Digital Transmitter: MATLAB Exercise for Quadrature Phase-Shift Keying[*]

Douglas L. Jones
Swaroop Appadwedula
Matthew Berry
Mark Haun
Jake Janovetz
Michael Kramer
Dima Moussa
Daniel Sachs
Brian Wade

This work is produced by The Connexions Project and licensed under the
Creative Commons Attribution License [†]

Abstract

You will simulate a quadrature phase-shift keying (QPSK) digital transmitter in MATLAB.

1 MATLAB Simulation

MATLAB is commonly used to design filters and determine frequency responses of systems, but it is also very useful as a simulation tool.

Use the following MATLAB code skeleton to simulate the QPSK transmitter from Digital Transmitter: Introduction to Quadrature Phase-Shift Keying[1] and fill in the incomplete portions. Note that the code is not complete and will not execute properly as written. How does the spectrum of the transmitted signal change with T_{symb}? How do you interpret the figure created by `plot(rI,rQ)`?

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

[*]Version 2.9: Feb 25, 2004 12:00 pm -0600

[†]http://creativecommons.org/licenses/by/1.0

[1]"Digital Transmitter: Introduction to Quadrature Phase-Shift Keying", Figure 1: QPSK Transmitter <http://cnx.org/content/m10042/latest/#fig1>

```
% MATLAB Code Skeleton for QPSK Digital Transmitter

% Generate random bits
bits_per_symbol=2;
num_symbols=128;
numbits=bits_per_symbol*num_symbols;
bits=rand(1,numbits)>0.5;

 Tsymb = 16;                  % symbol length
 omega = pi/2;                % carrier frequency

 %%%%%%%%%%%%%%%%%%%%%%%%
 % Transmitter section
                            % initialize transmit sequence
 t = zeros(1,num_symbols*Tsymb);
 i = 1;                       % initialize bit index
 n = 1;                       % initialize time index

 while (n <= num_symbols*Tsymb)
  if ( bits(i:i+1) == [ 0 0])
      Igain = 1/sqrt(2);
      Qgain = 1/sqrt(2);
  % ------>Insert code here<-------

  end;
  i = i+2;                    % next 2 bits

  % Generate symbol to be transmitted
  t(n:n+Tsymb-1) =    %------>Insert code here<-------

  n = n+Tsymb;                % next symbol
 end;

 % Show the transmitted signal and its spectrum
 % ------>Insert code here<-------

 % Show the transmitted signal constellation
 rI = t.*cos(omega*[1:num_symbols*Tsymb]);
 rQ = t.*sin(omega*[1:num_symbols*Tsymb]);

 % Filter out the double-frequency term
 low_pass=fir1(512,0.5);
 rI=conv(rI,low_pass);
 rQ=conv(rQ,low_pass);
 figure;
 plot(rI,rQ);
```