A Harmonic Balance Approach for Designing Compliant Mechanical Systems with Nonlinear Periodic Motions

PENGBIN TANG, Université de Montréal
JONAS ZEHNDER, Université de Montréal
STELIAN COROS, ETH Zürich
BERNHARD THOMASZEWSKI, Université de Montréal & ETH Zürich
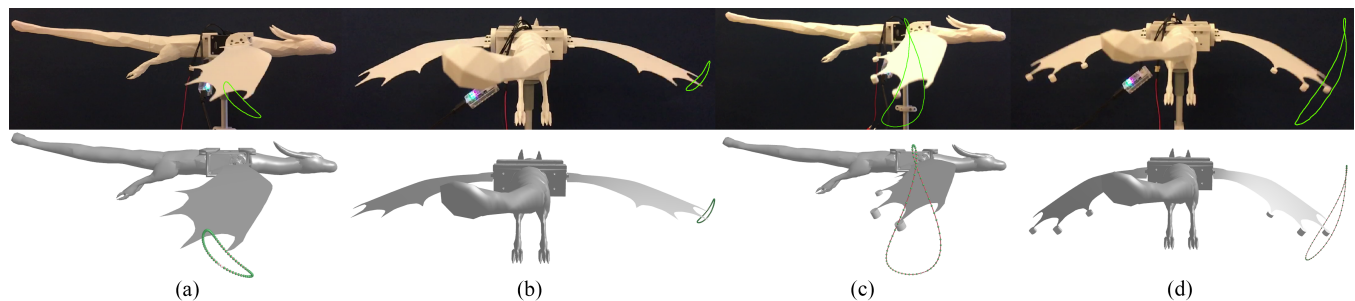
(a)   (b)   (c)   (d)

Fig. 1. Our method enables optimization-driven design of compliant mechanical systems with periodic large-amplitude motions. For this pair of dragon wings, the initial design (*a, b*) exhibits only small oscillation response when driven by harmonic forcing at a frequency of $2.5Hz$. Our approach automatically finds optimized design parameters (extra masses at the trailing edge of the wing) that lead to substantially increased amplitude (*c, d*).

We present a computational method for designing compliant mechanical systems that exhibit large-amplitude oscillations. The technical core of our approach is an optimization-driven design tool that combines sensitivity analysis for optimization with the Harmonic Balance Method for simulation. By establishing dynamic force equilibrium in the frequency domain, our formulation avoids the major limitations of existing alternatives: it handles nonlinear forces, side-steps any transient process, and automatically produces periodic solutions. We introduce design objectives for amplitude optimization and trajectory matching that enable intuitive high-level authoring of large-amplitude motions. Our method can be applied to many types of mechanical systems, which we demonstrate through a set of examples involving compliant mechanisms, flexible rod networks, elastic thin shell models, and multi-material solids. We further validate our approach by manufacturing and evaluating several physical prototypes.

CCS Concepts: • **Computing methodologies → Physical simulation**;
• **Computer graphics → Physically based modeling**; • **Applied computing → Computer-aided design**.

Authors' addresses: Pengbin Tang, Université de Montréal, pengbin.tang@umontreal.ca; Jonas Zehnder, Université de Montréal, jonas.zehnder@umontreal.ca; Stelian Coros, ETH Zürich, stelian@inf.ethz.ch; Bernhard Thomaszewski, Université de Montréal & ETH Zürich, bernhard@iro.umontreal.ca.

1 INTRODUCTION

From the seismic response of high-rise buildings, to the aeroelastic stability of turbine blades, and to the micro-vibrations of energy harvesting devices—understanding and controlling the behavior of mechanical systems subject to periodic forcing is key to many engineering applications. Designing for vibrations typically means bracing against *resonance*, i.e., the strong increase in oscillation amplitude that occurs when driving a system at its characteristic frequency. Unless explicitly prevented, resonance can induce increasingly large deformations that, ultimately, lead to failure. While avoiding resonance is therefore often a main design goal, in this paper we explore the design of flexible structures that exploit resonance to produce periodic motion in the form of large-amplitude oscillations.

Modeling Periodic Motion. Most real-world systems will reach a stable steady state motion—a so called limit cycle—when driven by a periodic force. However, before reaching its limit cycle, the system goes through a transient process of non-periodic motion whose length depends on the complexity as well as the elastic and viscous properties of the system. One approach to computing limit cycles is to simulate this transient process by numerically solving the equations of motion until some periodicity condition is met. However, using such a time-domain approach as a basis for computing system parameters that yield desired steady state behavior would require inverting a transient process of *a priori* unknown length, which is computationally all but intractable.

Frequency-Domain Approaches. A more promising approach for designing limit cycles is to use frequency-domain methods that directly solve for the system's steady state behavior without the need for simulating the transient process. As another advantage, periodicity is obtained by construction and does not have to be enforced explicitly. Arguably the most widely used frequency-domain method for vibration analysis is Linear Modal Analysis (LMA) [Shabana 1990]: based on the assumption of small-amplitude oscillations, the equations of motion are linearized around the origin and transformed to frequency space using generalized Eigenvalue decomposition. The resulting modal equations decouple and can be solved efficiently to yield the approximate limit response of the system to excitation at arbitrary frequencies. LMA is particularly well suited when vibrations and resonance are to be avoided and, hence, the assumption of small oscillations is valid. For compliant mechanisms and other systems with large-amplitude oscillations and finite rotations, however, nonlinearities play a critical role and LMA is unable to predict the steady state behavior in such cases.

To embrace nonlinearities from the start, we build our approach on the basis of the Harmonic Balance Method (HBM)—a nonlinear frequency-domain method that extends to a wide range of vibration problems [Krack and Gross 2019]. The basic idea of HBM is to project the time-continuous equations of motion to a finite-dimensional subspace spanned by a small number of trigonometric basis functions. These nonlinear basis functions allow for efficient modeling of nonlinearities in nodal positions, velocities, and forces.

Overview & Contributions. Using HBM as a basis, we propose an optimization-based design tool that leverages sensitivity analysis to automatically discover design parameters that best approximate target steady-state behavior. Our method is able to control steady-state motion based on high-level user input by adjusting mass distribution, stiffness, or shape parameters of the designs. By performing simulation and design optimization directly in frequency space, our method avoids long transition times, the need for periodicity constraints, and other difficulties associated with time-domain approaches. Our method is furthermore general with respect to mechanical models, and we present examples of planar mechanisms augmented with elastic elements, rod networks, thin shells, and multi-material solids. We demonstrate the capabilities of our method on a set of simulation examples and real-world prototypes that include mechanical legs, compliant mechanisms, and animatronic characters.

2 RELATED WORK

This work aims at designing real-world mechanical systems that exhibit desired large-amplitude oscillations. To our knowledge, this exact problem has not been considered before, but there are close ties to several sub-fields of visual computing and engineering.

Designing Mechanical Motion. The problem of designing mechanisms that exhibit desired kinematics has received considerable attention from the visual computing community in recent years [Bächer et al. 2015; Coros et al. 2013; Thomaszewski et al. 2014; Zhang et al. 2017]. Perhaps closest to our setting is the work by Ceylan et al. [2013] who optimize for periodic motion of their characters using a frequency-space formulation. However, their method

is purely kinematic and does not consider dynamics. Megaro et al. [2017] and Takahashi et al. [2019] proposed optimization-based design tools for mechanisms that are augmented with or partly replaced by compliant elements. While these examples go beyond pure kinematics, they did not consider dynamics. As one particular application, our method can be seen as a continuation of these previous works toward the design of compliant mechanisms that exhibit dynamic, periodic motion.

Few works from the graphics community on fabrication-oriented design have considered dynamics so far. Notable exceptions include the work by Chen et al. [2017], who propose a dedicated coarsening approach for optimization-based design of flexible objects with dynamic motion and contact. Another example is the work by Bächer et al. [2014], who design the mass distribution of rigid bodies in order to obtain sustained rotational motion. However, neither of these works considers periodic motion, forced vibrations, or resonance.

The work by Hoshyari et al. [2019] on motion control for robotic characters is related to our effort in that it predicts oscillations induced by external forcing. However, whereas their goal is to suppress vibrations by optimizing actuation parameters, we aim at generating large-amplitude oscillations by optimizing for shape, mass, and material parameters.

In order to create real-world animations of deformable characters, Skouras et al. [2013] optimize for multi-material distributions and the parameters of a string-based actuation system. While we also leverage heterogeneous material distributions for motion control, our actuation principle is based on resonance induced by harmonic forcing.

Modal Subspaces for Animation. The idea of using modal subspaces for efficient animation of flexible models in graphics goes back to the early work of Pentland and Williams [1989]. Due to the inherent inefficiency of linear modal bases to model large rotational displacements, many works have proposed more efficient alternatives using, e.g., modal derivatives [Barbič and James 2005], higher-order derivative information [Hildebrandt et al. 2011], or rotation-strain coordinates [Pan et al. 2015]. Adaptive basis selection has also been considered [Hahn et al. 2014; Kim and James 2009].

Compared to the problem of constructing efficient linear bases for large-deformation simulation, nonlinear bases have not received much attention so far. An exception is the work of Fulton et al. [2019], who use machine learning techniques to construct nonlinear subspaces from simulation data. But besides the fact that none of the above approaches aim at optimization, they are all time-domain methods, meaning that they must simulate the entire evolution leading up to the steady-state behavior. By building on the Harmonic Balance Method, our approach is able to sidestep the transient process and thus directly operate on the steady-state behavior.

Audible Vibrations. Simulating vibrations is also of importance for physics-based sound synthesis. For instance, Zheng et al. [2011] and Bonneel et al. [2008] used Linear Modal Analysis (LMA) to efficiently generate sound for rigid body impact. To increase richness and fidelity, nonlinear approaches have been investigated for vibrating thin-shell structures [Chadwick et al. 2009; Cirio et al. 2018].

In the context of fabrication-oriented design, LMA has been the predominant approach so far; see, e.g., the works of Umetani et al.

[2010] and Bharaj et al. [2015] on metallophones. Another stream of work in this context has investigated the simulation [Allen and Raghuvanshi 2015] and design [Umetani et al. 2016] of wind instruments and acoustic filters [Li et al. 2016]. While sound synthesis is concerned with high-frequency, small-amplitude vibrations, we consider the design of mechanical systems with low-frequency but large-amplitude oscillations.

Nonlinear Vibrations in Engineering. The study of nonlinear vibrations is fundamental for structural dynamics, fluid structure interaction, aeroelasticity, and many other fields of engineering. As an extension of LMA beyond the linear regime, nonlinear normal modes are often used to analyze nonlinear dynamical phenomena [Kerschen et al. 2009]. One approach to compute steady-state behavior is to use time-domain integration schemes for simulating the transient process of the system until stable, periodic motion is obtained. An alternative approach that avoids computing the entire transient process are shooting methods [Peeters et al. 2009] that simulate only a single cycle while optimizing initial conditions and period to obtain periodicity.

Instead of enforcing periodicity in the time domain, we leverage the Harmonic Balance Method (HBM) to directly compute nonlinear periodic motion in frequency space. HBM is a versatile approach for nonlinear vibration analysis and has been widely used for many applications including nonlinear circuits [Bandler et al. 1992], fluid dynamics [Hall et al. 2013], and nonlinear mechanical systems in general [Detroux et al. 2014]. Its basic principle is to represent motion as a truncated Fourier series composed of periodic, trigonometric functions with different frequencies and phase offsets. Unlike the shooting method which already requires optimization for computing steady-state behavior, HBM only requires the solution of a nonlinear root-finding problem. Moreover, once the steady-state behavior for a given driving frequency has been obtained, solutions for different inputs can be computed efficiently using numerical continuation.

Besides its advantages for solving forward simulation problems, HBM holds great promise for optimization-based design automation in elasticity and aeroelasticity applications [Engels-Putzka et al. 2019]. Perhaps closest to our work, Dou and Jensen [2015] combine HBM with sensitivity analysis to suppress vibrations in a simple beam by minimizing the amplitude at resonance. While we follow a similar methodology, our goal is not to suppress but to amplify oscillations and to control large-amplitude motion. To this end, we introduce amplitude and trajectory objectives that we integrate with forward and inverse design tools based on sensitivity analysis. Together, these tools enable interactive design space exploration as well as fully-automated design optimization of compliant systems with low-frequency, large-amplitude oscillations.

3 THEORY

Our optimization-based design tool builds on the Harmonic Balance Method, a frequency-domain method for simulating the steady-state behavior of nonlinear mechanical systems. We first lay out the parts of the theory as relevant to our setting and briefly describe the corresponding computational framework.

3.1 Equations of Motion in Frequency Space

To arrive at the frequency-domain formulation, we start with the canonical equations of motion for a forced dynamical system in the time domain,

$$\mathbf{M}\ddot{\mathbf{x}} + \mathbf{D}\dot{\mathbf{x}} = \mathbf{f}_{\text{int}}(\mathbf{x}) + \mathbf{f}_{\text{ext}}(\mathbf{x}, \omega, t) , \tag{1}$$

where $\mathbf{x}, \dot{\mathbf{x}}, \ddot{\mathbf{x}} \in \mathbb{R}^{3n}$ denote nodal displacement, velocity, and acceleration, respectively. Furthermore, $\mathbf{f}_{\text{int}}(\mathbf{x})$ is the nonlinear internal force and $\mathbf{f}_{\text{ext}}(\mathbf{x}, \omega, t)$ is the periodic external force with frequency ω. Finally, $\mathbf{M}$ denotes the mass matrix, $\mathbf{D} = D_\alpha \mathbf{M} + D_\beta \mathbf{K}(\mathbf{x})$ is the Rayleigh damping matrix, and

$$\mathbf{K}(\mathbf{x}) = \frac{\partial \mathbf{f}_{\text{int}}(\mathbf{x})}{\partial \mathbf{x}}$$

is the tangential stiffness matrix. To simplify the subsequent derivations, we separate linear and nonlinear terms as

$$\mathbf{M}\ddot{\mathbf{x}} + \hat{\mathbf{D}}\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \dot{\mathbf{x}}, \omega, t) , \tag{2}$$

where $\hat{\mathbf{D}} = D_\alpha \mathbf{M}$ and nonlinear forces are summarized as

$$\mathbf{f}(\mathbf{x}, \dot{\mathbf{x}}, \omega, t) = \mathbf{f}_{\text{int}}(\mathbf{x}) + \mathbf{f}_{\text{ext}}(\mathbf{x}, \omega, t) - D_\beta \mathbf{K}(\mathbf{x})\dot{\mathbf{x}} . \tag{3}$$

Discretization. At the steady-state solution, positions and forces are periodic functions. We approximate these time-domain functions in frequency-space as finite Fourier series

$$\mathbf{x}(t) \approx \mathbf{c}_0^x + \sum_{k=1}^{N_H} \left(\mathbf{s}_k^x \sin(k\omega t) + \mathbf{c}_k^x \cos(k\omega t) \right) \tag{4}$$

$$\mathbf{f}(t) \approx \mathbf{c}_0^f + \sum_{k=1}^{N_H} \left(\mathbf{s}_k^f \sin(k\omega t) + \mathbf{c}_k^f \cos(k\omega t) \right) \tag{5}$$

truncated to the N_H-th harmonic. In the above expressions, $\mathbf{s}_k^* \in \mathbb{R}^{3n}$ and $\mathbf{c}_k^* \in \mathbb{R}^{3n}$ are the vectors of Fourier coefficients decorated with x and f superscripts for positions and forces, respectively. Velocities $\mathbf{v}(t)$ follow through direct differentiation of (4) and, hence, require no additional coefficients. We gather position and force coefficients into vectors of size $(2N_H + 1) \cdot 3n$ as

$$\mathbf{z} = [(\mathbf{c}_0^x)^T \quad (\mathbf{s}_1^x)^T \quad (\mathbf{c}_1^x)^T \quad \ldots \quad (\mathbf{s}_{N_H}^x)^T \quad (\mathbf{c}_{N_H}^x)^T]^T , \tag{6}$$

$$\mathbf{b} = [(\mathbf{c}_0^f)^T \quad (\mathbf{s}_1^f)^T \quad (\mathbf{c}_1^f)^T \quad \ldots \quad (\mathbf{s}_{N_H}^f)^T \quad (\mathbf{c}_{N_H}^f)^T]^T . \tag{7}$$

Since the force $\mathbf{f}(t)$ in (3) is a nonlinear function of position and velocity, its Fourier coefficients $\mathbf{b}$ are functions of the position coefficients $\mathbf{z}$ and we write $\mathbf{b} = \mathbf{b}(\mathbf{z}, \omega)$.

Substituting (4) and (5) into (2) and balancing the harmonic terms with a Galerkin projection (see Appendix A for details) yields the following set of nonlinear equations in the frequency domain,

$$\mathbf{h}(\mathbf{z}, \omega) \equiv \mathbf{A}(\omega)\mathbf{z} - \mathbf{b}(\mathbf{z}, \omega) = 0 , \tag{8}$$

where $\mathbf{A} = \text{diag}(\mathbf{0}, \mathbf{A}_1, \ldots, \mathbf{A}_j, \ldots, \mathbf{A}_{N_H})$ is a square, block-diagonal matrix of dimension $(2N_H + 1) \cdot 3n$ describing the linear dynamics of the system. Each of the $6n$ blocks is defined as

$$\mathbf{A}_j = \begin{bmatrix} -(j\omega)^2 \mathbf{M} & -j\omega \hat{\mathbf{D}} \\ j\omega \hat{\mathbf{D}} & -(j\omega)^2 \mathbf{M} \end{bmatrix} . \tag{9}$$

Eq. (8) is the frequency-domain version of (1), projected to a finite-dimensional Fourier subspace. Once we find the root $\mathbf{z}^*$ of

(8), we can convert it to the corresponding time-domain motion $\mathbf{x}^*$ using inverse Fourier transformation.

3.2 Evaluation of Nonlinear Forces and Derivatives

We solve the system of nonlinear equations (8) using Newton's method. In every iteration, the solver requires the evaluation of $\mathbf{b}$ and the Jacobian $\partial\mathbf{h}/\partial\mathbf{z}$ for a given driving frequency ω. Since the elastic and viscous forces are nonlinear and often non-polynomial functions of position and velocity, expressing them directly in frequency space is difficult. An alternative approach is given by the alternating frequency/time-domain (AFT) technique [Cameron and Griffin 1989]. Using discrete Fourier transforms (DFT), motion is first converted from frequency space to the time domain where nonlinear forces are then evaluated and finally converted back to the frequency domain as

$$\mathbf{z} \xrightarrow{\text{DFT}^{-1}} [\mathbf{x}(t), \dot{\mathbf{x}}(t)] \rightarrow \mathbf{f}(\mathbf{x}, \dot{\mathbf{x}}, \omega, t) \xrightarrow{\text{DFT}} \mathbf{b}(\mathbf{z}) . \quad (10)$$

Let $t_i = i\Delta t$, $i = 1...N$ denote uniformly distributed samples in the time domain with $\Delta t = \frac{2\pi}{N\omega}$ and $N \geq 2N_H + 1$. We start by evaluating (4) at the N sample points to obtain the discrete time-domain trajectory $\tilde{\mathbf{x}} \in \mathbb{R}^{N \cdot 3n}$ and corresponding sample velocities $\tilde{\mathbf{v}}$. Due to the linearity of inverse DFT, we have

$$\tilde{\mathbf{x}} = \Gamma_x \mathbf{z} , \quad \text{and} \quad \tilde{\mathbf{v}} = \Gamma_v \mathbf{z} , \quad (11)$$

where Γ_x and Γ_v are sparse linear operators of size $3n \cdot N \times (2N_H + 1) \cdot 3n$. We then compute nonlinear forces for each sample point according to the mechanical model (see Appendix E) and store the result as $\tilde{\mathbf{f}} = (\tilde{\mathbf{f}}_1^T, ..., \tilde{\mathbf{f}}_N^T)^T$ where $\tilde{\mathbf{f}}_i = \mathbf{f}(\tilde{\mathbf{x}}_i, \tilde{\mathbf{v}}_i) \in \mathbb{R}^{3n}$ is the vector of nodal forces for sample t_i. The time-domain forces are transformed back to frequency space using DFT,

$$\mathbf{b} = \Gamma_f^{-1}\tilde{\mathbf{f}} , \quad (12)$$

where Γ_f^{-1} is again a sparse linear operator; see Appendix B. With these transformations, the Jacobian of (8) is obtained as

$$\frac{\partial\mathbf{h}}{\partial\mathbf{z}} = \mathbf{A} - \frac{\partial\mathbf{b}}{\partial\tilde{\mathbf{f}}}\left(\frac{\partial\tilde{\mathbf{f}}}{\partial\tilde{\mathbf{x}}}\frac{\partial\tilde{\mathbf{x}}}{\partial\mathbf{z}} + \frac{\partial\tilde{\mathbf{f}}}{\partial\tilde{\mathbf{v}}}\frac{\partial\tilde{\mathbf{v}}}{\partial\mathbf{z}}\right) = \mathbf{A} - \Gamma_f^{-1}\left(\frac{\partial\tilde{\mathbf{f}}}{\partial\tilde{\mathbf{x}}}\Gamma_x + \frac{\partial\tilde{\mathbf{f}}}{\partial\tilde{\mathbf{v}}}\Gamma_v\right) \quad (13)$$

where $\partial\tilde{\mathbf{f}}/\partial\tilde{\mathbf{x}}$ and $\partial\tilde{\mathbf{f}}/\partial\tilde{\mathbf{v}}$ can be computed analytically from (3).

3.3 Frequency Response Curves and Continuation

With the Jacobian (13) of the governing equations in hand, we can use Newton's method to compute the solution for a given driving frequency. In order to characterize the behavior of nonlinear mechanical systems, however, it is usually necessary to compute solutions for a range of driving frequencies, leading to so-called frequency response curves that plot amplitude as a function of driving frequency; see Fig. 2 for an example. Frequency response curves summarize important characteristics of nonlinear mechanical oscillators in a compact form. In particular, they reveal the number, location, amplitude, and sharpness of resonance peaks—quantities that are of interest for analysis as well as design. We start by introducing our measure of amplitude, then proceed to the conditions that characterize resonance, and finally explain how to compute frequency response curves numerically using continuation.
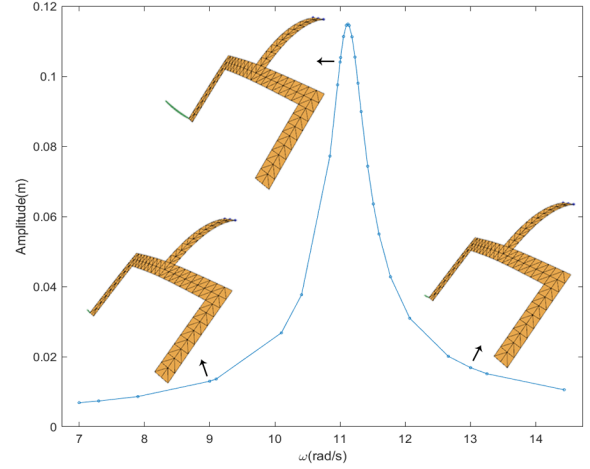
Fig. 2. Frequency response curve for a thin shell model. Amplitude is measured using the trajectory of a selected vertex shown in *green*. The inset figures illustrate maximum-deflection configurations at 9 rad/s, 11 rad/s and 13 rad/s, respectively.

Amplitude. To quantify motion magnitude, and to encourage large-amplitude oscillation during design optimization, we must first develop a general definition of amplitude. For the one-dimensional case, amplitude is defined as the maximum displacement over a period of oscillation. While seemingly simple, this concept does not readily generalize to higher dimensions and it translates into computational difficulties: maximizing displacement means solving for zeroes of the velocity function, which is a trigonometric root-finding problem with potentially many local extrema. To avoid these difficulties, we instead quantify the magnitude of motion for a given vertex $\mathbf{x}_i$ as the distance traveled within one period,

$$A_i(\mathbf{z}) = \int_0^T \|\mathbf{v}_i\| \, dt = \int_0^T \sqrt{(v_i^x)^2 + (v_i^y)^2 + (v_i^z)^2} \, dt , \quad (14)$$

where

$$v_i^j = \sum_{k=1}^{N_H} \mathbf{s}_k^{x_j} k\omega \cos(k\omega t) - \mathbf{c}_k^{x_j} k\omega \sin(k\omega t) . \quad (15)$$

While this expression could be evaluated in frequency space through quadrature, we simply compute the length of the piece-wise linear trajectory in the time domain using the AFT scheme described above.

Resonance. Having established a way of quantifying motion magnitude, we can now make precise the conditions for resonance as a local maximum of amplitude with respect to the driving frequency,

$$\omega_{\text{res}} = \arg\max_{\omega} A(\mathbf{z}(\omega)) , \quad (16)$$

where ω_{res} is the corresponding resonance frequency and $\mathbf{z}(\omega)$ are steady-state Fourier coefficients expressed as a function of the driving frequency. The map between $\mathbf{z}$ and ω is implicitly given through the dynamic equilibrium condition $\mathbf{h}(\mathbf{z}, \omega) = \mathbf{0}$. As a necessary condition for resonance, we require that

$$\frac{dA(\mathbf{z})}{d\omega} = \frac{\partial A(\mathbf{z})}{\partial\mathbf{z}}\frac{d\mathbf{z}}{d\omega} = \mathbf{0} . \quad (17)$$

Continuation. To avoid solving (8) from scratch every time when computing frequency response curves, we use numerical continuation to *follow* the path of the solution while changing the input frequency. Each continuation step consists of a prediction step $\mathbf{t}_i$ for updating the current state $[\mathbf{z}_i^T, \omega_i]^T$, followed by a projection step that enforces $\mathbf{h}(\mathbf{z}, \omega) = 0$. The prediction step requires the derivative of (8) with respect to both the state $\mathbf{z}$ and the driving frequency ω. The latter is obtained as

$$\mathbf{h}_\omega = \frac{\partial \mathbf{h}}{\partial \omega} = \frac{\partial \mathbf{A}}{\partial \omega} \mathbf{z} - \frac{\partial \mathbf{b}}{\partial \tilde{\mathbf{f}}} \frac{\partial \tilde{\mathbf{f}}}{\partial \omega} = \frac{\partial \mathbf{A}}{\partial \omega} \mathbf{z} - \Gamma_f^{-1} \frac{\partial \tilde{\mathbf{f}}}{\partial \omega} \quad (18)$$

where $\frac{\partial \tilde{\mathbf{f}}}{\partial \omega} = -\mathbf{D}_\beta \mathbf{K}(\mathbf{x}) \frac{\partial \dot{\mathbf{x}}}{\partial \omega} = -\mathbf{D}_\beta \mathbf{K}(\mathbf{x}) \frac{\Gamma_v \mathbf{z}}{\omega}$. Note that the derivative of the external force with respect to ω is zero; cf. the derivation of Γ and Γ^{-1} in Appendix B. The prediction step $\mathbf{t}_i$, tangent to the response curve at $[\mathbf{z}_i, \omega_i]^T$, is then determined as

$$\begin{bmatrix} \mathbf{h_z} & \mathbf{h}_\omega \\ \mathbf{t}_{i-1}^T \end{bmatrix} \mathbf{t}_i = \begin{bmatrix} \mathbf{0} \\ 1 \end{bmatrix} , \quad (19)$$

where $\mathbf{h_z}$ is a shorthand for (13). The first line in the above equation asks that the step should maintain force balance to first order. The second condition requires the prediction step to have a positive dot product with the previous step, thus preventing the continuation scheme from going backwards. The resulting prediction step is then used in conjunction with an arc-length control strategy [Seydel 2009] to compute an initial guess for Newton's method.

To reliably track down resonance peaks, we use condition (17) and monitor the sign of the gradient $\frac{dA}{d\omega}$. Whenever the sign changes from negative to positive between two samples, we compute the exact resonance frequency ω_res by solving (17). The Jacobian matrix $\frac{d\mathbf{z}}{d\omega}$ required for this procedure can be computed through sensitivity analysis on $\mathbf{h}(\mathbf{z}, \omega) = \mathbf{0}$.

4 COMPUTATIONAL DESIGN

Given an initial design, we would like to determine changes for parameters such as shape, mass distribution, and driving frequency such that the resulting steady-state motion best approximates a given target behavior. To this end, we consider two design approaches: user-driven forward exploration of the design space and optimization-driven inverse design.

4.1 Dynamical Equilibrium and Sensitivity

For both forward and inverse design, we must be able to predict the change in steady-state behavior induced by a given change in design parameters. To this end, we leverage the equilibrium constraints as an implicit map between parameters and state: the Fourier coefficients $\mathbf{z}$ must be a dynamic equilibrium configuration for the design parameters $\mathbf{p}$. We make this explicit by rewriting (8) as

$$\mathbf{h}(\mathbf{z}, \mathbf{p}, \omega) = \mathbf{A}(\mathbf{p}, \omega)\mathbf{z} - \mathbf{b}(\mathbf{p}, \mathbf{z}, \omega) = \mathbf{0} . \quad (20)$$

Since this relation must hold for every admissible choice of $\mathbf{p}$, the Fourier coefficients effectively become a function of the design parameters, i.e., $\mathbf{z} = \mathbf{z}(\mathbf{p})$. Moreover, any change to the parameters will entail a corresponding state change such that the system is again at equilibrium. More formally, we have

$$\frac{d\mathbf{h}}{d\mathbf{p}} = \frac{\partial \mathbf{h}}{\partial \mathbf{z}} \frac{d\mathbf{z}}{d\mathbf{p}} + \frac{\partial \mathbf{h}}{\partial \mathbf{p}} = \mathbf{0} , \quad (21)$$

from which we obtain the so-called design sensitivity matrix as

$$\mathbf{S} = \frac{d\mathbf{z}}{d\mathbf{p}} = -\frac{\partial \mathbf{h}}{\partial \mathbf{z}}^{-1} \frac{\partial \mathbf{h}}{\partial \mathbf{p}} . \quad (22)$$

The above equations form the basis of the forward and inverse design tools that we describe next.

Rank of Constraint Jacobian. Eq. (20) provides exactly as many constraints as there are degrees of freedom. In order for the constraint Jacobian in (22) to be invertible, the constraint gradients must be linearly independent. Even if the Jacobian is non-singular, numerical problems can still arise if the matrix is indefinite. While the conditions on rank and definiteness cannot be guaranteed for all configurations of nonlinear oscillators, we are only interested in stable steady-state solutions, which fulfill these properties by definition. If transient rank-deficiency or indefiniteness are still encountered (manifesting through failure of the LU solver or a large residual of the linear system), we apply adaptive diagonal regularization until a valid solution is found.

4.2 Forward Sensitivity Exploration

The sensitivity matrix provides an efficient tool for interactive exploration of the design space. For a given initial design $\mathbf{p}$ and corresponding state $\mathbf{z}$, we compute a first-order prediction for the new equilibrium state as

$$\mathbf{z}_p = \mathbf{z} + \frac{d\mathbf{z}}{d\mathbf{p}} \Delta \mathbf{p} . \quad (23)$$

Once the sensitivity matrix is computed, this prediction is instantaneous and thus enables interactive exploration of the design space around a given set of parameters. For larger parameter changes, however, the first-order prediction can become inaccurate, requiring a full, nonlinear update. In our interface, the user can manually issue such update commands, which entail re-simulation with the new parameters and re-computation of the sensitivity matrix.

Besides changing parameters individually, we found it useful to provide additional compound variables that change multiple parameters simultaneously along the gradient of selected design objectives—see Sec. 5.3 for an example.

4.3 Inverse Design

Even when aided by sensitivity information, forward design based on manual exploration soon becomes unattractive as the number of parameters increases. If design goals can be quantified, automatic parameter optimization can be a very efficient and effective alternative. With the implicit relation between $\mathbf{z}$ and $\mathbf{p}$ defined as above, we can cast design optimization as an unconstrained minimization problem, where we aim to minimize an objective function $f(\mathbf{z}(\mathbf{p}), \mathbf{p})$ that encodes various design goals as described below. Using Eq. (21), we obtain the objective gradient as

$$\frac{df}{d\mathbf{p}} = \frac{d\mathbf{z}}{d\mathbf{p}}^T \frac{\partial f}{\partial \mathbf{z}} + \frac{\partial f}{\partial \mathbf{p}} = -\frac{\partial \mathbf{h}}{\partial \mathbf{p}}^T \frac{\partial \mathbf{h}}{\partial \mathbf{z}}^{-T} \frac{\partial f}{\partial \mathbf{z}} + \frac{\partial f}{\partial \mathbf{p}} . \quad (24)$$

It is worth noting that this expression can be rearranged such that only a single linear system needs to be solved. We use this gradient in combination with L-BFGS-B [Byrd et al. 1995] to find parameters that minimize the objective function while enforcing bound constraints where applicable, e.g., positivity for mass values.

As we show in Sec. 5, HBM-based simulation and frequency-space sensitivity analysis for optimization combine into an efficient tool for inverse design. In order for this tool to be effective, however, we must define objective functions that allow users to express their design intents.

Trajectory Matching. An intuitive and direct way of specifying motion goals is by prescribing target trajectories that selected nodes should track. Given a time-domain target trajectory for a given vertex k as input, we first compute the corresponding Fourier coefficients $\hat{z} = (\hat{s}, \hat{c})$ using DFT. We then measure the distance between current and target trajectories in frequency space as

$$f_{\text{Dist}}(z) = \sum_{i=1}^{N_H} \sum_{j=1}^{3} \left((s_i^{3k+j} - \hat{s}_i^j)^2 + (c_i^{3k+j} - \hat{c}_i^j)^2 \right), \quad (25)$$

where the superscript selects individual components from the coefficient vectors. The objective in its above form assumes that both target and actual trajectory have the same phase offset, but this is generally not the case. To eliminate phase dependence, we evaluate (25) for different phase offsets and define the final objective value as the smooth minimum over the individual distances, i.e.,

$$f_{\text{Track}}(z) = \frac{\sum_{i=0}^{N-1} f_{\text{Dist}}(z, \phi_i) e^{-\alpha f_{\text{Dist}}(z, \phi_i)}}{\sum_{i=0}^{N-1} e^{-\alpha f_{\text{Dist}}(z, \phi_i)}}, \quad (26)$$

where α is a parameter controlling the sharpness of the smooth minimum function, $\phi_i = i \frac{2\pi}{N}$ are phase offsets, and $f_{\text{Dist}}(z, \phi)$ is the same as in Eq. (25) but with the Fourier coefficients of the target trajectory shifted by a phase offset ϕ.

Amplitude. Although the trajectory matching objective affords some degree of amplitude control, we have seen in our experiments that it can be difficult to obtain large amplitudes in this way—arguably because the tracking objective is biased towards a specific motion, rather than more general, large-amplitude oscillations. We therefore introduce an objective that explicitly aims to maximize the magnitude of motion for user-selected nodes. Based on the amplitude definition in Sec. 3.3, we define an amplitude objective for vertex k as

$$f_{\text{Ampl}}(z) = (A_k(z) - \bar{A}_k)^2 \quad (27)$$

where $\bar{A}_k$ is the target amplitude. Note that, besides optimizing for specific, reachable values for the amplitude, we can also simply encourage amplitude maximization by setting $\bar{A}$ to an arbitrarily large value. Likewise, with a straightforward modification of (14), it is possible to optimize amplitude only along selected dimensions.

5 RESULTS
We evaluate our method on a set of examples that highlight the impact of our design objectives both in simulation and on actual physical prototypes. We start by analyzing the accuracy of HBM
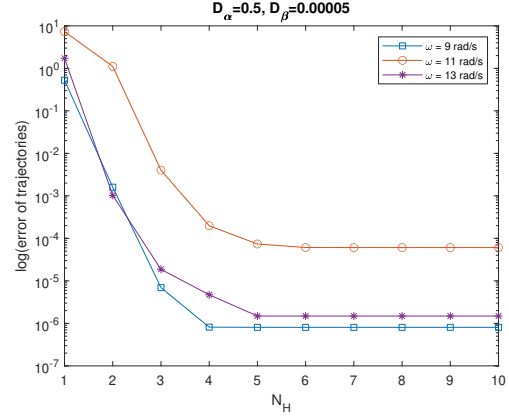
Fig. 3. Trajectory error for HBM compared to ground truth time-domain simulations. The plot shows error as a function of the number of harmonics using damping coefficients as indicated.

compared to ground truth time-domain simulations as well as a linear frequency-space approach.

5.1 Analysis & Validation
Comparison with Newmark. One of the central advantages of HBM is that it directly yields periodic steady-state solutions without having to simulate the transient process of the system. In order to analyze the accuracy of HBM in our setting, we compare to ground truth simulations obtained using the Newmark integration scheme, a time-domain method that is widely used for structural dynamics and vibration analysis in general; see Appendix C for details.

To this end, we consider a fork-shaped thin shell oscillator (188 elements, 144 nodes) with one end driven by a periodic force while the other two extremities are vibrating freely as shown in Fig. 2. To measure the difference in steady state solution for Newmark and HBM, we must first define an appropriate convergence criterion for the time-domain method. We measure the difference in trajectories between two successive periods as

$$e_p = \sum_{i=0}^{N_{NM}-1} \|x(t_{pN_{NM}+i}) - x(t_{(p+1)N_{NM}+i})\|, \quad (28)$$

where the subscripts of t_* refer to time-step indices, p denotes the index of the period, and $N_{NM} = \frac{2\pi}{\omega \Delta t}$ is the number of time steps used to simulate a single period.

Using this error metric, Fig. 3 shows accuracy plots for different driving frequencies and different numbers of harmonics. We use $N_{NM} = 4096$ for computing the step size and run the Newmark simulation until the trajectory difference between two successive periods satisfies $e_p < 1e^{-7}$. It can be seen that the error is larger for driving frequencies near resonance, which is at $\omega_{\text{res}} = 11.0$ rad/s for this example (see Fig. 2). This observation is explained by the fact that near-resonance frequencies lead to larger amplitude motion. Nevertheless, using $N_H = 5$ harmonics leads to sufficiently good accuracy, and more terms yield virtually no improvements. Additional analysis is given in Appendix D.
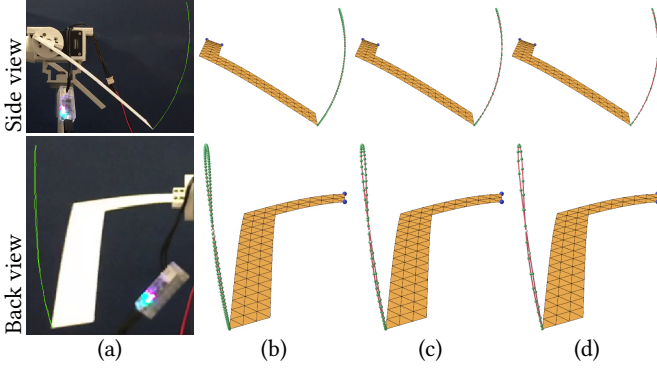
Fig. 4. Trajectory comparison between fabricated L-wing and HBM simulations at driving frequency of $2.0Hz$ with different numbers of sampling points N_{AFT}: physical prototype (a) and HBM simulation using (b) $N_{AFT} = 128$, (c) $N_{AFT} = 64$, and (d) $N_{AFT} = 32$.
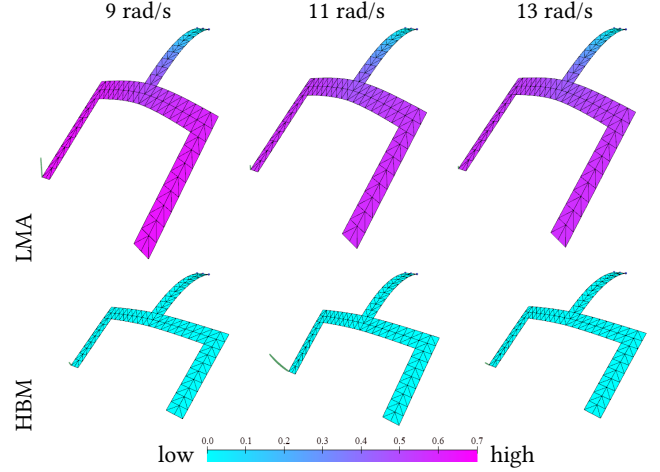

Fig. 5. Frequency responses computed with LMA and HBM for different driving frequencies with damping coefficients $D_\alpha = 0.5$ and $D_\beta = 0.00005$. Trajectory of a selected tip point (green) and color-coded maximum in-plane stretch. For LMA, the maximum strains over one period for the three driving frequencies are 0.655, 0.582, and 0.575. The corresponding values for HBM ($8.25e^{-5}$, $1.00e^{-4}$, and $8.50e^{-5}$) are 4-5 orders of magnitude smaller.

We furthermore conducted an experiment to analyze the impact of the number of samples used to evaluate the nonlinear forces in the time-domain using Eq. (3) for HBM. While a lower bound is given by the Nyquist limit, as can be seen from Fig. 4, the effect of using more samples is almost imperceptible. This visual impression is confirmed by an additional quantitative analysis, showing that the difference in Fourier coefficients is less than $1e^{-10}$ in this case.

To summarize our analysis of HBM and comparison with Newmark, we can conclude that, already with a small number of harmonics and time-domain samples, HBM accurately captures the nonlinear large-amplitude oscillation behavior that is the focus of this work. We note that, when using $N_H = 5$ and $N_{AFT} = 64$ for this example, HBM computes steady-state solutions one to two orders of magnitude faster than time-domain methods. For the general case, determining the *optimal* number of harmonics a priori is difficult. However, we found the following strategy to work well in practice: we first simulate using a small number of harmonics (e.g. $N_H = 3$), which we increase until the trajectory error between two successive runs falls below a given threshold value. While this threshold needs to be set by the user, setting it to a small fraction of the trajectory length (e.g. $1e^{-3}$) simplifies this task.

Before we present examples obtained using our optimization-based design method, we briefly comment on an alternative frequency-domain method.

Comparison with Linear Modal Analysis. In order to illustrate the importance of incorporating nonlinearities in the simulation model, we compare to Linear Modal Analysis (LMA), a frequency-space approach based on a linearization of modal dynamics around the rest state. As can be seen in Fig. 5, this first-order approximation leads to significant in-plane distortions. This is not surprising, as methods based on linearized deformation measures are known to introduce artifacts for rotational displacements. Perhaps more severe, however, is the fact that neither the resonance frequency nor the motion at resonance are captured with acceptable accuracy. These shortcomings effectively disqualify linear modal analysis as a basis for design and optimization in the large-amplitude setting.
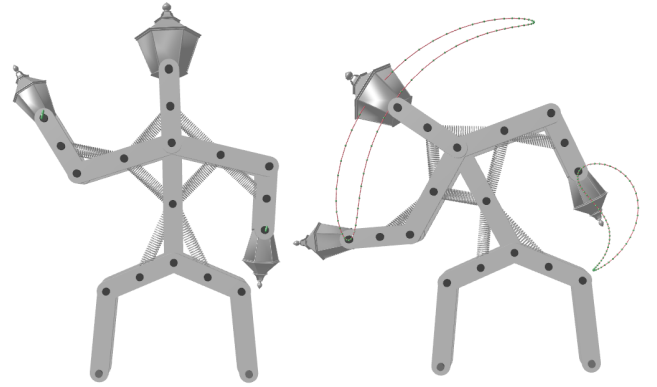

Fig. 6. Forward Design with Sensitivity Exploration illustrated on an animatronic character. The initial design (*left*) exhibits only small oscillations at the hands (indicated in *green*). After several steps of forward exploration, the final design exhibits an expressive large-amplitude motion (*right*).

5.2 Measuring Damping Parameters

Damping parameters play an important role for the dynamics of a mechanical system. To experimentally estimate damping coefficients for a given design task, we first choose a simple real-world specimen, e.g., the L-wing (see Fig. 4) for the thin shell model and the three-link mechanism (see Fig. 8) for compliant mechanisms. We then determine damping parameters such that the simulated steady-state motion is as close as possible to the corresponding real-world motion. Real-world trajectories are captured using off-the-shelf cameras for side and back views. We then extract trajectories for selected key points (such as wing tips) and use them to fit damping coefficients for simulation.
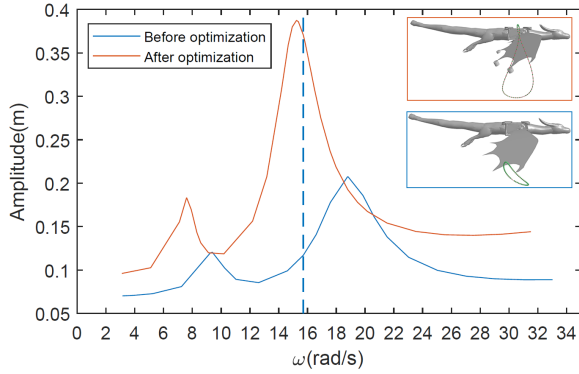
Fig. 7. Amplitude of the wing tip before (*blue*) and after (*orange*) optimization for the dragon example. The dashed line indicates the driving frequency (2.5*Hz*) used during optimization.

5.3 Forward Design with Sensitivity Exploration

We demonstrate the sensitivity-based forward design approach described in Sec. 4.2 on the mechanical character shown in Fig. 6. This character consists of 10 links connected through 6 joints and 8 springs. During interactive design exploration, the goal for the user is to obtain an understanding of the design space and to discover a large-amplitude motion that makes for an appealing animation. We drive the character by applying harmonic forcing to its feet, which we initially choose to act in the vertical direction and in-phase. The design parameters for this exploration are extra masses for each node as well as the phase offset and amplitude for the forcing. In addition to changing each parameter individually, we add controls to the interface that change parameters simultaneously along the gradient of the amplitude objective for selected nodes. This enables convenient exploration along parameter-space directions that lead to large motion amplification.

Having computed the initial frequency-response curve and the sensitivity matrix, the user starts exploring motion variations by changing design parameters. Our system then provides instant feedback on the predicted change in motion. As best seen in the accompanying video, this approach allows the user to quickly converge towards a large-amplitude motion. Re-simulation with the new parameters exhibits only little deviation from the first-order prediction shown during sensitivity exploration.

5.4 Optimization-Based Inverse Design

We demonstrate our optimization-driven approach on a set of examples that illustrate applications to a diverse range of mechanical models and highlight the impact of our design objectives. For validation, we manufacture physical prototypes for several examples and evaluate their performance.

Dragon Wings. Our first example is a dragon model obtained from a shape repository[1] for which we aim to create large amplitude oscillations for the wings such as to suggest flapping flight. The body of the dragon is kept static while its wings are driven with
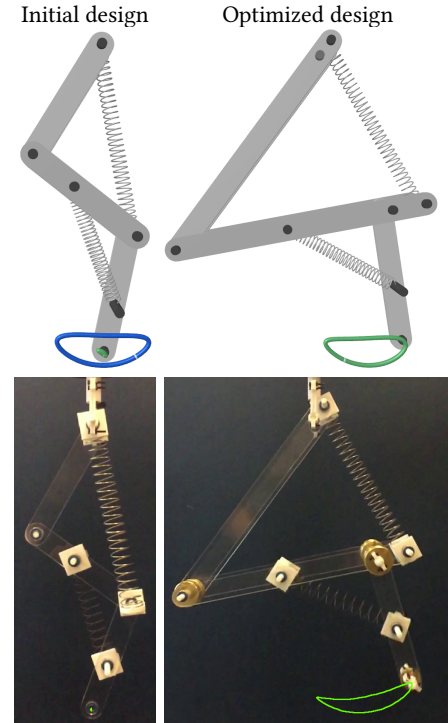
---

[1]https://www.thingiverse.com/thing:2714125

Initial design   Optimized design

Fig. 8. **Trajectory matching for the three-link compliant mechanism.** Initial design (*left*) and optimized design (*right*) with simulated (*top*) and real-world (*bottom*) end-effector trajectories shown in green. The target trajectory is shown in *blue*.

a servomotor that creates rotational motion with programmable frequency and an amplitude of 40 degrees; see Fig. 1. We model the wings using discrete shells [Grinspun et al. 2003] and fit elastic material parameters as described in [Pabst et al. 2008] to match the behavior of an FDM-printed PLA cantilever, from which we obtain a Young's modulus of 3.3*Gpa* and a Poisson's ratio of 0.36. With the elasticity coefficients determined, we experimentally set viscous parameters such as to minimize discrepancy between simulated and real-world motion using the setup shown in Fig. 4. We use as design parameters the masses of three points distributed along the trailing edge of the wing.

We start by computing the frequency response curve for the initial design, which shows only small-amplitude oscillations throughout the range of $0.5 - 5.0Hz$ (blue curve in Fig. 7). We then employ our amplitude objective to determine design parameters that lead to motion amplification at 2.5*Hz*. With an almost four-fold increase in amplitude, the optimized design exhibits greatly improved performance in simulation (orange curve in Fig. 7). This prediction is confirmed by the manufactured prototypes for the two designs, both of which show very good agreement with the corresponding HBM simulations. It should be noted that, rather than just amplifying the initial motion, the optimized trajectory at the wing tip is quite different from the original one. Whereas manually finding a sufficiently close target trajectory is a difficult task for the user in
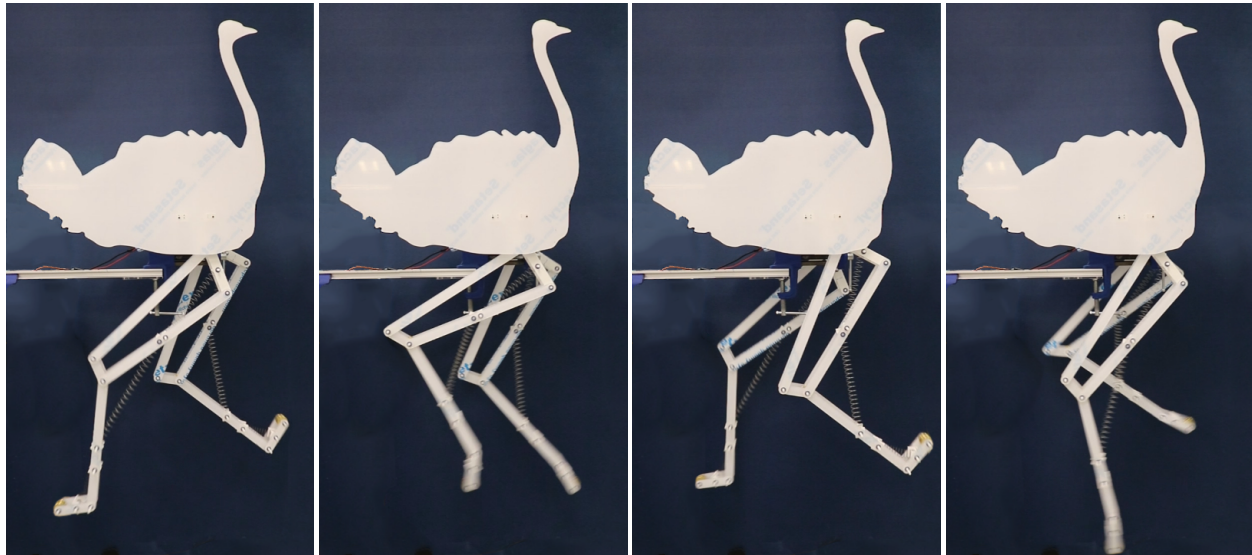
Fig. 9. **Running Ostrich.** Four images from a running sequence of our ostrich model with two legs driven at $1.2Hz$ with a phase offset of half a period.
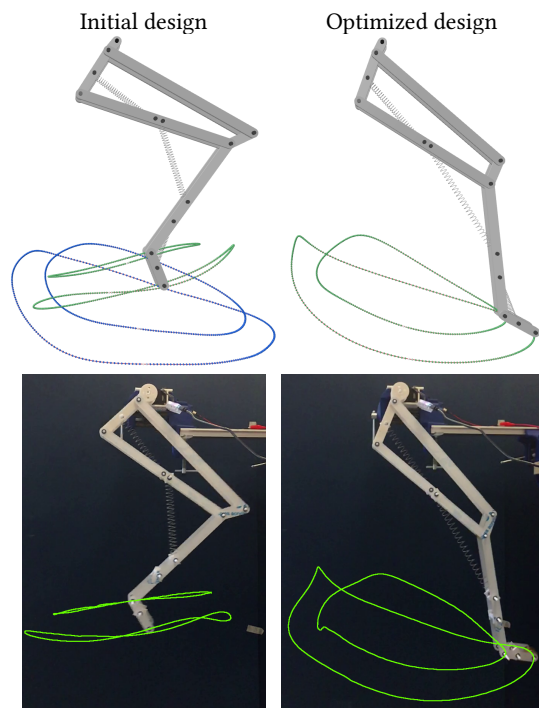

Initial design   Optimized design

Fig. 10. **Ostrich leg.** Performance of initial (*left*) design and optimized design (*right*) in simulation (*top*) and on the physical prototype (*bottom*). Target and actual trajectories are shown in *blue* and *green*, respectively.

this case, our amplitude objective provides the freedom needed to automatically discover this large-amplitude motion.

Compliant Three-Link Mechanism. The ability to control large-amplitude oscillations for nonlinear mechanical systems enables

new, efficient designs for robotics applications. We investigate the potential of this approach on two compliant mechanisms. The first design, shown in Fig. 8, is a simple three-segment chain augmented with two elastic springs. We use custom-made springs with stiffness coefficients of $70.2N/m$ and $16.0N/m$, respectively, and corresponding rest lengths of $0.146m$ and $0.088m$. We drive the hip joint with harmonic forcing in the vertical direction with an amplitude of $1mm$ and a frequency of $3Hz$ such as to mimic the footfall frequency of a fast quadrupedal walking gait [Moro et al. 2013]. To encourage walking-like motion, we prescribe a corresponding target trajectory for the bottom joint and optimize over joint masses, spring attachment points, and link lengths. As can be seen from Fig. 8 (*left*), the initial design shows little response. After optimization, however, the mechanism closely tracks the desired trajectory, thus converting a simple vertical input signal into a complex two-dimensional output motion. It is worth noting that the design changes found by our method are quite significant and leverage all available parameters.

Compliant Ostrich Leg. In our second mechanism example we consider a complex ostrich leg inspired by the work of Cotton et al. [2012]. Our leg design consists of 5 bars connected through 5 revolute joints and 3 custom-made springs with stiffnesses of $41.86N/m$, $70.2N/m$, and $16.0N/m$ and corresponding rest lengths of $0.108m$, $0.146m$, and $0.088m$. The leg is driven by a servomotor located at the hip joint that induces harmonic rotational oscillation in the attached link with a frequency of $1.2Hz$ and amplitude of 40 degrees. To generate large-amplitude oscillations that approximate the characteristic running motion observed in ostriches, we prescribe target trajectories for the toe and ankle joints that we manually extracted from real-world video footage.

As can be seen from Fig. 10 (*left*), the initial design produces trajectories that correspond to simple, reciprocating motion along a one-dimensional curve. After optimization, however, the motion closely tracks the target trajectories for both ankle and toe (see Fig.
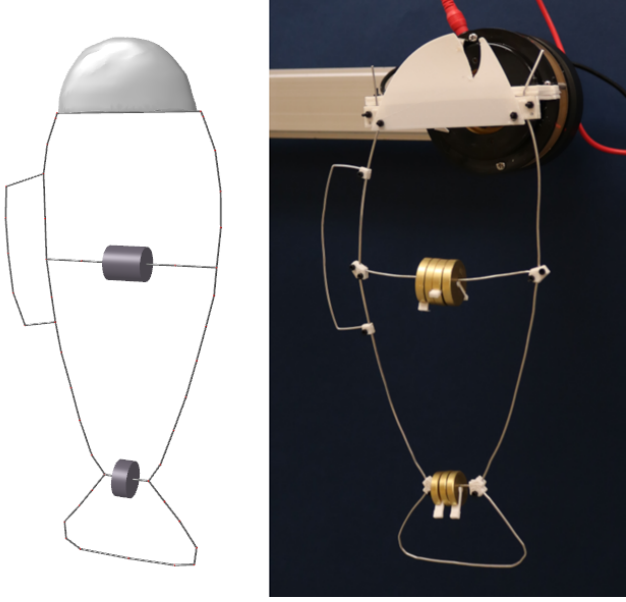
Fig. 11. For this animatronic wire character, we optimize the weights of three additional masses such as to achieve large-amplitude oscillation of its tail.
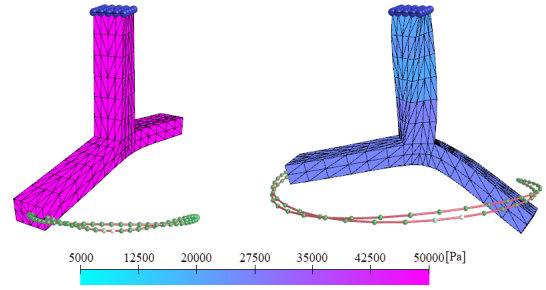

Fig. 12. We optimize per-layer material stiffness for this solid such as to maximize the amplitude of the selected vertex when driving the top face with harmonic rotational excitation at $2.0Hz$.

10 (*right*)). We optimize the mass for every joint to match the target trajectory during optimization. As best seen in the accompanying video, the optimized design also successfully reproduces the rotation of the toe between swing and stance phases that characterizes the real-world gait. We use the optimized design to build the physical ostrich model shown in Fig. 9, whose two legs are driven at a phase offset such as to mimic running motion.

Animatronic Wire Character. The HBM formulation extends to a large range of mechanical models and materials. Our optimization-driven approach is able to leverage this flexibility, which we demonstrate on two additional examples that use multi-material solids and elastic rods. Taking a result from Xu et al. [2018] as inspiration, we design a wire character in the form of a fish as illustrated in Fig. 11. We model this character using discrete elastic rods [Bergou et al. 2008] and the extension to networks described by Zehnder et al. [2016]. In order to control the frequency response of the character, we add three extra weights to the model whose mass we optimize such as to maximize the amplitude at the tail for a driving frequency of $2.0Hz$. For the physical prototype, we use standard aluminum wire with a diameter of $1.1mm$, a Young's modulus of $69GPa$, and density of $2.7g/cm^3$. We use brass weights customized according to the solution returned by the optimizer. Interestingly, the optimization completely removed the weight attached to the tip of the tail. To verify this somewhat counter-intuitive result, we experimented with manually-designed mass distributions with roughly the same total weight. As can be seen in the accompanying video, none of these alternatives is able to amplify the input motion, whereas the optimized design produces large-amplitude oscillation as predicted in simulation. We conclude that, even for seemingly simple cases,

manually finding parameters that lead to large-amplitude oscillations at the designated driving frequency can be very challenging. Our optimization-based approach removes this burden from the user.

Multi-material Solid. We investigate applications of our approach to material optimization for viscoelastic solids undergoing nonlinear vibrations. Our setup consists of an inverted T-shaped model shown in Fig. 12, whose upper face we drive through harmonic rotational excitation. The solid is structured vertically into 7 layers of homogeneous material, each of which can have different viscoelastic properties. For the inverse design problem, we aim to optimize the Young's modulus for each layer such that the amplitude of a selected node on the bottom extremity is maximized under a given driving frequency of $2.0Hz$. The per-layer material assignment found by the optimization successfully amplifies the motion by a factor of more than 4.

Eiffel Tower. Inspired by the work of [Skouras et al. 2013], we consider a material optimization problem for the Eiffel tower model show in Fig. 13. We actuate the base of the model using harmonic driving in horizontal direction. We use 2D constant strain triangle elements with an St.Venant-Kirchhoff material for simulation, and optimize for per-element stiffness coefficients such as to maximize the amplitude of the top of the tower at $2.0Hz$. The optimized design exhibits an increase in amplitude by a factor of more than 50.

5.5 Statistics & Additional Validation

Performance & Statistics. All examples run on a machine with an Intel Core i9-7900X 3.3GHz processor and 32 GB of RAM. Statistics are given in Tab. 1.

Constraint Violations. For the simulation of compliant mechanisms, we use stiff penalty terms to enforce angular and distance constraints for rigid joints and links. To analyze the validity of this approach, we monitored constraint violations, i.e., the change in angles of rigid joints and link lengths. We plot the corresponding maximum values as a function of time in Fig. 14, from which it can be seen that constraint violations are small for all three mechanism examples.

As another potential concern known from time-domain methods, stiff penalty terms can give rise to numerical damping when using
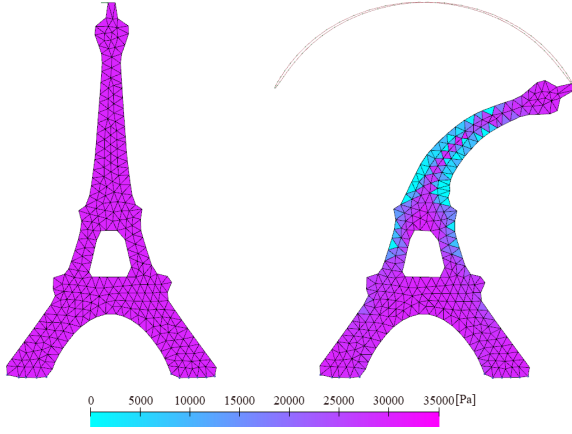
Fig. 13. Material optimization on an Eiffel Tower model. For the initial design with homogeneous material (*left*), the amplitude at the tip is almost the same as for the driving signal. After optimizing for per-element stiffness values, the tip amplitude is substantially increased (*right*).

Table 1. **Statistics for inverse design examples.** The columns list numbers of degrees of freedom (N_{DoF}), harmonics (N_H), sampling points (N_{AFT}), parameters (N_p), iterations required for convergence (N_{it}), as well as the total time spent on optimization.

Example	N_{DoF}	N_H	N_{AFT}	N_p	N_{it}	time [s]
Dragon	780	8	64	3	12	1098.09
Three-Link Leg	8	8	64	16	146	34.89
Ostrich Leg	14	14	64	7	128	163.98
Fish	187	8	64	3	46	456.96
3D Solid	918	8	64	7	10	7926.43
Eiffel Tower	658	8	64	525	150	5534.82

lower-order implicit integration methods. While we cannot guarantee that our HBM simulations are free from this effect, they closely track the Newmark solutions which are known to exhibit very little numerical dissipation.

Feasible Regions of Design Space. Understanding and navigating the space of feasible dynamics for a given input model is crucial for successful design. Our inverse design tool can be used to answer the question whether a given motion is achievable, and one positive answer is often enough to fulfill the user's intent. However, if the desired motion is infeasible, our method will return a design whose motion is, at least locally, as close as possible to the target behavior. If deviations are substantial, however, the solution computed in this way might not be subjectively optimal or even acceptable for the user. In such cases, finding good compromises requires further design space exploration. While our method does not offer an explicit representation of the space of feasible motions, if model complexity allows for it, our interactive tool provides an efficient way to explore the possibilities and limitations of a given input model.
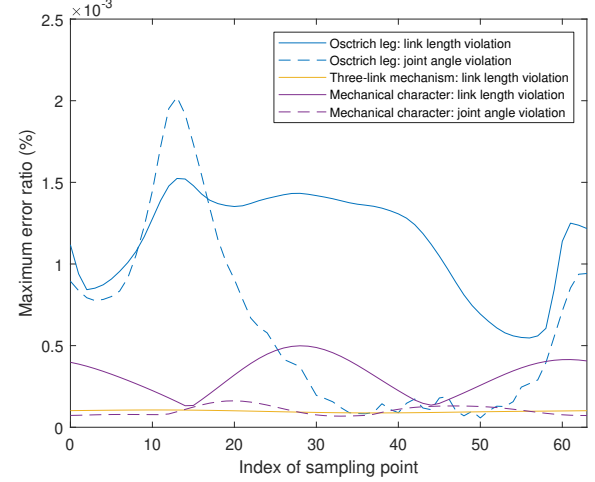

Fig. 14. **Maximum error of constraints violation.** We use a penalty stiffness of $1e^7$ and $N_{AFT} = 64$ time-domain samples for these three compliant mechanism examples. Each curve shows the maximum error (change in angle/length divided by corresponding original value for angular and distance constraints) over all constraints for each example within a period. It can be seen that constraint violations remain below $2e^{-3}\%$ at all times for all cases.

6 LIMITATIONS & FUTURE WORK

We presented an optimization-driven frequency-space approach for designing mechanical systems that exhibit desired nonlinear oscillations. Our results indicate that Harmonic Balance paired with Sensitivity Analysis is indeed an efficient and effective combination that enables the construction of powerful forward and inverse design tools.

There are several limitations of our method that we briefly discuss below along with other potential directions for future research. If constraints are present in the mechanical system, the truncated Fourier series will generally not satisfy them exactly. This would be the case, e.g., when modeling mechanisms as articulated multibody systems. Nevertheless, to satisfy the requirements of a specific application, constraint violations can be made arbitrarily small by using a sufficiently large number of harmonics.

Using our amplitude objective, we found it unnecessary to explicitly enforce resonance in order to generate large-amplitude motion—and the designs that our method discovered nevertheless proved to be at or close to resonance peaks. Other applications, however, require explicit control over resonance peaks and it would be interesting to extend our formulation in this direction.

We use relatively simple constitutive models for both elastic and viscous material behavior. While our damping model does not explicitly account for air drag, the mass contribution in the Rayleigh damping model emulates this effect to some extent. This choice is justified for problems in which air drag is insignificant, but other applications might require more accurate models.

In our examples, we have not tried to generate periodic motions that include contact or friction. However, HBM can be extended to handle these effects [Krack et al. 2017] and it would be worthwhile exploring their integration in our optimization-based design

approach. There are many other aspects of nonlinear vibrations that we deliberately chose to ignore in this work, including bifurcations, period doubling, and internal resonance. Nevertheless, incorporating these phenomena would increase the range of mechanical systems that can be designed with our approach.

The computational burden of our method depends on the complexity of the model, since each mesh vertex is endowed with $3 \cdot (2N_H + 1)$ Fourier coefficients. While model complexity might necessitate a large number of vertices, the low-frequency oscillations that we aim at are typically confined to a low-dimensional, albeit nonlinear, subspace. Extending our method towards nonlinear subspaces of frequency-space is a promising direction for future research.

We have shown applications of our approach to mechanical leg designs. To be useful for robotics applications, however, the weight carried by the legs (robot body and additional payload) must be accounted for during design. Another interesting direction would be to simultaneously optimize for different driving frequencies such as to adapt leg motion according to running speed.

ACKNOWLEDGMENTS

REFERENCES

Andrew Allen and Nikunj Raghuvanshi. 2015. Aerophones in Flatland: Interactive Wave Simulation of Wind Instruments. *ACM Trans. Graph.* 34, 4, Article Article 134 (July 2015), 11 pages. https://doi.org/10.1145/2767001

Moritz Bächer, Stelian Coros, and Bernhard Thomaszewski. 2015. LinkEdit: Interactive Linkage Editing Using Symbolic Kinematics. *ACM Trans. Graph.* 34, 4, Article Article 99 (July 2015), 8 pages. https://doi.org/10.1145/2766985

Moritz Bächer, Emily Whiting, Bernd Bickel, and Olga Sorkine-Hornung. 2014. Spin-It: Optimizing Moment of Inertia for Spinnable Objects. *ACM Trans. Graph.* 33, 4, Article Article 96 (July 2014), 10 pages. https://doi.org/10.1145/2601097.2601157

J. W. Bandler, R. M. Biernacki, and S. H. Chen. 1992. Harmonic balance simulation and optimization of nonlinear circuits. In *[Proceedings] 1992 IEEE International Symposium on Circuits and Systems*, Vol. 1. 85–88 vol.1. https://doi.org/10.1109/ISCAS.1992.230008

Jernej Barbič and Doug L. James. 2005. Real-Time Subspace Integration for St. Venant-Kirchhoff Deformable Models. *ACM Trans. Graph.* 24, 3 (July 2005), 982–990. https://doi.org/10.1145/1073204.1073300

Miklós Bergou, Max Wardetzky, Stephen Robinson, Basile Audoly, and Eitan Grinspun. 2008. Discrete Elastic Rods. In *ACM SIGGRAPH 2008 Papers (SIGGRAPH '08)*. Association for Computing Machinery, New York, NY, USA, Article 63, 12 pages. https://doi.org/10.1145/1399504.1360662

Gaurav Bharaj, David I. W. Levin, James Tompkin, Yun Fei, Hanspeter Pfister, Wojciech Matusik, and Changxi Zheng. 2015. Computational Design of Metallophone Contact Sounds. *ACM Trans. Graph.* 34, 6, Article Article 223 (Oct. 2015), 13 pages. https://doi.org/10.1145/2816795.2818108

Nicolas Bonneel, George Drettakis, Nicolas Tsingos, Isabelle Viaud-Delmon, and Doug James. 2008. Fast Modal Sounds with Scalable Frequency-Domain Synthesis. *ACM Trans. Graph.* 27, 3 (Aug. 2008), 1–9. https://doi.org/10.1145/1360612.1360623

Richard H Byrd, Peihuang Lu, Jorge Nocedal, and Ciyou Zhu. 1995. A limited memory algorithm for bound constrained optimization. *SIAM Journal on scientific computing* 16, 5 (1995), 1190–1208.

TM Cameron and JH Griffin. 1989. An alternating frequency/time domain method for calculating the steady-state response of nonlinear dynamic systems. *Journal of applied mechanics* 56, 1 (1989), 149–154.

Duygu Ceylan, Wilmot Li, Niloy J. Mitra, Maneesh Agrawala, and Mark Pauly. 2013. Designing and Fabricating Mechanical Automata from Mocap Sequences. *ACM Trans. Graph.* 32, 6, Article Article 186 (Nov. 2013), 11 pages. https://doi.org/10.1145/2508363.2508400

Jeffrey N Chadwick, Steven S An, and Doug L James. 2009. Harmonic shells: a practical nonlinear sound model for near-rigid thin shells. *ACM Trans. Graph.* 28, 5 (2009), 119–1.

Desai Chen, David I. W. Levin, Wojciech Matusik, and Danny M. Kaufman. 2017. Dynamics-Aware Numerical Coarsening for Fabrication Design. *ACM Trans. Graph.* 36, 4, Article Article 84 (July 2017), 15 pages. https://doi.org/10.1145/3072959.3073669

Gabriel Cirio, Ante Qu, George Drettakis, Eitan Grinspun, and Changxi Zheng. 2018. Multi-scale simulation of nonlinear thin-shell sound with wave turbulence. *ACM Transactions on Graphics (TOG)* 37, 4 (2018), 110.

Stelian Coros, Bernhard Thomaszewski, Gioacchino Noris, Shinjiro Sueda, Moira Forberg, Robert W. Sumner, Wojciech Matusik, and Bernd Bickel. 2013. Computational Design of Mechanical Characters. *ACM Trans. Graph.* 32, 4, Article Article 83 (July 2013), 12 pages. https://doi.org/10.1145/2461912.2461953

S. Cotton, I. M. C. Olaru, M. Bellman, T. van der Ven, J. Godowski, and J. Pratt. 2012. FastRunner: A fast, efficient and robust bipedal robot. Concept and planar simulation. In *2012 IEEE International Conference on Robotics and Automation*. 2358–2364.

T. Detroux, L. Renson, and G. Kerschen. 2014. The Harmonic Balance Method for Advanced Analysis and Design of Nonlinear Mechanical Systems. In *Nonlinear Dynamics, Volume 2*, Gaetan Kerschen (Ed.). Springer International Publishing, Cham, 19–34.

Suguang Dou and Jakob Søndergaard Jensen. 2015. Optimization of nonlinear structural resonance using the incremental harmonic balance method. *Journal of Sound and Vibration* 334 (2015), 239–254. https://doi.org/10.1016/j.jsv.2014.08.023

Anna Engels-Putzka, Jan Backhaus, and Christian Frey. 2019. Forced Response Sensitivity Analysis Using an Adjoint Harmonic Balance Solver. *Journal of Turbomachinery* 141, 3 (01 2019). https://doi.org/10.1115/1.4041700 031014.

Lawson Fulton, Vismay Modi, David Duvenaud, David I. W. Levin, and Alec Jacobson. 2019. Latent-space Dynamics for Reduced Deformable Simulation. *Computer Graphics Forum* (2019). https://doi.org/10.1111/cgf.13645

Eitan Grinspun, Anil N Hirani, Mathieu Desbrun, and Peter Schröder. 2003. Discrete shells. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation*. Eurographics Association, 62–67.

Fabian Hahn, Bernhard Thomaszewski, Stelian Coros, Robert W. Sumner, Forrester Cole, Mark Meyer, Tony DeRose, and Markus Gross. 2014. Subspace Clothing Simulation Using Adaptive Bases. *ACM Trans. Graph.* 33, 4, Article 105 (July 2014), 9 pages. https://doi.org/10.1145/2601097.2601160

Kenneth C. Hall, Kivanc Ekici, Jeffrey P. Thomas, and Earl H. Dowell. 2013. Harmonic balance methods applied to computational fluid dynamics problems. *International Journal of Computational Fluid Dynamics* 27, 2 (2013), 52–67. https://doi.org/10.1080/10618562.2012.742512 arXiv:https://doi.org/10.1080/10618562.2012.742512

Klaus Hildebrandt, Christian Schulz, Christoph Von Tycowicz, and Konrad Polthier. 2011. Interactive Surface Modeling Using Modal Analysis. *ACM Trans. Graph.* 30, 5, Article 119 (Oct. 2011), 11 pages. https://doi.org/10.1145/2019627.2019638

Shayan Hoshyari, Hongyi Xu, Espen Knoop, Stelian Coros, and Moritz Bächer. 2019. Vibration-minimizing motion retargeting for robotic characters. *ACM Transactions on Graphics (TOG)* 38, 4 (2019), 1–14.

Gaëtan Kerschen, Maxime Peeters, Jean-Claude Golinval, and Alexander F Vakakis. 2009. Nonlinear normal modes, Part I: A useful framework for the structural dynamicist. *Mechanical Systems and Signal Processing* 23, 1 (2009), 170–194.

Theodore Kim and Doug L. James. 2009. Skipping Steps in Deformable Simulation with Online Model Reduction. *ACM Trans. Graph.* 28, 5 (Dec. 2009), 1–9. https://doi.org/10.1145/1618452.1618469

Malte Krack and Johann Gross. 2019. *Harmonic Balance for Nonlinear Vibration Problems*. Springer.

Malte Krack, Loic Salles, and Fabrice Thouverez. 2017. Vibration prediction of bladed disks coupled by friction joints. *Archives of Computational Methods in Engineering* 24, 3 (2017), 589–636.

Dingzeyu Li, David I. W. Levin, Wojciech Matusik, and Changxi Zheng. 2016. Acoustic Voxels: Computational Optimization of Modular Acoustic Filters. *ACM Trans. Graph.* 35, 4, Article Article 88 (July 2016), 12 pages. https://doi.org/10.1145/2897824.2925960

Vittorio Megaro, Jonas Zehnder, Moritz Bächer, Stelian Coros, Markus H Gross, and Bernhard Thomaszewski. 2017. A computational design tool for compliant mechanisms. *ACM Trans. Graph.* 36, 4 (2017), 82–1.

Federico L. Moro, Alexander Spröwitz, Alexandre Tuleu, Massimo Vespignani, Nikos G. Tsagarakis, Auke J. Ijspeert, and Darwin G. Caldwell. 2013. Horse-like walking, trotting, and galloping derived from kinematic Motion Primitives (kMPs) and their application to walk/trot transitions in a compliant quadruped robot. *Biological*

Cybernetics 107, 3 (2013).

Simon Pabst, Sybille Krzywinski, Andrea Schenk, and Bernhard Thomaszewski. 2008. Seams and Bending in Cloth Simulation. In *Workshop in Virtual Reality Inter-actions and Physical Simulation "VRIPHYS" (2008)*, Francois Faure and Matthias Teschner (Eds.). The Eurographics Association. https://doi.org/10.2312/PE/vriphys/vriphys08/031-038

Zherong Pan, Hujun Bao, and Jin Huang. 2015. Subspace Dynamic Simulation Using Rotation-Strain Coordinates. *ACM Trans. Graph.* 34, 6, Article 242 (Oct. 2015), 12 pages. https://doi.org/10.1145/2816795.2818090

Maxime Peeters, Régis Viguié, Guillaume Sérandour, Gaëtan Kerschen, and J-C Golinval. 2009. Nonlinear normal modes, Part II: Toward a practical computation using numerical continuation techniques. *Mechanical systems and signal processing* 23, 1 (2009), 195–216.

A. Pentland and J. Williams. 1989. Good Vibrations: Modal Dynamics for Graphics and Animation. In *Proceedings of the 16th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '89)*. Association for Computing Machinery, New York, NY, USA, 215–222. https://doi.org/10.1145/74333.74355

Rüdiger Seydel. 2009. *Practical bifurcation and stability analysis.* Vol. 5. Springer Science & Business Media.

Ahmed A. Shabana. 1990. *Theory of Vibration, Volume II: Discrete and Continuous Systems.* Springer, New York, NY, USA.

Mélina Skouras, Bernhard Thomaszewski, Stelian Coros, Bernd Bickel, and Markus Gross. 2013. Computational Design of Actuated Deformable Characters. *ACM Trans. Graph.* 32, 4, Article Article 82 (July 2013), 10 pages. https://doi.org/10.1145/2461912.2461979

Mélina Skouras, Bernhard Thomaszewski, Peter Kaufmann, Akash Garg, Bernd Bickel, Eitan Grinspun, and Markus Gross. 2014. Designing Inflatable Structures. *ACM Trans. Graph. (Proc. SIGGRAPH)* 33, 4 (2014).

Takuto Takahashi, Jonas Zehnder, Hiroshi G. Okuno, Shigeki Sugano, Stelian Coros, and Bernhard Thomaszewski. 2019. Computational Design of Statically Balanced Planar Spring Mechanisms. *IEEE Robotics and Automation Letters* 4 (2019), 4438–4444.

Bernhard Thomaszewski, Stelian Coros, Damien Gauge, Vittorio Megaro, Eitan Grinspun, and Markus Gross. 2014. Computational Design of Linkage-Based Characters. *ACM Trans. Graph.* 33, 4, Article 64 (July 2014), 9 pages. https://doi.org/10.1145/2601097.2601143

Nobuyuki Umetani, Jun Mitani, Takeo Igarashi, and Kenshi Takayama. 2010. Designing Custommade Metallophone with Concurrent Eigenanalysis. In *New Interfaces for Musical Expression++ (NIME++)*. 26–30.

Nobuyuki Umetani, Athina Panotopoulou, Ryan Schmidt, and Emily Whiting. 2016. Printone: Interactive Resonance Simulation for Free-Form Print-Wind Instrument Design. *ACM Trans. Graph.* 35, 6, Article 184 (Nov. 2016), 14 pages. https://doi.org/10.1145/2980179.2980250

Hongyi Xu, Espen Knoop, Stelian Coros, and Moritz Bächer. 2018. Bend-it: Design and Fabrication of Kinetic Wire Characters. *ACM Trans. Graph.* 37, 6, Article 239 (Dec. 2018), 15 pages. https://doi.org/10.1145/3272127.3275089

Jonas Zehnder, Stelian Coros, and Bernhard Thomaszewski. 2016. Designing Structurally-Sound Ornamental Curve Networks. *ACM Trans. Graph.* 35, 4, Article 99 (July 2016), 10 pages. https://doi.org/10.1145/2897824.2925888

Ran Zhang, Thomas Auzinger, Duygu Ceylan, Wilmot Li, and Bernd Bickel. 2017. Functionality-Aware Retargeting of Mechanisms to 3D Shapes. *ACM Trans. Graph.* 36, 4, Article Article 81 (July 2017), 13 pages. https://doi.org/10.1145/3072959.3073710

Changxi Zheng and Doug L. James. 2011. Toward High-Quality Modal Contact Sound. *ACM Trans. Graph.* 30, 4, Article Article 38 (July 2011), 12 pages. https://doi.org/10.1145/2010324.1964933

A FREQUENCY-SPACE EQUILIBRIUM EQUATIONS

To derive the dynamic equilibrium equations in frequency space, we start by rewriting the time-dependent nodal positions (4) and forces (5) in matrix as

$$\mathbf{x}(t) = (\mathbf{Q}(t) \otimes \mathbb{I}_{3n})\mathbf{z} , \tag{29}$$

$$\mathbf{f}(t) = (\mathbf{Q}(t) \otimes \mathbb{I}_{3n})\mathbf{b} \tag{30}$$

where $\mathbf{z}$ and $\mathbf{b}$ are Fourier coefficients for positions and forces, and $\mathbf{Q}(t)$ holds the individual terms of the sine and cosine series

$$Q(t) = [1 \ \sin(\omega t) \ \cos(\omega t) \ \dots \ \sin(N_H \omega t) \ \cos(N_H \omega t)] . \tag{31}$$

Similarly, velocities and accelerations are expressed as

$$\dot{\mathbf{x}}(t) = (\dot{Q}(t) \otimes \mathbb{I}_{3n})\mathbf{z} = ((\mathbf{Q}(t)\nabla) \otimes \mathbb{I}_{3n})\mathbf{z} , \tag{32}$$

$$\ddot{\mathbf{x}}(t) = (\ddot{Q}(t) \otimes \mathbb{I}_{3n})\mathbf{z} = ((\mathbf{Q}(t)\nabla^2) \otimes \mathbb{I}_{3n})\mathbf{z} \tag{33}$$

where the derivative operators are defined as

$$\nabla = \mathrm{diag}(\mathbf{0}, \nabla_1, \dots, \nabla_j, \dots, \nabla_{N_H}) , \tag{34}$$

$$\nabla^2 = \mathrm{diag}(\mathbf{0}, \nabla_1^2, \dots, \nabla_j^2, \dots, \nabla_{N_H}^2) \tag{35}$$

with

$$\nabla_j = \begin{bmatrix} 0 & -j\omega \\ j\omega & 0 \end{bmatrix} \quad \text{and} \quad \nabla_j^2 = \begin{bmatrix} -(j\omega)^2 & 0 \\ 0 & -(j\omega)^2 \end{bmatrix} . \tag{36}$$

Substituting expressions (29, 30) and (32, 33) into the equations of motion (2) yields

$$\mathbf{M}((\mathbf{Q}(t)\nabla^2) \otimes \mathbb{I}_{3n})\mathbf{z} + \hat{\mathbf{D}}((\mathbf{Q}(t)\nabla) \otimes \mathbb{I}_{3n})\mathbf{z} = (\mathbf{Q}(t) \otimes \mathbb{I}_{3n})\mathbf{b} . \tag{37}$$

The mixed-product property of the Kronecker tensor product

$$(\mathbf{A} \otimes \mathbf{B})(\mathbf{C} \otimes \mathbf{D}) = (\mathbf{AC}) \otimes (\mathbf{BD})$$

is then applied to the left-hand side of the equation as

$$\mathbf{M}((\mathbf{Q}(t)\nabla^2) \otimes \mathbb{I}_{3n})\mathbf{z} = (1 \otimes \mathbf{M})((\mathbf{Q}(t)\nabla^2) \otimes \mathbb{I}_{3n})\mathbf{z}$$
$$= ((\mathbf{Q}(t)\nabla^2) \otimes \mathbf{M})\mathbf{z} ,$$
$$\hat{\mathbf{D}}((\mathbf{Q}(t)\nabla) \otimes \mathbb{I}_{3n})\mathbf{z} = (1 \otimes \hat{\mathbf{D}})((\mathbf{Q}(t)\nabla) \otimes \mathbb{I}_{3n})\mathbf{z}$$
$$= ((\mathbf{Q}(t)\nabla) \otimes \hat{\mathbf{D}})\mathbf{z} .$$

Using this reformulation, Eq. (37) is rewritten as

$$((\mathbf{Q}(t)\nabla^2) \otimes \mathbf{M})\mathbf{z} + ((\mathbf{Q}(t)\nabla) \otimes \hat{\mathbf{D}})\mathbf{z} = (\mathbf{Q}(t) \otimes \mathbb{I}_{3n})\mathbf{b} . \tag{38}$$

The time dependency can be removed by a Galerkin procedure, projecting (38) onto the orthogonal trigonometric basis $\mathbf{Q}(t)$ and integrating over the period T of the external force, to obtain

$$\left(\left(\frac{2}{T}\int_0^T \mathbf{Q}^T(t)\mathbf{Q}(t)dt\nabla^2\right) \otimes \mathbf{M}\right)\mathbf{z} + \left(\left(\frac{2}{T}\int_0^T \mathbf{Q}^T(t)\mathbf{Q}(t)dt\nabla\right) \otimes \hat{\mathbf{D}}\right)\mathbf{z} = \left(\left(\frac{2}{T}\int_0^T \mathbf{Q}^T(t)\mathbf{Q}(t)dt\right) \otimes \mathbb{I}_{3n}\right)\mathbf{b} \tag{39}$$

Since orthogonality of $\mathbf{Q}(t)$ implies

$$\frac{2}{T}\int_0^T \mathbf{Q}^T(t)\mathbf{Q}(t)dt = \mathbb{I}_{2N_H+1} , \tag{40}$$

Eq. (37) finally transforms into a set of algebraic equations

$$(\nabla^2 \otimes \mathbf{M})\mathbf{z} + (\nabla \otimes \hat{\mathbf{D}})\mathbf{z} = (\mathbb{I}_{2N_H+1} \otimes \mathbb{I}_{3n})\mathbf{b} \tag{41}$$

which can also be written in a more compact form to yield (8).

B DFT OPERATORS

Using Eq. (29), we can define the inverse DFT operator that transforms from Fourier coefficients to time domain positions as

$$\tilde{\mathbf{x}} = [\mathbf{Q}(t_1) \otimes \mathbb{I}_{3n}, \ \dots , \mathbf{Q}(t_N) \otimes \mathbb{I}_{3n}]^T \mathbf{z} = (\Gamma \otimes \mathbb{I}_{3n})\mathbf{z} \equiv \Gamma_x \mathbf{z} . \tag{42}$$

Similarly, using Eq. (32), the inverse DFT operator for velocity is obtained as

$$\tilde{\mathbf{v}} = ((\Gamma\Delta) \otimes \mathbb{I}_{3n})\mathbf{z} \equiv \Gamma_v \mathbf{z} . \tag{43}$$

Having transformed positions and velocities from frequency space to the time domain, we can evaluate the nonlinear forces for all

sampling points and then transform them back to the frequency domain as

$$\mathbf{b} = (\Gamma \otimes \mathbb{I}_{3n})^{-1}\tilde{\mathbf{f}} = (\Gamma^{-1} \otimes \mathbb{I}_{3n})\tilde{\mathbf{f}} \equiv \Gamma_f^{-1}\tilde{\mathbf{f}} \; . \tag{44}$$

The operators Γ and Γ^{-1} are given as

$$\Gamma = \begin{bmatrix} 1 & \sin(\theta_1) & \cos(\theta_1) & \dots & \sin(N_H\theta_1) & \cos(N_H\theta_1) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \sin(\theta_N) & \cos(\theta_N) & \dots & \sin(N_H\theta_N) & \cos(N_H\theta_N) \end{bmatrix} \quad , \tag{45}$$

$$\Gamma^{-1} = \frac{1}{N} \begin{bmatrix} 1 & \dots & 1 \\ 2\sin(\theta_1) & \dots & 2\sin(\theta_N) \\ 2\cos(\theta_1) & \dots & 2\cos(\theta_N) \\ \vdots & \vdots & \vdots \\ 2\sin(N_H\theta_1) & \dots & 2\sin(N_H\theta_N) \\ 2\cos(N_H\theta_1) & \dots & 2\cos(N_H\theta_N) \end{bmatrix} \quad . \tag{46}$$

It is worth noting that $\theta_i = \omega t_i = \omega(i\Delta t) = 2\pi i/N$ such that Γ and Γ^{-1} do not depend on ω.

C NEWMARK TIME INTEGRATION

The Newmark integration scheme determines end-of-step positions $\mathbf{x}_{t+\Delta t}$, velocities $\dot{\mathbf{x}}_{t+\Delta t}$ and accelerations $\ddot{\mathbf{x}}_{t+\Delta t}$ such as to satisfy

$$\dot{\mathbf{x}}_{t+\Delta t} = \dot{\mathbf{x}}_t + \Delta t(1 - \gamma)\ddot{\mathbf{x}}_t + \Delta t\gamma\ddot{\mathbf{x}}_{t+\Delta t} \; , \tag{47}$$

$$\mathbf{x}_{t+\Delta t} = \mathbf{x}_t + \Delta t\dot{\mathbf{x}}_t + \frac{1}{2}\Delta t^2\left((1 - 2\beta)\ddot{\mathbf{x}}_t + 2\beta\ddot{\mathbf{x}}_{t+\Delta t}\right) \tag{48}$$

where $0 \leq \beta \leq \frac{1}{2}$ and $0 \leq \gamma \leq 1$ are parameters that we set to $\beta = \frac{1}{4}$ and $\gamma = \frac{1}{2}$ to obtain second-order accuracy. We express $\dot{\mathbf{x}}_{t+\Delta t}$ and $\ddot{\mathbf{x}}_{t+\Delta t}$ in terms of $\mathbf{x}_{t+\Delta t}$ and known quantities at time t as

$$\ddot{\mathbf{x}}_{t+\Delta t} = \frac{1}{\beta\Delta t^2}(\mathbf{x}_{t+\Delta t} - \mathbf{x}_t) - \frac{1}{\beta\Delta t}\dot{\mathbf{x}}_t - \frac{1 - 2\beta}{2\beta}\ddot{\mathbf{x}}_t \; , \tag{49}$$

$$\dot{\mathbf{x}}_{t+\Delta t} = \frac{\gamma}{\beta\Delta t}(\mathbf{x}_{t+\Delta t} - \mathbf{x}_t) + \left(1 - \frac{\gamma}{\beta}\right)\dot{\mathbf{x}}_t + \Delta t\left(1 - \frac{\gamma}{2\beta}\right)\ddot{\mathbf{x}}_t \; . \tag{50}$$

Substituting the above equations into the time-domain equations of motion (1) yields

$$\mathbf{R}(\mathbf{x}_{t+\Delta t}) = \mathbf{M}\left(\frac{1}{\beta\Delta t^2}(\mathbf{x}_{t+\Delta t} - \mathbf{x}_t) - \frac{1}{\beta\Delta t}\dot{\mathbf{x}}_t - \frac{1 - 2\beta}{2\beta}\ddot{\mathbf{x}}_t\right) +$$

$$\left[D_\alpha\mathbf{M} + D_\beta\mathbf{K}(\mathbf{x}_{t+\Delta t})\right]\left(\frac{\gamma}{\beta\Delta t}(\mathbf{x}_{t+\Delta t} - \mathbf{x}_t) + \left(1 - \frac{\gamma}{\beta}\right)\dot{\mathbf{x}}_t +\right.$$

$$\left.\Delta t\left(1 - \frac{\gamma}{2\beta}\right)\ddot{\mathbf{x}}_t\right) - \mathbf{f}_{\text{int}}(\mathbf{x}_{t+\Delta t}) - \mathbf{f}_{\text{ext}}(\mathbf{x}_{t+\Delta t}) = \mathbf{0} \; . \tag{51}$$

We solve the above system of nonlinear equations using Newton's method. Note that the tangential stiffness matrix depends on the end-of-step positions $\mathbf{x}_{t+\Delta t}$.

D ACCURACY OF HBM VS. NEWMARK

To further investigate the accuracy of HBM compared to Newmark, we additionally analyze the impact of the number of sample points used to evaluate the nonlinear forces in the time domain.

It can be seen from Fig. 3 that, with increasing number of harmonics, the error between HBM and Newmark initially decreases very rapidly. For $N_H > 5$, however, the error stays almost constant. This seemingly odd behavior is explained by the fact that the reference
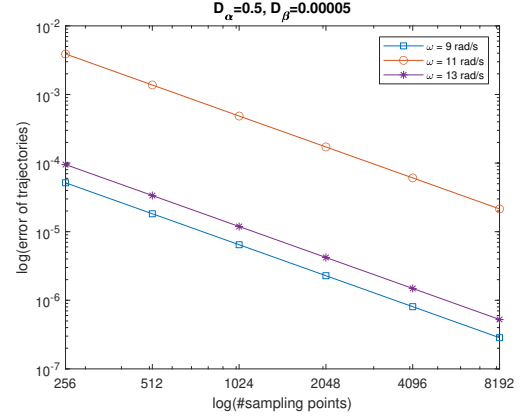
Fig. 15. Trajectory difference between HBM and Newmark as a function of the number of time-domain samples N_{AFT} for $N_H = 5$ harmonics. The step size for Newmark is set as $\Delta t = T/N_{AFT}$, where T is the period of the forcing.

solution, computed using Newmark, is itself subject to discretization error. We can reduce this time-domain error by using smaller step sizes and, as we increase both the number of harmonics N_H and time-domain samples N_{AFT}, the error should vanish. This expectation is supported by the results shown in Fig. 15, indicating a sustained decrease in error for increasing number of time-domain samples and, accordingly, the step size used for Newmark.

E MECHANICAL MODELS

Our approach extends to a large range of mechanical models and we show examples using compliant mechanisms, rod networks, thin shells, and volumetric solids. All computational models are implementations of standard approaches. We represent compliant mechanisms through their joint positions, enforcing angular and distance constraints for rigid joints and links with stiff penalty terms. Compliant elements are modeled using standard linear springs whose endpoints are expressed in local coordinates of the corresponding link. For the rod network example, we use discrete elastic rods [Bergou et al. 2008] together with the formulation by Zehnder et al. [2016] for network connections. For our thin shell examples, we use the discrete shell model by Grinspun et al. [2003] for bending and linear triangle finite elements with St. Venant-Kirchhoff material [Skouras et al. 2014] for in-plane deformations. Our solid example uses standard linear tetrahedron elements with a St. Venant-Kirchhoff material.

Each model gives rise to a discrete elastic energy, whose negative gradients with respect to time-domain positions and velocities determine internal forces which enter the frequency-space equations of motion (8) through the nonlinear force term (3). For sensitivity analysis, we require the derivatives of the forces with respect to material parameters, masses, and rest shape. All of these derivatives are computed symbolically using compile-time automatic differentiation. Additionally, we use the same penalty term for all mechanical model to enforce harmonic motion for the boundary nodes attached to the motor or driving element.