Thermal Modal Analysis with Static Correction: an efficient tool to model and design thermal compensation systems

Nicolas JOBERT

Synchrotron SOLEIL
Accelerators and Engineering Division

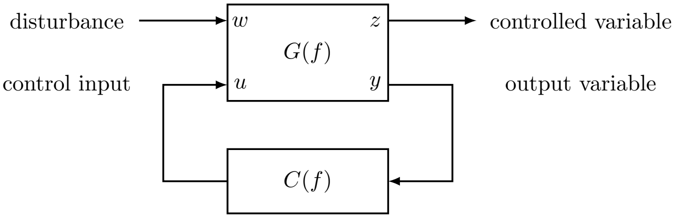
EUSPEN - SIG Meeting on Thermal Effects
Feb 26, 2020 Aachen

Outline

Outline

Objective (1/2)

Models in the context of thermal compensation systems



Objectives:

- minimize closed-loop response $T_{zw} = [G_{zw} + G_{zu}C(I - G_{yu}C)^{-1}G_{yw}]$
- guarantee stability

Need to know $G_{zw}, G_{yw}, G_{zu}, G_{yu}$

Objective (2/2)

What do we expect from a thermal model

A "good" model for thermal/thermal-elastic problems should be:

1. computationally efficient (fast and accurate)
2. compact in size (lightweight)
3. physically meaningfull (supports engineering judgment)

Modal decomposition only partially fullfils those requirements.

Outline

Thermal Response - Direct approach

Starting from conductivity [**K**] and capacity [**C**] matrices:

In time domain:

$$[\mathbf{C}]\dot{T} + [\mathbf{K}]T = P \qquad (1)$$

In frequency domain:

$$(j\omega[\mathbf{C}] + [\mathbf{K}])T = P \qquad (2)$$

Hence

$$\mathbf{G}(\omega) = (\mathbf{j}\omega[\mathbf{C}] + [\mathbf{K}])^{-\mathbf{1}} \qquad (3)$$

Most accurate numerically, but computationally demanding. Completely extensive but not really informative. Not suited for control loop design.

SOLEIL SYNCHROTRON

Thermal Response - Modal approach

Thanks to symetry, we can solve for the modes:

$$([\mathbf{C}] + \tau_i[\mathbf{K}])\Phi_i = 0 \tag{4}$$

Then the system "thermal compliance" reads:

$$G_{kl}(\omega) = \sum_{i=1}^{n_{dof}} \frac{\Phi_{ki}\Phi_{il}}{1 + j\omega\tau_i} \tag{5}$$

As accurate as direct method but only if all modes are extracted. Not feasible nor necessary in practice.
No clear-cut criterion to accept/reject truncated model.

Modal truncation error: simplification

Retaining only the first n_m modes $(n_m < n_{dof})$.

$$G_{kl}(\omega) = \sum_{i=1}^{n_m} \frac{\Phi_{ki}\Phi_{il}}{1 + j\omega\tau_i} + \sum_{i=n_m+1}^{n_{dof}} \frac{\Phi_{ki}\Phi_{il}}{1 + j\omega\tau_i} \tag{6}$$

Let ω_b be the bandwidth of the controller to be designed. Including all modes with $\tau_i >> 1/\omega_b$, the truncation error can be approximated as a *frequency independent* term.

$$R_{kl}(\omega) = \sum_{i=n_m+1}^{n_{dof}} \frac{\Phi_{ki}\Phi_{il}}{1 + j\omega\tau_i} \simeq \sum_{i=n_m+1}^{n_{dof}} \Phi_{ki}\Phi_{il} \tag{7}$$

Modal truncation error: estimation

Rewriting compliance in the static domain ($\omega = 0$)

$$G_{kl}(0) = \sum_{i=1}^{n_m} \Phi_{ki}\Phi_{il} + \sum_{i=n_m+1}^{n_{dof}} \Phi_{ki}\Phi_{il} \tag{8}$$

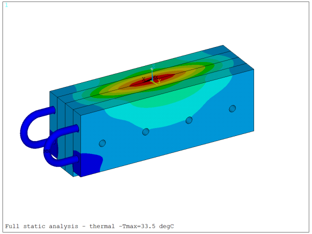
So that

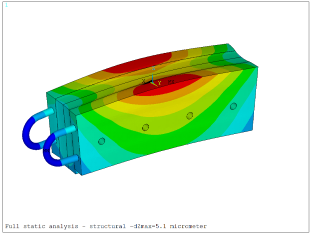
$$R_{kl} = G_{kl}(0) - \sum_{i=1}^{n_m} \Phi_{ki}\Phi_{il} \tag{9}$$

This term cand be added to "thermal compliance" so as to compensate for the "thermal flexibility" of discarded modes.

Outline

Beamline Primary Mirror





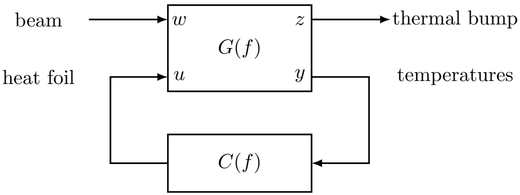
160x25mm^2 optical surface
SiC / Water cooled
heat load: 400W
drift rate: 1%/s
allowable slope error: 1μrad

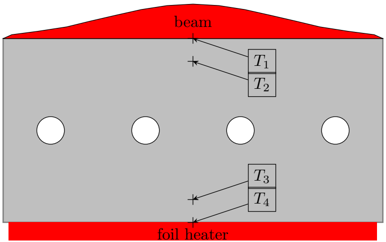
Technical Procedure:

1. Build plant thermal-elastic state space model (ANSYS / APDL Math)
2. Shape controller using LTI models (Matlab)
3. Build prototype
4. Validate optical performance (HASO at SOLEIL Optical Metrology Lab)

SOLEIL SYNCHROTRON

Controller Architecture

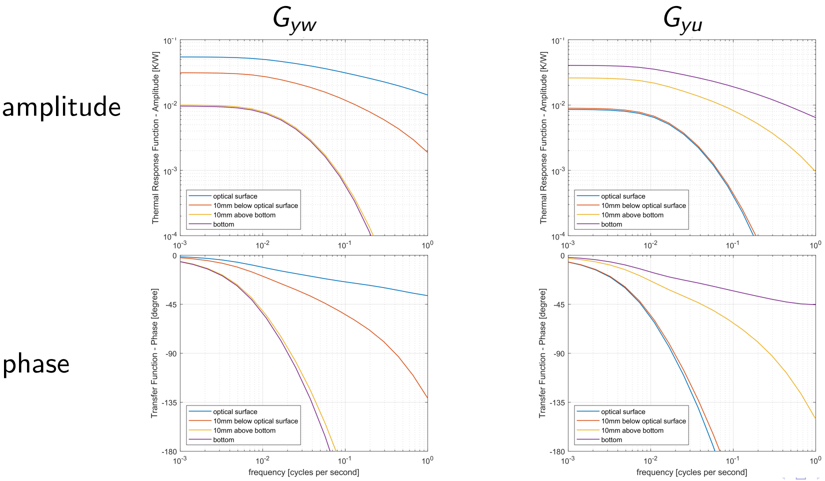




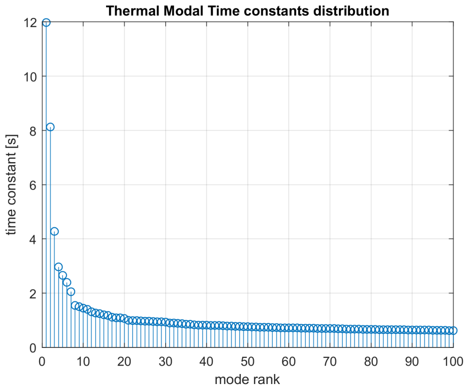
T_1	On upper surface
T_2	10mm below upper surface
T_3	10mm above lower surface
T_4	On lower surface

Ideal Case: $y = T_1 - T_4$
Real Case: $y = T_2 - T_3$
Performance: controlled by G_{yw}
Stability: controlled by G_{yu}

Plant Response - Reference Results Using Full Method



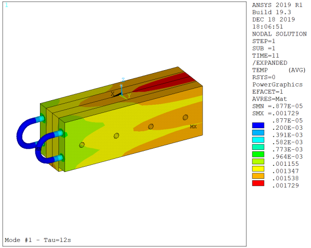
Results (1/5): Thermal Time Constants

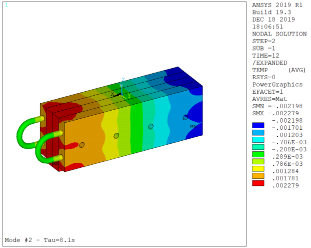


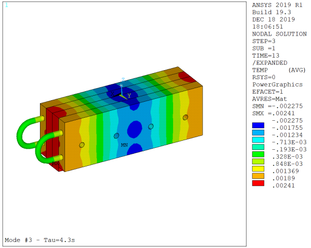
Desired bandwidth 10^{-1} *Hz*

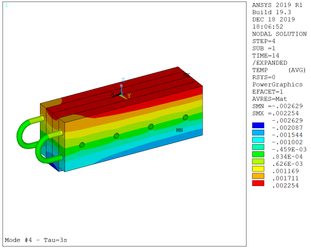
Let's retain modes with $\tau > 1.6s$

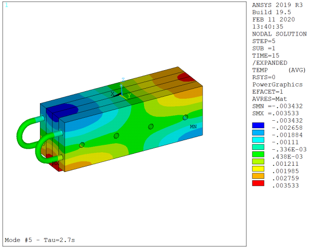
Results (2/5) : First 6 Thermal Mode Shapes

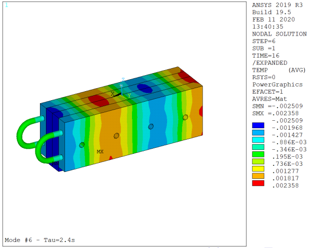








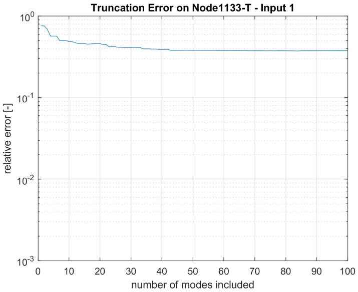




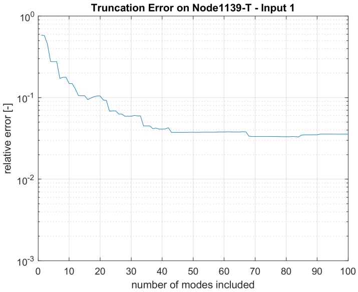
Results (3/5): Modal Method Convergence Rate

Static responses close to beam heat load

T_1



T_2



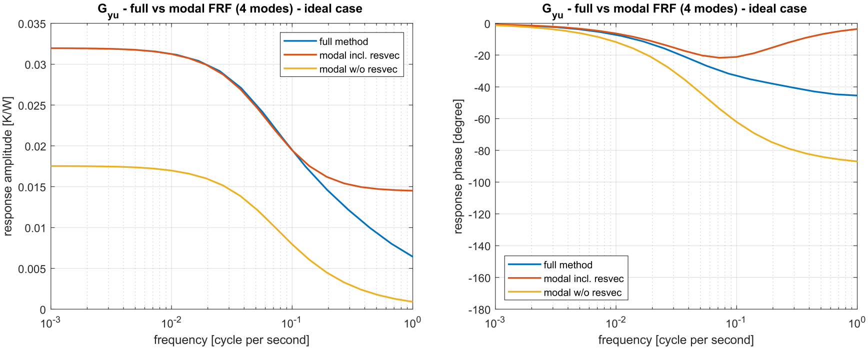
Close to heat load convergence is extremely slow: over 30% error with 100 modes included

Away from heat load convergence is faster

Brute force cannot be employed.

Results (4/5): Frequency Responses Compared

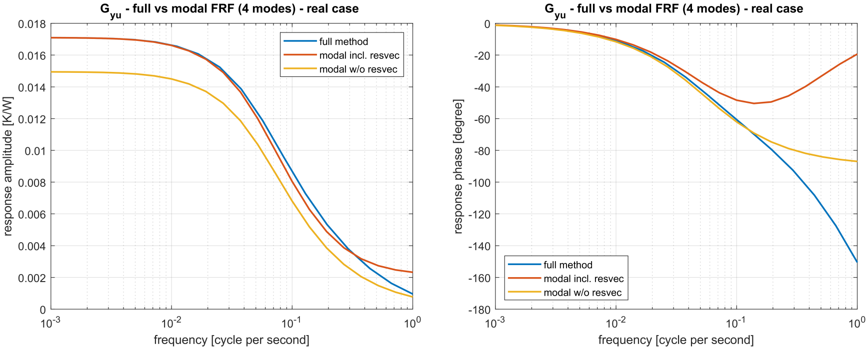
Direct vs Modal FRFs - Thermal probes at IDEAL positions



With residual vector, modal method yield exact results ($\pm 1dB$).
Above cutoff frequency phase is somewhat overpredicted, though

Results (5/5): Frequency Responses Compared

Direct vs Modal FRFs - Thermal probes at REAL positions



Again, modal method yield almost exact results

Above cutoff frequency phase becomes largely **overestimated**

Outline

Summary

Applicability and Benefits

For cases where *localized* thermal loads exist:

1. high spatial frequency thermal modes are excited
2. this requires inclusion of a very large number of modes to capture the local response
3. however, most of those modes have short time constants, hence respond quasi-statically
4. they can be lumped into a single, additionnal contribution, directly proportionnal to input, i.e. a residual vector
5. the modal basis can then be restricted to those modes that respond dynamically

In terms of state-space representation, this amounts to adding a feedthrough term, all other aspects remain unchanged

SOLEIL SYNCHROTRON

Outlook

Possible evolutions

Functionnality:

- Performance: Gain can be accurately obtained, even with a small number of modes, so that controller reduction estimates will be reliable
- Stability: phase is overpredicted at higher frequencies, hence stability *cannot* be guaranteed. This could be solved by replacing residual *vectors* by residual *modes*

Usability:

- Protoyping completed : APDL Math procedures perform Modal Analysis, Frequency Response, Residual Vector, State-Space Model, etc
- Next step: Encapsulate as an add-on ("ACT App")

Questions ? Comments?

Outline

Thermal Modal Analysis

```
/SOLU
ANTYPE,MODAL,NEW ! Modal analysis
modopt,LANB,nbModes,-1e-6,1/(2*PI*SQRT(TauMin)),,OFF ! Normalize to
unit mass
*EIGEN,MatK,MatC,,EiV,MatPhiSolv
!  internal to Boundary conditions mapping
*MULT,Nod2Bcs,TRAN,MatPhiSolv,,MatPhi
!  Check mass normalization
*MULT,MatC,,MatPhiSolv,,APhi
*MULT,MatPhiSolv,TRANS,APhi,,PhiTMPhi
!  PRINT THIS MATRIX: IT SHOULD BE [I]
*PRINT,PhiTMPhi,PhiTMPhi.txt
```

Extracts nb modes with $\tau > TauMin$

Estimate Generalized forces (Load vector)

```
!  Fi=matPhi x VecF
!  internal to Boundary conditions mapping
*MULT,MatPhiSolv,TRAN,vecF,,modalForcesVec
*IF,indLoad,EQ,1,THEN
*DMAT,modalForces,D,COPY,modalForcesVec
*ELSE
*MERGE,modalForces,modalForcesVec,indLoad,COL
*ENDIF
```

Fills the modalForces matrix with generalized forces (nbModesxnbLoad)

Residual Vectors (1/2)

```
*LSENGINE,BCS,MyBcsSolver,MatK
*LSFACTOR,MyBcsSolver
*do,indLoad,1,nbLoad *SMAT,vecF,D,IMPORT,MAT,RunThermalVecF%indLoad%
!  CONSTRUCT EXACT SOLUTION *LSBAC,MyBcsSolver,VecF,TBcsExact
*MULT,Nod2Bcs,TRAN,TBcsExact,,T_Exact
!  CONSTRUCT THE APPROXIMATE SOLUTION
*VEC,T_MODAL,D,ALLOC,T_EXACT_ROWDIM
*do,indMode,1,nbModes !  Extract one mode at a time
*VEC,currVec,D,LINK,MatPhi,indMode
*AXPY,TauArray(indMode)*modalForces(indMode,indLoad),0,currVec,1.,0,T_Modal
*enddo
```

Residual Vectors (2/2)

```
!  Estimate Error (=residual vector)
*VEC,T_RESVEC,D,ALLOC,T_EXACT_ROWDIM
*AXPY,1.,0.,T_EXACT,1.,0.,T_RESVEC *AXPY,-1.,0.,T_MODAL,1.,0.,T_RESVEC
*ENDDO
!  Store Residual Vector into matrix
*IF,indLoad,EQ,1,THEN *DMAT,T_RESVEC_MAT,D,COPY,T_RESVEC *ELSE
*MERGE,T_RESVEC_MAT,T_RESVEC,indLoad,COL
*ENDIF
```

Constructs Temperature Residual Vectors (nbNodesxnbLoad)