5G/LTE based UAV Network Communication

(TR 36.777 pathloss)

Software: NetSim Standard v14.4 (64 bit), Visual Studio 2022, MATLAB R2020b or higher, Simulink, Robotics and System Toolbox.

Project Download Link:

https://github.com/NetSim-TETCOS/5G-LTE-UAV-v14.4/archive/refs/heads/main.zip

Follow the instructions specified in the following link to download and set up the Project in NetSim:

https://support.tetcos.com/en/support/solutions/articles/14000128666-downloading-and-setting-up-netsim-file-exchange-projects

Objective

Performance analysis of UAV - Base station network communications. This involves interfacing NetSim and the UAV toolbox of MATLAB.

- For Each UE in NetSim, a UAV is instantiated in MATLAB as per the UE (UAV) ID.
- MATLAB calculates the flight path and passes the Mobility information to NetSim.
- In NetSim UE movement is modeled as per MATLAB UAV co-ordinates
- Pathloss calculations done in NetSim per 36.777.

NetSim UAV Simulation model

UAV Takes and moves away from eNB

UAV flies in a circular pattern over the eNB

Pathloss calculations done in NetSim per 36.777 for Rural Macro

$$PL_{RMa-AV-LOS} = max(23.9 - 1.8\,log_{10}(h_{UT}), 20)\,log_{10}(d_{3D}) + 20\,log_{10}\left(\frac{40\pi f_c}{3}\right)$$

Figure 1: NetSim UAV Simulation model

Code modifications

Files attached and their modifications are explained below.

1. LTENR_PropagationModel.c - Added LTE-UAV propagation model for Rural macro as per

3GPP 36.777 Standard.

2. UAVBasedMobility.c - A newly created added to the Mobility Project. The interface with MATLAB happens here to get coordinates from MATLAB and to pass it to NetSim.
3. OpSimulink.m - To initiate Simulink.

NetSim - Simulink Interfacing

Upon interfacing NetSim with MATLAB the following tasks are performed during the simulation start:

- MATLAB Engine process is initialized.
- MATLAB Desktop window is loaded.
- SIMULINK Model is loaded

Upon simulating a network created in NetSim the following tasks are performed periodically:

- SIMULINK Model is simulated
- SIMULINK Model is paused
- NetSim reads the data generated by SIMULINK from the MATLAB workspace.
- Appends the readings to the packet payload as packets are formed.

During the Simulation, the SIMULINK Model is started and paused several times for NetSim and SIMULINK simulations to run synchronously. The X, Y and Z coordinates obtained from SIMULINK are read from the MATLAB workspace and given as input to NetSim's Mobility model. In this example, coordinates are taken every second and updated to the device’s mobility.

Output/Metrics specific to this example

- Mobility of the devices configured in NetSim is given as input from MATLAB.
- Pathloss is implemented as per 36.777 Standard.
- NetSim Plot Results: Throughput analysis as the UAV moves away from the gNB

Modifications done to NetSim Source codes

Project: Mobility

Files:

- Mobility.c,
- Mobility.h,
- Added - UAVBasedMobility.c
- Mobility.vcxproj (Project file)

Sections of source code modified

- **Mobility.c**
    - **fn_NetSim_Mobility_Init():** call to **init_uav()** function
    - **fn_NetSim_Mobility_Run()**: Call to **uav_run()** function
    - **fn_NetSim_Mobility_Finish()**

- **Mobility.h**
    - MATLAB Engine variable - Used to initiate and interact with the MATLAB Engine process.
- **Mobility.vcxproj** - This is a Visual Studio project file that loads and manages the source codes related to Mobility in NetSim.
    - path to MATLAB application
    - path to MATLAB include directory.
    - path to MATLAB lib directory
    - information related to dependent MATLAB library files.
- **UAVBasedMobility.c**
    - **init_uav()**: Initializes MATLAB, Loads SIMULINK Model, starts and pauses SIMULINK simulation, and initializes the UAV devices in MATLAB to start simulation along with NetSim's simulation.
    - **uav_run()**: Starts NetSim and MATLAB simulation simultaneously and gets the coordinates from MATLAB workspace for every step size set in NetSim.
    - **uavcorr()**: Function to get coordinates from MATLAB.

Steps to simulate

1. In Control panel open>system>Advanced system settings>Edit the system environment variable>environment variables
2. Add the following MATLAB install directory path in the Environment PATH variable
    a. <MATLAB_INSTALL_DIRECTORY>\bin\win64
    b. For e.g.: C:\Program Files\MATLAB\R2020b\bin\win64

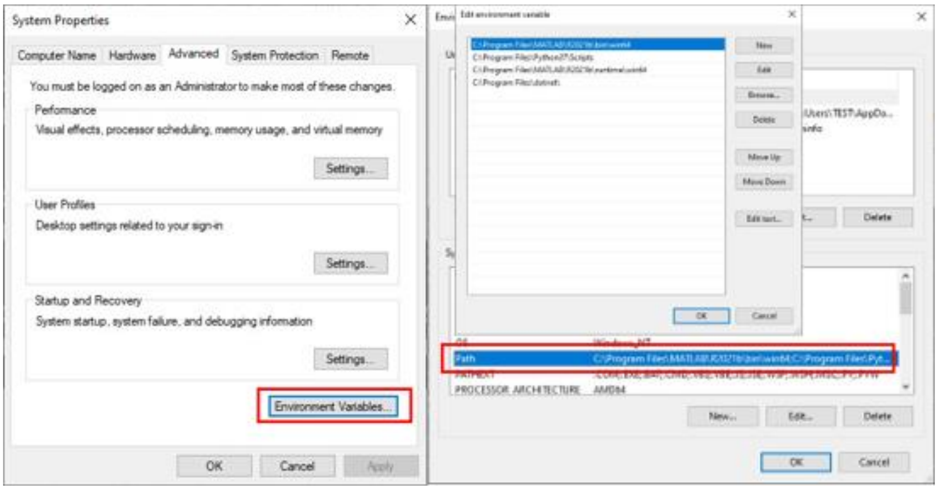


Figure 2: Environment PATH variable

Note: If the machine has more than one MATLAB installed, the directory for the target platform must be ahead of any other MATLAB directory (for instance, when compiling a 64-bit application, the directory in the MATLAB 64-bit installation must be the first one on the PATH).

1. Open the Command prompt as admin and execute the command “Matlab -regserver”. This will

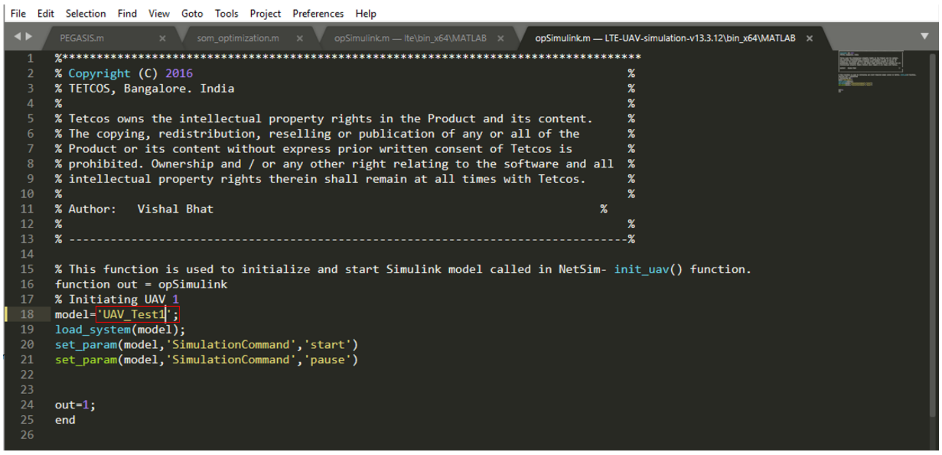
register MATLAB as a COM automation server and is required for NetSim to start the MATLAB automation server during runtime.

2. Copy the **UE.xml** file located in the NetSim_Files folder of the downloaded directory to the <NetSim Installation Directory>\Docs\UI_xml\Device_Properties folder.
3. Open the Source codes in Visual Studio by going to Your Work-> Workspace and Clicking on Open code.
4. Under the Mobility project in the solution explorer, you will see that and UAV_basedmobility.c files which contain source codes related to interactions with MATLAB and handling clustering in handling NetSim respectively.

Examples

1. The **5G-LTE-UAV-v14.4** workspace comes with a sample configuration that is already saved. To open this example, go to the Your Work option in the Home Screen of NetSim and click on the **5G_UAV_Example** or the **LTE_UAV_Example** from the list of experiments.
2. The project support two types of UAV mobility patterns:
    a. Case 1: UAV/Drone take-off - UAV_Test1
    b. Case 2: UAV/Drone circular pattern – UAV_Test2

   By default, the model is set to case 1. You can switch the model from the opsimulink.m file which is present in the workspace ..\5G-LTE-UAV-v14.4\bin_x64\MATLAB as shown below:



Uncomment the case-1 **UAV_Test 1** section while running case 1 and Comment on the Case-2 **UAV_Test 2** as mentioned below, then rebuild the project.

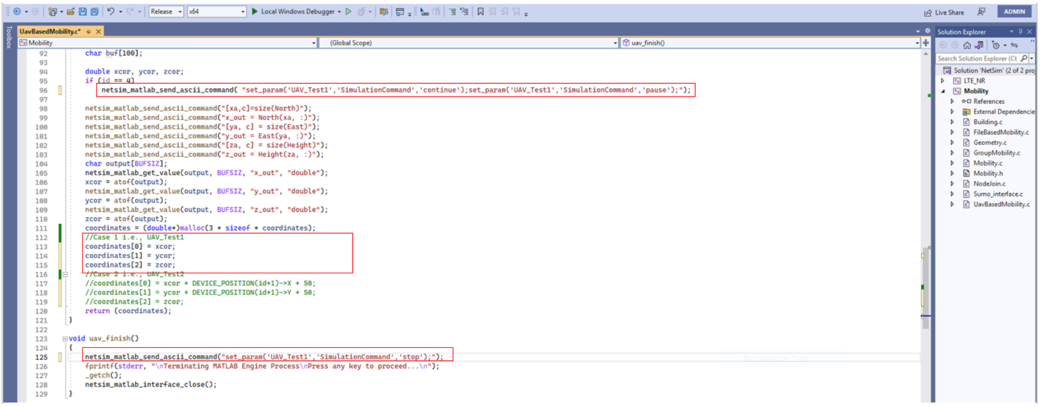


Figure 3: Source code changes for Case 1 in UAVBasedmobility.c

3. Run the simulation and press any key to continue.

4. It will open the MatlabInterface.exe console window. You will observe that as the simulation starts in NetSim, MATLAB gets initialized and UAV Animation for all the devices in NetSim gets initialized as the simulation runs there will be continuous interaction between NetSim and MATLAB where the X, Y, and Z co-ordinates will be updated in NetSim from MATLAB.

<u>5G_UAV_Example</u>

Case 1: UAV Drone Take off - Application throughput variation with Drone mobility.

NetSim Plot: Application Throughput vs. Time

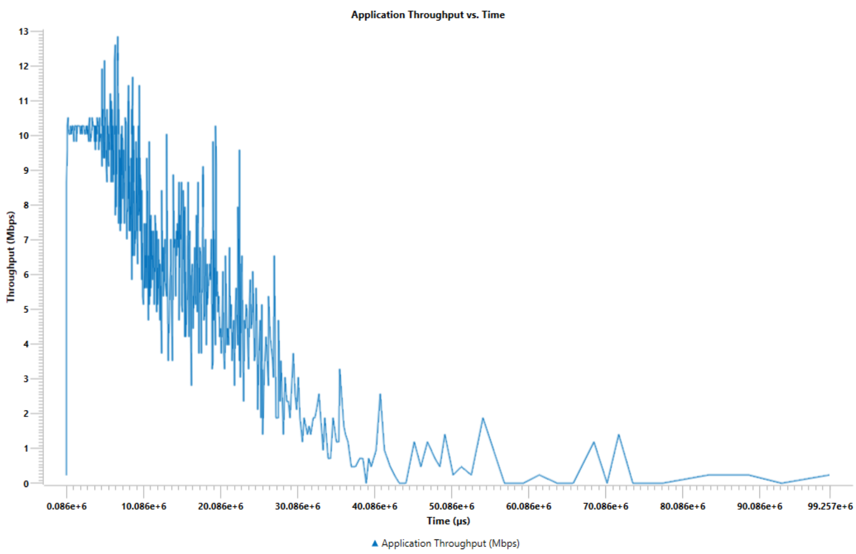


Figure 4: Application Throughput vs. Time from NetSim

MATLAB Flight Path

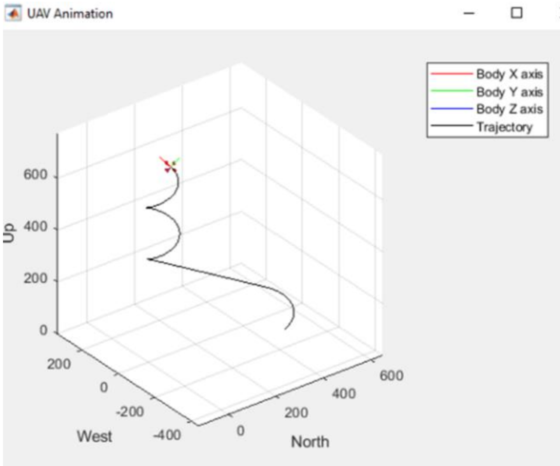
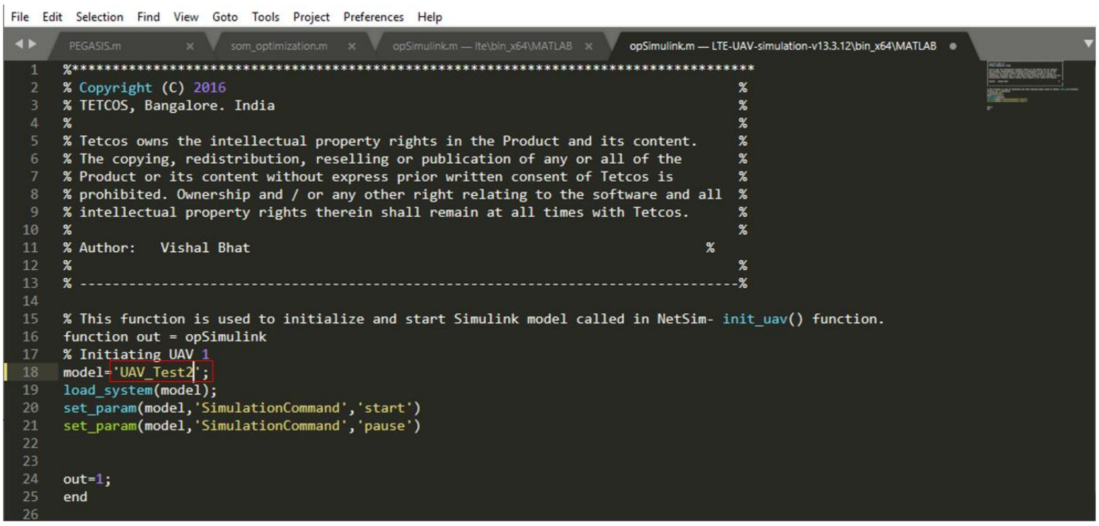


Figure 5: MATLAB Flight Path

Case 2: UAV Drone rotation around LTE-eNB

5. Before running Case 2 change the file name in the source code in UAV_Basedmobility.c as mentioned below. change the Model name in the opsimulink.m file which is present in the workspace inside the bin 64 folder.

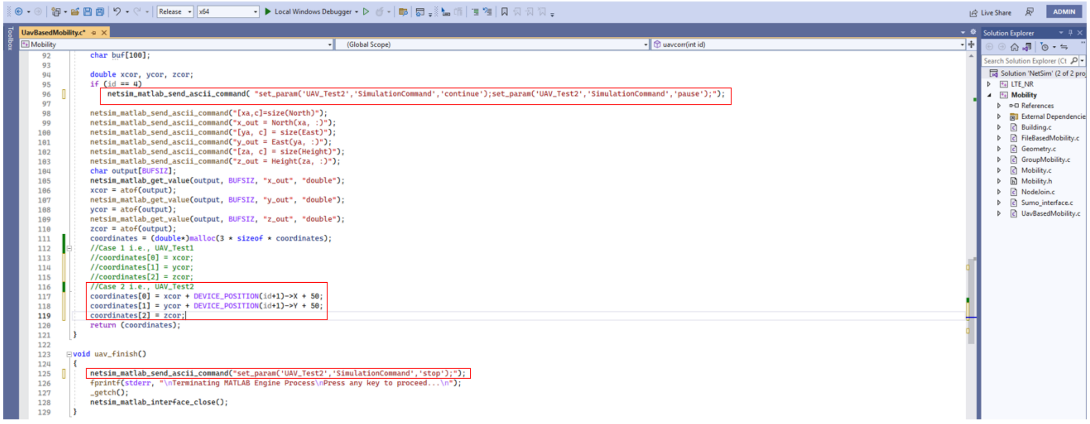



```
%*********************************************************************************
% Copyright (C) 2016                                                            %
% TETCOS, Bangalore. India                                                      %
%                                                                               %
% Tetcos owns the intellectual property rights in the Product and its content.  %
% The copying, redistribution, reselling or publication of any or all of the    %
% Product or its content without express prior written consent of Tetcos is    %
% prohibited. Ownership and / or any other right relating to the software and all %
% intellectual property rights therein shall remain at all times with Tetcos.   %
%                                                                               %
% Author:    Vishal Bhat                                                    %
%                                                                               %
% ------------------------------------------------------------------------------%

% This function is used to initialize and start Simulink model called in NetSim- init_uav() function.
function out = opSimulink
% Initiating UAV 1
model='UAV_Test2';
load_system(model);
set_param(model,'SimulationCommand','start')
set_param(model,'SimulationCommand','pause')


out=1;
end
```

6. Uncomment the case-2 **UAV_Test 2** section while running case 2 and Comment on the Case-1 **UAV_Test 1** as mentioned below.



```c
    char buf[100];

    double xcor, ycor, zcor;
    if (id == 4)
        netsim_matlab_send_ascii_command( "set_param('UAV_Test2','SimulationCommand','continue');set_param('UAV_Test2','SimulationCommand','pause');");

    netsim_matlab_send_ascii_command("[xa,c]=size(North)");
    netsim_matlab_send_ascii_command("x_out = North(xa, :)");
    netsim_matlab_send_ascii_command("[ya, c] = size(East)");
    netsim_matlab_send_ascii_command("y_out = East(ya, :)");
    netsim_matlab_send_ascii_command("[za, c] = size(Height)");
    netsim_matlab_send_ascii_command("z_out = Height(za, :)");
    char output[BUFSIZ];
    netsim_matlab_get_value(output, BUFSIZ, "x_out", "double");
    xcor = atof(output);
    netsim_matlab_get_value(output, BUFSIZ, "y_out", "double");
    ycor = atof(output);
    netsim_matlab_get_value(output, BUFSIZ, "z_out", "double");
    zcor = atof(output);
    coordinates = (double*)malloc(3 * sizeof * coordinates);
    //Case 1 i.e., UAV_Test1
    //coordinates[0] = xcor;
    //coordinates[1] = ycor;
    //coordinates[2] = zcor;
    //Case 2 i.e., UAV_Test2
    coordinates[0] = xcor + DEVICE_POSITION(id+1)->X + 50;
    coordinates[1] = ycor + DEVICE_POSITION(id+1)->Y + 50;
    coordinates[2] = zcor;
    return (coordinates);
}

void uav_finish()
{
    netsim_matlab_send_ascii_command("set_param('UAV_Test2','SimulationCommand','stop');");
    fprintf(stderr, "\nTerminating MATLAB Engine Process\nPress any key to proceed...\n");
    _getch();
    netsim_matlab_interface_close();
}
```

Figure 6: Source code changes for Case 1 in UAVBasedmobility.

NetSim Plot: Application Throughput Vs. Time

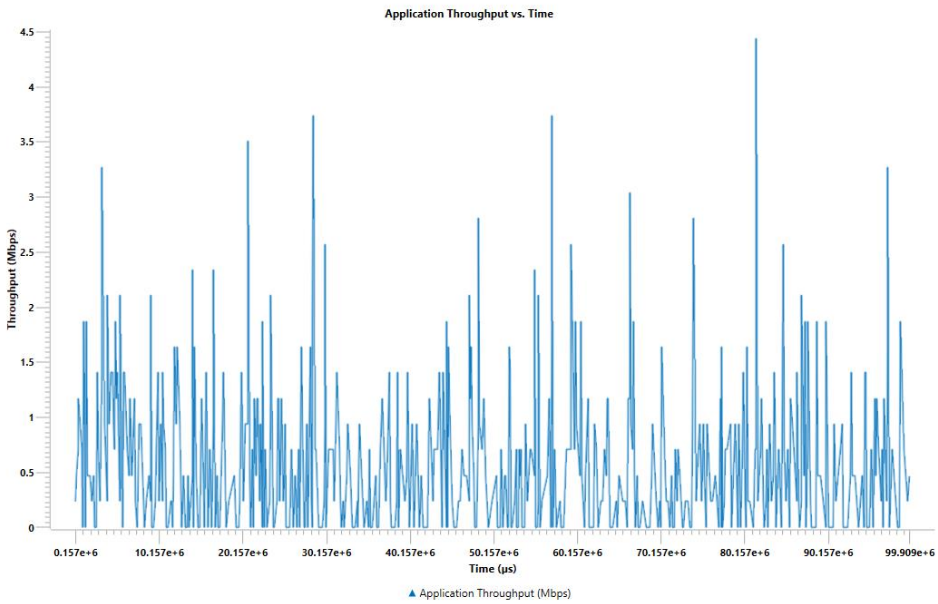


Figure 7: Application Throughput Vs. Time from NetSim

MATLAB Flight Path:

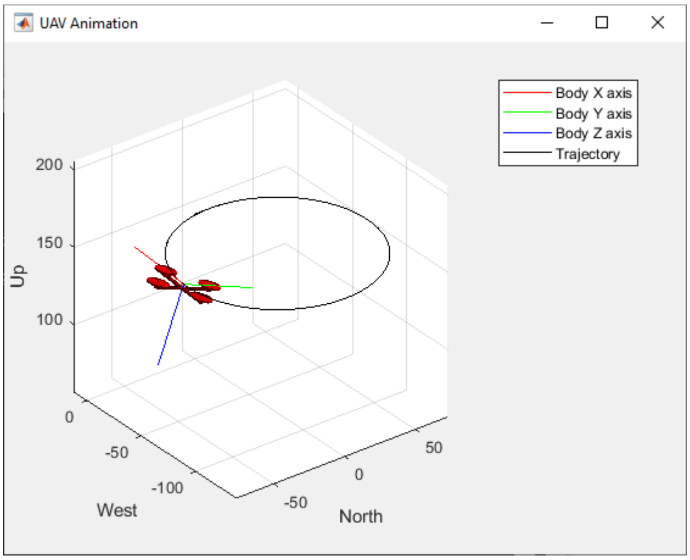


Figure 8: MATLAB Flight Path from NetSim

LTE_UAV_Example

Case 1: UAV Drone Take off - Application throughput variation with Drone mobility.

NetSim Plot: Application Throughput vs. Time

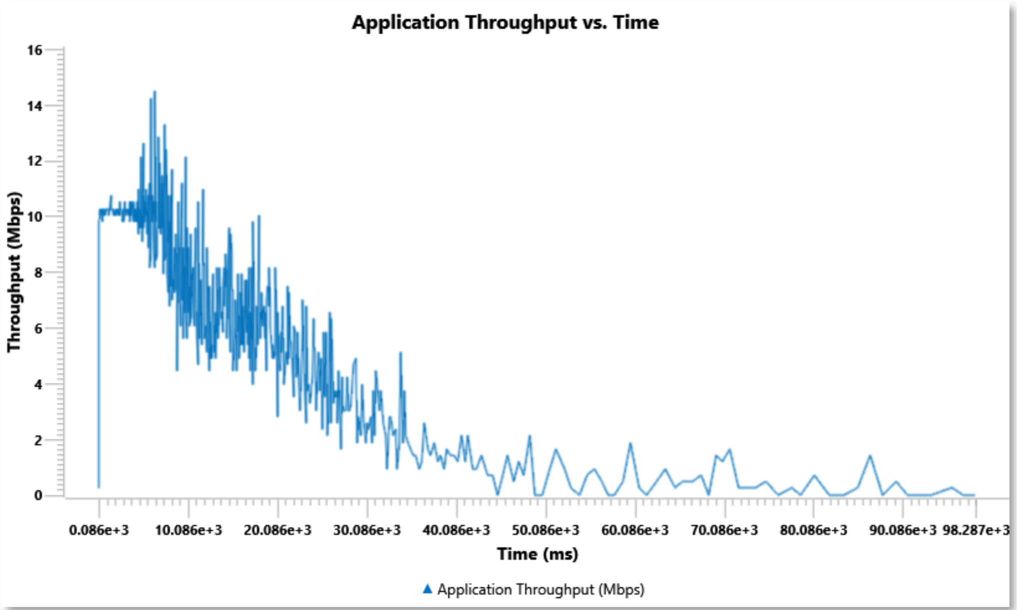


Figure 9: Application Throughput vs. Time from NetSim

1 MATLAB Flight Path

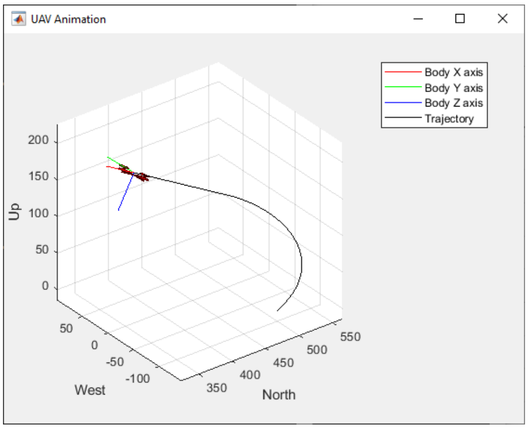


Figure 10: MATLAB Flight Path

Case 2: UAV Drone rotation around LTE-eNB

7. Before running Case 2 change the file name in the source code in UAV_Basedmobility.c as mentioned below. change the Model name in the opsimulink.m file which is present in the workspace inside the bin 64 folder.

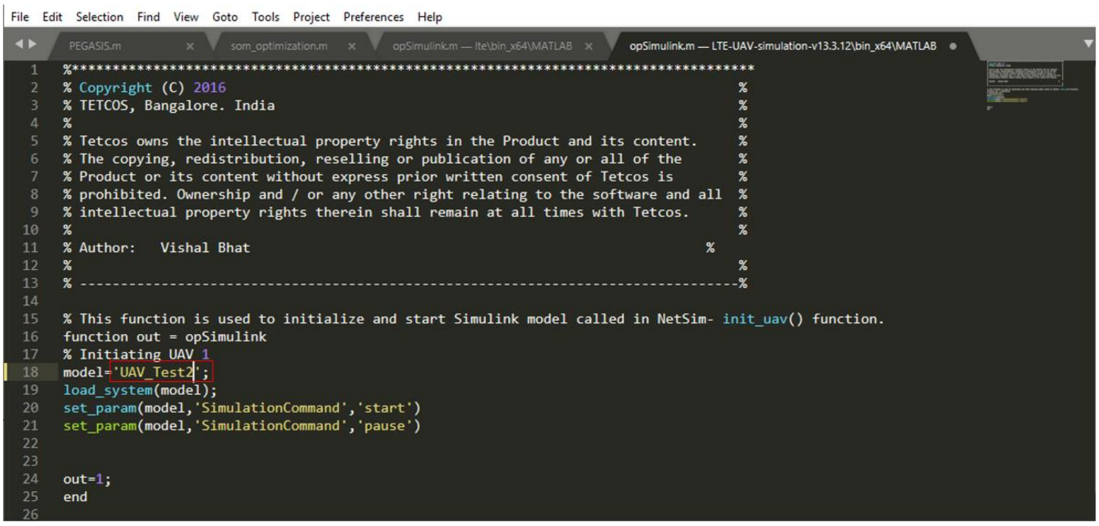


Figure 11: changes for Case 2 in opsimulink.m

8. Uncomment the case-2 **UAV_Test 2** section while running case 2 and Comment on the Case-1 **UAV_Test 1** as mentioned below.

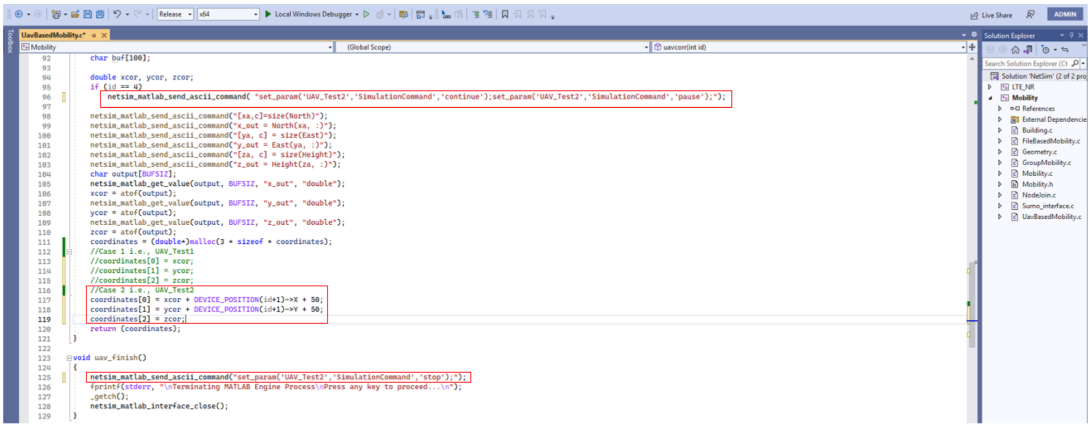


Figure 12:Source code changes for Case 2 in UAVBasedmobility.c

NetSim Plot: Application Throughput Vs. Time

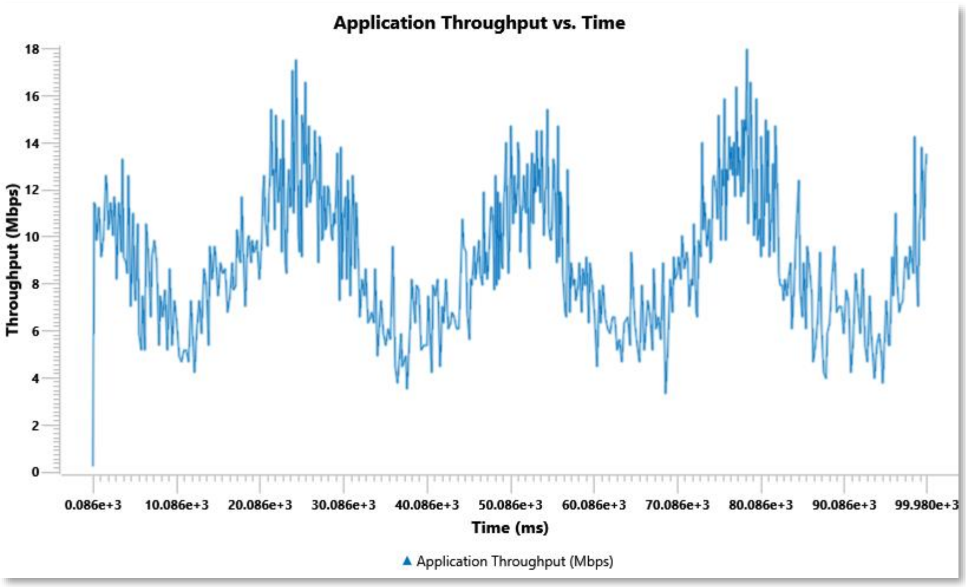


Figure 13: Application Throughput Vs. Time from NetSim

MATLAB Flight Path:

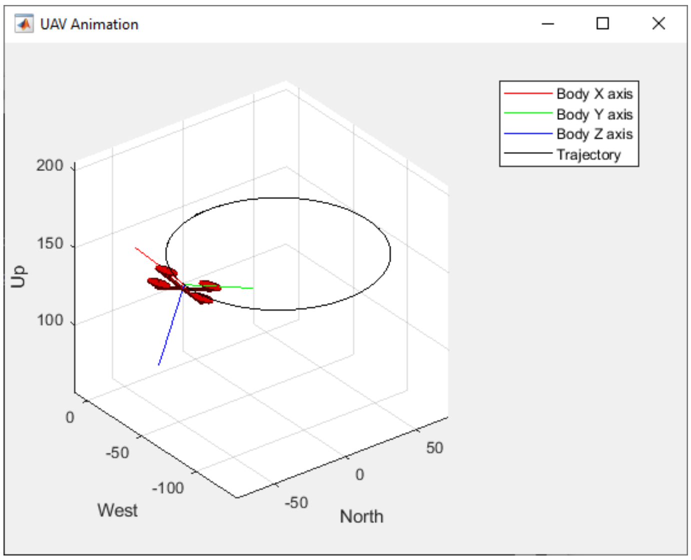


Figure 14: MATLAB Flight Path from NetSim