2013 IEEE International Conference on Robotics and Automation (ICRA)
Karlsruhe, Germany, May 6-10, 2013

Snakes on a Plan:
Toward Combining Planning and Control

Ross L. Hatton[1], Ross A. Knepper[2], Howie Choset[3], David Rollinson[3], Chaohui Gong[3], and Enric Galceran[4]

***Abstract*— Highly articulated robot locomotion systems, such as snake robots, present special motion planning challenges. They possess many degrees of freedom, and therefore are modeled by a high dimensional configuration space which must be searched to plan a path. Kinematic and dynamic constraints further complicate the selection of effective controls. Finally, snake robots often have multiple modes of interaction with the terrain as contacts are made and broken, leading to complex and imperfect motion models. We believe that the space of useful controls that provides desirable motions, however, is much smaller. Useful net motions for such systems are often generated via gaits, or cyclic motions in the shape space. Gaits transform a high-dimensional continuum search into a relatively tractable discrete search. In this paper, we put forward a framework which allows a planner to generate paths in a low dimensional work space and select among gaits, preplanned motions in the robot's shape space. The contribution of this paper rests on the "virtual chassis" which is a choice of body frame for the snake robot that allows the planner to efficiently select among and plan with gaits to direct the robot along the work space path. We demonstrate this planner running on a simulated snake robot navigating through a variety of clutter scenarios. The virtual chassis also has the benefit of allowing us to generalize notions of controllability to gait motions.**

I. INTRODUCTION

Highly articulated locomotors offer a versatility unavailable to simpler mobile robots. Snake robots, for instance, are often suggested for tasks such as urban search and rescue or covert surveillance, where their thin, flexible bodies allow them to fit into confined spaces and adapt to challenging terrain [1]. This flexibility, however, comes at a cost: planning and control of a highly articulated system requires effective management of a high-dimensional configuration space.

One means of reducing this complexity is to identify a library of useful motion primitives such as *gaits*—cyclic changes in the robot's shape (joint angles) that induce characteristic net displacements in its position—and plan locomotion tasks for the robot in terms of these gaits, rather than directly in terms of differential control inputs. A well-chosen library of gaits can significantly shrink the size of the robot's control space while maintaining much of its expressiveness: there is typically a preferred pattern of shape changes to move the system in any given direction. Once a library of gaits and other motion primitives has been identified for a system, planning its motions becomes a matter of sequencing the gaits to take the system between start and goal locations.

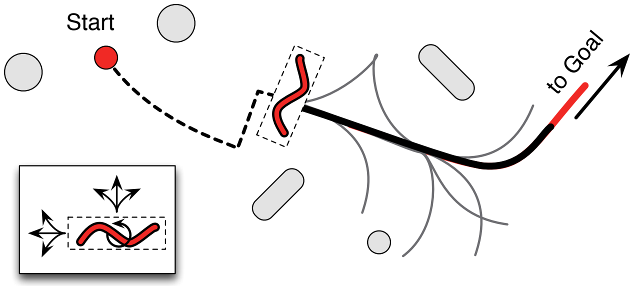


Fig. 1. Gait-based motion planning for a snake robot. The planner first finds a global path (shown in red) avoiding the obstacles, and then builds up short sequences of motion primitives to approximately follow this path. The currently selected sequence of primitives is represented by the thick black line, whereas rejected paths are shown as thin gray lines, and the path executed so far is represented by the dashed line trailing the snake robot. As shown in the inset, the robot can execute gaits to slither longitudinally, sidewind laterally, or turn in place. The gait sequencer uses a *virtual chassis* representation of the snake robot, abstracting the motion of the robot to that of a rectangular vehicle that can drive in multiple directions.

The snake robotics community has focused significant attention on gait generation and the physics of locomotion. Less attention, however, has been paid to algorithms for combining these gaits into higher-level motion plans. As a first step in filling this gap, this paper describes a planner that draws on our recent *virtual chassis* [2] and *virtual tread* [3] techniques to represent the bulk motion of a snake robot slithering and sidewinding as that of a simpler vehicle model that can drive both longitudinally and laterally. We then adapt a standard planner, originally developed for navigating car-like systems through obstacle fields, to work with this vehicular-snake model. As illustrated in Fig. 1, the planner generates a high-level path around the obstacles, then lays down short sequences of these "driving" motions to follow the path.

We demonstrate the efficacy of this planning approach as applied to our "Modsnake" [4] family of robots. In a series of simulation experiments, we show that the planner is effective in penetrating mild-to-moderate obstacle clutter with graceful

---

*This work is sponsored in part by The Boeing Company and the Collaborative Technology Alliance. We are grateful for this support.

[1] R.L. Hatton is with the School of Mechanical, Industrial, and Manufacturing Engineering, Oregon State University, Corvallis, OR, USA `ross.hatton@oregonstate.edu`

[2] R.A. Knepper is with the Computer Science and Artificial Intelligence Lab, Massachusetts Institute of Technology, 32 Vassar St, Cambridge, MA, USA `rak@csail.mit.edu`

[3] H. Choset, D. Rollinson, and C. Gong are with the Robotics Institute, Carnegie Mellon University, 5000 Forbes Ave, Pittsburgh, PA, USA `choset@ri.cmu.edu`

[4] E. Galceran is with the University of Girona, Edifici P-IV, Campus de Montilivi, Girona, Spain `enricgalceran@eia.udg.edu`

978-1-4673-5643-5/13/$31.00 ©2013 IEEE

degradation in performance in response to increasing clutter. We then show two experiments to serve as a physical poof-of-concept of the method.

II. PRIOR WORK

A. Prior Work in Primitives

Over the past decade, a variety of highly-articulated robots have utilized motion primitives in planning. One example includes work that uses primitives to plan paths for humanoid forms. Kajita et al. [5] created an abstraction of dynamic humanoid robot balance behavior embodied in the zero-moment point (ZMP) control algorithm by means of a simple model of a cart on a table. Kuffner et al. [6] then presented the first general motion planning algorithm to successfully execute motions on a real humanoid robot by generating dynamically-stable trajectories between consecutive pairs of precomputed statically-stable poses. Graphical character animation [7] approaches employ prerecorded motion capture data of a human figure performing a variety of motions. Given a desired trajectory for the character, motion primitives can be "played back" on the animated figure to produce realistic motion [8].

Planning with primitives often induces a hierarchical planner with a replanning step to accommodate constraints in the system and the environment. For example, Candido et al. [9] constructs a hierarchical planner where the global level constructs a graph and searches it to find an approximate path to the goal. The local planner then composes motion primitives in an attempt to follow that global path. When the local planner is unable to do so, the global planner forces a replan, presenting a new candidate path to follow. Recently, there has been success in using probabilistic techniques for the high and low-level planners. Hauser et al. [10] employ motion primitives to the problem of walking in rough terrain for two- and six-legged robots. Their approach begins with footstep planning and uses precomputed motion primitives to bias the search of a PRM toward a solution that connects pairs of footstep configurations. The net result is quasi-static walking and even climbing behavior. Shkolnik et al. [11] go a step further by achieving dynamic motion for a quadruped in rough terrain. They employ an RRT to compose specialized motion primitives designed to produce a bounding gait that can be tuned for foot placement in rough terrain.

B. Snake Robots: Mechanisms and Gaits

Snake robots—actuated chains that locomote by changing shape to push against their environments—have been studied since at least 1971 [1]. Their many internal degrees of freedom make snake robots extremely versatile, both in the sense that they can adopt widely varying shapes, and in their ability to use the full lengths of their bodies to execute a range of locomotion styles. Early snake robots, such as Hirose's pioneering Active Cord Mechanism (ACM) [1], were primarily planar and served as a means to study the mechanics of lateral undulation. Later versions, such as Yim's Polybot [12], Hirose's refinements of the ACM concept [13, 14], Shen's SuperBot [15], our modular snakes ("Modsnakes") [4, 16], SINTEF's Aiko [17], and Gonzalez-Gomez et al.'s Hypercube [18, 19], have included actuators to lift segments of the robot's body out of the plane, enabling locomotion modes such as sidewinding, lateral rolling, and, with sufficient actuator strength, helical pole climbing [4]. Most of these robots have been designed for operation in indoor lab space. Some, such as McIsaac and Ostrowski's anguilliform (eel-like) robot [20] and Hirose's ACM-R5 [14], have been sealed for underwater operation. A few others, including Borenstein's OmniTread [21] and recent iterations of our Modsnakes [4, 22], have been hardened for use in the dirty environments characteristic to realistic search-and-rescue scenarios.

Investigation of motion planning and control for snake robots ranges from largely theoretical studies on planar undulatory locomotion [23–25] to empirical investigations of how physical snake robots move in response to different inputs [4, 19]. Within this range, researchers have considered topics including inverse kinematics techniques to fit snake robots to three-dimensional spatial curves [26–29], the use of neurally-inspired central pattern generators to generate control signals [30], and means for exploiting obstacles in the environment as locomotive aids [4, 17, 31]

Much of this work has revolved around the design of *gaits*, cyclic variations in the robot's shape (as parametrized by its joint angles or other internal configuration variables), that produce useful displacements to its position and orientation in the world. Gaits employed by snake robots are similar to those seen in biological snakes, and typically take the form of bending waves that propagate along the body of the snake, inducing reaction forces against the ground that propel the robot through the world. Various approaches to gait design specify these waves in terms of body curvature [1, 4, 18, 19, 24, 29, 32] (often employing a *serpenoid wave* [1] in which the curvature changes sinusoidally along the length of the snake), three-dimensional shape [3, 28, 33, 34], or as the emergent behavior of central pattern generator (CPG) networks [30].

The most effective ground-traversal gaits that have been found for snake robots include slithering, in which the robot moves longitudinally (along the line from tail to head) and sidewinding, in which the robot moves laterally (orthogonally to the tail-head line).[1] Each mode of locomotion offers its own advantages. Slithering snakes are narrow in their direction of motion, as illustrated in Fig. 2, and so can fit through small gaps between obstacles. Sidewinders present a broader profile, but tend to be faster and more efficient; they use a treadlike rolling contact with the world [3, 35] instead of the dissipative sliding contact seen in slithering [1].

In their basic forms, slithering and sidewinding translate the robot without rotating it. Both gaits can also be "steered" via simple modifications: biasing the slithering waveform, so that the snake moves along an arc [20, 24], or tapering the

---

[1] Other gaits also exist, such as concertina motions for bracing against the walls of narrow corridors, and helical gaits for climbing poles. In this paper, we focus our attention on gaits for moving the robot across flat ground.

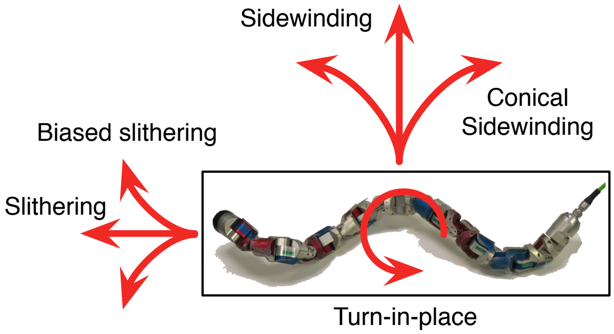


Fig. 2. Gait library with slow and narrow slithering, broad and fast sidewinding, as well as their variations for steering. The robot shown is the most recent from our "Modsnake" family of snake robots [22].

sidewinding shape into a cone so that one end of the robot moves faster than the other [36, 37]. Sidewinding can be further modified to turn the robot in place by reversing the motion pattern over half the snake [4], thus completing the library of gaits shown in Fig. 2.

III. PLANNING IN THE SPACE OF GAITS

The planning framework in this paper abstracts complex mechanical systems into two layers: mission-level and execution. This approach lets the planner focus on mission-level issues rather than on the particulars of the platform and environment dynamics. For snake robots, we separate the planning into two phases: a high-level phase that directs the overall macroscopic motion of the robot and a low-level planner that executes the gait to maneuver the robot in order to carry out the high level plan. Gaits are cyclic paths in the shape space where the robot starts and stops at the same internal shape. Much prior work in gait design seeks to establish a relationship between gaits in the shape space and displacements in the position space. In this paper, this relationship is achieved with the virtual chassis. With this capability in-hand, a planner can effectively generate a "low-level" compatible path and select among gait primitives to execute that path. Finally, we discuss in this section a notion of controllability which ensures that the library of gaits is sufficient to execute the path.

A. The Planner

The use of motion primitives to decompose motion instructions for dynamical systems into basic building blocks is not new. The contribution here is not the planner, nor the controller (gaits) for the snake robots, but rather how we combine planning and control while respecting local constraints—without suffering from an explosive growth in complexity.

The planning framework presumes that the configuration space of the robot can be separated into position variables, say x, y, and θ in the plane, and shape variables, the internal joint angles of the robot. When operating on flat ground, our 16 degree of freedom robot has 19 degrees of freedom: three for position and orientation, and 16 for shape. Assume without loss of generality that we used the gait design techniques from our prior work to create a set of maneuvers. Each gait has associated with it a bounding box which is the smallest rectangle that envelops the snake robot while it executes the gait. We also associate with each gait a rigid body transformation that describes the displacement between the start and goal pose of the rectangle when the gait is executed. With the rigid body displacements in-hand, the high-level planner creates a discrete policy to drive the abstract robot, *i.e.,* the bounding box, from any initial state in the free space to a goal state. Once the sequence of high level motions is created, then so are the sequence of gaits.

Note that the framework does not assume a specific planner and thus any conventional search, such as D* Lite [38] or RRTs [39], or the Model-Based Hierarchical Planner (MBHP) [40], can be used to create the initial policy. For the sake of explanation, assume the planner generates a policy using a tree data structure. The path tree grows incrementally by sampling unexplored actions—potential expansion sites—throughout the tree. The planner adds a set of new sites at the endpoint of each expanded path, corresponding to the allowed motion primitives. Many possible policies are available to guide the selection of the next site for expansion, including random or breadth-first strategies. One may bias the selection via an A*-like heuristic function [41], if desired.

When selecting an expansion site, the planner generates a new path segment in the search tree corresponding to some motion primitive. That primitive is taken as a control input used to steer the robot open-loop for one gait cycle. A forward model predicts the course and end-point of that path.

Successive path commands are planned in real time while the robot executes the prior command. Thus, there is no wait time between execution of consecutive motion primitives by the robot. Real-time execution imposes a deadline on the search process. Selection is performed in an anytime manner using an objective function based on the energy required for the robot to execute a given sequence of motion primitives. Each gait is associated with a characteristic power requirement, which is integrated over the length of time for which that gait is executed to get energy. Thus, the total energy expenditure of a given sequence of motion primitives can be estimated and the minimum-energy route to the goal selected for execution.

B. Virtual Chassis and Virtual Wheels

A convenient aspect of the gait library in Fig. 2 is that it lets us abstract away the internal motions of the snake robot and treat the system as a vehicle that can drive both laterally and longitudinally. In this abstraction, we take a bounding box around the snake (defined by the robot's geometric center and principal axes) as the "virtual chassis" [2] of the vehicle. The gait cycles then become "virtual wheels" or "virtual treads" [3], propelling the chassis in given body-frame directions.

Using the bounding box to represent the position and orientation of the robot plays two key roles in formulating the vehicle model. First, it lets us approximate the gaits in

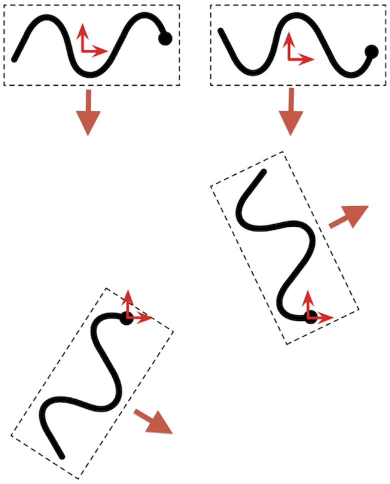


Fig. 3. Phase independence from the virtual chassis. When the overall position of the snake (its body frame) is defined by the center of mass and principal moments of inertia (top), the bounding box's placement relative to the body frame is independent of the gait phase. If another frame is chosen, such as the head (bottom), the bounding box moves relative to the body frame as the snake moves through different phases of its gait.

the library as constant body velocity controls, mathematically encoding notions such as "sidewinding produces lateral velocity." During sidewinding and slithering, symmetries in the gait waves mean that the the virtual chassis bounding box is propelled at a steady velocity (independent of the phase in the gait cycle) [2]. For other choices of body frame such as the head module, in contrast, the body velocity oscillates significantly with gait phase. Without phase independence afforded by the virtual chassis frame, the vehicle model would need to include these phases as extra parameters. Therefore, the virtual chassis offers considerable computational benefits over other frames, such as those affixed to a rigid body on the snake robot. In fact, for such "naive" choices of frame, the bounding box would oscillate so much that far more collision checks will be required to determine if it can pass from start to final internal configuration when executing the gait.

Second, for the collision checks we do need to execute, identifying the body frame with the bounding box simplifies the representation of robot-obstacle interactions. If we choose a constant-size bounding box (which may be slightly conservative for some gaits), the vehicle model becomes a rigid body translating and rotating in the plane. Obstacle expansion for collision detection with the bounding box can then be carried out via a standard $SE(2)$ obstacle expansion algorithm like Minkowski sum, avoiding the dimensional and geometric complexity of accounting for the robot's degrees of freedom.

C. Controllability

An effective gait library must contain enough gaits that it can usefully follow the planner's directives, but not so many that selecting one becomes a computationally burdensome task. The sufficiency of a given gait library is closely related to notions of *controllability* [42] and *maneuverability* [43]. At a high level, controllability (or more specifically, *small time local controllability*) establishes whether the actions available to a system locally span the space in which it operates. Maneuverability (an extension of robotic arms' *manipulability* [44] to locomoting systems) additionally considers the magnitude of this control authority in different directions. These notions are traditionally applied to systems with differential inputs, but as we explored in [43] have natural discrete equivalents when applied to systems moving via gaits.

The gait library shown in Fig. 2 satisfies this basic notion of discrete controllability, as the robot has actions to move it independently in x, y, and θ. The steering gaits (biased slithering and conical sidewinding) increase its maneuverability: rather than having to spend time switching between turning and translation gaits to follow a curved path, the snake can execute a single gait that combines these motions.

D. Obstacle Interaction

The response of a given control cannot be accurately precomputed due to interaction with obstacles. In conventional mobile robot motion planners, obstacle interaction is forbidden, and so colliding path candidates are culled away during planning. By contrast, many highly-articulated systems can increase their capabilities through contact with obstacles in the environment. This opportunity presents a tradeoff, though, as such obstacle interactions are also much harder to accurately predict. Consequently, we choose to allow the planner to select paths that make contact with obstacles, but we penalize such paths with a cost chosen to reflect these drawbacks. Thus, when a non-colliding trajectory is available, the planner chooses it over one that results in a collision.

In the event that the planner selects a colliding path, a subsequent replan is needed to account for the quantitative error. Indeed, the virtual chassis abstraction hides the true snake geometry, which is often vital in determining the true effect of obstacles on the robot. At best, the planner can only predict the result of a collision of the virtual chassis bounding box with an obstacle. That the planner can recover from disturbances introduced by such an imprecise model in a single step demonstrates the value of a reactive replanning strategy.

As a matter distinct from controllability, one might ask whether in the presence of obstacles a particular gait library is sufficient to find a path. The answer is ultimately a matter of resolution completeness. Since any of the planners we consider are resolution complete, there exists some sampling resolution sufficient to find a path, provided that one exists. Our fixed-resolution gait library is not guaranteed to solve all possible planning problems, but it comfortably solves the class of problems comprising mild-to-moderate obstacle clutter we target in this paper. For maneuvering in dense, three-dimensional clutter, other types of planner may prove more effective.

TABLE I

GAIT PARAMETERS

Gait	Speed	Power	Time	Distance	Efficiency
Sidewind	0.26 m/s	39.4 W	4.0 s	1.04 m	42.1 mWh/m
Slither	0.02 m/s	33.1 W	5.0 s	0.11 m	417.9 mWh/m
Turn-in-place	1.0 rad/s	39.0 W	0.8 s	0.8 rad	8.7 mWh/rad

Time and Distance describe one cycle of the gait motion. Speeds were determined empirically from the simulated snake. Power requirements are derived from Tesch et al. [45].

IV. SIMULATION RESULTS

The real robot under simulation is depicted in Fig. 2. Simulation is necessary in order to obtain a statistically significant quantity of results. In recognition of the complexity of snake dynamics, we utilize an external physics-based simulator implemented on top of the Open Dynamics Engine (ODE) library to represent ground truth. The planner itself has no access to this sophisticated simulator, instead employing its own faster but simpler predictive model. ODE qualitatively replicates the complex ground truth behavior of a snake robot executing a gait as well as the unpredictable interaction of the snake body with the ground and obstacles. Expected velocities for the gaits, shown in Table I, were empirically derived in the simulation environment and paired with experimentally-generated power costs from a Modsnake robot [4].

A. Setup

The Model-Based Hierarchical Planner (MBHP) [40] was used to select gaits for the simulated robot. A set of randomly-generated planning queries was presented to each planner at various levels of obstacle clutter. Planning problems consist of a 20m square room with uniformly distributed random 10cm-square obstacles. Workspace obstacle coverage densities vary from 0% (empty) to 1.75% (moderate clutter). Start and goal configurations are randomly chosen to be separated by 14 m.

Experiments were conducted in batches of one hundred problems with fixed-density, randomly-sampled, uniformly-distributed obstacles. Densities are reported as a fraction of the workspace covered, so even a 2% coverage translates into a densely-cluttered configuration space.

The planner returns success when the simulated robot reaches the goal position. Failure occurs when no sequence of motion primitives tested by the planner would make progress toward the goal. Due to the randomly generated nature of the planning problems, it is unclear whether failures indicate an impossible problem or a simple planning failure. Overall failure rates were sufficiently low as not to interfere with statistics involving the change in performance over a range of obstacle densities.

B. Results

Figure 4 illustrates the distribution in simulation time required to solve navigation problems at various obstacle densities. By comparison, Fig. 5 depicts the resulting path length. Whereas time continues to increase with elevated density, path length essentially plateaus above 0.5%. This observation suggests that the robot is coping with greater obstacle density by increasingly switching from sidewinding to slithering gaits, as would be expected when squeezing through tighter gaps.

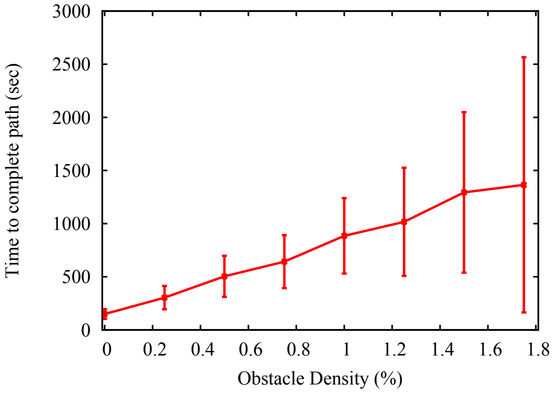


Fig. 4. Average time elapsed to execute one problem run of the simulated snake robot. Elapsed time to plan and execute a path grows linearly with increased obstacle density.

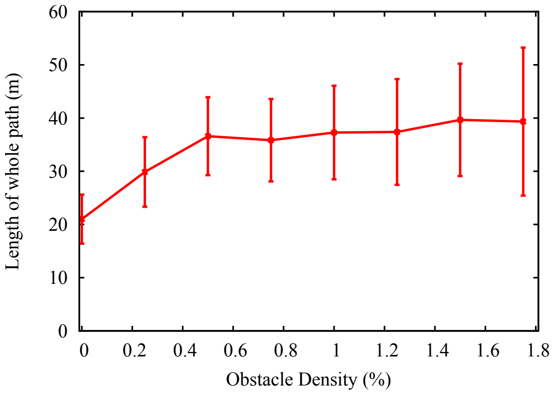


Fig. 5. Average path length to execute one problem run of the simulated snake robot. As obstacle density increases, the planner initially finds that sidewinding to circumnavigate obstacles remains most efficient. Above 0.5% obstacle density, however, the planner increasingly switches to slithering mode to squeeze through tight gaps, thus avoiding more circuitous traversals.

Figure 6 illustrates the effective overall speed of the robot as it executes a path between the start and goal states. Speeds are shown in comparison to steady-state sidewinding and slithering gaits as measured within the ODE simulation. Even in an obstacle free environment, average speed drops below what would be expected from a pure sidewinding solution. This effect results from inefficiencies in executing the transitions between gaits. As transitions were not a focus of this work, we believe that overall performance of gait-based planning can be expected to improve in future work.

V. EXPERIMENTS

We performed some initial experiments on the snake robots in our labs. We sidestepped the critical challenge of localization by either assuming a person is in the loop or using an overhead camera. Snake robot localization is a topic

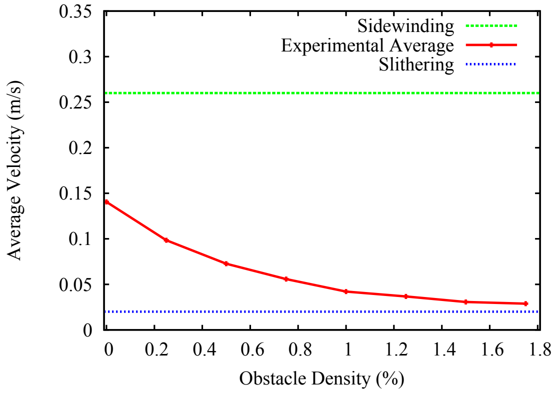


Fig. 6. Average robot velocity while executing problem runs of the simulated snake robot. Idealized steady-state sidewinding and slithering are shown for comparison. Gait transitions currently impose a time penalty on the overall velocity of a planned path.

for future research and is beyond the scope of this paper.

A. Path Following with Conical Sidewinding

Sidewinding is an efficient translational gait with wide adaptivity to various terrains. As well as linear translation, our prior work developed a "steering form" of sidewinding [46]. With this in-hand, we can create a sidewinding library whose component motions each correspond to a constant curvature path in the position space, as if the snake robot were a wheeled vehicle with Ackerman steering.

We conducted an experiment outdoors (Fig. 7) where a person served as the planner to select the appropriate gait to follow a desired path. The reason for having the person serve as the planner was mainly to close the loop; future work will develop an estimator for the snake robot so it can determine the bounding box's position and orientation with respect to a desired path. With such an estimator in-hand, autonomous planning would then be possible.

This experiment demonstrates the power of the gait and virtual chassis abstractions. A snake robot for which each of the sixteen joints must be individually controlled would be challenging for a human to drive. In contrast, with these two abstractions, a person can steer the robot using only the skills required for a video game, using minimal cognitive load.

B. Snake Robot Coverage

We are able to provide full autonomy of the snake robot in a controlled lab environment using a downward pointing camera from the lab's ceiling. The camera produces images which can be readily processed in real time to provide localization (position and orientation) for the snake robot. The image processing system runs at approximately 1 Hz. The objective of this experiment is to direct the snake robot to cover the boundary (circumnavigate) a target object as shown in Fig. 8.

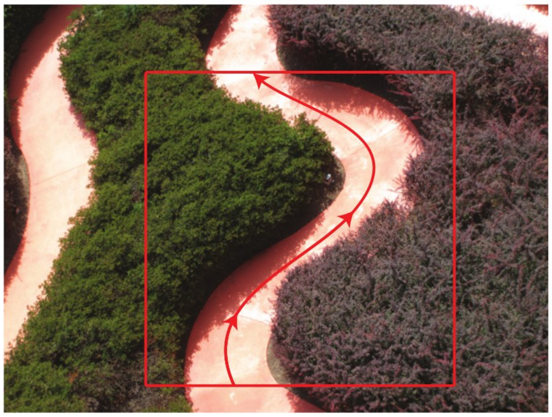
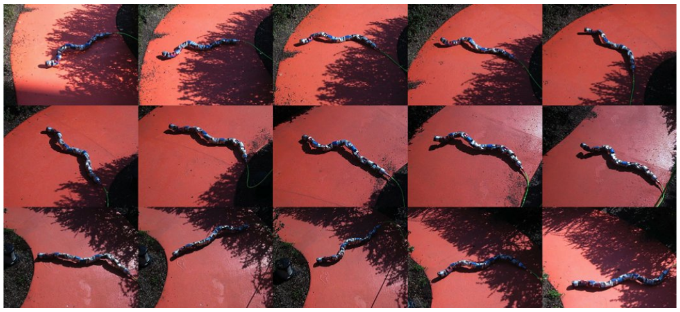



Fig. 7. A montage of a snake robot following a curved ground path using the sidewinding gait. The setting of the experiment (a sidwalk) is shown above.

Feedback from the image processing system (snake and target object positions and orientations) is used by the high level planner to select the appropriate gait from an available library and the appropriate gait parameters. Effectively, sidewinding gaits are executed to generate motions parallel to the edges of the target object. When a corner on the target object is reached, a positive-turning-radius conical sidewinding gait is used to orient the snake parallel with the next edge. The image sequence in Fig. 9 shows the snake robot successfully circumnavigating the boundary of the target object.

VI. CONCLUSION

By combining prior work in gait development and system representation for snake robots with modern vehicle planning algorithms, we have successfully demonstrated an approach to navigating a snake robot through a cluttered environment. The contribution of this paper is not the planner nor the low-level controller, but the "glue" that connects them. Already, we have seen that we can abstract gaits into shape-changing rectangles each associated with a motion that corresponds to a gait. The challenge then becomes: how do we ensure that the gaits fit into the high level planner in a meaningful and useful way. We handle that challenge with the virtual chassis and wheels. We must also guarantee that we have enough gait motions that are sufficient to execute the high-level plan. For single rigid body robots with non-holonomic constraints, this is often an issue of controllability. Here, we generalize this notion to ensure we can indeed follow a path from a high level planner using the existing set of gaits in our library. Many other planning frameworks must generate a costly, high-fidelity plan all the way to the goal.

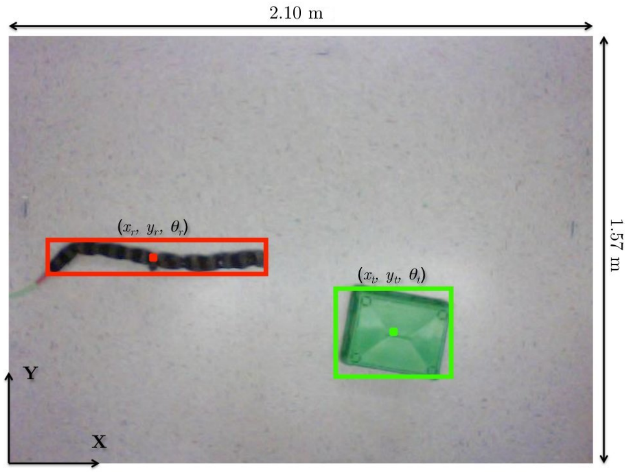
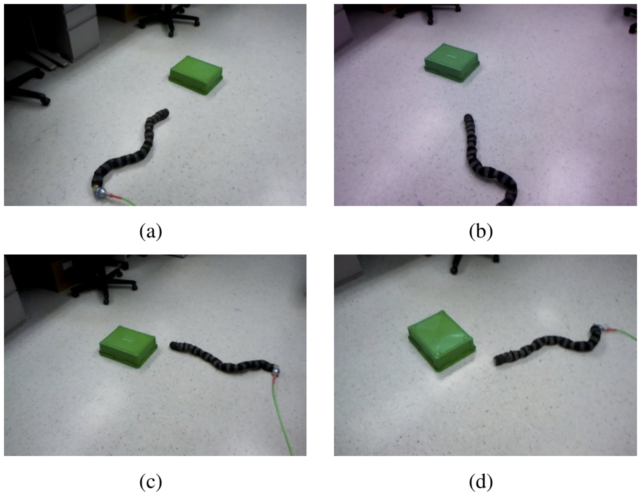


Fig. 8. Detection of the snake robot and the target object (green box).



(a) (b)

(c) (d)

Fig. 9. Snapshot sequence of the circumnavigation of the target object (green box) traced by the snake robot.

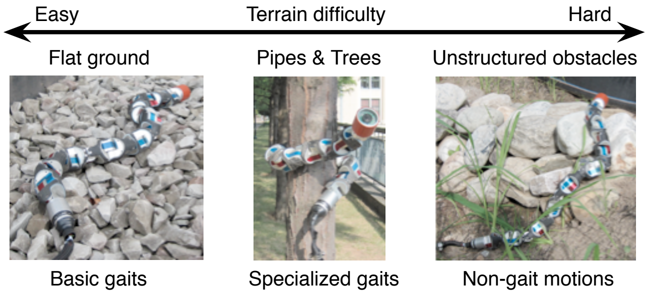


Fig. 10. Terrain complexity influences

By combining the motion concatenation with a low-fidelity global planner, our approach realizes the best aspects of both model-predictive planning and scalability.

The planning model presented in this paper assumes that the terrain is mostly flat ground, punctuated by sparsely or moderately cluttered obstacles. This is not to say that we envision snake robots operating only in such environments, or that we are “sending a snake to do a car’s job.” Rather, it reflects our view that a snake robot should take advantage of vehicle-like motions to move quickly when and where it can, reserving more sophisticated control schemes for situations in which they are necessary.

Further, we feel that the results in this paper are an important first step to using planners to guide snake robots over more challenging terrains, like those in the spectrum in Fig. 10. Field experiments at sites like the Disaster City® training ground at Texas A&M University [47] have suggested that the basic library of gaits we developed for smooth ground traversal is quite robust to changes in terrain. Over moderately-rough surfaces, executing these gaits is *sufficient* to propel the robot in the desired directions, even if the gaits are not theoretically-optimal motions through the environment. Future work will use planners that have real-valued terrain cost and incorporating specialized gaits or non-gait motion controllers as motion primitives, thus extending the best features of high-level planning and low-level control to snake robots operating in such realistic environments, being developed for force-based obstacle exploitation [31].

ACKNOWLEDGMENTS

We thank Matthew Tesch and Matthew Travers for their assistance in integrating the ODE-based Modsnake simulator with our motion planner.

REFERENCES

[1] S. Hirose, *Biologically Inspired Robots (Snake-like Locomotor and Manipulator)*. Oxford University Press, 1993.

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

[3] R. L. Hatton and H. Choset, “Sidewinding on slopes,” in *Proceedings of the IEEE International Conference on Robotics and Automation*, Anchorage, AK USA, May 2010, pp. 691–696.

[4] 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.

[5] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiware, K. Harada, K. Yokoi, and H. Hirukawa, “Biped walking pattern generation by using preview control of zero-moment point,” in *In Proceedings of the International Conference on Robotics and Automation*, Taipei, Taiwan, 2003.

[6] J. J. Kuffner, S. Kagami, K. Nishiwaki, M. Inaba, and H. Inoue, “Dynamically-stable motion planning for humanoid robots,” *Autonomous Robots*, vol. 12, pp. 105–118, 2002.

[7] J. Pettré, J.-P. Laumond, and T. Siméon, “A 2-stages locomotion planner for digital actors,” in *Eurographics/SIGGRAPH Symposium on Computer Animation*, 2003.

[8] M. Lau and J. J. Kuffner, “Precomputed search trees: Planning for interactive goal-driven animation,” in *SIGGRAPH/Eurographics Symposium on Computer Animation*, Sep. 2006, pp. 299–308.

[9] S. Candido, Y. Kim, and S. Hutchinson, “An improved hierarchical motion planner for humanoid robots,” in *Proceedings of the Internaional Conference on Humanoid Robots*, Daejeon, Korea, December 2008.

[10] K. Hauser, T. Bretl, J.-C. Latombe, K. Harada, and B. Wilcox, “Motion planning for legged robots on varied terrain,” *The International Journal of Robotics Research*, vol. 27, pp. 1325–1349, 2008.

[11] A. Shkolnik, M. Levashov, I. R. Manchester, and R. Tedrake, "Bounding on rough terrain with the LittleDog robot," *The International Journal of Robotics Research*, vol. 30, no. 2, pp. 192–215, February 2011.

[12] M. Yim, S. Homans, and K. Roufas, "Climbing with Snake-Like Robots," *IFAC Workshop on Mobile Robot Technology*, 2001.

[13] M. Mori and S. Hirose, "Three-dimensional Serpentine Motion and Lateral Rolling by Active Cord Mechanism ACM-R3," in *IROS*, 2002.

[14] H. Yamada, S. Chigisaki, M. Mori, K. Takita, and S. Hirose, "Development of amphibious snake-like robot acm-r5," in *The Proc. of 36th Int. Symposium on Robotics*, 2005.

[15] W.-M. Shen, M. Krivokon, H. Chiu, J. Everist, M. Rubenstein, and J. Venkatesh, "Multimode locomotion via superbot reconfigurable robots," *Autonomous Robots*, vol. 20, no. 2, pp. 165–177, 2006.

[16] C. Wright, A. Johnson, A. Peck, Z. McCord, A. Naaktgeboren, P. Gianfortoni, M. Gonzalez-Rivero, R. Hatton, and H. Choset, "Design of a Modular Snake Robot," in *Proceedigns of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, San Diego, CA USA, October 2007, pp. 2609–2614.

[17] A. A. Transeth, R. I. Leine, C. Glocker, K. Y. Pettersen, and P. Liljebäck, "Snake robot obstacle-aided locomotion: Modeling, simulations, and experiments," *IEEE Transactions on Robotics*, vol. 24, no. 1, pp. 88–104, February 2008.

[18] 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.

[19] J. Gonzalez-Gomez, H. Zhang, and E. Boemo, "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, 2007.

[20] K. McIsaac and J. P. Ostrowski, "Motion planning for anguilliform locomotion," *Robotics and Automation*, Jan 2003. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1220714

[21] J. Borenstein, M. Hansen, and A. Borrell, "The omnitread ot-4 serpentine robot—design and performance," *Journal of Field Robotics*, vol. 24, no. 7, pp. 601–621, 2007.

[22] 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 *Proceedings of the IEEE International Conference on Robotics and Automation*, Minneapolis, MN, May 2012.

[23] J. Ostrowski and J. Burdick, "The geometric mechanics of undulatory robotic locomotion," *International Journal of Robotics Research*, vol. 17, no. 7, pp. 683–702, 1998.

[24] M. Sfakiotakis and D. Tsakiris, "Biomimetic Centering for Undulatory Robots," *International Journal of Robotics Research*, November/December 2007.

[25] R. L. Hatton and H. Choset, "Geometric motion planning: The local connection, Stokes's theorem, and the importance of coordinate choice," *International Journal of Robotics Research*, vol. 30, no. 8, pp. 988–1014, July 2011.

[26] G. Chirikjian and J. Burdick, "A modal approach to hyper-redundant manipulator kinematics," *IEEE Transactions on Robotics and Autmation*, vol. 10, pp. 343–354, 1994.

[27] S. B. Andersson, "Discretization of a Continuous Curve," *IEEE Transactions on Robotics*, vol. 24, no. 2, pp. 456–461, April 2008.

[28] 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, pp. 271–281, April 2010.

[29] H. Yamada and S. Hirose, "Approximations to continuous curves of active cord mechanism made of arc-shaped joints or double joints," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, may 2010, pp. 703 –708.

[30] A. J. Ijspeert, "Central Pattern Generators for Locomotion Control in Animals and Robotics," *Neural Networks*, vol. 21, pp. 642–653, 2008.

[31] P. Liljebäck, K. Y. Pettersen, Ø. Stavdahl, and J. T. Gravdahl, "Experimental investigation of obstacle-aided locomotion with a snake robot," *IEEE Transactions on Robotics*, vol. 27, no. 4, pp. 792–800, 2011.

[32] K. Lipkin, I. Brown, A. Peck, H. Choset, J. Rembisz, P. Gianfortoni, and A. Naaktgeboren, "Differentiable and Piecewise Differentiable Gaits for Snake Robots," in *Proceedings of IEEE/RSJ Intl. Conference on Intelligent Robots and Systems*, San Diego, CA, USA, Oct 29 - Nov 2 2007, pp. 1864–1869.

[33] G. Chirikjian and J. Burdick, "Kinematics of Hyper-redundant Locomotion with Applications to Grasping," in *International Conference on Robotics and Automation*, 1991.

[34] G. Chirikjian, "Theory and Applications of Hyper-Redundant Robotic Manipulators," Ph.D. dissertation, California Institute of Technology, 1992.

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

[36] I. Tanev, T. Ray, and A. Buller, "Automated evolutionary design, robustness, and adaptation of sidewinding locomotion of a simulated snake-like robot," *Robotics, IEEE Transactions on*, vol. 21, no. 4, pp. 632–645, August 2005.

[37] C. Gong, R. L. Hatton, and H. Choset, "Conical sidewinding," in *IEEE International Conference on Robotics and Autmation*, 2012 (submitted).

[38] S. Koenig and M. Likhachev, "D*lite," in *AAAI/IAAI*, 2002, pp. 476–483.

[39] S. M. LaValle and J. J. K. Jr., "Randomized kinodynamic planning," in *IEEE International Conference on Robotics and Automation*, vol. 1, 1999, pp. 473–479.

[40] R. A. Knepper, S. S. Srinivasa, and M. T. Mason, "Hierarchical planning architectures for mobile manipulation tasks in indoor environments," in *Proceedings of International Conference on Robotics and Automation*, May 2010.

[41] P. Hart, N. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Transactions of Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, July 1968.

[42] A. M. Bloch *et al.*, *Nonholonomic Mechanics and Control*. Springer, 2003.

[43] R. L. Hatton, L. J. Burton, A. E. Hosoi, and H. Choset, "Geometric Maneuverability, with Applications to Low Reynolds Number Swimming," in *Proceedigns of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, San Francisco, CA USA, September 2011.

[44] T. Yoshikawa, "Manipulability of Robotic Mechanisms," *The International Journal of Robotics Research*, vol. 4, no. 3, 1985.

[45] 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*, no. 23, pp. 1131–1158, 2009.

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

[47] [Online]. Available: http://www.teex.com/teex.cfm?pageid=USARprog&area=USAR&templateid=1117