NetSim Multi-Parameter Sweeper MATLAB Script

Software: NetSim v14.4 (64 bit), DOT NET CORE SDK 3.1, MATLAB (2022 or higher)

Project Download Link:

https://github.com/NetSim-TETCOS/Multiparameter-Sweeper-MATLAB-v14.4/archive/refs/heads/main.zip

Introduction:

NetSim multi-parameter sweeper automates the network simulation process by running NetSim through the command line interface (CLI). Instead of manually adjusting parameters and analyzing results in the NetSim GUI.

This program simplifies the task of sweeping through multiple input parameters. Users define the parameters to be varied, and the sweeper automates the process of adjusting values, running simulations, saving results, and organizing data. The goal is to save time and effort in the exploration of different parameter sets. The results are conveniently stored and can be easily compared using spreadsheet software like Microsoft Excel.

File Organization:

The project directory consists of several binaries which are responsible for different tasks during a multi-parameter sweep:
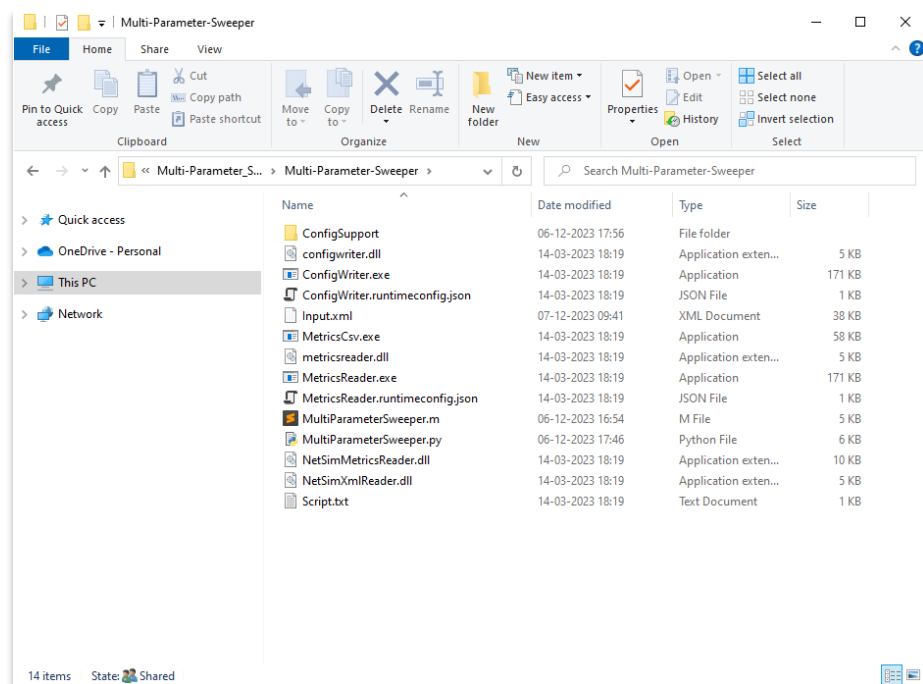

Figure 1: Project directory consists of several binaries

1. **input.xml:** This file contains the base NetSim network configuration that is to be simulated. This file is created by copy pasting the Configuration.netsim file that can be obtained by saving a network configuration in NetSim and renaming it to input.xml.

   Note: Copy the "Configsupport" folder along with the "input.xml" from the saved NetSim network.
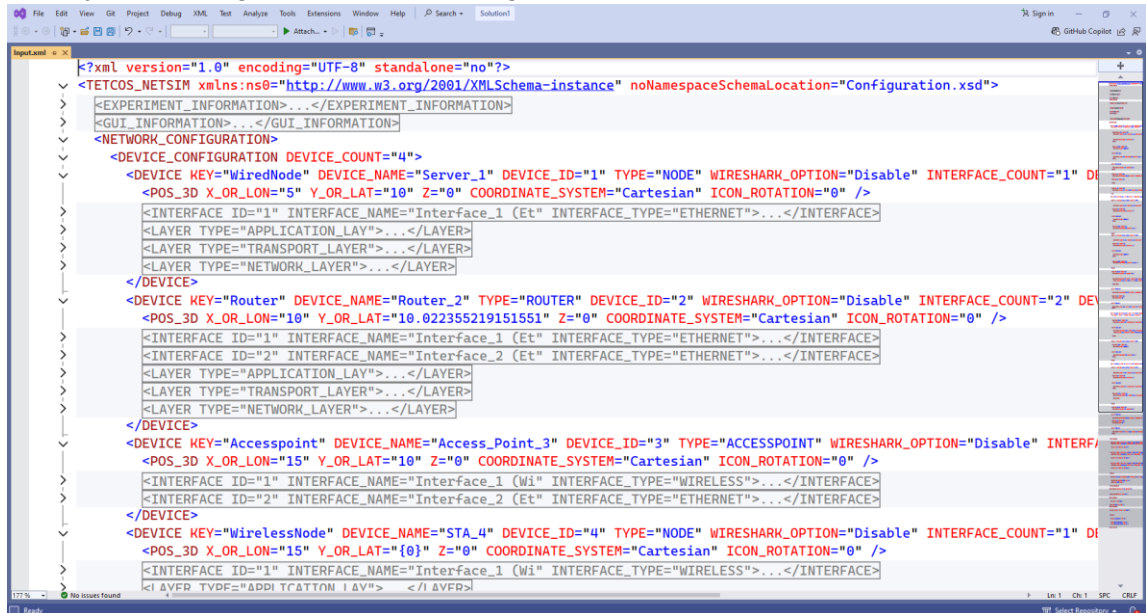

Figure 2: Input.xml file (NetSim Configuration file with the parameters to sweep)

- The values of parameters which are to be varied during each simulation run needs to be specified as {0}, {1}, {2}, etc. respectively.

- For Example, if the Y coordinates of a device is to be varied the values can be modified In the input.xml file as shown below:
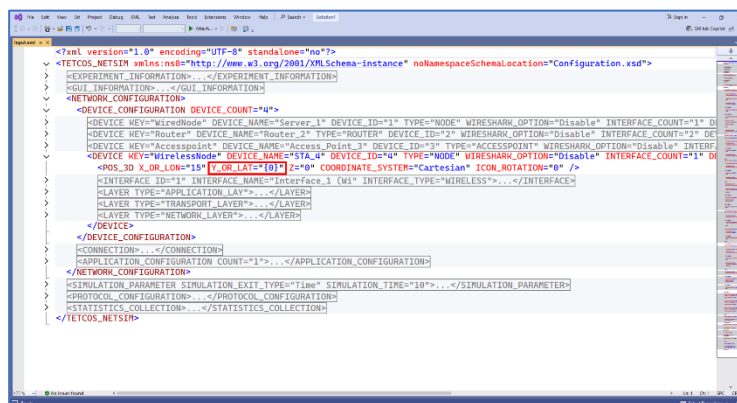

Figure 3: Modify X and Y coordinates in input input.xml file

2. **Script.txt:** This file should be updated with the parameter from the output metrics of NetSim that is to be logged at the end of each simulation run for the purpose of analysis.

At the end of every simulation, NetSim generates a Metrics.xml file which contain the performance metrics written in a specific format based on which it is loaded in the results dashboard.

Each Metric is part of a results table which can be accessed using a menu in the results dashboard. A NetSim Metrics.xml file is shown below:
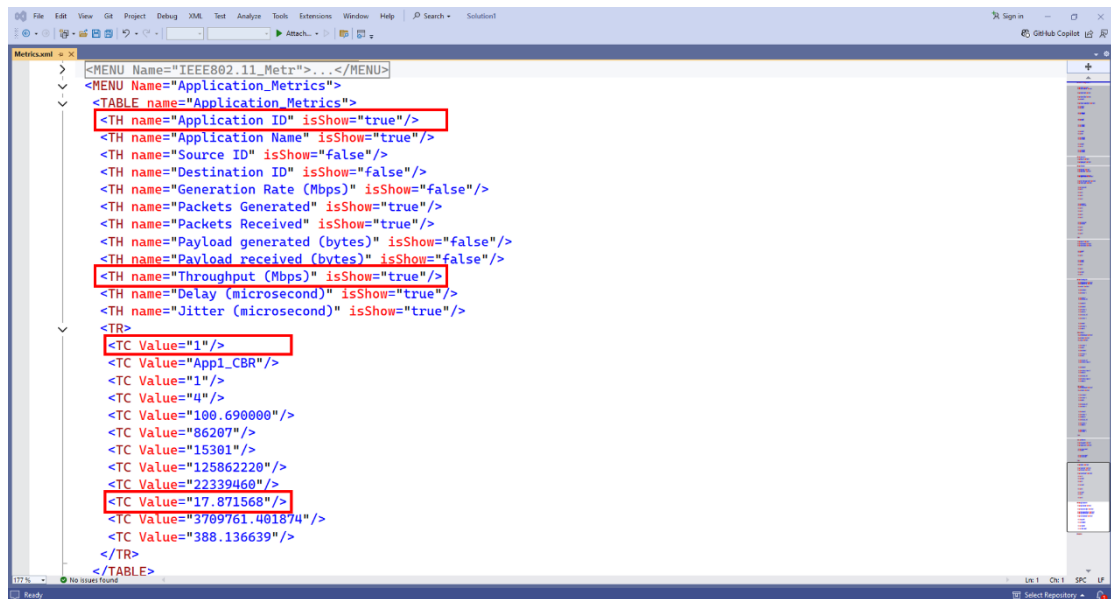

Figure 4: NetSim output Metrics.xml file

For Example, if the application throughput is to be logged for each simulation run then the script file can be updated as shown below:
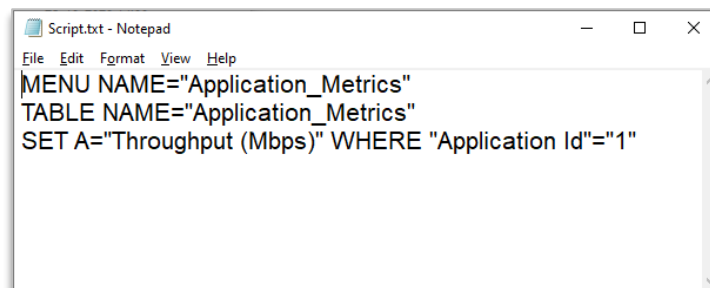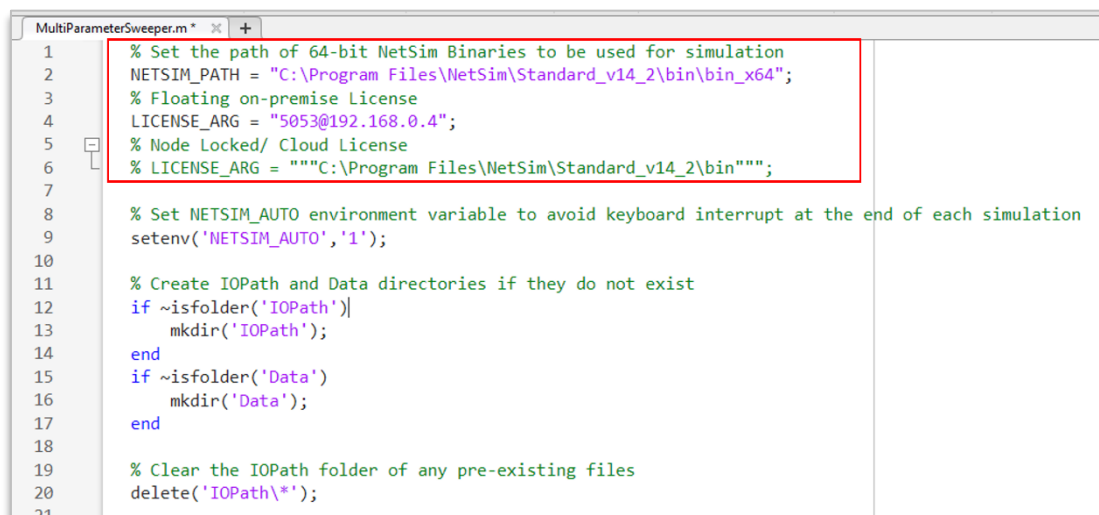

Figure 5: The application throughput is to be logged for each simulation modified in Script.txt

3. **ConfigWriter.exe**: This executable takes one or more command line arguments as input and generated Configuration.netsim file by replacing the arguments in place of the variable parameters specified in the input.xml file.

    If there are two variable parameters specified in the input.xml file ({0} and {1}) then two arguments need to be passed while calling ConfigWriter.exe.

4. **MetricsCSV.exe:** This executable is used to convert the Metrics.xml file present in the output folder into a comma separated file, MetricsPrint.csv.

5. **MetricsReader.exe**: This executable is responsible for reading the output parameter from the Metrics.xml file generated after each simulation and logging it to the results file.

   It uses the Script.txt file to determine which parameter to read from the Metrics file.

   If multiple parameters are to be read and logged, then the MetricsReader.exe can be called multiple times with Script.txt file having information about the parameter to be read each time.

6. **Supporting DLL's**: Some the supporting files such as ConfigWriter.dll, MetricsReader.dll, NetSimMetricsReader.dll, NetSimXmlReader.dll, etc. which are present in the project folder are used by other executable such as ConfigWriter.exe and MetricsReader.exe for various purposes during a multi-parameter sweep.

7. **MultiParameterSweeper.m** uses matlab programming language which is less complex and offers more flexibility as the number of input and output parameters increases.

   - Users can also write the script to run the multi-parameter sweep process in a preferred programming language as per the convenience.
   - The script can be configured to run multiple simulation iterations based on the number of parameters to be varied and the range of values of each parameter.
   - NETSIM_PATH variable can be set to the path of NetSim 64-bit binaries (bin_x64) in the install directory or workspace which is to be used to run Simulations.
   - LICENSE_ARG variable can be set to License server port and IP details in case of floating on premise licenses or the path of license file in case of node locked or cloud licenses.

```
1   % Set the path of 64-bit NetSim Binaries to be used for simulation
2   NETSIM_PATH = "C:\Program Files\NetSim\Standard_v14_2\bin\bin_x64";
3   % Floating on-premise License
4   LICENSE_ARG = "5053@192.168.0.4";
5   % Node Locked/ Cloud License
6   % LICENSE_ARG = """C:\Program Files\NetSim\Standard_v14_2\bin""";
7
8   % Set NETSIM_AUTO environment variable to avoid keyboard interrupt at the end of each simulation
9   setenv('NETSIM_AUTO','1');
10
11  % Create IOPath and Data directories if they do not exist
12  if ~isfolder('IOPath')
13      mkdir('IOPath');
14  end
15  if ~isfolder('Data')
16      mkdir('Data');
17  end
18
19  % Clear the IOPath folder of any pre-existing files
20  delete('IOPath\*');
```

Figure 6: Set the NetSim Path where `NetSimCore.exe` is located.

For example,

NETSIM_PATH= "C:\\Program Files\\NetSim\\Standard_v14_4\\bin\\bin_x64"
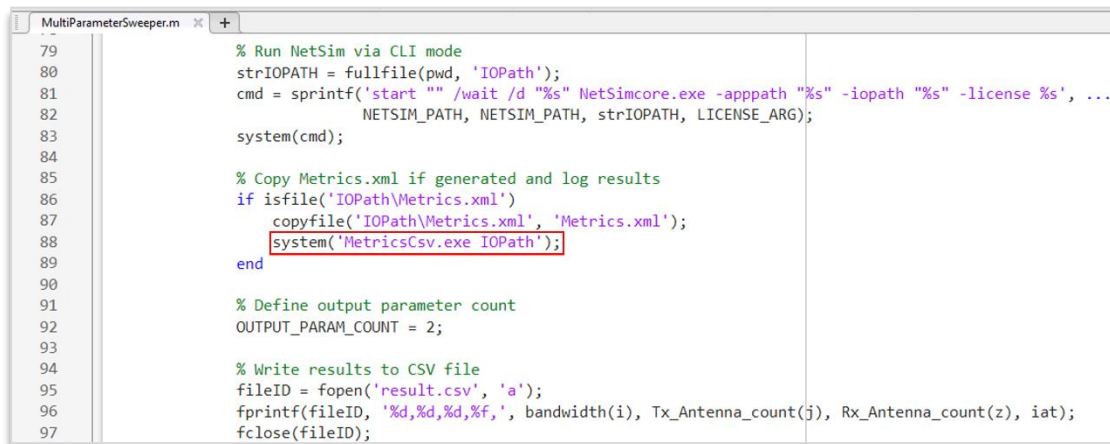
Floating on-premise license (<port no>@<server ip address>):

LICENSE_ARG = "5053@192.168.0.4"

Node Locked or Cloud licenses (path of license file):

LICENSE_ARG="\"C:\\Program Files\\NetSim\\Standard_v14_3\\bin\\bin_x64\""

- The MetricsCSV.exe will convert the Metrics.xml file inside the output folder into a csv file.

```
     MultiParameterSweeper.m   ×   +
79              % Run NetSim via CLI mode
80              strIOPATH = fullfile(pwd, 'IOPath');
81              cmd = sprintf('start "" /wait /d "%s" NetSimcore.exe -apppath "%s" -iopath "%s" -license %s', ...
82                      NETSIM_PATH, NETSIM_PATH, strIOPATH, LICENSE_ARG);
83              system(cmd);
84
85              % Copy Metrics.xml if generated and log results
86              if isfile('IOPath\Metrics.xml')
87                      copyfile('IOPath\Metrics.xml', 'Metrics.xml');
88                      system('MetricsCsv.exe IOPath');
89              end
90
91              % Define output parameter count
92              OUTPUT_PARAM_COUNT = 2;
93
94              % Write results to CSV file
95              fileID = fopen('result.csv', 'a');
96              fprintf(fileID, '%d,%d,%d,%f,', bandwidth(i), Tx_Antenna_count(j), Rx_Antenna_count(z), iat);
97              fclose(fileID);
```

Figure 7: MetricsPrint.csv file will be created in the IOPath and then copied into the respective output folder

8. Multi-Parameter Sweeping process can be started from MATLAB command line in the directory of the Multi-Parameter-Sweeping project as shown below:
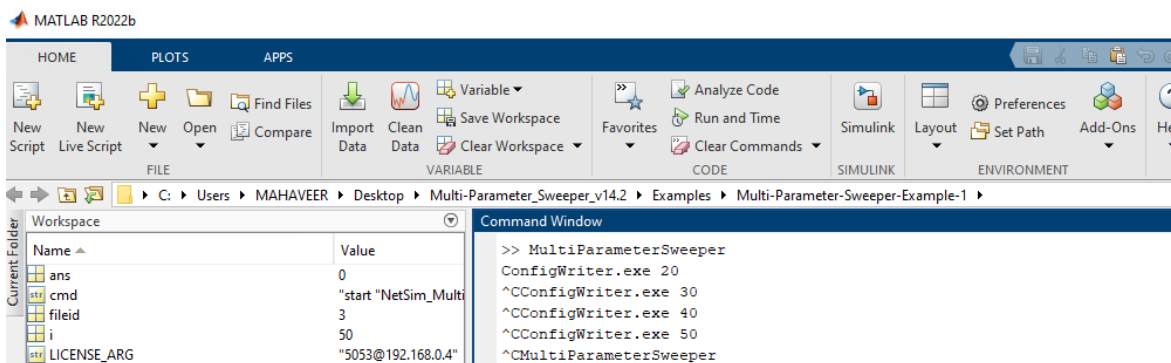

Figure 8: Running matlab script

9. This starts the Multi-Parameter-Sweeping process which runs NetSim simulations iteratively for different values of Y parameter of Wireless Node.

10. At the end of the process the Multi-Parameter-Sweeping folder will have the following file and folders created:
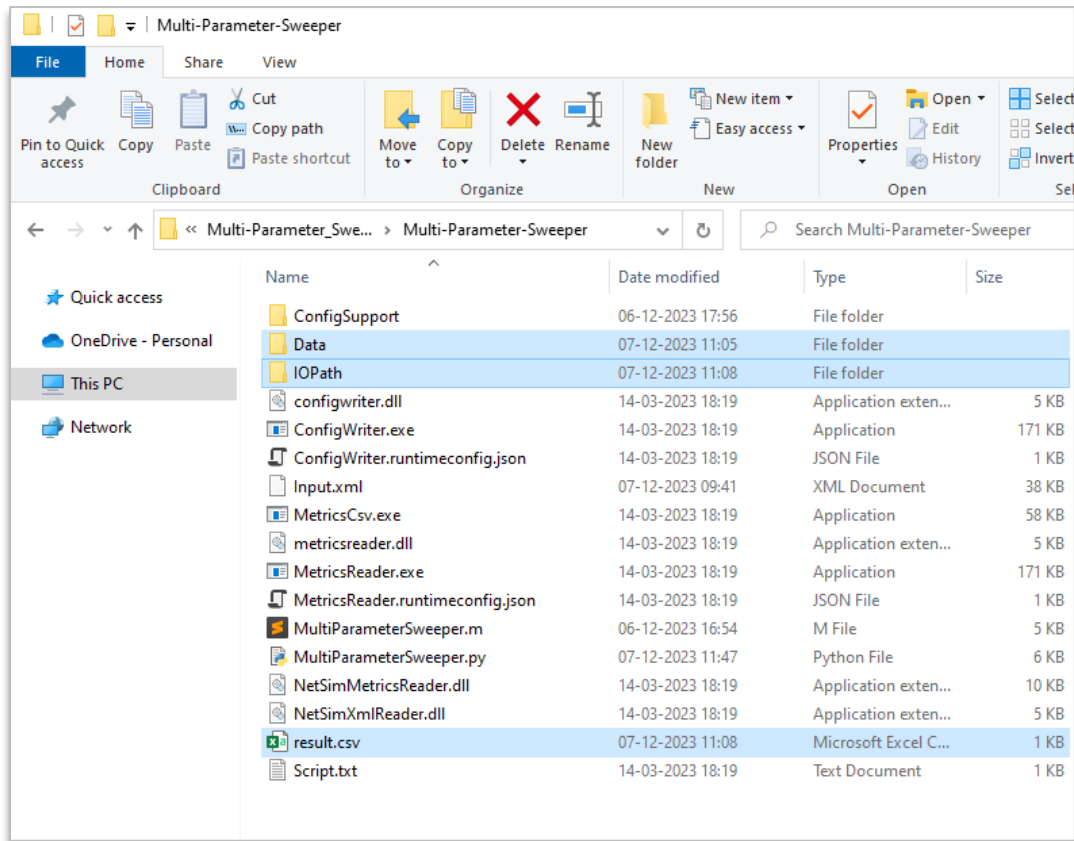
Figure 9: After Simulation Multi-Parameter-Sweeping folder contains output files like result.csv, Data etc

- **Data**: Contains multiple folders created based on date and time of simulation inside which multiple output folders corresponding to each simulation run, with its name including the value of the parameters in that iteration gets created.
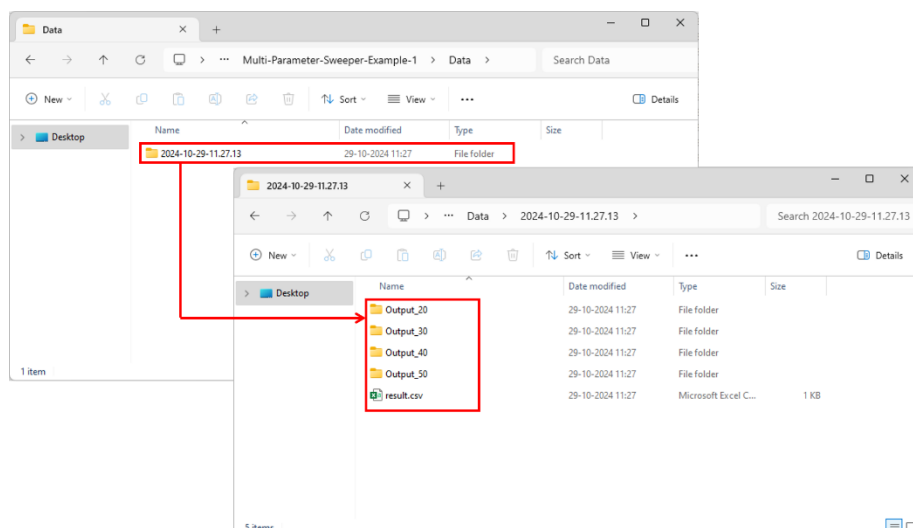

Figure 10: Based on values in the iteration, Output folder gets created in the Data directory inside a folder named as per date and time of simulation along with a copy of result.csv file.

- Each folder contains the all the output files associated with the simulation run.
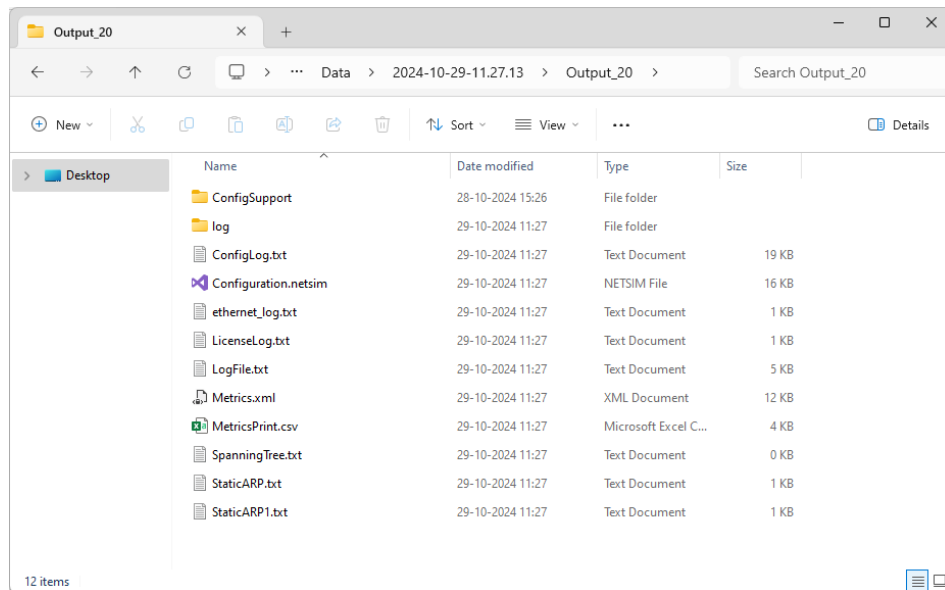
Figure 11: Each folder contains all the output files and the Metrics.xml file converted to MetricsPrint.csv file

Note: User should keep a back-up of the data folder to avoid data loss.

- **IO Path**: Used for storing the Configuration.netsim file and the simulation files generated during each simulation run.

- **Result.csv**: This is the output log which contains the parameter varied during each simulation run and the output parameter associated with each run. The result.csv file will also be copied into the output folder after each simulation.

Figure 12: Iterated value and Throughput obtained listed in result.csv

Example 1: Modifying a single input parameter and logging a single output parameter

Consider the following Internetwork scenario in NetSim, comprising of a Wired Node, Router, Access Point and a Wireless Node.
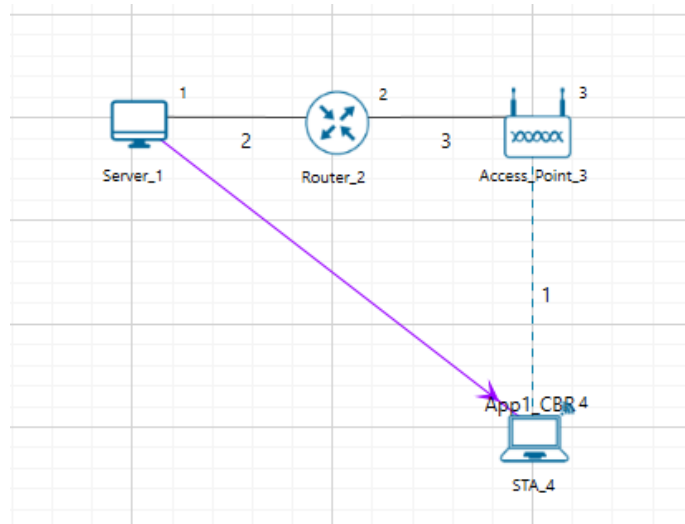
Figure 13: Network Topology

The network configuration has the initial distance between the Access Point and Wireless Node as 10 meters with the Access Point located at (15,10) and Wireless Node located at (15,20).

Multi-Parameter Sweeper is configured to run simulations for different distance between the Access Point and Wireless Node by varying the Wireless Node Y coordinate value from 20 to 50 in steps of 10 meters.

The network scenario is saved and the content of the Configuration.netsim file is copied to the Multi-Parameter-Sweeper directory and renamed as input.xml.

Refer to the Example 1 directory which is part of the project folder (Multi-Parameter-Sweeper_v14.3-main\Examples\Multi-Parameter-Sweeper-Example-1)
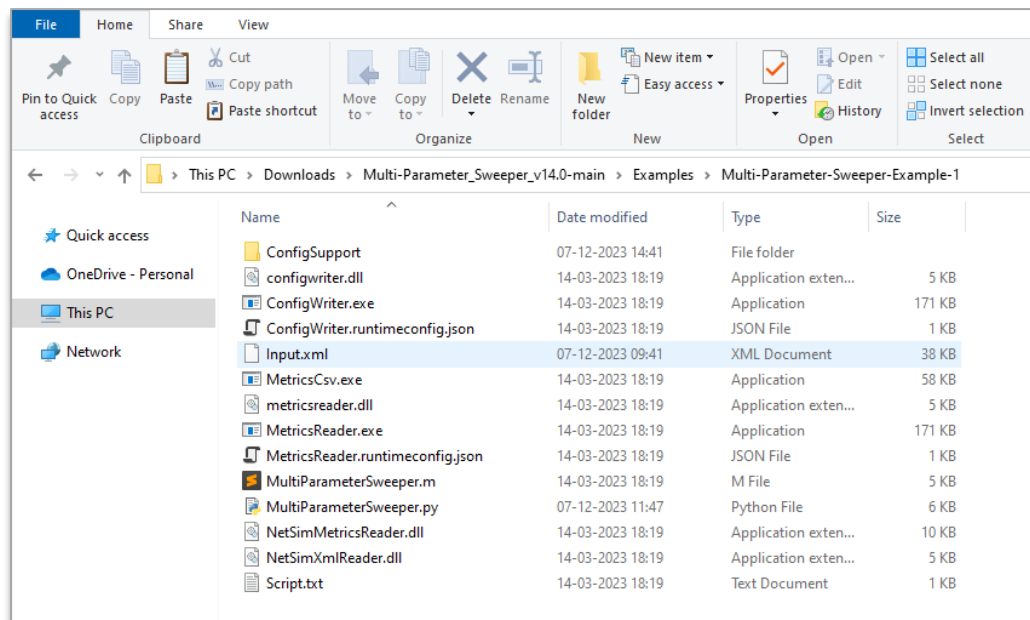
Figure 14: Renamed Configuration.netsim to input.xml and pasted in Multi-Parameter-Sweeper directory

1. The value of the Y coordinate of Wireless Node that is to be modified during each simulation run is updated ("{0}") in the configuration file as shown below:
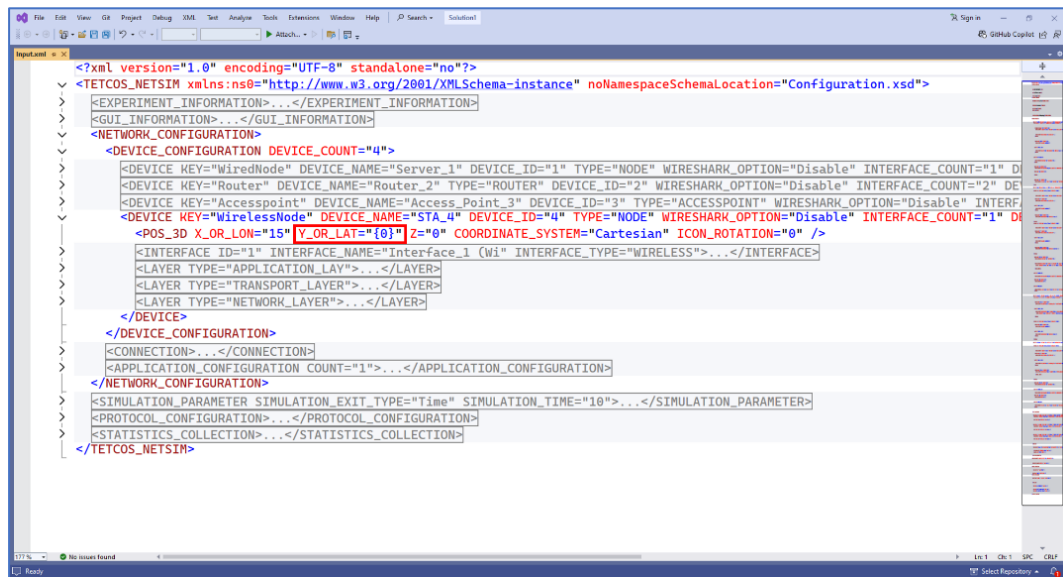
Figure 15: Modified Y coordinate of Wireless Node in input.xml

2. The Script.txt file is updated with the details of the output parameter to be read from the Metrics.xml file and added to the result csv log file. In this case the Application throughput is to be logged for each simulation run.
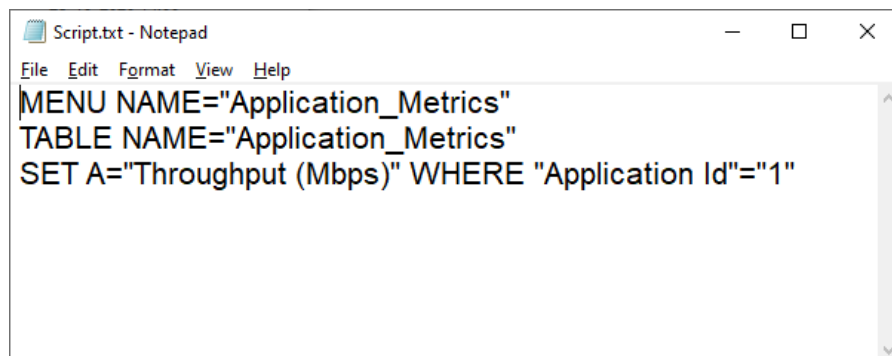

Figure 16: The application throughput is to be logged for each simulation modified in Script.txt

3. MultiParameterSweeper.m is updated to pass the Y coordinate value during each iteration to generate Configuration file run simulation and update the result csv log.

The MultiParameterSweeper script modified for running simulations for different values of Y coordinates starting from 20 up to 50 in steps of 10 is shown below:

- A result.csv file is created and added with headings Y and Throughput (Mbps).
- A MetricsPrint.csv is created inside every output folder.
- For loop is set to iteratively run simulations for values starting from 20 to 50 in steps of 10.
- The value of the parameter Y in the current iteration is written to the result log file for analysis.
- The value of the parameter Y in the current iteration is passed as input to ConfigWriter executable to generate Configuration.netsim file for each simulation.

- NetSim simulation is run via CLI mode by passing the apppath, iopath and license server information
- Configuration file and Metrics file are copied and renamed appending the value of the parameter in the current iteration.

The MultiParameterSweeper.m matlab script modified for running simulations for different values of Y coordinates starting from 20 up to 50 in steps of 10 is shown below:


```matlab
% Set the path of 64 bit NetSim Binaries to be used for simulation.
NETSIM_PATH = "C:\Program Files\NetSim\Standard_v14_2\bin\bin_x64";
LICENSE_ARG = "5053@192.168.0.4"; %Floating license
%LICENSE_ARG = """C:\Program Files\NetSim\Standard_v14_2\bin"""; %Node Locked license

% Set NETSIM_AUTO environment variable to avoid keyboard interrupt at the end of each simulation
setenv('NETSIM_AUTO','1');

% Create IOPath directory to store the input Configuration.netsim file and
% the simulation output files during each iteration
if(~isfolder('IOPath'))
    mkdir IOPath
end

% Create Data directory to store the Configuration.netsim and the
% Metrics.xml files associated with each iteration
if(~isfolder('Data'))
    mkdir Data
end

% Clear the IOPath folder if it has any files created during previous
delete('IOPath\*')

% Delete result.csv file if it already exists
if(isfile('result.csv'))
    delete result.csv
end

% create a csv file to log the output metrics for analysis csvfile
fileid=fopen('result.csv','w');

% Add headings to the CSV file
fprintf(fileid,'Y,Throughput(Mbps)');
fclose(fileid);
```

Figure 17: To Create result.csv file, added with headings Y and Throughput (Mbps) and NetSim installation Path and License information

- NETSIM_PATH variable is set to the path of NetSim 64-bit binaries in the install directory or workspace in the system.
- LICENSE_ARG variable is set to the license server details
- A result.csv file is created and added with headings Y and Throughput (Mbps).


```matlab
% Create a folder with name as year-month-day-hour.minute.seconds inside the data folder
today = datestr(now,'dd-mm-yyyy-HH.MM.SS');
mkdir('Data',today)

% Iterate based on the number of time simulation needs to be run and the
for i =  20:10:50

    if(isfile('Configuration.netsim'))
        delete Configuration.netsim
    end

    if(isfile("IOPath\\Configuration.netsim"))
        delete('IOPath\Configuration.netsim')
    end

    if(isfile('IOPath\Metrics.xml'))
        delete('IOPath\Metrics.xml')
    end

    %Call ConfigWriter.exe with arguments as per the number of variable parameters in the input.xml file
    cmd="ConfigWriter.exe "+string(i);
    disp(cmd)
    system(cmd);
```

Figure 18: Varying Distance and set license server information

- For loop is set to iteratively run simulations for values starting from 20 to 50 in steps of 10.

- The value of the parameter Y in the current iteration is passed as input to ConfigWriter executable to generate Configuration.netsim file for each simulation.
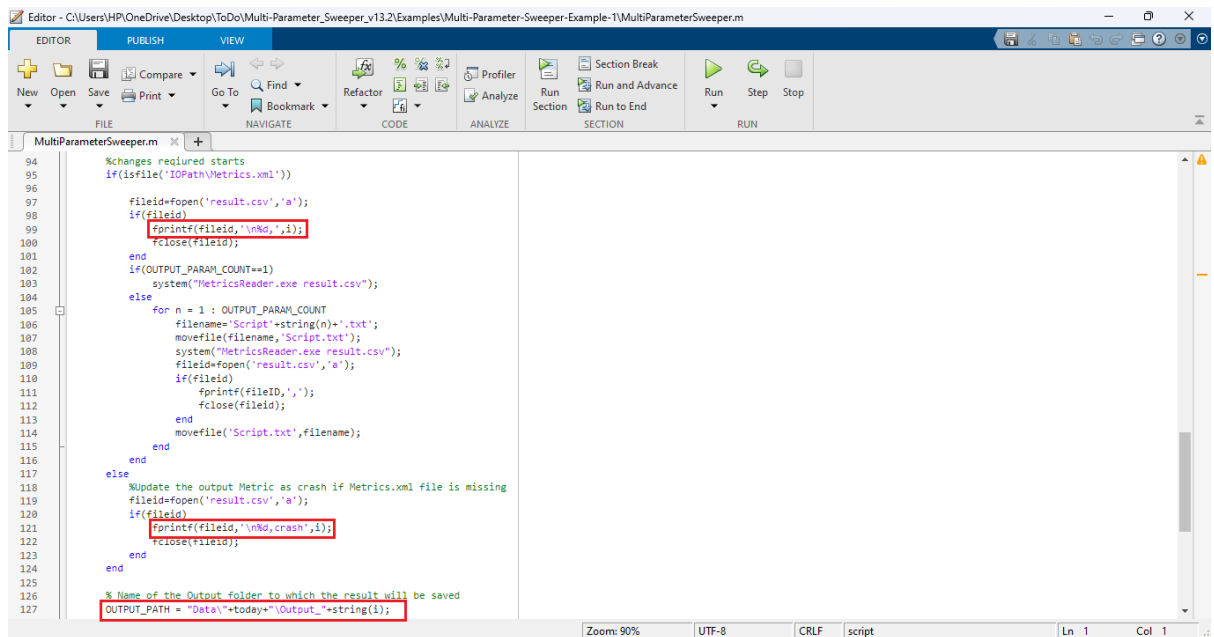

Figure 19: Modify parameters in MultiParameterSweeper.m

- The value of the parameter Y in the current iteration is written to the result log file for analysis.
- Configuration file and Metrics file are copied and renamed appending the value of the parameter in the current iteration.

4. Multi-Parameter Sweeping process can be started from MATLAB command line in the directory of the Multi-Parameter-Sweeping project as shown below:
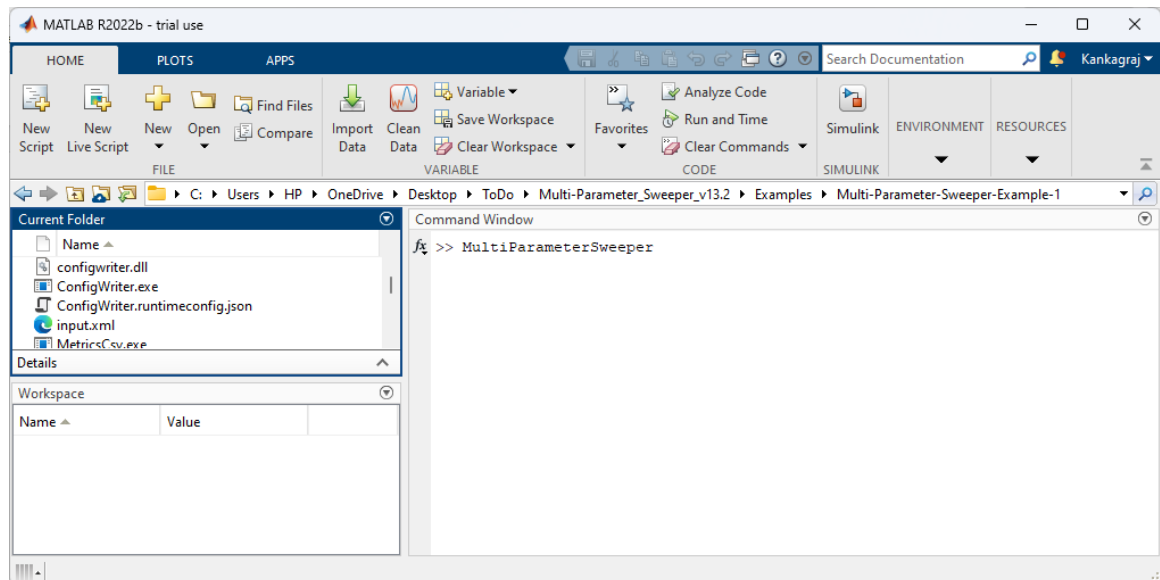
Figure 20: Running matlab script

This starts the Multi-Parameter-Sweeping process which runs NetSim simulations iteratively for different values of Y parameter of Wireless Node.

At the end of the process the Multi-Parameter-Sweeping folder will have the following file and folders created:
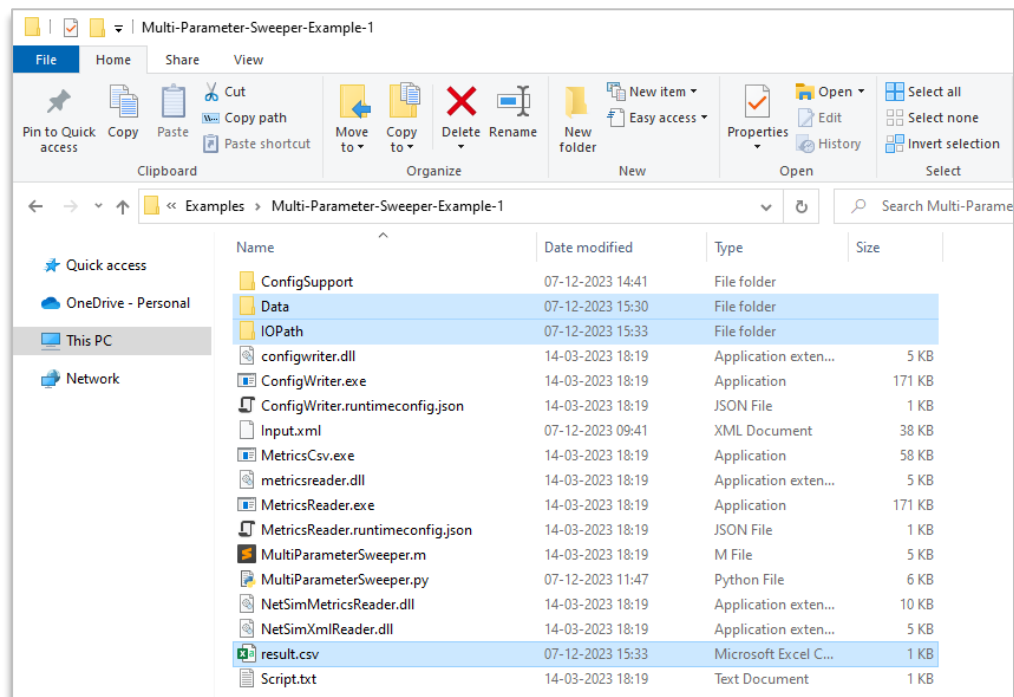
Figure 21: After Simulation Multi-Parameter-Sweeping folder contains output files like result.csv, Data etc

- **Data**: Contains multiple folders corresponding to each simulation run, with its name including the value of the parameters in that iteration. The output folders will be created inside folder with name in the format Year-Month-Day-Hours-Minutes-Seconds.
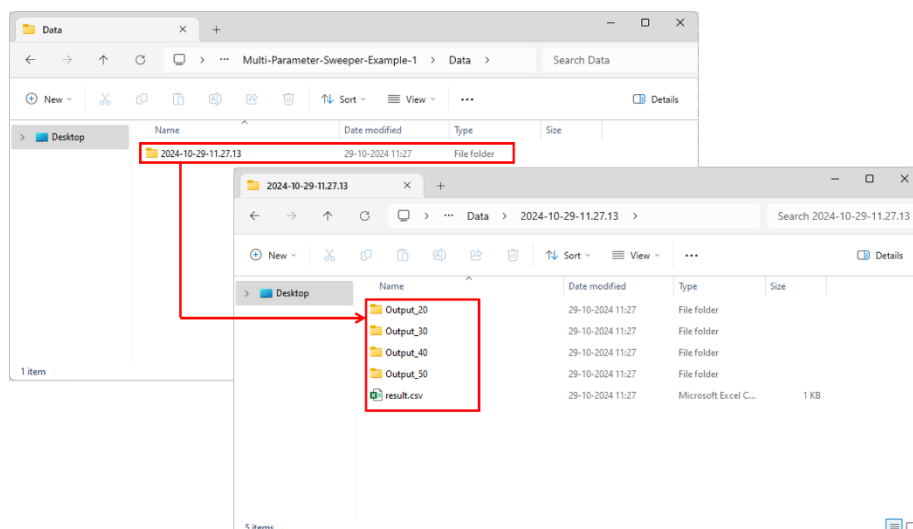
Figure 22: Based on distance Configuration.netsim files created in output folder

- Each folder contains the all the output files associated with the simulation run.
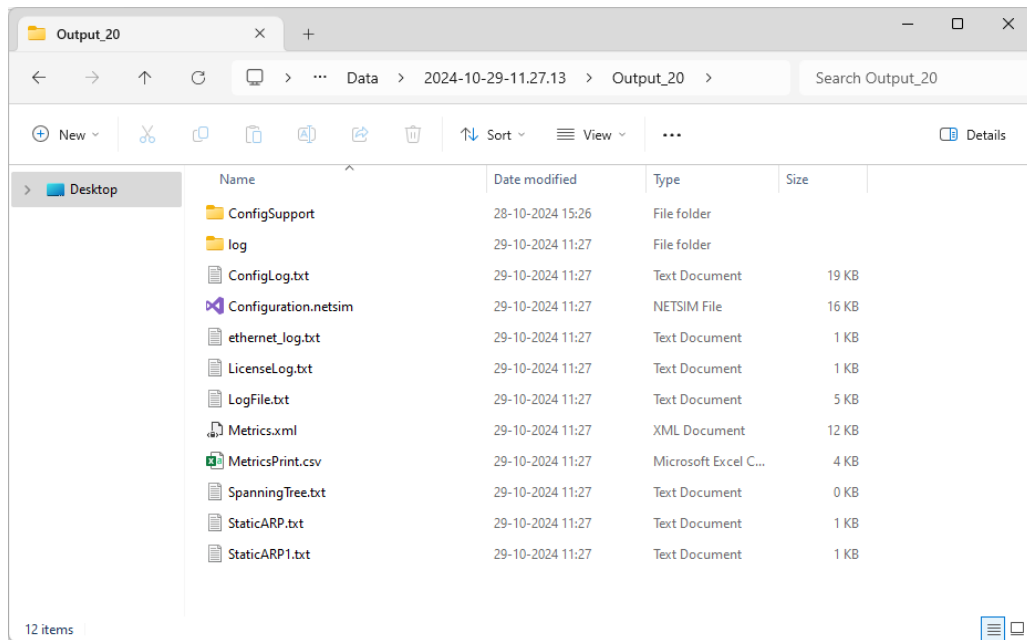
Figure 23: Each folder contains the all the output files

- **IOPath**: Used for storing the Configuration.netsim file and the simulation files generated during each simulation run.
- **Result.csv**: This is the output log which contains the parameter varied during each simulation run and the output parameter associated with each run.
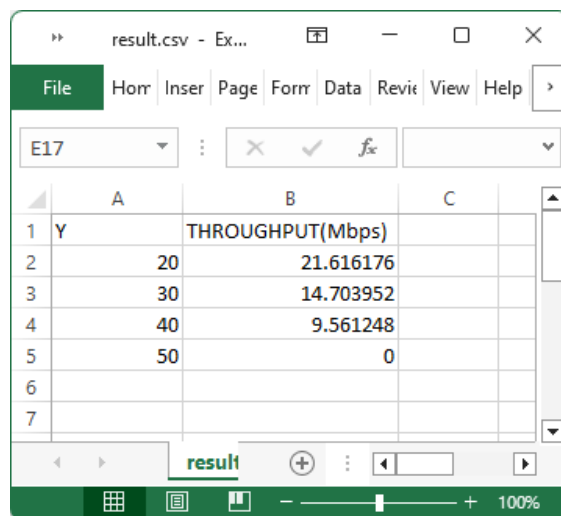
Figure 24: Distance Vs. Throughput obtained in result.csv

Varying multiple network parameters

In order to vary multiple network parameters during the multi-parameter sweep process each parameter in the input.xml file can be modified as {0},{1},{2},{3},..{n} respectively.

Logging multiple output parameters

Each output parameter that is to be logged should be part of the Script.txt file. However, the Script.txt file should contain only the details of one output parameter during the call to MetricsReader.exe.

To log multiple parameters, multiple script files can be used. If n output parameters are to be logged, then there can be script1.txt, script2.txt, script.txt in the sweeper folder.

For Example, there can be two Script files as shown below:

Figure 25: Multiple output parameters

Example 2: Modifying multiple input parameters and logging multiple output parameter

1. Consider the following interwork scenario in NetSim, comprising of a Wired Node, Router, Access point and a Wireless Node.
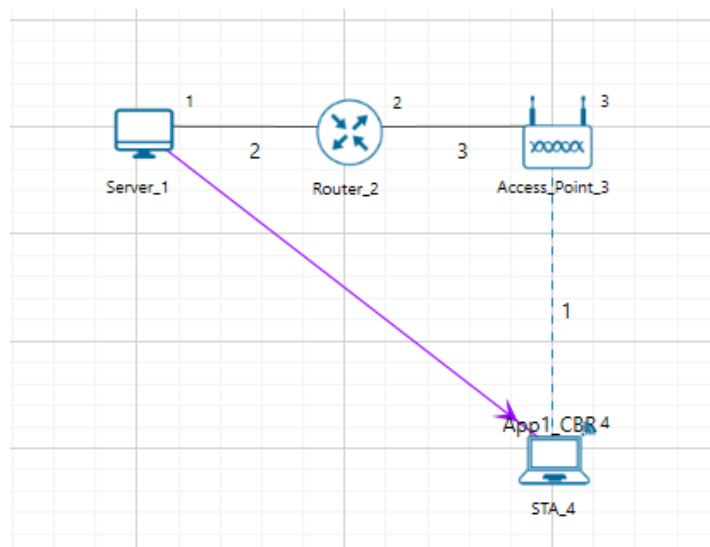
2. Set grid length to 60m x 60m grid setting property panel on the right. This needs to be done before the any device is placed on the grid.

3. Distance between Access Point and the Wireless Node is set to 10m

4. Set DCF as the medium access layer protocol under datalink layer properties of access point and wireless node. To configure any properties in the nodes, click on the node, expand the property panel on the right side, and change the properties as mentioned in the below steps.

5. WLAN Standard is set to 802.11ac and No. of Tx and Rx Antenna is set to 1 in access point and No. of Tx is 1 and Rx Antenna is set to 1 in wireless node (Right-Click Access Point or Wireless Node > Properties > Interface Wireless > Transmitting Antennas and Receiving Antennas) and Bandwidth is set to 20 MHz in both Access-point and wireless-node Transmitter Power set to 100mW in both Access-point and wireless-node.

6. Wired Link speed is set to 1Gbps and propagation delay to 10 μs in wired links.

7. Channel Characteristics: Path Loss Only, Path Loss Model: Log Distance, Path Loss Exponent: 3.5.

8. Configure an application between any two nodes by selecting a CBR application from the Set Traffic tab in the ribbon on the top. Click on created application and expand the application property panel on the right and set transport protocol to UDP, packet size to 1460 B and Inter arrival time to 116.8 μs.

9. Application Generation Rate: 100 Mbps (Packet Size: 1460, Inter Arrival Time: 116.8 μs)

10. Run the simulation for 10s.

11. The network scenario is saved and the content of the Configuration.netsim file is copied to the Multi-Parameter-Sweeper directory and renamed as input.xml.

12. Refer to the Example 2 directory which is part of the project folder (Multi-Parameter-Sweeper_v14.3\Examples\Multi-Parameter-Sweeper-Example-2).
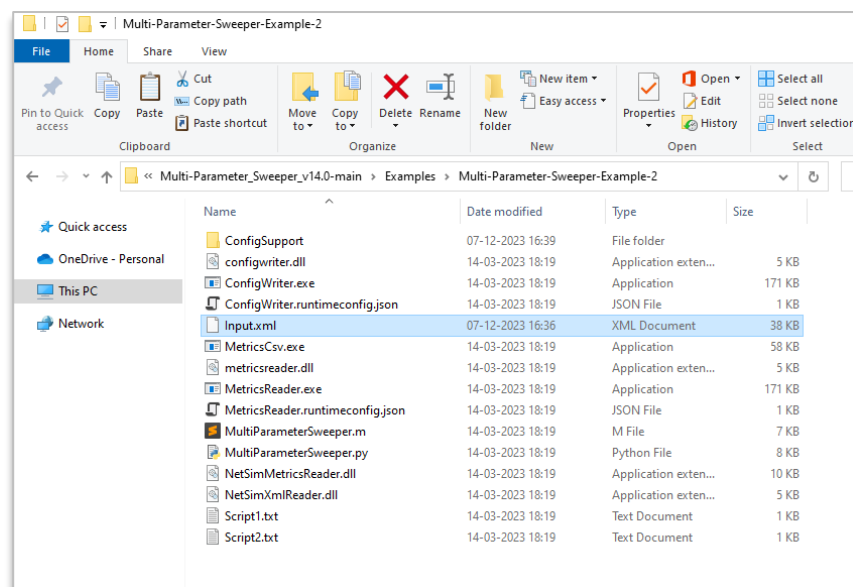
Figure 26: Renamed Configuration.netsim to input.xml and pasted in Multi-Parameter-Sweeper directory

13. In the Input.xml file the value of the input variables are modified as shown in the table below:

Input Variables	
Bandwidth (MHz)	{0}
Tx Antenna Count	{1}
Rx Antenna Count	{2}
Inter Arrival Time (Microseconds)	{3}

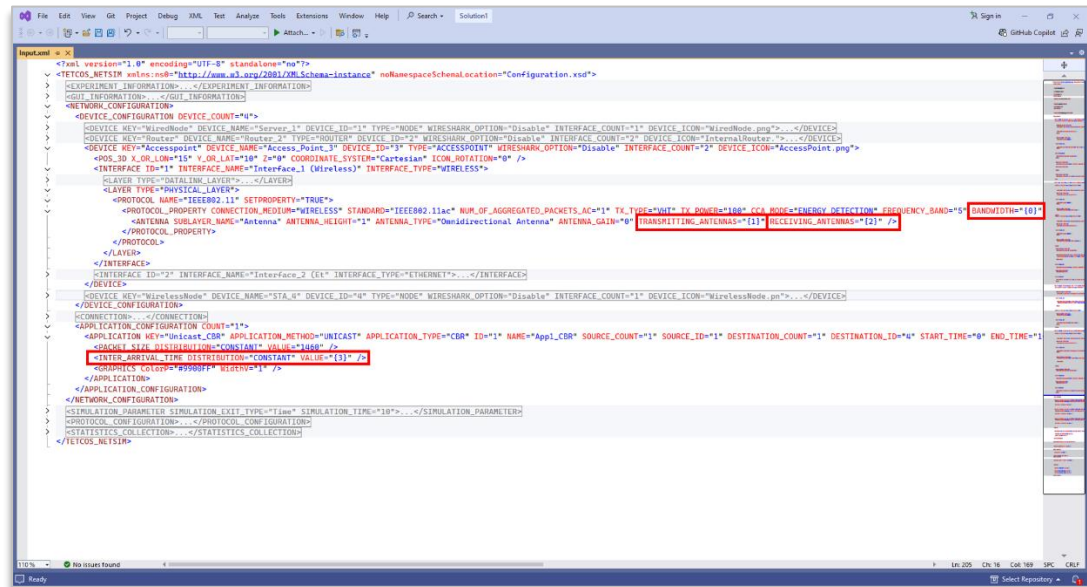Table 1: Variables are modified to the input.xml file

Figure 27: The above table 3 Variables are modified in input.xml file

14. The matlab script MultiParameterSweeper.m is modified to run simulation for all possible combinations of Bandwidth and Tx Antenna Count and Rx Antenna Count with the respective values of IAT that is calculated.
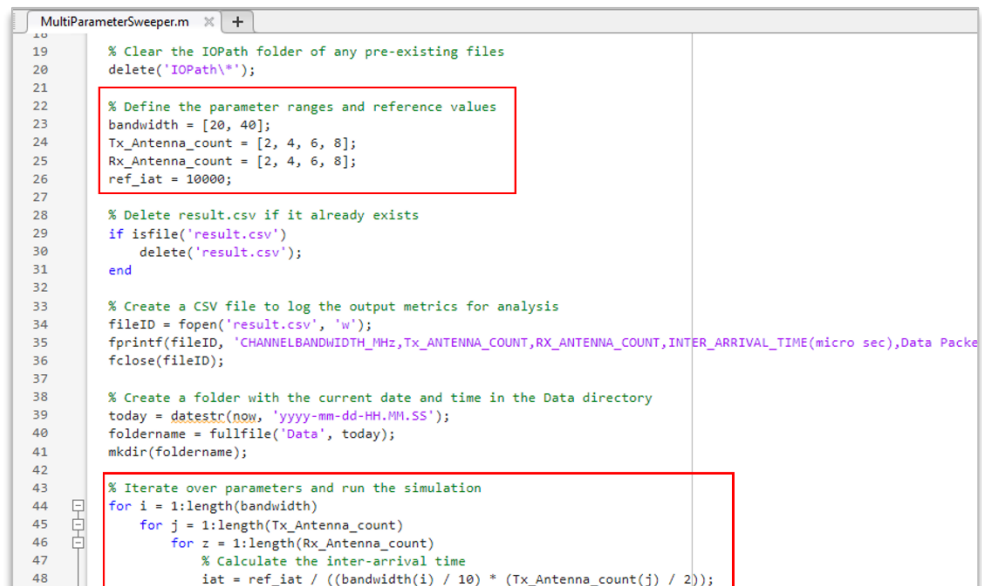

Figure 28: Modified MultiParameterSweeper.m based on input parameter
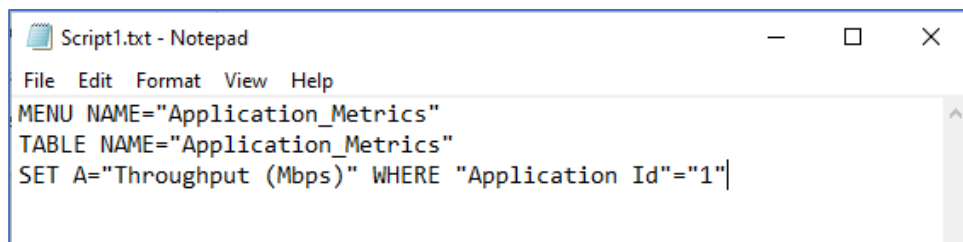
15. Multiple parameters are read from the Metrics.xml file and logged in the results.csv file along with the input parameters such as CHANNELBANDWIDTH_MHz, TX_ANTENNA_COUNT, RX_ANTENNA_COUNT, INTER_ARRIVAL_TIME (micro sec).

Output Parameters
Throughput (Mbps)
Data Packets transmitted

Table 2: User need to modify these two parameters in Script files

16. Two script text files namely Script1.txt and Script2.txt are created with information to read each of the parameters from the Metrics.xml file. The variable OUTPUT_PARAM_COUNT is set to 2 as per the number of Script files.
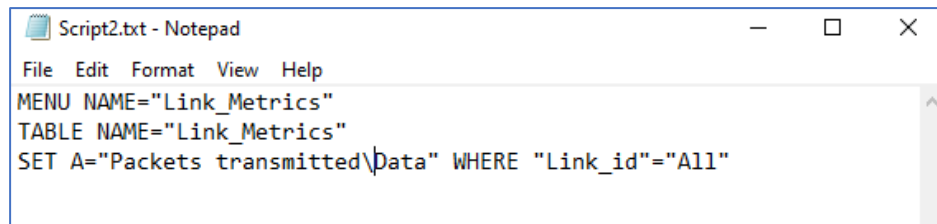
Script1.txt

Figure 29: The application throughput is to be logged for each simulation modified in Script1.txt

Scritp2.txt

Figure 30: The Data Packets transmitted is to be logged for each simulation modified in Script2.txt

17. In the matlab script MultiParameterSweeper.m, MetricsReader is called to log each parameter specified in the script text files separating the entries with a comma (","). If simulation crashes, without generating the output Metrics.xml, then "crash" message is written to the log for each output parameter. The input parameters that were varied during each simulation run are also logged in the results.csv file.
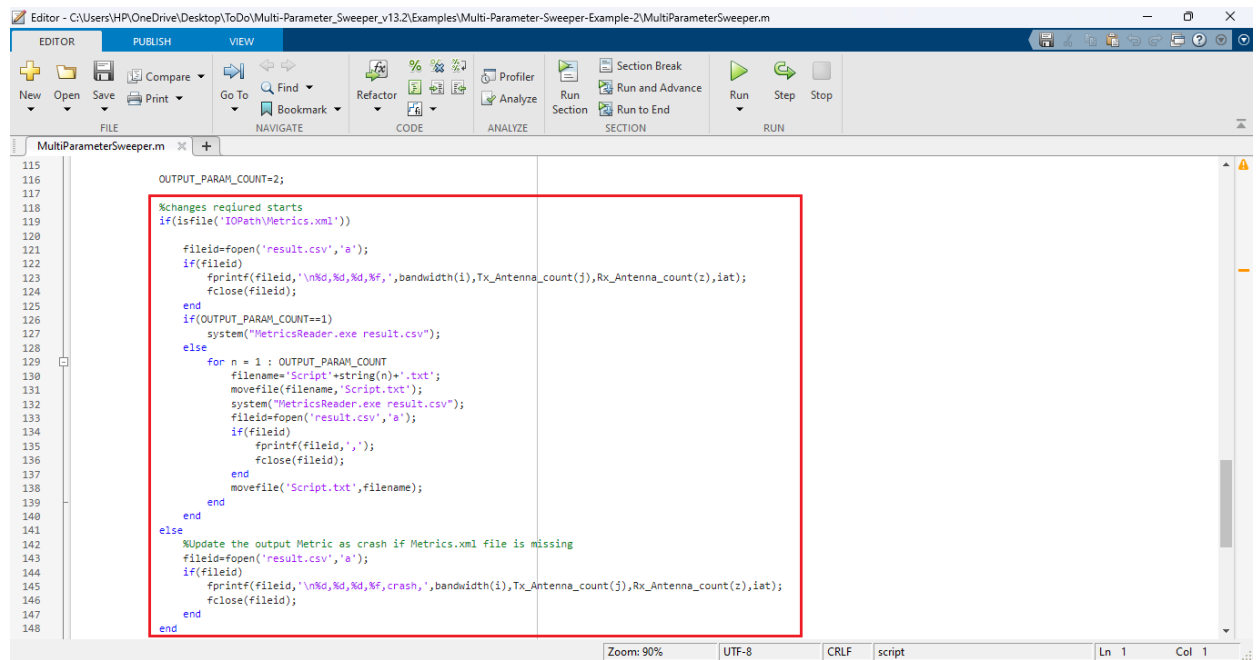
Figure 31: Modify matlab script MultiParameterSweeper.m file

18. The simulation Configuration file and all the output files associated with each simulation run is saved to folders with name including the bandwidth and DL MIMO count values that were used during each simulation run.
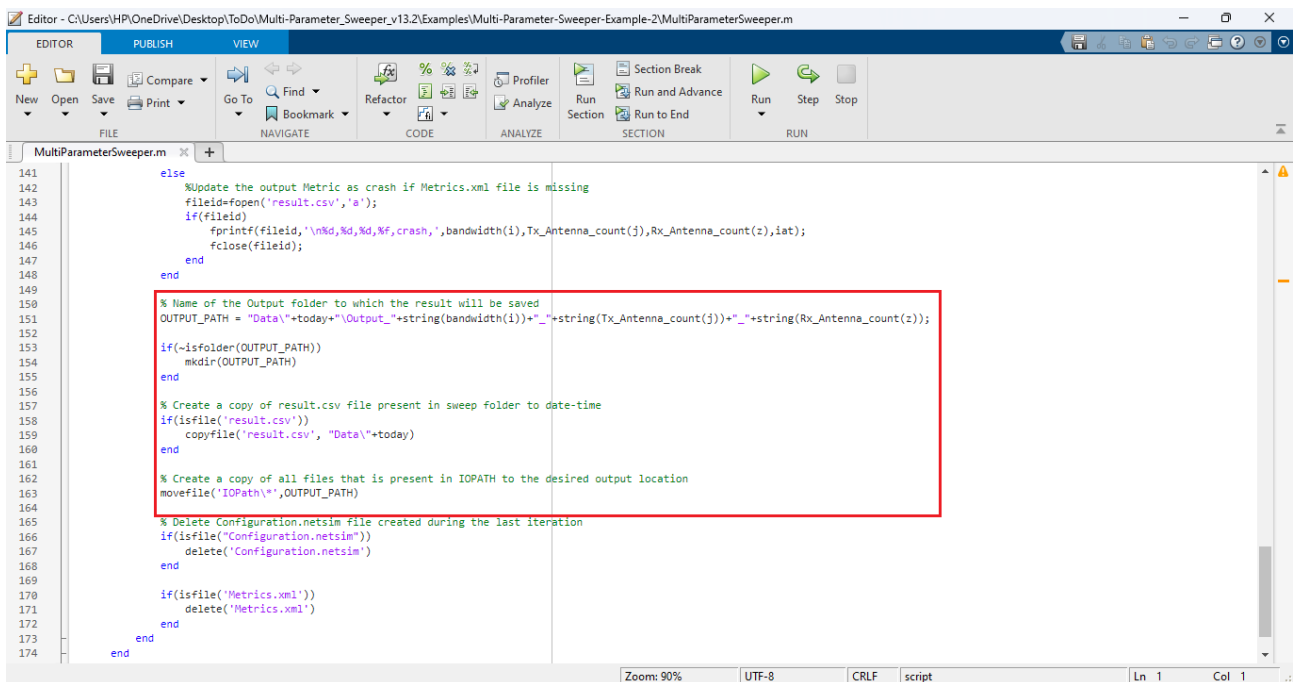

Figure 32: Modify matlab script MultiParameterSweeper.m file

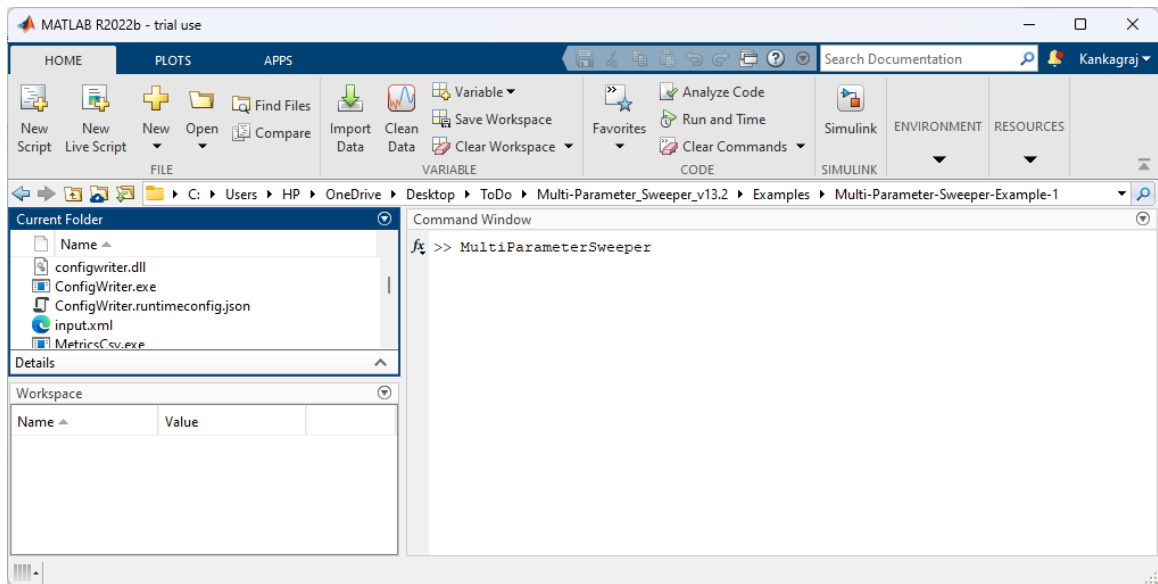19. Multi-Parameter Sweeping process can be started from MATLAB command line in the directory of the Multi-Parameter-Sweeping project as shown below:

Figure 33: Running MATLAB script.

This starts the Multi-Parameter-Sweeping process which runs NetSim simulations iteratively for different combinations of input parameters.

At the end of the process the Multi-Parameter-Sweeping folder will have the following file and folders created:
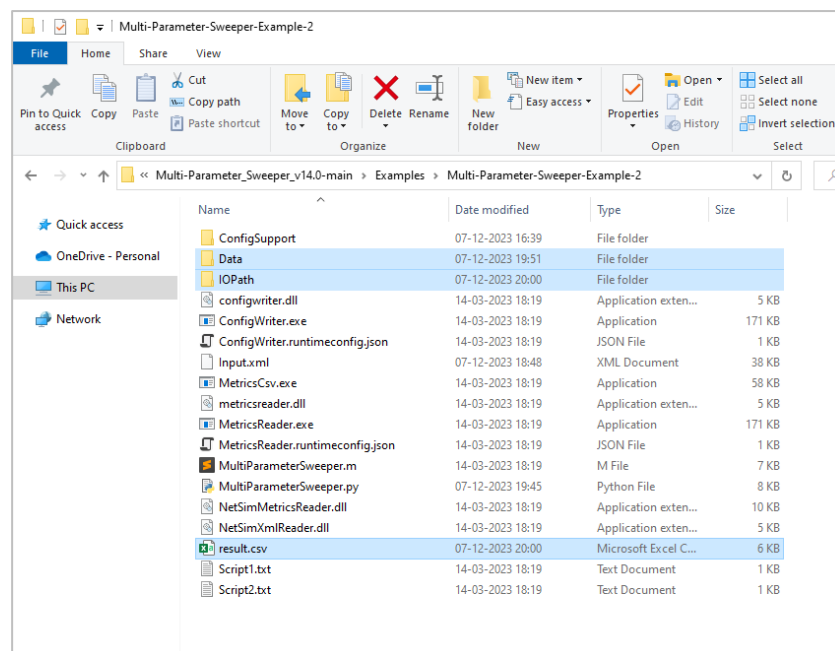
Figure 34: After Simulation Multi-Parameter-Sweeping folder contains output files like result.csv, Data etc

20. **Data**: The Data directory contains multiple output folders with the output files associated with each simulation run.
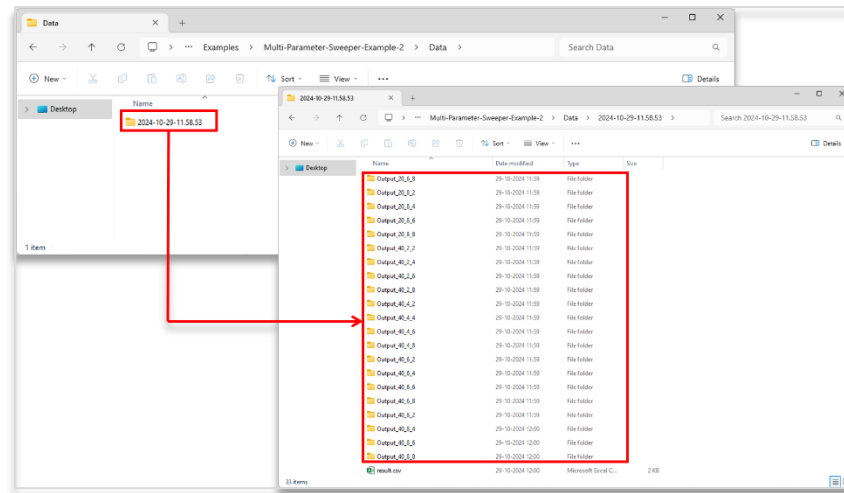
Figure 35: Based on input parameter Configuration.netsim and other files created in output folder

21. **IOPath**: Used for storing the Configuration.netsim file and the simulation files generated during each simulation run.

22. **Result.csv**: This is the output log which contains the parameter varied during each simulation run and the output parameter associated with each run.

Figure 36: Based on script result stored in result.csv file

Advanced use cases:

1. Running the sweeper with binaries of a modified workspace.
   When the source codes of a workspace are modified, a warning message will be displayed in the simulation console waiting for user interrupt, as shown below:

Figure 38: Running Sweeper through command prompt

To suppress this message and user interrupt, so that the sweeper runs without requiring any intervention, follow the steps given below:

- o Open the current workspace location by going to Your Work-> Workspaces-> Open Workspace Location.

- o Go the bin_x64 directory and rename the Checksum.txt file to say, Checksum_backup.txt as shown below:
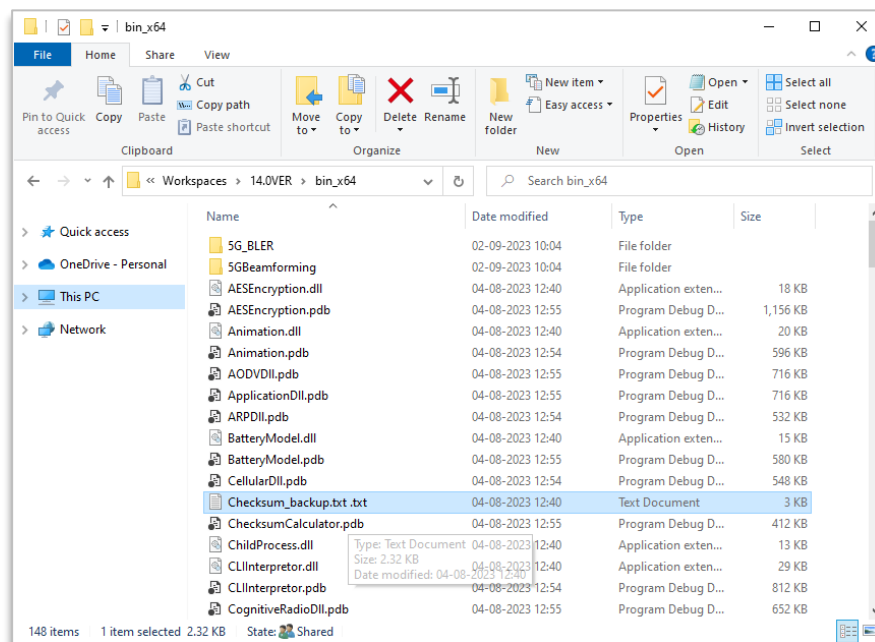

Figure 39: Renaming the checksum.txt file

Now upon running any further simulations, warning messages will not be printed in the simulation console and no user intervention will be required.

2. Sweeping Configuration files that has other associated files

There can be cases where a Configuration file requires other supporting files such as:

- Mobility – Mobility text file.
- Static route text files
- ACL input
- SUMO configuration files in case of VANET, etc

In such cases, the associated files can also be placed in the sweeper folder which contains the input.xml file and code can be slightly modified to copy all associated files when copying the Configuration file to the IOPath.