

# FFT-TO-SYMBOLS (RECEIVER) (MATLAB)\*

Bryan Paul  
Aditya Jain

This work is produced by OpenStax-CNX and licensed under the  
Creative Commons Attribution License 3.0<sup>†</sup>

## Abstract

The incoming OFDM symbol was generated by essentially performing a parallel to serial transformation via an Inverse Fourier Transform (IFT). This is frequently done in the digital domain by using the Inverse Fast Fourier Transform (FFT) algorithm. Thus, we wish to recover our spectrum where the bits were present in parallel. For more information on why we use IFFT in the chain of the OFDM transmitter, see the appropriate module. **NOTE: The MATLAB library was used in the initial version of the design but has since been replaced by an updated LabVIEW module. Please see the LabVIEW portion of the receiver for the most current version.**

```
%% FFT to Symbols (RECEIVER)
% -----
% Description: Takes a received OFDM signal that has been demodulated to
%             baseband and corrected for frequency/phase offset, and
%             extracts the complex envelope of each carrier. This
%             converts the time domain signal into a discrete set of
%             symbols that can be mapped to binary words.
%
% Inputs: re - Real channel of OFDM symbol
%         im - Imaginary channel of OFDM symbol
%         num_subcarriers - Number of subcarriers used in OFDM symbol
% Outputs: symbols - Extracted complex envelopes of each subcarrier

function symbols = fft2sym(re, im, num_subcarriers)
%%
    size_of_fft = size(re,2); % Implicit size of fft
    output = fft(re+j*im); % Time domain signal was generated using ifft()
    symbols = [ output(2:1+num_subcarriers/2) output(size_of_fft+1-num_subcarriers/2:size_of_fft) ];
    % Save only the FFT samples that contain complex symbols as specified
    % by the number of subcarriers.
end
```

---

\*Version 1.2: Mar 28, 2010 10:20 am +0000

<sup>†</sup><http://creativecommons.org/licenses/by/3.0/>