5-2020

The Development and Evaluation of a New Analytical Model for the Characterization of Friction Induced Disc Brake Vibration

Michael Vladimirov

THE UNIVERSITY OF NEW HAVEN

THE DEVELOPMENT AND EVALUATION OF A NEW ANALYTICAL MODEL FOR THE CHARACTERIZATION OF FRICTION INDUCED DISC BRAKE VIBRATION

A THESIS

submitted in partial fulfillment

of the requirements for the degree of

MASTER OF SCIENCE MECHANICAL ENGINEERING

BY

Michael Vladimirov

University of New Haven
West Haven, Connecticut
May 2020

THE DEVELOPMENT AND EVALUATION OF A NEW ANALYTICAL MODEL FOR THE CHARACTERIZATION OF FRICTION INDUCED DISC BRAKE VIBRATION

APPROVED BY:

Eric Dieckman, Ph.D.
Thesis Adviser

Samuel Bogan Daniels, Ph.D.
Committee Member

Ravi Gorthala, Ph.D.
Committee Member

Eric Dieckman, Ph.D.
Program Coordinator

Ronald S. Harichandran, Ph.D., P.E., F.ASCE
Dean of Tagliatela College of Engineering

Mario Thomas Gaboury, Ph.D.
Provost

Acknowledgements

I would like to thank the faculty of the Mechanical Engineering Department at the University of New Haven. In particular, I would like to express my gratitude to Dr. Eric Dieckman for inspiring and supporting my curiosity during my MS studies, as well as for helping to guide me through the development of this research. His mentorship has had a resounding impact on my career, both in industry and in academia.

Abstract

This research proposes a new method of modeling vibration of disc brake systems. This modeling method treats the brake pads of a disc braking system as lumped spring-mass-damper subsystems and the disc as a continuous bar. This modeling concept increases both the ease of analytical expression and the insight provided into the behavior of the real-world systems. This paper describes the analytical expression of this model and its implementation in Matlab. A parameter variation study yielded results that agree with existing research, which serves to validate the fundamental principles of this proposed modeling method.

Table of Contents

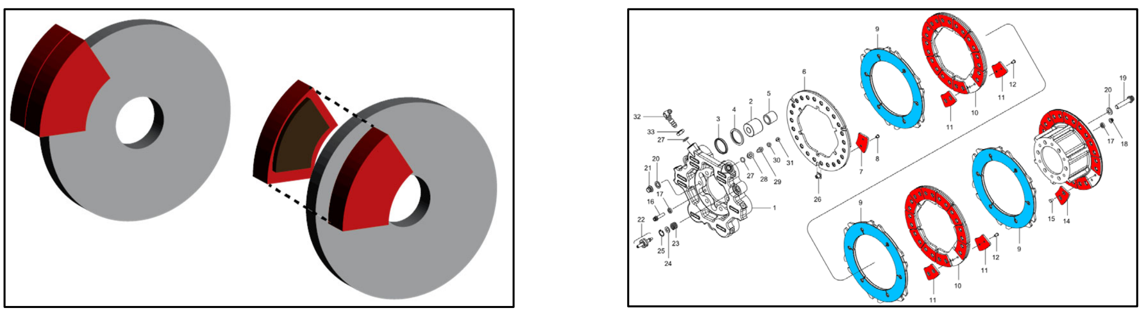
List of Tables

List of Figures

Introduction

Brake noise is an acoustic phenomenon that affects all forms of wheeled transportation. It is caused by vibrations that occur when the static portion of a brake assembly makes contact with the moving portion. In contemporary braking systems this vibration occurs when brake pads or stators contact the discs or rotors (Figure 1). It can be categorized in multiple ways that often correlate to specific frequency ranges (Figure 2) and has generally been considered a significant nuisance [1].

High and low frequency "squeal" observed in both contemporary automotive and aerospace braking systems. This type of brake noise occurs in the highly audible frequency range of 1-10 kHz and is caused by stick-slip motion that occurs between brake pads or stators and discs or rotors [2].



(a) (b)

Figure 1 – Components of typical brake assemblies, automotive (a) and aerospace (b). Automotive brake assemblies are typically composed of a single pair of brake pads that grip the rotor. Brake assemblies for business class and passenger aircraft employ multi rotor and multi brake pad (highlighted in blue and red, accordingly in b). The aerospace brake assembly shown here is used on aircraft such as the Pilatus PC-12 [21,22].

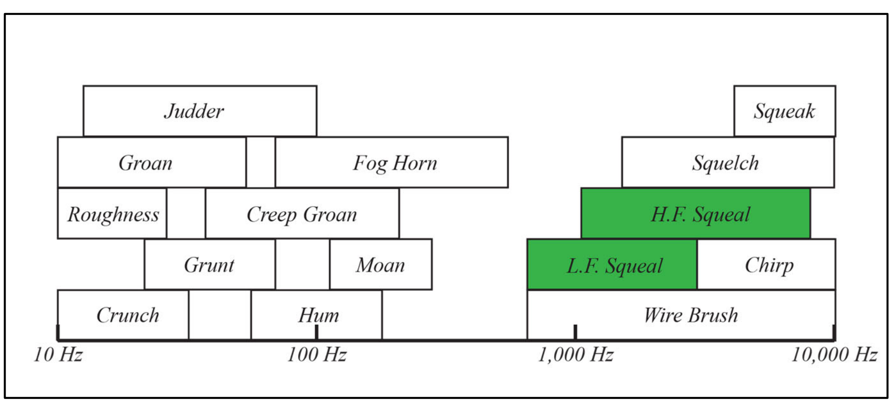
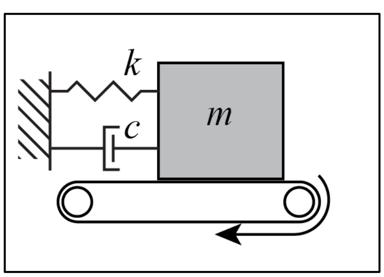


Figure 2 – Common categorization of brake noise and their approximate frequency ranges, with squeal marked in green [1].

Stick-slip motion can be generalized as periodic motion observed in elastic systems that are driven through frictional contact with a moving body. This behavior can be observed in the system shown in Figure 3. Assuming the spring is relatively weak and initially at its equilibrium position, the mass will move together with the treadmill, by virtue of the static friction between the mass and the treadmill. As the mass is displaced, the spring force acting on the spring increases. Once the magnitude of the spring force exceeds the magnitude of the force of static friction, the mass will begin to move in the direction opposite of the motion of the treadmill, towards its initial position. As this happens the spring force will decrease, and the mass will also be acted on by a kinetic friction force. At a certain point, the magnitude of the spring force will become smaller than that of the force of kinetic friction, which will cause the mass to cease moving in the direction opposite the treadmill's direction, and the mass will once again begin to move in the direction of the treadmill. This cycle repeats periodically as long as the treadmill is moving, with a period inversely proportional to the stiffness of the spring [3].

Systems exhibiting stick-slip motion can be considerably more complex. For example, stick-slip motion can act as a source of forced vibration to other areas of systems (Figure 3b). Similarly, stick-slip motion can often be observed in continuous systems, such as bowed musical instruments [4]. In certain cases, considering systems as being continuous, rather than in terms of lumped-mass models, can help to expose highly complex behaviors [5].



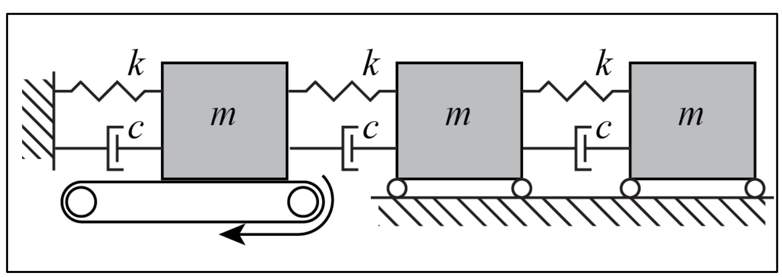


Figure 3 – Examples of systems with stick-slip motion. A simple sprung mass with rough surfaces on a rough moving treadmill (a). A more complex system in which stick-slip motion instigates the vibration other degrees of freedom of the system (b) [23].

Since the development and commodification of disc brakes, considerable efforts have been put forth to develop accurate mathematical models of brake squeal. The earliest work in this field comes from Jarvis and Mills [6] who used a pin-disk model to approximate an automotive disc brake assembly (Figure 4a). Their work proved the viability of accurately predicting unwanted vibration in an automotive braking system.

Jarvis and Mills' pin-disk model was used by Spurr to develop a cohesive theory on the role of stick-slip motion in the generation of brake squeal [7]. Spurr determined that brake squeal is the result of both stick-slip motion between the brake rotors and stators and a modified motion which he called "sprag-slip". Spurr defined spragging as the phenomenon in which stators lock in a fairly rigid manner, causing an abrupt reduction of momentum of a rotor (Figure 4b). Further studies determined that geometrical modifications to stator geometry, as well as to force application on the stator, can reduce spragging [2].

Earles et al. [8] went on to extensively elaborate on the work of Spurr and developed the first lumped mass models of pin-disk models (Figure 4c), which yielded reasonably accurate prediction of unstable disc brake vibration. Rather than considering a disc brake assembly in terms of two separate continuous components, Earles et al. approximated the dynamics of a stator in terms of three degrees of freedom and the dynamics of a rotor in terms of only a single degree of freedom. Subsequent research proved the merits of this modeling technique, but also repeatedly demonstrated that significant levels of in-plane vibrations occur during instances of brake squeal [9,10].

With time, this topic was pursued by other researchers who found methods of increasing model accuracy by identifying additional degrees of freedom that could be introduced. Notable efforts of these types were carried out by North, who introduced an 8 degree of freedom model [11], and Millner, who introduced a 6 degree of freedom model [12]. The efforts of Rudolph and Popp yielded a 14 degree of freedom model, which took into consideration a considerably larger

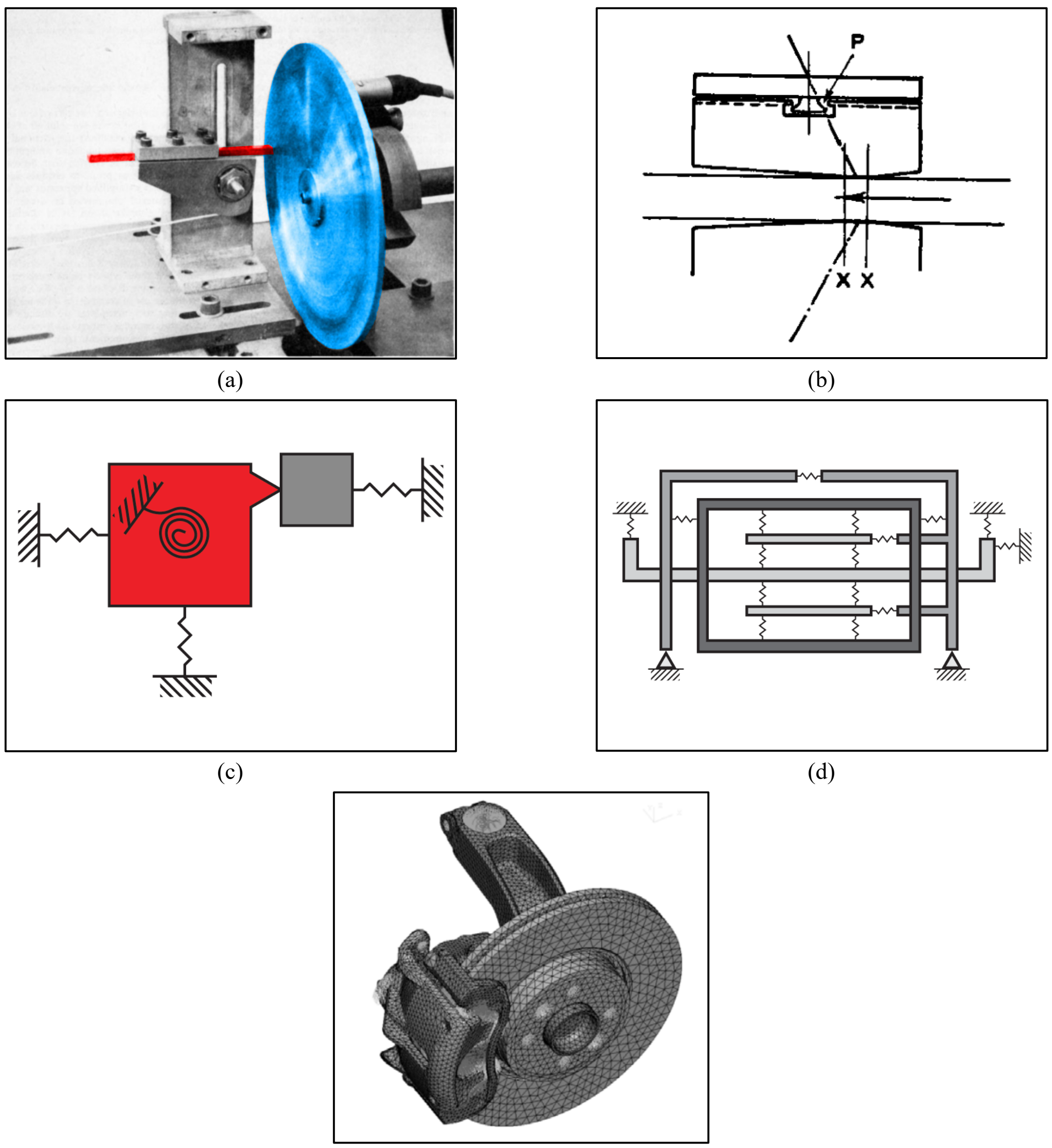


Figure 4 – Modeling methods that have been used to study disc brake squeal. Jarvis and Mills' experimental pin-disc model [4] pin in highlighted in red and disc highlighted in blue (a). Cross section of brake pads (stators) spragging against a disc (rotor)(b) [7]. *The lumped mass model of Earles et al. (c) The 14 degree of freedom model of Rudolph and Popp. (d) Finite element model of a disc brake assembly used for the study of mode coupling and damping (e)* [19].

number of the structural components of a disc brake assembly (Figure 4d) [2]. Similar efforts, in which a larger number of structural components are considered can be found in studies related to aerospace brake assemblies [13].

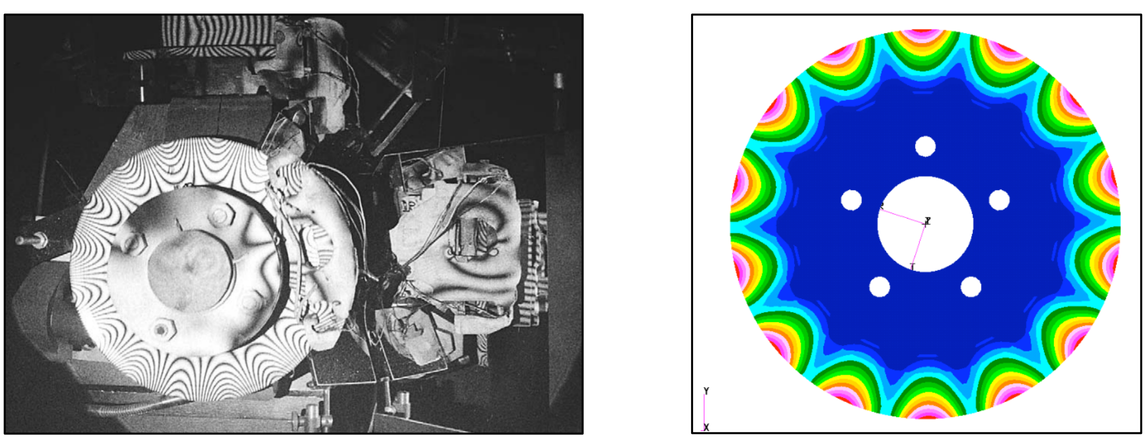
The commonality of most lumped mass modeling techniques is that complex eigenvalue analysis is used to identify Hopf bifurcation points or combinations of system parameters (e.g. component stiffnesses, damping, mass, friction, braking force) that may cause unstable vibration [2,12].

While these analytical models are useful in predicting the frequencies of vibration in terms of system parameters and determining parameter ranges in which these vibrations become unstable, they cannot provide insight into the continuous in-plane displacements exhibited by discs during vibration, as well as the subsequent associated stresses (Figure 5a). Furthermore, these models offer particularly limited utility in analyses related to contemporary aerospace brakes, which have multiple sets of brake pads applying pressure to discs, requiring extensive simplifying assumptions to be made. Typically, efforts to model squeal in aerospace braking systems will emphasize the mode coupling effects that stick-slip vibration has on components other than rotors and stators. These efforts will apply simplifying assumptions, which approximate multiple stator contact points as a single contact point, and typically make use of a two degree of freedom variant of the treadmill (Figure 3) to induce stick-slip motion in the system, as can be seen in the work of Nechak et al. [14].

Thus, finite element (FE) modeling methodologies of modeling disc brakes (Figure 4e) began to gain traction in the 1990’s and early 2000’s and have become increasingly more commonly used in the modeling of brake assemblies in recent years [15,16]. Academic efforts of this type have primarily focused on the modal analysis of disc brake vibration (Figure 5b), rather

than the study of actual vibration amplitudes and associated stress levels [17]. However, with contemporary FE software solutions, response analyses are relatively trivial, from a procedural standpoint, but not from a computational resource standpoint. The primary advantage of FE modeling for modal analyses is the ease with which it allows for the introduction of different types of damping terms [18,19]. Furthermore, FE analyses greatly reduce the difficulty of studying the effects of complex mode coupling between stators and rotors [20].

While finite element modeling methods have clear advantages over existing lumped mass models, they also have two significant disadvantages. The most obvious of these is that the preparation, validation, and solution of a finite element model is a fairly demanding process that may require considerable time and computational resources, making it an obstacle for product development in an industrial context. Furthermore, finite element modeling requires the availability of an initial geometry, which means that investigations into the vibration characteristics of a brake design can only be carried out after resources have already been invested into design, which has the potential to lead to project management inefficiencies.



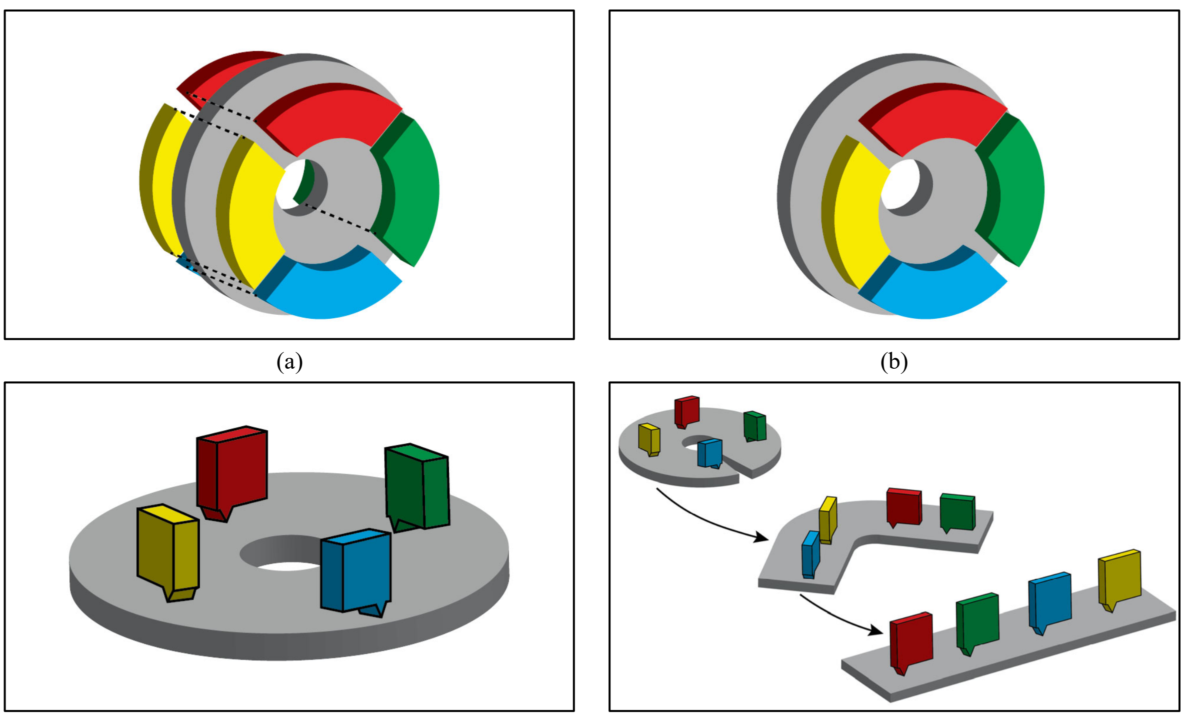
(a) (b)

Figure 5 – Examples of in-plane modal behavior exhibited by brake rotors, captured on an experimental rig with holographic interferometry in the work of Fieldhouse et al [9], *and in the results of a modal FE analysis* [17]

1. New Analytical Model Development

The proposed modeling method begins by considering a generic disc brake assembly abstractly in terms of its key components: multiple brake pads contacting a rotor. Symmetry conditions are applied, allowing for half of the assembly to be neglected. Like the pin-disk models proposed by Earles et al. [8], the brake pads are approximated by lumped masses that contact the disk at an infinitesimally small region.

A cut is then made to the toroidal disk, and it is "straightened" into an elastic bar-like entity (Figure 6d), with the lumped masses tracking this transformation. Periodic boundary conditions are assigned to the two ends of the bar; dynamic characteristics at one end of the bar are forced to be equal to those of the other end. Thus, rotational movement of the toroidal disk can be represented as linear movement of the bar (Figure 6).



(a) (b) (c) (d)

Figure 6 – The development of the proposed two dimensional analytical model begins by considering a disc brake assembly with multiple pairs of pads (b). Symmetry conditions allow for the depairing of pads (b). Single pads can be approximated by lumped masses (c). Periodic boundary conditions allow for the application of a transformation that converts the toroidal disk to a linear bar.

This system is now adequately represented by the 2-dimensional system shown in Figure 7. Movement of the lumped mass pads is defined in two parts: along the x-axis and along the y-axis. For the purpose of simplicity, the following derivation, below, considers only one brake pad (Figure 8).

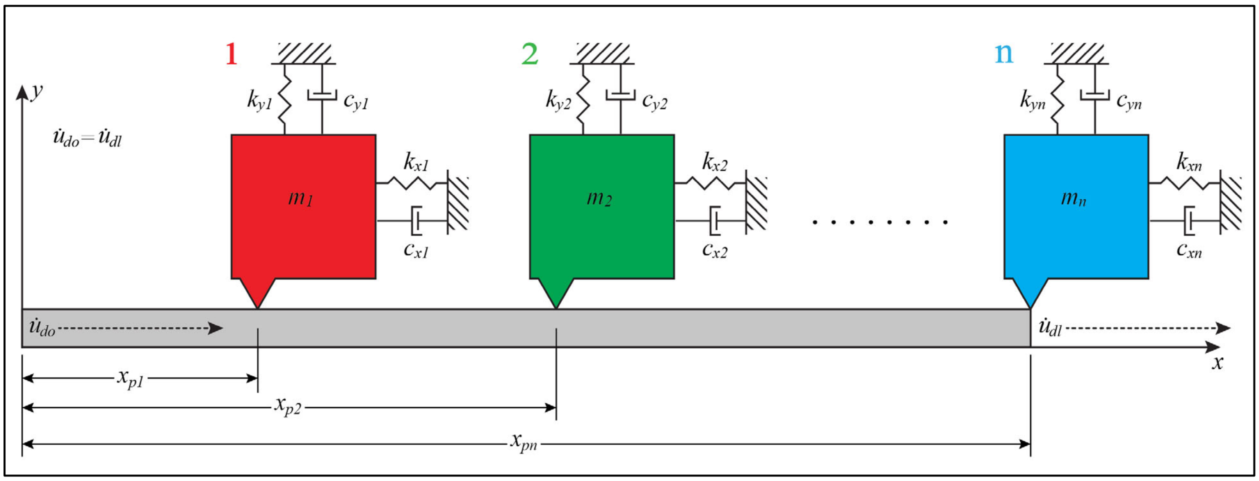


Figure 7 – Proposed two-dimensional hybrid lumped parameter-continuum model

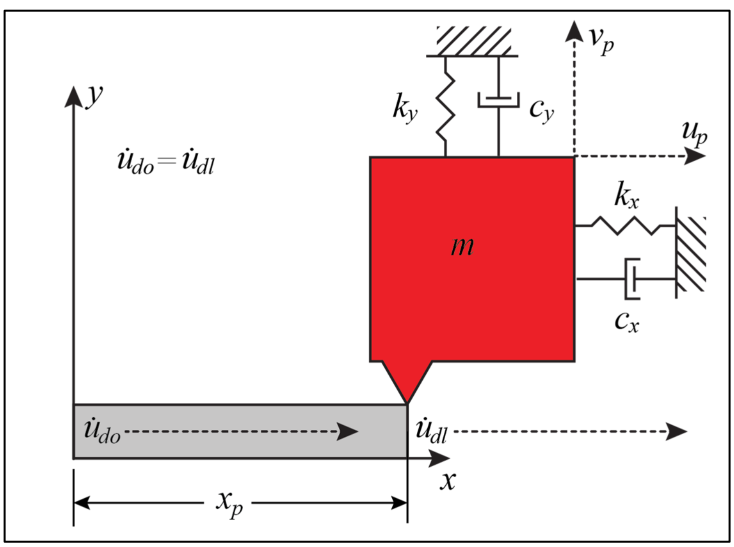


Figure 8 – Proposed two-dimensional analytical modeling methodology applied to a single brake pad pair

1.1. Initial Conditions of the System

The initial conditions of the system occur on a time interval $t_0 \leq t < t_1$, wherein t_0 is some arbitrary time and t_1 is the time at which the brake pads clamp the disc. During this time interval, the disc moves at some horizontal velocity, $\dot{u}_d(t)$. During this time, the deformation of the disc, δ_d, is equal to zero. Furthermore, the brake pads are entirely static, and its horizontal and vertical velocities, $\dot{u}_p(t)$ and $\dot{v}_p$ accordingly, are also equal to zero. Thus, the kinetic energy of the disc brake assembly during this time can be expressed entirely in terms of the mass and velocity of the disc:

$$K(t_0 \leq t < t_1) = \frac{m_d \dot{u}_d^2(t)}{2}. \tag{1}$$

1.2. Definition of Horizontal Stick-Slip Motion of Brake Pads

The lumped masses move along the x-axis in a periodic stick-slip fashion. When elastic reaction force, $F_{spring,x}(t)$, acting on the block (modelled by spring with stiffness k_x in Figure 8) is less than the force of static friction acting on the block, $F_{fs}(t)$, the block will "stick" to the bar and the two will have coupled displacement and velocity. When $F_{spring,x}(t)$ exceeds $f_{\mu_s}(t)$, the block will "slip". The system will cyclically alternate between these stick and slip states.

The elastic reaction force of the pad can be simplistically defined as

$$F_{spring,x}(t) = k_x u_p(t), \tag{2}$$

and the magnitude of static friction is defined in terms of the transient horizontal normal force which presses the pad against the brake pad, F_N, as

$$f_{\mu_s}\big(F_N(t)\big) = \begin{cases} \mu_s F_N(t) & F_N(t) > 0 \\ 0 & F_N(t) \leq 0. \end{cases} \tag{3}$$

This section is divided into three subsections that defines the horizontal motion of the system during the stick phase, the horizontal motion of the system during the slip phase, and the entirety of the cyclical stick-slip motion.

1.3. Definition of Stick Phase Motion

The first cycle of the stick phase occurs on an interval $t_1 < t < t_2$, wherein t_1 is the time at which the brake pad touches the disc and t_2 is the time at which slip phase begins. When the brake pad touches the bar its vertical displacement, $v_p(t)$, is said to be zero. As soon as the brake pad touches the bar, its horizontal velocity, $\dot{u}_p$, will be equal to that of the bar, $\dot{u}_d$. However, the coupling of the mass of the bar and the pad will increase the mass of the moving components of the system, which will cause them to move at a new and reduced velocity, $\dot{u}_c$

$$\dot{u}_p(t_1) = \dot{u}_d(\, t_1) = \dot{u}_c. \tag{4}$$

The value of $\dot{u}_c$ can be determined by considering the kinetic energy of the model at time $t = t_1$

$$K(t_1) = \frac{\left(m_d + m_p\right)\dot{u}_c^2}{2}. \tag{5}$$

Per conservation of energy, Equations 1 and 5 must be equal

$$\frac{m_d \dot{u}_d^2(t_1 - \Delta t)}{2} = \frac{\left(m_d + m_p\right)\dot{u}_c^2}{2}. \tag{6}$$

From Equation 6, it can be inferred that the velocity of the brake pad at t_1 is

$$\dot{u}_d(t_1) = \dot{u}_b(t_1) = \dot{u}_c = \sqrt{\frac{m_d \bullet \dot{u}_d^2(t_1 - \Delta t)}{m_d + m_p}}. \tag{7}$$

Thus, $\dot{u}_c$ is the initial velocity of the pad at the start of the stick phase of the stick-slip cycle.

The coupled disc-bar system will behave as a single degree of freedom oscillator during the stick phase. Newtonian force balance yields an equation of motion for the system

$$(m_d + m_p)\ddot{u}_p + c_x\dot{u}_p + k_x u_p = 0. \tag{8}$$

Substituting

$$\xi_{stick} = \frac{c_x}{2\omega_{stick}(m_d + m_p)}, \quad where\ \omega_{stick} = \sqrt{\frac{k_x}{(m_d + m_p)}}, \tag{9}$$

and rearranging the equation yields

$$\ddot{u}_p + 2\omega_{stick}\xi_{stick}\dot{u}_p + \omega_{stick}^2 u_p = 0. \tag{10}$$

It is assumed that the solution of Equation 10 takes the exponential form

$$u_{p,stick}(t) = Ce^{st}. \tag{11}$$

The substitution of Equation 11 into Equation 10 yields

$$s^2 + 2\omega_{stick}\xi_{stick}s + \omega^2 = 0. \tag{12}$$

The roots of this equation are

$$s = \omega_{stick}\xi_{stick} \pm \omega_{stick}\sqrt{\xi_{stick}^2 - 1}. \tag{13}$$

In this case, the viscoelastic portion of the system consists of a solid elastic material: the brake pad. Thus, it is reasonable to assume that the system will always be underdamped, i.e. $\xi_{stick}^2 < 1$. Under this assumption, the radicand of Equation 13 will be imaginary

$$s = \omega_{stick}\xi_{stick} \pm i\omega_{d,stick}. \tag{14}$$

In Equation 14, $\omega_{d,stick}$ is the damped natural frequency of the system and is defined as

$$\omega_{d,stick} = \omega_{stick}\sqrt{1-\xi_{stick}^2}. \tag{15}$$

Substitution of the roots (Equation 14) into the assumed solution (Equation 11) yields two solutions

$$u_{p,stick}(t) = Ce^{\left(-\omega_{stick}\xi_{stick}\pm i\omega_{d,stick}\right)t}, \tag{16}$$

which yields a general solution that consists of a linear combination of these solutions

$$u_{p,stick}(t) = C_1e^{\left(-\omega_{stick}\xi_{stick}+i\omega_{d,stick}\right)t} + C_2e^{\left(-\omega_{stick}\xi_{stick}-i\omega_{d,stick}\right)t} \tag{17}$$

This general solution can, alternatively be expressed in terms of a trigonometric sum

$$u_{p,stick}(t) = e^{-\omega_{stick}\xi_{stick}t}\left(A_{1,stick}\cos\left(\omega_{d,stick}t\right) + A_{2,stick}\sin(\omega_{d,stick}\,t)\right). \tag{18}$$

The constant terms, $A_{1,stick}$ and $A_{2,stick}$, in Equation 18 are dependent on initial conditions ($u_p(t=1) = u_{p_0}$and $\dot{u}_p(t=1) = \dot{u}_c$, per Equation 7)

$$A_{1,stick} = u_{p_0} \tag{19}$$

$$A_{2,stick} = \frac{\frac{\dot{u}_c}{\omega} + \xi u_{p_0}}{\sqrt{1-\xi^2}}. \tag{20}$$

Thus, displacement of the bar and pad during the stick phase is defined as

$$u_d(t) = u_p(t) = e^{-\omega_{stick}\xi_{stick}t}\left(A_{1,stick}\cos\left(\omega_{d,stick}t\right) + A_{2,stick}\sin\left(\omega_{d,stick}\,t\right)\right)$$
$$for\ f_{\mu_s} > F_{springx}. \tag{21}$$

1.3.1. Definition of Slip Phase Motion

When the magnitude of the reaction force exceeds that of static friction at a time t_2, the brake pad slides back towards its initial position ($x_p = 0$) until time t_3, at which the spring force is less than the friction. Because the mass is damped, this motion will occur in a transient viscously damped harmonic manner, with additional frictional damping.

If friction is temporarily ignored, the derivation of the definition of the brake pad's motion is almost identical to the derivation in Equations 4-21 in Section 1.3. The Newtonian force balance yields the equation of motion for this system is

$$m_p \ddot{u}_p + c_x \dot{u}_p + k_x u_p = 0. \tag{22}$$

Substituting

$$\xi_{slip} = \frac{c_x}{2\omega_{slip} m_p}, \quad where\ \omega_{slip} = \sqrt{\frac{k_x}{m_p}}, \tag{23}$$

and rearranging the equation yields

$$\ddot{u}_p + 2\omega_{slip}\xi_{slip}\dot{u}_p + \omega_{slip}^2 u_p = 0. \tag{24}$$

It is assumed that the solution of Equation 24 takes the exponential form

$$u_{p,slip}(t) = Ce^{st}. \tag{25}$$

Substituting Equation 25 into Equation 24 yields

$$s^2 + 2\omega_{slip}\xi s_{slip} + \omega^2 = 0. \tag{26}$$

The roots of this equation are

$$s = \omega_{slip}\xi_{slip} \pm \omega_{slip}\sqrt{\xi_{slip}^2 - 1}. \tag{27}$$

In this case, the viscoelastic portion of the system consists of a solid elastic material (i.e. the brake pad). Thus, it is reasonable to assume that the system will always be underdamped, i.e. $\xi_{slip}^2 < 1$. Under this assumption, the radicand of Equation 27 will be imaginary

$$s = \omega_{slip}\xi_{slip} \pm i\omega_{d,slip}. \tag{28}$$

In Equation 28, $\omega_{d,slip}$ is the damped natural frequency of the system and is defined as

$$\omega_{d,slip} = \omega_{slip}\sqrt{1 - \xi_{slip}^2}. \tag{29}$$

Substitution of the roots from Equation 28 into the assumed solution from Equation 25 yields two solutions

$$u_{p,slip}(t) = Ce^{\left(-\omega_{slip}\xi_{slip} \pm i\omega_{d,slip}\right)t}, \tag{30}$$

which yields a general solution that consists of a linear combination of these solutions

$$u_{p,slip}(t) = C_1e^{\left(-\omega_{slip}\xi_{slip} + i\omega_{d,slip}\right)t} + C_2e^{\left(-\omega_{slip}\xi_{slip} - i\omega_{d,slip}\right)t}. \tag{31}$$

This general solution can, alternatively be expressed in terms of a trigonometric sum

$$u_{p,slip}(t) = e^{-\omega_{slip}\xi_{slip}t}\left(A_1\cos\left(\omega_{d,slip}t\right) + A_2\sin(\omega_{d,slip}t)\right). \tag{32}$$

The constant terms, A_1 and A_2, in Equation 32 are dependent on initial condition ($u_p(t = 0) = u_{p_0}$ and $\dot{u}_p(t = 0) = \dot{u}_{p_0}$)

$$A_{1,slip} = u_{p_0} \tag{33}$$

$$A_{2,slip} = \frac{\frac{\dot{u}_{p0}}{\omega} + \xi_{slip} u_{p_0}}{\sqrt{1 - \xi_{slip}^2}}. \tag{34}$$

At this point, it makes sense to define the kinetic friction, f_{μ_k}, that exists in the system

$$f_{\mu_k}\big(F_N(t)\big) = \begin{cases} \mu_k F_N(t) & F_N(t) > 0 \\ 0 & F_N(t) \leq 0. \end{cases} \tag{35}$$

The effect that this friction will have on the system can be expressed as a Coulomb damping factor [3]

$$f_{coul}\big(F_N(t)\big) = \begin{cases} \frac{\mu_k F_N(t)}{k} & F_N(t) > 0 \\ 0 & F_N(t) \leq 0. \end{cases} \tag{36}$$

The deceleration of the bar due to friction will be defined by

$$\ddot{u}_d = -\frac{f_{coul}}{m_d}. \tag{37}$$

Taking the temporal integral of this expression over the stick phase yields the bar's velocity

$$\dot{u}_{d,slip} = \int \ddot{u}_d dt \tag{38}$$

$$\dot{u}_{d,slip} = -\int \frac{f_{coul}}{m_d} dt \tag{39}$$

$$\dot{u}_{d,slip} = -\frac{f_{coul} t}{m_d}. \tag{40}$$

Adding the f_{coul} term to Equation 32 yields

$$u_p(t) = \begin{cases} e^{-\omega_{slip}\xi_{slip}t}\big(A_1 \cos\big(\omega_{d,slip}t\big) + A_2 \sin(\omega_{d,slip}t)\big) - f_{coul} & \dot{u}_p(t) > 0 \\ e^{-\omega_{slip}\xi_{slip}t}\big(A_1 \cos\big(\omega_{d,slip}t\big) + A_2 \sin(\omega_{d,slip}t)\big) + f_{coul} & \dot{u}_p(t) < 0. \end{cases} \tag{41}$$

In this system, kinetic friction will only be observed during the slip phase, when $\dot{u}_p(t)$ is negative, per the chosen coordinate system. Therefore Equation 41 can be reduced to

$$u_p(t) = e^{-\omega_{slip}\xi_{slip}t}\left(A_1 \cos\left(\omega_{d,slip}t\right) + A_2 \sin(\omega_{d,slip}t)\right) + f_{coul}. \qquad (42)$$

1.3.2. Full Definition of Stick-Slip Motion

Thus, the horizontal component of the brake pad's periodic motion on a time interval $t_1 < t < t_3$ is fully defined as

$$u_p(t_1 < t < t_3) = \begin{cases} e^{-\omega_{stick}\xi_{stick}t}\left(A_{1,stick} \cos\left(\omega_{d,stick}t\right) + A_{2,stick} \sin(\omega_{d,stick}\, t)\right) & f_{\mu_s} > F_{springx} \\ e^{-\omega_{slip}\xi_{slip}t}\left(A_{1,slip} \cos\left(\omega_{d,slip}t\right) + A_{2,slip} \sin(\omega_{d,slip}t)\right) + f_{coul} & f_{\mu_k} < F_{springx}. \end{cases} \qquad (43)$$

Alternatively, this can be expressed in terms of t_1, t_2, t_3

$$u_p(t) = \begin{cases} e^{-\omega_{stick}\xi_{stick}t}\left(A_{1,stick} \cos\left(\omega_{d,stick}t\right) + A_{2,stick} \sin(\omega_{d,stick}\, t)\right) & t_1 < t < t_2 \\ e^{-\omega_{slip}\xi_{slip}t}\left(A_{1,slip} \cos\left(\omega_{d,slip}t\right) + A_{2,slip} \sin(\omega_{d,slip}t)\right) + f_{coul} & t_1 < t < t_3. \end{cases} \qquad (44)$$

Similarly, the velocity of the bar is fully defined as

$$u_d(t) = \begin{cases} e^{-\omega_{stick}\xi_{stick}t}\left(A_{1,stick} \cos\left(\omega_{d,stick}t\right) + A_{2,stick} \sin(\omega_{d,stick}\, t)\right) & t_1 < t < t_2 \\ -\dfrac{f_{coul}t}{m_d} & t_1 < t < t_3. \end{cases} \qquad (45)$$

1.4. Definition of the Vertical Motion of Brake Pads

The vertical motion of the brake pad $v_p(t)$ is driven by the motion of the disc's surface at the point of contact of the pad. The disc simultaneously vibrates and moves through space. As a

result, the resulting agitation applied to the pad is referred to as $\upsilon_d(t)$. Newtonian force balance of this system yields that the equation of motion is

$$m\ddot{\upsilon}_p + c\left(\dot{\upsilon}_p - \dot{\upsilon}_d\right) + k\left(\upsilon_p - \upsilon_d\right) = 0. \tag{46}$$

Making substitutions from Equation 23 and rearranging yields

$$\ddot{\upsilon}_p + 2\omega\xi\left(\dot{\upsilon}_p - \dot{\upsilon}_d\right) + \omega^2\left(\upsilon_p - \upsilon_d\right) = 0. \tag{47}$$

It is possible to separate this equation by its terms

$$\ddot{\upsilon}_p + 2\omega\xi\dot{\upsilon}_p + \omega^2\upsilon_p = 2\omega\xi\dot{\upsilon}_d + \omega^2\upsilon_d. \tag{48}$$

If it is assumed that the vibration of the disc is harmonic, the frequency of the disc's vibration is denoted as Ω, and the amplitude of the disc's vibration can be denoted as Y, then the vibration of the disk can be expressed as

$$\upsilon_d = V_d e^{i\Omega\mathrm{t}}. \tag{49}$$

Substituting Equation 49 into Equation 48 yields

$$\ddot{\upsilon}_p + 2\omega\xi\dot{\upsilon}_p + \omega^2\upsilon_p = 2\omega\xi i V_d \Omega e^{i\Omega\mathrm{t}} + \omega^2 V_d e^{i\Omega\mathrm{t}}. \tag{50}$$

Rearranging Equation 50 yields

$$\ddot{\upsilon}_p + 2\omega\xi\dot{\upsilon}_p + \omega^2\upsilon_p = \left(2\omega\xi i\Omega + \omega^2\right)V_d e^{i\Omega\mathrm{t}}. \tag{51}$$

For both notational brevity and ease of algebraic manipulation, it makes sense to define the ratio between the frequency of the disc's vibration and the natural frequency of the brake pad, Ω/ω, as

$$\overline{\Omega} = \frac{\Omega}{\omega}. \tag{52}$$

Substituting $\overline{\Omega}$ into Equation 51 yields

$$\ddot{\upsilon}_p + 2\omega\xi\dot{\upsilon}_p + \omega^2\upsilon_p = \left(2\xi i\overline{\Omega} + 1\right)\omega^2 V_d e^{i\Omega \mathrm{t}}. \tag{53}$$

This equation can be written as

$$\ddot{\upsilon}_p + 2\omega\xi\dot{\upsilon}_p + \omega^2\upsilon_p = \omega^2 V_d e^{i\Omega \mathrm{t}} + 2\omega^2\xi i\overline{\Omega} V_d e^{i\Omega \mathrm{t}}. \tag{54}$$

The principle of superposition allows for the separation of Equation 54 into two parts

$$\ddot{\upsilon}_p + 2\omega\xi\dot{\upsilon}_p + \omega^2\upsilon_p = \omega^2 V_d e^{i\Omega \mathrm{t}} \tag{55}$$

$$\ddot{\upsilon}_p + 2\omega\xi\dot{\upsilon}_p + \omega^2\upsilon_p = 2\omega^2\xi i\overline{\Omega} V_d e^{i\Omega \mathrm{t}}. \tag{56}$$

The total steady state solution of Equation 54, $\upsilon_{p_{ss}}$, will be equal to the sum of the steady state solutions of Equation 55 and Equation 56. Because Equation 55 and Equation 56 are functionally identical, their solution process is extremely similar.

This solution process begins with the assumption that $\upsilon_p(t)$ is exponential:

$$\upsilon_p(t) = V_p e^{i\Omega \mathrm{t}}. \tag{57}$$

Substituting this assumed value into Equation 55 and Equation 56 yields

$$-\Omega^2 V_p e^{i\Omega \mathrm{t}} + 2\omega\xi i\Omega V_p e^{i\Omega \mathrm{t}} + \omega^2 V_p e^{i\Omega \mathrm{t}} = \omega^2 V_d e^{i\Omega \mathrm{t}}. \tag{58}$$

$$-\Omega^2 V_p e^{i\Omega \mathrm{t}} + 2\omega\xi i\Omega V_p e^{i\Omega \mathrm{t}} + \omega^2 V_p e^{i\Omega \mathrm{t}} = 2\omega^2\xi i\overline{\Omega} V_d e^{i\Omega \mathrm{t}}. \tag{59}$$

It is readily apparent that the $e^{i\Omega \mathrm{t}}$ term cancels out in both Equation 58 and Equation 59

$$-\Omega^2 V_p + 2\omega\xi i\Omega V_p + \omega^2 V_p = \omega^2 V_d. \tag{60}$$

$$-\Omega^2 V_p + 2\omega\xi i\Omega V_p + \omega^2 V_p = 2\omega^2\xi i\overline{\Omega} V_d. \tag{61}$$

It is similarly apparent that the V_p term can be factored out of the polynomial on the left-hand sides of both equations, resulting in two equations

$$\left(-\Omega^2 + 2\omega\xi i\Omega + \omega^2\right)V_p = \omega^2 V_d \tag{62}$$

$$\left(-\Omega^2 + 2\omega\xi i\Omega + \omega^2\right)V_p = 2\omega^2\xi i\overline{\Omega}V_d. \tag{63}$$

Solving Equations 62and 63 for V_p yields two solutions

$$V_p = \begin{cases} \dfrac{\omega^2 V_d}{-\Omega^2 + 2\omega\xi i\Omega + \omega^2} \\ \dfrac{2\omega^2\xi i\overline{\Omega}V_d}{-\Omega^2 + 2\omega\xi i\Omega + \omega^2}. \end{cases} \tag{64}$$

Multiplying the numerators and denominators of these two solutions by $1/\omega^2$ allows for the use of the $\overline{\Omega}$ term defined in Equation 52, rather than Ω and ω

$$V_p = \begin{cases} \dfrac{V_d}{\left(1 - \overline{\Omega}^2 + 2\xi\overline{\Omega}i\right)} \\ \dfrac{2\xi i\overline{\Omega}V_d}{\left(1 - \overline{\Omega}^2 + 2\xi\overline{\Omega}i\right)}. \end{cases} \tag{65}$$

The imaginary term can be moved from the denominator to the numerator by the complex conjugate of the denominator

$$V_p = \begin{cases} \dfrac{V_d}{\left(1 - \overline{\Omega}^2 + 2\xi\overline{\Omega}i\right)} \bullet \dfrac{\left(1 - \overline{\Omega}^2 - 2\xi\overline{\Omega}i\right)}{\left(1 - \overline{\Omega}^2 - 2\xi\overline{\Omega}i\right)} \\ \dfrac{2\xi i\overline{\Omega}V_d}{\left(1 - \overline{\Omega}^2 + 2\xi\overline{\Omega}i\right)} \bullet \dfrac{\left(1 - \overline{\Omega}^2 - 2\xi\overline{\Omega}i\right)}{\left(1 - \overline{\Omega}^2 - 2\xi\overline{\Omega}i\right)} \end{cases} \tag{66}$$

$$V_p = \begin{cases} V_d \dfrac{\left(1-\overline{\Omega}^2\right) - 2i\overline{\Omega}\xi}{\left(1-\overline{\Omega}^2\right)^2 + \left(2\overline{\Omega}\xi\right)^2} \\ 2\xi i\overline{\Omega} V_d \dfrac{\left(1-\overline{\Omega}^2\right) - 2i\overline{\Omega}\xi}{\left(1-\overline{\Omega}^2\right)^2 + \left(2\overline{\Omega}\xi\right)^2}. \end{cases} \tag{67}$$

Rearranging Equation 67 yields

$$V_p = \begin{cases} V_d \left(\dfrac{\left(1-\overline{\Omega}^2\right)}{\left(1-\overline{\Omega}^2\right)^2 + \left(2\overline{\Omega}\xi\right)^2} + \dfrac{-2i\overline{\Omega}\xi}{\left(1-\overline{\Omega}^2\right)^2 + \left(2\overline{\Omega}\xi\right)^2} \right) \\ 2\xi i\overline{\Omega} V_d \left(\dfrac{\left(1-\overline{\Omega}^2\right)}{\left(1-\overline{\Omega}^2\right)^2 + \left(2\overline{\Omega}\xi\right)^2} + \dfrac{-2i\overline{\Omega}\xi}{\left(1-\overline{\Omega}^2\right)^2 + \left(2\overline{\Omega}\xi\right)^2} \right) \end{cases} \tag{68}$$

$$V_p = \begin{cases} V_d \left(\dfrac{\left(1-\overline{\Omega}^2\right)}{\left(1-\overline{\Omega}^2\right)^2 + \left(2\overline{\Omega}\xi\right)^2} + \dfrac{-2\overline{\Omega}\xi}{\left(1-\overline{\Omega}^2\right)^2 + \left(2\overline{\Omega}\xi\right)^2} i \right) \\ 2\xi i\overline{\Omega} V_d \left(\dfrac{\left(1-\overline{\Omega}^2\right)}{\left(1-\overline{\Omega}^2\right)^2 + \left(2\overline{\Omega}\xi\right)^2} + \dfrac{-2\overline{\Omega}\xi}{\left(1-\overline{\Omega}^2\right)^2 + \left(2\overline{\Omega}\xi\right)^2} i \right). \end{cases} \tag{69}$$

These expressions of V_p coincide with simple $a + bi$ form for complex numbers, in which

$$a = \frac{\left(1-\overline{\Omega}^2\right)}{\left(1-\overline{\Omega}^2\right)^2 + \left(2\overline{\Omega}\xi\right)^2} \tag{70}$$

$$b = \frac{-2\overline{\Omega}\xi}{\left(1-\overline{\Omega}^2\right)^2 + \left(2\overline{\Omega}\xi\right)^2}. \tag{71}$$

This form can be converted to exponential form, $Ae^{-i\phi}$, in which

$$A = \sqrt{a^2 + b^2} \tag{72}$$

$$\phi = \operatorname{atan}\left(\frac{b}{a}\right). \tag{73}$$

Substituting Equation 70 and Equation 71 into Equation 72 and Equation 73, accordingly, yields

$$A = \sqrt{\left(\frac{\left(1-\overline{\Omega}^2\right)}{\left(1-\overline{\Omega}^2\right)^2 + \left(2\overline{\Omega}\xi\right)^2}\right)^2 + \left(\frac{-2\overline{\Omega}\xi}{\left(1-\overline{\Omega}^2\right)^2 + \left(2\overline{\Omega}\xi\right)^2}\right)^2} \tag{74}$$

$$A = \frac{1}{\sqrt{\left(1-\overline{\Omega}^2\right)^2 + \left(2\overline{\Omega}\xi\right)^2}}. \tag{75}$$

$$\phi = \operatorname{atan}\left(\frac{\frac{-2\overline{\Omega}\xi}{\left(1-\overline{\Omega}^2\right)^2+\left(2\overline{\Omega}\xi\right)^2}}{\frac{\left(1-\overline{\Omega}^2\right)}{\left(1-\overline{\Omega}^2\right)^2+\left(2\overline{\Omega}\xi\right)^2}}\right) \tag{76}$$

$$\phi = \operatorname{atan}\left(\frac{-2\overline{\Omega}\xi}{\left(1-\overline{\Omega}^2\right)^2 + \left(2\overline{\Omega}\xi\right)^2} \bullet \frac{\left(1-\overline{\Omega}^2\right)^2 + \left(2\overline{\Omega}\xi\right)^2}{\left(1-\overline{\Omega}^2\right)}\right) \tag{77}$$

$$\phi = \operatorname{atan}\left(\frac{-2\overline{\Omega}\xi}{\left(1-\overline{\Omega}^2\right)}\right). \tag{78}$$

The A term from Equation 74 is the magnification factor and will be referred to as $\Gamma\left(\overline{\Omega}, \xi\right)$, to keep in line with standard notation within the field of vibrations

$$\Gamma\left(\overline{\Omega}, \xi\right) = \frac{1}{\sqrt{\left(1-\overline{\Omega}^2\right)^2 + \left(2\overline{\Omega}\xi\right)^2}} \tag{79}$$

Thus, V_p can be expressed as

$$V_p = \begin{cases} V_d\Gamma\left(\bar{\Omega}, \xi\right)e^{-i\phi} \\ 2\xi i\bar{\Omega}V_d\Gamma\left(\bar{\Omega}, \xi\right)e^{-i\phi} \end{cases} \tag{80}$$

Substituting these values of V_p into the assumed solution (Equation 57) yields to steady state solutions

$$v_{p_{ss1}}(t) = V_d\Gamma\left(\bar{\Omega}, \xi\right)e^{-i\phi}e^{i\Omega t} \tag{81}$$

$$v_{p_{ss2}}(t) = 2\xi i\bar{\Omega}V_d\Gamma\left(\bar{\Omega}, \xi\right)e^{-i\phi}e^{i\Omega t}. \tag{82}$$

Applying the principle of superposition yields

$$v_{p_{ss}}(t) = v_{p_{ss1}}(t) + v_{p_{ss2}}(t) \tag{83}$$

$$v_{p_{ss}}(t) = V_d\Gamma\left(\bar{\Omega}, \xi\right)e^{-i\phi}e^{i\Omega t} + 2\xi i\bar{\Omega}V_d\Gamma\left(\bar{\Omega}, \xi\right)e^{-i\phi}e^{i\Omega t} \tag{84}$$

$$v_{p_{ss}}(t) = V_d\Gamma\left(\bar{\Omega}, \xi\right)\left(1 + 2\xi i\bar{\Omega}\right)e^{(\Omega t - \phi)i}. \tag{85}$$

For the purpose of maximizing both algebraic and computational ease, it is desirable for all imaginary terms in Equation 85 to be in the exponent. Much like when dealing with Equation 69, the imaginary part of Equation 85, is in standard simple $a + bi$ form for complex numbers, in which

$$a = 1 \tag{86}$$

$$b = 2\xi\bar{\Omega}. \tag{87}$$

Substituting Equation 86 and Equation 87 into Equation 72 and Equation 73 allows for this imaginary term to be converted to converted to $Ae^{-i\psi}$ phasor form[1]

$$A = \sqrt{1 + \left(2\xi\overline{\Omega}\right)^2} \tag{88}$$

$$\psi = \mathrm{atan}\left(2\xi i\overline{\Omega}\right). \tag{89}$$

Thus

$$\left(1 + 2\xi i\overline{\Omega}\right) = \sqrt{1 + \left(2\xi i\overline{\Omega}\right)^2}\, e^{\psi i}. \tag{90}$$

Substituting Equation 90 into Equation 85 yields

$$v_{p_{ss}}(t) = V_d\Gamma\left(\overline{\Omega}, \xi\right)\sqrt{1 + \left(2\xi\overline{\Omega}\right)^2}\, e^{(\Omega \mathrm{t} - \phi + \psi)i}. \tag{91}$$

Through the use of trigonometric identities, the $(-\phi + \psi)$ portion of Equation 91 can be expanded

$$(-\phi + \psi) = -\,\mathrm{atan}\left(\frac{-2\overline{\Omega}\xi}{\left(1 - \overline{\Omega}^2\right)}\right) + \mathrm{atan}\left(2\xi i\overline{\Omega}\right) \tag{92}$$

$$(-\phi + \psi) = \mathrm{atan}\left(\frac{2\xi\overline{\Omega}^3}{\left(1 - \overline{\Omega}^2\right) + \left(2\overline{\Omega}\xi\right)^2}\right). \tag{93}$$

For brevity, the result of Equation 93 can be assigned to a new term, Ψ:

[1] ψ is used as the exponential coefficient in order to distinguish it from the definition of ϕ obtained in Equation 78.

$$\Psi = \operatorname{atan}\left(\frac{2\xi\overline{\Omega}^3}{\left(1-\overline{\Omega}^2\right)+\left(2\overline{\Omega}\xi\right)^2}\right) \tag{94}$$

Substituting Ψ into Equation 91 yields a concise expression for the vertical motion of the brake pad

$$v_{p_{ss}}(t) = V_d\Gamma\left(\bar{\Omega},\xi\right)\sqrt{1+\left(2\xi\overline{\Omega}\right)^2}\,e^{(\Omega t-\Psi)i} \tag{95}$$

1.5. Vibration of the Disc

The normal force $F_N(t)$ introduced in Equation 3 is defined in terms of the vertical agitation caused by the brake pad v_d and the braking force F_b which is the force with which the brake pad actuator acts on the brake pads

$$F_N(t) = F_b(t) + k_y v_p(t). \tag{96}$$

For this system, base motion $v_d(t)$ is a function of the axial deformation of the bar and the speed with which it moves. Prior to brake engagement, brake pads are not in contact with the bar and it is free to move. At the moment of engagement, the pad comes in contact with the moving bar and the stick phase begins. When the pad's movement transitions to the slip phase, it experiences a jerk, or a rapid rate of change in acceleration. This jerk acts to initiate free vibration of the disc and can be adequately approximated with a Dirac delta function[2].

At this point, it is assumed that the magnitude of normal force acting on the pad is large enough to induce the stick phase of the pad's stick-slip motion. Because the pad's resulting

[2] See Figure 12e for example of the jerk of the pad.

displacement will be orders of magnitude smaller than characteristic length of the bar that models the disc, the pad's relationship with the bar can be approximated as a rigid support. Because periodic boundary conditions are applied to the bar, the other end of the bar can be modelled as being rigidly supported at both ends (Figure 9).

A free body diagram of a small portion of the bar Δx can be drawn (Figure 10). Applying Newtonian force balance to Δx results in the equation of motion

$$(\rho A \Delta x)\frac{\partial^2 \delta}{\partial t^2} = P(x + \Delta x, t) - P(x, t)\ , \tag{97}$$

where ρ is the density of the bar, A is the cross-sectional area of the bar, and δ is the deformation of the bar. As Δx approaches zero, this equation becomes

$$(\rho A)\frac{\partial^2 \delta}{\partial t^2} = \frac{\partial P(x, t)}{\partial x}. \tag{98}$$

The force $P(x, t)$ in this equation can be expressed as

$$P(x, t) = AE\frac{\partial \delta}{\partial x}. \tag{99}$$

Substituting Equation 99 into Equation 98 and cancelling the cross-sectional area terms yields the homogeneous wave equation

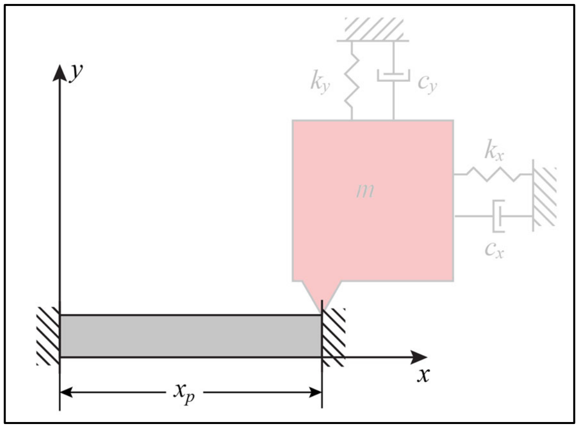


Figure 9 – Proposed model for derivation of rotor deformation with a fixed-fixed bar element

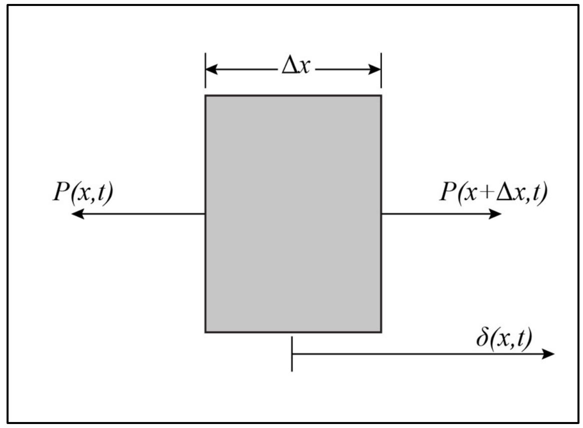


Figure 10 – Free body diagram of Δx

$$(\rho A)\frac{\partial^2 \delta}{\partial t^2} = AE\frac{\partial^2 \delta}{\partial x^2}. \tag{100}$$

$$\rho\frac{\partial^2 \delta}{\partial t^2} = E\frac{\partial^2 \delta}{\partial x^2}. \tag{101}$$

Through the principle of separation of variables, the solution of this partial differential equation can take the form

$$\delta(x,t) = g(x)h(t). \tag{102}$$

Substituting this solution into Equation 101 yields

$$\rho g(x)\ddot{h}(t) = Eg''(x)h(t). \tag{103}$$

Rearranging this equation yields

$$\frac{\ddot{h}(t)}{h(t)} = \frac{E}{\rho}\frac{g''(x)}{g(x)}. \tag{104}$$

This expression can be set equal to a value, $-\omega^2$

$$\frac{\ddot{h}(t)}{h(t)} = \frac{E}{\rho}\frac{g''(x)}{g(x)} - \omega^2, \tag{105}$$

Which allows for the variables, $g(x)$ and $h(t)$, to be separated

$$\begin{cases} \ddot{h}(t) + \omega^2 h(t) = 0 \\ g''(x) + \dfrac{\omega^2 \rho}{E} g(x) = 0 \end{cases} \tag{106}$$

The solution set to these to equations is

$$\begin{cases} h(t) = C_1 \cos(\omega t) + C_2 \sin(\omega t) \\ g(x) = C_3 \cos\left(\sqrt{\dfrac{\omega^2 \rho}{E}} x\right) + C_4 \sin\left(\sqrt{\dfrac{\omega^2 \rho}{E}} x\right) \end{cases} \tag{107}$$

The values of coefficients C_1 through C_4 are dependent on initial and boundary conditions. For brevity, a variable λ will be defined as

$$\lambda = \omega\sqrt{\frac{\rho}{E}}, \tag{108}$$

which allows for Equation 107 to be written as

$$\begin{cases} h(t) = C_1 \cos(\omega t) + C_2 \sin(\omega t) \\ g(x) = C_3 \cos(\lambda x) + C_4 \sin(\lambda x) \end{cases} \tag{109}$$

From the free body diagram, it is readily inferred that the boundary conditions of the bar can be expressed as

$$\frac{\partial \delta(x = 0, t)}{\partial x} = 0 \tag{110}$$

$$\frac{\partial \delta\left(x = x_p, t\right)}{\partial x} = 0 \tag{111}$$

Substituting the first boundary condition into $g(x)$, as shown in Equation 109, yields

$$g(0) = C_3 \cos(0) + C_4 \sin(0) = 0 \tag{112}$$

$$g(0) = C_3 = 0. \tag{113}$$

Thus, the expression for $g(x)$ becomes

$$g(x) = C_4 \sin(\lambda x) \tag{114}$$

In order to avoid a trivial solution, C_4 cannot be allowed to be equal to zero. Therefore, the argument of the sinusoidal expression must be such that it generates a value of zero. This yields a value for λ of

$$\lambda_n = \frac{n\pi}{L}, \qquad where\ n = 1,2, \dots \infty. \tag{115}$$

Because linear combination of a solution to a differential equation are solutions to the equation, the C_4 term in Equation 114 can be disregarded. Thus, the expression for $g(x)$ becomes

$$g(x) = \sin(\lambda x) = \sin\left(\frac{n\pi}{L}x\right), \tag{116}$$

which is also known as the modal function. Similarly, this yields a definition for ω

$$\omega_n = \frac{n\pi}{L}\sqrt{\frac{E}{\rho}}, \qquad where\ n = 1,2, \dots \infty. \tag{117}$$

Substituting the new expression for $g(x)$ and $h(t)$ with the new definition of ω_n into Equation 102 yields

$$\delta(x,t) = \sum_{n=1}^{\infty} \sin(\lambda_n x)\left[C_1 \cos(\omega_n t) + C_2 \sin(\omega_n t)\right]. \tag{118}$$

The constants C_1 and C_2 in $h(t)$ are found by considering the initial conditions

$$\delta(x, t = 0) = \delta_0(x) \tag{119}$$

$$\dot{\delta}(x, t = 0) = \dot{\delta}_0(x), \tag{120}$$

and applying them to Equation 118

$$\delta_0(x) = \sum_{n=1}^{\infty} \sin(\lambda_n x) \left[C_1\right] \tag{121}$$

$$\dot{\delta}_0(x) = \sum_{n=1}^{\infty} \sin(\lambda_n x) \left[\omega_n C_2\right]. \tag{122}$$

The values of C_1 and C_2 are solved for by multiplying these two expression by the m^{th} modal function and by the mass operator, $m(x)$, and integrating over the bar's spatial domain [3]

$$\int_0^L \sin(\lambda_n x)\, m(x)\delta_0(x)dx = \sum_{n=1}^{\infty} \left[\int_0^L \sin(\lambda_n x)\, m(x) \sin(\lambda_n x)\, dx\right] \left[C_1\right] \tag{123}$$

$$C_1 = \frac{\int_0^L \sin(\lambda_n x)\, m(x)\delta_0(x)dx}{\int_0^L \sin(\lambda_n x)\, m(x) \sin(\lambda_n x)\, dx} \tag{124}$$

$$\int_0^L \sin(\lambda_n x)\, m(x)\dot{\delta}_0(x)dx = \sum_{n=1}^{\infty} \left[\int_0^L \sin(\lambda_n x)\, m(x) \sin(\lambda_n x)\, dx\right] \left[\omega_n C_2\right]. \tag{125}$$

$$C_2 = \frac{\int_0^L \sin(\lambda_n x)\, m(x)\dot{\delta}_0(x)dx}{\int_0^L \sin(\lambda_n x)\, m(x) \sin(\lambda_n x)\, dx} \tag{126}$$

Because the bar has a constant cross section and density, the scalar mass operator, $m(x)$, is equal to the density of the bar, ρ

$$C_1 = \frac{\int_0^L \rho \sin(\lambda_n x)\, \delta_0(x)dx}{\int_0^L \rho \sin(\lambda_n x) \sin(\lambda_n x)\, dx} \tag{127}$$

$$C_2 = \frac{\int_0^L \rho \sin(\lambda_n x)\, \dot{\delta}_0(x) dx}{\int_0^L \rho \sin(\lambda_n x) \sin(\lambda_n x)\, dx}. \tag{128}$$

As ρ is a constant that is in both the numerator and denominator of both C_1 and C_2, it cancels out in both expressions

$$C_1 = \frac{\int_0^L \sin(\lambda_n x)\, \delta_0(x) dx}{\int_0^L \sin^2(\lambda_n x)\, dx} \tag{129}$$

$$C_2 = \frac{\int_0^L \sin(\lambda_n x)\, \dot{\delta}_0(x) dx}{\int_0^L \sin^2(\lambda_n x)\, dx}. \tag{130}$$

Evaluating the integrals in the denominators of both expressions yields

$$C_1 = \frac{\int_0^L \sin(\lambda_n x)\, \delta_0(x) dx}{\frac{L}{2} - \frac{\sin(2\lambda_n L)}{4\lambda_n}} \tag{131}$$

$$C_2 = \frac{\int_0^L \sin(\lambda_n x)\, \dot{\delta}_0(x) dx}{\frac{L}{2} - \frac{\sin(2\lambda_n L)}{4\lambda_n}}. \tag{132}$$

Expanding the λ_n term, per Equation 115, in the sinusoidal argument in the denominator of both expressions and simplifying yields

$$C_1 = \frac{\int_0^L \sin(\lambda_n x)\, \delta_0(x) dx}{\frac{L}{2} - \frac{\sin(2\frac{n\pi}{L}L)}{4\lambda_n}} \tag{133}$$

$$C_1 = \frac{\int_0^L \sin(\lambda_n x)\, \delta_0(x) dx}{\frac{L}{2} - \frac{\sin(2n\pi)}{4\lambda_n}} \tag{134}$$

$$C_2 = \frac{\int_0^L \sin(\lambda_n x)\, \dot{\delta}_0(x) dx}{\frac{L}{2} - \frac{\sin(2\frac{n\pi}{L}L)}{4\lambda_n}} \tag{135}$$

$$C_2 = \frac{\int_0^L \sin(\lambda_n x)\, \dot{\delta}_0(x) dx}{\frac{L}{2} - \frac{\sin(2n\pi)}{4\lambda_n}}. \tag{136}$$

To simplify the numerator, it is necessary to define the function that governs the initial deformation of the bar, $\delta_0(x)$.

The sum of the forces acting on the bar shown in Figure 11, can be expressed as:

$$\sum F = -P + R_A + R_B = 0. \tag{137}$$

The deformation of the left-hand side of the bar can be expressed as:

$$\delta_{ac} = \frac{R_A a}{EA}. \tag{138}$$

The deformation of the right-hand side of the bar can be expressed as:

$$\delta_{cb} = \frac{R_B b}{EA}. \tag{139}$$

Seeing as the overall displacement of the bar is equal to zero, the sum of the deformation equations must also be equal to zero

$$\delta_{ab} = \delta_{ac} - \delta_{cb} = 0 \tag{140}$$

$$\delta_{ab} = \frac{R_A a}{EA} - \frac{R_B b}{EA} = 0 \tag{141}$$

$$\delta_{ab} = \frac{R_A a - R_B b}{EA} = 0 \tag{142}$$

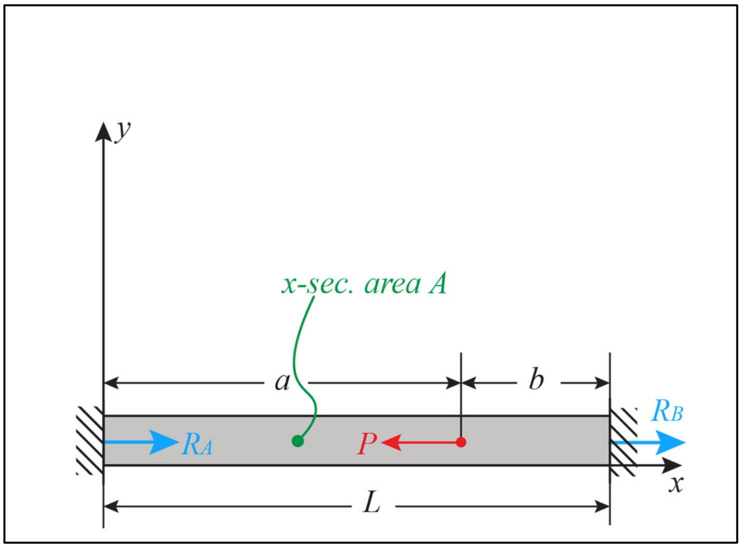


Figure 11 - Free body diagram of a bar with force P *applied at point* c*, and reactions* R_A *and* R_B

Thus, the reactions, R_A and R_B can be solved for

$$-P + R_A + R_B = 0 \tag{143}$$

$$R_A = P - R_B. \tag{144}$$

Substituting this value of R_A into Equation 142 yields

$$\delta_{ab} = \frac{(P - R_B)a - R_B b}{EA} = 0 \tag{145}$$

$$R_B = \frac{Pa}{a + b} \tag{146}$$

$$R_B = \frac{Pa}{L}. \tag{147}$$

Substituting this value of R_B into Equation 144 yields

$$R_A = P - \frac{Pa}{L}. \tag{148}$$

Thus, the deformation at point c, δ_c, can be expressed as

$$\delta_c = \delta_{ac} = \delta_{cb} = \frac{R_A a}{EA} = \frac{R_B b}{EA} \tag{149}$$

$$\delta_c = \frac{Pab}{LEA}. \tag{150}$$

Since the bar represents a metallic disc, the bar's material is linear and isotropic, the static deformation function for the bar shown in Figure 11 will be linear and piecewise

$$\delta(x) = \begin{cases} \frac{Pb}{LEA}x & x < c \\ -\frac{Pa}{LEA}x + \frac{Pa}{EA} & x > c. \end{cases} \tag{151}$$

Returning to the bar in the model, Equation 151 can be rewritten to express the initial displacement, $\delta_0(x)$ of the bar

$$\delta(x) = \begin{cases} \frac{P(L - x_p)}{LEA}x & x < x_p \\ -\frac{Px_p}{LEA}x + \frac{Px_p}{EA} & x > x_p. \end{cases} \tag{152}$$

If x_p is equal to L, then the deformation can be expressed as

$$\delta(x) = -\frac{P}{EA}x + \frac{PL}{EA}. \tag{153}$$

Thus, Equations 134 and 136 can be expressed as

$$C_1 = \frac{\int_0^L \sin(\lambda_n x)\left(\frac{PL}{EA} - \frac{P}{EA}x\right)dx}{\frac{L}{2} - \frac{\sin(2n\pi)}{4\lambda_n}} \tag{154}$$

$$C_2 = \frac{\int_0^L \sin(\lambda_n x)\left(\left(\frac{PL}{EA} - \frac{P}{EA}x\right)dt\right)dx}{\frac{L}{2} - \frac{\sin(2n\pi)}{4\lambda_n}} \tag{155}$$

Thus, the fully expanded expression for the horizontal deformation of the bar is

$$\delta(x,t) = \sum_{n=1}^{\infty} \sin\left(\frac{n\pi}{L}x\right)\left[\frac{\int_0^L \sin(\lambda_n x)\left(\frac{PL}{EA} - \frac{P}{EA}x\right)dx}{\frac{L}{2} - \frac{\sin(2n\pi)}{4\lambda_n}} \cos\left(\frac{n\pi}{L}\sqrt{\frac{E}{\rho}}\right)\right.$$
$$\left. + \frac{\int_0^L \sin(\lambda_n x)\left(\left(\frac{PL}{EA} - \frac{P}{EA}x\right)dt\right)dx}{\frac{L}{2} - \frac{\sin(2n\pi)}{4\lambda_n}} \sin\left(\frac{n\pi}{L}\sqrt{\frac{E}{\rho}}t\right)\right]. \tag{156}$$

This expression of the horizontal deformation of the bar can be used to express the axial strain of the bar

$$\varepsilon_{axial}(x,t) = \frac{\delta(x,t) - x}{x}. \tag{157}$$

Poison's ratio for a given material is defined as

$$\nu = \frac{\varepsilon_{transverse}}{\varepsilon_{axial}}. \tag{158}$$

From this, the transverse strain of the bar can be expressed as

$$\varepsilon_{transverse} = \varepsilon_{axial}\nu. \tag{159}$$

From this the vertical deformation of the disc, $\gamma(x,t)$ can be expressed as

$$\gamma(x,t) = (\varepsilon_{transverse} \bullet T), \tag{160}$$

where T is equal to the thickness of the bar. Similarly, from the strains above, the stresses experienced by the bar, or disc, can be determined.

In order to determine the vertical agitation of the brake pad, it is necessary to take into account the horizontal translational movement of the bar. This is done by adding a time dependent phase shift angle, $\phi(t)$, to the sinusoidal argument of the spatial term of the expression for $\delta(x,t)$:

$$\delta(x,t) = \sum_{n=1}^{\infty} \sin(\lambda_n x + \phi(t)) \left[C_1 \cos(\omega_n t) + C_2 \sin(\omega_n t)\right]. \tag{161}$$

This phase shift angle is dependent on the horizontal position of the bar, $u_d(t)$, and reflects the cyclical nature of the movement of the bar in this model:

$$\phi(t) = \frac{2\pi}{L} u_d(t). \tag{162}$$

2. Matlab Modeling

The mathematical model of disc brake vibration discussed in Section 1 was implemented in Matlab. This section discusses the implementation of this model.

2.1. Core Matlab Model Structure

The Matlab model is built around a “driver” script called BrkVib.m. This .m file calls the functions that define the characteristics and initial conditions of the system, as well as simulation settings. This .m file stores these values in the Matlab Workspace for the duration of the analysis. Once these values are stored, a loop that models the disc brake system is initiated. The number of steps in this loop is dependent on the simulation duration and time resolution, which are defined in the simulation settings. This loop considers the base-motion, stick slip vibration, and disc vibration. After the loop is complete, results are plotted and plots are saved.

These model components are described fully in the proceeding subsections. The full code is available in 4.2.Appendix B Core Matlab Code.

2.1.1. System Parameters

The system parameters are defined in SysPar.m. These system parameters are outlined in Table 1.

2.1.2. Derived Parameters

The system parameters that are derived from the parameters called out in Section 2.1.1 are defined in DerPar.m. These derived parameters outlined in Table *2*.

2.1.3. Simulation Settings

The simulation settings are defined in SimSet.m and are outlined in Table 3.

Table 1: System Parameters

Subsystem	Parameter	
General	Applied braking force	
	Acceleration due to gravity	
	Coefficients of static and kinetic friction	
Brake Pad	Brake pad mass	
	x-Direction:	Stiffness
		Damping
	y-Direction:	Stiffness
		Damping
Disc, represented as bar	Dimensional properties:	Length
		Thickness
		Cross sectional area
		Location of brake pad relative to length of bar
	Material properties:	Density
		Modulus of Elasticity
		Poisson's ratio
		Equivalent stiffness

Table 2: Derived System Parameters

Subsystem	Derived Parameter	
Brake Pad	x-Direction (both stick and slip):	Damped and undamped natural frequency
		Damping ratio
		Damped and undamped vibrational period
		Peak frequency
	y-Direction:	Damped and undamped natural frequency
		Damping ratio
		Damped and undamped vibrational period
		Peak frequency
Disc, represented as bar	Natural Frequency	

Table 3: Simulation Settings

Summation order (n in Equation 161)
Spatial resolution for integration (Equation 154, 155)
Location of brake pad, relative to disc
Simulation duration
Time resolution

2.1.4. Initial Conditions

The initial conditions (positions or displacements, and velocities) of all system components are defined in InitCond.m.

2.1.5. Simulation Loop

The simulation of the vibration of the disc brake system takes place in a for loop. This for loop carries out an iteration for every time step defined in the simulation settings this for loop calls on three functions – BasMot.m, StkSlp.m, and BarVib.m – to model the three coupled vibratory systems described in Section 1.

2.1.5.1. Base Motion

The modeling of base motion, as presented in Section 1.4 takes place in the function BasMot.m This function additionally call on a function, BasMot1.m, to define the normal force acting on the brake pad, which is used to in calculations related to stick-slip motion.

2.1.5.2. Stick Slip

The modeling of stick-slip motion occurs in StkSlp.m. This function calls on StkSlp1.m, StkSlp2.m, and StkSlp3.m to determine compare friction and spring forces to determine whether the system is in a state of stick or slip. StkSlp.m then goes on to determine a “local” time value, relative to the start of a stick or slip cyle. Similarly, StkSlp.m calls on StkSlp4.m to determine initial conditions for each stick and slip cycle throughout the simulation. After determining these values, StkSlp.m goes on to model the horizontal movement of the brakepad.

StkSlp.m call StkSlp5.m which, in turn, calls on either StkSlp5a.m or StkSlp5b.m, depending on whether the system is in a state of stick or slip. StkSlp5a. and StkSlp5b.m are simply Matlab implementations of the two cases of Equation 44. StkSlp.m additionally calls on StkSlp5c.m, which is simply a Matlab implementation of Equation 45.

2.1.5.3. Disc Vibration

The vibration of the disc is modelled in BarVib.m. This Matlab function is simply an implementation of Equation 161, followed by Equations 157-160.

2.2. Parameter Variation

In order to more robustly study the effects of different system parameters on the outputs of the system, a new script, ParameterStudy.m, was written. This script contains for loops that allow for the effects of ranges of system parameters on the behaviour of the system to be tested. A modified "driver" was created, BrkVib_Param.m was created to act as a function that could be called by ParameterStudy.m. In this version of the simulation code, all system parameters are provided by ParameterStudy.m, therefore SysPar.m is excluded. Additionally, a new function called PltPlt.m was create to generate plots associated with parameter variation studies. 4.2.Appendix C Parameter Variation Matlab Code (page 80) contains the full code of ParameterStudy.m, BrkVib_Param.m, and PltPlt.m.

3. Parameter Variation Study

This section describes a parameter variation study that was conducted with the Matlab code described in section 2.2 Parameter Variation and discusses its results. The purpose of this study is to demonstrate the flexibility of the developed model and how it can used be used to rapidly generate characterizations and predictions of complex coupled vibratory responses of disc braking systems with parameter variation. All results generated with this study are available in Appendix D Parameter Variation Study Results (page 98).

3.1. Description of Methods

This study began by defining a disc brake system in terms of the parameters described in section 2.1.1 System Parameters (page 36) and modulating the pad's mass, stiffness, and damping ratios. These parameters were selected to more readily validate results against those obtained by other researchers [13]. The pad's mass was modulated independently of all other parameters. The pad's horizontal stiffness and damping ratio were modulated independently of the vertical stiffness and damping ratio. Damping ratios were coupled to their corresponding stiffnesses and were defined by

$$c_x = \sqrt{k_x m_{pad}} \tag{163}$$

$$c_y = \sqrt{k_y m_{pad}}. \tag{164}$$

All quantities were modulated by 50% relative to their initial values. Sequential modulated parameter values differed from one another by 12.5% of the central value. Thus, 9 "equally spaced" parametric values were generated and range from 50% of the central value to 150% of the central value. The horizontal motion of the brake pad and the vertical motion of the disc were recorded with respect to time.

3.2. Study Results

A trend found in most simulations was that both the stick slip motion of the pad and the vibration of the disc required a certain amount of time to ramp-up to a stable vibrational pattern, with the disc typically having a substantially longer ramp-up time (Figure 12a, b). The ramp-up vibration of the disc was found to typically consist of two portions: a highly irregular phase (Figure 12c). and a traveling wave. After the travelling wave portion of ramp-up, disc vibration was found

exhibit a highly consistent standing wave (Figure 12d). The pad's displacement in the horizontal direction was found to consistently be non-linear. After ramp-up, it was found that the stick slip motion of pad predominantly had a reasonably low frequency. Intuitively, one might expect the pad's stick slip motion to have a skewed triangle waveform. However, due to damping and pad-disc mass coupling, as well as dynamic coupling between the system's three vibrational modes, stick slip motion was found to have a more complex and predominantly non-linear waveform in most parametric combinations. Furthermore, in parametric combinations that resulted in predominantly low frequency pad vibration, detailed observation made it clear that very low amplitude, high frequency vibration also existed after ramp-up. This suggest that a secondary stick-slip vibrational mode may exist. In most trials, modulation of parameters was found to simply lead to variation in ramp-up times (Figure 13).

However, it was also observed that certain parametric combinations led to markedly different behavior. Namely, was found that low pad mass, as well as low horizontal stiffness and damping, led to high amplitude, high frequency vibration immediately after ramp-up (Figure 14). The waveform of this vibration was more in line with the aforementioned skewed triangular shape that one would intuitively associate with stick slip vibration. This highly regular, high frequency vibration matches the characteristics of the vibration that causes undesirable, highly audible squeal [1]. Conversely, it was observed that low levels of vertical stiffness and damping resulted in irregular, lower frequency vibration (Figure 15), which can be generalized as being less bothersome and, therefore less undesirable. In a real-world system, vertical stiffness and damping corresponds to the characteristics of the actuation systems that depress the brake pads. Thus, these results suggest that, despite intuitive expectations, reducing actuator stiffness and damping is

actually an effective design strategy for mitigating brake noise. Similar results have been obtained in the modeling efforts of other researchers and have been validated on physical test beds [13].

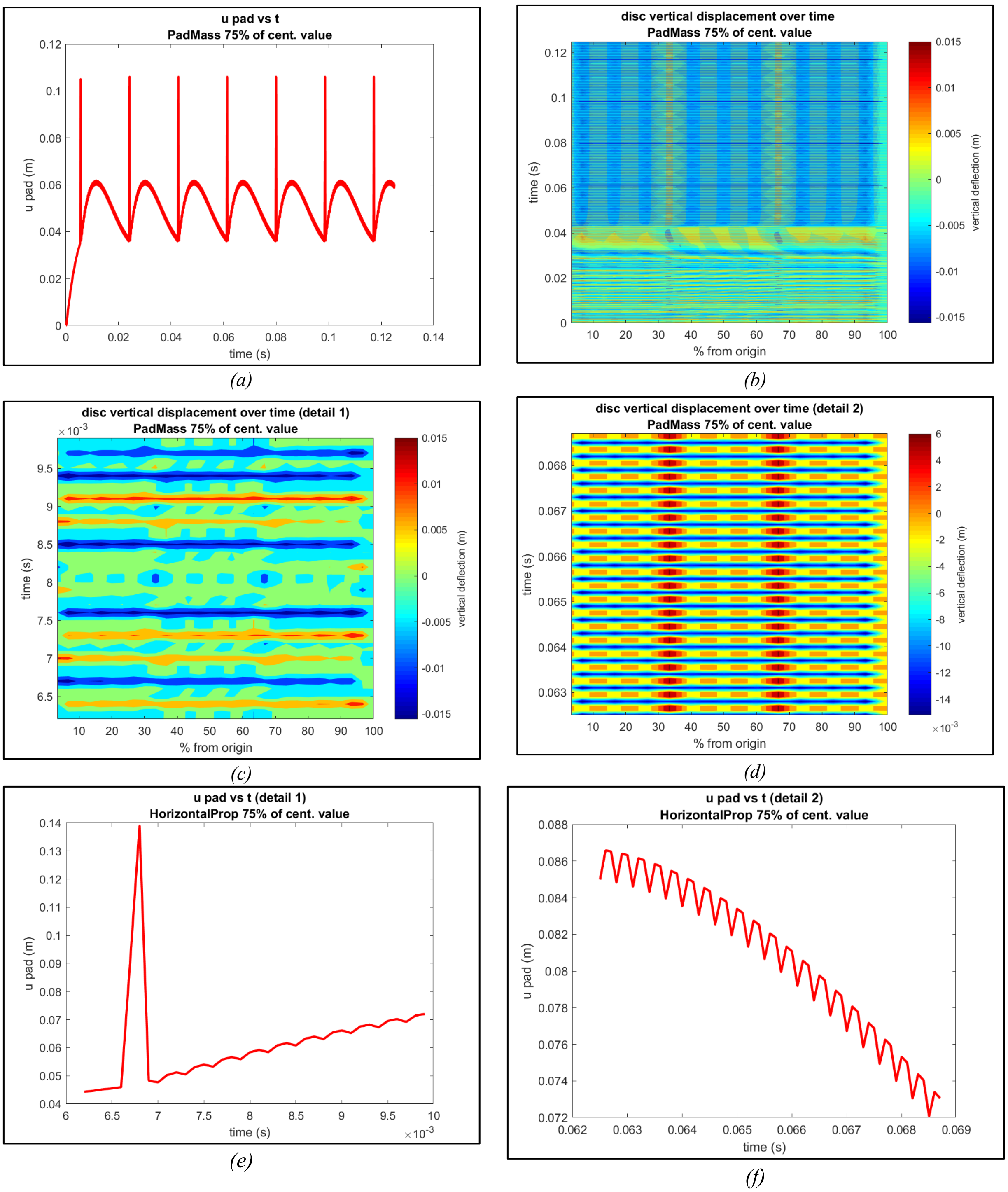


Figure 12 – Simulation results for 75% pad mass. Ramp-up time observed in both pad (a) and disc vibration (b), with disc ramp-up being substantially longer. Detail views show the highly irregular vibration of the early stage vibration of the disc (c) and the extremely consistent standing wave of the stable vibration phase (d). Detail views also show the consistent low amplitude, high frequency vibration of the pad that occurs after ramp-up (e, f).

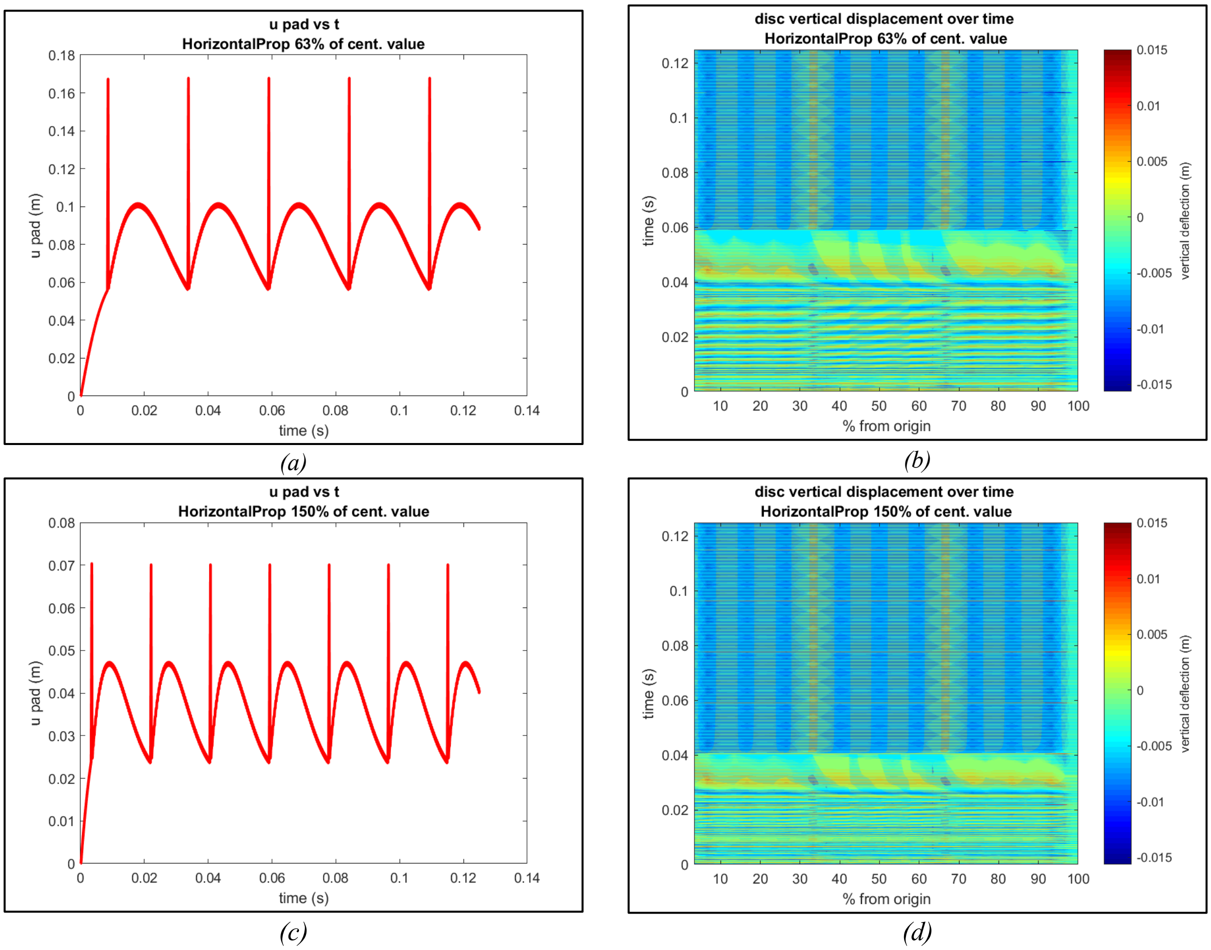
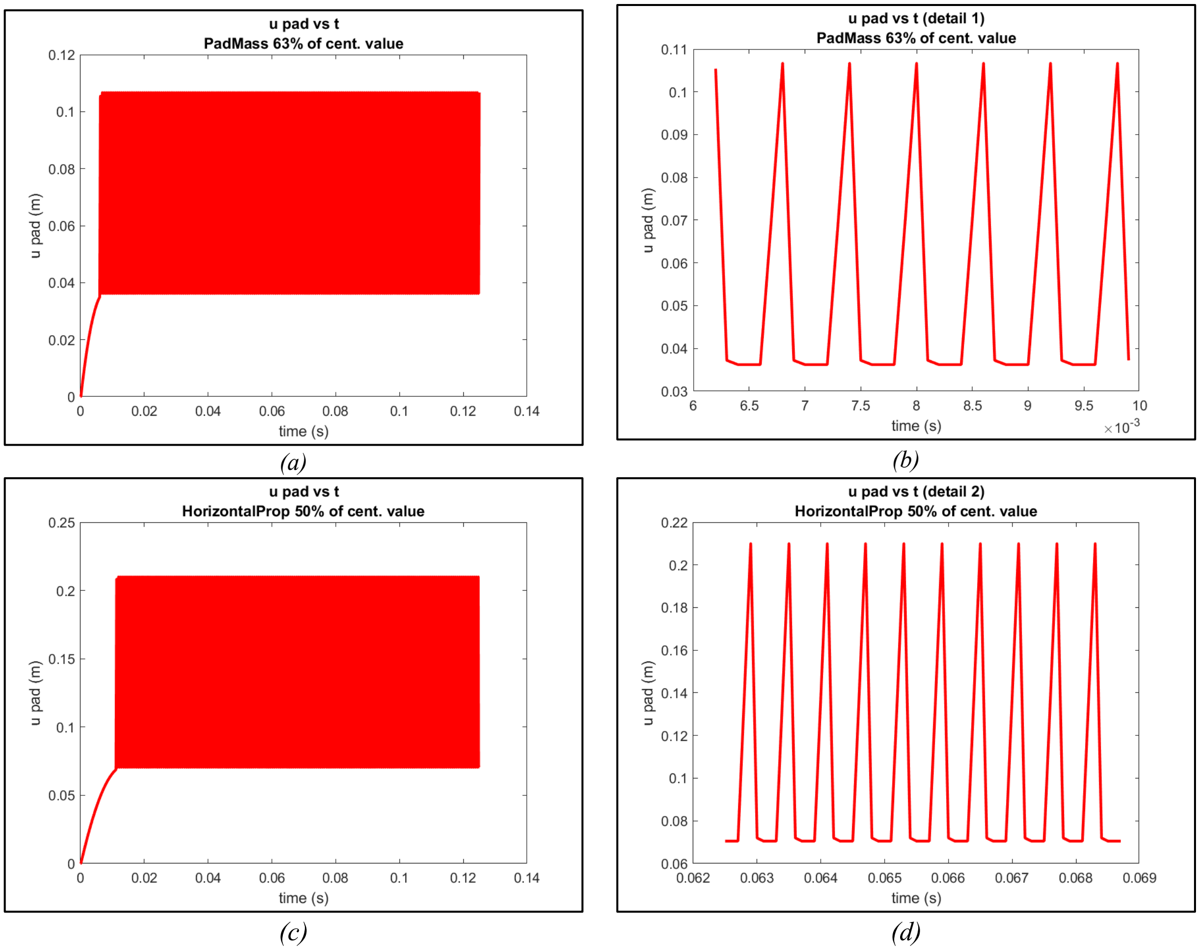


Figure 13 – Comparison of two parametric combinations with different ramp-up times. An increase in horizontal stiffness and damping led to a reduction in ramp-up time in both the pad and the disc. Of particular interest is the topologies of the displacement curves (a,b) and surfaces (b,d) do not change; especially note the same peak and trough locations and topologies in the traveling wave phase of the disc's vibration.



(a) *(b)* *(c)* *(d)*

Figure 14 – Comparison of two parametric combinations with high frequency pad vibration immediately after ramp-up. It is clear from both the full results (a, c) and the detail views (b, d) that only a single skewed triangular mode of vibration is present.

Similarly, it was observed that reduced vertical stiffness and damping, and increased brake pad mass led to the disc exhibiting no stable, standing wave response (Figure 15). As with brake pads, highly regular, high frequency standing wave vibration is not desirable in the disc, as they are most likely to cause irritable noise. Furthermore, regular exposure highly regular, high amplitude standing waves are more likely to cause fatigue and reduction in structural strength.

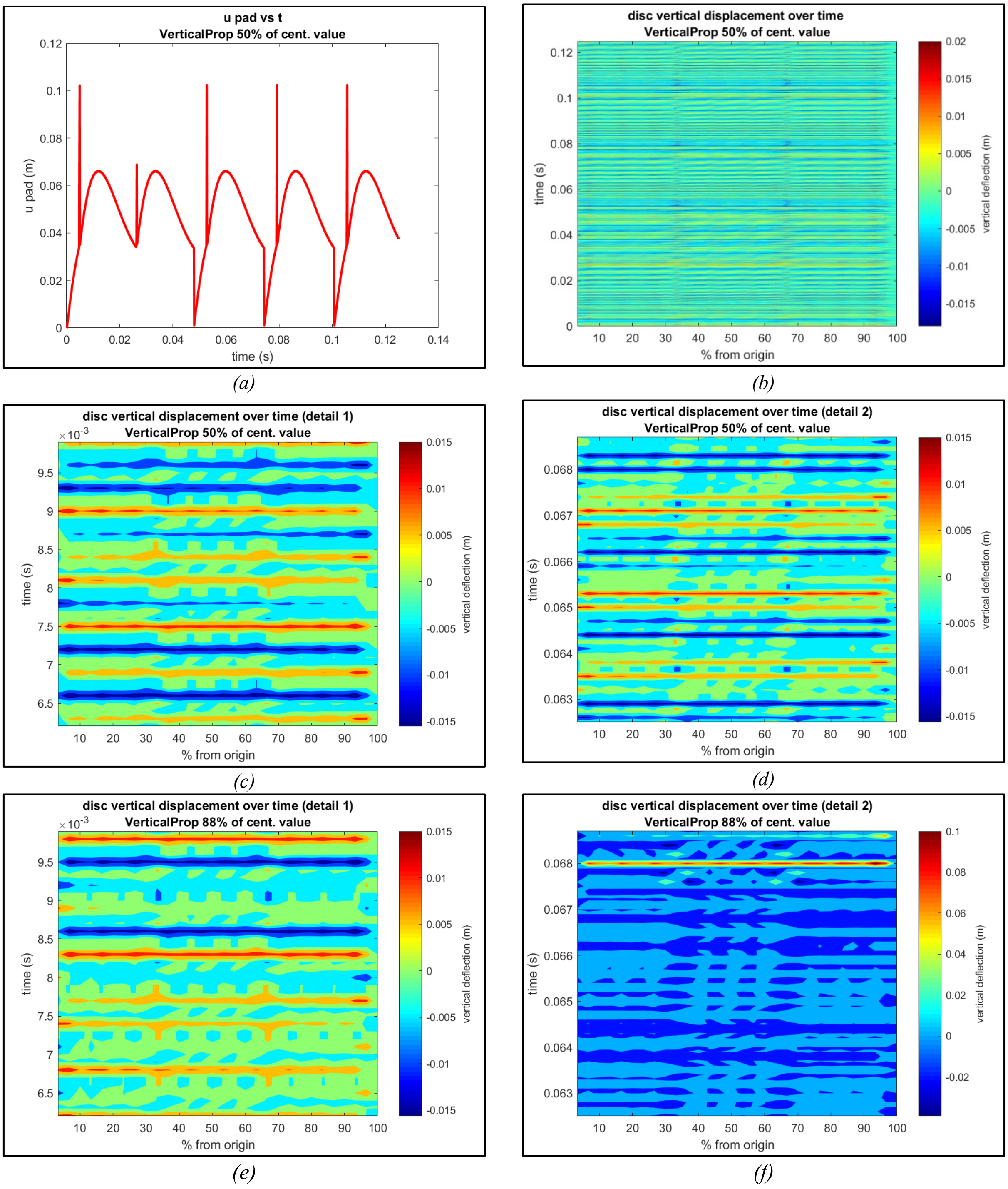


Figure 15 – Results of low vertical stiffness and damping. The brake pad vibrates irregularly and at predominantly low frequencies (a), while the disc exhibits little to no regularity in its vibration with the same parameters (b, c, d). Note also that the disc does not exhibit ramp-up with these parameters (b). Other values of low vertical stiffness and damping result in even less vibrational regularity in the disc (d, e).

4. Conclusions and Recommendations for Future Work

4.1. Conclusions

The developed model was used to carry out a relatively simple parameter variation study. The results of this parameter variation study indicate that the vibratory response of the system is highly dependent on its parametric combinations. Configurations with lower brake pad mass and higher horizontal stiffness and damping were found to cause both the brake pad and the disc to exhibit highly regular, high frequency vibrational modes that prior research has identified as being the culprit for generally undesirable brake noise. Furthermore, these vibrational modes can, over time, lead to the degradation of the mechanical properties of braking systems. Conversely, certain parametric combinations were found to result in both lower frequency and less regular vibration. Such vibration can be generalized as being less undesirable.

Specifically, it was found that reduction in stiffness and damping in the vertical direction of the model system results in more desirable vibratory response in both the brake pad and the disc. These findings mean that, in real-world braking systems, reduction in actuator stiffness and damping leads to a more desirable vibratory response. This result mirrors both the theoretical and experimental results obtained by other researchers [13], which serves to validate the fundamental principles of this model.

4.2. Recommendations for Future Work

As stated, this model is validated on only a fundamental, conceptual level. It has been demonstrated that, under parametric variation, the response of this model mirrors the response of both other widely accepted models and of real-world systems. However, it is not yet ready for use as a design tool. Because the a continuous disc body to a continuous bar body is not fully defined,

maximum displacement magnitudes are exceedingly large. Thus, it will be necessary to define this transformation in future work.

Similarly, on a cursory parameter variation study was carried out in this research for proof-of-concept and functionality testing purposes. A more complete validation of both the fundamental principles and practical functionality of this modeling method can be achieved through the conduction of a full “design of experiment”-style parameter variation study and comparing it against and industry standard tool, such as ANSYS DesignXplorer. This should only be undertaken after a definition of the disc-bar transformation is complete.

Appendices

[The remainder of this page is left intentionally blank.]

Appendix A. Symbols

Symbol	Value	Units
t	Time	s
x	Horizontal position	m
y	Vertical position	m
u_p	Brake pad horizontal displacement	m
$\dot{u}_p$	Brake pad horizontal velocity	m/s
$\ddot{u}_p$	Brake pad horizontal acceleration	m/s^2
v_p	Brake pad vertical displacement	m
$\dot{v}_p$	Brake pad vertical velocity	m/s
$\ddot{v}_p$	Brake pad vertical acceleration	m/s^2
u_d	Rotor horizontal position	m
$\dot{u}_d$	Rotor horizontal velocity	m/s
$\ddot{u}_d$	Rotor horizontal acceleration	m/s^2
δ	Rotor deformation	m
$\dot{\delta}$	Rate of rotor deformation	m/s
γ	Vertical rotor deformation	m
ρ	Density of the rotor	kg/m^3

Symbol	Value	Units
T	Thickness of the rotor	m
E	Modulus of elasticity of the rotor	Pa
A	Cross sectional area of the rotor	m^2
k_x	Horizontal stiffness of brake pad	N/m
k_y	Vertical stiffness of brake pad	N/m
$P(x,t)$	Force acting on the rotor	N
F_{px}	horizontal elastic reaction force of the pad	N
f_{μ_s}	magnitude of the force of static friction	N
F_N	Horizontal normal force acting on the brake pad	N
μ_s	Coefficient of static friction between brake pad and rotor	-
μ_k	Coefficient of kinetic friction between brake pad and rotor	-
m_p	Mass of the brake pad	kg
m_d	Mass of the rotor	kg
c_x	Horizontal damping coefficient of brake pad	-
ω_{px}	Horizontal natural frequency of brake pad	Hz
ω_{dpx}	Horizontal damped natural frequency of brake pad	Hz
ξ_x	Horizontal damping ratio of brake pad	-

Symbol	Value	Units
c_y	Vertical damping coefficient of brake pad	-
ω_{py}	Vertical natural frequency of brake pad	Hz
ω_{dpy}	Vertical damped natural frequency of brake pad	Hz
ξ_y	Vertical damping ratio of brake pad	-
Ω	Frequency of excitation of the brake pad	Hz
$\overline{\Omega}$	Ratio of excitation frequency to natural frequency	-
K	Kinetic energy of disc brake assembly	J
P	Potential energy of disc brake assembly	J

Appendix B. Core Matlab Code

This appendix contains the core Matlab code, as described in section 2.1 Core Matlab Model Structure (page 36).

[The remainder of this page is left intentionally blank.]

B.1. BrkVib.m

```
%Written by Michael Vladimirov
%University of New Haven
%Department of Mechanical Engineering
%Master of Science, Mechanical Engineering
%2019


clc
clear all
tic
%% Command Window Header
fprintf('--------------------------------------------------------------\n')
fprintf('**************************************************************\n')
fprintf('--------------------------------------------------------------\n')
fprintf('Disc Brake Vibration Modeling\n')
fprintf('\n')
fprintf('Michael Vladimirov\n')
fprintf('University of New Haven\n')
fprintf('Department of Mechanical Engineering\n')
fprintf('2019\n')
fprintf('--------------------------------------------------------------\n')
fprintf('**************************************************************\n')
fprintf('--------------------------------------------------------------\n')
%% Script Initiation
%This section initiates the script. It calls system parameters, derived
%vibration-related parameters from the system parameters, the initial
%conditions, and the simulation settings.
fprintf('SCRIPT INITIATION\n')
% Call system parameters
[udot_disc_init0,...
 g,...
 L,...
 rho,...
 T,...
 A,...
 E,...
 pois,...
 k_disc,...
 xp_loc,...
 k_x,...
 c_x,...
 k_y,...
 c_y,...
 m_pad,...
 m_disc,...
 m_stick,...
 mu_s,...
 mu_k,...
 F_brake] = SysPar();
fprintf('SysPar.m ok, loaded in %.01d s\n',toc)
fprintf('-------------------------------------------------------------\n')
```

```
%Call derived system properties
[w_x,...
 w2_x,...
 xi_x,...
 zi_x,...
 wd_x,...
 Period_x,...
 Period_damped_x,...
 freq_x,...
 freqd_x,...
 Wpk_x,...
 w_x_stick,...
 w2_x_stick,...
 xi_x_stick,...
 zi_x_stick,...
 wd_x_stick,...
 Period_x_stick,...
 Period_damped_x_stick,...
 freq_x_stick,...
 freqd_x_stick,...
 Wpk_x_stick,...
 w_y,...
 w2_y,...
 xi_y,...
 zi_y,...
 wd_y,...
 Period_y,...
 Period_damped_y,...
 freq_y,...
 freqd_y,...
 Wpk_y,...
 fdisc] = DerPar(k_x,...
                   c_x,...
                   m_pad,...
                   m_stick,...
                   k_y,...
                   c_y,...
                   L,...
                   E,...
                   A,...
                   rho);
fprintf('DerPar.m ok, loaded in %.01d s\n',toc)
fprintf('--------------------------------------------------------------\n')
```

```
% Call simulation settings
[n_max...
 xres,...
 xp_loc,...
 xsteps,...
 x_p,...
 j_xp,...
 tmax,...
 tres,...
 tsteps,...
 t]=SimSet(L,xp_loc);
fprintf('SimSet.m ok, loaded in %d s\n',toc)
fprintf('-------------------------------------------------------------\n')
fprintf('This analysis will model the system for %d',tmax)
fprintf(' seconds.\n')
fprintf('The time step resolution for this modeling is %d\n',tres)
fprintf('seconds.\n')
fprintf('This results in a total of .01%d timesteps.\n',tsteps)
fprintf('-------------------------------------------------------------\n')

% Call initial conditions
[u_disc,...
 udot_disc,...
 delta0,...
 delta_dot0,...
 u_pad,...
 udot_pad,...
 Fspring_x,...
 v_pad,...
 vdot_pad,...
 Fspring_y,...
 F_n,...
 Ffs,...
 Ffk,...
 StickOn,...
 Vdecay_x,...
 Vdecay_y,...
 x0,...
 v0,...
 time,...
 t_switch,...
 discy,...
 i_switch] = InitCond(tsteps,udot_disc_init0);
fprintf('InitCond.m ok, loaded in %.01d s\n',toc)
fprintf('--------------------------------------------------------------\n')
fprintf('**************************************************************\n')
fprintf('--------------------------------------------------------------\n')
```

```matlab
%% Vibration Loop

% fdisc=1;
fprintf('VIBRATION LOOP\n',toc)
for i=1:(size(t,2)-1)
    %%
    discy=sin(fdisc*t);
    %% Loop Header
    fprintf('___________')
    fprintf('Iteration: %d',i)
    fprintf(' of: %d',tsteps)
    fprintf('___________\n')
    %% Base Motion
        [v_pad(i),...
         Fnorm(i)] = BasMot(F_brake,...
                            k_y,...
                            v_pad(i),...
                            discy(i),...
                            fdisc,...
                            xi_y,...
                            w_y,...
                            t(i));

    fprintf('    BasMot.m ok\n')
    %% Stick Slip
    fcoul=(mu_k*Fnorm(i))/k_x;
    [u_disc(i+1),udot_disc(i+1),...
    u_pad(i+1),udot_pad(i+1),...
    StickOn(i+1),...
    time,t_switch,i_switch] = StkSlp(mu_k,...
                                    Fnorm,...
                                    k_x,...
                                    i,...
                                    u_pad,...
                                    udot_pad,...
                                    u_disc,...
                                    udot_disc,...
                                    m_pad,...
                                    m_disc,...
                                    x0,...
                                    v0,...
                                    tres,...
                                    w_x_stick,...
                                    wd_x_stick,...
                                    zi_x_stick,...
                                    zi_x,...
                                    w_x,...
                                    wd_x,...
                                    xi_x_stick,...
                                    StickOn,...
                                    time,...
                                    t_switch,...
                                    fcoul,...
                                    xi_x,...
                                    i_switch);
    fprintf('    StkSlp.m ok\n')
```

```matlab
    %% Bar Vibration
    [delta_3d,delta,...
            strain_axial,...
            strain_transverse,...
            deltay(:,i),...
            disc_y(i+1)] = BarVib(L,...
                                    n_max,...
                                    xsteps,...
                                    xres,...
                                    u_disc,...
                                    i,...
                                    time,...
                                    E,...
                                    rho,...
                                    A,...
                                    pois,...
                                    T,...
                                    j_xp);
    fprintf('    Time since analysis start: %0.01d s \n',toc)
end
```

B.2. SysPar.m

```
function [udot_disc_init0,...
            g,...
            L,...
            rho,...
            T,...
            A,...
            E,...
            pois,...
            k_disc,...
            xp_loc,...
            k_x,...
            c_x,...
            k_y,...
            c_y,...
            m_pad,...
            m_disc,...
            m_stick,...
            mu_s,...
            mu_k,...
            F_brake] = SysPar()
%Written by Michael Vladimirov
%University of New Haven
%Department of Mechanical Engineering
%Master of Science, Mechanical Engineering
%2019

%This function provides system paraters for the disc brake vibration script

%%
udot_disc_init0=10;              %initial disc speed
g=9.81;                           %gravity

%% Disc
L=0.15;                           %length
rho=8000;                         %density
T=0.01;                           %thickness
Width=0.05;                       %width
A=(Width*T);                      %area
E=200e9;                          %modulus elasticity
pois=0.3;                         %poisson ratio
k_disc=(A*E)/L;                   %equiv stiffness
xp_loc=0.5;                       %brake pad location, expressed as % of L

%% Masses
m_pad=1;                          %mass pad
m_disc=rho*L*A;                   %mass disc
m_stick=m_pad+m_disc;             %coupled disc and pad mass

%% x Spring Damper
k_x=29188;                            %spring stiffness
c_x=sqrt(k_x*m_pad);                  %damp.

%% y Spring Damper
k_y=29188;                            %spring stiffness
c_y=sqrt(k_y*m_pad);                  %damp.
```

```
%% Friction Coefs
mu_s=2;                          %static fric
mu_k=1;                          %kinetic fric

%% Applied Braking Force
F_brake=1000;                    %braking force


end
```

B.3. DerPar.m

```
%Written by Michael Vladimirov
%University of New Haven
%Department of Mechanical Engineering
%Master of Science, Mechanical Engineering
%2019

%   This function generates derived parameters for the brake vibration
%   script
function [w_x,...
            w2_x,...
            xi_x,...
            zi_x,...
            wd_x,...
            Period_x,...
            Period_damped_x,...
            freq_x,...
            freqd_x,...
            Wpk_x,...
            w_x_stick,...
            w2_x_stick,...
            xi_x_stick,...
            zi_x_stick,...
            wd_x_stick,...
            Period_x_stick,...
            Period_damped_x_stick,...
            freq_x_stick,...
            freqd_x_stick,...
            Wpk_x_stick,...
            w_y,...
            w2_y,...
            xi_y,...
            zi_y,...
            wd_y,...
            Period_y,...
            Period_damped_y,...
            freq_y,...
            freqd_y,...
            Wpk_y,...
            fdisc] = DerPar(k_x,...
                            c_x,...
                            m_pad,...
                            m_stick,...
                            k_y,...
                            c_y,...
                            L,...
                            E,...
                            A,...
                            rho)
```

```
% x-Direction Pad
%General Constant Values
w_x=sqrt(k_x/m_pad);                    %omega, natural freqency [rad/s]
w2_x=k_x/m_pad;                         %omega squared
xi_x=c_x/(2*w_x*m_pad);                 %squig, damping ratio
zi_x=sqrt((xi_x^2)-1);                  %freq. modifier for overdamped
wd_x=w_x*sqrt(1-(xi_x^2));              %omega sub d, damped nat. freq.
Period_x=(2*pi)/w_x;                    %undamped period
Period_damped_x=(2*pi)/wd_x;            %undamped period
freq_x=1/Period_x;                      %undamped system frequency [Hz]
freqd_x=1/Period_damped_x;              %undamped system frequency [Hz]
Wpk_x=sqrt(1-(2*(xi_x^2)));             %peak frequency
%Stick Values
w_x_stick=sqrt(k_x/(m_stick));              %Stick phase omega, natural
freqency [rad/s]
w2_x_stick=k_x/m_stick;                     %Stick phase omega squared
xi_x_stick=c_x/(2*w_x_stick*m_stick);       %Stick phase squig, damping
ratio
zi_x_stick=sqrt((xi_x_stick^2)-1);          %Stick phase freq. modifier for
overdamped
wd_x_stick=w_x_stick*sqrt(1-(xi_x_stick^2)); %Stick phase omega sub d, damped
nat. freq.
Period_x_stick=(2*pi)/w_x_stick;            %Stick phase undamped period
Period_damped_x_stick=(2*pi)/wd_x;          %Stick phase undamped period
freq_x_stick=1/Period_x;                    %Stick phase undamped system
frequency [Hz]
freqd_x_stick=1/Period_damped_x;            %Stick phase undamped system
frequency [Hz]
Wpk_x_stick=sqrt(1-(2*(xi_x_stick^2)));     %Stick phase peak frequency

% y-Direction Pad
%General Constant Values
w_y=sqrt(k_y/m_pad);                        %omega, natural freqency [rad/s]
w2_y=k_y/m_pad;                             %omega squared
xi_y=c_y/(2*w_y*m_pad);                     %squig, damping ratio
zi_y=sqrt((xi_y^2)-1);                      %freq. modifier for overdamped
wd_y=w_y*sqrt(1-(xi_y^2));                  %omega sub d, damped nat. freq.
Period_y=(2*pi)/w_y;                        %undamped period
Period_damped_y=(2*pi)/wd_y;                %undamped period
freq_y=1/Period_y;                          %undamped system frequency [Hz]
freqd_y=1/Period_damped_y;                  %undamped system frequency [Hz]
Wpk_y=sqrt(1-(2*(xi_y^2)));                 %peak frequency

% Disc
%General Constant Values
fdisc=(pi/L)*sqrt(E/rho);    %natural freqency of the disc [rad/s]
end
```

B.4. SimSet.m

```
function [n_max...
            xres,...
            xp_loc,...
            xsteps,...
            x_p,...
            j_xp,...
            tmax,...
            tres,...
            tsteps,...
            t]=SimSet(L,xp_loc)
%Written by Michael Vladimirov
%University of New Haven
%Department of Mechanical Engineering
%Master of Science, Mechanical Engineering
%2019

%This function provides simulation settings for the disc brake vibration
%script

n_max=10;                        %summation order
xres=0.01;                       %spatial resolution
xsteps=L/xres;                   %number spatial steps
x_p=L*xp_loc;                    %location of brake pad
j_xp=round(xp_loc*xsteps);       %spatial index of brake pad location

tmax=.125;                          %simulation time lenght
tres=0.0001;                        %temporal resolution
tsteps=(tmax/tres)+1;                %number time steps

t=0:tres:tmax;                   %seed value time step index
end
```

B.5. InitCond.m

```
function [u_disc,...
            udot_disc,...
            delta0,...
            delta_dot0,...
            u_pad,...
            udot_pad,...
            Fspring_x,...
            v_pad,...
            vdot_pad,...
            Fspring_y,...
            F_n,...
            Ffs,...
            Ffk,...
            StickOn,...
            Vdecay_x,...
            Vdecay_y,...
            x0,...
            v0,...
            time,...
            t_switch,...
            discy,...
            i_switch] = InitCond(tsteps,udot_discinit0);
%Written by Michael Vladimirov
%University of New Haven
%Department of Mechanical Engineering
%Master of Science, Mechanical Engineering
%2019

%This function provides initial conditions for the disc brake vibration
%script


%% Preallocation with zeros
% Disc
u_disc=zeros(1,tsteps);%disc position
udot_disc=zeros(1,tsteps);%disc velocity
delta0=0;%preallocation
delta_dot0=0;%preallocation
discy=0;

% x Direction
u_pad=zeros(1,tsteps);%x pad position
udot_pad=zeros(1,tsteps);%x pad velocity
Fspring_x=zeros(1,tsteps);%x-direction spring force
i_switch=1;

% y Direction
v_pad=zeros(1,tsteps);%y pad position
vdot_pad=zeros(1,tsteps);%y pad velocity
Fspring_y=zeros(1,tsteps);%y-direction spring force
```

```
% Friction
F_n=zeros(1,tsteps);%normal force
Ffs=zeros(1,tsteps);%static friction magnitude
Ffk=zeros(1,tsteps);%kinetic friction magnitude


% Stick-Slip
StickOn=zeros(1,tsteps);%=0 during stick phase, =1 during slip phase
%v0=zeros(1,tsteps);%combined disc-pad velocity for stick phase
time=0;%cyclical time
t_switch=0;%stick slip switch time



%% Define non-zero values for first time step
StickOn(1)=1;
x0=u_disc(1);%initial position for stick-slip system
udot_disc(1)=udot_discinit0;%disc velocity
v0(1)=udot_disc(1);%initial velocity for stick-slip system
Vdecay_x=0;%exponential viscious decay
Vdecay_y=0;%exponential viscious decay
end
```

B.6. BasMot.m

```
function [v_pad,...
            Fnorm] = BasMot(F_brake,...
                            k_y,...
                            v_pad,...
                            discy,...
                            fdisc,...
                            xi_y,...
                            w_y,...
                            t)
%Written by Michael Vladimirov
%University of New Haven
%Department of Mechanical Engineering
%Master of Science, Mechanical Engineering
%2019

%This function considers the base motion of the disc brake system.

%Time Independent Base Motion Values
WbarB=fdisc/w_y;%base freq to natural freq ratio
GammaB=(((1-(WbarB^2))^2)+((2*WbarB*xi_y)^2))^(-1/2);%magnification factor
gammatr= GammaB*sqrt(1+((2*WbarB*xi_y)^2));%transmissibility
PsiB=atan((2*xi_y*(WbarB^3))/((1-(WbarB^2))+((2*xi_y*WbarB)^2)));%ss resp.
offset
%Time Dependent Values
inB=(fdisc*t)-PsiB;%......................Exponential argument for ss
response

%Base Motion
v_pad=discy.*gammatr.*exp(i.*(inB));%.....ss response to applied force

%Normal Force
Fnorm=BasMot1(F_brake,k_y,v_pad);
end
```

B.7. BasMot1.m

```
function [Fnorm] = BasMot1(F_brake,k_y,v_pad)
%Written by Michael Vladimirov
%University of New Haven
%Department of Mechanical Engineering
%Master of Science, Mechanical Engineering
%2019

%This function calcluates the normal force acting on the disc

Fnorm = F_brake+(k_y*v_pad);
end
```

B.8. StkSlp.m

```
%Written by Michael Vladimirov
%University of New Haven
%Department of Mechanical Engineering
%Master of Science, Mechanical Engineering
%2019

%This is the top-level function that consideres the stick slip behavior of
%the stick-slip behavior of the disc brake system.


function [u_disc_out,udot_disc_out,...
            u_pad_out,udot_pad_out,...
            StickOn,...
            time,t_switch,i_switch] = StkSlp(mu_k,...
                                              Fnorm,...
                                              k_x,...
                                              i,...
                                              u_pad_in,...
                                              udot_pad_in,...
                                              u_disc_in,...
                                              udot_disc_in,...
                                              m_pad,...
                                              m_disc,...
                                              x0,...
                                              v0,...
                                              tres,...
                                              w_x_stick,...
                                              wd_x_stick,...
                                              zi_x_stick,...
                                              zi_x,...
                                              w_x,...
                                              wd_x,...
                                              xi_x_stick,...
                                              StickOn,...
                                              time,...
                                              t_switch,...
                                              fcoul,...
                                              xi_x,...
                                              i_switch)

%% StkSlp1 Determine magnitude of static friction force
Ffs = StkSlp1(mu_k,Fnorm);
fprintf('        StkSlp1.m ok\n')
%% StkSlp2 Determine magnitude of spring force
Fspring(i) = StkSlp2(u_pad_in,k_x,i);
fprintf('        StkSlp2.m ok\n')
%% StkSlp3 Determine whether system is in stick or slip state
StickOn(i) = StkSlp3(Ffs(i),Fspring(i));
fprintf('        StkSlp3.m ok\n')
```

```
%% Define time w/in cycle
%Determine if current time step is the first time step, if so, assign an
%initial value to "time", which is the temporal variable used by the
%function that models stick-slip oscillation (StkSlp5)
if i==1
    time=0;
end

%% StkSlp4 Determine cyclical initial conditions
[x0,v0,time,t_switch,i_switch] = StkSlp4(StickOn,...
                                        u_pad_in(i),...
                                        udot_pad_in(i),...
                                        udot_disc_in(i),...
                                        m_pad,...
                                        m_disc,...
                                        x0,...
                                        v0,...
                                        tres,...
                                        time,...
                                        i,...
                                        t_switch,...
                                        i_switch);
fprintf('        StkSlp4.m ok\n')
%% StkSlp5 Stick-slip oscillation modeling
[u_disc_SS,udot_disc_SS,...
      u_pad_SS,udot_pad_SS] = StkSlp5(xi_x_stick,...
                                 x0,...
                                 v0,...
                                 w_x,...
                                 wd_x,...
                                 w_x_stick,...
                                 wd_x_stick,...
                                 time,...
                                 zi_x_stick,...
                                 zi_x,...
                                 mu_k,...
                                 Fnorm,...
                                 k_x,...
                                 i,...
                                 u_disc_in,...
                                 u_pad_in,...
                                 udot_disc_in,...
                                 udot_pad_in,...
                                 StickOn(i),...
                                 tres,...
                                 t_switch,...
                                 fcoul,...
                                 xi_x,...
                                 i_switch);
fprintf('        StkSlp5.m ok\n')
%% Convert StickOn output from array to single scalar
StickOn=StickOn(i);
u_disc_out=u_disc_SS;
udot_disc_out=udot_disc_SS;
u_pad_out=u_pad_SS;
udot_pad_out=udot_pad_SS;
end
```

B.9. StkSlp1.m

```
function [Ffs] = StkSlp1(mu_k,Fnorm)
%Written by Michael Vladimirov
%University of New Haven
%Department of Mechanical Engineering
%Master of Science, Mechanical Engineering
%2019

%Determine magnitude of static friction force

Ffs=mu_k*Fnorm;
end
```

B.10. StkSlp2.m

```
function [Fspring] = StkSlp2(u_pad,k_x,i)
%Written by Michael Vladimirov
%University of New Haven
%Department of Mechanical Engineering
%Master of Science, Mechanical Engineering
%2019

%Determine magnitude of spring force

Fspring=u_pad(i)*k_x;
end
```

B.11. StkSlp3.m

```
function [StickOn] = StkSlp3(Ffs,Fspring)
%Written by Michael Vladimirov
%University of New Haven
%Department of Mechanical Engineering
%Master of Science, Mechanical Engineering
%2019

%Determine whether system is in stick or slip state by comparing magnitude
%of spring force and static friction.


if i==1
    StickOn=1;
else
    if Ffs>Fspring
        StickOn=1;
    else
        StickOn=0;
    end
end
end
```

B.12. StkSlp4.m

```
function [x0,v0,time,t_switch,i_switch] = StkSlp4(StickOn,...
                                                    u_pad,...
                                                    udot_pad,...
                                                    udot_disc,...
                                                    m_pad,...
                                                    m_disc,...
                                                    x0_old,...
                                                    v0_old,...
                                                    tres,...
                                                    time_old,...
                                                    i,...
                                                    tswitch,...
                                                    iswitch)
%Written by Michael Vladimirov
%University of New Haven
%Department of Mechanical Engineering
%Master of Science, Mechanical Engineering
%2019

%This function determines cyclical initial conditions for the stick and
%slip cycles
```

```
if i==1
    %for the first timestep:
    time=0;
    t_switch=i*tres;
    i_switch=iswitch;
    x0=u_pad;
    v0=sqrt((m_disc*(udot_pad^2))/(m_pad+m_disc));
else
    %for all timesteps AFTER the first timestep
    if StickOn(i)-StickOn(i-1)~=0
        %if this function senses that the value of "StickOn", the binary
        %inticator of whether the system is in a state of stick or slip,
        %changes value over a time step, then the values of the initial
        %conditions used in StkSlp5 get updated
        time=0;
        t_switch=i*tres;
        i_switch=i;
        if StickOn==1
            %if the system has switched into a stick state:
            x0=u_pad;
            v0=sqrt((m_disc*(udot_disc^2))/(m_pad+m_disc));
        else
            %if the system has switched into a slip state:
            x0=u_pad;
            v0=udot_pad;
        end
    else
        %if no change is detected, then the values remain the same
        x0=x0_old;
        v0=v0_old;
        time=time_old+tres;
        t_switch=tswitch;
        i_switch=iswitch;
    end
end
end
```

B.13. StkSlp5.m

```
%Written by Michael Vladimirov
%University of New Haven
%Department of Mechanical Engineering
%Master of Science, Mechanical Engineering
%2019

%This function models stick-slip oscillation

function [u_disc_SS,udot_disc_SS,...
            u_pad_SS,udot_pad_SS] = StkSlp5(xi_x_stick,...
                                             x0,...
                                             v0,...
                                             w_x,...
                                             wd_x,...
                                             w_x_stick,...
                                             wd_x_stick,...
                                             time,...
                                             zi_x_stick,...
                                             zi_x,...
                                             mu_k,...
                                             Fnorm,...
                                             k_x,...
                                             i,...
                                             u_disc_in,...
                                             u_pad_in,...
                                             udot_disc_in,...
                                             udot_pad_in,...
                                             StickOn,...
                                             tres,...
                                             t_switch,...
                                             fcoul,...
                                             xi_x,...
                                             i_switch)
```

```
%% stick slip displacement
if StickOn==1 %Stick cycle modeling
     u_pad_SS = StkSlp5a(xi_x_stick,...
                                 x0,...
                                 v0,...
                                 w_x_stick,...
                                 wd_x_stick,...
                                 time,...
                                 zi_x_stick);
    fprintf('        StkSlp5a.m ok\n')
    u_disc_SS=u_pad_SS;
else %Slip cycle modeling
     [u_pad_SS] = StkSlp5b(xi_x,...
                                 x0,...
                                 v0,...
                                 w_x,...
                                 wd_x,...
                                 time,...
                                 zi_x,...
                                 mu_k,...
                                 Fnorm,...
                                 k_x,...
                                 fcoul);
    fprintf('            StkSlp5b.m ok\n')
end
%% calculate disc and pad velocities
%initial values of disc velocity for slip phase linear reduction of disc
%velocity, as well as posoitions of pad and disc
if i==1
    udot_disc_switch=udot_disc_in(i);
    u_disc_old=u_disc_in(i);
    u_pad_old=u_pad_in(i);
else
    udot_disc_switch=udot_disc_in(i_switch);
    u_disc_old=u_disc_in(i-1);
    u_pad_old=u_pad_in(i-1);
end
%disc and pad velocities loop
[udot_disc_SS,udot_pad_SS] = StkSlp5c(i,...
                                         u_disc_in(i),...
                                         u_pad_in(i),...
                                         u_disc_old,...
                                         u_pad_old,...
                                         udot_disc_in(1),...
                                         udot_pad_in(1),...
                                         tres,...
                                         StickOn,...
                                         udot_disc_switch,...
                                         fcoul);
    fprintf('            StkSlp5c.m ok\n')

%calculate disc position if system is in slip state
if StickOn==0
    u_disc_SS=u_disc_old+(udot_disc_SS*tres);

end
```

B.14. StkSlp5a.m

```
%Written by Michael Vladimirov
%University of New Haven
%Department of Mechanical Engineering
%Master of Science, Mechanical Engineering
%2019

%This function the STICK cycle of stick-slip oscillation

function [u_pad_SS] = StkSlp5a(xi_x_stick,...
                                     x0,...
                                     v0,...
                                     w_x_stick,...
                                     wd_x_stick,...
                                     time,...
                                     zi_x_stick)

Vdecay=exp(-xi_x_stick*w_x_stick*time);%exponential viscious decay

% if (xi_x_stick^2)<1%underdamped case
    A1=x0;
    A2=(v0+(x0*xi_x_stick*w_x_stick))...
        /...
        (wd_x_stick);
    xc=Vdecay...
        .*...
        ((A1.*cos(wd_x_stick*time))+(A2.*sin(wd_x_stick*time)));
% elseif (xi_x_stick^2)==1%critically damped case
%     A1=x0;
%     A2=v0+(A1*w_x_stick);
%     xc=(A1+(A2.*time)).*exp((-w_x_stick).*time);
% else (xi_x_stick^2)>1%overdamped case
%     A1=x0;
%     A2= ((v0/w_x_stick)+(xi*x0))...
%         /...
%         zi_x_stick;
%     xc=Vdecay...
%         .*...
%         (...
%             (A1.*cosh(zi_x_stick*wd_x_stick*time))...
%             +...
%             (A2.*sinh(zi_x_stick*wd_x_stick*time))...
%         );
% end

u_pad_SS=xc;

end
```

B.15. StkSlp5b.m

```
%Written by Michael Vladimirov
%University of New Haven
%Department of Mechanical Engineering
%Master of Science, Mechanical Engineering
%2019

%This function the SLIP cycle of stick-slip oscillation

function [u_pad_SS] = StkSlp5b(xi_x,...
                                x0,...
                                v0,...
                                w_x,...
                                wd_x,...
                                time,...
                                zi_x,...
                                mu_k,...
                                Fnorm,...
                                k_x,...
                                fcoul)


Vdecay=exp(-xi_x*w_x*time);%exponential viscious decay

% if (xi_x^2)<1%underdamped case
    A1=x0;
    A2=(v0+(x0*xi_x*w_x))...
        /...
        (wd_x);
    xc=(Vdecay...
        .*...
        ((A1.*cos(wd_x*time))+(A2.*sin(wd_x*time))))...
        +...
        fcoul;
% elseif (xi_x^2)==1%critically damped case
%     A1=x0;
%     A2=v0+(A1*w_x);
%     xc=(A1+(A2.*time)).*exp((-w_x).*time);
% else (xi_x^2)>1%overdamped case
%     A1=x0;
%     A2= ((v0/w_x)+(xi*x0))...
%         /...
%         zi_x;
%     xc=Vdecay...
%         .*...
%         (...
%             (A1.*cosh(zi_x*wd_x*time))...
%             +...
%             (A2.*sinh(zi_x*wd_x*time))...
%         );
% end

u_pad_SS=xc;

end
```

B.16. StkSlp5c.m

```
%Written by Michael Vladimirov
%University of New Haven
%Department of Mechanical Engineering
%Master of Science, Mechanical Engineering
%2019

%This function calculates the velocities of the pad and disc during
%stick-slip motion
function [udot_disc,udot_pad] = StkSlp5c(i,...
                                            u_disc,...
                                            u_pad,...
                                            u_disc_old,...
                                            u_pad_old,...
                                            u_disc_0,...
                                            u_pad_0,...
                                            tres,...
                                            StickOn,...
                                            udot,...
                                            fcoul)

%Pad velocity
udot_pad=tDer(i,u_pad,u_pad_old,tres,u_pad_0);

%Disc velocity
if StickOn==1
    udot_disc=tDer(i,u_disc,u_disc_old,tres,u_disc_0);
else
    udot_disc=udot-fcoul;
end



end
```

B.17. tDer.m

```
%Written by Michael Vladimirov
%University of New Haven
%Department of Mechanical Engineering
%Master of Science, Mechanical Engineering
%2019

%This function calculates the time derivative of the input vector. During
%the first time step, it relies on a provided initial value.

function [xdot] = tDer(i,xnow,xold,tres,xdot_0)

if i>1
    xdot=(xnow-xold)/tres;
else
    xdot=xdot_0;
end
```

B.18. BarVib.m

```
%Written by Michael Vladimirov
%University of New Haven
%Department of Mechanical Engineering
%Master of Science, Mechanical Engineering
%2019

%This function models disc deformation

function [delta_3d,delta,...
            strain_axial,...
            strain_transverse,...
            deltay,...
            discy] = BarVib(L,...
                            n_max,...
                            xsteps,...
                            xres,...
                            u_disc,...
                            i,...
                            time,...
                            E,...
                            rho,...
                            A,...
                            pois,...
                            T,...
                            j_xp)
    P=10;
    delta0=((-P/(E*A))*(0:xres:xsteps))+((P*L)/(E*A));
    delta_dot0=0;
```

```
%% Disc Deformation Loop
for j=1:(xsteps)     %spatial loop
    x=j*xres;       %determine current value of x being evaluated
    %% Disc Deformation Summation Loop
    %% AXIAL:
    for n=1:n_max   %summation loop, evaluates summation for given value of x
        delta_pre(n)=...
            (...
                        sin((((n*pi)/L)*x)+((2*pi)/u_disc(i)) )...
%spatial portion, includes fi as "+ ((2*pi)/u_disc)"
            *...
                (...
%begin temporal portion
                    (...
                        (...
                            (...
                                trapz((sin((((n*pi)/L)*x)))*(delta0)))...
%C1num
                        /...
                            ((L/2)-((sin(2*n*pi))/(4*((n*pi)/L))))...
%C1denom
                        )...
                    *...
                        cos( (((n*pi)/L)*(sqrt(E/rho))) *time)...
%temporal cosine
                    )...
                +...
                    (...
                        (...
                            (trapz((sin((((n*pi)/L)*x)))*(delta_dot0)))...
%C2num
                        /...
                            ((L/2)-((sin(2*n*pi))/(4*((n*pi)/L))))...
%C2denom
                        )...
                    *...
                        sin( (((n*pi)/L)*(sqrt(E/rho))) *time)...
%temporal sine
                    )...
                )...
            )...
        ;


    delta_3d(j,n)=delta_pre(n);%array; x-t 2d domain w/ "stacks" of delta_pre
    end         %summation loop close
    delta(j,:)=sum(delta_pre);
    %end disc AXIAL deformation loop
    %% TRANSVERSE:
    strain_axial=delta./x;
    strain_transverse=strain_axial*pois;
    deltay=strain_transverse*T;


end             %spatial loop close
discy=deltay(j_xp,:);
end
```

Appendix C. Parameter Variation Matlab Code

As discussed in section 2.2 Parameter Variation (page 39), the majority of the parameter variation Matlab code is identical to core Matlab code. For this reason, this appendix only contains code that differs from the core Matlab code (provided in Appendix B Core Matlab Code page 52).

[The remainder of this page is left intentionally blank.]

C.1. ParameterStudy.m

```
%Written by Michael Vladimirov
%University of New Haven
%Department of Mechanical Engineering
%Master of Science, Mechanical Engineering
%2019


clc
clear all
tic
%% Command Window Header
fprintf('--------------------------------------------------------------\n')
fprintf('**************************************************************\n')
fprintf('--------------------------------------------------------------\n')
fprintf('Disc Brake Vibration Modeling\n')
fprintf('\n')
fprintf('Michael Vladimirov\n')
fprintf('University of New Haven\n')
fprintf('Department of Mechanical Engineering\n')
fprintf('2019\n')
fprintf('--------------------------------------------------------------\n')
fprintf('**************************************************************\n')
fprintf('--------------------------------------------------------------\n')
fprintf('Begin Parameter Study\n')
fprintf('--------------------------------------------------------------\n')
```

```matlab
%% Define System Parameters

%Variation Vector
variation=0.5:0.125:1.5;
%this variation vector is used to create 9 values of system parameters,
%+/-50% from a central value, all with a difference of 12.5% of the central
%value; i.e. the central value is the 5th value of the resulting array


%
udot_disc_init0=10;            %initial disc speed
g=9.81;                         %gravity

% Disc
L=0.15;                         %length
rho=8000;                       %density
T=0.01;                         %thickness
Width=0.05;                     %width
A=(Width*T);                    %area
E=200e9;                        %modulus elasticity
pois=0.3;                       %poisson ratio
k_disc=(A*E)/L;                 %equiv stiffness
xp_loc=0.5;                     %brake pad location, expressed as % of L

% Masses
m_pad=1*variation;                      %mass pad
m_disc=rho*L*A;                 %mass disc
m_stick=m_pad+m_disc;           %coupled disc and pad mass

% x Spring Damper

k_x=29188*variation;                        %spring stiffness
c_x=sqrt(k_x.*m_pad);                   %damp.

% y Spring Damper
k_y=29188*variation;                        %spring stiffness
c_y=sqrt(k_y.*m_pad);                   %damp.

% Friction Coefs
mu_s=2;                         %static fric
mu_k=1;                         %kinetic fric

% Applied Braking Force
F_brake=1000;                   %braking force
fprintf('Parameters loaded\n')
fprintf('--------------------------------------------------------------\n')
```

```
%% Simulation
%Indexing convention:
%i corresponds to variation in pad mass
%j corresponds to variation in pad horizontal properties
%k corresponds to variation in pad vertical properties

%% Test - no variation in system parameters
fprintf('Begin Test\n')
%Indexing:
i=5;
j=5;
k=5;

%Simulation Loop:

ParameterStudied=sprintf('Test');
ParameterValue=100;
[u_disc_test,...
 u_pad_test,...
 v_pad_test,...
 disc_y_test,...
 t,...
 deltay_test] = BrkVib_Param(udot_disc_init0,...
                                g,...
                                L,...
                                rho,...
                                T,...
                                A,...
                                E,...
                                pois,...
                                k_disc,...
                                xp_loc,...
                                k_x(j),...
                                c_x(j),...
                                k_y(k),...
                                c_y(k),...
                                m_pad(i),...
                                m_disc,...
                                m_stick(i),...
                                mu_s,...
                                mu_k,...
                                F_brake,...
                                ParameterStudied,...
                                ParameterValue);
fprintf('Test complete\n')
fprintf('-------------------------------------------------------------\n')
```

```
%% Pad Mass Variation
fprintf('Begin Pad Mass Variation\n')
%Indexing:
i=5;
j=5;
k=5;

%Simulation Loop:
for i=1:9
    ParameterStudied=sprintf('PadMass');
    ParameterValue=round((12.5*i)+37.5);
    [u_disc_mass(i,:),...
     u_pad_mass(i,:),...
     v_pad_mass(i,:),...
     disc_y_mass(i,:),...
     t,...
     deltay_mass(:,:,i)] = BrkVib_Param(udot_disc_init0,...
                                        g,...
                                        L,...
                                        rho,...
                                        T,...
                                        A,...
                                        E,...
                                        pois,...
                                        k_disc,...
                                        xp_loc,...
                                        k_x(j),...
                                        c_x(j),...
                                        k_y(k),...
                                        c_y(k),...
                                        m_pad(i),...
                                        m_disc,...
                                        m_stick(i),...
                                        mu_s,...
                                        mu_k,...
                                        F_brake,...
                                        ParameterStudied,...
                                        ParameterValue);
    fprintf('     Pad Mass Variation Iteration %d\n',i)
end
fprintf('Pad Mass Variation complete\n')
fprintf('--------------------------------------------------------------\n')
```

```
%% Pad Horizontal Properties Variation
fprintf('Begin Pad Horizontal Properties Variation\n')
%Indexing:
i=5;
j=5;
k=5;

%Simulation Loop:
for j=1:9
    ParameterStudied=sprintf('HorizontalProp');
    ParameterValue=round((12.5*j)+37.5);
    [u_disc_horiz(j,:),...
     u_pad_horiz(j,:),...
     v_pad_horiz(j,:),...
     disc_y_horiz(j,:),...
     t...
     deltay_horiz(:,:,j)] = BrkVib_Param(udot_disc_init0,...
                                         g,...
                                         L,...
                                         rho,...
                                         T,...
                                         A,...
                                         E,...
                                         pois,...
                                         k_disc,...
                                         xp_loc,...
                                         k_x(j),...
                                         c_x(j),...
                                         k_y(k),...
                                         c_y(k),...
                                         m_pad(i),...
                                         m_disc,...
                                         m_stick(i),...
                                         mu_s,...
                                         mu_k,...
                                         F_brake,...
                                         ParameterStudied,...
                                         ParameterValue);
    fprintf('     Pad Horizontal Properties Variation Iteration %d\n',j)
end
fprintf('Pad Horizontal Properties Variation complete\n')
fprintf('--------------------------------------------------------------\n')
```

```matlab
%% Pad Vertical Properties Variation
fprintf('Begin Pad Vertical Properties Variation\n')
%Indexing:
i=5;
j=5;
k=5;

%Simulation Loop:
for k=1:9
    ParameterStudied=sprintf('VerticalProp');
    ParameterValue=round((12.5*k)+37.5);
    [u_disc_vert(k,:),...
     u_pad_vert(k,:),...
     v_pad_vert(k,:),...
     disc_y_vert(k,:),...
     t...
     deltay_vert(:,:,k)] = BrkVib_Param(udot_disc_init0,...
                                         g,...
                                         L,...
                                         rho,...
                                         T,...
                                         A,...
                                         E,...
                                         pois,...
                                         k_disc,...
                                         xp_loc,...
                                         k_x(j),...
                                         c_x(j),...
                                         k_y(k),...
                                         c_y(k),...
                                         m_pad(i),...
                                         m_disc,...
                                         m_stick(i),...
                                         mu_s,...
                                         mu_k,...
                                         F_brake,...
                                         ParameterStudied,...
                                         ParameterValue);
    fprintf('     Pad Vertical Properties Variation Iteration %d\n',k)
end
fprintf('Pad Vertical Properties Variation complete\n')
fprintf('---------------------------------------------------------------\n')

fprintf('Parameter Study complete\n')
fprintf('---------------------------------------------------------------\n')
%%
SimDuration=toc
```

C.2. BrkVib_Param.m

```
function [u_disc,...
            u_pad,...
            v_pad,...
            disc_y,...
            t,...
            deltay] = BrkVib_Param(udot_disc_init0,...
                                   g,...
                                   L,...
                                   rho,...
                                   T,...
                                   A,...
                                   E,...
                                   pois,...
                                   k_disc,...
                                   xp_loc,...
                                   k_x,...
                                   c_x,...
                                   k_y,...
                                   c_y,...
                                   m_pad,...
                                   m_disc,...
                                   m_stick,...
                                   mu_s,...
                                   mu_k,...
                                   F_brake,...
                                   ParameterStudied,...
                                   ParameterValue)
%Written by Michael Vladimirov
%University of New Haven
%Department of Mechanical Engineering
%Master of Science, Mechanical Engineering
%2019


% clc
% clear all
% tic
%% Command Window Header
% fprintf('---------------------------------------------------------------\n')
% fprintf('***************************************************************\n')
% fprintf('---------------------------------------------------------------\n')
% fprintf('Disc Brake Vibration Modeling\n')
% fprintf('\n')
% fprintf('Michael Vladimirov\n')
% fprintf('University of New Haven\n')
% fprintf('Department of Mechanical Engineering\n')
% fprintf('2019\n')
% fprintf('---------------------------------------------------------------\n')
% fprintf('***************************************************************\n')
% fprintf('---------------------------------------------------------------\n')
```

```
%% Script Initiation
%This section initiates the script. It calls system parameters, derived
%vibration-related parameters from the system parameters, the initial
%conditions, and the simulation settings.
% fprintf('SCRIPT INITIATION\n')
% Call system parameters
% [udot_disc_init0,...
%  g,...
%  L,...
%  rho,...
%  T,...
%  A,...
%  E,...
%  pois,...
%  k_disc,...
%  xp_loc,...
%  k_x,...
%  c_x,...
%  k_y,...
%  c_y,...
%  m_pad,...
%  m_disc,...
%  m_stick,...
%  mu_s,...
%  mu_k,...
%  F_brake] = SysPar_Param();
% fprintf('SysPar.m ok, loaded in %.01d s\n',toc)
% fprintf('-----------------------------------------------------------------\n')
%Call derived system properties
[w_x,...
 w2_x,...
 xi_x,...
 zi_x,...
 wd_x,...
 Period_x,...
 Period_damped_x,...
 freq_x,...
 freqd_x,...
 Wpk_x,...
 w_x_stick,...
 w2_x_stick,...
 xi_x_stick,...
 zi_x_stick,...
 wd_x_stick,...
 Period_x_stick,...
 Period_damped_x_stick,...
 freq_x_stick,...
 freqd_x_stick,...
 Wpk_x_stick,...
 w_y,...
 w2_y,...
 xi_y,...
 zi_y,...
 wd_y,...
 Period_y,...
 Period_damped_y,...
 freq_y,...
```

```
 freqd_y,...
 Wpk_y,...
 fdisc] = DerPar(k_x,...
                    c_x,...
                    m_pad,...
                    m_stick,...
                    k_y,...
                    c_y,...
                    L,...
                    E,...
                    A,...
                    rho);
% fprintf('DerPar.m ok, loaded in %.01d s\n',toc)
% fprintf('-------------------------------------------------------------\n')
% Call simulation settings
[n_max...
 xres,...
 xp_loc,...
 xsteps,...
 x_p,...
 j_xp,...
 tmax,...
 tres,...
 tsteps,...
 t]=SimSet(L,xp_loc);
% fprintf('SimSet.m ok, loaded in %d s\n',toc)
% fprintf('-------------------------------------------------------------\n')
% fprintf('This analysis will model the system for %d',tmax)
% fprintf(' seconds.\n')
% fprintf('The time step resolution for this modeling is %d\n',tres)
% fprintf('seconds.\n')
% fprintf('This results in a total of .01%d timesteps.\n',tsteps)
% fprintf('-------------------------------------------------------------\n')

% Call initial conditions
[u_disc,...
 udot_disc,...
 delta0,...
 delta_dot0,...
 u_pad,...
 udot_pad,...
 Fspring_x,...
 v_pad,...
 vdot_pad,...
 Fspring_y,...
 F_n,...
 Ffs,...
 Ffk,...
 StickOn,...
 Vdecay_x,...
 Vdecay_y,...
 x0,...
 v0,...
 time,...
 t_switch,...
 discy,...
 i_switch] = InitCond(tsteps,udot_disc_init0);
```

```
% fprintf('InitCond.m ok, loaded in %.01d s\n',toc)
% fprintf('--------------------------------------------------------------\n')
% fprintf('**************************************************************\n')
% fprintf('--------------------------------------------------------------\n')
%% Vibration Loop


% fdisc=1;
% fprintf('VIBRATION LOOP\n',toc)
for i=1:(size(t,2)-1)
    %%
    discy=sin(fdisc*t);
    %% Loop Header
%     fprintf('___________')
%     fprintf('Iteration: %d',i)
%     fprintf(' of: %d',tsteps)
%     fprintf('___________\n')
    %% Base Motion
        [v_pad(i),...
         Fnorm(i)] = BasMot(F_brake,...
                                    k_y,...
                                    v_pad(i),...
                                    discy(i),...
                                    fdisc,...
                                    xi_y,...
                                    w_y,...
                                    t(i));

%     fprintf('    BasMot.m ok\n')
```

```
    %% Stick Slip
    fcoul=(mu_k*Fnorm(i))/k_x;
    [u_disc(i+1),udot_disc(i+1),...
    u_pad(i+1),udot_pad(i+1),...
    StickOn(i+1),...
    time,t_switch,i_switch] = StkSlp(mu_k,...
                                    Fnorm,...
                                    k_x,...
                                    i,...
                                    u_pad,...
                                    udot_pad,...
                                    u_disc,...
                                    udot_disc,...
                                    m_pad,...
                                    m_disc,...
                                    x0,...
                                    v0,...
                                    tres,...
                                    w_x_stick,...
                                    wd_x_stick,...
                                    zi_x_stick,...
                                    zi_x,...
                                    w_x,...
                                    wd_x,...
                                    xi_x_stick,...
                                    StickOn,...
                                    time,...
                                    t_switch,...
                                    fcoul,...
                                    xi_x,...
                                    i_switch);
%     fprintf('    StkSlp.m ok\n')
    %% Bar Vibration
    [delta_3d,delta,...
            strain_axial,...
            strain_transverse,...
            deltay(i,:),...
            disc_y(i+1)] = BarVib(L,...
                                n_max,...
                                xsteps,...
                                xres,...
                                u_disc,...
                                i,...
                                time,...
                                E,...
                                rho,...
                                A,...
                                pois,...
                                T,...
                                j_xp);
%     fprintf('    Time since analysis start: %0.01d s \n',toc)
end
```

```
%% Plotting

PltPlt(t,...
         u_pad,...
         deltay,...
         ParameterStudied,...
         ParameterValue);
```

C.3. PltPlt.m

```
function [] = PltPlt(t,...
                     u_pad,...
                     deltay,...
                     ParameterStudied,...
                     ParameterValue)
%Written by Michael Vladimirov
%University of New Haven
%Department of Mechanical Engineering
%Master of Science, Mechanical Engineering
%2019

%This script generates the following plots:
%   1. Horizonatal displacment of the pad
%       a. For entire duration of simulation
%       b. For early part of simulation
%       c. For stable part of simulation
%   2. Horizonatal displacment of the disc
%       a. For entire duration of simulation
%       b. For early part of simulation
%       c. For stable part of simulation
%All plots are directly saved to the active directory path as .png files
%and, in order to reduce script execution time, are NOT displayed.
%% Axis Values
%start stop values for early behavior
starter1=round(0.05*size(t,2));
ender1=round(0.08*size(t,2));

%start stop values for stable behavior
starter2=round(0.5*size(t,2));
ender2=round(0.55*size(t,2));

%meshgrid generation for disc deformation
[xx,tt]=meshgrid(...
                 ((1:size(deltay,2))/size(deltay,2))*100,...
                 t(1:(size(t,2)-1))...
                );
```

```
%% Pad Plots

%Pad horizontal velocity
f_u_pad=figure('visible', 'off');
p1=plot(t,real(u_pad),'r');
p1.LineWidth=2;
title(...
      {...
       'u pad vs t',...
       [ParameterStudied,' ',num2str(ParameterValue),'% of cent. value']...
      }...
     );
xlabel('time (s)');
ylabel('u pad (m)');
filename=['u pad vs t_',...
          ParameterStudied,...
          '_',...
          num2str(ParameterValue),...
          '%'];
saveas(f_u_pad,filename,'png');
close(f_u_pad)

%Pad horizontal velocity (detail 1)
f_u_pad_det1=figure('visible', 'off');
p2=plot(t(starter1:ender1),real(u_pad(starter1:ender1)),'r');
p2.LineWidth=2;
title(...
      {...
       'u pad vs t (detail 1)',...
       [ParameterStudied,' ',num2str(ParameterValue),'% of cent. value']...
      }...
     );
xlabel('time (s)');
ylabel('u pad (m)');
filename=['u pad vs t (detail 1)_',...
          ParameterStudied,...
          '_',...
          num2str(ParameterValue),...
          '%'];
saveas(f_u_pad_det1,filename,'png');
close(f_u_pad_det1)
```

```
%Pad horizontal velocity (detail 2)
f_u_pad_det2=figure('visible', 'off');
p3=plot(t(starter2:ender2),real(u_pad(starter2:ender2)),'r');
p3.LineWidth=2;
title(...
      {...
       'u pad vs t (detail 2)',...
       [ParameterStudied,' ',num2str(ParameterValue),'% of cent. value']...
      }...
     );
xlabel('time (s)');
ylabel('u pad (m)');
filename=['u pad vs t (detail 2)_',...
          ParameterStudied,...
          '_',...
          num2str(ParameterValue),...
          '%'];
saveas(f_u_pad_det2,filename,'png');
close(f_u_pad_det2)
```

```
%% Disc Plot

%Disc vertical displacement
f_deltay=figure('visible', 'off');
contourf(...
         xx,...
         tt,...
         real(deltay),...
         'LineStyle','none'...
        );
c1=colorbar;
c1.Label.String = 'vertical deflection (m)';
colormap jet;
title(...
      {...
       'disc vertical displacement over time ',...
       [ParameterStudied,' ',num2str(ParameterValue),'% of cent. value']...
      }...
     );
xlabel('% from origin');
ylabel('time (s)');
filename=['disc vertical displacement over time_',...
          ParameterStudied,...
          '_',...
          num2str(ParameterValue),...
          '%'];
saveas(f_deltay,filename,'png');
close(f_deltay)
```

```
%Disc vertical displacement (detail 1)
f_deltay_det1=figure('visible', 'off');
contourf(...
         xx(starter1:ender1,:),...
         tt(starter1:ender1,:),...
         real(deltay(starter1:ender1,:)),...
         'LineStyle','none'...
        );
c2=colorbar;
c2.Label.String = 'vertical deflection (m)';
colormap jet;
title(...
      {...
       'disc vertical displacement over time (detail 1)',...
       [ParameterStudied,' ',num2str(ParameterValue),'% of cent. value']...
      }...
     );
xlabel('% from origin');
ylabel('time (s)');
filename=['disc vertical displacement over time (detail )_',...
          ParameterStudied,...
          '_',...
          num2str(ParameterValue),...
          '%'];
saveas(f_deltay_det1,filename,'png');
close(f_deltay_det1)


%Disc vertical displacement (detail 2)
f_deltay_det2=figure('visible', 'off');
contourf(...
         xx(starter2:ender2,:),...
         tt(starter2:ender2,:),...
         real(deltay(starter2:ender2,:)),...
         'LineStyle','none'...
        );
colorbar;
c3=colorbar;
c3.Label.String = 'vertical deflection (m)';
colormap jet;
title(...
      {...
       'disc vertical displacement over time (detail 2)',...
       [ParameterStudied,' ',num2str(ParameterValue),'% of cent. value']...
      }...
     );
xlabel('% from origin');
ylabel('time (s)');
filename=['disc vertical displacement over time (detail 2)_',...
          ParameterStudied,...
          '_',...
          num2str(ParameterValue),...
          '%'];
saveas(f_deltay_det2,filename,'png');
close(f_deltay_det2)
end
```

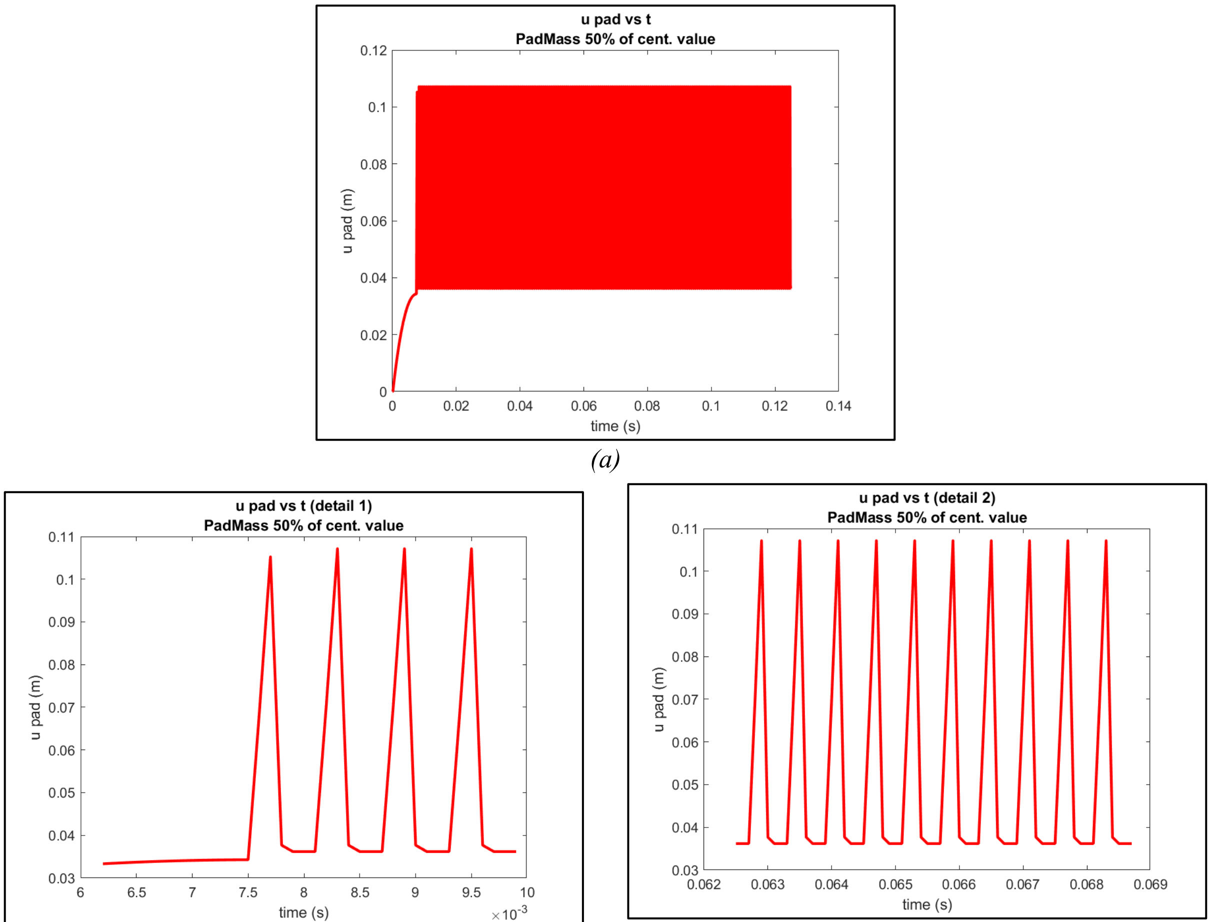
Appendix D. Parameter Variation Study Results

This appendix contains the full results from the parameter variation study discussed in section 3 Parameter Variation Study (page 39).

[The remainder of this page is left intentionally blank.]

D.1. Mass Variation

D.1.1. Pad Horizontal Displacement



(a)

(b) *(c)*

Figure 16 – Pad horizontal displacement with 50% pad mass, for full simulation duration (a), in early state of simulation (b), and during stable part of simulation (c)

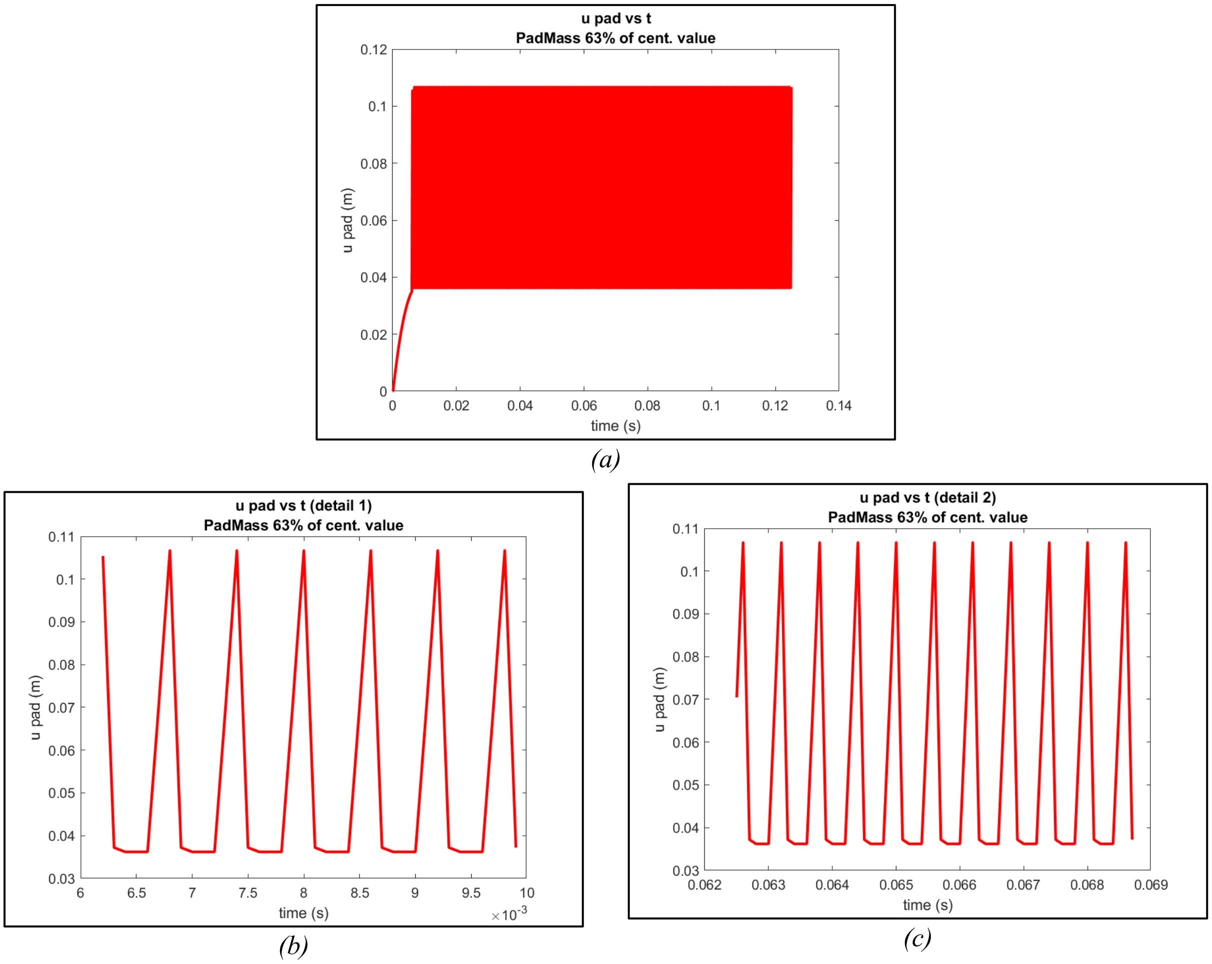
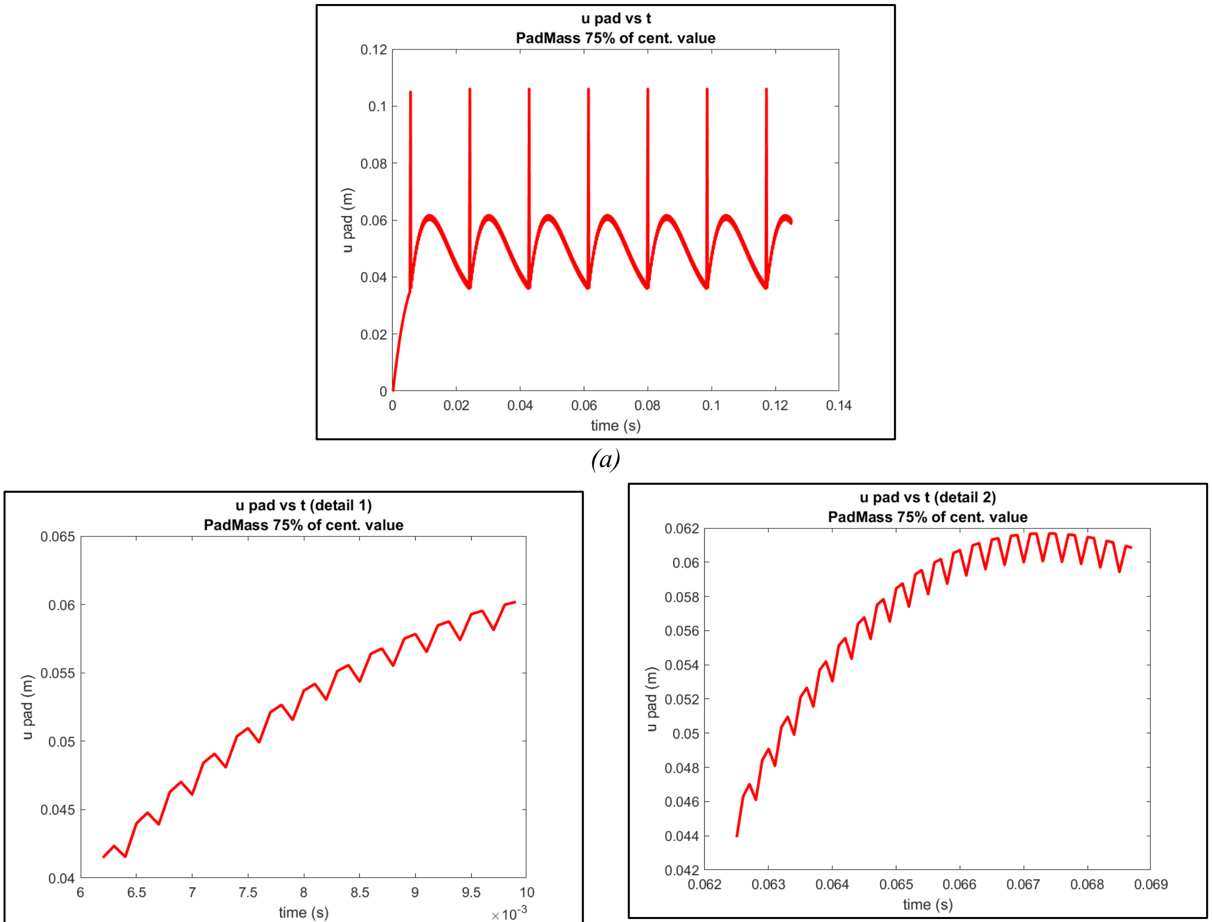


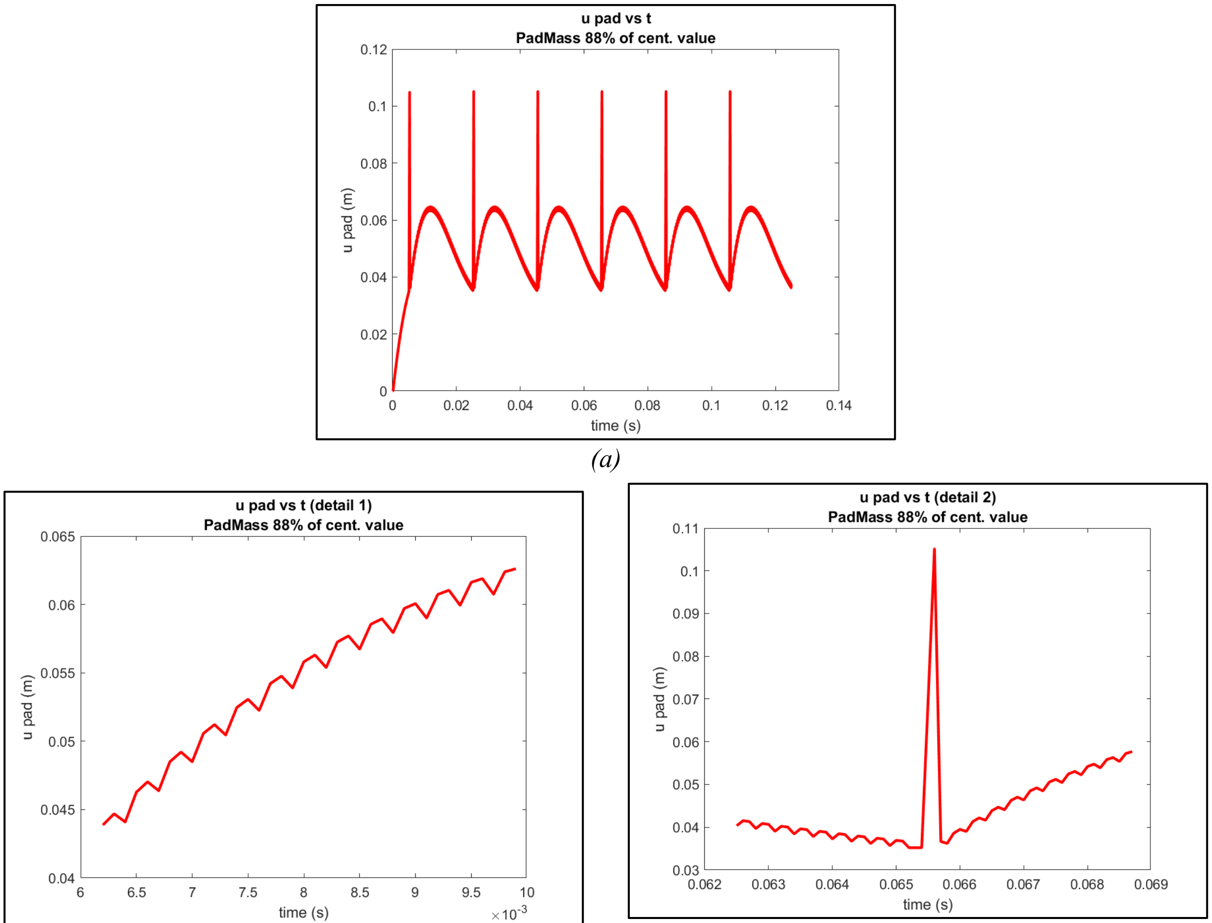
Figure 17 – Pad horizontal displacement with 63% pad mass, for full simulation duration (a), in early state of simulation (b), and during stable part of simulation (c)



(a)

(b) *(c)*

Figure 18 – Pad horizontal displacement with 75% pad mass, for full simulation duration (a), in early state of simulation (b), and during stable part of simulation (c)

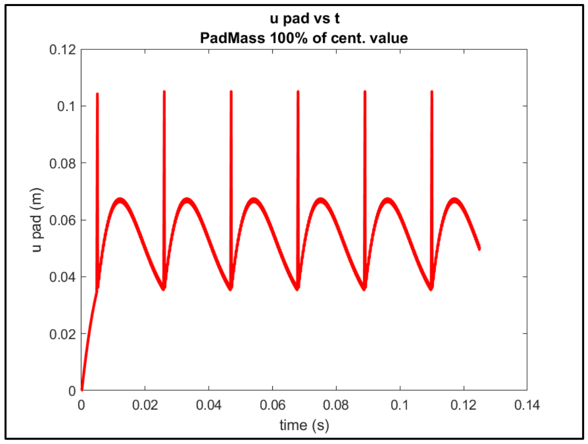


(a)

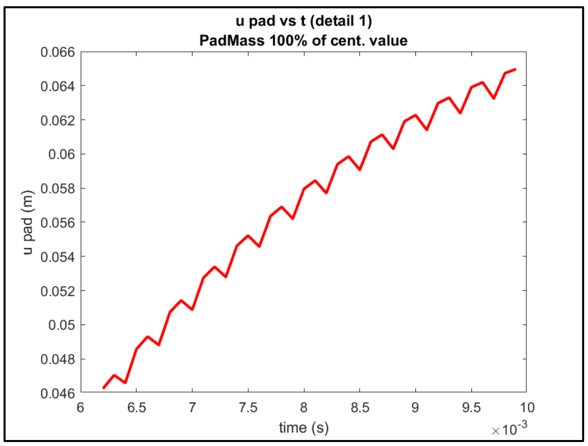
(b)

(c)

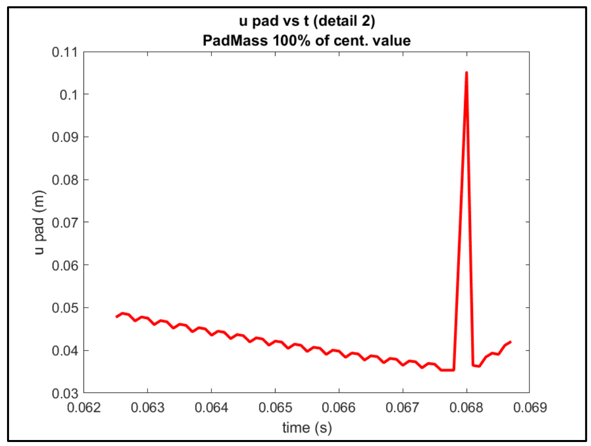
Figure 19 – Pad horizontal displacement with 88% pad mass, for full simulation duration (a), in early state of simulation (b), and during stable part of simulation (c)



(a)

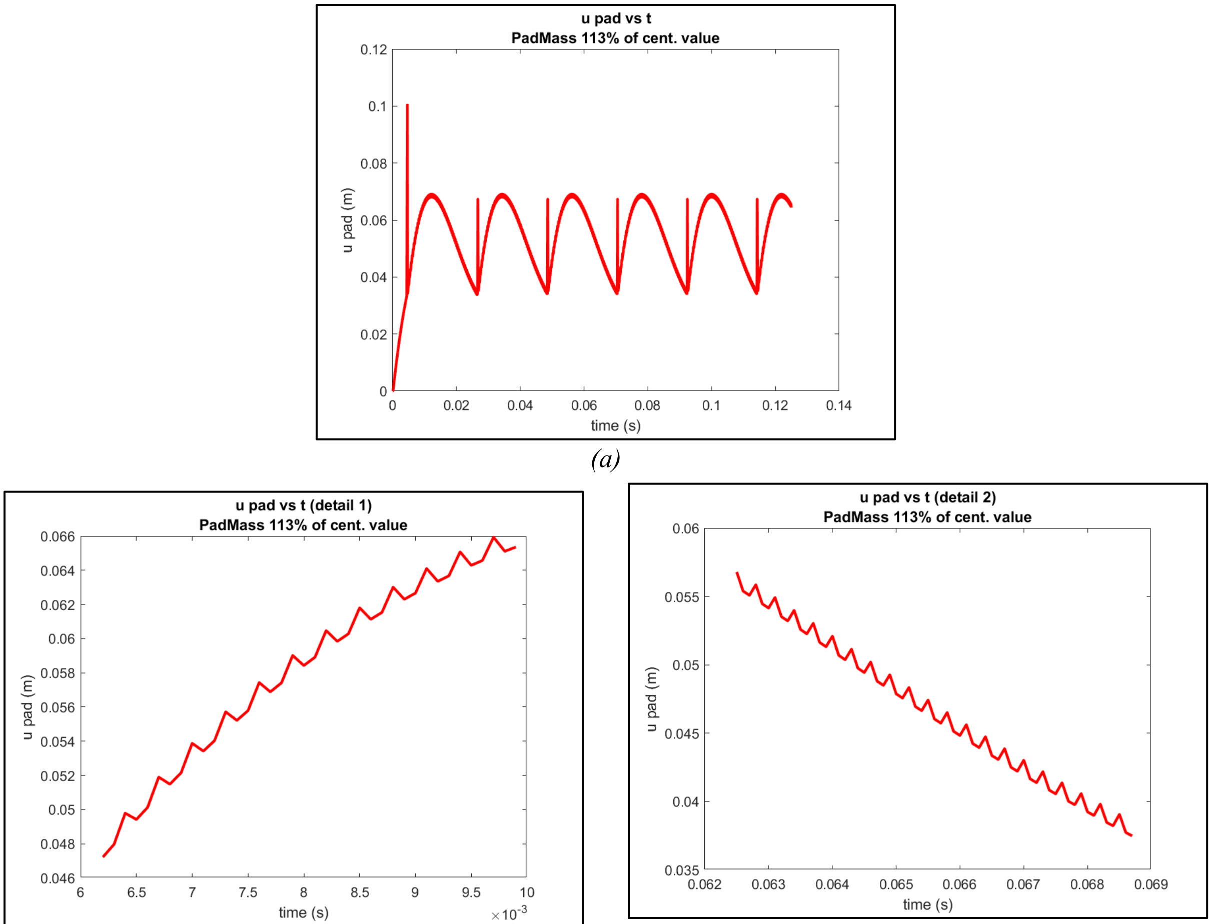


(b)



(c)

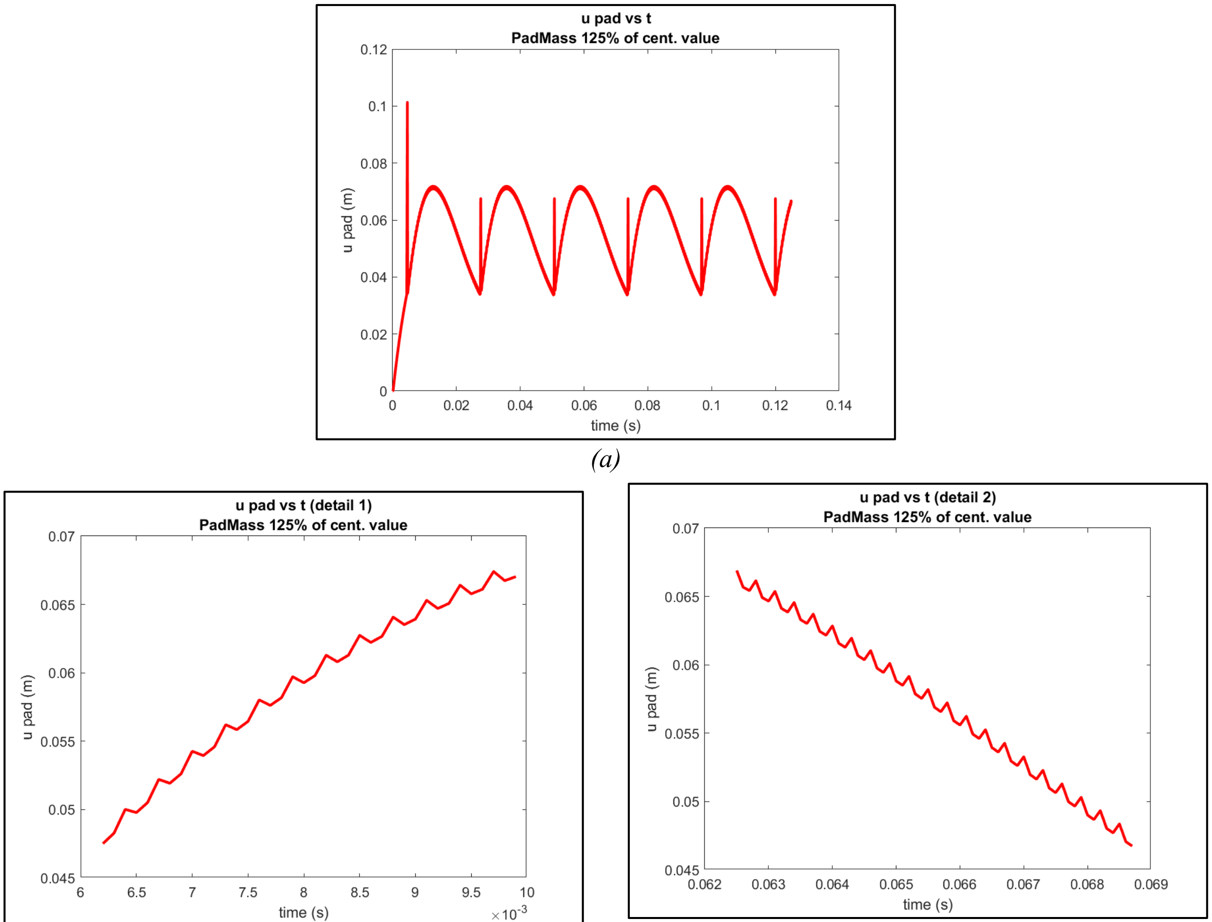
Figure 20 – Pad horizontal displacement with 100% pad mass, for full simulation duration (a), in early state of simulation (b), and during stable part of simulation (c)



(a)

(b) *(c)*

Figure 21 – Pad horizontal displacement with 113% pad mass, for full simulation duration (a), in early state of simulation (b), and during stable part of simulation (c)

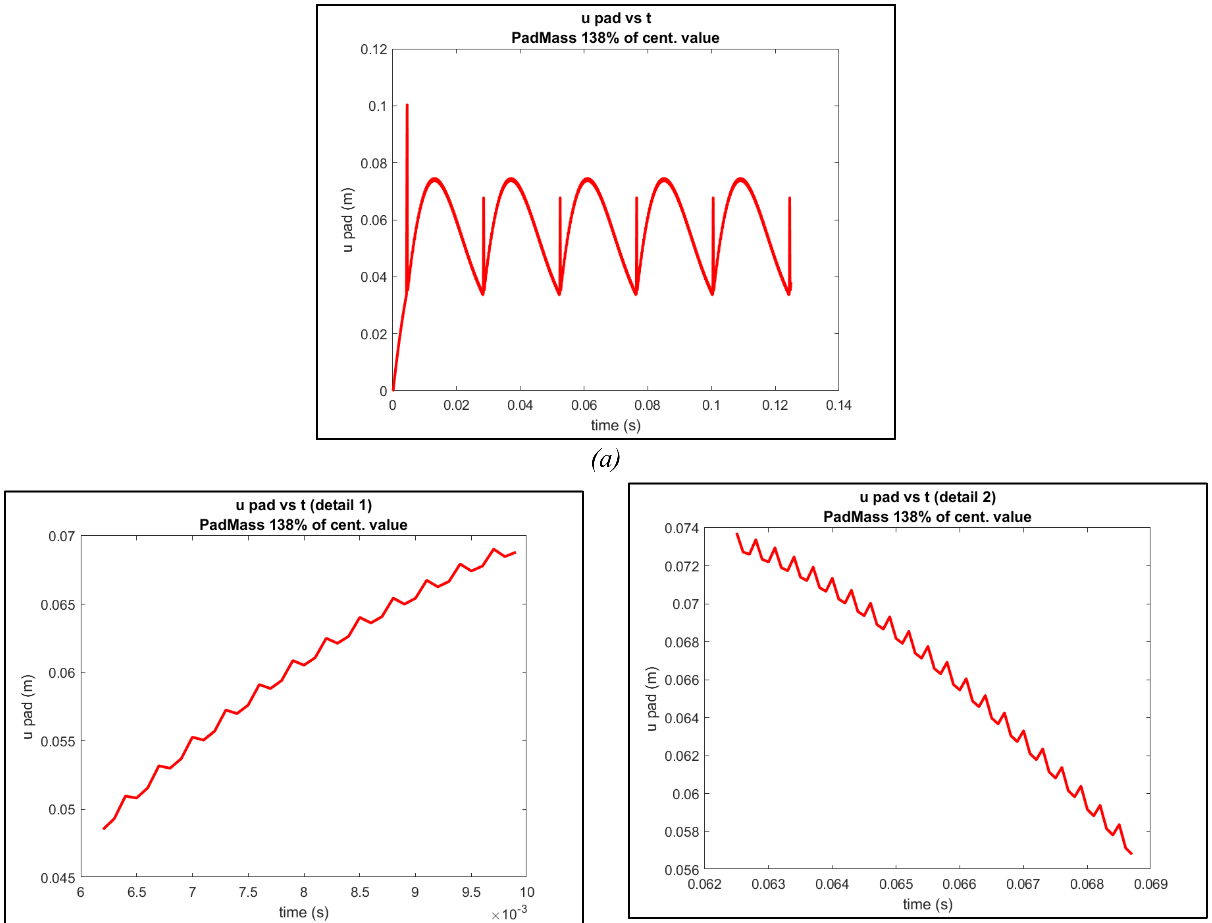


(a)

(b)

(c)

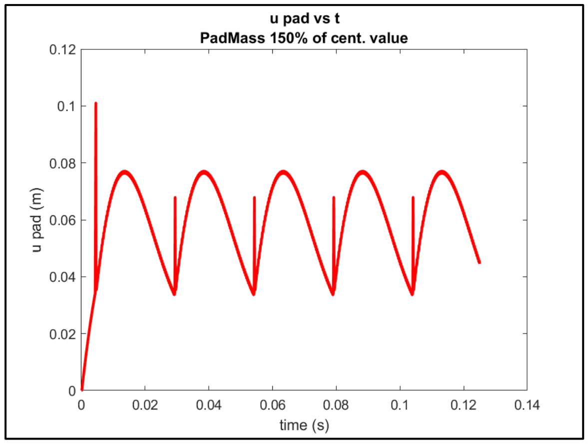
Figure 22 – Pad horizontal displacement with 125% pad mass, for full simulation duration (a), in early state of simulation (b), and during stable part of simulation (c)



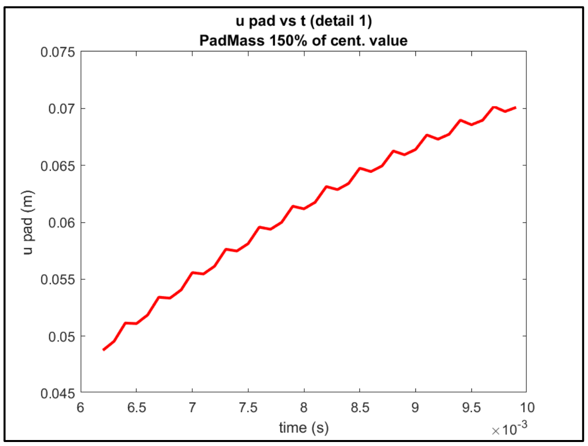
(a)

(b) *(c)*

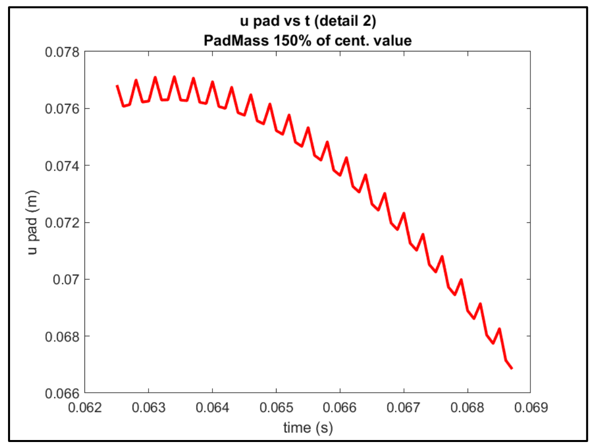
Figure 23 – Pad horizontal displacement with 138% pad mass, for full simulation duration (a), in early state of simulation (b), and during stable part of simulation (c)



(a)



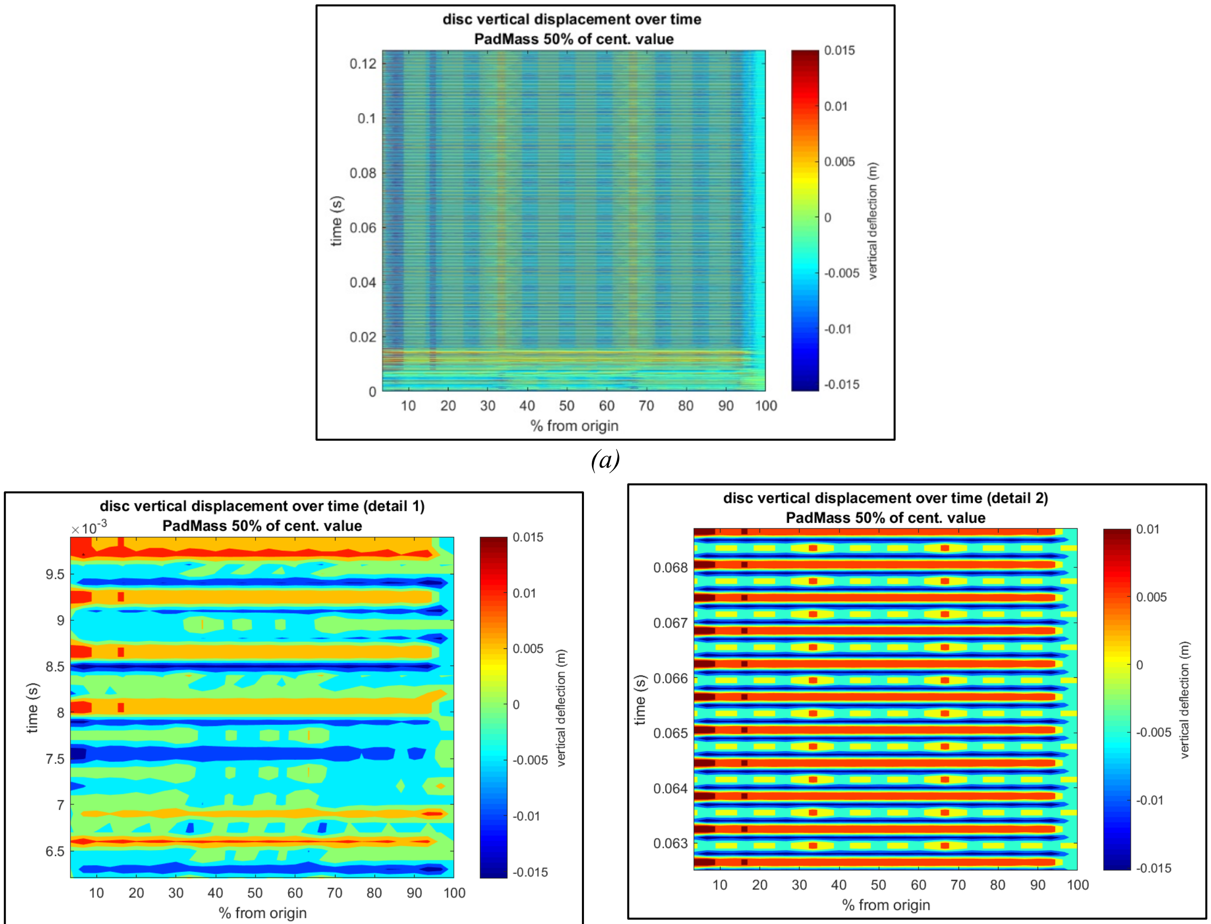
(b)



(c)

Figure 24 – Pad horizontal displacement with 150% pad mass, for full simulation duration (a), in early state of simulation (b), and during stable part of simulation (c)

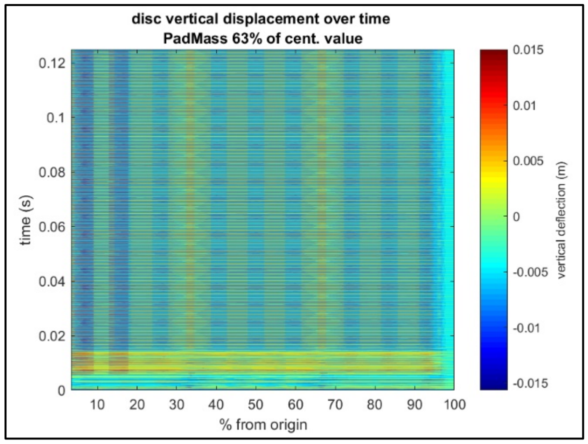
D.1.2. Disc Surface Vertical Displacement



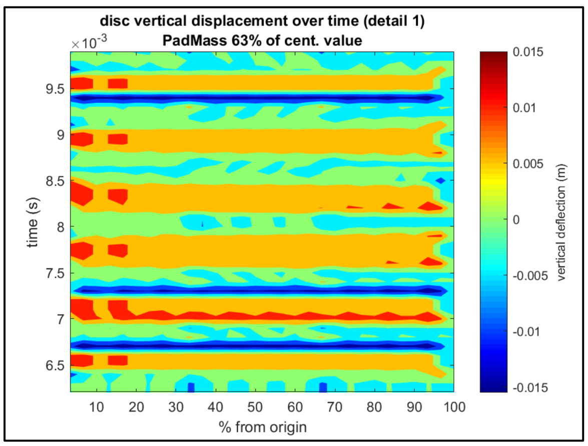
(a)

(b) *(c)*

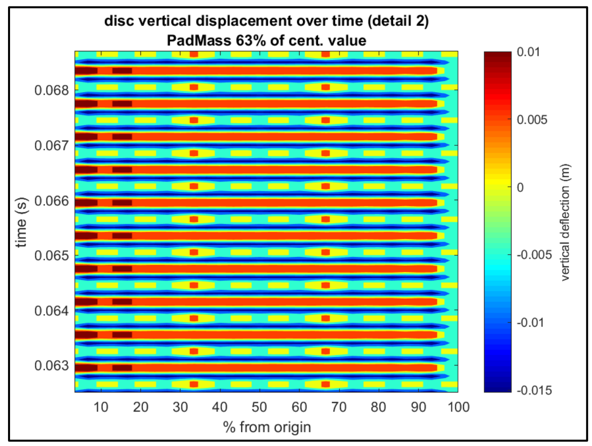
Figure 25 – Disc surface vertical displacement with 50% pad mass, for full simulation duration (a), in early state of simulation (b), and during stable part of simulation (c)



(a)



(b)



(c)

Figure 26 – Disc surface vertical displacement with 63% pad mass, for full simulation duration (a), in early state of simulation (b), and during stable part of simulation (c)

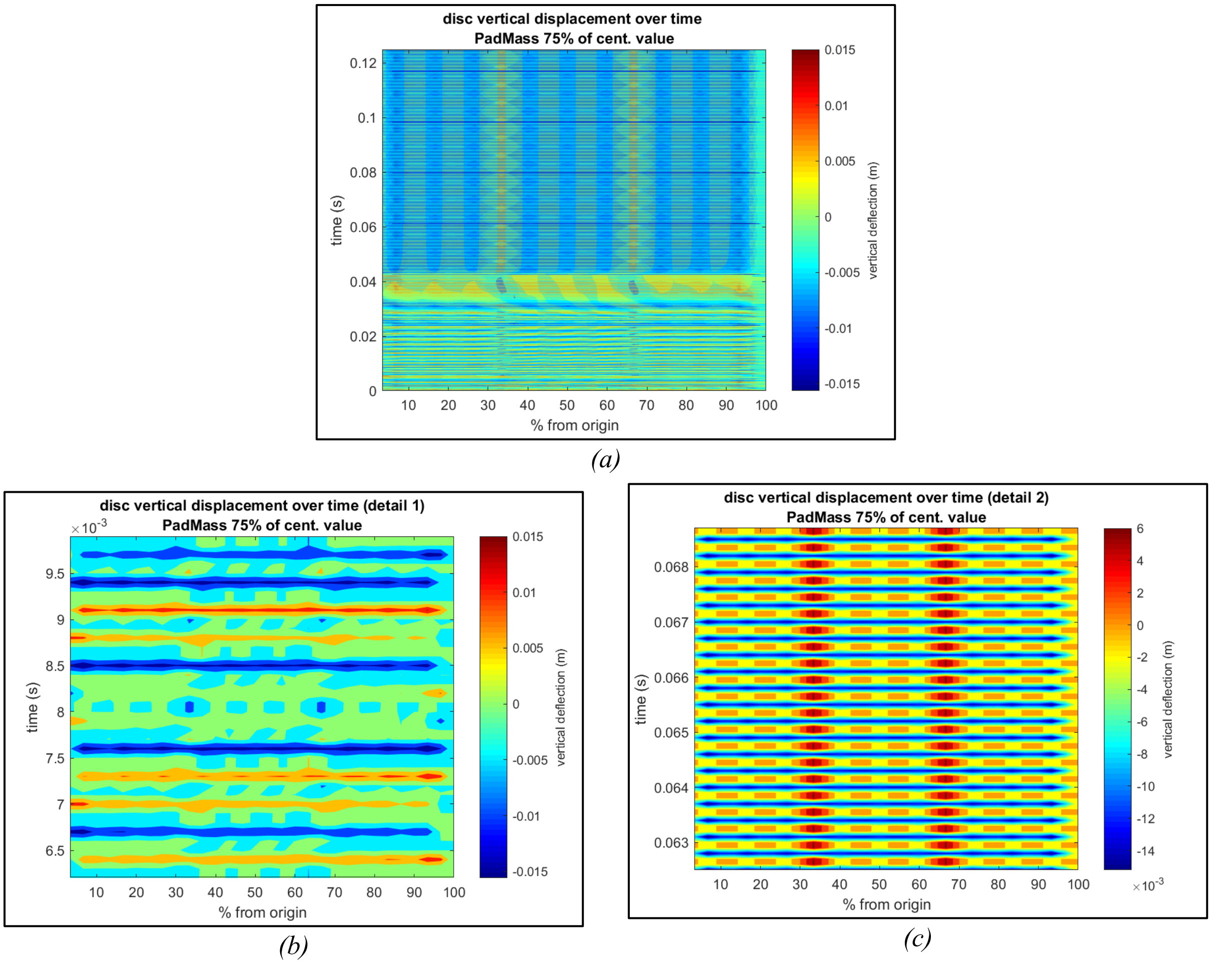


Figure 27 – Disc surface vertical displacement with 75% pad mass, for full simulation duration (a), in early state of simulation (b), and during stable part of simulation (c)

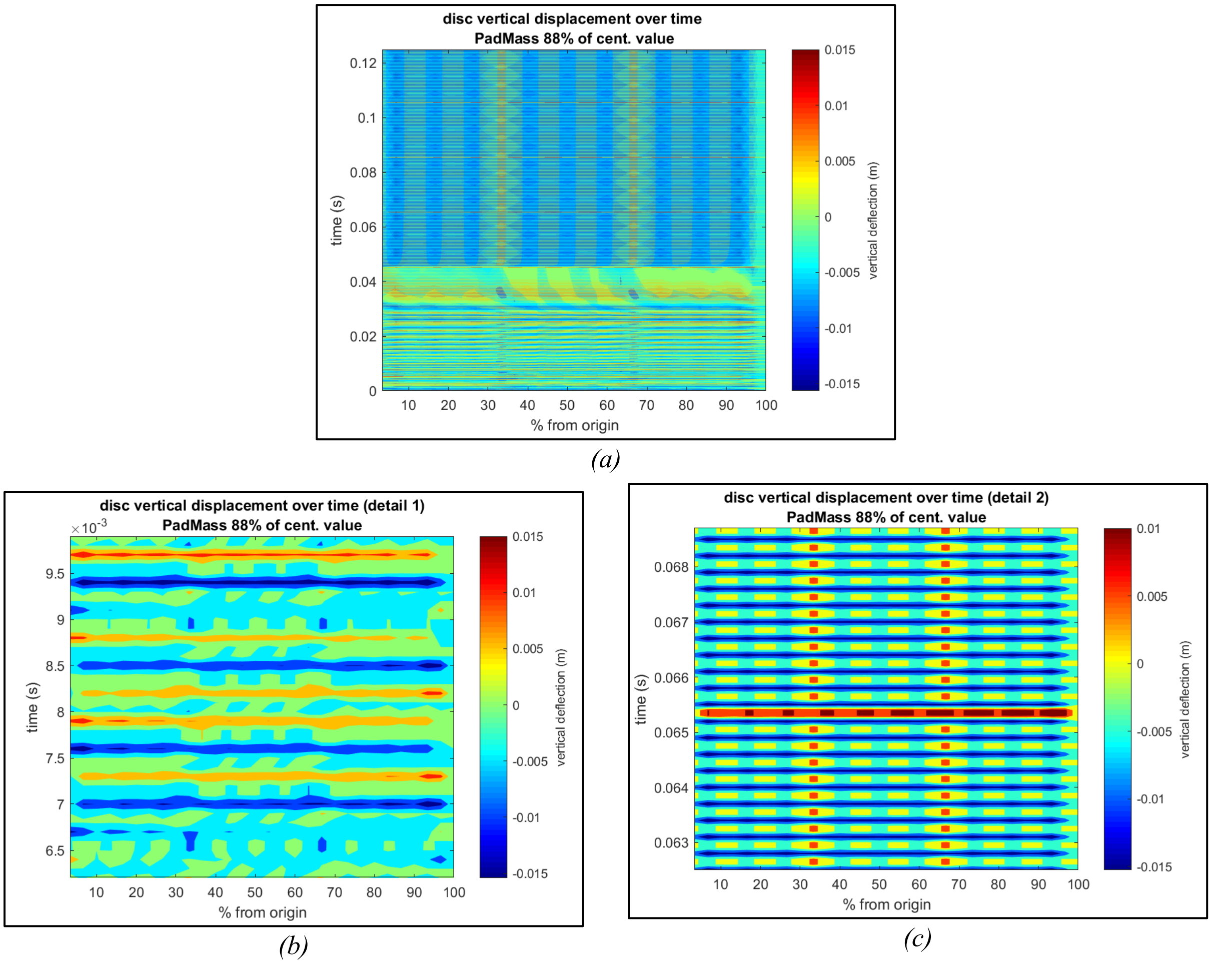
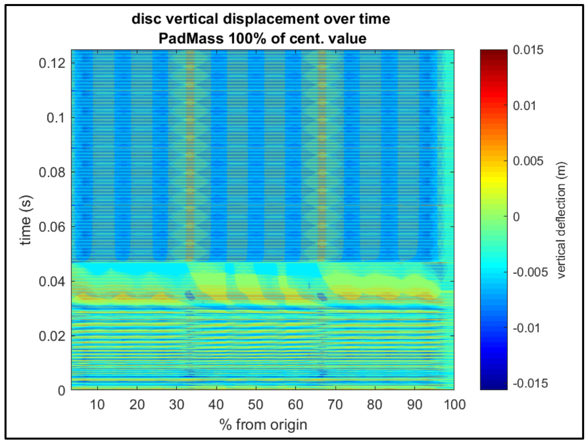
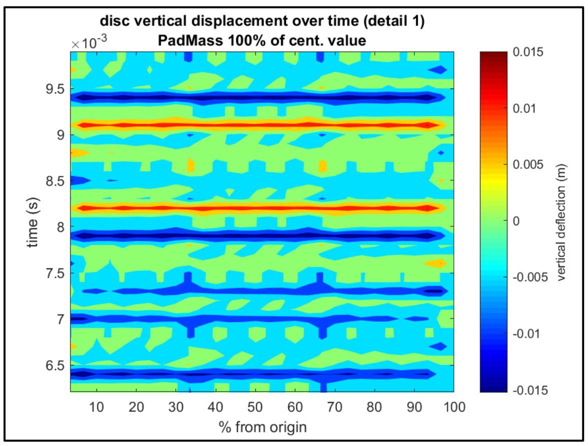


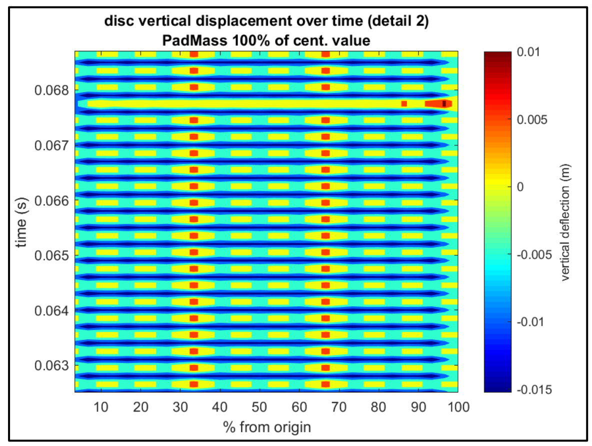
Figure 28 – Disc surface vertical displacement with 88% pad mass, for full simulation duration (a), in early state of simulation (b), and during stable part of simulation (c)



(a)



(b)



(c)

Figure 29 – Disc surface vertical displacement with 100% pad mass, for full simulation duration (a), in early state of simulation (b), and during stable part of simulation (c)

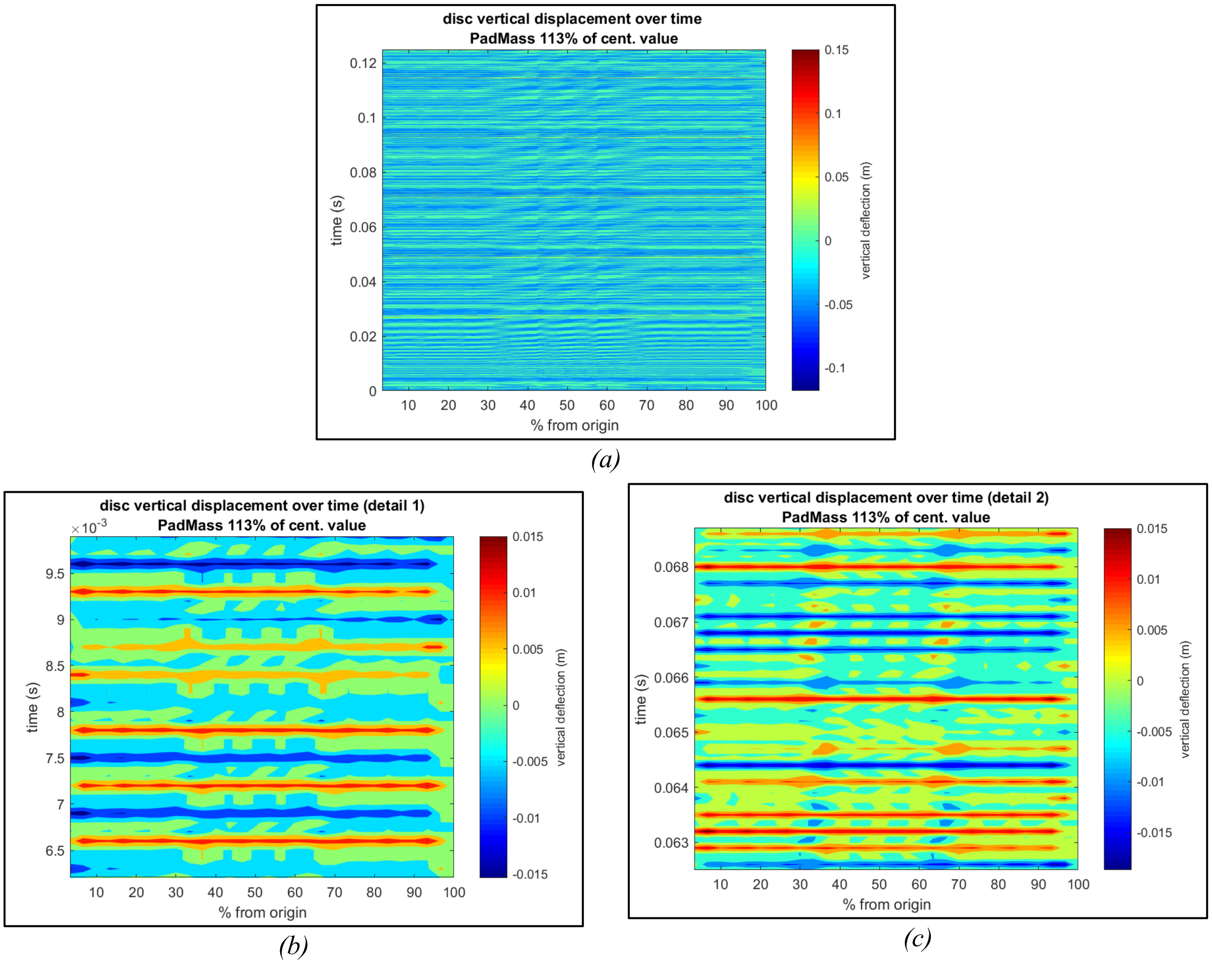


Figure 30 – Disc surface vertical displacement with 113% pad mass, for full simulation duration (a), in early state of simulation (b), and during stable part of simulation (c)

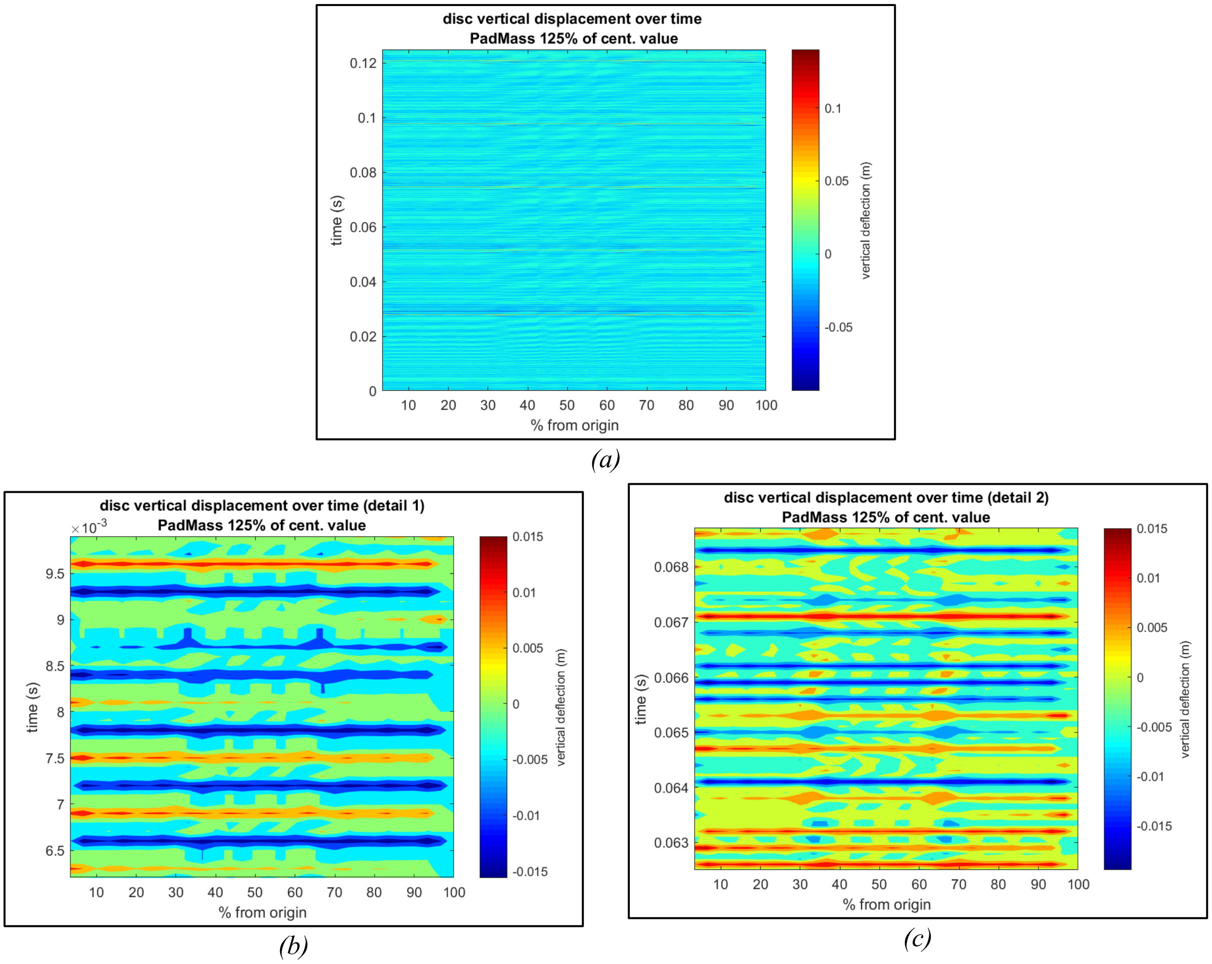


Figure 31 – Disc surface vertical displacement with 125% pad mass, for full simulation duration (a), in early state of simulation (b), and during stable part of simulation (c)

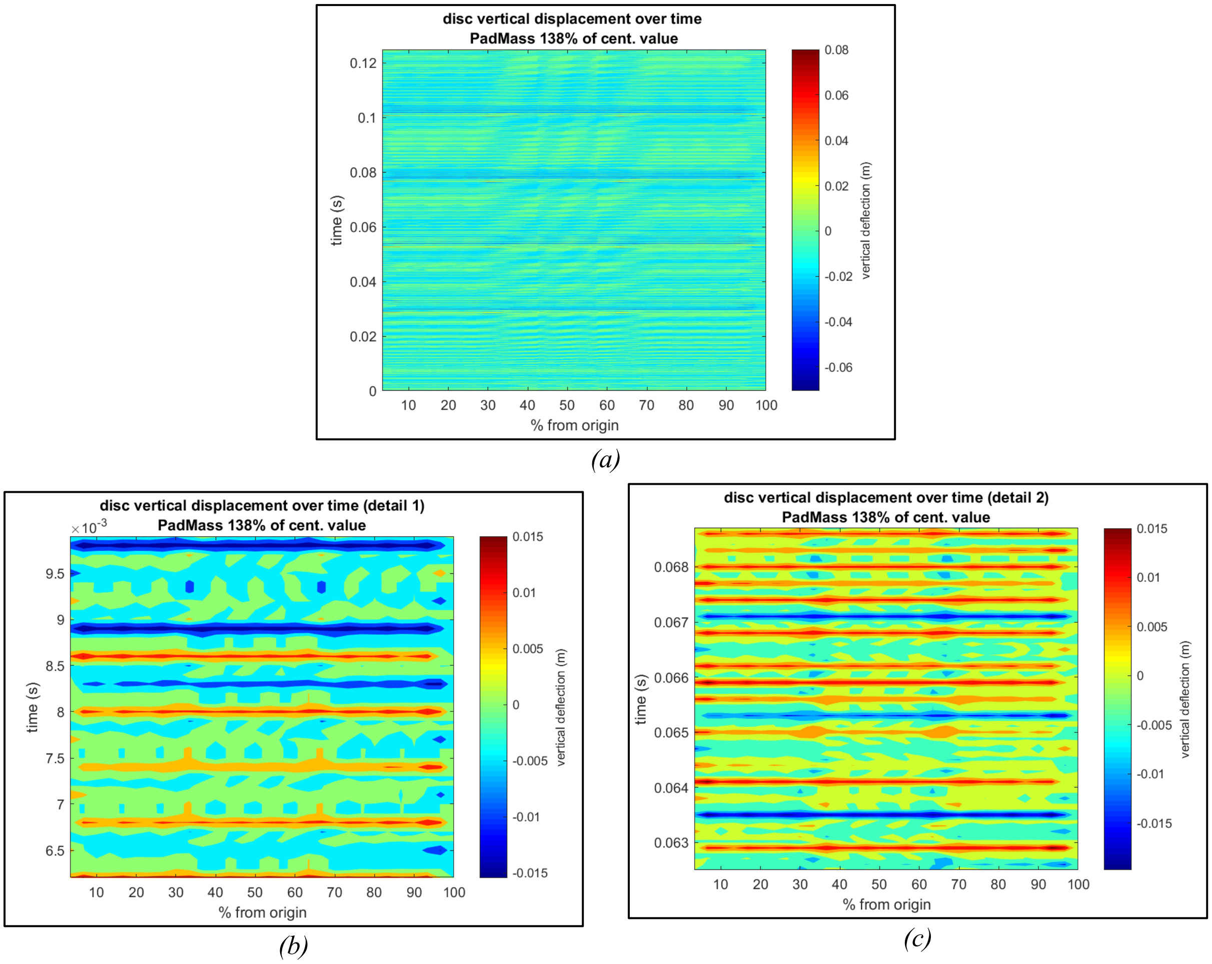
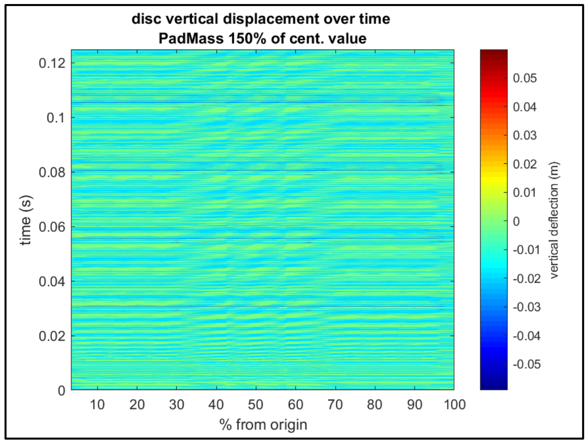
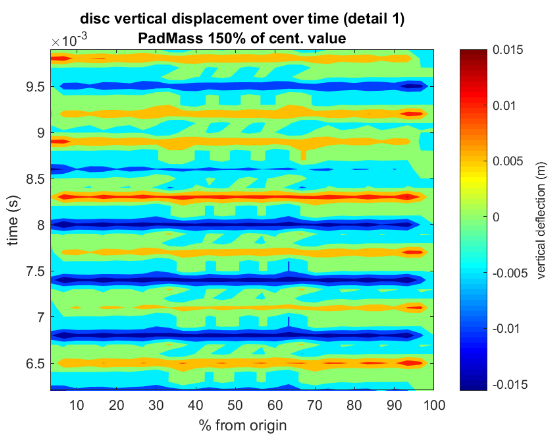


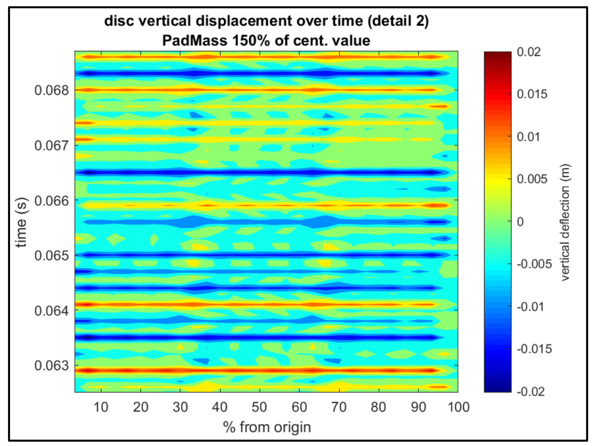
Figure 32 – Disc surface vertical displacement with 138% pad mass, for full simulation duration (a), in early state of simulation (b), and during stable part of simulation (c)



(a)



(b)

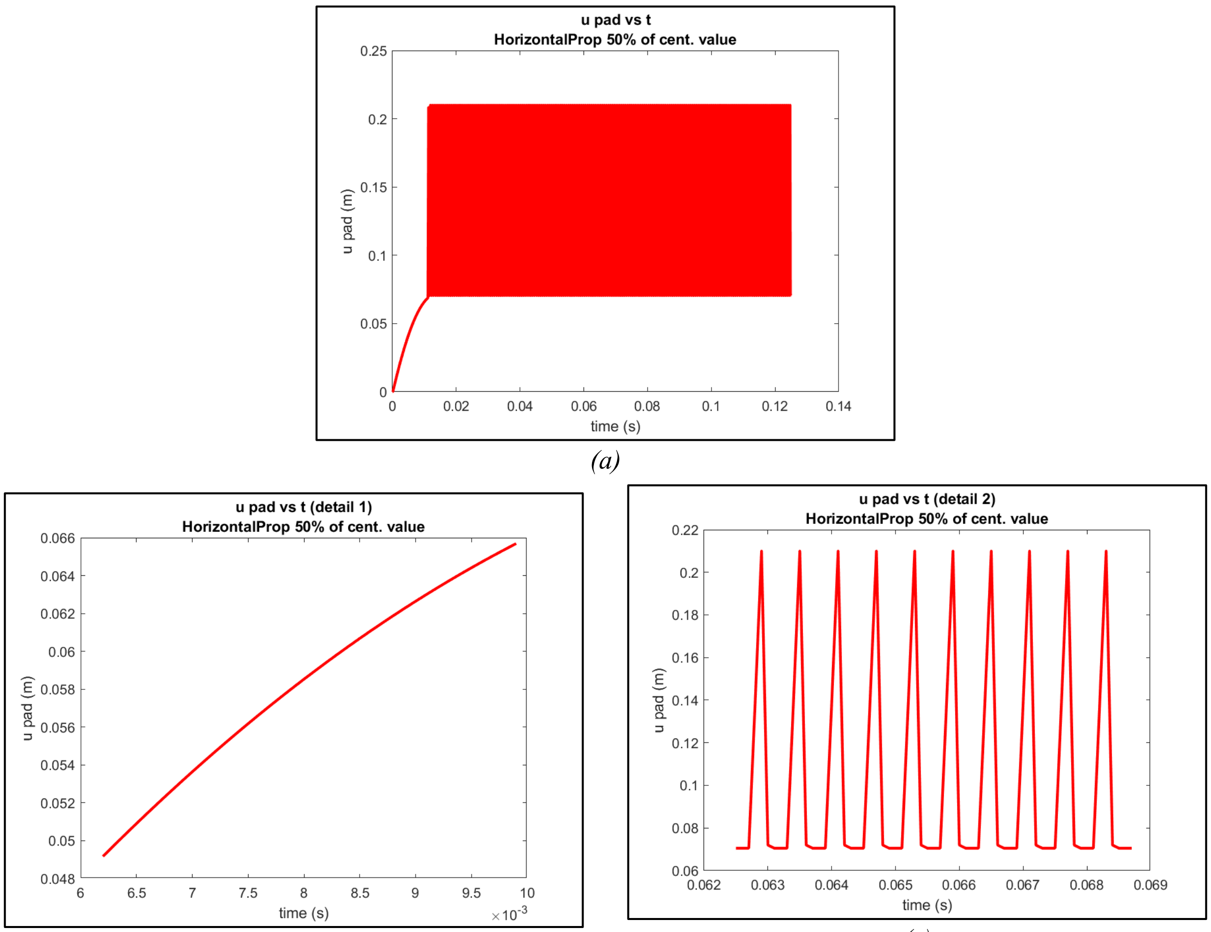


(c)

Figure 33 – Disc surface vertical displacement with 150% pad mass, for full simulation duration (a), in early state of simulation (b), and during stable part of simulation (c)

D.2. Horizontal Stiffness and Damping Ratio Variation

D.2.1. Pad Horizontal Displacement



(a)

(b) *(c)*

Figure 34 – Pad horizontal displacement with 50% horizontal stiffness and damping, for full simulation duration (a), in early state of simulation (b), and during stable part of simulation (c)

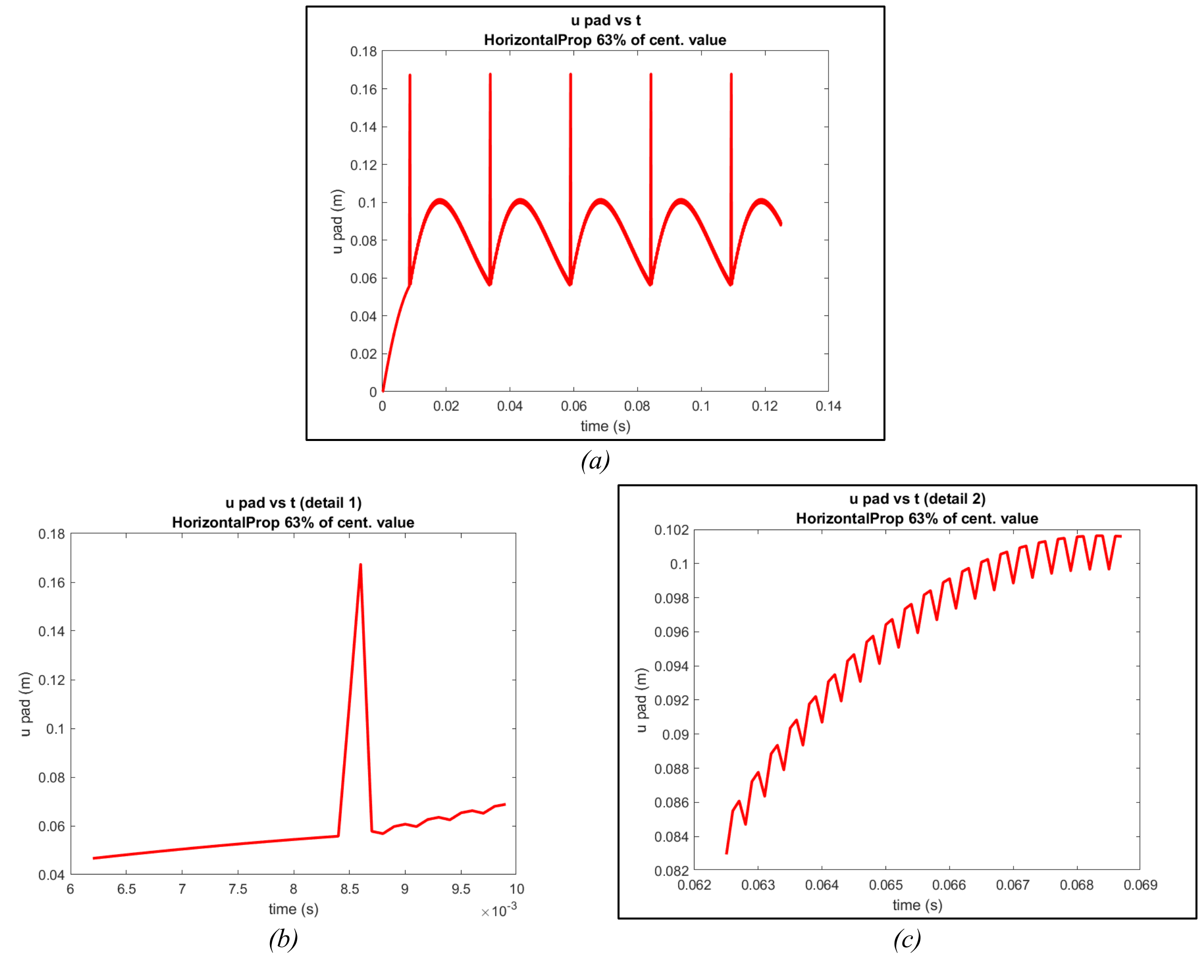
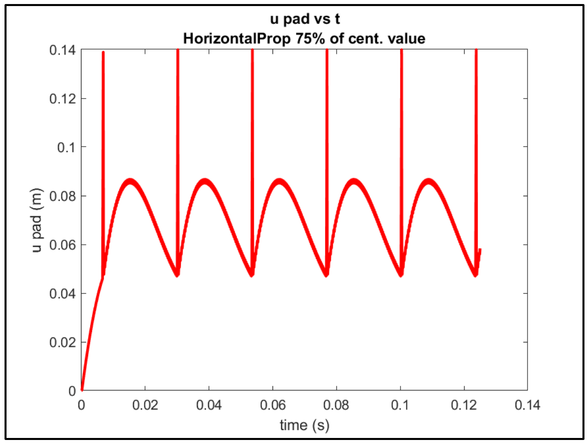
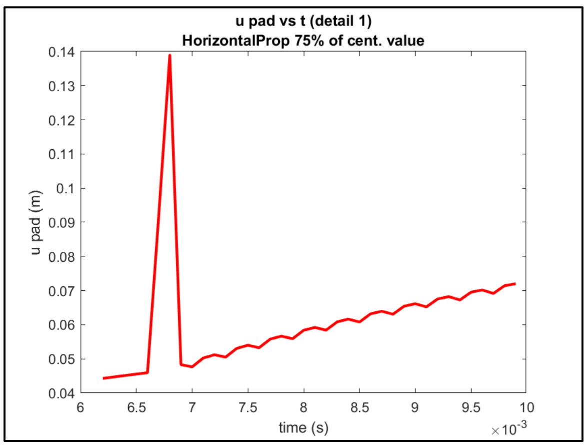


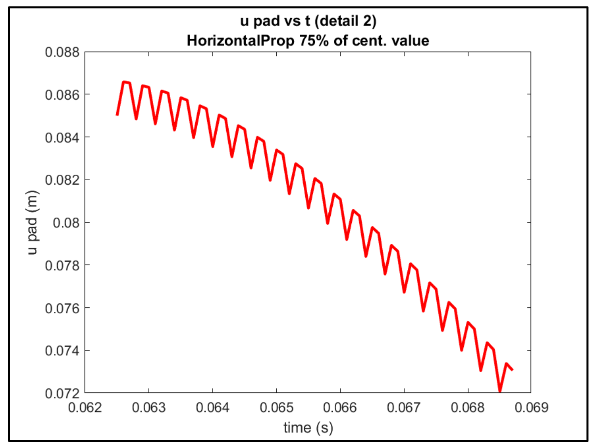
Figure 35 – Pad horizontal displacement with 63% horizontal stiffness and damping, for full simulation duration (a), in early state of simulation (b), and during stable part of simulation (c)



(a)

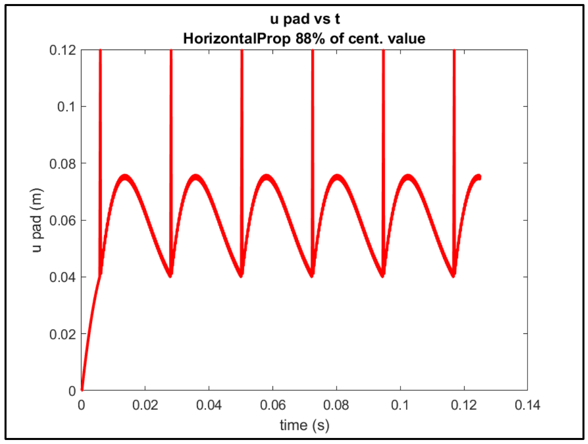


(b)

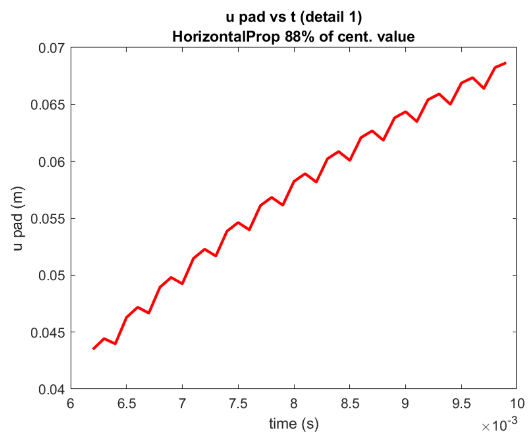


(c)

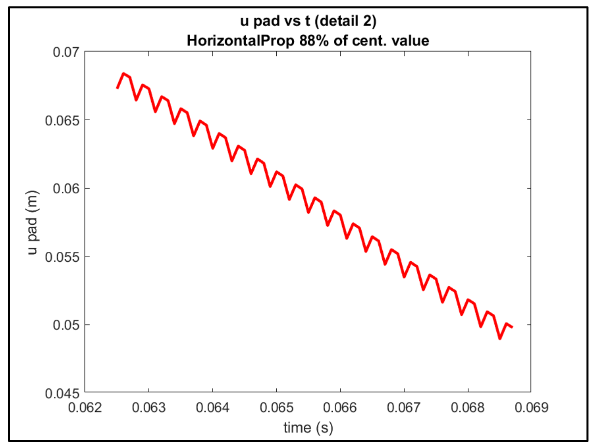
Figure 36 – Pad horizontal displacement with 75% horizontal stiffness and damping, for full simulation duration (a), in early state of simulation (b), and during stable part of simulation (c)



(a)



(b)



(c)

Figure 37 – Pad horizontal displacement with 88% horizontal stiffness and damping, for full simulation duration (a), in early state of simulation (b), and during stable part of simulation (c)

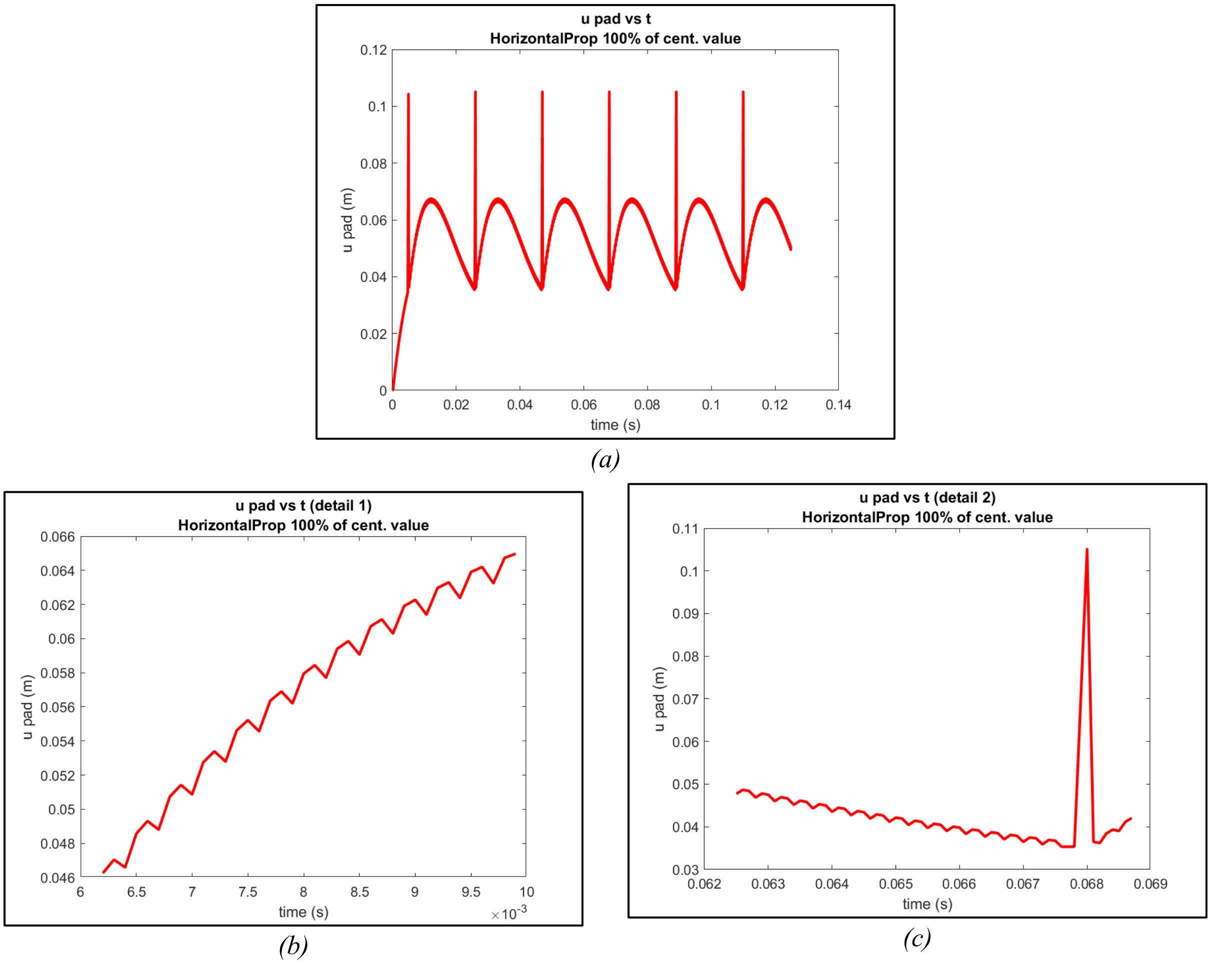
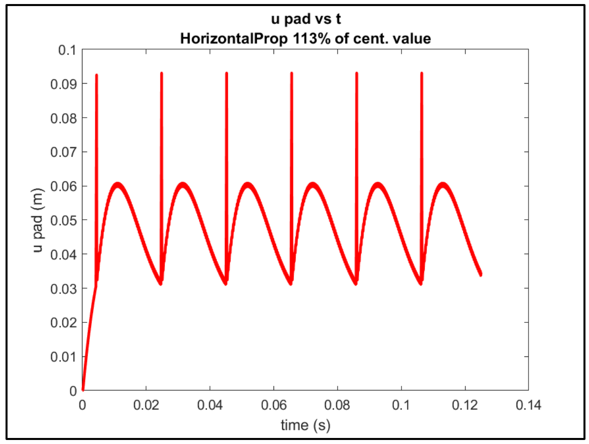
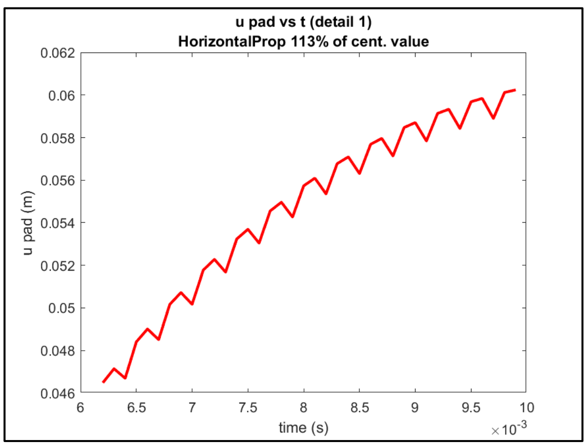


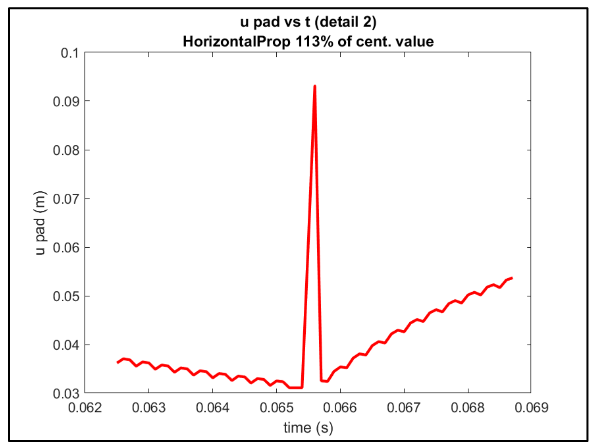
Figure 38 – Pad horizontal displacement with 100% horizontal stiffness and damping, for full simulation duration (a), in early state of simulation (b), and during stable part of simulation (c)



(a)

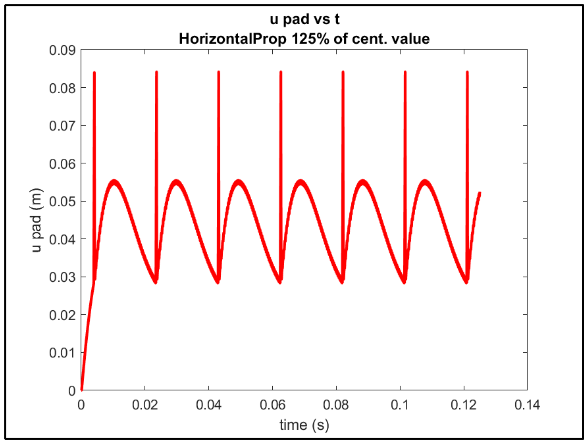


(b)

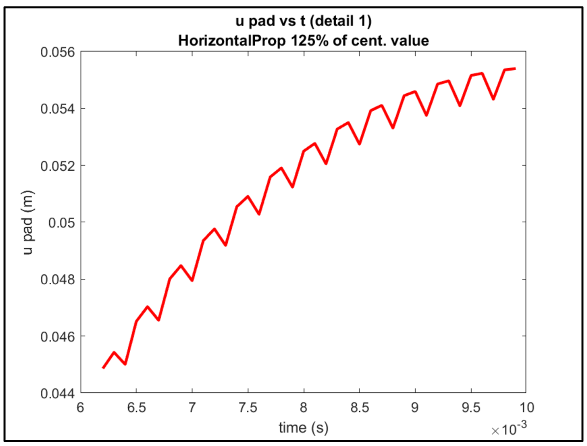


(c)

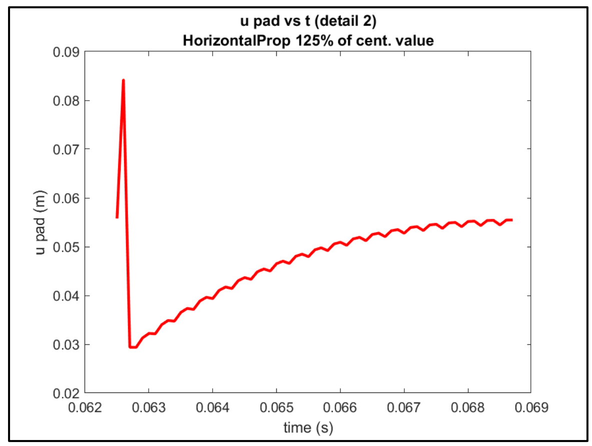
Figure 39 – Pad horizontal displacement with 113% horizontal stiffness and damping, for full simulation duration (a), in early state of simulation (b), and during stable part of simulation (c)



(a)

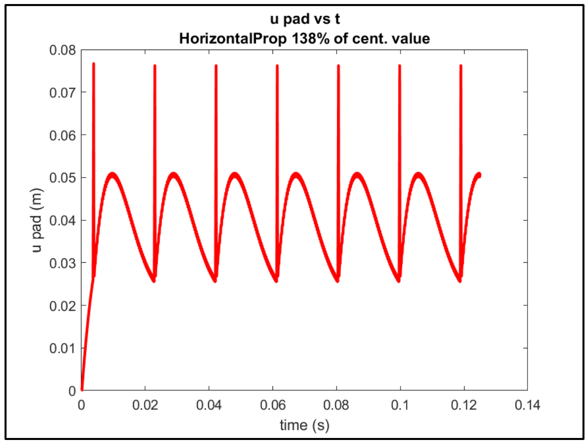


(b)

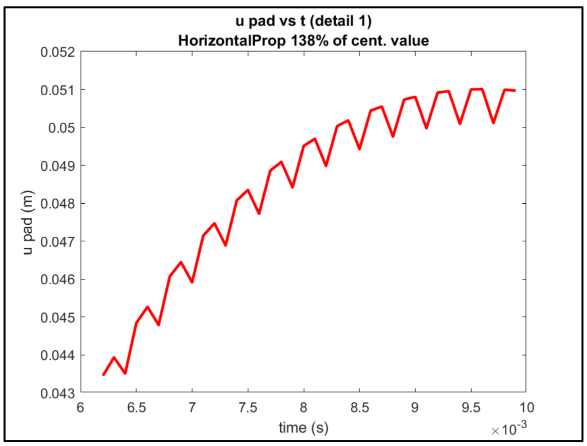


(c)

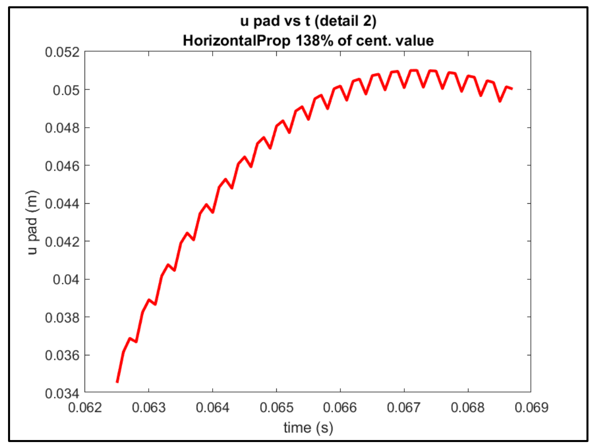
Figure 40 – Pad horizontal displacement with 125% horizontal stiffness and damping, for full simulation duration (a), in early state of simulation (b), and during stable part of simulation (c)



(a)

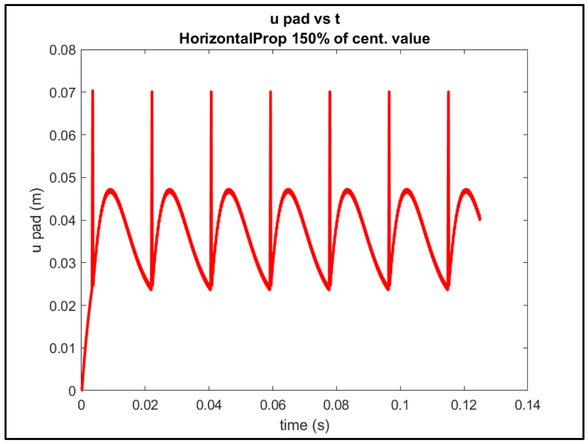


(b)

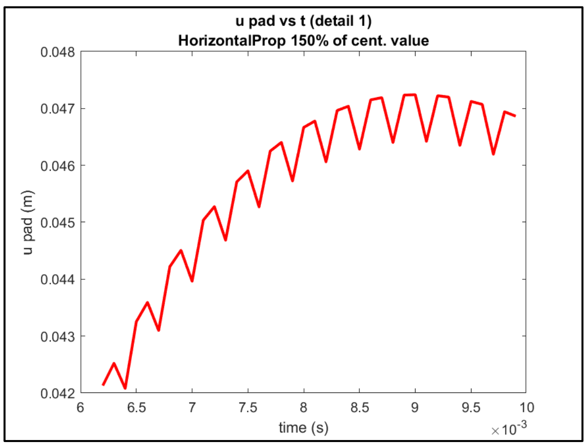


(c)

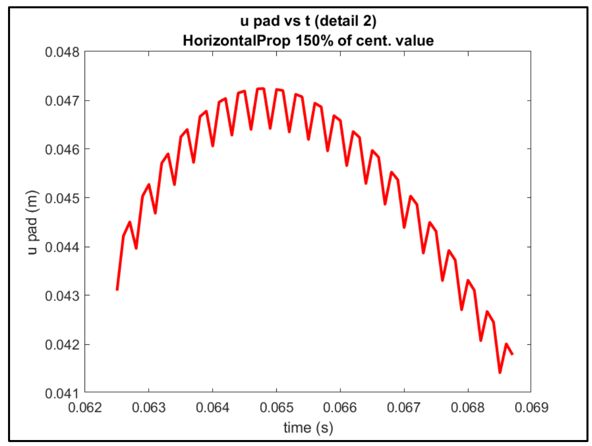
Figure 41 – Pad horizontal displacement with 138% horizontal stiffness and damping, for full simulation duration (a), in early state of simulation (b), and during stable part of simulation (c)



(a)



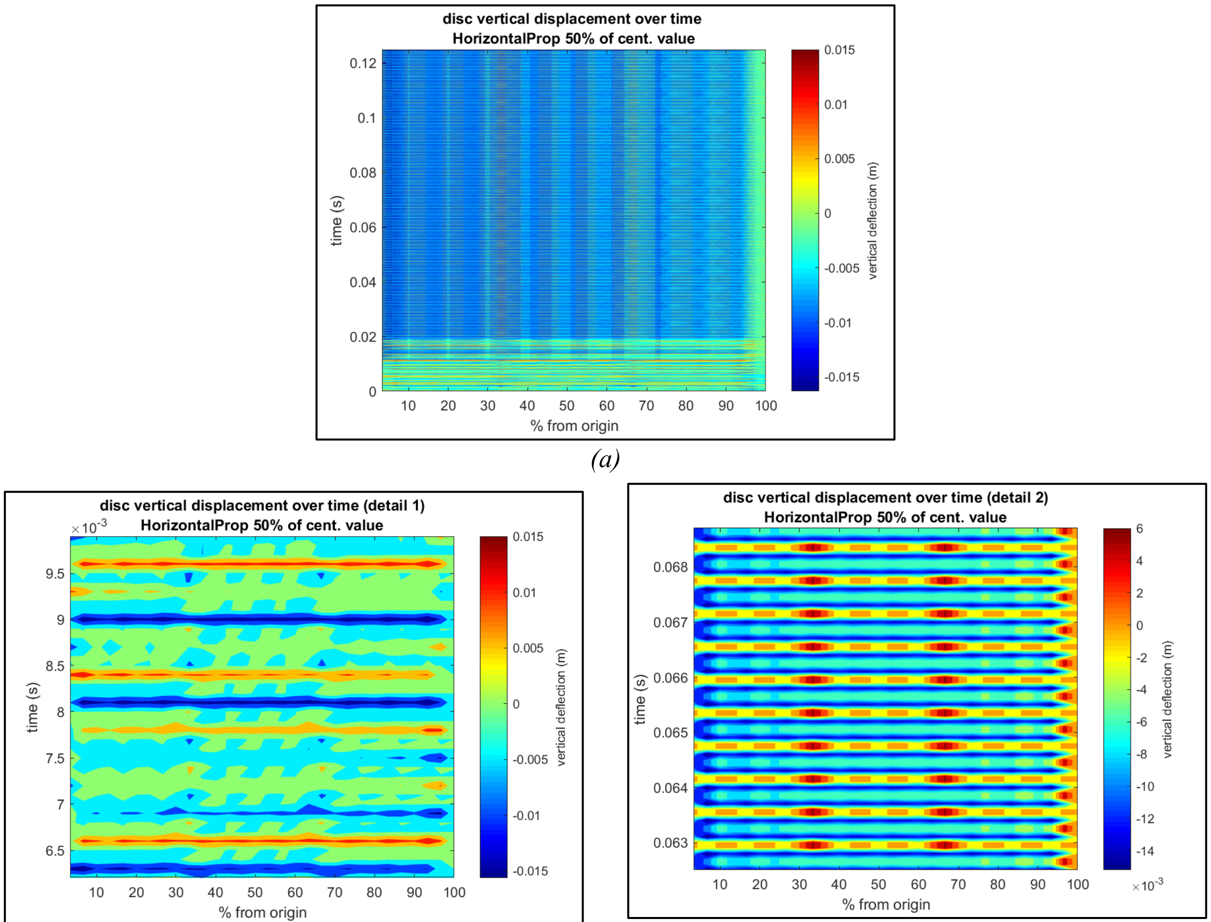
(b)



(c)

Figure 42 – Pad horizontal displacement with 150% horizontal stiffness and damping, for full simulation duration (a), in early state of simulation (b), and during stable part of simulation (c)

D.2.2. Disc Surface Vertical Displacement



(a)

(b) *(c)*

Figure 43 – Disc 50% horizontal stiffness and damping, for full simulation duration (a), in early state of simulation (b), and during stable part of simulation (c)

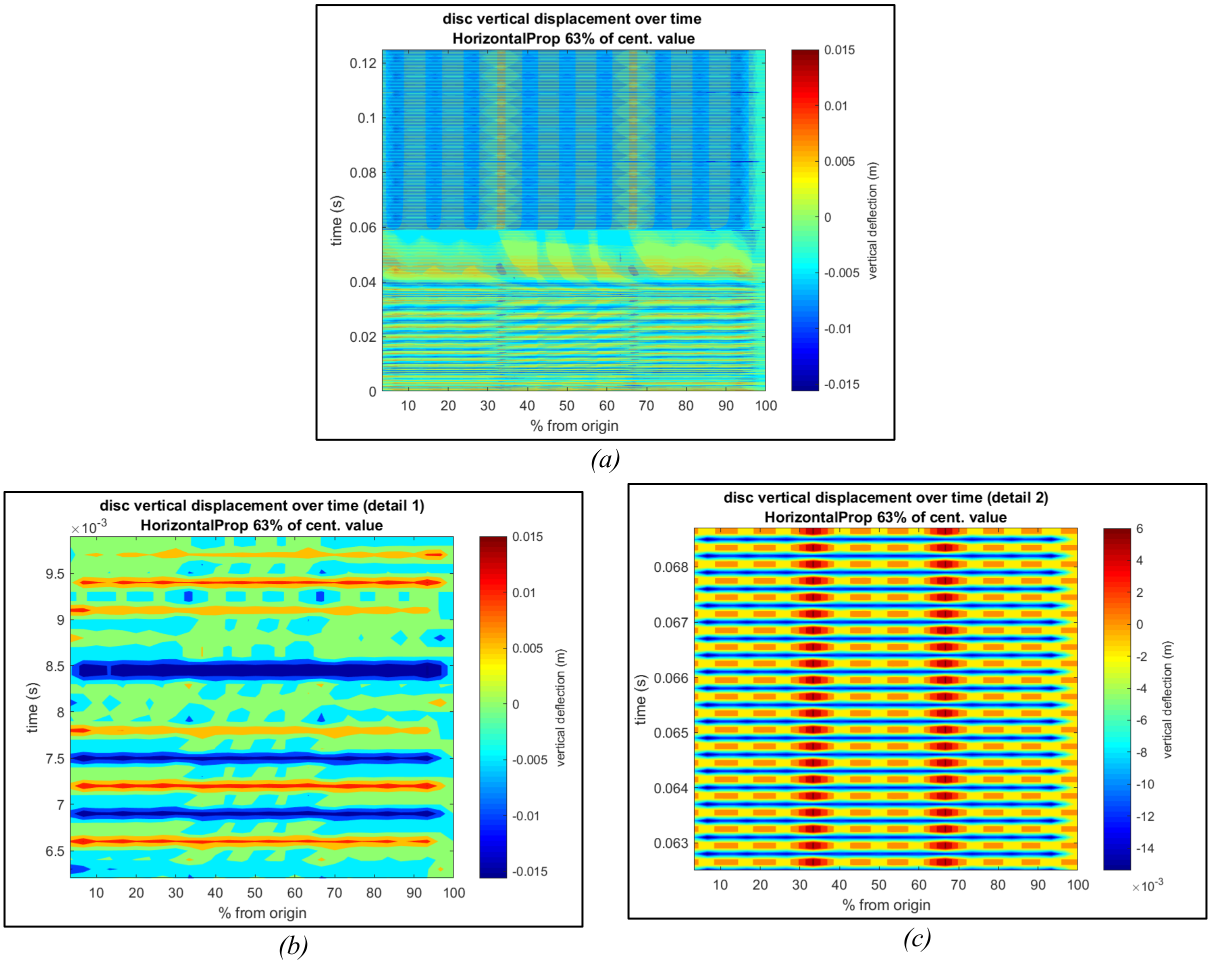


Figure 44 – Disc surface vertical displacement with 63% horizontal stiffness and damping, for full simulation duration (a), in early state of simulation (b), and during stable part of simulation (c)

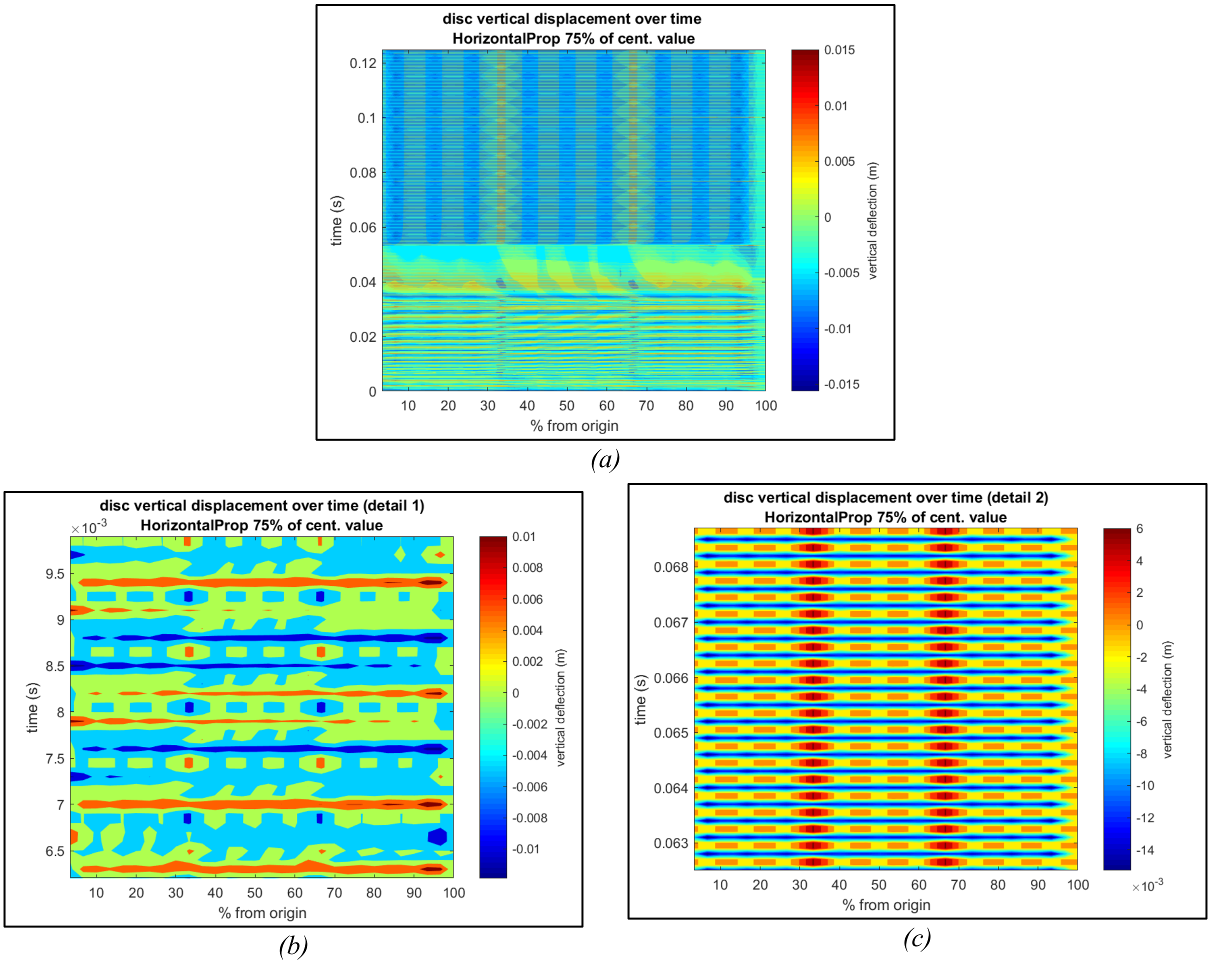


Figure 45 – Disc surface vertical displacement with 75% horizontal stiffness and damping, for full simulation duration (a), in early state of simulation (b), and during stable part of simulation (c)

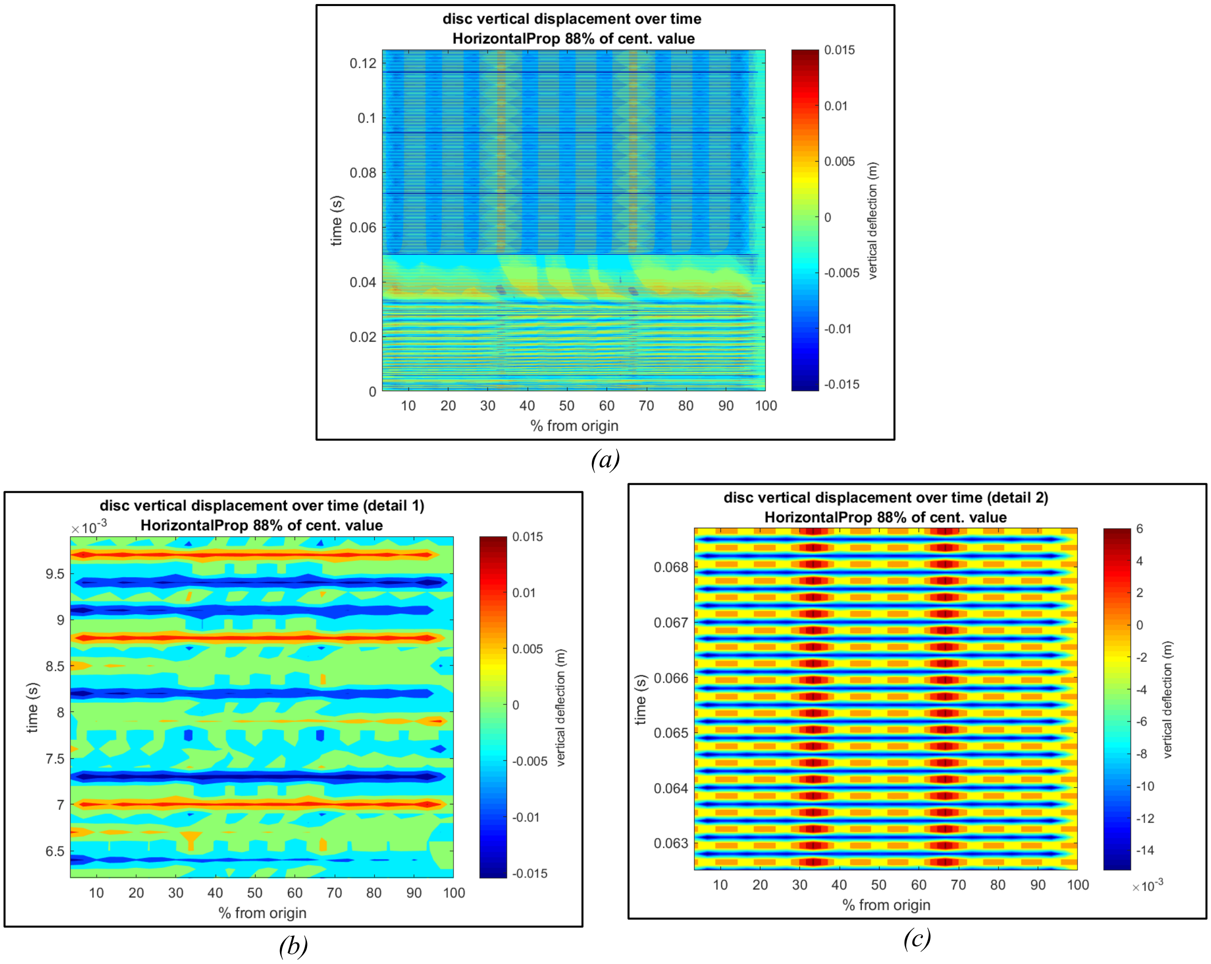


Figure 46 – Disc surface vertical displacement with 88% horizontal stiffness and damping, for full simulation duration (a), in early state of simulation (b), and during stable part of simulation (c)

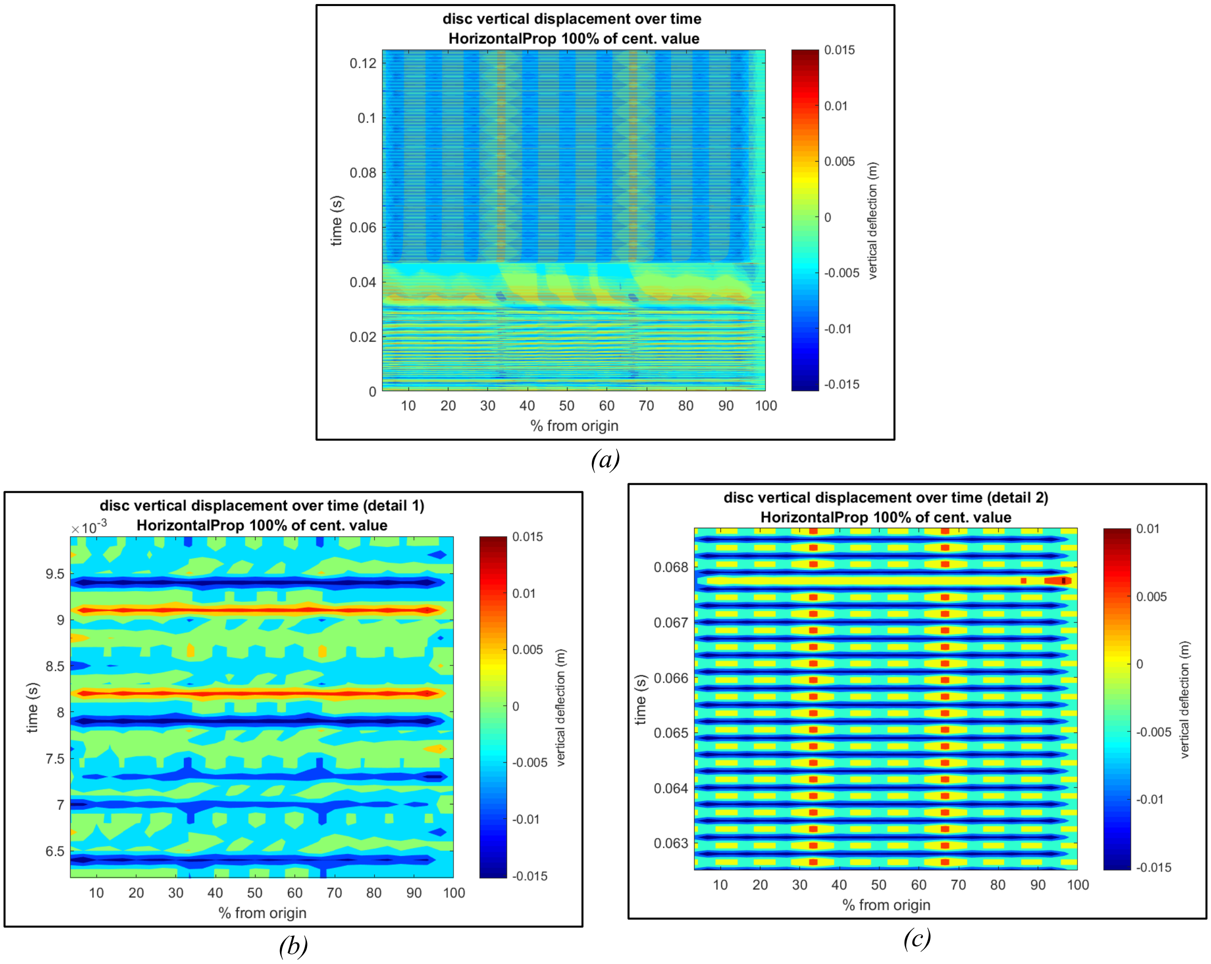
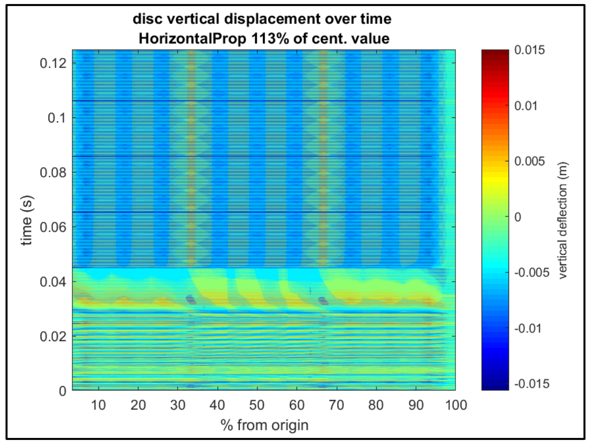
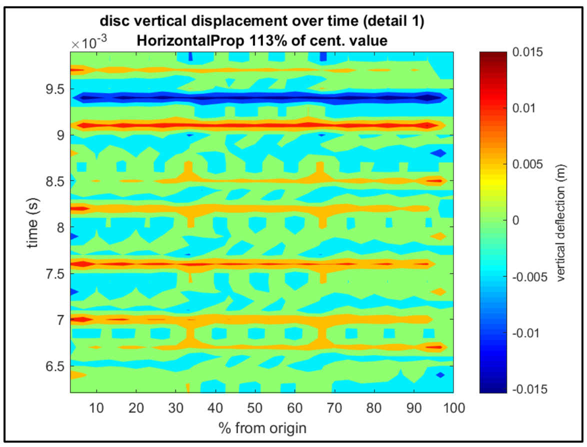


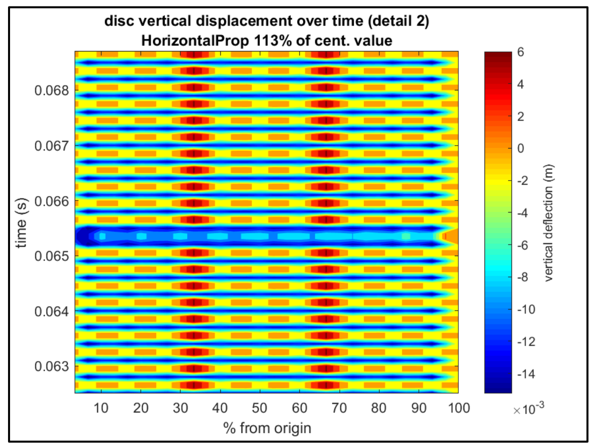
Figure 47 – Disc surface vertical displacement with 100% horizontal stiffness and damping, for full simulation duration (a), in early state of simulation (b), and during stable part of simulation (c)



(a)



(b)



(c)

Figure 48 – Disc surface vertical displacement with 113% horizontal stiffness and damping, for full simulation duration (a), in early state of simulation (b), and during stable part of simulation (c)

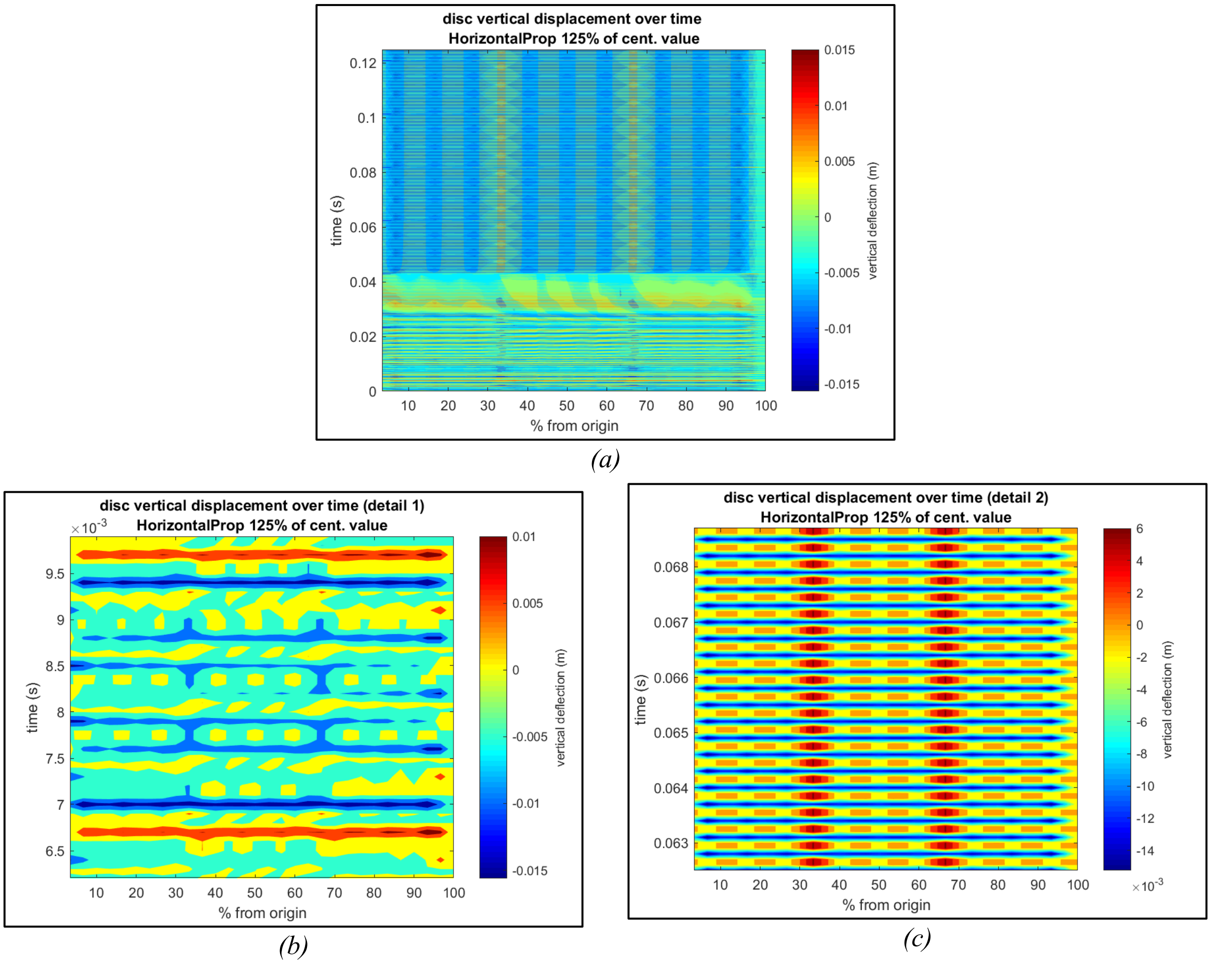


Figure 49 – Disc surface vertical displacement with 125% horizontal stiffness and damping, for full simulation duration (a), in early state of simulation (b), and during stable part of simulation (c)

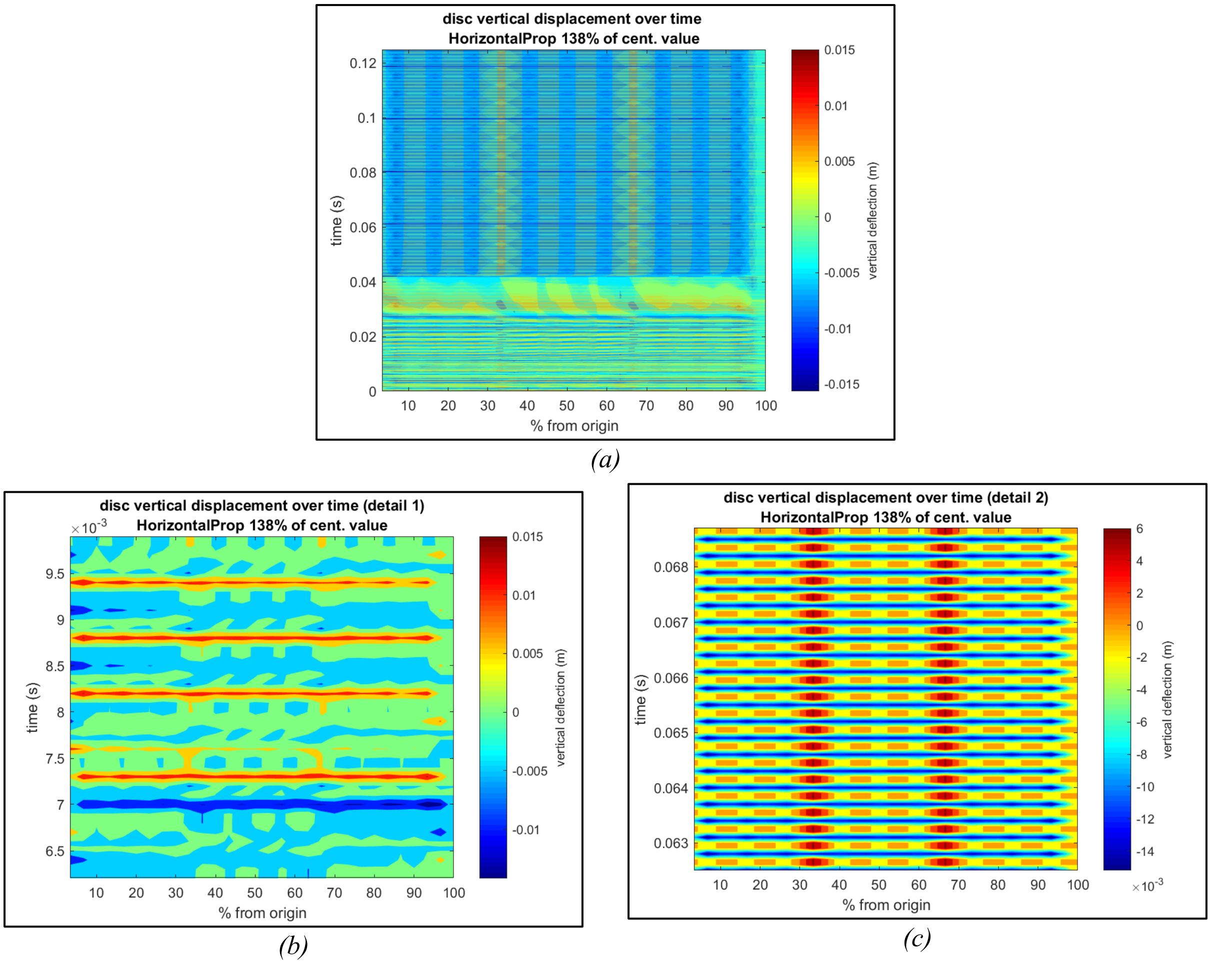


Figure 50 – Disc surface vertical displacement with 138% horizontal stiffness and damping, for full simulation duration (a), in early state of simulation (b), and during stable part of simulation (c)

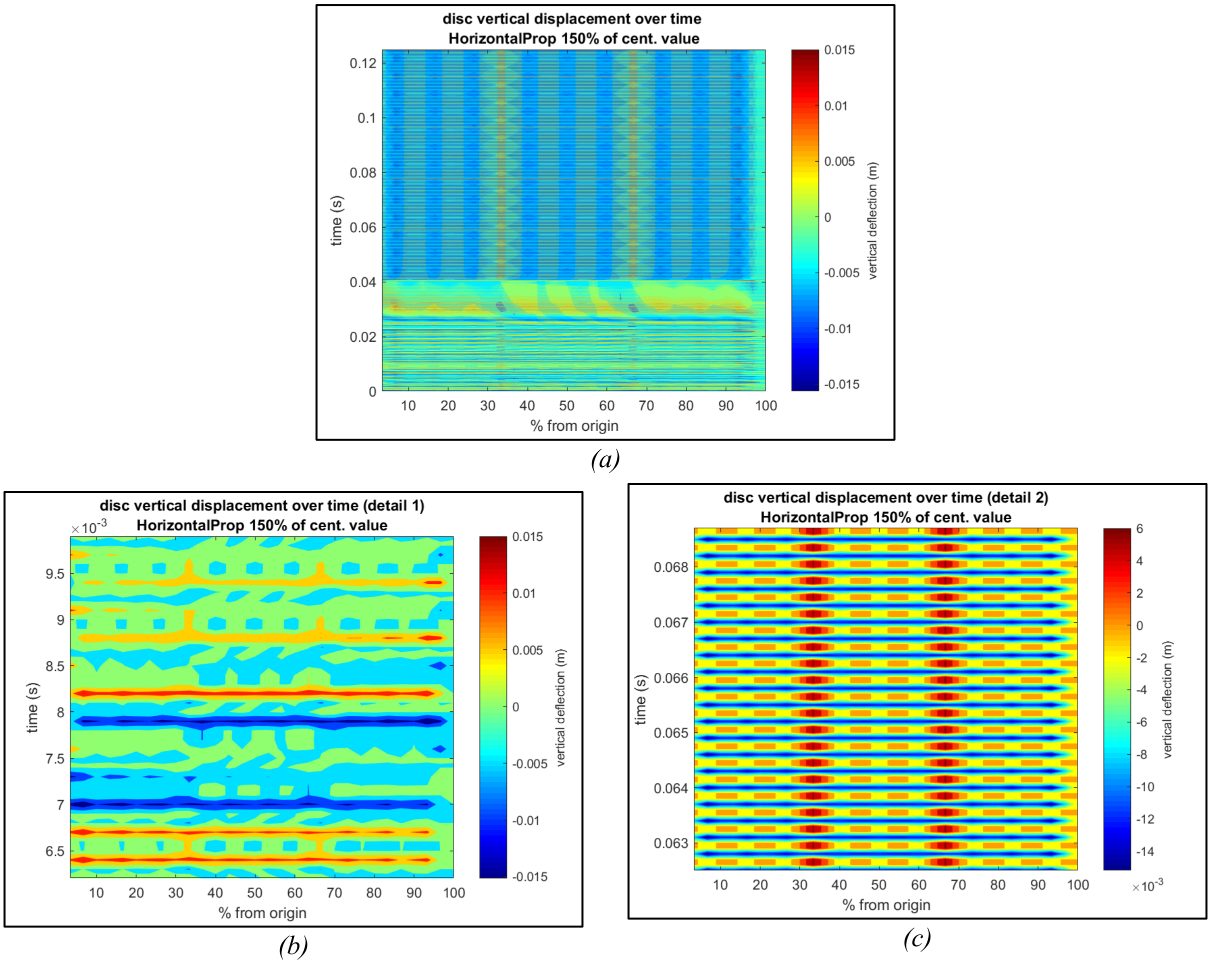
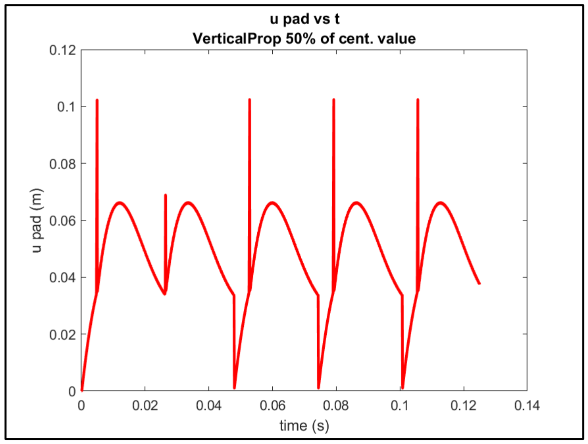


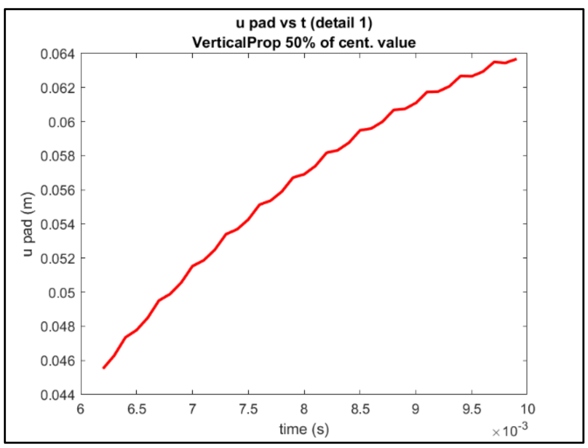
Figure 51 – Disc surface vertical displacement with 150% horizontal stiffness and damping, for full simulation duration (a), in early state of simulation (b), and during stable part of simulation (c)

D.3. Vertical Stiffness and Damping Ratio Variation

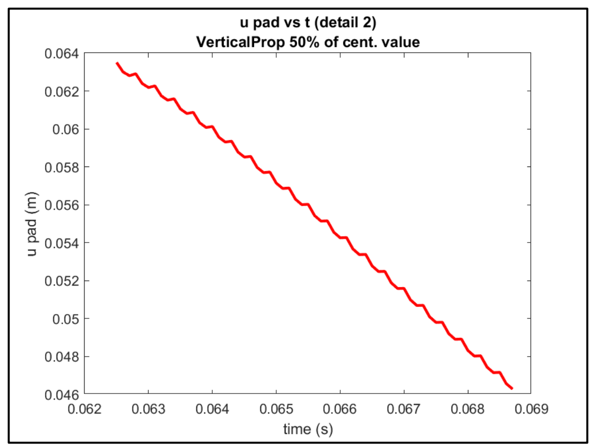
D.3.1. Pad Horizontal Displacement



(a)

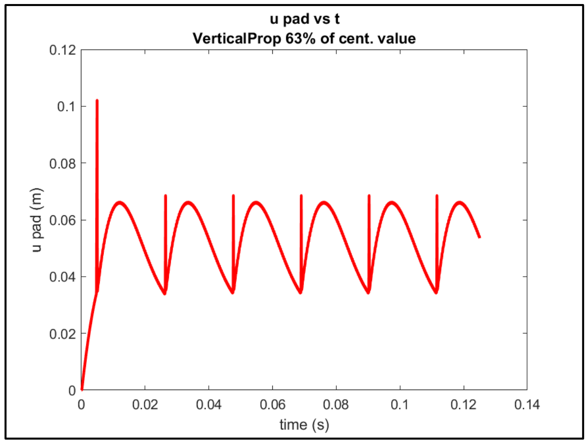


(b)

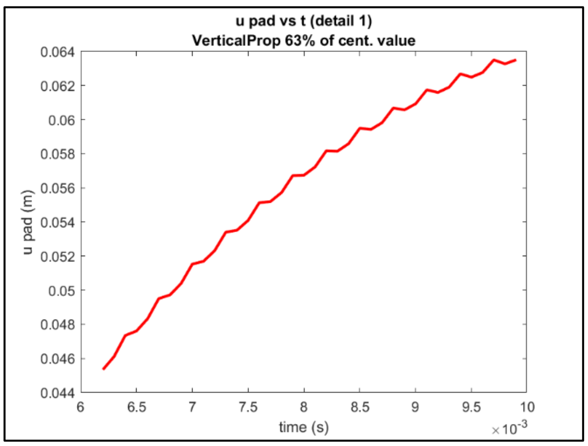


(c)

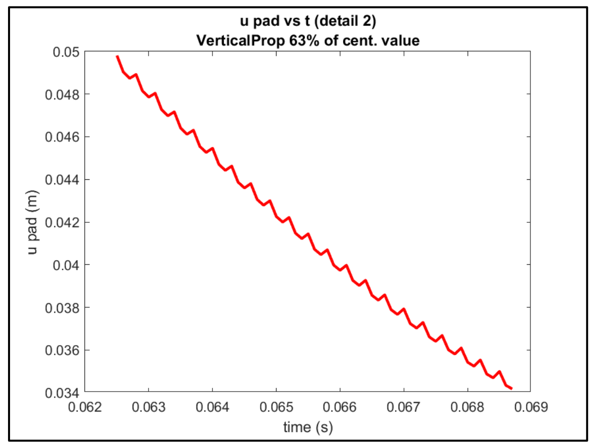
Figure 52 – Pad horizontal displacement with 50% vertical stiffness and damping, for full simulation duration (a), in early state of simulation (b), and during stable part of simulation (c)



(a)



(b)



(c)

Figure 53 – Pad horizontal displacement with 63% vertical stiffness and damping, for full simulation duration (a), in early state of simulation (b), and during stable part of simulation (c)

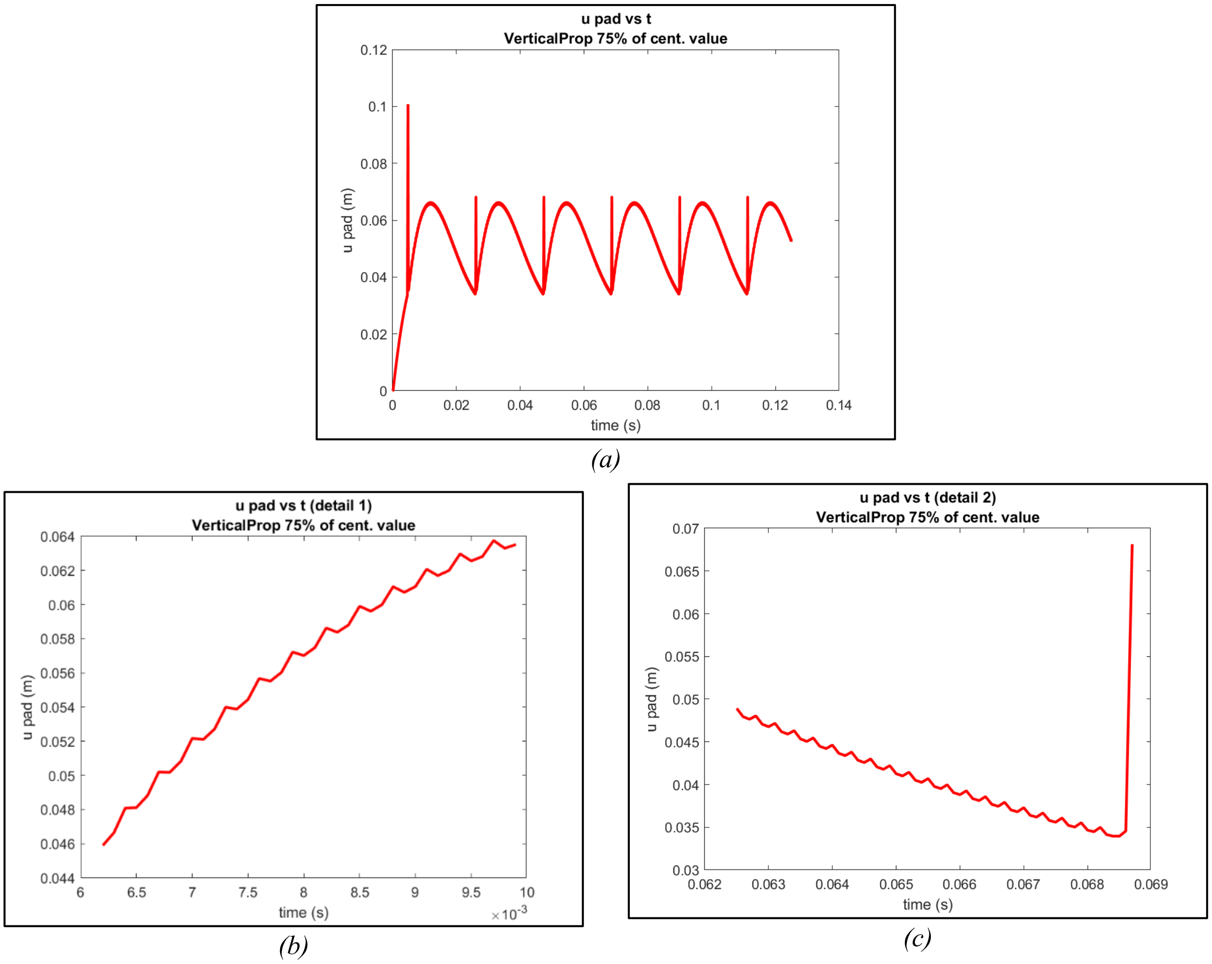
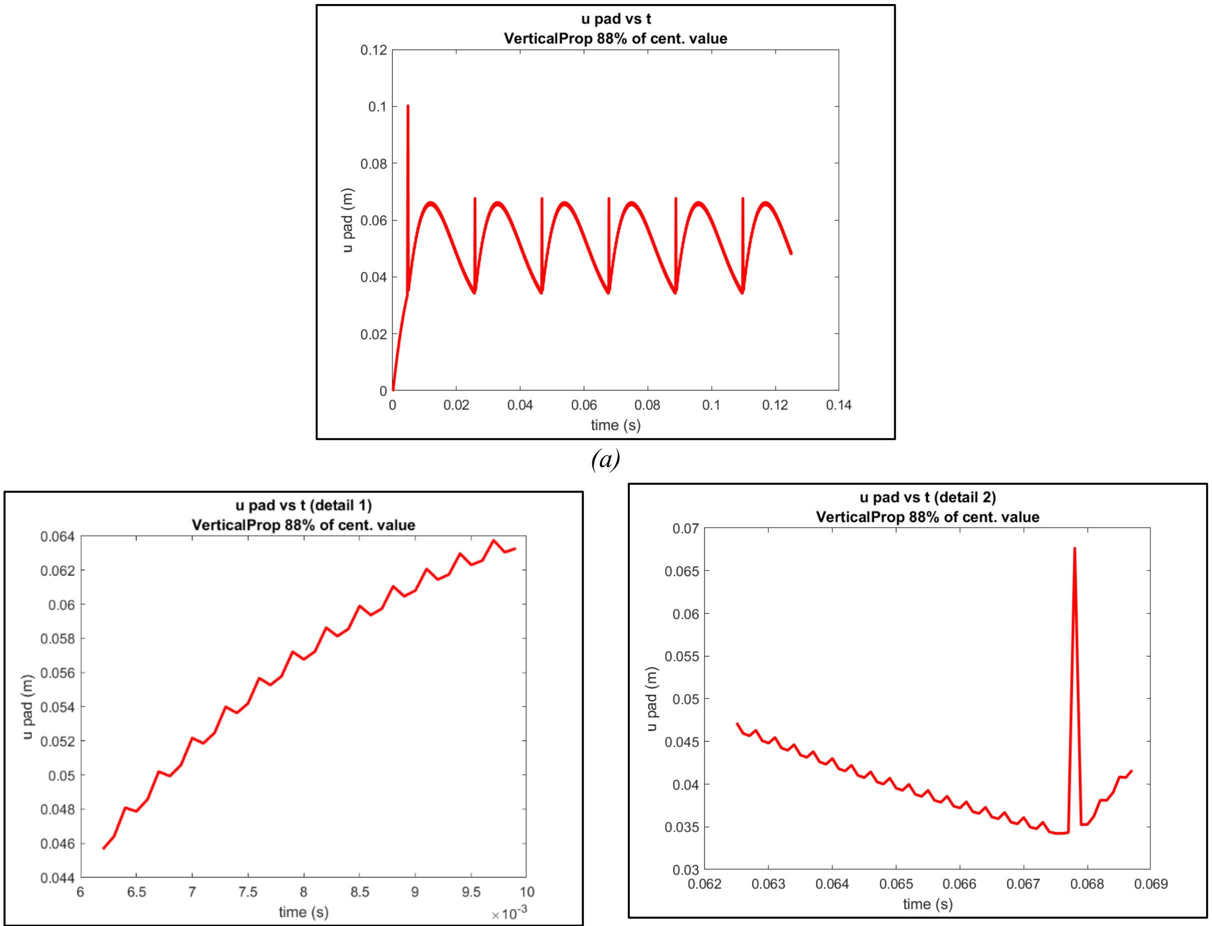
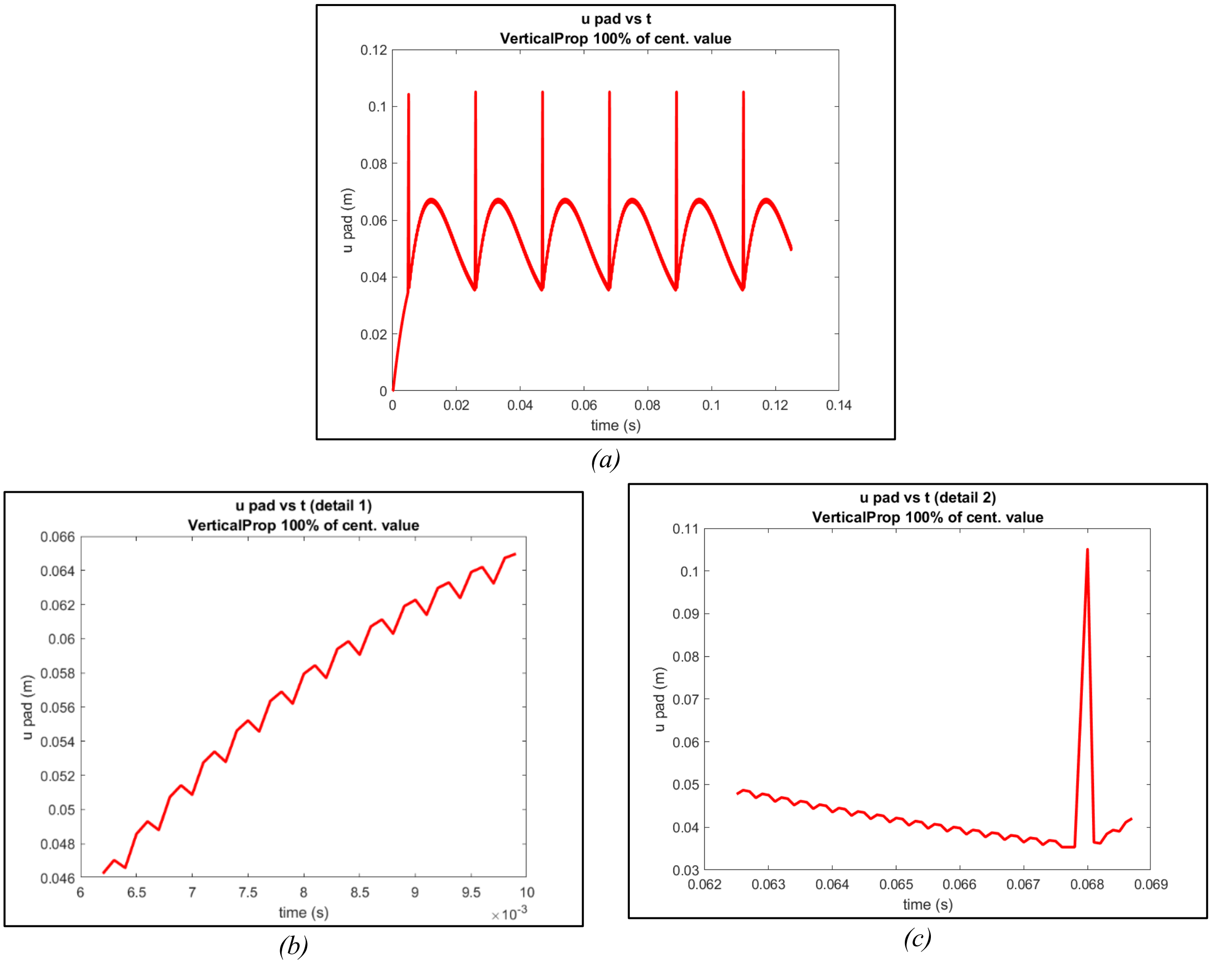


Figure 54 – Pad horizontal displacement with 75% vertical stiffness and damping, for full simulation duration (a), in early state of simulation (b), and during stable part of simulation (c)



(b) *(c)*

Figure 55 – Pad horizontal displacement with 88% vertical stiffness and damping, for full simulation duration (a), in early state of simulation (b), and during stable part of simulation (c)



(b) (c)

Figure 56 – Pad horizontal displacement with 100% vertical stiffness and damping, for full simulation duration (a), in early state of simulation (b), and during stable part of simulation (c)

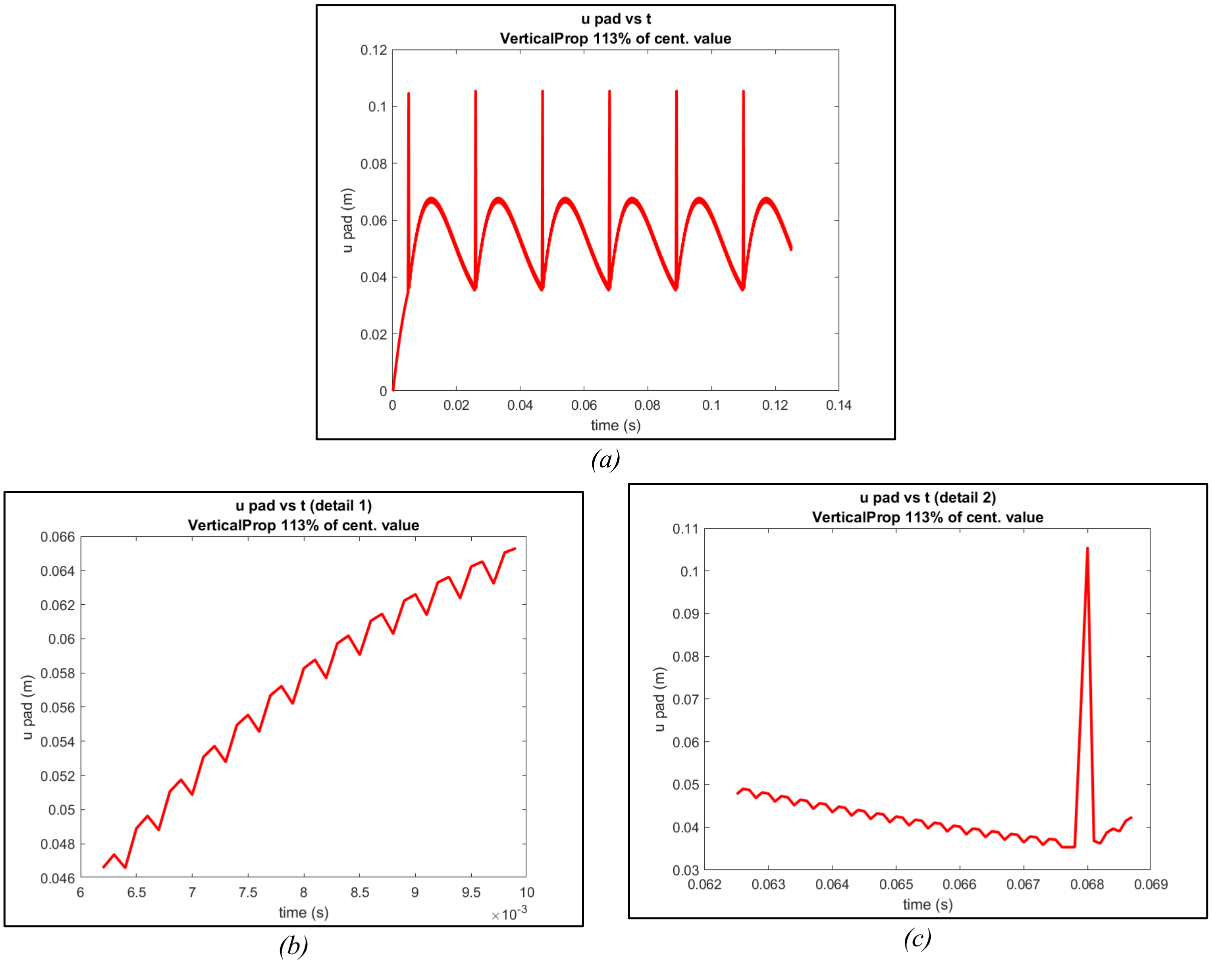
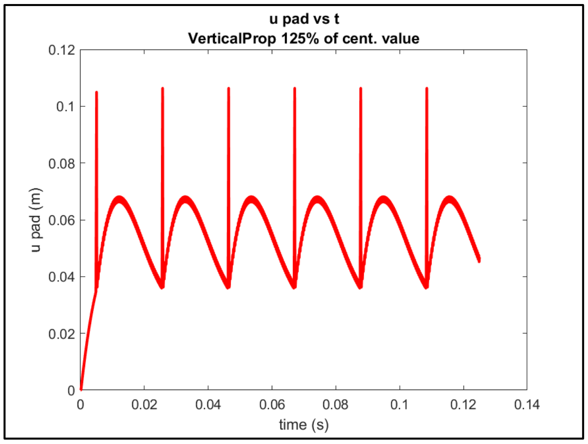
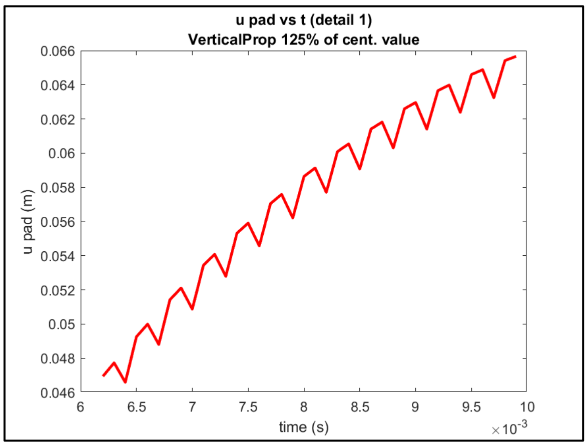


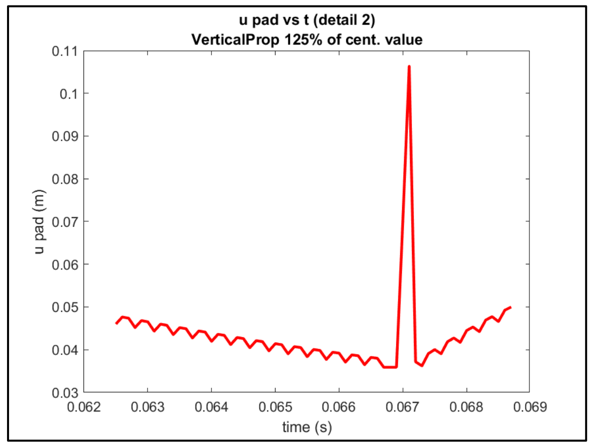
Figure 57 – Pad horizontal displacement with 113% vertical stiffness and damping, for full simulation duration (a), in early state of simulation (b), and during stable part of simulation (c)



(a)



(b)



(c)

Figure 58 – Pad horizontal displacement with 125% vertical stiffness and damping, for full simulation duration (a), in early state of simulation (b), and during stable part of simulation (c)

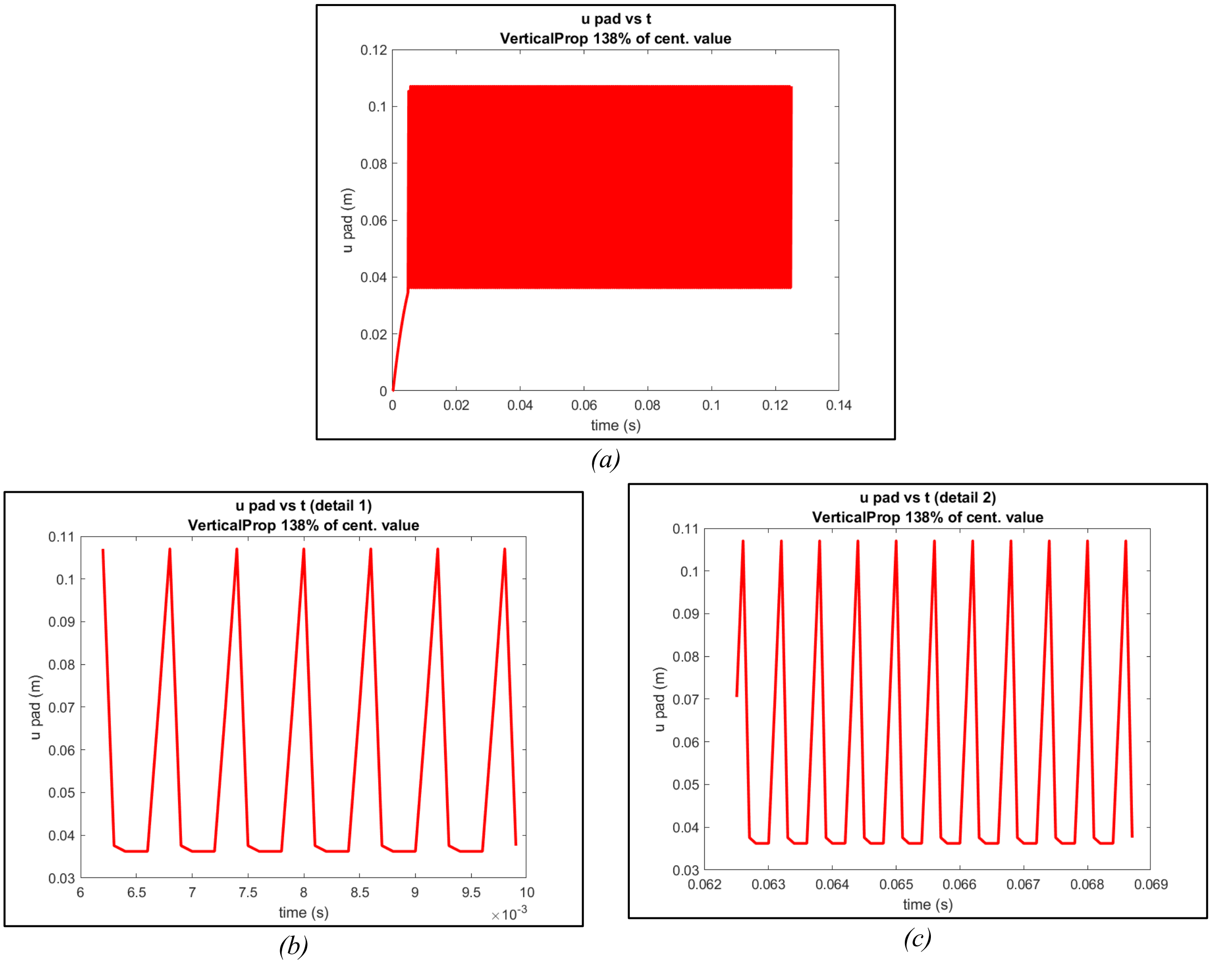


Figure 59 – Pad horizontal displacement with 138% vertical stiffness and damping, for full simulation duration (a), in early state of simulation (b), and during stable part of simulation (c)

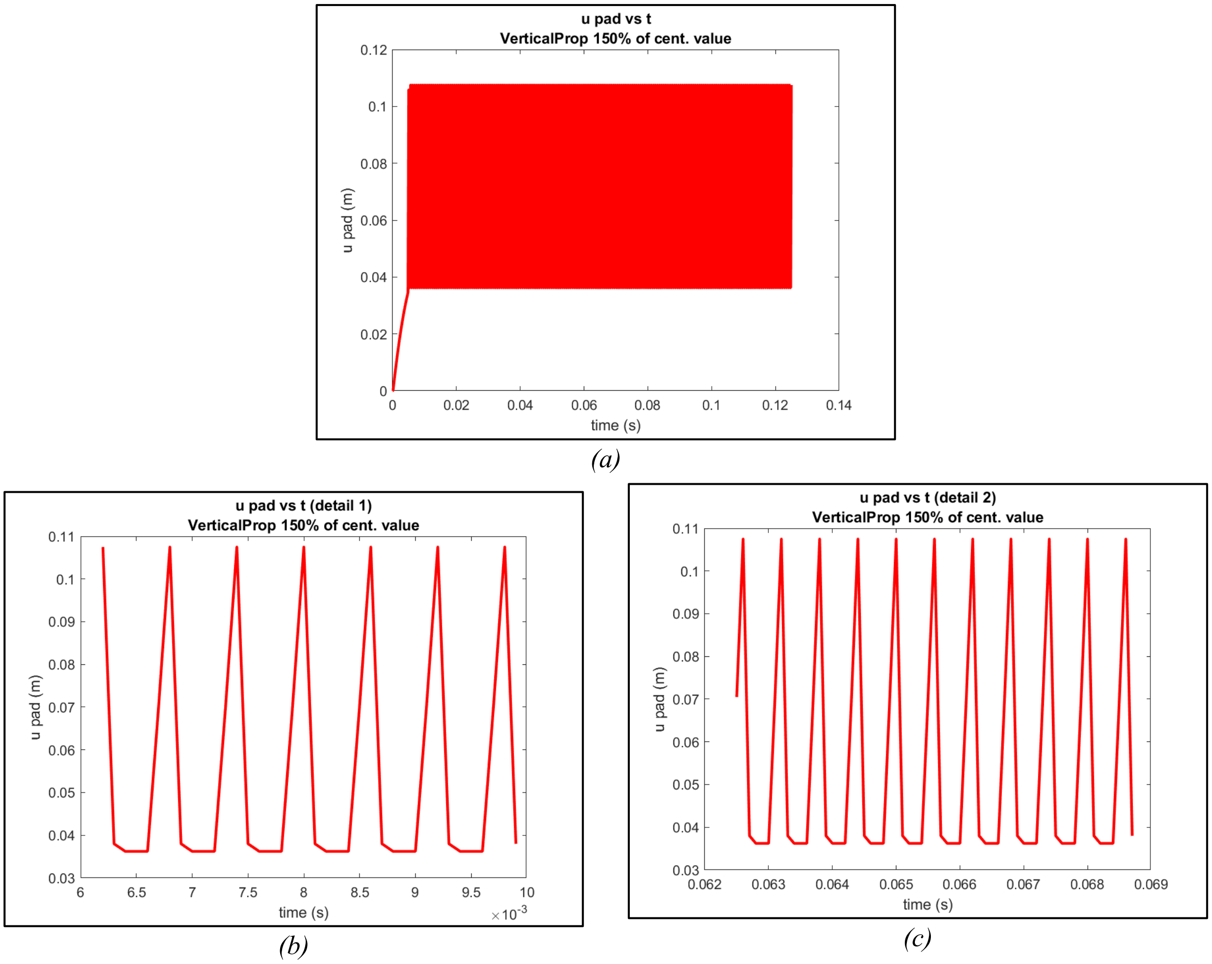
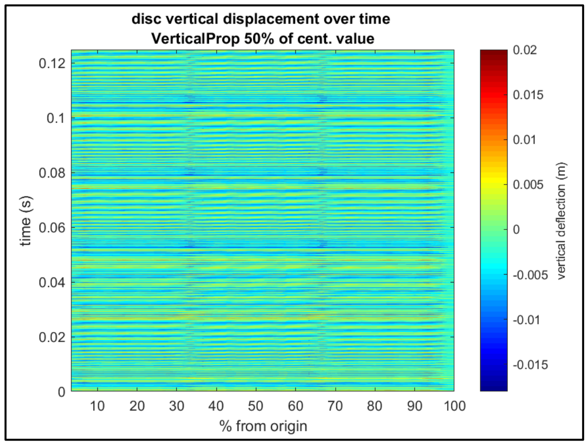
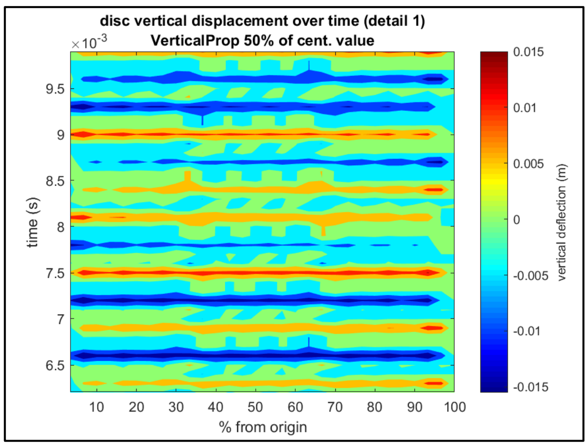


Figure 60 – Pad horizontal displacement with 150% vertical stiffness and damping, for full simulation duration (a), in early state of simulation (b), and during stable part of simulation (c)

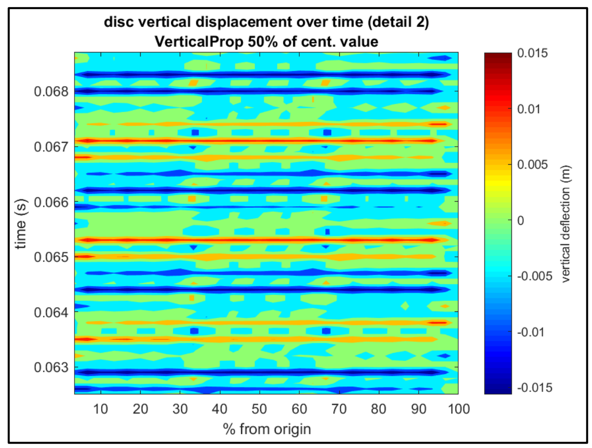
D.3.2. Disc Surface Vertical Displacement



(a)



(b)



(c)

Figure 61 – Disc surface vertical displacement with 50% vertical stiffness and damping, for full simulation duration (a), in early state of simulation (b), and during stable part of simulation (c)

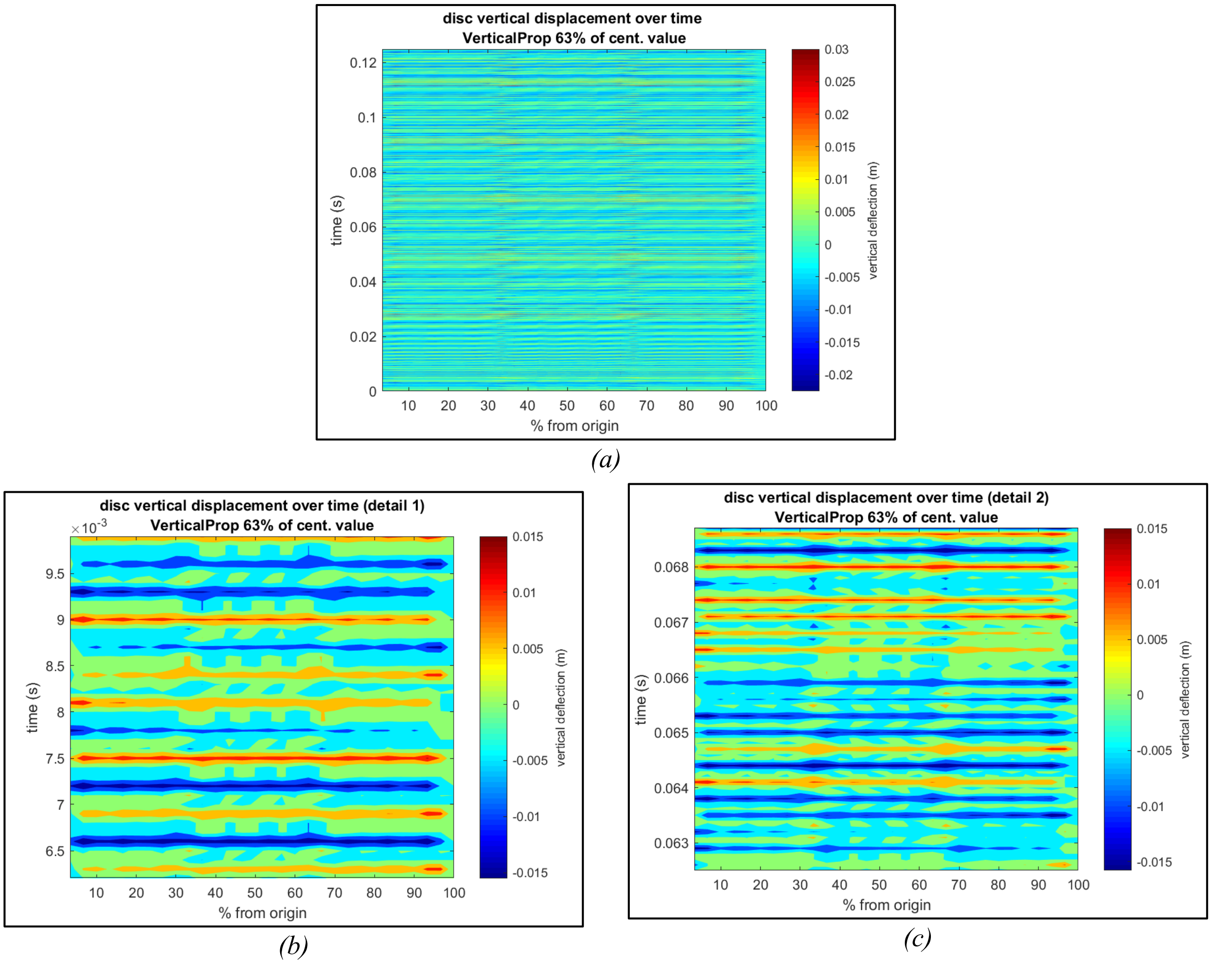
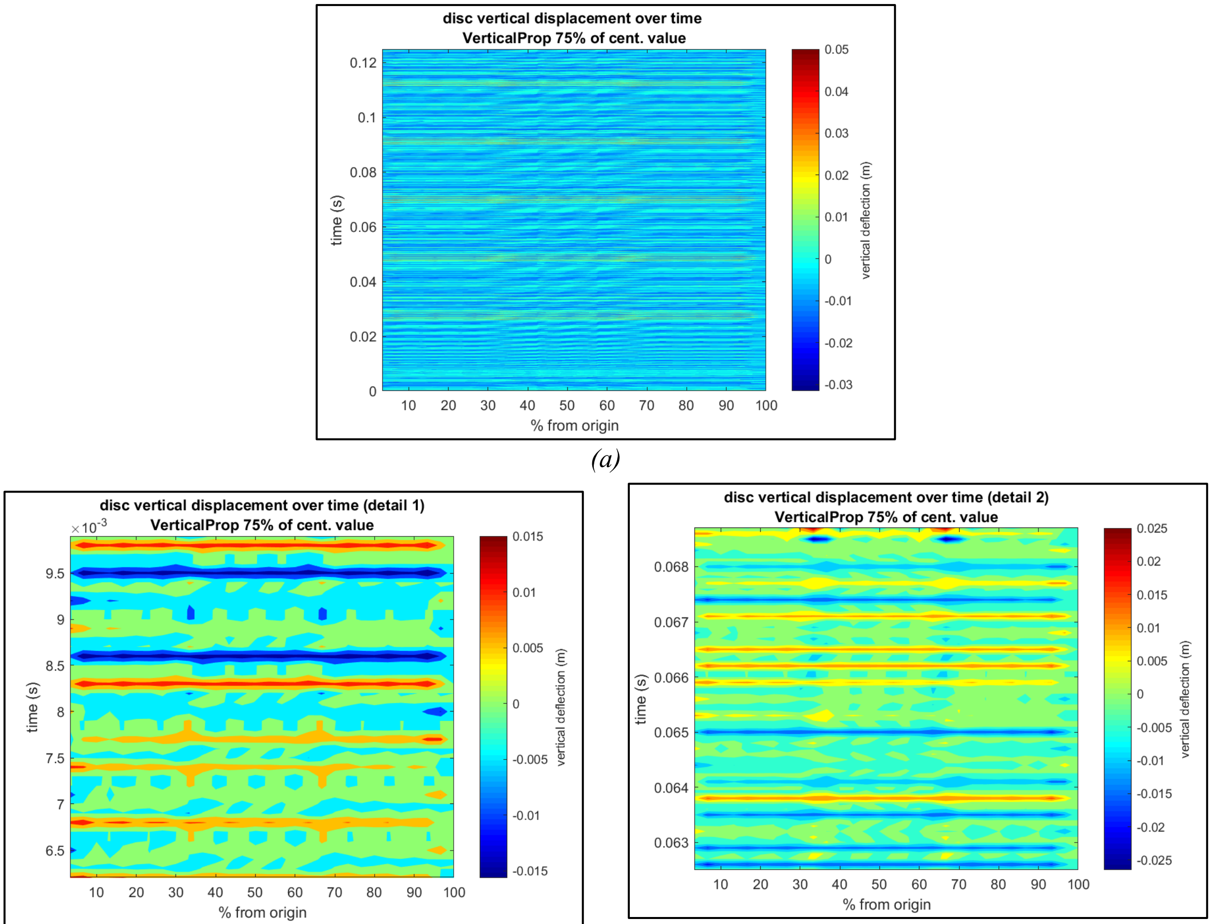


Figure 62 – Disc surface vertical displacement with 63% vertical stiffness and damping, for full simulation duration (a), in early state of simulation (b), and during stable part of simulation (c)



(a)

(b) *(c)*

Figure 63 – Disc surface vertical displacement with 75% vertical stiffness and damping, for full simulation duration (a), in early state of simulation (b), and during stable part of simulation (c)

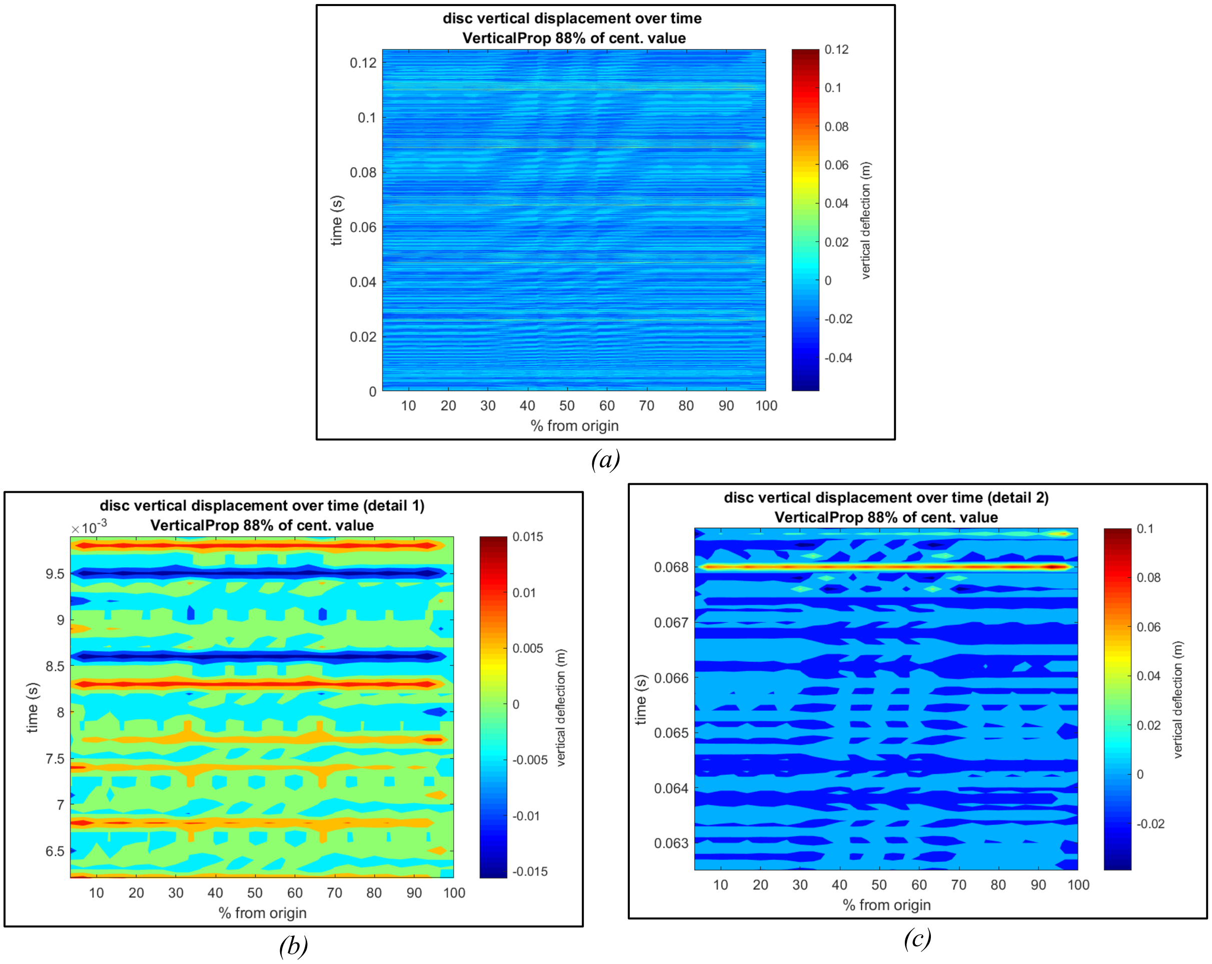


Figure 64 – Disc surface vertical displacement with 88% vertical stiffness and damping, for full simulation duration (a), in early state of simulation (b), and during stable part of simulation (c)

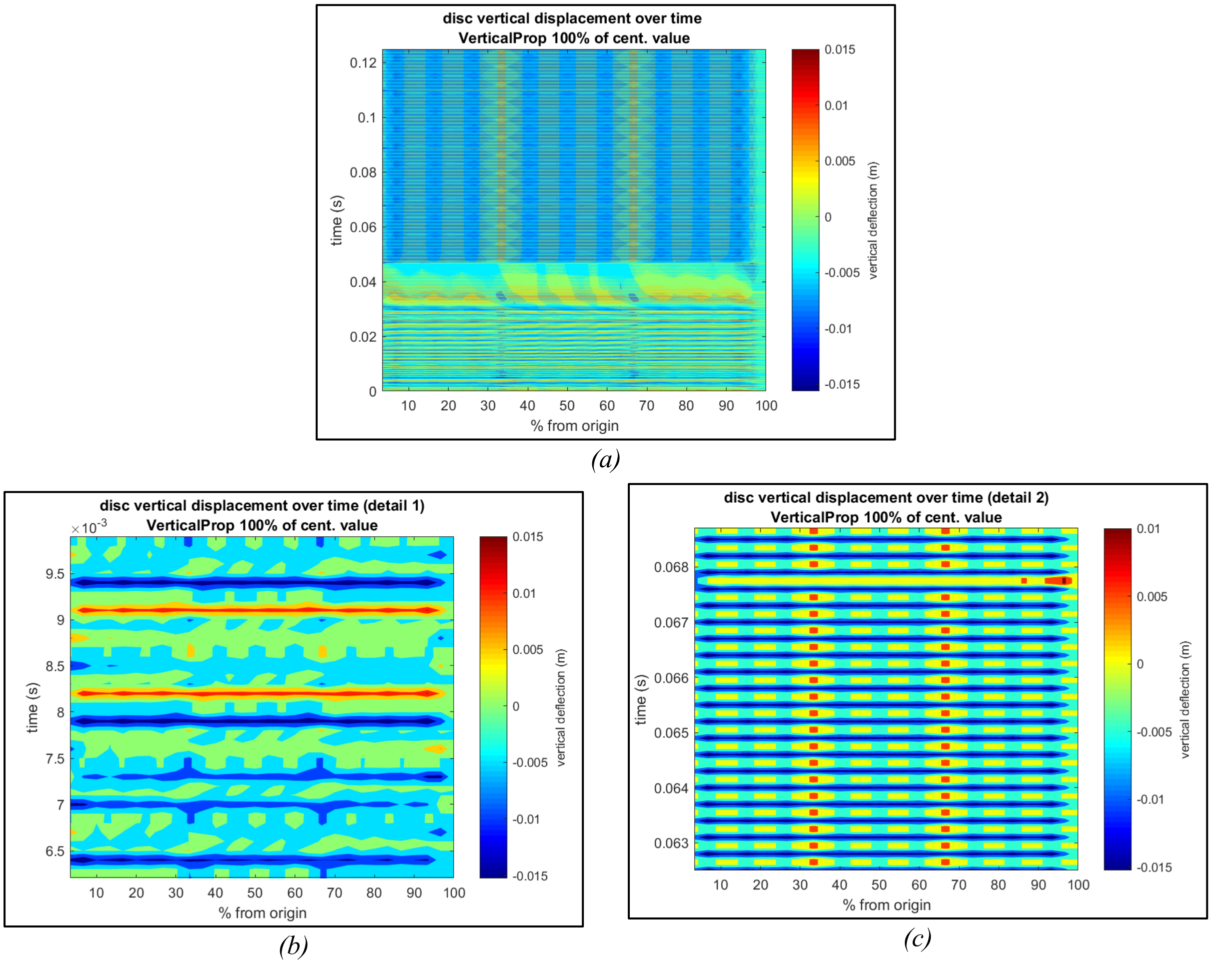
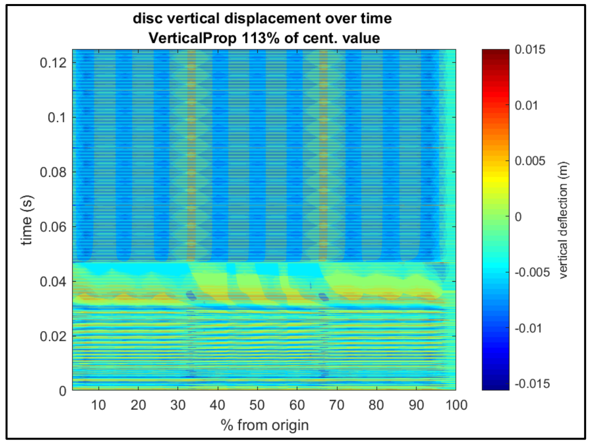
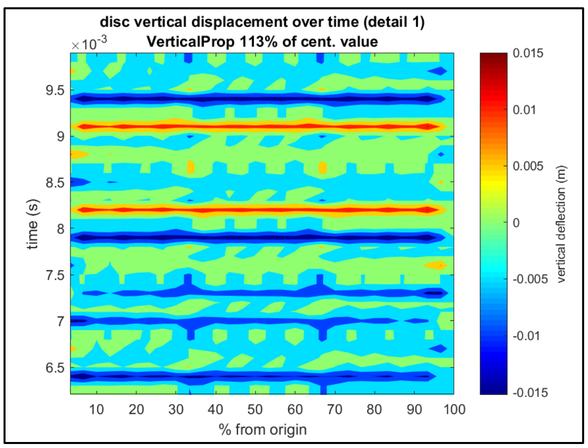


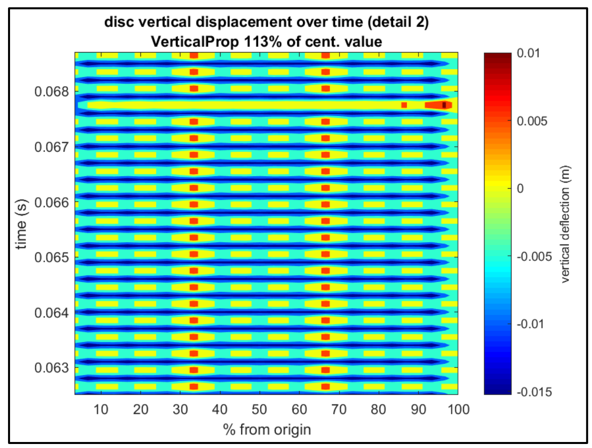
Figure 65 – Disc surface vertical displacement with 100% vertical stiffness and damping, for full simulation duration (a), in early state of simulation (b), and during stable part of simulation (c)



(a)

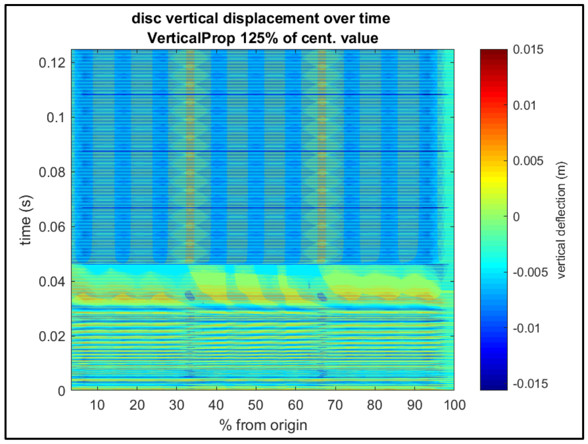


(b)

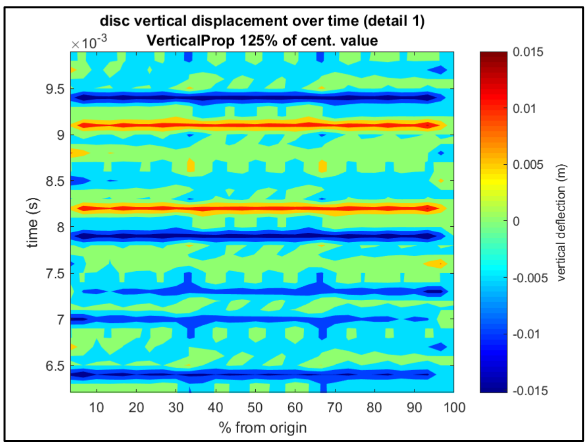


(c)

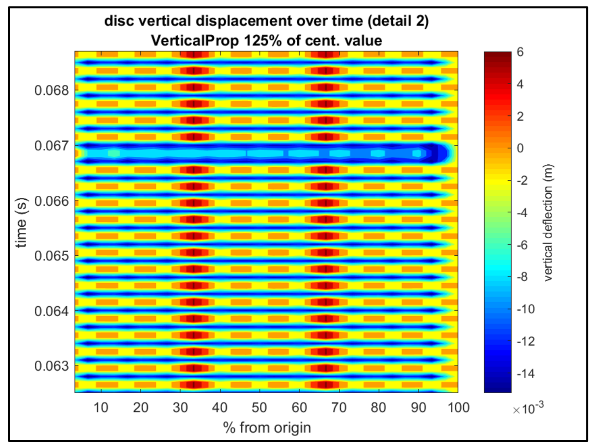
Figure 66 – Disc surface vertical displacement with 113% vertical stiffness and damping, for full simulation duration (a), in early state of simulation (b), and during stable part of simulation (c)



(a)



(b)



(c)

Figure 67 – Disc surface vertical displacement with 125% vertical stiffness and damping, for full simulation duration (a), in early state of simulation (b), and during stable part of simulation (c)

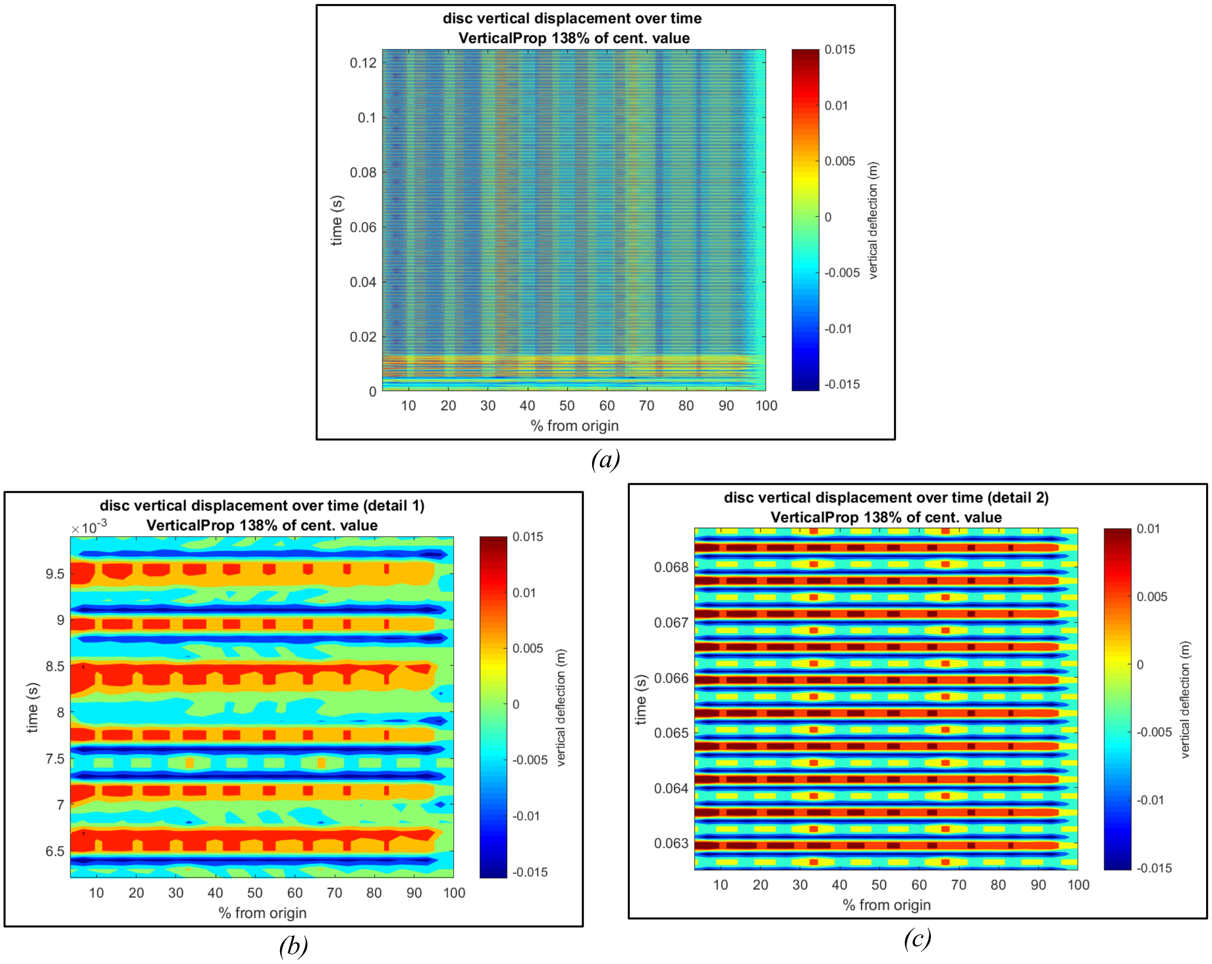
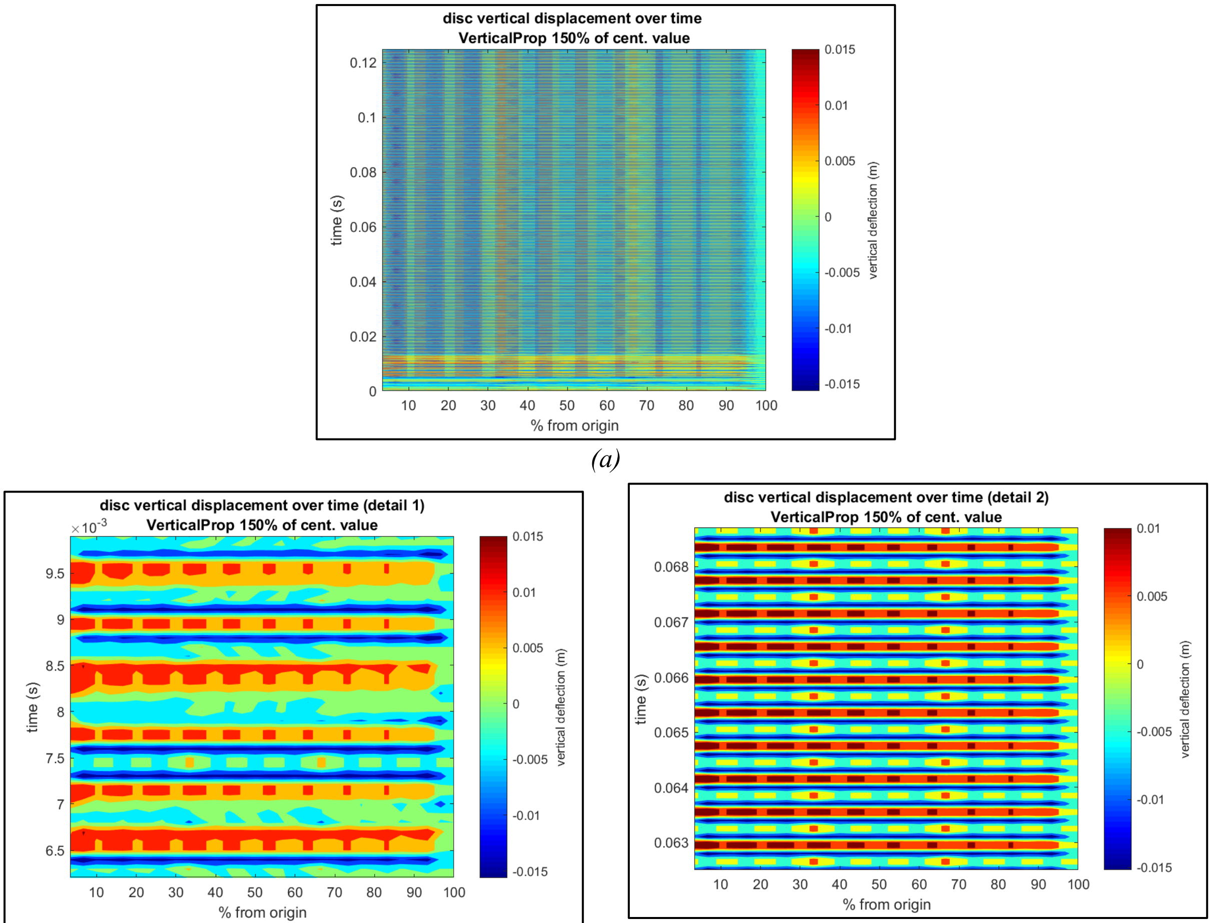


Figure 68 – Disc surface vertical displacement with 138% vertical stiffness and damping, for full simulation duration (a), in early state of simulation (b), and during stable part of simulation (c)



(a)

(b) *(c)*

Figure 69 – Disc surface vertical displacement with 150% vertical stiffness and damping, for full simulation duration (a), in early state of simulation (b), and during stable part of simulation (c)

References

[1] Akay, A., 2002, "Acoustics of Friction," J. Acoust. Soc. Am., **111**(4), pp. 1525–1548.

[2] Kinkaid, N. M., Reilly, O. M. O., and Papadopoulos, P., 2003, "Automotive Disc Brake Squeal," J. Sound Vib., **267**(1), pp. 105–166.

[3] Bottega, W. J., 2006, *Engineering Vibrations*, Taylor & Francis, Boca Raton, Fl.

[4] Woodhouse, J., and Galluzzo, P. M., 2004, "The Bowed String as We Know It Today," Acta Acust. united with Acust., **90**(4), pp. 579–589.

[5] Guettler, K., 1994, "Wave Analysis of a String Bowed to Anomalous Low Frequencies.," J. Catgut Acoust. Soc., **2**(6), pp. 8–14.

[6] Jarvis, B. R. P., and Mills, B., 1963, "Vibrations Induced by Dry Friction," *Proceedings of the Institution of Mechanical Engineers*, pp. 847–857.

[7] Spurr, B. R. T., Ph, D., and Sc, B., 1961, "A theory of brake squeal," *Proceedings of the Institution of Mechanical Engineers: Automobile Division*, pp. 33–52.

[8] Earles, S. W. E., and Lee, C. K., 1976, "Instabilities Arising From the Frictional Interaction of a Pin-Disk System Resulting in Noise Generation," J. Eng. Ind., **98**(1), pp. 81–86.

[9] Fieldhouse, J. D., and Newcomb, P., 1993, "The Application of Holographic Interferometry to the Study of Disc Brake Noise," SAE Tech. Pap., (No. 930805).

[10] Giannini, O., Akay, A., and Massi, F., 2006, "Experimental Analysis of Brake Squeal Noise on a Laboratory Brake Setup," J. Sound Vib., **292**(1–2), pp. 1–20.

[11] North, M. R., 1972, "Frictionally Induced, Self Excited Vibrations in a Disc Brake System," Loughborough University.

[12] Papinniemi, A., Lai, J. C. S., and Zhao, J., 2002, "Brake Squeal : A Literature Review," **63**, pp. 391–400.

[13] Sinou, J., Dereure, O., Mazet, G., Thouverez, F., and Jezequel, L., 2006, "Friction-Induced Vibration for an Aircraft Brake System — Part 1 : Experimental Approach and Stability Analysis," Int. J. Mech. Sci., **48**, pp. 536–554.

[14] Nechak, L., Berger, S., and Aubry, E., 2011, "European Journal of Mechanics A / Solids A Polynomial Chaos Approach to the Robust Analysis of the Dynamic Behaviour of Friction Systems," Eur. J. Mech. / A Solids, **30**(4), pp. 594–607.

[15] Computations, A. E., and Collection, I., 2002, "Judder Vibration in Disc Brakes Excited by Thermoelastic Instability."

[16] Allgaier, R., Gaul, L., Keiper, W., Willner, K., and Ho, N., 2002, "A study on brake squeal using a beam-on-disc model," pp. 528–534.

[17] Oberst, S., 1996, "A Critical Review of Brake Squeal and Its Treatment in Practice," (Figure 1).

[18] Nouby, M., Mathivanan, D., and Srinivasan, K., 2009, "A Combined Approach of Complex Eigenvalue Analysis and Design of Experiments (DOE) to Study Disc Brake Squeal," Int.

J. Eng. Sci. Technol., **1**(1), pp. 254–271.

[19] Fritz, G., Sinou, J., Duffal, J., Jezequel, L., Fritz, G., Sinou, J., Duffal, J., and Jezequel, L., 2008, “Effects of Damping on Brake Squeal Coalescence Patterns - Application on a Finite Element Model To Cite This Version : HAL Id : Hal-00207796 Effects of Damping on Brake Squeal Coalescence Patterns - Application on a Finite Element Model.”

[20] Ghazaly, N. M., Mohammed, S., and Abd-El-Tawwab, A. M., 2012, “Understanding Mode-Coupling Mechanism of Brake Squeal Using Finite Element Analysis,” Int. J. Eng. Res. Appl., **2**(1), pp. 241–250.

[21] “Product Catalog,” Clevel. Wheel. Brakes - Park. Hannifin Corp. Aircr. Wheel Brake Div., **July 01**, **2**(Issue 24), p. 174.

[22] Cleveland Wheels & Brakes - Parker Hannifin Corporation Aircraft Wheel & Brake Division, 2010, “Product Reference Memo: PC-12 Brake lining conditioning procedure.”

[23] Bengisu, M. T., and Akay, A., 1994, “Stability of Friction-Induced Vibrations in Multi-Degree-of-Freedom Systems,” J. Sound Vib., **171**(4), pp. 557–570.