WORCESTER POLYTECHNIC INSTITUTE

Locomotion Trajectory Generation For Legged Robots

by

Aditya Bhat

A thesis submitted in partial fulfillment for the degree of Master of Science

in

Robotics Engineering
WORCESTER POLYTECHNIC INSTITUTE

April 2017

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of Master Of Science.

Dr. Michael Gennert (Principal Advisor)

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of Master Of Science.

Dr. Jie Fu

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of Master Of Science.

Dr. Zhi (Jane) Li

WORCESTER POLYTECHNIC INSTITUTE

Abstract

Robotics Engineering
WORCESTER POLYTECHNIC INSTITUTE

Master Of Science

by Aditya Bhat

This thesis addresses the problem of generating smooth and efficiently executable locomotion trajectories for legged robots under contact constraints. In addition, we want the trajectories to have the property that small changes in the foot position generate small changes in the joint target path. The first part of this thesis explores methods to select poses for a legged robot that maximises the workspace reachability while maintaining stability and contact constraints. It also explores methods to select configurations based on a reduced-dimensional search of the configuration space. The second part analyses time scaling strategy which tries to minimize the execution time while obeying the velocity and acceleration constraints. These two parts effectively result in smooth feasible trajectories for legged robots. Experiments on the RoboSimian robot demonstrate the effectiveness and scalability of the strategies described for walking and climbing on a rock climbing wall.

Acknowledgements

I am deeply grateful to my adviser Prof. Michael Gennert, for his strong encouragement and guidance during my studies. He generously helped me work on my research interests and guided me throughout the process. He is a treasure trove of wisdom and patience. His ability to convey research ideas and patiently direct my research have helped me immensely during the process for which I offer my sincerest appreciation.

I would like to thank Prof. Jie Fu and Prof. Jane Li for their insightful discussions and support during the whole period.

I am deeply indebted to Dr. Kris Hauser who introduced me to the world of motion planning research. This thesis would not have been possible without his insights and and research acumen.

I would like to thank Andrew and Sung Hoon for their support in helping me run experiments on the robot.

Many thanks to my friends, Renato, Peng, Matt, Nathalie and Ulkesh with whom I've had invaluable discussions throughout our studies. Thanks to Mihir, Vinayak, Jayam, Andres, Benjamin, Josh, Chris for your friendship.

Finally, I would like to thank my parents for their lifelong love and support without whose sacrifices, I would never have been able to pursue my dreams.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Legged vehicles are potentially better than their wheeled counterparts in navigating cluttered environments and steep terrain primarily because of their ability to step on or over obstacles. This unique ability of legs makes legged robots a prime candidate for tasks such as search and rescue, planetary exploration, exploration of volcanoes and cliffs. Despite this advantage, legged robots have not been demonstrably efficient when compared to wheeled robots for most of the above mentioned tasks. This is primarily due to the complexity involved in large number of degrees of freedom in legged robots and their coordination to achieve high level goals such as walking or running.

Motion planning can be described as the process of planning paths for a system such that the configuration space path is obtained as a result of the planning process. This configuration space path reaches the goal location/region by taking into consideration the constraints on the robot and the environment. Motion planning can be applied to perform low level tasks of configuration space planning where high level planning can be performed either by an operator or using high level decision making algorithms which decide the goal. In case of legged robots, the feasible region of the configuration space is unlike most other fixed base manipulators or wheeled robots. The configuration space is a manifold in the high dimensional space of the degrees of freedom of the robot. These manifolds are formed due to the constraints added by feet on the ground.

Planning in the configuration space for legged robots has to be performed in the sub-manifold which includes stability constraints, torque constraints and contact constraints due to the footfalls of the end effectors. Also, the configuration space changes when contacts are broken or added. This makes tasks like walking a multi-modal planning problem, where each mode can be fully defined by the goal (such as swinging a leg) and the constraints (such as maintaining ground contacts in the non-moving legs).

The need for multi-modal planning is seen in grasping, legged locomotion and dexterous manipulation. This thesis focuses only on legged locomotion for multi-limbed robot in flat terrains and cliffs. It builds on the idea of multi-modal planning for legged robots to select configuration in a mode to come up with effective paths between multiple modes. Its secondary focus is to generate effective trajectories from the paths obtained by the multi-modal planners.

FIGURE 1.1: RoboSimian robot

The strategies and algorithms developed in this thesis have been used for various tasks for JPL's four legged RoboSimian robot (as shown in fig 1.1). These tasks range from planning paths to walk on flat ground to selecting grasp configurations for climbing on a rock-climbing wall. The selection strategies involve routines which prune unnecessary configurations, improve search and come up with guesses which are more effective in finding a path from the current mode to the next. Also, approaches to find effective start configurations or natural poses for the RoboSimian robot so as to improve its manipulability. Better timing generation strategies are also explored.

1.1 Overview of this thesis

This thesis addresses the following broad fields of motion planning research.

- *Configuration Selection.* The first component analyses methods for selection of configurations for maintaining repeatability of motion, increase probability of finding a path from the previous configuration and maximize the reachability region(Chapter 2).
- *Dimensionality Reduction and C-Space Analysis.* The second component analyses methods to visualize the configuration space and pick seed configurations to avoid singularities. This method can be effective in improving sampling strategies(Chapter 3).
- *Heuristic Based Velocity Optimization For Trajectories.* The third component analyses various velocity optimization strategies for a given configuration space path. This thesis introduces a heuristic method of generating trajectories from a configuration space path(Chapter 4).

These approaches are introduced in the following sections.

1.2 Configuration Selection

Most prior work on configuration selection has been a part of manipulability analysis for grasping problems and has not been applied to legged locomotion. Also, this manipulability analysis has been done for specific instances of grasping problems and very little work has been done on configuration selection for high dimensional legged robots. Chapter 2 comes up with algorithms for configuration selection for legged robots that makes generating walking paths easy. Most multi-modal planners are capable of planning paths between two configurations in discrete modes by taking into account the contact constraints. In case of high dimensional robots, there can be numerous configurations which meet the constraints posed by the individual modes (fig 1.2). As these multi-modal planners still use sampling based approaches like RRT and PRM to plan between these modes, they selection of a configuration in a mode can play a major role in the quality of path generated by the planning algorithm.

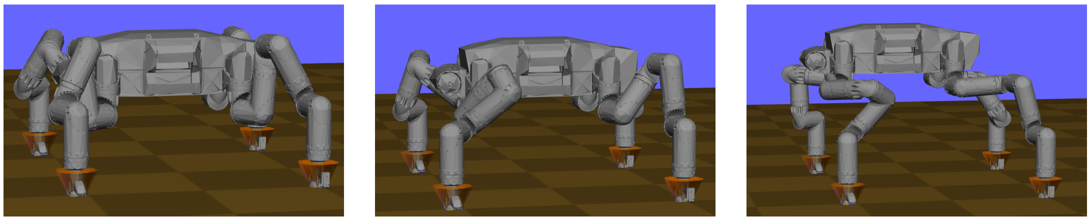


FIGURE 1.2: Multiple configurations for the same end effector position of the robot

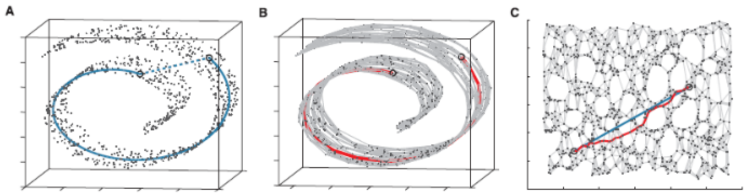
1.3 Dimensionality Reduction and C-Space Analysis

One of the drawbacks of sampling based approaches and configuration selection strategies is the inability to visualize configuration space manifolds for better selection of samples for planning feasible paths. This problem manifests itself in planning problems where suboptimal paths are formed for apparently small changes in end effector position. This leads to difficult smoothing process for trajectory optimization. This problem can also be reformulated as a redundancy resolution problem. The idea behind this is, every task space position must solve to give a single point in the configuration space. Chapter 3 explores existing techniques for dimensionality reduction for a more sophisticated selection of samples for planning.

The idea of projecting the configuration space onto a lower dimensional space is one of the two existing methods for visualizing configuration space, the other method being sampling exhaustively the linear manifolds in the configuration space. In this thesis, we explore a non-linear projection technique for selecting samples (as illustrated in 1.3). This form of analysis of the configuration space can be quite useful in repetitive tasks. By finding these configurations which would otherwise not be found by the Jacobian based approach, high quality paths can be generated without considering a large number of samples. This approach uses the analysis of the C-space to find configurations which are useful in generating high quality paths for tasks such as walking on flat ground.

Chapter 3 covers the approach considered for finding smooth paths for generating walking trajectories using the analysis of the configuration space by dimensionality reduction of the configuration space. This approach is used to generate walking paths for the RoboSimian. Simulation results of this approach show the that

Figure 1.3: Reduction in dimensionality from 3D space to a 2D plane



biasing samples based on results of dimensionality reduction can be effective in generating high quality paths.

1.4 Heuristic Based Time Optimization

Chapter 4 covers a minimum-time trajectory generation strategy. This approach works on a subset of the trajectory generation problems, i.e. the problems where the path has already been generated. Time optimization is performed by modelling curves for time to fit for paths between milestones. Two approaches are designed for modelling the velocity profile. The first approach fits a curve for any given path based on predefined curves for time values. The scale factor for the time curve is determined by the path. This is a heuristic based approach. The effectiveness of the time-optimization depends on the quality of the path. In the second case, a quadratically constrained linear program is considered for generating minimum time trajectories. In this case, velocity and acceleration limits of each joint is considered.

This approach is different from the existing time optimization strategies because current strategies solve an optimal control problems whereas this approach uses a heuristic based time scaling. Also, most approaches consider the dynamics of the robot while determining the minimum-time trajectory. The effectiveness of both the proposed approaches is demonstrated.

Motion planning literature shows various algorithms which work effectively in certain scenarios where parameters have to be tweaked to make effective generation of trajectories possible. Throughout the work of this thesis, the use of algorithms involving tweaking of parameters has been minimized. Also, the improvement in the quality of paths can be argued for other robots and scenarios as well, primarily due to reduction in the number parameters required to tune.

1.5 Summary Of Contributions

For this thesis work, I was able to develop:

- an effective configuration selection strategy for high dimensional legged robots for locomotion tasks.
- an effective start configuration and natural start pose for the robot.
- a locomotion planner for the RoboSimian quadruped robot to navigate on flat ground and a rock climbing wall.
- a heuristic based trajectory generation strategy.

Chapter 2

Configuration Selection

The problem of configuration selection is also called as the redundancy resolution problem. This problem is quite pervasive in robotics with contributions being made for grasping problems [1] and anthropomorphic movements [2] for exoskeletons on a regular basis. However, a generic configuration selection strategy has not been applied to locomotion problems involving legs with redundant degrees of freedom. This configuration selection strategy is quite important especially in cases where there are multiple constraints on the system. These constraints can be in the form of footstep locations, body posture or overall stability of the robot. With added constraints, the motion of the robot is in a small manifold on the configuration space. As the planning has to be done on this constrained manifold, the probability of finding a path in the manifold is low. Constraint biased planning techniques are effective in finding paths in the constrained manifold. However, in case of redundant manipulators, numerous configurations can satisfy the required constraints and the method to pick a configuration can significantly affect the planning process.

To plan for paths between configurations that satisfy the constraints, it is necessary to reduce the distance between the configurations in the constrained configuration manifold. This maximises the chances of a random sampling based planner such as PRM or RRT to find a configuration space path between the start and goal configuration within a reasonable time. Also configuration selection strategies must ensure that it does not move to a configuration difficult for it to get out of.

This chapter explores two approaches for configurations selection. These approaches considered involve both randomized sampling, and a more informed sampling of the configuration space.

In case of redundant robots, the default start configuration can play a major role in deciding the effectiveness of repeatable paths. This chapter also attempts to tackle the problem of selecting start configurations used to generate repeatable motions.

2.1 Background

The open challenge of quasi-static limb motion planning for kinematically redundant legged robots is a fairly new problem. This is primarily due to the fact that legs robots were not designed to be redundant. However, as robots are challenged to perform more complex tasks, legged robots with redundant degrees of freedom in the leg are being designed. This leads to the problem of redundancy resolution.

Some promising work in this field was done by Satzinger et. al.[3] on the RoboSimian robot. In this work, a reduced dimensional inverse kinematic look-up table is generated for a practical approach to configuration selection and walking. This approach consists of generating set of configurations with certain smoothness properties such as a path-wise redundancy between the configurations in the look-up table. This is done by classifying configurations into families of inverse kinematic solutions and considering configurations of only a few favourable families to populate the look-up table. This approach avoids picking a goal configurations arbitrarily, which in turn helps plan intermediate paths effectively. However, this approach keeps the robot in a conservative profile without utilizing the full capabilities of redundancy of the robot.

A more rigorous approach in solving the problem of redundancy resolution addressed by Hauser [4]. Here, the problem is posed as a problem of mapping a higher dimensional compact set to a lower dimensional set. This process also considers the start configuration of the manipulator so that a smooth path can be generated when the intermediate interpolated configurations between the start and the resolved goal configuration are generated. It works by building a database of configurations for a discretized set of points in the task space. Next, gradient descent is performed about the configurations to reduce the overall change in the robot limb movement between the start and goal configuration.

Other approaches in configuration selection include computing quality indices of configurations such as, the configuration's distance from singularities, or obstacles. One such approach is computing the manipulability index. This approach adapts the usage of the Yoshikawa's manipulability index [5] which is a quality measure for redundant manipulators to describe distances from singular configurations. This approach is based on the analysis of the ellipsoid spanned by the singular vectors of the Jacobian matrix.

[6], [7] and [8] provide variations to the Yoshikawa's manipulability index. The variations include adding penalization functions for joint limits and obstacles and including these parameters in the augmented Jacobian matrix. One of the methods considered for the configuration selection derives from this analysis. However it must be noted that these tasks involve selecting configurations for a fixed base manipulator.

Some work on selecting configurations for mobile manipulators, given end effector paths was done by Oriolo et. al.[9] where the ideas of fixed base manipulators were modified by exploiting the partition of the generalized coordinates between the manipulator and the moving base whose constraints were accounted for in the planning problem. It also uses a randomized configuration generation compatible with the end effector constraints. This problem, formulated as a Motion Planning along End effector paths (MPEP problem) has been solved in many ways including using an optimal control formulation [10], [11], [12]. This approach might not necessarily guarantee success as it can lead to a two point boundary value problem. Sampling based planning approaches are considered for the same as well. This includes variations of the RRT where a tree is grown in the constraint manifold along the discretized version of the task space path.

Zacharias et. al.[13] worked on methods for selecting the position of the robot to generate trajectories for the manipulator given significant improvements in global redundancy resolution. Global redundancy resolution methods must generate a single configuration for a given task space position also having smoothness properties so that discontinuities caused by singularities can be avoided. But, it must be noted that global redundancy resolution is still not efficiently solved for non-trivial robots (or robots with high degrees of freedom).

Finally, another approach considered is a learning based algorithms which tries to generate anthropomorphous movement of the manipulators [14]. These approaches have been applied extensively in exoskeletons. Other approaches involve constraining the swivel angle and solving the analytical inverse kinematic solution [15]. These are quite effective in case of movements which try to mimic human motion. However, these approaches do not help realize the full potential of non-anthropomorphic robots where the robot need not have any human like movements.

2.2 Configuration Selection For RoboSimian

Selecting a configuration given the inverse kinematic goal for a redundant robot manipulator can be performed using a simple heuristic based path-wise redundancy resolution technique for most cases. However, if the configuration is close to a singularity, this method might fail to provide a smooth path for which a different singularity based configuration selection strategy is developed. The configuration selection strategy consists two classes of approach. They can be described as follows:

- *Euclidean Distance Configuration Selection* This method uses the robot start configuration as the seed configuration. Using the Jacobian based inverse kinematic solver and the right seed, configurations which can be achieved by small movements in the joint positions can be produced.
- *Manipulability based Configuration Selection* This method uses the distance from singularity as the method for configuration selection. This method is especially effective in large movements of the manipulator.

Both these selection strategies are described in the following subsections.

2.2.1 Euclidean Distance Configuration Selection

For systems which require small movement of end effector, seeding the Jacobian solver with the start configuration (or the current configuration) will lead to joints moving small distances to reach the inverse kinematic goal. The Jacobian can be calculated numerically using the following:

$$J(\vec{q}) = \begin{bmatrix} \frac{\partial Q(\vec{q})}{\partial x_1} & \frac{\partial Q(\vec{q})}{\partial x_2} & \dots & \frac{\partial Q(\vec{q})}{\partial x_{\mathrm{n}}} \\ \dots\dots & \dots\dots & \dots\dots & \dots\dots \\ \xi_1 z_0(\vec{q}) & \xi_2 z_1(\vec{q}) & \dots & \xi_{\mathrm{n}} z_{\text{n-1}}(\vec{q}) \end{bmatrix} \tag{2.1}$$

If the Jacobian is calculated for the start configuration, a smooth inverse kinematic solution is likely for the goal configuration if it is calculated using the jacobian based inverse kinematic equation. The inverse kinematics can be given by the equation:

$$\frac{d\vec{q}}{dt} = J(\vec{q})^{-1}\frac{d\vec{x}}{dt} \tag{2.2}$$

Here $\vec{q}$ is the joint space position and $\vec{x}$ is the task space position. This method works in most cases where the start configuration is close to the goal. Also, this method ensures only one limb of the legged robot moves at a time. A major advantage of this approach is that planning can be performed in a smaller configuration space.

This process involves selecting random seed configurations and solving the Jacobian based inverse kinematic solution for multiple seeds. The euclidean distance between the generated configurations to the start configuration is measured and the configuration with the lowest distance is considered. In case of constraints with the inverse kinematic goal, a constraint biased configuration selection is performed.

This process is explained in the algorithm 1.

Data: Robot body height, Robot Kinematics, End effector position and orientation

Result: Configuration of the limb

database=[Configuration, Distance to start];

foreach *count* $\leq$ *100* **do**

    Sample random start configurations;

    if *SolveInverseKinematicConstraints(*IK_{G}*oals)* **then**

        distance=Calculate L2 norm of the new configuration and the start configuration;

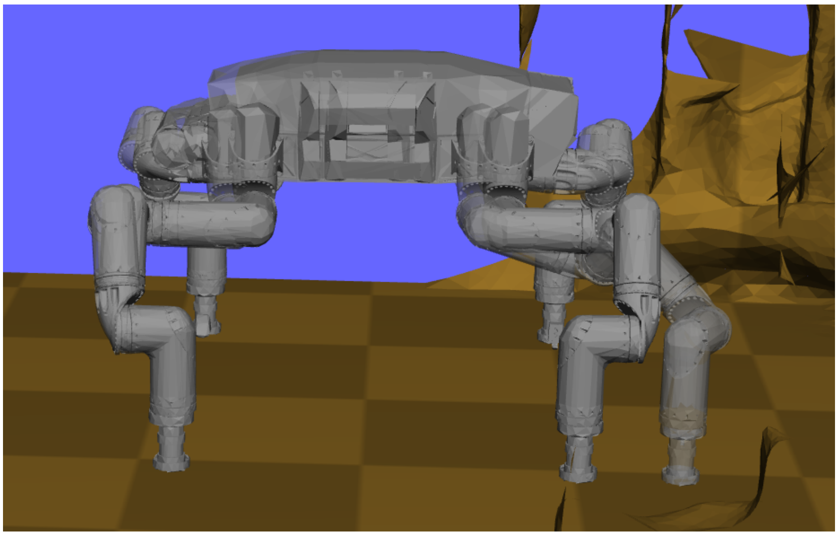
        AddToDataBase(Configuration, distance);

end

Pick configuration with the lowest distance;

Algorithm 1: Configuration Selection

FIGURE 2.1: Euclidean distance heuristic for goal selection



For the RoboSimian robot, when a leg is to be moved forward, the euclidean distance metric gives the goal configuration as shown in figure 2.1. The solid robot arm is the current configuration and the translucent robot arm is the configuration which is picked by the euclidean heuristic.

In more complex scenarios (when the configuration is close to a singularity), picking the right seed configuration to generate the Jacobian can be a non-trivial task. Picking poses by analysing the configuration space can be performed using a more informed approach. This is further discussed in chapter 3.

2.2.2 Singularity Based Configuration Selection

When the euclidean distance based configuration selection strategy fails due to proximity to a singularity, the quality metric based configuration selection is utilized. This approach is based on the idea of Yosikawa's manipulability index which describes the distance to singular configurations. The manipulability measure is given by the following:

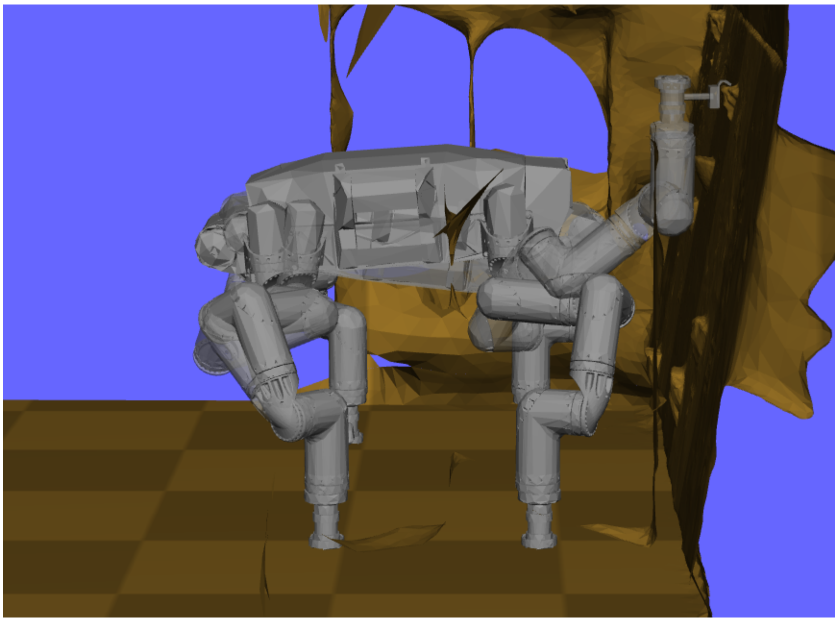
$$w = \sqrt{det(JJ^T)} = s_1.s_2.s_3.. \tag{2.3}$$

In equation 2.3, the w is calculated by considering the product of the singular values of the SVD of the jacobian matrix. In this work, I extend this work to ensure better selection of configurations. The fundamental problem with the Yoshikawa index is the possibility to compensate a large singular value with a small singular value which can lead to reduction in the manipulability index. This can have an adverse impact on the overall goal configuration selection strategy.

A solution to this problem is the to consider the strategy of maximizing the minimum singular value. This approach will ameliorate the problem of a large singular value being multiplied with a small singular value [Algorithm 2]. For the purposes of testing on the RoboSimian robot, the product of smallest two singular values are considered instead of the product of all the singular values.

The effect of this approach is more pronounced in case of the robot making large movements (such as the robot climbing a wall). The goal configuration selected using this approach for climbing problems is shown in figure 2.2

FIGURE 2.2: Euclidean distance heuristic for goal selection for climbing



A more comprehensive analysis of the results of this approach is presented in chapter 5.

Data: Robot body height, Robot Kinematics, End effector position and orientation
Result: Configuration of the limb
database=[Configuration, Singularity Based Maniplability Index];
foreach *count* $\leq$ *100* **do**
    Sample random start configurations;
    if *SolveInverseKinematicConstraints()==true* **then**
        Calculate SVD Of Jacobian Matrix;
        manipulability=Multiply smallest 2 singular values of jacobian
        AddToDataBase(Configuration, manipulability);
end
Pick configuration with the largest singularity based manipulability index;

Algorithm 2: Configuration Selection

2.3 Start Configuration Selection

One of the many unforeseen problems in case of non-anthropomorphic robots is designing their natural posture, start configuration and gait. In this section, we look into the strategy designed to pick a start configuration. The start configuration

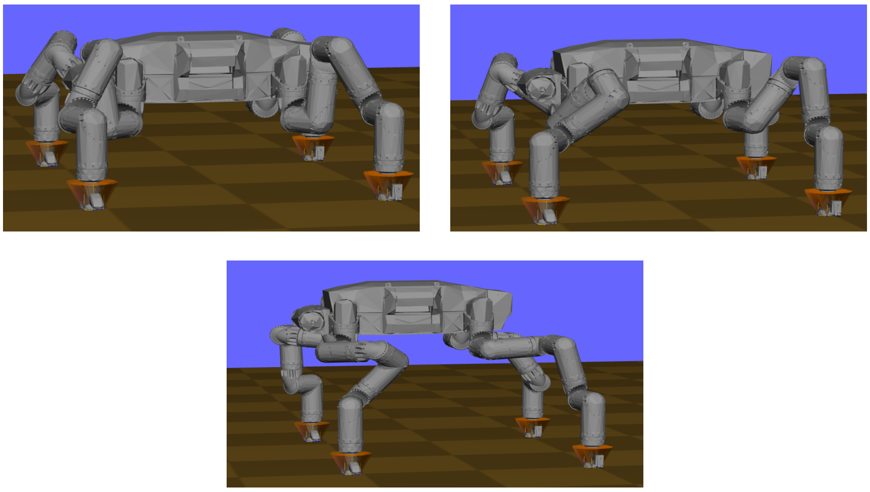


FIGURE 2.3: Start configuration selection for the RoboSimian

selection has a significant impact on the robot walking gait. Some configurations lead to walking gaits which are asymmetric due to the configuration's proximity to a singularity. This leads to more planning time to perform repetitive tasks in robots. The approach followed for the start configuration selection is described in algorithm 2. It must be noted that the limbs of the RoboSimian are symmetric. The configuration generated for one limb can be used for all the other limbs of the robot.

The algorithm to generate the start configuration involves discretizing the flat ground to solve inverse kinematics of the end effector and discretizing the height of the robot body to get a good clearance from the ground. This is followed by the singularity based configuration selection for the entire discretized space. This approach is explained in Algorithm 3.

It must be noted that this is a one time process which can be performed offline. Also, after the solution is obtained for one limb, the same solution can be mirrored to the other limbs.

Data: Robot Kinematics

Result: Start position and configuration of the end effector

Discretize the reachability region of the manipulator and height of the robot body;

database=[IK Goal, Robot Body Height, Configuration, Singularity Based Maniplability Index];

foreach *height in discretized heights* **do**

  foreach *discretized IK goal position at height* **do**

    foreach *count ¡100* **do**

      SampleRandomSeeds();;

      SolveInverseKinematicConstraints();;

      AddToDataBase(IK Goal, Robot Body Height, Configuration, SingularityBasedManipulability());;

    end

  end

end

Pick configuration and IK goal with the largest singularity based manipulability index;

Algorithm 3: Start Configuration Selection

The various configurations considered for the RoboSimain are shown in figure 2.3. The results of using these algorithms and their comparative study is provided in chapter 5.

To conclude, this chapter describes in detail, the two methods used to generate configurations using the start jacobian based method. It also extends these methods to a start configuration selection strategy. Configuration selection is employed as a step right after footstep planning. Given the footstep location and other positional constraints, the two methods described select the right configurations which helps simplify the planning process.

Chapter 3

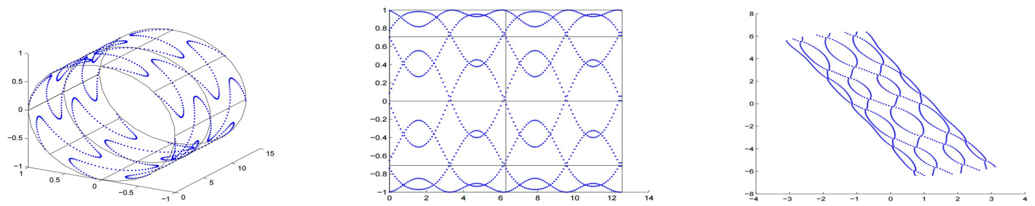
Configuration Space Analysis

Planning in high dimensional spaces can become computationally intractable even in simple cases. Due to the nature of the configuration space, it becomes difficult to find demonstrably simple paths in the task space. Current methods in analysing the configuration space involve taking slices of the configuration space by discretizing the space. This method fails to give a sense of the entire space especially in case of robots with a large number of degrees of freedom.

To analyse the configuration space for high dimensional spaces, I explored using a non-linear mapping technique called Sammon mapping, which is a very effective pattern recognition technique. This algorithm maps a data set of dimensionality d onto a non-linear subspace of m dimensions (where $m \leq d$). The one aspect of this non-linear mapping idea which makes it attractive is the distance preserving aspect in the lower dimensional manifold. This aspect can be powerful especially when using algorithms like nearest neighbor classifiers or picking seeds for the configuration selection.

3.1 Background

The simplest technique for dimensionality reduction is a straightforward linear projection approach, for example, the Principal Component Analysis(PCA). This approach maximizes the variance in the original data-set, but does not preserve complex manifolds or structures. However, the most commonly quoted example



(a) Data-set containing 8 sinusoids (b) PCA on the data-set (c) Sammon Mapping on the data-set

FIGURE 3.1: In a data-set containing sinusoids, PCA projection leads to intersections between the sinusoids (topology not preserved), Sammon mapping projection has very few intersections (although not perfect)

[16] where PCA fails is of a structure being a regular pattern over a curved manifold embedded in a high dimensional space (figure 3.1).

Work on generating a functional map of the inverse across the entire space is done by Hauser [4] to tackle the redundancy resolution problem. This work attempts to find resolution techniques where a given end effector position would yield only one configuration. This approach solves a gradient descent to perform redundancy resolution and is the basis of applying non-linear mapping for the configuration space.

One interesting work on non-linear mapping is performed by [17] where a humanoid motion planning is performed to swing a golf club using non-linear dimensionality reduction. In this approach, a Gaussian process latent variable model is created and its density function is used to generate a lower dimensional model. This lower dimensional model is used to optimize paths and the generated paths are mapped back to the pose space. This approach comes up with the notion of variance tubes which maps the smaller displacements in the pose space to the lower dimensional latent space.

The approach described in this chapter comes up with a natural way of preserving the distances in the lower dimensional space. Clustering of points in the lower dimensional space can be effective in planning smooth trajectories in the high dimensional configuration space and in the task space.

3.2 Sammon Mapping

To analyse the configuration space, we might not be interested in maximizing the variance but might be interested in preserving other aspects such as the degree to which the complex structures are preserved. Such measures, which are very essential to configuration selection and obstacle avoidance are available in a non-linear mapping strategy called Sammon mapping.

More specifically, the measure used by Sammon mapping is designed to minimize the difference between the inter-point distances in the two spaces. This is described as the transformation which conserves the distance between each pair of points. Also, this process ensures that the topology is not affected by the mapping. The effectiveness of the Sammon mapping is primarily because the function does not find a mapping from the high dimensional data set to the lower dimensional space, but to construct a lower dimensional data set which has a similar structure to that of the high dimensional data-set.

The procedure to perform Sammon mapping of the configuration space is described below. First, consider only one limb of the quadruped. Lock all the other limbs and joints to a fixed configuration for the remainder of the process. Next, discretize the reachable region of the movable end effector in the task space. Obviously, finer discretization leads to more processing time. Also, the end effector joint can be ignored as it just accounts for the position of the hook on the end effector and does not affect the overall configuration. Each configuration can now be considered as a vector (a 6 dimensional vector). Multiple samples must be considered for individual end effector position so as to ensure most redundant configurations are considered. Now, numerous 6 dimensional vectors are generated due to the sampling of the discretized space. Next, the inter-point distances d_{ij} is calculated and an error function is defined which shows how well the points the 6 dimensional configuration space fits a 2 or 3 dimensional reduced space. The error function is calculated as follows

$$E = \frac{1}{\sum_{i<j} d_{\mathrm{ij}}} \sum_{i<j}^{n} \frac{({d^*}_{\mathrm{ij}} - d_{\mathrm{ij}})^2}{{d^*}_{\mathrm{ij}}} \tag{3.1}$$

where d_{ij} is the pairwise distance between the points in the 6 dimensional configuration space and ${d^*}_{\mathrm{ij}}$ is the pairwise distance between the points in the reduced space.

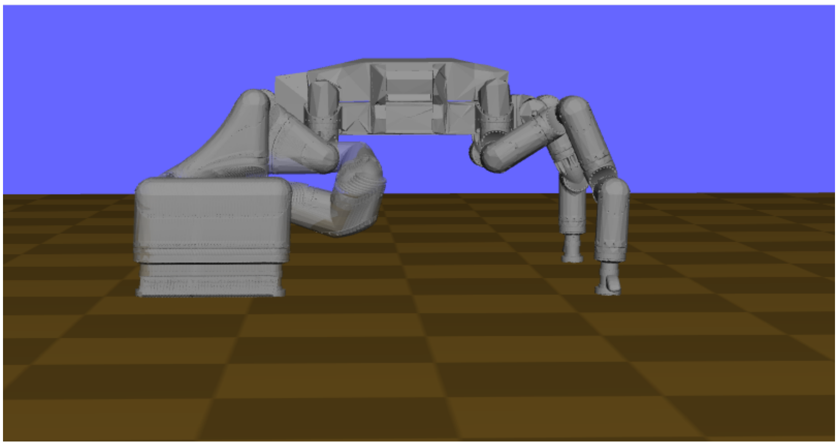
Using the steepest descent procedure, the minimum error has to be calculated so as to adjust the distance between the points in the lower dimensional space. This process leads to generation of effective reduced dimensional points in a lower dimensional space which has the same inter-point distances. One of the fundamental disadvantages of Sammon mapping is, unlike PCA, there is no algorithmic mapping for previously unseen data. So, in case a new point is to be mapped, the whole procedure is to be repeated again. This can be a significant problem if the process is used for planning paths with obstacles on the robot. However, as the proposed step is used more as an analysis tool to build repetitive motions offline or to understand the configurations and the shape of the C-space.

Sammon mapping can be intractable when the discretized space is all over the three dimensions of the task space. It is easier to consider multiple samples along a line or along an end effector trajectory. This approach can be effective in finding smooth trajectories along the end effector trajectory. Also, this process is quite useful in finding the seed configurations to calculate the jacobian matrix so that discontinuities in the inverse kinematic solution can be avoided.

3.3 Discussion

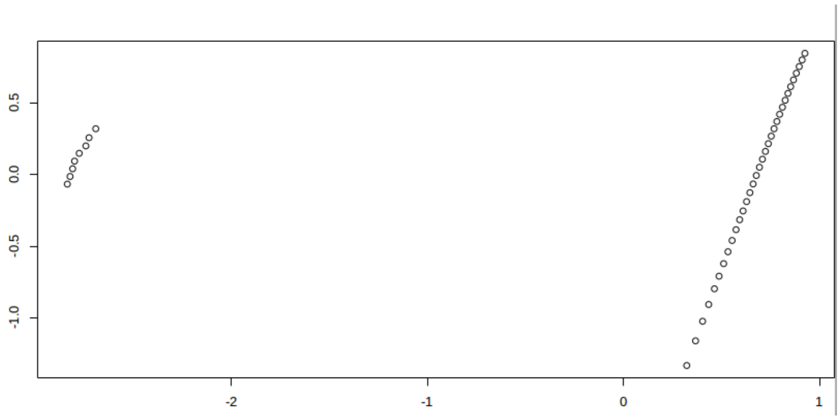
Now, to test the feasibility of a non-linear mapping strategy, we look at the sammon map of the configurations generated by moving one leg of the RoboSimain robot along the x axis. As shown in the figure 3.2, a discontinuity in the inverse kinematic solution is seen during the Jacobian based approach. When these con-

FIGURE 3.2: Lower dimensional graph of the discontiunity



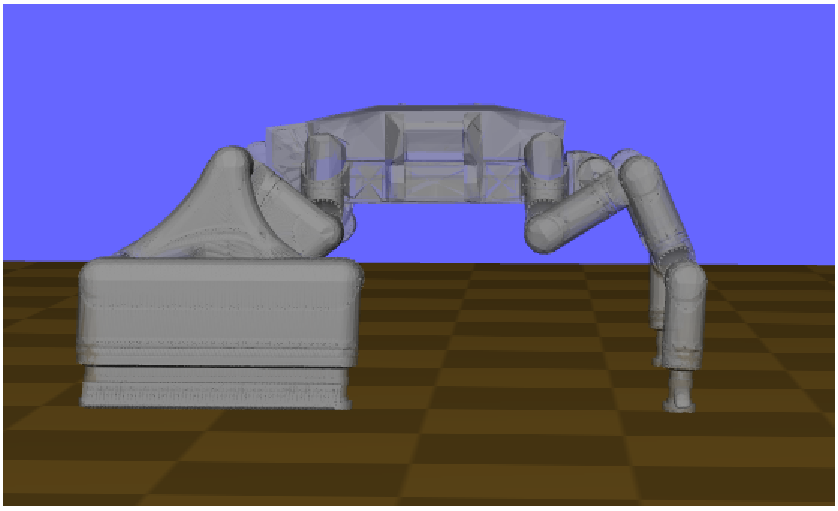
figurations are mapped onto a lower dimensional space, a clear pattern emerges. This can be seen in figure 3.3.

FIGURE 3.3: Discontinuity in the inverse kinematic solution found using the Jacobian based approach



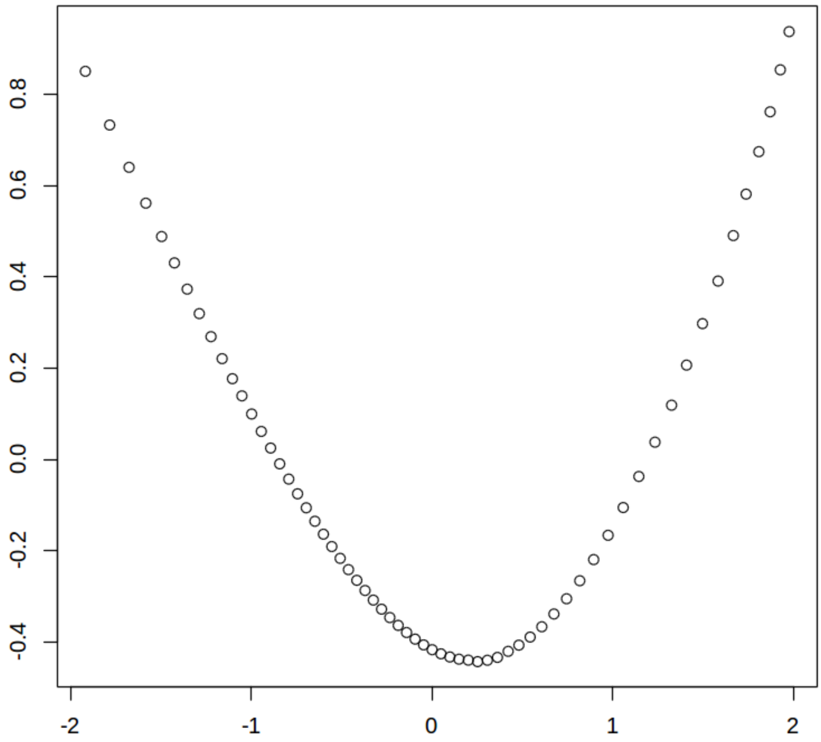
The x and y axes of this graph are just two dimensions in a lower dimensional space that is representative of the joint values. Clearly, a discontinuity is observed. Now, if the Jacobian is forced to seed a value whose lower dimensional equivalent is close to $(0, -1)$, we have a smooth solution along one axis. This is shown in figure 3.4. The sammon mapping of the same is seen in figure 3.5.

FIGURE 3.4: Discontinuity removed by seeding the sample close to the previous configuration in the configuration space



As the mapping(figure 3.5) looks like a parabola in the lower dimensional space, the discontinuity which has been caused in the image later can be ameliorated by picking a point along the parabola and mapping it back to the higher dimensional

FIGURE 3.5: Discontinuity removed by seeding the sample close to the previous configuration in the configuration space



space as the seed configuration for the Jacobian solver. This approach has been very effective in making smooth motions along a path.

Another approach has been to pick smooth trajectories given end effector position. Sammon mapping can be performed along a discretized grid of the end effector and smooth patterns in this lower dimensional space can be obtained. This lower dimensional pattern can be mapped back to the higher dimensional samples which will in turn generate smooth paths between configurations.

This approach can also be a test to check smoothness in case of convoluted designs of robots.

To summarize, this approach is very effective in visualizing the configuration space so as to pick the right configurations in a sampling based planning framework. But, as this approach is computationally quite intensive, planning in real-time using the Sammon mapping approach is not practical. Other optimization techniques have to be developed to map paths from the lower dimensional space to the configuration space.

Chapter 4

Speed Optimization For Paths

The minimum time required to execute a path has been studied since 1696 when the brachistochrone problem was posed by Johann Bernoulli. The problem involved finding the minimum time curve between 2 points for a frictionless ball . Faster trajectories allow for getting more quickly to the needed location, winning car races, intercepting enemy missiles and so on. Some of the more interesting applications of the speed optimization involve aircraft climbing, optimizing manipulator paths, generating optimal tracks for a race car etc.

This problem focuses on a small subset of trajectory generation problem where the path to the goal has already been determined. In this problem, we find a speed profile which fits the generated path. The goal is to minimize time while adhering to the path generated by the planner and avoiding collisions with the environment. Also, ensuring that the dynamics of the system do not modify the path of the planner is also important. This approach uses a high level generic planning algorithm which produces a collision free path that accomplish a task. The trajectory generator assigns the time to this generated path. This process is in contrast to the approach where the collision free paths are generated by one planner and the feasibility of the path given the dynamics are performed by the the optimization planner which generated the speed profile.

In this thesis, we concentrate on the former approach where a speed profile is fit to a path generated by a planner.

4.1 Background

This problem of time optimization is a well studied problem with a long history. Some of the early work such as the brachistochrone problem used calculus of variations to generate minimum time trajectories[18]. This method is still being used for many applications. Other promising approaches are discussed below.

Non-linear programming is a common approach for trajectory generation. This approach involves discretization of the space followed by using direct shooting methods. Some of the work on this topic are done by Betts et. al.[19] and Stryk et. al[20]. This is still quite early work in the field and was followed up with Pseudospectral methods[21] which generate a basis function to find solutions. The fundamental problem with these approaches is the speed of computation and also the effectiveness in following the trajectories of the planned path.

Other methods include graph search methods which discretize the configuration space or use sampling based approaches where the velocity profiles are determined by the nodes. These approaches include methods like potential fields, probabilistic roadmaps and more general graph search methods. These methods usually have uniform time assignment based on distance between the nodes in the configuration space which is not an effective strategy.

A slightly different approach being considered is the problem of path tracking. This mainly consists of 3 major approaches. The first type is using indirect methods which involve searching exhaustively over the task space to determine the switching points. This involves solving a planning problem using numerical methods and forward and backward integration.

The second category in this approach involves using dynamic programming methods. And the third approach involves direct conscription methods. These methods usually involve time energy optimality and other generic constraints which can be traded of with one another[22]. They include constraints like energy which make the system appear aggressive.

One of the most important works in this field is by Verscheure et. al.[23] which deals with the optimal path tracking problem with a single stage through non-linear change of variables. This approach goes beyond mere time optimality of path tracking by performing direct transcription where it proposes a convex optimization problem to minimize the time given the dynamics of the robot and the velocity constraints.

Last, Hauser[24], where the time optimization is performed by mapping a function by monotonically mapping a path configuration to time values. The problem requires the path to be continuous and twice differentiable. Also, another requirement for this function is the start and stop velocity to be zero. This process performs a piecewise quadratic time-scaling of the path function.

All these approaches work on local optimality of the the time parametrization problem. Getting a global optimal solution is not easy given these formulations. Some work on this is done by Shiller et. al.[25] where the the problem is formulated as an optimal control problem with linear system dynamics, differential states and and control inputs subject to non-linear state dependent constraints. This approach makes the problem of time optimization tractable.

It must be noted that all these methods are computationally inefficient and optimality in not a necessary condition for planning in case of legged robots. The only consideration would be to have a timing profile faster than the uniform time assignment to the interpolated path (which might be ineffective in many cases, but quite effective for legged robots). So a simple yet effective strategy would involve fitting a velocity profile which matches the constraints of the robot end effectors and dynamics of the system to effectively generate a trajectory.

This approach of generating a timing or speed profile which meets the constraints is explored in this thesis. Also, their computational simplicity and effectiveness is exploited for legged locomotion making this a heuristic based approach to trajectory generation.

4.2 Problem Formulation

The equations of motion of an n-DOF robot where torques $q \in R^{\mathrm{n}}$ can be written as a function of the joint positions, velocities and accelerations.

$$\tau = M(q)\ddot{q} + C(q, \dot{q}) + G(q)$$

Here $M(q)$ is the positive definite mass matrix,$C(q, \dot{q})$ accounts for the Coriolis terms and $G(q)$ accounts for the gravity matrix.

Now, consider a joint-space path q(s) as a function of the scalar function s. The trajectory is assumed to start at time $t = 0$ and end at $t = T$ and $s(0) = 0$ and $s(T) = 1$. s is a scaling/wrapping function which scales the path between 2 configurations based on the number of points in the interpolated path, N. Here,

$$\Delta s(t) = \frac{1}{N}$$

The time optimal planning problem can be framed as

$$\begin{aligned} \text{minimize} \quad & T \\ \text{subject to} \quad & s(0) = 0, \\ & s(T) = 1, \\ & \dot{s}(0) = \dot{s}_0, \\ & \dot{s}(T) = \dot{s}_{\mathrm{T}}, \\ & \tau = M(q)\ddot{q} + C(q, \dot{q}) + G(q) \\ & t \in [0, T] \end{aligned}$$

In most cases $\dot{s}_0$ and $\dot{s}_{\mathrm{T}}$ is 0. Now, the process involves fitting a curve which agrees to the above mentioned time optimization problem.

4.3 Trajectory Generation Strategies

In this section, the heuristic based trajectory generation strategy used for legged locomotion for the RoboSimian robot is explored. The process begins by considering the expected behavior: The legged robot must begin by moving its leg slowly followed by an increase in velocity and end its movement by slowing down. This behavior can be obtained by various curves which are capable of generating the mentioned behavior.

Consider a path $q(s)$ in the joint space where $s \in [0, 1]$. Here $q(0)$ is the start configuration and $q(1)$ is the final configuration of the trajectory of one leg. So, the parametrization for the time has to be done such that the function for time $t(s)$ gives the expected behavior.

If the parametrization for time has to be performed, a function which generates the expected behavior is needed. This function must have a steep slope close to 0 and 1 and a smaller slope in between. This can be obtained by the inverse cosine function, which shows the same behavior. However fitting just a velocity profile is not enough to create a quasi-static motion (as it might go beyond the velocity and acceleration limits of the the motors). Also the robot motion might no longer be quasi-static. The function needs to be scaled based on the total distance moved in the joint space.

In the following subsections, I present the two approaches which were used for time parametrization for generating trajectories.

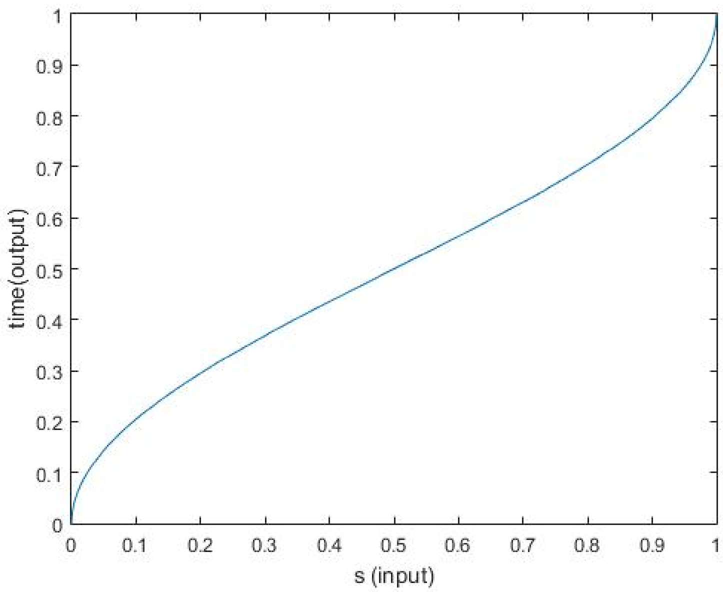
4.3.1 Heuristic Based Time Parametrization

In this parametrization technique, the inverse cosine function is relied upon. The interpolated path generated by a planning algorithm is used and the number of configurations generated by the interpolation of the path generated by the planner is taken into account. First, the time parametrization s is discretized based on the number of terms in the interpolated path. This value of s is used as the input to the time parametrization function. The function can be given as

$$t = scale * \frac{\cos^{-1}(1 - 2s)}{\pi}$$

The distance between the configurations in the configuration space is a parameter to determine the scale factor. Intuitively, larger the distance between the configurations, more time must be needed to execute the path. For walking trajectories, the scale factor can be given by the euclidean distance considered in the configuration space. The square root of the euclidean distance has been seen as a good heuristic for the scale factor. The square root of the euclidean distance for the scale factor has been obtained experimentally. Also, in case of an unsmoothend path generated by the planner, the scale factor for the heuristic might not be accurate especially if there are numerous milestones. To ameliorate this issue, this time parametrization technique can be used between two milestones. The practice of parametrization between milestones has proven to be more effective on the robot experimentally. The intuition behind using time parametrization between two milestones is due to the fact that the distance between milestones is not constant (due to the nature of the sampling based planners). In case of an unsmoothened path, the limb can move in any direction, between two milestones. To effectively nullify the effect of previous milestones in the path, coming to a halt at every milestone is the safest strategy to not involve dynamics.

FIGURE 4.1: The heuristic based trajectory generation graph for time assignment



The graph of this curve is shown in figure 4.1 which shows the expected behaviour of the leg, where the x axis represents the parameter and the y axis represents the time. This value of time is scaled based on the euclidean distance between the start and goal configurations.

The one drawback with this approach is it does not take into consideration the problem of being within the joint velocity limits. This issue has been addressed next.

4.3.2 Optimization of Heuristic Based Parametrization

The problem of using a heuristic based parametrization technique is the disregard for the velocity and acceleration limits of the robot. This can cause significant issues with path tracking and might lead to collisions as some joint values cannot keep up with the commanded joint values. To mitigate this problem, constraints are imposed on the parametrization by the heuristic based technique. This is discussed in detail below.

The process begins by first generating the heuristic based time parametrization. This heuristic based parametrization gives us the value of t_i which is considered as the ideal timing scheme. This heuristic based process acts as the input to the optimization process. The optimization process is formulated as follows:

$$
\begin{aligned}
\text{minimize} \quad & {t_\mathrm{n}}^{\text{optimization}} - {t_\mathrm{n}}^{\text{cosine}} \\
\text{subject to} \quad & s(0) = 0, \\
& s(T) = 1, \\
& \dot{s}(0) = 0, \\
& \dot{s}(T) = 0, \\
& \dot{s}(t) \leq t_\mathrm{n} * v_{\mathrm{max}}, \\
& \ddot{s}(t) \leq t_\mathrm{n} * a_{\mathrm{max}},
\end{aligned}
$$

This is a linear programming problem which can be solved effectively using well known optimization approaches. Also, it provides a more effective time optimization process by taking into consideration the joint velocities and accelerations. By constraining the end effector velocity constraints, the entire robot's time optimization can be performed.This approach has proved to be effective in generating quasi-static motions of the legged robot for walking and climbing.

To summarize, this chapter talks about generating trajectories after the path is generated by a planner. This can be considered as the final step before executing a path on the robot. Also, this finishes the whole planning process for a robot, which began with footstep planning, followed by configuration selection and planning intermediate paths.

Quantitative results for the time scaling operation and trajectory generation are provided in chapter 5.

Chapter 5

Results And Conclusions

The previous chapters gave details on the implementation of the various algorithms for configuration selection and trajectory generation using heuristic based time optimization. For an actual implementation on the robot, start configuration selection is performed first, followed by footstep planning (which is beyond the scope of this thesis). Configuration selection and path planning are performed next. The trajectory generation is the final step in the planning process.

This chapter presents the results of the configuration selection, start configuration selection strategy and trajectory generation strategies are presented. The effectiveness of the various approaches are compared. This is followed by discussions on gait analysis for the RoboSimain robot. Finally, some concluding remarks are made.

5.1 Configuration Selection

Two approaches for configuration selection are presented in chapter 2. The effectiveness of these approaches are discussed in this section. We begin by comparing the effectiveness of these algorithms on flat ground. This is followed by testing effectiveness on a rock climbing wall.

The euclidean distance heuristic approach is quite effective visually in generating small repeatable movements on flat ground. The selected configurations do not need any other sampling based planning strategies (assuming no obstacles in the planning path) as the configurations can be interpolated directly. This gives significant power to the configuration selection strategy.

The singularity based approach is also quite effective in planning paths from start configuration to the goal inverse kinematic constraint. However, visually, a smooth path is obtained even for large changes in inverse kinematic constraints. However, the penalty is the computational time (as it requires more sampled configurations and more processes such as calculating the SVD of the Jacobian).

A comparison between the euclidean distance metric and a pure Jacobian based selection is made for finding feasible paths using a pure interpolation method and a sampling based method. Also, singularity based analysis is performed for flat ground walking configurations. This is shown in table 5.1.

Flat Ground Walking			
Approach	Configuration Selection Time (seconds)	Interpolation Time (seconds)	RRT time (seconds)
Jacobian	0.01	1.3	134.7
Euclidean heuristic	0.23	1.2	8.6
Singularity	4.6	1.2	10.2

Table 5.1: Table with configuration selection and planning times for the RoboSimian robot to walk on flat ground

From this analysis, it is clear that for simple flat ground walking trajectories, although the Jacobian based configuration selection is faster, it is ineffective, primarily due to collisions or longer planning times. The euclidean based approach is the fastest with minimal overhead. Also, for all tests performed, a sampling based planning strategy was not necessary as obstacles were not present. This makes euclidean based configuration selection the most effective approach for flat ground walking.

Now, in case of more complicated goals (which are found in rock climbing problems), a comparison of the time for configuration selection using the strategies is performed in table 5.2.

This table shows the time required to plan two steps in sequence using the approaches mentioned in chapter 2. It is clear that the Jacobian approach is ineffective for planning a single step. But both the euclidean metric and the singularity based approach is effective in finding a path for one leg. But when the second step

Climbing				
Approach	Configuration Selection for first leg(seconds)	RRT time (seconds)	Configuration Selection for second leg(seconds)	RRT time (seconds)
Leg 1 followed by Leg 4(average of 5 tests)				
Jacobian	1.20	-	0.81	-
Euclidean heuristic	0.43	15.10	0.41	-
Singularity	9.19	9.40	14.20	16.30
Leg 4 followed by Leg 1 (average of 5 tests)				
Jacobian	0.40	-	1.51	-
Euclidean heuristic	0.71	52.83	1.04	-
Singularity	3.89	19.40	5.74	88.53

Table 5.2: Table with configuration selection and planning times for the RoboSimian robot to climb a rock climbing wall

is to be calculated, the euclidean metric fails to find a path within 3 minutes and the singularity based method is the only effective strategy when a path through sequence of configurations is to be found.

An interesting point to note is the time required for the robot to select a configuration for leg 4 is significantly more than leg 1. This is because of the proximity of the robot to the climbing wall which leads to a larger number of rejected samples (due to collisions), increasing the configuration selection time.

A linear interpolation approach is not effective for climbing. This is because, interpolation without way-points leads to collisions with the wall.

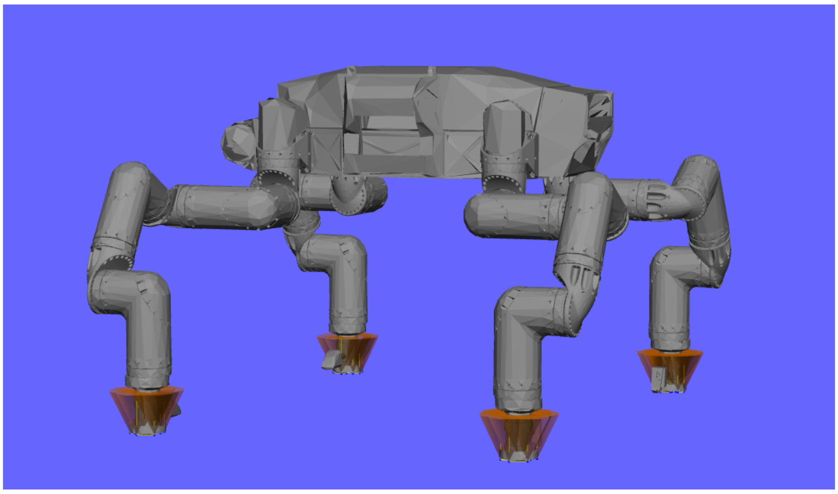
Clearly, the most important takeaway from these results is that for smaller movements, the euclidean based strategy works more effectively whereas for inverse kinematic goals farther away from the start and having an axis constraint different from the start, the singularity based approach is more effective. To generate small and repeatable motions, the euclidean metric based heuristic is the most effective.

5.2 Start Configuration Selection

Based on the tests, it was clear that start configurations are very important in generating effective paths. The impact of the start configuration for planning has been seen in generating walking trajectories for the RoboSimian. If an arbitrary start configuration is considered, configuration selection strategies fail to find a feasible repeatable path for walking on flat ground. As discussed in chapter 2, using singularity analysis, start configurations can be generated. The start configuration generated using the singularity analysis has been quite effective in generating a smooth trajectory. The effectiveness of the singularity based analysis can be shown using interpolation between the selected configurations. Assuming a random symmetric start configuration, an interpolation between start and goal configuration of a leg leads to change in a joint angle of over π radians. This is clearly not a suitable walking strategy. However, when start configurations are picked using the singularity analysis, these large changes in joint values are avoided.

The best configurations selected for the start of the robot are shown in figure 5.1.

Figure 5.1: Best start configuration for the RoboSimian



The effectiveness of selecting the start configuration is shown in figure. The first half shows an approach where the configuration is selected manually (by the programmer). The second half shows the start configuration selection using the singularity based approach. The generated path is simpler in case of the singularity based start configuration selection strategy. In case of the randomly selected configurations, the path is quite convoluted. This is shown in the RandomStartConfiguration.mp4.

5.3 Trajectory Generation

For the trajectory generation, given the path, generation of timing values so as to generate a time association to a configuration in the path is performed. This is done in two ways as described in chapter 4. The effectiveness of these approaches can only be evaluated using the time required to execute the path. The time required for taking 4 steps on flat ground using these approaches are given in table 5.3.

Time To Execute Trajectory	
Approach	Time To Execute Trajectory
Uniform Time	32.2
Inverse Cosine	14.8
Inverse Cosine with Optimization	12.0

Table 5.3: Time to execute trajectories from a given approach

Clearly, the trajectory generation using the inverse cosine interpolation followed by convex optimization is able to generate a trajectory which requires the least time to execute. This is followed by the inverse cosine interpolation approach. It must be noted that the convex optimization approach does not require any tuning of parameters to get an effective trajectory where the robot is in quasi-static equilibrium.

The inverse cosine interpolation approach does require tuning only one scale parameter and is by far more effective than the uniform trajectory generation.

The uniform trajectory generation technique generates the slowest trajectory (which is not always effective as the quasi-static equilibrium conditions are violated). This approach is not very effective for walking for legged robots.

Time To Generate Trajectory	
Approach	Time To Generate Trajectory
Uniform Time	0.3
Inverse Cosine	1.8
Inverse Cosine with Optimization	4.02

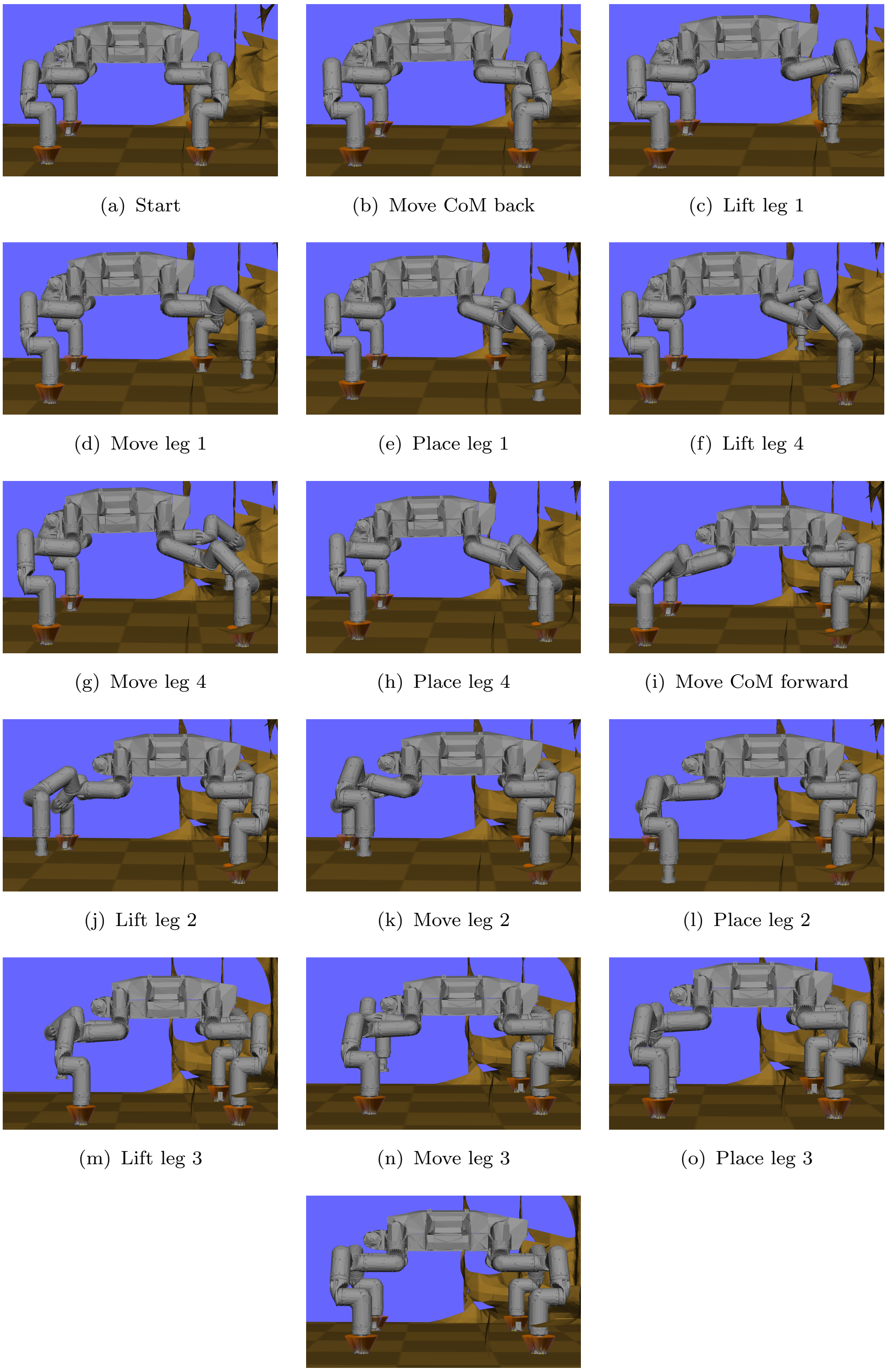
Table 5.4: Table with time to generate trajectories from a given approach

Finally, table 5.4 shows the time required to compute the trajectory for the transition from flat ground to a climbing wall. It is clear that the inverse cosine interpolation with the convex optimization technique is the most computationally expensive and requires the most time. However as the process is performed between individual milestones, the process can be parallelized. This can make the trajectory generation and execution quick.

5.4 Gait Analysis

To walk on flat ground, two variants of gaits were considered. The first gait involves moving the front legs forward followed by moving the hind legs. The second gait involves moving the two legs on the right first followed by moving the two legs on the left side. The disadvantage with the second gait is the smaller support polygon making stability during walking more complex. To simplify the planning, we concentrate more on the walking using the first gait (front legs followed by hind legs). This gait is described in detail below.

The first step is moving the CoM of the robot backwards such that it is well within the support polygon of when one of the front legs is lifted. Symmetry in the configurations makes the movement simple (the symmetry is maintained in the start configuration selection strategy). The moved configuration of the CoM is performed by the configuration selection strategies described in Chapter 2. After the CoM is moved back, leg 1 is lifted vertically (which can be done using Jacobian based approach). This is followed by moving the end effector forward by 0.23m. Heuristic based approaches for configuration selection is performed. Finally, the leg is moved down so that it touches the ground. The same process (leg lift, move and place down) is repeated for leg 4. Next, the CoM is moved forward (such that if one of the hind legs is lifted, the stability is still maintained). Finally, the same process of moving the legs is performed for the hind legs (leg 2 and leg 3). Interpolation is performed between the selected configurations to generate a path. A trajectory is generated using the heuristic based trajectory generation strategy. Walking using this gait is shown in figure 5.2.



(a) Start
(b) Move CoM back
(c) Lift leg 1
(d) Move leg 1
(e) Place leg 1
(f) Lift leg 4
(g) Move leg 4
(h) Place leg 4
(i) Move CoM forward
(j) Lift leg 2
(k) Move leg 2
(l) Place leg 2
(m) Lift leg 3
(n) Move leg 3
(o) Place leg 3
(p) Move CoM to start

FIGURE 5.2: RoboSimain Walking Gait

The same approach involving moving leg 1 and 4 followed by moving legs 2 and 3 can be used for climbing as well.

Videos of the robot walking and climbing are submitted.

5.5 Future Work

This thesis discusses the problems of configuration selection for high DoF robots and trajectory generation. Some of the future research areas to be explored which follow the work done in this thesis are discussed below.

Global Redundancy Resolution This thesis touches upon few elements of configuration selection for inverse kinematic goals. The approaches discussed in this thesis do not guarantee feasible paths to the selected configuration. A natural progression is finding a global redundancy resolution technique such that an inverse kinematic goal will always provide one solution in the configuration space and a feasible path exists to the goal configuration in a quasi-static planner(if obstacles are ignored).

Including Dynamics in Planning One of the major problems faced during the planning process for climbing the wall was the unavailability of a path where the intermediate configurations are statically stable. Some future work would include integrating dynamics into a sampling based planner where nodes in the configuration space can be connected even if they do not obey static stability conditions by integrating dynamics and trajectory generation into the configuration space planning. This makes tasks such as trotting, climbing and jumping feasible using a sampling based planner.

Footstep selection Footstep selection is still an unsolved problem for legged robots. Although footstep selection strategies exist for flat ground and rough terrain, there is little work on selecting footsteps for climbing robots. There is no clear process yet for selecting a good footstep/hold for climbing. This is an important problem to solve because if the wrong footsteps/holds are selected, the robot can get struck and not proceed in climbing the wall (this is experienced by human climbers as well).

5.6 Conclusions

This thesis addresses the problem of redundancy resolution for walking and climbing for the RoboSimian robot. It also introduces two effective trajectory generation strategies for the same tasks. However, difficulty certainly rises with more complex terrain and constraints on the robot capabilities. Planning for legged robots is difficult as the configuration space consists of manifolds of different dimensionality. The manifolds can also be overlapping one another. The work on this thesis addressed two general areas. They are:

- Redundancy Resolution
- Trajectory Generation

The approaches described in this thesis will simplify the planning process for legged locomotion.

Bibliography

[1] Y. W. Sung and D. Cho. A redundancy resolution method for humanoid robot grasping. In *2006 SICE-ICASE International Joint Conference*, pages 5888–5892, Oct 2006. doi: 10.1109/SICE.2006.315524.

[2] C. Lamperti, A. M. Zanchettin, and P. Rocco. A redundancy resolution method for an anthropomorphic dual-arm manipulator based on a musculoskeletal criterion. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1846–1851, Sept 2015. doi: 10.1109/IROS.2015.7353618.

[3] Brian W. Satzinger, Chelsea Lau, Marten Byl, and Katie Byl. Tractable locomotion planning for robosimian. *The International Journal of Robotics Research*, 34(13):1541–1558, 2015. doi: 10.1177/0278364915584947. URL http://dx.doi.org/10.1177/0278364915584947.

[4] Kris Hauser. Continuous pseudoinversion of a multivariate function: Application to global redundancy resolution. 2016.

[5] T. Yoshikawa. Manipulability and redundancy control of robotic mechanisms. In *Proceedings. 1985 IEEE International Conference on Robotics and Automation*, volume 2, pages 1004–1009, Mar 1985. doi: 10.1109/ROBOT.1985.1087283.

[6] N. Vahrenkamp, T. Asfour, G. Metta, G. Sandini, and R. Dillmann. Manipulability analysis. In *2012 12th IEEE-RAS International Conference on Humanoid Robots (Humanoids 2012)*, pages 568–573, Nov 2012. doi: 10.1109/HUMANOIDS.2012.6651576.

[7] P. Kaiser, D. Gonzalez-Aguirre, F. Schltje, J. Borrs, N. Vahrenkamp, and T. Asfour. Extracting whole-body affordances from multimodal exploration.

In *2014 IEEE-RAS International Conference on Humanoid Robots*, pages 1036–1043, Nov 2014. doi: 10.1109/HUMANOIDS.2014.7041332.

[8] A. M. Ghalamzan E., N. Mavrakis, M. Kopicki, R. Stolkin, and A. Leonardis. Task-relevant grasp selection: A joint solution to planning grasps and manipulative motion trajectories. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 907–914, Oct 2016. doi: 10.1109/IROS.2016.7759158.

[9] A. De Luca, G. Oriolo, and P. R. Giordano. Kinematic modeling and redundancy resolution for nonholonomic mobile manipulators. In *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, pages 1867–1873, May 2006. doi: 10.1109/ROBOT.2006.1641978.

[10] A. H. Quispe and M. Stilman. Deterministic motion planning for redundant robots along end-effector paths. In *2012 12th IEEE-RAS International Conference on Humanoid Robots (Humanoids 2012)*, pages 785–790, Nov 2012. doi: 10.1109/HUMANOIDS.2012.6651609.

[11] G. Oriolo and C. Mongillo. Motion planning for mobile manipulators along given end-effector paths. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pages 2154–2160, April 2005. doi: 10.1109/ROBOT.2005.1570432.

[12] G. Oriolo, M. Ottavi, and M. Vendittelli. Probabilistic motion planning for redundant robots along given end-effector paths. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 2, pages 1657–1662 vol.2, 2002. doi: 10.1109/IRDS.2002.1043993.

[13] F. Zacharias, C. Borst, and G. Hirzinger. Capturing robot workspace structure: representing robot capabilities. In *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3229–3236, Oct 2007. doi: 10.1109/IROS.2007.4399105.

[14] H. Kim, L. M. Miller, N. Byl, G. M. Abrams, and J. Rosen. Redundancy resolution of the human arm and an upper limb exoskeleton. *IEEE Transactions on Biomedical Engineering*, 59(6):1770–1779, June 2012. ISSN 0018-9294. doi: 10.1109/TBME.2012.2194489.

[15] Z. Li, K. Gray, J. R. Roldan, D. Milutinovi, and J. Rosen. The joint coordination in reach-to-grasp movements. In *2014 IEEE/RSJ International*

Conference on Intelligent Robots and Systems, pages 906–911, Sept 2014. doi: 10.1109/IROS.2014.6942667.

[16] Paul Henderson. Sammon mapping.

[17] Hyuk Kang and F. C. Park. Humanoid motion optimization via nonlinear dimension reduction. In *2012 IEEE International Conference on Robotics and Automation*, pages 1444–1449, May 2012. doi: 10.1109/ICRA.2012.6225043.

[18] W. Charlton and C. Chiarella. An optimization example: The brachistochrone problem. *Proceedings of the IEEE*, 61(12):1760–1761, Dec 1973. ISSN 0018-9219. doi: 10.1109/PROC.1973.9368.

[19] J. Betts. *Practical Methods for Optimal Control and Estimation Using Nonlinear Programming*. Society for Industrial and Applied Mathematics, second edition, 2010. doi: 10.1137/1.9780898718577. URL http://epubs.siam.org/doi/abs/10.1137/1.9780898718577.

[20] Oskar von Stryk. *Numerical Solution of Optimal Control Problems by Direct Collocation*, pages 129–143. Birkhäuser Basel, Basel, 1993. ISBN 978-3-0348-7539-4. doi: 10.1007/978-3-0348-7539-4_10. URL http://dx.doi.org/10.1007/978-3-0348-7539-4_10.

[21] Xiao Guo and Ming Zhu. Direct trajectory optimization based on a mapped chebyshev pseudospectral method. *Chinese Journal of Aeronautics*, 26(2):401 – 412, 2013. ISSN 1000-9361. doi: http://doi.org/10.1016/j.cja.2013.02.018. URL http://www.sciencedirect.com/science/article/pii/S1000936113000307.

[22] kos Nagy and Istvn Vajk. Lp-based velocity profile generation for robotic manipulators. *International Journal of Control*, 0(0):1–11, 0. doi: 10.1080/00207179.2017.1286535. URL http://dx.doi.org/10.1080/00207179.2017.1286535.

[23] D. Verscheure, B. Demeulenaere, J. Swevers, J. De Schutter, and M. Diehl. Time-optimal path tracking for robots: A convex optimization approach. *IEEE Transactions on Automatic Control*, 54(10):2318–2327, Oct 2009. ISSN 0018-9286. doi: 10.1109/TAC.2009.2028959.

[24] Kris Hauser. Fast interpolation and time-optimization with contact. *The International Journal of Robotics Research*, 33(9):1231–1250, 2014.

doi: 10.1177/0278364914527855. URL http://dx.doi.org/10.1177/0278364914527855.

[25] Z. Shiller. Time-energy optimal control of articulated systems with geometric path constraints. In *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*, pages 2680–2685 vol.4, May 1994. doi: 10.1109/ROBOT.1994.350931.