```
classdef NRgNBTxFD < matlab.System
    % 5G NR gNB transmitter class implemented in frequency domain
    properties
        % Configuration
        carrierConfig;   % Carrier configuration
        pdschConfig;     % PDSCH configuration

        % Coded bits transmitted on PDSCH
        txBits;

        % Transmitted symbols
        pdschSym;

        % Modulation parameters for test
        bitsPerSym = 2;

        % Channel
        txGridChan;
        chanNames;


    end
    methods
        function obj = NRgNBTxFD(carrierConfig, pdschConfig, ...
                varargin)
            % Constructor

            % Save the carrier and PDSCH configuration
            obj.carrierConfig = carrierConfig;
            obj.pdschConfig = pdschConfig;

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


        end

         function setAck(obj, iharq)
            % Set that the HARQ transmission was received correctly
            obj.newDataAvail(iharq) = 1;

        end
    end
    methods (Access = protected)

        function [txGrid] = stepImpl(obj)
            % step implementation. Creates one slot of samples for each
            % component carrier


            % Create the OFDM grid representing the array of modulation
            % symbols to be transmitted
            txGrid = nrResourceGrid(obj.carrierConfig, ...
                obj.pdschConfig.NumLayers);

            % TODO:  Get indices on where the PDSCH is allocated
            pdschInd = nrPDSCHIndices(obj.carrierConfig,obj.pdschConfig);
```

```
            % TODO:  Create random bits for the PDSCH
            % and modulate the bits to symbols.
            % Use obj.bitsPerSym to determine the modulation order
            obj.txBits = randi([0 1], length(pdschInd)*obj.bitsPerSym,1);
            obj.pdschSym = qammod(obj.txBits, 2^obj.bitsPerSym,...
                                  'UnitAveragePower', true, ...
                                  'InputType', 'bit');

            % Insert the PDSCH symbols into the TX grid
            txGrid(pdschInd) = obj.pdschSym;

            % Get the PT-RS symbols and indices and insert them
            % in the TX grid
            ptrsSym = nrPDSCHPTRS(obj.carrierConfig, obj.pdschConfig);
            ptrsInd = nrPDSCHPTRSIndices(obj.carrierConfig, obj.pdschConfig);
            txGrid(ptrsInd) = ptrsSym;

            % TODO:  Get the DM-R indices and symbols and insert them
            % in the TX grid
            dmrsSym = nrPDSCHDMRS(obj.carrierConfig, obj.pdschConfig);
            dmrsInd = nrPDSCHDMRSIndices(obj.carrierConfig, obj.pdschConfig);
            txGrid(dmrsInd) = dmrsSym;

            % For debugging, we create a grid with the labels for
            % the channel indices
            numPorts = 1;
            obj.txGridChan = nrResourceGrid(obj.carrierConfig, numPorts);
            obj.txGridChan(pdschInd) = 1;
            obj.txGridChan(dmrsInd) = 2;
            obj.txGridChan(ptrsInd) = 3;
            obj.chanNames = {'Other', 'PDSCH', 'DM-RS', 'PT-RS'};

        end

    end
end
```

```
Not enough input arguments.

Error in NRgNBTxFD (line 29)
            obj.carrierConfig = carrierConfig;
```

Published with MATLAB® R2020b