LIGO-E1300128-v1

aLIGO Interferometer Optics Positions and Orientations for Initial Alignment

Revision History

Version -v1

2/7/2013, D. Coyne: initial release to support update to the L1 PRC and BS alignment solutions

Version -v2

2/12/2013, D. Coyne
1) Fixed error in calculation to insure that the PR2, PR3, SR2, SR3 optics are oriented vertically with respect to the local gravity vector (i.e. in local coordinates). Previous calculation had a small pitch error (due to yaw rotation in the global coordinate frame instead of local frame).
2) completed calculations of the locations and orientations of all of the H1 PRC and SRC optics (was partial for -v1)
3) Made nicely formatted tables for the summary tables

■ Notes and Notation

Both Zemax and Optica (a TM package for optical design in Mathematica) can be used to ray trace through an existing placement of optics, but it is cumbersome to use for determining the aligned positions of a number of wedged optics (as is the case with the recycling cavity). This notebook determines the positions and orientations of the optics given the following information/assumptions:

1) the X- and Y-arms are assumed to exactly 90 degrees apart. (T960176-C indicates that the deviation from 90 degrees is 1.2 microradian).

2) the x and y axes are aligned parrallel to the Fabry-Perot arm cavity axes, with the origin defined at the intersection of the recycling cavity beams and the splitting surface of the BS
(Note that this is later converted to the LIGO global coordinate system; The LIGO coordinate system has its origin at the projected intersection of the BT cavity axes.)

3) the notation is as follows:
ui =ray unit vector for the beam incident upon surface i
vi = ray unit vector for the beam reflected from surface i
wi = ray unit vector for the beam refracted through surface i
ni = unit normal vector for surface i

i	surface
0	BS (AR surface)
1	ITMy (HR surface)
2	ITMy (AR surface)
3	CPy (surface adjacent to ITM)
4	CPy (other surface)
5	BS (50/50 surface)
6	PR3 (HR surface)
7	BS (AR surface, y-transmitted ray)

8 SR3 (HR surface)
9 ITMx (HR surface)
10 ITMx (AR surface)
11 CPx (surface adjacent to ITM)
12 CPx (other surface)
13 BS (AR surface, x-transmitted ray)
14 PR2 (HR surface)
15 PRM (HR surface)
16 SR2 (HR surface)
17 SRM (HR surface)

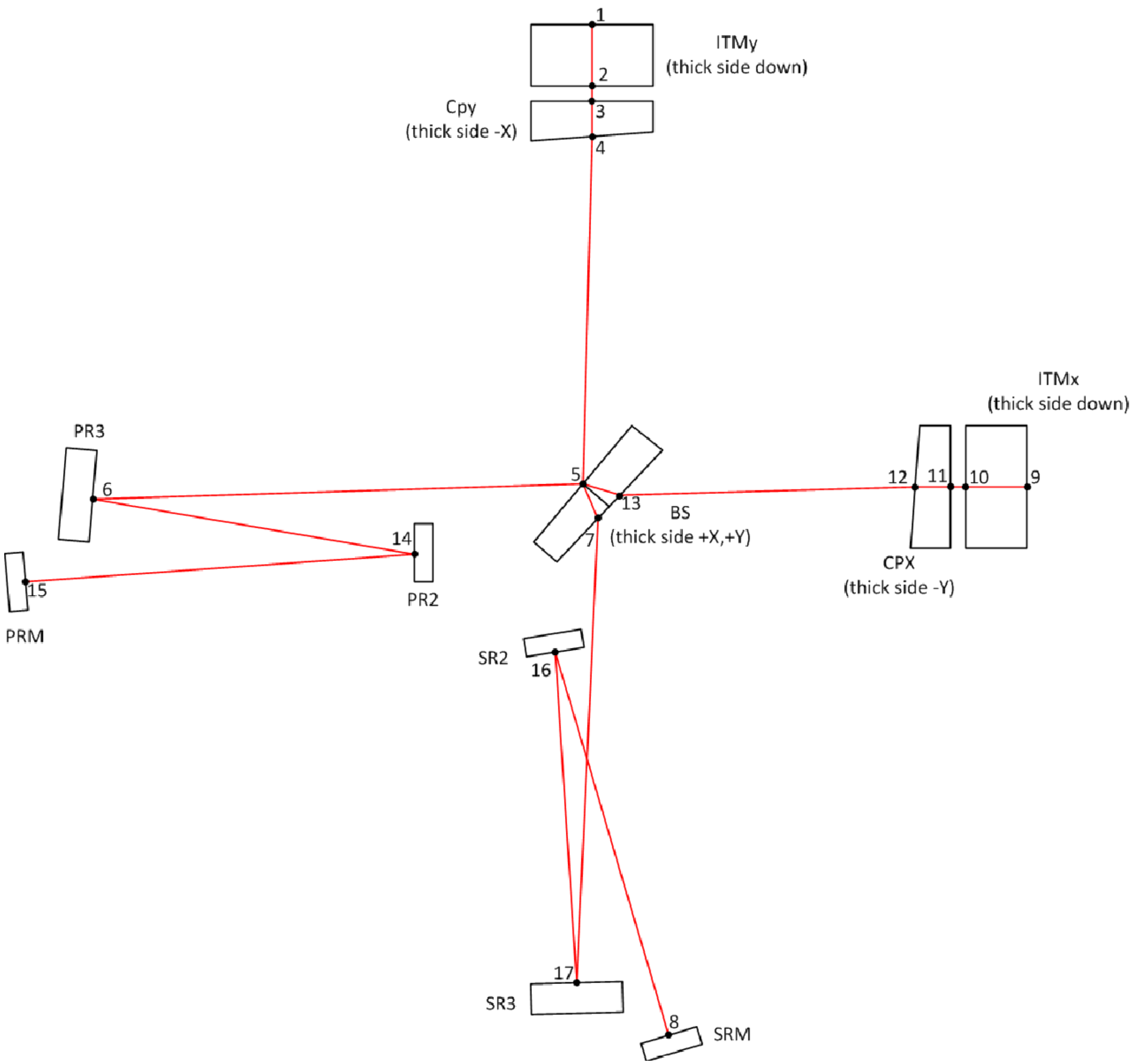
ar = anti-reflectance surface
hr = high reflectance surface
bs = beamsplitting surface

d12 = ITMy thickness at center, mm
d23 = ITMy to CPy gap distance, mm
d34 = CPy thickness at center, mm
d45 = CPy to BS(hr) distance, mm
d56 = BS(hr) to PR3 distance, mm
dBS = BS thickness at center, mm
d57 = BS(hr) to BS(ar) for y-beam, mm
d78 = BS(ar), at y-beam intercept, to SR3, mm
d513 = BS(hr) to BS(ar) for x-beam, mm
d910 = ITMx thickness at center, mm
d1011 = ITMx to CPx gap distance, mm
d1112 = CPx thickness at center, mm
d1213 = CPx to BS(ar) distance, mm
d614 = PR3 to PR2 distance, mm

d1415 = PR2 to PRM distance, mm

The “given” parameters (from Zemax layout, etc.) are as follows:
- ITM (hr) center positions from Zemax
- ITM thickness
- ITM to CP gaps
- CP thickness
- BS thickness

ITMx = vertical, thick side down
ITMy = vertical, thick side down
CPx = horizontal, thick side - Y
CPy = horizontal, thick side - X
BS = horizontal, thick side + X/+Y


1
ITMy
(thick side down)
2
Cpy
(thick side -X)
3
4
ITMx
(thick side down)
PR3
6
5
12
11
10
9
13
BS
(thick side +X,+Y)
14
7
15
PR2
CPX
(thick side -Y)
PRM
SR2
16
17
SR3
8
SRM

■ Intialization

(Debug) In[3582]:=

```
Off[General::"spell"]
Off[General::"spell1"]
```

(Debug) In[3584]:=

```
Needs["VectorAnalysis`"]
```

(Debug) In[3585]:=

```
Needs["Geometry`Rotations`"]
```

Needs[“Geometry`Rotations`”]
should no longer be needed. Use built - in RotationTransform and RotationMatrix
RotationMatrix uses yaw-pitch-roll angles rather than the Euler angles used by RotationMatrix3D
HOWEVER, didn’t Minimize function for determination of the BS orientation to work with RotateMatrix3D, so reverted to Rotate3D

(Debug) In[3586]:=

```
RotationMatrix3D[phi_,theta_,psi_]:=RotationMatrix[Pi - psi, {0, 0,
1}].RotationMatrix[
  theta, {1, 0, 0}].RotationMatrix[Pi - phi, {0, 0, 1}]
```

(Debug) In[3587]:=

```
RotationMatrix3D[phi, theta, psi]
```

(Debug) Out[3587]=

```
{{Cos[phi] Cos[psi] - Cos[theta] Sin[phi] Sin[psi],
  Cos[psi] Sin[phi] + Cos[phi] Cos[theta] Sin[psi], Sin[psi] Sin[theta]},
 {-Cos[psi] Cos[theta] Sin[phi] - Cos[phi] Sin[psi],
  Cos[phi] Cos[psi] Cos[theta] - Sin[phi] Sin[psi], Cos[psi] Sin[theta]},
 {Sin[phi] Sin[theta], -Cos[phi] Sin[theta], Cos[theta]}}
```

Note: Per J. Wertz, "Spacecraft Attitude Determination and Control", D. Reidel Pub., 1985, pp.763-764, This is a Type 2

Euler Angle representation with a z-x-z rotation sequence.

Needs["Graphics`Shapes`"]
The functionality of `RotateShape`, `TranslateShape`, and `AffineShape` is provided by the newly added kernel functions `Rotate`, `Translate`, `Scale` and `GeometricTransformation`

(Debug) In[3588]:=

```
pi = N[π, 10];
```

(Debug) In[3589]:=

```
Clear[Reflect];
Reflect[u_, n_] := Block[{temp, i, j}, temp = CrossProduct[u, n];
   i = temp / Sqrt[temp.temp]; j = CrossProduct[n, i]; Return[u.j j - u.n n]];
```

(Debug) In[3591]:=

```
Clear[Refract, ni, nt];
Refract[u_, n_, ni_, nt_] :=
  Block[{temp, i, j}, temp = CrossProduct[u, n]; i = temp / Sqrt[temp.temp];
   j = CrossProduct[n, i]; temp = ni u.j / nt; Return[temp j - Sqrt[1 - temp^2] n]];
```

Needs["Optica`Optica`"]
SetOptions[DrawSystem, QuickTrace -> False];

(Debug) In[3593]:=

```
DMS[rad_] := {IntegerPart[rad 180 / Pi],
   IntegerPart[FractionalPart[rad 180 / Pi] 60 + Sign[rad] 10^-18],
   Round[FractionalPart[(rad 180 / Pi) 60] 60]};
```

(Debug) In[3594]:=

```
RAD[deg_, min_, sec_] := (deg + min / 60 + sec / 3600) Pi / 180;
```

(Debug) In[3595]:=

```
formatedTablePositions := Grid[Flatten[{{
      {"Optic", "Global Coordinates\n(mm)", SpanFromLeft, SpanFromLeft,
       "Local Coordinates\n(mm)", SpanFromLeft, SpanFromLeft},
      {SpanFromAbove, "Xg", "Yg", "Zg", "Xl", "Yl", "Zl"}},
     Table[{opticLabel[[i]], NumberForm[opticPositionG[[i, 1]], {10, 1}],
       NumberForm[opticPositionG[[i, 2]], {10, 1}],
       NumberForm[opticPositionG[[i, 3]], {10, 1}], \
         NumberForm[opticPositionL[[i, 1]], {10, 1}],
       NumberForm[opticPositionL[[i, 2]], {10, 1}],
       NumberForm[opticPositionL[[i, 3]], {10, 1}]},
      {i, 1, nOptics}]}, 1], Frame → All];
```

(Debug) In[3596]:=

```
formatedTableNormals := Grid[Flatten[{{
      {"Optic", "Normal Unit Vector (global)\n(mm)", SpanFromLeft, SpanFromLef
       "Normal Unit Vector (local)\n(mm)", SpanFromLeft, SpanFromLeft},
      {SpanFromAbove, "Ug", "Vg", "Wg", "Ul", "Vl", "Wl"}},
     Table[{opticLabel[[i]], NumberForm[opticNormalG[[i, 1]], {10, 6}],
       NumberForm[opticNormalG[[i, 2]], {10, 6}],
       NumberForm[opticNormalG[[i, 3]], {10, 6}], \
         NumberForm[opticNormalL[[i, 1]], {10, 6},
        ExponentFunction -> (If[-30 < # < 30, Null, #] &)],
       NumberForm[opticNormalL[[i, 2]], {10, 6}, ExponentFunction ->
         (If[-30 < # < 30, Null, #] &)], NumberForm[opticNormalL[[i, 3]],
        {10, 6}, ExponentFunction -> (If[-30 < # < 30, Null, #] &)]},
      {i, 1, nOptics}]}, 1], Frame → All];
```

(Debug) In[3597]:=

```
formatedTableAngles := Grid[Flatten[{{
      {"Optic", "Yaw", "Yaw", SpanFromLeft, SpanFromLeft},
      {SpanFromAbove, "rad", "deg", "min", "sec"}},
     Table[{opticLabel[[i]], NumberForm[opticYaw[[i]], {10, 6},
        ExponentFunction -> (If[-10 < # < 10, Null, #] &)],
       NumberForm[DMS[opticYaw[[i]]][[1]], {10, 0}],
       NumberForm[DMS[opticYaw[[i]]][[2]], {10, 0}],
       NumberForm[DMS[opticYaw[[i]]][[3]], {10, 0}]},
      {i, 1, nOptics}]}, 1], Frame → All];
```

(Debug) In[3598]:=

```
formatedTableComparePositions := Grid[Flatten[{{
    {"Optic", "Global Coordinate\n Difference (mm)", SpanFromLeft,
     SpanFromLeft}, {SpanFromAbove, "ΔXg", "ΔYg", "ΔZg"}},
   Table[{compareOpticLabel[[i]], NumberForm[comparePositionG[[i, 1]],
      {10, 1}, ExponentFunction -> (If[-30 < # < 30, Null, #] &)],
     NumberForm[comparePositionG[[i, 2]], {10, 1}, ExponentFunction ->
       (If[-30 < # < 30, Null, #] &)], NumberForm[comparePositionG[[i, 3]],
      {10, 1}, ExponentFunction -> (If[-30 < # < 30, Null, #] &)]},
    {i, 1, nCompareOptics}]}, 1], Frame → All];
```

(Debug) In[3599]:=

```
formatedTableParameters := Grid[Flatten[{{
    {"Optic", "wedge", SpanFromLeft, "Thickness", "Diameter"},
    {SpanFromAbove, "rad", "deg", "mm", "mm"}},
   Table[{opticLabel[[i]], NumberForm[opticWedge[[i]], {10, 6}],
     NumberForm[If[opticWedge[[i]] != "NA", opticWedge[[i]] * 180 / pi, "NA"],
      {10, 6}], NumberForm[opticThick[[i]], {10, 6}],
     NumberForm[opticDiameter[[i]], {10, 6}]},
    {i, 1, nOptics}]}, 1], Frame → All];
```

■ Global to Local Coordinate Transformation Matrices

See Tables 10-14 and Tables 25-27 of T980044-v1(aka -10), "Determination of Local and Global Coordinate Axes for the LIGO Sites".

■ Rhc -> Hanford Corner station

(Debug) In[3600]:=

```
xangle = 619.49 × 10^-6;
yangle = 12.4832 × 10^-6;
```

(Debug) In[3602]:=

```
Rhc =
  RotationMatrix3D[0, yangle, 0].RotationMatrix3D[pi / 2, xangle, -pi / 2];
MatrixForm[
 Rhc]
```

(Debug) Out[3603]//MatrixForm=

$$\begin{pmatrix} 1. & 0. & -0.00061949 \\ 7.73322\times10^{-9} & 1. & 0.0000124832 \\ 0.00061949 & -0.0000124832 & 1. \end{pmatrix}$$

(Debug) In[3604]:=

```
MatrixForm[Rhc - IdentityMatrix[3]]
```

(Debug) Out[3604]//MatrixForm=

$$\begin{pmatrix} -1.91884\times10^{-7} & 0. & -0.00061949 \\ 7.73322\times10^{-9} & -7.79151\times10^{-11} & 0.0000124832 \\ 0.00061949 & -0.0000124832 & -1.91962\times10^{-7} \end{pmatrix}$$

(Debug) In[3605]:=

```
Rhcinv =
  RotationMatrix3D[Pi / 2, -xangle, -Pi / 2].RotationMatrix3D[0, -yangle, 0];
MatrixForm[Rhc.Rhcinv]
```

(Debug) Out[3606]//MatrixForm=

$$\begin{pmatrix} 1. & 1.65436\times10^{-24} & 1.0842\times10^{-19} \\ 1.65436\times10^{-24} & 1. & 0. \\ 1.0842\times10^{-19} & 0. & 1. \end{pmatrix}$$

(Debug) In[3607]:=

```
MatrixForm[Rhcinv]
```

(Debug) Out[3607]//MatrixForm=

$$\begin{pmatrix} 1. & 7.73322\times10^{-9} & 0.00061949 \\ 0. & 1. & -0.0000124832 \\ -0.00061949 & 0.0000124832 & 1. \end{pmatrix}$$

■ Rhxm -> Hanford x-mid station

(Debug) In[3608]:=

```
xangle = 305.827 × 10^-6;
yangle = 12.0075 × 10^-6;
```

(Debug) In[3610]:=

```
Rhxm =
  RotationMatrix3D[0, yangle, 0].RotationMatrix3D[Pi / 2, xangle, -Pi / 2];
MatrixForm[Rhxm]
```

(Debug) Out[3611]//MatrixForm=

$$\begin{pmatrix} 1. & 0. & -0.000305827 \\ 3.67222\times 10^{-9} & 1. & 0.0000120075 \\ 0.000305827 & -0.0000120075 & 1. \end{pmatrix}$$

(Debug) In[3612]:=

```
MatrixForm[Rhxm - IdentityMatrix[3]]
```

(Debug) Out[3612]//MatrixForm=

$$\begin{pmatrix} -4.67651\times 10^{-8} & 0. & -0.000305827 \\ 3.67222\times 10^{-9} & -7.209\times 10^{-11} & 0.0000120075 \\ 0.000305827 & -0.0000120075 & -4.68372\times 10^{-8} \end{pmatrix}$$

(Debug) In[3613]:=

```
Rhxminv =
  RotationMatrix3D[Pi / 2, -xangle, -Pi / 2]. RotationMatrix3D[0, -yangle, 0];
MatrixForm[Rhxm.Rhxminv]
```

(Debug) Out[3614]//MatrixForm=

$$\begin{pmatrix} 1. & 4.1359\times 10^{-25} & 0. \\ 4.1359\times 10^{-25} & 1. & 0. \\ 0. & 0. & 1. \end{pmatrix}$$

▪ Rhxe -> Hanford x-end station

(Debug) In[3615]:=

```
xangle = -7.8389 10^-6;
yangle = 11.5318 × 10^-6;
```

(Debug) In[3617]:=

```
Rhxe =
  RotationMatrix3D[0, yangle, 0].RotationMatrix3D[Pi / 2, xangle, -Pi / 2];
MatrixForm[Rhxe]
```

(Debug) Out[3618]//MatrixForm=

$$\begin{pmatrix} 1. & 0. & 7.8389 \times 10^{-6} \\ -9.03966 \times 10^{-11} & 1. & 0.0000115318 \\ -7.8389 \times 10^{-6} & -0.0000115318 & 1. \end{pmatrix}$$

(Debug) In[3619]:=

```
MatrixForm[Rhxe - IdentityMatrix[3]]
```

(Debug) Out[3619]//MatrixForm=

$$\begin{pmatrix} -3.07242 \times 10^{-11} & 0. & 7.8389 \times 10^{-6} \\ -9.03966 \times 10^{-11} & -6.64913 \times 10^{-11} & 0.0000115318 \\ -7.8389 \times 10^{-6} & -0.0000115318 & -9.72155 \times 10^{-11} \end{pmatrix}$$

(Debug) In[3620]:=

```
Rhxeinv =
  RotationMatrix3D[Pi / 2, -xangle, -Pi / 2].RotationMatrix3D[0, -yangle, 0];
MatrixForm[Rhxe.Rhxeinv]
```

(Debug) Out[3621]//MatrixForm=

$$\begin{pmatrix} 1. & 0. & 0. \\ 0. & 1. & 1.69407 \times 10^{-21} \\ 0. & 1.69407 \times 10^{-21} & 1. \end{pmatrix}$$

▪ Rhym -> Hanford y-mid station

(Debug) In[3622]:=

```
xangle = 619.97 × 10^-6;
yangle = 325.84 × 10^-6;
```

(Debug) In[3624]:=

```
Rhym =
  RotationMatrix3D[0, yangle, 0].RotationMatrix3D[Pi / 2, xangle, -Pi / 2];
MatrixForm[Rhym]
```

(Debug) Out[3625]//MatrixForm=

$$\begin{pmatrix} 1. & 0. & -0.00061997 \\ 2.02011\times10^{-7} & 1. & 0.00032584 \\ 0.00061997 & -0.00032584 & 1. \end{pmatrix}$$

(Debug) In[3626]:=

```
MatrixForm[Rhym - IdentityMatrix[3]]
```

(Debug) Out[3626]//MatrixForm=

$$\begin{pmatrix} -1.92181\times10^{-7} & 0. & -0.00061997 \\ 2.02011\times10^{-7} & -5.30859\times10^{-8} & 0.00032584 \\ 0.00061997 & -0.00032584 & -2.45267\times10^{-7} \end{pmatrix}$$

(Debug) In[3627]:=

```
Rhyminv =
  RotationMatrix3D[Pi / 2, -xangle, -Pi / 2]. RotationMatrix3D[0, -yangle, 0];
MatrixForm[Rhym.Rhyminv]
```

(Debug) Out[3628]//MatrixForm=

$$\begin{pmatrix} 1. & 2.64698\times10^{-23} & 0. \\ 2.64698\times10^{-23} & 1. & 0. \\ 0. & 0. & 1. \end{pmatrix}$$

■ Rhye -> Hanford y-end station

(Debug) In[3629]:=

```
xangle = 620.45 × 10^-6;
yangle = 639.20 × 10^-6;
```

(Debug) In[3631]:=

```
Rhye =
  RotationMatrix3D[0, yangle, 0].RotationMatrix3D[Pi / 2, xangle, -Pi / 2];
MatrixForm[Rhxe]
```

(Debug) Out[3632]//MatrixForm=

$$\begin{pmatrix} 1. & 0. & 7.8389 \times 10^{-6} \\ -9.03966 \times 10^{-11} & 1. & 0.0000115318 \\ -7.8389 \times 10^{-6} & -0.0000115318 & 1. \end{pmatrix}$$

(Debug) In[3633]:=

```
MatrixForm[Rhye - IdentityMatrix[3]]
```

(Debug) Out[3633]//MatrixForm=

$$\begin{pmatrix} -1.92479 \times 10^{-7} & 0. & -0.00062045 \\ 3.96592 \times 10^{-7} & -2.04288 \times 10^{-7} & 0.0006392 \\ 0.00062045 & -0.0006392 & -3.96767 \times 10^{-7} \end{pmatrix}$$

(Debug) In[3634]:=

```
Rhyeinv =
  RotationMatrix3D[Pi / 2, -xangle, -Pi / 2]. RotationMatrix3D[0, -yangle, 0];
MatrixForm[Rhye.Rhyeinv]
```

(Debug) Out[3635]//MatrixForm=

$$\begin{pmatrix} 1. & -5.29396 \times 10^{-23} & 0. \\ -5.29396 \times 10^{-23} & 1. & -1.0842 \times 10^{-19} \\ 0. & -1.0842 \times 10^{-19} & 1. \end{pmatrix}$$

(Debug) In[3636]:=

```
(180 / Pi) 619 × 10^-6 // N
```

(Debug) Out[3636]=

```
0.0354661
```

■ Rlc -> Livingston Corner station

(Debug) In[3637]:=

```
xangle = 312.0 × 10^-6;
yangle = -611.0 10^-6;
```

(Debug) In[3639]:=

```
Rlc =
  RotationMatrix3D[0, yangle, 0].RotationMatrix3D[Pi / 2, xangle, -Pi / 2];
MatrixForm[
 Rlc]
```

(Debug) Out[3640]//MatrixForm=

$$\begin{pmatrix} 1. & 0. & -0.000312 \\ -1.90632\times10^{-7} & 1. & -0.000611 \\ 0.000312 & 0.000611 & 1. \end{pmatrix}$$

(Debug) In[3641]:=

```
MatrixForm[Rlc - IdentityMatrix[3]]
```

(Debug) Out[3641]//MatrixForm=

$$\begin{pmatrix} -4.8672\times10^{-8} & 0. & -0.000312 \\ -1.90632\times10^{-7} & -1.8666\times10^{-7} & -0.000611 \\ 0.000312 & 0.000611 & -2.35332\times10^{-7} \end{pmatrix}$$

(Debug) In[3642]:=

```
Rlcinv =
  RotationMatrix3D[Pi / 2, -xangle, -Pi / 2] . RotationMatrix3D[0, -yangle, 0];
MatrixForm[Rlc.Rlcinv]
```

(Debug) Out[3643]//MatrixForm=

$$\begin{pmatrix} 1. & 0. & 0. \\ 0. & 1. & 1.0842\times10^{-19} \\ 0. & 1.0842\times10^{-19} & 1. \end{pmatrix}$$

■ Rlxe -> Livingston x-end station

(Debug) In[3644]:=

```
xangle = -315.0 10^-6;
yangle = -610.0 10^-6;
```

(Debug) In[3646]:=

```
Rlxe =
  RotationMatrix3D[0, yangle, 0].RotationMatrix3D[Pi / 2, xangle, -Pi / 2];
MatrixForm[Rlxe]
```

(Debug) Out[3647]//MatrixForm=

$$\begin{pmatrix} 1. & 0. & 0.000315 \\ 1.9215\times10^{-7} & 1. & -0.00061 \\ -0.000315 & 0.00061 & 1. \end{pmatrix}$$

(Debug) In[3648]:=

```
MatrixForm[Rlxe - IdentityMatrix[3]]
```

(Debug) Out[3648]//MatrixForm=

$$\begin{pmatrix} -4.96125\times10^{-8} & 0. & 0.000315 \\ 1.9215\times10^{-7} & -1.8605\times10^{-7} & -0.00061 \\ -0.000315 & 0.00061 & -2.35662\times10^{-7} \end{pmatrix}$$

(Debug) In[3649]:=

```
Rlxeinv =
  RotationMatrix3D[Pi / 2, -xangle, -Pi / 2].RotationMatrix3D[0, -yangle, 0];
MatrixForm[Rlxe.Rlxeinv]
```

(Debug) Out[3650]//MatrixForm=

$$\begin{pmatrix} 1. & 0. & 0. \\ 0. & 1. & 0. \\ 0. & 0. & 1. \end{pmatrix}$$

(Debug) In[3651]:=

```
MatrixForm[Rlxeinv - Transpose[Rlxe]]
```

(Debug) Out[3651]//MatrixForm=

$$\begin{pmatrix} 0. & 0. & 0. \\ 0. & 0. & 0. \\ 0. & 0. & 0. \end{pmatrix}$$

■ Rlye -> Livingston y-end station

(Debug) In[3652]:=

```
xangle = 311.0 × 10^-6;
yangle = 18.8 × 10^-6;
```

(Debug) In[3654]:=

```
Rlye =
  RotationMatrix3D[0, yangle, 0].RotationMatrix3D[Pi / 2, xangle, -Pi / 2];
MatrixForm[Rlye]
```

(Debug) Out[3655]//MatrixForm=

$$\begin{pmatrix} 1. & 0. & -0.000311 \\ 5.8468\times10^{-9} & 1. & 0.0000188 \\ 0.000311 & -0.0000188 & 1. \end{pmatrix}$$

(Debug) In[3656]:=

```
MatrixForm[Rlye - IdentityMatrix[3]]
```

(Debug) Out[3656]//MatrixForm=

$$\begin{pmatrix} -4.83605\times10^{-8} & 0. & -0.000311 \\ 5.8468\times10^{-9} & -1.7672\times10^{-10} & 0.0000188 \\ 0.000311 & -0.0000188 & -4.85372\times10^{-8} \end{pmatrix}$$

(Debug) In[3657]:=

```
Rlyeinv =
  RotationMatrix3D[Pi / 2, -xangle, -Pi / 2].RotationMatrix3D[0, -yangle, 0];
MatrixForm[Rlye.Rlyeinv]
```

(Debug) Out[3658]//MatrixForm=

$$\begin{pmatrix} 1. & 0. & 0. \\ 0. & 1. & 0. \\ 0. & 0. & 1. \end{pmatrix}$$

Vector Analysis for Wedge Position & Orientation Determination

■ Parameters

wl = 1.064;
nOptic = ModelRefractiveIndex[FusedSilica][WaveLength -> wl]
nAir = ModelRefractiveIndex[Air][WaveLength -> wl]
nVacuum = ModelRefractiveIndex[Vacuum][WaveLength -> wl]

(Debug) In[3659]:=

```
nOptic = 1.44963;
nAir = 1.0;
nVacuum = 1.0;
```

Note : D0901920-v12 &-v13 pdf file reports nOptic = 1.44963. In fact Zemax uses Suprasil-Ext index for 20C, zero pressure at 1064 nm which is 1.4500310
According to
- I. H. Malitson. Interspecimen Comparison of the Refractive Index of Fused Silica, J. Opt. Soc. Am. 55, 1205-1208 (1965) doi:10.1364/JOSA.55.001205
- Handbook of Optics, 3rd edition, Vol. 4. McGraw-Hill 2009
as calculated at this URL:
http://refractiveindex.info/wiki/Citing_RefractiveIndex.INFO
the index at 1064 nm is 1.44963

H1

Notes

1) The ITM optic assignments of specific serial numbered optics is given in T1200324-v2. The assignments of the CP and BS optics, and the parameters of each COC optic is given by serial number at https://nebula.ligo.caltech.edu/optics/, with the exception of CPy. GariLynn states that CP09 must be replaced (currently part of H2-ITMy destined to become H1-ITMy) and has chosen CP02 for its replacement.

ITMx is ITM09
ITMy is ITM06
CPx is CP01
CPy is CP02
BS is BS06

2) The magnitude and orientation of the wedge angles are as follows::
ITMx = 0.077 deg, vertical, thick side down
ITMy = 0.078 deg, vertical, thick side down
CPx = 0.069 deg, horizontal, thick side -Y
CPy = 0.069 deg, horizontal, thick side -X
BS = 0.076 deg, horizontal, thick side +X/+Y
The wedge angle values given in https://nebula.ligo.caltech.edu/optics/ are the "final" values reported by the polishing contractor. However there is some round-off/approx. in this data. GariLynn Billingsley suggests using the values reported in C1107164-v1.

■ H1 Unique Parameters

(Debug) In[3662]:=

```
wedgeITMx = 0.077*pi/180;
wedgeITMy = 0.078*pi/180;
wedgeCPx = 0.069*pi/180;
wedgeCPy = 0.069*pi/180;
wedgeBS = 0.076*pi/180;
```

(Debug) In[3667]:=

```
ITMxThick = 200.22;
ITMxDiameter = 340.13;
ITMyThick = 199.64;
ITMyDiameter = 340.06;
CPxThick = 99.82;
CPxDiameter = 340.13;
CPyThick = 99.91;
CPyDiameter = 340.22;
BSThick = 60.41;
BSDiameter = 369.85;
```

(Debug) In[3677]:=

```
p1 = {-200, 4983.1, -80};
p9 = {5013, -200, -80};
d12 = ITMyThick - (ITMyDiameter / 2) Tan[wedgeITMy];
d910 = ITMxThick - (ITMxDiameter / 3) Tan[wedgeITMx];
d23 = 20;
d1011 = 20;
d34 = CPyThick - (CPyDiameter / 2) Tan[wedgeCPy];
d1112 = CPxThick - (CPxDiameter / 2) Tan[wedgeCPx];
dBS = BSThick - (BSDiameter / 2) Tan[wedgeBS];
```

Note that the distances d513 and d57 are approximate -- they do not take into account the wedge angle of the BS

optic surface orientation determination

normal vectors for ITMs and CPs

(Debug) In[3686]:=
```
n1 = {0, 1, 0};
n2=RotationMatrix3D[0,wedgeITMy,0].n1
```
(Debug) Out[3687]=
```
{0., 0.999999, -0.00136136}
```
(Debug) In[3688]:=
```
n3 = n2;
n4 = RotationMatrix3D[0, 0, -wedgeCPy].n3
```
(Debug) Out[3689]=
```
{-0.00120428, 0.999998, -0.00136136}
```
(Debug) In[3690]:=
```
n9 = {1, 0, 0};
n10 = RotationMatrix3D[pi/2, -wedgeITMx, -(pi/2)].n9
```
(Debug) Out[3691]=
```
{0.999999, 0., -0.0013439}
```
(Debug) In[3692]:=
```
n11 = n10;
n12 = RotationMatrix3D[0, 0, wedgeCPx].n11
```
(Debug) Out[3693]=
```
{0.999998, -0.00120428, -0.0013439}
```

ITMy path to BS

(Debug) In[3694]:=

```
u1 = -n1;
w1 = u1;
u2 = w1;
w2 = Refract[u2, n2, nOptic, nVacuum]
```

(Debug) Out[3697]=

{0., -1., -0.000612107}

(Debug) In[3698]:=

```
u3 = w2;
w3 = Refract[u3, n3, nVacuum, nOptic]
```

(Debug) Out[3699]=

$\{0., -1., 2.1684 \times 10^{-19}\}$

(Debug) In[3700]:=

```
u4 = w3;
w4 = Refract[u4, n4, nOptic, nVacuum]
```

(Debug) Out[3701]=

{-0.00054148, -1., -0.000612108}

(Debug) In[3702]:=

```
u5y = w4;
```

ITMx path to BS

(Debug) In[3703]:=

```
u9 = -n9;
w9 = u9;
u10 = w9;
w10 = Refract[u10, n10, nOptic, nVacuum]
```

(Debug) Out[3706]=

```
{-1., 0., -0.00060426}
```

(Debug) In[3707]:=

```
u11 = w10;
w11 = Refract[u11, n11, nVacuum, nOptic]
```

(Debug) Out[3708]=

$\{-1., 0., -2.1684\times 10^{-19}\}$

(Debug) In[3709]:=

```
u12 = w11;
w12 = Refract[u12, n12, nOptic, nVacuum]
```

(Debug) Out[3710]=

```
{-1., -0.00054148, -0.000604261}
```

(Debug) In[3711]:=

```
u13 = w12;
```

Tried to convert from Rotate3D to RotationMatrix3D and didn’t get this to work, so reverting to old function Rotate3D in package “Geometry`Rotations`”
soln = Minimize[Abs[Reflect[u5y, RotationMatrix3D[-yaw, 0, 0].{0, 1, 0}].Refract[Refract[u13, RotatationMatrix3D[wedgeBS, 0, 0].-RotationMatrix3D[yaw, 0, 0].{0, 1, 0}, nVacuum, nOptic], -RotationMatrix3D[yaw, 0, 0].{0, 1, 0}, nOptic, nVacuum]], yaw]

Note also that I have assumed a zero pitch angle. This is approximately correct because the BS has no vertical wedge component. However the reflected and refracted rays have a pitch angle difference of ~8 microrad (very small).

(Debug) In[3712]:=

```
Clear[yaw];
soln = Maximize[
  Reflect[u5y, Rotate3D[{0, 1, 0}, -yaw, 0, 0]].Refract[Refract[u13,
     Rotate3D[-Rotate3D[{0, 1, 0}, -yaw, 0, 0], wedgeBS, 0, 0], nVacuum,
     nOptic], -Rotate3D[{0, 1, 0}, -yaw, 0, 0], nOptic, nVacuum], yaw]
```

(Debug) Out[3713]=

```
{1., {yaw → 0.784873}}
```

(Debug) In[3714]:=

```
BSyaw = -yaw /. soln[[2]];
BSyawDeg = BSyaw 180 / pi
```

(Debug) Out[3715]=

```
-44.9699
```

(Debug) In[3716]:=

```
n5y = RotationMatrix3D[BSyaw, 0, 0].{0, 1, 0}
n5x = -n5y;
```

(Debug) Out[3716]=

```
{-0.706736, 0.707478, 0.}
```

(Debug) In[3718]:=

```
v5 = Reflect[u5y, n5y]
```

(Debug) Out[3718]=

```
{-1., 0.000507929, -0.000612108}
```

(Debug) In[3719]:=

```
w5y = Refract[u5y, n5y, nVacuum, nOptic]
```

(Debug) Out[3719]=

```
{0.271849, -0.96234, -0.000422251}
```

(Debug) In[3720]:=

```
n13 = RotationMatrix3D[wedgeBS, 0, 0].n5x;
n7 = -n13
```

(Debug) Out[3721]=

```
{-0.705797, 0.708415, 0.}
```

(Debug) In[3722]:=

```
u7 = w5y;
w7 = Refract[u7, n7, nOptic, nVacuum]
```

(Debug) Out[3723]=

```
{0.00050405, -1., -0.000612108}
```

(Debug) In[3724]:=

```
w13 = Refract[u13, n13, nVacuum, nOptic];
u5x = w13;
w5x = Refract[u5x, n5x, nOptic, nVacuum]
```

(Debug) Out[3726]=

```
{-1., 0.000507926, -0.000604261}
```

■ BS position determination

(Debug) In[3727]:=

```
d513 = dBS / (-w13.n5x)
```

(Debug) Out[3727]=

```
68.9204
```

(Debug) In[3728]:=

```
d57 = dBS / (w5y.n5x)
```

(Debug) Out[3728]=

```
68.9204
```

(Debug) In[3729]:=

```
p2 = p1 + d12 w1
```

(Debug) Out[3729]=

```
{-200., 4783.69, -80.}
```

(Debug) In[3730]:=

```
p3 = p2 + d23 w2
```

(Debug) Out[3730]=

```
{-200., 4763.69, -80.0122}
```

(Debug) In[3731]:=

```
p4 = p3 + d34 w3
```

(Debug) Out[3731]=

```
{-200., 4663.99, -80.0122}
```

(Debug) In[3732]:=

```
p10 = p9 + d910 w9
```

(Debug) Out[3732]=

```
{4812.93, -200., -80.}
```

(Debug) In[3733]:=

```
p11 = p10 + d1011 w10
```

(Debug) Out[3733]=

```
{4792.93, -200., -80.0121}
```

(Debug) In[3734]:=

```
p12 = p11 + d1112 w11
```

(Debug) Out[3734]=

```
{4693.32, -200., -80.0121}
```

(Debug) In[3735]:=

```
Clear[dBSy, dBSx]
soln = Minimize[
  EuclideanDistance[p4 + dBSy w4, p12 + dBSx w12 + d513 w13], {dBSy, dBSx}]
```

(Debug) Out[3736]=

```
{0.0204444, {dBSy → 4847.8, dBSx → 4829.64}}
```

(Debug) In[3737]:=

```
d45 = dBSy /. soln[[2]];
d1213 = dBSx /. soln[[2]];
```

(Debug) In[3739]:=

```
p5 = p4 + d45 w4
p13 = p12 + d1213 w12
p13 + d513 w13
```

(Debug) Out[3739]=

```
{-202.625, -183.81, -82.9796}
```

(Debug) Out[3740]=

```
{-136.32, -202.615, -82.9304}
```

(Debug) Out[3741]=

```
{-202.625, -183.81, -82.9592}
```

(Debug) In[3742]:=

```
p7 = p5 + d57 u7
```

(Debug) Out[3742]=

```
{-183.889, -250.134, -83.0087}
```

■ PR3 position determination

Take as a given the global x - coordinate of the PR3 from the IO layout (E1100494-v3)

(Debug) In[3743]:=

```
pPR3 = {-19 740, -177.4, -94.6};
```

(Debug) In[3744]:=

```
xPR3 = pPR3[[1]];
```

(Debug) In[3745]:=

```
d56 = (xPR3 - p5[[1]]) / v5[[1]]
```

(Debug) Out[3745]=

```
19 537.4
```

(Debug) In[3746]:=

```
p6 = p5 + d56 v5
```

(Debug) Out[3746]=

```
{-19 740., -173.886, -94.9386}
```

■ PR2 position determination

PR3 is constrained to be vertical (zero pitch angle) in local coordinates (to minimize vertical bounce mode-to-length coupling). Consequently the normal vector for this optic must be pitched in global coordinates.
Take as a given the global x and y coordinates of the PR2 from the IO layout (because the IO layout position is used in the SolidWorks layouts used to physcially place the suspensions

pPR2 from IO layout (E1100495-v4) should equal p14

(Debug) In[3747]:=
```
pPR2io = {-3581.3, -530.4, -84.3};
```

(Debug) In[3748]:=
```
u6 = v5;
```

(Debug) In[3749]:=
```
Clear[s, yaw]
soln = Minimize[EuclideanDistance[
   p6 + s Reflect[u6, Rhc.Rotate3D[{1, 0, 0}, -yaw, 0, 0]], pPR2io], {s, yaw}]
```

(Debug) Out[3750]=
```
{0.504593, {s → 16162.6, yaw → -0.0112838}}
```

(Debug) In[3751]:=
```
PR3yaw = yaw /. soln[[2]];
d614 = s /. soln[[2]];
n6 = Rhc.Rotate3D[{1, 0, 0}, -PR3yaw, 0, 0]
v6 = Reflect[u6, n6];
p14 = p6 + d614 v6
```

(Debug) Out[3753]=
```
{0.999936, -0.0112836, 0.000619591}
```

(Debug) Out[3755]=
```
{-3581.3, -530.4, -84.8046}
```

■ PRM position determination

PR2 is constrained to be vertical (zero pitch angle) in local coordinates (to minimize vertical bounce mode-to-length coupling). Consequently the normal vector for this optic must be pitched in global coordinates.
Take as a given the global x and y coordinates of the PRM from the IO layout (because the IO layout position is used

in the SolidWorks layouts used to physcially place the suspensions
pPRM from IO layout (E1100494-v3) should equal p15

(Debug) In[3756]:=
```
pPRMio = {-20 189.6, -628.0, -94.2};
```

(Debug) In[3757]:=
```
u14 = v6;
```

(Debug) In[3758]:=
```
Clear[s, yaw]
soln = Minimize[EuclideanDistance[p14 +
    s Reflect[u14, Rhc.Rotate3D[{-1, 0, 0}, -yaw, 0, 0]], pPRMio], {s, yaw}]
```

(Debug) Out[3759]=
```
{0.769346, {s → 16 608.6, yaw → -0.0080916}}
```

(Debug) In[3760]:=
```
PR2yaw = yaw /. soln[[2]];
d1415 = s /. soln[[2]];
n14 = Rhc.Rotate3D[{-1, 0, 0}, -PR2yaw, 0, 0]
v14 = Reflect[u14, n14];
p15 = p14 + d1415 v14
```

(Debug) Out[3762]=
```
{-0.999967, 0.00809151, -0.000619571}
```

(Debug) Out[3764]=
```
{-20 189.6, -628., -94.9693}
```

(Debug) In[3765]:=
```
u15 = v14;
n15 = -v14;
Clear[yaw]
soln = Maximize[n15.Rotate3D[{1, 0, 0}, -yaw, 0, 0], {yaw}];
PRMyaw = yaw /. soln[[2]]
```
(Debug) Out[3769]=
```
0.00587651
```

■ SR3 position determination

Take as a given the global y - coordinate of the SR3 from Zemax layout (D0901920-v13).
pSR3 from Zemax should equal p8

(Debug) In[3770]:=
```
pSR3 = {-174.2, -19615.9, -94.5};
ySR3 = pSR3[[2]];
```
(Debug) In[3772]:=
```
d78 = (ySR3 - p7[[2]]) / w7[[2]]
```
(Debug) Out[3772]=
```
19365.8
```
(Debug) In[3773]:=
```
p8 = p7 + d78 w7
```
(Debug) Out[3773]=
```
{-174.128, -19615.9, -94.8627}
```
(Debug) In[3774]:=
```
n15 = -v14;
```

SR2 position determination

SR3 is constrained to be vertical (zero pitch angle) in local coordinates (to minimize vertical bounce mode-to-length coupling). Consequently the normal vector for this optic must be pitched in global coordinates.
Take as a given the global x and y coordinates of the SR2 from the Zemax layout (because the Zemax layout positions for the SRC are used in the SolidWorks layouts used to physcially place the suspensions
pSR2 from Zemax (D0902216-v8) should equal p16

(Debug) In[3775]:=
```
pSR2zemax = {-594.1, -4178.1, -104.4};
```

(Debug) In[3776]:=
```
u8 = w7;
```

(Debug) In[3777]:=
```
Clear[s, yaw]
soln = Minimize[
  EuclideanDistance[p8 + s Reflect[u8, Rhc.Rotate3D[{0, 1, 0}, -yaw, 0, 0]],
   pSR2zemax], {s, yaw}]
```

NMinimize::cvmit : Failed to converge to the requested accuracy or precision within 100 iterations. ≫

(Debug) Out[3778]=
```
{0.566256, {s → 15 443.5, yaw → 0.0138507}}
```

(Debug) In[3779]:=
```
SR3yaw = yaw /. soln[[2]];
d816 = s /. soln[[2]];
n8 = Rhc.Rotate3D[{0, 1, 0}, -SR3yaw, 0, 0]
v8 = Reflect[u8, n8];
p16 = p8 + d816 v8
```

(Debug) Out[3781]=
```
{-0.0138503, 0.999904, -0.0000210621}
```

(Debug) Out[3783]=
```
{-594.1, -4178.1, -104.966}
```

■ SRM position determination

SR2 is constrained to be vertical (zero pitch angle) in local coordinates (to minimize vertical bounce mode-to-length coupling). Consequently the normal vector for this optic must be pitched in global coordinates.
Take as a given the global x and y coordinates of the PRM from the Zemax layout (because the Zemax SRC layout positions are used in the SolidWorks layouts used to physcially place the suspensions
pSRM from Zemax (D0901920-v13) should equal p17

(Debug) In[3784]:=
```
pSRMzemax = {305.4, -19 908.6, -113.2};
```

(Debug) In[3785]:=
```
u16 = v8;
```

(Debug) In[3786]:=

```
Clear[s, yaw]
soln = Minimize[EuclideanDistance[
   p16 + s Reflect[u16, Rhc.Rotate3D[{0, -1, 0}, -yaw, 0, 0]],
   pSRMzemax], {s, yaw}]
```

(Debug) Out[3787]=

```
{0.858763, {s → 15756.2, yaw → 0.0421586}}
```

(Debug) In[3788]:=

```
SR2yaw = yaw /. soln[[2]];
d1617 = s /. soln[[2]];
n16 = Rhc.Rotate3D[{0, -1, 0}, -SR2yaw, 0, 0]
v16 = Reflect[u16, n16];
p17 = p16 + d1617 v16
```

(Debug) Out[3790]=

```
{0.0421461, -0.999111, 0.0000385812}
```

(Debug) Out[3792]=

```
{305.4, -19908.6, -114.059}
```

(Debug) In[3793]:=

```
u17 = v16;
n17 = -v16;
Clear[yaw]
soln = Maximize[n17.Rotate3D[{0, 1, 0}, -yaw, 0, 0], {yaw}];
SRMyaw = yaw /. soln[[2]]
```

(Debug) Out[3797]=

```
0.0571197
```

PRC, SRC & Schnupp Assymmetry Lengths

According to T0900043-v11, the recycling cavity lengths are supposed to be:
PRC length = 57656.0 mm
SRC length = 56008.0 mm
The Schnupp Assymmetry is supposed to be 80 mm, according to RODA M1200276-v1

(Debug) In[3798]:=

```
dPRC = d1415 + d614 + d56 + ((nOptic d12 + d23 + nOptic d34 + d45) +
      (nOptic d910 + d1011 + nOptic d1112 + d1213 + nOptic d513)) / 2
```

(Debug) Out[3798]=

```
57 651.3
```

(Debug) In[3799]:=

```
dSRC = d1617 + d816 + d78 + nOptic d57 + ((nOptic d12 + d23 + nOptic d34 + d45) +
      (nOptic d910 + d1011 + nOptic d1112 + d1213 + nOptic d513)) / 2
```

(Debug) Out[3799]=

```
56 008.1
```

(Debug) In[3800]:=

```
schnuppAssy = Abs[(nOptic d12 + d23 + nOptic d34 + d45) -
    (nOptic d910 + d1011 + nOptic d1112 + d1213 + nOptic d513)]
```

(Debug) Out[3800]=

```
82.5752
```

Summary

Optic Parameters

(Debug) In[3801]:=

```
opticLabel = {"PRM HR", "PR2 HR", "PR3 HR", "BS HR", "BS ARs",
   "SRM HR", "SR2 HR", "SR3 HR", "CPx", "ITMx", "CPy", "ITMy"};
nOptics = Length[opticLabel];
opticWedge = {"NA", "NA", "NA", wedgeBS, "NA",
   "NA", "NA", "NA", wedgeCPx, wedgeITMx, wedgeCPy, wedgeITMy};
opticThick = {"NA", "NA", "NA", BSThick, "NA", "NA", "NA",
   "NA", CPxThick, ITMxThick, CPyThick, ITMyThick};
opticDiameter = {"NA", "NA", "NA", BSDiameter, "NA", "NA", "NA",
   "NA", CPxDiameter, ITMxDiameter, CPyDiameter, ITMyDiameter};
```

(Debug) In[3806]:=

```
formatedTableParameters
```

(Debug) Out[3806]=

Optic	wedge		Thickness	Diameter
	rad	deg	mm	mm
PRM HR	NA	NA	NA	NA
PR2 HR	NA	NA	NA	NA
PR3 HR	NA	NA	NA	NA
BS HR	0.001326	0.076000	60.410000	369.850000
BS ARs	NA	NA	NA	NA
SRM HR	NA	NA	NA	NA
SR2 HR	NA	NA	NA	NA
SR3 HR	NA	NA	NA	NA
CPx	0.001204	0.069000	99.820000	340.130000
ITMx	0.001344	0.077000	200.220000	340.130000
CPy	0.001204	0.069000	99.910000	340.220000
ITMy	0.001361	0.078000	199.640000	340.060000

■ Results

(Debug) In[3807]:=

```
opticPositionG = {p15, p14, p6, p5, p7, p17, p16, p8, p11, p9, p3, p1};
opticPositionL = opticPositionG.Rhc;
opticNormalG = {n15, n14, n6, n5y, n13, n17, n16, n8, n11, n9, n3, n1};
opticNormalL = opticNormalG.Rhc;
opticYaw =
  {PRMyaw, PR2yaw, PR3yaw, BSyaw, 0, SRMyaw, SR2yaw, SR3yaw, 0, 0, 0, 0};
```

(Debug) In[3812]:=

```
formatedTablePositions
```

(Debug) Out[3812]=

Optic	Global Coordinates (mm)			Local Coordinates (mm)		
	Xg	Yg	Zg	Xl	Yl	Zl
PRM HR	-20189.6	-628.0	-95.0	-20189.7	-628.0	-82.5
PR2 HR	-3581.3	-530.4	-84.8	-3581.4	-530.4	-82.6
PR3 HR	-19740.0	-173.9	-94.9	-19740.1	-173.9	-82.7
BS HR	-202.6	-183.8	-83.0	-202.7	-183.8	-82.9
BS ARs	-183.9	-250.1	-83.0	-183.9	-250.1	-82.9
SRM HR	305.4	-19908.6	-114.1	305.3	-19908.6	-114.5
SR2 HR	-594.1	-4178.1	-105.0	-594.2	-4178.1	-104.7
SR3 HR	-174.1	-19615.9	-94.9	-174.2	-19615.9	-95.0
CPx	4792.9	-200.0	-80.0	4792.9	-200.0	-83.0
ITMx	5013.0	-200.0	-80.0	5012.9	-200.0	-83.1
CPy	-200.0	4763.7	-80.0	-200.0	4763.7	-79.8
ITMy	-200.0	4983.1	-80.0	-200.0	4983.1	-79.8

(Debug) In[3813]:=

```
formatedTableNormals
```

(Debug) Out[3813]=

Optic	Normal Unit Vector (global) (mm)			Normal Unit Vector (local) (mm)		
	Ug	Vg	Wg	Ul	Vl	Wl
PRM HR	0.999983	0.005876	0.000612	0.999983	0.005876	-0.000007
PR2 HR	-0.999967	0.008092	-0.000620	-0.999967	0.008092	-0.000000
PR3 HR	0.999936	-0.011284	0.000620	0.999936	-0.011284	0.000000
BS HR	-0.706736	0.707478	0.000000	-0.706736	0.707478	0.000447
BS ARs	0.705797	-0.708415	0.000000	0.705796	-0.708415	-0.000446
SRM HR	-0.057089	0.998369	0.000577	-0.057088	0.998369	0.000625
SR2 HR	0.042146	-0.999111	0.000039	0.042146	-0.999111	0.000000
SR3 HR	-0.013850	0.999904	-0.000021	-0.013850	0.999904	0.000000
CPx	0.999999	0.000000	-0.001344	0.999998	0.000000	-0.001963
ITMx	1.000000	0.000000	0.000000	1.000000	0.000000	-0.000619
CPy	0.000000	0.999999	-0.001361	-0.000001	0.999999	-0.001349
ITMy	0.000000	1.000000	0.000000	0.000000	1.000000	0.000012

(Debug) In[3814]:=

```
formatedTableAngles
```

(Debug) Out[3814]=

Optic	Yaw	Yaw		
	rad	deg	min	sec
PRM HR	0.005877	0.	20.	12.
PR2 HR	−0.008092	0.	−27.	−49.
PR3 HR	−0.011284	0.	−38.	−47.
BS HR	−0.784873	−44.	−58.	−12.
BS ARs	0.000000	0.	0.	0.
SRM HR	0.057120	3.	16.	22.
SR2 HR	0.042159	2.	24.	56.
SR3 HR	0.013851	0.	47.	37.
CPx	0.000000	0.	0.	0.
ITMx	0.000000	0.	0.	0.
CPy	0.000000	0.	0.	0.
ITMy	0.000000	0.	0.	0.

- **Zemax results**

Zemax results are from D0901920-v13
The wedge angle magnitudes (but not oreintations/signs) reported in the Zemax optical layout, D0901920-v13 differ, as follows:

ITMx = ITM10? = 0.077 deg (same value but serial number designation is wrong)
ITMy = ITM11? = 0.076 deg (value and serial number incorrect)
CPx = CP01 = 0.070 deg (serial number correct, used “final” wedge angle instead of C1107164-v1 value)
CPy = CP02 = 0.069 deg (serial number is incorrect)
BS = BS06 = 0.076 deg (same)

(Debug) In[3815]:=

```
opticLabelZemax = {"PRM HR", "PR2 HR", "PR3 HR", "BS HR", "BS ARs",
   "SRM HR", "SR2 HR", "SR3 HR", "ITMx", "ITMy", "ETMx", "ETMy"};
nOpticsZemax = Length[opticLabelZemax];
opticPosGZemax = {{-20 194.3, -628.0, -95.8}, {-3581.7, -530.4, -84.5},
   {-19 740.5, -174.0, -94.8}, {-202.6, -183.9, -82.9},
   {-184.0, -249.8, -82.9}, {305.4, -19 908.6, -113.2},
   {-594.1, -4178.1, -104.4}, {-174.2, -19 615.9, -94.5},
   {5013.0, -200.0, -80.0}, {-200.0 , 4983.1, -80.0},
   {3 999 498.0, -200.0, -80.0}, {-200.0, 3 999 468.1, -80.0}};
```

(Debug) In[3818]:=

```
compareOpticLabel = opticLabelZemax[[1 ;; 10]];
nCompareOptics = 10;
comparePositionG =
  Drop[opticPositionG, {9, 11, 2}] - opticPosGZemax[[1 ;; 10]];
```

(Debug) In[3821]:=

```
formatedTableComparePositions
```

(Debug) Out[3821]=

Optic	Global Coordinate Difference (mm)		
	ΔXg	ΔYg	ΔZg
PRM HR	4.7	−0.0	0.8
PR2 HR	0.4	−0.0	−0.3
PR3 HR	0.5	0.1	−0.1
BS HR	−0.0	0.1	−0.1
BS ARs	0.1	−0.3	−0.1
SRM HR	0.0	0.0	−0.9
SR2 HR	0.0	−0.0	−0.6
SR3 HR	0.1	0.0	−0.4
ITMx	0.0	0.0	0.0
ITMy	0.0	0.0	0.0

- **IO results**

IO Layout results for PRM, PR2 and PR3 are from E1100494-v3, E1100495-v4

(Debug) In[3822]:=

```
opticLabelIO = {"PRM HR", "PR2 HR", "PR3 HR"};
nOpticsIO = Length[opticLabelIO];
opticPosGIO = {{-20189.6, -628, -94.2},
   {-3581.3, -530.4, -84.3}, {-19740, -177.4, -94.6}};
```

(Debug) In[3825]:=

```
compareOpticLabel = opticLabelIO[[1 ;; 3]];
nCompareOptics = Length[compareOpticLabel];
comparePositionG = opticPositionG[[1 ;; 3]] - opticPosGIO;
```

(Debug) In[3828]:=

```
formatedTableComparePositions
```

(Debug) Out[3828]=

Optic	Global Coordinate Difference (mm)		
	ΔXg	ΔYg	ΔZg
PRM HR	0.0	-0.0	-0.8
PR2 HR	0.0	-0.0	-0.5
PR3 HR	-0.0	3.5	-0.3

■ L1

■ Notes

1) The ITM optic assignments of specific serial numbered optics is given in T1200324-v2. The assignments of the CP and BS optics, and the parameters of each COC optic is given by serial number at https://nebula.ligo.caltech.edu/optics/, with the exception of CPy. GariLynn states that CP09 must be replaced (currently part of H2-ITMy destined to become H1-ITMy) and has chosen CP02 for its replacement.

ITMx is ITM04

ITMy is ITM08

CPx is CP06

CPy is CP08

BS is BS02

2) The magnitude and orientation of the wedge angles are as follows::

ITMx = 0.0725 deg, vertical, thick side down

ITMy = 0.074 deg, vertical, thick side down

CPx = 0.073 deg, horizontal, thick side -Y

CPy = 0.066 deg, horizontal, thick side -X

BS = 0.070 deg, horizontal, thick side +X/+Y

The wedge angle values given in https://nebula.ligo.caltech.edu/optics/ are the "final" values reported by the polishing contractor. However there is some round-off/approx. in this data. GariLynn Billingsley suggests using the values reported in C1107164-v1.

■ L1 Unique Parameters

optic thicknesses are the as-built values reported in C1107164-v1 for the assigned optics

(Debug) In[3829]:=

```
wedgeITMx = 0.0725*pi/180;
wedgeITMy = 0.074*pi/180;
wedgeCPx = 0.073*pi/180;
wedgeCPy = 0.066*pi/180;
wedgeBS = 0.070*pi/180;
```

(Debug) In[3834]:=

```
ITMxThick = 200.27;
ITMxDiameter = 340.0;
ITMyThick = 199.61;
ITMyDiameter = 339.92;
CPxThick = 100.31;
CPxDiameter = 339.94;
CPyThick = 100.32;
CPyDiameter = 340.11;
BSThick = 59.88;
BSDiameter = 369.98;
```

(Debug) In[3844]:=

```
p1 = {-200, 4983.1, -80};
p9 = {5013, -200, -80};
d12 = ITMyThick - (ITMyDiameter / 2) Tan[wedgeITMy];
d910 = ITMxThick - (ITMxDiameter / 3) Tan[wedgeITMx];
d23 = 20;
d1011 = 20;
d34 = CPyThick - (CPyDiameter / 2) Tan[wedgeCPy];
d1112 = CPxThick - (CPxDiameter / 2) Tan[wedgeCPx];
dBS = BSThick - (BSDiameter / 2) Tan[wedgeBS];
```

Note that the distances d513 and d57 are approximate -- they do not take into account the wedge angle of the BS

▪ optic surface orientation determination

▪ normal vectors for ITMs and CPs

(Debug) In[3853]:=

```
n1 = {0, 1, 0};
n2=RotationMatrix3D[0,wedgeITMy,0].n1
```

(Debug) Out[3854]=

```
{0., 0.999999, -0.00129154}
```

(Debug) In[3855]:=

```
n3 = n2;
n4 = RotationMatrix3D[0, 0, -wedgeCPy].n3
```

(Debug) Out[3856]=

```
{-0.00115192, 0.999999, -0.00129154}
```

(Debug) In[3857]:=

```
n9 = {1, 0, 0};
n10 = RotationMatrix3D[pi/2, -wedgeITMx, -(pi/2)].n9
```

(Debug) Out[3858]=

```
{0.999999, 0., -0.00126536}
```

(Debug) In[3859]:=

```
n11 = n10;
n12 = RotationMatrix3D[0, 0, wedgeCPx].n11
```

(Debug) Out[3860]=

```
{0.999998, -0.00127409, -0.00126536}
```

ITMy path to BS

(Debug) In[3861]:=

```
u1 = -n1;
w1 = u1;
u2 = w1;
w2 = Refract[u2, n2, nOptic, nVacuum]
```

(Debug) Out[3864]=

```
{0., -1., -0.000580717}
```

(Debug) In[3865]:=

```
u3 = w2;
w3 = Refract[u3, n3, nVacuum, nOptic]
```

(Debug) Out[3866]=

```
{0., -1., 0.}
```

(Debug) In[3867]:=

```
u4 = w3;
w4 = Refract[u4, n4, nOptic, nVacuum]
```

(Debug) Out[3868]=

```
{-0.000517937, -1., -0.000580718}
```

(Debug) In[3869]:=

```
u5y = w4;
```

■ ITMx path to BS

(Debug) In[3870]:=

```
u9 = -n9;
w9 = u9;
u10 = w9;
w10 = Refract[u10, n10, nOptic, nVacuum]
```

(Debug) Out[3873]=

```
{-1., 0., -0.000568946}
```

(Debug) In[3874]:=

```
u11 = w10;
w11 = Refract[u11, n11, nVacuum, nOptic]
```

(Debug) Out[3875]=

```
{-1., 0., 0.}
```

(Debug) In[3876]:=

```
u12 = w11;
w12 = Refract[u12, n12, nOptic, nVacuum]
```

(Debug) Out[3877]=

```
{-1., -0.00057287, -0.000568947}
```

(Debug) In[3878]:=

```
u13 = w12;
```

Tried to convert from Rotate3D to RotationMatrix3D and didn’t get this to work, so reverting to old function Rotate3D in package “Geometry`Rotations`”
soln = Minimize[Abs[Reflect[u5y, RotationMatrix3D[-yaw, 0, 0].{0, 1, 0}].Refract[Refract[u13, RotatationMatrix3D[wedgeBS, 0, 0].-RotationMatrix3D[yaw, 0, 0].{0, 1, 0}, nVacuum, nOptic], -RotationMatrix3D[yaw, 0, 0].{0, 1, 0}, nOptic, nVacuum]], yaw]

Note also that I have assumed a zero pitch angle. This is approximately correct because the BS has no vertical wedge component. However the reflected and refracted rays have a pitch angle difference of ~8 microrad (very small).

(Debug) In[3879]:=
```
Clear[yaw];
soln = Maximize[
  Reflect[u5y, Rotate3D[{0, 1, 0}, -yaw, 0, 0]].Refract[Refract[u13,
     Rotate3D[-Rotate3D[{0, 1, 0}, -yaw, 0, 0], wedgeBS, 0, 0], nVacuum,
     nOptic], -Rotate3D[{0, 1, 0}, -yaw, 0, 0], nOptic, nVacuum], yaw]
```

(Debug) Out[3880]=
```
{1., {yaw → 0.784942}}
```

(Debug) In[3881]:=
```
BSyaw = -yaw /. soln[[2]];
BSyawDeg = BSyaw 180 / pi
```

(Debug) Out[3882]=
```
-44.9739
```

(Debug) In[3883]:=
```
n5y = RotationMatrix3D[BSyaw, 0, 0].{0, 1, 0}
n5x = -n5y;
```

(Debug) Out[3883]=
```
{-0.706784, 0.707429, 0.}
```

(Debug) In[3885]:=
```
v5 = Reflect[u5y, n5y]
```

(Debug) Out[3885]=
```
{-1., 0.000393622, -0.000580718}
```

(Debug) In[3886]:=
```
w5y = Refract[u5y, n5y, nVacuum, nOptic]
```
(Debug) Out[3886]=
```
{0.271891, -0.962328, -0.000400597}
```
(Debug) In[3887]:=
```
n13 = RotationMatrix3D[wedgeBS, 0, 0].n5x;
n7 = -n13
```
(Debug) Out[3888]=
```
{-0.70592, 0.708292, 0.}
```
(Debug) In[3889]:=
```
u7 = w5y;
w7 = Refract[u7, n7, nOptic, nVacuum]
```
(Debug) Out[3890]=
```
{0.000445259, -1., -0.000580718}
```
(Debug) In[3891]:=
```
w13 = Refract[u13, n13, nVacuum, nOptic];
u5x = w13;
w5x = Refract[u5x, n5x, nOptic, nVacuum]
```
(Debug) Out[3893]=
```
{-1., 0.000393615, -0.000568947}
```

■ BS position determination

(Debug) In[3894]:=
```
d513 = dBS / (-w13.n5x)
```
(Debug) Out[3894]=
```
68.3363
```

(Debug) In[3895]:=

```
d57 = dBS / (w5y.n5x)
```

(Debug) Out[3895]=

```
68.3363
```

(Debug) In[3896]:=

```
p2 = p1 + d12 w1
```

(Debug) Out[3896]=

```
{-200., 4783.71, -80.}
```

(Debug) In[3897]:=

```
p3 = p2 + d23 w2
```

(Debug) Out[3897]=

```
{-200., 4763.71, -80.0116}
```

(Debug) In[3898]:=

```
p4 = p3 + d34 w3
```

(Debug) Out[3898]=

```
{-200., 4663.59, -80.0116}
```

(Debug) In[3899]:=

```
p10 = p9 + d910 w9
```

(Debug) Out[3899]=

```
{4812.87, -200., -80.}
```

(Debug) In[3900]:=

```
p11 = p10 + d1011 w10
```

(Debug) Out[3900]=

```
{4792.87, -200., -80.0114}
```

(Debug) In[3901]:=
```
p12 = p11 + d1112 w11
```
(Debug) Out[3901]=
```
{4692.78, -200., -80.0114}
```
(Debug) In[3902]:=
```
Clear[dBSy, dBSx]
soln = Minimize[
  EuclideanDistance[p4 + dBSy w4, p12 + dBSx w12 + d513 w13], {dBSy, dBSx}]
```
(Debug) Out[3903]=
```
{0.040814, {dBSy → 4847.71, dBSx → 4829.55}}
```
(Debug) In[3904]:=
```
d45 = dBSy /. soln[[2]];
d1213 = dBSx /. soln[[2]];
```
(Debug) In[3906]:=
```
p5 = p4 + d45 w4
p13 = p12 + d1213 w12
p13 + d513 w13
```
(Debug) Out[3906]=
```
{-202.511, -184.127, -82.8268}
```
(Debug) Out[3907]=
```
{-136.766, -202.767, -82.7591}
```
(Debug) Out[3908]=
```
{-202.511, -184.127, -82.786}
```

(Debug) In[3909]:=
```
p7 = p5 + d57 u7
```
(Debug) Out[3909]=
```
{-183.931, -249.889, -82.8541}
```

PR3 position determination

Take as a given the global x - coordinate of the PR3 from the IO layout (because the IO layout position is used in the SolidWorks layouts used to physcially place the suspensions
pPR3 from the IO layout (E1100492-v11, E1100493-v9) should equal p6

(Debug) In[3910]:=
```
pPR3io = {-19 740.0, -177.4, -94.5};
xPR3 = pPR3io[[1]];
```
(Debug) In[3912]:=
```
d56 = (xPR3 - p5[[1]]) / v5[[1]]
```
(Debug) Out[3912]=
```
19 537.5
```
(Debug) In[3913]:=
```
p6 = p5 + d56 v5
```
(Debug) Out[3913]=
```
{-19 740., -176.436, -94.1725}
```

PR2 position determination

PR3 is constrained to be vertical (zero pitch angle) in local coordinates (to minimize vertical bounce mode-to-length coupling). Consequently the normal vector for this optic must be pitched in global coordinates.
Take as a given the global x and y coordinates of the PR2 from the IO layout (because the IO layout position is used in

the SolidWorks layouts used to physcially place the suspensions
pPR2 from the IO layout (E1100492-v11, E1100493-v9) should equal p14

(Debug) In[3914]:=

```
pPR2io = {-3579.2, -530.4, -94.1};
```

(Debug) In[3915]:=

```
u6 = v5;
```

(Debug) In[3916]:=

```
Clear[s, yaw]
soln = Minimize[EuclideanDistance[
   p6 + s Reflect[u6, Rlc.Rotate3D[{1, 0, 0}, -yaw, 0, 0]], pPR2io], {s, yaw}]
```

(Debug) Out[3917]=

```
{0.405734, {s → 16164.7, yaw → -0.0111462}}
```

(Debug) In[3918]:=

```
PR3yaw = yaw /. soln[[2]];
d614 = s /. soln[[2]];
n6 = Rlc.Rotate3D[{1, 0, 0}, -PR3yaw, 0, 0]
v6 = Reflect[u6, n6];
p14 = p6 + d614 v6
```

(Debug) Out[3920]=

```
{0.999938, -0.0111461, 0.00030517}
```

(Debug) Out[3922]=

```
{-3579.2, -530.4, -93.6943}
```

■ PRM position determination

PR2 is constrained to be vertical (zero pitch angle) in local coordinates (to minimize vertical bounce mode-to-length coupling). Consequently the normal vector for this optic must be pitched in global coordinates.

Take as a given the global x and y coordinates of the PRM from the IO layout (because the IO layout position is used in the SolidWorks layouts used to physcially place the suspensions
pPRM from the IO layout (E1100492-v11, E1100493-v9) should equal p15

(Debug) In[3923]:=
```
pPRMio = {-20 189.6, -628.0, -104.1};
```

(Debug) In[3924]:=
```
u14 = v6;
```

(Debug) In[3925]:=
```
Clear[s, yaw]
soln = Minimize[EuclideanDistance[p14 +
    s Reflect[u14, Rlc.Rotate3D[{-1, 0, 0}, -yaw, 0, 0]], pPRMio], {s, yaw}]
```

(Debug) Out[3926]=
```
{0.69607, {s → 16 610.7, yaw → -0.00801148}}
```

(Debug) In[3927]:=
```
PR2yaw = yaw /. soln[[2]];
d1415 = s /. soln[[2]];
n14 = Rlc.Rotate3D[{-1, 0, 0}, -PR2yaw, 0, 0]
v14 = Reflect[u14, n14];
p15 = p14 + d1415 v14
```

(Debug) Out[3929]=
```
{-0.999968, 0.00801159, -0.000307095}
```

(Debug) Out[3931]=
```
{-20 189.6, -628., -103.404}
```

(Debug) In[3932]:=

```
u15 = v14;
n15 = -v14;
Clear[yaw]
soln = Maximize[n15.Rotate3D[{1, 0, 0}, -yaw, 0, 0], {yaw}];
PRMyaw = yaw /. soln[[2]]
```

(Debug) Out[3936]=

```
0.00587578
```

■ SR3 position determination

Take as a given the global y - coordinate of the SR3 from Zemax.
pSR3 from Zemax should equal p8

(Debug) In[3937]:=

```
pSR3 = {-175.2, -19 615.9, -94.1};
ySR3 = pSR3[[2]];
```

(Debug) In[3939]:=

```
d78 = (ySR3 - p7[[2]]) / w7[[2]]
```

(Debug) Out[3939]=

```
19 366.
```

(Debug) In[3940]:=

```
p8 = p7 + d78 w7
```

(Debug) Out[3940]=

```
{-175.308, -19 615.9, -94.1003}
```

(Debug) In[3941]:=

```
n15 = -v14;
```

■ SR2 position determination

SR3 is constrained to be vertical (zero pitch angle) in local coordinates (to minimize vertical bounce mode-to-length coupling). Consequently the normal vector for this optic must be pitched in global coordinates.
Take as a given the global x and y coordinates of the SR2 from the Zemax layout (because the Zemax layout positions for the SRC are used in the SolidWorks layouts used to physcially place the suspensions
pSR2 from Zemax (D0902216-v8) should equal p16

(Debug) In[3942]:=

```
pSR2zemax = {-594.1, -4178.1, -84.4};
```

(Debug) In[3943]:=

```
u8 = w7;
```

(Debug) In[3944]:=

```
Clear[s, yaw]
soln = Minimize[
  EuclideanDistance[p8 + s Reflect[u8, Rlc.Rotate3D[{0, 1, 0}, -yaw, 0, 0]],
   pSR2zemax], {s, yaw}]
```

NMinimize::cvmit : Failed to converge to the requested accuracy or precision within 100 iterations. ≫

(Debug) Out[3945]=

```
{0.0670134, {s → 15443.5, yaw → 0.0137832}}
```

(Debug) In[3946]:=
```
SR3yaw = yaw /. soln[[2]];
d816 = s /. soln[[2]];
n8 = Rlc.Rotate3D[{0, 1, 0}, -SR3yaw, 0, 0]
v8 = Reflect[u8, n8];
p16 = p8 + d816 v8
```

(Debug) Out[3948]=
```
{-0.0137827, 0.999905, 0.000606642}
```

(Debug) Out[3950]=
```
{-594.1, -4178.1, -84.333}
```

SRM position determination

SR2 is constrained to be vertical (zero pitch angle) in local coordinates (to minimize vertical bounce mode-to-length coupling). Consequently the normal vector for this optic must be pitched in global coordinates.
Take as a given the global x and y coordinates of the PRM from the Zemax layout (because the Zemax SRC layout positions are used in the SolidWorks layouts used to physcially place the suspensions
pSRM from Zemax (D0902216-v8) should equal p17

(Debug) In[3951]:=
```
pSRMzemax = {305.0, -19 908.6, -93.2};
```

(Debug) In[3952]:=
```
u16 = v8;
```

(Debug) In[3953]:=

```
Clear[s, yaw]
soln = Minimize[EuclideanDistance[
   p16 + s Reflect[u16, Rlc.Rotate3D[{0, -1, 0}, -yaw, 0, 0]],
   pSRMzemax], {s, yaw}]
```

(Debug) Out[3954]=

```
{0.0111363, {s → 15 756.2, yaw → 0.0421077}}
```

(Debug) In[3955]:=

```
SR2yaw = yaw /. soln[[2]];
d1617 = s /. soln[[2]];
n16 = Rlc.Rotate3D[{0, -1, 0}, -SR2yaw, 0, 0]
v16 = Reflect[u16, n16];
p17 = p16 + d1617 v16
```

(Debug) Out[3957]=

```
{0.0420952, -0.999113, -0.000597325}
```

(Debug) Out[3959]=

```
{305., -19 908.6, -93.1889}
```

(Debug) In[3960]:=

```
u17 = v16;
n17 = -v16;
Clear[yaw]
soln = Maximize[n17.Rotate3D[{0, 1, 0}, -yaw, 0, 0], {yaw}];
SRMyaw = yaw /. soln[[2]]
```

(Debug) Out[3964]=

```
0.0570944
```

■ PRC, SRC & Schnupp Assymmetry Lengths

According to T0900043-v11, the recycling cavity lengths are supposed to be:
PRC length = 57656.0 mm
SRC length = 56008.0 mm
The Schnupp Assymmetry is supposed to be 80 mm, according to RODA M1200276-v1

(Debug) In[3965]:=

```
dPRC = d1415 + d614 + d56 + ((nOptic d12 + d23 + nOptic d34 + d45) +
      (nOptic d910 + d1011 + nOptic d1112 + d1213 + nOptic d513)) / 2
```

(Debug) Out[3965]=

```
57655.7
```

(Debug) In[3966]:=

```
dSRC = d1617 + d816 + d78 + nOptic d57 + ((nOptic d12 + d23 + nOptic d34 + d45) +
      (nOptic d910 + d1011 + nOptic d1112 + d1213 + nOptic d513)) / 2
```

(Debug) Out[3966]=

```
56007.6
```

(Debug) In[3967]:=

```
schnuppAssy = Abs[(nOptic d12 + d23 + nOptic d34 + d45) -
    (nOptic d910 + d1011 + nOptic d1112 + d1213 + nOptic d513)]
```

(Debug) Out[3967]=

```
81.9188
```

Summary

Optic Parameters

(Debug) In[3968]:=

```
opticLabel = {"PRM HR", "PR2 HR", "PR3 HR", "BS HR", "BS ARs",
   "SRM HR", "SR2 HR", "SR3 HR", "CPx", "ITMx", "CPy", "ITMy"};
nOptics = Length[opticLabel];
opticWedge = {"NA", "NA", "NA", wedgeBS, "NA",
   "NA", "NA", "NA", wedgeCPx, wedgeITMx, wedgeCPy, wedgeITMy};
opticThick = {"NA", "NA", "NA", BSThick, "NA", "NA", "NA",
   "NA", CPxThick, ITMxThick, CPyThick, ITMyThick};
opticDiameter = {"NA", "NA", "NA", BSDiameter, "NA", "NA", "NA",
   "NA", CPxDiameter, ITMxDiameter, CPyDiameter, ITMyDiameter};
```

(Debug) In[3973]:=

```
formatedTableParameters
```

(Debug) Out[3973]=

Optic	wedge		Thickness	Diameter
	rad	deg	mm	mm
PRM HR	NA	NA	NA	NA
PR2 HR	NA	NA	NA	NA
PR3 HR	NA	NA	NA	NA
BS HR	0.001222	0.070000	59.880000	369.980000
BS ARs	NA	NA	NA	NA
SRM HR	NA	NA	NA	NA
SR2 HR	NA	NA	NA	NA
SR3 HR	NA	NA	NA	NA
CPx	0.001274	0.073000	100.310000	339.940000
ITMx	0.001265	0.072500	200.270000	340.000000
CPy	0.001152	0.066000	100.320000	340.110000
ITMy	0.001292	0.074000	199.610000	339.920000

Results

(Debug) In[3974]:=

```
opticLabel = {"PRM HR", "PR2 HR", "PR3 HR", "BS HR", "BS ARs",
   "SRM HR", "SR2 HR", "SR3 HR", "CPx", "ITMx", "CPy", "ITMy"};
nOptics = Length[opticLabel];
opticPositionG = {p15, p14, p6, p5, p7, p17, p16, p8, p11, p9, p3, p1};
opticPositionL = opticPositionG.Rlc;
opticNormalG = {n15, n14, n6, n5y, n13, n17, n16, n8, n11, n9, n3, n1};
opticNormalL = opticNormalG.Rlc;
opticYaw =
  {PRMyaw, PR2yaw, PR3yaw, BSyaw, 0, SRMyaw, SR2yaw, SR3yaw, 0, 0, 0, 0};
```

(Debug) In[3981]:=

```
formatedTablePositions
```

(Debug) Out[3981]=

Optic	Global Coordinates (mm)			Local Coordinates (mm)		
	Xg	Yg	Zg	Xl	Yl	Zl
PRM HR	−20189.6	−628.0	−103.4	−20189.6	−628.1	−96.7
PR2 HR	−3579.2	−530.4	−93.7	−3579.2	−530.5	−92.3
PR3 HR	−19740.0	−176.4	−94.2	−19740.0	−176.5	−87.9
BS HR	−202.5	−184.1	−82.8	−202.5	−184.2	−82.7
BS ARs	−183.9	−249.9	−82.9	−184.0	−249.9	−82.6
SRM HR	305.0	−19908.6	−93.2	305.0	−19908.7	−81.1
SR2 HR	−594.1	−4178.1	−84.3	−594.1	−4178.2	−81.6
SR3 HR	−175.3	−19615.9	−94.1	−175.3	−19616.0	−82.1
CPx	4792.9	−200.0	−80.0	4792.8	−200.0	−81.4
ITMx	5013.0	−200.0	−80.0	5013.0	−200.0	−81.4
CPy	−200.0	4763.7	−80.0	−200.0	4763.7	−82.9
ITMy	−200.0	4983.1	−80.0	−200.0	4983.1	−83.0

(Debug) In[3982]:=

```
formatedTableNormals
```

(Debug) Out[3982]=

Optic	Normal Unit Vector (global) (mm)			Normal Unit Vector (local) (mm)		
	Ug	Vg	Wg	Ul	Vl	Wl
PRM HR	0.999983	0.005876	0.000585	0.999983	0.005876	0.000269
PR2 HR	−0.999968	0.008012	−0.000307	−0.999968	0.008011	0.000000
PR3 HR	0.999938	−0.011146	0.000305	0.999938	−0.011146	0.000000
BS HR	−0.706784	0.707429	0.000000	−0.706785	0.707429	−0.000212
BS ARs	0.705920	−0.708292	0.000000	0.705920	−0.708292	0.000213
SRM HR	−0.057063	0.998370	0.000562	−0.057063	0.998371	−0.000030
SR2 HR	0.042095	−0.999113	−0.000597	0.042095	−0.999114	−0.000000
SR3 HR	−0.013783	0.999905	0.000607	−0.013783	0.999905	−0.000000
CPx	0.999999	0.000000	−0.001265	0.999999	−0.000001	−0.001577
ITMx	1.000000	0.000000	0.000000	1.000000	0.000000	−0.000312
CPy	0.000000	0.999999	−0.001292	−0.000001	0.999998	−0.001903
ITMy	0.000000	1.000000	0.000000	−0.000000	1.000000	−0.000611

(Debug) In[3983]:=

```
formatedTableAngles
```

(Debug) Out[3983]=

Optic	Yaw	Yaw		
	rad	deg	min	sec
PRM HR	0.005876	0.	20.	12.
PR2 HR	−0.008011	0.	−27.	−32.
PR3 HR	−0.011146	0.	−38.	−19.
BS HR	−0.784942	−44.	−58.	−26.
BS ARs	0.000000	0.	0.	0.
SRM HR	0.057094	3.	16.	17.
SR2 HR	0.042108	2.	24.	45.
SR3 HR	0.013783	0.	47.	23.
CPx	0.000000	0.	0.	0.
ITMx	0.000000	0.	0.	0.
CPy	0.000000	0.	0.	0.
ITMy	0.000000	0.	0.	0.

- **Zemax results**

Zemax results are from D0902216 - v8

The wedge angle magnitudes (but not oreintations/signs) reported in the Zemax optical layout, D0902216-v8 match the serial numbers and wedge angle values given above.

(Debug) In[3984]:=

```
opticLabelZemax = {"PRM HR", "PR2 HR", "PR3 HR", "BS HR", "BS ARs",
   "SRM HR", "SR2 HR", "SR3 HR", "ITMx", "ITMy", "ETMx", "ETMy"};
nOpticsZemax = Length[opticLabelZemax];
opticPosGZemax = {{-20190.0, -628.0, -102.8}, {-3579.6, -530.4, -93.3},
   {-19740.5, -176.3, -93.9}, {-202.5, -184.0, -82.8},
   {-183.9, -250.0, -82.9}, {305.0, -19908.6, -93.2},
   {-594.1, -4178.1, -84.4}, {-175.2, -19615.9, -94.1},
   {5013.0, -200.0, -80.0}, {-200.0, 4983.1, -80.0},
   {3999498.0, -200.0, -80.0}, {-200.0 , 3999468.1, -80.0}};
```

(Debug) In[3987]:=

```
compareOpticLabel = opticLabelZemax[[1 ;; 10]];
nCompareOptics = 10;
comparePositionG =
  Drop[opticPositionG, {9, 11, 2}] - opticPosGZemax[[1 ;; 10]];
```

(Debug) In[3990]:=

```
formatedTableComparePositions
```

(Debug) Out[3990]=

Optic	Global Coordinate Difference (mm)		
	ΔXg	ΔYg	ΔZg
PRM HR	0.4	-0.0	-0.6
PR2 HR	0.4	-0.0	-0.4
PR3 HR	0.5	-0.1	-0.3
BS HR	-0.0	-0.1	-0.0
BS ARs	-0.0	0.1	0.0
SRM HR	-0.0	0.0	0.0
SR2 HR	0.0	0.0	0.1
SR3 HR	-0.1	0.0	-0.0
ITMx	0.0	0.0	0.0
ITMy	0.0	0.0	0.0

- **IO results**

IO Layout results for PRM, PR2 and PR3 are from E1100492-v11, E1100493-v9

(Debug) In[3991]:=

```
opticLabelIO = {"PRM HR", "PR2 HR", "PR3 HR"};
nOpticsIO = Length[opticLabelIO];
opticPosGIO = {{-20 189.6, -628, -104.1},
   {-3579.2, -530.4, -94.1}, {-19 740, -177.4, -94.5}};
```

(Debug) In[3994]:=

```
compareOpticLabel = opticLabelIO[[1 ;; 3]];
nCompareOptics = Length[compareOpticLabel];
comparePositionG = opticPositionG[[1 ;; 3]] - opticPosGIO;
```

(Debug) In[3997]:=

```
formatedTableComparePositions
```

(Debug) Out[3997]=

Optic	Global Coordinate Difference (mm)		
	ΔXg	ΔYg	ΔZg
PRM HR	-0.0	-0.0	0.7
PR2 HR	-0.0	-0.0	0.4
PR3 HR	-0.0	1.0	0.3