

## Base station - UAV Network Communication (TR 36.777 pathloss)

**Software:** NetSim Standard v14.2 (64 bit), Visual Studio 2022, MATLAB R2020b or higher. MATLAB, Simulink, Robotics and System Toolbox.

### Project Download Link:

[https://github.com/NetSim-TETCOS/5G\\_UAV\\_Simulation\\_v14.2/archive/refs/heads/main.zip](https://github.com/NetSim-TETCOS/5G_UAV_Simulation_v14.2/archive/refs/heads/main.zip)

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

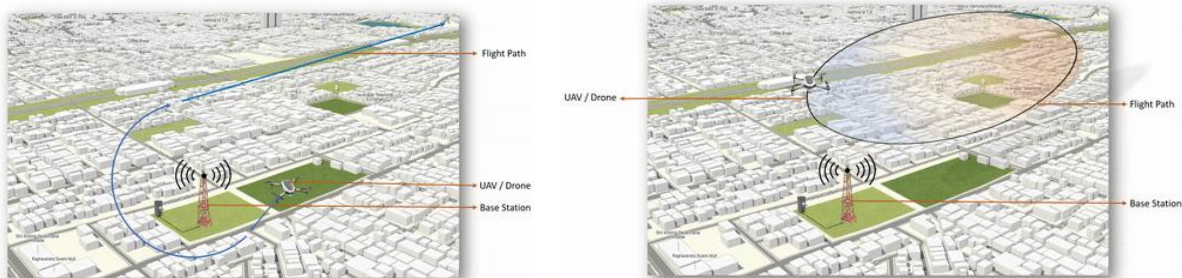
<https://support.tetcos.com/en/support/solutions/articles/14000128666-downloading-and-setting-up-netsim-file-exchange-projects>

## 1 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.

## 2 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

## 3 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.

## 4 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.

## 5 Output/Metrics specific to this example

- NetSim Animation: Mobility of the devices configured in NetSim is given as input from MATLAB.
- Pathloss is implemented as per 3GPP Standard.
- NetSim Plot Results: Throughput analysis as the UAV moves away from the eNB

## 6 Modifications done to NetSim Source codes

**Project:** Mobility

**Files:**

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

## 7 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.

## 8 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

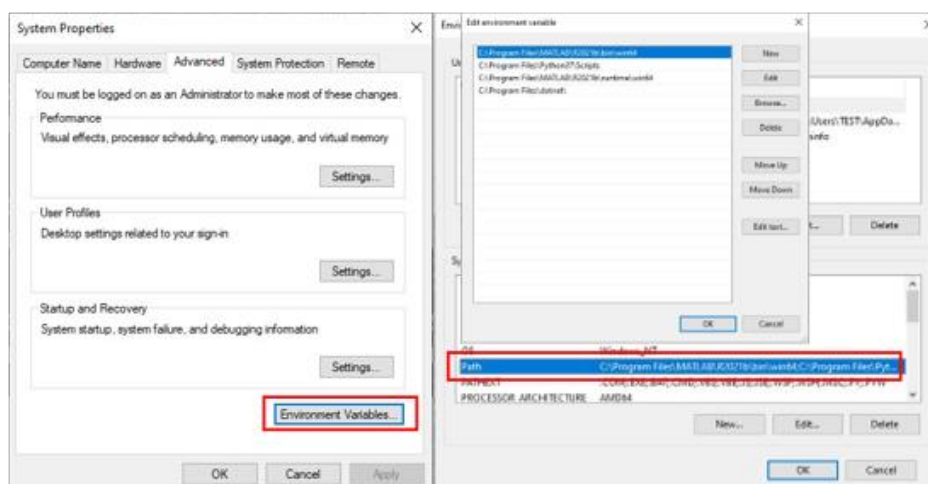


Figure 2: Environment PATH variable

**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.

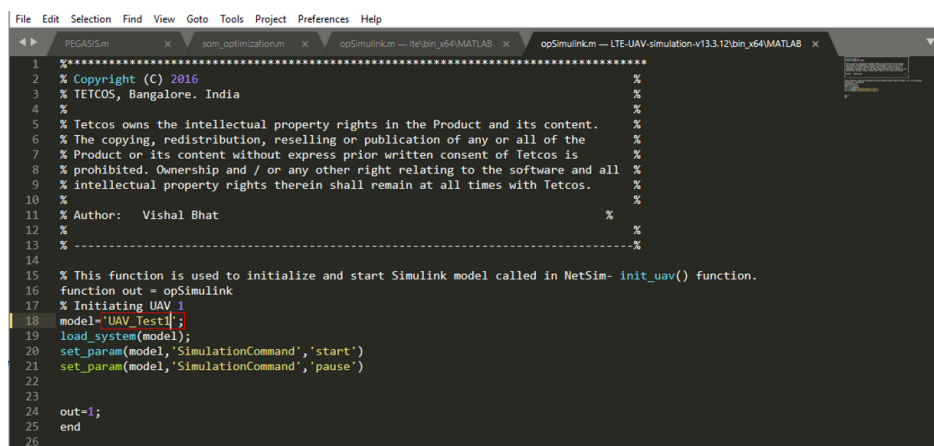
## 9 Examples

1. The **5G\_UAV\_Simulation\_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** from the list of experiments.
2. Run the simulation and press any key to continue.
3. 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.

## 10 Example

Before running Case 1 change the file name in the source code in UAV\_Basedmobility.c as mentioned below. change the model name in the opsimulink file which is present in the workspace inside the bin 64 folder.

Uncomment the case-1 **UAV\_Test 1** section while running case 1 and Comment on the Case-2 **UAV\_Test 2** as mentioned below.



```

1 %*****
2 % Copyright (C) 2016 %
3 % TETCOS, Bangalore. India %
4 % %
5 % Tetcos owns the intellectual property rights in the Product and its content. %
6 % The copying, redistribution, reselling or publication of any or all of the %
7 % Product or its content without express prior written consent of Tetcos is %
8 % prohibited. Ownership and / or any other right relating to the software and all %
9 % intellectual property rights therein shall remain at all times with Tetcos. %
10 % %
11 % Author: Vishal Bhat %
12 % %
13 % ----- %
14 %
15 % This function is used to initialize and start Simulink model called in NetSim- init_uav() function.
16 function out = opsimulink
17 % Initiating UAV 1
18 model='UAV_Test1';
19 load_system(model);
20 set_param(model,'SimulationCommand','start')
21 set_param(model,'SimulationCommand','pause')
22
23
24 out=1;
25 end
26

```

```
92 char buf[100];
93
94 double xcor, ycor, zcor;
95 if (fail == 0)
96     netSim_matlab_send_ascii_command("set_param('UAV_Test1','SimulationCommand','continue');set_param('UAV_Test1','SimulationCommand','pause');");
97
98 netSim_matlab_send_ascii_command("size North");
99 netSim_matlab_send_ascii_command("x_out = North(xa, :);");
100 netSim_matlab_send_ascii_command("ya_c = size(East);");
101 netSim_matlab_send_ascii_command("y_out = East(ya, :);");
102 netSim_matlab_send_ascii_command("za_c = size(Height);");
103 netSim_matlab_send_ascii_command("z_out = Height(za, :);");
104 char output[512];
105 netSim_matlab_get_value(output, BUF512, "x_out", "double");
106 xcor = atof(output);
107 netSim_matlab_get_value(output, BUF512, "y_out", "double");
108 ycor = atof(output);
109 netSim_matlab_get_value(output, BUF512, "z_out", "double");
110 zcor = atof(output);
111 coordinates = (double*)malloc(3 * sizeof * coordinates);
112 //Case 1 i.e., UAV_Test1
113 coordinates[0] = xcor;
114 coordinates[1] = ycor;
115 coordinates[2] = zcor;
116 //Case 2 i.e., UAV_Test2
117 //coordinates[0] = xcor + DEVICE_POSITION[id+1]-x + 50;
118 //coordinates[1] = ycor + DEVICE_POSITION[id+1]-y + 50;
119 //coordinates[2] = zcor;
120 return (coordinates);
121
122 void uav_finish()
123 {
124     netSim_matlab_send_ascii_command("set_param('UAV_Test1','SimulationCommand','stop');");
125     printf("UAV_Test1: Terminating MATLAB Engine Process-Press any key to proceed...\n");
126     getch();
127     netSim_matlab_interface_close();
128 }
129 }
```

Figure 3: Source code changes for Case 1 in UAVBasedmobility.c  
**Case 1: UAV Drone Take off - Application throughput variation with Drone mobility.**  
**NetSim Plot: Application Throughput vs. Time**

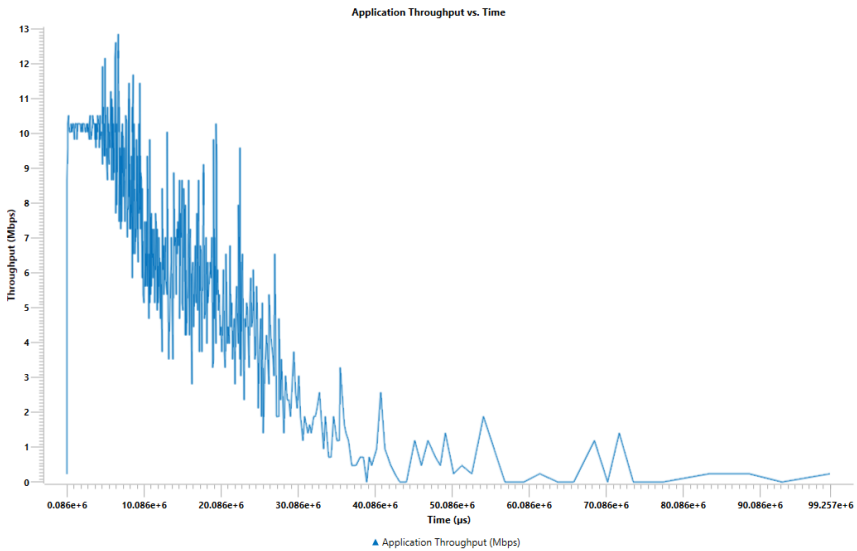


Figure 4: Application Throughput vs. Time from NetSim

# 11 MATLAB Flight Path

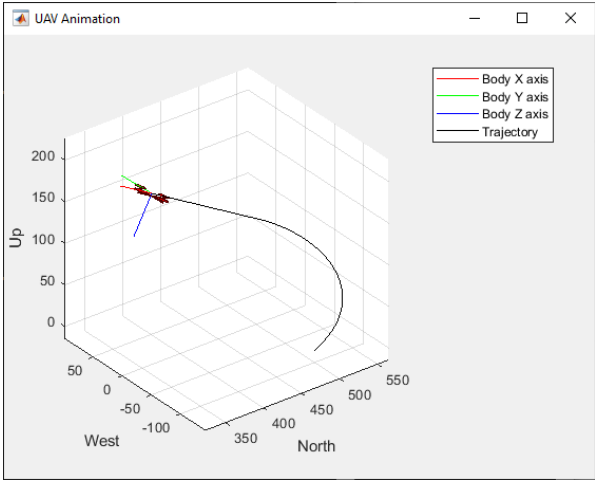


Figure 5: MATLAB Flight Path

## Case 2: UAV Drone rotation around LTE-eNB

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 file which is present in the workspace inside the bin 64 folder.

Uncomment the case-2 **UAV\_Test 2** section while running case 2 and Comment on the Case-1 **UAV\_Test 1** as mentioned below.

```

File Edit Selection Find View Goto Tools Project Preferences Help
opSimulink.m — LTE-UAV-simulation-v13.3.12\bin_x64\MATLAB
1 %*****
2 % Copyright (C) 2016 %
3 % TETCOS, Bangalore. India %
4 % %
5 % Tetcos owns the intellectual property rights in the Product and its content. %
6 % The copying, redistribution, reselling or publication of any or all of the %
7 % Product or its content without express prior written consent of Tetcos is %
8 % prohibited. Ownership and / or any other right relating to the software and all %
9 % intellectual property rights therein shall remain at all times with Tetcos. %
10 % %
11 % Author: Vishal Bhat %
12 % %
13 % ----- %
14 %
15 % This function is used to initialize and start Simulink model called in NetSim- init_uav() function.
16 function out = opSimulink
17 % Initiating UAV_1
18 model='UAV_Test2';
19 load_system(model);
20 set_param(model,'SimulationCommand','start')
21 set_param(model,'SimulationCommand','pause')
22
23
24 out=1;
25 end
26

```

```

UAVBasedmobility.c
112 char buf[100];
113 double xcor, ycor, zcor;
114 if (id == 4)
115     netsim_matlab_send_ascii_command("set_param('UAV_Test2','SimulationCommand','continue');set_param('UAV_Test2','SimulationCommand','pause');");
116
117 netsim_matlab_send_ascii_command("isa,cj=size(north)");
118 netsim_matlab_send_ascii_command("v_out = north(isa, :);");
119 netsim_matlab_send_ascii_command("isa, cj = size(east)");
120 netsim_matlab_send_ascii_command("y_out = east(isa, :);");
121 netsim_matlab_send_ascii_command("isa, cj = size(height)");
122 netsim_matlab_send_ascii_command("z_out = height(isa, :);");
123 char output[BUF_SIZE];
124 netsim_matlab_get_value(output, BUF_SIZE, "x_out", "double");
125 xcor = atof(output);
126 netsim_matlab_get_value(output, BUF_SIZE, "y_out", "double");
127 ycor = atof(output);
128 netsim_matlab_get_value(output, BUF_SIZE, "z_out", "double");
129 zcor = atof(output);
130 coordinates = (double*)malloc(3 * sizeof * coordinates);
131 //Case 1 i.e. UAV_Test1
132 //coordinates[0] = xcor;
133 //coordinates[1] = ycor;
134 //coordinates[2] = zcor;
135 //Case 2 i.e. UAV_Test2
136 coordinates[0] = xcor + DEVICE_POSITION[id+1]*x + 80;
137 coordinates[1] = ycor + DEVICE_POSITION[id+1]*y + 80;
138 coordinates[2] = zcor;
139 return (coordinates);
140
141 void uav_finish()
142 {
143     netsim_matlab_send_ascii_command("set_param('UAV_Test2','SimulationCommand','stop');");
144     fprintf(stderr, "Terminating MATLAB Engine Process! Press any key to proceed...\\n");
145     getch();
146     netsim_matlab_interface_close();
147 }

```

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

## NetSim Plot: Application Throughput Vs. Time

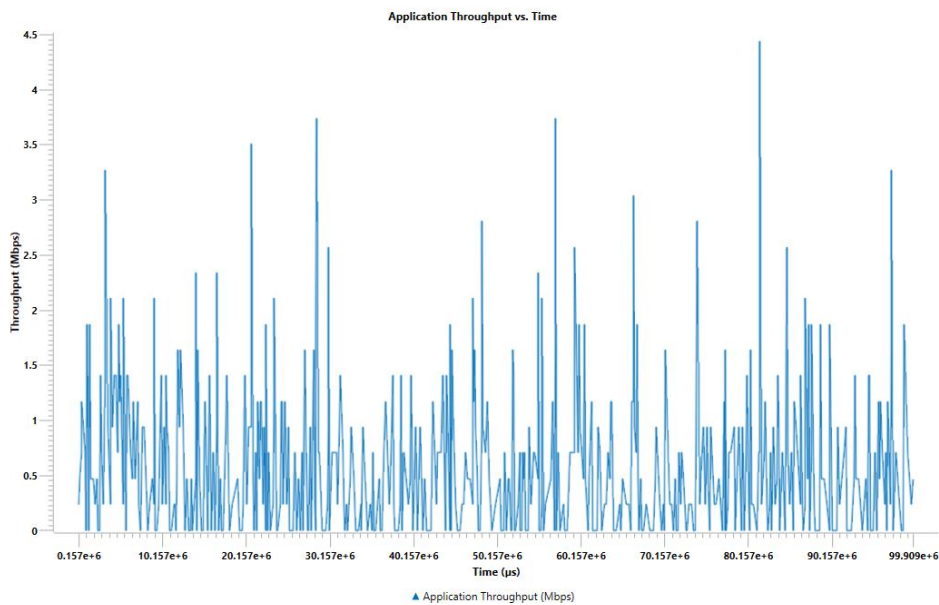


Figure 7: Application Throughput Vs. Time from NetSim

**MATLAB Flight Path:**

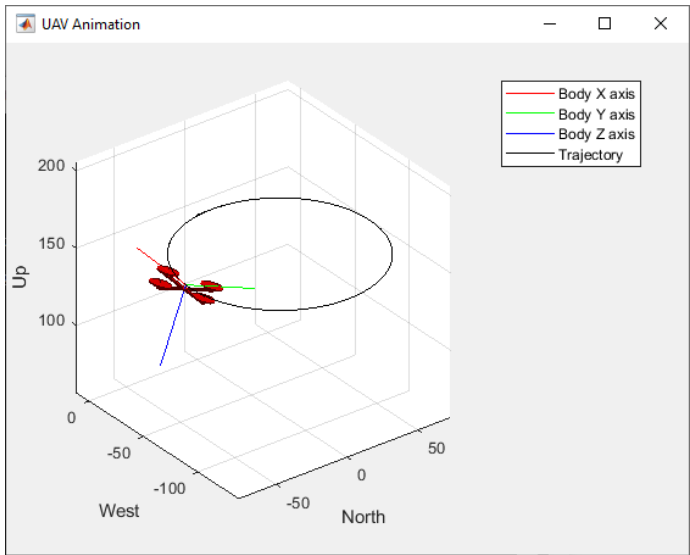


Figure 8: MATLAB Flight Path from NetSim