2015 IEEE International Conference on Robotics and Automation (ICRA)
Washington State Convention Center
Seattle, Washington, May 26-30, 2015

Locomotive Reduction for Snake Robots

Xuesu Xiao[1], Ellen Cappo[1], Weikun Zhen[1], Jin Dai[1], Ke Sun[1], Chaohui Gong[1],
Matthew J. Travers[2] and Howie Choset[2]

Abstract— **Limbless locomotion, evidenced by both biological and robotic snakes, capitalizes on these systems' redundant degrees of freedom to negotiate complicated environments. While the versatility of locomotion methods provided by a snake-like form is of great advantage, the difficulties in both representing the high dimensional workspace configuration and implementing the desired translations and orientations makes difficult further development of autonomous behaviors for snake robots. Based on a previously defined average body frame and set of motion primitives, this work proposes locomotive reduction, a simplifying methodology which reduces the complexity of controlling a redundant snake robot to that of navigating a differential-drive vehicle. We verify this technique by controlling a 16-DOF snake robot using locomotive reduction combined with a visual tracking system. The simplicity resulting from the proposed locomotive reduction method allows users to apply established autonomous navigation techniques previously developed for differential-drive cars to snake robots. Best of all, locomotive reduction preserves the advantages of a snake robot's ability to perform a variety of locomotion modes when facing complicated mobility challenges.**

I. Introduction

Snakes use various modes of locomotion to overcome challenging terrain, often allowing them to traverse environments inaccessible to conventional wheeled vehicles. We believe that snake robots modeled on biological snakes will possess a similar capability, enabling them to be used in applications involving complicated environments such as urban search and rescue. However, planning and control for a redundant locomotor is difficult because one must coordinate all of the internal degrees of freedom of the robot to simultaneously provide locomotive benefit and direct the robot in a desired direction. Our approach uses a variety of motion primitives (gaits) [1] [2] which we control online to direct the robot's motion. A subtle problem associated with defining primitives for limbless locomotors has to do with assigning the mechanism a body frame. Put more succinctly, it is not clear which way is forward. Fortunately, our prior work has defined a body frame called the virtual chassis which provides an intuitive coordinate frame [3], addressing the question of which way is forward. The main contribution of this paper is the prescription of the control law that uses the motion primitives and virtual chassis frame to follow a trajectory in the robot's workspace.

The virtual chassis in combination with a set of motion primitives allows us to control the motion of a redundant snake robot as if it were a differential-drive car, and we term this simplification *locomotive reduction*. As a result, planning and control algorithms previously developed for differential-drive cars can be directly employed on snake robots. In addition to the simplicity in control resulting from locomotive reduction, our method does not compromise the versatility of a redundant locomotor: whenever needed, the robot is still able to transition from the "differential-drive mode" back to any specialized snake behavior in order to engage in more complicated tasks which cannot be achieved by conventional wheeled systems (Fig. 1).

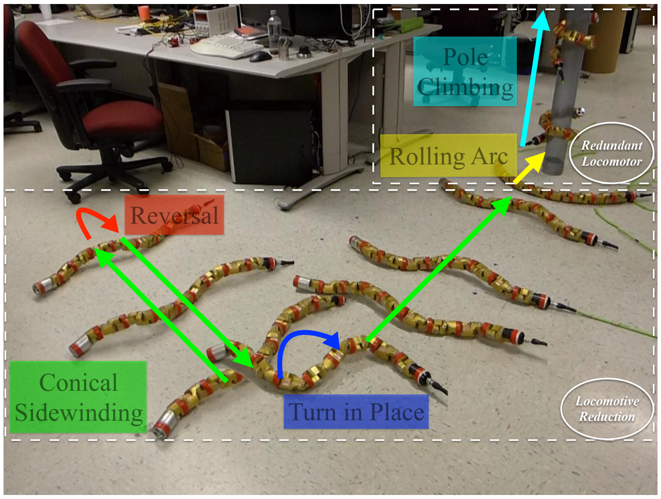

Fig. 1. A snake robot uses locomotive reduction to autonomously navigate to a pole, at which point it employs redundant locomotion modes to climb the pole and execute its task.

In this work, we demonstrate locomotive reduction by autonomously navigating a 16 degree of freedom snake robot as if controlling a differential-drive car. We experimentally validate our control method by showing autonomous navigation through a cluttered environment using visual feedback from overhead cameras, and show that the locomotive reduction method can be integrated with existing modes of redundant locomotion. Our result demonstrates that locomotive reduction is a tractable and intuitive technique which largely simplifies the motion control problem of hyper redundant snake robots, while additionally enabling operators to use established autonomous navigation techniques developed for use with traditional differential-drive cars. Further, the ability to autonomously follow a given path frees operators from

[1] Xuesu Xiao, Ellen Cappo, Weikun Zhen, Jin Dai, Ke Sun, and Chaohui Gong are students at the Robotics Institute, Carnegie Mellon University, Pittsburgh PA, 15213, USA `xuesux, ecappo, weikunz, jind, kesun, chaohuig@andrew.cmu.edu`

[2] Matthew J. Travers and Howie Choset are a project scientist and a professor at the Robotics Institute, Carnegie Mellon University, Pittsburgh PA, 15213, USA `mtravers@andrew.cmu.edu`, `choset@cmu.edu`

978-1-4799-6923-4/15/$31.00 ©2015 IEEE

tedious joint level control of the robot, allowing them to concentrate on task-level needs.

While previous work has looked at path following for snake robots [4], this work performs trajectory following by switching between pre-computed motions. The locomotive reduction technique allows for online control by directly modifying the gait in a continuous manner. This allows the robot to move in any direction as needed, rather than having to choose from a library of precomputed, discrete, motions.

This paper is divided into the following sections. In Section II, the background information of a modular snake robot and its related locomotion principles are presented. In Section III, the mapping from a hyper redundant snake robot to a simple differential-drive model is explained, and the high level line of sight and gait parameter controls are derived. In Section IV, we explain our experimental setup, and then present the results on a modular snake robot in Section V.

II. Background

A. Modular Snake Robot

Modular snake robots [5] are highly redundant mechanisms which are able to perform a variety of locomotion methods [6] [7] [8]. Much of biological snake behavior can be effectively mimicked by modular snake robots, such as lateral sidewinding [1], forward slithering [9], climbing [10], etc. The CMU modular snake robot used in this work has 16 degrees of freedom, alternating vertically and horizontally along its body [11], as seen in Figure 2. The joints of the snake are indexed from 1, starting from the head. Even-indexed joints control lateral rotation, while odd-indexed joints control dorsal rotation.

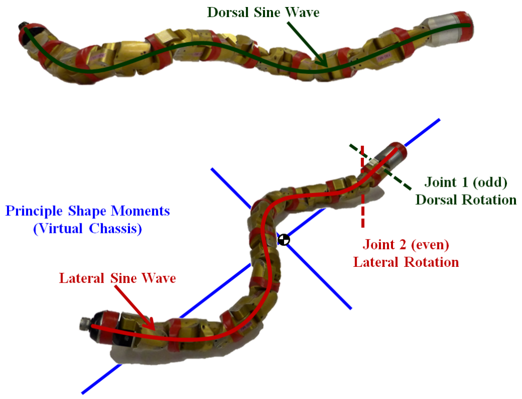

Fig. 2. The joints of the robot alternate axes of rotation about the lateral and dorsal directions, and are indexed from 1 starting with the joint closest to the head. In sidewinding, the lateral sine wave is controlled by the even-indexed joints, while the dorsal wave is controlled by odd-indexed joints. The virtual chassis is calculated by identifying the principle moments of shape, shown in blue.

B. Gaits

The locomotion of snake robots is typically controlled by gaits, cyclical controls that coordinate a system's internal degrees of freedom to produce net locomotion. Different motions are generated by executing different gaits. For example, some gaits produce linear displacement while others rotate the robot in the world frame.

In order to accomplish the desired net locomotion based on corresponding gaits, a gait equation is commonly used as in [12] and as shown below:

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

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

In equation 1, α describes the values of every joint angle based on joint index, n, and time, t. Further, β, A, θ, δ, Ω and ω are, respectively, offset, amplitude, frequency, phase shift, spatial frequency and temporal frequency. Joint movements can be compactly implemented by using this gait equation as the control signal specifying joint inputs [12] [13]. The even joints specify the lateral wave while the odd joints specify the vertical wave. The parameterized representation provided by a gait equation is smooth, so the motion properties can be adjusted by continuously varying gait parameters [2].

One of the most effective locomotive gaits for a snake robot is sidewinding, in which the snake moves laterally [6] [12]. Sidewinding is a fast and efficient gait, but most importantly, the heading of the sidewinding motion can be easily steered [2].

We are also able to effect two special cases of sidewinding, both of which are used in the proposed locomotive reduction in this work. Turn in place is a special case in sidewinding. By setting $\Omega_{odd} = 0.6\Omega_{even}$ (obtained by experimentation for our robot), the robot effectively performs a point turn by pivoting about its center. This gait is extremely useful when the robot is required to make tight turn in confined space. Reversal is achieved by increasing the phase shift δ by π, which offsets the dorsal sine wave by half a period, reversing the direction of travel of the robot.

We focus in this work on conical sidewinding [1] and this gait forms the basis for our locomotive reduction technique. Conical sidewinding is realized by substituting Aeven of the lateral wave in Eqn. 1 by a linear function in terms of joint index n.

$$A_{even} = kn + b \quad (3)$$

This linear equation controls the taper of a ”cone”, i.e., the taper of the snake's shape from head to tail. In Eqn. 3, k controls the degree of taper of the cone and b is an offset scaling the amplitude. The effect of these parameter choices cause the lateral wave (generated by the even-indexed joints) to be larger in magnitude at one end of the robot than the other, causing it to perform a turning motion. We denote the radius about which the robot turns as R, and can form a linear relationship between k and the reciprocal of the turning radius based on work shown in [2]. This is expressed as:

$$\frac{1}{R} = ak, \quad (4)$$

where the coefficient a can be calibrated for different snakes with different skins.

C. Virtual Chasis

Because of the complex way in which snake robots use internal shape changes to locomote and interact with the world, it is difficult to represent the motion of the snake robot in a consistent local frame. Lack of a consistent local frame makes it difficult to control the high level behavior of the robot. Therefore, we use an average body frame, called the ”virtual chassis” [3] to intuitively represent the configuration of the entire robot. The origin of the virtual chassis is located at the center of geometry (COG) of the robot and the axes are aligned with the robot's principal moments of shape. This yields a body frame which maintains consistency with the overall shape of the robot over all configurations. This also effectively separates the robot's unintuitive internal motions from the robot's external motions produced by interacting with the world. By establishing a consistent local body frame over the robot's internal motions, we are able to view the snake robot as a chassis of a differential-drive car.

D. Differential-drive Robot

The goal of this work is to make controlling high-DOF snake robots as simple as controlling a differential-drive car. The benefits of reducing a snake robot to a differential-drive vehicle are mainly twofold. First, the controllability of a differential-drive car has been proved, and a system resemble a differential-drive vehicle inherit these properties. Second, path and motion planning for differential-drive cars have been extensively studied [14] [15] [16], and these techinques can be directly employed in controlling a snake robot.

In this work, we establish a relationship between the modular snake robot and the differential-drive model in order to reduce the motion planning complexity for sidewinding, so that the snake can navigate autonomously in a simpler manner as shown in Fig. 3. The translational and rotational velocities of differential-drive systems can be expressed as

$$v = \frac{v_{left} + v_{right}}{2} \quad (5)$$

$$\omega = \frac{v}{R} \quad (6)$$

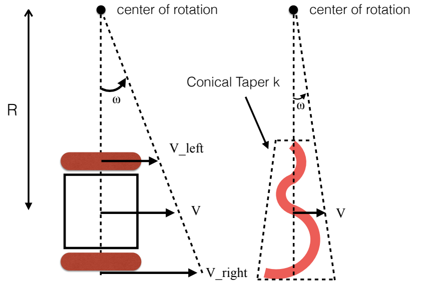

Fig. 3. The aim of locomotive reduction is to simplify the 16-DOF snake robot to a two-wheeled differential-drive vehicle with only three DOFs.

where R is the turning radius of the car, v and $omega$ are the linear and angular velocities, and v_{left} and v_{right} represent the linear velocities of the left and right wheels. In the following section, we establish a relationship between the parameters of the conical sidewinding gait and the model of the differential-drive car, allowing the easy forward, reverse, and turning control of the snake robot.

III. Control and Locomotive Reduction

Our primary interest is to simplify the control of a high degrees of freedom snake robot by means of locomotive reduction. Built upon the previously established gait, conical sidewinding, we show steering the motions of a snake robot is simply achieved by regulating only one parameter (corresponding to the steering angle of a differential-drive car). As a result of locomotive reduction, we further show path following control methods previously developed for differential-drive vehicles can be readily employed to navigate a snake robot.

We set up the problem of controlling the snake robot as a tiered structure. We first lay out the path the robot should follow using a line of sight control algorithm. Next, we explain how we steer the robot to follow the given path. Finally, we show how locomotive reduction is accomplished by substituting the steering control law into the conical sidewinding gait equation, reducing the snake robot system to an approximation of the simpler, differential-drive car model.

A. Line of Sight Control

Line of sight control is a well-developed navigation algorithm and is easy to implement online. Prior work has examined stability [17], implementation [18], and control laws for this technique [19]. A variety of conventional mobile systems [20] [21] and underactuated [22] systems are controlled based on this technique, and we apply it in this work due to the ease of implementation and control.

The planned path for the snake robot is an ordered set of target points. At each time step, P_{k-1} designates the previous waypoint visited by the robot while P_k denotes the next target waypoint, as shown in Fig. 4. A circle of radius L, centered at the COG of the snake, intersects the straight line segment between P_{k-1} and P_k at two points. Between these two intersection points, we designate the point closer to the target point P_k as P_{los}. When the COG of the snake reaches within a circle of acceptance of radius R_k around the target point P_k, the target point is updated to the next defined point P_{k+1}. The value for R_k is application dependent, and defines the tolerance with which the robot adheres to the target path. We require $L > R_k$ so that the circle enclosing the snake has a larger radius than the radius of the circle of acceptance; this ensures that the snake will always have a valid current target point.

B. Heading Direction Control

After defining the high level problem of line of sight path following, we control the heading of the snake robot to move the robot towards the line of sight point P_{los}. We assume the linear velocity while sidewinding is constant and base our heading control on the angular velocity of the robot. The

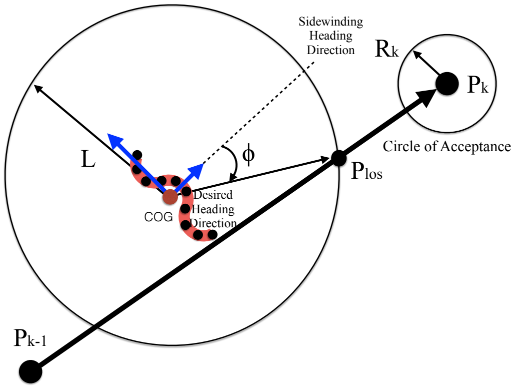

Fig. 4. Line-of-sight line following principle in a single path interval from P_{k-1} to P_k. Also showing the instantaneous targeting point, P_{los}.

asymptotic convergence of the line-of-sight control approach allows us to use a simple proportional control law to control the robot along a predefined path. Our reference signal is the angle between our desired and actual heading, ϕ, as shown in Fig. 4. In an ideal situation, the sidewinding heading direction should point to the current line of sight point P_{los}, meaning that our desired heading, ϕ_d, is zero. Proportional control about the error in ϕ is therefore

$$\omega = K_p(\phi_d - \phi_{measured}) = -K_p\phi_{measured}, \qquad (7)$$

where K_p is the proportional gain, ω is the angular velocity of the snake robot, and ϕ measured is the actually measured heading error.

C. Gait Parameter Control for Locomotive Reduction

The goal of gait parameter control in locomotive reduction is to steer the snake robot towards a desired position by changing a reduced number of gait parameters. In this work, we use only one parameter, the amplitude gradient k, as mentioned in Eqn. 4 to control the sidewinding motion of the snake robot . As shown in Eqn. 4, a larger k corresponds to a smaller turning radius. If we consider a constant linear velocity v, then Eqn. 6 gives us a larger angular velocity ω. Since k is proportional to ω, we simply let

$$k = \omega = -K_p\phi_{measured}. \qquad (8)$$

The proportional term a from Eqn. 4 can be ignored if desired, since its effect can be included when tuning the proportional gain Kp in Eqn. 7. Therefore, Eqn. 8 reduces the complexity of snake locomotion to a single amplitude gradient, k, and allows us to apply our heading direction control law directly into snake robot locomotion control.

Differential-drive cars control their linear velocity by averaging the velocity of the left and right wheels. A difference in wheel velocities causes the robot to turn. For our snake robot, as shown back in Figure 3, we control the angular velocity of the snake by adjusting the conical taper, k. Tuning k according to our proportional control law allows us to achieve a relative difference between the head and tail of the snake just as changing the difference between wheel speeds controls the angular velocity of the car. The relation established in Eqn. 8 indicates controlling a snake robot is at most as difficult as driving a differential-drive vehicle.

IV. System Implementation

To show that locomotive reduction allows for simpler control of a snake robot and leads to improved autonomous path following, we illustrate locomotive reduction control on the CMU snake robot. Using a finite state machine (Fig. 5), the snake can seamlessly transition in two different regimes, either moving as a differential-drive car or utilizing the specialized behaviors unique to a snake robot. We show we can successfully autonomously follow a given path using locomotive reduction, and further, that the locomotive reduction control method easily integrates with existing hyper-redundant locomotion modes for extended capability.

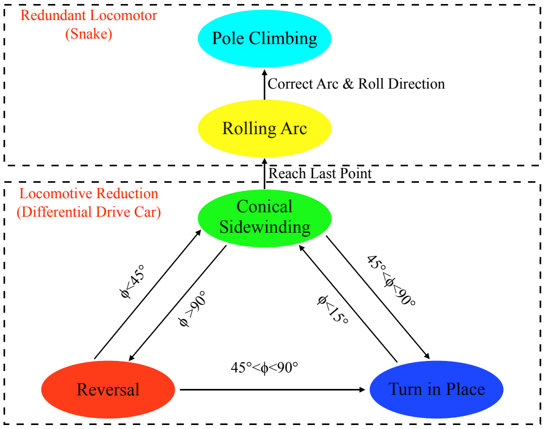

Fig. 5. Finite state transition machine of different gaits and transition conditions based on ϕ, heading angle error

Here, we explain our experimental setup and over-all control system. We illustrate the effectiveness of our approach by making the snake move through a mock urban search and rescue situation, which requires path generation, autonomous path following, switching between different locomotive reduction modes, and transitioning from the locomotive reduction mode to the hyper redundant gait of the rolling arc in order to climb a pole for increased surveillance capability.

A. Camera Cooperation and Map Generation

In order for the snake to navigate through an area, we must first be given (or generate) a map of the area of interest. In spaces such as buildings, surveillance cameras may exist and provide useful vision feedback for snake robot navigation. Therefore, we use two ceiling-mounted webcams in this work to present the generality of a multiple-camera vision feedback system. We transform the second camera into the same coordinate system as the first camera in order to get a uniform metric coordinate system. Using the combined fields of view, we take this top-down view of the operating area and segment the obstacles from the clear ground area. We further mark areas outside of the cameras range as ”obstacles” in order to force the robot to stay within camera range.

B. Path Planning

Because the locomotive reduction method makes use of the turn-in-place gait, we model the snake robot as a circle with a diameter equal to its body length. Using the generated map and the simplified heuristic modeling the robot as a circle capable of travel in any direction, an operator may use any path planning method of choice to generate a feasible trajectory for the snake robot. Optimal planners can easily be run over this space to return paths of shortest distance or which avoid specific areas. In our experiment however, we hand-picked path waypoints which forced the robot to perform non-optimal actions such as reversing and turning in order to illustrate the capabilities of locomotive reduction.

C. Vision Tracking

In order to calculate the error between the robot’s position and heading and the desired path, we use the cameras described above to provide visual feedback of the robot’s position and shape. We perform vision tracking by attaching 10 orange blocks evenly along the snake robot’s length to serve as fiducial markers. The visual tracker returns the coordinates of these markers, and these coordinates are then used to produce the estimate of COG location and heading direction using the virtual chassis described in Sec. II.

D. State Transitions

When executing path following and autonomous navigation, the snake is in ”differential-drive mode”, using locomotive reduction. After initialization, the robot is therefore in one of the three gait states: conical sidewinding, turn in place, or reversal. The transitions between gaits depend on the error in heading angle, ϕ. The state machine diagram describing the transitions between gaits is shown in Fig. 5.

In sidewinding, when the error angle ϕ is between 45 and 90 degrees, the gait state is switched to turn in place until the error angle is less than 15 degrees. Then the state is switched back to conical sidewinding and the snake resumes approaching the current target point. When the current target point is behind the snake’s heading direction, namely the error angle is greater than 90 degrees, ”reversal” will switch the sign of the snake’s orientation and sidewinding parameters. The heading direction is flipped so that the error angle is between 0 and 90 degrees. The conical sidewinding or turn in place gait will be selected if the new error angle is either less than or greater than 45 degrees, respectively. The snake will then move backwards. This is repeatable should the next target appear behind the new heading direction.

When the snake reaches the target location designated here as the base of the pole, the snake transitions to a redundant locomotor and forms a rolling arc to approach the pole. The rolling arc controller will correct the rolling direction and arc opening in order to guarantee that the snake robot is approaching the pole in a suitable way to climb. Finally, the robot transitions to the pole climbing gait and climbs the pole by winding onto it and climbing upward.

V. Results

We demonstrate the efficacy of the proposed framework by showing a robot autonomously navigating in a test environment. This test was designed to emphasize 1) the simplicity, resulted from locomotive reduction, in steering a snake robot, 2) the versatility of a snake robot and 3) most importantly, the efficacy of the proposed ”finite state machine” in managing the versatile gaits of a snake robot.

In this section we present the results of using the locomotive reduction technique to autonomously follow a given string of waypoints. As mentioned, these waypoints were chosen to force the robot to transition between each state of the control space. We show the locomotive reduction technique executing forward locomotion, reversing direction, and turning. We also show the transition between locomotive reduction to the rolling arc and pole climbing gaits – gaits performed by the robot as a hyper-redundant mechanism – in order to navigate to and climb a pole for increased surveillance.

Fig. 6 shows time-lapse shots of the robot as it follows the desired path. The information from the visual feedback system is overlayed in order to show the robot’s position relative to the desired path. The upper row shows the snake position (the fiducial marks highlighted in blue) in the overall map, while the lower row shows information from the line of sight controller overlaid on the image.

In the first column, the snake begins following the path using locomotive reduction, using sidewinding from its initial starting position. The spike in the first red region represents a ϕ angle nearly 180 degrees. It is immediately corrected by reversal. The second column shows the reversal state where the heading direction (blue arrow) is reversed. The second spike is a smaller ϕ angle than the first one, between 90 and 135 degrees, but still causes the robot to reverse. It is corrected by reversal first, leading to a new ϕ angle between 45 and 90 degrees (blue region). So in the third column, the snake robot chooses to executing the turn in place gait in order to further decrease ϕ and navigate the right angle turn. In column 4, the snake continues sidewinding along the path in the second camera view until reaching the final target point. At this point, the locomotive reduction mode ends and the snake transitions to a redundant locomotor. The snake then uses a rolling arc gait to approach the pole, shown in the last column.

VI. Conclusions

In conclusion, we have proposed a locomotive reduction technique which effectively reduced the complexity in planning motions for a redundant locomotor. The previously complicated navigation of the redundant mechanism is now as simple as controlling a differential-drive vehicle. Our robot demonstration empirically verified the efficacy of the concept of locomotive reduction. Built upon ”locomotive reduction”, we introduce a ”finite state machine” to manage the versatile motions of a snake robot. This ”finite state machine” balances between the complexity in planning and locomotive capability of a snake robot. In ”differential driving” mode, a

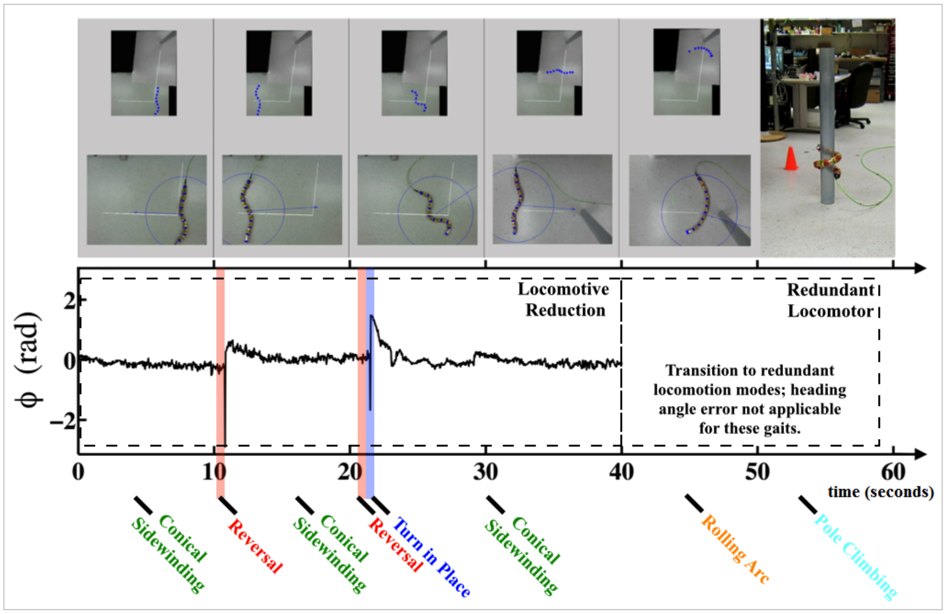

Fig. 6. Time lapse of the visual tracking system to execute the locomotive reduction for autonomous navigation

snake robot can be controlled with minimal effort. Whenever switched to the hyper-redundant mode, a snake robot remains its insurmountable locomotive capability. The work presented in this paper therefore serves as further motivation for future research on automation methodologies for redundant mechanisms based on the idea of locomotive reduction.

References

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

[2] C. Gong, M. J. Travers, X. Fu, and H. Choset, "Extended gait equation for sidewinding," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*. IEEE, 2013, pp. 5162–5167.

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

[4] R. L. Hatton, R. A. Knepper, H. Choset, D. Rollinson, C. Gong, and E. Galceran, "Snakes on a plan: Toward combining planning and control," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*. IEEE, 2013, pp. 5174–5181.

[5] 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 *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*. IEEE, 2007, pp. 2609–2614.

[6] S. Hirose and M. Mori, "Biologically inspired snake-like robots," in *Robotics and Biomimetics, 2004. ROBIO 2004. IEEE International Conference on*. IEEE, 2004, pp. 1–7.

[7] Z. Y. Bayraktaroglu, A. Kilicarslan, and A. Kuzucu, "Design and control of biologically inspired wheel-less snake-like robot," in *Biomedical Robotics and Biomechatronics, 2006. BioRob 2006. The First IEEE/RAS-EMBS International Conference on*. IEEE, 2006, pp. 1001–1006.

[8] S. Hirose and E. F. Fukushima, "Snakes and strings: new robotic components for rescue operations," *The International Journal of Robotics Research*, vol. 23, no. 4-5, pp. 341–349, 2004.

[9] D. L. Hu, J. Nirody, T. Scott, and M. J. Shelley, "The mechanics of slithering locomotion," *Proceedings of the National Academy of Sciences*, vol. 106, no. 25, pp. 10 081–10 085, 2009.

[10] M. Nilsson, "Snake robot free climbing," in *Robotics and Automation, 1997. Proceedings., 1997 IEEE International Conference on*, vol. 4. IEEE, 1997, pp. 3415–3420.

[11] 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*. IEEE, 2012, pp. 4347–4354.

[12] 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.

[13] J. Gonzalez-Gomez, H. Zhang, and E. Boemo, *Locomotion principles of 1D topology pitch and pitch-yaw-connecting modular robots*. na, 2007.

[14] S. Bhattacharya, R. Murrieta-Cid, and S. Hutchinson, "Optimal paths for landmark-based navigation by differential-drive vehicles with field-of-view constraints," *Robotics, IEEE Transactions on*, vol. 23, no. 1, pp. 47–59, 2007.

[15] D. J. Balkcom and M. T. Mason, "Time optimal trajectories for bounded velocity differential drive vehicles," *The International Journal of Robotics Research*, vol. 21, no. 3, pp. 199–217, 2002.

[16] ——, "Extremal trajectories for bounded velocity differential drive robots," in *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on*, vol. 3. IEEE, 2000, pp. 2479–2484.

[17] P. J. Kennedy and R. L. Kennedy, "Direct versus indirect line of sight (los) stabilization," *Control Systems Technology, IEEE Transactions on*, vol. 11, no. 1, pp. 3–15, 2003.

[18] J. Waldmann, "Line-of-sight rate estimation and linearizing control of an imaging seeker in a tactical missile guided by proportional navigation," *Control Systems Technology, IEEE Transactions on*, vol. 10, no. 4, pp. 556–567, 2002.

[19] K. Lim and G. Balas, "Line-of-sight control of the csi evolutionary model: μ control," in *American Control Conference, 1992*. IEEE, 1992, pp. 1996–2000.

[20] K. C. Tan, T. H. Lee, E. F. Khor, and D. Ang, "Design and real-time implementation of a multivariable gyro-mirror line-of-sight stabilization platform," *Fuzzy Sets and Systems*, vol. 128, no. 1, pp. 81–93, 2002.

[21] H. Yoon and P. Tsiotras, "Spacecraft line-of-sight control using a single variable-speed control moment gyro," *Journal of guidance, control, and dynamics*, vol. 29, no. 6, pp. 1295–1308, 2006.

[22] T. I. Fossen, M. Breivik, and R. Skjetne, "Line-of-sight path following of underactuated marine craft," *Proceedings of the 6th IFAC MCMC, Girona, Spain*, pp. 244–249, 2003.