*LIGO Laboratory / LIGO Scientific Collaboration*

LIGO-E1100995-v6 *LIGO* February 08, 2012

# SUS Testing Procedure

S. BISCANS, J. GARCIA, J.KISSEL

Distribution of this document:
Advanced LIGO Project

This is an internal working note
of the LIGO Laboratory

**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 – NW22-295**
**185 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

# Table of Contents

## Introduction

Each suspension system needs to go through a certain amount of testing steps before validation.

These steps will attest to the good behavior of the system.

Each step is described in this procedure.

**List of the different steps**:

- Step 1:  Set up offsets and gains of the OSEMs
- Step2: Centering OSEMs
- Step 3: Local static test
- Step 4: Euler static test
- Step 5: Linearity test
- Step 6: Diagonalisation
- Step 7: Pitch & Longitudinal transfer functions
- Step 8: Full set of transfer functions
- Step 9: plot the transfer functions

The results generated by the testing will be report in a excel data sheet (DCC #XXXX).

All the scripts used for this testing are stored in the SVN at:

/ligo/svncommon/SusSVN/sus/trunk/ Common/Generic_Testing_Scripts/

Some of the scripts are calling channels lists stored in the SVN at:

/ligo/svncommon/SusSVN/sus/trunk/Common/Channels_Lists/

Be sure these two folders are up to date before proceeding.

# I - Step 1 : Set up offsets and gains of the OSEMs

**Tools required → MEDM Screens, MATLAB function**

**Procedure to follow for this test:**

For this test, the OSEMs need to be pushed all the way back (full light).

- Open with Matlab the following script:
  /ligo/svncommon/SusSVN/sus/trunk/Common/Generic_Testing_Scripts/Offset_Gain_OSEM.m
- In the command window, run the function generated by this script. You need to provide three parameters in order to run this function: IFO, Optics and Channels_Lists.
  *Offset_Gain_OSEM('IFO','Optics',Channels_List)*

If you don't know the IFO or the Optics, look at one of the channel name. The channel names are built this way:

IFO:SUS-Optics_etc...

Example:
H2:SUS-ITMY_M0_OSEMINF_RT_OUT

In this example, the IFO is H2 and the Optics is ITMY.

The Channels_List parameter required a number, according to the type of suspension:

**HSTS → #1**
**HLTS → #2**
**QUAD → #3**
**FM → #4**

- Wait for the end of the process, until the result pulls out in the command window.
- Copy/paste the results into table 1 of the document #XXXX
- Fill up the MEDM with the new values (1st column: Offsets, 2nd column: gains)

## II - Step 2 : Centering OSEMs

**Tools required → Matlab function**

**Procedure to follow for this test:**

This test checks out the good centering of the OSEMs

- Open with Matlab the following script: /ligo/svncommon/SusSVN/sus/trunk/Common/Generic_Testing_Scripts/Mean_STD_OSEM.m
- In the command window, run the function generated by this script. You need to provide three parameters in order to run this function: IFO, Optics and Channels_Lists. *Mean_STD_OSEM('IFO','Optics',Channels_List)*

If you don't know the IFO or the Optics, look at one of the channel name. The channel names are built this way:

IFO:SUS-Optics_etc...

Example:
H2:SUS-ITMY_M0_OSEMINF_RT_OUT

In this example, the IFO is H2 and the Optics is ITMY.

The Channels_List parameter required a number, according to the type of suspension:

**HSTS → #1**
**HLTS → #2**
**QUAD → #3**
**FM → #4**

- Wait for the end of the process, until the result pulls out in the command window.
- Copy/paste the results into table 2 of the document #XXXX. The numbers should be between -10000 counts and +10000 counts.

## III - Step 3 : Local static test

**Tools required → Matlab function**

**Procedure to follow for this test:**

This test checks out the good sign of the OSEMs magnets

- Open with Matlab the following script: ligo/svncommon/SusSVN/sus/trunk/Common/Generic_Testing_Scripts/Static_Test_Local_Basis.m
- In the command window, run the function generated by this script. You need to provide three parameters in order to run this function: IFO, Optics and Channels_Lists. *Static_Test_Local_Basis('IFO','Optics',Channels_List)*

If you don't know the IFO or the Optics, look at one of the channel name. The channel names are built this way:

IFO:SUS-Optics_etc...

Example:
H2:SUS-ITMY_M0_OSEMINF_RT_OUT

In this example, the IFO is H2 and the Optics is ITMY.

The Channels_List parameter required a number, according to the type of suspension:

**HSTS → #1**
**HLTS → #2**
**QUAD → #3**
**FM → #4**

- Wait for the end of the process, until the result pulls out in the command window.
- Copy/paste the results into table 3 of the document #XXXX. The numbers on the diagonal should all be positives.

## IV - Step 4 : Euler static test

**Tools required → Matlab function**

**Procedure to follow for this test:**

- Open with Matlab the following script: ligo/svncommon/SusSVN/sus/trunk/Common/Generic_Testing_Scripts/Static_Test_Euler_Basis.m
- In the command window, run the function generated by this script. You need to provide three parameters in order to run this function: IFO, Optics and Channels_Lists.
  *Static_Test_Euler_Basis('IFO','Optics',Channels_List)*

If you don't know the IFO or the Optics, look at one of the channel name. The channel names are built this way:

IFO:SUS-Optics_etc...

Example:
H2:SUS-ITMY_M0_OSEMINF_RT_OUT

In this example, the IFO is H2 and the Optics is ITMY.

The Channels_List parameter required a number, according to the type of suspension:

**HSTS → #1**
**HLTS → #2**
**QUAD → #3**
**FM → #4**

- Wait for the end of the process, until the result pulls out in the command window.
- Copy/paste the results into table 4 of the document #XXXX. The numbers on the diagonal should all be positives.

## V - Step 5 : Linearity test

**Tools required → Matlab function**

**Procedure to follow for this test:**

This test checks out the linearity of the OSEMs

- Open with Matlab the following script: ligo/svncommon/SusSVN/sus/trunk/Common/Generic_Testing_Scripts/Linearity_Test_Awgstream.m
- In the command window, run the function generated by this script. You need to provide three parameters in order to run this function: IFO, Optics and Channels_Lists.
*Linearity_Test_Awgstream('IFO','Optics',Channels_List)*

If you don't know the IFO or the Optics, look at one of the channel name. The channel names are built this way:

IFO:SUS-Optics_etc...

Example:
H2:SUS-ITMY_M0_OSEMINF_RT_OUT

In this example, the IFO is H2 and the Optics is ITMY.

The Channels_List parameter required a number, according to the type of suspension:

**HSTS → #1**
**HLTS → #2**
**QUAD → #3**
**FM → #4**

- Wait for the end of the process, until the plot pulls up.
- Save the plot in the local folder corresponding to the system.

## VI - Step 6 : Diagonalization

**Tools required → Matlab function**

**Procedure to follow for this test:**

This test checks out the cross couplings between the OSEMs

- Open with Matlab the following script: ligo/svncommon/SusSVN/sus/trunk/Common/Generic_Testing_Scripts/Diagonalization.m
- In the command window, run the function generated by this script. You need to provide three parameters in order to run this function: IFO, Optics and Channels_Lists. *Diagonalization('IFO','Optics',Channels_List)*

If you don't know the IFO or the Optics, look at one of the channel name. The channel names are built this way:

IFO:SUS-Optics_etc...

Example:
H2:SUS-ITMY_M0_OSEMINF_RT_OUT

In this example, the IFO is H2 and the Optics is ITMY.

The Channels_List parameter required a number, according to the type of suspension:

**HSTS → #1**
**HLTS → #2**
**QUAD → #3**
**FM → #4**

- At he beginning of the script, there is a section called “Tunable parameters”.  Tuned those parameters before launching he function.
- Wait for the end of the process, until the plot pulls up.
- Save the plot in the local folder corresponding to the system.

## VII - Step 7 : Pitch & Longitudinal transfer functions

**Tools required → DTT , Matlab**

**Procedure to follow for this test:**

In progress …

# VIII - Step 8 : Full set of transfer functions

**Tools required → Matlab functions**

**Procedure to follow for this test:**

- Open **2 Matlab**
- In the first one, pull out the following script: ligo/svncommon/SusSVN/sus/trunk/Common/Generic_Testing_Scripts/Run_Exc_Batch.m
- At the beginning of the script, there is a section called “Input argument to edit”. Tuned those parameters before launching the program.
- Launch the script. This script will start the excitation.
- The measurement time will be displayed in the command window.
- In the meantime, open the following script in the other matlab: ligo/svncommon/SusSVN/sus/trunk/Common/Generic_Testing_Scripts/Run_Get_Batch.m
- In the command window, run the function generated by this script. You need to provide four parameters in order to run this function: IFO, Optics, Suboptics, ligo_dir.
  *Run_Get_Batch('IFO','Optics','Suboptics','ligo_dir')*

If you don't know the IFO or the Optics, look at one of the channel name. The channel names are built this way:

IFO:SUS-Optics_etc...

Example:
H2:SUS-ITMY_M0_OSEMINF_RT_OUT

In this example, the IFO is H2 and the Optics is ITMY.

The Suboptics parameter corresponds at the specific suspension that you are working on (ex: 'PR3').

The 'ligo_dir' will be most of the time 'ligo'. But, if you are working on the test stand, it might be different. For example at LHO, the parameter will be 'ligo3'.

- Wait the start of the excitation (generated by Run_Exc_Batch.m) before running the Run_Get_Batch function.
- When the excitation is done and the command window of the second matlab displays “Watching batch_file.mat for more parameter files...”, you can kill the process (Ctrl+C) in the command window).
- The result .mat file is now saved in the SVN at:
  /

ligo/svncommon/SusSVN/sus/trunk/'Optics'/'IFO'/'Suboptics'/Common/Data/Transfr_Functions/Measurements/Undamped/

– The name of the result file will have the following name: 'IFO'_'Suboptics'_Data_TF_'StartFrequency'Hz_'EndFrequency'Hz_YYYYMMDD_HHMMSS.mat
(YYYYMMDD_HHMMSS corresponds at the time when the measurement was launched).

# IX - Step 9 : Plot the transfer functions

**Tools required → Matlab function**

**Procedure to follow for this test:**

- Open with Matlab the following script:
  ligo/svncommon/SusSVN/sus/trunk/Common/Generic_Testing_Scripts/plot_TF_batch_file.m
- At the beginning of the script, there is a section called "General Parameters". Tuned those parameters.
- Under the section "Initial Data – Create a new case  with your data !", create a new case and put the name of the .mat file you want to plot, with a title and a date.

Example:

*Data_series=X;*
*switch Data_series*
*case X*
*data = {load ''IFO'_'Suboptics'_Data_TF_L2L_'StartFrequency'Hz_'EndFrequency'Hz_YYYYMMDD_HHMMSS.mat'};*
*title_str=({'Put a title here'});*
*date_str='YYYY_MM_DD';*
*end*

- Run the script
- Save the plot in the local folder corresponding to the system.