

# Conditioned Basis Array Factorization: An Approach to Gait Pattern Extraction

Chaohui Gong, Matthew Travers, Henry C. Astley, Lu Li, Joseph Mendelson  
David Hu, Daniel I. Goldman and Howie Choset

**Abstract**—Snakes locomote through sophisticated coordinated motions of their many degrees of freedom (DoFs). The exhibited regularity of their body undulation implies the existence of low dimensional representations of snake gaits. We posit that investigating the underlying motion patterns will lead to insights for understanding how animals control low-level joint motions in a coupled fashion to achieve behavior-level control targets. To study snake motions in a concise way, we develop a novel modal decomposition algorithm called *conditioned basis array factorization* (CBAF). Unlike most modal decomposition algorithms, CBAF uses analytical bases which can be identified with temporal, spatial, and behavioral (e.g., moving in a straight line, turning, etc.) components of snake motions. Applying CBAF to shape change data collected from a series of snake behaviors results in analytical representations of the recorded motions. These analytical representations provide insight into biological system models, as well as generate families of gaits for snake robots. Although this work focuses on snakes, the generality of the analysis techniques suggest that a similar approach can be used as an effective motion generation technique for any system whose locomotion is kinematic in nature.

## I. INTRODUCTION

Snakes, like other limbless organisms, manipulate their internal degrees of freedom to locomote through challenging environments. Snakes are unique that their locomotion are dominated by friction, causing inertial effects to be dissipated before significantly contributing to net motion. This implies that snake locomotion is primarily determined by the body-shape deformations. We hypothesize that, due to the inherently kinematic nature of this type of locomotion, recording, parameterizing, and playing back biological snake data can be used as an effective tool to design more realistic biological models as well as help us as researchers better explore the potential modes of locomotion for snake-like robots. We show that by leveraging intuitive analysis techniques, gait-based controllers can be derived from biological data, which not only endow snake robots with capabilities to simply *replay* data, but parameterize entire families of gaits based on observed biological motions.

Though the shape changes of biological snakes are observed using a vision system, it is difficult to represent them in an analytically concise fashion. Shape change information is collected and stored in high-dimensional discretized arrays that represent how points on the organism’s body move, as a function of time. To uncover the working principles of snake locomotion, tools that allow intuitive analysis of data from a variety of sources are needed. To accomplish this, we develop a modal decomposition algorithm called

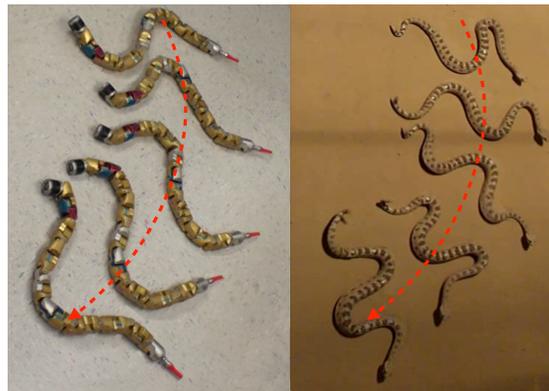


Fig. 1. Comparison between the robot and animal trajectories. The trajectory traced out by the snake robot (left) resembles the trajectory of the rattlesnake (right). Note that the head portion of the rattlesnake is not considered for motion reconstruction. See the supplementary video for more details.

conditioned basis array factorization (CBAF). We apply this algorithm to the shape change trajectory data collected from sidewinder rattlesnakes. The CBAF algorithm extracts motion patterns from biological data which are represented in concise analytical forms. With the analytical representations, we show that it is possible to playback the extracted shape patterns on a snake robot and reproduce biological motions. We also show that these representations can be used to construct families of gaits by varying the parameters in the analytical expressions. The resultant gait models lend valuable insight into the process of designing nontrivial sequences of motions. For example, the CBAF approach produced a parameterized gait that allowed the robot to sidewind along paths of varying curvatures (based on the parameter). This allowed the robot to make tight turns while moving forward, which improve performance in confined spaces.

Our successful application of CBAF in this paper shows that extracting information from biological systems is an effective means of generating behaviors for systems, like sidewinding snake robots, whose locomotion is predominantly governed by the first-order effects of body shape change patterns.

The paper is structured as follows. Before explaining the CBAF algorithm in mathematical detail, we provide the reader with an example by demonstrating the application of CBAF to motion tracking data collected from sidewinding rattlesnakes. In Sec. III, we proceed to show the validity of the extracted expression through the derivation of an analytical, parameterized gait demonstrated on a snake robot. Sec. IV then details the

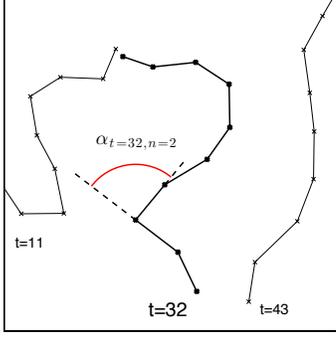


Fig. 2. An illustration of how the virtual joint angles are computed. Tracked marker positions are used to estimate the shape of the biological snake.  $t$  and  $n$  respectively denotes frame number and joint index.

derivation of the CBAF algorithm, the proof of NP-Hardness for finding optimal reconstruction, and the sub-optimality guarantee of CBAF. Finally, we conclude and present future lines of research in Sec. V.

## II. BIOLOGICAL MOTION PATTERN EXTRACTION

In this section, we focus on demonstrating the application of CBAF to shape change sequences collected from sidewinding rattlesnakes. The technical details of CBAF are presented in Sec. IV.

### A. Data Collection

We collected trajectory data from four sidewinder snakes (*Crotalus cerastes* of length =  $48 \pm 8$  cm, mass =  $114 \pm 55$  g) in a customized sand bed ( $2 \times 1 m^2$ ), equipped with four Optitrack Flex 13 IR motion capture cameras (at 120 Hz frame rate). The cameras were used to track 3D positions of 10 IR reflective patches attached to the snakes. The 3D position information provided an estimate of the shape changes as a function of time. Eleven total recordings of the snakes executing a sidewinding gait were taken, within which the snakes exhibited a variable sidewinding behavior, e.g., from straight motion to turning.

Though the tracking system provides 3D information, we focused our study on in-plane movement for two reasons. First, biological snakes move with low ground clearance, making out-of-plane movement hard to detect. Second, previous studies [12, 21] investigated the relation between the horizontal undulation and lifting motion of sidewinding. Hence, the out-of-plane motion can be inferred once the in-plane motions have been determined.

### B. Data Processing

We first trimmed the recorded videos to ensure that each of them contains one full cycle of shape change, i.e., a single period of a gait. In every video frame, we then computed a set of *virtual joint angles* as follows. Given 10 tracker positions on the body, the shape of the rattlesnake can be approximated

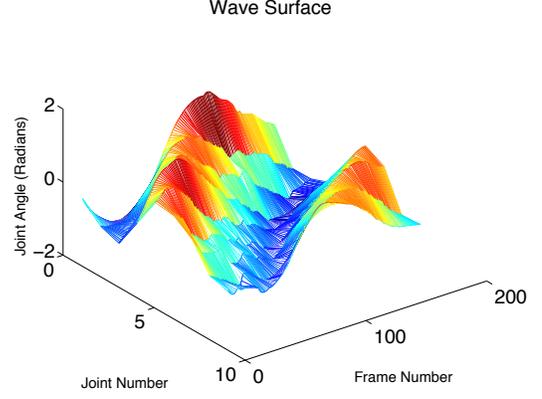


Fig. 3. Wave surface of one trimmed data sequence. The joint angles present a regular pattern across space (joint) and time while a rattlesnake performing sidewinding.

by 9 straight segments, see Fig. 2. Let  $\vec{p}_{t,n}$  denote the in-plane position of the tracker  $n$  at frame  $t$ . The orientation of link  $n$  can be represented by the unit vector,

$$\vec{l}_{t,n} = \frac{\vec{p}_{t,n+1} - \vec{p}_{t,n}}{\|\vec{p}_{t,n+1} - \vec{p}_{t,n}\|} \quad (1)$$

Then the virtual joint angle is computed as

$$\alpha_{t,n} = \tan^{-1} \left( \frac{\|\vec{l}_{t,n+1} - \langle \vec{l}_{t,n}, \vec{l}_{t,n+1} \rangle \vec{l}_{t,n}\|}{\langle \vec{l}_{t,n}, \vec{l}_{t,n+1} \rangle} \right), \quad (2)$$

where counter-clockwise rotations are defined to be positive, e.g., in Fig. 2,  $\alpha_{t=32, n=2}$  is negative. 8 ( $N=8$ ) joint angles are computed from each video frame, then be stored as a vector  $\vec{\alpha}_t$ . Every trimmed video sequence contains various number of frames, thus the shape data contained in each video are represented as a matrix  $A^j \in \mathbb{R}^{8 \times T_j}$ , where  $T_j$  denotes the number of frames in sequence  $j$ . The data matrix can be visualized as a *wave surface* to lend insight into what the system is simultaneously doing across space and time, see Fig. 3. Every point on the surface denotes the angle of a joint at a spatial location on the snake in a particular video frame.

In the final step of data processing, we grouped all the video data into a single multi-dimensional array. To accomplish this, the sequences must have the same dimensions. However, because the snake moves at varying speeds, the time consumed to finish one gait cycle is in general never quite the same. As a result, the data sequences had different lengths ( $T_{j_1} \neq T_{j_2}$ ). Interpolation was thus used to resize each video sequence into 128 ( $T=128$ ) frames,  $A^j \in \mathbb{R}^{8 \times 128}$ . The resized sequences were then grouped together to form a single array  $\mathcal{A} \in \mathbb{R}^{8 \times 128 \times 11}$ .

Representing the joint motion data as a multi-dimensional array maximally preserved data variations across space, time and various behaviors. The first index of the  $\mathcal{A}$  denotes joint angles sampled at different spatial locations on the snakes. Data along this direction contains shape information along the snake's body, hence we call it the *spatial-direction* (basis functions in this direction are called *spatial-bases*). Data

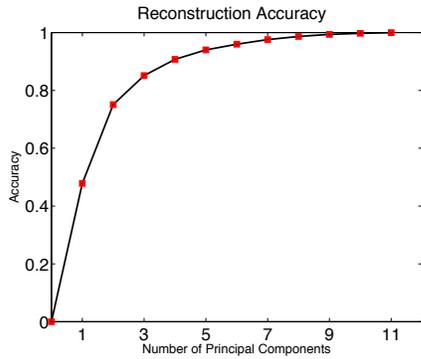


Fig. 4. The accuracy of data reconstruction with respect to the number of behavior-bases used. An over 75% reconstruction to the biological data can be achieved with only two behavior-bases.

listed along the second direction of  $\mathcal{A}$  captures the temporal variations of the joint movements and is termed *temporal-direction* (basis functions in this direction are called *temporal-bases*). The third direction encodes the shape change variations across different observed biological behaviors and is called *behavior-direction* (basis functions in this direction are called *behavior-bases*).

### C. Kinematic Motion Pattern Extraction

We applied the CBAF algorithm to  $\mathcal{A}$  to extract motion patterns underlying biological sidewinding. Fourier bases  $U^{(1)}$ ,  $U^{(2)}$  were used as the basis functions to represent the spatial and temporal variations. The column vectors (basis functions) of  $U^{(1)}$  and  $U^{(2)}$  are expressed as

$$\begin{aligned} \{e^{\frac{2\pi k}{N}ni} | k = 0, 1, \dots, N-1\} & \quad \text{spatial} \\ \{e^{\frac{2\pi k}{T}ti} | k = 0, 1, \dots, T-1\} & \quad \text{temporal} \end{aligned}$$

where  $i$  is the imaginary unit,  $N = 8$  and  $T = 128$ . Fourier bases are selected for three reasons, 1) sinusoids produce good models of undulatory motion [13], 2) phase shifts in periodic data can be easily handled, and 3) parameterization in terms of frequencies is unitless, hence it is insensitive to data collected from biological snakes of different lengths.

The third direction (behavior direction) of the data array captures the kinematic variations across the different behaviors, e.g., straight vs. turning sidewinding. Because there were no pre-known bases across the behavior space, we used singular-value decomposition (SVD) as a data driven method to identify the bases.

The raw data  $\mathcal{A}$  required 8 spatial-basis functions, 128 temporal-basis functions, and 11 behavior-basis functions for a perfect reconstruction. After the application of CBAF, we observed that two temporal  $V^{(2)} \in \mathbb{R}^{128 \times 2}$  and six spatial bases  $V^{(1)} \in \mathbb{R}^{8 \times 6}$  were needed to achieve over 0.99 reconstruction accuracy. The two temporal bases mean that all the joints moved at a temporal frequency equal to the gait. The outer product (see Sec. IV for definition) of the two temporal and six spatial bases spans a 12 dimensional spatial-temporal space, which corresponds to a parameterization of

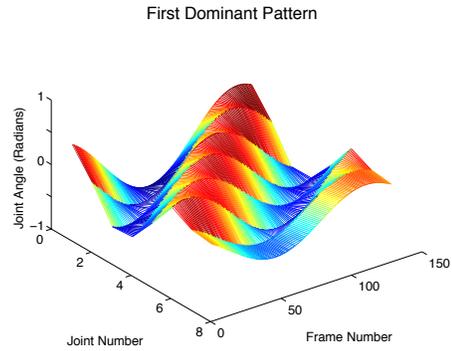


Fig. 5. Wave surface of the first principal component. Note this surface varies smoothly compared to Fig. 3.

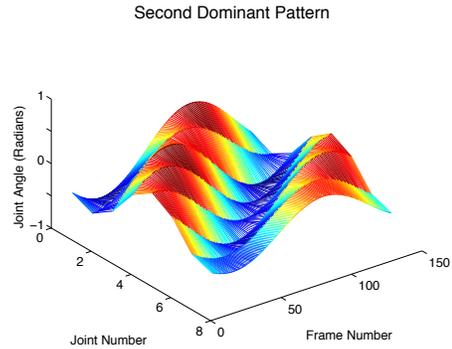


Fig. 6. Wave surface of the second principal component.

all the shape change sequences. The final reconstruction accuracy with respect to the number of bases used in the third behavioral direction is plotted in Fig. 4. We observed that two behavior-bases  $V^{(3)} \in \mathbb{R}^{12 \times 2}$  are sufficient to achieve a good approximation ( $> 75\%$ ). This result implies that the observed biological motions are the compositions of two motion patterns (motion modes), each of which is a function parameterized as the linear combination of 12 spatial-temporal bases. Fig.5 and Fig.6 show the wave surfaces of the two extracted motion patterns. The small number of modes associated with observed biological behaviors further suggests that snakes achieve high level control objectives through coupled low level control of joint motions. This biological insight is useful for developing control schemes for high DoFs systems.

## III. ROBOT VALIDATION

In the previous section, we demonstrated the successful application of the CBAF algorithm to trajectory data of biological sidewinding. CBAF identified two dominant motion patterns which capture the variations of the shape change sequences observed from different animal behaviors. Because the extracted motion patterns are represented as analytical forms, we were able to derive compact, well-parameterized families of gaits can be executed on snake robots.

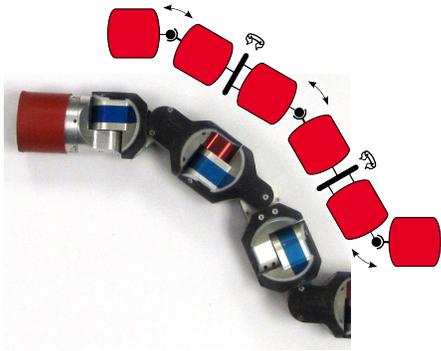


Fig. 7. The modular snake robot (shown in the lower portion of the picture). The cartoon above shows that the joints of the modular snake robot actuate alternatively in the dorsal and lateral directions.

### A. Platform

We conducted robot experiments on our 16 DoFs modular snake robot. The robot consisted of a chain of 16 identical modules with a specialized head and tail, see Fig. 7. The robot's joint axes were arranged such that neighboring joints alternate between actuating the dorsal and lateral degrees of freedom. Joint-level position control was implemented, using an on-board micro-controller, in each module to track desired reference trajectories.

### B. Prior Work on Sidewinding

Motion generation techniques that allow snake robots to execute sidewinding motions have been long investigated. The *gait equation* found in [21] is one technique for generating the sidewinding gait. The gait equation couples the robot's DoFs and expresses the temporal and spatial variations of joint angles as sinusoidal functions of time. In comparison to biological models, the gait equation produces motions with relatively small joint amplitudes. In addition, the gait equation is only capable of producing straight-line sidewinding, while the biological system can turn while sidewinding. In [10], it was shown that the limitations of the gait equation impedes generation of more complex modes of locomotion. The author proposed an extended form of the gait equation [10] in order to produce turning motion. Unfortunately, the derivation of the extended gait equation is based on a small angle approximation which limits the robot to move only through small amplitude motions.

Geometric modeling is a more expressive alternative than the gait equation for modeling sidewinding motion. The derivation of the geometric model starts by assuming that the robot is abstracted as a three-dimensional curve, called a *backbone curve* [4]. Sidewinding is modeled as a helical tread wrapping around the surface of an elliptically deformed cylinder [12] for linear motion or a cone [9] for turning. A fitting algorithm [11], which maps a discrete mechanism onto a continuous backbone curve, is employed to compute the discrete joint angles of the mechanical system. The resultant joint trajectories are represented in the form of lookup tables. The

drawbacks of this approach are the reliance on interpolation for smooth control inputs and its inability to generalize to robots with different numbers of joints.

### C. Gait Derivation

The motion generation technique we present in this work overcomes many of the limitations of the methods discussed in Sec. III-B. As pointed out, the gait expressions derived from CBAF are analytical and are extendable to a wide range of potential locomotive modes. The derivation of the CBAF gait expressions is as follows. Recall from the previous section that shape change information contained in every biologically generated data sequence can be captured via a linear combination of two motion patterns which are analytically represented. Hence, the joint trajectories of a gait are specified by only two coefficients,

$$\mathcal{A}^{\vec{c}} = \left( V^{(3)} \vec{c} \right) \cdot \left( \mathcal{I} \times_1 \left( V^{(1)} \right)^T \times_2 \left( V^{(2)} \right)^T \right)_{(3)}, \quad (3)$$

where  $\vec{c}$  is a  $2 \times 1$  column vector which contains the two coefficients associated with the two extracted motion patterns,  $\mathcal{I} \in \mathbb{R}^{6 \times 2}$  is a constant matrix of elements all equal to 1 and  $(\dots)_{(3)}$  is a mathematical operator called *unfolding* (see Sec. IV for more details). The outer product of the spatial bases in  $V^{(1)}$  and temporal bases in  $V^{(2)}$  span a 12 dimensional spatial-temporal space. And, the behavioral bases  $V^{(3)}$  contain the weights of the corresponding 12 spatial-temporal components. Because here the spatial-bases in  $V^{(1)}$  and temporal-bases in  $V^{(2)}$  are analytical functions (Fourier bases), Eqn. 3 is a smooth, analytical function describing the spatial-temporal variations of joint angles.

The expression derived above is for in-plane motions, which corresponds to control inputs for the lateral degrees of freedom. Studies in snake robot locomotion [21] and biological sidewinding [14] both suggest that the important vertical undulatory motion can be established by adding a  $\frac{\pi}{4}$  phase shift to  $\mathcal{A}^{\vec{c}}$  then multiplying a coefficient  $\eta$ . In practice, we set  $\eta = 0.2$ .

### D. Replaying Biological Motion

To playback the biological behaviors, the recorded data sequences are projected onto the parameterized motion patterns. According to Theorem. IV.1, the coefficients of a gait which replays observed biological motions are computed as

$$w^j = \left( V^{(3)} \right)^T \cdot \left( \left( \mathcal{A}^j \times_1 V^{(1)} \times_2 V^{(2)} \right)_{(3)} \right)^T, \quad (4)$$

where  $j$  is the index of the data sequence. The resultant gait expression is

$$\hat{\mathcal{A}}^j = \left( V^{(3)} w^j \right) \cdot \left( \mathcal{I} \times_1 \left( V^{(1)} \right)^T \times_2 \left( V^{(2)} \right)^T \right)_{(3)}. \quad (5)$$

Depending on which data sequence  $\mathcal{A}^j$  has been used, the reproduced motion exhibits different behaviors. Fig. 8 shows one example of the trajectories traced out by the robot compared to its biological counterpart. To study the relation between the

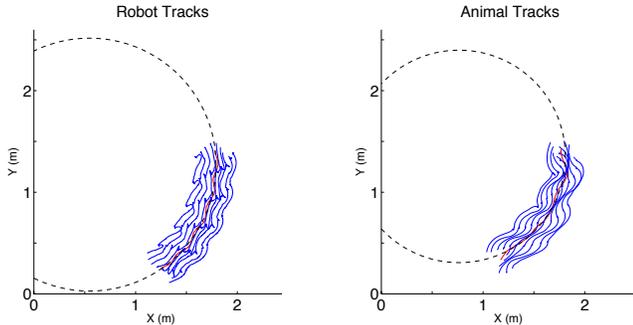


Fig. 8. Comparison between the markers traces on the robot and snake. The blue curves denote the marker trajectories, the red curves denote the COM trajectories and the dashed circles are the best fit circles to the COM trajectories.

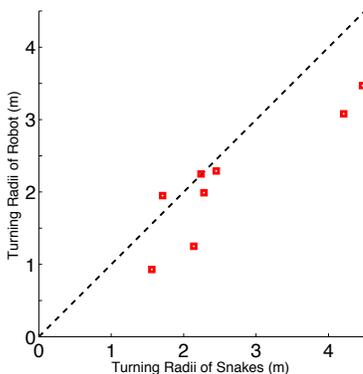


Fig. 9. Comparison of the radii of COM trajectories for snake trials and robots emulating the gait of these trials. Note that the points are close to but beneath the 1:1 line (dashed), indicating that robots accurately captured the behavior and even achieved slightly sharper turns than the snakes.

robot motions and the corresponding biological motions, we fit circles to the center of mass (COM) trajectories (approximated as the average position of the markers), then we compared the turning radii, as shown in Fig. 9. The turning radii of the robot trajectories do not exactly match with that of the animals, possibly due to the different body length and joint resolution. However, the fact that the robot turning radii correlates with the animals indicates that the parameterized gait expression indeed captures the key elements which contribute to the generation of versatile sidewinding behaviors.

### E. Beyond Biological Motion

With the parameterized gait expression we were able to execute biologically inspired motions which were not simply generated by playing back recorded biological data. By tuning the two parameters  $\vec{c}$  in the Eqn. 3, we were able to steer sidewinding motion. We observed that the motions built on biological motion patterns outperforms the *conical sidewinding* model [10]. Eqn. 3 is able to make the snake robot turn faster than conical sidewinding (measured as the change of heading in one gait cycle). We hand tuned the coefficients  $\vec{c}$  to find a gait for sharp turning motion and compared it with

conical sidewinding at maximal turning rate. On average (20 trials each), Eqn. 3 reorients the robot  $55.2 \pm 6.0$  degrees vs.  $30.1 \pm 2.9$  degrees of conical sidewinding in one gait cycle. This improvement directly benefits applications like search and rescue where a robot must make tight turns in confined spaces.

## IV. CONDITIONED BASIS ARRAY FACTORIZATION

This section presents the technical details of a novel modal decomposition algorithm, called *conditioned basis array factorization* (CBAF). This algorithm can be applied to multi-dimensional data array to extract patterns which are represented in analytical forms.

### A. Related Work

Many pattern extraction techniques have been previously proposed. Principal component analysis (PCA) has been used to extract human gait patterns for the purpose of gait recognition [23]. PCA has also been used in applications including textual information retrieval [3], facial recognition [24] and image classification [20]. One obvious limitation of PCA is the requirement for data to be written in vector format. In order to apply PCA, multi-dimensional data arrays first have to be flattened into vectors. Vectorizing multi-dimensional data array breaks data regularity and increases the risk of over-fitting [2]. A bilinear model was proposed to handle two-dimensional data (matrix), and was employed to study gait content and style [19]. Both PCA and the bilinear model cannot be directly applied to multi-dimensional data arrays without breaking original data structure. Array factorization [15] has been developed to process multi-dimensional data arrays and has a large range of applications including computer vision [22], data mining [1] and graph analysis [16]. Higher-order singular value decomposition (HOSVD) [8] and alternative least square (ALS) [17, 18] are two popular algorithms for array factorization. However, they are computationally intractable because they both take SVD as a subroutine to fit basis. Besides computational disadvantages, patterns extracted using these algorithms do not have an analytical representation, impeding the ability to gain intuitive insights for the purpose of understanding the fundamental components of biological locomotions.

### B. Mathematical Preliminaries

We here provide a brief review of array factorization. For a comprehensive treatment on array factorization, readers are referred to [15]. We borrow the tools from the array factorization community for the purpose of extracting dominant motion patterns from shape change sequences collected from biological snakes which are represented as a multi-dimensional (multi-mode) array  $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$ . In many circumstances, data are naturally represented as multi-dimensional arrays. Using snake locomotion as an example, shape deformations sampled at different positions (mode-1) along the snake, at different time steps (mode-2), and from different behaviors (mode-3) form a 3-dimensional array. In order to extend existing

techniques in linear algebra to multi-dimensional arrays, the following operations have been defined.

**Definition** The *mode-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 *mode-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, for a 2-dimensional array (matrix), the column vectors are the mode-1 vectors and the row vectors are the mode-2 vectors. One can view the unfolding operation as extracting all the mode-k vectors from  $\mathcal{A}$  and collecting them as the column vectors in  $A_{(k)}$ . By unfolding a multi-dimensional array into a matrix, techniques like PCA can now be applied to reduce the dimensionality of the vector space spanned by the mode-k vectors.

**Definition** The *mode-k folding* is the reverse operation of mode-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 multi-dimensional array  $\mathcal{A} \in \mathbb{R}^{I_1 \times \dots \times I_N}$ .

The mode-k dimensionality of an array can be reduced through an operation called mode-k multiplication.

**Definition** The *mode-k multiplication* between a multi-dimensional 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  $\mathcal{D} = \mathcal{A} \times_k U \in \mathbb{R}^{I_1 \times \dots \times I_{k-1} \times R_k \times I_{k+1} \times \dots \times I_N}$ , whose elements can be computed as

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

Mode-k multiplication  $\mathcal{A} \times_k U$  is equivalently represented as  $\text{fold}_k(U^T A_{(k)})$ , where  $T$  denotes matrix transpose. Intuitively, this process can be understood as first projecting all the mode-k vectors onto a vector space spanned by the column vectors of  $U$ , then reshaping the resultant matrix back into a high-dimensional array. If  $R_k < I_k$ , the mode-k dimensionality is reduced from  $I_k$  to  $R_k$ .

**Definition** The *inner product* between two multi-dimensional 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 multi-dimensional arrays  $\mathcal{A}$  and  $\mathcal{B}$  are called *orthogonal* if  $\langle \mathcal{A}, \mathcal{B} \rangle = 0$ .

**Definition** The *norm* of a multi-dimensional array  $\mathcal{A}$  is

$$\|\mathcal{A}\| = \langle \mathcal{A}, \mathcal{A} \rangle^{\frac{1}{2}} \quad (6)$$

**Definition** The *outer product* of two multi-dimensional 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

$$d_{i_1 \dots i_p, j_1 \dots j_q} = a_{i_1 \dots i_p} b_{j_1 \dots j_q} \quad (7)$$

This is the operator used to construct a high-dimensional array given basis vectors along every mode.

**Definition** A multi-dimensional array formed by the outer product of  $N$  non-zero basis vectors (one along each mode)  $\vec{u}^{(1)} \in \mathbb{R}^{I_1}, \dots, \vec{u}^{(N)} \in \mathbb{R}^{I_N}$  is called a *N-dimensional 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 multi-dimensional arrays. An array can always be represented by the linear combination of a set of rank-1 arrays.

**Definition** In this paper, we call a matrix  $U^{(k)} \in \mathbb{R}^{I_k \times R_k}$  the *mode-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 also call it *mode-k truncated basis matrix*.

We consider the column vectors of  $U^{(k)}$  as a set of basis vectors along mode-k.

With all the definitions above, we now show some important properties of multi-dimensional arrays.

**Theorem IV.1** ([5]). *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 can be computed as*

$$\|\mathcal{A} - \mathcal{T}\|^2 = \|\mathcal{A}\|^2 - \|\mathcal{T}\|^2 \quad (8)$$

Theorem IV.1 describes how to compute an optimal reconstruction given a set of rank-1 arrays.

**Theorem IV.2** ([5]). *Given (truncated) basis matrices for each mode  $V^{(1)} \in \mathbb{R}^{I_1 \times R_1}, \dots, V^{(N)} \in \mathbb{R}^{I_N \times R_N}$ , the array*

$$\mathcal{C} = \mathcal{A} \times_1 V^{(1)} \times_2 \dots \times_N V^{(N)}, \quad (9)$$

*is called the (truncated) core array. And the best rank- $(R_1, \dots, R_N)$  approximation to the original array is*

$$\hat{\mathcal{A}} = \mathcal{C} \times_1 (V^{(1)})^T \times_2 \dots \times_N (V^{(N)})^T \quad (10)$$

Theorem IV.2 shows how to find an optimal low dimensional approximation to an array with given basis matrices. 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,  $V^{(1)}, \dots, V^{(N)}$ .

**Theorem IV.3** ([7]). *Minimizing the reconstruction error  $\|\mathcal{A} - \hat{\mathcal{A}}\|$  is equivalent to maximizing  $\|\mathcal{C}\|$ .*

### C. Conditioned Basis Array Factorization

Our conditioned basis array factorization (CBAF) algorithm circumvents the limitation of ALS by using conditioned bases [2]. We prove 1) finding an error minimizing factorization is NP-hard and 2) the sub-optimality guarantee of CBAF. We restrict our proofs to three-dimensional arrays, though they can be easily generalized to higher order cases.

The purpose of array factorization is to find a low rank approximation to a multi-dimensional array while minimizing the reconstruction error. ALS spends a large amount of computational effort in fitting bases to minimize reconstruction error, running the risk of over-fitting.

One means to remedy the situation is to use *conditioned (pre-determined) basis* [2]. The choice of a conditioned basis matrix might come from domain knowledge for a specific application or based on an evaluation of different candidates. For example, the Fourier basis is often a good candidate for cyclic data, like gaits. With known bases, array factorization is reduced from a basis fitting problem to a basis selection problem (choose a subset of the column vectors from the conditioned basis matrix), which can be solved more efficiently. One additional benefit is that the patterns extracted from a multi-dimensional array can be represented in analytical forms. And these analytical forms carry intuitive meanings for understanding data variations.

Basis selection aims to minimize the reconstruction error  $\|\mathcal{A} - \hat{\mathcal{A}}\|$  or, equivalently, maximize the norm of the truncated core array  $\|\hat{\mathcal{C}}\|$ . Given the conditioned basis matrices  $U^{(1)} \in \mathbb{R}^{I_1 \times R_1}$ ,  $U^{(2)} \in \mathbb{R}^{I_2 \times R_2}$  and  $U^{(3)} \in \mathbb{R}^{I_3 \times R_3}$ , the core array can immediately be computed as,

$$\mathcal{C} = \mathcal{A} \times_1 U^{(1)} \times_2 U^{(2)} \times_3 U^{(3)}. \quad (11)$$

At this point, no reconstruction error has been introduced because all the basis matrices are of full rank. 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

$$\hat{\mathcal{C}} = \mathcal{A} \times_1 V^{(1)} \times_2 V^{(2)} \times_3 V^{(3)} \quad (12)$$

Note the only difference between Eqn. 11 and Eqn. 12 is that the basis matrices  $V^{(k)}$  in Eqn. 12 are sub-matrices of  $U^{(k)}$  in Eqn. 11. Hence, Eqn. 11 is performing a change of coordinates along every mode. Eqn. 12 can be understood as choosing subspaces in the new coordinate frames.

The goal of basis truncation then can be restated as constructing three subsets  $J_1 \subseteq \{1, 2, \dots, I_1\}$ ,  $J_2 \subseteq \{1, 2, \dots, I_2\}$  and  $J_3 \subseteq \{1, 2, \dots, I_3\}$  s.t.  $|J_1| = R_1$ ,  $|J_2| = R_2$ ,  $|J_3| = R_3$  which maximize

$$\|\hat{\mathcal{C}}\|^2 = \sum_{i_1 \in J_1} \sum_{i_2 \in J_2} \sum_{i_3 \in J_3} c_{i_1, i_2, i_3}^2 \quad (13)$$

We call this problem *low rank optimal reconstruction (LROR)*. It turns out that LROR is NP-Hard, see proof in Appendix. A1. Though LROR is NP-Hard, our *conditioned basis array factorization (CBAF)* algorithm efficiently finds a low dimensional approximation to a given multi-dimensional array with a provable sub-optimality bound. Before describing the algorithm, we first define a quantity called *energy* as a heuristic to estimate the important of a basis. The energy of a mode- $k$

### Algorithm 1: Conditioned Basis Array Factorization

```

1 compute core array:  $\mathcal{C}$ ;
2 for mode  $k = 1$  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

```

conditioned basis  $\bar{u}_{i_k}^{(k)}$  is computed as

$$\mathcal{E}_{i_k}^{(k)} = \sum_{\substack{j_k=i_k \\ j_1, \dots, j_N}} c_{j_1, \dots, j_N}^2 \quad (14)$$

Intuitively, the energy of a basis can be understood as the amount of data variations captured by the basis vector. The CBAF algorithm is described in Algorithm. 1. In Appendix A2, we prove CBAF guarantees a factor-3 suboptimal solution. This theoretical guarantee ensures the quality of the extracted motion patterns for kinematic gait modeling.

## V. CONCLUSION

The successful application of the CBAF algorithm verifies that studying shape changes of biological organisms is an effective means of generating gaits for physically similar robots whose locomotion is approximately kinematic. The analytical representation of extracted motion patterns allowed us to derive compact, analytical gait expression for a snake robot to reproduce observed biological behaviors, as well as produce a variety of motions beyond mimicking biological snakes. The underlying low dimensionality associated with the observed biological sidwinding motions suggests a bio-inspired method to control high DoF systems through coupling the joint movements.

The kinematic motion patterns extracted from biological sidwinding motivate us to further investigate their physical meanings and the underlying control principles of biological systems. With CBAF as an analysis tool, we can more closely inspect the complicated shape changes of biological systems in a concise way and gain further insights into the working principles of kinematic locomotion of biological systems.

## APPENDIX

### A. Conditioned Basis Array Factorization

1) *NP-Hardness of LROR*: To prove *low rank optimal reconstruction (LROR)* 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 3-dimensional array  $\mathcal{A}$ , conditioned basis matrices  $U^{(1)} \in \mathbb{C}^{I_1 \times R_1}$ ,  $U^{(2)} \in \mathbb{C}^{I_2 \times R_2}$ ,  $U^{(3)} \in \mathbb{C}^{I_3 \times R_3}$  and integers  $R_1, R_2, R_3$ , does there exist a rank- $(R_1, R_2, R_3)$  reconstruction with the squared norm of the truncated basis array  $\hat{\mathcal{C}}$  greater 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 *Exact Node Cardinality Decision Problem (ENCD)* [6] of biclique. ENCD is stated as follows: Given a bipartite graph  $G(V_1 \cup V_2, E)$  and two positive integers  $R_1, R_2$ , does  $G$  contain 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)  $A \in \mathbb{C}^{|V_1| \times |V_2|}$ . 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  $U^{(1)} \in \mathbb{C}^{|V_1| \times |V_1|}$  and  $U^{(2)} \in \mathbb{C}^{|V_2| \times |V_2|}$  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,

$$\sum_{v_1 \in V'_1, v_2 \in V'_2} a_{v_1, v_2} = R_1 R_2 = \sum_{v_1 \in V'_1, v_2 \in V'_2} a_{v_1, v_2}^2 \quad (15)$$

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 sub-array  $A' \in \mathbb{C}^{R_1 \times R_2}$  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. ■

2) *Sub-optimality Guarantee of CBAF:* Given the conditioned basis matrices  $U^{(1)} \in \mathbb{C}^{I_1 \times I_1}$ ,  $U^{(2)} \in \mathbb{C}^{I_2 \times I_2}$  and  $U^{(3)} \in \mathbb{C}^{I_3 \times I_3}$ , CBAF constructs sub-matrices  $V^{(1)} \in \mathbb{C}^{I_1 \times R_1}$ ,  $V^{(2)} \in \mathbb{C}^{I_2 \times R_2}$  and  $V^{(3)} \in \mathbb{C}^{I_3 \times R_3}$ . Denote the reconstructed array after mode-1 basis truncation as

$$\mathcal{A}^{(1)} = \mathcal{A} \times_1 V^{(1)} \times_1 (V^{(1)})^T \quad (16)$$

The residual array is

$$\mathcal{B}^{(1)} = \mathcal{A} - \mathcal{A}^{(1)} \quad (17)$$

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

$$\epsilon_{(1)} = \|\mathcal{B}^{(1)}\| \quad (18)$$

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

$$\mathcal{A}^{(1,2)} = \mathcal{A}^{(1)} \times_2 V^{(2)} \times_2 (V^{(2)})^T \quad (19)$$

$$\mathcal{B}^{(2)} = \mathcal{A}^{(1)} - \mathcal{A}^{(1,2)} \quad (20)$$

$$\epsilon_{(2)} = \|\mathcal{B}^{(2)}\| \quad (21)$$

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

$$\mathcal{A}^{(1,2,3)} = \mathcal{A}^{(1,2)} \times_3 V^{(3)} \times_3 (V^{(3)})^T \quad (22)$$

$$\mathcal{B}^{(3)} = \mathcal{A}^{(1,2)} - \mathcal{A}^{(1,2,3)} \quad (23)$$

$$\epsilon_{(3)} = \|\mathcal{B}^{(3)}\| \quad (24)$$

**Lemma A.2.** *The reconstruction error  $\epsilon$  of CBAF is bounded by,*

$$\max\{\epsilon_{(1)}, \epsilon_{(2)}, \epsilon_{(3)}\} \leq \epsilon \leq \epsilon_{(1)} + \epsilon_{(2)} + \epsilon_{(3)} \quad (25)$$

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

$$\begin{aligned} \epsilon &= \|\mathcal{A} - \mathcal{A}^{(1,2,3)}\| = \|\mathcal{A} - \mathcal{A}^{(1,2)} + \mathcal{B}^{(3)}\| \\ &= \|\mathcal{A} - \mathcal{A}^{(1)} + \mathcal{B}^{(2)} + \mathcal{B}^{(3)}\| \\ &= \|\mathcal{B}^{(1)} + \mathcal{B}^{(2)} + \mathcal{B}^{(3)}\| \\ &\leq \epsilon_{(1)} + \epsilon_{(2)} + \epsilon_{(3)} \quad (\text{Triangle Inequality}) \quad (26) \end{aligned}$$

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. IV.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.3.** *The reconstruction error of the optimal solution to LROR is bounded,*

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

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

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

■

**Theorem A.4.** *CBAF is a factor-3 approximation algorithm.*

*Proof:* From Lemma. A.2 and Lemma. A.3, it can be shown,

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

■

## REFERENCES

- [1] Evrim Acar, Seyit A Camtepe, and Bülent Yener. Collective sampling and analysis of high order tensors for chatroom communications. In *Intelligence and security informatics*, pages 213–224. Springer, 2006.
- [2] Ijaz Akhter, Tomas Simon, Sohaib Khan, Iain Matthews, and Yaser Sheikh. Bilinear spatiotemporal basis models. *ACM Trans. Graph.*, 31(2):17:1–17:12, April 2012. ISSN 0730-0301. doi: 10.1145/2159516.2159523. URL <http://doi.acm.org/10.1145/2159516.2159523>.
- [3] Michael W Berry, Susan T Dumais, and Gavin W O’Brien. Using linear algebra for intelligent information retrieval. *SIAM review*, 37(4):573–595, 1995.
- [4] Joel W Burdick, Jim Radford, and Gregory S Chirikjian. A’sidewinding’ locomotion gait for hyper-redundant robots. *Advanced Robotics*, 9(3):195–216, 1994.
- [5] Jie Chen and Yousef Saad. On the tensor svd and the optimal low rank orthogonal approximation of tensors. *SIAM Journal on Matrix Analysis and Applications*, 30(4):1709–1734, 2009.
- [6] Milind Dawande, Pinar Keskinocak, Jayashankar M Swaminathan, and Sridhar Tayur. On bipartite and multipartite clique problems. *Journal of Algorithms*, 41(2):388–403, 2001.
- [7] Lieven De Lathauwer, Bart De Moor, and Joos Vandewalle. On the best rank-1 and rank-( $r_1, r_2, \dots, r_m$ ) approximation of higher-order tensors. *SIAM Journal on Matrix Analysis and Applications*, 21(4):1324–1342, 2000.
- [8] Lieven De Lathauwer, Bart De Moor, and Joos Vandewalle. A multilinear singular value decomposition. *SIAM journal on Matrix Analysis and Applications*, 21(4):1253–1278, 2000.
- [9] Chaohui Gong, Ross L Hatton, and Howie Choset. Conical sidewinding. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 4222–4227. IEEE, 2012.
- [10] Chaohui Gong, M.J. Travers, Xiaozhou Fu, and H. Choset. Extended gait equation for sidewinding. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 5162–5167, 2013. doi: 10.1109/ICRA.2013.6631315.
- [11] Ross L Hatton and Howie Choset. Generating gaits for snake robots: annealed chain fitting and keyframe wave extraction. *Autonomous Robots*, 28(3):271–281, 2010.
- [12] Ross L Hatton and Howie Choset. Sidewinding on slopes. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 691–696. IEEE, 2010.
- [13] Shigeo Hirose. Biologically inspired robot. *Oxford University Press*, 1993.
- [14] Bruce C Jayne. Kinematics of terrestrial snake locomotion. *Copeia*, pages 915–927, 1986.
- [15] Tamara G Kolda and Brett W Bader. Tensor decompositions and applications. *SIAM review*, 51(3):455–500, 2009.
- [16] Tamara G Kolda, Brett W Bader, and Joseph P Kenny. Higher-order web link analysis using multilinear algebra. In *Data Mining, Fifth IEEE International Conference on*, pages 8–pp. IEEE, 2005.
- [17] Pieter M Krooneberg. *Three-mode principal component analysis: Theory and applications*, volume 2. DSWO press, 1983.
- [18] Pieter M Kroonenberg and Jan De Leeuw. Principal component analysis of three-mode data by means of alternating least squares algorithms. *Psychometrika*, 45(1):69–97, 1980.
- [19] Chan-Su Lee and A. Elgammal. Gait style and gait content: bilinear models for gait recognition using gait re-sampling. In *Automatic Face and Gesture Recognition, 2004. Proceedings. Sixth IEEE International Conference on*, pages 147–152, 2004. doi: 10.1109/AFGR.2004.1301522.
- [20] Pedro Quelhas, Florent Monay, J-M Odobez, Daniel Gatica-Perez, Tinne Tuytelaars, and Luc Van Gool. Modeling scenes with local descriptors and latent aspects. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 1, pages 883–890. IEEE, 2005.
- [21] Matthew Tesch, Kevin Lipkin, Isaac Brown, Ross Hatton, Aaron Peck, Justine Rembisz, and Howie Choset. Parameterized and scripted gaits for modular snake robots. *Advanced Robotics*, 23(9):1131–1158, 2009.
- [22] M Alex O Vasilescu and Demetri Terzopoulos. Multilinear analysis of image ensembles: Tensorfaces. In *Computer Vision ECCV 2002*, pages 447–460. Springer, 2002.
- [23] Liang Wang, Tieniu Tan, Huazhong Ning, and Weiming Hu. Silhouette analysis-based gait recognition for human identification. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 25(12):1505–1518, 2003.
- [24] Wenyi Zhao, Arvindh Krishnaswamy, Rama Chellappa, Daniel L Swets, and John Weng. Discriminant analysis of principal components for face recognition. In *Face Recognition*, pages 73–85. Springer, 1998.