```
 function [IVec,QVec] = if2iq(xVec,T,fIF)
 % IF2IQ : Convert intermediate frequency samples to baseband I and Q samples.
 %
 % Let x(n) = I(n*T)*cos(2*pi*fIF*n*T)- Q(n*T)*sin(2*pi*fIF*n*T) be a
 % discrete-time bandpass signal centered at the user-specified intermediate
 % frequency fIF, where T is the bandpass sampling interval. Then this
 % function converts the bandpass samples to quadrature samples from a complex
 % discrete-time baseband representation of the form xl(m*Tl) = I(m*Tl) +
 % j*Q(m*Tl), where Tl = 2*T.
 %
 %
 % INPUTS
 %
 % xVec-------- N-by-1 vector of intermediate frequency samples with
 %              sampling interval T.
 %
 % T----------- Sampling interval of intermediate frequency samples, in
 %              seconds.
 %
 % fIF--------- Intermediate frequency of the bandpass signal, in Hz.
 %
 %
 % OUTPUTS
 %
 % IVec-------- N/2-by-1 vector of in-phase baseband samples.
 %
 % QVec-------- N/2-by-1 vector of quadrature baseband samples.
 %
 %
 %+------------------------------------------------------------------------------+
 % References:
 %
 %
 %+==============================================================================+

Tl        = 2*T; % Quadrature sampling interval
r         = Tl/T; % Decimation factor
n         = (0:length(xVec)-1)';
InPhaComp = xVec.*2.*cos(2*pi*fIF*n*T); % In-phase component or I; Unfiltered;
QuadComp  = xVec.*2.*sin(2*pi*fIF*n*T); % Quadrature component or Q; Unfiltered;

% By default, decimate uses a lowpass Chebyshev Type I infinite
% impulse response (IIR) filter of order 8.
IVec = decimate(InPhaComp,r);
QVec = decimate(QuadComp,r);
```

```
Not enough input arguments.

Error in if2iq (line 36)
Tl        = 2*T; % Quadrature sampling interval
```

Published with MATLAB® R2023a