```
clc
clear
dataSheetCalibration = [15 2.75; 20 2.55; 30 2; 40 1.55; 50 1.25; 60 1.05; 70 0.9; 80 0.8; 90
[robotArduino, rawRangeIn, blinkLED, panServo, tiltServo] = setUp('COM6');
distanceData = calibrationLoop(robotArduino,rawRangeIn,blinkLED);
cftool(dataSheetCalibration(:,2),dataSheetCalibration(:,1))
plotCalibrationData(distanceData)
```

```
function [robotArduino, rawRangeIn, blinkLED, panServo, tiltServo] = setUp(COMPORT)

    robotArduino = arduino(COMPORT, 'Uno', 'Libraries', 'Servo');

    blinkLED = 'D13';
    configurePin(robotArduino, blinkLED, 'DigitalOutput');

    rawRangeIn = 'A1';
    configurePin(robotArduino,rawRangeIn,'AnalogInput');

    panServo = servo(robotArduino, 'D9', 'MinPulseDuration', 10*10^-6,...
    'MaxPulseDuration', 1925*10^-6);
    tiltServo = servo(robotArduino, 'D11', 'MinPulseDuration', 10*10^-6,...
    'MaxPulseDuration', 1925*10^-6);
    writePosition(panServo, 0.5); % always start servo-command at 0.5
    writePosition(tiltServo, 0.5); % always start servo-command at 0.5
    pause(5.0); % wait for arduino to send stable PWM
    pause(2.0);
end

function [] = Blink(a,LED,n)
        for bIndex = 1:n
            writeDigitalPin(a,LED,0);
            pause(0.2);
            writeDigitalPin(a,LED,1);
            pause(0.2);
        end
end

function rangeData = sensing(robotArduino, rawRangeIn)
    pause(0.01)
    rangeData = readVoltage(robotArduino, rawRangeIn);
end

function distanceData = calibrationLoop(robotArduino, rawRangeIn, blinkLED)
    numTests = input(['Enter number of tests']);
    clc;
    distanceData = zeros(numTests, 2);
    r = rateControl(0.25);
    reset(r);
    index = 1;
    while(index < numTests+1)
        rangeData = sensing(robotArduino, rawRangeIn)
        distanceData(index,1) = input('Enter distance(cm): ');
        gtest = input('Move to new distance then hit enter', 's');
        distanceData(index,2) = rangeData;
        Blink(robotArduino,blinkLED,1);
        waitfor(r);                     % wait for loop cycle to complete
        index = index+1;    % increment loop
    end
end

function plotCalibrationData(distanceData)
    % Plot and store command postion data vs. actual postionS
    plot(distanceData(:,1), distanceData(:,2), '*')
    xlabel('Actual position')
    ylabel('Measured Voltage')
    hold off
end

function [numTests, trialData] = positionOverTime(robotArduino, rawRangeIn, blinkLED)
    numTests = input(['Enter number of tests for trial']);
    trialData = zeros(numTests,1);
    index = 1;
    while(index < numTests + 1)
        rangeData = sensing(robotArduino, rawRangeIn);
        trialData(index) = rangeData;
        Blink(robotArduino,blinkLED,1);
        pause(.2)
        index = index + 1;
    end
    disp('Finished Collecting')
end


function [numTests, trialData] = positionOverAngle(robotArduino, rawRangeIn, panServo, tiltServ
    numTests = input(['Enter number of tests for trial']);
    trialData = zeros(numTests,3);
    angles = linspace(0,1,numTests);
    index = 1;
    for i = 1 : numTests
        writePosition(panServo, angles(i));
        pause(0.2);
        for j = 1 : numTests
            writePosition(tiltServo, angles(j));
            pause(0.2);
            %rangeData = sensing(robotArduino, rawRangeIn);
            rdat1 = (sensing(robotArduino, rawRangeIn));
            rdat2 = (sensing(robotArduino, rawRangeIn));
            rdat3 = (sensing(robotArduino, rawRangeIn));
            rangeData = mean([rdat1 rdat2 rdat3]);
            trialData(index,:,:) = [rangeData angles(i) angles(j)];
            Blink(robotArduino, blinkLED, 1);
            index = index + 1;
        end
    end
%     index = 1;
%     while(index < numTests + 1)
%         writePosition(panServo, (index*5)/360);
%         writePosition(tiltServo, (index*5)/360);
%         rangeData = sensing(robotArduino, rawRangeIn);
%         trialData(index) = rangeData;
%         Blink(robotArduino,blinkLED,1);
%         pause(.2)
%         index = index + 1;
%     end
    disp('Finished Collecting')
end

function commandAngle = movement_angle()
    commandAngle = input('Enter desired angle in degrees (0-180): ');
end
```