Algorithm Design for Real-Time Path Planning of Underactuated Dynamic Walking Using Virtual Constraints

A thesis submitted in partial fulfilment of the requirements for the degree of Bachelor of Engineering (Honours) and Bachelor of Science

Stephen Wardrop

September 3, 2014

Abstract

There is significant interest, both technical and aesthetic, in bipedal robots which locomote stably, efficiently and reliably over uneven terrain. This continues to present a significant challenge due to computational intractability, highly complex nonlinear dynamics and intrinsic static and dynamic instability. Successful methods which achieve walking gaits restrict the problem to particular domains or motions to overcome the intractability of the general problem. One such method is to prepare a library of motion primitives and thus to limit the on-line computation to the choice of a particular constraint within the library.

Previous work has demonstrated the in-principle effectiveness of the motion primitives approach for real-time path planning of dynamic walking over uneven terrain. The contribution of this thesis work is to present a general method for the production of a library of motion primitives which achieves sufficient coverage of the feasible motions of the robot along with a method for intelligently choosing between them.

Statement of student contribution

- I completed the literature review largely independently, with some papers being recommended to me by my supervisor
- I designed and implemented the simulator for the compass-gait robot and 5-link walker, with the code for generating the dynamics of the 5-link walker being sourced from Westervelt et al.
- I designed the method by which the motion primitives could be automatically generated (yet to complete...)
- I extended the algorithm designed by my supervisor and others to include a more intelligent heuristic for selection of motion primitives in real time. (yet to complete...)
- I carried out the experiments on the compass-gait walker, which was constructed prior to the commencement of my thesis work. (Hopefully)

The above represents an accurate summary of the student's contribution.

Signed ______________________ (student) ______________________ (supervisor)

Acknowledgements

I would like to thank my supervisor, Dr Ian Manchester, for his time and forbearance in helping me grasp the matter of this field. Along with him, I thank Jack Umemberger for his helpful comments and insights to get me up to speed with his previous work in this matter.

I thank <anybody who did this> for proofreading and advice given.

I also thank my wife for enduring the countless nights and weekends spent with a preoccupied or absent husband even while taking care of our young daughter. Your encouragement and patience have been vital.

Contents

List of Figures

List of Tables

CHAPTER 1 Introduction

Motivated by the dexterity, terrain scalability and reach availed to humans through the use of legs, as opposed to wheels, and for social and aesthetic reasons, robotics researchers have sought to create humanoid robots. While there are notable examples of significant success in this endeavour, the motion characteristics of these robotic walkers are noticeably divergent from our own. This is largely due to the restricting of the robotic systems to full actuation, a domain which avails to the control system designer convenient simplifying properties and robustness. This restriction imposes more than simply visual differences; the conservative motion restricts the speed at which these robots can move and, importantly, the efficiency with which the walkers achieve motion. Due to these facts, there exists interest in developing underactuated robotic walkers, which utilise the dynamics of the system to generate much of the motion, imposing actuation only where required to enforce a trajectory or respond to disturbance.

However, the path planning and control problems become significantly more complex outside of the domain of fully actuated motion. This is mainly due to two contributing factors. Firstly, the dynamics of walking robots are highly nonlinear, particularly about the foot impacts, thus the system model must be very complex, or else be a poor approximation. Secondly, the search space to find feasible paths is vast, since in general robotic walkers have many independent degrees of freedom, making traditional methods of planning and control computationally intractable. In addition, control problems of underactuated and fully actuated control are sufficiently different that much of the intuition and methods from the quite well studied field of fully actuated control are not easily applicable to underactuated cases.

Manchester et al [25] have proposed a method for path planning for underactuated dynamic walkers using a receding-horizon algorithm which draws from a library of motion primitives, defined as virtual holonomic constraints, to address the above mentioned challenges. The use of a library of motion primitives drastically reduces the search space and notionally limits the trajectories to feasible motion, subject to the avoidance of collisions. The difficulty of the nonlinear impact dynamics is isolated by the choice of motion primitives as being defined by the duration of one continuous phase – that is, the movement of the robotic walker between one foot impact and the next. Also, the use of virtual holonomic constraints allows for a partial closed-form solution of the dynamical equations to be computed off-line, thereby reducing the required on-line computation. This method has been proven to be capable of planning several footsteps ahead of the current position of the robot in a fraction of a second, thus being feasible for real-time control, on a compass-gait walker and more complex 5-link walker.

The ability of this approach to produce paths close to the true optimal is dependent upon the library of motion primitives and the manner by which one primitive is preferred over another. The current algorithm uses a greedy best-first search, with demonstrable improvements made in utilising an energy heuristic. The efficiency with which the algorithm obtains its result and its convergence to the most energy-efficient feasible path are the key performance indicators for the algorithm. Algorithmic efficiency is directly related to the required on-line computation. A less efficient algorithm will therefore consume more power (and time) to produce its result. Also, the power required to achieve the desired motion of the robot will clearly be greater if the path is less energy efficient. Naturally, more efficient algorithms which produce results which require less energy to achieve motion will reduce power consumption on board the robotic walker. Since power and energy densities pose strict limits on the dimensions and weight of walking robots with present technology, and due to ethical considerations in sustainable design, reducing energy consumption carries great importance.

As such, the work of this thesis is to refine the algorithms to enable efficient and robust real-time path planning of underactuated dynamic walkers. The algorithms demonstrably improve upon the current algorithms in running time and better convergence with optimal trajectories. This is realised by the design of data structures which are compatible with the implicit orderings of motion primitives paired with intelligent partitioning to enable an average case search time of $O(\log n)$. The path planning algorithm is improved such that the traversal of the decision tree down infeasible paths is guaranteed to be minimised and the choice of motion primitives is guaranteed to produce more energy-efficient motion in comparison to the current methods. This is achieved in a two-fold manner. Firstly, the library of motion primitives is generated automatically to optimally span the configuration space within some bounds. Secondly, the algorithm to choose between primitives is improved with an intelligent heuristic that is informed by the shape of the upcoming terrain. The algorithms are verified by simulation on a compass-gait and more complex walkers, and by experiment on a compass-gait walker.

CHAPTER 2 Literature Review

2.1 Background

This thesis work is focused on path planning for underactuated walking robots using virtual holonomic constraints as motion primitives. The objective is to allow for robust real-time approximately optimal control of underactuated walkers on rough terrain. This is based upon previous work in several fields; walking robot dynamics, motion planning and the application of virtual constraints in control and path planning. It also relies upon feedback control to enforce the desired trajectory.

Walking robots are a subset of mobile robots. Other common methods of locomotion include wheels, rails and tracks. Walking robots have gathered interest for their ability to traverse terrain which is impassable to these other classes of mobile robot. However, walking robots are typically significantly less efficient and difficult to plan and control in comparison to wheeled robots. In order to make legged locomotion viable, robust, efficient and rapid locomotion must be achievable. Also, other than for the limited applications in which motion can be pre-planned, it must be computable in real-time.

Feedback control has been a very important field of research in generating robust and useful robots. It has proven exceptionally effective at ensuring robotic motions match planned motions, particularly in systems of sufficiently low dimensionality and complexity. For conventional wheeled and fixed robots, feedback control laws are normally well-understood and behaviour can be prescribed which will be reliably and robustly tracked. However, for legged robots, especially underactuated robots, the required feedback control is not obvious. There has been a large amount of recent work in analysing the stability of control schemes in periodic and non-periodic walking cycles.

Motion planning has been a focus of robotics research from the beginnings of autonomous mobile robots. Simple conventional approaches which are applicable to wheeled robots have been developed which allow for very efficient motion planning to achieve particular goals within set of constraints. While the principles of motion planning are similar in legged robots, these techniques are most often not applicable due to the highly nonlinear, nonsmooth, nonconvex and discontinuous nature of legged walking. As such, there has been much work in recent times to address some of the problems of *kinodynamic* planning; planning which includes constraints on the velocities as well as the positions, as opposed to simply planning based upon positions.

In the last decade, virtual constraints have arisen as a method for reducing the complexity of the required computation for underactuated robotic walkers. The use of virtual constraints has allowed for the potentially high-dimensional dynamics to be reduced to dynamics in a single variable, which avails significantly simpler analysis. The advantages of utilising virtual constraints has been verified by many sources in contemporary literature.

This literature review will explore the classes of walking robots and motivate the need for developing controllable and robust underactuated walkers. Feedback control schemes will be explored, with a particular emphasis on those applicable to underactuated walkers. Motion planning methods and architectures are discussed, with a focus on the recent innovations in using constraints-based planning. The recent advancements in the understanding of the utility and properties of virtual holonomic constraints are expounded, with a view to implementing them as motion primitives for an underactuated dynamic walker. The literature survey is concluded by highlighting the state of the art of motion planning of underactuated robots, particularly those subject to virtual constraints, and explores the contributions of this thesis work in filling needs for solutions in this field.

2.2 Walking Robots

This thesis work is directed at the generation of stable *dynamic* walking for bipedal robots. This is a significantly more difficult problem than *static* walking. Static walking is the application of legged locomotion such that at all times the robot is statically stable. Such robots were built as early as the 1980's, see e.g. [32, 42]. It is worth noting that statically stable walking is only achievable in robots with at least four legs.

Interest in dynamic walking arises due to the fact that static walkers operate with restricted speed and efficiency in order that inertial effects remain minimal. Bipedal walking is especially of interest due to its similarity with human locomotion. There has been recent work in applying principles revealed in the study of bipedal dynamic walkers to improving rehabilitation of amputees with prosthetic limbs [27]. Bipedal walkers can be classed into three different broad types on the basis of how they are controlled; fully actuated, underactuated and passive.

2.2.1 Fully actuated bipedal walking robots

Full actuation presents a number of simplifying properties that make path planning and control design significantly easier than for underactuated systems. As a result, control of walking robots is made much simpler if the robots can remain fully actuated. This is the mode of control naturally applicable to static walking. It is possible to ensure bipedal robots remain in the realm of full actuation by restricting the motion such that the centre of pressure remains at all times underneath one of the robot's feet.

Possibly the most prominent example of a fully actuated bipedal walking robot is Honda's ASIMO robot. In [8], the process by which footsteps are planned for ASIMO are detailed. While not a trivial process, the choice of footstep placement is driven primarily by kinematic concerns. There are dynamic constraints on the robot which exist in order that it remains within the realm of full actuation, but within those constraints, footstep placement can be arbitrary. [7] outlines the robot's ability to extend this to predictably moving obstacles. In both examples footstep placement is achieved by employing the A* dynamic programming algorithm searching through a gridded map of the environment along with a set of possible actions that the robot's swing leg can achieve.

However, the ease with which fully actuated robots are controlled and guided come at significant cost; there are many sources, e.g. [1,5,28], which suggest that energy-efficient walking is achievable in imitating passive dynamic walking as closely as possible. That is, there is significant consensus that in order to achieve energy-efficient gaits, underactuated control must be utilised. This is intuitively clear; restricting motion to full actuation is in many ways similar to restricting the robot to static stability. In fact, in many cases the two are equivalent. From trivial observation of human gaits, we understand that efficient, rapid locomotion is generated in gaits which significantly diverge from fully actuated control, especially during accelerated motion such as running or jogging, where a large proportion of the motion is without contact with the ground.

2.2.2 Passive dynamic walkers

Interest in studying passive dynamic walkers as a key to unlocking efficient bipedal walking and better understanding human gaits was sparked by McGeer's seminal paper on the subject [28]. A class of mechanical systems for which there are natural stable periodic gaits which require no active control were introduced. This includes walking robots which have stiff legs as well as those which have articulated knees. McGeer validated the theory through experiment with a stiff-legged biped which was designed with four legs – outer legs were connected by a crossbar and the inner legs were fixed together such that the motion is constrained to two dimensions, simulating the theoretical saggital-plane (side-on) compass gait walker.

In [11], a more sophisticated passive dynamic walking robot was developed which verified by experiment the generalisation of McGeer's work to three-dimensional kneed bipedal walkers. This robot also had arms which provided counterweights to improve dynamic stability. The robot was able to generate stable walking gaits on a 3.1° slope, consuming 1.3 W to achieve 0.51 m/s forward-motion. This may be compared to its contemporary Honda P3 robot, which required 2 kW during walking.

However, passive dynamic walkers cannot feasibly provide reliable walking motions for general mobile robots, since they require downhill trajectories and are unable to be controlled, other than in setting initial conditions. They also suffer from high sensitivity to disturbances and require initialisation close to their dynamic equilibrium. For example, the Collins 3D walker was only able to be successfully launched 80% of the time, even with a practiced hand. It also suffered from low directional stability, unable to complete walking down a fairly narrow, straight 5m ramp without falling off in the majority of cases.

2.2.3 Underactuated dynamic walkers

The problems of a lack of controllability and robustness in passive dynamic walkers as well as a lack of efficiency in fully actuated walkers seem to find their natural solution in underactuated dynamic walking robots. Indeed, in both [11,28], the authors envisioned that the robots should be actuated to provide sufficient energy to walk on flat ground or climb hills. In [39], this was achieved; a simple stiff-legged 3D dynamic walker was actuated to allow for periodic walking up shallow slopes. Also in this walker, by engineering the feet such that the curve of the foot is higher than the centre of mass, standing still was made a statically stable configuration, a property that the above-mentioned passive walkers lack.

A slightly more mechanically complicated walker based upon the Collins passive dynamic walker is presented in [10]. Here, however, there are far more actuators present; they are used sparingly, to keep the motion close to the natural passive gait. It includes hip actuation as normal, ankle actuation used for pre-impact toe-offs and some active control to ensure that the knees lock and unlock at convenient times. The controller is very simple, implemented in 68 lines of C++ code as a set of switches to synchronise the actuators. The robot was designed to generate very efficient stable periodic walking which is more robust than in the passive dynamic walker, and to be able to walk on level ground. While the active control did produce the ability to walk on level ground, its robustness was still poor.

Much of the literature in underactuated dynamic bipedal walker control is, similarly to [10, 39], focused on the application of torques to achieve stable periodic motion, see e.g. [20, 33, 35]. This regulates the behaviour of the robot to the stable periodic cycles inherent within the dynamics of the unactuated walker. This method allows for the most energy-efficient stable walking possible, given the particular dynamics of the mechanical system, under the assumption of locomotion over somewhat flat ground. However, this approach does not provide a feasible way to efficiently and feasibly move through uneven terrain, since the periodic cycles generated may collide with step-ups in the environment as well as fail to adjust the motion intelligently to increases and decreases in gravitational potential energy.

Recently, attempts have been made to produce underactuated walking control methods to allow for more optimal walking on rough terrain. In [5], this was achieved by using a receding-horizon approach, using a value iteration algorithm [37] to find an approximately optimal step-to-step feedback policy chosen from a mesh of post-collision states. This used a combination of PD control to regulate the hip angle and an impulsive pre-collision toe-off which was verified by simulation to allow the robot to negotiate complex terrain successfully.

Underactuated robotic control is normally executed primarily at the hip, see e.g. [5, 24, 39], however the use of ankle-only actuation has been studied in [12]. Here, it is shown that by choosing particular mass distributions and foot shapes, stability and robustness of convergence to the limit cycle can be improved. An important result was also to show that pre-impact push-off of the stance leg improves mechanical energy efficiency by 25%. Of course, this requires the existence of feet on the ends of the robot's legs. In [1], it is demonstrated that the use of curved feet and judicious application of hip torque is sufficient to simulate the effects of actuated ankles. [26] confirms the efficiency benefit of curved feet over point feet.

Underactuated systems may classified on the basis of an important measure; the *degree* of underactuation. This is the number of generalised coordinates which describe the state of the system which are unactuated. Many of the applications of underactuated control in the literature deal with underactuation degree one systems, typically with the ankle angle being the single unactuated coordinate, e.g. [5, 43]. However, this typically applies only to saggital-plane models or robots specifically designed to exhibit two-dimensional motion characteristics. Even simple compass-gait-like walkers such as in [39] introduce many unactuated degrees of freedom when expanded to three-dimensional walking.

It is important to note that while underactuatuation in robotic walkers often implies that there are states which are not actuated by mechanical design, underactuation in general is dependent upon the state. That is, many of the examples of dynamic walkers are underactuated by virtue of having joints which do not have any corresponding control, in particular models of walkers with point-terminated legs, see e.g. [43]. However, underactuation includes a much broader class of robots and is based upon the control strategy. This is trivially proven by considering a robot with direct control of each joint for which the control strategy involves one or more zero-torque inputs. More generally, a robot enters the domain of underactuated control whenever the control inputs cannot arbitrarily alter the state evolution. This is more precisely mathematically defined in Section 3.1.

The underactuated robots which form the focus for this thesis work are saggital-plane underactuation degree one robots with no ankle actuation. This matches a large proportion of the literature and provides simplifying assumptions which allow for the development of techniques without imposing large difficulties in generalising to three-dimensional walkers.

2.3 Feedback Control

Control systems may be classed broadly in two different types; *open-loop* or *feedforward* control and *closed-loop* or *feedback* control. Open-loop control involves calculating the control inputs required for a desired outcome in the actuated system and simply feeding those values directly to the actuators. Closed-loop control instead compares the behaviour of the system to some reference behaviour and applies corrective control. While feedforward control is simpler conceptually, it requires a very accurate model of the system in order to achieve the desired results. In systems such as underactuated walkers with very complicated nonlinear dynamics, accurately modelling the system is both analytically and computationally difficult. Therefore, while some aspects of feedforward control can be used to improve the stability and responsiveness of controllers, feedback control is necessary to enforce the system's compliance with its target trajectory.

The most simple and universal form of feedback control is PID (Proportional Integral Differential) control [2]. It is a very simple controller, applying simple linear gains on the error, its derivative and its integral, but is very powerful and covers most control problems. PID controllers are very well studied; as early as 1986, there were very comprehensive design rules and guides on how to apply PID control to systems with many different models [31].

However, In the case of underactuated dynamic walkers operating under virtual constraints (see Section 2.5), PID control does not typically track the desired trajectory efficiently or reliably. In order to track the desired trajectory satisfactorily, a more intelligent choice of controller is required. This may include some aspect of feedforward control in which the controller attempts to enforce the constraint based upon some knowledge of the system dynamics and the feedback process merely corrects any deviation. This is broadly known as hybrid-zero dynamics control; the controller's internal dynamics are designed to match the zero dynamics of the system. This is shown to improve tracking in [35] and is proven to be robust to reasonable model uncertainty in [26] by applying a HZD controller with the assumption of point feet to a robot with curved feet.

In [24], a provably stabilising controller for an underactuated robot traversing rough terrain based on a modified Model Predictive Control (MPC) method combined with the notion of *transverse linearisation* is described and compared to a hybrid zero dynamics based controller. Importantly, the controller is designed for *non-periodic* trajectories. This is significant, since most work in controllers for underactuated walkers, e.g. [26, 30, 35] is primarily on generating stable periodic gaits. In particular, hybrid zero dynamics controllers are in general only provably stabilising for periodic gaits. This limits the ability of the robot to intelligently place its feet and to traverse rough terrain. The ability to reliably stabilise non-periodic gaits is important to the robustness and stability of a path planner such as the object of this thesis.

This thesis is largely theoretical; designing improvements to path planning algorithms assuming that the constraints are perfectly regulated. It is understood that in practice, these constraints must be enforced by feedback control. Hybrid zero dynamics control will in theory perfectly regulate the configuration constraints requested by the motion planner, however, this entirely ignores kinetic concerns, therefore the assumptions made about impact conditions may be violated. Manchester et al's transverse linearisation controller will alter the path such that the velocity constraints are well regulated at each impact. Given the existence of suitable alternatives to enforce the trajectories set by the planning algorithm, this thesis work will not place a large emphasis on control.

2.4 Motion Planning

2.4.1 Motion planning architectures

In order to be capable of achieving collision-free motion and to achieve objectives, mobile robots require plans of trajectories through their environments. In prepared and sterilised spaces, it is possible to achieve this in an entirely "open-loop" manner; all motion can be pre-planned and the robot exerts forces and torques to achieve the motion without regard to its state. It is clear that such planning is not robust to unexpected changes in the environment and requires complete knowledge of the system dynamics. This method of motion planning is found in the works of automata dating back as far as the 4th century BC, where Greek mathematician Archytas of Tarentum designed a steam-propelled mechanical bird.

This approach can be improved by the application of feedback control, as discussed in Section 2.3; while the trajectory is still set without any regard to the environment, the control inputs to track the trajectory are adjusted on-line to correct any deviations from the desired trajectory. Therefore, the motion of the mobile robot is made more robust to disturbances and errors in the dynamical model. It is important to note that while the control is closed-loop, the path planning is still computed in a wholly open-loop manner. This control scheme is suitable for robotic systems such as automated manufacturing plants and robots operating under holonomic constraints such as rails where wholly predefining the motion is applicable and safe. However, in general, mobile robotic platforms are not expected to remain in carefully prepared environments and thus the path that the robot takes must be informed by the terrain through which it must travel. Naturally, any approach to motion planning in general environments introduces significant complexity, since the goals of the robot may be significantly less clear and the actions required to optimally or even simply feasibly achieve such goals can prove very difficult to evaluate.

An approach which attempts to address the need to locomote with regard to the environment is the so-called SENSE-PLAN-ACT architecture, as successfully implemented in the Shakey Robot at Stanford Research Institute [29]. This introduces the important notion of using sensors to gather information about the environment to inform the robot's motion planner. It ought to be noted that this approach retains some sense of open-loop planning; the environment is assessed, a path through the environment is devised, then the action is executed. If there are any changes in the environment, or the model of the space has not been correctly generated, there are no means to correct this until the planned path is completed. Another limitation of the SENSE-PLAN-ACT architecture used in Shakey is the inability to service multiple concurrent goals. A solution to this limitation which retains the fundamental structure of the architecture is presented in [40]; concurrent tasks are scheduled, respecting time constraints and priorities. Planning with such open-loop architectures shall hereupon be labelled *deliberative* planning.

In response to the lack of robustness offered by deliberative planning, *reactive* architectures were devised. A compelling architecture based upon Behaviour Based control, known as *subsumption* is proposed in [4]. Brooks' subsumption builds robot behaviour by combining functional blocks whose behaviour may subsume that of lower-level blocks. These functional blocks describe robot behaviours such as following a wall and avoiding collisions. The combination of these simple reactive blocks was shown to be capable of producing a viable walking gait, amongst other high-level behaviours. However, such architecture is not easily used to service high-level goals. Alternative reactive architectures, such as that proposed in [18] include a complex closed-loop interaction between deliberative and reactive components. For example, the high-level planner can be interrupted if a collision is imminent. These allow for high-level goals to be set while also ensuring robustness to environmental or robot state changes. These are extended in [17], which presents a heterogeneous, asynchronous architecture which allows for a robot with multiple tasks to be controlled in a noisy environment. The asynchrony is important for the robustness of the architecture, using high-level planning to guide the robots actions, but not control them directly.

The motion planning algorithm proposed within this thesis work attempts to provide a robust, computationally efficient method of producing stable dynamic walking in underactuated robots. Within the framework of planning architectures, it may be considered to a functional block which could fit into Brook's subsumption architecture or Gat's heterogeneous, asynchronous planning architecture. It is not a complete solution for robot motion planning in itself; in its current form, this algorithm simply allows for sustained walking in a straight-line path. Clearly, this must be combined with other planning modules, depending on the purpose of the robot.

2.4.2 Path planning methods

Path planning methods define a trajectory through space which is constructed in service of a given set of goals and subject to a set of constraints. The optimal path is typically understood to be that which minimises the energy or time required to complete the motion. However, in many instances, it is understood that finding the true optimal path is computationally intractable, thus methods by which a feasible solution is guaranteed to be found, if one exists, are normally acceptable. Methods which produce a solution more quickly (i.e. are more computationally efficient) or produce a path which more closely resembles optimum are considered more favourable. In practice, most path planning methods establish a compromise between computational efficiency and convergence to the optimum path.

Conventional approaches to path planning include grid-based searches, which discretise the configuration space and use graph traversal algorithms to compute the optimum path. More computationally efficient methods such as Probabilistic Road Maps [3] can be used to achieve paths which converge to the optimal path with increasing sampling points. Potential field methods, which assign an attractive "force" to the goal pose and a repulsive force to the obstacles provide a means for avoiding complex geometric collision checks and attempt to place the robot with some reasonable clearance from obstacles. These methods are applicable to fully actuated robots since all that is required is to find a kinematically feasible path through the configuration space. However, since underactuated robots are subject to differential constraints, such methods will typically produce paths which are infeasible.

The most direct and obvious way to solve the path planning problem for underactuated robotic walking, therefore, is to pose it as an optimisation problem over state and control trajectories. This may be solved using nonlinear programming methods. However, since walking motion planning problems are typically high-dimensional, nonsmooth and nonconvex, these methods are very computationally expensive. [38] suggests the use of a smooth contact model to avoid the necessity of hybrid representations of dynamics introduced by the contacts. This results in the ability to formulate an unconstrained, continuous trajectory optimisation problem, which is solvable using standard nonlinear optimisation tools. However, with current technology, this formulation still requires minutes of computation time, which is clearly not applicable to real-time trajectory planning.

Rapidly-expanding Random Trees (RRT) [23] provide a method for *kinodynamic* planning; planning trajectories in state space, not just configuration space. This allows for nonholonomic differential constraints such as those present on the underactuated dynamic walker to be considered in the planning algorithm. These trees also allow for the curse of dimensionality to be somewhat mitigated in comparison to gridding approaches. In [13], real-time planning was achieved for a helicopter through the use of a modified RRT algorithm. However, for the infinite-dimensional case of planning walking motions, RRT's are not directly applicable.

Motion primitives provide a way to define feasible behaviour of the robot and to reduce the search space for the path planning problem. [9] demonstrates the utility of motion primitives in planning motions in cluttered environments and is validated on a 7DOF manipulator. In [14], motion primitives are used to produce near-optimal motions in non-linear systems with symmetries and validated in simulations with a helicopter. The combination of the use of motion primitives and RRT's to reduce search time is explored in [41]. This approach reduces the number of iterations required to find a feasible path to the goal compared to using an unmodified RRT algorithm, but naturally limits the motion to being composed of a set of the defined primitives. Another approach in which RRT's were combined with motion primitives, [34], was able to produce bounding motions in the quadruped LittleDog robot, however, the computation time was prohibitive for real-time applications.

In [25], the use of virtual holonomic constraints as motion primitives is presented. These primitives correspond to the continuous phase dynamics between each impact. There are numerous advantages to choosing primitives in this way which are explored in Section 2.5. This choice of method for motion planning forms the basis for this thesis work.

2.5 Virtual Holonomic Constraints

Virtual holonomic constraints are a method of forcing a nonholonomic system (that is, a system whose state evolution is subject to differential constraints) to adhere to a path through configuration space through the application of feedback control. This method of control is demonstrated in [33] as a tool for orbital stabilisation of underactuated nonlinear systems. The form of the zero dynamics of a virtually constrained physical system are presented. This formulation describes the dynamics of a *phase variable* to which all other coordinates are synchronised, under the assumption that the feedback control is perfectly regulated. Shiriaev et al prove that this equation has a general integral of motion and utilise it to reduce the complexity of the dynamics to allow for a more easily solvable control problem.

In [16], this property is used in a search for periodic cycles in the gait of a 2DOF compass-gait walker, which is otherwise a very computationally expensive task due to the impulsive impact conditions and nonlinearities in the dynamics. In [15], orbitally stable periodic motions of a two-link underactuated robot called the Pendubot were generated, again by exploiting this property. [6] surveys the use of virtual constraints in balancing and walking control systems in robots, particularly the Rabbit 7-DOF walker. Much of the uses of virtual constraints arise from the reduced dynamics availed by considering only the phase variable and are simplified further by the existence of the integral. Note, however, that the integral is not guaranteed to exist under all conditions.

In [24] and [25], the integral is slightly rearranged to yield a partial closed-form solution which relates the velocity of the phase variable to the phase variable itself. This allows for a partial solution of the zero dynamics to be computed *off-line*, greatly simplifying the required real-time computation for path planning and control. Along with this performance outcome, using virtual constraints as primitives offers two other advantages; they specify kinematic paths thus much of the reasoning applied to classical path planning problems may be applied, and the partial closed form solution yields an affine structure which affords an intelligent ordering. The existence of a clear ordering between primitives is significant, since it allows for much more efficient searches through sets of primitives (See Section 2.5.2).

2.5.1 Hybrid Zero Dynamics (HZD)

Understanding the motion of underactuated dynamic walkers subject to virtual constraints requires the analysis of the *hybrid zero dynamics*. This is the combination of continuous phases, where the motion is defined by a choice of virtual constraint, interspersed with impacts, which represent a discontinuity in the evolution of the state space. This approach was introduced in [20], proving that the use of virtual constraints allowed for a computationally tractable evaluation of asymptotic stability and for the construction of a stabilising controller. In [43], exponentially stable walking controllers for general underactuation degree one planar walkers were designed. Conditions were derived for periodic cycles that ensure that the impacts do not shift the system off the hybrid zero dynamics manifold, known as the *invariance* of the zero dynamics.

Fast dynamic walking of a multiple-degree underactuation walker was achieved in [35] by the application of full hybrid zero dynamics control schemes. The performance of this control was compared to classical PD control and was shown to be much more efficient and yields a much more accurate realisation of the desired virtual constraints. These benefits of the control scheme over PD control were very significant in achieving efficient rapid locomotion.

The controller strategies employed and suggested in this thesis work rely upon a HZD controller to ensure that, as in [35], the desired virtual constraints are reproduced faithfully. In addition, care must be taken in the construction of the motion primitives to ensure that the hybrid zero dynamics remain invariant.

2.5.2 Selection of motion primitives

In order to use motion primitives in path planning as a way to respond to the upcoming terrain, a rich library of motion primitives is required. Also required is a manner by which one primitive may be chosen over another. In [25], two important results are derived. First, the system must have sufficient energy such that the chosen primitive will complete its specified path and not fall back. This reduces to a simple mathematical check. Second, there is a method for logical ordering of the primitives. Both of these are based upon the concept of a *critical velocity*; the velocity of the phase coordinate as it passes through the *critical point*, the single point of peak potential energy in the system.

The generation of motion primitives is not well covered in the literature. It is a problem which remains to be properly solved. However, searching through decision trees is a somewhat general problem; it does not simply apply to this instance of searching for appropriate motion primitives. There are a great many techniques of decision tree traversal, such as $\alpha - \beta$ pruning [22], Beam search [36] and Hill climbing [19] among many others. The methods employed in [25] are a simple backtracking best-first search and a modified best-first search employing an energy heuristic. This is an appropriate choice, since the objective of the algorithm is to find a feasible solution subject to some constraints on energy. The three graph traversal algorithms listed and most others look for some form of optimum solution, which is not well defined in this case. A different type of search may be appropriate based upon some heuristic formulation of an optimum value, however, such a heuristic can trivially inform the best-first search.

2.6 Conclusion

2.6.1 State of the art

For the past decade, virtual constraint based controllers have been used to generate orbitally exponentially stable motions due to the simplification of the dynamics afforded by the reduction of the potentially high-dimensional, nonsmooth, nonconvex and nonlinear relations to a single nonlinear equation in the phase variable, along with the partial closed-form solution available in many cases. These controllers have been proven to be effective in domains of underactuated systems that previously were inaccessible due to the computational difficulty of analysing the system dynamics.

A key focus area of these advancements has been in developing underactuated dynamic walking robots of various complexity, from simple compass-gait walkers to higher degree of freedom robots such as the Rabbit 7-DOF walker. These systems often pose a more significant challenge due to the necessity of modelling the hybrid zero dynamics, imposed by the impact conditions. There are many examples of hybrid zero dynamics based controllers generating robust and efficient walking gaits.

Recent contributions by Mancester et al have been to introduce path planning to hybrid zero dynamics based walking robots, such that the robots may be able to adjust their behaviour ahead of time based upon the terrain. This extends the concepts of virtual constraints generating periodic motions to allow for each footstep to be characterised by a possibly unique constraint. The use of virtual constraints avails to motion planning the same benefits as control design; a partial closed-from solution of the zero dynamics and a means of ordering sets of constraints. This allows for very fast online selection of motion primitives which has been shown to facilitate locomotion over uneven terrain.

2.6.2 Current gaps in research

The technique introduced by Manchester et al in [25] is dependent upon the generation of a library of motion primitives along with an algorithm to select the "best" primitive based upon data of the terrain ahead. Therefore, the best use of this technique will be one which presents some kind of optimal set of motion primitives to be selected, and at each footstep, chooses the most favourable primitive based upon some metric.

The current method employed to generate the library of motion primitives is largely manual, with no guarantee of closeness to an optimal set. This presents two problems; the creation of motion primitives for high-DOF systems is not feasible under this method, and the set of primitives may not present coverage over the configuration space sufficiently to provide a motion primitive for every possibility which bears sufficient closeness to the optimum path.

The current algorithms suggested to select motion primitives are not based upon a true notion of optimality. A best-first search algorithm is suggested which will find a feasible path over the horizon in which it operates, if one exits, but it may do so in a manner which is not energy efficient and may choose footsteps which result in being unable to pass terrain which would have otherwise been passable. A simple energy heuristic involving looking at the highest point in a receding horizon is also suggested in [25]. This heuristic has proven to provide better outcomes in adding energy to the system pre-emptively when an increase in height is ahead, but offers no guarantee of near-optimality, particularly on more varied terrain.

2.6.3 Contribution of this thesis work

This thesis work extends the path planning approach raised in [25] by implementing a method by which the virtual constraint library can be automatically generated. This is scalable to high dimensionality and provides a measure of optimal coverage within given ranges. A heuristic search using a tree similar to that used in the best-first search algorithm from the previous work is also proposed, which more intelligently chooses a primitive based upon the upcoming terrain. This achieves a greater energy efficiency by utilising the gravitational potential energy released when travelling downhill and increases the traversability of rough terrain by ensuring the robot has sufficient kinetic energy when attempting to climb uphill slopes. The algorithm also improves upon the previous work by involving fewer explorations into infeasible paths of the decision tree.

The efficacy of the extensions to Manchester et al's work is demonstrated by validation with an experimental set-up involving a compass-gait walker attached through a boom to a central hub for stability.

CHAPTER 3 Technical Background

3.1 Full actuation and underactuation

Consider a general system with nonlinear time-varying dynamics:

$$\ddot{q}(t) = f\left(\dot{q}(t), q(t), t, u(t)\right) \tag{3.1}$$

where $q(t)$ is a vector of generalised coordinates, $\dot{q}(t)$ is the vector of velocities of those coordinates and $\ddot{q}(t)$ is the vector of accelerations, each of size n, and u is the vector of control inputs of size m.

If, as is typically the case in mechanical systems, the acceleration of the generalised coordinates is linear in the control input, this can be expressed as:

$$\ddot{q}(t) = f_1\left(\dot{q}(t), q(t), t\right) + f_2\left(\dot{q}(t), q(t), t\right) u(t) \tag{3.2}$$

For this class of mechanical systems, we say that the system is considered to be *fully actuated* if and only if $\text{rank}(f_2) = \dim(q)$. If the system is not fully actuated, it is considered to be *underactuated.* If f_2 has zero rank, or if u is empty, then the system is considered to be *unactuated.* Note that whether or not the system is fully actuated is possibly both time and state-dependent.

This mathematical formulation may be interpreted as the following statement:

A system is fully actuated if and only if the accelerations of the generalised coordinates of the system are able to be arbitrarily set through the application of control.

Note that this is an idealised statement; it ignores the necessity to accurately model f_1 and f_2, and physical limits such as torque limits in motors, which by very nature disallow controls to arbitrarily affect systems.

The underactuated systems to which the planning algorithms of this thesis apply, i.e. underactuated dynamic walkers, can be understood to be systems which have at least one less control input than the number of generalised coordinates. Thus even if f_2 has full rank, its rank will still be at most $\dim(q)$ - 1. It should be noted that systems which have actuators on all coordinates can still act as underactuated systems under certain conditions, since f_2 is state-dependent.

3.2 Hybrid Zero Dynamics

Hybrid zero dynamics is the term used to describe the dynamics of the phase variable of an underactuated walker which is subject to virtual constraints. The hybrid zero dynamics of a system are derived from the full state dynamics of continuous phase of the system, the impact map and the virtual constraint.

3.2.1 Dynamics of a general underactuated walker

The general equation of motion of a nonlinear time-invariant physical system may be written as:

$$M\left(q(t)\right)\ddot{q}(t) + C\left(q(t), \dot{q}(t)\right)\dot{q}(t) + G\left(q(t)\right) = B\left(q(t)\right)u(t) \tag{3.3}$$

where $M\left(q(t)\right)$ is the matrix of inertial terms, $C\left(q(t), \dot{q}(t)\right)$ is the matrix of Coriolis and centrifugal terms, $G\left(q(t)\right)$ is the gradient of the potential field and $B\left(q(t)\right)$ is some matrix which specifies the affect of control inputs $u(t)$.

As in [21], we assume that the collisions at each footstep are purely inelastic and the impact map has the following form:

$$q\left(t^+\right) = Rq\left(t^-\right) \tag{3.4}$$

$$\dot{q}\left(t^+\right) = R\Delta\left(q\left(t^-\right)\right)\dot{q}\left(t^-\right) \tag{3.5}$$

For the underactuated walkers considered in this thesis, R represents a relabelling of coordinates. An example is shown of such relabelling for the simple compass gait walker in Figure 3.1.

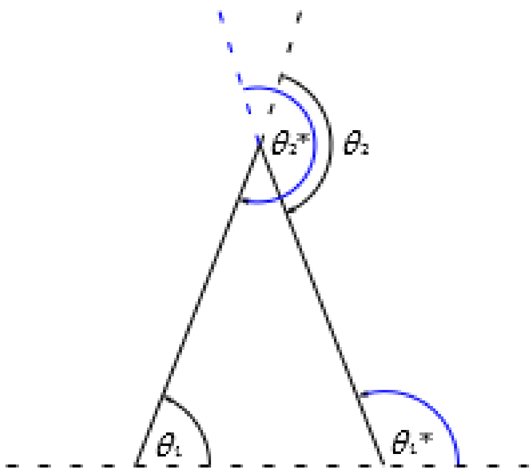

Figure 3.1: Change of coordinates at impact

3.2.2 Application of virtual constraints and zero dynamics

The application of virtual holonomic constraints is a method by which a system's generalised coordinates may be synchronised to a single coordinate, called the *phase variable*, θ. In order for this to be sensible, we assume that θ is increasing over some interval $[\theta_0, \theta_f]$. Under that assumption, we may construct functions for all of the generalised coordinates of the form:

$$q_i(t) = \phi\left(\theta(t)\right), \quad i = 1, 2, \dots, n \tag{3.6}$$

Using trivial differentiation rules, we obtain

$$\begin{aligned} \dot{q}_i(t) &= \frac{\partial \phi\left(\theta(t)\right)}{\partial \theta}\dot{\theta}, \quad i = 1, 2, \dots, n & (3.7) \\ \ddot{q}_i(t) &= \frac{\partial^2 \phi\left(\theta(t)\right)}{\partial \theta^2}\dot{\theta}^2 + \frac{\partial \phi\left(\theta(t)\right)}{\partial \theta}\ddot{\theta}, \quad i = 1, 2, \dots, n & (3.8) \end{aligned}$$

We define the following vector functions:

$$\begin{aligned} \Phi\left(\theta\right) &= \left[\phi_1\left(\theta\right), \phi_2\left(\theta\right), \dots, \phi_n\left(\theta\right)\right]^T & (3.9) \\ \Phi'\left(\theta\right) &= \left[\frac{\partial \phi_1\left(\theta\right)}{\partial \theta}, \frac{\partial \phi_2\left(\theta\right)}{\partial \theta}, \dots, \frac{\partial \phi_n\left(\theta\right)}{\partial \theta}\right]^T & (3.10) \\ \Phi''\left(\theta\right) &= \left[\frac{\partial^2 \phi_1\left(\theta\right)}{\partial \theta^2}, \frac{\partial^2 \phi_2\left(\theta\right)}{\partial \theta^2}, \dots, \frac{\partial^2 \phi_n\left(\theta\right)}{\partial \theta^2}\right]^T & (3.11) \end{aligned}$$

Under the assumption perfect regulation of the virtual constraints, thus the above relations holding, we can evaluate the zero dynamics by simple substitution into Equation 3.3:

$$M\left(\Phi(\theta)\right)\left[\Phi'(\theta)\ddot{\theta} + \Phi''\dot{\theta}^2\right] + C\left(\Phi(\theta), \Phi'(\theta)\dot{\theta}\right)\Phi'(\theta)\dot{\theta} + G\left(\Phi(\theta)\right) = B\left(\Phi(\theta)\right)u_c \tag{3.12}$$

where u_c is the control which achieves perfect regulation of the constraints. This may be rearranged into a more convenient form:

$$\alpha(\theta)\ddot{\theta} + \beta(\theta)\dot{\theta}^2 + \gamma(\theta) = 0 \tag{3.13}$$

where, if we denote $B^{\perp}(q)$ as a row vector which satisfies $B^{\perp}(q)B(q)u_c = 0$,

$$\begin{aligned} \alpha(\theta) &= B^{\perp}\left(\Phi(\theta)\right)M\left(\Phi(\theta)\right)\Phi'(\theta) \\ \beta(\theta) &= B^{\perp}\left(\Phi(\theta)\right)\left(M\left(\Phi(\theta)\right)\Phi''(\theta) + C\left(\Phi(\theta), \Phi'(\theta)\right)\Phi'(\theta)\right) \\ \gamma(\theta) &= B^{\perp}\left(\Phi(\theta)\right)G\left(\Phi(\theta)\right) & (3.14) \end{aligned}$$

3.2.3 Partial closed-form solutions for velocity and energy

One of the useful properties of virtual constraints is the ability that they lend the designer of a motion planner to precompute a partial closed-form solution for velocity and energy. The solution is partial in that we obtain an expression for $\dot{\theta}^2$ in terms of θ, rather than θ in terms of time. Under the assumption that θ is monotonic, it can be used as a new dependent variable:

$$\begin{aligned} \frac{d}{d\theta}\left[\dot{\theta}\left(t(\theta)\right)^2\right] &= \frac{d}{dt}\frac{dt}{d\theta}\left[\frac{d\theta}{dt}\left(t(\theta)\right)^2\right] \\ &= 2\ddot{\theta}\left(t(\theta)\right) \end{aligned} \tag{3.15}$$

Substituting Equation 3.13 into this expression, we arrive at a first-order ODE in $\dot{\theta}(\theta)^2$:

$$\frac{d}{d\theta}\dot{\theta}(\theta)^2 = -2\frac{\beta(\theta)}{\alpha(\theta)}\dot{\theta}(\theta)^2 - 2\frac{\gamma(\theta)}{\alpha(\theta)} \tag{3.16}$$

If we assume, as in [25], that for all $\theta \in [\theta_0, \theta_f]$, we have local instantaneous controllability, i.e. $\alpha(\theta) \neq 0$, then we may solve Equation 3.16 numerically over $[\theta_0, \theta_f]$, which yields an affine solution, i.e.

$$\dot{\theta}(\theta)^2 = \Gamma(\theta, \theta_0)\dot{\theta}_0^2 + \Psi(\theta, \theta_0) \tag{3.17}$$

From this, we may also derive the total mechanical energy of the system. For general systems with dynamics as expressed in Equation 3.3, the mechanical energy has the form:

$$H\left(q, \dot{q}\right) = \dot{q}^T M(q)\dot{q} + V(q) \tag{3.18}$$

Under perfectly regulated virtual constraints, this reduces to:

$$\bar{H}\left(\theta, \dot{\theta}\right) := \Upsilon(\theta)\dot{\theta}^2 + \Xi(\theta) \tag{3.19}$$

where

$$\begin{aligned} \Upsilon(\theta) &= \Phi'(\theta)^T M\left(\Phi(\theta)\right)\Phi'(\theta) \\ \Xi(\theta) &= V\left(\Phi(\theta)\right) \end{aligned}$$

Since this is affine in $\dot{\theta}^2$ for a given θ, we may trivially calculate the closed form:

$$H\left(\theta, \dot{\theta}\right) = \Upsilon(\theta)\Gamma\left(\theta, \theta_0\right)\dot{\theta}_0^2 + \Upsilon(\theta)\Psi\left(\theta, \theta_0\right) + \Xi(\theta) \tag{3.20}$$

3.3 Bézier curves as virtual constraints

Bézier curves provide a way to produce families of curves for particular start and end heights and are sparsely identified by only $n + 1$ points, where n is the degree of the curve. These points provide an intuitive way of defining the curve, in contrast with polynomial coefficients.

Theoretically, these curves need only be defined from the start to the endpoint of the continuous-phase which they specify. However, since the curve provides a virtual constraint to be enforced by a controller, it is necessary to define the curve over the full range of possible motion, here considered to be $\theta_1 \in [0, \pi]$, else it is possible for the walker to enter a region where the control signal is undefined. If we assume that the overshoot past the desired endpoint is small, then the shape of the curve should be flat outside the defined region. That is, when we leave the target region, we wish to set θ_2 to the closest defined θ_2.

For the compass-gait walker, a general Bézier curve is defined by the following parametric equation.

$$\begin{bmatrix}\theta_1 \\ \theta_2\end{bmatrix} = \sum_{i=0}^{n} \binom{n}{i} (1-t)^{n-i}\, t^i \begin{bmatrix}\theta_{1_i} \\ \theta_{2_i}\end{bmatrix} \tag{3.21}$$

Since this equation is not monotonic in θ_1, it is not a convenient expression. Therefore, we build families of Bézier curves with the following formulation:

$$t = \frac{\theta_1 - \theta_{1_0}}{\theta_{1_n} - \theta_{1_0}} \tag{3.22}$$

$$\theta_2 = \sum_{i=0}^{n} \binom{n}{i} (1-t)^{n-i}\, t^i \theta_{2_i} \tag{3.23}$$

This is expressible explicitly as

$$\theta_2 = \frac{1}{\left(\theta_{1_n} - \theta_{1_0}\right)^n} \sum_{i=0}^{n} \binom{n}{i} \left(\theta_{1_n} - \theta_1\right)^{n-i} \left(\theta_1 - \theta_{1_0}\right)^i \theta_{2_i} \tag{3.24}$$

This formulation removes our ability to arbitrarily define the control points $(\theta_{1_i}, \theta_{2_i})$, other than the endpoints. That is, this formulation produces curves of the form given in Equation 3.21 with

$$\theta_{1_i} = \frac{i}{n}\left(\theta_{1_n} - \theta_{1_0}\right) + \theta_{1_0} \quad \forall \quad i \in [1,\ n-1] \tag{3.25}$$

Since we can only define one of the two variables in each control point, to yield arbitrary curves of order n, before achievable with $n - 1$ free control points (i.e. non-endpoint control points), now requires $2n-2$ such points. Since it is desirable to have the ability to set the gradient of approach to the control points independently of the shape of the polynomial (achievable using a general cubic Bézier curve) we use a quintic curve in the new formulation. This requires six control points.

CHAPTER 4 Virtual Constraint Library

4.1 Data stored

4.2 Graphical interface for constraint design

Details on the GUI.

4.3 Automatic generation of constraints

I need to get this working very soon

4.4 Ordering methods

May be very relevant.

CHAPTER 5 Algorithm Design

Algorithm designs need to be designed!

CHAPTER 6 Verification

6.1 Simulation

1. Compass gait 2. 5-link Design of simulation, limitations, outputs

6.2 Experiment

Design of experiment; what was assessed.

CHAPTER 7 Results

Results from simulation and experiment. Hopefully good.

CHAPTER 8 Discussion

8.1 Outcomes of study

Maybe something clever can go here

8.2 Advantages of virtual constraints method

And here

8.3 Limitations of virtual constraints method

Maybe not too much here

8.4 Future work

Go wild here.

Appendices

APPENDIX A Worked example: compass-gait walker

The compass-gait walker is useful as an instructive example, since the derivation of relevant dynamics and application of virtual constraints is relatively simple, but it is straightforward to understand how the principle may be extended to more complex systems. This section details the application of the principles discussed in this thesis to the simple case of the saggital-plane compass-gait walker.

A.1 Continuous phase dynamics

Forward Kinematics

$$
\begin{aligned}
x_1 &= \frac{l_1}{2}\cos\theta_1 \\
\dot{x_1} &= -\frac{l_1}{2}\sin(\theta_1)\dot{\theta}_1 \\
y_1 &= \frac{l_1}{2}\sin\theta_1 \\
\dot{y_1} &= \frac{l_1}{2}\cos(\theta_1)\dot{\theta}_1 \\
x_2 &= l_1\cos\theta_1 + \frac{l_2}{2}\cos(\theta_1+\theta_2) \\
\dot{x_2} &= -l_1\sin(\theta_1)\dot{\theta}_1 - \frac{l_2}{2}\sin(\theta_1+\theta_2)(\dot{\theta}_1+\dot{\theta}_2) && \text{(A.1)} \\
y_2 &= l_1\sin\theta_1 + \frac{l_2}{2}\sin(\theta_1+\theta_2) \\
\dot{y_2} &= l_1\cos\theta_1\dot{\theta}_1 + \frac{l_2}{2}\cos(\theta_1+\theta_2)(\dot{\theta}_1+\dot{\theta}_2) && \text{(A.2)}
\end{aligned}
$$

Lagrangian Dynamics

In order to produce the dynamical equations for the two-link manipulator, we must compute the Lagrangian:

$$L = K - P$$
$$T_i = \frac{\partial}{\partial t}\left(\frac{\partial L}{\partial \dot{\theta}_i}\right) - \frac{\partial L}{\partial \theta_i} \tag{A.3}$$

The kinetic energy of the manipulator is given by the summation of the pure rotational kinetic energy of the first link about the origin and the rotational kinetic energy of the second link about its centre and the linear KE of its centre of mass.

$$K = \frac{1}{2}I_1\dot{\theta}_1^2 + \frac{1}{2}I_2(\dot{\theta}_1 + \dot{\theta}_2)^2 + \frac{1}{2}m_2V_2^2 \tag{A.4}$$

From equations A.1 and A.2, we have expressions for the components of V_2:

$$\begin{aligned} V_2^2 = &\left[l_1^2 s_1^2 \dot{\theta}_1^2 + l_1 l_2 s_1 s_{12}\dot{\theta}_1(\dot{\theta}_1 + \dot{\theta}_2) + \frac{l_2^2}{4}s_{12}^2(\dot{\theta}_1 + \dot{\theta}_2)^2\right] \\ &+ \left[l_1^2 c_1^2 \dot{\theta}_1^2 + l_1 l_2 c_1 c_{12}\dot{\theta}_1(\dot{\theta}_1 + \dot{\theta}_2) + \frac{l_2^2}{4}c_{12}^2(\dot{\theta}_1 + \dot{\theta}_2)^2\right] \end{aligned}$$

Combining terms and exploiting the trigonometric identities $\sin^2 x + \cos^2 x = 1$ and $\cos(x-y) = \cos x \cos y + \sin x \sin y$:

$$V_2^2 = l_1^2\dot{\theta}_1^2 + l_1 l_2 c_2 \dot{\theta}_1(\dot{\theta}_1 + \dot{\theta}_2) + \frac{1}{4}l_2^2(\dot{\theta}_1 + \dot{\theta}_2)^2 \tag{A.5}$$

Substituting equation A.5 into equation A.4:

$$\begin{aligned} K =&\frac{1}{2}I_1\dot{\theta}_1^2 + \frac{1}{2}I_2\left(\dot{\theta}_1 + \dot{\theta}_2\right)^2 \\ &+ \frac{1}{2}m_2\left[\dot{\theta}_1^2\left(l_1^2 + \frac{1}{4}l_2^2 + l_1 l_2 c_2\right) + \dot{\theta}_2^2\left(\frac{1}{4}l_2^2\right) + \dot{\theta}_1\dot{\theta}_2\left(\frac{1}{2}l_2^2 + l_1 l_2 c_2\right)\right] \\ P =&\frac{1}{2}l_1 s_1 m_1 g + \left(l_1 s_1 + \frac{1}{2}l_2 s_{12}\right)m_2 g \\ L =&\frac{1}{2}\dot{\theta}_1^2\left(I_1 + I_2 + m_2\left(l_1^2 + \frac{1}{4}l_2 + l_1 l_2 c_2\right)\right) + \frac{1}{2}\dot{\theta}_2^2\left(I_2 + \frac{1}{4}m_2 l_2^2\right) \\ &+ \dot{\theta}_1\dot{\theta}_2\left(I_2 + \frac{1}{2}m_2\left(\frac{1}{2}l_2^2 + l_1 l_2 c_2\right)\right) - \frac{1}{2}l_1 s_1 m_1 g - \left(l_1 s_1 + \frac{1}{2}l_2 s_{12}\right)m_2 g \end{aligned} \tag{A.6}$$

Now, we can determine equations for the torques using equations A.3 and A.6.

$$\begin{aligned} \frac{\partial L}{\partial \dot{\theta}_1} =&\dot{\theta}_1\left(I_1 + I_2 + m_2\left(l_1^2 + \frac{1}{4}l_2^2 + l_1 l_2 c_2\right)\right) + \dot{\theta}_2\left(I_2 + \frac{1}{2}m_2\left(\frac{1}{2}l_2^2 + l_1 l_2 c_2\right)\right) \\ \frac{\partial}{\partial t}\left(\frac{\partial L}{\partial \dot{\theta}_1}\right) =&\ddot{\theta}_1\left(I_1 + I_2 + m_2\left(l_1^2 + \frac{1}{4}l_2^2 + l_1 l_2 c_2\right)\right) - \dot{\theta}_1\dot{\theta}_2\left(m_2 l_1 l_2 s_2\right) \\ &+ \ddot{\theta}_2\left(I_2 + \frac{1}{2}m_2\left(\frac{1}{2}l_2^2 + l_1 l_2 c_2\right)\right) - \dot{\theta}_2^2\left(\frac{1}{2}m_2 l_1 l_2 s_2\right) \\ \frac{\partial L}{\partial \theta_1} =& -\frac{1}{2}l_1 c_1 m_1 g - m_2 g\left(l_1 c_1 + \frac{1}{2}l_2 c_{12}\right) \end{aligned}$$

Adding the components together, we get:

$$\begin{aligned}
T_1 =& \ddot{\theta}_1 \left(I_1 + I_2 + m_2 \left(l_1^2 + \frac{1}{4} l_2^2 + l_1 l_2 c_2 \right) \right) - \dot{\theta}_1 \dot{\theta}_2 \left(m_2 l_1 l_2 s_2 \right) \\
&+ \ddot{\theta}_2 \left(I_2 + \frac{1}{2} m_2 \left(\frac{1}{2} l_2^2 + l_1 l_2 c_2 \right) \right) - \dot{\theta}_2^2 \left(\frac{1}{2} m_2 l_1 l_2 s_2 \right) \\
&+ \frac{1}{2} l_1 c_1 m_1 g + m_2 g \left(l_1 c_1 + \frac{1}{2} l_2 c_{12} \right)
\end{aligned} \tag{A.7}$$

Likewise for T_2:

$$\begin{aligned}
\frac{\partial L}{\partial \dot{\theta}_2} =& \dot{\theta}_2 \left(I_2 + \frac{1}{4} m_2 l_2^2 \right) + \dot{\theta}_1 \left(I_2 + \frac{1}{2} m_2 \left(\frac{1}{2} l_2^2 + l_1 l_2 c_2 \right) \right) \\
\frac{\partial}{\partial t} \left(\frac{\partial L}{\partial \dot{\theta}_2} \right) =& \ddot{\theta}_2 \left(I_2 + \frac{1}{4} m_2 l_2^2 \right) + \ddot{\theta}_1 \left(I_2 + \frac{1}{2} m_2 \left(\frac{1}{2} l_2^2 + l_1 l_2 c_2 \right) \right) - \dot{\theta}_1 \dot{\theta}_2 \left(\frac{1}{2} m_2 l_1 l_2 s_2 \right) \\
\frac{\partial L}{\partial \theta_2} =& - \dot{\theta}_1^2 \left(\frac{1}{2} m_2 l_1 l_2 s_2 \right) - \dot{\theta}_1 \dot{\theta}_2 \left(\frac{1}{2} m_2 l_1 l_2 s_2 \right) - \frac{1}{2} m_2 g l_2 c_{12}
\end{aligned}$$

Adding components as for T_1:

$$\begin{aligned}
T_2 =& \ddot{\theta}_2 \left(I_2 + \frac{1}{4} m_2 l_2^2 \right) + \ddot{\theta}_1 \left(I_2 + \frac{1}{2} m_2 \left(\frac{1}{2} l_2^2 + l_1 l_2 c_2 \right) \right) \\
&+ \dot{\theta}_1^2 \left(\frac{1}{2} m_2 l_1 l_2 s_2 \right) + \frac{1}{2} m_2 g l_2 c_{12}
\end{aligned} \tag{A.8}$$

Thus we have an equation of the form

$$M\left(q(t)\right) \ddot{q}(t) + C\left(q(t), \dot{q}(t)\right) \dot{q}(t) + G\left(q(t)\right) = B\left(q(t)\right) u(t) \tag{A.9}$$

where

$$\begin{aligned}
M\left(q(t)\right) &= \begin{bmatrix} I_1 + I_2 + m_2 \left(l_1^2 + \frac{1}{4} l_2^2 + l_1 l_2 \cos q_2 \right) & I_2 + \frac{1}{2} m_2 \left(\frac{1}{2} l_2^2 + l_1 l_2 \cos q_2 \right) \\ I_2 + \frac{1}{2} m_2 \left(\frac{1}{2} l_2^2 + l_1 l_2 \cos q_2 \right) & I_2 + \frac{1}{4} m_2 l_2^2 \end{bmatrix} \\
C\left(q(t), \dot{q}(t)\right) &= \begin{bmatrix} -m_2 l_1 l_2 \sin(q_2) \dot{q}_2 & -\frac{1}{2} m_2 l_1 l_2 \sin(q_2) \dot{q}_2 \\ \frac{1}{2} m_2 l_1 l_2 \sin(q_2) \dot{q}_1 & 0 \end{bmatrix} \\
G\left(q(t)\right) &= \begin{bmatrix} \frac{1}{2} l_1 \cos q_1 m_1 g + m_2 g \left(l_1 \cos q_1 + \frac{1}{2} l_2 \cos\left(q_1 + q_2\right) \right) \\ \frac{1}{2} m_2 g l_2 \cos\left(q_1 + q_2\right) \end{bmatrix} \\
B\left(q(t)\right) &= \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \ u(t) = T_2, \ q(t) = \begin{bmatrix} q_1(t) \\ q_2(t) \end{bmatrix}
\end{aligned}$$

A.2 Impact dynamics

Need to write this up. Mostly following Westervelt's pg 422 method.

A.3 Application of virtual constraints

In the case of the compass-gait walker, of course, since there is only one variable other than the phase variable θ, these functions have only two elements each, one of which is trivial. Thus:

$$\begin{aligned}
\Phi(\theta) &= \begin{bmatrix} \theta \\ \phi(\theta) \end{bmatrix} \\
\Phi'(\theta) &= \begin{bmatrix} 1 \\ \frac{\partial \phi(\theta)}{\partial \theta} \end{bmatrix} \\
\Phi''(\theta) &= \begin{bmatrix} 0 \\ \frac{\partial^2 \phi(\theta)}{\partial \theta^2} \end{bmatrix}
\end{aligned}$$

Now, we have from Equation 3.24 (slightly adapted to suit new coordinate naming):

$$\phi(\theta) = \frac{1}{(\theta_f - \theta_0)^n} \sum_{i=0}^{n} \binom{n}{i} (\theta_f - \theta)^{n-i} (\theta - \theta_0)^i \vartheta_i \tag{A.10}$$

The derivative is

$$\begin{aligned}
\frac{\partial \phi}{\partial \theta} = \frac{1}{(\theta_f - \theta_0)^n} \Bigg(& n \left((\theta - \theta_0)^{n-1} \vartheta_n - (\theta_f - \theta)^{n-1} \vartheta_0 \right) \\
& + \sum_{i=1}^{n-1} \binom{n}{i} \left(i (\theta_f - \theta)^{n-i} (\theta - \theta_0)^{i-1} - (n-i)(\theta_f - \theta)^{n-i-1} (\theta - \theta_0)^i \right) \vartheta_i \Bigg)
\end{aligned} \tag{A.11}$$

The second derivative is

$$\begin{aligned}
\frac{\partial^2 \phi}{\partial \theta^2} = \frac{1}{(\theta_f - \theta_0)^n} \Bigg(& n(n-1) \Big[(\theta_f - \theta)^{n-2} \vartheta_0 + (\theta - \theta_0)^{n-2} \vartheta_n + (n-2) \Big((\theta - \theta_0)(\theta_f - \theta)^{n-3} \vartheta_1 + \\
& (\theta_f - \theta)(\theta - \theta_0)^{n-3} \vartheta_{n-1} \Big) - 2 \left((\theta_f - \theta)^{n-2} \vartheta_1 + (\theta - \theta_0)^{n-2} \vartheta_{n-1} \right) \Big] \\
& + \sum_{i=2}^{n-2} \binom{n}{i} \Big(i(i-1)(\theta_f - \theta)^{n-i} (\theta - \theta_0)^{i-2} - 2i(n-i)(\theta_f - \theta)^{n-i-1} (\theta - \theta_0)^{i-1} \\
& + (n-i-1)(n-i)(\theta_f - \theta)^{n-i-2} (\theta - \theta_0)^i \Big) \vartheta_i \Bigg)
\end{aligned} \tag{A.12}$$

Let us choose the simplest non-zero $B^{\perp}$:

$$B^{\perp} = \begin{bmatrix} 1 & 0 \end{bmatrix}$$

We also have the following. Note that for simplicity, $\phi(\theta) = \phi$.

$$\begin{aligned}
M(\Phi(\theta)) &= \begin{bmatrix} I_1 + I_2 + m_2 \left(l_1^2 + \frac{1}{4} l_2^2 + l_1 l_2 \cos\phi \right) & I_2 + \frac{1}{2} m_2 \left(\frac{1}{2} l_2^2 + l_1 l_2 \cos\phi \right) \\ I_2 + \frac{1}{2} m_2 \left(\frac{1}{2} l_2^2 + l_1 l_2 \cos\phi \right) & I_2 + \frac{1}{4} m_2 l_2^2 \end{bmatrix} \\
C(\Phi(\theta), \Phi'(\theta)) &= \sin(\phi) \begin{bmatrix} -m_2 l_1 l_2 \frac{\partial \phi}{\partial \theta} & -\frac{1}{2} m_2 l_1 l_2 \frac{\partial \phi}{\partial \theta} \\ \frac{1}{2} m_2 l_1 l_2 & 0 \end{bmatrix} \\
G(\Phi(\theta)) &= \begin{bmatrix} \frac{1}{2} l_1 m_1 g \cos\theta + m_2 g \left(l_1 \cos\theta + \frac{1}{2} l_2 \cos(\theta + \phi) \right) \\ \frac{1}{2} m_2 g l_2 \cos(\theta + \phi) \end{bmatrix}
\end{aligned}$$

A.4 Optimisation of single constraint

Worked example of optimisation of single constraint (simple since there are only 4 points which form the decision variables.

A.5 Optimisation of library of constraints

Worked example of how the optimisation of the library of motion primitives was developed for the compass gait. This is simple, since all that is required is to cover horizontal and vertical steps in contrast to more complex configurations with the 5-link.

APPENDIX B Numerical integration method

As in [25], we produce functions of the form

$$\alpha\left(\theta\right)\ddot{\theta}(t)+\beta\left(\theta\right)\dot{\theta}(t)^{2}+\gamma\left(\theta\right)=0 \tag{B.1}$$

with

$$\begin{aligned}
\alpha(\theta) &= B^{\perp}\left(\Phi(\theta)\right)M\left(\Phi(\theta)\right)\Phi'(\theta)\\
\beta(\theta) &= B^{\perp}\left(\Phi(\theta)\right)\left(M\left(\Phi(\theta)\right)\Phi''(\theta)+C\left(\Phi(\theta),\Phi'(\theta)\right)\Phi'(\theta)\right)\\
\gamma(\theta) &= B^{\perp}\left(\Phi(\theta)\right)G\left(\Phi(\theta)\right)
\end{aligned}$$

Then we get the differential equation:

$$\frac{d}{d\theta}\dot{\theta}\left(\theta\right)^{2}=-2\frac{\beta\left(\theta\right)}{\alpha\left(\theta\right)}\dot{\theta}\left(\theta\right)^{2}-2\frac{\gamma\left(\theta\right)}{\alpha\left(\theta\right)} \tag{B.2}$$

Solving this over any interval $\theta\in[\theta_0,\theta_f]$ yields

$$\dot{\theta}\left(\theta\right)^{2}=\Gamma\left(\theta,\theta_0\right)\dot{\theta}^{2}+\Psi\left(\theta,\theta_0\right) \tag{B.3}$$

This solution is achieved by using the general method for first-order linear ODEs with varying coefficients, i.e. given

$$y'(x)+f(x)y(x)=g(x)$$

the solution is

$$y=e^{-\int f(x)dx}\left(\int g(x)e^{\int f(x)dx}dx+\kappa\right)$$

Thus we have

$$\begin{aligned}
\Gamma(\theta) &= e^{-\int_{\theta_0}^{\theta}f(x)dx} && \text{(B.4)}\\
\Psi(\theta) &= e^{-\int_{\theta_0}^{\theta}f(x)dx}\int_{\theta_0}^{\theta}g(x)e^{\int_{\theta_0}^{\theta}f(x)dx} && \text{(B.5)}\\
f(x) &= 2\frac{\beta(x)}{\alpha(x)}\\
g(x) &= -2\frac{\gamma(x)}{\alpha(x)}
\end{aligned}$$

Bibliography

[1] F. Asano and Zhi-Wei Luo. Dynamic analyses of underactuated virtual passive dynamic walking. In *Robotics and Automation, 2007 IEEE International Conference on*, pages 3210–3217, April 2007.

[2] Karl Johan Åström and Tore Hägglund. The future of PID control. *Control Engineering Practice*, 9(11):1163–1175, 2001.

[3] Valérie Boor, Mark H Overmars, and A Frank van der Stappen. The gaussian sampling strategy for probabilistic roadmap planners. In *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on*, volume 2, pages 1018–1023. IEEE, 1999.

[4] R.A. Brooks. A robust layered control system for a mobile robot. *Robotics and Automation, IEEE Journal of*, 2(1):14–23, Mar 1986.

[5] Katie Byl and Russ Tedrake. Approximate optimal control of the compass gait on rough terrain. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pages 1258–1263. IEEE, 2008.

[6] Carlos Canudas-de Wit. On the concept of virtual constraints as a tool for walking robot control and balancing. *Annual Reviews in Control*, 28(2):157–166, 2004.

[7] J. Chestnutt, P. Michel, J. Kuffner, and T. Kanade. Locomotion among dynamic obstacles for the Honda ASIMO. In *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, pages 2572–2573, Oct 2007.

[8] Joel Chestnutt, Manfred Lau, German Cheung, James Kuffner, Jessica Hodgins, and Takeo Kanade. Footstep planning for the Honda ASIMO humanoid. In *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, pages 629–634. IEEE, 2005.

[9] Benjamin J Cohen, Gokul Subramanian, Sachin Chitta, and Maxim Likhachev. Planning for manipulation with adaptive motion primitives. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 5478–5485. IEEE, 2011.

[10] Steven H Collins and Andy Ruina. A bipedal walking robot with efficient and human-like gait. In *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, pages 1983–1988. IEEE, 2005.

[11] Steven H Collins, Martijn Wisse, and Andy Ruina. A three-dimensional passive-dynamic walking robot with two legs and knees. *The International Journal of Robotics Research*, 20(7):607–615, 2001.

[12] M. Franken, G. van Oort, and S. Stramigioli. Analysis and simulation of fully ankle actuated planar bipedal robots. In *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pages 634–639, Sept 2008.

[13] Emilio Frazzoli, Munther A Dahleh, and Eric Feron. Real-time motion planning for agile autonomous vehicles. *Journal of Guidance, Control, and Dynamics*, 25(1):116–129, 2002.

[14] Emilio Frazzoli, Munther A Dahleh, and Eric Feron. Maneuver-based motion planning for nonlinear systems with symmetries. *Robotics, IEEE Transactions on*, 21(6):1077–1091, 2005.

[15] L. Freidovich, A. Robertsson, A. Shiriaev, and R. Johansson. Periodic motions of the Pendubot via virtual holonomic constraints: Theory and experiments. *Automatica*, 44(3):785 – 791, 2008.

[16] Leonid B Freidovich, Uwe Mettin, Anton S Shiriaev, and Mark W Spong. A passive 2-DOF walker: Hunting for gaits using virtual holonomic constraints. *Robotics, IEEE Transactions on*, 25(5):1202–1208, 2009.

[17] Erann Gat. Integrating planning and reacting in a heterogeneous asynchronous architecture for controlling real-world mobile robots. In *AAAi*, volume 1992, pages 809–815, 1992.

[18] Michael P Georgeff and Amy L Lansky. Reactive reasoning and planning. In *AAAI*, volume 87, pages 677–682, 1987.

[19] Stephen M Goldfeld, Richard E Quandt, and Hale F Trotter. Maximization by quadratic hill-climbing. *Econometrica: Journal of the Econometric Society*, pages 541–551, 1966.

[20] Jesse W Grizzle, Gabriel Abba, and Franck Plestan. Asymptotically stable walking for biped robots: Analysis via systems with impulse effects. *Automatic Control, IEEE Transactions on*, 46(1):51–64, 2001.

[21] Yildirim Hurmuzlu and Dan B Marghitu. Rigid body collisions of planar kinematic chains with multiple contact points. *The International Journal of Robotics Research*, 13(1):82–92, 1994.

[22] Donald E Knuth and Ronald W Moore. An analysis of alpha-beta pruning. *Artificial intelligence*, 6(4):293–326, 1976.

[23] Steven M LaValle and James J Kuffner. Randomized kinodynamic planning. *The International Journal of Robotics Research*, 20(5):378–400, 2001.

[24] Ian R Manchester, Uwe Mettin, Fumiya Iida, and Russ Tedrake. Stable dynamic walking over uneven terrain. *The International Journal of Robotics Research*, 30(3):265–279, 2011.

[25] Ian R. Manchester and Jack Umenberger. Real-time planning with primitives for dynamic walking over uneven terrain. *CoRR*, abs/1310.7062, 2013.

[26] Anne E Martin, David C Post, and James P Schmiedeler. Design and experimental implementation of a hybrid zero dynamics-based controller for planar bipeds with curved feet. *The International Journal of Robotics Research*, page 0278364914522141, 2014.

[27] Anne E Martin and James P Schmiedeler. Predicting healthy human and amputee walking gait using ideas from underactuated robot control.

[28] Tad McGeer. Passive dynamic walking. *The International Journal of Robotics Research*, 9(2):62–82, 1990.

[29] N. J. Nilsson, C. A. Rosen, B. Raphael, G. Forsen, L. Chaitin, and S. Wahlstrom. Application of intelligent automata to reconnaissance. Technical report, Stanford Research Institute, December 1968.

[30] Marc Raibert, Kevin Blankespoor, Gabriel Nelson, Rob Playter, et al. Bigdog, the rough-terrain quadruped robot. In *Proceedings of the 17th World Congress*, pages 10823–10825, 2008.

[31] Daniel E Rivera, Manfred Morari, and Sigurd Skogestad. Internal model control: Pid controller design. *Industrial & engineering chemistry process design and development*, 25(1):252–265, 1986.

[32] M Russell. Odex I: The first functionoid. *Robotics Age*, 5(5):12–18, 1983.

[33] Anton Shiriaev, John W Perram, and Carlos Canudas-de Wit. Constructive tool for orbital stabilization of underactuated nonlinear systems: Virtual constraints approach. *Automatic Control, IEEE Transactions on*, 50(8):1164–1176, 2005.

[34] Alexander Shkolnik, Michael Levashov, Ian R Manchester, and Russ Tedrake. Bounding on rough terrain with the LittleDog robot. *The International Journal of Robotics Research*, 30(2):192–215, 2011.

[35] Koushil Sreenath, Hae-Won Park, Ioannis Poulakakis, and Jessy W Grizzle. A compliant hybrid zero dynamics controller for stable, efficient and fast bipedal walking on MABEL. *The International Journal of Robotics Research*, 30(9):1170–1193, 2011.

[36] Volker Steinbiss, Bach-Hiep Tran, and Hermann Ney. Improvements in beam search. In *ICSLP*, volume 94, pages 2143–2146, 1994.

[37] Richard S Sutton and Andrew G Barto. *Introduction to reinforcement learning*. MIT Press, 1998.

[38] Yuval Tassa, Tom Erez, and Emanuel Todorov. Synthesis and stabilization of complex behaviors through online trajectory optimization. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 4906–4913. IEEE, 2012.

[39] Russ Tedrake, Teresa Weirui Zhang, Ming-fai Fong, and H Sebastian Seung. Actuating a simple 3D passive dynamic walker. In *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, volume 5, pages 4656–4661. IEEE, 2004.

[40] Steven A Vere. Planning in time: Windows and durations for activities and goals. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, (3):246–267, 1983.

[41] Vojtech Vonasek, Martin Saska, Karel Kosnar, and Libor Preucil. Global motion planning for modular robots with local motion primitives. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 2465–2470. IEEE, 2013.

[42] Kenneth J Waldron and Robert B McGhee. The adaptive suspension vehicle. *IEEE Control Systems Magazine*, 6(6):7–12, 1986.

[43] Eric R Westervelt, Jessy W Grizzle, and Daniel E Koditschek. Hybrid zero dynamics of planar biped walkers. *Automatic Control, IEEE Transactions on*, 48(1):42–56, 2003.