Allison Mitchell & Jillian Zitcovitch
ECE 2774: Advanced Power Analysis

Circuit Simulator

Settings Class

The Settings class is used to set global variables for power base and frequency in a power system.

Attributes:

- f (float): The frequency of the power system.
- sbase (float): The power base of the system.

Methods:

- __init__(self, f = 60, sbase = 100): initializes a Settings object with the given parameters; if no frequency provided, defaults to 60 Hz; if no power base provided, defaults to 100 MVA.

Allison Mitchell & Jillian Zitcovitch
ECE 2774: Advanced Power Analysis

Bus Class

The Bus class represents a bus in a power system, with attributes for its name, base voltage, and an index that uniquely identifies each bus instance.

Attributes:

- numBus (class variable): A class-level variable that keeps track of the total number of Bus instances created. It is incremented each time a new Bus object is instantiated.
- name (str): The name of the bus.
- base_kv (float): The base voltage of the bus in kilovolts.
- index (int): A unique index assigned to each bus instance, based on the order of creation.

Methods:

- __init__(self, name: str, base_kv: float): initializes a Bus object with the provided parameters, increments the numBus class variable, and assigns the current value of numBus to the index attribute, ensuring each bus has a unique index.

Allison Mitchell & Jillian Zitcovitch
ECE 2774: Advanced Power Analysis

Conductor Class

The Conductor class represents an electrical conductor with attributes for its physical and electrical properties.

Attributes

- name (str): The name of the conductor.
- diameter (float): The outside diameter of the conductor in inches.
- GMR (float): The geometric mean radius of the conductor at 60 Hz.
- resistance (float): The electrical resistance of the conductor.
- amp (float): The current-carrying capacity (amperage) of the conductor.
- radius (float): The radius of the conductor in inches, calculated by converting the diameter to radius.

Methods

- __init__(self, name: str, diameter: float, GMR: float, resistance: float, amp: float): initializes a Conductor object with the provided parameters and calculates and updates the radius by dividing the diameter by 24 to convert it to inches.

Allison Mitchell & Jillian Zitcovitch
ECE 2774: Advanced Power Analysis

Bundle Class

The `Bundle` class represents a bundle of conductors used in power systems. It includes methods to calculate the equivalent distances for series inductance (DSL) and shunt capacitance (DSC).

Attributes:

- name (str): The name of the bundle.
- num_conductors (float): The number of conductors in the bundle.
- spacing (float): The spacing between conductors.
- conductor (Conductor): An instance of the Conductor class.
- Resistance (float) : The resistance of the Bundle
- DSL (float): The equivalent distance for series inductance
- DSC (float): The equivalent distance for shunt capacitance

Methods:

- __ init __(name: str, num_conductors: float, spacing: float, conductor: Conductor): Initializes Bundle object with given parameters, validates spacing and num_conductor parameters, and calculates the Bundle's resistance, DSL, and DSC. If there is an invalid parameter, it raises an error.

- calculate_DSL(self): Calculates and returns the equivalent distance for series inductance (DSL) based on the number of conductors; if there are more than 4 conductors, an error is raised.

- calculate_DSC(self): Calculates the and returns the equivalent distance for shunt capacitance (DSC) based on the number of conductors. If there are more than 4 conductors, an error is raised.

Allison Mitchell & Jillian Zitcovitch
ECE 2774: Advanced Power Analysis

Geometry Class:

The Geometry class represents the geometric configuration of a transmission line, with methods to calculate the equivalent distance (Deq) based on the coordinates of three points.

Attributes:

- name (str): The name of the geometry.
- xa (float): The x-coordinate of point A.
- ya (float): The y-coordinate of point A.
- xb (float): The x-coordinate of point B.
- yb (float): The y-coordinate of point B.
- xc (float): The x-coordinate of point C.
- yc (float): The y-coordinate of point C.
- Deq (float): The equivalent distance calculated based on the coordinates of points A, B, and C.
- Dab(float): The distance between points A and B.
- Dbc(float): The distance between points B and C.
- Dca(float): The distance between points C and A.

Methods:

- __init__(self, name: str, xa: float, ya: float, xb: float, yb: float, xc: float, yc: float): initializes a Geometry object with the given parameters and calls the calculate_Deq method to find the equivalent distance and updates the variable Deq.

- calculate_Deq(self): calculates and returns the equivalent distance (Deq) based on the distances between the points A, B, and C.

Allison Mitchell & Jillian Zitcovitch
ECE 2774: Advanced Power Analysis

Transmission Line Class

The TransmissionLine class represents a transmission line in a power system, with methods to calculate its series impedance, admittance, Y-matrix, and base values.

Attributes:

- name (str): The name of the transmission line.
- bus1 (Bus): The starting bus of the transmission line.
- bus2 (Bus): The ending bus of the transmission line.
- bundle (Bundle): The bundle of conductors used in the transmission line.
- geometry (Geometry): The geometric configuration of the transmission line.
- length (float): The length of the transmission line.
- R(float): The series resistance of the transmission line.
- X(float): The reactance of the transmission line.
- Z(float): The impedance of the transmission line.
- B(float): The susceptance of the transmission line.
- Y(float): The admittance of the transmission line.
- y_matrix(float): The Y-Matrix for the transmission line.
- matrix(float): The Y-Matrix in Dictionary format for printing.

Methods:

- __init__(self, name: str, bus1: Bus, bus2: Bus, bundle: Bundle, geometry: Geometry, length: float): initializes a TransmissionLine object with the given parameters; calls methods to calculate series impedance, admittance, Y-matrix, and base values.

- calculate_admittance(self): calculates admittance (Y) and susceptance (B) of the transmission line based on the DSC of the bundle.

- calculate_y_matrix(self): calculates Y-matrix (y_matrix) of transmission line, representing admittance between the buses.

- calculate_base_values(self): calculates base impedance (z_base) and admittance (y_base) of transmission line based on base voltage of starting bus and system power base.

- print_yprim(self): prints the Y-matrix of the transmission line in a formatted table using the pandas package.

Allison Mitchell & Jillian Zitcovitch
ECE 2774: Advanced Power Analysis

Transformer Class

The Transformer class represents an electrical transformer with attributes and methods to calculate various electrical parameters.

Attributes:

- name (str): The name of the transformer.
- bus1 (str): The name of the first bus.
- bus2 (str): The name of the second bus.
- power_rating (float): The power rating of the transformer in MVA.
- impedance_percent (float): The impedance of the transformer as a percentage.
- x_over_r_ratio (float): The X/R ratio of the transformer.
- z (complex): The complex impedance of the transformer.
- y (complex): The complex admittance of the transformer.
- g (float): The conductance of the transformer.
- b (float): The susceptance of the transformer.
- r (float): The resistance of the transformer.
- x (float): The reactance of the transformer.
- yprim (list[complex]): The primitive admittance matrix of the transformer.
- matrix (Dict[str, list[complex]]): The dictionary representation of the admittance matrix.

Methods:

- __init__(self, name: str, bus1: str, bus2: str, power_rating: float, impedance_percent: float, x_over_r_ratio: float): Initializes the transformer with the given parameters and calculates initial values for impedance and admittance.

- calc_z(self): Calculates the complex impedance of the transformer.

- calc_y(self): Calculates the complex admittance of the transformer.

- calc_x(self): Calculates the reactance of the transformer.

- calc_r(self): Calculates the resistance of the transformer.

- calc_g(self): Calculates the conductance of the transformer.

- calc_b(self) : Calculates the susceptance of the transformer.

- calc_yprim(self): Calculates the primitive admittance matrix and updates the yprim and matrix attributes.

- print_yprim(self): Prints the admittance matrix in a tabular format using pandas.

Allison Mitchell & Jillian Zitcovitch
ECE 2774: Advanced Power Analysis

Circuit Class

The Circuit class represents an electrical circuit, containing collections of buses, transformers, geometries, conductors, and transmission lines. It provides methods to add these components to the circuit while ensuring no duplicates are added.

Attributes

- name (str): The name of the circuit.
- buses (Dict[str, Bus]): A dictionary to store Bus objects, keyed by their names.
- transformers (Dict[str, Transformer]): A dictionary to store Transformer objects, keyed by their names.
- geometries (Dict[str, Geometry]): A dictionary to store Geometry objects, keyed by their names.
- conductors (Dict[str, Conductor]): A dictionary to store Conductor objects, keyed by their names.
- transmissionlines (Dict[str, TransmissionLine]): A dictionary to store TransmissionLine objects, keyed by their names.

Methods

- __init__(self, name: str): initializes a Circuit object with the provided parameters and initializes empty dictionaries for buses, transformers, geometries, conductors, and transmission lines

- add_bus(self, bus: Bus): adds a Bus object to Circuit object; checks if bus with same name exists and prints error message if so; otherwise, adds the bus to the buses dictionary.

- add_transformer(self, transformer: Transformer): adds a Transformer object to the circuit; checks if a transformer with the same name already exists in the circuit and prints error message if so; otherwise, adds the transformer to the transformers dictionary.

- add_geometry(self, geometry: Geometry): adds a Geometry object to the circuit; checks if a geometry with the same name already exists in the circuit and prints error message if so; otherwise, adds the geometry to the geometries dictionary.

- add_conductor(self, conductor: Conductor): adds a Conductor object to the circuit; checks if a conductor with the same name already exists in the circuit and prints error message if so; otherwise, adds the geometry to the conductors dictionary.

- add_transmissionline(self, transmissionline: TransmissionLine): adds a TransmissionLine object to the circuit; checks if a transmission line with the same name already exists in the circuit and prints error message if so; otherwise, adds the transmission line to the transmissionlines dictionary.

Allison Mitchell & Jillian Zitcovitch
ECE 2774: Advanced Power Analysis

Test

The Test file validates the functionality of the component class files (Transformer, Transmission LIne, Bundle, Geometry, and Bus). Each section of code creates an instance of each component, calls the calculation methods, and prints the results as well as the expected, hand-calculated values for visual inspection and validation.

- The first section, under heading “Transformer Validation”, creates a `Transformer` object and validates its properties and calculations. It checks the transformer's name, bus connections, power rating, impedance percentage, and ratio. It also calculates and prints the transformer's impedance (z) and admittance (y) as well as the expected values for this instance.

- The subsequent section, under heading “Conductor Validation”, creates a `Conductor` object and prints its properties, including name, diameter, geometric mean radius (GMR), resistance, and amperage as well as the expected properties for validation.

- The third section, under heading “Bus Validation”, creates two `Bus` objects and validates their properties, including name, base voltage, and index.

- The next section, under heading “Bundle Validation”, creates a `Bundle` object and prints its properties, including name, number of conductors, spacing, and derived series capacitance (DSC) and shunt inductance (DSL).

- The fifth section, under heading “Geometry Validation”, creates a `Geometry` object and validates its properties, including name and coordinates as well as the equivalent distance (`Deq`).

- The final section, under heading “Transmission Line Validation”, creates a `TransmissionLine` object and validates its properties, including name, length, impedance (Z), admittance (Y), and susceptance (B). It also prints the transmission line's admittance matrix using the included print_yprim method.