The recalculation of the two-wire simple model with blades - V3.0- LOS values

■ Switches, version numbers, paths etc

Switches to enable loading of previously saved results instead of recalculating from scratch

```
useprecomputed = False; (* set to True to use saved results from precomputed
subdirectory *)
If[useprecomputed,
    exceptdamping = False, (* False by default, True to recalculate just
damping-dependent stuff *)
    exceptdamping = True (* DON"T CHANGE *)
];
```

Definition of the current case - label, descriptive comment, and specification in terms of overrides relative to the default parameters
**** EDIT THIS SECTION TO DEFINE A NEW CASE ****

```
modelcase = "TT";

modelcasecomment = "LIGO-I ETM, SOS with blades creating the Undulators....";

overrides = {
    tx -> 0.0127,              (* thickness of optic, 1inch *)
    tr -> (68.2*10^-3)/2,     (* radius of optic, 3inch diameter *)
    m0 -> 124*10^-3,          (* mass of the compound 2" mirror and holder *)
    l0 -> 0.140,               (* wire length *)
    r0 -> (127*10^-6)/2,      (* radius of wire, not critical can be adjusted
                    available diameters: 101um, 113um, 127um, 143um, 160um,
                                 180um, 202um, 310um, 355um, 370um *)

    dpitch -> 1.5*10^-3,        (* height of wire break-off above c.o.m. *)
    dyaw1 -> 35*10^-3,        (* y-separation of wires at structure *)
    dyaw2 -> 76.2*10^-3,  (* y-separation of wires at optic,
                               optic diameter + 2*1mm standoffs *)
    kbz -> 75,                (* per-blade elastic constant in N/m *)
    I0y -> 48.6*10^-6,        (* moment of inertia (longitudinal, pitch) *)
    I0z -> 50.9*10^-6,         (* moment of inertia (yaw) *)
    I0x -> 91.5*10^-6           (* moment of inertia (roll) *)
};

nonoptvars = {

};
```

The path in which the core model definition and model-specific support files are installed.
**** EDIT THE PATHS AND/OR ADD CLAUSES FOR NEW SYSTEMS AS APPROPRIATE ****

The path in which custom packages are installed (if different from above).
**** EDIT THE PATHS AND/OR ADD CLAUSES FOR NEW SYSTEMS AS APPROPRIATE ****

Enable the mode shape plots

Enable export of state space to Matlab

The number of interesting low frequency modes

Usage notes for switches

■ Version history

■ Instructions

■ Preliminaries

The directory in which this copy of ASUSModelCalc.nb and the other files associated with this case are stored.

```
SetDirectory[ToFileName[{modeldirectory, modelcase}]];
```

The path in which the model definition and model-specific support files are installed.

The path in which custom packages are installed (if different from above).

```
$Path = Append[$Path, modelsupportdirectory];
```

Read in the model definition. First look for a customized model in the current directory, then look for the factory model in the directory above.

```
Get["TwoWireSimpleBladesDefn.m",Path->{Directory[],ParentDirectory[Directory[]]}];
```

Check that core model is loaded and correct

Archive some stuff that doesn't get recomputed here, just in case it's useful to look at

Open the status window

```
OpenStatusWindow[WindowSize->{500,300}];
```

```
Reset[All]
```

■ Listings and plots with Stage 0A Data

```
Calculate[Stage0A]
```

View the eigenfrequencies

```
Table[{Length[allvars]-index+1, Hz0A[[index]]}, {index, 1, Length[allvars]}]
```

```
{{6, 7.74666}, {5, 5.52132}, {4, 1.70726}, {3, 1.34762}, {2, 1.33201}, {1, 0.969375}}
```

View the important low-frequency modes

```
Join[
    {{"N","f","type"}},
    Table[
        Join[
            {i},
            {Hz0A[[-i]]},
            important[e2ni.eigenvectors0A[[-i]],0.7]
        ],
        {i,noofLF}
    ]
]//TableForm
```

N	f	type
1	0.969375	pitch0
2	1.33201	roll0
3	1.34762	pitch0
4	1.70726	yaw0
5	5.52132	z0
6	7.74666	roll0

View the higher frequency modes

Classify the modes

View the mode frequencies by groups

```
longpitchfreqs0 = Hz0A[[longpitchmodes0]]
```

```
{1.34762, 0.969375}
```

```
transrollfreqs0 = Hz0A[[transrollmodes0]]
```

```
{7.74666, 1.33201}
```

```
yawfreqs0 = Hz0A[[yawmodes0]]
```

```
{1.70726}
```

```
vertfreqs0 = Hz0A[[vertmodes0]]
```

```
{5.52132}
```

Export stuff for Matlab comparison

Mode shape listings/plots

#1

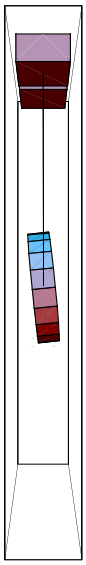
```
Hz0A[[-1]]
```

```
0.969375
```

```
pretty[Chop[e2ni.eigenvectors0A[[-1]], 10^-4]]
```

```
          x             y      z      yaw      pitch          roll
optic     0.00314899    0      0      0        -0.999995      0
```

```
DoWithStatus["Plotting stage 0A mode 1",
 eigenplot[eigenvectors0A[[-1]], .1, {0, -1, 0}, floatmatrix0A]]
```



```
- Graphics3D -
```

#2

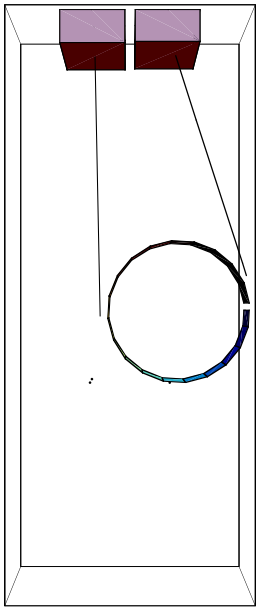
```
Hz0A[[-2]]
```

```
1.33201
```

```
pretty[Chop[e2ni.eigenvectors0A[[-2]], 10^-4]]
```

```
          x      y             z      yaw      pitch      roll
optic     0      -0.233751     0      0        0          -0.972297
```

```
DoWithStatus["Plotting stage 0A mode 2",
 eigenplot[eigenvectors0A[[-2]], .1, {-1, 0, 0}, floatmatrix0A]]
(*Status["Plotting stage 0A mode 2"] ;
 eigenplot[eigenvectors0A[[-2]],.1,{-1,0,0}, floatmatrix0A]  ;Done[]*)
```



```
- Graphics3D -
```

#3

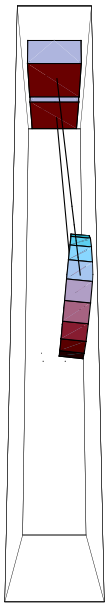
```
Hz0A[[-3]]
```

```
1.34762
```

```
pretty[Chop[e2ni.eigenvectors0A[[-3]], 10^-4]]
```

	x	y	z	yaw	pitch	roll
optic	0.12351	0	0	0	0.992343	0

```
Status["Plotting stage 0A mode 3"] ;
eigenplot[eigenvectors0A[[-3]], .1, {0, -1, -.25}, floatmatrix0A] ; Done[]
```



#4

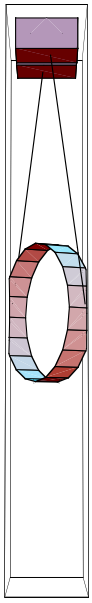
Hz0A[[-4]]

1.70726

pretty[Chop[e2ni.eigenvectors0A[[-4]], 10^-4]]

	x	y	z	yaw	pitch	roll
optic	0	0	0	1.	0	0

Status["Plotting stage 0A mode 4"] ;
eigenplot[eigenvectors0A[[-4]], .5, {0, -3, -.25}, floatmatrix0A] ; Done[]



#5

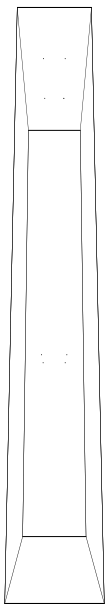
Hz0A[[-5]]

5.52132

pretty[Chop[e2ni.eigenvectors0A[[-5]], 10^-4]]

	x	y	z	yaw	pitch	roll
optic	0	0	-1.	0	0	0

Status["Plotting stage 0A mode 5"] ;
eigenplot[eigenvectors0A[[-5]], -.2, {0, -1, -.25}, floatmatrix0A] ; Done[]



#6

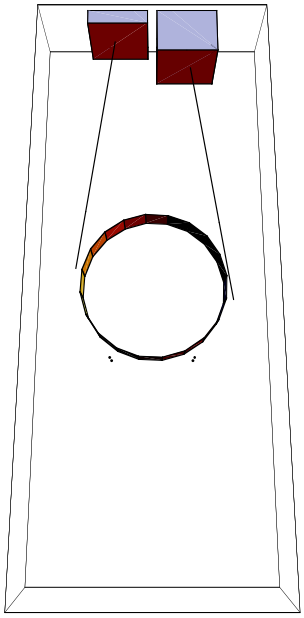
```
Hz0A[[-6]]
```

```
7.74666
```

```
pretty[Chop[e2ni.eigenvectors0A[[-6]], 10^-4]]
```

	x	y	z	yaw	pitch	roll
optic	0	0.00306932	0	0	0	-0.999995

```
Status["Plotting stage 0A mode 6"] ;
eigenplot[eigenvectors0A[[-6]], -.2, {-1, 0, -.25}, floatmatrix0A] ; Done[]
```



■ Listings and plots with Stage 2 Data

```
Calculate[Stage2]
```

View the eigenfrequencies

```
Table[{Length[allvars]-index+1, Hz2[[index]]}, {index, 1, Length[allvars]}]
```

```
{{6, 7.79005}, {5, 5.52132}, {4, 1.73746}, {3, 1.53387}, {2, 1.34264}, {1, 1.28626}}
```

View the important low-frequency modes

```
Join[
    {{"N","f","type"}},
    Table[
        Join[
            {i},
            {Hz2[[-i]]},
            important[e2ni.eigenvectors2[[-i]],0.7]
        ],
        {i,noofLF}
    ]
]//TableForm
```

N	f	type
1	1.28626	pitch0
2	1.34264	roll0
3	1.53387	pitch0
4	1.73746	yaw0
5	5.52132	z0
6	7.79005	roll0

View the higher frequency modes

```
Join[
    {{"N","f","type"}},
    Table[
        Join[
            {i},
            {Hz2[[-i]]},
            important[e2ni.eigenvectors2[[-i]],0.7]
        ],
        {i,noofLF+1,Length[allvars]}
    ]
]//TableForm
```

```
N      f      type
```

Classify the modes

View the mode frequencies by groups

Mode shape listings/plots

#1

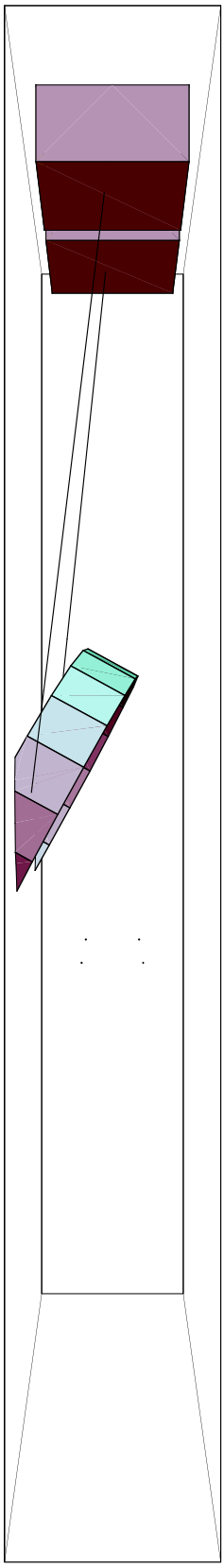
```
Hz2[[-1]]
```

```
1.28626
```

```
pretty[Chop[e2ni.eigenvectors2[[-1]], 10^-4]]
```

```
          x              y      z      yaw      pitch         roll
optic     -0.0325117     0      0      0        0.999471      0
```

```
Status["Plotting stage 2 mode 1"] ;
eigenplot[eigenvectors2[[-1]], .5, {0, -1, 0}, floatmatrix2] ; Done[]
```



#2

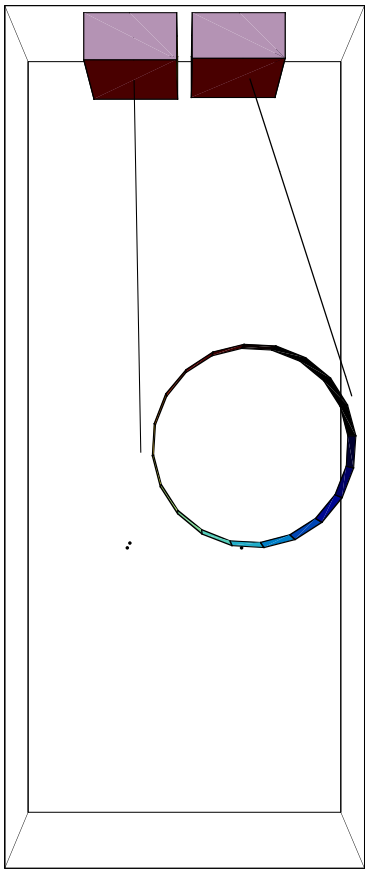
```
Hz2[[-2]]
```

```
1.34264
```

```
pretty[Chop[e2ni.eigenvectors2[[-2]], 10^-4]]
```

	x	y	z	yaw	pitch	roll
optic	0	-0.231778	0	0	0	-0.972769

```
Status["Plotting stage 2 mode 2"];
eigenplot[eigenvectors2[[-2]], .1, {-1, 0, 0}, floatmatrix2]; Done[]
```



#3

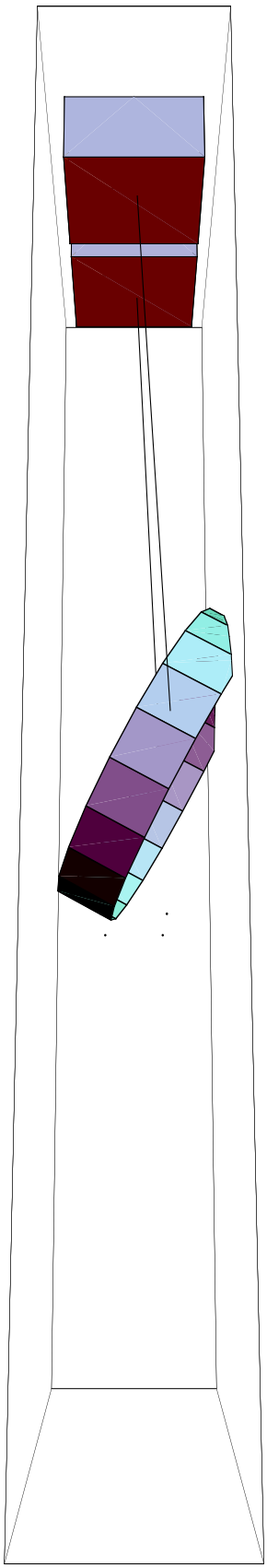
```
Hz2[[-3]]
```

1.53387

```
pretty[Chop[e2ni.eigenvectors2[[-3]], 10^-4]]
```

	x	y	z	yaw	pitch	roll
optic	0.012048	0	0	0	0.999927	0

```
Status["Plotting stage 2 mode 2"] ;
eigenplot[eigenvectors2[[-3]], .5, {0, -1, -.25}, floatmatrix2] ; Done[]
```



#4

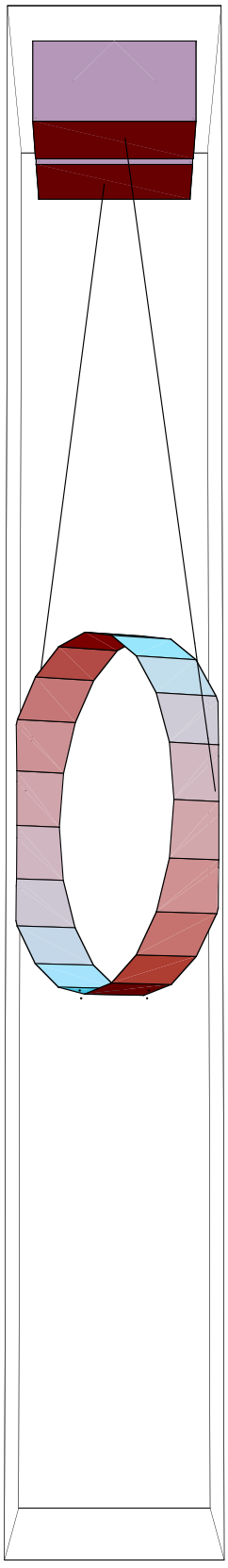
```
Hz2[[-4]]
```

```
1.73746
```

```
pretty[Chop[e2ni.eigenvectors2[[-4]], 10^-4]]
```

	x	y	z	yaw	pitch	roll
optic	0	0	0	1.	0	0

```
Status["Plotting stage 2 mode 4"];
eigenplot[eigenvectors2[[-4]], .5, {0, -3, -.25}, floatmatrix2]; Done[]
```



#5

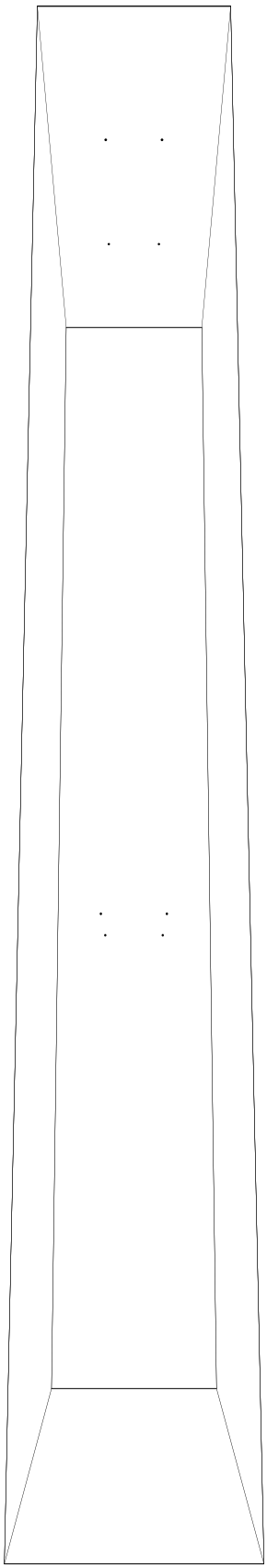
```
Hz2[[-5]]
```

```
5.52132
```

```
pretty[Chop[e2ni.eigenvectors2[[-5]], 10^-4]]
```

	x	y	z	yaw	pitch	roll
optic	0	0	-1.	0	0	0

```
Status["Plotting stage 2 mode 5"] ;
eigenplot[eigenvectors2[[-5]], -.2, {0, -1, -.25}, floatmatrix2] ; Done[]
```



#6

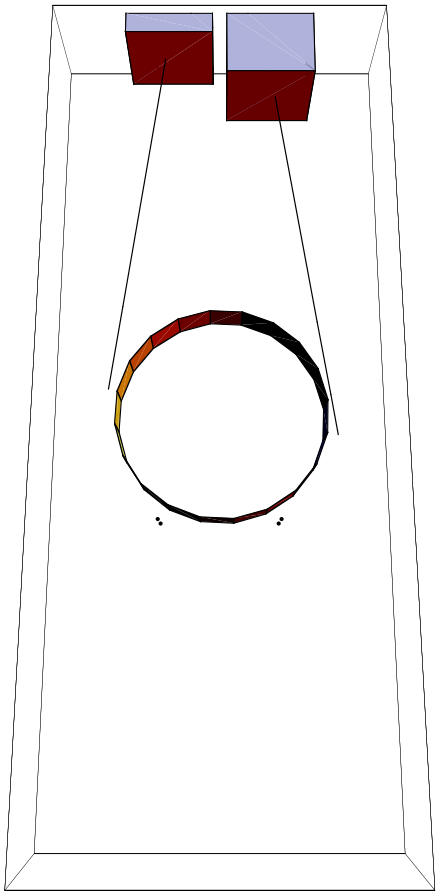
```
Hz2[[-6]]
```

```
7.79005
```

```
pretty[Chop[e2ni.eigenvectors2[[-6]], 10^-4]]
```

	x	y	z	yaw	pitch	roll
optic	0	0.00309696	0	0	0	-0.999995

```
Status["Plotting stage 2 mode 6"];
eigenplot[eigenvectors2[[-6]], -.2, {-1, 0, -.25}, floatmatrix2]; Done[]
```



Export as a state-space to Matlab/Simulink

```
Calculate[Stage2]

domatlabexport = True;

If[domatlabexport,
    MatlabExport[
        "twowiresimpleblades_"<>modelcase<>".m",
        {"mbtwsbA",ssAspoiler[potentialmatrices2,kineticmatrix,1.0,0]},
        {"mbtwsbB",ssB[coupling2,kineticmatrix,1.0]},
        {"mbtwsbC",ssC},
        {"mbtwsbD",ssD}
    ]
]
```