



# Kinematic gait synthesis for snake robots

Chaohui Gong<sup>1</sup>, Matthew J. Travers<sup>1</sup>, Henry C. Astley<sup>2</sup>, Lu Li<sup>1</sup>,  
Joseph R. Mendelson<sup>3,4</sup>, Daniel I. Goldman<sup>2</sup> and Howie Choset<sup>1</sup>

## Abstract

Snake robots are highly articulated mechanisms that can perform a variety of motions that conventional robots cannot. Despite many demonstrated successes of snake robots, these mechanisms have not been able to achieve the agility displayed by their biological counterparts. We suggest that studying how biological snakes coordinate whole-body motion to achieve agile behaviors can help improve the performance of snake robots. The foundation of this work is based on the hypothesis that, for snake locomotion that is approximately kinematic, replaying parameterized shape trajectory data collected from biological snakes can generate equivalent motions in snake robots. To test this hypothesis, we collected shape trajectory data from sidewinder rattlesnakes executing a variety of different behaviors. We then analyze the shape trajectory data in a concise and meaningful way by using a new algorithm, called conditioned basis array factorization, which projects high-dimensional data arrays onto a low-dimensional representation. The low-dimensional representation of the recorded snake motion is able to reproduce the essential features of the recorded biological snake motion on a snake robot, leading to improved agility and maneuverability, confirming our hypothesis. This parameterized representation allows us to search the low-dimensional parameter space to generate behaviors that further improve the performance of snake robots.

## Keywords

Bio-inspired robotics, modal decomposition, limbless locomotion

## 1. Introduction

Snakes show versatile locomotion, allowing them to move effectively in a wide range of challenging habitats. In particular, their limbless body plan allows excellent mobility in cluttered, complex habitats which thwart limbed organisms. Consequently, there has been substantial interest in emulating these advantages in snake robots (Hirose, 1993; Liljebäck et al., 2012; Transth et al., 2009; Yamada and Hirose, 2006). However, the hundreds of vertebrae of the snake body plan results in a very high number of degrees of freedom, which pose significant challenges to control and computation. Various computational techniques, including machine learning (Tesch et al., 2011) and central pattern generators (CPGs) (Ijspeert, 2008), have been proposed for designing useful motions for snake robots, including lateral undulation (Mori and Hirose, 2001), sinus lifting (Tanaka and Matsuno, 2009), sidewinding (Marvi et al., 2014) and climbing (Tesch et al., 2009). However, these capabilities are still relatively simple and do not exploit the true potential of snake robots or reflect the true diversity of snake behaviors. This paper presents an algorithm that reduces the complexity of snake motion so that we can prescribe simple control and planning for snake robots.

Some snake locomotion is approximately kinematic because their motion is dominated by friction (Hu et al., 2009), and thus purely governed by how their shapes change over time (Hatton and Choset, 2013; Ostrowski and Burdick, 1998; Shamma, 2006). It is thus possible to derive from snake shape trajectory data a controller that can produce snake-like behaviors in a snake robot, and consequently improve the performance of the robot. Doing so requires not only observation of the motion of biological snakes, but also a suitable computational method to identify the essential features of this data and apply those features to the robot.

<sup>1</sup>Robotics Institute, Carnegie Mellon University, USA

<sup>2</sup>School of Physics, Georgia Institute of Technology, USA

<sup>3</sup>School of Biology, Georgia Institute of Technology, USA

<sup>4</sup>Department of Herpetology, Zoo Atlanta, USA

### Corresponding author:

Chaohui Gong, Robotics Institute, Carnegie Mellon University, 5000 Forbes, Newell Simon Hall, PA 15213, USA.

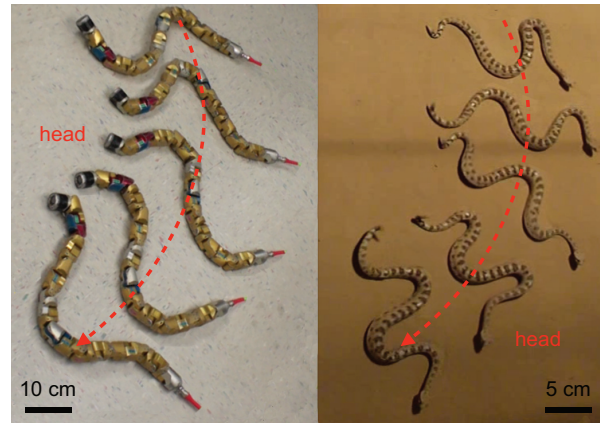
Email: chaohuig@cmu.edu

Although snake locomotion can be directly observed using a vision system, their complex and nearly continuous changes in body shape over time pose challenges in analyzing snake kinematics. Shape trajectory data collected from biological snakes are high-dimensional (Hatton and Choset, 2010), which makes identifying the underlying principles of locomotion from biological data difficult. To simplify the representation and analysis of high-dimensional data, modal decomposition (dimensionality reduction) techniques have been previously proposed (Stephens et al., 2008; Twining and Taylor, 2001). For example, principal component analysis (PCA) has been successfully applied to simplify the representation of shape data collected from nematodes (Stephens et al., 2008). Unfortunately, previously proposed techniques can only simplify the representation of instantaneous shapes, and analyzing how animals change shape over time remains a challenge. Furthermore, most of the previous studies have focused on the motion of snakes in the horizontal plane, with only a few exceptions studying three-dimensional (3D) snake undulations (Astley et al., 2015; Socha, 2002). Recent studies have revealed that the emergence of agile turns during sidewinding (technically, we know little/nothing about turns during other modes) is closely related to their ability to undulate in the vertical plane (Astley et al., 2015). Thus, the vertical component of 3D snake undulation cannot be neglected.

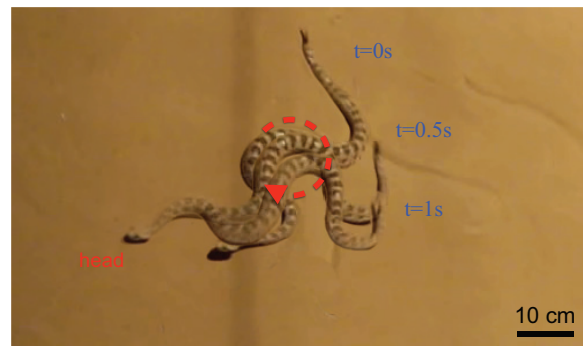
Our approach decomposes 3D biological snake undulation into two waves, one in the horizontal, and one in the vertical plane. Three-dimensional snake undulations are then represented using two sets of shape trajectory data (in the horizontal and vertical planes, respectively). To study high-dimensional snake shape trajectory data in a concise and intuitive way, we previously introduced *conditioned basis array factorization* (CBAF) (Gong et al., 2014), a modal decomposition technique that projects high-dimensional arrays onto a low-dimensional parameterized representation. Unlike previously proposed modal decomposition techniques, CBAF not only simplifies the representation of instantaneous snake shapes, but also simplifies the representation of how shapes change over time. By applying CBAF to shape trajectory data collected from sidewinders executing a variety of different behaviors, we achieve a low-dimensional parameterized representation of recorded snake motion. Implementing this low-dimensional representation on snake robots reproduces the essential features of the recorded biological motion. By exploring the small space of the parameters, we are able to synthesize novel behaviors, which significantly improve the performance of snake robots beyond simple replication of biological behaviors.

## 2. Collecting data from biological snakes

Using their many internal degrees of freedom, sidewinder rattlesnakes, *Crotalus cerastes*, can use their whole body undulations to achieve various modes of locomotion. Two different types of turning behavior, *differential turning*, see



**Fig. 1.** A snake robot (on hard ground) and a rattlesnake, *C. cerastes*, (on sand) executing differential turning behaviors. The red dashed arrows denote the direction of motion.

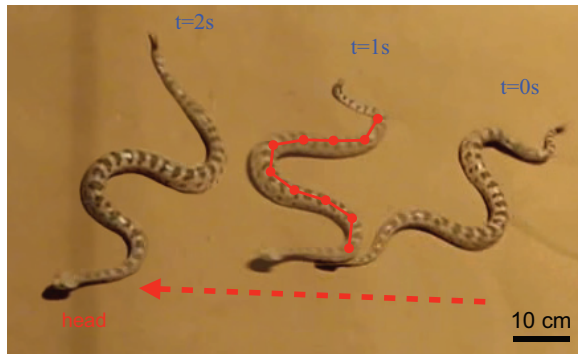


**Fig. 2.** A sidewinder rattlesnake, *C. cerastes*, performing the re-orientation portion of a reversal turn on level sand. The red dashed arrow denotes the direction of motion during the re-orientation stage.

Figure 1 (see also Extension 1), and *reversal turning*, see Figure 2 (see also Extension 2), have been observed for these rattlesnakes during sidewinding (Astley et al., 2015). *Differential turning* results in gradual turns accompanied by forward motion which can continue for many gait cycles, while *reversal turning* results in a rapid change of orientation with a very small turn radius. During a reversal turn, sidewinders come to a stop, perform a re-orienting behavior of variable magnitude, and resume moving with a vertical wave phase shifted by  $\pi$ . Reversal turning presents an exceptional level of agility, superior to any gait that has been previously implemented on snake robots (Gong et al., 2012, 2013; Tanev et al., 2005; Tesch et al., 2009). To systematically study how snakes modulate their body undulations to achieve these agile motions and for later emulating them on snake robots, we used a motion capture system to record the trajectories of sidewinding.

### 2.1. Biological snakes and vision system

The research subjects were four adult sidewinder rattlesnakes, *C. cerastes* ( $m = 98 \pm 18$  g,  $L = 48 \pm 6$  cm),



**Fig. 3.** Computing the virtual joint angles of a sidewinding rattlesnake. The red dots denote the positions of the markers pasted on the back of the snake. The virtual joint angles are the relative angles between the straight segments. The red dashed arrow denotes the direction of motion on sand.

collected near Yuma, AZ (under approval GT IACUC A11053). The snakes were placed in a 1 m  $\times$  2 m fluidization bed, filled with natural sand from Yuma, AZ and encouraged to display natural turning behaviors by body motions of the handlers. Four OptiTrack Flex 13 cameras (Natural Point, Inc.) operating at 120 FPS, fixed above the test bed, were used to track the 3D positions of 10 reflective markers pasted along the backs of the snakes, see Figure 3. The snakes were marked from the tail tips (the base of the rattle) to the first quarter. Although the anterior portion of the body was involved in the locomotion to some degree, we were not able to further mark the snakes due to safety concerns. Before each trial, the sand was fluidized to a loosely packed state to ensure consistent initial conditions of the test environment over all animal trials (Maladen et al., 2009).

The resulting trial data was manually segmented into 21 ( $G = 21$ ) sequences, each consisting of one complete cycle of sidewinding. Because the biological snakes moved at varying speeds, each video sequence may contain a different number of frames. We subsampled the video sequences to ensure each sequence contained  $T = 128$  frames.

## 2.2. Undulations in orthogonal planes

During sidewinding, the snake undulates in both the horizontal and vertical planes (Jayne, 1986; Marvi et al., 2014; Mosauer, 1930). In Gong et al. (2014), we assumed horizontal and vertical body undulations were coupled by a constant phase offset. However, recent experiments revealed that sidewinder rattlesnakes can regulate vertical undulations independent of their motion in the horizontal plane, and disparate motions emerged from altering the vertical motion while the horizontal motion remained the same (Astley et al., 2015). To construct a more complete representation of biological gaits and to capture a wider spectrum of locomotor behaviors, we modeled sidewinding as two independent body waves that travel posteriorly along the snake's body. Decoupling horizontal and vertical body

undulations simplified the analysis of snake kinematics in 3D. However, whether and to what extent the biological snakes independently actuated their bodies in the horizontal and vertical directions is beyond the scope of this study.

While snake body undulations in the horizontal plane were directly observed by the vision system, vertical undulations were challenging to discern because sidewinder rattlesnakes move with a small ground clearance ( $\leq 0.002$  m). Instead of measuring vertical movement directly using the vision system, we inferred vertical motion from their motion in the horizontal plane based on the key property that sidewinder rattlesnakes locomote via rolling-type contacts that do not slip, as evidenced by distinct, un-smearred scale imprints left in every track (Jayne, 1986; Mosauer, 1930). This slip-resistance property of sidewinding means that body segments in contact with the substrate (markers that have zero  $z$ -displacement) had zero velocity with respect to the ground. By making a simplifying assumption that the vertical displacement of a reflective marker was proportional to its instantaneous velocity, we interpolated the body shapes of the rattlesnakes in the vertical plane. The ratio of the vertical displacement to the instantaneous velocity was hand-tuned to ensure that the amplitude of the vertical waves remained small, but was able to generate sufficient vertical lifting motion on the robots.

We collected snake-shape trajectory data independently in the horizontal and vertical planes. In every video frame, the snake was first represented as being composed of 9 rigid links (in the horizontal and vertical planes, respectively) whose configurations were determined by the positions of the 10 reflective markers pasted on the snakes, as shown in Figure 3. We then computed the *virtual joint angles* (a discrete way to represent the continuous body curvature), respectively in the horizontal and vertical planes, as the  $N = 8$  relative angles between every pair of neighboring links. Therefore, shape trajectory data in every video sequence were eventually represented as two  $N \times T$  ( $N = 8$ ,  $T = 128$ ) tables, denoting the virtual joint angles sampled at different positions along the snakes (*spatial direction*) in different video frames (*temporal direction*).

## 2.3. Data as third-order arrays

The goal of this work is to identify a parameterized low-dimensional representation of the recorded biological snake motions, which not only parameterizes the shape trajectories of individual behaviors but also parameterizes the change in body undulations across different behaviors. To parameterize how shape trajectory data varied across different snake behaviors, we stacked all the  $N \times T$  tables, respectively in the horizontal and vertical planes, into two third-order data arrays of dimension  $N \times T \times G$ , where  $G$  denotes the number of animal trials. These two third-order data arrays represent the curvature information sampled at different positions along the snakes (*spatial direction*), at different phases in a gait cycle (*temporal direction*) and from different snake behaviors (*behavioral direction*).

### 3. Conditioned basis array factorization

To identify a low-dimensional parameterized representation of high-order data arrays, we developed a modal decomposition technique, called CBAF. CBAF can be directly applied to high-order data arrays to achieve a concise analytical representation. While the CBAF algorithm primarily builds on existing array factorization techniques (Akhter et al., 2012; De Lathauwer et al., 2000a), using factorization techniques to decompose recorded snake motions is primarily inspired by the *modal approach* (Chirikjian and Burdick, 1994), which models the shape of hyper-redundant mechanisms as the combination of a finite set of shape bases. The CBAF algorithm generalizes this idea by modeling snake locomotion as the linear combination of a few spatiotemporal modes.

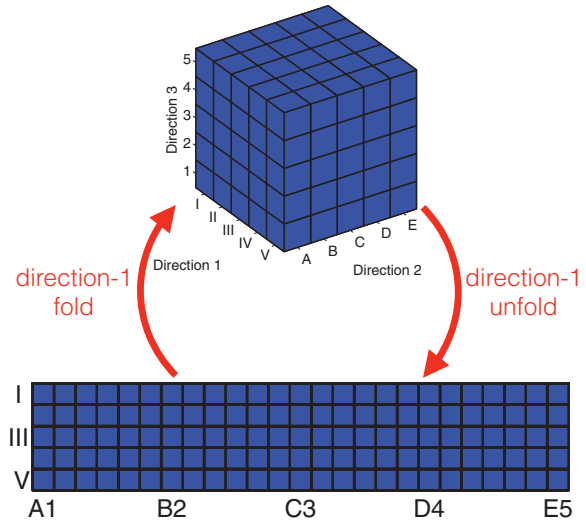
#### 3.1. Array factorization

Modal decomposition techniques are frequently used to simplify the analysis of high-dimensional data (Stephens et al., 2008). For example, PCA has been successfully applied in gait recognition (Wang et al., 2003), textual information retrieval (Berry et al., 1995), facial recognition (Zhao et al., 1998), as well as image classification (Quelhas et al., 2005). In spite of its ease in implementation, the requirement that data must be in the form of first-order arrays (vectors) restricts its utility. High-order data arrays must be vectorized before applying PCA, which inevitably breaks the structure (and probably the associated information) of the high-order data arrays and also increases the risk of over-fitting (Akhter et al., 2012). A bilinear model (Lee and Elgammal, 2004), which could factorize second-order data arrays (matrices), was previously proposed to separate content and style from human gait data.

With respect to the multidimensional structures of high-order data arrays, various array factorization techniques (Kolda and Bader, 2009) have been developed. Array factorization techniques have increasing importance in many applications including computer vision (Vasilescu and Terzopoulos, 2002), data mining (Acar et al., 2006) and graph analysis (Kolda et al., 2005). Among these array factorization techniques, higher order singular value decomposition (HOSVD) (De Lathauwer et al., 2000a) and alternating least-square (ALS) (Krooneberg, 1983; Kroonenberg and De Leeuw, 1980) are two widely accepted algorithms. However, their iterative procedures and reliance on *singular value decomposition* (SVD) make them computationally unattractive. More importantly, these numerical algorithms are incapable of generating analytical models for gaining insight into animal locomotion.

#### 3.2. Mathematical preliminaries

The limitations associated with existing array factorization techniques motivated us to develop the CBAF algorithm, which uses analytical basis functions to reduce the



**Fig. 4.** An illustration of the mathematical operators, direction-1 folding and direction-1 unfolding. Direction-1 unfolding collects all the direction-1 vectors (vectors parallel to the direction-1 axis in the cube shown above) into a matrix. Direction-1 fold reshapes a matrix back into a high-order array.

dimensionality of high-order data arrays. We here introduce some basics of array factorization, and refer the reader to Kolda and Bader (2009) for further information. A data array  $\mathcal{A} \in \mathbb{R}^{I_1 \times \dots \times I_N}$  is referred to as an  $N$ th order data array of dimension  $I_1 \times \dots \times I_N$ . The joint trajectory data of the biological snakes were represented as two third-order data arrays of dimension  $8 \times 128 \times 21$  ( $N = 8$ ,  $T = 128$ ,  $G = 21$ ), where  $N$ ,  $T$  and  $G$  are the dimensions in the spatial, temporal and behavioral directions.

**Definition.** The *direction- $k$  unfolding* of an array  $\mathcal{A} \in \mathbb{R}^{I_1 \times \dots \times I_N}$  is a matrix denoted as  $\text{unfold}_k(\mathcal{A}) = A_{(k)} \in \mathbb{R}^{I_k \times (I_1 \dots I_{k-1} I_{k+1} \dots I_N)}$ . The column vectors of  $A_{(k)}$  are called the *direction- $k$  vectors* whose elements are  $a_{i_1, \dots, i_N}$  with  $i_1, \dots, i_{k-1}, i_{k+1}, \dots, i_N$  being fixed and  $i_k = 1, 2, \dots, I_k$ .

For example, the column vectors of a second-order array (matrix) are the direction-1 vectors and the row vectors are the direction-2 vectors. As illustrated in Figure 4, the unfolding operator re-organizes the original data array into a collection of direction- $k$  vectors, which can be interpreted as the collection of trajectories along direction- $k$  (space, time or behaviors in the case of snake data). After unfolding a high-order array, techniques like PCA can be applied to reduce the dimensionality in direction- $k$ .

**Definition.** The *direction- $k$  folding* is the inverse operation of direction- $k$  unfolding, denoted as  $\text{fold}_k(A_{(k)})$ , which reshapes a matrix  $A_{(k)} \in \mathbb{R}^{I_k \times (I_1 \dots I_{k-1} I_{k+1} \dots I_N)}$  into a high-order array  $\mathcal{A} \in \mathbb{R}^{I_1 \times \dots \times I_N}$ .

The direction- $k$  dimensionality of an array can be reduced through an operation called *direction- $k$  multiplication*.

**Definition.** The *direction- $k$  multiplication* between a high-order array  $\mathcal{A} \in \mathbb{R}^{I_1 \times \dots \times I_N}$  and a matrix  $U \in \mathbb{R}^{I_k \times R_k}$  is denoted as  $\mathcal{D} = \mathcal{A} \times_k U$ , and  $\mathcal{D} \in \mathbb{R}^{I_1 \times \dots \times I_{k-1} \times R_k \times I_{k+1} \times \dots \times I_N}$ , whose elements are computed as

$$d_{i_1 \dots i_{k-1}, l, i_{k+1} \dots i_N} = \sum_{j=1}^{I_k} (u_{j,l}) (a_{i_1 \dots i_{k-1}, j, i_{k+1} \dots i_N})$$

Direction- $k$  multiplication  $\mathcal{A} \times_k U$  is equivalently represented as  $\text{fold}_k(U^T \mathcal{A}_{(k)})$ , where  $'$  denotes the matrix transpose. Intuitively, direction- $k$  multiplication first projects all the direction- $k$  vectors onto a vector space spanned by the column vectors of  $U$ , and then re-shapes the resultant matrix back into a high-order array. If  $R_k < I_k$ , the direction- $k$  dimensionality of the resultant array is reduced from  $I_k$  to  $R_k$ .

**Definition.** The *inner product* between two high-order arrays  $\mathcal{A} \in \mathbb{R}^{I_1 \times \dots \times I_N}$  and  $\mathcal{B} \in \mathbb{R}^{I_1 \times \dots \times I_N}$  is defined as

$$\langle \mathcal{A}, \mathcal{B} \rangle = \langle \mathcal{B}, \mathcal{A} \rangle = \sum_{i_1}^{I_1} \dots \sum_{i_N}^{I_N} a_{i_1 \dots i_N} b_{i_1 \dots i_N}$$

**Definition.** The high-order arrays  $\mathcal{A}$  and  $\mathcal{B}$  are called *orthogonal* if  $\langle \mathcal{A}, \mathcal{B} \rangle = 0$ .

**Definition.** The *norm* of a high-order array  $\mathcal{A}$  is

$$\|\mathcal{A}\| = \langle \mathcal{A}, \mathcal{A} \rangle^{\frac{1}{2}} \quad (1)$$

**Definition.** The *tensor product* of two high-order arrays  $\mathcal{A} \in \mathbb{R}^{I_1 \times \dots \times I_p}$  and  $\mathcal{B} \in \mathbb{R}^{J_1 \times \dots \times J_q}$  is denoted as  $\mathcal{D} = \mathcal{A} \otimes \mathcal{B}$ , whose elements are computed as

$$d_{i_1 \dots i_p, j_1 \dots j_q} = a_{i_1 \dots i_p} b_{j_1 \dots j_q} \quad (2)$$

A tensor product is the generalization of the outer product between vectors, which can be used to construct a high-order array given basis vectors along each direction.

**Definition.** A high-order array formed by the tensor product of  $N$  non-zero basis vectors (one along each direction)  $\vec{u}^{(1)} \in \mathbb{R}^{I_1}, \dots, \vec{u}^{(N)} \in \mathbb{R}^{I_N}$  is called an  $N$ th order *rank-1 array*  $\mathcal{A} = \vec{u}^{(1)} \otimes \vec{u}^{(2)} \otimes \dots \otimes \vec{u}^{(N)}$ .

Rank-1 arrays are the fundamental building blocks of high-order arrays and any array can be represented by the linear combination of a set of rank-1 arrays.

**Definition.** Let a matrix  $U^{(k)} \in \mathbb{R}^{I_k \times R_k}$  be the *direction- $k$  basis matrix* if it has  $R_k$  mutually orthogonal, unitary column vectors denoted by  $\vec{u}_{i_k}^{(k)} \in \mathbb{R}^{I_k}$ . If  $R_k < I_k$ , we call it a *direction- $k$  truncated basis matrix*.

The column vectors of  $U^{(k)}$  are termed *the basis vectors in direction- $k$* . An analytical basis function can be represented as an  $I_k$ -dimensional vector, whose elements are the values sampled from the basis function at  $I_k$  discrete points.

With all the definitions above, we show the key properties of high-order arrays.

**Theorem 3.1.** (Chen and Saad, 2009). Given  $R$  mutually orthogonal, unitary rank-1 arrays  $\mathcal{T}_i$ , the optimal approximation to the array  $\mathcal{A}$  is the linear combination  $\mathcal{T} = \sum_{i=1}^R \sigma_i \mathcal{T}_i$  such that  $\|\mathcal{A} - \mathcal{T}\|$  is minimized. The optimal weights are computed by the inner product,  $\sigma_i = \langle \mathcal{A}, \mathcal{T}_i \rangle$ . The difference between the original array and the optimal reconstruction  $\mathcal{A} - \mathcal{T}$  is orthogonal to every rank-1 array,  $\langle \mathcal{A} - \mathcal{T}, \mathcal{T}_i \rangle = 0$ . The squared reconstruction error is computed as

$$\|\mathcal{A} - \mathcal{T}\|^2 = \|\mathcal{A}\|^2 - \|\mathcal{T}\|^2 \quad (3)$$

Theorem 3.1 provides theoretical guidance to compute the optimal reconstruction given a set of rank-1 arrays.

**Theorem 3.2.** (Chen and Saad, 2009). Given (truncated) basis matrices along each direction,  $\mathbf{V}^{(1)} \in \mathbb{R}^{I_1 \times R_1}, \dots, \mathbf{V}^{(N)} \in \mathbb{R}^{I_N \times R_N}$ , the array

$$\mathcal{C} = \mathcal{A} \times_1 \mathbf{V}^{(1)} \times_2 \dots \times_N \mathbf{V}^{(N)} \quad (4)$$

is called the (truncated) core array. And the best rank- $(\mathbf{R}_1, \dots, \mathbf{R}_N)$  approximation to the original array is

$$\hat{\mathcal{A}} = \mathcal{C} \times_1 (\mathbf{V}^{(1)})^T \times_2 \dots \times_N (\mathbf{V}^{(N)})^T \quad (5)$$

Theorem 3.2 states how to find the optimal low-dimensional approximation to an array given basis matrices along all directions. The core array represents the projection of  $\mathcal{A}$  onto a lower dimensional space spanned by the rank-1 arrays constructed from the basis matrices,  $\mathbf{V}^{(1)}, \dots, \mathbf{V}^{(N)}$ .

**Theorem 3.3.** (De Lathauwer et al., 2000b). Minimizing the reconstruction error  $\|\mathcal{A} - \hat{\mathcal{A}}\|$  is equivalent to maximizing  $\|\mathcal{C}\|$ .

### 3.3. Factorization with conditioned bases

We refer to a set of pre-determined basis functions (basis matrix) the *conditioned bases* (Akhter et al., 2012) and refer to array factorization techniques that use conditioned bases CBAF. Conditioned bases may either be chosen based on domain knowledge or based on the evaluation of different candidate bases. Incorporating conditioned bases into the process of factorization immediately mitigates the computational disadvantages of HOSVD and ALS, and using basis functions from domain knowledge facilitates the discovery of low-dimensional representations (models) that have clear physical meanings.

Using conditioned bases reduces array factorization from a basis fitting problem to a basis selection problem (choosing a subset of the column vectors from the conditioned basis matrix), which can be solved efficiently. Without loss of generality, we here restrict our discussion to third-order arrays, and we remark that all of the presented proofs can be easily generalized to high-order arrays. Basis

selection aims to minimize the reconstruction error  $\|\mathcal{A} - \widehat{\mathcal{A}}\|$  or, equivalently, maximize the norm of the truncated core array  $\|\widehat{\mathcal{C}}\|$ . According to Theorem 3.2, given the conditioned basis matrices  $U^{(1)} \in \mathbb{R}^{I_1 \times I_1}$ ,  $U^{(2)} \in \mathbb{R}^{I_2 \times I_2}$  and  $U^{(3)} \in \mathbb{R}^{I_3 \times I_3}$ , the core array is immediately computed as

$$\mathcal{C} = \mathcal{A} \times_1 U^{(1)} \times_2 U^{(2)} \times_3 U^{(3)} \quad (6)$$

Equation (6) results in an exact representation of the original data array with no reconstruction error. In order to achieve a rank- $(R_1, R_2, R_3)$  approximation,  $R_k$  column vectors are chosen from  $U^{(k)}$ , which results in the truncated basis matrices  $V^{(1)} \in \mathbb{R}^{I_1 \times R_1}$ ,  $V^{(2)} \in \mathbb{R}^{I_2 \times R_2}$  and  $V^{(3)} \in \mathbb{R}^{I_3 \times R_3}$ . The truncated core array is then

$$\widehat{\mathcal{C}} = \mathcal{A} \times_1 V^{(1)} \times_2 V^{(2)} \times_3 V^{(3)} \quad (7)$$

Note the only difference between equations (6) and (7) is that the basis matrices  $V^{(k)}$  in equation (7) are the submatrices of  $U^{(k)}$  in equation (6), and  $U^{(k)}$  is a full rank basis matrix. Equation (6) actually performs a change of coordinates along every direction, while equation (7) projects data into a subspace in this new coordinate system.

Using Theorem 3.3, basis truncation (basis selection) can be alternatively stated as constructing three subsets  $J_1 \subseteq \{1, 2, \dots, I_1\}$ ,  $J_2 \subseteq \{1, 2, \dots, I_2\}$  and  $J_3 \subseteq \{1, 2, \dots, I_3\}$  so that  $|J_1| = R_1$ ,  $|J_2| = R_2$ ,  $|J_3| = R_3$  (here  $|\cdot|$  denotes the cardinality of a set), which maximize

$$\|\widehat{\mathcal{C}}\|^2 = \sum_{i_1 \in J_1} \sum_{i_2 \in J_2} \sum_{i_3 \in J_3} c_{i_1, i_2, i_3}^2 \quad (8)$$

We call this problem *low rank optimal reconstruction* (LROR), which is NP-hard, see the proof in Appendix B. However, CBAF can find a good approximate solution with a computational complexity of  $O(I_1 I_2 I_3)$ . To guide the choice of basis selection in each direction- $k$ , we define a quantity called *energy* which estimates the importance of one particular basis vector in representing the original data array. The energy of a direction- $k$  conditioned basis  $u_{i_k}^{(k)}$  is computed as

$$\mathcal{E}_{i_k}^{(k)} = \sum_{\substack{j_k = i_k \\ j_1, \dots, j_N}} c_{j_1, \dots, j_N}^2 \quad (9)$$

Intuitively, the energy of a basis can be understood as the amount of data variation captured by the basis vector.

Using energy as a heuristic, basis selection is efficiently achieved by CBAF. The pseudo-code of the CBAF algorithm is included in Algorithm 1. The computational complexity of the CBAF is analyzed in Appendix B and the proof of the optimality bound of the CBAF algorithm can also be found in Appendix B. This optimality bound guarantees the high fidelity of the reduced representation with respect to the original high-order data array.

---

**Algorithm 1.** Conditioned basis array factorization.

---

```

1  compute core array:  $\mathcal{C}$ ;
2  for direction  $k = 1$  to  $N$  do
3    for basis  $i = 1$  to  $I_k$  do
4      compute energy:  $\mathcal{E}_{i_k}^{(k)}$ ;
5    end
6    sort basis according to  $\mathcal{E}_{i_k}^{(k)}$  from high to low;
7    chose the first  $R_k$  basis as  $V^{(k)}$ ;
8  end

```

---

### 3.4. Synthetic gait: An example

We use a simple yet representative example to demonstrate the usage of the CBAF algorithm. Before working with real animal data, we synthesized shape trajectory data by sampling the discrete serpenoid curve (Hirose, 1993)

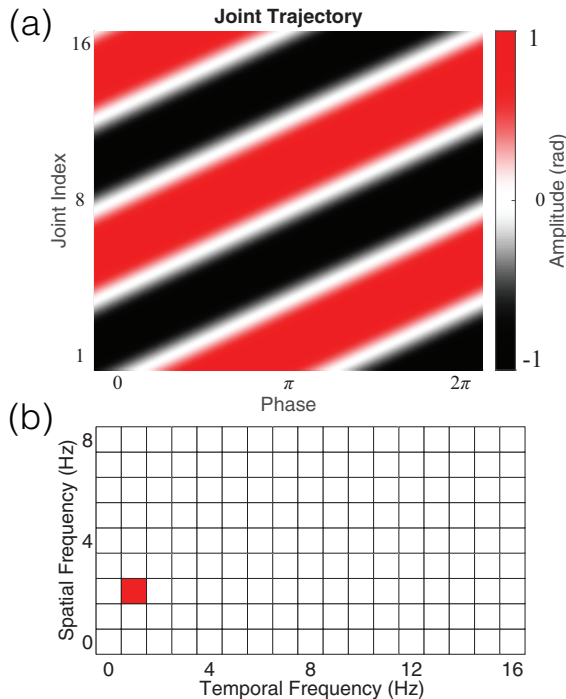
$$\alpha(n, t) = A \sin\left(\frac{4\pi}{N_s} n + 2\pi t\right) \quad (10)$$

on a 3D grid with  $n$  ranging from 1 to 16 ( $\Delta n = 1$ ),  $t$  ranging from 0 to 0.99 ( $\Delta t = 0.01$ ) and  $A$  ranging from 0 to 1.0 ( $\Delta A = 0.1$ ).  $A$  was used to parameterize the behavior-direction of the data array. The resulting data was hence a third-order array with dimension  $N_s \times T_s \times G_s$ , where the spatial dimension  $N_s = 16$ , the temporal dimension  $T_s = 100$  and the behavior-dimension  $G_s = 11$ . For every fixed value of  $A$ , the joint trajectory data (a second-order array of dimension  $N_s \times T_s$ ) resulting from equation (10) was concisely modeled as a spatiotemporal sinusoidal function, which composed one spatiotemporal mode. This fact can be seen from the power spectrum, as shown in Figure 5(b), which resulted from applying the two-dimensional fast Fourier Transform (FFT) to the joint trajectories with  $A = 1.0$ , as shown in Figure 5(a). Therefore, each  $N_s \times T_s$  second-order data array in the synthesized third-order data array can be uniquely represented by the single coefficient,  $A$ . This coefficient denotes the amplitude associated with the spatiotemporal sinusoidal function, which increases linearly from 0 to 1 along the behavior-direction.

To identify a parameterized model of the synthesized shape trajectory data, we applied the CBAF algorithm to the synthesized data array with sinusoids as the basis functions in the spatial and temporal directions, and with polynomials as the basis functions in the behavior-direction. Only one sinusoidal function in the spatial and temporal direction, and one linear basis in the behavior-direction were selected by CBAF to achieve a perfect reconstruction of the synthesized data array. It was not surprising that these selected basis functions recovered the analytical model, equation (10), which was used to generate the data array in the first place.

## 4. Modeling snake behaviors

Despite the high dimensionality of the shape trajectory data collected from the biological snakes, the CBAF

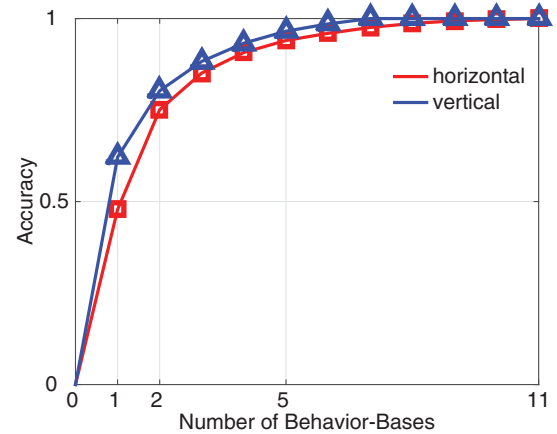


**Fig. 5.** Synthesized joint trajectory data and its power spectrum in the spatiotemporal frequency domain. (a) Joint trajectories sampled from the discrete serpenoid curve with  $A = 1.0$ . (b) The power spectrum of the synthesized joint trajectories in the spatiotemporal frequency domain. Squares at different locations denote sinusoidal functions at different spatiotemporal frequencies. A red square denotes a large amplitude and a white square denotes zero amplitude.

algorithm equipped us with mathematical tools to identify a meaningful low-dimensional representation of the biological data. We referred to this low-dimensional representation as the *gait model* of the observed biological motion.

#### 4.1. Choice of basis functions

To construct a meaningful gait model from the snake-shape trajectory data, we used sinusoids as the basis functions in the spatial and temporal directions. Sinusoidal basis functions in the spatial direction are known as the *serpenoid curves* (Hirose, 1993), whose curvature varies sinusoidally along the arc length. Serpenoid curves have proven effective in modeling snake undulations (Mori and Hirose, 2002; Ostrowski and Burdick, 1998; Tesch et al., 2009). Serpenoid curves at different spatial frequencies clearly depict the shape of the snake bodies (Sharpe et al., 2015). In the temporal direction, sinusoidal basis functions describe individual joint trajectories as sine waves at different temporal frequencies. Sinusoidal functions are well suited for approximating the periodic joint movements exhibited by snakes (Gonzalez-Gomez et al., 2006, 2007; Ijspeert, 2008; Ijspeert and Crespi, 2007). Besides the straightforward physical meanings as stated above, sinusoidal basis functions resulted in parameterizations that are unitless, because the data were parameterized in terms of



**Fig. 6.** Data reconstruction accuracy of the biological data arrays (on sand) versus the number of bases in the behavior-direction. The data reconstruction accuracy was computed as  $\|\mathcal{A} - \hat{\mathcal{A}}\|/\|\mathcal{A}\|$ .

different temporal and spatial frequencies. Therefore the resultant parameterizations can be easily modified to generate motion on snake robots with different numbers of joints, by normalizing the robots to unit length. The freedom to choose basis functions highlights the benefit of the CBAF algorithm, which allows the identification of meaningful parameterized models from high-dimensional data arrays. In the behavior-direction, there were no pre-established good bases for modeling joint variations across different snake behaviors. We hence used SVD to determine appropriate bases in the behavior-direction. Although SVD was used in CBAF, the CBAF algorithm did not require the repetitive use of SVD for fitting basis functions, as in HOSVD and ALS. As a result, the computational cost of CBAF was much lower than either HOSVD or ALS.

Using sinusoids as basis functions in the spatial and temporal directions allowed us to use the two-dimensional FFT to select sinusoidal basis functions in the spatial and temporal directions. We then applied SVD to the coefficients associated with the remaining spatiotemporal modes to fit bases in the behavior-direction.

#### 4.2. Gait model

To account for independent snake body undulations in the horizontal and vertical planes, we applied the CBAF algorithm to the horizontal and vertical data arrays, respectively.

*Horizontal wave.* For the horizontal data array, three sinusoids in the spatial direction and one sinusoid in the temporal direction were selected to capture  $>99\%$  of the data. In the behavior-direction, two behavior-bases were sufficient to achieve a  $>75\%$  final reconstruction accuracy, see Figure 6. This result means body undulations in the horizontal plane can be closely approximated by the composition of two spatiotemporal functions, each of which was represented as three coupled spatiotemporal sinusoidal

functions (see Appendix C). We call these two spatiotemporal functions the *horizontal gait patterns*. Therefore, snake body undulations in the horizontal plane were represented by two coefficients,  $w_1^h$  and  $w_2^h$ , denoting the weights associated with the two horizontal gait patterns.

The horizontal gait patterns revealed the characteristics of snake body undulations in the horizontal plane. In the temporal direction, only one sinusoid (at the same frequency as the gait) was required to approximate the temporal joint variations, indicating smooth and steady joint motions over time. However, three sinusoids were required in the spatial direction, denoting relatively complicated waveforms along the snakes. We believe such a complicated waveform is the key to the generation of differential turning, which requires non-uniform amplitudes along the body (Astley et al., 2015).

*Vertical wave.* Two sinusoids in the spatial direction, while three sinusoids in the temporal direction were required for capturing >99% of the data in the vertical direction. In the behavior-direction, we selected two behavior-bases to account for >80% of the data, as shown in Figure 6. Body undulations in the vertical plane were hence represented by the compositions of two *vertical gait patterns*, whose weights were denoted by  $w_1^v$  and  $w_2^v$ .

Snake body undulations in the vertical plane presented disparate characteristics from their motion in the horizontal plane. The two sinusoids in the spatial direction suggested relatively simple body waveforms compared to the horizontal undulations. However, three sinusoids in the temporal direction indicated complex temporal behaviors of individual joint movements. We believe rapid joint movements in time are essential to the generation of reversal turning, which requires a sudden  $\pi$  phase shift in the vertical wave, as shown in Figure 7(f).

## 5. Robot experiments

In this section, we show how the parameterized gait model can be used to define novel families of behaviors. Building upon the fundamental premise that sidewinding is principally kinematic, the gait model allowed us to duplicate snake motions on snake robots (Gong et al., 2014). Moreover, the parameterized gait model allowed us to synthesize novel behaviors that were not observed in the biological snakes. Implementing these behaviors on a snake robot resulted in significantly improved locomotive performance.

### 5.1. Two modes of turning

We used the CMU modular snake robot (Wright et al., 2012) to test the efficacy of the gait model in generating useful motions in snake robots. The unified snake robot is a serial chain of 16 identical modules, each consisting of one revolute joint. The joints of the robot alternatively rotate in the dorsal and lateral directions so that the robot can generate 3D body undulations. In the robot

experiments, we did not take into account the one-module offset between the odd and even joints (caused by the alternating joint arrangement), and directly used the odd joints for generating horizontal undulations and the even joints for generating vertical undulations.

To reproduce the observed snake motions on our snake robot, we computed the weights associated with the gait patterns by projecting the snake shape trajectory data onto the horizontal and vertical gait patterns (see Appendix C). Commanding the resultant joint trajectories to the unified snake robot resulted in two qualitatively different types of turning motion, as shown in Figure 7 (see also Extension 1 and Extension 3), which corresponded to the differential turning and reversal turning observed on the biological snakes. As shown in Figure 8, the workspace trajectory of robot differential turning closely resembled that of snake differential turning.

To the best of our knowledge, the re-orientation part of reversal turning has never been implemented on a snake robot before, though reversals without a re-orientation phase were implemented in Astley et al. (2015). Robot reversal turning achieved an improved level of agility, defined as the number of gait cycles required to rotate  $2\pi$  rad, and maneuverability, defined as the radius of the minimal enclosing circle of the center of mass (CoM) trajectory after rotating  $2\pi$  rad. Such a tight turning motion directly benefits applications in confined spaces, like those commonly encountered in urban search and rescue.

To highlight the improved locomotive performance of our snake robot, we compared robot reversal turning with *conical sidewinding* (Gong et al., 2013) and the *turn-in-place gait* (Tesch et al., 2009) in terms of agility and maneuverability, as shown in Figure 9. Conical sidewinding gait was implemented by using the following gait expression

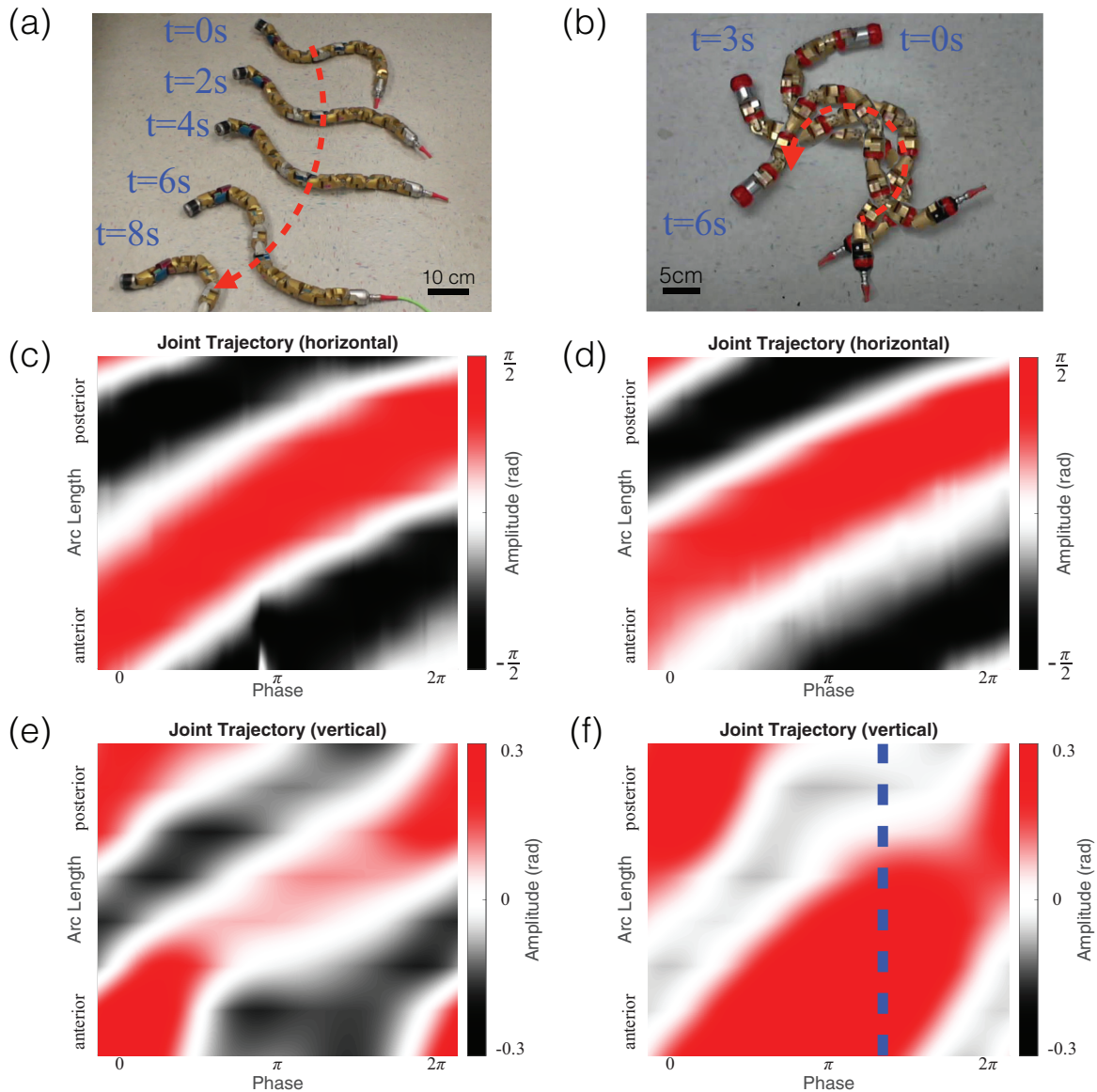
$$\alpha(n, t) = \begin{cases} A_c(n) \sin(\Omega n + \omega t) & \text{odd} \\ eA_c(n) \sin(\Omega n + \omega t + \frac{\pi}{2}) & \text{even} \end{cases} \quad (11)$$

where  $n, t, \Omega = \frac{2\pi}{12}, \omega = 2\pi, e = 0.3$  denote the joint index, time, spatial frequency, temporal frequency and aspect ratio, respectively. The amplitude function is expressed as

$$A_c(n) = a(n - \bar{n}) + 0.7 \quad (12)$$

where  $\bar{n} = 8$  denotes the index of the middle joint, and the amplitude gradient,  $a$ , regulates the speed of turning. We used  $a = 0.068$  to generate conical sidewinding at its maximum speed of turning and with the smallest turning radius (Gong et al., 2013). Robot experiments were repeated 20 times, and the agility and maneuverability in each robot trial is graphically represented as a red square in Figure 9. The turn-in-place gait was achieved by commanding two different halves of the robot to sidewind in opposite directions (Tesch et al., 2009). The green stars in Figure 9 show the performance of the turn-in-place gait in 10 robot trials. Robot reversal turning (blue triangles in Figure 9) presented superior agility ( $6.32 \pm 1.03$  cycles) to both conical sidewinding (agility =  $9.64 \pm 0.75$  cycles)





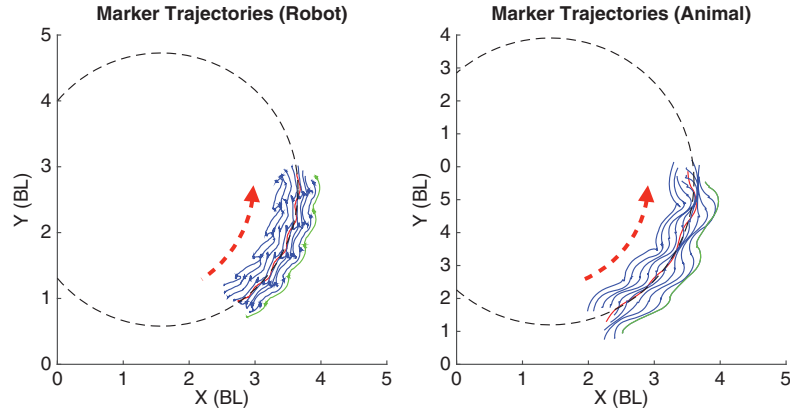
**Fig. 7.** A robot reproducing snake differential turning and reversal turning on hard ground. (a) Robot differential turning. The red dashed arrow denotes the direction of motion. (b) Robot reversal turning. The red dashed arrow denotes the direction of motion. (c) Horizontal joint trajectories of robot differential turning. The horizontal joint trajectories of robot differential turning exhibit a wave pattern traveling from head to tail. (d) Horizontal joint trajectories of robot reversal turning. The horizontal joint trajectories of robot reversal turning present a similar pattern to the horizontal joint trajectories of robot differential turning. (e) The vertical joint trajectories of robot differential turning. The vertical joint trajectories of robot differential turning present a wave pattern similar to (c) but with a phase shift. (f) Vertical joint trajectories of robot reversal turning. The vertical joint trajectories of robot reversal turning present an abrupt phase change (highlighted by the blue dashed line), which is responsible for the emergence of reversal turning.

and the turn-in-place gait (agility =  $7.50 \pm 0.67$  cycles). In terms of maneuverability, the reversal turning (maneuverability =  $0.15 \pm 0.01$  m) was comparable to the turn-in-place gait (maneuverability =  $0.16 \pm 0.03$  m) and outperformed conical sidewinding (maneuverability =  $0.50 \pm 0.05$  m).

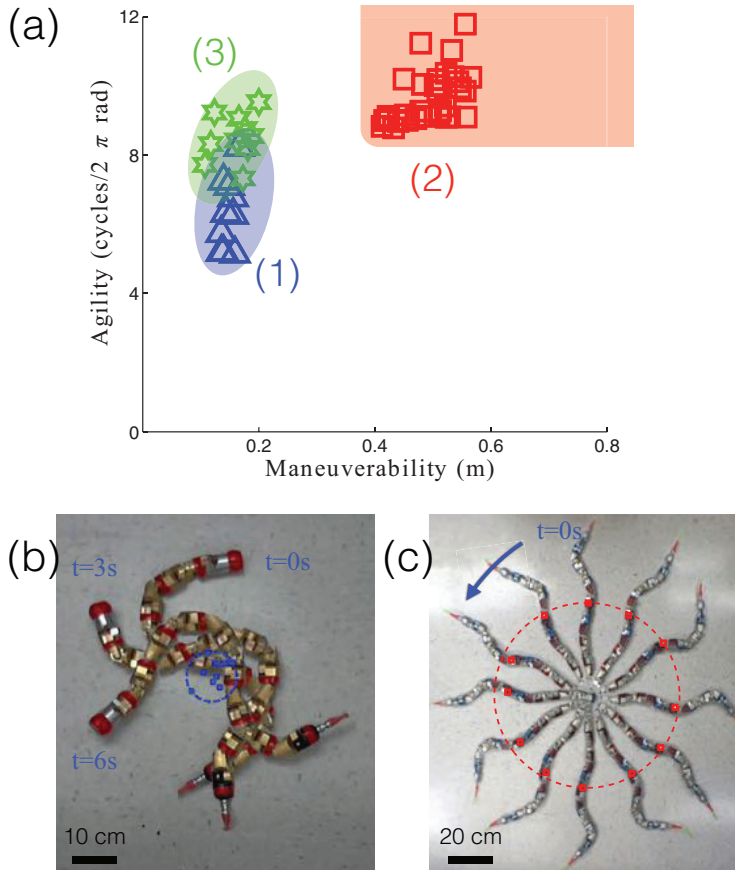
## 5.2. Model exploration

The concise and intuitive gait model enabled us to manually explore the parameter space for generating novel

behaviors on our snake robot. Building upon the robot reversal turning, increasing its horizontal amplitude by a factor of 1.2 while holding the vertical amplitude constant, produced a more rapid turning motion (agility =  $5.20 \pm 0.88$  cycles, maneuverability =  $0.20 \pm 0.01$  m). Further increasing the horizontal amplitude by a factor of 1.4 resulted in an even more agile turning motion (agility =  $4.27 \pm 0.87$  cycles) with a slight degeneration in maneuverability ( $0.22 \pm 0.02$  m).



**Fig. 8.** A comparison of the workspace trajectories, normalized by body length (BL), between robot and snake differential turning on sand. The blue curves denote the trajectories of individual markers. The red curves denote the CoM (approximated as the mean position of the markers) trajectories. The black dashed circles denote the best-fit circles to the CoM trajectories and the green curves denote the trajectories of the head. The red dashed arrows denote the direction of the motion. The turning radius of robot differential turning was 2.07 BL and the turning radius of snake differential turning was 2.17 BL.



**Fig. 9.** Comparison of the locomotive performance of different robot behaviors. (a) Locomotive performance of (1) robot reversal turning, (2) conical sidewinding and (3) the turn-in-place gait in terms of agility and maneuverability. The small number of gait cycles required to turn  $360^\circ$  means high agility, and a small radius of turning implies high maneuverability. The blue triangles, green stars and red squares denote different trials of reversal turning, the turn-in-place gait and conical sidewinding, respectively. The shaded areas approximate the distribution of locomotive performance of different behaviors. The red squares only represent conical sidewinding at its greatest agility and maneuverability. Conical sidewinding can continuously adjust its turning radius and turning speed in a range approximately represented by the shaded red region. (b) An illustration of the maneuverability of robot reversal turning. The blue squares denote the approximated CoM positions in a sequence of snapshots, and the blue dashed circle is the minimal enclosing circle. (c) An illustration of the maneuverability of conical sidewinding. The red squares denote the approximated CoM positions in a sequence of snapshots, and the red dashed circle is the minimal enclosing circle. The time interval between the snapshots is 2 s. The blue arrow denotes the direction of motion.

## 6. Conclusions and future work

This work presented *conditioned basis array factorization* for identifying concise analytical parameterizations of high-dimensional data arrays. Applying CBAF to the shape trajectory data collected from sidewinder rattlesnakes resulted in a concise analytical gait model, which allowed us to generate a wide range of behaviors through adjusting a few parameters. By allowing for independent modulations of body waves in the horizontal and vertical planes, the two-wave gait model successfully captured a wide range of snake behaviors, including differential turning and reversal turning. Implementing these behaviors on a snake robot improved its agility and maneuverability. The demonstrated success and generality of the presented analysis technique suggest that a similar approach could be used to generate useful motions for other systems whose motions are principally kinematic.

Decomposing 3D snake undulations into two orthogonal body waves is an intuitive way to study 3D limbless locomotion. Unlike horizontal body undulations, body undulations in the vertical plane can alter the position of contacts, which in turn change the system dynamics. Further investigations are necessary to better understand how the interplay between the two body waves determines locomotion in the workspace. While we only manually explored the parameter space of the two-wave gait model in this paper, optimization techniques could be employed to facilitate the discovery of useful motions within the two-wave gait model.

### Funding

This research was supported by the Army Research Laboratory and was accomplished under Cooperative Agreement Number W911NF-10-2-0016. DIG thanks the Physics of Living Systems program at the Natural Science Foundation and the Army Research Office.

### References

- Acar E, Çamtepe SA and Yener B (2006) Collective sampling and analysis of high order tensors for chatroom communications. In: Mehrotra S, Zeng DD, Chen H, Thuraisingham B and Wang F-Y (eds) *Intelligence and Security Informatics*. Berlin: Springer, pp. 213–224.
- Akhter I, Simon T, Khan S, Matthews I and Sheikh Y (2012) Bilinear spatiotemporal basis models. *ACM Transactions on Graphics* 31(2): 17.
- Astley H, Gong C, Travers M, Serrano M, Vela P, Choset H, et al. (2015) Modulation of orthogonal body waves enables high maneuverability in sidewinding locomotion. *Proceedings of the National Academy of Sciences* 112(19): 6200–6205.
- Berry MW, Dumais ST and O'Brien GW (1995) Using linear algebra for intelligent information retrieval. *SIAM Review* 37(4): 573–595.
- Chen J and Saad Y (2009) On the tensor SVD and the optimal low rank orthogonal approximation of tensors. *SIAM Journal on Matrix Analysis and Applications* 30(4): 1709–1734.
- Chirikjian G and Burdick J (1994) A modal approach to hyper-redundant manipulator kinematics. *IEEE Transactions on Robotics and Automation* 10(3): 343–354.
- Dawande M, Keskinocak P, Swaminathan JM and Tayur S (2001) On bipartite and multipartite clique problems. *Journal of Algorithms* 41(2): 388–403.
- De Lathauwer L, De Moor B and Vandewalle J (2000a) A multilinear singular value decomposition. *SIAM Journal on Matrix Analysis and Applications* 21(4): 1253–1278.
- De Lathauwer L, De Moor B and Vandewalle J (2000b) On the best rank-1 and rank- $(R_1, R_2, \dots, R_N)$  approximation of higher-order tensors. *SIAM Journal on Matrix Analysis and Applications* 21(4): 1324–1342.
- Gong C, Hatton R and Choset H (2012) Conical sidewinding. In: *2012 IEEE conference on robotics and automation (ICRA '12)*, Saint Paul, USA, 14–18 May 2012, pp. 4222–4227. Piscataway: IEEE Press.
- Gong C, Travers M, Astley H, Li L, Mendelson J, Hu D, et al. (2014) Conditioned basis array factorization: An approach to gait pattern extraction. In: *Proceedings of Robotics: science and systems X*, Berkeley, USA, 12–16 July 2014. Available at: <http://www.roboticsproceedings.org/rss10/p55.html>
- Gong C, Travers MJ, Fu X and Choset H (2013) Extended gait equation for sidewinding. In: *2013 IEEE conference on robotics and automation (ICRA '13)*, Karlsruhe, Germany, 6–10 May 2013, pp. 5162–5167. Piscataway: IEEE Press.
- Gonzalez-Gomez J, Zhang H and Boemo E (2007) Locomotion principles of 1D topology pitch and pitch-yaw-connecting modular robots. In: Habib MK (ed) *Bioinspiration and Robotics: Walking and Climbing Robots*. Rijeka: InTech Open Access.
- Gonzalez-Gomez J, Zhang H, Boemo E and Zhang J (2006) Locomotion capabilities of a modular robot with eight pitch-yaw-connecting modules. In: *9th international conference on climbing and walking robots*, Brussels, Belgium, 12–14 September 2006, pp. 402–428. Berlin: Springer.
- Hatton RL and Choset H (2010) Generating gaits for snake robots: Annealed chain fitting and keyframe wave extraction. *Autonomous Robots* 28(3): 271–281.
- Hatton RL and Choset H (2013) Geometric swimming at low and high Reynolds numbers. *IEEE Transactions on Robotics* 29(3): 615–624.
- Hirose S (1993) *Biologically Inspired Robots (Snake-like Locomotor and Manipulator)*. Oxford: Oxford University Press.
- Hu DL, Nirody J, Scott T and Shelley MJ (2009) The mechanics of slithering locomotion. *Proceedings of the National Academy of Sciences* 106(25): 10081–10085.
- Ijspeert AJ (2008) Central pattern generators for locomotion control in animals and robotics. *Neural Networks* 21(4): 642–653.
- Ijspeert AJ and Crespi A (2007) Online trajectory generation in an amphibious snake robot using a Lamprey-like central pattern generator model. In: *2007 IEEE conference on robotics and automation (ICRA '07)*, Roma, Italy, 10–14 April 2007, pp. 262–268. Piscataway: IEEE Press.
- Jayne BC (1986) Kinematics of terrestrial snake locomotion. *Copeia* 1986(4): 915–927.
- Kolda TG and Bader BW (2009) Tensor decompositions and applications. *SIAM Review* 51(3): 455–500.
- Kolda TG, Bader BW and Kenny JP (2005) Higher-order web link analysis using multilinear algebra. In: *5th IEEE international conference on data mining*, Houston TX, USA, 27–30 November 2005, pp. 242–249. Piscataway: IEEE Press.
- Krooneberg PM (1983) *Three-Mode Principal Component Analysis: Theory and Applications*. Leiden: DSWO Press.

- Kroonenberg PM and De Leeuw J (1980) Principal component analysis of three-mode data by means of alternating least squares algorithms. *Psychometrika* 45(1): 69–97.
- Lee CS and Elgammal A (2004) Gait style and gait content: Bilinear models for gait recognition using gait re-sampling. In: *6th IEEE international conference on automatic face and gesture recognition*, Soul, South Korea, 17–19 May 2004, pp. 147–152. Piscataway: IEEE Press.
- Liljebäck P, Pettersen KY, Stavadhø and Gravadahl JT (2012) *Snake Robots: Modelling, Mechatronics, and Control*. Berlin: Springer.
- Maladen RD, Ding Y, Li C and Goldman DI (2009) Undulatory swimming in sand: Subsurface locomotion of the sandfish lizard. *Science* 325(5938): 314–318.
- Marvi H, Gong C, Gravish N, Astley H, Travers M, Hatton RL, et al. (2014) Sidewinding with minimal slip: Snake and robot ascent of sandy slopes. *Science* 346(6206): 224–229.
- Mori M and Hirose S (2001) Development of active cord mechanism ACM-R3 with agile 3D mobility. In: *2001 IEEE/RSJ international conference on intelligent robots and systems (IROS'01)*, Maui HI, USA, 29 October–3 November 2001, vol. 3, pp. 1552–1557. Piscataway: IEEE Press.
- Mori M and Hirose S (2002) Three-dimensional serpentine motion and lateral rolling by active cord mechanism ACM-R3. In: *2002 IEEE/RSJ international conference on intelligent robots and systems (IROS'02)*, Lausanne, Switzerland, 30 September–4 October 2002, vol. 1, pp. 829–834. Piscataway: IEEE Press.
- Mosauer W (1930) A note on the sidewinding locomotion of snakes. *The American Naturalist* 64(691): 179–183.
- Ostrowski J and Burdick J (1998) The geometric mechanics of undulatory robotic locomotion. *International Journal of Robotics Research* 17(7): 683–702.
- Quelhas P, Monay F, Odobez JM, Gatica-Perez D, Tuytelaars T and Van Gool L (2005) Modeling scenes with local descriptors and latent aspects. In: *10th IEEE international conference on computer vision*, Beijing, China, 17–20 October 2005, vol. 1, pp. 883–890. Piscataway: IEEE Press.
- Shammas EA (2006) *Generalized motion planning for underactuated mechanical systems*. PhD Thesis, Carnegie Mellon University, USA.
- Sharpe SS, Koehler SA, Kuckuk RM, et al. (2015) Locomotor benefits of being a slender and slick sand swimmer. *Journal of Experimental Biology* 218(3): 440–450.
- Socha JJ (2002) Kinematics: Gliding flight in the paradise tree snake. *Nature* 418(6898): 603–604.
- Stephens GJ, Johnson-Kerner B, Bialek W and Ryu WS (2008) Dimensionality and dynamics in the behavior of *C. elegans*. *PLoS Computational Biology* 4(4): e1000028.
- Tanaka M and Matsuno F (2009) A study on sinus-lifting motion of a snake robot with switching constraints. In: *2009 IEEE conference on robotics and automation (ICRA'09)*, Kobe, Japan, 12–17 May 2009, pp. 2270–2275. Piscataway: IEEE Press.
- Tanev I, Ray T and Buller A (2005) Automated evolutionary design, robustness, and adaptation of sidewinding locomotion of a simulated snake-like robot. *IEEE Transactions on Robotics* 21(4): 632–645.
- Tesch M, Lipkin K, Brown I, Hatton R, Peck A, Rembisz J, et al. (2009) Parameterized and scripted gaits for modular snake robots. *Advanced Robotics* 23(9): 1131–1158.
- Tesch M, Schneider J and Choset H (2011) Using response surfaces and expected improvement to optimize snake robot gait parameters. In: *2011 IEEE/RSJ international conference on intelligent robots and systems (IROS'11)*, San Francisco, USA, 25–30 September 2011, pp. 1069–1074. Piscataway: IEEE Press.
- Transth AA, Pettersen KY and Liljebäck P (2009) A survey on snake robot modeling and locomotion. *Robotica* 27(7): 999–1015.
- Twining CJ and Taylor CJ (2001) Kernel principal component analysis and the construction of non-linear active shape models. In: *British machine vision conference (BMVC 2001)*, Manchester, UK, 10–13 September 2001, pp. 1–10. Durham: The British Machine Vision Association and Society for Pattern Recognition (BMVA).
- Vasilescu MAO and Terzopoulos D (2002) Multilinear analysis of image ensembles: Tensorfaces. In: Heyden A, Sparr G, Nielsen M and Johansen P (eds) *Computer Vision ECCV 2002*. Berlin: Springer, pp. 447–460.
- Wang L, Tan T, Ning H and Hu W (2003) Silhouette analysis-based gait recognition for human identification. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 25(12): 1505–1518.
- Wright C, Buchan A, Brown B, Geist J, Schwerin M, Rollinson D, et al. (2012) Design and architecture of the unified modular snake robot. In: *2012 IEEE conference on robotics and automation (ICRA'12)*, Saint Paul, USA, 14–18 May 2012, pp. 4347–4354. Piscataway: IEEE Press.
- Yamada H and Hirose S (2006) Development of practical 3-dimensional active cord mechanism ACM-R4. *Journal of Robotics and Mechatronics* 18(3): 305.
- Zhao W, Krishnaswamy A, Chellappa R, Swets DL and Weng J (1998) Discriminant analysis of principal components for face recognition. In: *Face Recognition*. Berlin: Springer, pp. 73–85.

## Appendix A: Index to Multimedia Extensions

Archives of IJRR multimedia extensions published prior to 2014 can be found at <http://www.ijrr.org>, after 2014 all videos are available on the IJRR YouTube channel at <http://www.youtube.com/user/ijrrmultimedia>

**Table of Multimedia Extensions**

Extension	Media type	Description
1	Video	Robot and snake differential turning
2	Video	Snake reversal turning
3	Video	Robot reversal turning

## Appendix B: Conditioned basis array factorization

### Computational complexity of CBAF

Computing the core array requires  $N$  matrix multiplications which cost  $O\left(N \prod_{k=1}^N I_k\right)$ . Along each direction, computing

the energy associated with all the bases takes  $O\left(\prod_{k=1}^N I_k\right)$  summations. Sorting the bases according to their energy, costs  $O(I_k \log I_k)$  along each direction- $k$ . Therefore, selecting truncated bases along all directions costs  $O\left(N \prod_{k=1}^N I_k + \sum_{k=1}^N I_k \log I_k\right)$ . The total computational cost of CBAF is hence bounded by  $O\left(N \prod_{k=1}^N I_k\right)$ .

### Low rank optimal reconstruction is NP-hard

To prove low rank optimal reconstruction is NP-hard, we first define the decision version of the LROR problem.

**Definition.** The decision version of LROR is defined as follows. Given a third-order array  $\mathcal{A}$ , conditioned basis matrices  $U^{(1)} \in \mathbb{R}^{I_1 \times I_1}$ ,  $U^{(2)} \in \mathbb{R}^{I_2 \times I_2}$ ,  $U^{(3)} \in \mathbb{R}^{I_3 \times I_3}$  and integers  $R_1, R_2, R_3$ , there exists a rank- $(R_1, R_2, R_3)$  reconstruction with the squared norm of the truncated basis array  $\hat{\mathcal{C}}$  greater than or equal to  $L$ ,  $\|\hat{\mathcal{C}}\|^2 \geq L$ .

**Theorem A.1.** Low rank optimal reconstruction is NP-hard.

**Proof.** To prove low rank optimal reconstruction is NP-hard, we show a reduction from the *Exact Node Cardinality Decision Problem (ENCD)* (Dawande et al., 2001) of biclique. ENCD is stated as follows: Given a bipartite graph  $G(V_1 \cup V_2, E)$  and two positive integers  $R_1, R_2$ ,  $G$  contains a biclique (complete bipartite graph) which has  $R_1$  nodes from  $V_1$  and  $R_2$  nodes from  $V_2$ . ENCD is known to be NP-complete.

*Reduction:* Given a bipartite graph  $G(V_1 \cup V_2, E)$ , construct a array (matrix)  $A \in \mathbb{R}^{|V_1| \times |V_2|}$ . For every pair of nodes  $(v_1, v_2)$ ,  $v_1 \in V_1$  and  $v_2 \in V_2$ , if  $(v_1, v_2) \in E$ , set the entry  $a_{v_1, v_2}$  in  $A$  to be 1. Otherwise, let  $a_{v_1, v_2} = 0$ . Let the conditioned basis matrices  $U^{(1)} \in \mathbb{R}^{|V_1| \times |V_1|}$  and  $U^{(2)} \in \mathbb{R}^{|V_2| \times |V_2|}$  be identity matrices. Then the original ENCD is formulated as the LROR problem: whether  $A$  has a rank- $(R_1, R_2)$  reconstruction with energy no less than  $R_1 R_2$ . This reduction is done in polynomial time.

*Equivalence:* Suppose  $G(V_1 \cup V_2, E)$  has a biclique  $G'(V'_1 \cup V'_2, E')$  with  $|V'_1| = R_1$ ,  $|V'_2| = R_2$  and  $|E'| = R_1 R_2$ . It is equivalent to saying

$$\sum_{v_1 \in V'_1, v_2 \in V'_2} a_{v_1, v_2} = R_1 R_2 = \sum_{v_1 \in V'_1, v_2 \in V'_2} a_{v_1, v_2}^2 \quad (13)$$

which results in a yes-instance of LROR. The second equivalence in the equation above has used the fact that  $a_{v_1, v_2} = 1$ . In the other direction, suppose there exists a rank- $(R_1, R_2)$  reconstruction with energy no less than  $R_1 R_2$ . Because the conditioned basis matrices are selected to be identity matrices, there exists a subarray  $A' \in \mathbb{C}^{R_1 \times R_2}$  with entries all being 1. Therefore, according to how  $A$  has been constructed, there exists a biclique  $G'(V'_1 \cup V'_2, E')$  with  $|V'_1| = R_1$ ,  $|V'_2| = R_2$ . In conclusion, LROR is NP-hard.

### Suboptimality guarantee of CBAF

Given the conditioned basis matrices  $U^{(1)} \in \mathbb{R}^{I_1 \times I_1}$ ,  $U^{(2)} \in \mathbb{R}^{I_2 \times I_2}$  and  $U^{(3)} \in \mathbb{R}^{I_3 \times I_3}$ , CBAF constructs submatrices  $V^{(1)} \in \mathbb{R}^{I_1 \times R_1}$ ,  $V^{(2)} \in \mathbb{R}^{I_2 \times R_2}$  and  $V^{(3)} \in \mathbb{R}^{I_3 \times R_3}$ . Denote the reconstructed array after direction-1 basis truncation as

$$\mathcal{A}^{(1)} = \mathcal{A} \times_1 V^{(1)} \times_1 (V^{(1)})^T \quad (14)$$

The residual array is

$$\mathcal{B}^{(1)} = \mathcal{A} - \mathcal{A}^{(1)} \quad (15)$$

The reconstruction error after direction-1 basis truncation can be computed as

$$\epsilon_{(1)} = \|\mathcal{B}^{(1)}\| \quad (16)$$

In the same manner, after direction-1 and direction-2 basis truncation

$$\mathcal{A}^{(1,2)} = \mathcal{A}^{(1)} \times_2 V^{(2)} \times_2 (V^{(2)})^T \quad (17)$$

$$\mathcal{B}^{(2)} = \mathcal{A}^{(1)} - \mathcal{A}^{(1,2)} \quad (18)$$

$$\epsilon_{(2)} = \|\mathcal{B}^{(2)}\| \quad (19)$$

After direction-1, direction-2 and direction-3 basis truncation

$$\mathcal{A}^{(1,2,3)} = \mathcal{A}^{(1,2)} \times_3 V^{(3)} \times_3 (V^{(3)})^T \quad (20)$$

$$\mathcal{B}^{(3)} = \mathcal{A}^{(1,2)} - \mathcal{A}^{(1,2,3)} \quad (21)$$

$$\epsilon_{(3)} = \|\mathcal{B}^{(3)}\| \quad (22)$$

**Lemma A.1** The reconstruction error  $\epsilon$  of CBAF is bounded by

$$\max\{\epsilon_{(1)}, \epsilon_{(2)}, \epsilon_{(3)}\} \leq \epsilon \leq \epsilon_{(1)} + \epsilon_{(2)} + \epsilon_{(3)} \quad (23)$$

**Proof** The reconstruction error of CBAF can be computed as

$$\begin{aligned} \epsilon &= \|\mathcal{A} - \mathcal{A}^{(1,2,3)}\| = \|\mathcal{A} - \mathcal{A}^{(1,2)} + \mathcal{B}^{(3)}\| \\ &= \|\mathcal{A} - \mathcal{A}^{(1)} + \mathcal{B}^{(2)} + \mathcal{B}^{(3)}\| \\ &= \|\mathcal{B}^{(1)} + \mathcal{B}^{(2)} + \mathcal{B}^{(3)}\| \\ &\leq \epsilon_{(1)} + \epsilon_{(2)} + \epsilon_{(3)} \text{ (triangle inequality)} \end{aligned} \quad (24)$$

which proves the upper bound. For the lower bound, it can be easily shown  $\mathcal{B}^{(1)}$ ,  $\mathcal{B}^{(2)}$  and  $\mathcal{B}^{(3)}$  are mutually orthogonal, as a fact of Theorem 3.1. As a result

$$\begin{aligned} \epsilon^2 &= \|\mathcal{B}^{(1)} + \mathcal{B}^{(2)} + \mathcal{B}^{(3)}\|^2 \\ &= \epsilon_{(1)}^2 + \epsilon_{(2)}^2 + \epsilon_{(3)}^2 \\ &\geq \max\{\epsilon_{(1)}^2, \epsilon_{(2)}^2, \epsilon_{(3)}^2\} \end{aligned}$$

**Lemma A.2** The reconstruction error of the optimal solution to LROR is bounded by

$$\epsilon^* \geq \max\{\epsilon_{(1)}, \epsilon_{(2)}, \epsilon_{(3)}\}$$

**Proof.** Let  $\bar{\epsilon}_{(k)}$  denote the reconstruction error after basis truncation along direction- $k$ . Note  $\bar{\epsilon}_{(k)}$  denotes the reconstruction error resulting from *only* truncating direction- $k$  bases while  $\epsilon_{(k)}$  denotes the additional reconstruction error after basis truncation along the first  $k - 1$  directions. Clearly,  $\bar{\epsilon}_{(k)} \geq \epsilon_{(k)}$ . According to the CBAF algorithm, because the criterion of basis truncation along each direction is minimal energy loss, the reconstruction error of the optimal choice of basis  $\epsilon^* \geq \bar{\epsilon}_{(k)}$  for every direction- $k$ . Therefore

$$\epsilon^* \geq \max\{\bar{\epsilon}_{(1)}, \bar{\epsilon}_{(2)}, \bar{\epsilon}_{(3)}\} \geq \max\{\epsilon_{(1)}, \epsilon_{(2)}, \epsilon_{(3)}\}$$

**Theorem A.2.** CBAF is a factor-3 approximation algorithm.

**Proof.** From Lemmas A.1 and A.2 it can be shown

$$\begin{aligned} \epsilon &\leq \epsilon_{(1)} + \epsilon_{(2)} + \epsilon_{(3)} \\ &\leq 3 \max\{\epsilon_{(1)}, \epsilon_{(2)}, \epsilon_{(3)}\} \leq 3\epsilon^* \end{aligned}$$

## Appendix C: Two-wave gait model

The two-wave gait model is composed of two independent wave components for modeling body undulations in the horizontal and vertical planes. Each of the two wave components in turn, is represented as the composition of two gait patterns. A gait is hence modeled as  $w_1^h P_1^h + w_2^h P_2^h$  and  $w_1^v P_1^v + w_2^v P_2^v$ , where  $P_1^h, P_2^h$  denote the horizontal gait patterns and  $P_1^v, P_2^v$  denote the vertical gait patterns.

### Gait patterns

The first horizontal gait pattern is expressed as

$$\begin{aligned} P_1^h &= 0.0098 \cos(2\pi t + 0.8312) \\ &+ 0.0371 \cos\left(2\pi t + \frac{2\pi}{N}n + 1.2856\right) \\ &+ 0.0189 \cos\left(2\pi t + \frac{4\pi}{N}n - 0.9547\right) \end{aligned}$$

where  $N$  denotes the number of joints,  $t$  denotes time and  $n$  denote the joint index. The second horizontal gait pattern is expressed as

$$\begin{aligned} P_2^h &= 0.0074 \cos(2\pi t + 2.3039) \\ &+ 0.0393 \cos\left(2\pi t + \frac{2\pi}{N}n + 2.8725\right) \\ &+ 0.0164 \cos\left(2\pi t + \frac{4\pi}{N}n + 0.5333\right) \end{aligned}$$

The first vertical gait pattern is expressed as

$$\begin{aligned} P_1^v &= 0.0074 \cos\left(\frac{2\pi}{N}n + 1.7281\right) \\ &+ 0.0022 \cos(2\pi t - 2.5029) \\ &+ 0.0207 \cos\left(2\pi t + \frac{2\pi}{N}n - 1.1720\right) \\ &+ 0.0044 \cos(4\pi t - 1.5464) \\ &+ 0.0041 \cos\left(4\pi t + \frac{2\pi}{N}n - 1.5908\right) \end{aligned}$$

The second vertical gait pattern is expressed as

$$\begin{aligned} P_2^v &= -0.0084 + 0.0127 \cos\left(\frac{2\pi}{N}n + 2.1094\right) \\ &+ 0.0203 \cos(2\pi t + 2.7282) \\ &+ 0.0307 \cos\left(2\pi t + \frac{2\pi}{N}n - 2.2616\right) \\ &+ 0.0042 \cos(4\pi t - 3.0905) \\ &+ 0.0112 \cos\left(4\pi t + \frac{2\pi}{N}n - 2.8590\right) \end{aligned}$$

### Model coefficients

The model parameters that duplicate snake motions were computed by projecting snake shape trajectory data onto the horizontal and vertical gait patterns. For differential turning, the parameters were  $w_1^h = 0.7519$ ,  $w_2^h = 20.8433$ ,  $w_1^v = 2.6800$  and  $w_2^v = -2.0366$ . And, the model parameters were  $w_1^h = -23.7439$ ,  $w_2^h = 5.4261$ ,  $w_1^v = 7.7737$  and  $w_2^v = 0.8774$  for producing reversal turning.