Simulation of Pedestrian-induced vibrations of a Footbridge (2D surface)

Report of Simulation Pedestrian-induced vibrations of a Footbridge (2D surface)

By,
Anuraj Bose,
B.Tech, Dept. Of Civil Engineering,
4th Year: 26301319063

June'2022

Department of Civil Engineering and Computer Science and Engineering

Regent Education and Research Foundation, Barrackpore

This is to certify that the dissertation entitled "**SIMULATION OF PEDESTRIAN INDUCED VIBRATIONS OF A FOOTBRIDGE (2D SURFACE)**" is a project of in Final Year completely done by **Anuraj Bose [26301319063]** under our guidance and supervision and this is the project is being submitted of the Regent Education and Research Foundation the partial fulfilment of the requirement of the degree Bachelor of Technology in Civil Engineering.

18.06.22

Mr. Yuvaraj Mondal / Mrs. Labani Nandi Paul
(HO.D / A H.O.D department of civil engineering)
Regent Education and Research Foundation

18/06/2022

Mr. Souvik Sarkar
Assistant Professor, Dept of C.E.
R.E.R.F

for Saikat Mondal

Mr. Kaustav Sarkar
Assistant Professor, Dept of C.S.E
R.E.R.F

Acknowledgements

This Project has been without doubt the most challenging and rewarding activity I have undertaken as an engineer since the day I started doing programming...so far. However, it has not been a solo effort; its successful completion was only possible with the support and help of many others. I would like to express my special thanks and gratitude to Sri Souvik Sarkar, Dept. Of Civil Engineering, Regent Education Research Foundation, and Mr Subhadeep Mondal Sir (Project Guide and Mentor) for Supporting and giving me opportunity in this project for Civil Engineering Part section and Mr Saikat Mondal, Regent Education and Research Foundation, Dept. Of Computer Science and Engineering for their willingness to provide freedom, time and discussion has allowed to research and develop this body of work to its current state. Their approach to building my project has been exemplary.

A special thanks to Sri Kaustav Sarkar Sir without his help in providing primary motivation and guiding the Computer Science and Coding and Scripting portion not only for this project, but building basic blocks what is coding, how coding can solve object or purpose oriented programming, critical thinking, problem solving ability in Data Structure and Algorithm works in programming how to choose or when to choose which type of Algorithm or rather growing interest in Computer Science would not be possible. Without him, I may never have found myself immersed in the fascinating world of Critical thinking of Solution Oriented Programming Skills.

CERTIFICATE

It is certified that the work contained in this report titled "**SIMULATION OF PEDESTRIAN INDUCED VIBRATIONS OF A FOOTBRIDGE (2D SURFACE)** " is the original work done by ANURAJ BOSE [26301319063] and has been carried out under our supervision of Mr. Souvik Sarkar and Mr. Kaustav Sarkar.

To get this project online : Click here or https://github.com/anurajbose/Simulation-of-Padestrian-induced-vibrations-of-a-Footbridge-2D-surface-By-using-Python.git

Abstract

Vibration induced by walking pedestrians has motivated my project in the Civil Engineering Concept of mutated with Computer Programming and building blocks with problem solving methodology to a sudden community in-between these years. An area within this broad field that has received particular attention is the dynamic interaction that can occur between pedestrians and laterally flexible bridge structures. Perhaps the most notable example occurring on the opening day of London's Millennium Bridge. The enduring interest in this project problem is fuelled by two of its key features; (i) To address the damping and the structural slope and deflection observed by a statically determined single-degree-of-freedom structure (ii) the spatial and temporal variation in flow characteristics exhibited by a pedestrian crowd and visualize them in todays days of technology. This has tried to both of these features are addressed herein.

In this project this has first trying to make a combination of Object Oriented Programming and base of Civil Engineering concept of a single-degree-of-freedom structure. In this structure it has worked with an aim of identifying the interaction mechanism by which pedestrians produce force harmonics, that reorients with the structure on which they walk. These self-excited forces have been experimentally identified by others but the underlying reason for their existence has remained an open quest for me. It is taken that this project is an initially a stage where Civil Engineering concepts and methodology can be expand throw the notice of compacted Stricture of Programming Language as an Object Oriented Programming.

By the phenomena of this project, for data and visualisation Python version 3.8 has used worked in an about the Duhamel Integral flow and how it can be used to simulate the dynamic response of a single degree of freedom system structure. Thus it tried to discuss how to solve the integral and then write scripting to implement the solution for any steady loading. In this project Duhamel Integral solver has been used to build a crowd loading simulation. This will allow to simulate the vibration response of a footbridge to pedestrian traffic.

Contents

1. Introduction

The problem of excessive lateral bridge vibration induced by pedestrian traffic has received much attention from the engineering community for more than a decade. Interest was sparked in 1999 by the closure of the Solferino Bridge in Paris [26]. However it was the closure of London's Millennium Bridge the following year [2] that brought the issue to the attention of the wider public. Much of the guidance available to engineers regarding human-induced bridge vibration and the lateral instability phenomenon that can occur has resulted from the investigation of these two loading events. This has resulted in restrictive design criteria for pedestrian numbers and acceleration limits. The financial cost of this potentially conservative design philosophy is difficult to asses, but a better understanding of the problem is almost certain to result in economies in design.

Duhamel integral in some special load cases to single degree of freedom (SDOF) mass system. Here, this integral is treated as a special form of the Laplace integral transformation. Only single degree of freedom response is investigated. The relations are given in a time domain. First, the theoretical background of the response of undamped and damped single degree of freedom models, subjected to general load, is reviewed in brief. Validations in some special dynamic load cases are also demonstrated and discussed. In details, some positive general load cases are investigated and the analytical results, if exist, are compared with numerical solutions, obtained by MathCAD/MatLab. The present research can be extended to the response of undamped and damped multi-degree of freedom (MDOF) models. Also a transformation in the frequency domain can be obtained by the methods of Dynamic of MDOF systems, but it is not included in the work. Examples by numerical integration are also given. Finally some basic application remarks are proposed.

To understand the Duhamel Integral laying to build a Stabilising Ground equation of a Statically Determined Structure to Built and illustrate the problems of pedestrian to walk and their loads deflects to the structure .

1.1 The Project challenge

From a civil engineering perspective, a walking pedestrian constitutes one of the most complex sources of dynamic excitation a structure is likely to experience. If the structure on which a pedestrian walks is excited laterally to a level perceived by the pedestrian, a complex interaction may develop between the walking pedestrian and flexible structure. The perceived motion may induce alterations in the pedestrian's gait with a knock-on influence on the footfall forces imposed on the structure. This can in turn generate larger amplitude structural oscillations. A feedback loop is thereby established between structural response and pedestrian balance behaviour. This can be considered a coupling between two dynamic systems, one of which is controlled by the human brain. This interaction mechanism is referred to as human-structure interaction (HSI) and its exact nature remains one of the biggest unanswered questions in the study of human-induced vibration. The problem becomes considerably more challenging when expanded to consider the dynamic influence of a pedestrian crowd. Not only must the interaction between each pedestrian and the structure be considered, but also the interactions between individual pedestrians. Pedestrians moving within a crowd are subject to many physical and psychological influences. Social norms tend to force pedestrians to maintain certain distances between each other, respecting so-called 'personal space'. Individual pedestrians will also have varying levels of motivation to reach their destination. Each pedestrian must navigate through the environment, avoiding obstacles and other pedestrians as efficiently as possible. These factors all influence the behaviour of each individual within the crowd. When the influence of these factors is considered for each pedestrian in parallel, the overall behaviour of the crowd emerges. Emerging crowd behaviour is characterised by a spatially and temporally varying distribution of crowd density and walking velocities.

1. A study of HSI in an effort to better understand the interaction between a walking pedestrian and laterally oscillating structure. This will be achieved through experimental investigation and numerical modelling and this Python Programming

2. development of a modelling framework that utilises the improved understanding of HSI and also captures the dynamic influence of an evolving pedestrian crowd with the coding is bit fearful. This will primarily be a modelling exercise with validation against data and observations in the literature.

2. Literature Reviews

The body of literature concerning lateral excitation of footbridges is extensive. What is reviewed in this chapter is a selection of work that is considered particularly relevant to this project.

In section 2.1 a review of documented loading events on full scale bridges is presented. This serves to highlight that although development of large amplitude lateral bridge response is relatively rare, it is not limited to the well known London Millennium Bridge.

In 2.1.2, test campaigns involving subjects walking on laboratory based oscillating platforms are discussed. The experimental campaign in this project is a development of some of the work discussed in 2.1.2.

Biomechanically based load models and their application in civil engineering are discussed in section 2.13. This section also includes a discussion of the seminal work of Hof et al. on frontal plane stability.

This is followed by a discussion of load models based on crowd-structure interaction modelling in section 2.4, a concept that is further developed in this project. A summary of the reviewed material and where progress remains to be made is given in 2.1.3. 2.1.4 and 2.1.5

2.1 Field observations and full scale testing

What follows in this section is a presentation of field observations and experimental investigations carried out on full scale bridges. The list is not exhaustive, however it encompasses the key landmarks in the literature. To aid the reader, where appropriate the key finding or outcome (in this author's opinion) from an investigation is presented in italic headline form at the start of the relevant subsection.

2.1.1 Paris pont de Solferino

An acceleration threshold of $0.1 m/s^2 - 0.2 m/s^2$ *is identified, at which a transition from random to synchronised pedestrian excitation occurs - step synchronisation is assumed to describe the interaction mechanism. The pont de Solferino footbridge (SF) is a steel arch footbridge that spans 140 m across the river Seine, Fig. 2.1. When it opened to the public in December 1999, large amplitude lateral vibrations developed in the first lateral mode at O.SI Hz. The bridge was temporarily closed shortly after to allow a campaign of testing and investigation to begin.*

The pont de Solferino footbridge (SF) is a steel arch footbridge that spans 140 m across the river Seine, Fig. 2.1. When it opened to the public in December 1999, large amplitude lateral vibrations developed in the first lateral mode at O.SI Hz. The bridge was temporarily closed shortly after to allow a campaign of testing and investigation to begin.

Figure 2.1: Pont de Solferino, Paris

Several crowd loading tests were carried out on the bridge during the initial closure in 2000 and later in 2002. In the latter test campaign, pedestrians were directed to circulate around a predefined route while the number circulating was progressively increased to a maximum of between 207 and 229 [3]. Figure 2.2 shows the lateral acceleration response recorded during one crowd loading test.

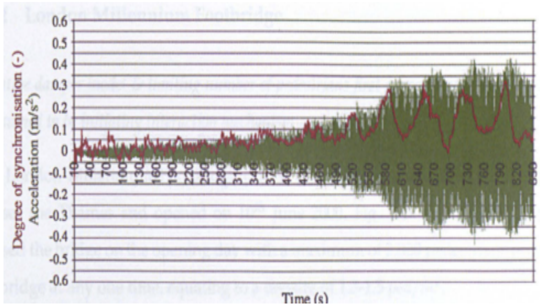


Figure 2.2: Lateral acceleration recorded on the SF during crowd loading tests in which pedestrian numbers were progressively increased to a maximum of 202

2.1.2 London Millennium Footbridge

Negative damper model & limiting number of pedestrians first proposed, step synchronisation is assumed to be initiating interaction mechanism.

The London Millennium Footbridge (LMB) is a shallow suspension footbridge that crosses the Thames and opened on 10th June 2000, Fig. 2.3. Up to 100,000 people crossed the bridge on the opening day with a maximum of 2,000 pedestrians occupying the bridge at anyone time, equating to a density of $1.3-1.5\, ped/m^2$

On the opening day, lateral vibration occurred on the south span at approximately 0.8 Hz (first lateral mode) and the central span at 0.5 Hz and 1.0 Hz (first and second lateral modes). Vibration also occurred less often on the north span at 1.0 Hz. The acceleration amplitude was estimated from video footage to be between 2 and 2.5 m/s^2 with maximum lateral displacement of around 70 mm. This response was sufficient to cause pedestrians to stop walking and use handrails for support where possible. As the crowd density increased so too did the magnitude of lateral oscillations, Although crowd loading tests carried out during the ensuing investigation revealed that the relationship between crowd density and magnitude of response was not linear. Fig. 2.4. It can be seen that that the lateral acceleration grows rapidly at approximately t = 1375 s, corresponding to only a marginal increase in the number of pedestrians on the bridge. The bridge was temporarily closed two days later.

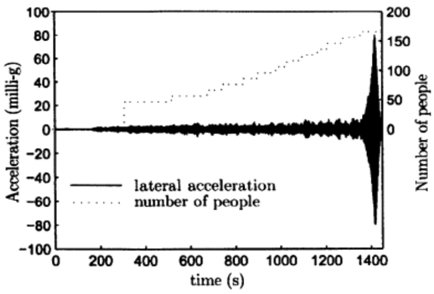


Figure 2.4: Lateral vibration of the north span at 1.0 Hz during crowd loading tests onthe LMB. The number of pedestrians present on the structure is also shown,

The lateral GRF is applied at half the pacing frequency, meaning a bandwidth of approximately 0.8 - 1.1 Hz is common among the population. This correlates well with the 0.8 Hz and 1.0 Hz vibration on LMB. However, the 0.5 Hz vibration was attributed to pedestrians possibly adopting a 'snaking walk' to maintain balance . No further discussion or evidence for this was presented.

Crowd loading tests on LMB showed that the lateral GRF generated by pedestrians was proportional to the local bridge velocity, Fig. 2.5.

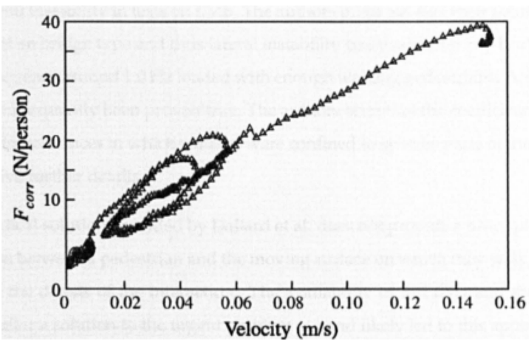


Figure 2.5: Typical lateral force versus velocity for LMB testing by GHD Link :

http://eprints.nottingham.ac.uk/28410/1/594602.pdf

2.1.3 Maple Valley Suspension Footbridge

Step synchronisation & de-tuning demonstrated on a Japanese footbridge suggesting the self limiting nature of SLE. The Maple Valley Great Suspension Bridge (MVB), Fig. 2.6, is located in Japan. It has a central span of 320 m and back-spans of 60 m. The bridge vibrates laterally in the third mode at 0.88 Hz and the fourth mode at 1.02 Hz, depending on *pedestrian distribution. Nakamura carried out an investigation, [4], in which a pedestrian, equipped with a waist mounted accelerometer traversed the bridge. Bridge response data was compared against that obtained from the waist mounted accelerometer, approximating the pedestrian's frontal plane centre of mass motion. In one case, bridge displacement amplitude reached 24 rnm while the pedestrian's CoM displacement amplitude was 43 rnm (displacements were derived from double integration of acceleration data). This data relates to a period when the pedestrian was in the immediate vicinity of the bridge mounted accelerometer. Comparison of the displacement-time histories revealed the pedestrian CoM motion to be synchronised with the bridge and between 1200 and 1600 ahead of the bridge, thus adding energy to the bridge system. In a second case, the bridge amplitude reached 45 rnm, Fig. 2.7. The pedestrian was initially synchronised with the bridge until they apparently 'de-tuned', possibly due to a loss of balance. Nakamura suggested that this is proof of pedestrian step synchronisation. He further proposed that pedestrian-induced lateral excitation is not a divergent phenomenon but converges to a level beyond which pedestrians cannot maintain a steady pace due to loss of balance. A drawback of Nakamura's analysis is that he did not present data showing a transition from an unsynchronised to a synchronised state. By merely showing a comparison between pedestrian and bridge motion for large amplitudes of bridge response, he has confirmed what was already visible from video footage of the LMB, i.e. for large vibration amplitudes pedestrians do synchronise their lateral forcing frequency to that of the oscillating structure and at some point they must stop walking to regain balance.*

Figure 2.6: The Maple Valley Great Suspension Bridge,Japan

A comparison of pedestrian and bridge motion covering a wider range of bridge vibration amplitudes would in theory capture a transition from unsynchronised walking to fully synchronised walking. It is this onset of synchronisation that is of more interest.

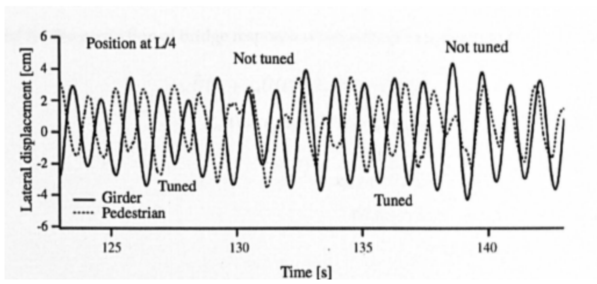


Figure 2.7: Lateral displacement of MVB and pedestrian

2.1.4 LalMandi Footbridge

The Lal Mandi Footbridge is a superimposition pedestrian located in the city of Srinagar in the Indian union territory of Jammu and Kashmir [34]. It has built on 2003-2005 but opened for 13 July 2005. It connects the Lal Mandi and Wazir Bagh areas of the city to the city name Lal Chowk. The bridge consists of Concrete and Steel composition total length of 130 meters (430ft) and width of 4 meters (13ft), in the spermatidal of longest span of 66 meters (217 ft). It is the first suspension-type bridge to come across the Jhelum in the city. The first bridge across the Jehlum was Aali Kadal, built by Sultan Ali Shah in 1415 AD. Six more bridges were built by later rulers, and by the nineteenth century Srinagar came to be known as the 'City of the Seven Bridges'. 'The view from any of the old city's bridges is wholly and unmistakably Kashmiri. Old brick buildings line the banks. The distinctive pagoda-like roof of a mosque or a shrine enlivens the horizon, and in the muddy water of the River Jhelum, a straggling row of doongas flanks the edges. These boats, with their shingled roofs, are the forerunners of Srinagar's houseboat. A particular community lives in them. Formerly this community was associated with ferrying people, livestock and food grains along the river. The past still lingers in their lifestyles even if their occupation has changed. Occasionally one may catch sight of a doonga making its stately progress down the river as the owner shifts residence. Doongas are sparsely furnished - virtually no furniture is seen except for the kitchen, which gleams with copper utensils of every description that line the shelves from floor to ceiling.'

Figure 2.8 LalMandi Footbridge located at connecting Abi Guzar Bund and Silk Factory road.

Figure 2.9 427 AD/Zaina Kadal/Sultan Zain ul Abdin

Two bridges were constructed during the rule of Bakshi Ghulam Mohammed Budshah Kadal and Zero Bridge. The former was built in 1957 across the river Jhelum to connect the Maulana Azad Road to the Civil Secretariat and was named after Zain-ul-Abidin (AD 1420-70), popularly known as 'Budshah', the great king of ancient Kashmir. Sultan Zain-ul-Abidin thus has the honour of having two bridges named after him- the 'Zaina' and 'Budshah' Kadals. Later this bridge named as 'Lal Mandi Footbridge'.

2.1.5 Sudama Setu

Sudama Setu is a pedestrian suspension bridge in Dwarka, Gujarat, India. It is named for Sudama, a childhood friend of Krishna.[35] Proposed in 2005, the bridge was opened in 2016.A pedestrian bridge over Gomti river connects Jagat Mandir of mainland Dwarka and Panchnad or Panchkui Tirth on the island in southeast of it. The bridge was proposed in 2005 to boost tourism on the island. The island has religious as well archaeological significance. There are five sweet water wells called Panchkui associated with five Pandava brothers of Mahabharata. Gomti river was crossed by boats until construction of the bridge. As Jagat Mandir is an Archaeological Survey of India (ASI) protected site, permission was needed for construction of the bridge. The Vadodara circle of ASI gave a nod to the bridge in

2008. The ground breaking ceremony for the bridge was held on 5 May 2011 by Gujarat Tourism minister Jaynarayan Vyas and Rajya Sabha MP Parimal Nathwani. The construction was delayed due to ASI Delhi not granting the permit until 2013. The bridge was completed in February 2016

Reliance Industries Limited, in collaboration with the Gujarat Tourism Department and Gujarat Pavitra Yatradham Vikas Board, built the bridge It is named for Sudama, a childhood friend of Krishna. The bridge was inaugurated by Gujarat Chief Minister Anandiben Patel on 11 June 2016.

The bridge is 166 meters long and 4.2 meters wide, having the capacity to carry 25,000 to 30,000 pedestrians an hour. The bridge is constructed with 40mm locked coil cable ropes provided by Usha Martin Limited.

Figure 2.12 : Image of Sudama Setu

3. Research motivation

Walking on a laterally moving bridge is fundamentally different from walking on stationary ground. In this case a pedestrian must decide where to place their foot based only on the sensory information they have at the time of foot placement. However, during the single stance phase the bridge oscillates, providing additional sensory stimulus altering the feeling of stability.

On this basis it is logical to hypothesise that external perturbations imposed by a swaying bridge during the single stance phase of walking may be corrected for through some form of active control implemented during the single stance phase. Essentially pedestrians may alter their balance in response to a continuously changing lateral inertial force on the body. Testing of this hypothesis is required to further validate the use of the inverted pendulum model in this context. Utilisation of motion capture technology to investigate human motion, a well establish practice in many fields, is now being adopted by civil engineers to record and analyse the dynamic influence of people on structures [12]. These techniques offer an excellent opportunity to investigate not only the GRFs generated on a moving structure but also the accompanying biomechanical behaviour. It is the biomechanical response to base motion, correlated with the GRF that is the missing information required to complete the picture of HSI. An optical assessment of pedestrian balance behaviour would also provide a convenient means of establishing the suitability of the inverted pendulum model. Understanding the detail of the HSI mechanism alone is not sufficient to simulate the dynamic influence of a pedestrian crowd. The work of Venusian et a1. has highlighted the potential of considering the pedestrian crowd as an integral part of the dynamic simulation. The crowd-structure interaction modelling approach offers the ability to simulate the effect of more realistic traffic flows. However there are key limitations inherent in a macroscopic characterisation of the crowd

Accepting the significance of crowd flow behaviour in the dynamic simulation of crowd induced vibration, there is a need to develop a crowd-structure interaction model that retains the benefits of interaction demonstrated by Venusian et al. and addresses the limitations listed above. Discrete element or agent-based crowd modelling may allow such a model to be developed

3.1 Aim & Objective

After Completed this has a simulation tool for assessing the vibration serviceability of any structure that experiences pedestrian loading. Specially this project learned and addressed different case types for Pedestrian loading and harmony to build the postillation, This project has two aims. The first is to improve the understanding of the lateral HSI mechanism. Built a Crowd-induced Load simulation in a Bridge under load. Displacement of a bridge while pedestrians are moving one side to another through graph and plotting, both static and dynamic displacement within oscillation zone. In the simplest terms, an answer is sought to the question; how does pedestrian balance behaviour in response to base motion lead to the generation of self-excited harmonics within the GRF spectrum. An essential step in answering this question is establishing the significance (if any) of active balance control within the HSI mechanism. To that end, based on the reviewed literature, and the reasoning set out in the following hypothesis statement is proposed; Active balance control in response to lateral base motion plays a significant role in determining frontal plane CoM motion, to the extent that the passive inverted pendulum alone can not adequately simulate pedestrian CoM motion during the single stance phase. It is not disputed that active balance control plays a crucial stabilising role during locomotion on stationary as well as moving structures. However what is of interest here is the extent to which it may cause CoM motion to deviate from that predicted by a passive inverted pendulum. This hypothesis will be considered disproven if it can be established that there is no practical difference between simulated and observed CoM motion.

3.1.1 Outcome for Society Betterment

1. Directly meeting needs
The basic modifying need has been trying to have a positive impact on society, this project can made a direct applications that address societal issues like visual improvisation of a pedestrian bridge how dynamics can work and data could be processed in a short span.

2. Empowering people who are often overlooked
An amazing benefit of software/or software based simulation is that it can be freely distributed to everyone, thanks to open-source contributors. This fact truly empowers people. When tools to build revenue and solve business problems are available to everyone—it can level the playing field.

3. Accelerating Bridge Health Monitoring
One of the most exciting facets of data visualisation is its power to improve and accelerate every other field. “Data science and artificial intelligence (AI) as subsets of computer science allow people and organizations to accelerate and ‘prepackage thought.’ In this way, computer science and artificial intelligence/simulation can make any other discipline many, many times better.”

4. Predicting and avoiding catastrophes

Computer science with referring to Civil Engineering is scaling—and scaling very fast for that matter, according to Nand. Applying computer science to prediction can have a huge impact on the world. Thus are predicting human behaviour and load towards bridge,; These kind of module predicting climates, seasons, ocean currents, etc. And can avoid unexpected collapse and catastrophes for humanity.

4.The Duhamel Integral - Basic Concept

The Duhamel integral is a simple way to simulate the response of a single-degree-of-freedom (SDoF) structure to any form of dynamic loading to a major load situated condition. When the loading is described by a simple mathematical function, It can analytically solve the integral. However, for practical application, this is very often not the case. If the loading is more complex or defined by a time-history of discrete force values, numerical integration can be employed. For the Indian clustered bridge scenario this is almost preferable to induce this.

Thus it will cover the details of how the integration can be handled in both cases. For now, it just need to understand the concept behind the Duhamel integration approach. **It can start by considering an impulse**. This is simply a force, $p(t)$, that is applied for a very short duration, say δt. The impulse of the force is defined as the area below the force-time history. So, a unit impulse would have an area of 1.

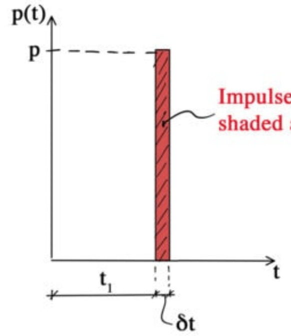


Figure 1: Impulse associated with the force p(t) applied for duration delta t

$$I=\int_{t}^{t+\delta t} p(t)\,dt=1(for\,a\,unit\,impulse)-----(Eq\,1)$$

Now, let's think about applying an impulse to a SDoF system. It would be seen below that I can determine an expression for the resulting free vibration response of the system. Remember, the impulse is a very short duration, high magnitude load. So, it will induce a short duration driven response followed by a decaying (we'll assume a damped system) free vibration response

So, how do it can use this new information to simulate the response of a SDoF system to general dynamic loading? **The proforma is to consider the general dynamic loading as a sequence of short duration impulses**. Each impulse will induce a free-vibration response. To obtain the overall dynamic response, we simply superimpose the free vibration responses due to each impulse, **Fig. 2**.

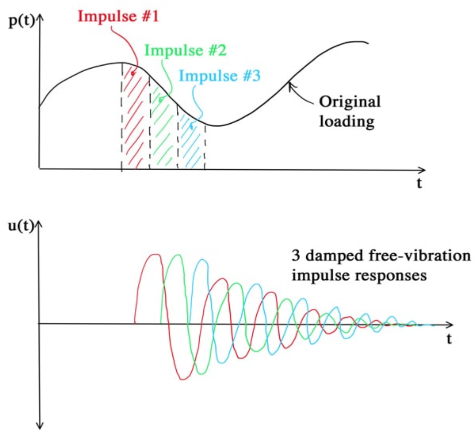


Figure 2. General dynamic loading considered as a sequence of N impulses yielding N free-vibration responses that can be superimposed to obtain the response to the original dynamic loading.

Hopefully it can see that at a concept level, there's not much complexity to the Duhamel integral. It's a very simple but versatile tool. There is one limitation to be aware of. Because we rely on superposition to obtain the overall structural response it must assume the behaviour of the dynamic system is linear. If this is not the case, an alternative solution strategy should be sought, such as direct integration of the equation(s) of motion.

5. Damped Impulse Response

It is observed next task is to determine an expression for the impulse response of our SDoF system. It will start by considering Newton's second law which states that the rate of change of velocity is directly proportional to the force that causes it, mathematically would have form **Indian Institute of Technology, Mandi curricular of Module 2:** it can be driven

$$F(\tau)=m\frac{dv}{d\tau}......Eq(2)$$

Rearranging slightly, it has,

$$[F(\tau)d\tau=mdv].......Eq(3)$$

Notice that $F(\tau)d\tau$ is actually an impulse; recall from above that the impulse associated with force $p(t)$ was the area below the force-time history $p(t)\delta t$. it can see that the impulse $F(\tau)d\tau$ induces an instantaneous change in velocity dv . Also note that it has introduced the time variable τ_1 , distinct from, t_1 to represent the time of application of a specific impulse. This will become clearer when it see both τ and t in the same expression below.

The instantaneous change in velocity dv , can be interpreted as an initial velocity v_0 associated with the application of the impulse

$$[dv=v_0=\frac{F(\tau)d\tau}{m}]-------Eq(4)$$

Now main problem reduces to determine the free vibration responses of a SDoF system with non-zero initial velocity v_0 at time τ . it can show that the damped free vibration response of a SDoF structure given by,

$$\upsilon(t)=e^{-\xi\omega_i t}[\frac{v_0+\upsilon_0\xi+\omega_n}{\omega_d}+\sin(\omega_d t)+\upsilon_0\cos(\omega_d t)]---------Eq(5)$$

Where ξ is the damping ratio and ω_n is the angular natural frequency, ω_d is the damped natural frequency and υ_0 is the initial displacement. It can assume the derivation for the sake of minimise the calculation from Structural Dynamics **Indian Standard Code IS 875 (Part 1, Part 2, and Part 5).**

Now If it is assume the impulse is applied at time τ LET, u_0 = 0 and let υ_0 equal to the value in equation 3 it can obtain an expression for the incremental response, $d\upsilon(t)$ due to the impulse $F(\tau)d\tau$ applied at time τ

$$d\upsilon(t)=e^{-\xi\omega_n(t-\tau)}[\frac{F(\tau)d\tau}{m\omega_d}\sin\omega_d(t-\tau)]---------Eq(6)$$

Note that it now have the quantity of time elapsed since the impulse was applied, denoted as $t-\tau$. This is why it introduced the time variable τ originally. Thus they are quite into the main equation to most of the performing test the total response of the system at time t , thus need to sum up of superimpose the influence of all of the impulses application of applied up to this point in time. It can do this by integrating both sides of equation 6.

$$u(t)=\frac{1}{m\,\omega_d}\int_0^t F(\tau)e^{-\xi\,\omega_n(t-\tau)}[\frac{F(\tau)\,d\,\tau}{m\,\omega_d}\sin\omega_d(t-\tau)]----------Eq(7)$$

In this step in the calculation purpose find that this expresses the expression what is referred as **Duhamel's Integral.**

6. Analytical Solutions

As it said at the outset, there are two approaches to solving the Duhamel integral (or any integration in practice), analytical solution and numerical solution. By far the most practically useful method is numerical solution. Most of the interesting problems, from an engineering perspective, require numerical solution. It is the most preferred for **Indian Suspension and Pedestrian bridge condition** Although a numerical solution will always be an approximation, any error introduced usually pales into insignificance when compared against other uncertainties bakes into this modelling. Numerical solution comes with the huge advantages of simplicity and versatility. Thus've always been more of a programmer than mathematician approach so this may be biased. Having said all of this, This will demonstrate an analytical solution first, for completeness; even then use of **Python Version 3.9.**

Analytical solution to ramp loading

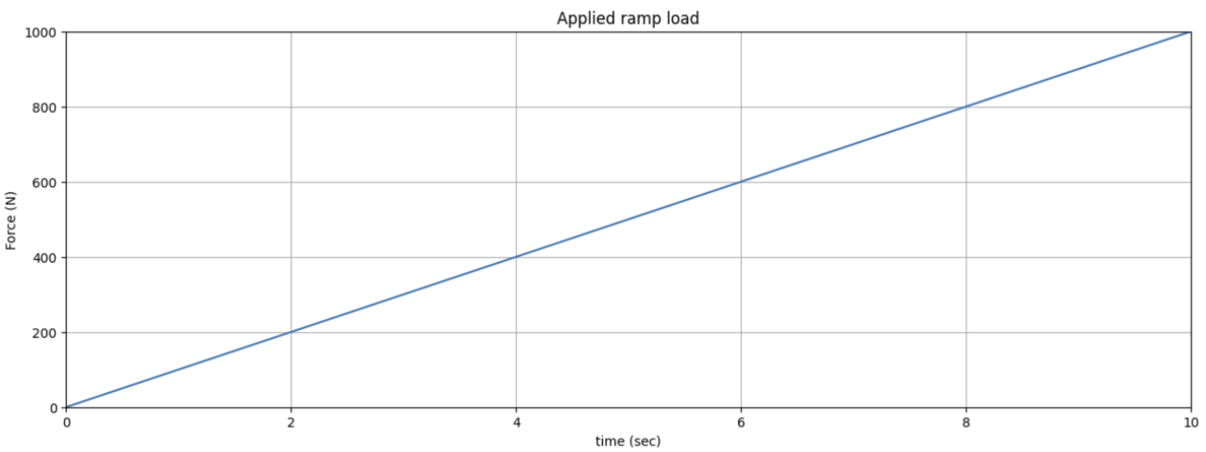
Consider a load that increase linearly from 0 to magnitude of P_0 between $t=0$ and $t=t_1$. it can represent the loading with a simple mathematical expression,

$$p(t)=\frac{P_0 t}{t_1}-------EQ(8)$$

The fact that applied loading can be captured with such a simple expression is our first clue that we might be able to solve this analytically. Assigning some arbitrary values to P_0 and t it can generate a plot of the loading.

```python
# DEPENDENCIES and DEFAULTS
import math #..................................Math functionality
import numpy as np #............................Numpy for working with arrays
import matplotlib.pyplot as plt #................Plotting functionality

P0 = 1000 #(N) Max load (arbitrary value for visualisation only)
t1 = 10 #(sec) Rise time (arbitrary value for visualisation only)
delT = 0.1 #(sec) Time-step
t = np.arange(0, t1+delT, delT) #Time vector
p = P0*t/t1 #Force vector

#Set up figure and axes
fig = plt.figure()
axes = fig.add_axes([0.1,0.1,2,1])
axes.plot(t,p)

#Housekeeping
axes.set_xlabel('time (sec)')
axes.set_ylabel('Force (N)')
axes.set_title('Applied ramp load')
axes.set_xlim([0,t1])
axes.set_ylim([0,P0])
plt.grid()
plt.show()
```

By the Fact the output will be



Next, it can insert our expression for the applied loading directly into the Duhamel integral expression.

$$u(t)=\frac{1}{m\omega_d}\int_0^t [\frac{P_0\tau}{t_1}e^{-\xi\omega_n(t-\tau)}\sin\omega_d(t-\tau)d\tau]---------Eq(9)$$

In this case, they are only considering the response of the system during the application of the ramp load and not the free vibration that would follow. As such, damping will not have an appreciable influence on the response. It can therefore simplify the task by considering an undamped system. As a result, thus the expression from **IS CODE 456: 2000** reduces to,

$$u(t)=\frac{P_0}{t_1 m\omega_n}\int_0^t [\tau\sin\omega_n(t-\tau)]---------Eq(10)$$

Now it need to integrate this expression; for this we'll use SymPy.

```
import sympy as sym #Importing SymPy
sym.init_printing() #printing formed equation

#Define some symbols to built the integration more pronounced
m = sym.Symbol('m')
w = sym.Symbol('w')
P0 = sym.Symbol('P0')
t1 = sym.Symbol('t1')
tau = sym.Symbol('tau')
t = sym.Symbol('t')

#Constructing the function to integrate
f = tau * sym.sin(w*t-w*tau)

# Use SymPy to get the definite itegral with respect to tau between 0 and t
defInt = sym.integrate(f,(tau,0,t))

print('\n SymPy generated the following expression for the definite integral:\n')
sym.simplify(defInt) #Attempt to simplify the definite integral
```

It will give the output,

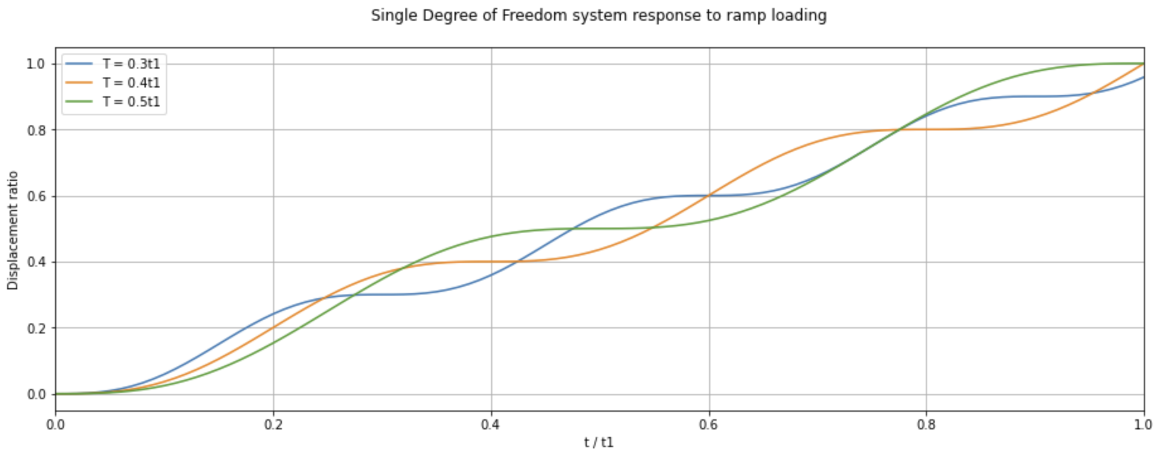
```
SymPy generated the following expression for the definite integral:
```

$$\begin{cases} \frac{tw-\sin(tw)}{w^2} & \text{for } w > -\infty \wedge w < \infty \wedge w \neq 0 \\ 0 & \text{otherwise} \end{cases}$$

To complete the ease, it can plot the response. When t_1 and normalise the displacement axis by the maximum static displacement, $\frac{P_0}{k}$ .It will perform the response calculation for a range of different SDoF systems with the same mass but different stiffness, yielding a range of natural frequencies. For each system it'll specify the period of the system as a proportion of the rise time of the load.

```python
P0 = 1000 #(N) Max load (arbitrary value for visualisation only)
t1 = 10 #(sec) Rise time (arbitrary value for visualisation only)
delT = 0.1 #(sec) Time-step
t = np.arange(0, t1+delT, delT) #Time vector defying steps of delT
p = P0*t/t1 #Force vector

m=20 #Mass of the system_constant
periodRange = [0.3,0.4,0.5] #Range of system periods as a propertion of rise time

#Initialise a figrue to plot onto
fig = plt.figure() #defying figure below
axes = fig.add_axes([0.1,0.1,2,1])

for pr in periodRange: #starting a loop through array of numbers preriodrange
    T = pr*t1
    wn = 2*math.pi/T
    k=m*wn**2
    u = (P0/k)*((t/t1)-((np.sin(wn*t))/(wn*t1)))
    axes.plot(t/t1,u/(P0/k), label=f"T = {pr}t1")

#Housekeeping
axes.set_xlabel('t / t1')
axes.set_ylabel('Displacement ratio')
axes.set_title('\n Single Degree of Freedom system response to ramp loading \n')
axes.legend(loc='upper left')
axes.set_xlim([0,1]) #making plot line to axis system to this below grid
plt.grid()
plt.show()
```

This example is a relatively trivial case but it demonstrates the analytical integration route. As here mentioned, these are not likely to deploy analytical integration too often, so it will leave this method behind and focus on numerical integration from here on.



7. Numerical Solutions

From what it has seen so far, solving the Duhamel integral analytically is certainly possible, but as it has already mentioned, real-life engineering needs more detailing and solvation and often It can't neatly describe in this applied loading in a mathematical function. This is where the numerical solution strategy comes in. In this section it is build a function that numerically solves the Duhamel integral for any arbitrary loading it pass into the function. It will start by restating the Duhamel integral for a damped system,

$$u(t)=\frac{1}{m\omega_d}\int_0^t [F(\tau)e^{-\xi\omega_n(t-\tau)}\sin\omega_n(t-\tau)d\tau]---------Eq(11)$$

It can separate out the time variables with the help of a trigonometric identity,

$$u(t)=\frac{1}{m\omega_d}\int_0^t F(\tau)e^{\xi\omega_n(t-\tau)}[\sin(\omega_n t)\cos(\omega_d\tau)-\cos(\omega_d t)\sin(\omega_d\tau)]d\tau---------Eq(12)$$

Extended Portion

Since it has integrating with respect to τ it can restructure our expression to isolate the terms requiring numerical integration,

$$u(t)=Ae^{-\xi\omega_n t}\sin(\omega_d t)-Be^{-\xi\omega_n t}\cos(\omega_d t)---------Eq(13)$$

where

$$A=\frac{1}{(m\omega_d)}\int_0^t e^{-\xi\omega_n t}F(\tau)\cos(\omega_d\tau)d\tau---------Eq(14)$$

$$A=\frac{1}{(m\omega_d)}\int_0^t e^{-\xi\omega_n t}F(\tau)\sin(\omega_d\tau)d\tau---------Eq(15)$$

A and B now represent two integrals that it can numerically evaluate. This in it uses the trapezoidal rule here,

If it consider the integrated (of A or B) as a simple function $f(\tau)$ we could plot the function on a $\tau-f(\tau)$ plane. We know that the integration is simply the area bounded by the function and the horizontal axis. Using the trapezoidal rule we can approximate this as.

$$Area=\int_0^{t_n} f(\tau)d\tau\approx\frac{\Delta_t}{2}[f(t_0)+2f(t_1)+......+2f(t_{n-1})+2f(t_n)]------EQ(16)$$

Now all that remains is to write a function to cycle through the time history computing the numerical integrations in equations 14 and 15 and then compute the response using equation 13.

8. Implementing Numerical Solutions

Thus this note all of it's code into a function that accepts a time and force vector. This way it be can easily reuse the function from this file and deploy it elsewhere down the road. Thus this can be start by defining Single Degree of Freedom system parameters; mass m, damping ratio ξ and natural frequency/angular natural frequency $\frac{f_n}{\omega_n}$. Where it can also define the damped natural frequency, $\omega_d = \omega_n \sqrt{1-\xi^2}$.

```
m = 1000 #(kg) Mass
xi = 0.05 # Damping ratio
f = 1.5 #(Hz) Natural frequency
wn = 2*math.pi*f #(rads/s) Angular natural frequency
wd = wn*math.sqrt(1-xi**2) #(rads/s) Damped angular natural frequency
```

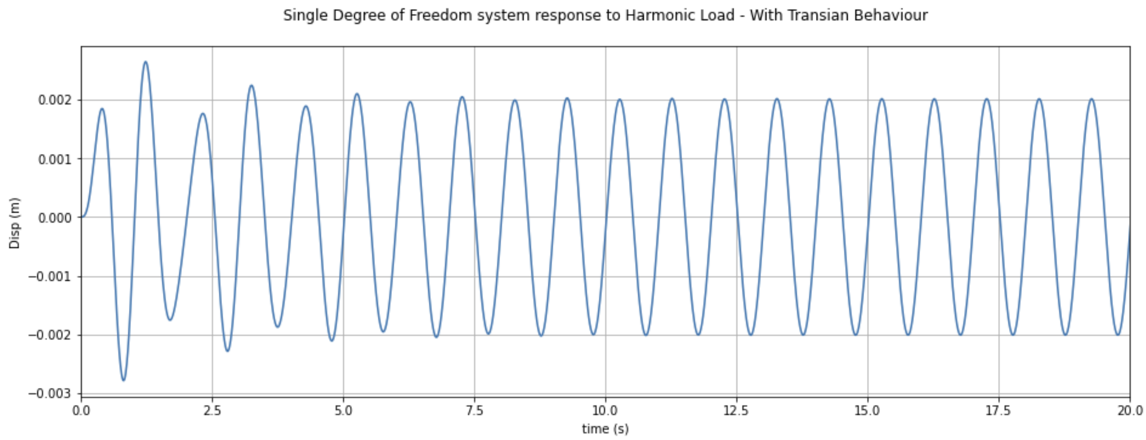
For the purposes of simulation here this define a harmonic forcing function. This will make it easier to validate our Duhamel integral function against a closed-form solution. First it can define a time vector in the usual way.

```
tMax = 20 #(s) Max time
delT = 0.01 #(s) Time-step
time = np.arange(0, tMax+delT, delT) #Time vector

f_Force = 1 #(Hz) Forcing frequency
wf = 2*math.pi*f_Force #(rads/s) Angular forcing frequency
P=100 #(N) Force amplitude loading
force = P*np.sin(wf*time) #Force vector
```

The function has write next will accept the time and force vectors as arguments. It rely on the fact that all of the system parameters are defined globally so it won't bother explicitly passing them into the function. Here basically need to cycle through the time record and compute the area of the i^{th} trapezoid and add this to a cumulative sum of areas as we move along the time axis. In this way we can calculate appropriate values for A and B at each time-step. Then, it can calculate the actual response for each time-step using equation 13.

```
#Defyning Duhamel function with Time Vector 'T' and Force Vector 'F'
def Duhamel(T, F):
    #Initialise a container of zeros to hold the displacement values
    U = np.zeros(len(T))

    #Initialise values for the cumulative sum used to calculate A and B at each time-step
    ACum_i=0
    BCum_i=0

    #Cycle through the time vector and evaluate the response at each time point
    for i, t in enumerate(T):
        #Only start calculating on the second iteration (need two values for trapezoidal area calculation)
        if i>0:
            #Calculate A[i] - equation 4
            y_i = math.e**(xi*wn*T[i]) * F[i] * math.cos(wd*T[i]) #Value of integrand at current time-step
            y_im1 = math.e**(xi*wn*T[i-1]) * F[i-1] * math.cos(wd*T[i-1]) #Value of integrand at previous time-step
            Area_i = 0.5*delT*(y_i+y_im1) #Area of trapezoid
            ACum_i += Area_i #Cumulative area from t=0 to current time
            A_i = (1/(m*wd))*ACum_i #Value of A for the current time-step

            #Calculate B[i] - equation 5 (same notes as for A above)
            y_i = math.e**(xi*wn*T[i]) * F[i] * math.sin(wd*T[i])
            y_im1 = math.e**(xi*wn*T[i-1]) * F[i-1] * math.sin(wd*T[i-1])
            Area_i = 0.5*delT*(y_i+y_im1)
            BCum_i += Area_i
            B_i = (1/(m*wd))*BCum_i

            #Calculate the response - equation 3
            U[i] = A_i*math.e**(-xi*wn*T[i])*math.sin(wd*T[i]) - B_i * math.e**(-xi*wn*T[i])*math.cos(wd*T[i])

    return U
```

Now that this function defined what it just need to call it and assign the output to a variable for plotting.



Thus this can see above that the response contains a transient component that dissipates due to damping, leaving a steady-state response which is exactly what we would expect from the harmonic excitation of a Single Degree of Freedom system.

9. Validating the Source Code

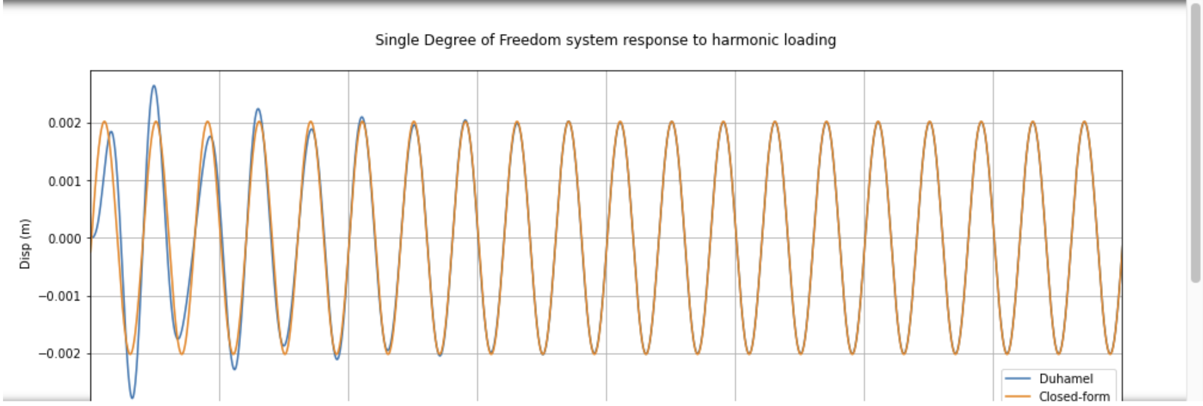
Now this is confirms that the validity of our Duhamel function by comparing the calculated response with the closed-form (analytical) solution for harmonic loading of a Single Degree of Freedom system. It could plot the complete closed-form response including the transient and steady-state components but for it's validation exercise, it will be sufficient to only plot the steady-state response $u_s t$ will be given by

$$u_s(t)=\frac{P}{k}\left[\frac{1}{[(1-\beta^2)^2+(2\,\xi\,\beta)^2]}\right][(1-\beta^2)\sin(\omega\,t)-2\,\xi\,\beta\cos(\omega\,t)]-----------(EQ\,17)$$

From the study of fundamental Structural Dynamics, it will recognise this as the dimensionless form of the steady-state response equation

```
#Steady-state response with the system
beta = wf/wd #Frequency ratio
k=m*wd**2 #(N/m) Stiffness

#Break equation into two for convenience
O = (P/k)*(1/((1-beta**2)**2 + (2*xi*beta)**2))
response_cf = O*((1-beta**2)*np.sin(wf*time) - 2*xi*beta*np.cos(wf*time)) #here xi is defined globally previously
```

```
fig = plt.figure()
axes = fig.add_axes([0.1,0.1,2,1])
axes.plot(time,response, '-', label='Duhamel')
axes.plot(time,response_cf,'-',label='Closed-form')

#Housekeeping
axes.set_xlabel('time (s)')
axes.set_ylabel('Disp (m)')
axes.set_title('\n Single Degree of Freedom system response to harmonic loading \n')
axes.legend(loc='lower right')
axes.set_xlim([0,tMax])
plt.grid()
plt.show()

print(" \n Dumahel's response and is the actual reponse that continiues to the left to end but, But trasnian responde w
```



Now it can see that both solutions agree once the transient response has decayed away. This gives sudden confidence in the Duhamel integral code this written on Jupiter NoteBook environment. This is quite a powerful tool that have at our disposal now. Although it only used it to calculate the response due to harmonic loading, it can feed it any forcing function and it will return the Single Degree of Freedom system response.

So, in the next step of this project, it will use our Duhamel integral function to do something a little more interesting. It going to estimate the vertical vibration response of a footbridge to pedestrian traffic. Although, the program will focus on a simply-supported footbridge, the same principles can be applied to crowd-induced vibration of any structure, although there are some specific caveats we'll discuss along the way.

10. Dynamic Crowd Loading Concepts

Volumes of research has been produced on the topic of human-induced vibration and in particular crowd-induced vibration. Research in this area has intensified in the last 20 years as structures, in particular floor plates, have become lighter and more slender. This has resulted in unacceptably large occupant-induced vibrations being observed in service more frequently.

For the purposes of this analysis project, when it say human-induced vibration, it is referring to the vibration induced by a single pedestrian as they walk. Crowd-induced vibration will refer to the response induced by multiple people simultaneously walking across a structure.

Human-induced vibration is a complex and fascinating dynamic problem because it involves interaction between the structure and the occupying pedestrian, both of which are dynamic systems. The problem gets even more interesting (and complex) when it start to introduce interaction between pedestrians, and try to capture the tendency for a pedestrian to be influenced by the behaviour of those around them. Thus it won't attempt to dive in to the detail too much here but would do need some understanding of the problem before we start stinting the code.

In addition to dividing the problem into human-induced and crowd-induced vibration, this can further divide the problem into one that considers vertical vibration and one that considers lateral vibration. Human locomotion response and balance behaviour is very different in each case. This project focus here on vertical vibration and leave the problem of lateral human-structure interaction and synchronisation for further work if needed.

Footfall force modelling

In the simplest of terms, when a person walks across a structure, in this case a linear footbridge, those will apply a vertical footfall force or **ground reaction force** (GRF) with each footstep. It can crudely model this GRF using a sine function, or more specifically a series of sine functions representing the harmonic components of the GRF. This assumes that the GRF is perfectly periodic which is itself a simplification but we can live with it. So, the GRF can be represented as the static weight of the pedestrian, G , added to a Fourier series representing the dynamic component of loading,

$$F_v(t)=G+\sum_{i=1}^{n} G\,DLF_i \sin(2\pi i f_v t-\phi_i)------EQ(18)$$

where DLF is i^{th} Fourier coefficient (also knows as a Dynamic Load Factor), ϕ_i is the phase of the i^{th} harmonic and f_v is the pacing frequency. It will only concern ourselves with the fundamental harmonic, i.e. that component of the GRF at the pacing frequency. It will ignore the higher integer harmonics in this example, but these may be significant, depending on the dynamic characteristics of the structure in question.

In order to model the GRF this needs to know the Fourier coefficient of the fundamental harmonic. Again there has been a lot of research work on this over the years but here it will use an early relationship proposed by [29] that relates the DLF to the pricing frequency f_v ,

$$DLF = 0.41(f_v - 0.95) \leq 0.56 - - - - - EQ(19)$$

This relationship is valid for pacing frequencies between $1\,Hz$ and $2.8\,Hz$ .Now, this raises the question of what should the pacing frequency be? For this this project rely on a relationship presented by *[30]* that was developed from the earlier work of [31]

$$f_v = 0.35\,\upsilon_p^3 - 1.56\,\upsilon_p^2 + 2.93\,\upsilon_p - - - - - - - EQ(20)$$

where υ_p is the working velocity of course now here is presented with the task of working out what a sensible walking velocity should be. Here this project rely on work by [31] who observed that walking velocities in a sample of pedestrians were normally distributed with a mean value of $1.3\,m/s$ and a standard deviation of $0.125\,m/s$

An import feature in the modelling of crowd-induced vibration is the flow behaviour of the pedestrian crowd and their evolving distribution across the structure. As a pedestrian crowd becomes more dense, walking velocities are reduced due to congestion. However in this analysis, thus ignore all traffic effects due to interaction between pedestrians. This is a reasonable assumption for *uncongested crowds where it's reasonable to expect that an individual's desired walking velocity won't be constrained by neighbouring pedestrians*.

Now at a point where this can approximate the GRF imparted by a pedestrian with a **mass of Indian Average body weight will be** $65\,kg$ walking with a velocity of $1.3\,m/s$ in **as per Indian Pedestrians** a straight line. If this assume the pedestrian walks across a $60\,m$ long footbridge, it will take then $\frac{60 \quad m}{1.3 \;\, m/s} = 76.9\,seconds$ to cross the bridge. This gives the duration of load application.

```
L = 60 #(m) Bridge span
vp = 1.3 #(m/s) Pedestrian walking velocity
tMax = L/vp #(s) Crossing time of bridge
m = 65 #(kg) Avarage Indian Pedestrian mass weight
G = 9.81*m #(N) Static weight of pedestrian

fv = 0.35*vp**3 - 1.59*vp**2 + 2.93*vp #(Hz) Pacing frequency
DLF = 0.41*(fv-0.95) #Dynamic load factor
print(f"- The Dynamic Load Factor = {round(DLF,3)} and the pacing frequency is {round(fv,2)} Hz ({round(fv,2)} steps pe
print(f"- Duration of a single step is {round(1/fv,2)} seconds")
```

```
- The Dynamic Load Factor = 0.386 and the pacing frequency is 1.89 Hz (1.89 steps per second)
- Duration of a single step is 0.53 seconds
```

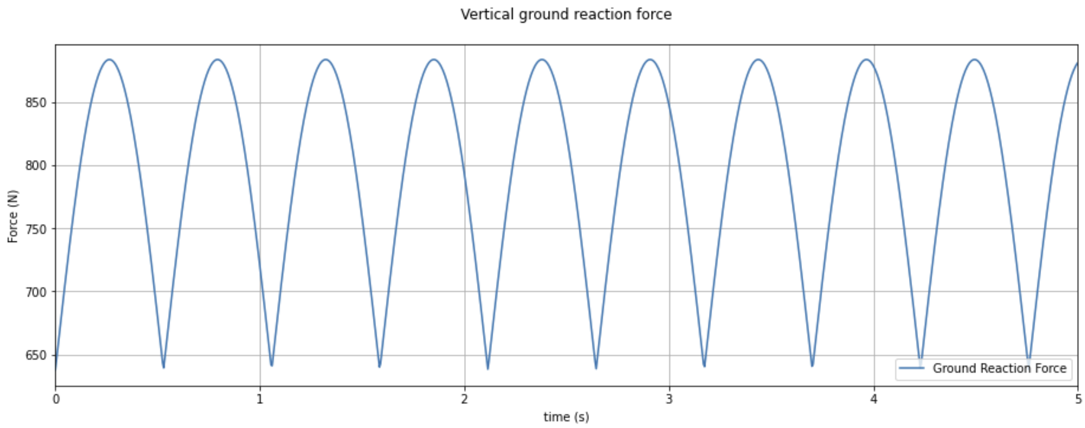
- The DLF = 0.386 and the pacing frequency is 1.89 Hz (1.89 steps per second)

- Duration of a single step is 0.53 seconds

Now it can define the time vector and plot the GRF. Note that here are only interested in the positive value of the GRF as the footfall force never acts to pull the walking surface upwards!

This means that a single sine wave actually models 2 footsteps—this is why the frequency is divided by 2 in the code below.

```
delT = 0.005 #(s) Time-step
time = np.arange(0, tMax+delT, delT) #Time vector
Fv = G + abs(G*DLF*np.sin(2*math.pi*(fv/2)*time)) #Static + Dynamic Ground Reaction Force

fig = plt.figure()
axes = fig.add_axes([0.1,0.1,2,1])
axes.plot(time,Fv, '-', label='Ground Reaction Force')

#Housekeeping
axes.set_xlabel('time (s)')
axes.set_ylabel('Force (N)')
axes.set_title('\n Vertical ground reaction force \n')
axes.legend(loc='lower right')
axes.set_xlim([0,5])
plt.grid()
plt.show()
```



Each of the half-sine peaks represents the force generated by one foot hitting the surface. Remember, a more rigorous model would need to consider the influence of varying walking speed. Even when it makes the assumption of constant walking speed, the GRF is not a perfectly periodic function so there is some approximation involved in representing it as a Fourier series.

Furthermore, here are only representing the fundamental harmonic component here – an actual recorded GRF resembles a square wave more than a half-sine wave. Nevertheless, this will still give here a reasonably realistic first approximation of human-induced vibration response when we apply it to a structure.

It's also worth pointing out that in this modelling it is going to assume that there is *no human-structure interaction.* This is generally a bigger problem for lateral oscillations, due to the fact that humans tend to exhibit higher sensitivity to lateral support motion versus vertical motion. This is likely a result of our bi-pedal nature and inherent instability in the lateral direction. So, just to be clear, here assuming the vertical bridge motion does not influence the pedestrians behaviour (and their GRF) in any way.

The bridge/beam structure

Now this can turn here attention to the bridge structure. It will keep things simple here and assume that module bridge can be modelled as a simply supported beam. This means that the modal properties are easily obtained. In particular we're interested in the mode shapes $\phi_n(\chi)$ which can be obtained from,

$$\phi_n(\chi)=\sin\frac{n\pi\chi}{L}------EQ(21)$$

where n is the mode number and x is the longitudinal coordinate along the bridge/beam. For a deeper dive into modal analysis, Here this project, it is going to assume that the fundamental vertical mode of the bridge has a frequency of $2.5\,Hz$. This is sufficiently close to the average pacing frequency that if this were a real bridge it would be concerned that unacceptable vibrations may be observed in service.

Also assume that the bridge has a damping ratio of 2.5%. is $M\,is\,2000\,kg/m$. **as per Indian Population declinable** which is going to make another simplifying assumption here; Thus assume that the mass of pedestrians on the bridge has no influence on the modal mass of the dynamic system being simulated. Our justification here is that the mass of pedestrians is insignificant compared to the mass of the bridge. If this were not the case it could consider modifying the modal mass to reflect the evolving distribution of pedestrians on the bridge. Refer to the *[31].* The modal mass, m_n remains constant and can be calculated from the expression,

$$m_n=M\int_0^L \phi_n(\chi)^2\,d\chi$$

$$m_n=M\int_0^L \sin^2\frac{\pi\chi}{L}\,d\chi$$

$$m_n=\frac{ML}{2}--------EQ(22)$$

So, for the first mode, the modal mass is simply half the actual mass of the structure.

11. Dynamic Analysis - Bridge + 1 Pedestrian

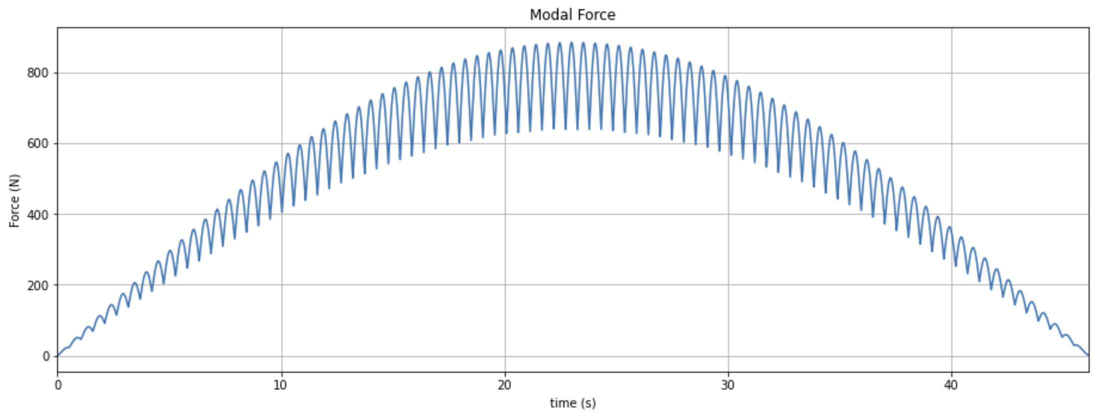
Since it is assuming that the bridge has only one modal frequency in the range of interest, it only need to simulate a single vibration mode. Conveniently for us and through the magic of modal analysis, this means that the dynamic analysis of the bridge reduces to simulating the response of a Single Degree of Freedom system. If this project needed to simulate the combined response of multiple modes, this is easily done through modal superposition.

The first task is to turn our Ground Reaction Force F_v into a modal force F_n . This simply means multiplying it by the unity-normalised mode shape value at the pedestrian's location, χ_p .

$$F_n = F_v \phi(\chi_p) ------- EQ(23)$$

Since **Indian pedestrian** is walking with a constant velocity on an average, it can easily determine their position, χ_p as a function of time. From here it can determine the value of the mode shape as a function of time and directly multiply this by the GRF calculated earlier to obtain the modal load applied to the SDoF modal model as a function of time.

```
fig = plt.figure()
axes = fig.add_axes([0.1,0.1,2,1])
axes.plot(time,Fn, '-', label='Modal load')

#Housekeeping
axes.set_xlabel('time (s)')
axes.set_ylabel('Force (N)')
axes.set_title('Modal Force')
axes.set_xlim([0,tMax])
plt.grid()
plt.show()
print("\n\t Dynamic Influance of the Padestrian on the first mode of Vibration growing to maximum at MID SPAN")

print("\n This is a Modeal Force Diagram for 1 Pedestrian is moving through start to end of a bridge and foot fall forc
```




```
	Dynamic Influance of the Padestrian on the first mode of Vibration growing to maximum at MID SPAN

 This is a Modeal Force Diagram for 1 Pedestrian is moving through start to end of a bridge and foot fall force has b
een scaled in Mode shape Value
```

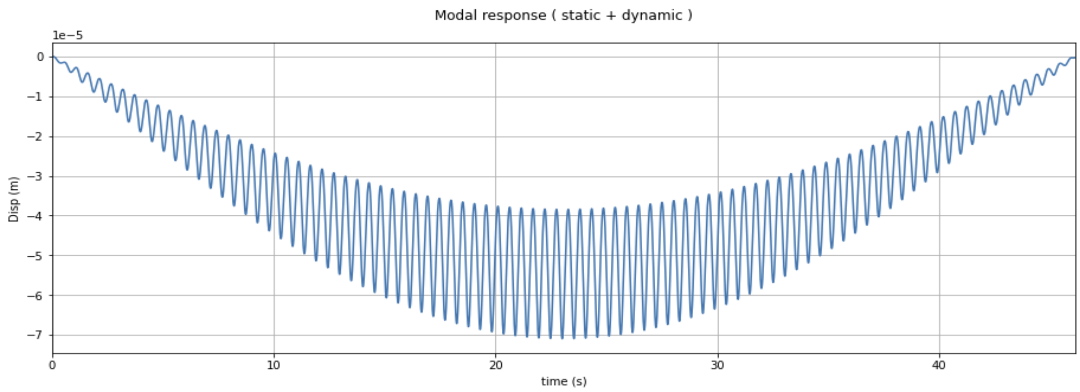
At this point it's helpful to visualise the modal force. Remember, this is the force applied to the first bridge/beam mode with a half-sine mode shape. It can think of this force as the influence of the GRF on mode one which is at a maximum at the centre-span position.

It can clearly see that the dynamic influence of the pedestrian's GRF is increasing as they move towards the centre of the bridge and diminishing thereafter. Now it's just a matter of providing this force along with the **time vector directly into our Duhamel function**. Since this function relies on globally defined variables, we need to be careful to redefine these for our bridge before calling the function.

```
M = 2000 #(kg/m) Mass per unit length]
m = 0.5*M*L #(kg) Modal mass of mode 1
xi = 0.025 #Damping ratio
fn = 2.5 #(Hz) Bridge modal frequency
wn = 2*math.pi*fn #(rads/s) Angular modal frequency
wd = wn*math.sqrt(1-xi**2) #(rads/s) Damped angular modal frequency
```

```
response = Duhamel(time, Fn) #Response calculated using the Duhamel integral function
```

```
fig = plt.figure()
axes = fig.add_axes([0.1,0.1,2,1])
axes.plot(time,-response, '-')

#Housekeeping
axes.set_xlabel('time (s)')
axes.set_ylabel('Disp (m)')
axes.set_title('Modal response ( static + dynamic )\n')
axes.set_xlim([0,tMax])
plt.grid()
plt.show()

print("\t This visualisation reflects the vertical oscillating of a Bridge when 1 Padestrian is moving ")
```



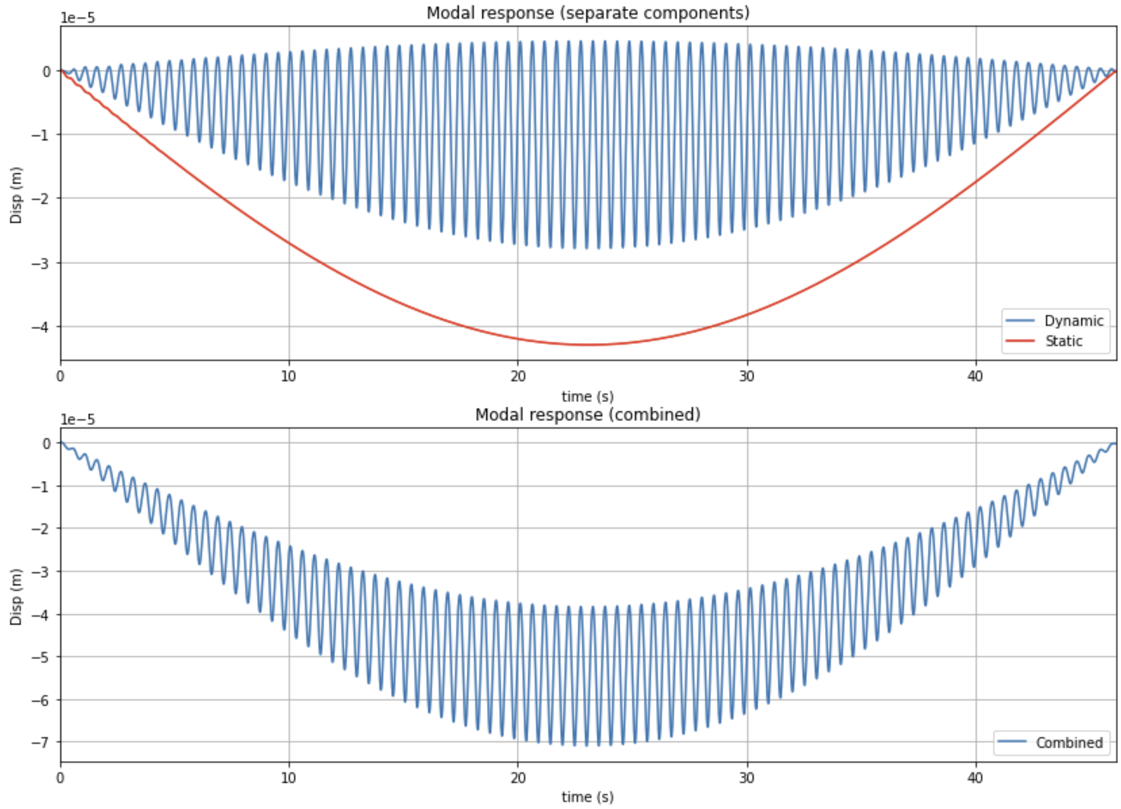

```
This visualisation reflects the vertical oscillating of a Bridge when 1 Padestrian is moving
```

After plotting the modal response here can see the oscillation behaviour superimposed on top of the static deflection. It can visualise this more clearly if we separate out the static and dynamic components of the GRF and simulate their influence separately.

```
Fn_static = G*phi #Static component of GRF
Fn_dynamic = abs(G*DLF*np.sin(2*math.pi*(fv/2)*time))*phi #Dynamic component of GRF

response_static = Duhamel(time, Fn_static) #Response due to constant magnitude moving load
response_dynamnic = Duhamel(time, Fn_dynamic) #Response due to footsteps (with static load component removed)
```

```
fig, axes = plt.subplots(figsize=(14,10),nrows=2,ncols=1)
axes[0].plot(time,-response_dynamnic, '-', label='Dynamic')
axes[0].plot(time,-response_static, 'r-', label='Static')
axes[0].set_xlabel('time (s)')
axes[0].set_ylabel('Disp (m)')
axes[0].set_title('Modal response (separate components)')
axes[0].legend(loc='lower right')
axes[0].set_xlim([0,tMax])
axes[0].grid()

axes[1].plot(time,-response, '-', label='Combined')
axes[1].set_xlabel('time (s)')
axes[1].set_ylabel('Disp (m)')
axes[1].set_title('Modal response (combined)')
axes[1].legend(loc='lower right')
axes[1].set_xlim([0,tMax])
axes[1].grid()

plt.show()

print(" \n Here, Upper Graph ")
```

Output is



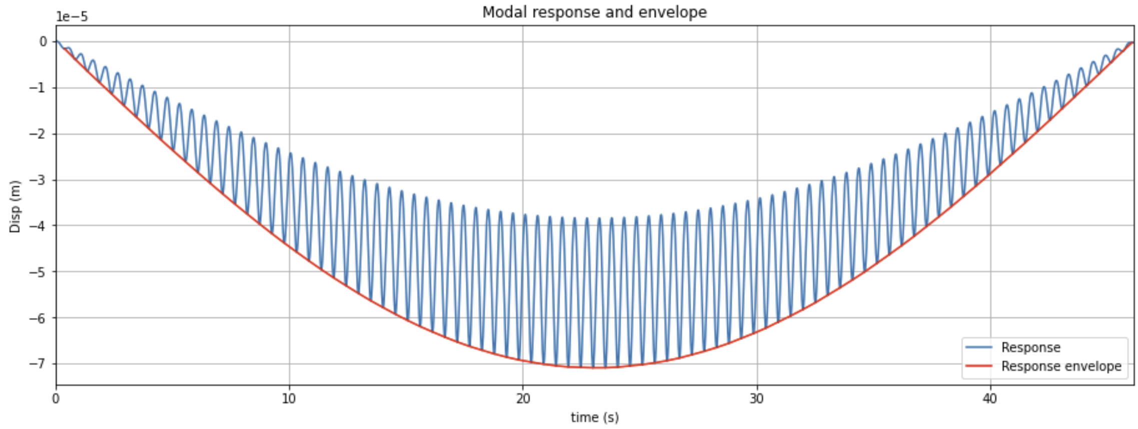
It will help this to more clearly visualise the response if here write a quick function to plot the displacement envelope. This will be particularly helpful later on when our response is derived from the influence of many pedestrians walking simultaneously. Here it will do this by cycling through the displacement record and extracting the oscillation peaks. A peak will be identified when the slope of the displacement changes sign from positive to negative.

This won't be a perfect strategy as it will also identify **'minor'** peaks due to frequencies other than the dominant oscillation frequency but it will be close enough for visualisation purposes. Again, we'll package the code into a function that can be easily called multiple times.

```
def Peaks(disp, time):
    #Initialise containers to hold peaks and their times
    peaks = np.empty([1,0])
    times = np.empty([1,0])

    #Calculate slopes for each data point
    slopes = np.zeros(len(disp))
    for i, u in enumerate(disp):
        if(i<len(disp)-1):
            slopes[i] = disp[i+1]-disp[i]

    #Cycle through all slopes and pick out peaks
    for i, s in enumerate(slopes):
        if (i<len(slopes)-1):
            if(slopes[i+1]<0 and slopes[i]>0):
                peaks = np.append(peaks,disp[i])
                times = np.append(times,time[i])

    return [peaks, times]
```

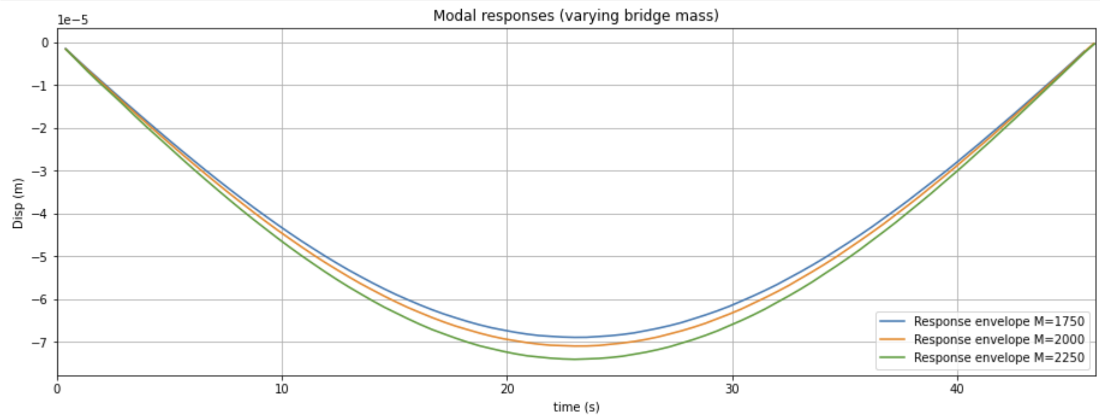
```
peaks, times = Peaks(response,time)
```

Now it can plot the response again but also plot an envelope line that spans between oscillation peaks. So that the graph plot will be as per the previous concepts



Now that it has a neater way of visualising the response, i.e. the response envelope, it can run a number of simulations to visualise the influence of any one of our simulation parameters. Let's consider the influence of bridge mass per unit length for example. Thus, start by calculating the bridge stiffness based on our original mass and frequency parameters and then just loop through the different masses.

```
k = m*wn**2 #(N/m) Original system stiffness
Masses = [1750, 2000, 2250] #(kg/m) Masses per unit length to test

fig = plt.figure()
axes = fig.add_axes([0.1,0.1,2,1])

for M in Masses:
    m = 0.5*M*L #(kg) Modal mass of mode 1
    wn = math.sqrt(k/m)
    wd = wn*math.sqrt(1-xi**2) #(rads/s) Damped angular modal frequency

    response = Duhamel(time, Fn) #Response calculated using the Duhamel integral function
    peaks, times = Peaks(response,time)

#     axes.plot(time,response,'-', label=f'Response M={M}')
    axes.plot(times,-peaks,'-', label=f'Response envelope M={M}')

#Housekeeping
axes.set_xlabel('time (s)')
axes.set_ylabel('Disp (m)')
axes.set_title('Modal responses (varying bridge mass)')
axes.legend(loc='lower right')
axes.set_xlim([0,tMax])
plt.grid()
plt.show()

print(" This is the Response Curve with Three Diffrent Situation ")
```



As it might expect, as the mass increases the modal frequency reduces bringing it closer to the pacing frequency. This obviously leads to larger amplitude oscillations as the frequency ratio approaches one. This could repeat this process and test the sensitivity of our response to any of the input parameters – the process is more or less the same. Instead here will do something a little more interesting and simulate the influence of multiple pedestrians crossing the bridge at the same time.

12. Dynamic Analysis - Bridge + N Pedestrian

One of the key assumptions in this project is that there is no interaction between pedestrians (i.e. no traffic effects) and that here is no human-structure interaction. As here are said already, this greatly simplifies the simulation task because now we can rely on superposition. So, to simulate the dynamic influence of a pedestrian crowd all it needs to do is superimpose the responses generated by N pedestrians walking individually across the bridge at different points in time. For this simulation, it will generate normally distributed random walking velocities. It could also generate random body-weights, in fact we could randomise any of the pedestrian parameters. Thus stick with walking speed for now as this demonstrates the concept and has the biggest impact.

Here theoretical approach will be to loop through each pedestrian, generate their GRF and simulate the response of the structure. After all responses have been calculated, superimpose them to determine the overall structural response during the simulation window.

Let's start by assuming that N pedestrians cross the bridge in a 30 minute window. For simplicity we'll assume that the arrival times (onto the bridge) are uniformly distributed in this window. An alternative would be to assume a Poisson distribution of arrival times. This is something you can look into if developing a more rigorous simulation.

```
N = 100 #Number of pedestrians that cross the bridge in the time window
window = 10*60 #(s) #Simulation window
buffer = 200 #(s) Additional seconds to allow simulation of response beyond window length (late finishers)
mp = 65 #(kg) AS per Indian Avg. Padestrian Weight
G = 9.81*mp #(N) Static weight of pedestrian
```

Next it can use **Numpy** Library to generate N random start times and walking velocities.

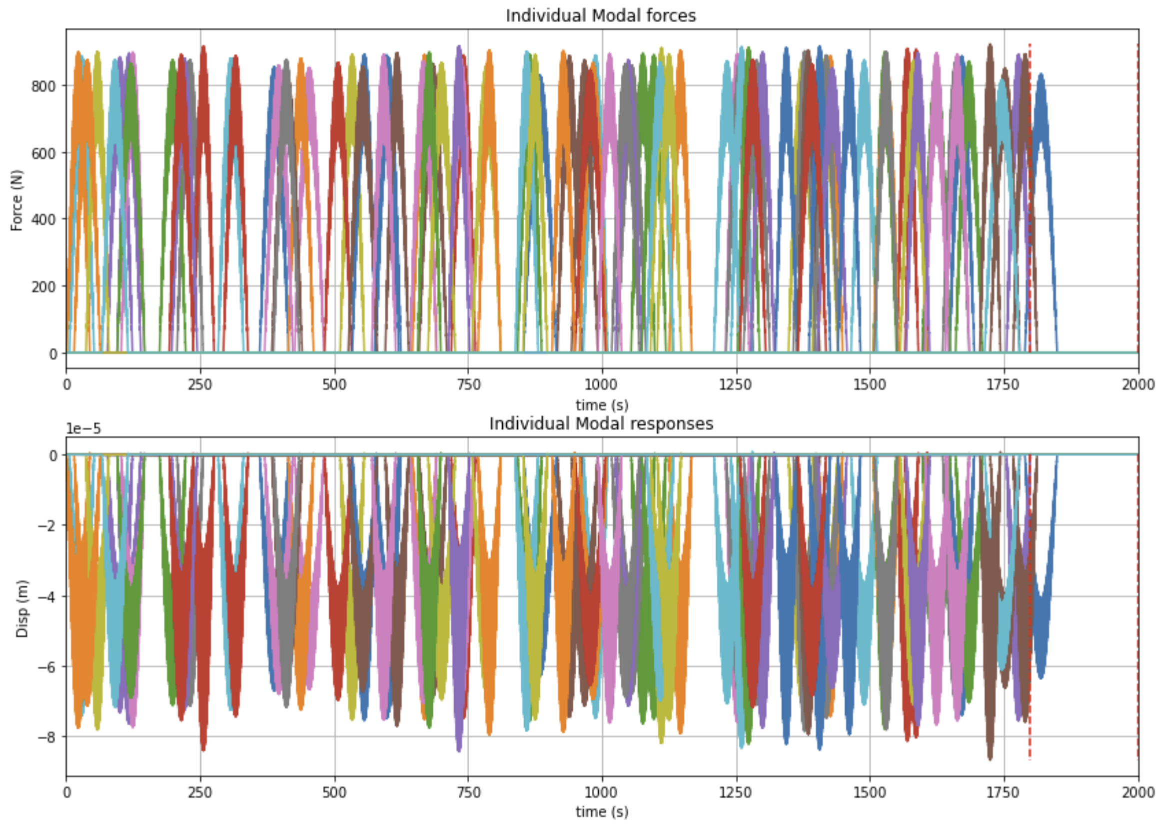
```
#Random variables
tStart = np.random.uniform(low=0.0, high=window, size=N) #Uniformly distributed start times
Vp = np.random.normal(loc=1.3, scale=0.125, size=N) #Normally distributed walking velocities
```

Now it can loop through the crowd and calculate the modal force and response generated by each pedestrian. We'll save the force and response for each person as a separate row in a 2D matrix for forces and responses. Note that this is not the most efficient way of running this simulation as it needs to call our Duhamel function N times. It would be more efficient to calculate the combined crowd force and call the Duhamel function once. But coding the less efficient way preserves the individual pedestrian responses. This is a matter of personal preference. If your simulation is taking too long to run, just reduce the duration or number of pedestrians (helpful while developing the code). But needs more precision.

```
#Set up the simulation time vector
tMax = window + buffer #(s) Max time
time = np.arange(0, tMax+delT, delT)

#Initialise containers to hold the individual forces and responses calculated for each pedestrian
crowdForce = np.zeros([N,len(time)])
crowdResponse = np.zeros([N,len(time)])

#For each pedestrian...
for i, n in enumerate(np.arange(N)):
    vp = Vp[i] #(m/s) Walking velocity
    startTime = tStart[i] #(s) Start time
    tCross = L/vp #(s) Crossing time
    tEnd = startTime + tCross #(s) Finish time

    fv = 0.35*vp**3 - 1.59*vp**2 + 2.93*vp #(Hz) Pacing frequency
    DLF = 0.41*(fv-0.95) #Dynamic load factor

    timeVector = np.arange(0, tCross+delT, delT) #Time vector for this pedestrian
    Fv = G + abs(G*DLF*np.sin(2*math.pi*(fv/2)*timeVector)) #Static + Dynamic GRF (ignore static component)

    xp = vp*timeVector #Position as a function of time
    phi = np.sin(math.pi*xp/L) #Mode shape at pedestrian's location
    Fn = Fv*phi #Modal force experienced by SDoF system

    response = Duhamel(timeVector, Fn) #Response calculated using the Duhamel integral function

    #Save the GRF and response for this pedestrian at the correct position in the overal simulation records
    iStart = round(startTime/delT) #Index for start time
    crowdForce[i, iStart:iStart+len(Fn)] = Fn
    crowdResponse[i,iStart:iStart+len(Fn)] = response
```

Now this has the modal force and response for each individual pedestrian shifted along the time axis to match their random start time. Now plot these to get a picture of what's in front.

```
fig, axes = plt.subplots(figsize=(14,10),nrows=2,ncols=1)

for i in np.arange(len(crowdForce)):
    axes[0].plot(time,crowdForce[i,:],'-')
    axes[1].plot(time,-crowdResponse[i,:],'-')

#Housekeeping
axes[0].plot([window, window],[0,np.max(crowdForce)],'r--')
axes[0].plot([window+buffer, window+buffer],[0,np.max(crowdForce)],'r--')
axes[0].set_xlabel('time (s)')
axes[0].set_ylabel('Force (N)')
axes[0].set_title('Individual Modal forces')
axes[0].set_xlim([0,tMax])
# axes[0].set_xlim([startTime,startTime+tCross])
axes[0].grid()

axes[1].plot([window, window],[0,-np.max(crowdResponse)],'r--')
axes[1].plot([window+buffer, window+buffer],[0,-np.max(crowdResponse)],'r--')
axes[1].set_xlabel('time (s)')
axes[1].set_ylabel('Disp (m)')
axes[1].set_title('Individual Modal responses')
axes[1].set_xlim([0,tMax])
# axes[1].set_xlim([startTime,startTime+tCross])
axes[1].grid()

plt.show()
```



Now this can calculate the combined crowd-induced loading and response simply by superimposing the data for each pedestrian individually.

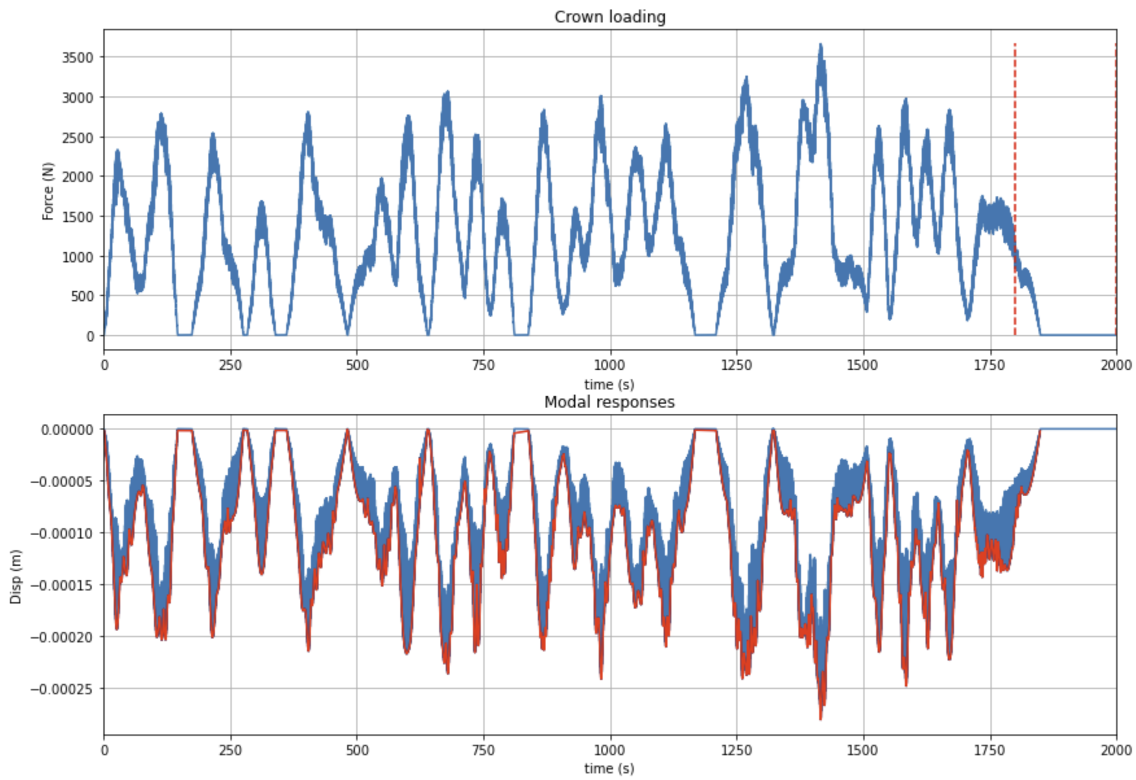
```
#Sum across rows of crowdForce and crowdResponse
F_Crowd = sum(crowdForce)
Res_crowd = sum(crowdResponse)
```

Now this can also make use of our Peaks function to extract the response envelope.

```
peaks, times = Peaks(Res_crowd,time)
```

Finally, it can plot the crowd loading and total modal response. Remember, this is the modal response so it corresponds to the response at the mid-span location of the beam. The response at any other point along the beam can be obtained directly from this by scaling using the mode shape of ϕ.

```
fig, axes = plt.subplots(figsize=(14,10),nrows=2,ncols=1)

axes[0].plot(time,F_Crowd,'-')
axes[1].plot(time,-Res_crowd,'-')
axes[1].plot(times,-peaks,'r-')

axes[0].plot([window, window],[0,max(F_Crowd)],'r--')
axes[0].plot([window+buffer, window+buffer],[0,max(F_Crowd)],'r--')
axes[0].set_xlabel('time (s)')
axes[0].set_ylabel('Force (N)')
axes[0].set_title('Crown loading')
axes[0].set_xlim([0,tMax])
axes[0].grid()

axes[1].set_xlabel('time (s)')
axes[1].set_ylabel('Disp (m)')
axes[1].set_title('Modal responses')
axes[1].set_xlim([0,tMax])
axes[1].grid()

plt.show()
```



Now that this has the crowd-induced dynamic displacement, it could numerically differentiate this vector of numbers twice to obtain the vertical acceleration of the bridge/beam. This would be a good next step as vibration serviceability limits are usually reported in terms of acceleration. Thus skip this step and move on to building an animation of our simulation.

13. Animation Concepts of Bridge Response

Building an animation to visualise the output of our simulation is usually a worthwhile step because it helps communicate what our simulation is telling us, in a more digestible format. This is particularly the case if you need to communicate to a non-expert audience; and it also just looks good! This is very much an optional step.

Thus it start by importing some animation functionality from **matplotlib** library which is famous among MATLAB users also and changing the plotting mode for Jupiter notebook env,

```
from matplotlib.animation import FuncAnimation
import matplotlib.gridspec as gridspec
%matplotlib notebook
```

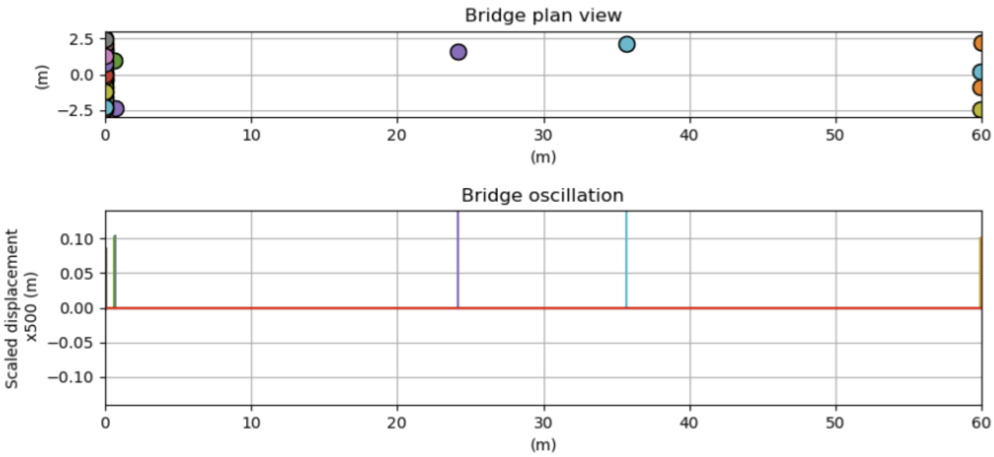
Next it can set up some animation parameters and set a scale factor for deflections so it can easily see them. This can set the duration of your animation to match the simulation length. But be warned, generating the animation make take quite a while depending on computer hardware. It may be better to simulate a window of time within the simulation. Here, thus animate the first **100 seconds**. Frame rate, or the number of frames (images) generated per second also has a **big impact on time taken**. The lower this number, the choppier the animation but the faster it runs and vice-versa.

```
#Animation parameters
animLength = 100 #(sec)
frameRate = 12 #(5,10,20) frames per second (too high and animation slows down)
plotInterval = 1/frameRate #(sec) time between frame plots
dataInterval = int(plotInterval/delT) #Plot moving elements every 'dataInterval-th' point
defScale = 500 #Scale factor on bridge deflection (for visibility)
```

Next, it defines the axes and add some subplots. The top sub-plot will be used to visualise our pedestrians walking across the bridge, viewed from above - a plan view. The bottom subplot will be used to visualise the oscillating beam. We'll also define a sensible set of limits for the x and y axes of both plots.

```
fig, (ax1,ax2) = plt.subplots(nrows=2,ncols=1, figsize=(10, 5)) #Define figure and subplots
gs = gridspec.GridSpec(2,1,height_ratios=[1,1]) #Control subplot layout
ax1 = plt.subplot(gs[0])
ax2 = plt.subplot(gs[1])

ax1.set_aspect('equal', adjustable='box') #Set equal scale for axes top subplot

#Set axis limits
ax1.set_xlim([0,L])
ax1.set_ylim([-3, 3])
yLim2 = defScale*max(Res_crowd)
ax2.set_xlim([0,L])
ax2.set_ylim([-yLim2, yLim2])

#Housekeeping
ax1.set_title('Bridge plan view')
ax1.set_xlabel('(m)')
ax1.set_ylabel('(m)')
ax1.grid()

ax2.set_title('Bridge oscillation')
ax2.set_xlabel('(m)')
ax2.set_ylabel(f'Scaled displacement\n x{defScale} (m)')
ax2.grid()

plt.show()
```

Now it can define the initial state of our plots. This simply requires to draw everything at time zero. This can then return to these elements and update them inside the animation loop



Then,

```
#Define initial state of pedestrians in top plot
topPedList = [] #Initialise an empty list to hold markers representing pedestrians
for i in np.arange(N):
    yPos = np.random.uniform(low=-2.5, high=2.5, size=1) #Random positions across bridge deck width
    pedTop, = ax1.plot(0,yPos,'o', markeredgecolor='k', markersize=10)
    topPedList.append(pedTop)

#Define initial state of pedestrians in bottom plot
btmPedList = []
for i in np.arange(N):
    ped, = ax2.plot([0,0],[0,0.6*yLim2])
    btmPedList.append(ped)

#Define the initial state of the beam in the bottom plot
xVals = np.arange(0,L+1,1) #An array of x-values along the beam
phiVals = np.sin(math.pi*xVals/L) #Corresponding y-values
beamDisp = 0*phiVals #Initial array of displacements along the beam

axisLine, = ax2.plot(xVals,beamDisp,'k') #Add a horizontal beam axis to plot
defLine, = ax2.plot(xVals,beamDisp,'r') #Add initial deflected shape to plot
```

Next it define the animation function that will perform the updates for this project. Here it has updating the pedestrian positions along the bridge in the top and bottom view. For the plan view it randomly assign positions along the width of the bridge deck. It is being updating the deflected shape of the beam in the bottom plot.

To give a sense of the dynamic load imposed by each pedestrian thus scale the height of the pedestrian in the bottom plot to represent the oscillating modal load applied by that pedestrian. As it will be using the modal load, the height scaling will become most obvious around the mid-span of the beam—which seems fairly intuitive as this is where the pedestrians are having most dynamic influence.

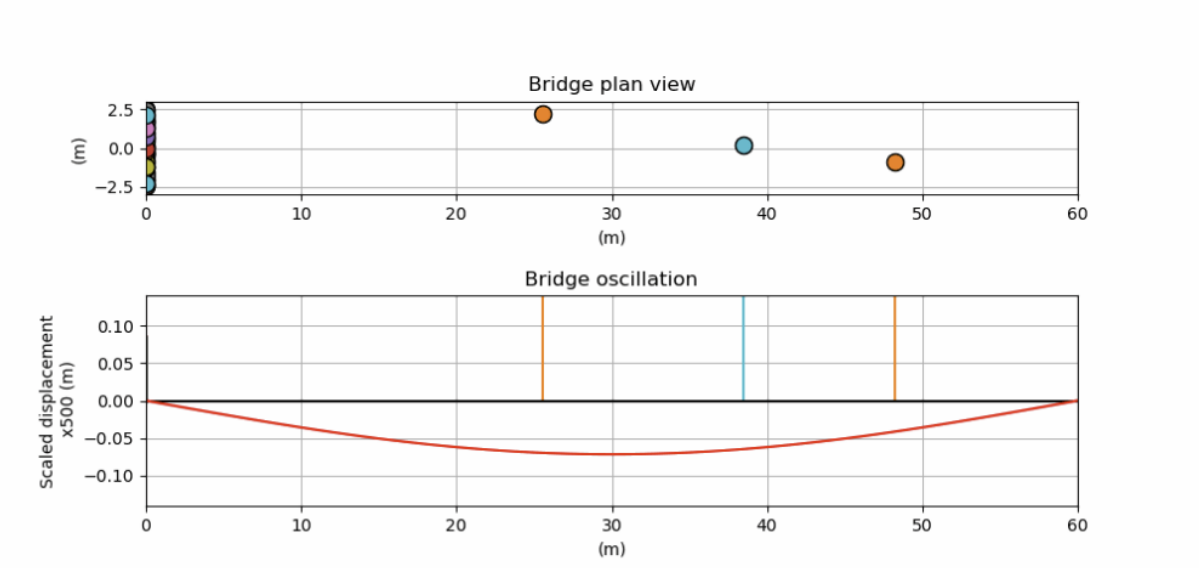
```
#Function to animate plot objects
def animate(i):
    frm = int(i*dataInterval) #Index of data for this frame
    simTime = time[frm] #Simulation time for this animation frame

    #Update the pedestrian positions (top plot) for the current frame
    for i in np.arange(N):
        if(simTime>tStart[i] and simTime<tStart[i] + L/Vp[i]):
            Pt = topPedList[i]
            pos = (simTime - tStart[i])*Vp[i]
            Pt.set_xdata([pos, pos])


    #Update the beam deflected shape for the current frame
    defLine.set_data(xVals, -defScale*phiVals*Res_crowd[frm])

    #Update the pedestrian positions (bottom plot) for the current frame
    for i in np.arange(N):
        if(simTime>tStart[i] and simTime<tStart[i] + L/Vp[i]):
            Pb = btmPedList[i]
            pos = (simTime - tStart[i])*Vp[i]
            h = 0.1 + 0.1*crowdForce[i,frm]/max(crowdForce[i,:])
            Pb.set_data([pos, pos],[0,h])

    return
```

Finally call the animation function and **save the resulting animation to a GIF file in local**. It will be easier to view rather than trying to watch the animation play back from the function call.

- **After the animation GIF has been generated, it can drag it back into the notebook and run it again so that the simulation will work smoothly.**

```
#Function to generate the animation
myAnimation = FuncAnimation(fig,
                            animate,
                            frames=int(1 + (animLength/plotInterval) ),
                            interval=plotInterval*1000,  #milliseconds
                            blit=True,
                            repeat=True)

plt.show()
myAnimation.save('Bridge_response.gif')
```

It is possible to embed an animated GIF in any PDF/word however it will be a still photo once saved. To keep the animation, you will need to use tools like PDF/word editors to be able to play it as it is once saved as PDF/word. So I am displaying the still version of the simulation.



14. Simulation Limitation

This is the development on this project. This is a good time to review the limitations of this simulation but also how what this project taught us can be generalised to the dynamic analysis of other structures and other scenarios.

This project relied pretty heavily on superposition in this analysis. This allowed the project maker to calculate the response generated by each person individually, ignoring interaction between pedestrians and between each pedestrian and the bridge. In many cases this will be acceptable, for example, only when large lateral oscillations occur do we expect to see significant feedback between walking pedestrians and the structures they're walking on, see London Millennium Footbridge *[33]*

So, vertical vibration assessment without interaction modelling is generally acceptable. Also ignored the mass effect of pedestrians on the basis that it was small compared to the bridge mass. If this assumption cannot be made, altering the modal mass to account for pedestrians is quite simple. In any event, the additional mass will just lead to a reduction in the modal frequency. This may or may not be a conservative simplification depending on the relationship between the modal frequency and pacing frequency.

So, unless this is attempting to generate exceptionally accurate simulation results, the type of crowd-induced vibration modelling demonstrated here is likely to suffice. At the very least, it should be sufficient to highlight the potential for serviceability problems in service. This is particularly the case when we consider all of the other additional sources of uncertainty that creep into numerical modelling.

The other big simplification project maker made in this analysis was this bridge model; it was a simply supported beam. This may at first appear like a significant drawback. However, in order to generalise everything thus done here to more complex structures, all it needs to know are the modal properties; frequency and mode-shape. Once these are known for the structure in question, we can directly apply the modal analysis that this demonstrated on a simple beam model.

15. Further Project Prospect

This is the power of modal analysis—it allows us to reduce the dynamic behaviour of complex structures down to the dynamics of a series of Single Degree of systems. This SDoF system response can then be mapped back onto the complex structure via the mode-shape.

1. In order to address the some of the further work, a parametric study of the proposed virtual crowd-structure interaction model can be carried out in cycle and moving portions. More complex structure can be addressed with help of more dynamic response.

2. A much needed bridge has other concepts like Cycling Pedestrian, running pedestrian, etc as classification can be added.

3. Those who are not moving frequently in bridge or acting as deal load stereotype can be added in future.

4. Bridge/Structure Categories or structure state limit can be added in future with dynamic material in study.

16. Reference:

1. Mod-03 Lec-07 Duhamel’s Integrals published by NPTC constructed by IIT,Madras Link : (https://www.youtube.com/watch?v=AVPK8pdWG_o)

2. Note made of Dynamics of Single Ocean Structure by Dr. Srinivasan Chandrasekharan, Link (https://nptel.ac.in/)

3. Paper: Dynamic Response of a Single Degree of Freedom (SDOF) System in some Special Load Cases, based on the Duhamel Integral by Konstantin Savkov Kazakov, Link (https://www.researchgate.net/publication/242521060_Dynamic_Response_of_a_Single_Degree_of_Freedom_SDOF_System_in_some_Special_Load_Cases_based_on_the_Duhamel_Integral)

4. Pedestrian Safety Prediction Methodology (2008), National Academies of Sciences, Engineering, and Medicine. 2008. Pedestrian Safety Prediction Methodology. Washington, DC: The National Academies Press. Link https://nap.nationalacademies.org/read/23083/chapter/6

5. Crowd-induced lateral bridge vibration by Sean P. Carroll, B.Eng (Hons), M.Sc Thesis submitted to The University of Nottingham for the degree of Doctor of Philosophy Link : http://eprints.nottingham.ac.uk/28410/1/594602.pdf

6. http://commons. wikimedia.org.

7. P. Dallard, A.J. Fitzpatrick, A. Flint, S. Le Bourva, A. Low, R.M. Ridsdill Smith, and M. Willford. The london millennium footbridge. The Structural Engineer, 79: 17-33, 2001.

8. E.T. Ingolfsson, c.T. Georgakis, and J. Jonsson. Pedestrian-induced lateral vibrations of footbridges: A literature review. Engineering Structures, 45:21-52,2012.

9. S. Nakamura. Field measurements of lateral vibration on a pedestrian suspension bridge. The Structural Engineer, 81:22-6, 2003.

10. S. Nakamura and T. Kawasaki. Lateral vibration of footbridges by synchronous walking. Journal of Constructional Steel Research, 62:1148-1160,2006.

11. J.M.W. Brownjohn, P. Fok, M. Roche, and P. Omenzetter. Long span steel pedestrian bridge at singapore changi airport - part 2: Crowd loading tests and vibration mitigation measures. The Structural Engineer, 82:28-34, 2004.

12. A. Ronnquist. Pedestrian induced lateral vibrations of slender footbridges. PhD thesis, Norwegian University of Science and Technology, 2005.

13. A. McRobie, G. Morgenthal, J. Lasenby, and M. Ringer. Section model tests on human-structure lock-in. Bridge Engineering, 156:71-79, 2003.

14. C.H. Butz, C.H. Heinemeyer, A. Goldack, A. Keil, M. Lukic, E. Caetano, and A. Cunha. Advanced load models for synchronous pedestrian excitation and optimised design guidelines for steel footbridges. Technical report, Office for Official Publications of the European Communities, 2008. 241 REFERENCES

15. E.T. Ingolfsson, c.T. Georgakis, F. Ricciardelli, and L. Proncino. Lateral humanstructure interaction on footbridges. In Recent Advances in Structural Dynamics, Southampton, 2010.
16. E.T. Ingolfsson, C.T. Georgakis, F. Ricciardelli, and J. Jonsson. Experimental identification of pedestrian-induced lateral forces on footbridges. Journal of SOllnd and Vibration, 330:1265-1284, 2011.

17. V. Racic, J.M.W. Brownjohn, and A. Pavic. Reproduction and application of human bouncing and jumping forces from visual marker data. Journal of Sound and Vibration, 329:3397 - 3416, 2010.

18. A.L. Hof, R.M. van Bockel, T. Schoppen, and K. Postema. Control of lateral balance in walking: Experimental findings in normal subjects and above-knee amputees. Gait & Posture, 25:250-258, 2007.

19. A.L. Hof, S. Vermerris, and W. Gjaltema. Balance response to lateral perturbations in human treadmill walking. The Journal of Experimental Biology, 213:2655-2664, 2010. [15] M. Bocian, J.H.G. Macdonald, and J.P. Bum. Biomechanically inspired modelling of pedestrian-induced forces on laterally oscillating structures. Journal of SOlmd and Vibration, 331:3914-3929, 2012.

20. Y.P. Ivanenko, R.E. Poppele, and F. Lacquaniti. Motor control programs and walking. The Neuroscientist, 12:339-348,2006. [17] E. Otten. Balancing on a narrow ridge: biomechanics and control. Philosophical Transactions of the Royal Society B: Biological Sciences, 354:869-875, 1999.

21. M.W. Whittle. Gait Analysis: An Introduction. Butterworth Heinemann, Edinburgh, 2004.

22. V.T. Inman, H.J. Ralston, and F. Todd. Human Walking. Williams and Wilkins, Baltimore, 1981.

23. J. Hammil and K.M. Knutzen. Biomechanical basis for human movement, 3rd Edn. Lippincott Williams and Wilkins, 2009. 242 REFERENCES [21] e.L. Vaughan, B.L. Davis, and J.e. 0' Conner. Dynamics of Human Gait. Kiboho Publishers, Cape Town, South Africa, 1999.

24. P. Charles and V. Bui. Transversal dynamic actions of pedestrians & synchronisa, tion. In Proceedings of Footbridge 2005, second international conference, Venice, 2005.

25. NumPy Action based on the information of NumPy website Link : https://numpy.org/learn/

26. UserGuide and Tutorial based on Visualisation of Backends and Python Integration, Link: https://matplotlib.org/stable/users/index

27. A. Low. Footbridge Vibration Design, chapter Design for dynamic effects in long span footbridges, pages 37-52. CRC Press, 2009.

29. S.C. Kerr, Human Induced Loading on Staircases, Ph.D. Thesis, University College London, Mechanical Engineering Department, London, UK, 1998.

29. F. Bruno, L.and Venuti. Crowd-structure interaction in footbridges: Modelling, application to a real case-study and sensitivity analyses. Journal of Sound and Vibration, 323:475–493, 2009.

30. J. E. A. Bertram and A. Ruina. Multiple walking speed-frequency relations are predicted by constrained optimization. Journal of Theoretical Biology, 209:445–453, 2001.

31. A. Pachi and T. Ji. Frequency and velocity of people walking. The Structural Engineer, 83:36–40, 2005.

32. S.P. Carroll, J.S. Owen, M.F.M. Hussein. Modelling crowd-bridge dynamic interaction with a discretely defined crowd. The Journal of Sound and Vibration, 331: 2685-2709, 2012

33.P. Dallard, A.J. Fitzpatrick, A. Flint, S. Le Bourva, A. Low, R.M. Ridsdill Smith, and M. Willford. The london millennium footbridge. The Structural Engineer, 79: 17–33, 2001.

34. LalMandi Footbridge by Wikipedia , Link https://en.wikipedia.org/wiki/Lal_Mandi_Footbridge.