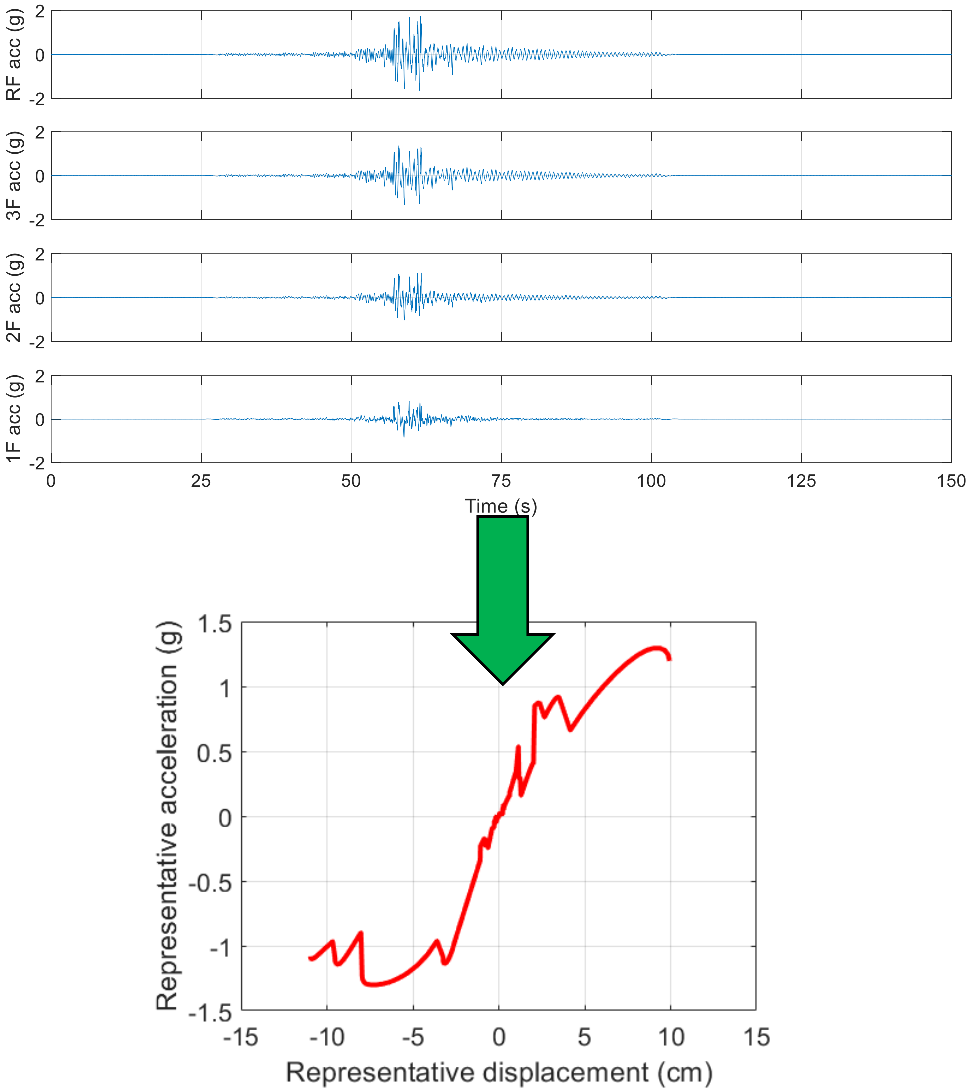
MAY 9, 2022



CAPACITY CURVE EXTRACTION

MATLAB AND PYTHON IMPLEMENTATION USAGE GUIDE

TREVOR ZHIQING YEOW
EARTHQUAKE RESEARCH INSTITUTE, UNIVERSITY OF TOKYO
tyeow.work@gmail.com

Version 1.0

Written on 2022/05/10

Version History

N/A

Contents

1.0 MOTIVATION OF CAPACITY CURVE EXTRACTION

1.1 Need for structural health monitoring

Following major seismic events, visual inspections of building damage are often required to evaluate if it can be safely occupied. Such inspections are time consuming, especially in an area with dense building stock and limited number of engineers. This may result in occupants being unnecessarily displaced from safe buildings for a long duration of time while waiting for their building to be evaluated which can cause significant loss of income or added strain on temporary shelters. On the other hand, such inspections may be subjective, and damage may be hidden behind other building components or misdiagnosed. This can lead to unsafe buildings mistakenly being judged as safe which could lead to significant injuries and death if it collapses during a strong aftershock or future seismic event.

Due to the above issues, there is a need for another method of evaluating building damage to either replace or supplement visual inspections. Recently, structural health monitoring is seen as a potential solution to these issues. While there are many methods available in literature, the capacity curve extraction method proposed by **Kusunoki et al. (2018)** with further modifications by **Yeow et al. (2022)** is the focus of this implementation. Details on the Matlab and Python implementations are provided in **Section 2.0**. Further descriptions on the theory and application of their methodology are described from **Sections 3.0** to **8.0**, and an example is provided in **Section 9.0**.

1.2 Benefits of adopted capacity curve extraction

As mentioned in the previous section, there are many methods available in literature to perform structural health monitoring, such as monitoring changes in dynamic behavior with time or using Kalman filters. However: (i) a building's dynamic properties may not be sensitive to building damage, (ii) hysteretic and/or building models often need to be created or assumed for such implementations, and (iii) such methods may be sensitive to noise in the recordings. On the other hand, the approach proposed by **Kusunoki et al. (2018)** only requires floor acceleration recordings and an estimate of the distribution of floor masses along the height of the building. Furthermore, modifications by **Yeow et al. (2022)** enabled the method to be automated and unbiased, which allows for it to be easily adopted in engineering practice.

1.3 Application to assessing building damage

The implemented methodology and theory discussed in this document are for the purposes of extracting out the capacity curve of a building's response during an earthquake. There are several possible ways to use such information to assess building damage which are currently not included in the implementations. These are described in **Kusunoki (2020)** and **Yeow et al. (2022)**. Some examples are shown in **Figure 1-1**, where building damage can be classified using the extent of ductility response (**Figure 1-1a**). Alternatively, the safety can be judged by the possibility of exceeding its safety limit during a strong aftershock of equal intensity to the main shock (**Figure 1-1b**).

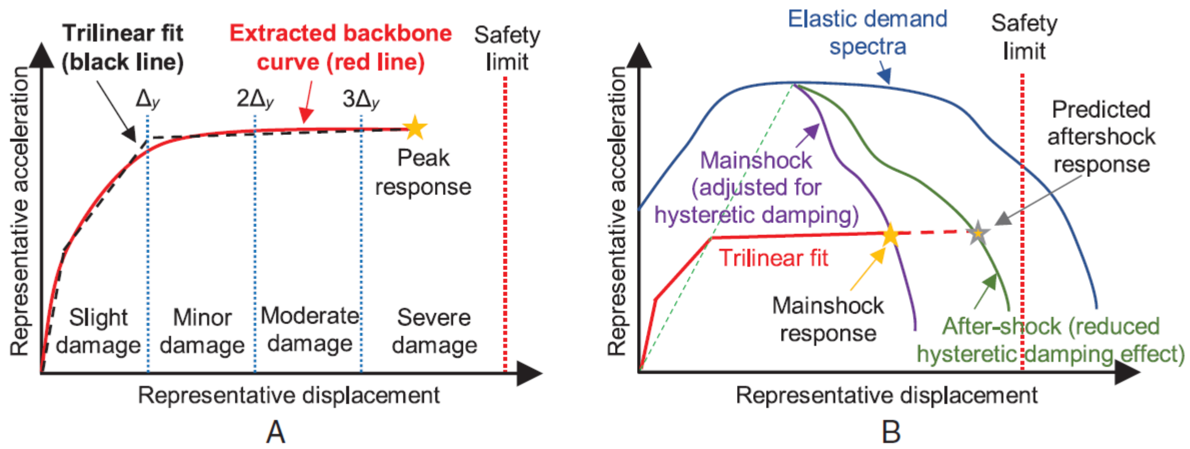


Figure 1-1. Applications of extracted capacity curve for evaluating damage (Yeow et al., 2022); (a) consideration of ductility response, and (b) evaluation of safety for aftershock of equal intensity as main shock

Kusunoki (2020) proposed that the boundary between moderate and severe damage can be based on the maximum ductility response permitted before the safety limit is exceeded during the strong aftershock, μ^*, and proposed the following to evaluate this ratio:

$$\mu^* = \frac{1}{16}\left(4.41\mu_{SL} + 7.98\sqrt{\mu_{SL}} + 3.61\right) \tag{1-1}$$

Where μ_{SL} is the ductility response corresponding to the safety limit state.

The boundary between minor and moderate damage is set to be the halfway point between yield (i.e., μ = 1.0) and μ^*. The values shown in **Figure 1-1a** correspond to μ_{SL} being approximately 6.

In order to determine the extent of ductility response, a trilinear fit is required for reinforced concrete buildings. Such methods are available in literature (i.e., **Wang et al. (2021)**) but was not incorporated into the Matlab and Python implementations at this stage as other potential methods of evaluating building damage are currently still under investigation.

1.4 Limitations of adopted capacity curve extraction method

Extensive validation of the methodology had been performed by **Yeow et al. (2022)** considering reinforced concrete (RC) frame buildings. However, they noted several limitations of the methodology. Users should be aware of these limitations before using the functions.

Firstly, the methodology was unable to properly consider long-period components of building displacement response. This is due to a combination of: (i) the zero-mean nature of wavelets which were used to filter out frequency components, (ii) the use of double integration to estimate displacements from acceleration data, and (iii) long period components being indistinguishable from noise components. This can lead to an underestimation of displacement response.

The second issue is the current inability to consider time-variation in the contribution of various frequency components to the predominant mode of response. In reality, building damage may result in the frequency of higher-mode components decreasing and overlapping that of the predominant response mode during elastic response, and hence higher-mode components may be incorrectly included in the final extracted capacity curve. Conversely, inelastic response may be incorrectly filtered since its frequency could overlap long-period noise components during elastic response. Methods to address the time-variation in the contributions are currently being researched.

The final issue is that while the methodology is applicable to multistory buildings, it essentially condenses the response down into an equivalent single-degree-of-freedom system. This raises the question on whether the method is capable of capturing the response of the building if a higher-mode type failure occurs, such as the case where a soft-story failure occurred near the top of the building. This mode of failure should be rare given modern engineering practice, but the ability to capture such damage mechanisms if it does occur is still beneficial.

Due to the above issues, users are cautioned against applying this framework to buildings exhibiting elastoplastic hysteretic behavior in its current state. Further research is being performed to address these limitations, and such changes will be implemented into the functions in the near future.

2.0 INSTALLATION AND USAGE GUIDE

2.1 Matlab version

2.1.1 Setting up Matlab version

The Matlab version of the code requires no special setup. However, it does require the Wavelet Toolbox product in addition to the basic version of Matlab. Once these are installed, the script can be immediately executed.

2.1.2 Matlab version description

Function call command

SelectedRanks = capcurvemaster(TotalAcc, dT, ...)

Input parameters

Required inputs:

Parameter	Description
TotalAcc	Matrix containing total floor acceleration data in units of "g". Matrix should have dimensions of *SigLength* by *NumF*, where *SigLength* is the length of each floor recording signal, and each *NumF* is the number of floors in the building (including ground and basement floors if applicable). Each subsequent column should correspond to each subsequent floor level in increasing order. E.g., column [1] should be the lowest floor, while column [*NumF*] should be the roof level
dT	Timestep between each recorded data. This is assumed to be constant and in units of seconds

Optional inputs:

Parameter	**Description**
Mass	A row of floor mass values, with each entry corresponding to the corresponding floor from " TotalAcc ". E.g., the last entry in Mass should correspond to the mass of the roof level. **Note 1:** if this row is not provided, it will be assumed that the mass is the same on all floors of the building. **Note 2:** if the user specifies this row themselves, it must have dimensions of 1 by *NumF*. If this condition is not satisfied, the function will terminate prematurely.
wname	The mother wavelet adopted for performing the discrete wavelet transform method. **DEFAULT - 'sym10'**
plotflag	A 1 by 3 entry which flags which figures should be plotted plotflag(1) = 1: plot individual rank hysteretic response plotflag(2) = 1: plot tentative hysteretic response plotflag(3) = 1: plot final extracted backbone curve **Note:** if any other values are used for flags, the corresponding plot will not be generated

Optional output file names (NOTE: if not provided, the corresponding output file will not be generated)

Parameter	**Description**
outrankd	Name for file containing representative response for each individual rank
outrank	Name for file containing properties of each individual rank
outcombd	Name for file containing floor response considered selected ranks only
outhyst	Name for file containing tentative representative displacement-acceleration hysteresis
outscurve	name for file containing skeleton curve

Output parameters

Parameter	**Description**
`SelectedRanks`	List of selected ranks

Required toolboxes

- Wavelet toolbox (https://www.mathworks.com/products/wavelet.html)

2.1.3 Matlab version call function application

As described in Section 2.1.2, the call function to run the application in Matlab is:

>>SelectedRanks = capcurvemaster(TotalAcc, dT, ...)

"TotalAcc" and "dT" are variables and can be applied without firstly defining the parameter name. On the other hand, all optional inputs require the parameter name to be called first before providing the corresponding value or text.

For example, if we have *m* floor recordings, each with a length of *n* with a timestep of T_{step}, and we want to export the final capacity curve into a file named 'Temp.txt', we can assign the variables as:

$$>>\text{TotalAcc} = \begin{bmatrix} A_{11} & A_{12} & A_{13} & \dots & A_{1m} \\ A_{21} & A_{22} & A_{23} & \dots & A_{2m} \\ A_{31} & A_{32} & A_{33} & \dots & A_{3m} \\ \dots & \dots & \dots & \dots & \dots \\ A_{n1} & A_{n2} & A_{n3} & \dots & A_{nm} \end{bmatrix};$$

>> dT = T_{step};

>> M = [M_1 M_2 M_3 … M_m];

>> filename = 'Temp.txt';

Where A_{ij} is the i^{th} total acceleration data of the j^{th} sensor and M_j is the mass corresponding to the floor where the j^{th} sensor was attached. We will need to state the 'Mass' and 'outscurve' string in order to call the function as follows:

>> SelectedRanks = capcurvemaster(TotalAcc, dT, 'Mass', M, 'outscurve', filename)

Alternatively:

>> SelectedRanks = capcurvemaster(TotalAcc, dT, 'Mass', [M_1,M_2,M_3,…,M_m];, 'outscurve', 'Temp.txt')

NOTE: The optional inputs do not have to be specified in the same order listed in Section 2.1.2. For example, 'outscurve' can be defined before 'Mass'. However, the subsequent input must be related to the specified parameter name (i.e., M must come after immediately 'Mass', and filename must come immediately after 'outscurve').

2.2 Python/Pycharm version

2.2.1 Setting up Python/Pycharm version

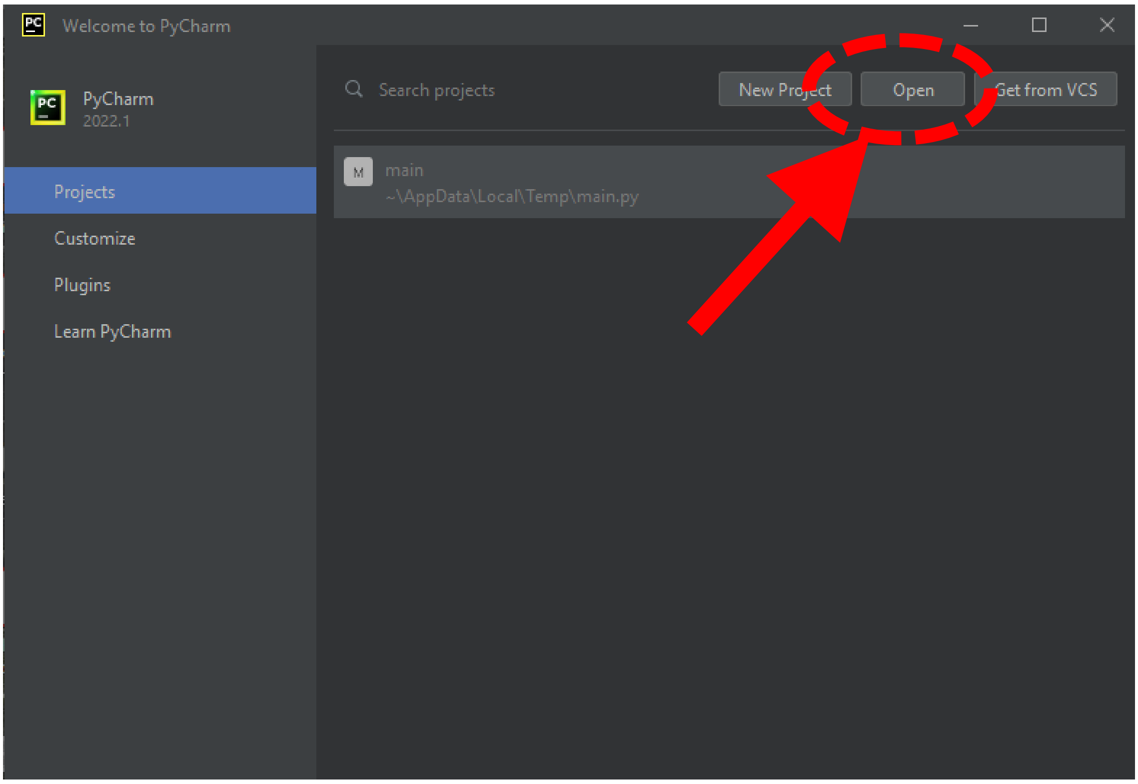
1. Download the latest version of Python from the following link: https://www.python.org/downloads/

2. Download the latest version of Pycharm, which is a Python user interface. Other Python user interfaces can be used, but the remaining instructions are specific to Pycharm. Link as follows: https://www.jetbrains.com/pycharm/

3. Install both Python and Pycharm. We recommend installing both in the same drive.

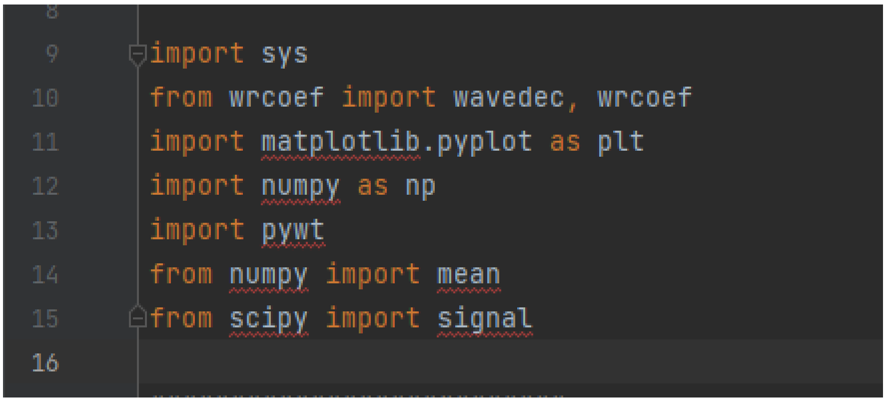
4. Download and unzip “CapCurveExtract.zip” into a folder of your choice.

5. Open Pycharm, select “open”, and select the “CapCurveEx” folder to open the project

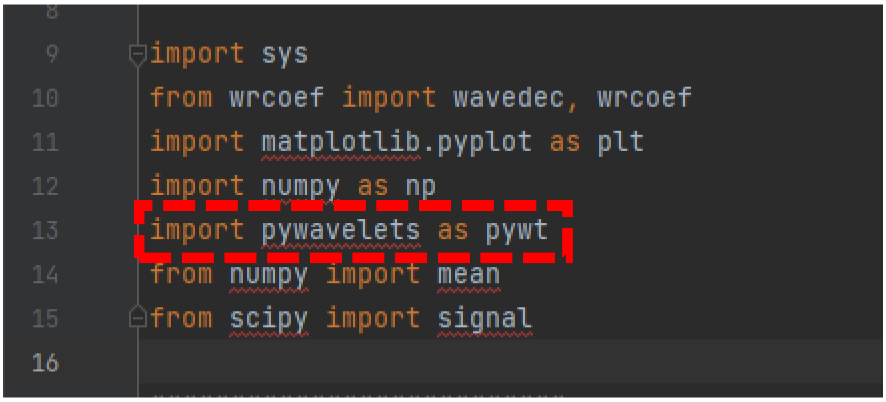


6. Open “capcurveextraction.py”. Expand line 9 to show required packages. If a package has not been installed, it will be underlined in red. In the following figure for example, “matplotlib”, “numpy”, “pywavelets”, and “scipy” have not been installed. Note that “mean” and “signal” are also underlined

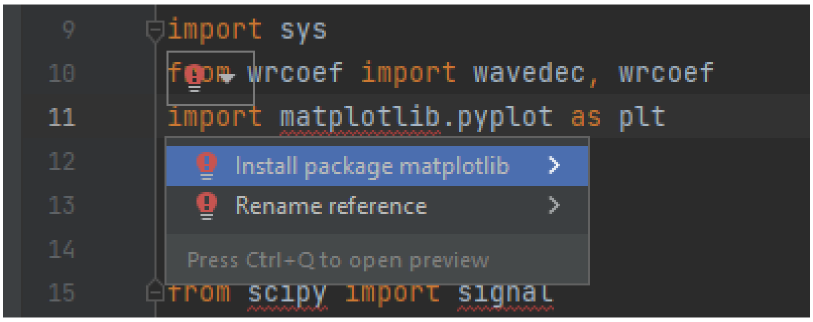
in red. However, these are imported from “numpy” and “scipy” packages, so it does not require any further installation as long as the latter two have been installed.

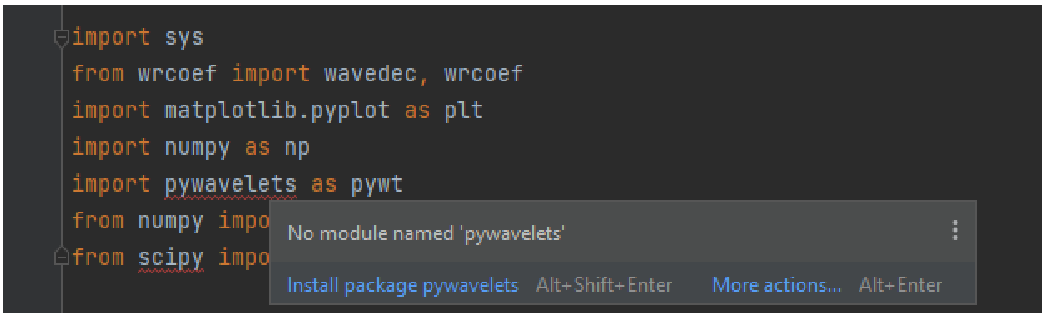


7. If pywt needs to be imported, replace “import pywt” with “import pywavelets as pywt”. This is because the package name we want to install is “pywavelet”, but the module name needed for the code is “pywt”.

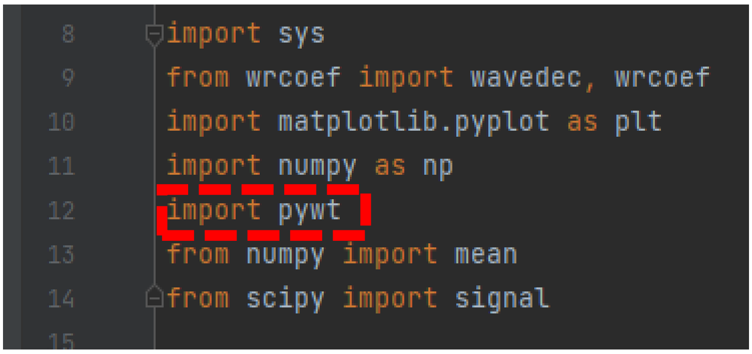


8. If there are any uninstalled packages, click on the package name. A red “!” mark will appear to the left. Click this mark, then “Install package (package name)”. Click on this option and Python will automatically install the package. You will require an internet connection if the package must be downloaded prior to installation. The following example shows this for “matplotlib”. Repeat for all other packages. Alternatively, the option to install the package may also appear by hovering the cursor over the package name underlined in red.





9. If step 7 was required, replace “import pywavelets as pywt” back to “import pywt”. This is because the script calls for the module name, not the package name. Now that it has been installed, we no longer need to call for “pywavelets”



10. Python and Pycharm should now be setup to call the function.

2.2.2 Python version description

Function call command

Selectedranks = capcurvemaster(data, dT, …)

Input parameters

Required inputs:

Parameter	**Description**
data	Matrix containing total floor acceleration data in units of "g". Matrix should have dimensions of *SigLength* by *NumF*, where *SigLength* is the length of each floor recording signal, and each *NumF* is the number of floors in the building (including ground and basement floors if applicable). Each subsequent column should correspond to each subsequent floor level in increasing order. E.g., column [0] should be the lowest floor, while column [*NumF*-1] should be the roof level
dT	Timestep between each recorded data. This is assumed to be constant and in units of seconds

Optional inputs:

Parameter	**Description**
Mass	A row of floor mass values, with each entry corresponding to the corresponding floor from "data". E.g., the last entry in Mass should correspond to the mass of the roof level. **Note 1:** if this row is not provided, it will be assumed that the mass is the same on all floors of the building. **Note 2:** if the user specifies this row themselves, it must have dimensions of 1 by *NumF*. If this condition is not satisfied, the function will terminate prematurely.
Wavelet	The mother wavelet adopted for performing the discrete wavelet transform method. **DEFAULT - 'sym10'**
plotflag	A 1 by 3 entry which flags which figures should be plotted plotflag[0] = 1: plot individual rank hysteretic response plotflag[1] = 1: plot tentative hysteretic response plotflag[2] = 1: plot final extracted backbone curve **Note:** if any other values are used for flags, the corresponding plot will not be generated

Optional output file names (NOTE: if not provided, the corresponding output file will not be generated

Parameter	**Description**
outrankd	Name for file containing representative response for each individual rank information
outrank	Name for file containing properties of each individual rank
outcombd	Name for file containing floor response considered selected ranks only
outhyst	Name for file containing tentative representative displacement-acceleration hysteresis
outscurve	name for file containing skeleton curve

Output parameters

Parameter	**Description**
`SelectedRanks`	List of selected ranks

Required modules

- Math
- sys
- matplotlib
- numpy
- pywavelets
- scipy

Additional code(s) from Github

wrcoef (developed by Ilya Zlotnik, 2017): https://github.com/izlotnik/wavelet-wrcoef

2.2.3 Python version call function application

As described in Section 2.2.2, the call function for the Python implementation is:

\>> Selectedranks = capcurvemaster(data, dT, …)

"data" and "dT" are parameter values and can be applied without firstly defining the parameter name. On the other hand, all optional inputs requires the parameter name to be called, followed by an equal sign (=) and immediately followed by the parameter values.

For example, if we have *m* floor recordings, each with a length of *n* with a timestep of T_{step}, and we want to export the final capacity curve into a file named 'Temp.txt', we can assign the variables as:

\>>TotalAcc = [A_{11}, A_{12}, A_{13}, …, A_{1m}
A_{21}, A_{22}, A_{23}, …, A_{2m}
A_{31}, A_{32}, A_{33}, …, A_{3m}
…, …, …, …, …
A_{n1}, A_{n2}, A_{n3}, …, A_{nm}];

\>> dT =T_{step};

\>> M = [M_1, M_2, M_3, …, M_m];

\>> filename = 'Temp.txt';

Where A_{ij} is the i^{th} total acceleration data of the j^{th} sensor and M_j is the mass corresponding to the floor where the j^{th} sensor was attached. We will need to state the 'Mass' and 'outscurve' string in order to call the function as follows:

\>> Selectedranks = capcurvemaster(TotalAcc, dT, Mass = M, outscurve=filename)

Alternatively:

\>> Selectedranks = capcurvemaster(TotalAcc, dT, Mass = [M_1, M_2, M_3, …, M_m], outscurve='Temp.txt')

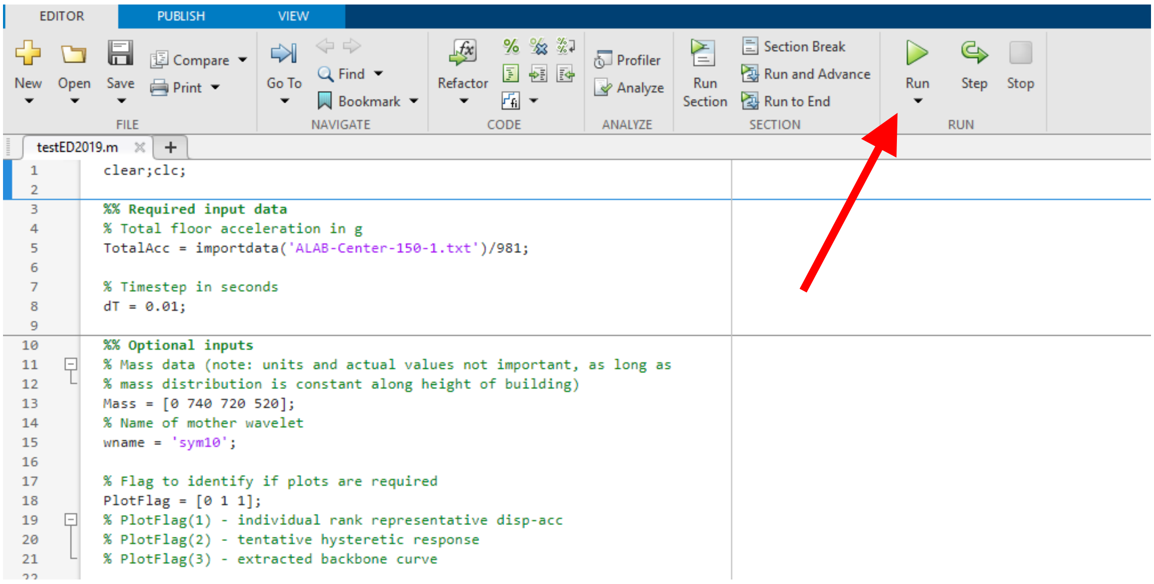
NOTE: The optional inputs do not have to be specified in the same order listed in Section 2.2.2. For example, 'outscurve' can be defined before 'Mass'.

2.3 Example of function application

For both the Matlab and Python implementations, an example script is provided to demonstrate how the call function works. The corresponding script is called ‘testED2019’ and uses data from a shake-table test of a 3-story RC disaster management center performed at E-Defense in 2019 (see **Yeow et a. (2021)** for more information on the test program). A detailed description of the calculations performed within the example is provided in **Section 9.0**.

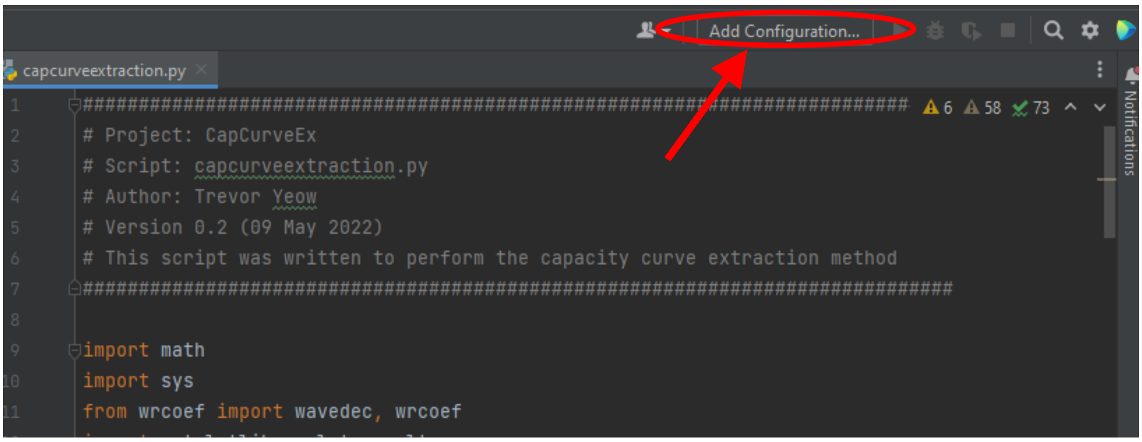
2.3.1 Calling example on Matlab

To call the example on Matlab, simply open the “testED2019.m” script, then select “run” under the “editor” tab near the top of the editor window.

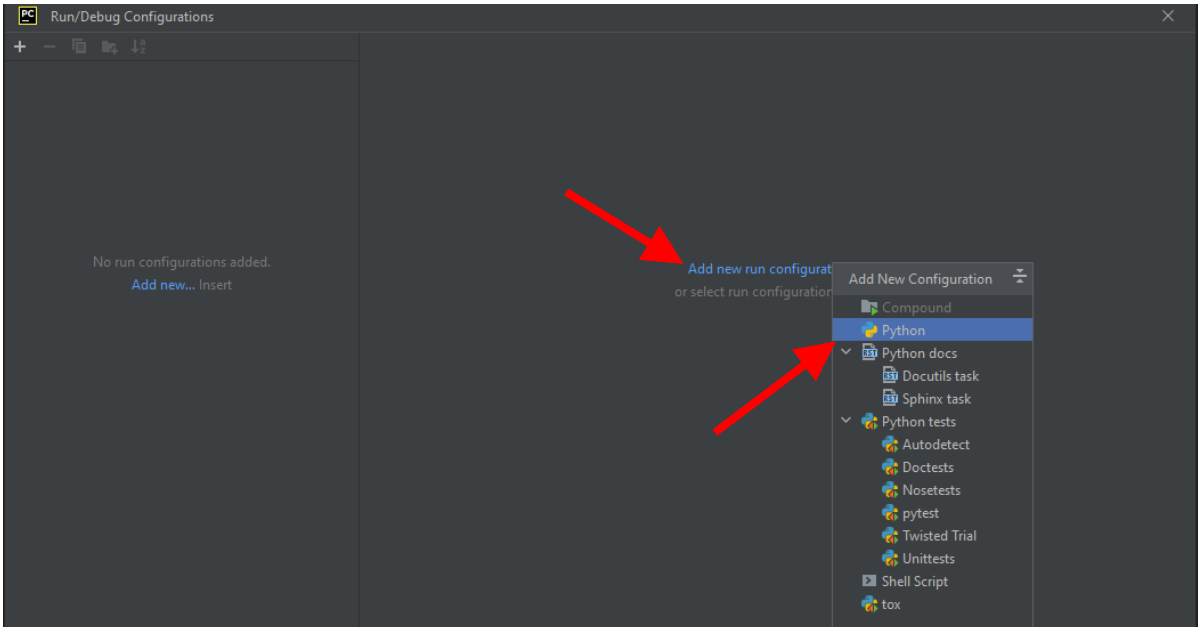


2.3.2 Calling example on Python

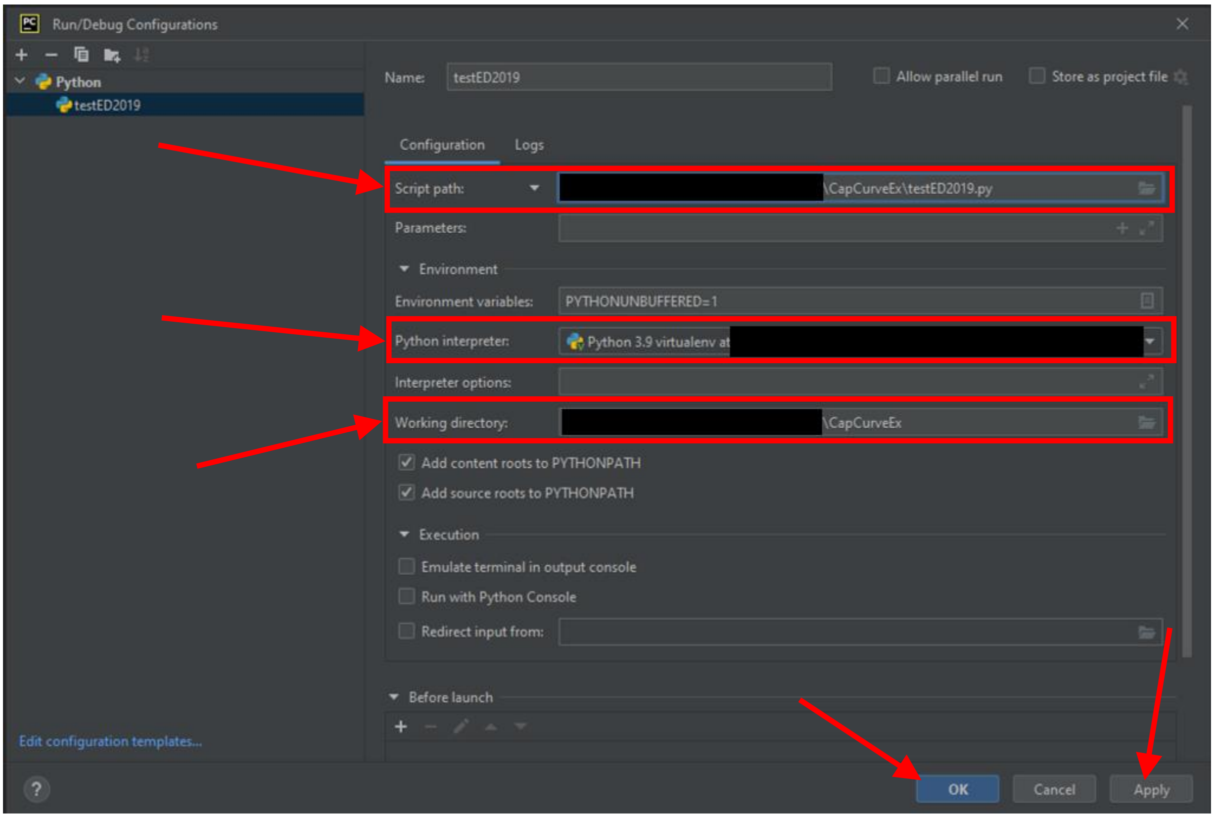
To call the example on Python, firstly select “Add Configuration” at the top right of the window.



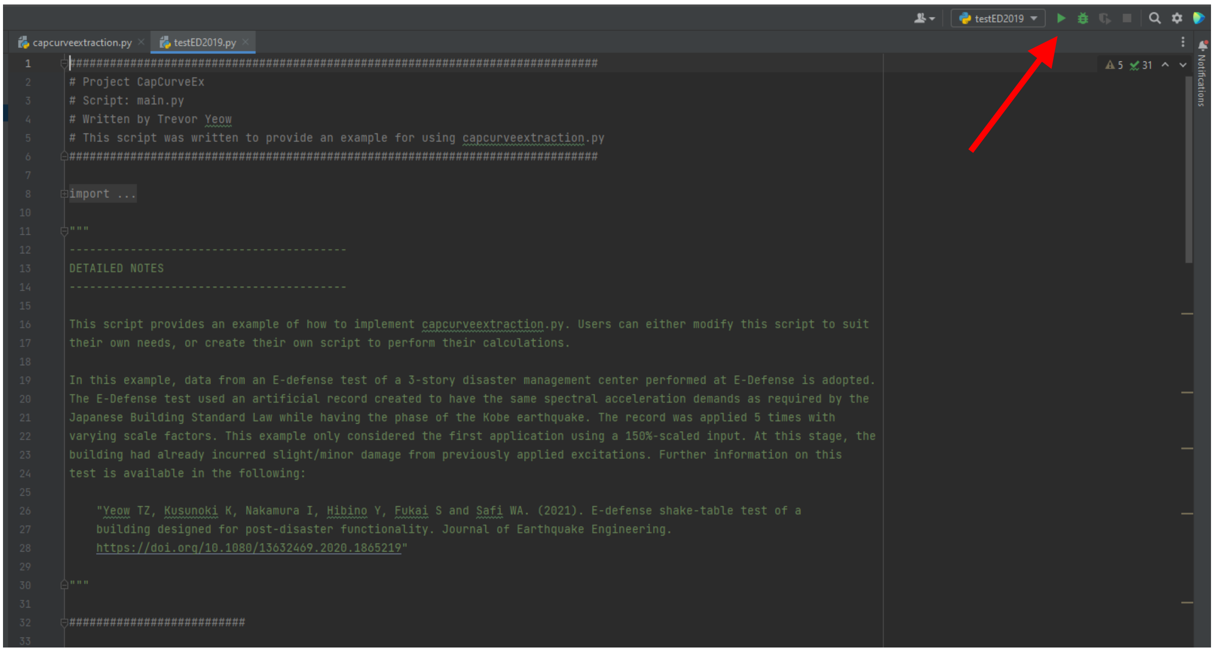
In the new window, select “add new run configuration” then “Python”



Open “testED2019.py” in the script path, select the correct Python interpreter and working directory. Click “Apply” then “OK”



The example can now be run by selecting the green triangle at the top right of the main window.



3.0 CAPACITY CURVE EXTRACTION OVERVIEW

The capacity curve extraction methodology follows that originally proposed by **Kusunoki et al. (2018)** with further modifications made by **Yeow et al. (2022)**. The methodology follows 8 key steps as follows:

1) Decompose total floor acceleration signals using the discrete wavelet transform method,
2) Calculate the relative displacement and acceleration response of the decomposed signals,
3) Derive the representative displacement and acceleration response, as well as the effective mass, for each decomposition level,
4) Calculate key parameters for the representative acceleration-displacement relationship for each decomposition level,
5) Select the decomposition levels which corresponds to the predominant mode of response based on the key parameters obtained in step (4),
6) Reconstruct the signal considering only the selected decomposition levels,
7) Derive the representative displacement and acceleration response for the reconstructed signal, and
8) Extract out the capacity curve.

Steps 1, 2 and 3 are described in greater detail in **Sections 4**, **5** and **6**, respectively. Steps 5 and 6 are detailed in **Section** 7, while the remaining steps are covered in **Section 8**. A detailed example demonstrating the application of the methodology is provided in **Section 9**.

4.0 DISCRETE WAVELET TRANSFORM

4.1 Overview

A key aspect of the capacity curve extraction methodology is the removal of frequency components corresponding to higher mode effects or long-period noise. To achieve this, the hysteretic response at a given frequency band is required to judge if it corresponds to the predominant response mode. This can be obtained by decomposing the floor acceleration signals into different frequency bands, followed by condensing the multi-degree-of-freedom (MDOF) response into a representative single-degree-of-freedom (SDOF) response for each individual frequency range. This section will cover the signal decomposition process.

Mathematical transforms are commonly adopted to decompose signals. A commonly applied approach is Fourier transforms which uses sine waves. However, since these sine waves are typically of infinite length and constant amplitude, the signal is only transformed in the frequency domain. Another method is the Wavelet Transform which uses wavelets instead. As wavelets are functions of finite length that begins and ends at zero, these are better able to capture localized changes in the signal.

There are two main types of wavelet transforms: (i) discrete (DWT), and (ii) continuous (CWT). The key difference between the two approaches is that the CWT approach uses finer discretization of frequency, whereas the DWT approach halves the frequency with each decomposition level. While the CWT method does provide more detailed information in the frequency domain, the higher number of frequency bands does create added complexity when determining range of frequencies to remove. Instead, the DWT method was adopted into the methodology. There is still potential in using the CWT method, but research is ongoing.

This section will briefly describe the DWT method in layman's terms. Users interested in more detailed theoretical background on the DWT are encouraged to refer to literature.

4.2 Mother wavelets

The first step in performing DWT is the selection of a "mother wavelet" which will be the basis of the wavelet shape used for sampling from the signal. There are several different wavelet families (e.g., Symlets, Daubechies, Meyer, etc). Within each family, the number of vanishing moments, N, can be specified. A mother wavelet can be selected by specifying the wavelet family and N. For example, on Matlab, "sym10" refers to the Symlet family with N = 10.

Examples of various mother wavelets are shown in **Figure 4-1**. Here, a larger *N* usually results in a more complex and wider wavelet shape to enable it to represent higher-order polynomial behavior. On one hand, a smaller *N* would be better able to capture extremely localized changes in a signal. On the other, there is a possibility that such sudden changes may be due to noise rather than it being important information on the building response, and that using a wider sampling range might result in response which is more representative of the frequency range of interest. Given that the use of wavelets is already far better at capturing localized changes in the signal, and that the purpose of this method was to remove noise error, a wider window was desired for defining the mother wavelet, though not too wide that no localized changes are captured. Based on this, 'sym10' was adopted for calibrating the capacity curve extraction method and is the default wavelet used in the Matlab and Python scripts if the user does not specify another mother wavelet. ***Users are warned that the framework may not work as well if another mother wavelet other than 'sym10' is adopted.***

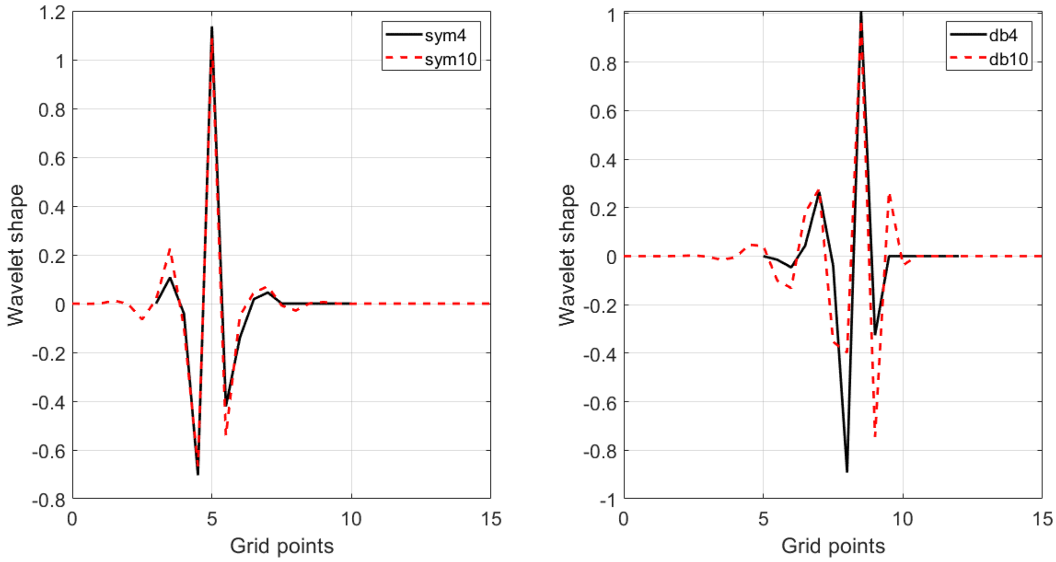


Figure 4-1. Examples of wavelet shape

For more information on other wavelet families, please refer to the link below. Note that while the descriptions in the provided link were for Matlab, Python also works in a similar fashion so the theory would still be applicable.
https://www.mathworks.com/help/wavelet/gs/introduction-to-the-wavelet-families.html

4.3 Child wavelets

Once a mother wavelet is selected, the wavelet's width is scaled and translated along the time-axis to create child wavelets for sampling the signal. With each subsequent decomposition level, the wavelet's width is doubled. An example of this is shown in **Figure 4-2**.

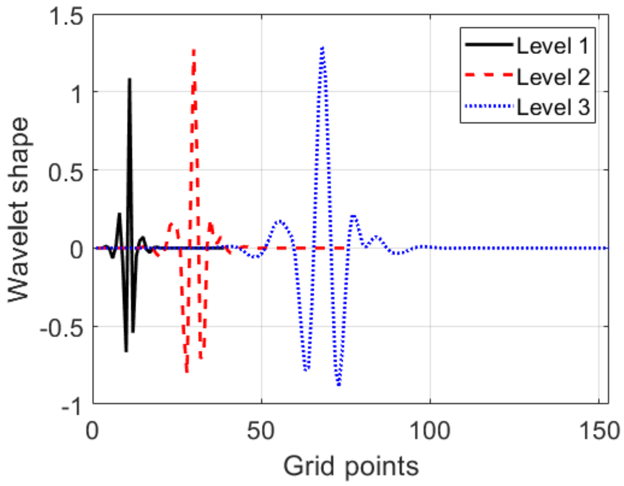


Figure 4-2. Examples of wavelets for different decomposition levels

4.4 Signal decomposition

For the first decomposition level, wavelets of the same shape as the mother wavelet with unscaled width was used for sampling the original signal. From this process, "detail coefficients", which are coefficients associated with the wavelet function, are obtained. The signal reconstructed using the detail coefficients contains high-frequency components of the original signal which was "filtered" out at this decomposition level. Additionally, "approximation coefficients" can be obtained to capture low-frequency information using scaling functions. To avoid confusing readers as to the details behind scaling functions and for simplicity purposes, signals reconstructed from approximation coefficients can be considered as the "residual signal". Denoting $g_0(t)$ as the original signal, and $f_1(t)$ and $g_1(t)$ as the signals reconstructed from detail and approximation coefficients during the 1st decomposition level, respectively, this decomposition can be simply represented as:

$$g_0(t) = f_1(t) + g_1(t) \tag{4-1}$$

At the next decomposition level, the child wavelet's width is doubled, and the DWT is performed on the residual signal from the previous decomposition level. Therefore, the i^{th} decomposition level can be represented as follows:

$$g_{i-1}(t) = f_i(t) + g_i(t) \tag{4-2}$$

Therefore, combining equations together gives the following after N decompositions were applied:

$$\begin{aligned} g_0(t) &= f_1(t) + g_1(t) \\ &= f_1(t) + f_2(t) + g_2(t) \\ &= f_1(t) + f_2(t) + f_3(t) + g_3(t) \\ &= f_1(t) + f_2(t) + \cdots + f_{N-1}(t) + f_N(t) + g_N(t) \end{aligned} \tag{4-3}$$

To illustrate **Equation (4-3)**, six levels of decomposition were applied to the roof total acceleration signal recorded from a 2019 E-Defense shake table test of a 3-story reinforced concrete disaster management center subjected to the first 150% scaled input excitation. Further information on this project can be found at **Yeow et al. (2022)**. The reconstructed signals are shown in **Figure 4-3**. Here, the recomposed signal using the first level detail coefficients, f_1, had very small amplitudes compared to the original signal. In contrast, the amplitudes of the recomposed signal using detail coefficients at the 5th and 6th decomposition level, f_5 and f_6, had large amplitudes, while the residual signal at the end of the 6th decomposition level g_6, was small. This demonstrates that the predominant frequency component of the building's response corresponds to that filtered out during the 5th and 6th decomposition levels.

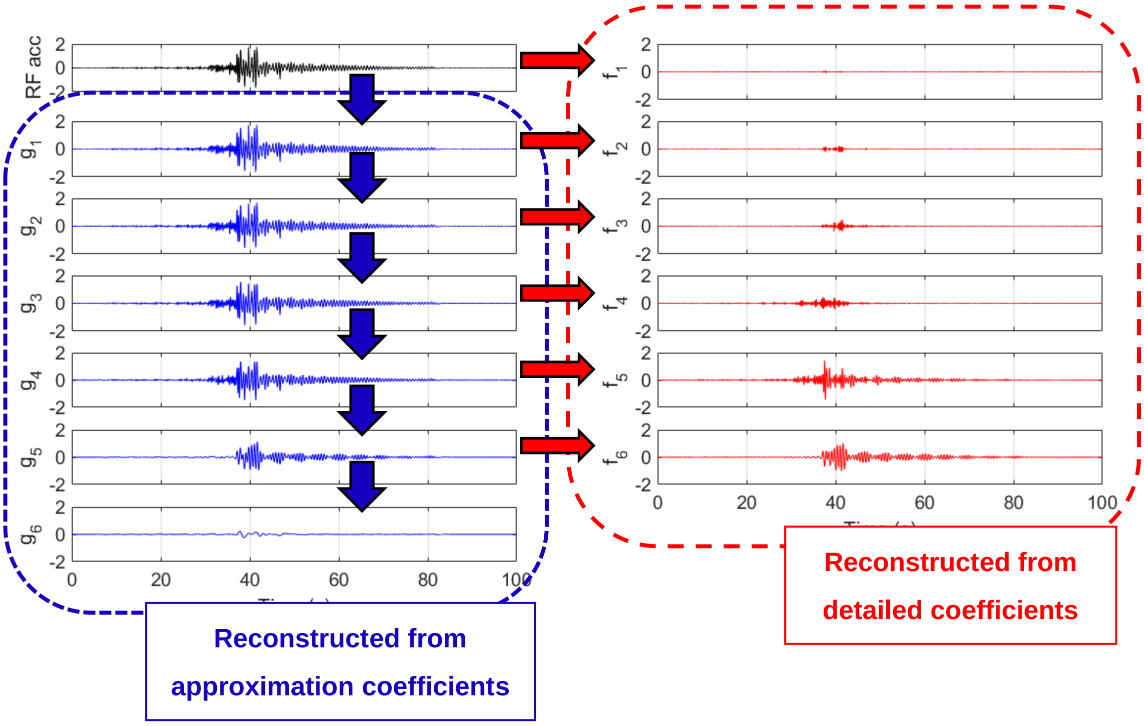


Figure 4-3. Demonstration of signal decomposition using RF acceleration signal of 2019 E-Defense shake table test of 3-story disaster management center (150%-1)

4.5 Maximum number of decomposition levels

Matlab and Python have functions to determine the maximum number of decomposition levels which can be applied. These consider the number of datapoints and the shape and order of the selected wavelet. If an even higher decomposition level was considered, the length of the wavelet would be longer than the remaining number of coefficients, resulting in the coefficients for the next decomposition level to be incorrect.

It should be noted that the maximum number of decomposition levels is generally smaller than that proposed by **Kusunoki et al. (2018)** which was as follows:

$$n_{level} \quad = log_2(ndata) \tag{4-4}$$

Where n_{level} is the maximum decomposition level which can be considered and *ndata* is the length of the signal. This approach is based on the fact that the number of coefficients half with each decomposition level. However, it did not take into account the length of the wavelet shape itself which the Matlab and Python implementation does, and thus the additional decomposition levels can be ignored.

5.0 MDOF RESPONSE AT EACH DECOMPOSED LEVEL

5.1 Overview

From **Section 4**, the total floor acceleration signals were decomposed using the DWT method. However, in order to obtain the capacity curve, the relative acceleration and displacement response are also required. This section briefly covers the calculation of the relative response.

5.2 Calculation of relative response

Let's denote X_{ij} and $\ddot{X}_{ij}$ as the j^{th} decomposition level's i^{th} floor total displacement and acceleration response, respectively, while x_{ij} and $\ddot{x}_{ij}$ are the corresponding relative response. $\ddot{X}_{ij}$ was obtained from **Section 4** using the DWT method. Knowing this information, we can simply obtain the total floor displacement as follows:

$$X_{ij}(t) = \iint \ddot{X}_{ij}(t).\,dt^2 \qquad (5\text{-}1)$$

Once X_{ij} and $\ddot{X}_{ij}$ have been obtained, the relative response can be computed as follows:

$$x_{ij}(t) = X_{ij}(t) - X_{0,j}(t) \qquad (5\text{-}2)$$

$$\ddot{x}_{ij}(t) = \ddot{X}_{ij}(t) - \ddot{X}_{0,j}(t) \qquad (5\text{-}3)$$

Note that due to the double integration process, baseline correction may be required for $X_{ij}(t)$ and/or $x_{ij}(t)$. In both the Matlab and Python implementation of this process, a simple linear detrend was adopted.

6.0 MDOF TO SDOF RESPONSE CONVERSION

6.1 Overview

This section describes the background and theory behind condensing the multi-degree-of-freedom (MDOF) response of the building down to an equivalent single-degree-of-freedom (SDOF) response.

6.2 Basic modal analysis theory

From modal analyses, the vector of relative displacements, $\{x\}$, can be represented as the sum of the s^{th} mode shape, $\{\Phi_s\}$, and amplitude, y_s, as follows:

$$\{x\} \quad = \sum_{s=1}^{M} \{\Phi\}_s . y_s = [\Phi] . \{y\}_s \tag{6-1}$$

Where $[\Phi]$ is a matrix containing the $\{\Phi_s\}$ vectors, $\{y\}_s$ is a vector containing y_s values, and M is the number of modes. Note that the s^{th} mode contribution to the relative displacements, $\{x\}_s$, can be expressed as:

$$\{x\}_s \quad = \{\Phi\}_s . y_s \tag{6-2}$$

The equation of motion for buildings subjected to seismic excitation is:

$$-[M]\{r\}\ddot{X}_g \quad = [M]\{\ddot{x}\} + [C]\{\dot{x}\} + [K]\{x\} \tag{6-3}$$

Where $[M]$ is the mass matrix, $[C]$ is the damping matrix, $[K]$ is the stiffness matrix, $\{r\}$ is the influence vector and $\ddot{X}_g$ is the total ground acceleration.

It should also be noted that the mode shapes have orthogonality properties. In other words:

$$\{\Phi\}_w^T[K]\{\Phi\}_v \quad = \{\Phi\}_w^T[M]\{\Phi\}_v = 0 \qquad \text{when } w \neq v \tag{6-4}$$

By substituting **Equation (6-1)** into **Equation (6-3)**, multiplying both sides by $[\Phi]^T$, and making use of the orthogonal properties highlighted in **Equation (6-4)**, we end up with M equation of motions as follows:

$$-\{\Phi\}_s^T[M]\{r\}\ddot{X}_g \quad = \{\Phi\}_s^T[M]\{\Phi\}_s . \ddot{y}_s + \{\Phi\}_s^T[C]\{\Phi\}_s . \dot{y}_s + \{\Phi\}_s^T[K]\{\Phi\}_s . y_s \tag{6-5}$$

and then dividing by $\{\Phi_s\}^T[M]\{\Phi_s\}$, the following equation of motion can be obtained for each mode:

$$-\beta_s\ddot{X}_g \quad = \ddot{y}_s + 2h_s\omega_s\dot{y}_s + \omega_s^2 y_s \tag{6-6}$$

Where β is the participation factor which is computed as:

$$\beta_s \quad = \frac{\{\Phi\}_s^T[M]\{r\}}{\{\Phi\}_s^T[M]\{\Phi\}_s} \tag{6-7}$$

If one denotes that $y_s = \beta_s D_s$, **Equation (6-6)** can be simplified further to:

$$-\ddot{X}_g \quad = \ddot{D}_s + 2h_s\omega_s D_s + \omega_s^2 D_s \tag{6-8}$$

Equation (6-8) is essentially the equation of motion for a single-degree-of-freedom oscillator with unit mass and a natural frequency of ω_s. When converting the MDOF response to an equivalent SDOF system, D_s will be the representative displacement of the system relative to the base of the building, while the representative total acceleration of the system is $\ddot{D}_s + \ddot{X}_g$. If the system is undamped (i.e., h_s = 0), then –($\ddot{D}_s + \ddot{x}_g$) is equivalent to the base shear of the system assuming unit mass.

6.3 Modifications to estimate representative displacements

From the previous section, we identified that D_s is the representative displacement relative to the base for an equivalent SDOF system. For the s^{th} mode of response, the relative displacements $\{x\}_s$, can be related to D_s as follows:

$$\begin{aligned} \{x\}_s \quad &= \{\Phi\}_s . y_s \\ &= \beta_s . \{\Phi\}_s . D_s \\ &= \frac{\{\Phi\}_s^T[M]\{r\}}{\{\Phi\}_s^T[M]\{\Phi\}_s} . \{\Phi\}_s . D_s \end{aligned} \tag{6-9}$$

We can make use of the relationship that $\{x\}_s/y_s = \{\Phi_s\}$ and simplify **Equation (6-9)** to:

$$\begin{aligned} \{x\}_s \quad &= \frac{(\{x\}_s^T/y_s)[M]\{r\}}{(\{x\}_s^T/y_s)[M](\{x\}_s/y_s)} . (\{x\}_s/y_s) . D_s \\ &= \frac{\{x\}_s^T[M]\{r\}}{\{x\}_s^T[M]\{x\}_s} . \{x\}_s . D_s \end{aligned} \tag{6-10}$$

Rearranging **Equation (6-10)**, we get:

$$\begin{aligned} D_s \quad &= \frac{\{x\}_s^T[M]\{x\}_s}{\{x\}_s^T[M]\{r\}} \\ &= \frac{\sum_{i=1}^{NF}\left(m_i . x_{i,s}^2\right)}{\sum_{i=1}^{NF}\left(m_i . x_{i,s}\right)} \end{aligned} \tag{6-11}$$

Where *i* represents the floor level and *NF* is the number of floors.

We can see from **Equation (6-11)** that D_s is only a function of floor mass and the s^{th} mode floor displacements relative to the ground. As such, if $\{x\}_s$ can be obtained and if we have a good estimate of floor mass, we can calculate the corresponding representative displacement of the s^{th} mode.

6.4 Modifications to estimate representative accelerations

From **Section 6.2**, we established that $\ddot{D}_s + \ddot{x}_g$ is the representative acceleration for the equivalent SDOF system corresponding to the s^{th} mode. If one assumes that the s^{th} mode SDOF system is undamped (i.e., h_s = 0), then the representative acceleration and displacement can be related by rearranging **Equation (6-8)** as:

$$-\left(\ddot{D}_s + \ddot{X}_g\right) = \omega_s^2 . D_s \quad (6\text{-}12)$$

If the system does not have unit mass, the s^{th} mode base shear demand can be computed as:

$$V_{demand,s} = M_{eff,s} . \left(\ddot{D}_s + \ddot{X}_g\right) \quad (6\text{-}13)$$

Where $M_{eff,s}$ is the effective mass of the s^{th} mode of response. Meanwhile, the s^{th} mode base shear resistance is:

$$\begin{aligned} V_{resistance,s} &= [K].\{x_s\} \quad (6\text{-}14) \\ &= \beta_s.[K].\{\Phi\}_s.D_s \\ &= \beta_s.[M].\{\Phi\}_s.\omega_s^2.D_s \\ &= -\beta_s.[M].\{\Phi\}_s.\left(\ddot{D}_s + \ddot{X}_g\right) \\ &= -\frac{\{\Phi\}_s^T[M]\{r\}}{\{\Phi\}_s^T[M]\{\Phi\}_s}.[M].\{\Phi\}_s.\left(\ddot{D}_s + \ddot{X}_g\right) \end{aligned}$$

By equating that $V_{demand,s}$ = $-V_{resistance,s}$, we can obtain $M_{eff,s}$ as:

$$\begin{aligned} M_{eff,s} &= -V_{resistance,s}/\left(\ddot{D}_s + \ddot{X}_g\right) \quad (6\text{-}15) \\ &= \frac{\{\Phi\}_s^T[M]\{r\}}{\{\Phi\}_s^T[M]\{\Phi\}_s}.[M].\{\Phi\}_s \\ &= \frac{(\{x\}_s^T/y_s)[M]\{r\}}{(\{x\}_s^T/y_s)[M](\{x\}_s/y_s)}.[M].(\{x\}_s/y_s) \\ &= \frac{\left(\sum_{i=1}^{NF}(m_i.x_{i,s})\right)^2}{\sum_{i=1}^{NF}(m_i.x_{i,s}^2)} \end{aligned}$$

To derive an expression for $\ddot{D}_s + \ddot{x}_g$, we can express the second to last line of **Equation (6-14)** as a sum of forces at each floor level and expand it as follows:

$$\begin{aligned} V_{demand,s} &= \sum_{i=1}^{NF} \left(\beta_s . m_i . \Phi_{i,s} . \left(\ddot{D}_s + \ddot{X}_g\right)\right) \\ &= \sum_{i=1}^{NF} \left(\beta_s . m_i . \Phi_{i,s} . \ddot{D}_s\right) + \sum_{i=1}^{NF} \left(\beta_s . m_i . \Phi_{i,s} . \ddot{X}_g\right) \\ &= \sum_{i=1}^{NF} \left(m_i . \ddot{x}_{i,s}\right) + \sum_{i=1}^{NF} \left(m_i . \beta_s . \Phi_{i,s} . \ddot{X}_g\right) \end{aligned} \tag{6-16}$$

From the second to last line of **Equation (6-14)**, we can see that by dividing both sides of the equation by $\sum_{i=1}^{NF}(\beta_s . m_i . \Phi_i)$, we can obtain an expression for $\ddot{D}_s + \ddot{X}_g$. The calculation is as follows:

$$\begin{aligned} \ddot{D}_s + \ddot{X}_g &= \frac{\sum_{i=1}^{NF}\left(m_i . \ddot{x}_{i,s}\right) + \sum_{i=1}^{NF}\left(m_i . \beta_s . \Phi_{i,s} . \ddot{X}_g\right)}{\sum_{i=1}^{NF}\left(m_i . \beta_s . \Phi_{i,s}\right)} \\ &= \frac{\sum_{i=1}^{NF}\left(m_i . \ddot{x}_{i,s}\right)}{\sum_{i=1}^{NF}\left(m_i . \beta_s . \Phi_{i,s}\right)} + \ddot{X}_g \\ &= \left[\frac{1}{\sum_{i=1}^{NF}\left(m_i . \beta_s . \Phi_{i,s}\right)} . \sum_{i=1}^{NF}\left(m_i . \ddot{x}_{i,s}\right)\right] + \ddot{X}_g \\ &= \left[\frac{1}{\beta_s} . \frac{1}{\sum_{i=1}^{NF}\left(m_i . \Phi_{i,s}\right)} . \sum_{i=1}^{NF}\left(m_i . \ddot{x}_{i,s}\right)\right] + \ddot{X}_g \\ &= \left[\frac{\{\Phi\}_s^T[M]\{\Phi\}_s}{\{\Phi\}_s^T[M]\{r\}} . \frac{1}{\sum_{i=1}^{NF}\left(m_i . \Phi_{i,s}\right)} . \sum_{i=1}^{NF}\left(m_i . \ddot{x}_{i,s}\right)\right] + \ddot{X}_g \\ &= \left[\frac{(\{x\}_s^T/y_s)[M](\{x\}_s/y_s)}{(\{x\}_s^T/y_s)[M]\{r\}} . \frac{1}{\sum_{i=1}^{NF}\left(m_i . \left(x_{i,s}/y_s\right)\right)} . \sum_{i=1}^{NF}\left(m_i . \ddot{x}_{i,s}\right)\right] + \ddot{X}_g \\ &= \left[\frac{\{x\}_s^T[M]\{x\}_s}{\{x\}_s^T[M]\{r\}} . \frac{1}{\sum_{i=1}^{NF}\left(m_i . x_{i,s}\right)} . \sum_{i=1}^{NF}\left(m_i . \ddot{x}_{i,s}\right)\right] + \ddot{X}_g \\ &= \left[\frac{\sum_{i=1}^{NF}\left(m_i . x_{i,s}^2\right)}{\sum_{i=1}^{NF}\left(m_i . x_{i,s}\right)} . \frac{1}{\sum_{i=1}^{NF}\left(m_i . x_{i,s}\right)} . \sum_{i=1}^{NF}\left(m_i . \ddot{x}_{i,s}\right)\right] + \ddot{X}_g \\ &= \left[\frac{\sum_{i=1}^{NF}\left(m_i . x_{i,s}^2\right)}{\left(\sum_{i=1}^{NF}\left(m_i . x_{i,s}\right)\right)^2} . \sum_{i=1}^{NF}\left(m_i . \ddot{x}_{i,s}\right)\right] + \ddot{X}_g \end{aligned} \tag{6-17}$$

From **Equation (6-17)**, we can see that $\ddot{D}_s + X_g$ is a function of floor mass, s^th mode displacements and accelerations relative to the base, and the total ground acceleration.

6.5 Applicability to nonlinear cases

Fundamental modal analysis theory was derived for linear elastically responding systems. This is reflected in the assumption that the stiffness and mass matrix are related via ω_s^2 in the derivation of equations. During inelastic response, the stiffness matrix can significantly decrease while the mass matrix stays constant, resulting in ω_s^2 changing with time. However, this issue can be circumvented by using the relative response of the building and the total ground acceleration at a specific time in the calculations directly, as derived in **Equations (6-11)** and **(6-17)**. Based on this, we can consider ω_s^2 to be the secant slope to reach the representative displacement and acceleration coordinates $D_s(t_i)$ and $\left(\ddot{D}_s(t_i) + \ddot{X}_g(t_i)\right)$ at time $t = t_i$, as shown in **Figure 6-1**. Thus, we can represent each point along the capacity curve as an equivalent linear elastic system even though nonlinear response had occurred.

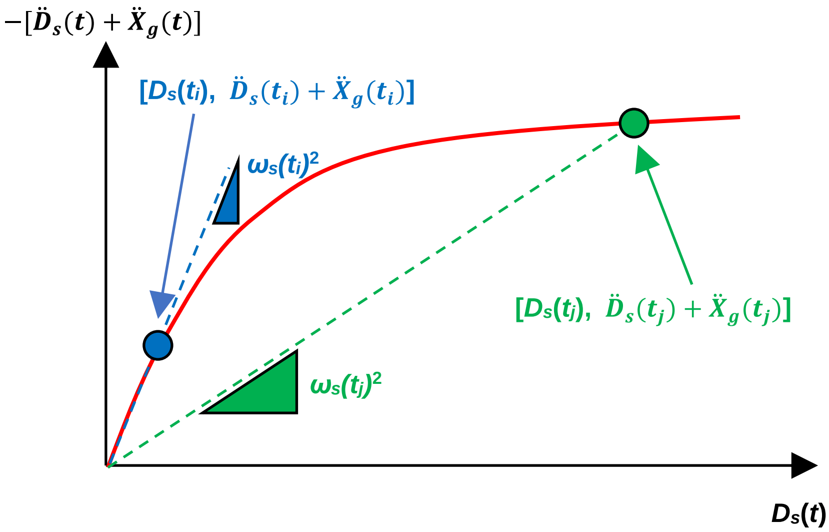


Figure 6-1. Equivalent linearization of SDOF response

6.6 Representative response for each decomposition level

From **Equations (6-11)** and **(6-17)**, the relative floor displacement and acceleration response corresponding to the s^{th} mode of response of the building is required to derive the representative displacement and acceleration response. Given that the building response was decomposed into different levels using the DWT method, rather than apply the modal analysis concepts to modes, they can be applied via decomposition levels instead to identify which decomposition levels correspond to the predominant mode of response.

In order to apply modal analysis concepts to decomposition levels instead of modes, **Equations (6-11)** and **(6-17)** need to be adjusted accordingly. Let's denote X_{ij} and $\ddot{X}_{ij}$ as the j^{th} decomposition level's i^{th} floor total displacement and acceleration response, respectively, while x_{ij} and $\ddot{x}_{ij}$ are the corresponding relative response. We can obtain $\ddot{X}_{ij}$ directly by applying DWT to recorded acceleration data as outlined in **Section 4**. For example, denoting the roof level as i = 4 for the E-Defense test floor acceleration recording shown in **Figure 4-3**, $\ddot{X}_{4,5}$ would correspond to the reconstructed signal f_5. Once $\ddot{X}_{ij}$ has been obtained, X_{ij}, x_{ij}, and $\ddot{x}_{ij}$, can be obtained from **Equations (5-1)** to **(5-3)**, respectively.

A major difference between applying the concept at a modal level versus decomposition level is the treatment of the ground acceleration. In the original formation of the equation of motion in **Equation (6-3)**, the relative response of the building was computed relative to the total ground response, X_g. As such, the total ground acceleration was used on the left-hand side of the **Equation (6-3)**, and carried all the way to **Equation (6-17)**. When using the results from DWT, all signals were decomposed, including the ground level. Thus, the relative response of the building within each decomposition level will be relative to the corresponding decomposed ground response. The difference between the two cases is shown in **Figure 6-2**.

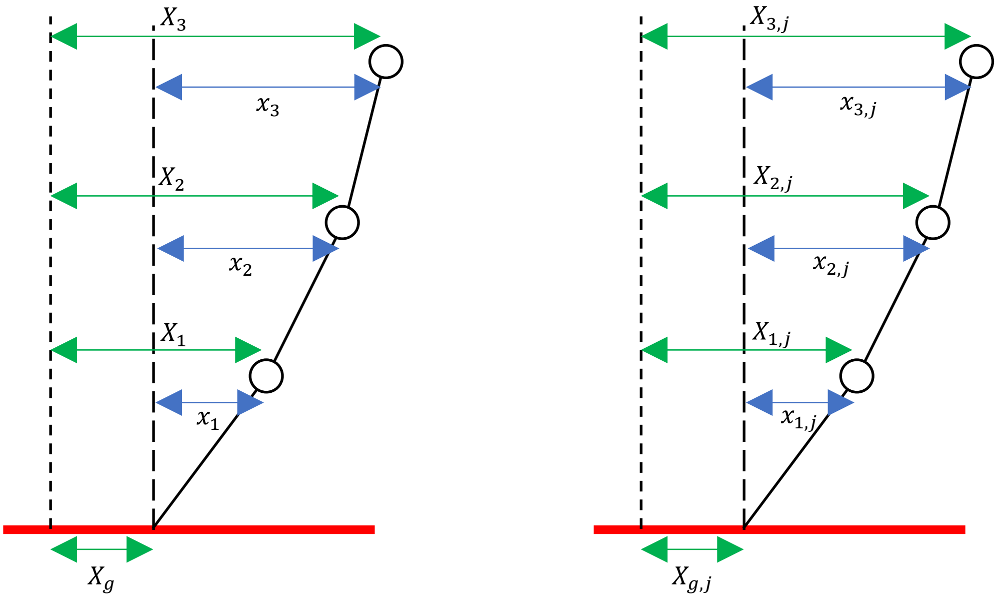


Figure 6-2. Difference in ground motion response between original signal and decomposed signal

Based on **Figure 6-2**, the equation of motion can be rewritten as:

$$-[M]\{r\}\ddot{X}_{g,j} = [M]\{\ddot{x}\}_j + [C]\{\dot{x}\}_j + [K]\{x\}_j \quad (6\text{-}18)$$

If one assumes that there is only a single mode within each decomposed level, then:

$$\{x\}_j = \{\Phi\}_j . \beta_j . D_j \quad (6\text{-}19)$$

Following the derivation steps in **Sections 6.3** and **6.4**, we can obtain the representative displacement and acceleration for each decomposition level as:

$$D_j = \frac{\sum_{i=1}^{NF}(m_i . x_{ij}^2)}{\sum_{i=1}^{NF}(m_i . x_{ij})} \qquad (6\text{-}20)$$

$$\ddot{D}_j + \ddot{X}_{g,j} = \left[\frac{\sum_{i=1}^{NF}(m_i . x_{ij}^2)}{\left(\sum_{i=1}^{NF}(m_i . x_{ij})\right)^2} . \sum_{i=1}^{NF}(m_i . \ddot{x}_{ij})\right] + \ddot{X}_{g,j} \qquad (6\text{-}21)$$

It should be noted that an alternate method could be to firstly obtain the relative floor responses and then apply DWT only to the relative response. From this approach, the undecomposed total ground acceleration $\ddot{X}_g$ would be used instead of $\ddot{X}_{gj}$. However, one reason for applying DWT to the total ground acceleration as well was to remove both high and low frequency noise components. If unfiltered, this may result in a poor estimate of relative response. Further research would be required to evaluate if applying DWT only to relative response would result in a better estimate of representative accelerations.

6.7 Large representative response near zero displacement

From **Equations (6-20)** and **(6-21)**, the relative response terms on the right-side of the equation requires a division by the sum of $m_i x_{ij}$. When the floor relative displacements are near zero, this can cause both the representative displacements and accelerations to become extremely large.

To address this issue, it was proposed by **Kusunoki et al. (2018)** that the relative terms can be multiplied by the effective mass ratio, which is the ratio between the effective mass of the j^{th} decomposition level, $M_{eff,j}$, and the total mass of the superstructure. This is termed the “tentative” response. The tentative representative displacement, $(D_j)^*$, and acceleration $(\ddot{D}_j + \ddot{X}_{g,j})^*$ and can be calculated as:

$$\begin{aligned}(D_j)^* &= \frac{\sum_{i=1}^{NF}(m_i . x_{ij}^2)}{\sum_{i=1}^{NF}(m_i . x_{ij})} . \frac{M_{eff,j}}{\sum_{i=1}^{NF} m_i} \\ &= \frac{\sum_{i=1}^{NF}(m_i . x_{ij}^2)}{\sum_{i=1}^{NF}(m_i . x_{ij})} . \frac{\left(\sum_{i=1}^{NF}(m_i . x_{ij})\right)^2}{\sum_{i=1}^{NF}(m_i . x_{ij}^2)} . \frac{1}{\sum_{i=1}^{NF} m_i} \\ &= \frac{\sum_{i=1}^{NF}(m_i . x_{ij})}{\sum_{i=1}^{NF} m_i}\end{aligned} \qquad (6\text{-}22)$$

$$\begin{aligned}(\ddot{D}_j + \ddot{X}_{g,j})^* &= \left[\frac{\sum_{i=1}^{NF}(m_i . x_{ij}^2)}{\left(\sum_{i=1}^{NF}(m_i . x_{ij})\right)^2} . \sum_{i=1}^{NF}(m_i . \ddot{x}_{ij})\right] . \frac{M_{eff,j}}{\sum_{i=1}^{NF} m_i} + \ddot{X}_{g,j} \\ &= \left[\frac{\sum_{i=1}^{NF}(m_i . x_{ij}^2)}{\left(\sum_{i=1}^{NF}(m_i . x_{ij})\right)^2} . \sum_{i=1}^{NF}(m_i . \ddot{x}_{ij})\right] . \frac{\left(\sum_{i=1}^{NF}(m_i . x_{ij})\right)^2}{\sum_{i=1}^{NF}(m_i . x_{ij}^2)} . \frac{1}{\sum_{i=1}^{NF} m_i} + \ddot{X}_{g,j} \\ &= \frac{\sum_{i=1}^{NF}(m_i . \ddot{x}_{ij})}{\sum_{i=1}^{NF} m_i} + \ddot{X}_{g,j}\end{aligned} \qquad (6\text{-}23)$$

Kusunoki et al. (2018) suggested that the tentative response is used to determine which decomposition levels contribute to the predominant mode of response and to obtain the hysteretic response. Once this was obtained, the backbone curve can be extracted. The backbone curve values can then be corrected to obtain the actual representative capacity curve as follows.

$$\begin{aligned} D_j &= \left(D_j\right)^* . \frac{\sum_{i=1}^{NF} m_i}{M_{eff,j}} \\ &= \left(D_j\right)^* . \frac{\sum_{i=1}^{NF}\left(m_i . x_{ij}^2\right)}{\left(\sum_{i=1}^{NF}\left(m_i . x_{ij}\right)\right)^2} . \sum_{i=1}^{NF} m_i \end{aligned} \tag{6-24}$$

$$\begin{aligned} \ddot{D}_j + \ddot{X}_{g,j} &= \left[\left(\ddot{D}_j + \ddot{X}_{g,j}\right)^* - \ddot{X}_{g,j}\right] . \frac{\sum_{i=1}^{NF} m_i}{M_{eff,j}} + \ddot{X}_{g,j} \\ &= \left[\left(\ddot{D}_j + \ddot{X}_{g,j}\right)^* - \ddot{X}_{g,j}\right] . \frac{\sum_{i=1}^{NF}\left(m_i . x_{ij}^2\right)}{\left(\sum_{i=1}^{NF}\left(m_i . x_{ij}\right)\right)^2} . \sum_{i=1}^{NF} m_i + \ddot{X}_{g,j} \end{aligned} \tag{6-25}$$

7.0 DECOMPOSITION LEVEL SELECTION

7.1 Overview

Once the tentative representative displacement and acceleration response of each decomposition level had been obtained following **Section 6.7**, the properties of each tentative acceleration-displacement relationship are used to select the decomposition levels which contribute to the predominant mode of response. This section will provide a summary of how this process is done. Further information on the calibration of the methodology can be found in **Yeow et al. (2022)**.

7.2 Key parameters calculation

The first step is to calculate the value of key parameters for each individual decomposition level response. There are five key parameters:

- Peak absolute tentative representative displacement,
- Peak absolute tentative representative acceleration,
- Mean effective mass ratio,
- Best-fit linear slope to tentative representative acceleration-displacement response, and
- Cumulative kinetic energy.

The peak absolute tentative representative displacement and acceleration of each individual decomposition level response are computed as follows:

$$\left(D_j\right)^*_{max} = max\left|\left(D_j\right)^*\right| \tag{7-1}$$

$$\left(\ddot{D}_j + \ddot{X}_{g,j}\right)^*_{max} = max\left|\left(\ddot{D}_j + \ddot{X}_{g,j}\right)^*\right| \tag{7-2}$$

The mean effective mass ratio cannot simply be calculated considering the entire length of the signal. This is because in cases where there is a long duration of ambient noise before and after the strong portion of the shaking, the mean effective mass ratio may be more reflective of the building's response during those portions rather than during the strong component of the earthquake. Instead, we firstly calculated the cumulative Arias Intensity, I_A, throughout the entire signal shown in **Equation (7-3)**, where t_d is the duration of the signal. Then, the times corresponding to 5% and 75% of maximum I_A, t_1 and t_2, respectively, as shown in **Equations (7-4)** and **(7-5).** Finally, the mean effective mass ratio was calculation considering the duration range between t_1 and t_2 following **Equation (7-6)**. Note that for the implementation of **Equation (7-3)** in the functions, only the term in the integral was considered. This is because the terms outside the integral are constant, so t_1 and t_2 would correspond to the same time regardless of whether the constants were included in the calculation.

$$I_{A,j}(t) = \frac{\pi}{2g}\int_0^{t_d}\left(\ddot{X}_{0,j}(t)\right)^2 dt \quad (7\text{-}3)$$

$$I_{A,j}(t_1) = 0.05 I_{A,j}(t) \quad (7\text{-}4)$$

$$I_{A,j}(t_2) = 0.75 I_{A,j}(t) \quad (7\text{-}5)$$

$$\begin{aligned} M_{ratio,j} &= \frac{\int_{t=t_1}^{t_2} M_{eff,j}(t)\,.\,dt}{\sum_{i=1}^{NF} m_i.\,(t_2 - t_1)} \quad (7\text{-}6) \\ &= \frac{\int_{t=t_1}^{t_2}\left(\frac{\left(\sum_{i=1}^{NF}\left(m_i.\,x_{i,j}(t)\right)\right)^2}{\sum_{i=1}^{NF}\left(m_i.\,x_{i,j}^2(t)\right)}\right).\,dt}{\sum_{i=1}^{NF} m_i.\,(t_2 - t_1)} \end{aligned}$$

The best-fit linear slope was computed using in-built first-degree “polyfit” functions in Matlab and Python. This is essentially a linear regression, which can be performed manually using the following equation, where $K_{fit,j}$ is the best-fit linear slope for the j^{th} decomposition level response, k represents the k^{th} data point and *ndata* is the total number of datapoints.

$$K_{fit,j} = -\frac{\sum_{k=1}^{ndata}\left[\left(\left(D_j\right)_k^* - mean\left(D_j\right)^*\right).\left(\left(\ddot{D}_j + \ddot{X}_{g,j}\right)_k^* - mean\left(\ddot{D}_j + \ddot{X}_{g,j}\right)^*\right)\right]}{\sum_{k=1}^{ndata}\left(\left(D_j\right)_k^* - mean\left(D_j\right)^*\right)^2} \quad (7\text{-}7)$$

The final parameter is the cumulative kinetic energy, C_{ekj}. To derive this value, we require the representative relative velocity of the system. This can be obtained by differentiating the representative displacements as follows:

$$\begin{aligned} \dot{D}_j(t) &= \frac{d\left(D_j(t)\right)}{dt} \quad (7\text{-}8) \\ &= \frac{d\left(\left(D_j(t)\right)^*.\frac{\sum_{i=1}^{NF} m_i}{M_{eff,j}(t)}\right)}{dt} \end{aligned}$$

The issue with **Equation (7-8)** is that $M_{eff,j}(t)$ can be near-zero are small relative floor displacements, which can result in significantly large values of representative velocity. However, if we make a crude assumption that that the mass ratio does not vary significantly and can be approximated by $M_{ratio,j}$, we can simplify the equation as follows:

$$\dot{D}_j(t) = \frac{d\left(\left(D_j(t)\right)^*.\frac{1}{M_{ratio,j}}\right)}{dt} = \frac{1}{M_{ratio,j}}.\frac{d\left(\left(D_j(t)\right)^*\right)}{dt} = \frac{1}{M_{ratio,j}}.\left(\dot{D}_j(t)\right)^* \quad (7\text{-}9)$$

The above equation allows for the representative velocity to be approximated by the mean mass ratio and the tentative representative velocity. The latter is easier to obtain since the tentative representative displacements, $(D_j)^*$, are used in the evaluations rather than the actual representative displacement, D_j.

The kinetic energy at a given point in time for an equivalent SDOF system is:

$$E_k(t) = \frac{1}{2}M_{eff,j}(t).\left(\dot{D}_j(t)\right)^2 \quad (7\text{-}10)$$

We can simplify this by again assuming that the effective mass does not vary significantly with time, and can thus be represented by the product of $M_{ratio,j}$ and the sum of floor masses. By making this assumption and substituting **Equation (7-9)** into **Equation (7-10)**, we obtain:

$$E_k(t) = \frac{1}{2}M_{ratio,j}.\sum_{i=1}^{NumF} m_j.\left(\frac{1}{M_{ratio,j}}.\left(\dot{D}_j(t)\right)^*\right)^2 = \frac{\sum_{i=1}^{NumF} m_j}{2}.\frac{\left(\left(\dot{D}_j(t)\right)^*\right)^2}{M_{ratio,j}}. \quad (7\text{-}11)$$

Thus, the cumulative kinetic energy is calculated as:

$$C_{ekj} = \int_{t=0}^{t_d} E_k(t).dt = \frac{\sum_{i=1}^{NumF} m_i}{2}.\frac{\int_{t=0}^{td}\left(\left(\dot{D}_j(t)\right)^*\right)^2.dt}{M_{ratio,j}}. \quad (7\text{-}12)$$

Unlike the calculation for $M_{ratio,j}$, there is no need to limit the integral duration to between t_1 and t_2. This is because the relative velocity before t_1 and after the system reaches steady-state response becomes near-zero, and thus would have little bearing on the actual cumulative kinetic energy. Furthermore, the calibration of the decomposition level selection was based on the entire signal duration. If the user chooses to shorten the integral duration to between t_1 and t_2, they will need to check whether additional calibration is required. Also, note that since the sum of floor masses and the ½ factor is the same regardless of which decomposition level was computed, these terms could be excluded in the calculations. This is because the actual value of C_{ekj} is not of importance. Rather, the relative size of the cumulative energies is essential to determine which decomposition levels to select to reconstruct the signal.

7.3 Selection of initial decomposition level

Once the key parameters were determined, the next step is to select an initial decomposition level. This initial decomposition level should be part of the final combination and will be used to determine which other decomposition levels should be selected or excluded from the final combination.

Yeow et al. (2022) proposed the following steps:

1) Normalize $\left(\ddot{D}_j + \ddot{X}_{g,j}\right)^*_{max}$ by the largest value out of all decomposition levels,
2) Disqualify all decomposition level with a normalized $\left(\ddot{D}_j + \ddot{X}_{g,j}\right)^*_{max}$ less than 0.25 from being selectable as the initial decomposition level,
3) Out of the remaining decomposition levels, select the one with the largest C_{ekj} as the initial decomposition level.

Steps 1 and 2 were performed to remove decomposition levels which correspond to long-period noise components. These components typically have large relative displacements which could result in large C_{ekj}, which could interfere with step 3. However, these have very small accelerations, and thus can be eliminated from by applying the tolerance in step 2.

Step 3 was performed to ensure that the initial decomposition level does not correspond to higher modes. This is because higher modes generally have smaller relative velocities, and thus would have smaller C_{ekj}. Since the long-period noise components were already removed, the decomposition level with the largest C_{ekj} would correspond to the predominant mode of response.

7.4 Selection of highest decomposition level

When selecting the highest decomposition level to be included when reconstructing the signal, we can make use of the property that long-period noise components would have small accelerations. Furthermore, if the best-fit slope is too different from that of the initial decomposition level, then it is unlikely to be representative of the predominant mode of response. Based on these concepts, **Yeow et al. (2022)** proposed that the highest decomposition level which satisfies the following condition will be selected as the highest decomposition level to be included in the final combination:

$$0.01 \leq \frac{K_{fit,j}.\left(\ddot{D}_j + \ddot{X}_{g,j}\right)^*_{max}}{K_{fit,initial}.\left(\ddot{D}_{initial} + \ddot{X}_{g,initial}\right)^*_{max}} \tag{7-13}$$

7.5 Selection of lowest decomposition level

To determine the lowest decomposition level to select for the final combination, we can make use of the fact that higher rank displacement response would likely be smaller than that of the predominant mode of response. Furthermore, the effective mass of higher modes would likely be smaller than that of the predominant mode of response. Based on these factors, **Yeow et al. (2022)** proposed that the lowest decomposition level which satisfies the following two conditions is selected as the lowest decomposition level to add to the final combination:

$$0.05 \leq \frac{(D_j)^*_{max}}{(D_{initial})^*_{max}} \tag{7-14}$$

$$0.65 \leq \frac{M_{ratio,j}}{M_{ratio,initial}} \tag{7-15}$$

7.6 Reconstruction of floor response

Denoting j_{max} and j_{min} as the highest and lowest decomposition level identified from **Sections 7.4** and **7.5**, respectively, the combined floor total acceleration, relative acceleration and relative displacements can be obtained as:

$$\ddot{X}_{i,combined}(t) = \sum_{j=jmin}^{jmax} \ddot{X}_{ij}(t) \tag{7-16}$$

$$\ddot{x}_{i,combined}(t) = \sum_{j=jmin}^{jmax} \ddot{x}_{ij}(t) \tag{7-17}$$

$$x_{i,combined}(t) = \sum_{j=jmin}^{jmax} x_{ij}(t) \tag{7-18}$$

Note that total floor displacements are not required in any representative response calculation and can thus be ignored.

8.0 DERIVATION OF HYSTERETIC RESPONSE

8.1 Overview

At the end of **Section 7**, we have obtained the reconstructed signals considering only the predominant mode of response. From this, we can obtain the hysteretic response and extract out the capacity curve.

8.2 Tentative hysteretic response

To obtain the tentative hysteretic response, one simply needs to reapply **Equations (6-22)** and **(6-23)**. However, instead of using the response from the j^{th} decomposition level, the combined response from *jmin* to *jmax* calculated from **Equations (7-16)** to **(7-18)** are used instead.

It should be noted that the tentative hysteretic response was used instead of the actual values due to the issue of large values are near-zero relative displacement response still being present.

8.3 Capacity curve extraction

The capacity curve can be extracted two ways. In the first approach, the tentative hysteretic response is first derived following **Section 8.2**. Afterwards, the following algorithm was used:

1) Set initial tentative representative displacement and acceleration of the capacity curve to be equal to that of the first datapoint of the tentative hysteretic response,
2) Move to the next datapoint,
3) If tentative representative displacement corresponding to the next datapoint is either smaller than the smallest displacement or larger than the largest displacement recorded for the capacity curve, add the datapoint to the capacity curve,
4) Repeat steps (2) and (3) until all datapoints have been checked,
5) Sort the extracted datapoints based on tentative representative displacements in ascending order,
6) Apply **Equations (6-24)** and **(6-25)** to obtain the actual representative displacement and accelerations, respectively.

Note that in some cases, step 6 may cause the actual representative displacements to be out of order. Outlier datapoints should be removed. However, it may be difficult to judge which datapoint is the outlier. This issue is addressed in the second approach, where the actual representative hysteretic response was considered explicitly as follows:

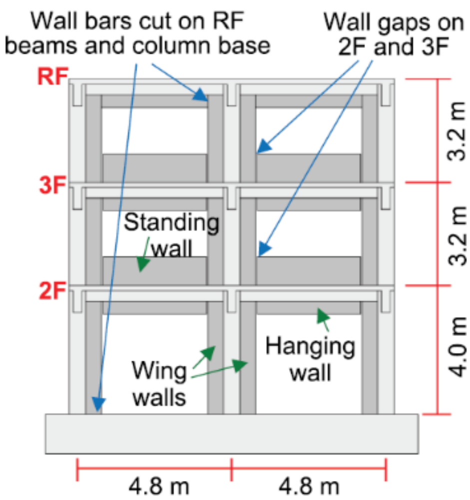
1) Apply **Equations (6-20)** and **(6-21)** to obtain the actual representative displacement and acceleration hysteretic response,
2) Set the initial tentative representative displacement and acceleration of the capacity curve to be zero,
3) Check the first datapoint,
4) If the effective mass ratio of the datapoint is less than 0.5, move to the next datapoint. Otherwise, move to step 5.
5) If the representative displacement corresponding to the next datapoint is either smaller than the smallest displacement or larger than the largest displacement recorded for the capacity curve, add the datapoint to the capacity curve then move to the next datapoint,
6) Repeat steps (4) to (5) until all datapoints have been checked,
7) Sort the extracted datapoints based on representative displacements in ascending order,

While this approach addresses the issue of determining the outliner datapoints, it does introduce an extra criterion than any datapoint with an effective mass ratio less than 0.5 should be eliminated. This was based on experience with applying this methodology rather than via a rigorous study and may require some further verification. Also, this approach assumes that the initial displacement and acceleration starts at 0. This is reasonable from the point of view of evaluate building damage relative to the initial condition of the building prior to applying the excitation. In cases where the building had received prior damage, the entire representative displacement response history would need to be adjusted accordingly.

9.0 DETAILED EXAMPLE

9.1 Example details

In this example, we will consider the first application of the 150%-scaled excitation used in the 2019 E-Defense test of a disaster management center. The structure is shown in **Figure 9-1** while the recorded floor accelerations are shown in **Figure 9-2**. This example will use the Matlab implementation of the capacity curve extraction methodology.



(a)

(b)

Figure 9-1. 2019 E-Defense test specimen; (a) layout of in-plane frame, and (b) photo

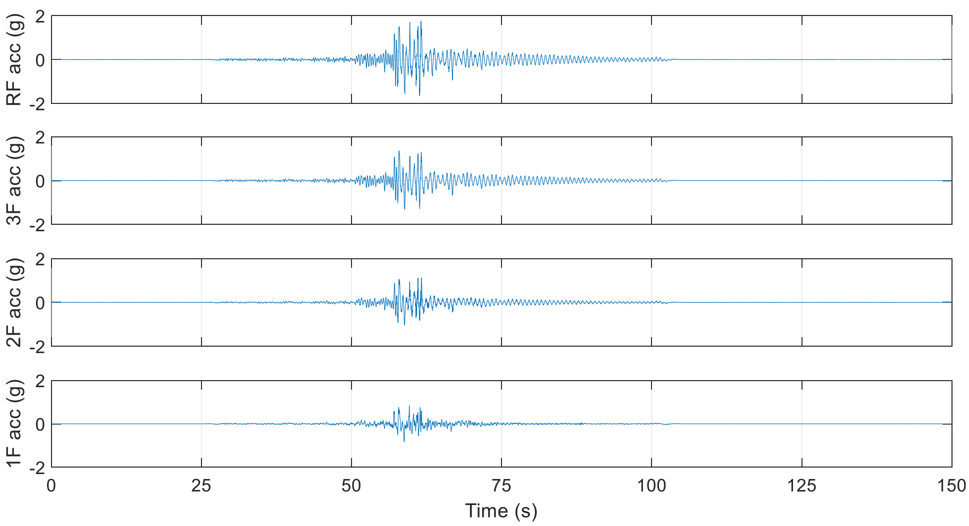


Figure 9-2. Floor acceleration recording to 150%-1 excitation

9.2 Call function and inputs

There are two required inputs and several optional inputs for the Matlab and Python implementation call functions.

The required inputs are the total floor acceleration data and the timestep. The total floor acceleration data for this example is stored in the "ALAB-Center-150-1.txt" file. It has four columns representing the recorded data at 1F, 2F, 3F and RF going from the left-most column to the right-most column. The columns were separated using a tab delimiter. Each row represents the recording data step. This format matches that required by the Matlab and Python implementations so no further modification to the formatting was required. However, the unit of the data is in gals (cm/s^2), so a 1/981 conversion was required since the implementation assumes that the unit of acceleration is in g. This data was imported using the following commands:

```
Matlab >>    TotalAcc = importdata('ALAB-Center-150-1.txt')/981
Python >>    TotalAcc = numpy.loadtxt('ALAB-Center-150-1.txt',
                             delimiter='\t')/981
```

The timestep was 0.01 s and was constant throughout the recording. The Matlab and Python implementations do not currently consider cases with inconsistent timesteps. This was assigned to variable *dT* as follows:

```
dT = 0.01
```

While there are numerous optional inputs, the two of greatest interest are the floor masses and the mother wavelet shape. The weight of 2F, 3F and RF were 740 kN, 720 kN and 520 kN, respectively. While this could be converted to mass by dividing everything by 9.81, it is not necessary. This is because in equations for calculating the representative response in **Section 6** and for the effective mass ratio in **Section 7**, the terms which uses mass data always has it present in both the numerator and denominator. This means that the actual value of mass itself is not of importance, but rather that the mass distribution is correct. These values were thus assigned to variable *Mass* as follows:

Mass = [0, 740, 720, 520]   Note: *Mass* = [0 , 740, 720, 520]/9.81 or [0, 1, 0.97, 0.70] or any other combination with identical proportion of masses with floor height are also acceptable

The mother wavelet shape was set to the 10th order of the Symlet wavelet. This was because the calibration of the decomposition level selection criteria by **Yeow et al. (2022)** was derived based on this criterion. Thus, *wname* is assigned as follows:

wname = *'sym10'*

All other optional inputs are for plotting purposes or exporting data and are thus not discussed here. Please refer to the sample script provided with the Matlab and Python implementations (named "testED2019.m" and "main.py", respectively).

To call the functions, the following was specified:

Matlab >>
```
SelectedRanks = capcurvemaster(TotalAcc,dT,'Mass',Mass,…
            'wname',wname);
```

Python >>
```
SelectedRanks = capcurvemaster(TotalAcc, dT, Mass=Mass,
            Wavelet=wname);
```

9.3 Calculation steps

The following section describes each calculation step in further detail. Note that while both Matlab and Python implementations produce similar results, the discrete wavelet transform calculation differs slightly which results in different values during intermediate steps. For the following calculation steps, the outputs from Matlab will be described.

Step 1: Applying DWT

The first step in the capacity curve extraction methodology is to apply DWT to the total floor acceleration signals from **Figure 9-2**. The "wmaxlev" function was used to determine that the maximum number of decomposition levels which can be considered is 10. Based on this, the "wavedec" function was applied to each floor acceleration signal using the "sym10" wavelet and considering 10 decomposition levels. The resulting decomposed total floor accelerations are shown in **Figure 9-3**.

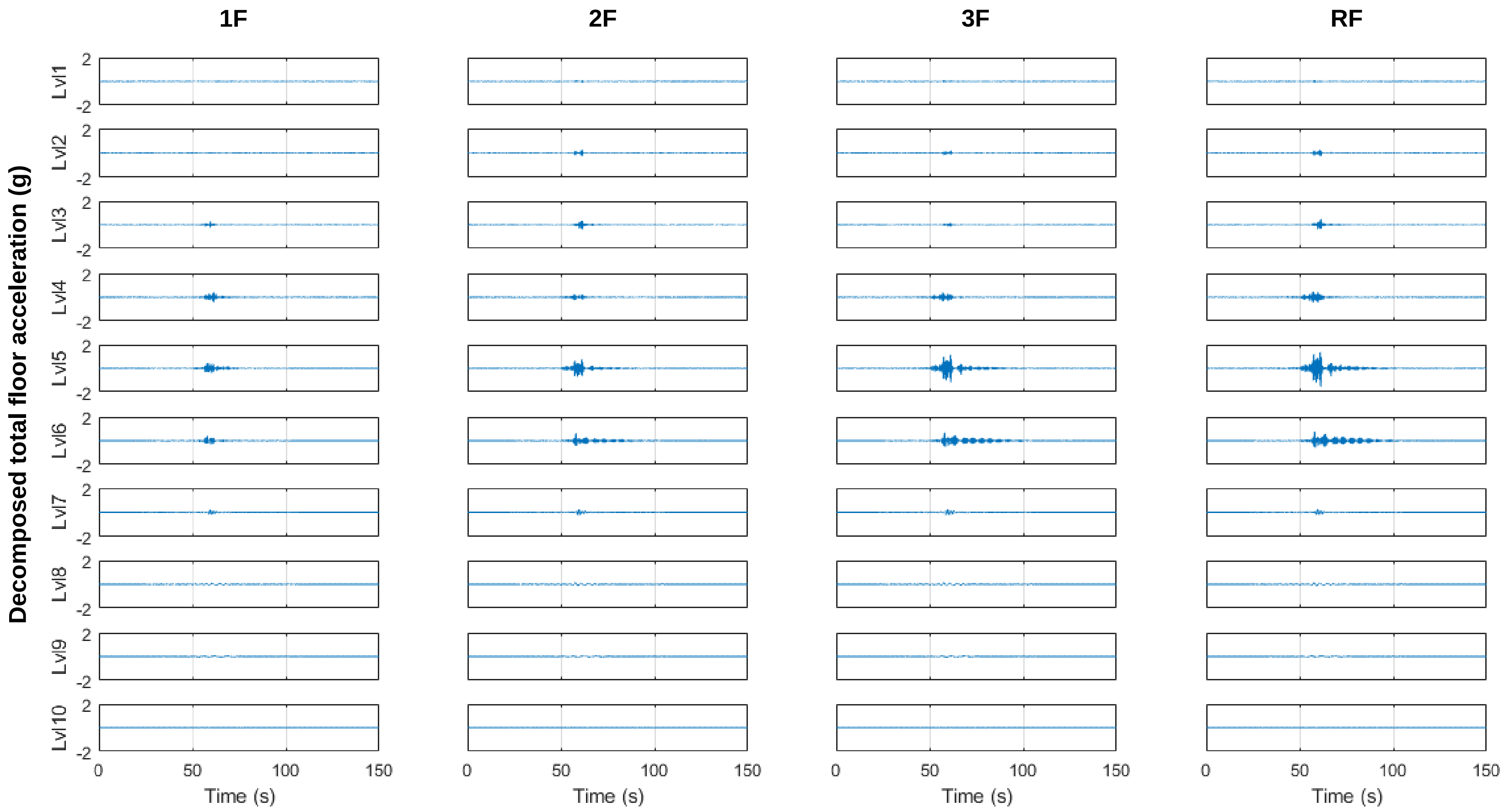


Figure 9-3. Decomposed total floor accelerations

Step 2: Obtaining relative MDOF response

The next step was to obtain the relative MDOF response. This was done by applying the “cumtrapz” function twice to each decomposed total floor acceleration signal to obtain the total floor displacements and multiplying the resultant signals by the square of the time interval between each datapoint. The baseline was corrected by applying a linear detrend. The total floor displacement and acceleration response from 1F was then subtracted from all other floors to obtain the relative responses. The relative displacement and acceleration floor response are shown in **Figure 9-4** and **Figure 9-5**, respectively.

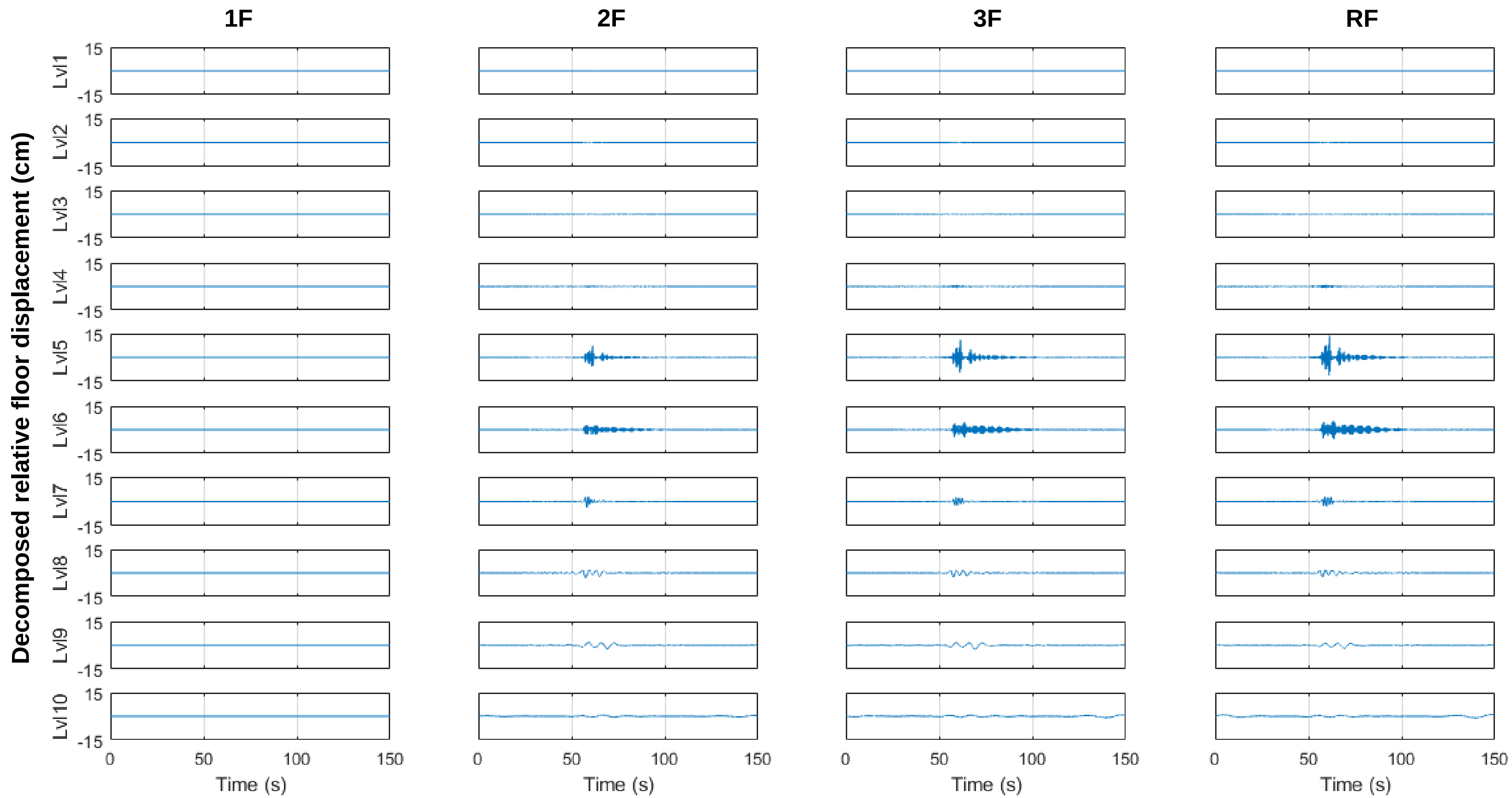


Figure 9-4. Decomposed relative floor displacement

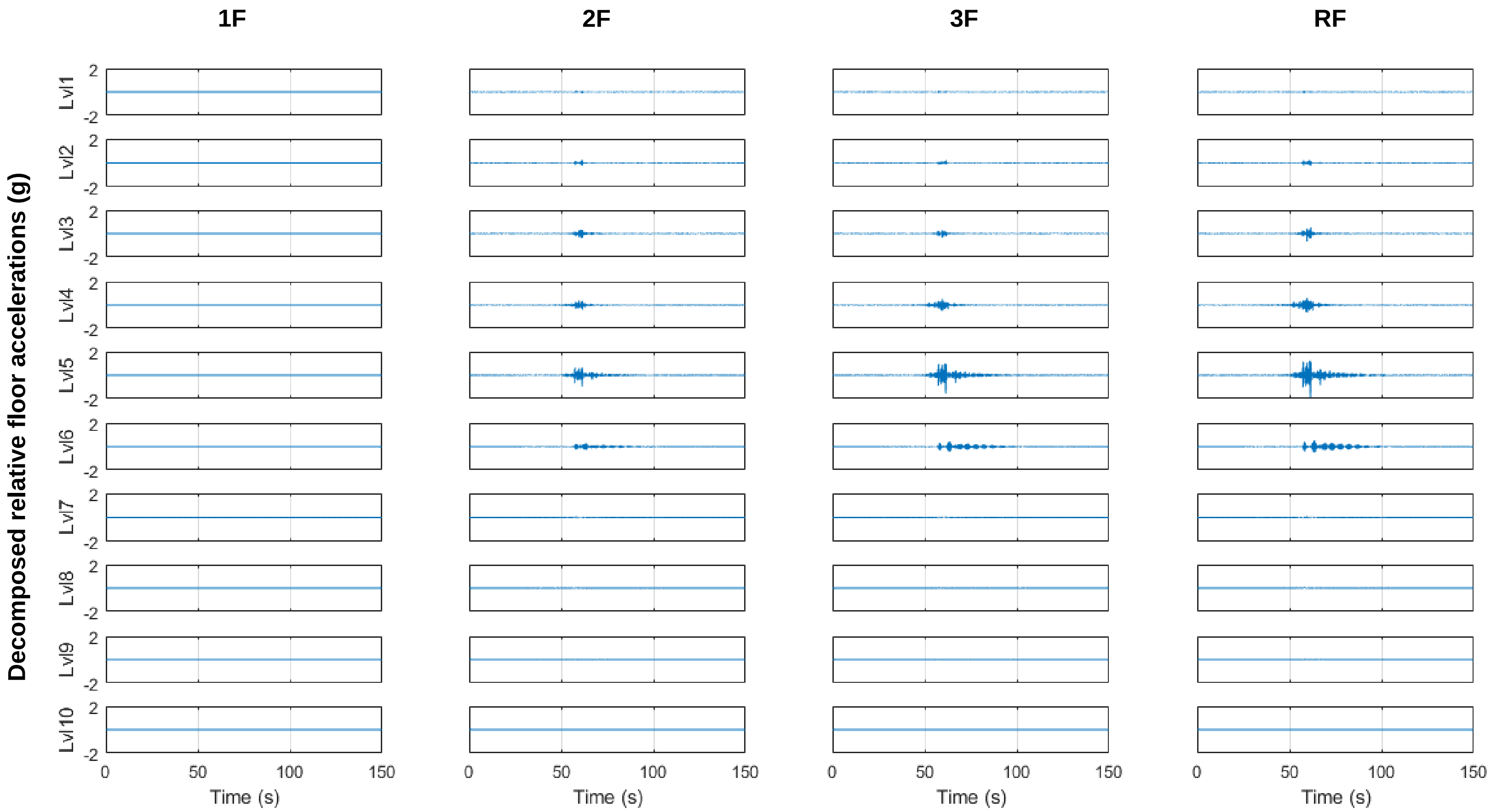


Figure 9-5. Decomposed relative floor accelerations

Step 3: Decomposition level representative response

Once the relative MDOF response had been obtained for each decomposition level, the tentative representative response can be calculated using **Equations (6-24)** and **(6-25)**. The resulting tentative responses are shown in **Figure 9-6**.

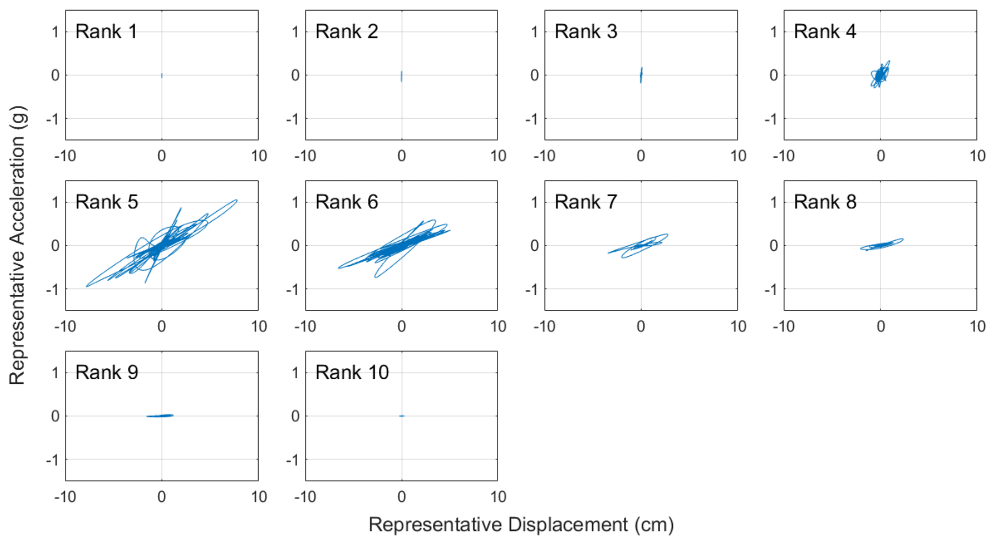


Figure 9-6. Tentative representative acceleration-displacement response for each decomposition level

Step 4: Calculation of key parameters

The peak representative displacement, peak absolute representative acceleration, mean effective mass ratio, best-fit linear slope and cumulative kinetic energy for each decomposition level was calculated using **Equations (7-1)**, **(7-2)**, **(7-6)**, **(7-7)** and **(7-12)**, respectively. The resulting values are shown in the following table:

Table 9-1. Key parameters for each decomposition level

Rank	$\left(D_j\right)^*_{max}$	$\left(\ddot{D}_j + \ddot{X}_{g,j}\right)^*_{max}$	$M_{ratio,j}$	$K_{fit,j}$	C_{ekj}
1	0.00	0.07	0.30	42.974	0.0
2	0.01	0.16	0.41	5.786	0.4
3	0.11	0.19	0.72	0.925	9.5
4	0.98	0.34	0.80	0.192	236.4
5	7.82	1.05	0.82	0.124	4952.0
6	6.56	0.73	0.82	0.093	4473.0
7	3.44	0.28	0.88	0.067	123.7
8	2.38	0.16	0.87	0.046	29.6
9	1.59	0.03	0.51	0.008	8.8
10	0.30	0.01	0.80	0.006	0.2

Step 5: Selection of decomposition levels

The steps outlined in **Sections 7.3** to **7.5** will be followed to determine the decomposition levels which should be combined to reconstruct the signal. The first step is to identify the initial decomposition level which forms the basis for the selection methodology. Firstly, the peak representative accelerations were normalized by the largest overall. This is shown in **Table 9-2**. From here, decomposition levels 1-3 and 8-10 are immediately excluded from consideration for the initial decomposition level since its normalized value was less than 0.25. Of the remaining levels, the 5^{th} decomposition level had the largest cumulative kinetic energy. As such, the 5^{th} decomposition level was selected as the initial decomposition level. In this case, the 5^{th} level had both the largest representative acceleration and cumulative kinetic energy. However, this will not always be the case, especially if the building has significant higher-mode response.

Table 9-2. Determining initial decomposition level

Rank	$\left(\ddot{D}_j + \ddot{X}_{g,j}\right)^*_{max}$	Normalized values	C_{ekj}
1	0.07	0.07	0.0
2	0.16	0.15	0.4
3	0.19	0.18	9.5
4	0.34	0.32	236.4
5	1.05	1.00	4952.0
6	0.73	0.70	4473.0
7	0.28	0.27	123.7
8	0.16	0.15	29.6
9	0.03	0.03	8.8
10	0.01	0.01	0.2

The second step was to identify the highest rank to include in the final combination. Firstly, the peak representative accelerations were multiplied by the best-fit linear slope. The product is then normalized by that from the initial decomposition level (i.e., the 5th level). The corresponding values are shown in **Table 9-3**. From there, the 8th decomposition level is the highest level where the normalized value was greater than 0.01 and is thus the highest decomposition level considered in the final combination following the criteria in **Equation (7-13)**.

Table 9-3. Determining highest decomposition level

Rank	$\left(\ddot{D}_j + \ddot{X}_{g,j}\right)^*_{max}$	$K_{fit,j}$	$\left(\ddot{D}_j + \ddot{X}_{g,j}\right)^*_{max}.K_{fit,j}$	Normalized by 5th level
5	1.05	0.124	0.131	1.00
6	0.73	0.093	0.068	0.52
7	0.28	0.067	0.019	0.14
8	0.16	0.046	0.007	0.06
9	0.03	0.008	0.000	0.00
10	0.01	0.006	0.000	0.00

The final step is to identify the lowest rank to add to the final combination. Based on the conditions outlined in **Equations (7-14)** and **(7-15)**, the lowest decomposition level with a normalized displacement and mean effective mass ratio equal to or greater than 0.05 and 0.65, respectively, is the lowest decomposition level to add into the final combination. Based on the normalized values shown in **Table 9-4**, this would correspond to the 4th decomposition level. Based on the findings from **Table 9-3** and **Table 9-4**, the 4th to the 8th decomposition level will be used to reconstruct the floor response signals.

Table 9-4. Determining lowest decomposition level

Rank	$\left(D_j\right)^*_{max}$	Normalized $\left(D_j\right)^*_{max}$	$M_{ratio,j}$	Normalized $M_{ratio,j}$
1	0.00	0.00	0.30	0.37
2	0.01	0.00	0.41	0.49
3	0.11	0.01	0.72	0.87
4	0.98	0.13	0.80	0.97
5	7.82	1.00	0.82	1.00

Step 6: Signal reconstruction

Equations (7-16), **(7-17)** and **(7-18)** were applied to reconstructed the signal considering the 4th to the 8th decomposition levels. The resulting floor total accelerations, relative accelerations and relative displacement signals are shown in **Figure 9-7** to **Figure 9-9**, respectively. The original signal is also shown in **Figure 9-7**, where the waveforms are similar though the reconstructed signal has smaller amplitude in some regions due to parts of the waveform being filtered out via the earlier steps.

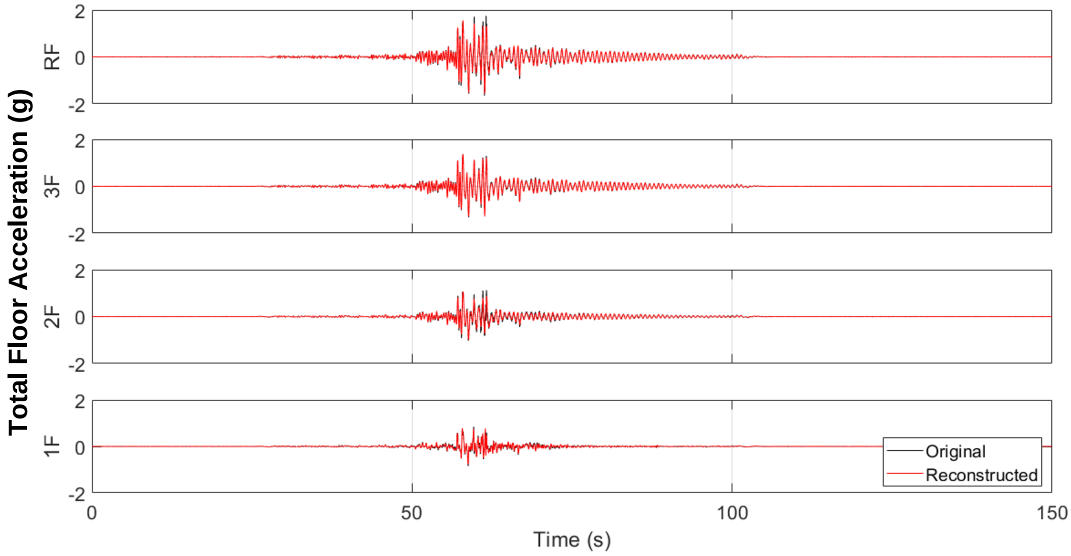


Figure 9-7. Reconstructed total floor accelerations

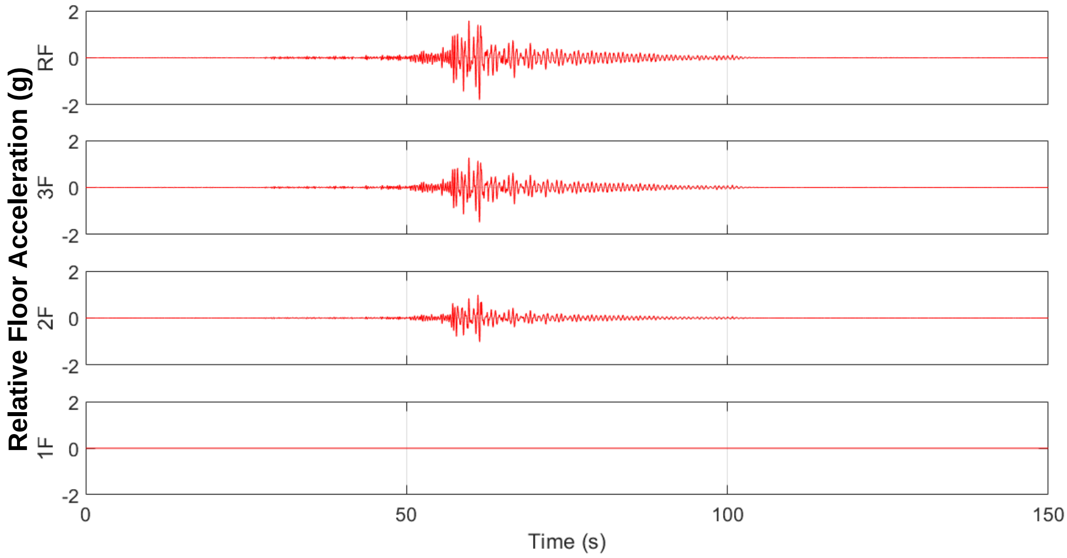


Figure 9-8. Reconstructed relative floor accelerations

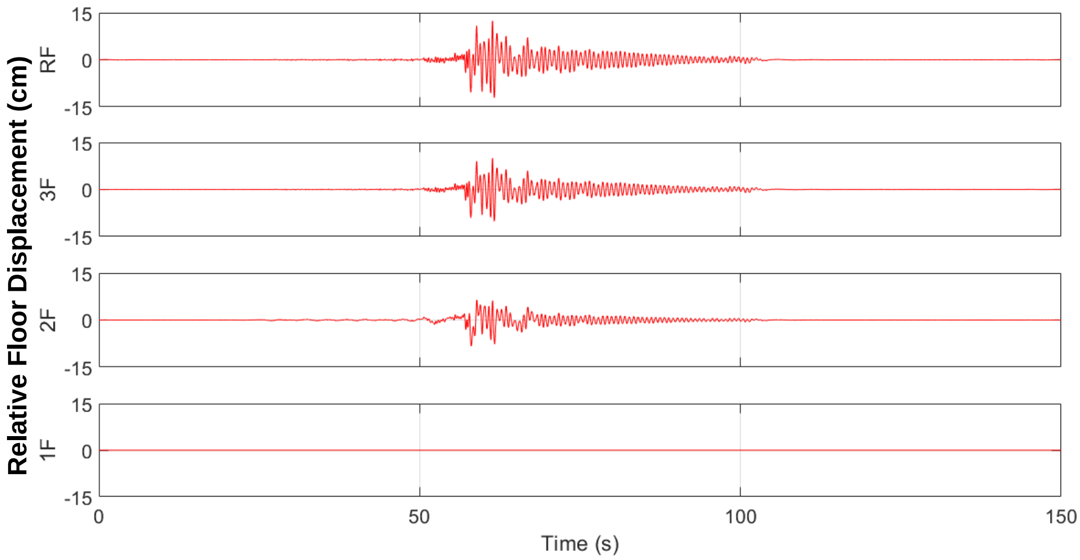


Figure 9-9. Reconstructed relative floor displacements

Step 7: Derivation of tentative hysteresis curve

Equations (6-22) and **(6-23)** were reapplied considering the reconstructed floor responses from step 6. The resulting tentative hysteretic curve is shown in **Figure 9-10** as the "applied methodology" curve. This is compared against the "experimental data" curve. The latter was derived by using accelerations recorded from a different accelerometer located near that used to derive the capacity curve and from displacements obtained via laser transducer readings. Higher mode effects were removed using the continuous wavelet transform with a cut-off frequency of 7 Hz. While not a perfect match, both curves show very similar response.

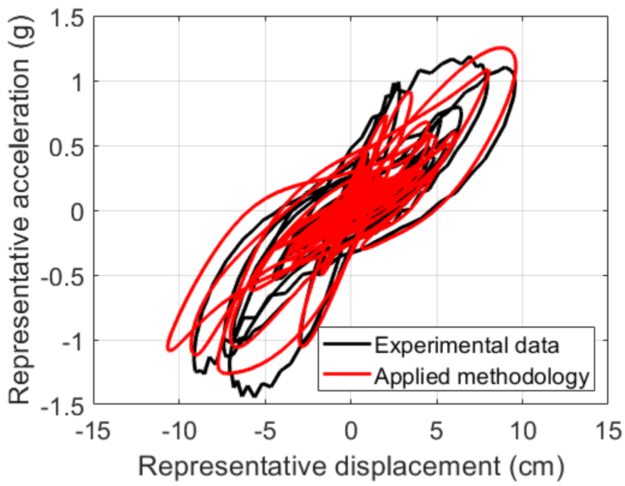


Figure 9-10. Tentative hysteretic response

Step 8: Derivation of tentative hysteresis curve

Following **Section 8.3**, the capacity curve was extracted from the hysteretic response. This is shown in **Figure 9-11** and was again compared against that extracted from experimental results. Similar to previous, both curves were not a perfect match, but are similar enough for assessing building damage.

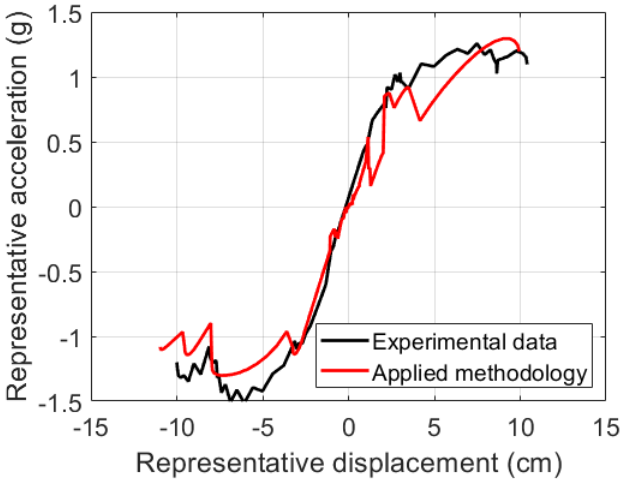


Figure 9-11. Extracted capacity curve

9.4 Validation

In addition to the validations shown in steps 7 and 8 in **Section 9.3**, The capacity curve extraction methodology was validated against large-scale shake-table tests performed at the E-Defense facility in Hyogo, Japan. These have been published by **Yeow et al. (2022)**.

10.0 REFERENCES

Kusunoki K, Hinata D, Hattori Y and Tasai A. (2018). *A new method for evaluating the real-time residual seismic capacity of existing structures using accelerometers: structures with multiple degrees of freedom*. **Japan Architectural Review**, 1(1), 77-86.

Kusunoki K. (2020). *Experimental study on the damage classification method with the performance curves of R/C structures derived from recorded accelerations.* **Journal of Structural and Construction Engineering** (Transactions of AIJ), 85(774), 1055-1065, (in Japanese).

Wang Z, Kusunoki K, Teshigawara M, Suwada H and Sakashita M. (2021). *Estimation and verification of the yield deformation of reinforced concrete members*. **Engineering structures**, 249, article 113252.

Yeow TZ, Kusunoki K, Nakamura I, Hibino Y, Fukai S and Safi WA. (2021). *E-Defense shake-table test of a building designed for post-disaster functionality*. **Journal of Earthquake Engineering**, published online.

Yeow TZ and Kusunoki K. (2022). *Unbiased rank selection for automatic hysteretic response extraction of RC frame buildings using acceleration recordings for post-earthquake safety evaluations*. **Earthquake Engineering & Structural Dynamics**, 51(3), 515-536.