The code simulates a **microgrid system**, uses a **state-space model** to represent its dynamics, trains an **Artificial Neural Network (ANN)** to predict system behavior, and implements **control mechanisms** (droop control and optical control) to manage the microgrid.

**1. Importing Libraries**

* **numpy**: Used for numerical computations and array manipulations.
* **tensorflow**: Used to build and train the ANN.
* **scipy.signal**: Provides tools for working with state-space models (StateSpace) and simulating system responses (lsim).
* **matplotlib**: Used for plotting and visualizing results.

**2. Defining System Parameters**

* **dt**: The time step for the simulation (0.01 seconds).
* **T**: Total simulation time (10 seconds).
* **time**: A time vector from 0 to 10 seconds with steps of 0.01 seconds.

**3. Defining System Component Parameters**

These parameters define the electrical components of the microgrid:

* **Rf, Lf, Cf**: Resistance, inductance, and capacitance of the filter.
* **Rc, Lc, Cc**: Resistance, inductance, and capacitance of the coupling network.

**4. Defining State-Space Matrices**

* **A**: The state matrix defines how the system's states evolve over time.
* **B**: The input matrix defines how the input affects the system's states.
* **C**: The output matrix defines how the states are mapped to the output.
* **D**: The feedthrough matrix defines how the input directly affects the output (often zero).

These matrices represent the dynamics of the microgrid system.

**5. Creating State-Space System Model**

* This creates a state-space model of the microgrid using the matrices A, B, C, and D.

**6. Generating Input Signal**

* The input signal simulates grid fluctuations. It is a sine wave with a frequency of 0.5 Hz, plus some random noise to simulate real-world variability.

**7. Simulating System Response**

* **lsim**: Simulates the response of the state-space system to the input signal.
* **y**: The output of the system (output voltage).
* **x**: The state variables of the system over time.

**8. Preparing Training Data for ANN**

* **X\_train**: The feature set for training the ANN. It includes:
  + The current state variables (x[:-1, 0] and x[:-1, 1]).
  + The current input signal (input\_signal[:-1]).
* **y\_train**: The target for training the ANN. It is the next state variable (x[1:, 0]).

**9. Defining ANN Model**

* The ANN has:
  + Two hidden layers with 16 neurons each and ReLU activation.
  + An output layer with 1 neuron (to predict the next state variable).

**10. Simulating System Response Without ANN**

* This simulates the system response without using the ANN.

**11. Simulating System Response With ANN Prediction**

* This loop uses the ANN to predict the output voltage at each time step.
* The ANN takes as input:
  + The previous output voltage (or 0 for the first time step).
  + The current input signal.

**12. Compiling and Training ANN**

* The ANN is compiled with the Adam optimizer and Mean Squared Error (MSE) loss.
* It is trained for 100 epochs on the training data.

**13. Droop Control Function**

* This function adjusts the frequency based on the power demand and the ANN's prediction.
* The droop coefficient determines how much the frequency changes with power demand.

**14. Optical Control Function**

* This function adjusts the power demand based on predefined thresholds:
  + If power demand is too high, reduce it.
  + If power demand is too low, increase it.

**15. Simulating Microgrid Operation**

* This loop simulates the microgrid operation over time:
  + Adjusts frequency using droop control.
  + Adjusts power demand using optical control.
  + Stores the results for visualization.

**16. Plotting Results**

* These plots visualize:
  + Frequency over time.
  + Power demand over time.
  + Input signal over time.
  + State variable over time.

**17. ANN Training Performance**

* This plots the training and validation loss over epochs to evaluate the ANN's performance.

**Summary**

The code:

1. Models a microgrid using state-space equations.
2. Simulates its response to an input signal.
3. Trains an ANN to predict system behavior.
4. Implements control mechanisms (droop control and optical control) to manage the microgrid.
5. Visualizes the results to analyze system performance.