Workspace Multiplicity and Fault Tolerance of Cooperating Robots

Daniel A. Brake[1⋆], Daniel J. Bates[2⋆⋆], Vakhtang Putkaradze[3⋆⋆⋆], and Anthony A. Maciejewski[4†]

[1] Department of Applied and Computational Mathematics and Statistics, University of Notre Dame
[2] Department of Mathematics, Colorado State University
[3] Department of Mathematical and Statistical Sciences, University of Alberta
[4] Department of Electrical and Computer Engineering, Colorado State University

Abstract. Cooperating robotic systems, especially in the context of fault-tolerance of complex robotic mechanisms, is an important question for theoretical and applied studies. In this paper, we focus on one measure of fault tolerance in robots, namely, the *multiplicity* of the configurations for reaching a particular point in the workspace, which is difficult to measure using traditional methods. As a particular example, we consider the case of a free-swinging failure of a robotic arm that is handled by having a cooperating functional robot grasp the link adjacent to the failed joint. We present an efficient method to compute the multiplicity measure of the workspace, based on the tools from numerical algebraic geometry, applied to the inverse kinematics problem re-cast in the form of a polynomial system. To emphasize the difference between our methods and more traditional approaches, we compute the measure of workspace based on the multiplicity of configurations, and optimize placement of synergistic robot arms and the optimal grasp point on each link of the broken robot based on this measure.

keywords: Workspace mapping, joint failure, homotopy continuation, Monte Carlo methods

1 Introduction

Fault-tolerance and robustness play important roles in the design of autonomous systems, including robotic arms. Often times, this has been achieved with either redundancy in drive systems, or cooperating robotic arms. Consideration of fault-tolerance is essential in dangerous locations, and continues to be studied extensively. Determining the reliability of a robot via fault-tree analysis appeared

⋆ Partially supported by grants NSF-DMS-09087551 and NSF-IIS-0812437.
⋆⋆ Partially supported by grants NSF DMS–0914674 and NSF DMS–1115668.
⋆⋆⋆ Partially supported by grant NSF-DMS-09087551.
† Partially supported by grant NSF-IIS-0812437.

in [1], which permits quantification of weaknesses in design. 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 [2–5]. Failure-tolerant regions of a workspace were defined in [6], which also gave a method for computing a fail-tolerance measure which allows a robot to operate in real-time within this fail-tolerant region. Other measures of fail-tolerance include analysis of the singular values of the Jacobian matrix for a robot [7] and a 'manipulability index' as defined in [8].

Anticipation of failures has also been explored considerably; decomposing a task into primary and secondary goals is one method of dealing with these problems. Robots can be designed and operated with fail-tolerance in mind, such as dual actuators at joints [9], kinematic redundancy [9–13], and reconfigurability [14]. Prioritization of tasks subject to other constraints (*e.g.*, environmental) was treated in [15], restriction of operation to a fault-tolerant region in [16], and anticipation of free-swinging joint failures in [17]. The overarching theme in this field is graceful degradation of performance.

In this paper, we consider autonomously operated robotic systems that are deployed into a hazardous or remote location, such that the repair of 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. Certainly, there are many ways for a robotic system to fail. Sensors give false or no readings; electronic components break; controllers fault; actuators seize or give way. We aim to supplement present methods regarding graceful joint failure, by informing designers and operators of possibilities for assistance to a broken robot by a functional one, should one be available. In the event of a free-swinging failure, our method provides information about optimal placement of a predetermined socket or other suitable apparatus on an articulated arm to allow assistance from a functional robot. Even with an actuator with non-free-swinging failure, such as unexpected resistance or friction, our method could contribute to mission success. The failure-proof of systems has been particularly important for applications such as space-faring vehicles [18].

To make our consideration more realistic, suppose a system operating in a remote environment possesses several robotic arms for various tasks. 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 can be restored. In this work, we shall assume that the relative position of the bases for robotic arms cannot change, and only one joint fails at a time.

As far as fault-tolerance consideration of a *single* robot goes, the multiplicity of configurations may not be a relevant quantity. For the purpose of restoring workspace using *cooperating* robotic arms the multiplicity of configurations is of crucial importance. As we shall see below, the workspaces of cooperating robots tend to consist of several isolated sets.

The main goal of this paper is to outline a method that is alternative to the traditional ways of workspace computation. The main difference is that we

use sampling *directly in the workspace*, and therefore are not subject to failures of a Jacobian method due to possible singularities of the forward kinematics mapping. Our method, which is based on the applications of ideas in algebraic geometry (homotopy continuation) to the solution of polynomial problems, can provide the exact number of solutions for each particular point of the workspace. 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. Methods of algebraic geometry have been applied in the kinematic description of robots, see for example [19–24]. However, analysis of the cooperation of multiple arms via the tools of numerical algebraic geometry has never been explored.

We shall outline a particular application of the method: computation of workspace and optimization of the grasping point for two cooperating robots, in case of joint failure of the first robot. This example was chosen as a realistic demonstration, leading to interesting shapes of workspaces with several sets of multiple solutions. For the optimization, we ask two questions: *a)* What is the best placement of the two robots in relation to each other? *b)* Is there a best place for the second arm to grasp the first? Clearly, the answer to both questions depends on the choice of measure for the post-failure workspace, which is problem dependent. In this paper, we introduce a multiplicity-weighted workspace measure, depending on both the number of robot configurations and the geometric size of the workspace. Combining this on the pre- and post-failure workspaces gives a single measure, parameterized by a user-determined weighting factor, and the distance between robot bases and grasp point.

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 joint values in isolated regions of joint-space.

In Section 2, we provide a formal statement of the general problem we are considering. Details about our method are described in Section 3, and background regarding the numerical solution of polynomial systems may be found in Section 4. Sections 5.1 and 5.2 respectively contain two and three dimensional examples.

2 Formal Problem Statement

Given two articulated arms, we optimize the placement of grasping sockets to maximize post-failure workspace (we assume a socket attachment mechanism of the two arms as considered in [14]). That is, if one robot has a free swinging joint failure as in [25, 26], we would like to ensure that when the functional robot joins at the socket to assist in completion of tasks, the resulting cooperative workspace is as large as possible. A simplified example of this problem is in Fig. 1.

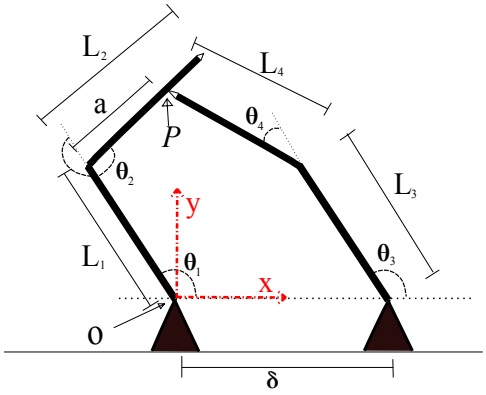


Fig. 1. Two robots in grasping configuration, with functional robot contacting the disabled unit at point P. This is a simplified 2D model of the general 6D position and orientation scenario. Parameter δ is the distance between the robot bases.

Each robot has its own pre-failure workspace, $W_1, W_2 \subset \mathbb{R}^n \times SO(N)$, where n and N depend on the particular mechanism type. When the two robots are placed near enough, 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 locations Robot 1 can reach, if Robot 2 attaches to a socket location on Robot 1. The measures $|W_\cap|$ and $|W_f|$ developed below in Equation (2) depend on the separation between the robots, as well as their orientation. 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, and to the measures of the various workspaces.

The parameters describing an optimal configuration of two cooperating robots include: separation between cooperating arms, relative orientation of the bases, and location of grasping point. In this paper, we do not optimize with respect to base position of each robot, as we consider that the position of the bases must be given from design limitations, *e.g.* the placement of power cords and motors on the apparatus. Nevertheless, our method is capable of optimizing socket placement for different base separations, and this is demonstrated in the examples.

In order to quantify post-failure workspace size and find an optimal configuration, we introduce a maximizing objective function Ω. There are many possible objective functions one can imagine, and the right choice depends on the application. In this paper, we define a mutiplicity-weighted measure of workspace W

and objective function Ω_λ through the multiplicity of configurations (i.e., the number of inverse kinematic solutions placing the robot in the specified configuration) at a given point x, denoted $m(x)$:

$$|W| = \int_W m(x)\mathrm{d}x, \tag{1}$$

$$\Omega_\lambda = (1-\lambda)|W_f| + \lambda(|W_1| - |W_\cap|), \tag{2}$$

where λ is a weighting factor to be chosen by the user. We have chosen Equation (2) 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 $\lambda(|W_1| - |W_\cap|)$, but W_f would be an empty set. Contrarily, we might find a configuration which results in full restoration of 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 prefer to balance between pre- and post-failure benefits of having two robots, and Equation (2) is one way of doing this.

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

In fact, the number of configurations could be infinite at some points in the workspace. However, this is an algebraic condition so this set of all points in the workspace for which this is true has measure zero. Such points are kinematic singularities and should be excluded in Equation (2) above. More practically, the Monte Carlo methods used later in the paper miss such points with probability one.

3 Workspace Computation

In this section, we describe our solution to the problem outlined in Section 2. We start with notation. Let the separation between bases be δ and without loss of generality let this translation between coplanar bases be entirely along the x-axis; we sort out-of-limit solutions in a post-processing procedure. We consider $0 < \delta < \delta_{\max}$, the largest value of δ corresponding to the the sum of the lengths of each robot fully extended. The normalized distance to the grabbing point or socket, measured from the failed joint, is denoted by a, with $0 < a < 1$; the point at which a socket is attached to the left robot, hereafter Robot 1, is P; a test point in space is Q; and the origin at the base of Robot 1. Finally, let the fully functional machine be known as Robot 2. A simplified version of this notation is found in Fig. 1.

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 (of non-zero length). We

use the Denavit-Hartenberg (DH) convention to describe the configurations of the two robots, and 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 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.

The equations for the inverse kinematics problem for each step of the method are first solved via a standard homotopy run at $\bar{p}^*$, a random point in complex parameter space. Then all subsequent runs at points in the workspace are treated as parameter homotopies, beginning at $\bar{p}^*$. For W_i, the parameters are the x, y, z coordinates of Q; for $W_\cap$, we add parameter δ; and for W_f, we add a. The software package *Bertini* [29] can be used to compute numerical approximations of the solutions of polynomial systems. *Bertini* works over $\mathbb{C}$, so 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 numerically real-valued are those lying within the workspace, while those with nonzero imaginary component lie outside.

We note that the inverse kinematics equations for a robot with rotary joints are trigonometric in nature. In order to use polynomial homotopy continuation, we write the equations in polynomial form by treating each sine-cosine pair as a separate variable, mapping $\cos(\theta_j) = c_j$ and $\sin(\theta_j) = s_j$ and coupling with the algebraic condition $c_j^2 + s_j^2 - 1 = 0$.

We compute the initial pre-failure workspaces for each robot via a random sampling method on an ambient set S of the workspace guaranteed to contain the workspace, and estimate the measure of the workspaces as in Equation (2) using the number of samples in the workspace pointwise multiplied by multiplicity factor for each point,

$$|W| = \lim_{r\to\infty} |S| \frac{1}{r} \sum_{j=1}^{r} m(x_j), \tag{3}$$

where $|S|$ is the typical Euclidean measure of the sampling space.

The method for computing optimal $\delta_{\max}$ and $a_{\max}$ is described in Algorithm 1. Starting from S, we compute each of the necessary workspaces. After computing W_1 we no longer need S, because $W_\cap, W_f \subset W_1$. Instead, all further computations are over W_1. Once we have the data for the multiplicities of the workspaces, we simply compute Ω_λ, and return its maximizing parameter values.

For the case of $N \geq 3$ joints working in 3 dimensions without orientation, there are 3 algebraic inverse kinematic equations in $2N$ variables, coupled with N Pythagorean identities, 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, the joint space consists of a set of higher $N-3$ dimensional manifolds, which could be described by defining a mesh of the same dimensionality as coordinates on the joints. Our method readily applies to this higher dimensional problem. 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

computing the solutions 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 corresponds to a two dimensional manifold in joint space. To cut down the manifold to be 0-dimensional for solving via *Bertini*, we could discretely sample each pair of joints possible, and solve for the remaining three. However, combinatorial growth issues arise, *e.g.* if we wanted to solve each spatial sample for each possible combination of joints.

4 Homotopy Continuation

In general, given an arbitrary (not necessarily robotic) set of polynomials $\bar{f} = \{f_1, \ldots, f_N\}$ in N variables for which we seek the solutions, homotopy methods begin by choosing and solving some other related polynomial system $\bar{g} = \{g_1, \ldots, g_N\}$ for which the solutions are easily found. By varying the coefficients in $\bar{g}$ to those of $\bar{f}$, statements in algebraic geometry guarantee that, with probability one, the solutions will vary continuously, thus forming *solution 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 Runge-Kutta and Newton's method. Further details may be found in [27, 20, 28].

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 aim to solve a large number of polynomial systems that differ only in coefficients (i.e., we aim 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*. In this setting, we first solve $\bar{f}(\bar{p})$ at a single instance of $\bar{p}^*$ (typically chosen as random complex numbers, for theoretical reasons, as described in §7.1 of [20]). This stage may require the tracking of a number of superfluous, divergent paths. However, all other instances of $\bar{f}(\bar{p})$ may then be solved by simply following the handful of finite solutions at $\bar{p}^*$ to any other choice of $\bar{p}$. For example, the system used to solve W_f, grasping on the third link, for an initial random complex parameter choice, requires following 20,736 paths to find the *sixteen* solutions of interest. Then, for all other points in the parameter space, it suffices to follow just sixteen paths.

Input : DH parameters for each robot, λ, samplings of δ, a, S
Output: Optimal $\delta_{\max}, a_{\max}$, $\Omega_\lambda(\delta_{\max}, a_{\max})$

Compute $W_1 = \{x \in S \,|\, m(x) > 0\}$
Compute $W_\cap(\delta)$ using W_1
Compute $W_f(\delta, a)$ using W_1
Evaluate $\Omega_\lambda(\delta, a)$ using Equations (2) and (3)
Find $\delta_{\max}, a_{\max}$ maximizing Ω_λ
return Maximizers $\delta_{\max}, a_{\max}$ and value $\Omega_\lambda(\delta_{\max}, a_{\max})$;

Algorithm 1: Optimization of Ω_λ

Again, there is much theory and detail underlying these methods, most of which may be found in [20, 28]. 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, this property could ultimately be used for more accurate and efficient prediction of, for example, workspace boundaries.

In this paper, it is adequate to accept homotopy continuation as a numerical method that will quickly provide accurate approximations to all isolated solutions of a polynomial system. Several software packages are available for these sorts of computations. We use a software package named *Bertini* [29], which has been under development for the past decade by D. Bates, J. Hauenstein, A. Sommese, and C. Wampler. The repeated calls to *Bertini* were parallelized for efficiency using the method described in [30].

5 Two Examples

5.1 2D case: Two link planar robots

We start with the illustration of the method for the two dimensional example that is shown in Fig. 1. This was first considered this [31]; the previous work has been expanded to include more general method and examples, as in Section 5.2, where the more challenging three-dimensional problem is considered.

The robots are identical, both having two joints, all link lengths having length one; the DH parameters are summarized in Table 1.

Table 1. Denavit-Hartenberg Parameters for two-link planar robot.

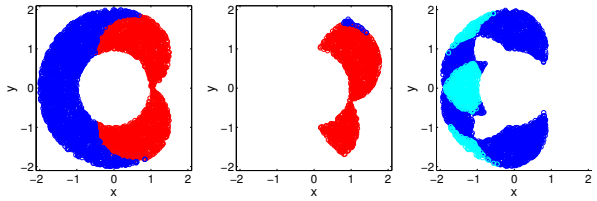
θ_j	α_j	a_j	d_j	$\theta_{j,min}$	$\theta_{j,max}$
θ_1	0	1	0	-120°	120°
θ_2	0	1	0	-120°	120°

Here, we define c_j and s_j to be the cosine and sine of the joint θ_j values, respectively, with $j = 1, 2$ corresponding to the failed robot and $j = 3, 4$ corresponding to the assisting robot. Let joint index $j \in \{1, 2, 3, 4\}$, so that we have a sine-cosine pair for each of the two joints in the two robots. In this notation, our equations for this step are,

$$0 = \begin{cases} c_1c_2 - s_1s_2 + c_1 - x = 0 \\ s_1c_2 + c_1s_2 + s_1 - y = 0 \\ ac_1c_2 - as_1s_2 + c_1 - (c_3c_4 - s_3s_4 + c_3 + \delta) = 0 \\ as_1c_2 + ac_1s_2 + s_1 - (s_3c_4 + c_3s_4 + s_3) = 0 \\ s_j^2 + c_j^2 - 1 = 0 \qquad \forall j \end{cases} \tag{4}$$

Table 2. Estimates of 2D workspaces and $\Omega_{1/3}$, grasping on the second link.

δ	a	$\lvert W_1\rvert$	$\lvert W_\cap\rvert$	$\lvert W_f\rvert$	$\Omega_{1/3}$
0.276	0.076	12.79	7.5810	4.424	4.687
0.276	0.406	12.79	7.5810	7.946	7.035
0.276	0.736	12.79	7.5810	9.927	8.356
1.606	0.076	12.79	1.8904	6.241	7.795
1.606	0.406	12.79	1.8904	5.622	7.383
1.606	0.736	12.79	1.8904	4.718	6.780
2.936	0.076	12.79	0.2031	2.119	5.610
2.936	0.406	12.79	0.2031	2.829	6.083
2.936	0.736	12.79	0.2031	3.635	6.621



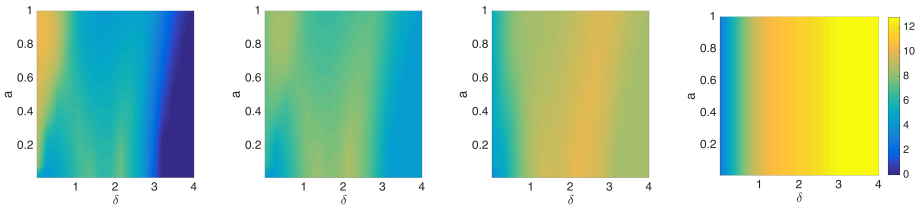
(a) W_1 (b) $W_\cap$ (c) W_f

Fig. 2. Examples of joint-limited workspaces, for $\delta = 1.87$, $a = 1$, and grasping on the *first* link, rotated relative to one another. Red indicates a point having one solution, blue indicates two, and cyan indicates four.

To demonstrate the method, we show results for an initial sampling of $r = 10^4$ points, taken from the two dimensional rectangle $Q \in [-2, 2] \times [-2, 2]$. Of these points, 7836 had at least one real solution; the estimated *Euclidean* size of the workspace would be $4^2 \times 7836/10000 \approx 12.5 \approx 4\pi$. The multiplicity measure of the joint-limited robot, sieved during post-processing, is ≈ 12.79.

The first step in the computation is to estimate the pre-failure workspaces for each robot, as described in Algorithm 1. Secondly, we intersect and obtain $W_\cap = W_1 \cap W_2$, for a specified set of δ values. Finally, for each pair of values (δ, a) for which we estimate the post-failure workspace, we solve Equations (4).

As the separation increases so that the workspaces barely overlap, Robot 2 may only grasp Robot 1 near the end effector, and the resulting W_f is small. See Fig. 2, and column four of Table 2. In these plots, cyan area is fully accessible from 4 configurations, blue corresponds to 2, and red is reachable from only one configuration. The inaccessible area generally increases when δ is increasing, and 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 method computes explicitly the number of configurations reaching the desired point in a post-failure workspace, even in the case when the configura-



(a) Ω_0. (b) $\Omega_{1/3}$. (c) $\Omega_{2/3}$. (d) Ω_1.

Fig. 3. Objective function contours for a grid of (δ, a) pairs for 2D cooperating robots, with the broken robot being grasped on the second link.

tions belong to isolated domains in the workspace. Such problems are usually challenging for traditional methods of workspace computation.

Finally, in Fig. 3 we show Ω_λ from Equation (2) to determine the optimal ball joint placement. For weighting factor $\lambda = 0$, Ω_0 is simply the value of $|W_f|$, and the maximizer is $\delta \approx 0$, and a has a range of maximizing possibilities. With $\lambda = 1/3$, the landscape is relatively flat. Increasing λ further makes the size of the intersection workspace overpower the post-failure workspace, and by $\lambda = 1$ the maximum of Ω_1 is invariant with respect to a. Therefore, the weighting factor λ plays a critical role in the determination of the optimal grabbing point.

5.2 3D Case: Three Joint Manipulators

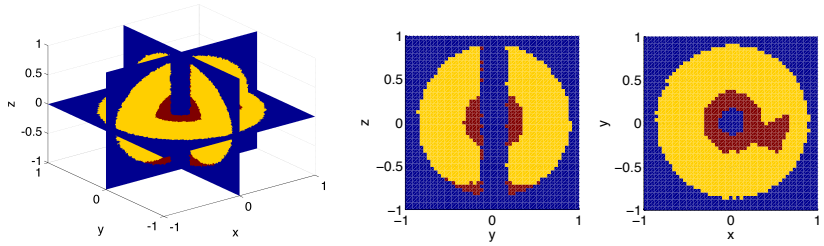
Equations determining the kinematics of spatial robots are more complicated than those of planar manipulators. This concerns both the higher number of relevant equations, due to the higher number of links in a typical 3D robots, and the structure of equations describing the motions. Yet, the method for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

Consider two cooperating PUMA 6-degree-of-freedom robots, ignoring orientation of the end effector. In this paper, we use DH parameters defined as in Table 3. Treating the first three links of the PUMA as our robot, we ignore the wrist; we do this to reduce the dimension and to 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.

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. In order to find $W_{1,2}$ we sample randomly an oversized rectangular box surrounding the robot. In order to determine good bounds for the workspaces, we first do forward kinematics by

Table 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	$-266°$	$266°$



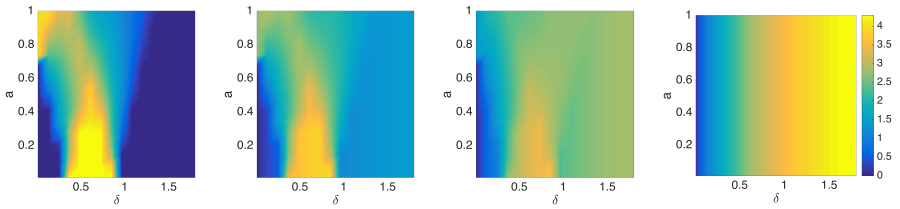
(a) (b) (c)

Fig. 4. Top: Slices of the pre-failure PUMA workspace W_1 by the planes $x = 0$ (b) and $z = 0$ (c). (a): 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 3. Red color: real solutions violating joint limits. Dark blue region: inaccessible.

randomly sampling the three joint variables $\theta_j \in [0, 2\pi]$. 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 3. Because joint limits are written as inequalities $\theta_{j,min} \leq \theta_j \leq \theta_{j,max}$, they are not algebraic equations. We simply use these joint limits in post-processing, only selecting the suitable joint angles among all real solutions.

Results for various workspace computations are shown in Fig. 4. These data are based on 10^4 points in (x, y, z) space. Of particular interest is the presence of voids inside of workspaces, plotted by gridding the Monte Carlo data. Unreachable 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 realization we have chosen makes it hard to visualize 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



(a) Ω_0. (b) $\Omega_{1/3}$. (c) $\Omega_{2/3}$. (d) Ω_1.

Fig. 5. 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)-c) Contours of the objective function Ω versus (δ, a) for 3D cooperating robots. Note that for the case a), $\lambda = 0$ and the objective function Ω is equal to the measure of post-failure workspace $|W_f|$.

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 Fig. 5. To calculate Ω_λ, we compute each of the workspaces W_1, $W_\cap$ and W_f. For the purposes of generating this figure, $W_\cap$ 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 $7.68 \cdot 10^6$.

The Ω_λ landscapes presented in Fig. 7 for the PUMA robot are similar to those for the 2D planar robot above in Fig. 3. Because each joint could fail independently of the others, we 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 for each joint are similar, so only those for the second link are presented here.

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 $(W_1 - |W_\cap|)$ or $|W_f|$ respectively. The maximum values of Ω occur with $\delta \approx 0.9$ meters between the robots. Around $\lambda = 1/3$, we put more emphasis on increasing the remainder of workspace accessible in grasping configuration; Ω clearly indicates to grasp near the end of the second link. In any case, the optimal distance and grasping location depends on the link and on λ, making user preference crucial in the optimization procedure.

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.

By using algebraic geometric methods, we avoid issues such as isolated domains and multiple solutions to inverse kinematics equations. Knowing multiple solutions may be important, especially for obstacle avoidance, where one or more solutions may not be collision free. 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 may 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) 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.

The method we present could be complemented by the software in [32]; while their focus is not on failure tolerance, their interactive CAD workspace mapper uses a Jacobian method to find singularities and determine workspace boundaries of parallel manipulators, which could be supplemented by homotopy continuation.

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. Until recently, it was nearly impossible to detect positive-dimensional *real* solutions. However, recent advances such as the software Bertini_real [33, 34] which implements numerical real algebraic curve and surface decompositions, have made it possible to compute algebraic objects including singularities. Above dimension two, the curse of dimensionality continues to constrain techniques.

References

1. C. Carreras and I. D. Walker, "Interval methods for fault-tree analysis in robotics," *IEEE Trans. Robot. Automat.*, vol. 50, no. 1, pp. 3–11, 2001.
2. 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," *Int. J. Manuf. Res.*, vol. 2, no. 3, pp. 279–302, 2007.
3. M. Ji and N. Sarkar, "Supervisory fault adaptive control of a mobile robot and its application in sensor-fault accommodation," *IEEE Trans. Robot.*, vol. 23, no. 1, pp. 174–178, 2007.
4. 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.*, May 2009, pp. 3359–3364.
5. D. Brambilla, L. Capisani, A. Ferrara, and P. Pisu, "Fault detection for robot manipulators via second-order sliding modes," *IEEE Trans. Ind. Electron.*, vol. 55, no. 11, pp. 3954–3963, 2008.

6. K. N. Groom, A. A. Maciejewski, and V. Balakrishnan, "Real-time failure-tolerant control of kinematically redundant manipulators," *IEEE Trans. Robot. Autom.*, vol. 15, no. 6, pp. 1109–1116, 1999.
7. A. A. Maciejewski, "Fault tolerant properties of kinematically redundant manipulators," in *Proc. IEEE Int. Conf. Robot. Automat.*, Cincinatti, OH, USA, 1990, pp. 638–642.
8. R. G. Roberts and A. A. Maciejewski, "A local measure of fault tolerance for kinematically redundant manipulators," *IEEE Trans. Robot. Automat.*, vol. 12, no. 4, pp. 543–552, 1996.
9. M. Hassan and L. Notash, "Optimizing fault tolerance to joint jam in the design of parallel robot manipulators," *Mech. Mach. Theory*, vol. 42, pp. 1401–1407, 2007.
10. J. E. McInroy, J. F. O'Brien, and G. W. Neat, "Precise, fault-tolerant pointing using a stewart platform," *IEEE/ASME Trans. Mechatronics*, vol. 4, no. 1, pp. 91–95, 1999.
11. 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. Autom.*, vol. 9, no. 5, pp. 675–684, 1993.
12. Y. Yi, J. E. McInroy, and Y. Chen, "Fault tolerance of parallel manipulators using task space and kinematic redundancy," *IEEE Trans. Robot.*, vol. 22, no. 5, pp. 1017–1021, 2006.
13. C. J. J. Paredis and P. K. Khosla, "Designing fault-tolerant manipulators: How many degrees of freedom?" *Int. J. Robot. Res.*, vol. 15, no. 6, pp. 611–628, Dec. 1996.
14. F. Aghili and K. Parsa, "A reconfigurable robot with lockable cylindrical joints," *IEEE Trans. Robot.*, vol. 25, no. 4, pp. 785–797, 2009.
15. Y. Chen, J. E. McInroy, and Y. Yi, "Optimal, fault-tolerant mappings to achieve secondary goals without compromising primary performance," *IEEE Trans. Robot.*, vol. 19, no. 4, pp. 680–691, 2003.
16. C. L. Lewis and A. A. Maciejewski, "Fault tolerant operation of kinematically redundant manipulators for locked joint failures," *IEEE Trans. Robot. Automat.*, vol. 13, no. 4, pp. 622–629, Aug. 1997.
17. J. D. English and A. A. Maciejewski, "Fault tolerance for kinematically redundant manipulators: Anticipating free-swinging joint failures," *IEEE Trans. Robot. Autom.*, vol. 14, no. 4, pp. 566–575, 1998.
18. 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. Autom.*, vol. 9, no. 5, pp. 675–684, 1993.
19. A. J. Sommese and C. W. Wampler, "Numerical algebraic geometry and algebraic kinematics," *Acta Numerica* vol. 20, pp. 469–567, 2011.
20. A. J. Sommese and C. W. Wampler, *The numerical solution to systems of polynomials arising in engineering and science.* Singapore: World Scientific, 2005.
21. A. Sommese, J. Verschelde, and C. W. Wampler, "Advances in polynomial continuation for solving problems in kinematics," *J. Mech. Des.*, vol. 126, no. 2, pp. 262–268, 2004.
22. C. W. Wampler and A. P. Morgan, "Solving the kinematics of general 6R manipulators using polynomial continuation," in *Robotics: Applied Mathematics and Computational Aspects*, K. Warwick, Ed. Oxford: Clarendon Press, 1993, pp. 57–69.
23. C. W. Wampler, A. P. Morgan, and A. J. Sommese, "Complete solution of the nine-point path synthesis problem for four-bar linkages," *J. Mech. Des.*, vol. 114, pp. 153–159, Mar. 1992.

24. C. W. Wampler, J. D. Hauenstein, and A. J. Sommese, "Mechanism mobility and a local dimension test," *Mech. and Mach. Theory*, vol. 46, no. 9, pp. 1193–1206, 2011.
25. J. D. English and A. A. Maciejewski, "Measuring and reducing the Euclidean-space measures of robotic joint failures," *IEEE Trans. on Robot. and Autom.*, vol. 16, no. 1, pp. 20–28, 2000.
26. J. D. English and A. A. Maciejewski, "Failure tolerance through active braking: A kinematic approach," *Int. J. Rob. Res.*, vol. 20, no. 4, pp. 287–299, 2001.
27. E. L. Allgower and K. Georg, *Numerical continuation methods, an introduction*. Berlin: Springer-Verlag, 1990.
28. D. J. Bates, , J. D. Hauenstein, A. J. Sommese, and C. W. Wampler, *Numerically solving polynomial systems with Bertini*, Philadelphia: SIAM, 2013.
29. D. J. Bates, J. D. Hauenstein, A. J. Sommese, and C. W. Wampler, *Bertini: Software for Numerical Algebaic Geometry*, 2015. [Online]. Available: http://bertini.nd.edu
30. D. J. Bates, D. A. Brake, and M. Niemerg, "Paramatopy: Parallel parameter homotopy via Bertini," *submitted*, 2015. Software available at `http://www.paramotopy.com`.
31. D. A. Brake, D. J. Bates, V. Putkaradze, and A. A. Maciejewski, "Illustration of numerical algebraic methods for workspace estimation of cooperating robots after joint failure," in *IASTED Technology Conferences*, Pittsburg, PN USA, Nov. 2010.
32. E. Macho, C. Pinto, E. Amezua, and A. Hernndez, "Software tool to compute, analyze and visualize workspaces of parallel kinematics robots," *Advanced Robotics*, vol. 25, pp. 675–698(23), 2011.
33. D. A. Brake, D. J. Bates, W. Hao, J. D. Hauenstein, A. Sommese, and C. W. Wampler, "Bertini_real: Software for real algebraic sets," 2015. Software available at `http://www.bertinireal.com`.
34. D. A. Brake, D. J. Bates, W. Hao, J. D. Hauenstein, A. Sommese, and C. W. Wampler, "Bertini_real: Software for One-and Two-Dimensional Real Algebraic Sets," in *Mathematical Software–ICMS 2014*. pp. 175–182, 2014.