classdef Motor\_spindownSerialReadTest < matlab.apps.AppBase

% Properties that correspond to app components

properties (Access = public)

UIFigure matlab.ui.Figure

GridLayout matlab.ui.container.GridLayout

LeftPanel matlab.ui.container.Panel

StartButton matlab.ui.control.Button

MotorButtonGroup matlab.ui.container.ButtonGroup

CButton matlab.ui.control.ToggleButton

BButton matlab.ui.control.ToggleButton

AButton matlab.ui.control.ToggleButton

Switch matlab.ui.control.Switch

ViscousDampingConstantEditFieldLabel matlab.ui.control.Label

ViscousDampingConstantEditField matlab.ui.control.NumericEditField

TorquefromCoulombFrictionEditField matlab.ui.control.NumericEditField

TorquefromCoulombFrictionEditFieldLabel matlab.ui.control.Label

UIAxes2 matlab.ui.control.UIAxes

RightPanel matlab.ui.container.Panel

UIAxes matlab.ui.control.UIAxes

end

% Properties that correspond to apps with auto-reflow

properties (Access = private)

onePanelWidth = 576;

end

methods (Access = private)

function results = Motor\_Dynamics(app)

%%%% motor motion Written by Max Austin... in a a day ... with no experience

%%%% sets up the motor properties

omeganot = 64; % rad/sec

ThetaNot = 0;

Props.Tstall = 0;

Props.Omegnoload = 64;

Props.Inert = 1; %kg\*m^2

Time = 0:.0005:6;

sets4ode = odeset('AbsTol',1e-10);

%%%% Define the motor Properties

switch(app.Switch.Value)

case 'Assignment 1'

Props.c = 0;

Props.b = app.ViscousDampingConstantEditField.Value;

Props.Tf = app.TorquefromCoulombFrictionEditField.Value;

case 'Assignment 2'

if(app.AButton.Value == true)

Props.c = 0;

Props.b = 0.00001;

Props.Tf = 9;

elseif(app.BButton.Value == true)

Props.c = 0;

Props.b = 1.2;

Props.Tf = 0;

else

Props.c = 0;

Props.b = 0.75;

Props.Tf = 7;

end

end

if ((Props.b > 0 ) && (Props.b <= 0.5) && (Props.Tf < 10))

Time = 0:.005:10;

end

[T,Omega] = ode45(@app.Motor\_EOM,Time,[ThetaNot,omeganot],sets4ode,Props);

TH = Omega(:,1);

OM = Omega(:,2);

%%% stopping

OM(OM <= 0) = 0;

TMP = find((OM <= 0),1);

TH(TMP:end) = TH(TMP-1);

results.T = T;

results.THETA = TH;

results.OMEGA = OM;

end

end

methods(Static)

function domegdt = Motor\_EOM(t,omeg,Props)

c = Props.c;

b = Props.b;

Tf = Props.Tf;

Tres = Tf + b\*omeg(2) + c\*omeg(2).^2;

Tmot = Props.Tstall -Props.Tstall/Props.Omegnoload\*omeg(2);

Tmot(Tmot >= Props.Tstall) = Props.Tstall;

domegdt =[omeg(2); (Tmot- Tres)/Props.Inert];

end

end

% Callbacks that handle component events

methods (Access = private)

% Button pushed function: StartButton

function StartButtonPushed(app, event)

Pltdata = Motor\_Dynamics(app);

time = Pltdata.T;

TH = Pltdata.THETA;

om = Pltdata.OMEGA;

%%%% Motor rotor

Rotlength = 0.9;

Rotor\_PPos = [Rotlength\*cos(TH),Rotlength\*sin(TH)];

Rotor\_MPos = [-Rotlength\*cos(TH),-Rotlength\*sin(TH)];

circx = 0.5\*cos(0:.001:6.29);

circy = 0.5\*sin(0:.001:6.29);

%%%% changing the number of frames

FR = 50;

NUM\_PLT\_FRAMES = floor(length(om)/FR);

ylim(app.UIAxes,[0,70])

xlim(app.UIAxes,[time(1),time(end)])

ylim(app.UIAxes2,[-1,1])

xlim(app.UIAxes2,[-1,1])

for INC = 1:NUM\_PLT\_FRAMES

PTPOINT = FR\*INC;

plot(app.UIAxes,time(1:PTPOINT),om(1:PTPOINT))

trnd = round(time(PTPOINT),3);

ornd = round(om(PTPOINT),4);

text(app.UIAxes,2.5,67,strcat('t = ',num2str(trnd),' s'),'FontSize',14)

text(app.UIAxes,4.25,67,strcat('\omega = ',num2str(ornd),' rad/s'),'FontSize',14)

cla(app.UIAxes2)

plot(app.UIAxes2,circx,circy,'Color','k')

hold(app.UIAxes2,'on')

plot(app.UIAxes2,[Rotor\_PPos(PTPOINT,1),Rotor\_MPos(PTPOINT,1)],[Rotor\_PPos(PTPOINT,2),Rotor\_MPos(PTPOINT,2)],'Color','c','LineWidth',3)

plot(app.UIAxes2,0,0,'o','Color','k','LineWidth',3)

drawnow

end

end

% Value changed function: TorquefromCoulombFrictionEditField

function TorquefromCoulombFrictionEditFieldValueChanged(app, event)

%%% dead function I can't delete

end

% Changes arrangement of the app based on UIFigure width

function updateAppLayout(app, event)

currentFigureWidth = app.UIFigure.Position(3);

if(currentFigureWidth <= app.onePanelWidth)

% Change to a 2x1 grid

app.GridLayout.RowHeight = {502, 502};

app.GridLayout.ColumnWidth = {'1x'};

app.RightPanel.Layout.Row = 2;

app.RightPanel.Layout.Column = 1;

else

% Change to a 1x2 grid

app.GridLayout.RowHeight = {'1x'};

app.GridLayout.ColumnWidth = {220, '1x'};

app.RightPanel.Layout.Row = 1;

app.RightPanel.Layout.Column = 2;

end

end

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.AutoResizeChildren = 'off';

app.UIFigure.Position = [100 100 662 502];

app.UIFigure.Name = 'UI Figure';

app.UIFigure.SizeChangedFcn = createCallbackFcn(app, @updateAppLayout, true);

% Create GridLayout

app.GridLayout = uigridlayout(app.UIFigure);

app.GridLayout.ColumnWidth = {220, '1x'};

app.GridLayout.RowHeight = {'1x'};

app.GridLayout.ColumnSpacing = 0;

app.GridLayout.RowSpacing = 0;

app.GridLayout.Padding = [0 0 0 0];

app.GridLayout.Scrollable = 'on';

% Create LeftPanel

app.LeftPanel = uipanel(app.GridLayout);

app.LeftPanel.Layout.Row = 1;

app.LeftPanel.Layout.Column = 1;

% Create UIAxes2

app.UIAxes2 = uiaxes(app.LeftPanel);

app.UIAxes2.Position = [0 1 196 138];

% Create TorquefromCoulombFrictionEditFieldLabel

app.TorquefromCoulombFrictionEditFieldLabel = uilabel(app.LeftPanel);

app.TorquefromCoulombFrictionEditFieldLabel.HorizontalAlignment = 'right';

app.TorquefromCoulombFrictionEditFieldLabel.Position = [39 406 163 22];

app.TorquefromCoulombFrictionEditFieldLabel.Text = 'Torque from Coulomb Friction';

% Create TorquefromCoulombFrictionEditField

app.TorquefromCoulombFrictionEditField = uieditfield(app.LeftPanel, 'numeric');

app.TorquefromCoulombFrictionEditField.ValueChangedFcn = createCallbackFcn(app, @TorquefromCoulombFrictionEditFieldValueChanged, true);

app.TorquefromCoulombFrictionEditField.Position = [71 385 100 22];

% Create ViscousDampingConstantEditField

app.ViscousDampingConstantEditField = uieditfield(app.LeftPanel, 'numeric');

app.ViscousDampingConstantEditField.Position = [71 325 100 22];

% Create ViscousDampingConstantEditFieldLabel

app.ViscousDampingConstantEditFieldLabel = uilabel(app.LeftPanel);

app.ViscousDampingConstantEditFieldLabel.HorizontalAlignment = 'right';

app.ViscousDampingConstantEditFieldLabel.Position = [52 346 150 22];

app.ViscousDampingConstantEditFieldLabel.Text = 'Viscous Damping Constant';

% Create Switch

app.Switch = uiswitch(app.LeftPanel, 'slider');

app.Switch.Items = {'Assignment 1', 'Assignment 2'};

app.Switch.Position = [87 279 45 20];

app.Switch.Value = 'Assignment 1';

% Create MotorButtonGroup

app.MotorButtonGroup = uibuttongroup(app.LeftPanel);

app.MotorButtonGroup.Title = 'Motor';

app.MotorButtonGroup.Position = [52 153 123 106];

% Create AButton

app.AButton = uitogglebutton(app.MotorButtonGroup);

app.AButton.Text = 'A';

app.AButton.Position = [11 53 100 22];

app.AButton.Value = true;

% Create BButton

app.BButton = uitogglebutton(app.MotorButtonGroup);

app.BButton.Text = 'B';

app.BButton.Position = [11 32 100 22];

% Create CButton

app.CButton = uitogglebutton(app.MotorButtonGroup);

app.CButton.Text = 'C';

app.CButton.Position = [11 11 100 22];

% Create StartButton

app.StartButton = uibutton(app.LeftPanel, 'push');

app.StartButton.ButtonPushedFcn = createCallbackFcn(app, @StartButtonPushed, true);

app.StartButton.Position = [71 449 100 22];

app.StartButton.Text = 'Start';

% Create RightPanel

app.RightPanel = uipanel(app.GridLayout);

app.RightPanel.Layout.Row = 1;

app.RightPanel.Layout.Column = 2;

% Create UIAxes

app.UIAxes = uiaxes(app.RightPanel);

title(app.UIAxes, 'Angular Speed of Motor')

xlabel(app.UIAxes, 't (s)')

ylabel(app.UIAxes, '\omega (m/s)')

app.UIAxes.XGrid = 'on';

app.UIAxes.YGrid = 'on';

app.UIAxes.Position = [15 44 389 427];

% 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 = Motor\_spindownSerialReadTest

% Create UIFigure and components

createComponents(app)

% Register the app with App Designer

registerApp(app, app.UIFigure)

if nargout == 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

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Vibrations GUI \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

classdef SDnV\_lab\_GUI\_Tower < matlab.apps.AppBase

% Properties that correspond to app components

properties (Access = public)

UIFigure matlab.ui.Figure

MotoronoffButton matlab.ui.control.Button

Slider matlab.ui.control.Slider

SliderLabel matlab.ui.control.Label

TextArea2 matlab.ui.control.TextArea

Lamp\_4 matlab.ui.control.Lamp

Lamp\_3 matlab.ui.control.Lamp

Lamp\_2 matlab.ui.control.Lamp

Lamp matlab.ui.control.Lamp

ConnecttoArduinoButton matlab.ui.control.Button

StopReadingButton matlab.ui.control.Button

StrartReadingButton matlab.ui.control.Button

SystemDynamicsandVibrationsLab5ExperimentalRootLocusLabel matlab.ui.control.Label

FilenameTextArea matlab.ui.control.TextArea

FilenameTextAreaLabel matlab.ui.control.Label

ExportDataButton matlab.ui.control.Button

UIAxes\_3 matlab.ui.control.UIAxes

UIAxes\_2 matlab.ui.control.UIAxes

UIAxes matlab.ui.control.UIAxes

end

properties (Access = private)

Acceleration % Description

Velocity % Description

Position % Description

Time % Description

sp % serial port object

STOPPED

iCOM

end

% Callbacks that handle component events

methods (Access = private)

% Button pushed function: StrartReadingButton

function StrartReadingButtonPushed(app, event)

% tell arduino to start recording

writeline(app.sp,"startDataread")

pause(0.5)

% app.Lamp\_2.Color.Enable = 'on';

% app.Lamp\_3.Color.Enable = 'off';

OverTimer = tic;

temptime = 0;

app.Time = 0;

app.Acceleration = 0;

app.STOPPED = 0;

while app.STOPPED <1

while (60 > toc(OverTimer))

% read line and if something is there record data

dataStr = readline(app.sp);

if isempty(dataStr) == 0

dataNum = str2double(dataStr);

%

%dataStr = splitlines(dataStr);

%now record time as thes data sets are sent seporate

temptime = str2double(readline(app.sp));

app.Time = [app.Time; temptime];

app.Acceleration = [app.Acceleration; dataNum];

app.TextArea2.Value = dataStr;

end

end

% % change the number of frames

% FR = 50;

% NUM\_PLT\_FRAMES = floor(length(om)/FR);

% record data and time arrays

end

% get location of file

end

% Button pushed function: ConnecttoArduinoButton

function ConnecttoArduinoButtonPushed(app, event)

if (isempty(app.iCOM)==1)

app.iCOM = 0;

end

if (isempty(app.sp)==1)

app.Lamp.Color = [1 0 0];

spl = serialportlist;

% for i = 1:length(spl)

app.iCOM = app.iCOM + 1;

serialObjectTemp = serialport(spl(app.iCOM), 9600);

configureTerminator(serialObjectTemp,"LF");

writeline(serialObjectTemp,"serialConnectNow")

pause(0.5)

% if serial available and reads "amHere"

app.TextArea2.Value = spl(app.iCOM);

% if isempty(readline(serialObjectTemp)) == 0

TtempText = readline(serialObjectTemp);

app.TextArea2.Value = TtempText;

if (class(TtempText) == "string")

TtempText = splitlines(TtempText);

if (TtempText(1) == "amHere")

clear serialObjectTemp;

app.sp = serialport(spl(app.iCOM), 9600);

app.Lamp.Color = [0 1 0];

% status = getpinstatus(app.sp);

% app.TextArea2.Value = status;

end

end

% end

% end

end

end

% Button pushed function: StopReadingButton

function StopReadingButtonPushed(app, event)

% app.Lamp\_3.Color.Enable = "ON";

% app.Lamp\_2.Color.Enable = "OFF";

%ylim(app.UIAxes,[app.Acceleration(min),70])

app.STOPPED = 1;

xlim(app.UIAxes,[app.Time(1),app.Time(end)])

xlim(app.UIAxes\_2,[app.Time(1),app.Time(end)])

xlim(app.UIAxes\_3,[app.Time(1),app.Time(end)])

offsetA = mean(app.Acceleration(end-21:end-1));

app.Acceleration = app.Acceleration(1:end-1) - offsetA;

app.Time = app.Time(1:end-1);

plot(app.UIAxes,app.Time,app.Acceleration,'g-')

% calculate position

app.Velocity = [];

app.Position = [];

for ii = 1:length(app.Time)-1

Vii = (app.Acceleration(ii+1) - app.Acceleration(ii))/(app.Time(ii+1) - app.Time(ii));

app.Velocity = [app.Velocity; Vii];

end

for ii = 1:length(app.Time)-length(app.Velocity)

app.Velocity = [app.Velocity; 0];

end

plot(app.UIAxes\_2,app.Time,app.Velocity,'g-')

for ii = 1:length(app.Time)-2

Pii = (app.Velocity(ii+1) - app.Velocity(ii))/(app.Time(ii+1) - app.Time(ii));

app.Position = [app.Position; Pii];

end

for ii = 1:length(app.Time)-length(app.Position)

app.Position = [app.Position; 0];

end

plot(app.UIAxes\_3,app.Time,app.Position,'g-')

end

% Button pushed function: ExportDataButton

function ExportDataButtonPushed(app, event)

% export data to text file

% Shape = {'Time';'Acceleration';'Velocity';'Position'};

% app.TextArea2.Value = string(length(app.Time(1:length(app.Velocity))));

% pause(1)

% app.TextArea2.Value = string(length(app.Velocity));

% pause(1)

% app.TextArea2.Value = string(length(app.Acceleration(1:length(app.Velocity))));

% pause(1)

% app.TextArea2.Value = string(length(app.Position));

% pause(1)

TimeT = app.Time(4:length(app.Velocity));

AccT = app.Acceleration(4:length(app.Velocity));

VelT = app.Velocity(4:end);

PosT = app.Position(4:end);

T = table(TimeT,AccT);

% rename

filemeData = string(app.FilenameTextArea.Value);

writetable(T,filemeData);

% type tabledata.txt

end

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.Color = [0.7176 0.8392 0.9216];

app.UIFigure.Position = [100 100 1129 876];

app.UIFigure.Name = 'MATLAB App';

% Create UIAxes

app.UIAxes = uiaxes(app.UIFigure);

xlabel(app.UIAxes, 'Time')

ylabel(app.UIAxes, 'Acceleration')

zlabel(app.UIAxes, 'Z')

app.UIAxes.Color = [0.902 0.902 0.902];

app.UIAxes.FontSize = 14;

app.UIAxes.Position = [450 423 612 365];

% Create UIAxes\_2

app.UIAxes\_2 = uiaxes(app.UIFigure);

ylabel(app.UIAxes\_2, 'Velocity')

zlabel(app.UIAxes\_2, 'Z')

app.UIAxes\_2.Color = [0.902 0.902 0.902];

app.UIAxes\_2.FontSize = 14;

app.UIAxes\_2.Position = [530 209 228 183];

% Create UIAxes\_3

app.UIAxes\_3 = uiaxes(app.UIFigure);

xlabel(app.UIAxes\_3, 'Time')

ylabel(app.UIAxes\_3, 'Position')

zlabel(app.UIAxes\_3, 'Z')

app.UIAxes\_3.Color = [0.902 0.902 0.902];

app.UIAxes\_3.FontSize = 14;

app.UIAxes\_3.Position = [864 240 183 152];

% Create ExportDataButton

app.ExportDataButton = uibutton(app.UIFigure, 'push');

app.ExportDataButton.ButtonPushedFcn = createCallbackFcn(app, @ExportDataButtonPushed, true);

app.ExportDataButton.FontSize = 16;

app.ExportDataButton.Position = [128 225 126 27];

app.ExportDataButton.Text = 'Export Data';

% Create FilenameTextAreaLabel

app.FilenameTextAreaLabel = uilabel(app.UIFigure);

app.FilenameTextAreaLabel.BackgroundColor = [0.9412 0.9412 0.9412];

app.FilenameTextAreaLabel.HorizontalAlignment = 'right';

app.FilenameTextAreaLabel.FontSize = 16;

app.FilenameTextAreaLabel.Position = [152 290 76 22];

app.FilenameTextAreaLabel.Text = 'Filename:';

% Create FilenameTextArea

app.FilenameTextArea = uitextarea(app.UIFigure);

app.FilenameTextArea.BackgroundColor = [0.9412 0.9412 0.9412];

app.FilenameTextArea.Position = [77 258 225 24];

% Create SystemDynamicsandVibrationsLab5ExperimentalRootLocusLabel

app.SystemDynamicsandVibrationsLab5ExperimentalRootLocusLabel = uilabel(app.UIFigure);

app.SystemDynamicsandVibrationsLab5ExperimentalRootLocusLabel.HorizontalAlignment = 'center';

app.SystemDynamicsandVibrationsLab5ExperimentalRootLocusLabel.FontSize = 22;

app.SystemDynamicsandVibrationsLab5ExperimentalRootLocusLabel.FontWeight = 'bold';

app.SystemDynamicsandVibrationsLab5ExperimentalRootLocusLabel.Position = [242 787 363 106];

app.SystemDynamicsandVibrationsLab5ExperimentalRootLocusLabel.Text = {'System Dynamics and Vibrations'; 'Lab 5: Experimental Root Locus'};

% Create StrartReadingButton

app.StrartReadingButton = uibutton(app.UIFigure, 'push');

app.StrartReadingButton.ButtonPushedFcn = createCallbackFcn(app, @StrartReadingButtonPushed, true);

app.StrartReadingButton.FontSize = 16;

app.StrartReadingButton.Position = [99 658 131 42];

app.StrartReadingButton.Text = 'Strart Reading';

% Create StopReadingButton

app.StopReadingButton = uibutton(app.UIFigure, 'push');

app.StopReadingButton.ButtonPushedFcn = createCallbackFcn(app, @StopReadingButtonPushed, true);

app.StopReadingButton.FontSize = 16;

app.StopReadingButton.Position = [99 594 131 42];

app.StopReadingButton.Text = 'Stop Reading';

% Create ConnecttoArduinoButton

app.ConnecttoArduinoButton = uibutton(app.UIFigure, 'push');

app.ConnecttoArduinoButton.ButtonPushedFcn = createCallbackFcn(app, @ConnecttoArduinoButtonPushed, true);

app.ConnecttoArduinoButton.FontSize = 16;

app.ConnecttoArduinoButton.Position = [82 759 168 42];

app.ConnecttoArduinoButton.Text = 'Connect to Arduino';

% Create Lamp

app.Lamp = uilamp(app.UIFigure);

app.Lamp.Position = [27 755 51 51];

app.Lamp.Color = [1 0 0];

% Create Lamp\_2

app.Lamp\_2 = uilamp(app.UIFigure);

app.Lamp\_2.Enable = 'off';

app.Lamp\_2.Position = [27 654 51 51];

% Create Lamp\_3

app.Lamp\_3 = uilamp(app.UIFigure);

app.Lamp\_3.Enable = 'off';

app.Lamp\_3.Position = [27 590 51 51];

app.Lamp\_3.Color = [1 0 0];

% Create Lamp\_4

app.Lamp\_4 = uilamp(app.UIFigure);

app.Lamp\_4.Enable = 'off';

app.Lamp\_4.Visible = 'off';

app.Lamp\_4.Position = [27 755 51 51];

% Create TextArea2

app.TextArea2 = uitextarea(app.UIFigure);

app.TextArea2.Position = [119 717 94 30];

% Create SliderLabel

app.SliderLabel = uilabel(app.UIFigure);

app.SliderLabel.HorizontalAlignment = 'right';

app.SliderLabel.Position = [142 478 36 22];

app.SliderLabel.Text = 'Slider';

% Create Slider

app.Slider = uislider(app.UIFigure);

app.Slider.Position = [199 487 150 3];

% Create MotoronoffButton

app.MotoronoffButton = uibutton(app.UIFigure, 'push');

app.MotoronoffButton.FontSize = 16;

app.MotoronoffButton.FontWeight = 'bold';

app.MotoronoffButton.FontColor = [0.4667 0.6745 0.1882];

app.MotoronoffButton.Position = [318 355 69 56];

app.MotoronoffButton.Text = {'Motor'; 'on/off'};

% 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 = SDnV\_lab\_GUI\_Tower

% Create UIFigure and components

createComponents(app)

% Register the app with App Designer

registerApp(app, app.UIFigure)

if nargout == 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