Chapter 6: Experimental and Simulation Results

Experimentation and simulation verified the effectiveness of the direct search inverse kinematics method for several different robots. The robots in these examples have especially difficult or interesting geometries. They cover a spectrum, from the industrial variety with six degrees of freedom to a conceptual hyper-redundant robot with twenty-one degrees of freedom. Most of the examples include a graph showing the solution's accuracy and timing results for the speed.

All numerical methods find solutions with a finite accuracy. For direct search, the step size for the perturbations determines the accuracy of the solution. A smaller step size results in a more accurate solution. The step size also influences the solution speed. A larger step size increases the solution speed. Clearly there is a trade-off between solution speed and accuracy. The resolution of an industrial robot determined the acceptable accuracy, and thus the step size, for the examples presented in this chapter. A good resolution for an industrial robot is .000254 meters (.001 inches) (Tesar and Butler, 1989a). The accuracy of the direct search solution corresponds to the resolution of the robot if there are no errors other than those introduced by the inverse kinematics method. An accuracy at least ten times better than the resolution of the robot ensures the direct search inverse kinematics solution has little effect on the overall resolution of the robot. An accuracy of .0000254 meters (.0001 inches) should therefore be sufficient.

Solution speed is a practical consideration for inverse kinematics algorithms. This chapter quotes speeds for algorithms executing on a Silicon Graphics R4000 personal workstation. This computer has a single reduced instruction set microprocessor operating at a frequency of fifty-million Hertz. The computer's numerical performance is sixteen MFLOPS (Gorey, 1992). All the inverse kinematics software discussed in this chapter was written in the "C" programming language and is completely portable.

Computer animation played an important part in the development of the direct search inverse kinematics method. This data visualization technique shows the complex kinematic coupling between the robot's joints in a particularly intuitive format.

6.1 ROBOTICS RESEARCH KB2107

This experiment tested the performance of a direct search inverse kinematics algorithm in both simulation and in application to an actual robot. Figure 6.1 shows a photograph of the robot, a Robotics Research kb2107. The experiment determined if application would expose any problems with the direct search technique that were not apparent in simulation. A direct search algorithm generated joint angles off-line as a simulated robot followed several different paths. The actual robot was then made to follow these paths by successively sending the joint angles to the robot's controller. The robot's controller measured and recorded the actual joint angles and joint torques as the robot followed the path. The difference between the desired and the actual joint angles shows how closely the robot was able to follow the path. Because there was no end-effector

sensing, this comparison is valid only at the joint level. The measured torques and several geometric performance criteria are compared along four different paths. These paths were chosen specifically to be difficult for the robot; with two of them being performed at its maximum payload.

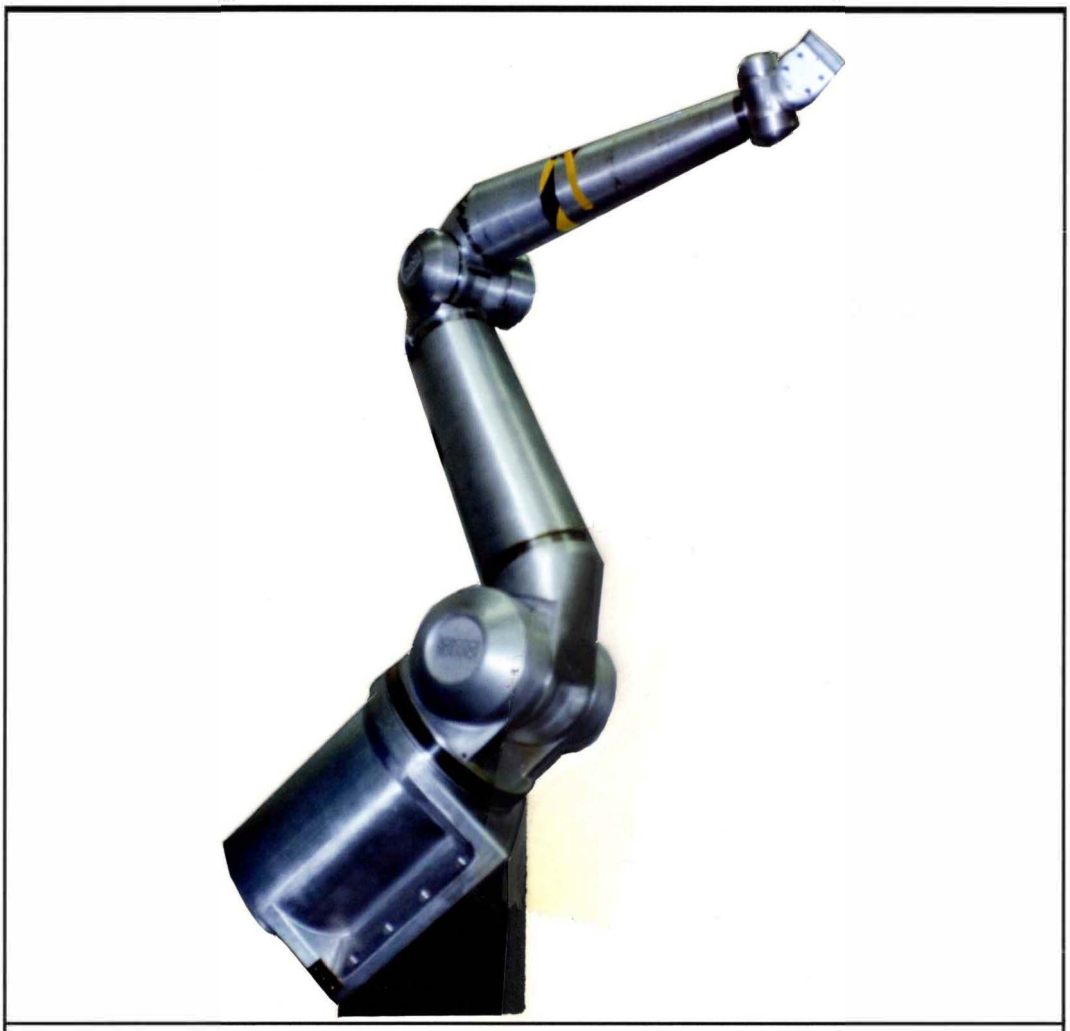

Figure 6.1 The Robotics Research kb2107 robot.

There were several reasons for performing this experiment. The first was to test the numerical performance of the algorithm as to the accuracy and speed of the solution. The second reason for performing the experiment was to bring to light any problems with the direct search solution which were not apparent during simulation. Frequency content that might cause an actual robot to resonate was considered to be a possible problem. Another possible problem concerned unrealistic demands being placed on the robot's controller and actuators. The final goal for this experiment was to measure and record the joint torques as the robot followed the paths. These joint torques were then to be compared with various performance criteria values along the paths.

The direct search algorithm employed an exhaustive exploration pattern to generate the trial solutions. A steepest descent decision making strategy chose a single solution from among the set of trial solutions. As has been discussed, this combination of exploration and decision making is guaranteed to find a local minimum of an unconstrained objective function. Two different objective functions were used, each of which contained a term generated by transforming the Cartesian error into a performance criterion. The difference was in the secondary performance criterion which were added to the objective function. In the first case, the two-norm of the joint speeds was combined with the Cartesian error to form the objective function. In the second case, the two-norm of the change in joint speeds with respect to time was combined with the Cartesian error to form the objective function. In both cases, the direct search was performed to a joint-level resolution of .001 degrees. All scaling factors were derived using the search-based scaling procedure described in Chapter 4. As just described, the

direct search algorithm's solution speed on a Silicon Graphics Indigo R4000 personal workstation was about .6 Hertz. This speed is about fifty times too slow for real-time application. Using one of the suboptimal exploration strategies would significantly increase the speed. For example, Section 6.7 describes results for a robot with nine degrees of freedom using the simple exploration strategy. The solution speed in this example is about ten Hertz. A faster computer would also increase the solution speed.

Table 6.1 Search parameters and results for the kb2107 experiment.

exploration pattern -	exhaustive
decision making strategy -	steepest descent
objective function (two cases) -	Cartesian error and two-norm of joint speeds Cartesian error and two-norm of rate of change of joint speeds
search resolution -	.001 degrees
scaling method -	global
solution speed -	.6 Hertz
maximum translational error -	.000018 meters (.00071 inches)
maximum rotational error -	.000029 radians
degrees of freedom -	7

This experiment used a Robotics Research Corporation kb2107 robot. It is a research robot with seven rotational joints. The robot is rather long at 2.1 meters and has an extremely low maximum payload of five pounds. Nonetheless, the robot's redundancy, instrumentation, and versatile computer interface allow it to be used for testing multicriteria inverse kinematics algorithms. NASA/Johnson

Space Center was gracious enough to provide the robot and the facilities for this experiment.

A discrete path was described at the joint-level by generating multiple sets of joint angles off-line. Every five milliseconds, these angle sets were communicated to the robot's controller as displacement commands at the joint level. Two and one-half milliseconds after each angle set was sent, the joint angles and joint torques were measured and recorded. The joint angles are measured at the motor, rather than at the output of the joint. Unfortunately, this robot employs harmonic drives as speed reducers within the joints. Because harmonic drives are very compliant, and they are between the motor and the output of the joint, there can be significant deviation between the measured and the actual joint angles. This severely limits the usefulness of these measurements. Only an impression of how well the motor and controller were able to track the commanded joint displacements can be obtained. The joint torques, however, are measured with strain gauge bridges at the outputs of each joint. This provides a valid measurement of the torque at the joints.

Two different end-effector paths were used along with the two different objective functions described above to generate four different paths for the robot to follow. The first end-effector path was a figure-8 performed in a vertical plane. This path was chosen because it is kinematically challenging for the robot. The second path was a simple lifting motion performed directly against gravity. Though not challenging kinematically, a five pound weight (the maximum payload of the kb2107) was attached to the robot's end-effector while it followed this path.

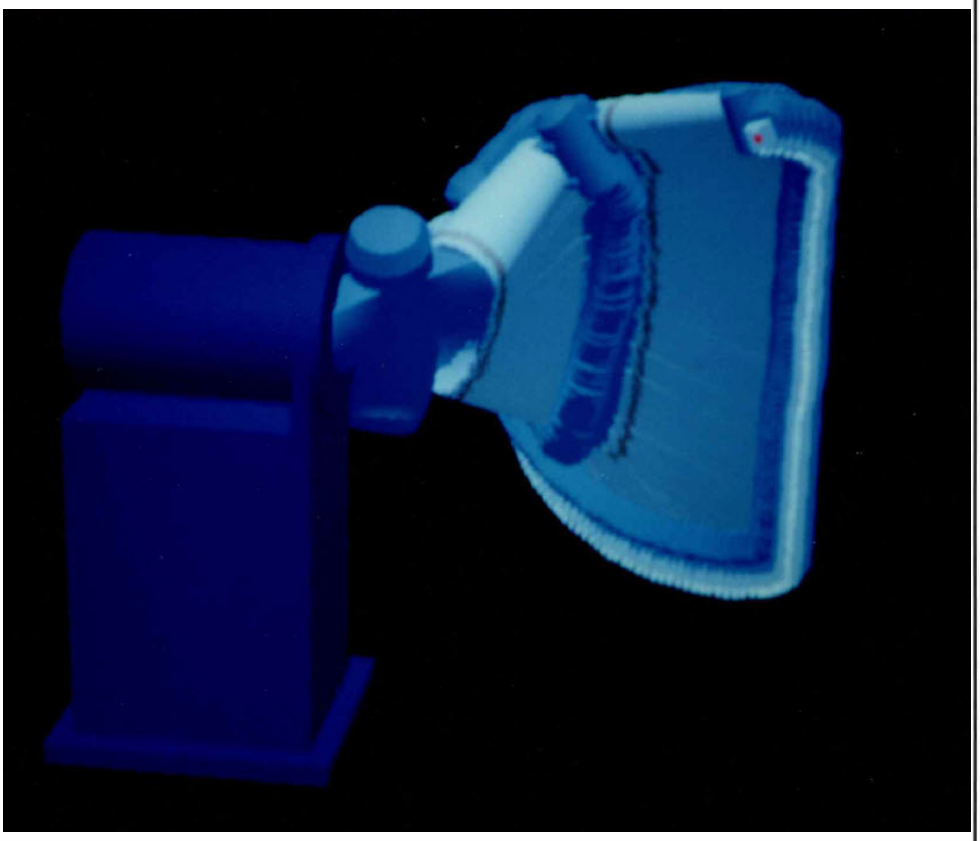

Figure 6.2 A computer generated trace of the Robotics Research kb2107 performing a "D" shaped path.

Figure 6.2 shows a computer generated trace of the robot as it performs a "D" shaped path with the acceleration term in the objective function. Figure 6.3 graphs the numerical accuracy of the solution as the robot follows the path. The maximum translational error is .000027 meters (.00011 inches) and the maximum rotational error is .000019 radians.

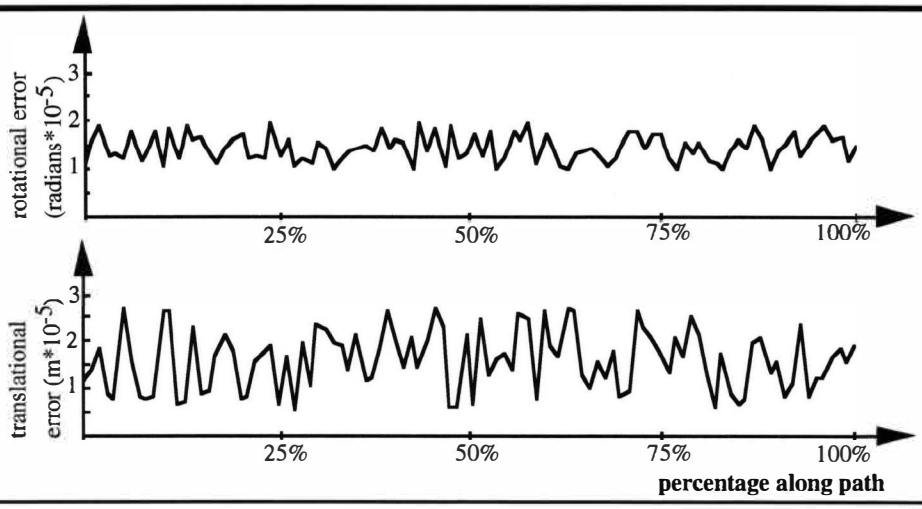

Figure 6.3 The magnitudes of the translational and rotational errors at the end-effector due to the direct search numerical algorithm with an acceleration term in the objective function.

In this example, the resolution of the direct search algorithm is at least an order of magnitude better than the resolution of the robot. Thus, the error due to the direct search will have little effect on the overall resolution of the system. The average difference between the measured and the commanded joint angles was less than .8 percent. This shows that the control system was able to track the input but, as has been discussed, does not measure the actual displacements at the outputs of the joints. Measurement of the joint torques showed that no actuators reached saturation. Figure 6.4 graphs the two-norm of the joint torques along this path. The peak value is 3,900 inch-pounds and the average value is 2,173 inch-pounds. For comparison, four different performance criteria are graphed above the torques in the same figure.

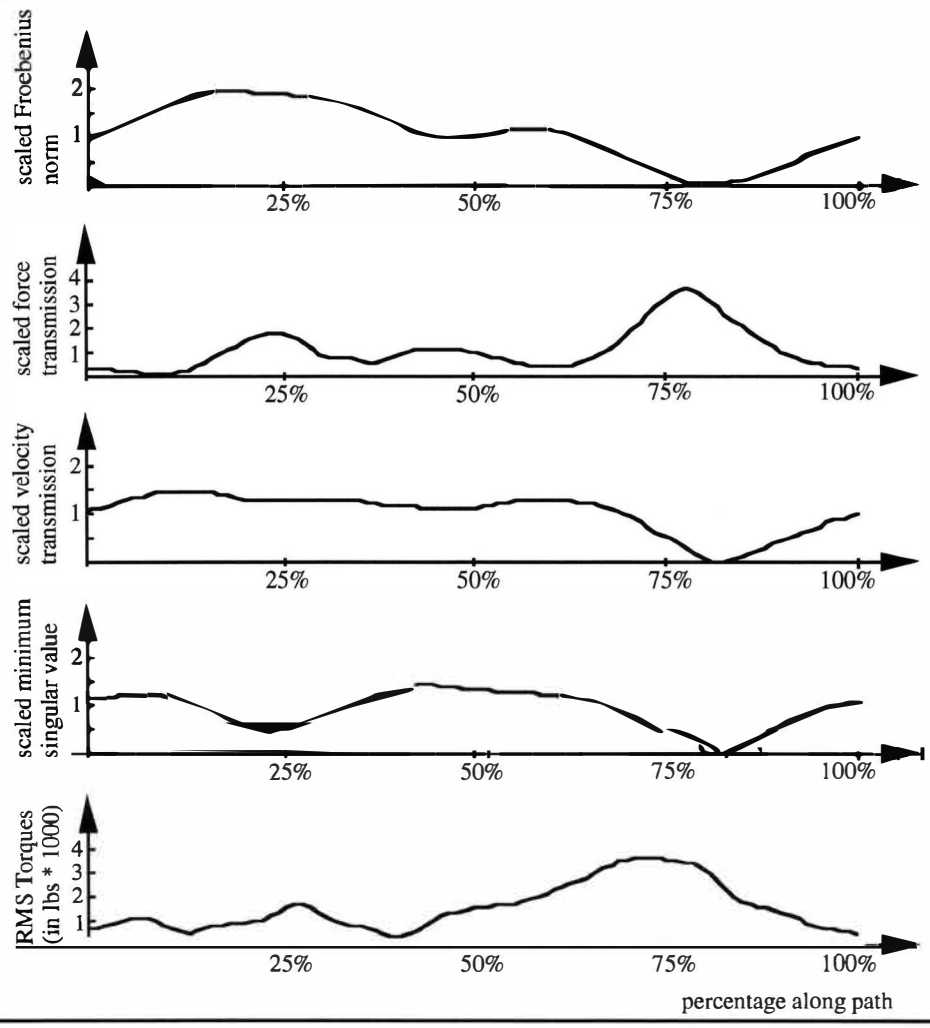

Figure 6.4 The values of the RMS torques and four geometric performance criteria as the robot follows the figure-8 path with the acceleration term in the objective function.

The scaled minimum singular value of the Jacobian is a singularity detection criterion that decreases as the robot approaches a singularity. A larger value of the velocity transformation criterion indicates increased transformation of joint speeds to end-effector velocities. A larger value of the force transmission

criterion indicates increased transmission of joint torques to end-effector forces and torques. The scaled Frobenius norm measures the size of the Jacobian. A complete physical interpretation of these, and the rest of the performance criteria, is very complex and the subject of ongoing research. Though there does seem to be some relationship between the criteria and the torque values, it is not necessarily true that optimizing any of these criteria would reduce torque demands.

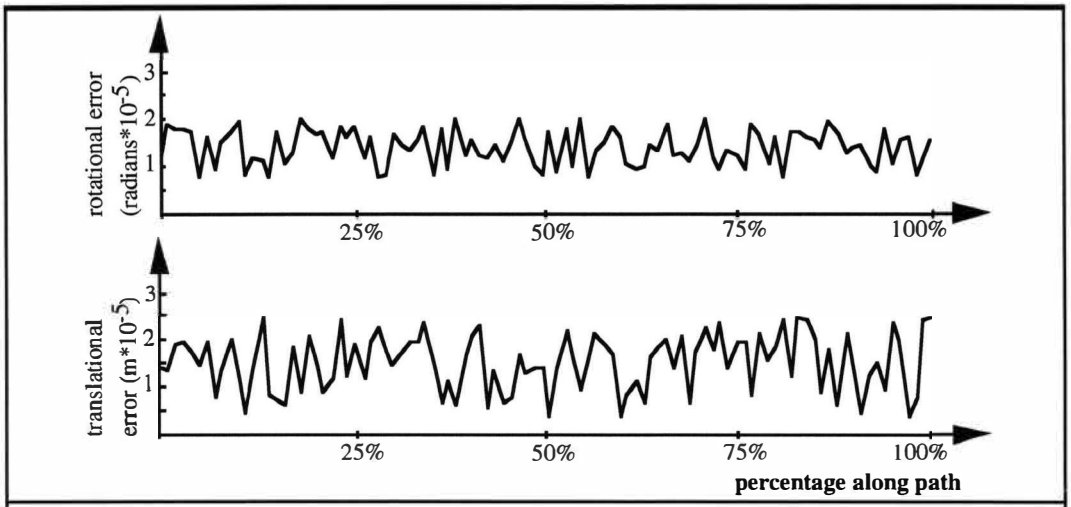

Figure 6.5 The magnitudes of the translational and rotational errors at the end-effector due to the direct search numerical algorithm with a velocity term in the objective function.

Figure 6.5 shows the error of the direct search as the robot follows the same end-effector path, but with a velocity term in the objective function. The maximum translational error is .000025 meters (.00098 inches) and the maximum rotational error is .000021 radians. Again, this numerical error is at least an order of magnitude lower than the resolution of the robot.

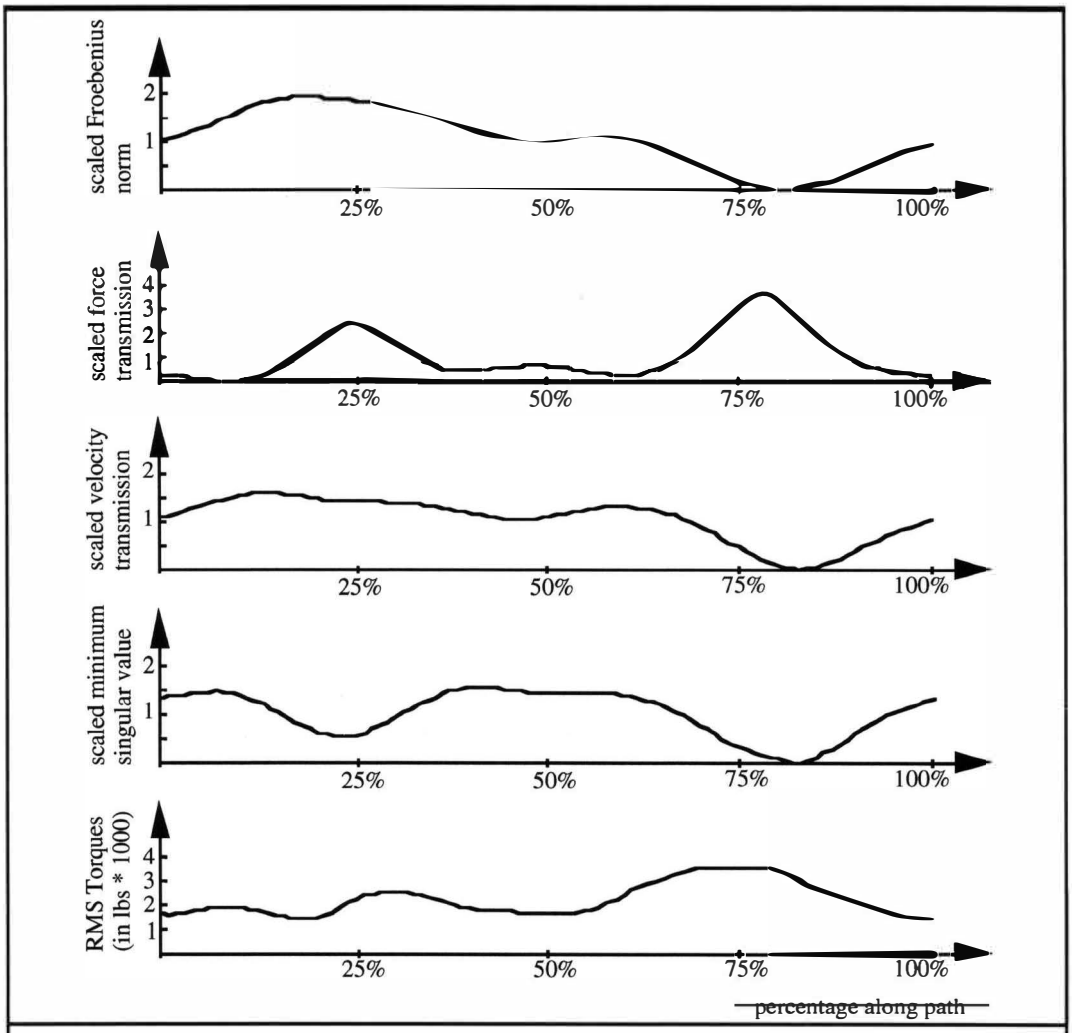

Figure 6.6 The values of the RMS torques and four geometric performance criteria as the robot follows the figure-8 path with the velocity term in the objective function.

The average difference between the measured and the commanded joint angles was again less than .8 percent. The torque measurements showed that no actuator was forced into saturation while the robot followed the path. Figure 6.6 graphs the two-norm of the joint torques as the robot followed the path. The

profile of the torques is very similar to that of the previous path, but the peak and average torques are somewhat lower at 3,500 and 2,160 inch-pounds respectively. The same four performance criteria are graphed above the torques in the same figure. The profiles of these performance criteria values are very similar to those of the figure-8 path with the acceleration term in the objective function.

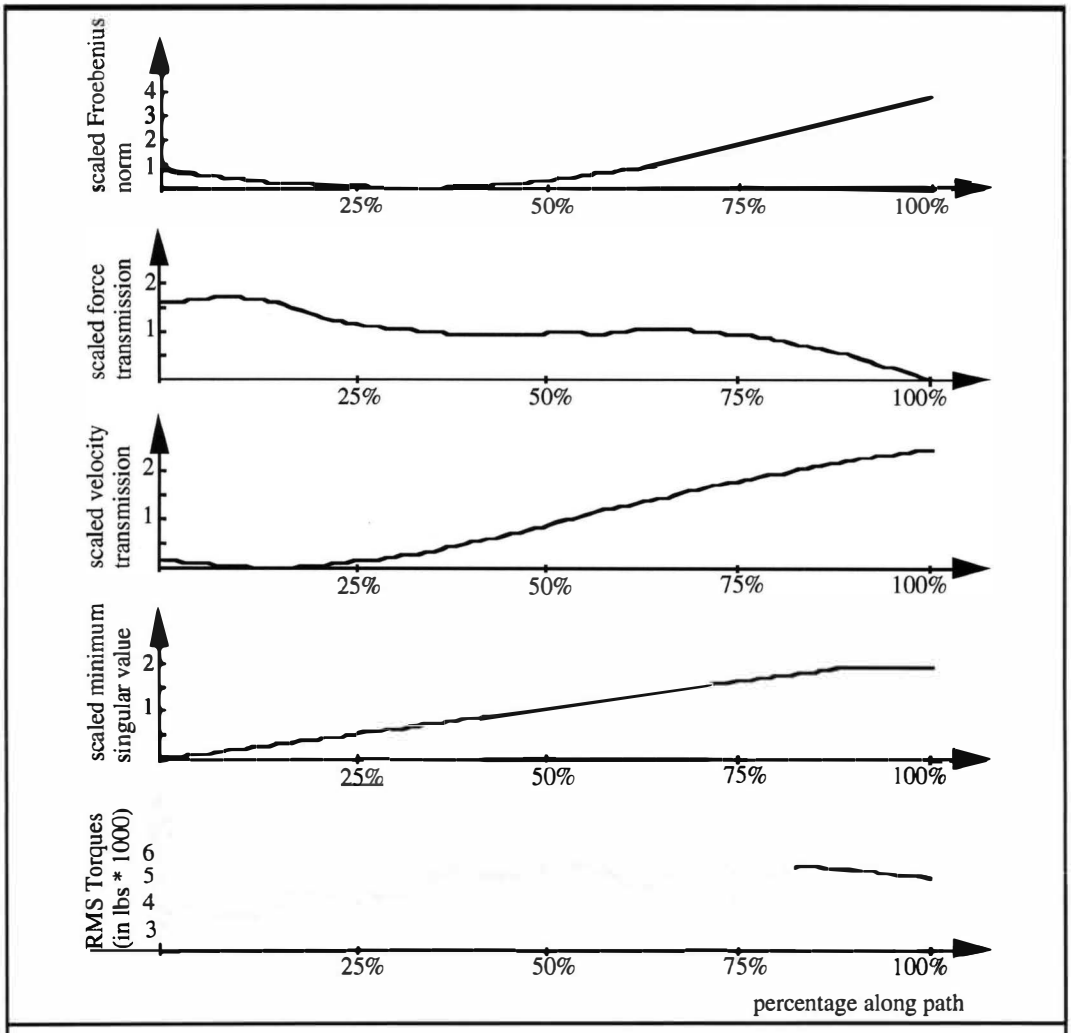

Figure 6.7 The values of the RMS torques and four geometric performance criteria as the robot follows the weight-lifting path with the acceleration term in the objective function.

The second end-effector path was a simple straight-line motion directly against gravity. Kinematically, it was much less challenging. However, the robot's maximum payload of 5 pounds was attached to the end-effector to increase the torque demands on the actuators. The first run was with the acceleration term in the objective function. Measurement of the joint torques showed that no actuators reached saturation. Figure 6.7 graphs the two-norm of the joint torques for this run. As before, the four performance criteria are graphed above the torques. For this weight-lifting path, there is much less correspondence between the performance criteria and the joint torques than there was for the figure-8 path.

The final test in this series of experiments is the weight-lifting path with the joint velocity term in the objective function. Measurement of the joint torques showed that no actuators reached saturation. Figure 6.8 shows the two-norm of the joint torques and the four performance criteria values along this path.

There are several results worth noting from this series of experiments. The first is that for each of these paths the direct search algorithm was able to solve the inverse kinematics problem to a resolution that is sufficient for modern robots. The second observation is that the control system was able to track the commanded inputs very closely without saturating the actuators. The third observation is that the paths with the velocity term in the objective function were somewhat better in terms of the smoothness and magnitudes of the joint torques. The final observation is that the criteria values correlated with the torque demands on the actuators much better for the figure-8 path than for the straight-line path.

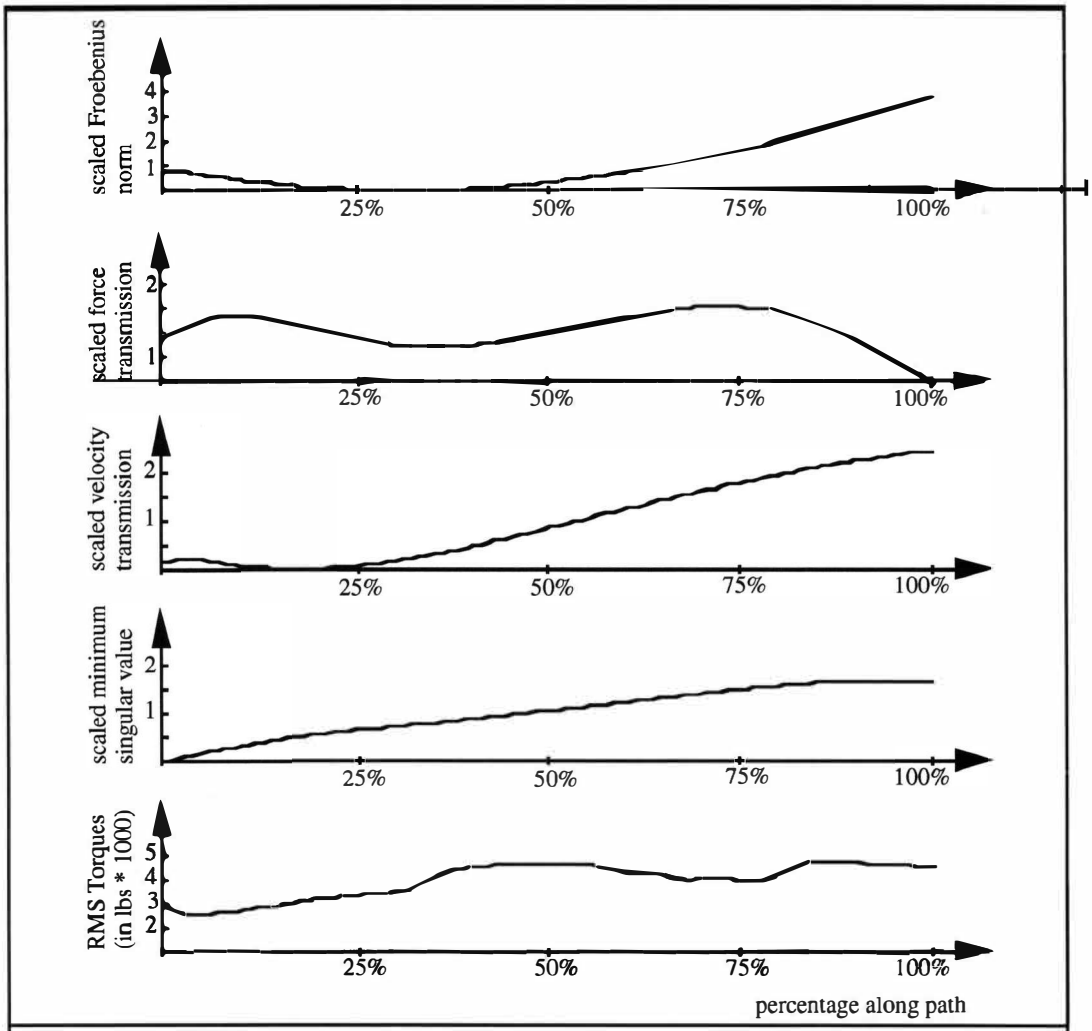

Figure 6.8 The values of the RMS torques and four geometric performance criteria as the robot follows the weight-lifting path with the velocity term in the objective function.

The difference in correlation is likely because of the relative complexity of the two end-effector paths. These geometric criteria are less important in the case of the simple straight-line path. Clearly, a complete physical interpretation of the performance criteria will be necessary before they can be used to optimize the performance of an actual robot.

6.2 LOBSTER'S ARM

A robot with the geometry of a lobster's arm was used for this simulation. A lobster's arm has six degrees of freedom with consecutive pair axes intersecting and mutually perpendicular. It is essentially three Hooke (universal) joints connected in series, as shown in Figure 6.9.

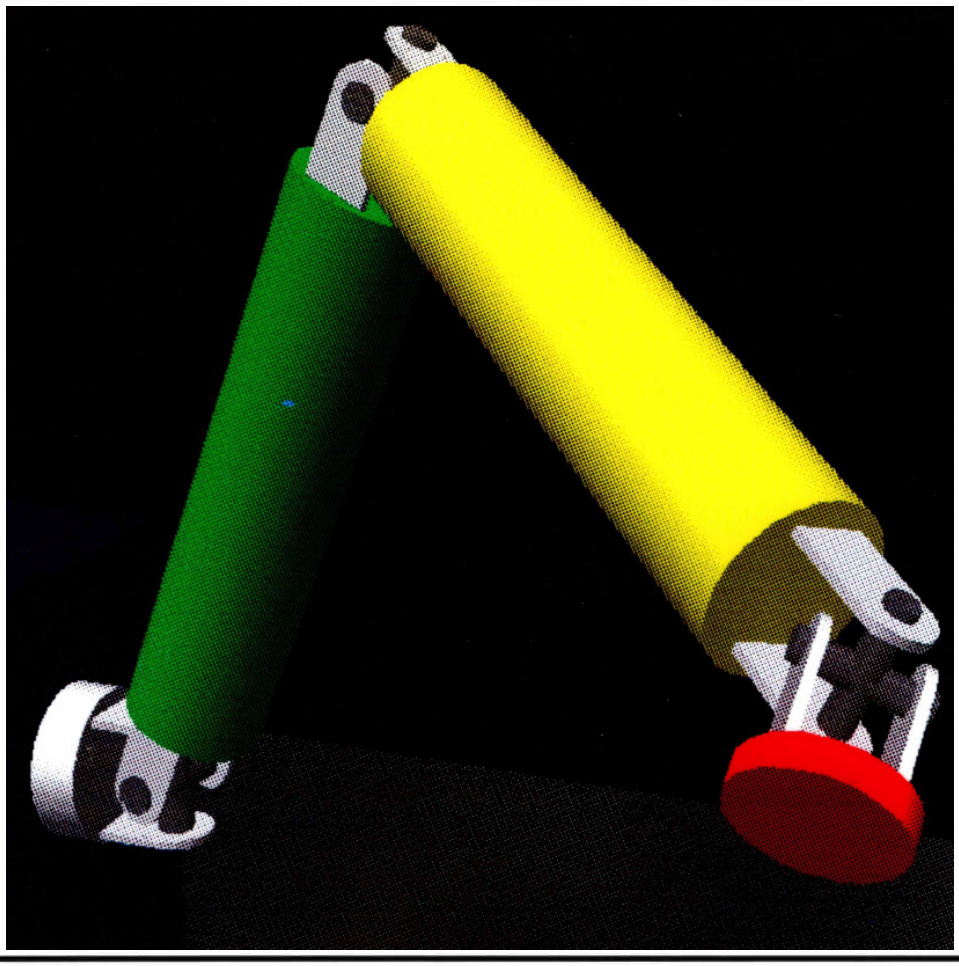

Figure 6.9 A graphical model of a lobster's arm.

Having been first described by Willis in 1841, the geometry of the lobster's arm is of historical interest (Duffy and Derby, 1979). It has also been the subject of current inverse kinematics research (Lin and Duffy, 1989). Removing the requirement that the intersecting axes must be mutually perpendicular forms a generalized lobster's arm. A direct search algorithm simulated this robot following a path somewhat like an actual lobster's arm might follow.

To give some historical perspective, the inverse kinematics problem for the generalized lobster's arm was first solved analytically in 1979 (Duffy and Derby). Ten years before this, Roth and Pieper (1969) had analyzed a variety of geometries with three consecutive axes intersecting at a point. Duffy and Derby believed the analysis of the generalized lobster's arm, with only two consecutive axes intersecting, was another step towards the solution of the inverse kinematics problem for a general six degree of freedom serial robot. In this context, the lobster's arm is indeed interesting. The lobster's arm is also interesting because it has an abundance of singularities scattered throughout the workspace. The singularities reflect the limited dexterity of this geometry, especially in its end-effector orientation capabilities.

An exhaustive exploration pattern and opportunistic decision making formed the direct search strategy for this lobster's arm example. This combination of exploration and decision making is guaranteed to find a local minimum of an unconstrained objective function. The Cartesian error formed the objective function and the direct search proceeded at a joint-level resolution of .001 degrees. A scaling factor for the rotational elements in the objective function was derived using the global scaling procedure described in Chapter 4. As it has just

been described, the solution speed on a Silicon Graphics Indigo R4000 personal workstation was about 2 Hertz.

Table 6.2 Search parameters and results for the Lobster's arm simulation.

exploration pattern -	exhaustive
decision making strategy -	opportunistic
objective function -	Cartesian error
search resolution -	.001 degrees
scaling method -	global
solution speed -	2 Hertz
maximum translational error -	.000022 meters (.00087 inches)
maximum rotational error -	.00012 radians
degrees of freedom -	6

The experiment first described in end-effector coordinates a path something like an actual lobster's arm might follow. Direct search inverse kinematics then found the corresponding joint angles. Observation of actual lobsters in a tank revealed that the geometry of a lobster's legs was essentially the same as that of its arms. For this reason, a path approximating a walking motion was devised. The path is a half circle with its two ends connected by a straight line. The straight line is where the lobster's leg would be in contact with the ground before advancing through the half circle towards another step. The direct search algorithm (described above) generated the joint angles corresponding to this end-effector path and computer animation displayed the results. The numerical error due to the direct search inverse kinematics algorithm was recorded as the simulated lobster's arm followed the path.

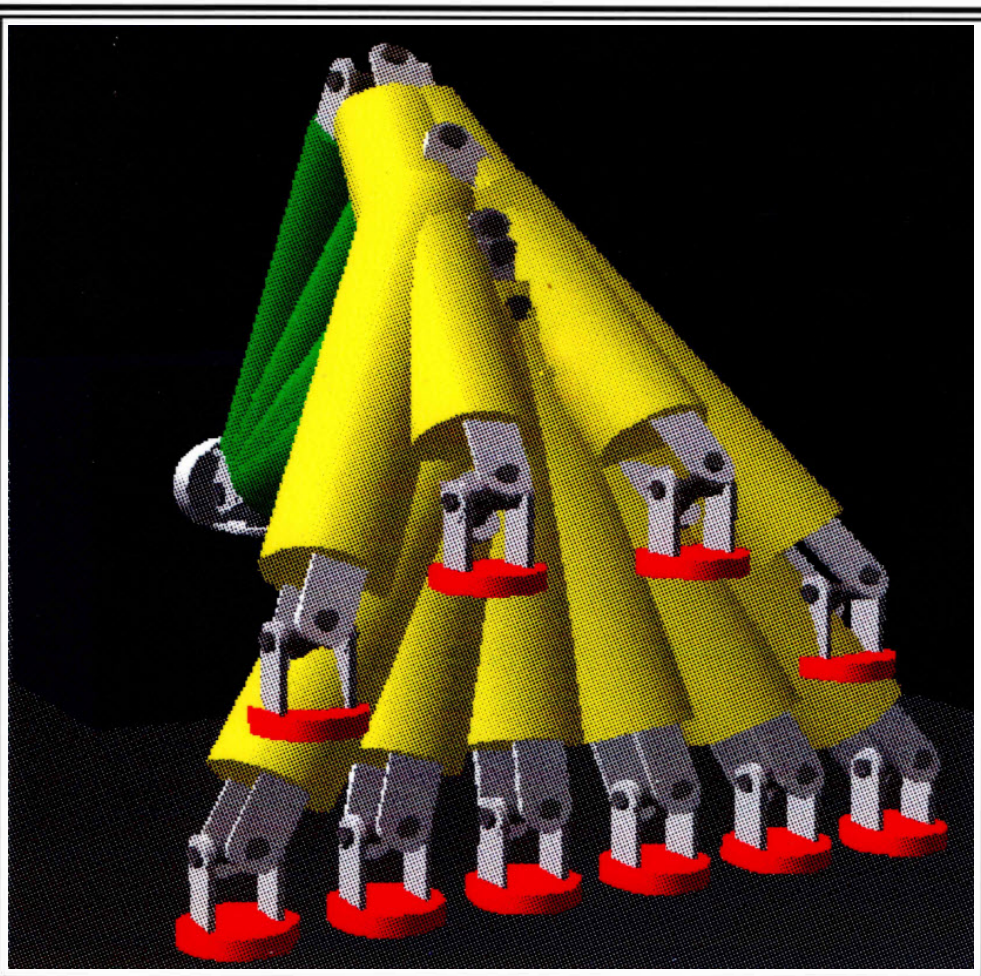

Figure 6.10 A computer generated trace of the lobster's arm following the walking path.

Figure 6.10 shows a computer generated trace of the lobster's arm following the path. The straight-line and semicircle parts are evident. More importantly, Figure 6.11 graphs the magnitudes of the translational and rotational components of the numerical error due to the direct search inverse kinematics algorithm.

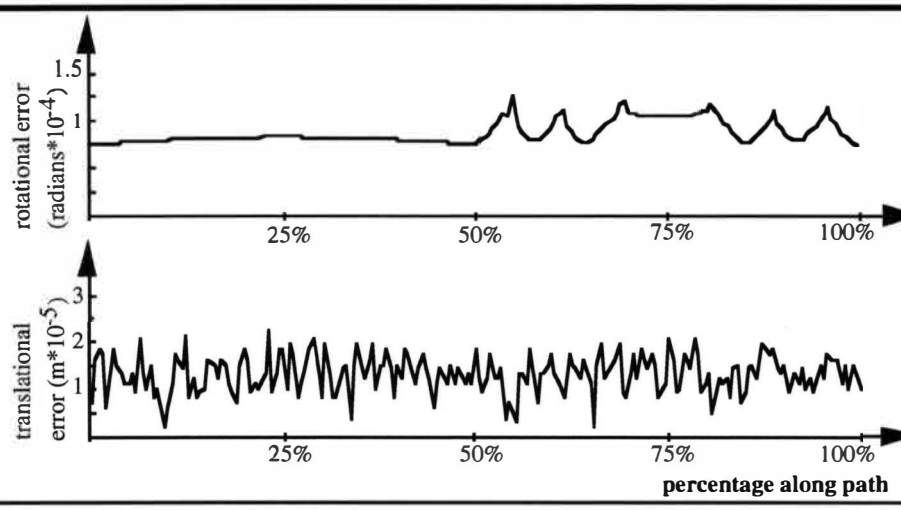

Figure 6.11 The magnitudes of the translational and rotational errors at the end-effector due to the direct search numerical algorithm.

The maximum translational error is .000022 meters (.00087 inches) and the maximum rotational error is .000122 radians. If these errors are unacceptable, the step-size can be reduced and a more accurate solution found, but at the cost of a slower solution speed.

6.3 PUMA

This example demonstrates a direct search inverse kinematics algorithm converging at a singularity. An end-effector path forces the familiar Puma geometry to pass through an exact singularity. Tracking the Cartesian error establishes the direct search converged to the singular solution.

An exhaustive exploration pattern combined with opportunistic decision making formed the direct search strategy for this example. The Cartesian error formed the objective function and the direct search proceeded at a joint-level resolution of .001 degrees. A scaling factor for the rotational elements in the

objective function was derived using the global scaling procedure from Chapter 4. As it has just been described, the direct search algorithm executed on a Silicon Graphics Indigo R4000 personal workstation at an approximate rate of 2 Hertz.

Table 6.3 Search parameters and results for the Puma simulation

exploration pattern -	exhaustive
decision making strategy -	opportunistic
objective function -	Cartesian error
search resolution -	.001 degrees
scaling method -	global
solution speed -	2 Hertz
maximum translational error -	.000058 meters (.0023 inches)
maximum rotational error -	.000011 radians
degrees of freedom -	6

The Puma geometry used in this example has six degrees of freedom and a spherical wrist. This is a very simple geometry and closed form solutions are easily derived. It is chosen mainly because many researchers are familiar with this geometry and its associated singularities. The singularity of interest in this example occurs when the fifth joint passes through the center of its travel. At this point, joint axes four and six are collinear and the robot is in a singular position.

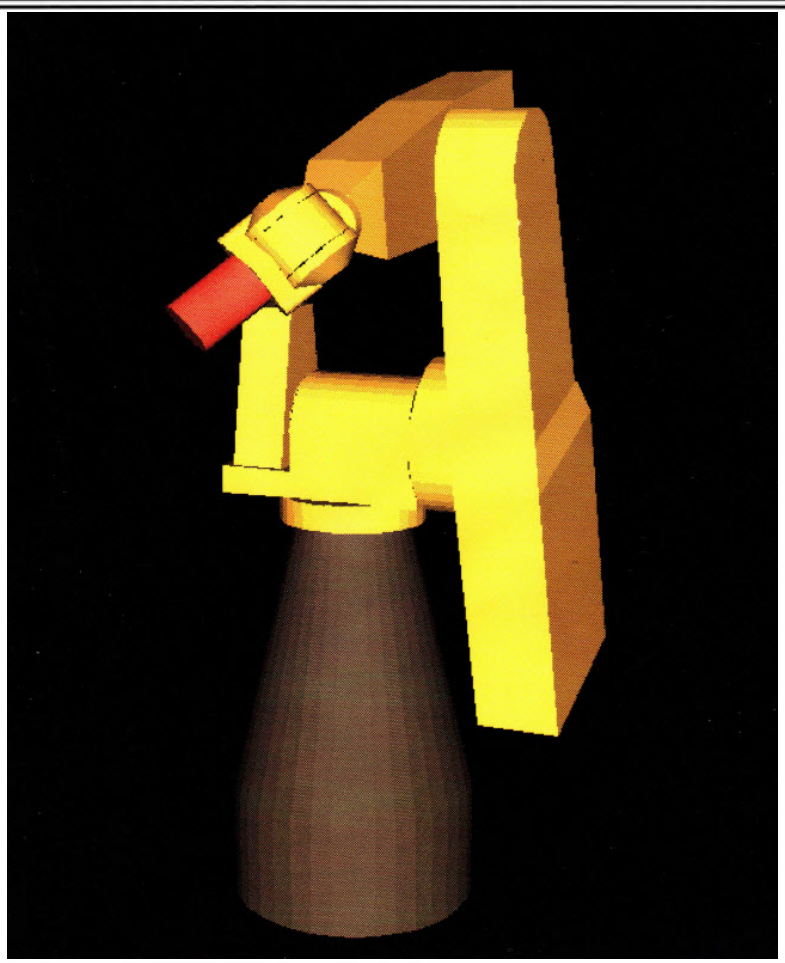

Figure 6.12 A computer generated image of the Puma robot approaching the wrist singularity.

Figure 6.12, shows the simulated Puma robot in the straight line path that passes directly through the wrist singularity described above. Figure 6.13 shows the fifth joint angle, the rotational error, and the translational error from top to bottom respectively. The singularity occurs when the fifth joint angle passes through zero. Examination of the translational and rotational errors shows

that the direct search algorithm did not exhibit any unusual errors as it successfully converged to, and passed through, the singularity.

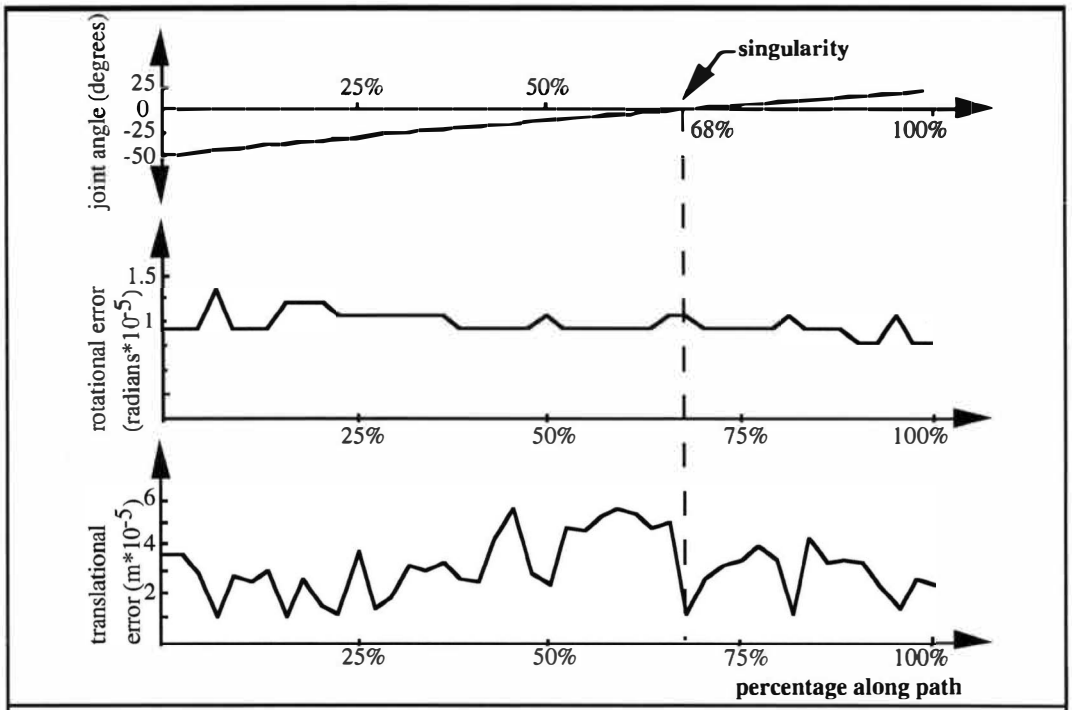

Figure 6.13 The magnitudes of the translational and rotational errors at the end-effector due to the direct search numerical algorithm. A singularity occurs when the joint angle passes through zero.

6.4 ALPHA

The University of Texas Robotics Research group is currently designing and building an advanced seven degree of freedom modular robot. The robot is called ALPHA, which stands for Advanced Lightweight Prototype High-power Arm. The generality of the direct search inverse kinematics method has allowed it to play an important part in the design phase of this robot. The ALPHA robot,

once completed, will also be an important test platform for future multicriteria inverse kinematics efforts.

The University of Texas Robotics Research Group believes that by bringing together all of their research history and expertise; they can build a truly advanced and revolutionary robot. A large part of the available expertise is in the area of dynamic modeling. Dynamic modeling determines the forces and torques throughout the robot. The forces and torques, when coupled with the robot's design, determine the deflections, stresses and actuator requirements. Before the dynamic modeling can take place, however, the joint's displacements, speeds, and accelerations must be determined for a given test path. This is essentially the inverse kinematics problem. The generality of the direct search method allows its application to many different designs as the design team evaluates them.

The ALPHA robot will also be an important test platform for many different leading-edge technologies being developed by the University of Texas Robotics Research Group. Multicriteria inverse kinematics is one of these technologies. Figure 6.14 shows one of the designs for the ALPHA robot currently being considered. The main kinematic feature of interest in this design is the spherical wrist. A robot has a spherical wrist if its last three joint axis intersect at a point. A spherical wrist allows an important geometric simplification in the inverse kinematics solution (discussed later in this section). This simplification significantly increases the solution speed of a direct search inverse kinematics solution. The only issue is solution speed – not solution tractability. The increase in speed should allow direct search to optimize multiple performance criteria in real-time with common computing hardware. These

geometric simplifications detract from the generality of the direct search, but they are currently required if optimizing anything more than very simple performance criteria in a real-time application.

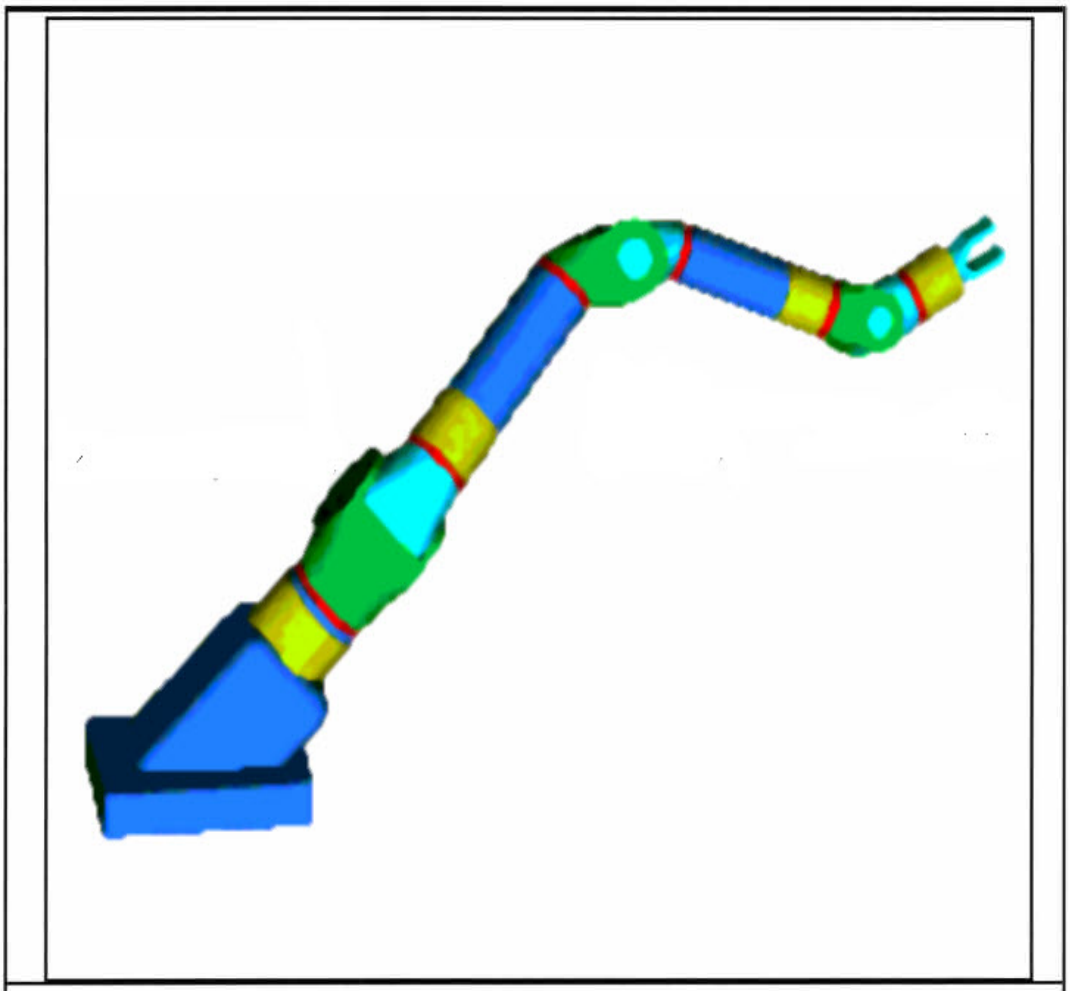

Figure 6.14 A computer graphic of one of the ALPHA robot designs.

This example discusses two different test cases for the ALPHA robot. For both cases, an exhaustive exploration pattern combined with opportunistic decision making to form the direct search strategy. This combination guarantees

local optimality. In the first case, the objective function contained the translational error, the rotational error, and the two-norm of the joint speeds. A direct search performed through all seven joints found the solution. In this case the solution speed was about .6 hertz on a Silicon Graphics Indigo R4000 personal workstation. In the second case, simple geometry calculated the three wrist angles in closed-form. The three wrist angles then satisfied the three rotational end-effector constraints. Since the rotational end-effector constraints are satisfied in closed form, they are removed from the objective function, which now contains just the translation error and the two-norm of joint speeds. Note that all of the joint speeds, including those at the wrist, are included in the objective function and hence the optimization. The wrist angles are simply found in closed form. Thus, the direct search must only search through the first four joints. This simplification increases the speed of execution to about thirty Hertz on the same Indigo computer. This is approximately the speed required for real-time performance, though only one very simple performance criteria is being considered.

For a six degree of freedom robot, both spherical shoulder and spherical wrist geometries have known closed-form solutions. This means that six out of the seven joint angles can be found in closed-form and the search need only be performed over one joint variable. This should allow the real-time application of a direct search inverse kinematics algorithm incorporating multiple complex performance criteria and a sophisticated decision-making strategy. Most of the geometric generality of the direct search inverse kinematics method will be lost, but the closed-form solution can be formulated so as to allow minor changes to

the robot's geometry (such as changing link lengths). Fault tolerance capabilities are also sacrificed because a fault in one of the joints will likely change the closed-form solution. This strategy maintains generality with respect to the performance criteria.

6.5 TITAN 7F

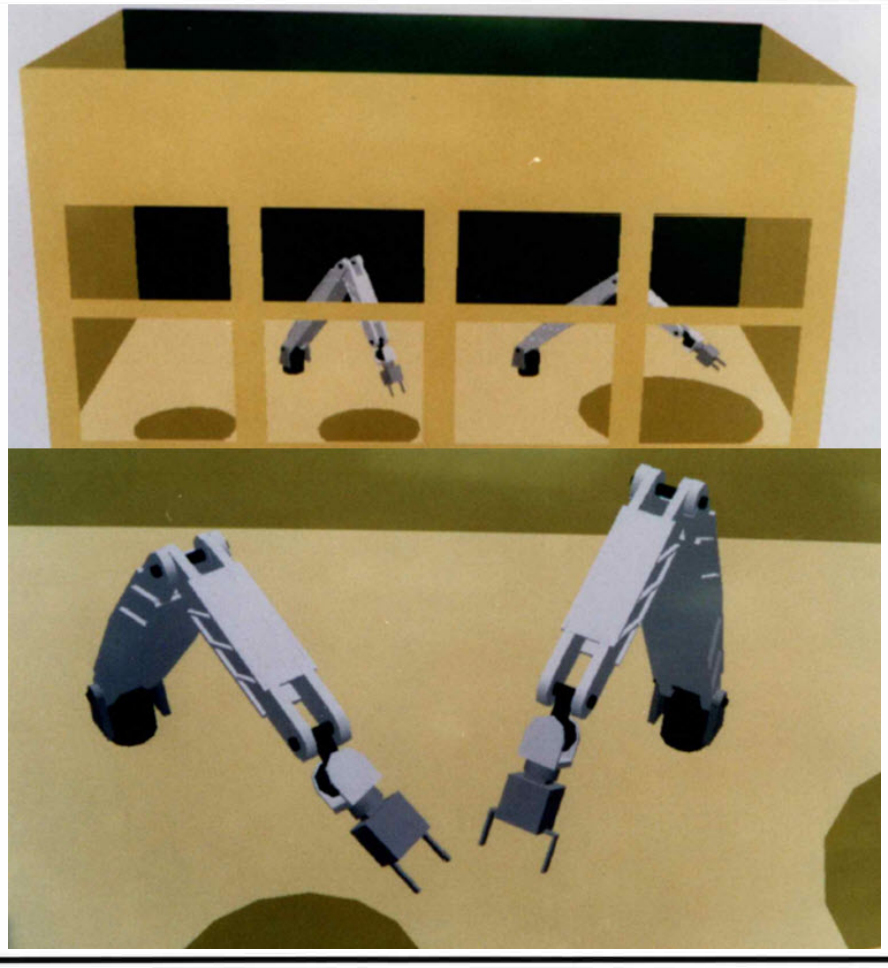

Figure 6.15 A computer graphic of two Titan 7F's in a simulated inspection cell.

A simulation of a robot inspecting nuclear waste was developed in conjunction with Argonne National Laboratory. The robot is a Schilling Titan 7F, which has six degrees of freedom. The task is to push a thin gas-sampling probe through a vent in the top of a fifty-five gallon drum and then withdraw the probe. The probe is 1/16th of an inch and the vent 1/8 of an inch in diameter. Figure 6.15 is one frame of an animation showing the robot performing this task. The gas-sampling task is described in end-effector space. A direct search algorithm that combined exhaustive exploration and opportunistic decision making translated the task into joint space. The interesting aspect of this example is the geometry of the robot. It is actually quite complex. Typically, industrial robots have at least two, and usually three, joint axes intersecting. This is so the robots have known closed-form inverse kinematic solutions.

The Titan 7F has no joint axes intersecting. These robots are supplied with master-slave manual controllers. This type of manual controller is a small kinematic replica of the actual robot. The robot exactly duplicates, in joint space, the motions of the manual controller. The joint angles are simply measured at the manual controller and relayed to the robot. This simulation, however, was created as part of a cell-layout effort. Cell-layout requires task simulation before the robot is put into service.

6.6 Wafer Handling

A direct search inverse kinematics algorithm aided in the development of a robotic semiconductor wafer handling simulation. The simulation was developed in conjunction with Sematech as part of an advanced semiconductor

manufacturing effort. The simulation shows three robots operating in one workcell. The robots move a number of wafers between storage and process equipment. Figure 6.16 shows one frame of an animation of this task. The robots are used instead of humans because of the desire for an extremely clean environment. The wafer handling task was described in end-effector coordinates, so this simulation required inverse kinematics to translate the task into joint space.

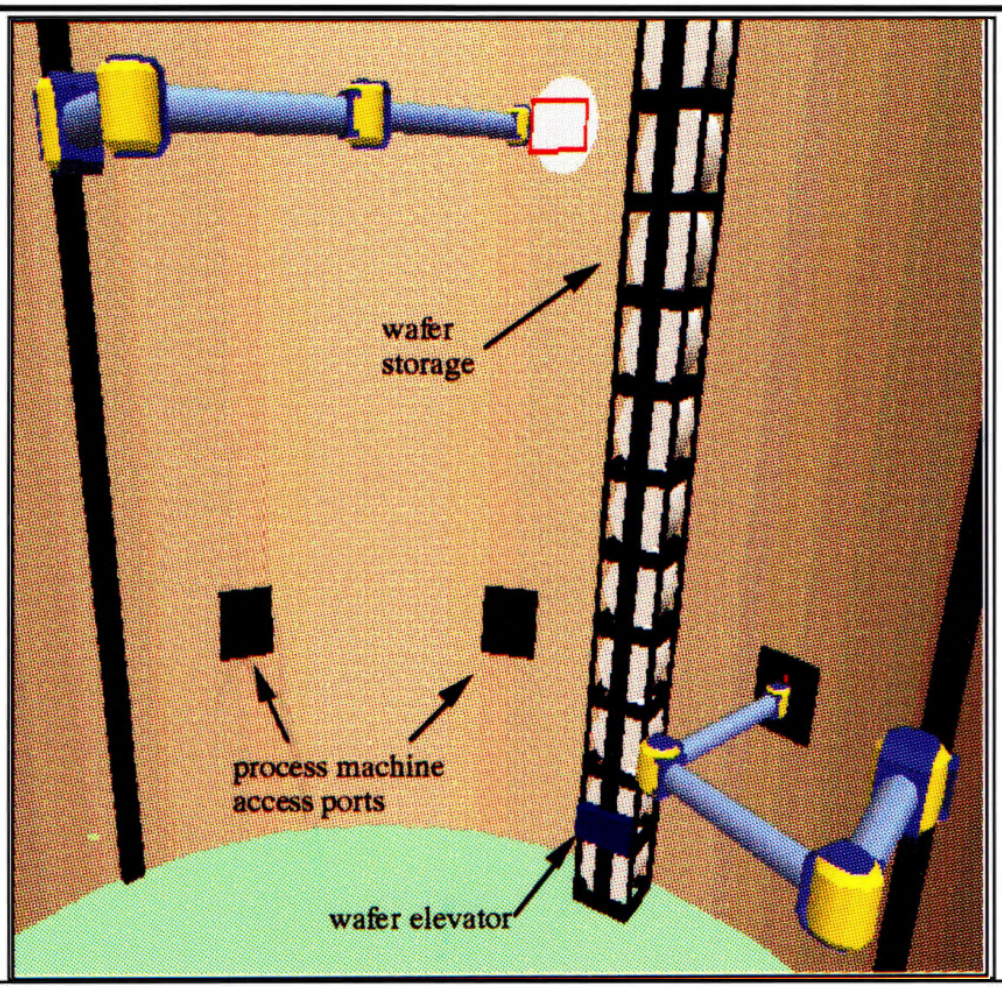

Figure 6.16 A computer graphic of two wafer-inspection robots in a simulated advanced semiconductor manufacturing facility.

6.7 Dexterous Robot with Nine Degrees of Freedom

There are two main features of interest in this example. The first is the robot, which has nine degrees of freedom and therefore three degrees of redundancy. The second feature is the direct search algorithm, which uses a sub-optimal exploration strategy.

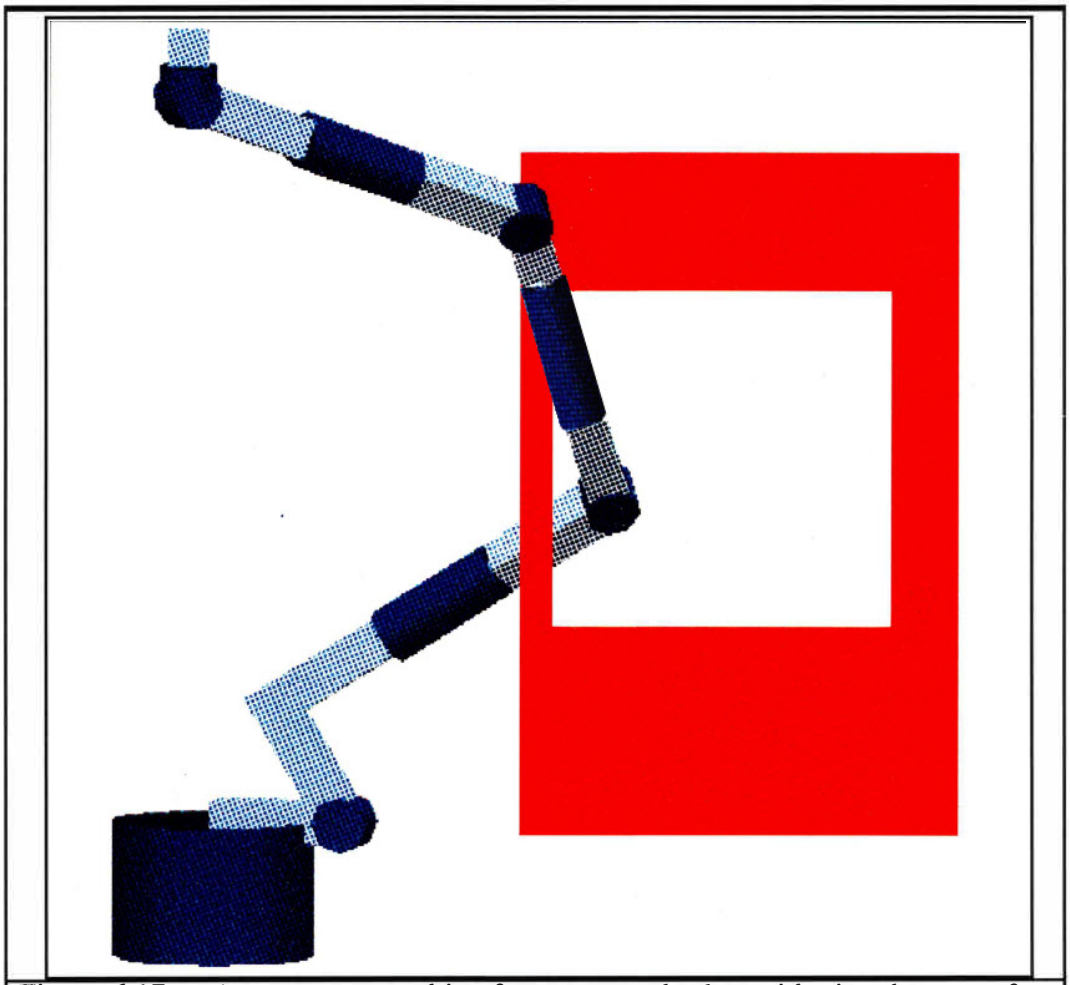

Figure 6.17 A computer graphic of a conceptual robot with nine degrees of freedom.

Figure 6.17 shows the conceptual robot with nine degrees of freedom. It is a sequence of alternating roll and pivot joints. The geometry is very much like many research robots with seven degrees of freedom, but with one extra pivot and one extra roll. Besides being highly redundant, this geometry is also complex because offsets at the pivot joints prevent any of the joint axes from intersecting.

Table 6.4 Search parameters and results for the simulation of the dexterous robot with nine degrees of freedom.

exploration pattern -	simple
decision making strategy -	opportunistic
objective function -	Cartesian error and the two-norm of joint angles
search resolution -	.001 degrees
scaling method -	search-based with forced convergence
solution speed -	10 Hertz
maximum translational error -	.000046 meters (.0018 inches)
maximum rotational error -	.00004 radians
degrees of freedom -	9

A simple exploration pattern combined with opportunistic decision making to form the direct search strategy. The simple exploration pattern does not guarantee local optimization. The objective function for the search contained the rotational and translational error as well as the two-norm of the joint displacements. The joint displacement criterion discourages the robot from approaching its joint limits while following the path. An infinite barrier constraint enforced the joint limits. The resolution for the search was .001 degrees. The

search-based scaling procedure generated the initial scaling factors. Because of the suboptimal search strategy, the scaling factors were modified during the search to force convergence. Chapter 4 discusses this procedure. Essentially, after each time the search terminates without finding a solution, the scaling factors of any unsatisfied constraints are increased by a power of ten. Only two constraints are included in the objective function. They are the magnitudes of the translational and the rotational errors. The translational constraint was satisfied if the error was less than .0000254 meters (.001 inches). This is ten times less than the resolution of a typical industrial robot. The global scaling method determined the corresponding factor for the rotational constraints as .0000318 radians.

A path described in end-effector space tested the direct search algorithm by recording the rotational and translational error at each point along the path. The end-effector path was three connected straight-line segments near the boundaries of the robot's workspace. Figure 6.18 shows a computer generated trace of the robot following the path. Figure 6.19 shows the error due to the direct search inverse kinematics algorithm. The maximum translational error is .000046 meters (.0018 inches) and the average is .000022 meters (.00087 inches). The maximum rotational error is .000040 radians and the average is .000016 radians. At this resolution, the direct search algorithm's solution speed was about ten Hertz on a Silicon Graphics Indigo R4000.

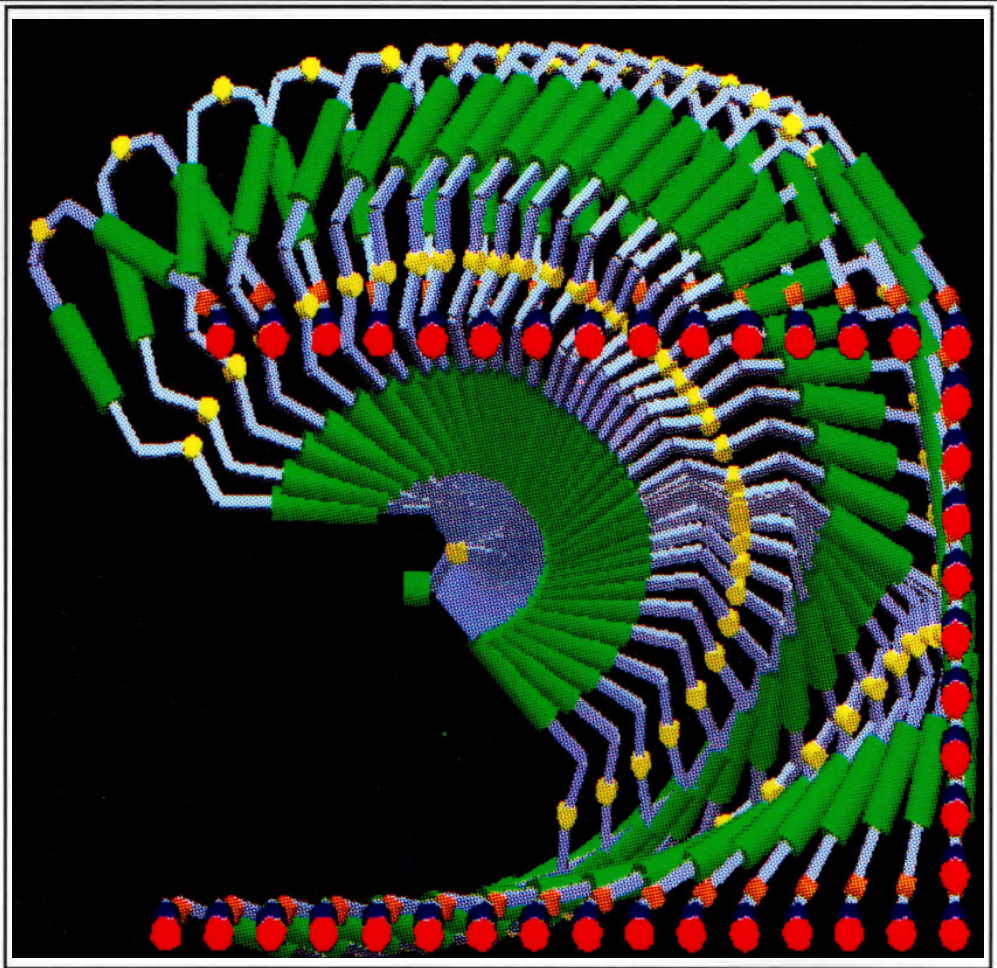

Figure 6.18 A computer generated trace of a conceptual robot with nine degrees of freedom performing a path composed of three connected straight-line segments near its workspace boundaries.

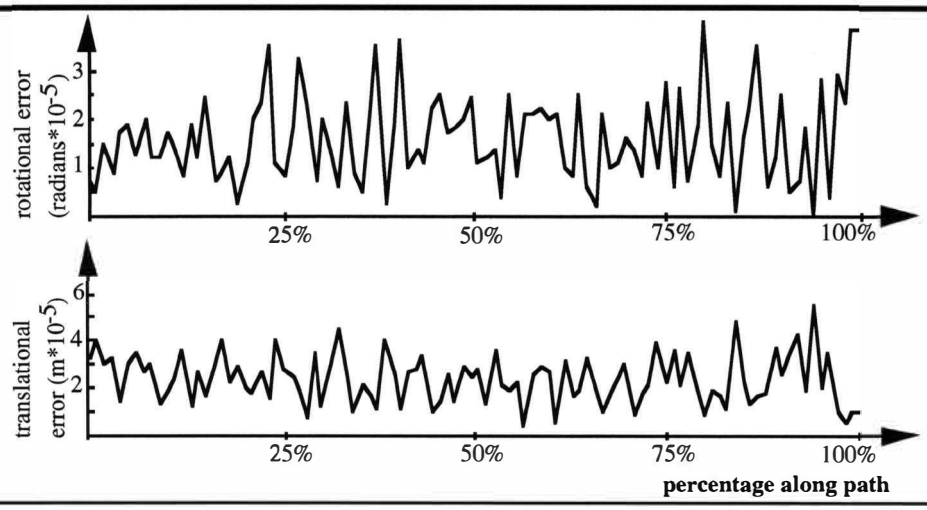

Figure 6.19 The magnitudes of the translational and rotational errors at the end-effector due to the direct search numerical algorithm.

6.8 Fault-Tolerant Robot with Twenty-One Degrees of Freedom

This final example simulates a hyper-redundant and fault-tolerant robot. The robot has twenty-one degrees of freedom (Figure 6.20). A robot with very many extra degrees of freedom is often called hyper-redundant. Hyper-redundancy enables the robot to continue operating effectively even if several joints fail. The application of direct search inverse kinematics as part of a fault-tolerant robotic system is shown to be particularly straightforward. Suddenly locking three of the joints while the robot is following a straight-line path simulates a fault. Though it is realized that suddenly locking any joints will have serious dynamic repercussions, this example only considers kinematics. The performance of the direct search algorithm during a brief interval immediately after the fault is particularly interesting.

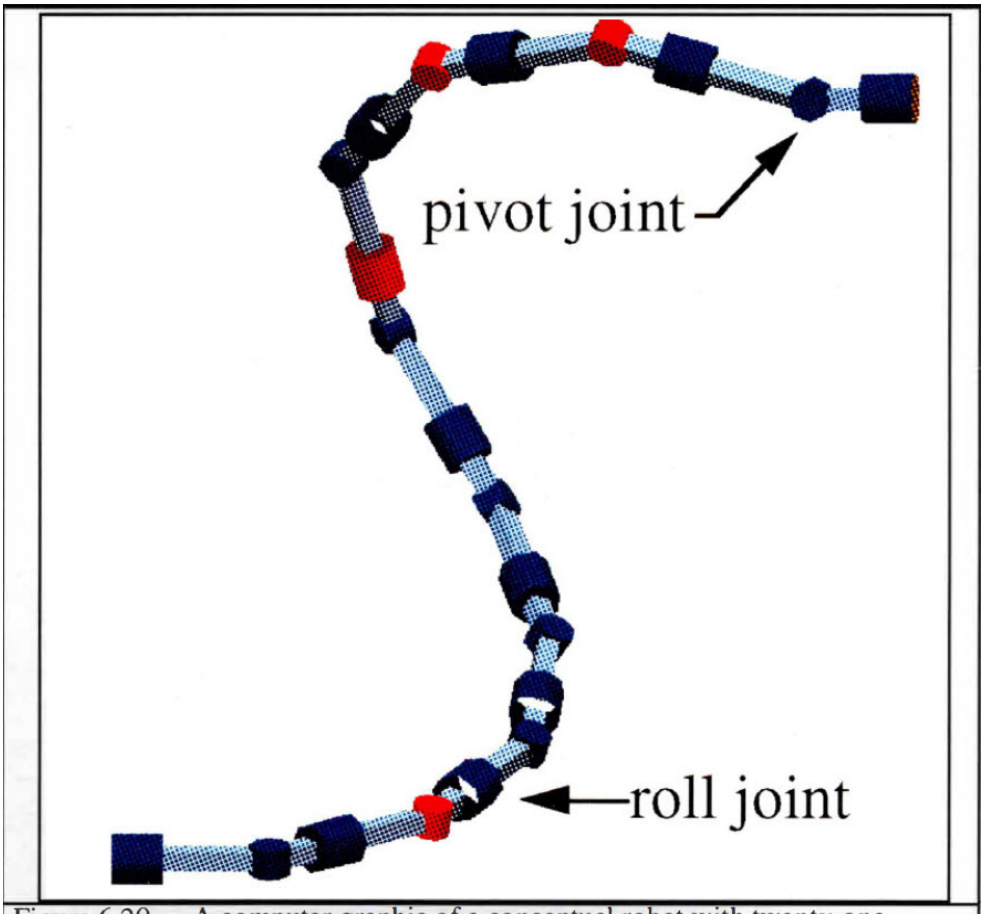

Figure 6.20 A computer graphic of a conceptual robot with twenty-one degrees of freedom.

A simple exploration pattern combined with opportunistic decision making to form the direct search inverse kinematics strategy. This combination does not guarantee a local optimum. This example used four different objective functions. They all contained terms for the Cartesian error. One was only the Cartesian error while the other three also included either the two-norm of the joint speeds, the two-norm of the joint accelerations, or both. These norms represent

very simple performance criteria; one at the velocity level and one at the acceleration level. The search-based scaling procedure generated the scaling parameters. Regardless of the objective function, the direct search algorithm's solution speed was about five Hertz.

Table 6.5 Search parameters and results from the simulation of the fault-tolerant robot with twenty-one degrees of freedom

exploration pattern -	simple
decision making strategy -	opportunistic
objective function (four cases) -	Cartesian error
	Cartesian error and two-norm of joint speeds
	Cartesian error and two-norm joint accelerations
	Cartesian error, two-norm of joint speeds, and two-norm of joint accelerations
search resolution -	.001
scaling method -	search based with forced convergence
solution speed -	5 Hertz
maximum translational error -	.000024 meters (.00095 inches)
maximum rotational error -	.00044 radians
degrees of freedom -	21 (before fault, 18 after)

Serial kinematic redundancy is one way of achieving fault tolerance. The robot in this example has enough extra joints to allow continued operation even if several joints fail. A properly chosen geometry will also allow a robot with less than this degree of redundancy, perhaps a robot with ten joints, to achieve fault-tolerance through serial kinematic redundancy. In this fault-tolerance scenario, a faulty joint is locked and prevented from turning. The faulty joint essentially

becomes part of the robot's structure. The incorporation of this fault-tolerance strategy as part of the direct search is straightforward. The failed joint is simply removed from the search space and the exploration process generates no perturbations of this joint. Thus, the decision making process is not presented with choices associated with moving the failed joint.

The robot in this example is interesting because of its very large number of joints. Application of a pseudoinverse would require the inversion of a twenty-one by twenty-one matrix; a process not likely to be performed in real-time (at least in the very near future). The robot's joints are arranged as the familiar sequence of roll and pivot, but there are no offsets at the pivot joints. This is very important because, without the offsets, every time one of the pivot joints passes through the center of its travel, the axes of the roll joints on either side are collinear. When these roll axes are collinear; the robot is in a singularity. For this robot, however, the singularities do not represent any loss of dexterity. The robot would be expected to pass through several of these singularities while following even a modestly demanding path. Avoiding these singularities would severely limit the dexterity of the robot. These singularities are not a problem for the direct search algorithm this example demonstrates. The algorithm does not explicitly address singularities at all. The decision making process simply evaluates the singular solutions in the same manner it evaluates any choices the exploration pattern generates.

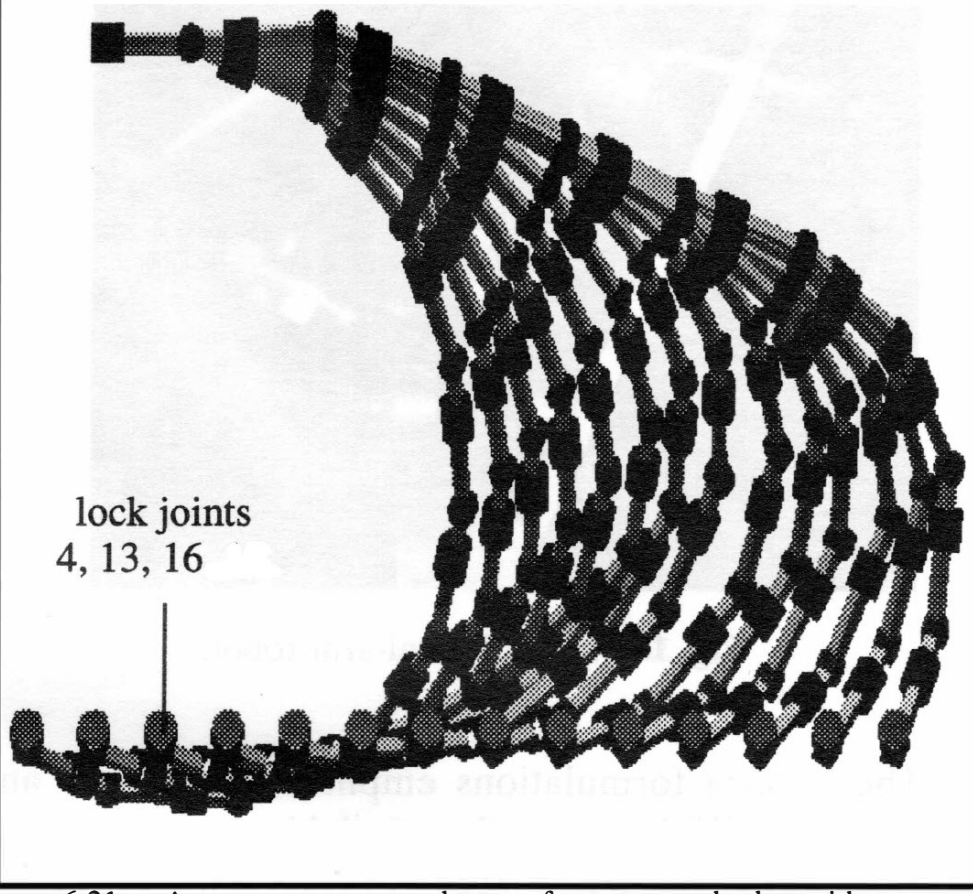

Figure 6.21 A computer generated trace of a conceptual robot with twenty-one degrees of freedom maintaining a linear path after a simulated failure of three joints. The failed joints change from blue to red when they fail.

This experiment simulated a fault by suddenly locking three separate joints. The simulated fault occurred while the robot was following a straight-line path. The Cartesian error, the two-norm of the joint speeds, and the two-norm of the change in joint speeds with respect to time, were recorded as the robot continued to follow the path after the fault. This experiment was performed four

different times, with the objective function for the direct search being changed each time as described above. In each case, the robot was in the same state before the fault. After the fault, the different objective functions cause the simulated robot to follow different joint-level paths, even though the end-effector path did not change.

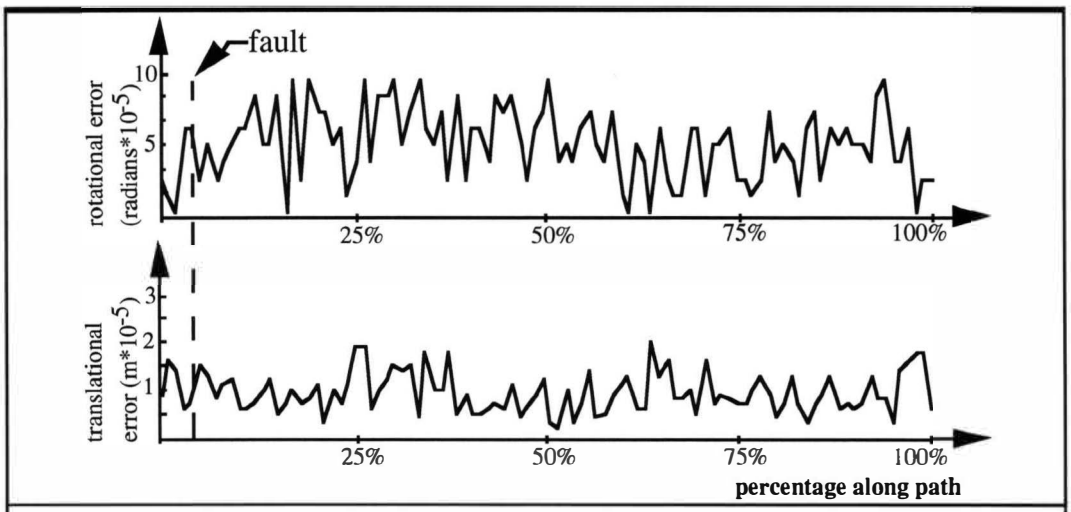

Figure 6.22 The magnitudes of the translational and rotational errors at the end-effector due to the direct search numerical algorithm.

Figure 6.22 graphs the rotational and translational errors for the path considering both speed and acceleration criteria. The fault had no effect on the ability of the direct search algorithm to properly place the robot's end-effector. For each of the four paths discussed in this example, the solution was always within the desired error bounds both before and after the fault. In some sense, solving the inverse kinematics problem for a robot with this many extra joints is less difficult than solving the problem for a robot with fewer degrees of freedom. The proliferation of solutions actually makes the solution more tractable.

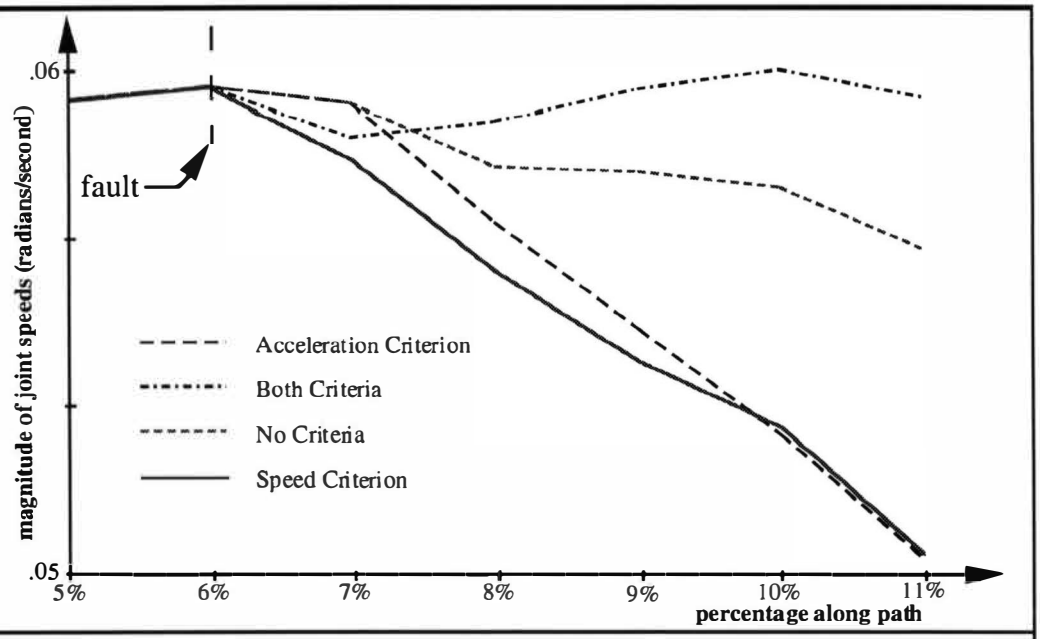

Figure 6.23 The joint speed criterion for each of the four paths in the time immediately after the fault.

Figure 6.23 shows the two-norms of the joint speeds in the time immediately after the fault for each of the four objective functions. The step size seems to be coarse because of the brief time interval being examined. During the first few steps after the fault, the objective function which only considers the two-norm of joint speeds is the best at minimizing the joint speed criterion. This is to be expected, and helps to validate that the direct search optimization scheme is working.

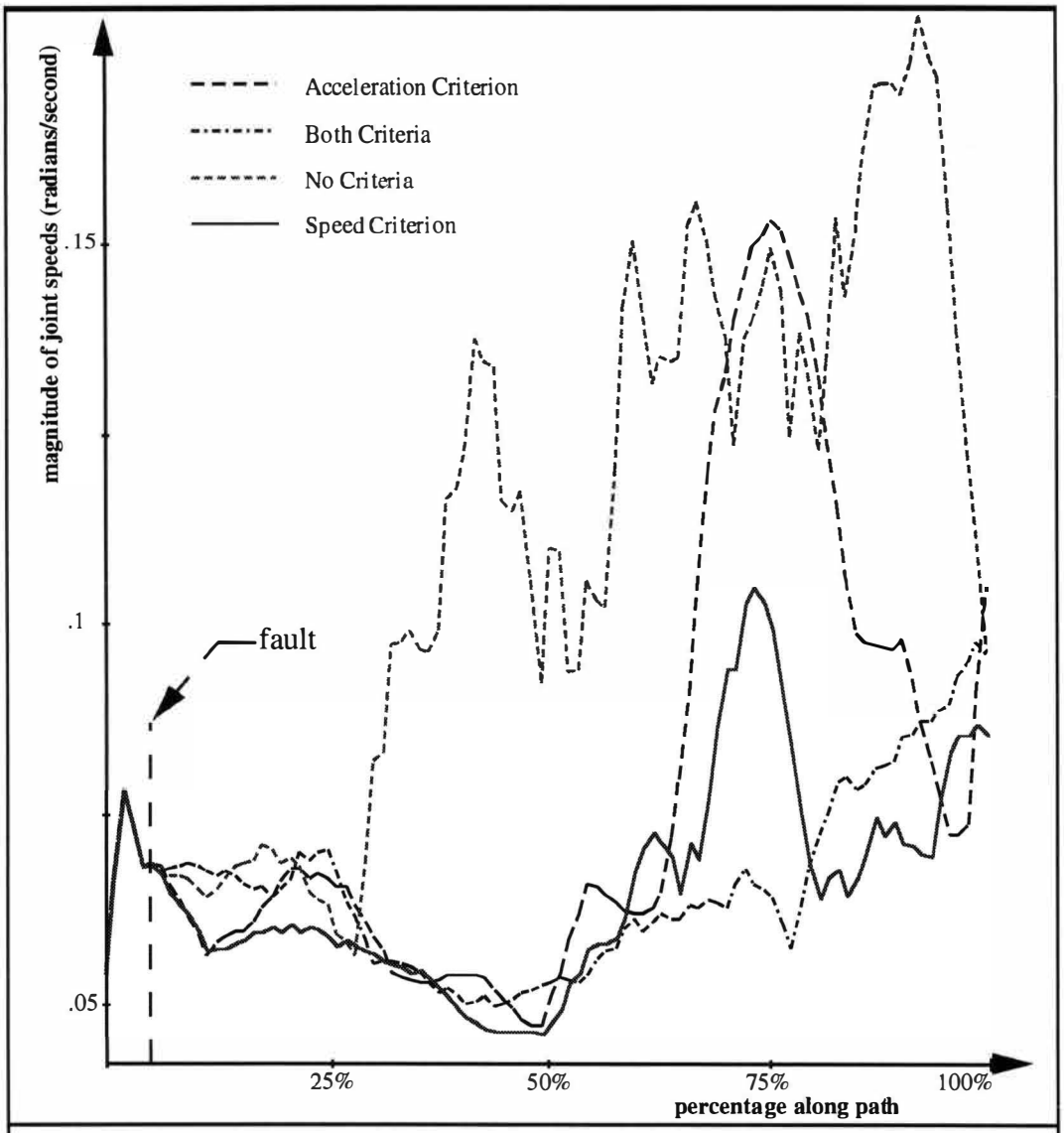

Figure 6.24 The joint speed criterion for each of the four paths.

Once the paths diverge there is no guarantee that they also won't cross. In other words, the best first step does not guarantee the best whole path. This is because of the local nature of the direct search inverse kinematics method. Figure 6.24 shows the path corresponding to the velocity criterion actually crosses the path corresponding to the mixed criterion several times.