%% Integral Pulse Frequency Modulation Model of heart beats

% integrates an input signal until it reaches a "threshold of unity"; at this point a pulse is produces and the integrator is reset to zero

% MATLABs built in modulate function will perform this for us using a

% rectangular integral approximation

close all;

clear all;

%% Constants

OFFSET = 1;

dt = 0.001;

t = 1:1000000;

%% Inputs based on the Paper

s = sin(2\*pi\*0.025\*t\*dt); % "period of the sympathetic oscillator is set to ?40 s, "

p = sin(2\*pi\*0.344\*t\*dt); % "parasympathetic oscillator is set to a period of ?3 s"..."This oscillator has a value of ?p larger than ?s, typically at the modeled respiration frequency." Normal Respiration rate is 12-20 breaths per minute

Cs = 0.01

Cp = 0.01 %balanced parasymathetic: sympathetic activity

HR0 = 1/.85; %HR corresponds to an R-R interval of 850 ms; HR is a variable parameter that represents mean heart rate

heart\_rate\_inputs = (Cs\*s)+(Cp\*p) + HR0;

% Frequency Modulate Sum of Inputs to the model

thresh = 1;

sum = 0;

beat\_times = zeros(1);

rr\_t = zeros(1);

rr\_y = zeros(1);

for i = 2:1000000

time = i\*dt;

sum = sum + dt\*heart\_rate\_inputs(i);

if sum > thresh

rr\_y(end+1) = time - beat\_times(end);

rr\_t(end+1) = time;

beat\_times(end+1) = time;

thresh = thresh + 1;

end

end

rr\_interval = diff(beat\_times);

length\_rr = length(rr\_interval);

%% Calculate Descriptive Statistics

rr = rr\_interval(1:(end-OFFSET)) .\* 1000;

rr\_delay = rr\_interval((OFFSET+1):end) .\* 1000;

x\_y = rr ./ rr\_delay;

y\_x = rr\_delay ./ rr;

data\_sd1 = std(y\_x); %perpendicular to line of identity

data\_sd2 = std(x\_y); %parallel to line of identity

%% Graph RR intervals

subplot(2,2,1);

plot(t\*dt, heart\_rate\_inputs, 'b.');

xlabel('Time (s)')

ylabel('Amplitude')

legend('Input to IPFM');

subplot(2,2,2);

plot(rr\_t, rr\_y, 'c.');

xlabel('Time (s)')

ylabel('Duration in Seconds')

legend('RR Interval');

subplot(2,1,2);

OFFSET = 1;

plot(rr, rr\_delay, 'k.'); %covert from seconds to ms

models\_ellipse = fit\_ellipse(rr, rr\_delay, gca) %print to see

xlabel('RR(n) in ms');

str = sprintf('RR(n + %d) in ms',OFFSET);

ylabel(str);

legend('Poincare Plot');