

# Transformation Matrices Between SEI/SUS Coordinates

Jeff Kissel, Brian Lantz, Chris Kucharczyk  
T1100617-v4, January 24, 2013

## Contents

<b>1</b>	<b>Summary</b>	<b>1</b>
<b>2</b>	<b>Introduction</b>	<b>1</b>
<b>3</b>	<b>Definitions</b>	<b>2</b>
<b>4</b>	<b>Generation and Use of the Projections</b>	<b>4</b>
<b>5</b>	<b>CART2EUL: PRM Example</b>	<b>7</b>
<b>6</b>	<b>EUL2CART: ETMY Example</b>	<b>9</b>
<b>7</b>	<b>The Transform Calculation</b>	<b>10</b>
7.1	Re-order SEI Signals . . . . .	11
7.2	Translate Origin . . . . .	11
7.3	Rotate About Z' . . . . .	13
7.4	The Full Transform . . . . .	14
<b>8</b>	<b>Practical Implementation</b>	<b>14</b>

## 1 Summary

We describe the coordinate transforms used to drive and monitor the ISI optical table in the Euler basis of the suspension on that table.

## 2 Introduction

We have created a set of basis transform matrices to allow one to easily transform between the motion of the ISI platform in the ISI's cartesian basis and the motion about the suspension point of the various optics in the Euler basis about the suspension point for the pendulum chain. In Advanced LIGO, the in-vacuum seismic isolation (ISI) platforms provide a new opportunity to characterize and control motion exciting the suspensions that are mounted to them. However, the suspension points of the pendulum chains are not colocated with the center of the ISI coordinate system, and in most cases the orientation

of the optics is not aligned with the global directions. Thus, while it is possible to monitor and drive the suspensions from the ISI tables, it is not always obvious how to do this in the Euler basis of the suspension. For example, pitch about the suspension point of MC1 involves significant contributions of the ISI signals in X, Y, Z, RX and RY. Further, there are often multiple suspensions per platform, where their local coordinate systems are misaligned from each other, see Figure 1 for example. Therefore we have created and implemented projection matrices in the control diagrams to make the transformations simple. At this writing, the transforms all occur in the ISI models, but we expect this will soon begin to happen in the SUS models, and perhaps in other places as well.

This document describes how to use the projection matrices, and also serves to supplement the function `projectSEI2SUS.m`, which generates a pair of  $6 \times 6$  transformation matrices  $\mathbf{M}(\vec{r}, \theta)$  which convert between the ISI's local Cartesian coordinate system and Euler coordinates about the suspension point of a given suspension chain.

$$\vec{S} = (\text{X}; \text{Y}; \text{RZ}; \text{Z}; \text{RX}; \text{RY}) \quad (1)$$

into the local SUS Euler basis

$$\vec{E} = (\text{L}; \text{T}; \text{V}; \text{R}; \text{P}; \text{Y}) \quad (2)$$

or Longitudinal, Transverse, Vertical, Roll, Pitch, and Yaw, such that

$$\vec{E} = \mathbf{M}(\vec{r}, \theta) \vec{S}. \quad (3)$$

where  $\vec{r}$  is a vector connecting the origin of the two systems, and  $\theta$  is the angle between the ISI's  $X$  axis, and the SUS  $L$  axis. Figure 2 shows the transformation between such coordinates in graphical form. Note that in this transformation, those displaced from each other, we assume that the SEI  $Z$  and SUS  $V$  axes remain aligned.

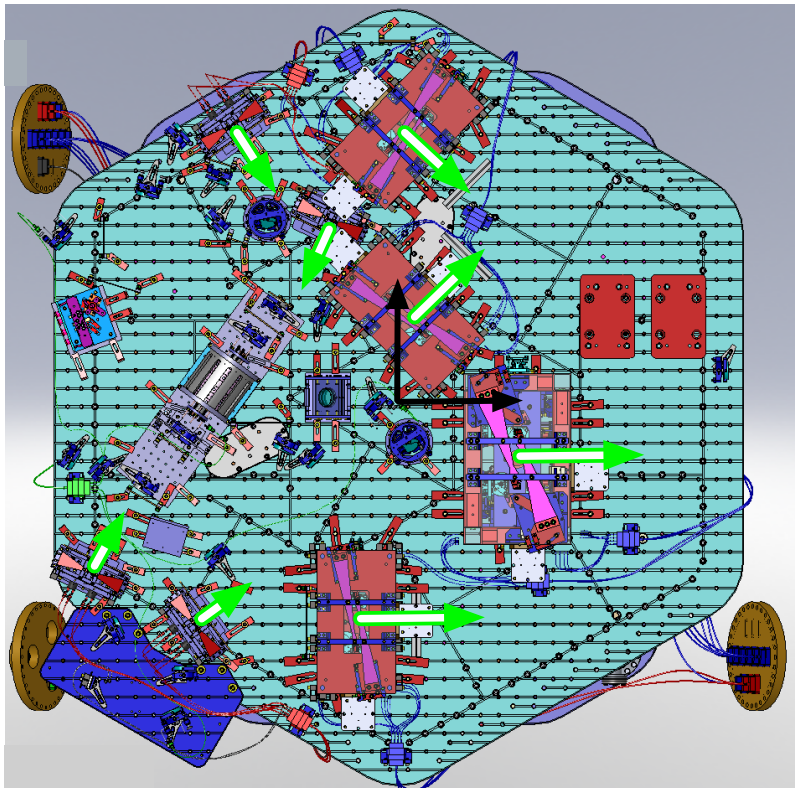


Figure 1: Top-down view of L1-HAM2, from D0900365. Each suspension's longitudinal axis is shown in green, and the HAM-ISI's  $X$  and  $Y$  axes are shown in black.

### 3 Definitions

The transforms are defined to be between the center of the Cartesian coordinate system for the ISI stage rigidly connected to the optical table (i.e. stage 1 for the HAM-ISI and stage 2 for the BSC-ISI) and the suspension point, treated as a single point centered vertically

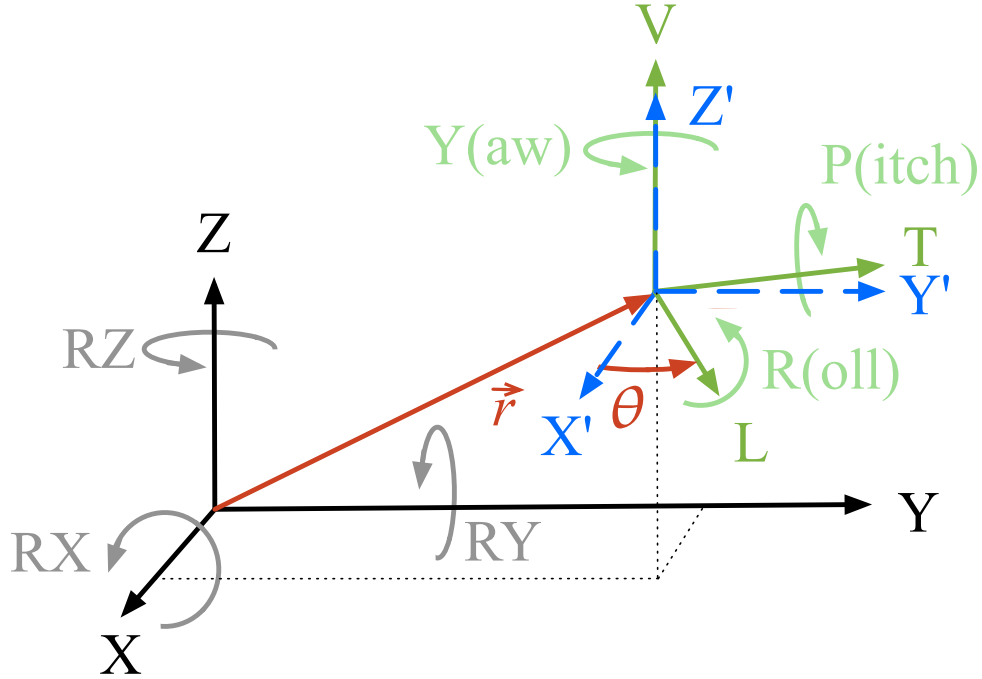


Figure 2: Transformation breakdown between an ISI's local Cartesian coordinate system (in black) and a given suspension's local coordinate system (in green), whose origins are separated by a vector  $\vec{r}$ , and  $X$  and  $L$  axes separated by an angle  $\theta$ .

with the plane formed by the physical wire break off points at the under-surface of top-most blade-wire clamps, and centered horizontally on a line parallel with the center of mass of the suspended optic of interest) of the pendulum. The suspension point location is projected from and given using the ISI coordinate system as the origin ( $x = y = z = 0$ ). We assume the optic face is perpendicular to the horizontal plane of the optics table, and therefore parallel with local gravity – the  $x, y$ -plane is parallel to the  $L, T$ -plane. If not, then you'll have to be smart enough to make your own transformation matrix code (see LIGO-P050030). The orientation,  $\theta$ , of the optic is defined by the angle of the Longitudinal direction for the optic, relative to the  $+x$  direction of the optical table, in the  $+rz$  direction of the ISI table (i.e. CCW as seen from above). For example, the HR coating of the ITMY optic is pointed in the  $+y$  direction, so we can say its orientation is  $+90$  degrees. The HR coating of the ETMX optic is pointed back along  $-x$ , so its orientation is  $+180$  degrees.

The distance between the vertical planes of the ISI coordinate system and Suspension point is derived from knowing the vertical distance between the ISI origin and the optical table,  $\Delta z_{ISI}$ , summed with the vertical distance between the optical table and the sus-

pension point,  $\Delta z_{SUS}$ , and any vertical spacers,  $\Delta z_{spacer}$ . In the ISI coordinate systems, the  $z=0$  plane is defined by the plane created by the inner-most stage horizontal actuators (i.e. stage 2 for a BSC-ISI and stage 1 for a HAM-ISI) – which should be pretty well aligned with that stage’s lower zero moment plane. The optical table for the BSC-ISI is 182 mm below the stage 1-2 actuator plane, so  $\Delta z = -0.182$  m (E1200419). The optical table for the HAM-ISI is 218 mm above the stage 0-1 actuator plane, so  $\Delta z = +0.218$  m(LIGO-T1000388). The vertical location of suspension point plane with respect for all suspension types are shown in Table 1. Spacer heights for HAM suspensions can be found in LIGO-T1000228, there are no spacers for BSC suspensions.

Table 2 shows the components of the final displacement vector  $\vec{r} = [x, y, z]$ , and orientation (yaw),  $\theta$ , between the ISI and Suspension Point coordinate systems in addition to a few other useful bits of information.

Table 1: Vertical location of the suspension point plane for each aLIGO suspension type. Distances were “measured” using the SolidWorks eDrawings cited, measuring from the optical table mounting surface to one the bottoms of the top-most blade wire clamps (assuming all suspension points are at the same height).

Suspension Type	$\Delta z_{SUS}$ [mm]	Reference
HAUX	394.865	LIGO-D1000120-v10
HTTS	248.451	LIGO-D1001396-v3
OMCS	646.990	LIGO-D0900295-v1
TMTS	????	LIGO-D0901880-v3
HSTS	826.596	LIGO-D020700-v1
HLTS	806.264	LIGO-D070447-v2
BSFM	-40.3	LIGO-D1000392-v2
QUAD	-100.3	LIGO-D0901346-v9

## 4 Generation and Use of the Projections

The projections are all calculated by the Matlab wrapper script  
`/${SUS_SVN}/Common/MatlabTools/make_ISI2SUS_projections.m`

The locations of the optics are stored in the function

`/${SUS_SVN}/Common/MatlabTools/SEI2SUScoordinates.m`

All the matrices, and some other info are all saved to a single file, which is saved in the ISC portion of the userapps SVN. It is saved as a single file because that was simpler. It is in ISC because projecting from one subsystem to another is a systems sort of issue. The data file is:

`/${userapps}/trunk/isc/common/projections/ISI2SUS_projection_file.mat`

Table 2: Location of the suspension point origin relative to the Cartesian coordinate system for the ISI stage to which it is rigidly connected.

IFO	Optic	Chamber	$x, y, z$ location of suspension point [m]	orientation ([deg], rel. to +X)	ref doc
H2	ETMY	BSC6	0.2, -0.3698, -0.2823	270 deg	LIGO-D0900515
H2	ITMY	BSC8	0.2, 0.426, -0.2823	90 deg	LIGO-D0900363
H2	FMY	BSC8	0.179, -0.386, -0.2223	45.4 deg	LIGO-D0900364
H1	MC1	HAM2	0.0227, 0.2288, 1.06	44.3 deg	LIGO-D0901088
H1	MC2	HAM3	0.035, 0.4874, 1.071	180 deg	LIGO-D0901099
H1	MC3	HAM2	0.0227, 0.7462, 1.06	-44.3 deg	LIGO-D0901089
H1	PRM	HAM2	-0.1098, -0.6282, 1.064	0.34 deg	LIGO-D0901090
H1	PR2	HAM3	0.2868, -0.5304, 1.073	180 deg	LIGO-D0901098
H1	PR3	HAM2	0.3308, -0.1768, 1.024	0.64 deg	LIGO-D0901086
H1	SRM	HAM5	0.3075, 0.1761, 1.116	93.3 deg	LIGO-D0901133
H1	SR2	HAM4	-0.5959, -0.3107, 1.126	-87.6 deg	LIGO-D0901128
H1	SR3	HAM5	-0.1755, 0.4554, 1.096	90.8 deg	LIGO-D0901132
H1	ITMY	BSC1	0.2, 0.2926, -0.2823	90 deg	LIGO-D0901140
H1	BS	BSC2	-0.1814, -0.2058, -0.2223	135 deg	LIGO-D0901145
H1	ITMX	BSC3	0.3305, -0.2, -0.2223	0 deg	LIGO-D0901149
H1	IM1	HAM2	-0.4669, 0.7631, 0.629	-54.6 deg	LIGO-D1200623
H1	IM2	HAM2	-0.8638, -0.4891, 0.629	65 deg	LIGO-D1200625
H1	IM3	HAM2	-0.193, 0.4989, 0.629	244 deg	LIGO-D1200626
H1	IM4	HAM2	-0.5696, -0.6383, 0.629	36.1 deg	LIGO-D1200624
L1	MC1	HAM2	0.0235, 0.2279, 1.05	44.3 deg	LIGO-D0900413
L1	MC2	HAM3	0.035, 0.4874, 1.055	180 deg	LIGO-D0900526
L1	MC3	HAM2	0.02268, 0.7462, 1.05	-44.3 deg	LIGO-D0900414
L1	PRM	HAM2	-0.1055, -0.6282, 1.054	0.3 deg	LIGO-D0900415
L1	PR2	HAM3	0.2981, -0.5304, 1.063	180 deg	LIGO-D0900524
L1	PR3	HAM2	0.3318, -0.1722, 1.024	0.65 deg	LIGO-D0900368
L1	SRM	HAM5	0.3075, 0.1761, 1.117	93.4 deg	LIGO-D0900463
L1	SR2	HAM4	-0.5959, -0.3107, 1.145	-87.6 deg	LIGO-D0900424
L1	SR3	HAM5	-0.1743, 0.4554, 1.115	90.8 deg	LIGO-D0900461
L1	ITMX	BSC3	0.3305, -0.2, -0.2223	0 deg	LIGO-D0900495
L1	BS	BSC2	-0.1814, -0.2058, -0.2223	135 deg	LIGO-D0900431
L1	ITMY	BSC1	0.2, 0.3006, -0.2823	90 deg	LIGO-D0900445
L1	IM1	HAM2	-0.4669, 0.7631, 0.6191	-54.6 deg	LIGO-D1101537
L1	IM2	HAM2	-0.8638, -0.4891, 0.6191	65 deg	LIGO-D1101539
L1	IM3	HAM2	-0.1933, 0.499, 0.6191	245 deg	LIGO-D1101540
L1	IM4	HAM2	-0.5695, -0.6383, 0.6191	36.1 deg	LIGO-D1101538

In the file are two data structures:  
`read_me` - Which is filled with hopefully useful information, and  
`ISI2SUSprojections` - filled with projection matrices.

So far, we have defined the following optics:

h2: itmy, fmy, etmy

l1: mc1, mc2, mc3, prm, pr2, pr3, srm, sr2, sr3, itmy, bs, itmx, im1 (sm1), im2 (pmmt1), im3 (pmmt2), im4 (sm2)

l1: mc1, mc2, mc3, prm, pr2, pr3, srm, sr2, sr3, itmy, bs, itmx, im1 (sm1), im2 (pmmt1), im3 (pmmt2), im4 (sm2)

(i.e. every H1 and L1 optic but the ETMs, OMC, Tip-Tilts and the TMSs).

The data structure is pretty simple:

For each optic, there is an entry in a large data structure. The entry includes reference info on the optic, including: the location of the optic w.r.t. the center of the optical table, the location of the optic w.r.t. the center of the ISI cartesian basis, and the yaw of the optic, CCW from the +x direction, in degrees, as seen from above. The optic's chamber, systems drawing number and url are also included. Each optic has the 6x6 matrix `EUL2CART`, for the projection from Euler to ISI cartesian basis, which is used for the drive for the ISI. The final entry is the 6x6 matrix `CART2EUL` (the inverse of `EUL2CART`) which is used to populate the monitor path matrix `CART2EUL`, which projects motion in the ISI cartesian basis (as measured by the GS13s) to motion in the Euler basis about the suspension point for monitoring input motion to the suspension.

The data structure is named `ISI2SUSprojections`, with fields for each IFO and subfields for each optic. Each optic has the 8 sub-sub fields listed above, i.e. `ISI2SUSprojections.ifo.optic.matrix.name`, etc.

For example, the data for the H1 MC3 is, in matlab,

```
>> ISI2SUSprojections.h1.mc3
```

```
ans =  
  location_from_ISI_origin: [0.0227 0.7462 1.0604]  
  location_from_optics_table: [0.0227 0.7462 0.84242]  
      yaw: -44.3  
  chamber: 'HAM2'  
  reference_dwg: 'D0901089'  
  reference_url: 'https://dcc.ligo.org/cgi-bin/private/...  
                DocDB/ShowDocument?docid=2670'  
  CART2EUL: [6x6 double]  
  EUL2CART: [6x6 double]
```

## 5 CART2EUL: PRM Example

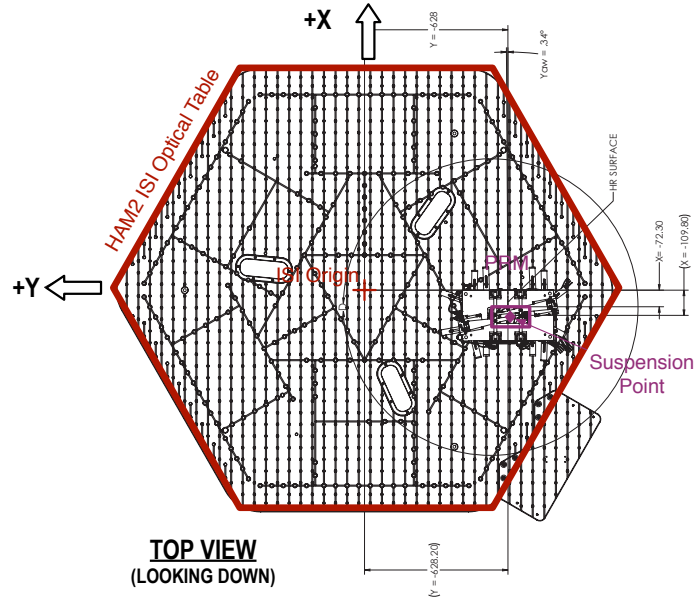


Figure 3: Top view of HAM2-ISI w/ PRM, from D0901090. The red hexagon outlines the HAM-ISI stage 1 optical table, and the red + is the center of the ISI Cartesian coordinate system. The purple diamond is the suspension point for the PRM optic (shown as a blue rectangle). The purple suspension point is at  $(x, y, z) = (-0.1098, -0.6282, 1.064)$  [m] from the ISI center. The HR surface of the optic points in the  $(+x, +y)$  direction, with an orientation of  $+0.34^\circ$ .

The matrix to convert the HAM-ISI Cartesian coordinate system to the PRM suspension point's Euler Basis is

**ISI2SUSprojections.h1.prm.CART2EUL.**

The value of this matrix is



	X	Y	RZ	Z	RX	RY
Long	1	0.0059	0.6275	0	-0.0063	1.0636
Trans	-0.0059	1	-0.1135	0	-1.0636	-0.0063
Vert	0	0	0	1	-0.6282	0.1098
Roll	0	0	0	0	1	0.0059
Pitch	0	0	0	0	-0.0059	1
Yaw	0	0	1	0	0	0

Table 3: Transformation matrix from the Cartesian Basis of the HAM-ISI to the Euler Basis of the PRM Suspension Point. A Longitudinal drive for the Suspension Point is a comprised of some component of X, Y, RZ, RX, and RY.

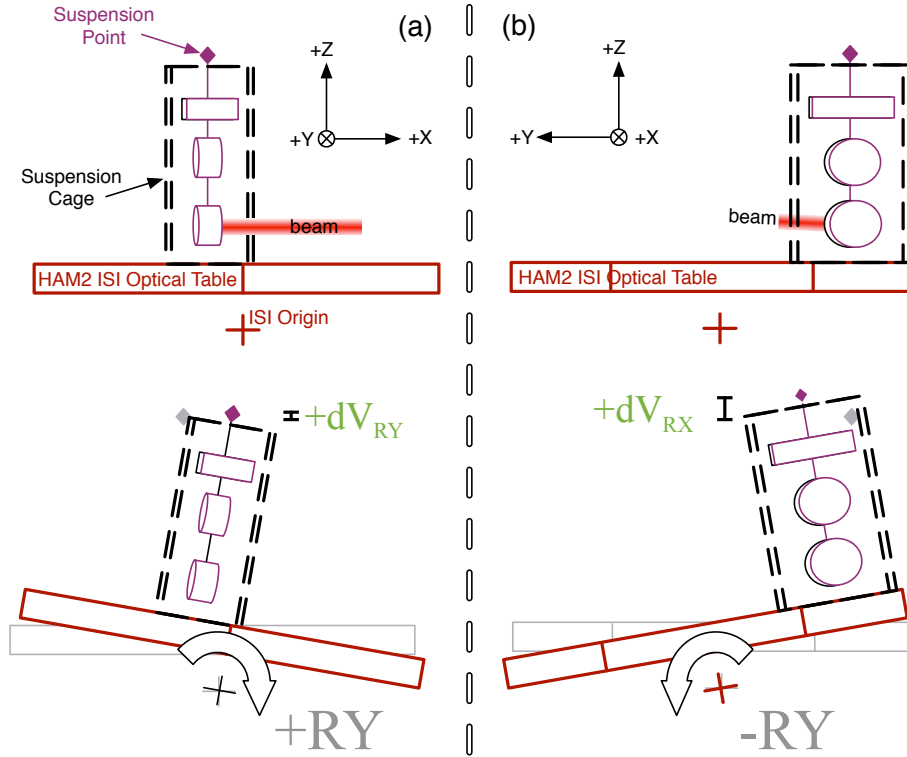


Figure 4: Two side-view drawings of HAM2-ISI with PRM, showing the (among other things) resulting positive vertical displacement,  $+dV$  of the suspension point from looking (a) along the  $+Y$  direction, with exaggerated  $+RX$  motion about the ISI coordinate center, and (b) along the  $+X$  direction, with exaggerated  $-RY$  motion about the ISI coordinate center. The red rectangle is the outline of the HAM ISI stage 1 optical table, and the red  $+$  is the center of the ISI cartesian coordinate system. The purple diamond is the suspension point for the PRM suspension chain. The purple suspension point is at  $(x, y, z) = (-0.1098, -0.6282, 1.064)$  [m] from the ISI center (though the cartoon is only approximately to scale, and the optic orientation, and  $+RX$  and  $-RY$  motions are exaggerated for clarity).

## 6 EUL2CART: ETMY Example

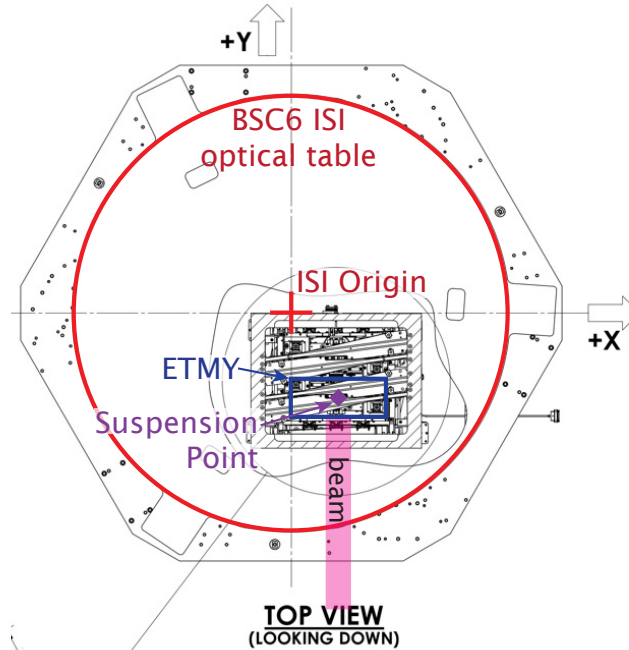


Figure 5: Top view of BSC6-ISI w/ ETMY, from D0900515. The red circle outlines the ISI stage 2 optical table, and the red + is the center of the ISI Cartesian coordinate system. The purple diamond is the suspension point for the ETMY optic (shown as a blue rectangle). The purple suspension point is at  $(x, y) = (+0.200, -0.3698)$  [m]. The HR surface of the optic points in the  $-Y$  direction, so the orientation is  $+270^\circ$ .

The matrix to convert from Euler basis coordinates to ISI coordinates is **ISI2SUSprojections.h2.etmy.EUL2CART**.

The value of this matrix is:

	Long	Trans	Vert	Roll	Pitch	Yaw
X	0	1.0000	0	-0.2860	0	-0.3698
Y	-1.0000	0	0	0	-0.2860	-0.2000
RZ	0	0	0	0	0	1.0000
Z	0	0	1.0000	-0.2000	0.3698	0
RX	0	0	0	0	1.0000	0
RY	0	0	0	-1.0000	0	0

Table 4: Transformation matrix from the Euler Basis of the ETMY suspension to the Cartesian basis of the ISI. A Longitudinal drive for the Suspension is a drive in the -Y direction of the ISI. To Pitch the optic about the suspension point, the ISI table will translate in the -Y direction, translate in the +Z direction, and rotate in the +RX direction, as shown in figure 6.

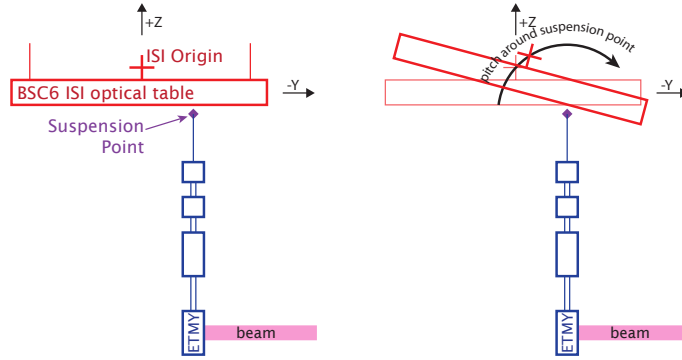


Figure 6: Side drawing of BSC6-ISI with ETMY, looking along the +X direction, showing (hopefully) exaggerated pitch motion about the ETMY suspension point. The red rectangle is the outline of the ISI stage 2 optical table, and the red + is the center of the ISI cartesian coordinate system. The purple diamond is the suspension point for the ETMY optical chain. The purple suspension point is at  $Y = -0.3698$  m,  $Z = -0.286$  m from the ISI center, or -0.104 meter from the optical table surface. If the optical table has positive pitch about the ETMY suspension point, the center of the ISI table will rotate in the +RX direction, and translate in +Z and -Y.

## 7 The Transform Calculation

We desire the matrix transformation that takes us from signals in the local SEI Cartesian basis (Eq. 1), into the local SUS Euler basis (Eq. 2), as in Eq. 3. This is done in three

steps, for clarity, by splitting the transformation into separate matrices, i.e.,

$$\mathbf{M} = \mathbf{R} \mathbf{T} \mathbf{C} \quad (4)$$

where

- $\mathbf{C}$  re-orders the SEI local coordinates from those motivated by its local sensors to a more traditional Cartesian order.
- $\mathbf{T}(\vec{r})$  translates the origin of the SEI basis to the SUS basis, creating an intermediate “primed” Cartesian coordinate basis.
- $\mathbf{R}(\theta)$  rotates the axes about the  $Z'$  axis.

## 7.1 Re-order SEI Signals

The SEI group has defined the order in which they propagate Cartesian signals through their subsystem in such a way that their internal basis transform matrices look roughly block-diagonal. As a result, the six cartesian degrees of freedom are in a “non-traditional” order, as seen in Eq. 1. Since the SUS group’s signal vector follow a more traditional ordering of translation-then-rotation degrees of freedom, we begin with a re-ordering matrix,

$$\vec{C} = \mathbf{C} \vec{S} \quad (5)$$

$$\begin{pmatrix} X \\ Y \\ Z \\ RX \\ RY \\ RZ \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} X \\ Y \\ RZ \\ Z \\ RX \\ RY \end{pmatrix} \quad (6)$$

where  $\vec{C}$  is a “traditionally ordered” cartesian signal vector.

## 7.2 Translate Origin

Next we translate the origin of the Cartesian basis along a vector  $\vec{r}$ , such that

$$\begin{pmatrix} X' \\ Y' \\ Z' \\ RX' \\ RY' \\ RZ' \end{pmatrix} = \mathbf{T}(\vec{r}) \begin{pmatrix} X \\ Y \\ Z \\ RX \\ RY \\ RZ \end{pmatrix} \quad (7)$$

Because the primed coordinates are not rotated with respect to the unprimed, we expect translation and rotation degrees of freedom do be the same. However, small rotations about the unprimed origin ( $\vec{C}(4 : 6) = \vec{R} = [RX; RY; RZ]$ ) cause some translation in the primed origin ( $\vec{C}'(1 : 3) = \vec{X}' = [X'; Y'; Z']$ ) because of the lever arm,  $\vec{r}$ , between them. To account for this, we need the cross product between the unprimed rotations and the lever arm such that the primed translational degrees of freedom,

$$\vec{X}' = \vec{X} + \vec{R} \times \vec{r} \quad (8)$$

Given that the vector  $\vec{r}$  is fixed, we can formulate the matrix notation for the cross product with a generic signal input vector  $\vec{R}$ ,

$$\begin{aligned} \left(\vec{R} \times \vec{r}\right)_i &= -\left(\vec{r} \times \vec{R}\right)_i \\ &= -\sum_{j=1}^3 \sum_{k=1}^3 \epsilon_{ijk} r_j R_k \\ &\quad \left(\sum_{j=1}^3 \epsilon_{ijk} a_j = c_{ik}^a = \begin{pmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{pmatrix}\right) \\ \left(\vec{R} \times \vec{r}\right)_i &= -\sum_{k=1}^3 c_{ik}^r R_k \\ \left(\vec{R} \times \vec{r}\right) &= -\mathbf{c}^r \vec{R} \end{aligned} \quad (9)$$

Otherwise there is no “cross coupling” between the degrees of freedom, so the upper left and lower right quadrants are identity, and the lower left quadrant is zeros. The translation matrix therefore takes the form

$$\begin{pmatrix} X' \\ Y' \\ Z' \\ RX' \\ RY' \\ RZ' \end{pmatrix} = \left( \begin{array}{c|c} \mathbf{I} & -\mathbf{c}^r \\ \hline \mathbf{0} & \mathbf{I} \end{array} \right) \begin{pmatrix} X \\ Y \\ Z \\ RX \\ RY \\ RZ \end{pmatrix}$$

or more explicitly,

$$\vec{C}' = \mathbf{T}(\vec{r}) \vec{C} \quad (10)$$

$$\begin{pmatrix} X' \\ Y' \\ Z' \\ RX' \\ RY' \\ RZ' \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & +r_z & -r_y \\ 0 & 1 & 0 & -r_z & 0 & +r_x \\ 0 & 0 & 1 & +r_y & -r_x & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ RX \\ RY \\ RZ \end{pmatrix} \quad (11)$$

where,  $r_x, r_y$  and  $r_z$  are the three Cartesian components of  $\vec{r}$  in the unprimed basis.

### 7.3 Rotate About $Z'$

Here, things are greatly simplified given that we only desire to rotate about the  $Z'$  (or  $V$ ) axis. The rotation will therefore effect only the  $X, Y$  and  $RX, RY$  degrees of freedom. The 2D rotation matrix that rotates the *coordinate system* (not the vector itself), is

$$\mathbf{R}^{2D}(\theta) = \begin{pmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{pmatrix} \quad (12)$$

This rotation applies to both the translation and the rotation axes. Since we know  $Z, RZ$  don't change, the full rotation matrix is simply

$$\vec{E} = \mathbf{R}(\theta) \vec{C}' \quad (13)$$

$$\begin{pmatrix} L \\ T \\ V \\ R \\ P \\ Y \end{pmatrix} = \begin{pmatrix} \cos \theta & \sin \theta & 0 & 0 & 0 & 0 \\ -\sin \theta & \cos \theta & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \cos \theta & \sin \theta & 0 \\ 0 & 0 & 0 & -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X' \\ Y' \\ Z' \\ RX' \\ RY' \\ RZ' \end{pmatrix} \quad (14)$$

## 7.4 The Full Transform

Just for completeness, if we combine Eqs. 6, 11, and 14, the total transformation matrix becomes

$$\begin{aligned}
 \vec{E} &= \mathbf{M}(\vec{r}, \theta) \vec{S} \\
 &= \mathbf{R}(\theta) \mathbf{T}(\vec{r}) \mathbf{C} \vec{S} \\
 \begin{pmatrix} L \\ T \\ V \\ R \\ P \\ Y \end{pmatrix} &= \begin{pmatrix} \cos \theta & \sin \theta & (r_x \sin \theta - r_y \cos \theta) & 0 & -r_z \sin \theta & r_z \cos \theta \\ -\sin \theta & \cos \theta & (r_x \cos \theta + r_y \sin \theta) & 0 & -r_z \cos \theta & -r_z \sin \theta \\ 0 & 0 & 0 & 1 & r_y & -r_x \\ 0 & 0 & 0 & 0 & \cos \theta & \sin \theta \\ 0 & 0 & 0 & 0 & -\sin \theta & \cos \theta \\ 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} X \\ Y \\ RZ \\ Z \\ RX \\ RY \end{pmatrix}
 \end{aligned} \tag{15}$$

## 8 Practical Implementation

For a detailed description of the practical implementation of these matrices, including calibration of the respective channels involved, see LHO aLOG 4553. Though the aLOG refers to H1 SUS PR2, and the matrices used are unique to that suspension, all calibration and AI/AA filters described the log are the same for every suspension (in fact, they are literally copy and pasted).