```
classdef FDChan < matlab.System
    % Frequency-domain multipath channel
    properties
        % Configuration
        carrierConfig;   % Carrier configuration

        % Path parameters
        gain;  % Path gain in dB
        dly;   % Delay of each path in seconds
        aoaAz, aoaEl; % Angle of arrival of each path in degrees
        fd;    % Doppler shift for each path

        rxVel = [30,0,0]';  % Mobile velocity vector in m/s
        fc = 28e9;    % Carrier freq in Hz

        gainComplex;  % Complex gain of each path

        % SNR parameters
        Etx = 1;       % average energy per PDSCH symbol
        EsN0Avg = 20;  % Avg SNR per RX symbol in dB

        % Symbol times
        symStart;  % symStart(i) = start of symbol i relative to subframe

    end
    methods
        function obj = FDChan(carrierConfig, varargin)
            % Constructor

            % Save the carrier configuration
            obj.carrierConfig = carrierConfig;


            % Set parameters from constructor arguments
            if nargin >= 1
                obj.set(varargin{:});
            end

            % TODO:  Create complex path gain for each path
            obj.gainComplex = 10.^(0.05*obj.gain).*exp(1i*(rand(size(obj.gain))*2*pi));

            % TODO:  Compute the Doppler shift for each path
            [x,y,z] = sph2cart(deg2rad(obj.aoaAz'),deg2rad(obj.aoaEl'),ones(size(obj.aoaAz')));
            u = [x; y; z]';
            obj.fd = u*obj.rxVel*(obj.fc/physconst('lightspeed'));

            % Compute unit vector in direction of each path

            % TODO:  Compute the vector of
            % symbol times relative to the start of the subframe
            durations = obj.carrierConfig.SymbolLengths ./ obj.carrierConfig.SampleRate;
            cumulative_sum = cumsum(durations);
            obj.symStart = [0, cumulative_sum(1:end-1)];

        end


    end
    methods (Access = protected)
```

```
        function [rxGrid, chanGrid, noiseVar] = stepImpl(obj, txGrid, sfNum, slotNum)
            % Applies a frequency domain channel and noise
            %
            % Given the TX grid of OFDM REs, txGrid, the function
            % *  Computes the channel grid, chanGrid, given the
            %    subframe number, sfNum, and slotNum.
            % *  Computes the noise variance per symbol, noiseVar,
            %    for a target SNR
            % *  Applies the channel and noise to create the RX grid
            %    of symbols, rxGrid.
            start_idx = 14*slotNum+1;
            % disp(obj.symStart(start_idx:start_idx+14-1));
            t = obj.symStart(start_idx:start_idx+14-1)' + sfNum*1e-3;

            [nsc, nsym] = size(txGrid);
            chanGrid = zeros(nsc, nsym);
            scs = obj.carrierConfig.SlotsPerSubframe*15*1e3;
            f = scs*(0:nsc-1)';

            for k = 1:length(obj.gainComplex)
                phase = obj.fd(k)*t' + obj.dly(k)*f;
                chanGrid = chanGrid + obj.gainComplex(k)*exp(2*pi*1i*phase);
            end

            E_ch_gain = sum(abs(obj.gainComplex).^2);
            noiseVar = obj.Etx*E_ch_gain/db2pow(obj.EsN0Avg);

            %chan_awgn = comm.AWGNChannel('NoiseMethod', 'Variance',...
            %'Variance', noiseVar);

            rxGrid = txGrid.*chanGrid + (randn(size(txGrid))+1i*randn(size(txGrid)))*sqrt(noiseVar/2);
        end

    end
end
```

```
Not enough input arguments.

Error in FDChan (line 31)
            obj.carrierConfig = carrierConfig;
```

Published with MATLAB® R2020b