Chapter 13
pyFBS: A Python Package for Frequency Based Substructuring and Transfer Path Analysis

Domen Ocepek, Miha Kodrič, Francesco Trainotti, Miha Pogačar, Tomaž Bregar, Daniel Jean Rixen, and Gregor Čepon

Abstract pyFBS is a Python package for Frequency Based Substructuring, Transfer Path Analysis, multi-reference modal identification, and a variety of different expansion techniques. It enables the user to use state-of-the-art dynamic substructuring methodologies in an intuitive manner. With the package, basic and application examples are also provided, together with real datasets, so you can directly try out the capabilities of pyFBS. It is currently being used by several undergraduate students and postgraduate researchers, but also offers great potential in various industrial projects. The pyFBS team plans to share the most advanced tools and methods in the field of dynamic substructuring and transfer path analysis. The goal is to create a common working base for scientists interested in this topic. Within this paper, the use of transfer path analysis within the pyFBS package is introduced. The package facilitates several approaches to model the transfer paths through the interface between the active and passive side of the assembly. This enables the indirect characterization of the source, commonly by estimating a set of equivalent forces that replicate the responses at the passive side of the assembly and lead to the identification of the critical transfer paths in terms of noise and vibration transfer. An example on a numerical case study is provided, demonstrating the efficiency and applicability of pyFBS in executing these tasks.

Keywords Python · Open Source · Transfer Path Analysis · Source Characterization · Numerical Case Study

Introduction

Transfer Path Analysis (TPA) is a useful engineering tool used to identify noise and vibration transfer paths traveling from the source substructure of an assembly to the connected passive substructures [1, 2]. TPA quantifies actively vibrating components, where vibrating mechanism are usually difficult to model or measure directly, and vibration transfer to connected passive substructures. By identifying the key contributors to noise and vibration, engineers can propose targeted solutions to reduce these unwanted effects.

There are conceptually different families of TPA methods with variations in their implementations [2]. Component-based TPA characterizes the source excitations by a set of equivalent (also known as blocked) forces, which are an inherent property of the active component. Equivalent forces are thus transferable to assemblies with modified passive sides, enabling component optimization of the assembly. Directly measured equivalent forces assume an infinitely rigid ground to which the source is mounted using force transducers. Load, measured during operation of the source, directly represents the blocked force set. Alternatively, an indirect approaches can be applied, where equivalent forces are estimated from responses measured near/at the interface and a separate admittance measurement of the structure. From the inverse problem, equivalent

Domen Ocepek · Miha Kodrič · Miha Pogačar · Gregor Čepon
‘ Faculty of Mechanical Engineering, University of Ljubljana, Aškerčeva 6, 1000 Ljubljana, Slovenia
e-mail: domen.ocepek@fs.uni-lj.si; miha.kodric@fs.uni-lj.si; pogacar.miha@fs.uni-lj.si; gregor.cepon@fs.uni-lj.si

Francesco Trainotti · Daniel Jean Rixen
Technical University of Munich, School of Engineering and Design, Boltzmannstr. 15, 85748 Garching, Germany
e-mail: francesco.trainotti@tum.de; rixen@tum.de

Tomaž Bregar
Gorenje d.o.o., Partizanska 12, Velenje, 3503, Slovenia
e-mail: tomaz.bregar@gorenje.si

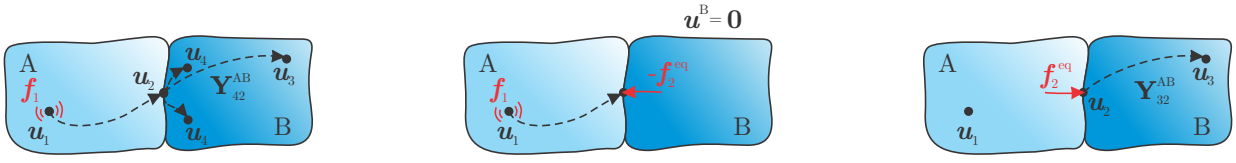
Matthew Allen et al. (eds.), *Special Topics in Structural Dynamics & Experimental Techniques, Vol. 5*, Conference Proceedings of the Society for Experimental Mechanics Series, https://doi.org/10.13052/97887-438-0150-4_13

force set is calculated that causes the same responses on the passive side when the source is turned off. Special attention is given to the *in-situ* approach [3] even eliminating the need to dismount any part of the assembly to obtain equivalent force set.

An open-source implementation of the TPA techniques is pyFBS [4], a python package for frequency based substructuring and transfer path analysis. Additionally, in recent years, pyFBS has developed into a compact and versatile package that expanded beyond FBS and TPA techniques only. Most recent additions to the package include various expansion techniques [5] and multi-reference modal identification tool [6]. This paper however, the use of transfer path analysis within the pyFBS package is introduced. First, short theoretical background on *in-situ* TPA is given, followed by a commonly adopted interface model using virtual point transformation [7]. Usage of the package is then demonstrated using numerical case study.

Theoretical background and notation

Consider a linear and time-invariant assembly of substructures A and B, coupled at the interface (Fig. 1a). Substructure A is an active component with the operational excitation $\mathbf{f}_1$. Meanwhile, no excitation force is acting on the passive substructure B. The responses on B are observed in three different sets of DoFs: at the interface DoFs ($\mathbf{u}_2$), in the proximity of the interface at the indicator DoFs ($\mathbf{u}_4$), and away from the interface at the target DoFs ($\mathbf{u}_3$).



(a) (b) (c)

Fig. 1 *In-situ* TPA: a) assembly of substructures A and B, b) $\mathbf{f}_2^{\mathrm{eq}}$ blocking the motion at the interface, c) replicating operational responses with $\mathbf{f}_2^{\mathrm{eq}}$.

Source excitations $\mathbf{f}_1$ are often not measurable in practice; therefore, *in-situ* TPA adopts a different approach for describing the operational excitations. A set of equivalent forces $\mathbf{f}_2^{\mathrm{eq}}$ is introduced, applied at the interface DoFs. If the source is deactivated, $\mathbf{f}_2^{\mathrm{eq}}$ yields the same responses on the passive side as $\mathbf{f}_1$[1]:

$$\mathbf{0} = \underbrace{\mathbf{Y}^{\mathrm{AB}}_{41}\mathbf{f}_1}_{\mathbf{u}_4} + \mathbf{Y}^{\mathrm{AB}}_{42}\left(-\mathbf{f}_2^{\mathrm{eq}}\right). \tag{1}$$

Expressing the equivalent forces $\mathbf{f}_2^{\mathrm{eq}}$ from the indicator responses $\mathbf{u}_4$ yields:

$$\mathbf{f}_2^{\mathrm{eq}} = \left(\mathbf{Y}^{\mathrm{AB}}_{42}\right)^{+}\mathbf{u}_4. \tag{2}$$

Receiver operational responses for the arbitrary assembly (with the same source) can be replicated by the set of equivalent forces (Fig. 1c) as follows:

$$\mathbf{u}_3 = \mathbf{Y}^{\mathrm{AB}}_{32}\mathbf{f}_2^{\mathrm{eq}}. \tag{3}$$

Interface modeling is critical to prevent redundancy/bad conditioning or neglecting important transfer paths through the interface. This means the experimentalist must account for all significant DoFs at the interface. Recently, virtual point transformation (VPT) has been proposed to model the interface where interface DoFs (rigid or flexible) are selected manually by the experimentalist to avoid redundancy yet retain the full controllability of the interface.

The loads $\mathbf{m}$ at the virtual point (VP) are obtained for a given vector of forces $\mathbf{f}$ in the proximity of the VP (Fig. 2).

The contribution from all the input forces can be combined and expressed as follows:

$$\mathbf{m} = \mathbf{R}_{\mathrm{f}}^{\mathrm{T}}\mathbf{f}, \tag{4}$$

where the IDM matrix $\mathbf{R}_{\mathrm{f}}^{\mathrm{T}} \in \mathbb{R}^{m \times n_{\mathrm{f}}}$ contains the positions and orientations for all the excitation locations with respect to the VP (Fig. 2). The inverse relationship of Eq. (4) is derived with a constrained minimization for forces:

$$\mathbf{f} = \mathbf{R}_{\mathrm{f}}\left(\mathbf{R}_{\mathrm{f}}^{\mathrm{T}}\mathbf{R}_{\mathrm{f}}\right)^{-1}\mathbf{m} = \mathbf{T}_{\mathrm{f}}^{\mathrm{T}}\mathbf{m} \quad \Rightarrow \quad \mathbf{T}_{\mathrm{f}}^{\mathrm{T}} = \mathbf{R}_{\mathrm{f}}\left(\mathbf{R}_{\mathrm{f}}^{\mathrm{T}}\mathbf{R}_{\mathrm{f}}\right)^{-1}. \tag{5}$$

[1] An explicit dependency on the frequency is omitted to improve the readability of the notation, as will be the case for the remainder of the paper.

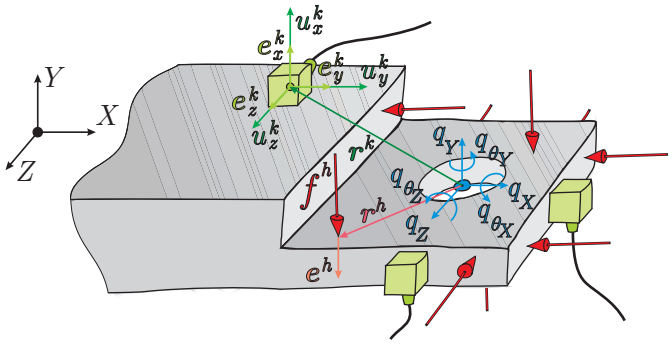


Fig. 2 Projection of responses on the k-th triaxial accelerometer and the h-th excitation onto the virtual point[2].

Transfer path admittance in Eqs. (2) and (3) can be expressed as follows with regards to the VPT:

$$\mathbf{Y}_{42}^{\mathrm{AB}} = \mathbf{Y}_{4\mathrm{f}}^{\mathrm{AB}}\,\mathbf{T}_{\mathrm{f}}^{\mathrm{T}}, \tag{6}$$

$$\mathbf{Y}_{32}^{\mathrm{AB}} = \mathbf{Y}_{3\mathrm{f}}^{\mathrm{AB}}\,\mathbf{T}_{\mathrm{f}}^{\mathrm{T}}. \tag{7}$$

Numerical case study

This numerical case study demonstrates how one can perform *in-situ* TPA using pyFBS. Within this example, the use of VPT is adopted to model the interface between the source and the receiver.

First, a 3D viewer within pyFBS is initialized, which is intended for the user to plan his experiment in virtual environment. *stl* model of the examined structure is displayed, alongside interactively positioned impacts and sensors for the purpose of obtaining admittance of the transfer paths:

```
# 3D view
view3D = pyFBS.view3D()

A = view3D.add_stl(stl_dir_A, color = "#83afd2", name = "A");
B = view3D.add_stl(stl_dir_B, name = "B");

# show sensors, channels and impacts from the dataframe
view3D.show_acc(df_acc_AB)
view3D.show_imp(df_imp_AB)
view3D.show_chn(df_chn_AB)
```

Listing 1 Initialize 3D viewer.

Full 3D display obtained using the code above is presented in Fig. 3.

The pyFBS package enables user-friendly modal analysis and FRF synthetization based on the mass and stiffness matrices imported from the FEM software. Currently, only data import from Ansys is supported. A numerical model of the previously showed AB assembly is imported, on which *in-situ* TPA will be showcased:

```
# MK model
MK_AB = pyFBS.MK_model('file.rst', 'file.full', no_modes=100, allow_pickle=True,
    recalculate=False, read_rst=True, scale=1)
```

Listing 2 Initialize MK model.

Next, locations of channels and impacts should be snapped to the nearest FE node, at which FRF will be evaluated later:

The position vector from the VP to the center of the sensor is denoted by $\mathbf{r}^k$. The unit vector for each accelerometer axis is $\mathbf{e}_i^k$ and the response in each axis is denoted by u_i^k ($i \in (x, y, z)$). The position vector from VP to the force impact is $\mathbf{r}^h$, the impact direction is $\mathbf{e}^h$ and the impact magnitude is f^h.

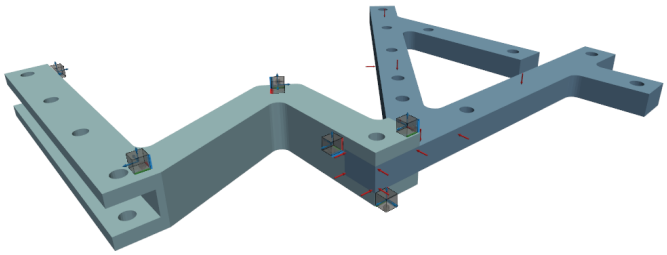


Fig. 3 3D viewer.

```
# Update locations of channels and impacts to snap to the nearest FE node.
df_chn_AB_up = MK_AB.update_locations_df(df_chn_AB)
df_imp_AB_up = MK_AB.update_locations_df(df_imp_AB)
```

Listing 3 Updating impact and channel locations.

And finally, for the numerical model, FRFs from the inputs at the impact locations/directions and outputs at the channel locations/directions are calculated using mode superposition method:

```
# Perform the FRF sythetization for each component based on the updated locations.
MK_AB.FRF_synth(df_chn_AB_up, df_imp_AB_up, f_start=0, f_end=1000, f_resolution=1,
    modal_damping=0.003, frf_type = "accelerance")

freq = MK_AB.freq
FRF = MK_AB.FRF
```

Listing 4 FRF synthetization.

For the VPT, positional data is required for channels, impacts and for virtual points. Note that only transformation of forces is required for successful identification of equivalent forces using in-situ TPA, but in order to define the VPT object in pyFBS, VP for displacements must also be defined (but neglected latter in the transformation):

```
# Prepare the reduction matrices for the VPT.
df_vp = pd.read_excel(pos_xlsx, sheet_name='VP_Channels')
df_vpref = pd.read_excel(pos_xlsx, sheet_name='VP_RefChannels')

vpt_AB = pyFBS.VPT(df_chn_AB_up, df_imp_AB_up, df_vp, df_vpref)

# Apply VPT
Y_um = FRF @ vpt_AB.Tf
```

Listing 5 Virtual point transformation.

With the admittance of the transfer paths, only operational responses must be imported. Source is then characterized in the following manner:

```
# Calculate blocked forces at the interface
f_eq = np.linalg.pinv(Y_um[:,:9,:6])@u4_op

# On-board validation
_Y_temp = Y_um[:,9:,:6]
u3_tpa = _Y_temp@f_eq
```

Listing 6 Obtaining equivalent force set and on-board validation.

The latter line already gives us on-board validation, responses at the target DoFs predicted directly from the obtained equivalent force set, when the source is turned off. Agreement between actual measured and predicted response for the given example is shown in Fig. 4.

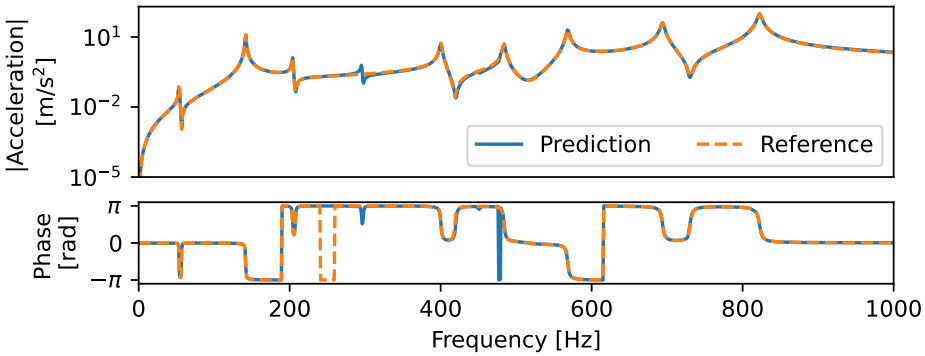


Fig. 4 On-board validation.

We can also calculate and evaluate partial transfer paths (Fig. 5):

```
# Partial transfer path can be calculated and evaluated.
sel_i = 1
u_partial = []

for j in range(6):
    gg = _Y_temp[:,sel_i:sel_i+1,j:j+1]@f_eq[:,j:j+1,0:1]
    u_partial.append(gg[:,0,0])

u_partial = np.asarray(u_partial).T
```

Listing 7 Partial transfer paths.

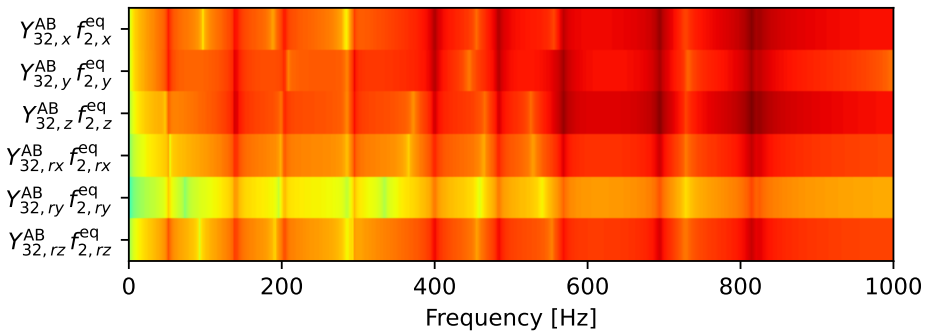


Fig. 5 Partial transfer path contribution to the full response of the target DoF.

The predicted response or each path contribution can be evaluated by listening to synthesized time response. Within pyFBS auralization can be performed with ease and the response can be played within Jupyter notebook with single line of code:

```
# Auralization of the operational response
import IPython.display as ipd

xt,s_op = pyFBS.auralization(freq,u3_tpa[:,sel_out,0])
ipd.Audio(s_op, rate=1/(xt[1]-xt[0]))
```

Listing 8 Auralization.

Conclusions

The pyFBS is intended as an open-source Python package for Frequency Based Substructuring and Transfer Path Analysis. The development of pyFBS is ongoing and is at present used as a research tool. Current status of the package enables user-friendly implementation of the Transfer Path Analysis methods, with the *in-situ* TPA showcased within this paper. For templates on other TPA approaches, describing the source either using equivalent or interface force set, an interested reader is referred to www.pyfbs.com, where they are elaborated in detail.

References

1. Verheij, J.W. *Multi-path sound transfer from resiliently mounted shipboard machinery: Experimental methods for analyzing and improving noise control*. PhD thesis, TU Delft (1982)
2. Van Der Seijs, M.V., De Klerk, D., and Rixen, D.J. "General framework for transfer path analysis: History, theory and classification of techniques". *Mechanical Systems and Signal Processing*, 68:217–244 (2016)
3. Moorhouse, A., Elliott, A., and Evans, T. "In situ measurement of the blocked force of structure-borne sound sources". *Journal of sound and vibration*, 325(4-5):679–685 (2009)
4. Bregar, T., Mahmoudi, A., Kodrič, M., Ocepek, D., Trainotti, F., Pogačar, M., Göldeli, M., Čepon, G., Boltežar, M., and Rixen, D. "pyfbs: A python package for frequency based substructuring". *Journal of Open Source Software*, 7(69):3399 (2022)
5. Pogačar, M., Ocepek, D., Trainotti, F., Čepon, G., and Boltežar, M. "System equivalent model mixing: A modal domain formulation". *Mechanical Systems and Signal Processing*, 177:109239 (2022)
6. Peeters, B., Van der Auweraer, H., Guillaume, P., and Leuridan, J. "The polymax frequency-domain method: a new standard for modal parameter estimation?". *Shock and Vibration*, 11(3-4):395–409 (2004)
7. Van Der Seijs, M.V., Van Den Bosch, D.D., Rixen, D.J., and de Klerk, D. "An improved methodology for the virtual point transformation of measured frequency response functions in dynamic substructuring". In *4th ECCOMAS thematic conference on computational methods in structural dynamics and earthquake engineering*, volume 4 (2013)