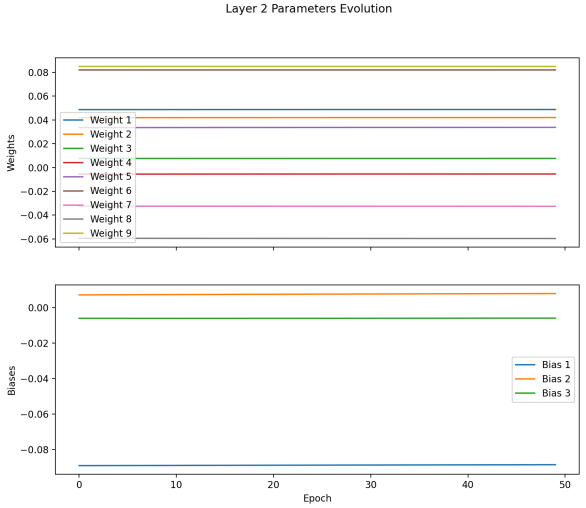
Run 2

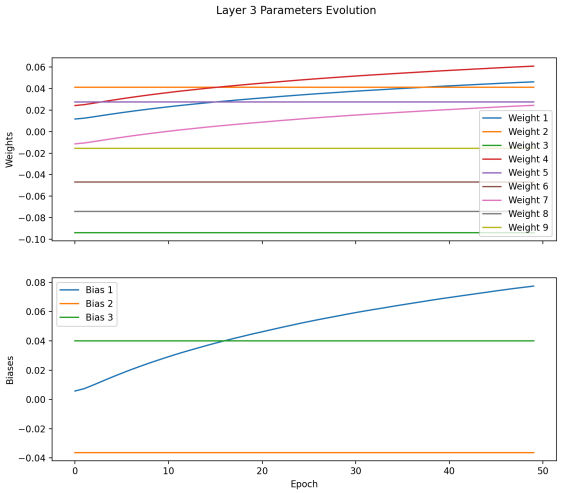
Object	Parameter	Value
Consys	Controller	ControllerNN
Consys	Plant	plantBathtub
Consys	Epochs	50
Consys	Timesteps	30
Consys	learning_rate	0.01
Consys	D_range	[-0.01, 0.01]
ControllerNN	Hidden Layers	[3, 3, 3]
ControllerNN	Activation Functions	[tanh, sigmoid, relu]
ControllerNN	Range Initialization	[-0.1, 0.1]
plantBathtub	cross_sectional_area	1
plantBathtub	drain_area	0.01
plantBathtub	initial_height	1

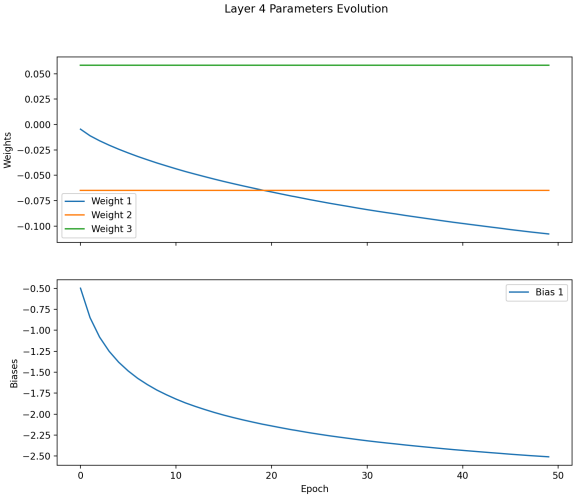
Kommentar: Her brukte jeg 3 skjulte lag, men det var bare de siste som oppdaterte seg. Jeg mistenker at det er fordi oppgaven ikke er mer komplisert enn så og krever ikke mer tilpasning av nettverket.


MSE over Epochs
MSE
Epoch
MSE


Layer 1 Parameters Evolution
Weights
Biases
Epoch
Weight 1
Weight 2
Weight 3
Weight 4
Weight 5
Weight 6
Weight 7
Weight 8
Weight 9
Bias 1
Bias 2
Bias 3


Layer 2 Parameters Evolution
Weights
Biases
Epoch
Weight 1
Weight 2
Weight 3
Weight 4
Weight 5
Weight 6
Weight 7
Weight 8
Weight 9
Bias 1
Bias 2
Bias 3

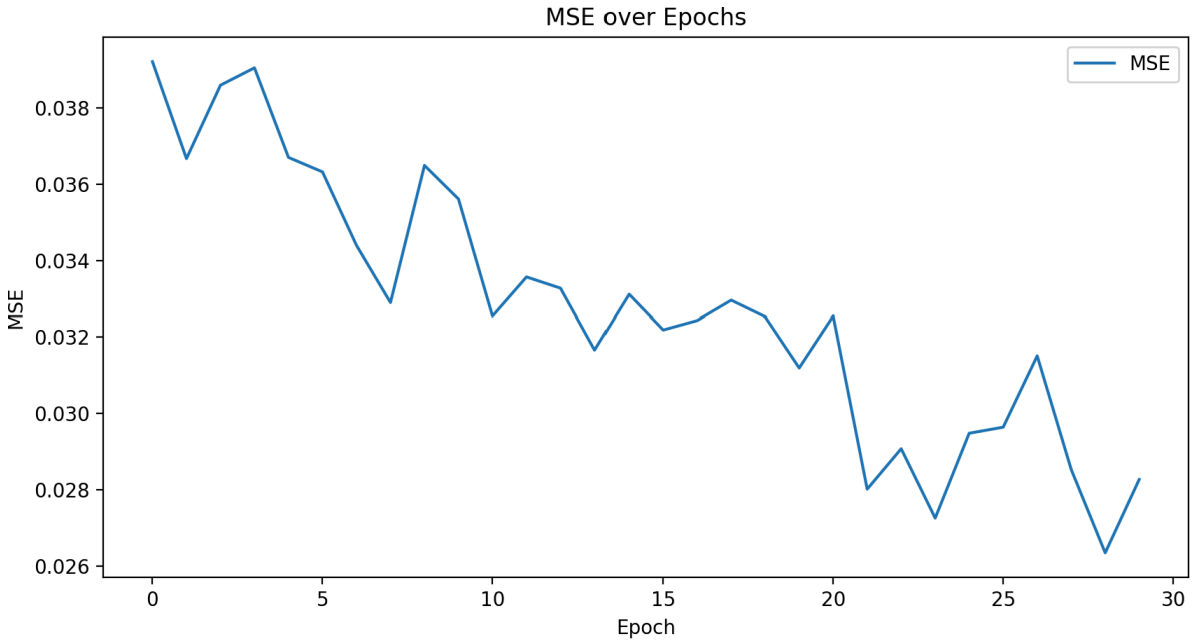

Layer 3 Parameters Evolution
Weights
Biases
Epoch
Weight 1
Weight 2
Weight 3
Weight 4
Weight 5
Weight 6
Weight 7
Weight 8
Weight 9
Bias 1
Bias 2
Bias 3

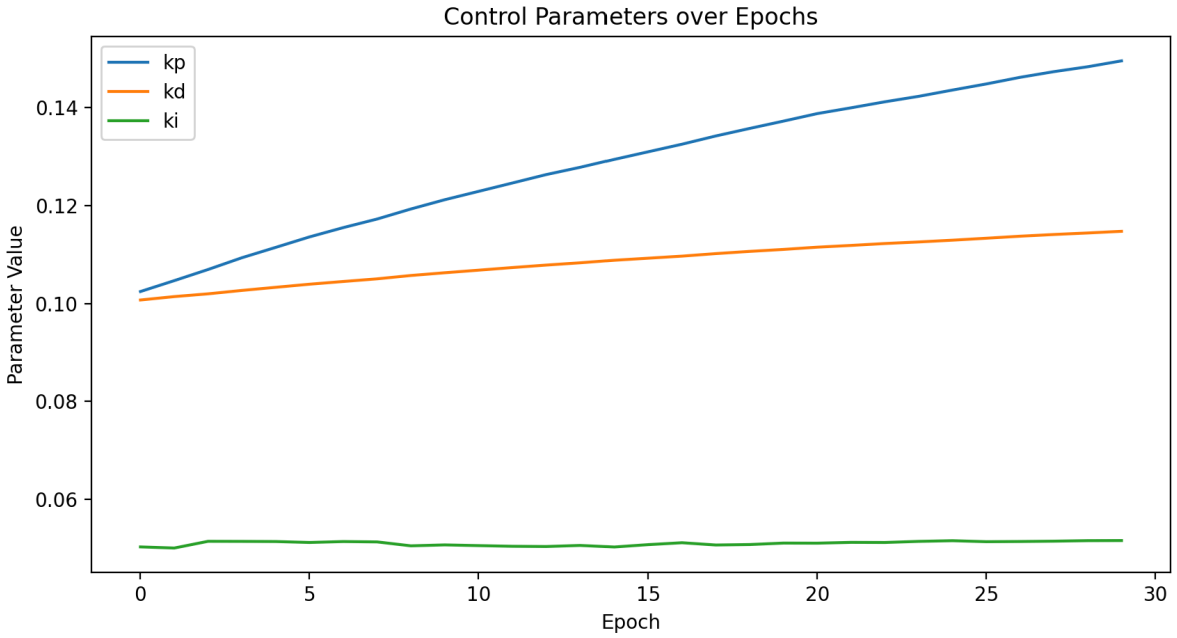

Layer 4 Parameters Evolution
Weights
Biases
Epoch
Weight 1
Weight 2
Weight 3
Bias 1

Run 3

Object	Parameter	Value
Consys	Controller	ControllerJax
Consys	Plant	plantCournot
Consys	Epochs	30
Consys	Timesteps	20
Consys	learning_rate	0.01
Consys	D_range	[-0.02, 0.02]
ControllerJax	kp	0.1
ControllerJax	kd	0.10
ControllerJax	ki	0.05
plantCournot	q1	0.5
plantCournot	q2	0.5
plantCournot	pMax	4
plantCournot	goal_state	2
plantCournot	marginalCost	0.2

Kommentar: Det vanskelige med Cournot var å finne en terget og “learning rate” som ga et meningsfullt resultat. Denne kombinasjonen ga mening.

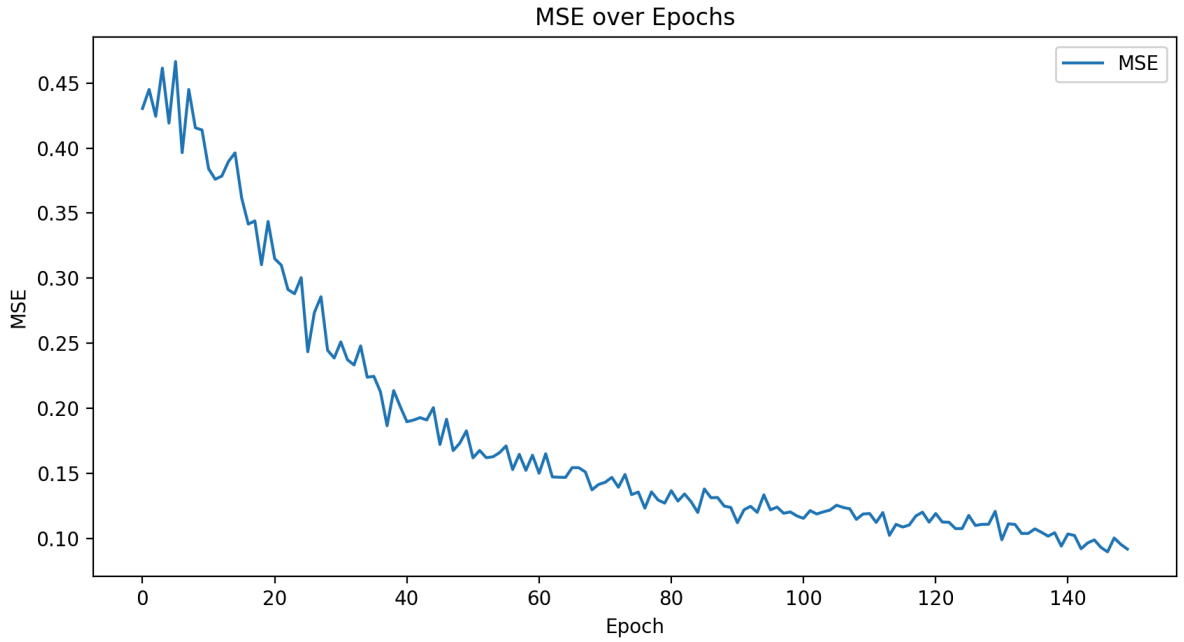


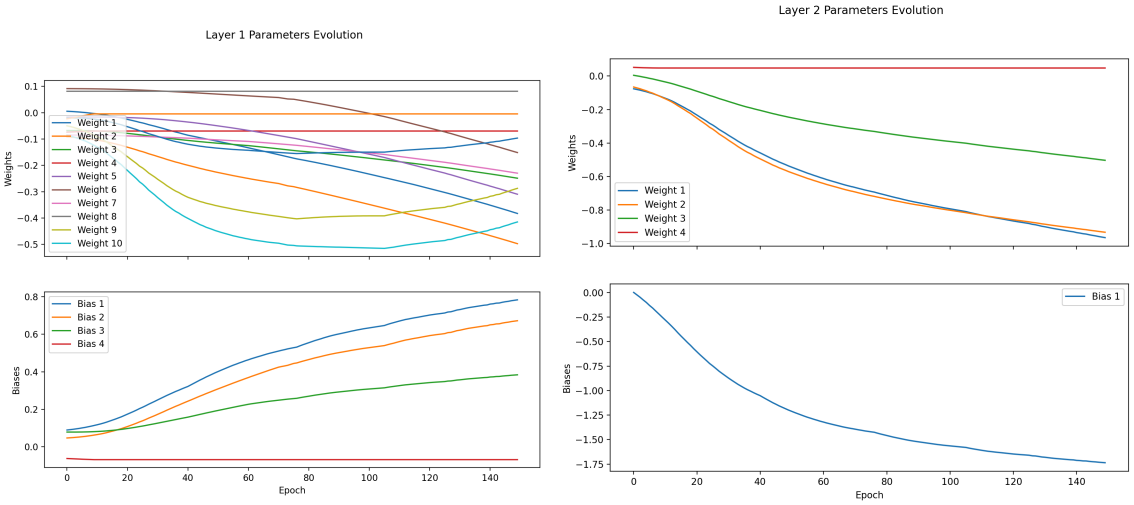


Run 4

Object	Parameter	Value
Consys	Controller	ControllerNN
Consys	Plant	plantCournot
Consys	Epochs	150
Consys	Timesteps	10
Consys	learning_rate	0.2
Consys	D_range	[-0.01, 0.01]
ControllerNN	Hidden Layers	[4]
ControllerNN	Activation Functions	[relu]
ControllerNN	Range Initialization	[-0.1, 0.1]
plantCournot	q1	0.5
plantCournot	q2	0.5
plantCournot	pMax	4
plantCournot	goal_state	2
plantCournot	marginalCost	0.2

Kommentar: Her slet jeg også med å finne en læringsrate fordi kontrolleren automatisk ga en høy output. Men gitt nok antall epoker klarte nettverket til slut å finne ut av det.



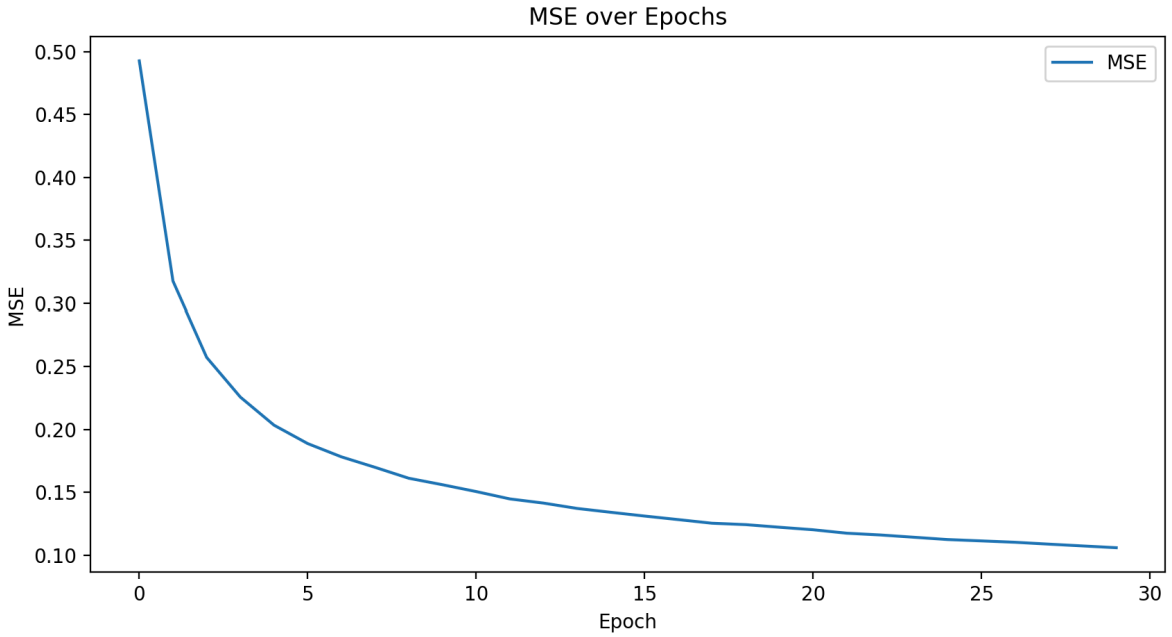


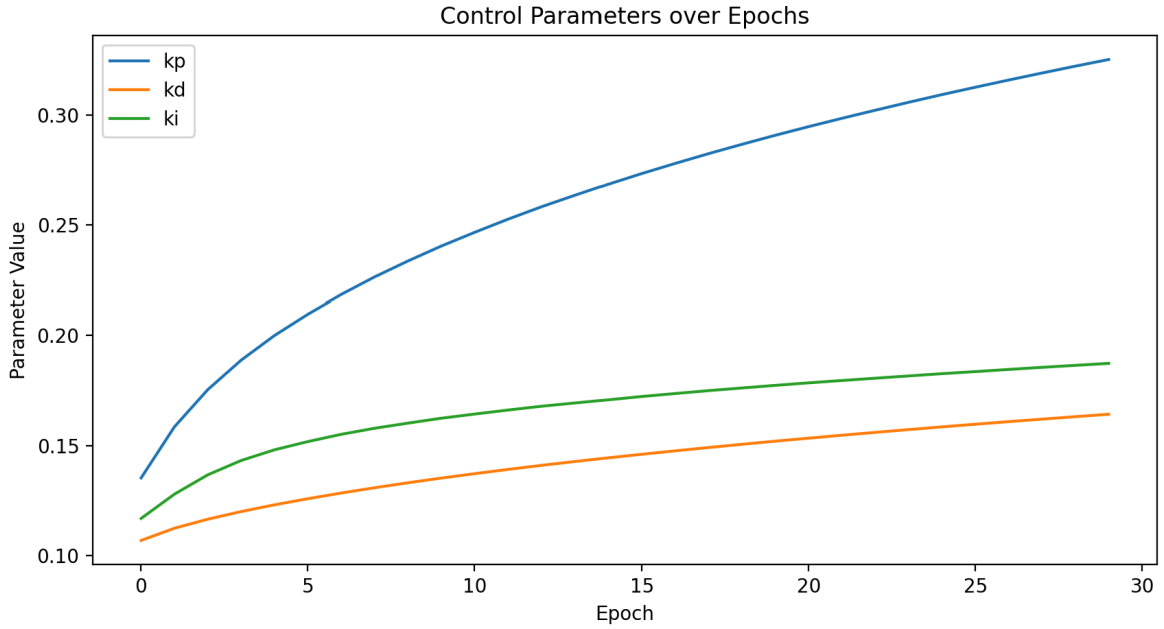
Run 5

Object	Parameter	Value
Consys	Controller	ControllerJax
Consys	Plant	plantLotkaVolterra
Consys	Epochs	30
Consys	Timesteps	20
Consys	learning_rate	0.01
Consys	D_range	[-0.02, 0.02]
ControllerJax	kp	0.1
ControllerJax	kd	0.10
ControllerJax	ki	0.05
plantLotkaVolterra	rabbits	1
plantLotkaVolterra	foxes	1

plantLotkaVolterra	goal_state	2
plantLotkaVolterra	rabbitBirthRate	0.1
plantLotkaVolterra	rabbitDeathRatePerFox	0.02
plantLotkaVolterra	foxDeathRate	0.1
plantLotkaVolterra	foxBirthRatePerRabbit	0.01

Kommentar: Denne fungerte direkte etter å ha skrevet koden, uten noen tilpassing. Det var veldig gøy!

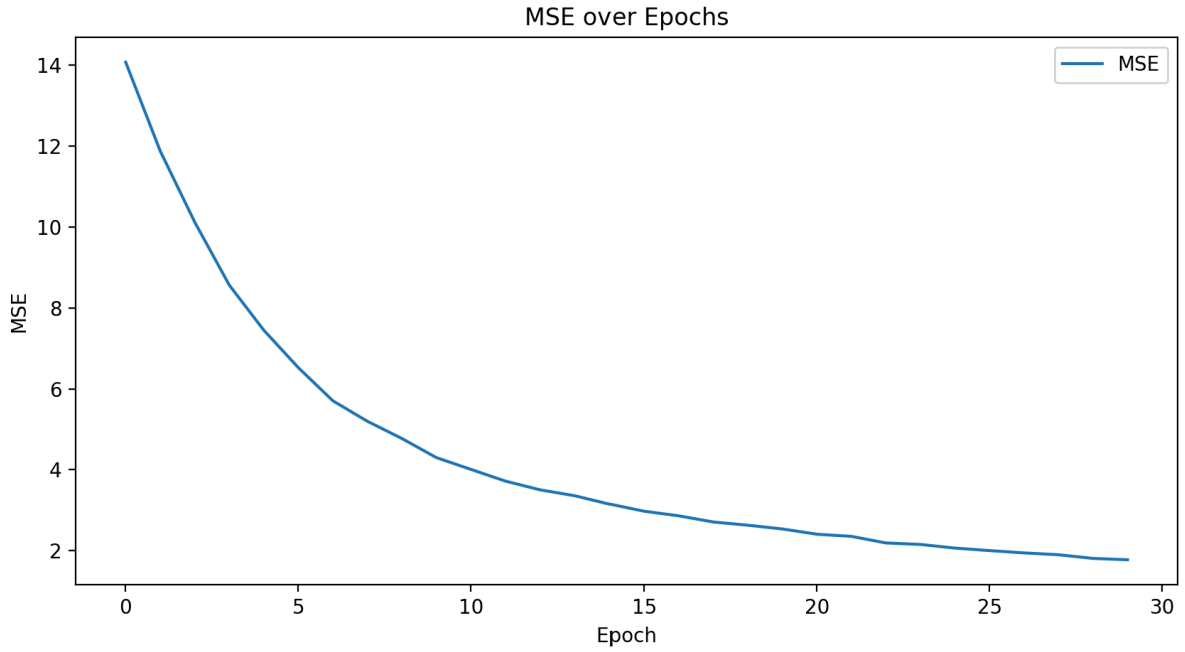


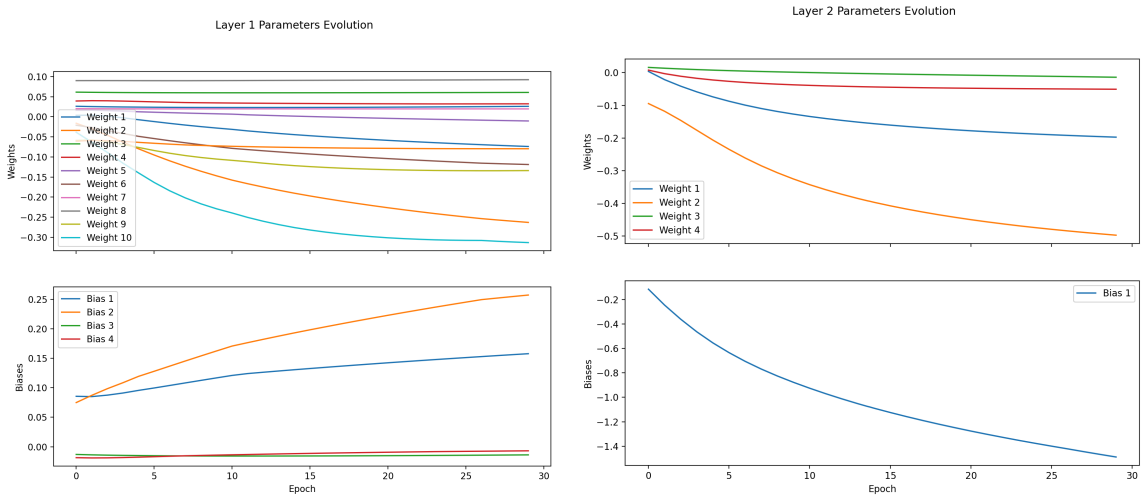


Run 6

Object	Parameter	Value
Consys	Controller	ControllerNN
Consys	Plant	plantLotkaVolterra
Consys	Epochs	30
Consys	Timesteps	10
Consys	learning_rate	0.01
Consys	D_range	[-0.02, 0.02]
ControllerNN	Hidden Layers	[4]
ControllerNN	Activation Functions	[relu]
ControllerNN	Range Initialization	[-0.1, 0.1]
plantLotkaVolterra	rabbits	1
plantLotkaVolterra	foxes	1
plantLotkaVolterra	goal_state	2
plantLotkaVolterra	rabbitBirthRate	0.1
plantLotkaVolterra	rabbitDeathRatePerFox	0.02
plantLotkaVolterra	foxDeathRate	0.1
plantLotkaVolterra	foxBirthRatePerRabbit	0.01

Kommentar: Samme her, den fungerte direkte og da gadd jeg ikke å prøve ut en masse forskjellige ting. Generelt så fikk jeg bare verre resultat da jeg la til flere skjulte lag, i alle 3 oppgavene.





3rd plant - Lotka-Volterra Plant Model

Equations:

For each timestep:

rabbits += input
foxes += disturbance

rabbits += (rabbitBirthRate * rabbits - rabbitDeathRatePerFox * rabbits * foxes)
foxes += (foxBirthRatePerRabbit * rabbits * foxes - foxDeathRate * foxes)

Where the number of rabbits and foxes can’t be negative.