SIMULATION QUICK START GUIDE

The simulations provided for this book are based on original code named ADMIRE (Aero-Data Model In a Research Environment). It was created by the Swedish Defence Research Agency (Svensk Försvarets forskningsinstitut, FOI). At the time of this writing version 4.1 of ADMIRE was available at:
https://www.foi.se/en/our-knowledge/aeronautics-and-air-combat-simulation/admire/downloads.html

The book website contains files for a linearized version of the simulation and a nonlinear version. If readers wish to use only the linearized version with the trim conditions supplied, it is not necessary to download the ADMIRE simulation. If readers wish to use the nonlinear simulation, or if they wish to create linear models from trim conditions not supplied, it is necessary to download and compile the ADMIRE files provided by the FOI from their website.

Requirements:

To run the simulations you need MATLAB/Simulink Release 13 or later. It appears most of the ADMIRE Simulink files were created with an earlier release, and the files may work with earlier releases. However, the authors have not tested the files with earlier releases and make no claims regarding compatibility with earlier releases. To run the nonlinear simulation you will also need an ANSI® C or C++ compiler.

A note on the Simulink files provided:

In R2012a the MathWorks introduced a new file type for Simulink models using a .slx file extension. The Simulink models supplied here are in the earlier .mdl format. This is done in an effort to make the models provided compatible with most versions of Simulink. According to the R2012a Release Notes "The MDL file format will continue to be supported, but, after R2012a, new features might be available only if you use the SLX file format."

The Simulink models provided have had the screen color changed to a color that may be easier on the eyes than the default settings. Since some users may not like this choice of screen color, a MATLAB function file, `recolor.m`, has been provided so users may set the screen color based on their own preferences.

PART 1: USING THE LINEAR SIMULATION

1. Organize the files downloaded from the book website.

Below is a list of files used by the linear simulation. Each name is followed by a brief description. Place all of these files in the same directory for ease of use.

Basic model files:

```
Linear_ADMIRE_sim.mdl    % Simulink model
INIT_LINEAR.m            % MATLAB script file used to initialize variables
                         used by the Simulink model.
```

Control Allocation MATLAB function files used to implement the various control allocation algorithms described in the book:

```
CGIwrap.m           % Implements the Cascading Generalized Inverse algorithm
DAwrap.m            % Implements the Direct Allocation algorithm
LPwrap.m            % Implements various Linear Programming algorithms
VJAwrap.m           % Implements the Vertex Jumping Algorithm
getUperp.m          % Calculates a vector in the null space of the control
                    effectiveness matrix, used in optimizing a cost function
```

Data files containing trim and linear model data:

```
Trim_M0p22ALT20_LinDATA.mat          % Mach = 0.22, Altitude = 20 m
Trim_M0p3ALT2000_LinDATA.mat  % Mach = 0.3, Altitude = 2000 m
```

2. Launch MATLAB

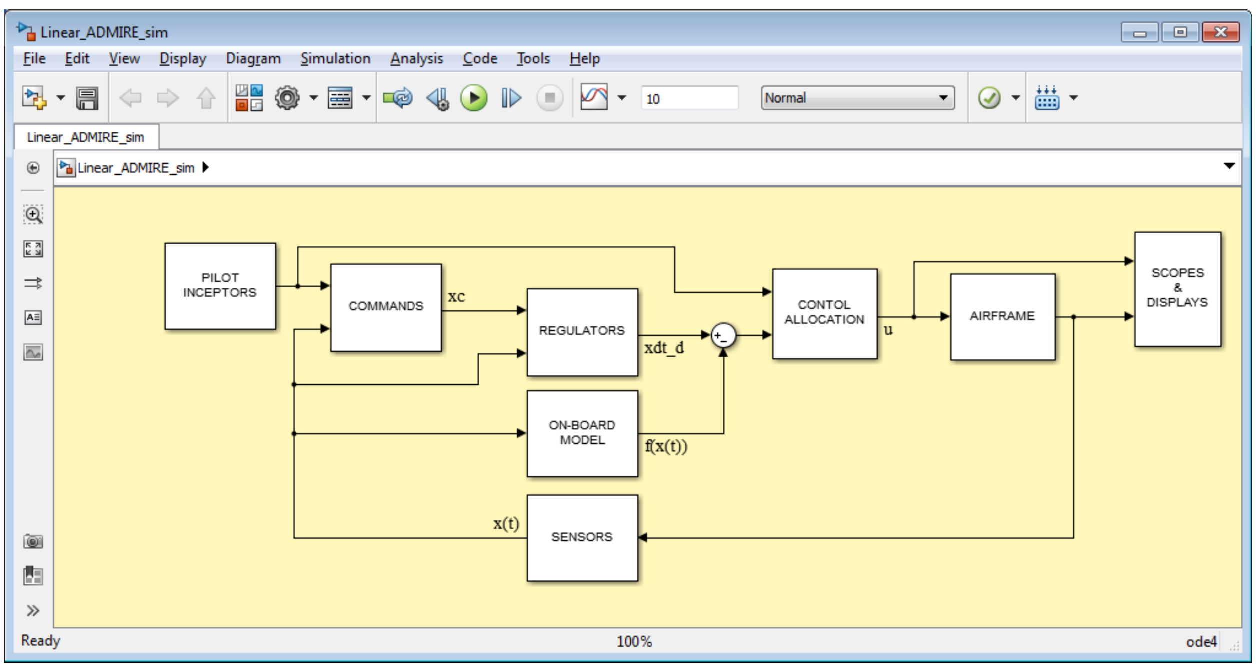
3. Change the current directory to the one containing the downloaded files. You can do this using the cd command or by clicking on the folder icons. For example,

```
>> cd C:\MATLAB\ACAfiles
```

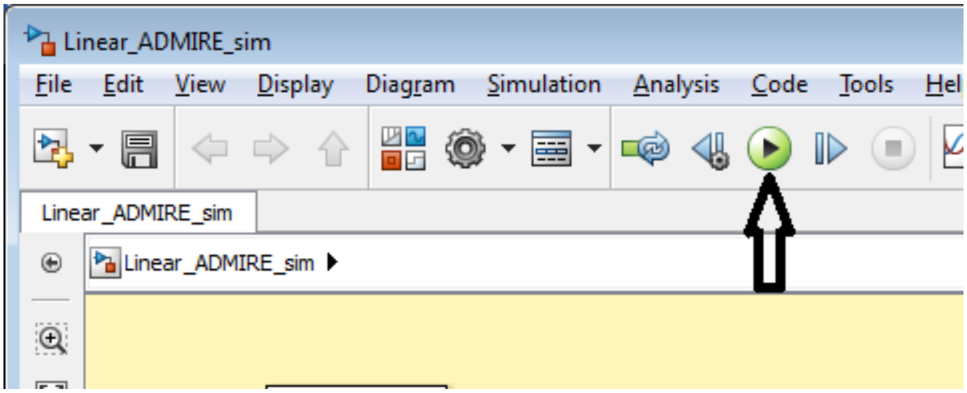
4. Open the file `Linear_ADMIRE_sim.mdl`

There are several methods which can be used to open a model file. One simple method is to type the name (without the file extension .mdl) at the command prompt.

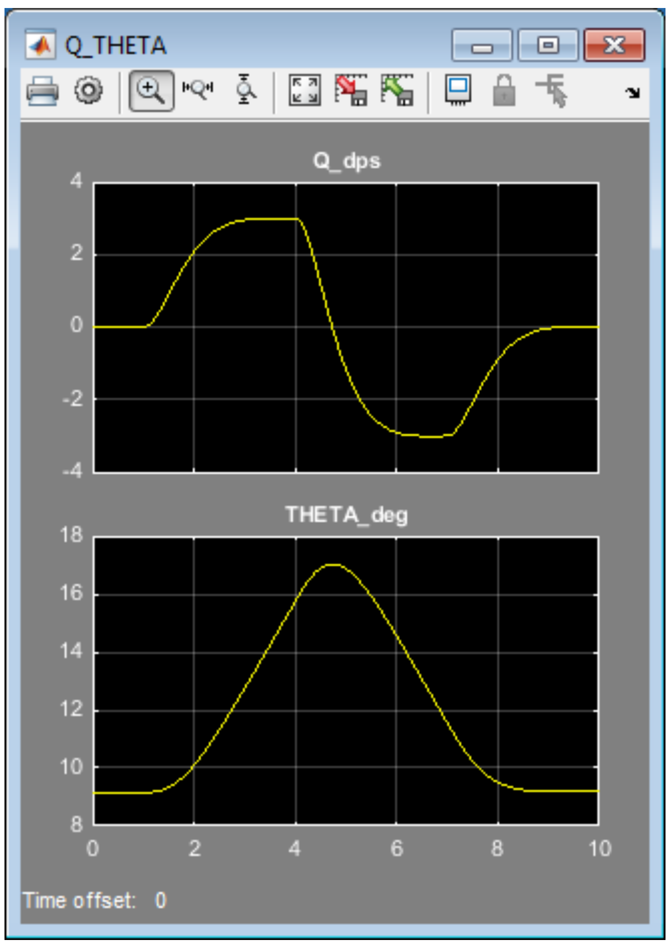
```
>> Linear_ADMIRE_sim
```



5. Run the simulation by pressing the green play button.



6. Open the block labeled "SCOPES & DISPLAYS" by double clicking on it. If you double click on the scope labeled Q_THETA, you should see a window displaying pitch rate and pitch angle that looks like:



If you see this figure then the simulation has successfully run. This is a ten second simulation showing the response to a pilot pitch doublet.

7. Making changes

The variables used by the simulation are defined in the MATLAB script file `INIT_LINEAR.m`. Open this file:

```
>> open INIT_LINEAR
```

This file is divided into several sections. Below is a brief description of some items users may wish to change.

Define the trim file to load in the section `%% LOAD TRIM DATA`. Make sure the file you choose is in the path that MATLAB uses to look for files. For more information on the MATLAB path, type

```
>> help path
```

Define the pilot input time histories in the section `%% PILOT INCEPTORS`. The pilot inceptor input values are implemented using two vectors, one containing time the other containing the value of the inceptor at the corresponding time. The values are denoted with `_v` and the time vectors are denoted with `_t`. If the time vector is not monotonically increasing, an error will occur.

Change control allocation methods in the section `%% CONTROL ALLOCATION`. The variable `CAmethod` is used to select one of the control allocation methods described in the book. If the Linear Programming option is selected (`CAmethod = 6`), then the variable `LPmethod` is used to select from the various linear programming implementations. The algorithms are implemented using functions, so that they may be used independently of the simulation.

PART 2: DOWNLOAD AND COMPILE ADMIRE SIMULATION

This section provides instructions for downloading and using the ADMIRE simulation files.

1. Go to the website:

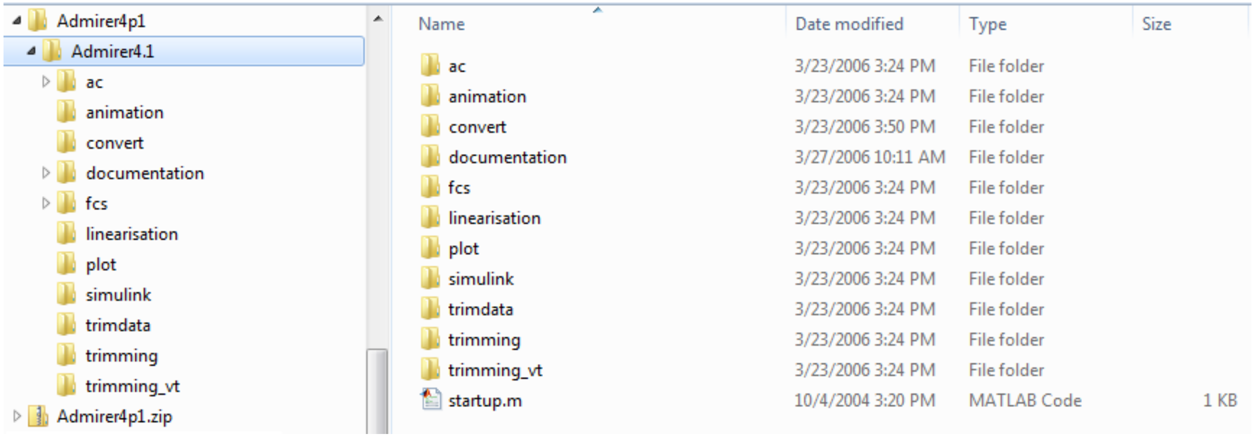
http://www.foi.se/en/Our-Knowledge/Aeronautics/Admire/Downloads/

Download the ADMIRE simulation file:

Admirer4p1.zip

There are other files available on this site, but only Admirer4p1.zip is required for this simulation.

2. Extract the files from the zip archive. This should produce the following directory tree:



3. Launch MATLAB

4. In MATLAB, change directory to the Admirer4.1 directory containing the file `startup.m`

5. Run the script file `startup.m` to setup paths needed for the ADMIRE simulation

```
>> startup
```

6. Compile the C-code used in the aircraft model and fcs model. NOTE: You need to have a C compiler installed on your computer.

Change directory to `\Admirer4.1\ac\c`

Run the script file `acmex.m`

```
>> acmex
```

Change directory to `\Admirer4.1\fcs\c`

Run the file `fcsmex.m`

```
>> fcsmex
```

The ADMIRE simulation can be used to perform several different tasks. Basic instructions for some of these tasks are included in Parts 3 through 5 of this document.

Part 3: Trim the aircraft in steady level flight

Part 4: Create a linear model based on a specified trim condition

Part 5: Run a non-linear simulation starting from a specified trim condition

For more detailed instructions see the documentation provided with the ADMIRE simulation in the directory `\Admirer4p1\Admirer4.1\documentation`.

PART 3: CREATING TRIM FILES

The ADMIRE simulation comes with a script file that can be used to trim the aircraft in level flight given a specified Mach number and altitude in meters.

1. Launch MATLAB

2. In MATLAB, change directory to the Admirer4.1 directory containing the file `startup.m`

3. Run the file `startup.m` to setup paths needed for the ADMIRE simulation

```
>> startup
```

4. Running the trim file that comes with the ADMIRE simulation will generate numerous warning messages. These are not errors and the program should be able to run successfully. An experienced Simulink user may know how to maneuver through the menus of a Simulink model to disable these warning messages. The messages can also be disabled through a series of command line instructions. To disable the warnings the following commands may be executed at the command line prior to executing the trim script.

```
>> open_system('admire_complete_trim')
>> set_param('admire_complete_trim','InitInArrayFormatMsg', 'None')
>> save_system('admire_complete_trim')
>> close_system('admire_complete_trim')
```

The `open_system` command will open a file. Leave this file open while running the other commands. The `close_system` command will close the file.

If you are running a newer version of MATLAB, you may see the following message when the save command runs:

```
A copy of the original file "admire_complete_trim" has been created because
it was last saved in an earlier version of Simulink. To recover the original
version, rename the file "admire_complete_trim.r11" as
"admire_complete_trim".
```

This message may be ignored.

5. Run the script file for a straight and level trim. This script will ask the user to input a Mach number and altitude in meters. The ADMIRE documentation states that the flight envelope is restricted to Mach numbers less than 1.2 and altitudes below 6 km. The trim script will issue warnings if you attempt to trim outside of this range. If you enter an altitude below 20 meters, the simulation will trim at 20 meters. The example below shows the results of running the script to trim the aircraft at Mach 0.3 and 2000 meters.

```
>>admtrim_sl


Mach number [-] : 0.3
Altitude    [m] : 2000

Trimming ADMIRE complete trim.......... DONE!
u0new(1)  = -0.03242 deg      ( Canard deflections , + = trailing edge down )
```

```
u0new(3)  =  1.56615 deg      ( Elevon deflections , + = trailing edge down )
u0new(7)  =  0.00000 deg      ( Rudder deflection  , + = trailing edge left )
u0new(8)  =  0.00000 deg      ( Leading edge defl. , + = leading edge down  )
u0new(10) =  0.05830          ( Throttle setting                            )

After trimming for AoA:
x0(1)  = 99.75875 ( Vt    [m/s]   )
x0(2)  =  6.64769 ( AoA   [deg]   )
x0(3)  =  0.00000 ( Beta  [deg]   )
x0(4)  =  0.00000 ( P     [deg/s] )
x0(5)  =  0.00000 ( Q     [deg/s] )
x0(6)  =  0.00000 ( R     [deg/s] )
x0(7)  =  0.00000 ( Psi   [deg]   )
x0(8)  =  6.64769 ( Theta [deg]   )
x0(9)  =  0.00000 ( Phi   [deg]   )
x0(10) =  0.00000 ( X     [m]     )
x0(11) =  0.00000 ( Y     [m]     )
x0(12) = -2000.00000    ( Z     [m]     )

Decelerating:   -0.3428 m/s^2
Run simulations with 'admire_sim.mdl' and view the results with command
'trimplot'
```

6. Save the trim data to a .mat file.

```
>> save('Trim_M0p3ALT2000')
```

The naming convention used in the trim files supplied is as shown above to indicate the trim Mach number and altitude in meters.

PART 4: CREATING LINEAR MODELS

1. Launch MATLAB

2. In MATLAB, change directory to the Admirer4.1 directory containing the file `startup.m`

3. Run the file `startup.m` to setup paths needed for the ADMIRE simulation

```
>> startup
```

4. Running the linearization file that comes with the ADMIRE simulation will generate numerous warning messages. These are not errors and the program should be able to run successfully. An experienced Simulink user may know how to maneuver through the menus of a Simulink model to disable these warning messages. The messages can also be disabled through a series of command line instructions. To disable the warnings the following commands may be executed at the command line prior to executing the linearization script.

```
>> open_system('admire_bare_lin')
>> set_param('admire_bare_lin','InitInArrayFormatMsg', 'None')
>> save_system('admire_bare_lin')
>> close_system('admire_bare_lin')
>> open_system('admire_fcs_lin')
>> set_param('admire_fcs_lin','InitInArrayFormatMsg', 'None')
```

```
>> save_system('admire_fcs_lin')
>> close_system('admire_fcs_lin')
```

If you are running a newer version of MATLAB, you may see the following messages when the save commands run:

```
A copy of the original file "admire_bare_lin" has been created because it was
last saved in an earlier version of Simulink. To recover the original
version, rename the file "admire_bare_lin.r11" as "admire_bare_lin".

A copy of the original file "admire_fcs_lin" has been created because it was
last saved in an earlier version of Simulink. To recover the original
version, rename the file "admire_fcs_lin.r11" as "admire_fcs_lin".
```

These messages may be ignored.

5. Run the trim script (see Part 3 above) or load in data created from the trim script.

```
>> load('Trim_M0p3ALT2000')
```

6. Run the linearization script.

```
>>adm_lin
```

This script linearizes the plant (bare airframe) and flight control system (fcs) and creates the following 8 matrices:

```
  Name         Size             Bytes  Class     Attributes
  Abare       28x28              6272  double
  Bbare       28x16              3584  double
  Cbare       59x28             13216  double
  Dbare       59x16              7552  double

  Name        Size             Bytes  Class     Attributes
  Afcs        4x4                128  double
  Bfcs        4x40              1280  double
  Cfcs       12x4                384  double
  Dfcs       12x40              3840  double
```

These matrices are used in the standard state-space representation:

$$\dot{\bar{x}} = A\bar{x} + B\bar{u}$$
$$\bar{y} = C\bar{x} + D\bar{u}$$

The linear model supplied with the textbook, `Linear_ADMIRE_sim.mdl`, uses the bare airframe state-space representation to model the aircraft.

$$\dot{\bar{x}} = A_{bare}\bar{x} + B_{bare}\bar{u}$$

$$\bar{y} = C_{bare}\bar{x} + D_{bare}\bar{u}$$

The fcs matrices are not used, because the ADMIRE flight control system has been replaced with a dynamic inversion controller.

For the bare airframe model there are 28 states, $\bar{x}$, 16 input variables, $\bar{u}$, and 59 output variables, $\bar{y}$. The names of these variables are included at the end of script files `INIT_LINEAR.m` and `INIT_NDI.m`.

7. Save the trim and linearization data to a .mat file.

```
>> save('Trim_M0p3ALT2000_LinDATA')
```

The naming convention used in the supplied files appends the name of the trim files to indicate the linear model data is also included.

PART 5: USING THE NONLINEAR SIMULATION

1. Organize the files downloaded from the book website.

Below is a list of files used by the nonlinear simulation.These files are not included in the ADMIRE download, but are these files available from the book website. Each name is followed by a brief description. Place all of these files in the `\Admirer4p1\Admirer4.1\simulink` directory for ease of use.

Basic model files:

```
admire_sim_NDI.mdl        % Simulink model containing Nonlinear Dynamic
                          Inversion Controller.
INIT_NDI.m                % MATLAB script file used to initialize variables
                          used by the Simulink model.
```

Control Allocation MATLAB function files used to implement the various control allocation algorithms described in the book:

```
CGIwrap.m          % Implements the Cascading Generalized Inverse algorithm
DAwrap.m           % Implements the Direct Allocation algorithm
LPwrap.m           % Implements various Linear Programming algorithms
VJAwrap.m          % Implements the Vertex Jumping Algorithm
getUperp.m         % Calculates a vector in the null space of the control
                   effectiveness matrix, used in optimizing a cost function
```

Data files containing trim and linear model data:

```
Trim_M0p22ALT20_LinDATA.mat         % Mach = 0.22, Altitude = 20 m
Trim_M0p3ALT2000_LinDATA.mat        % Mach = 0.3, Altitude = 2000 m
```

2. Launch MATLAB

3. In MATLAB, change directory to the Admirer4.1 directory containing the file `startup.m`

4. Run the file `startup.m` to setup paths needed for the ADMIRE simulation

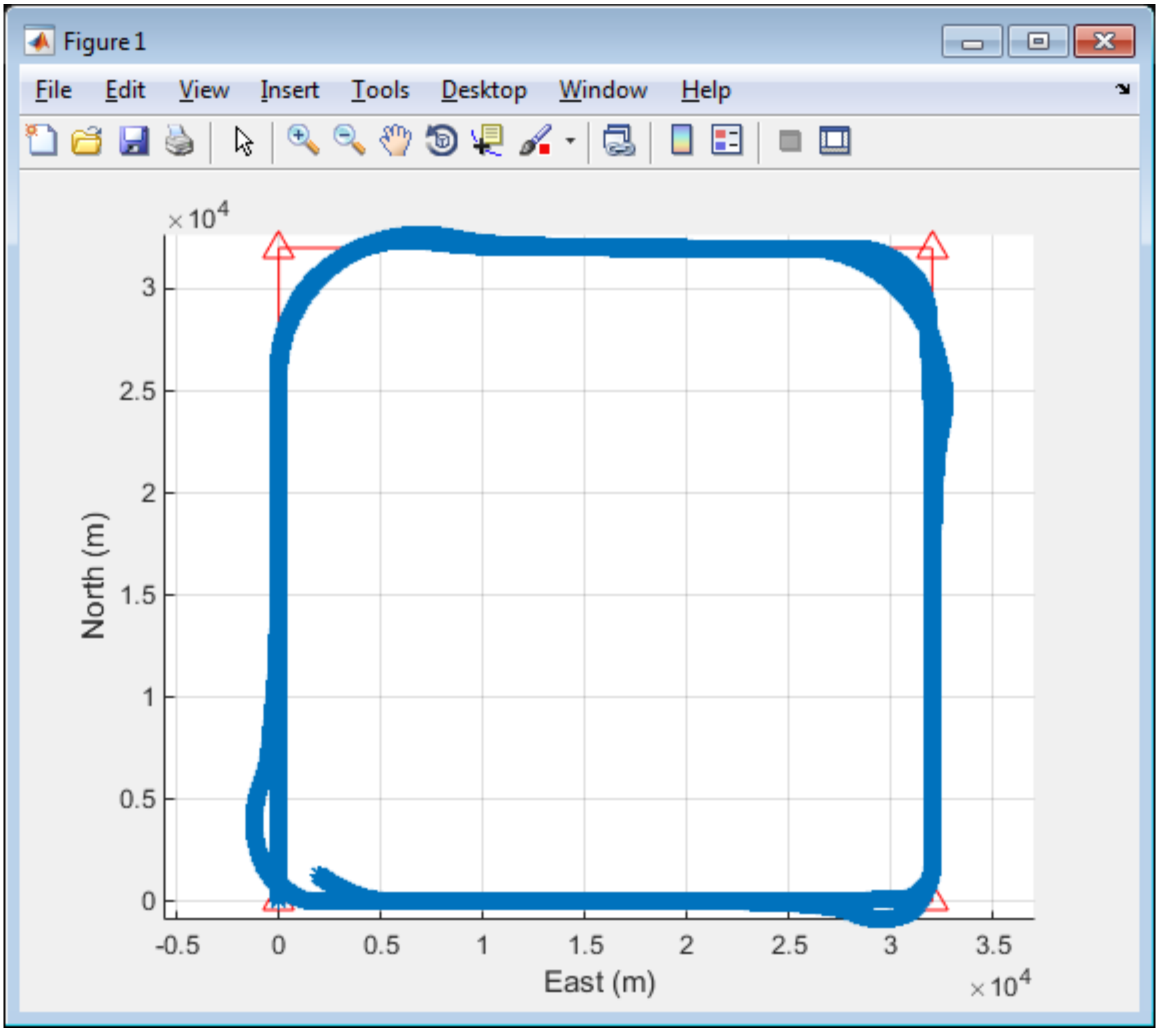
```
>> startup
```

5. Open the Simulink model file `admire_sim_NDI.mdl`

```
>> admire_sim_NDI
```

6. Run the simulation by pressing the green play button.

7. The simulation is set up to run for 1000 seconds, flying the aircraft in a rectangular pattern using an auto pilot. The aircraft flies through most of the available flight envelope starting at a low speed low altitude condition and ending at a high speed high altitude condition. If the simulation runs successfully, the following plot should appear when the simulation stops.



8. Making changes

The variables used by the simulation are defined in the MATLAB script file `INIT_NDI.m`. This file is divided into several sections similar to the file `INIT_LINEAR.m` described in Part 1. A detailed description of this simulation is included in Appendix B of the book.