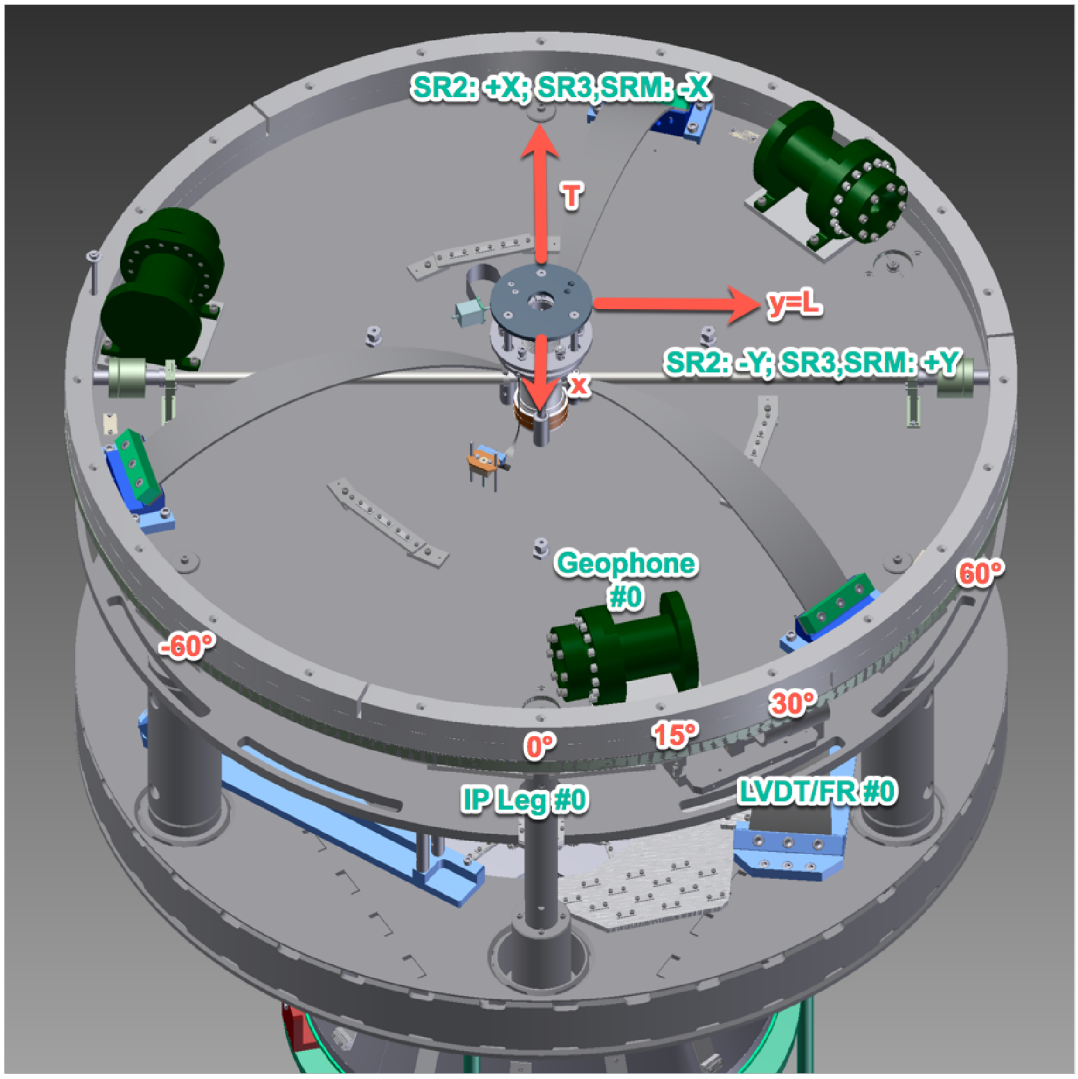
T1807663-v1 - SR Matrix Calc

2018-02-15, Mark Barton

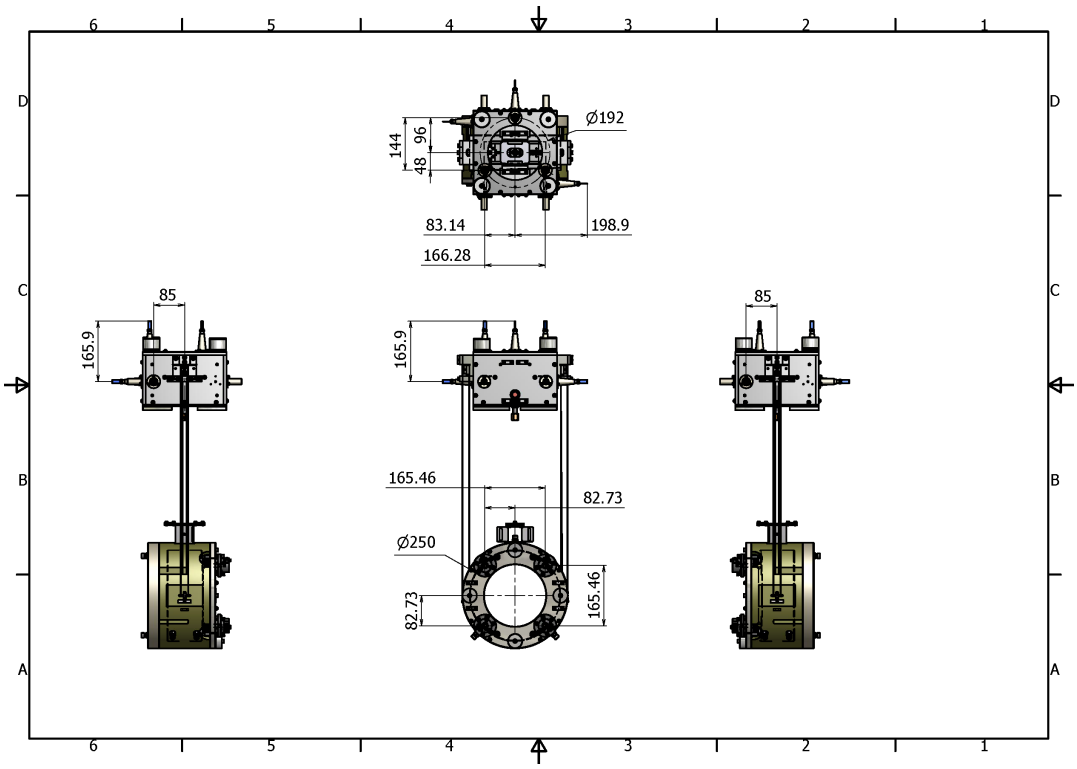
Data

X and Y are the standard interferometer global coordinates.
The HR surface faces -Y for SR2 and +Y for SR3 and SRM, and defines L (longitudinal).
T (transverse) is 90° anticlockwise from L (so as to make right handed LTV with vertical up).
For convenience, so that standard rotation matrices can be used, the calculation is done first in x/y coordinates where y=L and x=-T.
Leg #0 is in the +x direction and angles are measured anticlockwise from it.
Geophone #0 and LVDT/FR #0 are offset by 15° and 30° from Leg #0. (Note: in the BS, the FR was at -30°, but in the SR, the FR and LVDT are colocated at 30°.)
Legs #1 and #2 are at 120° and 240°.

PI Data



Payload Data



Data Summary

In[156]:=
```
vals = {
    (* IP stuff *)
    R -> 715. / 1000, (* PI radius *)
    Fradius -> 1268.8 / 1000 / 2, (* from center to FR stepper; 3D CAD *)
    Gradius -> 1183. / 1000 / 2, (* from center to geophone;3D CAD *)
    Lradius -> 1188. / 1000 / 2, (* from center to LVDT;
    3D CAD; CHECK ME! *)
    Eradius → 650. / 1000, (* from center to EQ stop hole; guesstimate *)
    Iradius → 650. / 1000, (* from center to leg; guesstimate *)
    Eradius2 → 80. / 1000 / 2, (* radius of EQ stop hole; guesstimate *)
    Iradius2 → 80. / 1000 / 2, (* radius of leg; guesstimate *)
    l -> 200. / 1000, (* arrow length, for diagram *)
    origin -> 0 °, (* 3 o'clock *)
    (* IM stuff *)
    IMV1L → 0.096,
    IMV2L → 0.096 * Sin[Pi / 6],
    IMV3L → 0.096 * Sin[Pi / 6],
    IMV1T → 0,
    IMV2T →  0.096 * Cos[Pi / 6],
    IMV3T →  0.096 * Cos[Pi / 6],
    IMH1L → 0,
    IMH2L → 0.085,
    IMH3L → 0.085,
    (* TM stuff *)
    TMH1V → 0.16546 / 2,
    TMH2V → 0.16546 / 2,
    TMH3V → 0.16546 / 2,
    TMH4V → 0.16546 / 2,
    TMH1T → 0.16546 / 2,
    TMH2T → 0.16546 / 2,
    TMH3T → 0.16546 / 2,
    TMH4T → 0.16546 / 2
   };
```

TM Geometry

Updated for SR. Relative to BS, lots of signs are reversed, because the OSEMs are on the back side of the optic (was front for the BS), but the numbering is as viewed from the back in both cases.

- **The test matrix, which gives the change at each sensor for excursions in L, P and Y.**

In[157]:=
```
(TMtest = Transpose[{
      {1, 1, 1, 1},
      {TMH1V, -TMH2V, TMH3V, -TMH4V},
      {-TMH1T, -TMH2T, TMH3T, TMH4T}
     }] /. vals
 ) // TableForm
```

Out[157]//TableForm=

1	0.08273	−0.08273
1	−0.08273	−0.08273
1	0.08273	0.08273
1	−0.08273	0.08273

- **The diagonalization matrix, which attempts to undo the test matrix. This matrix should be typed as-is into the OSEM2EUL screen and transposed into the EUL2OSEM screen.**

In[158]:=
```
(TM = {
      0.25 * {1, 1, 1, 1},
      0.25 * {1 / TMH1V, -1 / TMH2V, 1 / TMH3V, -1 / TMH4V},
      0.25 * {-1 / TMH1T, -1 / TMH2T, 1 / TMH3T, 1 / TMH4T}
     } /. vals
 ) // TableForm
```

Out[158]//TableForm=

0.25	0.25	0.25	0.25
3.02188	−3.02188	3.02188	−3.02188
−3.02188	−3.02188	3.02188	3.02188

- **Multiplying the diagonalization matrix and the test matrix should give the identity matrix.**

In[159]:=
```
TM.TMtest // TableForm
```

Out[159]//TableForm=

1.	0.	0.
0.	1.	0.
0.	0.	1.

IM Geometry

Updated for SR. Relative to the BS there are no changes to the symbolic version, only different numeric values.

- **The test matrix, which gives the change at each sensor for excursions in L, T, V, R, P and Y.**

In[160]:=
```
(IMtest = Transpose[{
      {0, 0, 0, -1, 0, 0}, (* L *)
      {0, 0, 0, 0, -1, 1}, (* T *)
      {-1, -1, -1, 0, 0, 0}, (* V *)
      {0, -IMV2T, IMV3T, 0, 0, 0}, (* R *)
      {IMV1L, -IMV2L, -IMV3L, 0, 0, 0}, (* P *)
      {0, 0, 0, 0, -IMH2L, -IMH3L} (* Y *)
     }] /. vals // N
 ) // Transpose // TableForm
```

Out[160]//TableForm=

0.	0.	0.	−1.	0.	0.
0.	0.	0.	0.	−1.	1.
−1.	−1.	−1.	0.	0.	0.
0.	−0.0831384	0.0831384	0.	0.	0.
0.096	−0.048	−0.048	0.	0.	0.
0.	0.	0.	0.	−0.085	−0.085

- **The diagonalization matrix, which attempts to undo the test matrix. This matrix should be typed as-is into the OSEM2EUL screen and transposed into the EUL2OSEM screen.**

In[161]:=
```
(IM = {
       {0, 0, 0, -1, 0, 0}, (* L *)
       {0, 0, 0, 0, -1, 1} / 2, (* T *)
       {-1, -1, -1, 0, 0, 0} / 3, (* V *)
       {0, -1 / IMV2T, 1 / IMV3T, 0, 0, 0} / 2, (* R *)
       {4 / IMV1L, -1 / IMV2L, -1 / IMV3L, 0, 0, 0} / 6, (* P *)
       {0, 0, 0, 0, -1 / IMH2L, -1 / IMH3L} / 2 (* Y *)
      } /. vals // N
 ) // TableForm
```

Out[161]//TableForm=

0.	0.	0.	-1.	0.	0.
0.	0.	0.	0.	-0.5	0.5
-0.333333	-0.333333	-0.333333	0.	0.	0.
0.	-6.01407	6.01407	0.	0.	0.
6.94444	-3.47222	-3.47222	0.	0.	0.
0.	0.	0.	0.	-5.88235	-5.88235

- **Multiplying the diagonalization matrix and the test matrix should give the identity matrix.**

In[162]:=
```
IM.IMtest // TableForm
```

Out[162]//TableForm=

1.	0.	0.	0.	0.	0.
0.	1.	0.	0.	0.	0.
0.	0.	1.	0.	0.	0.
0.	0.	0.	1.	0.	0.
0.	0.	0.	0.	1.	0.
0.	0.	0.	0.	0.	1.

PI Geometry

Mostly updated for SR. The only big change from the BS was that the FR is colocated with the LVDT at +30 relative to the leg in the Nikhef IPs instead of -30° in the old Promec IP. The radius values are still as for BS - this is probably right for geophones and LVDTs but possibly not for the new colocated FRs -> CHECK.

Calculation

- **90° Rotation matrix for x/y -> L/T.**

In[163]:=
```
Ninety = RotationMatrix[90 °]
```

Out[163]=
```
{{0, -1}, {1, 0}}
```

- **Angular locations of the various items**

IP legs

In[164]:=
```
Iangles = {0 °, 120 °, 240 °};
```

Geophones

In[165]:=
```
Gangles = 15 ° + Iangles
```

Out[165]=
```
{15 °, 135 °, 255 °}
```

LVDTs

In[166]:=
```
Langles = 30 ° + Iangles
```

Out[166]=
```
{30 °, 150 °, 270 °}
```

Fishing rod steppers

In[167]:=
```
Fangles = 30 ° + Iangles (* was -30° for BS *)
```

Out[167]=
$$\{30\,^\circ, 150\,^\circ, 270\,^\circ\}$$

EQ stop positions

In[168]:=
```
Eangles = -60 ° + Iangles
```

Out[168]=
$$\{-60\,^\circ, 60\,^\circ, 180\,^\circ\}$$

- **Positions of the various items as vectors**

These are calculated by rotating the x axis {1,0} by the corresponding angles.

In[169]:=
```
Ipositions = Iradius * Table[RotationMatrix[Iangles[[i]]].{1, 0}, {i, 1, 3}]
```

Out[169]=
$$\left\{\{\text{Iradius}, 0\}, \left\{-\frac{\text{Iradius}}{2}, \frac{\sqrt{3}\ \text{Iradius}}{2}\right\}, \left\{-\frac{\text{Iradius}}{2}, -\frac{\sqrt{3}\ \text{Iradius}}{2}\right\}\right\}$$

In[170]:=
```
Fpositions = Fradius * Table[RotationMatrix[Fangles[[i]]].{1, 0}, {i, 1, 3}]
```

Out[170]=
$$\left\{\left\{\frac{\sqrt{3}\ \text{Fradius}}{2}, \frac{\text{Fradius}}{2}\right\}, \left\{-\frac{\sqrt{3}\ \text{Fradius}}{2}, \frac{\text{Fradius}}{2}\right\}, \{0, -\text{Fradius}\}\right\}$$

In[171]:=
$$\left\{\left\{\frac{\sqrt{3}\ \text{Fradius}}{2}, -\frac{\text{Fradius}}{2}\right\}, \{0, \text{Fradius}\}, \left\{-\frac{\sqrt{3}\ \text{Fradius}}{2}, -\frac{\text{Fradius}}{2}\right\}\right\}$$

Out[171]=
$$\left\{\left\{\frac{\sqrt{3}\ \text{Fradius}}{2}, -\frac{\text{Fradius}}{2}\right\}, \{0, \text{Fradius}\}, \left\{-\frac{\sqrt{3}\ \text{Fradius}}{2}, -\frac{\text{Fradius}}{2}\right\}\right\}$$

In[172]:=
```
Gpositions = Gradius * Table[RotationMatrix[Gangles[[i]]].{1, 0}, {i, 1, 3}]
```

Out[172]=
$$\left\{\left\{\frac{\left(1+\sqrt{3}\right)\ \text{Gradius}}{2\sqrt{2}}, \frac{\left(-1+\sqrt{3}\right)\ \text{Gradius}}{2\sqrt{2}}\right\},\right.$$
$$\left.\left\{-\frac{\text{Gradius}}{\sqrt{2}}, \frac{\text{Gradius}}{\sqrt{2}}\right\}, \left\{-\frac{\left(-1+\sqrt{3}\right)\ \text{Gradius}}{2\sqrt{2}}, -\frac{\left(1+\sqrt{3}\right)\ \text{Gradius}}{2\sqrt{2}}\right\}\right\}$$

In[173]:=
```
Lpositions = Lradius * Table[RotationMatrix[Langles[[i]]].{1, 0}, {i, 1, 3}]
```

Out[173]=
$$\left\{\left\{\frac{\sqrt{3}\ \text{Lradius}}{2}, \frac{\text{Lradius}}{2}\right\}, \left\{-\frac{\sqrt{3}\ \text{Lradius}}{2}, \frac{\text{Lradius}}{2}\right\}, \{0, -\text{Lradius}\}\right\}$$

In[174]:=
```
Epositions = Eradius * Table[RotationMatrix[Eangles[[i]]].{1, 0}, {i, 1, 3}]
```

Out[174]=
$$\left\{\left\{\frac{\text{Eradius}}{2}, -\frac{\sqrt{3}\ \text{Eradius}}{2}\right\}, \left\{\frac{\text{Eradius}}{2}, \frac{\sqrt{3}\ \text{Eradius}}{2}\right\}, \{-\text{Eradius}, 0\}\right\}$$

- **Orientations of the various items as unit vectors**

These are calculated by rotating a unit y vector {0,1} (i.e., the tangent vector at {1,0}) by the same set of angles.

In[175]:=
```
Fvectors = Table[RotationMatrix[Fangles[[i]]].{0, 1}, {i, 1, 3}]
```

Out[175]=
$$\left\{\left\{-\frac{1}{2}, \frac{\sqrt{3}}{2}\right\}, \left\{-\frac{1}{2}, -\frac{\sqrt{3}}{2}\right\}, \{1, 0\}\right\}$$

In[176]:=
```
Gvectors = Table[RotationMatrix[Gangles[[i]]].{0, 1}, {i, 1, 3}]
```

Out[176]=
$$\left\{\left\{-\frac{-1+\sqrt{3}}{2\sqrt{2}}, \frac{1+\sqrt{3}}{2\sqrt{2}}\right\}, \left\{-\frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}}\right\}, \left\{\frac{1+\sqrt{3}}{2\sqrt{2}}, -\frac{-1+\sqrt{3}}{2\sqrt{2}}\right\}\right\}$$

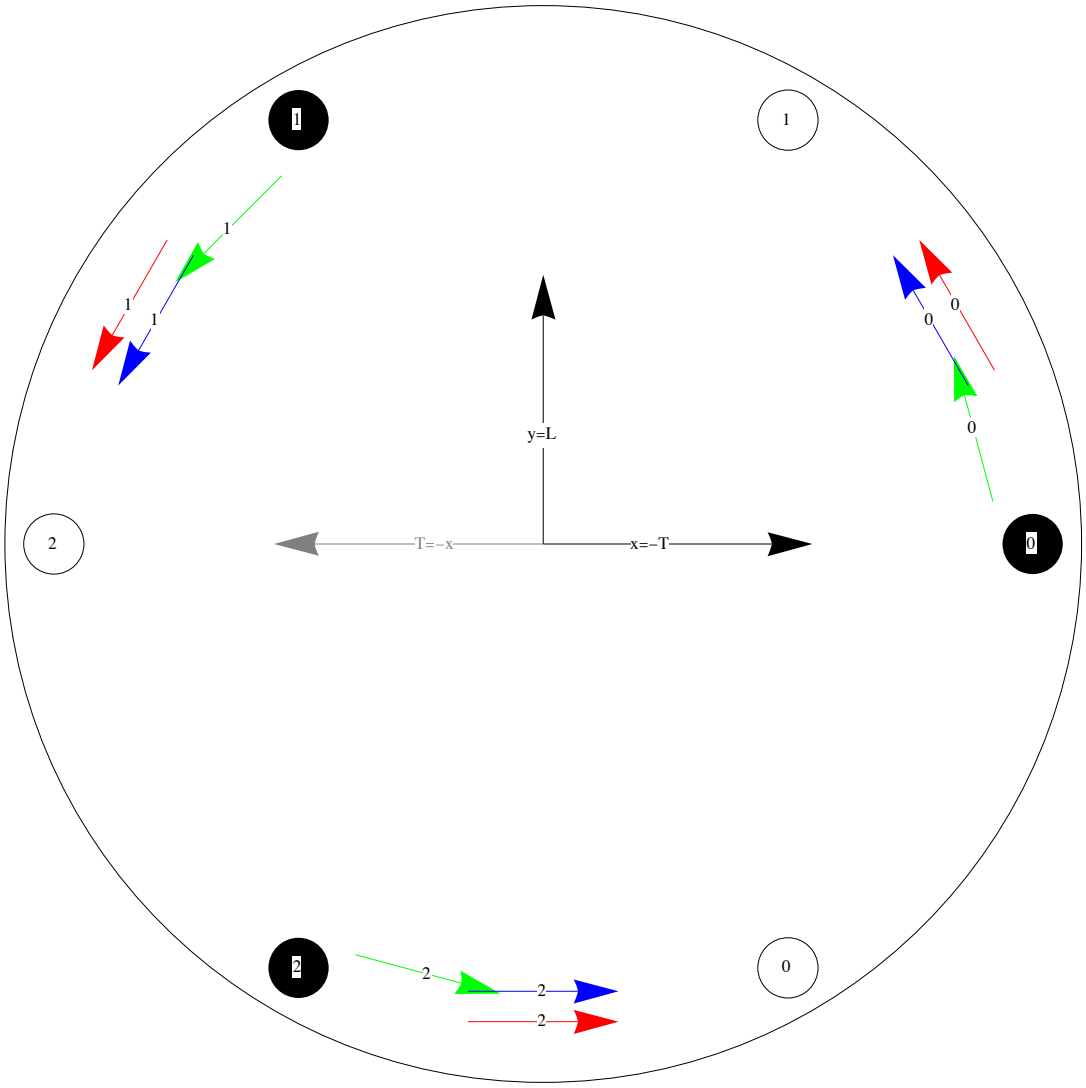
In[177]:=
```
Lvectors = Table[RotationMatrix[Langles[[i]]].{0, 1}, {i, 1, 3}]
```

Out[177]=
$$\left\{\left\{-\frac{1}{2}, \frac{\sqrt{3}}{2}\right\}, \left\{-\frac{1}{2}, -\frac{\sqrt{3}}{2}\right\}, \{1, 0\}\right\}$$

■ Diagram

In[178]:=
```
Graphics[{
  Circle[{0, 0}, R],
  Sequence[Table[Disk[Ipositions[[i]], Iradius2],
    {i, 1, 3}
   ]],
  Sequence[Table[Text[i - 1, Ipositions[[i]], Background → White], {i, 1, 3}]],
  Sequence[Table[Circle[Epositions[[i]], Eradius2],
    {i, 1, 3}
   ]],
  Sequence[Table[Text[i - 1, Epositions[[i]], Background → White], {i, 1, 3}]],
  Red,
  Sequence[Table[Arrow[{Fpositions[[i]] - Fvectors[[i]] * l / 2,
      Fpositions[[i]] + Fvectors[[i]] * l / 2}], {i, 1, 3}
   ]],
  Black, Sequence[Table[Text[i - 1, Fpositions[[i]], Background → White], {i, 1, 3}]],
  Green,
  Sequence[Table[Arrow[{Gpositions[[i]] - Gvectors[[i]] * l / 2,
      Gpositions[[i]] + Gvectors[[i]] * l / 2}], {i, 1, 3}
   ]],
  Black, Sequence[Table[Text[i - 1, Gpositions[[i]], Background → White], {i, 1, 3}]],
  Blue,
  Sequence[Table[Arrow[{Lpositions[[i]] - Lvectors[[i]] * l / 2,
      Lpositions[[i]] + Lvectors[[i]] * l / 2}], {i, 1, 3}
   ]],
  Black, Sequence[Table[Text[i - 1, Lpositions[[i]], Background → White], {i, 1, 3}]],
  Arrow[{{0, 0}, {R / 2, 0}}], Text["x=-T", {R / 5, 0}, Background → White],
  Arrow[{{0, 0}, {0, R / 2}}], Text["y=L", {0, R / 5}, Background → White],
  Gray,
  Arrow[{{0, 0}, {-R / 2, 0}}], Text["T=-x", {-R / 5, 0}, Background → White]
 }] /. vals
```

Out[178]=



In[179]:=

■ Matrices between L/T/Y and LVDTs

Usage: {L0, L1, L2}= LVDTfromLTY.{L, T, Y}

In[180]:=
```
(LVDTfromLTY = {
    Table[Lvectors[[i]].Ninety.{1, 0}, {i, 1, 3}],
    Table[Lvectors[[i]].Ninety.{0, 1}, {i, 1, 3}],
    Table[Lradius, {i, 1, 3}]
  }) // TableForm
```

Out[180]//TableForm=

$\frac{\sqrt{3}}{2}$	$-\frac{\sqrt{3}}{2}$	0
$\frac{1}{2}$	$\frac{1}{2}$	−1
Lradius	Lradius	Lradius

Usage: type these numbers into EUL2COIL

In[181]:=
```
Transpose[LVDTfromLTY] /. vals // N // TableForm
```

Out[181]//TableForm=

0.866025	0.5	0.594
−0.866025	0.5	0.594
0.	−1.	0.594

Usage: {L, T, Y}= LTYfromLVDT.{L0, L1, L2}

In[182]:=
```
(LTYfromLVDT = Inverse[LVDTfromLTY]) // TableForm
```

Out[182]//TableForm=

$\frac{1}{\sqrt{3}}$	$\frac{1}{3}$	$\frac{1}{3\,\text{Lradius}}$
$-\frac{1}{\sqrt{3}}$	$\frac{1}{3}$	$\frac{1}{3\,\text{Lradius}}$
0	$-\frac{2}{3}$	$\frac{1}{3\,\text{Lradius}}$

Usage: type these numbers into LVDT2EUL

In[183]:=
```
Transpose[LTYfromLVDT] /. vals // N // TableForm
```

Out[183]//TableForm=

0.57735	−0.57735	0.
0.333333	0.333333	−0.666667
0.561167	0.561167	0.561167

■ Matrices between L/T/Y and Geophones

Usage: {G0, G1, G2}= GfromLTY.{L, T, Y} (assuming the geophones had actuation)

In[184]:=
```
(GfromLTY = {
    Table[Gvectors[[i]].Ninety.{1, 0}, {i, 1, 3}],
    Table[Gvectors[[i]].Ninety.{0, 1}, {i, 1, 3}],
    Table[Gradius, {i, 1, 3}]
   }) // TableForm
```

Out[184]//TableForm=

$\frac{1+\sqrt{3}}{2\sqrt{2}}$	$-\frac{1}{\sqrt{2}}$	$-\frac{-1+\sqrt{3}}{2\sqrt{2}}$
$\frac{-1+\sqrt{3}}{2\sqrt{2}}$	$\frac{1}{\sqrt{2}}$	$-\frac{1+\sqrt{3}}{2\sqrt{2}}$
Gradius	Gradius	Gradius

Usage: you would type these numbers into EUL2ACC if it existed, which it doesn't because geophones don't have actuation.

In[185]:=
```
Transpose[GfromLTY] /. vals // N // TableForm
```

Out[185]//TableForm=

0.965926	0.258819	0.5915
−0.707107	0.707107	0.5915
−0.258819	−0.965926	0.5915

Usage: {L, T, Y}= LTYfromG.{G0, G1, G2}

In[186]:=
```
(LTYfromG = Inverse[GfromLTY] // Simplify) // TableForm
```

Out[186]//TableForm=

$\frac{3+\sqrt{3}}{3\sqrt{6}}$	$\frac{-1+\sqrt{3}}{3\sqrt{2}}$	$\frac{1}{3\,\text{Gradius}}$
$-\frac{\sqrt{2}}{3}$	$\frac{\sqrt{2}}{3}$	$\frac{1}{3\,\text{Gradius}}$
$\frac{-3+\sqrt{3}}{3\sqrt{6}}$	$-\frac{3+\sqrt{3}}{3\sqrt{6}}$	$\frac{1}{3\,\text{Gradius}}$

Usage: type these numbers into ACC2EUL

In[187]:=
```
Transpose[LTYfromG] /. vals // N // TableForm
```

Out[187]//TableForm=

0.643951	−0.471405	−0.172546
0.172546	0.471405	−0.643951
0.563539	0.563539	0.563539

■ Matrices between L/T/Y and FRs

Usage: {F0, F1, F2}= FfromLTY.{L, T, Y}

In[188]:=
```
(FfromLTY = {
    Table[Fvectors[[i]].Ninety.{1, 0}, {i, 1, 3}],
    Table[Fvectors[[i]].Ninety.{0, 1}, {i, 1, 3}],
    Table[Fradius, {i, 1, 3}]
   }) // TableForm
```

Out[188]//TableForm=

$\frac{\sqrt{3}}{2}$	$-\frac{\sqrt{3}}{2}$	0
$\frac{1}{2}$	$\frac{1}{2}$	−1
Fradius	Fradius	Fradius

In[189]:=

$\frac{\sqrt{3}}{2}$	0	$-\frac{\sqrt{3}}{2}$
$-\frac{1}{2}$	1	$-\frac{1}{2}$
Fradius	Fradius	Fradius

Out[189]= $\left\{\left\{\frac{\sqrt{3}}{2}, 0, -\frac{\sqrt{3}}{2}\right\}, \left\{-\frac{1}{2}, 1, -\frac{1}{2}\right\}, \{\text{Fradius}, \text{Fradius}, \text{Fradius}\}\right\}$

Usage: write a Python script to take DC force/torque requests in LTY coordinates, multiply by this matrix and output stepper motor movements in steps.

In[190]:=
```
FfromLTY /. vals // N // TableForm
```

Out[190]//TableForm=

0.866025	−0.866025	0.
0.5	0.5	−1.
0.6344	0.6344	0.6344

Usage: {L, T, Y}= LTYfromF.{F0, F1, F2}

In[191]:=
```
(LTYfromF = Inverse[FfromLTY] // Simplify) // TableForm
```

Out[191]//TableForm=

$\frac{1}{\sqrt{3}}$	$\frac{1}{3}$	$\frac{1}{3\,\text{Fradius}}$
$-\frac{1}{\sqrt{3}}$	$\frac{1}{3}$	$\frac{1}{3\,\text{Fradius}}$
0	$-\frac{2}{3}$	$\frac{1}{3\,\text{Fradius}}$

In[192]:=

$\frac{1}{\sqrt{3}}$	$-\frac{1}{3}$	$\frac{1}{3\,\text{Fradius}}$
0	$\frac{2}{3}$	$\frac{1}{3\,\text{Fradius}}$
$-\frac{1}{\sqrt{3}}$	$-\frac{1}{3}$	$\frac{1}{3\,\text{Fradius}}$

Out[192]= $\left\{\left\{\frac{1}{\sqrt{3}}, -\frac{1}{3}, \frac{1}{3\,\text{Fradius}}\right\}, \left\{0, \frac{2}{3}, \frac{1}{3\,\text{Fradius}}\right\}, \left\{-\frac{1}{\sqrt{3}}, -\frac{1}{3}, \frac{1}{3\,\text{Fradius}}\right\}\right\}$

Usage: you would type these numbers into FR2EUL, if it existed, which it doesn't because FRs don't have sensing.

In[193]:=
```
Transpose[LTYfromF] /. vals // N // TableForm
```

Out[193]//TableForm=

0.57735	−0.57735	0.
0.333333	0.333333	−0.666667
0.525431	0.525431	0.525431