2016 IEEE 55th Conference on Decision and Control (CDC)
ARIA Resort & Casino
December 12-14, 2016, Las Vegas, USA

Shape-based compliant control with variable coordination centralization on a snake robot

Julian Whitman, Francesco Ruscelli, Matthew Travers and Howie Choset

Abstract— **For highly articulated robots, there is a trade-off between the capability to navigate complex unstructured environments and the high computational cost of coordinating many degrees-of-freedom. In this work, an approach that we refer to as *shape-based control* helps to balance this trade-off using *shape functions*, geometric abstractions that determine the coupling between multiple degrees-of-freedom during locomotion. This approach provides a way to intuitively adapt the shape of highly articulated robots using joint-level torque feedback control, allowing a robot to compliantly feel its way through unstructured terrain. In this work we specifically focus on compliance in the spatial frequency and temporal phase parameters of a snake-like robot's wave-like periodic wave-like kinematics. We show how varying the spatial frequency within the shape-based control architecture allows a single controller to vary the degree to which different degrees-of-freedom are coupled throughout a mechanism's body, i.e., the controller's degree of centralization. We experimentally find that for a snake-like robot locomoting through an irregularly spaced peg array, shape-based control results in more effective locomotion when compared to a central pattern generator-based approach.**

I. Introduction

Snake robots, with their many degrees-of-freedom, have the potential to maneuver through unstructured terrain and confined spaces, making them attractive for tasks like urban search and rescue and inspection [1]. However, coordinating the motion of their internal degrees of freedom to react to unknown features in real-time is challenging. *Shape-based control* is an intuitive and computationally simple way to balance this trade-off, connecting high-level planning to low-level control. Using this approach, the control is executed in a lower dimensional space through *shape functions*, geometric abstractions which connect joint motion to desired motions in the world [2].

Our prior work showed some of the benefits of shape-based control [3]:

- a reduced number of controlled parameters compared to the number of joints, intuitive behavior specification,
- clear methods to use feedback to drive the system's behavior,
- predictable responses to perturbations,
- a framework for propagating information along the robot

Prior work also allowed sections of the robot to conform locally to environment features. However, during run-time the number of degrees-of-freedom coupled in each section, and therefore the level of control centralization, was a fixed design variable which limited the range of obstacles each section could conform to.

Carnegie Mellon University, 5000 Forbes Ave, Pittsburgh, PA 15213, USA, jwhitman@cmu.edu

This limitation occurred because the choice of controller for a robot, including the level of control centralization, influences locomotive efficacy in a particular environment [4]. In robotics there are two extremes of centralization in joint coordination: fully centralized, where a single controller coordinates all of the degrees-of-freedom in the system, and fully decentralized, where each degree-of-freedom is controlled independently. Between these extremes lies a spectrum in which the greater the level of centralization, the more each joint is influenced by the motion of the other joints in the system. The degree to which a controller is centralized is typically a fixed property in the controller specification.

Fully centralized control may be effective in a uniform environment, but can become computationally expensive in irregular environments [5]. The approach we present in this work implements a controller that is nominally in between the fully centralized and fully decentralized examples. This design is motivated by the desire to allow robot to traverse a variety of non-uniform environments without manual adjustment of the control strategy or heavy computation. Specifically, we provide a new means to comply a series-elastic actuated snake robot's gait in both a shape-based as well as a central pattern generator (CPG) framework developed to provide a basis of comparison. We include experimental results that compare the two frameworks, and show that shape-based control outperforms the CPG approach augmented with comparable sensing and compliance characteristics. We find that varying control centralization in the spatial parameters, amplitude and spatial frequency, results in smoother locomotion.

II. Background

Shape-based control is related to existing control methods that are reviewed in this section.

A. Trajectory generation with the serpenoid curve

The serpenoid equation, found throughout the snake-robot literature, is defined as [1], [6],

$$\theta_i = \kappa + A\sin(\omega_s s_i - \omega_T t) \tag{1}$$

where θ_i is the angle of joint i, κ is a constant offset, A is amplitude, ω_T is temporal frequency, and t is time. The spatial frequency ω_s determines the number of waves on the snake robot's body, and $s_i = i\delta s$ is the distance along the robot backbone from the head to joint i, with distance between joints δs. The parameters in the serpenoid curve can be changed to create varying gaits and behaviors [7].

978-1-5090-1836-9/16/$31.00 ©2016 IEEE

B. Curvature-derivative control and Model-Based methods

Curvature-derivative control is a reactive strategy for planar snake robots navigating obstacle-rich environments. Joint torque commands are generated based on the differences between neighboring joint angles [8]. Kano and Ishiguro [9] note that this control method may fail unless supplemented with local reflexes from contact sensors.

Existing model-based control methods for obstacle-aided snake robot locomotion rely on accurate modelling and sensing assumptions. The number, convexity, and detection of robot-obstacle collision points is assumed, and simulations rely on friction models [10], [11]. These assumptions are limiting in unstructured terrains where constraints may be both unsensed and difficult to model.

C. Central pattern generators

A central pattern generator (CPG) is a system of coupled oscillators that generate rhythmic open-loop desired joint trajectories [12]. A system defining a CPG for snake-like robots based on the work in [13], using one oscillator per joint, is:

$$\dot{\phi}_i = \omega_T + \sum_j w \sin(\phi_j - \phi_i - \epsilon_{ij}) \tag{2}$$

$$\ddot{A}_i = a_i\big(\frac{a_i}{4}(A_0 - A_i) - \dot{A}_i\big) \tag{3}$$

$$\theta_i = A_i \cos(\phi_i) \tag{4}$$

Equation (2) governs the evolution of the oscillator phase ϕ for each joint i. The phase of each oscillator evolves at a rate ω_T corresponding to the steady state wave temporal frequency, and synchronizes with neighboring oscillators with weight w governing the speed of convergence. A constant phase offset ϵ determines the steady state phase offset between neighboring oscillators and ultimately the spatial frequency of the steady state response. Equation 3 ensures that the amplitude A converges to a nominal value A_0. Equation 4 is the rhythmic joint output signal, which at steady state converges to a serpenoid curve.

Existing work has integrated feedback into CPGs by adding reflex reactions to Cartesian trajectory parameters [14], or by optimizing weights in a computationally expensive neural net that converts sensor readings into feedback terms in simulation [15]. CPG-based control architectures range from centralized to distributed, at a level typically fixed in the controller specification [4].

D. Admittance control

For a system of N joints, with angles $\theta \in \mathbb{R}^N$, that is subjected to external joint torques τ_{ext}, an admittance controller is specified by [16],

$$M(\ddot{\theta}_d - \ddot{\theta}_0) + B(\dot{\theta}_d - \dot{\theta}_0) + K(\theta_d - \theta_0) = \tau_{\text{ext}} \tag{5}$$

where M, B, K are effective mass, damping, and spring constant matrices. In (5), the desired joint angles θ_d respond to external forces with second-order dynamics equivalent to a forced spring-mass-damper around a nominal set point θ_0, which is the trajectory that the desired set point returns to at unforced steady state. Admittance control modifies desired joint angles that are tracked by a low-level controllers.

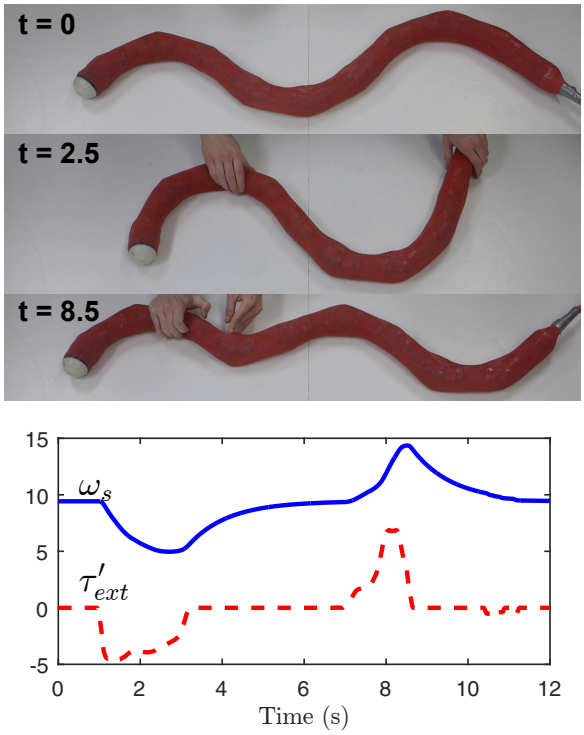


Fig. 1. Snake robot experiencing external forces, above, and corresponding plot of the shape force τ'_{ext} and serpenoid spatial frequency ω_s, below. When a forcing is applied, the torques measured at all joints are mapped into a shape force, resulting in a change in the wavelength of the serpenoid curve. When forcing is removed, the wavelength returns to its nominal value. ω_s has a nominal set point producing 1.5 body waves, and complies in this example to one or two body waves.

III. Shape-based compliance

A shape function $h : \Sigma \to \mathbb{R}^N$ maps a point σ in the shape parameter space $\Sigma \in \mathbb{R}^M$, into the joint space of an N-joint mechanism $\theta \in \mathbb{R}^N$,

$$\theta_\gamma = h(\sigma_\gamma, \beta) \tag{6}$$

where $\gamma = \{0, d\}$ correspond to the nominal and desired shape parameters, respectively. Shape functions are composed of shape parameters and shape bases β. One example of a shape parameter space is the collection of parameters in the serpenoid curve in (1). Shape bases determine the spatial coupling between different degrees-of-freedom, encoding static shapes for a particular mechanism. In prior work [2], [3], the serpenoid curve was chosen as the shape function, and shape parameters for curvature offset and amplitude $\sigma = \{\kappa, A\}$ were allowed to vary compliantly.

Shape-based control is based on an admittance control framework, where the control takes place directly in the shape-parameter space. Desired shape parameters σ_d are allowed to vary around a nominal shape parametrization σ_0,

$$M_\sigma(\ddot{\sigma}_d) + B_\sigma(\dot{\sigma}_d) + K_\sigma(\sigma_d - \sigma_0) = \tau'_{\text{ext}} \tag{7}$$

where the matrices M_σ, B_σ, and K_σ control the dynamic response of the shape parameters in response to a forcing term τ'_{ext}. These matrices are currently tuned by hand, and depend on physical aspects such as control loop frequency, actuator properties, and friction. We are currently investigating how to obtain these matrices generally. The forcing function is selected to be a function of external joint torques τ_{ext},

$$\tau'_{\text{ext}} = J\tau_{\text{ext}} \tag{8}$$

The Jacobian $J = \frac{\partial h(\sigma)}{\partial \sigma}$ is defined by the shape function, and maps torques from joint to shape space.

When the serpenoid shape parameters are uniform along the backbone curve, i.e., as they are in (1), the shape controller will be fully centralized. Fully centralized control may be effective in a uniform environment, such as a planar peg array with equally spaced pegs, but in environments that do not contain regularized structures it is desirable to allow portions of the snake robot's body to deform in response to *local* features. For example, in prior work [3] local variations in the amplitude of an underlying serpenoid equation were made with the shape function

$$h = A(s,t)\sin(\omega_s s - \omega_T t), \tag{9}$$

where $A(s,t)$ was defined as a sum of Gaussians, centered at positions μ_j along the backbone curve,

$$A(s,t) = \sum_{j=1}^{W} A_j \exp\left(-\frac{(s-\mu_j(t))^2}{2\psi^2}\right) \tag{10}$$

where W is the number of *activation windows* with independent amplitudes A_j and window widths specified by ψ. In previous work, the number of activation windows was statically set, and determined the degree of centralization used in corresponding shape-based controllers [3].

The coordination of joints located on the length of the backbone curve which is covered by a single window will be coupled. In the upper limit, a single window covering the full body length results in fully coupled amplitude modulation. Additional activation windows decentralize the coordination. Windows move along the body at the same rate as the serpenoid travelling wave, maintaining their independent shape parameters. This process provides an intuitive method for passing information along the body.

IV. Shape-based compliance in serpenoid phase

In this section we extend the work in [3] to include compliance in the spatial frequency and temporal phase of a serpenoid shape function.

A. Spatial frequency compliance and variable centralization

In a fully centralized controller, locally applied forces change the shape of the entire body. With an intermediate level of centralization, the shape can deform locally to environmental features. Here we use a modified version of the activation windows discussed in Section III. The shape function in Equations 9, 10 contains a summation of local variations in a shape parameter. The formerly used Gaussian

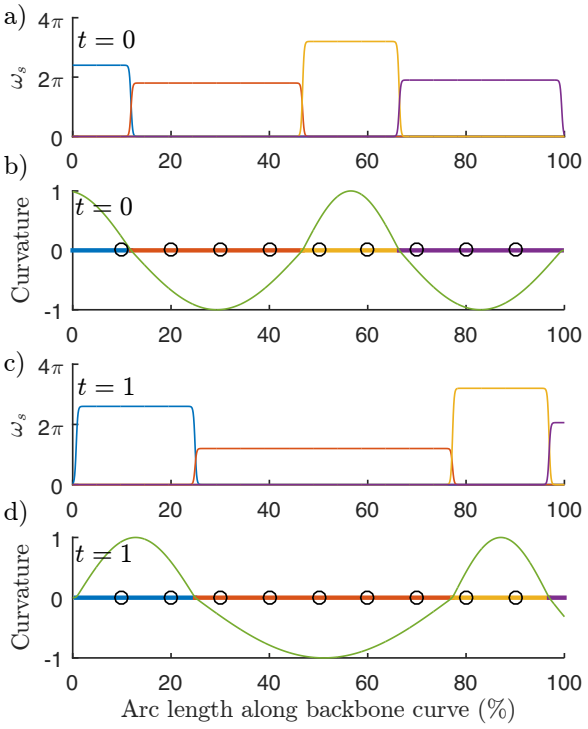


Fig. 2. An illustration of activation windows, which locally modulate the spatial frequency ω_s of the shape. a) Smooth-rectangular activation windows at $t = 0$. The sigmoid-rectangular windows overlap minimally. b) Curvature along the backbone at $t = 0$. The window edges are attached to the points of zero curvature on the backbone curve. Black circles represent joints, which lie at fixed locations along the backbone arc. c) Smooth-rectangular activation windows at $t = 1$. The wave has moved along the backbone, and the wavelength in the red window has grown. Where it previously coupled three joints, it now couples five. d) Curvature along the backbone at $t = 1$.

function windows overlap when summed, preventing each window from responding independently. We desire the activation windows to independently comply in response to local external forcing, so we modify the activation function to a smooth-rectangular step function defined by a sum of sigmoids. Specifically, the spatial frequency parameter is locally modulated as

$$\omega_s(s) = \sum_{j=1}^{W} \omega_{s,j}\left(\frac{1}{1+e^{-m(s-s_{j,\text{start}})}} + \frac{1}{1+e^{m(s-s_{j,\text{end}})}}\right) \tag{11}$$

where m controls the length of the transition period between shape parameters in neighboring windows, W is the number of windows, window j spans the backbone over arc length $(s_{j,\text{start}}, s_{j,\text{end}})$, and the spatial frequency in window j is $\omega_{s,j}$. The amplitude parameter A in this work is also modified with sigmoid functions, and the resulting shape function, similar to Equation 9, is

$$h = A(s,t)\sin(\omega_s(s,t)s - \omega_T t). \tag{12}$$

Each activation window spans a length of the backbone between its start and end locations, $(s_{j,\text{start}}, s_{j,\text{end}})$ in (11). These locations are attached to, and move with, specific points on the wave. We choose to attach window start and

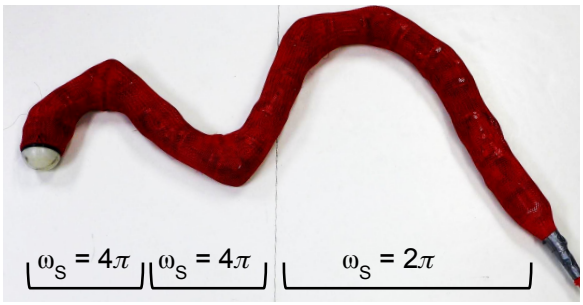


Fig. 3. A demonstration of multiple independent activation windows. In this case, there are three activation windows, each of which covers half of a wave length. The first two windows have a high spatial frequency, and coordinate only a few joints each, while the third has a low spatial frequency and coordinates many joints.

end locations to points where the serpenoid shape function is equal to zero. Then, as the shape spatial frequency within a window changes, the distance between attachment points changes, which results in the total number of windows varying.

The number and length of activation windows dynamically change during runtime. As the travelling wave moves along the backbone, new windows are generated at the head. The length of each window is determined by the spatial frequency shape parameter, and shape parameters change in response to torques, as specified by (7) and (8), resulting in changes in window length. If a window near the front shrinks, those near the back move forwards, and new windows are generated at the tail to cover the remaining backbone.

The desired angle for each joint is calculated based on its distance along the backbone in the shape function in Equation 12. As shown in Figure 2, the number of joints covered by a window varies with window size. Each window covers a section of the backbone and coordinates the joints located at backbone lengths between the start and end of that window. This means that the degree to which the control in the joint space is centralized can vary widely within the same controller in real-time in response to external forces.

B. Temporal phase compliance

The temporal frequency ω_T of the serpenoid curve (1) determines the speed at which the curvature wave travels along the backbone. In this work we allow the temporal phase $\omega_T t$ to be compliant, which allows the serpenoid wave to slow down in response to external forces. We define a temporal phase compliant parameter $\sigma = \Omega_t$, with a nominal value that changes in time, $\Omega_{t,0} = \omega_T t$.

When the snake robot becomes jammed between obstacles, and can no longer glide forward, the phase will be delayed. The delay in phase causes the spring-like dynamics of the phase parameter to wind up, causing the torque applied by the joints to grow while spatial shape parameters adjust. The spatial parameters will adjust until the robot conforms to the obstacles. Once the robot has conformed, it will be able to glide forward, and the phase will unwind, as shown in Figure 4. This behavior is shown in a supplementary video at `https://youtu.be/kzFhENsUx3o`. The temporal

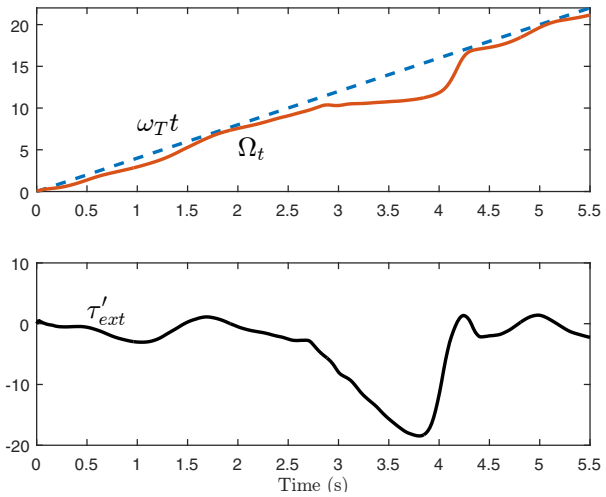


Fig. 4. The response of the temporal phase Ω_t to an external forcing τ'_{ext} in a fully centralized shape-based controller. A force is manually applied at $t = 2.75$ to slow the phase, and released at $t = 3.75$, after which the phase increases to return to its nominal value, $\omega_T t$.

phase synchronizes the motion of the locally compliant activation windows. As such, similar to the way in which the phase of all hips in a walking robot are coupled in a CPG framework [15], the independent sections of the snake robot all move at the same speed, and the temporal phase is a fully centralized parameter.

C. Central pattern generator comparison

A modified CPG-based controller was derived to serve as the basis of comparison in this work. We looked to past methods of feedback incorporation, such as in [15], in which feedback terms are added to the phase and amplitude evolution equations. For a valid comparison to be made to our shape-based controller, the CPG defined in Section II-C was modified to include forcing terms that vary the spatial frequency and amplitude based on external joint torques.

Two versions of this control modification were tested to create centralized or decentralized compliance of the amplitude and frequency parameters. In the centralized case, the amplitude and spatial frequency parameters in the CPG are changed globally based on a synthesis of all joint torques. The CPG phase offset ϵ, which determines the steady state spatial frequency, is governed by a forced oscillator around a nominal phase offset ϵ_0. The sensed joint torques τ_{ext} are multiplied by the sign of the joint angles and averaged for a net body forcing,

$$M_\epsilon \ddot{\epsilon} + B_\epsilon \dot{\epsilon} + K_\epsilon(\epsilon - \epsilon_0) = -\overline{\tau_{\text{ext}} \operatorname{sign}(\theta)} \qquad (13)$$

where the constants M_ϵ, B_ϵ, and K_ϵ control the dynamic response of the global phase offset. As in [13], the amplitude A converges to a set value A_0, but here we have added a forcing from external joint torques,

$$\ddot{A}_i = a_i\big(\frac{a_i}{4}(A_0 - A_i) - \dot{A}_i - \overline{\tau_{\text{ext}} \operatorname{sign}(\theta)}\big) \qquad (14)$$

This method, which we term the "centralized feedback" CPG adjusts the amplitude and spatial frequency to all oscillators equally.

In the a "decentralized feedback" case, individual oscillator parameters are changed based on local joint torques. In this case we modify the evolution of the phase offset ϵ between neighboring oscillators to include feedback only from joint torques on their corresponding joints,

$$M_\epsilon \ddot{\epsilon}_{i,i+1} + B_\epsilon \dot{\epsilon}_{i,i+1} + K_\epsilon(\epsilon_{i,i+1} - \epsilon_0) = -\frac{\tau_i \operatorname{sign}(\theta_i) + \tau_{i+1} \operatorname{sign}(\theta_{i+1})}{2}. \quad (15)$$

The amplitude of each oscillator is here forced only by its corresponding joint,

$$\ddot{A}_i = a_i\big(\frac{a_i}{4}(A_0 - A_i) - \dot{A}_i - \tau_i \operatorname{sign}(\theta_i)\big). \quad (16)$$

This is comparable to a shape-based controller with fixed sized windows anchored over each joint. These CPG-with-feedback approaches are similar to the shape-based approach. However, these CPGs do not explicitly pass shape information down the body of the robot with the underlying wave, nor pass amplitude or spatial frequency information from one oscillator to the next. The CPGs described here do not adaptively change the level of decentralization, as the number of oscillators is constant during run-time.

V. Experimental comparison of controllers

An experimental comparison of controller performance was conducted with a snake robot compliantly moving through the peg board shown in Figure 5. The robot used throughout this work was composed of eighteen identical series-elastic actuated modules [17], arranged such that the axes of rotation of neighboring modules were torsionally rotated ninety degrees relative to each other; in this work, only planar gaits were tested, so only nine modules were active. Deflection between the input and output of a rubber torsional elastic element was measured using two absolute angular encoders and used to compute the torque experienced by each module. The pegs were placed in an irregular pattern. The robot was covered with a braided polyester sleeve to reduce friction, with reflective markers attached along the backbone over the joint axes. The isotropic friction of the polyester sleeve prevented the robot from moving without using the pegs. Data was collected with a four-camera OptiTrack motion capture system (NaturalPoint Inc., 2011).

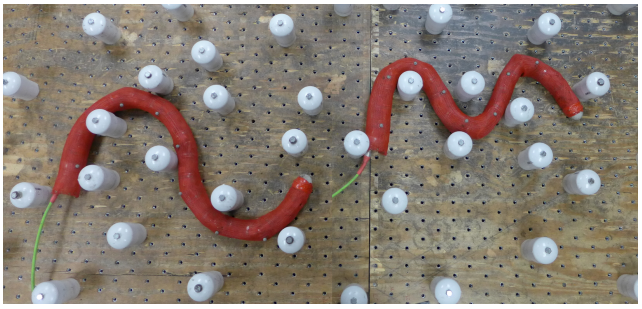


Fig. 5. The snake robot compliantly adjusts its spatial frequency as it moves through an unknown environment. A larger spatial frequency corresponds to more waves present on the robot.

For each trial, the robot initial conditions were randomized, and the position of the robot was tracked until either two minutes passed or the front half of the robot left the outer boundary of the pegs. Ten trials were conducted for each controller, with the same nominal wave parameters ($\omega_{s,0} = 3\pi$, $A_0 = \frac{\pi}{4}$). The distance traversed was measured in terms of the smoothed arc length travelled by the averaged marker position. Performance was compared with a "time to traverse" metric, which measures the number of body wave cycles required to move one meter. Locomotive speed could be increased by increasing the nominal temporal frequency; normalizing by number of cycles removes this effect. Controllers that result in poor locomotion performance either through becoming trapped frequently or thrashing in place will have a high value, and those that progress smoothly will have a lower value.

Our prior experiments identified the best anchor points for activation windows [3], and found that shape-based control performed best when the nominal window corresponded to one half wavelength, as shown in Figure (2). Further experiments compare the distributed-feedback CPG against the shape-based controller with either one, two, or three compliant parameters. Figure 6 presents the results of experiments comparing implementations of shape-based compliant and CPG-with-feedback control. The average performance for each controller is represented by the red bars with standard error.

VI. Discussion

Shape-based compliance outperformed the CPG controller, taking fewer cycles (less normalized time) on average to traverse the pegs. Shape-based compliance lowered both the average and variance of the time to traverse, and qualitatively made the path travelled smoother and more natural. This makes the robot's behavior more predictable and potentially easier to steer, either via a high level path planner or a remote human operator. The addition of spatial frequency compliance results in the snake "gliding" more naturally through the pegs, as its shape conforms more easily to the peg spacings. The addition of temporal phase compliance results in smoother gaits, because momentary jams cause a delay in phase while other parameters comply to the tight space. The data tend to be dominated by events in which the robot can not progress, which are heavily penalized via the time to traverse metric. In fully centralized and fully decentralized controllers, these events happen more frequently than for the best controllers tested in our full experiment. Videos of the snake locomoting through the peg array can be found at `https://youtu.be/kzFhENsUx3o`.

Both the CPG and the shape-based control are locally compliant in modulating serpenoid amplitude and wavelength with damped-spring dynamics. However, shape-based control offers a number of advantages. A shape-based controller is able to propagate shape information along the shape curve, which would be difficult to implement with CPGs. The incorporation of joint torque feedback is more intuitive in shape-based control, as it provides a mapping function

between joint and shape space. Both methods coordinate joints and can range in centralization, but shape-based control allows the level of centralization to vary, so the robot can comply to the environment while still maintaining shape structure. Shape-based control offers predictable, intuitive responses; joints will be coordinated within in a well defined shape even when significantly perturbed from its steady state trajectory. We found that an intermediate level of centralization performed best for spatial parameters, but that a fully centralized level performed best for the temporal parameter. In the CPG approach, which is more decentralized in both spatial and temporal parameters, we found that the waveform degenerated in highly constrained positions.

VII. Conclusions

This work extended shape-based control to spatial frequency and temporal phase serpenoid shape parameters.With this controller, a snake robot navigated its way autonomously through unknown, irregular environment with only joint-level torque feedback. The shape-based framework provides intuitive ways to design behaviors and synthesize feedback. This approach works equivalently with any number of joints, as the joint angles are taken from a continuous shape curve, and is fully extensible to three-dimensional motion. The approach is purely a reactive control strategy, which does not require a physical model of the robot or the environment. Because our snake robot operates in friction dominated domains, a full model including contact between the robot and terrain has not been necessary, and may be computationally prohibitive. These results, without a full analytical model, are experimental in nature. A theoretical analysis on how shape parameter variations interact with robot and environment geometry within our method is a direction of future work.

Adding spatial frequency and temporal phase compliance resulted in a more natural, smooth motion, while decreasing the variability in locomotor performance. In the future we will test whether this is the case for a wider range of peg spacing, both irregular and regular. Simply making more parameters locally compliant within a shape does not always improve performance. Similarly, varying the centralization was beneficial for spatial parameters, but unproductive in temporal parameters. For a given terrain, there does appear to be a level of centralization, shape structure, and compliance that maximizes performance. In future work, we will investigate how to select these factors from the environment and robot design. We will also seek to incorporate additional sensing modes, such as contact sensors, IMUs, vision, and directional steering, to our controller.

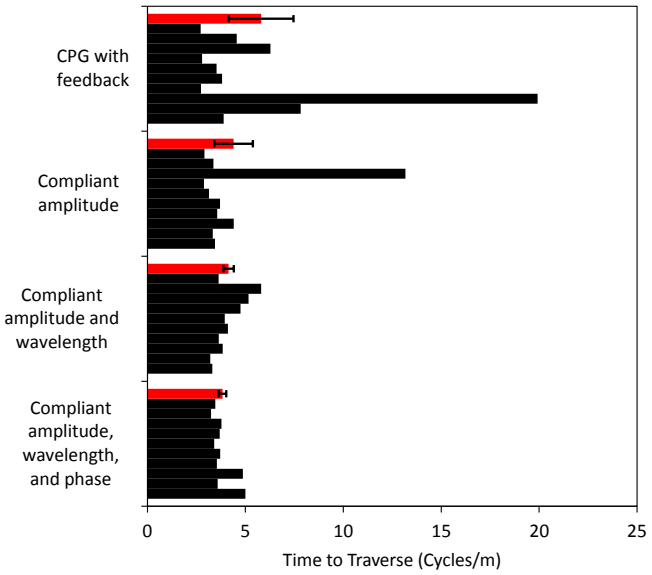


Fig. 6. Comparison of shape-based and distributed-feedback CPG control methods for a snake robot traversing a irregularly spaced peg array. Black bars represent 10 individual trials and the red bars controller mean, with standard error bars. A highly successful controller will minimize both the time to traverse and the variance in the time to traverse, as large variance indicates that the robot was stuck in place rather than travelling smoothly.

References

[1] 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 23, no. 9 (2009): 1131-1158.

[2] M. Travers, and H. Choset. "Shape-Constrained Whole-Body Adaptivity." International Symposium on Safety, Security, and Rescue Robotics, West Lafayette, IN (2015)

[3] M. Travers, J. Whitman, P. Schiebel, D. Goldman, and H. Choset. "Shape-Based Compliance in Locomotion." In Robotics: Science and Systems (RSS) 2016.

[4] D. E. Koditschek, R. J. Full, and M. Buehler. "Mechanical aspects of legged locomotion control." Arthropod structure and development 33, no. 3 (2004): 251-272.

[5] M. Kalakrishnan, J. Buchli, P. Pastor, M. Mistry, and S. Schaal. "Learning, planning, and control for quadruped locomotion over challenging terrain." International Journal of Robotics Research 30, no. 2 (2011): 236-258.

[6] S. Hirose, "Biologically Inspired Robots." Oxford Univ. Press, 1993

[7] D. Rollinson and H. Choset. "Gait-based compliant control for snake robots." In Robotics and Automation (ICRA), pp. 5138-5143. IEEE, 2013.

[8] H. Date and Y. Takita. "Adaptive locomotion of a snake like robot based on curvature derivatives." IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 3554-3559. IEEE, 2007.

[9] T. Kano and A. Ishiguro. "Obstacles are beneficial to me! Scaffold-based locomotion of a snake-like robot using decentralized control." In 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 3273-3278. IEEE, 2013.

[10] P. Liljebck, K. Y. Pettersen, O. Stavdahl, and J. T. Gravdahl. "Snake robot locomotion in environments with obstacles." IEEE/ASME Transactions on Mechatronics, 17, no. 6 (2012): 1158-1169.

[11] A. A. Transeth, S. A. Fjerdingen, and P. Liljeback. "Snake robot obstacle-aided locomotion on inclined and vertical planes: Modeling, control strategies and simulation." In Mechatronics (ICM), pp. 321-328. IEEE, 2013.

[12] A. J. Ijspeert. "Central pattern generators for locomotion control in animals and robots: A review." Neural Networks, 21:642653, 2008

[13] A. J. Ijspeert, and A. Crespi. "Online trajectory generation in an amphibious snake robot using a lamprey-like central pattern generator model." In Robotics and Automation, pp. 262-268. IEEE, 2007.

[14] V. Barasuol, J. Buchli, C. Semini, M. Frigerio, E. R. De Pieri, and D. G. Caldwell. "A reactive controller framework for quadrupedal locomotion on challenging terrain." In Robotics and Automation, pp. 2554-2561. IEEE, 2013.

[15] S. Gay, J. Santos-Victor, and A. J. Ijspeert. "Learning robot gait stability using neural networks as sensory feedback function for central pattern generators." In Intelligent Robots and Systems (IROS) , pp. 194-201. IEEE, 2013.

[16] C. Ott, R. Mukherjee, and Y. Nakamura. "Unified impedance and admittance control." In Robotics and Automation (ICRA), pp. 554-561. IEEE, 2010.

[17] D. Rollinson, Y. Bilgen, B. Brown, F. Enner, S. Ford, C. Layton, J. Rembisz et al. "Design and architecture of a series elastic snake robot." In IROS, pp. 4630-4636. 2014.