Optimizing Footfall Patterns for Gait Transitions

Shuang Su

CMU-CS-17-111
May 2017

Computer Science Department
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee:
Stelian Coros, Chair
Nancy Pollard

Submitted in partial fulfillment of the requirements for the degree of Master of Science

Keywords: footfall pattern, gaits, transition, mixed integer programming, optimization

To all who have helped me attain today.

Abstract

To perform different types of tasks, robots are required to adopt various gaits and smoothly transition between them. With current techniques, it is possible to create full-body robot motions starting from a description of a footfall pattern, as well as high-level specifications such as the desired walking speed. However, the input footfall pattern needs to be hand-designed, which is both tedious and potentially error-prone[1].

The purpose of this study is to develop an automatic approach to generating locomotion gaits for periodic robot motions as well as footfall patterns for transitions. To address this challenge, we propose a mixed-integer quadratic optimization model that generates stable footfall patterns starting from a set of objectives and constraints. We demonstrate the versatility of this model by generating a variety of gaits and transitions for quadruped, hexapod and dodecapod robots, which are tested both in simulation and on physical platforms.

Acknowledgements

I first want to give special thanks to my instructor, Professor Stelian Coros, who has instructed me on this project step by step. From the starting point of building up the interface to designing the algorithms for footfall patterns, from validating the simulation results on a real robot to summarizing the results into a thesis report, Professor Coros has given me numerous guidance, encouragement, and support which gradually build up my research abilities as well as my confidence. I'm encouraged to explore my own ways of solving the problems, but whenever I get stuck, the guidance is always there. I would also thank Professor Nancy Pollard for being my thesis committee on a short request and gives valuable and constructive comments on this work.

In addition, I would like to thank Julian Whiteman for all the brilliant ideas generated for the project, and for all the contributions which make this work turn into a paper submitted to IROS in March. Also special thanks for Professor Coros's and Professor Choset's detailed instructions and contributions to the paper.

Last but not least, thanks Professor Dave A. Eckhardt for all his guidance along my master's career, and Tracy Farbacher for her kindness and patience whenever I go to her for help. Thanks CMU for giving me such a wonderful journey in the garden of robotics and computer science.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Through the process of natural selection, animals have evolved a rich set of gaits which are defined by footfall patterns, to obtain a trade off between agility, robustness, and energy efficiency when moving on arbitrary terrains[14]. Most mammals, for example, can employ a walking, trotting, or galloping gait, depending on the speed they wish to move with. Mother gorilla and other great apes, while usually walking on 4 limbs, needs to balance on 3 when she is holding a baby. Gaits can therefore not only be used to allow animals to move with different speeds, but also give them increased versatility.

Figure1.1: The left image shows mother gorilla walking on three limbs holding a baby while the right one depicts a dodecapod carrying a box, using a rear limb to inspect a pipe.

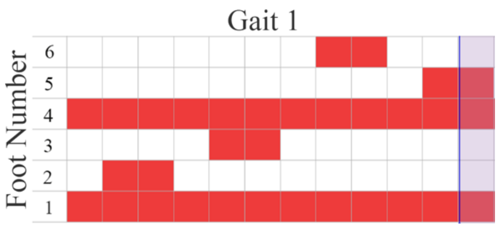
Getting inspirations from animals, modular robots, which is a family of robot systems consisted of inter-connected small units called modules[22], also need to perform various kinds of tasks like manipulation, locomotion, and inspection[19], and therefore need footfall patterns to transition between the gaits of those tasks.

In legged robot motion planning, optimization factors such as stability and energy consumption must also be carefully considered[16]. Current techniques can generate smooth robot motions by optimizing robot joint angles using gradient-based methods[1] given an input footfall pattern. However, the input footfall pattern needs to be hand-designed, which is both tedious and potentially error-prone.

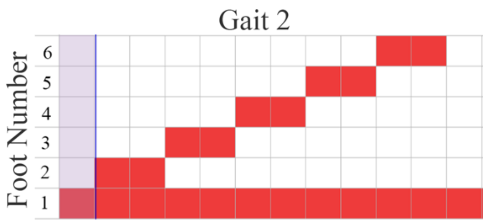
Our research, therefore, is concerned with generating the footfall patterns automatically for transitions between various gaits.

Problem Statement and Objectives

Animal gaits are defined by footfall patterns, which describe the sequence of swing and stance phases for each leg. As visualized in Figure 1.4, we create the following interface to visualize and specify the footfall patterns for the gaits and transitions. The x axis stands for the time sample points, which represent the phase of the locomotion cycle. For this work, we discretize the phase into a number of sample points, rather than assuming a continuous representation. The y axis records the foot numbers. We then use red to represent swing states, and white to characterize stance states.



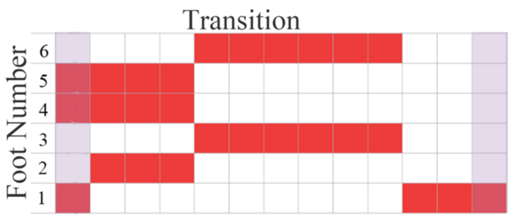
Gait1: 4-limb gait (the robot's two front limbs, limb1 and limb 4, are in swing all the time), the blue cursor is selecting a boundary constraint for the initial state of the transition



Gait2: 5-limb gait (one of the robot's two front limbs is in swing all the time), the blue cursor is selecting a boundary constraint for the end state of the transition

Figure1.2: 4-limb gait and 5-limb gait

For example, the above figure depicts a five-limb gait and a four-limb gait. The red row denotes a limb swinging in the air all the time. Our goal is to generate a footfall pattern, like the one shown below, to enable smooth transitions between steady-state gaits.



Optimized transition footfall pattern result

Figure1.3: Possible optimized transition footfall pattern result

To get a footfall pattern which could generate stable robot motions, the design is supposed to meet a few constraints both spatially and temporarily which we will introduce in the later part of this thesis.

Contributions and Outline

Chapter 2 explores relevant literature. The existing methods for footfall pattern design are either limited to some specific cases[7] or too time-consuming[6]. Based on current motion-planning techniques[1] as well as the inspirations from animal world[14], our method outperforms most of the current methods and generate footfall patterns automatically within limited time.

Chapter 3 details our method. We propose a few constraints regarding various factors like robot stability and coordination in movements, and solve the problem using Gurobi Optimizer[4].

Chapter 4 shows our implementation results on real robots like hexapods and dodecapods. We validate our techniques on several different conditions including limb-broken cases and multi-task cases. The results turn out to be promising, with 90 successful footfall patterns out of 100 cases.

Chapter 2

Literature Review

Legged robots get their prototypes from animals[21]. In this chapter, we investigate different methods of generating gaits and footfall patterns in robotics field to show our method's superiority.

2.1 Existing Motion Planning Methods

A large amount of research has been done on designing gaits in the process of motion plan optimization. Eldershow and Yim, for example, proposed a model for quadrupedal locomotion by computing the robot's overall motion first including the location and orientation of the robot's center of gravity, and then searching for a set of leg motions which enable the vehicle to move along the path[23]. Specifically, they built up a decision tree to select the most suitable locomotion strategy along the path: to lean forward with all the legs on the ground, or to lift up one leg but with the center of gravity staying at the original place. By focusing on high-level trajectory planning first, the method reduced the dimension of the planning problem, and could implement more subtle adjustments when it came to the low-level foot planning.

Another type of motion planners require us to provide an input footfall pattern before planning motions[16]. In the field of legged robots, for example, Vittorio and Bernhard came up with a method which could generate motion plans for 3D-printed robots of arbitrary shapes[1]. Bretl and Lall proposed a multi-step motion planning approach for various types of climbing robots[19]. Both of the methods asked for a footfall pattern input to generate motions. How to actively design these footfall patterns without previous knowledge of the robot's states, therefore, becomes a crucial problem, which is also the main focus of our research.

2.2 Gait Generation Method

There has been a long research history for robot gait and footfall pattern design. In real world, animals adopt gaits, and initial robot gait design, especially the design for those bio-inspired robots, are mostly based on imitation from the animals.

Central Pattern Generator Methods

Central pattern generator gets its inspiration from animal central neural system which could produce rhythmic signals and control the behavior of locomotion. In robotics, a central pattern generator is a system with several neurons which could generate rhythmic signals and actively interact with each other[5].

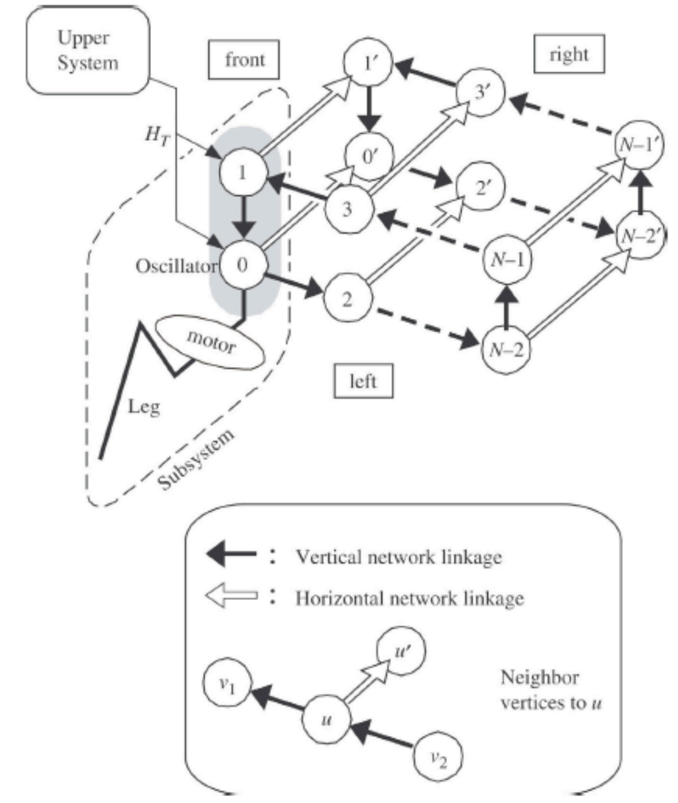


Figure2.1: A central pattern generator model

Each neuron is a single oscillating unit, and is influenced by other neurons linked to it. In the above figure, the network linkage is classified into vertical ones and horizontal ones. Each couple of vertical-linked neurons forms a periodic signal to control a robot leg and the movement signals are propagated between the legs through horizontal network linkage, which results in a phase delay between the periodic movements of two neighboring legs. The method could generate very animal-alike gaits since animals adopt similar strategies(each leg performs the same movements but at different phases) in real world. However, it will need a more sophisticated network when an artificially-deigned gait is involved, especially when each leg does not have the same length of duration. The assumption that each limb's movement is influenced through the horizontal linkage by other legs could also cause problem when one limb is damaged and fast computation of a new gait for a prompt reaction is required in real-world practice. The newly generated gait will be based on nothing from the previous design since the movements of other limbs are all, more or less, affected by the broken one. What is more, CPG unfortunately adopts a computationally expensive global stochastic search to find the oscillators' parameters, which makes this process of regeneration even slower. One plausible way is to take all broken cases into account and generate a corresponding gait for each one in advance. However, the process of this preparation may be extremely tedious. Given a robot composed of 12 or fewer modules, for example, it will take CPG about two weeks to go though all the possible cases and prepare for a prompt reaction when some limbs are damaged[6].

Robot-specific Methods

Robot-specific methods are methods designed aiming at one specific type of robot. They regard the footfall patterns as one of the optimized variables in the motion-plan optimization process, and imposed constraints to those variables based on the robot's body structure. A shape basis optimization algorithm has been applied on a snake robot by approximating the shapes of systems using a linear combination of two shape basis functions. The geometric mechanics formed based on these basis functions is then used to design gaits by adjusting the angle of each snake joint, taking advantage of existing techniques like connection vector fields and height functions[7].The methods could only be adopted to a limited number of robots, and therefore lacks generality when a new type of robot is introduced.

2.3 Transition Footfall Pattern Generation Method

Footfall patterns in the transition process between gaits is harder to imitate as animals usually perform them in a relatively short time. Scientists have attempted oceans of methods including joint angle interpolation and CPG we would mention below to solve the gait transition problem. In some extreme cases, they even let the robot stop for a while, and then restart to transit to a new state. Though feasible, the solution would win no preference when faced with tasks asking for a prompt reaction. Below are two other popular methods in designing footfall patterns for the transition process.

Joint Angle Interpolation Method

The gait transition process could be obtained from joint angle interpolation[9]. Each joint's angle at a specific time point during transition is obtained though the interpolation of its values in the given two gaits. For example, [10] adopted fifth-polynomial interpolation to propose transitions between quadruped gaits. Although easy to implement, the method has its own drawback like lacking generality and suffering from a limitation in moving speed. We will show them later in the result part, and present the advantage of our method.

Central Pattern Generator Method

Central pattern generator could also be used in the creation of gait transitions. Given the oscillators' parameters of two gaits, current techniques could generate the robot footfall patterns for transitions by assigning the vertical and horizontal network linkage mentioned above. However, the transition process still needs to be recomputed when one of the limbs is damaged, and the precomputation process is extremely time-consuming.

Our research overcomes the disadvantages mentioned above, generating promising gaits and footfall patterns within limited time.

Chapter 3

Method

3.1 Problem Formulation

Our goal is to automatically design footfall patterns for transitions between two arbitrary gaits.

In the area of locomotion, footfall patterns are sequence of swing and stance phases for each leg, which form various types of animal gaits. For a specific limb i, we use a variable $x_{i,j}$ to represent if it is swinging in the air moving to the next locomotion point($x_{i,j} = 0$) or being on the ground to support the body($x_{i,j} = 1$) at a certain time point j in time, and formulated the problem as a mixed-integer quadratic model in our research.

We design the following interface to represent the footfall patterns for the gaits and transitions. X axis stands for the time sample points, while y axis records the foot numbers. We then use red to represent swing states, and white to characterize stance states.

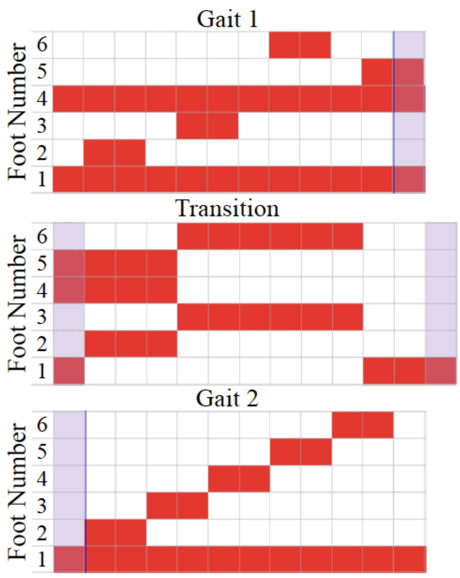


Figure3.1: Gait and transition interface

Each row i represents the time line for a limb, and each column j symbolizes a time sample point. The corresponding $x_{i,j}$ at row i, column j therefore, characterizes the state for a limb i at a specific time sample point j: be in swing (red, $x_{i,j} = 0$) or be in stance(white, $x_{i,j} = 1$). For a robot with n_F feet and a plan with m time steps, we have: $i \in \{1, \ldots, n_F\}$, $j \in \{1, \ldots, m\}$.

If $x_{i,j}$'s state is not changeable in our process of optimization, it would be masked by purple. The cursor denotes the current position on the timeline.

To meet all of our expectations for the transition footfall pattern, the variables $x_{i,j}$ are supposed to satisfy certain constraints. Our goal, then, is to find an optimized solution for those variables to get an optimal footfall pattern for the transition.

Our footfall pattern definition is very flexible, and can represent a vast array of different gaits, both for steady-state motions, and for transitions. The question addressed in this work is the following: from the space of all possible footfall patterns, which ones lead to smooth transitions between pairs of input gaits? To answer this question, we develop an optimization-based approach to automatically compute values for the variables defining footfall patterns.

3.2 Constraints and Objectives

3.2.1 Time-line frequency objective

We desire transition movements which appear coordinated, and therefore try to minimize the number of changes between swing and stance states. We use a variable y_j to denote whether any foot switches between swing and stance at each time step j.

$$\forall j \in \{1, \ldots, m-1\}, y_j = \begin{cases} 1 & \sum_{i=1}^{N_F} \left(x_{i,j} - x_{i,j+1}\right)^2 > 0 \\ 0 & \sum_{i=1}^{N_F} \left(x_{i,j} - x_{i,j+1}\right)^2 = 0 \end{cases}$$

then minimize the number of such time steps.

$$\underset{\substack{x \in \{0,1\}^{(n_F \times m)}, \\ y \in \{0,1\}^{(m-1)}, \\ z \in \{0,\ldots,n_F\}^{m-1}}}{\text{minimize}} \sum_{j=1}^{m} y_j$$
$$\text{subject to}$$
$$h(x, y, z) = 0$$
$$g(x, y, z) \leq 0$$

Here, $x \in \{0, 1\}^{(n_F \times m)}$ represents the set of binary variables that encode swing or stance phases for each limb $i(i \in \{1, \ldots, n_F\})$ of the robot at each time sample $j(j \in \{1, \ldots, m\})$. The set of variables $y \in \{0, 1\}^{(m-1)}$ corresponds to the number of transitions between stance and swing phases, or vice versa, at each time sample, which is computed with the aid of auxiliary variables $z \in \{0, \ldots, n_F\}^{m-1}$ and constraints h(x; y; z). The first equality constraint h_1 sets z_j to be equal to the number of transitions between swing and stance phases,

$$h_1 = z_j - \sum_{i=1}^{n_F} (x_{i,j} - x_{i,j+1})^2 = 0, \forall j \in \{1,2, \ldots, m-1\}$$

The second equality constraint h_2 sets y to be 1 for every time step that exhibits at least transition between phases, and 0 otherwise,

$$h_2 = (y_j - 1)z_j = 0, \forall j \in \{1,2, \ldots, m-1\}$$

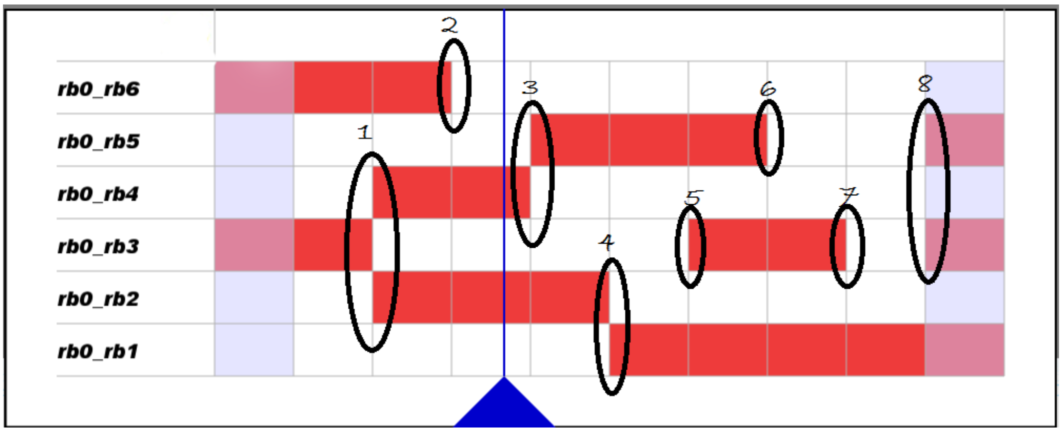


Figure3.2: Time-line frequency objective. This sample has a frequency count of 8.

While the number of transitions between swing and stance phases is used as a regularizer, the constraints g(C) that guide the emergence of transition foot fall patterns, are formally defined below.

3.2.2 Boundary constraint

The status of the start and end states of the transition, which lies in the two target gaits, is first introduced by adding boundary constraints, which include each foot's swing or stance status. The bounded rectangles are protected by the purple masks in the interface, which means that their values would not change in the optimization process:

$$g_1 = x_{i,j} = x\_bound_{i,j}, \forall i \in \{1, \ldots, n_F\}, j \in \{1, m\}$$

$x\_bound_{i,j}$ represents the bounding constraint value got from the two target gaits.

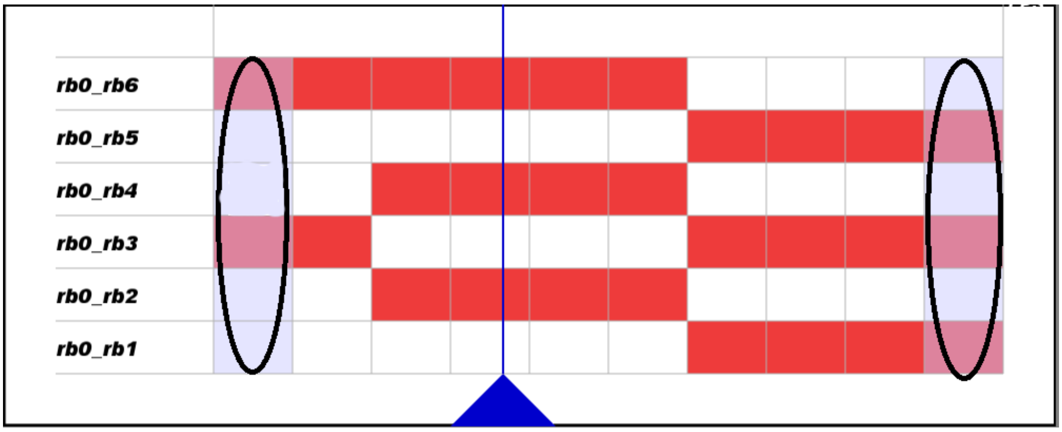


Figure3.3: Boundary constraint.

3.2.3 No single swing constraint

g_2 requires that each swing phase last more than one time step,

$$g_2 = x_{i,j} - x_{i,j+1} + x_{i,j+2} - 1 \leq 0, \forall i \in \{1, \ldots, n_F\}, j \in \{1, \ldots, m-2\}$$

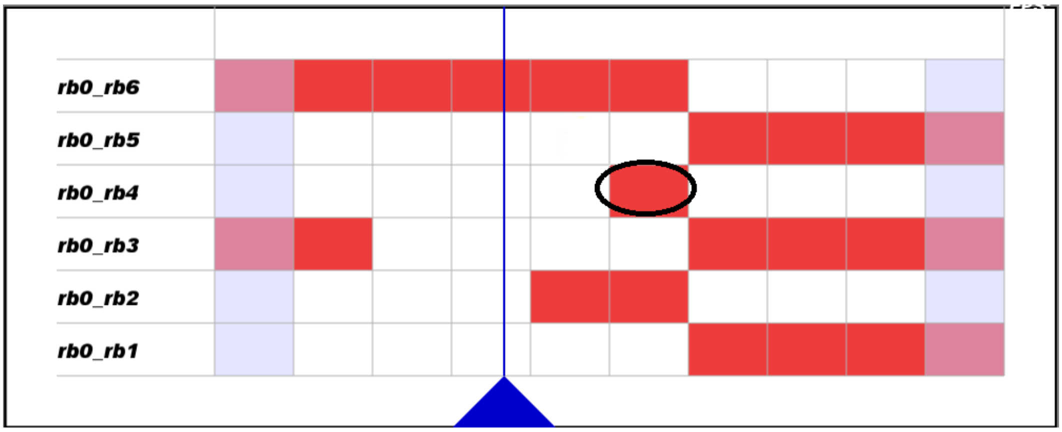


Figure3.4: No single swing constraint. This sample has a single-swing block(marked by the black eclipse), and therefore does not meet our requirements.

3.2.4 Support polygon constraint

g_3 requires that at least three limbs be in stance at each time step,

$$g_3 = -\sum_{i=1}^{N_F} x_{i,j} + n_F - 3 \leq 0, \ \ \forall j \in \{1, \ldots, m\}$$

This constraint helps ensure stance stability. However, not all combinations of stance feet configurations are equally desirable from a stability standpoint. For example, solutions that satisfy g_3 would include all feet on the left side of a hexapedal robot being in swing, while the others are in stance. Constraints g_4 and g_5 ask that at least one leg on each side is in stance at any moment in time.

$$g_4 = -\sum_{i=1}^{\frac{n_F}{2}} x_{i,j} + 1 \leq 0, \forall j \in \{1, \ldots, m\}$$

$$g_5 = -\sum_{i=\frac{n_F}{2}+1}^{n_F} x_{i,j} + 1 \leq 0, \forall j \in \{1, \dots, m\}$$

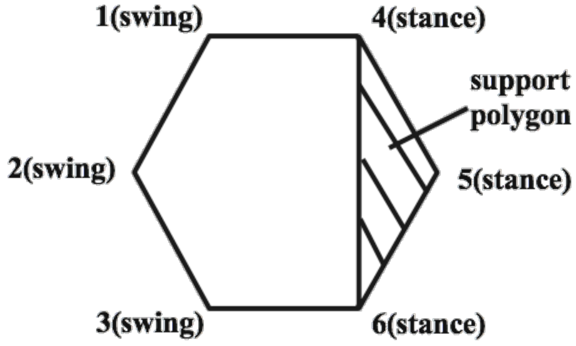


Figure3.5: limb1, limb2, and limb3 are all on the same side of the robot being in swing, a situation we would like to avoid in the footfall pattern design.

3.2.5 Limb symmetry constraint

To ensure that the feet share walking workload, g_6 and g_7 requires that the difference in time spent in swing phase between any two feet is no more than two time steps,

$$g_6 = \sum_{j=1}^{m} x_{i_1,j} - \sum_{j=1}^{m} x_{i_2,j} - 2 \leq 0,$$

$$g_7 = -\sum_{j=1}^{m} x_{i_1,j} + \sum_{j=1}^{m} x_{i_2,j} - 2 \leq 0,$$

$$\forall i_1 \in \{1, \dots, n_F\}, \forall i_2 \in \{1, \dots, n_F\}, i_1 \neq i_2$$

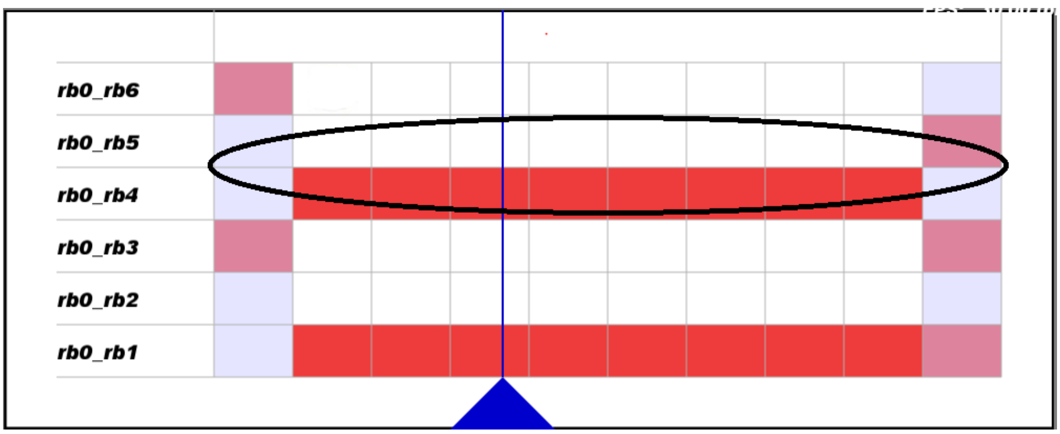


Figure3.6: Limb symmetry constraint. This sample has an uneven distribution of work load for the each limb, and therefore does not meet our requirements.

3.2.6 Limb frequency constraint

Constraint set g_8 is a limit on the number of changes between swing and stance phases for an individual foot. Over a transition period, each foot is allowed to change from swing to stance, or vice versa, at most twice,

$$g_8 = \sum_{j=1}^{m} (x_{i,j} - x_{i,j+1})^2 - 2 \leq 0\,, \forall i \in \{1, \dots, n_F\}$$

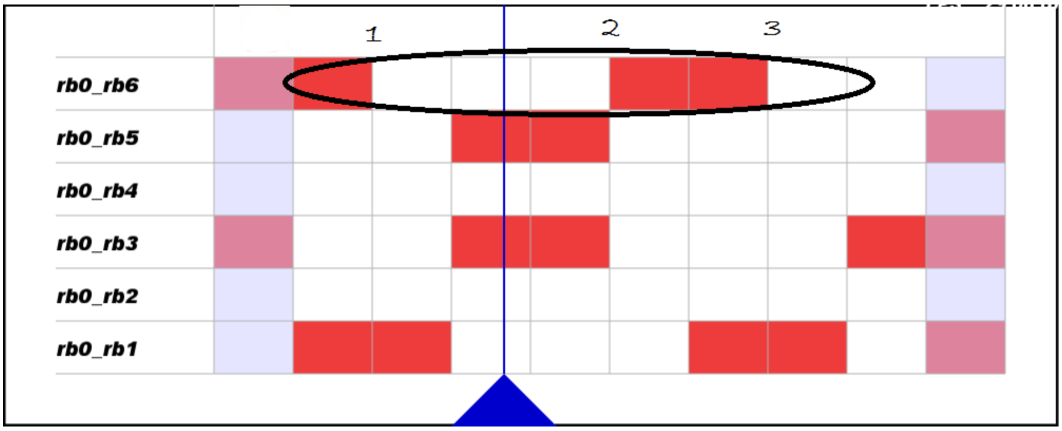


Figure3.7: Limb frequency constraint. This sample has a limb frequency count of 3 for limb6, and therefore does not meet our requirements.

3.2.7 Stance period constraint

g_9 prevents stance phases of one time step in length,

$$g_9 = -x_{i,j} + x_{i,j+1} - x_{i,j+2} \leq 0, \;\; \forall i \in \{1, \dots, n_F\}, j \in \{1, \dots, m-2\}$$

and g_{10} prevents stance phases of two time steps in length,

$$g_{10} = -x_{i,j} + x_{i,j+1} + x_{i,j+2} - x_{i,j+3} - 1 \leq 0, \forall i \in \{1, \dots, n_F\}, j \in \{1, \dots, m-3\}$$

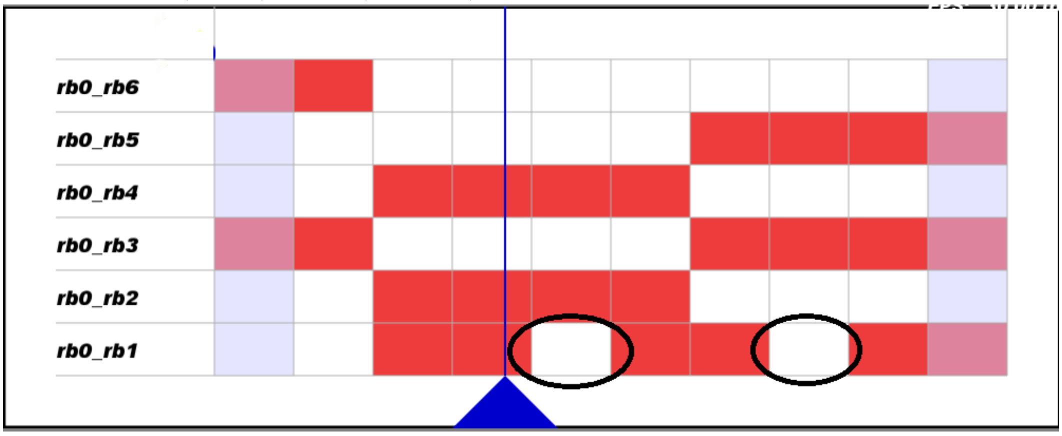


Figure3.8: Limb frequency constraint. This sample has a stance period of 1 when limb1 swings twice, and therefore does not meet our requirements.

These two heuristic constraint sets ensure that each stance phase lasts for at least three time steps, which we found to empirically result in more ”natural” appearing behavior. g_{11} sets an upper limit to the total length of the transition phase each foot may spend in stance phase, forcing each limb to be in swing for at least two time steps,

$$g_{11} = \sum_{j=1}^{m} x_{i,j} - m + 2 \leq 0, \forall i \epsilon \{1, \ldots, n_F\}$$

All of the constraints are expressed as linear or quadratic inequalities, which are later solved by the Gurobi optimizer[4], one of the state of the art optimizers for dealing with mixed-integer quadratic problems.

Chapter 4

Results

4.1 Results

As described in the previous sections, we used optimization from previous section to generate footfall patterns as input, taking advantage of an existing method[1] to generate full-body motions, and realized successful simulations and implementations on the transitions between those gaits for the following robots.

4.1.1 Quadruped Robot

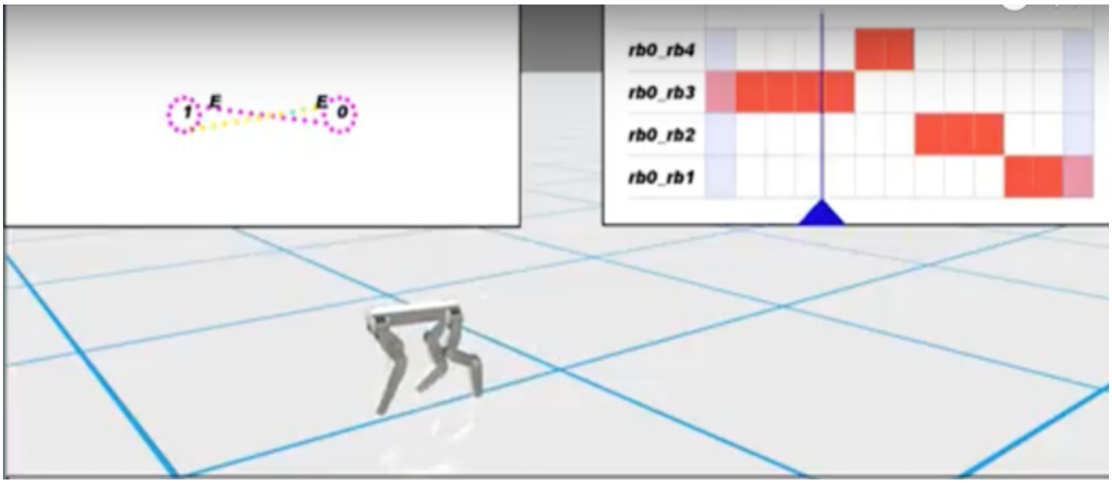
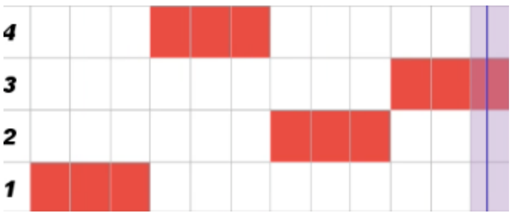
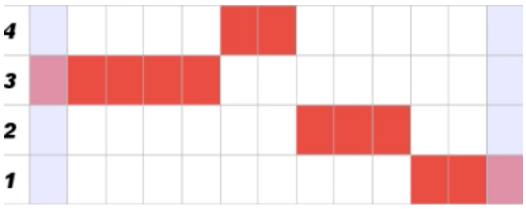


Figure4.1: Simulation on a quadruped robot

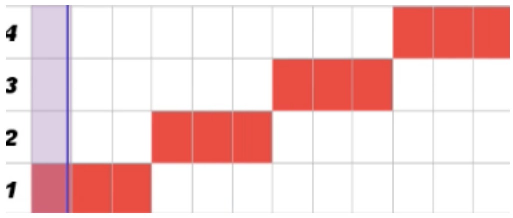
The center of mass for dog-like quadruped robots is pretty high, which makes it hard for us to ensure the stability of the robot. However, we effectively generated several stable wave gaits for the quadruped, and the transitions produced automatically between them worked surprisingly well. Here is a sample case:



Gait1: Quadruped wave gait 1, the blue cursor is selecting a boundary constraint for the initial state of the transition



Optimized transition footfall pattern result



Gait2: Quadruped wave gait 2, the blue cursor is selecting a boundary constraint for the end state of the transition

Figure4.2: 2 quadruped wave gaits, and the transition between them

4.1.2 Hexapod Robot

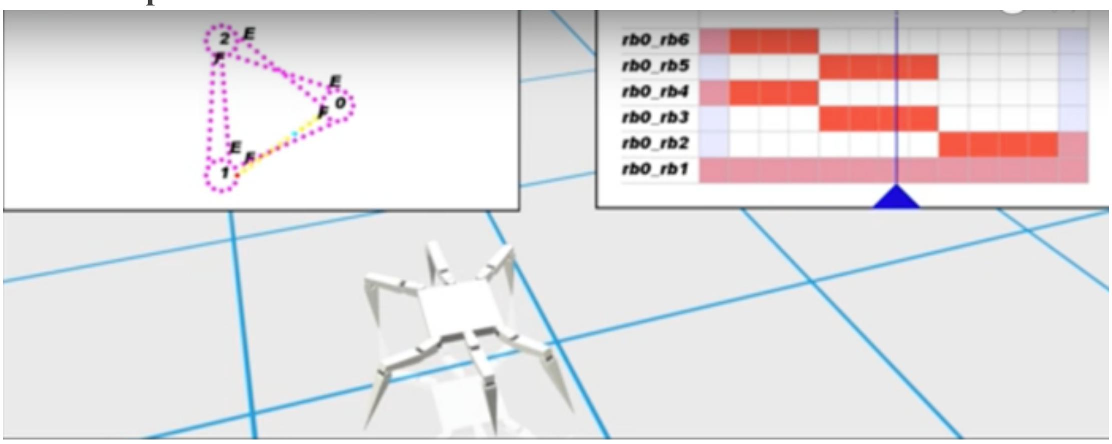
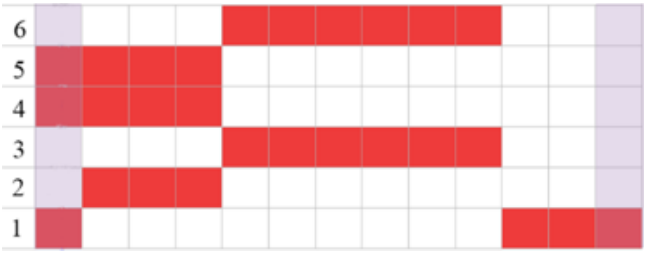


Figure4.3: Simulation on a hexapod robot

Hexapod robots have a much broader choice of gaits. Besides wave gait, they could move in tripod gait, ripple gait, and even leave some limbs in swing to perform other tasks like manipulation. With some previous constraints, like limb 1 and limb 7 are supposed to be in air all the time to carry a box, our optimizer could work out the corresponding reliable gait in a minute. The transitions created between the gaits are also both stable and delicate. Here is a sample transition formed between a 4-limb gait and a 5-limb gait.

Gait1: 4-limb gait (the robot’s two front limbs are in swing all the time), the blue cursor is selecting a boundary constraint for the initial state of the transition



Optimized transition footfall pattern result

Gait2: 5-limb gait (one of the robot’s two front limbs is in swing all the time), the blue cursor is selecting a boundary constraint for the end state of the transition

Figure4.4: 2 quadruped wave gaits, and the transition between them

We then implemented the simulation results on a real legged robot with 20 modular actuators, 3 for each of the 6 limbs and 1 for each of the 2 grippers.

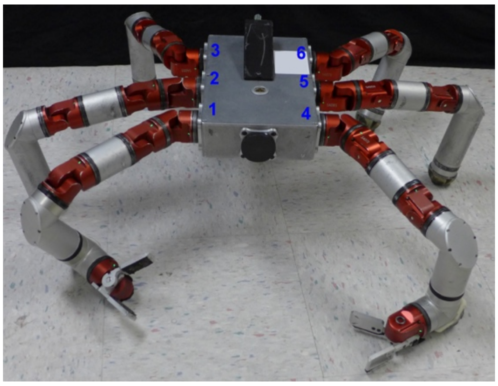


Figure4.5: Hexapod robot.

The results are promising. Approximately 90 trails would succeed in generating stable transitions out of 100 experiments. Here are a few examples:

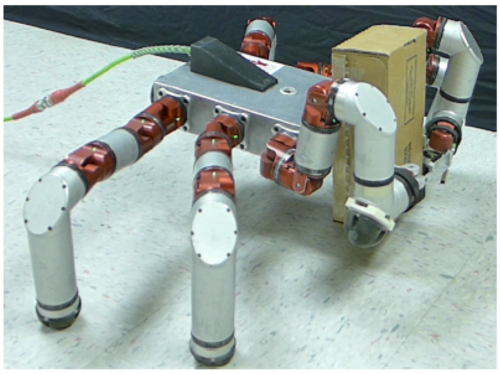


Figure4.6: A quadruped gait using two front limbs to carry a box

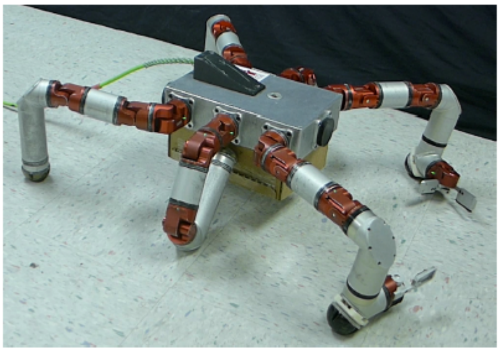


Figure4.7: A quadruped gait using two mid limbs to carry a box

4.1.3 Dodecapod Robot

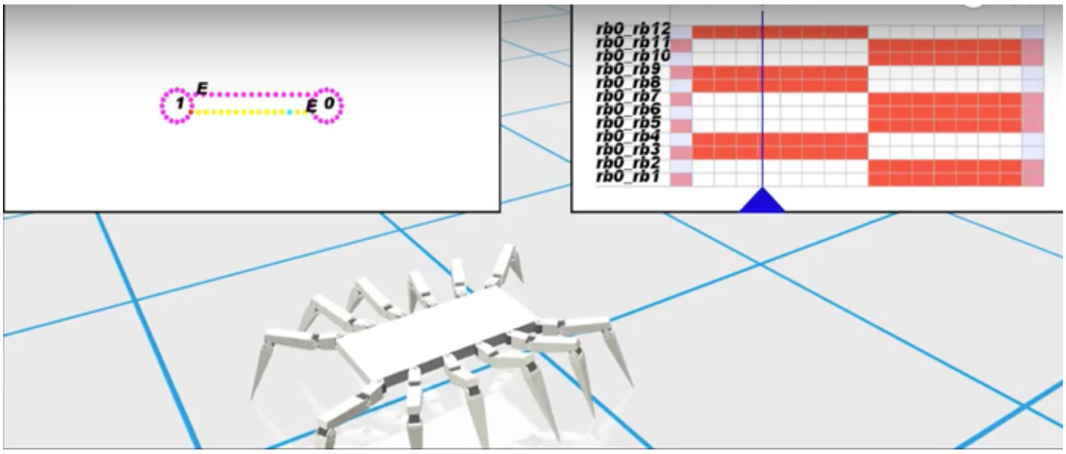
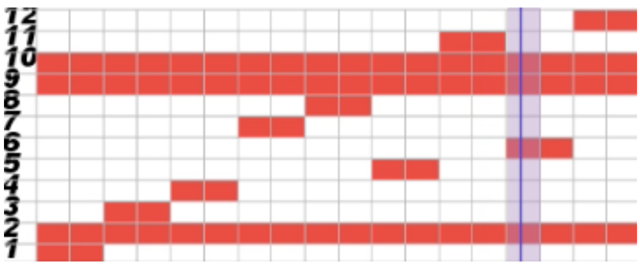
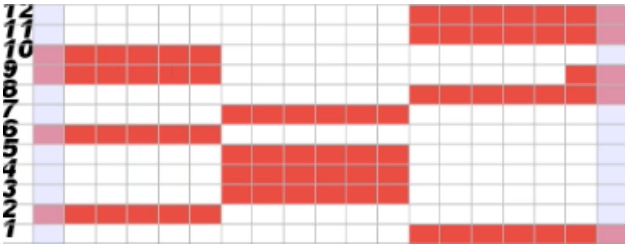


Figure4.8: Simulation on a dodecapod robot

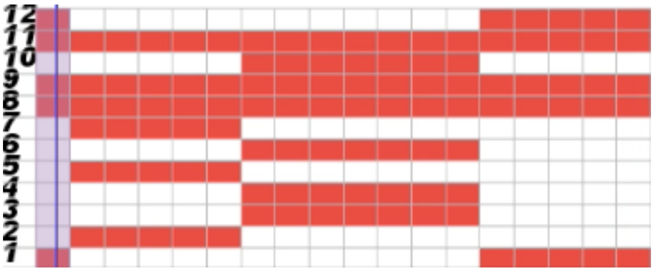
Equipped with 12 limbs, a dodecapod robot could perform almost any tasks it likes, from locomotion, manipulation, to inspection. To test our gait-generation algorithm, we randomly chose a set of limbs to be reassigned to other tasks, producing the new gaits which turned out to be pretty dependable. We then created transitions between the gaits to check the transition algorithm. Here is one sample case:



Gait1: 9-limb gait 1(3 of the robot's limbs are in swing all the time), the blue cursor is selecting a boundary constraint for the initial state of the transition



Optimized transition footfall pattern result



Gait2: 9-limb gait 2(3 of the robot's limbs are in swing all the time),

the blue cursor is selecting a boundary constraint for the end state of the transition

Figure4.9: 2 9-limb gaits for dodecapod, and the transition between them

We demonstrated the simulation results on a dodecapod with 38 modular actuators, 3 for each of the 12 limbs and 1 for each of the 2 grippers. At the end of one leg, a camera was mounted for inspection tasks.

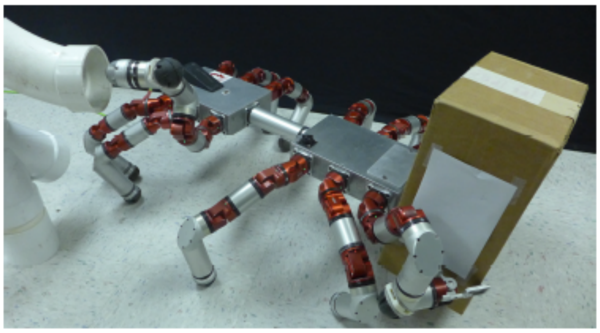


Figure4.10: Dodecapod robot.

The implementation results turned out to be promising.

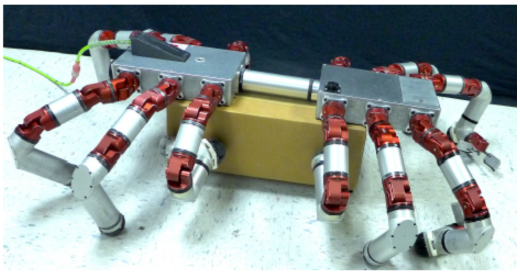


Figure4.11: A dodecapod carrying a box using its four mid limbs

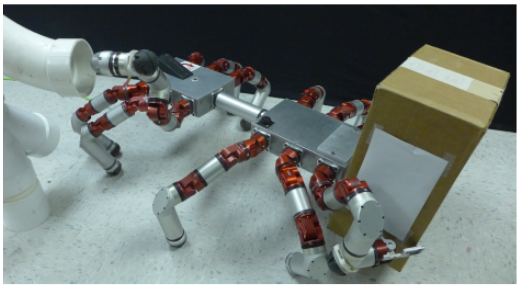


Figure4.12: A dodecapod using its two front limbs to carry a box, while one rear limb with camera to inspect the pipe

Here is a demo video for all of our implementation results:
https://youtu.be/xXfi55HPOfs

4.2 Comparison with the Interpolation Method

The method could not only generate promising gaits, but also allow the robot to move forward in the transition process. In the following section, we will compare the speed of our method with the interpolation method.

In the following table, gaits are labelled by the number of locomotive limbs: “6” for an alternating tripod gait, “5” for a pentapedal gait in which one front limb is nonlocomotive, “4” for a quadrupedal gait where both of the front limbs are lifted. The direct interpolation velocity appears to be approximately equal to or lower than the optimized transition velocity in all of the tests. What is more, in our process of simulation, the transitions generated by the joint-angle interpolation method may not be stable, and the robot may even fall down in some extreme cases.

Starting gait->Ending gait	Direct interpolation velocity(cm/s)	Optimized transition velocity(cm/s)
4->5	6.4	6.4
4->6	6.4	6.2
5->4	0.08	5.9
5->6	5.2	5.9
6->4	-4.0	5.2
6->5	4.8	6.0

Table4.1: Comparison between interpolation method and our method.

Chapter 5

Conclusion

Getting inspirations from animals, we presented a novel approach for designing transition footfall patterns between arbitrary gaits. The approach formulated the problem as a mixed-integer programming model, constructing the constraints based on various factors like robot stability and coordination among limbs. The optimization speed achieved could let researchers generate footfall pattern on line.

The approach could be applied to numerous scenarios even when some of the robot limbs are damaged or used for other functions like manipulation and inspection, which was validated on a real robot – CMU snake monster. As future work, we plan to explore more in the automatic recognition of the potential ‘feet’ limbs to generate more stable robot motions.

.

Bibliography

[1] V. Megaro, B. Thomaszewski, M. Nitti, O. Hilliges, M. Gross, and S. Coros, "Interactive design of 3d-printable robotic creatures," ACM Transactions on Graphics (TOG), vol. 34, no. 6, p. 216, 2015.
[2] Haynes, Galen C., Fred R. Cohen, and Daniel E. Koditschek. "Gait transitions for quasi-static hexapedal locomotion on level ground." Robotics Research. Springer Berlin Heidelberg, 2011.105-121.
[3] Haynes, G. Clark, and Alfred A. Rizzi. "Gaits and gait transitions for legged robots." Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.. IEEE, 2006.
[4] Gurobi optimizer reference manual. Accessed 2017-2-28. [online]. Available: http://www.gurobi.com/resources/getting-started/mip-basics.
[5] Ijspeert, Auke Jan. "Central pattern generators for locomotion control in animals and robots: a review." Neural networks 21.4 (2008): 642-653.
[6] Cully, Antoine, et al. "Robots that can adapt like animals." Nature 521.7553 (2015): 503-507.
[7] Gong, Chaohui, Daniel I. Goldman, and Howie Choset. "Simplifying Gait Design via Shape Basis Optimization."
[8] Weingarten, Joel D., et al. "Automated gait adaptation for legged robots." Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on. Vol. 3. IEEE, 2004.
[9] Gehring, Christian, et al. "Control of dynamic gaits for a quadrupedal robot." Robotics and automation (ICRA), 2013 IEEE international conference on. IEEE, 2013.
[10] Nansai, Shunsuke, et al. "A novel approach to gait synchronization and transition for reconfigurable walking platforms." Digital Communications and Networks 1.2 (2015): 141-151.
[11] Chen, Weihai, et al. "Smooth transition between different gaits of a hexapod robot via a central pattern generators algorithm." Journal of Intelligent & Robotic Systems 67.3-4 (2012): 255-270.
[12] Gan, Zhenyu, and C. David Remy. "A passive dynamic quadruped that moves in a large variety of gaits." Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on. IEEE, 2014.
[13] Schmitt, John, and Philip Holmes. "Mechanical models for insect locomotion: dynamics and stability in the horizontal plane–II. Application." Biological cybernetics 83.6 (2000): 517-527.
[14] Alexander, R. McNeill. Principles of Animal Locomotion. STU - Student edition ed., Princeton University Press, 2003, www.jstor.org/stable/j.ctt4cg9j1.

[15] Alexander, RMcN, and A. S. Jayes. "A dynamic similarity hypothesis for the gaits of quadrupedal mammals." Journal of zoology 201.1 (1983): 135-152.
[16] Hauser, Kris, et al. "Motion planning for legged robots on varied terrain." The International Journal of Robotics Research 27.11-12 (2008): 1325-1349.
[17] Hauser, Kris, Timothy Bretl, and J-C. Latombe. "Non-gaited humanoid locomotion planning." Humanoid Robots, 2005 5th IEEE-RAS International Conference on. IEEE, 2005.
[18] Bretl, Tim, et al. "Multi-step motion planning for free-climbing robots." Algorithmic Foundations of Robotics VI (2005): 59-74.
[19] Yim, Mark, et al. "Modular self-reconfigurable robots." Encyclopedia of complexity and systems science. Springer New York, 2009. 5618-5631.
[20] http://biorobotics.ri.cmu.edu/robots/SEAModularity.php
[21] Silva, Manuel Fernando, and JA Tenreiro Machado. "A literature review on the optimization of legged robots." Journal of Vibration and Control18.12 (2012): 1753-1767.
[22] Moubarak, Paul, and Pinhas Ben-Tzvi. "Modular and reconfigurable mobile robotics." Robotics and Autonomous Systems 60.12 (2012): 1648-1663.
[23] Eldershaw, Craig, and Mark Yim. "Motion planning of legged vehicles in an unstructured environment." Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on. Vol. 4. IEEE, 2001.