Parametric estimation of a static model for a position transducer using the Set Membership approach

1. Linear approximation of the position-voltage characteristic
The position-voltage characteristic of the transducer seen in the laboratory #1 shows a nearly linear behaviour in the interval between 1.3 and 3.5 cm. In such a linearity interval, the characteristic can be described by the following model

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

where the gain K_t and the offset voltage V_o are unknown constant parameters to be estimated. Since the most relevant error is due to the position measurement (the greatest position error is about 0.5 mm), the model equation can be more suitably rewritten in the following form, where the measurement error e is explicitly taken into account:

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

By considering the N measurements collected in the linearity interval, the following system of linear equations is derived:

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

where $V_{z,i}$ is the voltage provided by the transducer when the position value is z_i.
   By introducing the unknown parameters:

$$\theta_1 = \frac{1}{K_t}, \quad \theta_2 = -\frac{V_o}{K_t}$$

the previous equations can be rewritten as:

$$
\begin{aligned}
z_1 &= V_{z,1} \cdot \theta_1 + \theta_2 + e_1 \\
z_2 &= V_{z,2} \cdot \theta_1 + \theta_2 + e_2 \\
&\vdots \\
z_N &= V_{z,N} \cdot \theta_1 + \theta_2 + e_N
\end{aligned}
$$

or, in matrix form:

$$
\begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_N \end{bmatrix}
=
\begin{bmatrix} V_{z,1} & 1 \\ V_{z,2} & 1 \\ \vdots & \vdots \\ V_{z,N} & 1 \end{bmatrix}
\cdot
\begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix}
+
\begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_N \end{bmatrix}
$$

i.e., the estimation problem is recast in the standard form:

$$y = \Phi \cdot \theta + e$$

where $y \in \mathbb{R}^N$, $\Phi \in \mathbb{R}^{N \times 2}$, $e \in \mathbb{R}^N$ and $\theta \in \mathbb{R}^2$ is the unknown.

Using the least squares algorithm as estimation method:

$$\hat{\theta} = A \cdot y, \qquad \text{with } A = \left(\Phi^T \cdot \Phi\right)^{-1} \Phi^T \doteq \Phi^{\dagger}$$

it results that:

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

2. Evaluation of the estimate uncertainty intervals EUI^{∞}

The main error source in the experimental measurements is the measurement of the position z and the greatest error is about 0.5 mm. The measurement error e can then be considered as unknown but bounded and, in particular;

$$e \in \mathcal{B}_e^{\infty} = \left\{ \tilde{e} \in \mathbb{R}^N : |\tilde{e}_k| \leq \varepsilon, \; k = 1, \ldots, N \right\}, \quad \text{with } \varepsilon = 5 \cdot 10^{-4}$$

Note that $\mathcal{B}_e^{\infty}$ is a N-dimensional hypercube with side length $2 \cdot \varepsilon$ centered in the origin.

The estimate uncertainty intervals EUI^{∞} are defined as:

$$EUI_j^{\infty} = \left[\hat{\theta}_j^m = \min_{\theta \in EUS^{\infty}} \theta_j, \, \hat{\theta}_j^M = \max_{\theta \in EUS^{\infty}} \theta_j \right], \quad j = 1, 2$$

where the estimate uncertainty set EUS^{∞} is the image under the linear mapping A of the measurement uncertainty set MUS^{∞}, which is a N-dimensional hypercube with side length $2 \cdot \varepsilon$ centered in y:

$$EUS^{\infty} = A \cdot MUS^{\infty} = A \cdot (y \oplus \mathcal{B}_e^{\infty}) = A \cdot \left\{ \tilde{y} \in \mathbb{R}^N : |\tilde{y}_i - y_i| \leq \varepsilon, \; i = 1, \ldots, N \right\}$$

Since it can be proven that:

$$\hat{\theta}_j^m = \min_{\theta \in EUS^{\infty}} \theta_j = \sum_{k=1}^N A_{jk} \cdot [y_k - \varepsilon \cdot \text{sign}(A_{jk})]$$

$$\hat{\theta}_j^M = \max_{\theta \in EUS^{\infty}} \theta_j = \sum_{k=1}^N A_{jk} \cdot [y_k + \varepsilon \cdot \text{sign}(A_{jk})] = 2 \cdot \hat{\theta}_j - \hat{\theta}_j^m$$

in this case it results that:

$$EUI_1^{\infty} = \left[\hat{\theta}_1^m, \hat{\theta}_1^M \right] = [1.6996 \cdot 10^{-3}, 1.9392 \cdot 10^{-3}]$$

$$EUI_2^{\infty} = \left[\hat{\theta}_2^m, \hat{\theta}_2^M \right] = [2.2291 \cdot 10^{-2}, 2.3291 \cdot 10^{-2}]$$

and then the uncertainty intervals on the estimates $\hat{K}_t$ and $\hat{V}_o$ are equal to:

$$\left[\hat{K}_t^m, \hat{K}_t^M \right] = \left[\frac{1}{\hat{\theta}_1^M}, \frac{1}{\hat{\theta}_1^m} \right] = [515.67, 588.36] \, \text{V/m}$$

$$\left[\hat{V}_o^m, \hat{V}_o^M \right] = \left[-\frac{\hat{\theta}_2^M}{\hat{\theta}_1^m}, -\frac{\hat{\theta}_2^m}{\hat{\theta}_1^M} \right] = [-13.703, -11.495] \, \text{V}$$

In the figure below, it is plotted the envelope of the static characteristics of models whose parameters θ are taken as the extremes of the estimate uncertainty intervals EUI_j^{∞}, $j = 1, 2$.

Position transducer (uncertainty region due to EUI)
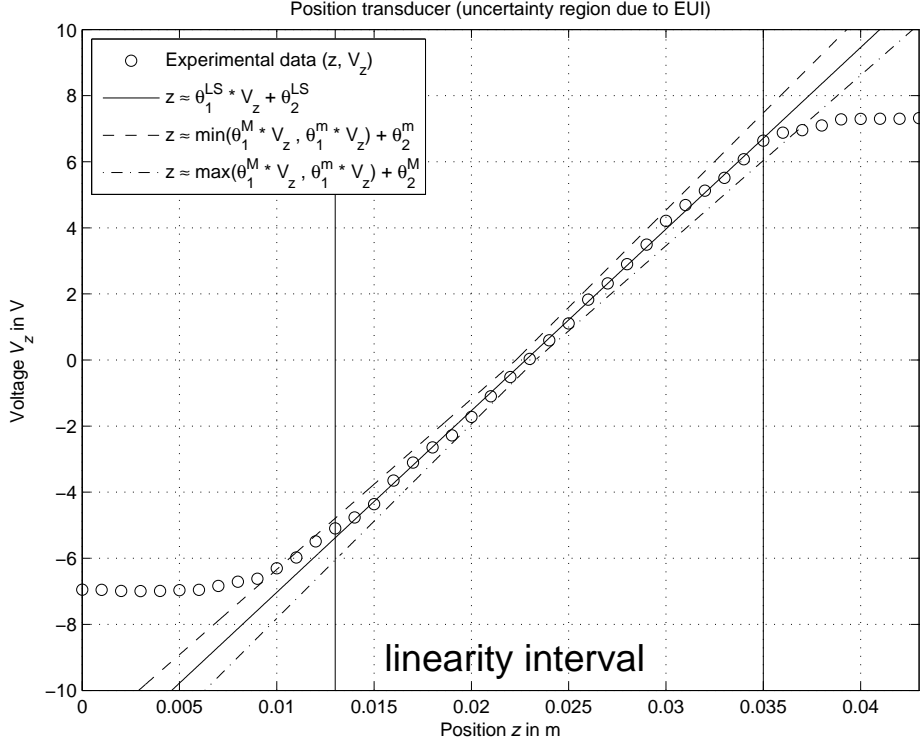
3. *Evaluation of the feasible parameter uncertainty intervals PUI^∞*

The feasible parameter uncertainty intervals PUI^∞ are defined as:

$$PUI_j^\infty = \left[\min_{\theta \in FPS^\infty} \theta_j, \max_{\theta \in FPS^\infty} \theta_j \right] \subseteq EUI_j^\infty, \quad j = 1, 2$$

where the feasible parameter set FPS^∞ is a polytope (i.e., a convex polyhedron) generated by linear inequalities:

$$FPS^\infty = \left\{ \tilde{\theta} \in \mathbb{R}^{dim(\tilde{\theta})} : \left| y_i - [\Phi \cdot \tilde{\theta}]_i \right| \leq \varepsilon, \ i = 1, \ldots, N \right\}$$

The extremes of the intervals PUI_j^∞ are obtained as solutions of linear programming problems. To this purpose, the MATLAB command `linprog` in its actual (R2014A) form `linprog(c,M,b,[],[],[],[],[],optimset(optimset('linprog'),'Algorithm','simplex'))` allow to solve the linear programming optimization problem

$$\min_x c^T \cdot x \quad \text{with the constraint } M \cdot x \leq b$$

Note that `linprog` returns as first output argument the vector $x \in \mathbb{R}^{dim(\theta)}$ that minimizes the objective function $c^T \cdot x$ where $c \in \mathbb{R}^{dim(\theta)}$, i.e., not the minimum of $c^T \cdot x$.
It is then necessary to suitably rewrite the inequalities that define FPS^∞:

$$\left| y_i - [\Phi \cdot \tilde{\theta}]_i \right| \leq \varepsilon \Leftrightarrow -\varepsilon \leq y_i - [\Phi \cdot \tilde{\theta}]_i \leq \varepsilon \Leftrightarrow \left\{ \begin{array}{l} [\Phi \cdot \tilde{\theta}]_i \leq y_i + \varepsilon \\ -[\Phi \cdot \tilde{\theta}]_i \leq -y_i + \varepsilon \end{array} \right. , \ i = 1, \ldots, N$$

and then it follows that:

$$\begin{aligned} \theta_j^m &= \min_{\theta \in FPS^\infty} \theta_j = \min_{M \cdot \theta \leq b} c^T \cdot \theta \\ \theta_j^M &= \max_{\theta \in FPS^\infty} \theta_j = -\min_{\theta \in FPS^\infty} (-\theta_j) = -\min_{M \cdot \theta \leq b} (-c)^T \cdot \theta \end{aligned}$$

where:

$$M = \left[\begin{array}{c} \Phi \\ -\Phi \end{array} \right], \quad b = \left[\begin{array}{c} y \\ -y \end{array} \right] + \varepsilon, \quad c = j\text{-th column of the identity matrix } I_{2 \times 2}$$

3

In this case it results that:

$$PUI_1^\infty = \left[\theta_1^m = \min_{\theta \in FPS^\infty} \theta_1, \theta_1^M = \max_{\theta \in FPS^\infty} \theta_1\right] = [1.7909 \cdot 10^{-3}, 1.8484 \cdot 10^{-3}]$$

$$PUI_2^\infty = \left[\theta_2^m = \min_{\theta \in FPS^\infty} \theta_2, \theta_2^M = \max_{\theta \in FPS^\infty} \theta_2\right] = [2.2596 \cdot 10^{-2}, 2.2807 \cdot 10^{-2}]$$

and then the uncertainty intervals on the feasible parameters K_t e V_o are equal to:

$$\left[K_t^m, K_t^M\right] = \left[\frac{1}{\theta_1^M}, \frac{1}{\theta_1^m}\right] = [541.01, 558.38] \text{ V/m}$$

$$\left[V_o^m, V_o^M\right] = \left[-\frac{\theta_2^M}{\theta_1^m}, -\frac{\theta_2^m}{\theta_1^M}\right] [-12.735, -12.225] \text{ V}$$

4. Evaluation of the central estimate of the parameters

The central estimate of unknown parameter θ is given by:

$$\theta^C = \left[\begin{array}{c} \theta_1^C \\ \theta_2^C \end{array}\right] = \left[\begin{array}{c} \left(\min_{\theta \in FPS^\infty} \theta_1 + \max_{\theta \in FPS^\infty} \theta_1\right)\Big/ 2 \\ \left(\min_{\theta \in FPS^\infty} \theta_2 + \max_{\theta \in FPS^\infty} \theta_2\right)\Big/ 2 \end{array}\right] = \left[\begin{array}{c} 1.8196 \cdot 10^{-3} \\ 2.2702 \cdot 10^{-2} \end{array}\right]$$

and the corresponding central estimate of the parameters K_t e V_o is equal to:

$$K_t^C = \frac{1}{\theta_1^C} = 549.56 \,\text{V/m}$$

$$V_o^C = -\frac{\theta_2^C}{\theta_1^C} = -12.476 \,\text{V}$$

In the following figure, it is plotted the envelope of the static characteristics of models whose parameters θ are taken as the extremes of the parameter uncertainty intervals PUI_j^∞, $j = 1, 2$.
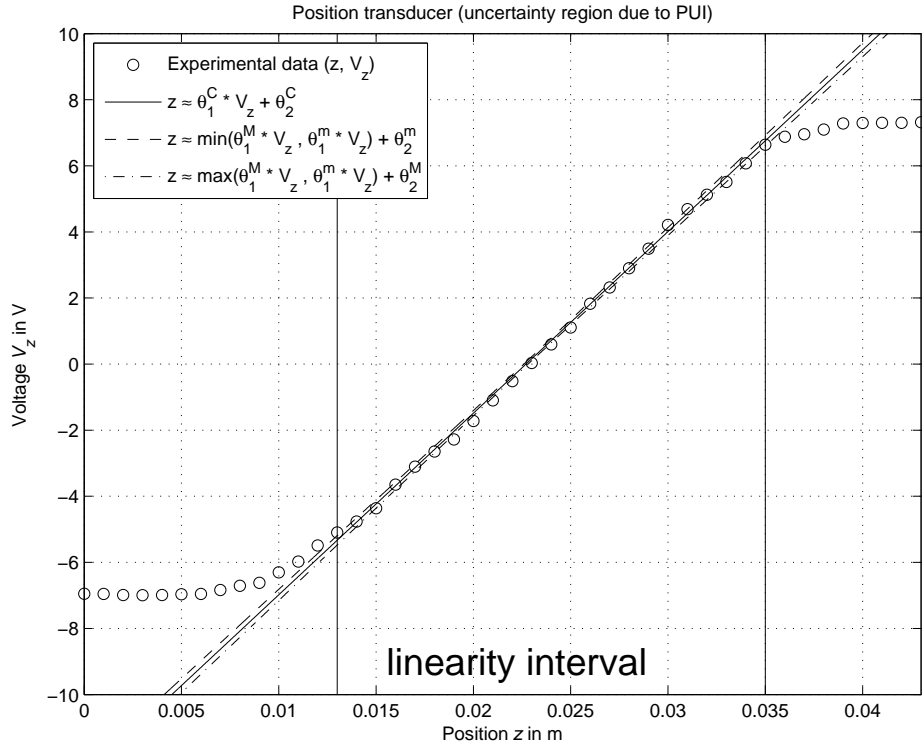
In the figure below, a polytopic outer approximation of the feasible parameter set FPS^∞ (continuous line) and the set of estimates given by the extremes of the feasible parameter uncertainty intervals PUI_j^∞, $j = 1, 2$, are plotted.

Feasible Parameter Set FPS^∞
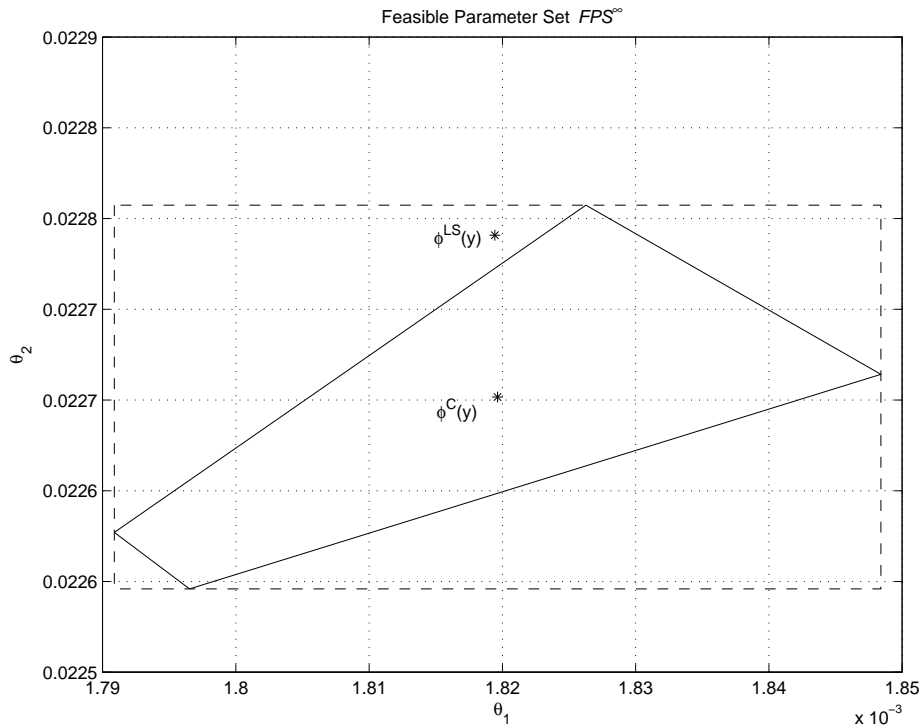
5. *Main MATLAB commands to take into account*

- `linprog`
- `convhull`

Possible solution under MATLAB (file sensor_sm.m)

```
%% Laboratory 2 - Estimation, filtering and system identification - Prof. M. Taragna
% *Parametric estimation of a position transducer model*
% *using the Set Membership approach*
%% Introduction
% The program code may be splitted in sections using the characters "%%".
% Each section can run separately with the command "Run Section"
% (in the Editor toolbar, just to the right of the "Run" button). You can do the
% same thing by highlighting the code you want to run and by using the button
% function 9 (F9). This way, you can run only the desired section of your code,
% saving your time. This script can be considered as a reference example.

clear all, close all, clc

%% Procedure
% # Load the file |sensor.mat| containing the position/voltage data
% # Plot the voltage measurements versus the position measurements
% # Estimate the parameters of the linear approximation using least squares
% # Plot the estimated approximation versus the experimental data
% # Compute the estimate uncertainty intervals EUI in l-infinity norm
% # Plot the EUI versus the estimated approximation
% # Compute the parameter uncertainty intervals PUI in l-infinity norm
% # Compute the parameter central estimate in l-infinity norm
% # Plot the PUI and the central estimate versus the experimental data
% # (Optional) Compute an approximation of the feasible parameter set FPS
% # (Optional) Plot PUI and FPS approximation versus the estimated parameters

%% Problem setup

% Step 1: load of data

load sensor
% z  = position measured in meters, without position offset
% Vz = voltage measured in volts
```

```matlab
% Step 2: plot of data

figure, plot(z,Vz,'o'),
axis([min(z),max(z),-10,10]), grid, title('Position transducer'),
xlabel('Position{\it z} in m'), ylabel('Voltage{\it V_z} in V')

%% Parametric estimation of a linear model (w.r.t. data) using least squares

% Step3: computation of the parameter estimate

% definition of the linearity interval of the characteristic
i1=14; % z(11) = 0.01, z(14) = 0.013; use: i1=11 (at first), i1=14 (at the end)
i2=36; % z(36) = 0.035
z_lin=z(i1:i2);
Vz_lin=Vz(i1:i2);
N_lin=length(z_lin);

% parameter estimation by means of least squares algorithm
Phi=[Vz_lin, ones(N_lin,1)];
p=Phi\z_lin; % Form #1: using the "\" operator (more realiable)
Kt=1/p(1)
Vo=-p(2)/p(1)

A=pinv(Phi); % more realiable than inv(Phi'*Phi)*Phi'
p_=A*z_lin;  % Form #2: using the pseudoinverse matrix
Kt_=1/p_(1)
Vo_=-p_(2)/p_(1)

% Step 4: graphical comparison of the results

Vz0=linspace(-10,10,1000);
z_hat=p(1)*Vz0+p(2);
figure, plot(z,Vz,'o', z_hat,Vz0,'-', ...
             z_lin(1)*[1,1],[-10,10],'-', z_lin(end)*[1,1],[-10,10],'-'),
axis([min(z),max(z),-10,10]), grid, title('Position transducer'),
xlabel('Position{\it z} in m'), ylabel('Voltage{\it V_z} in V'),
text(0.015,-9,'\fontsize{20} linearity interval')

%% Evaluation of estimate uncertainty intervals EUI in l-infinity norm

% Step 5: computation of the estimate uncertainty intervals EUI in l-infinity norm

% bound on the greatest measurement error of the position z
format compact, format short e
epsilon=0.5e-3

% EUI algorithm in compact form
for ind=1:length(p),
    p_min(ind,1)=A(ind,:)*(z_lin-epsilon*sign(A(ind,:)))';
end
p_max=2*p-p_min;
EUI=[p_min, p_max]

% EUI algorithm in extended form, using a loop (equivalent)
p_min_=zeros(size(p)); ; p_max_=zeros(size(p));
for k=1:length(A),
    p_min_(1)=p_min_(1)+A(1,k)*(z_lin(k)-epsilon*sign(A(1,k)));
    p_min_(2)=p_min_(2)+A(2,k)*(z_lin(k)-epsilon*sign(A(2,k)));
    p_max_(1)=p_max_(1)+A(1,k)*(z_lin(k)+epsilon*sign(A(1,k)));
    p_max_(2)=p_max_(2)+A(2,k)*(z_lin(k)+epsilon*sign(A(2,k)));
end
EUI_=[p_min_, p_max_]

format compact, format short
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)

% Step 6: graphical comparison of the results

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_hat,Vz0,'-', ...
             z_min,Vz0,'--', z_max,Vz0,'-.', ...
             z_lin(1)*[1,1],[-10,10],'-',z_lin(end)*[1,1],[-10,10],'-')
axis([min(z),max(z),-10,10]), grid,
title('Position transducer (uncertainty region due to EUI)'),
xlabel('Position{\it z} in m'), ylabel('Voltage{\it V_z} in V'),
legend('Experimental data (z, V_z)_{ }^{  }', ...
       'z \approx \theta_{1}^{LS} * V_{z} + \theta_2^{LS}', ...
       'z \approx min(\theta_{1}^{M} * V_{z} , \theta_{1}^{m} * V_{z}) + \theta_{2}^{m}', ...
       'z \approx max(\theta_{1}^{M} * V_{z} , \theta_{1}^{m} * V_{z}) + \theta_{2}^{M}', 2),
text(0.015,-9,'\fontsize{20} linearity interval')

%% Evaluation of parameter uncertainty intervals PUI and central estimate in l-infinity norm

% Step 7: computation of the parameter uncertainty intervals PUI in l-infinity norm

options_old=optimset('linprog'); % default value for 'Algorithm' option is 'interior-point'
options_new=optimset(options_old, 'Algorithm','simplex');
PUI(1,1)=[1; 0]'*linprog( [1; 0],[Phi; -Phi],[z_lin+epsilon; -z_lin+epsilon], ...
                          [],[],[],[],[],options_new);
PUI(2,1)=[0; 1]'*linprog( [0; 1],[Phi; -Phi],[z_lin+epsilon; -z_lin+epsilon], ...
                          [],[],[],[],[],options_new);
PUI(1,2)=[1; 0]'*linprog(-[1; 0],[Phi; -Phi],[z_lin+epsilon; -z_lin+epsilon], ...
                          [],[],[],[],[],options_new);
PUI(2,2)=[0; 1]'*linprog(-[0; 1],[Phi; -Phi],[z_lin+epsilon; -z_lin+epsilon], ...
                          [],[],[],[],[],options_new);

format compact, format short e
PUI

format compact, format short
Kt_min_PUI=1/PUI(1,2)
Kt_max_PUI=1/PUI(1,1)
Vo_min_PUI=-PUI(2,2)/PUI(1,1)
Vo_max_PUI=-PUI(2,1)/PUI(1,2)

% Step 8: computation of the central estimate

format compact, format short e
p_central=[(PUI(1,1)+PUI(1,2))/2; (PUI(2,1)+PUI(2,2))/2]

format compact, format short
Kt_central=1/p_central(1)
Vo_central=-p_central(2)/p_central(1)

% Step 9: graphical comparison of the results

z_hat_central=p_central(1)*Vz0+p_central(2);
z_min_central=min([PUI(1,2)*Vz0; PUI(1,1)*Vz0])+PUI(2,1);
z_max_central=max([PUI(1,2)*Vz0; PUI(1,1)*Vz0])+PUI(2,2);

figure, plot(z,Vz,'o', z_hat_central,Vz0,'-', ...
             z_min_central,Vz0,'--', z_max_central,Vz0,'-.', ...
             z_lin(1)*[1,1],[-10,10],'-',z_lin(end)*[1,1],[-10,10],'-')
axis([min(z),max(z),-10,10]), grid,
title('Position transducer (uncertainty region due to PUI)'),
xlabel('Position{\it z} in m'), ylabel('Voltage{\it V_z} in V'),
legend('Experimental data (z, V_z)_{ }^{  }', ...
       'z \approx \theta_{1}^{C} * V_{z} + \theta_2^{C}', ...
       'z \approx min(\theta_{1}^{M} * V_{z} , \theta_{1}^{m} * V_{z}) + \theta_{2}^{m}', ...
       'z \approx max(\theta_{1}^{M} * V_{z} , \theta_{1}^{m} * V_{z}) + \theta_{2}^{M}', 2),
```

```
text(0.015,-9,'\fontsize{20} linearity interval')

%% (Optional) Evaluation of an approximation of the feasible parameter set FPS

% Step 10: computation of a polytopic outer approximation of the FPS

n=16; % Number of vertices of the polytope
theta_vect=[];
for k=0:n-1,
    theta=linprog([sin(2*pi*k/n),cos(2*pi*k/n)],[Phi;-Phi],[z_lin+epsilon;-z_lin+epsilon],...
                  [],[],[],[],[],options_new);
    theta_vect=[theta_vect, theta];
end

% Step 11: graphical comparison of the results

k=convhull(theta_vect(1,:),theta_vect(2,:));
figure, plot(theta_vect(1,k),theta_vect(2,k),'-',p(1),p(2),'*', ...
             [PUI(1,1),PUI(1,:),PUI(1,2:-1:1)], [PUI(2,2:-1:1),PUI(2,:),PUI(2,2)],'--', ...
             p_central(1),p_central(2),'*k'),
grid on, text(p(1)*0.9975,p(2),'\phi^{LS}(y)'),
text(p_central(1)*0.9975,p_central(2)*0.9997,'\phi^{C}(y)'),
title('Feasible Parameter Set {\it FPS^\infty}'),
xlabel('\theta_{1}'), ylabel('\theta_{2}')
```