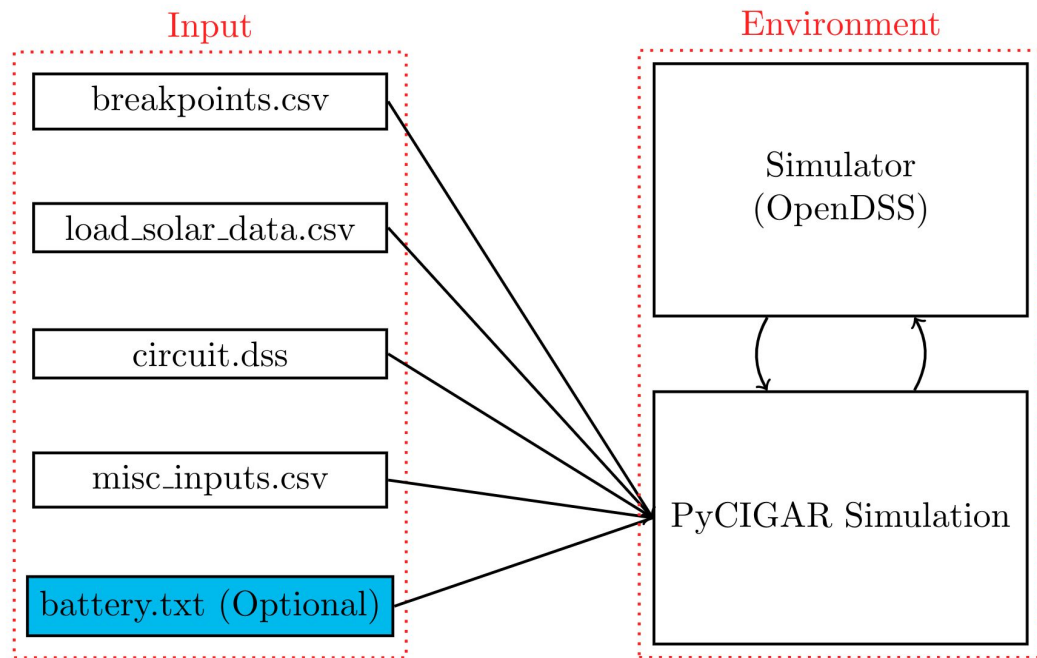


PyCIGAR Battery Storage Device and Control Module

SPADES Workshop

2020.12.02

PyCIGAR Overview



Battery Storage Module - Device Configuration

```
nodes:
- name: s701a
  devices:
  - name: pv_1
    device: pv_device
    custom_device_configs: {default_control_setting: [0.98, 1.01, 1.01, 1.04, 1.06],
                           gain: 1e5,
                           low_pass_filter_measure_mean: 1.2,
                           low_pass_filter_output_mean: 0.115}
    controller: fixed_controller
    custom_controller_configs: {default_control_setting: [0.98, 1.01, 1.01, 1.04, 1.06]}
    # adversary_controller: fixed_controller
    # adversary_custom_controller_configs: {default_control_setting: [1.014, 1.015, 1.015]}
    # hack: [300, 0.4]
  - name: bsd_1
    device: battery_storage_device_advanced
    custom_device_configs: {
      default_control_setting: 'standby',
      total_capacity: 10,
      current_capacity: 9.0,
      max_charge_power: 20.00,
      max_discharge_power: 25.00,
      max_ramp_rate: 0.1,
      max_SOC: 1.0,
      min_SOC: 0.2
    }
}
```

Battery Storage Module - Device and Controller Pairing

```
- name: s702a
  devices:
    - name: bsd_2
      device: battery_storage_device_advanced
      custom_device_configs: {
        default_control_setting: 'standby',
        total_capacity: 10,
        current_capacity: 8.0,
        max_charge_power: 20.00,
        max_discharge_power: 30.00,
        max_ramp_rate: 0.1,
        max_SOC: 1.0,
        min_SOC: 0.2
      }

- name: s703a
  devices:
    - name: bsd_3
      device: battery_storage_device_advanced
      custom_device_configs: {
        default_control_setting: 'standby',
        total_capacity: 10,
        current_capacity: 8.0,
        max_charge_power: 5.00,
        max_discharge_power: 10.00,
        max_ramp_rate: 0.1,
        max_SOC: 1.0,
        min_SOC: 0.2
      }
```

```
controllers:
  - name: psc_1
    controller: battery_peak_shaving_controller_dist
    custom_controller_configs: {
      default_control_setting: 'standby',
      lowpass_filter_frequency: 0.01,
      active_power_target: 750,
      eta: 0.01
    }
    list_devices: ['bsd_1']

  - name: psc_2
    controller: battery_peak_shaving_controller_dist
    custom_controller_configs: {
      default_control_setting: 'standby',
      lowpass_filter_frequency: 0.01,
      active_power_target: 750,
      eta: 0.01
    }
    list_devices: ['bsd_2']

  - name: psc_3
    controller: battery_peak_shaving_controller_dist
    custom_controller_configs: {
      default_control_setting: 'standby',
      lowpass_filter_frequency: 0.01,
      active_power_target: 750,
      eta: 0.01
    }
    list_devices: ['bsd_3']
```

Battery Interface - Utility Text File (battery.txt)

battery_inputs.txt

```
device class=battery_storage_device_advanced name=bsd01 node=s701a default_control_setting=standby min_soc=0.2 max_soc=0.2
total_capacity=10 current_capacity=9 max_charge_power=20 max_discharge_power=25 max_ramp_rate=0.1
```

```
device class=battery_storage_device_advanced node=s702a default_control_setting=standby name=bsd02 min_soc=0.2 max_soc=0.2
total_capacity=10 current_capacity=8 max_charge_power=20 max_discharge_power=30 max_ramp_rate=0.1
```

```
device class=battery_storage_device_advanced node=s703a default_control_setting=standby name=bsd03 min_soc=0.2 max_soc=0.2
total_capacity=10 current_capacity=8 max_charge_power=20 max_discharge_power=30 max_ramp_rate=0.1
```

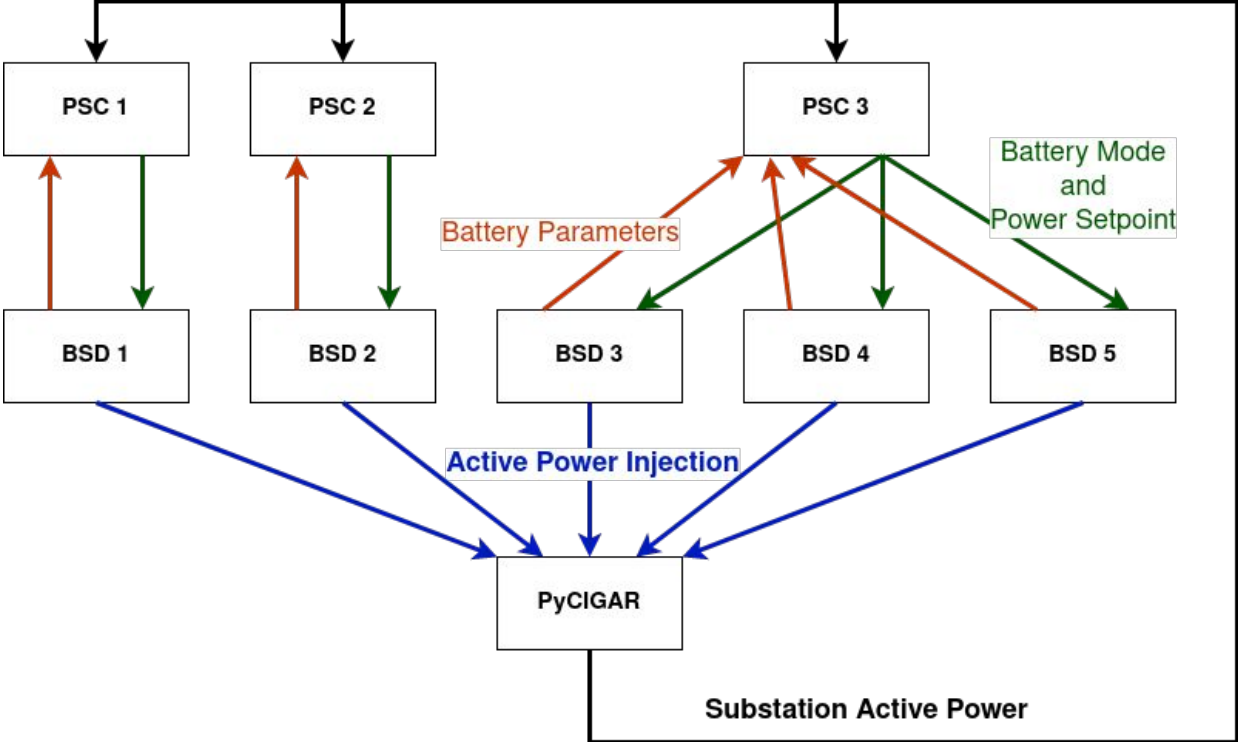
```
controller class=battery_peak_shaving_controller _dist low_pass_filter_freq=0.1 default_control_setting=standby devices=[bsd01]
```

```
controller class=battery_peak_shaving_controller _dist low_pass_filter_freq=0.1 default_control_setting=standby
devices=[bsd02,bsd03]
```

Battery Storage Device and Controller Architecture

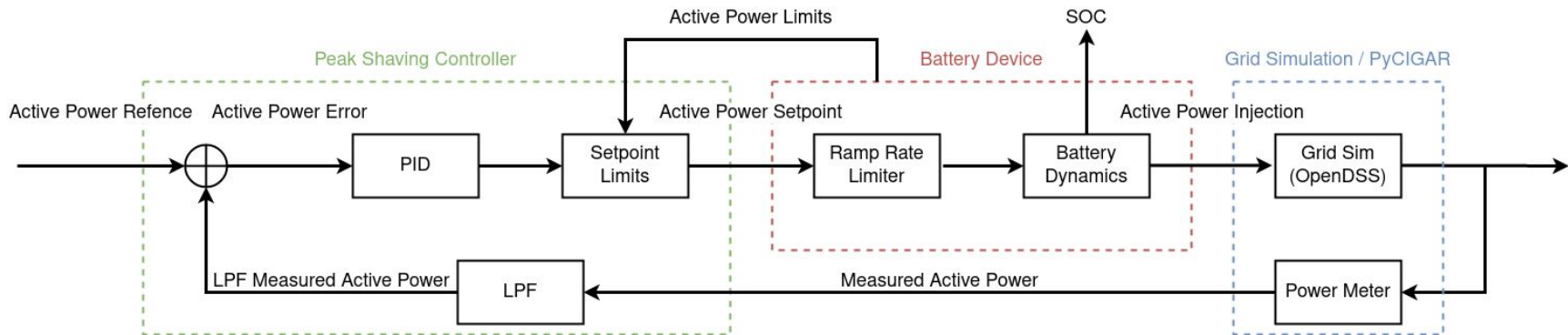
- Controller to battery storage device pairing (BSD)
 - One to one
 - One to many
 - One to one and one to many
- BSD has several internal control logic
 - Volt-Watt
 - Max charge/discharge
 - Minimum or maximum SOC
- Controller can indirectly and directly control BSD
 - Indirectly: set control mode to internal control logic
 - Directly: set mode and setpoint(s)

BSD and Controller Architecture



BSD and Controller Architecture

- Controller obtains BDS parameters
- Controller exercises direct control of battery
 - Passes operation mode (charge/discharge)
 - Passes additional setpoints to battery device (active power setting)
- Battery sdevice has internal logic
 - Ramp rates
 - SOC dynamics

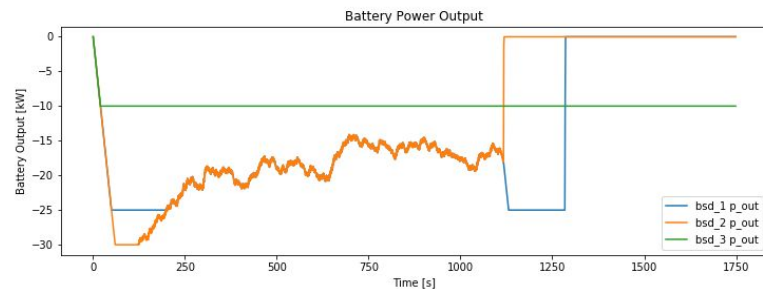
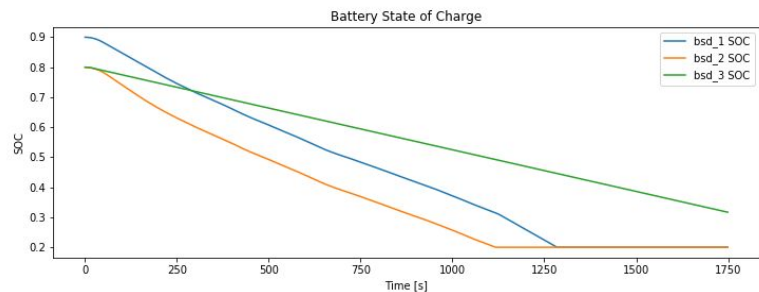
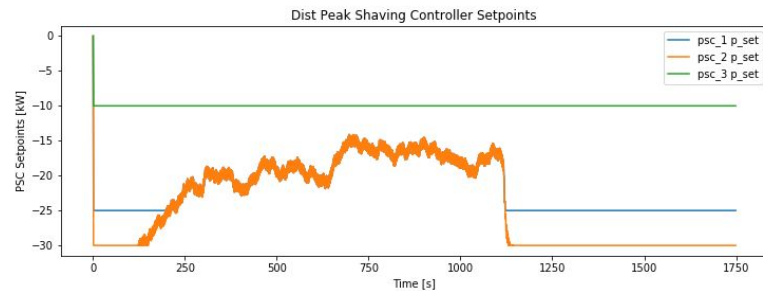
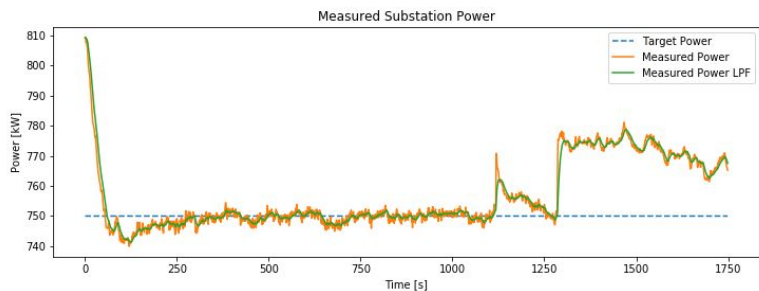


BSD Module - Extensibility

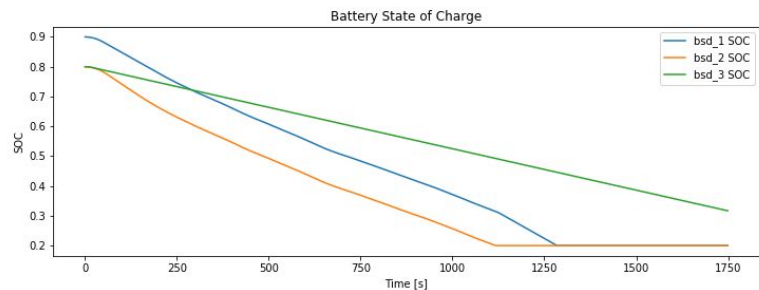
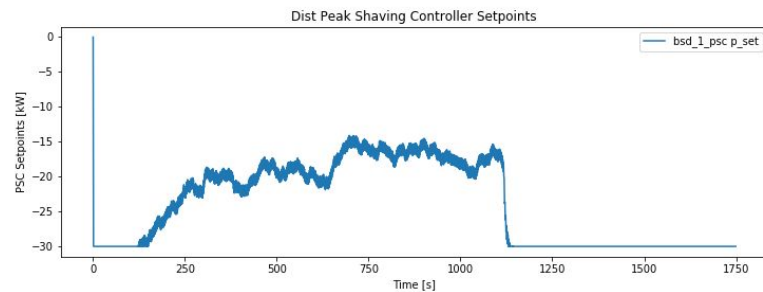
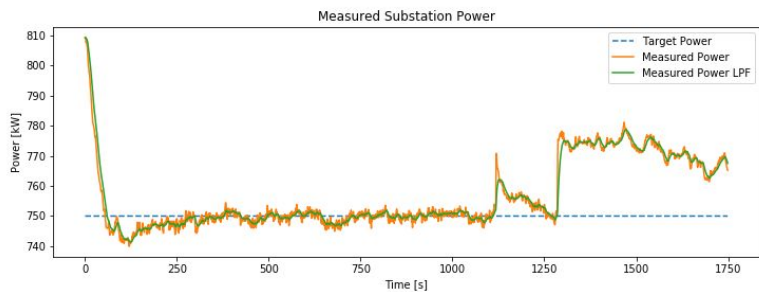
PyCIGAR supports:

- Adding custom battery model, custom battery controller
- Pairing controller - battery/batteries via battery input file
- Controller can control battery mode (charge, discharge or standby) and custom battery settings

Battery Module Simulation - Distributed PSC



Battery Module Simulation - Centralized PSC



Questions?