

TEORIA DELLA STIMA SET MEMBERSHIP

**Esercitazione:
identificazione del modello statico
di un trasduttore di posizione**

Michele TARAGNA

*Dipartimento di Automatica e Informatica
Politecnico di Torino*

**Corso di III Livello
“Experimental modeling:
costruzione di modelli da dati sperimentali”**

Esercizio #1: identificazione del modello statico di un trasduttore di posizione mediante l'approccio statistico

1. Descrizione del sistema

Il trasduttore di posizione considerato fa uso di un sensore ottico costituito da 2 schiere di fotodiodi ed è utilizzato per valutare gli spostamenti verticali di un corpo metallico all'interno di un levitatore magnetico. È realizzato in configurazione differenziale, in modo tale da discriminare i movimenti verticali del corpo da quelli laterali.

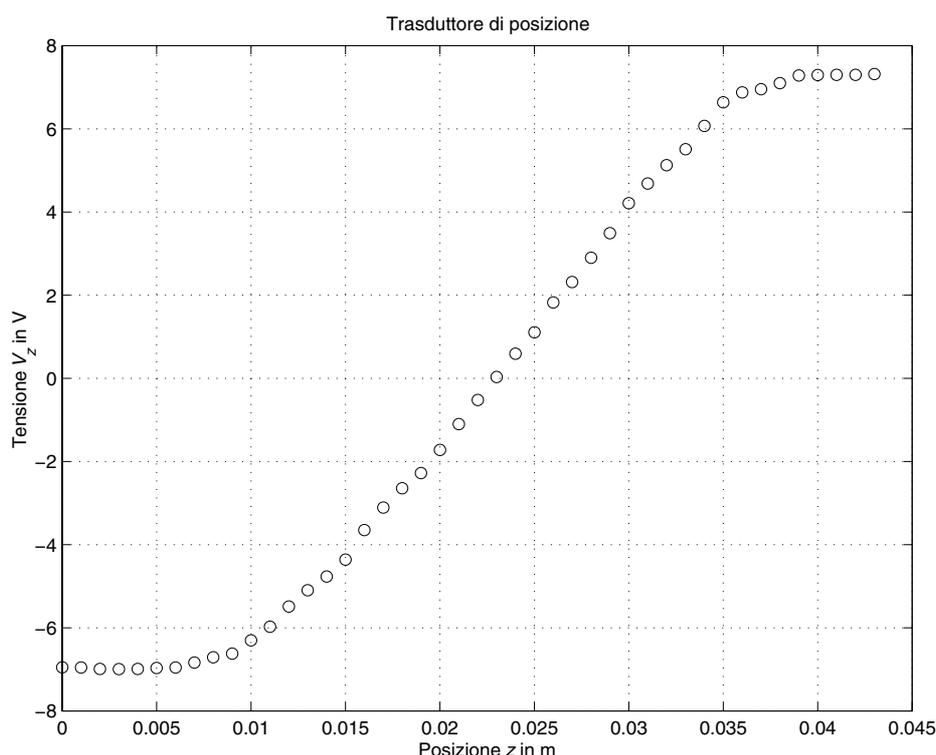
Tale trasduttore si può considerare statico in quanto la sua velocità di risposta ovvero le sue costanti di tempo risultano, da progetto, di alcuni ordini di grandezza superiori a quelle delle restanti parti del sistema.

La caratteristica statica del trasduttore è sicuramente non lineare man mano che ci si avvicina agli estremi del suo campo di azione. Occorre quindi rilevare l'uscita stazionaria (in tensione) del trasduttore in funzione della posizione (in metri) del corpo metallico.

2. Misura della caratteristica posizione-tensione

Per ottenere la caratteristica del trasduttore si può agire come segue:

1. si pone pari a zero il comando dell'attuatore;
2. si porta il corpo nella massima posizione superiore e quindi si agisce sulla ghiera di posizionamento (1mm/giro) per abbassarlo;
3. per ogni posizione, si riporta in una tabella la tensione V_z di uscita del trasduttore (misurata in volt) e la posizione z (misurata in metri) letta sulla scala graduata a fianco della vite di posizionamento.



3. Approssimazione lineare della caratteristica posizione-tensione

Osservando la caratteristica posizione-tensione del trasduttore, si può notare un comportamento quasi lineare nell'intervallo approssimativamente compreso fra 1 e 3.5 cm. Nell'intervallo di linearità si può quindi assumere una legge del tipo

$$V_z = K_t \cdot z + V_o$$

in cui il guadagno K_t e la tensione di offset V_o sono costanti da determinarsi. Poiché l'errore più rilevante è quello commesso posizionando la pallina mediante la ghiera di

posizionamento (l'errore massimo di posizione è di circa 0.5 mm), è opportuno riscrivere l'equazione del modello nella forma seguente, in cui si tiene esplicitamente conto anche dell'errore di misura e :

$$z = \frac{1}{K_t} \cdot V_z - \frac{V_o}{K_t} + e$$

Se a questo punto si tiene conto delle N misure effettuate in regime di linearità, si ricava il sistema di equazioni:

$$\begin{aligned} z_1 &= \frac{1}{K_t} \cdot V_{z,1} - \frac{V_o}{K_t} + e_1 \\ z_2 &= \frac{1}{K_t} \cdot V_{z,2} - \frac{V_o}{K_t} + e_2 \\ &\vdots \\ z_N &= \frac{1}{K_t} \cdot V_{z,N} - \frac{V_o}{K_t} + e_N \end{aligned}$$

dove $V_{z,i}$ è la tensione fornita dal trasduttore in corrispondenza della posizione z_i .

Introducendo i parametri incogniti

$$p_1 = \frac{1}{K_t}, \quad p_2 = -\frac{V_o}{K_t}$$

si possono così riscrivere le relazioni precedenti:

$$\begin{aligned} z_1 &= V_{z,1} \cdot p_1 + p_2 + e_1 \\ z_2 &= V_{z,2} \cdot p_1 + p_2 + e_2 \\ &\vdots \\ z_N &= V_{z,N} \cdot p_1 + p_2 + e_N \end{aligned}$$

oppure, in forma matriciale:

$$\begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_N \end{bmatrix} = \begin{bmatrix} V_{z,1} & 1 \\ V_{z,2} & 1 \\ \vdots & \vdots \\ V_{z,N} & 1 \end{bmatrix} \cdot \begin{bmatrix} p_1 \\ p_2 \end{bmatrix} + \begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_N \end{bmatrix}$$

cioè nella forma

$$z^N = L \cdot p + e^N$$

dove $z^N \in \mathfrak{R}^N$, $L \in \mathfrak{R}^{N \times 2}$, $p \in \mathfrak{R}^2$ ed $e^N \in \mathfrak{R}^N$. Poiché l'incognita è il vettore p , il problema che si vuole risolvere è sovradeterminato, in quanto il numero delle incognite è inferiore al numero delle equazioni di misura utilizzate e pertanto in generale il sistema di equazioni non ammette soluzioni (non è possibile infatti invertire la matrice L).

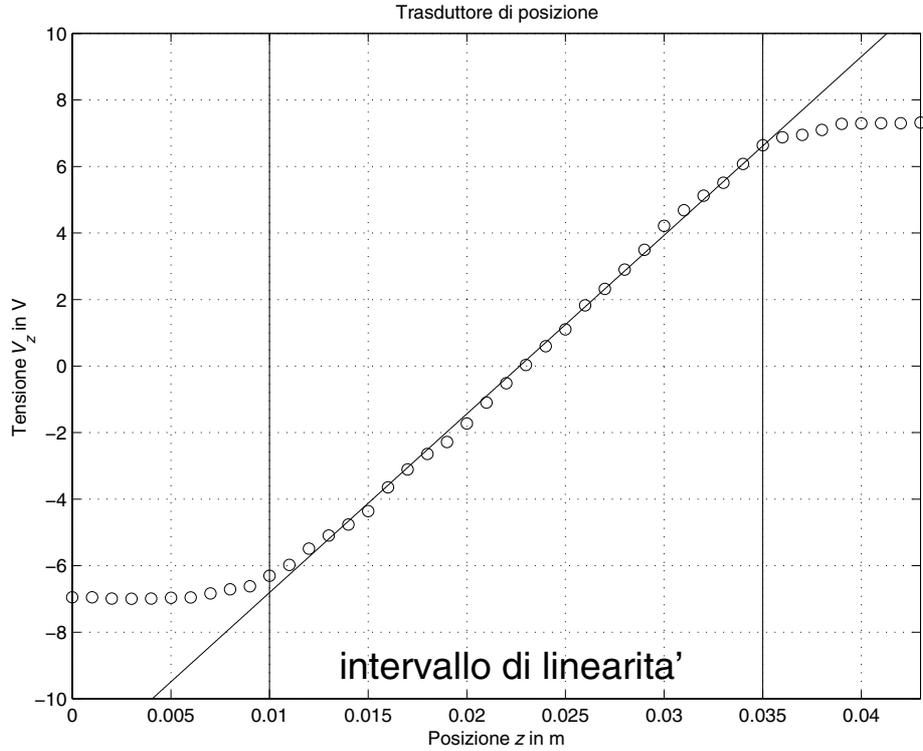
Utilizzando come metodo di stima l'algoritmo dei minimi quadrati:

$$\hat{p} = (L^T \cdot L)^{-1} L^T \cdot z^N$$

si ottiene:

$$\hat{p} = \begin{bmatrix} \hat{p}_1 \\ \hat{p}_2 \end{bmatrix} = \begin{bmatrix} 0.0019 \\ 0.0227 \end{bmatrix} \Rightarrow \hat{K}_t = \frac{1}{\hat{p}_1} = 537.0036 \text{V/m}, \quad \hat{V}_o = -\frac{\hat{p}_2}{\hat{p}_1} = -12.1779 \text{V}$$

In MATLAB, la soluzione ai minimi quadrati si può ricavare mediante l'operatore “\” oppure l'istruzione `mldivide`. Si veda a tal proposito l'help-on-line mediante `help mldivide`.



4. Determinazione degli intervalli di confidenza dei parametri stimati

a) Caso di rumore i.i.d. gaussiano di varianza nota o ricavabile da informazioni a priori.

Si è detto che la principale fonte di errore nelle misure sperimentali è costituita dalla misura della posizione z e che l'errore massimo è di circa 0.5 mm. Ipotizzando che tale errore abbia una distribuzione normale $\mathcal{N}(0, \Sigma_e)$, la matrice di covarianza Σ_e può essere assunta pari a $\Sigma_e = \sigma_e^2 \cdot I_{N \times N}$, dove la varianza σ_e^2 si ricava in base alle seguenti considerazioni:

- la probabilità che il valore osservato di una variabile casuale ξ avente d.d.p. normale differisca dal valor medio $E[\xi]$ al più di k volte la deviazione standard σ_ξ è pari a

$$P = P(|\xi - E[\xi]| \leq k \cdot \sigma_\xi) = 1 - \frac{2}{\sqrt{2\pi}} \int_k^\infty e^{-t^2/2} dt$$

ed in particolare risulta che:

k	P
1	68.3%
2	95.4%
3	99.7%

- assumendo che il range di variazione dell'errore di misura $[-5 \cdot 10^{-4} \text{m}, 5 \cdot 10^{-4} \text{m}]$ corrisponda ad un intervallo di confidenza sufficientemente elevato, ad esempio quello al 95%, si ricava che:

$$5 \cdot 10^{-4} = 2 \cdot \sigma_e \quad \Rightarrow \quad \sigma_e = \frac{5 \cdot 10^{-4}}{2} = 2.5 \cdot 10^{-4}$$

La matrice di covarianza della stima di Gauss-Markov \hat{p} è data:

$$\Sigma_{\hat{p}} = (L^T \cdot \Sigma_e^{-1} \cdot L)^{-1} = \sigma_e^2 \cdot (L^T \cdot L)^{-1} = \begin{bmatrix} \sigma_{\hat{p},11}^2 & \sigma_{\hat{p},12}^2 \\ \sigma_{\hat{p},12}^2 & \sigma_{\hat{p},22}^2 \end{bmatrix} = \begin{bmatrix} 0.0149 & 0.0014 \\ 0.0014 & 0.2405 \end{bmatrix} \cdot 10^{-8}$$

e quindi le deviazioni standard delle densità di probabilità marginali dei parametri sono:

$$\begin{aligned} \sigma_{\hat{p}_1} &= \sqrt{\sigma_{\hat{p},11}^2} = 1.2187 \cdot 10^{-5} \\ \sigma_{\hat{p}_2} &= \sqrt{\sigma_{\hat{p},22}^2} = 4.9043 \cdot 10^{-5} \end{aligned}$$

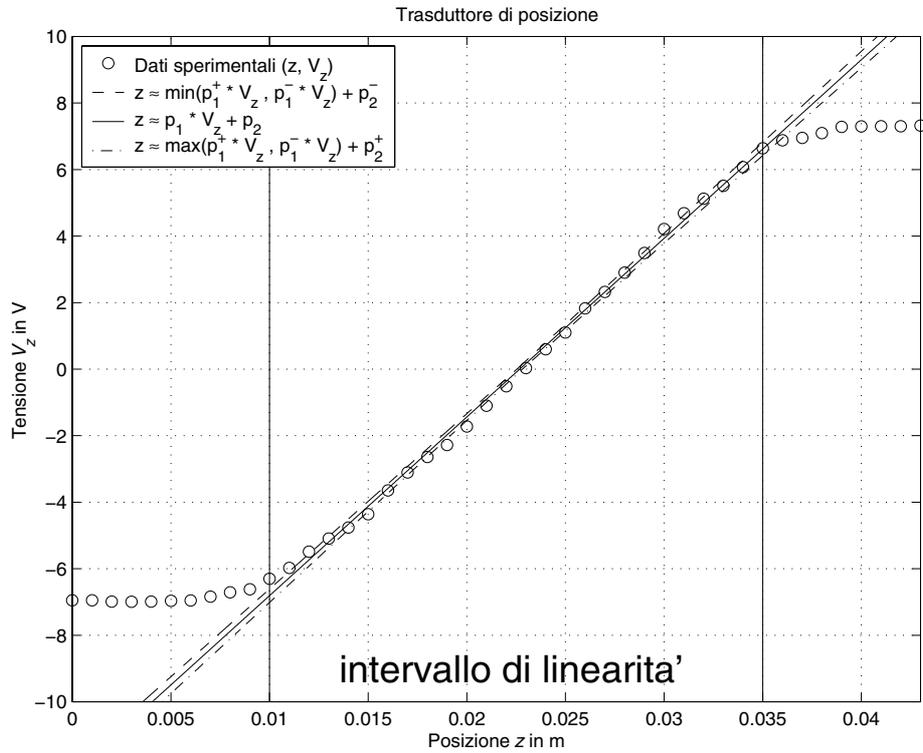
Gli intervalli di confidenza dei parametri \hat{p}_i , calcolati ad esempio al 95%, sono dati da:

$$\begin{aligned} [\hat{p}_1^-, \hat{p}_1^+] &= [\hat{p}_1 - \delta_{\hat{p}_1}, \hat{p}_1 + \delta_{\hat{p}_1}] = [0.0018, 0.0019], \text{ con } \delta_{\hat{p}_1} = 2 \cdot \sigma_{\hat{p}_1} = 2.4374 \cdot 10^{-5} \\ [\hat{p}_2^-, \hat{p}_2^+] &= [\hat{p}_2 - \delta_{\hat{p}_2}, \hat{p}_2 + \delta_{\hat{p}_2}] = [0.0226, 0.0228], \text{ con } \delta_{\hat{p}_2} = 2 \cdot \sigma_{\hat{p}_2} = 9.8086 \cdot 10^{-5} \end{aligned}$$

e quindi gli intervalli di confidenza al 95% di \hat{K}_t e \hat{V}_o sono pari a:

$$\begin{aligned} [\hat{K}_t^-, \hat{K}_t^+] &= \left[\frac{1}{\hat{p}_1^+}, \frac{1}{\hat{p}_1^-} \right] = [530.1, 544.1] \text{ V/m} \\ [\hat{V}_o^-, \hat{V}_o^+] &= \left[-\frac{\hat{p}_2^+}{\hat{p}_1^-}, -\frac{\hat{p}_2^-}{\hat{p}_1^+} \right] = [-12.39, -11.97] \text{ V} \end{aligned}$$

Nella figura sottostante è riportato l'involuppo dell'insieme di modelli i cui parametri \hat{p} appartengono agli intervalli di confidenza al 95%.



b) Caso di rumore i.i.d. gaussiano di varianza non nota.

In tal caso, assumendo che $\Sigma_e = \hat{\sigma}_e^2 \cdot I_{N \times N}$, la stima di massima verosimiglianza della deviazione standard del rumore è data da:

$$\hat{\sigma}_e = \sqrt{\frac{1}{N} (z^N - L \cdot p)^T \cdot (z^N - L \cdot p)} = 3.5618 \cdot 10^{-4}$$

Procedendo come nel caso precedente, si ricavano gli intervalli di confidenza di \hat{K}_t e \hat{V}_o che, se calcolati al 95%, sono pari a:

$$\begin{aligned} [\hat{K}_t^-, \hat{K}_t^+] &= [527.2, 547.2] \text{ V/m} \\ [\hat{V}_o^-, \hat{V}_o^+] &= [-12.49, -11.88] \text{ V} \end{aligned}$$

5. Approssimazioni polinomiali della caratteristica posizione-tensione

Si può cercare di approssimare la caratteristica posizione-tensione mediante polinomi di ordine dispari più elevato, per catturare il fenomeno di saturazione presente agli estremi.

Ad esempio, si può utilizzare il seguente modello del III ordine:

$$z = p_1 \cdot V_z^3 + p_2 \cdot V_z^2 + p_3 \cdot V_z + p_4 + e$$

che, tenendo conto di tutte le misure effettuate, porta a considerare le seguenti equazioni:

$$\begin{aligned} z_1 &= p_1 \cdot V_{z,1}^3 + p_2 \cdot V_{z,1}^2 + p_3 \cdot V_{z,1} + p_4 + e_1 \\ z_2 &= p_1 \cdot V_{z,2}^3 + p_2 \cdot V_{z,2}^2 + p_3 \cdot V_{z,2} + p_4 + e_2 \\ &\vdots \\ z_N &= p_1 \cdot V_{z,N}^3 + p_2 \cdot V_{z,N}^2 + p_3 \cdot V_{z,N} + p_4 + e_N \end{aligned}$$

oppure in forma matriciale:

$$\begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_N \end{bmatrix} = \begin{bmatrix} V_{z,1}^3 & V_{z,1}^2 & V_{z,1} & 1 \\ V_{z,2}^3 & V_{z,2}^2 & V_{z,2} & 1 \\ \vdots & \vdots & \vdots & \vdots \\ V_{z,N}^3 & V_{z,N}^2 & V_{z,N} & 1 \end{bmatrix} \cdot \begin{bmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \end{bmatrix} + \begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_N \end{bmatrix}$$

cioè nella forma

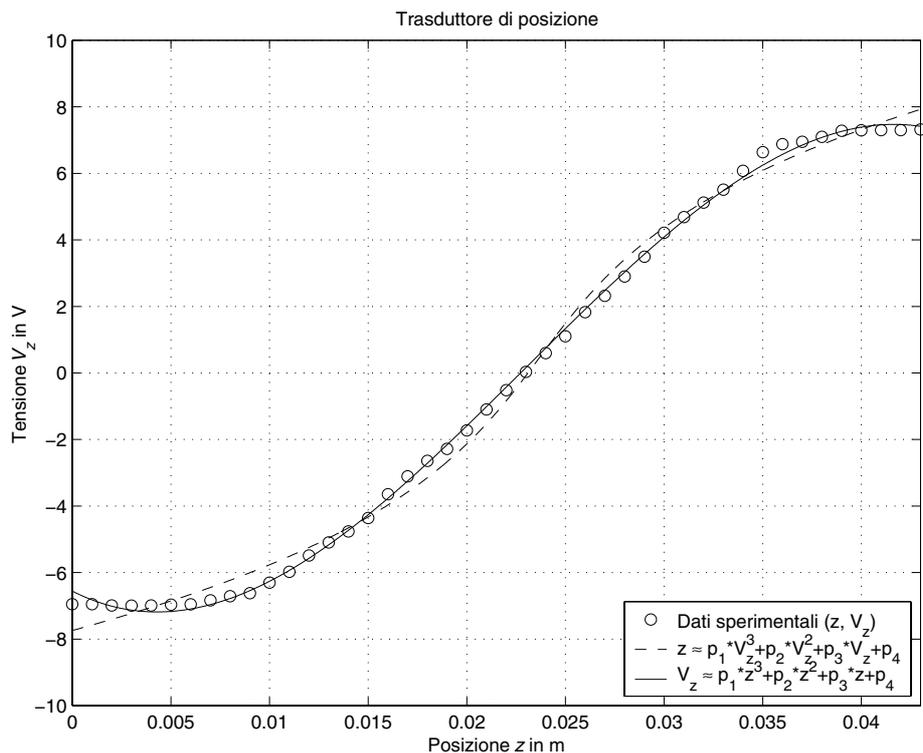
$$z^N = L \cdot p + e^N$$

dove $z^N \in \mathbb{R}^N$, $L \in \mathbb{R}^{N \times 4}$, $p \in \mathbb{R}^4$ ed $e^N \in \mathbb{R}^N$.

Utilizzando come metodo di stima l'algoritmo dei minimi quadrati, si ottiene:

$$\hat{p} = (L^T \cdot L)^{-1} \cdot L^T \cdot z^N = \begin{bmatrix} 2.4097 \cdot 10^{-5} \\ -3.4590 \cdot 10^{-5} \\ 0.0013 \\ 0.0231 \end{bmatrix}$$

In MATLAB, tale soluzione ai minimi quadrati si può ricavare anche mediante l'istruzione `polyfit`. Si veda a tal proposito l'help-on-line mediante `help polyfit`.



6. Principali comandi MATLAB da prendere in considerazione

- mldivide
- polyfit
- polyval

Possibile soluzione mediante MATLAB (file sensore.m)

```
% Identificazione del modello di un sensore di posizione

clear all, close all, pack, clc

% posizione misurata in metri, con offset di posizione
z_misurata=(0.037:0.001:0.08)';

% posizione misurata in metri, senza offset di posizione
z=z_misurata-z_misurata(1);

% tensione di uscita misurata in volt
Vz=[-6.9522; -6.9530; -6.9873; -6.9922; -6.9868; -6.9649; -6.9563; -6.8359; ...
    -6.7097; -6.6211; -6.3014; -5.9746; -5.4875; -5.0964; -4.7653; -4.3603; -3.6474; ...
    -3.1056; -2.6439; -2.2799; -1.7218; -1.0987; -0.5180; 0.0326; 0.5958; 1.1052; ...
    1.8264; 2.3193; 2.9003; 3.4942; 4.2122; 4.6871; 5.1238; 5.5129; 6.0731; ...
    6.6401; 6.8797; 6.9531; 7.0979; 7.2823; 7.2941; 7.2998; 7.3008; 7.3191];

save sensore z Vz

plot(z,Vz,'o'), grid, title('Trasduttore di posizione'),
xlabel('Posizione{\it z} in m'), ylabel('Tensione{\it V_z} in V'),
print -deps sensore1.eps

%-----
% Stima dei parametri di un modello lineare con i minimi quadrati
%-----

% indici corrispondenti all'intervallo di linearita' della caratteristica
i1=11;
i2=36;

% identificazione dei parametri mediante i minimi quadrati;
L=[Vz(i1:i2), ones(i2-i1+1,1)];
p=L\z(i1:i2); % con l'operatore "\"
Kt=1/p(1)
Vo=-p(2)/p(1)

p_ =inv(L'*L)*L'*z(i1:i2); % calcolando la pseudoinversa
Kt_ =1/p_(1)
Vo_=-p_(2)/p_(1)

% calcolo degli intervalli di confidenza all'x %
% x=95 => k=2 ("2 sigma"); x=99.7 => k=3 ("3 sigma")
k_e=2; k_p=2;
rumore_max=5e-4;

% caso #1: varianza nota
sigma_e=rumore_max/k_e
Sigma_e=sigma_e^2*eye(i2-i1+1);
Sigma_p=inv(L'*inv(Sigma_e)*L);
sigma_p=sqrt(diag(Sigma_p));
delta_p=k_p*sigma_p;
p_min=p-delta_p;
p_max=p+delta_p;
Kt_min=1/p_max(1)
Kt_max=1/p_min(1)
Vo_min=-p_max(2)/p_min(1)
Vo_max=-p_min(2)/p_max(1)
```

```

% caso #2: varianza da stimare
sigma_e_mv=sqrt((z(i1:i2)-L*p)'*(z(i1:i2)-L*p)/(i2-i1+1))
Sigma_e=sigma_e_mv^2*eye(i2-i1+1);
Sigma_p=inv(L'*inv(Sigma_e)*L);
sigma_p=sqrt(diag(Sigma_p));
delta_p=k_p*sigma_p;
p_min=p-delta_p;
p_max=p+delta_p;
Kt_min=1/p_max(1)
Kt_max=1/p_min(1)
Vo_min=-p_max(2)/p_min(1)
Vo_max=-p_min(2)/p_max(1)

% Confronto dei risultati ottenuti

Vz0=linspace(-10,10,1000);
z_hat=p(1)*Vz0+p(2);
z_min=min([p_max(1)*Vz0; p_min(1)*Vz0])+p_min(2);
z_max=max([p_max(1)*Vz0; p_min(1)*Vz0])+p_max(2);

figure, plot(z,Vz,'o',z(i1)*[1,1],[-10,10],'-',z(i2)*[1,1],[-10,10],'-',z_hat,Vz0),
axis([min(z),max(z),-10,10]), grid, title('Trasduttore di posizione'),
xlabel('Posizione{\it z} in m'), ylabel('Tensione{\it V_z} in V'),
text(0.013,-9,'\fontsize{20} intervallo di linearita'''),
print -deps sensore2.eps

figure, plot(z,Vz,'o', z_min,Vz0,'--', z_hat,Vz0,'-', z_max,Vz0,'-.', ...
z(i1)*[1,1],[-10,10],'-', z(i2)*[1,1],[-10,10],'-'),
axis([min(z),max(z),-10,10]), grid, title('Trasduttore di posizione'),
xlabel('Posizione{\it z} in m'), ylabel('Tensione{\it V_z} in V'),
legend('Dati sperimentali (z, V_z)_{ }{ }', ...
'z \approx min(p_{1}^{+} * V_{z} , p_{1}^{-} * V_{z}) + p_{2}^{-}', ...
'z \approx p_{1} * V_{z} + p_{2}', ...
'z \approx max(p_{1}^{+} * V_{z} , p_{1}^{-} * V_{z}) + p_{2}^{+}', 2),
text(0.013,-9,'\fontsize{20} intervallo di linearita'''),
print -deps sensore3.eps

%-----
% Modello polinomiale di ordine 3 con i minimi quadrati
%-----

% a)  $z = p(1)*Vz^3 + p(2)*Vz^2 + p(3)*Vz + p(4)$ 
L=[Vz.^3, Vz.^2, Vz, Vz.^0];
p3a=L\z
z_pol=polyval(p3a,Vz0);

% b)  $Vz = p(1)*z^3 + p(2)*z^2 + p(3)*z + p(4)$ 
L=[z.^3, z.^2, z, z.^0];
p3b=L\Vz
z0=linspace(min(z),max(z),1000);
Vz_pol=polyval(p3b,z0);

figure, plot(z,Vz,'o',z_pol,Vz0,'--',z0,Vz_pol,'-'),
axis([min(z),max(z),-10,10]), grid, title('Trasduttore di posizione'),
xlabel('Posizione{\it z} in m'), ylabel('Tensione{\it V_z} in V'),
legend('Dati sperimentali (z, V_z)_{ }{ }', ...
'z \approx p_1*V_z^3+p_2*V_z^2+p_3*V_z+p_4', ...
'V_z \approx p_1*z^3+p_2*z^2+p_3*z+p_4',4),
print -deps sensore4.eps

%-----
% Modello polinomiale di ordine 3 calcolato con polyfit
%-----

modello_ordine3=polyfit(z,Vz,3);
Vz_modello_ordine3=polyval(modello_ordine3,z0);

```

Esercizio #2: identificazione del modello statico di un trasduttore di posizione mediante l'approccio set membership

1. Approssimazione lineare della caratteristica posizione-tensione

Osservando la caratteristica posizione-tensione del trasduttore di cui all'esercizio #1, si può notare un comportamento quasi lineare nell'intervallo approssimativamente compreso fra 1.3 e 3.5 cm. In tale intervallo di linearità si può quindi assumere una legge del tipo

$$V_z = K_t \cdot z + V_o$$

in cui il guadagno K_t e la tensione di offset V_o sono costanti da determinarsi. Poiché l'errore più rilevante è quello commesso posizionando la pallina mediante l'apposita ghiera (l'errore massimo compiuto nella lettura della posizione z è di circa 0.5 mm), è opportuno riscrivere l'equazione del modello nella forma seguente, in cui si tiene esplicitamente conto anche dell'errore di misura e :

$$z = \frac{1}{K_t} \cdot V_z - \frac{V_o}{K_t} + e$$

Introducendo i parametri incogniti

$$\nu_1 = \frac{1}{K_t}, \quad \nu_2 = -\frac{V_o}{K_t}$$

le N misure effettuate in regime di linearità danno luogo al sistema di equazioni:

$$\begin{aligned} z_1 &= V_{z,1} \cdot \nu_1 + \nu_2 + e_1 \\ z_2 &= V_{z,2} \cdot \nu_1 + \nu_2 + e_2 \\ &\vdots \\ z_N &= V_{z,N} \cdot \nu_1 + \nu_2 + e_N \end{aligned}$$

dove $V_{z,i}$ è la tensione fornita dal trasduttore in corrispondenza della posizione z_i . Le relazioni precedenti possono essere riscritte nella seguente forma matriciale:

$$\begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_N \end{bmatrix} = \begin{bmatrix} V_{z,1} & 1 \\ V_{z,2} & 1 \\ \vdots & \vdots \\ V_{z,N} & 1 \end{bmatrix} \cdot \begin{bmatrix} \nu_1 \\ \nu_2 \end{bmatrix} + \begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_N \end{bmatrix}$$

cioè nella forma:

$$z^N = L \cdot \nu + e^N$$

dove $z^N \in \mathfrak{R}^N$, $L \in \mathfrak{R}^{N \times 2}$, $e^N \in \mathfrak{R}^N$ e l'incognita è il vettore $\nu \in \mathfrak{R}^2$.

Utilizzando come metodo di stima l'algoritmo dei minimi quadrati:

$$\hat{\nu} = A \cdot z^N, \quad \text{con } A = (L^T \cdot L)^{-1} L^T$$

si ottiene:

$$\hat{\nu} = \begin{bmatrix} \hat{\nu}_1 \\ \hat{\nu}_2 \end{bmatrix} = \begin{bmatrix} 1.8194 \cdot 10^{-3} \\ 2.2791 \cdot 10^{-2} \end{bmatrix} \Rightarrow \hat{K}_t = \frac{1}{\hat{\nu}_1} = 549.62 \text{ V/m}, \quad \hat{V}_o = -\frac{\hat{\nu}_2}{\hat{\nu}_1} = -12.526 \text{ V}$$

2. Determinazione degli intervalli di incertezza delle stime EUI_∞

Si è detto che la principale fonte di errore nelle misure sperimentali è costituita dalla misura della posizione z e che l'errore massimo è di circa 0.5 mm. Pertanto l'errore di misura e^N può essere considerato non noto ma limitato ed in particolare

$$e^N \in \mathcal{E}_\infty = \left\{ \tilde{e}^N \in \mathfrak{R}^N : |\tilde{e}_k| \leq \varepsilon, k = 1, \dots, N \right\}, \quad \text{con } \varepsilon = 5 \cdot 10^{-4}$$

Si osservi che \mathcal{E}_∞ è un cubo di lato $2 \cdot \varepsilon$ centrato nell'origine.

Gli intervalli di incertezza delle stime EUI_∞ sono definiti come:

$$EUI_{\infty,j} = \left[\hat{\nu}_j^m = \min_{\nu \in EUS_\infty} \nu_j, \hat{\nu}_j^M = \max_{\nu \in EUS_\infty} \nu_j \right], \quad j = 1, 2$$

dove l'insieme delle stime possibili EUS_∞ è l'immagine mediante A dell'insieme delle misure possibili MUS_∞ , costituito da un cubo di lato $2 \cdot \varepsilon$ centrato in z^N :

$$EUS_\infty = A \cdot MUS_\infty = A \cdot (z^N \oplus \mathcal{E}_\infty) = A \cdot \{ \tilde{z}^N \in \mathfrak{R}^N : |\tilde{z}_k - z_k| \leq \varepsilon, k = 1, \dots, N \}$$

Poiché si dimostra che:

$$\hat{\nu}_j^m = \min_{\nu \in EUS_\infty} \nu_j = \sum_{k=1}^N A_{jk} \cdot [z_k - \varepsilon \cdot \text{sign}(A_{jk})]$$

$$\hat{\nu}_j^M = \max_{\nu \in EUS_\infty} \nu_j = \sum_{k=1}^N A_{jk} \cdot [z_k + \varepsilon \cdot \text{sign}(A_{jk})] = 2 \cdot \hat{\nu}_j - \hat{\nu}_j^m$$

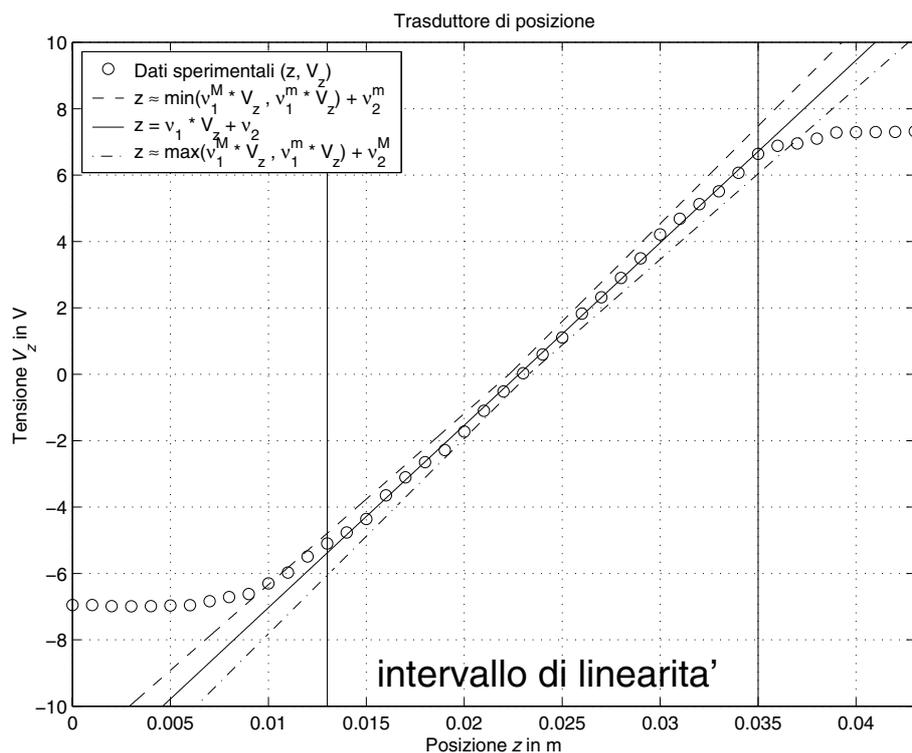
nel caso in esame risulta che:

$$\begin{aligned} [\hat{\nu}_1^m, \hat{\nu}_1^M] &= [1.6996 \cdot 10^{-3}, 1.9392 \cdot 10^{-3}] \\ [\hat{\nu}_2^m, \hat{\nu}_2^M] &= [2.2291 \cdot 10^{-2}, 2.3291 \cdot 10^{-2}] \end{aligned}$$

e quindi gli intervalli di incertezza delle stime \hat{K}_t e \hat{V}_o sono pari a:

$$\begin{aligned} [\hat{K}_t^m, \hat{K}_t^M] &= \left[\frac{1}{\hat{\nu}_1^M}, \frac{1}{\hat{\nu}_1^m} \right] = [515.67, 588.36] \text{ V/m} \\ [\hat{V}_o^m, \hat{V}_o^M] &= \left[-\frac{\hat{\nu}_2^M}{\hat{\nu}_1^m}, -\frac{\hat{\nu}_2^m}{\hat{\nu}_1^M} \right] = [-13.703, -11.495] \text{ V} \end{aligned}$$

Nella figura sottostante è riportato l'involuppo dell'insieme di modelli i cui parametri $\hat{\nu}$ appartengono agli estremi degli intervalli di incertezza $EUI_{\infty,j}$ con $j = 1, 2$.



3. Determinazione degli intervalli di incertezza dei parametri compatibili PUI_∞

Gli intervalli di incertezza dei parametri compatibili PUI_∞ sono definiti come:

$$PUI_{\infty,j} = \left[\min_{\nu \in FPS_\infty} \nu_j, \max_{\nu \in FPS_\infty} \nu_j \right] \subseteq EUI_{\infty,j}, \quad j = 1, 2$$

dove l'insieme dei parametri compatibili FPS_∞ è un politopo (cioè un poliedro convesso) generato da disuguaglianze lineari:

$$FPS_\infty = \left\{ \tilde{\nu} \in \mathfrak{R}^{dim(\tilde{\nu})} : |z_k - [L \cdot \tilde{\nu}]_k| \leq \varepsilon, \quad k = 1, \dots, N \right\}$$

Gli estremi degli intervalli $PUI_{\infty,j}$ si ricavano come soluzione di problemi di programmazione lineare. In MATLAB, si può utilizzare l'istruzione `lp` nella forma `lp(c,M,b)` che permette di risolvere il problema

$$\min_x c^T \cdot x \quad \text{col vincolo } M \cdot x \leq b$$

Si tratta quindi di riscrivere opportunamente le disuguaglianze che definiscono l' FPS_∞ :

$$|z_k - [L \cdot \tilde{\nu}]_k| \leq \varepsilon \Leftrightarrow -\varepsilon \leq z_k - [L \cdot \tilde{\nu}]_k \leq \varepsilon \Leftrightarrow \begin{cases} [L \cdot \tilde{\nu}]_k \leq z_k + \varepsilon \\ -[L \cdot \tilde{\nu}]_k \leq -z_k + \varepsilon \end{cases}, \quad k = 1, \dots, N$$

da cui risulta che:

$$\begin{aligned} \min_{\nu \in FPS_\infty} \nu_j &= \min_{M \cdot \nu \leq b} c^T \cdot \nu \\ \max_{\nu \in FPS_\infty} \nu_j &= - \min_{M \cdot \nu \leq b} (-c)^T \cdot \nu \end{aligned}$$

dove:

$$M = \begin{bmatrix} L \\ -L \end{bmatrix}, \quad b = \begin{bmatrix} z^N \\ -z^N \end{bmatrix} + \varepsilon, \quad c = j\text{-esima colonna della matrice identità } I_{2 \times 2}$$

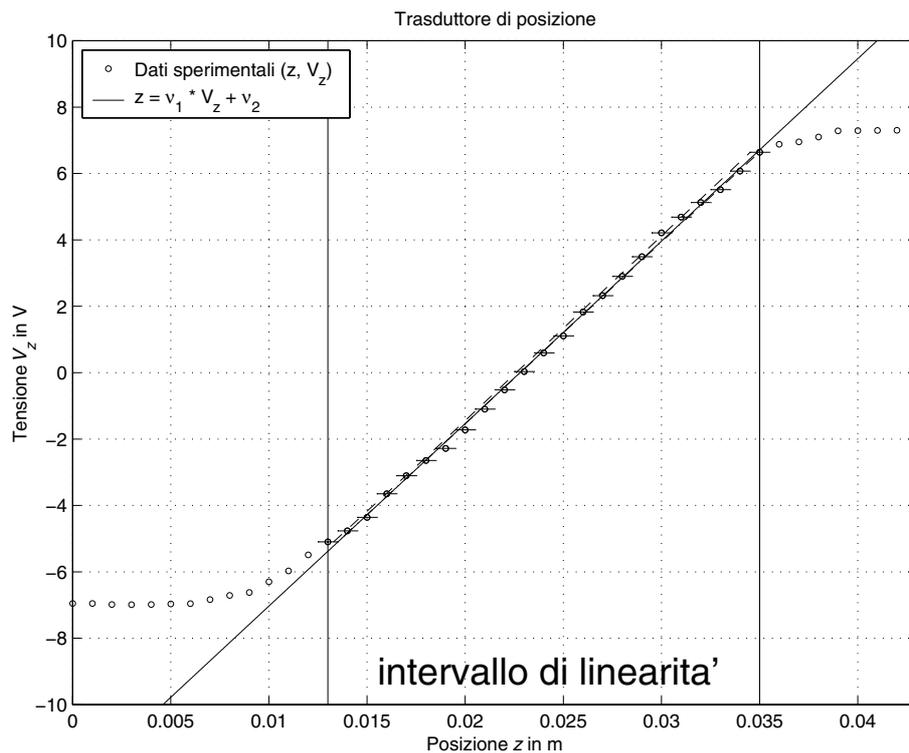
Nel caso in esame, risulta che:

$$\begin{aligned} \left[\min_{\nu \in FPS_\infty} \nu_1, \max_{\nu \in FPS_\infty} \nu_1 \right] &= [1.7909 \cdot 10^{-3}, 1.8484 \cdot 10^{-3}] \\ \left[\min_{\nu \in FPS_\infty} \nu_2, \max_{\nu \in FPS_\infty} \nu_2 \right] &= [2.2596 \cdot 10^{-2}, 2.2807 \cdot 10^{-2}] \end{aligned}$$

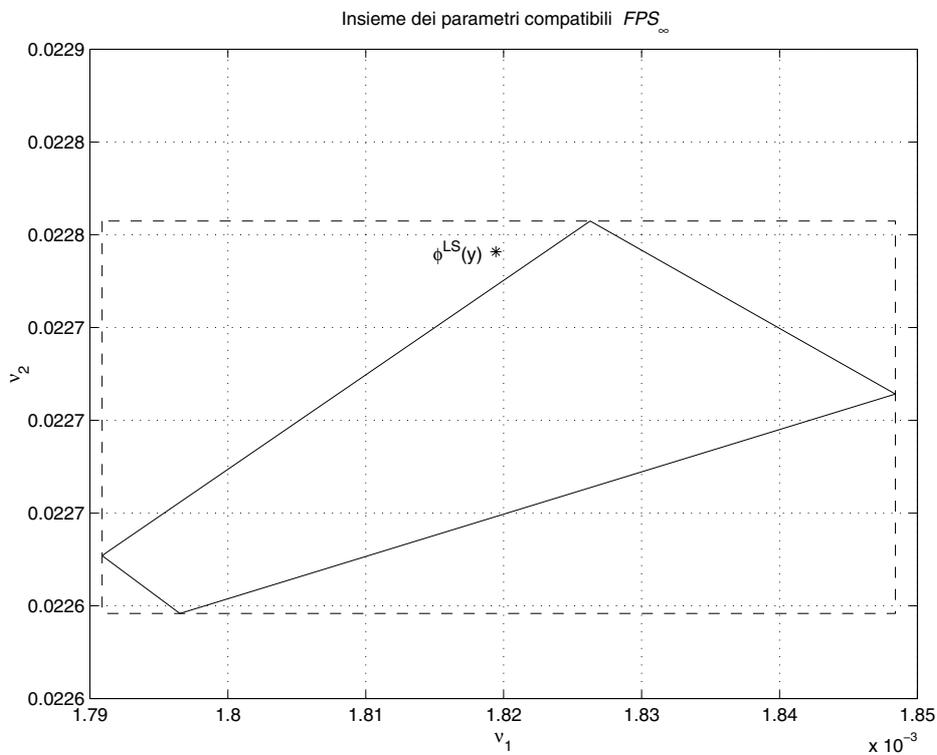
e quindi gli intervalli di incertezza dei parametri \hat{K}_t e \hat{V}_o compatibili sono pari a:

$$\begin{aligned} [\hat{K}_t^m, \hat{K}_t^M] &= [541.01, 558.38] \text{ V/m} \\ [\hat{V}_o^m, \hat{V}_o^M] &= [-12.735, -12.225] \text{ V} \end{aligned}$$

Nella figura seguente è riportato l'involuppo dell'insieme di modelli i cui parametri \hat{v} appartengono all'insieme dei parametri compatibili FPS_∞ .



Nella figura sottostante sono riportati sia l'insieme dei parametri compatibili FPS_∞ sia la regione individuata dagli estremi degli intervalli di incertezza dei parametri compatibili PUI_∞ .



4. Principali comandi MATLAB da prendere in considerazione

- lp
- convhull

Possibile soluzione mediante MATLAB (file sensore_sm.m)

```
% Identificazione set membership del modello di un sensore di posizione

clear all, close all, pack, clc
format compact, format short e

load sensore

% limite sull'errore massimo di misura della posizione z
epsilon=5e-4

% indici corrispondenti all'intervallo di linearita' della caratteristica
i1=14; % usare prima 11 e poi 14
i2=36;

z_lin = z(i1:i2);
Vz_lin=Vz(i1:i2);

%-----
% Stima dei parametri di un modello lineare con i minimi quadrati
%-----

% stima dei parametri mediante i minimi quadrati
L=[Vz_lin, ones(i2-i1+1,1)];
A=inv(L'*L)*L';

nu_hat=L\z_lin % calcolo mediante l'operatore "\"
nu_hat_=A*z_lin; % calcolo mediante la pseudoinversa A

%-----
% Calcolo degli intervalli di incertezza EUI in norma l-infinito
%-----

for ind=1:length(nu_hat),
    nu_min(ind,1)=A(ind,:)*(z_lin-epsilon*sign(A(ind,:)))';
end
nu_min
nu_max=2*nu_hat-nu_min

% verifica

nu_min_=[0;0]; nu_max_=[0;0];
for k=1:length(A),
    nu_min_(1)=nu_min_(1)+A(1,k)*(z_lin(k)-epsilon*sign(A(1,k)));
    nu_min_(2)=nu_min_(2)+A(2,k)*(z_lin(k)-epsilon*sign(A(2,k)));
    nu_max_(1)=nu_max_(1)+A(1,k)*(z_lin(k)+epsilon*sign(A(1,k)));
    nu_max_(2)=nu_max_(2)+A(2,k)*(z_lin(k)+epsilon*sign(A(2,k)));
end
nu_min_
nu_max_
nu_max__=2*nu_hat-nu_min_

EUI=[nu_min, nu_max]
Kt_min=1/EUI(1,2)
Kt_max=1/EUI(1,1)
Vo_min=-EUI(2,2)/EUI(1,1)
Vo_max=-EUI(2,1)/EUI(1,2)

% Confronto dei risultati ottenuti

Vz0=linspace(-10,10,1000);
z_hat=nu_hat(1)*Vz0+nu_hat(2);
z_min=min([nu_max(1)*Vz0; nu_min(1)*Vz0])+nu_min(2);
z_max=max([nu_max(1)*Vz0; nu_min(1)*Vz0])+nu_max(2);

figure, plot(z,Vz,'o', z_min,Vz0,'--', z_hat,Vz0,'-', z_max,Vz0,'-.', ...
            z(i1)*[1,1], [-10,10], '-.', z(i2)*[1,1], [-10,10], '-'),
axis([min(z),max(z),-10,10]), grid, title('Trasduttore di posizione'),
```

```

xlabel('Posizione{\it z} in m'), ylabel('Tensione{\it V_z} in V'),
legend('Dati sperimentali (z, V_z)_{\it }^{\it }', ...
      'z \approx \min(\nu_{1}^{\{M\}} * V_{z} , \nu_{1}^{\{m\}} * V_{z}) + \nu_{2}^{\{m\}}', ...
      'z = \nu_{1} * V_{z} + \nu_{2}', ...
      'z \approx \max(\nu_{1}^{\{M\}} * V_{z} , \nu_{1}^{\{m\}} * V_{z}) + \nu_{2}^{\{M\}}', 2),
text(0.015,-9,'\fontsize{20} intervallo di linearita'''),
print -deps sensore5.eps

%-----
% Calcolo degli intervalli di incertezza PUI in norma l-infinito
%-----

PUI(1,1)=[1; 0]*lp( [1; 0],[L; -L],[z_lin+epsilon; -z_lin+epsilon]);
PUI(2,1)=[0; 1]*lp( [0; 1],[L; -L],[z_lin+epsilon; -z_lin+epsilon]);
PUI(1,2)=[1; 0]*lp(-[1; 0],[L; -L],[z_lin+epsilon; -z_lin+epsilon]);
PUI(2,2)=[0; 1]*lp(-[0; 1],[L; -L],[z_lin+epsilon; -z_lin+epsilon]);

PUI
Kt_min=1/PUI(1,2)
Kt_max=1/PUI(1,1)
Vo_min=-PUI(2,2)/PUI(1,1)
Vo_max=-PUI(2,1)/PUI(1,2)

% Confronto dei risultati ottenuti

clear z_min z_max
nu_vect=[];
for k=1:length(Vz_lin),
    nu=lp([Vz_lin(k); 1],[L; -L],[z_lin+epsilon; -z_lin+epsilon]);
    nu_vect=[nu_vect, nu];
    z_min(k)=[Vz_lin(k), 1]*nu;
    nu=lp(-[Vz_lin(k); 1],[L; -L],[z_lin+epsilon; -z_lin+epsilon]);
    nu_vect=[nu_vect, nu];
    z_max(k)=[Vz_lin(k), 1]*nu;
end

figure, plot(z,Vz,'o')
child=get(get(gcf,'Child'),'Child'); set(child(1),'MarkerSize',3);
hold on,
plot(z_hat,Vz0,'-', z_min,Vz_lin,'--', z_max,Vz_lin,'-.', ...
      z(i1)*[1,1],[-10,10],'-', z(i2)*[1,1],[-10,10],'-'),
plot_dx(z_lin,Vz_lin,epsilon,epsilon*0)
hold off,
axis([min(z),max(z),-10,10]), grid on, zoom on, title('Trasduttore di posizione'),
xlabel('Posizione{\it z} in m'), ylabel('Tensione{\it V_z} in V'),
legend('Dati sperimentali (z, V_z)_{\it }^{\it }', ...
      'z = \nu_{1} * V_{z} + \nu_{2}', 2),
text(0.015,-9,'\fontsize{20} intervallo di linearita'''),
print -deps sensore6.eps

k=convhull(nu_vect(1,:),nu_vect(2,:));
figure, plot(nu_vect(1,k),nu_vect(2,k),'-',nu_hat(1),nu_hat(2),'*', ...
      [PUI(1,1),PUI(1,:),PUI(1,2:-1:1)], [PUI(2,2:-1:1),PUI(2,:),PUI(2,2)],'--'),
grid on, text(nu_hat(1)*0.9975,nu_hat(2),'phi^{\{LS\}}(y)'),
title('Insieme dei parametri compatibili {\it FPS_{\infty}}'),
xlabel('\nu_{1}'), ylabel('\nu_{2}')
print -deps sensore7.eps

```