```
function helperslexMonostaticRadarParam
% This function is only in support of slexMonostaticRadarExample.
% It may be removed in a future release.

%   Copyright 2014 The MathWorks, Inc.

    [propSpeed, fc, pulseBw, prf, fs, txGain, peakPower, ...
     matchingCoeff, metersPerSample, rangeOffset, rangeLoss, ...
     referenceLoss, target1Rcs, target1Pos, target1Vel] = calcParams();

    % Environment
    paramRadar.propSpeed = propSpeed;
    paramRadar.fc = fc;
    % Waveform parameters
    paramRadar.pulseBw = pulseBw;
    paramRadar.prf = prf;
    paramRadar.fs = fs;
    % Transmitter parameters
    paramRadar.txGain = txGain;
    paramRadar.peakPower =  peakPower;
    % Matched filter parameters
    paramRadar.matchingCoeff = matchingCoeff;
    % Time varying gain parameters
    paramRadar.metersPerSample = metersPerSample;
    paramRadar.rangeOffset = rangeOffset;
    paramRadar.rangeLoss = rangeLoss;
    paramRadar.referenceLoss = referenceLoss;
    % Radar parameters
    paramRadar.target1Rcs = target1Rcs;
    paramRadar.target1Pos = target1Pos;
    paramRadar.target1Vel = target1Vel;

    assignin('base','paramRadar',paramRadar);

end

function [propSpeed, fc, pulseBw, prf, fs, txGain, peakPower, ...
          matchingCoeff, metersPerSample, rangeOffset, rangeLoss, ...
          referenceLoss, target1Rcs, target1Pos, target1Vel]  = calcParams()
    % Environment
    propSpeed = physconst('LightSpeed');   % Propagation speed
    fc = 10e9;           % Operating frequency
    lambda = propSpeed/fc;


    % Constraints
    maxRange = 5000;    % Maximum unambiguous range
    rangeRes = 50;      % Required range resolution
    pd = 0.9;            % Probability of detection
    pfa = 1e-6;          % Probability of false alarm
    tgtRcs = 1;         % Required target radar cross section
    numPulseInt = 10;  % Integrate 10 pulses at a time


    % Waveform parameters
    pulseBw = propSpeed/(2*rangeRes);    % Pulse bandwidth
    pulseWidth = 1/pulseBw;               % Pulse width
    prf = propSpeed/(2*maxRange);         % Pulse repetition frequency
    fs = 2*pulseBw;
```

```
    % Transmitter parameters
    snrMin = albersheim(pd, pfa, numPulseInt);
    txGain = 20;
    peakPower = ((4*pi)^3*noisepow(1/pulseWidth)*maxRange^4*...
        db2pow(snrMin))/(db2pow(2*txGain)*tgtRcs*lambda^2);


    % Matched filter parameters
    hwav = phased.RectangularWaveform(...
        'PulseWidth',1/pulseBw,...
        'PRF',prf,...
        'SampleRate',fs);
    matchingCoeff = getMatchedFilter(hwav);

    % Delay introduced due to filter
    matchingDelay = size(matchingCoeff,1)-1;

    % Time varying gain parameters
    fastTimeGrid = unigrid(0,1/fs,1/prf,'[)');
    rangeGates = propSpeed*fastTimeGrid/2;
    metersPerSample = rangeGates(2);
    rangeOffset = -rangeGates(2)*matchingDelay;
    rangeLoss = 2*fspl(rangeGates,lambda);
    referenceLoss = 2*fspl(maxRange,lambda);

    %Radar parameters
    target1Rcs = 0.6;
    target1Pos = [1988.66;0;0];
    target1Vel = [ 0; 0; 0 ];

end

function helperslexMonostaticRadarMultipleTargetsParam
% This function is only in support of slexMonostaticRadarExample.
% It may be removed in a future release.

%   Copyright 2014 The MathWorks, Inc.

    [propSpeed, fc, pulseBw, prf, fs, txGain, peakPower, ...
     matchingCoeff, metersPerSample, rangeOffset, rangeLoss, ...
     referenceLoss, targetRcs, targetPos, targetVel,lambda] = calcParams();

    % Environment
    paramRadarMT.propSpeed = propSpeed;
    paramRadarMT.fc = fc;
    paramRadarMT.lambda = lambda;
    % Waveform parameters
    paramRadarMT.pulseBw = pulseBw;
    paramRadarMT.prf = prf;
    paramRadarMT.fs = fs;
    % Transmitter parameters
    paramRadarMT.txGain = txGain;
    paramRadarMT.peakPower =  peakPower;
    % Matched filter parameters
    paramRadarMT.matchingCoeff = matchingCoeff;
    % Time varying gain parameters
    paramRadarMT.metersPerSample = metersPerSample;
    paramRadarMT.rangeOffset = rangeOffset;
```

```
    paramRadarMT.rangeLoss = rangeLoss;
    paramRadarMT.referenceLoss = referenceLoss;
    % Radar parameters
    paramRadarMT.targetRcs = targetRcs;
    paramRadarMT.targetPos = targetPos;
    paramRadarMT.targetVel = targetVel;

    assignin('base','paramRadarMT',paramRadarMT);

end

function [propSpeed, fc, pulseBw, prf, fs, txGain, peakPower, ...
          matchingCoeff, metersPerSample, rangeOffset, rangeLoss, ...
          referenceLoss, targetRcs, targetPos, targetVel, lambda]  = calcParams()
    % Environment
    propSpeed = physconst('LightSpeed');   % Propagation speed
    fc = 10e9;           % Operating frequency
    lambda = propSpeed/fc;


    % Constraints
    maxRange = 5000;    % Maximum unambiguous range
    rangeRes = 50;      % Required range resolution
    pd = 0.9;            % Probability of detection
    pfa = 1e-6;          % Probability of false alarm
    tgtRcs = 1;         % Required target radar cross section
    numPulseInt = 10;  % Integrate 10 pulses at a time


    % Waveform parameters
    pulseBw = propSpeed/(2*rangeRes);    % Pulse bandwidth
    pulseWidth = 1/pulseBw;               % Pulse width
    prf = propSpeed/(2*maxRange);         % Pulse repetition frequency
    fs = 2*pulseBw;

    % Transmitter parameters
    snrMin = albersheim(pd, pfa, numPulseInt);
    txGain = 20;
    peakPower = ((4*pi)^3*noisepow(1/pulseWidth)*maxRange^4*...
        db2pow(snrMin))/(db2pow(2*txGain)*tgtRcs*lambda^2);


    % Matched filter parameters
    hwav = phased.RectangularWaveform(...
        'PulseWidth',1/pulseBw,...
        'PRF',prf,...
        'SampleRate',fs);
    matchingCoeff = getMatchedFilter(hwav);

    % Delay introduced due to filter
    matchingDelay = size(matchingCoeff,1)-1;

    % Time varying gain parameters
    fastTimeGrid = unigrid(0,1/fs,1/prf,'[)');
    rangeGates = propSpeed*fastTimeGrid/2;
    metersPerSample = rangeGates(2);
    rangeOffset = -rangeGates(2)*matchingDelay;
    rangeLoss = 2*fspl(rangeGates,lambda);
    referenceLoss = 2*fspl(maxRange,lambda);
```

```
    %Radar parameters
    targetRcs = [0.6 2.2 1.05 .5];
    targetPos = [1988.66 3532.630 3845.04 1045.04;...
                                  0 0 0 0 ; ...
                                  0 0 0 0 ];
    targetVel = zeros(3,4);

end
```