

# Simplified Motion Modeling for Snake Robots

Florian Enner, David Rollinson and Howie Choset

**Abstract**— We present a general method of estimating a snake robot’s motion over flat ground using only knowledge of the robot’s shape changes over time. Estimating world motion of snake robots is often difficult because of the complex way a robot’s cyclic shape changes (gaits) interact with the surrounding environment. By using the *virtual chassis* to separate the robot’s internal shape changes from its external motions through the world, we are able to construct a motion model based on the differential motion of the robot’s modules between time steps. In this way, we effectively treat the snake robot like a wheeled robot where the bottom-most modules propel the robot in much the way the bottom of the wheels would propel the chassis of a car. Experimental results using a 16-DOF snake robot are presented to demonstrate the effectiveness of this method for a variety of gaits that have been designed to traverse flat ground.

## I. INTRODUCTION

Snake robots are a class of hyper-redundant mechanisms [1] consisting of kinematically constrained links chained together in series. Their many degrees of freedom given them the potential to navigate a wide range of environments. Our group has developed modular snake robots that rely solely on their internal shape changes to locomote through their environment [2]. To simplify control of the robot’s many degrees of freedom, we have developed cyclic motions, known as gaits, that undulate the robot’s joints according to parameterized sine waves [3]. We have developed and implemented gaits that can traverse a variety of terrains, including flat ground, slopes, and pipes. In particular, for flat ground there are a number of gaits that are used to move the snake forward, sideways, or turn in place as shown in Fig. 1.

Despite the successful implementation of these gaits, representing and estimating external motion of these gaits through the world has proven to be difficult. The internal shape changes that a snake robot uses to locomote involve the entire body and introduce two significant problems. First, no coordinate frame that is static with respect to a single point on the robot intuitively represents the pose of the entire robot at all times. Second, the segments of the robot that are interacting with ground (and thus inducing motion in the world) are constantly changing. To address the first problem, we have developed the virtual chassis [4] that relies on the overall shape of the robot to define a unique body frame at every point in time. In particular, this body



Fig. 1: Examples of a snake robot executing the gaits discussed in this paper. Moving across from top left to bottom right: rolling, sidewinding, slithering, and turn-in-place.

frame has the property that it is constantly aligned with the principle components of the robot’s overall shape. It provides a formally defined body frame which closely matches the intuitive way in which an operator thinks about the position and orientation of the robot.

The second problem is addressed in this work by exploiting the observation that gaits consistently interact with the ground parallel to the flattest side of the robot’s shape (or more formally, the plane normal to the third principal axis of inertia). By defining the “bottom” side of the robot in this way we are able to approximate the effects that the relevant modules have on the overall motion of the robot. By averaging the translational motions induced by the bottom modules, together with the rotational motions of these modules around the geometric center of the robot, we are able to estimate the net translation and rotation of the robot through arbitrary shape changes. This simplified model estimates the motion of a snake robot in much the same way that a simple no-slip assumption is frequently (and effectively) used to estimate the motion of a wheeled robot. It is our hope that this technique will begin to allow many planning and estimation tools from the wheeled robot community to be used on hyper-redundant and articulated locomoting robots.

Animations of this motion model are presented in an accompanying video.

## II. PRIOR WORK

There is a significant amount prior work in the study of the motion of biological snakes [5], [6] and snake robots [1], [7].

Florian Enner is a graduate student in the Salzburg University of Applied Sciences, Austria [fenner.its-m2010@fh-salzburg.ac.at](mailto:fenner.its-m2010@fh-salzburg.ac.at)

David Rollinson is a graduate student in the Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, 15213 [drollins@cs.cmu.edu](mailto:drollins@cs.cmu.edu)

Howie Choset is an associate professor in the Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, 15213 [choset@cs.cmu.edu](mailto:choset@cs.cmu.edu)

A survey of a wide variety of snake robots and snake robot locomotion is presented in [8]. More recent research on both biological snakes [9], [10] and robotic snakes [11], [12] has focused on a snake’s interaction with its environment during locomotion.

Our method differs from [11] in that we model the robot as being purely kinematic. While [1] also presents a kinematic model for hyper-redundant systems, the model relies on having a continuous backbone curve in order to understand and control the desired macroscopic shape of the robot. The assumptions made by our model are in much the same spirit as previous work that has been done for sidewinding in both biological snakes [13] and robotic snakes [14], [15]. Our model differs in that it uses the shape of the robot directly and is agnostic to the underlying functions used for motion control.

Rather than fully account for the true forces acting on the robot or the true shape of terrain with which we interact, our goal is to present a simplified model that is still a reasonable approximation of the robot’s motion. Furthermore we would like to pursue a model that is computationally inexpensive and can be easily implemented for systems that lack the necessary tactile, force, torque, or pose sensors needed to inform a full dynamic motion model. These traits make this method potentially desirable to be used as the underlying motion model of various state estimation techniques such as particle filters or kalman filters that are widely used in the wheeled robot community [16], [17].

### III. MOTION MODEL

In this section, we describe our mathematical model for the robot’s external motion due to its internal shape changes. We use the virtual chassis [4] at each timestep  $t$  as the body frame from which to observe the motion of the robot’s modules. This body frame has the property that it is constantly aligned with the snake robot’s overall shape, and effectively identifies the flattest dimension of the snake’s shape (the dimension corresponding to the third principal moment of inertia). It should be assumed that any use of the term ‘body frame’ in this paper means the body frame defined by the robot’s virtual chassis.

Our model assumes that the axis corresponding to the flattest principal component is aligned with the ground surface normal, and this axis is assigned to the  $z$  axis of the virtual chassis. This makes our model similar to that of most flat ground wheeled vehicle models, where we assume that translations occur only in the  $x$ - $y$  plane and that rotations occur about only the  $z$  axis. In a way this motion model can be thought of as a numeric derivation of the common no-slip model for wheeled robots. It would be analogous to observing the differential motion of the bottom of a vehicle’s wheels in the body frame at each time step and using that motion to predict the motion of the vehicle in the world.

#### A. Gaits and Robot Kinematics

To simplify control of the 16 degrees of freedom used to locomote our robot, we rely on gaits that are pre-defined

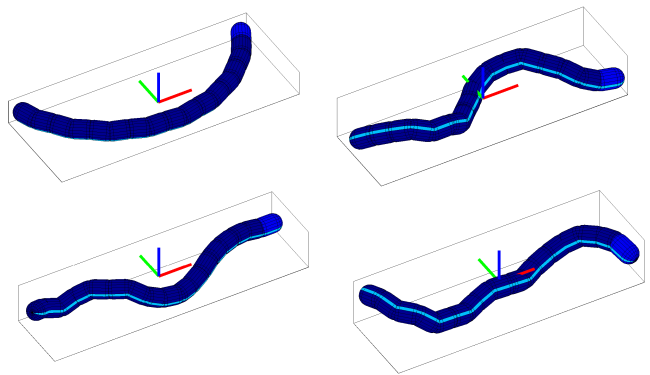


Fig. 2: Examples of the virtual chassis body frame for the four gaits in this paper. Moving across from top left to bottom right: rolling, sidewinding, slithering, and turn in place.

cyclic undulations that are passed through the length of the snake. All of the gaits presented in this paper use parameterized sine waves that are similar to Hirose’s serpenoid curve [7], and its 3D extensions [18]. The serpenoid curve describes the curvature of a backbone as a function of time, position on the backbone, and other gait-specific parameters.

To provide 3D mobility and manipulation, the robot’s joints are alternately oriented in the lateral and dorsal planes of the robot. Because of this design, our framework for gaits consists of separate parameterized sine waves that propagate through the lateral (even-numbered) and dorsal (odd-numbered) joints. We refer to this framework as the *compound serpenoid curve*,

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

$$\begin{aligned} \xi_{\text{odd}} &= \psi_{\text{odd}}n + \nu_{\text{odd}}t \\ \xi_{\text{even}} &= \psi_{\text{even}}n + \nu_{\text{even}}t. \end{aligned} \quad (2)$$

In (1)  $\beta$ ,  $A$  and  $\eta$  are respectively the angular offset, amplitude, and phase shift between the lateral and dorsal joint waves. In (2) the parameter  $\psi$  describes the spatial frequency of the macroscopic shape of the robot with respect to module number,  $n$ . The temporal component  $\nu$  determines the frequency of the actuator cycles with respect to time,  $t$ .

Using equations (1) and (2), the parameters describing a given gait, and the robot’s mechanical design parameters (configuration of joint axes and the distance between joints), we can generate the 3-dimensional shape of the snake robot from the forward kinematics of the system. At each time step, the virtual chassis body frame for the robot’s shape is calculated as described in [4]. The virtual chassis for the gaits presented in this paper is shown in Figure 2.

#### B. Module Motion

The means in which a module’s motion can cause external movement are twofold: a translation of a module’s center in the body frame and a rotation about a module’s center that causes a translation at the surface of the module touching the ground (Fig. 3). The effect of rotational module motion is mostly seen in gaits like rolling (top-left in Figs. 1 and

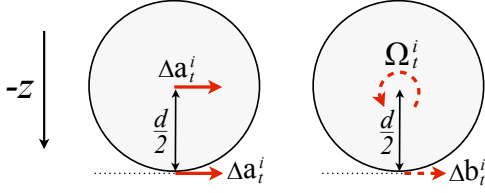


Fig. 3: A simplified 2-D diagram showing the effects of translation and rotation of a module on the point that is assumed to be in contact with the ground.

2) where module positions stay fixed in the body frame, but rotate in place. More complicated motions such as sidewinding and slithering (top-right and bottom-left in Figs. 1 and 2) are driven primarily by the translational module motion where the positions of the modules themselves move with respect to the body frame.

To calculate a module's translational motion at each point in time, we subtract the module's position in the body frame at the previous timestep from its current position in the body frame. The position of the  $i$ th module of the robot at time  $t$  is

$$\mathbf{a}_t^i = \begin{bmatrix} x^i \\ y^i \\ z^i \end{bmatrix}_t. \quad (3)$$

Thus the differential motion of the  $i$ th module due to its translation in the body frame at time  $t$  is

$$\Delta \mathbf{a}_t^i = \mathbf{a}_t^i - \mathbf{a}_{t-1}^i. \quad (4)$$

To calculate the effects of a module's rotational motion, each module can be pictured as a wheel. Modules are modeled as spheres, where we are interested in the point on the sphere that points down (in the  $-z$  direction of the body frame) at each point in time. We can define this direction in the frame of each module by

$$\mathbf{r}_t^i = (\mathbf{R}_t^i)^{-1} \begin{bmatrix} 0 \\ 0 \\ -d/2 \end{bmatrix}. \quad (5)$$

In (5)  $d$  is the diameter of the module, and  $\mathbf{R}_t^i$  is the rotation matrix that describes the orientation of the  $i$ th module in the body frame at time  $t$ . We can describe the differential rotation of the  $i$ th module (in that module's frame) at time  $t$  by

$$\Omega_t^i = (\mathbf{R}_{t-1}^i)^{-1} \mathbf{R}_t^i. \quad (6)$$

We can apply this differential rotation both forwards and backwards to the point  $\mathbf{r}_t^i$ , average the displacements due to the rotations, and rotate the result back into the body frame

$$\Delta \mathbf{b}_t^i = \mathbf{R}_t^i \frac{\Omega_t^i \mathbf{r}_t^i - (\Omega_t^i)^{-1} \mathbf{r}_t^i}{2}. \quad (7)$$

Finally, we can sum the motion component due to a module's translation,  $\Delta \mathbf{a}_t^i$ , and a module's rotation,  $\Delta \mathbf{b}_t^i$ ,

$$\Delta \mathbf{p}_t^i = \Delta \mathbf{a}_t^i + \Delta \mathbf{b}_t^i \quad (8)$$

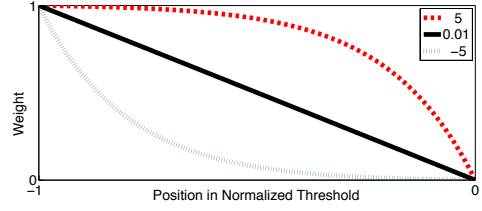


Fig. 4: Plot of our weighting function over different curvature parameters.

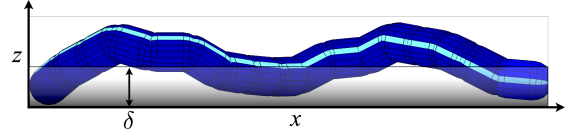


Fig. 5: Diagram qualitatively illustrating a possible threshold range for sidewinding.

in order to capture the full motion that a module would impart on the body frame of the robot, if it were contacting the ground in the  $-z$  direction in the body frame.

### C. Ground Contact Model

The next step of the motion model is to determine which modules are assumed to be in contact with the ground. Since for most cases the flat axis of the virtual chassis is relatively normal to the ground, we assume that modules within a threshold of the lowest position in the  $z$ -axis of the body frame (axis closest to pointing upwards in the real world) are having ground contact. For reference, each module of the robot is 5 cm in diameter. The average module range of motion in the  $z$ -axis of the virtual chassis body frame is shown in table I.

To avoid discontinuities in the model as modules pass in an out of the ground contact threshold, we introduce a normalized exponential smoothing function that weights the lowest modules more strongly and gradually decreases the weight to zero through the range of the threshold,

$$\gamma_t^i = \begin{cases} 1 - \frac{z_t^i - z_{\min}}{\tau}, & \text{if } z_t^i - z_{\min} < \tau \\ 0, & \text{otherwise} \end{cases} \quad (9)$$

$$w_t^i = \frac{1 - e^{-\delta \gamma_t^i}}{1 - e^{-\delta}} \quad (10)$$

$$\hat{w}_t^i = \frac{w_t^i}{\sum_{i=1}^n w_t^i}. \quad (11)$$

In (9),  $\tau$  is the threshold parameter that controls the range of positions for which modules are considered to be in contact with the ground,  $z_t^i$  corresponds to the  $z$  position

TABLE I: Gait Z-Axis Range of Motion

Gait	Range (cm)
Rolling	0.4
Sidewind	4.5
Slither	4.7
Turn-in-place	4.8

of the  $i$ th module and  $z_{\min}$  is the minimum  $z$  position of all modules at time  $t$ . In (10),  $\delta$  is a curvature parameter that controls the shape of the weighting function through the threshold. Figure 4 shows the effect of  $\delta$  on the shape of the weighting function.

This simplification introduces errors which depend partly on the angle between  $z$ -axis of the virtual chassis and the vector normal to the actual ground as well the number of modules that might be contacting the ground simultaneously, due to the irregular shape of the modules. Since these traits differ for every gait, we explored optimizing  $\tau$  and  $\delta$  on a gait-by-gait basis. In section IV we present results for motions that are optimized for translating gaits, as well as results for a general set of parameters that have been optimized across a range of motions.

#### D. Body Frame Translation

Using the estimated differential motion for each module, and the weights that represent assumed ground contact, we are able to estimate the overall motion of the robot. The predicted translational motion of the robot is calculated by the weighted average of module motions in the body frame

$$\Delta \mathbf{m}_t = - \sum_{i=1}^n \hat{w}_t^i \Delta \mathbf{p}_t^i. \quad (12)$$

While the predicted translation will contain components in the  $x$ ,  $y$  and  $z$  directions, the component of motion in the  $z$  direction is small compared to  $x$ - $y$  and will be neglected due to the constraints of the planar motion model. It is important to note the negative sign on the sum in (12), since our model assumes the motion of the robot is due to the reaction of the displacements of modules in the body frame.

#### E. Body Frame Rotation

The rotational motion of the robot can also be calculated in a similar manner as translation. Because the motion model is constrained to representing translations and rotations in the  $x$ - $y$  plane, we compute a weighted average of the rotations that the modules induce about the geometric center of the robot (the origin of the body frame). For this, we first compute moment vectors from the cross product of the position of each module in the body frame with the  $z$ -axis and normalize the resulting vector to be unit length

$$\mathbf{d}_t^i = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \times \mathbf{a}_t^i \quad (13)$$

$$\bar{\mathbf{d}}_t^i = \frac{\mathbf{d}_t^i}{\|\mathbf{d}_t^i\|} \quad (14)$$

We then calculate the weighted sums of the dot product of each module's displacement with the module's corresponding moment vector divided by the distance of that module to the geometric center of the robot

$$\Delta \theta_t = - \sum_{i=1}^n \hat{w}_t^i \frac{(\Delta \mathbf{p}_t^i \cdot \bar{\mathbf{d}}_t^i)}{\|\mathbf{a}_t^i\|}. \quad (15)$$

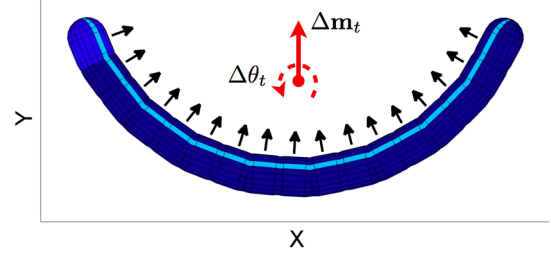


Fig. 6: Diagram showing the estimated translation and rotation of the robot by averaging the weighted motion of all modules.

As with the robot's translation, the sign of the summed rotation is flipped because fact that the robot's rotation is caused by the reaction of the modules' displacements about the geometric center of the robot as view in the body frame.

#### F. Full Body Frame Motion

The total translational and rotational motion at time  $t$  can be combined into a homogeneous transform that can be applied to the body frame of the robot at each point in time

$$\mathbf{T}_t = \begin{bmatrix} \cos(\Delta \theta_t) & -\sin(\Delta \theta_t) & 0 & \Delta m_t^x \\ \sin(\Delta \theta_t) & \cos(\Delta \theta_t) & 0 & \Delta m_t^y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (16)$$

In (16),  $\Delta m_t^x$  and  $\Delta m_t^y$  are respectively the  $x$  and  $y$  components of the robot's translational motion,  $\Delta \mathbf{m}_t$ , from (12).

## IV. EXPERIMENT

To verify our model, we ran experiments using the four gaits mentioned previously: rolling, sidewind, slither, and turn-in-place. To optimize  $\tau$  and  $\delta$  we performed test runs of the different gaits on our lab floors. Four trials of turn-in-place and six trials each of rolling, sidewind, and slither were conducted. The runs were started at different phases within their respective gait cycles, moved over various distances, and run in forward and reverse in order to give a wide sampling of external motion within the real world. For each trial we recorded the feedback joint angles from the robot and manually measured the change in position and orientation of the robot in the world. After tuning the model parameters, 24 new trials were run to test the model's accuracy. The model was also tested on previously logged data of the robot moving in a motion capture lab.

#### A. Model Optimization

To represent the position and orientation of the robot we chose to use polar coordinates instead of cartesian coordinates, since it more intuitively and stably represents motions where the movements are primarily along a single axis

$$\epsilon_r = \frac{r_{\text{meas}} - r_{\text{pred}}}{r_{\text{meas}}} \quad (17)$$

$$\epsilon_\phi = \frac{\phi_{\text{meas}} - \phi_{\text{pred}}}{\pi} \quad (18)$$

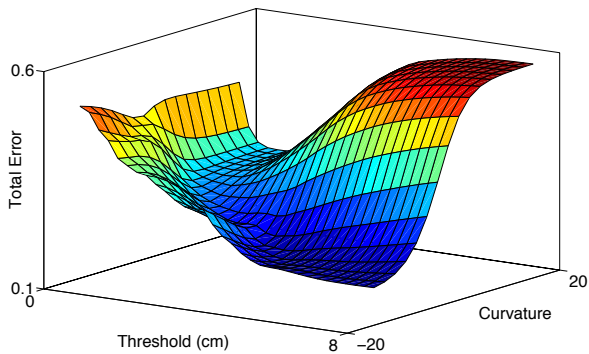


Fig. 7: Averaged total errors for all of the training trials, plotted for a range of thresholds and curvatures.

$$\epsilon_{\theta} = \frac{\theta_{\text{meas}} - \theta_{\text{pred}}}{\pi} \quad (19)$$

In our parameterization  $r$  is the distance the robot traveled from the origin,  $\phi$  is the angle of the robot’s final position, and  $\theta$  is the angle of the robot’s orientation. These errors were normalized to be percent error for the distances (17) and by  $\pi$  radians for the angles (18) (19).

The total error function is a weighted summed squares

$$\epsilon_{\text{total}} = \sqrt{\lambda_r(\epsilon_r)^2 + \lambda_{\phi}(\epsilon_{\phi})^2 + \lambda_{\theta}(\epsilon_{\theta})^2} \quad (20)$$

where  $\lambda$  varies the relative weighting of the different errors. For the gaits that are used for translating the snake (sidewind, rolling and slither) we used the weights:  $\lambda_r = 0.2, \lambda_{\phi} = 1, \lambda_{\theta} = 1$ . For the turn-in-place gait where  $\phi$  is not meaningful and  $\theta$  is more important, we used the weights:  $\lambda_r = 0.05, \lambda_{\phi} = 0, \lambda_{\theta} = 1$ . Figure 7 shows the total error for a range of thresholds and curvatures, averaged for the 22 training trials of the various gaits. It illustrates the fairly wide region of usable parameters for the model.

The optimal parameters for predicting motion varied according to the type of gait, and are shown in Table II. Notably, the predicted motion for the turn-in-place gait became more accurate with a much higher threshold and steeper curvature. Since the other gaits are less sensitive to varying the model parameters, the overall best parameters based on the training data were very similar to the ones for turn-in-place.

### B. Results

To test the accuracy of the model, 6 new trials of each gait were conducted for a total of 24 trials. As with the previous trials, gaits were started at various points in their respective cycles and run in different directions in order sample a large range of motion. Tables III and IV present the means and standard deviations for both the specific and general model parameters for each gait. Note that the results for the turn-in-place gait are presented differently from the other gaits. Due to the fact that the goal of the turn-in-place gait is to rotate the snake, the yaw motions are presented in terms of percentage error, rather than degrees. Similarly, the translation of the robot is so small that they are presented in

TABLE II: Table of Parameters

Gait Type	$\tau$ (cm)	$\delta$
Rolling / Sidewind / Slither	0.75	1.0
Turn-in-place	7.5	-15.0
Overall	7.5	-15.0

percentage of robot length instead of percentage of distance traveled.

The motion model was also tested on previously logged motion capture data of a similar robot and environment. The robot in the motion capture data had a different head, tail, and tether from the configuration used in the other trials. It was also moving on the concrete floor of the motion capture studio, instead of the tile floor in the lab. In Fig. 8 we show the estimated motion from the logged joint angles, compared to the motion capture data of the robot’s actual motion.

## V. CONCLUSIONS

We have presented a general method for estimating the motion of a high DOF snake robot on flat ground. Our method makes use of the robot’s virtual chassis body frame to separate the robot’s internal shape changes from external motion in the world as well as to identify the parts of the robot that are in contact with the ground. The utility of this model lies in its simplicity. As long as one knows the kinematic configuration at each timestep, this method can be applied to a robot that lacks force feedback, tactile sensing, or even orientation sensing (beyond what is needed to properly initialize the virtual chassis).

Even though the assumptions of our model neglect dynamic effects like inertia, frictional forces, and intermittent ground contact, it provides a relatively accurate estimate of motion at much less computational expense compared to full 3D dynamic simulations or more complex dynamic models.

TABLE III: Motion Model Errors - Rolling / Sidewind / Slither Parameters

	$r$ (%)		$\phi$ (deg)		$\theta$ (deg)	
	Mean	Dev.	Mean	Dev.	Mean	Dev.
Rolling	9 %	2 %	10°	7°	15°	6°
Sidewind	5 %	2 %	11°	6°	19°	11°
Slither	6 %	3 %	13°	9°	21°	3°

	$r$ (%)		$\phi$ (deg)		$\theta$ (%)	
	Mean	Dev.	–	–	Mean	Dev.
Turn-in-place	12 %	4 %	–	–	70 %	2 %

TABLE IV: Motion Model Errors - General Parameters

	$r$ (%)		$\phi$ (deg)		$\theta$ (deg)	
	Mean	Dev.	Mean	Dev.	Mean	Dev.
Rolling	9 %	2 %	9°	3°	12°	11°
Sidewind	18 %	2 %	11°	7°	14°	6°
Slither	6 %	3 %	9°	6°	10°	7°

	$r$ (%)		$\phi$ (deg)		$\theta$ (%)	
	Mean	Dev.	–	–	Mean	Dev.
Turn-in-place	9 %	7 %	–	–	34 %	11 %

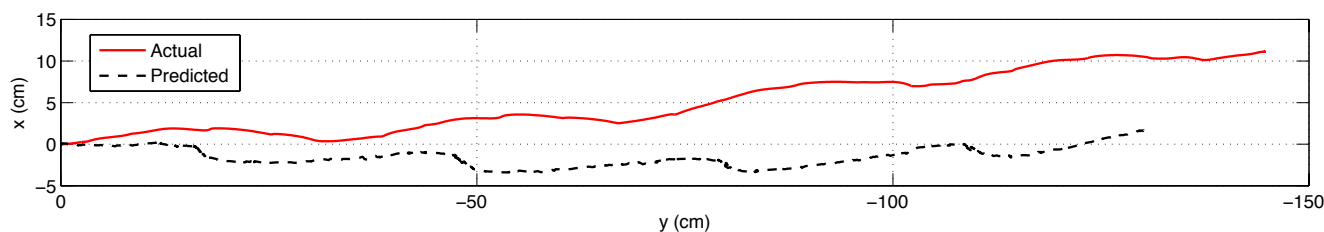


Fig. 8: Comparison between motion capture data and estimated motion for sidewinding at a speed of approximately 10 cm/s.

Although the parameters of the model can be optimized for a given gait or motion, a general set of model parameters have been demonstrated to be effective for a wide range of gaits that are frequently used for traversing flat ground. Further trials will be performed with additional motions and on a variety of terrains (carpet, grass, asphalt, etc.) to see how effectively this model extends to other operating conditions.

## VI. FUTURE WORK

### A. Gait Design and Motion Planning

One of the ongoing challenges with controlling snake robots is designing efficient and useful motions that take into account interaction with the world. Most of the gaits developed by our lab have been the result of hand-tuned functions followed by either experimental refinement or some form of offline optimization [15], [19]. One line of future work could involve using this model as part of more interactive tools for gait development, or serve as the model for online optimization of gait transitions or other non-cyclic motions.

### B. State Estimation / SLAM

Perhaps the most relevant extension of this work would be incorporate it into state estimation or use as a motion model for SLAM algorithms. In many ways the approximation of this model mirrors the no-slip assumption often made in the motion model of many wheeled robotic systems. Given the success that wheeled robots have had in using a simplified and often poor assumption of ground interaction, we feel that motion model framework presented in this work may fulfill such a role for snake robots and articulated locomoting systems.

### C. Extending to Other Terrains

The most significant limitations of the current model is the assumption of flat ground. However, this assumption is used primarily as a stand-in for a lack of true ground contact sensing. If the robot's modules were equipped with tactile sensors or some other method of determining ground contact was available this model could be adapted to handle full 3D motion in a fairly straight-forward manner. In the meantime, other structured environments for which to adapt this model are the insides and outsides of pipes.

## VII. ACKNOWLEDGEMENTS

The authors would like to thank the members of the Biorobotics Lab and Robotics Institute, particularly Ross Hatton, Matt Tesch, Austin Buchan and Justin MacEy.

## REFERENCES

- [1] G. Chirikjian and J. Burdick, "The kinematics of hyper-redundant robot locomotion," *IEEE Transactions on Robotics and Automation*, vol. 11, no. 6, pp. 781–793, 1995.
- [2] C. Wright, A. Johnson, A. Peck, Z. McCord, A. Naaktgeboren, P. Gianfortoni, M. Gonzalez-Rivero, R. L. Hatton, and H. Choset, "Design of a Modular Snake Robot," *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, Oct. 2007.
- [3] M. Tesch, K. Lipkin, I. Brown, R. L. Hatton, A. Peck, J. Rembisz, and H. Choset, "Parameterized and Scripted Gaits for Modular Snake Robots," *Advanced Robotics*, vol. 23, pp. 1131–1158, June 2009.
- [4] D. Rollinson and H. Choset, "Virtual Chassis for Snake Robots," *IEEE International Conference on Intelligent Robots and Systems (accepted)*, 2011.
- [5] J. Gray, "The Mechanism of Locomotion in Snakes," *Journal of Experimental Biology*, vol. 23, pp. 101–123, Dec. 1946.
- [6] B. C. Jayne, "Kinematics of Terrestrial Snake Locomotion," *Copeia*, no. 4, pp. 915–927, 1986.
- [7] S. Hirose, *Biologically Inspired Robots*. Oxford University Press, 1993.
- [8] A. A. Transeth, K. Y. Pettersen, and P. I. Liljebäck, "A survey on snake robot modeling and locomotion," *Robotica*, vol. 27, p. 999, Mar. 2009.
- [9] D. I. Goldman and D. L. Hu, "The mechanics of slithering locomotion depend on the surroundings," *American Scientist*, pp. 314–323, 2010.
- [10] Z. V. Guo and L. Mahadevan, "Limbleless undulatory propulsion on land," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 105, pp. 3179–84, Mar. 2008.
- [11] A. Transeth, R. Leine, C. Glocker, and K. Pettersen, "3-D Snake Robot Motion: Nonsmooth Modeling, Simulations, and Experiments," *IEEE Transactions on Robotics*, vol. 24, pp. 361–376, Apr. 2008.
- [12] A. A. Transeth, R. I. Leine, C. Glocker, K. Y. Pettersen, and P. I. Liljebäck, "Snake Robot Obstacle-Aided Locomotion: Modeling, Simulations, and Experiments," *IEEE Transactions on Robotics*, vol. 24, pp. 88–104, Feb. 2008.
- [13] W. Mosauer, "A Note on the Sidewinding Locomotion of Snakes," *The American Naturalist*, vol. 64, pp. 179–183, Mar. 1930.
- [14] J. Burdick, J. Radford, and G. Chirikjian, "A Sidewinding Locomotion Gait for Hyper-Redundant Robots," *Robotics and Automation*, vol. 3, pp. 101–106, 1993.
- [15] R. L. Hatton and H. Choset, "Sidewinding on slopes," in *2010 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 691–696, IEEE, 2010.
- [16] H. Choset, K. M. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L. E. Kavraki, and S. Thrun, *Principles of Robot Motion: Theory Algorithms, and Implementations*. 2005.
- [17] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. MIT Press, 2000.
- [18] H. Ohno and S. Hirose, "Design of slim slime robot and its gait of locomotion," *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 707–715, 2001.
- [19] M. Tesch, J. Schneider, and H. Choset, "Using Response Surfaces and Expected Improvement to Optimize Snake Robot Gait Parameters (2011)," in *International Conference on Intelligent Robots and Systems*, 2011.