15

A Fuzzy Finite Element Method Programmed in MATLAB for the Analysis of Uncertain Control Systems of Structures

Angel L. Morales[1], Jem A. Rongong[2] and Neil D. Sims[2]
[1]*Universidad de Castilla - La Mancha*
[2]*The University of Sheffield*
[1]*Spain*
[2]*United Kingdom*

1. Introduction

Vibration is an important aspect of many engineering systems. In most cases, such vibration is undesirable and requires attenuation or control, which explains the huge quantity of vibration control techniques that can be found in the literature. Nevertheless, sometimes all this knowledge is not enough to guarantee the success in this purpose. Model inaccuracies or parameter uncertainty are unavoidable in all relevant practical application and not only can they degrade the performance of an otherwise well-designed control system, but they may also lead to instability and even structural failure.

For these reasons, robustness is of crucial importance in control-system design. Although always being appreciated, it was not until late 1970s when a theory capable to explicitly handle the robustness issue in feedback design was developed: the H_∞ optimal control theory (Zames, 1981; Zames & Francis, 1983). This optimization approach is well developed and we can even find a full *Robust Control Toolbox* in MATLAB which, in collaboration with the *Control System Toolbox*, let us identify worst-case scenarios and generate optimal controllers so that the stability and good performance of the plant is ensured (Gu et al., 2005).

Obviously, the H_∞ optimal control theory shows some disadvantages, like the high level of mathematical understanding needed to apply them successfully and the fact that it sometimes leads to potentially very conservative results (Gu et al., 2005). Indeed, even when a solution via H_∞ techniques is eventually found, the process may have been too expensive in terms of designer's time and the sensitivity to the uncertainty sources is not analyzed. These motivations make us think that a straightforward possibilistic methodology for simulating and analyzing the uncertainty propagation in control system would be a very valuable, complementary and time saving tool.

This work focuses on how MATLAB can supply suitable tools in order to manage uncertainty propagation in finite element models of structures from the point of view of fuzzy arithmetic. More specifically, we will describe a methodology fully programmed in MATLAB in order to deal with uncertain control systems of vibrating structures and we will illustrate the application of this approach to a case study.

In order to develop this work we need to integrate two different problems: the calculation of the dynamic response of a vibrating structure via finite element models and the study of the

uncertainty propagation via fuzzy arithmetic. Both problems can be solved using MATLAB code obtaining very good results in terms of accuracy and computational cost.
The motivation behind this work is threefold: firstly, it is meant to show how MATLAB provides suitable functions and specialized toolboxes in order to study the static or dynamic response of a controlled structure via finite elements and control theory; secondly, it demonstrates the suitability of MATLAB to solve uncertainty propagation problems through fuzzy arithmetic minimizing the computational cost; and thirdly, it illustrates the application of a Fuzzy Finite Element procedure, which integrates these two tools, with a specific case study.
The chapter is organized in the following way: Section 1 has summarized the aims and concerns of the chapter; Section 2 deals with dynamic analysis of structures with active damping by means of the *OpenFEM Toolbox* and the *Control System Toolbox* of MATLAB; Section 3 discusses the different types of uncertainty and how they could be analyzed using fuzzy arithmetic programmed in MATLAB; Section 4 describes a Fuzzy Finite Element procedure in MATLAB in order to study the propagation of uncertain parameters in control systems of structures; Section 5 shows a case study in which the Fuzzy Finite Element procedure is tested; Section 6 deals with the main conclusions of this work; and finally the acknowledgements and references included through the text are included.

2. Dynamic analysis and control of structures

2.1 Finite Element analysis and the OpenFEM Toolbox

The study of the dynamic behaviour of a structure can be carried out by means of finite element analysis. Among all the different possibilities to solve finite elements problems, in this chapter we focus on the use of a specific toolbox of MATLAB: the *OpenFEM Toolbox* (Balmes et al., 2009).
It is an open-source toolbox for finite element analysis within the matrix computing environment MATLAB. The software is developed in collaboration between Macs and the SDTools Company and it is distributed under a GNU Lesser General Public License, that is, a free software license which also allows the user to make modifications in the code. Performing finite element analysis within a matrix computing environment is of considerable interest, in particular as regards the ease of new developments, integration of external software, portability, post-processing, etc. This relatively young software is already quite successful in the finite element community as it is proven by the about 300 downloads per month (OpenFEM, 2011).
Though the toolbox architecture includes graphical user interfaces for visualization, analysis and animation of results, the powerful of this toolbox does not lie in this fact but in the standard and advanced methods which are implemented. Moreover, the provided functions are open and can be easily extended to suit particular needs of the user (Balmes et al., 2009), as it is our case.
Let us consider a vibrating structure. The first step in order to carry out its finite element analysis consists in discretizing the structure. The mesh nodes, elements which join the nodes, material properties, element properties and any additional information such as boundary conditions, loads, etc., are stored in a *struct* class variable which defines the model of the multi degree of freedom vibrating structure which is governed by the next set of equations of motion

$$M\ddot{x} + C\dot{x} + Kx = f \tag{1}$$

where M, K and C are the mass, the stiffness and the damping matrices, respectively, x the vector of displacements of all the degrees of freedom of the structure and f a general vector of forces. Matrices M and K arise from the finite element analysis of the structure whereas C is usually obtained under the hypothesis of Rayleigh damping.

Both in the state space approach and in the transfer function approach, working with physical coordinates is not practical because of the usually high number of degrees of freedom. Thus, a change of variables into modal coordinates is suggested. In addition, in this way the problem can be restricted to the bandwidth of interest, neglecting the high frequency dynamics of the system. After the computation of modes via finite element analysis one obtains the natural frequencies of the modes and the matrix of the mode shapes Φ. After performing the change of variables $x = \Phi z$, where z is the vector of modal coordinates and Φ is normalized in such a way that the modal mass matrix is the identity matrix, the governing equations in modal coordinates read

$$\ddot{z} + 2\xi\Omega\dot{z} + \Omega^2 z = \Phi^{\mathrm{T}} f \tag{2}$$

where Ω and ξ define the diagonal matrices of natural frequencies and damping factors.

Finally, the software also provides several commands related to the definition of the actuators and sensors or, strictly speaking, the input force influence vector L_u and the sensor influence vector L_y, both of them required for the definition of either the transfer function or the state space matrices.

The finite element models which are obtained with this toolbox can be used to solve both static and dynamic problems, the results of which being ready to be plotted or post-processed under any other MATLAB toolbox. In this work, we blended these commands which those corresponding to the *Control System Toolbox*.

2.2 Active damping and the Control System Toolbox

The *Control System Toolbox* provides a collection of matrix-based functions, usually expressed as M-files, for the topic of control engineering. The control systems may be modelled as transfer functions, zero-pole-gain or even using the state space approach, which allows the designer to apply classic techniques as well as modern techniques. However, it is not our purpose to study thoroughly this topic, so we will concentrate on briefly reviewing, as an example, one of the most common active control techniques of structures (Preumont, 2002): active damping with collocated piezoelectric pairs via Positive Position Feedback, which, in the end, will be the case study discussed in Section 5.

The transfer function of the multi degree of freedom vibrating structure obtained via modal expansion reads

$$G(\omega) = \frac{X(\omega)}{F(\omega)} = \sum_{n=1}^{\infty} \frac{\Phi_n \Phi_n^{\mathrm{T}}}{\omega_n^2 + 2\mathrm{j}\omega\xi_n\omega_n - \omega^2} \tag{3}$$

where Φ_n, ω_n and ξ_n are the mode shape, the natural frequency and the damping ratio for the n^{th} mode of vibration, respectively. The same model based upon the first N modes of vibration leads to the expression

$$\hat{G}(\omega) \approx \sum_{n=1}^{N} \frac{\Phi_n \Phi_n^{\mathrm{T}}}{\omega_n^2 + 2\mathrm{j}\omega\xi_n\omega_n - \omega^2} \tag{4}$$

where the residual term due to the contributions of the truncated nodes has been neglected.

Combining the equations for the surface-bonded actuator and sensor (Dosch et al., 1992; Sims et al., 2005) and Equation 4 which governs the structural dynamics, we obtain the frequency response function between the voltage V_a applied to the actuator and the voltage V_s at the output of the charge amplifier of the sensor

$$H(\omega) = \frac{V_s(\omega)}{V_a(\omega)} = k_a k_s \sum_{n=1}^{N} \frac{L_u^{\mathrm{T}} \Phi_n \Phi_n^{\mathrm{T}} L_y}{\omega_n^2 + 2\mathrm{j}\omega\xi_n\omega_n - \omega^2} \tag{5}$$

where k_s and k_a are the sensor gain and the actuator gain, respectively.
As far as the Positive Position Feedback scheme is concerned, it is appropriate for a structure equipped with strain actuators and sensors (Preumont, 2002). The objective of a Positive Position Feedback controller is to use a second order filter to improve the roll-off of the control system, allowing high frequency gain stabilization. The implementation of this control system is given by the following equations

$$\begin{gathered} M\ddot{x} + C\dot{x} + Kx = L_u u \\ y = L_u^T x \\ \ddot{v} + \beta_f \dot{v} + \Omega_f^2 v = y \\ u = -Gv \end{gathered} \tag{6}$$

where u is the control force acting on the structure through the influence vector L_u, y is the difference of slope between the ends of the sensor, $G = \mathrm{diag}(g_i)$ is the positive gain matrix, and v is the output of the second order filter defined by the matrices $\beta_f = \mathrm{diag}(2\xi_f\omega_f)$ and $\Omega_f^2 = \mathrm{diag}(\omega_f^2)$. Note that the fact that the same localization vector (L_u) rules the actuator and sensors is a consequence of collocation.
The transfer function of the controller (a second order low pass filter) can be written as follows

$$gD(w) = g\frac{\omega_f^2}{\omega_f^2 + 2\mathrm{j}\omega\xi_f\omega_f - \omega^2} \tag{7}$$

This compensator has both advantages and drawbacks. On the one hand, the open-loop transfer function has a roll-off of −40dB/decade, larger in comparison to Direct Velocity Feedback (−20dB/decade), which may decrease the risk of destabilizing high frequency dynamics. On the other hand, there is a stability limit which is reached when the open-loop static gain is equal to 1, this fact being independent of the damping (Preumont, 2002).
Once the open loop transfer functions of the plant and the regulator are defined, they can be used in the *Control System Toolbox* as *transfer function* class variables. Then, the desired active damping can be obtained by selecting the proper gain g of the regulator. In order to do this, several techniques can be used such as Bode Diagram Design, Root Locus Design, Nichols Plot design, etc.
Finally, the *Control System Toolbox* also provides valuable functions in order to analyze the stability and performance in terms of stability margins, time responses or the position of the closed loop poles and zeros in the pole-zero map. They will be used in our Fuzzy Finite Element methodology as fuzzy outputs in order to analyze the behaviour of the uncertain control system.

3. Uncertainty and fuzzy arithmetic

In the literature, the use of the term uncertainty is slightly ambiguous. In order to establish the nomenclature for the ensuing description, the reader is referred to the terminology proposed in (Oberkampf et al., 2004).

On the one hand, we will use the term aleatory uncertainty to describe the inherent variation associated with the physical system or the environment under consideration. These sources of uncertainty can be represented as randomly distributed quantities, usually in terms of probability distribution functions. Typical examples of this kind of uncertainty are manufacturing tolerances, environmental effects, properties of non-uniform materials or any other identifiable disturbances. On the other hand, epistemic uncertainty derives from some level of ignorance of the system or the environment in any activity of the modelling process. In this case, its definition stresses the lack of knowledge as the origin of this uncertainty. Thus, unlike aleatory uncertainty, a mathematical representation of epistemic uncertainty has proven to be much more of a challenge since it is subjective and based on some expert opinion to a certain extent. Typical examples of epistemic uncertainties are some models for boundary conditions, simplified models for joints, models for material damping or unpredictable model changes due to ageing, loading, etcetera.

Uncertainty propagation in dynamic systems has been a very frequent topic in the literature, which has been tackled from two different points of view (De Gersem et al., 2005): *probabilistic* and *possibilistic* approaches. Probabilistic approaches, such as the well-known Monte Carlo Simulation method, investigate the influence of aleatory uncertainties, this method being particularly suitable for finite element models with certain uncertainties. Possibilistic approaches are complementary to probabilistic approaches, since not only are they able to describe problems of aleatory uncertainties to a certain extent, yet without statistical interpretation, but they can also handle problems where incomplete information (epistemic uncertainty) is available. We consider that computationally expensive "brute-force" probabilistic methods are not worthwhile in most cases, where an intuitive worst-case scenario provides enough information. Thus, a possibilistic method in terms of fuzzy arithmetic will be the tool that we develop and propose in order to simulate and analyze uncertainty propagation in control systems.

Fuzzy arithmetic has been used by various researchers as a technique for propagating uncertainty or variability through complex engineering models. The origins of this approach can be found in the theory of Fuzzy Sets (Zadeh, 1965). In contrast to the classical set theory, the elements of a fuzzy set are assigned a degree of membership to the set, which is referred to as the membership level μ. The core of the set is defined as the subset of elements for which $\mu = 1$, whilst the support is the subset for which $\mu > 0$. A fuzzy number is a fuzzy set that is convex and normal, and whose membership function is piecewise continuous.

The issue of performing mathematical operations on fuzzy numbers instead of on conventional numbers can be a challenge rather than a trivial problem. In fact, the Fuzzy Arithmetic processes are usually performed by decomposing the fuzzy numbers into a number of intervals given by the α-cuts at the α-levels μ_i, $(i = 0, 1, \ldots, m)$ with

$$\mu_i = \frac{i}{m} \tag{8}$$

Then, a interval analysis via interval arithmetic can be carried out separately to each membership level μ_i.

For monotonic problems this process may be trivial, since the maxima or minima of the inputs will lead to the maxima or minima for the output. Conversely, for non-monotonic problems this is no longer the case (Sims et al., 2010). In addition, the Standard Fuzzy Arithmetic may become problematic and lead to different results for the same problem depending on the form in which the solution procedure is applied, as can be seen in the examples gathered in (Hanss, 2002). In short, the application of Standard Fuzzy Arithmetic usually leads to overestimate results to a lesser or greater extent and alternative methods must be employed.

The Transformation Method, introduced in (Hanss, 2002), represents a special implementation of Fuzzy Arithmetic that avoids the undesired overestimation which usually arises when Fuzzy Arithmetic is reduced to interval computation. A later efficient implementation of this method was carried out in (Klimke, 2003), taking advantage of the matrix computing environment of MATLAB and providing enhanced features such as fast processing of discretized fuzzy numbers through multi-dimensional arrays, elimination of recurring permutations, automatic decomposition of models, treatment of single occurrences of variables through interval arithmetic and a monotonicity test based on automatic differentiation. Nevertheless, we have to note that the fuzzy parameters are restricted to convex fuzzy numbers when using the Transformation Method.

Specially relevant for an efficient implementation of the Transformation Method is the elimination of recurring permutations. The decomposition scheme of this method sometimes produces recurring points which in the end generate recurring permutations. By removing these recurring permutations from the evaluation procedure one could save a very valuable computation time. Indeed, we could even try to reuse as many points for different α-cuts as possible by selecting only inner points which have already occurred in a higher-level α-cut. For symmetric triangular membership functions, the obtained discretization is identical to the original formulation, but for other arbitrarily shaped membership functions, the distribution of the points is less regular, but of similar density (see Figure 1). However, for the same number of α-cuts, the less regular distribution of the inner points results in less accurate results compared to the original formulation.

The number of permutations r by using this efficient implementation of the Transformation Method is given by the following equation

$$r = m^n + (m+1)^n \tag{9}$$

where $m+1$ is the number of α-cuts and n is the number of fuzzy parameters. As proven in (Klimke, 2003), the new scheme provides a valuable improvement in terms of computational cost unless the number of α-cuts is too large compared to the number of fuzzy inputs. If that was the case, one could disregard the removing of recurring permutations in favour of more accurate results.

4. Fuzzy Finite Element procedure in MATLAB

The fuzzy theory has led to the development of Fuzzy Finite Element methods as that described in (De Gersem et al., 2005; De Munck et al., 2008; Moens & Vandepitte, 2005) for analyzing the propagation of uncertainties in the Frequency Response Functions of a dynamic system. When using this methodology, the uncertainties in the model parameters can be taken into account by fuzzy numbers with their shape obtained from experimental data or assumed from expert knowledge. Then, the numerical procedure for the implementation of these Fuzzy Finite Element methods consists of a sequence of Interval Finite Element procedures based on a general strategy for computing the fuzzy result from operations on fuzzy operands, i.e. the α-sublevel or α-cuts technique.

Figure 1 clarifies this procedure. This technique subdivides the membership range into a number of α-cuts. At each level, the intersection with the membership function of the input uncertainties results in an interval. Based on these input intervals for all uncertain parameters, an interval analysis can be performed obtaining the interval of the output variables at the considered α-cut. Strictly speaking, the interval analysis for a specific α-cut consists in running all the finite elements calculations corresponding to all the possible permutations among input variables at said level and choosing the lower and upper bound of the required output

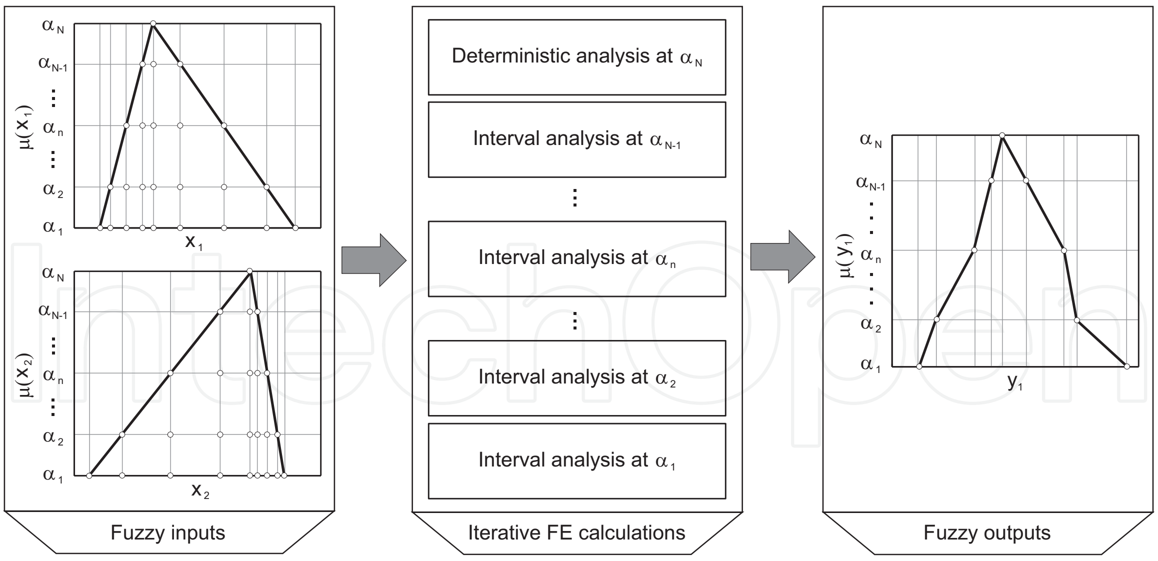
α_N
α_{N-1}
α_n
α_2
α_1
$\mu(x_1)$
x_1
$\mu(x_2)$
x_2
Fuzzy inputs
Deterministic analysis at α_N
Interval analysis at α_{N-1}
Interval analysis at α_n
Interval analysis at α_2
Interval analysis at α_1
Iterative FE calculations
$\mu(y_1)$
y_1
Fuzzy outputs

Fig. 1. Scheme of the numerical procedure to perform a Fuzzy Finite Element analysis using four α-cuts.

variables. Note that if $\mu = 1$, the interval analysis degenerates into the deterministic analysis. Finally, the membership functions of the required output variables are assembled from the interval results at all sublevels.

The Fuzzy Finite Element analysis is based on the results of a previous deterministic design of the control system. The control engineer must set the control parameters and gains of the regulator in order to accomplish all the features required for the active damping in terms of both stability and performance. After this process, which is usually easier and faster than other complex control techniques such as those derived from H_∞ control, the Fuzzy Finite Element analysis will simulate the behaviour of the control system when some parameters are considered uncertain. This method is not a robust designing technique but a possibilistic method which tests whether the deterministic design is robust enough to keep the closed loop behaviour of the structure inside the allowable bounds for its stability and performance features.

The set of calculations which are carried out at each permutation of input values is compound of these steps:

Step 1: Finite Element calculations via the *OpenFEM Toolbox*

Step 2: Closed loop simulation of the control system via the *Control System Toolbox*

Step 3: Analysis of stability and performance via fuzzy variables such as stability margins or active damping.

The method chosen for carrying out the interval fuzzy arithmetics was the efficient implementation of the Transformation Method proposed in (Klimke, 2003). Nevertheless, we have included some modifications which will be introduced in the ensuing.

The first modification lies in the fact that Klimke's MATLAB script is made under the assumption that the fuzzy output is the result of evaluating an analytical expression. In our case, we need to replace the evaluation of such analytical function (*inline* class) with a list of MATLAB commands which develops the aforementioned steps. This also allows us to include the possibility of achieving several fuzzy outputs instead of only one. The second modification was not necessary but it endows the method with enhance capabilities. It consists on using *cell* class arrays instead of *double* class arrays to define the fuzzy outputs, so an extra field may be included where information about each iteration is stored (for instance, the exact value of the fuzzy inputs at each permutation). Figures 2 and 3 show the MATLAB implementation of the efficient Transformation Method after modifying the Klimke's implementation. Note that the MATLAB implementation of the external function in Figure 3 only shows an example corresponding to the evaluation of a test function. In our case, this function would contain all the required commands which carry out the three foregoing calculating steps.

In order to prove the validity of our modified version of the Klimke's Transformation Method, a comparative fuzzy analysis was run. We took as a reference one example found in (Klimke, 2003) where the test function $f(x_1, x_2) = \cos(\pi x_1)x_2$ is subjected to the triangular fuzzy inputs

```
function [fzr]=gtrmrecur_modif(varargin)
% Angel L. Morales, 2011.
% FZR=GTRMRECUR_MODIF(FZ1,FZ2,...,FZN) calls the general transformation
% method implemented by Klimke in 2003 with extended capabilities.
% The function allows N fuzzy numbers in alpha-cut representation.
% An accompanying function called GTRMRECUR_EXTERNAL_FCN is required.
fzin=varargin;
n = nargin;
m = size(fzin{1},1) - 1;
indices.subs = {};
indices.type = ’()’;
fzr = cell(m+1,2);
for k = 1:2
t = (floor((m+2-k)/2)+rem(m+2-k,2))*2-2+k;
for i = 1:n
c = reshape(fzin{i}(end+1-k:-2:1,:)’,t+2-k,1);
c = c(3-k:end);
repvec = t*ones(1,n);
repvec(i) = 1;
x{i} = repmat(shiftdim(c,1-i),repvec);
end
s = k;
z_cell_matrix=gtrmrecur_external_fcn(x);
for j = m+2-k:-2:1
indices.subs(1:n) = {1:s};
w = reshape(subsref(z_cell_matrix,indices),s^n,1);
wc = zeros(size(w));
for cont=1:length(wc) wc(cont)=w{cont}{2}; end;
[none,pos_min]=min(wc); fzr{j,1} = w{pos_min};
[none,pos_max]=max(wc); fzr{j,2} = w{pos_max};
s = s + 2;
end
end
for j = m:-1:1
if fzr{j+1,1}{2}>=fzr{j,1}{2}; fzr{j,1}=fzr{j,1}; else fzr{j,1}=fzr{j+1,1}; end
if fzr{j+1,2}{2}>=fzr{j,2}{2}; fzr{j,2}=fzr{j+1,2}; else fzr{j,2}=fzr{j,2}; end
end
```

Fig. 2. MATLAB implementation of the efficient Transformation Method after modifying Klimke's script.

```
function [z_cell_matrix]=gtrmrecur_external_fcn(x)
% Angel L. Morales, 2011.
% Z_CELL_MATRIX=GTRMRECUR_EXTERNAL_FCN(FZ1,FZ2,...,FZN) calls the function
% which executes the commands in order to obtain the fuzzy output.
% It is called during the execution of FZR=GTRMRECUR_MODIF(FZ1,FZ2,...,FZN)
func = inline(’cos(pi.*x1).*x2’);
x_vector=cell(size(x));
for cont=1:length(x)
    x_vector{cont} = reshape(x{cont},numel(x{cont}),1);
end
iterations=length(x_vector{1});
z_cell=cell(1,2); z_cell_array=cell(1,iterations);
for iteration=1:iterations
    x1=double(x_vector{1}(iteration));
    x2=double(x_vector{2}(iteration));
    z_cell{1}=[x1 x2];
    z_cell{2}=feval(func,x_vector{1}(iteration),x_vector{2}(iteration));
    z_cell_array{iteration}=z_cell;
end
z_cell_matrix=reshape(z_cell_array,size(x{1}));
```

Fig. 3. Example of the MATLAB function called by the modified Transformation Method which contains the commands for the obtention of the fuzzy output

$\tilde{p}_1 = \langle 2.5, 2.5, 2.5 \rangle_{\text{TFN}}$ and $\tilde{p}_2 = \langle 3, 2, 2 \rangle_{\text{TFN}}$, in accordance to the nomenclature proposed in (Dubois & Prade, 1980). The fuzzy output obtained with the modified version clearly matched the results obtained with the original version. Nevertheless, an unavoidable loss of computational efficiency arose.

Our modified version results more time-consuming because of the different way in which the simulated function is handled, as it can be seen in Table 1. In the original version, the *inline* class analytical expression can be executed only once with the array containing all the required permutations of fuzzy inputs at a time. In the modified version, the expression is evaluated separately at each permutation as it would be the case when a control system is simulated with several MATLAB commands. The elapsed time during these operations is called t_{fcn}. As far as the obtention of the fuzzy output as a *cell* array instead of as a *double* array, this fact does not add any noticeable increase of computational cost but it does lead to heavier variables and higher memory requirements. The elapsed time during these operations is called t_{max}. In all the cases, the loss of efficiency is acceptable with regard to the benefits they provide, specially considering that the elapsed time in any finite element simulation may be on the order of seconds and it has much more restrictive memory requirements.

5. A case study of uncertainty propagation

Consider an aluminium plate in free conditions, the dimensions and material properties of which are detailed in Table 2. The finite element model of this plate was carried out by means of the *OpenFEM Toolbox* of MATLAB (Balmes et al., 2009). After defining the mesh, the matrices of mass (M) and stiffness (K) which define the undamped model were obtained. The eigenvalue problem can then be solved, the natural frequencies and mode shapes being shown in Figure 4. The damping matrix (C) is constructed considering a structural damping of $\xi_{\text{i}} = 0.6\%$ in all the modes.

The piezoelectric actuator and sensor, the properties of which are detailed in Table 3, are collocated in the centre of the plate. As described in (Dosch et al., 1992; Sims et al., 2005), the piezoelectric actuator and sensor may be modelled as a beam which provides bending moments at its ends and measures a voltage proportional to the difference of slope of its ends. Due to collocation, the localization vectors of actuator (L_u) and sensor (L_y) are identical and equal to

$$L_u^{\text{T}} = L_y^{\text{T}} = (0, \ldots, -1, 0, \ldots, 1, 0, \ldots, 0) \tag{10}$$

	Original	Modified
t_{tot} (s)	0.047	0.344
t_{tot} (%)	100.00	100.00
t_{fcn} (s)	0.016	0.313
t_{fcn} (%)	34.04	90.99
t_{max} (s)	0.000	0.000
t_{max} (%)	0.000	0.000
t_{oth} (s)	0.031	0.031
t_{oth} (%)	65.96	9.01
Weight (bytes)	336	8568

Table 1. Efficiency comparison between the original and the modified Klimke's implementation. The total time (t_{tot}), the time for evaluating the analytical expression (t_{fcn}), the time for calculating the maxima and minima of the output fuzzy number (t_{max}) and the time for other calculations (t_{oth}) are shown. The weight means the size in bytes of the output fuzzy variable.

Width (*mm*)	300.00
Length (*mm*)	500.00
Thickness (*mm*)	3.00
Young's modulus (GPa)	72.00
Poisson's coefficient	0.30
Density (kg/m^3)	2700.00
Structural damping (%)	0.60

Table 2. Geometries and material properties of the plate.

Width (*mm*)	40
Length (*mm*)	100
Thickness (*mm*)	0.250
d_{33} (m/V)	$0.30 \cdot 10^{-9}$
d_{31} (m/V)	$-0.15 \cdot 10^{-9}$
e_{31} (N/Vm)	-7.5
k_{33}	0.7
Young's modulus (GPa)	50
Maximum traction (MPa)	80
Maximum compression (MPa)	600
Maximum electric Field (V/mm)	2000
Density (kg/m^3)	7600
Maximum strain	Brittle
Maximum temperature (C)	80°C - 150°C

Table 3. Geometries and material properties of the piezoelectric (Preumont, 2002).

where the non-zero values are placed in the position corresponding to the active rotational degree of freedom of the nodes in which the piezoelectric is located.

As described in the Section 2.2, a Positive Position Feedback control strategy designed with the *Control System Toolbox* of MATLAB is to be used. Figure 5 presents the root locus of the control system after tuning the second order filter parameters (ξ_f and ω_f) so that the first flexible mode of vibration become actively damped. After the tuning process ($\xi_f = 0.9$ and $\omega_f = 900$ rad/s), the next step consists of selecting the control gain which most approaches

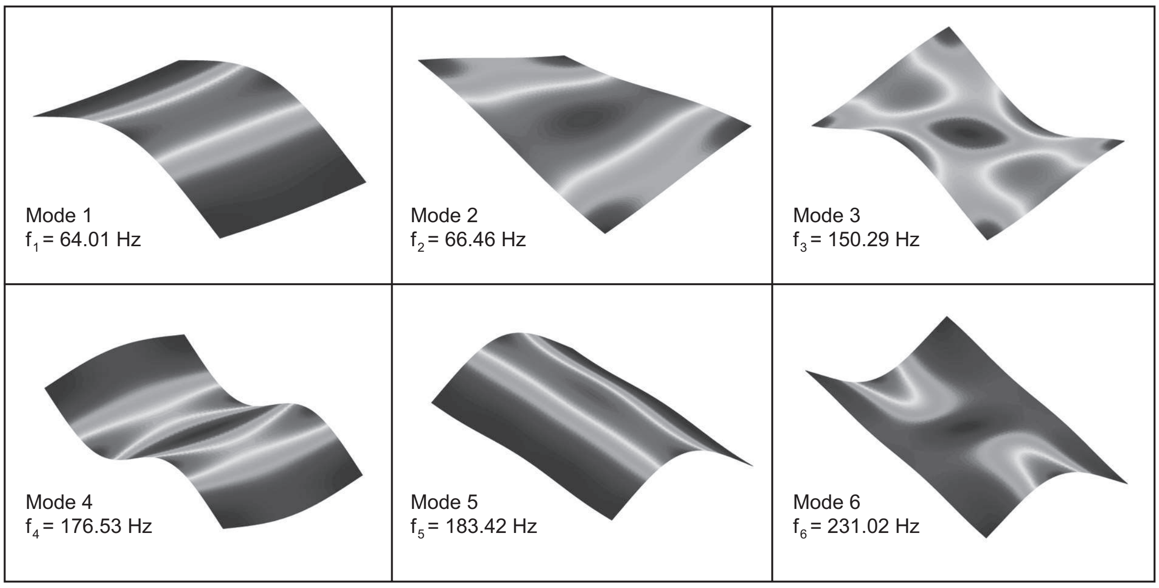

Fig. 4. Mode shapes and natural frequencies for the plate in free conditions.

the maximum damping of the first vibration mode ($\xi_{1,\max}$). In this particular case, a gain of $g = 0.200$ entails a selected damping coefficient of $\xi_{1,\text{sel}} = 88.30\%$. Figure 5 also shows the characteristic stability limit of the Positive Position Feedback control systems and the fact that only the first and the fifth modal shapes seem to be controllable in the view of the wide loops going from the poles to the zeros (Preumont, 2002).

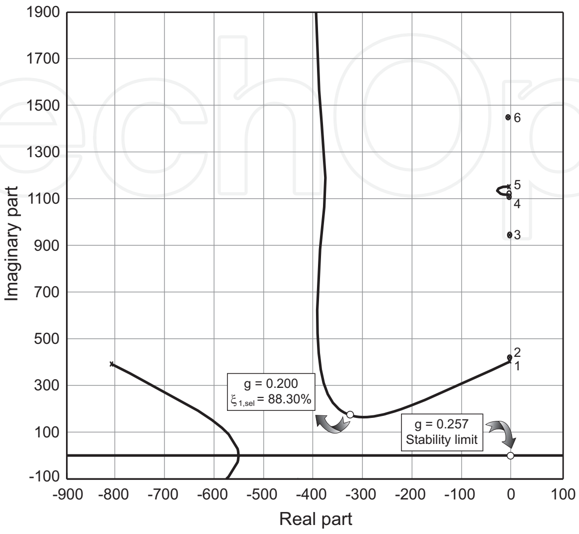

Fig. 5. Root locus of the control system after tuning the second order filter parameters to the first flexible mode of vibration ($\xi_{\text{f}} = 0.9$ and $\omega_{\text{f}} = 900$ rad/s).

In the proposed case study the collocated pair of actuator and sensor are subjected to uncertainty. More precisely, the position of the centre of both the actuator and sensor are considered to be uncertain parameters. Additionally, another source of uncertainty corresponding to the structural damping of the plate is included as this is known to be uncertain in practice. Thus, we have a total of five different fuzzy numbers: $\tilde{x}_a$ (position "x" of the centre of the actuator), $\tilde{y}_a$ (position "y" of the centre of the actuator), $\tilde{x}_s$ (position "x" of the centre of the sensor), $\tilde{y}_s$ (position "y" of the centre of the sensor) and $\tilde{\xi}$ (structural damping). The membership functions of these fuzzy parameters, assuming that they are both symmetric and triangular, can be seen in Figure 6.

After applying the Fuzzy Finite Element algorithm described in Section 4 one can obtain several fuzzy outputs which describe the stability and performance of the control system. In fact, it is possible to choose those variables which better fit the effects which one is interested in. In this particular case, the following are calculated:

$\tilde{G}$ Gain margin.

$\tilde{P}$ Phase margin.

$\tilde{\xi}_{1,\max}$ Maximum damping coefficient of the first mode.

$\tilde{\xi}_{1,\text{sel}}$ Selected damping coefficient of the first mode.

The gain margin $\tilde{G}$ is the increase in the system gain when the phase is -180 that will result in a marginally stable system with intersection of the $-1 + j0$ point on the Nyquist diagram. The phase margin $\tilde{P}$ is the amount phase shift of the system at unity magnitude that will result in a marginally stable system with intersection of the $-1 + j0$ point on the Nyquist diagram. The maximum damping coefficient $\tilde{\xi}_{1,\max}$ is the highest damping achievable in the first mode loop independently of the selected gain, whilst the selected damping coefficient $\tilde{\xi}_{1,\text{sel}}$ is the

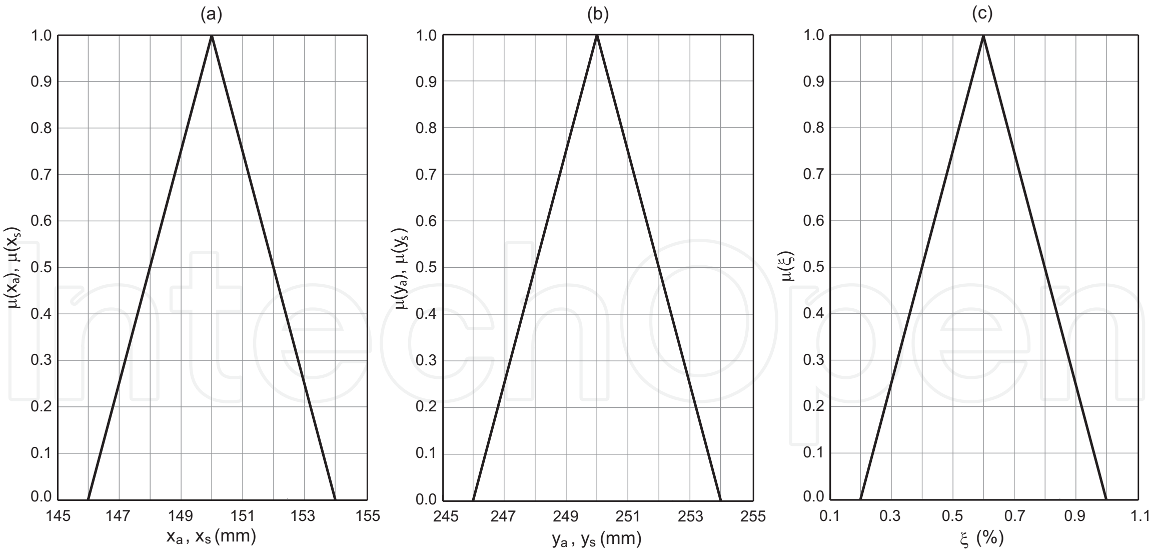

Fig. 6. Membership functions of the input fuzzy numbers: (a) coordinates of the centre of the actuator ($\tilde{x}_\text{a}$ and $\tilde{x}_\text{s}$), (b) coordinates of the centre of the sensor ($\tilde{y}_\text{a}$ and $\tilde{y}_\text{s}$) and (c) structural damping of the plate ($\tilde{\xi}$).

damping that the control systems provides at the value of gain selected in the deterministic case. Clearly, the gain and phase margins can be used to quantify controller stability, whilst the damping ratios serve to quantify performance and reliability.

The membership functions of these variables are shown in Figure 7. We observe that the uncertainty considered may lead to a both better and worse behaviour of the control system in comparison to the considered deterministic situation.

Another possibility for obtaining a global view of the whole control system lies in plotting its root locus for different α-sublevels or α-cuts. In particular, we show in Figure 8 the root locus for two different values of the membership. Although the fourth mode has not been

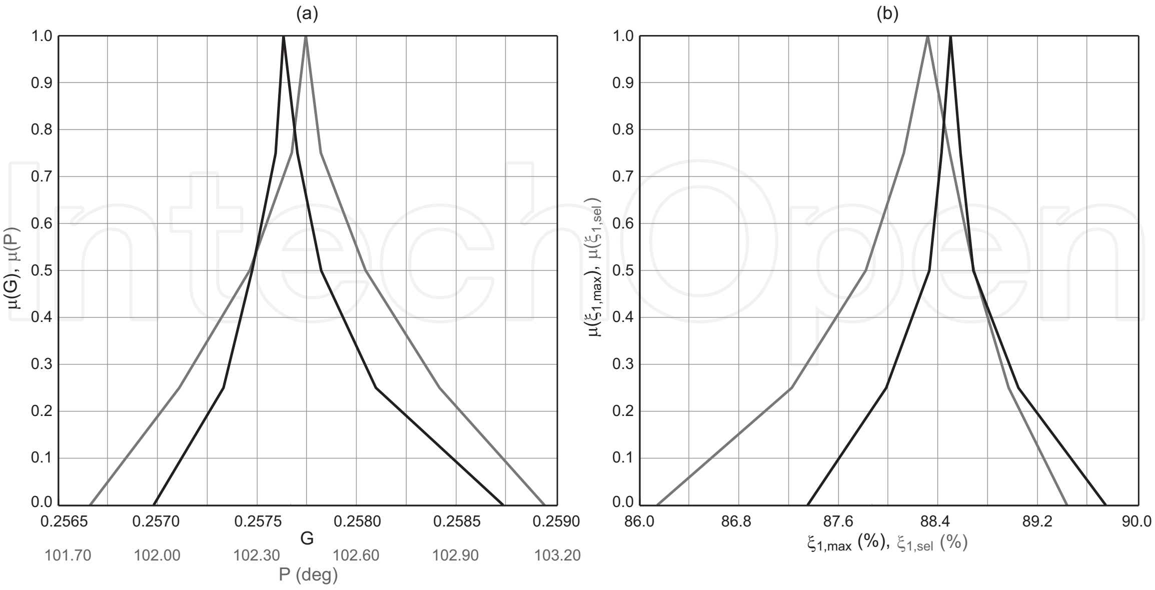

Fig. 7. Membership function of the output fuzzy numbers: (a) stability indicators ($\tilde{G}$ in black color and $\tilde{P}$ in gray color) and (b) performance indicators ($\tilde{\xi}_{1,\max}$ in black color and $\tilde{\xi}_{1,\text{sel}}$ in gray color).

specifically analyzed via fuzzy variables, we can note that its zero crosses the imaginary axis and becomes unstable for some values of the gain. Note that zeros are essentially related to mode shapes of the structure, which tend to be more sensitive to the parameter uncertainty than the natural frequencies. This fact warns us that a higher gain level may lead to instability of the fourth mode when actively damping the first, which can be detected easily by means of including new fuzzy variables in the analysis such as the maximum real part of the closed loop poles.

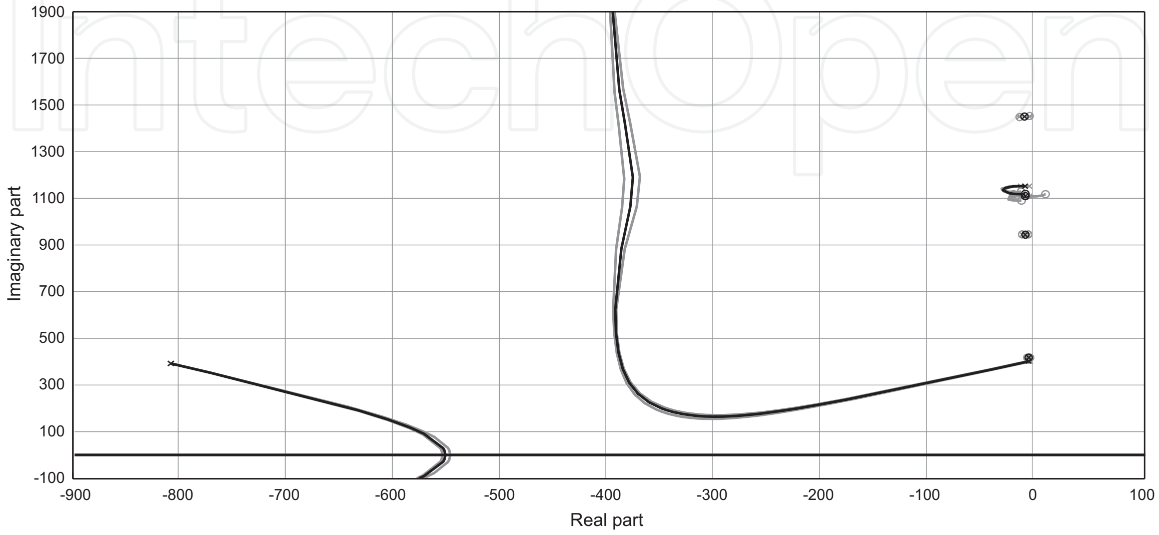

Fig. 8. Root locus of the control system for two different α-cuts: $\mu = 1$ (black) and $\mu = 0$ (grey).

6. Conclusion

In this chapter we have shown the suitability of the matrix computing environment MATLAB to solve control problems of structures via finite element analysis and root locus design and how these two issues have been blended and integrated into a Fuzzy Finite Element methodology for the analysis and simulation of uncertain control of structures.

The Fuzzy Finite Element procedure has been described. The approach relies on the α-cut technique which subdivides the membership range into a number of α-levels. The fuzzy arithmetic is based upon an efficient implementation in MATLAB code for the Transformation Method. This implementation has been modified and improved with some enhanced features. This methodology has been shown to be applicable to a realistic case study related to structural control using a Positive Position Feedback technique and piezoelectric actuators and sensors. In the light of the obtained results, the proposed procedure may prove useful to other analysis involving any other control technique or uncertainty source. In addition, the efficient implementation obtained due to MATLAB programming allows us to achieve high accuracy results together with low computational costs.

Finally, further research may be carried out in this topic. Firstly, a more efficient implementation of the Transformation Method can be found when using *cell* arrays instead of *double* arrays or when considering multiple fuzzy outputs. Secondly, the Transformation Method also allows a sort of sensitivity analysis which should be included in the MATLAB implementation in order to endow our method with extended capabilities. Thirdly, and in-depth comparison between this possibilistic methodology and the robust H_∞ optimal control theory must be carried out in order to identify more precisely the situations in which one method stand out from the other.

7. Acknowledgment

The authors are grateful for the support received from the “Ministerio de Educación” of the Spanish government through the “Programa Nacional de Movilidad de Recursos Humanos del Plan Nacional de I-D+i 2008-2011” and from the EPSRC through grant reference EP/D078601/1.

8. References

Balmes, E., Bianchi, J.-P. & Leclére, J.-M. (2009). *Structural Dynamics Toolbox for use with MATLAB: User's guide*.

De Gersem, H., Moens, D., Desmet, W. & Vandepitte, D. (2005). A fuzzy finite element procedure for the calculation of uncertain frequency response functions of damped structures: Part 2 - numerical case studies, *Journal of Sound and Vibration* 288: 463–486.

De Munck, M., Moens, D., Desmet, W. & Vandepitte, D. (2008). A response surface based optimisation algorithm for the calculation of fuzzy envelope frfs of models with uncertain properties, *Computers & Structures* 86: 1080–1092.

Dosch, J. J., Daniel, D. J. & Garcia, E. (1992). A self-sensing piezoelectric actuator for collocated control, *Journal of Intelligent Material Systems and Structures* 3: 166–185.

Dubois, D. & Prade, H. M. (1980). *Fuzzy sets and systems : theory and applications*, Mathematics in science and engineering, Academic Press, New York.

Gu, D.-W., Petkov, P. H. & Konstantinov, M. M. (2005). *Robust control design with MATLAB*, Springer, London.

Hanss, M. (2002). The transformation method for the simulation and analysis of systems with uncertain parameters, *Fuzzy Sets and Systems* 130: 277–289.

Klimke, A. (2003). An efficient implementation of the transformation method of fuzzy arithmetic, (IANS), extended preprint 2003/009, *Technical report*, University of Stuttgart.

Moens, D. & Vandepitte, D. (2005). A fuzzy finite element procedure for the calculation of uncertain frequency-response functions of damped structures: Part 1 - procedure, *Journal of Sound and Vibration* 288: 431–462.

Oberkampf, W. L., Helton, J. C., Joslyn, C. A., Wojtkiewicz, S. F. & Ferson, S. (2004). Challenge problems: uncertainty in system response given uncertain parameters, *Reliability Engineering & System Safety* 85: 11–19.

OpenFEM (2011).
URL: *http://support.sdtools.com/gf/project/openfem*

Preumont, A. (2002). *Vibration control of active structures: An introduction*, Kluwer Academic, Dordrecht.

Sims, N. D., Bayly, P. V. & Young, K. A. (2005). Piezoelectric sensors and actuators for milling tool stability lobes, *Journal of Sound and Vibration* 281: 743–762.

Sims, N. D., Manson, G. & Mann, B. (2010). Fuzzy stability analysis of regenerative chatter in milling, *Journal of Sound and Vibration* 329: 1025–1041.

Zadeh, L. A. (1965). Fuzzy sets, *Information and Control* 8: 338–353.

Zames, G. (1981). Feedback and optimal sensitivity: Model reference transformations, multiplicative seminorms and approximate inverses, *IEEE Transactions on Automatic Control* 26: 301–320.

Zames, G. & Francis, B. A. (1983). Feedback, minimax sensitivity and optimal robustness, *IEEE Transactions on Automatic Control* 28: 585–601.

MATLAB - A Ubiquitous Tool for the Practical Engineer
Edited by Prof. Clara Ionescu

ISBN 978-953-307-907-3
Hard cover, 564 pages
Publisher InTech
Published online 13, October, 2011
Published in print edition October, 2011

A well-known statement says that the PID controller is the â€œbread and butterâ€ of the control engineer. This is indeed true, from a scientific standpoint. However, nowadays, in the era of computer science, when the paper and pencil have been replaced by the keyboard and the display of computers, one may equally say that MATLAB is the â€œbreadâ€ in the above statement. MATLAB has became a de facto tool for the modern system engineer. This book is written for both engineering students, as well as for practicing engineers. The wide range of applications in which MATLAB is the working framework, shows that it is a powerful, comprehensive and easy-to-use environment for performing technical computations. The book includes various excellent applications in which MATLAB is employed: from pure algebraic computations to data acquisition in real-life experiments, from control strategies to image processing algorithms, from graphical user interface design for educational purposes to Simulink embedded systems.

Angel L. Morales, Jem A. Rongong and Neil D. Sims (2011). A Fuzzy Finite Element Method Programmed in MATLAB for the Analysis of Uncertain Control Systems of Structures, MATLAB - A Ubiquitous Tool for the Practical Engineer, Prof. Clara Ionescu (Ed.), ISBN: 978-953-307-907-3, InTech, Available from: http://www.intechopen.com/books/matlab-a-ubiquitous-tool-for-the-practical-engineer/a-fuzzy-finite-element-method-programmed-in-matlab-for-the-analysis-of-uncertain-control-systems-of-

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821