**Battery Balancing simulator**

**1. REQUIREMENTS:**

The aim of the assignment is to design and implement a battery balancing algorithm in discharge cycle of the battery and a simulator to test the algorithm.

Simulator must provide these APIs,

* set initial voltages of the cells
* set series and load resistances
* get output voltage and current
* get current sourced by each cell
* get state of the switches

2. The system is divided into three parts:

* Battery Pack
* Simulator
* CLI

2.1 Battery Pack

The battery pack is the main component of the system. A battery pack consists of one or more cells and equal number of switches. A battery pack can be connected with a load and then it will run the circuit and update the cells and switches status.

2.1.1 Cell

The cell is the fundamental element of a battery pack. A cell consists of a voltage source and an internal series resistance. The cell has its own characteristics like discharge curve, capacity and voltage.

2.2 Simulator

The simulator is a component that binds the battery pack with load and provides API to start, get and set various parameters.

2.3 CLI

CLI provides environment to run the simulator. This program implements a command line interpreter that will take user input and perform necessary action.

3.1 Cell

The cell is a voltage source with an internal series resistance. The voltage source has its characteristics discharge curve and capacity.

The cell provides functions to set and get initial voltage, current voltage, series resistance, status of cell’s switch and for the calculating the voltage across the cell.

For calculating the voltage few assumptions are made from the discharge curve of the cell.

Shift1 and shift2 are on the capacity axis representing the value of capacity at that point.

Drop1 and drop2 are the voltage drops in percentage at point shift1 and shift2 respectively.

Voltage is calculated as:

V = Cd\*Slope+ constant.

Where Cd = discharge capacity of cell

3.2 Battery

The Battery resembles a battery pack with one or more cells. This implementation assumes three cells are present in a battery pack. The battery provides output voltage and when connected to a load also the output current.

The design assumes that the output voltage is the voltage of the connected cell that has minimum cell voltage. It also assumes that the internal series resistance are negligible compared to the connected load.

The battery actually implements the balancing algorithm by operating the switches when the battery is connected to a load and running, i.e. closed circuit.

3.3 Simulator

The Simulator connects the Battery pack and the load. And it provides various APIs to operate the battery. It starts, stops, pauses and resumes the simulation. These operations are actually wrapper to the battery APIs.

3.4 CLI

CLI takes user input and gives the command to execute. The command parser takes a valid set of commands and sub commands. It provides a command line prompt for user input.

When user gives any command, the parser matches them with the valid command set and returns the function number if there is a match. According to the function number appropriate actions can be taken.

Commands that are supported are:

* set volt x y z: set the initial voltages for each cells.
* get volt : get the current voltages of the cells
* set resis x y z: set the series resistances for each cells.
* get resis : get the series resistances for each cells.
* set load x : set the load resistance for the battery.
* get load : get the load resistance.
* get switch : gets the status of each switch for the cell.
* get current : gets the source current from the battery.
* start : starts the battery simulation.