DESIGNING AND LEARNING CPG GAITS FOR SPHERICAL TENSEGRITY ROBOTS USING BAYESIAN OPTIMIZATION

BY COLIN M. RENNIE

A thesis submitted to the

School of Graduate Studies

Rutgers, The State University of New Jersey

in partial fulfillment of the requirements

for the degree of

Master of Science

Graduate Program in Computer Science

Written under the direction of

Dr. Kostas E. Bekris

and approved by

New Brunswick, New Jersey

January, 2018

ABSTRACT OF THE THESIS

Designing and Learning CPG Gaits for Spherical Tensegrity Robots Using Bayesian Optimization

by Colin M. Rennie

Thesis Director: Dr. Kostas E. Bekris

This thesis presents a framework for developing a library of gaits for highly non-linear, hyper-redundant, potentially compliant robotic systems. Examples of such systems that motivate this work include tensegrity robots, which combine both soft and rigid elements, and snake-like salamander robots. A library of gaits for such complex robots can be integrated with a search-based method so as to achieve more efficient exploration of the underlying state space when solving trajectory planning problems.

The first component of the work corresponds to the definition of a Central Pattern Generator (`CPG`) for a spherical tensegrity robot inspired by similar solutions in the domain of modular, bio-inspired snake robots. The `CPG` provides a reparametrization of the underlying system, which can easily result in the generation of rhythmic gaits. The second component is a novel framework for simultaneously discovering effective gaits along different directions of motion by searching the space of `CPG` parameters. The framework defines multiple objectives, which are maximized though a parallel Bayesian Optimization (`BO`) process. The samples, which correspond to different gait parameters, are biased towards areas of previously observed high reward using a set of binary `kNN` classifiers with on-line updates. This integrated method is shown to be more efficient than Monte Carlo sampling of gait parameters or `BO` without classification

or the classification only approach. The evaluation is performed in simulation using a high-dimensional spherical tensegrity robot.

Acknowledgements

I'd like to thank Dr. Kostas Bekris for giving me the opportunity to learn so much and to grow by being a part of his research lab for the last few years. I'd like to thank my labmates for their inspiration, enthusiasm, and also for their commiseration when the frustrations of research work arose.

I'd especially like to thank my family - my mother, father, and brothers - for being unflinchingly supportive of my decisions throughout life, and for trying their best to do everything they could to help me along the way. I'd also like to thank Annie for the same reasons and for her patience and friendship. Thank you for keeping me sane and supporting me in this journey.

This work was performed in collaboration with NASA Ames Research Center and was partially supported by NASA Grant Number NNX15AU47GVT.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

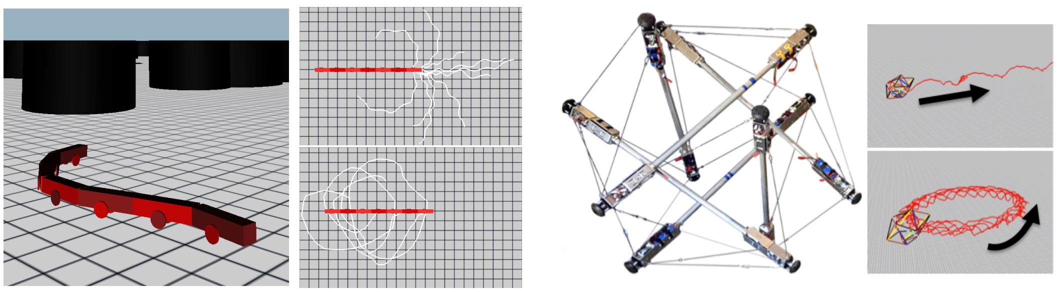

Figure 1.1: Left: The physically-simulated snake robot model used in these experiments alongside visualizations of the snake's head link when executing the 15 velocity-optimized gaits resulting from methods presented in this thesis, and when executing 15 randomly sampled 9-dimensional gaits. Right: SUPERBall tensegrity system prototype at NASA Ames, which motivates this work. In the current work, the goal is to develop a set of periodic, stable gaits with predefined motions. Demonstrated are two example gaits; one moving the center of mass along a straight line (top) and one resulting in a tight rotation.

A critical performance limitation in motion planning for many exciting redundant robotic systems is the computation needed to explore the underlying high-dimensional control space. This is true for many popular paradigms, such as search-based [Cohen et al., 2010] or sampling-based frameworks [Karaman and Frazzoli, 2011, LaValle, 1998]. These computational requirements are further increased when modeling the underlying dynamics requires frequent and expensive calls to a black-box forward propagation model. This is especially true when a system is modeled through a physics engine [Smith et al., 2005], which can express effects of contact forces and the dynamic interaction these forces can have on the rest of the system.

Such complex simulation tools are becoming increasingly popular in real world applications, where the underlying dynamics of the system are significant. For example, several teams competing in the DARPA Urban Challenge made use of precomputed trajectories in the form of constant curvature arcs, which were then used to plan dynamic trajectories online during the challenge [Rauskolb et al., 2008, Von Hundelshausen et al., 2008]. Beyond wheeled systems, complex dynamics also arise in locomotive systems or highly-redundant systems, such as snake robots [Wright et al., 2007]. Dealing with complex dynamics can be approached with search-based [Cohen et al., 2010] or sampling-based planners [LaValle, 1998], for which case asymptotically optimal methods have been recently developed [Li et al., 2016, Karaman and Frazzoli, 2011]. Alternatively, one can develop control policies through learning algorithms [Iscen et al., 2014, Lessard et al., 2015, Kulkarni et al., 2016]. Both paradigms address the control problem in its full capacity, typically requiring more computation than can be facilitated for real time planning.

Tensegrity structures, an example of the emerging paradigm of hybrid soft-rigid robots, are made up of sets of rigid elements which are interconnected with flexible, dynamic supports, such as cables. Allowing the lengths and resulting tensions of the dynamic elements to be dynamically controlled gives rise to a class of actuated tensegrity robots, which are able to dynamically adapt their global shape and rigidity. The shape deformation capability allows for locomotion as the system's center of mass (CoM) is shifted and the structure begins to move [Koizumi et al., 2012, Iscen et al., 2014]. Dynamic structural rigidity allows adaptation when interacting with different terrains [Skelton and de Oliveira, 2009], allowing contact forces to be distributed throughout the structure to varying extents, which improves the system's durability and adaptability to unpredictable contacts. The combination makes this class of systems ideally suited for exploring dynamic and unknown terrains, such as in the context of planetary exploration, where contact forces of all magnitudes may be encountered and must be absorbed by the system without hindrance to the trajectory of the robot [Caluwaerts et al., 2014, Mirletz et al., 2015].

Despite these desirable characteristics, tensegrity systems are not without their own set of challenges. They are generally of very high dimensionality both in terms of their state and control space, they are highly nonlinear, and exhibit complex contact dynamics. Furthermore, these are robots that do not have direct control of their position but only of their shape. These aspects complicate the process of developing robust methodologies for effective locomotion with such systems.

1.2 Summary of Contributions

This thesis develops a learning framework for developing a library of gaits for solving trajectory planning problems with a search-based planner. The learning framework explores the system's control space by sampling the parameter space of rhythmic gaits and examining their effects in an online manner, selecting each sequential control point to simulate based on the output of one of several optimization functions. The optimization functions are designed with the goal of searching the control space for those gaits which constitute a maximally expressive gait library, a notion closely related to that of path diversity [Branicky et al., 2008, Knepper and Mason, 2009]. Moreover, while maximizing these objectives, a second model is trained as a binary classifier, learning the probability that any given point in the control space will result in an effective gait. The framework then uses this probability to bias samples toward the control subspace which is likely to yield the most effective gaits with respect to the optimization functions. Through this biasing and the use of Gaussian Process-based Bayesian optimization learning models, a gait library generation process is achieved that is more sample-efficient than Monte Carlo evaluation. In particular, given a smaller set of samples it can achieve similar performance to a more exhaustive search process. This is demonstrated on a spherical tensegrity system with 12 DoF control space and a seven-link, physically-simulated snake robot using a Central Pattern Generator control parameterization [Ijspeert and Crespi, 2007, Buchli and Ijspeert, 2008] in simulation. The effectiveness of the resulting gaits to solve motion planning problems is additionally demonstrated on the snake system.

1.3 Overview of Thesis Presentation

The remainder of this thesis is presented as follows. In chapter 2, related work is presented relating to the three problem domains that this thesis touches on, and the contributions of this thesis pertaining to each are described. In chapter 3 and 4, the `CPG` designs for these systems and an outline of the parallel learning framework, respectively, are presented. Chapter 5 presents experimental results which validate the approach this thesis takes to the optimization problem, and emphasizes the usefulness of solving gait optimization problems when addressing motion planning problems for complex systems in unknown terrain. Lastly, in chapter 6, a summary of the presented findings is given, and potential future directions are briefly discussed.

Chapter 2

Related Work

Gait generation corresponds to designing curves in the base space of a robotic system, i.e., the space which represents the shape of the robot [Shammas et al., 2005]. A critical performance limitation in addressing this problem for many exciting redundant systems, including tensegrity robots, is the computational effort needed to explore the underlying control space. For instance, both discretizing the control space and searching it exhaustively or sampling random controls, will result in dimensionality challenges when exploring the control space of the 12-actuated tensegrity robot shown in Figure 1.1. The computational requirements are further increased when modeling the underlying dynamics requires frequent and expensive calls to a black-box forward propagation model [Tesch, 2013]. This is the case when a physics engine is required to express effects of contact forces and the dynamic interaction these forces can have on the rest of the system.

Prior related work has provided control policies for the target spherical tensegrity robot focused at removing the platform from craters [Lessard et al., 2015]. These policies were generated via Monte Carlo search combined with evolutionary algorithms. The focus of this thesis is to generate gaits that are more dynamic in nature and also to provide a data-efficient process for optimizing gaits given an appropriate parameterization of the robot. To the best of the author's knowledge, the proposed process in the current work has resulted in the fastest gaits that have been generated for the simulated version of the tensegrity system.

One idea in the literature for gait generation is to introduce a reduction process. Early work has shown that by taking advantage of symmetries it is possible in many cases to project the entire dynamics of a system onto the shape space of a robot [Walsh and Sastry, 1995, Ostrowski and Burdick, 1998, Ostrowski et al., 2000]. Analytical expressions can be defined in certain cases expressing the relationship between shape and position velocities giving rise to a non-linear control system. Unfortunately, however, such analytical expressions and controllers are difficult to define for systems as complex as tensegrities as it is a challenging task to define a mapping from shape space to joints when there exists global interconnectedness between actuated rods and cables. The ideas of reducing the dimensionality of the problem and introducing periodic control functions, however, remain important for generating gaits in these systems.

2.1 CPG System Design

Another direction for gait generation is biomimicry [Hirose, 1993, Wright et al., 2007, Hopkins and Gupta, 2014], as this type of locomotion via shape changes has roots in nature: biological snakes locomote by undulating their bodies, and many bacteria locomote by changing their shapes. For instance, robotic snake locomotion can be achieved by attaching passive wheels on the bottom of a chain of rigid links and forcing the shape of the chain to move along a serpenoid curve inspired by biological snakes.

The concept of Central Pattern Generators (`CPG`) as an oscillatory gait generation mechanism has been explored specifically for redundant, bio-inspired systems. There has been wide adaptation of the `CPG` framework in the control of snake robots [Sfakiotakis and Tsakiris, 2007, Herreo-Carron et al., 2011, Bing et al., 2017]. `CPG`s define sinusoidal system equations for variables, which define desired setpoints for the joints in a system and allow the setpoints to evolve in constant phase offset with one another [Ijspeert and Crespi, 2007]. The result, when coupled with a `PD` controller, is an oscillatory control function, which is parameterized by frequency, amplitude, and phase offset between the joints. Under this control structure, a given parameter setting defines a gait for the system, which will provide oscillatory control input even at steady state. Grid search of the parameter space has been performed in the past in order to identify single gaits corresponding to the most rapid displacement of a snake-like system [Crespi and Ijspeert, 2008]. Evolutionary algorithms have also been used in order to search the parameter space of `CPG`s [Oliveira et al., 2013]

The focus of this thesis is on inducing effective position and orientation change of a prototypical spherical tensegrity platform by changing its shape, which corresponds to the gait generation problem [Shammas et al., 2005]. The platform considered, known as SUPERBall, is shown in Figure 1.1. This is a system that has been built at NASA Ames for space exploration purposes and for which there is access to an open-source physics-based simulation model [Iscen et al., 2014]. This work proposes a parametrization for this platform that allows the generation of rhythmic gaits. Furthermore, an underlying data-efficient optimization method for generating effective gaits is defined, which is agnostic to the parametrization and the specific platform.

2.2 Multi-Objective Optimization

Given the ability to generate rhythmic motions for SUPERBall through a `CPG` architecture, the next step is to identify the parameters of the `CPG` that optimize desirable objectives. The proof-of-concept objectives considered in this work are designed with the goal of generating a library of fast, dynamic gaits that are able to move the center of mass (CoM) of SUPERBall in a variety of different directions. The motivation comes from work in the path planning literature on path diversity [Branicky et al., 2008, Knepper and Mason, 2009]. Once gaits are available that can move the CoM along different directions, they can then be utilized by a motion planner to explore an environment, which may contain obstacles (e.g., craters, rocks).

Optimization of these objectives is achieved by choosing a small set of target variables, i.e., forward and angular velocity of the system's CoM, and modeling these target variables as a function of the `CPG` parameter inputs. The process outlined in this paper begins with little knowledge of the parameter space and selects samples on-line in an effort to maximize a set of optimization objectives. The different optimization objectives correspond to different angular velocities and resulting directions of motion. In particular, the space of resulting angular velocities for the robot's CoM is discretized. All gaits that result in locomotive behavior within a predefined range correspond to a single class of gaits. The objective is to maximize the system's velocity within each gait class, while simultaneously minimizing the search effort required in order to identify these optimized gaits, i.e., to come up with optimized gaits in an unknown parameter space with minimum computation and without access to a large volume of candidates.

2.3 Parallelized Gait Optimization Framework

With the above objective in mind, the framework uses two classes of predictive models: one class to express each of the target variables (forward and angular velocity of the CoM) as a function of the CPG parameter inputs. The effect of the CPG parameters on the forward velocity of the system's CoM at steady state is modeled by a set of Gaussian process (GP) regression models: a class of predictive models belonging to the family of Bayesian nonparametrics [Williams and Rasmussen, 1996]. These models are often chosen in learning methods where data is limited and the goal is to approximate a smooth nonlinear function from a sparse set of data [Deisenroth et al., 2015, Calandra et al., 2016, Deisenroth and Rasmussen, 2011]. As such, they have gained popularity in robotics applications where sampling is expensive. The GP models are optimized through a process called Bayesian Optimization (BO), which aims to optimize the output (forward CoM velocity) with respect to the parameter input by selecting samples to evaluate in an on-line manner. In the current work, one BO process is modeled for each gait direction to be optimized. For n gaits desired as output, n corresponding BO processes perform the underlying optimizations, all of which are executed in parallel and communicate with each other in an on-line and asynchronous parallelized framework.

The second class of models is composed of a set of K-Nearest Neighbor classification (kNN) models, which are chosen to model the effect of parameter inputs on the angular velocity of the system's CoM. With the assumption that, as a function of the control inputs, the angular velocity target variable exhibits local smoothness, the kNN classifiers are used to perform binary classification for each of the gait direction objectives and the results of this classification are used to bias the samples fed to the BO process. In this way, the kNN processes are used to learn the effective subspaces residing within the bounds of the CPG parameter space and to bias samples fed to the optimization processes toward these effective subspaces. Training and communication of this set of models is also performed on-line and asynchronously, specifically tailoring this learning framework to tackle the optimization objectives in a parallelized manner.

Chapter 3

CPG Design

Previous work developing models of CPG oscillatory structures mimics the oscillatory center in the lamprey: thousands of neurons located in the animal's spinal cord able to produce the oscillatory motions necessary for movement [Crespi and Ijspeert, 2008]. In this line of work, the authors attempt to replicate these biological behaviors by designing a CPG structure for an 8-link salamander robot. By defining a set of differential equations governing the time evolution of phase and amplitude state variables, the resulting double chain oscillatory structure is shown to produce traveling wave motions from the head to the tail similar to those seen in the robot's biological counterpart.

3.1 7-Link Snake Robot

As a benchmark problem, this thesis makes the choice to model a seven-link snake-like system and draws on the large body of literature over the last decade investigating the problem of locomotion in a snake or lamprey-like system [Ijspeert and Crespi, 2007, Crespi et al., 2005]. The system is composed of seven identically sized and weighted links with actuator situated at each of its six joints. Two small passive wheels are attached to the base of each link in order to provide the asymmetrical friction force necessary in order to facilitate realistic locomotion.

Consistent with established work in the corresponding community, a Central Pattern Generator (CPG) control parametrization is selected for this system. The implementation follows prior work [Ijspeert and Crespi, 2007], creating a dual-chain system of oscillatory CPG nodes (illustrated in Figure 3.1). A left and a right node are fixated at each joint of the snake system, which when taken together, define a desired joint angle setpoint, which is then followed by a low-level PD controller. Each joint setpoint

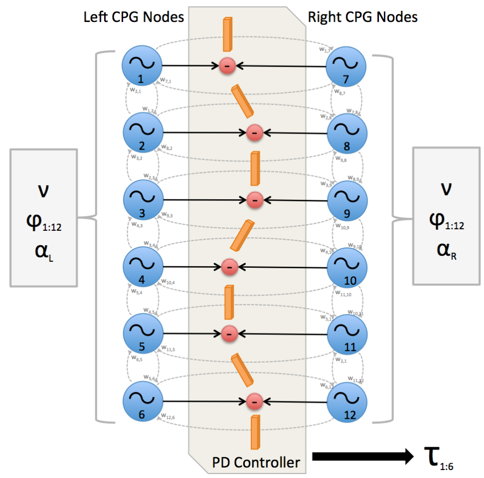

Figure 3.1: CPG system designed for the 7-link snake in simulation, inspired by previous designs for other lamprey-like systems. In blue, the nodes of the CPG system are combined together to induce time-varying, oscillatory controls with constant phase offsets between links. Grey dashed lines represent mutual influence via a weighting mechanism between neighboring nodes, and free parameters of the system are shown in black.

generally behaves as a waveform which is directly influenced by the parameters of the CPG.

3.2 SUPERBall Tensegrity System

In designing the current prototype for the SUPERball concept robot, size and weight requirements led to a design choice of including motors on 12 of the system's cables. The other 12 cables are limited to "passive" functionality. The actuation scheme consists of a single actuated ring around the surface of the robot and two opposing actuated equilateral triangles on each side of this ring. Each of the system's faces contains at least one actuated cable. The benefit of placing the 12 actuators in this pattern is that the system is never in a stable resting configuration without at least one actuated cable on the ground. In this way, basic locomotion for the SUPERball system can be efficiently achieved by "flopping" from face to neighboring face by retracting cables of the resting surface such that the center of gravity of the system moves outside of this support surface. The design choice for the 12-actuated version of the system enables the system

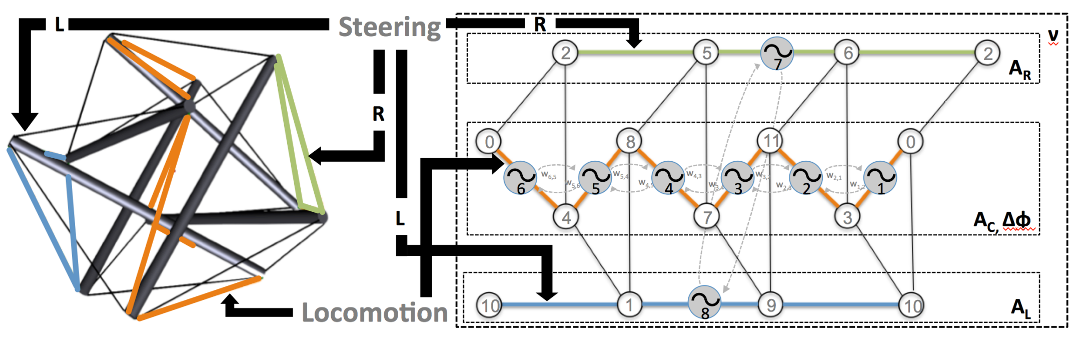

Figure 3.2: The 12 actuated cables of the SUPERBall prototype color-coded and mapped to corresponding nodes within the `CPG` architecture. Locomotion is achieved in the current design by periodically contracting and expanding the lengths of the system's longest actuation chain (orange - center ring). This action effectively shifts the system's CoM and causes it to "roll" onto the next surface in the chain. Steering is achieved by contracting the actuated cables on the side of the system (green, blue). This causes the locomotion behavior to move with a lateral component. The nodes and interconnections of the `CPG` system are shown in light gray and the free parameters of the reduced-dimensionality `CPG` control, as well as the nodes which they individually affect, are shown in black.

to never get "stuck" by resting on an entirely passive face. In practice, the strategy for locomotion using this system involves sequentially flopping onto the faces actuated by the main actuation ring, while "steering" the system by expanding/restricting the side triangles in order to shift the system's center of mass in an effective way to facilitate this flopping behavior.

In prior work on salamander systems, the design objective has been to take advantage of a variety of symmetries in the system to constrain the free parameters in the `CPG` structure, resulting in a parameterization consisting of only 5 free parameters: ν the oscillatory frequency, $\Delta\phi$ the phase offset between nodes, A_L and A_R the left and right oscillatory chain's amplitudes, and α governing an increasing or decreasing multiplicative coefficient on the amplitudes of individual signals from head to tail. Parameterized in this way, each actuated joint of the robot receives a time-varying oscillatory control calculated by combining the time-varying outputs of its corresponding nodes from each of the left, x_L, and right, x_R, oscillatory chains. The desired angular setpoint for the ith actuated joint in the robot is then calculated as:

$$\varphi_i = x_L - x_R$$

and fed to a PD controller. Notably, this control structure is shown to result in asymptotically stable limit cycle oscillatory control signals for biomimetic robots:

$$\varphi_i^{\infty}(t) = \alpha_i(A_L - A_R + (A_L + A_R) \cdot \cos(2\pi\nu t + i\Delta\phi + \phi_0))$$

where ϕ_0 depends on the initial conditions of the system. Importantly, the above choice gives rise to a 5-dimensional parameter space for controlling the system and defining gaits which converge to predictable, periodic behaviors. For salamander robots, the resulting gaits are largely composable, i.e., due to the underlying system dynamics, transitions between gaits happen in a smooth and continuous (rather than discrete switching) manner.

In designing a CPG system structure for controlling the 12-actuator SUPERball prototype, the guiding principle was to build upon this bioinspired line of work while designing a system flexible enough to be able to make use of the physical structure of the prototype, i.e., separately controlling the central ring of actuators for forward locomotion and the two side triangles for steering purposes. The CPG system design consists of a single chain controlling the main actuation ring of the prototype, and a second chain consisting of two nodes controlling the behavior of the left and right actuated triangles (Figure 2). In each chain, nodes are coupled in series, with neighboring nodes exhibiting bi-directional influence on each other.

The present design leverages prior work on salamander CPG design by taking similar advantage of the symmetries of the system in order to reduce the number of free parameters in the CPG structure, including setting all neighboring connections' weights to equal values [Ijspeert and Crespi, 2007]. The free parameters in the end design are: ν the oscillation frequency, $\Delta\phi$ the phase offset between nodes in the main actuation ring (nodes in the secondary chain are constrained to anti-phase), and A_L, A_C, and A_R the amplitudes of the left actuated triangle, main center ring, and right triangle respectively. Collectively, these five parameters are denoted as Θ. Experimentally, this design also exhibits stable limit cycle behavior in the SUPERBall robot, which is denoted as a function of the free parameters in the system:

$$g(\Theta) = \varphi^{\infty}_{1 \ldots N}(t)$$

Notably, while the coupling of the second series of nodes renders it entirely separate from the first, parameters governing frequency are shared, with the frequency of the `CPG` chain governing the side triangles being 3x higher than that of the main actuation ring Experimentally, this gave the best results in terms of achieving flopping locomotion in the system.

Chapter 4

Multi-Objective Optimization

Given the above 5-dimensional `CPG` control parameterization, the output of which consists of rhythmic gaits, the goal then becomes optimizing the gait parameters in order to maximize the effectiveness of a gait "library" according to desirable objectives. As proof-of-concept objectives, this work presents a center of mass (CoM) task space abstraction, within which a set of optimization objectives are defined so as to allow the generation of gaits that move the platform along different directions with high velocity.

4.1 Defining Objectives to Optimize

The motivation for the optimization objectives comes from the need to define gaits that can be used by a motion planner, e.g., search-based or sampling-based. This implies that a discrete set of gaits must be defined which maximize the probability of finding a successful path from a wide variety of starting configurations to a variety of end states under a variety of obstacle configurations in the environment. Furthermore, the robot needs to be able to reach a desired goal configuration fast.

The problem of selecting the best sparse set of candidate paths has been studied in both the static and dynamic planning settings [Knepper and Mason, 2009]. Specific to the dynamic planning setting, several teams competing in the DARPA Urban Challenge made use of precomputed trajectories in the form of constant curvature arcs, which were then used to plan trajectories in an online manner during the challenge [Rauskolb et al., 2008, Von Hundelshausen et al., 2008]. In another line of work, the authors create a modular snake-like robot with a `CPG` control structure and define an abstraction corresponding to this system as the SE(2) configuration of the minimum enclosing bounding box for the system [Hatton et al., 2013]. In the full `CPG` control space, the

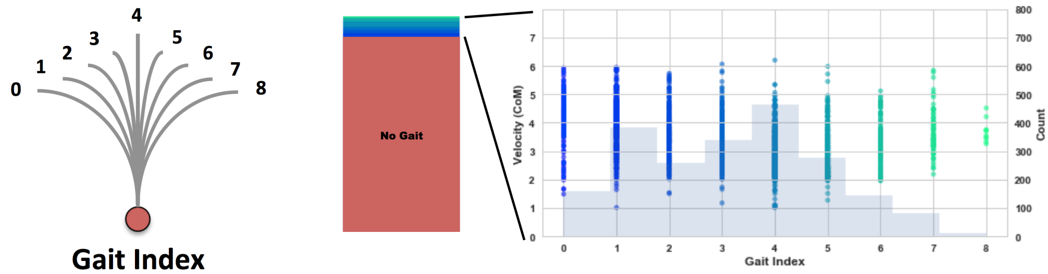

Figure 4.1: Left: Illustration of the pre-selected directions used as optimization objectives for CoM motion of SBB12. Right: Resulting velocities of 20,000 random samples in the proposed `CPG` parameter space. The red area shows that roughly 89.4% of samples correspond to gaits which do not have a stable limit cycle behavior. Among the samples resulting in stable behaviors, a crude normal distribution can be observed among the defined gait angular velocity ranges (shaded columns). Individual data points correspond to forward velocities achieved within each angular velocity range.

authors then define a set of "expressive" gaits, where expressiveness is similar to the notion of controllability in the low dimensional abstraction (i.e., the ability to move the system arbitrarily in this space). Using an informed planner, plans are first generated using knowledge of the expected result of applying individual gaits in the abstracted space, and then are validated on the full system in simulation.

The objective in this work follows closely in spirit to that of constant curvature arcs and low-dimensional expressiveness of gaits. A task space abstraction of the SUPERBall system is first defined as the forward velocity and angular velocity of the system's center of mass (CoM). The angular velocity dimension is discretized into the following 9 directions, illustrated visually in Figure 4.1:

Gait Idx	0	1	2	3	4	5	6	7	8
Min	-10.0	-5.0	-3.0	-1.5	-0.5	0.5	1.5	3.0	5.0
Max	-5.0	-3.0	-1.5	-0.5	0.5	1.5	3.0	5.0	10.0

Then, the objective of maximizing the velocity is defined for each of the discretized directions. By maximizing the system's velocity along several discrete directions of angular velocity, the resulting gaits follow the spirit of a constant curvature arc strategy while also maximizing the velocity achieved along each direction for the spherical SUPERBall. It is then possible to formalize the objectives:

$$\text{Objective i:} \quad \arg\max_{\Theta_i} Vel(g(\Theta_i))$$

where Θ_i are the `CPG` parameters of a gait in the i-th discretization of the angular velocity, and $Vel(g(\Theta_i))$ returns the velocity of the center of mass of the SUPERBall for the corresponding parameters.

4.2 Bayesian Optimization of Individual Objectives

Each of the objective functions to maximize takes as input a 5-dimensional `CPG` parameter set, evaluates the resulting trajectory of the SUPERBall robot using physics-based simulation until a steady state is reached, and returns the resulting forward velocity and angular velocities of the CoM of the robot at steady state behavior. Optimizing these functions lends itself particularly well to the framework of Bayesian optimization (`BO`) specifically because evaluation is expensive, requiring physics-based simulation over a relatively long time horizon. The Bayesian optimization framework is a popular optimization procedure in these types of problems as the aim is to find a global function maximum through a minimal amount of sampled evaluations. This is achieved by maintaining a surrogate function meant to approximate the true objective function, and performing optimization on this surrogate function to solve a sequential decision making problem: at each time step the framework aims to find the sample location, which maximizes the surrogate function, and thus is the current best estimate of the global objective function maximum [Močkus, 1975].

Given the related literature, the target variables are modeled by a Gaussian process (`GP`) regression model using a radial basis function kernel, which exhibits several advantages. `GP` predictions are modeled by smooth, continuous functions, which in the current problem formulation is a property that the output is expected to exhibit (e.g., very similar control inputs result in similar task space behaviors for the system). Also, GP prediction at unseen test points is simply a matrix multiplication operation – which allows for the framework to sample predictions from many points quickly and efficiently.

While the problem of optimizing a single walking gait has been addressed using BO methods previously [Lizotte et al., 2007], the current goal deviates from the standard BO objective. Instead of searching for a single global maximum, the problem of developing a library of gaits dictates that several maximums be found – corresponding to maximizing each of the individual gait velocities. In the current work, these objectives are achieved by specifying the acquisition function for the ith BO node for a given control input Θ:

$$f^i_{acq}(\Theta) = \widehat{Vel}(g(\Theta)) + \sigma(\widehat{Vel}(g(\Theta)))$$

Where $\widehat{Vel}(g(\Theta))$ represents the BO node's estimated velocity resulting from control input Θ and the second term represents the uncertainty about that estimate, which promotes exploration and helps to combat falling into local maxima.

While the individual gait objectives will necessarily have maxima located in distinct locations in the input space, the objectives nevertheless operate in the same space – sharing both domain (input CPG controls) and range (CoM velocity).

4.3 Challenge: Sparsity of the Solution Space

The problem of addressing multiple maximization objectives within a single space presents a unique challenge. In order to address the problem efficiently, a parallelized architecture is designed which facilitates communication between simultaneous processes. This architecture also addresses potential sparseness of good quality solutions within the input space – a common problem when defining and exploring new control spaces for complex systems.

When developing a set of 9 gaits over different ranges of task space angular velocities between $-10°$/s and $10°$/s, Figure 4.1 illustrates the sparsity of the solution space in greater detail. Among 20k randomly sampled points in the CPG control space, only roughly 10% of these meet a minimum criteria for stability (i.e., after converging to the limit cycle behavior, the angular velocity of the robot varies by no more than $\pm 5°$/s, and the linear velocity is greater than 1.0m/s). Notably, if the goal was to find a stable gait with steady state angular velocity in the range $[2.5, 5.0)^\circ$/s (gait index 7 in the presented analyses), only 0.4% of samples would meet those criteria.

4.4 CPG Control Space Classification

In the case of such sparse solution subspaces, the problem of efficient creation of a gait library can be broken down into two subproblems: (a) the exploration of the space in search of the valid regions, which yield usable and desirable motions for the target system and, (b) within those regions, the search for individual gaits which optimize the current objectives. The framework presented in the current work addresses these problems by tailoring predictive models for each task individually. For a problem where a gait library of N different gaits is desired, a set of N binary `kNN` classifiers are trained on samples (both positive and negative) in a streaming fashion as the space is being explored.

These classifiers' predictions are used to bias sampling among the `BO` framework toward subspaces that are predicted to be the most likely to yield maximizing results. In practice, this biasing is achieved through a rejection sampling process: Candidate points are sampled until a target number of points have been found that are predicted by the classifier to belong to a certain gait class. Once found, those promising points are then supplied as candidates to the `BO` process, meaning that any optimization – and ultimately the next evaluated sample location – occurs at a point predicted by the classifier to be promising.

4.5 Parallelized BO Architecture

By creating a parallel architecture for the described `BO` framework, two goals are achieved. First, the design of the architecture initializes N `BO` nodes each focusing on optimizing only a single gait objective, which allows each node to search for a single global maximum in the control space. Second, it allows the processing to be done in a

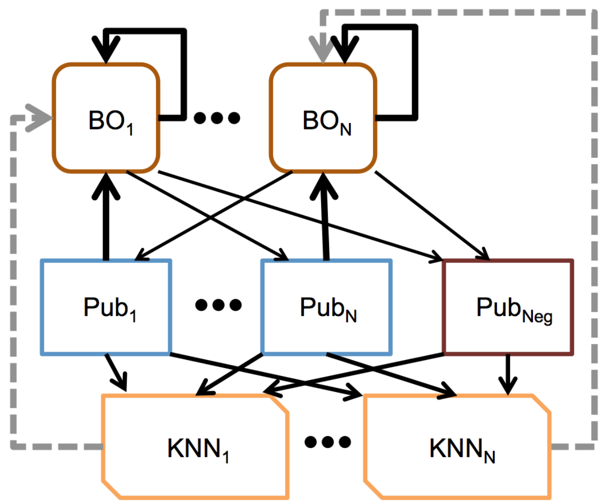

Figure 4.2: Parallel system architecture. Top level nodes $\mathrm{BO}_{1:N}$ produce and simulate samples according to the `BO` objectives. Middle row nodes represent communication nodes: gathering samples generated across all sources and relaying the labeled data back to the proper objectives (thick arrows). The kNN nodes read in all information from positive samples, $\mathrm{Pub}_{1:N}$, and negative samples, Pub_{Neg}. They train a binary classifier, and communicate predictions back to the `BO` nodes in order to help focus their search in the `CPG` space.

maximally parallel fashion. Nodes don't have to wait for all other gait objectives to select and simulate their next sample location as this processing is done in parallel rather than sequentially, while each node also simultaneously is able to leverage information gained from other nodes through asynchronous communication channels.

In Figure 4.5, the resulting architecture is illustrated. In order to optimize N gaits, N `BO` nodes are initialized. Simulation results in terms of velocity and angular velocity of the system's CoM from any of these nodes running in parallel are communicated to one of $N+1$ communication channels corresponding to the N gait objectives (e.g., if node BO_3 samples a gait which ends up belonging to gait objective 4, that control and result is published to channel Pub_4). There is also an additional negative sample class for the points which are deemed unstable. `BO` nodes, however, only take in and train on data from their corresponding channels (BO_k subscribes to channel Pub_k), effectively focusing each node on only the positive samples corresponding to its own objective. The binary `kNN` classifiers, conversely, are continuously trained on data from all channels in order to make the most accurate predictions possible.

Chapter 5

Experimental Results

The efficiency of the presented framework in developing a maximal-velocity gait library can be compared against the natural, uninformed baseline of Monte Carlo sampling. Additionally, the combined parallel framework is compared against results from its component methods (i.e., classification-based region-biasing, and multi-objective gait optimization) in order to evaluate the contributions of each individual component.

5.1 Evaluation of Optimization Framework

In this experimental setup for SBB12, four algorithms were tested in simulation: Monte Carlo sampling of the `CPG` parameters (`MC`), kNN classification-based region biasing (`kNN`), multi-objective Bayesian optimization (`BO`), and the combined approach (`BOkNN`). All algorithms were executed for 10 repeated trials (with different random seeds) for a time of 6,000 seconds for each trial. Simulation was performed using NTRT, an open source tensegrity simulation software package developed at NASA which has been experimentally validated using the SUPERBall hardware [Caluwaerts et al., 2014]. Initial experiments were allowed to run for even longer durations but it was experimentally observed that in the majority of cases the results largely leveled-off prior to 6,000 seconds. Each method was given the objective of maximizing the velocities of 9 individual gaits with angular velocities consisting of a non-overlapping discretization of the range $[-10°/\text{s}, 10°/\text{s}]$, and was correspondingly allowed 9 parallel processes for the duration of each trial's duration. During execution, every 25 iterations the best gaits for each objective in terms of the velocity of SUPERBall's CoM found up to that point were recorded. In order to aggregate and compare results from the repeated trials, a running average was computed from the 10 runs for each of the methods.

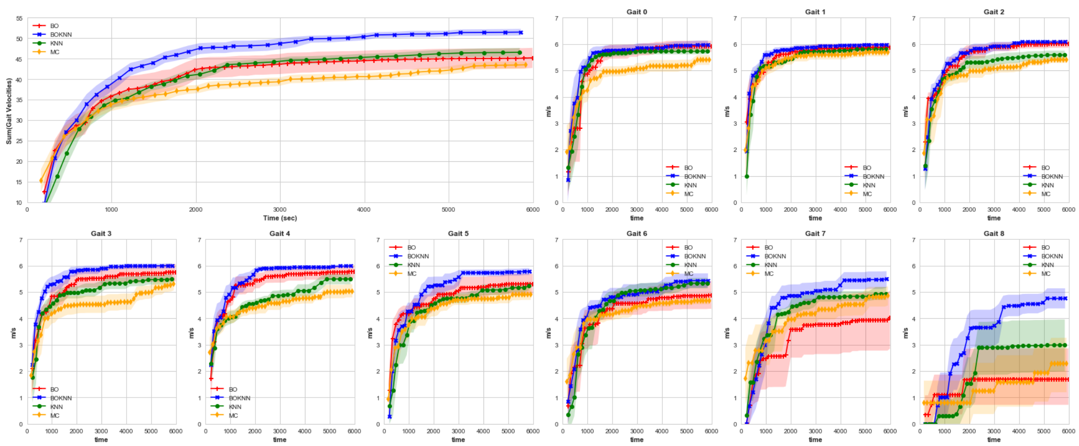

Figure 5.1: The effectiveness of the combined (BOkNN) approach against Monte Carlo (MC), Bayesian Optimization alone (BO), and kNN biasing alone (kNN). Resulting velocities of the best optimized gaits in each direction over a period of 6,000 seconds were recorded and averaged over 10 trial runs per method. Top left shows the aggregate velocities of all optimized gaits, and individual gaits are broken out in smaller graphs over the same time frame. 90% confidence intervals are shown as opaque shading.

Figure 6 shows the results from these trials. Overall (top left), the velocities of all gaits discovered increase over time for each of the individual component biasing and optimization methods. The proposed combined framework, however, achieves a more significant improvement, which begins early on during a trial. Across the plots for individual angular velocity ranges, it can be seen that while there is some variation in the relative performances of the BO and kNN methods, the combined BOkNN approach outperforms the other methods across the board, discovering higher velocity gaits in faster time for every objective. Notably, the relative improvement of the combined method is the greatest in the areas of greatest sparsity (gaits 7 and 8), which is also the areas where optimization alone fails to outperform even the baseline Monte Carlo approach! This result supports the effectiveness of the biasing component in the framework, even in very sparse spaces.

Overall, each individual component of the combined framework outperforms the random sampling strategy frequently, which validates their sample-efficiency properties. Nevertheless, in situations when one component fails, the other component is able to counterbalance (e.g., when `BO` alone struggles in particularly sparse spaces, the `kNN` component intervenes by successfully focusing the optimization process on better quality samples). This counterbalancing behavior is able to provide the combined method with superior results across the board.

5.2 Use of Optimized Gaits in Motion Planning Problems

To evaluate the gait libraries and compare them to the randomly-generated control inputs, an environment without obstacles is considered for evaluation of the optimized set of gaits developed for the 7-link snake system. The comparative metric corresponds to success in getting to different goal locations. Eight different goal locations are chosen as shown in Figure 5.2 to evaluate whether a given control set is able to reorient the snake and traverse toward the goal points. In addition, with no obstacles in the environment, edges can only be removed due to violating state space bounds. These bounds correspond to states that tip the snake over, acquire too much angular velocity (causing ballistic trajectories), or exceed translational bounds. A five minute time limit is imposed on each planning instance, and no solution is reported if this time limit is reached. The randomly generated control experiments are repeated multiple times to account for pseudo-random number generation differences between runs, while experiments using the gait libraries are deterministic, thus requiring only one execution.

Table 5.2 outlines the results for planning along each cardinal direction from the starting state of the snake. Randomly-sampled torques have no success in driving the snake toward any of the goal locations. This is expected since independently sampled torques are unlikely to coordinate the motion of the articulated structure of the snake. Random CPG parameters do achieve some success, but not in 100% of cases. Especially when the snake has a goal that requires fully turning to face a direction opposite of its start, random CPG parameters are difficult to sample that can perform this reorientation. Both gait libraries are able to solve these problems with similar runtime requirements. The learned library using the parallel optimization framework employs less sampled control points to build its library and achieves similar performance to a

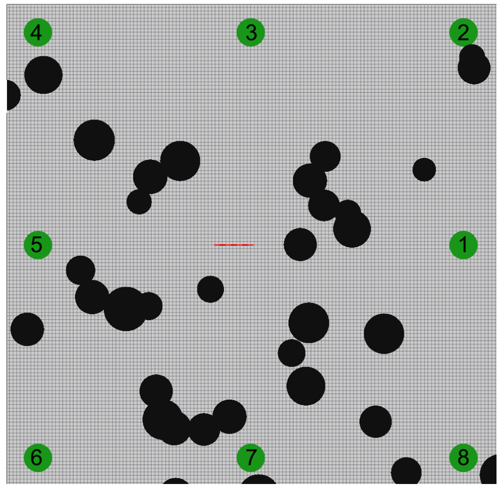

Figure 5.2: The eight goal locations used in the planning evaluations (shown in green), with cylindrical obstacles shown in black. The goal locations evaluate the ability to move effectively along each of the eight cardinal directions. The same goal locations are used for both trials (obstacles and obstacle-free).

Monte Carlo-generated library. For subsequent experiments with obstacles, the randomly sampled controls are omitted and no longer considered viable. This is also due to being unable to use the informed planner without preprocessed knowledge that the gait libraries have.

An A*-like informed hierarchical planning mechanism is used to achieve efficient trajectory generation in an environment with obstacles using the same goal locations as above. In this scenario, obstacles are placed in order to limit the maneuverability of the snake. For each goal location, both gait libraries produced feasible solution trajectories, often in execution time significantly less than the time budget given (10 minutes). Much of this efficiency is owed to the ability to use the expected state transition from each gait, thereby delaying expensive physics simulation to instances where it is absolutely necessary. Figure 5 provides a visualization of the set of expanded trajectories by the planner for different control sets. For the gaits optimized by the parallel framework (BOkNN), a significantly smaller set of nodes needs to be expanded, while still being able to compute a good quality path using the informed planner.

A* Planning Results - Avg. Nodes Expanded by Method								
	Goal 1	Goal 2	Goal 3	Goal 4	Goal 5	Goal 6	Goal 7	Goal 8
Gait Library	Exp.	Exp.	Exp.	Exp.	Exp.	Exp.	Exp.	Exp.
Random Control (6DoF)	161.6	162.6	161.1	157.6	165.7	164.0	177.0	186.8
Random CPG (9DoF)	215.0	240.0	236.0	249.0	241.0	253.0	227.0	237.0
MC (9DoF)	101.0	131.0	101.0	221.0	201.0	221.0	101.0	101.0
BOkNN (9DoF)	111.0	151.0	111.0	241.0	211.0	221.0	111.0	111.0

Table 5.1: Average number of nodes expanded in motion planning experiments involving the 7-Link Snake system in an obstacle-free environment.

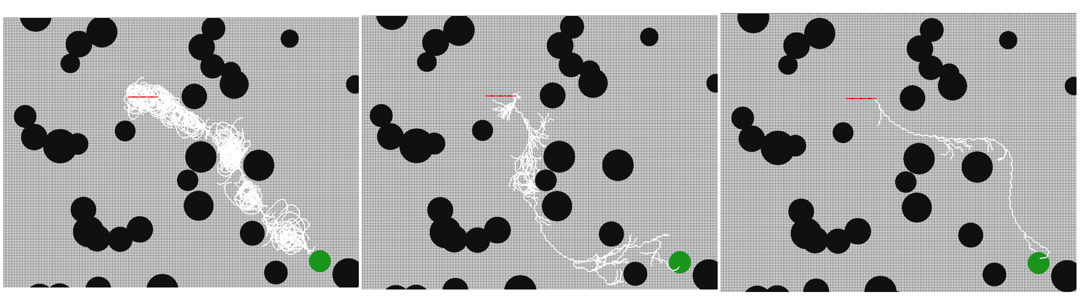

Figure 5.3: Visualization of the expansion of nodes and propagation of controls for various planning algorithms and gait libraries in successfully planned trajectories to goal 8 (green). Left: A* with random CPG parameters. Middle: Monte Carlo generated gait library under the informed planning mechanism. Right: Optimized (`BOkNN`) gait library under the informed planner in an example where the optimized gaits returned the most efficient trajectory. The trajectory was found after only 31 node expansions and approximately 70s of (physics-based) computation.

Chapter 6

Conclusion

This thesis describes a novel `CPG` structure for a complex tensegrity robot which yields periodic locomotive behaviors with varying steady-state effects on the angular velocity and velocity of the robot's center of mass. It then outlines a parallel approach for simultaneously biasing toward promising sample points, while optimizing the velocities of motions in several different directions. To the authors' knowledge, the `CPG` design, while inspired by prior work in the area, is the first successful such design for a spherical tensegrity robot. Furthermore, this thesis also represents the first parallel optimization strategy for simultaneously developing an entire gait library - as opposed to developing individual optimized gaits. The gait velocities achieved in the experimental simulation of the parallel framework outperform anything achieved by random sampling alone for SUPERBall, even in trials run for significantly longer periods of time. Moreover, the velocities achieved by the optimized gaits are greater than those achieved by prior efforts in kinodynamic planning [Littlefield et al., 2016].

The optimization framework is particularly useful when dealing with new systems with broad control spaces, such as the `CPG` formulation it was developed to optimize. Because it is sample efficient, the methodology could also be applied when developing and optimizing gaits for an existing system but in a new environment with, e.g., different terrains or different physical parameters (e.g., exploring a new planet with a different gravity coefficient).

Exciting future work includes examining integration of the gaits with a search-based planner in order to leverage and evaluate the gaits' composability to solve motion planning problems for systems with dynamics as complex as those of the SUPERBall system. This kind of evaluation could also lead to more complex optimization objectives relative to those considered here. For example, if optimizing the gait library for composition of gaits, it may be desirable to evaluate the pairwise (rather than individual) velocities of gaits in order to maximize the effectiveness of the gaits when composed together to form complex trajectories. Lastly, an interesting line of work would be to use the current framework to optimize for low velocity gaits, which might still be very useful for, e.g., lateral movement in the presence of an unknown obstacle. Even more exciting is the idea of allowing a higher-level autonomous process to define the set of gait objectives on the fly depending on the environment it is exploring in a planning-optimization feedback loop.

References

[Bing et al., 2017] Bing, Z., Cheng, L., Chen, G., Röhrbein, F., Huang, K., and Knoll, A. (2017). Towards Autonomous Locomotion: CPG-based control of smooth 3D slithering gait transition of a snake-like robot. *Bioinspiration & Biomimetics.*

[Branicky et al., 2008] Branicky, M. S., Knepper, R. A., and Kuffner, J. J. (2008). Path and trajectory diversity: Theory and algorithms. *Proceedings - IEEE International Conference on Robotics and Automation*, pages 1359–1364.

[Buchli and Ijspeert, 2008] Buchli, J. and Ijspeert, A. J. (2008). Self-organized adaptive legged locomotion in a compliant quadruped robot. *Autonomous Robots*, 25(4):331–347.

[Calandra et al., 2016] Calandra, R., Seyfarth, A., Peters, J., and Deisenroth, M. P. (2016). Bayesian optimization for learning gaits under uncertainty. *Annals of Mathematics and Artificial Intelligence*, 76(1-2):5–23.

[Caluwaerts et al., 2014] Caluwaerts, K., Despraz, J., Işçen, A., Sabelhaus, A. P., Bruce, J., Schrauwen, B., and SunSpiral, V. (2014). Design and control of compliant tensegrity robots through simulation and hardware validation. *Journal of The Royal Society Interface*, 11(98):20140520.

[Cohen et al., 2010] Cohen, B. J., Chitta, S., and Likhachev, M. (2010). Search-based planning for manipulation with motion primitives. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 2902–2908. IEEE.

[Crespi et al., 2005] Crespi, A., Badertscher, A., Guignard, A., and Ijspeert, A. J. (2005). Swimming and crawling with an amphibious snake robot. In *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, pages 3024–3028. IEEE.

[Crespi and Ijspeert, 2008] Crespi, A. and Ijspeert, A. J. (2008). Online optimization of swimming and crawling in an amphibious snake robot. *IEEE Transactions on Robotics*, 24(1):75–87.

[Deisenroth and Rasmussen, 2011] Deisenroth, M. and Rasmussen, C. E. (2011). Pilco: A model-based and data-efficient approach to policy search. In *Proceedings of the 28th International Conference on machine learning (ICML-11)*, pages 465–472.

[Deisenroth et al., 2015] Deisenroth, M. P., Fox, D., and Rasmussen, C. E. (2015). Gaussian processes for data-efficient learning in robotics and control. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(2):408–423.

[Hatton et al., 2013] Hatton, R. L., Knepper, R. a., Choset, H., Rollinson, D., Gong, C., and Galceran, E. (2013). Snakes on a plan: Toward combining planning and control. *Proceedings - IEEE International Conference on Robotics and Automation*, pages 5174–5181.

[Herreo-Carron et al., 2011] Herreo-Carron, F., Rodriguez, F. B., and Varona, P. (2011). Bio-inspired Design Strategies for Central Pattern Generator Control in Modular Robotics. *Bioinspiration & Biomimetics.*

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

[Hopkins and Gupta, 2014] Hopkins, J. and Gupta, S. K. (2014). Design and Modeling of a New Drive System and Exaggerated Rectilinear-gait for a Snake-Inspired Robot. *ASME Journal of Mechanisms and Robotics*, 6(2).

[Ijspeert and Crespi, 2007] Ijspeert, A. J. and Crespi, A. (2007). Online trajectory generation in an amphibious snake robot using a lamprey-like central pattern generator model. *Proceedings - IEEE International Conference on Robotics and Automation*, pages 262–268.

[Iscen et al., 2014] Iscen, A., Agogino, A., SunSpiral, V., and Tumer, K. (2014). Flop and roll: Learning robust goal-directed locomotion for a tensegrity robot. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2236–2243.

[Karaman and Frazzoli, 2011] Karaman, S. and Frazzoli, E. (2011). Sampling-based algorithms for optimal motion planning. *The international journal of robotics research*, 30(7):846–894.

[Knepper and Mason, 2009] Knepper, R. A. and Mason, M. T. (2009). Path diversity is only part of the problem. *Proceedings - IEEE International Conference on Robotics and Automation*, pages 3224–3229.

[Koizumi et al., 2012] Koizumi, Y., Shibata, M., and Hirai, S. (2012). Rolling tensegrity driven by pneumatic soft actuators. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 1988–1993. IEEE.

[Kulkarni et al., 2016] Kulkarni, T. D., Narasimhan, K. R., Saeedi CSAIL, A., and Tenenbaum BCS, J. B. (2016). Hierarchical Deep Reinforcement Learning: Integrating Temporal Abstraction and Intrinsic Motivation. *ArXiv*, pages 1–13.

[LaValle, 1998] LaValle, S. M. (1998). Rapidly-exploring random trees: A new tool for path planning. Technical report, Iowa State.

[Lessard et al., 2015] Lessard, S., Bruce, J., Agogino, A., Sunspiral, V., and Teodorescu, M. (2015). Robust Monte Carlo Control Policies to Maneuver Tensegrity Robots out of Obstacles. In *ARMS*, pages 1–16.

[Li et al., 2016] Li, Y., Littlefield, Z., and Bekris, K. E. (2016). Asymptotically Optimal Sampling-Based Kinodynamic Planning. *International Journal of Robotics Research.*

[Littlefield et al., 2016] Littlefield, Z., Caluwaerts, K., Bruce, J., SunSpiral, V., and Bekris, K. E. (2016). Integrating simulated tensegrity models with efficient motion planning for planetary navigation.

[Lizotte et al., 2007] Lizotte, D. J., Wang, T., Bowling, M. H., and Schuurmans, D. (2007). Automatic gait optimization with gaussian process regression. In *IJCAI*, volume 7, pages 944–949.

[Mirletz et al., 2015] Mirletz, B. T., Bhandal, P., Adams, R. D., Agogino, A. K., Quinn, R. D., and SunSpiral, V. (2015). Goal-directed cpg-based control for tensegrity spines with many degrees of freedom traversing irregular terrain. *Soft Robotics*, 2(4):165–176.

[Močkus, 1975] Močkus, J. (1975). On bayesian methods for seeking the extremum. In *Optimization Techniques IFIP Technical Conference*, pages 400–404. Springer.

[Oliveira et al., 2013] Oliveira, M., Matos, V., Santos, C. P., and Costa, L. (2013). Multi-objective Parameters CPG Optimization for Gait Generation of a Biped Robot. In *IEEE International Conference on Robotics and Automation (ICRA)*.

[Ostrowski and Burdick, 1998] Ostrowski, J. and Burdick, J. (1998). The Mechanics and Control of Undulatory Locomotion. *International Journal of Robotics Research (IJRR)*, 17(7):683–701.

[Ostrowski et al., 2000] Ostrowski, J., Desai, J., and Kumar, V. (2000). Optimal Gait Selection for Non-holonomic Locomotion Systems. *International Journal of Robotics Research (IJRR)*.

[Rauskolb et al., 2008] Rauskolb, F. W., Berger, K., Lipski, C., Magnor, M., Cornelsen, K., Effertz, J., Form, T., Graefe, F., Ohl, S., Schumacher, W., et al. (2008). Caroline: An autonomously driving vehicle for urban environments. *Journal of Field Robotics*, 25(9):674–724.

[Sfakiotakis and Tsakiris, 2007] Sfakiotakis, M. and Tsakiris, D. (2007). Neuromuscular Control of Reactive Behaviors for Undulatory Robots. *Neurocomputing*, (70):1907–13.

[Shammas et al., 2005] Shammas, E., Choset, H., and Rizzi, A. (2005). Natural Gait Generation Techniques for Principally Kinematic Mechanical Systems. In *Robotics: Science and Systems (RSS)*.

[Skelton and de Oliveira, 2009] Skelton, R. E. and de Oliveira, M. C. (2009). *Tensegrity systems*, volume 1. Springer.

[Smith et al., 2005] Smith, R. et al. (2005). Open dynamics engine.

[Tesch, 2013] Tesch, M. (2013). *Improving Robot Locomotion Through Learning Methods for Expensive Black-Box Systems*. PhD thesis, Carnegie Mellonw University.

[Von Hundelshausen et al., 2008] Von Hundelshausen, F., Himmelsbach, M., Hecker, F., Mueller, A., and Wuensche, H.-J. (2008). Driving with tentacles: Integral structures for sensing and motion. *Journal of Field Robotics*, 25(9):640–673.

[Walsh and Sastry, 1995] Walsh, G. and Sastry, S. (1995). On Reorienting Linked Rigid Bodies using Internal Motions. *IEEE Transactions on Robotics and Automation*, 11(1):139–146.

[Williams and Rasmussen, 1996] Williams, C. K. and Rasmussen, C. E. (1996). Gaussian processes for regression. *Advances in neural information processing systems*, pages 514–520.

[Wright et al., 2007] Wright, C., Johnson, A., Peck, A., McCord, Z., Naaktgeboren, A., Gianfortoni, P., Gonzalez-Rivero, M., Hatton, R., and Choset, H. (2007). Design of a modular snake robot. In *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, pages 2609–2614. IEEE.