

# NetSim Multi-Parameter Sweeper MATLAB Script

**Software:** NetSim v13.2 (64 bit), DOT NET CORE SDK 3.1, MATLAB (2022 or higher)

## **Project Download Link:**

[https://github.com/NetSim-TETCOS/Multi-Parameter\\_Sweeper\\_v13.3/archive/refs/heads/main.zip](https://github.com/NetSim-TETCOS/Multi-Parameter_Sweeper_v13.3/archive/refs/heads/main.zip)

## **Introduction**

When users want to sweep one or more parameters, they change their values between simulation runs, and compare and analyse the performance metrics from each run. NetSim multi-parameter sweeper enables users to automate the sweep process.

Consider an example, where a user wishes to create and simulate a network scenario for all possible values of one or more parameters in combination and analyse a set of performance metrics across the simulation runs. This is extremely time consuming to do manually using the NetSim GUI.

The multi-parameter sweep program enables users to automate the sweep process across multiple input parameters, simulate each run, save each result, and compare specific output metrics via a spreadsheet software like MS Excel.

The sweep program runs NetSim via its CLI interface.

## **File Organization**

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

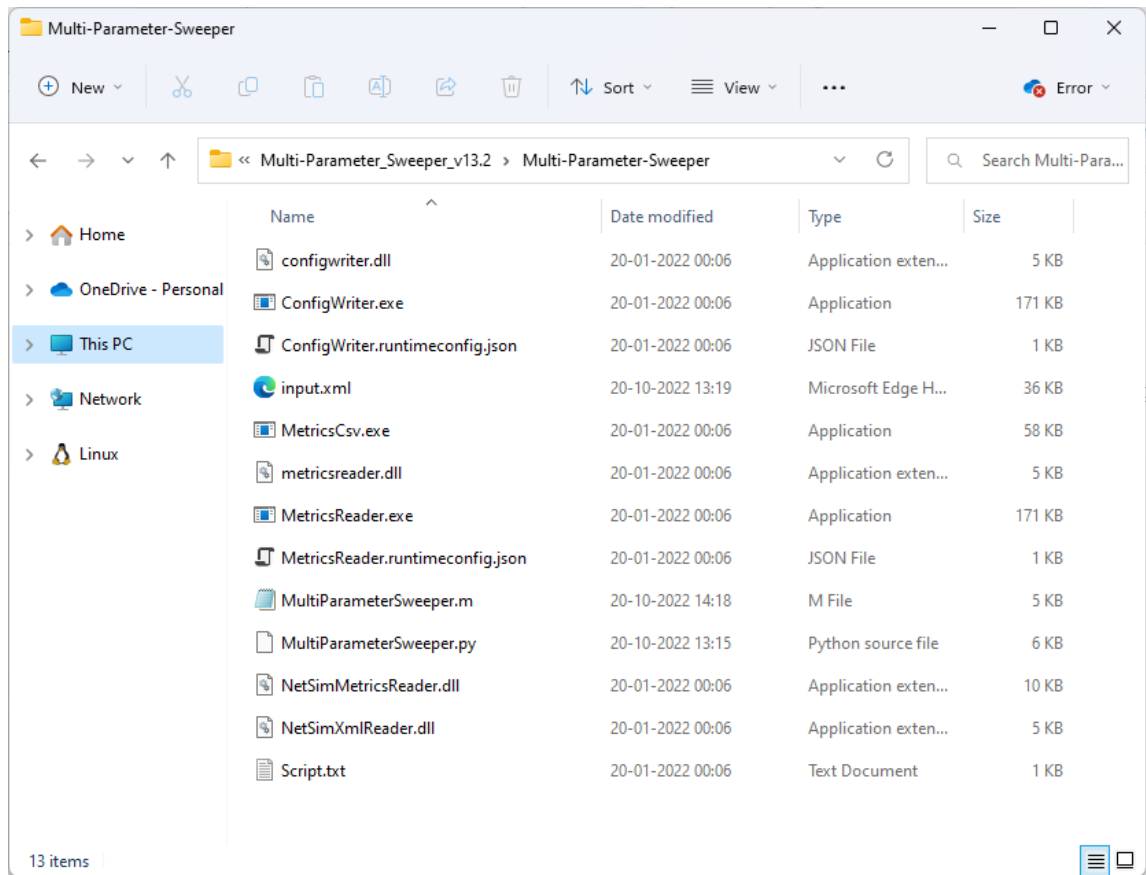


Figure 1: project directory consists of several binaries

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.

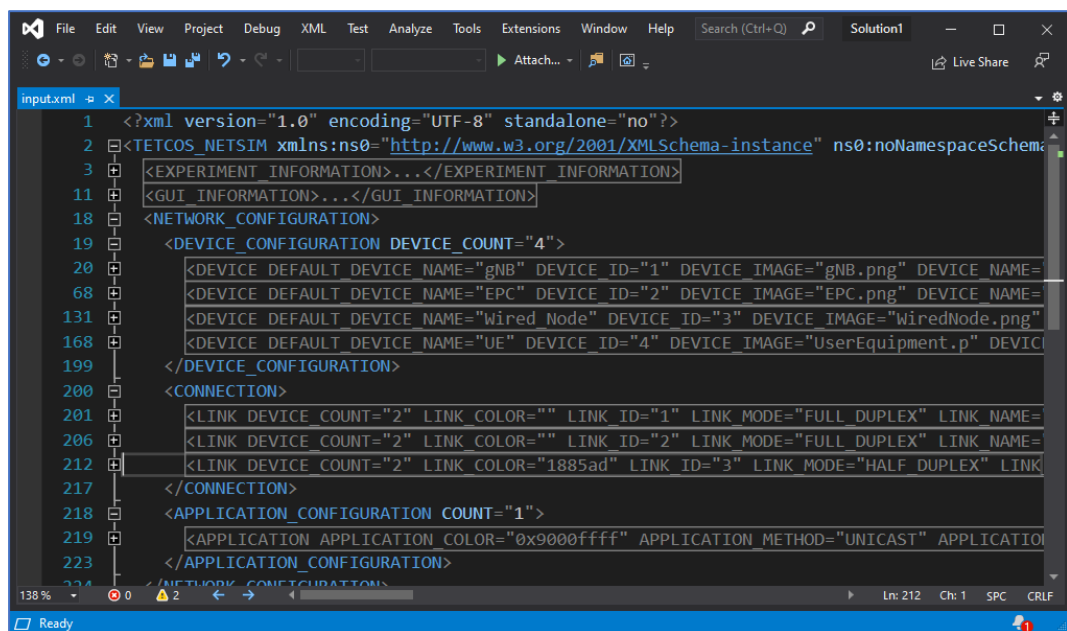
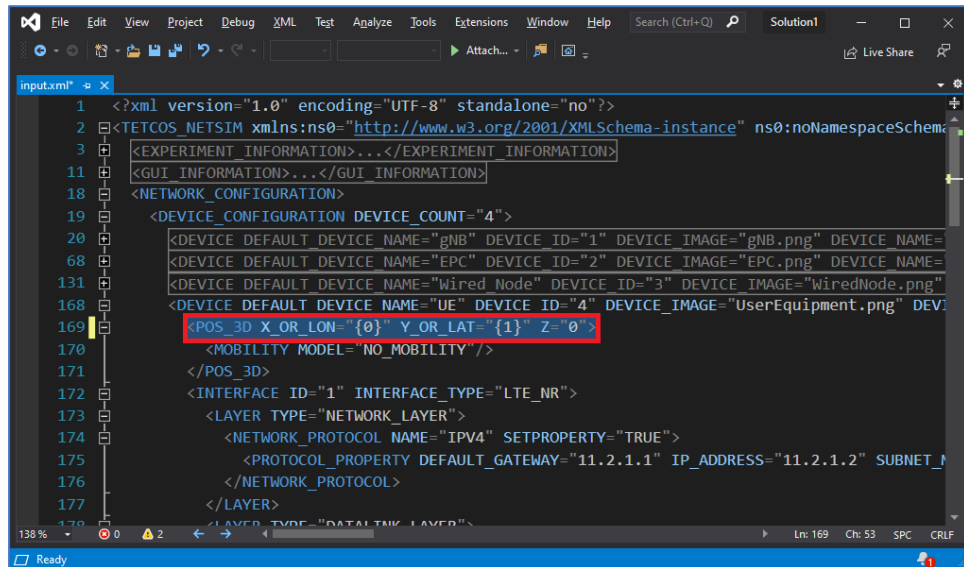


Figure 2: NetSim input Configuration.netsim file

- 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 X and 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:

```

360 <MENU Name="UDP_Metrics">...</MENU>
383 <MENU Name="Application_Metrics">
384 <TABLE name="Application_Metrics">
385 <TH name="Application Id" isShow="true"/>
386 <TH name="Throughput Plot" isShow="true"/>
387 <TH name="Application Name" isShow="true"/>
388 <TH name="Source Id" isShow="false"/>
389 <TH name="Destination Id" isShow="false"/>
390 <TH name="Packet generated" isShow="true"/>
391 <TH name="Packet received" isShow="true"/>
392 <TH name="Payload generated (bytes)" isShow="false"/>
393 <TH name="Payload received (bytes)" isShow="false"/>
394 <TH name="Throughput (Mbps)" isShow="true"/>
395 <TH name="Delay(microsec)" isShow="true"/>
396 <TH name="Jitter(microsec)" isShow="true"/>
397 <TR>
398 <TC Value="1"/>
399 <TC>...</TC>
402 <TC Value="App1_CBR"/>
403 <TC Value="3"/>
404 <TC Value="4"/>
405 <TC Value="25000"/>
406 <TC Value="17946"/>
407 <TC Value="36500000"/>
408 <TC Value="26201160"/>
409 <TC Value="4192.185600"/>
410 <TC Value="7188.952970"/>

```

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:

```

MENU NAME="Application_Metrics"
TABLE NAME="Application_Metrics"
SET A="Throughput (Mbps)" WHERE "Application Id"="1"

```

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.

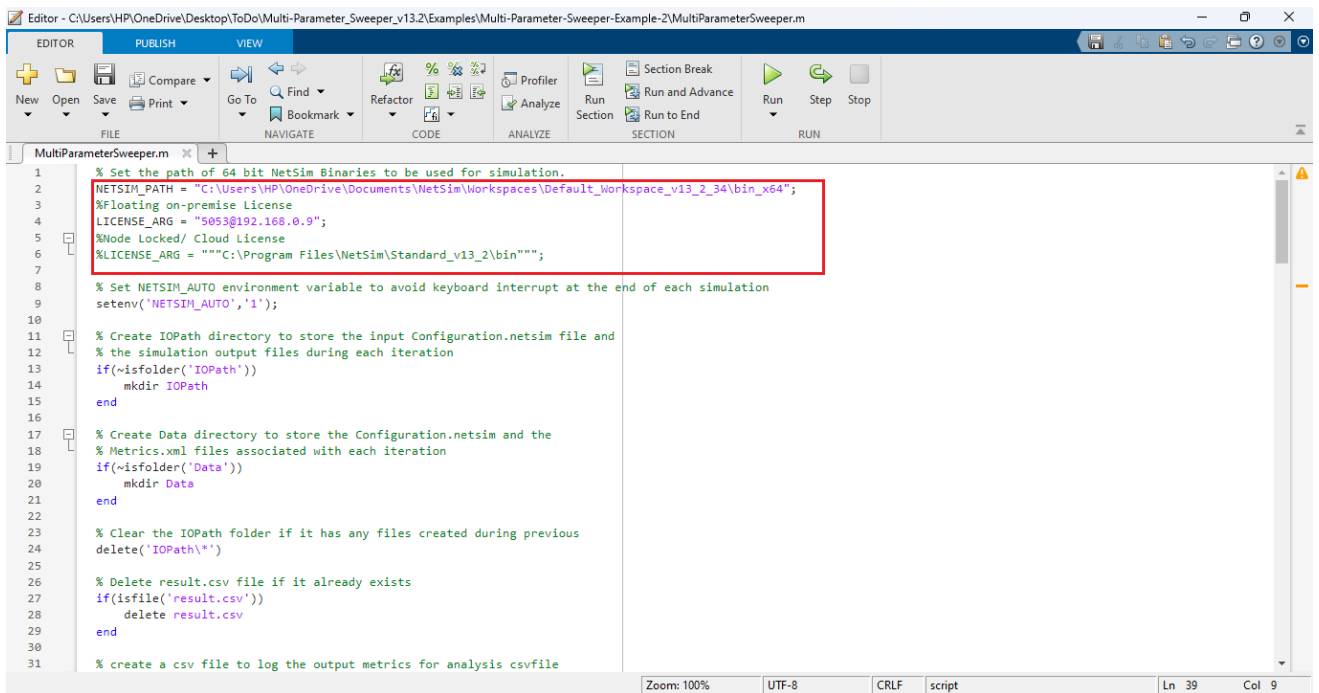
It uses the Script.txt file to determine which parameter to read from the Metrics file.

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.



**Figure 6:** User need to set NetSim Path based on 64-bit

For example,

NETSIM\_PATH

=

"C:\Users\HP\OneDrive\Documents\NetSim\Workspaces\Default\_Workspace\_v13\_2\_34\bin\_x64";

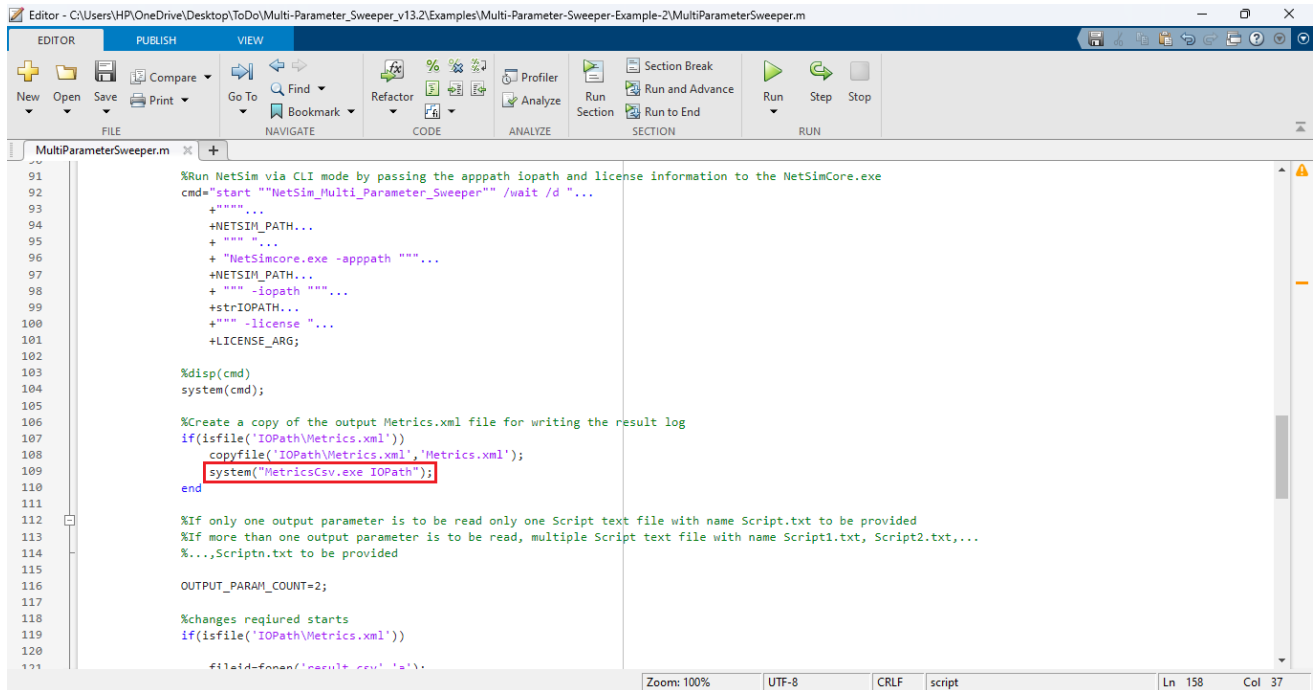
**Floating on-premise license (<port no>@<server ip address>):**

LICENSE\_ARG = "5053@192.168.0.9";

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

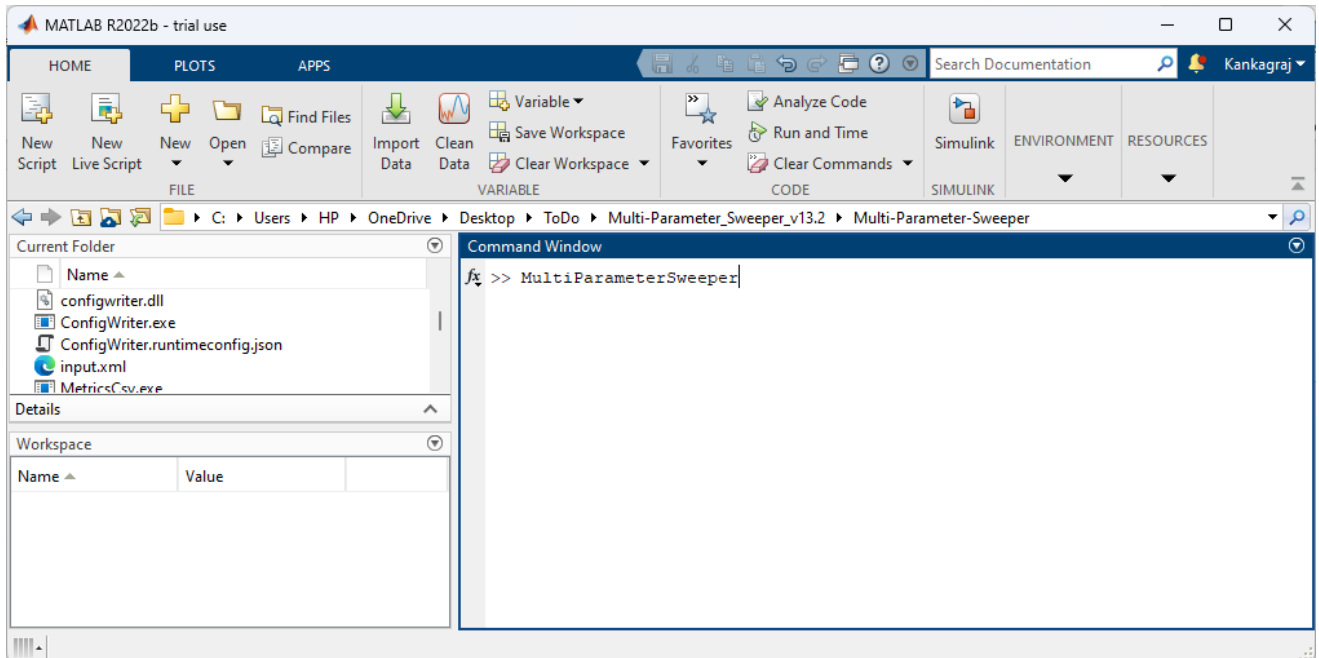
LICENSE\_ARG = ""C:\Program Files\NetSim\Standard\_v13\_2\bin"";

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



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

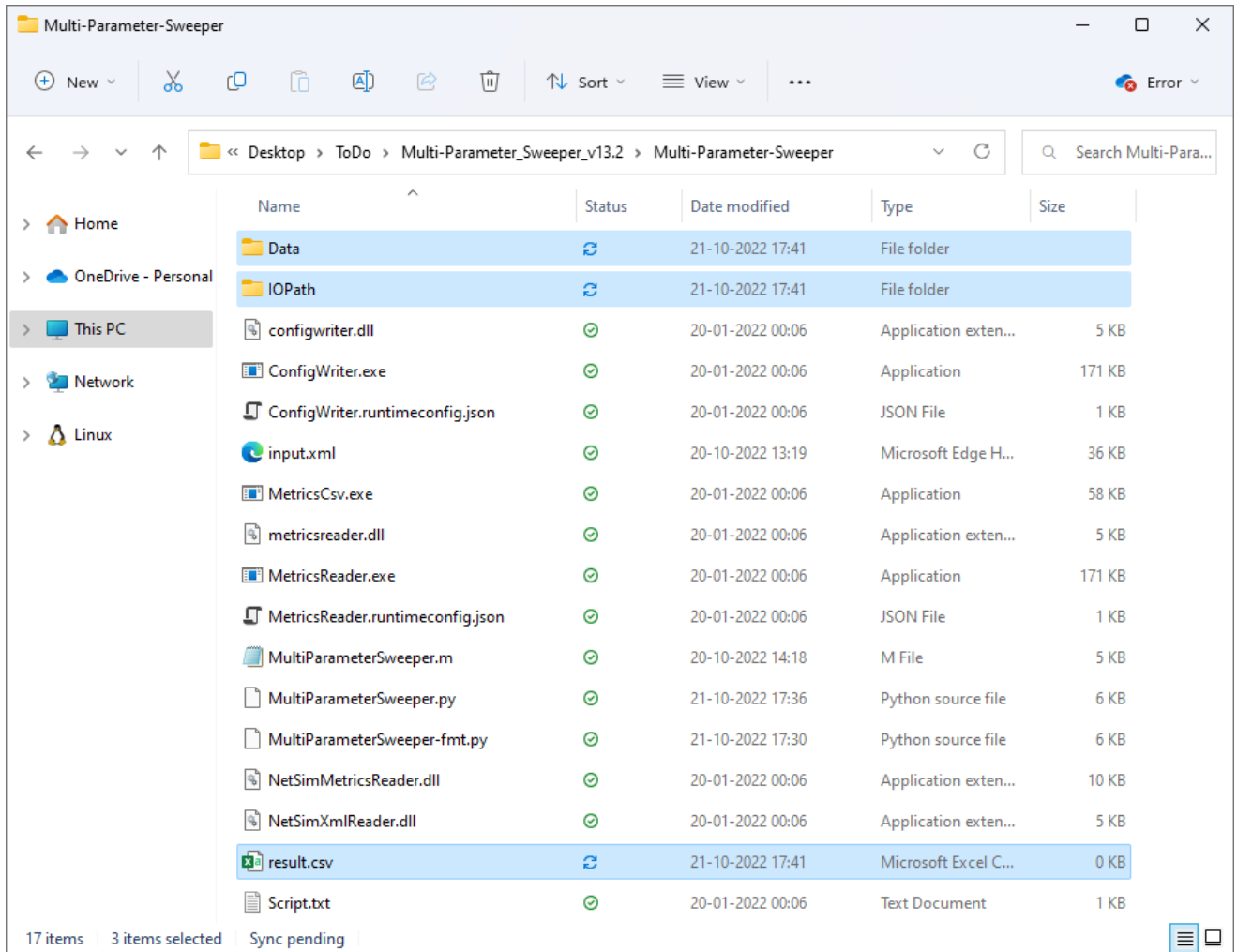
8. Multi-Parameter Sweeping process can be started from MATLAB command line in the directory of the Multi-Parameter-Sweeping project as shown below:



**Figure 8:** Running MATLAB script using cmd prompt

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

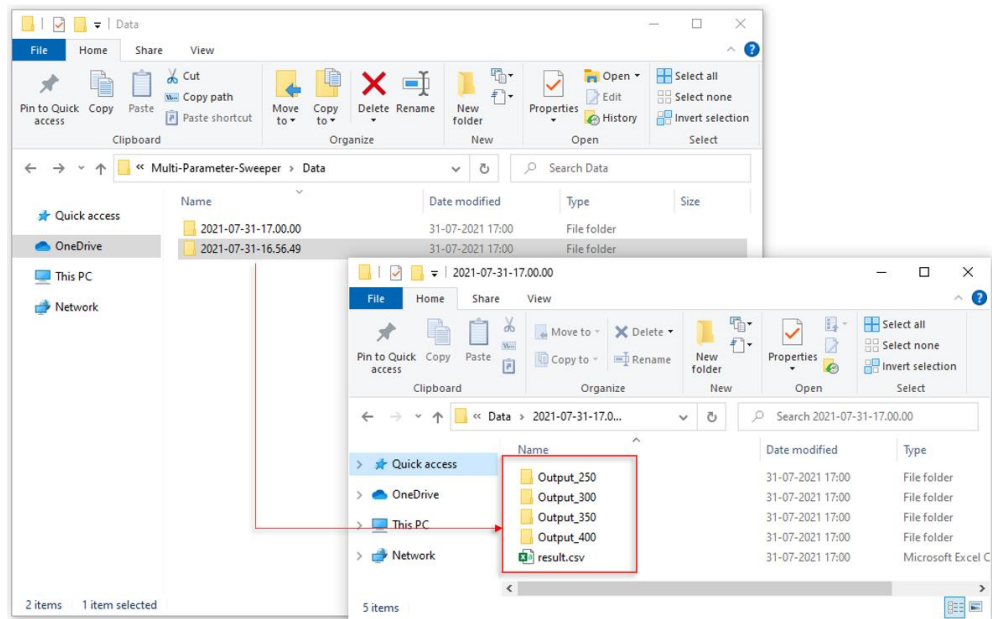
10. At the end of the process the Multi-Parameter-Sweeping folder will have the following file and folders created:



**Figure 9:** After Simulation Multi-Parameter-Sweeping folder contains output files like result.scv, Data etc

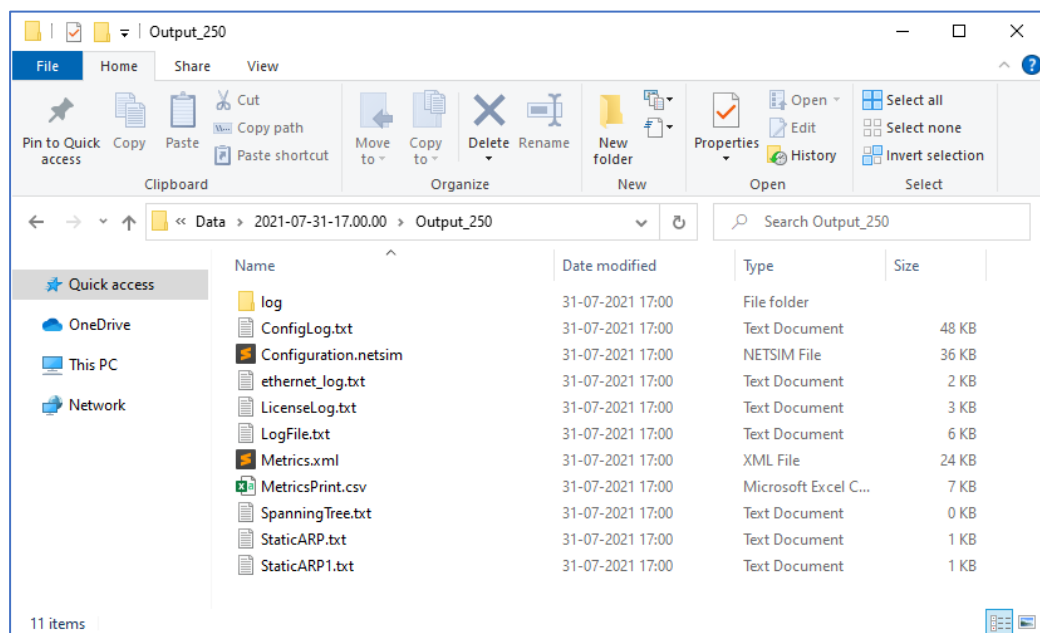
- **Data:** Contains multiple folders created based on date and time of simulation inside which multiple output folders corresponding to each simulation run, with its name including the value of the parameters in that iteration gets created.





**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.

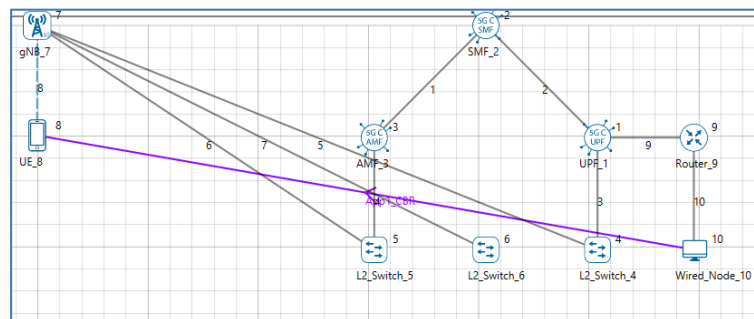
The screenshot shows a Microsoft Excel window titled 'result.csv - Microsoft Excel'. The table contains the following data:

	A	B	C	D	E
1	Y	THROUGHPUT(Mbps)			
2	250	4192.1856			
3	300	4192.1856			
4	350	2278.5344			
5	400	1352.7776			
6					
7					
8					
9					

**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 network 5G network scenario in NetSim, comprising of a Wired Node, Router, gNB and a UE.



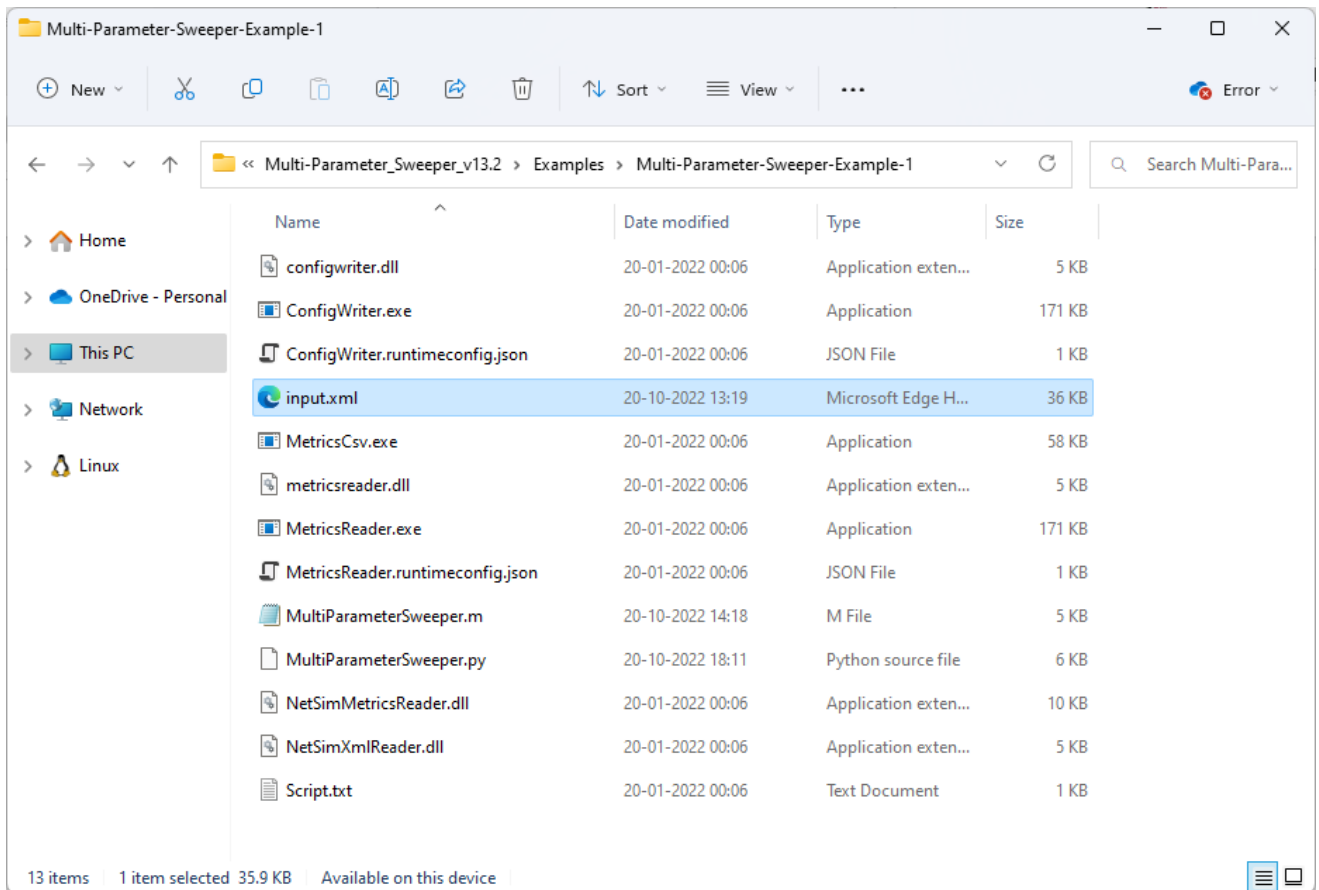
**Figure 13:** Network Topology

The network configuration has the initial distance between the gNB and UE as 50 meters with the gNB located at (500,0) and UE located at (500,500).

Multi-Parameter Sweeper is configured to run simulations for different distance between the gNB and UE by varying the UE Y coordinate value from 500 to 2000 in steps of 500 meters.

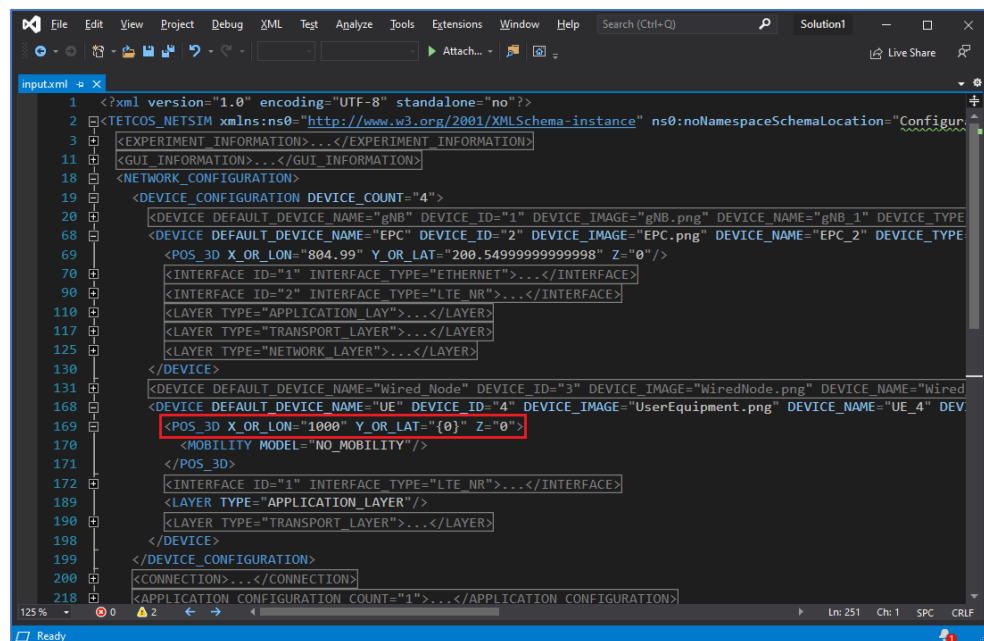
The network scenario is saved and the content of the Configuration.netsim file is copied to the Multi-Parameter-Sweeper directory and renamed as input.xml.

Refer to the Example 1 directory which is part of the project folder (Multi-Parameter-Sweeper\_v13.1\Examples\Multi-Parameter-Sweeper-Example-1)



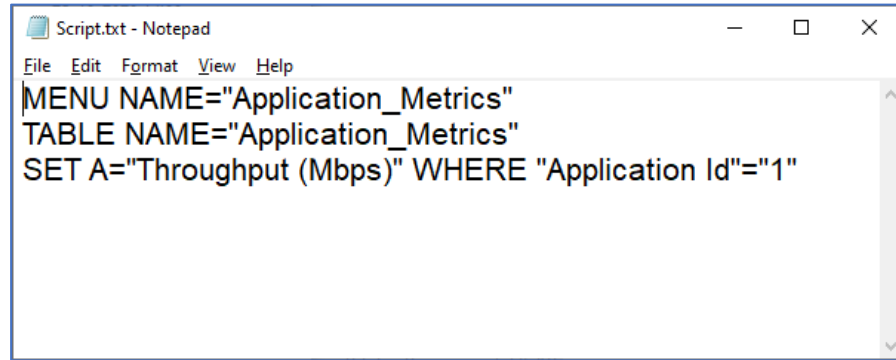
**Figure 14:** Renamed Configuration.netsim to input.xml and pasted in Multi-Parameter-Sweeper directory

1. The value of the Y coordinate of UE 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 UE 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.bat batch script modified for running simulations for different values of Y coordinates starting from 500 up to 2000 in steps of 500 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 500 to 2000 in steps of 500.
- 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.netstim 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 500 up to 2000 in steps of 500 is shown below:

```

1 % Set the path of 64 bit NetSim Binaries to be used for simulation.
2 NETSIM_PATH = "C:\Users\HP\OneDrive\Desktop\ToDo\Multi-Parameter-Sweeper_v13.2\Examples\Multi-Parameter-Sweeper-Example-1\MultiParameterSweeper.m";
3 LICENSE_ARG = "5053@192.168.0.9"; %Floating license
4 %LICENSE_ARG = ""C:\Program Files\NetSim\Standard_v13_2\bin""; %Node Locked license
5
6 % Set NETSIM_AUTO environment variable to avoid keyboard interrupt at the end of each simulation
7 setenv('NETSIM_AUTO','1');
8
9 % Create IOPATH directory to store the input Configuration.netsim file and
10 % the simulation output files during each iteration
11 if(~isfolder('IOPATH'))
12     mkdir IOPATH
13 end
14
15 % Create Data directory to store the Configuration.netsim and the
16 % Metrics.xml files associated with each iteration
17 if(~isfolder('Data'))
18     mkdir Data
19 end
20
21 % Clear the IOPATH folder if it has any files created during previous
22 delete('IOPATH'*);
23
24 % Delete result.csv file if it already exists
25 if(isfile('result.csv'))
26     delete result.csv
27 end
28
29 % create a csv file to log the output metrics for analysis csvfile
30 fileid=fopen('result.csv','w');
31
32 % Add headings to the CSV file
33 fprintf(fileid,'Y,Throughput(Mbps)');
34 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).

```

39
40 % Iterate based on the number of time simulation needs to be run and the
41 for i = 500:500:2000
42     if(isfile('Configuration.netsim'))
43         delete Configuration.netsim
44     end
45
46     if(isfile('IOPATH\Configuration.netsim'))
47         delete('IOPATH\Configuration.netsim')
48     end
49
50     if(isfile('IOPATH\Metrics.xml'))
51         delete('IOPATH\Metrics.xml')
52     end
53
54 %Call ConfigWriter.exe with arguments as per the number of variable parameters in the input.xml file
55 cmd="ConfigWriter.exe "+string(i);
56 disp(cmd);
57 system(cmd);
58
59 %Copy the Configuration.netsim file generated by ConfigWriter.exe to IOPATH directory
60 if(isfile('Configuration.netsim'))
61     copyfile('Configuration.netsim','IOPATH');
62 end
63
64 strIOPATH=pwd+'IOPATH';
65
66 %Run NetSim via CLI mode by passing the apppath iopath and license information to the NetSimCore.exe
67 cmd="start ""NetSim_Multi_Parameter_Sweeper"" /wait /d " strIOPATH;
68
69

```

**Figure 18:** Varying Distance and set license server information

- For loop is set to iteratively run simulations for values starting from 500 to 2000 in steps of 500.
- 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.

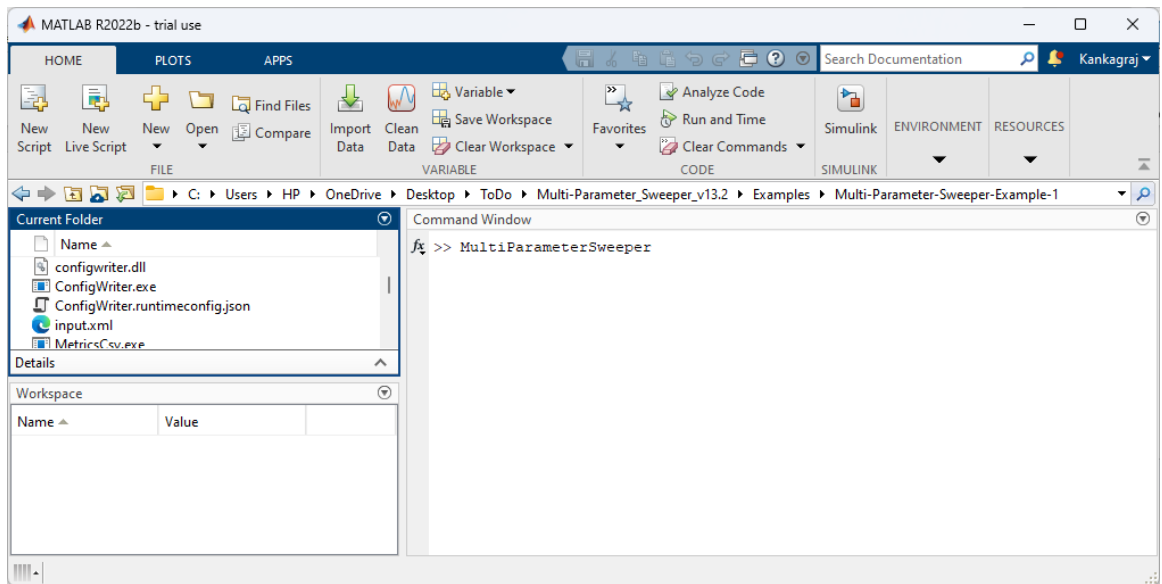
```

94 %changes required starts
95 if(isfile('IOPATH\Metrics.xml'))
96
97     fileid=fopen('result.csv','a');
98     if(fileid)
99         fprintf(fileid,'\n%d',i);
100     fclose(fileid);
101 end
102 if(OUTPUT_PARAM_COUNT==1)
103     system('MetricsReader.exe result.csv');
104 else
105     for n = 1 : OUTPUT_PARAM_COUNT
106         filename='Script'+string(n)+'.txt';
107         movefile(filename,'Script.txt');
108         system('MetricsReader.exe result.csv');
109         fileid=fopen('result.csv','a');
110         if(fileid)
111             fprintf(fileid,'\n');
112             fclose(fileid);
113         end
114         movefile('Script.txt',filename);
115     end
116 else
117 %Update the output Metric as crash if Metrics.xml file is missing
118 fileid=fopen('result.csv','a');
119 if(fileid)
120     fprintf(fileid,'\n%d,crash',i);
121     fclose(fileid);
122 end
123 end
124
125 % Name of the Output folder to which the result will be saved
126 OUTPUT_PATH = 'Data/'+today+'Output_'+string(i);
127

```

**Figure 19:** Modify parameters in MultiParameterSweeper.m

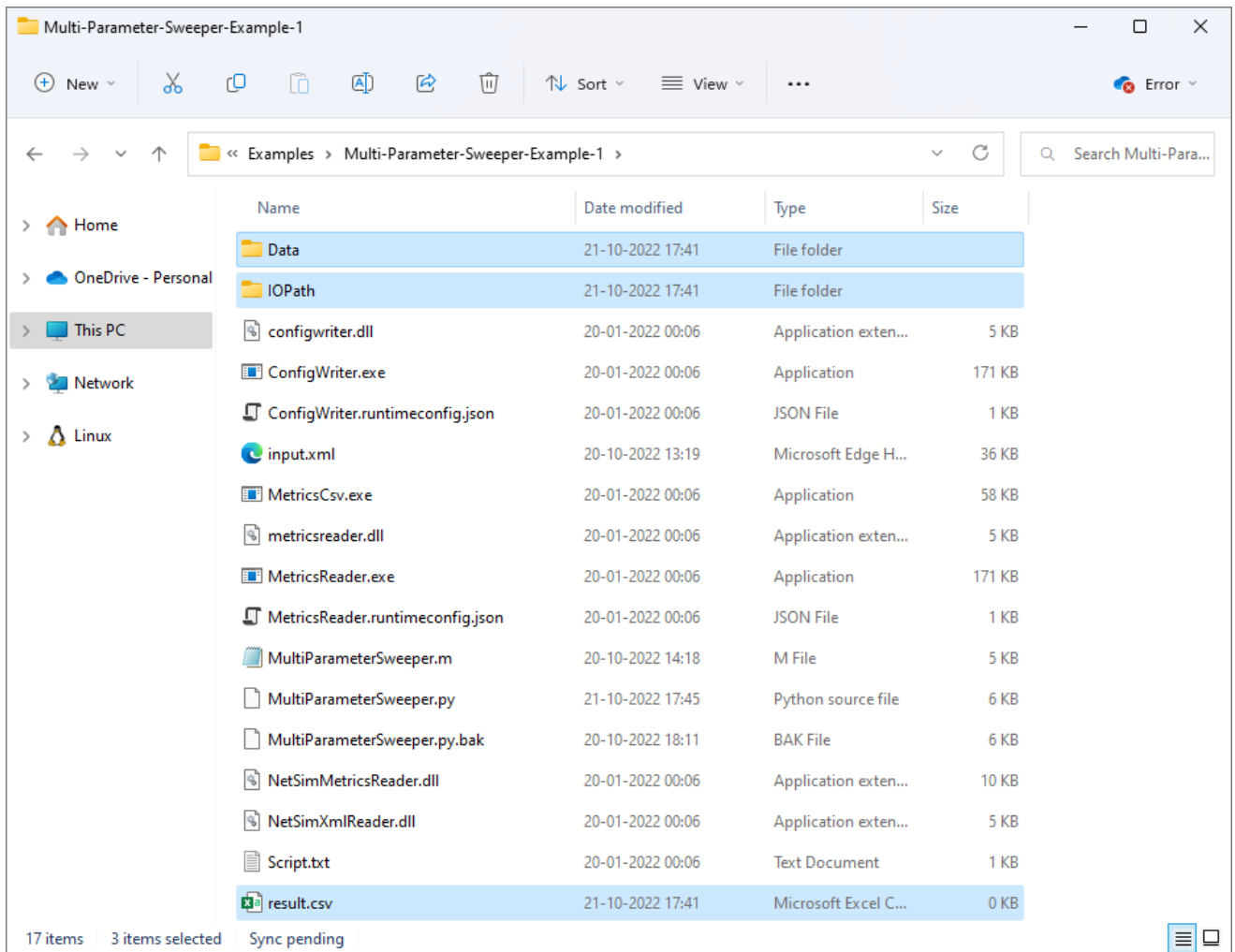
- The value of the parameter Y in the current iteration is written to the result log file for analysis.
  - Configuration file and Metrics file are copied and renamed appending the value of the parameter in the current iteration.
4. Multi-Parameter Sweeping process can be started from MATLAB command line in the directory of the Multi-Parameter-Sweeping project as shown below:



**Figure 20:** Running MATLAB script using cmd prompt

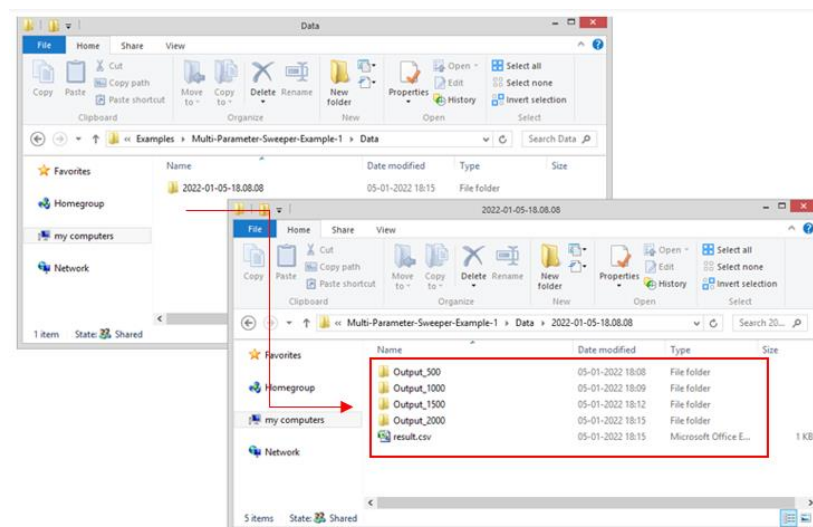
This starts the Multi-Parameter-Sweeping process which runs NetSim simulations iteratively for different values of Y parameter of UE.

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

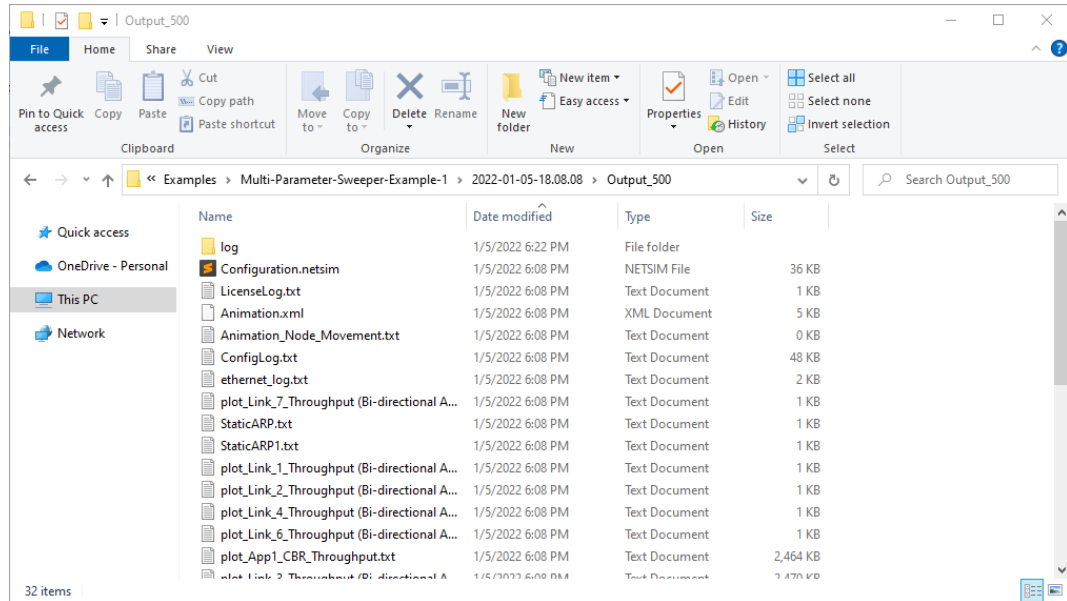
- **Data:** Contains multiple folders corresponding to each simulation run, with its name including the value of the parameters in that iteration. The output folders will be created inside folder with name in the format Year-Month-Day-Hours-Minutes-Seconds.





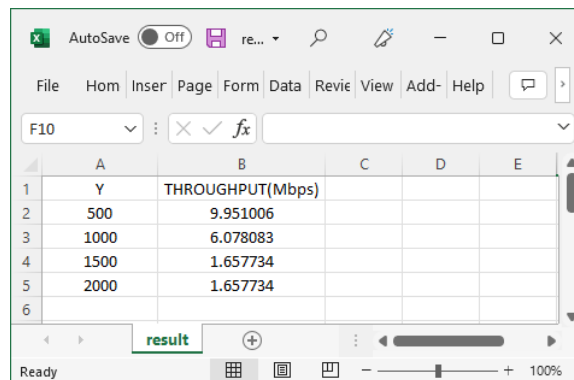
**Figure 22:** Based on distance Configuration.netsim files created in output folder

- 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

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**

1. Consider the following network 5G network scenario in NetSim, comprising of a Wired Node, Router, gNB and a UE.



2. Properties configured in the LTE\_NR interface of the gNB is shown in the table below:

Interface(5G_RAN) Properties	
Tx_Power(dBM)	40
Tx_Antenna_Count	8
Rx_Antenna_Count	4
CA_Type	Single Band
CA_Configuration	n78
CA_Count	1
Numerology	0
Channel Bandwidth (MHz)	10
PRB Count	52
MCS Table	QAM64
CQI Table	Table 1
X_Overhead	XOH0
DL UL Ratio	4:1
Outdoor Scenario	Rural Macro
LOS Mode	Standard
Wireless Link Properties	
Channel Characteristics	No_Pathloss
Wired Link Properties	
Link Speed (Mbps)	10000
BER	0
Propagation Delay ( $\mu$ s)	0
Application Properties	
Packet Size (Byte)	1460
Inter Arrival Time ( $\mu$ s)	166
Generation Rate (Mbps)	100
Transport Control	UDP
Start Time (s)	1
QoS	BE
Simulation Parameters	
Simulation Time (s)	1.1

**Table 1:** gNB Properties

3. Traffic is generated at a rate of 70 Mbps and upon running simulation, the throughput achieved is 59.95 Mbps.
4. We now find the max throughput for each possible bandwidth; Tx Antenna count and Rx Antenna count combination varying the generation rate based accordingly.
5. Two more parameters to be taken care include, the PRB Count and Guard Band (KHz) which vary with respect to the bandwidth.

Input Variables	Value Range
Channel Bandwidth (MHz)	10,15,20,25,30,40,50
Tx_Antenna_Count	1,2,4,8,16,32,64,128
Rx_Antenna_Count	1,2,4,8,16
PRB Count	52,79,106,133,160,216,270
Guard Band (KHz)	312.5,382.5,452.5,522.5,592.5,552.5,692.5
Reference Inter Arrival Time (Microseconds)	166
Reference Bandwidth	10
Reference DL MIMO Layer Count	2

**Table 2:** Input variable values

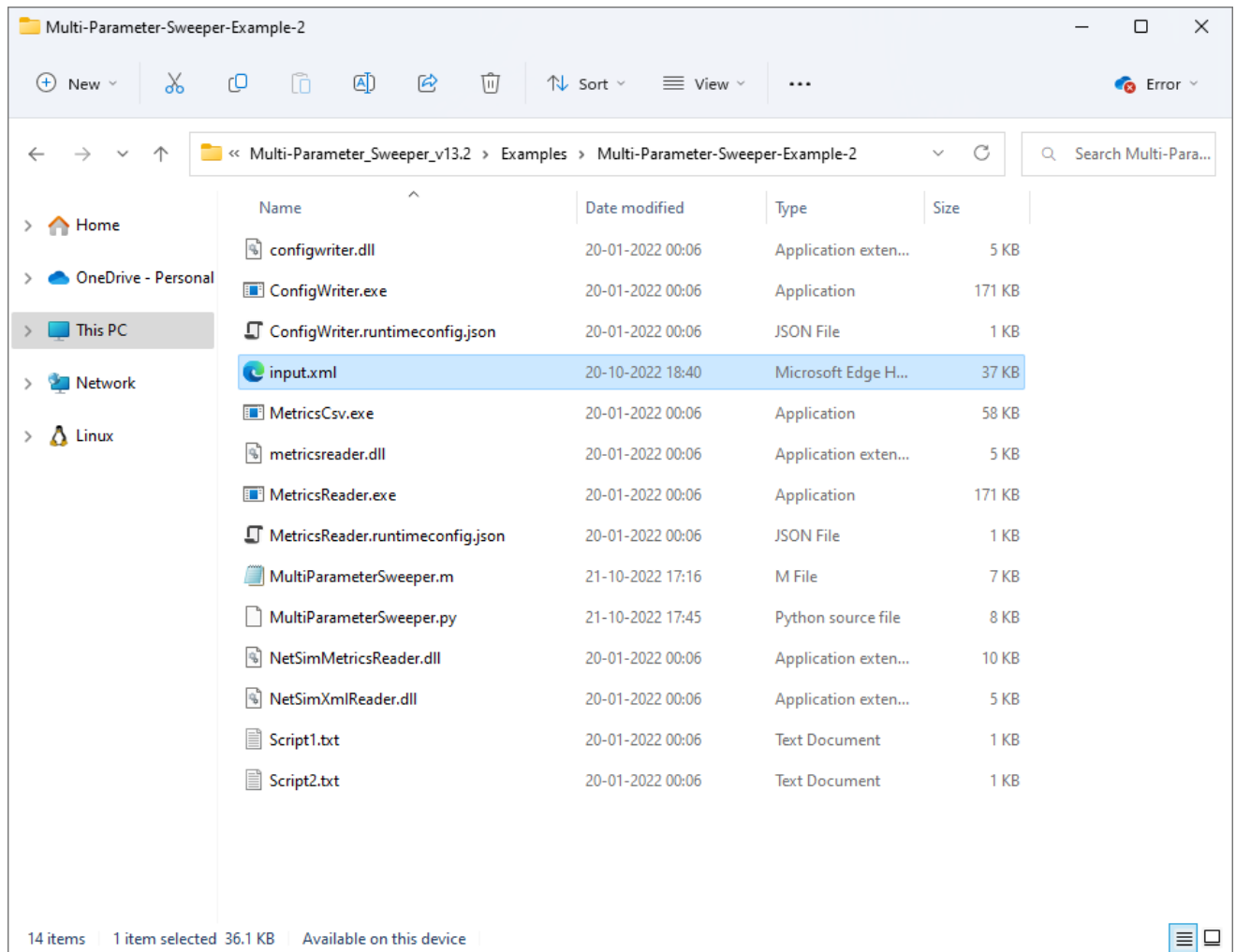
6. Inter Arrival Time for each case is calculated based on the Reference IAT Bandwidth and DL MIMO Layer Count as shown below:

$$Inter\ Arrival\ Time\ (Micro\ Seconds) = \frac{Ref\ IAT}{\left(\frac{Curr\ BW}{Ref\ BW}\right) * \left(\frac{Curr\ DL\ MIMO\ Count}{Ref\ DL\ MIMO\ Count}\right)}$$

For E.g. In case of Bandwidth of 20 MHz and DL MIMO Count of 4 inter arrival time is

$$Inter\ Arrival\ Time\ (Micro\ Seconds) = \frac{166}{\left(\frac{20}{10}\right) * \left(\frac{4}{2}\right)} = 41.5\ Mbps$$

7. 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.
8. Refer to the Example 2 directory which is part of the project folder (Multi-Parameter-Sweeper\_v13.1\Examples\Multi-Parameter-Sweeper-Example-2).

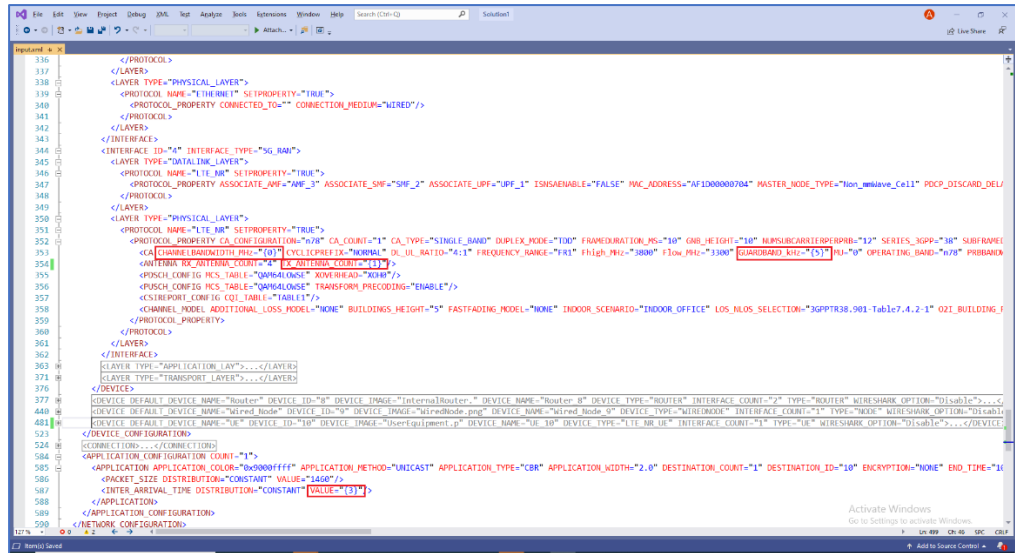


**Figure 27:** Renamed Configuration.netsim to input.xml and pasted in Multi-Parameter-Sweeper directory

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

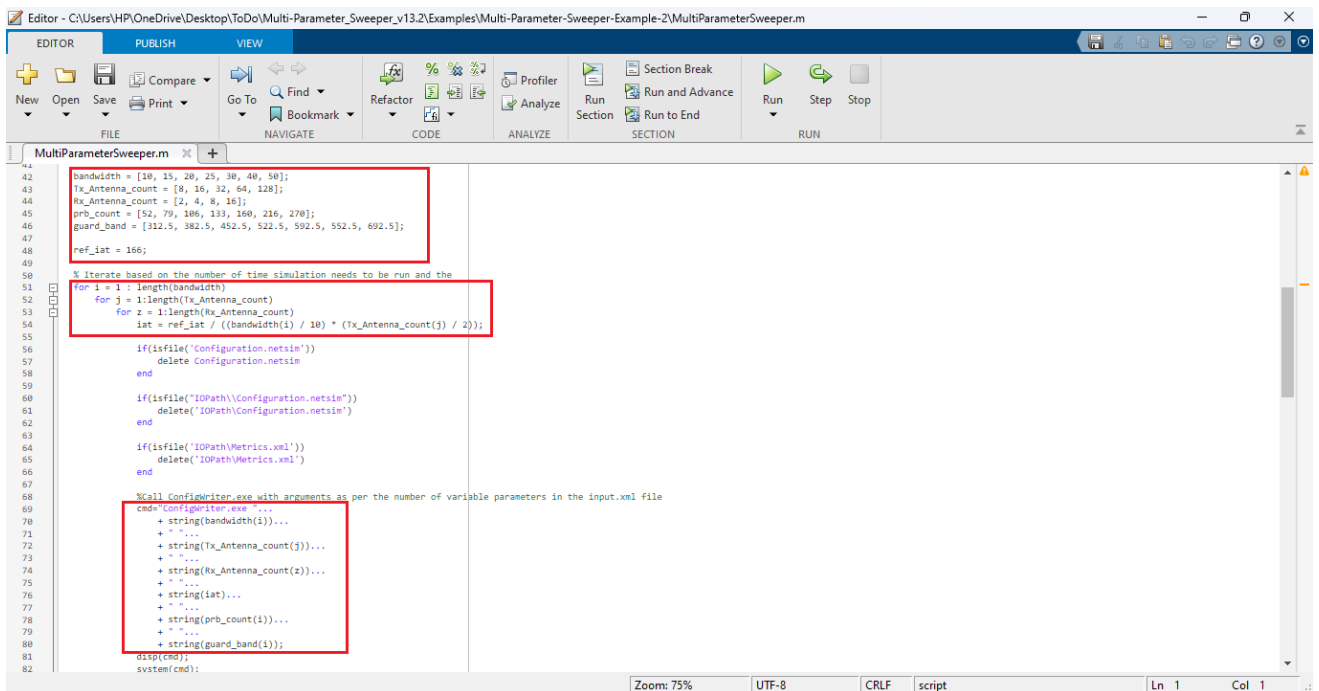
Input Variables	
Channel Bandwidth (MHz)	{0}
Tx Antenna Count	{1}
Rx Antenna Count	{2}
Inter Arrival Time (Microseconds)	{3}
PRB Count	{4}
Guard Band (KHz)	{5}

**Table 3:** Variables are modified to the input.xml file



**Figure 28:** The above table 3 Variables are modified in input.xml file

10. 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 Guard Band, PRB Count and the IAT that is calculated.



**Figure 29:** Modified MultiParameterSweeper.m based on input parameter

11. 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

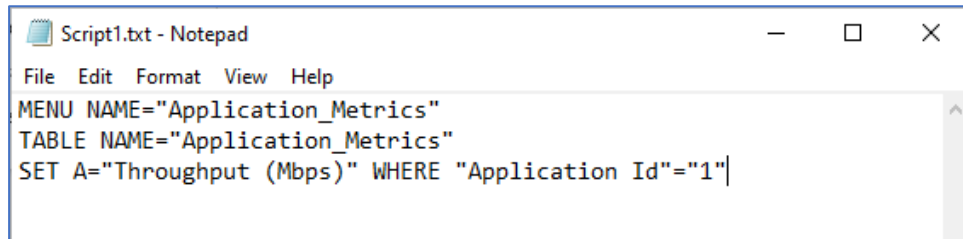
Throughput (Mbps)

## Data Packets transmitted

**Table 4:** User need to modify these two parameters in Script files

- 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.

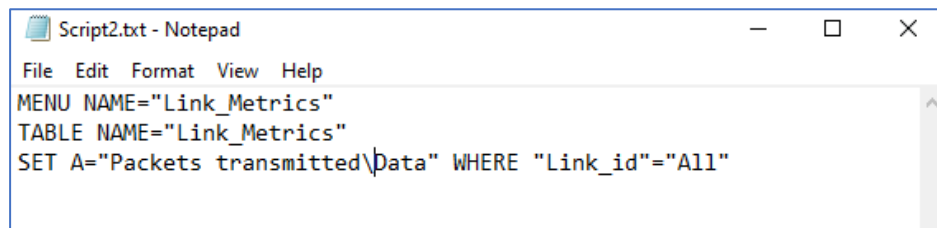
### Script1.txt



```
Script1.txt - Notepad
File Edit Format View Help
MENU NAME="Application_Metrics"
TABLE NAME="Application_Metrics"
SET A="Throughput (Mbps)" WHERE "Application Id"="1"
```

**Figure 30:** The application throughput is to be logged for each simulation modified in Script1.txt

### Script2.txt



```
Script2.txt - Notepad
File Edit Format View Help
MENU NAME="Link_Metrics"
TABLE NAME="Link_Metrics"
SET A="Packets transmitted\Data" WHERE "Link_id"="All"
```

**Figure 31:** The Data Packets transmitted is to be logged for each simulation modified in Script2.txt

- 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.

```

115 OUTPUT_PARAM_COUNT=2;
116
117 %changes required starts
118 if(isfile('IOPATH\Metrics.xml'))
119
120     fileid=fopen('result.csv','a');
121     if(fileid)
122         fprintf(fileid,'\n%d,%d,%d,%f',bandwidth(i),Tx_Antenna_count(j),Rx_Antenna_count(z),iat);
123         fclose(fileid);
124     end
125     if(OUTPUT_PARAM_COUNT==1)
126         system("MetricsReader.exe result.csv");
127     else
128         for n = 1 : OUTPUT_PARAM_COUNT
129             filename='Script'+string(n)+'.txt';
130             movefile(filename,'Script.txt');
131             system("MetricsReader.exe result.csv");
132             fileid=fopen('result.csv','a');
133             if(fileid)
134                 fprintf(fileid,',');
135                 fclose(fileid);
136             end
137             movefile('Script.txt',filename);
138         end
139     end
140 else
141     %Update the output Metric as crash if Metrics.xml file is missing
142     fileid=fopen('result.csv','a');
143     if(fileid)
144         fprintf(fileid,'\n%d,%d,%d,%f,crash',bandwidth(i),Tx_Antenna_count(j),Rx_Antenna_count(z),iat);
145         fclose(fileid);
146     end
147 end

```

**Figure 32:** Modify MATLAB script MultiParameterSweeper.m file

14. 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.

```

141 else
142     %Update the output Metric as crash if Metrics.xml file is missing
143     fileid=fopen('result.csv','a');
144     if(fileid)
145         fprintf(fileid,'\n%d,%d,%d,%f,crash',bandwidth(i),Tx_Antenna_count(j),Rx_Antenna_count(z),iat);
146         fclose(fileid);
147     end
148 end
149
150 % Name of the Output folder to which the result will be saved
151 OUTPUT_PATH = "Data\*today*\Output_"+string(bandwidth(i))+"_"+string(Tx_Antenna_count(j))+"_"+string(Rx_Antenna_count(z));
152
153 if(~isfolder(OUTPUT_PATH))
154     mkdir(OUTPUT_PATH)
155 end
156
157 % Create a copy of result.csv file present in sweep folder to date-time
158 if(isfile('result.csv'))
159     copyfile('result.csv', "Data\*today")
160 end
161
162 % Create a copy of all files that is present in IOPATH to the desired output location
163 movefile('IOPATH\*',OUTPUT_PATH)
164
165 % Delete Configuration.netsim file created during the last iteration
166 if(isfile("Configuration.netsim"))
167     delete('Configuration.netsim')
168 end
169
170 if(isfile('Metrics.xml'))
171     delete('Metrics.xml')
172 end
173 end
174

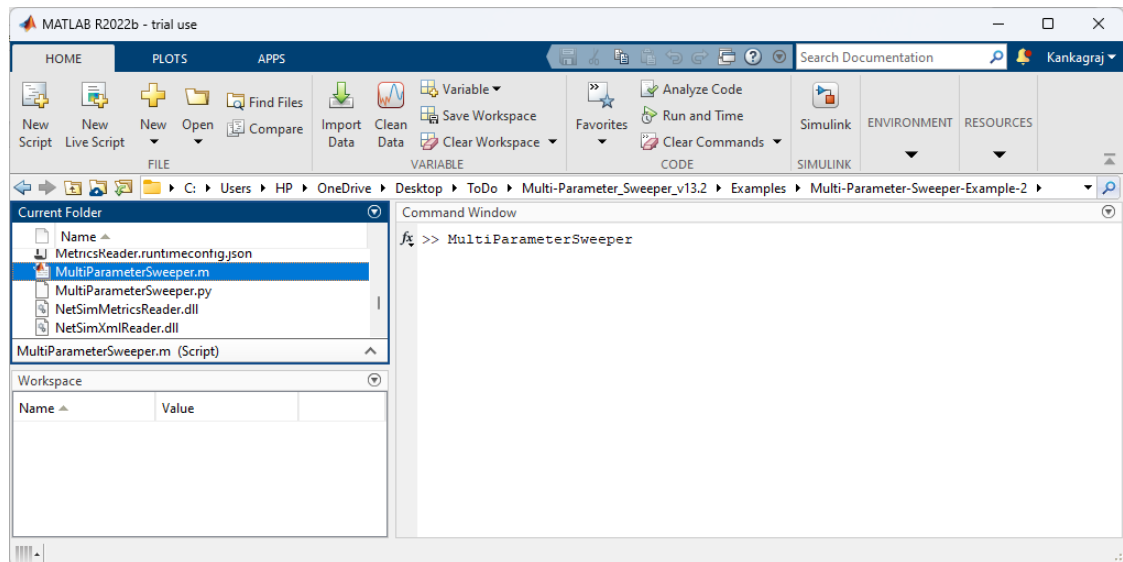
```

**Figure 33:** Modify MATLAB script MultiParameterSweeper.m file

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

**MATLAB Script:**

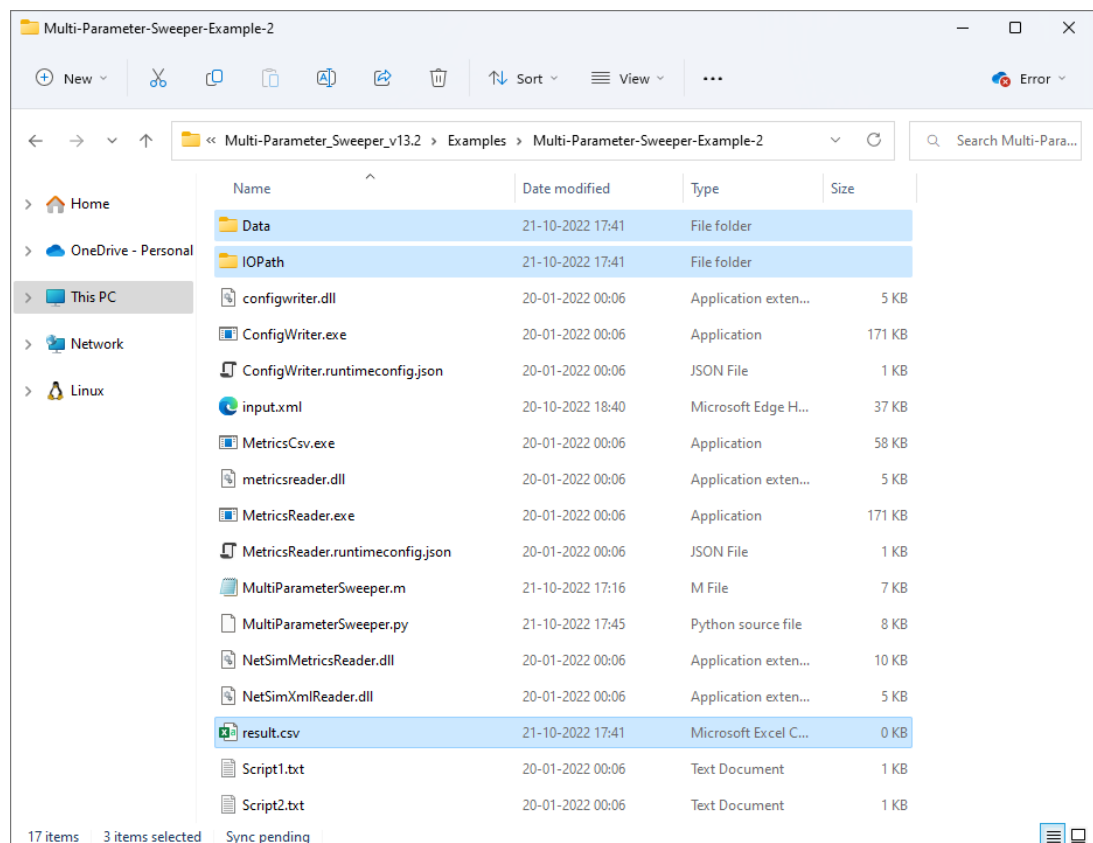




**Figure 34:** Running MATLAB Script using cmd Prompt

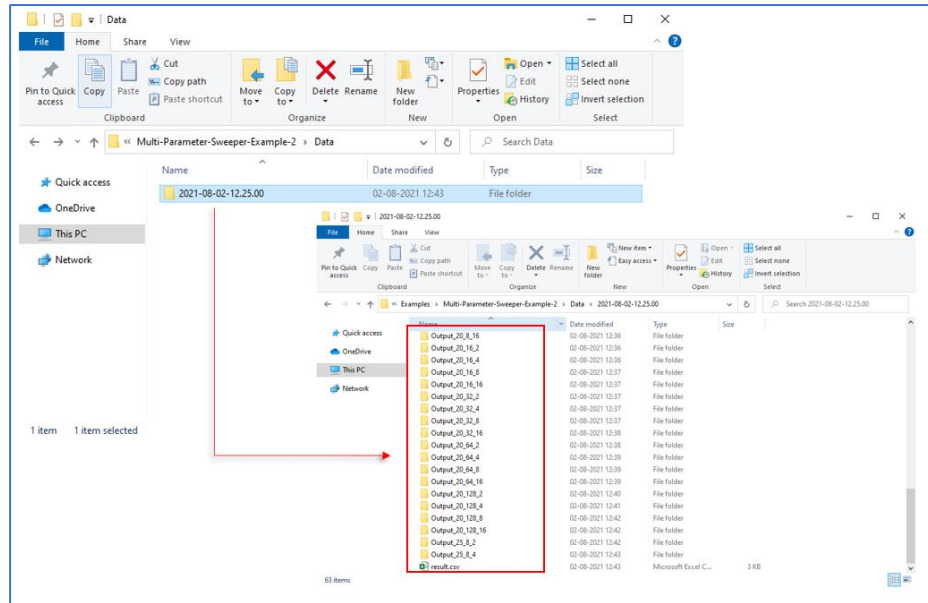
This starts the Multi-Parameter-Sweeping process which runs NetSim simulations iteratively for different combinations of input parameters.

At the end of the process the Multi-Parameter-Sweeping folder will have the following file and folders created:



**Figure 35:** After Simulation Multi-Parameter-Sweeping folder contains output files like result.csv, Data etc

16. **Data:** The Data directory contains multiple output folders with the output files associated with each simulation run.



```
C:\Users\HP\OneDrive\Documents\NetSim\Workspaces\Default_Workspace_v13_2_34\bin_x64\NetSimCore.exe
```

```
Today's date is "Oct 22 2022.14:56:23"  
Binary build date is "Oct 13 2022.20:14:17"  
  
NetSim License Manager will first check for node lock licenses.  
If not available, it will then check for floating/cloud licenses  
NetSim License Manager Start. Checking for licenses available (this may take upto 2 min) -  
No license for product (-1)  
  
NetSim License Manager Start. Checking for licenses available (this may take upto 2 min) -  
  
License Manager Output. Product>Edition>Maj_ver>Min_ver>Lic_type>Components>  
netsim>std>13>2>rlm_hw>111111111111>10000>  
+ [33m-[1m  
*****  
WARNING:  
Detected a change in following:  
libLTE_NR.dll  
This message is normally shown if users link their own code to NetSim.  
*****  
Press any key to continue....  
_
```

- 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.