ESNsummary

October 29, 2025

[25]:

```python
import numpy as np
import pandas as pd
from ESNmodel import EchoStateNetwork, ESNPipeline
from ESNplots import internalStatesHistogram, predictionAnalysis
```

[26]:

```python
from reservoirpy.datasets import mackey_glass
timeseries = mackey_glass(15000)
timeseries.shape
```

[26]:

```
(15000, 1)
```

[27]:

```python
esn_params = {
    'inputSize': 1,
    'reservoirSize': 347,
    'outputSize': 1,
    'spectralRadius': 0.93,
    'inputScaling': 0.39,
    'leakingRate': 0.34,
    'sparsity': 0.77,
    'ridgeParam': 0.25,
    'activation': np.tanh,
    'feedback': False,
    'stateNoise': 0.0,
    'randomSeed': 42
}
```

[28]:

```python
pipeline = ESNPipeline(esn_params=esn_params)
```

[29]:

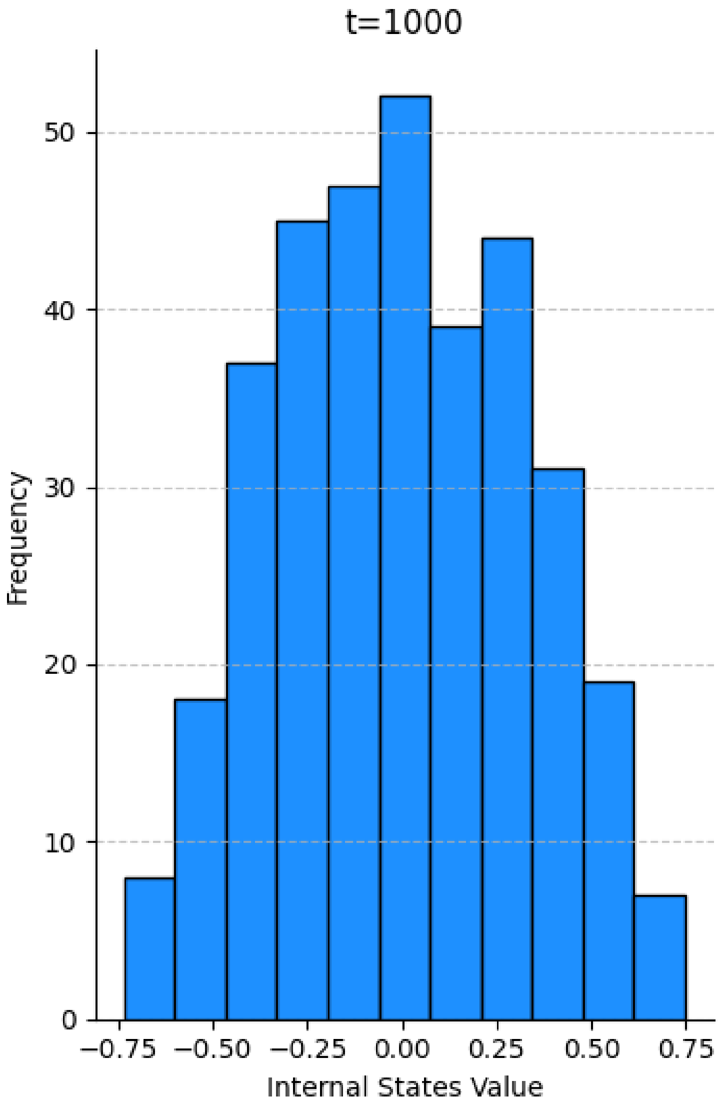
```python
predictions, actuals, metrics, internal_states = pipeline.run(
    timeseries,
    trainRatio=0.7,
    predictionHorizon=1,
    washout_train=100,
    washout_pred=0,
)
```

[30]:

```python
print(pd.DataFrame([metrics]))
```

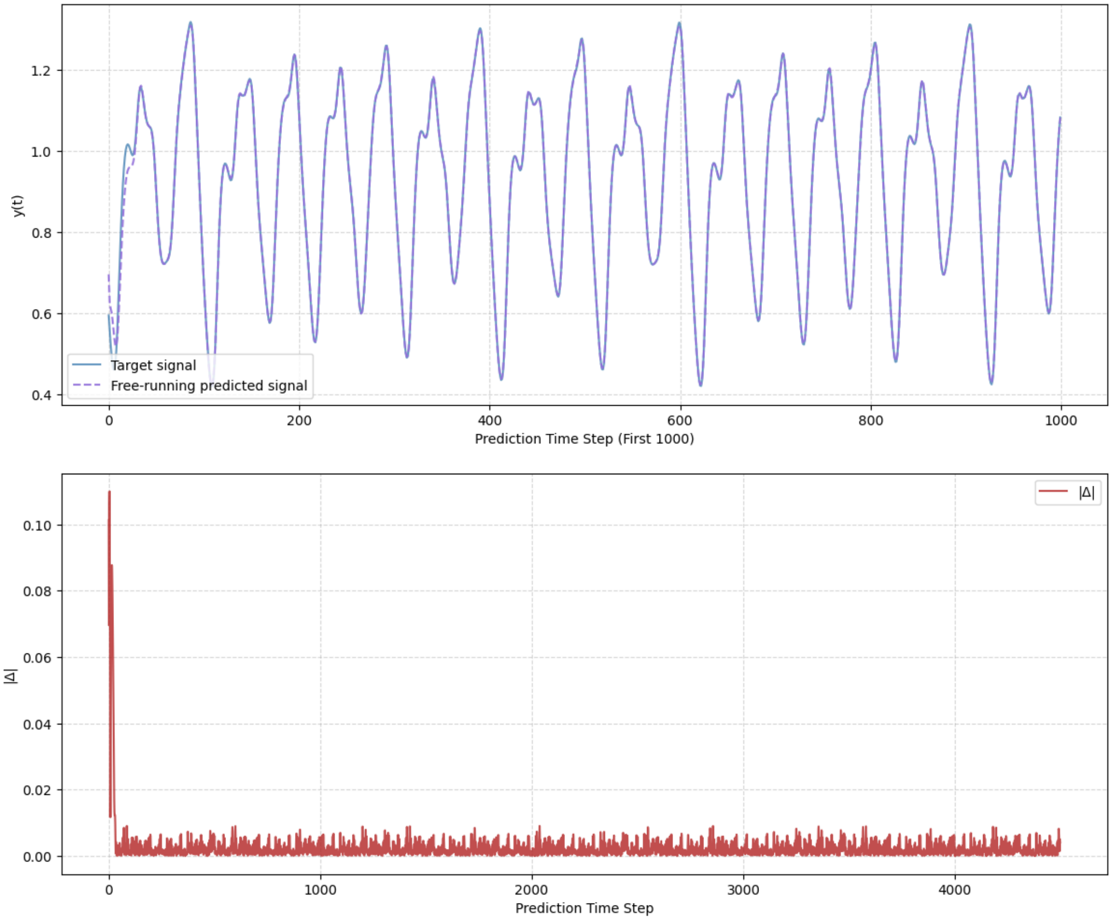
```
        mse      rmse       mae     nrmse
0  0.000039  0.006212  0.002651  0.006904
```

```
[31]: step_to_plot = 1000
      if internal_states is not None and internal_states.shape[1] > step_to_plot :
          internalStatesHistogram(
              states=internal_states,
              time_step_index=step_to_plot,
              reservoir_size=esn_params['reservoirSize'],
          )
```



[32]:

```python
predictionAnalysis(
    predictions=predictions,
    actuals=actuals,
    zoom_limit=1000
)
```



[33]:

```python
from reservoirpy.datasets import henon_map
timeseries = henon_map(15000, a=1.4, b=0.3)
timeseries.shape
```

[33]:

```
(15000, 2)
```

[34]:

```python
N_henon = timeseries.shape[1]
```

[35]:

```python
henon_esn_params = {
    'inputSize': N_henon,
    'reservoirSize': 300,
    'outputSize': N_henon,
    'spectralRadius': 0.9,
    'inputScaling': 0.4,
    'leakingRate': 0.3,
    'sparsity': 0.7,
    'ridgeParam': 1e-7,
    'activation': np.tanh,
    'feedback': False,
    'stateNoise': 0.0,
    'randomSeed': 456
}
```

[36]:

```
pipeline = ESNPipeline(esn_params=henon_esn_params)
predictions, actuals, metrics, internal_states = pipeline.run(
    timeseries,
    trainRatio=0.7,
    predictionHorizon=1,
    washout_train=100,
    washout_pred=0,
)
```

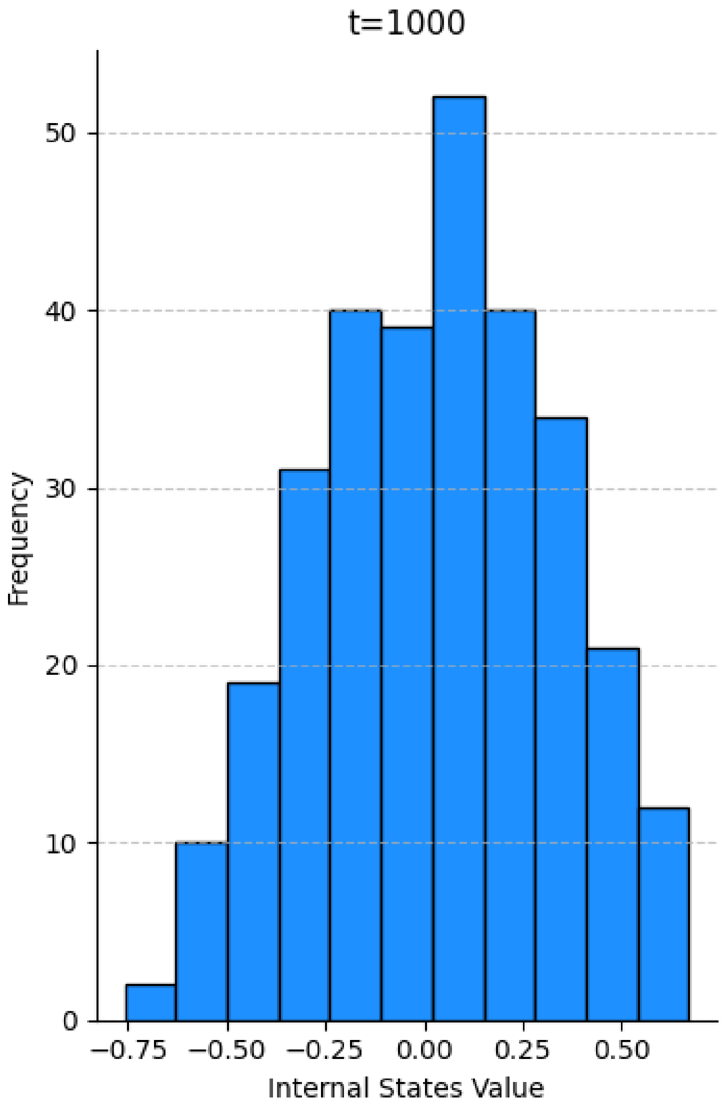
[37]:

```
print(pd.DataFrame([metrics]))
```

```
        mse      rmse       mae     nrmse
0  0.010609  0.102999  0.005019  0.040279
```

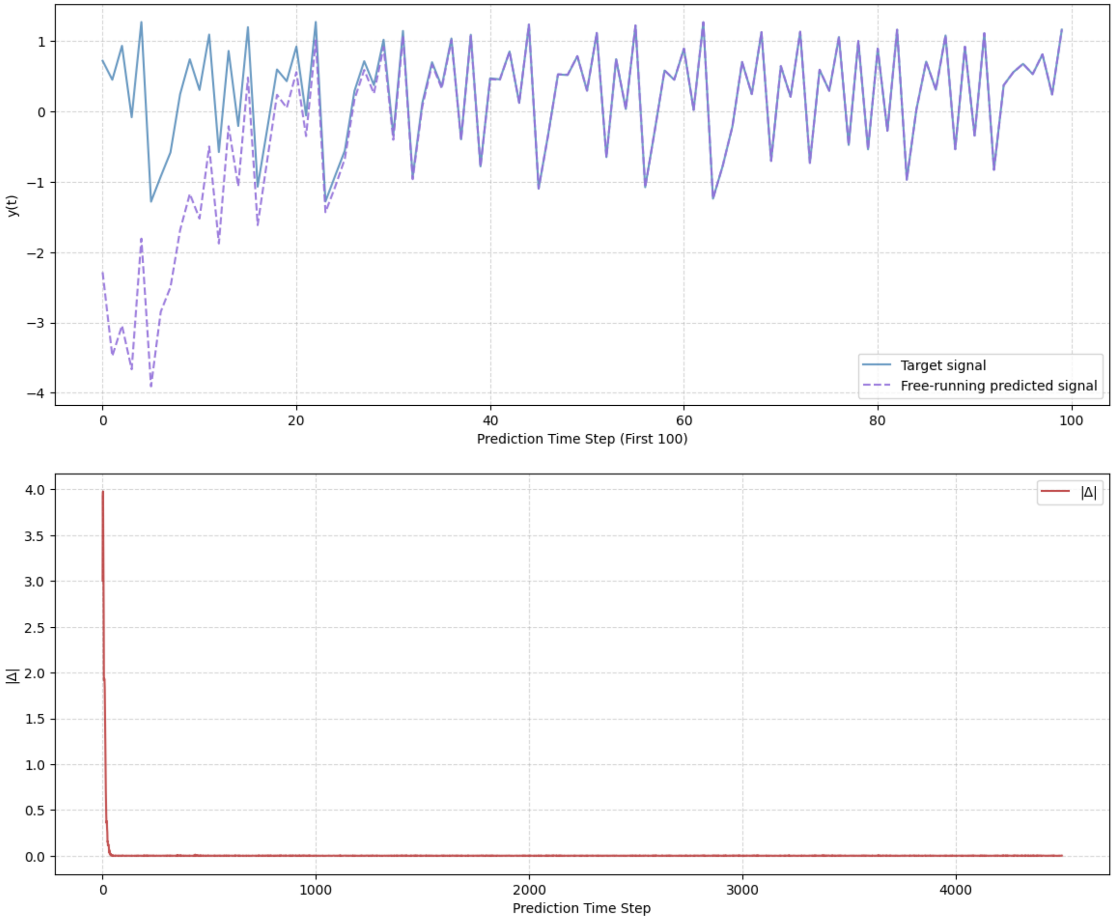
[38]:

```
step_to_plot_henon = 1000
if internal_states is not None and internal_states.shape[1] >␣
 ↪step_to_plot_henon :
    internalStatesHistogram(
        states=internal_states,
        time_step_index=step_to_plot_henon,
        reservoir_size=esn_params['reservoirSize'],
    )
```



[39]:

```
predictionAnalysis(
        predictions=predictions[:, 0],
        actuals=actuals[:, 0],
        zoom_limit=100
    )
```


y(t)
1
0
−1
−2
−3
−4
0
20
40
60
80
100
Prediction Time Step (First 100)
Target signal
Free-running predicted signal
|Δ|
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
0
1000
2000
3000
4000
Prediction Time Step
|Δ|