LASER INTERFEROMETER GRAVITATIONAL WAVE OBSERVATORY

LIGO Laboratory / LIGO Scientific Collaboration

LIGO-E1300480 *LIGO* 3 Oct 2013

Software Manual for the Vacuum System Toolbox

Derek Bridges

Distribution of this document:
LIGO Science Collaboration

This is an internal working note
of the LIGO Project.

California Institute of Technology
LIGO Project – MS 18-34
1200 E. California Blvd.
Pasadena, CA 91125
Phone (626) 395-2129
Fax (626) 304-9834
E-mail: info@ligo.caltech.edu

Massachusetts Institute of Technology
LIGO Project – NW17-161
175 Albany St.
Cambridge, MA 02139
Phone (617) 253-4824
Fax (617) 253-7014
E-mail: info@ligo.mit.edu

LIGO Hanford Observatory
P.O. Box 1970
Mail Stop S9-02
Richland, WA 99352
Phone (509) 372-8106
Fax (509) 372-8137

LIGO Livingston Observatory
P.O. Box 940
Livingston, LA 70754
Phone (225) 686-3100
Fax (225) 686-7189

http://www.ligo.caltech.edu/

1 Introduction

This document lists and describes the functions included in the LIGO Vacuum System Toolbox. The Vacuum System Toolbox is intended to be available on any CDS computer, or by anyone who wants to download it to his or her own computer. All of the functions below are written in Matlab® code, so Matlab® is required to use the Vacuum System Toolbox (CDS computers should have Matlab® installed by default).

2 Abilities, Restrictions and Assumptions

2.1 Abilities of the Vacuum System Toolbox

The Vacuum System Toolbox has the ability to simulate a number of variables as a time series and as a function of distance along the beamtube. These variables, which include pressure, temperature, and outgassing rate, can then be presented as graphical plots or as data files, so that the data can be preserved and used at a later date.

This simulation includes molecular motion, pumping, outgassing, adsorption (surface physics), and leaks. Variables such as beamtube radius and temperature can be varied over the length of the beamtube and over time, provided that the user supplies a schedule defining the variation. The simulation is generally intended to be used for water vapor, but other gases can be substituted if certain properties of the gas (such as available pumping speed and other properties related to surface physics) are known or assumed.

The Vacuum System Toolbox also has the ability to calculate the position of a leak along the beamtube, given a time history of the measured pressures at the both ends of the beamtube. The position of the leak can be calculated using the time history of the measured pressures with the data taken simultaneously (the dynamic method) or with the data taken at one end with the valve closed at the other end and vice versa (the valve modulation method).

2.2 Restrictions of the Vacuum System Toolbox

Currently, the Vacuum System Toolbox is able to simulate pumping at the ends of the beamtube only. This restriction could be circumvented by dividing the beamtube at pump locations other than the ends.

For the leak localization, the Vacuum System Toolbox currently assumes that there is one leak along the beamtube. Multiple leaks can be simulated, but multiple leaks cannot be located without pumps at locations other than ends of the beamtube.

2.3 Assumptions Used in the Vacuum System Toolbox

For the current simulation routine used in Vacuum System Toolbox, the following assumptions apply:

- The effects of baffles are ignored (baffles reduce the beamtube radius for a very small distance along the beamtube, which reduces pumping speed)

- Surface physics effects are based on water vapor, unless data on other gases can be found or calculated (i.e. the number of monolayers of water vapor on the surface of the beamtube is assumed to be 150 prior to bakeout or 6 after bakeout).

For the leak localization, the main assumption is that one leak exists between each pair of pumps (as described in the section above).

3 Functions and Their Descriptions

3.1 Simulation of Conditions Within the Beamtube

3.1.1 Graphical User Interface Functions

The graphical user interface (GUI) functions are used to allow a user to select options without having to know the syntax of the calculation functions.

3.1.1.1 Main Interface Function (vacuumsim.m)

This function brings up the main interface screen that shows the list of all options, along with spaces to enter values or specify choices.

3.1.1.2 Execution Function (VSexec.m)

This function (which is executed by pressing a button on the GUI) is a wrapper for the calculation and plotting functions. The execution function reads in values from the GUI, checks them for validity and consistency, and passes them to the calculation and plotting functions by calling other functions listed below.

3.1.1.3 Default Settings Function (VSdefault.m)

This function allows the user to change the settings in the GUI to a default set of values.

3.1.1.4 Settings Check Function (VSsettings.m)

This function checks the user-defined properties to be sure that they are consistent.

3.1.1.5 Initialization Function (VSinit.m)

This function reads the user-defined properties from the GUI, stores them for use in the calculation and plotting functions, and writes the values to a text file for later use.

3.1.2 Calculation Function (VSpressure.m)

This calculation function defines a differential equation for pressure in the form dp/dt = p(t,x), where p is pressure, t is time, and x is the distance along the beamtube. The function is given in differential form so that the pressure can be simulated over time using ODE solvers built in to Matlab (such as "ode45"). This function can be used with the GUI or as a standalone function, for users who are more familiar with the Toolbox.

3.1.3 Calculation of Beamtube Pressure

This function will compute the pressure within a beamtube of specified diameter and length as a function of length, given the size and placement of pumps or leaks in the beamtube. This function will be conceptually based on Rai Weiss' "*btleakfind5.for*" Fortran code.

3.1.4 Simulation of Surface Physics Within the Beamtube

This function will compute the effects of outgassing (including effects on pressure) within a beamtube of specified diameter and length. This function can be used for a number of gases, although at first, the gas will be assumed to be water vapor. The physics used in these calculations will be conceptually based on Rai Weiss' "btwaterdistribution2a.for", "*btwaterdistsurf2b.for*", and "*waterbakesm.for*" Fortran codes.

3.1.5 Combined Calculation of Beamtube Pressure and Simulation of Surface Physics

Although the functionality for computing beamtube pressure without simulating the effects of outgassing will be developed first, the pressure function will include hooks where results can be passed back and forth with the outgassing code.

3.2 Plotting Functions

The plotting functions are used to display the results of the calculation functions. Again, they can be used with the GUI, or as standalone functions.

3.2.1 Pressure as a Function of Position Along the Beamtube

This function will create an x-y plot of beamtube pressure versus distance along the beamtube. The plot should also indicate the positions (and possibly strengths) of pumps and leaks along the beamtube. If the outgassing simulation function is used, the effects of outgassing on the pressure will be included.

3.2.2 Visualization of Outgassing Results

This function will provide plots related to results of outgassing; for example, one plot would show pumpdown time versus leak size for a given set of conditions.

3.2.3 Data Storage

While not strictly not plotting functions, these function will deal with the results of the calculation functions – one function will store the results, as well as the initial conditions and another function will read back stored results so that further calculations or plots can be made.

3.3 Leak Localization

3.3.1 Graphical User Interface Functions

The graphical user interface (GUI) functions are used to allow a user to select options without having to know the syntax of the calculation functions.

3.3.1.1 Main Interface Function (leakfind.m)

This function brings up the main GUI screen that shows the list of all options, along with spaces to enter values or specify choices.

3.3.1.2 Execution Function (LLexec.m)

This function (which is executed by pressing a button on the GUI) is a wrapper for the calculation and plotting functions. The execution function reads in values from the GUI, checks them for validity and consistency, and passes them to the calculation and plotting functions by calling other functions listed below.

3.3.1.3 Default Settings Function (LLdefault.m)

This function allows the user to change the settings in the GUI to a default set of values.

3.3.1.4 Settings Check Function (LLsettings.m)

This function checks the user-defined properties to be sure that they are consistent.

3.3.1.5 Initialization Function (LLinit.m)

This function reads the user-defined properties from the GUI, stores them for use in the calculation and plotting functions, and writes the values to a text file for later use.

3.3.2 Calculation Functions

3.3.2.1 Leak Localization Calculation (LLcalc.m)

This function calculates the location of a leak along the beamtube based on pressure readings at the ends of the tube.

3.3.2.2 Pressure Gradient Calculation (LLgradient.m)

This function estimates the change in pressure over time, given a time series of pressure readings.

3.3.3 Plotting Functions (multiple m-files)

The plotting functions are used to display the results of the calculation functions. Again, they can be used with the GUI, or as standalone functions.

3.3.4 Data Storage and Retrieval Functions (multiple m-files)

While not strictly not plotting functions, these function will deal with the results of the calculation functions – one function will store the results, as well as the initial conditions and another function will read back stored results so that further calculations or plots can be made.

4 Bibliography

1.) LIGO Document T880003, “Pressure changes in the beam tubes with temperature and some theory of outgassing,” R. Weiss.

2.) LIGO Document T940009, “Water Outgassing Data and Model for the LIGO Beam Tubes,” R. Weiss.

3.) LIGO Document T0800330-00, “Reanalysis of average pressure in the beamtube as a function of water injected at the ends,” R. Weiss, 30 October 2008.

    This note describes the effects of water injected at the ends of the beamtube, including the effects of adsorption and reemission dynamics.

4.) LIGO Document T1200375-v2, “Transient response of a beam tube during leak checking,” M. Zucker, R. Weiss, 30 July 2012.

    This note shows the effects (pressure versus length and pressure versus time) on injecting helium into the beamtube at a given location at a given rate for a given time.

5.) LIGO Document T1200399-v1, “Consequences of the LLO beamtube leak and strategies to deal with them,” R. Weiss, 17 August 2012.

    This note describes the effects of a leak in the LLO Y-arm beamtube and a number of strategies to remove water from the beamtube, including additional pumps and/or a bakeout of the beamtube.

6.) LIGO Document T1200518-v1, “Report on the Leak Localization on the Y arm at LLO,” M. Meyer et al., 30 November 2012.

    This note presents the analysis methods and results of numerical models used to localize the position of a leak or leaks in the LLO Y-arm beamtube.

7.) LIGO Document T1300553-v1, “Effect of leak-test port separation; measurement of He transient at z = 250m,” M. Zucker, 17 June 2013.

8.) LIGO Document T1300560-v1, “Helium leak detection response of the LIGO beamtube using a finite element diffusion model,” R. Weiss, 19 June 2013.