DREAMIN'SG Integrated Simulation Platform

Srijith Balakrishnan

Jul 28, 2021

CONTENTS

CHAPTER

ONE

INTRODUCTION

TThis package is developed as part of the Disaster REsilience Assessment, Modelling, and INno-vation Singapore (DREAMIN'SG) project at the Future Resilient Systems of the Singapore-ETH Centre. The DREAMIN'SG project is funded by the National Research Foundation, Singapore under the Intra-CREATE grant program. In the wake of increasing threats posed by climate change on urban infrastructure [Nissen and Ulbrich (2017)], this project aims at studying the effects of policy interventions and network characteristics on the resilience of urban infrastructure networks. Given the critical nature of the above infrastructure systems, their interdependencies need to be considered in pre- and post-disaster resilience actions.

In specific, the DREAMIN'SG project envisages to build a methodology to assess and predict the resilience of urban infrastructure systems and propose new pathways to develop innovative technologies and services for its improvement. The urban infrastructure system is modeled as an interdependent power-, water-, and transportation network that interact with each other before, during and after a disaster. The researchers are developing an integrated simulation model to study the performance of the interdependent infrastructure network under various disruption and recov-ery scenarios. Based on the simulation-generated datasets, machine learning algorithms would be implemented to understand the causal relationship between topological and policy-related interventions and disaster risks. Building on the understanding of success features that make an urban system resilient, Design Science approaches will be used to develop new solutions that mitigate the consequences of disruptions and accommodate constraints in the analysed case studies. The results of the research will support local governments and system managers to improve infrastructure resilience against weather-related disruptions. The methodology adopted in the study is presented in Fig. 1.

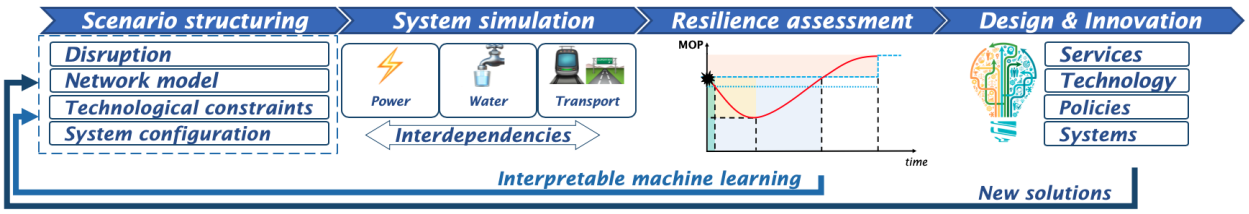

Fig. 1: Figure 1. Methodological framework of DREAMIN'SG project

The steps of the project are summarized as follows:

1. Multiple scenarios are generated by considering different disruptions, network models, technological constraints and system configurations.
2. A simulation model is created for the interdependent power grid, water distribution system, and road transportation system.
3. Resilience is assessed based on the simulated performance of the three systems.
4. An interpretable machine learning algorithm is implemented to analyze the scenarios and extract information related to key system features that influence resilience.
5. The identified system features inform the design of new services, technologies, and prod-ucts that are able to simultaneously enhance resilience and accommodate the technological constraints.

For further information and updates on the project, please visit the DREAMIN'SG webpage. In the rest of the documentation, the details of the interdependent infrastructure simulation platform,including its modeling, installation, and usage are discussed.

CHAPTER

TWO

INTEGRATED SIMULATION PLATFORM

The integrated simulation model has been developed as a Python-based package consisting of modules for simulation of system- and network-level cascading effects resulting from component failures. The overall structure of the integrated simulation platform is illustrated in Figure 2.

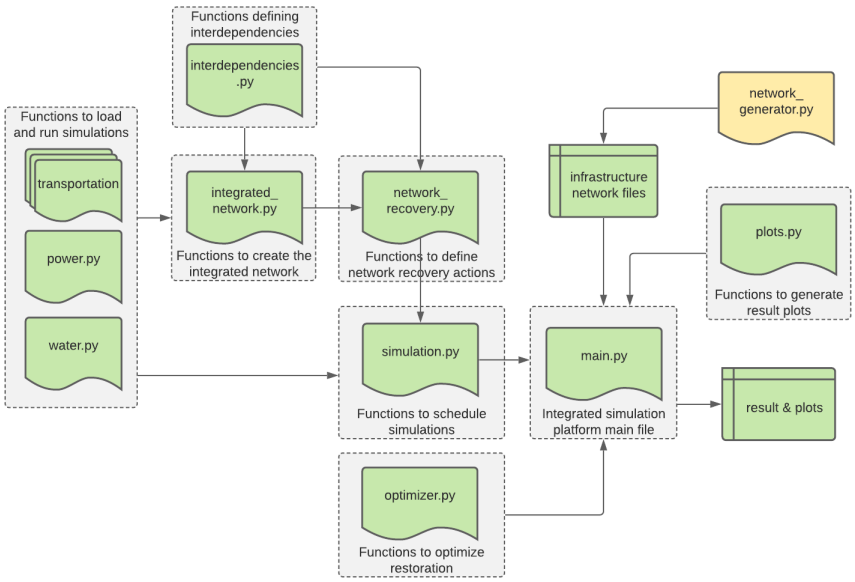

Fig. 1: Figure 2. DREAMIN'SG integrated simulation platform structure

The model is capable of initializing disaster scenarios in interdependent power, water and transportation networks and evaluating resilience strategies by generating operational performance curves (Figure 3). The resilience strategies that can be tested include pre-disaster interventions, such as system redundancy enhancements and post-disaster recovery optimization.

The model is developed by integrating existing flow-based water, power and transportation network models. The whole model can be divided into three broad modules, namely, the integrated infrastructure network module, network recovery module, and the recovery optimization module.

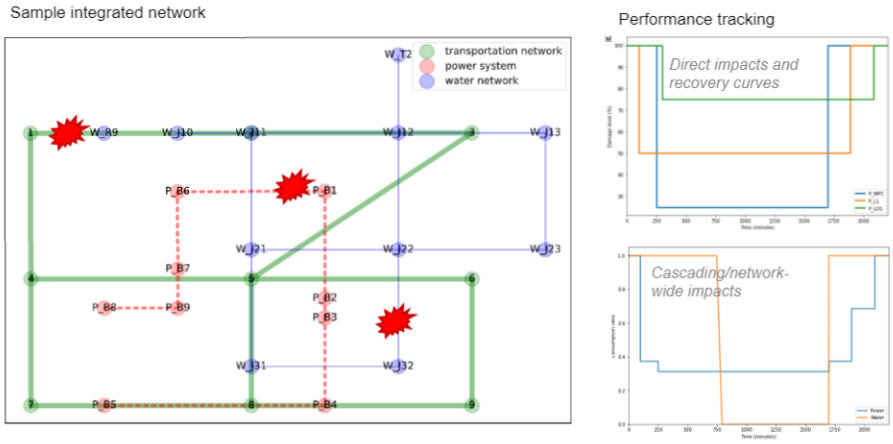

Fig. 2: Figure 3. Implementation of the simulation platform to generate performance curves

2.1 Integrated infrastructure network

This module houses the three infrastructure network models which are used to simulate power-, water- and transportation networks independently. These are developed using existing Python-based packages and the details are presented in Table 2. The module also consists of an interdependency sub-module which serves as an interface between infrastructure network pairs. Currently the following dependencies are considered in the interdependent simulation platform.

a. Power-water dependencies include dependency of water pumps on electric motors and generators on reservoirs.

b. Dependencies also exist between traffic networks and the other two infrastructure models, as the former provides access to the latter, which is critical during the recovery phase. The module also stores the details of the states of all network components, including their operational status after a disaster.

2.2 Network recovery

The recovery module consists of functions to develop an event table to schedule disruptive events and restoration actions after a disaster is initiated in the model. The simulation platform uses this table as a reference to modify the operational status of network components during a simulation, so that the consequences of disaster events and repair actions are reflected while simulating network performance. The recovery module also stores the details such as the number of repair crew for every infrastructure network, their initial locations, etc.

2.3 Recovery optimization

This module determines the order in which the repair actions are carried out. Currently, the approach of the optimization module leverages on the methodology of Model Predictive Control (MPC). In this approach, first, out of *N* repair steps, the solution considering only *k* steps, called the prediction horizon, is computed. Next, the first step of the obtained solution is applied to the system and then the process is being repeated for the remaining *N-1* components until all components have been scheduled for repair. In the context of the integrated infrastructure simulation, the optimizer module evaluates repair sequences of the length of the prediction horizon for each infrastructure (assuming that each of the infrastructure has a separate recovery crew) based on a resilience metric. In the model, the integral loss of service (ILOS) is used as the resilience metric. The ILOS is calculated as follows:

$$ILOS = w_P \sum_k DPower \times dt_P(k) + w_W \sum_k DWater \times dt_w(k) + w_T \sum_k DTransport \times dt_T(k)$$

where, $DPower$, $DWater$ and $DTransport$ are the respective demands not served, $dt_P(k)$, $dt_W(k)$ and $dt_T(k)$ the repair-time specific time steps between the repair actions and wp,ww,wTthe weights. The optimal repair sequence is found by minimizing the ILOS. At this stage the optimal repair action in each prediction horizon is computed using a brute-force approach where the ILOS is evaluated for each of the repair sequences.

Currently, the network data, interdependency data and infrastructure disruption data are to be manually fed into the model to run the network simulations. However, efforts are being made to include separate modules for network generation and hazard initiation in the simulation platform, to enhance the scope of the model. The following improvements will also be made to the model:

a. Realistic policies for network recovery.

b. Additional interdependencies.

CHAPTER

THREE

DATA REQUIREMENTS

CHAPTER

FOUR

INSTALLATION

4.1 Stable release

To install dreaminsg-integrated-model, run this command in your terminal:

```
$ pip install dreaminsg_integrated_model
```

This is the preferred method to install dreaminsg-integrated-model, as it will always install the most recent stable release.

If you don't have pip installed, this Python installation guide can guide you through the process.

4.2 From sources

The sources for dreaminsg-integrated-model can be downloaded from the Github repo.

You can either clone the public repository:

```
$ git clone git://github.com/srijithabalakrishnan/dreaminsg_integrated_model
```

Or download the tarball:

```
$ curl -OJL https://github.com/srijithabalakrishnan/dreaminsg_integrated_model/tarball/
↪master
```

Once you have a copy of the source, you can install it with:

```
$ python setup.py install
```

You may need to create a new Python environment that has all required packages and dependencies installed before start using the package. Run the following comment.

```
$ conda env create --name ENV_NAME --file=environment.yml
```

CHAPTER

FIVE

GETTING STARTED

CHAPTER

SIX

SUPPORT

The model is still in development stage. To report any issues in the model, suggest changes, or obtain support with installation or use, please write to srijith.balakrishnan[at]sec.ethz.ch

CHAPTER

SEVEN

API DOCUMENTATION

7.1 dreaminsg_integrated_model.main module

This is the main module of the integrated infrastructure model where the simulations are performed.

main()
This is the main function that contains the whole simulation workflow.

7.2 dreaminsg_integrated_model.src package

7.2.1 dreaminsg_integrated_model.src.network_recovery

Functions to generate and save disruptive scenarios.

class NetworkRecovery(*network*, *sim_step*)
Bases: object

Generate a disaster and recovery object for storing simulation-related information and settings.

fail_transpo_link(*link_compon*)
Fails the given transportation link by changing the free-flow travel time to a very large value.

Args: link_compon (string): Name of the transportation link.

get_event_table()
Returns the event table.

initiate_next_recov_scheduled()
Flag to identify when the crew must stop at a point and schedule next recovery

reset_networks()
Resets the IntegratedNetwork object within NetworkRecovery object.

restore_transpo_link(*link_compon*)
Restores the disrupted transportation link by changing the free flow travel time to the original value.

Args: link_compon (string): Name of the transportation link.

schedule_recovery(*repair_order*)
Generates the unexpanded event table consisting of disruptions and repair actions.

Parameters **repair_order** (*list of strings.*) – The repair order considered in the current simulation.

set_initial_crew_start(*repair_order*)
Sets the initial start times at which the respective infrastructure crews start from their locations post-disaster.

Parameters **repair_order** (*list of strings.*) – The repair order considered in the current simulation.

update_directly_affected_components(*time_stamp*, *next_sim_time*)
Updates the operational performance of directly impacted infrastructure components by the external event.

Parameters

- **time_stamp** (*integer*) – Current time stamp in the event table in seconds.
- **next_sim_time** (*integer*) – Next time stamp in the event table in seconds.

update_traffic_model()
Updates the static traffic assignment model based on current network conditions.

link_close_event(*wn*, *pipe_name*, *time_stamp*, *state*)
Closes a pipe.

Parameters

- **wn** (*wntr network object*) – Water network object.
- **pipe_name** (*string*) – Name of the pipe.
- **time_stamp** (*integer*) – Time stamp at which the pipe must be closed in seconds.
- **state** (*string*) – The state of the object.

Returns The modified wntr network object after pipe splits.

Return type wntr network object

link_open_event(*wn*, *pipe_name*, *time_stamp*, *state*)
Opens a pipe.

Parameters

- **wn** (*wntr network object*) – Water network object.
- **pipe_name** (*string*) – Name of the pipe.
- **time_stamp** (*integer*) – Time stamp at which the pipe must be opened in seconds.
- **state** (*string*) – The state of the object.

Returns The modified wntr network object after pipe splits.

Return type wntr network object

pipe_leak_node_generator(*network*)
Splits the directly affected pipes to induce leak during simulations.

Parameters

- **wn** (*wntr network object*) – Water network object.
- **disaster_recovery_object** (*DisasterAndRecovery object*) – The object in which all disaster and repair related information are stored.

Returns The modified wntr network object after pipe splits.

Return type wntr network object

7.2.2 dreaminsg_integrated_model.src.network_generator

Functions to generate the infrastructure networks used for simulation.

generate_powern(*file_name*)
Generates a power system network using the pandapower package and saves it to local directory.

Parameters **file_name** (*string*) – Name of the *.json file to be saved including path.

generate_watern(*file_name*)
Generates a water network using the wntr package and saves it to local directory.

Parameters **file_name** (*string*) – Name of the *.inp file to be saved including path.

7.2.3 dreaminsg_integrated_model.src.simulation

Functions to implement the various steps of the interdependent infrastructure network simulations.

class NetworkSimulation(*network_recovery*, *sim_step*)
Bases: object

Methods to perform simulation of interdependent effects.

expand_event_table(*add_points*)
Expands the event table with additional time_stamps for simulation.

Parameters **add_points** (*integer*) – A positive integer denoting the number of extra time-stamps to be added to the simulation.

get_components_repaired()
Returns the list of components that are already repaired.

Returns list of components which are already repaired.

Return type list of strings

get_components_to_repair()
Returns the remaining components to be repaired.

Returns The list of components

Return type list of strings

simulate_interdependent_effects(*network_recovery*)
Simulates the interdependent effect based on the initial disruptions and subsequent repair actions.

Parameters **network_recovery** (*NetworkRecovery object*) – A integrated infrastructure network recovery object.

Returns lists of time stamps and resilience values of power and water supply.

Return type lists

update_repaired_components(*component*)
Update the lists of repaired and to be repaired components.

Parameters **component** (*string*) – The name of the component that was recently repaired.

write_results(*time_tracker*, *power_consump_tracker*, *water_consump_tracker*, *location*, *plotting=False*)
Write the results to local directory.

Parameters

- **time_tracker** (*list of integers*) – List of time stamps.

- **power_consump_tracker** (*list of floats*) – List of corresponding power resilience metric value.
- **water_consump_tracker** (*list of floats*) – List of corresponding water resilience metric value.
- **location** (*string*) – The location to which the results are to be saved.
- **plotting** (*bool, optional*) – True if the plots are to be generated., defaults to False

7.2.4 dreaminsg_integrated_model.src.optimizer

class BruteForceOptimizer(*prediction_horizon=None*)

Bases: *dreaminsg_integrated_model.src.optimizer.Optimizer*

A Brute Force Optimizer class

Parameters **Optimizer** (*Optimizer abstract class.*) – An optimizer class.

find_optimal_recovery(*simulation*)

Identifies the optimal recovery strategy using the Model Predictive Control principle.

Parameters **simulation** (*Simulation object.*) – The infrastructure network simulation object.

get_optimization_log()

Returns the optimization log.

Returns A table consisting of the AUC values from the network simulations.

Return type pandas dataframe.

get_repair_permutations(*simulation*)

Returns all possible permutations of the repair order.

Parameters **simulation** (*Simulation object*) – An integrated infrastructure network simulation object.

Returns A nested list of all possible repair permutations for the given list of components.

Return type list of lists of strings.

get_trackers()

Returns the time, power consumption ratio and water consumption ratio values.

Returns: lists: lists of lists

class Optimizer(*prediction_horizon=None*)

Bases: abc.ABC

The Optimizer class defines an interface to a discrete optimizer or can be implemented as such. This optimizer takes a network object and a prediction horizon and should compute the best steps of the length of the prediction_horizon

7.2.5 dreaminsg_integrated_model.src.resilience_metrics

Resilience metric classes to be used for optimizing recovery actions.

class ResilienceMetric
Bases: abc.ABC

The ResilienceMetric class defines an interface to a resilience metric.

class WeightedResilienceMetric
Bases: dreaminsg_integrated_model.src.resilience_metrics.ResilienceMetric

Methods to calculated the weighted ILOS estimates without normalization.

calculate_power_resmetric(*network_recovery*)
Calcualtes the power resilience metric.

Parameters **network_recovery** (*NetworkRecovery object*) – The network recovery object

Returns Power resilience metric value

Return type float

calculate_water_resmetric(*network_recovery*, *wn_results*)
Calculates and returns the water resilience metric.

Parameters

- **network_recovery** (*NetworkRecovery object*) – The network recovery object
- **wn_results** (*wntr object*) – The water network simulation results for the current time interval

Returns water resilience metric value

Return type float

7.2.6 dreaminsg_integrated_model.src.plots

Functions to generate infrastructure network plots and result plots.

plot_integrated_network(*pn*, *wn*, *tn*, *plotting=False*)
Generates the integrated networkx object.

Parameters

- **pn** (*pandapower network object*) – Power network object.
- **wn** (*wntr network object*) – Water network object.
- **tn** (*STA network object*) – Traffic network object.
- **plotting** (*bool, optional*) – Generates plots, defaults to False.

Returns The integrated infrastructure graph.

Return type networkx object

plot_interdependent_effects(*power_consump_tracker*, *water_consump_tracker*, *time_tracker*, *scatter=True*)
Generates the network-level performance plots.

Parameters

- **power_consump_tracker** (*list of floats*) – A list of power consumption resilience metric values.

- **water_consump_tracker** (*list of floats*) – A list of water consumption resilience metric values.
- **time_tracker** (*list of floats*) – A list of time-stamps from the similation.
- **scatter** (*bool, optional*) – scatter plot, defaults to True

plot_power_net(*net*)
Generates the power systems plot.

Parameters **net** (*pandapower network object.*) – The power systems network.

plot_repair_curves(*disrupt_recovery_object*, *scatter=False*)
Generates the direct impact and repair level plots for the failed components.

Parameters

- **disrupt_recovery_object** (*DisasterAndRecovery object*) – The disrupt_generator.DisruptionAndRecovery object.
- **scatter** (*bool, optional*) – scatter plot, defaults to False

plot_transpo_net(*transpo_folder*)
Generates the transportation network plot.

Parameters **transpo_folder** (*string*) – Location of the .tntp files.

plot_water_net(*wn*)
Generates the water network plot.

Parameters **wn** (*wntr network object.*) – The water network.

7.2.7 dreaminsg_integrated_model.src.network_sim_models.integrated_network

class IntegratedNetwork
Bases: *dreaminsg_integrated_model.src.network_sim_models.integrated_network.Network*

An integrated infrastructure network class

generate_dependency_table(*dependency_file*)
Generates the dependency table from an input file.

Parameters **dependency_file** (*string*) – The location of the dependency file in csv format.

generate_integrated_graph(*plotting=False*, *legend_size=12*, *font_size=8*, *figsize=(10, 7)*, *line_width=2*)
Generates the integrated Networkx object.

Parameters

- **plotting** (*bool, optional*) – Generates plots, defaults to False., defaults to False
- **legend_size** (*int, optional*) – Legend font size, defaults to 12
- **font_size** (*int, optional*) – Text font size, defaults to 8
- **figsize** (*tuple, optional*) – Size of final figure, defaults to (10, 7)
- **line_width** (*int, optional*) – Width of lines, defaults to 2

get_disrupted_components()
Returns the list of disrupted components.

Returns current list of disrupted components.

Return type list of strings

get_disrupted_infra_dict()
Returns the disrupted infrastructure components dictionary.

Returns The disrupted infrastructure components dictionary.

Return type dictionary

get_disruptive_events()
Returns the disruptive event data

Returns: pandas dataframe: The table with details of disrupted components and the respective damage levels.

get_power_crew_loc()
Returns the current power crew location.

Returns Power crew location

Return type string

get_transpo_crew_loc()
Returns the current transportation crew location.

Returns Transportation crew location

Return type string

get_water_crew_loc()
Returns the current water crew location.

Returns Water crew location

Return type string

load_networks(*water_file*, *power_file*, *transp_folder*)
Loads the water, power and transportation networks.

Parameters

- **water_file** (*string*) – The water network file (*.inp).
- **power_file** (*string*) – The power systems file (*.json).
- **transp_folder** (*string*) – The local directory that consists of required transportation network files.

pipe_leak_node_generator()
Splits the directly affected pipes to induce leak during simulations.

reset_crew_locs()
Resets the location of infrastructure crews.

set_disrupted_components(*scenario_file*)
Sets the disrupted components in the network.

Parameters **scenario_file** (*string*) – The location of the disruption scenario file in the list.

set_disrupted_infra_dict()
Sets the disrupted infrastructure components dictionary with infrastructure type as keys.

set_init_crew_locs(*init_power_loc*, *init_water_loc*, *init_transpo_loc*)
Sets the intial location of the infrastructure crews. Assign the locations of the respective offices.

Parameters

- **init_power_loc** (*string*) – Location (node) of the power crew office.

- **init_water_loc** (*string*) – Location (node) of the water crew office.
- **init_transpo_loc** (*string*) – Location (node) of the transportation crew office.

set_power_crew_loc(*power_crew_loc*)
Sets the location of the power crew.

Parameters **power_crew_loc** (*string*) – The name of the location (transportation node)

set_transpo_crew_loc(*transpo_crew_loc*)
Sets the location of the transportation crew.

Parameters **transpo_crew_loc** (*string*) – The name of the location (transportation node)

set_water_crew_loc(*water_crew_loc*)
Sets the location of the water crew.

Parameters **water_crew_loc** (*string*) – The name of the location (transportation node)

class Network
Bases: abc.ABC

This is an abstract class of integrated infrastructure network, defining an interface to other code. This interface needs to be implemented accordingly.

7.2.8 dreaminsg_integrated_model.src.network_sim_models.interdependencies

Classes and functions to manage dependencies in the integrated infrastructure network.

class DependencyTable
Bases: object

A class to store information related to dependencies among power, water and transportation networks.

add_gen_reserv_coupling(*water_id*, *power_id*)
Creates a generator-on-reservoir dependency entry in the dependency table.

Parameters

- **water_id** (*string*) – The name of the reservoir in the water network model.
- **power_id** (*string*) – The name of the generator in the power systems model.

add_pump_motor_coupling(*water_id*, *power_id*)
Creates a pump-on-motor dependency entry in the dependency table.

Parameters

- **water_id** (*string*) – The name of the pump in the water network model.
- **power_id** (*string*) – The name of the motor in the power systems model.

add_transpo_access(*integrated_graph*)
Creates a mapping to nearest road link from every water/power network component.

Parameters **integrated_graph** (*[networkx object]*) – The integrated network as networkx object.

build_power_water_dependencies(*dependency_file*)
Adds the power-water dependency table to the DependencyTable object.

Parameters **dependency_file** (*string*) – The location of the dependency file containing dependency information.

build_transportation_access(*integrated_graph*)
Adds the transportatio naccess table to the DependencyTable object.

Parameters **integrated_graph** (*Nextworkx object*) – The integrated network as Networkx object.

update_dependencies(*network*, *time_stamp*, *next_time_stamp*)
Updates the operational performance of all the dependent components in the integrated network.

Parameters

- **network** (*An IntegratedNetwork object*) – The integrated infrastructure network object.
- **time_stamp** (*integer*) – The start time of the current iteration in seconds.
- **next_time_stamp** (*integer*) – The end tiem of the iteration.

find_connected_power_node(*component*, *pn*)
Finds the bus to which the given power systems component is connected to. For elements which are connected to two buses, the start bus is returned.

Parameters

- **component** (*string*) – Name of the power systems component.
- **pn** (*pandapower network object*) – The power network the origin node belongs to.

Returns Name of the connected bus.

Return type string

find_connected_transpo_node(*component*, *tn*)
Finds the bus to which the given power systems component is connected to. For elements which are connected to two buses, the start bus is returned.

Parameters

- **component** (*string*) – Name of the power systems component.
- **pn** (*pandapower network object*) – The power network the origin node belongs to.

Returns Name of the connected bus.

Return type string

find_connected_water_node(*component*, *wn*)
Finds the water network node to which the water component is connected to.

Parameters

- **component** (*string*) – Name of the water network component.
- **wn** (*wntr network object*) – The water distribution network the origin node belongs to.

Returns Name of the water network node.

Return type string

get_compon_details(*compon_name*)
Fetches the infrastructure type, component type, component code and component actual name.

Parameters **compon_name** (*string*) – Name of the component.

Returns Infrastructure type, component type, component code and component actual name.

Return type list of strings

get_nearest_node(*integrated_graph*, *connected_node*, *target_type*)
Finds the nearest node belonging to a specific family from a given node and the distance between the two.

Parameters

- **integrated_graph** (*netwrokx object*) – The integrated network in networkx format.
- **connected_node** (*string/integer*) – Name of the node for which the nearest node has to be identified.
- **target_type** (*string*) – The type of the target node (power_node, transpo_node, water_node)

Returns Nearest node belonging to target type and the distance in meters.

Return type list

7.2.9 dreaminsg_integrated_model.src.network_sim_models.power.power_system_model

Functions to implement power systems simulations.

get_power_dict()
Creates a dictionary of major power system components in a network. Used for naming automatically generated networks.

Returns Mapping of infrastructure component abbreviations to names.

Return type dictionary of string: dictionary of string: string

load_power_network(*network_json*)
Loads the power system model from a *.json file.

Parameters **network_json** (*string*) – Location of the *.json power system file generated by pandapower package.

Returns The loaded power system model object.

Return type pandapower network object

run_power_simulation(*pn*)
Runs the power flow model for an instance.

Parameters **pn** (*pandapower network object*) – A power system model object generated by pandapower package.

7.2.10 dreaminsg_integrated_model.src.network_sim_models.water.water_network_model

Functions to implement water network simulations.

get_water_dict()
Creates a dictionary of major water distribution system components. Used for naming automatically generated networks.

Returns Mapping of infrastructure component abbreviations to names.

Return type dictionary of string: string

load_water_network(*network_inp*, *initial_sim_step*)
Loads the water network model from an *.inp file.

Parameters

- **network_inp** (*string*) – Location of the *.inp water network file.

- **initial_sim_step** (*integer*) – The initial iteration step size in seconds.

Returns The loaded water wntr network object.

Return type wntr network object

run_water_simulation(*wn*)

Runs the simulation for one time step.

Parameters **wn** (*[type]*) – Water network model object.

Returns Simulation results in pandas tables.

Return type ordered dictionary of string: pandas table

CHAPTER

EIGHT

SEC DISCLAIMER

This research is carried out by Singapore ETHC Centre through its Future Resilient Systems module funded by the National Research Foundation (NRF) Singapore. It is subject to Agency's review and hence is for internal use only. Not the contents necessarily reflect the views of the Agency. Mention of trade names, products, or services does not convey official NRF approval, endorsement, or recommendation.

CHAPTER

NINE

FUNDING STATEMENT

The project is funded by the National Research Foundation Singapore through the Inter-CREATE program.

9.1 Indices and tables

To read the various modules and methods in the package in the alphabetical order, click on Index. To read about the module-wise details, click on Module index.

- genindex
- modindex
- search

PYTHON MODULE INDEX

d