25_산학협력프로젝트- 매트랩 코드 분석

참고 코드 : https://kr.mathworks.com/help/satcom/ug/analyze-ntn-coverage-and-capacity-for-leo-mega-constellation.html

목적 : 비지상파 네트워크(NTN) 애플릴케이션을 위한 LEO 위성의 메가콘스텔레이션에 대한 인구 밀도가 매우 낮은 지역의 커버리지와 용량을 분석

3가지의 서로 다른 저궤도 위성 컨스텔레이션 정의

(위성의 고도 및 경사각 / 궤도면 개수 및 위성 개수)

⇒ 이 중 제일 좋은 궤도면 개수 및 위성 개수 보이는 걸 선택하여 하나만 사용

```
% Define the satellite constellations.
% All these fields must have same number of entries: ShellNames,
% ShellAltitude, ShellInclination, NumOrbits, and NumSatellitesPerOrbit
constellation = struct;
constellation(1).Name = "Constellation 1";
constellation(1).ShellNames = ["Altitude 1", "Altitude 2"];
constellation(1).ShellAltitude = [1015 1325];            % In km
constellation(1).ShellInclination = [98.98 50.88];       % In degrees
constellation(1).NumOrbits = [3 6];                      % Number of orbital plane
s
constellation(1).NumSatellitesPerOrbit = [5 5];          % Number of satellite
s per orbital plane

constellation(2).Name = "Constellation 2";
constellation(2).ShellNames = ["Altitude 1", "Altitude 2"];
constellation(2).ShellAltitude = [1015 1325];            % In km
constellation(2).ShellInclination = [98.98 50.88];       % In degrees
constellation(2).NumOrbits = [6 20];                     % Number of orbital plan
es
constellation(2).NumSatellitesPerOrbit = [13 11];        % Number of satellite
s per orbital plane
```

```
constellation(3).Name = "Constellation 3";
constellation(3).ShellNames = ["Altitude 1", "Altitude 2"];
constellation(3).ShellAltitude = [1015 1325];             % In km
constellation(3).ShellInclination = [98.98 50.88];        % In degrees
constellation(3).NumOrbits = [12 20];                     % Number of orbital plan
es
constellation(3).NumSatellitesPerOrbit = [13 22];         % Number of satellit
es per orbital plane

% Validate dimensions of each constellation / 차원 검증
validateConstellation(constellation)
```

위성 송신기 구조 설정

```
txConfig = struct;
txConfig.Frequency = 2e9;  %  반송파 주파수 (2GHz)
txConfig.Power = 20;       % 송신 전력
txConfig.BitRate = 10;     % Bit rate in Mbps
txConfig.SystemLoss = 0;   % 송신 측 시스템 손실 in dB
txConfig.Bandwidth = 10e6; % Link bandwidth in Hz
```

관심지역 및 사용자 장비 수 설정

```
lat = [20 70];     % Limits of latitude in degrees ([min max])
lon = [-90 20];    % Limits of longitude in degrees ([min max])
numUEs = 50;       % Number of UE
minElevAngle = 30; % Minimum elevation angle in degrees
```

지상국에 대한 수신기 매개변수

```
rxConfig = struct;
rxConfig.MaxGByT = -5;        % Maximum gain-to-noise-temperature in dB/
K
rxConfig.SystemLoss = 0;      % Receiver system loss in dB
```

```
rxConfig.PreReceiverLoss = 0; % Pre-receiver loss in dB
rxConfig.RequiredEbNo = 11;   % Required bit energy to noise power spectr
al density ratio in dB
```

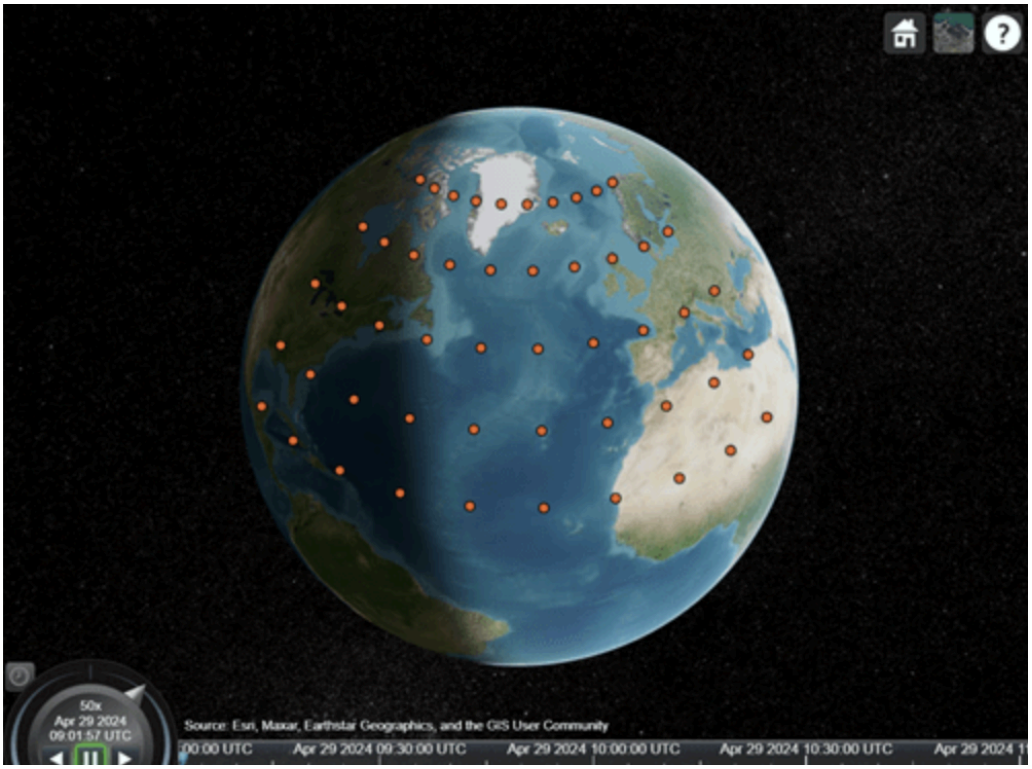
위성 시나리오에 관심영역 표시

```
% Find the coordinates of uniformly spaced UEs within the ROI
[latSpacing,lonSpacing] = findClosestFactors(numUEs);
latPts = linspace(lat(1),lat(2),latSpacing);
lonPts = linspace(lon(1),lon(2),lonSpacing);
[latMesh,lonMesh] = meshgrid(latPts,lonPts);
latCoord = latMesh(:);
lonCoord = lonMesh(:);

% Add the ground station to the scenario
ue = groundStation(sc,latCoord,lonCoord);
[ue.MinElevationAngle] = minElevAngle;

% Add the receive antenna to the ground station
isotropic = arrayConfig(Size=[1 1]);
rx = receiver(ue,Antenna=isotropic, ...
    SystemLoss=rxConfig.SystemLoss, ...
    PreReceiverLoss=rxConfig.PreReceiverLoss, ...
    GainToNoiseTemperatureRatio=rxConfig.MaxGByT);

% Show the ROI
s = satelliteScenarioViewer(sc,ShowDetails=false);
```



서로 다른 저궤도 위성 컨스텔레이션 적용 범위 지표 계산

```
% Delete viewer to avoid animating all the constellation and temporary
% links
delete(s)

% For each timestamp and each UE, find the maximum signal power
timeSteps = sc.StartTime:seconds(sc.SampleTime):sc.StopTime;
numTimeSteps = numel(timeSteps);
% Store the results in an output structure
resultPerConstellation = struct("SatelliteVisibility",[], ...
    "LinkAvailability",[], ...
    "MinLinkAvailability",[], ...
    "MinSatelliteVisibility",[], ...
    "MaxEbNo",[], ...
    "MaxCNR",[], ...
    "Capacity",[], ...
    "NumTotalSatellites",[]);
results = repmat(resultPerConstellation,numel(constellation),1);
```

```
% Loop each constellation
for constIdx = 1:numel(constellation)
    % Display status message at the start of processing loop
    fprintf("Computing coverage statistics for constellation " + constIdx)

    % Initialize the variables
    [linkAvailability,satelliteVisibility, ...
      maxEbNo] = deal(zeros(numUEs,numTimeSteps));

    % Add satellites to the scenario
    sat = addSatellites(sc,constellation(constIdx));
    % Add transmitter to the satellites
    tx = transmitter(sat, ...
        Frequency=txConfig.Frequency, ...
        Power=txConfig.Power, ...
        SystemLoss=txConfig.SystemLoss, ...
        BitRate=txConfig.BitRate, ...
        Antenna=arrayConfig(Size=[1 1]));

    for ueIdx = 1:numUEs
        % Display progress through dots in intervals of 20%
        if mod(ueIdx,ceil(0.2*numUEs)) == 0
            fprintf(".")
        end

        % Find the elevation angle of all the satellites with respect to
        % UE
        [~,el] = aer(ue(ueIdx),sat);
        % Find the satellites that have elevation angles greater than or equal
        % to minimum elevation angle
        elIdx = el >= ue(ueIdx).MinElevationAngle;
        % Satellite visibility: Set to 1 when a satellite is visible to the
        % UE
        satelliteVisibility(ueIdx,:) = any(elIdx,1);
        % Create temporary links for all the satellites that are visible over
        % the simulation duration
        validTx = tx(any(elIdx,2));
        if ~isempty(validTx)
```

```
            links = link(validTx,rx(ueIdx));
            % Calculate the received Eb/No
            ebByNo = ebno(links);
            % Get the maximum Eb/No when link is closed
            maxEbNo(ueIdx,:) = max(ebByNo);
            % Link Availability: Set to 1 when the link is closed
            linkAvailability(ueIdx,:) = any(linkStatus(links));
            % Delete links
            delete(links)
        end
    end

    % Compute the max CNR based on max Eb/No
    cno = maxEbNo + pow2db(tx(1).BitRate) + 60;  % Bit rate is in Mbps
    cnrdB = cno - pow2db(txConfig.Bandwidth);
    cnr = db2pow(cnrdB);
    % Compute capacity using Shannon theorem, only for available links
    capacity = txConfig.Bandwidth*log2(1+cnr); % bps → Shannon Capacity
공식

    % Capture the results to the output structure
    results(constIdx).SatelliteVisibility = satelliteVisibility;
    results(constIdx).MinSatelliteVisibility = min(mean(satelliteVisibility,2));
    results(constIdx).LinkAvailability = linkAvailability;
    results(constIdx).MinLinkAvailability = min(mean(linkAvailability,2));
    results(constIdx).MaxEbNo = maxEbNo;
    results(constIdx).MaxCNR = cnr;
    results(constIdx).Capacity = capacity;
    results(constIdx).NumTotalSatellites = numel(sat);

    % Delete the satellites
    delete(sat)

    % Display status message at end of processing loop
    fprintf(newline + "Computed coverage statistics for constellation " + con
stIdx + newline)
end
```

위성 가시성 (관측 가능한 시간 인스턴스 수 / 전체 시간 인스턴스 수 ⇒ 해당 지역의 모든 사용자 기기 중 가장 낮은 가시성 수준으로 결정)

```
satelliteVisibility = [results.MinSatelliteVisibility]'*100; % Value in percentag
e
```

링크 가용성 (음수가 아닌 링크 마진을 갖는 시간 인스턴스 수 / 전체 시간 인스턴스 수)

```
inkAvailability = [results.MinLinkAvailability]'*100;
```

임계값보다 큰 용량을 가진 UE의 확률

```
coveragePercent = zeros(numel(results),1);
threshold = 1;                                  % In Mbps
for i = 1:numel(results)
    cTemp = results(i).Capacity;
    coveragePercent(i) = mean(cTemp > threshold*1e6,[1 2])*100;
end
```

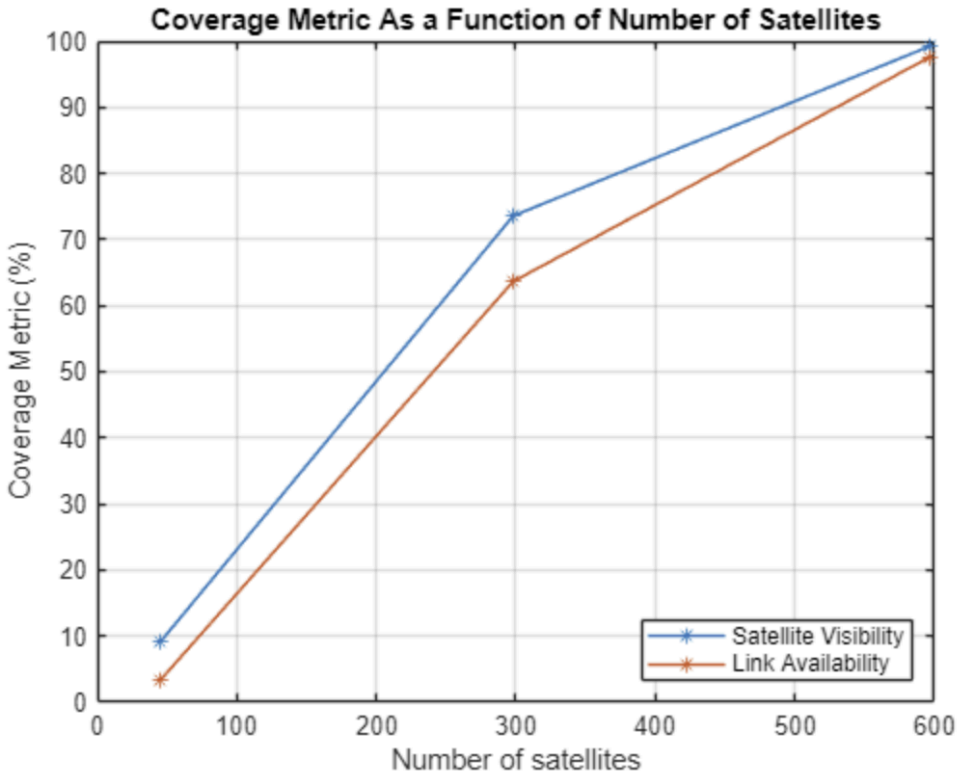
결과 표로 정리

```
numTotalSatellites = [results.NumTotalSatellites];
constellationNames = [constellation.Name];
resultTable = table(constellationNames',numTotalSatellites',satelliteVisibilit
y,linkAvailability,coveragePercent, ...
    VariableNames={'Constellation Name','Num. Satellites','Satellite Visibility
(%)','Link Availability (%)', ...
    ['Capacity > ',num2str(threshold),' Mbps (%x)']})
    % 이름 / 위성 수 / 위성 가시성 / 링크 가용성 / 용량 > 임계값
```

위성 가시성과 링크 가용성 그래프 표시

```
figure(1)
% Plot the percentage of satellite visibility of the region for the whole
% simulation duration
```

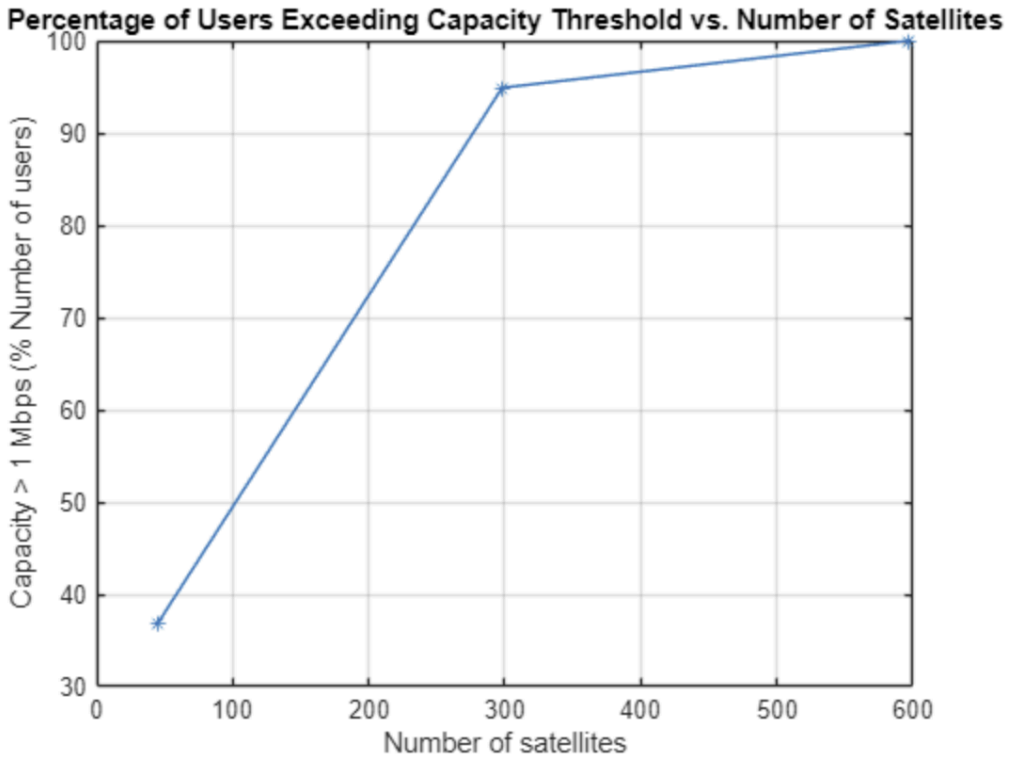
```
plot(numTotalSatellites,satelliteVisibility,"-*")
grid on
box on
xlabel("Number of satellites")
ylabel("Coverage Metric (%)")
hold on
% Plot the percentage of link availability of the region for the whole
% simulation duration
plot(numTotalSatellites,linkAvailability,"-*")
% Add legend to the plot
legend("Satellite Visibility","Link Availability", ...
    Location="southeast")
hold off
title("Coverage Metric As a Function of Number of Satellites")
```



해당 지역 사용자 기기 평균 수 기준 임계값보다 큰 용량을 가진 사용자 기기 백분위 표시

```
figure(2)
plot(numTotalSatellites,coveragePercent,"-*")
grid on
```

```
box on
xlabel("Number of satellites")
ylabel("Capacity > " + num2str(threshold) + " Mbps (% Number of users)")
title("Percentage of Users Exceeding Capacity Threshold vs. Number of Sa
tellites")
```



사용자 기기에 대한 차트 표시

```
figure(3)
% Scale the status with the UE number for better visualization
colors = colororder;
firstLineColor = colors(1,:);
ueIndex = (1:numUEs)';
linkAvailabilityFirstConst = results(1).LinkAvailability;
linkAvailabilityFirstConst(linkAvailabilityFirstConst == 0) = nan;
plot(timeSteps,linkAvailabilityFirstConst.*ueIndex, ...
    Color=firstLineColor,LineWidth=1)
xlim([timeSteps(1) timeSteps(end)])
ylim([0 numUEs+1])
xlabel("Time")
```

```
ylabel("UE Index")
title("Link Availability Chart")
```