SIMA EXAMPLE P2

Multibody Floating Photovoltaic (FPV) Concept

Valid from Sima version 4.6

Sima Example P2

Multibody Floating Photovoltaic (FPV) Concept

Date: October 2024

Valid from Sima version 4.6

Prepared by: Digital Solutions at DNV

E-mail support: software.support@dnv.com

E-mail sales: digital@dnv.com

1 INTRODUCTION

This example introduces a floating photovoltaic (FPV) array as shown in Figure 1-1 which is loosely based on [1] but does not include the hydrodynamic couplings between the floaters. The modelling is performed by creating a single floater as the base model and then copying the rest of the model using SimaPy library in Python. This example assumes a basic understanding of SimaPy, please refer to Sima example P1 “Getting Started with SimaPy” to learn more about it.

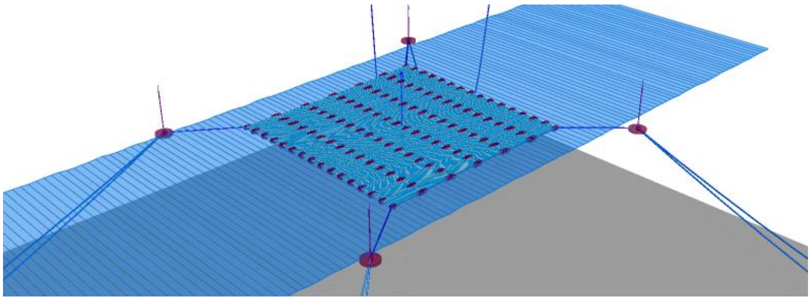


Figure 1-1 The 3D view of the FPV array

This example is intended to show how to utilize SimaPy to modify a base model iteratively. The workflow is divided into these three steps:

1. Preparing the base model (single floater) in Sima

2. Copying and connecting the rest of the array with SimaPy

3. Reimporting the model back to Sima

2 PREPARING THE BASE MODEL (SINGLE FLOATER) IN SIMA

Open Sima and import the *PV1_start.stask* file. You should see two bodies (*body1* and *moor_hub1*), a simple wire coupling, and four fixed elongation couplings) as shown in the Figure 2-1.

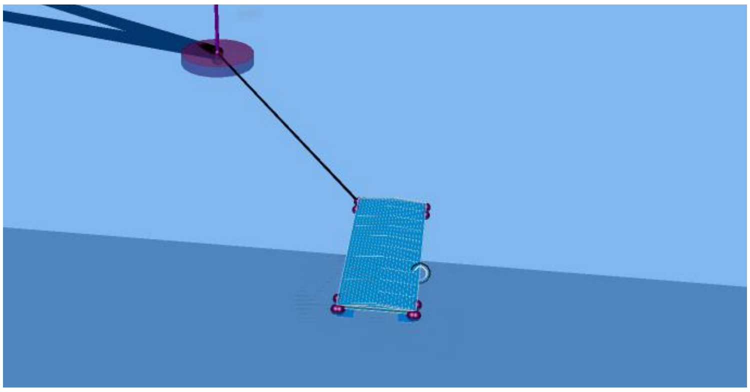


Figure 2-1 The base model

Note that the fixed elongation couplings are not complete, their end point 2 is not yet defined. Using this fixed elongation coupling configuration, we should be able to limit the degree of freedom between floaters.

Export the base model as JSON file. Right-click *PV1_start* task and select *Export*. Then select *SIMA > Json Model Export*. Name the file as “PV1_start.json” and place it in your SimaPy workspace folder.

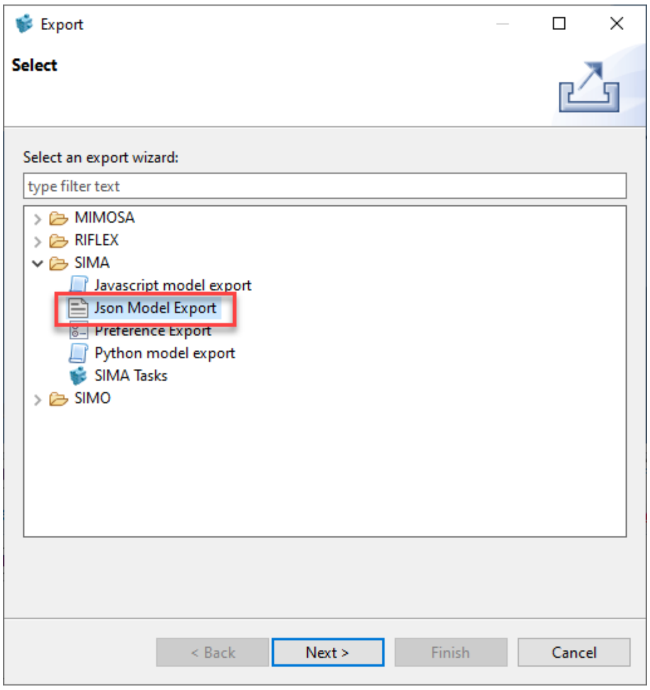


Figure 2-2 Exporting the model as JSON file

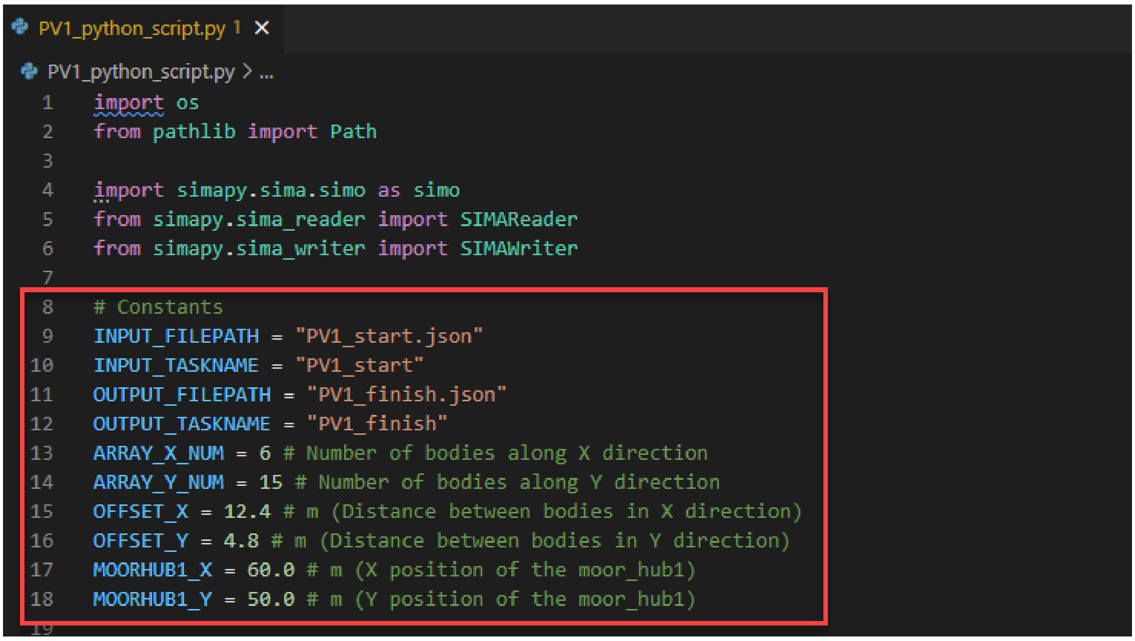
3 COPYING AND CONNECTING THE REST OF THE ARRAY WITH SIMAPY

In this step, the base model will be copied iteratively and connected to each other with the fixed elongation couplings using SimaPy library in Python.

Open the *PV1_python_script.py* in any text editor. This Python script should be self-explanatory with comments explaining each part, but in general, the script is separated into these parts:

- Define some constants (number of floater in the array, distance between floaters, etc.)
- Define some helper functions
- Read *PV1_start.json* from the previous step
- Modify the model (copy floaters, copy connections, copy mooring hubs)
- Write finished model as JSON file

As a user, you would want to modify the constants marked in Figure 3-1. For example, you can specify the file path/names and the number of floaters in each direction.



```python
import os
from pathlib import Path

import simapy.sima.simo as simo
from simapy.sima_reader import SIMAReader
from simapy.sima_writer import SIMAWriter

# Constants
INPUT_FILEPATH = "PV1_start.json"
INPUT_TASKNAME = "PV1_start"
OUTPUT_FILEPATH = "PV1_finish.json"
OUTPUT_TASKNAME = "PV1_finish"
ARRAY_X_NUM = 6 # Number of bodies along X direction
ARRAY_Y_NUM = 15 # Number of bodies along Y direction
OFFSET_X = 12.4 # m (Distance between bodies in X direction)
OFFSET_Y = 4.8 # m (Distance between bodies in Y direction)
MOORHUB1_X = 60.0 # m (X position of the moor_hub1)
MOORHUB1_Y = 50.0 # m (Y position of the moor_hub1)
```

Figure 3-1 Constants in the Python script

Run the Python script and verify if the JSON file is outputted successfully. If you did not modify the script, you should get a file named *PV1_finish.json*.

4 REIMPORTING THE MODEL BACK TO SIMA

Go back to Sima and import the *PV1_finish.json* file. From the menu bar, select File > Import. Then, select SIMA > Json Model Import. Browse the *PV1_finish.json* file from the previous step.

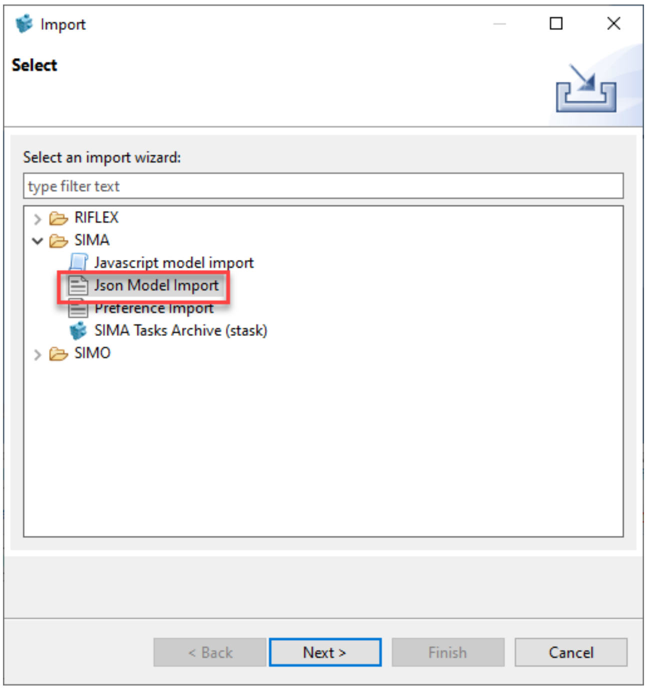


Figure 4-1 Importing a JSON file

When done correctly, a new *PV1_finish* task should be imported. Open it in the 3D view and you will see the completed FPV array as shown in Figure 4-2.

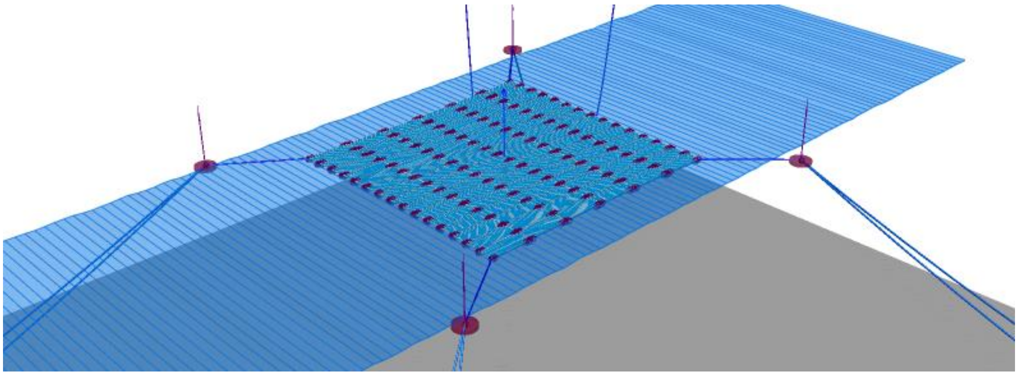


Figure 4-2 Completed FPV array

You can then run the analysis and check the results. You can also modify some constants in the Python script to create other configurations, for example, by changing the number of floaters.

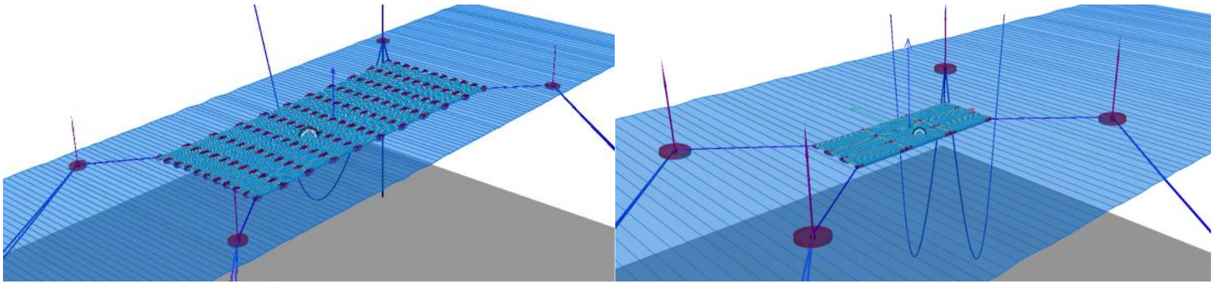


Figure 4-3 Other array configurations

5 REFERENCES

1. Zhang, F., Shi, W. and Wang, Q. (2023) 'A study on the hydrodynamics and coupling effects of the multibody floating photovoltaic (FPV) concept,' *Journal of Marine Science and Engineering*, 11(8), p. 1491. https://doi.org/10.3390/jmse11081491.

About DNV

DNV is an independent assurance and risk management provider, operating in more than 100 countries. Through its broad experience and deep expertise DNV advances safety and sustainable performance, sets industry standards, and inspires and invents solutions.

Digital Solutions

DNV is a world-leading provider of digital solutions and software applications with focus on the energy, maritime and healthcare markets. Our solutions are used worldwide to manage risk and performance for wind turbines, electric grids, pipelines, processing plants, offshore structures, ships, and more. Supported by our domain knowledge and Veracity assurance platform, we enable companies to digitize and manage business critical activities in a sustainable, cost-efficient, safe and secure way.

DNV | digital@dnv.com | dnv.com/digital