```
function [out] = MSKmod(data, varargin)
%
%args: A, FS, SYMBOL_FREQUENCIES, DATA_RATE MODULATION_T
%

numvarargs = length(varargin);
if numvarargs > 5
    error('too many arguments');
end

optargs = {1, 44100, 551.25, 1600, 551.25 , 1/551.25*40};

% 1.58 Khz, 1.36kHz


optargs(1:numvarargs) = varargin; %set unset args to defaults

% Load default arguments
[A,FS,DATA_RATE,CENTER_FREQ, clock_ref_freq, clock_ref_duration] =
 optargs{:};

% MSK symbol frequencies defined
SYMBOL_FREQUENCIES = [CENTER_FREQ - DATA_RATE/4 CENTER_FREQ +
 DATA_RATE/4];
% Sampling period
SAMPLING_T = 1/FS;
% Modulation period, period over which a single bit is modulated
MODULATION_T = 1/DATA_RATE;

validateattributes(data,{'numeric'},{'binary'})
validateattributes(data,{'numeric'},{'row'})


fc = CENTER_FREQ;

% Change input from being 0-1 based to 1 and -1 based
nrz_data = ones(size(data));
nrz_data(data == 0) = -1;


startTime = 0; %start=0 seconds
endTime = MODULATION_T*length(data)+clock_ref_duration; %end=1 seconds
Trange = startTime:SAMPLING_T:endTime-SAMPLING_T;

curTime = startTime;
y=[zeros(1,FS/5),
 A*cos(2*pi*clock_ref_freq*(startTime:SAMPLING_T:clock_ref_duration -
 SAMPLING_T)), zeros(1,FS/5)];
curTime = curTime + clock_ref_duration;

idx = 1;
phase_k = 0;
```

```
phase = [];
for d_k=nrz_data
    % get the time vector for this bit
    t = (curTime+(0:SAMPLING_T:MODULATION_T-SAMPLING_T));
    %
    if idx > 1
        % Phase progresses 90 degrees each symbol
        phase_k = phase_clip(phase_k + (idx-1)*pi/2*(nrz_data(idx-1) -
 nrz_data(idx)));
    end
    % Keep track of phase vectors
    phase = [phase phase_k];
    y = [y cos(2*pi*(fc+d_k/(4*MODULATION_T))*t + phase_k)];
    curTime = curTime + MODULATION_T;
    idx = idx+1;
end
% PLOT TOOLS
% subplot(3,1,1);
% title('TX binary data');
% stairs(0:length(data)-1,data, '-.or');
% ylim([-.2 1.2]);
% subplot(3,1,2);
% title('Phase output')
% plot(0:length(data)-1,phase,'.-b');
% ylim([-.1 2.1*pi]);
% ylabel('Radians');
% xlabel('Time');
%
% subplot(3,1,3);
% title('TX output waveform')
% plot(Trange,y,'.-b');
% ylim([-1.2*A 1.2*A]);
% ylabel('Amplitude');
% xlabel('Time');
out = [y zeros(1,FS/5)];
end

function [phase_angle] = phase_clip(phase_angle)
while phase_angle > 2*pi
    phase_angle = phase_angle - 2*pi;
end
while phase_angle < 0
    phase_angle = phase_angle + 2*pi;
end
end
```

Published with MATLAB® R2015b