Grid Connection Study of Battery Energy Storage System (BESS) to IEEE 30 Bus System using PSSE

1. Objectives

The purpose of this laboratory exercise is to simulate and analyze the behavior of a Battery Energy Storage System (BESS) integrated into a power transmission network using the IEEE 30-bus system as a reference model. Students will gain hands-on experience with:

- Modelling a BESS in a single machine infinite bus (SMIB) configuration.
- Understanding the implications of BESS integration on steady-state power flow.
- Performing dynamic simulations to observe system behavior under disturbances such as faults.
- Automating simulation execution and data extraction using Python scripting.

2. Software and Files Required

To conduct this lab, the following tools and files must be available:

- **Software:** PSSE version 34 or later; Python 3.7+ with libraries psspy, dyntools, matplotlib, and pandas.
- **Files:**
  - IEEE 30-bus .raw (network data) file.
  - IEEE 30-bus .dyr (dynamic data) file.
  - Template scripts for automation.
  - Pre-configured BESS dynamic model files (REGCAU, REECAU, REPCAU).

3. System Description

A 50 MW / 25 MWh BESS is to be integrated into the IEEE 30-bus transmission system. The system represents a simplified model of a real-world utility grid with a diverse mix of generators and loads. The BESS serves both energy storage and grid support functions by injecting or absorbing real and reactive power depending on system conditions.

The BESS is initially modelled in an SMIB setup for verification before being integrated into the IEEE 30-bus system at a suitable location, such as Bus 8.

4. Part A: SMIB Model for BESS

4.1 Create BESS in SMIB

- Develop a minimal system with 5 buses:
  - Bus 31: Generator (BESS)

- Bus 32: MV Bus (33 kV)
- Bus 33: Second MV Bus
- Bus 34: HV Bus (132 kV)
- Bus 35: Infinite Bus (Slack Generator)

- The BESS is connected via transformers to step voltage from 0.6 kV to 132 kV.

4.2 BESS Generator Configuration

- Active Power: 50 MW (Pgen), Pmax: 55 MW
- Reactive Limits: Qmax = +25 Mvar, Qmin = -25 Mvar
- Voltage control mode: remote voltage regulation (at POC)
- Source impedance: Infinite (typical for converter-interfaced sources)
- Add REGCAU, REECAU, and REPCAU dynamic models with reasonable time constants.

4.3 Validate SMIB

- Solve power flow and verify:
  - Voltage at each bus
  - Power flow from BESS
  - Transformer tap settings

5. Part B: IEEE 30-Bus System Preparation

5.1 Import and Prepare Network

- Load the IEEE 30-bus .raw and .dyr files into PSSE.
- Perform a base case power flow to confirm convergence.

5.2 Select and Configure Point of Connection (POC)

- Choose Bus 8 due to proximity to load and accessibility.
- Ensure voltage compatibility (i.e., step down HV BESS to local voltage level if needed).

5.3 Bus Number and Voltage Level Matching

- Confirm no conflict in bus numbers between SMIB and IEEE 30-bus.
- Add a dummy transformer or line to facilitate integration.

5.4 Network Merge

- Integrate the BESS SMIB into the IEEE 30-bus network.
- Use an ideal transformer model to step down from BESS HV (132 kV) to grid level (if different).
- Merge .raw files and regenerate network topology.

6. Part C: Steady-State Analysis

6.1 Solve Power Flow

- Run power flow simulations with and without BESS.
- Observe changes in:
  - Voltage profiles
  - Active/reactive power flows
  - Line loadings

6.2 Document Key Metrics

- Tabulate voltage at all buses.
- Identify buses violating standard voltage limits (typically 0.95-1.05 PU).
- Evaluate transformer and line loadings near the POC.

7. Part D: Dynamic Simulation

7.1 Configure Dynamic Events

- Apply a three-phase fault at Bus 5, clear after 0.1s.
- Monitor system response for a total duration of 5s.

7.2 Measurement Points

- Bus 5: fault location
- Bus 8: BESS connection
- Slack Bus (Bus 1): system reference

7.3 Expected Results

- Voltage and frequency sag and recovery at Bus 8
- Active/Reactive power output of BESS
- Oscillations damping rate

7.4 Plot and Interpret

- Generate plots:
  - Voltage (V vs Time)
  - Frequency (f vs Time)
  - P and Q from BESS (MW/Mvar vs Time)

8. Part E: Automation using Python

8.1 Environment Setup

- Install required packages using pip.

- Set PSSE environment variables in Python.

8.2 Script Execution

- Load network using .sav file
- Apply event using Python interface to PSSE
- Save .out file and extract data using dyntools

8.3 Plot Generation

- Use matplotlib to generate:
    - Time-series plots
    - Event markers (e.g., fault application and clearing)
- Optionally, export plots to PDF

9. Submission Requirements

Prepare the following files and documents for submission:

- PSSE Files: .raw, .dyr, .sav, .snp, .out
- Python Scripts: clearly commented, runnable
- Plots: PNG or PDF format
- Report (PDF) including:
    - System configuration diagrams
    - Simulation parameters and assumptions
    - Results (tables + figures)
    - Analysis and interpretation of findings

Questions:

1. What are the key benefits of integrating a BESS into a transmission grid?
2. How does the BESS contribute to frequency and voltage stability during a fault event?
3. What are the typical limitations of PSSE simulations in modeling converter-interfaced resources?
4. Why is it necessary to match voltage levels when integrating a BESS into an existing grid?
5. Explain how the REGCAU, REECAU, and REPCAU models work together in PSSE to simulate a BESS.
6. What are the advantages of using Python automation in grid simulation studies?
7. How does the placement of the BESS within the IEEE 30-bus system affect the power flow and stability outcomes?
8. What differences did you observe in steady-state power flow before and after integrating the BESS?
9. How would you improve the dynamic model of a BESS for more accurate real-world simulation?
10. What are the critical parameters to consider when evaluating the effectiveness of BESS integration during a disturbance?