Assignment 3

1.

Piezo stack actuators are used as linear electromechanical drives or motors. They act mainly like an expanding element generating a compressive force. The complete motion cycle is nearly proportional to the voltage signal input from DC source. We have a piezoelectric actuator in the lab, but the specifications are unknown. We are interested in the gain of the actuator (G), which is the ratio of the resulted linear displacement to the applied electrical voltage.

We prepared the following test setup (Fig. 1). We send a signal from PC through an AD/DA board and a voltage amplifier to the actuator, and record the corresponding displacement of the actuator by a laser sensor.

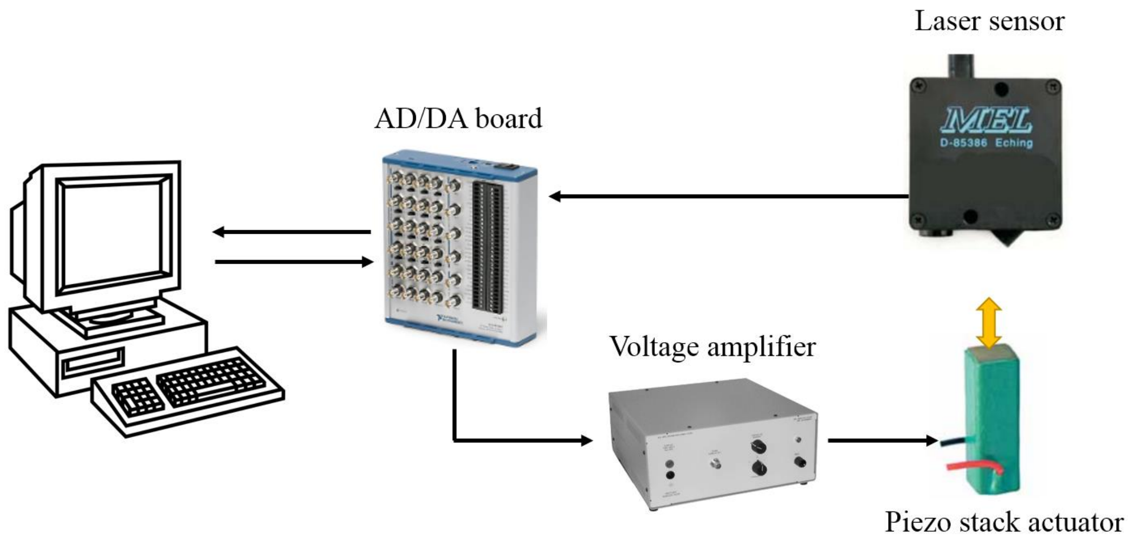

Figure 1 – Schematics of the test setup

In this exercise, we want to estimate the gain of the actuator G based on the recorded data,

$$d = Gv$$

Where v is the applied internal voltage, and d is the displacement of the actuator. To estimate G, our method is to excite the actuator with a known voltage and measure the displacement with the laser sensor. Then, we can estimate G using the aforementioned equation.

We have performed a series of measurements and the results are stored in data file named “data1.mat”. Load the data file “data1.mat”. The applied voltages are stored in the variable “v”, and the corresponding displacement are stored in the variable “d”. Use MATLAB curve fitting toolbox to estimate G. Estimate the displacement of the actuator for the internal voltage of 75 volts. (Alternatively, you can use polyval and polyfit functions to estimate the parameters)

Report the relevant information including the estimated gain G, plot of data points, plot of fitted function for $v \epsilon [0,100]$ and the estimated value for $v = 75\ volts$.

2.

Particle tracking velocimetry (PTV) can be used to model the motion of objects in field-based manipulation techniques. In this method, we record the motion of objects with a camera, and track the objects frame by frame. Then, we can form a data structure including the initial positions of the objects and the recorded displacements. Fig. 2 illustrates a schematic of particle tracking velocimetry on a vibrating plate.

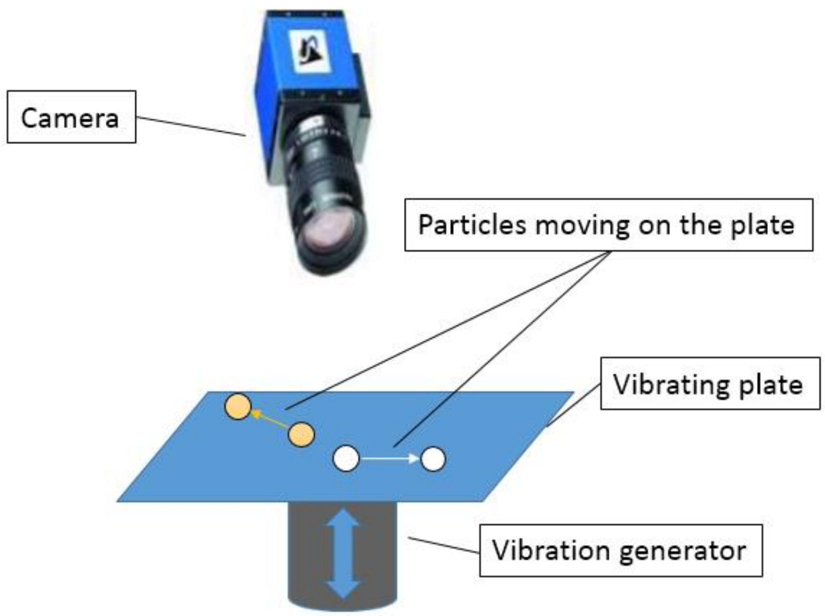

Figure 2 – Particle tracking velocimetry on a vibrating plate

Load the data file "data2.mat". Consider x and y as training inputs and u and v as training outputs. Fig.3 represents the schematic top-view of the plate. (x, y) pairs represent a point on a plate, and (u, v) pairs represent the displacements of particles on the same point in x and y directions (Fig 3). The goal in this exercise is to find the displacements (u, v) as functions of position (x, y).

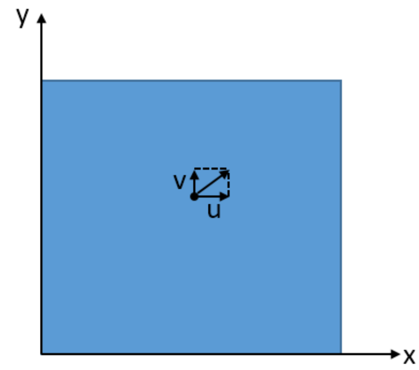

Figure 3 – Particles are positioned on a plate, u and v are displacement components of particles as functions of position (x,y)

a) Fit polynomial surfaces f and g to data with the following definition. Assume second order polynomials for both fitted functions in both x and y directions.

$$u = f(x, y)$$

$$v = g(x, y)$$

Generate a function named "createFit.m" using MATLAB **curve fitting toolbox**, which estimates the parameters with the same structure of fitted functions.

function [fitresult, gof] = createFit(x, y, u)

b) Write a script named "problem2.m". This script should call "createFit.m", and calculate u and v for the following grid space.

[gridx,gridy] = meshgrid(0:0.05:1,0:0.05:1);

Finally, visualize the function using quiver plot. (Tip: see MATLAB command "quiver")

3.

Load the data file "data3.mat". Consider x and y as training input and output, and xv and yv as validation input and output. Write a script called "problem3.m". The script should perform the following tasks. You are not allowed to use MATLAB curve fitting toolbox for this question.

a) Draw the scatter plot of (x, y) data. Estimate the parameters k_i in the following model using the **<u>closed-form solution</u>** for the given data.

$$y = k_1 x^{k_2}$$

b) Plot the estimated function for $x \epsilon [0{,}15]$ on the same scatter plot of data together with the validation data. Calculate SSE and R^2 for the training data. Calculate SSE for the validation data.

c) Estimate a polynomial model for the given data using MATLAB polynomial estimation functions, "polyfit" and "polyval". What polynomial order do you choose for the given data? Why?

4.

Load the data file “data4.mat”. Consider x and y as training input and output. Estimate the parameters k_i in the following nonlinear model. Write a script called “problem4.m”. The script should perform the following tasks. You are not allowed to use MATLAB curve fitting toolbox for this question.

$$\boldsymbol{y = k_1(1 - e^{-k_2 x})}$$

a) Write MATLAB a function named “computeCost.m” to compute the sum of squared error between the data and the aforementioned function.

b) Write a MATLAB script named “problem4.m”. This script should use MATLAB minimization search algorithms (for instance “fminsearch”) to find the estimation parameters by minimizing the cost function. You can initialize the search from the following values,

$$k_1 = 1$$
$$k_2 = 1$$

c) Visualize the cost function in the neighborhood of the global minima. (Tip: use “surf” command in MATLAB)

What to return?

You are supposed to submit your assignment to the related link for assignment 3 in MyCourses. Your submission should include one zip file “Assign03_student number.zip” consisting of a pdf file “Assign03_student number.pdf”, and five MATLAB scripts “problem2.m”, “createFit.m”, “problem3.m”, “problem4.m”, and “computeCost.m”.

The hard deadline for submission of this assignment is 21.10.2018 at 23:55.