# APPENDICE A

# A .1 Gui Matlab 6.5:algoritmo drag and drop

L'interfaccia utente dell'applicazione è stata realizzata tenendo presente i suggerimenti dei più comuni software di grafica 3D ed in particolare un approccio intuitivo per chi dovrà usare l'applicazione.

La definizione delle traiettorie nello spazio è facilitata dall'uso di oggetti predefiniti modificabili attraverso parametri caratteristici, così non è per il *vettore velocità di volo* e il *vettore angolo di rollio*, per la definizione dei quali ho implementato la funzionalità drag and drop.

L'ambiente di sviluppo MATLAB mette a disposizione gli oggetti e le funzioni base per il plotting dei dati o come nel caso del rendering delle patch.

La Gui MATLAB permette di:

- Associare ad ogni oggetto presente nello spazio cartesiano un handle grafico
- Di definire le proprietà di ogni handle grafico
- Di valutare le proprietà di ogni handle grafico

Sulla base di queste considerazioni per implementare la funzionalità drag and drop ho considerato ogni punto del vettore velocità di volo *( o del*

*vettore angolo di rollio)* come un handle grafico. Di seguito le proprietà associate ad un handle grafico

```
Color = [1 0 0]
EraseMode = normal
LineStyle = -
LineWidth = [0.5]
Marker = none
MarkerSize = [6]
MarkerEdgeColor = auto
MarkerFaceColor = none
XData = [10]
YData = [10]
ZData = [10]

BeingDeleted = off
ButtonDownFcn =
Children = []
Clipping = on
CreateFcn =
DeleteFcn =
BusyAction = queue
HandleVisibility = on
HitTest = on
Interruptible = on
Parent = [101.001]
Selected = off
SelectionHighlight = on
Tag =
Type = line
UIContextMenu = []
UserData = []
Visible = on
```

Le proprietà utilizzate per implementare la funzionalità drag and drop sono:

```
XData = []
YData = []
ZData = []
Color=[ , ,]
ButtonDownFcn = 'ButtonDownPointFcn_callback(hobject,eventdata,handles)'
```

Di seguito il codice associato al click del mouse dei punti costituenti il *vettore angolo di rollio*.

```
function ButtonDownPointFcn_callback(hObject, eventdata, handles)


handlesphipoint=getappdata(handles.Main,'handlesphipoint');
dim=size(handlesphipoint);
X=get(hObject,'XData')
Y=get(hObject,'YData')

[j,window]=gcbo;

if(strcmp(get(window,'pointer'),'circle'))

button=whichbutton;

switch (button)


    case 0
        set(hObject,'Color',[1 0 0]);
        for i=1:dim(2)
            if hObject==handlesphipoint(i) break,end
        end

        setappdata(handles.Main,'Selectedpoint',i);
        X=getappdata(handles.Main,'X');
        Y=getappdata(handles.Main,'Y');
        Z=getappdata(handles.Main,'Z');

        set(handles.vista3d,'Nextplot','add');
        h=plot3(X(i),Y(i),Z(i),'.','Markersize',20,'Color',[1 0 0],'Parent',handles.vista3d);
        setappdata(handles.Main,'viewpoint',h);
    case 1
        if get(hObject,'Color')==[1 0 0]

            setappdata(handles.Main,'Movepoint',0)


            set(hObject,'Color',[ 0 1 0]);

            phiangle=getappdata(handles.Main,'phiangle');
            [n,m]=size(phiangle);
            set(handles.angles,'Nextplot','replacechildren')
            phin=[];
            for i=1:m
                phin(i)=i;
            end
            j=getappdata(handles.Main,'Selectedpoint')
            vectorphi=getappdata(handles.Main,'Vectorphi');


            dim=size(vectorphi);
            if dim(2)==2
                if j==vectorphi(2)||j==vectorphi(1)
                    vectorphi=[vectorphi(1),vectorphi(2)]

                else

                    vectorphi=[vectorphi(1),j,vectorphi(2)]
                end

            else
                for k=1:dim(2)
                    if j<=vectorphi(k) break,end;
                end
                if j==vectorphi(k)

                else
```

```
                if(k==2)
                    vectorphi=[vectorphi(1),getappdata(handles.Main,'Selectedpoint'),vectorphi(k:dim(2))]
                else

                    vectorphi=[vectorphi(1:k-1),getappdata(handles.Main,'Selectedpoint'),vectorphi(k:dim(2))]
                end
            end

        end


        setappdata(handles.Main,'Vectorphi',vectorphi);
         h=getappdata(handles.Main,'viewpoint');
        if ishandle(h)
            delete(h)
        end
    end

case 2

case 3
     if get(hObject,'Color')==[1 0 0]
        setappdata(handles.Main,'Movepoint',1)
    end


end
end
```

La variabile '*Vectorphi'* associata alla Gui *(handles. Main)* è un array contenente i punti del *vettore angolo di rollio,* modificati trascinandoli con il mouse, e permette di implementare l'interpolazione su tutti i punti trascinati o solo su quelli vicini al punto selezionato.

Selezionato un punto , il trascinamento è effettuato associando alle proprietà XData, YData e ZData le coordinate puntate dal mouse sullo spazio cartesiano.

```
function Main_WindowButtonMotionFcn(hObject, eventdata, handles)
% hObject    handle to Main (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
[j,window]=gcbo;
switch (getappdata(handles.Main,'Operationset'))
    case 0
        pointwindow=get(window,'Currentpoint');
        posangle=get(handles.angles,'Position');
        c1=pointwindow(1)>posangle(1) & pointwindow(1)<posangle(1)+posangle(3) & pointwindow(2)>posangle(2)
        if(c1)
            get(handles.angles,'CurrentPoint');
            set(window,'pointer','circle');
        else
            set(window,'pointer','arrow');
        end

        if(getappdata(handles.Main,'Movepoint')==1)&& strcmp(get(window,'pointer'),'circle')
            j=getappdata(handles.Main,'Selectedpoint');
            phiangle=getappdata(handles.Main,'phiangle');
            [n,m]=size(phiangle);
            curpoint=get(gca,'Currentpoint');
            phiangle(j)=curpoint(1,2);
                        m=getappdata(handles.Main,'npoint');
```

- 
- 
- 
- 

```
            all=getappdata(handles.Main,'allplotphi');
            set(all,'XData',phin,'YData',phiangle);
            handlesvelpoint=getappdata(handles.Main,'handlesphipoint');
            dim=size(handlesvelpoint);
            for i=1:dim(2)
                if phiangle(i)<-180
                    phiangle(i)=-180;
                end
                if phiangle(i)>180
                    phiangle(i)=180;
                end

                set(handlesvelpoint(i),'XData',phin(i),'YData',phiangle(i));
            end
                        setappdata(handles.Main,'phiangle',phiangle);
        end
```

La porzione di codice riportata sopra è una funzione membro dell'handle associato alla figura corrente ed in essa è contenuto il codice che modifica per effetto del trascinamento le proprietà degli handle associati al *vettore angolo di rollio.*

# A .2 Limiti di Matlab Compiler 3: Image control tree

Matlab Compiler 3 purtroppo non prevede la possibilità di compilare codice matlab contenente porzioni di codice che fanno riferimento alle seguenti funzioni

- Actxcontrol
- Actxserver
- Eval

Per arricchire l'interfaccia dell'applicazione realizzata e permettere all'utente di avere una visione generale della pianificazione dell'air show, ho importato l'oggetto *image control tree di Microsoft* usando la funzione *'actxcontrol'*, ciò ha compromesso la possibilità di produrre un'eseguibile.

Per permettere la compilazione dell'applicazione ho realizzato nell'ambito di questa tesi l'oggetto image control tree come funzione matlab. Di seguito il codice che crea l'oggetto e che permette di modificarlo.

```
function htreel(fig,numeronodi,x,y,PictureNode,PictureNodefiglio,noditext,handles)


icone=load('Iconshtree');
set(fig,'Render','Opengl');

handles = guihandles(fig)
handles.icone = icone;
if (y-(10-numeronodi)*40)<0
    yt=40;
else
    yt=y-(10-numeronodi)*40;
end

base=axes('Unit','pixel','Position',[x+81 y 200 40*10+1],'Parent',fig,'XTickLabel','','YTickLabel','',...
        'XColor',[1 1 1],'YColor',[1 1 1]);

basel=axes('Unit','pixel','Position',[x y 80 400+1],'Parent',fig,'XTickLabel','','YTickLabel','',...
        'XColor',[1 1 1],'YColor',[1 1 1]);
Nodi=[];
for i=1:numeronodi


    plusminus=axes('Unit','pixel','Position',[x y+(i-1)*40+(10-numeronodi)*40 40 40],'Parent',fig,'XTickLabel','','YTickLabel','',...
        'XColor',[1 1 1],'YColor',[1 1 1]);

    picture=axes('Unit','pixel','Position',[x+40 y+(i-1)*40+(10-numeronodi)*40 40 40],'Parent',fig,'XTickLabel','','YTickLabel','',...
        'XColor',[1 1 1],'YColor',[1 1 1]);

    textnode = uicontrol('Style','text', ...
        'Parent',fig,...
        'Units','pixel', ...
        'HorizontalAlignment','left',...
        'Position',[x+81 (y+13)+(i-1)*40+(10-numeronodi)*40 150 17], ...
        'Fontsize',10,...
        'String',noditext{numeronodi-(i-1)},'BackgroundColor',[1 1 1],'Callback','set(gco,''String'',''click far');

    Nodo = struct ('objectgraph',[],...
                   'picture',[],...
                   'textnode',[],...
                   'Children',[],...
                   'status',1);

    Img=image(icone.plus, 'Parent', plusminus);
    set(plusminus, ...
        'Visible', 'off', ...
        'YDir'   , 'reverse'       , ...
        'XLim'   , get(Img,'XData'), ...
        'YLim'   , get(Img,'YData')  ...
        );

    Img=image(PictureNode, 'Parent', picture);
    set(picture, ...
        'Visible', 'off', ...
        'YDir'   , 'reverse'       , ...
        'XLim'   , get(Img,'XData'), ...
        'YLim'   , get(Img,'YData')  ...
        );

    Nodo.objectgraph=plusminus;
    Nodo.picture=picture;
    Nodo.textnode=textnode;


    Nodo.status=1;
    Nodi=[Nodi,Nodo];
end

Nodireverse=[];
for i=numeronodi:-1:1
    Nodireverse=[Nodireverse,Nodi(i)];

end

setappdata(fig,'Nodi',Nodireverse);% handle associato alla struttura dati nodi utilizzata
                                   % per modificare l'oggetto.
if (y-(10-numeronodi)*40)<0
    yt=40;
else
    yt=y-(10-numeronodi)*40;
end
slider = uicontrol('Style','slider', ...
    'Parent',fig,...
    'Units','pixel', ...
    'Position',[x+281 y 20 400],'Max',numeronodi,'Value',numeronodi);
guidata(gcbo, handles);
setappdata(handles.Main,'axeshtree',[base,basel,slider]);
```

```
function [ultimox,ultimoy]=visibile(obj,Nodopadre,posizionepadre)


% Funzione che rende visibile i nodi
nodi=getappdata(obj,'Nodi');
Nodifiglio=nodi(Nodopadre).Children;
dimnodifiglio=size(Nodifiglio);

for i=1:dimnodifiglio(2)
    posizionefiglio=get(Nodifiglio(i).objectgraph,'Position');
    set(Nodifiglio(i).objectgraph,'Position',[posizionefiglio(1) ,posizionepadre(2)-i*40,40,40]);
    set(Nodifiglio(i).picture,'Position',[posizionefiglio(1)+40 ,posizionepadre(2)-i*40,40,40]);
    set(Nodifiglio(i).textchildnode,'Position',[posizionefiglio(1)+80 ,(posizionepadre(2)-i*40)+13,150,17]);
    Nodifiglio(i).status=1;
end
nodi(Nodopadre).Children=Nodifiglio;
posizioneultimo=get(Nodifiglio(i).objectgraph,'Position');
setappdata(obj,'Nodi',nodi);
ultimox=posizioneultimo(1);
ultimoy=posizioneultimo(2);
```

```
function nonvisibile(obj,Nodopadre)

% Funzione che nasconde gli handle spostandoli in posizioni non visibili

nodi=getappdata(obj,'Nodi');
Nodifiglio=nodi(Nodopadre).Children;
dimnodifiglio=size(Nodifiglio);

for i=1:dimnodifiglio(2)
    posizionefiglio=get(Nodifiglio(i).objectgraph,'Position');
    set(Nodifiglio(i).objectgraph,'Position',[posizionefiglio(1) ,2000-(i-1)*40,40,40]);
    set(Nodifiglio(i).picture,'Position',[posizionefiglio(1)+40 ,2000-(i-1)*40,40,40]);
    set(Nodifiglio(i).textchildnode,'Position',[posizionefiglio(1)+80 ,(2000+13)-(i-1)*40,150,17]);
    Nodifiglio(i).status=0;
end
nodi(Nodopadre).Children=Nodifiglio;
setappdata(obj,'Nodi',nodi);
```

```
function spostasuccessivigiu(obj,index,ultimox,ultimoy)

% Aggiorna l'oggetto spostando i successivi nodi rispetto a quello
% selezionato verso il basso (al click su un nodo padre chiuso)

nodi=getappdata(obj,'Nodi');
numeronodipadre=size(nodi);
if numeronodipadre(2)==index
else
    for i=index+1:numeronodipadre(2)
        objectgraphpos=get(nodi(i).objectgraph,'Position');
        set(nodi(i).objectgraph,'Position',[objectgraphpos(1),ultimoy-40,40,40]);
        picturepos=get(nodi(i).picture,'Position');
        set(nodi(i).picture,'Position',[picturepos(1),ultimoy-40,40,40]);
        textnodepos=get(nodi(i).textnode,'Position');
        set(nodi(i).textnode,'Position',[textnodepos(1),ultimoy-40+13,150 17]);
        ultimoy=ultimoy-40;
        nodi(i).status;
        if nodi(i).status==0
            dimfiglio=size(nodi(i).Children);
            Nodifiglio=nodi(i).Children;
            for j=1:dimfiglio(2)
                posizionefiglio=get(Nodifiglio(j).objectgraph,'Position');
                set(Nodifiglio(j).objectgraph,'Position',[posizionefiglio(1) ,ultimoy-40,40,40]);
                set(Nodifiglio(j).picture,'Position',[posizionefiglio(1)+40 ,ultimoy-40,40,40]);
                set(Nodifiglio(j).textchildnode,'Position',[posizionefiglio(1)+80 ,ultimoy-40+13,150,17]);
                ultimoy=ultimoy-40;
            end

        end
    end
end
setappdata(obj,'Nodi',nodi);
```

```
function spostasuccessivisu(obj,index,ultimoy)

% Aggiorna l'oggetto spostando i successivi nodi rispetto a quello
% selezionato verso l'alto (al click su un nodo padre aperto)

nodi=getappdata(obj,'Nodi');
numeronodipadre=size(nodi);

if numeronodipadre(2)==index
else
    for i=index+1:numeronodipadre(2)
        objectgraphpos=get(nodi(i).objectgraph,'Position');
        set(nodi(i).objectgraph,'Position',[objectgraphpos(1),ultimoy-40,40,40]);
        picturepos=get(nodi(i).picture,'Position');
        set(nodi(i).picture,'Position',[picturepos(1),ultimoy-40,40,40]);
        textnodepos=get(nodi(i).textnode,'Position');
        set(nodi(i).textnode,'Position',[textnodepos(1),ultimoy-40+13,150 17]);
        ultimoy=ultimoy-40;
        nodi(i).status;
        if nodi(i).status==0
            dimfiglio=size(nodi(i).Children);
            Nodifiglio=nodi(i).Children;
            for j=1:dimfiglio(2)
                posizionefiglio=get(Nodifiglio(j).objectgraph,'Position');
                set(Nodifiglio(j).objectgraph,'Position',[posizionefiglio(1) ,ultimoy-40,40,40]);
                set(Nodifiglio(j).picture,'Position',[posizionefiglio(1)+40 ,ultimoy-40,40,40]);
                set(Nodifiglio(j).textchildnode,'Position',[posizionefiglio(1)+80 ,ultimoy-40+13,150,17]);
                ultimoy=ultimoy-40;
            end
        end
    end
end
setappdata(obj,'Nodi',nodi);
```

Di seguito il codice che è necessario inserire nella funzione che gestisce l'interazione del mouse con l'handle grafico che contiene l'oggetto *control tree image*.

```
function Main_WindowButtonDownFcn(hObject, eventdata, handles)

icone=load('Iconshtree');
[j,window]=gcbo;

pointwindow=get(window,'Currentpoint')
try
    set(gco,'BackgroundColor',[0 0 1]);
    set(gco,'ForegroundColor',[1 1 1]);
catch
end
nodi=getappdata(handles.Main,'Nodi');
boolean=0;
h=get(gco,'Parent');
dimnodi=size(nodi);
padre=0;
figlio=0;
dimnodichild=size(nodi(1).Children);
for i=1:dimnodi(2)

    if h==nodi(i).objectgraph||h==nodi(i).picture||gco==nodi(i).textnode
        boolean=1;
        padre=i;
        break,
    end

    figli=nodi(i).Children;
    for j=1:dimnodichild(2)

        if h==figli(j).objectgraph||h==figli(j).picture||gco==figli(j).textchildnode
            boolean=2;
            figlio=j;
            padre=i;
            break,
```

```
        end

    end

end
if boolean==1
    if h==nodi(padre).objectgraph
        if nodi(padre).status==1
            Img=image(icone.minus, 'Parent', h);

            set(h, ...
                'Visible', 'off', ...
                'YDir'   , 'reverse'       , ...
                'XLim'   , get(Img,'XData'), ...
                'YLim'   , get(Img,'YData')  ...
                );
            nodi(padre).status=0;
            setappdata(handles.Main,'Nodi',nodi);
            posizionepadre=get(nodi(padre).objectgraph,'Position');

            numfigli=size(nodi(padre).Children);
            spostasuccessivigiu(handles.Main,padre,posizionepadre(1),(posizionepadre(2)-numfigli(2)*40))

            [ultimox,ultimoy]=visibile(handles.Main,padre,get(nodi(padre).objectgraph,'Position'))
            setappdata(handles.Main,'Nodi',nodi);
            size(nodi(padre).Children)
        else
            Img=image(icone.plus, 'Parent', h);

            set(h, ...
                'Visible', 'off', ...
                'YDir'   , 'reverse'       , ...
                'XLim'   , get(Img,'XData'), ...
                'XLim'   , get(Img,'XData'), ...
                'YLim'   , get(Img,'YData')  ...
                );
            nodi(padre).status=1;
            setappdata(handles.Main,'Nodi',nodi);
            objectgraphpos=get(nodi(padre).objectgraph,'Position')
            nonvisibile(handles.Main,padre)

            spostasuccessivisu(handles.Main,padre,objectgraphpos(2))
            setappdata(handles.Main,'Nodi',nodi);
        end
    elseif  gco==nodi(padre).textnode
    end
    setappdata(handles.Main,'Nodi',nodi);
elseif boolean==2
    figli=nodi(padre).Children;
    if gco==figli(figlio).textchildnode
        for i=1:dimnodi(2)
            set(nodi(i).textnode,'BackgroundColor',[1 1 1]);
            set(nodi(i).textnode,'ForegroundColor',[0 0 0]);

            figli=nodi(i).Children;
            for j=1:dimnodichild(2)
                set(figli(j).textchildnode,'BackgroundColor',[1 1 1]);
                set(figli(j).textchildnode,'ForegroundColor',[0 0 0]);

            end
            nodi(i).Children=figli;
        end
        set(gco,'BackgroundColor',[0 0 160/255]);
        set(gco,'ForegroundColor',[1 1 1]);
    else
    end
    setappdata(handles.Main,'numvel',padre)
    setappdata(handles.Main,'numfig',figlio)

    showtraj(handles,padre,figlio)
else

end

setappdata(handles.Main,'Nodi',nodi);
```

Per completezza riporto un esempio di aggiunta di un nodo figlio all'oggetto control tree image realizzato

```
    nodi=getappdata(handles.Main,'Nodi');
    pos=get(nodi(1).objectgraph,'Position');
    x=pos(1);
    dimnodi=size(nodi);
    for i = 1:dimnodi(2)
        manovreitem.NomeManovra=nomemanovra;
        manovreitem.Fig=ManovraAresti;
        manovreitem.Fig.startposition=posizioneformaz(:,i);

        airfigure=[airfigure;manovreitem];
        T=axes('Unit','pixel','Position',[x+40 2000-(j-1)*40 40 40],'Parent',handles.Main,'XTickLabel','','YTickLabel','',...
            'XColor',[1 1 1],'YColor',[1 1 1]);
        %
        Img=image(icone.T, 'Parent', T);

        set(T, ...
            'Visible', 'off', ...
            'YDir'   , 'reverse'       , ...
            'XLim'   , get(Img,'XData'), ...
            'YLim'   , get(Img,'YData')  ...
            );
        %
        picture=axes('Unit','pixel','Position',[x+80 2000-(j-1)*40 40 40],'Parent',handles.Main,'XTickLabel','','YTickLabel','',...
            'XColor',[1 1 1],'YColor',[1 1 1]);

        Img=image(icone.aereoicon2, 'Parent', picture);

        set(picture, ...
            'Visible', 'off', ...
            'YDir'   , 'reverse'       , ...
            'XLim'   , get(Img,'XData'), ...
            'YLim'   , get(Img,'YData')  ...
            );
        textnode = uicontrol('Style','text', ...
              'Parent',handles.Main,...
              'Units','pixel', ...
              'HorizontalAlignment','left',...
              'Position',[x+300 (2000+13)-(j-1)*40 150 17], ...
              'Fontsize',10,...
              'String',nomemanovra,'BackgroundColor',[1 1 1],'Callback','set(gco,''String'',''click far');

         child = struct ('objectgraph',T,...
             'picture',picture,...
             'textchildnode',textnode,...
             'status',0);
         nodi(i).Children=[nodi(i).Children,child];

    end
    AirShow.Manovre= [AirShow.Manovre,airfigure];
    AirShow.Count=AirShow.Count+1;
    setappdata (handles.Main,'AirShow',AirShow);
    setappdata (handles.Main,'Nodi',nodi);

end
nodi=getappdata(handles.Main,'Nodi');
pos=get(nodi(1).objectgraph,'Position');
x=pos(1);
hidden = uicontrol('Style','text', ...
    'Parent',handles.Main,...
    'Units','pixel', ...
    'HorizontalAlignment','left',...
    'Position',[x 10 300 170], ...
    'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));

frameinformation = uicontrol('Style','frame', ...
    'Parent',handles.Main,...
    'Units','pixel', ...
```

```
    'Position',[18 43 291 128], ...
    'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'),...
    'ForeGroundColor',[220 213 184]/255);

textinformation = uicontrol('Style','text', ...
    'Parent',handles.Main,...
    'Units','pixel', ...
    'HorizontalAlignment','left',...
    'Position',[23 48 276 118], ...
    'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));


axestree=getappdata(handles.Main,'axeshtree');
axestree=[axestree,textinformation,hidden];
setappdata(handles.Main,'axeshtree',axestree);
```

Come si può notare dalla figura A.1 riportata sotto l'oggetto realizzato come funzione matlab è del tutto somigliante all'*image control tree Microsoft*

**Figura A .1**