

Base station - UAV Network Communication (TR 36.777 pathloss)

Software: NetSim Standard v13.2 (64 bit), Visual Studio 2022, MATLAB R2020b or higher. MATLAB, Simulink, Robotics and System Toolbox.

Project Download Link:

https://github.com/NetSim-TETCOS/LTE_UAV_simulation_v13.2/archive/refs/heads/main.zip

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

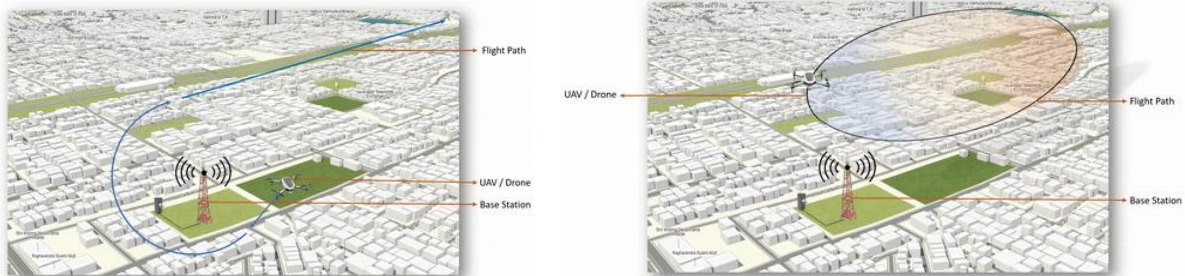
<https://support.tetcos.com/en/support/solutions/articles/14000128666-downloading-and-setting-up-netsim-file-exchange-projects>

Objective

Performance analysis of UAV - Base station network communications. This involves interfacing NetSim and 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 modelled 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 co-ordinates 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 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 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 MATLAB workspace and given as input to NetSim's Mobility model. In this example, coordinates are taken every one second and updated to the device mobility.

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

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 MATLAB Engine process
- **Mobility.vcxproj** - This is a Visual Studio project file used to load and manage the source codes related to the 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 co-ordinates from MATLAB workspace for every step size set in NetSim.
- **uavcorr()**: Function to get co-ordinates from MATLAB.

Steps to simulate

1. Add the following MATLAB install directory path in the Environment PATH variable
 <MATLAB_INSTALL_DIRECTORY>\bin\win64
 For eg: C:\Program Files\MATLAB\R2020b\bin\win64

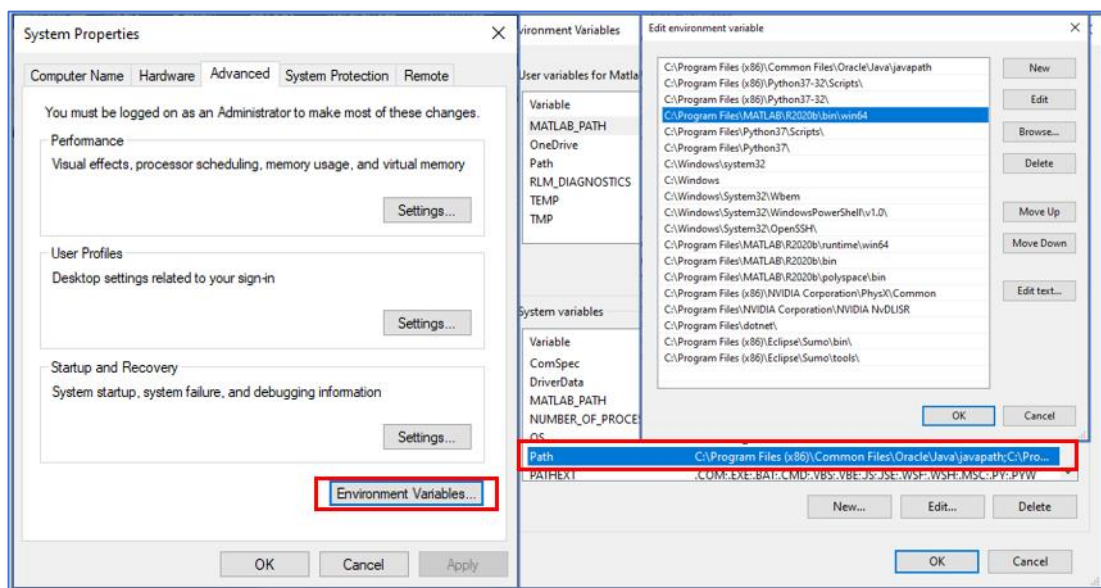


Figure 2: Environment PATH variable

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).

2. Open 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 MATLAB automation server during runtime.
3. Copy **GeneralProperties.xml** file to <NetSim Installation Directory>\ Docs\xml\Properties. And copy **Configuration.xsd** to <Netsim Installation Directory>\Bin path.
4. Open the Source codes in Visual Studio by going to Your Work-> Workspace Options and Clicking on Open code button.
5. Open the Source codes in Visual Studio by going to Your Work-> Workspace Options and Clicking on Open code button.
6. Under the Mobility project in the solution explorer, you will be able to see that and **UAV_Matlab_Interface.c** files which contain source codes related to interactions with MATLAB and handling clustering in NetSim respectively.
7. Run NetSim as Administrative mode.

Examples

1. The **LTE_UAV_Simulation_Workspace** comes with a sample configuration that is already saved. To open this example, go to Your Work option in the Home Screen of NetSim and click on the **LTE_UAV_Example** from the list of experiments.
2. Run simulation and press any key to continue.
3. It will open 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 and 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.

Example

Case 1: UAV Drone Takeoff - Application throughput variation with Drone mobility

- **NetSim Plot: Application Throughput vs. Time**

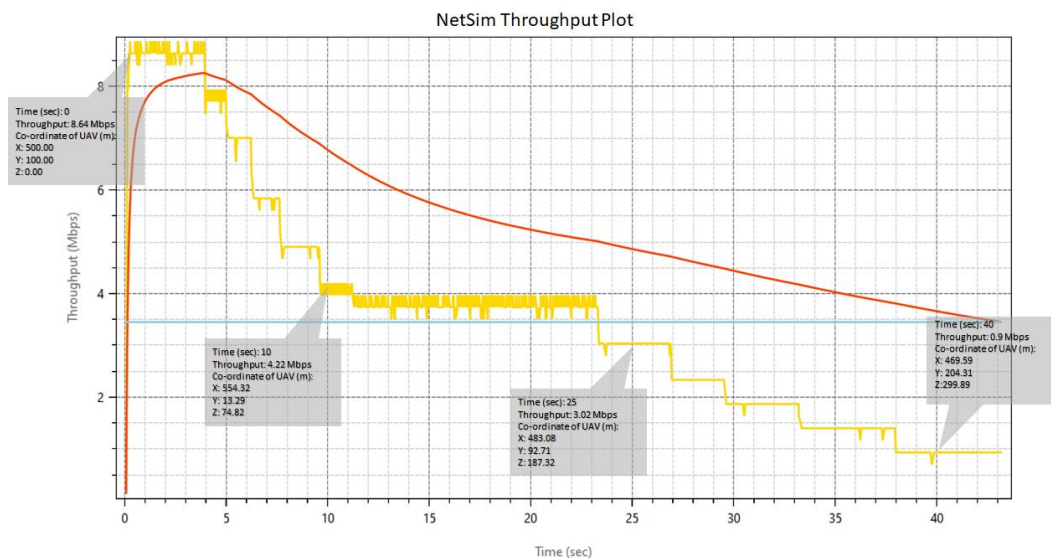


Figure 3: Application Throughput vs. Time from NetSim

- **MATLAB Flight Path**

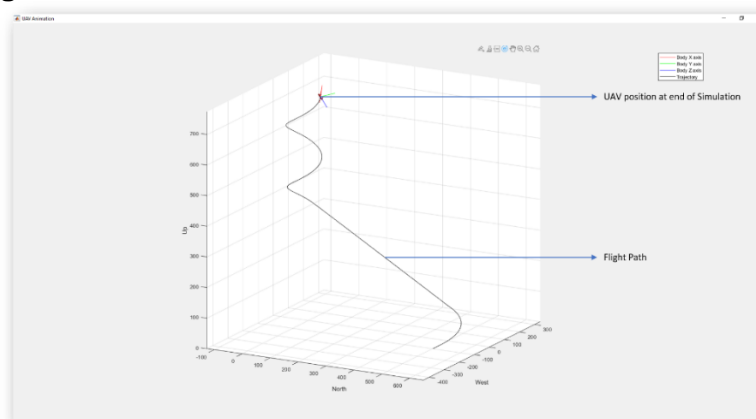


Figure 4: MATLAB Flight Path

Case 2: UAV Drone rotation around LTE-eNB

- **NetSim Plot: Application Throughput Vs. Time**

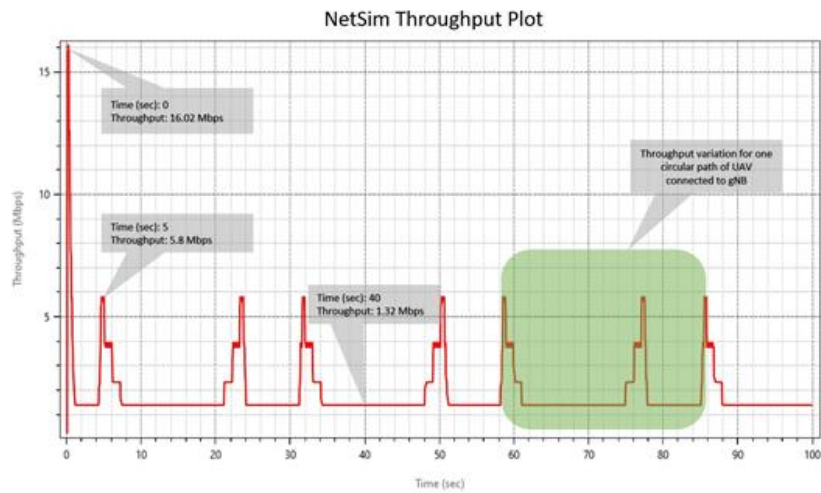


Figure 5: Application Throughput Vs. Time from NetSim

- MATLAB Flight Path**

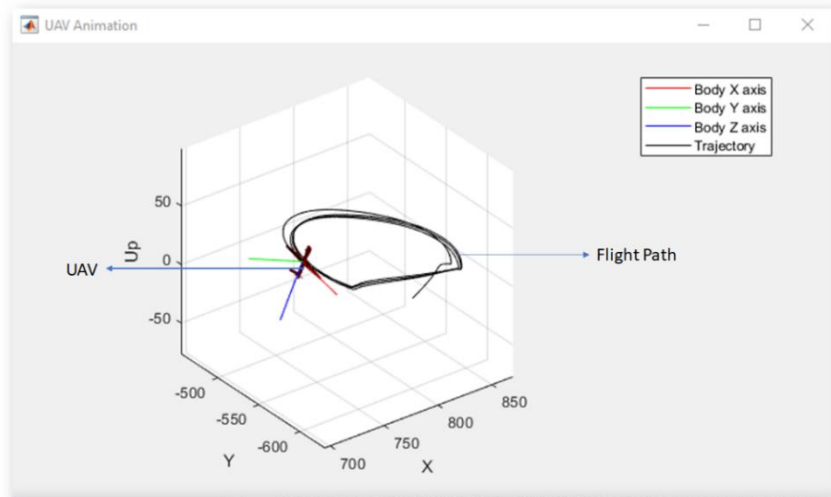


Figure 6: MATLAB Flight Path from NetSim