

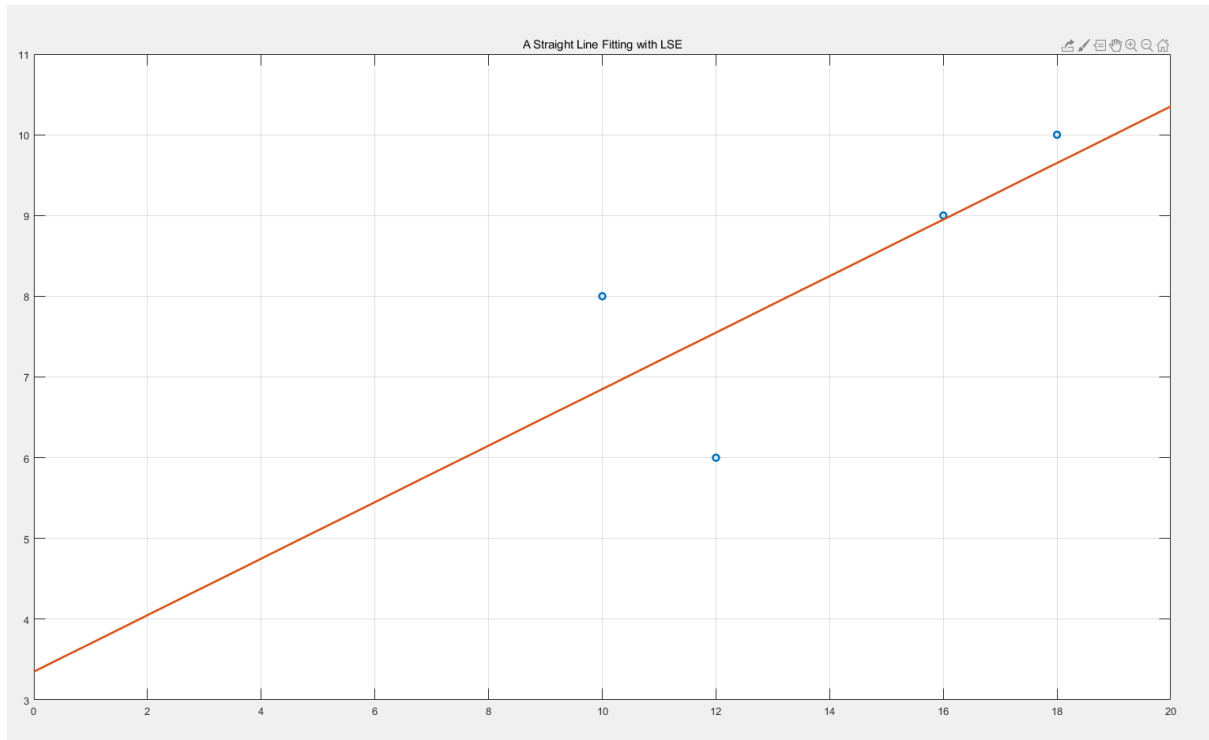
73.019: Advanced Photogrammetry

Lab #3: Least Squares Estimation for a Non-linear System

전일서

1. GMM을 이용하여 최소제곱법 함수 만들기

Lab3_Q1.m
<pre> % input % A; Design matrix % y; observations % Dy; Q cofactor matrix % N; square matrix % c; the right back side of least square equation ksihat=inv(A'PA)*(c=A'Py) % output % kt; ksihat % et; residuals % vct; variance component estimate % Dkt; cofactor estimate function [kt, et, vct, Dkt] = LSE (A, y, Dy); P = inv(Dy); N = A'*P*A; c = A'*P*y; iN = inv(N); kt = iN * c; et = y - A*kt; vct = et'*P*et / (length(y) - rank(A)); Dkt = vct * iN; </pre>
<p>LSE 함수를 생성한다.</p> <p>각 변수에 대한 의미는 주석에서 확인할 수 있다.</p>
<pre> x=[10,12,16,18]'; y=[8,6,9,10]'; A=[x, ones(4,1)]; Dy=eye(size(y,1)); %LSE [kt, et, vct, Dkt] = LSE (A, y, Dy); xt=linspace(0,20); yt=kt(1)*xt+kt(2); figure(); plot(x,y,'o',xt,yt,'LineWidth',2); grid on title('A Straight Line Fitting with LSE') </pre>
<p>X와 y의 실제 값을 임의로 입력하고, LSE함수를 이용하여 직선 피팅을 수행한다.</p> <p>결과는 다음과 같다.</p>



2. Horizontal network adjustment 수행하기

식 세우기

P는 실제 차량의 위치 (x,y)를 의미한다. 차량은 3개가 있다고 가정한다.

G는 차량의 GPS에서 취득된 (x,y)좌표를 의미한다. 차량이 3개이므로 3개의 식이 성립된다.

O는 한 차량에서 다른 차량을 바라본 좌표 값 차이를 의미한다.

T는 로드맵 위에 정지된 물체의 위치를 의미한다.

V는 각 관측치로부터 생길수 있는 오차를 의미한다.

$$\begin{aligned}
 G_1 + v_1 &= P_1 \\
 G_2 + v_2 &= P_2 \\
 G_3 + v_3 &= P_3 \\
 O_{12} + v_4 &= P_2 - P_1 \\
 O_{23} + v_5 &= P_3 - P_2 \\
 O_{31} + v_6 &= P_1 - P_3 \\
 T_1 + v_7 &= P_1 \\
 T_2 + v_8 &= P_2
 \end{aligned}$$

Lab3_Q2.m

```
%% real location of each car a, b, c
real_loc=[800,200;
          300,600;
          200,800];
no_data=3;
```

차량의 실제 위치와 수를 입력한다.

```
%% weight unit:m
G_Q = 20; % from GPS
```

```
O_Q = 10; % from relative distances
T_Q = 8; % from static objects(traffic light poles)
```

GPS, 상대적인 위치, 신호등에 의한 위치 정밀도를 입력한다.

```
%% observation data
gps_loc=real_loc+normrnd(0, G_Q); % 3x2

Oab=real_loc(2,:)-real_loc(1,:)+normrnd(0, O_Q); % 1x2
Obc=real_loc(3,:)-real_loc(2,:)+normrnd(0, O_Q); % 1x2
Oca=real_loc(1,:)-real_loc(3,:)+normrnd(0, O_Q); % 1x2

T(1,:)=real_loc(1,:)+normrnd(0, T_Q); % 1x2
T(2,:)=real_loc(2,:)+normrnd(0, T_Q); % 1x2
```

각 오차가 포함된 관측치들을 임의로 계산한다.

```
%% LSE
A = [1 0 0;
     0 1 0;
     0 0 1;
     -1 1 0;
     0 -1 1;
     1 0 -1;
     1 0 0;
     0 1 0];
y=[gps_loc;Oab;Obc;Oca;T(1,:);T(2,:)];
D=[G_Q, G_Q, G_Q, O_Q, O_Q, O_Q, T_Q, T_Q];
Dy=eye(size(A,1)).*D;

[kt_x, et_x, vct_x, Dkt_x] = LSE (A, y(:,1), Dy);
[kt_y, et_y, vct_y, Dkt_y] = LSE (A, y(:,2), Dy);
```

처음에 제시한 식에 따라 design matrix를 선언하고,
x좌표, y좌표를 따로 계산한다.

```
%% visualization
figure()
hold on
for i=1:no_data
    plot(real_loc(i,1),real_loc(i,2),'ro','MarkerSize', 10,
         'DisplayName', sprintf('real location of car %d',i));
    plot(gps_loc(i,1),gps_loc(i,2),'g*','MarkerSize', 10,'DisplayName',
         sprintf('gps location of car %d',i));
    plot(kt_x(i), kt_y(i),'b.','MarkerSize', 10,'DisplayName',
         sprintf('adjusted location of car %d',i));
end
daspect([1, 1, 1])
grid on
legend
title('Horizontal Network Adjustment with LSE')
```

다음의 순으로 가시화한다.

첫번째 plot은 실제 차량의 위치

두번째 plot은 gps로 취득된 위치

세번째 plot은 조정된 차량의 위치

