```
%% PART I
clear; close all; clc;

addpath(genpath('../'))

Nmc = 1000;                 % Number of Monte Carlo Runs

sigma_a = 0.5;              % Standard Deviation of a
sigma_b = 0.2;              % Standard Deviation of b

y_ = 0;                     % Expected Mean
sigma_y = sqrt(9*sigma_a^2 ...
    + 16*sigma_b^2);        % Expected Standard Devation

for i = 1:Nmc
    a = sigma_a*randn(1);   % State
    b = sigma_b*randn(1);   % State
    y(i) = 3*a - 4*b;       % Combined State
end
y_s = mean(y);              % Sample Mean
sigma_ys = std(y);          % Sample Standard Deviation

g_x = linspace(min(y), max(y), Nmc);
g_y = normpdf(g_x, y_, sigma_y);

fprintf('Sample Mean from Monte Carlo: %0.3g\n', y_s);
fprintf('Sample Standard Deviation from Monte Carlo: %0.3g\n\n', sigma_ys);
fprintf('Expected Mean: %0.3g\n', y_);
fprintf('Expected Standard Deviation: %0.3g\n\n', sigma_y);

figure();
hold('on')
title('Histogram of Y');
histogram(y, 'Normalization','pdf');
plot(g_x, g_y, 'LineWidth', 2);
xlabel('Y-Value');
ylabel('Occurrences');
ax = gca;
ax.FontSize = 14;

f = gcf;
exportgraphics(f,'figures/p1_histogram.png','Resolution',300)

%% PART II
clear;

Nmc = 1000;

dt = 1;                 % Time Step [s]
time = 0:dt:10*60;      % Time [s]
sigma1 = 0.1;           % Standard Deviation 1
sigma2 = 0.01;          % Standard Deviation 2

x1 = zeros(length(time),Nmc);     % Integrated White Noise Vector
x2 = zeros(length(time),Nmc);     % Integrated White Noise Vector
for i = 1:Nmc
    for j = 2:length(time)
        x1(j,i) = x1(j-1,i) + (sigma1*randn(1))*dt;
        x2(j,i) = x2(j-1,i) + (sigma2*randn(1))*dt;
    end
end

x1_ = mean(x1,2);
sigma_x1_n = std(x1,1,2);
x2_ = mean(x2,2);
sigma_x2_n = std(x2,1,2);

sigma_x1_a = sigma1.*sqrt(time.*dt);
sigma_x2_a = sigma2.*sqrt(time.*dt);

secs = duration(0,0,time-time(1));
figure();
hold('on');
title('Mean of Integrated White Noise vs. Time');
plot(secs, x1_, 'LineWidth', 2);
plot(secs, x2_, 'LineWidth', 2);
yline(0,'k');
xlabel('Time (hh:mm:ss)');
ylabel('Mean');
legend('\sigma_w=0.1', '\sigma_w=0.01', 'Zero-Mean');
ax = gca;
ax.FontSize = 14;

f = gcf;
exportgraphics(f,'figures/p2_rw_mean.png','Resolution',300)

figure();
tiledlayout(2,1);
nexttile();
hold('on');
title('Standard Deviation of Integrated Noise with \sigma_w=0.1 vs. Time')
plot(secs, sigma_x1_a, 'LineWidth', 2);
plot(secs, sigma_x1_n, 'LineWidth', 2);
xlabel('Time (hh:mm:ss)');
ylabel('Standard Deviation');
legend('Analytical', 'Numeric');
ax = gca;
ax.FontSize = 14;

nexttile();
hold('on');
title('Standard Deviation of Integrated Noise with \sigma_w=0.01 vs. Time')
plot(secs, sigma_x2_a, 'LineWidth', 2);
plot(secs, sigma_x2_n, 'LineWidth', 2);
xlabel('Time (hh:mm:ss)');
ylabel('Standard Deviation');
legend('Analytical', 'Numeric');
ax = gca;
ax.FontSize = 14;

f = gcf;
exportgraphics(f,'figures/p2_rw_sigma.png','Resolution',300)

figure();
hold('on');
title('Integrated White Noise vs. Time');
```

```
subtitle('Run #1 of the Monte Carlo')
plot(secs, x1(:,1), 'LineWidth', 2);
xlabel('Time (hh:mm:ss)');
ylabel('Integrate White Noise');
ax = gca;
ax.FontSize = 14;

f = gcf;
exportgraphics(f,'figures/p2_rw_run.png','Resolution',300)

g_xt1 = linspace(min(x1(2,:)), max(x1(2,:)), Nmc);
g_xt2 = linspace(min(x1(round(length(time)/4),:)), max(x1(round(length(time)/4),:)),↙
Nmc);
g_xt3 = linspace(min(x1(3*round(length(time)/4),:)), max(x1(3*round(length(time)/4),:)),↙
Nmc);
g_xt4 = linspace(min(x1(length(time),:)), max(x1(length(time),:)), Nmc);
g_yt1 = normpdf(g_xt1, 0, sigma_x1_a(2));
g_yt2 = normpdf(g_xt2, 0, sigma_x1_a(round(length(time)/4)));
g_yt3 = normpdf(g_xt3, 0, sigma_x1_a(3*round(length(time)/4)));
g_yt4 = normpdf(g_xt4, 0, sigma_x1_a(length(time)));

tiledlayout(2,2)
nexttile
hold('on')
title(['Histogram of t=',num2str(time(2))]);
histogram(x1(2,:), 'Normalization','pdf');
plot(g_xt1, g_yt1, 'LineWidth', 2);
xlabel('Value');
ylabel('Occurrences');
ax = gca;
ax.FontSize = 14;

nexttile
hold('on')
title(['Histogram of t=', num2str(time(round(length(time)/4)))]);
histogram(x1(round(length(time)/4),:), 'Normalization','pdf');
plot(g_xt2, g_yt2, 'LineWidth', 2);
xlabel('Value');
ylabel('Occurrences');
ax = gca;
ax.FontSize = 14;

nexttile
hold('on')
title(['Histogram of t=', num2str(time(3*round(length(time)/4)))]);
histogram(x1(3*round(length(time)/4),:), 'Normalization','pdf');
plot(g_xt3, g_yt3, 'LineWidth', 2);
xlabel('Value');
ylabel('Occurrences');
ax = gca;
ax.FontSize = 14;

nexttile
hold('on')
title(['Histogram of t=', num2str(time(length(time)))]);
histogram(x1(length(time),:), 'Normalization','pdf');
plot(g_xt4, g_yt4, 'LineWidth', 2);
xlabel('Value');
```

```
ylabel('Occurrences');
ax = gca;
ax.FontSize = 14;

f = gcf;
exportgraphics(f,'figures/p2_bonus.png','Resolution',300)

% B)
tau1 = 1;               % Time Constant [s]
tau2 = 100;             % Time Constant [s]

x1t1 = zeros(length(time),Nmc);
x1t2 = zeros(length(time),Nmc);
x2t1 = zeros(length(time),Nmc);
x2t2 = zeros(length(time),Nmc);
for i = 1:Nmc
    for j = 1:length(time)-1
        x1t1_dot = -(1/tau1)*x1t1(j,i) + sigma1*randn(1);
        x1t2_dot = -(1/tau2)*x1t2(j,i) + sigma1*randn(1);
        x2t1_dot = -(1/tau1)*x2t1(j,i) + sigma2*randn(1);
        x2t2_dot = -(1/tau2)*x2t2(j,i) + sigma2*randn(1);

        x1t1(j+1,i) = x1t1(j,i) + x1t1_dot*dt;
        x1t2(j+1,i) = x1t2(j,i) + x1t2_dot*dt;
        x2t1(j+1,i) = x2t1(j,i) + x2t1_dot*dt;
        x2t2(j+1,i) = x2t2(j,i) + x2t2_dot*dt;
    end
end

x1t1_ = mean(x1t1,2);
sigma_x1t1_n = std(x1t1,1,2);
x1t2_ = mean(x1t2,2);
sigma_x1t2_n = std(x1t2,1,2);
x2t1_ = mean(x2t1,2);
sigma_x2t1_n = std(x2t1,1,2);
x2t2_ = mean(x2t2,2);
sigma_x2t2_n = std(x2t2,1,2);

At1 = (1 - (dt/tau1));
At2 = (1 - (dt/tau2));

sigma_x1t1_a = sigma1*dt.*sqrt((At1.^(2*time) - 1)./(At1^2 - 1));
sigma_x1t2_a = sigma1*dt.*sqrt((At2.^(2*time) - 1)./(At2^2 - 1));
sigma_x2t1_a = sigma2*dt.*sqrt((At1.^(2*time) - 1)./(At1^2 - 1));
sigma_x2t2_a = sigma2*dt.*sqrt((At2.^(2*time) - 1)./(At2^2 - 1));

figure();
hold('on');
title('Mean of 1st Order Guas-Markov Process vs. Time');
plot(secs, x1t1_, 'LineWidth', 2);
plot(secs, x1t2_, 'LineWidth', 2);
plot(secs, x2t1_, 'LineWidth', 2);
plot(secs, x2t2_, 'LineWidth', 2);
yline(0,'k', 'LineWidth', 2)
xlabel('Time (hh:mm:ss)');
ylabel('Mean');
legend('\sigma_w=0.1; \tau=1', '\sigma_w=0.1; \tau=100', ['\sigma_w=0.01; ' ...
    '\tau=1'], '\sigma_w=0.01; \tau=100', 'Zero-Mean');
```

```
ax = gca;
ax.FontSize = 14;

f = gcf;
exportgraphics(f,'figures/p2_gm_mean.png','Resolution',300)

figure();
tiledlayout(2,1);
nexttile();
hold('on');
title('Standard Deviation of 1st Order Gauss-Markov Process with \sigma_w=0.1 vs. Time')
plot(secs, sigma_x1t1_a, 'LineWidth', 2);
plot(secs, sigma_x1t1_n, 'LineWidth', 2);
plot(secs, sigma_x1t2_a, 'LineWidth', 2);
plot(secs, sigma_x1t2_n, 'LineWidth', 2);
xlabel('Time (hh:mm:ss)');
ylabel('Standard Deviation');
legend('Analytical; \tau=1', 'Numeric; \tau=1', 'Analytical; \tau=100', 'Numeric;↙
\tau=100');
ax = gca;
ax.FontSize = 14;

nexttile();
hold('on');
title('Standard Deviation of 1st Order Gauss-Markov Process with \sigma_w=0.01 vs. Time')
plot(secs, sigma_x2t1_a, 'LineWidth', 2);
plot(secs, sigma_x2t1_n, 'LineWidth', 2);
plot(secs, sigma_x2t2_a, 'LineWidth', 2);
plot(secs, sigma_x2t2_n, 'LineWidth', 2);
xlabel('Time (hh:mm:ss)');
ylabel('Standard Deviation');
legend('Analytical; \tau=1', 'Numeric; \tau=1', 'Analytical; \tau=100', 'Numeric;↙
\tau=100');
ax = gca;
ax.FontSize = 14;

f = gcf;
exportgraphics(f,'figures/p2_gm_sigma.png','Resolution',300)

%% PART V
clear;

Xs = [  0 300;
      100 400;
      700 400;
      800 300];
Xu = [401 0];
Xr = [400 0];

r_ur = norm(Xu - Xr);
rho_u = vecnorm(Xs - Xu, 2, 2);
rho_r = vecnorm(Xs - Xr, 2, 2);

% 2 SVs
dX = 1e3*ones(1,2);
Xu_2SVs = zeros(1,2);
while norm(dX) > 1e-4
    r = vecnorm((Xs(1:2,:) - Xu_2SVs), 2 ,2);
    U = (Xs(1:2,:) - Xu_2SVs)./r;
    H = -U;
    dX = pinv(H)*(rho_u(1:2) - r);
    Xu_2SVs = Xu_2SVs + dX';
end

DOP_2SVs = inv(H'*H);
PDOP_2SVs = sqrt(sum(diag(DOP_2SVs(1:2,1:2))));

fprintf('PDOP w/ 2SVs: %0.4g\n', PDOP_2SVs)

% 4 SVs
dX = 1e3*ones(1,2);
Xu_4SVs = zeros(1,2);
while norm(dX) > 1e-4
    r = vecnorm((Xs - Xu_4SVs), 2 ,2);
    U = (Xs - Xu_4SVs)./r;
    H = -U;
    dX = pinv(H)*(rho_u - r);
    Xu_4SVs = Xu_4SVs + dX';
end

DOP_4SVs = inv(H'*H);
PDOP_4SVs = sqrt(sum(diag(DOP_4SVs(1:2,1:2))));

fprintf('PDOP w/ 4SVs: %0.4g\n', PDOP_4SVs)


% 4 SVs + Clock Bias
dX = 1e3*ones(1,3);
Xu_4SVs_bias = zeros(1,3);
while norm(dX) > 1e-4
    r = vecnorm((Xs - Xu_4SVs_bias(1:2)), 2 ,2);
    U = (Xs - Xu_4SVs_bias(1:2))./r;
    H = [-U ones(length(U),1)];
    rho_hat = r + Xu_4SVs_bias(3);
    dX = pinv(H)*(rho_u - rho_hat);
    Xu_4SVs_bias = Xu_4SVs_bias + dX';
end

DOP_4SVs_bias = inv(H'*H);
PDOP_4SVs_bias = sqrt(sum(diag(DOP_4SVs_bias(1:2,1:2))));

fprintf('PDOP w/ 4SVs and a Clock Bias: %0.4g\n', PDOP_4SVs_bias)

% DGPS Single Difference
rho_ru = rho_r - rho_u;
r = vecnorm((Xs - Xr(1:2)), 2 ,2);
U = (Xs - Xr(1:2))./r;
H = [-U ones(length(U),1)];
Xru_DGPS1 = pinv(H)*(rho_ru);

DOP_DGPS1 = inv(H'*H);
PDOP_DGPS1 = sqrt(sum(diag(DOP_DGPS1(1:2,1:2))));

fprintf('PDOP w/ Single Difference: %0.4g\n', PDOP_DGPS1)

% DGPS Double Difference
```

```
rho_ru = rho_r - rho_u;
rho_ru_double = rho_ru(2:end) - rho_ru(1);
r = vecnorm((Xs - Xr(1:2)), 2 ,2);
U = (Xs - Xr(1:2))./r;
U = U(2:end,:) - U(1,:);
H = -U;
Xru_DGPS2 = pinv(H)*(rho_ru_double);

DOP_DGPS2 = inv(H'*H);
PDOP_DGPS2 = sqrt(sum(diag(DOP_DGPS2(1:2,1:2))));

fprintf('PDOP w/ Single Difference: %0.4g\n', PDOP_DGPS2)

% Unit Variance Accuracies
sigma2 = 1;
acc_2SVs = PDOP_2SVs*sigma2;
acc_4SVs = PDOP_4SVs*sigma2;
acc_4SVs_bias = PDOP_4SVs_bias*sigma2;
acc_DGPS1 = PDOP_DGPS1*sqrt(2)*sigma2;
acc_DGPS2 = PDOP_DGPS2*sqrt(2*2)*sigma2;

x = ["2 SVs" "4 SVs" "4 SVs + Clock Bias" "Single Diff." "Double Diff."];
y = [PDOP_2SVs PDOP_4SVs PDOP_4SVs_bias PDOP_DGPS1 PDOP_DGPS2;
     acc_2SVs acc_4SVs acc_4SVs_bias acc_DGPS1 acc_DGPS2];

figure();
hold("on");
title('PDOP & Errors for Varying Scenarios');
bar(x,y);
xlabel('Scenario');
ylabel('DOP/RMS Error (m)');
legend('PDOP', 'Position Error');
ax = gca;
ax.FontSize = 14;

f = gcf;
exportgraphics(f,'figures/p5_pdop.png','Resolution',300)

%% PART VI
clear;
filename = 'RCVR_S1_data.mat';
c = physconst('LightSpeed');
fL1 = 1575.42e6;

% Load Data
addpath(genpath("./"));
data = load(filename).RCVR_S1;
ephem = data.ephem;
psrL1 = data.measurements.L1.psr;
dopL1 = data.measurements.L1.doppler;
psrVarL1 = data.measurements.L1.psr_variance;
time = data.GPS_time.seconds;

prns = 1:32;
supersvprns = prns(sum(~isnan(psrL1)) == length(time));
supersvprns = supersvprns(1:4);

X0 = zeros(8,1);
```

```
Xu = zeros(length(X0),length(time));
[svPos, svVel, svB, svD, svPrns] = sv_positions(ephem, psrL1(1,:), time(1));

temp = 1:length(svPrns);
newPrns = temp(ismember(svPrns, supersvprns));

Xs = [svPos svB svVel svD];
Xs = Xs(newPrns,:);

rhoL1 = psrL1(1, supersvprns)' + c*svB(newPrns);
rho_dot = (-c/fL1)*dopL1(1, supersvprns)' + c*svD(newPrns);

Xu(:,1) = gnssPVT(Xs, X0, rhoL1', rho_dot');

for i = 2:length(time)
    [svPos, svVel, svB, svD, svPrns] = sv_positions(ephem, psrL1(i,:), time(i));

    temp = 1:length(svPrns);
    newPrns = temp(ismember(svPrns, supersvprns));

    Xs = [svPos svB svVel svD];
    Xs = Xs(newPrns,:);

    rhoL1 = psrL1(i, supersvprns)' + c*svB(newPrns);
    rho_dot = (-c/fL1)*dopL1(i, supersvprns)' + c*svD(newPrns);
    [Xu(:,i), HL1] = gnssPVT(Xs, Xu(:,i-1), rhoL1', rho_dot');

    DOP = inv(HL1'*HL1);
    PDOP_4SVs(i) = sqrt(sum(diag(DOP(1:3,1:3))));
end

secs = duration(0,0,time-time(1));
figure();
plot(secs, PDOP_4SVs, '*');
title('PDOP w/ 4SVs vs. Time');
xlabel('Time (hh:mm:ss)');
ylabel('PDOP');
ax = gca;
ax.FontSize = 14;

f = gcf;
exportgraphics(f,'figures/p6_4svs_pdop.png','Resolution',300)

prns = 1:32;
supersvprns = prns(sum(~isnan(psrL1)) == length(time));

X0 = zeros(8,1);
Xu = zeros(length(X0),length(time));
[svPos, svVel, svB, svD, svPrns] = sv_positions(ephem, psrL1(1,:), time(1));

temp = 1:length(svPrns);
newPrns = temp(ismember(svPrns, supersvprns));

Xs = [svPos svB svVel svD];
Xs = Xs(newPrns,:);

rhoL1 = psrL1(1, supersvprns)' + c*svB(newPrns);
rho_dot = (-c/fL1)*dopL1(1, supersvprns)' + c*svD(newPrns);
```

```
Xu(:,1) = gnssPVT(Xs, X0, rhoL1', rho_dot');

for i = 2:length(time)
    [svPos, svVel, svB, svD, svPrns] = sv_positions(ephem, psrL1(i,:), time(i));

    temp = 1:length(svPrns);
    newPrns = temp(ismember(svPrns, supersvprns));

    Xs = [svPos svB svVel svD];
    Xs = Xs(newPrns,:);

    rhoL1 = psrL1(i, supersvprns)' + c*svB(newPrns);
    rho_dot = (-c/fL1)*dopL1(i, supersvprns)' + c*svD(newPrns);
    [Xu(:,i), HL1] = gnssPVT(Xs, Xu(:,i-1), rhoL1', rho_dot');

    DOP = inv(HL1'*HL1);
    PDOP_8SVs(i) = sqrt(sum(diag(DOP(1:3,1:3))));
end

secs = duration(0,0,time-time(1));
figure();
plot(secs, PDOP_8SVs, '*');
title('PDOP w/ 8SVs vs. Time');
xlabel('Time (hh:mm:ss)');
ylabel('PDOP');
ax = gca;
ax.FontSize = 14;

f = gcf;
exportgraphics(f,'figures/p6_8svs_pdop.png','Resolution',300)

%% PART VII
clear;

prn4 = genCA(4,1023);
figure();
hold('on');
title('PRN #4 C/A Code');
stem(prn4(1:16), 'LineWidth', 2);
stem(prn4(end-15:end), 'LineWidth', 2);
xlabel('Index');
ylabel('C/A Code');
legend('First 16', 'Last 16');
ax = gca;
ax.FontSize = 14;

f = gcf;
exportgraphics(f,'figures/p7_prn4_16.png','Resolution',300)

prn4_2046 = genCA(4,2046);
prn4_1023_1 = prn4_2046(1:1023);
prn4_1023_2 = prn4_2046(1024:2046);
corr4 = scorr(prn4_1023_1, prn4_1023_2);

figure();
hold('on');
title('PRN #4 Autocorrelation');
```

```
plot(corr4, 'LineWidth', 2);
xlabel('Shift');
ylabel('Correlation');
ax = gca;
ax.FontSize = 14;

f = gcf;
exportgraphics(f,'figures/p7_prn4_acorr.png','Resolution',300)

prn7 = genCA(7);
figure();
hold('on');
title('PRN #7 C/A Code');
stem(prn7(1:16), 'LineWidth', 2);
stem(prn7(end-15:end), 'LineWidth', 2);
xlabel('Index');
ylabel('C/A Code');
legend('First 16', 'Last 16');
ax = gca;
ax.FontSize = 14;

f = gcf;
exportgraphics(f,'figures/p7_prn7_16.png','Resolution',300)

%% PART VIII
clear;

% Q2 Variables
N = 100;
a1 = 2*ceil(rand(N,1)-0.5)-1;
a2 = 2*ceil(rand(N,1)-0.5)-1;
N_long = 1000;
a1_long = 2*ceil(rand(N_long,1)-0.5)-1;
a2_long = 2*ceil(rand(N_long,1)-0.5)-1;

% HW1-2A
figure();
tiledlayout(2,1);
nexttile();
histogram(a1);
xlabel("Value");
ylabel("Occurences");
title("Histogram of Random Sequence 1");
nexttile();
histogram(a2);
xlabel("Value");
ylabel("Occurences");
title("Histogram of Random Sequence 2");

f = gcf;
exportgraphics(f,'figures/p8_hw1_2a.png','Resolution',300)

% HW1-2B
figure();
tiledlayout(2,1);
nexttile();
plot(abs(fft(a1)));
title("PSD of Random Sequence 1");
```

```
nexttile();
plot(abs(fft(a2)));
title("PSD of Random Sequence 2");

f = gcf;
exportgraphics(f,'figures/p8_hw1_2b.png','Resolution',300)

% HW1-2C
acorr1 = scorr(a1);
acorr2 = scorr(a2);
figure();
tiledlayout(2,1);
nexttile();
plot(-N:N, acorr1);
title("Autocorrelation of Random Sequence 1");
nexttile();
plot(-N:N, acorr2);
title("Autocorrelation of Random Sequence 2");

f = gcf;
exportgraphics(f,'figures/p8_hw1_2c.png','Resolution',300)

% HW1-2D
xcorr12 = scorr(a1,a2);
xcorr21 = scorr(a2,a1);
figure();
tiledlayout(2,1);
nexttile();
plot(-N:N, xcorr12);
title("Cross Correlation of Random Sequence 1 w/ 2");
nexttile();
plot(-N:N, xcorr21);
title("Cross Correlation of Random Sequence 2 w/ 1");

f = gcf;
exportgraphics(f,'figures/p8_hw1_2d.png','Resolution',300)

% HW1-2BONUS
acorr1_long = scorr(a1_long);
acorr2_long = scorr(a2_long);
figure();
tiledlayout(2,1);
nexttile();
plot(-N_long:N_long, acorr1_long);
title("Autocorrelation of Random Sequence 1");
nexttile();
plot(-N_long:N_long, acorr2_long);
title("Autocorrelation of Random Sequence 2");

f = gcf;
exportgraphics(f,'figures/p8_hw1_2bonus1.png','Resolution',300)

xcorr12_long = scorr(a1_long,a2_long);
xcorr21_long = scorr(a2_long,a1_long);
figure();
tiledlayout(2,1);
nexttile();
plot(-N_long:N_long, xcorr12_long);
```

```
title("Cross Correlation of Random Sequence 1 w/ 2");
nexttile();
plot(-N_long:N_long, xcorr21_long);
title("Cross Correlation of Random Sequence 2 w/ 1");

f = gcf;
exportgraphics(f,'figures/p8_hw1_2bonus2.png','Resolution',300)

prn4 = genCA(4);
prn7 = genCA(7);

prn4(prn4 == 0) = -1;
prn7(prn7 == 0) = -1;

figure();
histogram(prn4);
title('PRN #4 Histogram');
xlabel('Value');
ylabel('Occurrences');
ax = gca;
ax.FontSize = 14;

f = gcf;
exportgraphics(f,'figures/p8_prn4_hist.png','Resolution',300)

figure();
histogram(prn7);
title('PRN #7 Histogram');
xlabel('Value');
ylabel('Occurences');
ax = gca;
ax.FontSize = 14;

f = gcf;
exportgraphics(f,'figures/p8_prn7_hist.png','Resolution',300)

figure();
plot(abs(fft(prn4)), 'LineWidth', 2);
title('PRN #4 Power Spectral Density');
xlabel('Frequency (Hz)');
ylabel('Power');
ax = gca;
ax.FontSize = 14;

f = gcf;
exportgraphics(f,'figures/p8_prn4_psd.png','Resolution',300)

figure();
plot(abs(fft(prn7)), 'LineWidth', 2);
title('PRN #7 Power Spectral Density');
xlabel('Frequency (Hz)');
ylabel('Power');
ax = gca;
ax.FontSize = 14;

f = gcf;
exportgraphics(f,'figures/p8_prn7_psd.png','Resolution',300)
```

```
figure();
plot(scorr(prn4), 'LineWidth', 2)
title('PRN #4 Autocorrelation');
xlabel('Shift');
ylabel('Correlation');
ax = gca;
ax.FontSize = 14;

f = gcf;
exportgraphics(f,'figures/p8_prn4_acorr.png','Resolution',300)

figure();
plot(scorr(prn7), 'LineWidth', 2)
title('PRN #7 Autocorrelation');
xlabel('Shift');
ylabel('Correlation');
ax = gca;
ax.FontSize = 14;

f = gcf;
exportgraphics(f,'figures/p8_prn7_acorr.png','Resolution',300)

figure()
plot(scorr(prn4,prn7), 'LineWidth', 2)
title('PRN #4 & #7 Cross Correlation');
xlabel('Shift');
ylabel('Correlation');
ax = gca;
ax.FontSize = 14;

f = gcf;
exportgraphics(f,'figures/p8_prn47_xcorr.png','Resolution',300)

%% PART IX
clear;

fL1 = 1575.42e6;
SR = (10*fL1);
time = (0:10*1023);
carrL1 = exp(2*pi*fL1*time*-1i);

prn4 = genCA(4);
prn7 = genCA(7);

prn4(prn4 == 0) = -1;
prn7(prn7 == 0) = -1;
prn4 = upsample(prn4,length(time));
prn7 = upsample(prn7,length(time));

prn4_carrL1 = prn4.*carrL1;
prn7_carrL1 = prn7.*carrL1;
psd4 = fftshift(abs((fft(prn4_carrL1))));
psd7 = fftshift(abs((fft(prn7_carrL1))));

fx = linspace(SR/-2, SR/2, length(time)) + fL1;

figure();
plot(fx, psd4, 'LineWidth', 2);
```

```
title('PRN #4 w/ L1 Power Spectral Density');
xlabel('Frequency (Hz)');
ylabel('Power');
ax = gca;
ax.FontSize = 14;

f = gcf;
exportgraphics(f,'figures/p9_prn4_psd.png','Resolution',300)

figure();
plot(fx, psd7, 'LineWidth', 2);
title('PRN #7 w/ L1 Power Spectral Density');
xlabel('Frequency (Hz)');
ylabel('Power');
ax = gca;
ax.FontSize = 14;

f = gcf;
exportgraphics(f,'figures/p9_prn7_psd.png','Resolution',300)
```