Title

Design with Robotic Intelligence in Mind---An Analysis Using Specific Examples

Author

Pogue, Alexandra

Publication Date

2024

UNIVERSITY OF CALIFORNIA
Los Angeles

Design with Robotic Intelligence in Mind—An Analysis Using Specific Examples

A dissertation submitted in partial satisfaction
of the requirements for the degree
Doctor of Philosophy in Mechanical Engineering

by

Alexandra Nguyen Pogue

2024

ABSTRACT OF THE DISSERTATION

Design with Robotic Intelligence in Mind—An Analysis Using Specific Examples

by

Alexandra Nguyen Pogue
Doctor of Philosophy in Mechanical Engineering
University of California, Los Angeles, 2024
Professor Dennis W. Hong, Co-Chair
Professor Ankur M. Mehta, Co-Chair

This collection of studies explores various facets of robotic intelligence, emphasizing the life-cycle of a robot from inception to advanced operation. Beginning with the design phase, the concept of "physical intelligence" is demonstrated using the Non-Anthropomorphic Biped-Soleus (NABi-S) robot. The NABi-S robot, with its unique leg alignment and compliant soleus mechanism, demonstrates stability and agility while allowing for recovery from perturbations without complex control systems. Simplified kinematic analyses and open-loop control algorithms are all that are necessary for natural motion.

In cases where closed-loop control is necessary, intuitive design approaches are taken to hybridize the CPG controller with zero dynamic control methods in order to render marginally stable and unstable zero dynamics of a cart-pole system, stable. Without the use of rigorous control theory, empirical tuning approaches are all that is necessary to synchronize a CPG state to the cart's pivot state to robustly stabilize the system. We then extend closed-loop CPG control to a hybrid dynamical system, the Simplest Walker, and show that stability regions surrounding walking cycles can be extended. This leads to the discussion of the synergistic benefits of hybridizing a CPG controller with an HZD controller. Through analysis, it is found that robots like NABi-S that have a higher number of degrees of freedom (DOFs) and walk unconventionally can be controlled successfully with this algorithm.

In consideration of communication and cooperation, it is shown that the integration of Multi-Input Multi-Output (MIMO) network theory enhances spatial intelligence and awareness among UAV teams, allowing for high-level trajectory optimization in simultaneous data aggregation and communication tasks. By making small adjustments to pre-existing sensing tasks, this approach achieves significant improvements in network efficiency, demonstrating how “spatially aware” modifications to data collection plans can lead to substantial gains in performance.

Lastly, for robotic perception, SLAM (Simultaneous Localization and Mapping) techniques are crucial. The Block Online Expectation Maximization (BOEM) SLAM algorithm presents a robust hybrid approach to visual-inertial navigation that combines filtering and optimization techniques. Unlike optimization-based SLAM methods, this method does not require the processing of an entire batch of data at once, yet it achieves greater accuracy than filtering methods alone. By efficiently fusing visual and inertial data, BOEM-SLAM supports accurate realtime localization while lowering hardware requirements, making it useful for the simplification of robotic systems by algorithmic means.

Together, these studies underscore the multifaceted intelligence required for modern robots, from foundational physical design to advanced spatial and visual capabilities.

The dissertation of Alexandra Nguyen Pogue is approved.

Tetsuya Iwasaki

Veronica Santos

Dennis W. Hong, Committee Co-Chair

Ankur M. Mehta, Committee Co-Chair

University of California, Los Angeles

2024

To my dad, who asks me every time we speak if I am "piled higher and deeper yet?"—
to which I can finally say, "yes."

Table of Contents

List of Figures

List of Tables

NOMENCLATURE

Central Pattern Generator Modeling

ξ	oscillator state
α	oscillator amplitude
ω	oscillator frequency
μ	oscillator convergence parameter
z	oscillator state in complex form

Zero Dynamics Control Design

θ	parameterizing robot state defining a point in configuration space
ϕ	parameterizing CPG state defining a point in configuration space
q	vector of robot generalized coordinates
$\mathcal{Z}$	the zero dynamics manifold
$\tilde{\mathcal{Z}}$	the zero dynamics manifold arising from hybridization with CPG control

Hybrid CPG-HZD Control Theory

φ	phase of the CPG
$\mathcal{Q}$	configuration space
$T\mathcal{Q}$	tangent bundle of the configuration space
$\mathcal{S}$	impact surface
Δ	impact map
$\mathcal{W}$	the intersection of $\mathcal{Z}$ and $\tilde{\mathcal{Z}}$
$\dim(\cdot)$	$\cdot$ -dimensional

ACKNOWLEDGMENTS

Obtaining this degree has been the hardest thing I have ever done. First, I would like to acknowledge myself. This experience has been a true test of character, and I am certainly better for it.

Otherwise, I would like to acknowledge the people who helped me through this degree: teD Iwasaki, Kenny Chen, Sep Ghassemi, Josh Hooks, Matt Dotson, Samer Hanna, Kevin Liu, Tsang-Kai Chang, and Anyone else I May have overlooked. Your kindness and generosity of spirit made my time here measurably better. I appreciate you and I thank you so much.

VITA

2013	B.S., Chemical Engineering, University of California, San Diego
2016	M.S., Mechanical Engineering, University of California, Los Angeles
2019	Advanced to Ph.D. Candidacy, University of California, Los Angeles

PUBLICATIONS

J. Friesen, A. Pogue, T. Bewley, M. de Oliveira, R. Skelton, and V. Sunspiral, "DuCTT: A tensegrity robot for exploring duct systems," 2014 IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, China, 2014, pp. 4222-4228, doi: 10.1109/ICRA.2014.6907473.

J. Yu, J. Hooks, S. Ghassemi, A. Pogue and D. Hong, "Investigation of a non-anthropomorphic bipedal robot with stability, agility, and simplicity," 2016 13th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI), Xi'an, China, 2016, pp. 11-15, doi: 10.1109/URAI.2016.7734010.

A. Pogue, A. Bianes, D. Hong and T. Iwasaki, "NABI-S: A compliant robot with a CPG for locomotion," 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, Canada, 2017, pp. 3366-3371, doi: 10.1109/IROS.2017.8206175.

A. Pogue, S. Hanna, A. Nichols, X. Chen, D. Cabric and A. Mehta, "Path Planning Under MIMO Network Constraints for Throughput Enhancement in Multi-robot Data Aggregation Tasks," 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA, 2020, pp. 11824-11830, doi: 10.1109/IROS45743.2020.9341096.

K. Chen, A. Pogue, B. T. Lopez, A. -A. Agha-Mohammadi and A. Mehta, “Unsupervised Monocular Depth Learning with Integrated Intrinsics and Spatio-Temporal Constraints,” 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Prague, Czech Republic, 2021, pp. 2451-2458, doi: 10.1109/IROS51168.2021.9636030.

T. -K. Chang, A. Pogue and A. Mehta, “BOEM-SLAM: A Block Online EM Algorithm for the Visual-Inertial SLAM Backend,” 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Kyoto, Japan, 2022, pp. 6420-6427, doi: 10.1109/IROS47612.2022.9982015.

CHAPTER 1

Introduction

Robotics is a rapidly evolving field that encompasses various aspects of robot design, control, and perception. As robots become more prevalent in our daily lives, it is crucial to design them with intelligence in mind to ensure efficient, stable, and adaptable performance. This thesis explores several examples of robotic intelligence in different contexts, highlighting the challenges and innovative solutions in each case.

One of the fundamental challenges in humanoid bipedal robots is the three-dimensional control and balancing problem, mainly due to the hip sway that occurs during weight transfer between the support and swing legs. This sway causes periodic roll moments on the robot's center of mass (CoM), while the swing foot induces pitch moments in the plane of forward motion. Minimizing the effects of out-of-plane motion on overall stability often requires expensive force/torque and inertial sensors for closed-loop control [1]. The Non-Anthropomorphic Biped (NABi) robot addresses this issue through its unique leg alignment in the sagittal plane, enabling agile motion without complex control. NABi-S, a smaller version of NABi, incorporates a spring or "soleus" mechanism to store energy and stabilize walking, demonstrating agile motion using open-loop, Central Pattern Generator (CPG) control.

When the control loop is closed using sensory feedback, CPGs offer advantages such as limit cycle stability and entrainment, which allow robots to synchronize with natural dynamics and achieve stable motion. This research uses these principles to impose a virtual holonomic constraint on a mechanical system with a single degree of underactuation while also stabilizing a periodic orbit within the constraint manifold. To achieve this, a stable CPG limit cycle is embedded within the zero dynamics manifold of a cart-pole system, enabling stabilizing control in both downward and upward positions.

The foundation of this work lies in the concept of zero dynamics (ZD), which describes the internal dynamics of a system when the output is constrained to be identically zero. While zeroing the system output ensures the manifold containing the zero dynamics is invariant and attractive, stabilizing the zero dynamics remains a challenge when they are inherently unstable. This research integrates the Andronov-Hopf (AHO) CPG model, a nonlinear oscillator that guarantees global convergence to a limit cycle trajectory to achieve this stabilization. The AHO oscillator is time-invariant and exponentially stable, making its dynamics ideal. In order to embed the AHO state within the constraint manifold, it is first shaped using a Fourier series, then the unstable robot state is input as sensory feedback to the CPG. This ensures that the CPG manifold is fixed to the zero dynamic manifold within the domain of the desired trajectory *and* the neighboring states.

Building on the foundational work in continuous dynamical systems, the next chapter delves into the hybrid extension of CPG control, focusing on closed-loop CPG control and its application to hybrid systems through phase resetting at impacts. By employing the Simplest Walking Model, we illustrate how the hybrid CPG model can stabilize walking gaits that are otherwise unstable under open-loop conditions. Additionally, the application of HZD control to side-walking is investigated, utilizing the NABi robot as a case study. The combination of CPG and HZD controls aims to overcome the limitations of traditional time-based synchronization, providing an intuitive and adaptive framework for dynamic robotic gaits.

Future work centers around exploring the theoretical foundation and practical implementation of CPG-HZD control, specifically targeting stable periodic orbits within the constraint manifold. Through the analysis of energy balance and the use of the Poincaré return map, this research aims to identify stable periodic solutions that ensure robust and efficient locomotion. This approach not only simplifies CPG control design but also has the potential to enhance the overall stability of HZD control, thus paving the way for an advanced, nonlinear control method with enhanced stability properties.

In the context of multi-robot systems, efficient data aggregation and communication are essential for coordinated task execution. Under line-of-sight (LOS) network conditions, multi-input multi-output (MIMO) wireless communications can increase the channel capacity between a team of robots and a multi-antenna array at a stationary base station. By positioning robots to maximize complex channel orthogonality between each robot and receiver antenna, the system can achieve greater data throughput and shorter task completion times. Geometrically motivated assumptions can be used to derive transmitter spacing rules that improve backhaul throughput for data offloading from the robot team, with minimal impact on other system objectives.

Visual-inertial Simultaneous Localization and Mapping (SLAM) is another crucial aspect of robotic intelligence, enabling robots to create globally consistent trajectories and maps of their environment. However, the growing computational costs and storage requirements of SLAM backends pose a significant challenge. BOEM-SLAM addresses this issue by leveraging the hidden Markov model structure to summarize historical data into sufficient statistics and discard it, making it a data-efficient algorithm. Compared to other fundamental approaches, BOEM-SLAM demonstrates considerably lower computation time with comparable estimation performance.

In this thesis we analyze these specific examples of robotic intelligence, focusing on the design considerations and innovative solutions that enable efficient, stable, and adaptable robot performance. By examining the challenges and successes in each case, we can gain valuable insights into the future of robotic intelligence and its potential applications in various domains.

The following chapters of this thesis are arranged as follows, Ch. 2 provides an overview of the mechanical design and development of the Non-Anthropomorphic Biped (NABi) robot. It details the NABiRoS compliant foot mechanism, introduces the NABi-S variant, and presents the design and function of the Soleus mechanism. Ch. 3 explores foundational concepts in motion planning using a Central Pattern Generator (CPG). It presents a simple method to achieve open-loop trajectories along with results of walking speed trials. Ch. 4 introduces the design and implementation of a closed-loop CPG-ZD controller. It includes the background on zero dynamics and virtual constraints, details the design of the CPG-ZD controller, and presents simulations using the cart-pole model. Ch. 5 applies the CPG to a hybrid dynamical system using the Simplest Walker model and phase resetting techniques. This chapter also addresses the challenges and methodologies for integrating CPG control with hybrid zero dynamics for side-walking. Ch. 6 discusses the importance of spatial awareness in enhancing robotic communication. It addresses several communication models and explores the effects of transmitter position on MIMO network performance. Ch. 7 presents the Block Online Expectation Maximization (BOEM) approach to SLAM. It compares BOEM-SLAM with other backend algorithms in realistic simulation scenarios, highlighting its efficiency and suitability for energy-constrained environments. The final chapter, Ch. 8, outlines future directions for the development of CPG-HZD control. It includes stability analysis, theoretical development for side-walking, and implementation strategies. The chapter also discusses the synergistic benefits of combining CPG and HZD control, offering a promising pathway for advanced robotic locomotion.

CHAPTER 2

Biped Robot Design by Physical Intelligence

2.1 Background: Non-Anthropomorphic Biped (NABi)

In the following sections, usage of terms such as “ankle” and “toes” normally attributed to humans, etc. in addition to those shown in Fig. 2.6 will be used to refer to robot parts.

2.1.1 NABi Mechanical Design

The leg mechanical design in the NABi family of robots is essentially the same across robot models. The legs are designed to address the “hip sway” that occurs in conventional biped walking. The out-of-plane moments don’t exist if humanoid legs were rotated by 90 degrees, we can think of it like a “side step” when trying to squeeze through tight spaces (Fig. 2.1). A typical humanoid leg has 6 degrees of freedom (DOFs): roll and pitch at the ankle, pitch at the knee, and roll, pitch and yaw at the hip. To take side steps, fewer are necessary to maintain balance because the hip sway that occurs when shifting weight stays in the plane of motion. Thus, we keep all pitch DOFs, because those are the only *necessary* DOFs for forward motion. Like Raibert’s hopper studies in [7] we are focused only on a single element of the motion problem, with the current subset of DOFs the robot can propel forward and backwards only, it does not turn. A later version of the NABi robot with yaw DOFs at the hip joint can be seen in Sec. A.

The NABiRoS hardware prototype is pictured in Fig. 2.2. The robot stands 1 meter tall when in a typical “crouched” position and 1.35 meters tall when including the attached cardboard box that acts as a “torso” for weight distribution purposes. The robot has two

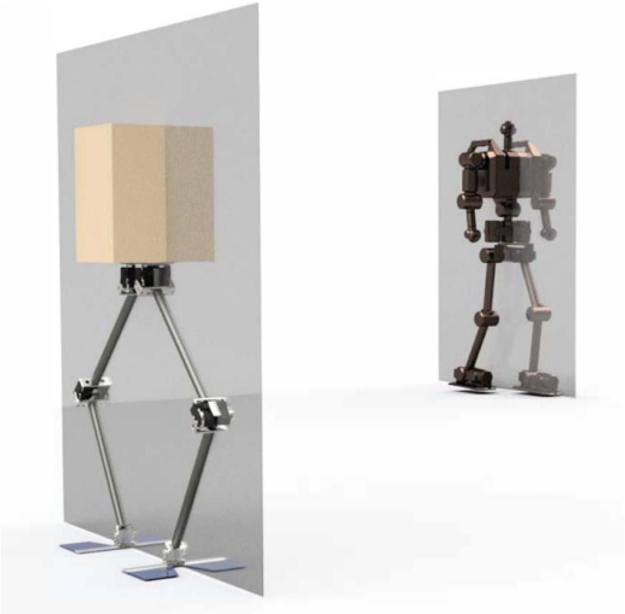


Figure 2.1: NABiRoS version robot showing rotation of legs in the sagittal plane when compared to a “typical” humanoid robot [1].

Figure 2.2: NABiRoS hardware prototype [1].

actuated DOFs located at the hip and the knee. An actuated ankle DOF is unnecessary and has been replaced by a rigid ankle linkage connecting the tibia to a simple foot element. The robot can rotate about the ankle-foot linkage in the same way a point foot acts as a pinned connection in dynamic legged locomotion [7], but the foot serves to stabilize the robot while moving, facilitating quasi-static locomotion techniques. The removal of motors at the ankle also significantly reduces the weight and moment of inertia of each leg [1]. The legs of the robot are made of carbon fiber tubes. The body (above the hips) holds the onboard computer. Using carbon fiber tubes rather than aluminum for the leg linkages reduces the weight of the robot without sacrificing rigidity. Aluminum is used for motor bracketing only. Including the external power infrastructure, the robot is a total of 3.97 kg [1].

2.1.2 NABiRoS Complaint Foot Mechanism

Compliant feet are used on NABiRoS to account for small perturbations and uneven terrain, without the need for an ankle joint. Legged robots designed to be mechanically rigid with high-gain, position controlled joints require foot trajectories to match terrain contours perfectly. This type of design works well in a laboratory setting where the environment is known, but fails when introduced to real world conditions [8]. A widely used solution to this problem is the series elastic actuator (SEA), which adds a compliant element in series with a traditional actuator. The SEA analog in the NABiRoS system is captured in the compliant feet. These end effectors are in series with the motors via the rigid tibia-ankle linkage, a simpler mechanical configuration than a typical SEA which attaches the elastic element directly to the actuator power train. Adding this elastic element to NABiRos leads to better energy efficiency via spring storage, disturbance rejection via foot stabilization, and allows for control system robustness to unmodeled robot dynamics and environmental conditions.

Looking at Fig. 2.3 we can see that the brackets that connect the foot to the tibia (effectively the ankle) are configurable. This allows for varying spring foot engagement angles. When the robot walks, the feet can be configured such that they only take action

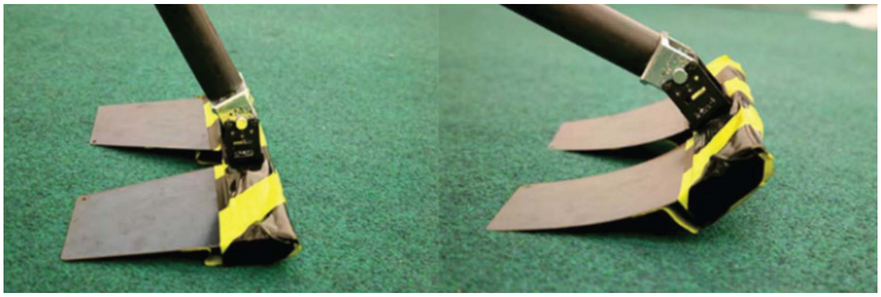


Figure 2.3: NABiRoS foot mechanism showing adjustable rigid ankle connected to a spring steel stabilizing plate

during specific motions to avoid the negative effects of an SEA such as the dampening of robot dynamics [9]. This way they serve as SEA's during specific portions of the walking cycle, while allowing for fast response times in others when the spring system is not engaged.

The foot is made of 1095 blue tempered spring steel [10]. This material has a higher carbon content than other spring steels for wear resistance and continual stress. This makes the material sufficiently "stiff" to hold the weight of the robot as it "leans" or shifts its weight. The spring model is nonlinear so it was simplest to determine the appropriate foot length empirically by trial and error. The spring is purposed to function such that when it is engaged, there is a restoring force of desired magnitude depending on the amount the spring is deflected passed its equilibrium position, i.e. the amount by which the robot "leans" on the spring (see Fig. 2.3, image on the right).

The details of the walking algorithm and results can be found here [1]. The foot mechanism's impact on NABiRoS walking is addressed specifically by Yu et al. when they stated, "The inclusion of the compliant element showed immense improvements in the stability and robustness of walking gaits on the prototype, allowing the robot to remain stable during locomotion without any inertial feedback control."

2.2 Non-Anthropomorphic Biped-Soleus (NABi-S)

The soleus muscle, shown in Fig. 2.4, is a wide flat leg muscle found on the posterior of the tibia. It runs from just below the knee to the heel via the Achilles tendon. This muscle together with the gastrocnmius and plantaris muscles form the calf muscle on the human leg. Soleus contraction allows for plantarflexion and helps to maintain posture by preventing the body from falling forward at the ankle while standing [11].

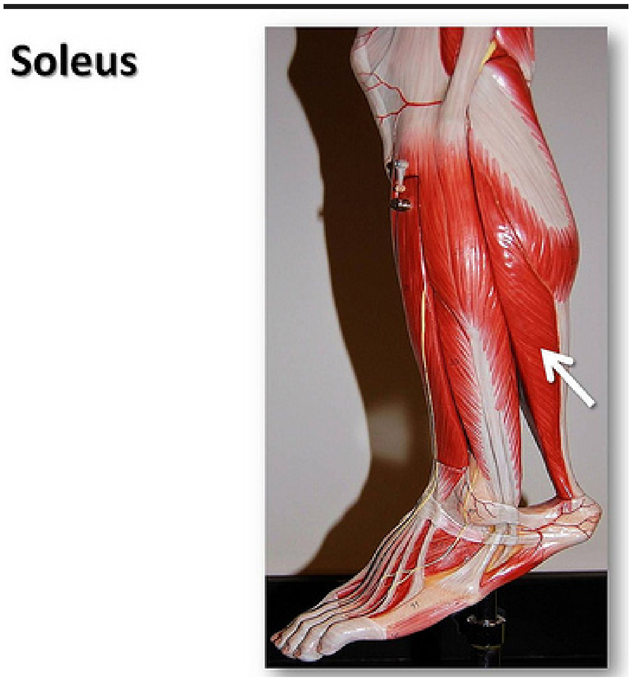


Figure 2.4: The soleus muscle on the human leg [2].

NABi-S, shown in Fig. 2.5, is a smaller version of NABiRoS and thus shares the same stability properties as its larger counterpart. The main difference is in the construction of the ankle and foot components, where here we introduce a novel “soleus” mechanism that replaces the rigid ankle and spring steel end effectors on NABiRoS.

The NABi leg design makes it nearly impossible to tip over in the sagittal plane because the leg that is taking a step acts as an “anchor” on the robot center of mass, keeping it within a constrained space. The ankle portion of the soleus mechanism on NABi-S is not rigid, however, which increases the the angular momentum of the robot about its stance foot considerably compared to NABiRoS. Parallel to the tibia, the soleus is made up of spring elements; they engage when the robot rotates about the stance foot, stabilizing motion and turning the angular momentum into potential energy. The feet on NABi-S were fashioned from aluminum in a “truss-like” design for strength-to-weight efficiency. They were also

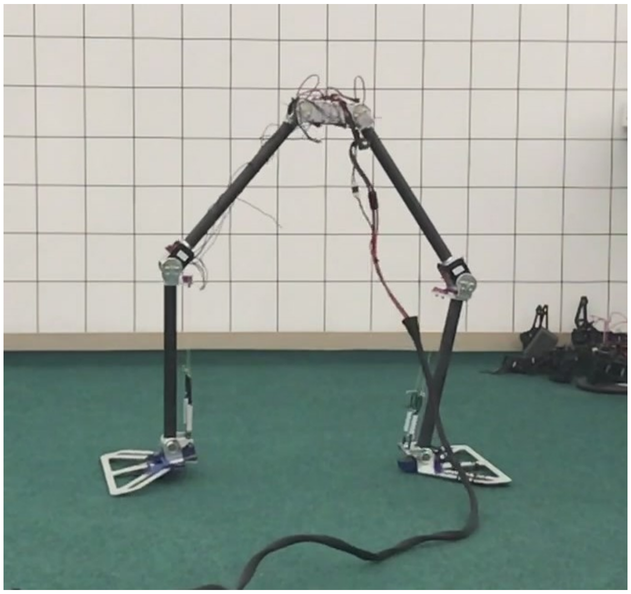


Figure 2.5: NABi-S hardware prototype showing soleus mechanism.

made sufficiently wide to prevent possible out-of-plane tipping.

2.3 Mechanical Platform

2.3.1 Overall Design and Mechanical Components

A labeled model of NABi-S is shown in Fig. 2.6. The robot stands .8 m in height, 20 cm smaller than NABiRoS. The robot weighs 2.86 kg. Tubes and brackets have been reduced in size to minimize weight. The 2.54 cm diameter, 30.48 cm length structural tubes are made from carbon fiber. They are light and small in diameter to minimize the inertia of the legs. There are two Robotis Dynamixel MX-106 servomotors that operate each of the four degrees of freedom located at the hips and knees [12].

2.3.2 Soleus Mechanism

The soleus was conceived to closely approximate a torsional spring at the passive ankle over a range of movement. This allows for estimation of the spring's role in dynamical analysis. The design was built with additional functionality in mind such as variable pretension,

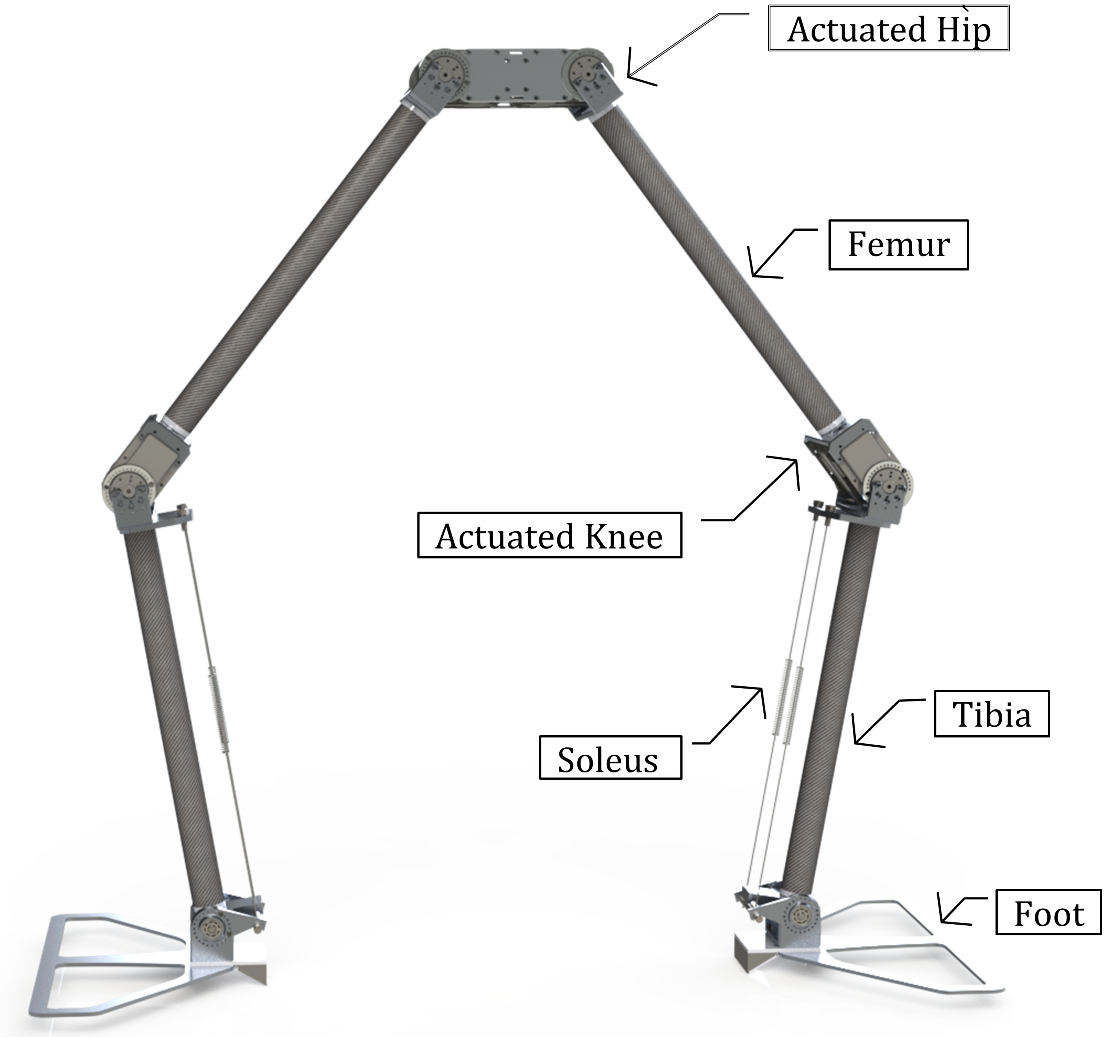


Figure 2.6: NABi-S robot design with labeled components.

engagement, and stiffness properties, which can be configured manually. As previously stated, mechanism design morphology emulates the soleus muscle of a human, and shares its role in walking and maintaining standing posture.

2.3.2.1 Spring Determination

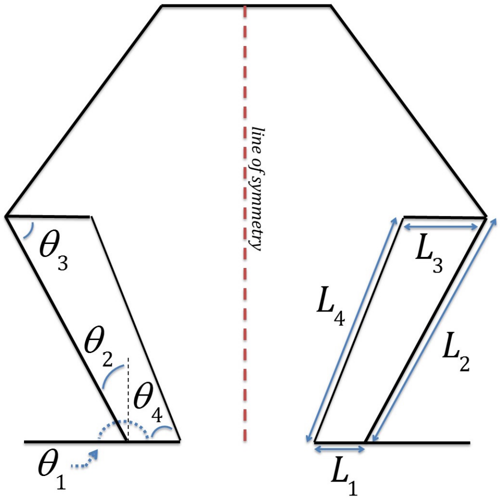


Figure 2.7: Soleus mechanism schematic with labeling.

Referring to Fig. 2.7, the distance between fixed points on the foot (Link 1) and tibia (Link 3) changes as the ankle rotates. The placement of a tension spring between these two points exploits this change by storing potential energy and providing a restoring force. The static force propogation method was used to determine approximate spring stiffness for the soleus [13]. The simplest approach considered the robot as a structure, with mass concentration at the center of mass. A virtual link was created from the center of mass to the hip joint. This load was then propagated to the ankle.

The approach is a type of force and moment balance. Subscripts denote force or moments exerted on link i by link $i-1$, and so on:

$$f_i = f_{i+1}. \tag{2.1a}$$

If P is defined as the position vector from link i to link $i+1$, and moments lie along joint axes, then the torque at joint i is defined as

$$\tau_i = \tau_{i+1} + P_{i+1} \times f_{i+1}. \tag{2.1b}$$

In implementation, separate reference frames were assigned to each link, and a rotation matrix was used to describe force and moment relationships in a single reference frame. From the above equations, the torque at the ankle was determined. Referring to Fig. 2.7, the distance $L4$ increases as the angle θ_2 increases. The moment arm to the force generated by the spring lying along Link 4 is $L1$. Using this relationship, the spring constant, k, is determined by the equation,

$$k = \triangle L4 \times L1/\tau_{ankle}. \tag{2.1c}$$

Using Equations 2.1a through 2.1c, a robot position requiring maximum use of the spring-standing on one foot with tibia 25 degrees from vertical, generated a torque of 99.84 Ncm. Using link length and angle data associated with prototype parameters defined in the following section, Sec. 2.3.2.2, a spring constant of 20.43 N/cm was determined.

2.3.2.2 Design Implementation

The soleus mechanism, pictured in Fig. 2.8, was designed and manufactured with rapid prototyping in mind. The knee motor brackets were the strongest point of attachment for Link 3 and there was sufficient clearance to attach them there, resulting in an L_2 of 32.4 cm. Attaching Link 3 perpendicular to the tibia was simplest, resulting in a θ_3 value equal to 90 degrees. Remaining link lengths $L1$ and $L3$, equal to 3.56 cm, were chosen to be small for weight reduction.

Referring to Fig. 2.8, the mechanism consists of a large foot plate for stability (A), ball bearings for smooth rotation at the joint, and thrust bearings to maintain rigidity (B). Soft EVA foam was placed in a hinged joint at the heel to absorb impact, and between the foot and the ankle to dampen rotation (C). To ensure the spring always stays in tension, the spring is attached to a custom bracket at the base of the tibia via its connection to a dowel pin (D). When the ankle moves through the configured point of engagement, the base connection of the spring is passed from the tibia bracket to the "wing-set" brackets attached to the foot (E). These brackets are configurable at 15 degree intervals to allow for a range of spring engagement. The spring attachment to the opposite end of the tibia is via an upper plate that attaches to the knee bracket, with vented screws to allow for pretensioning (F). Fig. 2.9 depicts the foot bracket wing-set in not engaged and engaged positions. The mechanism works reliably, including testing where the spring is transferred to the foot at high frequencies.

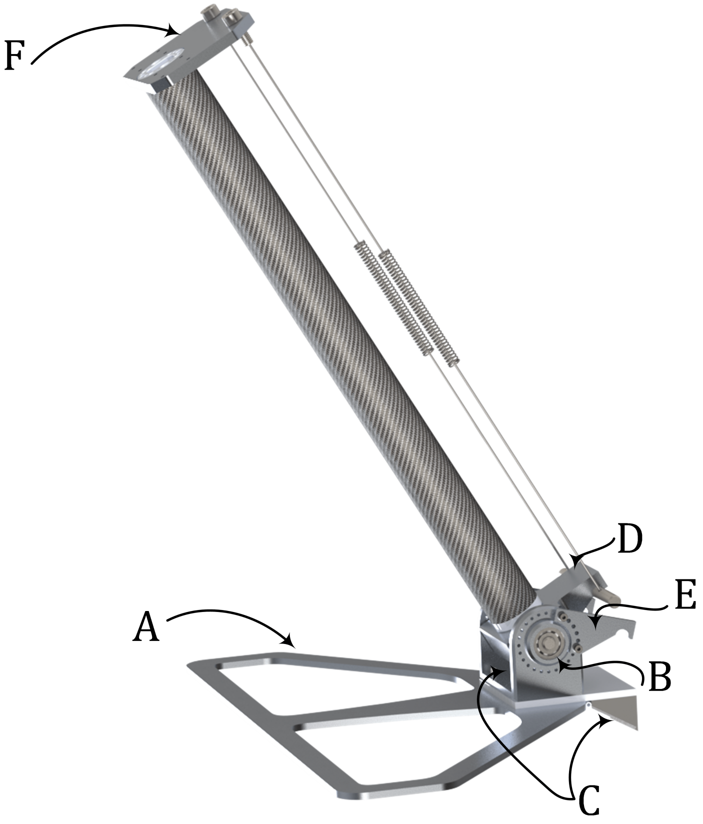


Figure 2.8: Prototype design of the soleus *(*Note: in the current "engaged" position, the dowel pin connection should be to Bracket E not D).*

The soleus was designed to increase stability when walking. Specific tests verifying its effect, however, were not conducted. For the many gait parameters used in initial CPG tests, it was observed that the mechanism turned backwards motion into forward motion when the back leg turned the potential energy created by a center of mass shift into kinetic energy of

(a) Spring not engaged

(b) Spring engaged

Figure 2.9: Closeup of the soleus base attachment.

forward motion. When there is no spring to load as the robot leans back a “rearing” motion is created as the swing foot kicks upwards. Eliminating instances where the legs move too much in the vertical direction protects the motors from impulses created at foot impact with the ground.

Spring stiffness for speed tests performed in Section 3.3 were 21.02 N/cm in total for each mechanism (6 lbs/in/spring). The mechanism wing-set was fixed at a middle setting, i.e. θ_1 equal to 180 degrees for spring engagement to begin at a θ_2 of 0 degrees (Fig. 2.7). This value was held fixed for all testing due to gait behavior change for different settings.

2.3.2.3 Kinematic Analysis

The mechanism performance was analyzed using the spring stiffness and other parameters determined in the preceeding sections, to asses whether it approximates torsional spring behavior. The mechanism was treated as an open kinematic chain containing three rigid links, and kinematic position analysis was applied between Links 1 and 3 using a Newton-Raphson numerical method [14]. At full range of motion the spring is non-linear due to the decrease in θ_4 as θ_2 increases, causing torque output attenuation (Fig. 2.7).

If the range θ_2 is truncated to have a maximum angle of 35 degrees, however, the coef-

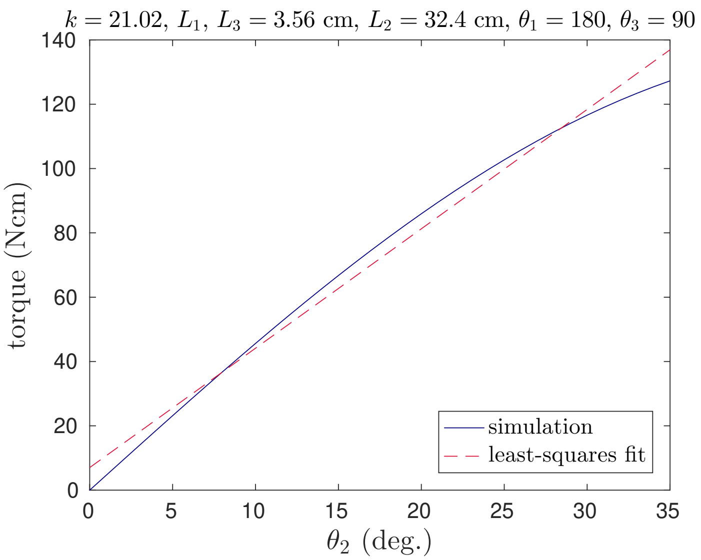


Figure 2.10: Mechanism performance over truncated range.

ficient of determination, or R-squared value is equal to .9886. Fig. 2.10 depicts mechanism performance for the actual prototype using previously described link lengths, and a 21.02 N/cm linear spring.

2.3.3 Soleus In Action

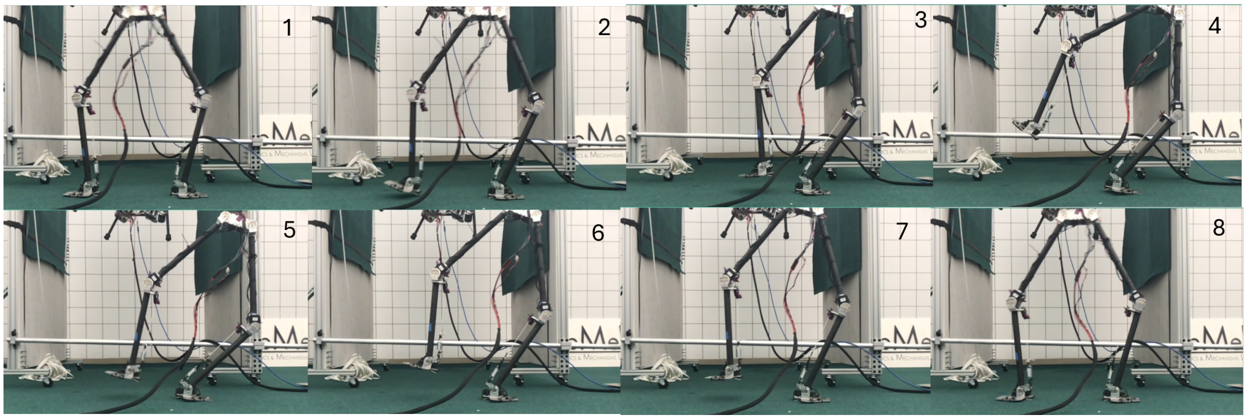


Figure 2.11: Timelapse showing the robot leaning against the Soleus without falling over.

Fig. 2.11 shows timelapsed images from video footage of NABi-S kicking up one leg as it leans against the Soleus at the ankle. The images clearly show that despite the smaller feet, the mechanism facilitates very agile, stable motion.

CHAPTER 3

Simple Approaches to Generating Movement

3.1 Background: Central Pattern Generator Motion Planning

There is evidence that humans and animals achieve some forms of motion using a neural network known as a central pattern generator (CPG) [15]. A CPG is a low-level pattern generator used by animals in repetitive motions. A CPG is used in this portion of the research as a biologically-inspired means to generating motor trajectories. This is a favorable approach because empirically tuning oscillator parameters provides intuitive results for robot motion without much effort.

CPGs have been successful in simulation, but fewer studies have been done with robots. Endo et al. had positive results, achieving over 50 steps with their in-plane biped, although their studies included a boom attachment for stabilization of lateral movement [16]. The balancing issue may be why CPGs are more often associated with quadrupeds such as Salamandra Robotica and Tekken [17], [18]. Quadrupeds exhibit self-stabalizing behavior because they keep two or more feet on the ground most of the time [19]. Studies have shown CPGs for locomotion are found in the spinal cord, at a low-level of motor function [15]. This means the rhythmic motion CPGs produce are generated without conscious effort. Quadrupeds are currently best suited for this low-level type of control because unlike agile biped robots, they do not require sophisticated balancing algorithms. A CPG alone can be sufficient for locomotion. With the use of our physically intelligent biped system, however, we are able to perform stable biped walking using a CPG without the use of a boom for stabilization assistance.

3.2 Control

Only an open-loop CPG was necessary to achieve walking with NABi-S. To determine trajectory parameters, first a closed-form approach was taken in designing the CPG model. Then empirical methods were used to further tune trajectory states during implementation.

3.2.1 Control Theory

Multivariable harmonic balance equations (MHB) were used to generate the desired oscillation profile for walking [20]. Using this theory, a four neuron model was developed, each neuron representing an actuated degree of freedom. Neuron dynamics are described by the following set of equations. Input u_i is equal to the weighted sum of the output of presynaptic neurons,

$$u_i = \sum_{j=1}^{n} \mu_{ij} v_j \tag{3.1a}$$

where μ_{ij} is a real, scalar valued weight of the synaptic connection from the j^{th} neuron. Low pass filter dynamics have the property $1/w_o$, representing the cell membrane time constant and inverse of the filter pass band of neuron i,

$$\dot{q}_i = -w_o q_i + w_o u_i \tag{3.1b}$$

where the neuron signal frequencies must have the property $w \leq w_o$ for the CPG model to exhibit oscillatory behavior. To maintain oscillation without degradation, and capture the threshold and saturation properties of the neuron, a nonlinear, sigmoid operation is performed on q_i,

$$v_i = \varphi(q_i). \tag{3.1c}$$

A hyperbolic tangent function satisfies these characteristics, and was used for this research. For vectors $q \in \mathbb{R}^{n\times 1}$ and $v \in \mathbb{R}^{n\times 1}$, whose i^{th} entries are q_i and v_i respectively, the neuron model dynamics can be arranged according to the following systems of equations:

$$\dot{q} = -w_o q + w_o M v, \tag{3.2a}$$

$$v = \Phi(q). \tag{3.2b}$$

For a system composed of n neurons, matrix $M \in \mathbb{R}^{n\times n}$ may be constructed, with row i containing presynaptic weights corresponding to the i^{th} neuron. Thus, matrix M is described as a connectivity matrix. If neuron j excites neuron i, μ_{ij} is positive, otherwise if it inhibits neuron i the value is negative. For the CPG controlled system, Eqs. (3.2a) and (3.2b) generate quasi-sinusoidal waves. A set of n amplitudes and n phase relationships, described by $q_i \simeq \alpha_i \sin(wt + \phi_i)$ are desired. To determine M such that neurons exhibit a desired frequency and the proper gain/phase relationships, the following linear matrix inequality is solved:

$$\min_{M} \|M\| \text{ s.t.} \begin{cases} MKR = R\Omega \\ N^T MKN + (N^T MKN)^T < 0 \end{cases} \tag{3.3a}$$

where Eq. (3.2b) is approximated by $v \cong \mathcal{K}(\alpha)q$, with $\mathcal{K}(\alpha) := diag(k_i(\alpha_i))$. The i^{th} diagonal entry of $\mathcal{K}(\alpha)$ is taken from the describing function of $\tanh(q_i)$. Among the remaining parameters in Eq. (3.3a) are

$$\Omega := \begin{bmatrix} 1 & -\bar{w} \\ \bar{w} & 1 \end{bmatrix}, \quad \bar{w} = j(1 - \frac{1}{f(jw)}), \quad K := \mathcal{K}(\alpha) \tag{3.3b}$$

where $f(jw)$ is equal to the filter transfer function. Matrix N forms an orthonormal basis for the null space of R^T, with matrix R defined as

$$R := \begin{bmatrix} \alpha_1 \sin(\phi_1) & \alpha_1 \cos(\phi_1) \\ \vdots & \vdots \\ \alpha_n \sin(\phi_n) & \alpha_n \cos(\phi_n) \end{bmatrix}. \tag{3.3c}$$

3.2.2 Control Implementation

Empirical tuning methods were used to generate the oscillation profile for walking. First forward kinematics were used to establish nominal robot appendage placements [21]. Approximate command signal amplitudes and phase relationships were determined using this method prior to testing on hardware. Referring to Fig. 3.1, the robot hip angles are measured from the axis perpendicular to the rotating ϕ frame and the knees are measured relative to the hips.

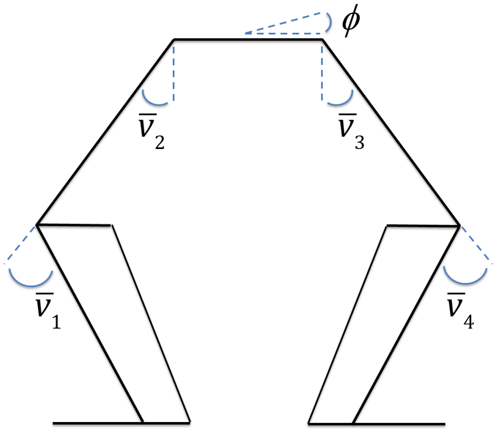


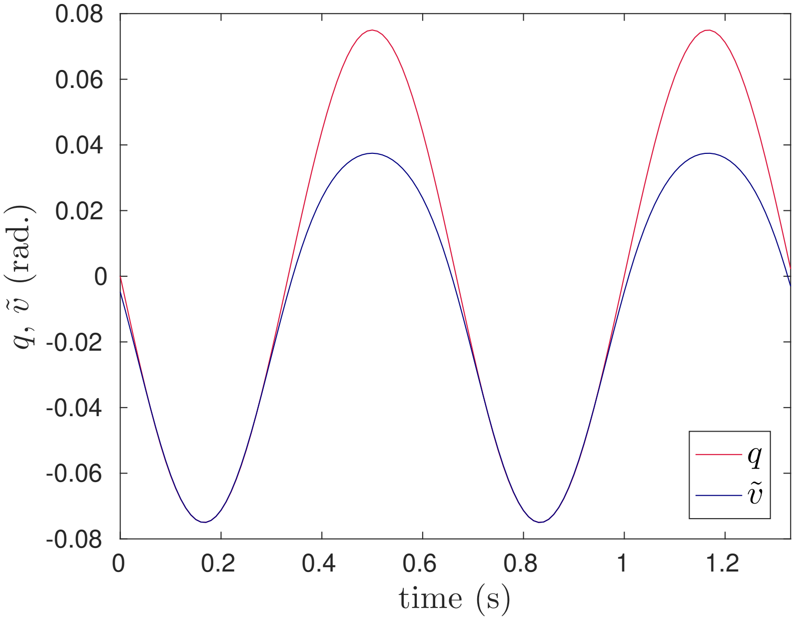
Figure 3.1: Joint angle convention for a given position.

Hardware tests indicated the need to slow leg velocity and keep the feet from getting too close on an inward swing. To do so oscillation profiles were modified using weights applied to the sigmoid function in Eq. (3.2b). The alteration produced clear stance and swing legs in the walking cycle. Thus, q computed from Eq. (3.2a) followed two network paths. The path for sigmoid operation, $\phi(\cdot)$, went back through the CPG. This path determined signal frequency and phase relationships. The modified sigmoid, $\phi_i(\cdot)$, of the form

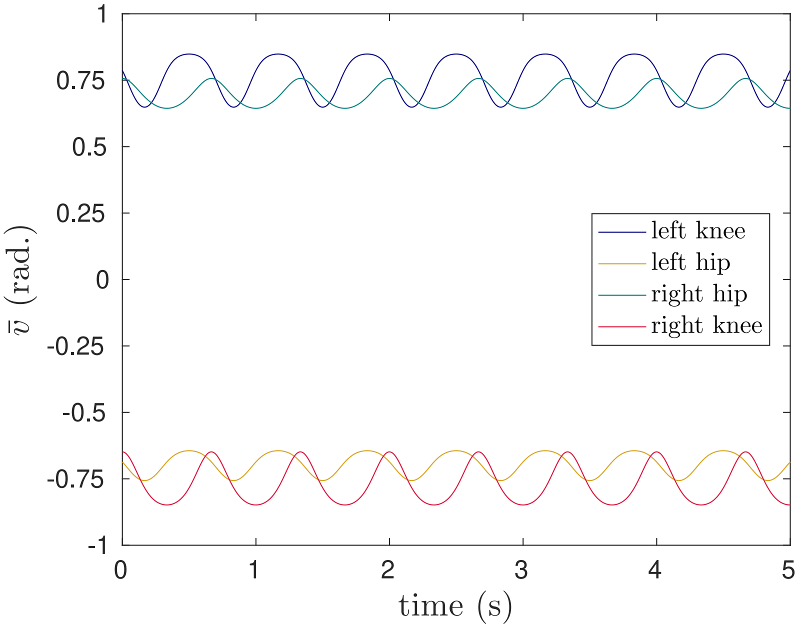
$$\tilde{v}_i = \varsigma_i \tanh(\beta_i q_i + \sigma_i) + \delta_i \tag{3.4}$$

shaped the signal in such a way that it appeared saturated (Fig. 5.2a). This allows the leg to swing outward more than inward, for example. The signals were then shifted vertically, $\bar{v} := \tilde{v} + \rho,\ \rho \in \mathbb{R}^{n\times 1}$. These position commands to the motors are shown in Fig. 5.2b. The shift by ρ ensured leg swing occurred about a crouched position.

Fig. 5.2b shows example motor signals for walking. There is a phase lead (ϕ) of 90 degrees by the right side of the robot if following the angle convention defined in Fig. 3.1. In this configuration, at $t = 0$, the right leg is extended. The robot walks in the direction of phase lead. The phase relationship between hip and knee, ν_i, was fixed at zero degrees. This



(a) CPG input signal q, and output signal $\tilde{v}$ (left hip)



(b) Motor signals $\bar{v}$, for walking

Figure 3.2: Example CPG and motor signals.

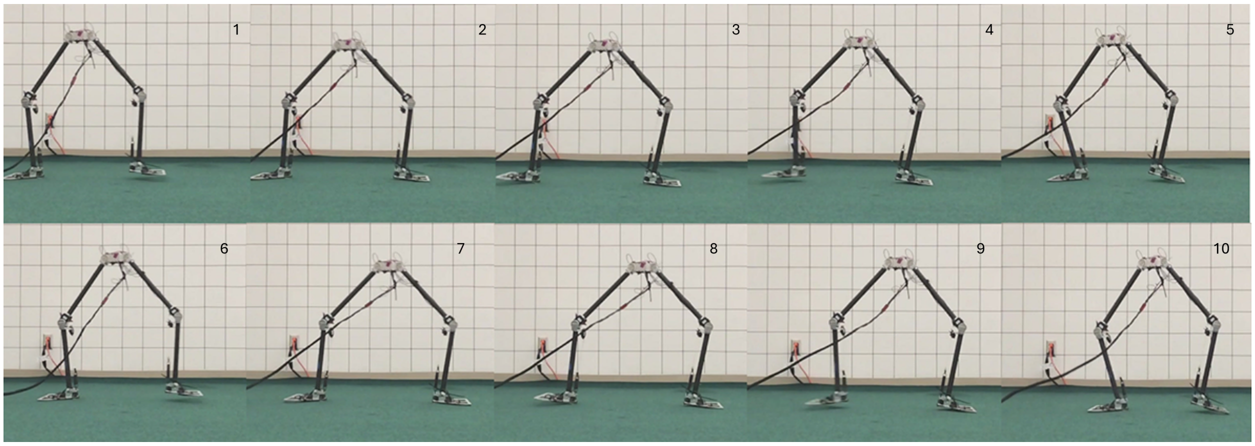


Figure 3.3: Frames depicting two steps of robot walking.

relationship, such that the foot extends outward at the same time the hip swings the femur, was deemed acceptable for walking. Amplitudes (Table 3.1 of Section 3.3) show more swing from the knee than the hip, to limit vertical motion of the leg. Frequencies (f) for Fig. 3.2 are 1.5 Hz.

Fig. 3.3 shows two steps of walking typical of phase leads and frequencies in the range of 70-100 degrees, and 1.5 to 1.8 Hz. The required trajectory signal amplitudes are proportional to the spring stiffness used, meaning less leg swing is required when using a lower spring constant (14.01 N/cm pictured). In walking the robot leans back to lift the front leg, shifts its body forward on the spring to pick up its back foot, then shifts back to begin the cycle again. Walking in this range is consistent from step to step, and robot movement looks natural. Stepping is not perfectly symmetrical however, with more lift coming from the front foot.

3.3 Parameter Space Search Results

A parameter space search determined the frequency and phase lead to produce the best walking results. Speed and a consistent gait were metrics for good walking. Frequency and phase lead ranges were 1.4-1.9 Hz and 70-120 degrees, determined based on initial tests showing (relatively) high walking speeds in the approximate mid-range. Referring to

Table 3.1: Fixed Parameters for Walking Trials

Parameter	$\boldsymbol{\varsigma_i}$	$\boldsymbol{\beta_i}$	$\boldsymbol{\sigma_i}$	$\boldsymbol{\gamma_i}$	$\boldsymbol{\nu_i}$
Left Knee	0.1846	5.4824	0.6344	0.6828	0 deg.
Left Hip	0.1334	7.6070	0.5265	-0.7509	/
Right Hip	0.1387	-1.5022	0.5719	0.3958	/
Right Knee	-1.0814	-2.8076	1.4354	0.1815	0 deg.

Eq. (3.4), all parameters held fixed are shown in Table 3.1, with $\gamma_i := \delta_i + \rho_i$. Five trials were taken per frequency and phase combination to produce average speed data. The robot was timed as it walked a distance of 1.22 m.

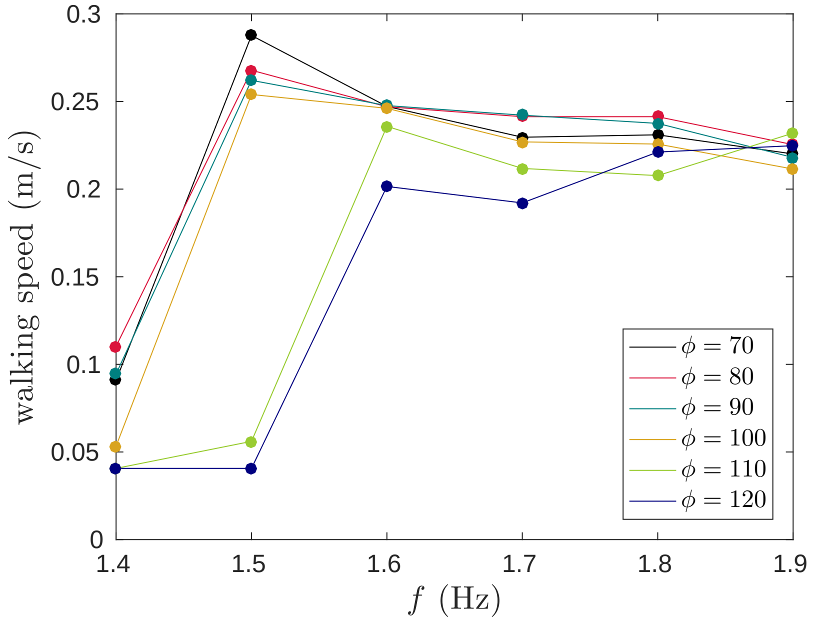


Figure 3.4: Average speed (over 5 trials per data point) as a function of frequency and phase.

Fig. 3.4 shows average speed data located at points on the line plot. At 1.4 Hz the spring strength contributed to a lack of foot lift, causing slow walking. Speeds increased significantly at 1.5 and 1.6 Hz. The peak walking speed of .288 m/s was recorded at a phase of 70 degrees, and frequency of 1.5 Hz. This speed is slightly faster than the fastest recorded speed of NABiRoS, a larger robot by .2 m in hip height [1]. As frequency increased, all speeds converged to approximately .222 m/s. Observation of walking indicated the highest average speeds were also the most consistent between trials. Lack in repeatability was a product suboptimal stepping such as skipping and backwards stepping, that decreased averages. For clearer trends, more than 5 trials per data point are needed. Variance data could then be taken to quantify walking consistency.

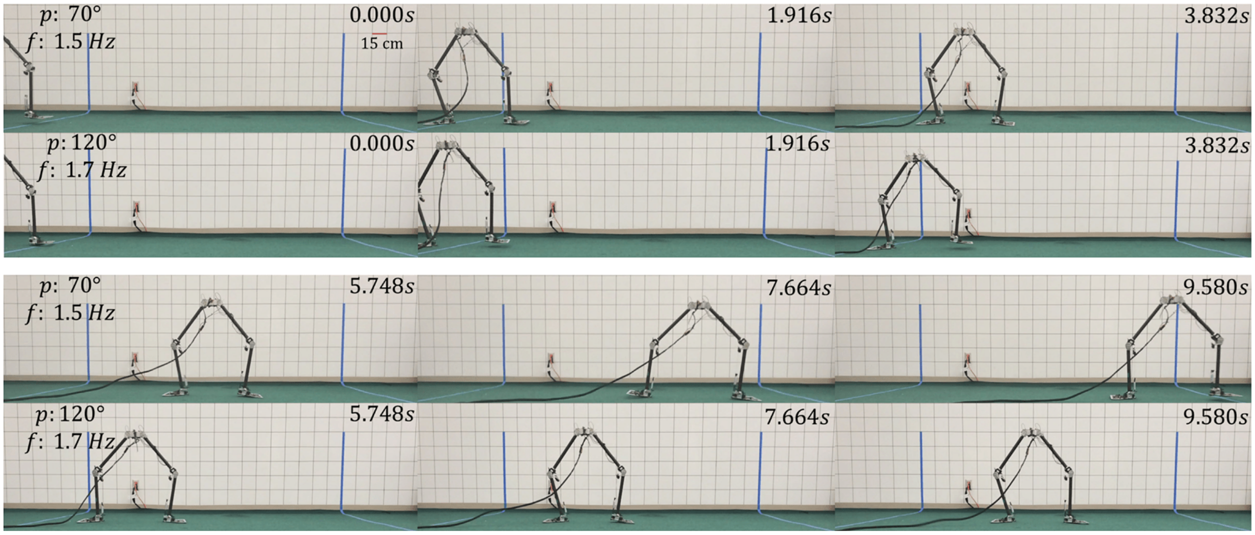


Figure 3.5: Frames comparing two separate speed trials.

Fig. 3.5 shows a side by side comparison of walking produced by two different sets of gait parameters. The comparison clearly demonstrates the impact that frequency and phase have on walking speed. The trial for optimized walking (70 degrees, 1.5 Hz) finished the distance of 2.7 m in a total of 9.58 seconds. The walking speed of .282 m/s is within the expected range for this frequency and phase. The trial using unoptimized gait parameters (120 degrees, 1.7 Hz) finished the distance in 15.55 seconds. This speed of .174 m/s is below the expected range (-.0082 m/s).

3.4 Future Work in Trajectory Planning

For future work, gait optimization strategies for improved lift on the back foot will be studied [22]. In order to achieve limit cycle walking, the loop will be closed on the CPG. Using a band-pass filter to model neuronal dynamics may make entrainment possible [23]. Work has already been done to derive the robot's dynamical equations for feedback control. Motor control methods for these measures will change to torque control. Torque control can create both proprioceptive actuation and a robot adaptable to the environment.

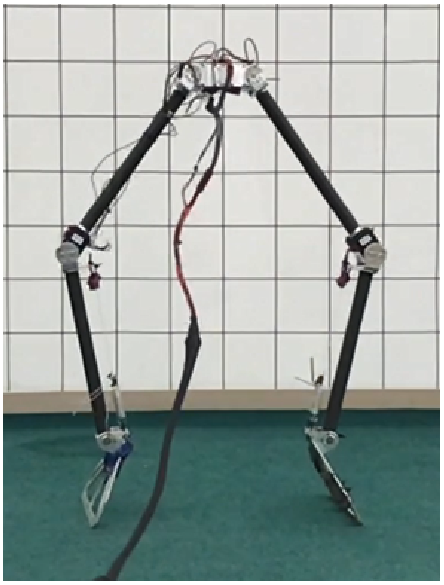


Figure 3.6: Image showing NABi-S using the soleus mechanism for plantarflexion.

Like the soleous of the human, the wing-set on the elastic mechanism can perform plantar flexion, that is, increase the angle between the foot and tibia so the robot stands on its toes (similar to a pantograph leg, see Fig. 3.6). Using stronger springs, a toe-walking gate will be investigated. The springs must be stiffer and will impact the ankle more in this configuration. The study will help determine how the CPG performs when compliance and morphology change.

CHAPTER 4

A Novel, Closed-Loop CPG-ZD Controller

Using a CPG with sensory feedback has benefits such as limit cycle stability and entrainment. Entrainment is the ability to synchronize with natural dynamics. Limit cycle stability is defined as a robot state that does not require local stability at every point along its trajectory, but is stable because a given trajectory will eventually converge to the desired trajectory over a number of time steps [24]. These aspects of CPG control make the production of natural, stable gaits possible as an alternative to high gain position control.

While the research within this chapter is purposed as a foundational controller for walking, it takes steps to solve an open area of research in the process—here we explore the problem of imposing a virtual holonomic constraint on a mechanical system with a single degree of underactuation while also stabilizing a periodic orbit within the constraint manifold[1]. This research is important when designing controllers that enable robots to perform complex, repetitive motions. Rigorous control theory applying linear control techniques have been applied to this problem in [25]. To the best of our knowledge, this is the only nonlinear approach to solve this problem. In our approach we have imbedded a stable CPG limit cycle within the zero dynamics manifold of a cart-pole system to control it in both the downward and upward positions, as the contribution of this research.

---

[1] This problem was considered an open area of research in the year 2018.

4.1 Background: Zero dynamics and Virtual Constraints

The zero dynamics of a system, the nonlinear analog to transmission zeros, are the internal dynamics created once an initial condition and input are chosen such that the output of the system is constrained to be identically zero [26]. In this case, input-output linearization is used to configure the system output as an integrator chain amenable to linear control techniques for exponential convergence to the origin. Zeroing the output of the system does not necessarily mean its zero dynamics are asymptotically stable, but only that the manifold containing the zero dynamics is invariant and attractive. The analysis of systems for which the zero dynamics are unstable is still considered substantially unexplored and an open area of research [25], [26].

The motivation for ZD control[2] is clearly stated in [27]. In summary, zeroing the outputs of the system is analogous to imposing *virtual holonomic constraints*—holonomic constraints parameterized by a state via feedback. This process can be imagined via a system mechanically constrained to move with a motorized DOF, such as a four bar linkage. The kinematic behavior in mechanically constraining a system and virtually constraining it is the same, but in the latter case, the system is easily reconfigurable. The idea that linking rigid bodies reduces the DOFs applies here, satisfying the virtual constraints effectively reduces the control problem to the dimension of the zero dynamics, which in general is considerably less than the dimension of the system.

This chapter introduces the methods used to achieve CPG-ZD control as a precursor to CPG-HZD control for closed-loop control of robot walking. In theory, the approach to control design remain intact when the two forms of control are combined. Thus it is helpful to get accustomed to the implementation of each control strategy separately. Hybridizing the controllers is simply a matter of constraining the robot to move with a CPG state rather

---

[2] The chapters of this thesis will use the term CPG-ZD and CPG-HZD to denote control for continuous dynamical systems (consistent contact with the environment) and hybrid dynamical systems (intermittent contact), respectively.

than a robot state, making combining the controllers trivial. Acquiring synergistic properties however, is non-trivial and requires careful analysis and a thoughtful approach.

4.2 Design of a CPG-ZD Controller

As long as there is a one-to-one relationship between θ, the parameterizing state in Fig. 4.1, and the robot configuration, ZD control is applicable. The simple cart-pole under this holonomic constraint satisfies the one-to-one requirement. In general, however, ZD control requires that the zero dynamics of the system are stable [27]. In the downward position, the zero dynamics of the cart-pole are *marginally stable* and in the inverted position, the zero dynamics of the cart-pole are *unstable*. In this research, we first show that in realistic conditions, ZD control fails to control the cart-pole when the pendulum is in the downward position. We then show that when CPG-ZD control is used, the cart pole can be controlled as the CPG state entrains to the zero dynamics of the system. We then further illustrate how the CPG is able to provide a *stable* limit cycle state as the parameterizing state when the pendulum is in both the downward and the upward positions.

4.2.1 The Andronov-Hopf CPG Model

The CPG dynamical model used in this research is an Andronov-Hopf (AHO) model. The AHO is a simple, planar nonlinear oscillator. Every nontrivial trajectory in the orbit of the state space converges to a single, circular limit cycle. The time courses of the state variables are sinusoidal. The amplitude and frequency of oscillation are specified by certain model parameters. The dynamical model of the AHO is given by,

$$\begin{aligned} \begin{bmatrix} \dot{\xi}_1 \\ \dot{\xi}_2 \end{bmatrix} &= \begin{bmatrix} \sigma(\xi_1, \xi_2) & \omega \\ -\omega & \sigma(\xi_1, \xi_2) \end{bmatrix} \begin{bmatrix} \xi_1 \\ \xi_2 \end{bmatrix}, \\ \sigma(\xi_1, \xi_2) &\coloneqq \mu(\alpha^2 - \xi_1^2 - \xi_2^2) \end{aligned} \tag{4.1}$$

where $\xi_i(t) \in \mathbb{R}$ for $i = 1, 2$ are the states, α and ω are the amplitude and frequency parameters, respectively, and $\mu > 0$ specifies the convergence rate. For later convenience in temporal shaping of the oscillator signal, (4.1) can be put into complex form,

$$\dot{z} = \mu(\alpha^2 - z^2)z + j\omega z \tag{4.2}$$

with $z(t) \in \mathbb{C}$ defined as $z \coloneqq \xi_1 + j\xi_2$. For complete stability analysis, the proof for global convergence to the orbit $z = \alpha e^{jwt}$ is given in [28].

4.2.2 Dynamic Equations of the Cart-Pole Model

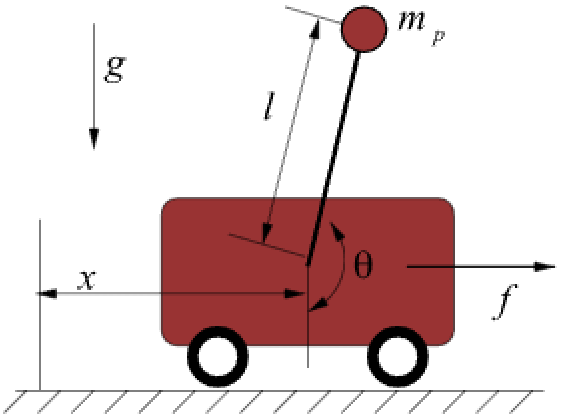


Figure 4.1: Schematic for the cart-pole problem [3].

Referring to Fig. 4.1[3], the pole is massless and the pivot is unactuated and has no friction. There is a point mass a distance l from the pivot. Theta is measured from the downward position. The corresponding equations of motion are,

$$\begin{aligned} (m_c + m_p)\ddot{x} + m_p l\ddot{\theta} cos(\theta) - m_p l\dot{\theta}^2 sin(\theta) = f \\ m_p l\ddot{x} cos(\theta) + m_p l^2\ddot{\theta} + m_p g l sin(\theta) = 0. \end{aligned} \tag{4.3}$$

---

[3] It should be noted that in [3] a nonlinear controller is used to stabilize θ about a fixed point, not an orbit as shown in our research.

To simplify the problem, we set the point mass, the mass of the cart, and the length of the pole equal to 1,

$$\begin{gathered} 2\ddot{x} + \ddot{\theta}cos(\theta) - \dot{\theta}^2 sin(\theta) = f \\ \ddot{x}cos(\theta) + \ddot{\theta} + gsin(\theta) = 0. \end{gathered} \tag{4.4}$$

It will be helpful later on to solve (4.4) directly for the accelerations,

$$\begin{aligned} \ddot{x} &= \frac{1}{1 + sin^2(\theta)}\Bigg[f + sin(\theta)(\dot{\theta}^2 + gcos(\theta))\Bigg] \\ \ddot{\theta} &= \frac{1}{1 + sin^2(\theta)}\Bigg[-fcos(\theta) - \dot{\theta}^2 cos(\theta)sin(\theta) - 2gsin(\theta)\Bigg]. \end{aligned} \tag{4.5}$$

4.2.3 Determining the Zero Dynamics of the Cart-Pole using ZD Control

In simple systems that don't require a Beziér curve to parameterize the zero dynamic state, input-output linearization and ZD control are effectively the same. Here we use input-output linearization to *theoretically* drive the cart-pole states to the zero dynamics manifold when the pendulum is in the downward position. The motion we want to achieve involves the cart rolling back and forth in the x direction as the pendulum mass remains fixed along a vertical line.

Thus, we constrain x to move as a function of theta,

$$y = x - x_d(\theta), \tag{4.6}$$

such that $y \equiv 0$ implies hat $x \equiv x_d(\theta)$. As long as $\partial x_d / \partial \theta \neq 0$, the output (4.6) is of relative degree 2. Differentiating yields,

$$\begin{aligned} \dot{y} &= \dot{x} - x_d(\theta)'\dot{\theta}, \\ \ddot{y} &= \ddot{x} - x_d(\theta)''\dot{\theta}^2 - x_d(\theta)'\ddot{\theta}. \end{aligned} \tag{4.7}$$

Plugging in the proper expression for $\ddot{x}$ from (4.3) and with $f \equiv u$,

$$\ddot{y} = 1/2(u - cos(\theta)\ddot{\theta} + sin(\theta)\dot{\theta}^2) - x_d(\theta)''\dot{\theta}^2 - x_d(\theta)'\ddot{\theta}. \tag{4.8}$$

The feedforward control,

$$u^* = cos(\theta)\ddot{\theta} - sin(\theta)\dot{\theta}^2 + 2x_d(\theta)''\dot{\theta}^2 + 2x_d(\theta)'\ddot{\theta}, \tag{4.9}$$

requires the elimination of $\ddot{\theta}$ using (4.5), for state feedback:

$$\begin{aligned} u &= u^* + v, \\ u^* &= \frac{-1}{x_d(\theta)'cos(\theta)+1}\Bigg[sin(\theta)\big[(sin^2(\theta) - x_d(\theta)'cos(\theta))\dot{\theta}^2 \\ &\quad + g(2x(\theta)' + cos(\theta))\big] - x_d(\theta)''\dot{\theta}^2(2 - cos^2(\theta))\Bigg], \\ v &= -(K_D\dot{y} + K_P y). \end{aligned} \tag{4.10}$$

When (4.10) is applied to (4.8),

$$\ddot{y} + K_D\dot{y} + K_P y = 0. \tag{4.11}$$

For $K_D, K_P > 0$, the solutions of (4.11) converge exponentially quickly to zero. For $y \equiv 0$ that is, $x \equiv x_d(\theta)$, the system's state evolves on the set,

$$\mathcal{Z} := \left\{(\theta, \dot{\theta}, x, \dot{x}) \in \mathbb{S} \times \mathbb{R}^3 \middle| x - x_d(\theta) = 0, \dot{x} - x_d(\theta)'\dot{\theta} = 0\right\}. \tag{4.12}$$

Evaluating (4.4) on the zero dynamics manifold (4.12), with u equal to u^*, yields the zero dynamics

$$(1 + cos(\theta)x_d(\theta)')\ddot{\theta} + cos(\theta)x_d(\theta)''\dot{\theta}^2 + gsin(\theta) = 0. \tag{4.13}$$

4.2.4 A CPG-ZD Approach to Determining the Zero Dynamics of the Cart-Pole

We now use a CPG state, ϕ, entrained to move with the zero dynamic state, θ, and again reveal the cart-pole's equivalent CPG-zero dynamics.

For this approach the x state is made a function of the CPG state ϕ,

$$y = x - x_d(\phi), \tag{4.14}$$

such that $y \equiv 0$ implies hat $x \equiv x_d(\phi)$. As long as $\partial x_d/\partial\theta \neq 0$, the output (4.14) is of relative degree 2. Differentiating yields,

$$\begin{aligned} \dot{y} &= \dot{x} - x_d(\phi)'\dot{\phi}, \\ \ddot{y} &= \ddot{x} - x_d(\phi)''\dot{\phi}^2 - x_d(\phi)'\ddot{\phi}. \end{aligned} \tag{4.15}$$

Plugging in the proper expression for $\ddot{x}$ from (4.3) and with $f \equiv u$,

$$\ddot{y} = 1/2(u - cos(\theta)\ddot{\theta} + sin(\theta)\dot{\theta}^2) - x_d(\phi)''\dot{\phi}^2 - x_d(\phi)'\ddot{\phi}. \tag{4.16}$$

The state variable feedback,

$$\begin{aligned} u &= u^* + v, \\ u^* &= cos(\theta)\ddot{\theta} - sin(\theta)\dot{\theta}^2 + 2x_d(\phi)''\dot{\phi}^2 + 2x_d(\phi)'\ddot{\phi}, \\ v &= -(K_D\dot{y} + K_P y), \end{aligned} \tag{4.17}$$

results in,

$$\ddot{y} + K_D\dot{y} + K_P y = 0. \tag{4.18}$$

For $K_D, K_P > 0$, the solutions of (4.18) converge exponentially quickly to zero. For $y \equiv 0$ that is, $x \equiv x_d(\phi)$, the system's state evolves on the set,

$$\tilde{\mathcal{Z}} := \left\{ (\phi, \dot{\phi}, \theta, \dot{\theta}, x, \dot{x}) \in \mathbb{S}^2 \times \mathbb{R}^4 \middle| x - x_d(\phi) = 0, \dot{x} - x_d(\phi)'\dot{\phi} = 0 \right\}. \tag{4.19}$$

Evaluating (4.4) on the zero dynamics manifold (4.19), with u equal to u^*, yields zero dynamics

$$\ddot{\theta} + (x_d(\phi)''\dot{\phi}^2 + x_d(\phi)'\ddot{\phi})cos(\theta) + gsin(\theta) = 0. \tag{4.20}$$

4.2.5 Designing the CPG Controller Using the AHO Oscillator

In this problem, we are interested in seeing whether we can make the unactuated state, θ, follow a periodic motion. The virtual holonomic constraint is introduced, $x_d = a - \mathrm{L}sin(v)$. v represents the parameterization state. $v := \theta$ in the ZD approach, and $v := \phi$ in the CPG-ZD approach. The objective is to preserve a particular point of the pendulum's rod (a distance L from the pivot point) on the vertical line $x = a$. The chosen point is L meters from the pivot.

The zero dynamics were analyzed assuming that the input u from (4.10) and (4.17) successfully drove the system states to the zero dynamics manifold. The zero dynamics for the ZD approach, or the explicit form of (4.13) is

$$(1 - Lcos^2(\theta))\ddot{\theta} + sin(\theta)(Lcos(\theta)\dot{\theta}^2 + g) = 0. \tag{4.21}$$

Thus the system state has been effectively reduced from $\mathbb{S} \times \mathbb{R}^3$ to $\mathbb{S} \times \mathbb{R}$, and stability analysis may be conducted a priori [4]. Here we analyze an example in the pendulum down position. The zero dynamics are marginally stable for $L < 1$, and orbit shape is dependent on the initial condition. In this case, $L = .9$ and $q_0 = [\pi/8, 0]^T$.

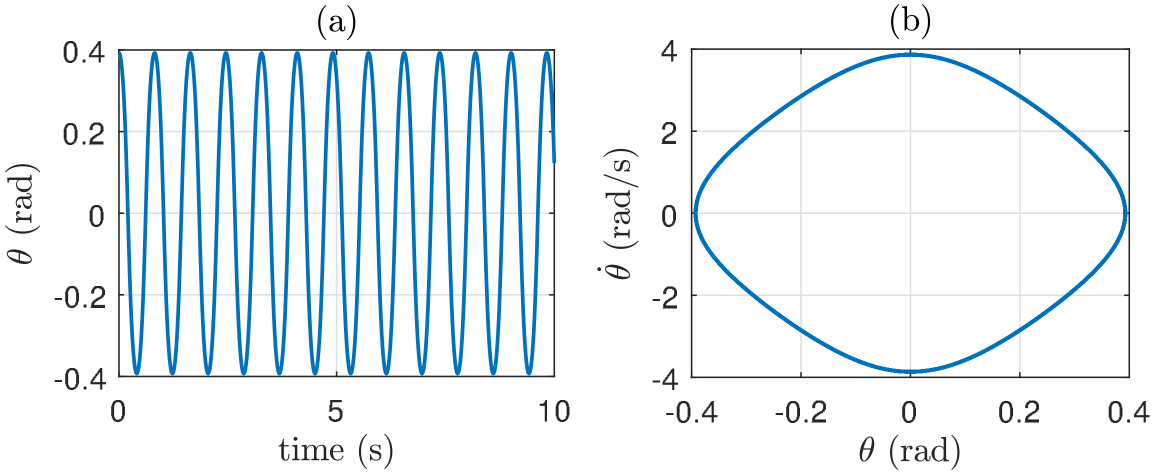


Figure 4.2: Simulation of zero dynamics for the cart-pole.

Fig. 4.2 shows the simulation of (4.21). Fig. 4.2a is the time simulation of the pivot. Fig. 4.2b is the trajectory in state space. This is the trajectory we will attempt to converge to using the CPG. It is important in controller design that the CPG state ϕ entrains to or synchronizes with the robot state θ. If it does not, then the location of the robot state in configuration space will not reside in the desired zero dynamic manifold, i.e. $q := [x, \dot{x}, \theta, \dot{\theta}]^T \in \tilde{\mathcal{Z}} \notin \mathcal{Z}$. In this event, the desired holonomic constraint x_d such that $v \approx \theta$ is no longer satisfied. To achieve entrainment, we will shape complex form of the AHO state, z. Because the time courses of the state variables are sinusoidal, any piecewise continuous temporal profile $\mathfrak{s}_i$ can be approximated by a finite Fourier series:

$$\mathfrak{s}_i(\zeta_i) \cong \sum_{l=0}^{\ell} \Re[c_{li} e^{jl\zeta_i}], \quad c_{li} \in \mathbb{C}, \quad i \in \mathbb{I}_n, \tag{4.22}$$

---

[4] It should be noted that while the zero dynamics can be isolated and analyzed, it is not worthwhile to attempt isolate them via ZD control, because in practice the system output is not identically zero for all time.

with greater accuracy the larger the number of terms ℓ. Our parameterizing state $\phi \approx \theta$, can be generated as an output that depends polynomially on z using (4.22) [28].

The design for the cart-pole was formed by 3 complex oscillators.

$$\begin{aligned} \dot{z} &= \Psi(\mid z \mid)z + Mz + \vartheta, \quad z(t) \in \mathbb{C}^n, \quad M \in \mathbb{C}^{nxn}, \\ \Psi(\mid z \mid) &:= \mathrm{diag}(\mu(\alpha_i^2 - |z_i|^2 + jw). \end{aligned} \tag{4.23}$$

where M is a matrix coupling individual subsystems $\dot{z}_i = \psi(\mid z_i \mid)z_i$ with $i \in \mathbb{I}_n$. The input ϑ is defined as,

$$\vartheta = \mathrm{diag}(\Phi - \Theta)L, \quad \Phi \in \mathbb{R}^{n\times 1}, \quad \Theta \in \mathbb{R}^{n\times 1}, \quad L \in \mathbb{R}^{n\times 1}. \tag{4.24}$$

where $\Phi := [\ \phi\ \dot{\phi}\ \ddot{\phi}\ ]^T$, $\phi_i = \mathfrak{s}_i(z_i)$, and $\Theta := [\ \theta\ \dot{\theta}\ \ddot{\theta}\ ]^T$[5] and L is a vector of gains for adjusting entrainment properties. From this section we see that designing the CPG is extremely straightforward with the help of holonmic constraints. The holonomic constraints provide the limb coordination and the stability of the full dimensional system is contingent only on stabilizing the zero dynamics. Designing the CPG simply requires building a Fourier series to track the zero dynamics, then tuning once connected to the full dimensional system.

4.2.6 Simulating the Cart-Pole Using ZD Control

We will now attempt to force the system state $q \in \mathcal{Z}$ by *holonomic constraint*, i.e. by simulating $q \in \mathbb{S} \times \mathbb{R}^3$, and imposing the constraint $y = x - x_d(\theta)$ from 4.6. With this method of control, constraining the cart to move with the θ state is not enough to keep it on its desired trajectory. In other words, theoretically ZD control will work to keep the cart on a periodic trajectory by constraining it to the marginally stable state θ. But practically speaking, the cart-pole and its environment are non-deterministic, and any small perturbation will cause instability because the zero dynamic state is not stable and attractive.

Fig. 4.3a shows the unactuated pivot state, analogous to the robot's unactuated ankle in the hybrid case. Fig. 4.3b shows the trajectory in state space. Fig. 4.3c shows the actuated

---

[5] Acceleration states are only used for sensory feedback to the CPG.

cart trajectory, the holonomic constraint, and the error $y = x - x_d(\theta)$. The cart state is analogous to the actuated joints in the hybrid case. Fig 4.3d is a closeup of the error, y. These plots show that while there were no errors on initial condition or disturbances imposed on the cart-pole, errors arising from discrete simulation were enough to make it go unstable. Referring to (4.11), $K_D,\ K_P = 1e3$, and $q_0 = [\ \pi/8,\ 0,\ -.338,\ 0]^T$.

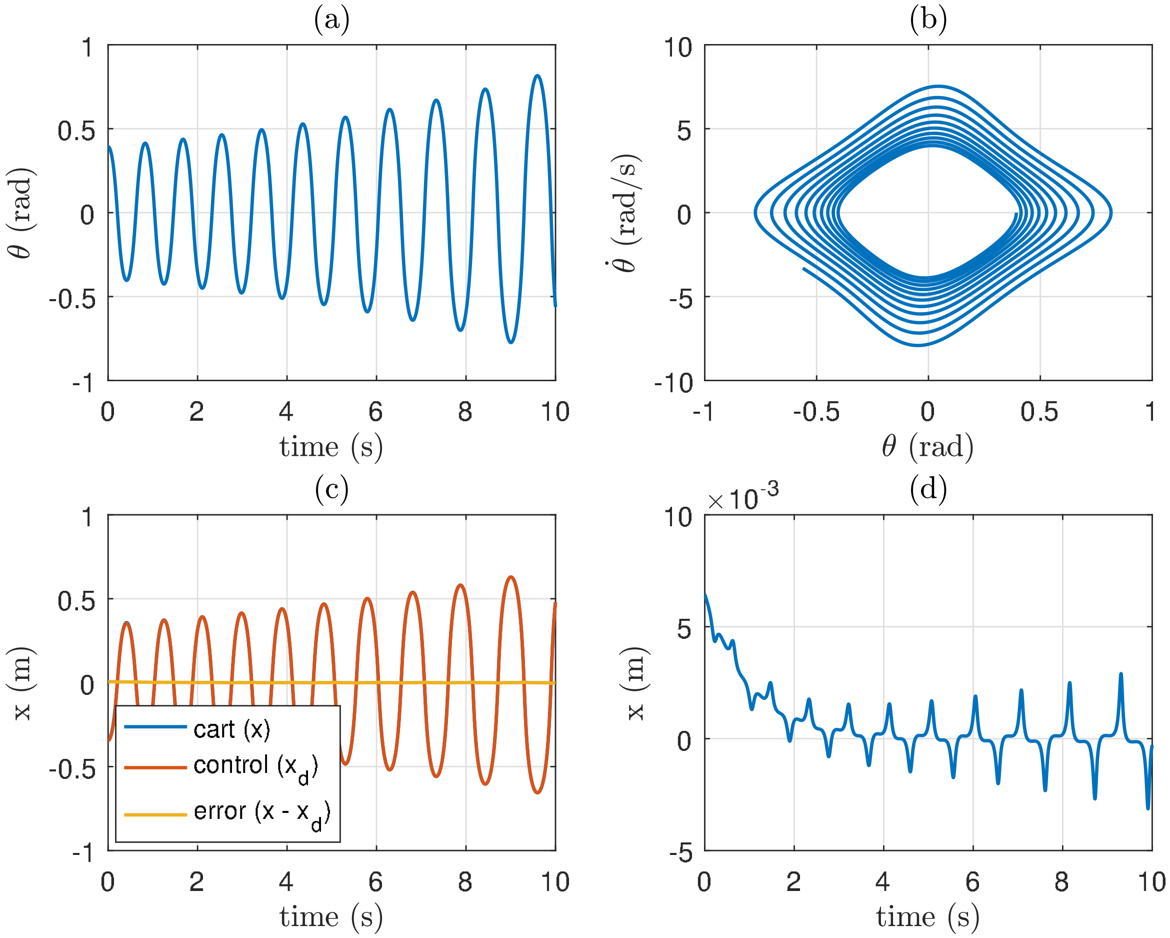


Figure 4.3: Simulation of the cart-pole under ZD control.

4.2.7 Simulating the Cart-Pole Using CPG-ZD Control

4.2.7.1 Pendulum Down Position

We will now attempt to force the system state $q \in (\tilde{\mathcal{Z}} \cap \mathcal{Z})$ in the same manner as described in Sections 4.2.4 and 4.2.5. It is not straightforward to do. Satisfaction of the desired holonomic constraint for a given input and initial condition—the cart approximately tracks the desired trajectory—can result in bifurcation or convergence to orbit of different amplitude or center than that given by (4.13). Fig. 4.4 shows the simulation using CPG-ZD control with .1 radians (5.7 degrees) of error in initial condition imposed upon the pivot.

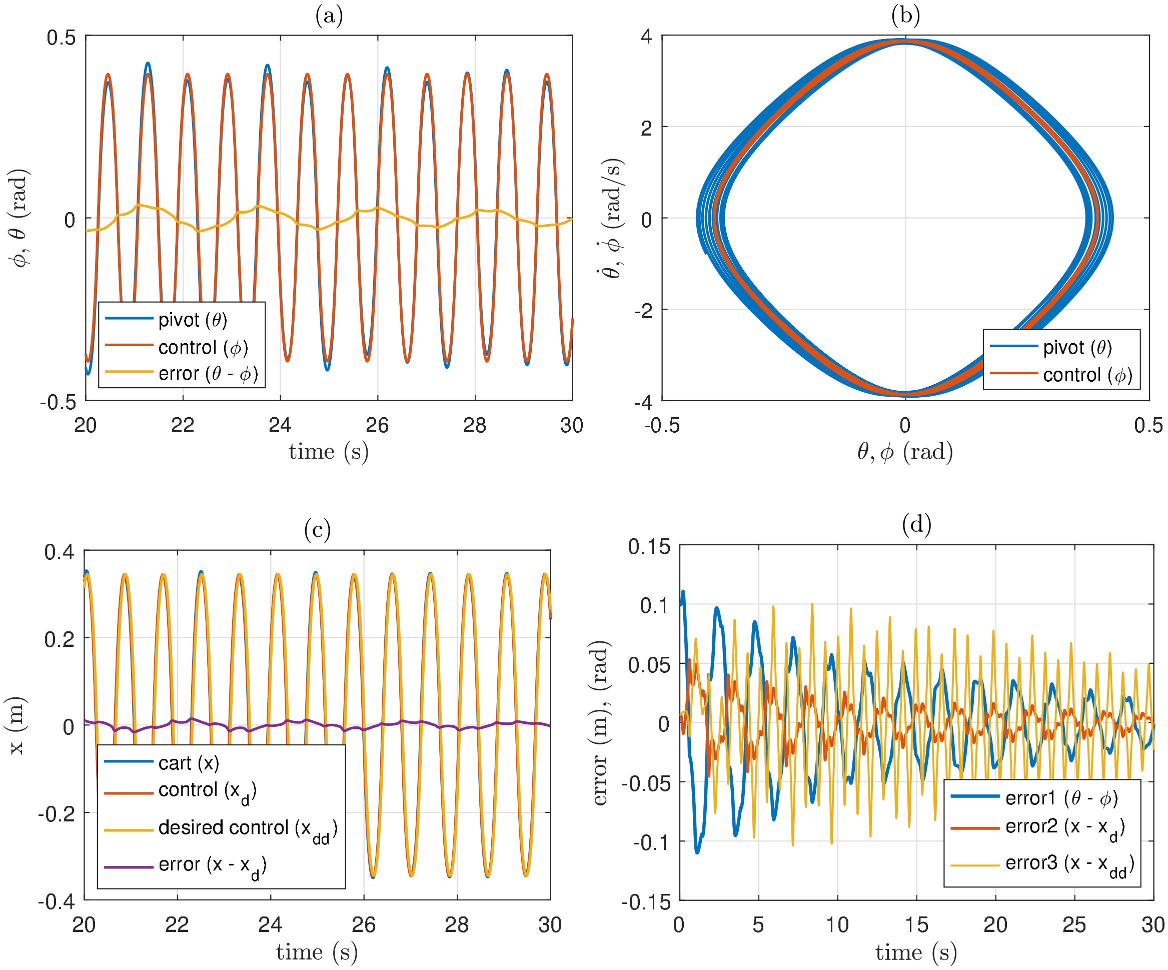


Figure 4.4: Simulation of the cart-pole under CPG-ZD control with a .1 radian initial condition error imposed on the pivot.

Fig. 4.4a shows the pivot, θ, the parameterization CPG state ϕ, and the error between the states. Entrainment occurs as the two states converge. Fig. 4.4b shows the trajectory in state space of the pivot and the CPG states. The first two signals in Fig. 4.4c are the actuated cart trajectory x and desired trajectory, x_d. The third signal, x_{dd}, is the "ideal" desired trajectory using (4.21), which differs from x_d because its orbit is fixed. It is not subject to the dynamics of the cart or the CPG.

Using methods from Section 4.2.5, the CPG signal $\phi = \mathfrak{s}(z)$ was shaped to track the desired θ trajectory. It is clear from Fig. 4.4c that the holonomic constraint using ϕ and θ track quite well. The purple signal is the holonomic constraint, $y = x - x_d(\phi)$. Fig 4.4d is a closeup of the error signal from subfigures a and c, respectively. The yellow error signal in Fig. 4.4 shows that while the cart state converges to the desired trajectory that is a function of the fixed orbit, it converges more slowly than to the desired trajectory that is a function of the actual zero dynamic state given by the CPG, ϕ. This indicates that like a "cart driving the horse", the erroneous θ state is pulling the CPG state off its orbit slightly. This is a natural product of CPG entrainment where the oscillator naturally adapts to the dynamics of a robot. Again referring to (4.18), K_D, $K_P = 1e3$, $q_0 = [\ \pi/8 + .1,\ \ 0,\ \ -.338,\ \ 0]^T$, $z_0 = [\ 1\ 1\ ]^T$. The CPG parameters (4.23) $\omega = 7.67$ rad/s, $\alpha = 1$, $L = [\ 2\ 2\ 2\ ]^T$ and $\mu = 1e4$.

4.2.7.2 Pendulum Up Position

Fast controller convergence is required when the cart-pole in the inverted position. This is a function of the undamped dynamics of the system and the fact that the zero dynamics are unstable. Intuitively, this is more difficult because the cart is autonomously "bootstrapping" stability as it uses its motion to stabilize a state that it is constrained to follow. The effect of the θ state on the cart dynamics is not as pronounced in the CPG-ZD controller because the unstable zero dynamics interact with the CPG rather than directly impacting the cart dynamics. Regardless, the controller dynamics must be at least as fast as the zero dynamics when they are subject to gravity as a destabilizing force.

In Fig. 4.5 an empirical approach was taken to stabilize the pole in the inverted position a distance 1.5 meters from the pivot point. A form of integral control was included, and feedback from the cart in addition to feedback from the pivot were sent to the CPG for entrainment. A 5.7 degree error was imposed upon the pivot at $t = 0$. Fig. 4.5a shows the pivot θ and the CPG control ϕ entraining, Fig. 4.5b shows the trajectory in state space, Fig. 4.5c shows the cart first stabilizing the pole then driving the output y to zero, Fig. 4.5d is a visualization of the cart-pole over 1 period, color coded to show evolution of time (cool to warm).

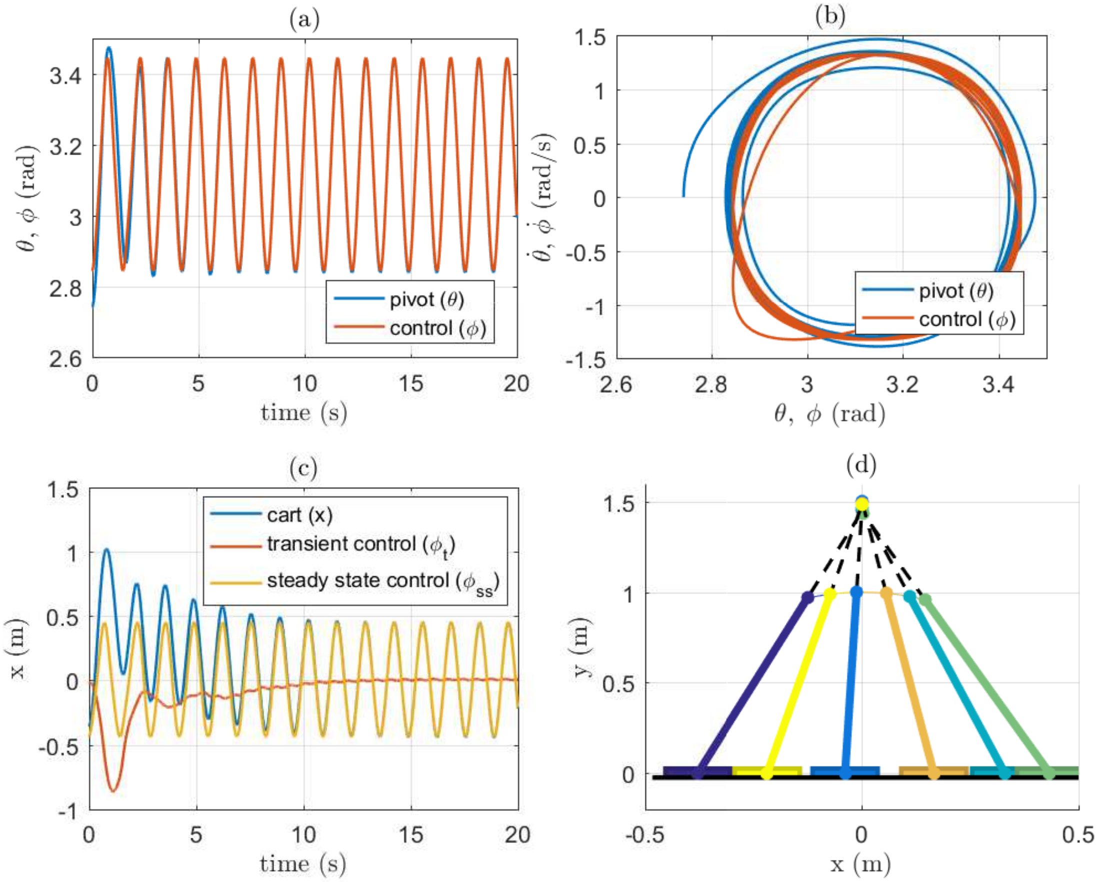


Figure 4.5: Cart-Pole simulation with the pole in the inverted position.

In the inverted cart-pole experiment, $K_1 = [\,350\;400\,]$, $K_2 = [\,400\;300\,]$ $q_0 = [\,2.74,\ 0,\ -.4371,\ 0]^T$, $z_0 = [\ 1\ \ 1\ ]^T$. The CPG parameters (4.23) $\omega = 4.7346$ rad/s, $\alpha = 1$, $L_1 = [\ 15\ \ 15\ \ 15\ ]^T$,

$L_2 = [\ 3\ 3\ 10\ ]^T$ and $\mu = 1e3$. K_1 represents the holonomic constraints, K_2 represents the gains on integrated error, L_1 is the gain on sensory feedback from the pivot and L_2 is the gain on sensory feedback from the cart.

In this chapter we have shown that a CPG-ZD controller can stabilize marginally stable and unstable zero dynamic orbits. In future work, refinement of the CPG controller behavior and peformance optimization studies will be done. This would involve analysis of CPG feedback mechanisms and why entraiment to both θ and the actuated cart state x was necessary in the inverted case. If it needs feedback from the actuated states in order to produce synergistic stabilizing effects, this will present a scaling problem. Currently as the DOFs scale up, the dimension the CPG is 2. For a dim(N) robot, 2N signals cannot be sent back to the CPG without having a deleterious effect on sensory feedback performance. The CPG cannot decipher a blend of 2N signals, and thus many CPG modules would have to be made to accommodate the increase in signals. This raises more questions like how this sort of decoupling of signal feedback affects the control of a highly coupled system. In circumstances where very fast convergence is required, it would be useful if these questions were addressed.

CHAPTER 5

Extension of CPG-ZD Control to the Hybrid Dynamical Case

In theory, the approach to control design remains intact when CPG-ZD control from Ch. 4 is extended to the hybrid dynamical case. Like CPG-ZD control, integrating CPG and HZD controllers is simply a matter of constraining the robot to move with a CPG state rather than a robot state—combining the controllers is trivial. Acquiring synergistic properties, however, is not trivial and requires careful analysis and a thoughtful approach. This chapter will first consider closed-loop CPG control for hybrid systems, then it will explore HZD control for non-anthropomorphic side walking.

5.1 Background: Hybrid extension of CPG Control

5.1.1 Closed-Loop CPG Control

Using a CPG with sensory feedback has benefits such as limit cycle stability and entrainment. As stated in Ch. 4, entrainment is the ability to synchronize with natural dynamics. Limit cycle walking is defined as walking that does not require local stability at every point along a robot's trajectory, but is stable because a given trajectory will eventually converge to the desired trajectory over a number of steps [24]. As a result, high gain control is not needed, permitting the use of natural dynamics.

CPGs can be modeled by networks of nonlinear oscillators like the Matsuoka oscillator or the AHO. CPGs have been successful in simulation, but fewer studies have been done with robots [16]. Endo et al. had positive results with their in-plane biped robot, but

CPGs have had more success with quadrupeds such as Salamandra Robotica and Cheetah-Cub [17], [29]. Quadrupeds can navigate uncertain terrain using a CPG because they have a large base of support. Always having two or more feet on the ground aid in keeping the body upright. There is not a standard framework in place for implementation of the CPG on robots, especially bipedal robots. This is in part due to the complexities of coupled nonlinear oscillators that ultimately require empirical methods for positive results. Furthermore, CPGs are generally used for joint-space, kinematics-based control that focuses on stabilizing joints individually. In a quadruped decoupling DOFs is not a large problem because of their inherent stability. In a biped that requires *active stabilization*, this decoupling can render the robot unstable.

There are significant studies using a CPG in bipedal walking that directly relate to the research of this thesis. In 2003 Yamasaki et al. analyzed the basins of attraction arising from holonomically constraining a robot to move with the phase of a CPG [30]. When the state of the robot was pushed outside the basin of attraction, a phase reset was issued to return the state to the basin. Yamasaki also noticed that phase resetting reduced convergence time of the oscillator. This research is similar to the CPG-HZD method. In this case however the robot was fully actuated, and trajectories were synthesized using human subjects. This means that limit cycles were not induced by design of the controller, but that they worked with the stability properties that arose from obtaining holonomic constraints from human subjects. Stability analysis for the full dimensional system was also required.

Nakanishi et al. (2004) successfully implemented a CPG on a 5 DOF planar robot via imitation learning of human motion primitives to generate natural walking [31]. They did not holonomically constrain the limbs, but used a phase oscillator for each joint with a fixed phase relationship to a reference oscillator that received sensory feedback. Their argument for the reference oscillator over directly introducing a phase reset as in [30] was that sensory feedback allows phase estimation based on events and multi-modal information. They also increased the autonomy of the controller by developing frequency adaptation based on phase resetting and entrainment. In our research the reference oscillator accepting sensory feedback is analogous to the parameterization state that is entrained to the robot state.

5.1.2 Hybrid Extension in CPG Control Using the Simplest Walker

The CPG is a continuous dynamical system. Hybrid extension requires that the CPG's phase be reset at impact. This phase reset means the CPG will begin at the zero phase point of its orbit. Because the CPG has a fixed temporal waveform, it is a single attractive orbit in phase space, hybrid invariance is guaranteed. To illustrate the hybrid extension of the CPG model, the Simplest Walking Model is used [4].

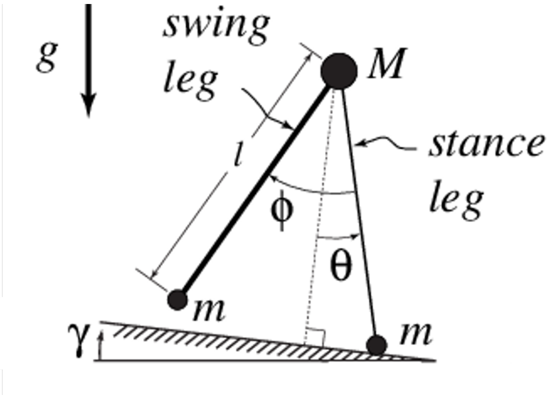


Figure 5.1: Simplest Walker Model with Labeling [4].

Referring to Fig. 5.1, the CPG control effort is applied to actuate the hip, with associated generalized coordinate ϕ. The equations of motion, for an approximation making the feet lighter than the center of mass are the following,

$$\begin{aligned} &\ddot{\theta} - \sin(\theta - \gamma) = 0 \\ &(\cos(\phi) - 1)\ddot{\theta} + \ddot{\phi} - \dot{\theta}^2 \sin(\phi) - \sin(\theta - \phi - \gamma) + \frac{u_r}{ml^2} = 0 \end{aligned} \tag{5.1}$$

Where γ is defined as the slope angle of the ground. The continuous dynamics are used until heelstrike, where a transition rule applies to account for the impact of the foot with the ground,

$$\begin{pmatrix} \theta \\ \dot{\theta} \\ \phi \\ \dot{\phi} \end{pmatrix}^{+} = \begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & \cos 2\theta & 0 & 0 \\ -2 & 0 & 0 & 0 \\ 0 & \cos 2\theta(1-\cos 2\theta) & 0 & 0 \end{pmatrix} \begin{pmatrix} \theta \\ \dot{\theta} \\ \phi \\ \dot{\phi} \end{pmatrix}^{-} \tag{5.2}$$

When the control loop is closed, as in Ch. 4, the AHO produces a dynamic trajectory for the robot to track,

$$\dot{z} = (1-|z|^2)z + j\omega z + u_c \tag{5.3}$$

In Eqs. 5.1 and 5.3, the inputs u are defined as follows,

$$\begin{aligned} u_r &= K(s(z) - \phi) \\ u_c &= L(s(z) - \phi) \end{aligned} \tag{5.4}$$

5.1.3 CPG Entrainment and Phase Resetting

In Fig. 5.2, the Simplest Walker is simulated using traditional forward walking. The passive walker was sent down a decline of .009 rad with an initial condition of $q = [\ .194\ .338\ -1.4275\ -.16016\ ]^T$. States represent the positions of the stance leg, θ, and inter-leg angle, ϕ, respectively, and their velocities. This initial condition represents a small perturbation from an open-loop stable walking: $q = [\ .2003\ .4006\ -1.3996\ -.11108\ ]^T$. For brevity, model parameters are omitted. The perturbation causes the robot to stumble while converging to the orbit. The blue curve represents the CPG state ϕ_{C_x} and the yellow curve represents the robot state θ.

In Fig. 5.2a the CPG state is kept separate from the robot state as it stumbles around in the first few steps. Without sensory feedback, the CPG does not respond and a phase lag occurs. In Fig 5.2b, sensory feedback (u_c in (5.4)) is fed back to the CPG and it entrains, i.e. it synchronizes with the robot state. In Fig. 5.2c, the robot is also holonomically constrained with nonzero proportional feedback. The CPG entrainment is not enough, it requires too many steps to sync with the state ϕ. We see here that closing the loop destabilizes the robot. Fig. 5.2d shows the closed loop system with proportional control and phase resetting. To reset the phase $z = 1$ is the initial condition after impact, representing zero phase. In this experiment the robot is not destabilized. For the single controlled state L is the gain for entrainment and K the proportional control. In 5.2a L=0 and K = 0, in 5.2b L = -5 and K = 0, in 5.2c L = -5 and K = 1, in 5.2d L = -5 and K = 1 and z is reset to 1 after each impact.

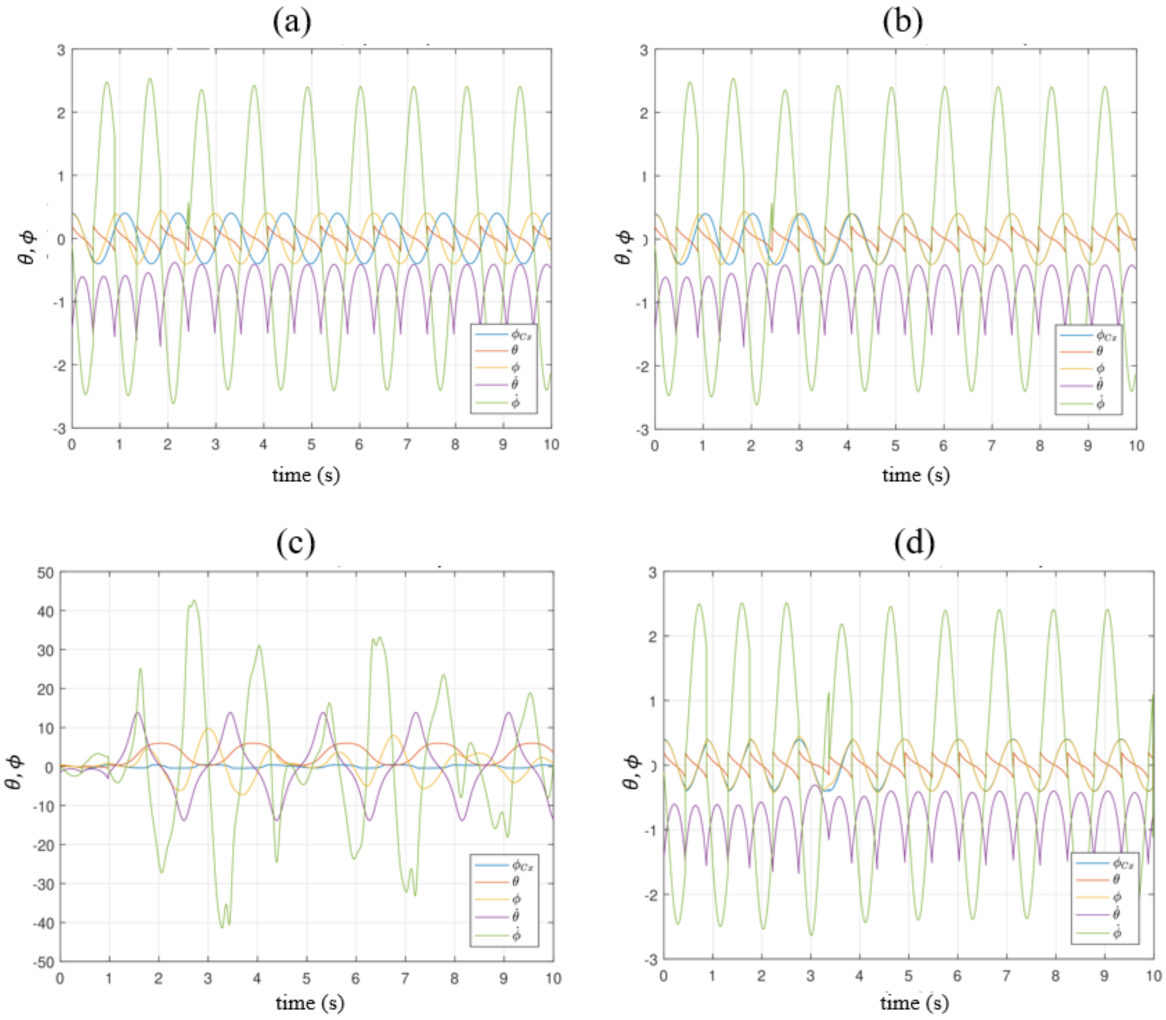


Figure 5.2: Plots showing CPG entrainment and phase resetting.

5.1.4 Using the AHO with P Control

If I set θ_0 to a value of .196 radians, the robot cannot recover and walk stably. In open-loop, robot walking is unstable.

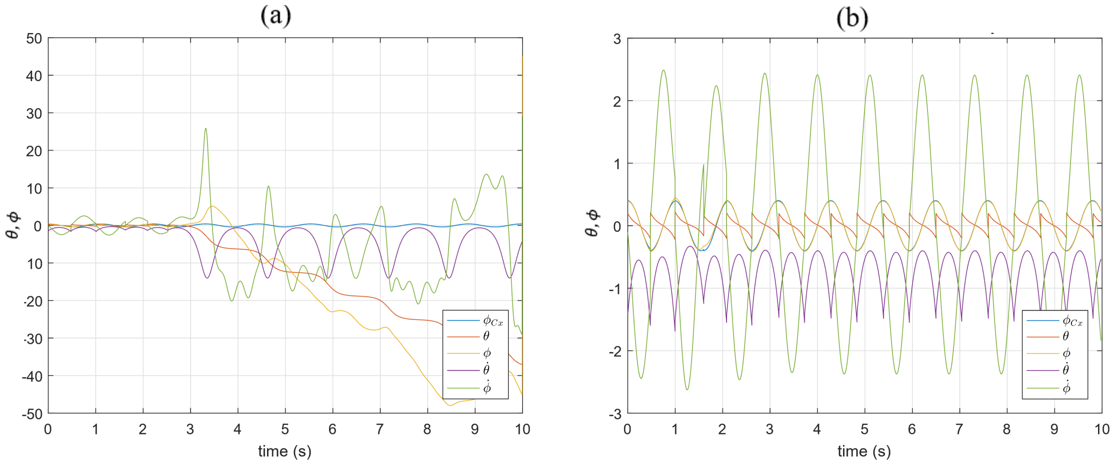


Figure 5.3: Simplest Walker parameters for $\mu = 1$, $\theta_0 = .196$, L= 0, K= 0 (left) and L= -5, KD= 1 (right).

When using phase resetting, Fig. 5.3b shows under closed-loop CPG control, the robot is successfully stabilized.

5.1.5 Using the AHO with PD Control

For PD control, two Andronov-Hopf oscillators (AHO) produce a dynamic trajectory for the robot to track, for $z \in \mathbb{C}^2$. The inputs u are defined as follows,

$$
\begin{aligned}
&\dot{z} = \phi(z)z + jwz + u_c, \quad \phi(z) = \begin{bmatrix} (1 - |z_1|^2) \\ (1 - |z_2|^2) \end{bmatrix}, \\
&u_r = K_p(\mathfrak{s}(z_1) - \phi) + K_D(\mathfrak{s}(z_2) - \dot{\phi}), \\
&u_c = \begin{bmatrix} L_P(\mathfrak{s}(z_1) - \phi) \\ L_D(\mathfrak{s}(z_2) - \dot{\phi}) \end{bmatrix},
\end{aligned}
\tag{5.5}
$$

and $\mathfrak{s}(z)$ is the shaping filter [28].[1]

---

[1] Note the oscillators have not yet been connected to make a network.

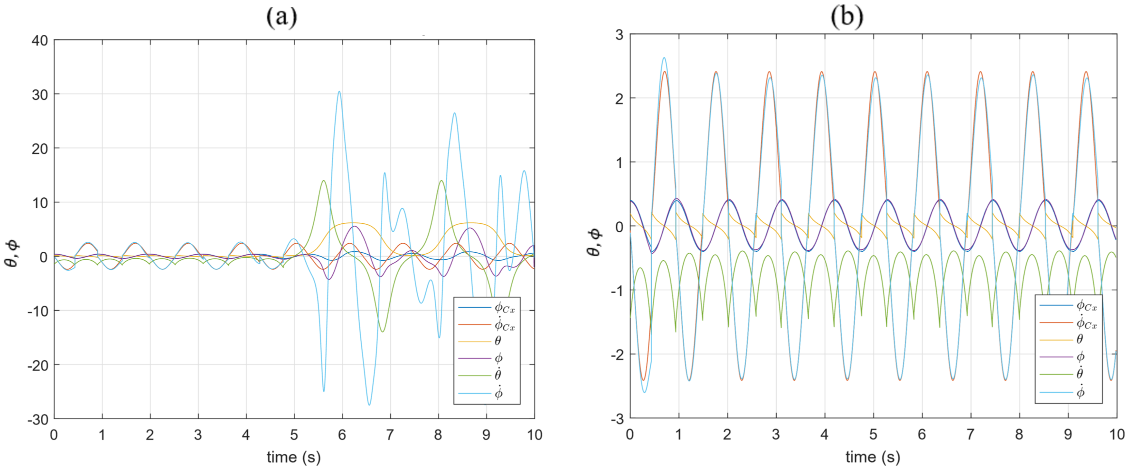


Figure 5.4: Simplest Walker parameters for $\mu = 1$, $\theta_0 = .192$, LP= -3, KP= -.5, LD= 0, KD= 0 (left) and LD= .02, KD= -.005 (right).

When θ_0 is set to a value of .192 radians, the robot cannot recover and walk stably in open-loop or with position control alone (Fig. 5.4a). However, when the velocity controller is added and empirically tuned, the walking gate is stabilized 5.4b.

In this section, we illustrated the CPG's ability to adapt to natural dynamics through entrainment and hybrid extension of the CPG via phase resetting. Using the Simplest Walker model, the CPG controller can stabilize walking gates that are otherwise unstable under certain initial conditions. This control can be thought of as CPG-HZD control for two reasons: firstly, in a two state walking model the states are holonomically constrained by definition, and secondly, using a "hybrid" dynamical CPG state via phase resetting we are able to drive the hybrid robot to a stable periodic orbit. It would be interesting to compare the stability regions around stable orbits using CPG-HZD control versus HZD control alone for this system.

5.2 Background: Hybrid Zero Dynamic (HZD) Control

In 2003 Grizzle et al. introduced the HZD method to control RABBIT, a planar pair of robotic legs with point feet [5]. The feet are a point of underactuation, the interaction of the foot with the ground behaves like a revolute joint. The reason for studying robots with point feet were two-fold: 1) like Raibert they were interested in proving that walking and running motions were achievable on a mechanically simple robot and 2) underactuation of this kind is analogous to rotation of the foot about the toe, and the HZD-method sought to break from flat-footed walking of their time for a more human-like gait.

Point feet use dynamic walking gaits because without a base of support (i.e. contact convex hull), there will always be a net moment on the body. Westervelt et al. identified several challenges associated with dynamic walking [27]. Principle among them were *limb coordination*, which must be sophisticated enough to maintain balance despite static instability, and *the design of limit cycles* which are necessary to achieve periodic and in a sense "stable" walking, but difficult to induce in high DOF, hybrid systems. The HZD method uses the theoretical framework of nonlinear zero dynamics to create holonomic constraints to address limb coordination. Satisfying these constraints effectively reduces the control problem to the dimension of its zero dynamics, which makes the search for and stability analysis of limit cycles much simpler. Holonomically constraining the robot effectively eliminates the controller's dependence on time; it is an autonomous form of control. The objective is to eliminate the need to resynchronize with time to overcome a disturbance. It is more important to track an orbit in state space, phasing error should not be a factor in stability.

The HZD-method of control has had much success since its inception on RABBIT. Chevallereau et al. extended the controller to 3D bipeds, Poulakakis et al. produced a compliant HZD controller for MABEL's series elastic actuation, Sreenath et al. successfully achieved stable running on MABEL, Ames et al. scaled the controller to achieve walking on the high DOF system DURUS, and Grizzle et al. is currently focused on non-deterministic HZD control on CASSIE to achieve high mobility in unstructured environments [32], [33], [34], [35].

Despite the ongoing interest in the HZD-method of control, it has not yet been implemented on any form of non-traditional walking like side-walking. This may be due in part to the requirement that the parameter used to holomically constrain robot walking be *strictly* *monotonic*. To eliminate time, and thus configurational ambiguity in the controller strict monotonicity is necessary. If the parameter tracks the rotation of the stance leg in forward walking this constraint is satisfied. In non-traditional walking the same parameter may not be monotonic. A solution for this issue is to hybridize the control method with one that is also autonomous, such as the CPG.

5.3 Augmenting HZD Control for Side-walking

The NABi robot is a good candidate for HZD-style control because like Rabbit, it has 1 degree of underactuation per leg. The large feet and passive ankle function exactly like point feet in the sagittal plane, but offer roll support in the frontal plane. The main issue with the use of HZD-style control is the requirement that the parameter used to eliminate time be strictly monotonic. Intuitively there is not a “natural” way to find such a parameter in non-traditional walking like side-walking. Perhaps through an exhaustive search of virtual geometry there may be a strictly monotonic parameter, but HZD design using a CPG should circumvent the monotonicity constraint because the CPG uses an AHO. The AHO has a strictly monotonic phase implicit in its state. The phase of the AHO would ensure that limbs maintain a phase relationship with each other—all its signals are phase-locked. This is the property that makes CPGs time-invariant, which makes it a fitting partner for time-invariant HZD control. If the AHO state entrains to the usual robot parameter used in HZD control, θ, then by way of the AHO the limbs are phase-locked to θ. Depiction of an intuitive virtual support leg is shown in Fig. 5.5, with q representing generalized coordinates, and $\theta = q_1 + q_2/2$.

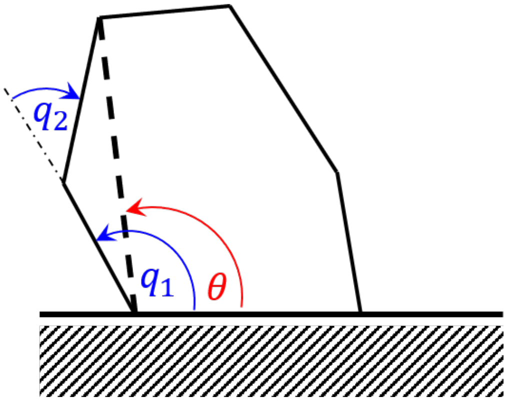


Figure 5.5: Intuitive choice for parameter θ.

Using this as the parameterizing state reduces stability analysis to this state alone. Intuitively, stabilizing this state means active stabilization of the entire body because the robot will remain in an upright position. A random set of CPG trajectories were commanded to the NABi robot in simulation. The parameter θ was tracked throughout the walking sequence and the results are shown in Fig. 5.6. Fig. 5.6a shows trajectories from the front leg and Fig. 5.6b from the back. It is clear that the intuitive choice for a parameterizing state for side-walking is not monotonic.

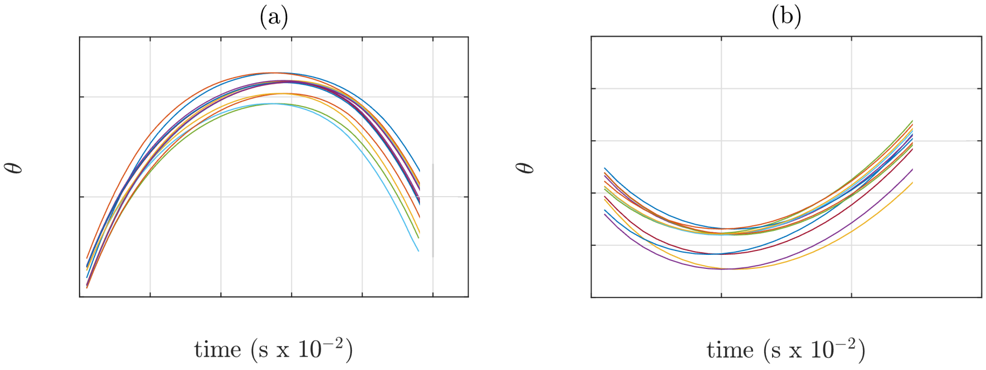


Figure 5.6: Plot NABi θ parameter over 12 steps.

Other constraints and assumptions on the walking gait are thoroughly covered in [27], Sec. A.2, and Sec. A.3.

CHAPTER 6

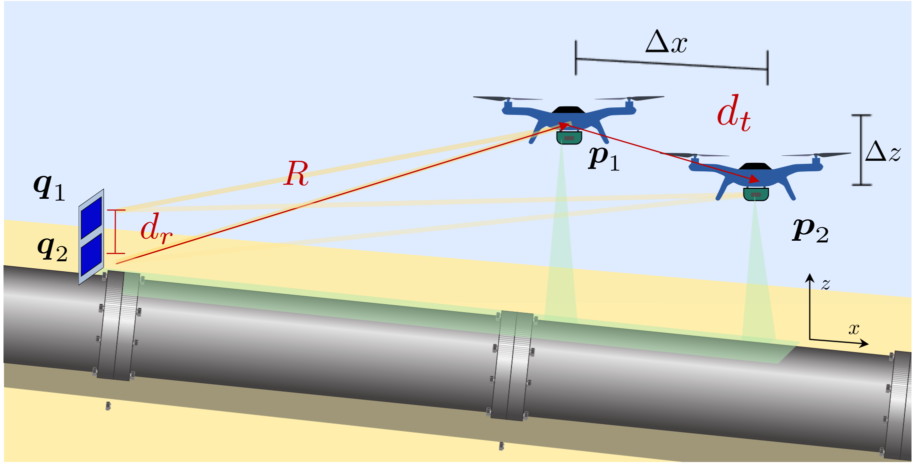
Enhancing Communication Through Spatial Awareness

6.1 Background: UAVs for Communication and Sensing

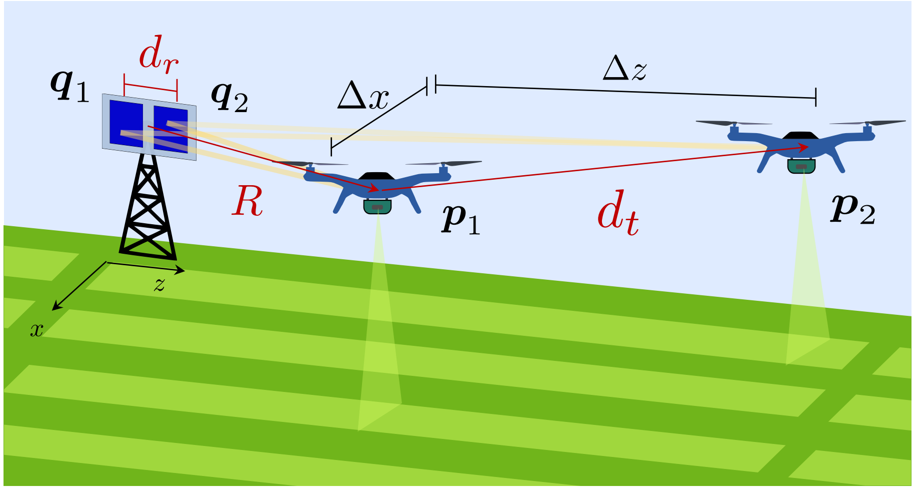
In recent years there has been considerable focus on deployment of multi-robot teams for cooperative field missions [36]. To work around issues of intermittent connectivity and limited communication bandwidth robots must operate autonomously when disconnected, periodically seeking the location of high-strength communication links to transmit collected data and receive updates from the base station [37–43]. This limits the ability for continuous data streaming throughout a sensing task. Real-time streaming is particularly necessary in applications where autonomous operation is currently unsolved or would require more computation than is reasonable to carry on the robots, such as in tracking and surveillance or human-driven inspection tasks. Other applications, in particular search and rescue missions, may also require real-time streaming due to the time critical immediacy of the underlying objective. Several approaches to maintaining connectivity for all time are possible; many relevant robotics works in path planning, however, pose sensing and capacity as a multi-objective optimization problem subject to a static channel map [44, 45].

In this research, we employ spatial multiplexing (SM), a distributed multi-input multi-output (MIMO) communication technique [46–48]. The SM communication system attempts to increase the communication capacity (i.e. data throughput) by using multiple transmit and receive antennas to send unique data streams simultaneously. SM is typically used in cluttered (or rich scattering) environments because independent fading aids in decoding signals at the receiver. SM can provide throughput gains in line-of-sight (LOS) environments as well, but requires care in placement of antennas in addition to typical signal processing for decoding streams [49–52]. Work has been done to create virtual antenna arrays using unmanned aerial vehicles (UAVs) as dedicated network enhancers for signal relay applications, among others [53–56]. In this work we use the same principles given in [49–52] to create *mobile* virtual arrays capable of coordination within a cluster to enhance network link quality while simultaneously moving and collecting data. SM can significantly enhance the quality of network links subject to path loss to facilitate improved real-time data aggregation in LOS-dominant areas, i.e. areas where other causes of intermittent signals such as shadowing or multipath fading are not pronounced. In this work we mitigate data stream correlation caused by interference among robots by positioning transmitters such that communication channels have an optimized phase offset. We do so by constraining robot transmitters to uniform linear arrays (ULA) [49, 50]. In largely unobstructed areas, boustrophedon flight plans are the standard for aerial coverage missions [57, 58]. In this work we show the flight planner can enjoy significant capacity gains by integration of ULA configurations into pre-existing boustrophedon plans, with little effort in adoption. Futhermore, our ULA adaptation ensures that the ground coverage of the sensing field of view is a superset of the coverage set of the prior unmodified trajectories.

Fig. 6.1 shows an example of two situations where ULA adaptation is used to increase communication throughput. Robots maintain an inter-transmitter distance, d_t, for a total distance of $(m-1) \cdot d_t$ from the 1^{st} to the m^{th} robot in the direction $\vec{d_t}$, where M is the total number of robot transmitters. Inter-transmitter distance changes with respect to the x-coordinate of the 1^{st} robot. In Fig. 6.1a robots adjust the altitude differential prescribed by d_t as they collect data along a path. This configuration captures applications such as powerline [59–62], bridge [63–67], and pipeline infrastructure inspection [68–70]. In Fig. 6.1b robots maintain constant altitude while d_t changes within the horizontal plane containing the receiver antennas. This approach is appropriate for applications requiring real-time transmission in coverage missions such as search and rescue [45, 71] and surveillance [72–74].



(a) When sensing data is distributed along one dimension, transmitters can change their altitude to generate separation Δz to improve communication capacity.



(b) When sensing data is distributed along two dimensions, transmitters maintain a fixed altitude and travel with separation Δz within the horizontal plane to improve communication capacity.

Figure 6.1: Two example scenarios of a multi-receiver, multi-transmitter system where communication-aware techniques are applied to improve transmission throughput in time-critical sensing missions.

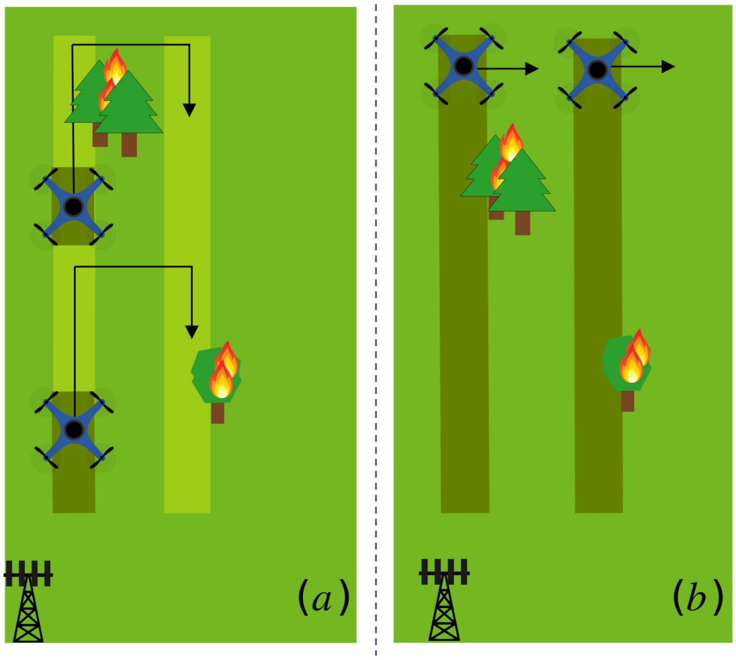


Figure 6.2: Two possible paths transmitters may take without changing the mission objective. Black arrows indicate direction of travel. The robots traverse the dark green paths in 1 hour. Because robots minimize interference in (b), they travel much farther.

6.1.1 Motivating Example

In remote missions with MIMO wireless infrastructure, robot teams may sacrifice available throughput via naïve positioning. In a UAV coverage mission, there are several boustrophedon possibilities; considering various options can mitigate interference and make a substantial difference in the rate of data transmission.

Using the communication model given in Sec. 6.2.2, we create the example depicted in Fig. 6.2 where UAVs sweep an area of interest without a particular spatio-temporal objective, such as in the monitoring of a forest fire. The UAV transmitters in scenario (a) are subject to nearly identical (highly correlated) SM channels due to poor positioning. UAVs achieve an average communication capacity of 2.41 bps/Hz and 1.51 bps/Hz for the closer and more remote transmitters respectively. In scenario (b) the UAVs reorient with respect to receivers. Here both transmitters achieve approximately the same average capacity of 7.08 bps/Hz. This means that on average they can transmit 3.6 times the data over the same time interval.

Assuming we always meet sensing requirements, we motivate our work using time as a performance metric. For a clear side-by-side comparison of MIMO methods, we enforce the even distribution of data over a spanned area. Transmitter velocity is defined as a function of available capacity: transmitters may only move as fast as the rate at which they can transmit data in real-time. Reassessing the previous example, the dark green colored portion of paths represent the sensing area covered by UAVs in 1 hour. In scenario (a) the UAVs collectively cover 554 m of ground while in scenario (b) the UAVs cover 2 km. It is clear then that scenario (b) is preferable, and at no cost to the sensing objective.

6.1.2 Contributions

This research develops the integration of ULA formations with flight plans for real-time data aggregation tasks. To the best of the authors' knowledge, this is the first work to incorporate SM MIMO approaches based on the analysis of complex communication signals into a mobile, simultaneous sensing and transmitting multi-robot system. This research characterizes the advantages of LOS SM approaches as as tool for communications enhancement, and serves as a foundation for solving SM-aware path plans for sophisticated sensing objectives. Results showing up to a 42% improvement in task completion times in analytical examples and 29% in realistic simulation further demonstrate the effectiveness of this research.

6.2 Problem Setup

M robots are simultaneously sending independent data streams using SM communications to N fixed ground receivers, where $N \geq M$. We examine two tasks where behavior is constrained to an in-plane Cartesian workspace $\mathcal{W} \subset \mathbb{R}^2$. In all cases, robot velocity is throttled by available capacity in the "sensing direction". In the one-dimensional (1D) sensing case robots travel in series, and collect data along a line. Robots may change altitude at any velocity to improve communication throughput, but may not move along their sensing path faster than they can transmit data. This scenario is depicted in Fig. 6.1a. In the two-dimensional (2D) sensing case data is distributed in two directions; thus robot velocity is constrained in both directions even when robots change proximity to one another along the z-direction shown in Fig. 6.1b to improve communication throughput.

6.2.1 Motion Model

At time instant t, the m^{th} transmitter is located at $p_m(t)$ and the n^{th} receiver is located at q_n. Mobile transmitter motion is defined by a first-order, discrete-time, kinematic model, $p_{m_{i+1}} = p_{m_i} + v_{m_i} dt_i$, where $p_m,\ v_m \in \mathbb{R}^2$ are the transmitter position and velocity for transmitter m respectively. The propagation time from state i to $i+1$ is the scalar value dt_i. All solutions are found using a grid-based approach consisting of equally spaced waypoints.

6.2.2 Communication Model

We derive our model based on perfect knowledge of a path LOS-dominant channel, an approximation suitable for open spaces with minimal obstruction. When a transmitter sends the data symbols $x \in \mathbb{C}^M$, the received symbols will be as follows,

$$\mathbf{y} = \mathbf{H}x + \mathbf{n} \tag{6.1}$$

where $\mathbf{H} \in \mathbb{C}^{N \times M}$ is the channel matrix, and $[\mathbf{H}]_{n,m}$ represents the channel between the m^{th} transmitter and the n^{th} receiver. $\mathbf{n} \in \mathbb{C}^N$ is a Gaussian random variable modeling the additive white Gaussian noise. One way to measure the amount of information that can be sent over a communication channel is the capacity. For a MIMO channel, the capacity is given by,

$$C = \log\left(\det\left(\mathbf{I} + \rho \mathbf{H}^H \mathbf{H}\right)\right), \tag{6.2}$$

where ρ is the signal-to-noise-ratio of the signal.

In this research, our focus is on scenarios that occur in an open environment, where the communication channel is assumed to be dominated by the line-of-sight (LOS) component.

The elements of the channel matrix in that case can be modeled using,

$$h_{n,m} = [\mathbf{H}]_{n,m} = \frac{\lambda}{4\pi \|p_m - q_n\|} \exp\left(j \frac{-2\pi \|p_m - q_n\|}{\lambda} \right), \tag{6.3}$$

where $\|p_m - q_n\|$ is the distance between the transmitter and the receiver and λ is the wavelength. The transmitters are assumed to be sending independent data streams. Each transmitter does not have the data of the other transmitter, hence, all the processing has to be done at the receiver. At the receiver side, we assume that linear minimum-mean-squared-error MMSE combining is used. When using MMSE combining, the maximum rate obtained by the mth stream is given by [48]

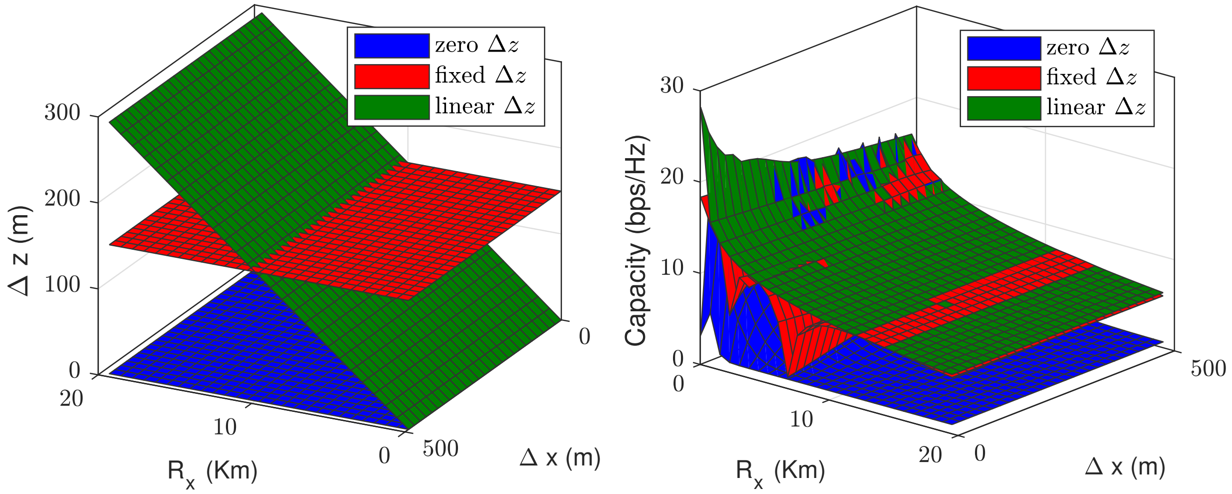
$$C_m = \log\left(1 + P_m \mathbf{h}_m^H \left(N_0 \mathbf{I} + \sum_{i \neq m}^{M} P_i \mathbf{h}_i \mathbf{h}_i^H \right)^{-1} \mathbf{h}_m \right), \tag{6.4}$$

where P_m is the transmit power of the m^{th} transmitter, N_0 is the noise power spectral density, $\mathbf{h}_m$ is the m^{th} column of $\mathbf{H}$, and $(\cdot)^H$ denotes the Hermitian transpose. In this analysis we assume that all transmitters have the same power P , i.e, $P_m = P$ for all m.

6.3 Effects of Transmitter Position on the MIMO Network

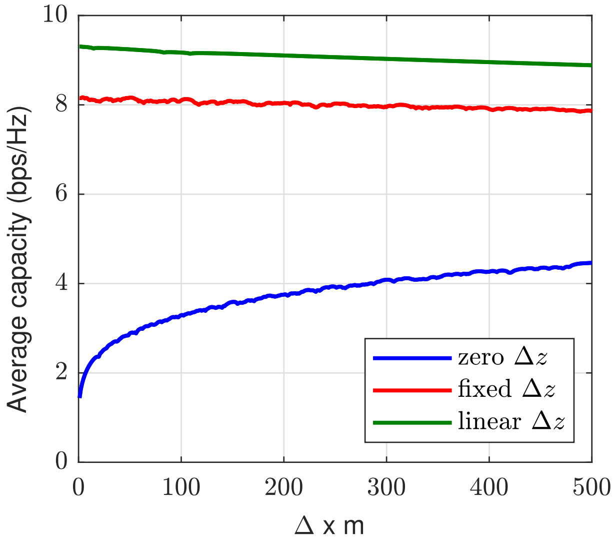
In the LOS environment, changing path length with respect to receivers effects the channel and capacity according to (6.3), (6.4) respectively. If robots are transmitting far away from receivers, small displacements have little impact on channel magnitude, which can be approximated by $\|h_{m,n}\| \approx \frac{\lambda}{4\pi R}\ \forall\ m, n$. R in this equation approximates $d_{m,n} = \|p_m - q_n\|$ defined in (6.3), constraining robots to a small cluster. Under these conditions we generate uncorrelated channels by positioning robots to achieve streams with a $360° \cdot 1/M$ phase offset to a given receiver. Formally, the eigenvalues of expression $\mathbf{H}\mathbf{H}^H$ should be nonzero to achieve a high rank LOS channel matrix and high capacity, which can be achieved by consideration of channel phase alone.

In the adaptation of path plans we use heuristics from [49,50] to pose the communication problem as a *mobile* ULA. A ULA assumes at a given way point, robots are equally spaced.



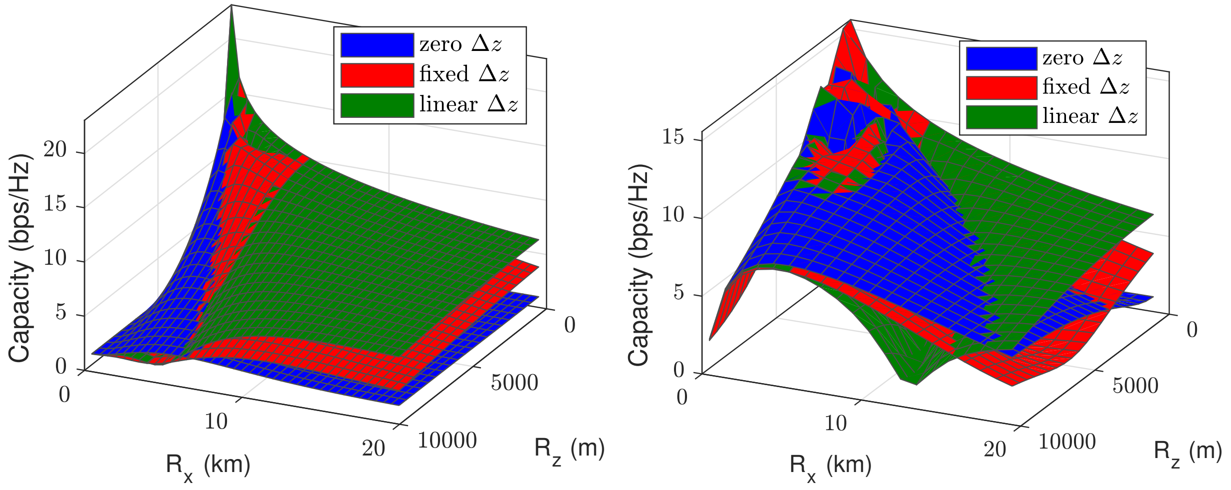
(a) Communication parameter Δz is plotted as a function of distance in the x-direction from receivers and the sensing parameter Δx. Parameter Δz varies with R_x only.

(b) Capacity is plotted as a function of distance in the x-direction from receivers and the sensing parameter Δx. Here $R_z = 0$.



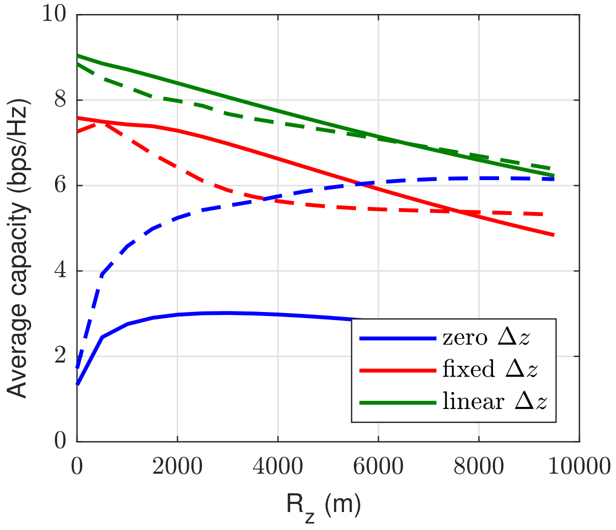
(c) Plot depicting average capacity over total x-distance traveled for varying values of Δx.

Figure 6.3: Plots showing the desired Δz and resulting capacity calculated for 3 SM path planning approaches for a 2 transmitter, 2 receiver system where travel is in the x-direction with varying ground distance between receivers, Δx. The plot in (a) also corresponds to Fig. 6.4.



(a) Capacity is plotted for as a function of distance from receivers in the x and z-directions using a boustrophedon path plan for the value $\Delta x = 50$.

(b) Capacity is plotted for as a function of distance from receivers in the x and z-directions using a boustrophedon path plan for the value $\Delta x = 500$.



(c) Plot depicting average capacity over total x-distance traveled and lateral distance R_z from receivers.

Figure 6.4: Plots showing the capacity calculated for 3 SM path planning approaches in a 2 receiver, 2 transmitter system where travel is in the x and z-directions using boustrophedon path plans of differing transmitter separation, Δx.

Specifically, signal orthogonality can be achieved through state-dependent phase adjustment,

$$\sum_{m=0}^{M-1} \exp\left(jm\frac{-2\pi d_r \Delta z}{\lambda R_x}\right) \;=\; 0. \tag{6.5}$$

This phasing relationship implies the relation,

$$\frac{d_r \Delta z}{R_x} = \frac{\lambda}{M} \tag{6.6}$$

between adjacent robots. The above relation generally applies when R is large, and when arrays are *parallel*, i.e. when R is equal to its projection R_x and $d_t = \Delta z$. In this research we apply (6.6) to scenarios outside of these constraints, and find it remains useful when constraints are relaxed.

For simplification, we analyze the capacity of transmitter 2, C_2, using the $N \times M = 2 \times 2$ case. We assess relative performance of three SM approaches to path planning using (6.4). In all approaches $d_r = 5$ m. Increasing d_r serves to decrease the separation Δz required to achieve a high rank H. For all parameters used to evaluate capacity, see Table 6.1. The first of three approaches is a naïve baseline where $\Delta z = 0$ m, the second is a method with a fixed value of $\Delta z = 150$ m, and the third uses the linear relation (6.6). In each approach $p_1(t) = f(R(t))$ is specified by the path planner. The location of the second robot transmitter, $p_2(t)$, is then fully defined by the sensing parameter Δx and the communication parameter Δz, thus $p_2(t) = f(R(t), \Delta x, \Delta z)$.

We analyze each method in Figs. 6.3 and 6.4. From (6.6) we see that Δz is a function of the x-position of robot 1, R_x (Fig. 6.3a). Fig. 6.3b maps C_2 as robot transmitters move along a line 20 km in the x-direction, sweeping over transmitter ground separation, $\Delta x \in [0,\ 500]$ m. The plot illustrates the effect of changing the Δx parameter on capacity in the path following 1D scenario. Here Δx can be small if sensor redundancy is required, or UAVs performing heterogeneous sensing tasks travel in a tight cluster. Otherwise large ground separation can allow for robots to collect data independently. To better approximate a parallel array system in the 1D case, if receivers are on the ground, transmitter 1 moves at ground level, i.e. $R = R_x$. As ground distance from receivers increases, we see attenuation of capacity from path loss. We also see slight attenuation as transmitter ground separation, Δx, increases. This is expected given we are relaxing the small cluster and parallel array relations.

Figs. 6.4a and 6.4b refer to the 2D scenario, where sensing requirements generate motion in both the x and z-directions. Two values of Δx were chosen, 50 and 500 m, for the purpose of demonstrating the effect of the parallel array and small cluster relaxations. The plots illustrate the effect of motion in the z-direction that causes receiver and transmitter arrays to become "less parallel". The map is generated via the calculation of C_2 as robot transmitters move along a boustrophedon path (see Fig. 6.1b). The robots move in the x-direction, R_x, by the same distance as the shown in the 1D case. They move 10 km in the z-direction, with robot 2 moving outside the sensing region of interest (ROI) to satisfy the Δz requirement (see Sec. 6.4 for explanation). With receivers located at the origin, the plots give us information about a square 20×20 km region due to symmetry. As expected, we see the attenuation of C_2 as R becomes large. To address parallel ULA constraints, methods accounting for R_z in (6.6) should be applied [51].

All capacity maps demonstrate the effectiveness of the mobile ULA, or linear Δz method in mitigating interference issues in broad areas of transmitter travel. This is also demonstrated in Figs. 6.3c and 6.4c. In Fig. 6.3c, methods are compared based on the average capacity over the entire distance of robot travel, for each value of Δx. We see in this figure that because R_z was constrained to be zero, the more "parallel" relationship between transmitter and receiver arrays resulted in clearer performance gains. Fig. 6.4c shows the average capacity value for total travel in the x-direction over the increasing interval $[0,\ R_z]$. The solid line corresponds to Fig. 6.4a and the dashed to Fig. 6.4b. The convergence in dashed lines as R_z increases shows the compounding effect of a large $\frac{d_t}{R}$ ratio, in addition to large R_z.

6.4 Motion Planning

Using the scenario from Sec. 6.2, time is tabulated as a performance measure for different SM approaches in a 2×2 MIMO system. SM methods from the previous section are applied, and the widely adopted time-division multiple access MIMO approach (TDMA) is also included [46]. Sec. 6.4.1 uses analytical equations given by the communication model while Sec. 6.4.2 uses realistic channel data taken from a ray tracing simulation. Robots are constrained to move at the same velocity, and the previously defined motion model is used to calculate the total time to complete a task, $T = \sum_{i=0}^{n-1} dt_i$.

6.4.1 Motion Planning Using the LOS Equation

Temporal solutions to more specific path plans using the general concepts covered are given in this section. Simulation parameters are listed in Table 6.1.

Table 6.1: Simulation Parameters

Parameter	**Definition**
bandwidth	1 MHz
data density	25 Mb/m
sensing velocity	$C_2/25$ m/s
carrier frequency	1 GHz
transmission power	20 dBm
d_r	5 m
Δx	50, 150, 300, 450 m
1D R_x, R_z	$R_x \in [1, 3]$ km, $R_z = 70, 220$ m (depending on method)
2D R_x, R_z	$R_x \in [1, 3]$ km, $R_z = [0, 500]$ m
grid resolution	10 m

6.4.1.1 1D Path Planning

Adjustment to 1D path plans are made using the four SM approaches–zero Δz, fixed Δz, linear Δz, and TDMA. In the first, both robots are constrained to have the same altitude. Under this constraint, empirically, performance increases with an increase in altitude—$R_z =$

220 m and $\Delta z = 0$. The second maintains a fixed altitude differential, $R_z = 70$ and $\Delta z = 150$ m. The third uses the linear approach, where $R_z = 70$ m and Δz varies according to (6.6). Depending on the needs of the application, we assume these altitude constraints—amounting to focal distance changes in an imaging application, for example—can be accommodated. Waypoints for the TDMA approach are equivalent to the zero Δz method. The capacity for this approach is given by,

$$C_m = f_m \log\left(1 + \rho \mathbf{h}_m^H \mathbf{h}_m\right). \tag{6.7}$$

It assumes maximal ratio combining, f_m is the fraction of time allocated to UAV m. Results are shown in Table 6.2. Results are consistent with previous analysis, however, because robots maintain an altitude above the receiver array ground position as they travel, we expect some changes in performance, namely the degradation of capacity gains for the linear method.

Table 6.2: Trial Times (hr) and Gains

Case	**Δx (m)**	**TDMA**	**Zero Δz**	**Fixed Δz**	**Linear Δz**	**Gain %**
1D	50	2.172	2.116	1.704	**1.086**	36.27
	150	2.196	1.455	1.676	**1.109**	23.78
	300	2.232	1.273	1.689	**1.163**	8.64
	450	2.266	**1.307**	1.739	1.356	-3.75
2D	50	11.415	15.695	11.351	**6.521**	42.55
	150	3.806	3.380	3.764	**2.456**	27.34
	300	2.208	1.670	1.935	**1.644**	1.56
	450	1.681	**1.228**	1.340	1.351	-10.02

In Table 6.2 gain values listed are computed gain $= \frac{T_a - T_l}{T_a} \times 100$. The linear Δz method is denoted by subscript l versus lowest alternative method a.

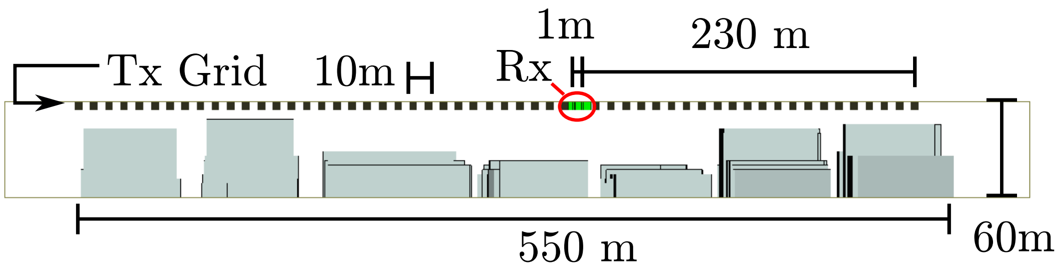
6.4.1.2 2D Path Planning

The same four Δz constraints from 1D planning are used to compute solutions to boustrophedon complete coverage flight plans. A sensing ROI consists of all waypoints defining the zero Δz method. If a z-offset occurs between robots, one robot will travel outside the ROI. An example of a boustrophedon for the linear Δz method is shown in Fig. 6.6a. Similar to 1D planning, this research assumes path plan augmentation can be accommodated. We choose a region where the upper bound of R_z is kept proportionately low, from previous analysis we expect the linear method to be effective. Table 6.2 demonstrates the ROI generates large performance gains by the linear method when inter-robot spacing is lower. Given transmitter travel of $[1, 3]$ km in the x-direction is not large, performance loss by increasing Δx is pronounced.

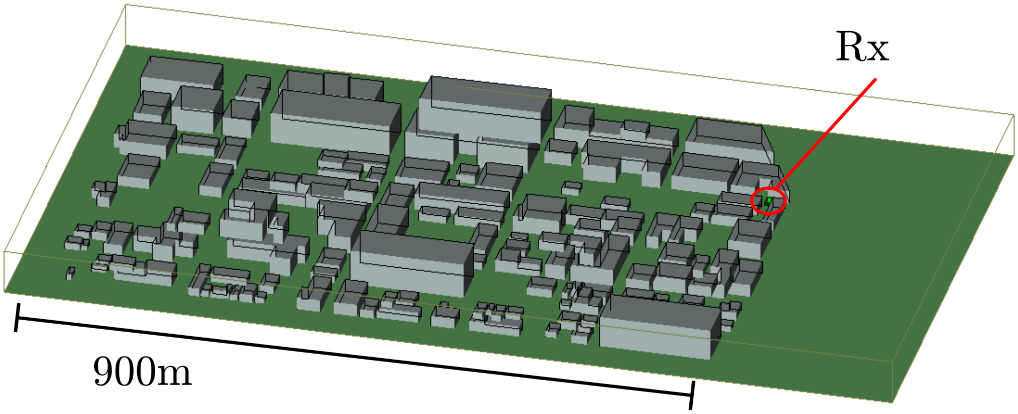
6.4.2 Evaluation under realistic conditions

To determine how MIMO SM methods perform under realistic conditions, the channel was simulated using ray tracing software [75]. The ray tracing simulation accounts for reflected paths due to the environment in addition to LOS. A dataset was generated for an urban environment in the city of Ottowa shown in Fig. 6.5. The transmitters and receivers were placed on a plane at a height of 60 meters, above all buildings. This setup emulates a base station placed on a tower receiving data from UAVs. The channel was simulated using a grid of transmitters spanning the city with a separation of 10 m communicating to two receivers 1 m apart located 320 meters west of origin. Capacity was then calculated according to (6.2) as 2 transmitters were positioned at points on the grid.

The path plan for the case where the distance between boustrophedon rows, Δx, is equal to 50 m is shown as a visual example in Fig. 6.6a. The robots begin at $z = 0$ and move in a back and forth “lawnmower” motion. The boxes indicate an assumed sensing field of view. The dotted lines show the robot trajectories, where it is assumed robots are not imaging as



(a) South view of simulation environment. Transmitters and receivers are on the same plane at a height of 60 m above all buildings. The transmitters were simulated on a grid with separation of 10 m, while the receivers have a separation of 1 m.

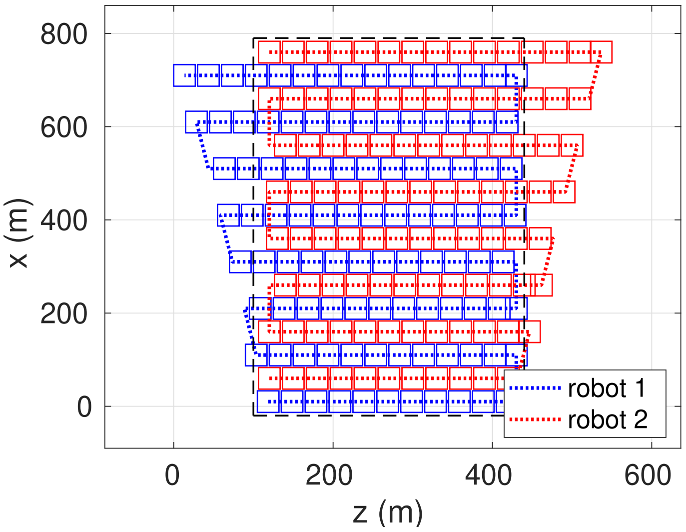


(b) South-east view of the environment showing the simulated city.

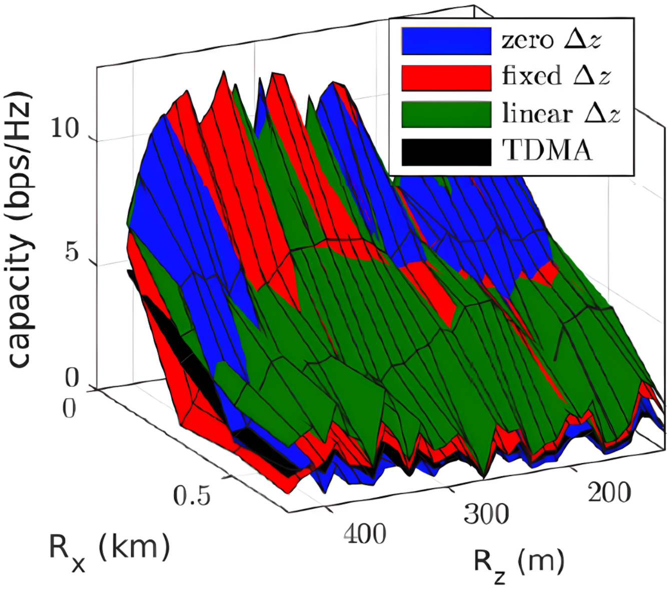
Figure 6.5: The ray tracing simulation was performed in an urban environment spanning 550 m × 900 m. The simulated channel accounts for reflections occurring due to buildings along with the LOS path.

they travel from one boustrophedon row to the next. The dashed black rectangle is the ROI, spanning 775 m in the x-direction and 350 m in the z-direction. It is assumed all robots with nonzero Δz offset were continuously imaging, thus their velocity was throttled even when travel outside of the ROI occurred.

Data from trials for varying separation values between transmitters is shown in Table 6.3. The fixed $\Delta z = 30$ m for all trials. A 30 m distance was determined empirically as a fixed separation that performed well over trials, but further study is required to determine an optimal separation. From the simulation results it is observed that realistic channel



(a) Two robots move in a boustrophedon with a row separation of 50 m. The dashed rectangle is the sensing ROI, all methods cover this region. The linear Δz method is shown.



(b) Capacity is calculated using a ray tracing simulation of city conditions for a boustrophedon path plan with $\Delta x = 100$. The receivers are located at $z = 320$ m.

Figure 6.6: A boustrophedon map corresponding to $\Delta x = 50$ m, for the linear Δz method is shown (top); this method is one of four MIMO approaches to coverage planning in a realistic city environment. Results for all four methods are shown for a $\Delta x = 100$ m (bottom).

Table 6.3: Trial Times (hr) and Gains

Δx (m)	**TDMA**	**Zero Δz**	**Fixed Δz**	**Linear Δz**	**Gain %**
12.5	33.31	43.57	30.52	**29.95**	1.87
25	15.56	19.63	**12.75**	16.41	-28.71
50	9.16	13.73	7.87	**6.94**	11.82
100	5.35	7.00	5.32	**3.74**	29.70

properties result in lower channel magnitudes and higher mission completion times. For the $\Delta x = 25$ m case, the fixed method proves best by a wide margin. Capacity for the case of $\Delta x = 100$ m is shown in Fig. 6.6b. We see in this plot the same trends for the capacity of the linear Δz method, where performance gains are highest as R increases. Across trials, we see an increase in relative performance as Δx increases. This may be due to the geometry of the coverage problem, i.e. fewer instances when robots must travel outside the ROI and less lateral travel in the z-direction.

6.4.3 Consideration of z-direction in Motion Plans Will Improve Results

In this work we analyzed the benefit of SM methods for MIMO network enhancement in data aggregation tasks. By characterizing the capacity within a workspace we identified trends impacting communications throughput for a path following and coverage task. Based on this analysis we identified regions where ULA heuristics for static array design can improve capacity in a mobile sensing task. We validated our approach via simulations using both analytical and realistic approaches. In the future we plan account for motion in the z-direction of travel [51, 52]. This extension allows for inclusion of more sophisticated sensing tasks while still achieving optimized communication.

CHAPTER 7

Reducing Computational Complexity Using the BOEM-SLAM

7.1 Background: A Block Online Approach to SLAM

To interact with the world, a robot must establish a spatial relationship between itself and its surroundings. Simultaneous localization and mapping (SLAM) enables a robot to intelligently and autonomously obtain these spatial relationships [76–78]. After 30 years of extensive study, sophisticated visual-inertial SLAM algorithms like ORB-SLAM [79–81] and VINS-Mono [82] have achieved unprecedented performance. However, the computational complexity of these systems, particularly in the backend, remains a challenge that limits their efficiency and scalability [76].

Typically, a visual-inertial SLAM system is divided into a frontend and a backend. The frontend abstracts sensor data and provides a local trajectory estimation. Due to problems of inevitable drift associated with frontend estimation, the backend takes the results from the frontend and computes a globally consistent solution using loop closure. Due to the computational complexity of inferring a globally consistent solution, the backend is the major factor in determining the time it takes to run a SLAM system [76].

This research proposes to fundamentally rethink the backend formulation to alleviate the computational burden without sacrificing performance. We define an *online* SLAM process as one that can discard historical data while still producing an optimal solution. In contrast, an *offline* backend must store the complete time history of data to achieve optimal inference. By focusing on the underlying problem formulation rather than incremental computational or frontend improvements, we aim to develop a truly efficient and scalable backend solution.

The computational bottleneck in current backend algorithms stems from formulating SLAM as a maximum likelihood (ML) trajectory estimation problem, leading to a nonlinear least squares (NLS) optimization that grows in complexity over time (the comprehensive discussion for offline SLAM algorithms are provided in Section II of [83]). While techniques like iSAM2 [84, 85] can incrementally update the solution in many cases, they must still solve the complete problem under certain loop closure conditions. Other methods resort to approximate solutions by pruning less informative nodes, sacrificing solution optimality for efficiency [76, 79, 81].

To enable an online backend, we propose modeling SLAM as an ML problem of landmark positions with the trajectory as latent variables, following the EM-SLAM framework [86]. By recognizing the general hidden Markov model (HMM) structure of SLAM, we can apply the block online EM (BOEM) algorithm [87] to develop BOEM-SLAM. This approach summarizes historical data into landmark estimates and the most recent state distribution, allowing each iteration to operate on non-overlapping blocks and discard past statistics.

While the BOEM algorithm has been applied in simple 2D scenarios [88] and wireless sensor networks [89, 90], to the best of our knowledge, this work is the first to extend BOEM-SLAM to 3D visual-inertial systems. By tackling the backend complexity at its core, we aim to develop a truly online, efficient, and scalable SLAM solution that maintains optimal performance.

We apply BOEM-SLAM to 3D visual-inertial systems using a monocular camera and inertial measurement unit (IMU), and address several critical implementation issues that were insufficiently covered in the original EM-SLAM paper [86]. First, as the E-step of the BOEM algorithm involves Kalman filtering and Rauch-Tung-Striebel (RTS) smoothing, we expand these filtering methods to incorporate rotation estimation using a Lie group representation. This is crucial for accurately estimating the 3D pose of the robot in a visual-inertial system. Second, to handle outliers in visual measurements that can greatly deteriorate SLAM performance, we incorporate an outlier removal approach similar to the robust cost functions used in optimization-based methods. By addressing these key challenges, we adapt the BOEM algorithm to the specific requirements of 3D visual-inertial SLAM, ensuring its robustness and accuracy in real-world scenarios.

7.1.1 Implementation Details in the Visual-Inertial System

We now turn to the implementation aspects of BOEM-SLAM developed by Chang et al. in [83]. For reference, the relevant portions of this work are repeated here. The calculation of the conditional expectation in the E-step is typical in filtering and smoothing problems, which are typically solved by the Kalman filter and the RTS smoother. One of the main challenges is that we have to take the manifold structure of the rotation into consideration. The Kalman filer on Lie group is well-developed [91], while the RTS smoother on Lie group receives little discussion. We then follow the derivatives on Lie groups in [92] to design the RTS smoother on the Lie group.

The state space $\mathbb{S} = S^3 \times \mathbb{R}^3 \times \mathbb{R}^3$ is the direct product of three Lie groups, and thus it is also a Lie group. The group multiplication, the exponential and logarithm maps, and even the plus and minus operations, are all well-defined in $\mathbb{S}$ by inheriting from individual Lie groups. We use $\bar{s}_{t|t'}$ for conditional expectation of s_t given $o_{1:t'}$, and $\Sigma_{t|t'}$ for the associated covariance matrix. With the derivatives on Lie groups, we can then use the following iterative equations

$$\bar{s}_{t|T} = \bar{s}_{t|t} \circ \exp\left(C_t\left(\bar{s}_{t+1|T} \ominus f(\bar{s}_{t|t}, u_t)\right)\right), \tag{7.1}$$

and

$$C_t = \Sigma_{t|t} F (F \Sigma_{t|t} F + Q)^{-1}, \tag{7.2}$$

for the estimate update in the RTS smoother, while the covariance update remains identical to the original RTS smoother.

Another key factor in the SLAM performance in the visual feature outlier detection and removal. In the optimization-based approaches, robust cost functions are often employed to reduce the effect of outliers, for example the Huber loss function. In our experiments, we implement the similar detection scheme to remove the outliers. We reject the visual measurements that are far from the predicted measurements. We also reject the observation update in the Kalman filter that changes the trajectory abruptly. While the current methods are straightforward, we believe that more sophisticated outlier removal methods can be proposed with more investigation.

7.2 BOEM-SLAM Versus Other Backend Algorithms in a Circular Trajectory Simulation

In this section, we compare the accuracy and computational cost of different SLAM backend algorithms. We focus on the explicit-map algorithms, which simultaneously estimate the trajectory and the landmarks. In particular, we consider three algorithms: the optimization-based SLAM (labeled opt., in Section II of [83]), the EM-SLAM and the proposed BOEM-SLAM.[1]

To ensure fair comparison of processing times, the simulations and the experiments are all performed on a single computer equipped with an Intel i5-8250U CPU @ 1.60 GHz and 8 GB of DDR4 RAM, and all optimization problems are solved by Ceres [93]. In addition all inputs to backends and optimization parameters are the same for each SLAM method.

7.2.1 Landmark and Trajectory Results

We test all three SLAM algorithms on a circular trajectory with discrete time interval 30 ms as in Fig. 7.1. The camera on the robot is constrained to face outward to observe landmarks on the walls throughout the trajectory. We then generate the IMU measurements and camera observations for the robot with random noise. All the SLAM results together with the dead reckoning trajectory are shown in Fig. 7.1. Although representative SLAM

---

[1] The source code and the parameters of the simulation and the experiment are available at `https://github.com/tsangkai/slam_demo`.

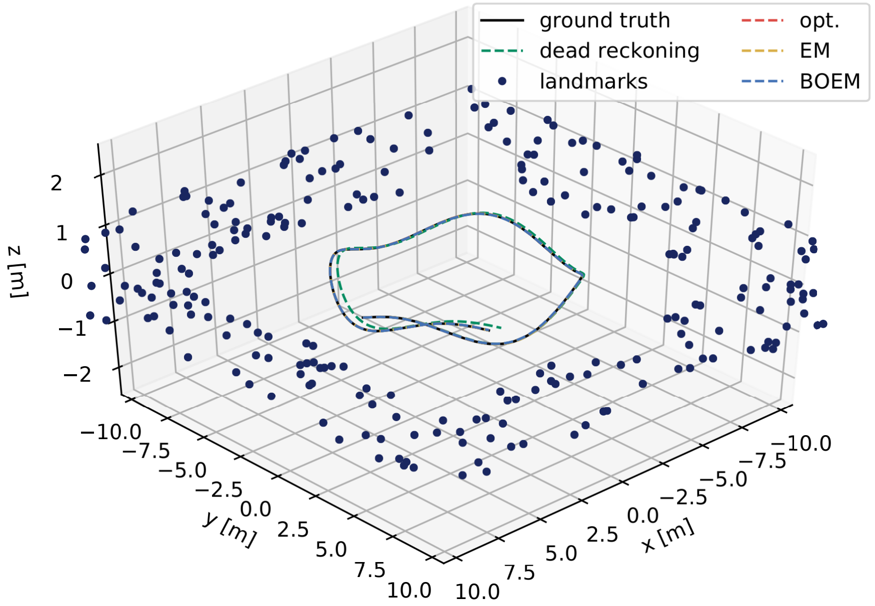


Figure 7.1: The first 16 sec of the simulated trajectories. Three SLAM algorithms are presented in dashed lines, including optimization-based algorithm (opt.), EM-SLAM and BOEM-SLAM. 200 landmarks are randomly generated on the walls of a 7.5 × 7.5 m box.

sessions are short, we expect similar outcomes over longer durations.

7.2.2 Rotation and Position Error Results

To verify the effectiveness of BOEM-SLAM, we plot the rotation and the position estimation errors of all SLAM algorithms over 50 sec in Fig. 7.2. All SLAM algorithms maintain reasonable estimation errors throughout this interval. The only noticeable difference between BOEM-SLAM and other algorithms is that the estimation trajectories from BOEM-SLAM have larger variance. Since BOEM-SLAM processes less data in each time block, we expect that it needs more data for convergence compared to other offline algorithms. Overall, the estimation accuracy of these 3 algorithms are comparable.

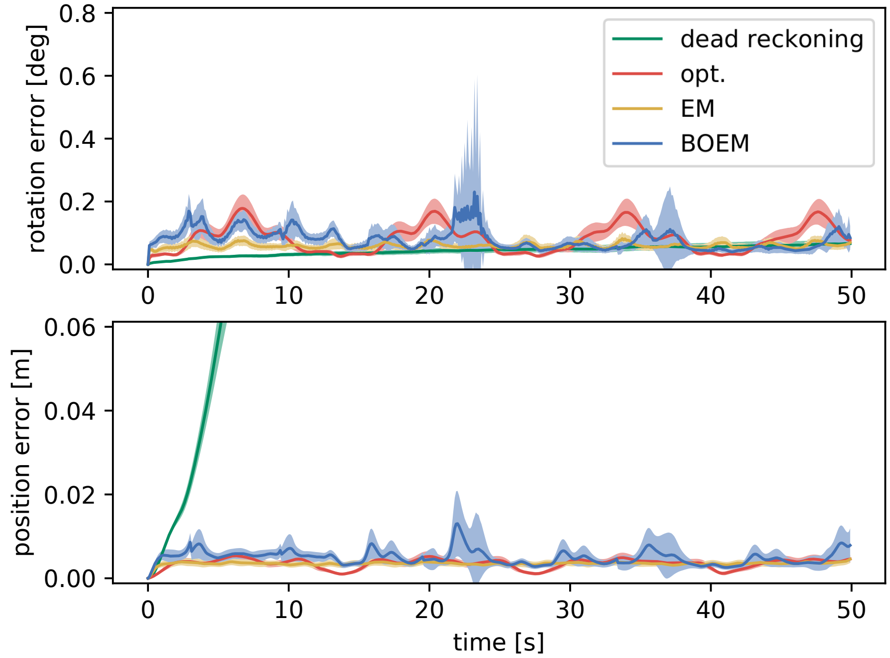


Figure 7.2: The rotation and the position errors of the SLAM algorithms. The shaded areas show 1 standard deviation error bar over 50 trials. All three SLAM algorithms show comparable accuracy in the simulation experiment.

7.2.3 Analysis of Processing Time Over Expanding SLAM Operating Windows

To emphasize the importance of the online feature for robotic applications, we consider the computation time with various durations. The duration of the trajectory is increased by 15 sec, and the performance analyzing tool `perf` is used to calculate the processing time, with the results shown in Fig. 7.3. Although all algorithms achieve similar estimation accuracy, the processing times are significantly different. The NLS optimization procedure dominates the computation cost[2] while an improvement of processing time can be achieved by decreasing the size of the optimization procedure, which makes EM-SLAM more efficient over the optimization-based SLAM [86]. Furthermore, as an online algorithm, BOEM-SLAM

---

[2] There are no realtime enhancements on any optimization procedure, it is likely that additional processing of this kind would greatly decrease computation time across algorithms.

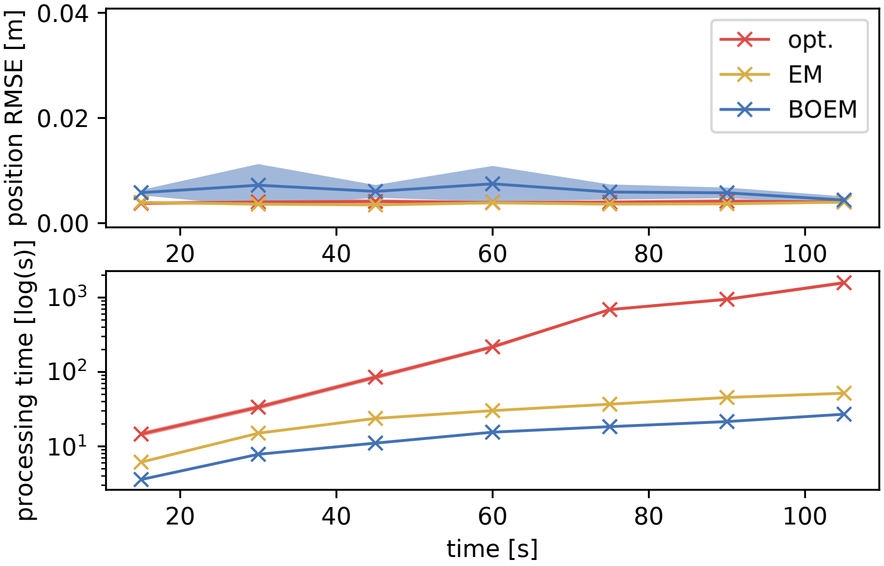


Figure 7.3: The trajectory estimation accuracy and the processing time with increasing time intervals. The shaded areas show 1 standard deviation error bar over 20 trials. As an online algorithm, BOEM-SLAM can effectively discard information, and thus it has a lower processing time.

can effectively summarize and discard past data, and thus has the least computational cost. For the duration of 150 sec, while optimization SLAM needs more than 1000 sec, BOEM-SLAM only takes around 20 sec to complete the task.

7.3 The BOEM-SLAM Method is Ideal for Use Cases Requiring Energy Efficiency

To validate the effectiveness of our proposed BOEM-SLAM approach, we compared its performance to other common SLAM backend algorithms using synthetic data. The simulation results demonstrate that BOEM-SLAM achieves comparable estimation accuracy while reducing computational cost. This reduction is due to BOEM-SLAM's ability to recursively update map and trajectory estimates using only the most recent data and a compact representation of past information. In this study we highlight the relationship between the increasing time intervals and the resulting computational cost. This realistic scenario considers the fact that data keep arriving in robotic applications, and therefore online SLAM algorithms are a preferable first step to better efficiency and scalability.

As an example use case, BOEM-SLAM is particularly relevant for multi-robot systems operating as wireless sensor networks [94]. In applications involving real-time data collection and transfer to a base station, energy efficiency is crucial due to a reliance on on-board batteries for travel, sensing, and communication. By reducing the computational complexity inherent in traditional SLAM algorithms, BOEM-SLAM minimizes the need for high-performance, power-consumptive processors. This reduced hardware demand not only lowers the initial cost of deploying robotic systems but also significantly decreases energy consumption during operation. This leads to longer operational times between recharges or battery replacements, which is crucial in remote or hard-to-access areas where maintaining power supply can be challenging.

CHAPTER 8

Future Work: Formulating CPG-HZD Control

8.1 Stability Analysis of CPG-HZD Control

In the analysis of energy balance in a gait, [95] noted the existence of a limit cycle in a system is associated with a contraction of phase (or state) space volume as time evolves. Dissipative elements in a system cause this kind of contraction which increases the likelihood (but does not guarantee) the existence of a limit cycle. The contraction mechanism in walking occurs with kinetic energy loss at impact. Thus, while in the cart-pole case of Ch. 4 there were no stable cycles, in CPG-HZD we can look for stable cycles a priori.

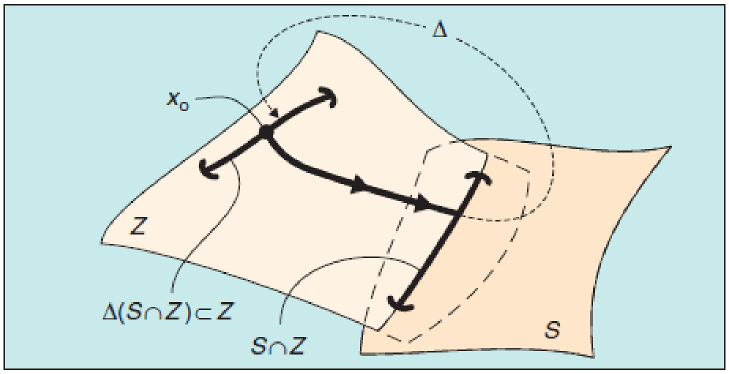


Figure 8.1: Invariance of the constraint surface under impact [5].

A complete description of the extension of ZD control to the hybrid case is given in [5]. Referring to Fig. 8.1, $\mathcal{Z}$ is the zero dynamic manifold, $\mathcal{S}$ is the impact surface, and Δ is the impact map. To be stable, we look for periodicity from step to step. The initial condition after impact must therefor be constant. The k holonomic constraints (on configuration) must be designed such that the new initial condition arising from the impact map is once again on $\mathcal{Z}$, i.e. $\Delta(\mathcal{S} \cap \mathcal{Z}) \subset \mathcal{Z}$. It will be convenient at this point to denote $\mathcal{W} \equiv \mathcal{Z} \cap \tilde{\mathcal{Z}}$. We shall then modify the hybrid invariance statement to say $\Delta(\mathcal{S} \cap \mathcal{W}) \subset \mathcal{W}$. This notation should be trivial in stability analysis, because invariance of the CPG state under impact is guaranteed.

8.2 Developing CPG-HZD Theory for Side-Walking

It is first necessary to search for periodic orbits of the hybrid zero dynamics of the CPG-HZD system. $\mathbb{W}$ is of dimension $2(N-k)$. For one degree of underactuation, the manifold is two dimensional. Its associated poincaré return map, ρ_{zero} is diffeomorphic to a scalar LTI system and may be computed in closed form with its convergence domain clearly defined[1]. The scalar representation of the poincarḿap in terms of energy makes the analysis intuitive and tractable.

Poincare analysis uses the momentum balance theorem,

$$\dot{\sigma} = Mgx_c \tag{8.1}$$

where x_c is the distance from the stance leg to the robot center of mass in the x-direction. Using this theorem, the swing phase zero dynamics can be written in the form,

$$\begin{aligned} \dot{\theta} &= \frac{1}{I(\theta)}\sigma_{\mathcal{W}} \\ \dot{\sigma}_{\mathcal{W}} &= Mgx_c(\theta) \end{aligned} \tag{8.2}$$

where $\sigma_{\mathcal{W}}$ is the angular momentum of robot restricted to $\mathcal{W}$ and $I(\theta)$ is the robot's inertia. The main result of computing ρ_{zero} can be presented in two steps: a) there exists a periodic solution to the CPG-HZD iff $\delta^2_{zero} \neq 1$ and

$$\frac{\delta^2_{zero}}{1-\delta^2_{zero}} V_{zero}(\theta^-) + V^{max}_{zero} > 0 \tag{8.3}$$

---

[1] If synergistic properties arise from HZD-CPG new stability analysis may be required.

where,

$$\begin{aligned} V_{zero}(\theta) &:= -\int_{\theta^+}^{\theta} I(\zeta) M g x_c(\zeta) d\zeta \\ V_{zero}^{max} &:= \max_{\theta} V_{zero}(\theta). \end{aligned} \tag{8.4}$$

b) there exists an exponentially stable periodic solution iff (8.3) and (8.4) hold and

$$0 \ < \ \delta_{zero}^2 \ < \ 1. \tag{8.5}$$

In (8.3) the direction of the inequality has been reversed from HZD-based control design. In the case of side-walking, this equation states that the energy release must be large enough such that the initial angular momentum is *insufficiently* large to overcome the potential energy barrier corresponding to the maximum potential energy, V_{zero}^{max}. The angular momentum changes via impact, $\sigma_{\mathcal{W}}^{+} = \delta_{zero}\sigma_{\mathcal{W}}^{-}$, where $\delta_{zero} < 1$ means there is a contraction in phase space. During swing phase energy is conserved, $K_{zero} + V_{zero} = C$, with $K_{zero} = 1/2\sigma_{\mathcal{W}}^2$ and the potential energy V_{zero} defined in (8.4). These values have units of energy if scaled by $I(\theta)$.

In forward walking, (8.3) indicates robustness. In forward walking, the equation stipulates that there must be sufficient angular momentum at the beginning of the step such that it is nonzero at V_{zero}^{max}. Forward gates simply need sufficient speed to satisfy this requirement, with error in modeling or disturbance altering walking speed before the robot is destabilized. This is not the case in side-walking, the angular momentum must be insufficient to overcome the barrier, but sufficient for forward motion. Intuitively this upper and lower bound on angular momentum may degrade the robustness of side-walking.

It is not clear whether using the CPG state, ϕ, or its phase value, φ, will be more advantageous for synergistic effects in CPG-HZD design. The design process using φ has fewer steps in the latter case, so this case will be studied first. The search for periodic walking motion can now be cast as a nonlinear optimization problem. The holonomic constraint on actuated states q_a,

$$h_{\gamma}(q_a, \varphi) := q_a - h_d(\varphi, \gamma), \tag{8.6}$$

will be constructed from a parametric curve such as a Bézier polynomial, with coefficients γ

introduced as the free parameter to generate a minimum-energy cost criterion:

$$J(\gamma) = \frac{1}{L_s}\int_0^T \|u_\gamma^*\|_2^2 dt. \tag{8.7}$$

With L_s the step length, T the walking period, and $u_\gamma^* \coloneqq -[\underbrace{L_g L_f h_\gamma}_{H}]^{-1} L_f^r h_\gamma$. The definition of u_γ^* is in terms of its Lie derivative to emphasize H must be nonsingular for the system to have a well defined relative degree. If H is everywhere invertible about the periodic orbit, then the input uniquely constrains the robot to the zero dynamics surface. The problem is subject to the following constraints,

1) Inequality constraints

- the swing leg is always positioned above the ground: $y_{stance} > 0$
- the ground reaction force does not permit rebound: $F^N > 0$
- no slip constraint: $\left|\frac{F^T}{F^N}\right| < \mu_f$
- entrainment constraint: $\phi - \theta < \epsilon$
- bounds on other kinematic parameters (hip height, etc.) to obtain desired walking style

2) Equality constraints

- average walking rate: $v = \frac{L_s}{T}$
- the existence of the fixed point, ζ^* (8.3)
- the stability of the fixed point, ζ^* (8.5)

In the optimization problem, once the holonomic constraints have been determined, the zero dynamics can be isolated as in Section 4.2.4. While in the cart-pole problem the dimension of the zero dynamics was equal to the dimension of the cart, the controller design is meant for higher DOF systems. The dimension of the zero dynamics manifold is $2(N-k)+2$, with N the number of DOFs and k the number of virtual constraints. If entrainment occurs, stability analysis occurs in the manifold $\mathcal{W}$ which is dim(2). This reduction is by the same principal as the dimension reduction provided by holonomic constraint. We hope that $\phi - \theta \approx 0$.

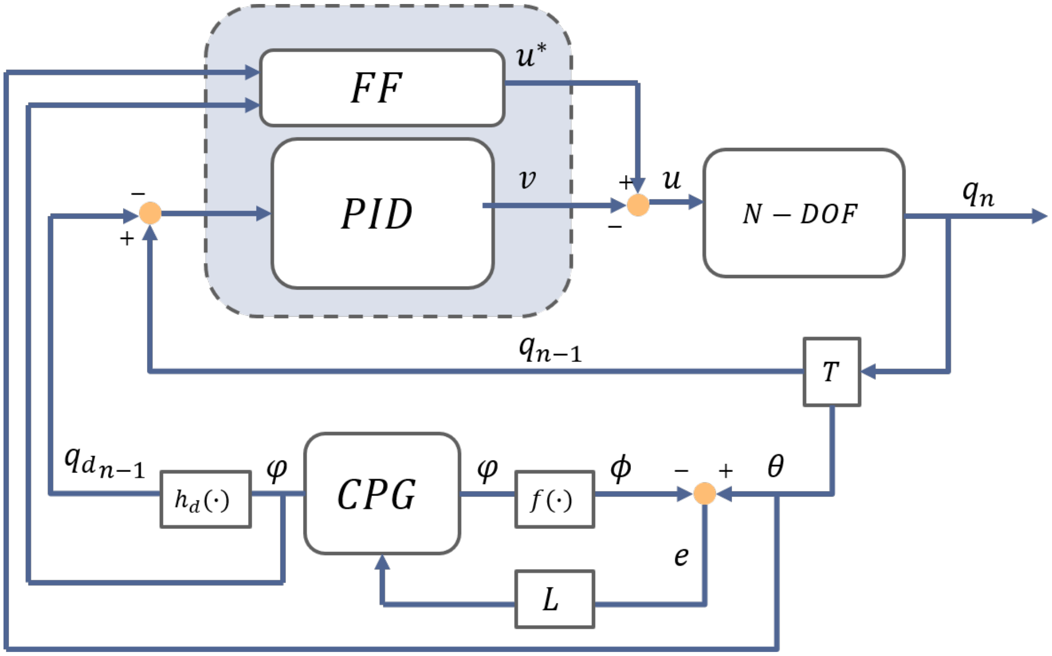


Figure 8.2: Block Diagram for CPG-HZD control.

The block diagram for the proposed form of control is given in Fig. 8.2. The blue block represents the control required to constrain the robot to the hybrid zero dynamic manifold. The T block is a transformation matrix to decouple the states, the L block tunes entrainment properties, and the function h_d represents the Beziér curve to map the phase to the holonomic constraint. It is unclear whether a Fourier series or Beziér curve would be best to map the φ state to ϕ.

8.3 Implementation of The CPG approach to HZD control

In this chapter, an approach to CPG-HZD control has been formulated. The outcome looks promising, because the design is simple, and stability features from each form of control should remain intact in combination. The controller will be tested in simulation first on the Compass-Gait walker (see Sec. A.4), then on a simulated version of NABi-V2. A thorough analysis of controller performance will need to be conducted to determine controlled invariant sets. Any auxiliary controllers will also be designed if needed. Formulating an approach to obtain the optimal performance is also necessary. For example, would it be easier to use a Matlab toolbox, or trajectory optimization software? Would the use of the Fourier series shaping function to include the CPG's waveform shape in addition to its phasing further encourage synergistic effects? Is there an alternative method to shaping the CPG signal ϕ during optimization, perhaps in some model-predictive way rather than by gradient decent? Once a simple, feasible, and optimal approach is created, it will be compiled with all provable theoretical analysis as content for the dissertation.

The CPG-HZD control relies heavily on contact information to switch control commands when the stance and swing leg switch, and to reset the CPG phase to maintain hybrid invariance. The NABi-V2 robot comes with proprioceptive motors suitable for detecting contact without requiring sensors on end effectors. [96] used contact model fusion within the framework of Kalman filtering to estimate impact events in locomotion to high accuracy. A generalized-momentum observer is used to estimate contact force, and this signal is blended with gait scheduling and kinematic information to estimate the likelihood of contact. I plan to implement a similar estimator to achieve impact event detection. In addition friction compensation may be necessary, if the models used for control design have frictionless joints.

8.4 CPG-HZD Control Has Synergistic Benefits

The HZD-method of control makes designing the CPG highly tractable. The NABi robot has $q \in \mathbb{R}^{10\times 1}$. This would require creating 10 CPG trajectories, and analyzing a dim(9) poincaré map. Moreover there is not a solid framework for CPG design for hybrid systems, because these systems often require active balancing control. CPGs have historically been applied to joint-space control, which is not intuitive for walking. The HZD-method eliminates these issues completely. The HZD reduces the CPG design to dim(2), with a poincaré map of dim(1)[2], and it becomes a problem of *analysis* not *synthesis*. The HZD method synthesizes the controller because limb coordination is provided by holonomic constraint. The only design effort to create the CPG trajectories is ensuring they track the robot parameter of interest, $\Theta = [\ \theta\ \dot{\theta}\ ]^T$. Combining controllers can also create synergistic properties for added stability—the CPG may increase stability regions in the zero dynamic manifold because the orbit entrained to θ is exponentially stable.

---

[2] if entrainment occurs, $\phi \approx \theta$.

CHAPTER 9

Conclusion

This thesis has explored various aspects of robotic intelligence, focusing on the design considerations and innovative solutions that enable efficient, stable, and adaptable robot performance. Through the analysis of specific examples, we have highlighted the challenges and successes in each case, providing valuable insights into the future of robotic intelligence. The development of the compliant, Non-Anthropomorphic Biped NABi-S has shown the viability of open-loop CPG control for stable locomotion. The compliant mechanism designed for the ankle has proven effective in maintaining robot stability, and the ease of implementation of the CPG using the MHB method has been demonstrated. Future work will focus on closing the loop on the CPG control of NABi-S to further enhance the robot's performance.

Steps have already been taken in designing a CPG controller via a CPG-ZD control method where the closed-loop form of control was able to stabilize both marginally stable and unstable zero dynamics of a cart-pole system. In addition, a closed-loop hybrid extension of the CPG controller was designed that was able to increase the stability regions of the periodic walking gate of the Simple Walker model. Hybridization of CPG and HZD control was also discussed, where synergistic properties can be expected as an added feature. For example, the CPG's limit cycle stability may improve the function of the HZD, expanding stability regions around stable zero dynamics, potentially increasing robustness to disturbances. In turn, the ease of design and stability analysis given by the HZD method open a door for using a CPG on a system requiring complex limb coordination to actively balance.

In the context of MIMO network enhancement for data aggregation tasks, we have characterized the capacity within a workspace and identified trends impacting communications throughput for path following and coverage tasks. By identifying regions where ULA heuristics for static array design can improve capacity in a mobile sensing task, we have demonstrated the potential for optimized communication in complex sensing tasks. Future work will involve extending the current ULA approximation to account for the z-direction of travel, which would extend the algorithm to a more general set of flight plans while still achieving optimized communication.

The proposed BOEM-SLAM backend for visual-inertial SLAM systems has shown significant improvements in computational efficiency without sacrificing solution optimality. By downsizing data requirements and eliminating the need for the entire history of data, we have shown the potential for BOEM-SLAM to enhance SLAM-assisted robotic autonomy. Furthermore, while BOEM-SLAM and MIMO communication techniques do not directly integrate, they can work in tandem to improve the overall efficiency of the system. BOEM-SLAM provides accurate and computationally efficient pose estimation, enabling robots to maintain a reliable understanding of their position and orientation in an environment. This, in turn, allows for better coordination and path planning among the robots, which can be leveraged to optimize the placement of the robots for improved MIMO communication performance, as demonstrated in this thesis. By combining the benefits of BOEM-SLAM and MIMO communication techniques, multi-robot systems can achieve enhanced localization accuracy, efficient data aggregation, and faster data transmission, ultimately leading to improved performance in real-world applications.

In conclusion, the examples presented in this thesis demonstrate the importance of designing robots with intelligence in mind. By addressing challenges such as three-dimensional control and balancing, efficient data aggregation and communication, and computationally efficient SLAM, we can create robots that are more adaptable, stable, and capable of performing complex tasks. As the field of robotic intelligence continues to evolve, it is essential to build upon these successes and explore new innovative solutions to push the boundaries of what robots can achieve.

APPENDIX A

Supplementary Information

A.1 Related Hardware Designs

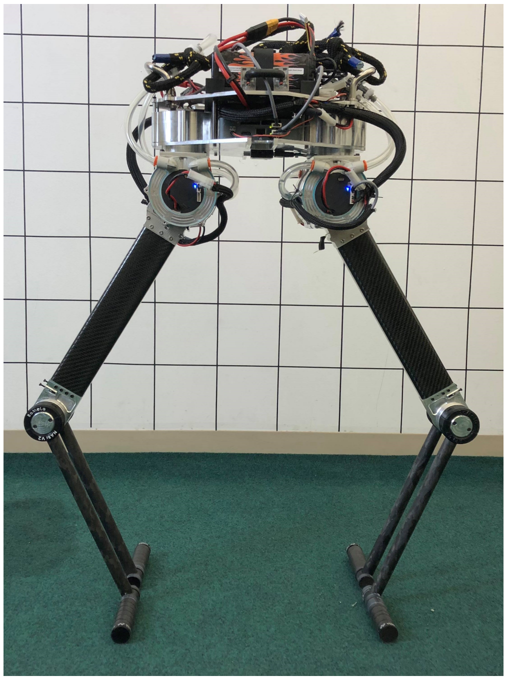


Figure A.1: NABi-V2 version robot added yaw DOFs at the hip joint [6].

A.2 Bipedal Walking Assumptions

The research of this thesis focuses on control methods for planar bipeds with point feet. The point foot, where the robot makes contact with the ground, represents one degree of underactuation in the system, i.e. for a robot comprised of N links, $N-1$ of these links are actuated. The robot is assumed to have frictionless joints, connected by rigid links. When walking the robot alternates between a phase of single support and double support. In single support, the supporting leg or *stance leg* acts as a pivot. Like the ground reaction force of a floating body, the vertical component of the pivot's reaction force must be pointing upwards, and the ratio of horizontal component to vertical component must not exceed the coefficient of static friction. At the end of a step the moving leg or *swing leg* makes an inelastic impact with the ground. During impact, the dynamic model transitions from an open kinematic chain to a floating body to allow a discontinuous jump in velocity. The impact represents an infinitesimal period of double support, followed once again by single support. The dynamical model is *hybrid* due to the discontinuity in the velocity state at impact. A separate set of dynamical equations are used for the single support and double support phases.

A.3 Walking Gait Hypothesis for Side-Walking

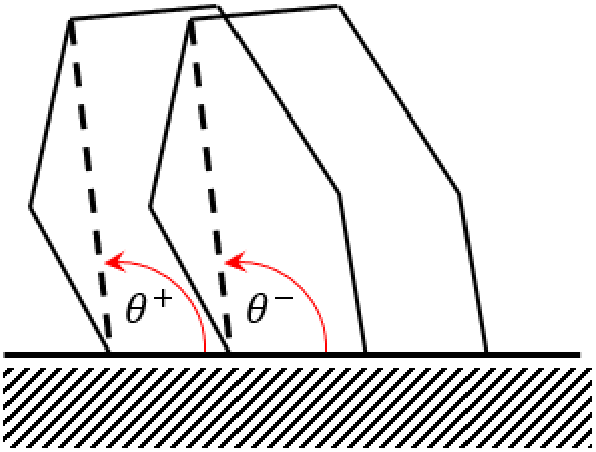


Figure A.2: Step definition for side-walking with '+' and '-' labels for the stance foot at the beginning and end of a step.

The CPG-HZD controller will render gait behavior with the same properties as [27], with the exception of - *HGW5) in steady state, the motion is symmetric with respect to the two* *legs; HGW6) in each step, the swing leg starts from strictly behind the stance leg and is placed strictly in front of the stance leg at impact.* The preceding gait hypotheses are related by the definition of a step. In forward walking, the robot makes net forward movement by the process of HGW6. In the case of side-walking, there are two cycles of alternating support before the robot makes net forward movement (Fig. A.2). In this case it may be over-constraining to impose HGW5. Thus we make the following modifications - $\overline{HGW5}$*) in steady state, the motion is not necessarily symmetric with respect to the two legs;* $\overline{HGW6}$*) in each step, there are two phases of alternating support associated with the net forward movement of both legs to create net forward movement of the body.*

A.4 Walker Dynamics for CPG-HZD Controller

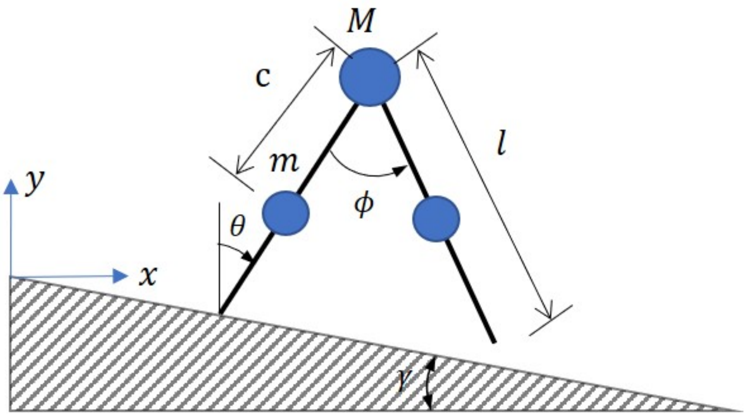


Figure A.3: Schematic for the compass-gait walker.

The compass-gait walker is the dynamical model that will be used to test the CPG-HZD controller. A schematic of the walker is shown in Fig. A.3. The walker is underactuated, with a zero torque connection between the stance leg and the ground. All joints are assumed frictionless. The walker has a point mass at the hip, labeled M, and a point mass on each leg, labeled m. The point masses on the legs are a distance c from the hip, and the legs are l in length. Given coordinates shown in Fig. A.3, the robot walks from left to right, in the positive x direction. The z axis for the inertial and all relative frames points outward. Thus the robot as it stands in Fig. A.3 has a negative θ value and a positive ϕ value.

A.4.1 Swing phase dynamics

In the single support phase, the point foot of the walker is kinematically constrained to the ground, forming an open kinematic chain. The walker has two generalized coordinates, $q = [\theta, \phi]^T$. The equations of motion are found by applying the method of Lagrange. The model takes the form,

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = Bu. \tag{A.1}$$

The equations of motion are:

$$\begin{aligned}
M(q_2)_{1,1} &= Ml^2 + 2m(c^2 + l^2) - 2clm(1 + cos(q_2)) \\
M(q_2)_{1,2} &= cm(c - lcos(q_2)) \\
M(q_2)_{2,1} &= cm(c - lcos(q_2)) \\
M_{2,2} &= c^2 m \\
\\
C(q_2,\dot{q_2})_{1,1} &= 2clm(sin(q_2))\dot{q_2} \\
C(q_2,\dot{q_2})_{1,2} &= clm(sin(q_2))\dot{q_2} \\
C(q_2,\dot{q_1})_{2,1} &= -clmsin(q_2)\dot{\theta} \\
C_{2,2} &= 0 \\
\\
G(q_1,q_2)_1 &= cgm(sin(-\gamma + q_1 + q_2)) + (Mgl - cgm + 2glm)(sin(\gamma - q_1)) \\
G(q_1,q_2)_2 &= cgm(sin(-\gamma + q_1 + q_2)) \\
\\
B_1 &= 0 \\
B_2 &= 1.
\end{aligned} \tag{A.2}$$

It is important to monitor the ground reaction force (GRF) during the swing phase.

Because the point foot is effectively pinned to the ground, the GRF is internal to the system[1], it does not explicitly appear in the Euler-Lagrange equation. In order to get the GRF to appear explicitly, 2 states x, y were added to the walker. These states represent the position of the standing foot. The Lagrange multiplier method was then utilized, with the constraint that the position of the standing foot should remain constant: $h_{st}(x, y) \in \mathbb{R}^{2\times 1} = constant$. For the GRF $\lambda_{st} \in \mathbb{R}^{2\times 1}$, the equation is,

$$\begin{aligned} \frac{d}{dt}\frac{\partial L(q)}{\partial \dot{x}} - \frac{\partial L(q)}{\partial x} &= \sum_{i=1}^{2} \lambda_{st_i} \frac{\partial h_{st_i}(x, y)}{\partial x} \\ \frac{d}{dt}\frac{\partial L(q)}{\partial \dot{y}} - \frac{\partial L(q)}{\partial y} &= \sum_{i=1}^{2} \lambda_{st_i} \frac{\partial h_{st_i}(x, y)}{\partial y}. \end{aligned} \tag{A.3}$$

The constraint on the standing foot is enforced by λ_{st}. The force on the right hand side of (A.3) appears by virtual work. The four equations of motion resulting from adding a third and fourth state were a function of $\ddot{x}, \ddot{y}, \dot{x}, \dot{y}$ and x, y. Setting $\ddot{x}, \ddot{y}, \dot{x}, \dot{y} = 0$ in the two equations associated with θ, ϕ, the equations of motion given by (A.2) emerge, and the system of equations, (A.3), may be solved for λ_{st}. λ_{st_1} and λ_{st_2} represent the friction and normal forces, respectively. For brevity the equations are not given here, see Appx. A. Note: the figure is a little off, that line theta is measured from needs to look normal to the incline.

A.4.2 Impact dynamics

At impact the GRF becomes an *external* force. To be external to the robot, the system becomes a floating body. The floating body may translate in addition to changing shape and orientation. Thus the generalized coordinates become $q = [x, y, \theta, \phi]^T$, with x, y describing the location of the stance foot. Impact is inelastic, and instantaneous- when the swing foot makes contact with the ground, the stance foot lifts from the ground. The external GRF is impulsive, causing a discontinuous jump in joint velocities, and thus a new initial condition. Due to the exchange of stance and swing leg roles, coordinates are relabeled after the phase

---

[1] In this case an internal force is synonymous with a conservative force, it does no work. The force from gravity is technically considered *external*, it affects a system's momentum.

of double support as well.

Utilizing the Lagrange multiplier method in the case of impact, the forces on the system due to the GRF on the swing foot, $\lambda_{sw} \in \mathbb{R}^{2\times 1}$, has the constraint $h_{sw}(q) \in \mathbb{R}^{2\times 1} = constant$, where $h_{sw}(q)$ represents the position of the swing foot. The Euler-Lagrange equation for the floating body is

$$\frac{d}{dt}\frac{\partial L(q)}{\partial \dot{q}_i} - \frac{\partial L(q)}{\partial q_i} = \sum_{i=1}^{m} Q_i + \sum_{i=1}^{2} \lambda_{sw_i} \frac{\partial h_{sw_i}(q)}{\partial q_i}. \tag{A.4}$$

where Q[2] represents m other external forces on the robot.

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = Bu + J(q)\lambda_{st} + W(q)\lambda_{sw}. \tag{A.5}$$

When the system of dynamical equations is created, the matrix[3] $W \in \mathbb{R}^{n\times 2}$ represents the mapping of the swing GRF forces to the joints of the robot. The velocity of the swing foot, $\frac{dh}{dt}$ can be otherwise expressed as $\frac{\partial h(q)}{\partial q}\frac{dq}{dt}$ or $W^T\dot{q}$. If the impact is perfectly inelastic, then the coefficient of restitution $e = 0$. With the velocity of the ground labeled as v_e and using Newton's law of impact:

$$\begin{aligned} v_e^+ - W^T\dot{q}^+ &= e(W^T\dot{q}^- - v_e^-) \\ W^T\dot{q}^+ &= -eW^T\dot{q}^- \\ W^T\dot{q}^+ &= 0. \end{aligned} \tag{A.6}$$

In (A.6) q^-, q^+ refers to the state values before and after impact, respectively. An inelastic collision indicates that the swing foot hits the ground without rebound or slip, so its velocity at impact is zero. The change in momentum of the robot is largely due to the impulsive

---

[2] In the derivation it is assumed that the stance foot GRFs are grouped in with Q.

[3] In some cases the $f(q)$ will be dropped to make derivations simpler to read.

force, f_{imp},[4]

$$\begin{aligned} M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) &= Bu + J(q)\lambda_{st} + W(q)\lambda_{sw} \\ M(q)\frac{d\dot{q}}{dt} &= W(q)\lambda_{sw} \\ \int_{\dot{q}^-}^{\dot{q}^+} M(q)d\dot{q} &= W(q)\underbrace{\int_{t^-}^{t^+} \lambda_{sw}dt}_{f_{imp}} \\ M(\dot{q}^+ - \dot{q}^-) &= Wf_{imp} \\ W^T\dot{q}^+ - W^T\dot{q}^- &= W^TM^{-1}Wf_{imp}. \end{aligned} \tag{A.7}$$

From (A.7) we see that

$$\begin{aligned} f_{imp} &= -(W^TM^{-1}W)^{-1}W^T\dot{q}^- \\ \dot{q}^+ &= \dot{q}^- + M^{-1}Wf_{imp}. \end{aligned} \tag{A.8}$$

In (A.8), we know that the stance foot is stationary, thus the first two states of $\dot{q}^-$, $\dot{x}$ and $\dot{y}$ are equal to zero. After impact, $\dot{q}^+$ is such that the velocity of the swing foot is equal to zero.

After impact the swing foot is now the stance foot, and the coordinates must be relabeled. The states are relabeled using a linear transformation matrix, R, a circular matrix ($RR = I$). For the robot states shown in Fig. A.3,

$$R = \begin{bmatrix} 1 & 1 \\ 0 & -1 \end{bmatrix}. \tag{A.9}$$

The configuration values of the robot don't change after impact, $q^+ = q^-$. For q_O, q_N designating old and new coordinates respectively, we have,

$$\begin{aligned} q_N &= Rq_O \\ \dot{q}_N^+ &= R(\dot{q}_O^- + M^{-1}Wf_{imp}). \end{aligned} \tag{A.10}$$

---

[4] it is important to note that the impulsive force is *internal* to the robot-earth system, the momentum of this system is conserved in the collision.

A.4.2.1 Compact notation

Putting (A.6) and (A.8) together, a compact form of the impact equations are,

$$\begin{bmatrix} \dot{q}_N^+ \\ f_{imp} \end{bmatrix} = \begin{bmatrix} R & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} M & -W \\ W^T & 0 \end{bmatrix}^{-1} \begin{bmatrix} M \\ -eW^T \end{bmatrix} \dot{q}_O^- \tag{A.11}$$

A.4.2.2 Conditions for impact

The conditions for impact are 1) the sign of the states q *do not* change, and 2) $q_2 = -2q_1$. This condition may be relaxed as 1) when the height of the swing foot is equal to zero, and 2) $\dot{q}_2 > 0$. It is not immediately apparent whether either of these sets of conditions will result in adequate side-walking. Using the relationship between q_1 and q_2, another method for determining the first equation of (A.10) can be found for $q_1 \coloneqq \theta$ and $q_2 \coloneqq \phi$ because the values of the position states are the same before and after impact:

$$\begin{aligned} \theta_N &= \theta_O + \phi_O \\ \phi_O &= -2\theta_O \\ \theta_N &= -\theta_O, \end{aligned} \tag{A.12}$$

the same is true for the ϕ state:

$$\begin{aligned} \phi_N &= -\phi_O \\ \phi_O &= -2\theta_O \\ \phi_N &= 2\theta_O. \end{aligned} \tag{A.13}$$

A.5 Side-Walking Gait modeling

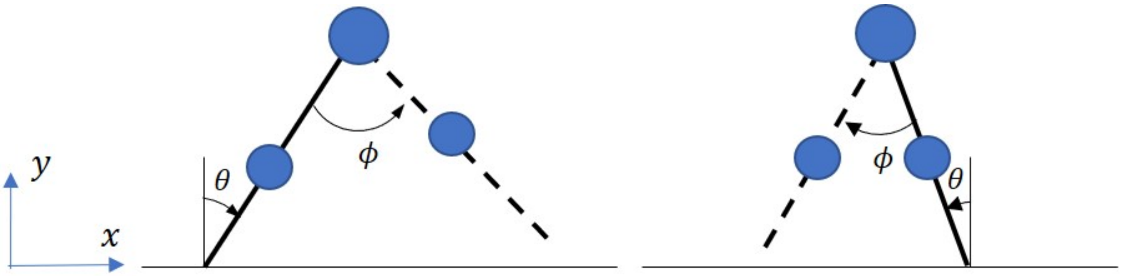


Figure A.4: Schematic showing one side-walking step.

Referring to Fig. A.4, the dashed line is the swing leg. Each leg has the role of stance/swing in each step. The following changes are made to facilitate side-walking-

1. In steady state, the motion of the robot is not necessarily symmetric with respect to the two legs.
2. In each step, there are two phases of alternating support to achieve net forward movement of the body.
3. In its initial position the robot has its swing leg in front.
4. Throughout a step, the sign of θ, ϕ does not change.
5. At impact, $\phi = -2\theta$.

A.6 Verifying Model dynamics

Using [97] to compare the Compass-Gait Walker's dynamical model-

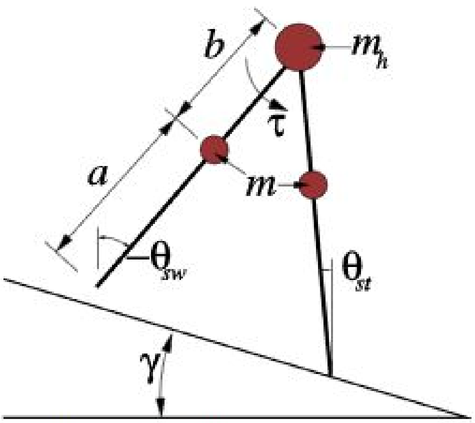


Figure 4.7 - The compass gait

Figure A.5: Schematic for the compass-gait walker.

Tedrake et al. uses states $\tilde{q} = [\theta_{st},\ \theta_{sw}]$, referring to Fig. A.3, the states used in this model are $q = [\theta,\ \phi]$. To check that the auto-derivation program for the dynamics works, a comparison is made between this work and [97]. Tedrake et al. use the subscripts "sw" and "st" to mean swing state and stance state, with reference to the inertial frame, respectively (Fig. A.5). A linear relationship exists between the two models:

$$\begin{bmatrix} \theta_{st} \\ \theta_{sw} \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}}_{T} \begin{bmatrix} \theta \\ \phi \end{bmatrix} \tag{A.14}$$

Thus to compare the equations, the following change of coordinates is made to Tedrake's model (subscript T indicates Tedrake model, γ set to 0):

$$T'M(q)_T T\ddot{q} + T'C(q,\dot{q})_T T\dot{q} + T'G(q)_T = T'B_T u \tag{A.15}$$

The following results are achieved, the output for my auto-derivation code, (A.2) is-

$M(q) =$

```
[ M*l^2 + 2*c^2*m + 2*l^2*m - 2*c*l*m - 2*c*l*m*cos(phi), m*c^2
 - l*m*cos(phi)*c]
[m*c^2 - l*m*cos(phi)*c, c^2*m]
```

$C(q,\dot{q})\dot{q} =$

```
[c*l*m*sin(phi)*dphi^2 + 2*c*dtheta*l*m*sin(phi)*dphi]
[-c*dtheta^2*l*m*sin(phi)]
```

$G(q) =$

```
[c*g*m*sin(theta) - M*g*l*sin(theta)- 2*g*l*m*sin(theta) +
 c*g*m*cos(phi)*sin(theta) + c*g*m*cos(theta)*sin(phi)]
[c*g*m*cos(phi)*sin(theta) + c*g*m*cos(theta)*sin(phi)]
```

$B =$

```
   [0 1]'
```

The output from (A.15) is-

$T'M(q)_T T =$

```
[ M*l^2 + 2*c^2*m + 2*l^2*m - 2*c*l*m - 2*c*l*m*cos(phi), m*c^2
 - l*m*cos(phi)*c]
[m*c^2 - l*m*cos(phi)*c, c^2*m]
```

$T'C(q,\dot{q})_T T\dot{q} =$

```
[c*l*m*sin(phi)*dphi^2 + 2*c*dtheta*l*m*sin(phi)*dphi]
[-c*dtheta^2*l*m*sin(phi)]
```

$T'G(q)_T =$

```
[c*g*m*sin(theta) - M*g*l*sin(theta)- 2*g*l*m*sin(theta) +
 c*g*m*cos(phi)*sin(theta) + c*g*m*cos(theta)*sin(phi)]
[c*g*m*cos(phi)*sin(theta) + c*g*m*cos(theta)*sin(phi)]
```

$T'B_T =$

```
   [0 1]'
```

References

[1] J. Yu, J. Hooks, S. Ghassemi, and A. Pogue, "Investigation of a non-anthropomorphic bipedal robot with stability, agility, and simplicity," in *International Conference on Ubiquitous Robots and Ambient Intelligence*, 2016.

[2] PrimeCare Physiotherapy, "Soleus muscle pain," https://primecarephysio.ca/soleus-muscle-pain/, 2022, accessed: 2024-06-08.

[3] R. Tedrake, "Underactuated robotics: Algorithms for walking, running, swimming, flying, and manipulation," https://underactuated.mit.edu/index.html, 2024, accessed: 2024-06-13.

[4] M. Garcia, A. Chatterjee, A. Ruina, and M. Coleman, "The simplest walking model: Stability, complexity, and scaling," *Journal of Biomechanical Engineering*, vol. 120, no. 2, pp. 281–288, April 1998.

[5] C. Chevallereau, G. Abba, Y. Aoustin, F. Plestan, E. R. Westervelt, C. Canudas-De-Wit, and J. W. Grizzle, "Rabbit: a testbed for advanced control theory," *IEEE Control Systems Magazine*, vol. 23, no. 5, pp. 57–79, Oct 2003.

[6] J. Yu, J. Hooks, X. Zhang, M. Sung Ahn, and D. Hong, "A proprioceptive, force-controlled, non-anthropomorphic biped for dynamic locomotion," in *2018 IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids)*, 2018, pp. 1–9.

[7] M. H. Raibert, *Legged Robots That Balance.* Cambridge, MA, USA: Massachusetts Institute of Technology, 1986.

[8] J. Buchli, M. Kalakrishnan, M. Mistry, P. Pastor, and S. Schaal, "Compliant quadruped locomotion over rough terrain," in *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on.* St. Louis, MO: IEEE, 2009, pp. 814–820.

[9] G. A. Pratt and M. M. Williamson, "Series elastic actuators," in *Intelligent Robots and Systems 95. 'Human Robot Interaction and Cooperative Robots', Proceedings. 1995 IEEE/RSJ International Conference on*, vol. 1. Pittsburgh, PA: IEEE, 1995, pp. 399–406.

[10] McMaster-Carr, "Wear-resistant 1095 spring steel sheets and strips," https://www.mcmaster.com/products/spring-steel/wear-resistant-1095-spring-steel-sheets-and-strips/, accessed: 2024-06-08.

[11] Kenhub, "Soleus muscle," https://www.kenhub.com/en/library/anatomy/soleus-muscle, accessed: 2024-06-08.

[12] ROBOTIS, "Dynamixel mx-106t," https://www.robotis.us/dynamixel-mx-106t/, accessed: 2024-06-08.

[13] J. J. Craig, *Introduction to robotics: mechanics and control.* Pearson Prentice Hall Upper Saddle River, 2005, vol. 3.

[14] J. J. Uicker, G. R. Pennock, J. E. Shigley *et al.*, *Theory of machines and mechanisms.* Oxford University Press New York, 2011, vol. 1.

[15] M. MacKay-Lyons, "Central pattern generation of locomotion: a review of the evidence," *Physical therapy*, vol. 82, no. 1, pp. 69–83, 2002.

[16] G. Endo, J. Morimoto, J. Nakanishi, and G. Cheng, "An empirical exploration of a neural oscillator for biped locomotion control," in *Proc. IEEE Int. Conf. Robotics and Automation ICRA '04*, vol. 3, Apr. 2004, pp. 3036–3042 Vol.3.

[17] A. Crespi, K. Karakasiliotis, A. Guignard, and A. J. Ijspeert, "Salamandra robotica ii: An amphibious robot to study salamander-like swimming and walking gaits," *IEEE Transactions on Robotics*, vol. 29, no. 2, pp. 308–320, Apr. 2013.

[18] Y. Fukuoka, H. Kimura, Y. Hada, and K. Takase, "Adaptive dynamic walking of a quadruped robot 'tekken' on irregular terrain using a neural system model," in *Proc. IEEE Int. Conf. Robotics and Automation (Cat. No.03CH37422)*, vol. 2, Sep. 2003, pp. 2037–2042 vol.2.

[19] J. Lee, D. J. Hyun, J. Ahn, S. Kim, and N. Hogan, "On the dynamics of a quadruped robot model with impedance control: Self-stabilizing high speed trot-running and period-doubling bifurcations," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sept 2014, pp. 4907–4913.

[20] T. Iwasaki, "Multivariable harmonic balance for central pattern generators," *Automatica*, vol. 44, no. 12, pp. 3061–3069, 2008.

[21] S. Kajita, H. Hirukawa, K. Harada, and K. Yokoi, *Introduction to humanoid robotics.* Springer, 2014, vol. 101.

[22] J. Morimoto, J. Nakanishi, G. Endo, G. Cheng, C. G. Atkeson, and G. Zeglin, "Poincare-map-based reinforcement learning for biped walking," in *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on.* IEEE, 2005, pp. 2381–2386.

[23] T. Iwasaki and M. Zheng, "Sensory feedback mechanism underlying entrainment of central pattern generator to mechanical resonance," *Biological cybernetics*, vol. 94, no. 4, pp. 245–261, 2006.

[24] D. Hobbelen, T. de Boer, and M. Wisse, "System overview of bipedal robots flame and tulip: Tailor-made for limit cycle walking," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, Sep. 2008, pp. 2486–2491.

[25] A. Mohammadi, M. Maggiore, and L. Consolini, "Dynamic virtual holonomic constraints for stabilization of closed orbits in underactuated mechanical systems," *Automatica*, vol. 94, pp. 112 – 124, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0005109818302085

[26] A. Isidori, "The zero dynamics of a nonlinear system: From the origin to the latest progresses of a long successful story," *European Journal of Control*, vol. 19, no. 5, pp. 369 – 378, 2013, the Path of Control. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0947358013000836

[27] E. Westervelt, J. Grizzle, C. Christine, J. Choi, and B. Morris, *Feedback Control of Dynamic Bipedal Robot Locomotion*, 01 2007.

[28] K. Ren and T. Iwasaki, "Design of complex oscillator network with multiple limit cycles," 12 2018, pp. 115–120.

[29] S. Rutishauser, A. Sprowitz, L. Righetti, and A. J. Ijspeert, "Passive compliant quadruped robot using central pattern generators for locomotion control," in *Biomedical Robotics and Biomechatronics, 2008. BioRob 2008. 2nd IEEE RAS & EMBS International Conference on.* IEEE, 2008, pp. 710–715.

[30] T. Yamasaki, T. Nomura, and S. Sato, "Possible functional roles of phase resetting during walking," *Biological Cybernetics*, vol. 88, no. 6, pp. 468–496, Jun 2003.

[31] J. Nakanishi, J. Morimoto, G. Endo, G. Cheng, S. Schaal, and M. Kawato, "Learning from demonstration and adaptation of biped locomotion with dynamical movement primitives," *Robotics and Autonomous Systems - RaS*, vol. 2004, 01 2003.

[32] C. Chevallereau, J. W. Grizzle, and C. Shih, "Asymptotically stable walking of a five-link underactuated 3-d bipedal robot," *IEEE Transactions on Robotics*, vol. 25, no. 1, pp. 37–50, Feb 2009.

[33] K. Sreenath, H.-W. Park, I. Poulakakis, and J. Grizzle, "Embedding active force control within the compliant hybrid zero dynamics to achieve stable, fast running on mabel," *The International Journal of Robotics Research*, vol. 32, no. 3, pp. 324–345, 2013.

[34] A. Hereid, C. M. Hubicki, E. A. Cousineau, and A. D. Ames, "Dynamic humanoid locomotion: A scalable formulation for hzd gait optimization," *IEEE Transactions on Robotics*, vol. 34, no. 2, pp. 370–387, April 2018.

[35] Y. Gong, R. Hartley, X. Da, A. Hereid, O. Harib, J.-K. Huang, and J. Grizzle, "Feedback control of a cassie bipedal robot: Walking, standing, and riding a segway," 09 2018.

[36] M. A. Hsieh, A. Cowley, J. F. Keller, L. Chaimowicz, B. Grocholsky, V. Kumar, C. J. Taylor, Y. Endo, R. C. Arkin, B. Jung, D. F. Wolf, G. S. Sukhatme, and D. C. MacKenzie, "Adaptive teams of autonomous aerial and ground robots for situational awareness: Field reports," *J. Field Robot.*, vol. 24, no. 11-12, pp. 991–1014, Nov. 2007.

[37] Y. Kantaros and M. M. Zavlanos, “Distributed Intermittent Connectivity Control of Mobile Robot Networks,” *IEEE Transactions on Automatic Control*, vol. 62, no. 7, pp. 3109–3121, Jul. 2017.

[38] A. Muralidharan and Y. Mostofi, “Path Planning for Minimizing the Expected Cost until Success,” *arXiv:1802.00898 [cs]*, Aug. 2018, arXiv: 1802.00898.

[39] U. Ali, H. Cai, Y. Mostofi, and Y. Wardi, “Motion-Communication Co-Optimization With Cooperative Load Transfer in Mobile Robotics: An Optimal Control Perspective,” *IEEE Transactions on Control of Network Systems*, vol. 6, no. 2, pp. 621–632, Jun. 2019.

[40] Yuan Yan and Y. Mostofi, “Communication and path planning strategies of a robotic coverage operation,” in *2013 American Control Conference*. Washington, DC: IEEE, Jun. 2013, pp. 860–866.

[41] A. Muralidharan and Y. Mostofi, “Path Planning for a Connectivity Seeking Robot,” in *2017 IEEE Globecom Workshops (GC Wkshps)*. Singapore: IEEE, Dec. 2017, pp. 1–6.

[42] A. Muralidharan and Y. Mostofi, “Distributed beamforming using mobile robots,” in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, March 2016, pp. 6385–6389.

[43] ——, “Energy optimal distributed beamforming using unmanned vehicles,” vol. 5, no. 4, 2018, pp. 1529–1540.

[44] A. Ghaffarkhah and Y. Mostofi, “Path planning for networked robotic surveillance,” *IEEE Transactions on Signal Processing*, vol. 60, no. 7, pp. 3560–3575, July 2012.

[45] S. Hayat, E. Yanmaz, T. X. Brown, and C. Bettstetter, “Multi-objective UAV path planning for search and rescue,” in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, May 2017, pp. 5569–5574, iSSN: null.

[46] A. Goldsmith, *Wireless Communications*. USA: Cambridge University Press, 2005.

[47] A. J. Paulraj, D. A. Gore, R. U. Nabar, and H. Bolcskei, “An overview of MIMO communications - a key to gigabit wireless,” *Proceedings of the IEEE*, vol. 92, no. 2, pp. 198–218, Feb 2004.

[48] D. Tse and P. Viswanath, *Fundamentals of wireless communication*. Cambridge university press, 2005.

[49] T. Haustein and U. Kruger, “Smart geometrical antenna design exploiting the LOS component to enhance a MIMO system based on rayleigh-fading in indoor scenarios,” in *14th IEEE Proceedings on Personal, Indoor and Mobile Radio Communications, 2003. PIMRC 2003.*, vol. 2. Beijing, China: IEEE, 2003, pp. 1144–1148.

[50] D. Gesbert, H. Bolcskei, D. Gore, and A. Paulraj, “Outdoor MIMO wireless channels: models and performance prediction,” *IEEE Transactions on Communications*, vol. 50, no. 12, pp. 1926–1934, Dec. 2002.

[51] F. Bohagen, P. Orten, and G. E. Oien, “Design of Optimal High-Rank Line-of-Sight MIMO Channels,” *IEEE Transactions on Wireless Communications*, vol. 6, no. 4, pp. 1420–1425, Apr. 2007.

[52] F. Bohagen, P. Orten, and G. Oien, “Optimal Design of Uniform Rectangular Antenna Arrays for Strong Line-of-sight MIMO Channels,” *EURASIP J. Wirel. Commun. Netw.*, vol. 2007, no. 2, pp. 12–12, Jan. 2007.

[53] S. Hanna, H. Yan, and D. Cabric, “Distributed uav placement optimization for cooperative line-of-sight mimo communications,” in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2019, pp. 4619–4623.

[54] N. Chatzipanagiotis, Y. Liu, A. Petropulu, and M. M. Zavlanos, “Controlling groups of mobile beamformers,” in *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*, Dec. 2012, pp. 1984–1989.

[55] S. Hanna, E. Krijestorac, H. Yan, and D. Cabric, “UAV Swarms as Amplify-and-Forward MIMO Relays,” in *2019 IEEE 20th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, Jul. 2019, pp. 1–5.

[56] W. Du, W. Ying, P. Yang, X. Cao, G. Yan, K. Tang, and D. Wu, “Network-Based Heterogeneous Particle Swarm Optimization and Its Application in UAV Communication Coverage,” *IEEE Transactions on Emerging Topics in Computational Intelligence*, pp. 1–12, 2019, citation Key Alias: du_network-based_2019-1.

[57] T. Cabreira, L. Brisolara, and P. R. Ferreira Jr., “Survey on Coverage Path Planning with Unmanned Aerial Vehicles,” *Drones*, vol. 3, no. 1, p. 4, Jan. 2019.

[58] E. Galceran and M. Carreras, “A survey on coverage path planning for robotics,” *Robot. Auton. Syst.*, vol. 61, no. 12, p. 1258–1276, Dec. 2013.

[59] C. Deng, S. Wang, Z. Huang, Z. Tan, and J. Liu, “Unmanned Aerial Vehicles for Power Line Inspection: A Cooperative Way in Platforms and Communications,” *Journal of Communications*, vol. 9, no. 9, pp. 687–692, 2014.

[60] “Automatic system for overhead power line inspection using an Unmanned Aerial Vehicle — RELIFO project,” May 2013, pp. 244–252.

[61] Y. Zhang, X. Yuan, Y. Fang, and S. Chen, “UAV Low Altitude Photogrammetry for Power Line Inspection,” EARTH SCIENCES, preprint, Aug. 2016.

[62] Y. Zhang, X. Yuan, W. Li, and S. Chen, “Automatic Power Line Inspection Using UAV Images,” *Remote Sensing*, vol. 9, no. 8, p. 824, Aug. 2017.

[63] V. T. Hoang, M. D. Phung, T. H. Dinh, and Q. P. Ha, “Angle-Encoded Swarm Optimization for UAV Formation Path Planning,” *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5239–5244, Oct. 2018, arXiv: 1812.07873.

[64] V. T. Hoang, M. D. Phung, T. H. Dinh, and Q. Ha, “System architecture for real-time surface inspection using multiple uavs,” 07 2019.

[65] L. Duque, “UAV-Based Bridge Inspection and Computational Simulations,” Master’s thesis, South Dakota State University, 2017.

[66] C.-H. Yang, M.-C. Wen, Y.-C. Chen, and S.-C. Kang, “An optimized unmanned aerial system for bridge inspection,” in *ISARC. Proceedings of the International Symposium on Automation and Robotics in Construction*, vol. 32. IAARC Publications, 2015, p. 1.

[67] J. Seo, L. Duque, and J. Wacker, “Drone-enabled bridge inspection methodology and application,” *Automation in Construction*, vol. 94, pp. 112–126, Oct. 2018.

[68] D. Hausamann, W. Zirnig, and G. Schreier, “Monitoring of Gas Transmission Pipelines – a Customer Driven Civil UAV Application,” in *ODAS Conference*, 2003, p. 15.

[69] A. Shukla, H. Xiaoqian, and H. Karki, “Autonomous tracking of oil and gas pipelines by an unmanned aerial vehicle,” in *2016 IEEE 59th International Midwest Symposium on Circuits and Systems (MWSCAS)*, Oct. 2016, pp. 1–4, iSSN: 1558-3899.

[70] L. I. Kochetkova, “Pipeline monitoring with unmanned aerial vehicles,” *Journal of Physics: Conference Series*, vol. 1015, p. 042021, May 2018.

[71] S. Waharte and N. Trigoni, “Supporting Search and Rescue Operations with UAVs,” in *2010 International Conference on Emerging Security Technologies*, Sep. 2010, pp. 142–147, iSSN: null.

[72] P. Cohn, A. Green, M. Langstaff, and M. Roller, “Commercial drones are here: The future of unmanned aerial systems,” McKinsey & Company, Tech. Rep., 2018.

[73] A. Ahmadzadeh, J. Keller, G. Pappas, A. Jadbabaie, and V. Kumar, “An optimization-based approach to time-critical cooperative surveillance and coverage with uavs,” in *Experimental Robotics*. Springer, 2008, pp. 491–500.

[74] S. Hayat, E. Yanmaz, and R. Muzaffar, “Survey on Unmanned Aerial Vehicle Networks for Civil Applications: A Communications Viewpoint,” *IEEE Communications Surveys & Tutorials*, vol. 18, no. 4, pp. 2624–2661, 2016.

[75] Remcom, “Wireless EM Propagation Software - Wireless InSite,” https://www.remcom.com/wireless-insite-em-propagation-software.

[76] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard, “Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age,” *IEEE Transactions on Robotics*, vol. 32, no. 6, pp. 1309–1332, 2016.

[77] G. Bresson, Z. Alsayed, L. Yu, and S. Glaser, “Simultaneous localization and mapping: A survey of current trends in autonomous driving,” *IEEE Transactions on Intelligent Vehicles*, vol. 2, no. 3, pp. 194–220, 2017.

[78] M. Servières, V. Renaudin, A. Dupuis, and N. Antigny, “Visual and visual-inertial SLAM: State of the art, classification, and experimental benchmarking,” *Journal of Sensors*, vol. 2021, p. e2054828, Feb. 2021.

[79] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós, “ORB-SLAM: A versatile and accurate monocular SLAM system,” *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

[80] R. Mur-Artal and J. D. Tardós, “Visual-inertial monocular SLAM with map reuse,” *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 796–803, 2017.

[81] ——, “ORB-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-D cameras,” *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.

[82] T. Qin, P. Li, and S. Shen, “VINS-Mono: A robust and versatile monocular visual-inertial state estimator,” *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 1004–1020, 2018.

[83] T.-K. Chang, A. Pogue, and A. Mehta, “Boem-slam: A block online em algorithm for the visual-inertial slam backend,” in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022, pp. 6420–6427.

[84] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. Leonard, and F. Dellaert, “iSAM2: Incremental smoothing and mapping with fluid relinearization and incremental variable reordering,” in *2011 IEEE International Conference on Robotics and Automation*, 2011, pp. 3281–3288.

[85] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. J. Leonard, and F. Dellaert, “isam2: Incremental smoothing and mapping using the bayes tree,” *The International Journal of Robotics Research*, vol. 31, no. 2, pp. 216–235, 2012.

[86] Z. Sjanic, M. A. Skoglund, and F. Gustafsson, “EM-SLAM with inertial/visual applications,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 53, no. 1, pp. 273–285, 2017.

[87] S. Le Corff and G. Fort, “Online expectation maximization based algorithms for inference in hidden Markov models,” *Electronic Journal of Statistics*, vol. 7, pp. 763–792, 2013.

[88] S. Le Corff, G. Fort, and E. Moulines, “New online EM algorithms for general hidden Markov models. Application to the SLAM problem,” in *Latent Variable Analysis and Signal Separation*, ser. Lecture Notes in Computer Science, F. Theis, A. Cichocki, A. Yeredor, and M. Zibulevsky, Eds. Berlin, Heidelberg: Springer, 2012, pp. 131–138.

[89] T. Dumont and S. Le Corff, "Online EM for indoor simultaneous localization and mapping," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, May 2013, pp. 6431–6435, iSSN: 2379-190X.

[90] ——, "Simultaneous localization and mapping in wireless sensor networks," *Signal Processing*, vol. 101, pp. 192–203, Aug. 2014.

[91] G. Bourmaud, R. Mégret, A. Giremus, and Y. Berthoumieu, "Discrete extended Kalman filter on Lie groups," in *21st European Signal Processing Conference (EUSIPCO 2013)*, 2013, pp. 1–5.

[92] J. Solà, J. Deray, and D. Atchuthan, "A micro Lie theory for state estimation in robotics," *arXiv:1812.01537 [cs]*, 2020.

[93] S. Agarwal, K. Mierle, and Others, "Ceres solver," http://ceres-solver.org.

[94] A. Pogue, S. Hanna, A. Nichols, X. Chen, D. Cabric, and A. Mehta, "Path planning under mimo network constraints for throughput enhancement in multi-robot data aggregation tasks," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 11 824–11 830.

[95] A. Goswami, B. Espiau, and A. Keramane, "Limit cycles in a passive compass gait biped and passivity-mimicking control laws," *Autonomous Robots*, vol. 4, no. 3, pp. 273–286, Sep 1997.

[96] G. Bledt, P. M. Wensing, S. Ingersoll, and S. Kim, "Contact model fusion for event-based locomotion in unstructured terrains," *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1–8, 2018.

[97] R. Tedrake, "Underactuated robotics: Algorithms for walking, running, swimming, flying, and manipulation," 2023, accessed: 2024-06-17. [Online]. Available: http://underactuated.mit.edu/underactuated.html?chapter=simple_legs