Software Listing

The Graphical User Interface (GUI) and Fuzzy Inference System (FIS) are developed in MATLAB. Software code is developed for different sections like Weaving section, Motor Status, Environment, Emergency Faults and Oil tank condition. Some software code listing is in following sections.

- GUI code for different sections is in following sections

```matlab
function varargout = Weaving_Section_1(varargin)
% WEAVING_SECTION_1 M-file for Weaving_Section_1.fig

    gui_Singleton = 1;
    gui_State = struct('gui_Name', mfilename, ...
        'gui_Singleton', gui_Singleton, ...
        'gui_OpeningFcn', @Weaving_Section_1_OpeningFcn, ...
        'gui_OutputFcn', @Weaving_Section_1_OutputFcn, ...
        'gui_LayoutFcn', [], ...
        'gui_Callback', []);

    if nargin && ischar(varargin{1})
        gui_State.gui_Callback = str2func(varargin{1});
    end

    if nargout
        [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
    else
        gui_mainfcn(gui_State, varargin{:});
    end

% End initialization code - DO NOT EDIT

% --- Executes just before Weaving_Section_1 is made visible.
function Weaving_Section_1_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OpeningFcn.
% hObject    handle to figure
% eventdata   reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to Weaving_Section_1 (see VARARGIN)

% Choose default command line output for Weaving_Section_1
handles.output = hObject;

% Update handles structure
guida(hObject, handles);
```
% UIWAIT makes Weaving_Section_1 wait for user response (see \texttt{UIRESUME}).
% \texttt{uiwait(handles.figure1)};

% --- Outputs from this function are returned to the command line.
\textbf{function} \texttt{varargout = Weaving\_Section\_1\_OutputFcn(hObject, eventdata, handles)}
% \texttt{varargout} cell array for returning output args (see \texttt{VARARGOUT});
% \texttt{ hObject} handle to figure
% \texttt{ eventdata} reserved - to be defined in a future version of MATLAB
% \texttt{ handles} structure with handles and user data (see \texttt{GUIDATA})

% Get default command line output from handles structure
\texttt{varargout{1} = handles.output};

% --- Executes on button press in environment\_button.
\textbf{function} \texttt{environment\_button\_Callback(hObject, eventdata, handles)}
% \texttt{(hObject} handle to environment\_button (see \texttt{ GCBO})
% \texttt{ eventdata} reserved - to be defined in a future version of MATLAB
% \texttt{ handles} structure with handles and user data (see \texttt{ GUIDATA})
%--------------------------------------------------------------
\texttt{Environment\_ConditionFigureHandle = Environment\_Condition;} %stores
the figure handle of Quan's GUI here
\texttt{Environment\_ConditionData = guidata(Environment\_ConditionFigureHandle);} %stores the GUI data from Quan's GUI here
\texttt{Environment\_Condition\_input = get(Environment\_ConditionData.state\_environment,'String');} %now we can access any of the data from Quan's GUI!!!!
\texttt{Environment\_Condition\_input = get(Environment\_ConditionData.state\_environment,'String');}

%set the static text on Daniel's GUI to match the
%input text from Quan's GUI
\texttt{set(handles.state\_environment,'String',Environment\_Condition\_input)};

%notice that quanData is the structure containing the data from
Quan's GUI
%notice that handles is the structure containing data from Daniel's
GUI,
%which is the local GUI (i.e., the GUI that this function is running from)
%--------------------------------------------------------------
% --- Executes on button press in pushbutton2.
\textbf{function} \texttt{pushbutton2\_Callback(hObject, eventdata, handles)}
% \texttt{ hObject} handle to pushbutton2 (see \texttt{ GCBO})
% \texttt{ eventdata} reserved - to be defined in a future version of MATLAB
% \texttt{ handles} structure with handles and user data (see \texttt{ GUIDATA})
%--------------------------------------------------------------
\texttt{motor\_conditionFigureHandle = motor\_condition;} %stores the figure
handle of Quan's GUI here
\texttt{motor\_conditionData = guidata(motor\_conditionFigureHandle)};
\texttt{motor\_condition\_input = get(motor\_conditionData.state\_motor,'String');}
set(handles.state_motor,'String',motor_condition_input);
%--------------------------------------------------------------
% --- Executes on button press in oil_tank_button.
function oil_tank_button_Callback(hObject, eventdata, handles)
% hObject    handle to oil_tank_button (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
%--------------------------------------------------------------
Oil_TankFigureHandle = Oil_Tank;  %stores the figure handle of
Quan's GUI here
%stores the GUI data from Quan's GUI here
%now we can access any of the data from Quan's GUI!!!
Oil_TankData = guidata(Oil_TankFigureHandle);
%store the input text from Quan's GUI
%into the variable quan_input
Oil_TankData_input = get(Oil_TankData.state_oil,'String');
set(handles.state_oil,'String',Oil_TankData_input);
%--------------------------------------------------------------
% --- Executes on button press in pushbutton4.
function pushbutton4_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% --- Executes on button press in motherboard_button.
function motherboard_button_Callback(hObject, eventdata, handles)
% hObject    handle to motherboard_button (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
function varargout = Environment_Condition(varargin)  
% ENVIRONMENT_CONDITION M-file for Environment_Condition.fig  
% Begin initialization code - DO NOT EDIT  
gui_Singleton = 1;  
gui_State = struct('gui_Name', mfilename, ...  
    'gui_Singleton', gui_Singleton, ...  
    'gui_OpeningFcn',  
    @Environment_Condition_OpeningFcn, ...  
    'gui_OutputFcn',  
    @Environment_Condition_OutputFcn, ...  
    'gui_LayoutFcn', [], ...  
    'gui_Callback', []);  
if nargin && ischar(varargin{1})  
gui_State.gui_Callback = str2func(varargin{1});  
end  
if nargout  
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});  
else  
    gui_mainfcn(gui_State, varargin{:});  
end  
% End initialization code - DO NOT EDIT  

% --- Executes just before Environment_Condition is made visible.  
function Environment_Condition_OpeningFcn(hObject, eventdata, handles, varargin)  
% This function has no output args, see OutputFcn.  
% hObject    handle to figure  
% eventdata  reserved - to be defined in a future version of MATLAB  
% handles    structure with handles and user data (see GUIDATA)  
% varargin   command line arguments to Environment_Condition (see VARARGIN)  

% Choose default command line output for Environment_Condition  
handles.output = hObject;  

% Update handles structure  
guidata(hObject, handles);  

% UIWAIT makes Environment_Condition wait for user response (see UIRESUME)  
% uiwait(handles.figure1);  

% --- Outputs from this function are returned to the command line.  
function varargout = Environment_Condition_OutputFcn(hObject, eventdata, handles)  
% varargout    cell array for returning output args (see VARARGOUT);  
% hObject      handle to figure  
% eventdata    reserved - to be defined in a future version of MATLAB  
% handles      structure with handles and user data (see GUIDATA)
% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
environment = evalin('base','environment')
a = environment(1,5)

switch (a)
    case 1
        set(handles.state_environment,'String','Bad');
        set(handles.causes_environment,'visible', 'on');
        set(handles.possible_envir,'visible', 'on');
        set(handles.causes_environment,'String','Warning !!! Please check Sprinkler and exaust fan also');
    case 2
        set(handles.state_environment,'String','Medium');
        set(handles.causes_environment,'visible', 'on');
        set(handles.possible_envir,'visible', 'on');
        set(handles.causes_environment,'String','Please check Exaust fan');
    case 3
        set(handles.state_environment,'String','Good');
        set(handles.causes_environment,'visible', 'off');
        set(handles.possible_envir,'visible', 'off');
    otherwise
        set(handles.state_environment,'String','--');
        set(handles.causes_environment,'visible', 'on');
        set(handles.possible_envir,'visible', 'on');
        set(handles.causes_environment,'String','');
end

a = environment(1,2)
set(handles.temperature,'String',a);

a = environment(2,2)
set(handles.humidity,'String',a);
```matlab
function varargout = Motor_Condition(varargin)

% MOTOR_CONDITION M-file for Motor_Condition.fig
%      MOTOR_CONDITION, by itself, creates a new MOTOR_CONDITION or
% raises the existing
%      singleton*.

% gui_Singleton = 1;
% gui_State = struct('gui_Name',       mfilename, ...
%   'gui.Singleton',   gui.Singleton, ...
%   'gui_OpeningFcn', @Motor_Condition_OpeningFcn, ...
%   'gui_OutputFcn',  @Motor_Condition_OutputFcn, ...
%   'gui_LayoutFcn',  [],    ...
%   'gui_Callback',   []);

if nargin && ischar(varargin{1})
   gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
   [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
   gui_mainfcn(gui_State, varargin{:});
end

% End initialization code - DO NOT EDIT

% --- Executes just before Motor_Condition is made visible.
function Motor_Condition_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to Motor_Condition (see VARARGIN)

% Choose default command line output for Motor_Condition
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes Motor_Condition wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = Motor_Condition_OutputFcn(hObject, eventdata, handles)
% varargout    cell array for returning output args (see VARARGOUT);
% hObject      handle to figure
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;
```
function edit1_CreateFcn(hObject, eventdata, handles)
    % hObject    handle to edit1 (see GCBO)
    % eventdata  reserved - to be defined in a future version of MATLAB
    % handles    empty - handles not created until after all CreateFcns called

    function causes_motor_Callback(hObject, eventdata, handles)
        % hObject    handle to causes_motor (see GCBO)
        % eventdata  reserved - to be defined in a future version of MATLAB
        % handles    structure with handles and user data (see GUIDATA)

        % Hints: get(hObject,'String') returns contents of causes_motor as text
        % str2double(get(hObject,'String')) returns contents of causes_motor as a double

        % --- Executes during object creation, after setting all properties.
        function causes_motor_CreateFcn(hObject, eventdata, handles)
            % hObject    handle to causes_motor (see GCBO)
            % eventdata  reserved - to be defined in a future version of MATLAB
            % handles    empty - handles not created until after all CreateFcns called

            % Hint: edit controls usually have a white background on Windows.
            % See ISPC and COMPUTER.
            if ispc && isequal(get(hObject,'BackgroundColor'),
                            get(0,'defaultUicontrolBackgroundColor'))
                set(hObject,'BackgroundColor','white');
            end

        % --- Executes on button press in pushbutton1.
        function pushbutton1_Callback(hObject, eventdata, handles)
            % hObject    handle to pushbutton1 (see GCBO)
            % eventdata  reserved - to be defined in a future version of MATLAB
            % handles    structure with handles and user data (see GUIDATA)
            motor = evalin('base','motor')
            a = motor(4,5)
            switch a
                case 1
                    set(handles.state_motor,'String','Open Phase');
                    set(handles.causes_motor,'String','Warning !!! Check the phases');
                    set(handles.causes_motor,'visible', 'on');
                    set(handles.possible_motor,'visible', 'on');
                case 2
                    set(handles.state_motor,'String','Damage');
                    set(handles.causes_motor,'String','Warning !!! Motor Damage, Please check load');
                    set(handles.causes_motor,'visible', 'on');

    % --- Executes during object creation, after setting all properties.
    function pushbutton1_CreateFcn(hObject, eventdata, handles)
        % hObject    handle to pushbutton1 (see GCBO)
        % eventdata  reserved - to be defined in a future version of MATLAB
        % handles    empty - handles not created until after all CreateFcns called

        % Hint: edit controls usually have a white background on Windows.
        % See ISPC and COMPUTER.
        if ispc && isequal(get(hObject,'BackgroundColor'),
                        get(0,'defaultUicontrolBackgroundColor'))
            set(hObject,'BackgroundColor','white');
        end

    % --- Executes on button press in pushbutton1.
    function pushbutton1_Callback(hObject, eventdata, handles)
        % hObject    handle to pushbutton1 (see GCBO)
        % eventdata  reserved - to be defined in a future version of MATLAB
        % handles    structure with handles and user data (see GUIDATA)
        motor = evalin('base','motor')
        a = motor(4,5)
        switch a
            case 1
                set(handles.state_motor,'String','Open Phase');
                set(handles.causes_motor,'String','Warning !!! Check the phases');
                set(handles.causes_motor,'visible', 'on');
                set(handles.possible_motor,'visible', 'on');
            case 2
                set(handles.state_motor,'String','Damage');
                set(handles.causes_motor,'String','Warning !!! Motor Damage, Please check load');
                set(handles.causes_motor,'visible', 'on');

    % --- Executes on button press in pushbutton1.
    function pushbutton1_Callback(hObject, eventdata, handles)
        % hObject    handle to pushbutton1 (see GCBO)
        % eventdata  reserved - to be defined in a future version of MATLAB
        % handles    structure with handles and user data (see GUIDATA)
        motor = evalin('base','motor')
        a = motor(4,5)
        switch a
            case 1
                set(handles.state_motor,'String','Open Phase');
                set(handles.causes_motor,'String','Warning !!! Check the phases');
                set(handles.causes_motor,'visible', 'on');
                set(handles.possible_motor,'visible', 'on');
            case 2
                set(handles.state_motor,'String','Damage');
                set(handles.causes_motor,'String','Warning !!! Motor Damage, Please check load');
                set(handles.causes_motor,'visible', 'on');

    % --- Executes during object creation, after setting all properties.
    function causes_motor_CreateFcn(hObject, eventdata, handles)
        % hObject    handle to causes_motor (see GCBO)
        % eventdata  reserved - to be defined in a future version of MATLAB
        % handles    structure with handles and user data (see GUIDATA)

        % Hints: get(hObject,'String') returns contents of causes_motor as text
        % str2double(get(hObject,'String')) returns contents of causes_motor as a double

        % --- Executes during object creation, after setting all properties.
        function causes_motor_CreateFcn(hObject, eventdata, handles)
            % hObject    handle to causes_motor (see GCBO)
            % eventdata  reserved - to be defined in a future version of MATLAB
            % handles    empty - handles not created until after all CreateFcns called

            % Hint: edit controls usually have a white background on Windows.
            % See ISPC and COMPUTER.
            if ispc && isequal(get(hObject,'BackgroundColor'),
                            get(0,'defaultUicontrolBackgroundColor'))
                set(hObject,'BackgroundColor','white');
            end

    % --- Executes on button press in pushbutton1.
    function pushbutton1_Callback(hObject, eventdata, handles)
        % hObject    handle to pushbutton1 (see GCBO)
        % eventdata  reserved - to be defined in a future version of MATLAB
        % handles    structure with handles and user data (see GUIDATA)
        motor = evalin('base','motor')
        a = motor(4,5)
        switch a
            case 1
                set(handles.state_motor,'String','Open Phase');
                set(handles.causes_motor,'String','Warning !!! Check the phases');
                set(handles.causes_motor,'visible', 'on');
                set(handles.possible_motor,'visible', 'on');
            case 2
                set(handles.state_motor,'String','Damage');
                set(handles.causes_motor,'String','Warning !!! Motor Damage, Please check load');
                set(handles.causes_motor,'visible', 'on');
set(handles.possible_motor,'visible', 'on');
case 3
    set(handles.state_motor,'String','Critically Overload');
    set(handles.causes_motor,'String','Warning !!! Load is very heavy, Please check load');
    set(handles.causes_motor,'visible', 'on');
    set(handles.possible_motor,'visible', 'on');
    case 4
    set(handles.state_motor,'String','Overload');
    set(handles.causes_motor,'String','Warning !!! Please check load');
    set(handles.causes_motor,'visible', 'on');
    set(handles.possible_motor,'visible', 'on');
    case 5
    set(handles.state_motor,'String','Good');
    set(handles.causes_motor,'visible', 'off');
    set(handles.possible_motor,'visible', 'off');
    otherwise
    set(handles.state_motor,'String','--');
    set(handles.causes_motor,'String','');
    set(handles.causes_motor,'visible', 'on');
    set(handles.possible_motor,'visible', 'on');
end

a = motor(4,2)
set(handles.IR1,'String',a);

a = motor(4,3)
set(handles.IY1,'String',a)

a = motor(4,4)
set(handles.IB1,'String',a);
function varargout = Oil_Tank(varargin)
  % OIL_TANK M-file for Oil_Tank.fig
  gui_Singleton = 1;
  gui_State = struct('gui_Name',       mfilename, ...
  'gui_Singleton',  gui_Singleton, ...
  'gui_OpeningFcn', @Oil_Tank_OpeningFcn, ...
  'gui_OutputFcn',  @Oil_Tank_OutputFcn, ...
  'gui_LayoutFcn',  [], ...
  'gui_Callback',   []);
  if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
  end
  if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
  else
    gui_mainfcn(gui_State, varargin{:});
  end
  % End initialization code - DO NOT EDIT

  % --- Executes just before Oil_Tank is made visible.
  function Oil_Tank_OpeningFcn(hObject, eventdata, handles, varargin)
    % This function has no output args, see OutputFcn.
    % hObject    handle to figure
    % eventdata  reserved - to be defined in a future version of MATLAB
    % handles    structure with handles and user data (see GUIDATA)
    % varargin   command line arguments to Oil_Tank (see VARARGIN)

    % Choose default command line output for Oil_Tank
    handles.output = hObject;

    % Update handles structure
    guidata(hObject, handles);

    % UIWAIT makes Oil_Tank wait for user response (see UIRESUME)
    % uiwait(handles.figure1);
  end

  % --- Outputs from this function are returned to the command line.
  function varargout = Oil_Tank_OutputFcn(hObject, eventdata, handles)
    % varargout    cell array for returning output args (see VARargout);
    % hObject     handle to figure
    % eventdata   reserved - to be defined in a future version of MATLAB
    % handles     structure with handles and user data (see GUIDATA)

    % Get default command line output from handles structure
    varargout{1} = handles.output;

  end

  % --- Executes on button press in pushbutton1.
  function pushbutton1_Callback(hObject, eventdata, handles)
    % hObject     handle to pushbutton1 (see GCBO)
    % eventdata   reserved - to be defined in a future version of MATLAB
    % handles     structure with handles and user data (see GUIDATA)
oil = evalin('base','oil')
a = oil(2,5)

switch (a)
    case 1
        set(handles.state_oil,'String','Open Phase');
        set(handles.causes_oil,'String','Warning !!! Check wiring or mains supply');
        set(handles.causes_oil,'visible', 'on');
        set(handles.possible_oil,'visible', 'on');
    case 2
        set(handles.state_oil,'String','Damage');
        set(handles.causes_oil,'String','Warning !!! Please check short');
        set(handles.causes_oil,'visible', 'on');
        set(handles.possible_oil,'visible', 'on');
    case 3
        set(handles.state_oil,'String','Good');
        set(handles.causes_oil,'visible', 'off');
        set(handles.possible_oil,'visible', 'off');
    otherwise
        set(handles.state_oil,'String','--');
        set(handles.causes_oil,'visible', 'on');
        set(handles.possible_oil,'visible', 'on');
        set(handles.causes_oil,'String','');
end

a = oil(1,2)
set(handles.pressure,'String',a);

a = oil(2,2)
set(handles.quantity,'String',a);
function varargout = Weaving_GUI(varargin)
% WEAVING_GUI M-file for Weaving_GUI.fig
%      WEAVING_GUI, by itself, creates a new WEAVING_GUI or raises the existing
% gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton', gui_Singleton, ...
                   'gui_OpeningFcn', @Weaving_GUI_OpeningFcn, ...
                   'gui_OutputFcn',  @Weaving_GUI_OutputFcn, ...
                   'gui_LayoutFcn',  [], ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end
if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before Weaving_GUI is made visible.
function Weaving_GUI_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to Weaving_GUI (see VARARGIN)

% Choose default command line output for Weaving_GUI
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes Weaving_GUI wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = Weaving_GUI_OutputFcn(hObject, eventdata, handles)
% varargout    cell array for returning output args (see VARARGOUT);
% hObject      handle to figure
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout(1) = handles.output;

%[x,map]=imread('IRA.jpg','jpg');
%image(x),colormap(map),axis off,hold on

%------------------start clock and data-----------------------------
--
% set(handles.date1,'String',date);
% x=1
% while x
% t=clock;
% set(handles.time1,'String',[num2str(fix(t(4))),':',num2str(fix(t(5)))
,':',num2str(fix(t(6)))]);
% drawnow
% end
%-------------------end time and date-------------------------
% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
%--------------------------------------------------------------
weaving_section_1FigureHandle  = weaving_section_1;  %stores the
figure handle of Quan's GUI here

%stores the GUI data from Quan's GUI here
%now we can access any of the data from Quan's GUI!!!
weaving_section_1Data = guidata(weaving_section_1FigureHandle);

%store the input text from Quan's GUI
%weaving_section_1_input =
get(weaving_section_1Data.window2_edit1,'String');

%---------------------------------------------------------------
% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
%--------------------------------------------------------------

% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
%--------------------------------------------------------------

% --- Executes on button press in pushbutton4.
function pushbutton4_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
close all

% --- Executes on button press in pushbutton4.
function pushbutton5_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
function varargout = Emergency_faults(varargin)
% EMERGENCY_FAULTS M-file for Emergency_faults.fig
% EMERGENCY_FAULTS, by itself, creates a new EMERGENCY_FAULTS or raises the existing
% gui_Singleton = 1;
% gui_State = struct('gui_Name',       mfilename, ...
%   'gui_Singleton',  gui_Singleton, ...
%   'gui_OpeningFcn', @Emergency_faults_OpeningFcn, ...
%   'gui_OutputFcn',  @Emergency_faults_OutputFcn, ...
%   'gui_LayoutFcn',  [], ...
%   'gui_Callback',   []);
% if nargin && ischar(varargin{1})
%     gui_State.gui_Callback = str2func(varargin{1});
% end
% if nargout
%   [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
% else
% gui_mainfcn(gui_State, varargin{:});
% end
% End initialization code - DO NOT EDIT

% --- Executes just before Emergency_faults is made visible.
function Emergency_faults_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to Emergency_faults (see VARARGIN)

% Choose default command line output for Emergency_faults
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes Emergency_faults wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = Emergency_faults_OutputFcn(hObject, eventdata, handles)
% varargout    cell array for returning output args (see VARARGOUT);
% hObject      handle to figure
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout(1) = handles.output;
% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
    % hObject    handle to pushbutton1 (see GCBO)
    % eventdata  reserved - to be defined in a future version of MATLAB
    % handles    structure with handles and user data (see GUIDATA)
    environment = evalin('base','environment')
    a = environment(1,5)

    switch (a)
        case 1
            set(handles.state_environment,'String','Bad');
            set(handles.causes_environment,'visible', 'on');
            set(handles.possible_envir,'visible', 'on');
            set(handles.causes_environment,'String','Warning !!! Please check Sprinkler and exaust fan also');
        case 2
            set(handles.state_environment,'String','Medium');
            set(handles.causes_environment,'visible', 'on');
            set(handles.possible_envir,'visible', 'on');
            set(handles.causes_environment,'String','Please check Exaust fan');
        case 3
            set(handles.state_environment,'String','Good');
            set(handles.causes_environment,'visible', 'off');
            set(handles.possible_envir,'visible', 'off');
        otherwise
            set(handles.state_environment,'String','--');
            set(handles.causes_environment,'visible', 'on');
            set(handles.possible_envir,'visible', 'on');
            set(handles.causes_environment,'String','');
    end

    a = environment(1,2)
    set(handles.temperature,'String',a);

    a = environment(2,2)
    set(handles.humidity,'String',a);
FIS code for different sections is in following sections

[System]
Name='Environment'
Type='mamdani'
Version=2.0
NumInputs=2
NumOutputs=1
NumRules=9
AndMethod='min'
OrMethod='max'
ImpMethod='min'
AggMethod='max'
DefuzzMethod='centroid'

[Input1]
Name='Temperature'
Range=[0 60]
NumMFs=3
MF1='Low':'trapmf',[-21.6 -2.4 14.5238095238095 24.8]
MF2='Medium':'trapmf',[17.2238095238095 25.1538095238095 33.2238095238095 39.7238095238095]
MF3='High':'trapmf',[34.6 42.1428571428571 62.4 81.6]

[Input2]
Name='Humidity'
Range=[0 100]
NumMFs=3
MF1='Low':'trapmf',[-36 -4 4 36]
MF2='Medium':'trapmf',[18 35 58.9 78.7]
MF3='High':'trapmf',[60 81.8783068783069 100 132]

[Output1]
Name='Environment__Condition'
Range=[0 1]
NumMFs=3
MF1='Bad':'trimf',[-0.4 0 0.4]
MF2='Medium':'trimf',[0.1 0.5 0.9]
MF3='Good':'trimf',[0.6 1 1.4]

[Rules]
1 1, 1 (1) : 1
1 2, 1 (1) : 1
1 3, 1 (1) : 1
2 1, 1 (1) : 1
2 2, 3 (1) : 1
2 3, 2 (1) : 1
3 1, 1 (1) : 1
3 2, 1 (1) : 1
3 3, 1 (1) : 1

[System]
Name='Motor__Condition'
Type='mamdani'
Version=2.0
NumInputs=3
NumOutputs=1  
NumRules=28  
AndMethod='min'  
OrMethod='max'  
ImpMethod='min'  
AggMethod='max'  
DefuzzMethod='centroid'

[Input1]  
Name='Ia'  
Range=[0 50]  
NumMFs=5  
MF1='Z':'trimf',[-17.2 0 1.6]  
MF2='S':'trapmf',[1 2 4 5]  
MF3='M':'trapmf',[4 9 22 26]  
MF4='B':'trapmf',[24 31 36 43]  
MF5='VB':'trimf',[40 50 51]

[Input2]  
Name='Ib'  
Range=[0 50]  
NumMFs=5  
MF1='Z':'trimf',[-17.2 0 1.6]  
MF2='S':'trapmf',[1 2 4 5]  
MF3='M':'trapmf',[4 9 22 26]  
MF4='B':'trapmf',[24 31 36 43]  
MF5='VB':'trimf',[40 50 51]

[Input3]  
Name='Ic'  
Range=[0 50]  
NumMFs=5  
MF1='Z':'trimf',[-17.2 0 1.6]  
MF2='S':'trapmf',[1 2 4 5]  
MF3='M':'trapmf',[4 9 22 26]  
MF4='B':'trapmf',[24 31 36 43]  
MF5='VB':'trimf',[40 50 51]

[Output1]  
Name='Motor__Condition'  
Range=[0 1]  
NumMFs=5  
MF1='GOOD':'trapmf',[0.805555555555556 0.956 1.04 1.04]  
MF2='Damage':'trapmf',[0.1 0.188 0.288 0.388]  
MF3='Over__Loaded':'trapmf',[0.541 0.64179894179894 0.719 0.853]  
MF4='Critically__Loaded':'trapmf',[0.35 0.4 0.5 0.6]  
MF5='Open__Phase':'trapmf',[-0.0437037037037037 -0.0037037037037037 0.0962962962962963 0.1562962962962963]

[Rules]  
1 1 1, 5 (1) : 1  
-1 1 -1, 5 (1) : 1  
-1 -1 1, 5 (1) : 1  
2 2 2, 1 (1) : 1  
2 2 3, 3 (1) : 1  
2 3 2, 3 (1) : 1
[System]
Name='OIL'
Type='mamdani'
Version=2.0
NumInputs=2
NumOutputs=1
NumRules=9
AndMethod='min'
OrMethod='max'
ImpMethod='min'
AggMethod='max'
DefuzzMethod='centroid'

[Input1]
Name='Quantity'
Range=[0 20]
NumMFs=3
MF1='Low':'trimf',[-10 0 5.164]
MF2='Medium':'trapmf',[2 6 13 17]
MF3='High':'trimf',[15 20 28]

[Input2]
Name='Pressure'
Range=[0 5]
NumMFs=3
MF1='Low':'trimf',[-2 0 1]
MF2='Medium':'trapmf',[0.5 2 3 4.3]
MF3='High':'trimf',[3.5 5 7]

[Output1]
Name='Oil_Tank'
Range=[0 1]
NumMFs=3
MF1='Good':'trimf',[-0.4 0 0.4]
MF2='Medium':'trimf',[0.1 0.5 0.9]
MF3='Bad':'trimf',[0.6 1 1.4]

[Rules]
1 1, 3 (1) : 1
2 1, 3 (1) : 1
3 1, 3 (1) : 1
2 2, 2 (1) : 1
3 3, 1 (1) : 1
2 3, 1 (1) : 1
1 3, 2 (1) : 1
1 2, 2 (1) : 1
3 2, 3 (1) : 1