NATIONAL INSTITUTE OF AEROSPACE

Final Report For:
NASA # NNL06AC78T

Task Title:
Wake Vortex Inverse Model User's Guide

Prepared By:
David Y. Lai
Donald P. Delisi

Task Monitor:
Dr. Fred H. Proctor, NASA Technical Monitor

Reporting Period:
October, 2008

NWRA NorthWest Research Associates – www.nwra.com

4118 148th Ave NE, Redmond, WA 98052 Tel: 425.556.9055 Fax: 425.556.9099

NWRA-SEA-08-R351R1 **2 October 2008**

Wake Vortex Inverse Model User's Guide

By

David Y. Lai
and
Donald P. Delisi

Subcontract TO6-6000-NWRA, Subtask 3.3.2

For
NATIONAL INSTITUTE OF AEROSPACE
Dr. David J. Peake, Vice President of Research and Program Development
100 Exploration Way
Hampton, VA 23666-6147

NASA LANGLEY RESEARCH CENTER
Dr. Fred H. Proctor, NASA Technical Monitor
Hampton, VA 23681

Table of Contents

1. Introduction

NorthWest Research Associates (NWRA) has developed an inverse model for inverting landing aircraft vortex data. The data used for the inversion are the time evolution of the lateral transport position and vertical position of both the port and starboard vortices. The inverse model performs iterative forward model runs using various estimates of vortex parameters, vertical crosswind profiles, and vortex circulation as a function of wake age. Forward model predictions of lateral transport and altitude are then compared with the observed data. Differences between the data and model predictions guide the choice of vortex parameter values, crosswind profile and circulation evolution in the next iteration. Iterations are performed until a user-defined criterion is satisfied. Currently, the inverse model is set to stop when the improvement in the rms deviation between the data and model predictions is less than 1 percent for two consecutive iterations. The forward model used in this inverse model is a modified version of the Shear-APA model. A detailed description of this forward model, the inverse model, and its validation are presented in a different report (Lai, Mellman, Robins, and Delisi, 2007).

This document is a User's Guide for the Wake Vortex Inverse Model. Section 2 presents an overview of the inverse model program. Execution of the inverse model is described in Section 3. When executing the inverse model, a user is requested to provide the name of an input file which contains the inverse model parameters, the various datasets, and directories needed for the inversion. A detailed description of the list of parameters in the inversion input file is presented in Section 4. A user has an option to save the inversion results of each lidar track in a mat-file (a condensed data file in Matlab format). These saved mat-files can be used for post-inversion analysis. A description of the contents of the saved files is given in Section 5. An example of an inversion input file, with preferred parameters values, is given in Appendix A. An example of the plot generated at a normal completion of the inversion is shown in Appendix B.

2. Overview of the Inverse Model Program

The Wake Vortex Inverse Model was developed using Matlab Version 5.3.0 (R11). The core of the inverse model is in the m-file "vortexinverse16.m," which performs all the necessary inverse model iterations. The forward model, called from vortexinverse16, is in the form of a Dynamic-Link Library (DLL), shrapa.dll. This DLL is created by compiling C-coded routines using the Matlab Version 5 MEX functions. In addition to using Matlab Version 5.3, we have also tested the inverse model with Matlab Version 7.0. However, it should be noted that it is possible that shrapa.dll and/or the inverse model may not work properly when a version of Matlab different from Version 5.3.0 (R11) is used.

To facilitate efficient execution of the inverse vortex model, a Matlab program is written to (1) provide the necessary data and model parameters for the inverse model, (2) perform plotting of the data and inverse model results, and (3) save the inverse model results. This program is "inverse_lidar_ver7.m" which works with both the SFO and DEN lidar data.

2.1 Mat-files of Lidar Data

There are two prerequisites before performing the vortex data inversion. The first prerequisite is the conversion of the original lidar data files into a mat-file. The purpose of this conversion is to facilitate uniform and efficient access to the lidar data. The conversion of the SFO OGE and DEN OGE lidar data has been done, and they are provided to the users in two separate mat-files:

"sfo_lidar_oge.mat" and
"den_lidar_270deg_2006.mat".

The first file contains the OGE lidar data taken by the FAA at SFO in September 2001. The second file contains the OGE lidar data taken by the FAA at DEN from April to June 2006. In addition, crosswind profiles taken by the lidar before and after each lidar vortex track have also been converted to a mat-file. These lidar-measured crosswind profiles are not used in the inversion, but they are plotted at the end of the inversion for comparison with the crosswind profiles obtained from the inverse model. The two lidar-measured wind mat-files corresponding to the above two lidar data mat-files are

"sfo_lidarwind_oge.mat" and
"den_lidarwind_270deg_2006.mat".

2.2 Initial Vortex Separation and Circulation Database

The second prerequisite is to have a database of initial vortex separation (b_0) and initial circulation (Γ_0) for different types of aircraft. These parameters are used as initial guesses for the inverse modeling. The preferred method for providing these parameters is to obtain an estimate of the initial vertical descent rate (V_0) for each vortex track and then calculate the b_0 and Γ_0 based on conservation principles (Betz). These estimations and computations have been performed for the SFO and DEN OGE lidar data for the most frequent aircraft. The two databases are provided in comma-delimited files:

ddd_OGE_xxxx_v0b0.csv

where ddd represents the airport (SFO or DEN), and xxxx the aircraft type (e.g. B747). Details of the methodology in obtaining these estimates are presented in a different report (Lai and Delisi, 2007).

Instead of providing the b_0 and Γ_0 for each aircraft, a user has the option to use the elliptical vortex separation and circulation as initial guesses for b_0 and Γ_0 for the inverse modeling. This option is not recommended, since we are discovering that an elliptical value of b_0 may be in error. A database of elliptical parameters for 11 aircraft is provided in a comma-delimited file "AircraftSpecs_NWRA_ellip.csv". It should be noted that the elliptical circulation shown in this database is based on a specific air density, which may not be correct for different airports. Therefore, the elliptical circulation in the database is not used in the inversion. Instead, the elliptical circulation for a particular airport is computed in the inverse model using the density of air provided in the inversion input parameter file.

2.3 Selection of Lidar Tracks for Inversion

The Wake Vortex Inverse Model can be executed for a single track at a time, or for a group of tracks. A user has two options in choosing the lidar vortex tracks. The first option is to identify the tracks using the parameter runlist in the input file. This option requires knowing the exact filenames of the lidar data tracks to be used.

The second option uses the sequential number of the lidar tracks in the lidar data mat-file to identify the tracks for inversion. In this option, a user does not need to know exact filenames of the lidar data tracks. Instead, the two parameters in the input file: ntrk_start and delta_ntrk, are used. Inversion is performed starting with the ntrk_start track in the mat-file and ends after delta_ntrk tracks.

For either option, the chosen lidar tracks have to be associated with aircraft of the same/similar model. This requirement is accomplished by using the group_flag parameter in the input parameter file, and putting the designated runlist of tracks under the correct group flag.

2.4 List of m-files Needed for Inverse Model

The following m-files and DLLs are needed for the inverse model:

inverse_sfo_lidar_ver7.m vortexinverse16.m
get_stindex_stl.m pmodel_equivalence.m
get_sfo_all_v0b0_ver2.m get_v0b0.m
get_sfo_all_v0b0.m get_elliptical.m
get_vortex_separation.m plot_inverse_ver5.m
in_inverse_sfo_lidar_oge_all.m (other input parameter files can be used)
shrapa.dll convert.dll

The following data mat-files are included:

sfo_lidar_oge.mat sfo_lidarwind_oge.mat
den_lidar_270deg_2006.mat den_lidarwind_270deg_2006.mat

The following data files are needed for the inverse model:

AircraftSpecs_NWRA_ellip.csv
SFO_OGE_B733_v0b0.csv SFO_OGE_B738_v0b0.csv
SFO_OGE_B747_v0b0.csv SFO_OGE_B757_v0b0.csv
SFO_OGE_B767_v0b0.csv SFO_OGE_B777_v0b0.csv
SFO_OGE_A319_v0b0.csv SFO_OGE_A320_v0b0.csv
SFO_OGE_DC9_v0b0.csv SFO_OGE_DC10_v0b0.csv
DEN_OGE_B733_v0b0.csv DEN_OGE_B738_v0b0.csv

DEN_OGE_B747_v0b0.csv	DEN_OGE_B757_v0b0.csv
DEN_OGE_B767_v0b0.csv	DEN_OGE_B777_v0b0.csv
DEN_OGE_A319_v0b0.csv	DEN_OGE_A320_v0b0.csv

3. Inverse Model Execution

In Matlab, execute the inverse model by entering
>>Inverse_sfo_lidar_ver7 <Return>

A prompt will appear stating
>>Enter m-filename with parameters:

Enter the name of the input parameter file, such as
>> in_inverse_sfo_lidar_oge_all <Return>

(The filename entered has to be an m-file, i.e. with a .m extension. A user can create different files, each of which is associated with different sets of data or parameters, and enter the appropriate filename accordingly.)

The program will start executing the inverse model. For a normal completion of the inversion, a plot showing the inverse results and the observed data will be generated. If the save plot option is chosen, the plot on the computer screen will disappear and a tiff file of the plot will be saved in a designated directory. Otherwise, the plot remains on the computer screen. If the save file option is chosen, the inversion results will be saved in a mat-file in a designated directory.

Occasionally, the inverse model may end abnormally (i.e. crash). Crashes of the inverse model are mostly due to, in the search of the optimal model parameters, the exceedance of the parameter regimes provided by the data. Currently, we suggest skipping the inversion of the lidar tracks that cause these crashes. These lidar tracks can be identified by knowing the sequential number of the track (nf) in the inverse model run sequence by entering nf and <Return> after the crash. If the runlist option is chosen, skipping of this track is done by deleting the nf^{th} track name in the runlist. If the ntrk_start and delta_ntrk option is used, run the inverse model after changing the parameter ntrk_start to be have a value of nf+1.

4. Parameter Input File

This section provides detailed descriptions of the parameters and options in the input file for the inverse modeling. An example of the input file, with standard data files and preferred parameter values, is shown in Appendix A.

lidar_file	Name of lidar data mat-file, e.g. 'sfo_lidar_oge.mat'
lidarwindfile	Name of lidar wind mat-file, e.g. 'sfo_lidarwind_oge.mat'

ac_database	Name of comma-delimited data file of aircraft elliptical parameters e.g. 'AircraftSpecs_NWRA_ellip.csv'
loc	Location of airport, e.g. 'SFO' or 'DEN'
tile	Tile, e.g. 'OGE'
ellip_flag	Defines initial guesses of b_0 and Γ_0 used in the inversion 1 = uses elliptical b_0 and Γ_0 0 = uses b_0 and Γ_0 provided in the v0b0_file
plot_lidarwind_flag	1 = plot lidar wind profiles on the inversion plot 0 = no plotting of lidar wind profiles
saveplot_flag	1 = save the plot of inversion results as a tiff file. The plot will disappear from the computer screen when this option is chosen. The plot filename is in the form of ddd_Wnnnnnnnnnnnn_mm_inv_ps_xxxx.tif where ddd is the airport location, nnnnnnnnnnnn is the track number, mm is the lidar angle, and xxxx is the aircraft model. 0 = plot is not saved and remains on the computer screen.
savefile_flag	1 = save inversion results in a mat-file, one file for each lidar track. The name of the mat-file is in the form of ddd_inverse_Wnnnnnnnnnnnn_mm_ps_xxxx.mat. A list of the parameters saved is given in Section 5.
group_flag	Select the aircraft model for inversion 1 for B733 (B733 to B735), 2 for B738, 3 for DC10, 4 for DC9, 5 for B747, 6 for B777, 7 for B757; 8 for B767, 9 for A319, 10 for A320
ntrk_start, delta_ntrk	These two parameters are used when data in the lidar_file mat-file is used to define the tracks for inversion. Inversion is performed starting with the ntrk_start track in the mat-file and ends after delta_ntrk tracks. See the parameter runlist for an alternative method to define lidar tracks for inversion.
cwobs_flag	0 = use zero crosswind as the initial guess for inversion (standard). 1 = use observed lidar crosswwind as the initial crosswind guess. This option only works for earlier versions of the inverse model. In this current version, the value of 0 should be used. (In practice, this flag is not essential, since a value of crosswind close to the lidar crosswind is determined by the model in the first iteration.)

The next 23 items are inverse model parameters. Their recommended values are listed.

maxiter = 15;	Maximum number inversion iterations allowed
wcond = 1e-11;	Damping
wvst = 3200;	Crosswind curvature constraint
wtdc = 1;	Crosswind offset constraint
wtsl = 1;	Crosswind slope constraint
wgst = 5;	Circulation constraint
wtgamdc = 0;	DC offset constraint for circulation as a function of age
wtgamsl = 0;	Linear slope constraint for circulation as a function of age
bzst = 10;	b0 constraint

bgst = 500;	b0*gamma0 (Betz) constraint
wyst = 0.1;	y0 constraint
wzst = 0.1;	z0 constraint
wtsep = 1;	Vortex separation (as a function of age) constraint
t_setup = 0;	Define the wake age (in seconds) below which the weights wttys, wttyp, wttzs and wttzp are used. A value of zero means weights wtys, wtyp, wtzs and wtzp are used for all wake ages.
wttys = 1;	Weight for starboard vortex lateral transport for wake age<t_setup
wttyp = 1;	Weight for port vortex lateral transport for wake age<t_setup
wttzs = 1;	Weight for starboard vortex altitude for wake age<t_setup
wttzp = 1;	Weight for port vortex altitude for wake age<t_setup
wtys = 1;	Weight for starboard vortex lateral transport for all wake ages
wtyp = 1;	Weight for port vortex lateral transport for all wake ages
wtzs = 1;	Weight for starboard vortex altitude for all wake ages
wtzp = 1;	Weight for port vortex altitude for all wake ages
zfix = 0;	1=fix z0 during inversion, 0=let z0 change during inversion This option is no longer used in this version.

The next 6 parameters vary according to the group_flag used.

runlist	Define a list of tracks for inversion, e.g. ['W092401030701.28.trk'; 'W092401191800.28.trk'; 'W092401192022.28.trk'] defines three lidar tracks for inversion. There is no limit on the total number of tracks able to be defined. However, when using this option, the tracks defined must be from the same type of aircraft as defined in the group_flag parameter. Otherwise, the inversion will abort. When runlist does not exist in the input file (e.g. commented out), the lidar tracks in the lidar data mat-file, together with ntrk_start and delta_ntrk, are used to define the tracks for inversion.
pmodel	Defines the aircraft names used in the mat-file. This parameter list is necessary because of the different ways an aircraft is named in the data. A user does not need to deal with this parameter list unless datasets other than the SFO and DEN OGE lidar data are used.
plotdirectory	Defines the directory where the tiff files of plots of the inversion results are to be stored, e.g. 'c:\ SFO_lidar_OGE_B733_plots\'. This parameter is ignored if saveplot_flag is not set.
inv_directory	Defines the directory where the mat-files of the inversion results are to be stored, e.g. 'c:\ SFO_lidar_OGE_B733_inversion\'.
v0b0_file	Name of file containing the appropriate V_0 and b_0 for each lidar track, e.g. 'SFO_OGE_B733_v0b0.csv';
plots_grouplabel	Used for labeling plot files
plots_grouplabel	= [plots_grouplabel,pmodel(1,:)] is used to identify mat-files and plots of inversion results

The rest of the parameters deal with airport, runways, and lidar location.

runway_left	Name of left runway, e.g. '28L' for SFO

runway_right	Name of right runway, e.g. '28R' for SFO
runway_angle	Orientation (degrees True) of runway, e.g. 282.2 for SFO
rho_air	Air density (kg/m^3) at airport, e.g. 1.2 for SFO, 1.06 for DEN
m2feet	Conversion factor from meters to feet, 3.2808
y0_lidar	Lidar lateral position (in meters) relative to the runway centerline, e.g. -3450/m2feet for SFO OGE data
y0_left	Lateral position (meters) of left runway relative to the left runway
y0_right	Lateral position (meters) of right runway relative to the left runway e.g. 750/m2feet for SFO
z0_lidar	Lidar altitude (in meters).
z0_left	Altitude of the ground under the intersection between the extended left runway and the lidar range plane.
z0_right	Altitude of the ground under the intersection between the extended right runway and the lidar range plane.
	The last three parameters are used to adjust the vortex altitude measured by the lidar

5. Mat-file of Inversion Results

A user has the option of saving the inversion results of a lidar vortex track in a mat-file, which can later be used for post-inversion analysis. The content of this mat-file is listed below.

loc	Airport
header_note1	Header note for this file
header_note2	Additional header note for this file
lidar_str	Identify lidar track
model_case	A/C model for this track
bz_init	Initial guess of vortex separation (b_0) used in the inversion
gamma_init	Initial guess of vortex circulation used in the inversion
yz, zz, wz	Initial guess of vortex lateral position, altitude and vertical descent rate used in the inversion
agep, yp, zp, cp	lidar observations of wake age, lateral transport, altitude and circulation for the port vortex
ages, ys, zs, cs	lidar observations of wake age, lateral transport, altitude and circulation for the starboard vortex
tpmodel, ypmodel, zpmodel, cpmodel	wake age, lateral transport, altitude and circulation for the port vortex from the inversion
tsmodel, ysmodel, zsmodel, csmodel	wake age, lateral transport, altitude and circulation for the starboard vortex from the inversion
z_cw, cw	altitude and crosswind speed (m/s) of the crosswind profile from the inversion
b0, y0, z0	Initial vortex separation, lateral location, and altitude from the inversion

controlparams	List of inverse model parameters used in the inversion
rmserr	rms difference between the observations and the inverse model results
niter	Total number of iterations performed in the inversion

References

Lai, D.Y., G. Mellman, R. Robins and D.P. Delisi, 2007. Wake vortex inverse model and its validation. NorthWest Research Associates Technical Report NWRA-Bell-07-R359.

Lai, D.Y. and D.P. Delisi, 2007. Initial wake vortex separation estimates. NorthWest Research Associates Technical Report NWRA-Bell-07-R353.

Appendix A: An example of the input file used for vortex inverse model

```
% input parameters for inversion of SFO lidar data
%
lidar_file = 'sfo_lidar_oge.mat';
lidarwindfile = 'sfo_lidarwind_oge.mat';
ac_database = 'AircraftSpecs_NWRA_ellip.csv';
%
loc = 'SFO';
tile = 'OGE';
%
ellip_flag = 0;
plot_lidarwind_flag=1;
saveplot_flag = 0;          %1 = save plot
savefile_flag = 0;          %1 = save result in a matfile
%
% select A/C group for inversion
%       1=733, 2=738, 3=DC10, 4=DC9, 5=747, 6=777, 7=757; 8=767; 9=A319;
10=A320
group_flag=4;
ntrk_start = 1;
delta_ntrk = 20;
%
cwobs_flag=0;         %always use 0
%
% ===== inverse model parameters ======
maxiter = 15;         % max number iterations allowed
wcond = 1e-11;        % damping
wvst = 3200;          % crosswind curvature constraint
wtdc = 1;             % crosswind offset constraint
wtsl = 1;             % crosswind slope constraint
wgst = 5;             % circ. constraint
wtgamdc = 0;          % gamma(t) dc offset constraint
wtgamsl = 0;          % gamma(t) linear slope constraint
bzst = 10;            % b0 constraint
bgst = 500;           % b0*gamma0 constraint
wyst = 0.1;           % y0 constraint
wzst = 0.1;           % z0 constraint
wtsep = 1;            % separation(t) constraint
t_setup = 0;          % time separates weights used in inversion
%
wttys = 1;            %weight for ys for time<t_setup;
wttyp = 1;            %weight for yp for time<t_setup;
wttzs = 1;            %weight for zs for time<t_setup;
wttzp = 1;            %weight for zp for time<t_setup;
wtys = 1;             %weight for ys for all times;
```

```
wtyp = 1;                %weight for yp for all times;
wtzs = 1;                %weight for zs for all times;
wtzp = 1;                %weight for zp for all times;
zfix = 0;                %1=>z0 fixed, 0=>z0 float;
% ====================================
%
if group_flag==1;
  %runlist = ['W091001031001.28.trk'; 'W091001161438.28.trk'];
  pmodel = ['B733 '; 'B732 '; 'B734 '; 'B735 '; '737_2'; '737_3'; '737_4'; '737_5';];
  plotdirectory = 'c:\home-d\SFO_plots\SFO_lidar_OGE_B733_inversion\';
  inv_directory = 'c:\home\mat-vortinv\inv_SFO_B733\';
  v0b0_file = 'SFO_OGE_B733_v0b0.csv';
elseif group_flag==2;
  pmodel = ['B737 '; 'B738 '; '737_7'; '737_8'];
  plotdirectory = 'c:\home-d\SFO_plots\SFO_lidar_OGE_B738_inversion\';
  inv_directory = 'c:\home\mat-vortinv\inv_SFO_B738\';
  v0b0_file = 'SFO_OGE_B738_v0b0.csv';
elseif group_flag==3;
  pmodel = ['DC10'; 'DC_1'];
  plotdirectory = 'c:\home-d\SFO_plots\SFO_lidar_OGE_DC10_inversion\';
  inv_directory = 'c:\home\mat-vortinv\inv_SFO_DC10\';
  v0b0_file = 'SFO_OGE_DC10_v0b0.csv';
elseif group_flag==4;
  pmodel = ['DC9 '; 'DC_9'];
  plotdirectory = 'c:\home-d\SFO_plots\SFO_lidar_OGE_DC9_inversion\';
  inv_directory = 'c:\home\mat-vortinv\inv_SFO_DC9\';
  v0b0_file = 'SFO_OGE_DC9_v0b0.csv';
elseif group_flag==5;
  pmodel = ['747'; 'B74'];
  plotdirectory = 'c:\home-d\SFO_plots\SFO_lidar_OGE_B747_inversion\';
  inv_directory = 'c:\home\mat-vortinv\inv_SFO_B747\';
  v0b0_file = 'SFO_OGE_B747_v0b0.csv';
elseif group_flag==6;
  pmodel = ['777'; 'B77'];
  plotdirectory = 'c:\home-d\SFO_plots\SFO_lidar_OGE_B777_inversion\';
  inv_directory = 'c:\home\mat-vortinv\inv_SFO_B777\';
  v0b0_file = 'SFO_OGE_B777_v0b0.csv';
elseif group_flag==7;
  pmodel = [ '757'; 'B75'];
  plotdirectory = 'c:\home-d\SFO_plots\SFO_lidar_OGE_B757_inversion\';
  inv_directory = 'c:\home\mat-vortinv\inv_SFO_B757\';
  v0b0_file = 'SFO_OGE_B757_v0b0.csv';
elseif group_flag==8;
  pmodel = ['767'; 'B76'];
  plotdirectory = 'c:\home-d\SFO_lidar_OGE_B767_inversion\';
  inv_directory = 'c:\home-d\inv_SFO_OGE_B767\';
```

```
  v0b0_file = 'SFO_OGE_B767_v0b0.csv';
elseif group_flag==9;
  pmodel = [ 'A319 '; 'A_319'];
  plotdirectory = 'c:\home-d\SFO_lidar_OGE_A319_inversion\';
  inv_directory = 'c:\home-d\inv_SFO_OGE_A319\';
  v0b0_file = 'SFO_OGE_A319_v0b0.csv';
elseif group_flag==10;
  pmodel = [ 'A320'];
  plotdirectory = 'c:\home-d\SFO_lidar_OGE_A320_inversion\';
  inv_directory = 'c:\home-d\inv_SFO_OGE_A320\';
  v0b0_file = 'SFO_OGE_A320_v0b0.csv';
end
plots_grouplabel = 'ps_';                 % for label plotfiles
plots_grouplabel = [plots_grouplabel,pmodel(1,:)];          %label matfile and plots
%
% ========== SFO Lidar Data and runway paramters ====================
runway_left='28L';
runway_right='28R';
runway_angle = 282.2;                     %for SFO
rho_air = 1.2;                            %kg/m^3
%
m2feet = 3.2808;                          % m to feet
y0_lidar = -3450/m2feet;                  %approx lidar location in meters
z0_lidar = 0;
z0_left = 0;
z0_right = 0;
y0_left = 0;
y0_right = 750/m2feet;
%
```

Appendix B: Example of a plot of inversion results generated at a normal completion of an inversion run.