Shape-Based Compliance in Locomotion

Matt Travers, Julian Whitman, Perrin Schiebel, Dan Goldman, and Howie Choset

***Abstract*—Having many degrees of freedom is both a blessing and a curse. A mechanism with a large number of degrees of freedom can better comply to and therefore better move in complex environments. Yet, possessing many degrees of freedom is only an advantage if the system is capable of coordinating them to achieve desired goals in realtime. This work supports the belief that a middle layer of abstraction between conventional planning and control is needed to enable robust locomotion of articulated systems in complex terrains. The basis for this abstraction is the notion that a system's shape can be used to capture joint-to-joint coupling and provide an intuitive set of controllable parameters that adapt the system to the environment in real time. This paper presents a generalizable framework that specifies desired shapes in terms of *shape functions*. We show how shape functions can be used to link low-level controllers to high-level planners in a compliant control framework that directly controls *shape parameters*. The resultant shape-based controllers produce behaviors that enable robots to robustly feel their way through unknown environments. This framework is applied to the control of two separate mechanisms, a snake-like and a hexapod robot.**

I. Introduction

Decoupling planning and control for locomotion in complex terrains does not, in general, produce robust real-time solutions. A planner can generate a motion plan, and then a closed-loop controller can execute it, but any slight uncertainly in modeling can easily cause the system to fail. In other words, decoupled planning/control solutions tend to be brittle. To address this, we fuse planning and control with a coherent middle layer that adds robustness to and decreases the complexity of coordinating many degrees of freedom during locomotion. *Shape functions* are presented as the element that helps define this middle layer for articulated systems. A shape function 1) analytically encodes the interdependent joint-to-joint motions necessary to produce desired behaviors in nominal locomotion, 2) provides sets of parameters that can be controlled to adapt mechanisms to complex environments while moving through them, and 3) offers low-dimensional, straight-forward connections to higher-level planners.

In terms of adapting locomotion in unstructured terrains, this work shows that admittance control can be specified in terms of *shape parameters*, which are the parameters used to define shape functions. By mapping joint torques into equivalent forces on shape, shape-based compliant controllers autonomously modify the system's shape to better comply to the environment during locomotion. Results of implementing shape-based controllers on two systems - a snake-like robot and a walking hexapod - are presented to highlight both the generality of this approach, as well as to provide different examples of systems compliantly feeling their way through complex terrains.

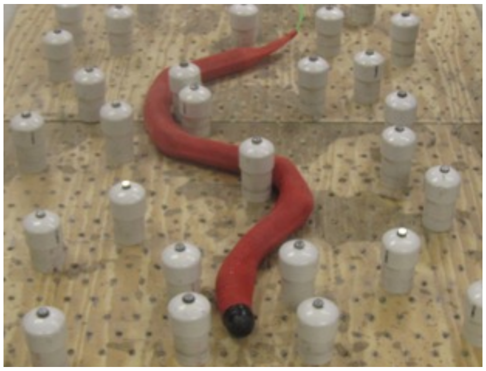

Fig. 1: Snake-like robot autonomously moving through a peg array with no a priori knowledge of the array parameters.

Our approach to mid-level, shape-based control bears resemblance to concepts in the motion planning and control communities. In particular, the controllers derived in this work can be interpreted as special cases of dynamic movement primitives (DMPs) [8], [5]. More specifically, decentralized implementations of shape-based compliance are closely related to work that appears in [7], which presents a method for online trajectory generation of an articulated body using a central pattern generator (CPG) [6]. Relative to these prior works, we make extensions that show how distributed joint torque measurements can be incorporated into shape-change dynamics. Additionally, we show how decentralized control methods can be modified to propagate shape information that improves locomotive performance. Direct experimental comparisons between our approach, a CPG-based controller, as well as a torque-modified CPG controller are presented.

II. Related Work

The shape-based compliant control framework presented in this work combines and makes extensions to several existing components in the literature on snake robot control, admittance control, and DMPs. Brief reviews of these topics are provided in this section.

A. Snake-Like Robots

Previous work on the control of snake-like robots can be grouped into two general categories: 1) model-based control and 2) model-free control. Model-based snake-robot control methods have previously been derived for both kinematic as well as dynamic system models. Examples of model-based controllers for snake-like robots include work in [12], [19], [9], [10], [15]. Model-based approaches make it possible to apply conventional planning and control techniques to snake robots, but, like all model-based approaches, rely on the accuracy

M. Travers, J. Whitman, and H. Choset are with the Robotics Institute at Carnegie Mellon University, Pittsburgh, PA, USA. D. Goldman and P. Schiebel are with the School of Physics at Georgia Institute of Technology, Atlanta, GA, USA.

of the environmental model. This is an especially limiting assumption in complex terrains.

Model-free control methods typically define feed-forward joint-space trajectories that are shown to empirically produce desired net behavior of the robot. Examples of model-free controllers for snake-like robots are presented in Hirose's pioneering work in [1], the pedal wave approach found in [22], the modal approach of Chirikjian [2], compound serpenoid model in [20], and a variety of others [21]. Recent work in [16] presents an interesting extension to model-free control for snake robots. Using an approximate force measurement, [16] derived an impedance-like controller [4] that adapted parameters in a feed-forward gait model. The work we present is related to [16], but extends to a variety of unstructured environments.

B. Admittance Control

Model-free control methods, like those discussed for snake-like robots in Section II-A, provide one way to connect high-level motion planning to low-level control under nominal conditions. A high-level planner selects between different pre-defined feed-forward trajectories, and low-level controllers regulate the system to follow these desired open-loop policies. In environments that make nominal operation either ineffective or impossible, we believe that locomotion control is better suited to adapt nominal feed-forward signals, θ_0, to produce augmented desired signals θ_d, than to attempt to generate entirely new motion plans altogether. The dynamic relationship that specifies how θ_0's are transformed to produce θ_d's forms the basis for what we consider to be a middle layer between planning and control for locomotion.

Admittance control provides one way to augment nominal system trajectories in the presence of external forcing. For example, consider a system with N joints, $\theta \in \mathbb{R}^N$, that is in contact with the environment. Considering a simplified dynamic model, an admittance controller for this system is specified by the following second-order system dynamics [13],

$$m\ddot{\theta} + k_d\dot{\theta} + k_p(\theta - \theta_d) = \tau_{\text{ext}} \qquad (1)$$

$$M_d(\ddot{\theta}_d - \ddot{\theta}_0) + B_d(\dot{\theta}_d - \dot{\theta}_0) + K_d(\theta_d - \theta_0) = \tau_{\text{ext}}, \qquad (2)$$

where m is an inertia, k_d and k_p are positive gains, τ_{ext} is the torque due to the external force, M_d is an effective mass, B_d an effective damping, and K_d an effective spring constant. Equation (1) represents the controlled second-order dynamics for the system, where the low-level controller is a standard PD control law. What makes this an admittance controller is that the desired set point θ_d in (1) has its own set of dynamics that adjust θ_d in response to external forcing, measured by τ_{ext}. Equation (2) specifies that the desired joint signal θ_d has the linear dynamics of a forced spring-mass-damper, with a nominal trajectory defined by θ_0. Note that in the absence of external forcing, the desired trajectory θ_d will converge to the nominal trajectory θ_0, and thus the admittance controller becomes a standard PD controller.

C. Dynamic Movement Primitives

Similar to the method of adjusting desired trajectories as a function of external forcing in admittance control, the method of dynamic movement primitives (DMP) also assigns dynamics to parameters that govern open-loop desired trajectory generation [17], [8]. While DMPs are more general than admittance controllers, the fundamental concept is similar: assign relatively simple stable dynamics that are modulated to create desired behaviors in the region of an attractor (or nominal set-point). For example, as noted in [8], the most basic form of a DMP is

$$\epsilon\ddot{y} = \nu(\rho(g - y) - \dot{y}) + f, \qquad (3)$$

which is a forced spring-mass-damper system, where ϵ and $\nu, \rho > 0$ govern the dynamic response of the system. The term f in (3) is a forcing term that can be used to modulate the response of y. The freedom in choosing f is what makes DMPs more general than admittance controllers. In other words, an admittance controller can be thought of as a special case of a DMP, where the forcing signal f in (3) is explicitly a function of proprioceptively-measured environmental forces. This point is made clear by rewriting (3) as,

$$M\ddot{y} + B\dot{y} + K(y - g) = f \qquad (4)$$

where $M = \epsilon$, $B = \nu$, and $K = \nu\rho$.

III. Shape-Based Compliant Control

Shape-based compliant control provides a middle-layer of abstraction between planning and control that naturally extends admittance control to articulated locomotion. This extension begins with shape functions. These geometric abstractions are composed of two core elements: shape bases and shape parameters. Shape bases determine the spatial coupling between different degrees of freedom. Shape parameters determine how a system's shape evolves as a function of time, i.e., shape parameters determine how different shape bases are blended together during locomotion.

A shape function h is formally defined as a function that maps a point σ in the shape parameter space (or just shape space) Σ, into the joint space of a mechanism, i.e., $h : \Sigma \to \mathbb{R}^N$, where $\theta \in \mathbb{R}^N$ defines the joint space for an N-joint mechanism.

A. Linear Shape Functions

We consider first the special case where h is a linear function in the shape parameters σ, i.e.,

$$h(\sigma) = \sum_{i=1}^{M} \sigma_i(t)\beta_i(s), \qquad (5)$$

where the $\beta_i(s)$'s are linearly independent shape basis functions. These basis functions spatially encode static joint-to-joint coordination, i.e., these basis functions encode static shapes for a particular mechanism. The expression for h in (5) can be used to map (1) and (2), that define an admittance controller in joint space, into an equivalent expression in the shape parameter space Σ using the following relationships,

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

$$\dot{\theta}_\gamma = \frac{\partial h(\sigma_\gamma)}{\partial \sigma_\gamma}\dot{\sigma}_\gamma = J\dot{\sigma}_\gamma \tag{7}$$

$$\ddot{\theta}_\gamma = \dot{J}\dot{\sigma}_\gamma + J\ddot{\sigma}_\gamma, \tag{8}$$

where $\gamma = \{d, 0\}$; σ_d defines a desired shape parameterization and σ_0 a nominal shape parameterization. The Jacobian matrix J in (7) is an analytical Jacobian that arises from changing coordinates from the shape parameter space to the joint space [18]. Substituting (6)-(8) into (2) yields,

$$M'_d(\ddot{\sigma}_d - \ddot{\sigma}_0) + B'_d(\dot{\sigma}_d - \dot{\sigma}_0) + K'_d(\sigma_d - \sigma_0) = \tau'_{\text{ext}}, \tag{9}$$

where $M'_d = J^T M_d J$, $B'_d = J^T B_d J + J^T M_d \dot{J}$, $K'_d = J^T K_d J$, and $\tau'_{\text{ext}} = J^T \tau_{\text{ext}}$. The second-order dynamics in (9) along with the shape function (5) provide an analytical basis as well as interpretation for shape-based compliant control. The compliant control of shape parameters is effectively equivalent to compliant control of joint angles, but the overall shapes a mechanism is allowed to take on are constrained to remain within a particular basis (specified by the β_i's). While a shape can freely change within a particular basis, the fundamental structure of the shape will remain fixed relative to the nominal shape pattern. Note that, similarly to admittance control in joint space, admittance control in shape space double integrates (9) at each time step to solve for σ_d. This value is then used to define a set of desired joint trajectories via the shape function, i.e., $\theta_d = h(\sigma_d)$.

B. General Shape Functions

In the case where the shape function h is not necessarily linear in the shape parameters, or where the shape-basis functions $\beta_i(s)$ are not linearly independent, shape-based compliant control is no longer equivalent to admittance control defined in the joint space. We discuss this case in Section 4, noting that admittance control is a special case of a DMP and that (9) is thus effectively a special case of a DMP defined in shape space. Furthermore, dropping the explicit relationship on J in the definitions of M'_d, B'_d, and K'_d in (9), a general DMP in the shape space can be defined by

$$M^\sigma_d \ddot{\sigma}_d + B^\sigma_d \dot{\sigma}_d + K^\sigma_d(\sigma_d - \sigma_0) = F_\sigma, \tag{10}$$

where the values of M_d, B_d, and K_d directly control the dynamic response of the shape parameters and F_σ is an arbitrary forcing term. Based on the fact that we know from (9) that the Jacobian J maps joint torques into equivalent shape forces, this work selects F_σ to be an explicit function of the externally applied forces, e.g., $F_\sigma = J\tau_{\text{ext}}$ where $J = \frac{\partial \tilde{h}(\sigma)}{\partial \sigma}$ and $\tilde{h}(\sigma)$ is a general shape function on Σ.

IV. Shape-Based Compliance for Snake-Like Robots

Shape-based compliance is implemented on a snake-like robot in this section and a number of experimental results are provided. The robot hardware used in the experiments, and used throughout the snake robot experiments in the rest of this work, was composed of eighteen identical series-elastic actuated modules [14]. The modules were arranged such that the axes of rotation of neighboring modules were torsionally rotated ninety degrees relative to each other. Deflection in a torsional elastic element, made from natural rubber, was measured at the output of the motor drives using two absolute angular encoders. The value of this deflection was used to compute the output torque experienced by each module.

A. Whole-Body Compliance

The serpenoid model, defined in the snake robot literature, has the following form,

$$\theta = \kappa + A\sin(\eta s - \omega t), \tag{11}$$

where κ is an offset parameter, A an amplitude parameter, η is referred to as a spatial frequency parameter, ω is a temporal frequency, $s = \{0, \delta s, 2{\cdot}\delta s, \ldots, N{\cdot}\delta s\}$, δs is a link length, and N is the total number of joints in the mechanism. Note that in (11), $\theta, \kappa, t \in \mathbb{R}^N$, where κ and t are actually $N \times 1$ vectors of ones multiplied by the corresponding parameter values, and $\sin()$ acts component-wise. The serpenoid model in (11) is a whole-body kinematic model in the sense that changing any of the parameters κ, A, η, or ω changes the shape of the entire robot.

Equation (11) is conventionally used to specify open-loop desired trajectories θ_d, defined as static functions of κ, A, η, and ω, that are tracked by low-level motor controllers under nominal conditions, i.e., when the system is unconstrained by the environment. This work extends conventional approaches by using (11) to define both nominal as well as desired trajectories, where the nominal signals are held fixed in κ, A, η, and ω, and the desired signals are enabled to compliantly vary in these parameters.

To accomplish this, the expression for θ in (11) is first rewritten in the form of a shape function as,

$$\theta = h(\sigma) = \kappa + A_1(t)\sin(\eta s) + A_2(t)\cos(\eta s), \tag{12}$$

where $A_1(t) = A\cos(\omega t)$ and $A_2(t) = -A\sin(\omega t)$, and $\sigma = (\kappa, A_1, A_2)$ and $\beta = (1, \sin(\eta s), \cos(\eta s))$. Because this shape function is linear in the elements of σ, a shape-based compliant controller of the form (9) can be derived and implemented on the snake robot hardware.

It is possible to use the formulation (9) to produce simultaneous compliance in $\sigma = (\kappa, A_1(t), A_2(t))$, but we isolate here only the amplitude parameters for simplicity of presentation. Figure 2 shows an experiment where the amplitude parameters of a static wave, i.e., $\omega = 0$, were compliantly controlled. Note that in this case $A_1(t) = A$ and $A_2(t) = 0$.

The plot in the left column of Figure 2 shows that during the trial, at time $t = 1$s, a force was applied to the robot that produced a corresponding negative force (τ'_{ext}) on the desired amplitude parameter A_d. The effect of this force caused the value of A_d to decrease. The pictures in the right column of Figure 2 show the original shape of the robot at $t = 0$s, as well

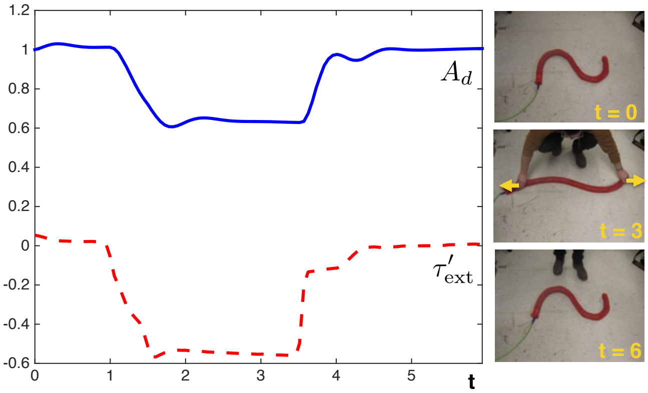

Fig. 2: Plot of the offset parameter A in (11) as well as forcing term τ'_{ext} in (9). The right column shows picture of the robot at different times during the experiment.

as the augmented shape with decreased amplitude at $t = 3$s. The external force was removed at $t = 3.5$s, after which the the robot's shape returned to its unperturbed form, as shown in amplitude plot as well as picture taken at $t = 6$s in Figure 2.

B. Localized Shape Modulation

Whole-body shape compliance is a centralized method of controlling the shape of a snake-like robot; all of the robot's degrees of freedom are simultaneously controlled to achieve a single objective. In environments with regularized structures like the peg array shown in Figure 3(a), a centralized shape controller may be a reasonable control strategy. In environments that do not contain regularized structure, like the randomized peg array shown in Figure 3(b), a decentralized extension, that allows the system to locally comply different portions of its body, is desirable.

1) Shape Modulation : This section introduces spatial activation windows as a method of decentralizing shape-based control for snake-like robots. More specifically, we define the following extension to the serpenoid function (11),

$$\theta = A(s,t)\sin(\eta s - \omega t), \tag{13}$$

where

$$A(s,t) = \sum_{j=1}^{W} A_j \exp\left(-\frac{(s-\mu_j(t))^2}{2\psi^2}\right), \tag{14}$$

and W is the number of desired activation windows, each A_j is the independent amplitude in window, ψ defines the window widths, and each μ_j defines the location of the center of a window as a function of arc length s. Note that, without loss of generality, we have dropped the dependency on the curvature offset term κ in (13). Equation 13 makes it possible to independently modulate the amplitude of different portions of the waveform $\sin(\eta s - \omega t)$ by independently adjusting the amplitude parameters A_j in (14).

It is possible to freely choose the window center locations, the number of windows, as well as the window widths. When adjusted together, these parameters provide an intuitive means for specifying different spatial coupling relationships between portions of the robot's body. For example, in the lower limit, selecting a single window located at each joint with a small window width effectively decouples shape modulation completely. In the upper limit, a single window located in the center of the mechanism with a large window width represents entirely coupled shape modulation. The fact that any joint-to-joint coupling relationship between these two limits can be specified, by defining different shape functions, implies that the degree to which a shape controller is centralized or decentralized is effectively a free parameter in the control design.

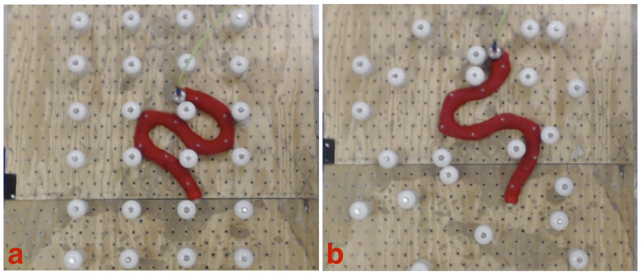

Fig. 3: Snake robot in a) structured peg array and b) randomized peg array. Videos of the robot traversing the randomized peg array are included in the supplementary material.

One additional freedom in (13) is the ability to specify how or if the activation windows move as a function of time. The method chosen in this work is to allow activation windows to move with the waveform $\sin(\eta s - \omega t)$ as it travels down the robot's body. Allowing the activation windows to move with the waveform provides an intuitive method for *passing shapes* down the robot's body (see Figure 4). This behavior is helpful during locomotion, as passing shapes down their bodies appears to dramatically help snakes, biological and robotic, propel themselves forward.

Allowing the spatial activation windows to move with the waveform $\sin(\eta s - \omega t)$ necessitates either an infinite number of windows be defined, or, as we select, that a spatial recirculation condition be introduced. For example, assuming that the number of waves specified by η in (13) remains constant, e.g., $\eta = 1.5$, a static number of activation windows is defined, e.g., $W = 2\eta$, and is used to subsequently define the window center locations as,

$$\mu_j(t) = (1 + 2\cdot(j-2))\cdot\frac{\pi}{2}\frac{1}{\eta} + \text{mod}\left(\frac{\omega}{\eta}t, \frac{\pi}{\eta}\right), \tag{15}$$

where $j = 1, 2, \ldots, W$. In this example, the window center locations at $t = 0$ are chosen to correspond with the points of highest curvature on the waveform $\sin(\eta s - \omega t)$. The modulus π/η in (15) was chosen to correspond to the static distance between these window centers. When each window center travels the distance that brings it to the original position of the window center in front of it, the window is recirculated back to its original position.

When recirculating the window centers, an additional set of constraints are needed to ensure that shapes are smoothly transferred from one window to the next. These constraints

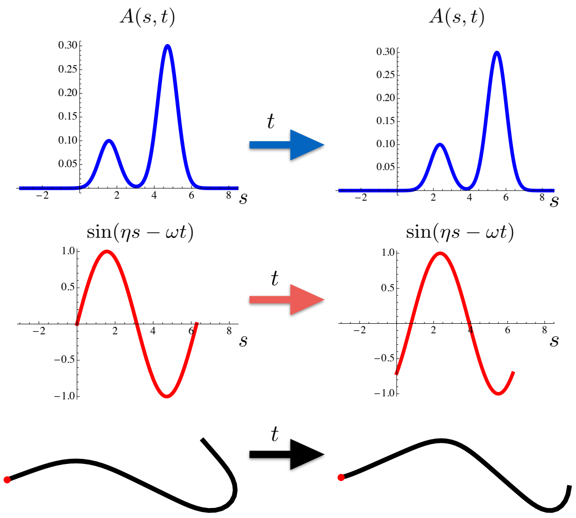

Fig. 4: Amplitude modulation signal $A(s,t)$ moves with carrier waveform $\sin(\eta s - \omega t)$ as a function of time. The effect of this choice is that shapes propagate down snake-like mechanism from head (red dot) to tail.

are defined by

$$A_{k+1}(t) = A_k(t)|_{t=\pi/\omega} \tag{16}$$

where $k = [1, 2, \ldots, W - 1]$. The constraints in (16) pass shape parameters from one window to the next at the times when the window center locations are recirculated.

2) Compliant Shape Modulation : It is not possible to extend the same derivation of admittance control in the shape space presented in Section IV-A to the shape function associated with (13), as the shape basis functions in this function will no longer be linearly independent. Thus, the DMP approximation to shape-base admittance control (10) is used in this case.

Figure 5 shows an experiment using an implementation of DMP-based shape compliance, where, $W = 3$ and $\sigma = (A_1, A_2, A_3)$. The initial shape of the robot was specified by the parameters $A_1 = A_2 = A_3 = 1$, $\psi = \pi/6$, $\eta = 1.5$, $\omega = 0$, and $\mu_1 = \pi/3$, $\mu_2 = \pi$, $\mu_3 = 5\pi/3$. During the experiment, at about $t = 2$s, forces were applied to the robot near the middle of its body as well as near its head, as shown in the pictures in the right column of Figure 5. The result of applying these forces was a decrease in the value of the amplitude parameters in the first-two windows centered at μ_1 and μ_2. At $t = 5$s, the forces on the front portion of the robot were relaxed and the system restored to nearly its original shape (where $A_1 = A_2 = A_3 = 1$). At $t = 6$s, a set of forces was again applied to the robot, but in the middle and back portions of its body. This combination of forces resulted in the amplitude of the windows centered at μ_2 and μ_3 to be decreased. At $t = 8$s the second set of forces were released and the robot again converged back to nearly its original shape.

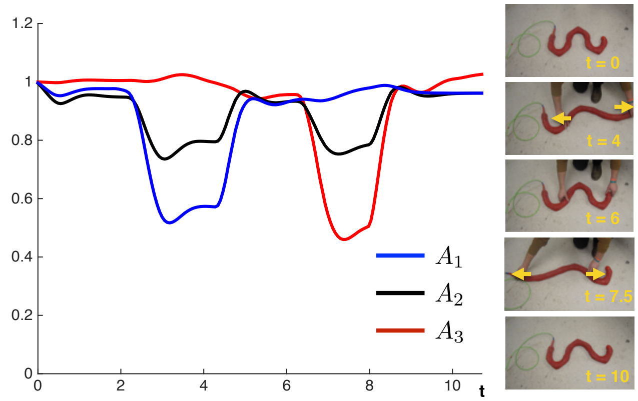

Fig. 5: Independent amplitude modulation in three spatial activation windows.

C. Three-Dimensional Shape Modulation

One of the main benefits in the shape function and resultant shape-based compliant control framework is that this approach is easily generalizable to a variety of different shape models. For example, this section presents an extension of the model in (13) that enables compliant shape control in three-dimensional spaces.

The snake robot used in the experimental demonstrations in this work has N-joints, where every other joint is rotated ninety degrees out-of-plane relative to its neighbors. The following shape model can thus be defined in terms of the odd o and even e joints for the mechanism as,

$$\theta_i^o = \sum_{k=1}^{W} A_k \cos(\varphi_k) \exp\left(\frac{-(s - \mu_k(t))^2}{2\psi^2}\right) \sin(\eta s - \omega t) \tag{17}$$

$$\theta_j^e = \sum_{k=1}^{W} A_k \sin(\varphi_k) \exp\left(\frac{-(s - \mu_k(t))^2}{2\psi^2}\right) \sin(\eta s - \omega t), \tag{18}$$

where $i = 1, 3, \ldots, N - 1$ and $j = 2, 4, \ldots, N$. The inclusion of the $\sin(\varphi_k)$ and $\cos(\varphi_k)$ terms in (17) and (18), respectively, makes it possible to effectively rotate amplitudes A_k, in different spatial windows, from one plane in the joint space (odd/even) into the other (even/odd). For example, if $\varphi_k = 0$, the amplitude of the $\sin(\eta s - \omega t)$ waveform in that window will be contained in the plane of the odd joints. As φ_k is increased (or decreased), the waveform will begin to twist out of the odd-joint plane and into the even-joint plane. At $\varphi_k = \pi/2$, the $\sin(\eta s - \omega t)$ waveform in window k will be completely contained in the plane of the even joints.

An example in which the rotational parameters φ_k are compliantly controlled using a compliant DMP defined in the shape space, where $W = 3$ and $\sigma = (\varphi_1, \varphi_2, \varphi_3)$, is shown in Figure 6. The robot is this experiment was initialized with $\eta = 1.5$, $\omega = 0$, $A_1 = A_2 = A_3 = 1$, $W = 3$, $\psi = \pi/6$, and $\mu_1 = \pi/3$, $\mu_2 = \pi$, $\mu_3 = 5\pi/3$. The pictures in the right column of Figure 6 show that the robot was placed in a position at $t = 0$s where a significant portion of its body was

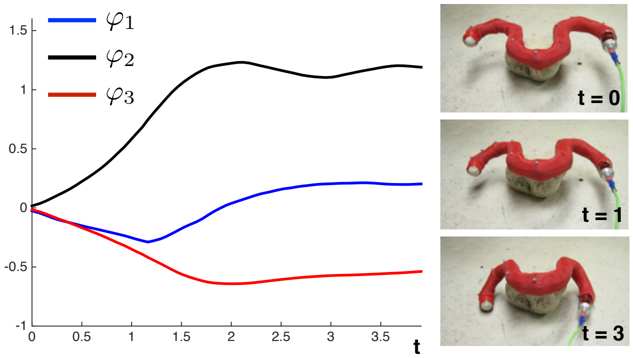

Fig. 6: Independent torsional compliance in three spatial activation windows.

cantilevered out from its middle section. When the experiment began, the plot in the left column of Figure 6 shows that the values of φ_1 and φ_3 initially decreased, while the value φ_2 increased. As the robot settled, the value of φ_1 returned to nearly zero. The picture of the robot in the bottom right corner of Figure 6 shows that the net effect of this torsional compliance at $t = 3$s caused the robot's body to conform to the rock feature that it was originally resting.

Figure 7 shows the same controller as that used in the experiment shown in Figure 6, except where $\omega = 1$, i.e., the waveform $\sin(\eta s - \omega t)$ was propagated down the robot's body during this trial. Figure 7 shows, to the best of the authors knowledge, the first autonomous trial of a snake robot adapting three-dimensional locomotion without a priori knowledge of the terrain. The rock pile shown in Figure 7 is meant to approximate unstructured rubble, and thus this type of control may have a potential impact on the practical application of snake-like robots in search and rescue tasks.

V. Shape-Based Compliant Control in Legged Locomotion

Shape-based compliance is not limited to snake-like robots. In terms of the framework presented in this work, any robot for which a reasonable shape function can be defined and that can sense torques at its joints, is able to be compliantly shape-controlled. To demonstrate this, a shape function as well as shape-based compliant controller was developed for the hexapod robot shown in the top row of Figure 8. This legged robot was constructed using the same series-elastic modules used to construct the snake robot used in the experiments included in Section IV.

Defining a shape function for the hexapod robot began by first considering each of the mechanism's legs independently. Each leg is composed of three actuated joints. Relative to a frame attached to the base of each leg, the proximal joint is aligned with yaw axis, followed by two distal joints each aligned with the pitch axis. Motions that effect only the two distal joints of each leg are considered. This choice is made for the sake of simplicity. The presented approach can easily be extended to more complicated examples.

We derive an isolated shape function for each leg by first defining the set of desired leg shapes to be all configurations that approximately match the geometry of the leg in the workspace to continuous sigmoid curves of the form,

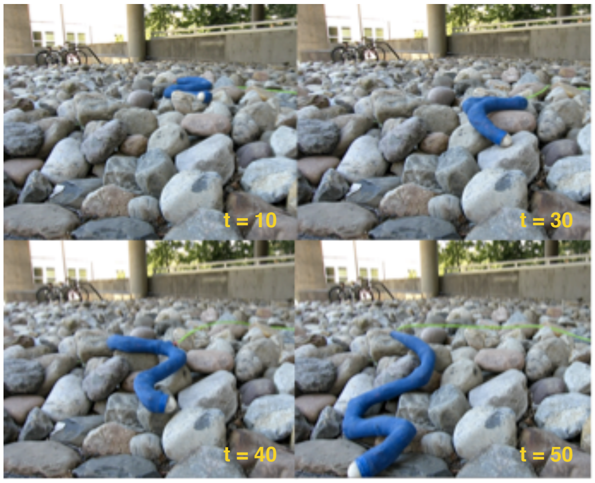

Fig. 7: Autonomous locomotion over unknown rock pile using torsional shape compliance. A video of this behavior appears in the supplemental material.

$$\alpha_i(s) = \frac{A_i}{1 + \exp(\lambda(s - \mu))}, \tag{19}$$

where the parameters λ and μ fit the curve α to the physical parameters of leg and s is arc length measured from the base of the leg outward toward the toe. Figure 8b shows a desired configuration of the hexapod leg where the amplitude parameter A_i in (19) is positive. Figure 8c shows a desired configuration of the hexapod leg where the amplitude parameter A_i in (19) is negative.

A shape function that produces the desired leg configurations in the workspace can be defined by intuition, i.e., $h(\sigma) = A_i \beta_i$, where $\beta_i = [1, -1]$ and A_i is appropriately scaled for the leg. In more general cases, the curvature of continuous shapes in the work space, specified by smooth functions of s, e.g., (19), can be used to derive shape functions. This is related to the fact that joint angles approximate continuous curvature defined as function of arc length.

Returning to the composite robot, i.e., considering all six legs, the shape function can be defined defined by $\theta = h(\sigma) = \sum_{i=1}^{6} \sigma_i \tilde{\beta}_i$, where each $\tilde{\beta}_i \in \mathbb{R}^{12 \times 1}$ and contains all zeros except for the two entries corresponding to β_i. A DMP-based shape compliant controller for this system was then derived by computing $J = \partial h / \partial \sigma$, which was used to project the measured joint torques into shape forces.

Figure 9 shows an example of shape-based compliant control applied to the hexapod robot. During the experiment, the value of F_σ in (10) included a term that compensated for gravity. While the robot is standing in a nominal position, the joint torques will measure the gravitational load produced by the robot's body. A feedforward gravity model (F_g) was thus incorporated in the the compliant controller. Additionally, the robot in this experiment was anti-compliant, i.e., $F_\sigma = F_g - J^T \tau_{\text{ext}}$. Anti-compliance caused the robot's shape to reject disturbances as opposed to being compliant to them. We

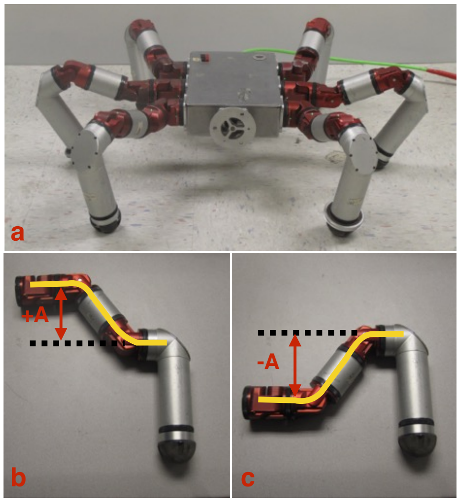

Fig. 8: Walking hexapod robot (a). Valid configurations for hexapod leg in positive (b) and negative (c) amplitude poses.

note that the experiment in this section is related to [3], except we assume the robot uses only joint torque measurements to adapt to changing slopes.

During the experiment shown in Figure 9, the height of the robot's feet on the left side of its body was suddenly changed by raising the platform the robot was standing on. Using no inertial sensors, the robot compliantly adjusted its body pose to counteract the forces induced on its body by raising the platform.

VI. Comparison of Control Methods

This section presents experimental results that compare several different implementations of shape-based compliant control, an implementation of a CPG-based control method, as well as a compliant CPG controller.

A modification to the dynamic shape model in (10) was necessary in the peg experiments due to the fact that torque measurements measured at the joints included frictional forces in addition to forces generated by interacting with the pegs (due to the robot's undulatory motion). Due to the complexity of modeling frictional interaction between the robot and the ground, we used an engineered solution to generate a "feed-forward frictional model." Specifically, we add an offset term directly to F_σ in (10), i.e., $F_\sigma = J\tau_{\text{ext}} - F_{\text{offset}}$, where the magnitude of F_{offset} was selected so that the mechanism maintained its nominal shape during unconstrained locomotion.

The experimental testbed used to compare the different controllers consisted of the snake robot and the peg board shown in Figure 3b. The pegs were placed in an approximately random pattern. The robot was covered with a braided polyester expanding sleeve to reduce friction, with nine 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).

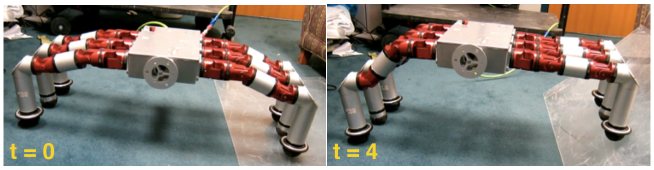

Fig. 9: Hexapod robot anti-compliantly adapting to variation in slope. A video of this behavior appears in the supplemental material.

For each trial, the position of the robot was tracked until either two minutes passed or the front half of the robot left the convex hull of the pegs. Five trials were conducted for each controller, with the same wave parameters ($\eta = 1.5$, $A_0 = 1$, $\omega = 1$) and controller constants held fixed.

The robots were compared by a metric that divided the number of cycles, which is the number of times a nominal wave propagated from the robot's head to its tail, by the total distance traveled during a trial, measured in terms of the integral of arc length in meters. This metric provided a measure of how many cycles it took for the robot to travel one meter. Controllers that result in the robot frequently thrashing in place therefore have a high value, and controllers that allow the robot to move smoothly through the pegs have a low value.

Figure 10 presents data associated with four experiments that tested implementations of shape-based compliant control (labeled "Single Amplitude" through "Nine Windows") and two experiments that implemented variants of CPG-based control. The data for each of the five trials conducted in each experiment is represented by the black horizontal bars, where the average performances for each experiment is represented by the corresponding red bar.

The four different shape-based compliant control experiments, for which data is presented in Figure 10, varied the number of spatial activation windows used to define the shape space in each case. "Single amplitude" corresponds to a centralized shape controller where the amplitude parameter was compliantly controlled over the entire body of the robot as it traversed the pegs. Following the example in Section IV-B1, "three windows" corresponds to placing three moving windows at the points of highest curvature on the underlying waveform $\sin(\eta s - \omega t)$. In the "six windows" experiment, six spatial activation windows were placed at the points of highest curvature as well as points of zero curvature of the underlying $\sin(\eta s - \omega t)$ waveform, starting from the head. The "nine windows" experiment corresponds to assigning independent, but spatially traveling, spatial activation windows for each joint (the robot contained nine in-plane joints). The data in Figure 10 shows the general trend that up to a certain point, as the number of spatial activation windows was increased, the robot's performance also increased, as the time taken to traverse the pegs decreased.

The data for the two CPG experiments represented in Figure 10 was collected using modifications of the CPG model presented in [7]. This CPG approach represents a decentralized control scheme. For the "CPG" experiment, the algorithm in

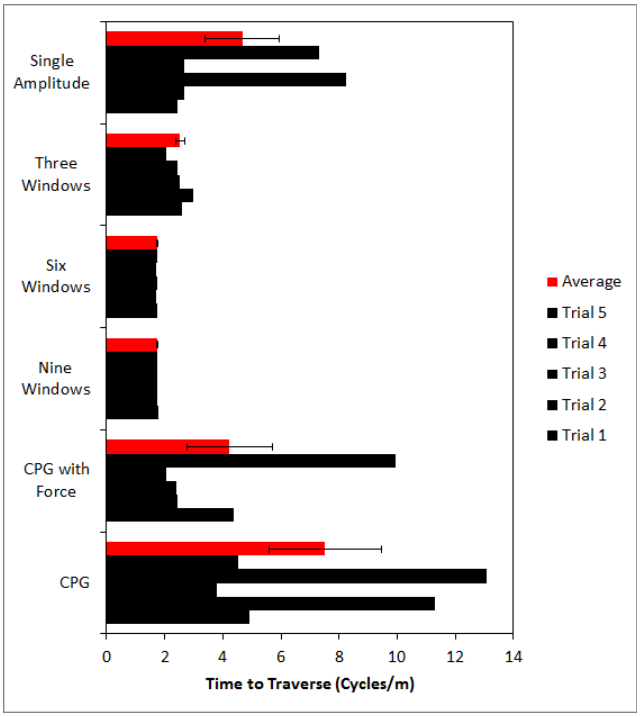

Fig. 10: Comparison of control methods using snake robot traversing the randomized peg array in Figure 3. The black bars represent individual trials and the red bars the mean, with error bars included, for each experiment.

[7] was modified to include estimated values of the individual amplitude parameters in each oscillator model (using a simple fitting procedure). Additionally, only a single oscillator was used for each degree of freedom. The fundamental frequency parameters for each oscillator were all assumed to be equal. In the "CPG with force" experiment, the CPG model in [7] was modified such that the stabilizing amplitude dynamics in the oscillator equations were augmented to include an "amplitude force" equal to that used in the nine-window shape controller, i.e., $F_\sigma = J\tau_{\text{ext}} - F_{\text{offset}}$. The resulting amplitude dynamics for the forced CPG controller were thus equivalent to those used in the shape-based compliant control implementation. The only practical difference between the forced CPG control and the nine window shape-based compliance was that the shape-based controller used traveling spatial windows where the CPG approach had windows fixed at the joint positions. We note that each of these two approaches are related to the simulated results in [11], but where admittance controllers in shape space are used instead of a more conventional impedance controller.

The CPG experiments showed that in the case where force measurements were not included, the CPG-based controller did significantly worse than each of the compliant control implementations. When force measurements were included, the CPG approach did better but was still outperformed by each of the shape-based controllers that included spatial activation windows, i.e., "three" through "nine windows."

VII. Conclusion

This work put forth the idea that shape is an important component in creating a middle layer that links high-level motion planning to low-level control in the robust locomotion of articulated systems in complex terrains. Shape functions were introduced as the layer of abstraction that formed the basis for creating this middle layer. Shape functions for two different systems, with very different nominal locomotive modes, were derived and subsequently used to define shape-based compliant controllers, highlighting the generality of this approach.

The general framework for shape-based compliant control enabled a snake-like robot to traverse random peg arrays and unknown rock piles, and enabled a hexapod platform to autonomously adapt to variations in slope. These robust behaviors were made possible using model-free controllers that relied only on the ability to sense joint torques and on the existence of shape functions.

The experimental results in Section VI present evidence for several preliminary conclusions that can be made with respect to the application of shape-based compliant control to snake-like robots. First, adding shape compliance in general improves performance. Second, the degree to which shape-based control is decentralized increases performance and reduces uncertainty, but only up to a certain point. This suggests that the additional complexity associated with adding new parameters to account for each possible degree of freedom may not be necessary. Third, the comparison of the "nine window" shape controller to the "CPG with force" controller, shows that there are measurable increases in performance when shape information is spatially propagated.

In the future, the framework for shape-based compliant control will be extended in two primary directions: 1) automatic shape function generation and 2) adaptive shape-parameter modulation. Section V presented a generalizable method for extracting shape functions, but we would like to have a method that truly automates this process, starting from either a generic kinematic specification of a robot or from biological data. We will also develop methods that adaptively tune the nominal shape of different mechanisms to different environments. The compliant shape modulation presented in this work produces robust locomotion, but still requires the nominal parameters to be reasonably tuned for a given platform in a given environment. By enabling platforms to adaptively learn the best shape parameters for an environment as they move through it, the role of shape-based compliant control can be expanded beyond serving as the middle layer between planning and control. In effect, adaptively learning nominal shape parameters will enable systems to autonomously modify motion plans in addition to robustly connecting the plans to lower-level controllers.

References

[1] *Biologically Inspired Robots*. Oxford University Press, 2003.

[2] G. Chirikjian and J. Burdick. A modal approach to hyper-redundant manipulator kinematics. *IEEE Transactions on Robotics and Automation*, 10:343–354, 1994.

[3] S. Gay, J. Santos-Victor, and A. Ijspeert. Learning robot gait stability using neural networks as sensory feedback function for central pattern generators. In *IEEE International Conference on Intelligent Robots and Systems*, 2013.

[4] N. Hogan. Impedance control: An approach to manipulation. In *American Control Conference*, 1984.

[5] N. Hogan and D. Sternad. Dynamic primitives in the control of locomotion. *Frontiers in Computational Neuroscience*, 7(71):1–16, 2013.

[6] A. J. Ijspeert. Central pattern generators for locomotion control in animals and robots: A review. *Neural Networks*, 21:642–653, 2008.

[7] A. J. Ijspeert and A. Crespi. Online trajectory generation in an amphibious snake robot using a lamprey-like central pattern generator model. In *IEEE international conference on robotics and automation*, 2007.

[8] A. J. Ijspeert, J. Nakanishi, H. Hoffmann, P. Pastor, and S. Schaal. Dynamical movement primitives: Learning attractor models for motor behaviors. *Neural Computation*, 25:328–373, 2013.

[9] F. Matsuno and K. Mogi. Redundancy controllable system and control of snake robots based on kinematic model. In *IEEE Conference on Decision and Control*, 2000.

[10] F. Matsuno and K. Suenaga. Control of redundant 3d snake robot based on kinematic model. In *IEEE International Conference on Robotics and Automation*, 2003.

[11] B.T. Mirletz, R.D. Quinn, and V. SunSpiral. Cpgs for adaptive control of spine-like tensegrity structures. In *IEEE International Conference on Robotics and Automation*, 2015.

[12] J. Ostrowski and J. Burdick. Gait kinematics for a serpentine robot. In *IEEE International Conference on Robotics and Automation*, 1996.

[13] Christian Ott, R. Mukherjee, and Y. Nakamura. Unified impedance and admittance control. In *IEEE International Conference on Robotics and Automation*, 2010.

[14] G. Pratt and M. Williamson. Series elastic actuators. In *International Conference on Intelligent Robots and Systems*, 1995.

[15] P. Prautsch and T. Mita. Control and analysis of the gait of snake robots. In *IEEE International Conference on Control Applications*, 1999.

[16] D. Rollinson and H. Choset. Gait-based compliant control for snake robots. In *IEEE International Conference on Robotics and Automation*, 2013.

[17] S. Schaal. Dynamic movement primitives - a framework for motor control in humans and humanoid robots. In *The International Symposium on Adaptive Motion of Animals and Machines*, 2003.

[18] B. Siciliano, L. Sciavicco, L. Villani, , and G. Oriolo. *Robotics: Modelling, Planning and Control*. Springer Publishing Company, Incorporated, 1st edition, 2008.

[19] M. Tanaka and F. Matsuno. Control of 3-dimensional snake robots by using redundancy. In *IEEE International Conference on Robotics and Automation*, 2008.

[20] 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*, 23:1131–1158, 2009.

[21] D.P. Tsakiris, M. Sfakiotakis, A. Menciassi, G. La Spina, and P. Dario. Polychaete-like undulatory robotic locomotion. In *IEEE International Conference on Robotics and Automation*, 2005.

[22] H. Yamada and S. Hirose. Steering of pedal wave of a snake-like robot by superposition of curvatures. In *IEEE International Conference on Intelligent Robots and Systems*, 2010.