Modal Locomotion:
Animating Virtual Characters with Natural Vibrations

P. G. Kry[1] and L. Reveret[2] and F. Faure[2] and M.-P. Cani[2]

[1]McGill University, School of Computer Science
[2]Grenoble Universities & CNRS, Jean Kuntzmann Lab, INRIA Grenoble Rône-Alpes

Abstract

We present a general method to intuitively create a wide range of locomotion controllers for 3D legged characters. The key of our approach is the assumption that efficient locomotion can exploit the natural vibration modes of the body, where these modes are related to morphological parameters such as the shape, size, mass, and joint stiffness. The vibration modes are computed for a mechanical model of any 3D character with rigid bones, elastic joints, and additional constraints as desired. A small number of vibration modes can be selected with respect to their relevance to locomotion patterns and combined into a compact controller driven by very few parameters. We show that these controllers can be used in dynamic simulations of simple creatures, and for kinematic animations of more complex creatures of a variety of shapes and sizes.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism - Animation

1. Introduction

Locomotion is a fundamental activity of living creatures. It is possible that animals have evolved to optimize this energetically expensive function, but they also learn to use their body in efficient ways to produce locomotion styles that depend on speed and task [Ale96]. For instance, an animal might use a slow gait to search for shelter, and use a different faster gait to escape from a predator. Because the different gaits of a particular animal strongly depend on the morphology of that animal, it is possible that we can deduce information about plausible modes of locomotion just by observing that animal's musculoskeletal structure.

We present a method for generating animation of locomotion, which does not use motion capture, but instead uses only information of the physical model of the virtual animal. For a given model we compute a set of basis motions that can be combined in small number to describe different gaits for that model. Changes in the virtual animal model, such as weight, injury, rest pose, and muscle tension and relaxation, all produce subtle yet important differences in the basis motions and the resulting generated animation.

Skeletal structure, deformation of tissues, and stiffness of muscles and tendons all play an important role in the dynamics of real animals. An appealing strategy for modelling the dynamics of a virtual animal is to simplify the biomechanical system as a set of rigid bones connected by compliant joints, where the stiffness of the joints approximates the combined action of tendons, the activation of muscles, and the deformation of the surrounding tissues. Even with this simplified model of the biomechanics, most vertebrates have hundreds of degrees of freedom making it challenging to animate virtual models of these animals. While it is possible to remove some degrees of freedom from the skeletal structure (for instance, a simplified spine), we instead perform a modal analysis to identify a reduced set of coordinated joint motions that are useful for animating a virtual animal.

The natural vibrations identified by the modal analysis provide an analytical solution to the passive response of the elastic dynamics of the system [PW89]. We focus on lower frequency modes because at a given energy level they involve larger displacements than higher frequency modes and damping dissipates this energy more slowly. The intuition is that our virtual animals might take advantage of swinging

limbs and elastic compression at joints in a dynamic cycle that produces an efficient locomotion; efficient because it is the passive response to a given set of initial conditions, or more accurately, almost passive with only small regular pushes from the muscles being necessary to inject lost energy back into the system.

In this paper we show that modal vibrations can be combined in small number to produce locomotion controllers for a variety of virtual animals. The locomotion controllers can be dynamic, but they can also be purely kinematic. This is a significant contribution of our approach, since it is very easy to create kinematic motions with our technique which have a natural appearance due to the physically based nature of the basis motions. Instead of animating individual joints, for instance, the hip joints of a human, we can animate a mode that exhibits a leg swinging motion. But this leg swinging mode will also include nuanced motion in the rest of the body (such as upper body and spinal motions that conserve angular momentum). In general, searching for controller parameters is simplified since we only need to consider a small number of intuitive modes rather than a large number of degrees of freedom. This search can be easily done by hand, but in many cases it is also possible to do so automatically with a set of heuristics.

2. Related Work

Modeling and animating locomotion has attracted a lot of attention for many years in Computer Graphics, in addition to a huge amount of research in robotics and biomechanics. Locomotion controllers have been proposed for the computer animation of all sorts of virtual creatures. Motor control has been used to animate cockroaches [MZ90], fish [GT95], and birds [WP03]. Sun and Metaxas [SM01] demonstrated a gait generation technique which produces kinematic motion of human walking on a variety of different terrains. Also with respect to human models, Hodgins proposed controllers for human athletics [HWBO95] and later studied the way to adapt some of them to new characters [HP97]. This last work demonstrates that tuning a controller to a different morphology, even within the same species, is far from obvious. Therefore, there is interest in working towards more automatic generation of animation of arbitrary virtual animals [HRE*08].

Controlling locomotion is complex because many degrees of freedom must be managed simultaneously. The large dimensionality of the search space is a significant problem in designing a controller. A common approach for automatic generation of controllers is off-line optimization (maximization of the covered distance) over a large space of possible controllers. Sims used genetic programming to cope with the size of the search space, and evolved both the creature's structure and the controller simultaneously [Sim94]. In contrast, our interest is in the generation of plausible controllers for a given input bio-mechanical structure. In other work,

van de Panne [vF93, vKF94, vdP96] used optimization to tune the weights of simplified neural networks representing controllers. Grzeszczuk and Terzopoulos [GT95] introduced a multi-level learning process and a compact representation of controllers based on Fourier analysis, enabling them to be combined within a sensor feedback loop. They later developed neural networks as a way to efficiently emulate the control of physically based motion [GTH98].

The automation of controller generation can be simplified through *a priori* knowledge or example data to speed up or guide results. For instance, general momentum parametric templates inspired by biomechanics can be used to create complex dynamic motion [LP02]. This idea of motion-patterns is also illustrated in the work of [WP03] to represent the motor control of the beat of a bird wing. While recent work in graphics is now largely focusing on motion capture, much of this work is still concerned with optimal physically based modifications or identifying underlying physical parameters. Low-dimensional subspaces learned from motion capture databases can help with optimization of realistic human animation [SHP04]. Likewise, when an accurate database of real motion data is available, Liu et al. [LHP05] have shown that physical properties of the model (e.g., stiffness) can be estimated. Motion capture of quadrupeds, however, is not as widely available as it is for humans. However, foot prints can be successfully used as input data and combined with a simplified physically based representation of the skeleton to resythesize skeletal motion [TvdP98]. Other work has exploited side-view video data to reconstruct some types of 3D motion [Wil97, FRDC04]. In contrast, the only *a priori* knowledge we use is the morphological parameters (that is, the animal shape, size, mass, rest pose, joint stiffness). These parameters also effectively capture style (as in [LHP05]). Neff also demonstrates that modulating stiffness, specifically muscular tension and relaxation, is necessary for tuning the expression and style of an articulated motion [NF02]. This concept is also critical in our work because muscular tension (stiffness) directly affects the vibration modes and frequencies. Similar to our work, Faloutsos et al. use a reduced number of deformation modes, though user designed, to simplify the control problem for deformable models [FvdPT97].

The term passive dynamics was coined by McGeer [McG90] to describe an approach of controlling robotic movement that takes advantage of the passive movements, such as swinging legs and arms, rather than active control with motors applying torques at joints. Passive dynamics is relevant to locomotion because it is energetically efficient (that is, locomotion by passive dynamics is effectively free). Passive dynamic locomotion has been demonstrated with mechanical leg systems (no motors and no control) that produce dynamic walk cycles that resemble human walking [McG90, CWR01]. Collins et al. [CRTW05] provide a good recent survey of the vast amount of work done by a variety of people. This survey

also addresses work on efficient walkers (e.g., [Kuo02]) that walk on flat ground in contrast to the completely passive walkers which only walk down a gentle slope. In general, this view of efficient locomotion is a key inspiration to our work. Our model of locomotion as a superposition of natural vibration is also related to work on coupled oscillators in neural models for controlling human locomotion [Tag95]. The use of central pattern generators for modeling and controlling locomotion in robots and simulation has been and is still an active area or research; see [Ijs08] for a recent survey.

Finally, much like other work [MA90, RH91], we note that a skeletal system stores energy during locomotion through the compression of elastic joints. While the idea that this is a key component of locomotion is well known in biomechanics [Ale88], it is most commonly illustrated with simple models since these simple models are still a powerful tool for answering important questions about different gaits. A key contribution of our technique is that modal analysis similarly reduces the complexity of a 3D model while preserving important characteristics and behaviour of the original model, such as mass distribution and spinal flexion.

3. Dynamics of Virtual Animals

Our approach to creating virtual animal locomotion starts with the physically based model of the creature we want to animate. In practice, we use a slightly simplified skeleton to avoid slow simulations involving every vertebra of the spine (nonetheless, we could easily use the full model in our vibration analysis). For example, the dog model in Figure 1 has only 80 degrees of freedom; just the same, this is a large number of joints if we were to animate every joint by hand.

Each mechanical link, that is, each bone or small set of bones, is assigned a mass and an inertia matrix. These values can be assigned based on those reported in the biomechanics literature; however, in our case we compute them automatically from the geometry of the given model. We set the stiffness of each joint to account for the combined action of muscles and tendons; this models a given level of tension or relaxation about a given pose. This pose is effectively a rest configuration with respect to the equilibrium-point model of human motor control [Fel86, BHMIG92]. Stiffness values estimated by measurement can be found in the literature as well, but we find that a reasonable initial set of values can be obtained by choosing a minimal stiffness that allows the virtual animal to maintain an upright natural rest posture in a simulation with normal gravity. For our quadruped example, this consists of a lower stiffness for the legs, while the back and neck are somewhat stiffer to maintain the head posture. The resulting motions in this case will be in a relaxed style; changing the stiffness and the equilibrium pose will alter the style, but we will discuss this in greater detail later.

Given this physical model consisting of the equilibrium pose, the stiffness of the joints, and the mass of the links, we can use a physically based simulation to evolve its motion over time. We use a standard constrained multibody approach, which we describe in detail in Appendix A; however, let us quickly review that this motion is given by the Newton-Euler equation

$$M\dot{\phi} = w_\phi + w + G^T \lambda. \tag{1}$$

Here M is the mass matrix, and ϕ and w are block column vectors containing velocities and wrenches (i.e., forces and torques) of all bodies. We express all quantities in the moving body frames, so we must include the wrench w_ϕ due to the current spatial velocities. Joint constraints, and other constraints, require the solution to satisfy $G\phi = 0$, thus Equation 1 also includes constraint forces $G^T \lambda$. Note that w includes wrenches due to gravity, damping, and external forces, but most importantly it includes wrenches due to the displacement of the joints from the equilibrium pose. By defining R to map joint torques to opposing wrenches on body pairs, this wrench has the form $w_k = R^T K(\theta - \theta_0)$, where K is a diagonal matrix containing the stiffness of each joint. Note, R also transforms the body velocities to angular velocities at the joints, $\dot{\theta} = R\phi$.

3.1. Modal Analysis with Constraints

We are not interested in just simulating the dynamics of a virtual animal; we want to generate animations of locomotion. As outlined in the introduction we do this by combining selected low-frequency natural vibrations as basis motions. In this section we briefly describe how we compute the modal analysis of our virtual animals (for more detail we refer the reader to [Sha97]). Note that the analysis we show here differs from that which is typically done in graphics. We deal with constraints and rigid articulated systems in contrast to the analysis of jelly-like objects, trees, and other elastic materials [PW89, Sta97].

Suppose that the compliant joints in our virtual animal's skeleton are in equilibrium. Given a small twist away from equilibrium, ϕh, the wrenches due to the compliant joints will be $R^T K R\phi h$. Ignoring damping for now, we return to Equation 1, but omit the constraint forces, and drop the w_ϕ term since our analysis concerns small low-frequency vibration about the equilibrium with insignificant changes in body frame orientation. Replacing w with this spring force above gives

$$M\dot{\phi} = R^T K R\phi h. \tag{2}$$

When calculating the vibration modes, we must only take the permissible directions of constrained motion into account. The degrees of freedom of the system are described by the null space of G. For simple chains and trees this null space is straightforward to construct and related to the manipulator Jacobian. However, for loops (e.g., hands holding a basket or pushing a shopping cart) we must solve for the free directions. We use singular value decomposition of G to find

the null space N, which provides the spatial twists that form the minimal set of kinematically admissible directions. Thus, $\phi = N\dot{x}$, where $\dot{x}$ is the velocity in admissible coordinates. Substituting x for $\dot{x}h$, and letting $\dot{\phi} = N\ddot{x}$ (since we are only looking at small vibrations we let $\dot{N} = 0$), we project Equation 2 into these minimal coordinates to obtain

$$\mathsf{M}\ddot{x} = \mathsf{K}x, \tag{3}$$

with $\mathsf{M} = N^T MN$ and $\mathsf{K} = N^T R^T KRN$. While K will typically be rank deficient, M is always invertible so we can easily solve for the eigenvalues and eigenvectors of Equation 3 as either a generalized eigenvalue problem or via $\mathsf{M}^{-1}\mathsf{K}$. When K is not full rank, the resulting eigenvectors include rigid motions with corresponding eigenvalues equal to zero.

The eigenvectors u_i of $\mathsf{M}^{-1}\mathsf{K}$ are the modes shapes, and can be seen as involving coordinated joint motions RNu_i about the equilibrium pose. The eigenvalues λ_i give the corresponding frequency, $f_i = \sqrt{\lambda_i}/2\pi$ Hz. In this basis, the behaviour of the system separates into a set of independent differential equations. The solution of the entire system is equivalent to the superposition of the solutions to these independent equations. Adding Rayleigh damping to the system (i.e., a damping matrix equal to a combination of the mass and stiffness matrices) does not change the shapes of the modes, but does alter the frequencies of the resulting modes.

Figure 1 depicts the first four modes for a dog model. All but one of the first four modes are parallel to the sagittal plane of the dog and are relevant for locomotion. Mode u_7, however, does not show this symmetry (twist of the whole spine) and as such can be easily rejected as not relevant to locomotion. We will mention more about selecting appropriate modes to generate interesting locomotion animations in Section 4.3.

Finally, note that a joint coordinate formulation of the mass and stiffness matrices would lead us to the same vibration shapes. But our formulation with constraints has the advantage that we can easily add additional constraints in the same framework. For instance, we can easily model a virtual human pushing a cart, or holding an object in two hands; we do this by adding constraints to the hands (forming a closed loop). We can also add biological constraints, such as a constraint to fix the orientation of the head so that it remains stable during locomotion.

4. Creating Locomotion

At this point we know how to compute a set of natural vibration modes for any given animal model. Let us now describe how we can use these modes to create locomotion.

4.1. Dynamic control of simple creatures

Let us start with very simple creatures to see how we can create open loop controllers which produce locomotion for
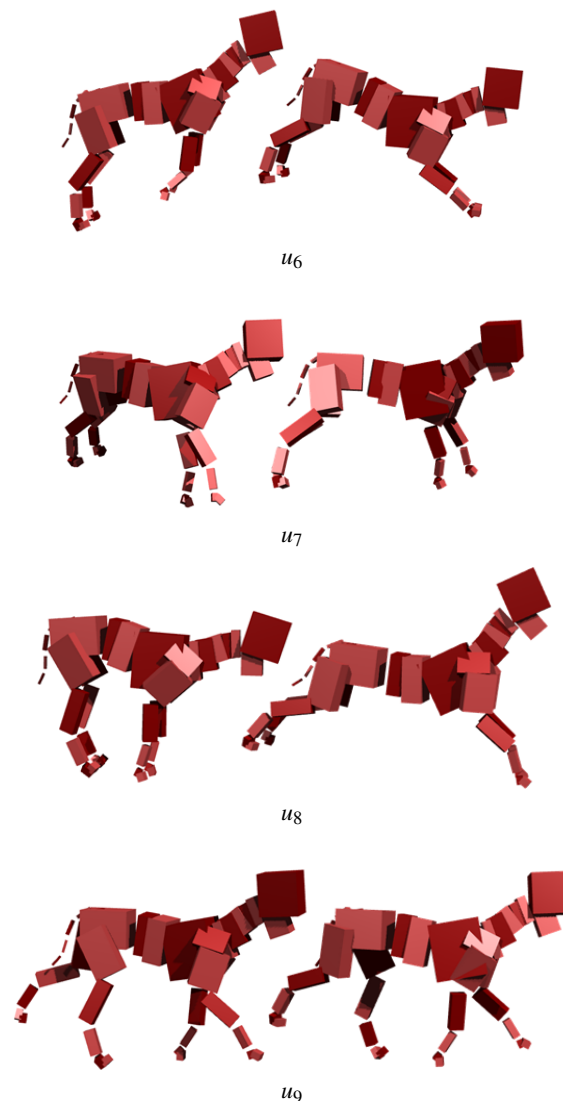

u_6

u_7

u_8

u_9

Figure 1: *The first four non rigid vibration modes of a dog model with 80 degrees of freedom. These modes can be described as bounding, back twisting, stretching, and alternating legs, respectively.*

these creatures in a physically based simulation. Consider first the simple example of a loop shaped creature modeled as an articulated chain of identical rigid bodies. Computing vibration modes for such a structure yields vibrations shapes and frequencies depicted in Figure 2 (top). A small push from the muscles in the direction of a particular mode shape will cause the whole body to vibrate with that shape. The resulting vibration is a sinusoid with an exponential decay. Thus, the goal of our controller is to pump energy into some selected modes so that they oscillate without decay at coordinated frequencies and selected phases. In this case,
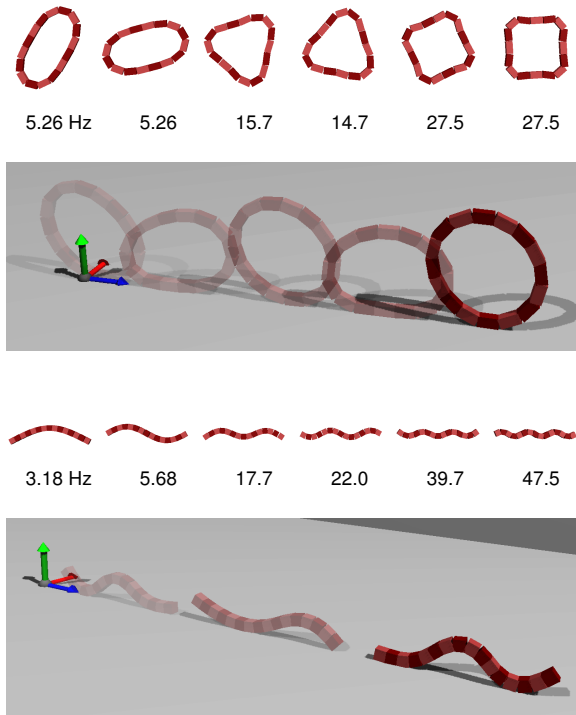
| 5.26 Hz | 5.26 | 15.7 | 14.7 | 27.5 | 27.5 |

| 3.18 Hz | 5.68 | 17.7 | 22.0 | 39.7 | 47.5 |

Figure 2: *A simple loop shaped creature (top) and a simple linear creature (bottom) showing each creature's vibration modes along with examples of locomotion obtained with by mixing only the first two modes 90 degrees out of phase of each other.*

the selected modes are the two lowest frequency modes and the controller pushes these at a frequency of 5.26 Hz, but 90 degrees out of phase. Instead of directly applying joint torques, we combine the selected vibration shapes to produce an equilibrium point for use in a PD controller where the stiffness of the controller is exactly that which was used in the analysis. The choice of modes along with their amplitudes and phases in this case are easily adjusted by hand to create an open loop controller that produce locomotion.

Figure 2 (bottom) shows a slightly different simple linear creature. Just like the loop creature, we construct a locomotion controller by combining the two lowest vibration modes, with the second mode shifted by 90 degrees. Note that in this case the vibrations have different frequencies, but this does not pose a serious problem. Damping in the system widens the resonance response of different modes, and likewise contact helps regulate the motion as collisions remove energy from the system.

We have had some success to applying this approach to more complex animals, such as dogs and humans, but ultimately balance becomes a much more significant problem in these cases. Just the same, the natural vibrations of these

complex animals still provide an excellent basis for creating kinematic animations of locomotion as we describe in the next section.

4.2. Kinematic animation

The analytic solution of a virtual animal's elastic vibrations provides a straightforward way of creating kinematic animations of locomotion. We can simply move the joints using selected mode shapes modulated by sinusoids, much like the real physical response of the system, except that we can discard the effect of damping. In this case, we also superimpose the selected modes at the frequency of the gait, or in some cases, at twice the frequency in order to animate desirable physical effects. For instance, a head bob might occur on every step, or in the case of a quadruped, its back would normally flex twice during each cycle of a trot.

We can formalize our kinematic control as follows. Note that RNu_i gives the shape of mode i in joint coordinates, and let $\tilde{f}_i$ and γ_i be the selected frequency and phase respectively. Over time, the positions of the joints are set as

$$\theta(t) = \sum_i \alpha_i RNu_i \sin(2\pi \tilde{f}_i t + \gamma_i) + \theta_0. \quad (4)$$

Here, α_i is a control parameter used to independently tune the amplitude of each mode. Note that only a small set of α_i need to be non-zero in order to create a locomotion, and as such the description is very compact. Note also that Equation 4 is very similar to the equilibrium point used in the previous section, except that here it is used to directly provide the virtual animal's configuration at each point in time.

Our virtual animals have joint limits, and there are naturally situations where these limits are reached. For example, if the rest pose has joints near their limits then oscillation around this pose may exceed the limits. We address this by simply modifying the angles produced by Equation 4 when they exceed their limits. To avoid joint popping artefacts that result from simply clamping joint angles to their limits, we instead impose the limits in a soft manner. We let each joint exceed its limits at most by a small amount $\Delta\theta_{soft}$. Any motion of a joint past its limit, which we denote d, is reduced by $d^2/(d + \Delta\theta_{soft})$.

While our kinematic controllers constrain joint motion to respect joint limits, they do not currently treat ground contact. Layering an inverse kinematics solution on top of this motion is one possible solution that we leave for future work.

4.3. Parameter selection

Let us now describe how we choose mode combinations to animate locomotion of complex virtual animals. Recall that we focus on the low-frequency modes since they require the least energy to produce and maintain large deformations of the structure. The frequencies of these modes are also appropriate for locomotion. For all the virtual animals we used

in this work, we observe that the lower modal frequencies always correspond well with the period of real gaits.

The heuristics for selecting important modes are simple. We chose modes that involve leg swings in the direction of movement, and knee or elbow bending. Modes with jumping jack motions and body twists are easily discarded (though perhaps useful for other kinds of animation). Phases can also be automatically assigned the integer multiples of 90 degrees that produces a forward cyclic motion useful for locomotion. Specifically, for walking and running we select modes that exhibit maximal alternating knee bends (A), alternating forward motion of the feet (B), and alternating vertical motion of the feet (C). Knee bending and vertical motion modes are given a phase offset of 90 degrees and we automatically enable the top ranked mode in each category. For jumping, we similarly identify modes that exhibit maximal synchronous knee bends (D), and synchronous vertical motion of the feet (E). Likewise, for quadrupeds we also include heuristics that identify bending (F) and swinging (G) of the front legs. Finally, the magnitudes can be easily tuned to adjust the motion (e.g., increase stride length). Figure 3 summarizes a number of different animations that can be easily produced with these heuristics.

Note that the equilibrium pose around which the modal vibrations are computed has an influence on the resulting locomotion. Figure 3 shows an example of this where a T-pose can be used to create a walk with the arms extended, while a rest pose involving bent knees and bent elbows is more suitable for a run.

While we can alter the rest pose as a means of modifying the style of an animation, we can also alter the style of a locomotion animation by using the mode shapes for postural changes. That is, we can add a term $\beta_i R N u_i$ to Equation 4 and set some small number of β_i to non-zero. An initial walking motion can be easily modified so that the legs are closer together, or so that arms are raised or lowered. Examples are shown in the accompanying video.

Lastly, layering mode shapes to build locomotion constrains the user to select motions from an intuitive palette of natural vibrations, ensuring plausibility and leading to low energy motions. Indeed there are still other important factors beyond mechanical efficiency that effect how an animal moves, including, for example, constraints on posture to improve sensory input from the eyes or the ears. Figure 3, shows an example where head rotation was controlled with a constraint, and thus the modal basis consists of motions that also respect this constraint.

5. Results and Discussion

The accompanying video shows examples of dynamic locomotion for the simple loop shaped and linear creatures, as well kinematic animations of dog and human models for the actions described in Figure 3. The video also shows examples of how it is easy to alter the posture of these motions by adding an offset of a low-frequency mode shape. For instance, the jumping jack animation can be modified so that the character appears to be supporting more of the weight on one leg. The last examples show the benefit of using additional constraints to animate a running human with a stable head, and a walking human pushing an imaginary cart.

Combining a small number of basis motions is an attractive method for creating animation. The natural vibrations give us an intuitive palette for creating animations of cyclic motions such as locomotion. In comparison, building motion curves for individual joints is very time consuming. Instead, we can quickly adjust a small set of automatically identified basis motions that capture important and natural physically based movements of the virtual character. This is the central contribution of our approach. This work also leads in a number of interesting directions for future work, for example, the development of motor control of fully dynamic virtual animals that builds on modal vibrations but also correctly treat ground contact and balance.

Limitations: It is important to correctly set stiffness for proper control in a dynamic simulation, but this is less important in producing a useful palette of global body motions. Changing the stiffness will alter the frequency, and some modes may have their frequencies aligned as a result. For example, in the quadruped model with uniform stiffness, lowering the stiffness on the back legs results in modal vibrations where all legs (front and back) move at once in two different modes. Regardless of how we might tune the stiffnesses, all the necessary motions for building a locomotion animation are available and can automatically be identified with heuristics. While altering the stiffness does change the exact mode shapes, this can be seen as adjusting the style of the basis motions. For example, a stiff leg will alter the leg swinging modes, but the increased stiffness of this leg will also be present in all the other modal vibrations, as well as final motions that we create.

6. Conclusion

This paper has revisited motion control of virtual animals by expressing the degrees of freedom for a controller in a new basis, the space of low-frequency vibration modes of the underlying mechanical model. This brings a number of benefits. Firstly, if associated with a good balance controller, vibration modes could be used to generate controllers for physically based motions of complex animals. Just the same, they provide a good basis for creating kinematic animations of locomotion since they involve simultaneous motion at all joints, and have a natural appearance and a physical justification. For this same reason, the modes serve as intuitive building blocks for tuning these animations. Because we limit our search to low-frequency modes and identify important modes automatically with heuristics, the search space for

model / action	DOFs	rest pose	stiffness	constraints	parameters	sample frame
loop creature	19		1 N/rad uniform	none	$\alpha_6=1$ $\gamma_6=0$ $\tilde{f}_6=5.25$ $\alpha_7=1$ $\gamma_7=1.57$ $\tilde{f}_7=5.25$	
linear creature	21		1 N/rad uniform	none	$\alpha_6=1$ $\gamma_6=0$ $\tilde{f}_6=5$ $\alpha_7=1$ $\gamma_7=1.57$ $\tilde{f}_7=5$	
dog trot	80		20 N/rad spine ×5	none	$\alpha_9=0.5$ $\gamma_9=0$ $\tilde{f}_9=2$ $\alpha_{11}=0.5$ $\gamma_{11}=0$ $\tilde{f}_{11}=2$ $\alpha_{19}=0.5$ $\gamma_{19}=1.57$ $\tilde{f}_{19}=2$ $\alpha_{21}=0.5$ $\gamma_{21}=1.57$ $\tilde{f}_{21}=2$	
dog amble	80		20 N/rad spine ×5	none	$\alpha_{10}=0.5$ $\gamma_{10}=0$ $\tilde{f}_{10}=2$ $\alpha_{11}=0.1$ $\gamma_{11}=0$ $\tilde{f}_{11}=2$ $\alpha_{19}=0.5$ $\gamma_{19}=1.57$ $\tilde{f}_{19}=2$ $\alpha_{21}=0.3$ $\gamma_{21}=1.57$ $\tilde{f}_{21}=2$	
dog gallop	80		20 N/rad spine ×5	none	$\alpha_4=0.2$ $\gamma_4=0$ $\tilde{f}_4=2$ $\alpha_6=1.0$ $\gamma_6=1.57$ $\tilde{f}_6=2$ $\alpha_8=1.0$ $\gamma_8=0$ $\tilde{f}_8=2$ $\alpha_{10}=0.2$ $\gamma_{10}=1.57$ $\tilde{f}_{10}=2$ $\alpha_{11}=0.7$ $\gamma_{11}=1.57$ $\tilde{f}_{11}=2$ $\alpha_{22}=1.0$ $\gamma_{22}=1.57$ $\tilde{f}_{22}=2$	
human T-pose walk	47		100 N/rad spine ×10	none	$\alpha_6=0.5$ $\gamma_6=0$ $\tilde{f}_6=2$ $\alpha_{17}=0.76$ $\gamma_{17}=1.57$ $\tilde{f}_{17}=2$	
human jumping	47		100 N/rad spine ×10	none	$\alpha_4=0.75$ $\gamma_6=0$ $\tilde{f}_6=2$ $\alpha_{16}=0.75$ $\gamma_{16}=0$ $\tilde{f}_{16}=2$ $\alpha_{24}=0.75$ $\gamma_{24}=0$ $\tilde{f}_{24}=2$	
human jumping jacks	47		100 N/rad spine ×10	none	$\alpha_4=0.75$ $\gamma_6=0$ $\tilde{f}_6=2$ $\alpha_7=0.35$ $\gamma_7=0.79$ $\tilde{f}_7=1$ $\beta_7=-0.3$ $\alpha_{10}=0.95$ $\gamma_{10}=0.79$ $\tilde{f}_{10}=1$ $\alpha_{16}=0.75$ $\gamma_{16}=0$ $\tilde{f}_{16}=2$ $\alpha_{24}=0.75$ $\gamma_{24}=0$ $\tilde{f}_{24}=2$	
human run	47		100 N/rad spine ×10	none	$\alpha_6=0.5$ $\gamma_6=0$ $\tilde{f}_6=2$ $\alpha_{13}=0.5$ $\gamma_{13}=1.57$ $\tilde{f}_{13}=2$ $\alpha_{15}=0.75$ $\gamma_{15}=1.57$ $\tilde{f}_{15}=2$	
human run	44		100 N/rad uniform	head	$\alpha_5=0.5$ $\gamma_5=0$ $\tilde{f}_5=2$ $\alpha_7=0.5$ $\gamma_7=0$ $\tilde{f}_7=2$ $\alpha_{13}=0.5$ $\gamma_{13}=1.57$ $\tilde{f}_{13}=2$ $\alpha_{14}=0.5$ $\gamma_{14}=1.57$ $\tilde{f}_{14}=2$	
human pushing cart	42		100 N/rad spine ×10	hands	$\alpha_2=0.5$ $\gamma_2=0$ $\tilde{f}_2=2$ $\alpha_9=0.5$ $\gamma_9=1.57$ $\tilde{f}_9=2$ $\alpha_{10}=0.5$ $\gamma_{10}=1.57$ $\tilde{f}_{10}=2$	

Figure 3: *An overview of different animation parameters for a variety of animals, gaits and situations. Heuristics (A) through (F) were used to identify the modes in all cases except the linear and loop shaped and creatures, the dog gallop, and the human jumping jacks.*

the animation parameters is greatly simplified, especially in the case of complex virtual animals. Finally, our method for finding these mode shapes is specially tuned to the case of skeletal animals; the bones are rigid while the joints are elastic, and we allow for the creation of additional constraints as desired.

Appendix A: Constrained Rigid Body Dynamics

We use a standard method for modelling and simulating a constrained system of rigid bodies (for an explanation in depth, see [Fea87] or [MLS94]).We place the local frame for body a at the body's center of mass and with axes aligned with the principle axes of inertia. The homogeneous coordi-

nates of frame a with respect to frame b is given by the 4×4 matrix

$$\,_{a}^{b}\mathsf{E} = \begin{pmatrix} \Theta & p \\ 0 & 1 \end{pmatrix},$$

where Θ is a 3×3 rotation matrix, and p is a 3×1 displacement. We write the spatial velocity of body a with respect to the world frame W as a column vector $\,^{a}\phi_{a-W} = (\omega^{T}, v^{T})^{T}$, where ω is the angular velocity and v is the linear velocity. The leading superscript denotes that the vector is expressed in the coordinates of body a. Spatial forces, called wrenches, are written $\,^{a}w = (\tau^{T}, f^{T})^{T}$ where τ is the rotational torque and f is the translational force. Spatial velocities transform according to the adjoint transformation $\,_{a}^{b}\mathsf{A}d$, wrenches transform with the inverse transpose, i.e., $\,^{b}w = \,_{b}^{a}\mathsf{A}d^{T}\,^{a}w$. The 6×6 adjoint matrix is defined by

$$\,_{a}^{b}\mathsf{A}d = \begin{pmatrix} \Theta & 0 \\ [p]\Theta & \Theta \end{pmatrix},$$

where $[p]$ denotes the skew symmetric 3×3 matrix equivalent to the cross product $p\times$.

The motion of a set of rigid bodies is given by the Newton-Euler equation

$$M\dot{\phi} = w_{\phi} + w. \qquad (5)$$

Here M is a diagonal mass matrix assembled for all bodies, and we let ϕ and w (without leading superscripts) refer to the block colum vectors containing velocities and wrenches of all bodies. Since we express these quantities in the moving body frames, we must include the wrench w_{ϕ} due to the current spatial velocities. Lastly, w represents wrenches due to gravity, muscle stiffness, and damping.

Following [BW98], we discretize the system by replacing the acceleration $\dot{\phi}$ with the approximation $(\phi^{t+h} - \phi^{t})/h$ and solve directly for the spatial velocities after a small time step h. Stiff forces in w are made implicit by replacing w with a Taylor approximation of its value at the next time step and moving $T\phi^{t+h}$, the terms involving ϕ^{t+h}, to the left hand side.

$$(M - hT)\phi^{t+h} = h(w_{\phi} + w) + M\phi^{t}. \qquad (6)$$

Constraints

The bones in the skeleton of our animal models must stay connected and can only rotate about certain axes. These constraints change Equation 5 into a differential algebraic equation with a constraint for each joint, $g_{j} = 0$ (a function of the current state). We use a standard technique for solving the constrained equation [AP98]; we differentiate the constraint once to turn it into a constraint on the velocity, and then we stabilize the solutions to prevent numerical drift.

We define a frame j for each joint, located at the joint, and with the x axis purposely aligned with the free rotation for rotary joints. The velocity constraint for joint j connecting

body a and body b has a very simple form when the relative velocity of a and b is written in the coordinates of the frame j. For a single axis rotary joint, it states that the velocity must be zero in all directions exept for rotation about the x axis, i.e.,

$$I_{2:6}\,^{j}\phi_{a-b} = 0, \qquad (7)$$

where $I_{2:6}$ is the 5×6 matrix consisting of the bottom 5 rows of the size 6 identity matrix and $\,^{j}\phi_{a-b} = \,_{a}^{j}\mathsf{A}d\,^{a}\phi_{a-W} - \,_{b}^{j}\mathsf{A}d\,^{b}\phi_{b-W}$. For a spherical joint we use $I_{4:6}$. Written in matrix form the constraints require $G\phi$ to be zero, where the sparse matrix G has 3 rows for every ball joint, and 5 rows for every hinge joint. To solve for the new constrained spatial velocities, we solve the Lagrangian,

$$\begin{pmatrix} (M - hT) & G^{T} \\ G & 0 \end{pmatrix} \begin{pmatrix} \phi^{t+1} \\ \lambda \end{pmatrix} = \begin{pmatrix} h(w + w_{\phi}) + M\phi^{t} \\ b \end{pmatrix}, \qquad (8)$$

where b provides Baumgarte stabilization.

Ground contacts and joint limits are unilateral constraints. While these constraints could be properly accounted for in simulations by setting up a linear complementarity problem, we find that intermittent stiff implicit penalty forces are sufficient for our virtual animals. For modal analysis, however, we can compute modes involving ground contacts by including a bilateral constraint for the contacts.

Compliant Joints

The muscles and tendons of an animal produce torques at its joints. We model this with spring and damping torques. It is the inclusion of stiffness in our constrained rigid bodies that allows elastic oscillations in the system, for which we use modal analysis to determine which vibrations are the most important.

The torque at a compliant single axis joint with stiffness k_{j} is $\,^{j}\tau_{jx} = k_{j}(\theta_{j} - \theta_{j0})$, where θ_{j} is the current joint angle and θ_{j0} is the desired angle. The torque acting upon each of the connected bodies is equal but opposite: $\,^{a}w_{k} = \,_{a}^{j}\mathsf{A}d^{T}I_{1:1}^{T}\,^{j}\tau_{jx}$ and $\,^{b}w_{k} = -\,_{b}^{j}\mathsf{A}d^{T}I_{1:1}^{T}\,^{j}\tau_{jx}$. Here the 1×6 matrix $I_{1:1}$ is the first row of the identity matrix and corresponds with the degrees of freedom that were not constrained in Equation 7. In a similar manner to G, we construct a matrix R which allows the spring wrenches to be written in matrix form, $w_{k} = R^{T}K(\theta - \theta_{0})$. Here K is the diagonal stiffness matrix containing the stiffness of each joint. Note that R also transforms the spatial velocities of the bodies to angular velocities at the joints, $\dot{\theta} = R\phi$. Thus, damping wrenches can be computed as $w_{c} = R^{T}CR\phi$, where C is the diagonal matrix of damping coefficients.

References

[Ale88] ALEXANDER R. M.: *Elastic Mechanisms in Animal Movement*. Cambridge University Press, 1988.

[Ale96] ALEXANDER R. M.: *Optima for Animals*. Princeton University Press, 1996.

[AP98] ASCHER U. M., PETZOLD L. R.: *Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1998.

[BHMIG92] BIZZI E., HOGAN N., MUSSA-IVALDI F. A., GISZTER A.: Does the nervous system use equilibrium-point control to guide single and multiple joint movements? *Behavioral and Brain Sciences 15* (1992), 603–613.

[BW98] BARAFF D., WITKIN A.: Large steps in cloth simulation. In *SIGGRAPH '98: Proceedings of the 25th annual conference on Computer graphics and interactive techniques* (1998), pp. 43–54.

[CRTW05] COLLINS S., RUINA A., TEDRAKE R., WISSE M.: Efficient bipedal robots based on passive-dynamic walkers. *Science 307* (2005), 1082–1085.

[CWR01] COLLINS S. H., WISSE M., RUINA A.: A three-dimensional passive-dynamic walking robot with two legs and lees. *The International Journal of Robotics Research 20*, 2 (2001), 607–615.

[Fea87] FEATHERSTONE R.: *Robot dynamics algorithms*. Kluwer, 1987.

[Fel86] FELDMAN A. G.: Once more on the equilibrium-point hypothesis (lambda model) for motor control. *Journal of Motor Behavior 18*, 1 (March 1986), 17–54.

[FRDC04] FAVREAU L., REVERET L., DEPRAZ C., CANI M.-P.: Animal gaits from video. In *SCA '04: Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation* (2004), pp. 277–286.

[FvdPT97] FALOUTSOS P., VAN DE PANNE M., TERZOPOULOS D.: Dynamic free-form deformations for animation synthesis. *IEEE Transactions on Visualization and Computer Graphics 3*, 3 (September 1997), 201–214.

[GT95] GRZESZCZUK R., TERZOPOULOS D.: Automated learning of muscle-actuated locomotion through control abstraction. In *SIGGRAPH '95: Proceedings of the 22nd annual conference on Computer graphics and interactive techniques* (1995), pp. 63–70.

[GTH98] GRZESZCZUK R., TERZOPOULOS D., HINTON G.: Neuroanimator: Fast neural network emulation and control of physics-based models. In *Proceedings of SIGGRAPH 98* (1998), pp. 9–20.

[HP97] HODGINS J. K., POLLARD N. S.: Adapting simulated behaviors for new characters. In *SIGGRAPH '97: Proceedings of the 24th annual conference on Computer graphics and interactive techniques* (1997), pp. 153–162.

[HRE*08] HECKER C., RAABE B., ENSLOW R. W., DEWEESE J., MAYNARD J., VAN PROOIJEN K.: Real-time motion retargeting to highly varied user-created morphologies. In *SIGGRAPH '08: ACM SIGGRAPH 2008 papers* (New York, NY, USA, 2008), ACM, pp. 1–11.

[HWBO95] HODGINS J. K., WOOTEN W. L., BROGAN D. C., O'BRIEN J. F.: Animating human athletics. In *SIGGRAPH '95: Proceedings of the 22nd annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 1995), ACM Press, pp. 71–78.

[Ijs08] IJSPEERT A. J.: Central pattern generators for locomotion control in animals and robots: a review. *Neural Networks 21*, 4 (2008), 642–653.

[Kuo02] KUO A. D.: Energetics of actively powered locomotion using the simplest walking model. *Journal of Biomechanical Engineering*, 124 (2002), 113–120.

[LHP05] LIU C. K., HERTZMANN A., POPOVIC Z.: Learning physics-based motion style with nonlinear inverse optimization. *ACM Transactions on Graphics 24*, 3 (2005), 1071–1081.

[LP02] LIU C. K., POPOVIC Z.: Synthesis of complex dynamic character motion from simple animations. In *SIGGRAPH '02: Proceedings of the 29th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 2002), ACM Press, pp. 408–416.

[MA90] MCGEER T., ALEXANDER R.: Passive bipedal running. *Proceedings of the Royal Society of London, Series B, Biological Sciences 240* (1990).

[McG90] MCGEER T.: Passive dynamic walking. *International Journal of Robotics Research 9*, 2 (1990), 62–82.

[MLS94] MURRAY R., LI Z., SASTRY S. S.: *A mathematical introduction to robotic manipulation*. CRC Press, 1994.

[MZ90] MCKENNA M., ZELTZER D.: Dynamic simulation of autonomous legged locomotion. In *SIGGRAPH '90: Proceedings of the 17th annual conference on Computer graphics and interactive techniques* (1990), pp. 29–38.

[NF02] NEFF M., FIUME E.: Modeling tension and relaxation for computer animation. In *SCA '02: Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation* (2002), pp. 81–88.

[PW89] PENTLAND A., WILLIAMS J.: Good vibrations: modal dynamics for graphics and animation. In *SIGGRAPH '89* (1989), ACM Press, pp. 215–222.

[RH91] RAIBERT M. H., HODGINS J. K.: Animation of dynamic legged locomotion. In *SIGGRAPH '91: Proceedings of the 18th annual conference on Computer graphics and interactive techniques* (1991), pp. 349–358.

[Sha97] SHABANA A. A.: *Vibration of Discrete and Continuous Systems*. Springer, 1997.

[SHP04] SAFONOVA A., HODGINS J. K., POLLARD N. S.: Synthesizing physically realistic human motion in low-dimensional, behavior-specific spaces. In *ACM Transactions on Graphics 23(3), SIGGRAPH 2004* (2004).

[Sim94] SIMS K.: Evolving virtual creatures. In *SIGGRAPH '94: Proceedings of the 21st annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 1994), ACM Press, pp. 15–22.

[SM01] SUN H. C., METAXAS D. N.: Automating gait generation. In *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques* (2001), pp. 261–270.

[Sta97] STAM J.: Stochastic dynamics: Simulating the effects of turbulence on flexible structures. *Computer Graphics Forum 16*, 3 (1997).

[Tag95] TAGA G.: A model of the neuro-musculo-skeletal system for human locomotion. *Biological Cybernetics*, 73 (1995), 97–1995.

[TvdP98] TORKOS N., VAN DE PANNE M.: Footprint–based quadruped motion synthesis. In *Graphics Interface* (1998), pp. 151–160.

[vdP96] VAN DE PANNE M.: Parameterized gait synthesis. *IEEE Computer Graphics and Applications. 16*, 2 (1996), 40–49.

[vF93] VAN DE PANNE M., FIUME E.: Sensor-actutor networks. In *Computer Graphics Proceedings, ACM SIGGRAPH, Proceedings of SIGGRAPH '93* (1993), pp. 335–342.

[vKF94] VAN DE PANNE M., KIM R., FIUME E.: Virtual wind-up toys for animation. In *Proceedings of Graphics Interface '94* (Banff, Alberta, Canada, 1994), pp. 208–215.

[Wil97] WILHELMS J.: Animals with anatomy. *IEEE Comput. Graph. Appl. 17*, 3 (1997), 22–30.

[WP03] WU J.-C., POPOVIC Z.: Realistic modeling of bird flight animations. *ACM Trans. Graph. 22*, 3 (2003), 888–895.