Relatório - miniprojeto 5

Laís Saloum Deghaide, nUSP: 11369767

1. Introdução

Neste miniprojeto, foi pedida a medição indireta de trajetórias de móveis, onde o movimento de uma partícula é detectado por 210 sensores que apenas podem medir posição ao longo das suas respectivas linhas de ação, que são direções unitárias conhecidas: $\widehat{d}_1, ..., \widehat{d}_s$.

Também são medidos os 210 sinais, $Y(1: 210)$ correspondentes aos afastamentos da partícula da origem nas direções 1 a 210, respectivamente. Isto é, cada medição $Y_i(t)$ satisfaz:

$$Y_i(t) = \widehat{d}_i \cdot X(t) + \xi_i(t),$$

sendo $\xi_i(t)$o erro Gaussiano independente, não correlacionado no tempo, de desvio padrão constante σ ($\sigma \simeq 0.1$).

2. Objetivo

São tomadas n = 1701 medições a intervalos unitários de tempo.

Nosso objetivo é recuperar, com a máxima precisão possível, a trajetória do móvel $X(t)$ (tridimensional) a partir de$\{ Y_1(t),..., Y_s(t) \}$. Para isso, devemos encontrar o plano no qual a trajetória tem a melhor precisão.

Temos, nas configurações estudadas, que os 210 sensores estão distribuídos numa semi-esfera de raio 50. O móvel sendo monitorado mexe na vizinhança da origem, perto do centro. Todos os sensores são iguais e não correlacionados, com erro de medição (Gaussiano) de 0.1cm aproximadamente.

3. Desenvolvimento

As direções unitárias medidas por cada sensor foram fornecidas no arquivo pp_2021_dire.txt, e as medições obtidas (valores de Y) em experimentos diferentes estão nos arquivos p67_2021_y.txt.

Desejamos um algoritmo que calcule o vetor normal a esse plano com o menor erro possível.

Parte A: Primeira estimação de X

Queremos relacionar a atuação de cada sensor no afastamento da partícula na origem. A utilização de sistemas sobredeterminados aqui será uma excelente ferramenta.

Os sistemas sobredeterminados nos ajudarão a entender as informações dadas e relacioná-las.

A relação entre as variáveis que queremos é dada pela fórmula:

$$Y_i(t) = \widehat{d_i} \cdot X(t) + \xi_i(t)$$

Essa relação nos dá apenas uma tendência de como nossos dados vão se comportar. Se tivéssemos um ajuste perfeito, existiria um vetor X tal que $A * x = b$.

Mas, como nosso problema é tridimensional, não temos um $X(t)$ que ajusta perfeitamente o sistema linear, ou seja, não temos um único plano que contém todos os pontos. Sendo assim, devemos encontrar o plano que melhor se ajusta aos pontos.

Teremos três planos calculados com o ajuste de $A * x = b$ e queremos encontrar X*. Porém, o problema é ajustar b como combinação linear das colunas de A. Um sistema sobredeterminado, em geral, não possui solução a não ser que o membro b seja um elemento da imagem de A. Então, para um b arbitrário podemos procurar um vetor X* que minimize a norma euclidiana do resíduo. Sendo esse vetor X* a solução dos mínimos quadrados do sistema sobredeterminado $A \cdot x = b$. Podemos resolver como $x = A \backslash b$. Isto nos fornece a solução de mínimos quadrados que tem norma mínima.

No nosso caso, temos a relação linear :

$$Y = G * X.$$

$b = Y\ (medições\ obtidas\ dos\ sensores)$ e $A = G\ (direção\ dos\ 210\ sensores)$

O método dos mínimos quadrados é uma técnica de otimização matemática que procura encontrar o melhor ajuste para um conjunto de dados tentando minimizar a soma dos quadrados das diferenças entre o valor estimado e os dados observados, sendo essas diferenças o que chamamos de resíduo.

Essa técnica pode ser implementada, no Octave, através da barra invertida "\", e com ela, conseguimos encontrar o melhor plano que minimiza a norma euclidiana usual do resíduo.

Através da fórmula da relação linear descrita acima, temos $Y(210 \times 1701)$ que representa a distância de cada sensor até a origem, $G(210 \times 3)$ que representa a direção de cada sensor em cada uma das direções x, y, z, e a matriz $X(3 \times 1701)$ que representa a posição do móvel em relação a x, y, z e que queremos encontrar.

Primeiramente, carreguei os arquivos txt que possuem as direções dos sensores e as medições obtidas, depois encontrei X.

```
% txt com as direções dos 210 sensores
G = load("pp_2021_dire.txt");

% txt com as mediçoes obtidas, contendo ruído
Y = load("p67_2021_y.txt");


% a) Primeira estimação de X

% temos a relação linear dada por: Y = G*X + ruído
% vetor 1071 x 3
X = G\Y';
X = X';
```

Em seguida, plotei as posições estimadas $X(:, t)$ tridimensionalmente, como uma nuvem de pontos e calculei os máximos e mínimos das três coordenadas.

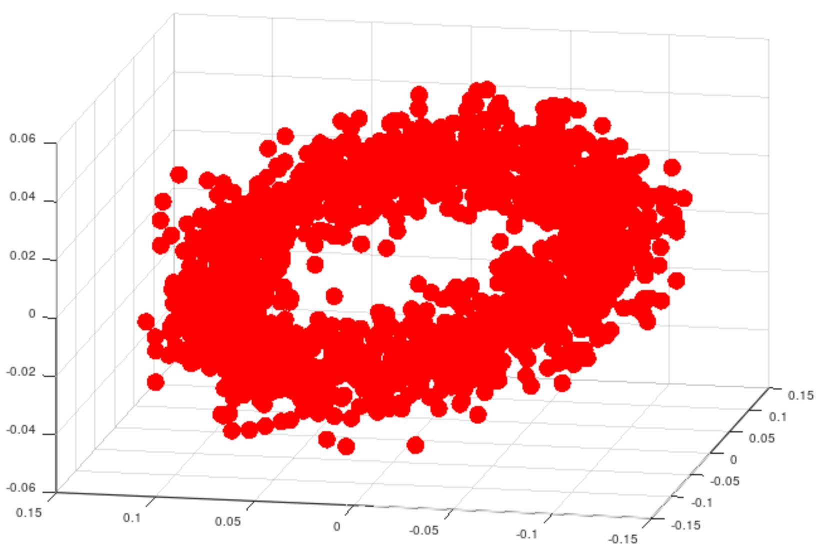
```
hold on
scatter3(X(:, 1), X(:, 2), X(:, 3), "r", "filled")
x = X(:, 1);
y = X(:, 2);
z = X(:, 3);

% cálculo dos máximos e mínimos quadrados
minx = min(x);
maxx = max(x);

miny = min(y);
maxy = max(y);

minz = min(z);
maxz = max(z);
```

Nuvem de pontos obtida:



Máximos e mínimos obtidos:

```
minx = -0.1305
maxx = 0.1388


miny = -0.1389
maxy = 0.1380


minz = -0.048829
maxz = 0.049700
```

Parte B: Ajuste do plano por mínimos quadrados

Tendo $X(t)$, conseguimos calcular o plano da trajetória utilizando ajuste de dados. Mais especificamente, realizar três ajustes de planos com os mesmos dados: $z = k_1x + k_2y$, $x = k_1y + k_2z$, $y = k_1x + k_2z$, e calcular a normal unitária de cada um deles.

Ou seja, as seguintes equações:

$[X\ Y] * k = Z$

$[Y\ Z] * k = X$

$[X\ Z] * k = Y$

Temos, $X,\ Y\ e\ Z$, devemos encontrar o vetor coluna k que são os nossos coeficientes.

Assim como na parte A, caímos em um sistema sobredeterminado e as relações acima não nos dão uma solução exata para o sistema.

Sendo assim, se não temos um k único que satisfaça o sistema linear, então não existe um único plano que contenha todos os pontos.

Então, devemos encontrar o plano que melhor se ajusta aos pontos e isso pode ser obtido através do cálculo dos vetores normais, dado por: $k = [X\ Y] \setminus Z$

$$k = [Y\ Z] \setminus X$$

$$k = [X\ Z] \setminus Y$$

Com isso, encontramos os seguintes vetores normais:

$$n_z = (k_1, k_2, -1)$$

$$n_x = (-1, k_1, k_2)$$

$$n_y = (k_1, -1, k_2)$$

Em seguida, calculamos os versores de cada um dos planos, dados por:

$$v_z = \frac{n_z}{\sqrt{K_1^2 + K_2^2 + (-1)^2}}$$

$$v_x = \frac{n_x}{\sqrt{(-1)^2 + K_1^2 + K_2^2}}$$

$$v_y = \frac{n_y}{\sqrt{K_1^2 + (-1)^2 + K_2^2}}$$

Código feito para calcular as fórmulas apresentadas acima:

```
% primeiro ajuste: z = k1*x + k2*y
A_z = [x y];
k_z = A_z\z;
n_z = [k_z(1), k_z(2), -1];
norma_z = n_z / sqrt(n_z(1)^2 + n_z(2)^2 + n_z(3)^2);
```

```
% segundo ajuste: x = k1*y + k2*z
A_x = [y z];
k_x = A_x\x;
n_x = [-1, k_x(1), k_x(2)];
norma_x = n_x / sqrt(n_x(1)^2 + n_x(2)^2 + n_x(3)^2);
```

```
% terceiro ajuste: y = k1*z + k2*x
A_y = [z x];
k_y = A_y\y;
n_y = [k_y(1), -1, k_y(2)];
norma_y = n_y / sqrt(n_y(1)^2 + n_y(2)^2 + n_y(3)^2);
```

Resultados das normas unitárias calculadas:

```
norma_z =

   0.200464   -0.092226   -0.975350


norma_x =

  -0.286309    0.090128    0.953889


norma_y =

  -0.9378   -0.2889    0.1926
```

Agora que temos as normas e os vetores de cada plano, conseguimos encontrar qual é o plano que melhor ajusta os pontos. Ou seja, conseguimos determinar qual possui a solução que minimiza a norma do vetor resíduo dado por: $r(t) = A \cdot k - b$.

Temos então os três vetores resíduos dados por cada um dos três planos:

$$r_z = A_z \cdot k - b$$

$$r_x = A_x \cdot k - b$$

$$r_y = A_y \cdot k - b$$

Tendo esses três vetores, podemos calcular o erro de cada um e encontrar o menor.

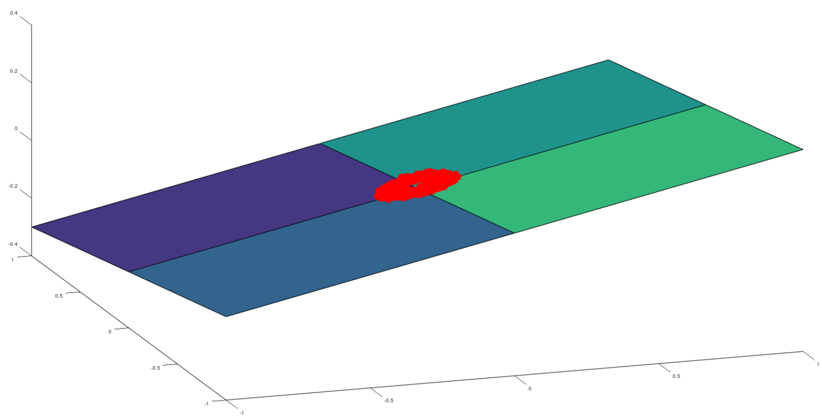
Código que realiza o passo a passo descrito acima:

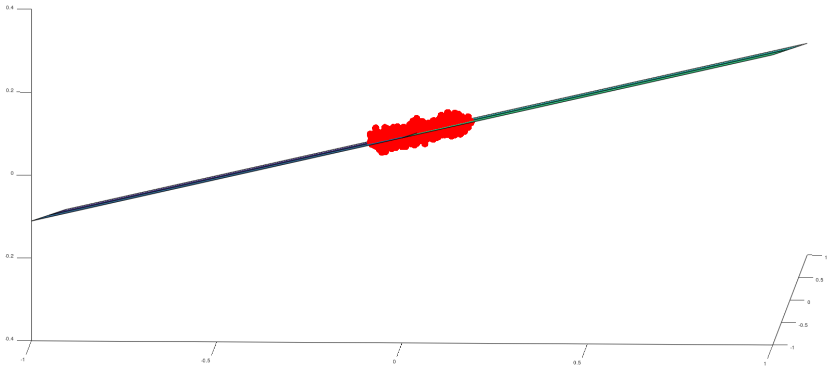
```
erro_resz = 0.4134
erro_resx = 1.6644
erro_resy = 2.4220
```

4. Resultados e Análise gráfica

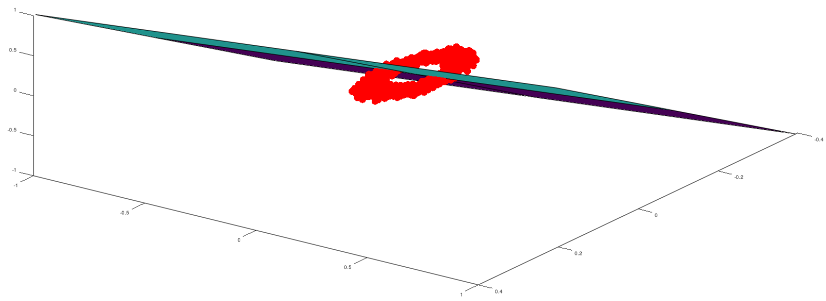
Podemos ver que o menor erro é o do plano z e, portanto, ele é o plano que melhor ajusta os pontos. Então, a estimativa de normal recomendada é a de z dada por: $n_z = (0.200464, -0.092226, -0.975350)$.

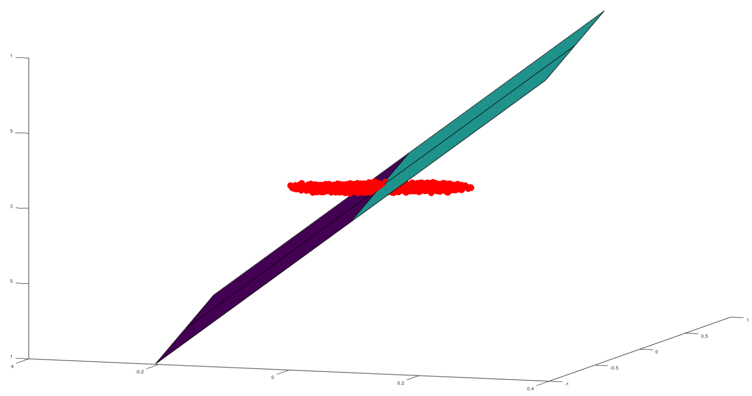
Analisando graficamente a nuvem de pontos para cada um dos planos:



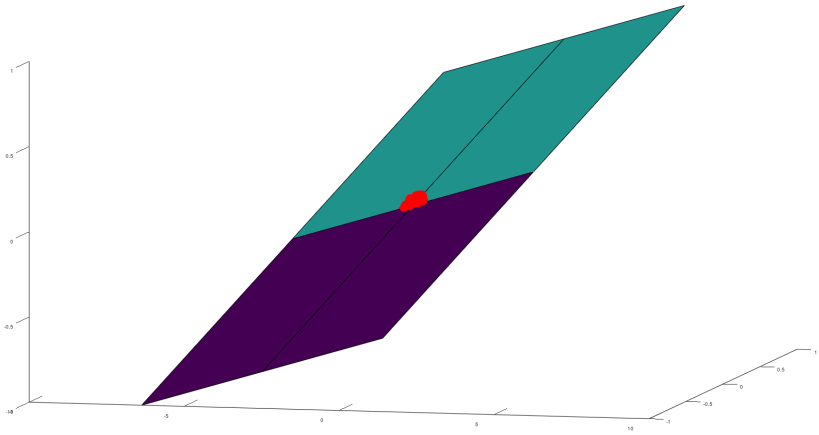


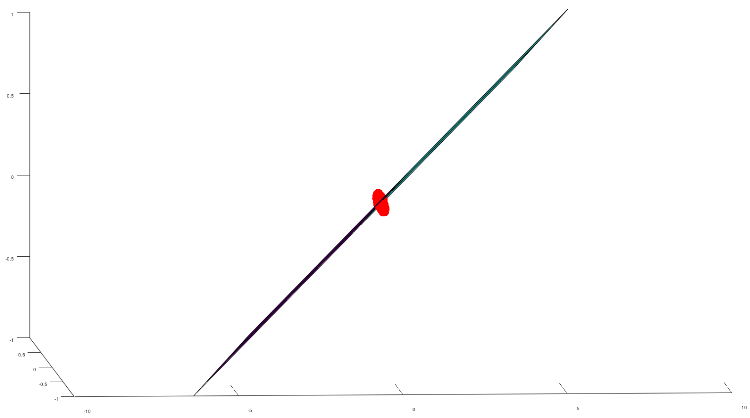
Plano ajustado z



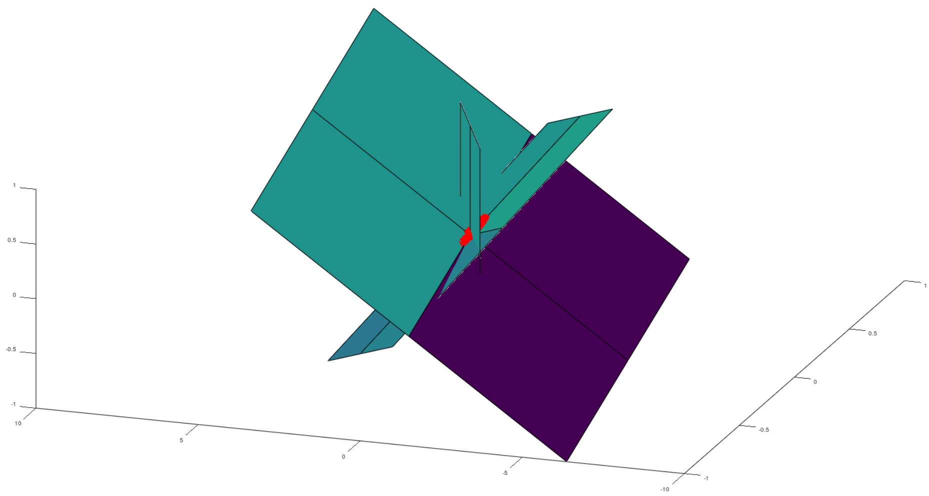


Plano ajustado x





Plano ajustado y



Os três planos juntos