Overview

Package name is **REoptScenarios**,which currently exports three functions.

-**initialize\_system**: sets up system sizing

-**initialize\_grid\_scenarios:** adds grid scenarios for given parameters

-**initialize\_outage\_events:** adds outage events for given hours and durations

initialize system adds system sizing variables to model and outputs a dictionary of system relevant information. It also initializes the respective technologies and adds their parameters into structs which are added to the system parameter dictionary. For example, params[“pv”].tilt obtains the tilt parameter from the PV struct which is included in the system parameters dictionary.

initialize\_grid\_scenarios takes the system parameters, main system inputs, and an additional file with any scenario specific inputs. If the separate scenario file is not included then a single dispatch scenario is implemented using the values from the main input file. Otherwise, a dispatch scenario is created using the main input file as a basecase and changing the parameters for each scenario input. Dispatch variables and constraints are added to the model. Initialize grid scenarios outputs a named tuple of (costs, system\_state) where costs is all dispatch related costs (mainly grid purchases) and system\_state is a dictionary of state variables such as storage soc.

initialize\_outage\_events dispatches the system for outages given in start\_times, and lasting 1-max\_outage\_duration, with the probability of an outage occurring in a given period given by start\_probs, and the probability of outages lasting various durations is given by outage\_duration\_probs. Note that outage\_duration\_probs is the probability an outage ends in a given duration. Initialize\_outage\_events adds outage variables and constraints to the model and returns a named tuple (costs).

Code Flow

Each technology has an associated file (see example in code structure for how technology files are setup). Each technology file has an associated struct to store parameters and an associated function **setup\_TECH\_inputs** to parse inputs into the struct. Technology sizing decisions go into the **TECH\_system** function, which initializes the sizing variables, capital costs, and the tech struct. Dispatch decisions go into the **TECH\_scenario** function. This function is used for both grid scenario and outage events.

the file **electric\_load.jl** contains functions to initialize the site’s electric load profile. **electric\_tariff.jl** contains functions to initialize the site’s utility tariff, and **financial.jl** contains functions to initialize a struct which contains financial parameters. **energy\_costs.jl** contains additional functions to calculate the total energy cost for the system purchasing electricity from the grid.

Each of these are called in **initialize\_system**, in the **setup\_system.jl** file. This initializes the system parameters and adds system variables + capital costs to the model. **initialize\_grid\_scenarios** in **setup\_grid\_scenario.jl** adds dispatch variables, constraints, and costs to the model as well.

**setup\_outage\_event.jl** contains the **initialize\_outage\_events** function which adds outages dispatch variables, constraints, and costs to the model.