Inferring Distributions of Parameterized Controllers for Efficient Sampling-Based Locomotion of Underactuated Robots

Raghu Aditya Chavali[1], Nathan Kent[2], Michael E. Napoli[2],
Thomas M. Howard[2], Matthew Travers[1]

***Abstract*— Sampling-based motion planning algorithms provide a means to adapt the behaviors of autonomous robots to changing or unknown a priori environmental conditions. However, as the size of the space over which a sampling-based approach needs to search is increased (perhaps due to considering robots with many degree of freedom) the computational limits necessary for real-time operation are quickly exceeded. To address this issue, this paper presents a novel sampling-based approach to locomotion planning for highly-articulated robots wherein the parameters associated with a class of locomotive behaviors (e.g., inter-leg coordination, stride length, *etc.*) are inferred in real-time using a sample-efficient algorithm. More specifically, this work presents a data-based approach wherein offline-learned optimal behaviors, represented using central pattern generators (CPGs), are used to train a class of probabilistic graphical models (PGMs). The trained PGMs are then used to inform a sampling distribution of inferred walking gaits for legged hexapod robots. Simulated as well as hardware results are presented to demonstrate the successful application of the online inference algorithm.**

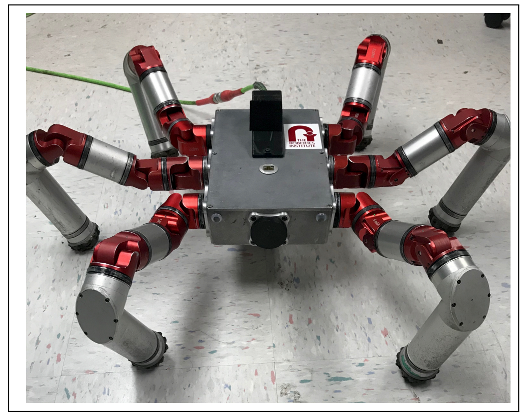

Fig. 1. Hexapod robot on which the efficient sampling-based planning framework developed in this work is demonstrated.

I. INTRODUCTION

Autonomous systems currently suffer from an inability to safely control, or rather adapt, their behavior to achieve high-level goals in unstructured environments. In light of these limitations, one potential means to achieve the level of adaptation necessary for autonomous systems to successfully operate in unstructured environments is through sampling-based planning techniques (this potential is well documented in the motion planning literature). However, conventional sampling-based approaches tend to be computationally inefficient when sampling in high dimensional spaces, e.g., sampling the optimal parameters, relative to some high-level locomotive objective, for highly-articulated robots moving through uneven terrains. This work thus presents a new approach to sampling-based navigation planning for highly-articulated robots wherein a class of probabilistic graphical models is used to dramatically limit the effective size of the search space. More specifically, we show how to encode sets of features related to the kinematics, task objective, and environmental context in a PGM that is used within an online sampling-based inference algorithm to efficiently determine optimal motion parameters for underactuated robots moving through nontrivial terrains.

[1] Raghu Aditya Chavali and Matthew Travers are with the Biorobotics Lab, Carnegie Mellon University, 5000 Forbes ave, Pittsburgh, PA 15213, USA. <rchavali, mtravers> @andrew.cmu.edu

[2] Nathan Kent, Michael E. Napoli, and Thomas M. Howard are with the Robotics and Artificial Intelligence Laboratory, University of Rochester, Joseph C Wilson Blvd. Rochester, NY 14627, USA.<nkent2, mnapoli>@cs.rochester.edu, <thoward>@ece.rochester.edu

The hexapod robot shown in Fig. 1 provides an example of the class of articulated robots considered in this work. To determine an "optimal behavior" for such a robot using a conventional sampling-based planning technique would require exhaustively searching a very large dimensional space (at minimum forty-eight dimensions). The resulting planning algorithm is therefore extremely time consuming and thus not applicable to online implementations. However, given some a priori knowledge that determines regions of the search space are "most relevant" to the hexapod achieving high-level goals has the potential to dramatically reduce the effective size of the search space and thus lead to a solution that does run in real-time.

This work quantifies what regions of the search space for a highly articulated robot are most relevant to the system achieving high-level objectives using an offline framework for generating a library of optimal locomotive behaviors. The behaviors are represented in terms of parameterized CPGs that implicitly define cyclic locomotive behaviors, i.e., locomotive gaits. For a given high-level objective, the CPG-based behaviors are learned using a gradient-free, genetic algorithm that determines the optimal set of CPG parameters for a given robot in a specific environment.

The main contribution of this work uses the offline-learned CPG-based behaviors to train a graphical model that encodes features describing a system's kinematics, a high-level objective, and environmental description (this work manually defines the obstacle dimensions and does not use an onboard sensor). The trained PGM is then used to learn how to efficiently sample from the underlying distribution of

CPG parameters during online implementation. We present experimental results that show the framework developed in this work outperform traditional sampling-based techniques.

The remainder of this paper is organized as follows. In Section II, we present related works with an emphasis on managing the behaviors of highly-articulated, underactuated systems. In Section III, we present a high-level overview of the automated framework for online sampling based inference and the probabilistic inference framework developed for this approach. Section IV describes the methods applied for evaluating the performance of the proposed framework. Section V presents results that show how our framework compares to traditional sample based techniques for inferring the motion parameters of articulated robots moving through semi-complex terrains. Finally, in Section VI, we conclude with a discussion of the limitations and future scope of the work presented in this paper.

II. BACKGROUND AND RELATED WORKS

A brief background on the different components that form the foundation of this work are incorporated to provide context.

A. Central Pattern Generator

Central Pattern Generators (CPGs) are a tool for controlling the motion in highly-articulated systems [2]. CPGs are defined as a set of coupled ordinary differential equations that model the interconnected relationship between different degrees of freedom in such systems. For example, for an n-legged articulated robot with two rotary shoulder joints per leg (as the motions of subsequent joints can be calculated using forward kinematics), let $x(t) = [x_1(t), ..., x_n(t)]$ represent the joint angles of the shoulder joints of each leg in the axial plane (proximal), and $y(t) = [y_1(t), ..., y_n(t)]$ represent the joint angles of the shoulder joints in the sagittal plane (middle joint). Given these assumptions, the CPG model used in this work (as presented in [5]) is defined by

$$\dot{x}_i(t) = -\omega\ \partial H_{yi} + \gamma(1 - H(x_i(t), y_i(t)))\ \partial H_{xi} \quad (1)$$

$$\dot{y}_i(t) = \omega\ \partial H_{xi} + \gamma(1 - H(x_i(t), y_i(t)))\ \partial H_{yi} + (\lambda \Sigma_j K_{ij} y_j(t)) \quad (2)$$

where ω is the speed of each oscillator's phase, γ represents the "forcing" to the CPG cycle, λ represents the strength of inter-oscillator coupling, and K defines the coupling matrix that is used to determine the phase relationship between oscillators. The parameter H here defines a super-elliptical shape of the oscillator wherein

$$H_c(x, y) = \left|\frac{x - c_x}{a}\right|^n + \left|\frac{y - c_y}{b}\right|^n \quad (3)$$

and $\partial H_\theta = \frac{\partial H}{\partial \theta}(x_i(t), y_i(t))$ where $\theta \in xi, yi$ in Equation 1. Note that, a and b represent the semi-minor and semi-major axes of the elliptical cycle respectively and n governs the overall shape of the oscillator trajectory (e.g., n = 2 results in a round elliptical shape and n = 4 a rectangular shape). An elliptical shape is chosen as this family of limit cycles gives us independent control over step height, step length and the shape of the cycle by varying the constants in Equation 3. Lastly, c_x and c_y are constant offset parameters that determine the nominal posture of the system for stable locomotion (As shown in Fig. 1, the front and back legs are spaced away from the centre of the robot to give it a stable grip).

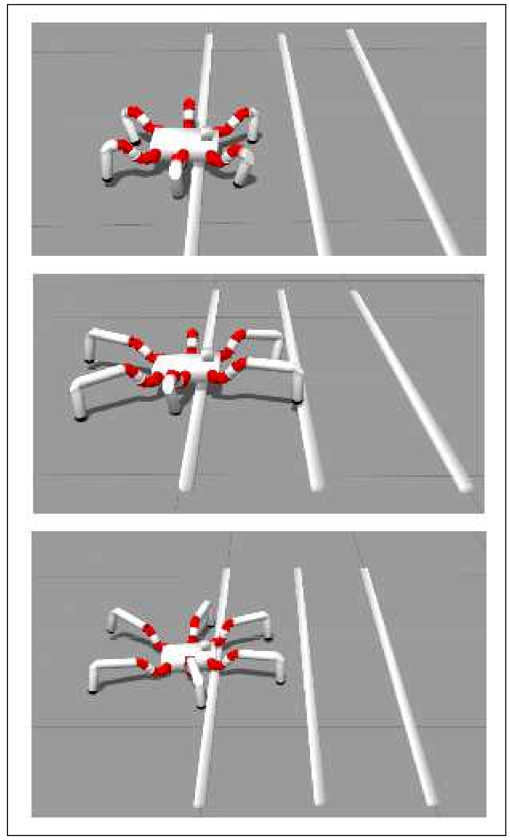

Fig. 2. The figure shows three different kinematic configurations of the simulated hexapod used for learning CPG parameters. The top picture shows the normal configuration of the hexapod. The middle picture shows an symmetric hexapod with four long legs and two short legs. The bottom picture shows the tall configuration of the hexapod.

Variations of such models are used in the bio-inspired [3] control of multi-legged robots [4], such as quadrapeds [9], swimming lamprey [6] and amphibious snake robots [7][8]. Works presented by Sartoretti et. al[5] and Righetti et.al[10] incorporate the use of sensory and intertial feedbacks for stabilizing locomotion on legged systems. Though previous work in the CPG literature does consider augmenting CPG models to adapt the underlying behavior of a given system, these prior works have attempted to hand craft methods that incorporate feedback into CPG models and do not account for adapting the motion parameters in real time based on uncertainty. In this work, we develop a novel means to adapt CPG-based locomotion models through efficient sampling-based methods made available by leveraging novel PGMs.

B. Probabilistic Graphical Models

For robots to reliably perform difficult tasks in unstructured environments they need to be able to reason about many different forms of uncertainty. PGMs permeate nearly all aspects of robotic intelligence and serve as one of the primary means to make autonomous decisions under uncertainty. For example, in various motion planning algorithms PGMs span sampling-based techniques for representing the state-action space [14], to those that explicitly reason about the uncertainty of transitions [15] and observations [16] for localization [17] and mapping[18] [19] tasks.

Since PGMs can be used to efficiently perform inference by naturally exploiting conditional independence to limit underlying complexity, this basic concept has divers application in the context of Natural Language Processing (NLP). Methods such as Generalized Grounding Graph [13], Distributed Correspondence Graph [1], Hierarchical Distributed Correspondence Graph [11] and Adaptive Distributed Correspondence Graph [12] build graphical models that independently infer distributions of symbols that represent objects, spatial relationships, constraints, trajectories, etc. for individual phrases forming the natural language statement. The distributed corresponding graph assumes conditional independence across constituents of the symbolic representation to efficiently infer distributions of symbols. In this work, we draw inspiration from [1], [11] in recognizing that the mathematical framework for efficiently inferring distributions of symbols for natural language understanding can be applied to the problem of inferring CPG parameters for locomotive behaviors of highly articulated robots moving through complex terrains.

A type of graphical model that is closely related to this work is Partially Observable Markov Decision Processes (POMDPs) serve as a useful framework for decision-making tasks under uncertainity. However, they tend to be computationally infeasible to solve directly. Joelle Pineau et. al [25], Sreenath et.al[28] and Simbro et. al [27] present POMDP based approaches to large-scale problems, but restrict its useage as a high level planner due to its computational cost, requiring a low level non probabilistic planner for robot control. Fuko et.al, [26] presents a novel hierarchical POMDP algorithm for autonomous robot nagivation in real time. However, the authors do not discuss how these methods generalize across systems with different configurations, one of the problems that is directly addressed in this work.

III. TECHNICAL APPROACH

Figure 3 highlights the different components of the framework for inferring locomotive behaviors for articulated robots developed in this work. First, an offline data set consisting of CPG-based behaviors training collected across robots with varying kinematics in different environments. Specifically, we learn in an offline framework the constant values a (the step height), b (the sweep) and coupling matrix K for each of the legs of the platform considered using a state genetic learning algorithm [22]. To evaluate an individual in the population, the CPG parameters (a,b,K) were simulated for a short time, with the reward being calculated as

$$\text{Reward}(x_{dist}, y_{dist}) = y_{dist} - 0.5\sqrt{|x_{dist}|} \tag{4}$$

where x_{dist} and y_{dist} are the distance travelled in the X and Y directions in the world frame. This reward function rewards forward progress whilst penalizing weaning off course. However, It should be noted that the magnitude of reward is greater than the penalty. The collection of training data forms the offline behavior library. This data is then used to train the graphical model used to reduce the complexity of online, sampling-based inference. The PGM training process involves encoding sets of features containing the behavior parameters for the different kinematic configurations of the hexapod robot, the environment and the end goal into a factor graph. These encoded feature sets allow the graphical model to learn how to sample from the otherwise high dimensional search space during run-time.

Leveraging recent work in probabilistic graphical models for natural language understanding of robot instructions [24], [23], we propose a factor graph to infer a set of CPG parameters ($\mathcal{P}$) from a space of CPG parameters $\mathbf{P}$ conditioned on random variables that represent behavior ($\mathcal{B}$), environment ($\mathcal{E}$), and model ($\mathcal{M}$). The distribution over behaviors represents uncertainty in what activity should be performed from a task planning framework. The distribution over environments models the sensor noise and state estimation inaccuracy. The distribution over models represents the uncertainty in internal parameters or structure of the robot model (e.g., range of motion of joints, motor performance), i.e.,

$$\mathcal{P}^* = \underset{\mathcal{P}\in\mathbf{P}}{\arg\max}\; p\left(\mathcal{P}|\mathcal{B},\mathcal{E},\mathcal{M}\right) \tag{5}$$

We model this distribution as a factor graph with known binary correspondence (ϕ), behavior, environment, and task variables and unknown parameter variables. As described [24], a binary correspondence variable with a "true" logic value models the probability of a symbol in the context of the other random variables. We reformulate the expression described in Equation 5 as a factor graph with known binary correspondence variables, i.e.,

$$\mathcal{P}^* = \underset{\mathcal{P}\in\mathbf{P}}{\arg\max}\; p\left(\phi = true|\mathcal{P},\mathcal{B},\mathcal{E},\mathcal{M}\right) \tag{6}$$

The primary issue with this formulation is that we are sampling parameters from a distribution $\mathbb{R}^n$, where n is the number of free parameters in the CPGs that describe the motion of the underactuated system. Using the insight made available by Distributed Correspondence Graphs [23] for approximating models with large symbolic representations, we assume conditional independence across constituents of the symbolic representation to make inference tractable, i.e.,

$$\mathcal{P}^* = \underset{p_1\dots p_n\in\mathbf{P}}{\arg\max}\prod_{i=1}^{|\mathrm{N}|} p\left(\phi_i = true|p_i,\mathcal{B},\mathcal{E},\mathcal{M}\right) \tag{7}$$

As in [24], [23] we model the expressions for conditional probability distributions using log-linear models that learn features weights from a set of annotated examples (environment, behavior, model, and learned parameters) obtained from the genetic algorithm. A graphical representation of the factor graph is illustrated in Figure 4. We approximate the

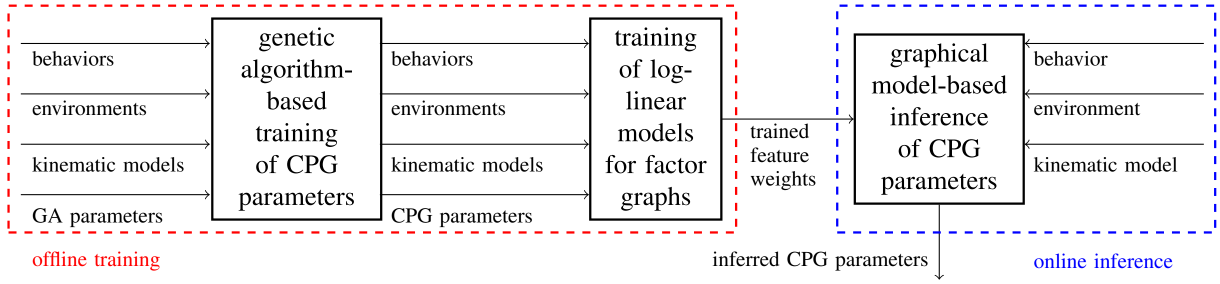

Fig. 3. The automated framework showing the offline (red box) and the online inference (blue box) modules. The genetic algorithm-based training of CPG parameters learns an optimal set of CPG parameters for every sampled behavior, environment, and kinematic model. Those examples are collected to train feature weights for log-linear models in factor graph-based probabilistic graphical models. At inference time, a factor graph infers a distribution of most likely CPG parameters given the current behavior, environment, and task.

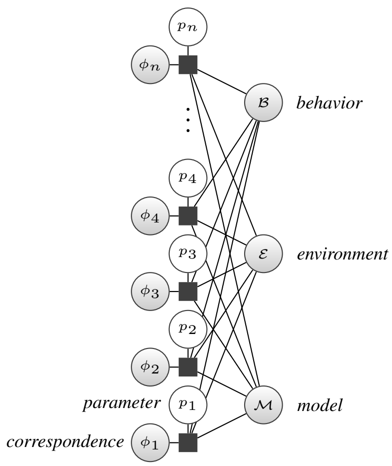

Fig. 4. A representation of the factor graph used for probabilistic inference of CPG parameters. The model exploits conditional independence across constituents of the parameterization to infer a distribution of CPG parameters ($\mathcal{P}$) using random variables that represent environment ($\mathcal{E}$), behavior ($\mathcal{B}$), and model ($\mathcal{M}$).

inferred distribution of CPG parameters from this model at run-time using beam search and use the most likely parameters to inform our sampling of candidate CPG parameters for performing the given task. This is done by modeling the conditional probability in Equation 7 as a learned function, i.e.,

$$\mathcal{P}^* = \underset{p_1 \dots p_n \in \mathbf{P}}{\arg\max} \prod_{i=1}^{|\mathrm{N}|} f\left(\phi_i = true, p_i, \mathcal{B}, \mathcal{E}, \mathcal{M}\right) \tag{8}$$

IV. EXPERIMENTAL DESIGN

To evaluate the performance of the proposed model for inferring CPG parameters, we follow the procedure outlined in Figure 3 and vary parameters of the kinematic model and environment. The genetic algorithm-based training procedure varied the obstacle step height from 0m to 0.05m in increments of 0.01m. The procedure also varied the kinematic parameters $0.05\text{m} < L_{x1} < 0.1\text{m}$, $0.0191\text{m} < L_{x2} < 0.03\text{m}$, and $0.1206\text{m} < L_{x3} < 0.24\text{m}$, where L_{xi} represents the i^{th} joint of the x^{th} leg. The training procedure produced twenty-seven learned sets of CPG parameters. Among these sets thirteen examples were symmetric and had L_{x1}, L_{x2}, and L_{x3} values of 0.05m, 0.0191m, and 0.1206m respectively. We will refer to this model as the "normal" model which is illustrated in Figure 1 . Seven of the examples had models that were symmetric and had longer L_{x1}, L_{x2}, and L_{x3} values of 0.1m, 0.03m, and 0.24m respectively. We will refer to this model as the "tall" model. Five more of these models were based on the longer configuration but had shorter middle legs with L_{x1}, L_{x2}, and L_{x3} values of 0.05m, 0.0191m, and 0.1206m respectively. The final two configurations were based on the first (symmetric) configuration with long middle legs with L_{x1}, L_{x2}, and L_{x3} values of 0.1m, 0.03m, and 0.24m and an asymmetric configuration with three normal length legs of L_{x1}, L_{x2}, and L_{x3} values of 0.05m, 0.0191m, and 0.1206m and three long length legs of 0.1m, 0.03m, and 0.24m. A behavior describing a locomotion tasks was assumed for all examples. The genetic algorithm training assumed population sizes that ranged from 50 to 100 and trained of 10 to 40 generations.

From these learned parameter values we define a search space of twenty-one uniformly sampled values of forty-eight CPG parameters (one a, one b, and six k parameters for each of the six legs, as defined in Equations 1 and 3) across the ranges $0.0 \leq a_i \leq 4.5$, $0.0 \leq b_i \leq 2.5$, and $-3.5 \leq k_{ij} \leq 3.5$. For each example we label the sampled CPG parameter with the value that is closest to the trained model as a "true" correspondence and all others as a "false" correspondence. From these twenty-seven trained CPG parameter values we accrue 27,216 training examples that we use to train a log-linear model with 2,280,960 weighted features. As in [24], [23], the feature vector is generated by a Cartesian product of 362 binary features divided into four groups composed of two correspondence features, 180 CPG parameter value features, 48 CPG parameter property features, and 132 model and environment property features.

To analyze the performance of the graphical model, the CPG parameters were inferred for two simulated variations of the hexapod kinematic chain. In these experiments the "normal" and "tall" configurations were evaluated. The CPGs were inferred by searching the factor graph using beam search with a beam width of 8, resulting in the 8 samples with the highest approximated log-likelihood. For each of

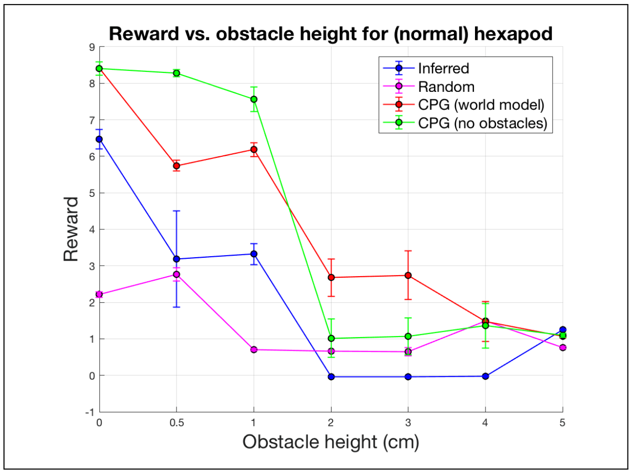

Fig. 5. Performance of nominal hexapod locomotion using the three baseline approaches and graphical model to select CPG parameters. The error bars represent 95% confidence intervals around the median. Both factors (CPG generation method, obstacle height) and their interaction were found to be statistically significant ($p < 0.01$) using an N-way Analysis of Variances (ANOVA).

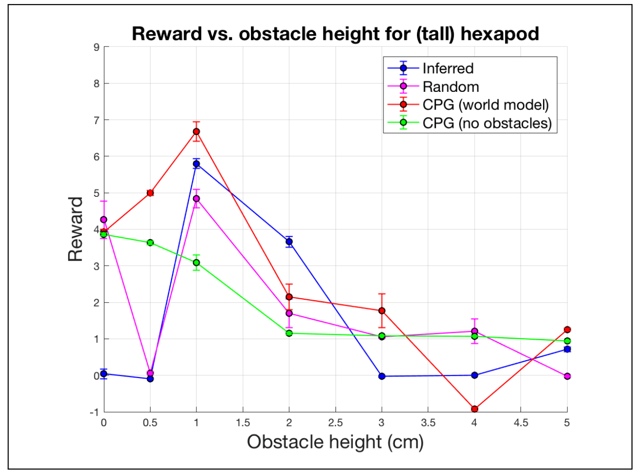

Fig. 6. Performance of the "tall hexapod" locomotion using the three baseline approaches and graphical model to select CPG parameters. The error bars represent 95% confidence intervals around the median. Both factors (CPG generation method, obstacle height) and their interaction were found to be statistically significant ($p < 0.01$) using an N-way Analysis of Variances (ANOVA).

the 8 beam samples, 8 simulations were conducted and the reward function from Equation 4 was evaluated. The median of the 8 resulting reward values for each beam sample was calculated and the beam sample with the highest median was chosen as the best inferred solution for that combination of kinematic chain and obstacle height.

Three baseline CPG approaches were used for comparison with the graphical model. The first baseline is the set of CPG parameters obtained by the genetic algorithm for each combination of kinematic chain and obstacle height, which is essentially the training data used to learn the graphical model. The second baseline is set of CPG parameters obtained by the genetic algorithm for the appropriate kinematic chain, but trained in an obstacle free environment. This explores the effectiveness of CPGs learned in free space when they are applied to a world with obstacles. The third baseline is a randomly generated set of CPG parameters sampled from a multivariate uniform distribution with ranges that span the values obtained by the genetic algorithm for each parameter. This baseline illustrates the difficulty of the sampling space and represents an uninformed sampling method applied to the CPG parameter distribution. During the random sampling procedure, the coupling matrix was constrained to be symmetric with all diagonal elements equal to 0. Obstacles heights were varied using the set of values $0.0\text{m}, 0.005\text{m}, 0.01\text{m}, 0.02\text{m}, 0.03\text{m}, 0.04\text{m}, 0.05\text{m}$ and both the normal and tall kinematic chains were used. For each baseline, 8 simulations were performed for each combination of obstacle height and kinematic chain. To analyze the randomly generated CPGs consistently with the graphical model, 8 random CPGs were generated and 8 simulations were performed. The random CPG set with the highest median reward was chosen as the best randomly generated solution.

V. EXPERIMENTAL RESULTS

This section presents the experimental results of this work. Figure 5 presents our initial results where the red line shows the locomotion of the hexapod using the CPG parameters trained on flat ground tested on different obstacle heights, and the blue line corresponds to the hexapod's locomotion using the CPG parameters trained on the appropriate obstacle height. These results indicate that factors corresponding to choosing the appropriate CPG parameters and the obstacle height are statistically significant.

Figure 8 shows the hardware results of the hexapod trying to cross obstacles of height 5cm using the CPG parameters for navigating on flat ground. As expected, the robot fails in achieving its task. Figure 9 shows the hardware results of the hexapod trying to cross obstacles of height 5cm using the learned CPG parameters corresponding to the object height. Here we see that the hexapod is easily able to maneuver past the obstacles in the same environment. Figure 5 and Figure 9 motivate learning CPG parameters for a world model.

Figure 7 show the hardware results of the hexapod trying to locomote on flat ground using the CPG parameters obtained from the graphical model inference described in section III. The results show that the motion of the hexapod is subpar compared to the learned CPG parameters for the same task. However, the experimental results exhibits a new, never before seen gait for locomotion on flat ground.

In the experiments we also explored the run-time performance of beam search in the probabilistic graphical model used to infer CPG parameters. Assuming a beam width of eight and the trained features set and search space described in IV, we observed an average run-time of 0.075 seconds for probabilistic inference across the environments, kinematic models, and behaviors explored in the twenty-seven training examples generated from the genetic algorithm training procedure.

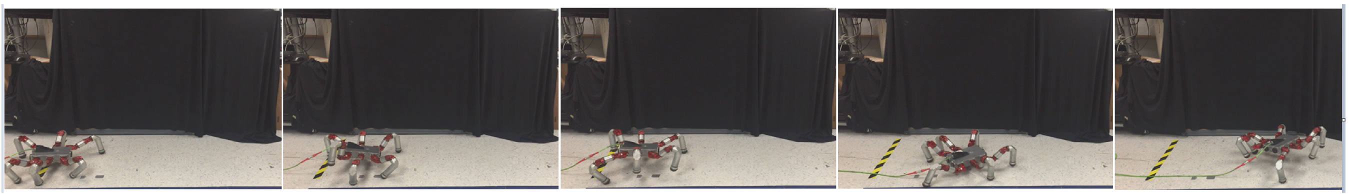

Fig. 7. The figure shows the hexapod attempting to locomote on flat ground using the CPG parameters obtained from probabilistic inference.

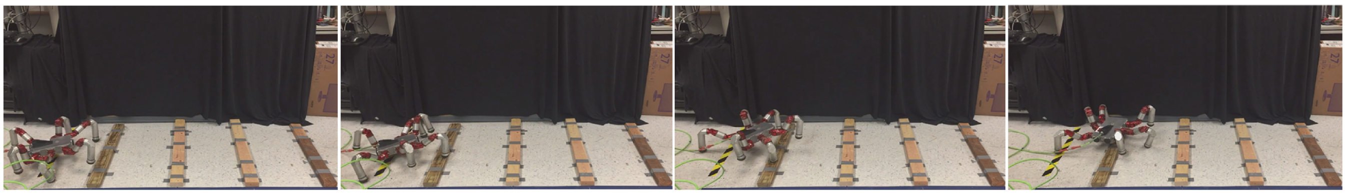

Fig. 8. The figure shows the hexapod attempting to cross obstacles using the CPG parameters for flat ground locomotion. We see that the robot is unable to make it past the first obstacle.

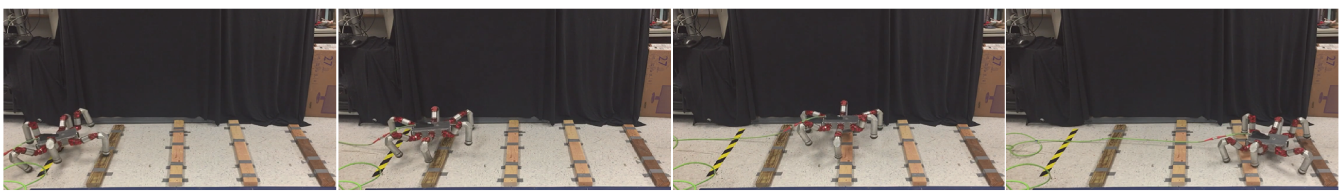

Fig. 9. The figure shows the hexapod attempting to cross obstacles using the CPG parameters learned for obstacle height 5cm to cross obstacles of height 5cm. We see that the robot easily makes past all the obstacles in front of it.

The results for the simulation experiments described in Section IV for the hexapod robot with the normal kinematic chain are shown in Figure 5. The results indicate that the CPG parameters learned from the genetic algorithm with a world model outperform the CPG parameters learned with no obstacles in environments with higher obstacle height. The results for the graphical model outperform the random sampling in cases with small to no obstacles. The results obtained by the genetic algorithm outperform both the inference and the random sampling approaches. The results for the simulation experiments described in Section IV for the hexapod robot with the tall kinematic chain are shown in Figure 6. The results indicate that, although in some cases the models were able to successfully locomote, in most cases the methods were unable to consistently learn across all obstacle heights.

VI. CONCLUSIONS

This paper presents a new approach to sampling-based inference techniques through the use of probabilistic graphical models to adapt the search space for sampling-based locomotion. We show how information acquired during the offline training of a probabilistic graphical model can be used to learn models for inferring CPG parameters from behavior, environment, and kinematic model. The learned graphical model that encodes parameters for all examples of kinematic models, behaviors, and environments explored during the offline training procedure and outperforms random sampling at low obstacle heights for the most common kinematic configuration and approaches the performance of the training example that was tuned for a specific obstacle height and kinematic configuration. In ongoing work, we are diversifying and expanding the training examples to more effectively sample from the space of kinematic models, environments, and behaviors and introduce new features and/or models for approximating the conditional probability distributions encoded by factors in the graphical model.

This work opens up several interesting directions for future work. A model that is able to infer parameterized control primitives based solely on behavior, environment, and kinematic structure allows the model to extend to other applications and diverse robot platforms. We are particularly interested in exploring whether training examples from diverse platforms such as hexapods and humanoids can be encoded into a single graphical model. We are also interested in adapting this framework for reacting to changes in robot performance, such as failure of actuators or limited motion in joints. Enabling a robot to identify changes in its motion model and infer corrections to parameterized control primitives based on this information can improve the robustness of autonomous systems operating in non-trivial environments. Lastly, we seek to expand the complexity of the graphical model used for inferring CPG parameters to continuously estimate constituents of the random variables that represent behavior, environment, and kinematic model during the inference procedure.

ACKNOWLEDGMENT

This work was supported in part by the National Science Foundation under grants IIS-1724000 and IIS-1723972. The authors would like to thank Blake Buchanan for helping with the hardware experiments included in this work.

REFERENCES

[1] Howard, T.M, Tellex, S., Roy, N., A natural language planner interface for mobile manipulators, International Conference for Robotics and Automation (ICRA), 2014, pages 6652-6659.

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

[3] Arena, P., Fortuna, L., Frasca, M., Sicurella, G., An adaptive, self-organizing dynamical system for hierarchical control of bio-inspired locomotion. IEEE Transactions on Systems, Man and Cybernetics, Part B, 34(4),18231837. 2004.

[4] Inagaki, S., Yuasa, H., Arai, T. CPG model for autonomous decentralized multi-legged robot systemgeneration and transition of oscillation patterns and dynamics of oscillators. Robotics and Autonomous Systems,44(34),171179, 2003

[5] Sartoretti, G., Shaw, S., Lam, K., Fan, N., Travers, M., Choset, M., Central Pattern Generator with Inertial Feedback for Stable Locomotion and Climbing in Unstructured Terrain, ICRA, 2018.

[6] Arena, P., A mechatronic lamprey controlled by analog circuits, In Proceedings of the 9th IEEE Mediterranean conference on control and automation, 2009.

[7] Crespi, A., Ijspeert, A.J.,Online optimization of swimming and crawling in an amphibious snake robot. IEEE Transactions on Robotics ,24(1),7587, 2008.

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

[9] Fukuoka, Y., Kimura, H., Cohen, A. H. Adaptive dynamic walking of a quadruped robot on irregular terrain based on biological concepts. The International Journal of Robotics Research,34, 187202, 2003.

[10] Righetti, L., Ijspeert, A. J., Pattern generators with sensory feedback for the control of quadruped locomotion. In IEEE International Conference on Robotics and Automation (ICRA),pp. 819-824, 2008.

[11] Chung, I., Propp, O., Walter, M., and Howard, T.M., Performance of hierarchical distributed correspondence graphs for efficient symbol grounding of robot instructions, International Conference on Intelligent Robots and Systems IEEE/RSJ, 2015.

[12] Paul, R., Arkin, J., Roy, N., and Howard, T.M., Efficient grounding of abstract spatial concepts for natural language interaction with robot manipulators, Robotics: Science and Systems Conference, 2016.

[13] Tellex, S., Kollar, T., Dickerson, S., Walter, M.R., Banerjee, A.G., Teller, S., Roy, N., Understanding natural language commands for robotic navigation and mobile manipulation, Artificial Intelligence, San Francisco, CA, 2011.

[14] Kavraki, L., Svestka, P., Latombe, J.C., Overmars, M.H., Probabilistic roadmaps for path planning in high-dimensional configuration spaces, IEEE Transactions on Robotics and Automation,1996.

[15] Roy, N., Thrun, S., Motion planning through policy search, 2002.

[16] Littman, M.L., Cassandra, A.R., and Kaelbling, L.P., Learning policies for partially observable environments: Scaling up, International Conference on Machine Learning (ICML), 1995.

[17] Fox, D., Burgard, W., Dellaert, F., Thrun, S., Monte carlo localization: Efficient position estimation for mobile robots, AAAI Conference on Artificial Intelligence, 1999.

[18] Elfes, A., Sonar-based real-world mapping and navigation, IEEE Journal on Robotics and Automation, 3(3):249-265, 1987.

[19] Thrun, S., Montemerlo, M., Koller, D., Wegbreit, B., Nieto, J., Nebot, E., Fastslam: An efficient solution to the simultaneous localization and mapping problem with unknown data association, Journal of Machine Learning Research,2004.

[20] Kovar, L., Gleicher, M., Pighin, F., SIGGRAPH - Motion graphs, 2002

[21] Macallister, B., Butzke, J., Kushleyev, A., Pandey, H., Likhachev, M., Path planning for non-circular micro aerial vehicles in constrained environments, IEEE Conference on Robotics and Automation, 2013.

[22] Bck, T., Fogel, D.B., Michalewicz, Z., eds. Evolutionary computation 1: Basic algorithms and operators. Vol. 1. CRC press, 2000.

[23] Paul, R., Arkin, J., Aksaray, D., Roy, N., Howard, T.M., Efficient Grounding of Abstract Spatial Concepts for Natural Language Interaction with Robot Platforms, International Journal of Robotics Research, June 2018.

[24] Tellex, S., Kollar, T. Dickerson, S., Walter, M.R., Banerjee, A., Teller, S., and Roy, N., Understanding natural language commands for robotic navigation and mobile manipulation, Proc. of the Twenty-Fifth Conference on Artificial Intelligence, 2011.

[25] Pineau, J.,Thrun, S., Hierarchical POMDP Decomposition for A Conversational Robot, ICML Workshop on Hierarchy and Memory, 2001.

[26] Foka, A., Trahanias, P., Real-time hierarchical POMDPs for autonomous robot navigation, Robotics and Automation Systems, January 2007.

[27] Simbro-Cuaya, G., Angelica, . ADAPTIVE LOCOMOTION FOR A HEXAGONAL HEXAPOD ROBOT BASED ON A HIERARCHICAL MARKOV DECISION PROCESS, 2008, $10.1142/9789812835772_0085$.

[28] Sreenath, K., Hill, C., Kumar, V., A Partially Observable Hybrid System Model for Bipedal Locomotion for Adapting to Terrain Variations,Proceedings of the 16th international conference on Hybrid systems: computation and control, Pages 137-142