

DISSERTATION

APPLICATION OF HOMOTOPY CONTINUATION METHODS

Submitted by
Daniel Abram Brake
Department of Mathematics

In partial fulfillment of the requirements
For the Degree of Doctor of Philosophy
Colorado State University
Fort Collins, Colorado
Fall 2012

Doctoral Committee:

Advisor: Vakhtang Putkaradze
Coadvisor: Tony Maciejewski

Dan Bates
Mario Marconi

ABSTRACT

APPLICATION OF HOMOTOPY CONTINUATION METHODS

Abstract goes here.

Daniel Abram Brake
Department of Mathematics
Colorado State University
Fort Collins, Colorado 80523
Fall 2012

Contents

1	Introduction	1
2	Homotopy Continuation	2
2.1	Theory, <i>Bertini</i>	2
2.2	<i>Paramatopy</i>	2
2.2.1	High Performance Computing and <i>Paramatopy</i>	2
3	Intrinsic Localized Modes	3
3.1	Introduction	3
3.2	Setup of the Problem	3
3.3	Results	6
3.4	History and Motivation	7
3.5	Single Molecule Detection through Intrinsic Localized Modes	7
3.6	Using <i>Bertini</i> to Find ILM	7
4	Cooperative Robotic Workspaces	9
4.1	Formal Problem Statement	10
4.2	Homotopy Continuation	11
4.3	Workspace Computation	12
4.4	2D case: Two link planar robots	14
4.5	3D Case: Three Joint Manipulators	17
4.6	Conclusion	22
4.7	eqns	23
5	Conclusion	24

Chapter 1

Introduction

This is the introduction to my dissertation. Why homotopy continuation is awesome. Some theoretical background. Discussion of ILM, Robots, Supercomputing.

Chapter 2

Homotopy Continuation

2.1 Theory, *Bertini*

2.2 *Paramatopy*

2.2.1 High Performance Computing and *Paramatopy*

Chapter 3

Intrinsic Localized Modes

3.1 Introduction

The existence of the Intrinsic Localized Modes, or ILMs, has been demonstrated in many coupled systems of oscillators. These coherent oscillations were first discovered in [41, 31]. ILMs have then been found in a wide variety of structures, and there has been substantial interest in the theoretical analysis and practical applications of these structures [21, 23, 34, 36, 38, 22, 35, 19, 39, 24, 25, 37]. In general, most of these works have concentrated on the analysis of the oscillations being described by one variable that is perhaps allowed to be complex. In particular case of ILM in cantilever arrays, the one-dimensionality of vibrations was enforced by designing the experiments with the cantilever thickness in the direction of vibration to be much smaller than in the other direction. However, modern application of ILMs to areas like sensing require miniaturization of the nanopillar arrays to sub-micron levels [40]. Cantilever arrays of this size are fabricated using techniques like nanoheteropitaxy [18], typically producing pillars with similar transversal sizes, which thus do not typically possess a preferred direction for vibration. Driving these cantilever arrays at their resonant frequencies typically excites vibration modes with deflection in both directions. The goal of this paper is to analyze the appearance of ILMs in the case when oscillations in both directions are allowed. The two-dimensional motion of the pillars is similar to having two coupled oscillator arrays, or an array of oscillators with an internal degree of freedom. While there has been considerable amount of work in this area already, see for example [29, 44, 14], the difference between our work and previous studies is in the structure of elastic coupling between two directions of oscillation.

3.2 Setup of the Problem

For simplicity, in this paper we assume that each crystalline pillar in the array has a square cross-section, with the flat sides being parallel to the axis of the crystal, and the pillar material is such that crystalline axes are parallel to the pillars' sides and perpendicular to each other. These cantilever arrays, when driven by an external vi-

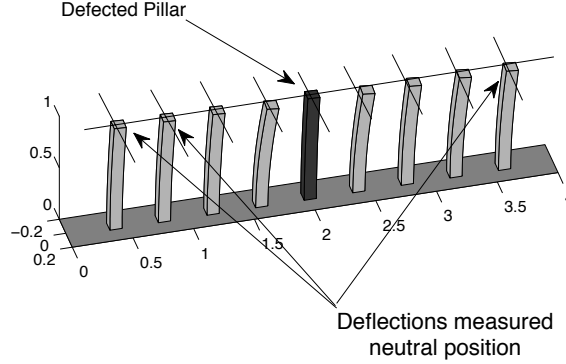


Figure 3.1: A cartoon of one dimensional cantilever array that is able to undergo vibrations in two directions. For illustration, we have also drawn a defected pillar that is used to pin the ILM appearing from random initial condition on the center pillar.

bration source, function as coupled oscillators. The material properties determine the governing differential equations, which are a generalization of the nonlinear equations for pillar vibrations derived earlier. In order to reliably pin the ILM to the desirable location (center pillar), we introduce a small defect in that pillar's properties, and make sure that the ILM is attractive to the defect. The setup for the problem of interest is shown on Fig. 3.1. In an experiment with sub-micron nanopillar arrays, the forcing will most likely be distributed among many pillars and artificial preparation of an ILM will be experimentally unfeasible. Thus, in all our simulations, we have analyzed ILMs that spontaneously and robustly appear from random initial conditions, under the influence of distributed forcing.

The equations of motion for the pillars are formulated as follows. Let u_k be the deflection of the k -th crystal pillar from its neutral position. A typical equation, describing the evolution of one dimensional deflection in a one-dimensional array is given by the equation

$$\ddot{u}_k = -\alpha_1 u_k - \alpha_2 ((u_k - u_{k-1}) + (u_k - u_{k+1})) - \beta_1 u_k^3 - \beta_2 ((u_k - u_{k-1})^3 + (u_k - u_{k+1})^3) \quad (3.1)$$

Here, $\alpha_1, \alpha_2, \beta_1, \beta_2$ are the constants that are determined by the material properties of the pillars as well as the geometry. In order to describe the two-dimensional deflection of the pillars, we introduce two components of the pillar deflection as $\mathbf{u}_k = (u_{k,x}, u_{k,y})$ that are not necessarily aligned with the crystal axes. The linear components of the stress can be computed as follows. Suppose for the given deflection, the linear part of deformation energy is given by the quadratic form $E_2 = \frac{1}{2} \mathbf{u}_k^T Q_2 \mathbf{u}_k$. For the purpose of this paper, we choose a parameterization of the symmetric matrix Q_2 with three

parameters $\alpha_{1,x}$, $\alpha_{1,y}$ and α_3 as follows:

$$E_2 = \frac{1}{2} \sum_k \alpha_{1,x} u_{k,x}^2 + \alpha_{1,y} u_{k,y}^2 + \alpha_3 (u_{k,x} - u_{k,y})^2. \quad (3.2)$$

For the purposes of simplified analysis in this paper, we only consider $\alpha_{1,x} = \alpha_{1,y} = \alpha_1$ which we treat as a parameter. Then, we study the behavior of the system as a function of parameters α_1 and α_3 . Note that this is somewhat different from the standard normalization, where one of the natural frequencies would be normalized to 1. Also, it is important to notice that the eigenvalues of the Hessian of this matrix are strictly positive and distinct for $\alpha_1 > 0$ and $\alpha_3 > 0$ so there are no degeneracies or unphysical values of parameters in the system.

The nonlinear (cubic) term in the equation leads to the fourth order term in energy, described by a fourth order tensor Q_4 , *i.e.*, $E_4 = Q_4 \cdot \mathbf{u}_k \cdot \mathbf{u}_k \cdot \mathbf{u}_k \cdot \mathbf{u}_k$. Even when proper symmetries are included, the number of non-zero components of Q lead to the exceedingly large number of parameters. It is possible to estimate some of these components analytically if the information about the orientation of crystalline axes, pillar shape and nonlinear elasticity is known. This will be done in further studies; for the purpose of this work, we shall consider a simpler particular case of the nonlinear coupling energy as

$$E_4 = \frac{1}{4} \sum_k \beta_2 (u_{k,x}^4 - u_{k,y}^4) + \beta_3 (u_{k,x} - u_{k,y})^4. \quad (3.3)$$

The total potential energy is then $E = E_2 + E_4$. For symmetry reasons, in the case considered here there is no cubic term in the energy. However, note that a cubic term may appear for a general arrangement of the crystal axes and pillar facets, in the geometries breaking the reflection symmetry of the system. The appearance and role of a cubic term in energy, leading to quadratic terms in the equations, is very interesting and will be addressed in further studies. This will lead to the necessity of investigation of a large number of parameters, which we are not going to do here. The corresponding equations of motion for two directions are

$$\begin{aligned} u_{k,x}'' &= -\alpha_{1,x} u_{k,x} - \alpha_2 ((u_{k,x} - u_{k-1,x}) + (u_{k,x} - u_{k+1,x})) \\ &\quad - \beta_1 u_{k,x}^3 - \beta_2 ((u_{k,x} - u_{k-1,x})^3 + (u_{k,x} - u_{k+1,x})^3) \\ &\quad - \alpha_3 (u_{k,x} - u_{k,y}) - \beta_3 (u_{k,x} - u_{k,y})^3 \\ &\quad - \gamma \dot{u}_{k,x} + \sigma \sin(t) \end{aligned} \quad (3.4)$$

$$\begin{aligned} u_{k,y}'' &= -\alpha_{1,y} u_{k,y} - \alpha_2 ((u_{k,y} - u_{k-1,y}) + (u_{k,y} - u_{k+1,y})) \\ &\quad - \beta_1 u_{k,y}^3 - \beta_2 ((u_{k,y} - u_{k-1,y})^3 + (u_{k,y} - u_{k+1,y})^3) \\ &\quad - \alpha_3 (u_{k,y} - u_{k,x}) - \beta_3 (u_{k,y} - u_{k,x})^3 \\ &\quad - \gamma \dot{u}_{k,y} + \sigma \sin(t) \end{aligned} \quad (3.5)$$

This functional form of directional coupling allows a comprehensive study of ILM formation for two parameters, α_3 and β_3 . In addition, we have added the dissipation

in the pillars, described by the term $\gamma \dot{\mathbf{u}}_k$, and the forcing term, proportional to σ . The coupling parameters α_3 and β_3 can be estimated numerically, but their precise value for a given experiment is generally unknown. Thus, they must be treated as parameters in the problem.

3.3 Results

Here, we present the results for spontaneous formation of ILMs in the system. The values of other parameters are presented in the table below. For completeness, we present the values of the other parameters used in simulations below. While the typical value of α_1 are around 1, we have chosen to scan a large set of values in order to accommodate both cases when there is large discrepancy in natural frequencies in two directions, and, alternatively, when the natural frequencies of vibration are very close to each other.

Parameter	Value
α_1	0.0001 ... 1
α_3	0.0001 ... 1
β_1	0.01
β_2	0.0001
β_3	0, 0.001
γ	0.001
σ	0.01

The results of our simulations are presented on Fig. 3.2 and Fig. 3.3. We plot the ILM detectability, *i.e.*, the energy concentrated in an ILM divided by the average energy of a pillar, which is non-zero due to the competition between the forcing and dissipation. Each point on this plot is the result of simulation until $t = 300$, with the amplitude of ILM computed from the last half of the time interval. The horizontal and vertical axes are α_3 and α_1 , respectively. Red areas represent high values of detectability, and blue represent the low values. As we see, there are areas of parameter where ILM appearance is very robust; however, these areas are relatively small. There is a highly intricate pattern describing high detectability of ILMs, and it is more typical to observe the energy concentration in ILM to be rather small, as is evident from our results. Thus, the preliminary work presented here warrants more detailed studies of ILM formation and detectability, and the necessity of improved design of nanopillar arrays for nanotechnology applications.

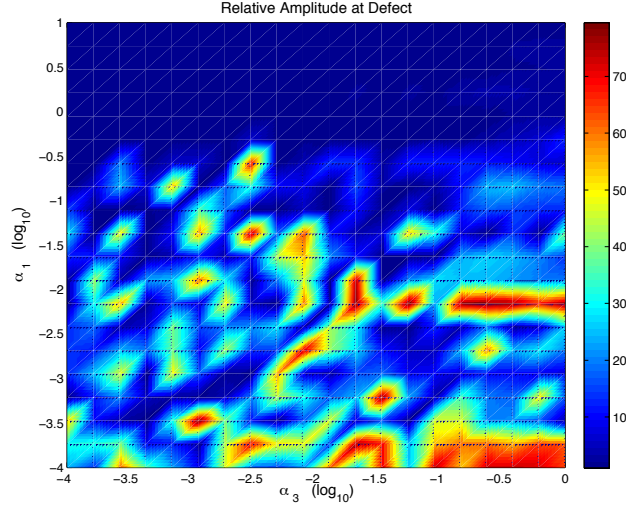


Figure 3.2: Phase diagram for ILM presence in 9-pillar crystal array. Coupling parameters α_1 and α_3 are examined for $\beta_3 = 0$. Red denotes strong ILM formation whereas blue indicates practically no spontaneously forming ILMs present in the system. In order to scan large areas of parameters we use logarithmic scale for α_1 and α_3 .

3.4 History and Motivation

3.5 Single Molecule Detection through Intrinsic Localized Modes

3.6 Using *Bertini* to Find ILM

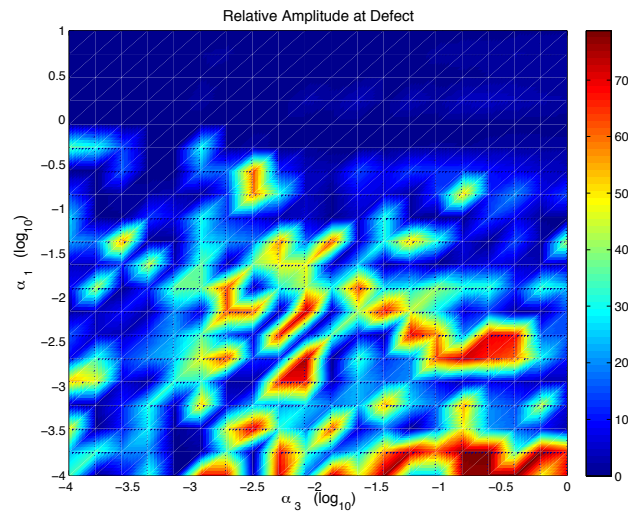


Figure 3.3: Same phase diagram as in Figure 3.2 but computed for $\beta_3 = 0.001$

Chapter 4

Cooperative Robotic Workspaces

Fault-tolerance and robustness play important roles in the design of autonomous systems, including robotic arms. In this paper, we consider autonomously operated robotic systems that are deployed into a hazardous or remote location, such that the repair on a failed joint is impractical or impossible. If the system has to operate for an extended period of time, we want the system to operate to the best of its remaining physical capability after a joint failure.

Suppose such a system possesses several robotic arms for various tasks. While no robot can be completely fail-proof, under normal operating conditions the probability of several joints failing simultaneously is low. Thus, we are considering only one joint failure at a time. We also assume the robots cannot change their position with respect to one another. Then, should one joint in one arm fail, the presence of a second arm may open the way to preserve some of the workspace of the failed arm. In particular, if the second arm can grasp the first, perhaps some of the lost workspace of the first arm can be restored.

Immediately, two questions arise: *a)* What is the best placement of the robots in relation to each other? *b)* Is there a best place for the second arm to grasp the first? We should mention here that the answer to both questions clearly depends on the measure of the post-failure workspace, which is problem dependent. In this paper, we use the Euclidean measure in \mathbb{R}^3 , corresponding to the embedding of the workspace, to quantify the size of the workspace. In general, the user suitably introduces a measure corresponding to the robot's embedding space including position and orientation, and problem type; other measures (for example, closeness to a target) will change the details of the optimization procedure, but the basic structure of the solution process should remain unchanged. Our focus in this paper is to find the optimal separation and grasp point for one robot to grasp another in the event of a free-swinging joint failure, so as to maximize the area of the post-failure workspace. To solve this problem, we use the method of homotopy continuation. In particular, the inverse kinematics problem can be cast as a polynomial system, and homotopy continuation provides a means for efficiently producing numerical approximations of *all* isolated complex and real-valued solutions of the polynomial system.

Consideration of fault-tolerance is essential in dangerous locations, and has been studied extensively. Difficulties arising from considerations of reliability and safety

were pointed out in the review [15]. The detection and isolation of faults, in various components of a robot, such as sensors, controllers, and actuators, as well as collision with environmental elements, was treated in [3, 20, 27, 7]. Anticipation of failures has also been explored considerably; decomposing a task into primary and secondary goals is one method of dealing with these problems. Prioritization of tasks subject to other constraints (*e.g.*, environmental) was treated in [9, 26], while paper [13] studied the anticipation of free-swinging joint failures. The failure-tolerant region of workspace was defined in [16], which also gave a method for computing a fail-tolerance measure which allows a robot to operate within this fail-tolerant region in real time. Other measures of fail-tolerance include analysis of the singular values of the Jacobian matrix for a robot [28] and a ‘manipulability index’ as defined in [33]. Robots can be designed and operated with fail-tolerance in mind, such as dual actuators at joints [17], kinematic redundancy [30, 17, 48, 49, 32], and reconfigurability [1]. Determining the reliability of a robot via fault-tree analysis appeared in [8], which allows to quantify weaknesses in design.

The key feature of this manuscript is the application of methods of algebraic geometry to the description of fault tolerance of two cooperating robots. These techniques are guaranteed to give all isolated solutions to algebraic equations describing the positions of robotic arms, and thus robustly find solutions in arbitrarily complex settings, such as multiple roots and joint values in isolated regions of joint-space. Thus, these techniques are capable of *autonomously* completing the workspace analysis, which can be useful in robots operating with high degree of independence, with little or no communication with the operator. The trade-off for the completeness of analysis is the computational complexity as compared to *e.g.* the Jacobian control methods, as we show below. If extra information about the workspace, such as the exact number of isolated domains and multiplicity of solutions is available, then traditional methods will be advantageous computationally. If no such *a priori* information is available, the methods we describe here should provide precise information about the isolated domains and solution multiplicity which, if desired, can be further refined by Jacobian control methods.

Methods of algebraic geometry have been applied in the kinematic description of robots as well, see for example [43, 42, 46, 47, 45]. However, analysis of the cooperation of multiple arms via the tools of numerical algebraic geometry has never been explored. In the next section, we provide a formal statement of the general problem we are considering. Background regarding the numerical solution of polynomial systems may be found in Section 4.2. Details about our method are described in Section 4.3, while Sections 4.4 and 4.5 respectively contain two and three dimensional examples.

4.1 Formal Problem Statement

Given two articulated arms, we seek to optimize the placement of grasping sockets to maximize the post-failure workspace (we assume a socket attachment mechanism of the first arm to the second as considered in [1]). That is, if one robot has a free

swinging joint failure [11, 12], we would like to ensure that when the functional robot joins at the socket, the resulting cooperative workspace is as large as possible.

Each robot has its own independent workspace, (let us call them W_1, W_2), so when the two robots are placed near enough to each other, there is an intersection workspace $W_\cap = W_1 \cap W_2$. In a grasping configuration, there is a post-failure workspace W_f , which contains all remaining workspace, if robot 2 attaches to the socket location on robot 1. It is clear that $|W_\cap|$ depends on the separation between the robots, as well as their orientation, as does $|W_f|$. Note that the intersection workspace does not depend on the grasping point, as it reflects the workspace prior to failure for each robot. In both pre- and post-failure workspaces, we take into account joint limits, which are an important consideration for practical applications. Indeed, robots typically have limited range of movement for each joint, which reduces the size of all workspaces. These limits introduce a dependence on the relative rotations of the robots to each other, to the measures of the various workspaces.

An optimal configuration of two cooperating robots would include: the separation between the cooperating arms, the relative orientation of the bases, and the location of the grasping point for each link. In order to give quantifications about the post-failure workspace size and consequently find such an optimal configuration, we introduce an objective function, which we seek to maximize. There are many possible objective functions one can imagine, and the right choice depends on the application. In this paper, our objective function to be maximized is a weighted sum of $|W_f|$, $|W_1|$, and $|W_\cap|$:

$$\Omega = (1 - \lambda) \cdot |W_f| + \lambda(|W_1| - |W_\cap|). \quad (4.1)$$

We have chosen (4.1) for an example objective function in order to balance between the benefit of having a second robot, and the maximization of the post-failure workspace. A configuration which imparts entirely distinct workspaces would maximize the term $\lambda(|W_1| - |W_\cap|)$, but W_f would be an empty set. Contrarily, we might find a configuration which results in full restoration of workspace W_f upon entering grasping stance, but which makes the pre-failure workspaces overlap greatly, so $\lambda(|W_1| - |W_\cap|)$ would be small. We would prefer to balance between pre- and post-failure benefits of having two robots, and (4.1) is one way of doing this.

4.2 Homotopy Continuation

We recast the inverse kinematic problems for computation of (4.1) in the form of polynomial systems, and use the methods of numerical algebraic geometry, particularly homotopy continuation, to find the solutions of these polynomial systems. Given a set of polynomials f_1, \dots, f_N in N variables for which we seek the solutions, homotopy methods begin by choosing and solving some other (related) polynomial system g_1, \dots, g_N for which the solutions are easily found, e.g., polynomials such as $x^6 - 1$. By varying the coefficients of \bar{g} to those of \bar{f} , mathematical theory guarantees that, with probability one, the solutions will vary continuously, thus forming *solutions curves* or *paths* from the solutions of \bar{g} to those of \bar{f} . These solution curves may then be

tracked numerically with standard predictor/corrector methods, such as a combination of RK45 and Newton’s method. Further details may be found in [2, 43].

There is a setting in which homotopy methods are particularly effective and of which we make use in this paper. If instead of solving one polynomial system \bar{f} we wish to solve a large number of polynomial systems that differ only in their coefficients (i.e., we wish to solve $\bar{f}(\bar{p})$, where \bar{p} is some set of parameters), there is an especially efficient homotopy method known as a *parameter homotopy*. The key of the method is to have a simple system \bar{g} for $\bar{p} = \bar{p}^*$, and find a path in the parameter space leading to the system of interest as $\bar{p} = \bar{p}_0$. When constructing a starting system \bar{g} , it is possible that there will be many more solutions of \bar{g} than \bar{f} . As a result, it may occur that hundreds of thousands of solution curves are tracked in order to find only a few solutions. However, in the special case of a parameter homotopy, we first solve $\bar{f}(\bar{p})$ at a single instance of \bar{p}^* (typically chosen as random complex numbers, for theoretical reasons). This stage may involve a number of wasted paths. All other instances of $\bar{f}(\bar{p})$ may then be solved by simply following the handful of finite solutions from \bar{p}^* to any other choice of \bar{p}_0 . For example, for an initial, random complex parameter choice, we may need to follow thousands of paths to find the six solutions of interest. Then, for all other points in the parameter space, it suffices to follow just six paths. Again, there is much theory and detail underlying these methods, most of which may be found in [43]. During the process of homotopy continuation, a certain number of paths will fail as they near a singularity in parameter space. In the context of robotic workspaces, these failed paths indicate proximity to workspace boundary or kinematic singularity. Right now, we are not using this information, and simply ignore failed paths; however, that property could ultimately be used for more accurate and efficient prediction of, for example, workspace boundaries.

For this paper, it is enough to accept homotopy continuation as a numerical method that will provide approximations to all isolated solutions of a polynomial system. Several software packages are available for these sorts of computations. We use a freely-available software package named *Bertini* [5], which has been under development for the past decade by the second author, J. Hauenstein, A. Sommese, and C. Wampler. The repeated calls to *Bertini* were parallellized for efficiency using the method described in [4].

4.3 Workspace Computation

We compute the initial pre-failure workspaces for each robot via a random sampling method. We sample a subset of the space which embeds the workspace, that is guaranteed to contain the workspace, using a uniform pseudo-random distribution with the built-in MATLAB random number generator, solving the inverse kinematics problem at each of these sample points, as follows.

The equations for the inverse kinematics problem are first solved via a standard homotopy run at \bar{p}^* , a random point in complex parameter space. All subsequent runs at parameter points in the workspace are treated as parameter homotopies, and all such runs begin at $\bar{p} = \bar{p}^*$. Because *Bertini* will find all complex solutions to a

polynomial system, the samples that result in real inverse kinematic solutions are inside the workspace, while samples that gave only complex solutions are outside the workspace. If we have r_i samplings, and r_f of these have real solutions, we can estimate the (Euclidian) measure of the workspace as the product of the volume of the sampled box, and the ratio r_i/r_f .

We shall note that the inverse kinematics equations for a robot with rotary joints are trigonometric in nature. In order to use homotopy continuation, we write the trigonometric equations in a polynomial form by treating each sine and cosine pair as a separate variable, mapping $\cos(\theta_i) = c_i$ and $\sin(\theta_i) = s_i$. Then, trigonometric variables can be recast into algebraic conditions using $c_i^2 + s_i^2 = 1$.

For the case of $N \geq 3$ joints working in 3 dimensions without orientation, there are 6 inverse kinematic equations in $2N$ variables when computing W_i . As long as the number of joints equals the number of degrees of freedom in the ambient workspace, *Bertini* will find all solutions, and we will be able to measure $|W_i|$. For kinematically redundant robots, W_i is a set of higher $2N - 6$ dimensional manifolds, which could be described by defining a mesh of the same dimensionality as coordinates on the joints. Our method readily applies to that higher dimensional problem as well. However, the issues we are facing are related to the curse of dimensionality, and hence in the computation of the data as a whole, not in the solution finding at a particular point in space, which is fast. For example, with $N = 5$ joints, and a 3 dimension workspace, each point in the workspace is a two dimensional manifold in joint space. To cut down the manifold to be 0-dimensional for solving via *Bertini*, we could sample each of the two redundant joints' values. Combinatorial issues arise as well, if we wanted to solve each spatial sample for each possible combination of redundant joints. The work load increases dramatically with the growth of dimensionality for both the robot and ambient workspace. Another problem is whether to use random spatial sampling or a mesh, and how to optimize the number of samples. The optimal sampling will depend on the geometry of the manifolds, and has to be studied in more detail elsewhere.

A similar computation is performed to find an estimate of $|W_\cap|$. Because we know that $W_\cap \subset W_2$, we simply solve the inverse kinematics of robot 1 at the r_f points that gave real solutions for W_2 . Suppose that of r_i initial samples, r_f gave real solutions for W_2 , and r_k correspond to a real solution belonging to W_1 , then we can estimate the intersection work space to have measure

$$|W_\cap| \approx \frac{r_f}{r_k}.$$

The final component for calculating the objective function is to estimate the post-failure space $|W_f|$. Note that there is an associated space for each link of the failed robot, as we could potentially join a specially placed socket on each link. We know that $W_f \subset W_1$, so as for W_\cap , we need only consider the r_f points we computed in W_1 . The major difference between the computation of $W_{1,2}$ and W_f is the number of equations which is twice as high for the W_f computation.

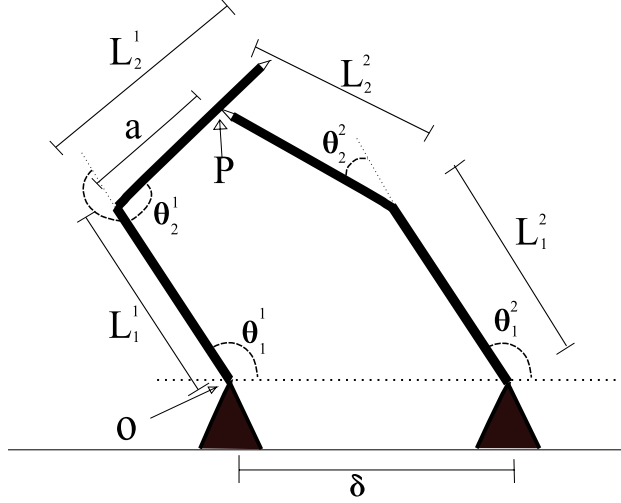


Figure 4.1: Two robots touching at point P . This is a simplified 2D model of a more general scenario. In this paper we will not consider the final arm orientation.

4.4 2D case: Two link planar robots

We start with the illustration of the method for a two dimensional example that is shown in Figure 4.1. We have chosen this example because almost all inverse kinematic equations can be explicitly inverted, so this problem represents a good test case for our method. We first considered this in [6]. The more challenging three-dimensional problem is considered later in Section 4.5.

Let the separation distance be δ and let this translation between coplanar bases be entirely along the x -axis; we will sort out-of-limit solutions in a post-processing procedure. We consider $0 < \delta < 4$, the largest value of δ corresponding to the robots barely reaching each other. The normalized distance to the grabbing point measured from the failed joint is denoted by a , with $0 < a < 1$. Also let the point at which a socket is attached to the left robot, hereafter robot 1, be point P , with the origin at the base of robot 1. Finally, let the fully functional machine be known as robot 2.

There is some probability that each joint could fail, so it is important to take into consideration the placement of a socket on each link. We use the Denavit-Hartenberg convention to describe the configurations of the two robots. Then we can describe the contact point P in terms of the DH parameters for robot 1, as P is some distance a from the origin of the previous frame. A spatial sample Q can then be said to be in the post-failure workspace of robot 1 for a parameter pair (δ, a) , if there exists a set of joint angles such that end effector of robot 1 is at Q , and the end effector of robot 2 is at P .

It seems that one obvious way to maximize $|W_f|$ is to set $\delta = 0$ so the two robots have exactly the same base point, and make the two robots identical. However, this is an impractical situation. First of all, there may be much greater benefit to having greater pre-failure workspaces by placing the robots further apart, and second, depending on the robot geometry, it could be awkward or impossible to place two

Table 4.1: Denavit-Hartenberg Parameters for two-link planar robot.

θ_i	α_i	a_i	d_i	$\theta_{i,min}$	$\theta_{i,max}$
θ_1	0	1	0	-120°	120°
θ_2	0	1	0	-45°	45°

identical robots operating from the same base point.

To demonstrate the method, we show results for a uniform random sampling of $n = 10^3$ points for contour plots, and $n = 10^5$ for W_f plots. The robots are identical, both having two joints, all normalized to length one. To start with, we uniformly sample a two dimensional rectangle which surely contains the workspace of robot two. A reasonable set of bounds are $\bar{p} = (x, y) \in [-2, 2] \times [-2, 2]$. To see the typical computation workload involved in the computation of W_i , we sampled initially $r_i = 10000$ times, and only $r_f = 7836$ of these points had at least one real solution, the estimated size of the workspace would be $|W_1| = 4^2 \times 7836 / 10000 \approx 12.5$. However, it is often more convenient to work with normalized measures; we will hereafter normalize with respect to the measure of the initial sample. Therefore the normalized measure of an initial workspace is simply r_i / r_f , and of this example is $7836 / 10000 \approx 0.783$.

Let $i \in \{1, 2, 3, 4\}$, so that we have a sine-cosine pair for each of the two joints in the two robots. We want to place the robots in such a way that robot 2 touches robot 1 at point P .

Here, we define c_i and s_i to be the cosine and sine, respectively, with $i = 1, 2$ corresponding to the first (failed) robot and $i = 3, 4$ corresponding to the second (assisting) robot. In this notation, our equations for this step are,

$$0 = \begin{cases} c_1 c_2 - s_1 s_2 + c_1 - x \\ s_1 c_2 + c_1 s_2 + s_1 - y \\ ac_1 c_2 - as_1 s_2 + c_1 - (c_3 c_4 - s_3 s_4 + c_3 + \delta) \\ as_1 c_2 + ac_1 s_2 + s_1 - (s_3 c_4 + c_3 s_4 + s_3) \\ s_i^2 + c_i^2 - 1 \end{cases} \quad (4.2)$$

We consider δ, a, x, y to be parameters for *Bertini*. We again solve these systems in two steps. Using the two-step procedure described above, the initial random solve involves 16 paths, and the subsequent parameter solves involve only 4 paths.

For each point in the sampling, we determine whether it lies in the desired space, by the above described method of homotopy continuation, using the inverse kinematic equations for the Denavit-Hartenberg parameters.

Because *Bertini* solves over the complex numbers for each variable, we find solutions for each point, regardless of whether it lies inside the workspace. However, the points that have solutions for which all variables are real-valued (numerically real-valued, calling zero any imaginary part less than 10^{-4}), are those lying within the workspace. The first step in the computation is to compute the workspace for both robots, as described in Sec. 4.3. Because the robots are identical in this example, there is only one workspace computation. The second step in computation is the

Table 4.2: Estimates on the normalized size of post-failure workspaces, with robot 2 grasping robot 1.

δ	a	$ W_{\cap} / W_1 $	$ W_f / W_1 $
0	$\forall a \in [0, 1]$	1	1
1	0.1	0.68	0.98
	0.5		0.86
	0.9		0.72
2	0.1	0.39	0.64
	0.5		0.54
	0.9		0.42
3	0.1	0.14	0.07
	0.5		0.14
	0.9		0.15
> 4	$\forall a \in [0, 1]$	0	0

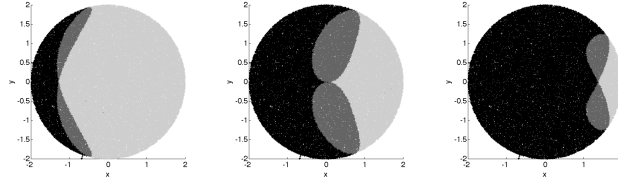


Figure 4.2: Example of W_f for $a = 0.5$ and $\delta = 1, 2, 3$. Black points are not reachable in the post-failure space. Dark gray areas are reachable in two configurations, as one for robot 1, and two for robot 2. Light gray areas are reachable in four configurations (two per robot).

intersection of the two workspaces, for a specified set of δ values. Finally, for each pair of values (δ, a) for which we estimate the post-failure grasping workspace, we solve a set of equations (4.2). For this example, we computed W_f for the set of δ and a , with $|W_f|$ normalized with respect to $|W_1|$.

Using a sampling of $n = 1000$ points, we computed both $|W_{\cap}|$ and $|W_f|$. The size of the intersection workspace is monotonically decreasing as we increase the separation distance. This makes geometric sense because W_{\cap} is merely the intersection of two circles. See Figure 4.3(a) and column three of Table 4.4.

The post-failure grasping workspace is more interesting, and depends both on δ and a . For close placements of robot bases, robot 2 can grasp robot 1 at almost any spot on the second link; W_f is almost identical to W_1 . As the robots move further apart, δ increases and the region robot 1 can reach while being grasped at P decreases. Not surprisingly, $|W_f|$ is greatest, for $\delta \in [0, 2]$ and $a \approx 0$, see Figure 4.3(b). Thus, we should place a ball joint near the base of joint 2 to ensure largest W_f for closer robot bases.

As the separation increases so that the workspaces barely overlap, robot 2 may

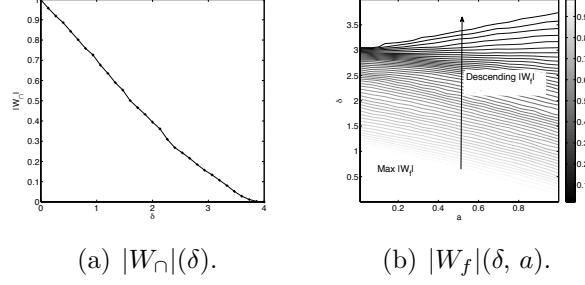


Figure 4.3: (a): The curve of $|W_{\cap}|$ in depending on δ is monotonically decreasing, with some wavering due to low number of points in the sample. (b): $|W_f|$ depending on (δ, a) . For small δ , robot 2 can grasp robot 1 at nearly any point P . As δ increases, a becomes more important. For δ near 4, the distance above which the robots cannot contact each other, there is a small range of a values that give nonempty W_f .

only grasp robot 1 near the end effector, and the resulting W_f is small. See Figure 4.2, and column four of Table 4.4. In these plots, the black area is inaccessible, light gray is fully accessible, and dark gray is reachable from a limited number of configurations. The inaccessible area clearly increases when δ is increasing. In general, larger δ lead to smaller post-failure workspaces for fixed a . The converse is not true: $|W_f|$ is not a monotonic function of a for a fixed δ . It is interesting to note that our methods computes explicitly the number of configurations reaching the desired point in a post-failure workspace, even in the case when the configurations belong to isolated domains in the workspace. Such problems are usually challenging for traditional methods of workspace computation.

Finally, in Figure 4.4 we show the contour levels of the objective function Ω from (4.1) to determine the optimal ball joint placement. For weighting factor $\lambda = 0$, Ω is simply the value of $|W_f|$, and the maximizer for any given δ is $a = 0$. With $\lambda = 1/3$, the optimal joint location is still at the base of the second link, but occurs with separation $\delta = 1$. Increasing λ further makes the size of the intersection workspace overpower the post-failure workspace, and by $\lambda = 2/3$ the maximum of Ω is invariant with respect to a . Therefore, the weighting factor λ plays a critical role in the determination of the optimal grabbing point.

4.5 3D Case: Three Joint Manipulators

Equations determining the kinematics of three dimensional robots are more complicated than those of planar manipulators. This concerns both the higher number of relevant equations, due to the higher number of links in a typical 3D robots, and the structure of equations describing the motions. Yet, the method for optimizing cooperative workspaces remains fundamentally the same, as those equations can be brought to an algebraic form and then solved using the methods outlined above in Section 4.3. Thus for the sake of brevity, in this section we only write a sketch of the

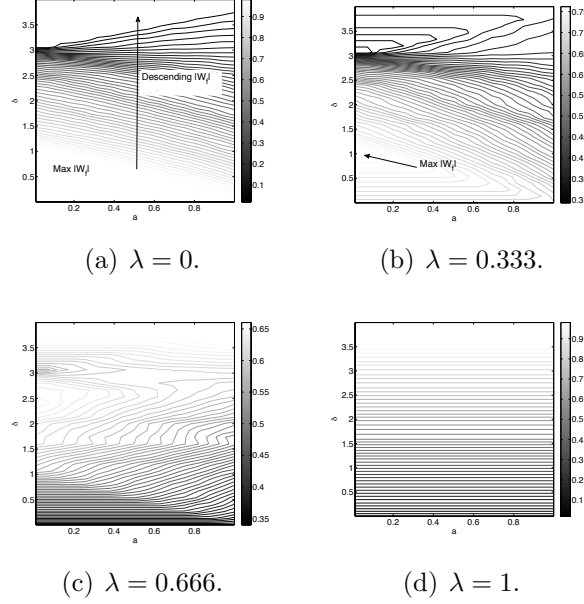


Figure 4.4: Objective function contours for a grid of (δ, a) pairs for 2D cooperating robots.

Table 4.3: Denavit-Hartenberg Parameters for PUMA robot.

θ_i	α_i	a_i (m)	d_i (m)	$\theta_{i,min}$	$\theta_{i,max}$
θ_1	0	0	0	-160°	160°
θ_2	$-\pi/2$	0.4318	0.2435	-225°	45°
θ_3	0	-0.0203	-0.0934	-45°	225°
θ_4	$\pi/2$	0	0.4331	-110°	170°
θ_5	$-\pi/2$	0	0	-100°	100°
θ_6	$\pi/2$	0	0.5625	-266°	266°

method in 3D.

In the example for this section, we consider two cooperating PUMA robots, ignoring orientation of the end effector. There is some disagreement in the relevant literature regarding which parameters should be used in describing such a robot [10]. In this paper, we use DH parameters defined as in Table 4.3. Treating the first three links of the PUMA as our robot, we ignore the wrist; we do this to limit the curse of dimensionality and reduce the number of points to analyze. Instead, we use a tool frame to translate from the arm to the end effector. The frame we used retains the orientation of the arm, and translates along the final z -axis by 0.4331 meters.

In order to save space and simplify the computations, we shall only consider the robots cooperating through the first three links. Also, we assume the bases to be coplanar, and the robots to be oriented with respect to this plane in the same direction; this allows us to parametrize the separation of the PUMAs with only one pa-

parameter. Each of the workspaces W_1 , W_2 , W_\cap will have six equations in six variables, which are solved via *Bertini* after bringing the trigonometric part of these equations into algebraic form. Correspondingly, there will be 12 equations defining the post-failure workspaces with 12 variables; these equations appear in Appendix 4.7. The space of possible positions for each manipulator is now a subset of \mathbb{R}^3 . In order to find $W_{1,2}$ we sample randomly an oversized rectangular box surrounding the robot, which will include the workspace. In order to determine good bounds for the workspaces, we first do forward kinematics by randomly sampling the three joint variables $\theta_i \in [0, 2\pi]$. The result of this estimate is that the cubic box $[-0.9, 0.9] \times [-0.9, 0.9] \times [-0.9, 0.9]$ contains W_i , and is thus a good starting point for finding the cooperative workspaces in which we are interested.

The PUMA has joint limits that reduce its workspaces significantly. The limits we use for this example appear in Table 4.3. because joint limits are written as inequalities $\theta_{i,min} \leq \theta_i \leq \theta_{i,max}$, they are not algebraic equations. There are two possibilities for dealing with this problem. One could compute the workspace boundaries at the joint limits $\theta_i = \theta_{i,max}$ and $\theta_i = \theta_{i,min}$. However, these constraints do not always lead to meaningful boundaries in cooperating workspaces, and even when they do, it is often hard to define the volumes they bound. Instead, we simply use these joint limits in post-processing, only selecting the suitable joint angles among all real solutions.

Our results for various workspaces are shown in Figure (4.5). These data are based on 10,000 points in the (x, y, z) space. Of particular interest here is the presence of voids inside of all workspaces. Because the data plotted is generated from a random sampling, to plot nicely we called the Matlab method `griddata`, and the result is a blocky pseudo-color plot. Forbidden area is represented by the dark blue color. The red and yellow colors show the accessible work space, with the yellow areas being accessible only if the joint limits are satisfied. The resulting workspace is essentially a torus, although the particular realization we have chosen makes it hard to visualize this as a volume in the 3D space, because of the relative narrowness of the “hole”. Instead, we have chosen to represent the workspace through the slices. Two slices by the planes $x = 0$ and $z = 0$ are shown in the top of the Figure, and the combined figure presenting the slices is shown in the bottom. The voids in W_i come from the offsets of the arm, and they expand for cooperating robots. Our results show that great care needs to be taken in designing and arranging robotic arms for cooperation, as it may lead to large inaccessible regions of workspace.

We also present the plot of objective function versus parameters δ and a in Figure 4.5, chosen to extend the results presented above for 2D robots. To calculate Ω , we compute each of the workspaces W_1 , W_\cap and W_f . Initially, W_1 is determined using the results shown in Figure 4.5, computed with 10^4 points. Then W_\cap is computed over a discretization of $0 \leq \delta \leq 1.8$ into 16 values. Finally, we compute W_f , for a particular value of δ and a . The results are repeated over a 16×16 regular grid of $0 \leq \delta \leq 1.8$, and $0 \leq a \leq 1$, and considering grasping the failed robot on each of the three possible links. Thus, the total number of parameter combinations we considered for W_f is $7.68 \cdot 10^6$. However, we do not need to treat that many paths by *Bertini*, as we throw out the space samples that are not in the initial workspace for subsequent computations.

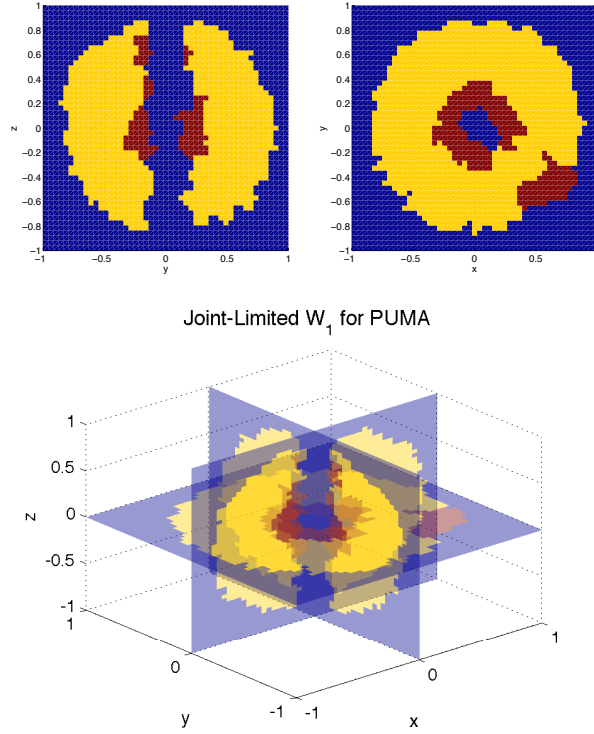


Figure 4.5: Top: Slices of the pre-failure PUMA workspace W_1 by the planes $x = 0$ (left) and $z = 0$ (right). Bottom: Combined picture of workspace with three slices $x = 0$, $y = 0$, $z = 0$. Yellow color: areas accessible the angles satisfying joint limits given in Table 4.3. Red color: real solutions violating joint limits. Dark blue region: inaccessible.

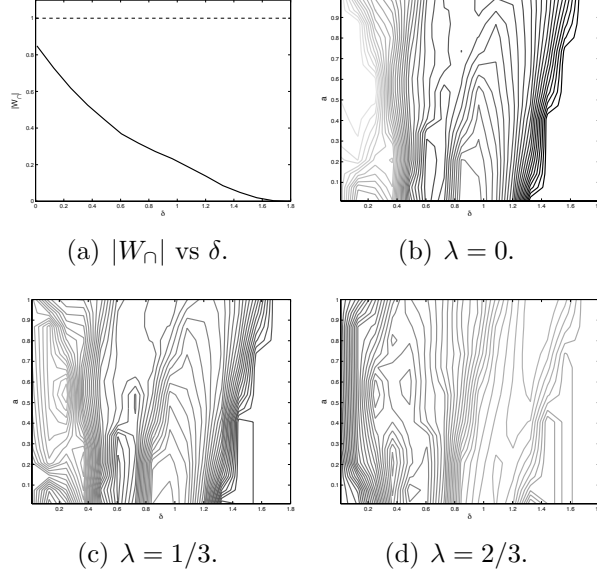


Figure 4.6: Considering grasping on the second link to restore workspace after failure of joint 2 of a PUMA robot. a) Measurement of the intersection workspace $|W_{\cap}|$, as a function of δ . b)-d) Contours of the objective function Ω versus (δ, a) for 3D cooperating robots. Note that for the case b), $\lambda = 0$ and the objective function Ω is equal to the measure of post-failure workspace $|W_f|$.

The objective function landscapes presented in Figure 4.5 for the PUMA robot are very similar to those for the 2D planar robot above in Figure 4.4. Because each joint could fail independently of the others, we need to consider a grasping location for each link of the robot; hence, we have an objective function landscape for each of the three links of the PUMA. However, the landscapes are similar, so only those for the second link are presented here. The values of $|W_{\cap}|$ monotonically decrease with increase of δ , with $W_1 = W_{\cap}$ for $\delta = 0$. For results on W_f , we average over the sizes of W_f for grasping on each link, at normalized distance a from the base. This would be user-defined in practice; for example, if we knew the third joint was most likely to fail, we might compute Ω using only W_f for this joint. This extra information will also give substantial computational savings.

As with the 2D example above, the weighting factor λ plays a crucial role in optimizing. The limiting cases of $\lambda = 1$ and $\lambda = 0$ return simply $(1 - |W_{\cap}|)$ or $|W_f|$ respectively. Around $\lambda = 2/3$, we are more concerned with maximizing pre-failure workspace. The maximum values of Ω appear at the right of the figure, with large separation between the robots. Around $\lambda = 1/3$, we put more emphasis on increasing the remainder of workspace accessible in grasping configuration;. For that value of λ , objective function Ω is mostly flat, with a rise near $\delta = 0.2$, and $a = 0.5$. The optimal distance and grasping location depends on the link, and on the weighting factor, making user preference crucial in the optimization procedure.

4.6 Conclusion

We used homotopy continuation, as implemented in *Bertini*, to estimate the size of workspaces, the intersection of workspaces, and post-failure grasping workspaces in the case of having two serial robots placed near one another, in two and three dimensions. We also solve the problem of the optimal configuration for these robots for one example of a user-defined objective function. A general algorithm for solving the problem of finding optimal placement and configuration of two such robots was also presented.

Yet again, one could get some results for this optimization problem using standard methods, such as Jacobian control. However, by using algebraic geometric methods, we avoid issues such as isolated domains and multiple solutions to inverse kinematics equations. For example, multiple isolated solutions can appear even in the pre-failure workspace in the case of a robot having to work around an obstacle, with the target reachable in multiple ways. The homotopy continuation algorithms will not encounter any difficulty in that case, whereas the Jacobian control method will need to be augmented with the specific knowledge from the problem and yet fail to find all isolated solutions. Our algorithm can be extended to more general cases. For example, it will be relatively straightforward to account for different designs for the two robots (including unequal link lengths) or introduce the dynamical load variables in the objective function. Also, methodological choices in the algorithm, such as the use of homotopy continuation, have been made to make the generalization to higher-dimensional workspaces possible.

It should be noted that the methods of numerical algebraic geometry may be used to compute *complex* positive-dimensional components of the solutions sets of polynomial systems. However, it is nearly impossible to detect positive-dimensional *real* solutions, particularly above dimension one. Therefore, kinematically redundant robots present difficulty for our method. Perhaps we could systematically reduce the underconstrained inverse kinematic system for such a robot to a fully constrained system; however, the curse of dimensionality will likely be problematic. This is a direction of ongoing research.

One can say even though the method is highly comprehensive, the disadvantage lies in the relative computation intensity and large number of data generated by the method. In particular, all the results in this paper have been obtained on a 72-core Dell cluster; the PUMA data took about 24 hours to generate. We shall note we tried to keep the algorithm as general as possible, without referring to the knowledge of particular workspace or robotic geometries, that may speed up the computation considerably. Also, because the homotopy continuation algorithm is pleasantly parallel, for practical applications it can be extended to run with high efficiency on modern GPU workstations. Thus, even though the computations for this paper took considerable time, it is not inconceivable that it can be made to run for real-time robot guidance in the near future.

4.7 eqns

The inverse kinematic equations for the first three joints of the PUMA robot are, as used in this paper are $f_i = 0$ for $i = 1, \dots, 12$ with f_i defined as:

$$\begin{aligned}
f_1 &= 0.4318c_1c_2 - 0.2435s_1 + 0.0934as_1 + 0.4331c_1c_2s_3 + \\
&\quad 0.4331c_1c_3s_2 + 0.0203ac_1s_2s_3 - 0.0203ac_1c_2c_3 - \\
&\quad (0.4318c_4c_5 - 0.1501s_4 - 0.0203c_4c_5c_6 + \\
&\quad 0.4331c_4c_5s_6 + 0.4331c_4c_6s_5 + 0.0203c_4s_5s_6 + \delta) \\
f_2 &= 0.2435c_1 - 0.0934ac_1 + 0.4318c_2s_1 + 0.4331c_2s_1s_3 + \\
&\quad 0.4331c_3s_1s_2 + 0.0203as_1s_2s_3 - 0.0203ac_2c_3s_1 - \\
&\quad (0.1501c_4 + 0.4318c_5s_4 - 0.0203c_5c_6s_4 + \\
&\quad 0.4331c_5s_4s_6 + 0.4331c_6s_4s_5 + 0.0203s_4s_5s_6) \\
f_3 &= 0.4331c_2c_3 - 0.4318s_2 - 0.4331s_2s_3 + 0.0203ac_2s_3 + \\
&\quad 0.0203ac_3s_2 - (0.4331c_5c_6 - 0.4318s_5 + 0.0203c_5s_6 + \\
&\quad 0.0203c_6s_5 - 0.4331s_5s_6) \\
f_4 &= 0.4318c_1c_2 - 0.1501s_1 - 0.0203c_1c_2c_3 + \\
&\quad 0.4331c_1c_2s_3 + 0.4331c_1c_3s_2 + 0.0203c_1s_2s_3 - x \\
f_5 &= 0.1501c_1 + 0.4318c_2s_1 - 0.0203c_2c_3s_1 + \\
&\quad 0.4331c_2s_1s_3 + 0.4331c_3s_1s_2 + 0.0203s_1s_2s_3 - y \\
f_6 &= 0.4331c_2c_3 - 0.4318s_2 + 0.0203c_2s_3 + \\
&\quad 0.0203c_3s_2 - 0.4331s_2s_3 - z \\
f_7 &= s_1^2 + c_1^2 - 1 \\
f_8 &= s_2^2 + c_2^2 - 1 \\
f_9 &= s_3^2 + c_3^2 - 1 \\
f_{10} &= s_4^2 + c_4^2 - 1 \\
f_{11} &= s_5^2 + c_5^2 - 1 \\
f_{12} &= s_6^2 + c_6^2 - 1
\end{aligned}$$

The coefficients appearing in these equations are 4 digit approximations of exact numbers coming from the DH parameters for the PUMA. The terms x, y, z, δ, a appear in the equations as parameters for *Bertini*. Functions $f_1 - f_6$ couple the positions of the two cooperating robots, and functions $f_7 - f_{12}$ are Pythagorean identities tying cosines and sines together.

Chapter 5

Conclusion

Bibliography

- [1] Farhad Aghili and Kouros Parsa. A reconfigurable robot with lockable cylindrical joints. *Trans. Rob.*, 25(4):785–797, 2009.
- [2] Eugene L. Allgower and Kurt Georg. *Numerical continuation methods, an introduction*. Springer-Verlag, Berlin, 1990.
- [3] M. Anand, T. Selvaraj, S. Kumanan, and J. Janarthanan. A hybrid fuzzy logic artificial neural network algorithm-based fault detection and isolation for industrial robot manipulators. *International Journal of Manufacturing Research*, 2(3):279–302, 2007.
- [4] Dan J. Bates, Daniel A. Brake, and Matthew Niemerg. Paramatopy: parallel parameter homotopy via Bertini. *in preparation*, 2011.
- [5] Daniel J Bates, John D Hauenstein, Andrew J Sommese, and Charles W Wampler. Bertini, 2010.
- [6] Daniel A. Brake, Dan J. Bates, Vakhtang Putkaradze, and Anthony A. Maciejewski. Illustration of numerical algebraic methods for workspace estimation of cooperating robots after joint failure. In *IASTED Technology Conferences*, Pittsburg, PN USA, November 2010.
- [7] D. Brambilla, L. Capisani, A. Ferrara, and P. Pisu. Fault detection for robot manipulators via second-order sliding modes. *IEEE Trans. Industrial Electronics*, 55(11):3954–3963, 2008.
- [8] C. Carreras and I. D. Walker. Interval methods for fault-tree analysis in robotics. *IEEE Trans. Robot. Automat.*, 50(1):3–11, 2001.
- [9] Y. Chen, J. E. McInroy, and Y. Yi. Optimal, fault-tolerant mappings to achieve secondary goals without compromising primary performance. *IEEE Trans. Robotics*, 19(4):680–691, 2003.
- [10] P.I. Corke and B. Armstrong-Helouvry. A search for consensus among model parameters reported for the PUMA 560 robot. In *Robotics and Automation, 1994. Proceedings., 1994 IEEE International Conference on*, pages 1608 –1613 vol.2, may 1994.

- [11] J. English and A. Maciejewski. Measuring and reducing the Euclidean-space measures of robotic joint failures. *IEEE Transactions on Robotics and Automation*, 16(1):20–28, 2000.
- [12] J. English and A. Maciejewski. Failure tolerance through active braking: A kinematic approach. *The International Journal of Robotics Research*, 20(4):287–299, 2001.
- [13] J. D. English and A. A. Maciejewski. Fault tolerance for kinematically redundant manipulators: Anticipating free-swinging joint failures. *IEEE Trans. Robot. Automat.*, 14(4):566–575, 1998.
- [14] L. Q. English, F. Palmero, A. J. Sievers, P. G. Kevrekidis, and D. H. Barnak. Traveling and stationary intrinsic localized modes and their spatial control in electrical lattices. *Phys. Rev. E*, 81(4):046605, Apr 2010.
- [15] B. S. Dhillon A. R. M. Fashandi and K. L. Liu. Robot systems reliability and safety: A review. *Journal of Quality in Maintenance Engineering*, 8(3):170–212, 2002.
- [16] K. N. Groom, A. A. Maciejewski, and V. Balakrishnan. Real-time failure-tolerant control of kinematically redundant manipulators. *IEEE Trans. Robot. Automat.*, 15(6):1109–1116, 1999.
- [17] M. Hassan and L. Notash. Optimizing fault tolerance to joint jam in the design of parallel robot manipulators. *Mech. Mach. Theory*, 42:1401–1407, 2007.
- [18] S. D. Hersee, X. Y. Sun, X. Wang, M. N. Fairchild, J. Liang, and J. Xu. Visualizing intrinsic localized modes with a nonlinear micromechanical array. *J. Appl. Phys.*, 97(12):124308, 2005.
- [19] V Hizhnyakov, A Shelkan, M Klopov, A J Sievers, and M Haas. Intrinsic localized modes and trapped phonons in crystal lattices. *Journal of Physics: Conference Series*, 92(1), 2007.
- [20] M. Ji and N. Sarkar. Supervisory fault adaptive control of a mobile robot and its application in sensor-fault accommodation. *IEEE Trans. Robotics*, 23(1):174–178, 2007.
- [21] P. G. Kevrekidis, A. R. Bishop, and K. Ø. Rasmussen. Twisted localized modes. *Phys. Rev. E*, 63(3):036603, Feb 2001.
- [22] P. G. Kevrekidis, S. V. Dmitriev, S. Takeno, A. R. Bishop, and E. C. Aifantis. Rich example of geometrically induced nonlinearity: From rotobreathers and kinks to moving localized modes and resonant energy transfer. *Phys. Rev. E*, 70(6):066627, Dec 2004.
- [23] P. G. Kevrekidis and V. V. Konotop. Bright compact breathers. *Phys. Rev. E*, 65(6):066614, Jun 2002.

- [24] M. Kimura and T. Hikiyara. Capture and release of traveling intrinsic localized mode in coupled cantilever array. *Chaos*, 19(1):13138, 2009.
- [25] Masayuki Kimura and Takashi Hikiyara. Coupled cantilever array with tunable on-site nonlinearity and observation of localized oscillations. *Physics Letters A*, 373(14):1257 – 1260, 2009.
- [26] C. L. Lewis and A. A. Maciejewski. Fault tolerant operation of kinematically redundant manipulators for locked joint failures. *IEEE Trans. Robot. Automat.*, 13(4):622–629, August 1997.
- [27] A. De Luca and L. Ferrajoli. A modified Newton-Euler method for dynamic computations in robot fault detection and control. In *IEEE Int. Conf. Robot. Automat.*, pages 3359–3364, May 2009.
- [28] A. A. Maciejewski. Fault tolerant properties of kinematically redundant manipulators. In *Proc. IEEE Int. Conf. Robot. Automat.*, pages 638–642, Cincinnati, OH, USA, 1990.
- [29] Boris A. Malomed, P. G. Kevrekidis, D. J. Frantzeskakis, H. E. Nistazakis, and A. N. Yannacopoulos. One- and two-dimensional solitons in second-harmonic-generating lattices. *Phys. Rev. E*, 65(5):056606, Apr 2002.
- [30] J. E. McInroy, J. F. O’Brien, and G. W. Neat. Precise, fault-tolerant pointing using a stewart platform. *IEEE/ASME Trans. Mechatronics*, 4(1):91–95, 1999.
- [31] J. B. Page. Asymptotic solutions for localized vibrational modes in strongly anharmonic periodic systems. *Phys. Rev. B*, 41(11):7835–7838, Apr 1990.
- [32] C. J. J. Paredis and P. K. Khosla. Designing fault-tolerant manipulators: How many degrees of freedom? *Int. J. Robot. Res.*, 15(6):611–628, December 1996.
- [33] R. G. Roberts and A. A. Maciejewski. A local measure of fault tolerance for kinematically redundant manipulators. *IEEE Trans. Robotics Automat.*, 12(4):543–552, 1996.
- [34] M. Sato, B. E. Hubbard, L. Q. English, A. J. Sievers, B. Ilic, D. A. Czaplewski, and H. G. Craighead. Study of intrinsic localized vibrational modes in micromechanical oscillator arrays. *Chaos*, 13(2):702, May 2003.
- [35] M. Sato, B. E. Hubbard, and A. J. Sievers. Colloquium: Nonlinear energy localization and its manipulation in micromechanical oscillator arrays. *Rev. Mod. Phys.*, 78(1):137–157, Jan 2006.
- [36] M. Sato, B. E. Hubbard, A. J. Sievers, B. Ilic, D. A. Czaplewski, and H. G. Craighead. Observation of locked intrinsic localized vibrational modes in a micromechanical oscillator array. *Phys. Rev. Lett.*, 90(4):044102, Jan 2003.

- [37] M. Sato and A. Sievers. Experimental and numerical exploration of intrinsic localized modes in an atomic lattice. *Journal of Biological Physics*, 35:57–72, 2009.
- [38] M. Sato and A. J. Sievers. Direct observation of the discrete character of intrinsic localized modes in an antiferromagnet. *Nature*, 432(7016):486–488, 11 2004.
- [39] M. Sato and A. J. Sievers. Visualizing intrinsic localized modes with a nonlinear micromechanical array. *Low Temperature Physics*, 34(7):543–548, 2008.
- [40] M. Sato and A. J. Sievers. Visualizing intrinsic localized modes with a nonlinear micromechanical array. *Fizika Nizkikh Temperatur*, 34(7):687–694, June 2008.
- [41] A. J. Sievers and S. Takeno. Intrinsic localized modes in anharmonic crystals. *Phys. Rev. Lett.*, 61(8):970–973, Aug 1988.
- [42] Andrew Sommese, Jan Verschelde, and Charles W Wampler. Advances in polynomial continuation for solving problems in kinematics. *ASME J. Mechanical Design*, 126(2):262–268, 2004.
- [43] Andrew J Sommese and Charles W. Wampler. *The numerical solution to systems of polynomials arising in engineering and science*. World Scientific, Singapore, 2005.
- [44] R Basu Thakur, L Q English, and A J Sievers. Driven intrinsic localized modes in a coupled pendulum array. *Journal of Physics D: Applied Physics*, 41(1):015503, 2008.
- [45] Charles W. Wampler, Jonathan D. Hauenstein, and Andrew J. Sommese. Mechanism mobility and a local dimension test. *Mechanism and Machine Theory*, 46(9):1193–1206, 2011.
- [46] Charles W Wampler and A. P. Morgan. Solving the kinematics of general 6R manipulators using polynomial continuation. In K Warwick, editor, *Robotics: Applied Mathematics and Computational Aspects*, pages 57–69. Clarendon Press, Oxford, 1993.
- [47] Charles W Wampler, A. P. Morgan, and Andrew J Sommese. Complete solution of the nine-point path synthesis problem for four-bar linkages. *Journal of Mechanical Design*, 114:153–159, March 1992.
- [48] E. C. Wu, J. C. Hwang, and J. T. Chladek. Fault-tolerant joint development for the space shuttle remote manipulator system: Analysis and experiment. *IEEE Trans. Robot. Automat.*, 9(5):675–684, 1993.
- [49] Y. Yi, J. E. McInroy, and Y. Chen. Fault tolerance of parallel manipulators using task space and kinematic redundancy. *IEEE Trans. Robotics*, 22(5):1017–1021, 2006.