

```
In [2]: #####
#
# DEPENDENCIES: python 3.12, tensorflow 3.18.0
# main libraries: numpy, scipy, matplotlib, tensorflow
# sublibraries: tensorflow.keras.layers, scipy.integrate.odeint, matplotlib.
# HW6utils training_curves

import numpy as np
import tensorflow as tf

from tensorflow import keras
from tensorflow.keras import layers
from scipy.integrate import odeint
import matplotlib.pyplot as plt

from HW6utils import training_curves
```

```
In [5]: #####
#
# TO DO:
# 1) Use the scipy function odeint to numerically integrate Chua's state equations over the interval [0,30] with initial states  $x(0)=.1$ ,  $y(0)=z(0)=0$ . Sample the trajectory every 0.1 seconds. Assume the parameters for the system equations are  $\alpha = 16$ ,  $\beta = 28$ ,  $m_0 = -1.2$ , and  $m_1 = -0.7$ .
# 2) Use your simulation to generate 25,000 different starting trajectories for the circuit where each sample trajectory is run over the interval [0,5] and sampled every 0.1 seconds. The only difference between these trajectories is a randomly selected initial state, whose components are uniformly distributed in a random manner over [-0.5,0.5].
# 3) From your 25000 trajectories create a numpy array (input_bucket) of shape (25000,6) whose ith slice (i,:) contains the states of the ith trajectory's first 2.8 seconds. Add a slice of white noise whose samples are uniformly distributed between [-.1,.1]. Create a numpy array (target_bucket) of shape (25000,6) whose ith slice (i,:) has the two trajectory points in the noise-free trajectory. Plot one of the 3D trajectories showing the data input samples and associated targets, showing the input in blue.

t_0 = 0
dt = 1e-1
t_final = 5
t = np.arange(t_0, t_final, dt)
num_points = int(t_final/dt)

alpha = 16
beta = 28
m0 = -1.2
m1 = -0.7

def chua(u,t):
    x, y, z = u
```

```

    phi = m1*x+0.5*(m0-m1)*(abs(x+1)-abs(x-1))
    randx = np.random.normal(0,5e-5,3)
    dxdt = alpha*(y-x-phi)
    dydt = x-y+z
    dzdt = -beta*y
    dudt = [dxdt, dydt, dzdt]
    return dudt

delay = 22
target_length = 2

# integrate ode system
num_runs = 25000
input_bucket = np.ndarray((num_runs,num_points-delay,3))
target_bucket = np.ndarray((num_runs,target_length*3))

for i in range(num_runs):
    if i%1000==0:
        print(str(i)+'-',end='')
    # initial conditions
    u0 = .5*np.random.uniform(-1,1,3)
    # integrate ode system
    sol = odeint(chua, u0, t)
    noise = .1*np.random.uniform(-1,1,(sol.shape))
    noise[num_points-target_length:,:]=0
    sol += noise
    input_bucket[i] = sol[:num_points-delay,:]
    tmp_target = sol[num_points-target_length:,:]
    target_bucket[i] = np.reshape(tmp_target,target_length*3)

sample = int(np.random.uniform(0,num_runs,1)[0])

input_sample = input_bucket[sample,:,:]
target_sample = target_bucket[sample,:]
target_sample = np.reshape(target_sample,(2,3))

fig = plt.figure(1)
ax = fig.add_subplot(111, projection='3d')
ax.set_xlabel('x')
ax.set_ylabel('y')
ax.set_zlabel('z')

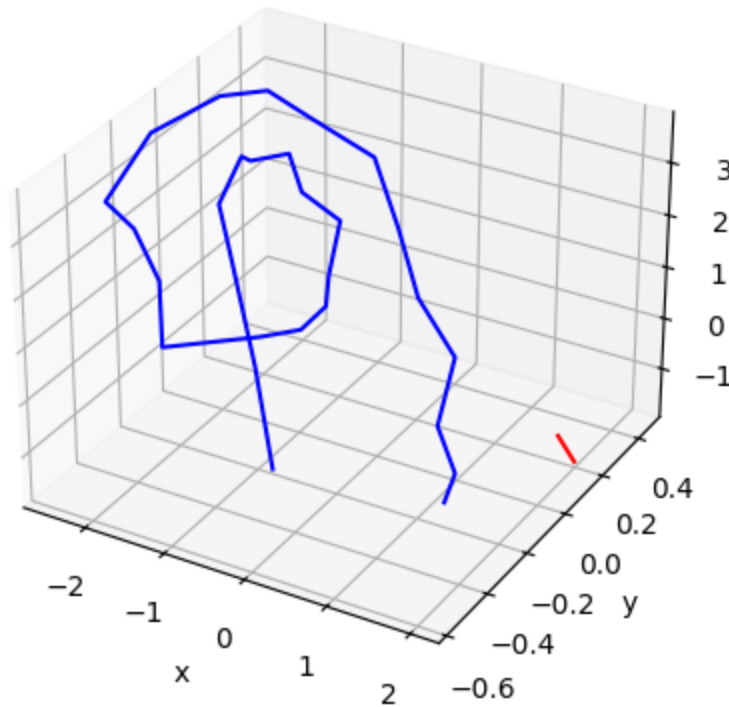
ax.plot(input_sample[:,0],input_sample[:,1],input_sample[:,2],'b')
ax.plot(target_sample[:,0],target_sample[:,1],target_sample[:,2],'r')
tstring = 'input sample = '+str(sample)
ax.set_title(tstring)

```

0-1000-2000-3000-4000-5000-6000-7000-8000-9000-10000-11000-12000-13000-14000
-15000-16000-17000-18000-19000-20000-21000-22000-23000-24000-

Out[5]: Text(0.5, 0.92, 'input sample = 1721')

input sample = 1721



```
In [ ]: #####
#
# TO DO:
# 1) Split the data in input_bucket and target_bucket into training and test
#    Split the training data into a p-training and validation set assuming 2
# 2) Create three tensorflow dataset objects for these data sets assuming a
# 3) Instantiate and train a three layer stack of LSTM networks each with 25
#    using your dataset objects. Show the training curves and compute the tes
#    with the smallest validation loss.
#
# NOTE: training this model takes about 2-3 hours.

num_all_training_samples = int(num_runs*.75)
all_training_inputs = input_bucket[:num_all_training_samples,:,:)
all_training_targets = target_bucket[:num_all_training_samples,:,:)
testing_inputs = input_bucket[num_all_training_samples,:,:)
testing_targets = target_bucket[num_all_training_samples,:,:)

num_ptraining_samples = int(num_all_training_samples*.75)
ptraining_inputs = all_training_inputs[:num_ptraining_samples,:,:)
ptraining_targets = all_training_targets[:num_ptraining_samples,:,:)
validation_inputs = all_training_inputs[num_ptraining_samples,:,:)
validation_targets = all_training_targets[num_ptraining_samples,:,:)

batch_size = 128

ptrain_ds = tf.data.Dataset.from_tensor_slices((ptraining_inputs,ptraining_t
ptrain_ds = ptrain_ds.batch(batch_size)

val_ds = tf.data.Dataset.from_tensor_slices((validation_inputs,validation
```

```

val_ds      = val_ds.batch(batch_size)

test_ds     = tf.data.Dataset.from_tensor_slices((testing_inputs,testing_target))
test_ds     = test_ds.batch(batch_size)

inputs = keras.Input(shape=(num_points-delay, 3))
x = layers.LSTM(256,return_sequences=True)(inputs)
x = layers.LSTM(256,return_sequences=True)(x)
x = layers.LSTM(256)(x)
outputs = layers.Dense(target_length*3)(x)
model = keras.Model(inputs, outputs)

model.compile(optimizer="rmsprop", loss = "mse", metrics=["mae"])

model.summary()

callbacks = [
    keras.callbacks.ModelCheckpoint(
        filepath="HW6-best-model-1.keras",
        save_best_only = True,
        monitor = "val_loss"
    )
]

num_epochs = 150

history = model.fit(ptrain_ds,
                    epochs = num_epochs,
                    validation_data = val_ds,
                    callbacks = callbacks)

best_model = keras.models.load_model("HW6-best-model-1.keras")
test_loss, test_mae = best_model.evaluate(test_ds,verbose=1)
print(f"Test MAE = {test_mae :.2f}")

training_curves(history)

```


Model: "functional_1"


Layer (type)	Output Shape	Par
input_layer_1 (InputLayer)	(None, 28, 3)	
lstm_3 (LSTM)	(None, 28, 256)	266
lstm_4 (LSTM)	(None, 28, 256)	525
lstm_5 (LSTM)	(None, 256)	525
dense_1 (Dense)	(None, 6)	1


Total params: 1,318,406 (5.03 MB)


Trainable params: 1,318,406 (5.03 MB)


Non-trainable params: 0 (0.00 B)


Epoch 1/150
110/110  **31s** 271ms/step - loss: 1.8795 - mae: 1.0015 - val_loss: 1.1634 - val_mae: 0.7213


Epoch 2/150
110/110  **29s** 264ms/step - loss: 1.0739 - mae: 0.6966 - val_loss: 1.2028 - val_mae: 0.6677


Epoch 3/150
110/110  **30s** 275ms/step - loss: 0.8575 - mae: 0.5707 - val_loss: 0.9517 - val_mae: 0.5505


Epoch 4/150
110/110  **30s** 269ms/step - loss: 0.6868 - mae: 0.4843 - val_loss: 0.9089 - val_mae: 0.5227


Epoch 5/150
110/110  **29s** 267ms/step - loss: 0.6381 - mae: 0.4562 - val_loss: 0.7878 - val_mae: 0.4732


Epoch 6/150
110/110  **29s** 268ms/step - loss: 0.5742 - mae: 0.4221 - val_loss: 0.6300 - val_mae: 0.4228


Epoch 7/150
110/110  **28s** 256ms/step - loss: 0.5300 - mae: 0.3976 - val_loss: 0.4594 - val_mae: 0.3417


Epoch 8/150
110/110  **29s** 263ms/step - loss: 0.5024 - mae: 0.3795 - val_loss: 0.5162 - val_mae: 0.3493


Epoch 9/150
110/110  **28s** 253ms/step - loss: 0.4819 - mae: 0.3646 - val_loss: 0.5457 - val_mae: 0.3795


Epoch 10/150
110/110  **28s** 259ms/step - loss: 0.4685 - mae: 0.3533 - val_loss: 0.4252 - val_mae: 0.3256


Epoch 11/150
110/110  **27s** 245ms/step - loss: 0.4297 - mae: 0.3330 - val_loss: 0.3843 - val_mae: 0.3021


Epoch 12/150
110/110  **28s** 251ms/step - loss: 0.4265 - mae: 0.3282 - val_loss: 0.4067 - val_mae: 0.3054


Epoch 13/150
110/110  **27s** 246ms/step - loss: 0.4050 - mae: 0.3147 - val_loss: 0.3747 - val_mae: 0.2832


Epoch 14/150
110/110  **27s** 250ms/step - loss: 0.3901 - mae: 0.3060 - val_loss: 0.3677 - val_mae: 0.2781



















Epoch 15/150
110/110  **27s** 243ms/step - loss: 0.3836 - mae: 0.3019 - val_loss: 0.4047 - val_mae: 0.3000


Epoch 16/150
110/110  **30s** 275ms/step - loss: 0.3726 - mae: 0.2947 - val_loss: 0.3617 - val_mae: 0.2710


Epoch 17/150
110/110  **28s** 251ms/step - loss: 0.3666 - mae: 0.2892 - val_loss: 0.3789 - val_mae: 0.2823


Epoch 18/150
110/110  **28s** 254ms/step - loss: 0.3613 - mae: 0.2864 - val_loss: 0.3173 - val_mae: 0.2544


Epoch 19/150
110/110  **29s** 260ms/step - loss: 0.3482 - mae: 0.2792 - val_loss: 0.3173 - val_mae: 0.2544


al_loss: 0.3224 - val_mae: 0.2580
Epoch 20/150
110/110  **29s** 265ms/step - loss: 0.3410 - mae: 0.2753 - v
al_loss: 0.3036 - val_mae: 0.2432
Epoch 21/150
110/110  **28s** 254ms/step - loss: 0.3284 - mae: 0.2676 - v
al_loss: 0.3185 - val_mae: 0.2493
Epoch 22/150
110/110  **31s** 279ms/step - loss: 0.3323 - mae: 0.2679 - v
al_loss: 0.2716 - val_mae: 0.2277
Epoch 23/150
110/110  **30s** 274ms/step - loss: 0.3135 - mae: 0.2593 - v
al_loss: 0.2871 - val_mae: 0.2397
Epoch 24/150
110/110  **27s** 242ms/step - loss: 0.3108 - mae: 0.2568 - v
al_loss: 0.3423 - val_mae: 0.2524
Epoch 25/150
110/110  **30s** 272ms/step - loss: 0.3068 - mae: 0.2563 - v
al_loss: 0.2887 - val_mae: 0.2414
Epoch 26/150
110/110  **30s** 272ms/step - loss: 0.3005 - mae: 0.2525 - v
al_loss: 0.2873 - val_mae: 0.2369
Epoch 27/150
110/110  **29s** 262ms/step - loss: 0.2892 - mae: 0.2467 - v
al_loss: 0.2469 - val_mae: 0.2182
Epoch 28/150
110/110  **29s** 261ms/step - loss: 0.2744 - mae: 0.2397 - v
al_loss: 0.2446 - val_mae: 0.2176
Epoch 29/150
110/110  **30s** 273ms/step - loss: 0.2727 - mae: 0.2376 - v
al_loss: 0.2457 - val_mae: 0.2128
Epoch 30/150
110/110  **29s** 261ms/step - loss: 0.2699 - mae: 0.2369 - v
al_loss: 0.2905 - val_mae: 0.2477
Epoch 31/150
110/110  **27s** 248ms/step - loss: 0.2621 - mae: 0.2334 - v
al_loss: 0.2395 - val_mae: 0.2189
Epoch 32/150
110/110  **30s** 273ms/step - loss: 0.2477 - mae: 0.2261 - v
al_loss: 0.2732 - val_mae: 0.2269
Epoch 33/150
110/110  **29s** 266ms/step - loss: 0.2362 - mae: 0.2196 - v
al_loss: 0.2630 - val_mae: 0.2267
Epoch 34/150
110/110  **28s** 255ms/step - loss: 0.2341 - mae: 0.2173 - v
al_loss: 0.2955 - val_mae: 0.2369
Epoch 35/150
110/110  **29s** 268ms/step - loss: 0.2273 - mae: 0.2151 - v
al_loss: 0.2582 - val_mae: 0.2094
Epoch 36/150
110/110  **30s** 270ms/step - loss: 0.2261 - mae: 0.2120 - v
al_loss: 0.3106 - val_mae: 0.2369
Epoch 37/150
110/110  **30s** 270ms/step - loss: 0.2131 - mae: 0.2075 - v
al_loss: 0.3608 - val_mae: 0.2567
Epoch 38/150


110/110  **31s** 278ms/step - loss: 0.2169 - mae: 0.2074 - val_loss: 0.2508 - val_mae: 0.2071
Epoch 39/150


110/110  **33s** 303ms/step - loss: 0.2062 - mae: 0.2005 - val_loss: 0.2158 - val_mae: 0.2010
Epoch 40/150


110/110  **31s** 284ms/step - loss: 0.2107 - mae: 0.2035 - val_loss: 0.2616 - val_mae: 0.2329
Epoch 41/150


110/110  **28s** 258ms/step - loss: 0.2042 - mae: 0.1986 - val_loss: 0.3092 - val_mae: 0.2330
Epoch 42/150

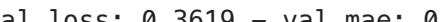
110/110  **29s** 262ms/step - loss: 0.1972 - mae: 0.1954 - val_loss: 0.2765 - val_mae: 0.2205
Epoch 43/150


110/110  **29s** 266ms/step - loss: 0.1910 - mae: 0.1915 - val_loss: 0.2400 - val_mae: 0.2046
Epoch 44/150


110/110  **31s** 282ms/step - loss: 0.1865 - mae: 0.1897 - val_loss: 0.3638 - val_mae: 0.2481
Epoch 45/150


110/110  **30s** 276ms/step - loss: 0.1906 - mae: 0.1903 - val_loss: 0.2420 - val_mae: 0.2045
Epoch 46/150


110/110  **29s** 268ms/step - loss: 0.1814 - mae: 0.1851 - val_loss: 0.3158 - val_mae: 0.2299
Epoch 47/150

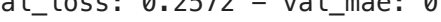
110/110  **31s** 282ms/step - loss: 0.1792 - mae: 0.1833 - val_loss: 0.3619 - val_mae: 0.2507
Epoch 48/150

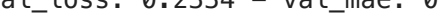
110/110  **29s** 264ms/step - loss: 0.1828 - mae: 0.1839 - val_loss: 0.3450 - val_mae: 0.2413
Epoch 49/150


110/110  **29s** 265ms/step - loss: 0.1652 - mae: 0.1751 - val_loss: 0.2620 - val_mae: 0.2084
Epoch 50/150


110/110  **37s** 337ms/step - loss: 0.1694 - mae: 0.1765 - val_loss: 0.2530 - val_mae: 0.2030
Epoch 51/150


110/110  **30s** 272ms/step - loss: 0.1653 - mae: 0.1727 - val_loss: 0.3032 - val_mae: 0.2217
Epoch 52/150




















110/110  **31s** 285ms/step - loss: 0.1655 - mae: 0.1744 - val_loss: 0.2572 - val_mae: 0.2065
Epoch 53/150

110/110  **31s** 286ms/step - loss: 0.1735 - mae: 0.1757 - val_loss: 0.2334 - val_mae: 0.1932
Epoch 54/150


110/110  **27s** 245ms/step - loss: 0.1522 - mae: 0.1664 - val_loss: 0.2740 - val_mae: 0.2096
Epoch 55/150


110/110  **31s** 285ms/step - loss: 0.1521 - mae: 0.1656 - val_loss: 0.2439 - val_mae: 0.1943
Epoch 56/150


110/110  **29s** 267ms/step - loss: 0.1494 - mae: 0.1640 - val_loss: 0.2731 - val_mae: 0.2063


Epoch 57/150
110/110  **38s** 345ms/step - loss: 0.1584 - mae: 0.1662 - val_loss: 0.2665 - val_mae: 0.1996
Epoch 58/150
110/110  **29s** 265ms/step - loss: 0.1589 - mae: 0.1678 - val_loss: 0.3610 - val_mae: 0.2518
Epoch 59/150
110/110  **28s** 256ms/step - loss: 0.1547 - mae: 0.1663 - val_loss: 0.2489 - val_mae: 0.1954
Epoch 60/150
110/110  **34s** 306ms/step - loss: 0.1568 - mae: 0.1672 - val_loss: 0.3934 - val_mae: 0.2630
Epoch 61/150
110/110  **28s** 259ms/step - loss: 0.1507 - mae: 0.1636 - val_loss: 0.2913 - val_mae: 0.2210
Epoch 62/150
110/110  **29s** 267ms/step - loss: 0.1425 - mae: 0.1582 - val_loss: 0.2850 - val_mae: 0.2244
Epoch 63/150
110/110  **29s** 264ms/step - loss: 0.1383 - mae: 0.1580 - val_loss: 0.4309 - val_mae: 0.2810
Epoch 64/150
110/110  **35s** 259ms/step - loss: 0.1488 - mae: 0.1621 - val_loss: 0.2488 - val_mae: 0.2023
Epoch 65/150
110/110  **25s** 224ms/step - loss: 0.1486 - mae: 0.1617 - val_loss: 0.2394 - val_mae: 0.1965
Epoch 66/150
110/110  **24s** 215ms/step - loss: 0.1357 - mae: 0.1533 - val_loss: 0.2885 - val_mae: 0.2126
Epoch 67/150
110/110  **25s** 231ms/step - loss: 0.1443 - mae: 0.1573 - val_loss: 0.3608 - val_mae: 0.2373
Epoch 68/150
110/110  **24s** 221ms/step - loss: 0.1441 - mae: 0.1580 - val_loss: 0.2254 - val_mae: 0.1942
Epoch 69/150
110/110  **24s** 222ms/step - loss: 0.1353 - mae: 0.1536 - val_loss: 0.3181 - val_mae: 0.2278
Epoch 70/150
110/110  **25s** 224ms/step - loss: 0.1349 - mae: 0.1523 - val_loss: 0.2047 - val_mae: 0.1806
Epoch 71/150
110/110  **25s** 227ms/step - loss: 0.1310 - mae: 0.1497 - val_loss: 0.2490 - val_mae: 0.1961
Epoch 72/150
110/110  **26s** 236ms/step - loss: 0.1244 - mae: 0.1471 - val_loss: 0.1963 - val_mae: 0.1734
Epoch 73/150
110/110  **28s** 250ms/step - loss: 0.1216 - mae: 0.1453 - val_loss: 0.2583 - val_mae: 0.1933
Epoch 74/150
110/110  **26s** 235ms/step - loss: 0.1313 - mae: 0.1498 - val_loss: 0.2326 - val_mae: 0.2032
Epoch 75/150
110/110  **26s** 239ms/step - loss: 0.1299 - mae: 0.1490 - val_loss: 0.2326 - val_mae: 0.2032


```
al_loss: 0.1954 - val_mae: 0.1795
Epoch 76/150
110/110 ██████████ 26s 241ms/step - loss: 0.1262 - mae: 0.1481 - v
al_loss: 0.2266 - val_mae: 0.1907
Epoch 77/150
110/110 ██████████ 27s 242ms/step - loss: 0.1202 - mae: 0.1423 - v
al_loss: 0.2095 - val_mae: 0.1846
Epoch 78/150
110/110 ██████████ 25s 231ms/step - loss: 0.1194 - mae: 0.1431 - v
al_loss: 0.2223 - val_mae: 0.1904
Epoch 79/150
110/110 ██████████ 26s 238ms/step - loss: 0.1155 - mae: 0.1409 - v
al_loss: 0.2078 - val_mae: 0.1871
Epoch 80/150
110/110 ██████████ 28s 256ms/step - loss: 0.1164 - mae: 0.1412 - v
al_loss: 0.1820 - val_mae: 0.1675
Epoch 81/150
110/110 ██████████ 27s 244ms/step - loss: 0.1115 - mae: 0.1386 - v
al_loss: 0.1892 - val_mae: 0.1710
Epoch 82/150
110/110 ██████████ 27s 244ms/step - loss: 0.1093 - mae: 0.1376 - v
al_loss: 0.1852 - val_mae: 0.1749
Epoch 83/150
110/110 ██████████ 28s 258ms/step - loss: 0.1073 - mae: 0.1368 - v
al_loss: 0.2278 - val_mae: 0.1912
Epoch 84/150
110/110 ██████████ 27s 244ms/step - loss: 0.1191 - mae: 0.1411 - v
al_loss: 0.1848 - val_mae: 0.1654
Epoch 85/150
110/110 ██████████ 27s 245ms/step - loss: 0.1060 - mae: 0.1356 - v
al_loss: 0.1669 - val_mae: 0.1621
Epoch 86/150
110/110 ██████████ 27s 247ms/step - loss: 0.1088 - mae: 0.1367 - v
al_loss: 0.2977 - val_mae: 0.2103
Epoch 87/150
110/110 ██████████ 26s 239ms/step - loss: 0.1164 - mae: 0.1413 - v
al_loss: 0.2142 - val_mae: 0.1811
Epoch 88/150
110/110 ██████████ 26s 240ms/step - loss: 0.1036 - mae: 0.1346 - v
al_loss: 0.2883 - val_mae: 0.2129
Epoch 89/150
110/110 ██████████ 27s 242ms/step - loss: 0.1134 - mae: 0.1391 - v
al_loss: 0.2188 - val_mae: 0.1829
Epoch 90/150
110/110 ██████████ 28s 251ms/step - loss: 0.1021 - mae: 0.1332 - v
al_loss: 0.2098 - val_mae: 0.1767
Epoch 91/150
110/110 ██████████ 28s 254ms/step - loss: 0.0982 - mae: 0.1303 - v
al_loss: 0.1853 - val_mae: 0.1620
Epoch 92/150
110/110 ██████████ 26s 239ms/step - loss: 0.1038 - mae: 0.1349 - v
al_loss: 0.2432 - val_mae: 0.1896
Epoch 93/150
110/110 ██████████ 28s 252ms/step - loss: 0.1018 - mae: 0.1331 - v
al_loss: 0.2301 - val_mae: 0.1833
Epoch 94/150
```


110/110  **26s** 237ms/step - loss: 0.1003 - mae: 0.1315 - val_loss: 0.1860 - val_mae: 0.1739
Epoch 95/150


110/110  **27s** 247ms/step - loss: 0.0969 - mae: 0.1323 - val_loss: 0.1819 - val_mae: 0.1636
Epoch 96/150


110/110  **26s** 241ms/step - loss: 0.0992 - mae: 0.1329 - val_loss: 0.2425 - val_mae: 0.1878
Epoch 97/150


110/110  **26s** 238ms/step - loss: 0.1037 - mae: 0.1335 - val_loss: 0.2218 - val_mae: 0.1938
Epoch 98/150


110/110  **26s** 237ms/step - loss: 0.1043 - mae: 0.1322 - val_loss: 0.1911 - val_mae: 0.1689
Epoch 99/150


110/110  **27s** 250ms/step - loss: 0.0995 - mae: 0.1306 - val_loss: 0.1841 - val_mae: 0.1620
Epoch 100/150


110/110  **26s** 235ms/step - loss: 0.0882 - mae: 0.1253 - val_loss: 0.1947 - val_mae: 0.1690
Epoch 101/150


110/110  **26s** 238ms/step - loss: 0.0825 - mae: 0.1216 - val_loss: 0.2179 - val_mae: 0.1831
Epoch 102/150


110/110  **25s** 232ms/step - loss: 0.0841 - mae: 0.1231 - val_loss: 0.2457 - val_mae: 0.1950
Epoch 103/150


110/110  **27s** 242ms/step - loss: 0.0941 - mae: 0.1275 - val_loss: 0.2310 - val_mae: 0.1795
Epoch 104/150


110/110  **27s** 248ms/step - loss: 0.0866 - mae: 0.1233 - val_loss: 0.2444 - val_mae: 0.1870
Epoch 105/150


110/110  **29s** 264ms/step - loss: 0.0926 - mae: 0.1276 - val_loss: 0.1934 - val_mae: 0.1626
Epoch 106/150


110/110  **28s** 255ms/step - loss: 0.0860 - mae: 0.1241 - val_loss: 0.1853 - val_mae: 0.1603
Epoch 107/150


110/110  **27s** 243ms/step - loss: 0.0907 - mae: 0.1262 - val_loss: 0.2163 - val_mae: 0.1733
Epoch 108/150


110/110  **27s** 245ms/step - loss: 0.0881 - mae: 0.1257 - val_loss: 0.2105 - val_mae: 0.1764
Epoch 109/150


110/110  **27s** 245ms/step - loss: 0.0923 - mae: 0.1262 - val_loss: 0.1862 - val_mae: 0.1596
Epoch 110/150


110/110  **28s** 255ms/step - loss: 0.0843 - mae: 0.1228 - val_loss: 0.2496 - val_mae: 0.1979
Epoch 111/150


110/110  **27s** 250ms/step - loss: 0.0798 - mae: 0.1216 - val_loss: 0.2440 - val_mae: 0.1939
Epoch 112/150


110/110  **26s** 240ms/step - loss: 0.0855 - mae: 0.1232 - val_loss: 0.1942 - val_mae: 0.1656


Epoch 113/150
110/110  **28s** 258ms/step - loss: 0.0863 - mae: 0.1228 - val_loss: 0.2203 - val_mae: 0.1796


Epoch 114/150
110/110  **28s** 257ms/step - loss: 0.0859 - mae: 0.1249 - val_loss: 0.1856 - val_mae: 0.1657


Epoch 115/150
110/110  **27s** 247ms/step - loss: 0.0708 - mae: 0.1162 - val_loss: 0.1924 - val_mae: 0.1613


Epoch 116/150
110/110  **26s** 241ms/step - loss: 0.0821 - mae: 0.1215 - val_loss: 0.2071 - val_mae: 0.1747


Epoch 117/150
110/110  **26s** 237ms/step - loss: 0.0748 - mae: 0.1171 - val_loss: 0.2167 - val_mae: 0.1803


Epoch 118/150
110/110  **27s** 247ms/step - loss: 0.0856 - mae: 0.1227 - val_loss: 0.2166 - val_mae: 0.1797


Epoch 119/150
110/110  **26s** 235ms/step - loss: 0.0791 - mae: 0.1199 - val_loss: 0.2296 - val_mae: 0.1831


Epoch 120/150
110/110  **27s** 243ms/step - loss: 0.0770 - mae: 0.1174 - val_loss: 0.1946 - val_mae: 0.1667


Epoch 121/150
110/110  **26s** 239ms/step - loss: 0.0680 - mae: 0.1139 - val_loss: 0.1767 - val_mae: 0.1557


Epoch 122/150
110/110  **28s** 254ms/step - loss: 0.0643 - mae: 0.1111 - val_loss: 0.2000 - val_mae: 0.1723


Epoch 123/150
110/110  **26s** 234ms/step - loss: 0.0636 - mae: 0.1114 - val_loss: 0.2198 - val_mae: 0.1817


Epoch 124/150
110/110  **26s** 238ms/step - loss: 0.0749 - mae: 0.1179 - val_loss: 0.1814 - val_mae: 0.1559


Epoch 125/150
110/110  **26s** 238ms/step - loss: 0.0706 - mae: 0.1148 - val_loss: 0.1830 - val_mae: 0.1563


Epoch 126/150
110/110  **28s** 253ms/step - loss: 0.0661 - mae: 0.1104 - val_loss: 0.1786 - val_mae: 0.1544



















Epoch 127/150
110/110  **26s** 235ms/step - loss: 0.0695 - mae: 0.1136 - val_loss: 0.1740 - val_mae: 0.1533

Epoch 128/150
110/110  **26s** 235ms/step - loss: 0.0605 - mae: 0.1073 - val_loss: 0.2287 - val_mae: 0.1882

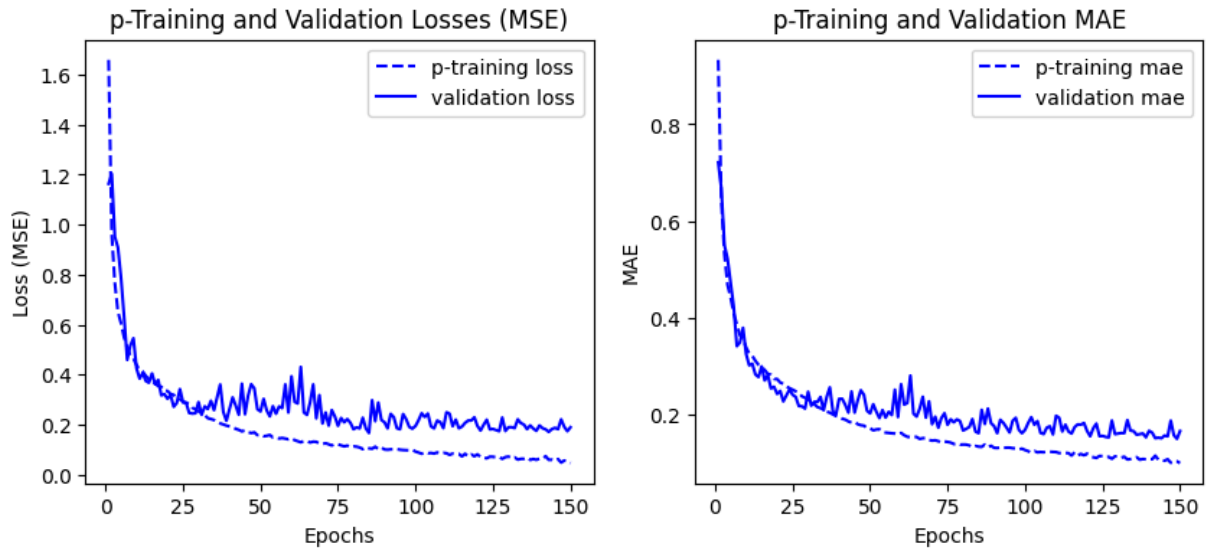
Epoch 129/150
110/110  **25s** 228ms/step - loss: 0.0744 - mae: 0.1173 - val_loss: 0.1749 - val_mae: 0.1601

Epoch 130/150
110/110  **25s** 229ms/step - loss: 0.0651 - mae: 0.1122 - val_loss: 0.1898 - val_mae: 0.1611

Epoch 131/150
110/110  **25s** 230ms/step - loss: 0.0665 - mae: 0.1126 - val_loss: 0.1898 - val_mae: 0.1611

al_loss: 0.1901 - val_mae: 0.1624
Epoch 132/150
110/110  27s 242ms/step - loss: 0.0654 - mae: 0.1119 - v
al_loss: 0.1863 - val_mae: 0.1621
Epoch 133/150
110/110  27s 242ms/step - loss: 0.0589 - mae: 0.1080 - v
al_loss: 0.2212 - val_mae: 0.1890
Epoch 134/150
110/110  29s 260ms/step - loss: 0.0617 - mae: 0.1083 - v
al_loss: 0.2064 - val_mae: 0.1675
Epoch 135/150
110/110  28s 256ms/step - loss: 0.0652 - mae: 0.1109 - v
al_loss: 0.2003 - val_mae: 0.1663
Epoch 136/150
110/110  28s 257ms/step - loss: 0.0660 - mae: 0.1124 - v
al_loss: 0.1822 - val_mae: 0.1589
Epoch 137/150
110/110  27s 250ms/step - loss: 0.0589 - mae: 0.1066 - v
al_loss: 0.1971 - val_mae: 0.1602
Epoch 138/150
110/110  28s 257ms/step - loss: 0.0650 - mae: 0.1106 - v
al_loss: 0.1847 - val_mae: 0.1614
Epoch 139/150
110/110  26s 232ms/step - loss: 0.0567 - mae: 0.1076 - v
al_loss: 0.1753 - val_mae: 0.1536
Epoch 140/150
110/110  26s 239ms/step - loss: 0.0592 - mae: 0.1075 - v
al_loss: 0.1970 - val_mae: 0.1670
Epoch 141/150
110/110  26s 232ms/step - loss: 0.0582 - mae: 0.1057 - v
al_loss: 0.1859 - val_mae: 0.1632
Epoch 142/150
110/110  25s 224ms/step - loss: 0.0765 - mae: 0.1169 - v
al_loss: 0.1814 - val_mae: 0.1518
Epoch 143/150
110/110  25s 226ms/step - loss: 0.0608 - mae: 0.1079 - v
al_loss: 0.1713 - val_mae: 0.1528
Epoch 144/150
110/110  25s 227ms/step - loss: 0.0616 - mae: 0.1077 - v
al_loss: 0.1824 - val_mae: 0.1516
Epoch 145/150
110/110  24s 222ms/step - loss: 0.0563 - mae: 0.1051 - v
al_loss: 0.1873 - val_mae: 0.1571
Epoch 146/150
110/110  25s 228ms/step - loss: 0.0598 - mae: 0.1072 - v
al_loss: 0.1821 - val_mae: 0.1556
Epoch 147/150
110/110  25s 225ms/step - loss: 0.0440 - mae: 0.0981 - v
al_loss: 0.2213 - val_mae: 0.1880
Epoch 148/150
110/110  26s 239ms/step - loss: 0.0605 - mae: 0.1078 - v
al_loss: 0.1916 - val_mae: 0.1579
Epoch 149/150
110/110  25s 225ms/step - loss: 0.0520 - mae: 0.1036 - v
al_loss: 0.1739 - val_mae: 0.1504
Epoch 150/150

110/110 ————— 26s 236ms/step – loss: 0.0413 – mae: 0.0976 – val_loss: 0.1897 – val_mae: 0.1665
 49/49 ————— 5s 96ms/step – loss: 0.1600 – mae: 0.1580
 Test MAE = 0.15



```
In [7]: #####
#
# TO DO:
# 1) For the trained layered model, generate all of the predictions made by
#    test dataset's inputs. Randomly select 10 of these predictions and plot
#    portrait of the input (blue), the actual target (red), and the predicted
#    samples, does your model appear to predict the future behavior of this
#    chaotic system.

best_model = keras.models.load_model("HW6-best-model-1.keras")
test_loss, test_mae = best_model.evaluate(test_ds, verbose=1)
print(f"Test MAE = {test_mae :.2f}")

predictions = best_model.predict(testing_inputs)

num_testing_samples = num_runs - num_all_training_samples
fig = plt.figure(figsize=plt.figaspect(0.5))
for i in range(10):
    sample = int(np.random.uniform(0, num_testing_samples, 1)[0])
    input_sample = testing_inputs[sample, :, :]
    target_sample = testing_targets[sample, :]
    target_sample = np.reshape(target_sample, (2, 3))
    predict_sample = predictions[sample]
    predict_sample = np.reshape(predict_sample, (2, 3))

    ix = i%2
    iy = int(i/2)

    ax = fig.add_subplot(2, 5, i+1, projection='3d')

    ax.set_xlabel('x')
    ax.set_ylabel('y')
    ax.set_zlabel('z')
```

```
ax.plot(input_sample[:,0],input_sample[:,1],input_sample[:,2],'b')
ax.plot(target_sample[:,0],target_sample[:,1],target_sample[:,2],'r')
ax.plot(predict_sample[:,0],predict_sample[:,1],predict_sample[:,2],'g--')
```

49/49 ————— 4s 81ms/step - loss: 0.1454 - mae: 0.1527

Test MAE = 0.15

196/196 ————— 9s 45ms/step

