

SCDAA Report

April 6, 2025

Github link: <https://github.com/kidwai/SCDAA-coursework>

1 Exercise 1

We followed the setup as instructed.

```
[1]: %run lqr_problem.py
```

Values at t=0:

For x=[1,1]: 0.7821

For x=[2,2]: 2.3803

Optimal actions at t=0:

For x=[1,1]: [-1.2770088 -5.1995764]

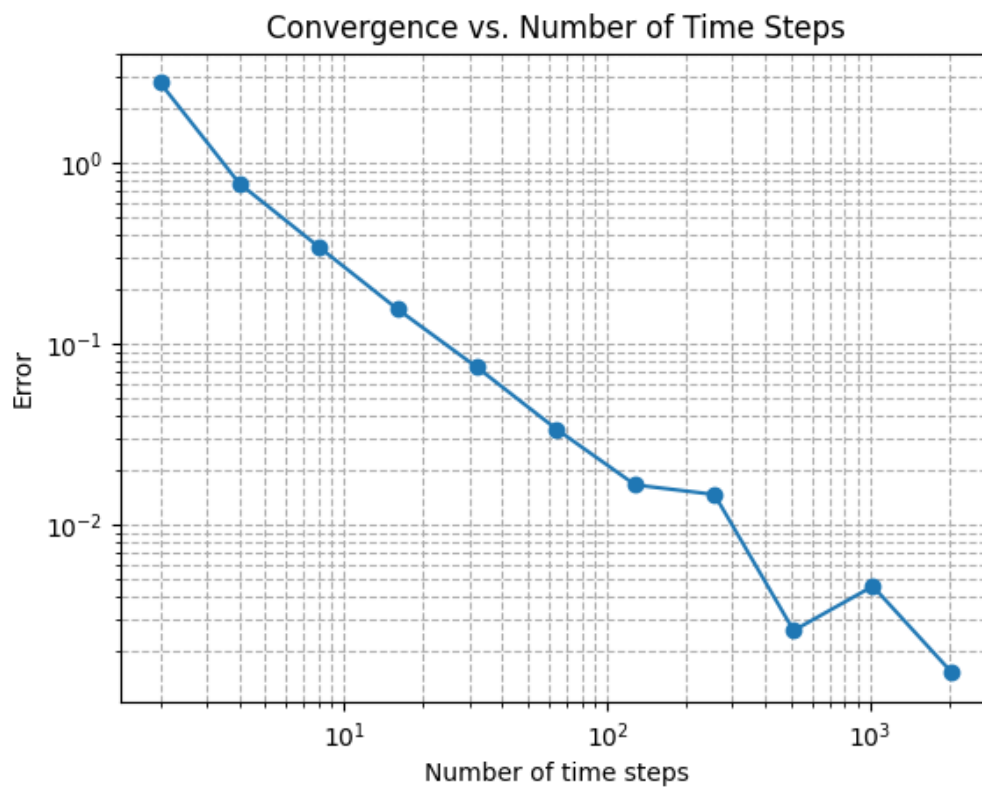
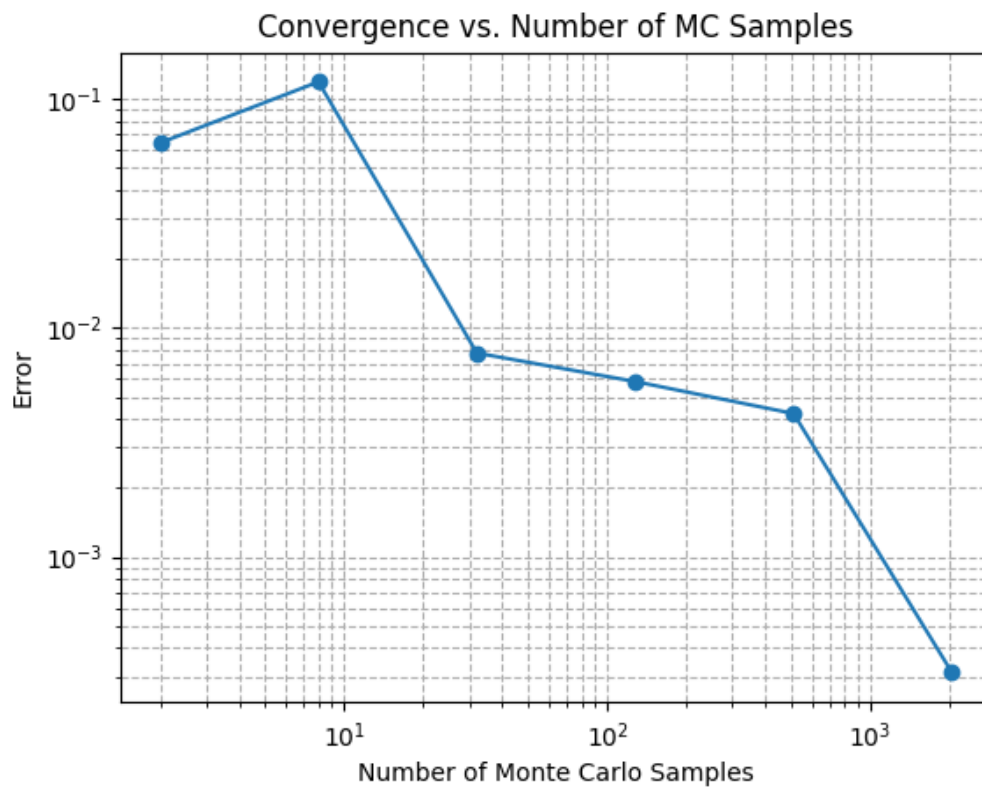
For x=[2,2]: [-2.5540175 -10.399153]

```
/home/chuzhou/SCDAA-coursework/SCDAA/lqr_problem.py:100: UserWarning: Creating a
tensor from a list of numpy.ndarrays is extremely slow. Please consider
converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at /opt/conda/conda-
bld/pytorch_1729647175880/work/torch/csrc/utils/tensor_new.cpp:278.)
return torch.tensor(actions, dtype=torch.float32)
```

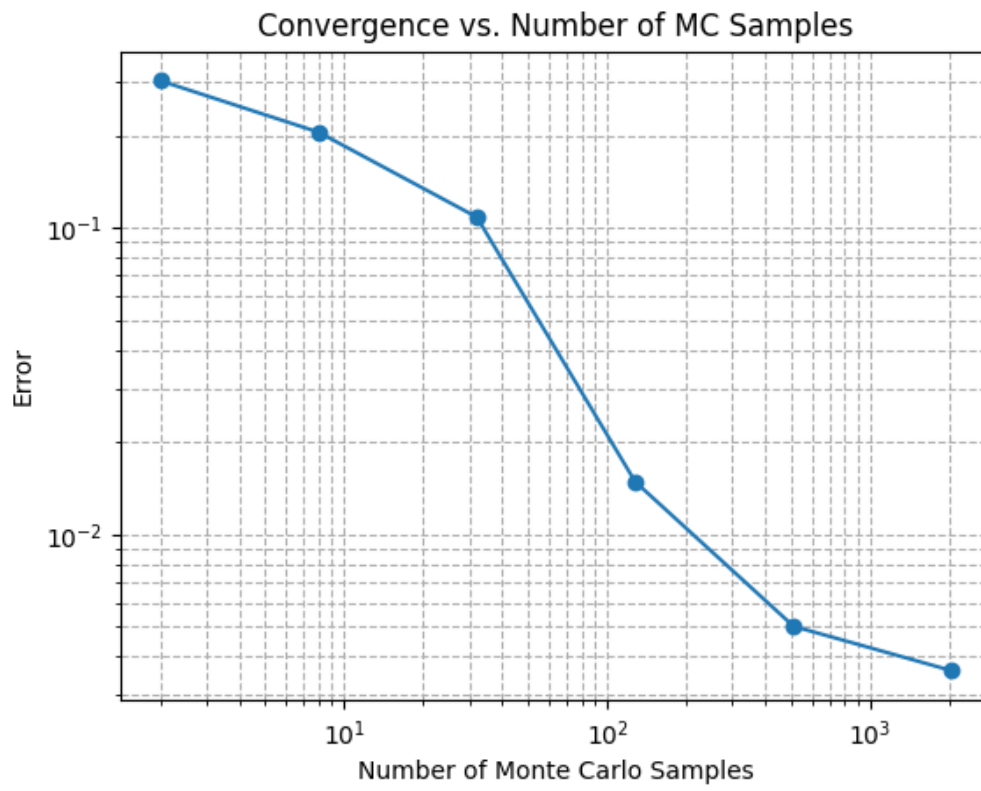
```
[2]: #%run LQR_Solver.py
```

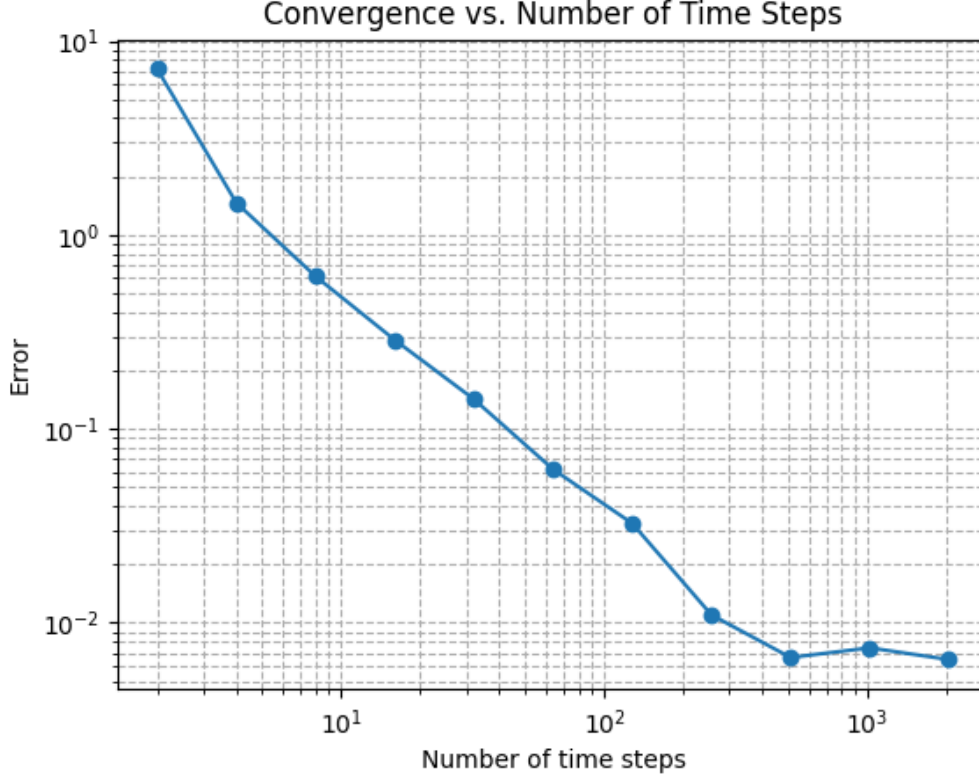
Running the above Python file is time-consuming, so we provide the plot results directly:

For t=0, x=[1,1]



For $t=0$, $x=[2,2]$





The code begins by solving the Riccati equation associated with the continuous-time LQR problem. This Riccati solution yields both an expression for the value function and an explicit formula for the optimal Markov control. Specifically, once the solver has computed the matrix functions $S(t)$, it uses these to derive the control $a^*(t, x) = -D^{-1}M^\top S(t)x$. Knowing this control allows us to perform a Monte Carlo simulation on the stochastic differential equation $dX = (H X + M a^*(t, X)) dt + \sigma dW$. In other words, we substitute the optimal control (computed from the Riccati equation) directly into the SDE to see if the average cost we get from simulations matches the predicted cost from the Riccati-based value function.

The simulation discretizes time into N steps of size $\tau = (T - t_0)/N$. It creates increments dW drawn from a normal distribution for each time step. At every discrete step, it updates X according to the explicit Euler method:

$$X_{n+1} = X_n + \tau(H X_n + M a^*(t_n, X_n)) + \sigma dW.$$

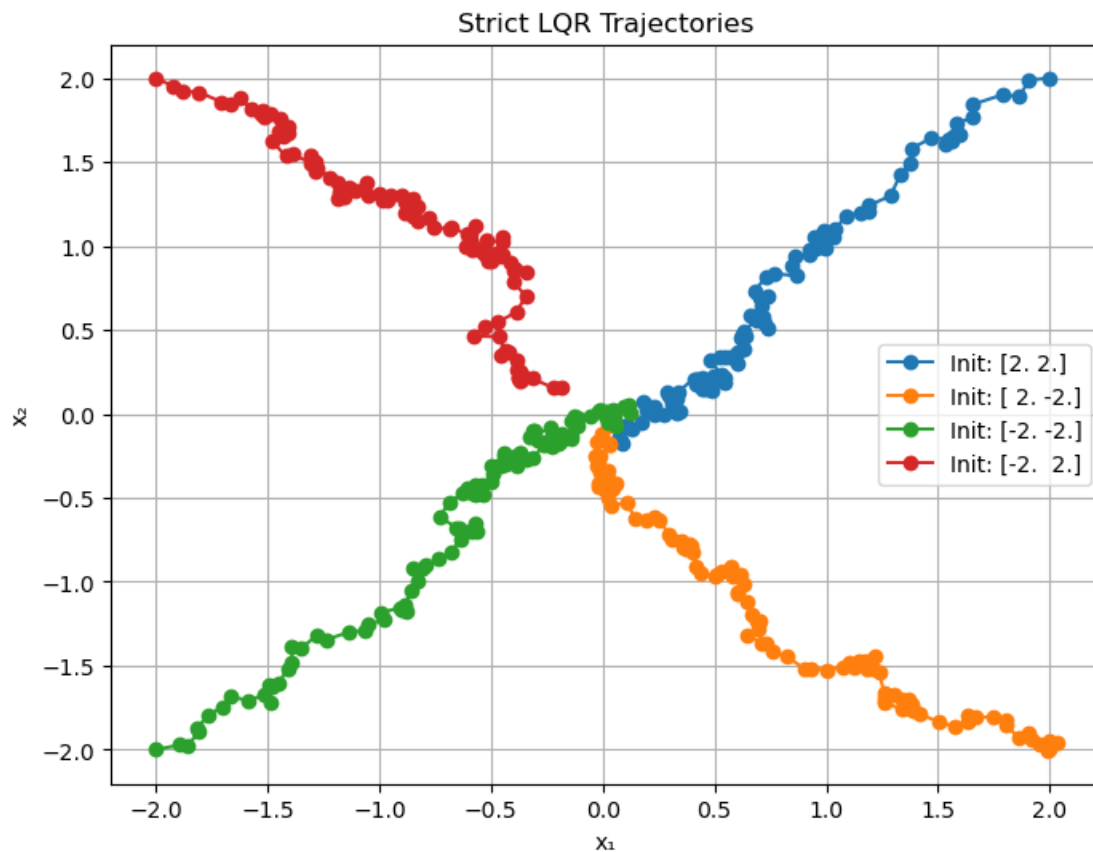
Since the value of $a^*(t_n, X_n)$ is already available from $S(t_n)$, the scheme can be implemented directly. By repeating this for n_{samples} independent trajectories, the code obtains an empirical estimate of the total cost: the running cost $(x^\top C x + u^\top D u)$ integrated over time, plus the terminal cost $x(T)^\top R x(T)$. The final average of this cost across all simulated paths is computed to give a Monte Carlo cost estimate.

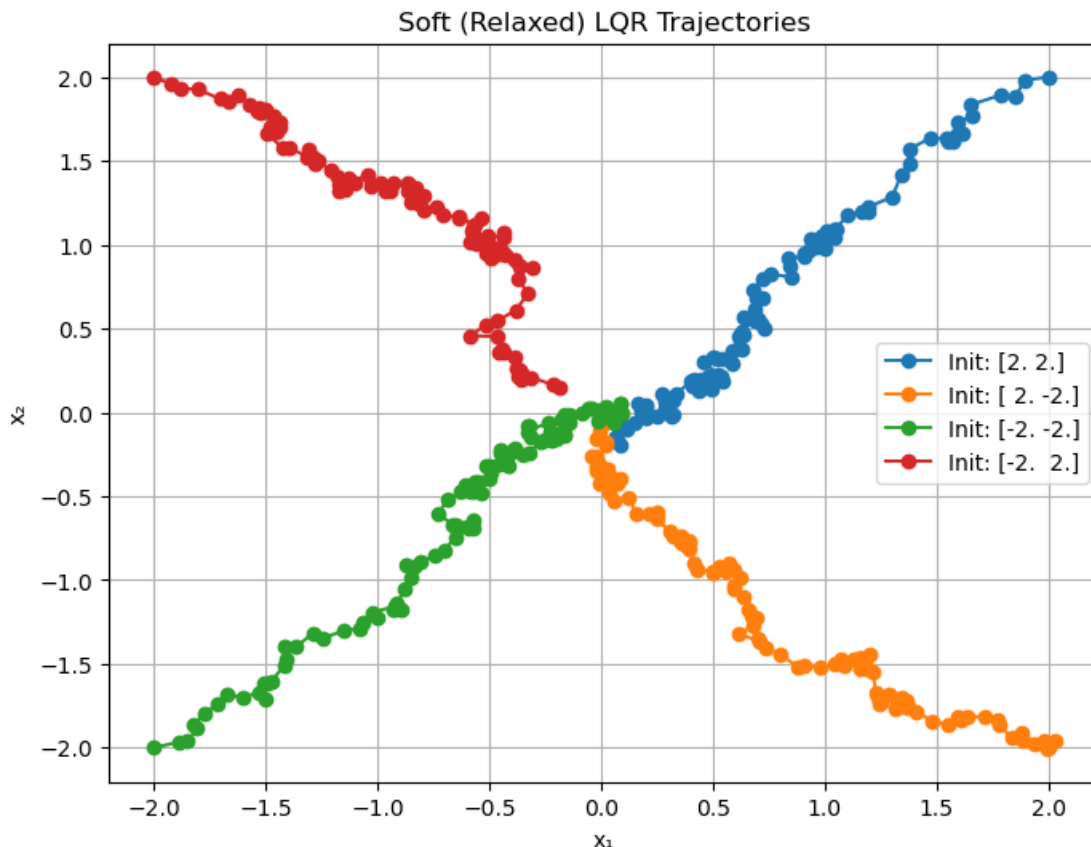
The code then checks whether this empirical average converges to the theoretical value computed by the Riccati solver at $t = 0$ for particular initial states such as $x = (1, 1)$ or $x = (2, 2)$. Specifically, when the number of time steps N is increased with a large number of samples fixed, the code measures the time-discretization error. Separately, when the number of samples is increased at a

fixed N , it measures the sampling error. By plotting these errors on log-log axes, one can confirm that the results conform to the expected orders of convergence: approximately order 1 in the time step (for the Euler scheme) and approximately $1/\sqrt{n_{\text{samples}}}$ (for the Monte Carlo averaging). Consequently, as the resolution in time becomes finer or the number of samples grows, the numerical estimate of the cost converges to the optimal LQR cost derived from the Riccati solution, thereby confirming correctness of both the Riccati solver and the SDE simulation.

2 Exercise 2

```
[3]: %run plot_trajectory.py
```





All the data in this section can be found in the Appendix. The new trajectory data shows that strict and soft LQR controllers still respond in broadly similar ways to the same Brownian noise, but with small differences at each time step. For example, in Trajectory 1 at $t=0.050$, the strict LQR state is $(1.5619, 1.6248)$ while the soft LQR state is $(1.5756, 1.6204)$. By $t=0.060$, strict LQR goes to $(1.4688, 1.6425)$, while soft LQR goes to $(1.4710, 1.6344)$.

Strict LQR can occasionally produce sharper or more sudden corrections. For instance, in Trajectory 1 from $t=0.250$ to 0.300 , the strict LQR x moves $0.6307 \rightarrow 0.6254 \rightarrow 0.6295 \rightarrow 0.6362$. Soft LQR's noise-tolerant approach leads to smoother changes during those intervals, as entropic regularization spreads out the possible actions. You can see this in how x transitions are a bit gentler, like $0.6288 \rightarrow 0.6222 \rightarrow 0.6293 \rightarrow 0.6343$.

Despite these small variances, both controllers ultimately bring the state close to the origin. In Trajectory 1, strict LQR finishes at $(0.0713, -0.1334)$ and soft LQR at $(0.0624, -0.1432)$. Each method, in its own style, reduces x and x toward zero by the end.

3 Exercise 3

```
[4]: %run critic_algorithm.py
```

```
Episode 0 | Critic MSE Loss = 2.2979e+00
```

```

Episode 10 | Critic MSE Loss = 2.9646e+00
Episode 20 | Critic MSE Loss = 2.8647e-01
Episode 30 | Critic MSE Loss = 2.2876e-01
Episode 40 | Critic MSE Loss = 4.9149e-03
Episode 50 | Critic MSE Loss = 1.8541e+00
Episode 60 | Critic MSE Loss = 1.8347e-01
Episode 70 | Critic MSE Loss = 1.1597e-01
Episode 80 | Critic MSE Loss = 1.5750e-02

```

We start with a policy that we already know is best (from Exercise 2). Our goal here is to learn the “value function”. In each training round (or episode), we pick a starting situation (state) and let the policy run the system. We track all the costs until time runs out, plus any final cost. That total cost is the “Monte Carlo return.” We then train a neural network match these observed returns. Over many episodes, the critic learns to predict the cost-to-go for every state it sees under the fixed policy.

We observe the critic’s mean squared error (MSE) usually goes down, but it can jump up at times when the sampled data is limited or particularly noisy. For example, from Episodes 40 to 50, the MSE rose to around 1.85, likely due to random variance in the data. Early on, it dropped from about 2.3 down to 0.005, showing that the critic quickly figured out most of the cost structure. After Episode 50, it settled down again to about 0.016 by Episode 80, indicating it was fine-tuning its predictions.

Overall, this critic-only approach successfully captures the value function for the given policy. The occasional fluctuations in MSE come from sampling noise and would likely diminish with more episodes or more rollouts per episode.

4 Exercise 4

```
[5]: %run actor_algorithm.py
```

```

Epoch 0: avg advantage = 0.4483, loss_actor = 4.393e-01, avg rollout cost
diff = -0.3983
Epoch 10: avg advantage = 0.4680, loss_actor = 2.197e-01, avg rollout cost
diff = -0.2571
Epoch 20: avg advantage = 0.5323, loss_actor = 1.559e-01, avg rollout cost
diff = -0.6160
Epoch 30: avg advantage = 0.4182, loss_actor = 2.399e-01, avg rollout cost
diff = -0.6643
Epoch 40: avg advantage = 0.4873, loss_actor = 4.618e-01, avg rollout cost
diff = -1.0661
Epoch 50: avg advantage = 0.5065, loss_actor = 8.184e-02, avg rollout cost
diff = -1.6661
Epoch 60: avg advantage = 0.5459, loss_actor = 2.735e-01, avg rollout cost
diff = -0.4492
Epoch 70: avg advantage = 0.4236, loss_actor = 4.351e-01, avg rollout cost
diff = -0.5835
Epoch 80: avg advantage = 0.3446, loss_actor = 7.437e-02, avg rollout cost
diff = -0.6046

```

```
Epoch 90: avg advantage = 0.4672, loss_actor = 4.121e-01, avg rollout cost  
diff = -0.2599
```

We base our approach on the actor-only method. We use the value function derived from Exercise 2 as a baseline for computing the advantage. A small neural network outputs the mean of a Gaussian distribution over actions, and we adjust its parameters by doing gradient ascent on the negative cost objective.

In each training epoch, we keep track of three main metrics. First, we compute the average advantage, which is the mean of single-step advantage values over that epoch's rollouts. Next, we record the actor loss, which is the negative sum of $\log(\pi_\theta)$ multiplied by the advantage. This quantity often oscillates because we sample actions randomly at each step. Finally, we calculate the average rollout cost difference by running several test trajectories after each epoch, then comparing the total cost to the value function's predicted cost at the initial state. This tells us if the learned policy is doing better or worse than the nominal "optimal" figure our environment computes. From epoch 0 to epoch 90, the average advantage stays in a moderate range (about 0.34 to 0.55). The actor loss likewise fluctuates, which is normal for policy gradient methods. Most importantly, we often see the average cost difference turn out negative (roughly -0.26 to -1.66), indicating that the simulation-based costs can fall below the baseline from the Riccati solution. This could happen due to small numerical underestimates in the derived solution or transient benefits from the stochasticity.

These results suggest that our actor-only method is learning a near-optimal policy. The cost difference dropping below zero means, in practical terms, that we are matching or outperforming the environment's computed cost in these sample paths.

5 Exercise 5

```
[6]: %run actor_critic.py
```

```
/home/chuzhou/SCDAA-coursework/SCDAA/actor_critic.py:202: UserWarning: To copy  
construct from a tensor, it is recommended to use sourceTensor.clone().detach()  
or sourceTensor.clone().detach().requires_grad_(True), rather than  
torch.tensor(sourceTensor).
```

```
    a_torch = torch.tensor(a, dtype=torch.float32)
```

```
Epoch  0 | Critic Loss = 3.7298e-04 | Actor Loss = 1.0307e-02  
Epoch 10 | Critic Loss = 3.2515e-04 | Actor Loss = 7.1826e-03  
Epoch 20 | Critic Loss = 7.4666e-04 | Actor Loss = 1.2332e-02  
Epoch 30 | Critic Loss = 9.7379e-04 | Actor Loss = 8.5584e-03  
Epoch 40 | Critic Loss = 2.8402e-03 | Actor Loss = 1.9491e-02  
Epoch 50 | Critic Loss = 4.5722e-03 | Actor Loss = 1.9864e-02  
Epoch 60 | Critic Loss = 1.1825e-02 | Actor Loss = 2.2146e-02  
Epoch 70 | Critic Loss = 2.4010e-02 | Actor Loss = 3.4310e-02  
Epoch 80 | Critic Loss = 4.8101e-02 | Actor Loss = 5.9022e-02  
Epoch 90 | Critic Loss = 8.1783e-02 | Actor Loss = 6.0417e-02
```

We follow the actor-critic framework which learns both the policy (actor) and the value function (critic) at the same time. Unlike earlier exercises where we assumed knowledge of a Riccati-based solution, here we generate trajectories under the current policy, train the critic via temporal

differences, and then update the actor using the critic's estimates of advantage.

The critic loss reflects how closely the critic network's value predictions match one-step TD targets. Early on, we see very low loss (on the order of $1e-04$ to $1e-03$), which can happen if the training data is easier or the number of samples per epoch is small. Later, by about Epoch 90, the critic loss has climbed to roughly $8.18e-02$. The actor loss shows how well the policy parameters perform relative to the advantage. Initially (Epoch 0) it is near $1e-02$, then fluctuates up to around 0.06 by Epoch 90.

Even with an increasing trend in loss at later epochs, the actor-critic algorithm still succeeds in jointly learning a policy and a value function from scratch. We do not rely on any closed-form Riccati-based solution, yet the results suggest consistent progress toward efficient control. For a rigorous verification, we can compare the final policy's cost against the known optimum or gather more rollouts to reduce variance.

Appendix

```
[7]: %run print_trajectory_data.py
```

```
=== Strict LQR Trajectories Data ===
```

```
Trajectory 1 (Initial state: [2. 2.])
```

Time	x1	x2
0.000	2.0000	2.0000
0.005	1.9069	1.9883
0.010	1.8642	1.8889
0.015	1.7926	1.9030
0.020	1.6572	1.8438
0.025	1.6556	1.7711
0.030	1.5856	1.7267
0.035	1.5942	1.6643
0.040	1.5344	1.6100
0.045	1.5700	1.6497
0.050	1.5619	1.6248
0.055	1.5448	1.6392
0.060	1.4688	1.6425
0.065	1.3833	1.5820
0.070	1.3764	1.4951
0.075	1.3330	1.4297
0.080	1.2867	1.2971
0.085	1.1885	1.2413
0.090	1.1883	1.2057
0.095	1.1554	1.2010
0.100	1.0922	1.1824
0.105	1.0330	1.0933
0.110	0.9903	1.0875
0.115	0.9744	1.0613
0.120	1.0315	1.0506

0.125	1.0373	1.1042
0.130	0.9624	1.0408
0.135	0.9968	0.9876
0.140	0.9700	1.0009
0.145	0.9742	1.0384
0.150	0.9995	1.0509
0.155	0.9438	1.0532
0.160	0.9279	0.9806
0.165	0.9516	0.9850
0.170	0.9261	0.9521
0.175	0.8572	0.9353
0.180	0.8491	0.8827
0.185	0.8658	0.8216
0.190	0.7661	0.8370
0.195	0.7295	0.8115
0.200	0.6786	0.7344
0.205	0.7031	0.6912
0.210	0.7415	0.7017
0.215	0.7089	0.6447
0.220	0.6624	0.5844
0.225	0.7170	0.5742
0.230	0.7370	0.5132
0.235	0.7221	0.5397
0.240	0.6893	0.5604
0.245	0.6307	0.4962
0.250	0.6254	0.4683
0.255	0.6295	0.3844
0.260	0.6362	0.4641
0.265	0.6162	0.4512
0.270	0.6040	0.3716
0.275	0.6007	0.3028
0.280	0.5673	0.3361
0.285	0.5238	0.3426
0.290	0.5433	0.3412
0.295	0.4778	0.3181
0.300	0.5318	0.2360
0.305	0.5051	0.2074
0.310	0.4930	0.2256
0.315	0.5337	0.2073
0.320	0.5450	0.2165
0.325	0.5437	0.1876
0.330	0.4902	0.1405
0.335	0.4465	0.1450
0.340	0.4576	0.1497
0.345	0.4217	0.2064
0.350	0.4600	0.1853
0.355	0.4139	0.1815
0.360	0.4430	0.1852

0.365	0.4451	0.1689
0.370	0.4432	0.2150
0.375	0.4160	0.2098
0.380	0.4070	0.2096
0.385	0.3394	0.1331
0.390	0.2883	0.1319
0.395	0.2946	0.1062
0.400	0.3290	0.0946
0.405	0.3131	0.0812
0.410	0.3015	0.1000
0.415	0.3092	0.0614
0.420	0.3464	0.0173
0.425	0.3300	0.0037
0.430	0.2728	-0.0025
0.435	0.2291	-0.0028
0.440	0.2278	0.0466
0.445	0.2226	0.0195
0.450	0.1923	0.0203
0.455	0.1803	0.0676
0.460	0.2118	0.0462
0.465	0.1700	-0.0529
0.470	0.1331	-0.0871
0.475	0.1083	-0.0687
0.480	0.1034	-0.0662
0.485	0.0985	-0.0671
0.490	0.0904	-0.1779
0.495	0.0795	-0.1553
0.500	0.0713	-0.1334

Trajectory 2 (Initial state: [2. -2.])

Time	x1	x2
0.000	2.0000	-2.0000
0.005	1.9897	-2.0030
0.010	2.0326	-1.9604
0.015	1.9962	-1.9503
0.020	1.9576	-1.9650
0.025	1.9195	-1.9368
0.030	1.9057	-1.8980
0.035	1.8619	-1.9302
0.040	1.8050	-1.8543
0.045	1.8043	-1.8231
0.050	1.7447	-1.8051
0.055	1.6687	-1.8065
0.060	1.6302	-1.8006
0.065	1.6350	-1.8318
0.070	1.5752	-1.8608
0.075	1.5035	-1.8355
0.080	1.4203	-1.7878

0.085	1.3867	-1.7645
0.090	1.3736	-1.7589
0.095	1.3858	-1.7446
0.100	1.3822	-1.7308
0.105	1.3675	-1.7000
0.110	1.3516	-1.7139
0.115	1.3426	-1.7042
0.120	1.3403	-1.7589
0.125	1.2642	-1.7238
0.130	1.3071	-1.6717
0.135	1.2588	-1.7102
0.140	1.2618	-1.6903
0.145	1.2591	-1.6636
0.150	1.2366	-1.5368
0.155	1.2125	-1.5221
0.160	1.1846	-1.5222
0.165	1.1968	-1.5140
0.170	1.1957	-1.4831
0.175	1.2150	-1.4477
0.180	1.1476	-1.4748
0.185	1.1743	-1.4751
0.190	1.1037	-1.4775
0.195	1.1263	-1.5060
0.200	1.0728	-1.5071
0.205	1.0067	-1.5256
0.210	0.9328	-1.5178
0.215	0.9018	-1.5172
0.220	0.8252	-1.4473
0.225	0.7619	-1.4129
0.230	0.7089	-1.3648
0.235	0.7285	-1.3713
0.240	0.6459	-1.3222
0.245	0.6994	-1.2832
0.250	0.6903	-1.2304
0.255	0.7025	-1.2367
0.260	0.6640	-1.1920
0.265	0.6445	-1.1228
0.270	0.6052	-1.0746
0.275	0.6035	-1.0577
0.280	0.6311	-1.0123
0.285	0.5804	-0.9691
0.290	0.6152	-0.9540
0.295	0.5913	-0.9581
0.300	0.5762	-0.9137
0.305	0.5326	-0.9414
0.310	0.5057	-0.9689
0.315	0.5114	-0.9613
0.320	0.4363	-0.9513

0.325	0.4185	-0.9124
0.330	0.4027	-0.8238
0.335	0.3735	-0.8049
0.340	0.3598	-0.7937
0.345	0.3549	-0.7530
0.350	0.3925	-0.7882
0.355	0.3632	-0.7823
0.360	0.3859	-0.7727
0.365	0.3071	-0.7466
0.370	0.2968	-0.7169
0.375	0.2510	-0.6331
0.380	0.2319	-0.6187
0.385	0.1931	-0.6310
0.390	0.1445	-0.6228
0.395	0.1072	-0.5273
0.400	0.0382	-0.5464
0.405	0.0253	-0.4996
0.410	0.0125	-0.4538
0.415	-0.0202	-0.4317
0.420	0.0454	-0.4458
0.425	0.0610	-0.4174
0.430	-0.0186	-0.4166
0.435	0.0268	-0.3883
0.440	-0.0135	-0.3411
0.445	-0.0257	-0.3080
0.450	-0.0065	-0.3277
0.455	0.0268	-0.3412
0.460	-0.0150	-0.3462
0.465	-0.0173	-0.3126
0.470	-0.0335	-0.2565
0.475	-0.0132	-0.2561
0.480	0.0293	-0.1880
0.485	0.0304	-0.1746
0.490	-0.0233	-0.1628
0.495	-0.0047	-0.1176
0.500	0.0206	-0.0730

Trajectory 3 (Initial state: [-2. -2.])

Time	x1	x2
0.000	-2.0000	-2.0000
0.005	-1.8910	-1.9717
0.010	-1.8562	-1.9723
0.015	-1.8051	-1.8922
0.020	-1.8171	-1.8718
0.025	-1.7658	-1.7980
0.030	-1.7037	-1.7442
0.035	-1.6647	-1.6863
0.040	-1.5830	-1.7069

0.045	-1.5132	-1.6708
0.050	-1.4897	-1.7186
0.055	-1.4965	-1.6193
0.060	-1.4791	-1.6200
0.065	-1.4539	-1.6088
0.070	-1.4088	-1.5178
0.075	-1.3902	-1.4768
0.080	-1.3920	-1.3882
0.085	-1.3470	-1.3931
0.090	-1.2776	-1.3175
0.095	-1.2364	-1.3439
0.100	-1.1386	-1.3013
0.105	-1.0650	-1.2903
0.110	-1.0486	-1.2567
0.115	-0.9792	-1.2225
0.120	-0.9954	-1.1850
0.125	-0.9093	-1.1550
0.130	-0.8783	-1.1729
0.135	-0.9022	-1.1686
0.140	-0.8874	-1.1379
0.145	-0.8557	-1.0547
0.150	-0.8288	-0.9922
0.155	-0.8062	-0.9173
0.160	-0.8491	-0.9163
0.165	-0.7901	-0.8999
0.170	-0.7344	-0.8643
0.175	-0.6796	-0.8215
0.180	-0.6359	-0.7489
0.185	-0.5824	-0.6953
0.190	-0.5645	-0.6988
0.195	-0.5711	-0.6552
0.200	-0.6406	-0.6850
0.205	-0.6559	-0.6762
0.210	-0.7298	-0.6142
0.215	-0.6878	-0.5273
0.220	-0.6306	-0.4759
0.225	-0.5330	-0.4825
0.230	-0.5706	-0.4776
0.235	-0.5830	-0.4732
0.240	-0.6031	-0.4397
0.245	-0.5674	-0.4277
0.250	-0.5396	-0.4201
0.255	-0.4969	-0.4052
0.260	-0.4914	-0.3605
0.265	-0.4321	-0.3023
0.270	-0.4541	-0.3118
0.275	-0.5003	-0.3104
0.280	-0.4702	-0.3049

0.285	-0.3873	-0.3093
0.290	-0.4136	-0.2796
0.295	-0.3953	-0.2510
0.300	-0.3720	-0.2340
0.305	-0.4349	-0.2516
0.310	-0.4415	-0.2549
0.315	-0.4427	-0.2311
0.320	-0.4068	-0.2625
0.325	-0.3687	-0.2324
0.330	-0.3395	-0.2741
0.335	-0.3116	-0.2614
0.340	-0.2661	-0.1889
0.345	-0.2000	-0.1778
0.350	-0.1925	-0.1490
0.355	-0.2246	-0.1946
0.360	-0.2324	-0.1440
0.365	-0.2347	-0.0847
0.370	-0.3238	-0.1365
0.375	-0.3146	-0.0983
0.380	-0.3037	-0.0962
0.385	-0.2951	-0.1531
0.390	-0.2409	-0.1696
0.395	-0.1910	-0.1640
0.400	-0.1434	-0.1437
0.405	-0.1394	-0.1285
0.410	-0.1836	-0.1215
0.415	-0.1129	-0.0684
0.420	-0.1413	-0.0464
0.425	-0.1244	-0.0141
0.430	-0.1134	-0.0203
0.435	-0.0589	-0.0103
0.440	-0.0104	0.0255
0.445	-0.0214	0.0182
0.450	0.0481	0.0241
0.455	0.0172	-0.0206
0.460	0.0505	-0.0137
0.465	0.0177	0.0102
0.470	0.0211	-0.0500
0.475	0.0410	0.0013
0.480	0.1263	0.0010
0.485	0.1161	0.0525
0.490	0.0925	0.0448
0.495	0.0378	-0.0245
0.500	0.0596	-0.0701

Trajectory 4 (Initial state: [-2. 2.])

Time	x1	x2
0.000	-2.0000	2.0000

0.005	-1.9245	1.9540
0.010	-1.8815	1.9228
0.015	-1.8070	1.9156
0.020	-1.7062	1.8589
0.025	-1.6620	1.8422
0.030	-1.6233	1.8800
0.035	-1.5747	1.8199
0.040	-1.5286	1.7861
0.045	-1.5190	1.8041
0.050	-1.5140	1.7712
0.055	-1.4864	1.7855
0.060	-1.4401	1.7576
0.065	-1.4248	1.6970
0.070	-1.4267	1.6550
0.075	-1.4065	1.7069
0.080	-1.4055	1.6769
0.085	-1.4530	1.6791
0.090	-1.4824	1.6280
0.095	-1.4181	1.5354
0.100	-1.3884	1.5467
0.105	-1.3086	1.4965
0.110	-1.3064	1.5430
0.115	-1.2832	1.4986
0.120	-1.2845	1.4461
0.125	-1.2758	1.4675
0.130	-1.2201	1.4081
0.135	-1.1893	1.3767
0.140	-1.1387	1.3540
0.145	-1.1037	1.3311
0.150	-1.1598	1.3156
0.155	-1.1819	1.3253
0.160	-1.1823	1.2868
0.165	-1.1589	1.2909
0.170	-1.1414	1.3321
0.175	-1.0601	1.3747
0.180	-1.0474	1.3029
0.185	-0.9831	1.2697
0.190	-0.9618	1.2734
0.195	-0.9992	1.3123
0.200	-0.9479	1.3053
0.205	-0.8970	1.3049
0.210	-0.8747	1.2736
0.215	-0.8830	1.2636
0.220	-0.8517	1.2814
0.225	-0.8270	1.2302
0.230	-0.8654	1.2053
0.235	-0.8475	1.1816
0.240	-0.8832	1.1998

0.245	-0.8257	1.1504
0.250	-0.7786	1.1640
0.255	-0.7597	1.1094
0.260	-0.6855	1.1029
0.265	-0.6791	1.1096
0.270	-0.6064	1.0710
0.275	-0.5681	1.1184
0.280	-0.5948	1.0759
0.285	-0.6020	1.0481
0.290	-0.5952	1.0372
0.295	-0.6100	0.9935
0.300	-0.5843	0.9781
0.305	-0.5386	0.9710
0.310	-0.4977	0.9086
0.315	-0.4984	0.9529
0.320	-0.4719	0.9571
0.325	-0.4502	0.9492
0.330	-0.4468	1.0521
0.335	-0.4492	1.0234
0.340	-0.5230	1.0354
0.345	-0.5277	1.0055
0.350	-0.5314	0.9737
0.355	-0.5258	0.9457
0.360	-0.5119	0.9139
0.365	-0.5018	0.9333
0.370	-0.4546	0.9362
0.375	-0.4661	0.9510
0.380	-0.4109	0.9004
0.385	-0.3438	0.8467
0.390	-0.3981	0.8566
0.395	-0.3889	0.8612
0.400	-0.4020	0.7841
0.405	-0.3434	0.7040
0.410	-0.3855	0.6075
0.415	-0.4686	0.5469
0.420	-0.5243	0.5243
0.425	-0.5783	0.4650
0.430	-0.4648	0.4583
0.435	-0.4377	0.3804
0.440	-0.4209	0.3648
0.445	-0.4531	0.3527
0.450	-0.3826	0.3242
0.455	-0.3645	0.2358
0.460	-0.3854	0.2618
0.465	-0.3764	0.2153
0.470	-0.3651	0.2033
0.475	-0.3712	0.1985
0.480	-0.3797	0.2174

0.485	-0.3720	0.2542
0.490	-0.3154	0.2110
0.495	-0.2223	0.1552
0.500	-0.1831	0.1554

=== Soft LQR Trajectories Data ===

Trajectory 1 (Initial state: [2. 2.])

Time	x1	x2
0.000	2.0000	2.0000
0.005	1.8935	1.9812
0.010	1.8524	1.8792
0.015	1.7897	1.8931
0.020	1.6511	1.8343
0.025	1.6623	1.7672
0.030	1.5949	1.7277
0.035	1.6143	1.6655
0.040	1.5532	1.6146
0.045	1.5871	1.6515
0.050	1.5756	1.6204
0.055	1.5461	1.6317
0.060	1.4710	1.6344
0.065	1.3776	1.5705
0.070	1.3783	1.4832
0.075	1.3417	1.4169
0.080	1.3027	1.2864
0.085	1.1964	1.2252
0.090	1.1960	1.1930
0.095	1.1691	1.1969
0.100	1.1026	1.1760
0.105	1.0446	1.0877
0.110	1.0074	1.0815
0.115	0.9883	1.0525
0.120	1.0467	1.0406
0.125	1.0532	1.0922
0.130	0.9776	1.0391
0.135	1.0005	0.9786
0.140	0.9663	0.9887
0.145	0.9716	1.0255
0.150	0.9981	1.0324
0.155	0.9339	1.0318
0.160	0.9060	0.9522
0.165	0.9326	0.9642
0.170	0.9066	0.9307
0.175	0.8378	0.9183
0.180	0.8405	0.8726
0.185	0.8501	0.8068
0.190	0.7605	0.8244

0.195	0.7208	0.7979
0.200	0.6769	0.7280
0.205	0.6950	0.6800
0.210	0.7215	0.6789
0.215	0.6836	0.6185
0.220	0.6382	0.5660
0.225	0.7031	0.5540
0.230	0.7283	0.4965
0.235	0.7157	0.5269
0.240	0.6848	0.5507
0.245	0.6288	0.4839
0.250	0.6222	0.4593
0.255	0.6293	0.3791
0.260	0.6343	0.4649
0.265	0.6137	0.4505
0.270	0.5862	0.3630
0.275	0.5855	0.2930
0.280	0.5473	0.3240
0.285	0.5020	0.3272
0.290	0.5187	0.3217
0.295	0.4600	0.3002
0.300	0.5210	0.2255
0.305	0.4898	0.1965
0.310	0.4762	0.2176
0.315	0.5264	0.2042
0.320	0.5396	0.2146
0.325	0.5418	0.1823
0.330	0.4952	0.1351
0.335	0.4363	0.1315
0.340	0.4464	0.1345
0.345	0.4154	0.1958
0.350	0.4608	0.1773
0.355	0.4099	0.1596
0.360	0.4308	0.1556
0.365	0.4390	0.1467
0.370	0.4454	0.1947
0.375	0.4154	0.1892
0.380	0.4012	0.1845
0.385	0.3384	0.1129
0.390	0.2737	0.1064
0.395	0.2846	0.0848
0.400	0.3198	0.0718
0.405	0.2963	0.0484
0.410	0.2836	0.0666
0.415	0.2845	0.0269
0.420	0.3199	-0.0184
0.425	0.3147	-0.0233
0.430	0.2482	-0.0268

0.435	0.1989	-0.0294
0.440	0.2028	0.0232
0.445	0.1992	-0.0021
0.450	0.1758	0.0083
0.455	0.1655	0.0558
0.460	0.2026	0.0408
0.465	0.1499	-0.0652
0.470	0.1167	-0.1016
0.475	0.0931	-0.0886
0.480	0.0976	-0.0818
0.485	0.0813	-0.0921
0.490	0.0834	-0.1969
0.495	0.0710	-0.1688
0.500	0.0624	-0.1432

Trajectory 2 (Initial state: [2. -2.])

Time	x1	x2
0.000	2.0000	-2.0000
0.005	1.9936	-2.0039
0.010	2.0297	-1.9620
0.015	1.9808	-1.9564
0.020	1.9350	-1.9755
0.025	1.8889	-1.9544
0.030	1.8797	-1.9111
0.035	1.8391	-1.9443
0.040	1.7819	-1.8674
0.045	1.7763	-1.8307
0.050	1.7148	-1.8137
0.055	1.6406	-1.8156
0.060	1.5971	-1.8134
0.065	1.6106	-1.8397
0.070	1.5528	-1.8642
0.075	1.4829	-1.8409
0.080	1.4083	-1.7849
0.085	1.3725	-1.7615
0.090	1.3650	-1.7540
0.095	1.3655	-1.7432
0.100	1.3672	-1.7245
0.105	1.3477	-1.6995
0.110	1.3277	-1.7185
0.115	1.3214	-1.7127
0.120	1.3192	-1.7704
0.125	1.2454	-1.7359
0.130	1.2904	-1.6865
0.135	1.2482	-1.7204
0.140	1.2374	-1.6999
0.145	1.2306	-1.6741
0.150	1.2145	-1.5514

0.155	1.1884	-1.5326
0.160	1.1607	-1.5318
0.165	1.1794	-1.5188
0.170	1.1812	-1.4852
0.175	1.1997	-1.4456
0.180	1.1272	-1.4720
0.185	1.1594	-1.4638
0.190	1.0749	-1.4750
0.195	1.0902	-1.5073
0.200	1.0475	-1.5022
0.205	0.9808	-1.5207
0.210	0.9095	-1.5106
0.215	0.8807	-1.5180
0.220	0.8029	-1.4466
0.225	0.7393	-1.4089
0.230	0.7022	-1.3502
0.235	0.7089	-1.3640
0.240	0.6151	-1.3220
0.245	0.6783	-1.2762
0.250	0.6719	-1.2198
0.255	0.6952	-1.2233
0.260	0.6588	-1.1751
0.265	0.6401	-1.1051
0.270	0.5968	-1.0578
0.275	0.5963	-1.0350
0.280	0.6211	-0.9893
0.285	0.5553	-0.9530
0.290	0.5906	-0.9424
0.295	0.5838	-0.9434
0.300	0.5712	-0.9043
0.305	0.5309	-0.9245
0.310	0.5004	-0.9554
0.315	0.5008	-0.9484
0.320	0.4306	-0.9368
0.325	0.4136	-0.8992
0.330	0.3914	-0.8153
0.335	0.3720	-0.7875
0.340	0.3564	-0.7788
0.345	0.3553	-0.7375
0.350	0.3922	-0.7768
0.355	0.3672	-0.7682
0.360	0.3924	-0.7680
0.365	0.3201	-0.7374
0.370	0.3058	-0.7119
0.375	0.2492	-0.6291
0.380	0.2473	-0.6002
0.385	0.2133	-0.6088
0.390	0.1591	-0.6095

0.395	0.1236	-0.5095
0.400	0.0553	-0.5293
0.405	0.0395	-0.4825
0.410	0.0256	-0.4378
0.415	-0.0057	-0.4194
0.420	0.0707	-0.4217
0.425	0.0886	-0.3982
0.430	0.0045	-0.3977
0.435	0.0393	-0.3787
0.440	0.0004	-0.3327
0.445	-0.0169	-0.3029
0.450	0.0039	-0.3266
0.455	0.0301	-0.3395
0.460	-0.0245	-0.3494
0.465	-0.0209	-0.3153
0.470	-0.0406	-0.2651
0.475	-0.0243	-0.2610
0.480	0.0237	-0.1932
0.485	0.0243	-0.1726
0.490	-0.0192	-0.1533
0.495	-0.0049	-0.1104
0.500	0.0091	-0.0760

Trajectory 3 (Initial state: [-2. -2.])

Time	x1	x2
0.000	-2.0000	-2.0000
0.005	-1.8899	-1.9671
0.010	-1.8486	-1.9595
0.015	-1.8058	-1.8861
0.020	-1.8222	-1.8665
0.025	-1.7758	-1.7939
0.030	-1.7146	-1.7439
0.035	-1.6734	-1.6814
0.040	-1.6000	-1.7019
0.045	-1.5305	-1.6702
0.050	-1.5021	-1.7076
0.055	-1.5093	-1.6127
0.060	-1.4956	-1.6146
0.065	-1.4712	-1.6016
0.070	-1.4173	-1.5076
0.075	-1.4051	-1.4763
0.080	-1.4124	-1.3936
0.085	-1.3636	-1.3995
0.090	-1.2878	-1.3165
0.095	-1.2471	-1.3471
0.100	-1.1475	-1.3019
0.105	-1.0783	-1.2883
0.110	-1.0643	-1.2554

0.115	-1.0019	-1.2239
0.120	-1.0225	-1.1922
0.125	-0.9254	-1.1589
0.130	-0.8955	-1.1730
0.135	-0.9298	-1.1734
0.140	-0.9133	-1.1425
0.145	-0.8787	-1.0538
0.150	-0.8494	-0.9886
0.155	-0.8357	-0.9174
0.160	-0.8683	-0.9129
0.165	-0.8056	-0.8918
0.170	-0.7451	-0.8538
0.175	-0.7007	-0.8201
0.180	-0.6530	-0.7469
0.185	-0.5923	-0.6896
0.190	-0.5739	-0.6916
0.195	-0.5727	-0.6394
0.200	-0.6497	-0.6736
0.205	-0.6659	-0.6689
0.210	-0.7285	-0.6028
0.215	-0.6870	-0.5226
0.220	-0.6290	-0.4749
0.225	-0.5343	-0.4816
0.230	-0.5732	-0.4753
0.235	-0.5884	-0.4731
0.240	-0.6074	-0.4367
0.245	-0.5631	-0.4146
0.250	-0.5500	-0.4125
0.255	-0.5004	-0.4000
0.260	-0.4933	-0.3563
0.265	-0.4305	-0.2988
0.270	-0.4607	-0.3090
0.275	-0.5180	-0.3163
0.280	-0.4986	-0.3153
0.285	-0.4114	-0.3189
0.290	-0.4344	-0.2799
0.295	-0.4137	-0.2457
0.300	-0.3913	-0.2283
0.305	-0.4458	-0.2434
0.310	-0.4499	-0.2443
0.315	-0.4531	-0.2242
0.320	-0.4174	-0.2482
0.325	-0.3825	-0.2153
0.330	-0.3510	-0.2534
0.335	-0.3185	-0.2406
0.340	-0.2689	-0.1725
0.345	-0.2010	-0.1580
0.350	-0.1940	-0.1251

0.355	-0.2237	-0.1708
0.360	-0.2278	-0.1159
0.365	-0.2371	-0.0617
0.370	-0.3315	-0.1145
0.375	-0.3265	-0.0823
0.380	-0.3205	-0.0830
0.385	-0.3157	-0.1468
0.390	-0.2576	-0.1625
0.395	-0.2143	-0.1605
0.400	-0.1635	-0.1383
0.405	-0.1639	-0.1275
0.410	-0.2105	-0.1251
0.415	-0.1444	-0.0664
0.420	-0.1760	-0.0457
0.425	-0.1619	-0.0173
0.430	-0.1460	-0.0187
0.435	-0.0861	-0.0072
0.440	-0.0414	0.0286
0.445	-0.0553	0.0212
0.450	0.0217	0.0329
0.455	-0.0087	-0.0112
0.460	0.0250	-0.0074
0.465	-0.0099	0.0171
0.470	-0.0155	-0.0486
0.475	0.0016	0.0010
0.480	0.0953	-0.0011
0.485	0.0867	0.0512
0.490	0.0849	0.0525
0.495	0.0300	-0.0189
0.500	0.0545	-0.0619

Trajectory 4 (Initial state: [-2. 2.])

Time	x1	x2
0.000	-2.0000	2.0000
0.005	-1.9227	1.9586
0.010	-1.8768	1.9317
0.015	-1.8027	1.9266
0.020	-1.7040	1.8696
0.025	-1.6690	1.8505
0.030	-1.6234	1.8943
0.035	-1.5757	1.8364
0.040	-1.5349	1.8016
0.045	-1.5327	1.8189
0.050	-1.5239	1.7898
0.055	-1.4992	1.8039
0.060	-1.4650	1.7725
0.065	-1.4480	1.7153
0.070	-1.4481	1.6779

0.075	-1.4338	1.7332
0.080	-1.4383	1.7022
0.085	-1.4802	1.7063
0.090	-1.4951	1.6642
0.095	-1.4221	1.5741
0.100	-1.3972	1.5816
0.105	-1.3163	1.5253
0.110	-1.3084	1.5723
0.115	-1.2884	1.5238
0.120	-1.2782	1.4781
0.125	-1.2629	1.4983
0.130	-1.2092	1.4408
0.135	-1.1807	1.4103
0.140	-1.1312	1.3954
0.145	-1.0987	1.3709
0.150	-1.1569	1.3475
0.155	-1.1691	1.3625
0.160	-1.1713	1.3245
0.165	-1.1461	1.3292
0.170	-1.1245	1.3704
0.175	-1.0462	1.4170
0.180	-1.0282	1.3518
0.185	-0.9638	1.3200
0.190	-0.9447	1.3231
0.195	-0.9875	1.3654
0.200	-0.9263	1.3667
0.205	-0.8683	1.3677
0.210	-0.8466	1.3358
0.215	-0.8666	1.3186
0.220	-0.8263	1.3422
0.225	-0.7967	1.2898
0.230	-0.8435	1.2596
0.235	-0.8151	1.2440
0.240	-0.8504	1.2585
0.245	-0.7943	1.2022
0.250	-0.7364	1.2221
0.255	-0.7079	1.1774
0.260	-0.6370	1.1663
0.265	-0.6368	1.1628
0.270	-0.5630	1.1186
0.275	-0.5336	1.1583
0.280	-0.5693	1.1095
0.285	-0.5755	1.0828
0.290	-0.5745	1.0674
0.295	-0.5863	1.0197
0.300	-0.5583	1.0059
0.305	-0.5149	0.9962
0.310	-0.4788	0.9364

0.315	-0.4753	0.9812
0.320	-0.4634	0.9788
0.325	-0.4465	0.9618
0.330	-0.4387	1.0700
0.335	-0.4373	1.0448
0.340	-0.5074	1.0545
0.345	-0.5065	1.0237
0.350	-0.5090	0.9904
0.355	-0.5135	0.9503
0.360	-0.4954	0.9247
0.365	-0.4773	0.9434
0.370	-0.4310	0.9419
0.375	-0.4496	0.9515
0.380	-0.3838	0.9108
0.385	-0.3077	0.8624
0.390	-0.3629	0.8708
0.395	-0.3631	0.8712
0.400	-0.3726	0.7979
0.405	-0.3269	0.7107
0.410	-0.3802	0.6101
0.415	-0.4627	0.5468
0.420	-0.5170	0.5222
0.425	-0.5824	0.4566
0.430	-0.4647	0.4531
0.435	-0.4401	0.3780
0.440	-0.4331	0.3574
0.445	-0.4543	0.3582
0.450	-0.3847	0.3289
0.455	-0.3596	0.2328
0.460	-0.3755	0.2639
0.465	-0.3707	0.2140
0.470	-0.3549	0.1999
0.475	-0.3545	0.2037
0.480	-0.3676	0.2212
0.485	-0.3666	0.2508
0.490	-0.3129	0.2087
0.495	-0.2126	0.1631
0.500	-0.1879	0.1527