COMILLAS
UNIVERSIDAD PONTIFICIA
ICAI

MÁSTER UNIVERSITARIO EN INGENIERÍA INDUSTRIAL

TRABAJO FIN DE MÁSTER

ALTERNATIVES TO THE SINE TEST IN SATELLITE VERIFICATION

Autor: Marcos Feria Cerrada

Director: Miguel Such Taboada

Co-Director: Alberto Carnicero López

Madrid

Agosto de 2019

AUTORIZACIÓN PARA LA DIGITALIZACIÓN, DEPÓSITO Y DIVULGACIÓN EN RED DE PROYECTOS FIN DE GRADO, FIN DE MÁSTER, TESINAS O MEMORIAS DE BACHILLERATO

1º. Declaración de la autoría y acreditación de la misma.

El autor D. **Marcos Feria Cerrada**____

DECLARA ser el titular de los derechos de propiedad intelectual de la obra: **Alternatives to the sine test in satellite verification**_____________________, que ésta es una obra original, y que ostenta la condición de autor en el sentido que otorga la Ley de Propiedad Intelectual.

2º. Objeto y fines de la cesión.

Con el fin de dar la máxima difusión a la obra citada a través del Repositorio institucional de la Universidad, el autor **CEDE** a la Universidad Pontificia Comillas, de forma gratuita y no exclusiva, por el máximo plazo legal y con ámbito universal, los derechos de digitalización, de archivo, de reproducción, de distribución y de comunicación pública, incluido el derecho de puesta a disposición electrónica, tal y como se describen en la Ley de Propiedad Intelectual. El derecho de transformación se cede a los únicos efectos de lo dispuesto en la letra a) del apartado siguiente.

3º. Condiciones de la cesión y acceso

Sin perjuicio de la titularidad de la obra, que sigue correspondiendo a su autor, la cesión de derechos contemplada en esta licencia habilita para:

a) Transformarla con el fin de adaptarla a cualquier tecnología que permita incorporarla a internet y hacerla accesible; incorporar metadatos para realizar el registro de la obra e incorporar "marcas de agua" o cualquier otro sistema de seguridad o de protección.
b) Reproducirla en un soporte digital para su incorporación a una base de datos electrónica, incluyendo el derecho de reproducir y almacenar la obra en servidores, a los efectos de garantizar su seguridad, conservación y preservar el formato.
c) Comunicarla, por defecto, a través de un archivo institucional abierto, accesible de modo libre y gratuito a través de internet.
d) Cualquier otra forma de acceso (restringido, embargado, cerrado) deberá solicitarse expresamente y obedecer a causas justificadas.
e) Asignar por defecto a estos trabajos una licencia Creative Commons.
f) Asignar por defecto a estos trabajos un HANDLE (URL *persistente)*.

4º. Derechos del autor.

El autor, en tanto que titular de una obra tiene derecho a:

a) Que la Universidad identifique claramente su nombre como autor de la misma
b) Comunicar y dar publicidad a la obra en la versión que ceda y en otras posteriores a través de cualquier medio.
c) Solicitar la retirada de la obra del repositorio por causa justificada.
d) Recibir notificación fehaciente de cualquier reclamación que puedan formular terceras personas en relación con la obra y, en particular, de reclamaciones relativas a los derechos de propiedad intelectual sobre ella.

5º. Deberes del autor.

El autor se compromete a:

a) Garantizar que el compromiso que adquiere mediante el presente escrito no infringe ningún derecho de terceros, ya sean de propiedad industrial, intelectual o cualquier otro.
b) Garantizar que el contenido de las obras no atenta contra los derechos al honor, a la intimidad y a la imagen de terceros.
c) Asumir toda reclamación o responsabilidad, incluyendo las indemnizaciones por daños, que pudieran ejercitarse contra la Universidad por terceros que vieran infringidos sus derechos e intereses a causa de la cesión.
d) Asumir la responsabilidad en el caso de que las instituciones fueran condenadas por infracción de

derechos derivada de las obras objeto de la cesión.

6º. Fines y funcionamiento del Repositorio Institucional.

La obra se pondrá a disposición de los usuarios para que hagan de ella un uso justo y respetuoso con los derechos del autor, según lo permitido por la legislación aplicable, y con fines de estudio, investigación, o cualquier otro fin lícito. Con dicha finalidad, la Universidad asume los siguientes deberes y se reserva las siguientes facultades:

- La Universidad informará a los usuarios del archivo sobre los usos permitidos, y no garantiza ni asume responsabilidad alguna por otras formas en que los usuarios hagan un uso posterior de las obras no conforme con la legislación vigente. El uso posterior, más allá de la copia privada, requerirá que se cite la fuente y se reconozca la autoría, que no se obtenga beneficio comercial, y que no se realicen obras derivadas.
- La Universidad no revisará el contenido de las obras, que en todo caso permanecerá bajo la responsabilidad exclusive del autor y no estará obligada a ejercitar acciones legales en nombre del autor en el supuesto de infracciones a derechos de propiedad intelectual derivados del depósito y archivo de las obras. El autor renuncia a cualquier reclamación frente a la Universidad por las formas no ajustadas a la legislación vigente en que los usuarios hagan uso de las obras.
- La Universidad adoptará las medidas necesarias para la preservación de la obra en un futuro.
- La Universidad se reserva la facultad de retirar la obra, previa notificación al autor, en supuestos suficientemente justificados, o en caso de reclamaciones de terceros.

Madrid, a 31 de Agosto de 2019

ACEPTA

Fdo……………………………………………

Motivos para solicitar el acceso restringido, cerrado o embargado del trabajo en el Repositorio Institucional:

Declaro, bajo mi responsabilidad, que el Proyecto presentado con el título

Alternatives to the sine test in satellite verification

en la ETS de Ingeniería - ICAI de la Universidad Pontificia Comillas en el

curso académico 2018-2019 es de mi autoría, original e inédito y

no ha sido presentado con anterioridad a otros efectos.

El Proyecto no es plagio de otro, ni total ni parcialmente y la información que ha sido tomada

de otros documentos está debidamente referenciada.

Marcos Feria Cerrada

Fdo.:…………………………………………..   Fecha:…31…./…08…./…2019….

Autorizada la entrega del proyecto

LOS DIRECTORES DEL PROYECTO

Miguel Such Taboada

Digitally signed by Miguel Such Taboada
Date: 2019.09.02 10:13:22 +02'00'

Fdo.: ………………………………

Alberto Carnicero López

Firmado digitalmente por carnicero@comillas.edu
Nombre de reconocimiento (DN): cn=carnicero@comillas.edu
Fecha: 2019.09.01 22:54:36 +02'00'

Fdo.: ……………………………………

Fecha: ………./………./……….

ALTERNATIVES TO THE SINE TEST IN SATELLITE VERIFICATION

Autor: Feria Cerrada, Marcos.

Directores: Such Taboada, Miguel
Carnicero López, Alberto.

Entidad Colaboradora: ICAI - Universidad Pontificia Comillas

RESUMEN DEL PROYECTO

INTRODUCCION

Los satélites, como cualquier otra estructura, se diseñan y ensayan a través de procesos de verificación en base a especificaciones que describen las cargas que deben soportar. La definición de estas especificaciones no es trivial, ya que las cargas más críticas durante el lanzamiento son transitorias y de naturaleza compleja.

Actualmente, existen procesos de verificación y especificaciones para cada uno de los diferentes entornos presentes durante un lanzamiento. Aunque han demostrado ser fiables y son ampliamente utilizados en la industria espacial, aún tienen algunas limitaciones importantes. Dado que abordar todas las limitaciones es demasiado ambicioso, este trabajo se centrará en estudiar las metodologías utilizadas para el entorno de baja frecuencia, que se basan en lo que se conoce como ensayo seno.

El ensayo seno, consiste en aplicar señales sinusoidales barridas en frecuencia en la base de la estructura a través de agitadores electrodinámicos. Tal proceso, tiene varias desventajas. En primer lugar, la equivalencia establecida entre barridos sinusoidales y transitorios reales de baja frecuencia es cuestionable. En segundo lugar, estas pruebas se llevan a cabo eje por eje y aisladas de otros entornos, mientras que durante un lanzamiento real diferentes entornos actúan simultáneamente y generan cargas en todas las direcciones. Además, la larga duración del ensayo seno compromete los tiempos de entrega del satélite. Por estas razones, es necesario estudiar nuevas alternativas que ayuden a superar estos y otros problemas.

El objetivo de este trabajo es comprender más profundamente las limitaciones del ensayo seno y estudiar dos métodos alternativos de generación de señales, el barrido sinusoidal rápido y el método de ondículas. Para ello, se realizarán simulaciones en un modelo de elementos finitos de un satélite simplificado y se compararán los resultados obtenidos en cada uno de los métodos.

METODOLOGÍA

En primer lugar, partiendo de la suposición de que dos señales con espectros de respuesta de choque similares causan daños equivalentes, se han estudiado los tres métodos de generación de señales.

Para verificar si la suposición de equivalencia basada en el espectro de respuesta de choque era razonable, era necesario simular la respuesta de un satélite a la excitación en su base con las diferentes señales sintéticas. Con este propósito, se ha desarrollado el modelo de elementos finitos de un satélite simplificado para *AFECTOS* en *MATLAB*. Para calcular la respuesta de dicho modelo a excitaciones en la base, se han considerado dos enfoques: el Análisis Modal y el método de integración en el tiempo de Newmark. El Análisis Modal no tiene en cuenta efectos transitorios y su aplicación se limita a señales armónicas, lo que significa que solo podría usarse para estudiar el ensayo seno. Por otro lado, el método de Newmark tiene en cuenta los transitorios y puede usarse independientemente de las características de la excitación, lo que lo hizo adecuado para estudiar todos los métodos considerados en este trabajo.

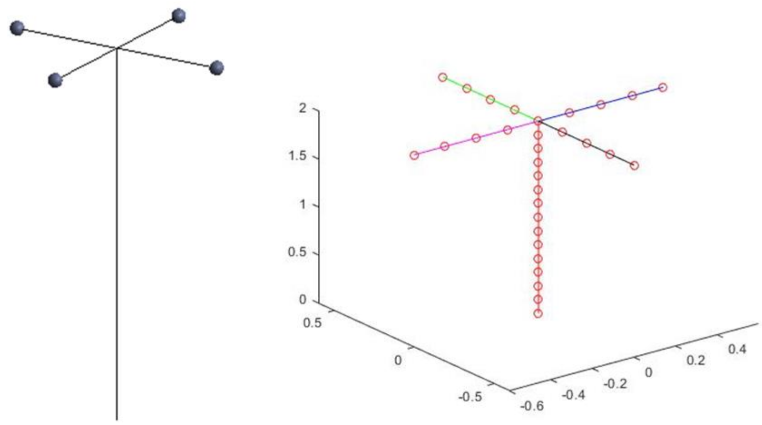

Figura 1: Modelo de elementos finitos simplificado de un satélite

Las primeras simulaciones se han realizado con ondas sinusoidales simples de amplitud y frecuencia constantes. Estas simulaciones han validado que el Análisis Modal y el Método Newmark fueron programados correctamente. El siguiente paso ha sido simular el ensayo seno obtenido a partir de un transitorio simple ficticio. Finalmente, los ensayos seno obtenidos a partir de un transitorio de vuelo real se han simulado. La comparación de los resultados obtenidos con cada uno de los métodos, era deseable para determinar la influencia de los efectos transitorios, que no se consideraron con el análisis modal. Además, los resultados también se han comparado con la respuesta del satélite al transitorio de vuelo original para determinar si la equivalencia entre el ensayo seno y la señal original es razonable.

Después de todas estas simulaciones, era hora de estudiar los otros métodos de generación de señal, el barrido sinusoidal rápido y el método de ondículas. Dado que las señales sintéticas obtenidas con estos métodos no son harmónicas, el cálculo de la respuesta del satélite a las señales resultantes se realizó con el método Newmark. Los resultados de las simulaciones se han comparado con los de las simulaciones del ensayo seno y el transitorio de vuelo original.

Finalmente, se ha estudiado la posibilidad de usar el análisis mediante ondículas para sintetizar señales más realistas. Los métodos descritos anteriormente pudieron capturar las frecuencias y amplitudes más importantes de la señal original, pero no los puntos temporales en los que tuvieron lugar. La dimensión del tiempo es importante ya que la acumulación de daño depende del historial de tiempo de las cargas. Al usar el análisis mediante ondículas, ha sido posible sintetizar señales que tienen el mismo contenido en frecuencia que el transitorio de vuelo original. La excitación del satélite con estas señales también se ha simulado y se han sacado conclusiones.

RESULTADOS Y CONCLUSIONES

La motivación del presente trabajo surgió de las limitaciones involucradas al utilizar el ensayo seno para simular y ensayar cargas de baja frecuencia en satélites. Los primeros resultados tienen como propósito analizar estas limitaciones:

- Se ha observado que usando una velocidad de barrido razonable, el ensayo seno para una señal de 1 segundo tiene una duración de varios minutos. Teniendo en cuenta que normalmente son necesarios varios ensayos por eje, está claro que el ensayo seno es una metodología lenta.
- Los resultados muestran que los barridos sinusoidales son capaces de conseguir una función espectral de potencia similar a la de señales simples. Sin embargo, cuando las señales son más complejas, como la de un transitorio de vuelo, aparece un desajuste importante en las funciones espectrales de potencia.
- Como se puede ver en la *tabla 1*, las cargas máximas que genera el barrido sinusoidal equivalente en el satélite no son tan parecidas a las cargas máximas generadas por el transitorio de vuelo original. En otras palabras, la equivalencia asumida entre barridos sinusoidales y transitorios reales no es rigurosa.
- La velocidad de barrido afecta a la amplitud de la respuesta del sistema. Además, los resultados muestran que en un sistema de múltiples grados de libertad, la influencia de la velocidad de barrido depende del intervalo de frecuencia. En nuestro modelo, una tasa de barrido positiva condujo a aceleraciones más altas entre 25 y 70 Hz, aceleraciones más bajas entre 10 y 25 Hz y aceleraciones similares entre 70 y 100 Hz.
- Cuando se excita con el barrido sinusoidal, el modelo está sujeto a un número de ciclos excesivamente alto. En un ensayo real, esto implica que la estructura correrá el riesgo de sufrir daños por fatiga.

Los resultados descritos hasta el momento, muestran que las limitaciones del ensayo seno son importantes. Por lo tanto, tiene sentido que se investiguen nuevos métodos alternativos. En este trabajo se han estudiado tres alternativas:

- Método de Barrido Sinusoidal Rápido (BSR)
- Método de ondículas
- Análisis mediante ondículas para sintetizar señales con un contenido en frecuencia similar al del transitorio de vuelo original

Los primeros dos métodos sintetizan señales basadas en los espectros de frecuencia del transitorio de vuelo original, pero no consideran los instantes de tiempo en que actúan esas frecuencias. Después de implementar ambos métodos y llevar a cabo las simulaciones, se obtuvieron los siguientes resultados:

- Las señales sintéticas resultantes tienen una duración similar a la del transitorio de vuelo original y el número de ciclos se reduce considerablemente.
- Las señales generadas tienen una función espectral de potencia que coincide con la del transitorio de vuelo original, algo que el barrido sinusoidal clásico no era capaz de conseguir. El método de ondículas parece ser más preciso en esta tarea, pero es importante señalar que se simplificó el proceso de optimización del barrido sinusoidal rápido. La implementación de una optimización más rigurosa es interesante para futuros estudios.
- Los resultados indican que la introducción de las cargas eje por eje no es equivalente a las condiciones reales en las que existen condiciones de carga en varios ejes. Cuando se introducen excitaciones en varios ejes simultáneamente, las cargas resultantes son más altas.
- De la *tabla 1* se puede concluir que cuando las simulaciones y las pruebas se llevan a cabo eje por eje, ambos métodos reproducen cargas similares a las generadas por el transitorio original. Sin embargo, estas cargas no siempre son más altas que las generadas por la señal original. Por otro lado, *la tabla 2* muestra que cuando se introducen cargas en múltiples ejes simultáneamente, el método del Barrido Sinusoidal Rápido produce cargas más altas que las de la señal original.

Aceleración máxima de las masas del satélite (eje por eje) [m/s^2]				
	Transitorio de vuelo	Ensayo seno	BSR	Ondículas
X	12.33 (mass 2)	10.18 (mass 2)	12.4 (mass 2)	13.04 (mass 2)
Y	1.62 (mass 3)	1.427 (mass 3)	1.21 (mass 3)	1.42 (mass 3)
Z	34.6 (mass 3)	28.42 (mass 3)	39.18 (mass 3)	35.36 (mass 3)

Tabla 1: Aceleración máxima de las masas del satélite (excitación eje por eje)

Aceleración máxima de las masas del satélite (múltiples ejes) [m/s^2]				
	Transitorio de vuelo	Ensayo seno	BSR	Ondículas
X	12.45 (2)	10.13 (2)	12.5 (2)	12.93 (2)
Y	1.503 (3)	1.346 (3)	2.657 (3)	1.398 (3)
Z	42.45 (3)	40.39 (1)	44.95 (3)	41.73 (3)

Tabla 2: Aceleración máxima de las masas del satélite (múltiples ejes)

Finalmente, el Análisis mediante Ondículas, sí considera el dominio del tiempo en la síntesis de señales equivalentes. Al realizar las simulaciones, se han obtenido los siguientes resultados:

- Las señales sintéticas resultantes tienen la misma duración que el transitorio de vuelo original y el número de ciclos casi idéntico.
- La coincidencia de las funciones espectrales de potencia es peor que con el BSR y el método de Ondículas.
- Se pueden obtener múltiples señales de un solo transitorio y la variabilidad con respecto a las cargas que generan es importante. Si se eligen las señales más críticas, es muy posible que las cargas sean algo superiores a las del transitorio original.

Teniendo en cuenta los resultados resumidos anteriormente, se pueden extraer las siguientes conclusiones generales:

- El ensayo seno es el peor enfoque entre todos los métodos estudiados en este trabajo. La señal es completamente diferente al transitorio original y las cargas que genera no pueden considerarse del todo equivalentes. Además, el barrido sinusoidal influye en la amplitud de las cargas resultantes.
- Los métodos BSR y Ondículas son alternativas interesantes, pueden sintetizar señales que son más similares a los transitorios de vuelo y las cargas resultantes también son más similares a las producidas por los dichos transitorios. La especificación de cargas de baja frecuencia para satélites podría definirse adecuadamente en función de las señales obtenidas con ambos métodos. El proceso para obtener una especificación válida a partir de varios transitorios de vuelo se deja como un tema pendiente para futuros estudios.
- El método de Análisis mediante Ondículas ha demostrado que es posible obtener señales con un contenido de frecuencia similar al transitorio de vuelo original. Sin embargo, los resultados obtenidos muestran que las cargas generadas son variables dependiendo de la señal sintética que se elija entre todas las posibilidades existentes. Encontrar las funciones de modulación que son más apropiadas para definir la especificación de las cargas de baja frecuencia para satélites debería ser posible y se deja como una tarea pendiente para futuros estudios.

MULTIDISCIPLINARY SIMULATION OF THE INTERACTION BETWEEN PANTOGRAPH AND RIGID CATENARY

Author: Feria Cerrada, Marcos.

Directors: Carnicero López, Alberto.
Jiménez Octavio, Jesús Ramón.

Collaboration entity: ICAI - Universidad Pontificia Comillas

SUMMARY

INTRODUCTION

Satellites, like any other structure, are designed and tested through a verification process based on specifications that describe the loads they must withstand. The definition of these specifications is not trivial since most critical loads during launch are transient and depend on the nature of different complex environments.

Currently, verification processes and specifications exist for each of the different environments. Although they have proved to be reliable and are widely used in the space industry, they still have some important limitations. Since tackling all of them is too ambitious, this work will only focus in studying those aspects that can be improved in the methodologies used for the low frequency environment, which are based on what is known as the sine-test or Classical Sine Sweep.

Sine-testing consists on applying swept-sine excitation signals at the base of the structure in electrodynamic shakers. Such a process has several disadvantages. In first place, the established equivalence between sine-sweeps and real low frequency transients is questionable. Secondly, these tests are carried out axis by axis and isolated from other environments while during a real launch different environments act simultaneously and generate loads in every direction. Moreover, tests' long duration compromise the assembly planning. For these reasons, studying new alternatives that help to overcome these and other problems is necessary.

The aim of this work is to understand more deeply the limitations of the sine-test and to study two alternative signal-generation methods, the Fast Sine Sweep and the Wavelet method. This will be done by carrying out simulations on a Finite Element Model of a simplified satellite and comparing the results obtained for each of the methods.

METHODOLOGY

Firstly, the project has studied the three signal-generation methods based on the assumption that acceleration time histories with similar Shock Response Spectrum cause equivalent damage. In this way, the different methods have been compared based on their ability to synthesize signals that match a given SRS.

In order to check if the assumption of equivalence based on SRS was reasonable, simulating the excitation of a satellite at its base with the different signals was necessary. With this purpose, a Finite Element Model of a simplified satellite has been developed for *AFECTOS* in *MATLAB*. To calculate the response of such model to base excitations, two approaches have been considered: Modal Analysis and Newmark's time integration method. Modal Analysis neglects transient effects and is restricted to harmonic signals, which means that it could only by used to study the sine-test. On the other hand, the Newmark method takes transients into account and can be used regardless of the characteristics of the excitation, which made it suitable to study all the methods considered in this work.

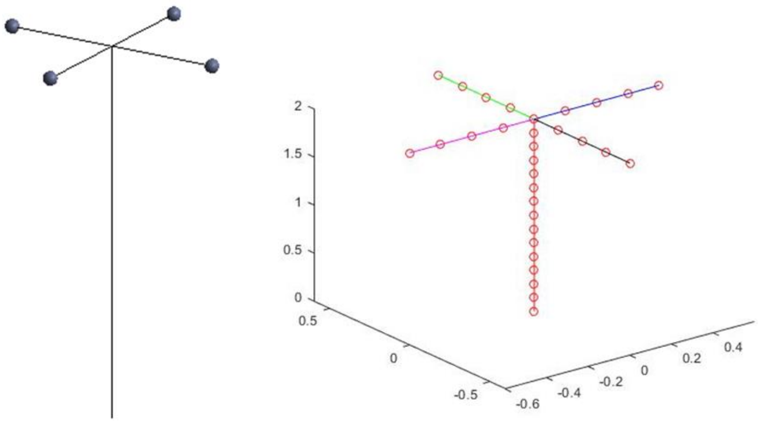

Figure 1: Simplified Finite Element Model of a satellite

The first simulations of the FEM have been done with simple sine waves with constant amplitude and frequency. These simple simulations have validated that the Modal Analysis and the Newmark Method approaches were correctly programmed. The next step has been to simulate the sine-tests or Classical Sine Sweeps obtained from a simple fictitious transient. Finally, sine-tests obtained from a real flight transient have been simulated using both Modal Analysis and Newmark's time integration method. The comparison of the results obtained with each of the methods was desired to determine the influence of transient effects, which were not considered with Modal Analysis. Moreover, the results have also been compared with the response of the satellite to the original flight transient in order to determine if the equivalence between the sine-test and the original signal is reasonable.

After all these simulations, it was time to study the Fast Sine Sweep and the Wavelets signal-generation methods. Given the non-harmonic characteristic of the signals obtained with these methods, the calculation of the response of the satellite to the resulting signals was done with the Newmark method. The results of the simulations have been compared with those from the simulations of the sine-test and the original flight transient.

Finally, the possibility of using Wavelet Analysis to synthesize more realistic signals has been studied. Previous described methods, were able to capture the most important frequencies and amplitudes of the original signal but not the time points in which they took place. The time dimension is important since the accumulation of damage depends on the time history of loads. By using Wavelet Analysis it has been possible to synthesize signals that have the same frequency content than the original flight transient. The excitation of the satellite with these signals has also been simulated and conclusions have been drawn.

RESULTS AND CONCLUSIONS

The motivation of the present work arose from the limitations involved when using the sine-test to simulate and test low-frequency loads in satellites. The first results concern the analysis of these limitations:

- It has been shown that when using a reasonable sweep rate the equivalent sine-sweep for a *1 second* signal has a duration of *350 seconds*. Taking into account that normally several runs for each axis are necessary, it is clear that the sine-test is a slow methodology.
- The results show that equivalent sine-sweeps are able to match the SRS of simple signals. However, when signals are more complex, such as a flight transient, an important mismatch in the SRS appears.
- As it can be seen in *table 1,* the maximum loads that the equivalent sine-sweep generates in the satellite are not as high as the maximum loads generated by the original flight transient. In other words, the assumed equivalence between sine-sweeps and real transients is not rigorous.
- The sweep rate affects the amplitude of the response of the system. Moreover, the results show that in a MDOF system, the influence of the sweep rate depends on the frequency interval. In our specific model, a positive sweep rate led to higher accelerations between 25 to 70 Hz, lower accelerations between 10 and 25 Hz and similar accelerations between 70 and 100 Hz.
- When excited with the sine-sweep, the model is subjected to an unreasonably high number of cycles. In a real test this implies that the structure will be over-tested from a fatigue point of view.

The above-mentioned points show that the limitations of the sine-test are in fact important. Therefore it makes sense that new and alternative approaches are investigated. In this work three alternatives have been studied:

- Fast Sine Sweep method
- Wavelets method
- Wavelet Analysis to synthesize signals with equal frequency content than the original flight transient

The first two methods synthesize signals based on the frequency spectra of the original flight transient, but do not consider the time instants in which those frequencies act. After implementing both methods and carrying out the simulations the following results were obtained:

- The resulting synthetic signals now have a similar duration than the original flight transient and the number of cycles is considerably reduced.
- The signals generated with the FSS and the Wavelets method match correctly the SRS of the original flight transient, something the sine-test was not able to do. The Wavelet method seems to be more precise in this task, but it is important to remark that the optimization process of the FSS was simplified. The implementation of the FSS method with a more rigorous optimization is interesting for future studies.
- The results indicate that introducing the loads axis by axis is not equivalent to real conditions in which multiple-axis loading conditions exist. When excitations are introduced in multiple axes simultaneously the resulting loads are higher.

- From *table 1* it can be concluded that when simulations and tests are carried axis by axis both methods reproduce loads that are similar to those generated by the original transient. However, these loads are not always higher than those generated by the original signal, which means that the model could be under-tested. On the other hand, *table 2* shows that when loads are introduced in multiple axes simultaneously the FSS method does generate higher loads.

Maximum acceleration of the masses (axis by axis) [m/s^2]				
	Real flight transient	Sine sweep	FSS	Wavelets
X	12.33 (mass 2)	10.18 (mass 2)	12.4 (mass 2)	13.04 (mass 2)
Y	1.62 (mass 3)	1.427 (mass 3)	1.21 (mass 3)	1.42 (mass 3)
Z	34.6 (mass 3)	28.42 (mass 3)	39.18 (mass 3)	35.36 (mass 3)

Table 1: Maximum accelerations with different excitations (axis by axis)

Maximum acceleration of the masses (multiple axes) [m/s^2]				
	Real flight transient	Sine sweep	FSS	Wavelets
X	12.45 (2)	10.13 (2)	12.5 (2)	12.93 (2)
Y	1.503 (3)	1.346 (3)	2.657 (3)	1.398 (3)
Z	42.45 (3)	40.39 (1)	44.95 (3)	41.73 (3)

Table 2: Maximum accelerations with different excitations (axis by axis)

Finally, Wavelet Analysis considers the time domain in the synthesis of equivalent signals and has led to the following results:

- The resulting synthetic signals have equal duration than the original flight transient and the number of cycles almost identical.
- The matching of the SRS is worse than with the FSS and the Wavelets method.
- Multiple signals can be obtained from a single transient and the variability with respect to the loads they generate is important. If most critical signals are chosen then the satellite will be over-tested with respect to the original transients.

Taking into account the results summarized above, the following general conclusions can be drawn:

- The sine-test is the worst approach among all the methods studied in this work. The signal is completely different to the original transient and the loads it generates cannot be considered equivalent. Moreover, the sine-sweep influences the amplitude of the resulting loads.
- The FSS and Wavelets methods are interesting alternatives, they are able to synthesize signals that are more similar to the original transients and the resulting loads are also more similar to those produced by the original transients. The low frequency load specification for satellites could be appropriately defined based on signals obtained with both of the methods. The process to obtain a valid specification from several flight transients is left as an interesting topic for future studies.

- The Wavelet Analysis method has shown that it is possible to obtain signals with a frequency content that is similar to the one of the original transient. However, the obtained results show that the generated loads are variable depending on the synthetic signal that is chosen from all the existing possibilities. Finding the modulating functions that are most appropriate to define the specification of the low frequency loads for satellites should be possible and is left as an interesting task for future studies.

MÁSTER UNIVERSITARIO EN INGENIERÍA INDUSTRIAL

TRABAJO FIN DE MÁSTER

ALTERNATIVES TO THE SINE TEST IN SATELLITE VERIFICATION

Autor: Marcos Feria Cerrada

Director: Miguel Such Taboada

Co-Director: Alberto Carnicero López

Madrid

Agosto de 2019

Abstract

Satellites are designed and tested through verification processes based on specifications that describe the different loads they must withstand. Sine-testing is one of the stages in this whole process. It consists on applying swept-sine excitation signals at the base of the structure in electrodynamic shakers to verify that the satellite will survive to the low frequency transients present during launch. Such a test however, has several disadvantages. The aim of this work is to understand more deeply the limitations of the sine-test and to study two alternative signal-generation methods, the Fast Sine Sweep and the Wavelet method. This will be done by carrying out simulations on a Finite Element Model of a simplified satellite and comparing the results obtained for each of the methods.

Resumen

Los satélites se diseñan y ensayan a través de procesos de verificación basados en especificaciones que describen las cargas que deben soportar. El ensayo seno es una de las herramientas utilizadas en este proceso. Consiste en aplicar señales sinusoidales barridas en frecuencia en la base de la estructura a través de agitadores electrodinámicos para simular las cargas de baja frecuencia presentes durante un lanzamiento. Este ensayo, sin embargo, tiene varias desventajas. El objetivo de este trabajo es comprender más profundamente las limitaciones del ensayo seno y estudiar dos métodos alternativos de generación de señales, el Barrido Sinusoidal Rápido y el método de Ondículas. Para ello se realizarán simulaciones en un modelo de elementos finitos de un satélite simplificado y se compararán los resultados obtenidos para cada uno de los métodos.

Table of contents

Table of Figures

Table of Tables

Acronyms

QSL	Quasi Static Loads
MAC	Mass Acceleration Curves
FEM	Finite Element Model
SPL	Sound Pressure Level
PSD	Power Spectral Density
RMS	Root Mean Square
SEA	Statistical Energy Analysis
SRS	Shock Response Spectrum
SDRA	Shock Damage Risk Assessment
ESI	Equivalent Sine Input
RRS	Random Response Spectrum
MDOF	Multiple Degree Of Freedom
GA	Genetic Algorithm
CLA	Coupled Load Analysis

Chapter 1
Introduction

During launch, satellites structures experience different types of mechanical loads. The process to establish reasonable loads during the design and verification of the satellite is called load analysis. Predicting and analyzing loads is extremely important not only for designing the structure of the satellite, but also for verifying the design by analysis and test. This is not a trivial task as the loads are transient and depend on the nature of the environments and the boundary elements that cannot be fully represented during the verification process.

The aim of the present work is to study the low-frequency mechanical specification used to design and verify satellites, which is generally represented by a sine sweep test. Different approaches to derive such specification will be discussed and compared by performing Finite Element Analysis in a simplified model of a satellite. Data extracted from a real flight transient will be used during this process. With the obtained results, an attempt to draw conclusions about the advantages and disadvantages of each of the approaches will be done.

1.1. Motivation and objectives

The main motivation of the present work arises from the current over-testing of satellite structures under sine vibration environment which, simultaneously, does not fully envelope the qualification needs given the differences between the real flight phenomena and the test configuration.

The main objective will be to compare different signal-generation methods and to draw conclusions about which of them are more appropriate to define the low-frequency mechanical specification in order to minimize the over-testing and still ensure a broad qualification.

Among the studied signal-generation methods, there will be the traditional sine sweep, currently used in the space industry and also other approaches that are yet not used due to its novelty.

The objective will be pursued by studying the different methods individually and later on comparing them by carrying out finite element analysis on a simplified model of a satellite using real flight transient data. If the above-mentioned objective is fulfilled, an attempt to propose a new and personal approach will be done.

1.2. Methodology

In first place, the field of verification of satellites will be studied thoroughly. This will imply understanding the loads that are present during a satellite's launch and the different tests that are required during the verification stage. Throughout this learning process, it will also be necessary to start working with some mathematical concepts that will be necessary for the development of the study, such as Shock Response Spectrum, Wavelet Theory, Newmark's time integration method and Modal Analysis.

Once the required knowledge has been acquired, the work will focus on the low frequency environment. The first step will be to study different approaches to synthesize signals that match the Shock Response Spectrum of a real flight transient. The idea is to obtain signals that have a similar damage capacity on satellites than real flight transient signals. The first approach to study will be the classical sine sweep, which is currently the most used. This part will be followed with the study of two other more novel approaches, the Fast Sine Sweep method and the Wavelets method.

Until this point, different methods will have only been compared based on their ability to match a given Shock Response Spectrum. However, to further determine if these methods are appropriate, Finite Element Analysis with a simplified model of a satellite will be carried out. Given a real flight transient, the methodology will be to synthesize different signals according to the already mentioned methods and to determine if the damage that the satellite will suffer is equivalent to the one if the real signal was introduced. In order to do all this, some of the steps will be:

1. Find or code a program that calculates the Shock Response Spectrum of an acceleration time history.
2. Find or code different programs to apply each of the different methods to synthesize the new acceleration time histories.
3. Learn to use *AFECTOS*, a Finite Element Analysis tool developed for *MATLAB.*
4. Learn to use genetic algorithms to perform optimizations.

5. An implementation of the Newmark time integration method to calculate the response of the satellite to each of the input signals in *AFECTOS*.
6. Perform the analysis with the different methods and compare the results with those expected from a coupled transient.

Finally, conclusions will be drawn and recommendations will be given. If possible, further investigations or approaches will be proposed.

1.3. Resources

- During the learning and study phase of the work, information will be gathered from different sources. Books, articles and other reading materials will be used to acquire the necessary knowledge.
- To perform complex mathematical calculations and to apply the different methods to synthesize new acceleration time histories *MATLAB* will be used.
- To carry out the Finite Element Analysis *AFECTOS*, will be used.
- Other Finite Element programs like *ANSYS* will be used to compare the results of some of the analysis.
- *EXCEL* will be used as a secondary program to collect and process data.

1.4. Structure of the work

The present work has a total of nine chapters structured in the following way:

- **Chapter 1:** Introduction to the work. The motivation and objectives are explained, the methodology that has been implemented is presented and the necessary resources to carry out the work are enumerated.
- **Chapter 2:** The State of the Art of mechanical load analysis and verification of satellites is summarized. A description of the different loads that satellites must withstand is given followed by an explanation of the most common simulations and tests for their verification. Finally, the current limitations of these processes and the need for new methods are explained.
- **Chapter 3:** The concept and applications of the Shock Response Spectrum is explained. Based on this concept, three different methods to synthesize equivalent acceleration time histories are studied and compared: Classical Sine Sweep, Fast Sine Sweep and Wavelets.

- **Chapter 4:** Preparation of the dynamic analysis. The simplified Finite Element Model of the satellite is described and the implementation of the Newmark time integration method to calculate the response of a Finite Element Model excited at its base is explained, programmed and validated.
- **Chapter 5:** The response of the satellite to the Classical Sine Sweep is calculated using Modal Analysis. The steady state response of the satellite is calculated according to the analytical solution of a Multiple Degree of Freedom system subjected to a harmonic excitation. The response is then compared with the one obtained with the Newmark time integration method with the objective of determining the importance of transient effects.
- **Chapter 6:** This chapter focuses on studying the influence of the sweep rate when calculating the response of a satellite to a Classical Sine Sweep.
- **Chapter 7:** Dynamic analysis of the satellite subjected to synthetic excitations obtained from a real flight transient using the Fast Sine Sweep and the Wavelets methods. The response of the satellite to these synthetic signals is compared with the response of the satellite to the original flight transient and the Classical Sine sweep.
- **Chapter 8:** Application of Wavelet Analysis to synthesize acceleration time histories that have the same frequency content than real flight transients. The dynamic analysis using the obtained synthetic signals is carried out and compared with previous results.
- **Chapter 9:** Summary of the conclusions of the work.

Chapter 2
State of the Art

The loads that the satellite will experience depend on the properties of its own structure. Simultaneously, this structure is designed to withstand the loads. Therefore, load analysis and structural design are an iterative process that is known as “Loads cycles” [1]. In each of the iterations of this process we can usually identify four main activities:

- Generation of mathematical models of the present design.
- Simulation of critical environments and calculation of loads.
- Evaluation of results and identification of modifications or improvements.
- Implementation of modifications.

Projects in the space industry perform more than one loads cycle. Usually there is a preliminary design and a final design followed by a final verification.

The sequence followed in the design and verification of a satellite will depend on a wide range of factors and will be different for each satellite. However, it is possible to identify some general stages. The first task is to identify the main architecture of the structure of the satellite and to roughly dimension it. This preliminary dimensioning is done using the load factors provided by the launch vehicle guide. Once the dimensioning is finished it is possible to develop a preliminary Finite Element Model and carry out a Frequency Response Analysis which will simulate the sine vibration test and will help to do an initial definition of the primary notching. This step will be followed by a preliminary Coupled Load Analysis.

After these preliminary stages, more detailed Finite Element Models can be developed and improvements in the design of the primary structure can be done. Once the final structure design is reached, more Frequency Response Analysis are carried out, followed by Random Vibrations Analysis and Vibro-Acoustic Analysis. Finally, the definitive Coupled Load Analysis is done and the whole process finishes with the verification of compliance.

2.1. Mechanical loads for design and verification of satellites

The structure of a satellite will usually experience the most critical loads during launch. These loads will be generated by multiple sources in different flight environments. According to their characteristics they can be classified into:

- Static and quasi-static loads, introduced by the steady state acceleration of the launcher that is either constant or changes slowly with time.
- Low-frequency dynamic response due to transient events.
- High-frequency random vibration transmitted by the launch vehicle.
- High-frequency dynamic response to the acoustic pressure environment.
- Shock loads as a consequence of transient mechanical events of very short durations such as pyrotechnic detonations.

The design and analysis process of the satellite is performed based on an estimation of the loads provided by the launch company or based on past experience. After this, environmental testing is carried out. The main objectives of the tests are:

- To verify mechanical strength.
- To validate the satellite mathematical models with the measured responses.
- To verify that no defects in the system will cause problems.

The tests are based on environmental predictions calculated with data from previous flight or tests. However, to account for the uncertainties involved in these predictions, environments during the tests need to be more severe than in reality.

Load	**Verification by analysis**	**Verification by test**
Static & Quasi-static	Static Analysis	Static test Sine burst test Sine vibration test
Low frequency transient environment	Transient Analysis Frequency Response Analysis	Sine vibration test

Shock & high frequency transient	Transient Analysis Shock propagation assessment	Shock test
Random & acoustic vibration	Random Vibration Analysis Vibro-Acoustic Analysis	Random vibration test Acoustic noise test

Table 2.1: Verification of different types of loads

2.1.1. Static and quasi-static loads

Ideally, a static load is a load with constant magnitude and direction. However, it is common to work with some loads as if they were static even if they are not. These loads are called quasi-static loads and are equivalent static loads obtained as a combination of static and dynamic loads. The established equivalence, is based on the fact that the new static force at the center of gravity of the structure will lead to similar interface forces to those produced by the most severe conditions that will be encountered during the launch of the satellite.

2.1.1.1. Quasi-stati loads specification

As it has been explained previously, during launch the flight environment results in a combination of structural loads of different natures. All of these loads will be computed separately and later combined with combination methods that give the final Quasi-Static Loads (QSL) or “load factors” [2].

The QSL is then used for the preliminary dimensioning of the primary structure of the satellite. In case of secondary equipment or satellite hardware, the preliminary design is based on empirical curves called Mass-Acceleration Curves (MAC).

2.1.1.2. Quasi-static loads for secondary structures

The Quasi-Static Loads to define the mechanical environment for secondary structures, instruments and units of the satellite can be predicted with a base-drive analysis using the QSL for the primary structure. The prediction is done by performing a Frequency Response Analysis. An advantage of this method is that it avoids the Coupled Load Analysis. However, it also has some relevant limitations:

- Does not account for impedance effects that are important for significant modes. Usually a notching strategy has to be applied.
- Important errors can result from an incorrect estimation of damping.
- High computational effort for models with a high number of relevant modes.
- High frequency response is unreliable.

2.1.1.3. Static tests

Once the design is finished and verification by analysis has been done, static tests are carried out. The main objectives of these tests are to verify that the design will be capable of withstanding the loads and to determine how the structure will distribute them.

Generally, static tests can be classified into:

- **Development tests:** are performed on mechanical parts or assembly components but rarely on the whole structure. They are used to assess design decisions, verify analysis or determine limit loads.
- **Qualification tests:** tests usually performed on mechanical parts, assemblies or the whole structure and their main objective is to verify that the structure satisfies the specification requirements. They verify that the structure will be able to withstand the worst conditions that will be present during launch, even though these conditions will not be present in the test.
- **Acceptance tests:** qualification tests that are performed on flight hardware to demonstrate it fulfils the specification requirements. In this case the worst functioning conditions will be present during test.

2.1.2. Low frequency dynamic loads, sine vibration

Low frequency vibrations, usually up to 100 Hz, excite the spacecraft during flight. The sources of these excitations are the vibration induced by the launcher and the dynamic coupling between all the different elements of the system.

It is necessary to verify that the satellite will be able to withstand these loads without suffering any damage. The most common way to carry out these verifications is through sine testing. In this section some relevant aspects concerning sine vibration specification and testing will be discussed.

2.1.2.1. Sine loads for primary structures

The low frequency environment to which the satellite will be exposed is specified by the launcher authority. The loads are usually given as acceleration amplitudes over frequency to be applied through a base excitation sine sweep test. An example of the defined sine environment for "Ariane 5" launcher is given below:

Direction	Frequency band (Hz)	Sine amplitude (g)
Longitudinal	2-50	1
	50-100	0.8
Lateral	2-25	0.8
	25-100	0.6

Table 2.2: Specification of sine loads for "Ariane 5" launcher

Source: Spacecraft Mechanical Loads Analysis Handbook [1]

2.1.2.2. Sine loads for equipment

Frequently, the instruments and equipment used in the space industry are not designed for a specific project but rather for using them in different spacecraft. Therefore the specification of the sine environment depends on different parameters such as mass and location of the unit and the main direction of the excitation. An example of this specification is given below:

Axis	Frequency (Hz)	Qualification	Acceptance
Out of plane	5-20	15 mm	9.9 mm
	20-100	24 g	16 g
In plane	5-20	9.9 mm	6.6 mm
	20-100	16 g	10.7 g
Sweep rate		2 Oct/min	4 Oct/min

Table 2.3: Example of sine loads specification for equipment

Source: Spacecraft Mechanical Loads Analysis Handbook [1]

Once detailed mathematical models of the satellite with the equipment are available, it is possible to verify if the excitation of the equipment satisfies its specification requirements.

Usually, in the case of not satisfying the requirements, instead of re-designing the satellite the equipment is accommodated in a different configuration.

2.1.2.3. Sine vibration simulation

The sine vibration simulation is carried out with a Finite Element Model of the satellite. Different levels of detail will be developed in the different stages of the design. Usually, before doing the Sine Response Analysis, a Modal Analysis is carried out [3]. Once the modal representation of the model is clear it is time to perform the sine vibration simulation. This is done by applying the loads in the three spatial directions at the base of the satellite. The main goals of this simulation are:

- Predict if the satellite will withstand the sine vibration environment.
- Predict the originated loads in the satellite structure and in the interface with the launcher.
- Predict the loads for instruments and equipment in the satellite.
- Identify the areas in which notching is necessary.

When developing the sine vibration simulation, there are three key factors that should be considered in order to obtain reliable results:

- Boundary conditions
- Damping
- Notching

In first place, boundary conditions will influence the dynamic response of the excited structure. It is important to notice the differences between the ideal boundary conditions in the simulation and the real ones during the test in order to perform a reasonable comparison. Usually, for the analysis, five degrees of freedom are clamped and one is imposed to be the sinusoid. In the real test, the structure is clamped into the shaker and the main problem is to model the torque transferred via the interface bolts.

Damping of the system will also influence the prediction of loads of the structure. The damping ratio of the structure depends on several parameters (such as materials, joints, excitation levels…) and will usually be an unknown parameter before the real test.

Finally, notching is the reduction of the sine acceleration amplitude around resonant frequencies. During real flight the low frequency vibrations at specific frequency values will have a transient nature and resonance is usually not present. For this reason, notching is done so that the structure is not unnecessarily over tested. Notching depends both on damping and the quality of the FEA. However, while uncertainties related with damping can usually be neglected, a good quality of the FEA is essential to accurately predict modal parameters and perform a valid notching. Notching is done taking into account both primary structure (primary notching) and equipment (secondary notching) of the satellite. When implementing it during the test, notch profiles can be automatic or manual. Automatic notches are controlled during the actual test by measuring the response of the structure in real time whereas manual notches are pre-defined reductions at the input excitations. An example of a notch profile is given in *figure 2.1*.

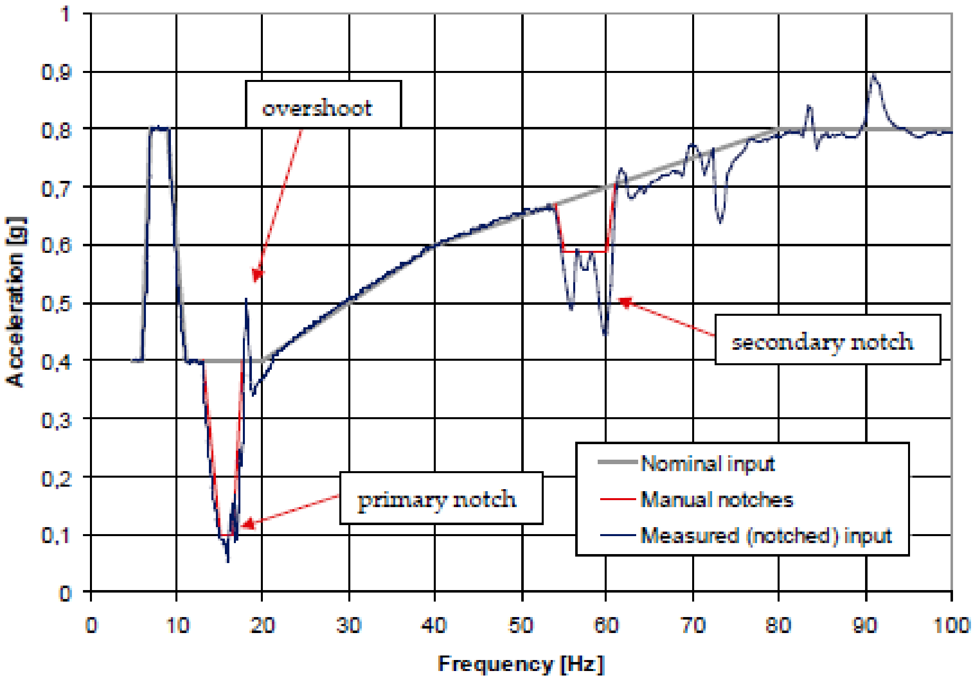

Figure 2.1: Notch profile

2.1.2.4. Sine vibration test

Sine vibration tests are performed one axis at a time in electro-dynamic shakers [4]. The excitation in each of the spatial directions will have a time-variant frequency, from lower to higher frequencies. This frequency variation in time is what is called to sweep the frequency and the velocity with which it variates is called the sweep rate.

During the sine vibration test, several runs are carried out in the shaker. Firstly, a low level run is performed with the purpose of finding resonance frequencies. After scaling the results of this first run, it is possible to check if higher level runs will result in any limit exceedance.

Moreover, this first run will allow estimating the damping factors of the structure. With all these results a notch profile can be defined and a new prediction of a higher level run can be done to check that no limits will be reached. This process is repeated through different iterations until the full level run that should test the structure at the qualification or acceptance level.

For models that will be exposed to a flight environment, the test configuration should be as much representative of the real configuration as possible. Differences between real and test configuration should be evaluated and specified in the results. For this reason, flight models are commonly tested in fully integrated configurations with equipment and instruments included.

2.1.3. Acoustics and random vibration

Both the engine operation and aerodynamic phenomena during launch generate acoustic pressure fluctuations. This acoustic environment induces random vibrations in the structures. The response is more intense in structures that are light and have a big surface area. Although heavier and larger structures experiment a much less significant response to acoustics, they will also vibrate as a consequence of the excitation of the more responsive structures.

The acoustic environment is usually defined with what is called Sound Pressure Level (SPL) that is the root mean square pressure expressed in decibels in a frequency band. In the case of random vibration Power Spectral Density (PSD) is used. Power Spectral Density can be used to represent different types of parameters along a frequency range being the most common acceleration, velocity, displacement, force and stress. Both SPL and PSD are typically represented in a logarithmic scale.

Due to the fact that the random vibrations strongly depend on responsive structures, not all launch manuals provide the PSD specification and it has to be predicted by analysis or test.

2.1.3.1. Random vibration loads

When describing random vibration, the term "random" implies that the value of the load in a certain point in time it is specified by a probability distribution function. In order to give a valid specification of this vibration three steps are taken:

I. Find the PSD of the random vibration that is directly transmitted to the satellite by the launch vehicle through the contact interfaces. This information is given for a specific launch vehicle.
II. Predict with analysis the response of the satellite to the vibro-acoustic environment. In the case that low frequency vibration predictions are needed, the finite element method is a valid approach. However, many times random vibration is a high frequency phenomena and therefore Statistical Energy Analysis (SEA) is a much more effective and accurate predictor.
III. Combine the predicted response (II) and the original data (I) to have a combined random vibration. This combined random vibration specification covers the main sources of random vibration and is used as the acceptance test level.

It is important to notice that the obtained random vibration specification will only be valid if the analysis in the second step is performed with a detailed model of the satellite. The problem is that in early stages of the design these models are not yet available and therefore other methods have to be used to define the random vibration specification. The explanation of these methods falls outside the scope of this work.

2.1.3.2. Vibro-acoustic loads

Acoustic loads are usually represented as Sound Pressure Levels in a diffuse sound field. The diffuse sound field supposition implies that the intensity of sound is equal in every direction. An example of an acoustic load spectrum is shown in the table below:

Octave Centre Frequency (Hz)	**Flight Limit Level (dB)**
31.5	128
63	131
125	136
250	133
500	129
1000	123

Table 2.4: Example of sine loads specification for equipment

Source: Spacecraft Mechanical Loads Analysis Handbook [1]

2.1.4. Shock

During launch, satellites also suffer several energetic shock events such as separation from the launcher or the deployment of some components. The severity of these events should be taken into consideration during the design and verification processes. This is a difficult task given the nature of shock events:

- They are transient mechanical loads.
- They have a short duration.
- They involve high frequency and high amplitude accelerations.
- They have quick initial rising times.

These characteristics, make the shock environment potentially dangerous because of its impact on electronics, on structural materials (cracks, deformation, acceleration of fatigue processes), mechanisms etc.

In order to process shock data, the common procedure is to convert the accelerations in the time domain data into what is called Shock Response Spectrum (SRS) [5]. The SRS is by definition the maximum response of one degree of freedom system expressed as a function of its natural frequency for a given damping ratio. The reason why the SRS is used is that the maximum response already contains the notion of severity to define specifications. However, the use of the SRS also has some disadvantages:

- Mathematical complexity.
- Irreversibility. Data in the time domain corresponds to a unique SRS but a SRS can come from infinite possible time histories. Because of the fact that the SRS only represents amplitudes, phase information and consequently the notion of duration is lost.

2.1.4.1. Shock verification

The first step in the shock verification process is to determine the shock environment for the whole satellite and for each of the units and equipment. The problem is that the state of the art for shock environment prediction is still less developed than other mechanical disciplines. However, a reliable shock environment can be obtained through similarity-heritage-extrapolation methods. Similarity-heritage methods predict the shock environment based on data from past experiences with similar structures and shock sources. These methods are used in early stages of the project in which no shock data of the structure is available. Extrapolation methods are used in later stages and use data available for the given structure to predict shock environments with different but similar shock sources.

Once the shock environment is characterized with sufficient confidence, the shock specification can be established. After that, the next objective is to prove through analysis or test that the different components withstand the environment defined in the specifications. However, there are cases in which shock qualification is not possible. In these cases a Shock Damage Risk Assessment (SDRA) can be performed to determine in terms of risk the worthiness of using the unit.

2.2. Equivalence criteria in the loads verification process

In the loads verification process, acceleration is the most commonly used parameter to measure the severity of an environment, since it is easily measured and directly related to forces and stresses. However, the load specification is different for each of the environments:

- Equivalent accelerations in the center of gravity for Quasi-Static Loads.
- Sine spectra of the acceleration for low frequency transient or harmonic loads.
- Power Spectral Densities of the acceleration for random vibrations.
- Shock Response Spectra of the acceleration for shock loads.

Establishing equivalence between the real dynamic environments and the environments used in analysis and verification processes involves comparing the structural responses in each case. This process is complex and has some limitations that lead to conservative approaches. This means that tests will have to be more severe than real operating conditions to account for the uncertainties involved in the establishment of equivalences.

2.2.1. Equivalence between dynamic loads and Quasi-Static Loads

Quasi-static loads are equivalent static loads for combinations of static and dynamic loads and are typically expressed as accelerations in the center of gravity of the structure.

The equivalence is based on the fact that the QSL will be defined for the most severe combination of loads found in the dynamic environment. This means that the QSL is basically maximum and minimum values of the acceleration. If done correctly the QSL should reproduce similar interface forces than the dynamic loads.

Even though the use of QSL is a common practice, it has some limitations that should be remarked. Firstly, in many situations, the QSL has been obtained taking only into account static accelerations and low frequency transients. Consequently, in these cases the QSL cannot predict relevant effects of high frequency vibration. Moreover, QSL implies a "quasi-rigid" behavior and therefore it is not an appropriate representation for dynamic loads in which non–primary modes are relevant.

2.2.2. Sine vibration equivalence, Equivalent Sine Input (ESI)

The verification of the low frequency transient loads is commonly done with "equivalent" sine inputs during analysis and testing. The main reason why this is done is that it is an easy way to simulate the flight transient behavior in electrodynamic shakers.

A detailed explanation of the concept and the application of the ESI will be given later on in *section 3.3.*

2.2.3. Random vibration and vibro-acoustics equivalences

Different random vibration environments are considered equivalent if they have the same durations and are represented by the same PSD. Using the RMS value of an input acceleration to evaluate its severity is a common mistake since it depends strongly on the PSD at really high frequencies that are usually irrelevant.

Equivalence between random vibration environments and vibro-acoustic environments using the PSD can also be done and have important practical implications. However, in this case the equivalence is more limited due to their significant physical differences.

2.2.4. Shock equivalence

Two mechanical shocks will be equivalent if they are represented by the same Shock Response Spectrum. The duration of the shocks should also be similar since the SRS definition does not allow duration comparison.

In some situations, equivalence between random vibrations and shocks are also established. This is based on the idea that both phenomena are responses to a single event, being the main difference the duration of the event. In this case the PSD that defines the acceleration in the random environment is converted into a kind of SRS called the random response spectrum RRS.

2.3. Limitations of the verification process

The methodologies that have been explained up to this point have proved to be reliable and are widely used in the space industry. However, there are still some limitations that should be mentioned:

- Vibration and shock generating equipment continue to test structures one axis at a time while real dynamic environments involve multiple-axis loads.
- In practice, tests simulate environments one by one even if they happen simultaneously.
- Simulation and testing of low frequency transients done with the sine-test. The equivalence assumed between environments generates unwanted margins and the long duration of the test is inconvenient.
- Simulation of the vibro-acoustic environment with random vibration tests even though there are important physical differences between both environments.
- The common practice of using the interface accelerations from flight data as input accelerations for testing ignoring the existence of antiresonances. This practice and the use of "infinite" mechanical impedance in the shaker constitute the main source of over testing in the verification process.

Trying to tackle of all the above-mentioned limitations in a single project is too ambitious. The present work will focus on studying alternative methods to define the low frequency loads specification which is currently based on the sine-test.

2.4. Disadvantages of sine-sweeps and possible alternatives

Currently, the low frequency loads specification is given by sine-testing based on swept-sine excitation signals applied axis by axis in electrodynamic shakers. Such a process has several disadvantages:

- Long duration of the test. Ground vibration tests are on the critical path of aircraft and spacecraft final assemblies [6].
- An initial knowledge of the modal behavior is necessary so that the structure is not over tested by being excited at resonant frequencies.
- The value of the sweep rate can influence the results if not chosen adequately.

- The established equivalence between sine-sweeps and real low frequency vibrations is questionable given the obvious differences between both environments.
- During flight, most severe conditions are caused by transients. Establishing equivalence between transients and sines leads to the generation of unwanted margins.

Defining new excitation signals that help to overcome the limitations imposed by the sine-sweeps is necessary. These new excitation signals have to be more representative of real flight transients.

The synthesis of equivalent acceleration time histories is an important and developed field from which possible alternatives to the sine-sweep already exist. The present work will study two of these alternatives, the Fast Sine Sweep and the Wavelets methods. Both methods are able to synthesize signals that are more similar to flight transients than the Classical Sine Sweep. However, to determine whether these methods are appropriate to define the low frequency loads specification for satellites, the response of a simplified model of a satellite to the synthetic signals obtained with these methods has to be compared with the response of the model to real flight transients. If the response of the model is similar in both situations then it can be concluded that the equivalence is assumable.

Chapter 3
Synthesis of acceleration time histories based on Shock Response Spectrum

During flight, the resulting transient accelerations on a satellite will be waveforms that are too complex to be described as mathematical functions. Generally, other tools are needed to assess the impact of transient loads on satellites when carrying out analysis verification. Moreover, reproducing these accelerations in an electrodynamic shaker during a qualification test is really difficult. Consequently, another approach is also required for testing.

The Shock Response Spectrum (SRS) is a mathematical tool that is frequently used to estimate the potential damage that a complex transient can cause. Engineers use the SRS to establish equivalences between different acceleration time histories. The assumption is that two acceleration time histories will have similar impact on a structure if their SRS are similar.

Using the concept of the SRS, engineers can establish equivalence between the real acceleration time histories and easier synthetic acceleration time histories that can be used during the analysis and testing stages.

3.1. Model for the calculation of the SRS

The SRS is a function that is calculated by applying an acceleration time history as a base excitation $\ddot{Y}$ to a combination of single-degree-of-freedom systems. $\ddot{X}_i$ is the resulting acceleration and M_i, K_i, C_i are the mass, the stiffness and the damping of each individual system, although the damping of all systems is typically assumed to be the same. If the damping of the studied structure is unknown, it is common to use a value of $Q = 10$, where $Q = \frac{1}{2\zeta}$ and $\zeta = 5$ %. The natural frequency, however, is unique for each of the SDOF systems.

Taking this model into consideration, the SRS will be calculated as the peak acceleration response of each SDOF system to the input base acceleration. The result is a curve that gives peak acceleration as a function of the natural frequency.

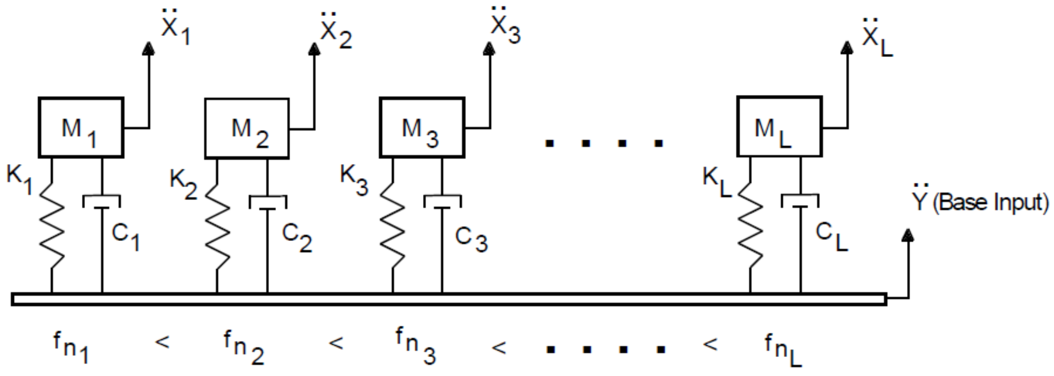

Figure 3.1: Shock Response Spectrum model

Source: An introduction to the Shock Response Spectrum [5]

In order to calculate the peak acceleration response of each SDOF system, a convolution integral approach is used to solve the equations of motion. A complete demonstration of the resulting formula for the peak acceleration, *equation 1*, can be found in [5].

$$\begin{aligned} \ddot{x}_i = {} & 2\exp[-\zeta\omega_n\Delta t]\cos[\omega_d\Delta t]\,x_{i-1} - \exp[-2\zeta\omega_n\Delta t]\,x_{i-2} + 2\zeta\omega_n\Delta t\ddot{y}_i \\ & + \omega_n\Delta t\exp[-\zeta\omega_n\Delta t]\left\{\left[\frac{\omega_n}{\omega_d}(1-2\zeta^2)\right]\sin[\omega_d\Delta t]\right. \\ & \left. - 2\zeta\cos[\omega_d\Delta t]\right\}\ddot{y}_{i-1} \end{aligned} \tag{1}$$

3.2. Implementation of the Shock Response Spectrum

In the present work, the calculation of the SRS will be done with *MATLAB.* Many codes performing this calculation are available on the web. Since writing a new one would have been time-consuming, this work will make use of a script written by Tom Irvine (I).

Even though the source is reliable, an example of the calculation of the SRS is given below. In this example the input acceleration will be a half sine pulse. The Shock Response Spectrum of this signal is available in the literature and will serve as a validation of the correctness of the script.

The half sine transient is very used for representing shocks due to its simplicity. The obtained Shock Response Spectrum when using $Q = 10$, is shown in *figure 3.3.*

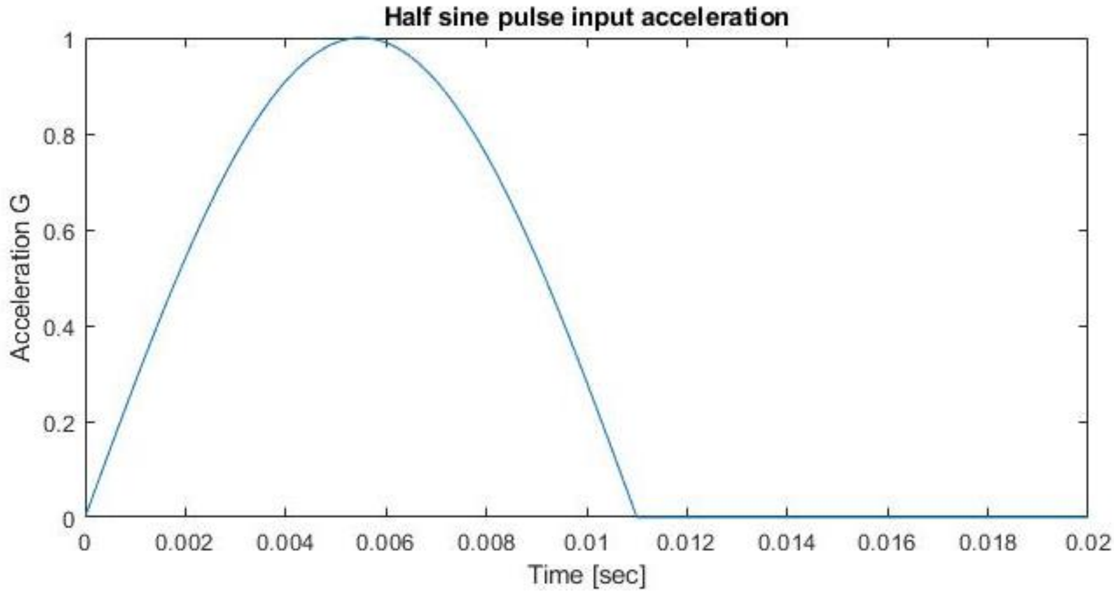

Figure 3.2: Half sine pulse

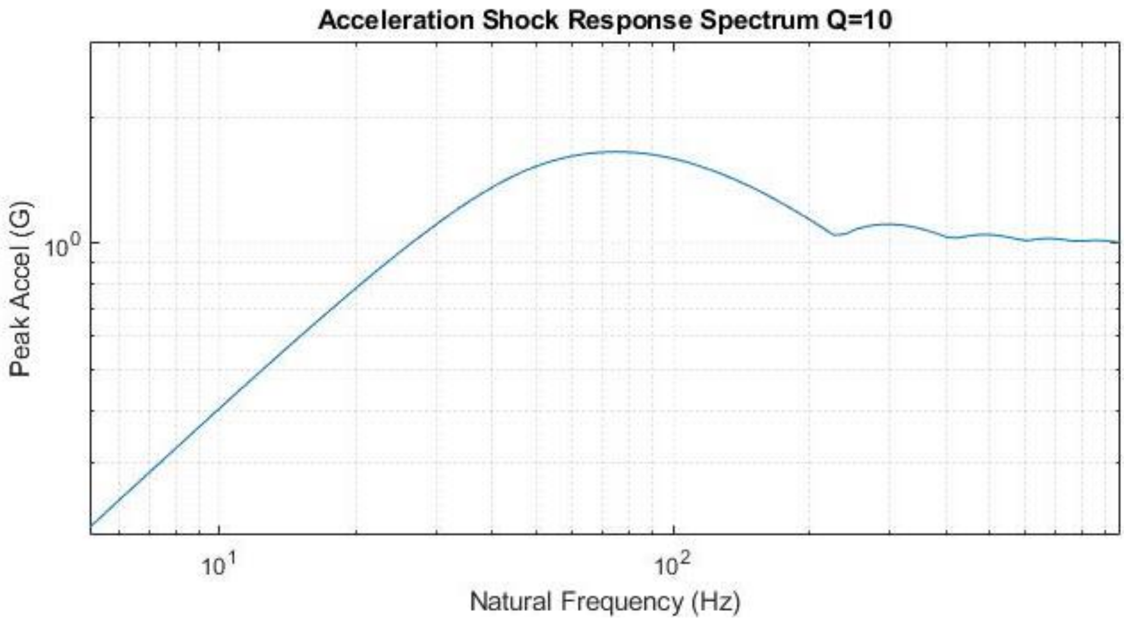

Figure 3.3: Shock Response Spectrum of a half sine pulse

Once the Shock Response Spectrum of a flight transient has been calculated, there are different approaches to synthesize signals that have a similar SRS. This study will focus in three of these methods:

- Classical sine sweep, ESI
- Wavelets
- Fast sine sweep

3.3. Classical Sine Sweep, Equivalent Sine Input (ESI)

This approach is applied to establish the equivalence between a low frequency transient acceleration and what is known as the Equivalent Sine Input. The main reason why the ESI is so commonly used is that the resulting signal can be easily applied in an electrodynamic shaker.

The Equivalent Sine Input for a given frequency f of a transient acceleration $\ddot{u}(t)$, is defined as the sinusoidal excitation at the base that would make the system reach an acceleration amplitude equal to the one of the Shock Response Spectrum (SRS(f)) that describes the response of the structure to $\ddot{u}(t)$ at frequency f.

The following steps are performed to obtain the ESI:

- Determine the interface accelerations of the satellite with the coupled load analysis.
- Obtain the acceleration Shock Response Spectrum. Since the ESI aims to study the low frequency environment, the SRS will be calculated only for low frequencies.
- Calculate the Equivalent Sine Input for each frequency of the SRS using the following expression [1]:

$$ESI(f) = SRS\ (f)/\sqrt{Q^2+1} \qquad (2)$$

Once the Equivalent Sine Input is known for each of the frequencies of the Shock Response Spectrum, it is possible to create a sine sweep. The sine sweep is a sine with variable frequency and amplitude. The amplitudes for each frequency are those specified by the ESI. The frequency will go from low to high values and the velocity with which it variates will be called the sweep rate.

In the Classical Sine Sweep, the sweep rate is slow enough to respect the steady-state assumption. If the sweep rate is too fast, the response does not reach the steady-state at each frequency.

3.3.1. Generation of a Classical Sine Sweep that matches a desired SRS

In this example, the ESI concept will be used to synthesize an acceleration time history that has a SRS that matches the SRS obtained from a fictitious acceleration time history. This fictitious acceleration time history is shown in *figure 3.4.*

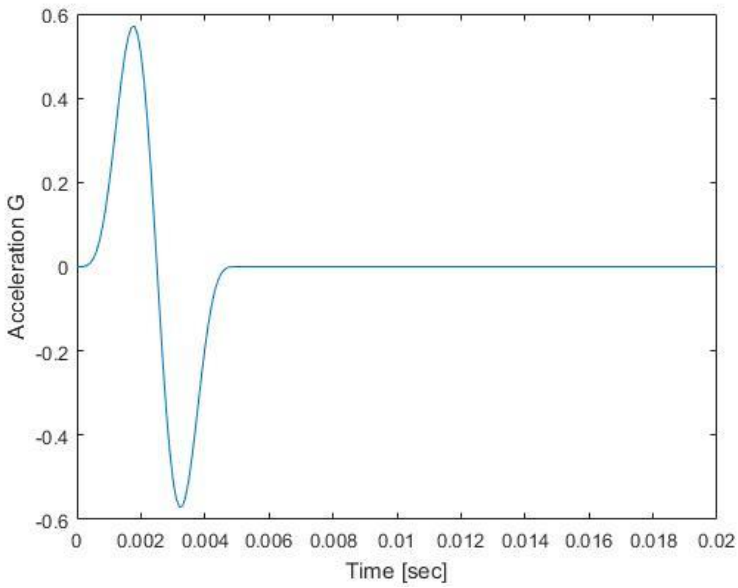

Figure 3.4: Fictitious acceleration time history

The resulting synthesized signal will be a sine with a frequency that varies with time and with different amplitudes for each frequency, in other words, a sine sweep. The amplitude for each frequency can be calculated using the ESI definition provided in last section. As a simplification, we will exaggerate this synthesized signal by maintaining each of the frequencies of the original SRS calculation for a certain time, instead of constantly sweeping the frequency. In this particular case, each frequency was maintained for 5 seconds.

The calculation and plotting of the ESI was done in *MATLAB*. The script can be found in *Annex C*. The obtained synthesized signal is given in *figure3.5.*

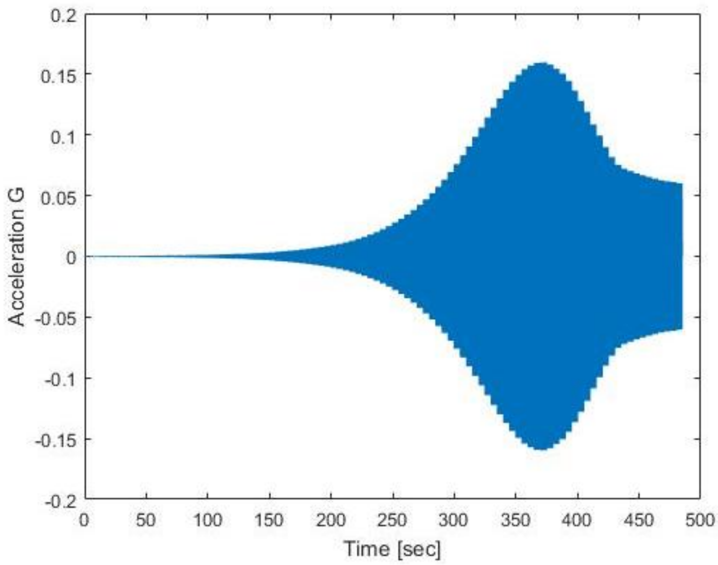

Figure 3.5: Equivalent Classical Sine Sweep

Now the SRS of the synthesized signal can be obtained to check if it is similar to the one of the departing fictitious signal. A comparison of both SRS is shown in *figure3.6.* As it can be seen, the obtained SRS is almost identical to the one of the original signal.

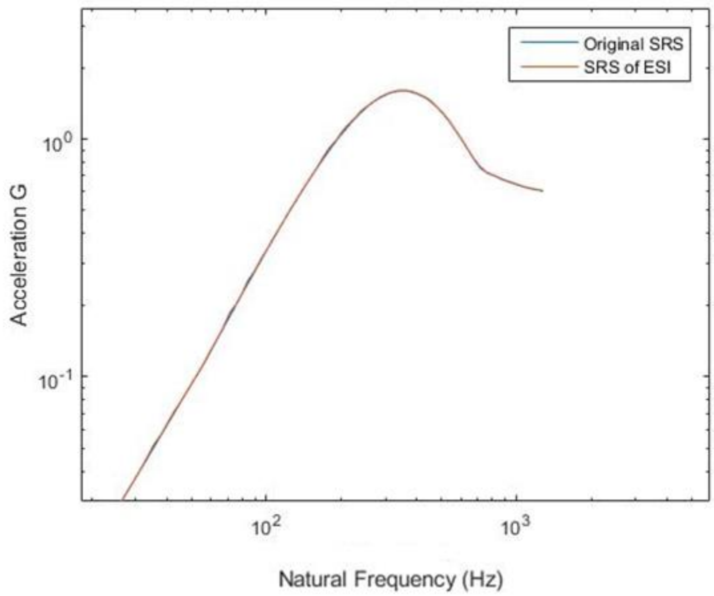

Figure 3.6: SRS comparison between fictitious transient and Classical Sine Sweep

The studied case is an example of how it is possible to match a SRS with a signal that is completely different from the original. The original signal lasted for only 20 milliseconds and the synthesized signal lasted for more than 450 seconds. This exaggeration shows that the SRS concept has to be used carefully, since the ideal would be to synthesize signals with similar characteristics to the original.

3.4. Fast Sine Sweep as an alternative to Classical Sine Sweep

In the previous section, it has been shown how a long duration sine sweep can have a similar Shock Response Spectrum than a short duration flight transient. However, this sine sweep is not a realistic representation of what is actually happening in the original transient. Establishing equivalence between these two different environments leads to unwanted margins. As an alternative to the Classical Sine Sweep, the Fast Sine Sweep approach attempts to reduce these margins by generating a sine sweep that represents better the transient environment and that leads to more realistic tests. This approach and its results were first presented at ECSSMET 2016 in Toulouse [7]. The information given in this section is based on this presentation.

In the Fast Sine Sweep method, the equivalence between environments is still based on the concept of Shock Response Spectrum. However, the new approach is to preserve the most important properties of the original environment. This basically means preserving levels, durations and frequency content. Taking this into account, the method proposes to use amplitude and sweep rate as the main parameters to match the SRS. The synthesized signal will be of the form of *equation 3.*

$$\ddot{u}(t) = A(t) \cdot \sin(E(t)) \tag{3}$$

Where $A(t)$ and $E(t)$ are functions of time used to modulate the amplitude and the phase of the sine respectively. $E(t)$ is related to the sweep rate $V(t)$ since $f(t) = \frac{1}{2\pi}\frac{dE(t)}{dt}$ and $V(t) = \frac{df(t)}{dt}$. The objective will be to optimize $A(t)$ and $V(t)$ to match the SRS of the original transient. $F(x)$ can be defined as the function accounting for the error when matching the SRS. In [7] the proposed optimization is based on a minimization of $F(x)$ using non-linear least square methods and Taylor expansions of $F(x)$.

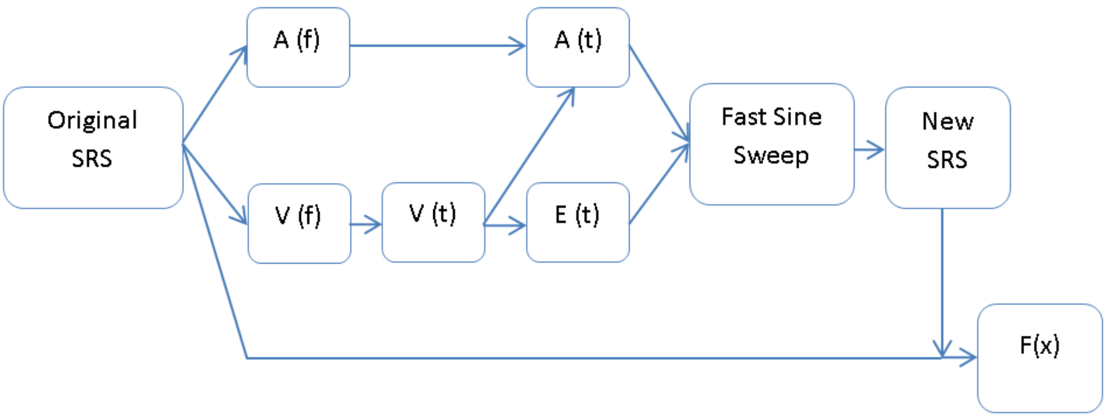

Figure 3.7: Steps to implement the Fast Sine Sweep method

The different steps of this approach are summarized in *figure3.7.* In first place, different values for the amplitude and for the sweep rate are chosen for different frequencies. Since the sweep rate is the variation of the frequency with time, integration can be performed to obtain amplitude and sweep rate as a function of time. Once we have the phase and the amplitude, the new synthesized signal can be obtained with *equation 3*. Finally, the SRS of the new signal can be calculated and compared with the original. New values for the amplitude and the sweep rate will be selected until the error is sufficiently small.

3.4.1. Implementation of the Fast Sine Sweep method

The aim of this section is to use the Fast Sine Sweep method to match the SRS of the acceleration time history used in the previous section (*figure 3.4*). After presenting the results, we will analyze whether this approach is more appropriate than the Classical Sine Sweep.

The implementation was done in *MATLAB* and is mostly based on the theoretical description of the method given by [7]. The amplitude and sweep rate will be the parameters to optimize in order to match the SRS. However, the optimization method will not be the one presented in [7]. This method was mathematically complex and therefore the following alternatives were investigated:

- Trial and error optimization
- Optimization with genetic algorithms
- Combination of trial and error and genetic algorithms

3.4.1.1. FSS with trial and error optimization

The first implementation of the method was done with a trial and error optimization of the amplitude and sweep rate values. This alternative was the simplest one and therefore the most appropriate to reach a better understanding of the method.

The necessary input values are the values of the Shock Response Spectrum for different natural frequencies. These values are given as an array $SRS\,(f)$. In a first attempt the following steps were considered:

- Create $A\,(f)$ and $V\,(f)$ arrays with the same dimensions than $SRS\,(f)$.
- Define maximum and minimum values for the amplitude and the sweep rate and a desired resolution for both parameters.
- Try all possible combinations of amplitudes and sweep rates, calculate the Fast Sine Sweep and its Shock Response Spectrum compare it with the original and stay with the combination that minimizes the error.

While implementing this first approach a main difficulty was found. The required computation time was too high since there were a lot of possible combinations. A possibility to reduce the computation time was to use a smaller resolution for amplitude and sweep rate. However, when fixing $V\,(f)$ and only optimizing $A\,(f)$, it was found that the best results were those in which the amplitude had a small variation with time. Instead of changing suddenly from one frequency point to the next one, different sections with similar amplitude values could be identified along the time dimension. The same applied when fixating $A(f)$ and optimizing V (f). Taking this finding into account, a new approach was considered:

- Create $A\,(f)$ and $V\,(f)$ arrays with the same dimensions than $SRS\,(f)$.
- Define maximum and minimum values for the amplitude and the sweep rate, a desired resolution for both parameters and a desired number of sections in the frequency domain.
- Try all possible combinations of amplitudes and sweep rates with the condition that all the points belonging to the same section have equal amplitude factor and sweep rate. Stay with the combination that minimizes the error.

By dividing the frequency domain in relatively big sections the computation time was drastically reduced. The process of determining how many sections to use was also a trial and error process.

When applying the FSS approach with trial and error optimization, the following synthesized signal was obtained:

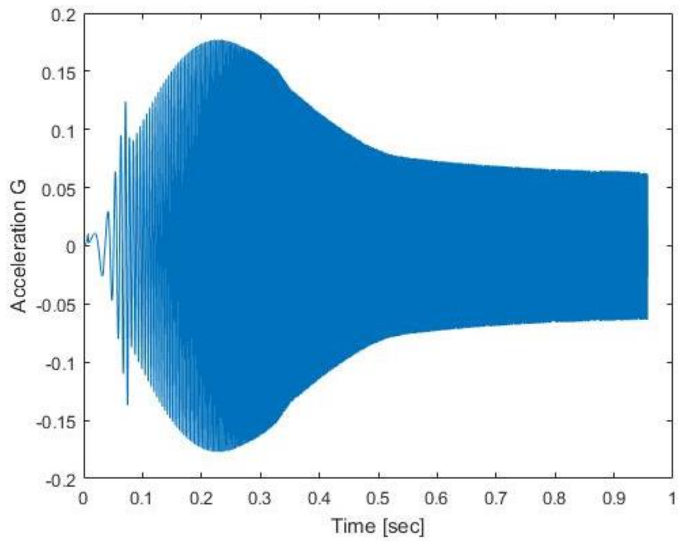

Figure 3.8: Acceleration time history obtained with FSS (trial & error optimization)

After calculating the acceleration time history with the Fast Sine Sweep approach, the Shock Response Spectrums of this signal and the original can be compared. In *figure 3.9*, it can be seen that the resulting Shock Response Spectrum stays within a reasonable tolerance.

It is important to remark that although now the synthesized signal is more similar to the original transient in terms of levels and durations, in comparison with the ESI method, there are still significant differences, see *table 3.1*. This happens because the optimization is selecting those sweep rates that lead to minimum error without considering the duration of the signal important. Additional attempts were done in which the dime duration of the signal was considered as a constraint. However, a shorter duration led to unacceptable increases in the total error.

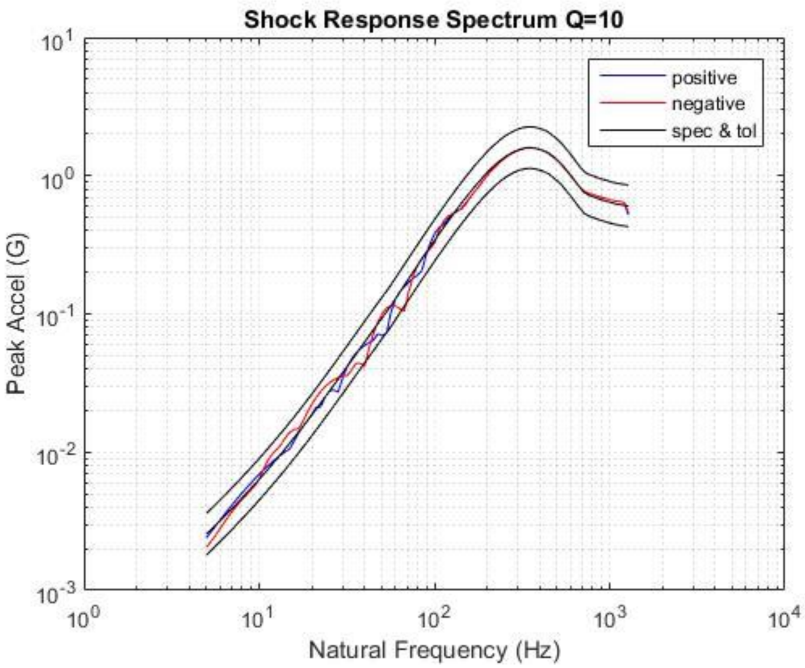

Figure 3.9: SRS comparison between original fictitious transient and FSS (trial & error)

	Original transient	Synthesized signal	ESI
Maximum Amplitude [G]	0.57	0.18	0.15
Duration [s]	0.02	0.95	470

Table 3. 1: Original transient VS FSS (trial & error)

3.4.1.2. Genetic algorithm optimization

Genetic algorithms (GA), is an optimization technique based on the mechanisms of genetics and natural selection [8]. Over the past years, GA has received a lot of attention due to its novelty and multiple advantages to deal with complex optimization problems.

The optimization starts with a set of randomly generated solutions called *population*. Each individual solution, which will be called a *chromosome*, evolves through several iterations, which will be called generations. Chromosomes will be evaluated according to one or multiple fitness functions in each generation. Those chromosomes that perform the best according to the fitness function will be selected as *parents* while the others are rejected. The next generation will be made of new chromosomes that can be created by combining two of the existing chromosomes (*crossover operation*), or by modifying a chromosome (*mutation operation*). In each new generation, the number of newly created chromosomes is the same than the number of rejected chromosomes in order to keep the size of the population constant. Ideally, after some generations, the best chromosome is found, which represents an optimal solution to the problem.

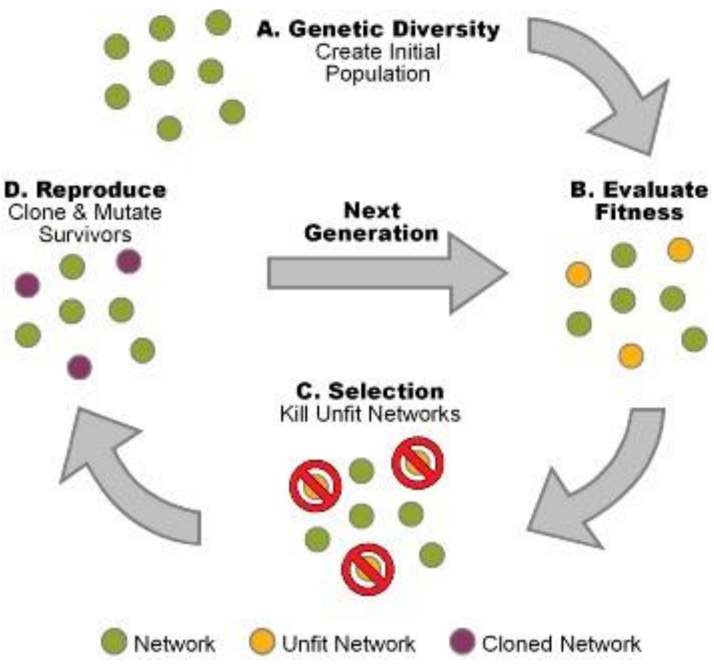

Figure 3.10: Genetic Algorithm scheme

Source: https://quantdare.com/ga-to-define-a-trading-system/

GA has three main advantages with respect to other optimization techniques. The most important one with respect to the scope of this project is its adaptability. GA can be applied to every type of problem regardless of its mathematic representation or complexity (linear/ non-linear, discrete/continuous etc.). In the FSS problem, the concept of Shock Response Spectrum already has a complex mathematical representation. Moreover, the solution we pretend to find is a set of amplitudes and sweep rates for each natural frequency of the SRS, which results in a solution vector with a significant dimension. The other two advantages of GA are robustness and flexibility. Robustness in the sense that GA has been proved to be efficient when performing global search and flexibility because it can be combined with any other heuristic to improve the optimization.

With the purpose of improving the trial and error optimization implemented in the previous section, GA have been used. This has been done with *Matlab`s Genetic Algorithms Toolbox*. The fitness function for this optimization is the error of the new SRS with respect to the original one.

However, when performing the GA optimization, it was soon realized that the required computation time to find a reasonable solution would be too high. The algorithm did not converge in any of the different attempts. The main problem was the high number of variables in the problem. It is not just one parameter that has to be optimized, but rather two parameters (amplitude and sweep rate) for each natural frequency considered in the SRS curve. As an example, in a SRS curve in which 100 points have been considered, the algorithm has to optimize at least 200 variables.

In order to solve this problem, it was decided to divide the optimization in two major stages. In the first stage, a trial and error optimization with low resolution in amplitude and sweep rate will be done. This approximated solution would now serve as an already good initial solution for the second stage in which GA was used to reach a better approximation.

Using GA for the second stage of the optimization led to an improvement of the error. However, as it can be seen in the convergence of the Genetics Algorithm (*figure 3.11*), this improvement is not as relevant as it was expected. *Figure 3.12* shows the SRS comparison for the found solution.

Figure 3.13 shows the resulting acceleration time history obtained with the Fast Sine Sweep approach using optimization based on a combination of trial and error and genetic algorithms.

Error improvement was not the only goal in this process. Keeping the levels and durations similar to the original signal was also desirable. However, adding the second stage of GA gives us little additional control over durations and levels.

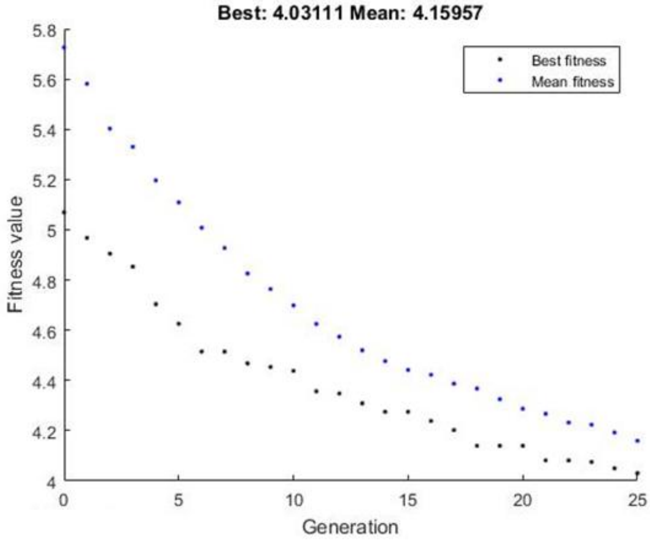

Figure 3.11: Genetic Algorithm convergence when optimizing the FSS

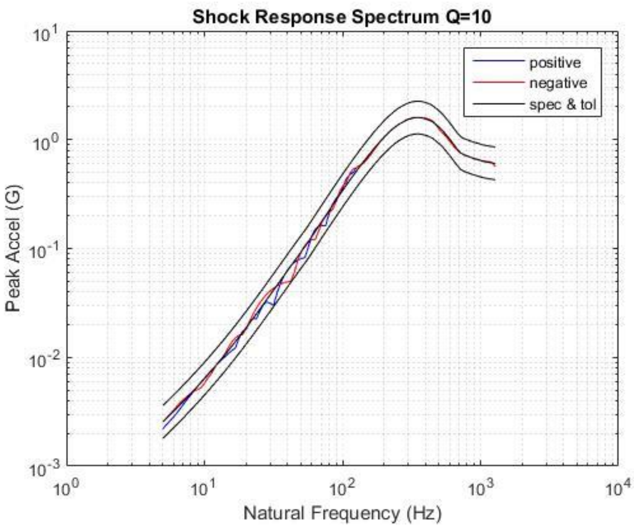

Figure 3.12: SRS comparison fictitious transient VS FSS (Trial & error with GA)

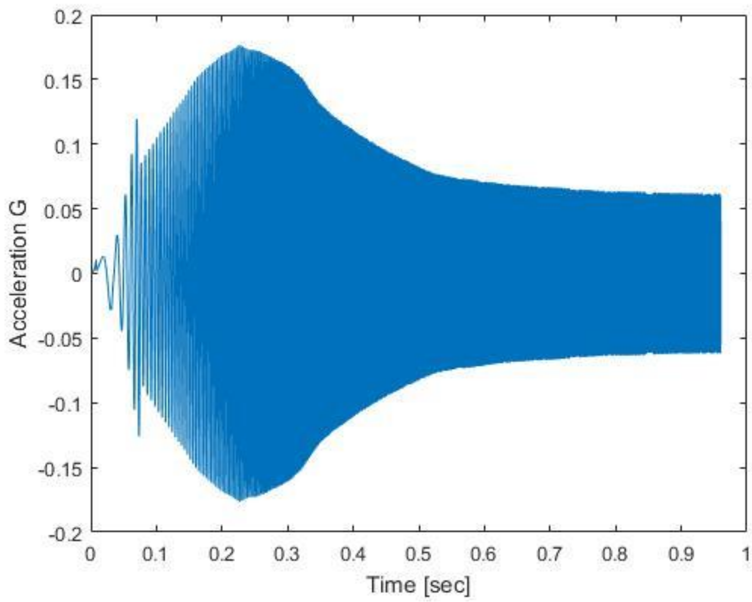

Figure 3.13: Resulting FSS (Trial & error with GA)

3.5. Wavelets as an alternative to Classical Sine Sweep

Wavelet theory has been used in many different applications in the field of signal processing. It was in fact independently developed in each of those applications, resulting in different views of what could be considered a single theory. The development of the Wavelet Transform concept is one of the most important examples. Its development provided an alternative to the Short-Time Fourier Transform to analyze non-stationary signals by decomposing signals onto a set of basic functions called wavelets.

One of the particularly interesting applications of wavelets for this study is synthesis of signals. Wavelets can be synthesized into a time history to match a given Shock Response Spectrum. In this section we will study if it is in fact a better approach than the classical sine sweep (ESI) or the Fast Sine Sweep.

The mathematical description of an individual wavelet is complex and this work will not get into much detail regarding its explanation. The formulation of the wavelet used in this section is given by *equation 4*.

$$W_m(t) = \begin{cases} 0, for\ t < t_{dm} \\ A_m \sin[\frac{2\pi f_m}{N_m}(t - t_{dm})] \quad \sin[2\pi f_m(t - t_{dm})], for\ t_{dm} \leq t \leq [t_{dm} + \frac{N_m}{2f_m}] \\ 0, for\ t > [t_{dm} + \frac{N_m}{2f_m}] \end{cases} \tag{4}$$

- $W_m(t)$ is the acceleration of wavelet m at time t
- A_m is amplitude of the acceleration of wavelet m
- f_m is frequency of wavelet m
- N_m is the number of half sines in the wavelet m
- t_{dm} is the time delay of the wavelet m

3.5.1. Implementation of the Wavelets method

With the purpose of comparing this approach to the already presented methods, an already existing code for *MATLAB* was used. This code belongs to *Tom Irvine*. This code needs a SRS input and uses it to synthesize a signal by combining basic wavelets. The input SRS will be the same than the one studied in the previous methods (*figure 3.4*).

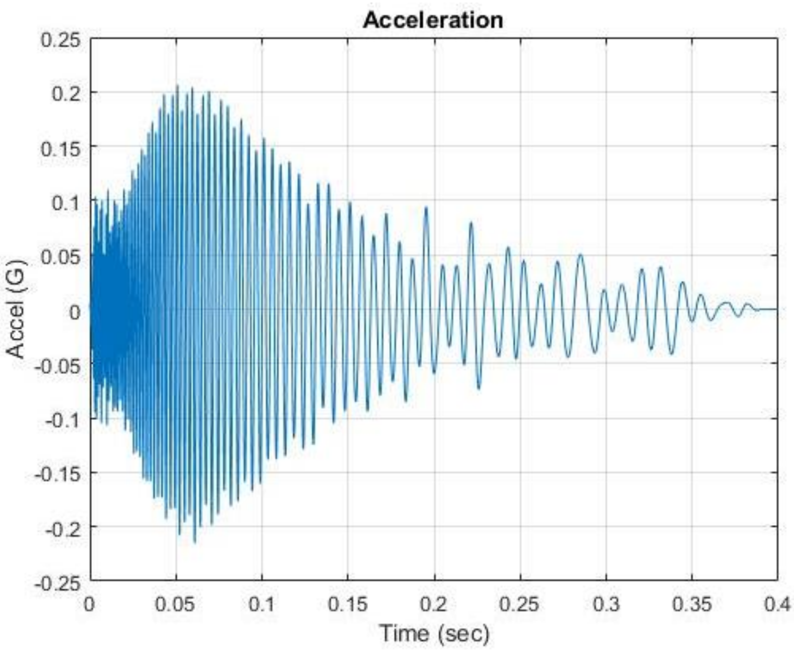

Figure 3.14: Acceleration time history obtained with Wavelets method

The resulting synthesized acceleration time history is shown in *figure3.14*. The signal has a similar shape to the one obtained with the FSS method but has a shorter duration and higher amplitudes. The error is still really small. As it can be seen in *figure 3.15*, matching the original Shock Response Spectrum was not a problem.

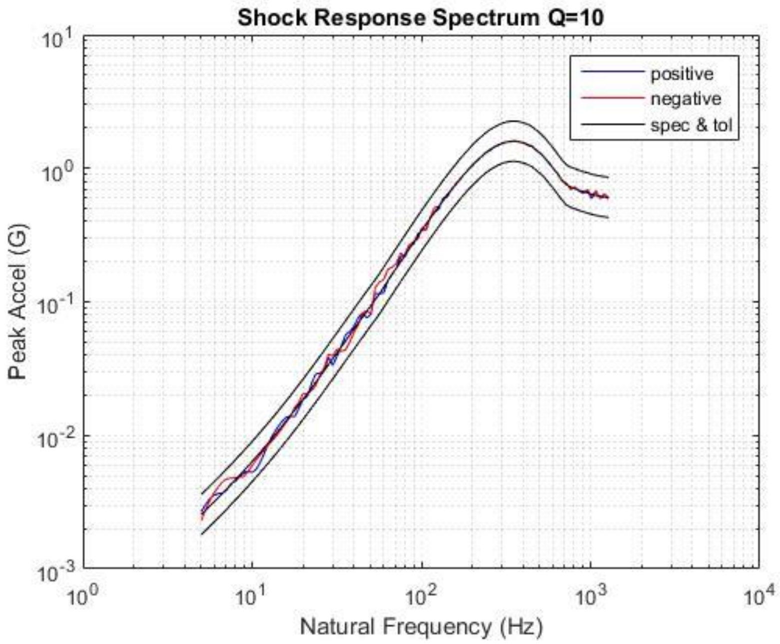

Figure 3.15: SRS comparison, fictitious transient VS Wavelets method

3.6. Conclusions

This chapter has studied different approaches to synthesize signals that match a given Shock Response Spectrum. The purpose behind these methods was to find acceleration time histories that cause equivalent damage than real flight transients.

The first method that has been studied is the Classical Sine Sweep or ESI. This method has been a common practice in the verification of satellites for a long. However, as it was explained earlier, the acceleration time histories obtained with this method have really long durations in comparison with the real flight transients they try to represent. Establishing equivalence between environments with really different characteristics led to undesirable margins.

Two additional alternatives to the classical sine sweep have been considered. Both of them try to create signals that are more similar to the original transients in terms of levels and durations. The summarized results of the studied examples are given in *table 3.2*. These results show that it is difficult to decide which of the two methods will be better. The FSS method led to smaller errors than the wavelets. However, the wavelets method could generate signals with durations and levels that were closer to the original transient.

The common disadvantage of these two new alternatives is the increased complexity of the synthetic signal, which makes tests in electrodynamic shakers harder. While a demonstration of the feasibility of the FSS on standard shaker can be found in [7], the compatibility of wavelets still has to be studied in more depth.

	Original transient	**FSS**	**Wavelets**
Max Amplitude [G]	0.57	0.18	0.2
Duration [s]	0.02	0.95	0.4
Mean square error	-	4.03	5.65

Table 3. 2: Comparison of the different synthesis methods

In this chapter, the methods have been compared according to their ability to match a given SRS. However, the fact that the resulting synthesized time histories will faithfully represent the real transients because of the similarities between the SRS was an assumption. In order to determine whether this assumption is acceptable, in following chapters these methods will be compared by carrying out simulations on a simplified model of a satellite.

Chapter 4
Development of a FEM of a satellite and preparation for dynamic analysis

The finite element analysis for this study will be carried out in *AFECTOS* in *MATLAB*. *AFECTOS* is a non-commercial finite element program developed for *MATLAB* by the *Institute for Research in Technology* of *Universidad Pontificia Comillas*. The program is a tool for solving different types of structures using the Finite Element Method (FEM).

The first step will be to create a finite element model of a simplified satellite that works in *AFECTOS.* A replica of the model will be created in *ANSYS WORKBENCH*. A Static Structural Analysis and a Modal Analysis will be carried out in both finite element programs in order to validate that the model created in *AFECTOS* is correct and that the results of more complex analysis will be reliable.

After this validation, dynamic simulations will be prepared for *AFECTOS.* These simulations will basically consist of a satellite being loaded at the base with an excitation. A program compatible with *AFECTOS* that carries out the time integration to calculate the dynamic behavior of the structure will be developed. The program will simulate the conditions during a test in an electrodynamic shaker. The time integration method used will be Newmark Beta method.

The chapter will finished with a brief explanation of Rayleigh damping. The assumption of proportional damping will be made to simplify the calculations of the dynamic analysis.

4.1. Description of the FEM of a simplified satellite

The finite element model described here is not developed to design a real satellite. A detailed and realistic calculation of the loads is completely unnecessary regarding the objectives of this work. The main purpose is to compare different verification approaches. Therefore, a really simple model will be enough.

The model will consist of one vertical bar and four horizontal bars. The horizontal bars will have different sections. At the extremes of the horizontal bars, different point masses will be added.

The dimensions and masses of the whole structure should be similar to the ones of a normal satellite. For this study the total mass of the structure will be around *2000 kg* including the point masses ($m_1 = 87.5\ kg, m_2 = 75\ kg, m_3 = 37.5\ kg, m_4 = 50\ kg$). The dimensions will be *2 m* for the vertical bar and *0.6 m* for the horizontal bars.

Given the simplicity of the geometry, the entire model has been created using beam elements with specified cross sections. *Figure 4.1* shows the geometry and the mesh of the model.

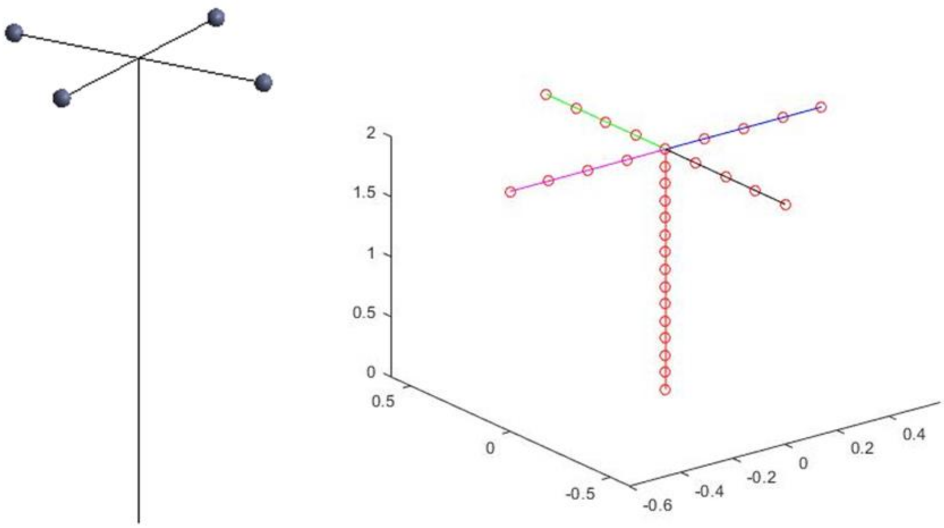

Figure 4.1: Geometry and mesh of the simplified model of a satellite

4.2. Modal analysis of the FEM

It is common practice to carry out a modal analysis as a starting point before other more complex analysis. This analysis determines which are the modes and the natural frequencies of a structure. Since the purpose of this chapter is to simulate a satellite loaded with a base excitation, knowing the vibration characteristics of the model is important to explain the response of the structure. Moreover, the damping definition for the dynamic analysis will depend on the natural frequencies of the model.

In order to validate that the model was correctly programmed in *AFECTOS*, the modal analysis has been done both in *ANSYS WORKBENCH* and *AFECTOS.* The results of both programs were the same so we can assume that the model is correct. The results are shown below. *Figure 4.2* gives the natural frequencies of the satellite and *figure 4.3* shows the mode shapes associated to those natural frequencies.

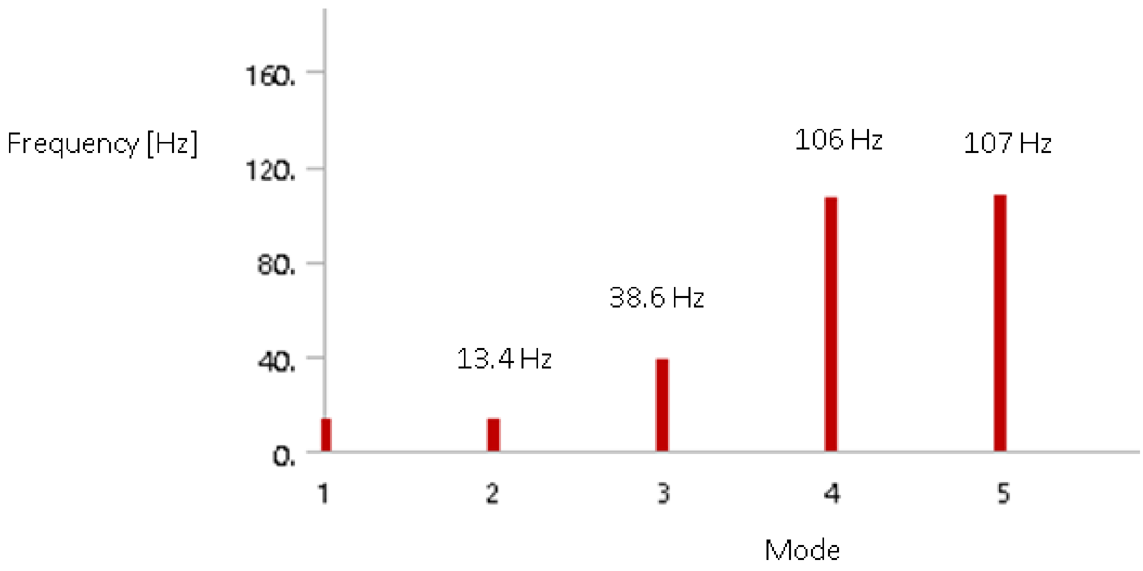

Figure 4.2: Natural frequencies of the satellite

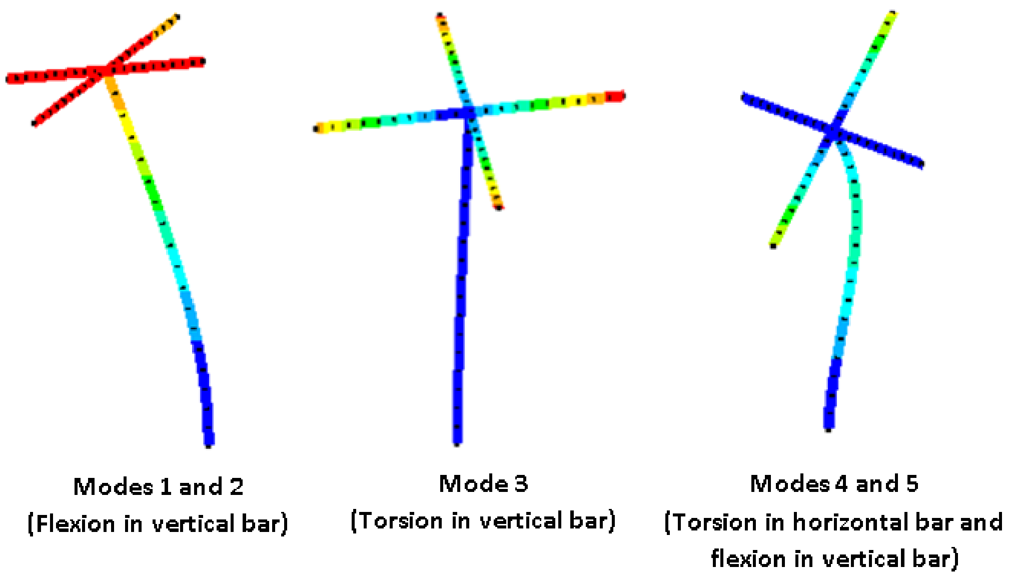

Figure 4.3: Mode shapes of the satellite

4.3. Static structural analysis of the FEM

A static structural analysis of the simplified satellite has also been carried out in *ANSYS WORKBENCH* and *AFECTOS* before the dynamic analysis. This previous step will help to verify that the response of the model to loads is normal.

In this analysis, the satellite will be clamped at its base and subjected to a force at the end point of one of the horizontal bars. A representation of these conditions can be seen in *figure 4.4. Figure 4.5* shows the deformation of the structure.

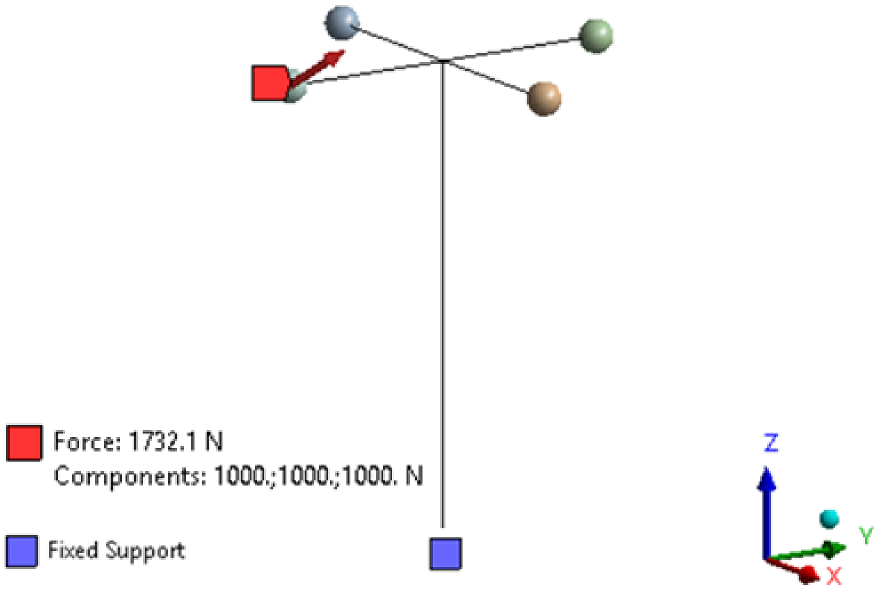

Figure 4.4: Load conditions in static structural analysis

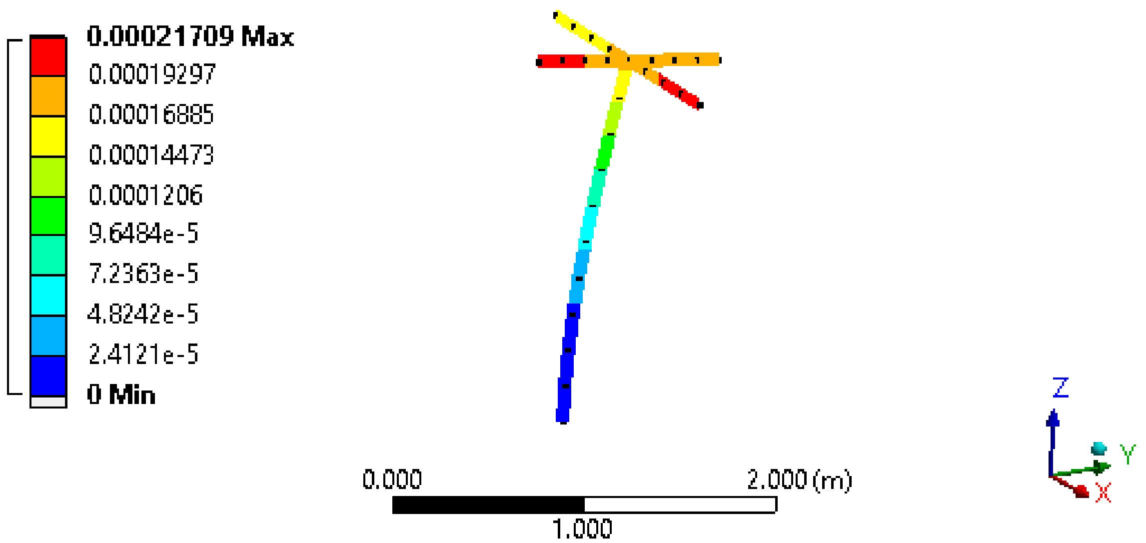

Figure 4.5: Deformation of the satellite under load conditions of figure 4.4

Table 4.1 shows the displacements of the point subjected to the load. As it was expected, both programs give the same results. This reaffirms that the model is correctly defined in *AFECTOS*.

	AFECTOS	***ANSYS***
Displacement in X direction [mm]	0.141	0.143
Displacement in Y direction [mm]	0.145	0.146
Displacement in Z direction [mm]	0.073	0.074

Table 4. 1: Displacements of loaded point of the satellite

4.4. Newmark-beta method

Even though *AFECTOS* already had a Newmark solver available, a different version of the solver has been developed. The main reason is that the original formulation was not that convenient for the specific case of a structure subjected to ground acceleration. Moreover, from an academic point of view, programming a new version was really interesting to gain a better understanding of time integration methods in transient analysis.

4.4.1. General description of the Newmark-beta method

The Newmark-beta method is a widely known step-by-step integration method for dynamic analysis [9]. Step-by-step integration methods are commonly used for solving the equations of motion of structures subjected to dynamic environments.

For a multiple degree of freedom (MDOF) system the equation of motion can be expressed as:

$$M\ddot{u} + C\dot{u} + Ku = f \tag{5}$$

Where:

- M, K and C are the mass, stiffness and damping matrices.
- $u, \dot{u}$ and $\ddot{u}$ are the displacement, velocity and acceleration vectors.
- $F$is an external force vector

Newmark method can be used to calculate the approximate solutions of the equations of motions. The general formulation is presented below:

1. Initial calculations:

1.1. $\ddot{u}_0 = \frac{p_0 - c\dot{u}_0 - ku_0}{m}$

1.2. Select Δt

1.3. $\hat{k} = k + \frac{\gamma}{\beta \Delta t} c + \frac{1}{\beta \Delta t^2} m$

1.4. $a = \frac{1}{\beta \Delta t} m + \frac{\gamma}{\beta} c$ and $b = \frac{1}{2\beta} m + \Delta t (\frac{\gamma}{2\beta} - 1) c$

2. Calculations for each time step:

2.1. $\Delta \hat{p}_i = \Delta p_i + a\dot{u}_i + b\ddot{u}_i$

2.2. $\Delta u_i = \frac{\Delta \hat{p}_i}{\hat{k}}$

2.3. $\Delta \dot{u}_i = \frac{\gamma}{\beta \Delta t} \Delta u_i - \frac{\gamma}{\beta} \dot{u}_i + \Delta t (1 - \frac{\gamma}{2\beta}) \ddot{u}_i$

2.4. $\Delta \ddot{u}_i = \frac{1}{\beta \Delta t^2} \Delta u_i - \frac{1}{\beta \Delta t} \dot{u}_i - \frac{1}{2\beta} \ddot{u}_i$

3. Repetition for the next time step

For the average acceleration method, then $\gamma = \frac{1}{2}$ and $\beta = \frac{1}{2}$, for the linear acceleration method then $\gamma = \frac{1}{2}$ and $\beta = \frac{1}{6}$.

4.4.2. Implementation of the Newmark-Beta method

In the dynamic simulations of the satellite that will be done in this study, the input excitation will always be a ground acceleration. The problem with this is that the excitation is introduced in the same point where the boundary conditions are defined.

There are different ways to solve this situation. In this case, the decision was to follow a similar approach to the one that is normally used when studying the effect of earthquakes in structures.

The approach consists in working with displacements, velocities and accelerations relative to the ground. Considering the single degree of freedom system subjected to a ground acceleration shown in *figure 4.6*, the incremental equation of motion can be written as:

$$m\Delta\ddot{x}_i + c\Delta\dot{x}_i + k\Delta x_i = -m\Delta\ddot{x}_g \qquad (6)$$

The increment of displacement, velocity and acceleration relative to the ground is due to the increment of ground acceleration at each time step.

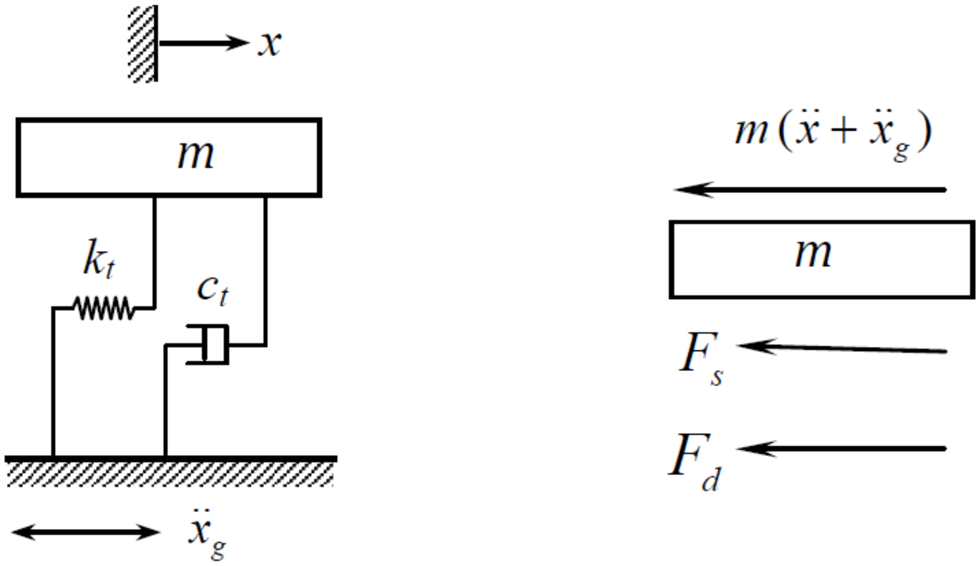

Figure 4.6: Single degree of freedom system subjected to ground acceleration

Source: Chapter 7, Non-linear Seismic Response of Structures [10]

In a multiple degree of freedom system, the approach will be applied in the same way. But now, not all the degrees of freedoms are affected equally. Only those degrees of freedom that represent the same direction in which the ground acceleration is acting will be affected. This is expressed by including the influence coefficient vector r, which has ones in the affected directions and zeros in the rest. Now, the incremental equation of motion for the MDOF system is expressed as:

$$M\Delta\ddot{x}_i + C\Delta\dot{x}_i + K\Delta x_i = -Mr\Delta\ddot{x}_g \qquad (7)$$

Taking this into account, we can now implement the Newmark Beta method for the specific case of the satellite subjected to a base excitation.

1. Departing from the known ground acceleration time history $\ddot{x}_g$, the initial calculations are:

   1.1. $\ddot{x}_0 = \frac{-Mr\ddot{x}_{g0} - C\dot{x}_0 - Kx_0}{M}$

   1.2. $\widehat{K} = K + \frac{\gamma}{\beta \Delta t} C + \frac{1}{\beta \Delta t^2} M$

   1.3. $a = \frac{1}{\beta \Delta t} m + \frac{\gamma}{\beta} c$ and $b = \frac{1}{2\beta} m + \Delta t (\frac{\gamma}{2\beta} - 1) c$

2. Calculations for each time step

   2.1. $\Delta \ddot{x}_g = \ddot{x}_g(i) - \ddot{x}_g(i-1)$

   2.2. $\Delta \hat{p}_i = -Mr\Delta \ddot{x}_g + a\dot{x}_{i-1} + b\ddot{x}_{i-1}$

   2.3. $\Delta x_i = \frac{\Delta \hat{p}_i}{\widehat{K}}$

   2.4. $\Delta \dot{x}_i = \frac{\gamma}{\beta \Delta t} \Delta x_i - \frac{\gamma}{\beta} \dot{x}_{i-1} + \Delta t (1 - \frac{\gamma}{2\beta}) \ddot{x}_{i-1}$

   2.5. $\Delta \ddot{x}_i = \frac{1}{\beta \Delta t^2} \Delta x_i - \frac{1}{\beta \Delta t} \dot{x}_{i-1} - \frac{1}{2\beta} \ddot{x}_{i-1}$

It is important to remark once more that the calculated $x_i, \dot{x}_i$ and $\ddot{x}_i$ are magnitudes relative to the ground.

4.4.3. Validation of the Newmark method implemented in *AFECTOS*

In order to validate that the Newmark time integration method is programmed correctly, in this section, we will carry out a simple dynamic analysis with a beam that is clamped at the base. The clamped beam will be subjected to a sine wave excitation. The analysis, once again, will be carried out both in *AFECTOS* and in *ANSYS WORKBENCH*.

Figure 4.7 shows the results for the displacement of the end point of the beam. Both programs give almost the exact same response. Therefore we can say that the Newmark time integration method is working correctly. It must be said that for this specific example the damping of the beam was exaggerated in *MATLAB*. In the following section, the Rayleigh damping will be introduced in order to calculate a realistic damping matrix for the simplified model of the satellite.

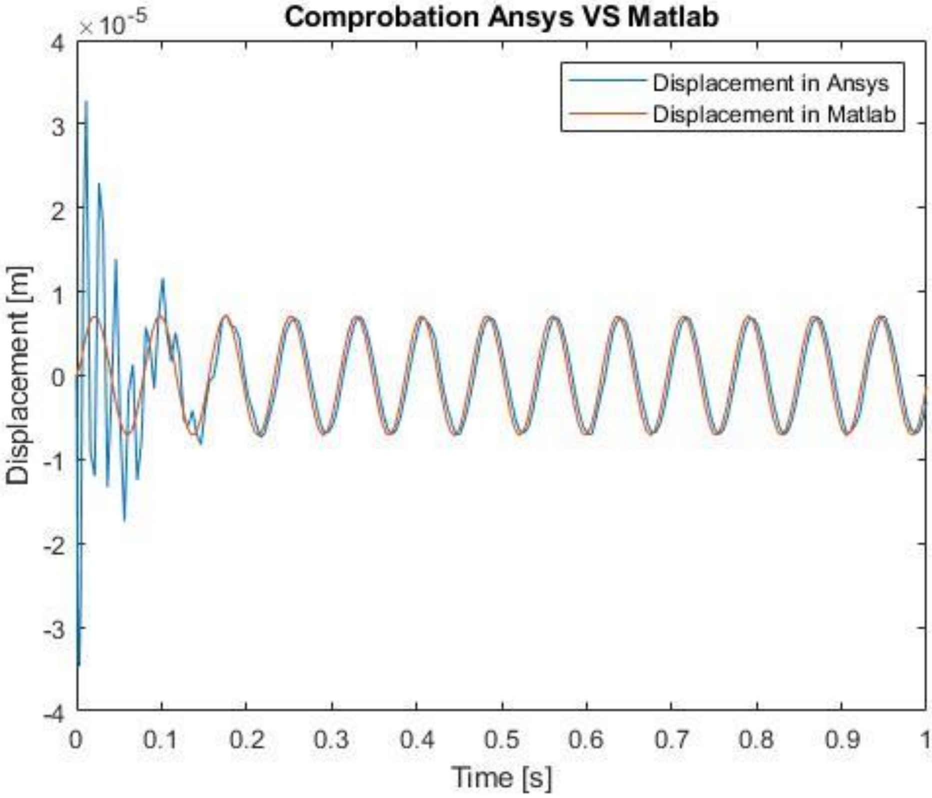

Figure 4.7: Response of a clamped beam to a sine wave excitation at its base

4.5. Rayleigh Damping

The damping matrix *C* of a structure is necessary to solve the equations of motion with step by step integration methods. One of the most common methods to calculate this matrix is with Rayleigh Damping.

Rayleigh Damping [11] considers *C* as a linear combination of the mass matrix *M* and the stiffness matrix *K:*

$$C = a_0 M + a_1 K \tag{8}$$

Where a_0 and a_1 are constants calculated with respect to two known modes of vibration using the following expression:

$$\zeta_i = \frac{a_0}{2\, W_{ni}} + \frac{a_1 W_{ni}}{2} \tag{9}$$

Being ζ_i the damping factor and W_{ni} the natural frequency of mode i.

The selection of which modes will be considered in the Damping Matrix is an important issue. Usually, for frequencies close to the natural frequency of some modes, the stationary response of the structure is approximately the one of those modes. Since the frequency content of a flight transient is really diverse, the Damping Matrix should be changing with each dominant frequency. For simplicity however, in this study we will work with a constant damping matrix calculated with $W_{n1} = 13.4\ Hz/S$ and $W_{n4} = 106\ Hz/S$. These frequencies are close enough to the natural frequencies of most part of the modes calculated in *section 4.2*.

Applying *equation 9* and a damping factor of $\zeta_i = 0.05 \quad (Q = 10)$, $a_0 = 7.475$ and $a_1 = 0.0001$.

Chapter 5
Response of the satellite to a Classical Sine Sweep using Modal Analysis

In *chapter 4*, the response of the finite element model of the satellite to a base excitation has been calculated with the Newmark time integration method. However, there are other ways to solve the equations of motion.

In this chapter, the steady state response of the system will be calculated according to the analytical solution of a Multiple Degree Of Freedom system subjected to harmonic forced vibration. In contrast with the Newmark time integration method, with this approach, only the steady state response of the system for each frequency is known, neglecting the transient effects.

In order to understand how the calculations will be made in the following sections, we will first have to give a brief explanation on some fundamental concepts of Vibration theory.

5.1. Response of a MDOF system to harmonic excitations

According to Frequency Response theory [12], the response of a multiple degree of freedom system subjected to harmonic external excitations can be analytically calculated. The equation of motion will have the following form:

$$M\ddot{q}(t) + C\dot{q}(t) + kq(t) = Q(t) \tag{10}$$

Since the excitation vector $Q(t)$ is harmonic, it can be expressed:

$$Q(t) = Q_0(t)e^{i\alpha t} \tag{11}$$

The steady-state response of the system will also be harmonic and can be expressed:

$$q(t) = q_0(t)e^{i\alpha t} \tag{12}$$

Relating the response of the system $q(t)$ and the excitation vector $Q(t)$ through a matrix of frequency response functions $G(i\alpha)$ we have:

$$q(t) = G(i\alpha)Q_0e^{i\alpha t} \tag{13}$$

The matrix $G(i\alpha)$ is built with the transfer functions that result from the Laplace transformation of the equation of motion.

The main problem with the described approach is that it is only feasible for systems with not too many degrees of freedom. When the degrees of freedom of the system increase, an approach based on the concept of decoupling the equations of motion becomes necessary.

5.2. Modal vectors and the eigenvalue problem

In an MDOF undamped system in which the motion is described by *equation 14*, there are some special motions in which the coordinates of different degrees of freedom change with the same proportion in time. This type of motion is known as synchronous motion. When this happens, the different masses of the system form patterns or displacement configurations that are constant in time. In other words, the relationship between the motions of all the degrees of freedom does not change, only the amplitude does.

$$M\ddot{x}(t) + Kx(t) = 0 \tag{14}$$

The described property can be expressed with *equation 15* in which $f(t)$ represents the amplitude that changes with time and u is a modal vector, or mode shape that represents the displacement pattern.

$$x(t) = f(t)\, u \tag{15}$$

After introducing a special notation and working with these equations, the relation known as the *eigenvalue problem* arises:

$$Ku = \lambda\, Mu \tag{16}$$

This problem can be solved algebraically. The eigenvectors of this problem are the modal vectors or natural modes of the system. The eigenvalues λ are related to the natural frequencies w of the system ($\lambda = w^2$). Moreover, a system with n degrees of freedom will have n natural modes and n natural frequencies. Finally, the matrix U built with the modal vectors will be known as Modal matrix.

5.3. Coupling, natural coordinates and orthogonality of modal vectors

In the undamped multiple degree of freedom described by *equation 14*, the mass matrix is usually diagonal and the stiffness matrix is not. This happens when the chosen coordinates represent the displacements of the masses. If a different coordinate system based on the elongation of the spring elements was chosen, the stiffness matrix would be diagonal and the mass matrix would not. In both situations, at least one of the matrixes is not diagonal and therefore the equations are *coupled*, each equation depends on more than one degree of freedom.

However, *coupling* depends on the selected coordinates to describe the motion of the system and it is not an inherent property of the system. In fact, for every MDOF system, some coordinate system exists, that makes both the mass and stiffness matrix diagonal. These coordinates are known as *natural coordinates* and are closely related to modal vectors.

Modal vectors have the property of being orthogonal with respect to both the mass matrix and stiffness matrix. Consequently, a coordinate transformation based on the modal matrix can diagonalize both matrixes and therefore decouple the equations of motion, transforming them into independent equations. The methodology of solving the equations of motion by means of decoupling them using modal vectors is known as *Modal Analysis.*

5.4. Response of a MDOF system subjected to harmonic excitation using proportional damping and Modal Analysis

In the previous sections, the concept of decoupling the equations of motion has been explained for undamped systems. The problem is that when the damping matrix is taken into account, a coordinate transformation using the Modal matrix does not necessarily make the damping matrix diagonal and the equations of motion remain coupled.

However, there is a special case in which diagonalizing the damping matrix with the modal matrix is possible. This happens when proportional damping has been assumed and the damping matrix has been expressed as a linear combination of the mass and stiffness matrices. In this situation using *Modal Analysis* to determine the response of the system to a harmonic excitation is possible. The mathematical demonstration of the solution to the decoupled equations of motion is outside the scope of this work. In what follows, only the final equations will be presented. Departing from the equation of motion:

$$M\ddot{q}(t) + C\dot{q}(t) + kq(t) = Q(t) \tag{17}$$

Expressing the harmonic excitation with:

$$Q(t) = Q_0(t)e^{i\alpha t} \tag{18}$$

The response of the n degree of freedom system is given by the following equation:

$$q(t) = \sum_{r=1}^{n} \frac{u_r^T Q_0}{w_r^2 - \alpha^2 + i2\zeta_r w_r \alpha} u_r e^{i\alpha t} \tag{19}$$

Where:

- Q_0 is the amplitude and α is the frequency of the harmonic excitation
- w_r are the natural frequencies of the system
- u_r are the modal vectors of the system
- ζ_r are the damping factors, related to the natural frequencies and the proportional damping coefficients with the following expression:

$$\zeta_r = \frac{\alpha + \beta w_r^2}{2w_r} \tag{20}$$

Since the response of the system is expressed in complex notation, we have to take into account that:

- If the excitation is of the form $Q(t) = Q_0(t)\cos(\alpha t)$, we stay with the real part of the response
- If the excitation is of the form $Q(t) = Q_0(t)\sin(\alpha t)$, we stay with the imaginary part of the response

In this work, the described methodology will be applied to calculate the response to a Classical Sine Sweep. Therefore we will stay with the imaginary part of the response.

5.5. Calculating the response of the FEM of the satellite when subjected to a sine excitation of constant amplitude and frequency

The finite element model of the satellite that was developed in *AFECTOS* has 180 degrees of freedom. Therefore, using Modal analysis to solve the equations of motion seemed the most reasonable approach. From the already finished FEM, the mass and stiffness matrices were available. Moreover, decoupling the equations was possible since the assumption of proportional damping was already made and the damping matrix was already calculated. The next step was to calculate the modal vectors and the natural frequencies of the system. Finally, *equation 19* can be used to obtain the response of the system to the harmonic excitation. All of these calculations were done in *MATLAB* (Script can be found in *Annex C*).

An example in which the satellite is subjected to a sine base excitation with amplitude of $10\,m/s^2$ and frequency of $5.5\,Hz$ was studied. *Figure5.1* shows the results of the relative displacement of *mass 4* of the model with respect to the displacement in the base both calculated with the Newmark time integration method and with Modal Analysis. As expected, both methods give the same steady state response.

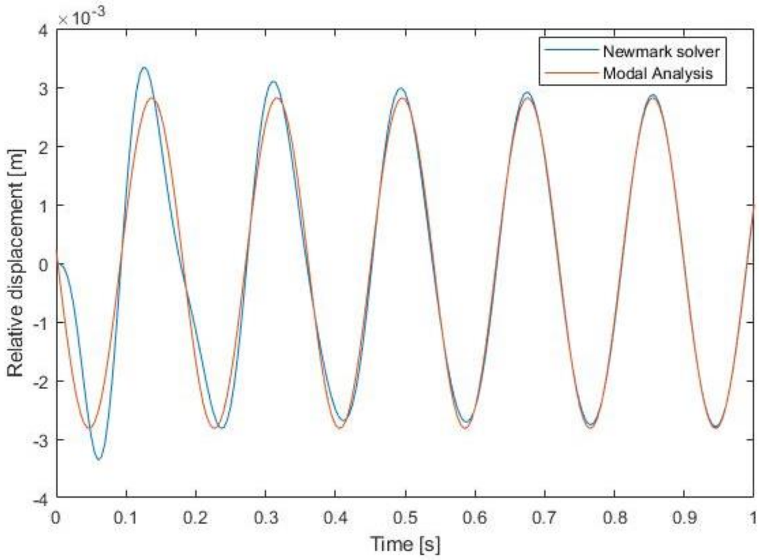

Figure 5.1: Relative displacement of mass 4 of the satellite

5.6. Calculating the response of the FEM of the satellite when subjected to the Classical Sine Sweep obtained from a flight transient

The aim of this section is to calculate the response of the satellite to a Classical Sine Sweep obtained from a real flight transient. The calculations will be made axis by axis with the two approaches described until now:

- Newmark time integration method (*chapter 4).*
- Modal analysis

In order to synthesize the Classical Sine Sweep, the first necessary step is to calculate the Shock Response Spectrum of the flight transient in each direction. *Figures 5.2*, *5.3* and *5.4* show the original flight transient and the obtained SRS.

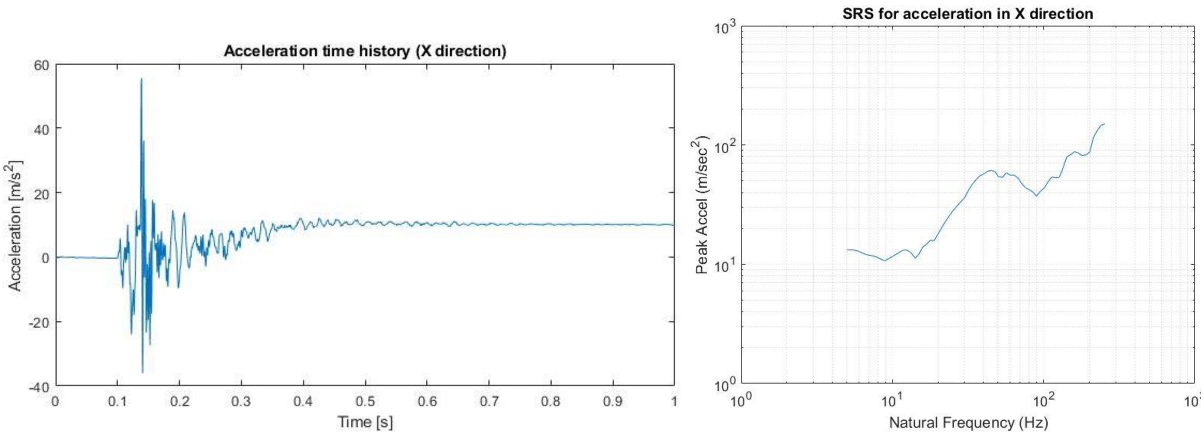

Figure 5.2: Flight transient in X direction

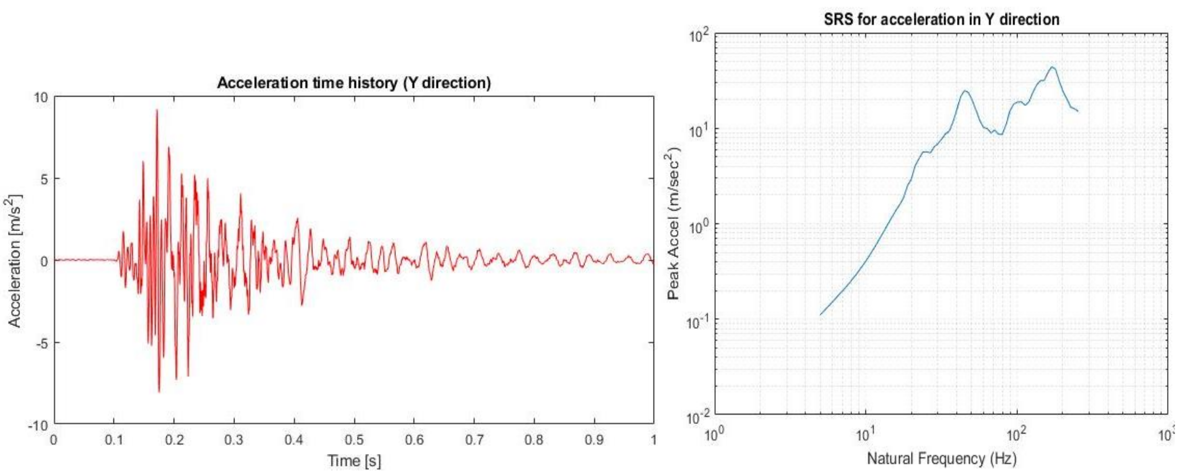

Figure 5.3: Flight transient in Y direction

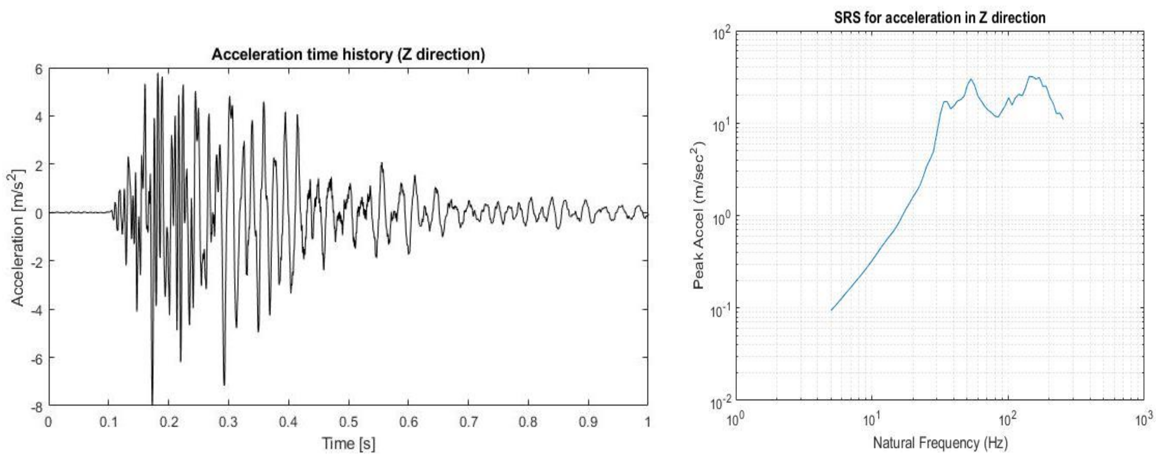

Figure 5.4: Flight transient in Z direction

Once the SRS for each of the spatial directions has been obtained, we can synthesize the Classical Sine Sweep using *equation 2* in *section 3.3* in *chapter 3*:

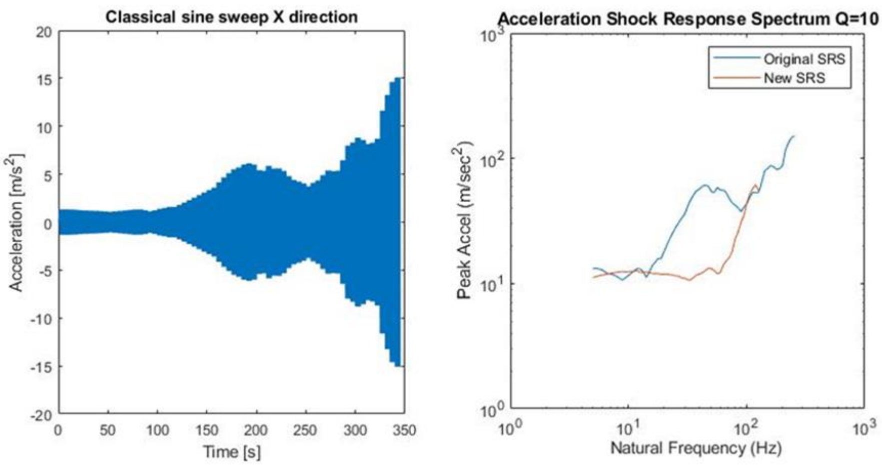

Figure 5.5: Classical Sine Sweep in X direction

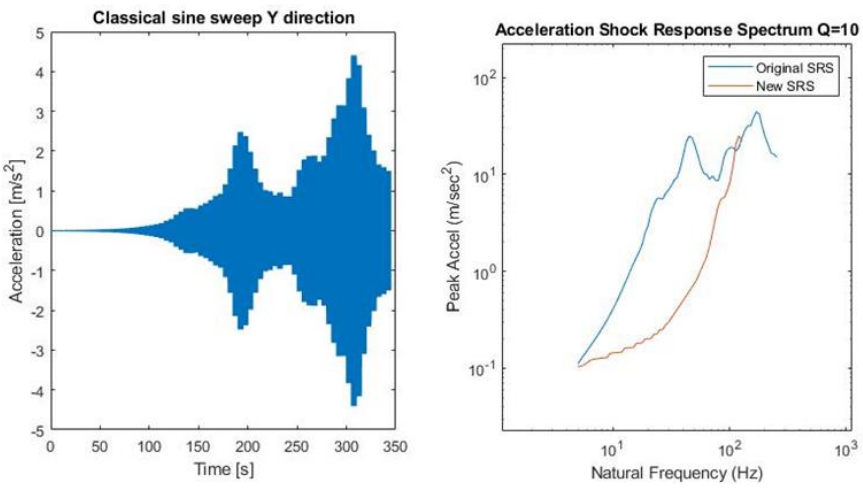

Figure 5.6: Classical Sine Sweep in Y direction

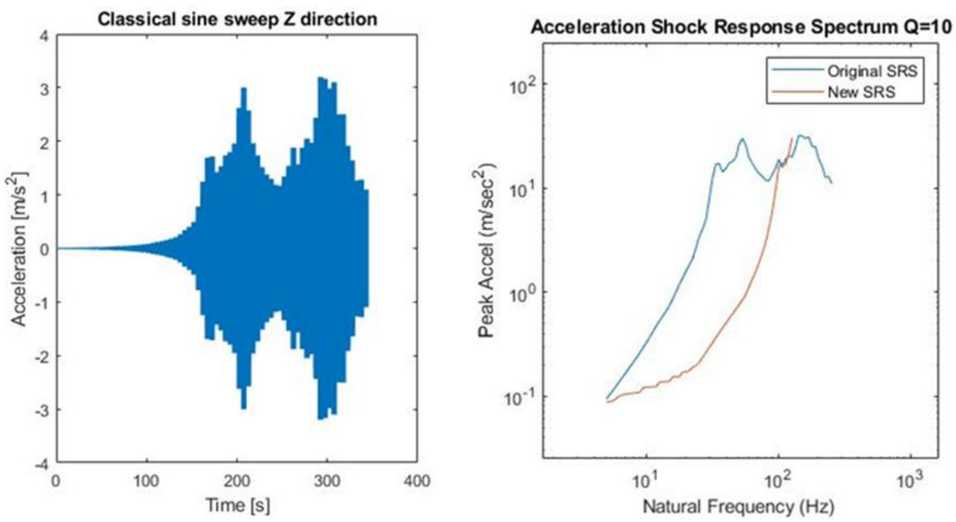

Figure 5.7: Classical Sine Sweep in Z direction

In *chapter 1*, it was shown that a simple signal could be translated into a completely different one in the form of a sine sweep that had the same SRS than the original signal. This worked for the simple example of a half sine pulse. However, apparently, the Classical Sine Sweep method is not that appropriate for short duration complex signals such as the flight transient in question. It can be clearly observed that the SRS of the sine sweeps and the original signals have important differences. Despite these bad results, the study will still consider the classical sine sweep method as an option.

To calculate the response of the system to a sine sweep excitation with modal analysis, the methodology will be exactly the same than in the last section. The only difference is that now the amplitude and the frequency of the sine wave will change with time. Every time they change, the modal analysis calculations will have to be repeated. The table below gives the results for each of the spatial directions.

MAXIMUM ACELERATION OF THE MASSES [m/s^2]			
Direction	Real flight transient	ESI (Modal analysis)	ESI (Newmark)
X	12.33 (mass 2)	9.93 (mass 2)	10.18 (mass 2)
Y	1.62 (mass 3)	1.848 (mass 3)	1.42 (mass 3)
Z	34.6 (mass 3)	37.23 (mass 3)	28.42 (mass 3)

Table 5.1: Response of the satellite to Classical Sine Sweep

5.7. Conclusions

In *chapter 4*, the response of the satellite to a simple sine wave was calculated using Newmark's time integration method. In this chapter, an alternative analytical method based on Modal Analysis has been implemented. The two methods have been applied to find the response of the FEM model of the satellite to the Equivalent Sine Input obtained from a real flight transient.

Both of the methods have been able to roughly predict the maximum accelerations of the satellite. However, the results have shown that the obtained response of the satellite to the Classical Sine Sweep is slightly different depending on which method is used to solve the equations of motion. The main difference between both methods is that Modal Analysis does not take into account transient effects while Newmark time integration method does.

When the Classical Sine Sweep is applied in the X direction, the transient effects lead to higher acceleration values while the opposite is true for the Y and Z directions. Therefore, from the obtained results, it is not possible to determine if transient effects lead to more or less critical conditions, the only thing it can be concluded is that these effects do influence the response of the satellite.

During the Modal Analysis implementation, we have learnt that it is a fast method of calculating the response of a finite element model to a harmonic excitation of constant amplitude and frequency. However, when this tool is applied to obtain the response of a Sine Sweep in which frequency and amplitude change with time, the speed is compromised. Moreover, since it is not able to take into account transient effects which do influence the response of the model, we believe that the Newmark time integration method is a better approach when carrying out vibration simulations. However, despite its disadvantages, carrying out the Modal Analysis has been useful to verify that the Newmark method was correctly programmed.

:

Chapter 6
Influence of the sweep rate

In *chapter 5* the signal used as a Classical Sine Sweep was simplified with the objective of making the calculations easier. Instead of sweeping the frequency with a constant sweep rate, the frequency was constant for a specific time interval and then changed and maintained for the next interval.

The advantage of this approach is that the steady state response will be reached for each of the frequencies for which the Shock Response Spectrum was calculated. The disadvantage is that the satellite will only be tested on those frequencies.

Usually sine sweep tests are carried out with a constant sweep rate that covers the low frequency band. As it was explained in the *chapter 2,* it is advisable to use a slow sweep rate so that the transient effects of constantly changing the frequency are as small as possible. On the other hand, using slow sweep rates make the tests more time consuming.

In this chapter, we will abandon the simplification used in the previous chapter and implement the Classical Sine Sweep according to its strict definition. As a first approach, simple invented sine sweeps will be applied to the model of the satellite in different short frequency bands.

Later on, the dynamic analysis will be carried out with the sine sweeps obtained from the SRS of the real flight transient. The results will be compared with the ones obtained with the simplified approach. This comparison will enable us to draw conclusions on the influence of the sweep rate.

6.1. First approach with simple sine sweeps

In this example we will calculate the response of the simplified model of the satellite to invented sine sweeps with different sweep rates. The sine sweeps will have variable amplitudes depending on the frequency as shown in *figure 6.1*.

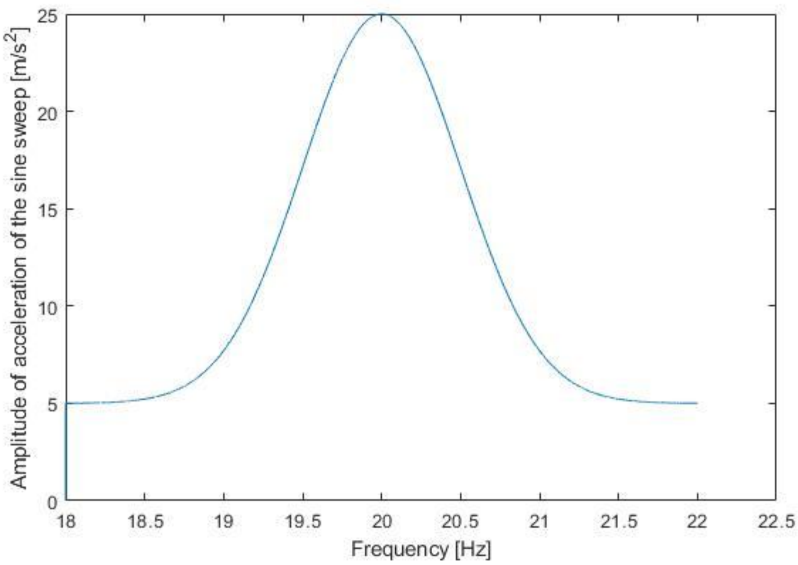

Figure 6.1: Amplitude envelope

From this amplitude envelope, different sine sweeps can be obtained depending on the desired sweep rate using *equation 21*. In this case we will study sweep rates of $R = 1\ (oct/min)$ and $R = -1(oct/min)$.

$$f(t) = f_0 2^{\frac{Rt}{60}} \tag{21}$$

Once the acceleration time histories are calculated for the different sweep rates, we can carry out the dynamic analysis. Since we want to take into account the transient effects, the dynamic analysis will be carried out with the Newmark solver. *Figure 6.2* shows an example of the sine sweep for $R = 1\ (oct/min)$.

To find out the influence of the different sweep rates, we will compare the obtained responses with the response of the satellite to the case in which $R = 0\ (oct/min)$. Of course, a sine sweep with a sweep rate of zero is not possible since the frequency remains constant. However, we can obtain the ideal steady state response for each of the frequencies studied in the $R = 1\ (oct/min)$ and $R = -1(oct/min)$ cases and treat it as the $R = 0\ (oct/min)$ "sine sweep". This fictitious case will be calculated with the Modal Analysis methodology explained in *chapter 5*.

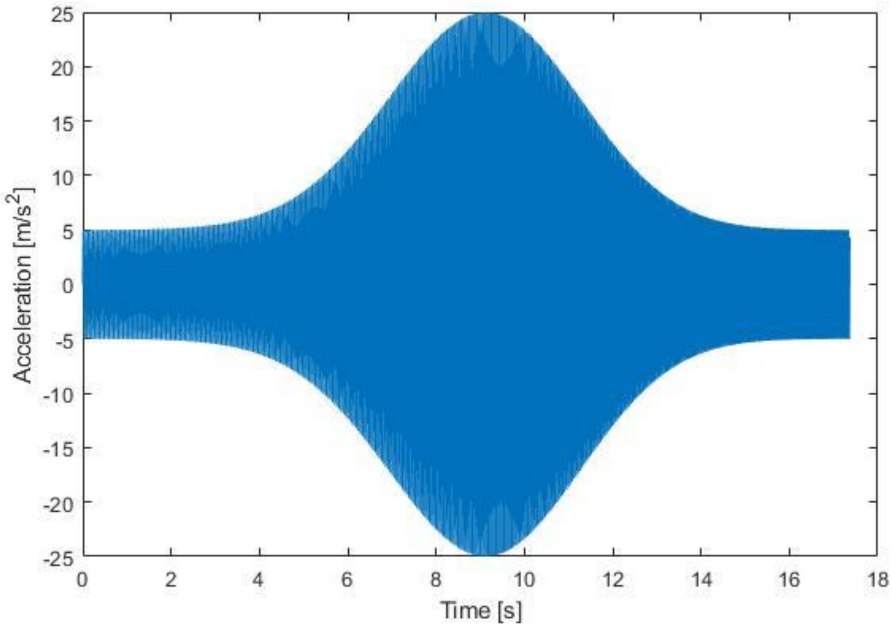

Figure 6.2: Sine sweep with R =1 (oct/min)

6.1.1. First experiment

The first case that was studied was the response of mass number 2 of the model with a sine sweep with frequencies varying from 18 Hz to 22 Hz. The results are shown in *figure 6.3*. It can be seen that the positive sweep rate leads to a decrease in the amplitude of the response. A negative sweep rate surprisingly has the opposite effect.

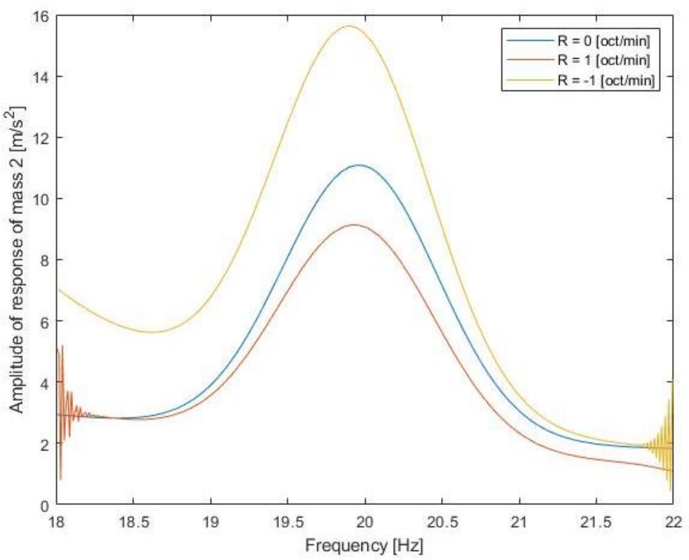

Figure 6.3: Influence of sweep rate in mass 2 with frequencies between 18 and 22 Hz

However, this effect is not the same for all of the excited frequencies. The response of mass 2 was also studied for sine sweeps with higher frequencies. We can observe in *figure 6.4* that when the frequencies vary from 38 Hz to 42 Hz, the positive sweep rate now leads to higher amplitudes in the response. When the frequencies are even higher (*figure 6.5*), the results are the same than for the lower frequencies

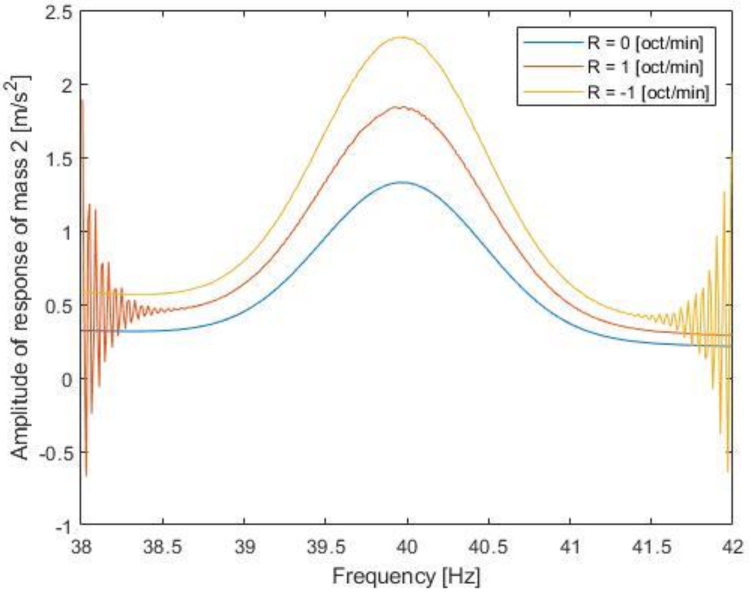

Figure 6.4: Influence of sweep rate in mass 2 with frequencies between 38 and 42 Hz

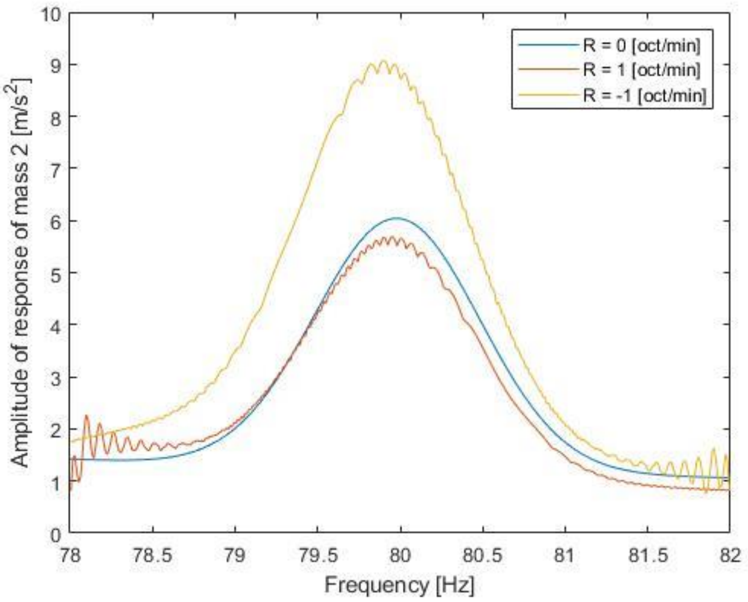

Figure 6.5: Influence of sweep rate in mass 2 with frequencies between 78 and 82 Hz

The response of mass number 3 of the satellite was also studied. The results are the same than for mass 2:

- Between 18 Hz and 22 Hz a positive sweep rate decreases the amplitude of the response and a negative one increases it.
- Between 38 Hz and 42 Hz a positive sweep rate increases the amplitude of the response and a negative also increases it.
- Between 78 Hz and 82 Hz a positive sweep rate decreases the amplitude of the response and a negative one increases it.

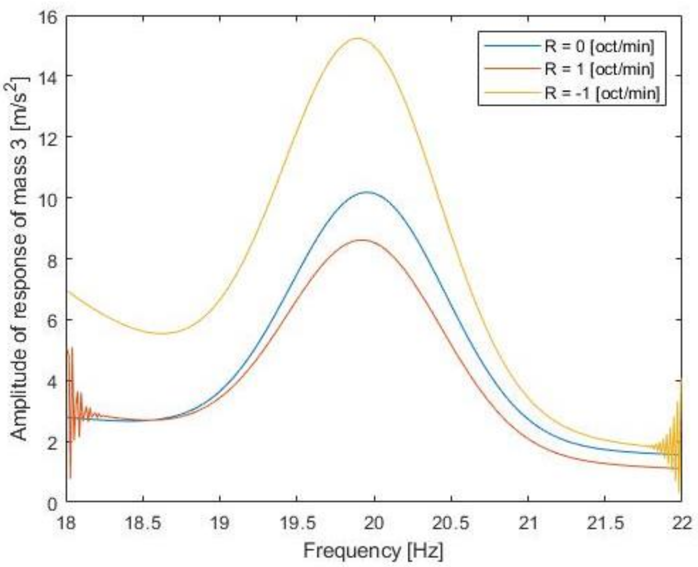

Figure 6.6: Influence of sweep rate in mass 3 with frequencies between 18 and 22 Hz

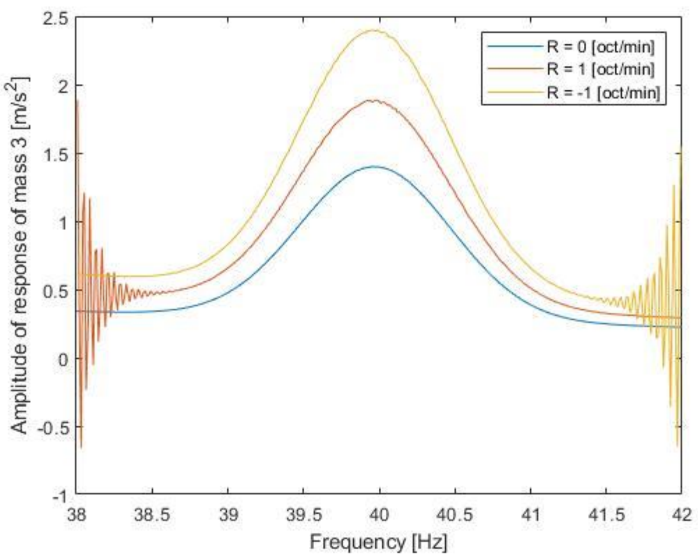

Figure 6.7: Influence of sweep rate in mass 3 with frequencies between 38 and 42 Hz

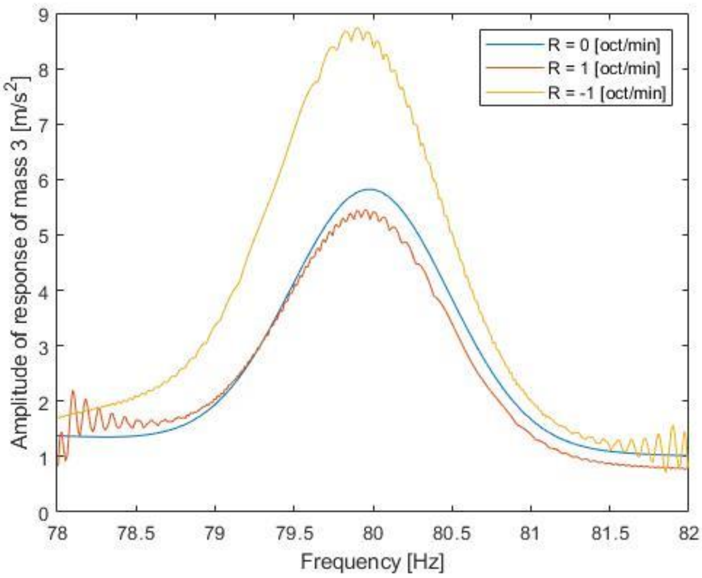

Figure 6.8: Influence of sweep rate in mass 3 with frequencies between 78 and 82 Hz

6.1.2. Repetition of the experiment for all the low frequency band

From the obtained results, it is difficult to predict how the sweep rate will affect the response of the satellite during a real sine sweep test. For some frequencies the sweep rate will lead to higher amplitudes. For other frequencies, the sweep rate will lead to lower amplitudes. In order to determine which frequencies have which effect, the same experiment will be carried out covering the whole low frequency band.

The problem with carrying out this experiment is the high computation time. The sine sweep has to cover all the frequencies from 5 Hz to 100 Hz at a low sweep rate. With a sweep rate of $R = 1\ (oct/min)$ the sine sweep lasts for around 260 seconds. At the same time, the time step has to be small enough so that the sine wave with the highest frequency is appropriately represented. An appropriate time step will be around 0.0001 seconds. Therefore, the dynamic analysis has to solve the equations of motion for 2600000 points, which takes considerable time. For this reason, the experiment was only done for a sweep rate of $R = 1\ (oct/min)$, which is commonly used in sine sweep tests. *Figure 6.9* shows the obtained results.

From these results we can define three frequency intervals in which the effect of a positive sweep rate is different:

- From 10 to 25 Hz, a positive sweep rate leads to lower accelerations

- From 25 to 70 Hz, a positive sweep rate leads to higher accelerations
- For frequencies higher than 70 Hz, the sweep rate has little effect

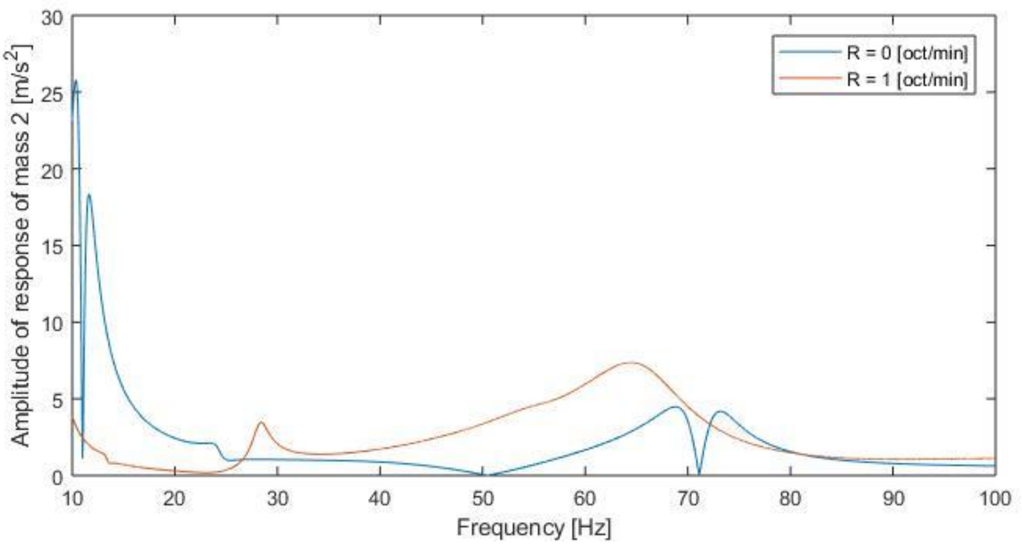

Figure 6.9: Influence of sweep rate along the entire low frequency band

6.2. Influence of the sweep rate in the ESI obtained from a real flight transient

In the previous sections, the influence of the sweep rate has been studied with simple fictitious sine sweeps. In this section, we will study this influence in the real sine sweep simulation which applies the sine sweep obtained from the flight transient. We will determine if the conclusions extracted in the previous section are still true in the real simulation. Given the high computation time of these simulations, we will only study the sine sweep in the X direction with a sweep rate of $R = 1\ (oct/min)$.

MAXIMUM ACELERATION OF THE MASSES [m/s^2]		
Direction	R=1 (oct/min)	R=0 (oct/min)
X	10.45	10.17

Table 6. 1: Results for sine sweeps with different sweep rates

The results show that a positive sweep rate leads to a higher maximum acceleration in the X direction. The reaction force at the base is also higher. According to the results obtained in last section, this happened when the frequency of the sine sweep was between 25 and 70 Hz. If we check the original Shock Response Spectrum in the X direction, shown in *figure 6.10*, the maximum input acceleration has a frequency of 45 Hz, which is in the predicted interval.

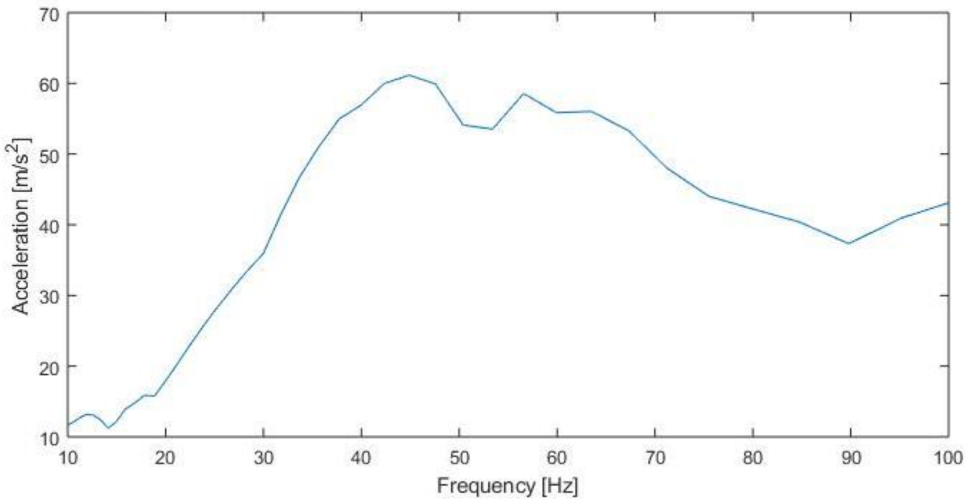

Figure 6.10: SRS of flight transient in X direction

6.3. Conclusions

The objective of this chapter was to find out the influence of the sweep rate on the Classical Sine Sweep. This has been done in three steps:

- Study the response of the satellite to simple sine sweeps with different sweep rates in small frequency intervals (18-22 Hz, 38-42 Hz and 78-82 Hz)
- Study the response of the satellite to a simple sweep in a big frequency interval (10-100 Hz)
- Simulation of the sine sweep obtained from the flight transient to verify the conclusions extracted from the previous steps

The first important conclusion is that the sweep rate does affect the amplitude of the response of the system, even with a really slow sweep rate as $R = 1\ (oct/min)$. Moreover, the results have shown that in a multiple degree of freedom system, the influence of a positive sweep rate depends on the frequency interval. In our specific model, a positive sweep rate led to higher accelerations between 25 to 70 Hz, lower accelerations between 10 and 25 Hz and similar accelerations between 70 and 100 Hz.

The second conclusion is that it is possible to make a rough prediction of the Classical Sine Sweep results if we have the Modal Analysis results (R = 0 oct/min, or steady state results). If the Shock response Spectrum of the flight transient shows that the maximum acceleration amplitudes apply for frequencies between 25 and 70 Hz then the acceleration results of the Classical Sine Sweep will be higher than those obtained with the Modal Analysis. Moreover, we could also say that in this case, the satellite will be over-tested as a consequence of the sweep rate. If the maximum accelerations in the SRS happen for frequencies below the specified frequency interval, we can predict that the satellite will be under-tested as a consequence of the sweep rate. Finally if maximum accelerations of the SRS happen for higher frequencies, the results of the Classical Sine Sweep and the Modal Analysis should be similar.

Chapter 7
Comparison of alternative synthesis methods by FEA

The purpose of this chapter is to carry out Finite Element Analysis on the simplified model of a satellite to compare the alternative synthesis methods discussed in *chapter 3* (Fast Sine Sweep and Wavelets). This will allow us to determine whether establishing equivalence between a flight transient and a synthesized signal is valid or not. Furthermore, if a synthesis method is able to reproduce the most critical loads of flight transients with a positive margin, this will mean that the method could be used to define loads specifications.

The methodology for each of the studied synthesis approaches will be:

- Given real transient flight data, calculate its Shock Response Spectrum. This step was already done in *chapter 5* (*figures 5.2*, 5.3 and *5.4*)
- Apply the specific synthesis method described in *chapter 3* to obtain an acceleration time history that gives a similar Shock Response Spectrum
- Perform a simulation with a simplified model of a satellite in which the flight transient acceleration is introduced as a base excitation
- Perform the same simulation with the synthesized signal
- Compare the results of both simulations

7.1. Synthesis of signals from the original flight transient

7.1.1. Synthesis of the FSS

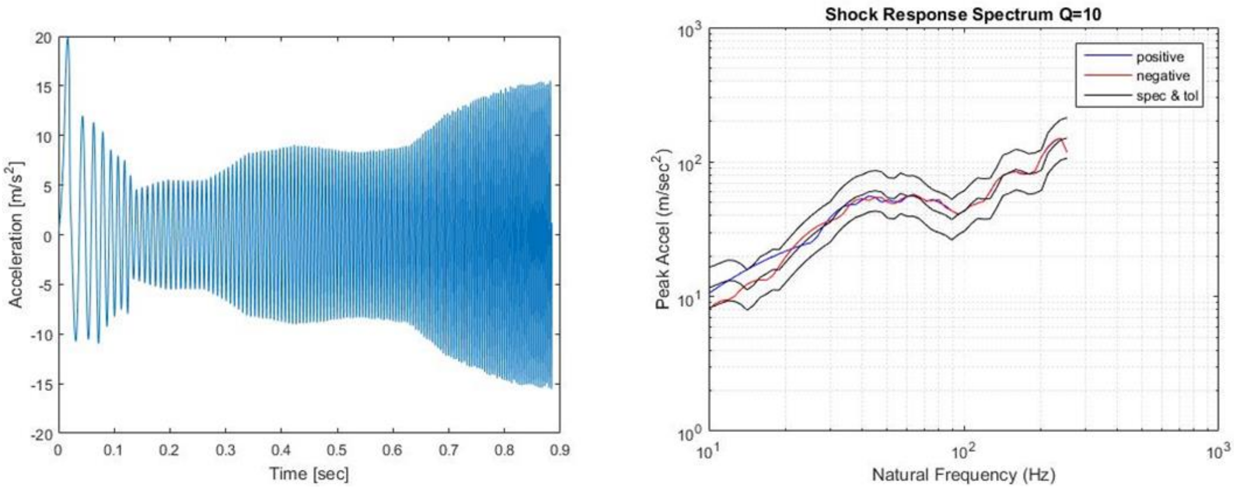

Figure 7.1: FSS obtained from flight transient in X direction

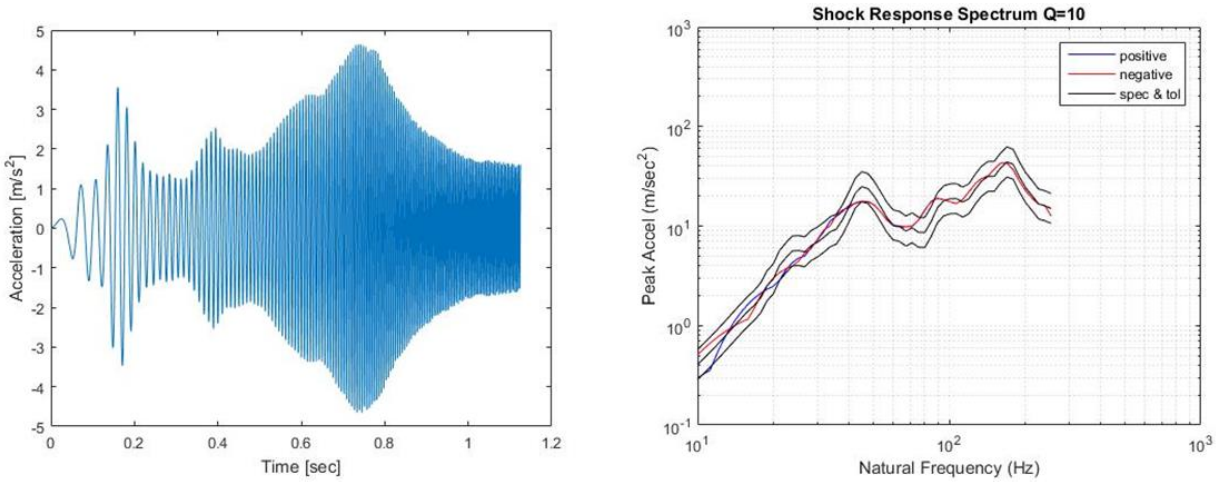

Figure 7.2: FSS obtained from flight transient in Y direction

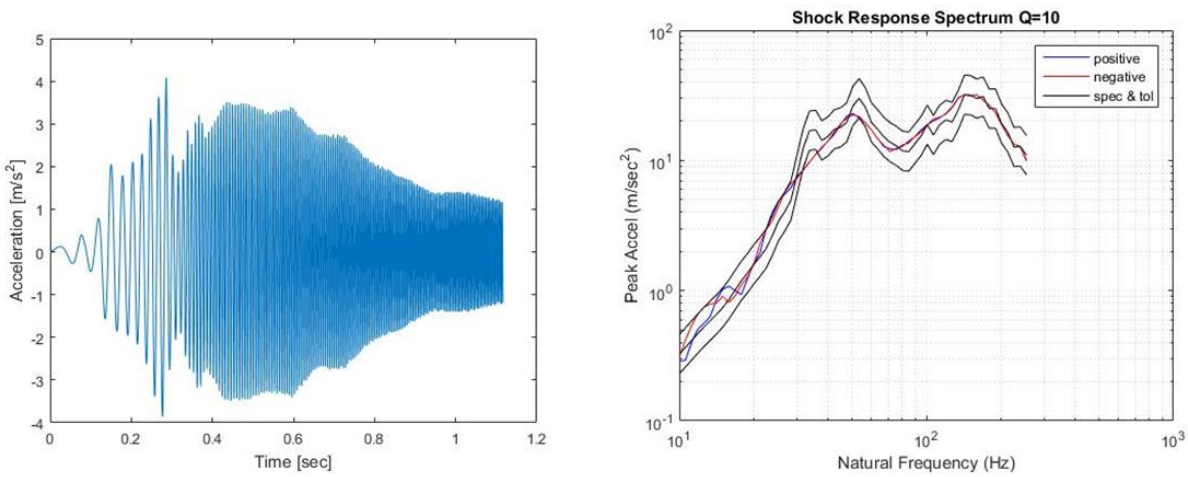

Figure 7.3: FSS obtained from flight transient in Z direction

7.1.2. Synthesis of signals using Wavelets

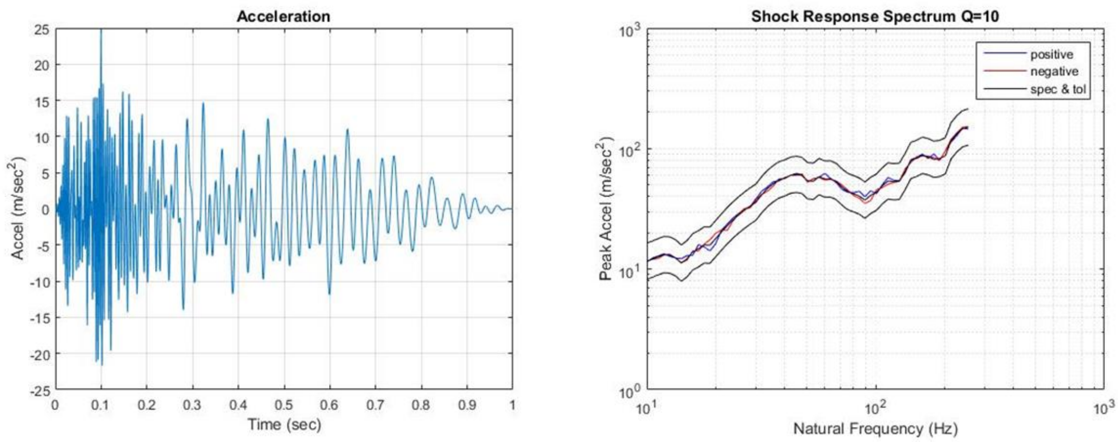

Figure 7.4: Synthetic signal using wavelets obtained from flight transient in X direction

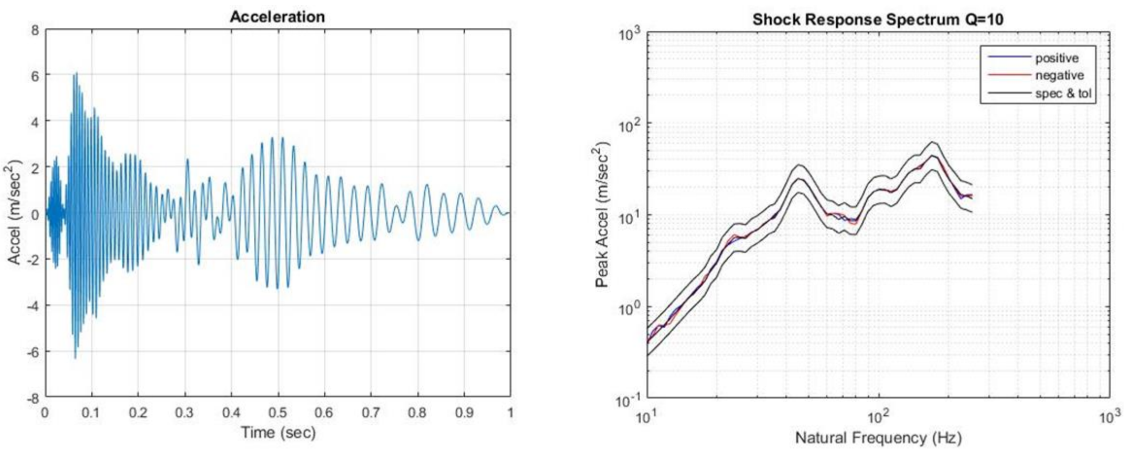

Figure 7.5: Synthetic signal using wavelets obtained from flight transient in Y direction

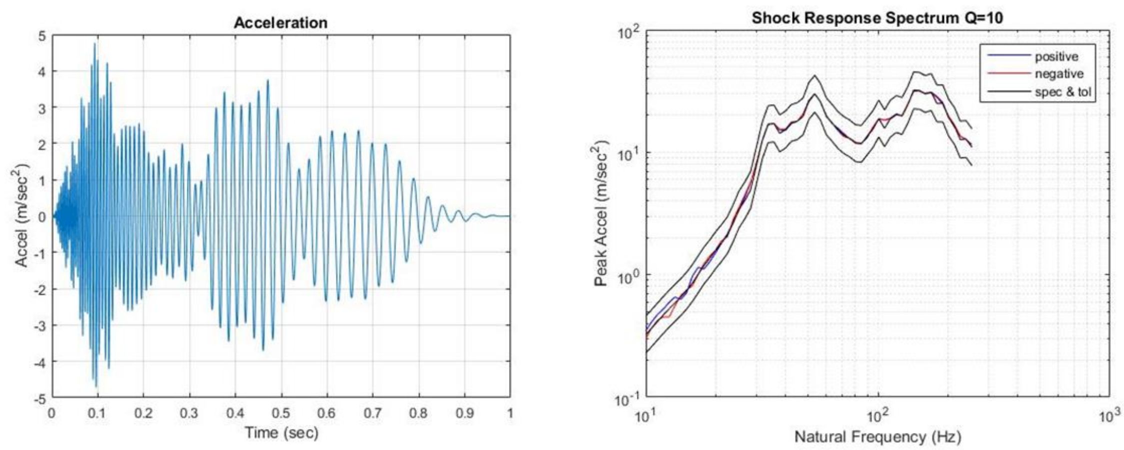

Figure 7.6: Synthetic signal using wavelets obtained from flight transient in Z direction

7.2. Dynamic analysis with the synthetic signals

This section will study the response of the simplified model of the satellite when subjected to acceleration time histories in its base. This analysis is a simulation of how the satellite will respond during flight.

Four situations will be analyzed:

- Application of the acceleration time history of the original flight transient. This case will serve as the base for assessing the validity of each of the tests.
- Test 1: Application of the synthesized acceleration time history obtained with the Fast Sine Sweep method
- Test 2: Application of the synthesized acceleration time history obtained with the Wavelets method.

The variables to study will be:

- Force at the base of the satellite
- Acceleration at the end points of the satellite
- Number of cycles at the base and the end points of the satellite

As it was explained in *chapter 2*, tests are carried out axis by axis in electrodynamic shakers, but in a real flight, all of the loads are simultaneous. In this study, the first simulations will be done applying the base excitation axis by axis in only one of the spatial directions and later on, the excitation will be introduced in all of the directions simultaneously. By comparing the results we will be able to determine if these two situations are or not equivalent. The results shown in this chapter are a summary of all the available results of the dynamic analysis. Additional more detailed results are available in *Annex C*.

7.2.1. Results when the base excitation is introduced axis by axis

7.2.1.1. Results for X direction

MAXIMUM ACCELERATION OF THE MASSES [m/s^2]				
Direction	Real flight transient	ESI	FSS	Wavelets
X	12.33 (mass 2)	10.18 (mass 2)	12.4 (mass 2)	13.04 (mass 2)

MAXIMUM FORCES AT THE BASE [N]				
	Real flight transient	ESI	FSS	Wavelets
F max (X)	16536	21038	24167	26960
F min (X)	-32210	-21039	-27745	-26869

MAX NUMBER OF CYCLES			
Real flight transient	ESI	FSS	Wavelets
259 (mass 3, Y)	22110 (mass 2, Z)	144 (mass 2, Z)	52 (mass 2, Z)

Table 7. 1: Results of the dynamic analysis when excitations are applied in X direction

In first place, all of the approaches agree that the maximum accelerations will occur in mass number 2, which is what the dynamic analysis predicts for the flight transient. Moreover, the FSS and the Wavelets are able to reproduce similar but higher accelerations, which means that they could be used for verification of the accelerations in the X direction. In this case, FSS will be the best option to avoid over-testing.

Secondly, analyzing the number of cycles during simulation is interesting from the point of view of material fatigue. Material fatigue studies the failure of structures subjected to cyclic loads. When a large number of cycles occur, microscopic cracks start to appear in the material. Later those cracks start growing with each cycle until the structure fails.

Taking this into account, the ideal will be to design a test with a low number of cycles but that still verifies that the structure is strong enough. As it can be seen in *table 7.1*, this is one of the biggest disadvantages of the Classical Sine Sweep. Due to the high number of cycles, a test in which the Classical Sine Sweep is used for verification has the risk of damaging the structure through mechanical fatigue. On the other hand, the wavelets and the FSS approach generate a really low number of cycles. For this reason, these approaches will be the most appropriate to minimize the damage during test.

7.2.1.2. Results for Y direction

The results obtained in the Y direction show that, although they are close, none of the synthetic signals could be used in a qualification test. The accelerations that these signals generate are smaller than those that would appear during a real flight. However, once more, all of the approaches predict the same mass subjected to the maximum acceleration, in this case, mass 3.

Regarding mechanical fatigue, wavelets seem to be the best approach again.

MAXIMUM ACCELERATION OF THE MASSES [m/s^2]				
Direction	Real flight transient	ESI	FSS	Wavelets
Y	1.62 (mass 3)	1.42 (mass 3)	1.21 (mass 3)	1.42 (mass 3)

MAXIMUM FORCES AT THE BASE [N]				
	Real flight transient	ESI	FSS	Wavelets
F max (Y)	2115	2260	3260	1965
F min (Y)	-2470	-2914	-3377	-2281

MAX NUMBER OF CYCLES			
Real flight transient	ESI	FSS	Wavelets
210(mass 3, Z)	22123 (mass 3, Z)	161 (mass 3, Z)	102 (mass 3, Z)

Table 7. 2: Results of the dynamic analysis when excitations are applied in Y direction

7.2.1.3. Results for Z direction

Now, regarding maximum accelerations, the best approach would be the Wavelets. In fact, it is the only approach that reproduces higher loads than the flight transient and therefore the only one that could be used for a qualification test.

Moreover, the Wavelets are also the only method that reproduces similar loads at the base than the flight transient.

MAXIMUM ACCELERATION OF THE MASSES [m/s^2]				
Direction	Real flight transient	ESI	FSS	Wavelets
Y	34.6 (mass 3)	28.42 (mass 3)	31.23 (mass 3)	35.36 (mass 3)

MAXIMUM FORCES AT THE BASE [N]				
	Real flight transient	ESI	FSS	Wavelets
F max (Y)	42918 (Y)	43446 (Z)	37699 (Z)	45397 (Y)
F min (Y)	-47951 (Y)	-43446 (Z)	-37675 (Z)	-43365 (Y)

MAX NUMBER OF CYCLES			
Real flight transient	ESI	FSS	Wavelets
191 (mass 2, Z)	21920 (mass 3, Z)	163 (mass 3, Z)	73 (mass 3, Z)

Table 7. 3: Results of the dynamic analysis when excitations are applied in Z direction

7.2.1.4. Combination of the axis by axis results

In order to have a more general perspective of the results, the following tables combine the maximum values of all the single-axis tests. These tables will be used to compare the axis by axis simulations with the multi-axis simulations.

MAXIMUM ACCELERATION OF THE MASSES [m/s^2]				
	Real flight transient	ESI	FSS	Wavelets
X	12.33 (mass 2)	10.18 (mass 2)	12.4 (mass 2)	13.04 (mass 2)
Y	1.62 (mass 3)	1.427 (mass 3)	1.21 (mass 3)	1.42 (mass 3)
Z	34.6 (mass 3)	28.42 (mass 3)	39.18 (mass 3)	35.36 (mass 3)

MAXIMUM FORCES AT THE BASE [N]				
	Real flight transient	ESI	FSS	Wavelets
F max (Z)	42918	43446	37699	45397
F min (Z)	-47951	-43446	-37675	-43365

Table 7. 4: Results of the dynamic analysis when combining single axis results

From the above results we could say that the FSS and Wavelets are better approaches than the Classical Sine Sweep. However, both of the approaches fail to reach the real maximum acceleration in Y direction. Fortunately, the loads in the Y direction are really low in comparison to the loads in X and Z directions and the difference with the flight transient is not that important.

7.2.2. Results when the base excitation is in multiple axis simultaneously

When the base excitation is introduced in all of the directions simultaneously, the following results are obtained:

MAXIMUM ACCELERATION OF THE MASSES [m/s^2]				
	Real flight transient	ESI	FSS	Wavelets
X	12.45 (2)	10.13 (2)	12.5 (2)	12.93 (2)
Y	1.503 (3)	1.346 (3)	2.657 (3)	1.398 (3)
Z	42.45 (3)	40.39 (1)	44.95 (3)	41.73 (3)

MAXIMUM FORCES AT THE BASE [N]				
	Real flight transient	ESI	FSS	Wavelets
F max	41463(Z)	37566 (Z)	50586 (Z)	43711(Z)
F min	-47687 (Z)	-37581 (Z)	-49257 (Z)	-45867 (Z)

Table 7. 5: Results of the dynamic analysis when excitation is introduced in multiple axis

The most important difference with respect to the axis by axis results appears in the Z direction. When applied simultaneously, the base excitations lead to higher accelerations and loads in this direction than when applied separately. As it was expected, this means that axis by axis simulations and tests will not cover the most critical conditions of a flight transient.

Regarding the selection of the specification method, the FSS seems to be the best approach since it is the only that reproduces the most critical conditions found during the flight transient. However, it is important to remark that these results were obtained with only one flight transient. The results for a different flight transient could differ.

7.3. Conclusions

The purpose of this chapter was to compare the different synthesis methods discussed in *chapter 3* by performing finite element analysis on a simplified model of a satellite. The comparisons have been made in two different situations:

- Axis by axis: The excitations at the base of the satellite are introduced one axis at a time.
- Multiple-axis: The excitations are introduced in all the spatial directions simultaneously.

The first conclusion that should be remarked is that the classical sine sweep is not as good as the wavelet and the fast sine sweep methods when synthesizing signals that match a desired Shock Response Spectrum.

Secondly, we have observed that the axis by axis and the multiple-axis approaches lead to different results. In fact, in the specific transient that has been studied, the multiple-axis situation is more critical than the axis by axis one. Therefore, it would be more reasonable to carry out multiple-axis vibration simulations and tests instead of axis by axis ones when designing satellites.

Finally, regarding the usefulness of the methods to define the low frequency vibration specification for a satellite:

- The classical sine sweep method has been found insufficient in both axis by axis and multiple-axis situations.
- The wavelet method is only sufficient in axis by axis conditions.
- The Fast Sine Sweep method is the best option in both types of conditions.

Chapter 8

A new proposal based on Wavelet Analysis

The previous methods studied in this project have one common limitation, they are based on frequency spectra which contains no information of the time at which each frequency appears. Therefore, as we have seen the synthetic signals generated with these methods are not similar to real launch acceleration time histories.

To generate signals that resemble more real conditions, an approach that is able to capture the frequency content along the time dimension is required. This chapter will study the possibility of doing this through what is called wavelet analysis, an approach used in the field of seismic analysis. However, before diving into the implementation of this approach, some general theory on the field will be given.

8.1. Stationary and non-stationary models

The different models to generate synthetic acceleration time histories can be classified into stationary and non-stationary. In summary, stationary models are those in which the statistical properties, such as the mean and the variance, remain constant whereas on non-stationary models those properties change over time.

The amplitude variation and the frequency content of acceleration time histories created with stationary models will be quite uniform along time. For this reason, these models are not the best option to represent accelerations produced during launch in which both frequency content and the amplitude are variable. Even though the signals created with them can match the shock response spectrum of original transients, the damage they provide to the structure of a satellite might be considerably different to the one that a real launch will generate since the accumulation of damage due to fatigue relies on the time history of loads. An example of a stationary model is applying the Fourier transform to express an acceleration time history as a sum of harmonic signals of different frequencies.

Non-stationary processes are able to generate acceleration time histories that are more similar to a real launch. These processes are based on what is called as a modulating function which are the elements in charge of providing the non-uniformity of amplitude variation and frequency content that characterizes non-stationary processes. If the modulating functions are independent of the frequency, the process is uniformly modulated. If they depend on time and frequency the process is non-uniformly modulated.

A uniformly modulated process can be expressed in the form of:

$$a_s(t) = m(t)\int_{-\infty}^{+\infty} e^{iwt} dZ(w) = m(t)\, s(t) \qquad (22)$$

where $m(t)$ is the modulating function independent of the frequency and $s(t)$ an aleatory and stationary Gaussian process.

A non-uniformly modulated process can be expressed as a superposition of uniformly modulated processes:

$$a_s(t) = \sum_k m_k(t)\, s_k(t) \qquad (23)$$

where k represents a specific frequency band of the process and the modulating functions are normalized so that:

$$\int_{-\infty}^{+\infty} {m_k}^2(t) dt = 1 \qquad (24)$$

Although these concepts have not been introduced before in the project, they have been already applied. The classical sine sweep, the fast sine sweep and the wavelet method developed earlier in the project, are examples of uniformly modulated processes. In all of them, the modulating function is the time dependent amplitude which guaranteed that the shock response spectrums match. However, to capture the variation with time of frequency content during launch, a non-uniformly modulated process is required.

8.2. Introduction to wavelet analysis

According to the concepts explained in the previous section, in order to obtain a realistic synthetic acceleration time history of a launch through a non-stationary and non-uniformly modulated model, the modulating functions for different frequency bands must be obtained. However, the observation of what is happening in different frequency bands is restricted by the uncertainty principle which poses that in time-frequency analysis it is not possible to obtain good resolution in both domains simultaneously.

As an example, a valid method to identify the variation of spectral characteristics with time is to apply the time dependent Fourier transform defined as:

$$F(w,T) = \int_{-\infty}^{+\infty} w(t-T)a_s(t)e^{iwt}dt \tag{25}$$

where $w(t)$ is the band where the signal is filtered and its width depends on the stationarity assumptions in the signal. Since in the acceleration time history of a launch the stationarity is really low, the width of the analysis band $w(t)$ in the time domain is s to be really little and due to the uncertainty principle, the resolution in the frequency domain will be bad and the identification of the modulating functions really poor.

As it will be explained in the following sections, wavelet analysis is can manage these restrictions imposed by the uncertainty principle and constitute a suitable tool to calculate the modulating functions.

8.3. Wavelet transform

Until the invention of wavelets by Morlet in 1982, it was thought that it was impossible to have functions with good localization properties in both the time and the frequency domain [13]. Through what is known as the wavelet transform, signals can be decomposed simultaneously in the time and the frequency domain and provide much more information about non-stationary signals than the Fourier transform.

Both the wavelet and the Fourier transform work by comparing the signal we want to decompose with other more simple signals with known characteristics. In the case of the Fourier transform, the original signal is compared with sinusoidal functions of infinite duration and constant frequency and amplitude expressed in the form of e^{iwt}. As it can be seen in the *equation 24*, the comparison is made by integrating the product of the original signal and the comparing function. When both signals have similar characteristics, the integral will have a higher value. The result is a number of coefficients $F(w)$ that contain important information of the frequency content of the original signal. Moreover, once the coefficients are available, the original signal can be recovered by performing the inverse Fourier Transform.

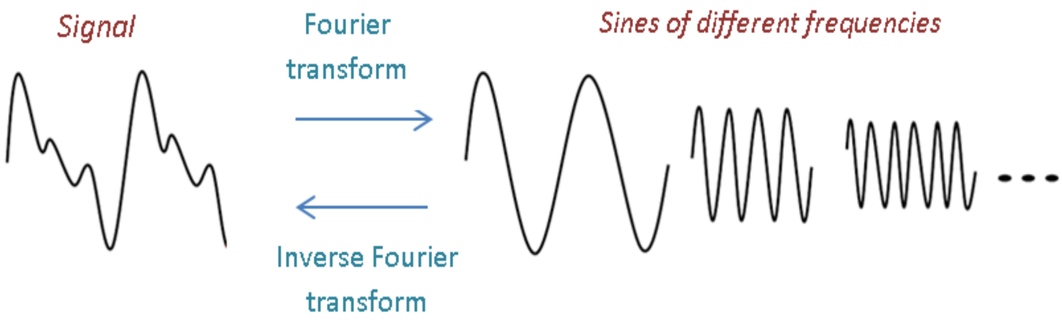

Figure 8. 1: Fourier Transform

In the case of the Wavelet transform, the signal is decomposed in located waves called wavelets. These wavelets have a mean value of zero and have a finite number of oscillations. Although there are different wavelets that have been proposed, the wavelet defined by Morlet has the following expression:

$$\Psi(t) = e^{i\Omega t} e^{-\frac{1}{2}\left(\frac{t}{\sigma}\right)^2} \tag{26}$$

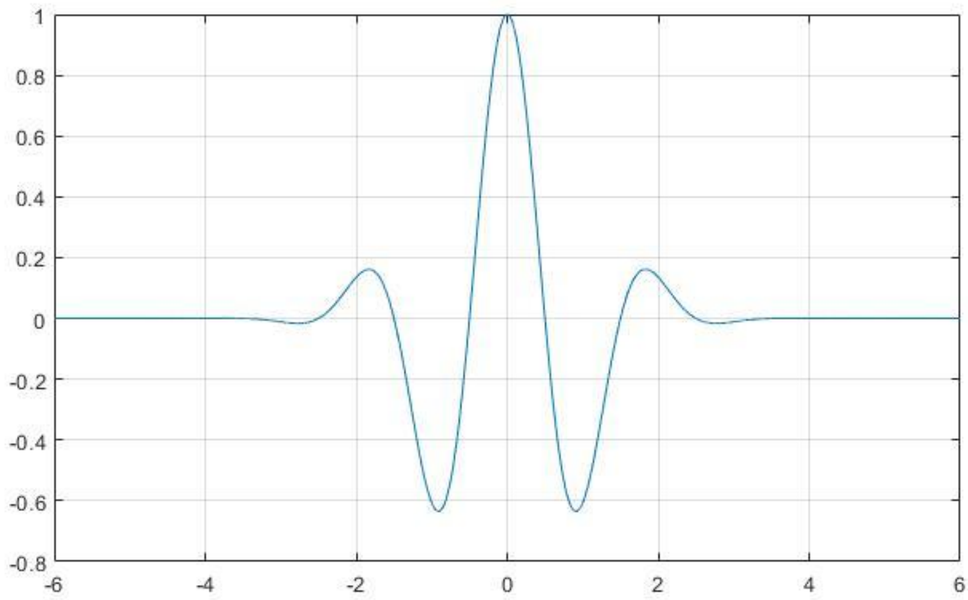

Figure 8. 2: Morlet's "mother wavelet" with $\boldsymbol{\Omega} = \boldsymbol{\pi}$ *and* $\boldsymbol{\sigma} = \mathbf{1}$

Figure 8.2 shows the function $\Psi(t)$ with $\Omega = \pi$ and $\sigma = 1$. In theory, time will extend from $-\infty$ to $+\infty$. However the wavelet is centered in $t = 0$ and only has a significant value between two time instants dependent of Ω and σ. To center the wavelet in a different time instant "T" and to scale it with a factor of "a", the wavelet function has to be evaluated in $t' = \frac{t-T}{a}$ and multiplied by a factor of $\frac{1}{\sqrt{a}}$:

$$\Psi_{a,T} = \frac{1}{\sqrt{a}}\Psi\left(\frac{t-T}{a}\right) \tag{27}$$

Moreover, the factor a is also the parameter that defines the frequency of the wavelet. It can be demonstrated that a is inversely proportional to a frequency that depends of the type of wavelet that is used. In the case of Morlet's wavelet the relationship between the scale a and the frequency ω is:

$$\omega = \frac{\Omega}{a} \tag{28}$$

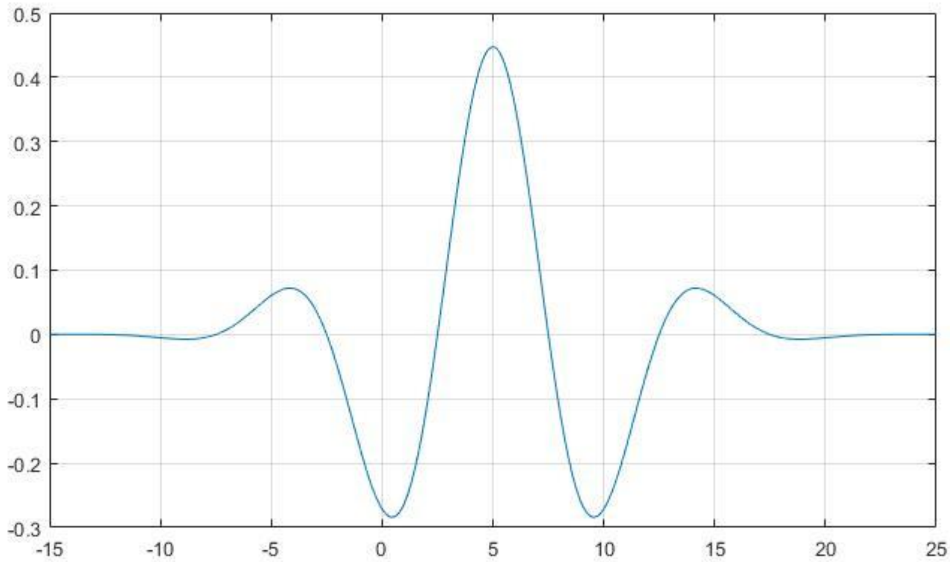

Figure 8. 3: Morlet's wavelet centered in $\boldsymbol{T = 5}$ *seconds and scaled with a factor* $\boldsymbol{a = 10}$

Figure 8.3 shows a wavelet that has been centered in $T = 5$ seconds and scaled with a factor $a = 10$. The original unscaled wavelet centered in $T = 0$ is usually called '*mother wavelet'* and the wavelets $\Psi_{a,T}$ will be the comparison functions that substitude the e^{iwt} functions used in the Fourier transform to create the Wavelet transform:

$$C(a,T) = \int_{-\infty}^{+\infty} a_s(t)\Psi_{a,T}dt \tag{29}$$

The result of the Wavelet transform will be the decomposition of the signal $a_s(t)$ in a sum of wavelets $\Psi_{a,T}$ weighted with the coefficients $C(a,T)$. Moreover, like with the Fourier Transform, the signal $a_s(t)$ can be recovered if the wavelet coefficients are known using the Inverse Wavelet Transform:

$$a_s(t) = \frac{1}{K_\Psi}\int_{a=0}^{+\infty}\int_{T=-\infty}^{+\infty} C(a,T)\Psi_{a,T}\, dT\,\frac{da}{a^2} \tag{30}$$

where $K_\Psi$depends on which “Mother wavelet” has been selected.

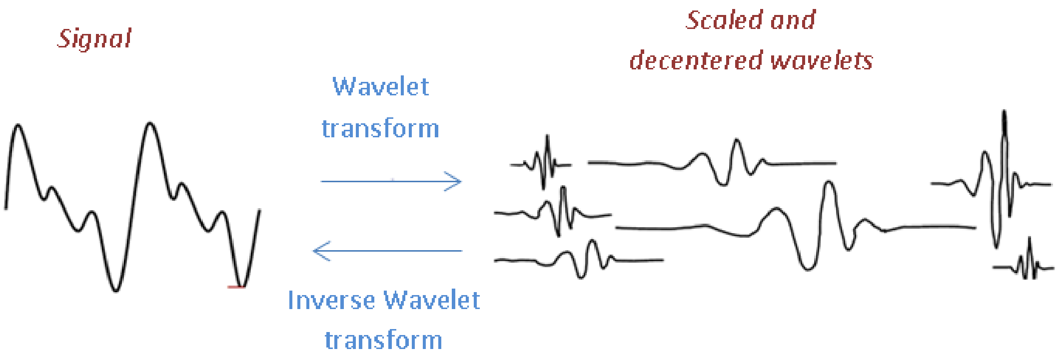

Figure 8. 4: Wavelet Transform

Finally, it is important to remark that the Wavelet transform can be performed with different types of Wavelets. In other words, there is more than one possible comparison function. Determining which of the wavelets types is the most appropriate to decompose a signal is a process of trial an error.

8.4. Wavelet analysis

In the previous section, the wavelet transform has been defined in the continuous version. However, the acceleration time histories in this project are discretized. Therefore, a discrete version of the wavelet transform is required.

However, discretizing times and frequencies T, ω where the wavelet transform is calculated, is not as intuitive as it may seem since the time and frequency are interrelated due to uncertainty principle [14]. For high frequencies time intervals will have to be narrow and for low frequencies they will have to be wide if we expect the wavelets to extract appropriately the frequency content.

Taking this into account, if dt is the time step of the acceleration time history $a_s(t)$ that wants to be decomposed, the frequency ω will be discretized following this expression:

$$\omega_k = \frac{\pi}{dt \cdot 2^{(k_{max}-k)/12}} \; ; \; a_k = \frac{1}{\omega_k} \; k \in Z \tag{31}$$

Where k is an integer that represents the different frequency bands and k_{max} is calculated from the number of points N of the acceleration time history:

$$k_{max} = 12 \cdot \frac{\log N}{\log 2} \tag{32}$$

The different times T where each wavelet will be centered will follow the following expression:

$$T_{k,n} = ndt2^{(k_{max}-k)/12} \; ; \; k, n \in Z \tag{33}$$

For each frequency k, n number of wavelets will be required to cover the whole time domain. Since the time interval a wavelet covers will be smaller with higher frequencies, n will increase with the frequency.

By performing this discretization, we will have $k \cdot n$ different wavelets $\Psi_{k,n}$that constitute the comparison functions for the Discrete Wavelet Transform:

$$C_{k,n}{}^{a_s} = dt \cdot \sum_{t=0}^{t_{end}} a_s(t) \cdot \Psi_{k,n} \tag{34}$$

The coefficients $C_{k,n}{}^{a_s}$ obtained with the Wavelet transform have the information of the variation of the frequency content and the amplitude of the signal. Therefore they are really closely related to the modulating functions $m_k(t)$ that were defined in *equation 23*. The only difference is that since we have $k \cdot n$ comparison functions we need the same number of modulating functions that have to satisfy *equation* 24. To fulfill this condition, the coefficients can be normalized, resulting in the desired functions $m_{k,n}(t)$:

$$m_{k,n} = \frac{{C_{k,n}}^{a_s}}{\sqrt{\Delta T_k \sum_{k,n} ({C_{k,n}}^{a_s})^2}} \; ; \; \Delta T_k = 2^{(k_{max}-k)/12} \tag{35}$$

Once the modulating functions are calculated, they can be used combined with the values of the Shock Response Spectrum (SRS_k) of the original signal to randomly generate new wavelet coefficients ${C_{k,n}}'$ with which a lot of synthetic acceleration time histories can be built that have the same frequency content variation than the original one:

$$a_{new}(t) = \sum_k \sum_n {C_{k,n}}' \Psi_{k,n} \tag{36}$$

8.5. Implementation of Wavelet Analysis

8.5.1. Decomposition of a signal using the wavelet transform

As a first step to implement Wavelet Analysis, a program has been written in *MATLAB* to decompose a discrete signal $x(t)$ into the sum of wavelets. This program can be found in *Annex C.*

In first place the program discretizes the frequency points ω_k and the time points $T_{k,n}$ taking into account the duration and time step of $x(t)$ according to the *equation 31*. The following step is to calculate all the wavelets $\Psi_{k,n}$ that will be used as comparison functions. In this particular case, since Morlet's wavelet has been used:

$$\Psi_{k,n}(t) = \frac{1}{\sqrt{a_k}} \cdot e^{i\Omega\left(\frac{t-T_{k,n}}{a_k}\right)} \cdot e^{-\frac{1}{2}\left(\frac{t-T_{k,n}}{a_k \sigma}\right)^2} \tag{37}$$

Later on the wavelets coefficients can be calculated with the wavelet transform:

$${C_{k,n}}^x = dt \cdot \sum_{t=0}^{t_{end}} x(t) \cdot \Psi_{k,n} \tag{38}$$

Finally the original signal can be recovered by applying the inverse wavelet transform.

In order to test the program, an invented signal $x(t)$ shown in *figure 8.5* was considered. This signal is divided into three clearly differentiated parts. Each of these parts has different frequencies and amplitudes.

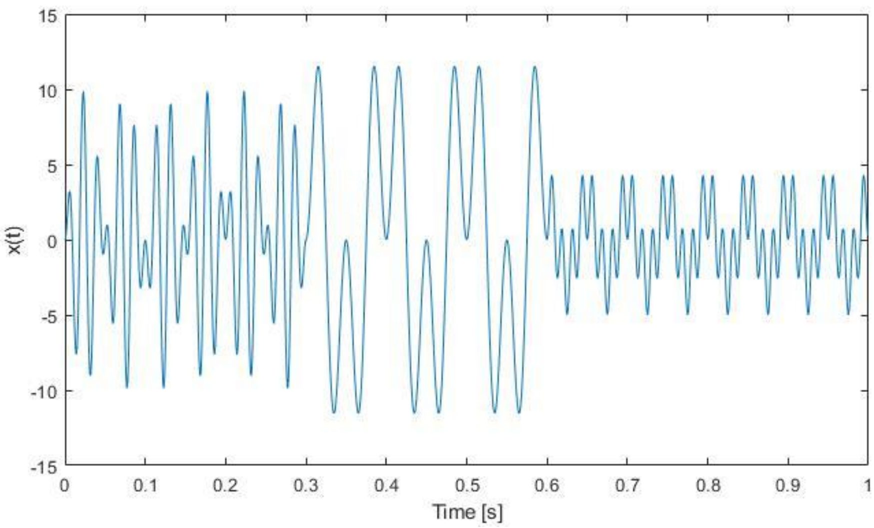

Figure 8. 5: Signal x(t) with three frequency-differentiated sections

To determine the frequencies that appear in each part of the signal we can perform the Fourier Transforms:

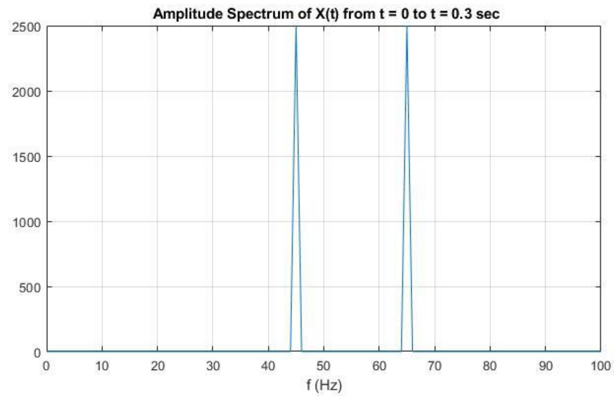

Figure 8. 6: Fourier Transform of signal $x(t)$ in the first section

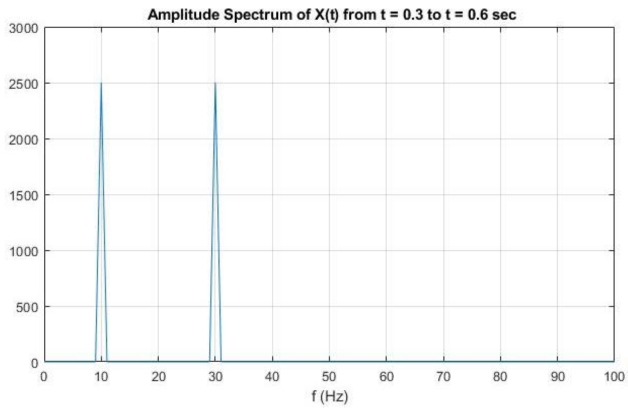

Figure 8. 7: Fourier Transform of signal $x(t)$ in the second section

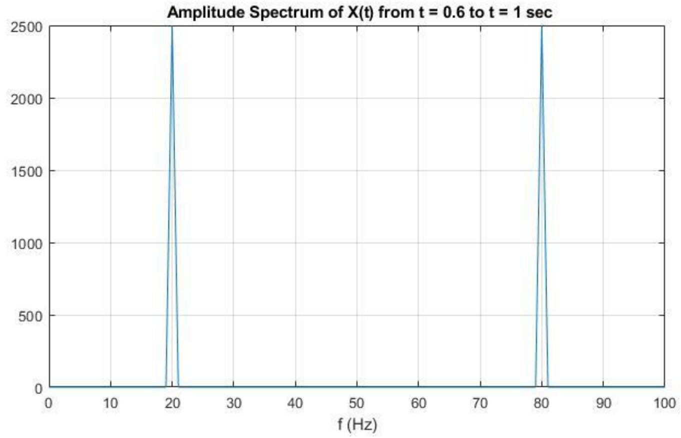

Figure 8. 8: Fourier Transform of signal $x(t)$ in the third section

From these figures we can conclude that the most important frequencies are:

- 45 Hz and 65 Hz in the first time interval
- 10 Hz and 30 Hz in the second time interval
- 20 Hz and 80 Hz in the third time interval

Once the most important frequencies in the three different sections are known, it is possible to check if the wavelet transform is able to locate those frequencies in the time domain by plotting the wavelet coefficients as a function of time and frequency. The resulting surface is shown in *figures 8.9* and *8.10.* It is not hard to observe that the coefficients are higher for the dominant frequencies of the signal in the correct time interval.

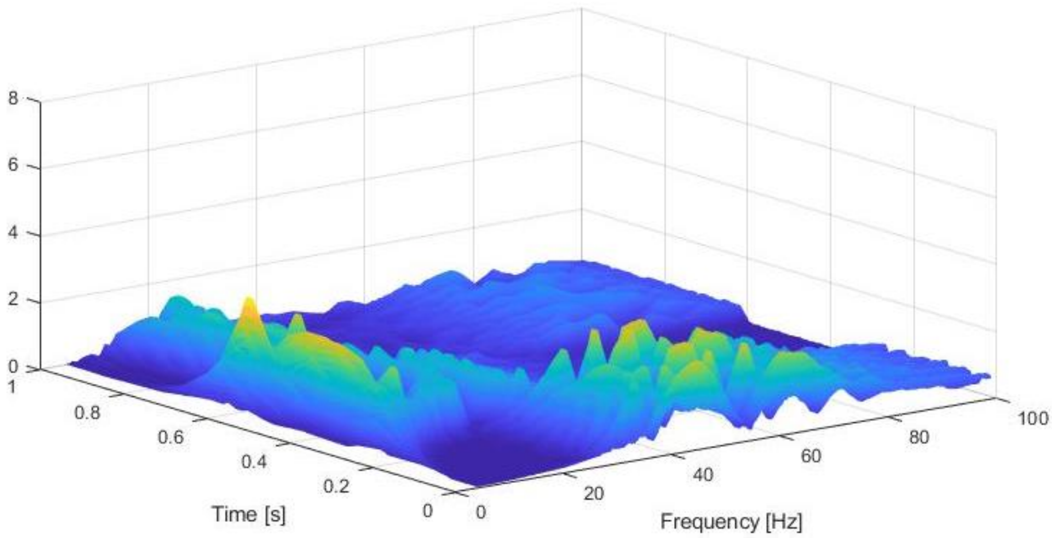

Figure 8. 9: Wavelet coefficients 3D view

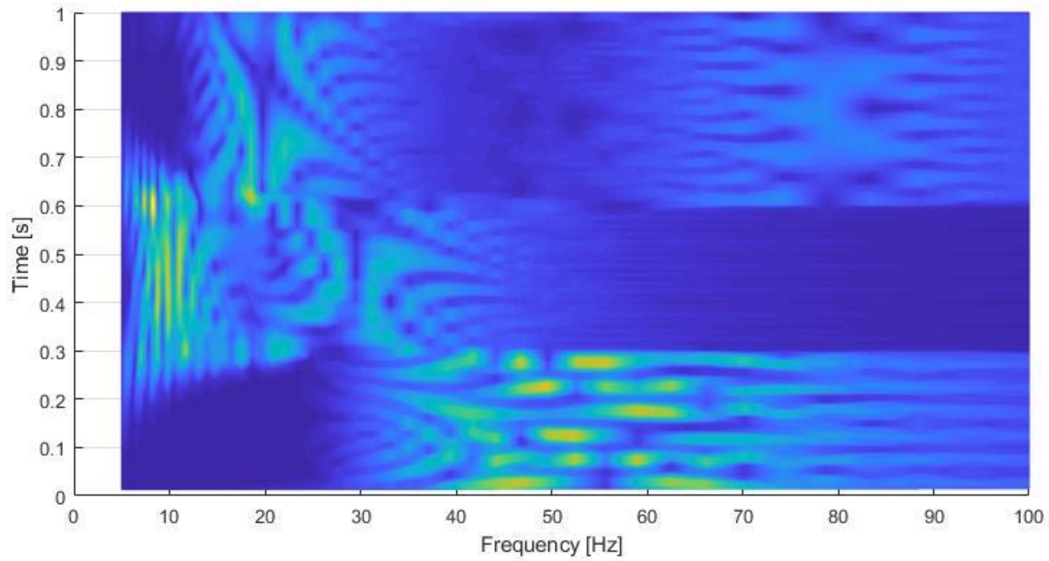

Figure 8. 10: Wavelet coefficients top view

Since the wavelet coefficients seem to make sense, the last step is to apply the inverse Wavelet transform to recover the original signal, expressed as a sum of all the individual wavelets. The result is shown in *figure 8.11*. Although the reconstruction is not completely perfect, the results are satisfactory. The existing error is normal taking into account that the discrete wavelet transform is just an approximation.

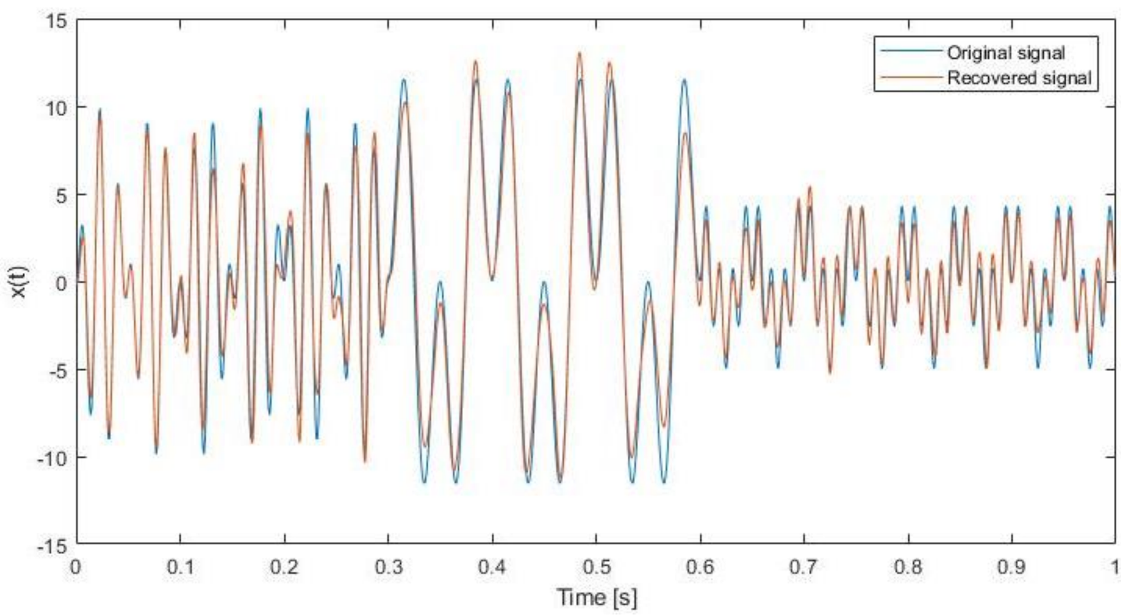

Figure 8. 11: Signal recovery from wavelet coefficients

8.5.2. Generation of synthetic signals using Wavelet Analysis

As an example, we would like to obtain a signal that has the same frequency content variation and similar SRS than signal $x(t)$ shown in *figure 8.5*. This example was implemented in *MATLAB* and can be found in *Annex C*. The first step was to calculate the Shock Response Spectrum of the original signal $x(t)$, which is shown below:

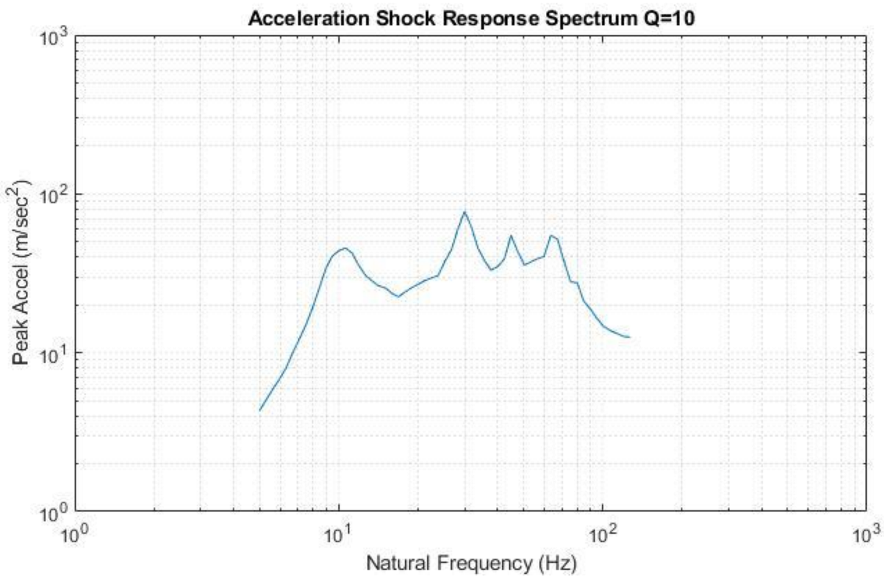

Figure 8. 12: SRS of signal x(t)

The wavelet coefficients calculated in the previous section already have the information of the frequency content variation. From these coefficients we can calculate the modulating functions using *equation 35*. Once the modulating functions are available, they can be used to calculate new wavelet coefficients to generate a new signal that is different to the original but that has the same frequency content and similar Shock Response Spectrum. To match the desired Shock Response Spectrum, a trial and error process was implemented.

In the calculation of the new wavelet coefficients, the sign is assigned randomly. As a consequence, there are a lot of possibilities for the new equivalent signal. *Figures 8.13, 8.14* and *8.15*, are three of these possibilities that were obtained.

The results show that the random generation of the new coefficients affects the form of the SRS curves. Moreover, although the matching of these curves is not perfect, it is reasonable. The existing error makes sense taking into account that a trial and error process was used. Reducing this error is possible at the expense of higher computation time which was not desired.

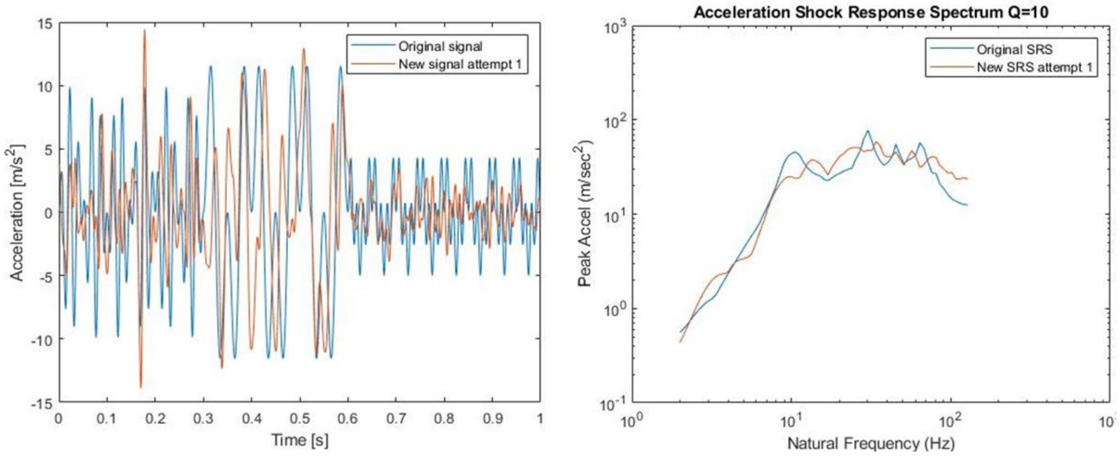

Figure 8. 13: New signal attempt 1

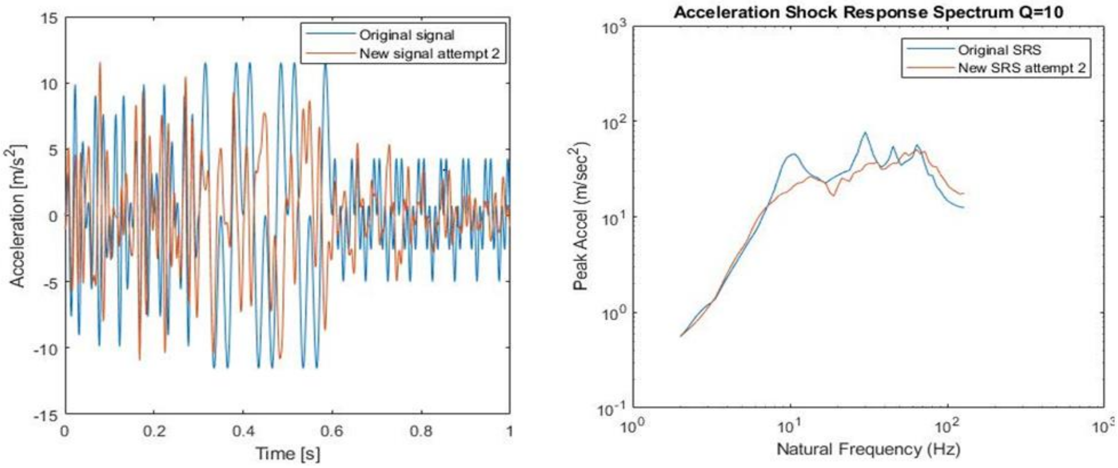

Figure 8. 14: New signal attempt 2

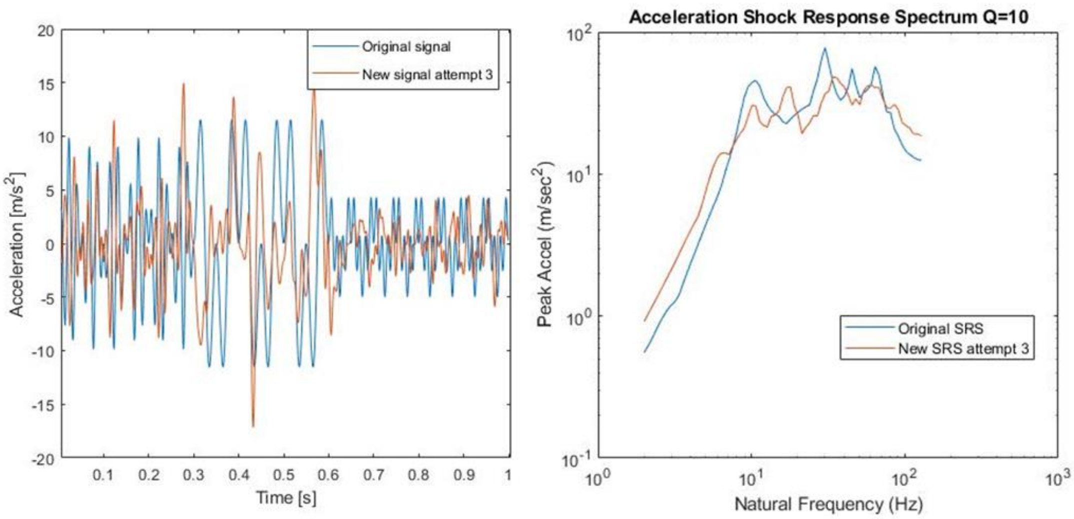

Figure 8. 15: New signal attempt 3

8.5.3. Generation of synthetic signals that are equivalent to a flight transient

In the previous sections, the theory of wavelet analysis has been explained and implemented on a simple example with good results. Now the objectives will be:

- Apply the same methodology to generate synthetic acceleration time histories from the data collected during the launch of a satellite
- Carry out simulations with the finite element model of the satellite with the new signals
- Compare results and determine whether the obtained synthetic signals are a good representation of the original flight transient

8.5.3.1. Transient in Y direction

The flight transient that has been studied in this work has a simpler shape in the Y direction and for this reason it will be studied in first place. *Figures 8.16*, *8.17* and *8.18* show the obtained synthetic signals and the SRS comparisons of three different attempts when applying Wavelet Analysis. Two important observations should be made:

- As it was expected, due to the random generation of the wavelet coefficients, the result obtained in each attempt is different.
- The maximum values of the obtained synthetic signals exceed those of the real flight transient.

Once the synthetic signals have been obtained, it is possible to simulate the response of the satellite using the dynamic model described in previous chapters. The results of the simulations are given in *table 8.1*.

MAXIMUM ACELERATION OF THE MASSES [m/s^2]				
Direction	Real flight transient	Attempt 1	Attempt 2	Attempt 3
Y	1.62 (mass 3)	2.55 (mass 3)	3.11 (mass 3)	2.94 (mass 3)

MAX NUMBER OF CYCLES			
Real flight transient	Attempt 1	Attempt 2	Attempt 3
210(mass 3, Y)	206 (mass 3, Y)	200 (mass 3, Y)	202 (mass 3, Y)

Table 8. 1: Dynamic analysis results using signals synthesized with Wavelet Analysis (Y)

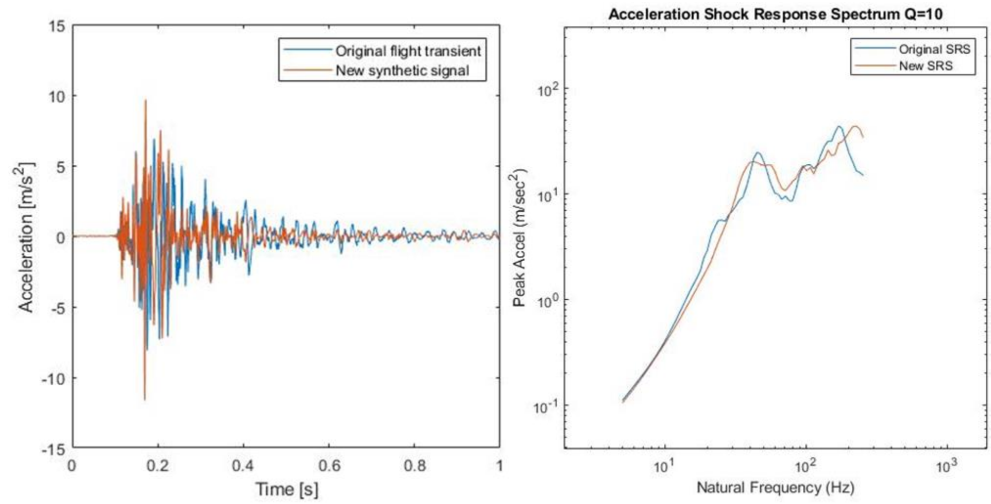

Figure 8. 16: Synthetic signal generated using Wavelet Analysis (Attempt 1)

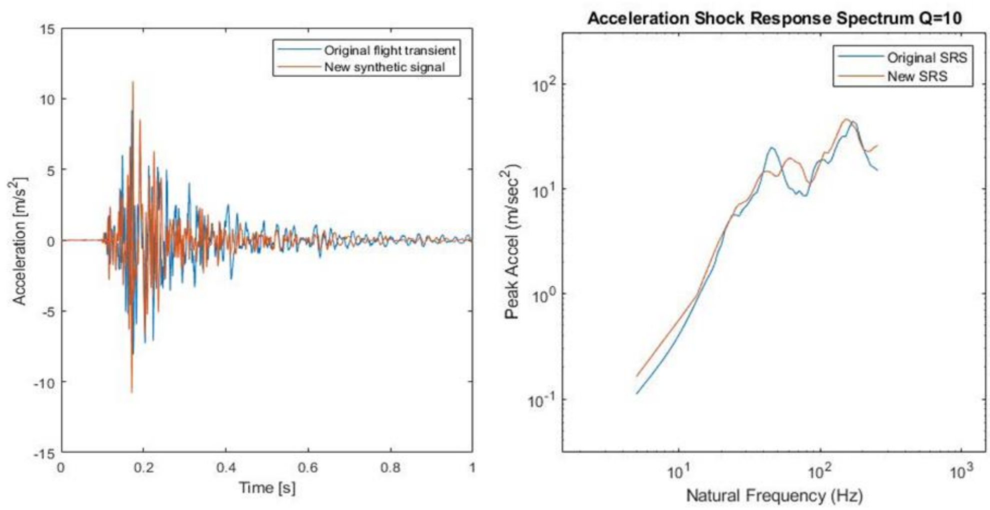

Figure 8. 17: Synthetic signal generated using Wavelet Analysis (Attempt 2)

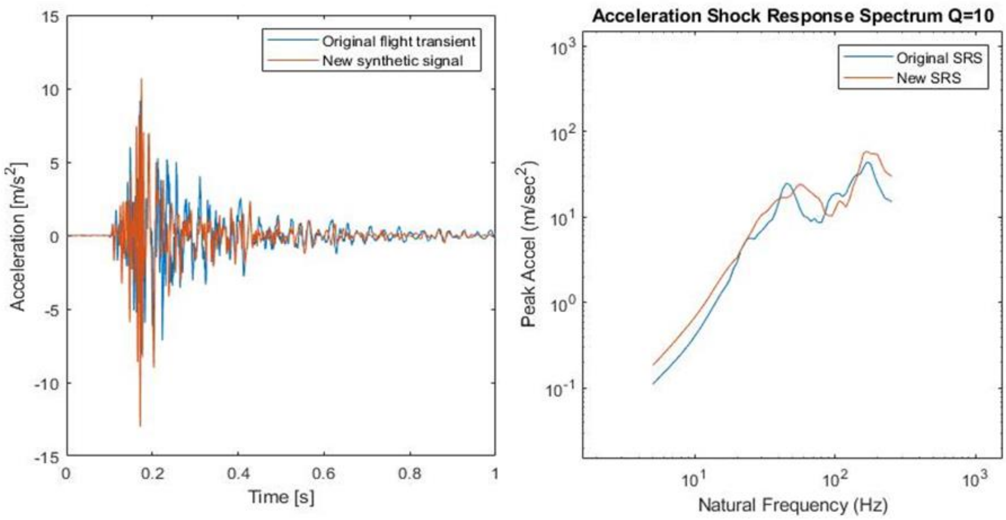

Figure 8. 18: Synthetic signal generated using Wavelet Analysis (Attempt 3)

We can clearly observe that the acceleration values are higher than those obtained with the original flight transient. This is probably related to the fact that the maximum values of the synthetic signals where higher than the maximum values of the original transient. Since there are many possible signals with the same frequency content, a solution might be to look for those possibilities that have maximum values that do not exceed those of the original signal. Applying the methodology with this new restriction, the new synthetic signals are much similar to the originals and the SRS is still similar. *Figures 8.19* to *8.25* show six new attempts. The simulation results for each of these attempts are given in *table 8.2.*

MAXIMUM ACELERATION OF THE MASSES [m/s^2]				
Direction	Real flight transient	Attempt 1	Attempt 2	Attempt 3
Y	1.62 (mass 3)	1.67 (mass 3)	2.67 (mass 3)	1.93 (mass 3)

Direction	Attempt 4	Attempt 5	Attempt 6
Y	1.65 (mass 3)	2.69 (mass 3)	1.60 (mass 3)

MAX NUMBER OF CYCLES			
Real flight transient	Attempt 1	Attempt 2	Attempt 3
210(mass 3, Y)	205 (mass 3,Y)	201 (mass 3, Y)	208 (mass 3, Y)

Attempt 4	Attempt 5	Attempt 6
205 (mass 3, Y)	209 (mass 3, Y)	204 (mass 3, Y)

Table 8. 2: Results using signals synthesized with Wavelet Analysis with restrictions (Y)

With the new restriction, the acceleration results are closer to the results obtained with the original flight transient. However, a noticeable variability still exists among the different attempts. Despite having equal frequency content and similar maximum values, each synthetic signal leads to slightly different responses of the satellite. The conclusion that should be extracted from these results is that among all the possibilities of signals with similar levels and equal frequency content, there will be some that subject the satellite to more critical conditions than others. Therefore, satellites should be tested on those most critical conditions found when applying this methodology.

Regarding the number of cycles, the results are really similar to those obtained with the original flight transient. In first place, this means that the synthetic signal properly describes the real phenomena. Secondly, given the low number of cycles, the application of this methodology in vibration tests will reduce fatigue damage.

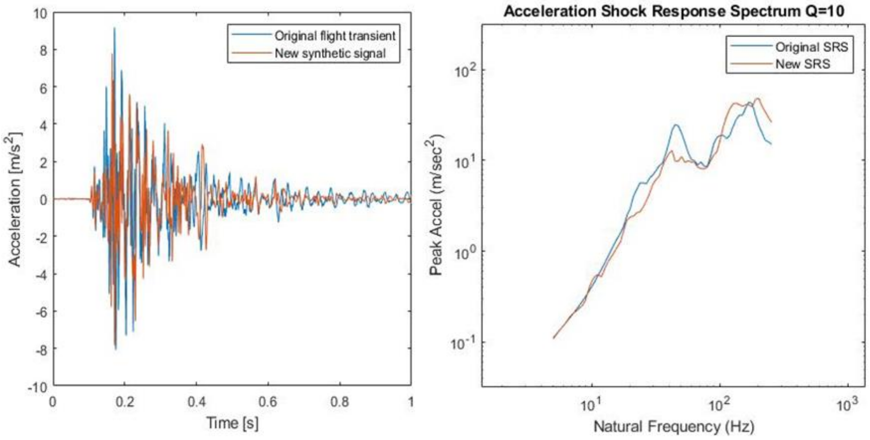

Figure 8. 19: Synthetic signal using Wavelet Analysis with restrictions (Attempt 1Y)

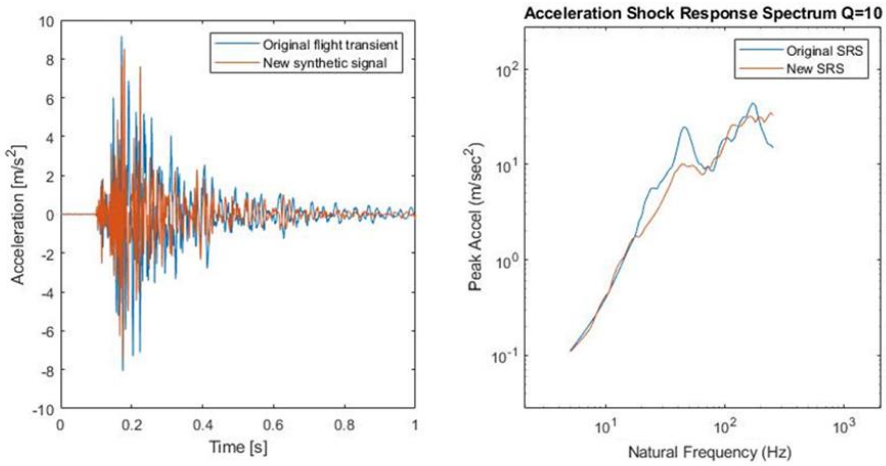

Figure 8. 20: Synthetic signal using Wavelet Analysis with restrictions (Attempt 2Y)

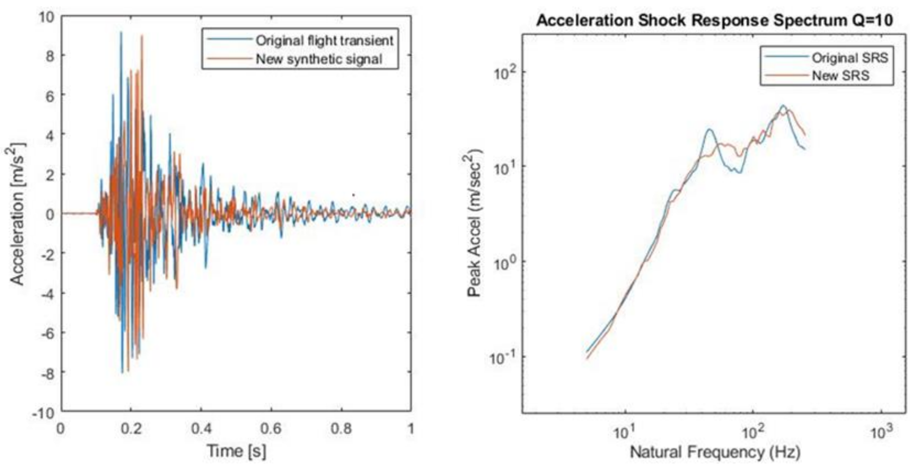

Figure 8. 21: Synthetic signal using Wavelet Analysis with restrictions (Attempt 3Y)

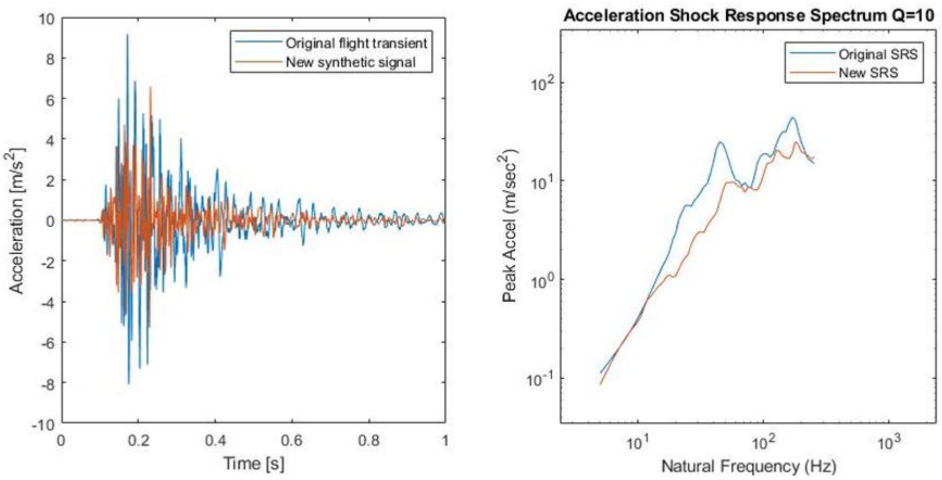

Figure 8. 22: Synthetic signal using Wavelet Analysis with restrictions (Attempt 4Y)

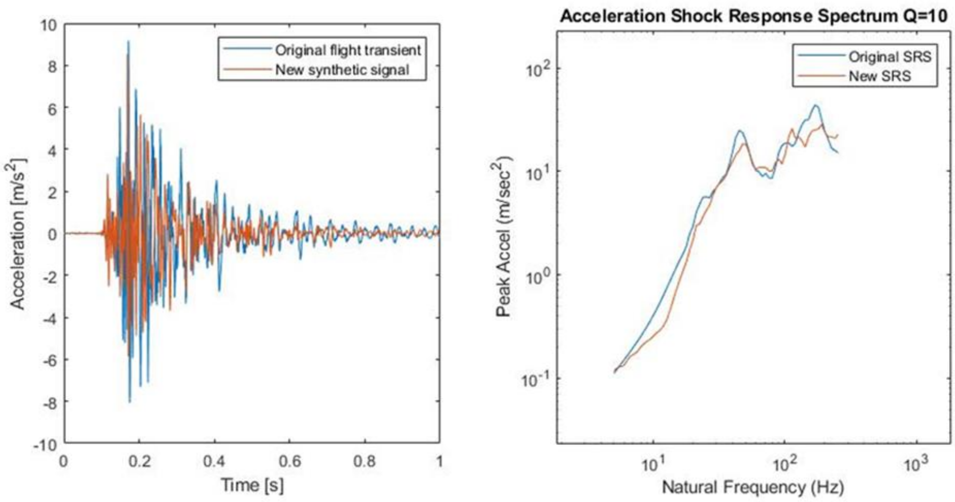

Figure 8. 23: Synthetic signal using Wavelet Analysis with restrictions (Attempt 5Y)

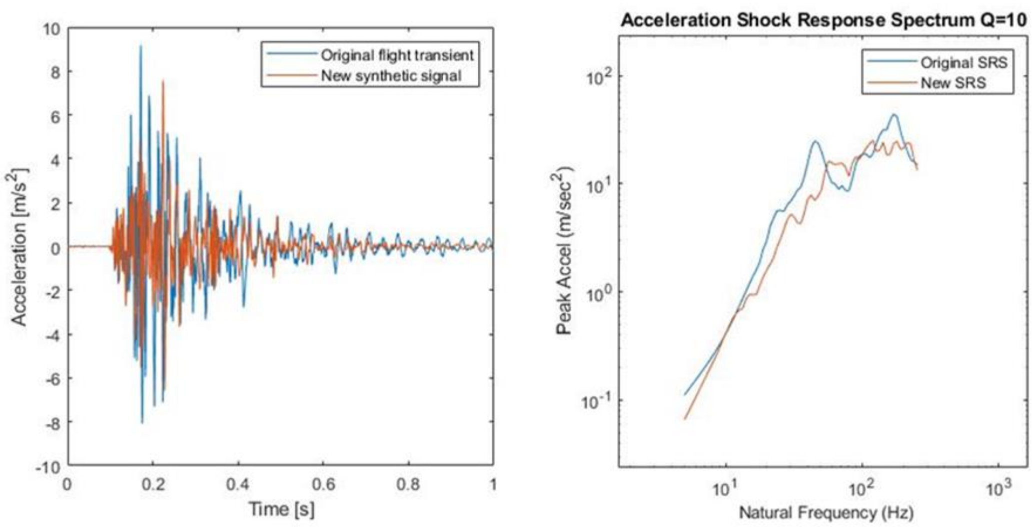

Figure 8. 24: Synthetic signal using Wavelet Analysis with restrictions (Attempt 6Y)

8.5.3.2. Transient in Z direction

Taking into consideration the conclusions extracted from the study of the transient in Y direction, the methodology has been applied also with the transient in Z direction. The results for six different attempts are shown in *table 8.3*. The most critical attempt is shown in *figure 8.25*.

MAXIMUM ACELERATION OF THE MASSES [m/s^2]				
Direction	Flight transient	Attempt 1	Attempt 2	Attempt 3
Z	34.60 (mass 3)	35.27 (mass 3)	22.89 (mass 3)	30.33 (mass 3)

Direction	Attempt 4	Attempt 5	Attempt 6
Z	29.97 (mass 3)	23.28 (mass 3)	30.72 (mass 3)

MAX NUMBER OF CYCLES			
Flight transient	Attempt 1	Attempt 2	Attempt 3
259 (mass 3, Y)	242 (mass 3,Y)	236 (mass 3, Y)	247 (mass 3, Y)

Attempt 4	Attempt 5	Attempt 6
239(mass 3, Y)	251 (mass 3, Y)	247(mass 3, Y)

Table 8. 3: Results using signals synthesized with Wavelet Analysis with restrictions (Z)

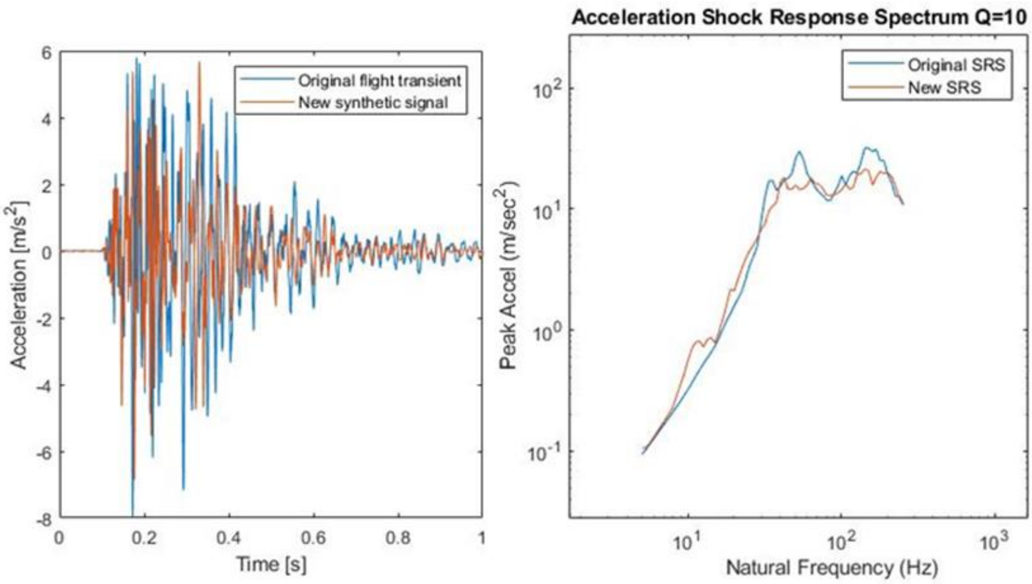

Figure 8. 25: Synthetic signal using Wavelet Analysis with restrictions (Attempt 1Z)

The conclusions that can be extracted from these results are the same than for transient in Y direction:

- The accelerations are similar to those obtained with the original transient but some variability exists among different attempts.
- The most critical attempt should be used to test the satellite.
- The number of cycles is also similar and will not lead to significant fatigue damage.

8.5.3.3. Transient in X direction

The same methodology has been applied for the transient in X direction. However, this case is somehow different to the previous cases since the steady state mean value of the acceleration is not zero. The wavelets used for the decomposition of signals in this chapter oscillate around zero value and cannot reproduce signals with non-zero mean values. Although a more complex decomposition could be implemented, we will ignore the effects of this steady state acceleration.

MAXIMUM ACELERATION OF THE MASSES [m/s^2]				
Direction	Flight transient	Attempt 1	Attempt 2	Attempt 3
X	12.33 (mass 2)	11.01 (mass 3)	13.93 (mass 3)	13.10 (mass 3)

Direction	Attempt 4	Attempt 5	Attempt 6
X	11.36 (mass 3)	13.40 (mass 3)	10.67 (mass 3)

MAX NUMBER OF CYCLES			
Real flight transient	Attempt 1	Attempt 2	Attempt 3
179 (mass 3, Y)	189 (mass 3,Y)	187 (mass 3, Y)	187 (mass 3, Y)

Attempt 4	Attempt 5	Attempt 6
179 (mass 3, Y)	188 (mass 3, Y)	188 (mass 3, Y)

Table 8. 4: Results using signals synthesized with Wavelet Analysis with restrictions (X)

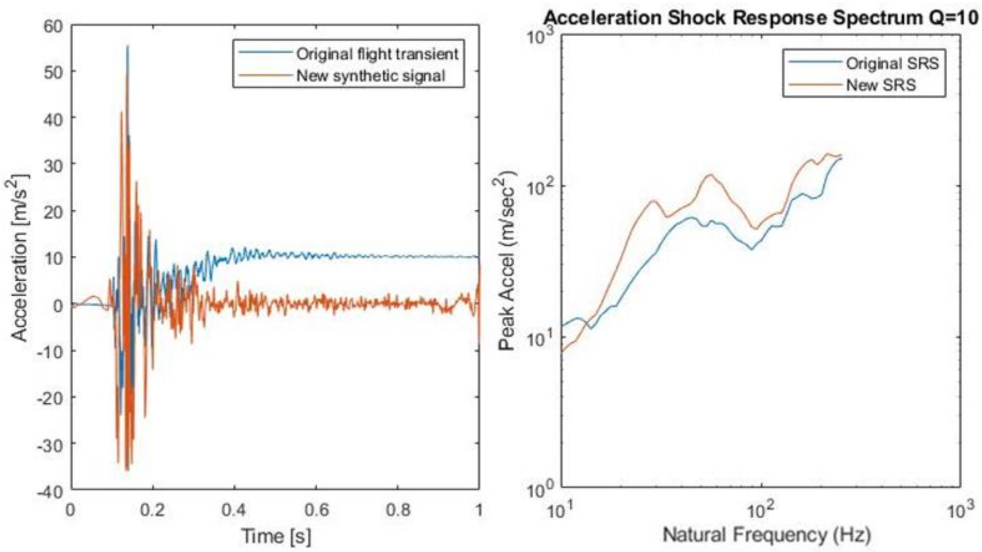

Figure 8. 26: Synthetic signal using Wavelet Analysis with restrictions (Attempt 2X)

8.5.3.4. Transient in all directions simultaneously

Finally, it is interesting to determine what will happen if the most critical synthetic signals in each axis are applied simultaneously. The results are given in the following table:

MAXIMUM ACELERATION OF THE MASSES [m/s^2]		
	Real flight transient	Most critical attempts
X	12.45 (2)	15.63 (2)
Y	1.503 (3)	2.83 (3)
Z	42.45 (3)	42.08 (3)

MAX NUMBER OF CYCLES	
Real flight transient	Most critical attempts
215 (mass 1, Y dir)	221 (mass 3, Y dir)

Table 8. 5: Results using signals synthesized with Wavelet Analysis with restrictions (X,Y,Z)

The acceleration results are higher in both X and Y direction, but not in the Z direction. This shows that there are important differences between axis by axis and multiple-axis vibration. More importantly, it shows that the most critical single axis vibrations do not lead to the most critical multiple-axis vibrations.

8.6. Conclusions

This chapter has studied the possibility of using Wavelet Analysis to generate synthetic signals that are more similar to the real flight transients. In previous chapters other methods had been studied and their ability to reproduce a similar response in the satellite than the original transient has been assessed. These methods were based on the hypothesis that signals with equal Shock Response Spectrums will generate equal damage on the structure, without taking into account the time dimension. However, in reality, the accumulation of damage due to fatigue relies on the time history of loads. By using Wavelet Analysis it is possible to capture not only the frequencies and the amplitudes of the original signal but also the moment and the order in time in which they appear.

It has been shown that an aleatory signal can be recovered using the Wavelet coefficients obtained by applying the Wavelet Transform. However, using this signal to test the satellite would make no sense since it would have the same effect than using the original transient in the first place. The key idea is that from these coefficients it is possible to calculate what it has been called as modulating functions that can be used to randomly obtain a lot of signals that are different to the original transients but that have the same frequency content and similar SRS. These signals can be useful to test the satellites since flight transients are not fully predictable and are never the same.

From the implementation of this methodology we have observed that in average, the randomly generated signals produce a similar response in the model of the satellite than the original transient. For this reason we do consider this methodology a viable approach that could be used to replace the Classical Sine Sweep in the task of defining the low frequency vibration specification for the design and testing of satellites. However, a small variability exists and some of the signals led to higher accelerations and others to smaller accelerations. The conclusion extracted is that among all the signals with equal frequency content and similar SRS there are some that produce a more critical response in the satellite than others and therefore these signals should be the ones considered during design and test of the satellites.

Finally, it must not be forgotten that in this study only one flight transient has been considered. The modulating functions are able to generate signals that resemble the conditions of the specific flight transient from which they have been calculated. It cannot be assumed that these signals obtained from a single flight transient will reproduce the conditions of a future real flight. For this reason, if Wavelet Analysis is to be considered seriously to define the low frequency vibration specification, a different approach that considers more than one flight transient has to be used. Perhaps, if sufficient flight transient data is available, it is possible to statistically find some modulating functions that reproduce most of flight transients. Moreover, only Morlet's wavelet has been considered. It would be really interesting to repeat these experiments with different wavelets and perhaps find one that improves the results.

Chapter 9

Conclusions

9.1. Analysis of the work

The present work has started by giving an overview of the State of the Art of loads verification in satellites. Understanding the limitations of current methodologies is essential to carry out further investigations and make progress in the field. The simulation and testing of low frequency transients with the sine-test is one of the methodologies that could be improved. The long duration of the test and the unwanted margins generated from the assumption that both environments are equivalent constitute important disadvantages. For this reason, it was decided that the study of the sine-test and two possible alternatives, the Fast Sine Sweep and the Wavelets methods, should be the focus of the investigation in this work.

Firstly, the project has studied the three signal-generation methods based on the assumption that acceleration time histories with similar Shock Response Spectrum cause equivalent damage. In this way, the different methods have been compared based on their ability to synthesize signals that match a given SRS.

In order to check if the assumption of equivalence based on SRS was reasonable, simulating the excitation of a satellite at its base with the different signals was necessary. With this purpose, a Finite Element Model of a really simplified satellite has been developed for *AFECTOS* in *MATLAB*. To calculate the response of such model to base excitations, two approaches have been considered: Modal Analysis and Newmark's time integration method. Modal Analysis neglects transient effects and is restricted to harmonic signals, which means that it could only by used to study the sine-test. On the other hand, the Newmark method takes transients into account and can be used regardless of the characteristics of the excitation, which made it suitable to study all the methods considered in this work.

The first simulations of the FEM have been done with simple sine waves with constant amplitude and frequency. These simple simulations have validated that the Modal Analysis and the Newmark Method approaches were correctly programmed. The next step has been to simulate the sine-tests or Classical Sine Sweeps obtained from a simple fictitious transient.

Finally, sine-tests obtained from a real flight transient have been simulated using both Modal Analysis and Newmark's time integration method. The comparison of the results obtained with each of the methods was desired to determine the influence of transient effects, which were not considered with Modal Analysis. Moreover, the results have also been compared with the response of the satellite to the original flight transient in order to determine if the equivalence between the sine-test and the original signal is reasonable.

The sine-tests considered until this point, were a simplification and did not sweep the frequencies like in real tests but rather maintained specific frequencies in time intervals. For this reason, carrying out simulations with non-simplified sine-tests has been the next step of the work. From the results of these simulations, conclusions regarding the influence of the sweep rate can be drawn.

After all these simulations of the sine test, it was time to study the Fast Sine Sweep and the Wavelets signal-generation methods. Given the non-harmonic characteristic of the signals obtained with these methods, the calculation of the response of the satellite to the resulting signals was done with the Newmark method. The results of the simulations have been compared with those from the simulations of the sine-test and the original flight transient.

Finally, the possibility of using Wavelet Analysis to synthesize more realistic signals has been studied. Previous described methods, were able to capture the most important frequencies and amplitudes of the original signal but not the time points in which they took place. As it has been explained, the time dimension is important since the accumulation of damage depends on the time history of loads. By using Wavelet Analysis it has been possible to synthesize signals that have the same frequency content than the original flight transient. The excitation of the satellite with these signals has also been simulated and conclusions have been drawn.

9.2. Results and general conclusions

The motivation of the present work arose from the limitations involved when using the sine-test to simulate and test low-frequency loads in satellites. The first results concern the significance of these limitations:

- Duration of the sine-test: it has been shown that when using a reasonable sweep rate the equivalent sine-sweep for a *1 second* signal has a duration of *350 seconds*. Taking into account that normally several runs for each axis are necessary, it is clear that the sine-test is a slow methodology.

- Ability of the sine-sweep to match the SRS of the original flight transient: The results show that equivalent sine-sweeps are able to match the SRS of simple signals. However, when signals are more complex, such as a flight transient, an important mismatch in the SRS appears.
- Capacity of the sine-sweep to generate similar loads in the satellite than the original flight transient: As it can be seen in *table 9.1*, the maximum loads that the equivalent sine-sweep generates in the satellite are not as high as the maximum loads generated by the original flight transient. In other words, the assumed equivalence is not true.

MAXIMUM ACELERATION OF THE MASSES [G]		
Direction	Real flight transient	ESI (Newmark)
X	12.33 (mass 2)	10.18 (mass 2)
Y	1.62 (mass 3)	1.42 (mass 3)
Z	34.6 (mass 3)	28.42 (mass 3)

Table 9. 1: Maximum accelerations with the sine-test

- Influence of the sweep rate: the sweep rate affects the amplitude of the response of the system. Moreover, the results show that in a MDOF system, the influence of the sweep rate depends on the frequency interval. In our specific model, a positive sweep rate led to higher accelerations between 25 to 70 Hz, lower accelerations between 10 and 25 Hz and similar accelerations between 70 and 100 Hz.
- Number of cycles: When excited with the sine-sweep, the model is subjected to an unreasonably high number of cycles. In a real test this implies that the structure will be over-tested from a fatigue point of view.

The above-mentioned points show that the limitations of the sine-test are in fact important. Therefore it makes sense that new and alternative approaches are investigated. In this work three alternatives have been studied:

- Fast Sine Sweep method

- Wavelets method
- Wavelet Analysis to synthesize signals with equal frequency content than the original flight transient

The first two methods synthesize signals based on the frequency spectra of the original flight transient, but do not consider the time instants in which those frequencies act. After implementing both methods and carrying out the simulations the following results were obtained:

- Duration and number of cycles: the resulting synthetic signals now have a similar duration than the original flight transient and the number of cycles is considerably reduced.
- Ability of the signals to match the SRS of the flight transient: the signals generated with the FSS and the Wavelets method match correctly the SRS of the original flight transient, something the sine-test was not able to do. The Wavelet method seems to be more precise in this task, but it is important to remark that the optimization process of the FSS was simplified. The implementation of the FSS method with a more rigorous optimization is interesting for future studies.
- Influence of multiple-axis loading conditions: The results indicate that introducing the loads axis by axis is not equivalent to real conditions in which multiple-axis loading conditions exist. When excitations are introduced in multiple axes simultaneously the resulting loads are higher.
- Capacity of the signals to generate similar loads than the original flight transient: From *table 7.4* it can be concluded that when simulations and tests are carried axis by axis both methods reproduce loads that are similar to those generated by the original transient. However, these loads are not always higher than those generated by the original signal, which means that the model could be under-tested. On the other hand, *table 7.5* shows that when loads are introduced in multiple axes simultaneously the FSS method does generate higher loads.

Finally, Wavelet Analysis considers the time domain in the synthesis of equivalent signals and has led to the following results:

- Duration and number of cycles: the resulting synthetic signals have equal duration than the original flight transient and the number of cycles almost identical.

- Ability of the signals to match the SRS of the flight transient: the matching of the SRS is worse than with the FSS and the Wavelets method.
- Capacity of the signals to generate similar loads than the original flight transient: multiple signals can be obtained from a single transient and the variability with respect to the loads they generate is important. If most critical signals are chosen then the satellite will be over-tested with respect to the original transients.

Taking into account the results summarized above, the following general conclusions can be drawn:

- The sine-test is the worst approach among all the methods studied in this work. The signal is completely different to the original transient and the loads it generates cannot be considered equivalent. Moreover, the sine-sweep influences the amplitude of the resulting loads.
- The FSS and Wavelets methods are interesting alternatives, they are able to synthesize signals that are more similar to the original transients and the resulting loads are also more similar to those produced by the original transients. The low frequency load specification for satellites could be appropriately defined based on signals obtained with both of the methods. The process to obtain a valid specification from several flight transients is left as an interesting topic for future studies.
- The Wavelet Analysis method has shown that it is possible to obtain signals with a frequency content that is similar to the one of the original transient. However, the obtained results show that the generated loads are variable depending on the synthetic signal that is chosen from all the existing possibilities. Finding the modulating functions that are most appropriate to define the specification of the low frequency loads for satellites should be possible and is left as an interesting task for futures studies.

Bibliography

[1] Spacecraft Mechanical Loads Analysis Handbook, ECSS-E-HB-32-26A, Noordwijk, The Netherlands, February 2013

[2] Space engineering, Structural factors of safety for spaceflight hardware, ECSS-E-ST-32-10C Rev. 1, Noordwijk, The Netherlands, March 2009

[3] Ewins D. J. – Modal Testing: Theory, Practice and Applications – Research Studies Press LTD, Baldock, England, Second Edition, 2000

[4] Underwood, M. A., & Keller, T. (2001). Recent system developments for multi-actuator vibration control. *Sound and Vibration*, *35*(10), 16-23.

[5] Irvine, T. (2002). An introduction to the shock response spectrum. *Rev P, Vibrationdata*.

[6] Giclais, S., Lubrina, P., Stéphan, C., Böswald, M., Govers, Y., Ufer, J., & Botargues, N. (2011). New excitation signals for aircraft ground vibration testing. In *International Forum on Aeroelasticity and Structural Dynamics 2011*.

[7] E.Cavro, N.Roy, A. Girard, P.E. Dupuis, A. Hot, F. Gant. Fast Sine Sweep as an alternative to classical sine sweep for S/C qualification.

[8] Gen, M., & Lin, L. (2007). Genetic algorithms. *Wiley Encyclopedia of Computer Science and Engineering*, 1-15.

[9] Chang, S. Y. (2004). Studies of Newmark method for solving nonlinear systems: (I) basic analysis. *Journal of the Chinese Institute of Engineers*, *27*(5), 651-662.

[10] Chapter 7, Non-linear Seismic Response of Structures, Introduction to earthquake engineering, *NPTEL Civil Engineering*

[11] Liu, M., & Gorman, D. G. (1995). Formulation of Rayleigh damping and its extensions. *Computers & structures*, *57*(2), 277-285.

[12] Meirovitch, L. (2010). *Fundamentals of vibrations*. Waveland Press.

[13] Montejo, L. A., & Suárez, L. E. (2007). Aplicaciones de la Transformada Ondícula (“Wavelet”) en ingeniería estructural. *Mecánica Computacional*, *26*, 2742-2753.

[14] Carnicero, A. (1999). Evaluación de la vulnerabilidad sísmica de puentes.

Internet References

(I) http://www.vibrationdata.com/SRS.htm

Annex C: Matlab codes

Synthesis of a simplified equivalent sine sweep from SRS data

```
%% Simplified sine sweep

% Marcos Feria Cerrada
% Delft 2019

% This program gives the acceleration time history of a simplified sine
% sweep. This sine sweep maintains each frequency for a desired time
% interval so that the steady state response is reached.
% The sine sweep amplitudes are calcualted according to the ESI, equivalent
% sine input.

% Necessary input:
% SRS_max matrix
% The first column of SRS_max is the frequency points of the SRS
% The second column of SRS_max is the acceleration values of the SRS

clear all

% Parameters

Q = 10; % Damping factor
deltat = 0.001; % Time step ot the acceleration time history
time_per_frequency = 5; % Time each frequency will be maintained

%% ESI

esi = [SRS_max(:,1) SRS_max(:,2)/Q];

time = 0:deltat:length(SRS_max)*time_per_frequency;
time = transpose(time);
f = zeros(length(time),1); % frequency
a = zeros(length(time),1); % amplitude

for i=1:length(SRS_max)
    f(((i-
1)*(time_per_frequency/deltat)+1):(i*(time_per_frequency/deltat))+1)=SRS_max(i,1);
    a(((i-
1)*(time_per_frequency/deltat)+1):(i*(time_per_frequency/deltat))+1)=SRS_max(i,2)/Q;
end

acc_time_history = zeros(length(time),2);
acc_time_history(:,1)=time;
acc_time_history(:,2)=a.*sin((2*pi).*(f.*time));

plot(acc_time_history(:,1),acc_time_history(:,2));
xlabel('Time [s]');
ylabel('Acceleration m/s^2');
title('Equivalent Sine Input Acceleration');

% Output acc_time_history
```

Synthesis of an equivalent sine sweep from SRS data

```matlab
%%Synthesis of an equivalent sine sweep from SRS data

% Marcos Feria Cerrada
% Delft 2019


% This program gives the acceleration time history of a sine sweep
% The necessary input data is 2 column matrix, in this case called SRS_max
% The first column of SRS_max is the frequency points of the SRS
% The second column of SRS_max is the acceleration values of the SRS
% Calculation of sine sweep based on ESI, Equivalent Sine Input
% Output name is acc_time_history = [time ; acceleration]

%% DESIGN PARAMETERS

Q = 10; % damping factor, typical value Q = 10
R = 1; % desired sweep rate [octaves/minute], typcial value R = 1
max_freq = 100; % maximum frequency to consider
dt = 0.0001; % time step for the acceleration time history

%% Synthesis of sine sweep

freq = SRS_max(:,1);
amp = SRS_max(:,2);

these = find(freq > max_freq);
freq(these(1):end)=[];
amp(these(1):end) = [];


f = min(freq):1:max(freq);
a = interp1(freq,amp,f);
t = log(f/min(freq))/log(2)*60/R;
time = 0:dt:max(t);
f= interp1(t,f,time);
a = interp1(t,a,time);
esi = a/Q;

acc = esi.*sin(2*pi*f.*time);
acc = transpose(acc);
time = transpose(time);

acc_time_history = [time acc];
```

Synthesis of a FSS from SRS data

```
%%Synthesis of a fast sine sweep from SRS data

% Marcos Feria Cerrada
% Delft 2019

% This program gives the acceleration time history of a fast sine sweep
% The resulting fast sine sweep satisfies a shock response spectrum
% specification
% Necessary input SRS_mnax
% The first column of SRS_max is the frequency points of the SRS
% The second column of SRS_max is the acceleration values of the SRS

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
clear all
load('matlab.mat');
SRS1 = SRS_max;

fig_num=1;
tpi=2.*pi;
th=zeros(K100,1);
vth=zeros(K100,1);

% Introduce SRS matrix
disp(' ');
disp(' Introduce SRS : an internal Matlab array');
disp(' ');
%
THM1=input(' Enter the array name:  ');
f1=THM1(:,1);
a1=THM1(:,2);
clear length;
n1=length(a1);
%
clear aspec1;
aspec1=a1;

ffirst1=f1(1);
flast1=f1(n1);
last_f1=f1(n1);
last_a1=a1(n1);
clear fr1;
clear r1;
fr1=f1;
r1=a1;

%%  Calculate slopes between input points to SRS
clear length;
num1=length(fr1);
%
s1=zeros(num1-1,1);
%
for i=1:(num1-1)
    a1=(log(r1(i+1))-log(r1(i)));
    b1=(log(fr1(i+1))-log(fr1(i)));
    s1(i)=a1/b1;
end

clear f1;
clear spec1;
clear a1
clear b1
%
```

```
%%  Interpolate for SRS

out1=sprintf('\n\n Enter octave spacing.\n 1= 1/3 2= 1/6 3= 1/12 \n');
disp(out1);
%
ioct=input(' ');
%
octave=(1./3.);

f1 = zeros(1,length(r1));
spec1 = zeros(1,length(r1));

f1(1)= fr1(1);
fb1=fr1(1);
spec1(1)=r1(1);

if ioct==2
    octave=(1./6.);
end

if ioct==3
    octave=(1./12.);
end

f1(1)=fr1(1);
spec1(1)=r1(1);
i=2;
%

while(1)
    ff1=(2.^octave)*fb1;
    fb1=ff1;
    if(ff1>fr1(num1))
        break;
    end
%
    if( ff1 >= fr1(1))
%
        for j=1:num1
%
            if(ff1 == fr1(j))
                        f1(i)=ff1;
                        spec1(i)=r1(j);
                        nspec1=i;
                        i=i+1;
                        break;
            end
            if(ff1 < fr1(j) && j>1)
%
                        f1(i)=ff1;
                        az1=(log10(r1(j-1)));
                        az1=az1+(s1(j-1)*(log10(ff1)-log10(fr1(j-1))));
                        spec1(i)=10.^az1;
                        nspec1=i;
                        i=i+1;
                        break;
            end
%
        end
    end
end
%
frlast1=max(fr1);
%
if(frlast1 > f1(nspec1))
          nspec1=nspec1+1;
             f1(nspec1)=fr1(num1);
          spec1(nspec1)=r1(num1);
```

```
end
%
if(nspec1 > NUM)
           out1=sprintf('\n Warning: number of specification points reduced. ');
           disp(out1);
           nspec1=NUM;
end

clear at
clear bt

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

out1=sprintf('\n Enter damping format for SRS. \n 1= damping ratio   2= Q \n');
disp(out1);
%
idamp=input(' ');
%
while(1)
%
    if(idamp == 1 )
            disp(' Enter damping ratio  (typically 0.05) ');
            damp1=input(' ');
            Q1 = (1./(2.*damp1));
    else
            disp(' Enter first SRS amplification factor Q (typically 10) ');
            Q1=input(' ');
            damp1 = (1./(2.*Q1));
    end
    if((damp1<1.0))
            break;
    end
%
end

amp_start1= spec1/10;
arlast1=amp_start1(nspec1);

while(1)
%
    out1=sprintf('\n\n Enter units ');
    out2=sprintf('    1=English:  G,       in/sec, in   ');
    out3=sprintf('    2=metric:   G,       m/sec,  mm   ');
    out4=sprintf('    3=metric:   m/sec^2, m/sec,  mm  \n');
%
    disp(out1);
    disp(out2);
    disp(out3);
    disp(out4);
%
    iunit=input(' ');
%
    if( iunit==1 || iunit==2 || iunit==3 )
            break;
    end
end
%
dunit='mm';
vunit='m/sec';
aunit='G';
%
if(iunit==1)
          dunit='inch';
          vunit='in/sec';
end
if(iunit==3)
          aunit='m/sec^2';
end
```

```
%

frmax=max(fr1);
%
sr=(10*max(fr1));
%
out1=sprintf(' Recommend sample rate >= %10.6g samples/sec',sr );
out2=sprintf('\n Enter sample rate ');
%
disp(out1);
disp(out2);
%
sr=input(' ');
%
if(sr < 4*frmax)
        out1=sprintf('\n Current sr=%8.4g   frmax=%8.4g Hz',sr,frmax);
        disp(out1);
        sr= 4*frmax;
        out1=sprintf('\n Warning: sample rate reset to %f ',sr);
        disp(out1);
end
%
dt=(1./sr);

out2=sprintf('set duration \n');
%
disp(out1);
disp(out2);
%
idur=input(' ');

out1=sprintf('\n\n Enter duration (sec) ');
out2=sprintf(' (Recommend %6.3f or greater) \n',2.0/f1(1));

disp(out1);
disp(out2);

dur=input(' ');

nt=round(dur/dt);

if(nt>K100)
        out1=printf('\n\n Warning: duration reduced. \n');
        disp(out1);
        nt=K100;
end

out1=sprintf('\n dt=%9.4g sec   dur=%8.4f sec  sr=%9.4g sample/sec  nt=%ld
\n',dt,dur,sr,nt);
disp(out1);
%
if(dur < 1.5/f1(1))
%
        dur=1.5/f1(1);
        out1=sprintf('\n\n Warning: duration is too short.\n\n Duration is reset to %f
',dur);
        disp(out1);
%
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
tic
disp(' ');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
R = (flast1-ffirst1)/(dur*log(2));

out1=sprintf('\n\n Enter maximum sweep rate [oct/sec] suggested around %f ',R);
disp(out1);
```

```
Rmax = input(' ');

%
if(nt < K100)

    omegaf=tpi*f1;
    nspec1=max(size(spec1));
    nspec2=max(size(spec2));
%
    out1=sprintf('\n\n Enter resolution of optimisation suggested around %f ',0.5);
    disp(out1);
    resolution = input(' ');

%   OPTIMIZATION METHOD BASED ON ERROR, AND FREQUENCY SECTIONS

    out1=sprintf('\n\n Enter number of sections for optimisation suggested around %f
',5);
    disp(out1);
    numberofsections = input(' ');

    section_length = round((nspec1/numberofsections)+0.5);
    numberofsections = round(nspec1/section_length+0.5);
    amp_start1 = spec1/round(sqrt(Q1^2+1));

    finish = 0; % When we finish optimising CURVE 1 then finish = 1
    local_amp_record = 100000000; %% To compare with the error
    optimise_amp_section = 1; % Section being optimized
    optimise_amp_value = round(sqrt(Q1^2+1)); % Amplitude for that section
    rend = Rmax*ones(1,nspec1);

    factor = resolution;
    amp_start1(1:section_length)=spec1(1:section_length)/factor;

    limit = round(sqrt(Q1^2+1));

    while (finish~=1)

       if (finish==0)

            amp_start = amp_start1;
            nspec = nspec1;
            spec = spec1;

       end

        points_time = zeros(1,length(f1));
        points_time(1)=0;
        E = zeros(1,length(f1));
        E(1) = 0;

        for mr=2:length(f1)
            deltaf = f1(mr)-f1(mr-1);
            points_time(mr)=points_time(mr-1)+(deltaf/(log(2)))*(1/rend(mr-1));
        end

        for mr=2:length(f1)
            deltat = points_time(mr)-points_time(mr-1);
            E(mr) = E(mr-1)+deltat*2*pi*f1(mr-1);
        end

        %Now we have points for amplitude and E in time but we need more
        %points, INTERPOLATE
        dur = max(points_time);
        dt = dur/(sr);
        nt=round(dur/dt);
        [a1,a2,b1,b2,b3]=srs_coefficients(f1,damp1,dt);

        am_start = zeros(1,sr);
```

```
        am_points = zeros(1,sr);
        def_time = zeros(1,sr);
        E_sin = zeros(1,sr);
        E_points = zeros(1,sr);
        %
        st=zeros(length(points_time)-1,1);
        %
        for i=1:(length(points_time)-1)
            at=(log(amp_start(i+1))-log(amp_start(i)));
            bt=(log(points_time(i+1))-log(points_time(i)));
            st(i)=at/bt;
        end
        %
        clear at
        clear bt
        %
        def_time(1)=points_time(1);
        t1_start=points_time(1);
        am_start(1)= amp_start(1);

        %
        i=2;
        %
        while(1)
            tt1=dt+t1_start;
            t1_start=tt1;
            if(tt1>max(points_time))
                break;
            end
        %
            if( tt1 >= points_time(1))
        %
                for j=1:length(points_time)
        %
                    if(tt1 == points_time(j))
                                def_time(i)=tt1;
                                am_start(i)=amp_start(j);
                                npoints=i;
                                i=i+1;
                                break;
                    end
                    if(tt1 < points_time(j) && j>1)
        %
                                def_time(i)=tt1;
                                az1=(log10(amp_start(j-1)));
                                az1=az1+(st(j-1)*(log10(tt1)-log10(points_time(j-1))));
                                am_points(i)=10.^az1;
                                npoints=i;
                                i=i+1;
                                break;
                    end
        %
                end
            end
        end
        %
        points_timelast1=max(points_time);
        %
        if(points_timelast1 > def_time(npoints))
                  npoints=npoints+1;
                  def_time(npoints)=points_time(length(points_time));
                  am_points(npoints)=amp_start(length(points_time));
        end


         am_points(isnan(am_points)) = 0;

% Now interpolate for E
```

```
         se=zeros(length(points_time)-1,1);
         %
         for i=1:(length(points_time)-1)
             ae=(log(E(i+1))-log(E(i)));
             be=(log(points_time(i+1))-log(points_time(i)));
             se(i)=ae/be;
         end
         %
         clear ae
         clear be
         %

         def_time(1)=points_time(1);
         t1_start=points_time(1);
         E_sin(1)= E(1);

         %
         i=2;
         %
         while(1)
             tt1=dt+t1_start;
             t1_start=tt1;
             if(tt1>max(points_time))
                 break;
             end
         %
             if( tt1 >= points_time(1))
         %
                 for j=1:length(points_time)
         %
                     if(tt1 == points_time(j))
                                 def_time(i)=tt1;
                                 E_sin(i)=E(j);
                                 npoints=i;
                                 i=i+1;
                                 break;
                     end
                     if(tt1 < points_time(j)  && j>1)
         %
                                 def_time(i)=tt1;
                                 az1=(log10(E(j-1)));
                                 az1=az1+(se(j-1)*(log10(tt1)-log10(points_time(j-1))));
                                 E_points(i)=10.^az1;
                                 npoints=i;
                                 i=i+1;
                                 break;
                     end
         %
                 end
             end
         end
         %
         points_timelast1=max(points_time);
         %
         if(points_timelast1 > def_time(npoints))
                   npoints=npoints+1;
                   def_time(npoints)=points_time(length(points_time));
                   E_points(npoints)=E(length(points_time));
         end
        these = find(isnan(E_points));
        E_points(these) = 0;

         th = am_points.*sin(E_points);
%
         [xxmax,xxmin,xmax,xmin]=ws_srs(nspec,th,a1,a2,b1,b2,b3,f1);
%
       %% Check if new record
```

```
      clear vector;
      vector = ((optimise_amp_section-1)* section_length+1):((optimise_amp_section-1)*
section_length+section_length);

       if (max(vector)>nspec)
           these = find(vector>nspec);
           vector(these) = [];
       end

      [error,local_error]=ws_srs_error(spec,xmax,xmin,vector,nspec);

      if(error<local_amp_record)
          local_amp_record = error;
          optimise_amp_value = factor;
      end

      factor = factor + resolution;

      amp_start1(vector) = spec(vector)/factor;

      if(factor > limit)
          if(finish == 0)

            amp_start1(vector) = spec(vector)/optimise_amp_value;
            optimise_amp_section = optimise_amp_section+1;
            factor = resolution;
            optimise_amp_value = round(sqrt(Q1^2+1));


            if(optimise_amp_section ==  numberofsections)
              finish = 1;
            end
          end

      end

    end
%

amp_start = amp_start1;

% Now we do a finer approximation, repeat calculation with finer resolution

if(finish == 1)

    out1=sprintf('\n\n Enter factor for optimisation of sweep rate  suggested around %f
',0.1);
    disp(out1);
    resolution = input(' ');
    finish = 0; % When we finish optimising CURVE 1 then finish = 1
    optimise_amp_section = 1; % Section being optimized
    optimise_amp_value = 1; % Amplitude value for that section

    factor = resolution;
    rend(1:section_length)=rend(1:section_length)*factor;

    limit = 1;

    while (finish~=1)

       if (finish==0)

            nspec = nspec1;
            spec = spec1;

       end

        points_time = zeros(1,length(f1));
```

```matlab
points_time(1)=0;
E = zeros(1,length(f1));
E(1) = 0;

for mr=2:length(f1)
    deltaf = f1(mr)-f1(mr-1);
    points_time(mr)=points_time(mr-1)+(deltaf/(log(2)))*(1/rend(mr-1));
end

for mr=2:length(f1)
    deltat = points_time(mr)-points_time(mr-1);
    E(mr) = E(mr-1)+deltat*2*pi*f1(mr-1);
end

%Now we have points for amplitude and E in time but we need more
%points, INTERPOLATE

dur = max(points_time);
dt = dur/(sr);
nt=round(dur/dt);
[a1,a2,b1,b2,b3]=srs_coefficients(f1,damp1,dt);

am_start = zeros(1,sr);
am_points = zeros(1,sr);
def_time = zeros(1,sr);
E_sin = zeros(1,sr);
E_points = zeros(1,sr);

%
st=zeros(length(points_time)-1,1);
%
for i=1:(length(points_time)-1)
    at=(log(amp_start(i+1))-log(amp_start(i)));
    bt=(log(points_time(i+1))-log(points_time(i)));
    st(i)=at/bt;
end
%
clear at
clear bt
%

def_time(1)=points_time(1);
t1_start=points_time(1);
am_start(1)= amp_start(1);

i=2;
%
while(1)
    tt1=dt+t1_start;
    t1_start=tt1;
    if(tt1>max(points_time))
        break;
    end
%
    if( tt1 >= points_time(1))
%
        for j=1:length(points_time)
%
            if(tt1 == points_time(j))
                        def_time(i)=tt1;
                        am_start(i)=amp_start(j);
                        npoints=i;
                        i=i+1;
                        break;
            end
            if(tt1 < points_time(j)  && j>1)
%
                        def_time(i)=tt1;
```

```
                              az1=(log10(amp_start(j-1)));
                              az1=az1+(st(j-1)*(log10(tt1)-log10(points_time(j-1))));
                              am_points(i)=10.^az1;
                              npoints=i;
                              i=i+1;
                              break;
                 end
%
              end
          end
      end
      %
      points_timelast1=max(points_time);
      %
      if(points_timelast1 > def_time(npoints))
              npoints=npoints+1;
              def_time(npoints)=points_time(length(points_time));
              am_points(npoints)=amp_start(length(points_time));
      end


       am_points(isnan(am_points)) = 0;

% Now interpolate for E

      %
      se=zeros(length(points_time)-1,1);
      %
      for i=1:(length(points_time)-1)
          ae=(log(E(i+1))-log(E(i)));
          be=(log(points_time(i+1))-log(points_time(i)));
          se(i)=ae/be;
      end
      %
      clear ae
      clear be
      %

      def_time(1)=points_time(1);
      t1_start=points_time(1);
      E_sin(1)= E(1);

      i=2;
      %
      while(1)
          tt1=dt+t1_start;
          t1_start=tt1;
          if(tt1>max(points_time))
              break;
          end
%
          if( tt1 >= points_time(1))
%
              for j=1:length(points_time)
%
                  if(tt1 == points_time(j))
                              def_time(i)=tt1;
                              E_sin(i)=E(j);
                              npoints=i;
                              i=i+1;
                              break;
                  end
                  if(tt1 < points_time(j) && j>1)
%
                              def_time(i)=tt1;
                              az1=(log10(E(j-1)));
                              az1=az1+(se(j-1)*(log10(tt1)-log10(points_time(j-1))));
                              E_points(i)=10.^az1;
```

```
                                npoints=i;
                                i=i+1;
                                break;
                    end
        %
                end
            end
        end
        %
        points_timelast1=max(points_time);
        %
        if(points_timelast1 > def_time(npoints))
                  npoints=npoints+1;
                  def_time(npoints)=points_time(length(points_time));
                  E_points(npoints)=E(length(points_time));
        end
       these = find(isnan(E_points));
       E_points(these) = 0;

        th = am_points.*sin(E_points);
        [xxmax,xxmin,xmax,xmin]=ws_srs(nspec,th,a1,a2,b1,b2,b3,f1);

      %% Check if new record
      clear vector;
      vector = ((optimise_amp_section-1)* section_length+1):((optimise_amp_section-1)*
section_length+section_length);

       if (max(vector)>nspec)
           these = find(vector>nspec);
           vector(these) = [];
       end

      [error,local_error]=ws_srs_error(spec,xmax,xmin,vector,nspec);

      if(error<local_amp_record)
          local_amp_record = error;
          optimise_amp_value = factor;
      end

      factor = factor + resolution;

      rend(vector) = rend(vector)*factor;

      if(factor > limit)
          if(finish == 0)

            rend(vector) = Rmax*optimise_amp_value;
            optimise_amp_section = optimise_amp_section+1;
            factor = resolution;
            optimise_amp_value = 1;


            if(optimise_amp_section ==  numberofsections)
              finish = 1;
            end
          end

      end

    end

%% ACTUALIZE LAST TIME
        points_time = zeros(1,length(f1));
        points_time(1)=0;
        E = zeros(1,length(f1));
        E(1) = 0;

        for mr=2:length(f1)
```

```
    deltaf = f1(mr)-f1(mr-1);
    points_time(mr)=points_time(mr-1)+(deltaf/(log(2)))*(1/rend(mr-1));
end

for mr=2:length(f1)
    deltat = points_time(mr)-points_time(mr-1);
    E(mr) = E(mr-1)+deltat*2*pi*f1(mr-1);
end

%Now we have points for amplitude and E in time but we need more
%points, INTERPOLATE

dur = max(points_time);
dt = dur/(sr);
[a1,a2,b1,b2,b3]=srs_coefficients(f1,damp1,dt);
nt=round(dur/dt);

am_start = zeros(1,sr);
am_points = zeros(1,sr);
def_time = zeros(1,sr);
E_sin = zeros(1,sr);
E_points = zeros(1,sr);

st=zeros(length(points_time)-1,1);
%
for i=1:(length(points_time)-1)
    at=(log(amp_start(i+1))-log(amp_start(i)));
    bt=(log(points_time(i+1))-log(points_time(i)));
    st(i)=at/bt;
end
%
clear at
clear bt
%
def_time(1)=points_time(1);
t1_start=points_time(1);
am_start(1)= amp_start(1);
%
i=2;
%
while(1)
    tt1=dt+t1_start;
    t1_start=tt1;
    if(tt1>max(points_time))
        break;
    end
%
    if( tt1 >= points_time(1))
%
        for j=1:length(points_time)
%
            if(tt1 == points_time(j))
                   def_time(i)=tt1;
                   am_start(i)=amp_start(j);
                   npoints=i;
                   i=i+1;
                   break;
            end
            if(tt1 < points_time(j) && j>1)
%
                   def_time(i)=tt1;
                   az1=(log10(amp_start(j-1)));
                   az1=az1+(st(j-1)*(log10(tt1)-log10(points_time(j-1))));
                   am_points(i)=10.^az1;
                   npoints=i;
                   i=i+1;
                   break;
            end
```

```
        %
                end
            end
        end
        %
        points_timelast1=max(points_time);
        %
        if(points_timelast1 > def_time(npoints))
                  npoints=npoints+1;
                  def_time(npoints)=points_time(length(points_time));
                  am_points(npoints)=amp_start(length(points_time));
        end


         am_points(isnan(am_points)) = 0;

% Now interpolate for E

        %
        se=zeros(length(points_time)-1,1);
        %
        for i=1:(length(points_time)-1)
            ae=(log(E(i+1))-log(E(i)));
            be=(log(points_time(i+1))-log(points_time(i)));
            se(i)=ae/be;
        end
        %
        clear ae
        clear be
        %


        def_time(1)=points_time(1);
        t1_start=points_time(1);
        E_sin(1)= E(1);

        %
        i=2;
        %
        while(1)
            tt1=dt+t1_start;
            t1_start=tt1;
            if(tt1>max(points_time))
                break;
            end
        %
            if( tt1 >= points_time(1))
        %
                for j=1:length(points_time)
        %
                    if(tt1 == points_time(j))
                          def_time(i)=tt1;
                          E_sin(i)=E(j);
                          npoints=i;
                          i=i+1;
                          break;
                    end
                    if(tt1 < points_time(j) && j>1)
        %
                          def_time(i)=tt1;
                          az1=(log10(E(j-1)));
                          az1=az1+(se(j-1)*(log10(tt1)-log10(points_time(j-1))));
                          E_points(i)=10.^az1;
                          npoints=i;
                          i=i+1;
                          break;
                    end
        %
```

```
                end
            end
        end
        %
        points_timelast1=max(points_time);
        %
        if(points_timelast1 > def_time(npoints))
                  npoints=npoints+1;
                  def_time(npoints)=points_time(length(points_time));
                  E_points(npoints)=E(length(points_time));
        end
       these = find(isnan(E_points));
       E_points(these) = 0;
%

        th = am_points.*sin(E_points);
%
        [xxmax,xxmin,xmax,xmin]=ws_srs(nspec,th,a1,a2,b1,b2,b3,f1);
%

      [error,local_error]=ws_srs_error(spec,xmax,xmin,vector,nspec);
%
time_history = transpose([def_time;th]);
fig_num = 1;
[Shock_Response_Spectrum,fig_num]=
fss_synth_srs_plot(f1,fr1,xmax,xmin,aspec1,damp1,iunit,fig_num);

end

else
%
    out1=sprintf('\n\n  Error: too many samples.\n\n');
    disp(out1);
%
end  %% end nt loop
%
disp(' ');
toc
```

Finite Element Model for AFECTOS

```
% Marcos Feria Cerrada
% Delft 2019

%% Model definition for Afectos
%% This model does not inclued the point masses yet

nnp_vertical     = 15; %% Minimum points = 2
nnp_transversal   = 5; %% Minimum points = 2
nnp = nnp_vertical+(nnp_transversal-1)*4;


nel_vertical      =  nnp_vertical-1;
nel_transversal = nnp_transversal-1;
nel = nel_vertical+4*nel_transversal;
nen      =  2;
ndf      =  6;


% Bars are squares, the size of the squares are:
bvertical = 0.2;
bhorizontal1 = 1.5*0.2;
bhorizontal2 =1.25*0.2;
bhorizontal3= 1*0.2;
bhorizontal4 = 1.25*0.2;

MP(1).element = {'beam3D'};
MP(1).properties =
[200e9,bvertical^2,(bvertical^4)/12,(bvertical^4)/12,7850,793e8,0.00022779,0,0,0,0,0,0]
;
MP(1).options = {''};

MP(2).element = {'beam3D'};
MP(2).properties =
[200e9,bhorizontal1^2,(bhorizontal1^4)/12,(bhorizontal1^4)/12,7850,793e8,0.0011532,0,0,
0,0,0,0];
MP(2).options = {''};

MP(3).element = {'beam3D'};
MP(3).properties =
[200e9,bhorizontal2^2,(bhorizontal2^4)/12,(bhorizontal2^4)/12,7850,793e8,
0.00055613,0,0,0,0,0,0];
MP(3).options = {''};

MP(4).element = {'beam3D'};
MP(4).properties =
[200e9,bhorizontal3^2,(bhorizontal3^4)/12,(bhorizontal3^4)/12,7850,793e8,0.00022779,0,0
,0,0,0,0];
MP(4).options = {''};

MP(5).element = {'beam3D'};
MP(5).properties =
[200e9,bhorizontal4^2,(bhorizontal4^4)/12,(bhorizontal4^4)/12,7850,793e8,0.00055613,0,0
,0,0,0,0];
MP(5).options = {''};

% Geometry definition
l_vertical     = 2;
l_transversal= 0.6;

x_vertical = linspace(0,0,nnp_vertical);
x_transversal1 = linspace(0,l_transversal,nnp_transversal);
x_transversal1(1) = [];
x_transversal2 = linspace(0,0,nnp_transversal);
x_transversal2(1) = [];
x_transversal3 = x_transversal1*-1;
```

```
x_transversal4 = linspace(0,0,nnp_transversal);
x_transversal4(1) = [];

y_vertical = linspace(0,0,nnp_vertical);
y_transversal1 =linspace(0,0,nnp_transversal);
y_transversal1(1) = [];
y_transversal2 = linspace(0,l_transversal,nnp_transversal);
y_transversal2(1) = [];
y_transversal3 = linspace(0,0,nnp_transversal);
y_transversal3(1) = [];
y_transversal4 = -1*y_transversal2;


z_vertical = linspace(0,l_vertical,nnp_vertical);
z_transversal1 = linspace(l_vertical,l_vertical,nnp_transversal);
z_transversal1(1) = [];
z_transversal2 = linspace(l_vertical,l_vertical,nnp_transversal);
z_transversal2(1) = [];
z_transversal3 = linspace(l_vertical,l_vertical,nnp_transversal);
z_transversal3(1) = [];
z_transversal4 = linspace(l_vertical,l_vertical,nnp_transversal);
z_transversal4(1) = [];


X       = [[x_vertical x_transversal1 x_transversal2 x_transversal3 x_transversal4];
[y_vertical y_transversal1 y_transversal2 y_transversal3 y_transversal4]; [z_vertical
z_transversal1 z_transversal2 z_transversal3 z_transversal4]];
X = transpose(X);

% Code to check the mesh of the model:

%  plot3(X(:,1),X(:,2),X(:,3),'Marker','o');

% for k=1:nnp
%      text(X(:,1),X(:,2),X(:,3),num2str(k),'FontSize',10,'Rotation',+45);
% end


% Element definition

for e=1:nel_vertical
  EN(e,:) = [e e+1 1];
end

% Element definition for transversal beams
for transversal=1:4
    newstartpoint = nnp_vertical+(transversal-1)*nel_transversal+1;
    EN(newstartpoint-1,:)=[nnp_vertical newstartpoint transversal+1];
    for e=newstartpoint:newstartpoint+nel_transversal-2
        EN(e,:) = [e e+1 transversal+1];
    end
end


% Boundary conditions
FG.type={'displacement'};

FG.point=[1];

FG.value=[0,0,0,0,0,0];
FG.conditions=[1,1,1,1,1,1];

body = struct('x',X,'EN',EN,'MP',MP);

actions = struct('FG',FG);

options = struct( ...
```

```
                  'verbosity',10,                                         ... [integer
[0-10], {5}]
                  'gravity','off',                                        ... [ 'on' |
'off' ]
                  'NR',[1e-4 20],                                         ... [ [tol
n_iter] ]
                  'solver','non_linear',                                  ...
                  'Key',{{'test' 'beam_satellite' 'bidi_post_process'}},      ...
string
                  'analysis','structural_static',                                   ...
[ static | dynamic ]
                  'newmark',[0 1 0 0],                                    ... [ dt nts
beta gamma ]
                  'post_options',{{'none' 'epsc'}},                       ...
                  'qint','');
```

Response of a mdof system to an base excitation with Newmark solver

```
% Marcos Feria Cerrada
% Delft 2019

% This program gives the dynamic response of a mdof system subjected to a
% base excitation in a isngle axis (x, y or z)

% NECESSARY TO HAVE AFECTOS FILES IN MATLAB FOLDER

% Direction of the excitaiton defined with dir in parameters
% Used method is Newmark Beta method with linear variation of acceleration
% in each time step

%Excitation is a matrix with first column indicating time and second
%acceleration

% Acceleration time history already loaded in matlab, t = time and acc =
% acceleration

% We suppose the problem is linear: m,k,c matrixes dont change

%% Parameters

dir =2; %1 = x  2 = y  3 = z

% Newmark parameters
% Average acceleration method gama = 1/2 and beta = 1/4
% Linear acceleration method gama = 1/2 and beta = 1/6

gama = 0.5;
beta = 0.25;

seta = 0.05; % Damping


%% Introduction of acceleration

excitation = transpose([t; acc]);

acc = excitation(:,2);
time = excitation(:,1);
vel = cumtrapz(time, acc);    % Ground velocity
posi = cumtrapz(time, vel);  % Ground displacement

plot(time,acc,'k');
xlabel('Time [s]');
ylabel('Acceleration [m/s^2]')
if dir == 1
    title('Acceleration time history (X direction)');
elseif dir == 2
    title('Acceleration time history (Y direction)');
elseif dir == 3
     title('Acceleration time history (Z direction)');
end




%% Newmark Method

dt = excitation(2,1)-excitation(1,1);
meshinfo=compute_meshinfo(body, actions, options); %% AFECTOS folder
eq_coeff=getFEM(body,actions,options,meshinfo); %% AFECTOS folder
```

```
m=full(eq_coeff{1});
c = full(eq_coeff{2});
k=full(eq_coeff{3});
n = length(m);

% Addition of the point masses to the model

extreme_point_number = zeros(1,4);
m_puntual(1) = 1.75*50;
m_puntual(2) = 1.5*50;
m_puntual(3) = 0.75*50;
m_puntual(4) = 1*50;

for i=1:4
    extreme_point_number(i) = nel_vertical+ nel_transversal*i;
    additional_matrix = [m_puntual(i) 0 0 ;0 m_puntual(i) 0 ;0 0 m_puntual(i) ];
    m(((extreme_point_number(i)-1)*6+1):1:((extreme_point_number(i)-
1)*6+3),((extreme_point_number(i)-1)*6+1):1:((extreme_point_number(i)-1)*6+3)) =
m(((extreme_point_number(i)-1)*6+1):1:((extreme_point_number(i)-
1)*6+3),((extreme_point_number(i)-1)*6+1):1:((extreme_point_number(i)-1)*6+3)) +
additional_matrix;
end

% Determine damping matrix (Rayleigh damping)

% damping

primer_modo =13.4; % First mode of the mdof system
segundo_modo = 106; % Second mode of the mdof system

w1 = 2*pi*primer_modo;
w2 = 2*pi*segundo_modo;

matrix = 0.5.*[1/w1 w1;1/w2 w2];
a = (matrix^-1)*[seta;seta];

c = a(1)*m+a(2)*k; %

% Compute influence coefficient matrix

r = zeros(n,1); % Influence coefficient matrix
% This matrix has ones in the dof in the direction of the excitation

if dir==1
r(1:6:n)=1;
elseif dir==2
r(2:6:n) =1;
elseif dir==3
r(3:6:n)=1;
end


xppgi = transpose(excitation(:,2)); %Acceleration at the base
xppi = zeros(n,length(xppgi)); % Relative acceleration at the base
xppi(:,1) = 0;
xpi = zeros(n,length(xppgi)); % Relative velocity at the base
xpi(:,1) = 0;
xi = zeros(n,length(xppgi)); % Relative displacement at the base
xi(:,1) = 0;

keff = (1/((dt^2)*beta)).*m+(gama/(dt*beta)).*c+k;
xppi(:,1) = -r*xppgi(1); %% Initial acceleration of the floor is not zero

% Newmark coefficients
coef1 = (1/(beta*dt))*m+(gama/beta)*c;
coef2 = (1/(2*beta))*m+dt*((gama/(2*beta))-1)*c;
```

```
for i = 2:length(xppgi)

    deltaxppgi = xppgi(i)-xppgi(i-1);
    peff = -m*r*((deltaxppgi))+coef1*xpi(:,i-1)+coef2*xppi(:,i-1); %% Peff es en i+1
    deltaxi = keff\peff;
    deltaxpi = (gama/(beta*dt))*deltaxi-(gama/beta)*xpi(:,i-1)+dt*(1-
gama/(2*beta))*xppi(:,i-1);
    xi(:,i)=xi(:,i-1)+deltaxi;
    xpi(:,i) = xpi(:,i-1)+deltaxpi;
    deltaxppi = (1/(beta*(dt^2)))*deltaxi-(1/(beta*dt))*xpi(:,i-1)-
(1/(2*beta))*xppi(:,i-1);
    xppi(:,i) = xppi(:,i-1)+deltaxppi;

end

u = xi; % Relative displacement of every dof
v = xpi; % Relative velocity of every dof
```

Response of a mdof system to an harmonic excitation with modal analysis

```
%% Response of a mdof system to an harmonic excitation with modal analysis

% Marcos Feria Cerrada
% Delft 2019
% This program calculates the displacements of a MDOF system when subjected
% to an harmonic excitation

clear all

% Necessary input:
% m = mass matrix ; k = stiffness matrix; c = damping matrix
% SRS  matrix
% The first column of SRS is the frequency points of the SRS
% The second column of SRS is the acceleration values of the SRS

load('k.mat');
load('c.mat');
load('m.mat');

%% Parameters

dir = 1; % Direction of the excitation 1 = x, 2 = y, 3 = z

% Parameters of the harmonic signal
alpha = 35; % frequency of signal rad/s
Q = 10; % Amplitude of signal G

% Parameters of proportional damping
a1 = 7.4746;
a2 = 0.0001;

%% Harmonic excitation

t = transpose(0:0.005:1);
acceleration = Q*sin(alpha*t);

%% Modal analysis

landa = eig(k,m); % Eigenvalues
[U,landa_matrix] = eig(k,m); % U is the modal matrix
% containing the modes already normalized so that UtMU = I

natural_freq = sqrt(landa); % rad/s

dof = length(k);
time_points = length(t);

% Compute influence coefficient matrix
R = zeros(dof,1); % Influence coefficient matrix
% since only 1d it is a vector with ones in the excitation direction

if dir==1
R(1:6:dof)=1;
elseif dir==2
R(2:6:dof) =1;
elseif dir==3
R(3:6:dof)=1;
end

q = zeros(dof,time_points);

for time = 1:time_points

for r = 1:dof
```

```
   setar = (a1+a2*(natural_freq(r)^2))/(2*natural_freq(r));
    q(:,time) = q(:,time)+((transpose(U(:,r))*(-        m*Q*R))/(natural_freq(r)^2-
    alpha^2+1i*2*setar*natural_freq(r)*alpha))*U(:,r);
end
   q(:,time) = q(:,time)*exp(1i*alpha*t(time));
   q(:,time) = imag(q(:,time));
end


% q, gives the displacements for all the degrees of freedom of the system
```

Calculation of the reaction forces at the base from known displacements

```
% Marcos Feria Cerrada
% Delft 2019

% This program calculates the forces at the base of a mdof system subjected to a
% base excitation

% The program also gives maximum forces and number of cycles

% The displacements, velocities and acceleration have already been solved with
% Newmark solver or Modal analysis

% So necessary input is xi, xpi, xppi and dir (direction of excitation: 1, 2 or 3)

% NECESSARY TO HAVE AFECTOS FILES IN MATLAB FOLDER



FG.type={'displacement'};
FG.point=[nnp];
FG.value=[0,0,0,0,0,0];
FG.conditions=[1,1,1,1,1,1];
body = struct('x',X,'EN',EN,'MP',MP);
actions = struct('FG',FG);

meshinfo=compute_meshinfo(body, actions, options);
eq_coeff=getFEM(body,actions,options,meshinfo);

m=full(eq_coeff{1});
c = full(eq_coeff{2});
k=full(eq_coeff{3});
n = length(m);

extreme_point_number = zeros(1,4);
m_puntual(1) = 1.75*50;
m_puntual(2) = 1.5*50;
m_puntual(3) = 0.75*50;
m_puntual(4) = 1*50;

for i=1:4
    extreme_point_number(i) = nel_vertical+ nel_transversal*i;
    additional_matrix = [m_puntual(i) 0 0 ;0 m_puntual(i) 0 ;0 0 m_puntual(i) ];
    m(((extreme_point_number(i)-1)*6+1):1:((extreme_point_number(i)-
1)*6+3),((extreme_point_number(i)-1)*6+1):1:((extreme_point_number(i)-1)*6+3)) =
m(((extreme_point_number(i)-1)*6+1):1:((extreme_point_number(i)-
1)*6+3),((extreme_point_number(i)-1)*6+1):1:((extreme_point_number(i)-1)*6+3)) +
additional_matrix;
end

w1 = 2*pi*primer_modo;
w2 = 2*pi*segundo_modo;

matrix = 0.5.*[1/w1 w1;1/w2 w2];
a = (matrix^-1)*[seta;seta];

c = a(1)*m+a(2)*k; %

Fx = zeros(length(xppgi),1);
Fy = zeros(length(xppgi),1);
Fz = zeros(length(xppgi),1);

isx = 0;
isy = 0;
isz = 0;
```

```
if dir==1
    isx = 1;
elseif dir == 2
    isy = 1;
elseif dir ==3
    isz = 1;
end

for i = 1:length(xppgi)-1
    fuerza_suelo = m(1,:)*r*(xppgi(i));
    Fx(i) = -m(1,:)*xppi(:,i)-c(1,:)*xpi(:,i)-k(1,:)*xi(:,i)- fuerza_suelo*isx;
    Fy(i) = -m(2,:)*xppi(:,i)-c(2,:)*xpi(:,i)-k(2,:)*xi(:,i)- fuerza_suelo*isy;
    Fz(i) = -m(3,:)*xppi(:,i)-c(3,:)*xpi(:,i)-k(3,:)*xi(:,i)- fuerza_suelo*isz;

end

Fx(end) = Fx(end-1);
Fy(end) = Fy(end-1);
Fz(end) = Fz(end-1);

figure
plot(excitation(:,1),Fx);
title('Reaction in X direction at the base');
xlabel('Time [s]');
ylabel('Force [N]');

figure
plot(excitation(:,1),Fy,'r');
title('Reaction in Y direction at the base');
xlabel('Time [s]');
ylabel('Force [N]');

figure
plot(excitation(:,1),Fz,'k');
title('Reaction in Z direction at the base');
xlabel('Time [s]');
ylabel('Force [N]');



Fmax_x = max(Fx)
tmax_x = time(find(Fx == max(Fx)))
Fmax_y = max(Fy)
tmax_y = time(find(Fy == max(Fy)))
Fmax_z = max(Fz)
tmax_z = time(find(Fz == max(Fz)))


Fmin_x = min(Fx)
tmin_x = time(find(Fx == min(Fx)))
Fmin_y = min(Fy)
tmin_y = time(find(Fy == min(Fy)))
Fmin_z = min(Fz)
tmin_z = time(find(Fz == min(Fz)))

mean_x = mean(Fx)
mean_y = mean(Fy)
mean_z = mean(Fz)

uextreme = zeros(length(xppgi),4);

cycles_base_x = length(findpeaks(Fx))
cycles_base_y = length(findpeaks(Fy))
cycles_base_z = length(findpeaks(Fz))
```

Calculation of acceleration of point masses

```
% Marcos Feria Cerrada
% Delft 2019

% This program calculates the acceleration at the point masses of the model

% The program also gives maximum accelerations and number of cycles

% The displacements, velocities and acceleration have already been solved with
% Newmark solver or Modal analysis

% So necessary input is xi, xpi, xppi and dir (direction of excitation: 1, 2 or 3)

% Output: acceleration of mass i is acc_extreme(:,i)


% NECESSARY TO HAVE AFECTOS FILES IN MATLAB FOLDER

for j = 1:3

for i=1:4
    extreme_point_number(i) = nel_vertical+ nel_transversal*i;
    if j == dir
        uextreme(:,i) = transpose(u(((extreme_point_number(i)-1)*6+j),:))+posi;
    else
        uextreme(:,i) = transpose(u(((extreme_point_number(i)-1)*6+j),:));
    end
end

vel_extreme = zeros(length(xppgi)-1,4);
acc_extreme = zeros(length(xppgi)-2,4);

for i=1:4
   vel_extreme(:,i) = diff(uextreme(:,i))./diff(time);
   acc_extreme(:,i)=diff(vel_extreme(:,i))./diff(time(2:end));
end

%% Number of cycles
if j == 1
    cycles_x_1 = length(findpeaks(acc_extreme(:,1)))
    cycles_x_2 = length(findpeaks(acc_extreme(:,2)))
    cycles_x_3 = length(findpeaks(acc_extreme(:,3)))
    cycles_x_4 = length(findpeaks(acc_extreme(:,4)))
    maximum_acceleration_x = max([max(acc_extreme(:,1)) max(acc_extreme(:,2))
max(acc_extreme(:,3)) max(acc_extreme(:,4))]);
    if (maximum_acceleration_x == max(acc_extreme(:,1)))
        mass1_x = 1
    elseif maximum_acceleration_x == max(acc_extreme(:,2))
        mass2_x = 1
    elseif maximum_acceleration_x == max(acc_extreme(:,3))
        mass3_x = 1
    elseif maximum_acceleration_x == max(acc_extreme(:,4))
        mass4_x = 1
    end

elseif j == 2
    cycles_y_1 = length(findpeaks(acc_extreme(:,1)))
    cycles_y_2 = length(findpeaks(acc_extreme(:,2)))
    cycles_y_3 = length(findpeaks(acc_extreme(:,3)))
    cycles_y_4 = length(findpeaks(acc_extreme(:,4)))

     maximum_acceleration_y = max([max(acc_extreme(:,1)) max(acc_extreme(:,2))
max(acc_extreme(:,3)) max(acc_extreme(:,4))]);
    if (maximum_acceleration_y == max(acc_extreme(:,1)))
        mass1_y = 1
```

```
    elseif maximum_acceleration_y == max(acc_extreme(:,2))
        mass2_y = 1
    elseif maximum_acceleration_y == max(acc_extreme(:,3))
        mass3_y = 1
    elseif maximum_acceleration_y == max(acc_extreme(:,4))
        mass4_y = 1
    end
elseif j == 3
    cycles_z_1 = length(findpeaks(acc_extreme(:,1)))
    cycles_z_2 = length(findpeaks(acc_extreme(:,2)))
    cycles_z_3 = length(findpeaks(acc_extreme(:,3)))
    cycles_z_4 = length(findpeaks(acc_extreme(:,4)))

     maximum_acceleration_z = max([max(acc_extreme(:,1)) max(acc_extreme(:,2))
max(acc_extreme(:,3)) max(acc_extreme(:,4))]);
    if (maximum_acceleration_z == max(acc_extreme(:,1)))
        mass1_z = 1
    elseif maximum_acceleration_z == max(acc_extreme(:,2))
        mass2_z = 1
    elseif maximum_acceleration_z == max(acc_extreme(:,3))
        mass3_z = 1
    elseif maximum_acceleration_z == max(acc_extreme(:,4))
        mass4_z = 1
    end

end

end

maximum_acceleration_x
maximum_acceleration_y
maximum_acceleration_z
```

Additional tools

Random acceleration time history

```
% Marcos Feria Cerrada
% Delft 2019

% This program gives a random acceleration time history
% Output: matrix = [time acceleration]

%% Parameters

duration = 0.02;
time_step = 0.0001; %

%%
time = 0:time_step:duration;
time =transpose(time);
f = 1/(22*10^-3);
w = 2*pi*f;
acceleration = zeros(length(time),1);
for i=1:length(acceleration)
acceleration(i)  = 50*rand*sin((w*time(i)));
end
acceleration((0.011/0.0001):end)=0;

figure
plot(time,acceleration);
xlabel('Time [sec]');
ylabel('Acceleration G');
title('Random input pulse acceleration');
matrix = [time acceleration];
```

Decomposition of a signal using the wavelet transform

```
%% Decomposition of a signal using discrete wavelet transform

% Marcos Feria Cerrada
% Delft 2019

% This program decomposes a signal in a sum of Morlet wavelets

% Necessary input:
% signal time history

%% Parameters

rel_octavas = 1/12; % Octave discretization of frequency
f_max = 150; % in hz
f_min = 5;

%% Definition of signal time history

time = 0:1/1000:1-1/1000;
acc = zeros(1,length(time));

for i=1:length(time)
    if time(i)<0.3
        acc(i) = 10*sin(2*pi*10*time(i))*sin(2*pi*55*time(i));
    elseif time(i)<0.6
        acc(i) = 15*sin(2*pi*20*time(i))*sin(2*pi*10*time(i));
    else
        acc(i) = 5*sin(2*pi*30*time(i))*sin(2*pi*50*time(i));
    end

end

% Fourier transform to check frequencies of signal

figure
y = fft(acc);

f = (0:length(y)-1)*1000/length(y);

plot(f,abs(y))
title(' Amplitude Spectrum of X(t) from t = 0.6 to t = 1 sec')
xlabel('f (Hz)')
xlim([0 100])
xticks(0:10:150);
grid
% Discretization of frequency and time for wavelet definition

tmax = time(end);
dt = time(2)-time(1);
N = 1/dt;

kmax = round(log(N+1)/log(2)*(1/rel_octavas));
kmin = round(kmax-log(1/(f_max*2*dt))/log(2)*(1/rel_octavas));
kminmin = round(kmax-log(1/(2*f_min*dt))/log(2)*(1/rel_octavas));

omega = pi;
sigma = 1;

k = kminmin:1:kmin;

dim_n = zeros(1,length(k));

for i = 1:length(k)
    dim_n(i) = round(tmax/(dt*2^(rel_octavas*(kmax-k(i)))));
```

```
end

w=pi./(dt*2.^((kmax-k)*rel_octavas)); % Frecuencies

a = omega./w; % Scale

tkn = zeros(length(k),max(dim_n));% Time

for i = 1:length(k) % All k
    for j = 1:dim_n(i) % All n

        tkn(i,j) = j*dt*2^((kmax-k(i))*rel_octavas);

    end
end

% Evaluation of matrix with wavelet functions

phikn = zeros(length(k),max(dim_n),length(time));

for i = 1:length(k) % Todos los k

    for ii = 1:dim_n(i) % todos los n
        for iii = 1:length(time)
            phikn(i,ii,iii) = real(exp(1i*omega*((time(iii)-
tkn(i,ii))/a(i))).*exp(-0.5*(((time(iii)-tkn(i,ii))/(a(i)*sigma)).^2)));

        end

    end
end

% Wavelet coeficients

ckn= zeros(length(k),max(dim_n));

for i = 1:length(k) % Todos los k

    for ii = 1:dim_n(i) % todos los n

        suma = 0;
        for iii = 1:length(time)
            suma = suma + acc(iii)*phikn(i,ii,iii);
        end

        ckn(i,ii) = suma*dt/(sqrt(a(i)));
    end
end

% Plot wavelet coefficients

surf_matrix = zeros(length(k)*max(dim_n),3);
i = 1;

while i<length(k)*max(dim_n)
    for ii=1:length(k)
        for iii = 1:dim_n(ii)

            surf_matrix(i,2) = tkn(ii,iii);
            surf_matrix(i,1) = w(ii)/(2*pi);
            if tkn(ii,iii)<0.3
            surf_matrix(i,3) = ckn(ii,iii)*1.5; % Factors to visualize better the
surface
            elseif tkn(ii,iii)<0.6
            surf_matrix(i,3) = ckn(ii,iii)*0.7;
            else
            surf_matrix(i,3) = ckn(ii,iii)*2;
            end
```

```
            i = i+1;
        end
        i = i+max(dim_n)-iii;
    end
end

[xq,yq] = meshgrid(f_min:0.05:100, 0:0.001:1);
vq = griddata(surf_matrix(:,1),surf_matrix(:,2),abs(surf_matrix(:,3)),xq,yq);

vq =2*medfilt1(vq,25);
mesh(xq,yq,vq);
 axis([0 100 0 1 0 8])
 view(0,90)

% Inverse wavelet transform

D = zeros(length(k),length(time));
deltat = tkn(1,2)-tkn(1,1);
for i = 1:length(k)

    for ii = 1:length(time)
        suma = 0;
        for iii = 1:dim_n(i)
            suma = suma + ckn(i,iii)*phikn(i,iii,ii);
        end
        D(i,ii) =deltat*suma/(a(i)^(2.5));
    end
    if i<length(k)
        deltat = tkn(i+1,2)-tkn(i+1,1);
    end

end

recomposicion = zeros(1,length(time));
for i = 1:length(time)
    suma = 0;
    deltaa = a(1);
    for j = 1:length(a)
        suma = suma + D(j,i)*deltaa;
        if j<length(a)
            deltaa = a(j+1)-a(j);
        end
    end
    recomposicion(i) = -1*suma;
end
```

Generation of synthetic signals using the wavelet analysis

This code is a continuation of previous code. Wavelet functions and wavelet coefficients are already calculated.

```
%% Generation of synthetic signals using wavelet analysis

% Marcos Feria Cerrada
% Delft 2019

% This program is a continuation of "Decomposition of a signal using discrete
wavelet transform"

% Necessary input:
% signal time history
% Wavelet coefficients and wavelet functions already calculated


% Calculation of modulating functions by normalizing wavelet coefficients

mkn= zeros(length(k),max(dim_n));
sumatoriockn = sum(sum(ckn.^2));

for i = 1:length(k) % Todos los k


    for ii = 1:dim_n(i) % todos los n
            mkn(i,ii) = abs(ckn(i,ii))/(sqrt(sumatoriockn));
    end
end

% New wavelet coefficients selected in a random process


cknuevo= zeros(length(k),max(dim_n));
for i = 1:length(k) % Todos los k

    deltatk = pi*2^((kmax-(k(i)))/12);
    for ii = 1:dim_n(i) % todos los n

        sign = -1+2*rand(1,1);
        while(sign == 0)
             sign = -1+2*rand(1,1);
        end

        if sign<0
            sign = -1;
        else
            sign = 1;
        end

        if i<length(k)/3
            cknuevo(i,ii)=sign*mkn(i,ii)*s(i)/27*50;
        elseif i<length(k)*2/3
            cknuevo(i,ii)=sign*mkn(i,ii)*s(i)/27*50;
        else
            cknuevo(i,ii)=sign*mkn(i,ii)*s(i)/27*50;

        end

    end
end

% Inverse wavelet transform

D = zeros(length(k),length(time));
deltat = tkn(1,2)-tkn(1,1);
for i = 1:length(k)
```

```
    for ii = 1:length(time)
        suma = 0;
        for iii = 1:dim_n(i)
            suma = suma + cknuevo(i,iii)*phikn(i,iii,ii);
        end
        D(i,ii) =deltat*suma/(a(i)^(2.5));
    end
    if i<length(k)
        deltat = tkn(i+1,2)-tkn(i+1,1);
    end

end

recomposicion = zeros(1,length(time));
for i = 1:length(time)
    suma = 0;
    deltaa = a(1);
    for j = 1:length(a)
        suma = suma + D(j,i)*deltaa;
        if j<length(a)
            deltaa = a(j+1)-a(j);
        end
    end
    recomposicion(i) = -1*suma;
end
transient(:,1) = time;
transient(:,2) = recomposicion;
```