Auton Robot (2010) 28: 271–281
DOI 10.1007/s10514-009-9175-2

Generating gaits for snake robots: annealed chain fitting and keyframe wave extraction

Ross L. Hatton · Howie Choset

Received: 1 April 2009 / Accepted: 18 December 2009 / Published online: 31 December 2009

Abstract Snake robots have many degrees of freedom, which makes them both extremely versatile and complex to control. They are often controlled with *gaits*, coordinated cyclic patterns of joint motion. Using gaits simplifies the design of high-level controllers, but shifts the complexity burden to designing the gaits. In this paper, we address the gait design problem by introducing two algorithms: *Annealed chain fitting* and *Keyframe wave extraction*. Annealed chain fitting efficiently maps a continuous backbone curve describing the three-dimensional shape of the robot to a set of joint angles for a snake robot. Keyframe wave extraction takes joint angles fit to a sequence of backbone curves and identifies parameterized periodic functions that produce those sequences. Together, they allow a gait designer to conceive a motion in terms three-dimensional shapes and translate them into easily manipulated wave functions, and so unify two previously disparate gait design approaches. We validate the algorithms by using them to produce rolling and sidewinding gaits for crawling and climbing, with results that match previous empirical investigations.

Keywords Snake robot · Gait · Curve discretization

1 Introduction

Snake robots are actuated kinematic chains that locomote via coordinated bending of their bodies. Typically, snake robots possess many active joints which gives them great versatility and freedom of movement. This versatility comes at a cost, as it is infeasible for an operator to individually control each actuator. A common solution to this difficulty is to define parameterized *gaits*, or patterned joint motions, and have the operator select among them to direct the robot. The key question then becomes how to design useful gaits.

Two dominant approaches to gait design in the literature specify gaits either with animated "backbone curves" describing the macroscopic three-dimensional shape of the robot, or by explicitly setting the joint angles as functions of joint number and time. Each of these approaches has strengths and weaknesses. Backbone curves let the gait designer think in terms of real-world geometry, but any controller based on them must incorporate a nonlinear fitting operation to generate the low-level actuator inputs. These fitting algorithms have historically assumed a mechanism with bidirectional bending capability at each joint, which limits their applicability. Conversely, the wave approaches operate directly on the actuator inputs, but choosing functions and parameters to generate desired real-world effects is difficult and non-intuitive.

In this paper, we take a hybrid approach to the gait design problem, drawing on the strengths of the two techniques described above while mitigating their weaknesses. First, we introduce a new fitting algorithm, *annealed chain fitting*, which does not require the bidirectional bending capabilities of previous algorithms and is thus useful for a wider domain of mechanism configurations. We then apply this algorithm to series of static backbone curves, generated by discretizing desired moving backbone curves in time. Borrowing a term from the animation literature, we refer to these static curves as "key frames." Assembling the joint angles from the fitting operations produces a trajectory of positions for each joint during the gait. We gain further insight into the operation

R.L. Hatton (✉)
Robotics Institute and Mechanical Engineering,
Carnegie Mellon University, Pittsburgh, USA
e-mail: rlhatton@cmu.edu

H. Choset
Robotics Institute, Carnegie Mellon University, Pittsburgh, USA
e-mail: choset@cmu.edu

of the gait by collecting these trajectories into discrete functions of joint number and time and then using our *keyframe wave extraction* process to identify parameterized analytical functions that minimally capture the form of the discrete functions. Taken together, these two procedures allow us to start by specifying gaits as high-level three-dimensional shapes and finish with low-level patterns of joint angles. The parameters of these patterns can be manipulated in real time and can thus be tightly integrated into motion controllers.

2 Background

Snake robots have been studied since at least 1971, with Hirose's pioneering work on the Active Cord Mechanism (ACM) (Hirose 1993). While this and other early snake robots were confined to planar motion, much recent effort has been directed towards mechanisms that can assume full three-dimensional shapes. Notable developments in this area include Yim's Polybot (Yim et al. 2001; Zhang et al. 2003a, 2003b); Mori and Hirose's ACM-R3 (Mori and Hirose 2002), our modular snakes ("modsnakes"), shown in Fig. 1 (Wright et al. 2007), Gonzalez-Gomez et al.'s Hypercube (Gonzalez-Gomez et al. 2006, 2007), and Goldman and Hong's HyDRAS (Goldman and Hong 2007, 2008).

The theoretical contribution in this work draws on two previous approaches to snake robot gait design: backbone approaches, which specify the gait as a moving curve, and wave approaches, which directly oscillate the joint angles. Hirose's early work on snake robots identified the *serpenoid curve* (Hirose 1993) as the fundamental shape function describing the backbones of biological snakes while executing common gaits, and used this curve as the basis for robotic gaits. Chirikjian and Burdick (1991, Chirikjian 1992) drew on this notion of a backbone curve and developed modal functions for describing arbitrary backbone curves and algorithms for fitting discrete serial mechanisms to these curves. This work paved the way for analysis of complex gait motions, such as Burdick, Radford, and Chirikjian's study of sidewinding (Burdick et al. 1993). More recently, Yamada and Hirose have proposed an alternate backbone curve formulation (Yamada and Hirose 2006) and Goldman and Hong (2007, 2008) have used a variation on backbone fitting to examine the kinematics of their robot climbing a pole. The present work is an expansion of our preliminary work on annealed chain fitting, the results of which were reported in Hatton and Choset (2009).

A second branch of gait development, also inspired by Hirose's work, has taken a lower-level, wave-based approach. The joint angles in a serpenoid-curve gait are specified by a sinusoidal traveling wave function, and, by extension, other gaits can be produced by modifying the form of this wave (Sfakiotakis and Tsakiris 2007). We have found much success in applying a three-dimensional form of this approach involving "lateral" and "lifting" waves to generate crawling, climbing, and swimming gaits (Lipkin et al. 2007; Tesch et al. 2009); similar results have been demonstrated by several other researchers, including Yu et al. (2008) and Gonzalez-Gomez et al. (2006, 2007), the latter of whom used central pattern generators (CPGs) to generate the waves in a distributed fashion. The use of CPG-generated waves is becoming increasingly widespread in locomotion control and is thoroughly reviewed by Ijspeert (2008).

Fig. 1 Our modular snake ("modsnake") robots have sixteen joints, arranged to allow full workspace flexibility. In past work, we have demonstrated a variety of successful motion strategies for these robots, including traversal of both smooth and rough terrain, swimming, and pole climbing (shown here)

3 Gait design for modsnake robots

Our modsnake robots are constructed as kinematic chains of single-degree-of-freedom modules. Each module consists of a joint and a link, and the joint axis of each module is rotated around the central axis of the snake by 90° with respect to the previous joint, as illustrated in Fig. 2 (Wright et al. 2007). This geometry allows full three-dimensional motion while maintaining smaller individual links and providing greater robustness than equivalent designs combining pairs of modules into active universal joints. We use the convention that the nth link is distal to the nth joint, and that the angle of the nth joint thus determines the position of link n with respect to link $n-1$.

We control these robots by designing gaits which propagate waves along their bodies (Lipkin et al. 2007). The first stage in this wave-based approach is to divide the modules into odd and even sets by joint number, such that all the joint axes in each set are parallel when the snake robot is stretched out. We then take parameterized, periodic functions in time and joint number for each set and manipulate the parameters

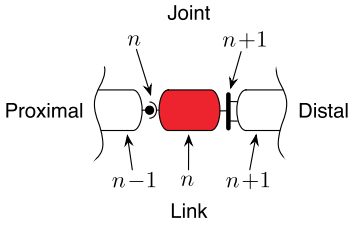


Fig. 2 Modsnake geometry. The nth link is distal to the nth joint, and together they make up the nth module. Each joint is rotated around the central axis of the snake by 90° with respect to the previous joint; in this illustration, joint n generates rotation in the plane, while joint $n+1$ generates rotation out of the plane

to design useful gaits. For example, a basic form of this approach is to define $\alpha(n,t)$, the angle of the nth joint at time t, as

$$\alpha(n,t) = \begin{cases} \beta_{\text{odd}} + A_{\text{odd}} \sin(\theta_{\text{odd}}) & \text{odd} \\ \beta_{\text{even}} + A_{\text{even}} \sin(\theta_{\text{even}} + \delta) & \text{even} \end{cases} \quad (1)$$

$$\theta_{\text{odd,even}} = (\Omega_{\text{odd,even}} n + \omega_{\text{odd,even}} t), \quad (2)$$

where β, A, θ, and δ are respectively offset, amplitude, frequency, and phase shift. The parameter Ω describes the spatial frequency of the macroscopic shape of the robot and the temporal component ω determines the frequency of the actuator cycles. This single wave model encompasses a wide variety of gaits, from slithering and sidewinding to the rolling helix used to climb the pole in Fig. 1, and is similar to the parameterized CPG functions used in Gonzalez-Gomez et al. (2006, 2007).

This wave-based approach is a powerful tool, as it greatly reduces the complexity of designing gaits. Rather than individually choosing a trajectory for each joint angle, the gait designer can work with a smaller set of parameters that apply across all the joints. It is not a complete solution, though, as it still leaves open the question of how to select regions of this parameter space that correspond to useful gait types.

To date, our efforts to find good choices of wave parameters, while fruitful, have been largely empirical. For instance, through experimentation we have determined that a phase shift of $\delta = \frac{\pi}{4}$ can be used to generate a sidewinding motion, while a phase shift of $\delta = \frac{\pi}{2}$ corresponds to rolling motions (Lipkin et al. 2007). Unfortunately, this empirical approach is limited in that it is difficult for a gait designer to think in terms of bending wave parameters; it would be much easier to think in terms of *backbone curves* (Burdick et al. 1993), *i.e.*, the three-dimensional shapes assumed by the robot. Without considerable experience, it is hard to translate the latter into the former, especially if the snake is convoluted enough that the "odd" and "even" joints no longer correspond to "horizontal" and "vertical" bending. It is even harder to identify when a conceived gait cannot be realized via the wave model in (1), and then to generate a new wave function with which to express it.

4 Annealed chain fitting

As translating from a three-dimensional backbone curve to a set of joint angles is difficult for a gait designer, the first stage of our approach automates this operation. While in principle it would be possible to solve directly for the set of joint angles which best fit the robot to the backbone curve, such an operation would be computationally prohibitive. Instead, we take an iterative approach, and progressively "sculpt" the robot onto the backbone in a series of smaller, more stable optimizations. If the robot had spherical or universal joints, as in Andersson (2008), this sculpting process would be trivially accomplished by sequentially setting each joint angle to place the distal end of the corresponding link directly on the curve. With an alternating-axis geometry, however, the alignment of a given joint axis with the curvature of the backbone (and thus the ability of that module to fit to the curve) depends on the joint angles of the preceding modules. Because our fitting algorithm must consequently allow for the adjustment of previously fit joint angles as each new module is considered, we refer to this process as *annealed chain fitting*.

4.1 Backbone curves

In our fitting algorithm, we specify the desired three-dimensional shape of the robot with continuous backbone curves. As did Burdick et al. (1993, Chirikjian and Burdick 1991), we take the backbone curve to be a spatial curve which has both an associated coordinate system along its length, such as a Frenet frame, and a *roll distribution*, which dictates the orientation of each portion of the robot's body with respect to this coordinate system.

The structure of our snake robots allows us to simplify the general backbone definition. The arrangement of rotary joints on the robot is *torsion free* (Nilsson 1998), in the sense that the robot can bend but not twist with respect to itself. This is a property of snake robots with universal joints or alternating-axis geometry, and is not shared with robots that have spherical joints or axes of rotation aligned with the "spine" of the mechanism. For this torsion-free configuration, we need only to consider the roll distribution of the first module, as it in turn dictates the roll distribution over the rest of the robot. Specifically, we encode this roll as the *twist angle* ϕ of the first module around the tangent line at the base of the curve, as illustrated in Fig. 3. Note that because the modules we are fitting to the curve have finite length, we use the tangent chord, rather than the instantaneous tangent vector at the base of the backbone.

4.2 Fitting algorithm

The annealed chain fitting algorithm starts by fixing the first module to the base of the backbone curve, with orientation

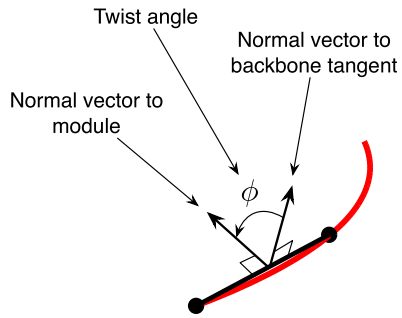


Fig. 3 The twist angle ϕ is the angle between a chosen normal vector on the backbone's tangent line and a chosen normal vector on the first module. Note that because the modules we are fitting to the curve have finite length, we use a chord of one module length, rather than the instantaneous tangent vector at the base of the backbone

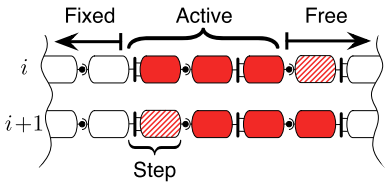


Fig. 4 Annealed chain fitting, with a window size of three and a step size of one. In each iteration, the active window steps along the snake robot. As the step size illustrated here is one module, one free module becomes active and one active module becomes fixed between iterations i and $i+1$

specified by the twist angle ϕ. The joint angles to fit the rest of the modules to the curve are then found by the following iterative procedure. At each iteration in the algorithm, we separate the joints of the snake robot into three categories: *fixed joints*, which have been previously fit to the curve and will not be changed; *active joints*, which are being fit to the curve; and *free joints*, which will be fit to the curve in future steps. As shown in Fig. 4, the joints are assigned to these categories via a moving *active window*. The size of this window determines the number of active joints, and the step size controls how many active joints become fixed (and how many free joints are made active) between each iteration.

Once the active modules have been selected for an iteration, their joint angles are optimized to fit their links as closely as possible to the backbone curve. In this paper, we find this best fit by minimizing the cost function

$$D_i(\alpha_{\text{active}}) = \sum_{\text{active}} d_j^2 + \kappa(\alpha_j), \tag{3}$$

i.e. the sum of squared distances of the distal ends of the modules from the backbone, with a penalty function

$$\kappa(\alpha) = \max(0, (\text{abs}(\alpha) - 0.8)^3) \tag{4}$$

to push the solution away from extreme angles that "bunch up" along the curve. This objective function essentially creates "sliding springs" which pull the modules towards the curve, as in Fig. 5. This sliding action means that the point on the backbone that attracts each module varies over the optimization, so we do not attempt to analytically solve for the best fit and instead use a numeric solver, *e.g.* the Nelder-Mead simplex algorithm (Nelder and Mead 1965) to find the optimal joint angles.

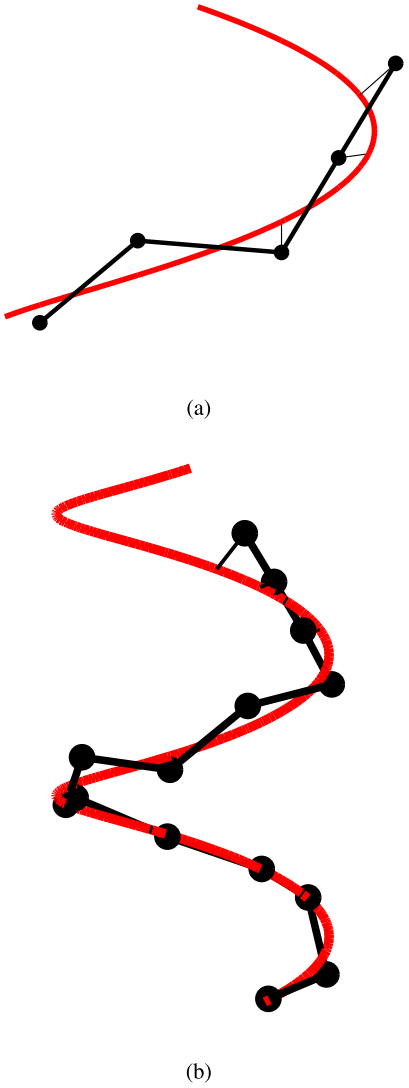


Fig. 5 Fitting a snake robot to a helix. (**a**) The three active (distal) modules are pulled towards the backbone by the thin attraction lines connecting them to the curve. The fixed modules have already been fit to the curve and are not adjusted in subsequent iterations. (**b**) As the fitting progresses along the snake robot, the alternating axes of bending produce a characteristic zigzag pattern centered on the backbone curve

Careful consideration of the active window and step sizes is required. For an N-module robot, with active window and step sizes a and s respectively, the annealed chain fitting algorithm requires searching $\frac{N-a}{s}$ a-dimensional spaces. A large value of a increases the degrees of freedom in the active region, and hence its ability to fit to the backbone.

However, this benefit comes at the cost of exponentially increasing the size of each search space while only linearly decreasing the number of searches. Similarly, large values of s reduce the number of searches but also result in several modules being added in each iteration, which can trap the system in undesirable local minima of the objective function, such as bridging over the coils of a helix. Also, too large a ratio of $\frac{s}{a}$ reduces the number of iterations in which a given module is allowed to move and thus reduces the "annealing" properties of the algorithm. Given these considerations, along with observed diminishing returns in fit quality for $a > 3$, we used values of $a = 3$ and $s = 1$ for the examples in this paper.

4.3 Boundary conditions

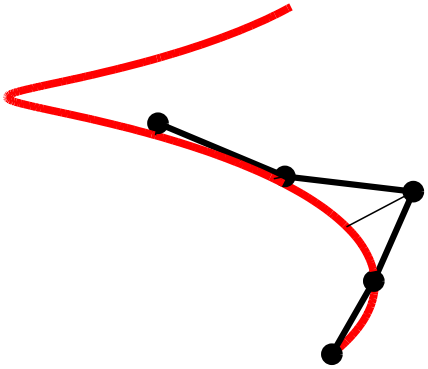
Appropriately placing the first link of the snake robot can significantly affect the quality of the fit. For instance, a reasonable first guess is to place the first link along a tangent line or chord to the curve. However, the plot of the fitting process in Fig. 5 illustrates that the optimal joint angles do not in general place the links tangent to the backbone curve, but instead produce a zigzag pattern centered on the curve. It follows, then, that imposing a tangency constraint on the first module distorts this zigzag pattern, and sub-optimally fits the first several modules to the curve, as seen in Fig. 6(a).

One means of improving the fitting quality of the first modules would be to make a second pass of the fitting algorithm, fixing the last module to the backbone curve and propagating towards the first module, using the results of the initial pass as starting guesses. Unfortunately, this process has the potential to significantly shift the location of the first module; in the next section, we will be fitting the snake robot to *sequences* of backbone curves, and will need tighter control over the placement of the first link.

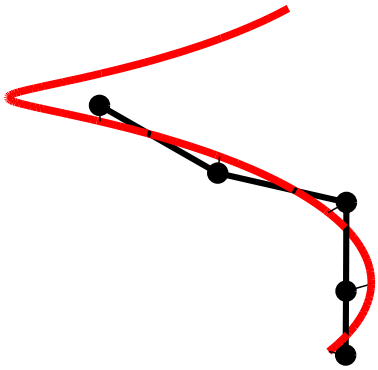
To improve the initial boundary condition while maintaining some control over the first link's placement, we instead add an extra optimization stage between the first and second iterations. In this stage, the placement of the "tangent" line at the base of the curve is allowed to float. The placement of this line is then optimized along with the first set of joint angles to minimize the average distance of the endpoints of all the links from the curve, using the tangent position and joint angles from the initial stage as starting guesses. As shown in Fig. 6(b), this additional stage brings the first link into the pattern of the rest of the modules. Once the new boundary condition is calculated, the annealed fitting operation proceeds along the body of the robot.

4.4 Summary of annealed chain fitting

In summary, the annealed chain fitting process is as follows. First, the head link is placed along the base of the curve with a specified twist angle ϕ. Second, the first a modules are fit to the curve, with the head link fixed. Third, the placement of the head link is relaxed. Finally, the overlapping sets of a modules are fit to the curve.



(a) Simple tangency boundary condition.



(b) Relaxed boundary condition.

Fig. 6 Optimal fit for the first four modules, given (**a**) the first module placed tangent to the curve, and (**b**) a relaxed boundary condition where the first module is allowed to float

5 Keyframe wave extraction

We can use annealed chain fitting's mapping from three-dimensional shape to joint angles as a means for designing gait controllers. By taking a gait as being defined by its time-varying backbone curve, we can sample the backbone configuration at a series of times t_k and apply the chain fitting algorithm to each of these backbone curves. Borrowing a term from the animation community, we refer to the backbone and joint angle data at these time samples as *key frames*. Collecting the fitting results from across all the key frames produces a discrete function

$$\alpha = g(n, t_k) \tag{5}$$

for the joint angles, which can be fed into the snake robot to execute the gait.

This approach becomes significantly more powerful when augmented with a second fitting procedure that extracts parametric functions describing the joint angles. In its raw form, the function g acts as a script for executing the exact set of backbone curves used to create it; the nonlinear and numeric relationship between the backbone curves and the corresponding joint angles means that altering the gait requires repeating the entire fitting process, which is too computationally costly to perform in real time. To avoid this cost, we can instead identify parametric functions of joint number and time which produce the same joint angles as g, so that the real-time controllers can act directly on the parameters.

The first step in this *keyframe wave extraction* process is to separate out the odd and even joint angles into the functions

$$\alpha_{\text{odd}} = g_{\text{odd}}(n_{\text{odd}}, t_k) \tag{6}$$

and

$$\alpha_{\text{even}} = g_{\text{even}}(n_{\text{even}}, t_k). \tag{7}$$

We then find continuous parameterized functions $f_{\text{odd}}(l,t)$ and $f_{\text{even}}(l,t)$ which evaluate to g_{odd} and g_{even} at $l = n_{\text{odd}}$ and $l = n_{\text{even}}$, respectively.

6 Experiments

We have applied annealed chain fitting and keyframe wave extraction to a variety of snake robot gaits, with wave-pattern results matching those found in Lipkin et al. (2007), and, where applicable (Gonzalez-Gomez et al. 2007). These are not novel gaits, but serve as good examples for the workings of our new algorithms, and the agreement of the gaits they produce with the previous empirical studies serves as an important validation of the algorithms.

6.1 Rolling in an arc

Consider the rolling gait illustrated in Fig. 7. In this gait, the snake robot forms an arc in the plane of the ground and rolls laterally. To find the joint angles which form this gait, we chain fit the snake robot to the arc at a full revolution's worth of twist angles, corresponding to the shapes shown in Fig. 7 and their intermediate positions. Extracting the individual joint angles from the fitting data results in the plots of g_{odd} and g_{even} in Fig. 8(a).

The plots in Fig. 8(a) are line plots, as the time discretization can be made small enough to approach continuity, but the spatial resolution is fundamentally limited to the actual modules of the robot. There are clear patterns to g_{odd} and g_{even}, making it easy to find by inspection the parameterized

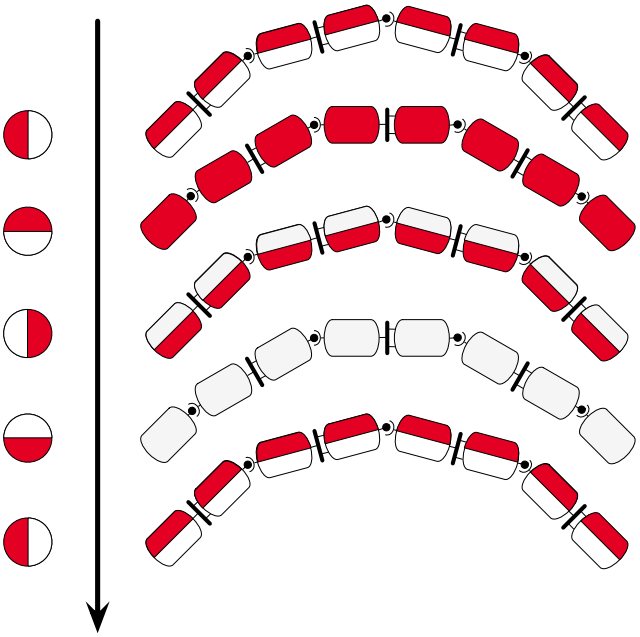


Fig. 7 Rolling in an arc. In our basic rolling gait, the snake robot forms an arc in the plane of the ground and rolls laterally. As the snake robot rolls, the joints vary between being perpendicular and parallel to the ground plane. At *left*, an end-view of the modules illustrates the wheel-like motion

functions which describe them. Each is sinusoidal in time and shows no variation with respect to module number. As there is a quarter-period offset in time between the odd and even functions and they have equal magnitudes, we can assign a pair of continuous functions of the form

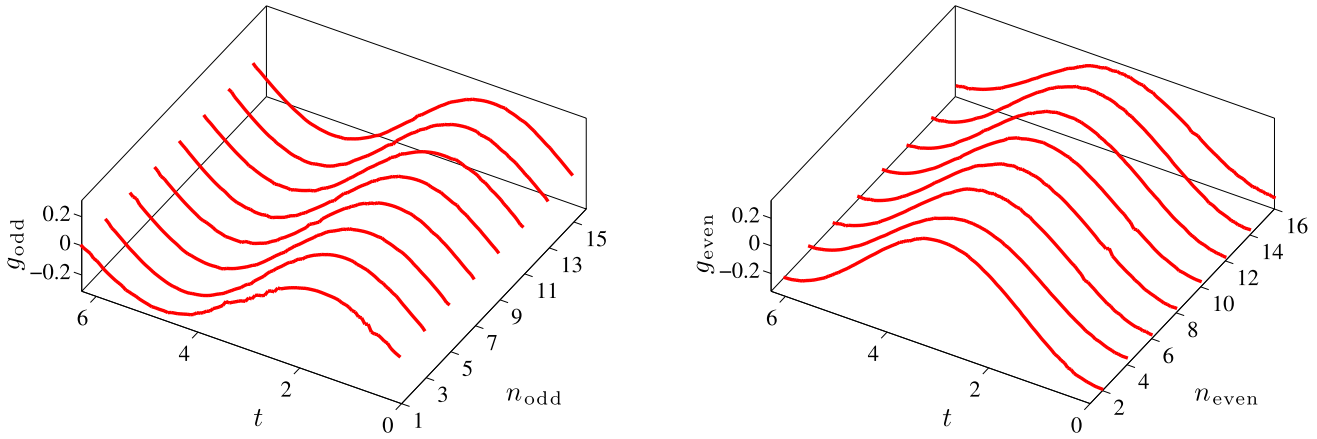
$$f_{\text{odd}}(l,t) = A\sin(\omega t) \tag{8}$$

$$f_{\text{even}}(l,t) = A\sin\left(\omega t - \frac{\pi}{2}\right) \tag{9}$$

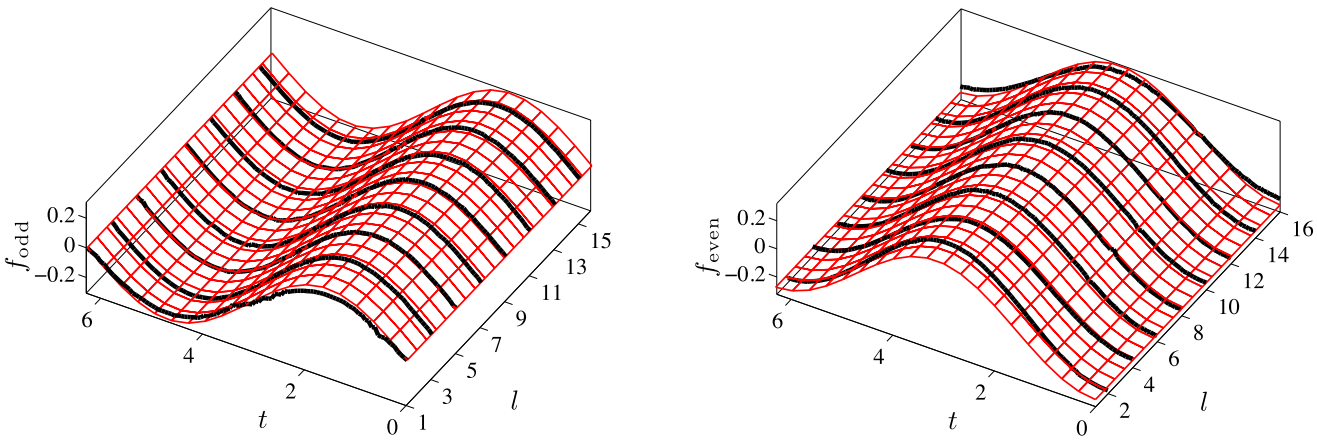
to the gait, which, as in Fig. 8(b), are equal to g_{odd} and g_{even} at $l = n_{\text{odd}}$ and $l = n_{\text{even}}$, respectively. This waveform matches our previous results for generating rolling gaits (Tesch et al. 2009), where we experimentally determined that restrictions of (1) with $A_{\text{odd}} = A_{\text{even}}$, $\beta_{\text{odd}} = \beta_{\text{even}} = 0$, $\Omega_{\text{odd}} = \Omega_{\text{even}} = 0$, $\omega_{\text{odd}} = \omega_{\text{even}}$ and $\delta = -\frac{\pi}{2}$ form the snake robot into the rolling arc shown in Fig. 9.

6.2 Rolling in a helix

One of the more intriguing features of a snake robot is its ability to climb a pole by wrapping around it in a helix, as shown in Fig. 1, and then rolling up the pole. In this gait, as illustrated in Fig. 11, each segment of the body rolls along the pole, effectively acting as a wheel. As with the arc gait, this rolling motion is entirely driven by bending, with no rotary joints aligned with the direction of rolling. We have previously used our empirical approach to find one such gait



(a) Joint angles of the nth modules generated by chain fitting to a rolling arc, as in Fig. 7.



(b) Standing waves which generate the joint angles for the rolling arc.

Fig. 8 Waveform for a rolling arc gait. As in Fig. 7, at each time t, the first link of the snake robot is made tangent to the arc with a twist angle of $\phi = t$, and the remaining links are fit to the curve. With the exception of the first two odd modules (which are distorted by boundary condition effects) there is no variation in the joint angles with respect to module number, and sinusoidal variation with respect to time. These joint angles are described by a pair of standing waves of equal magnitude which are phaseshifted from each other by $\delta = \pi/2$

Fig. 9 Snake robot in its rolling arc configuration

for our snake robots (Lipkin et al. 2007). Others have found an analytical solution for a rolling helix gait for a universal-joint snake robot (Goldman and Hong 2007, 2008), but it relies on both the bidirectional bending capabilities of the universal joints and the symmetries inherent in rolling a robot with a cylindrical cross-section up a cylindrical pole, and is thus not readily extensible to an alternating joint configuration or non-cylindrical geometry. The chain fitting process does not share these requirements, and we can use it to systematically find joint angles for a rolling helix gait without relying on symmetries of the problem.

Generating the rolling helix gait follows much the same process as did generating the rolling arc. At each time t, we place the first link tangent to the helical backbone with twist angle $\phi = t$ and iteratively fit the rest of the links to the curve, as illustrated in Fig. 5. During this fitting, the mechanism takes on a characteristic zig-zag pattern around the backbone curve, which stems from the alternating-joint configuration of the robot. Extracting the odd and even angles gives the plots of g_{odd} and g_{even} in Fig. 10.

Once again, a clear pattern emerges in the joint angles, which can be seen to vary sinusoidally with respect to both module number and time. This pattern is a traveling wave, and the continuous functions f_{odd} and f_{even} in Fig. 10 are

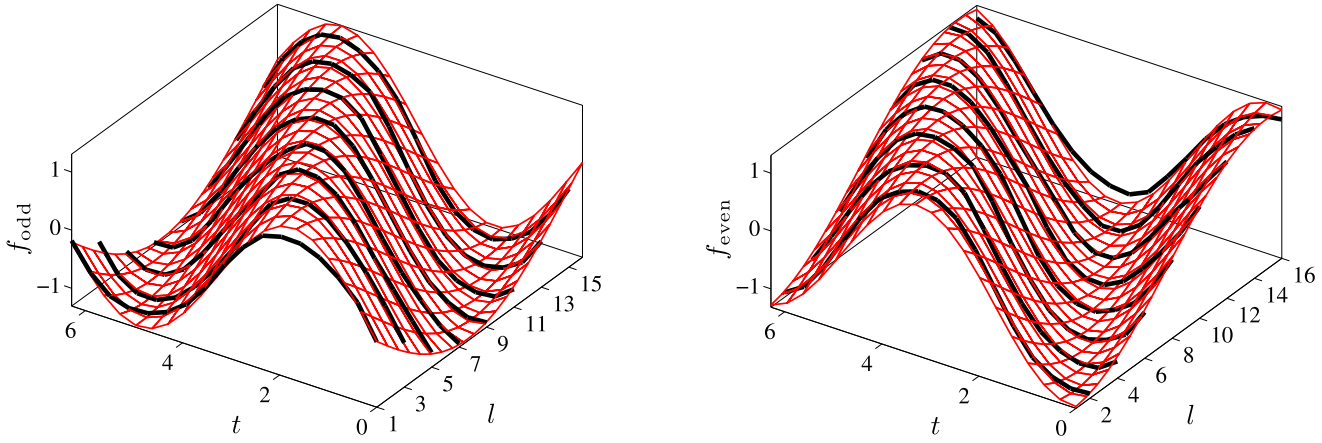


Fig. 10 Waveform for a rolling helix gait. At each time t, the first link of the snake robot is made tangent to the helix with a twist angle of $\phi = t$, and the remaining links are fit to the curve, following the pattern in Fig. 5. For this gait, the joint angle varies sinusoidally with respect to module number as well as time, and thus is described by a pair of traveling waves. As with the rolling arc, the phase shift between the odd and even waves is $\delta = -\pi/2$, which is characteristic of all rolling gaits

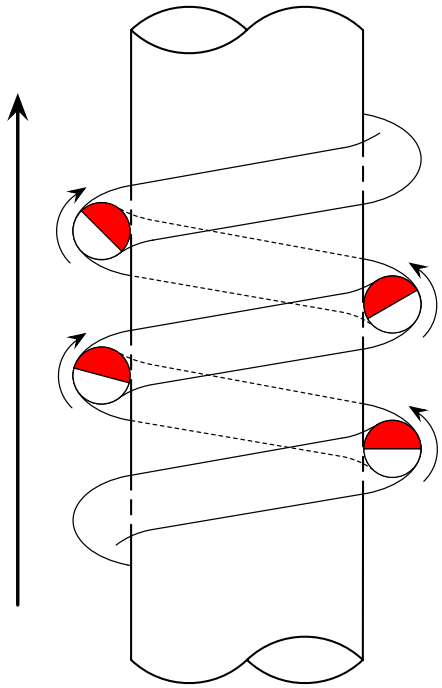


Fig. 11 Rolling up a pole as a helix. In this gait, the snake robot forms a helix around a pole, then twists on its own central axis, so each body segment rolls against the pole with the same net direction of travel

of the form in (1), with parameters $A_{\text{odd}} = A_{\text{even}}$, $\beta_{\text{odd}} = \beta_{\text{even}} = 0$, $\Omega_{\text{odd}} = \Omega_{\text{even}}$, $\omega_{\text{odd}} = \omega_{\text{even}}$ and $\delta = -\frac{\pi}{2}$. These parameters match our previous empirical results (Tesch et al. 2009), validating that chain fitting at various twist angles indeed produces the rolling helix gait used to climb the pole in Fig. 1. Note that the phase shift of $\delta = \pm\frac{\pi}{2}$ is characteristic of rolling gaits, and reflects the fact that the odd and even modules exchange their direction of bending at every quarter-rotation.

6.3 Rolling in an "S"

Some gaits, even simple ones, cannot be described by the single-wave model, and our new approach reveals appropriate models to use to generate these gaits. For instance, consider a variation on the rolling gait, in which the robot assumes the "S" shape in Fig. 13. No choice of parameters in (1) will produce this motion, but by applying chain fitting and wave extraction to this gait, as in Fig. 12, we can find a suitable function. By inspection, the two sets of joint angles are described by parameterized functions of the form

$$f_{\text{odd}}(l,t) = A_{\text{odd}} \sin(\Omega_{\text{odd}} l + \Delta_{\text{odd}}) \sin(\omega_{\text{odd}} t) \tag{10}$$

$$f_{\text{even}}(l,t) = A_{\text{even}} \sin(\Omega_{\text{even}} l + \Delta_{\text{even}}) \sin(\omega_{\text{even}} t + \delta), \tag{11}$$

where the spatial and temporal components are multiplied together, rather than added as in the previous gaits, and Δ and δ represent the spatial and temporal shifts, respectively.

6.4 Sidewinding

Sidewinding is an efficient ground-crossing gait that is especially effective when traversing loose or slippery terrain. In contrast to the first three gaits we examined, it involves changing the locus of points in the backbone curve, rather than the roll distribution. Burdick et al. (1993) characterized the backbone shape during sidewinding as consisting of *ground contact segments* and *arch segments*. The ground contact segments are parallel to each other and in static contact with the environment. The "S"-shaped arch segments connect the ground contact segments and are lifted from the ground. As illustrated in Fig. 14, the snake progressively

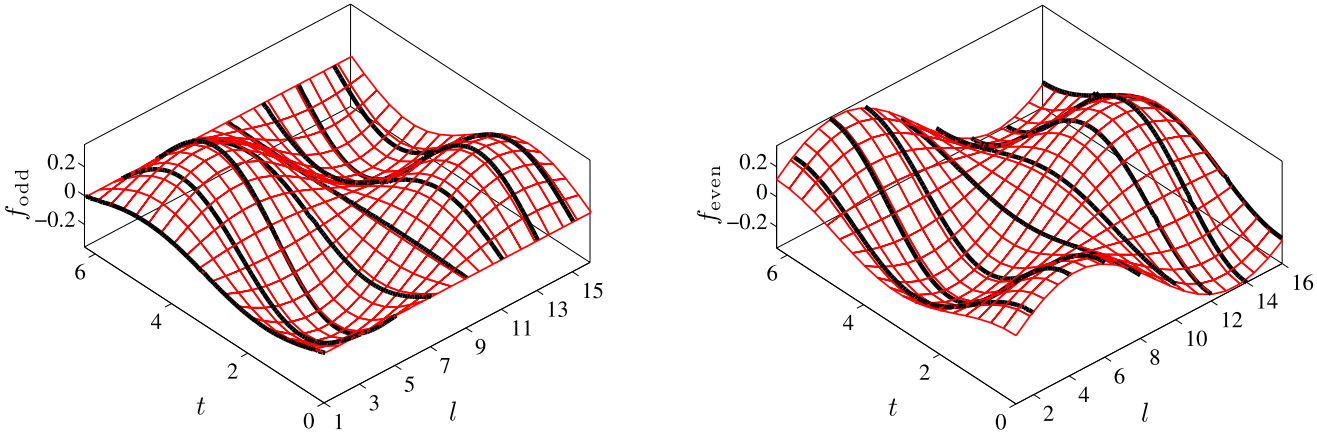


Fig. 12 Wave functions for a rolling "S" gait. The joint angles are described by standing waves, in which the spatial and temporal components are multiplied together, which differs from the additive traveling waves which produce the rolling helix

Fig. 13 Snake robot in its rolling "S" configuration

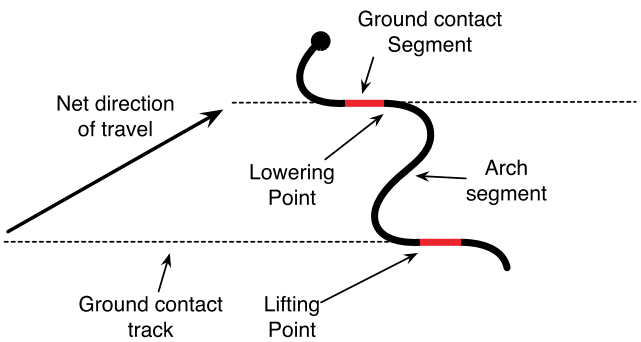


Fig. 14 Sidewinding motion. The ground contact segments are in static contact with the environment, while the arch segments are lifted from the ground and experience no friction. By progressively lifting the ground contact segments into arch segments while laying arch segments down into ground contact segments, the snake or snake robot translates its body mass in the direction shown

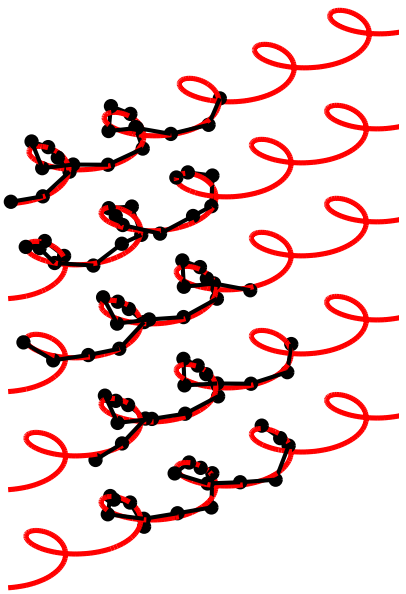


Fig. 15 Fitting to the sidewinding backbone curve. The backbone curves for the key frames of the side winding motion are sections taken progressively from the extended backbone curve. Note that the net motion is a combination of flow along the extended backbone curve and the lateral displacement of the curve itself. This lateral displacement occurs as the snake robot lays down portions of its body to the left of the static contact points, and lifts those to the right

raises one end of each ground contact segment into an arch segment, and lays down the arch segments into the other ends of the ground contacts. In doing so, the snake transfers its body mass between the *ground contact tracks*, producing a net displacement.

To construct the backbone curves for sidewinding, we make some simplifications to the general model. Rather than using the full structure of ground contact and arch segments, we observe that their combined footprint in Fig. 14 resembles a sine wave, and use this wave to specify the x and y components of the backbone curve locus. We then specify the z component of the locus with a cosine wave that raises the portions of the curve corresponding to arch segments with respect to those representing ground contact segments.

The last procedure needed to specify the backbone curve is choosing the coordinate frame of the starting tangent and the roll of the first module with respect to this frame. Here, we choose the tangent's frame by finding the yaw and pitch spherical coordinates which rotate the vector $\begin{bmatrix}1 & 0 & 0\end{bmatrix}^T$

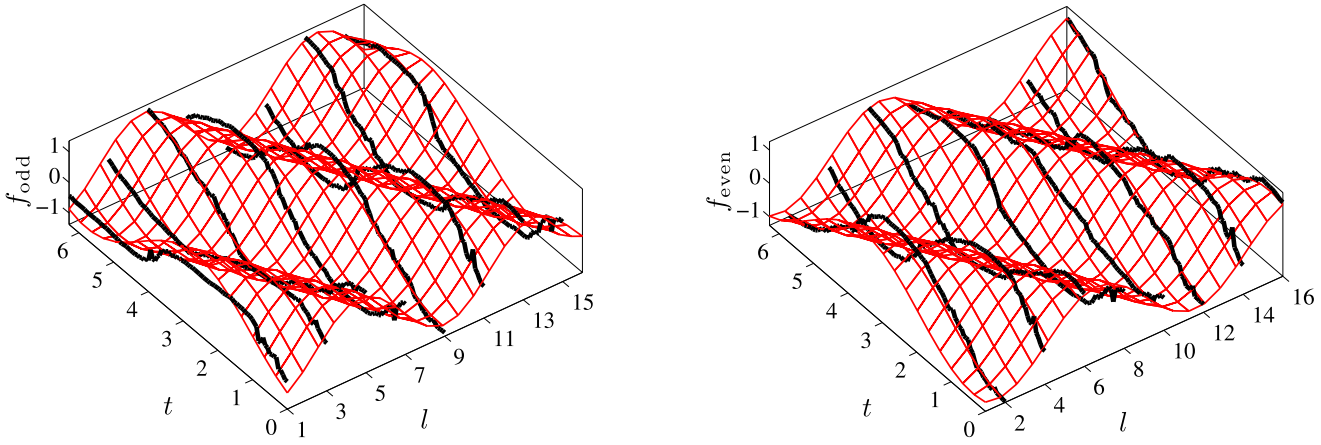


Fig. 16 Wave functions for the sidewinding gait. As with the rolling helix, the sidewinding gait is produced by a pair of traveling waves. The key difference between the two gaits is that the phase shift between the odd and even modules is $\pi/4$ in sidewinding, rather than $\pi/2$ for the rolling helix

to align with the starting tangent, and place the first module at zero roll around the tangent. This may not be the ideal choice, but it is close enough that the wave extraction process corrects for any errors.

With the backbone defined, we generated the sidewinding gait angles by fitting the modules to the extended backbone curve from a sequence of starting points, as shown in Fig. 15. The raw angles found in this fitting process are somewhat noisy, but as shown in Fig. 16 conform to a pattern of two traveling waves offset by a phase shift of $\frac{\pi}{4}$, which once again matches our previous empirical results (Lipkin et al. 2007) and those of Gonzalez-Gomez et al. (2007).

7 Conclusions

In combining a high-level backbone approach with low-level specifications of joint angle waves, we have unified two previously disparate gait design techniques. On their own, backbone methods provide a means of shaping the robot in terms of real-world features, but any adjustment to the backbone requires repeating the fitting process, which is too slow for realtime control. Conversely, joint angle wave approaches provide gait functions which are easily manipulated in real time, but are often not intuitively related to the macroscopic shape of the system. By fitting to a small set of representative backbones and then extracting the corresponding joint angle waves, we gain the benefits of both approaches.

In this paper, we have explored the use of chain fitting and wave extraction to generate a variety of rolling and sidewinding gaits. The waveforms we found match our previous empirical results, confirming the underlying principle of our approach. In our future work, we will be particularly interested in applying these techniques to novel gaits which cannot be described by our single-wave models, and are thus beyond the scope of our previous design efforts. This work will include automatic identification of the waveforms described by the chain-fit angles, removing the bottleneck of human intervention present in the current work.

Viewing gaits as surfaces on the joint-number/time space raises interesting questions about the fundamental nature of gaits. Many previous gait design efforts can be regarded as building these surfaces from simple sinusoidal basis functions; are there other useful gaits to be constructed from different or higher-order basis functions? Also, we may be able to define equivalence classes for gaits based on their surface features that apply across choices of basis functions. For instance, the general sidewinding definition in Burdick et al. (1993) admits gaits that cannot be described in joint space by first order sinusoidal functions; are there shared features on those gaits' wave surfaces that link them together? Further, if those features exist, how do transitions between equivalence classes in the joint space correspond to objective changes in the motion of the three-dimensional backbone?

Acknowledgements We would like to thank Matthew Tesch and Marissa Jacovich for their input on this work.

References

Andersson, S. B. (2008). Discretization of a continuous curve. *IEEE Transactions on Robotics*, *24*(2), 456–461.

Burdick, J., Radford, J., & Chirikjian, G. (1993). A "sidewinding" locomotion gait for hyper-redundant robots. In *IEEE international conference on robotics and automation (ICRA)* (Vol. 3, pp. 101–106).

Chirikjian, G. (1992). *Theory and applications of hyper-redundant robotic manipulators*. Ph.D. thesis, California Institute of Technology.

Chirikjian, G., & Burdick, J. (1991). Kinematics of hyper-redundant locomotion with applications to grasping. In *IEEE international conference on robotics and automation (ICRA)*.

Goldman, G., & Hong, D. W. (2007). Determination of joint angles for fitting a serpentine robot to a helical backbone curve. In *International conference on ubiquitous robots and ambient intelligence*.

Goldman, G., & Hong, D. W. (2008). Considerations for finding the optimal design parameters for a novel pole climbing robot. In *ASME mechanisms and robotics conference*.

Gonzalez-Gomez, J., Zhang, H., Boemo, E., & Zhang, J. (2006). Locomotion capabilities of a modular robot with eight pitch-yaw-connecting modules. In *9th international conference on climbing and walking robots*.

Gonzalez-Gomez, J., Zhang, H., & Boemo, E. (2007). Locomotion principles of 1D topology pitch and pitch-yaw-connecting modular robots. In *Bioinspiration and robotics: walking and climbing robots*. Advanced Robotics Systems International and I-Tech Education and Publishing.

Hatton, R. L., & Choset, H. (2009). Generating gaits for snake robots by annealed chain fitting and keyframe wave extraction. In *IEEE/RSJ international conference on intelligent robots and systems (IROS)*.

Hirose, S. (1993). *Biologically inspired robots (snake-like locomotor and manipulator)*. Oxford: Oxford University Press.

Ijspeert, A. J. (2008). Central pattern generators for locomotion control in animals and robotics. *Neural Networks*, *21*, 642–653.

Lipkin, K., Brown, I., Peck, A., Choset, H., Rembisz, J., Gianfortoni, P., & Naaktgeboren, A. (2007). Differentiable and piecewise differentiable gaits for snake robots. In *IEEE/RSJ international conference on intelligent robots and systems (IROS)*, San Diego, CA, USA (pp. 1864–1869).

Mori, M., & Hirose, S. (2002). Three-dimensional serpentine motion and lateral rolling by active cord mechanism ACM-R3. In *IEEE/RSJ international conference on intelligent robots and systems (IROS)*.

Nelder, J. A., & Mead, R. (1965). A simplex method for function minimization. *Computer Journal*, *7*, 308–313.

Nilsson, M. (1998). Why snake robots need torsion-free joints and how to design them. In *IEEE international conference on robotics and automation (ICRA)*.

Sfakiotakis, M., & Tsakiris, D. (2007). Biomimetic centering for undulatory robots. *International Journal of Robotics Research*, *26*(11–12), 1267–1282.

Tesch, M., Lipkin, K., Brown, I., Hatton, R., Peck, A., Rembisz, J., & Choset, H. (2009). Parameterized and scripted gaits for modular snake robots. *Advanced Robotics*, *23*(9), 1131–1158.

Wright, C., Johnson, A., Peck, A., McCord, Z., Naaktgeboren, A., Gianfortoni, P., Gonzalez-Rivero, M., Hatton, R., & Choset, H. (2007). Design of a modular snake robot. In *IEEE/RSJ international conference on intelligent robots and systems (IROS)*, San Diego, CA, USA (pp. 2609–2614).

Yamada, H., & Hirose, S. (2006). Study on the 3D shape of active cord mechanism. In *IEEE international conference on robotics and automation (ICRA)*.

Yim, M., Homans, S., & Roufas, K. (2001). Climbing with snake-like robots. In *IFAC workshop on mobile robot technology*.

Yu, S., Ma, S., Li, B., & Wang, Y. (2008). Analysis of helical gait of a snake-like robot. In *IEEE/ASME international conference on advanced intelligent mechatronics*.

Zhang, Y., Yim, M., Eldershaw, C., Duff, D., & Roufas, K. (2003a). Phase automata: a programming model of locomotion gaits for scalable chain-type modular robots. In *IEEE/RSJ international conference on intelligent robots and systems (IROS)*.

Zhang, Y., Yim, M., Eldershaw, C., Duff, D., & Roufas, K. (2003b). Scalable and reconfigurable configurations and locomotion gaits for chain-type modular reconfigurable robots. In *IEEE symposium on computational intelligence in robotics and automation (CIRA)*.

Ross L. Hatton received an SB in Mechanical Engineering from the Massachusetts Institute of Technology in 2005 and a MS in the same from Carnegie Mellon University in 2007. He is currently a PhD student in Robotics and Mechanical Engineering at Carnegie Mellon University. His research focuses on understanding the fundamental mechanics of locomotion and on finding abstractions that facilitate human control of unconventional locomotors.

Howie Choset is an Associate Professor of Robotics at Carnegie Mellon University, where he conducts research in path planning, motion planning, estimation, mechanism design and hybrid controls. Much of this work has two foci: snake robots for search and rescue, manufacturing and medical robotics, and coverage for demining and autobody painting. He directs the Undergraduate Robotics Minor at Carnegie Mellon University and teaches an overview course on Robotics which uses series of custom developed Lego Labs to complement the course work. His students have won Best Paper awards at the RIA in 1999 and ICRA in 2003, he has been nominated for Best Papers at ICRA in 1997 and IROS in 2003 and 2007, and won Best Paper at IEEE Bio Rob in 2006. In 2002 the MIT Technology Review elected him as one of its top 100 innovators in the world under 35. In 2005, MIT Press published a textbook, lead authored by him, entitled Principles of Robot Motion.