Snakes on an Inclined Plane: Learning an Adaptive Sidewinding Motion for Changing Slopes

Chaohui Gong, Matthew Tesch, David Rollinson and Howie Choset
{chaohuig, mtesch, drollins, choset}@cmu.edu

***Abstract*— Sidewinding is an efficient gait adopted by biological and robotic snakes for locomoting on various terrains. The mechanics of this motion on flat ground and steady state terrains have been thoroughly investigated, while its capability to adapt to changing environments is not as well studied. We demonstrate the capability of a snake robot to automatically adjust gait parameters to optimally move up and down slopes of varying angle. This capability is achieved by three components. First, an efficient offline learning algorithm finds a policy mapping the estimated slope angle to the optimal gait parameters. Next, a robust online state estimation technique infers the local terrain characteristics. Finally, the precomputed policy is consulted online to select the optimal gait parameters for this slope. The efficacy of this approach is verified by robot experiments.**

I. Introduction

Snakes use cyclic internal shape changes called gaits to navigate through a variety of terrains. This versatility is the result of their ability to sense the changes of surroundings and act accordingly. Snake robots, on the other hand, have not begun showing equivalent capabilities mainly because they do not yet possess levels of sensing and control comparable to biological snakes. In order to reduce this gap between snake robots and their biological counterparts, this work presents an approach which integrates robust state estimation to improve sensing and an efficient learning algorithm to generate an optimal controller. Applying this approach to the snake robot in our lab has endowed the robot with the ability to autonomously sidewind efficiently through environments which vary in slope.

The snake robot needs to use its sensing capability to gather information about the terrain inclination and respond appropriately. As the snake robot moves on inclined surfaces, its pose is determined by the local terrain profile. It is hence possible to infer the slope angle using the estimated pose of the snake robot. Using a previously established state estimation technique [1], we empirically derive a relation between the estimated pose and the slope angle. This relation allows the snake robot to infer the local terrain inclination in real time.

Sensing the terrain is critical, but successful deployments of snake robots into the real world require being able to robustly control the system appropriately depending on the sensed terrain. Even with compact analytical controllers [2], describing the appropriate control parameters for each possible sensed state is tedious. When working with physical robots or other systems which are *expensive* to run (requiring significant time, money, or other resources), it is often intractable to densely sample over the space of all control parameters for all possible sensed states. Instead, an efficient search over the set of possible sensed states is imperative. For this reason, we use offline efficient learning methods for expensive systems to find an optimal policy mapping the sensed state (slope angle) to the optimal control parameter (gait parameter).

Fig. 1: The 16 DoFs modular snake robot. The robot is sidewinding up a sandy surface with a changing slope.

After learning the optimal control policy offline, we run an online closed-loop controller integrating the slope estimation and the learned policy. This enables the snake robot to minimize time required to move across a simulated environment by adapting gait parameters to changing inclined surfaces.

In the following sections, we first introduce background information and fundamental concepts which are extensively used throughout this work. We then present our approach for online estimation of the slope angle. Next, we find the speed-optimal policy for the sidewinding gait using an efficient policy optimization method. Finally, we verify the efficacy of this approach using robot experiments, outperforming strategies which use fixed control parameters.

II. Background

This work builds on a previously developed controller, called the *gait equation* [2], and a state estimation technique [1] for snake robots.

A. Gait Equation

Our research group has developed the 16 degree of freedom (DOF) modular snake robot [3], see Fig. 1. The joints are alternately aligned in the dorsal and lateral directions of the robot. Such a configuration allows the robot to execute gaits which require full 3-D motion (out of the forward/lateral plane). Inside every module, there is a 3-axis

accelerometer, 3-axis gyroscope, and an encoder at the joint. These sensor readings can be used to robustly estimate the orientation of the robot in the world.

A large body of research has been dedicated into the control of such a redundant system and many types of controllers have been previously proposed [4], including central pattern generation (CPG) [5], the *backbone curve* [6] and the gait equation. In this work, we adopt the gait equation representation because its compact, analytical representation is well-suited into our learning framework. The gait equation expresses all the joint movements as sinusoids,

$$\alpha(n,t) = \begin{cases} A\sin(\beta) & \text{odd} \\ eA\sin(\beta+\delta) & \text{even} \end{cases} \quad (1)$$

$$\beta = \Omega n + \omega t, \quad (2)$$

where A, e, β and δ are respectively amplitude, aspect ratio, phase and the phase shift between odd joints and even joints. The phase shift δ in the gait equation directly affects the characteristics of a gait. In [2], [7], it has been shown that sidewinding motion can be produced when $\delta = \frac{\pi}{4}$.

B. State Estimation

The first challenge of state estimation for a snake robot is to find a principled way to represent the system's configuration with respect to the world. A snake robot moves using cyclic internal shape changes. As a result, it is difficult to use any frame rigidly attached to the body to intuitively represent the configuration of a snake robot. Rollinson et al. [8] proposed an average body frame, called the *virtual chassis*, to represent the system. The virtual chassis frame originates at the center of mass (CoM) and its axes are aligned in the directions of the principal moments of inertia. In [9] and [1], it has been shown that the virtual chassis is stable relative to the world and can be used to intuitively represent the orientation of the system. Fig. 2 is an example showing the representational advantage of using the *virtual chassis* versus a body frame rigidly attached onto a module. The second challenge is to take full advantage of the redundant sensor measurements. In [1], the authors adopt an extended Kalman filter (EKF) for estimating the state of the virtual chassis. This method fuses the sensor measurements in the virtual chassis frame. In the present work, we build upon this state estimation for the robot to estimate the inclination of the terrain over which it is moving.

III. Slope Angle Estimation

The accurate estimation of the local terrain profile plays a critical role in the successful deployment of a snake robot in the field. Underestimating terrain inclination may lead to an aggressive choice of gait parameter (small amplitude which leads to rapid down slope motion) and, as a result, the robot may lose stability and tumble down an ascending slope. An overestimated slope angle may cause a conservative choice of gait parameters (large amplitude which results in better stability), sacrificing the traveling speed. Using previously established state estimation technique, we show that the terrain information can be inferred from the estimated robot states.

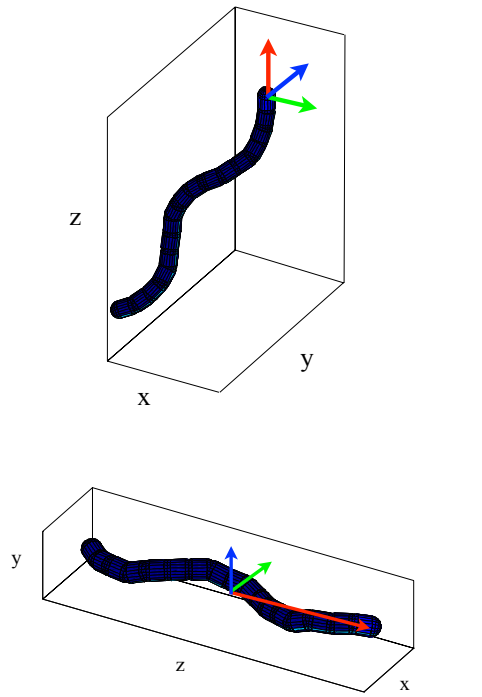


Fig. 2: When viewed in a frame fixed to the first module (top), the direction of motion of the snake during sidewinding is not intuitively related to the principle axis directions. When represented in the virtual chassis frame (bottom), the direction of motion is almost entirely aligned with the principle axis directions, making the frame much more intuitive.

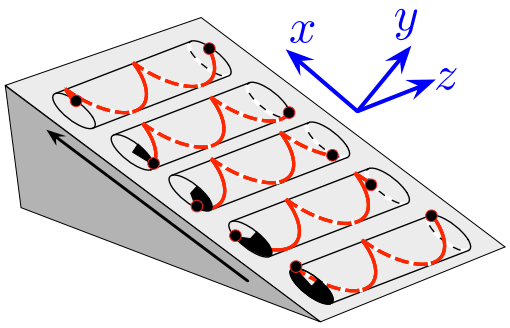


Fig. 3: The shape of a sidewinding snake resembles a helical tread rolling on the ground. The forward direction is defined as x in the virtual chassis frame.

A. Slope Angle Inference

As the robot sidewinds on inclined surfaces, the state estimation technique provides a robust orientation estimate of the virtual chassis,

$$R_{\text{VC}} = \left[\vec{r_x}, \vec{r_y}, \vec{r_z}\right] = \begin{bmatrix} r_{x1} & r_{y1} & r_{z1} \\ r_{x2} & r_{y2} & r_{z2} \\ r_{x3} & r_{y3} & r_{z3} \end{bmatrix}. \quad (3)$$

By solving for a rotation about the longitudinal $\vec{z}$ axis which embeds the $\vec{x}$ axis in the world-frame horizontal (x-y)

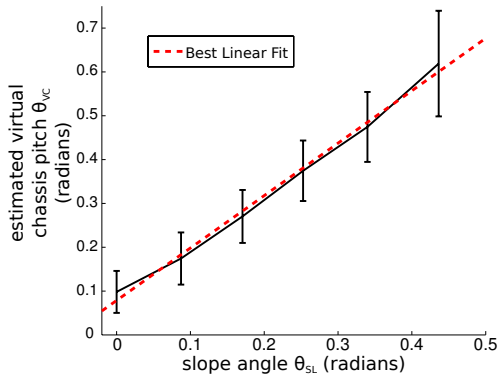


Fig. 4: The bars in the figure denote the variance of estimated virtual chassis pitch θ_{VC} with different gait parameters and on different slopes. A linear relation between the slope angle θ_{SL} and the virtual chassis pitch θ_{VC} is revealed.

plane, the estimated pitch angle of the virtual chassis can be obtained as $\theta_{\text{VC}} = \text{atan2}(|r_{x3}|, r_{y3})$.

Note that the direction of movement of the sidewinding gait is approximately aligned with the $\vec{x}$ axis of the *virtual chassis* (see Fig. 3); therefore, as the robot sidewinds on inclined surfaces, the pitch of the virtual chassis is strongly correlated to the inclination of the surfaces. However, as the snake robot is moving, the x-z plane of the virtual chassis in not guaranteed to be parallel to the slope. To correct for this effect, a relation between the estimated virtual chassis pitch θ_{VC} and the angle of the terrain slope θ_{SL} is empirically determined:

$$\theta_{\text{SL}} = \frac{1}{1.20}(\theta_{\text{VC}} - 7.85 \times 10^{-2}). \tag{4}$$

This empirical relation is based on measurements from a series of experiments in which the robot was commanded to sidewind 10 times, with different gait parameters, on slopes of 6 different inclinations. Fig. 4 shows a summary of the experimental data along with the obtained linear best-fit relationship between θ_{VC} and θ_{SL}.

To further improve real-time estimation, the estimated slope angle is then fed into a simple low-pass filter to smooth out noisy measurements,

$$\tilde{\theta}_{\text{SL}}[k] = \tilde{\theta}_{\text{SL}}[k-1] + \frac{\Delta t}{T}(\theta_{\text{SL}}[k] - \tilde{\theta}_{\text{SL}}[k-1]), \tag{5}$$

where Δt, T, $\theta_{\text{SL}}[k]$, $\tilde{\theta}_{\text{SL}}[k]$ are respectively the time interval, gait period, estimated slope angle, and smoothed slope angle at time step k.

IV. Optimal Policy

The goal of controlling the snake robot with an *optimal policy* is to maximize some measure of overall performance as the robot moves up and down various slopes. We define a *policy* as a map from slope (within the range that the robot may encounter) to control (i.e., gait) parameters. The optimal policy is then the policy which maps any slope to the best parameter for that slope. This can also be thought of as maximizing performance on each individual slope; therefore, finding the optimal policy requires solving an optimization problem for every possible slope (a potentially infinite number). Below we describe how we can build an approximation to this policy offline while still budgeting the number of tests on the physical robot.

A. Optimization using Unbiased Expected Improvement

In this paper, we define performance to be the speed of the robot as it climbs or descends, or more precisely the net distance (in the desired direction) traveled in 10 seconds on a given slope. Optimizing such a quantity and finding an optimal policy is difficult for such complex robotic systems, because (1) they exhibit hybrid, multi-mode dynamics for which we don't have an accurate model that would enable generation of a policy offline, and (2) we are limited in the time that can be spent measuring the performance of different gait parameters on different slopes (i.e., running experiments).

We are limited in the amount of testing we can do on the robot, yet the aforementioned optimization requires a potentially infinite number of optimization problems (one for each slope). To overcome this apparent roadblock, we draw on work from [10] which allows us to search for such an optimal policy while conserving the number of tests which must be run using the robot. Briefly, this is accomplished by using the observation that the cost function is likely to be continuous and reasonably smooth not only with respect to the control parameters, but also with respect to the slope we are climbing; similar controls when run on similar slopes will result in similar performance (for example, see the final estimate of the reward function in Figure 6(b) or Figure 7(b)).

In more detail, this method sequentially chooses which parameters to sample, demonstrating that careful choice of these parameters using all available knowledge of the previous tests is more effective than simple, random selection when samples are limited. After an initial sampling, a Gaussian process (GP) [11] is used as a function regression method to fit the sampled points with a surface $\hat{f}$. The predictive distribution generated by the GP is then used by a probabilistic sampling method to select the next control parameter and slope angle to evaluate the performance at.

The sample chosen is that which maximizes the *Unbiased Expected Improvement* (see the derivation in [10] for more detail). This approach adapts the basic idea of *expected improvement* [12], widely used in the global optimization of expensive-to-evaluate objective functions, to the explicit separation of the environment (slope) and control (gait parameter) space. I consider the expected improvement of sampling at some combination of environment and control parameters $x^t = (x_e^t, x_c^t)$, but measure improvement over the maximum estimated value *for that environment (slope)* (when $x_e = x_e^t$), or $y^*_{x_e^t} = \max_{x_c \in X_c} \hat{f}_\mu(x_c, x_e^t)$, instead of over the best objective evaluation so far, $\max(\tilde{Y})$. This gives the Unbiased Expected Improvement (UEI):

$$\text{UEI}(x^t) = \omega(x_e^t) \int_{y^*_{x_e^t}}^{\infty} p_Y^{x^t}(y)(y - y^*_{x_e^t})\, dy, \tag{6}$$

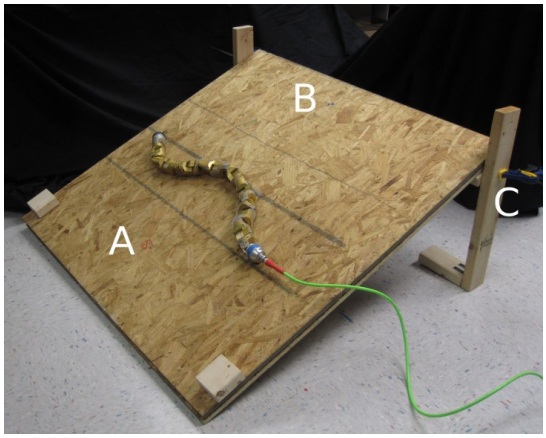


Fig. 5: The training setup consisted of a 4 foot by 4 foot plywood sheet. The robot began at a small circle at point A for uphill trials, and one at point B for downhill slopes. The height of the test setup was changed when requested by the optimization algorithm by adjusting one quick clamp on each side of the plywood sheet, and clamping on to small blocks of wood attached to the underside of the sheet. The distance traveled was measured by comparison of the snake's position at the end of ten seconds to ruled markings drawn on the wood.

where $p_Y^{x^t}$ is the predictive distribution generated by the GP at x^t. This point is then sampled on the true objective (i.e., an experiment is run using the robot), and the selection process repeated.

After this careful experiment selection process exhausts the budget of experiments available, an optimal policy is generated. The GP can be used as a "cheap-to-evaluate" surrogate for the real robot, and is used to generate policies which, while not provably optimal, have been empirically shown to be very good for such a small number of tested data points.

B. Experimental Methodology

The reward function, as described above, is the distance (projected onto the direction of the slope gradient) between the start point of the robot's center of mass, and the end point after 10 seconds of gait execution. Note that negative values were reported if a robot traveled a negative distance (such as rolling back down hill after tipping over while traveling uphill). When the robot exceeded the allowable range (such as fast rolling motions downhill), the robot was timed and a constant velocity assumption was used to extrapolate downhill distance. When collecting data, we chose to reduce the noise inherent in our data by running each sample 5-8 times (outliers were removed, and extra samples were not taken when data was extremely consistent from the first samples). Notably, this reward function produced a reasonably well behaved cost function, allowing the adaptive expensive optimization method to be used. The test setup can be seen in Figure 5.

A total of 25 slope/control parameter combinations were tested; after the initial 10 random samples, the experiment selection method selected the next 15. The parameter ranges included a slope from -25 to $+25$ degrees for the environment parameter, and a horizontal amplitude of 0 to 0.9 as per Eqn. (2). The ratio of horizontal to vertical amplitude was restricted to $8/3$, and the spatial period was fixed at one and a half wavelengths (1.5 periods of waveforms are generated along a 16 module snake robot).

C. Results of the Optimization

As an illustration of the optimization process, Figure 6 shows the predicted reward function and resulting experiment selection metric during a subsequence of the samples that were taken. The entire distribution of the sample locations is shown in Figure 7(a). Note that random selection would not define the boundaries of the policy as well, and that simple optimization of the reward over all slopes would focus towards the overall peak, resulting in a policy performing well only for slopes near that peak. The adaptive optimization approach seeks a policy which performs well on all slopes.

Finally, the resulting optimized policy is shown in 7(b). Note that it can be discontinuous, as its value (the control for a given slope) at each slope is determined by running any standard optimization method over the (cheap to evaluate) "slice" of the GP's predicted surface for that slope; this is independently done for each slope, and therefore allows for jumps in the optimal control between one slope and another. In our implementation, we ran a set of such optimizations over a dense sampling of the slopes using the GP (this only took around 3 seconds on a modern laptop), and stored these values in a simple lookup table so they could be retrieved later during online testing.

V. Experiments

A. Test Environment

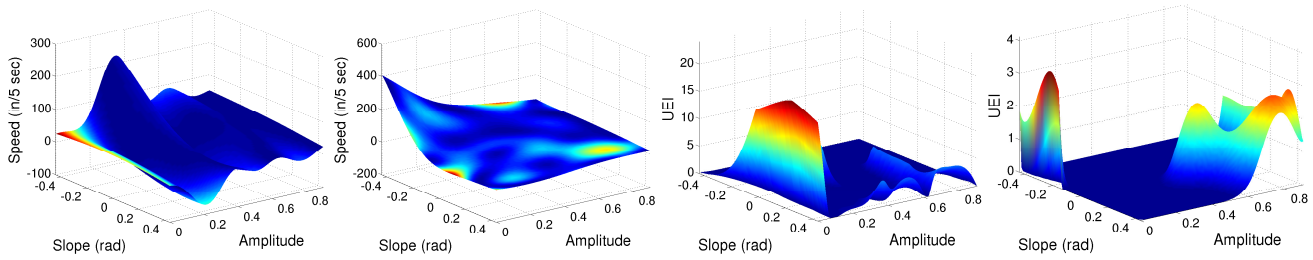
Our test environment (see Fig. 8) is built of 4 boards of different slope angles chained together. The tilt angle of each board relative to the ground can be independently adjusted; different slope angles are constructed by adjusting the heights of the two ends of a board. The movement of the snake robot traversing the test environment is videotaped to record the time required to complete each for each slope segment. The moving speed can be calculated by differentiating the positions of the snake robot at two consecutive time steps.

B. Motion Control

As the snake robot moves along a board, there is a chance for the snake robot to fall off because small disturbances can cause the robot to not necessarily move straight. To avoid falling, we command the snake to move in the middle of the boards by using *conical sidewinding*, which is an approach proposed to steer the heading direction of sidewinding [13]. During the execution of sidewinding on the test environment, we manually adjust the heading direction to prevent the snake robot from falling off the test field.

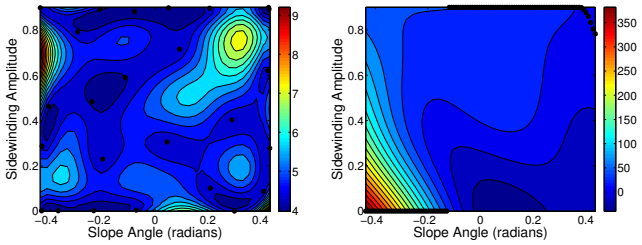
C. Results

We test the performance of the real time gait adaptation of sidewinding in two settings of our test environment. The tilt angles of the four boards relative to the ground are respectively 0.21, 0.43, -0.33 and -0.31 radians. As the snake robot climbs up a slope, a large amplitude in (2) is



(a) Cost function estimate after 18 tests (b) Cost function estimate after 24 tests (c) Selection metric after 18 tests (d) Selection metric after 24 tests

Fig. 6: The top row shows the changes in the cost function as the number of samples increases. The height of the surface indicates the predicted value, whereas the color indicates the uncertainty. The left corner is untested and uncertain after 18 and 21 experiments; after 24, however, the function has been sampled at this location and causes a significant change in the surface prediction. The bottom row shows the value of the selection metric, the unbiased expected improvement. As opposed to a simple optimization of the space, samples are still collected for slope angles which have no high predicted speed; this allows the algorithm to learn a policy that performs well for all values of the environment parameter rather than only the "easy" environments.



(a) (b)

Fig. 7: **(a)** The uncertainty in the cost function estimate after the sampling budget has been exhausted. The black dots in the figure are the 25 sampled points. Note the uncertainty is highest furthest from the sampled points. **(b)** The predicted cost function after the sampling budget has been exhausted. The optimal policy is shown by the large black dots; note that at steep downhill slopes, the optimization determines that a zero-amplitude sidewinding gait is most effective; this gait rolls down the hill very quickly. At a critical slope, the optimal control switches to a high amplitude sidewinding gait. Interestingly, at very steep slopes the optimal control decreases slightly; perhaps this decreased amplitude increases the stability or repeatability of the motion.

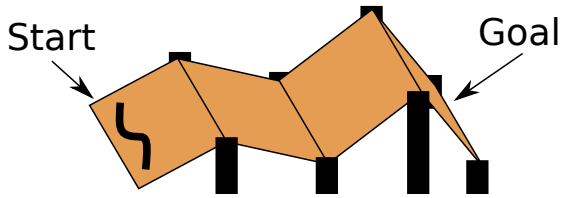


Fig. 8: Diagram of test course. The robot was placed at the start location to begin, with the task of reaching the goal location in the minimum amount of time. The slope of each segment can be set independently, and is estimated in real time as the robot moves along the segment.

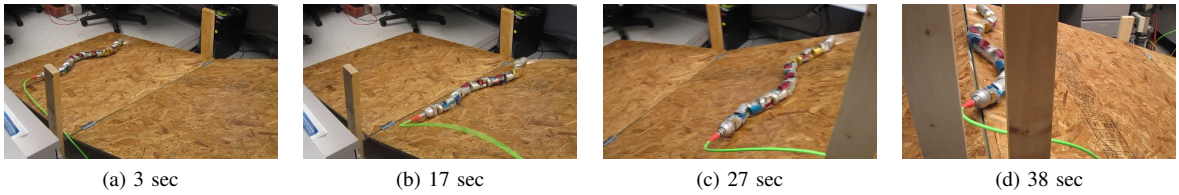
selected according to the optimal policy and when the snake robot is on a negative slope, zero amplitude is chosen to make the snake robot quickly roll down the hill (c.f. the selected frames from a recording of this test in Fig. 9). The plot in Fig. 10 shows the traveled distance along the boards versus the time consumed in this experiment. It can be easily seen that fast movement is achieved by setting the amplitude to zero and rolling down the negative slopes in the second half of the experiment.

To evaluate the improvement gained through this method, we compare the performance of adaptive sidewinding to the performance of sidewinding with fixed gait parameters on a second test field, with angles of the four boards relative to the ground of 0.32, -0.32, 0.34, and -0.34 radians (forming a 'M' shape terrain). As a conservative choice of gait parameter which provides good performance on an ascending slope, we choose the amplitude to be 1. We also compare the performance of sidewinding with a small amplitude $A = 0.45$, which is most effective on negative slopes. Fig. 11 shows the performance of sidewinding with different strategies. As we can observe from Fig. 11, sidewinding with a low amplitude cause the robot to tumble down 3 times before climbing up an ascending slope (0.32 radians). Though sidewinding with large amplitude (conservative) succeeded in traversing the whole test field, the traveling speed is much lower compared to the adaptive sidewinding, especially on negative slopes. Sidewinding guided by our optimal policy consumed only 36 seconds passing through the test field compared to 71 seconds required by sidewinding with large amplitude.

VI. Conclusions and Future Work

In this work, we propose an approach to infer the tilt angle of a slope, on which a snake robot is moving, based on the estimated state of its *virtual chassis*. Also, we efficiently find an optimal policy which maximizes the traveling speed of sidewinding based on the inferred slope angle. By integrating these two parts of work, a sidewinding snake robot can automatically adjust its gait parameters to maximize its moving speed of movement on changing slopes. Improved performance is achieved compared to sidewinding without adaptivity to a changing environment.

For the future work, we would like to pay attention to the transition behavior of sidewinding from one slope to another. Also, we plan to take the dynamics of the snake movement into account to achieve better performance. For example, the inertia of a snake robot builds up when it is rolling down a negative slope, and we can take advantage



(a) 3 sec (b) 17 sec (c) 27 sec (d) 38 sec

(e) 40 sec (f) 42 sec (g) 43 sec

Fig. 9: For this setup of our test course, the optimal control policy chose a sidewinding parameterization with a wide base for the uphill segments, and quickly transitioned to a zero-amplitude roll for the downhill segments. Note the time each frame was taken, and the difference in speed when moving up- versus down-hill.

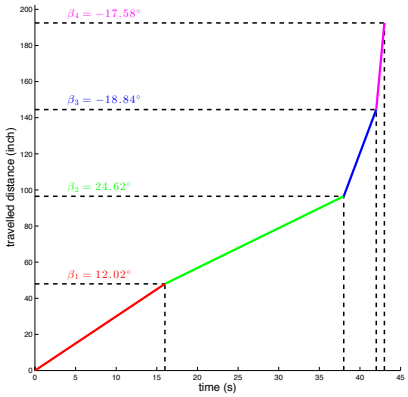


Fig. 10: This figure shows the performance of adaptive sidewinding. The slope of each line denotes the traveling speed, and different colors mean boards at different tilt angles.

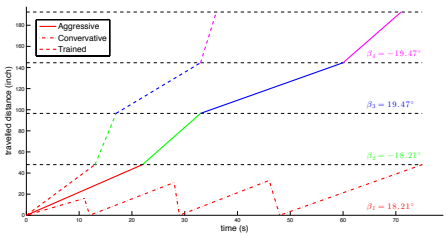


Fig. 11: This shows a comparison among three strategies (showed in different line types). Sidewinding with a small amplitude failed to finish the task, and sidewinding with a large amplitude (conservative) leads to poor performance.

of this inertia to keep the snake robot rolling up an adjacent ascending slope. Another improvement is the possibility of optimizing the policy over a larger parameter space (e.g., more gait parameters), or potentially allowing the gait to be selected as well (adding this as an additional discrete parameter). Finally, we can improve the autonomy of our system by integrating a method to detect position relative to the boundaries while on a slope, and eliminating the need for manual control of the steering to keep the system centered.

ACKNOWLEDGMENTS

We would like to acknowledge Ross L. Hatton for the permission to use Fig. 3.

REFERENCES

[1] D. Rollinson, A. Buchan, and H. Choset, "State estimation for snake robots," in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, sept. 2011, pp. 1075 –1080.

[2] M. Tesch, K. Lipkin, I. Brown, R. Hatton, A. Peck, J. Rembisz, and H. Choset, "Parameterized and scripted gaits for modular snake robots," *Advanced Robotics*, vol. 23, no. 9, pp. 1131–1158, 2009.

[3] C. Wright, A. Buchan, B. Brown, J. Geist, M. Schwerin, D. Rollinson, M. Tesch, and H. Choset, "Design and architecture of the unified modular snake robot," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, may 2012, pp. 4347 –4354.

[4] A. A. Transeth, K. Y. Pettersen, and P. Liljebäck, "A survey on snake robot modeling and locomotion," *Robotica*, vol. 27, no. 07, pp. 999–1015, 2009.

[5] J. Gonzalez-Gomez, H. Zhang, E. Boemo, and J. Zhang, "Locomotion Capabilities of a Modular Robot with Eight Pitch-Yaw-Connecting Modules," in *9th International Conference on Climbing and Walking Robots.*, 2006.

[6] J. Burdick, J. Radford, and G. Chirikjian, "A "Sidewinding" Locomotion Gait for Hyper-redundant Robots," *Robotics and Automation*, vol. 3, pp. 101–106, 1993.

[7] R. L. Hatton and H. Choset, "Generating gaits for snake robots: Annealed chain fitting and keyframe wave extraction," *Autonomous Robots, Special Issue on Locomotion*, vol. 28, no. 3, April 2010.

[8] D. Rollinson and H. Choset, "Virtual chassis for snake robots," in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, sept. 2011, pp. 221 –226.

[9] F. Enner, D. Rollinson, and H. Choset, "Simplified motion modeling for snake robots," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*. IEEE, 2012, pp. 4216–4221.

[10] M. Tesch, J. Schneider, and H. Choset, "Adapting Control Policies for Expensive Systems to Changing Environments," in *International Conference on Intelligent Robots and Systems*, 2011.

[11] C. K. I. Williams and C. E. Rasmussen, "Gaussian processes for regression," *Advances in Neural Information Processing Systems*, 1996. [Online]. Available: http://eprints.aston.ac.uk/651/

[12] J. Mockus, V. Tiesis, and A. Zilinskas, "The application of Bayesian methods for seeking the extremum," *Towards Global Optimization*, vol. 2, pp. 117–129, 1978.

[13] C. Gong, R. Hatton, and H. Choset, "Conical sidewinding," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, may 2012, pp. 4222 –4227.