AN ILLUSTRATION OF NUMERICAL ALGEBRAIC METHODS FOR WORKSPACE ESTIMATION OF COOPERATING ROBOTS AFTER JOINT FAILURE

ABSTRACT
We consider the estimation of the post-failure workspace of two two-link serial robots where the free-swinging failure of one robot's last joint is handled by having the functional robot grasp the final link of the broken robot. We present an algorithm for finding the optimal placement of such synergistic robot arms and the optimal grasp point on the final link of the broken robot. To determine whether a point in the pre-failure workspace of the broken robot remains in the post-failure workspace, we solve an inverse kinematics problem in the form of a polynomial system. Homotopy continuation provides an efficient means for solving such polynomial systems, so we use the software package *Bertini*. Finally, Monte-Carlo methods are used to estimate the post-failure workspace.

KEY WORDS
Kinematic redundancy, robot arms, workspace, joint failure, homotopy continuation, Monte Carlo methods.

1 Introduction

Fail-proofing and robustness play an important role in the design of autonomous systems, including robotic arms. While no robot can be completely fail-proof, the probability of several joints failing simultaneously is low. 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. Suppose the system has to operate for an extended period of time, and 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. Should 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. While, in principle, these arms should operate in three dimensions, here, we simplify our considerations for the case of planar arms for the sake of simplicity. Our methods are directly transferrable to three dimensions without substantial alteration. Thus, in this paper, we consider two *two-dimensional two-linked rotary robots* working sufficiently close to each other, such that their workspaces overlap. Then, if a joint on one robot fails, the other grasps a given point on the broken robot, thus restoring some of the lost workspace.

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 clearly problem dependent. In this paper, we use the standard Euclidian measure to quantify the size of the workspace. A choice of 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. These generalizations lie beyond the scope of this paper. Our focus in this paper is to find the optimal separation and grasp point for a two-jointed robot to grasp another two-jointed robot 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 solutions of the polynomial system.

We recognize that, for this particular case, the problem could be solved in almost closed form. However, this traditional approach fails to generalize when additional degrees of freedom are present, *e.g.* higher dimensions or more links, whereas our approach is directly transferrable to higher dimensions.

Consideration of fault-tolerance is essential in dangerous locations, and has been studied extensively, as in [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14], and the references therein. Methods of algebraic geometry have been applied in the kinematic setting as well. See for example [15, 16, 17, 18]. However, analysis of the cooperation of multiple arms via the tools of numeric algebraic geometry has never been explored, making the approach of this article significantly different from all related literature.

In the next section, we provide a formal statement of the general problem we are considering, though the example in this article is less general. Background regarding the numerical solution of polynomial systems may be found in Section 3. Details about our method are described in Section 4, while Section 5 contains an example.

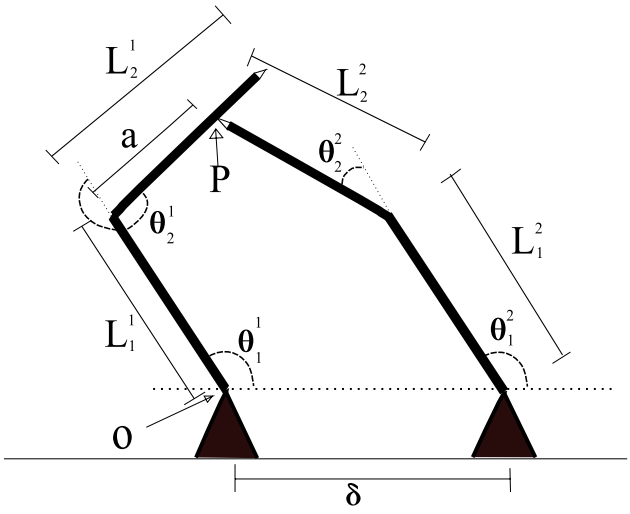


Figure 1. Two robots touching at point P. This is a simplified 2D model of a more general 3D case. In this paper we will not consider the final arm orientation so the workspace is a subset of $\mathbb{R}^2$.

2 Formal Problem Statement

Given two robot geometries, 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 [19]). That is, if one robot has a free swinging joint failure, described as in [20, 21, 22], we would like to ensure that when the functional robot joins at the socket, the resulting cooperative workspace is as large as possible. The geometry of the problem is illustrated at Figure 1. Let the separation distance be δ, and without loss of generality, let this translation between bases be entirely along the x-axis. 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 right robot be robot 2.

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 second frame. A point Q in space 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. Consider two identical planar rotary joint robots offset along the x-axis by distance δ, with each joint length equal. Then, in Figure 1, we have $L_1^1 = L_1^2 = L_2^1 = L_2^2 = L$. Furthermore, normalize link lengths to $L = 1$. Then the DH parameter table for each robot i is identical, and equal to:

$$DH = \begin{array}{c} \begin{matrix} \theta & \alpha & a & d \end{matrix} \\ \begin{pmatrix} \theta_1 & 0 & 1 & 0 \\ \theta_2 & 0 & 1 & 0 \end{pmatrix} \end{array}.$$

In our case, the post-failure workspace, W_f is a subset $\mathbb{R}^2$; if we are considering orientation at each point in this space, we must consider the direct product of $SO(2)$, the set of rotations in two-space, and $\mathbb{R}^2$. However, we neglect the orientation at the end effector of either robot, and consider only the Euclidian measure on $\mathbb{R}^2$, so it makes sense to talk about the size of W_f. 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 as first of all, there would be little reason to have two robots to begin with, and second, δ is usually set by the original design of the system. Thus, we will consider δ to be a fixed parameter of the problem, but examine a set of different δ's to show the dependence of the problem on that parameter. Indeed, each robot has its own independent workspace, W_1, W_2, and when the two robots are placed near enough to each other, there is an intersection workspace $W_\cap = W_1 \cap W_2$. It is clear that $|W_\cap|$ depends on δ.

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 $|W_\cap|$:

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

For a given δ, we seek to maximize Ω in terms of a, which controls the placement of the joining socket.

3 Homotopy Continuation

Because the inverse kinematics problems of the next section can be realized as polynomial systems, our approach is to use the well-developed methods of numerical algebraic geometry, particularly homotopy continuation, to estimate the solutions of these polynomial systems. Given a set of polynomials $f_1, \ldots, f_N$ in N variables for which we seek the solutions, homotopy methods begin by producing and solving some other (related) polynomial system $g_1, \ldots, g_N$ 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. This is a very simple, high-level description of these methods, necessitated by the brevity of this article. Further details may be found in [23, 15].

There is one particular setting in which homotopy methods are particularly effective, 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 for which we will choose a large number of instances), there is an especially

efficient homotopy method known as a *parameter homotopy*. When constructing a starting system $\bar{g}$ as in the previous paragraph, 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 $\bar{p}$, say $\bar{p}^*$ (typically chosen as random complex numbers, for theoretical reasons). This stage may involve a number of wasted paths (as above). However, 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 $\bar{p}$. In other words, for an initial 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 are many theorems and details underlying these methods, most of which may be found in [15].

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* [24].

4 Workspace Computation

4.1 W_i Computation

We compute the workspaces for each robot via a random sampling method. We sample a subset of $\mathbb{R}^2$ that is guaranteed to contain the workspace using a uniform pseudo-random distribution with the built-in MATLAB random number generator. We then solve the inverse kinematics problem at each of these sample points. The inverse kinematics problem is 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* does not distinguish between real and complex solutions to a polynomial system, the points that resulted in real inverse kinematic solutions are inside the workspace, and points that gave only complex solutions are outside the workspace. If we have n samplings, and m 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 m/n. For example, for the robots under consideration here, a reasonable set of bounds for sampling are $P = (P_x, P_y) \in [-2, 2] \times [-2, 2]$. Then if we sampled $\mathbb{R}^2$ $n = 1500$ times, and only $m = 1174$ of these points had at least one real solution, the estimated size of the workspace would be $|W_1| = 4 \times 4 \times 1174/1500 \approx 12.5$.

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 as a separate variable, mapping $\cos(\theta_i) = c_i$ and $\sin(\theta_i) = s_i$. Taking $L = 1$, we write the inverse kinematics equations as:

$$0 = \begin{cases} c_1c_2 - s_1s_2 + c_1 - P_x \\ s_1c_2 + c_1s_2 + s_1 - P_y \end{cases} . \tag{1}$$

These equations are augmented by trigonometric identities,

$$c_i^2 + s_i^2 - 1 = 0, \quad i = 1, 2. \tag{2}$$

Four total equations (1,2), in four unknowns, constitute the polynomial system we will solve using the homotopy continuation solver *Bertini*. We treat P as a parameter.

As described above, the solve is performed in two steps: first, an initial solve for random complex values of P_x, P_y; second, a solve for each of the n samples, with (P_x, P_y) as the sample itself. The advantage of using a parameter homotopy in this case is a reduction of computation time by a factor of four: the initial solve tracks eight paths to find two complex solutions, and the second solve tracks two paths to two solutions. In other words, without using parameter homotopies, each solve would require the tracking of eight paths rather than two.

4.2 $W_\cap$ Computation

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 m points that gave real solutions for W_2. Suppose that of n initial samples, m gave real solutions for W_2, and k correspond to a real solution belonging to W_1, then we can estimate the intersection work space to have the measure

$$|W_\cap| \approx \frac{k}{m}|W_2|.$$

The equations and method for this step of the program are identical to those in Section (4.1), so we will not present them here for brevity.

4.3 W_f Computation

The final component for calculating the objective function is to estimate the post-failure space $|W_f|$. We know that $W_f \subset W_1$, so as for $W_\cap$, we need only consider the m points we computed in W_1. The major difference in this step are the equations. Let $i \in \{1, 2, 3, 4\}$, so that we have a sin-cosine pair for each of the two joints in the two robots. We want to place the robots such that robot 2 touches robot 1 at point P, so for a given value of parameter a, we have a secondary DH table for robot 1:

$$DH = \begin{pmatrix} \theta_1 & 0 & 1 & 0 \\ \theta_2 & 0 & a & 0 \end{pmatrix}.$$

Here, we define c_i and s_i to be the cosine and since, respectively, with $i = 1, 2$ corresponding to the first (failed) robot

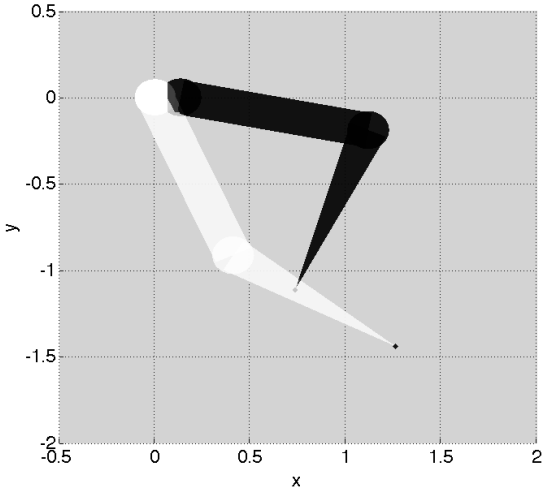


Figure 2. Demonstration of grasping configuration. In this randomly chosen example, $a = 0.381$ for $\delta = 0.133$.

and $i = 3, 4$ corresponding to the second (assisting) robot. In this notation, our equations for this step are,

$$0 = \begin{cases} c_1c_2 - s_1s_2 + c_1 - P_x \\ s_1c_2 + c_1s_2 + s_1 - P_y \\ ac_1c_2 - as_1s_2 + c_1 - (c_3c_4 - s_3s_4 + c_3 + \delta) \\ as_1c_2 + ac_1s_2 + s_1 - (s_3c_4 + c_3s_4 + s_3) \\ s_i^2 + c_i^2 - 1 \end{cases}$$

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

5 Example

The maximum resulting workspace with respect to varying a for a given δ determines the optimal placement of, for example, a grasping location on robot one, to ensure the largest restored workspace in the event of a free-swinging failure of the second joint of robot one.

To demonstrate the method, we show results for a 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, as in Figure (1). For preparation, we sample $\mathbb{R}^2$, uniformly distributed, using the built in MATLAB random number generator. We sample over a rectangle which surely contains the workspace of robot two. In this example, we sampled over $[-2\,,\,2] \times [-2\,,\,2]$. 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.

δ	a	$\lvert W_\cap\rvert/\lvert W_1\rvert$	$\lvert W_f\rvert/\lvert W_1\rvert$
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

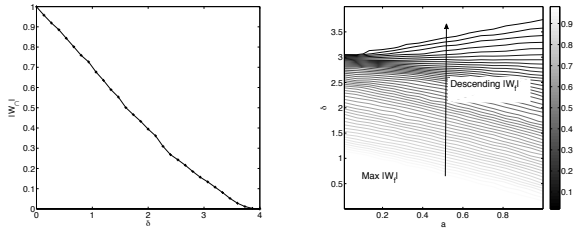
Table 1. Estimates on the normalized size of post-failure workspaces, with robot 2 grasping robot 1.

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 which have solutions for which all variables are real-valued (numerically real-valued, calling any imaginary part less than 10^{-4} zero), are those lying within the workspace. The first step in the computation is to compute the workspace for both robots, as described in Sec. 4.1. Because the robots are identical in this example, there is only one workspace computation. The second step in computation is the intersection of the two workspaces, for a specified set of δ values, as in Sec. 4.2. Finally, for each pair of values δ and a for which we estimate the post-failure grasping workspace, we solve a set of equations as described in Sec. 4.3. 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 appears from this coarse approximation to be monotonically decreasing as we increase the separation distance, and in fact in this case we know it is, because it is merely the intersection of two circles. See Figure 4(a) and column three of Table 1.

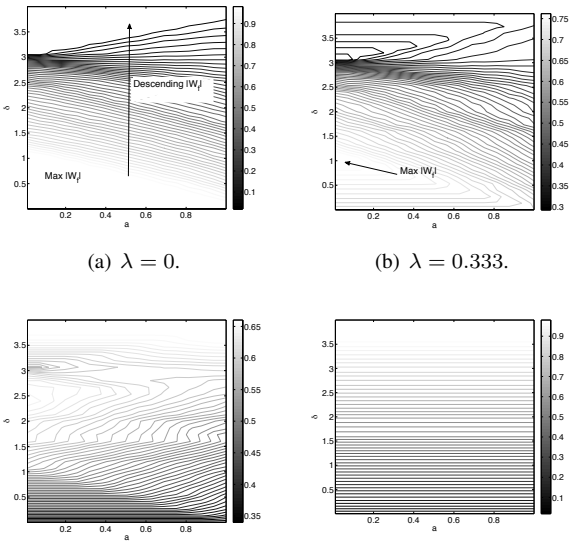
The post-failure grasping workspace is more interesting, and depends both on δ and a. For close placements of robot bases, robot 2 can grasp robot 1 at almost any spot on the second link; W_f is almost identical to W_1. As the robots move further apart, δ increases and the number of points robot 1 can reach while being grasped at P decreases. Not surprisingly, $|W_f|$ is greatest, for $\delta \in [0, \approx 2]$ and $a \approx 0$, see Figure 4(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 only grasp robot 1 near the end effector, and the resulting W_f is small. Also, the computed $|W_f|$ has small variation, even though the shape of the workspace is quite different. See Figure 3, and column four of Table 1. In these plots, the black area is inaccesi-



(a) $|W_\cap|(\delta)$. (b) $|W_f|(\delta,\, a)$.

Figure 4. (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 $(\delta,\, a)$. 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.



(a) $\lambda = 0$. (b) $\lambda = 0.333$.

(c) $\lambda = 0.666$. (d) $\lambda = 1$.

Figure 5. Objective function contours for a grid of $(\delta,\, a)$ pairs.

ble, light gray is fully accessible, and dark gray is reachable from a limited number of configurations.

In particular, among the nine workspaces in Figure 3, the one that is clearly largest is for $(\delta\,, a) = (1, 0.1)$, representing 98% of the original workspace. Among the three workspaces for $\delta = 3$, however, that corresponding to $a = 0.1$ is much smaller; for $a = 0.5$ and $a = 0.9$ the two spaces have almost identical numerical size, with the shape being different.

Finally, when we weight $|W_\cap|$ with $|W_f|$ in our objective function Ω, we can determine an optimal ball joint placement. See Figure 5. 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, and in fact the choice of Ω itself, is important.

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 two-link serial robots placed near one another. The optimal configuration for two *two joint rotary* robots depends on user preference, one example of which was presented. A general algorithm for solving the problem of finding optimal placement and configuration of two such robots was presented.

The work carried out so far has not been comprehensive. For example, we have not taken into account different designs for the two robots (including unequal link lengths) or effects of gravity in the case the two-dimensional workspaces are not orthogonal to the gravity field. However, we anticipate that the general algorithm presented in this article can be expanded with little difficulty to include such considerations. Choices in the algorithm, such as the use of homotopy continuation, have been made to make the generalization of this method to higher-dimensional workspaces feasible. These sorts of generalizations are presently being considered by the authors.

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 another direction of ongoing research.

References

[1] 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), 2003, 680–691.

[2] R. Colbaugh and M. Jamshidi, Robot manipulator control for hazardous waste-handling applications, *J. Robot. Syst.*, 9(2), 1992, 215–250

[3] J. D. English and A. A. Maciejewski, Fault tolerance for kinematically redundant manipulators: Anticipat-

ing free-swinging joint failures, *IEEE Trans. Robot. Automat.*, 14 (4), 1998, 566–575.

[4] K. N. Groom, A. A. Maciejewski, and V. Balakrishnan, Real-time failure-tolerant control of kinematically redundant manipulators, *IEEE Trans. Robot. Automat.*, 15(6), 1999, 1109–1116.

[5] 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), 1999, 91–95.

[6] L. Notash and L. Huang, On the design of fault tolerant parallel manipulators, *Mech. Mach. Theory*, 38, 2003, 85-101.

[7] 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), 1993, 675–684.

[8] Y. Yi, J. E. McInroy, and Y. Chen, Fault tolerance of parallel manipulators using task space and kinematic redundancy, *IEEE Trans. Robotics*, 22(5), 2006, 1017–1021.

[9] 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), 2002, 170–212.

[10] C. J. J. Paredis and P. K. Khosla, Designing fault-tolerant manipulators: How many degrees of freedom?, *Int. J. Robot. Res.*, 15(6), Dec. 1996, 611-628.

[11] C. Carreras and I. D. Walker, Interval methods for fault-tree analysis in robotics, *IEEE Trans. Robot. Automat.*, 50(1), 2001, 3–11.

[12] A. A. Maciejewski, Fault tolerant properties of kinematically redundant manipulators, *Proc. IEEE Int. Conf. Robot. Automat.*, Cincinnati, OH, USA, 1990, 638–642.

[13] R. G. Roberts and A. A. Maciejewski, A local measure of fault tolerance for kinematically redundant manipulators, *IEEE Trans. Robotics Automat.*, 12(4), 1996, 543–552.

[14] C. L. Lewis and A. A. Maciejewski, Fault tolerant operation of kinematically redundant manipulators for locked joint failures, *IEEE Trans. Robot. Automat.*, 13(4), Aug. 1997, 622–629.

[15] A.J. Sommese and C.W. Wampler. *The numerical solution to systems of polynomials arising in engineering and science.* (Singapore:World Scientific, 2005).

[16] A. Sommese, J. Verschelde, and C. W. Wampler, Advances in polynomial continuation for solving problems in kinematics, *ASME J. Mechanical Design*, 126(2), 2004, 262-268.

[17] C. W. Wampler and A. P. Morgan, Solving the kinematics of general 6R manipulators using polynomial continuation, *Robotics: Applied Mathematics and Computational Aspects*, K. Warwick, ed., Clarendon Press, Oxford, 1993, 57-69.

[18] C. W. Wampler, A. P. Morgan, and A. J. Sommese, Complete solution of the nine- point path synthesis problem for four-bar linkages, *Journal of Mechanical Design*, 114(1), March 1992, 153-159

[19] F. Aghili and K. Parsa, A reconfigurable robot with lockable cylindrical joints, *Trans. Rob.*, 25(4), 2009, 785-797.

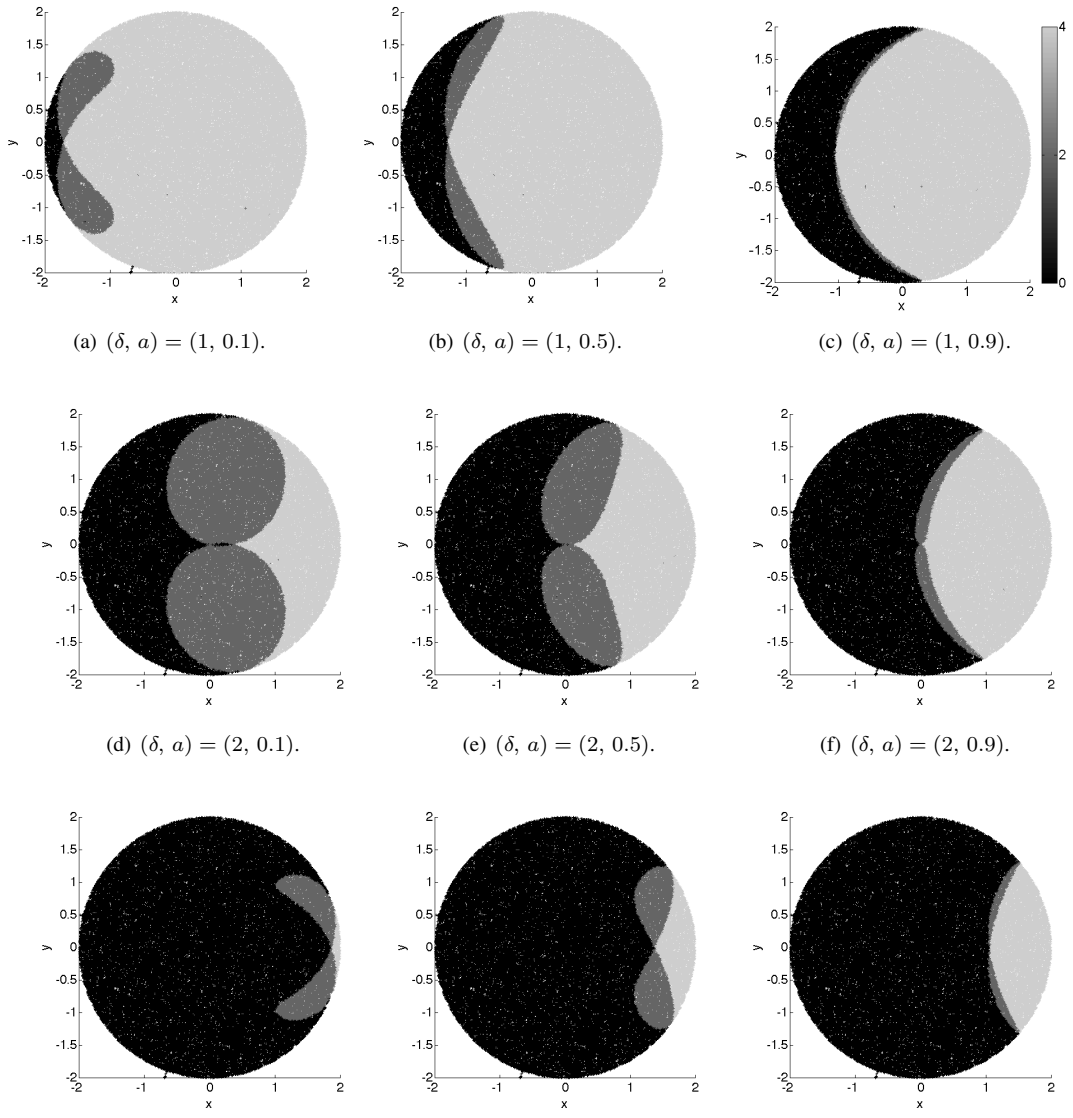
[20] J. English and A. Maciejewski, Robotic workspaces after a free-swinging failure, *Journal of Intelligent and Robotic Systems*, 1997, 19(1), 55-72.

[21] J. English and A. Maciejewski, Measuring and reducing the Euclidean-space measures of robotic joint failures, *IEEE Transactions on Robotics and Automation*, 16(1), 2000, 20-28.

[22] J. English and A. Maciejewski, Failure tolerance through active braking: a kinematic approach, *The International Journal of Robotics Research*, 20(4), 2001, 287-299.

[23] E.L. Allgower and K. Georg. *Numerical continuation methods, an introduction*. (Berlin: Springer-Verlag, 1990).

[24] D.J. Bates, J.D. Hauenstein, A.J. Sommese, and C.W. Wampler. *Bertini: Software for numerical algebraic geometry.* Software available at `http://www.nd.edu/~sommese/Bertini`.



(a) $(\delta,\, a) = (1,\, 0.1)$. (b) $(\delta,\, a) = (1,\, 0.5)$. (c) $(\delta,\, a) = (1,\, 0.9)$.

(d) $(\delta,\, a) = (2,\, 0.1)$. (e) $(\delta,\, a) = (2,\, 0.5)$. (f) $(\delta,\, a) = (2,\, 0.9)$.

(g) $(\delta,\, a) = (3,\, 0.1)$. (h) $(\delta,\, a) = (3,\, 0.5)$. (i) $(\delta,\, a) = (3,\, 0.9)$.

Figure 3. Example W_f. 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, as two per robot.