



**Politechnika
Śląska**

Synteza i analiza złożonych układów piezoelektrycznych
z uwzględnieniem metod klasycznych i nieklasycznych

Jacek Harazin

Wydział Mechaniczny Technologiczny
Politechnika Śląska

Promotorzy:

Dr hab. inż. Andrzej Wróbel, prof. PŚ. – promotor

Dr hab. inż. Marek Płaczek, prof. PŚ. – promotor pomocniczy

```
%=====
%Comparisons between used methods (whole script contained)
%=====
```

```
%Automatic chain fraction synthesizer algorithm
```

```
RF = [];
ARF = [];
Fs = [];
Fc = [];
Us = [];
Uc = [];
```

```
%Insert resonant frequencies here:
```

```
RF(1) = sym('23000');
RF(2) = sym('27000');
%RF(3) = sym('67000');
%RF(4) = sym('80000');
%RF(5) = sym('95000');
%RF(6) = sym('110000');
```

```
%Insert anti-resonant frequencies here (number of resonant frequencies - 1):
```

```
ARF(1) = sym('23000');
%ARF(2) = sym('62000');
%ARF(3) = sym('76000');
%ARF(4) = sym('90000');
%ARF(5) = sym('105000');
%ARF(6) = sym('110000');
```

```
%Conversion from Hertz to rad/s
```

```
RF = sym(round(RF*2*vpa(pi)))
```

```
RF = (138230 169646)
```

```
ARF = sym(round(ARF*2*vpa(pi)))
```

```
ARF = 144513
```

```
%Range setting for function type of solver
```

```
RangeStart = 20000;
RangeEnd = 29000;
```

```
%Range setting for array type of solver
```

```
precision = 1;
increment = round(RF(1)/100);
ranlen = double(2*increment+1)
```

```
ranlen = 2765
```

```
range = zeros(length(RF),ranlen);
for i=1:length(RF)
    range(i,:) = (RF(i)-increment):precision:(RF(i)+increment);
end
```

```
%Model forces (with sine function)
```

```
%Us(1) = sym('1');
%Us(2) = sym('1');
%Us(3) = sym('1');
%Us(4) = sym('1');
%Us(5) = sym('1');
%Us(6) = sym('1');
```

```
Fs(1) = sym('0');
Fs(2) = sym('0');
%Fs(3) = sym('0');
%Fs(4) = sym('0');
%Fs(5) = sym('0');
%Fs(6) = sym('0');
```

```
%Model forces (with cosine function)
```

```
%Uc(1) = sym('150');
%Uc(2) = sym('1');
%Uc(3) = sym('1');
%Uc(4) = sym('1');
%Uc(5) = sym('1');
%Uc(6) = sym('1');
```

```
Fc(1) = sym('0');
Fc(2) = sym('1');
%Fc(3) = sym('200');
%Fc(4) = sym('200');
%Fc(5) = sym('200');
```

```

%Fc(6) = sym('200');

%Scaling factor
H = 0.0100;

%Initial transfer function calculation
syms s Tn Td Rest Rn Rd div
Tn = 1;
Td = 1;
for i=1:length(RF)
    Tn = Tn*(s^2 + RF(i)^2);
end
for i=1:length(ARF)
    Td = Td*(s^2 + ARF(i)^2);
end
Td = Td*s/H;
F = H*Tn/Td

```

```

F =

$$\frac{(s^2 + 19107532900)(s^2 + 28779765316)}{10000s(s^2 + 20884007169)}$$


```

```

%Components extraction into a matrix form
Rn = [];
Rd = [];
springs = [];
masses = [];
Rn = coeffs(Tn, 'All');

for i=1:length(Rn)-1
    Rn = coeffs(Tn, 'All');
    Rd = coeffs(Td, 'All');

    div(i) = sym(Rn(1))/sym(Rd(1));
    Tn = poly2sym(Rn,s);
    Td = poly2sym(Rd,s);
    Rest = Tn - div(i)*s*Td;

    Tn = Td;
    Td = Rest;

    if mod(i,2)==0
        springs(i)=1/div(i);
    else
        masses(i)=div(i);
    end
end

%List of initial model components
springs(springs==0)=[]

```

```

springs = 1x2
1010 x
    0.0270    1.0587

```

```

masses(masses==0)=[]

```

```

masses = 1x2
    0.0100    0.5199

```

```

%List of derived damping components from Rayleigh equations
beta=sym(2/(min(RF,[], "all")+max(RF,[], "all")));
dampers = double(RF*beta);

%Model damping elements (set "no damping" with DampON=0)
DampON = 1;
Gchange = 0;
if DampON==0
    dampers = dampers;
elseif DampON==0
    dampers = 0.000000000001 + zeros(1,length(dampers));
elseif DampON==2
    if Gchange==1
        dampers = dampers/100*100;
    elseif Gchange==0
        dampers(1) = dampers(1)/100*20;
        dampers(2) = dampers(2)/100*100;
    end
end

dampers

```

```

dampers = 1x2

```

```

%===== (END OF SYNTHESIS) =====

%=====
%Matrix method calculations
%=====

%This algorithm calculates the resonant frequency peaks for a cascade model fixed on one side

M = masses;
K = springs;
B = dampers;

%Motion equation generator
syms Fun omega
X = sym('x%d%d',[3 length(M)]);

if length(M)==1
    Fun(i) = M(i)*X(3,i)+B(i)*X(2,i)+K(i)*X(1,i);
else
    for i=1:length(M)
        if i==1
            Fun(i) = M(i)*X(3,i)+B(i)*(X(2,i)-X(2,i+1))+K(i)*(X(1,i)-X(1,i+1));
        elseif i>1 && i<length(M)
            Fun(i) = M(i)*X(3,i)+B(i)*(X(2,i)-X(2,i+1))-B(i-1)*(X(2,i-1)-X(2,i))+K(i)*(X(1,i)-X(1,i+1))-K(i-1)*(X(1,i-1)-X(1,i));
        elseif i==length(M)
            Fun(i) = M(i)*X(3,i)+B(i)*X(2,i)-B(i-1)*(X(2,i-1)-X(2,i))+K(i)*X(1,i)-K(i-1)*(X(1,i-1)-X(1,i));
        end
    end
end

Y = sym('A%d%d',[length(M) 2]);
SizeX = size(X);

for i=1:SizeX(2)
    Z(1,i) = Y(i,1)*sin(omega)+Y(i,2)*cos(omega);
    Z(2,i) = Y(i,1)*cos(omega)*omega-Y(i,2)*sin(omega)*omega;
    Z(3,i) = -Y(i,1)*sin(omega)*omega^2-Y(i,2)*cos(omega)*omega^2;
end

Fun = subs(Fun,X,Z);

%Separation of sine and cosine coefficients
syms Funsine Funcosine

for i=1:length(Fun)
    Funsine(i)=subs(Fun(i),cos(omega),0);
    Sinecoeffs(i)=subs(Funsine(i),sin(omega),1);
end

for i=1:length(Fun)
    Funcosine(i)=subs(Fun(i),sin(omega),0);
    Cosinecoeffs(i)=subs(Funcosine(i),cos(omega),1);
end

Sinecoeffs=collect(Sinecoeffs,Y);
Cosinecoeffs=collect(Cosinecoeffs,Y);

%Final set of equations of motion
syms Eqn
for i=1:length(M)
    Eqn(2*i-1)=Sinecoeffs(i)==Fs(i);
    Eqn(2*i)=Cosinecoeffs(i)==Fc(i);
end

SizeY = size(Y);

for i=1:SizeY(1)
    vars(2*i-1) = Y(i,1);
    vars(2*i) = Y(i,2);
end

%Matrix solver
[Coeffs,F] = equationsToMatrix(Eqn,vars);
V = vars.';

V = linsolve(Coeffs,F);

%Amplitude calculator
syms MxA

for i=1:length(M)

```



```

MxA(i) = sym(sqrt(V(2*i-1)^2+V(2*i)^2));
end

%===== (END OF MATRIX METHOD) =====

%=====
%Structural number method of model analysis
%=====

%This algorithm calculates the resonant frequency peaks for a cascade model fixed on one side

%Graph edge generator
for i=1:length(M)
    S(i)=sym(-(M(i)*omega^2));
    C(i)=sym(-(M(i)*omega^2));

    for j=1:length(K)
        S(i+j)=sym(K(j));
        C(i+j)=sym(K(j));

        for l=1:length(B)
            S(i+j+l)=sym(-(B(l)*omega));
            C(i+j+l)=sym(B(l)*omega);
        end
    end
end

%Graph node generator
for i=1:length(M)
    if i==1
        vector{i}=[i length(M)+i length(M)+length(K)+i];
    else
        vector{i}=[i length(M)+i-1 length(M)+i length(M)+length(K)+i-1 length(M)+length(K)+i];
    end
end

%Graph structural number generator

%Combining all possible combinations of elements of each vector into one matrix
graph = combvec(vector{:});

%Eliminating columns with repeating numbers in two or more rows
graph=sortrows(graph.',1);
for i=size(graph,1):-1:1
    discard = graph(i,:);
    discard = unique(discard);
    if length(discard)<length(graph(i,:))
        graph(i,:)=[];
    end
end
graph = graph.';

%Eliminating pairs of repeated columns
graph=sort(graph,1);
graph=sortrows(graph.',1).';
[graphu, ia, ic] = unique(graph.', 'rows', 'stable');
counts = accumarray(ic,1);
maph = counts(ic);
graph = [graph.', maph];
for i=size(graph,1):-1:1
    if mod(graph(i,length(M)+1),2)==0
        graph(i,:)=[];
    end
end
graph(:,length(M)+1) = [];

%Resulting structural number of the graph
graph = unique(graph, 'rows', 'stable').';

%Structural number derivatives after their main branches (describing inertias)
for i=1:length(M)
    D{i}=[];
    for j=1:size(graph,1)
        for k=1:size(graph,2)
            if graph(j,k) == i
                D{i} = cat(2,D{i},graph(:,k));
            end
        end
    end
    lenD = size(D{i},2);
    D{i} = D{i}(~ismember(D{i},i));
    D{i} = reshape(D{i},length(M)-1,lenD);
end

```

```

%Simultaneity functions for derivatives
Simu={};
for i=1:length(D)
    for j=1:length(D)
        Simu{i,j}=intersect(D{i}.,D{j}.,'rows').';
    end
end

%Denominator of the sine graph structural equation
DnmS = sym(zeros(size(graph,2)));
for i=1:size(graph,2)
    for j=1:size(graph,1)
        if j==1
            DnmS(i) = S(graph(j,i));
        else
            DnmS(i) = DnmS(i).*S(graph(j,i));
        end
    end
end
DnmS = sum(DnmS);
end

%Denominator of the cosine graph structural equation
DnmC = sym(zeros(size(graph,2)));
for i=1:size(graph,2)
    for j=1:size(graph,1)
        if j==1
            DnmC(i) = C(graph(j,i));
        else
            DnmC(i) = DnmC(i).*C(graph(j,i));
        end
    end
end
DnmC = sum(DnmC);
end

%Nominator components for each derivative and simultaneity funtion for sine graph
NumS = sym(zeros(size(Simu,1),size(Simu,2)));
for i=1:size(Simu,1)
    for j=1:size(Simu,2)
        Simfunc = Simu{i,j};
        Expr = sym(0);
        for k=1:size(Simu{i,j},2)
            for l=1:size(Simu{i,j},1)
                if l==1
                    Expr(k) = S(Simfunc(l,k));
                else
                    Expr(k) = Expr(k).*S(Simfunc(l,k));
                end
            end
        end
        Expr = sum(Expr);
    end
    NumS(i,j) = Expr;
end
end

%Nominator components for each derivative and simultaneity funtion for cosine graph
NumC = sym(zeros(size(Simu,1),size(Simu,2)));
for i=1:size(Simu,1)
    for j=1:size(Simu,2)
        Simfunc = Simu{i,j};
        Expr = sym(0);
        for k=1:size(Simu{i,j},2)
            for l=1:size(Simu{i,j},1)
                if l==1
                    Expr(k) = C(Simfunc(l,k));
                else
                    Expr(k) = Expr(k).*C(Simfunc(l,k));
                end
            end
        end
        Expr = sum(Expr);
    end
    NumC(i,j) = Expr;
end
end

%Calculating final equations for both types of graphs
Dnm = sqrt(DnmS^2 + DnmC^2);
for i=1:size(NumS,1)
    for j=1:size(NumS,2)
        Num(i,j) = sqrt(NumS(i,j)^2 + NumC(i,j)^2);
    end
end
end

for i=1:length(Fs)
    F(i) = sqrt(Fs(i)^2 + Fc(i)^2);
end

```

```

end

%Amplitude calculator
syms StrA

for i=1:length(M)
    Expr = sym(0);
    for j=1:length(M)
        Expr(j) = Num(i,j)*F(j)/Dnm;
    end
    StrA(i) = sum(Expr);
end

%Finding extrema of both solutions
MxA_extr = [];
StrA_extr = [];

for i=1:length(M)
    dMxA(i) = diff(MxA(i));
    dStrA(i) = diff(StrA(i));

    MxA_temp = vpa(solve(dMxA(i)==0, omega));
    StrA_temp = vpa(solve(dStrA(i)==0, omega));
    MxA_temp = MxA_temp(MxA_temp>0);
    StrA_temp = StrA_temp(StrA_temp>0);
    MxA_temp = (MxA_temp(imag(MxA_temp)==0)).';
    StrA_temp = (StrA_temp(imag(StrA_temp)==0)).';

    MxA_extr{i} = MxA_temp;
    StrA_extr{i} = StrA_temp;
    MxA_temp = [];
    StrA_temp = [];
end

%Calculating amplitude values for extrema
for i=1:length(M);
    for j=1:length(MxA_extr{i})
        MxA_val{i}(j)=subs(MxA(i),omega,MxA_extr{i}(j));
    end
    for j=1:length(StrA_extr{i})
        StrA_val{i}(j)=subs(StrA(i),omega,StrA_extr{i}(j));
    end
end

%======(END OF STRUCTURAL METHOD)=====

%=====  

%Comparisons between both methods and results  

%=====  

for i=1:length(M)
    figure
    fplot(MxA(i), [RangeStart RangeEnd], 'b')
    hold on
    fplot(StrA(i), [RangeStart RangeEnd], '--r')
    plot(MxA_extr{i},MxA_val{i},'bo',StrA_extr{i},StrA_val{i},'rx')
    hold off
    xlabel('Frequency [rad/s]')
    ylabel('Amplitude [m]')
    title('Amplitudes with different methods')
    legend('Matrix method','Structural method')
    ylim([0 0.00001])
end

```

```
classdef Piezo_app < matlab.apps.AppBase
```

```
% Properties that correspond to app components
```

```
properties (Access = public)
```

```
UIFigure matlab.ui.Figure
TabGroup matlab.ui.container.TabGroup
SynthesismethodTab matlab.ui.container.Tab
GridLayout4 matlab.ui.container.GridLayout
ChoiceButton_2 matlab.ui.control.Button
ChoiceButton_1 matlab.ui.control.Button
PiezoelectricmodelsynthesizerLabel matlab.ui.control.Label
ChoosedesiredsynthesismethodLabel matlab.ui.control.Label
SetinitialparametersTab matlab.ui.container.Tab
GridLayout matlab.ui.container.GridLayout
XButton matlab.ui.control.Button
IndividualdampingratiosperstageCheckBox matlab.ui.control.CheckBox
NextstepButton matlab.ui.control.Button
CalculateButton matlab.ui.control.Button
UITable_3 matlab.ui.control.Table
ScalingfactorHEditField matlab.ui.control.NumericEditField
ScalingfactorHLabel matlab.ui.control.Label
Label matlab.ui.control.Label
GlobaldampingratioEditField matlab.ui.control.NumericEditField
DampingratioLabel matlab.ui.control.Label
FillinthesetofdesiredfrequenciesLabel matlab.ui.control.Label
MechanicalmodelparameterssynthesizerLabel matlab.ui.control.Label
UITable matlab.ui.control.Table
SetnumberofstagesEditField matlab.ui.control.NumericEditField
SetnumberofstagesLabel matlab.ui.control.Label
SelectMaterialsTab matlab.ui.container.Tab
GridLayout2 matlab.ui.container.GridLayout
XButton_2 matlab.ui.control.Button
NextstepButton_2 matlab.ui.control.Button
ClearstagesButton matlab.ui.control.Button
SetstageSpinner matlab.ui.control.Spinner
SetstageSpinnerLabel matlab.ui.control.Label
ApplytostageButton matlab.ui.control.Button
LoadFileButton matlab.ui.control.Button
UITable_7 matlab.ui.control.Table
ListofavailableexistingpiezoelectricmaterialsLabel matlab.ui.control.Label
SetplatedimensionsTab matlab.ui.container.Tab
GridLayout3 matlab.ui.container.GridLayout
XButton_3 matlab.ui.control.Button
RoundbaseCheckBox matlab.ui.control.CheckBox
Label_2 matlab.ui.control.Label
FillindesiredplatedimensionsLabel matlab.ui.control.Label
UITable_6 matlab.ui.control.Table
UITable_5 matlab.ui.control.Table
Analysemode1Button_3 matlab.ui.control.Button
CalculateButton_2 matlab.ui.control.Button
BacktobeginningButton matlab.ui.control.Button
PiezoelectricparameterssynthesizerLabel matlab.ui.control.Label
SelectMaterialsTab_2 matlab.ui.container.Tab
GridLayout2_2 matlab.ui.container.GridLayout
XButton_6 matlab.ui.control.Button
NextstepButton_4 matlab.ui.control.Button
ClearstagesButton_2 matlab.ui.control.Button
SetstageSpinner_2 matlab.ui.control.Spinner
SetstageSpinnerLabel_2 matlab.ui.control.Label
ApplytostageButton_2 matlab.ui.control.Button
LoadFileButton_2 matlab.ui.control.Button
UITable_12 matlab.ui.control.Table
ListofavailableexistingpiezoelectricmaterialsLabel_2 matlab.ui.control.Label
SetinitialparametersTab_2 matlab.ui.container.Tab
GridLayout_2 matlab.ui.container.GridLayout
Label_4 matlab.ui.control.Label
XButton_4 matlab.ui.control.Button
NextStepButton matlab.ui.control.Button
CalculateButton_3 matlab.ui.control.Button
UITable_9 matlab.ui.control.Table
ScalingfactorHEditField_2 matlab.ui.control.NumericEditField
ScalingfactorHLabel_2 matlab.ui.control.Label
FillinthesetofdesiredfrequenciesLabel_2 matlab.ui.control.Label
InitialparameterselectorLabel matlab.ui.control.Label
UITable_8 matlab.ui.control.Table
PlatecalculatorTab matlab.ui.container.Tab
GridLayout3_2 matlab.ui.container.GridLayout
SpecifycustomdimensionsButtonGroup matlab.ui.container.ButtonGroup
CustomthicknessButton matlab.ui.control.RadioButton
CustomsurfaceareaButton matlab.ui.control.RadioButton
SpecifyplatebasetypeButtonGroup matlab.ui.container.ButtonGroup
RectangularbaseButton matlab.ui.control.RadioButton
SquarebaseButton matlab.ui.control.RadioButton
RoundbaseButton matlab.ui.control.RadioButton
XButton_5 matlab.ui.control.Button
Label_3 matlab.ui.control.Label
UITable_11 matlab.ui.control.Table
UITable_10 matlab.ui.control.Table
Analysemode1Button matlab.ui.control.Button
CalculateButton_4 matlab.ui.control.Button
BacktobeginningButton_2 matlab.ui.control.Button
PiezoelectricparameterssynthesizerLabel_2 matlab.ui.control.Label
ModelanalyzerTab matlab.ui.container.Tab
GridLayout5 matlab.ui.container.GridLayout
OffsetButtonGroup_2 matlab.ui.container.ButtonGroup
cosButton_2 matlab.ui.control.RadioButton
sinButton_2 matlab.ui.control.RadioButton
OffsetButtonGroup matlab.ui.container.ButtonGroup
cosButton matlab.ui.control.RadioButton
sinButton matlab.ui.control.RadioButton
SetexternalvoltageButton matlab.ui.control.Button
SetexternalforceButton matlab.ui.control.Button
```

```

AnalyseButton          matlab.ui.control.Button
ProgressinfoLabel     matlab.ui.control.Label
Image_2               matlab.ui.control.Image
SetfreqrangeButton    matlab.ui.control.Button
SetaccuracyButton     matlab.ui.control.Button
ShowgraphsButton      matlab.ui.control.Button
UITable2              matlab.ui.control.Table
BackButton            matlab.ui.control.Button
XButton_7             matlab.ui.control.Button
Image                 matlab.ui.control.Image
ModelanalyzerLabel    matlab.ui.control.Label
CancelAnalysisTab     matlab.ui.container.Tab
CancelButton_2        matlab.ui.control.Button
ProceedButton_2       matlab.ui.control.Button
RestartthesynthesisLabel_3 matlab.ui.control.Label
CanceltheanalysisLabel matlab.ui.control.Label
ResetTab              matlab.ui.container.Tab
CancelButton           matlab.ui.control.Button
ProceedButton         matlab.ui.control.Button
RestartthesynthesisLabel_2 matlab.ui.control.Label
CancelthesynthesisLabel matlab.ui.control.Label
end

properties (Access = private)
    No_stages          % Number of stages for a new piezoelectric system
    Freq_array         % Array of resonant and anti resonant frequencies
    Damping_percent    % Percentage of maximum damping ratio
    H_Factor           % H scaling factor setting
    Init_synth_params  % Synthesized parameters of an initial mechanical system
    Plate_dims         % Array with dimensions of each plate
    Piezo_params       % A calculated set of piezoelectric parameters for each plate
    File_available_materials % Table of available piezoelectric materials loaded from excel file
    Chosen_mat         % Material selected from the list of available materials, ready to be applied to stages in the model
    Sel_stage = 1      % Stage selected for existing material application
    Chosen_mat_stages % list of chosen materials for each stage of the model
    Individual_damping = 0 % Toggle for individual damping rations for the model
    Round_plates = 0  % Toggle between the choice of rectangular or round plates
    Recent_tab        % Contains recently opened tab
    Base_plate_type   % Chosen shape of the piezoelectric plate base for synthesis with plate dimension goal
    Selected_dim      % Chosen dimension of plates to be synthesized by the program
    Calculated_dims   % Dimensions of piezoelectric plate calculated from the process
    Piezo_names       % Table storing piezoelectric material names taken from the catalogues

    Analysis_input = [] % Table of piezoelectric parameters extracted from both synthesis processes
    Analysis_shape = 0 % Indicator of a piezoelectric plate shape chosen in the synthesis process
    Analysis_prev_synth_choice = 0 % Indicator of what synthesis method was chosen
    Analysis_range = [] % Analysis frequency range
    Analysis_range_set = 0 % Range has been set flag
    Analysis_accuracies = 0 % Accuracies of the graph for regions where resonance occurs and between them
    Analysis_accuracies_set = 0 % Accuracies set flag
    Analysis_ext_force = 0 % External force set up in the analysis
    Analysis_ext_force_set = 0 % External force set up indicator
    Analysis_ext_force_sincos = 0 % External force offset
    Analysis_ext_voltage = 0 % External voltage set up in the analysis
    Analysis_ext_voltage_set = 0 % External voltage set up indicator
    Analysis_ext_voltage_sincos = 0 % External voltage offset
    Analysis_results = 0 % Aplitudes and extrema calculated from the analysis
    Analysis_results_figures = 0 % Figures generated for the analysis

    Temp_rho_ce

end

% Callbacks that handle component events
methods (Access = private)

% Code that executes after component creation
function startupFcn(app)
    app.ResetTab.Parent = [];
    app.SetinitialparametersTab.Parent = [];
    app.SelectMaterialsTab.Parent = [];
    app.SetplatedimensionsTab.Parent = [];
    app.SelectMaterialsTab_2.Parent = [];
    app.SetinitialparametersTab_2.Parent = [];
    app.PlatecalculatorTab.Parent = [];
    app.ModelanalyzerTab.Parent = [];
    app.CancelAnalysisTab.Parent = [];
end

% Button pushed function: ChoiceButton_1
function ChoiceButton_1Pushed(app, event)
    app.SetinitialparametersTab.Parent = app.TabGroup;
    app.SelectMaterialsTab.Parent = app.TabGroup;
    app.SetplatedimensionsTab.Parent = app.TabGroup;
    app.TabGroup.SelectedTab = app.SetinitialparametersTab;
    app.SynthesismethodTab.Parent = [];
    app.No_stages = 1;
    app.H_Factor = 0;
    app.Damping_percent = 0;
    app.SetnumberofstagesEditField.Value = app.No_stages;
    app.ScalingfactorHEditField.Value = app.H_Factor;
    app.GlobaldampingratioEditField.Value = app.Damping_percent;
end

% Button pushed function: ChoiceButton_2
function ChoiceButton_2Pushed(app, event)
    app.SelectMaterialsTab_2.Parent = app.TabGroup;
    app.SetinitialparametersTab_2.Parent = app.TabGroup;
    app.PlatecalculatorTab.Parent = app.TabGroup;
    app.TabGroup.SelectedTab = app.SelectMaterialsTab_2;
    app.SynthesismethodTab.Parent = [];
    app.No_stages = 0;
    app.H_Factor = 0;
    app.Sel_stage = 1;

```

```

app.SetstageSpinner.Value = app.Sel_stage;
end

% Value changed function: SetnumberofstagesEditField
function SetnumberofstagesEditFieldValueChanged(app, event)
app.No_stages = app.SetnumberofstagesEditField.Value;
if app.Individual_damping == 0
    app.Freq_array = zeros(app.No_stages, 2);
    app.UITable.ColumnName = ["Resonant freq [Hz]" "Anti-resonant freq [Hz]"];
    app.UITable.ColumnEditable = true(1,2);
else
    app.Freq_array = zeros(app.No_stages, 3);
    app.UITable.ColumnName = ["Resonant freq [Hz]" "Anti-resonant freq [Hz]" "Damping ratio [%]"];
    app.UITable.ColumnEditable = true(1,3);
end
app.Plate_dims = zeros(app.No_stages, 2);
app.UITable.Data = app.Freq_array;
app.UITable.RowName = 'numbered';
app.UITable_6.RowName = 'numbered';
if app.Round_plates == 0
    app.Plate_dims = zeros(app.No_stages,3);
    app.UITable_6.Data = app.Plate_dims;
    app.UITable_6.ColumnName = ["Thickness [mm]" "Width [mm]" "Length [mm]"];
    app.UITable_6.ColumnEditable = true(1,3);
    app.Analysis_shape = 3;
elseif app.Round_plates == 1
    app.Plate_dims = zeros(app.No_stages,2);
    app.UITable_6.Data = app.Plate_dims;
    app.UITable_6.ColumnName = ["Thickness [mm]" "Diameter [mm]"];
    app.UITable_6.ColumnEditable = true(1,2);
    app.Analysis_shape = 2;
end
end

% Value changed function: ScalingfactorHEditField,
% ScalingfactorHEditField_2
function ScalingfactorHEditFieldValueChanged(app, event)
if app.TabGroup.SelectedTab == app.SetinitialparametersTab;
    app.H_Factor = app.ScalingfactorHEditField.Value;
elseif app.TabGroup.SelectedTab == app.SetinitialparametersTab_2;
    app.H_Factor = app.ScalingfactorHEditField_2.Value;
end
end

% Value changed function: IndividualdampingratiosperstageCheckBox
function IndividualdampingratiosperstageCheckBoxValueChanged(app, event)
app.Individual_damping = app.IndividualdampingratiosperstageCheckBox.Value;
if app.Individual_damping == 1
    app.GlobaldampingratioEditField.Editable = 'off';
    app.GlobaldampingratioEditField.Enable = 'off';
else
    app.GlobaldampingratioEditField.Editable = 'on';
    app.GlobaldampingratioEditField.Enable = 'on';
end
if isempty(app.Freq_array) == 0
    if app.Individual_damping == 1
        app.Freq_array = cat(2, app.Freq_array, zeros(app.No_stages,1));
        app.UITable.Data = app.Freq_array;
        app.UITable.ColumnName = ["Resonant freq [Hz]" "Anti-resonant freq [Hz]" "Damping ratio [%]"];
        app.UITable.ColumnEditable = true(1,3);
    else
        app.Freq_array(:,3) = [];
        app.UITable.Data = app.Freq_array;
        app.UITable.ColumnName = ["Resonant freq [Hz]" "Anti-resonant freq [Hz]"];
        app.UITable.ColumnEditable = true(1,2);
    end
end
end

% Value changed function: GlobaldampingratioEditField
function GlobaldampingratioEditFieldValueChanged(app, event)
app.Damping_percent = app.GlobaldampingratioEditField.Value;
end

% Cell edit callback: UITable, UITable_8
function UITableCellEdit(app, event)
indices = event.Indices;
app.Freq_array(indices(1),indices(2)) = event.NewData;
end

% Button pushed function: CalculateButton
function CalculateButtonPushed(app, event)
app.Init_synth_params = Synthesizer(app.Individual_damping,app.Damping_percent,app.Freq_array,app.H_Factor,0,[]);
app.UITable_3.Data = zeros(length(app.No_stages),3);
app.UITable_3.Data = app.Init_synth_params(:,[1,2,3]);
end

% Button pushed function: NextstepButton
function NextstepButtonPushed(app, event)
app.TabGroup.SelectedTab = app.SelectMaterialsTab;
end

% Button pushed function: LoadFileButton, LoadFileButton_2
function LoadFileButtonPushed(app, event)
file = 0;
app.UITable_7.Data = [];
app.UITable_7.ColumnName = [];
app.UITable_7.RowName = [];
app.UITable_12.Data = [];
app.UITable_12.ColumnName = [];
app.UITable_12.RowName = [];
file = uigetfile('*.xlsx'), 'Choose a File');
opts = detectImportOptions(file);

```

```

if file ~= 0
    app.File_available_materials = readtable(file,opts);
end
app.UITable_7.Data = app.File_available_materials(1:end, 2:end);
app.UITable_7.ColumnName = opts.VariableNames(2:end);
rownames = app.File_available_materials(1:end, 1);
app.UITable_7.RowName = table2cell(rownames);
app.UITable_12.Data = app.File_available_materials(1:end, 2:end);
app.Piezo_names = opts.VariableNames(2:end);
app.UITable_12.ColumnName = opts.VariableNames(2:end);
app.UITable_12.RowName = table2cell(rownames);
end

% Value changed function: SetstageSpinner, SetstageSpinner_2
function SetstageSpinnerValueChanged(app, event)
    if app.TabGroup.SelectedTab == app.SelectMaterialsTab
        app.Sel_stage = app.SetstageSpinner.Value;
    elseif app.TabGroup.SelectedTab == app.SelectMaterialsTab_2
        app.Sel_stage = app.SetstageSpinner_2.Value;
    end
end

% Cell selection callback: UITable_12, UITable_7
function UITable_7CellSelection(app, event)
    app.Chosen_mat = event.Indices;
    app.Chosen_mat = app.Chosen_mat(1,2);
end

% Button pushed function: ApplytostageButton
function ApplytostageButtonPushed(app, event)
    app.Chosen_mat_stages(app.Sel_stage) = app.Chosen_mat;
end

% Button pushed function: ApplytostageButton_2
function ApplytostageButton_2Pushed(app, event)
    app.Chosen_mat_stages(app.Sel_stage) = app.Chosen_mat;
    app.No_stages = length(app.Chosen_mat_stages);
    app.Freq_array = zeros(app.No_stages, 2);
    app.UITable_8.ColumnName = ["Resonant freq [Hz]" "Antiresonant freq [Hz]"];
    sg = uistyle("BackgroundColor",[0.9 0.9 0.9]);
    s0 = uistyle("BackgroundColor",[1 0.8 0.8]);
    removeStyle(app.UITable_8);
    addStyle(app.UITable_8,sg,"column",2)
    app.UITable_8.ColumnEditable = true(1,1);
    app.UITable_8.Data = app.Freq_array;
    if length(app.Chosen_mat_stages)>1
        addStyle(app.UITable_8,s0,"Row",find(app.Chosen_mat_stages==0));
    end
end

app.Init_synth_params = [];
app.UITable_9.Data = [];
app.CustomthicknessButton.Value = 1;
app.RoundbaseButton.Value = 1;
app.Analysis_shape = 2;
app.Base_plate_type = "Round base";
app.Selected_dim = "Custom thickness";
app.Temp_rho_ce = "cE";
app.UITable_11.ColumnName = ["Thickness [mm]" "Diameter [mm]"];
removeStyle(app.UITable_11);
addStyle(app.UITable_11,sg,"column",2);
app.Calculated_dims = zeros(app.No_stages,2);
app.UITable_11.Data = app.Calculated_dims;
app.UITable_11.ColumnEditable = [true false];
end

% Button pushed function: ClearstagesButton
function ClearstagesButtonPushed(app, event)
    app.Chosen_mat_stages = [];
end

% Button pushed function: ClearstagesButton_2
function ClearstagesButton_2Pushed(app, event)
    app.Chosen_mat_stages = [];
    app.No_stages = 0;
    app.Freq_array = zeros(app.No_stages, 2);
    app.UITable_8.Data = app.Freq_array;
    app.Init_synth_params = [];
    app.UITable_9.Data = [];

    app.RoundbaseButton.Value = 1;
    app.Analysis_shape = 1;
    app.CustomsurfaceareaButton.Value = 1;
    app.Calculated_dims = [];
    app.UITable_11.Data = [];
end

% Button pushed function: NextstepButton_2
function NextstepButton_2Pushed(app, event)
    app.TabGroup.SelectedTab = app.SetplatedimensionsTab;
end

% Button pushed function: NextstepButton_4
function NextstepButton_4Pushed(app, event)
    app.TabGroup.SelectedTab = app.SetinitialparametersTab_2;
    app.ScalingfactorHEditField_2.Value = app.H_Factor;
end

% Button pushed function: CalculateButton_3
function CalculateButton_3Pushed(app, event)
    for i=1:size(app.Chosen_mat_stages,2)
        k33s(i) = table2array(app.File_available_materials(1,app.Chosen_mat_stages(i)+1));
        MQs(i) = table2array(app.File_available_materials(7,app.Chosen_mat_stages(i)+1));
    end
    app.Init_synth_params = Synthesizer(app.Individual_damping,app.Damping_percent,app.Freq_array,app.H_Factor,1,k33s,MQs);
end

```

```

app.UITable_3.Data = zeros(length(app.No_stages),3);
app.UITable_3.Data = app.Init_synth_params(:,[1,2,3]);
app.Freq_array(:,2) = app.Init_synth_params(:,5);
app.UITable_8.Data = app.Freq_array;
app.UITable_8.ColumnFormat = repmat({'bank'},1,size(app.UITable_8.Data,2));
app.UITable_9.Data = zeros(length(app.No_stages),3);
app.UITable_9.Data = app.Init_synth_params(:,[1,2,3]);
end

% Button pushed function: NextStepButton
function NextStepButtonPushed(app, event)
    app.TabGroup.SelectedTab = app.PlatecalculatorTab;
end

% Button pushed function: BacktobeginningButton
function BacktobeginningButtonPushed(app, event)
    app.TabGroup.SelectedTab = app.SetinitialparametersTab;
end

% Value changed function: RoundbaseCheckBox
function RoundbaseCheckBoxValueChanged(app, event)
    app.Round_plates = app.RoundbaseCheckBox.Value;
    if app.Round_plates == 0
        app.Analysis_shape = 3;
        app.Plate_dims = zeros(app.No_stages,3);
        app.UITable_6.Data = app.Plate_dims;
        app.UITable_6.ColumnName = ["Thickness [mm]" "Width [mm]" "Length [mm]"];
        app.UITable_6.ColumnEditable = true(1,3);
    elseif app.Round_plates == 1
        app.Analysis_shape = 2;
        app.Plate_dims = zeros(app.No_stages,2);
        app.UITable_6.Data = app.Plate_dims;
        app.UITable_6.ColumnName = ["Thickness [mm]" "Diameter [mm]"];
        app.UITable_6.ColumnEditable = true(1,2);
    end
end

% Cell edit callback: UITable_6
function UITable_6CellEdit(app, event)
    indices = event.Indices;
    app.Plate_dims(indices(1),indices(2)) = event.NewData;
end

% Button pushed function: CalculateButton_2
function CalculateButton_2Pushed(app, event)
    app.Piezo_params = [];
    [names, app.Piezo_params] = Piezo_calculator(app.Freq_array,app.Init_synth_params,app.Plate_dims,app.Round_plates);
    app.UITable_5.Data = zeros(10,length(app.No_stages));
    numbers = string(reshape(repmat([1:size(app.Freq_array,1)],2,1),1,[]));
    strings = string(zeros(1,size(app.Freq_array,1)));
    strings = append(repmat(["Stage_synth " "Stage_mat "],1,size(app.Freq_array,1)),numbers);
    insert = zeros(10,size(app.Freq_array,1));
    if length(app.Chosen_mat_stages)<size(app.Freq_array,1)
        app.Chosen_mat_stages = cat(2,app.Chosen_mat_stages,zeros(1,size(app.Freq_array,1)-length(app.Chosen_mat_stages)));
    end
    for i=1:size(app.Freq_array,1)
        if app.Chosen_mat_stages(i) ~= 0
            insert(:,i) = table2array(app.File_available_materials(:,app.Chosen_mat_stages(i)+1));
        end
    end
    output = zeros(10,2*size(app.Freq_array,1));
    output(:,1:2:2*size(app.Freq_array,1)) = app.Piezo_params(:,1:end);
    output(:,2:2:2*size(app.Freq_array,1)) = insert(:,1:end);
    app.UITable_5.ColumnName = strings;
    app.UITable_5.RowName = names;
    app.UITable_5.Data = output;
    [row,col] = find(output==0);
    s0 = uistyle("BackgroundColor",[1 0.8 0.8]);
    sg = uistyle("BackgroundColor",[0.9 0.9 0.9]);
    removeStyle(app.UITable_5);
    addStyle(app.UITable_5,"column",(2:2:2*size(app.Freq_array,1)));
    addStyle(app.UITable_5,s0,"cell",[row,col]);
end

% Selection changed function: SpecifyplatebasetypeButtonGroup
function SpecifyplatebasetypeButtonGroupSelectionChanged(app, event)
    app.Base_plate_type = app.SpecifyplatebasetypeButtonGroup.SelectedObject.Text;
    sg = uistyle("BackgroundColor",[0.9 0.9 0.9]);

    if app.RoundbaseButton.Value == 1
        app.Analysis_shape = 2;
        app.UITable_11.ColumnName = ["Thickness [mm]" "Diameter [mm]"];
        app.Calculated_dims = zeros(app.No_stages,2);
        app.UITable_11.Data = app.Calculated_dims;
        if app.CustomsurfaceareaButton.Value == 1
            removeStyle(app.UITable_11);
            addStyle(app.UITable_11,sg,"column",1);
            app.UITable_11.ColumnEditable = [false true];
        elseif app.CustomthicknessButton.Value == 1
            removeStyle(app.UITable_11);
            addStyle(app.UITable_11,sg,"column",2);
            app.UITable_11.ColumnEditable = [true false];
        end
    elseif app.SquarebaseButton.Value == 1
        app.Analysis_shape = 1;
        app.UITable_11.ColumnName = ["Thickness [mm]" "Width [mm]"];
        app.Calculated_dims = zeros(app.No_stages,2);
        app.UITable_11.Data = app.Calculated_dims;
        if app.CustomsurfaceareaButton.Value == 1
            removeStyle(app.UITable_11);
            addStyle(app.UITable_11,sg,"column",1);
            app.UITable_11.ColumnEditable = [false true];
        elseif app.CustomthicknessButton.Value == 1
            removeStyle(app.UITable_11);

```



```

        addStyle(app.UITable_11,sg,"column",2);
        app.UITable_11.ColumnEditable = [true false];
    end
elseif app.RectangularbaseButton.Value == 1
    app.Analysis_shape = 3;
    app.UITable_11.ColumnName = ["Thickness [mm]" "Width [mm]" "Length [mm]"];
    app.Calculated_dims = zeros(app.No_stages,3);
    app.UITable_11.Data = app.Calculated_dims;
    if app.CustomsurfaceareaButton.Value == 1
        removeStyle(app.UITable_11);
        addStyle(app.UITable_11,sg,"column",1);
        app.UITable_11.ColumnEditable = [false true true];
    elseif app.CustomthicknessButton.Value == 1
        removeStyle(app.UITable_11);
        addStyle(app.UITable_11,sg,"column",[2 3]);
        app.UITable_11.ColumnEditable = [true false false];
    end
end
end
end

% Selection changed function: SpecifycustomdimensionsButtonGroup
function SpecifycustomdimensionsButtonGroupSelectionChanged(app, event)
    app.Selected_dim = app.SpecifycustomdimensionsButtonGroup.SelectedObject.Text;
    sg = uistyle("BackgroundColor",[0.9 0.9 0.9]);

    if app.RoundbaseButton.Value == 1
        app.UITable_11.ColumnName = ["Thickness [mm]" "Diameter [mm]"];
        app.Calculated_dims = zeros(app.No_stages,2);
        app.UITable_11.Data = app.Calculated_dims;
        if app.CustomsurfaceareaButton.Value == 1
            removeStyle(app.UITable_11);
            addStyle(app.UITable_11,sg,"column",1);
            app.UITable_11.ColumnEditable = [false true];
        elseif app.CustomthicknessButton.Value == 1
            removeStyle(app.UITable_11);
            addStyle(app.UITable_11,sg,"column",2);
            app.UITable_11.ColumnEditable = [true false];
        end
    elseif app.SquarebaseButton.Value == 1
        app.UITable_11.ColumnName = ["Thickness [mm]" "Width [mm]"];
        app.Calculated_dims = zeros(app.No_stages,2);
        app.UITable_11.Data = app.Calculated_dims;
        if app.CustomsurfaceareaButton.Value == 1
            removeStyle(app.UITable_11);
            addStyle(app.UITable_11,sg,"column",1);
            app.UITable_11.ColumnEditable = [false true];
        elseif app.CustomthicknessButton.Value == 1
            removeStyle(app.UITable_11);
            addStyle(app.UITable_11,sg,"column",2);
            app.UITable_11.ColumnEditable = [true false];
        end
    elseif app.RectangularbaseButton.Value == 1
        app.UITable_11.ColumnName = ["Thickness [mm]" "Width [mm]" "Length [mm]"];
        app.Calculated_dims = zeros(app.No_stages,3);
        app.UITable_11.Data = app.Calculated_dims;
        if app.CustomsurfaceareaButton.Value == 1
            removeStyle(app.UITable_11);
            addStyle(app.UITable_11,sg,"column",1);
            app.UITable_11.ColumnEditable = [false true true];
        elseif app.CustomthicknessButton.Value == 1
            removeStyle(app.UITable_11);
            addStyle(app.UITable_11,sg,"column",[2 3]);
            app.UITable_11.ColumnEditable = [true false false];
        end
    end
end
end

% Cell edit callback: UITable_11
function UITable_11CellEdit(app, event)
    indices = event.Indices;
    app.Calculated_dims(indices(1),indices(2)) = event.NewData;
end

% Button pushed function: BacktobeginningButton_2
function BacktobeginningButton_2Pushed(app, event)
    app.TabGroup.SelectedTab = app.SelectMaterialsTab_2;
end

% Button pushed function: CalculateButton_4
function CalculateButton_4Pushed(app, event)
    densities = table2array(app.File_available_materials(10,app.Chosen_mat_stages+1));
    stiffnesses = table2array(app.File_available_materials(6,app.Chosen_mat_stages+1));
    output = Dim_calculator(app.Init_synth_params,app.Calculated_dims,app.Base_plate_type,app.Selected_dim,densities,stiffnesses,app.1);
    if app.Selected_dim == "Custom surface area"
        app.Calculated_dims(:,1) = output;
        app.UITable_11.Data = app.Calculated_dims;
    elseif app.Selected_dim == "Custom thickness"
        if app.Base_plate_type == "Round base" | app.Base_plate_type == "Square base"
            app.Calculated_dims(:,2) = output;
            app.UITable_11.Data = app.Calculated_dims;
        elseif app.Base_plate_type == "Rectangular base"
            app.Calculated_dims(:,[2 3]) = output;
            app.UITable_11.Data = app.Calculated_dims;
        end
    end
end
app.UITable_10.RowName = table2cell(app.File_available_materials(:,1));
app.UITable_10.ColumnName = app.Piezo_names(1,app.Chosen_mat_stages);
app.UITable_10.Data = app.File_available_materials(:,app.Chosen_mat_stages+1);
end

% Button pushed function: AnalysemodelButton, AnalysemodelButton_3
function AnalysemodelButton_Pushed(app, event)
    if app.TabGroup.SelectedTab == app.SetplatedimensionsTab
        app.Analysis_prev_synth_choice = 1;
    end
end

```

```

if app.Analysis_shape == 1 || app.Analysis_shape == 2
    app.Analysis_input = app.UITable_6.Data.';
    app.Analysis_input = cat(1,app.Analysis_input,zeros(1,size(app.Analysis_input,2)));
elseif app.Analysis_shape == 3
    app.Analysis_input = app.UITable_6.Data.';
end

app.SetinitialparametersTab.Parent = [];
app.SelectMaterialsTab.Parent = [];
app.SetplatedimensionsTab.Parent = [];

app.Analysis_input = cat(1,app.Analysis_input,app.UITable_3.Data.');
app.Analysis_input = cat(1,app.Analysis_input,app.UITable_5.Data(:,1:2:end));
app.Analysis_input = cat(1,app.Analysis_input,app.UITable.Data(:,1:1:2).');

elseif app.TabGroup.SelectedTab == app.PlatecalculatorTab
    app.Analysis_prev_synth_choice = 2;

    if app.Analysis_shape == 1 || app.Analysis_shape == 2
        app.Analysis_input = app.UITable_11.Data.';
        app.Analysis_input = cat(1,app.Analysis_input,zeros(1,size(app.Analysis_input,2)));
    elseif app.Analysis_shape == 3
        app.Analysis_input = app.UITable_11.Data.';
    end

    app.Analysis_input = cat(1,app.Analysis_input,app.UITable_9.Data.');
    app.Analysis_input = cat(1,app.Analysis_input,table2array(app.UITable_10.Data));
    app.Analysis_input = cat(1,app.Analysis_input,app.UITable_8.Data.');

    app.SelectMaterialsTab_2.Parent = [];
    app.SetinitialparametersTab_2.Parent = [];
    app.PlatecalculatorTab.Parent = [];
end

app.ModelanalyzerTab.Parent = app.TabGroup;
app.TabGroup.SelectedTab = app.ModelanalyzerTab;

app.UITable2.Data = app.Analysis_input;
array = string(1:size(app.Analysis_input,2));
strings = append('Stage ', array);
app.UITable2.ColumnName = strings;
app.UITable2.ColumnFormat = repmat({'char'},1,size(app.Analysis_input,2));

if app.Analysis_shape == 1
    shape = zeros(1,size(app.Analysis_input,2))+1;
elseif app.Analysis_shape == 2
    shape = zeros(1,size(app.Analysis_input,2))+2;
elseif app.Analysis_shape == 3
    shape = zeros(1,size(app.Analysis_input,2))+3;
end
app.Analysis_input = cat(1,app.Analysis_input,shape);

if app.Analysis_shape == 1
    app.UITable2.RowName = ["Thickness [mm]" "Width [mm]" "---" "Eq Mass [g]" "Eq stiffness [N/m]" "Eq Damping [Ns/m]" "Coupling k"];
elseif app.Analysis_shape == 2
    app.UITable2.RowName = ["Thickness [mm]" "Diameter [mm]" "---" "Eq Mass [g]" "Eq stiffness [N/m]" "Eq Damping [Ns/m]" "Coupling"];
elseif app.Analysis_shape == 3
    app.UITable2.RowName = ["Thickness [mm]" "Width [mm]" "Length [mm]" "Eq Mass [g]" "Eq stiffness [N/m]" "Eq Damping [Ns/m]" "Cc"];
end

app.UITable2.ColumnEditable = true(1,size(app.Analysis_input,2));

app.ShowgraphsButton.Enable = 0;

app.Analysis_range_set = 0;
app.Analysis_range = [];
app.Analysis_ext_force = 0;
app.Analysis_ext_force_sincos = 0;
app.sinButton.Value = 1;
app.Analysis_ext_force_set = 0;
app.Analysis_ext_voltage = 0;
app.Analysis_ext_voltage_sincos = 0;
app.sinButton_2.Value = 1;
app.Analysis_ext_voltage_set = 0;
app.Analysis_accuracies = 0;
app.Analysis_accuracies_set = 0;
end

% Button pushed function: SetfreqrangeButton
function SetfreqrangeButtonPushed(app, event)
    prompt = {'Enter lowest frequency in the analysed range:','Enter highest frequency in the analysed range:'};
    dlgtitle = 'Input';
    fieldsize = [1 45; 1 45];
    definput = {'0','0'};
    app.Analysis_range = inputdlg(prompt,dlgtitle,fieldsize,definput);
    if isempty(app.Analysis_range) == 1
        return;
    end
    first = str2double(app.Analysis_range{1});
    last = str2double(app.Analysis_range{2});
    app.Analysis_range = [first last];
    app.Analysis_range_set = 1;

    app.ShowgraphsButton.Enable = 0;
end

% Button pushed function: SetaccuracyButton
function SetaccuracyButtonPushed(app, event)
    prompt = {'Enter the sample interval of the high accuracy ranges:','Enter the sample interval of the low accuracy ranges:'};
    dlgtitle = 'Input';
    fieldsize = [1 45; 1 45];
    definput = {'5','1000'};
    app.Analysis_accuracies = inputdlg(prompt,dlgtitle,fieldsize,definput);

```

```

if isempty(app.Analysis_accuracies) == 1
    return;
end
first = str2double(app.Analysis_accuracies{1});
last = str2double(app.Analysis_accuracies{2});
app.Analysis_accuracies = [first last];
app.Analysis_accuracies_set = 1;

app.ShowgraphsButton.Enable = 0;
end

% Button pushed function: SetexternalforceButton
function SetexternalforceButtonPushed(app, event)
    prompt = {'Set external force applied to the stack [N]:'};
    dlgtitle = 'Input';
    fieldsize = [1 45];
    definput = {'0'};
    check = inputdlg(prompt,dlgtitle,fieldsize,definput);
    if isempty(check) == 1
        return;
    end
    zer = zeros(1,size(app.Analysis_input,2)-1);
    forc = cell2sym(check);
    app.Analysis_ext_force = double(cat(2,forc,zer));
    app.Analysis_ext_force_set = 1;

    app.ShowgraphsButton.Enable = 0;
end

% Selection changed function: OffsetButtonGroup
function OffsetButtonGroupSelectionChanged(app, event)
    if app.sinButton.Value == 1
        app.Analysis_ext_force_sincos = 0;
    elseif app.cosButton.Value == 1
        app.Analysis_ext_force_sincos = 1;
    end

    app.ShowgraphsButton.Enable = 0;
end

% Button pushed function: SetexternalvoltageButton
function SetexternalvoltageButtonPushed(app, event)
    prompt = {'Set external voltage applied to the stack [V]:'};
    dlgtitle = 'Input';
    fieldsize = [1 45];
    definput = {'0'};
    check = inputdlg(prompt,dlgtitle,fieldsize,definput);
    if isempty(check) == 1
        return;
    end
    app.Analysis_ext_voltage = double(zeros(1,size(app.Analysis_input,2)) + cell2sym(check));
    app.Analysis_ext_voltage_set = 1;

    app.ShowgraphsButton.Enable = 0;
end

% Selection changed function: OffsetButtonGroup_2
function OffsetButtonGroup_2SelectionChanged(app, event)
    if app.sinButton_2.Value == 1
        app.Analysis_ext_voltage_sincos = 0;
    elseif app.cosButton_2.Value == 1
        app.Analysis_ext_voltage_sincos = 1;
    end

    app.ShowgraphsButton.Enable = 0;
end

% Cell edit callback: UITable2
function EditPlateDetails(app, event)
    indices = event.Indices;
    app.Analysis_input(indices(1),indices(2)) = event.NewData;
end

% Button pushed function: AnalyseButton
function AnalyseButtonPushed(app, event)
    if app.Analysis_range_set == 0 || app.Analysis_ext_force_set == 0 || app.Analysis_ext_voltage_set == 0 || app.Analysis_accuracies_set == 0
        msgbox("Please finish setting up initial starting conditions for the analysis process","Error","error");
        return
    end
    app.Analysis_results = Piezo_analyzer(app.Analysis_input,app.Analysis_range,app.Analysis_ext_force,app.Analysis_ext_force_sincos,app.Analysis_ext_voltage,app.Analysis_ext_voltage_sincos);
    app.ShowgraphsButton.Enable = 1;
end

% Button pushed function: ShowgraphsButton
function ShowgraphsButtonPushed(app, event)
    Graph_generator(app.Analysis_results,app.Analysis_range,app.Analysis_input,app.Analysis_accuracies)
end

% Button pushed function: BackButton
function BackButtonPushed(app, event)
    app.CancelAnalysisTab.Parent = app.TabGroup;
    app.Recent_tab = app.TabGroup.SelectedTab;
    app.TabGroup.SelectedTab = app.CancelAnalysisTab;
end

% Button pushed function: ProceedButton_2
function ProceedButton_2Pushed(app, event)
    app.UITable2.Data = [];
    app.Analysis_input = [];
    app.CancelAnalysisTab.Parent = [];
    app.ModelanalyzerTab.Parent = [];

    if app.Analysis_prev_synth_choice == 1
        app.SetinitialparametersTab.Parent = app.TabGroup;
    end
end

```

```

        app.SelectMaterialsTab.Parent = app.TabGroup;
        app.SetplatedimensionsTab.Parent = app.TabGroup;
        app.TabGroup.SelectedTab = app.SetplatedimensionsTab;
    elseif app.Analysis_prev_synth_choice == 2
        app.SelectMaterialsTab_2.Parent = app.TabGroup;
        app.SetinitialparametersTab_2.Parent = app.TabGroup;
        app.PlatecalculatorTab.Parent = app.TabGroup;
        app.TabGroup.SelectedTab = app.PlatecalculatorTab;
    end
end

% Button pushed function: CancelButton_2
function CancelButton_2Pushed(app, event)
    app.TabGroup.SelectedTab = app.Recent_tab;
    app.CancelAnalysisTab.Parent = [];
end

% Callback function: PlatecalculatorTab, XButton, XButton_2,
% XButton_3, XButton_4, XButton_5, XButton_6, XButton_7
function XButtonPushed(app, event)
    app.ResetTab.Parent = app.TabGroup;
    app.Recent_tab = app.TabGroup.SelectedTab;
    app.TabGroup.SelectedTab = app.ResetTab;
end

% Button pushed function: ProceedButton
function ProceedButtonPushed(app, event)
    app.SynthesismethodTab.Parent = app.TabGroup;
    app.TabGroup.SelectedTab = app.SynthesismethodTab;
    app.No_stages = 1;
    app.Freq_array = 0;
    app.Damping_percent = 0;
    app.H_Factor = 0;
    app.Init_synth_params = 0;
    app.Plate_dims = 0;
    app.Piezo_params = 0;
    app.Chosen_mat = 0;
    app.Sel_stage = 1;
    app.Chosen_mat_stages = [];
    app.Individual_damping = 0;
    app.Round_plates = 0;
    app.Analysis_shape = 0;
    app.Recent_tab = 0;
    app.Analysis_prev_synth_choice = 0;
    app.UITable.Data = {};
    app.UITable2.Data = {};
    app.UITable_3.Data = {};
    app.UITable_5.Data = {};
    app.UITable_6.Data = {};
    app.UITable_7.Data = {};
    app.UITable_8.Data = {};
    app.UITable_9.Data = {};
    app.UITable_10.Data = {};
    app.UITable_11.Data = {};
    app.ResetTab.Parent = [];
    app.SetinitialparametersTab.Parent = [];
    app.SelectMaterialsTab.Parent = [];
    app.SetplatedimensionsTab.Parent = [];
    app.SetinitialparametersTab_2.Parent = [];
    app.SelectMaterialsTab_2.Parent = [];
    app.PlatecalculatorTab.Parent = [];
    app.ModelanalyzerTab.Parent = [];
    app.CancelAnalysisTab.Parent = [];
    app.Base_plate_type = "Round base";
    app.Selected_dim = "Custom thickness";
end

% Button pushed function: CancelButton
function CancelButtonPushed(app, event)
    app.TabGroup.SelectedTab = app.Recent_tab;
    app.ResetTab.Parent = [];
end

% Button down function: SetinitialparametersTab_2
function SetinitialparametersTab_2ButtonDown(app, event)
    app.ScalingfactorHEditField_2.Value = app.H_Factor;
end

% Component initialization
methods (Access = private)

% Create UIFigure and components
function createComponents(app)

    % Create UIFigure and hide until all components are created
    app.UIFigure = uifigure('Visible', 'off');
    app.UIFigure.Position = [100 100 799 656];
    app.UIFigure.Name = 'MATLAB App';
    app.UIFigure.Tag = '2';

    % Create TabGroup
    app.TabGroup = uitabgroup(app.UIFigure);
    app.TabGroup.Position = [2 1 798 656];

    % Create SynthesismethodTab
    app.SynthesismethodTab = uitab(app.TabGroup);
    app.SynthesismethodTab.Title = 'Synthesis method';

    % Create GridLayout4
    app.GridLayout4 = uigridlayout(app.SynthesismethodTab);
    app.GridLayout4.ColumnWidth = {'1x', 200, 200, '1x'};
    app.GridLayout4.RowHeight = {'1x', '1x', '1x', '1x', '1x', '1x', '1x', '1x', '1x', '1x'};

```

```

% Create ChoosedesiredsynthesismethodLabel
app.ChoosedesiredsynthesismethodLabel = uilabel(app.GridLayout4);
app.ChoosedesiredsynthesismethodLabel.HorizontalAlignment = 'center';
app.ChoosedesiredsynthesismethodLabel.FontSize = 20;
app.ChoosedesiredsynthesismethodLabel.Layout.Row = 5;
app.ChoosedesiredsynthesismethodLabel.Layout.Column = [1 4];
app.ChoosedesiredsynthesismethodLabel.Text = 'Choose desired synthesis method';

% Create PiezoelectricmodelsynthesizerLabel
app.PiezoelectricmodelsynthesizerLabel = uilabel(app.GridLayout4);
app.PiezoelectricmodelsynthesizerLabel.HorizontalAlignment = 'center';
app.PiezoelectricmodelsynthesizerLabel.FontSize = 36;
app.PiezoelectricmodelsynthesizerLabel.Layout.Row = 3;
app.PiezoelectricmodelsynthesizerLabel.Layout.Column = [1 4];
app.PiezoelectricmodelsynthesizerLabel.Text = 'Piezoelectric model synthesizer';

% Create ChoiceButton_1
app.ChoiceButton_1 = uibutton(app.GridLayout4, 'push');
app.ChoiceButton_1.ButtonPushedFcn = createCallbackFcn(app, @ChoiceButton_1Pushed, true);
app.ChoiceButton_1.WordWrap = 'on';
app.ChoiceButton_1.FontSize = 16;
app.ChoiceButton_1.Layout.Row = 6;
app.ChoiceButton_1.Layout.Column = 2;
app.ChoiceButton_1.Text = 'Synthesize piezoelectric parameters';

% Create ChoiceButton_2
app.ChoiceButton_2 = uibutton(app.GridLayout4, 'push');
app.ChoiceButton_2.ButtonPushedFcn = createCallbackFcn(app, @ChoiceButton_2Pushed, true);
app.ChoiceButton_2.WordWrap = 'on';
app.ChoiceButton_2.FontSize = 16;
app.ChoiceButton_2.Layout.Row = 6;
app.ChoiceButton_2.Layout.Column = 3;
app.ChoiceButton_2.Text = 'Synthesize piezoelectric plate dimensions';

% Create SetinitialparametersTab
app.SetinitialparametersTab = uitab(app.TabGroup);
app.SetinitialparametersTab.Title = 'Set initial parameters';

% Create GridLayout
app.GridLayout = uigridlayout(app.SetinitialparametersTab);
app.GridLayout.ColumnWidth = {40, 60, '0.7x', 120, 25, 120, '1x', 40};
app.GridLayout.RowHeight = {40, 40, 27, 27, 27, 27, 161, '2.65x', 27, 50, 26};
app.GridLayout.ColumnSpacing = 1.7;
app.GridLayout.Padding = [1.7 10 1.7 10];

% Create SetnumberofstagesLabel
app.SetnumberofstagesLabel = uilabel(app.GridLayout);
app.SetnumberofstagesLabel.FontSize = 18;
app.SetnumberofstagesLabel.Layout.Row = 3;
app.SetnumberofstagesLabel.Layout.Column = [3 4];
app.SetnumberofstagesLabel.Text = '< Set number of stages';

% Create SetnumberofstagesEditField
app.SetnumberofstagesEditField = uieditfield(app.GridLayout, 'numeric');
app.SetnumberofstagesEditField.Limits = [1 Inf];
app.SetnumberofstagesEditField.ValueChangedFcn = createCallbackFcn(app, @SetnumberofstagesEditFieldValueChanged, true);
app.SetnumberofstagesEditField.BusyAction = 'cancel';
app.SetnumberofstagesEditField.HorizontalAlignment = 'center';
app.SetnumberofstagesEditField.FontSize = 18;
app.SetnumberofstagesEditField.Layout.Row = 3;
app.SetnumberofstagesEditField.Layout.Column = 2;
app.SetnumberofstagesEditField.Value = 1;

% Create UITable
app.UITable = uitable(app.GridLayout);
app.UITable.ColumnName = '';
app.UITable.RowName = {};
app.UITable.ColumnEditable = true;
app.UITable.CellEditCallback = createCallbackFcn(app, @UITableCellEdit, true);
app.UITable.Multiselect = 'off';
app.UITable.BusyAction = 'cancel';
app.UITable.Layout.Row = [7 8];
app.UITable.Layout.Column = [2 4];
app.UITable.FontSize = 14;

% Create MechanicalmodelparameterssynthesizerLabel
app.MechanicalmodelparameterssynthesizerLabel = uilabel(app.GridLayout);
app.MechanicalmodelparameterssynthesizerLabel.HorizontalAlignment = 'center';
app.MechanicalmodelparameterssynthesizerLabel.FontSize = 24;
app.MechanicalmodelparameterssynthesizerLabel.Layout.Row = 1;
app.MechanicalmodelparameterssynthesizerLabel.Layout.Column = [1 8];
app.MechanicalmodelparameterssynthesizerLabel.Text = {'Mechanical model parameter synthesizer'; ''};

% Create FillinthesetofdesiredfrequenciesLabel
app.FillinthesetofdesiredfrequenciesLabel = uilabel(app.GridLayout);
app.FillinthesetofdesiredfrequenciesLabel.FontSize = 18;
app.FillinthesetofdesiredfrequenciesLabel.Layout.Row = 6;
app.FillinthesetofdesiredfrequenciesLabel.Layout.Column = [2 4];
app.FillinthesetofdesiredfrequenciesLabel.Text = {'Fill in the set of desired frequencies'; ''};

% Create DampingratioLabel
app.DampingratioLabel = uilabel(app.GridLayout);
app.DampingratioLabel.FontSize = 18;
app.DampingratioLabel.Layout.Row = 5;
app.DampingratioLabel.Layout.Column = [3 4];
app.DampingratioLabel.Text = '< Global damping ratio [%]';

% Create GlobaldampingratioEditField
app.GlobaldampingratioEditField = uieditfield(app.GridLayout, 'numeric');
app.GlobaldampingratioEditField.Limits = [0 100];
app.GlobaldampingratioEditField.ValueChangedFcn = createCallbackFcn(app, @GlobaldampingratioEditFieldValueChanged, true);
app.GlobaldampingratioEditField.BusyAction = 'cancel';
app.GlobaldampingratioEditField.HorizontalAlignment = 'center';
app.GlobaldampingratioEditField.FontSize = 18;

```

```

app.GlobaldampingratioEditField.Layout.Row = 5;
app.GlobaldampingratioEditField.Layout.Column = 2;

% Create Label
app.Label = uilabel(app.GridLayout);
app.Label.HorizontalAlignment = 'center';
app.Label.FontSize = 40;
app.Label.Layout.Row = [6 8];
app.Label.Layout.Column = 5;
app.Label.Text = '>';

% Create ScalingfactorHLabel
app.ScalingfactorHLabel = uilabel(app.GridLayout);
app.ScalingfactorHLabel.FontSize = 18;
app.ScalingfactorHLabel.Layout.Row = 4;
app.ScalingfactorHLabel.Layout.Column = [3 4];
app.ScalingfactorHLabel.Text = '< Scaling factor H';

% Create ScalingfactorHEditField
app.ScalingfactorHEditField = uieditfield(app.GridLayout, 'numeric');
app.ScalingfactorHEditField.Limits = [0 Inf];
app.ScalingfactorHEditField.ValueChangedFcn = createCallbackFcn(app, @ScalingfactorHEditFieldValueChanged, true);
app.ScalingfactorHEditField.BusyAction = 'cancel';
app.ScalingfactorHEditField.HorizontalAlignment = 'center';
app.ScalingfactorHEditField.FontSize = 18;
app.ScalingfactorHEditField.Layout.Row = 4;
app.ScalingfactorHEditField.Layout.Column = 2;

% Create UITable_3
app.UITable_3 = uitable(app.GridLayout);
app.UITable_3.ColumnName = {'Masses [g]'; 'Springs [N/m]'; 'Dampers [Ns/m]'};
app.UITable_3.RowName = {};
app.UITable_3.Layout.Row = [3 8];
app.UITable_3.Layout.Column = [6 7];

% Create CalculateButton
app.CalculateButton = uibutton(app.GridLayout, 'push');
app.CalculateButton.ButtonPushedFcn = createCallbackFcn(app, @CalculateButtonPushed, true);
app.CalculateButton.FontSize = 16;
app.CalculateButton.Layout.Row = 10;
app.CalculateButton.Layout.Column = 4;
app.CalculateButton.Text = 'Calculate';

% Create NextstepButton
app.NextstepButton = uibutton(app.GridLayout, 'push');
app.NextstepButton.ButtonPushedFcn = createCallbackFcn(app, @NextstepButtonPushed, true);
app.NextstepButton.FontSize = 16;
app.NextstepButton.Layout.Row = 10;
app.NextstepButton.Layout.Column = 6;
app.NextstepButton.Text = 'Next step';

% Create IndividualdampingratiosperstageCheckBox
app.IndividualdampingratiosperstageCheckBox = uicheckbox(app.GridLayout);
app.IndividualdampingratiosperstageCheckBox.ValueChangedFcn = createCallbackFcn(app, @IndividualdampingratiosperstageCheckBoxValue
app.IndividualdampingratiosperstageCheckBox.Text = ' Individual damping ratios per stage';
app.IndividualdampingratiosperstageCheckBox.FontSize = 18;
app.IndividualdampingratiosperstageCheckBox.Layout.Row = 9;
app.IndividualdampingratiosperstageCheckBox.Layout.Column = [2 4];

% Create XButton
app.XButton = uibutton(app.GridLayout, 'push');
app.XButton.ButtonPushedFcn = createCallbackFcn(app, @XButtonPushed, true);
app.XButton.IconAlignment = 'center';
app.XButton.BackgroundColor = [1 0 0];
app.XButton.FontName = 'Arial';
app.XButton.FontSize = 26;
app.XButton.FontWeight = 'bold';
app.XButton.FontColor = [1 1 1];
app.XButton.Layout.Row = 1;
app.XButton.Layout.Column = 8;
app.XButton.Text = 'X';

% Create SelectMaterialsTab
app.SelectMaterialsTab = uitab(app.TabGroup);
app.SelectMaterialsTab.Title = 'Select Materials';

% Create GridLayout2
app.GridLayout2 = uigridlayout(app.SelectMaterialsTab);
app.GridLayout2.ColumnWidth = {40, 120, '1x', 73, 120, 120, 120, 40};
app.GridLayout2.RowHeight = {40, 40, '23.6x', 50, 26};
app.GridLayout2.ColumnSpacing = 2.666666666666667;
app.GridLayout2.Padding = [2.666666666666667 10 2.666666666666667 10];

% Create ListofavailableexistingpiezoelectricmaterialsLabel
app.ListofavailableexistingpiezoelectricmaterialsLabel = uilabel(app.GridLayout2);
app.ListofavailableexistingpiezoelectricmaterialsLabel.HorizontalAlignment = 'center';
app.ListofavailableexistingpiezoelectricmaterialsLabel.FontSize = 24;
app.ListofavailableexistingpiezoelectricmaterialsLabel.Layout.Row = 1;
app.ListofavailableexistingpiezoelectricmaterialsLabel.Layout.Column = [1 8];
app.ListofavailableexistingpiezoelectricmaterialsLabel.Text = 'List of available existing piezoelectric materials';

% Create UITable_7
app.UITable_7 = uitable(app.GridLayout2);
app.UITable_7.ColumnName = '';
app.UITable_7.RowName = {};
app.UITable_7.SelectionType = 'column';
app.UITable_7.CellSelectionCallback = createCallbackFcn(app, @UITable_7CellSelection, true);
app.UITable_7.Multiselect = 'off';
app.UITable_7.Layout.Row = 3;
app.UITable_7.Layout.Column = [2 7];

% Create LoadFileButton
app.LoadFileButton = uibutton(app.GridLayout2, 'push');
app.LoadFileButton.ButtonPushedFcn = createCallbackFcn(app, @LoadFileButtonPushed, true);

```



```

app.LoadFileButton.FontSize = 16;
app.LoadFileButton.Layout.Row = 4;
app.LoadFileButton.Layout.Column = 2;
app.LoadFileButton.Text = 'Load File';

% Create ApplytostageButton
app.ApplytostageButton = uibutton(app.GridLayout2, 'push');
app.ApplytostageButton.ButtonPushedFcn = createCallbackFcn(app, @ApplytostageButtonPushed, true);
app.ApplytostageButton.FontSize = 16;
app.ApplytostageButton.Layout.Row = 4;
app.ApplytostageButton.Layout.Column = 5;
app.ApplytostageButton.Text = 'Apply to stage';

% Create SetstageSpinnerLabel
app.SetstageSpinnerLabel = uilabel(app.GridLayout2);
app.SetstageSpinnerLabel.HorizontalAlignment = 'right';
app.SetstageSpinnerLabel.FontSize = 16;
app.SetstageSpinnerLabel.Layout.Row = 4;
app.SetstageSpinnerLabel.Layout.Column = 3;
app.SetstageSpinnerLabel.Text = 'Set stage';

% Create SetstageSpinner
app.SetstageSpinner = uispinner(app.GridLayout2);
app.SetstageSpinner.Limits = [1 Inf];
app.SetstageSpinner.ValueChangedFcn = createCallbackFcn(app, @SetstageSpinnerValueChanged, true);
app.SetstageSpinner.FontSize = 16;
app.SetstageSpinner.Layout.Row = 4;
app.SetstageSpinner.Layout.Column = 4;
app.SetstageSpinner.Value = 1;

% Create ClearstagesButton
app.ClearstagesButton = uibutton(app.GridLayout2, 'push');
app.ClearstagesButton.ButtonPushedFcn = createCallbackFcn(app, @ClearstagesButtonPushed, true);
app.ClearstagesButton.FontSize = 16;
app.ClearstagesButton.Layout.Row = 4;
app.ClearstagesButton.Layout.Column = 6;
app.ClearstagesButton.Text = 'Clear stages';

% Create NextstepButton_2
app.NextstepButton_2 = uibutton(app.GridLayout2, 'push');
app.NextstepButton_2.ButtonPushedFcn = createCallbackFcn(app, @NextstepButton_2Pushed, true);
app.NextstepButton_2.FontSize = 16;
app.NextstepButton_2.Layout.Row = 4;
app.NextstepButton_2.Layout.Column = 7;
app.NextstepButton_2.Text = 'Next step';

% Create XButton_2
app.XButton_2 = uibutton(app.GridLayout2, 'push');
app.XButton_2.ButtonPushedFcn = createCallbackFcn(app, @XButtonPushed, true);
app.XButton_2.IconAlignment = 'center';
app.XButton_2.BackgroundColor = [1 0 0];
app.XButton_2.FontName = 'Arial';
app.XButton_2.FontSize = 26;
app.XButton_2.FontWeight = 'bold';
app.XButton_2.FontColor = [1 1 1];
app.XButton_2.Layout.Row = 1;
app.XButton_2.Layout.Column = 8;
app.XButton_2.Text = 'X';

% Create SetplatedimensionsTab
app.SetplatedimensionsTab = uitab(app.TabGroup);
app.SetplatedimensionsTab.Title = 'Set plate dimensions';

% Create GridLayout3
app.GridLayout3 = uigridlayout(app.SetplatedimensionsTab);
app.GridLayout3.ColumnWidth = {40, '1x', 120, 47, 26, 47, 120, '1x', 40};
app.GridLayout3.RowHeight = {40, 40, 32, 27, '3.61x', 161, 80, 50, 26};

% Create PiezoelectricparametersynthesizerLabel
app.PiezoelectricparametersynthesizerLabel = uilabel(app.GridLayout3);
app.PiezoelectricparametersynthesizerLabel.HorizontalAlignment = 'center';
app.PiezoelectricparametersynthesizerLabel.FontSize = 24;
app.PiezoelectricparametersynthesizerLabel.Layout.Row = 1;
app.PiezoelectricparametersynthesizerLabel.Layout.Column = [1 9];
app.PiezoelectricparametersynthesizerLabel.Text = {'Piezoelectric parameter synthesizer'; ''};

% Create BacktobeginningButton
app.BacktobeginningButton = uibutton(app.GridLayout3, 'push');
app.BacktobeginningButton.ButtonPushedFcn = createCallbackFcn(app, @BacktobeginningButtonPushed, true);
app.BacktobeginningButton.WordWrap = 'on';
app.BacktobeginningButton.FontSize = 16;
app.BacktobeginningButton.Layout.Row = 8;
app.BacktobeginningButton.Layout.Column = 3;
app.BacktobeginningButton.Text = 'Back to beginning';

% Create CalculateButton_2
app.CalculateButton_2 = uibutton(app.GridLayout3, 'push');
app.CalculateButton_2.ButtonPushedFcn = createCallbackFcn(app, @CalculateButton_2Pushed, true);
app.CalculateButton_2.FontSize = 16;
app.CalculateButton_2.Layout.Row = 8;
app.CalculateButton_2.Layout.Column = [4 6];
app.CalculateButton_2.Text = 'Calculate';

% Create AnalysemodelButton_3
app.AnalysemodelButton_3 = uibutton(app.GridLayout3, 'push');
app.AnalysemodelButton_3.ButtonPushedFcn = createCallbackFcn(app, @AnalysemodelButton_Pushed, true);
app.AnalysemodelButton_3.FontSize = 16;
app.AnalysemodelButton_3.Layout.Row = 8;
app.AnalysemodelButton_3.Layout.Column = 7;
app.AnalysemodelButton_3.Text = 'Analyse model';

% Create UITable_5
app.UITable_5 = uitable(app.GridLayout3);
app.UITable_5.ColumnName = '';

```

```

app.UITable_5.RowName = {};
app.UITable_5.Layout.Row = [3 7];
app.UITable_5.Layout.Column = [6 8];

% Create UITable_6
app.UITable_6 = uitable(app.GridLayout3);
app.UITable_6.ColumnName = '';
app.UITable_6.RowName = {};
app.UITable_6.ColumnEditable = true;
app.UITable_6.CellEditCallback = createCallbackFcn(app, @UITable_6CellEdit, true);
app.UITable_6.Multiselect = 'off';
app.UITable_6.BusyAction = 'cancel';
app.UITable_6.Layout.Row = [5 7];
app.UITable_6.Layout.Column = [2 4];
app.UITable_6.FontSize = 14;

% Create FillindesiredplatedimensionsLabel
app.FillindesiredplatedimensionsLabel = uilabel(app.GridLayout3);
app.FillindesiredplatedimensionsLabel.FontSize = 18;
app.FillindesiredplatedimensionsLabel.Layout.Row = 3;
app.FillindesiredplatedimensionsLabel.Layout.Column = [2 5];
app.FillindesiredplatedimensionsLabel.Text = {'Fill in desired plate dimensions'; ''};

% Create Label_2
app.Label_2 = uilabel(app.GridLayout3);
app.Label_2.HorizontalAlignment = 'center';
app.Label_2.FontSize = 40;
app.Label_2.Layout.Row = [5 7];
app.Label_2.Layout.Column = 5;
app.Label_2.Text = '>';

% Create RoundbaseCheckBox
app.RoundbaseCheckBox = uicheckbox(app.GridLayout3);
app.RoundbaseCheckBox.ValueChangedFcn = createCallbackFcn(app, @RoundbaseCheckBoxValueChanged, true);
app.RoundbaseCheckBox.Text = 'Round base';
app.RoundbaseCheckBox.FontSize = 18;
app.RoundbaseCheckBox.Layout.Row = 4;
app.RoundbaseCheckBox.Layout.Column = [2 5];

% Create XButton_3
app.XButton_3 = uibutton(app.GridLayout3, 'push');
app.XButton_3.ButtonPushedFcn = createCallbackFcn(app, @XButtonPushed, true);
app.XButton_3.IconAlignment = 'center';
app.XButton_3.BackgroundColor = [1 0 0];
app.XButton_3.FontName = 'Arial';
app.XButton_3.FontSize = 26;
app.XButton_3.FontWeight = 'bold';
app.XButton_3.FontColor = [1 1 1];
app.XButton_3.Layout.Row = 1;
app.XButton_3.Layout.Column = 9;
app.XButton_3.Text = 'X';

% Create SelectMaterialsTab_2
app.SelectMaterialsTab_2 = uitab(app.TabGroup);
app.SelectMaterialsTab_2.Title = 'Select Materials';

% Create GridLayout2_2
app.GridLayout2_2 = uigridlayout(app.SelectMaterialsTab_2);
app.GridLayout2_2.ColumnWidth = {40, 120, '1x', 73, 120, 120, 40};
app.GridLayout2_2.RowHeight = {40, 40, '23.6x', 50, 26};
app.GridLayout2_2.ColumnSpacing = 2.666666666666667;
app.GridLayout2_2.Padding = [2.666666666666667 10 2.666666666666667 10];

% Create ListofavailableexistingpiezoelectricmaterialsLabel_2
app.ListofavailableexistingpiezoelectricmaterialsLabel_2 = uilabel(app.GridLayout2_2);
app.ListofavailableexistingpiezoelectricmaterialsLabel_2.HorizontalAlignment = 'center';
app.ListofavailableexistingpiezoelectricmaterialsLabel_2.FontSize = 24;
app.ListofavailableexistingpiezoelectricmaterialsLabel_2.Layout.Row = 1;
app.ListofavailableexistingpiezoelectricmaterialsLabel_2.Layout.Column = [1 8];
app.ListofavailableexistingpiezoelectricmaterialsLabel_2.Text = 'List of available existing piezoelectric materials';

% Create UITable_12
app.UITable_12 = uitable(app.GridLayout2_2);
app.UITable_12.ColumnName = '';
app.UITable_12.RowName = {};
app.UITable_12.SelectionType = 'column';
app.UITable_12.CellSelectionCallback = createCallbackFcn(app, @UITable_7CellSelection, true);
app.UITable_12.Multiselect = 'off';
app.UITable_12.Layout.Row = 3;
app.UITable_12.Layout.Column = [2 7];

% Create LoadFileButton_2
app.LoadFileButton_2 = uibutton(app.GridLayout2_2, 'push');
app.LoadFileButton_2.ButtonPushedFcn = createCallbackFcn(app, @LoadFileButtonPushed, true);
app.LoadFileButton_2.FontSize = 16;
app.LoadFileButton_2.Layout.Row = 4;
app.LoadFileButton_2.Layout.Column = 2;
app.LoadFileButton_2.Text = 'Load File';

% Create ApplytostageButton_2
app.ApplytostageButton_2 = uibutton(app.GridLayout2_2, 'push');
app.ApplytostageButton_2.ButtonPushedFcn = createCallbackFcn(app, @ApplytostageButton_2Pushed, true);
app.ApplytostageButton_2.FontSize = 16;
app.ApplytostageButton_2.Layout.Row = 4;
app.ApplytostageButton_2.Layout.Column = 5;
app.ApplytostageButton_2.Text = 'Apply to stage';

% Create SetstageSpinnerLabel_2
app.SetstageSpinnerLabel_2 = uilabel(app.GridLayout2_2);
app.SetstageSpinnerLabel_2.HorizontalAlignment = 'right';
app.SetstageSpinnerLabel_2.FontSize = 16;
app.SetstageSpinnerLabel_2.Layout.Row = 4;
app.SetstageSpinnerLabel_2.Layout.Column = 3;
app.SetstageSpinnerLabel_2.Text = 'Set stage';

```



```

% Create SetstageSpinner_2
app.SetstageSpinner_2 = uispinner(app.GridLayout2_2);
app.SetstageSpinner_2.Limits = [1 Inf];
app.SetstageSpinner_2.ValueChangedFcn = createCallbackFcn(app, @SetstageSpinnerValueChanged, true);
app.SetstageSpinner_2.FontSize = 16;
app.SetstageSpinner_2.Layout.Row = 4;
app.SetstageSpinner_2.Layout.Column = 4;
app.SetstageSpinner_2.Value = 1;

% Create ClearstagesButton_2
app.ClearstagesButton_2 = uibutton(app.GridLayout2_2, 'push');
app.ClearstagesButton_2.ButtonPushedFcn = createCallbackFcn(app, @ClearstagesButton_2Pushed, true);
app.ClearstagesButton_2.FontSize = 16;
app.ClearstagesButton_2.Layout.Row = 4;
app.ClearstagesButton_2.Layout.Column = 6;
app.ClearstagesButton_2.Text = 'Clear stages';

% Create NextstepButton_4
app.NextstepButton_4 = uibutton(app.GridLayout2_2, 'push');
app.NextstepButton_4.ButtonPushedFcn = createCallbackFcn(app, @NextstepButton_4Pushed, true);
app.NextstepButton_4.FontSize = 16;
app.NextstepButton_4.Layout.Row = 4;
app.NextstepButton_4.Layout.Column = 7;
app.NextstepButton_4.Text = 'Next step';

% Create XButton_6
app.XButton_6 = uibutton(app.GridLayout2_2, 'push');
app.XButton_6.ButtonPushedFcn = createCallbackFcn(app, @XButtonPushed, true);
app.XButton_6.IconAlignment = 'center';
app.XButton_6.BackgroundColor = [1 0 0];
app.XButton_6.FontName = 'Arial';
app.XButton_6.FontSize = 26;
app.XButton_6.FontWeight = 'bold';
app.XButton_6.FontColor = [1 1 1];
app.XButton_6.Layout.Row = 1;
app.XButton_6.Layout.Column = 8;
app.XButton_6.Text = 'X';

% Create SetinitialparametersTab_2
app.SetinitialparametersTab_2 = uitab(app.TabGroup);
app.SetinitialparametersTab_2.Title = 'Set initial parameters';
app.SetinitialparametersTab_2.ButtonDownFcn = createCallbackFcn(app, @SetinitialparametersTab_2ButtonDown, true);

% Create GridLayout_2
app.GridLayout_2 = uigridlayout(app.SetinitialparametersTab_2);
app.GridLayout_2.ColumnWidth = {40, 60, '0.7x', 120, 25, 120, '1x', 40};
app.GridLayout_2.RowHeight = {40, 40, 27, 27, 27, 27, 161, '2.65x', 27, 50, 26};
app.GridLayout_2.ColumnSpacing = 1.7;
app.GridLayout_2.Padding = [1.7 10 1.7 10];

% Create UITable_8
app.UITable_8 = uitable(app.GridLayout_2);
app.UITable_8.ColumnName = '';
app.UITable_8.RowName = {};
app.UITable_8.ColumnEditable = true;
app.UITable_8.CellEditCallback = createCallbackFcn(app, @UITableCellEdit, true);
app.UITable_8.Multiselect = 'off';
app.UITable_8.BusyAction = 'cancel';
app.UITable_8.Layout.Row = [5 9];
app.UITable_8.Layout.Column = [2 4];
app.UITable_8.FontSize = 14;

% Create InitialparameterselectorLabel
app.InitialparameterselectorLabel = uilabel(app.GridLayout_2);
app.InitialparameterselectorLabel.HorizontalAlignment = 'center';
app.InitialparameterselectorLabel.FontSize = 24;
app.InitialparameterselectorLabel.Layout.Row = 1;
app.InitialparameterselectorLabel.Layout.Column = [1 8];
app.InitialparameterselectorLabel.Text = 'Initial parameter selector';

% Create FillinthesetofdesiredfrequenciesLabel_2
app.FillinthesetofdesiredfrequenciesLabel_2 = uilabel(app.GridLayout_2);
app.FillinthesetofdesiredfrequenciesLabel_2.FontSize = 18;
app.FillinthesetofdesiredfrequenciesLabel_2.Layout.Row = 4;
app.FillinthesetofdesiredfrequenciesLabel_2.Layout.Column = [2 4];
app.FillinthesetofdesiredfrequenciesLabel_2.Text = {'Fill in the set of desired frequencies'; ''};

% Create ScalingfactorHLabel_2
app.ScalingfactorHLabel_2 = uilabel(app.GridLayout_2);
app.ScalingfactorHLabel_2.FontSize = 18;
app.ScalingfactorHLabel_2.Layout.Row = 3;
app.ScalingfactorHLabel_2.Layout.Column = [3 4];
app.ScalingfactorHLabel_2.Text = '< Scaling factor H';

% Create ScalingfactorHEditField_2
app.ScalingfactorHEditField_2 = uieditfield(app.GridLayout_2, 'numeric');
app.ScalingfactorHEditField_2.Limits = [0 Inf];
app.ScalingfactorHEditField_2.ValueChangedFcn = createCallbackFcn(app, @ScalingfactorHEditFieldValueChanged, true);
app.ScalingfactorHEditField_2.BusyAction = 'cancel';
app.ScalingfactorHEditField_2.HorizontalAlignment = 'center';
app.ScalingfactorHEditField_2.FontSize = 18;
app.ScalingfactorHEditField_2.Layout.Row = 3;
app.ScalingfactorHEditField_2.Layout.Column = 2;

% Create UITable_9
app.UITable_9 = uitable(app.GridLayout_2);
app.UITable_9.ColumnName = {'Masses [g]'; 'Springs [N/m]'; 'Dampers [Ns/m]'};
app.UITable_9.RowName = {};
app.UITable_9.Layout.Row = [3 9];
app.UITable_9.Layout.Column = [6 7];

% Create CalculateButton_3
app.CalculateButton_3 = uibutton(app.GridLayout_2, 'push');

```

```

app.CalculateButton_3.ButtonPushedFcn = createCallbackFcn(app, @CalculateButton_3Pushed, true);
app.CalculateButton_3.FontSize = 16;
app.CalculateButton_3.Layout.Row = 10;
app.CalculateButton_3.Layout.Column = 4;
app.CalculateButton_3.Text = 'Calculate';

% Create NextStepButton
app.NextStepButton = uibutton(app.GridLayout_2, 'push');
app.NextStepButton.ButtonPushedFcn = createCallbackFcn(app, @NextStepButtonPushed, true);
app.NextStepButton.FontSize = 16;
app.NextStepButton.Layout.Row = 10;
app.NextStepButton.Layout.Column = 6;
app.NextStepButton.Text = 'Next Step';

% Create XButton_4
app.XButton_4 = uibutton(app.GridLayout_2, 'push');
app.XButton_4.ButtonPushedFcn = createCallbackFcn(app, @XButtonPushed, true);
app.XButton_4.IconAlignment = 'center';
app.XButton_4.BackgroundColor = [1 0 0];
app.XButton_4.FontName = 'Arial';
app.XButton_4.FontSize = 26;
app.XButton_4.FontWeight = 'bold';
app.XButton_4.FontColor = [1 1 1];
app.XButton_4.Layout.Row = 1;
app.XButton_4.Layout.Column = 8;
app.XButton_4.Text = 'X';

% Create Label_4
app.Label_4 = uilabel(app.GridLayout_2);
app.Label_4.HorizontalAlignment = 'center';
app.Label_4.FontSize = 40;
app.Label_4.Layout.Row = [5 9];
app.Label_4.Layout.Column = 5;
app.Label_4.Text = '>';

% Create PlatecalculatorTab
app.PlatecalculatorTab = uitab(app.TabGroup);
app.PlatecalculatorTab.SizeChangedFcn = createCallbackFcn(app, @XButtonPushed, true);
app.PlatecalculatorTab.Title = 'Plate calculator';

% Create GridLayout3_2
app.GridLayout3_2 = uigridlayout(app.PlatecalculatorTab);
app.GridLayout3_2.ColumnWidth = {40, '1x', 120, 47, 26, 47, 120, '1x', 40};
app.GridLayout3_2.RowHeight = {40, 40, 90, 80, '1x', 50, 26};

% Create PiezoelectricparametersynthesizerLabel_2
app.PiezoelectricparametersynthesizerLabel_2 = uilabel(app.GridLayout3_2);
app.PiezoelectricparametersynthesizerLabel_2.HorizontalAlignment = 'center';
app.PiezoelectricparametersynthesizerLabel_2.FontSize = 24;
app.PiezoelectricparametersynthesizerLabel_2.Layout.Row = 1;
app.PiezoelectricparametersynthesizerLabel_2.Layout.Column = [1 9];
app.PiezoelectricparametersynthesizerLabel_2.Text = {'Piezoelectric parameter synthesizer'; ''};

% Create BacktobeginningButton_2
app.BacktobeginningButton_2 = uibutton(app.GridLayout3_2, 'push');
app.BacktobeginningButton_2.ButtonPushedFcn = createCallbackFcn(app, @BacktobeginningButton_2Pushed, true);
app.BacktobeginningButton_2.WordWrap = 'on';
app.BacktobeginningButton_2.FontSize = 16;
app.BacktobeginningButton_2.Layout.Row = 6;
app.BacktobeginningButton_2.Layout.Column = 3;
app.BacktobeginningButton_2.Text = 'Back to beginning';

% Create CalculateButton_4
app.CalculateButton_4 = uibutton(app.GridLayout3_2, 'push');
app.CalculateButton_4.ButtonPushedFcn = createCallbackFcn(app, @CalculateButton_4Pushed, true);
app.CalculateButton_4.FontSize = 16;
app.CalculateButton_4.Layout.Row = 6;
app.CalculateButton_4.Layout.Column = [4 6];
app.CalculateButton_4.Text = 'Calculate';

% Create AnalysemodelButton
app.AnalysemodelButton = uibutton(app.GridLayout3_2, 'push');
app.AnalysemodelButton.ButtonPushedFcn = createCallbackFcn(app, @AnalysemodelButton_Pushed, true);
app.AnalysemodelButton.FontSize = 16;
app.AnalysemodelButton.Layout.Row = 6;
app.AnalysemodelButton.Layout.Column = 7;
app.AnalysemodelButton.Text = 'Analyse model';

% Create UITable_10
app.UITable_10 = uitable(app.GridLayout3_2);
app.UITable_10.ColumnName = '';
app.UITable_10.RowName = {};
app.UITable_10.Layout.Row = [3 5];
app.UITable_10.Layout.Column = [6 8];

% Create UITable_11
app.UITable_11 = uitable(app.GridLayout3_2);
app.UITable_11.ColumnName = '';
app.UITable_11.RowName = {};
app.UITable_11.ColumnEditable = true;
app.UITable_11.CellEditCallback = createCallbackFcn(app, @UITable_11CellEdit, true);
app.UITable_11.Multiselect = 'off';
app.UITable_11.BusyAction = 'cancel';
app.UITable_11.Layout.Row = 5;
app.UITable_11.Layout.Column = [2 4];
app.UITable_11.FontSize = 14;

% Create Label_3
app.Label_3 = uilabel(app.GridLayout3_2);
app.Label_3.HorizontalAlignment = 'center';
app.Label_3.FontSize = 40;
app.Label_3.Layout.Row = [3 5];
app.Label_3.Layout.Column = 5;
app.Label_3.Text = '>';

```

```

% Create XButton_5
app.XButton_5 = uibutton(app.GridLayout3_2, 'push');
app.XButton_5.ButtonPushedFcn = createCallbackFcn(app, @XButtonPushed, true);
app.XButton_5.IconAlignment = 'center';
app.XButton_5.BackgroundColor = [1 0 0];
app.XButton_5.FontName = 'Arial';
app.XButton_5.FontSize = 26;
app.XButton_5.FontWeight = 'bold';
app.XButton_5.FontColor = [1 1 1];
app.XButton_5.Layout.Row = 1;
app.XButton_5.Layout.Column = 9;
app.XButton_5.Text = 'X';

% Create SpecifyplatebasetypeButtonGroup
app.SpecifyplatebasetypeButtonGroup = uibuttongroup(app.GridLayout3_2);
app.SpecifyplatebasetypeButtonGroup.SelectionChangedFcn = createCallbackFcn(app, @SpecifyplatebasetypeButtonGroupSelectionChanged,
app.SpecifyplatebasetypeButtonGroup.Title = 'Specify plate base type';
app.SpecifyplatebasetypeButtonGroup.BackgroundColor = [1 1 1];
app.SpecifyplatebasetypeButtonGroup.Layout.Row = 3;
app.SpecifyplatebasetypeButtonGroup.Layout.Column = [2 4];
app.SpecifyplatebasetypeButtonGroup.FontSize = 16;

% Create RoundbaseButton
app.RoundbaseButton = uiradiobutton(app.SpecifyplatebasetypeButtonGroup);
app.RoundbaseButton.Tag = '1';
app.RoundbaseButton.Text = 'Round base';
app.RoundbaseButton.FontSize = 16;
app.RoundbaseButton.Position = [11 38 108 22];
app.RoundbaseButton.Value = true;

% Create SquarebaseButton
app.SquarebaseButton = uiradiobutton(app.SpecifyplatebasetypeButtonGroup);
app.SquarebaseButton.Tag = '2';
app.SquarebaseButton.Text = 'Square base';
app.SquarebaseButton.FontSize = 16;
app.SquarebaseButton.Position = [11 8 113 22];

% Create RectangularbaseButton
app.RectangularbaseButton = uiradiobutton(app.SpecifyplatebasetypeButtonGroup);
app.RectangularbaseButton.Tag = '3';
app.RectangularbaseButton.Text = 'Rectangular base';
app.RectangularbaseButton.FontSize = 16;
app.RectangularbaseButton.Position = [151 38 147 22];

% Create SpecifycustomdimensionsButtonGroup
app.SpecifycustomdimensionsButtonGroup = uibuttongroup(app.GridLayout3_2);
app.SpecifycustomdimensionsButtonGroup.SelectionChangedFcn = createCallbackFcn(app, @SpecifycustomdimensionsButtonGroupSelectionCf
app.SpecifycustomdimensionsButtonGroup.Title = 'Specify custom dimensions';
app.SpecifycustomdimensionsButtonGroup.BackgroundColor = [1 1 1];
app.SpecifycustomdimensionsButtonGroup.Layout.Row = 4;
app.SpecifycustomdimensionsButtonGroup.Layout.Column = [2 4];
app.SpecifycustomdimensionsButtonGroup.FontSize = 16;

% Create CustomsurfaceareaButton
app.CustomsurfaceareaButton = uiradiobutton(app.SpecifycustomdimensionsButtonGroup);
app.CustomsurfaceareaButton.Tag = '1';
app.CustomsurfaceareaButton.Text = 'Custom surface area';
app.CustomsurfaceareaButton.FontSize = 16;
app.CustomsurfaceareaButton.Position = [11 30 171 22];
app.CustomsurfaceareaButton.Value = true;

% Create CustomthicknessButton
app.CustomthicknessButton = uiradiobutton(app.SpecifycustomdimensionsButtonGroup);
app.CustomthicknessButton.Tag = '2';
app.CustomthicknessButton.Text = 'Custom thickness';
app.CustomthicknessButton.FontSize = 16;
app.CustomthicknessButton.Position = [11 4 148 22];

% Create ModelanalyzerTab
app.ModelanalyzerTab = uitab(app.TabGroup);
app.ModelanalyzerTab.Title = 'Model analyzer';

% Create GridLayout5
app.GridLayout5 = uigridlayout(app.ModelanalyzerTab);
app.GridLayout5.ColumnWidth = {40, '1x', 20, '1x', 20, '1x', 80, 40, 40};
app.GridLayout5.RowHeight = {40, 40, '1x', '1x', '1x', '1x', '1x', '1x', '1x', 50, 26};

% Create ModelanalyzerLabel
app.ModelanalyzerLabel = uilabel(app.GridLayout5);
app.ModelanalyzerLabel.HorizontalAlignment = 'center';
app.ModelanalyzerLabel.FontSize = 24;
app.ModelanalyzerLabel.Layout.Row = 1;
app.ModelanalyzerLabel.Layout.Column = [1 9];
app.ModelanalyzerLabel.Text = 'Model analyzer';

% Create Image
app.Image = uiimage(app.GridLayout5);
app.Image.BackgroundColor = [1 1 1];
app.Image.Layout.Row = 10;
app.Image.Layout.Column = [2 8];

% Create XButton_7
app.XButton_7 = uibutton(app.GridLayout5, 'push');
app.XButton_7.ButtonPushedFcn = createCallbackFcn(app, @XButtonPushed, true);
app.XButton_7.IconAlignment = 'center';
app.XButton_7.BackgroundColor = [1 0 0];
app.XButton_7.FontName = 'Arial';
app.XButton_7.FontSize = 26;
app.XButton_7.FontWeight = 'bold';
app.XButton_7.FontColor = [1 1 1];
app.XButton_7.Layout.Row = 1;
app.XButton_7.Layout.Column = 9;
app.XButton_7.Text = 'X';

```

```

% Create BackButton
app.BackButton = uibutton(app.GridLayout5, 'push');
app.BackButton.ButtonPushedFcn = createCallbackFcn(app, @BackButtonPushed, true);
app.BackButton.Icon = 'arrow.png';
app.BackButton.IconAlignment = 'center';
app.BackButton.BackgroundColor = [0 0.4431 0.7412];
app.BackButton.FontName = 'Arial';
app.BackButton.FontSize = 26;
app.BackButton.FontWeight = 'bold';
app.BackButton.FontColor = [1 1 1];
app.BackButton.Layout.Row = 1;
app.BackButton.Layout.Column = 8;
app.BackButton.Text = '';

% Create UITable2
app.UITable2 = uitable(app.GridLayout5);
app.UITable2.ColumnName = '';
app.UITable2.RowName = {};
app.UITable2.CellEditCallback = createCallbackFcn(app, @EditPlateDetails, true);
app.UITable2.Layout.Row = [3 9];
app.UITable2.Layout.Column = [2 6];

% Create ShowgraphsButton
app.ShowgraphsButton = uibutton(app.GridLayout5, 'push');
app.ShowgraphsButton.ButtonPushedFcn = createCallbackFcn(app, @ShowgraphsButtonPushed, true);
app.ShowgraphsButton.FontSize = 16;
app.ShowgraphsButton.Layout.Row = 9;
app.ShowgraphsButton.Layout.Column = [7 8];
app.ShowgraphsButton.Text = 'Show graphs';

% Create SetaccuracyButton
app.SetaccuracyButton = uibutton(app.GridLayout5, 'push');
app.SetaccuracyButton.ButtonPushedFcn = createCallbackFcn(app, @SetaccuracyButtonPushed, true);
app.SetaccuracyButton.FontSize = 16;
app.SetaccuracyButton.Layout.Row = 4;
app.SetaccuracyButton.Layout.Column = [7 8];
app.SetaccuracyButton.Text = 'Set accuracy';

% Create SetfreqrangeButton
app.SetfreqrangeButton = uibutton(app.GridLayout5, 'push');
app.SetfreqrangeButton.ButtonPushedFcn = createCallbackFcn(app, @SetfreqrangeButtonPushed, true);
app.SetfreqrangeButton.FontSize = 16;
app.SetfreqrangeButton.Layout.Row = 3;
app.SetfreqrangeButton.Layout.Column = [7 8];
app.SetfreqrangeButton.Text = 'Set freq range';

% Create Image_2
app.Image_2 = uiimage(app.GridLayout5);
app.Image_2.BackgroundColor = [0.3098 0.6902 1];
app.Image_2.Layout.Row = 10;
app.Image_2.Layout.Column = [2 4];

% Create ProgressinfoLabel
app.ProgressinfoLabel = uilabel(app.GridLayout5);
app.ProgressinfoLabel.HorizontalAlignment = 'center';
app.ProgressinfoLabel.FontSize = 20;
app.ProgressinfoLabel.Layout.Row = 10;
app.ProgressinfoLabel.Layout.Column = [2 8];
app.ProgressinfoLabel.Text = 'Progress info';

% Create AnalyseButton
app.AnalyseButton = uibutton(app.GridLayout5, 'push');
app.AnalyseButton.ButtonPushedFcn = createCallbackFcn(app, @AnalyseButtonPushed, true);
app.AnalyseButton.FontSize = 24;
app.AnalyseButton.Layout.Row = 10;
app.AnalyseButton.Layout.Column = [2 8];
app.AnalyseButton.Text = 'Analyse';

% Create SetexternalforceButton
app.SetexternalforceButton = uibutton(app.GridLayout5, 'push');
app.SetexternalforceButton.ButtonPushedFcn = createCallbackFcn(app, @SetexternalforceButtonPushed, true);
app.SetexternalforceButton.WordWrap = 'on';
app.SetexternalforceButton.FontSize = 16;
app.SetexternalforceButton.Layout.Row = 5;
app.SetexternalforceButton.Layout.Column = [7 8];
app.SetexternalforceButton.Text = 'Set external force';

% Create SetexternalvoltageButton
app.SetexternalvoltageButton = uibutton(app.GridLayout5, 'push');
app.SetexternalvoltageButton.ButtonPushedFcn = createCallbackFcn(app, @SetexternalvoltageButtonPushed, true);
app.SetexternalvoltageButton.WordWrap = 'on';
app.SetexternalvoltageButton.FontSize = 16;
app.SetexternalvoltageButton.Layout.Row = 7;
app.SetexternalvoltageButton.Layout.Column = [7 8];
app.SetexternalvoltageButton.Text = 'Set external voltage';

% Create OffsetButtonGroup
app.OffsetButtonGroup = uibuttongroup(app.GridLayout5);
app.OffsetButtonGroup.SelectionChangedFcn = createCallbackFcn(app, @OffsetButtonGroupSelectionChanged, true);
app.OffsetButtonGroup.TitlePosition = 'centertop';
app.OffsetButtonGroup.Title = 'Offset';
app.OffsetButtonGroup.BackgroundColor = [1 1 1];
app.OffsetButtonGroup.Layout.Row = 6;
app.OffsetButtonGroup.Layout.Column = [7 8];
app.OffsetButtonGroup.FontSize = 16;

% Create sinButton
app.sinButton = uiradiobutton(app.OffsetButtonGroup);
app.sinButton.Tag = '1';
app.sinButton.Text = 'sin';
app.sinButton.FontSize = 16;
app.sinButton.Position = [10 2 42 22];
app.sinButton.Value = true;

```

```

% Create cosButton
app.cosButton = uiradiobutton(app.OffsetButtonGroup);
app.cosButton.Tag = '3';
app.cosButton.Text = 'cos';
app.cosButton.FontSize = 16;
app.cosButton.Position = [72 2 47 22];

% Create OffsetButtonGroup_2
app.OffsetButtonGroup_2 = uibuttongroup(app.GridLayout5);
app.OffsetButtonGroup_2.SelectionChangedFcn = createCallbackFcn(app, @OffsetButtonGroup_2SelectionChanged, true);
app.OffsetButtonGroup_2.TitlePosition = 'centertop';
app.OffsetButtonGroup_2.Title = 'Offset';
app.OffsetButtonGroup_2.BackgroundColor = [1 1 1];
app.OffsetButtonGroup_2.Layout.Row = 8;
app.OffsetButtonGroup_2.Layout.Column = [7 8];
app.OffsetButtonGroup_2.FontSize = 16;

% Create sinButton_2
app.sinButton_2 = uiradiobutton(app.OffsetButtonGroup_2);
app.sinButton_2.Tag = '1';
app.sinButton_2.Text = 'sin';
app.sinButton_2.FontSize = 16;
app.sinButton_2.Position = [10 2 42 22];
app.sinButton_2.Value = true;

% Create cosButton_2
app.cosButton_2 = uiradiobutton(app.OffsetButtonGroup_2);
app.cosButton_2.Tag = '3';
app.cosButton_2.Text = 'cos';
app.cosButton_2.FontSize = 16;
app.cosButton_2.Position = [72 2 47 22];

% Create CancelAnalysisTab
app.CancelAnalysisTab = uitab(app.TabGroup);
app.CancelAnalysisTab.Title = 'Cancel Analysis';

% Create CanceltheanalysisLabel
app.CanceltheanalysisLabel = uilabel(app.CancelAnalysisTab);
app.CanceltheanalysisLabel.HorizontalAlignment = 'center';
app.CanceltheanalysisLabel.FontSize = 36;
app.CanceltheanalysisLabel.Position = [1 370 795 52];
app.CanceltheanalysisLabel.Text = 'Cancel the analysis';

% Create RestartthesynthesisLabel_3
app.RestartthesynthesisLabel_3 = uilabel(app.CancelAnalysisTab);
app.RestartthesynthesisLabel_3.HorizontalAlignment = 'center';
app.RestartthesynthesisLabel_3.WordWrap = 'on';
app.RestartthesynthesisLabel_3.FontSize = 18;
app.RestartthesynthesisLabel_3.Position = [2 286 794 85];
app.RestartthesynthesisLabel_3.Text = {'You are about to cancel the existing model analysis process. '; 'All the data will be rese

% Create ProceedButton_2
app.ProceedButton_2 = uibutton(app.CancelAnalysisTab, 'push');
app.ProceedButton_2.ButtonPushedFcn = createCallbackFcn(app, @ProceedButton_2Pushed, true);
app.ProceedButton_2.FontSize = 16;
app.ProceedButton_2.Position = [257 226 120 50];
app.ProceedButton_2.Text = 'Proceed';

% Create CancelButton_2
app.CancelButton_2 = uibutton(app.CancelAnalysisTab, 'push');
app.CancelButton_2.ButtonPushedFcn = createCallbackFcn(app, @CancelButton_2Pushed, true);
app.CancelButton_2.FontSize = 16;
app.CancelButton_2.Position = [417 226 120 50];
app.CancelButton_2.Text = 'Cancel';

% Create ResetTab
app.ResetTab = uitab(app.TabGroup);
app.ResetTab.Title = 'Reset';

% Create CancelthesynthesisLabel
app.CancelthesynthesisLabel = uilabel(app.ResetTab);
app.CancelthesynthesisLabel.HorizontalAlignment = 'center';
app.CancelthesynthesisLabel.FontSize = 36;
app.CancelthesynthesisLabel.Position = [1 370 795 52];
app.CancelthesynthesisLabel.Text = 'Cancel the synthesis';

% Create RestartthesynthesisLabel_2
app.RestartthesynthesisLabel_2 = uilabel(app.ResetTab);
app.RestartthesynthesisLabel_2.HorizontalAlignment = 'center';
app.RestartthesynthesisLabel_2.WordWrap = 'on';
app.RestartthesynthesisLabel_2.FontSize = 18;
app.RestartthesynthesisLabel_2.Position = [2 286 794 85];
app.RestartthesynthesisLabel_2.Text = {'You are about to cancel the existing synthesis process. '; 'All the data will be reset anc

% Create ProceedButton
app.ProceedButton = uibutton(app.ResetTab, 'push');
app.ProceedButton.ButtonPushedFcn = createCallbackFcn(app, @ProceedButtonPushed, true);
app.ProceedButton.FontSize = 16;
app.ProceedButton.Position = [257 226 120 50];
app.ProceedButton.Text = 'Proceed';

% Create CancelButton
app.CancelButton = uibutton(app.ResetTab, 'push');
app.CancelButton.ButtonPushedFcn = createCallbackFcn(app, @CancelButtonPushed, true);
app.CancelButton.FontSize = 16;
app.CancelButton.Position = [417 226 120 50];
app.CancelButton.Text = 'Cancel';

% Show the figure after all components are created
app.UIFigure.Visible = 'on';
end
end

```

```
% App creation and deletion
methods (Access = public)

% Construct app
function app = Piezo_app

    % Create UIFigure and components
    createComponents(app)

    % Register the app with App Designer
    registerApp(app, app.UIFigure)

    % Execute the startup function
    runStartupFcn(app, @startupFcn)

    if nargin == 0
        clear app
    end
end

% Code that executes before app deletion
function delete(app)

    % Delete UIFigure when app is deleted
    delete(app.UIFigure)
end
end
end
```

```

datapath{1} = dir('C:\Users\Jacek\Desktop\Badanie_przy_1V\1_Plytka10x10x2\*.csv');
datapath{2} = dir('C:\Users\Jacek\Desktop\Badanie_przy_1V\2_Plytka10x10x2\*.csv');
datapath{3} = dir('C:\Users\Jacek\Desktop\Badanie_przy_1V\Plytka5x5x2\*.csv');
datapath{4} = dir('C:\Users\Jacek\Desktop\Badanie_przy_1V\Stos_dwie_rozne\*.csv');
%datapath{5} = dir('C:\Users\Jacek\Desktop\POWTORKA\Dwie_plytki_rozne_0-1V\*.csv');

name{1} = 'Pierwsza płytką 10x10x2mm 1V';
name{2} = 'Druga płytką 10x10x2mm 1V';
name{3} = 'Płytką 5x5x2mm 1V';
%name{5} = 'Stos dwóch płytek 10x10x2mm 1V';
name{4} = 'Stos różnych płytek 1V';

for i=1:length(name)
    filepaths{i} = fullfile({datapath{i}.folder},{datapath{i}.name});
    extended_names{i} = fullfile({datapath{i}.name});
    rawnames{i} = str2double(erase(extended_names{i}, ".csv"));
    rawnames{i} = rawnames{i}.*1000;
end

for i=1:length(name)
    for j=1:length(filepaths{i})
        measurements_no(i) = length(filepaths{i});
        data{i}{j} = csvread(filepaths{i}{j},2,0);
        datatime{i}{j} = data{i}{j}(:,1);
        datavalues{i}{j} = data{i}{j}(:,2);
        data{i}{j} = [];
        fftdata{i}{j} = fft(datavalues{i}{j});
        samplenum{i}{j} = length(datavalues{i}{j});
        samplefreq{i}{j} = mean(diff(datatime{i}{j}));
        frequency{i}{j} = (0:samplenum{i}{j}-1) * 1/samplefreq{i}{j}/samplenum{i}{j};
        fftdata_normalised{i}{j} = fftdata{i}{j}/samplenum{i}{j};
        fftdata_normalised{i}{j} = fftdata_normalised{i}{j}(1:length(fftdata_normalised{i}{j}));
        fftdata_normalised{i}{j} = abs(fftdata_normalised{i}{j}).*50;
        frequency{i}{j} = frequency{i}{j}(1:length(frequency{i}{j})/2);
        differences{i}{j} = abs(frequency{i}{j}-rawnames{i}(j));
        [placeholder regfreqidx{i}{j}] = min(differences{i}{j});
        regfreq{i}(j) = frequency{i}{j}(regfreqidx{i}{j});
        regamp{i}(j) = fftdata_normalised{i}{j}(regfreqidx{i}{j});
    end
end

for i=1:length(name)
    figure
    plot(regfreq{i},regamp{i},'+')
    xlim([0 400000])
    ylim([0 100])
    title(name{i})
    xlabel('Częstotliwość [Hz]')
    ylabel('Amplituda odkształcenia [nm]')
    saveMX = cat(2,regfreq{i}.,regamp{i}.');
    filename1 = 'Wynik_badania_';
    filename2 = num2str(i);
    filename = [filename1 filename2 '.xls'];
    writematrix(saveMX, filename);
end

```