

```

function [xVec] = iq2if(IVec,QVec,Tl,fIF)
% IQ2IF : Convert baseband I and Q samples to intermediate frequency samples.
%
% Let  $x_l(m \cdot T_l) = I(m \cdot T_l) + j \cdot Q(m \cdot T_l)$  be a discrete-time baseband
% representation of a bandpass signal. This function converts  $x_l(n)$  to a
% discrete-time bandpass signal  $x(n) = I(n \cdot T) \cdot \cos(2 \cdot \pi \cdot f_{IF} \cdot n \cdot T) -$ 
%  $Q(n \cdot T) \cdot \sin(2 \cdot \pi \cdot f_{IF} \cdot n \cdot T)$  centered at the user-specified intermediate
% frequency  $f_{IF}$ , where  $T = T_l/2$ .
%
%
% INPUTS
%
% IVec----- N-by-1 vector of in-phase baseband samples.
%
% QVec----- N-by-1 vector of quadrature baseband samples.
%
% Tl----- Sampling interval of baseband samples (complex sampling
%            interval), in seconds.
%
% fIF----- Intermediate frequency to which the baseband samples will
%            be up-converted, in Hz.
%
%
% OUTPUTS
%
% xVec----- 2*N-by-1 vector of intermediate frequency samples with
%            sampling interval  $T = T_l/2$ .
%
%
%+-----+
% References:
%
%
%+=====+

R = 2; % for iq2if. (Given)
IVecResampled = interp(IVec,R);
QVecResampled = interp(QVec,R);
T=Tl/2;
n=(0:length(IVecResampled)-1)';
xVec = IVecResampled.*cos(2*pi*fIF*n*T)-QVecResampled.*sin(2*pi*fIF*n*T);

```

Not enough input arguments.

Error in iq2if (line 37)  
IVecResampled = interp(IVec,R);

