Simulating a Robotic Arm in a Box: Redundant Kinematics, Path Planning, and Rapid Prototyping for Enclosed Spaces

Joseph T. Wunderlich
Elizabethtown College
Computer Science and Computer Engineering Programs
Elizabethtown, PA 17022
wunderjt@etown.edu

Robotic arms often perform industrial tasks requiring complex dextrous manipulation within constrained spaces. For example, automobile unibody assembly can require more than 5000 welds, with many performed within the vehicle's interior. An arm can be designed specifically for this type of task by permuting link lengths and degrees of freedom (DOF) to find a set of feasible designs. Each design can be evaluated for joint-angle displacement, dexterity, simulated speed, and consumption of available redundancy. A heuristic search increases the probability of having the needed kinematic structure. Hyperredundant designs of up to 10 DOF can be created, and searches often yield minimized-DOF designs. The path-planning technique combines pseudo-inverse velocity control with the concept of attractive poles to allow maneuvering through complex enclosures while avoiding many obstacles. This research provides a means of rapid-prototyping robotic arms for enclosed spaces and can yield many designs locally optimized for given tasks and environments.

Keywords: Robotic arms, simulation, rapid prototyping, path planning

1. Introduction

The work presented here was inspired by tours of automobile assembly plants and the desire to improve complex industrial automation tasks. The constrained workspace of an automobile unibody interior is shown in Figure 1. Industrial robots performing unibody assembly are shown in Figure 2. Most industrial arms are general purpose, and the trajectory of the tool center point (TCP) of a task is often programmed by a technician using a *teach pendant* to physically move the robot into position while in *teach mode*. The design of task-specific arms can be easily justified if tasks are repeated millions of times per year for a fixed workspace. For example, the average automobile unibody can require more than 5000 separate welds, and with thousands of automobiles assembled every year, millions of motions can be repeated per year for a single arm [1, 2]. Once a task-specific arm is designed, the physical implementation of the design can be programmed in teach mode or by using any of a variety of other path-planning techniques.

The design of a robotic arm to perform tasks in an enclosed space can be achieved by designing the arm for the most difficult task in the most constricted arm configuration. Figure 3 shows the modeling of an automobile interior and an initial guess of an arm design, and Figure 4 shows this design performing a task on the ceiling of the interior. If the workspace is assumed to be approximately a cube, a 2-D square *slice* of the cube can be used to approximate the most constricted work area. Once an arm design is found for this environment, it can be permuted to yield a set of feasible designs. Future plans for this research include extending this methodology into a complete 3-D workspace (funding for this is presently being sought).

1.1 Robotic Arm Model

A 2 degree-of-freedom (DOF), 2-D (planar) robotic arm is shown in Figure 5.

In general, for an n-DOF planar arm, the end-effector position is given by

$$\mathbf{x}_e = \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{n} L_i * \left(\cos \left(\sum_{j=1}^{i} \theta_j \right) \right) \\ \sum_{i=1}^{n} L_i * \left(\sin \left(\sum_{j=1}^{i} \theta_j \right) \right) \end{bmatrix}, \quad (1)$$

and the end-effector velocity is
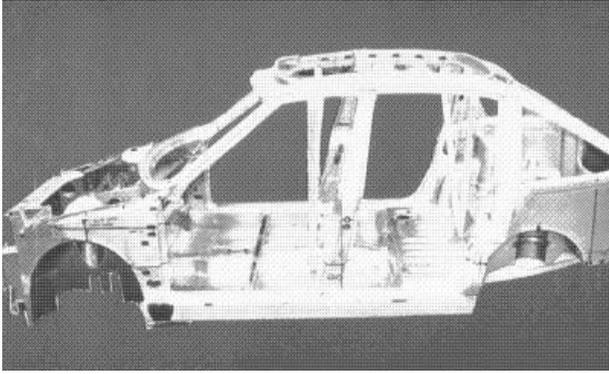
Figure 1. An automobile unibody [3]
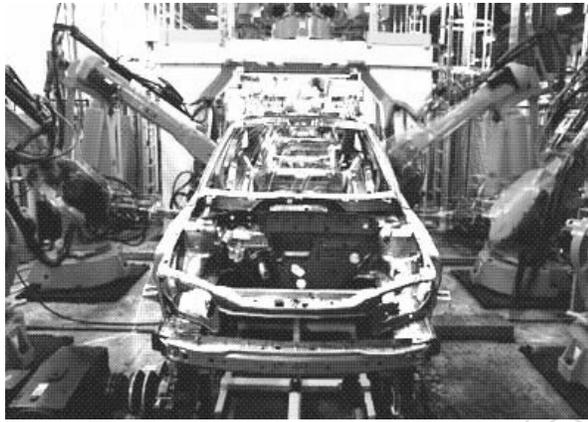
Figure 2. Robots assembling unibody [3]

$$\dot{\mathbf{x}}_e = \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} \partial x/\partial\theta_1 & \partial x/\partial\theta_2 \cdots & \partial x/\partial\theta_n \\ \partial y/\partial\theta_1 & \partial y/\partial\theta_2 \cdots & \partial y/\partial\theta_n \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \vdots \\ \dot{\theta}_n \end{bmatrix},$$
(2)

or simply

$$\dot{\mathbf{x}}_e = \mathbf{J}_e \dot{\theta},$$
(3)

where $\mathbf{J}_e$ is the Jacobian matrix. Assuming $n > m$, where m is the dimension of the workspace, we have a redundant manipulator, and the general form of the least squares approximate solution to this underdetermined set of linear equations is

$$\dot{\theta} = \mathbf{J}_e^{\#} \dot{\mathbf{x}}_e + (\mathbf{I} - \mathbf{J}_e^{\#}\mathbf{J}_e)\dot{\Psi},$$
(4)

where $\mathbf{J}_e^{\#}$ is the pseudo-inverse, $\mathbf{I}$ is an identity matrix, $\dot{\Psi}$ is an arbitrary joint-velocity vector that can be used for a variety of optimization and path-planning tasks, and $(\mathbf{I} - \mathbf{J}_e^{\#}\mathbf{J}_e)\dot{\Psi}$ is the projection of $\dot{\Psi}$ onto the null space of $\mathbf{J}_e$ [4]. The pseudo-inverse $\mathbf{J}_e^{\#}$ is

$$\mathbf{J}_e^{\#} = \mathbf{J}_e^T (\mathbf{J}_e\mathbf{J}_e^T)^{-1}$$
(5)

here since $(m < n)$, and $\mathbf{J}_e$ is assumed to be of rank m. Equation (4) represents the least squares solution that minimizes the error norm,

$$\min \left\| \dot{\mathbf{x}}_e - \mathbf{J}_e\dot{\theta} \right\|,$$
(6)

and focuses on the *exactness* of the solution [5]. The first term of (4) represents the minimum norm solution among all solutions provided by (4) by also satisfying

$$\min \left\| \dot{\theta} \right\|,$$
(7)

which relates to the *feasibility* of implementing a solution since excessively large joint-angle velocities are not realizable [5].

In Hanafusa, Yoshikawa, and Nakamura [6], $\dot{\Psi}$ in (2) is used to avoid obstacles by commanding joint-angle velocities to drive an arm's configuration toward a predetermined configuration previously proven to avoid a given obstacle. In Nakamura, Hanafusa, and Yoshikawa [7], the null space defined by the second term in (2) is used to define secondary and tertiary priority tasks. In Maciejewski and Klein [8], this approach is modified to command a Cartesian velocity to repel the point on an arm closest to an obstacle directly away from the obstacle while attempting to maintain a *fixed* end-effector trajectory. However, in other studies [9-13], it is noted that there are always stability issues to be addressed when using pseudo-inverse velocity control.

In Khatib [14], *local attractors* pull the end effector toward an attractive pole while joints are repelled from obstacles modeled as repulsive geometries. The end-effector trajectory is not fixed. One limitation of this technique is that a trajectory can get stuck in potential wells (i.e., a local minima) [15]. This can be addressed with a randomized path planner that takes "random walks" to escape local minima [16, 17]. A related issue in path planning is whether a technique is "probabilistically complete," in which a path will always be found if one exists [16-18].

In Chirikjian and Burdick [15], hyperredundant arms maneuver into enclosures by navigating through tunnels defined through the workspace. The arm is modeled as a continuous curve *threaded* through the tunnels and discretized into short equal-length link lengths. This approach does not, however, consider unequal link lengths and does not make use of much of the space within the enclosure. In other studies [3, 19, 20], many criteria for designing robotic arms are discussed, including mechanical constructability and kinematic simplicity. In Shiller and Sundar [21], the link lengths of a two-link arm are optimized for maximum
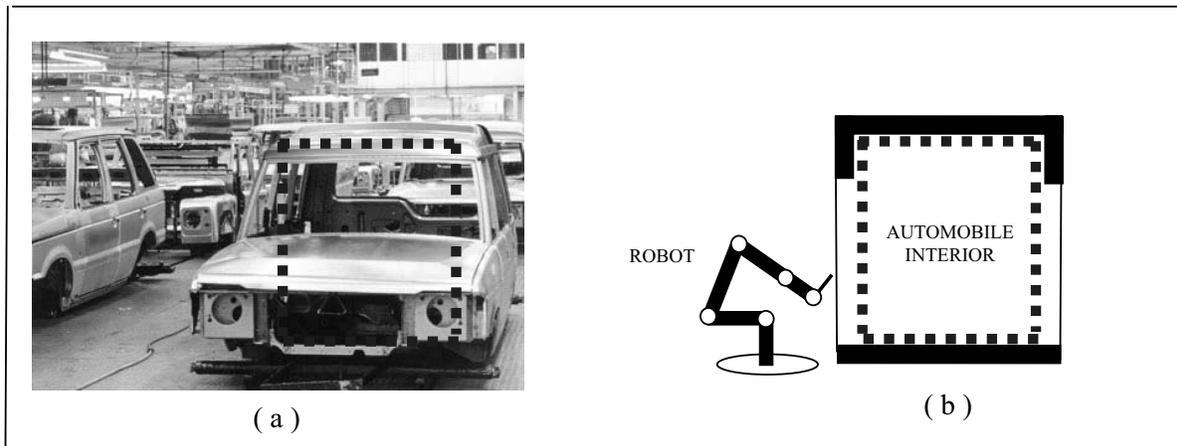
Figure 3. Robot workspace inside automobile unibody
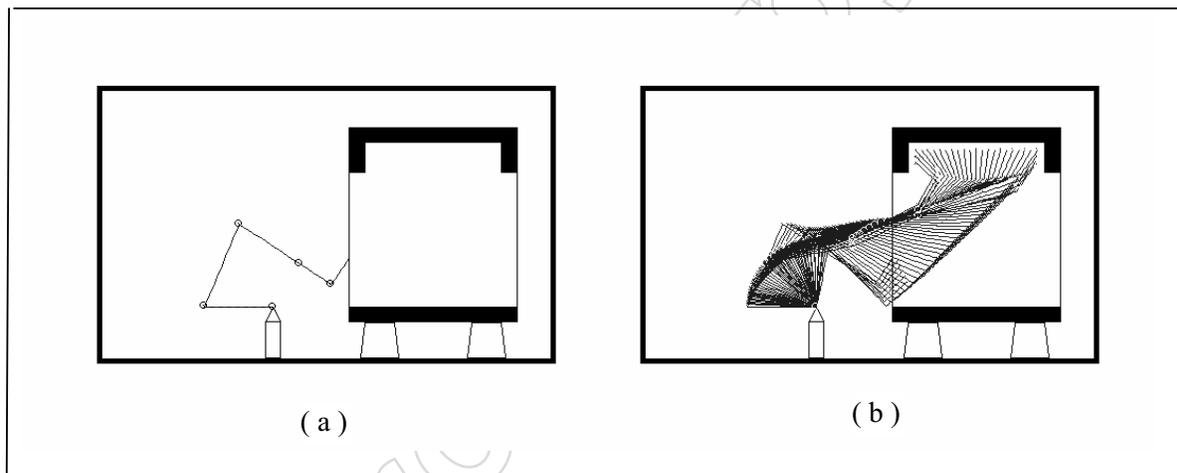
Figure 4. Robot performing task on ceiling of automobile interior

acceleration, and in Mayorga, Ressa, and Wong [22], link lengths of several commercial arms are evaluated using a dexterity measure (the condition number of the Jacobian). In Paredis and Khosia [23], link lengths are sized for specific tasks using random search algorithms.

Many researchers have explored using fixed–link length, fixed-DOF robotic arms in constrained spaces [6-8, 14, 15, 24], and research has been conducted on optimizing link lengths of nonredundant robotic arms in unconstrained spaces [21-23]. The proposed method finds link lengths and DOF for redundant arms in constrained workspaces, and a path-planning technique is developed to test candidate designs. This technique combines variations of the pseudo-inverse velocity control technique in Maciejew-

ski and Klein [8] with the attractive-pole concept of the potential-fields technique in Khatib [14], and although the method is not "probabilistically complete," it is shown to be quite robust while testing trajectories and leads to many resulting arm designs.

Many performance measures have been proposed for evaluating arm trajectories. These include kinetic energy, joint torque [20], and many dexterity measures [25-27]. The performance measures considered here are joint-angle displacement, dexterity, simulated speed, and a new measure: the consumption of available redundancy (COAR). These measures are used to compare the performance of robotic arm permutations over test trajectories in a given enclosure.
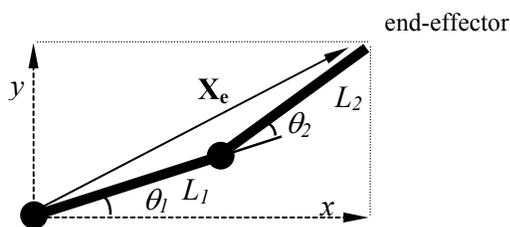
Figure 5. A 2 degree-of-freedom robotic arm

2. Path Planning and Obstacle Avoidance

The pseudo-inverse velocity-control part of the proposed path-planning technique is a variation of that used in Maciejewski and Klein [8], where $\dot{\Psi}$ in (4) is used to repel a point X_o on an arm away from obstacles by commanding a Cartesian velocity:

$$\dot{\mathbf{x}}_o = \mathbf{J}_o \dot{\theta}, \tag{8}$$

where (o) designates the point on the arm closest to an obstacle (*obstacle-avoidance point*). In Maciejewski and Klein [8], (4) is substituted into (8) to yield

$$\dot{\mathbf{x}}_o = \mathbf{J}_o \mathbf{J}_e^\# \dot{\mathbf{x}}_e + \mathbf{J}_o (\mathbf{I} - \mathbf{J}_e^\# \mathbf{J}_e) \dot{\Psi}, \tag{9}$$

where $\mathbf{J}_o \mathbf{J}_e^\# \dot{\mathbf{x}}_e$ is the Cartesian motion at the obstacle-avoidance point to satisfy the end-effector velocity constraint. The second term of (9) represents the mapping of the $(\mathbf{I} - \mathbf{J}_e^\# \mathbf{J}_e) \dot{\Psi}$ null-space joint-velocity vector to a Cartesian vector at the obstacle-avoidance point. The vector $\dot{\Psi}$ is found by rewriting (9) as

$$\dot{\mathbf{x}}_o - \mathbf{J}_o \mathbf{J}_e^\# \dot{\mathbf{x}}_e = \mathbf{J}_o (\mathbf{I} - \mathbf{J}_e^\# \mathbf{J}_e) \dot{\Psi}, \tag{10}$$

where $\dot{\mathbf{x}}_o - \mathbf{J}_o \mathbf{J}_e^\# \dot{\mathbf{x}}_e$ is the desired obstacle-avoidance point Cartesian velocity:

$$\dot{\Lambda} = \dot{\mathbf{x}}_o - \mathbf{J}_o \mathbf{J}_e^\# \dot{\mathbf{x}}_e, \tag{11}$$

and $\mathbf{J}_o (\mathbf{I} - \mathbf{J}_e^\# \mathbf{J}_e)$ is the transformation of the orthogonal projection operator from the end effector to the obstacle-avoidance point:

$$\Gamma = \mathbf{J}_o (\mathbf{I} - \mathbf{J}_e^\# \mathbf{J}_e). \tag{12}$$

Using (11) and (12), we can rewrite (10) as

$$\dot{\Lambda} = \Gamma \dot{\Psi}, \tag{13}$$

where the general form of the least squares $\dot{\Psi}$ solution is

$$\dot{\Psi} = \Gamma^\# \dot{\Lambda} + (\mathbf{I} - \Gamma^\# \Gamma) \dot{\beta}, \tag{14}$$

where $\mathbf{I}$ is an identity matrix, $\dot{\beta}$ is an arbitrary vector, and $(\mathbf{I} - \Gamma^\# \Gamma) \dot{\beta}$ is the projection of $\dot{\beta}$ into the null space of Γ.

This equation represents the least squares solution that minimizes the error norm:

$$\min \left\| \dot{\Lambda} - \Gamma \dot{\Psi} \right\|. \tag{15}$$

The first term of (14) represents the minimum-norm solution among all of the solutions provided by (14) by also satisfying

$$\min \left\| \dot{\Psi} \right\|, \tag{16}$$

which has the effect of increasing the minimum obstacle distance [8].

Substituting (14) into (4) yields

$$\dot{\theta} = \mathbf{J}_e^\# \dot{\mathbf{x}}_e + (\mathbf{I} - \mathbf{J}_e^\# \mathbf{J}_e) \Gamma^\# \dot{\Lambda} \\ + (\mathbf{I} - \mathbf{J}_e^\# \mathbf{J}_e) \left[(\mathbf{I} - \Gamma^\# \Gamma) \dot{\beta} \right]. \tag{17}$$

In Nakamura, Hanafusa, and Yoshikawa [7], the third term of (17) is used for a tertiary-priority task if *enough* available redundancy remains after the higher priority tasks are satisfied. In Maciejewski and Klein [8], this term is dropped, and (17) is expanded to yield

$$\dot{\theta} = \mathbf{J}_e^\# \dot{\mathbf{x}}_e + (\mathbf{I} - \mathbf{J}_e^\# \mathbf{J}_e) \\ \left[\mathbf{J}_o (\mathbf{I} - \mathbf{J}_e^\# \mathbf{J}_e) \right]^\# (\dot{\mathbf{x}}_o - \mathbf{J}_o \mathbf{J}_e^\# \dot{\mathbf{x}}_e). \tag{18}$$

In Maciejewski and Klein [8], it is shown that the second term of (18) can be reduced to $[\mathbf{J}_o (\mathbf{I} - \mathbf{J}_e^\# \mathbf{J}_e)]^\# (\dot{\mathbf{x}}_o - \mathbf{J}_o \mathbf{J}_e^\# \dot{\mathbf{x}}_e)$ since the projection operator $(\mathbf{I} - \mathbf{J}_e^\# \mathbf{J}_e)$ is both hermetian and idempotent, and therefore joint-angle velocities are governed by

$$\dot{\theta} = \mathbf{J}_e^\# \dot{\mathbf{x}}_e + \left[\mathbf{J}_o (\mathbf{I} - \mathbf{J}_e^\# \mathbf{J}_e) \right]^\# \\ (\dot{\mathbf{x}}_o - \mathbf{J}_o \mathbf{J}_e^\# \dot{\mathbf{x}}_e), \tag{19}$$

by specifying a desired end-effector velocity $\dot{\mathbf{x}}_e$ and a desired obstacle-avoidance point velocity $\dot{\Lambda} = \dot{\mathbf{x}}_o - \mathbf{J}_o \mathbf{J}_e^\# \dot{\mathbf{x}}_e$ through the selection of $\dot{\mathbf{x}}_o$.

The proposed technique for avoiding many obstacles within a complex enclosure is a variation of (19) and is given by

$$\dot{\theta} = \mathbf{J}_e^\# \dot{\mathbf{x}}_e + \sum_{i=1}^{N} \left[\mathbf{J}_{o_i} (\mathbf{I} - \mathbf{J}_e^\# \mathbf{J}_e) \right]^\# \dot{\Lambda} \right], \tag{20}$$

where $\dot{\Lambda}$ is commanded directly, and where N is the number of obstacle-avoidance points, one fixed at each of the arm's joints (with the exception of the first two since they remain outside the enclosure). Up to eight obstacle-avoidance points have been simulated (i.e., on a 10-DOF arm). Here, unlike in (19), there is no need to locate these points on the arm since they are fixed; however mid-link collisions with obstacles become possible and are avoided

by setting minimum allowable distances from obstacles. In Maciejewski and Klein [8], avoiding multiple obstacles as multiple secondary-priority tasks is proposed instead of using the third term in (17) for tertiary tasks; the second term of (19) is split and scaled in proportion to obstacle proximity. Although this technique was experimented with here, it was found that (20), combined with the following measures, provided greater success. The following measures make the control scheme feasible:

1. The end effector is guided by *attractive poles* (similar to those in Khatib [14]) such that the trajectory can vary when available redundancy becomes diminished.

2. Available redundancy is conserved by repelling obstacle-avoidance points in a direction related to the end-effector TCP trajectory.

3. Obstacle-avoidance points are repelled only within a designated area *close* to the enclosure's surface (termed a *repelling field*). This allows the arm to maneuver relatively unconstrained throughout a significant part of the enclosure.

4. Obstacle-avoidance points are repelled with a velocity inversely related to obstacle proximity. This provides smooth transitions for obstacle-avoidance points into and out of the repelling fields and often results in most repelling occurring at the outer edge of a repelling field (i.e., away from the highest magnitude repelling at the enclosure surface), as shown in Figure 6. This significantly conserves the consumption of the available redundancy (i.e., null space) over a trajectory and typically results in the accumulative effect of repelling velocity magnitudes being much less than the commanded end-effector TCP velocity (i.e., the overall commanded joint velocity vector is "feasible").

It is important to note that regardless of the use of null space in (17), (19), or (20), excessively large repelling velocitiy magnitudes are never realizable once the available redundancy has been consumed.

2.1 Consumption of Available Redundancy

To help evaluate robotic arm trajectories, a new measure is developed to approximate the COAR over each trajectory. Although the projection operator $(\mathbf{I} - \mathbf{J}_e^{\#} \mathbf{J}_e)$ in (4) maps any specified n-dimensional joint-velocity vector $\dot{\boldsymbol{\Psi}}$ to a vector orthogonal to the m-dimensional Cartesian end-effector velocity manifold, the use of (17), (19), or (20) cannot guarantee a fixed end-effector trajectory if the available redundancy is exceeded. This is illustrated in Figure 7, in which one obstacle-avoidance point on a 5-DOF arm is repelled at different angles, while the end effector is directed toward a goal and stops within a fixed distance of the goal
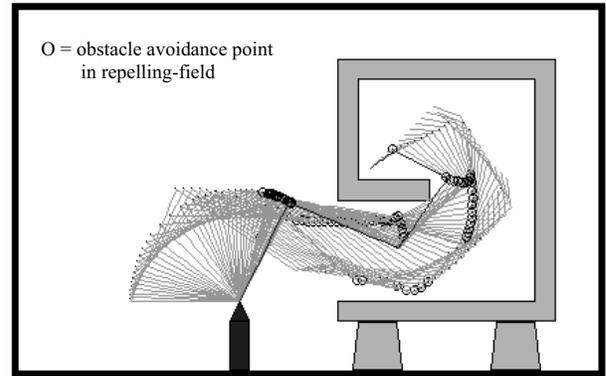
Figure 6. Obstacle-avoidance point activity in repelling fields

(to prevent overshoot or oscillation around the goal); the obstacle-avoidance point repelling velocity varies linearly over the *repelling field* from zero at the left outer boundary to 50% of the end-effector velocity at the enclosure surface. To evaluate the effect of obstacle avoidance, the following measure has been developed:

$$
\begin{aligned}
COAR &= \left[\frac{\left\| (\mathbf{I} - \mathbf{J}_e^{\#} \mathbf{J}_e) \dot{\boldsymbol{\Psi}} \right\|}{\left\| \mathbf{J}_e^{\#} \dot{\mathbf{x}}_e \right\|} \right] \\
&= \left[\frac{\left\| \left[\mathbf{J}_o (\mathbf{I} - \mathbf{J}_e^{\#} \mathbf{J}_e) \right]^{\#} (\dot{\mathbf{x}}_o - \mathbf{J}_o \mathbf{J}_e^{\#} \dot{\mathbf{x}}_e) \right\|}{\left\| \mathbf{J}_e^{\#} \dot{\mathbf{x}}_e \right\|} \right].
\end{aligned}
\tag{21}
$$

Termed the *COAR*, this measure gives an indication of how the available redundancy is used over a trajectory. Dividing by $\left\| \mathbf{J}_e^{\#} \dot{\mathbf{x}}_e \right\|$ in (21) normalizes the measure. The COAR varies significantly over a trajectory when the $d\theta_i$s at each simulation step vary significantly due to obstacle avoidance. In contrast, $d\theta_i$s for a minimum-norm solution (i.e., no obstacle avoidance) vary the least and therefore result in the smallest Euclidean norm and zero COAR. In Figure 7, 0-degree and 45-degree repelling do not significantly effect the end-effector trajectory, and 0-degree repelling consumes the least available redundancy. Both trajectories reach the goal in 20 simulation steps (the same number as in the "no-repelling" minimum-norm case). Repelling at 90 degrees, however, consumes enough available redundancy that the end-effector trajectory is adversely affected, and the goal is not reached in 20 simulation steps (see Fig. 7d). For multiple obstacle-avoidance points, the consumption of available redundancy is
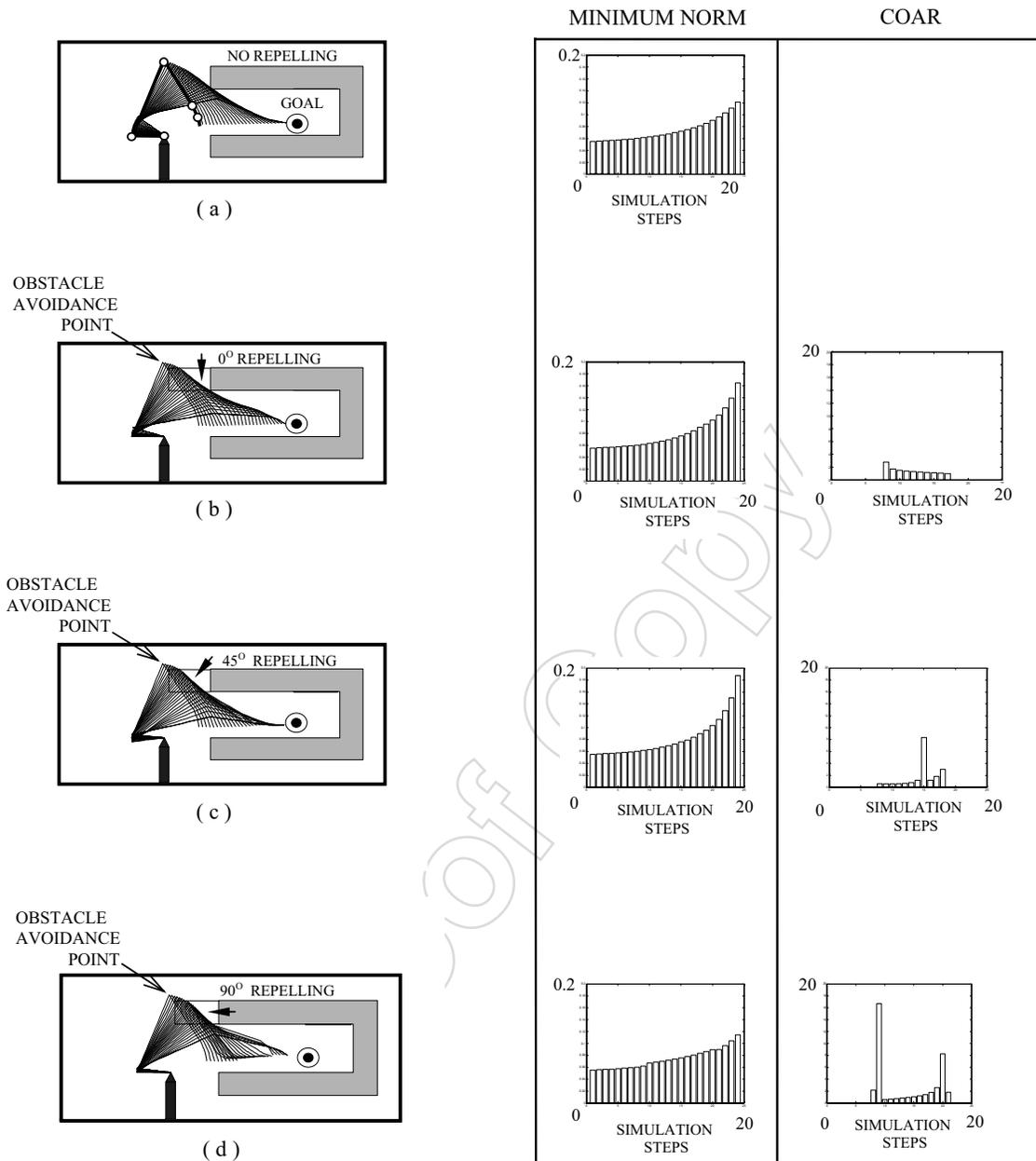
Figure 7. Comparison of obstacle-avoidance point repelling angles on a 5 degrees-of-freedom arm

$$COAR = \left[\frac{\left\| (\mathbf{I} - \mathbf{J}_e^{\#} \mathbf{J}_e) \dot{\Psi} \right\|}{\left\| \mathbf{J}_e^{\#} \dot{\mathbf{x}}_e \right\|} \right]$$

$$= \left[\frac{\left\| \sum_{i=1}^{N} \left[\mathbf{J}_{o_i} (\mathbf{I} - \mathbf{J}_e^{\#} \mathbf{J}_e) \right]^{\#} (\dot{\mathbf{x}}_{o_i} - \mathbf{J}_{o_i} \mathbf{J}_e^{\#} \dot{\mathbf{x}}_e) \right\|}{\left\| \mathbf{J}_e^{\#} \dot{\mathbf{x}}_e \right\|} \right],$$

$$(22)$$

where N is the number of obstacle avoidance points.

Although an excess of DOF can sometimes allow more maneuverability in the free space between repelling fields, observation of thousands of test trajectories has not revealed a direct relationship between COAR and DOF, and often the effect of DOF on COAR is secondary to the combination of link lengths, which allow obstacle avoidance points to stay out of the repelling fields. COAR seems to be more related to the overall geometry of the enclosure

and robotic arm than to DOF alone. Future research could include deriving a relationship between COAR and DOF and also minimizing COAR directly such that a trajectory's success could be guaranteed in advance.

3. Modeling the Constrained Workspace

Each constrained workspace is contained within an enclosure that can be constructed by assembling the simulation primitives shown in Figure 8. Figure 9 shows an example enclosure. The end effector is drawn toward *attractive poles* in each primitive, unless a goal or a *fixed* trajectory is specified. Approximate fixed-trajectory tasks are performed by disabling attractive poles and substituting closely spaced *subgoals* along the desired path. A goal or subgoal is considered reached when the end effector is within a fixed radius.

All obstacle-avoidance points are repelled using equation (20) with the following velocity:

$$\dot{\Lambda}_{o_i} = K_j(u_j), \tag{23}$$

where u_j is a unit vector defining the direction of repelling (*repelling angle*) within repelling-field j, and K_j is the repelling-velocity magnitude:

$$K_j = V_j V_e \left[1 - \left(\frac{d_j - d_{ABORT}}{t_j} \right)^E \right], \tag{24}$$

where V_j scales K_j to help allow the end-effector velocity V_e to be "feasible" and to make equation (20) achievable, d_j is the distance of an obstacle-avoidance point from the enclosure, d_{ABORT} is the distance from the enclosure to be considered a crash, and t_j is the distance that repelling-field j extends from the enclosure. An example of repelling velocity magnitudes is shown in Figure 8.

The V_js for a given enclosure and initial arm design are found using repeated trial trajectories, and the u_js are defined in an estimated direction to minimize COAR. For simplicity, only u_js of ±0, ±45, or ±90 degrees are used.

Figure 10 shows an example 5-DOF robotic arm and enclosure. The link lengths are 100, 120, 85, 60, and 30 cm. Figure 10a shows this arm reaching a goal, and Figure 10b shows the COAR over this trajectory. Note that the COAR is smallest when the arm is pushed into the free space between repelling fields, and it is largest when the arm is bent backwards to reach the goal near the end of the trajectory.

3.1 Singularities and Local Minima

Two singularity conditions to consider are position singularities and reduced-rank Jacobian singularities [9, 10, 13, 28-31]. A position singularity will occur if a planned trajectory is out of the reach of a fully extended arm. This type of singularity is avoided here by simply requiring that all arm designs have a total length greater than that needed

to reach the furthest point within the 3-D workspace of the given enclosure. Several techniques have been considered for dealing with reduced-rank Jacobian singularities, including the following:

1. singularity management techniques such as "damped least squares" or a "weighting matrix" to allow joint configurations to pass near or through singularities without excessively large velocities or divide-by-zero conditions [10, 31],

2. treating singular configurations as obstacles to be avoided using available redundancy [28], and

3. measuring "manipulability" as an indication of how close a trajectory comes to becoming singular and then comparing all candidate designs that have been proven feasible of maneuvering within the enclosure.

The third approach is the one implemented here. This was chosen for the following reasons:

- Treating singularities as obstacles would likely push the arm into the repelling fields from the free space in between since this is where some singularities occur.

- A 2-DOF slice of a 3-DOF workspace is being used in the design process, and designing an arm that is singularity free would first require exploring all trajectories over the entire 3-DOF workspace.

- Allowing many trajectories with reduced-rank Jacobians to fail (e.g., due to excessive velocities or dividing by zero) is a crude form of natural selection in the heuristic search (i.e., permuting only "best-fit" designs). Future research will likely consider all possible feasible designs, problematic ones included.

This same rationale is used for local minima where trajectories get caught in repelling fields; they are usually considered simply a poor "geometric fit" and are allowed to fail. An exception to this is when a trajectory is only temporarily diverted into local minima that can be easily escaped without backtracking. Figure 10 is an example of this, in which local minima are encountered when the TCP moves along the ceiling of the enclosure and when the TCP moves along the ledge; the TCP temporarily moves away from the enclosure walls for both cases.

4. Search for Feasible Designs

Once the enclosure is defined, a search for feasible arm designs is made. First, an initial guess is made of arm kinematics, as shown in Figure 12a. The link lengths and DOF are selected to allow the arm to reach the furthest point within the enclosure while maintaining some length for maneuverability and reaching the furthest point in the 3-D space. An initial configuration is then selected, as shown in Figure 12b.

	TUNNEL	LEFT ELBOW	RIGHT ELBOW	TERMINATOR
ATTRACTIVE POLE (⊚) **REPELLING ANGLES** (u_j) ↗				
REPELLING FIELD WIDTH (t_j)	OUTER-BANK: 30% OF ENCLOSURE WIDTH INNER-BANK: 40% OF ENCLOSURE WIDTH	OUTER-BANK: 20% OF ENCLOSURE WIDTH INNER-BANK: 40% OF ENCLOSURE WIDTH		30% OF ENCLOSURE WIDTH
(E)	OUTER-BANK: E = 0.1 INNER-BANK: E = 1.0	OUTER-BANK: E = 0.1 INNER-BANK: E = 0.0		E = 0.1

EXAMPLE K_j's for $t_j = 20$, $d_{ABORT} = 0$, $V_j = V_e = 1$

$$K_j = V_j V_e \left[1 - \left(\frac{d_j - d_{ABORT}}{t_j} \right)^E \right]$$

Note: If a goal or *fixed*-trajectory task is specified within primitive, the attractive pole is disabled and repelling-angles are set to 90 degrees.

Figure 8. Simulation primitives for constructing enclosures
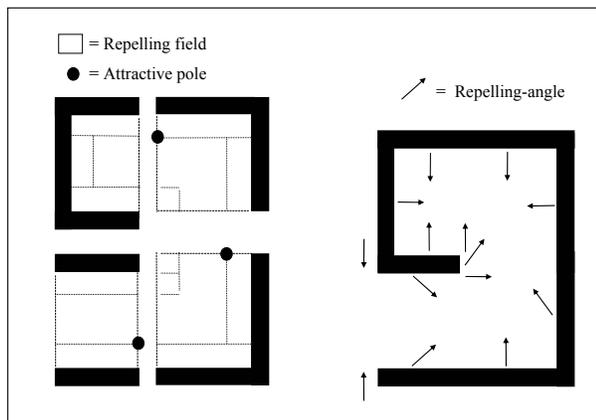
Figure 9. Example enclosure constructed from simulation primitives

Figure 11. Trajectory escaping local minima

(a)

AVERAGE COAR = 0.9532

(b)

Figure 10. Consumption of available redundancy (COAR) over a test trajectory

All designs are tested for maneuverability within the specified target workspace shown in Figure 12b. This is accomplished by commanding the end-effector TCP to follow a path along the enclosure's walls after it reaches the target workspace boundary. Figure 12c shows a *successful design* performing the test trajectory. Repelling-velocity magnitudes for an initial design are found using repeated trial trajectories and are not changed during a search. Through observation of thousands of simulation runs, there is a noticeable correlation between initial kinematic orientation (i.e., joint angles) and the required repelling field strengths. However, once an initial-orientation/field-strengths pair is established, the methodology proves to be quite robust, with often thousands of new designs generated from one initial design. Also, when this research was first undertaken, many initial configurations and the robustness of the overall methodology were tested [13]. Figure 13a,b shows two very different initial orientations with identical repelling-field settings. Both succeeded at performing within the environment. In general, almost any initial orientation other than one fully contracted (such as in Fig. 14) can succeed with the proper repelling-field settings, and even the configuration in Figure 14 managed to reach the goal (but not with a desirable trajectory). Once an initial design is found, a heuristic search is used to generate new designs by changing link lengths and testing each new design in the target workspace. To improve search efficiency, link lengths are permuted such that total robotic arm length remains constant and such that each new design does not *significantly* differ from its parent (i.e., by only changing two link lengths each permutation). The first iteration of the search yields 20 permutations for a 5-DOF design. Successful designs and *minor-hit* designs produce 20 more permutations each; "minor-hit" designs are those that the end effector can follow the walls of the target workspace without any obstacle-avoidance point hitting the enclosure, but the arm hits the enclosure mid-link (i.e., between obstacle-avoidance points). Although not considered
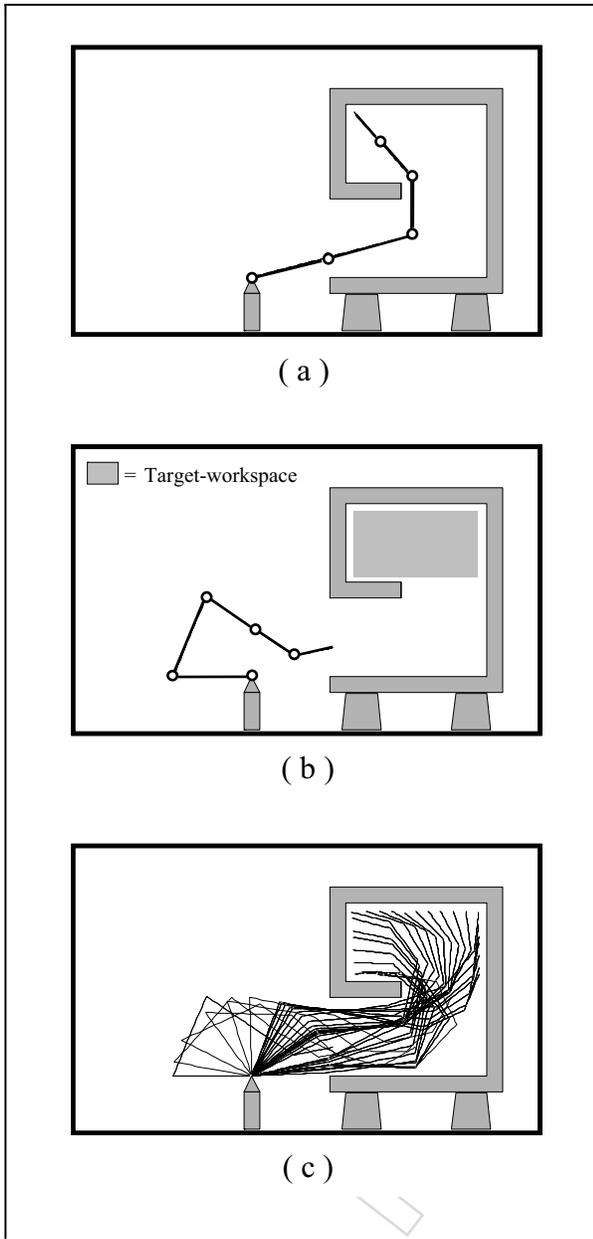
Figure 12. Search initialization: (a) initial guess of link lengths and degrees of freedom, (b) initial configuration and specified target workspace, and (c) manipulator reaching all of target workspace

successful, these permutations often lead to successful designs. Any new permutations identical to a previously generated design are eliminated (before testing). The search is continued until no new permutations are successful.

If no initial design is easily found, an extra link is added. In this manner, a hyperredundant robotic arm may result.

An excess of links can be determined by setting the minimum link length to zero; therefore, a search may result in the elimination of links, minimizing the required DOF.

One variation of the search strategy is to permute links such that only one link at a time is targeted for reduction; this often quickly results in minimal DOF designs.

5. Selecting a Robotic Arm

The following performance equation was developed to aid in the final selection of a robotic arm from a set of successful designs yielded from a search:

$$P = k_{DOF}\left(\frac{\text{DOF}}{\text{DOF}_{MAX}}\right) + k_{COAR}\left(\frac{\overline{COAR}_{02}}{\overline{COAR}_{02MAX}}\right)$$
$$+ k_R\left(\frac{R_{02}}{R_{02MAX}}\right) - k_w\left(\frac{\hat{\overline{w}}_{12}}{\hat{\overline{w}}_{12MAX}}\right) + k_S\left(\frac{S_{02}}{S_{02MAX}}\right), \tag{25}$$

where R_{02}, $\hat{\overline{w}}_{12}$, and S_{02} are measures of joint-angle displacement, manipulability (dexterity), and simulation steps (simulated speed). Subscripts indicate when the criterion is measured. The (max) indicates maximum measured values for successful designs during a search. The ks are weights to specify the contribution of each measure.

The average consumption of available redundancy is

$$\overline{COAR}_{02} = \left[\frac{\int_{t0}^{t_2}(COAR)dt}{t_2 - t_0}\right] \tag{26}$$

and is lowest when the arm can avoid deep penetration into the repelling fields and maneuver in the *free space* in between. t_0 is the initial time the end effector enters the enclosure, t_1 is when the arm enters the target workspace, and t_2 is when it completes the target workspace task. This measure gives an indication of the obstacle avoidance effort required to *fit* a design in a given enclosure.

Joint-angle displacement is measured by

$$R_{02} = \int_{t0}^{t_2}\left(\sum_{i=1}^{DOF}|\Delta\theta_i(t)|\right)dt, \tag{27}$$

where $\Delta\theta_i(t)$ is the change in joint-angle θ_i during a simulation step. R_{02} is related to the mechanical work required to maneuver through an enclosure and therefore is related to the financial operating expenses and the usable life of an arm.

Although many robotic arm dexterity measures have been proposed [25-27], one of the most referenced is the *measure of manipulability* [25]:

$$w = \sqrt{\det(\mathbf{JJ}^T)}, \tag{28}$$
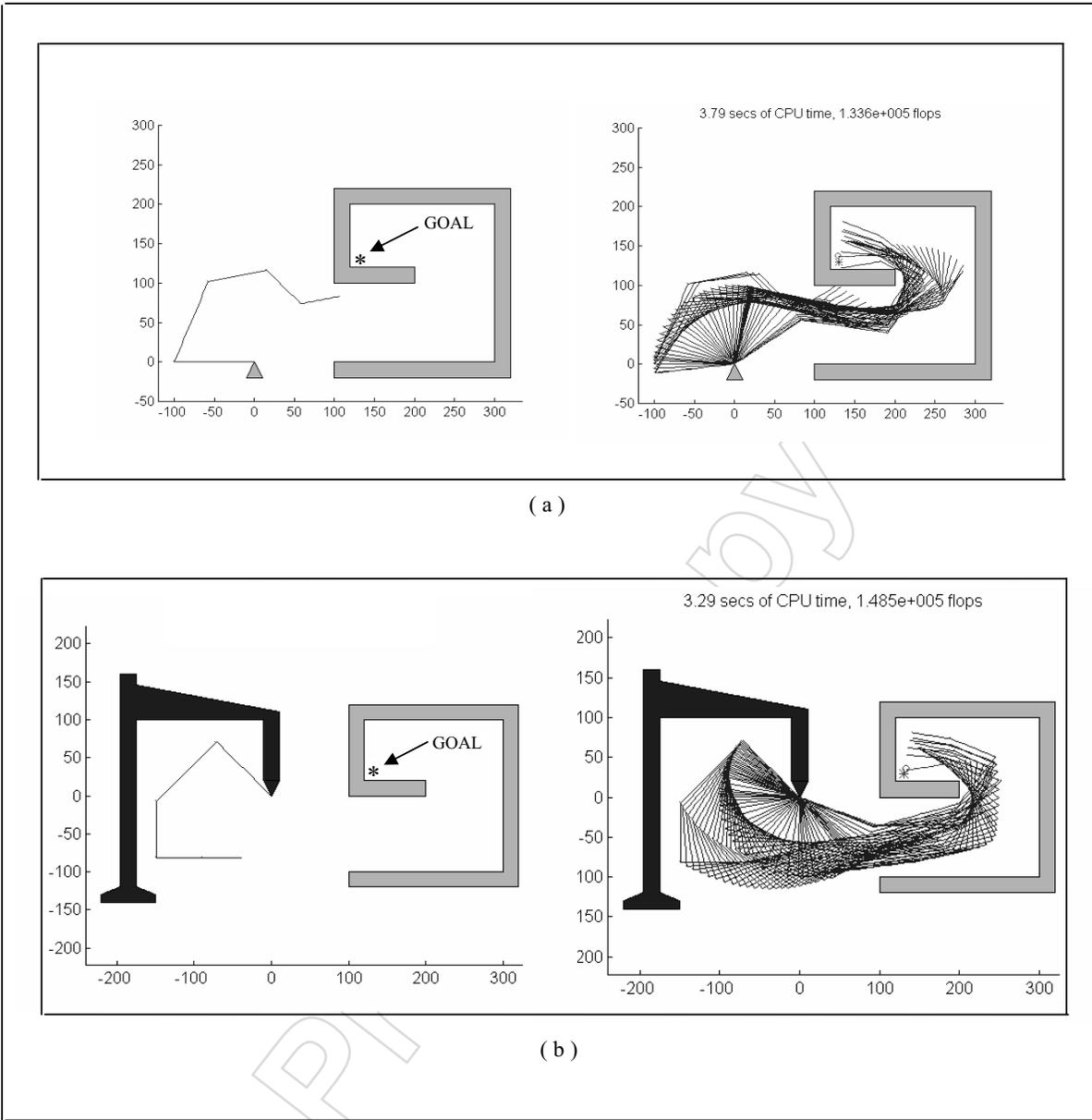
Figure 13. Two successful designs with very different initial configurations but using identical repelling-field settings

which gives an indication of how far a robotic arm configuration is from a singularity (i.e., $w = 0$ at a singularity) and is measured here in the target workspace to focus on the arm's dexterity while performing tasks; an average manipulability is defined for all configurations along this trajectory as

$$\overline{w}_{12} = \left[\frac{\int_{t1}^{t2} \left(\sqrt{\det(\mathbf{JJ}^T)} \right) dt}{t_2 - t_1} \right], \qquad (29)$$

and since manipulability is a function of link lengths, the measure is normalized here:

$$\hat{\overline{w}}_{12} = \left[\frac{\overline{w}_{12}}{\hat{w}_{max}} \right], \qquad (30)$$

termed the *normalized average manipulability*, ranging from 0 to 1. The maximum manipulability $\hat{w}_{max}$ for each design corresponds to an optimal configuration $\theta_{\hat{w}_{max}}$, which is found at
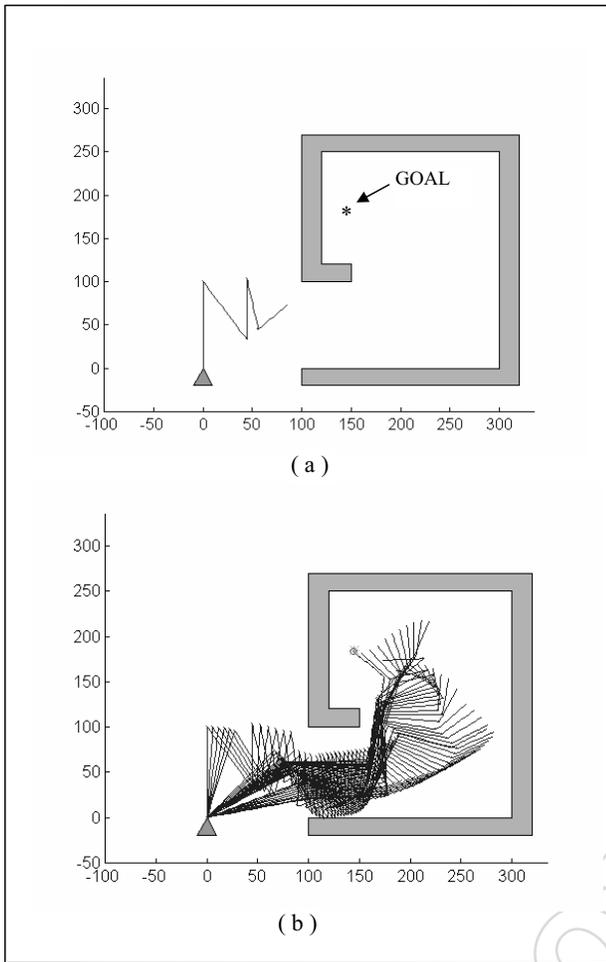
Figure 14. An initial arm configuration that results in an undesirable trajectory (even though goal reached)

$$\nabla_\theta w(\theta)\big|_{\theta=\theta_{\hat{w}_{\max}}} = 0. \tag{31}$$

With a good initial guess, $\theta_{\hat{w}_{\max}}$ is found for each design using only a few gradient steps.

The gradient-search algorithm is

$$\theta_t = \theta_{t-1} + \eta\left[\nabla w(\theta_{t-1})\right], \tag{32}$$

where η is the constant gradient step size selected for fast convergence without overshoot or divergent oscillation. *Manipulability* gives an indication here of the potential singularity problems that may occur when the arm is used for tasks in the given environment; this includes tasks other than the one tested for.

Simulated speed (S_{02}) is simply a measure of the number of simulation steps in a trajectory and gives an indication of

how the desired end-effector TCP velocity is compromised to achieve obstacle avoidance; for example, if local minima or high COAR are encountered, the simulated speed will likely be reduced.

DOF is likely the most significant measure to consider since it can have the greatest impact on financial operating costs and initial expense.

6. Search Results

6.1 Example Search 1

In this example, a heuristic search was applied to the (100, 110, 75, 60, 50)-cm 5-DOF initial design and complex enclosure constructed from the simulation primitives shown in Figures 8 and 9. This search only changes link lengths by 10 cm, two links at a time. A total of 3326 designs were tested, resulting in the 31 new designs shown in Table 1 and Figure 15. Nine of 174 *minor-hit* designs led to successful designs and are also shown in Figure 14. Three 4-DOF designs were produced from the initial 5-DOF design (i.e., designs 22, 25, and 28 in Table 1), with the third of these having the lowest average COAR (0.16) and the second having the highest normalized average manipulability within the target workspace (0.73). Design 10 had the least joint-angle displacement (11.77 radians), and 3 designs used the least simulated time (101 steps) (i.e., designs 2, 4, and 24 in Table 1). Using equation (25), with all $ks = 1$, design 22 in Table 1 can be chosen as the final design (i.e., $P = 1.97$); this design is shown in Figure 16.

6.2 Example Search 2

In this example, a heuristic search was applied to the (90, 120, 95, 50, 40)-cm 5-DOF initial design and unibody automobile interior shown in Figure 4. This search also changes link lengths by 10 cm, two links at a time. For this simpler example, the enclosure was not constructed from simulation primitives, but all repelling fields were established using the same rules as those used for creating primitives. A local attractor was set at the bottom of the enclosure opening to help unfold the arm. This search tested 3123 arm trajectories and yielded 2189 new designs, including 104 four-DOF designs. Another search (a "reduced heuristic search") that targets link 4 for reduced DOF quickly yielded 15 four-DOF designs (and 41 four-DOF designs). One of these is shown in Figure 17. This design also had some of the best manipulability, COAR, simulated speed, and joint-angle displacement over its test trajectory.

7. Hyperredundant Design

A 10-DOF hyperredundant robotic arm design is shown in Figure 18a, and in Figure 18b, it is shown reaching all of the target workspace. This demonstrates that as many as eight obstacle-avoidance points can be controlled simultaneously.

SEARCH
ITERATION

4 = 4-DOF DESIGN
C = LEAST CONSUMPTION OF AVAILABLE REDUNDANCY
R = LEAST JOINT-ANGLE DISPLACEMENT
w = HIGHEST NORMALIZED AVERAGE MANIPULABILITY
S = HIGHEST SIMULATED SPEED
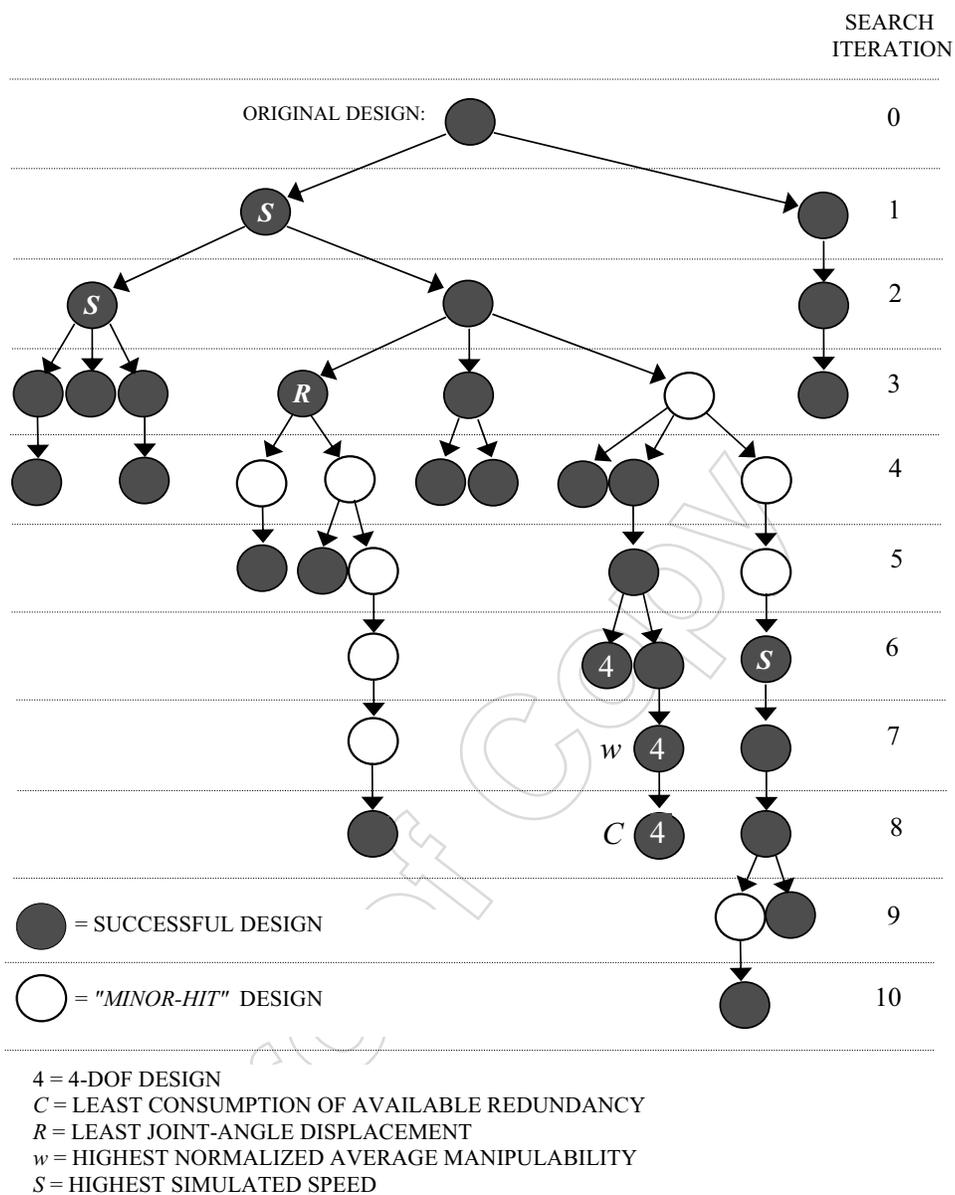
Figure 15. Examples from example search 1
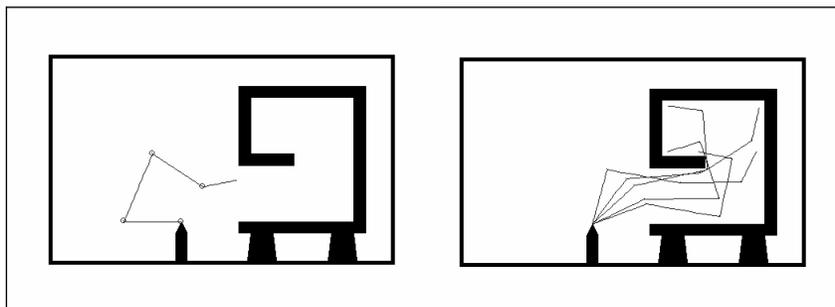
Figure 16. Selected design from example search 1

Table 1. Results for example search 1

| # | Link Lengths (cm) | | | | | S_{02} | R_{02} | $\hat{\bar{w}}_{12}$ | DOF | $\overline{COAR}_{02}$ |
	L1	L2	L3	L4	L5					
1	100	110	75	60	50	102	12.71	0.72	5	0.40
2	100	110	85	50	50	**101**	12.94	0.70	5	0.44
3	100	110	85	60	40	104	12.75	0.70	5	0.39
4	110	110	85	50	40	**101**	12.72	0.70	5	0.60
5	100	110	95	40	50	103	12.53	0.71	5	0.31
6	100	120	85	60	30	104	13.35	0.68	5	0.37
7	110	100	95	50	40	103	13.25	0.69	5	0.42
8	110	110	95	40	40	103	12.90	0.69	5	0.30
9	110	110	95	50	30	102	12.85	0.68	5	0.26
10	90	120	95	40	50	102	**11.77**	0.70	5	0.25
11	100	120	95	40	40	102	12.27	0.70	5	0.36
12	90	120	95	60	30	104	13.09	0.70	5	0.37
13	120	90	95	50	40	103	12.54	0.70	5	0.37
14	110	110	95	60	20	102	13.13	0.69	5	0.30
15	100	120	105	30	40	103	12.73	0.70	5	0.34
16	100	120	105	40	30	102	12.75	0.68	5	0.25
17	110	110	105	20	50	102	12.63	0.70	5	0.29
18	100	120	105	20	50	104	12.39	0.70	5	0.26
19	80	140	85	40	50	102	13.89	0.69	5	0.54
20	90	130	105	20	50	103	12.08	0.70	5	0.22
21	100	120	105	10	60	104	12.24	0.72	5	0.22
22	100	130	105	**0**	60	116	12.46	0.71	**4**	0.17
23	100	110	105	10	70	104	12.12	0.73	5	0.28
24	120	100	105	30	40	**101**	11.89	0.68	5	0.17
25	110	110	105	**0**	70	107	12.35	**0.73**	**4**	0.28
26	120	100	95	30	50	105	13.05	0.70	5	0.44
27	90	120	75	30	80	105	13.05	0.72	5	0.54
28	120	110	95	**0**	70	115	13.39	0.69	**4**	**0.16**
29	120	100	85	40	50	103	12.45	0.70	5	0.40
30	120	100	75	40	60	118	14.13	0.70	5	0.60
31	120	90	75	40	70	106	13.12	0.71	5	0.57

[WHAT DO BOLD NUMBERS REPRESENT?]
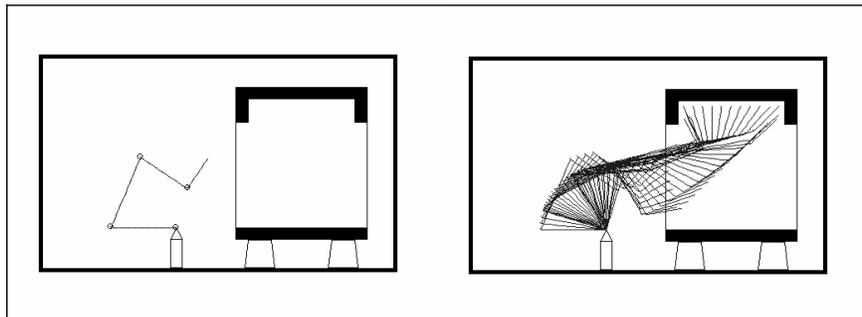
Figure 17. Selected design from example search 2

8. Future Research

Future plans for this research include extending the methodology to a complete 3-D workspace (funding for this is presently being sought). Future research will also likely include deriving a relationship between COAR and DOF and minimizing COAR directly such that a trajectory's success can be guaranteed in advance.

9. Conclusions

Many researchers have explored fixed link length, fixed DOF robotic arms in constrained spaces, and variable link-length arms in unconstrained spaces. This article presents a simulation for designing redundant robotic arms for enclosed spaces by permuting link lengths and DOF, then comparing feasible designs for maximum simulated
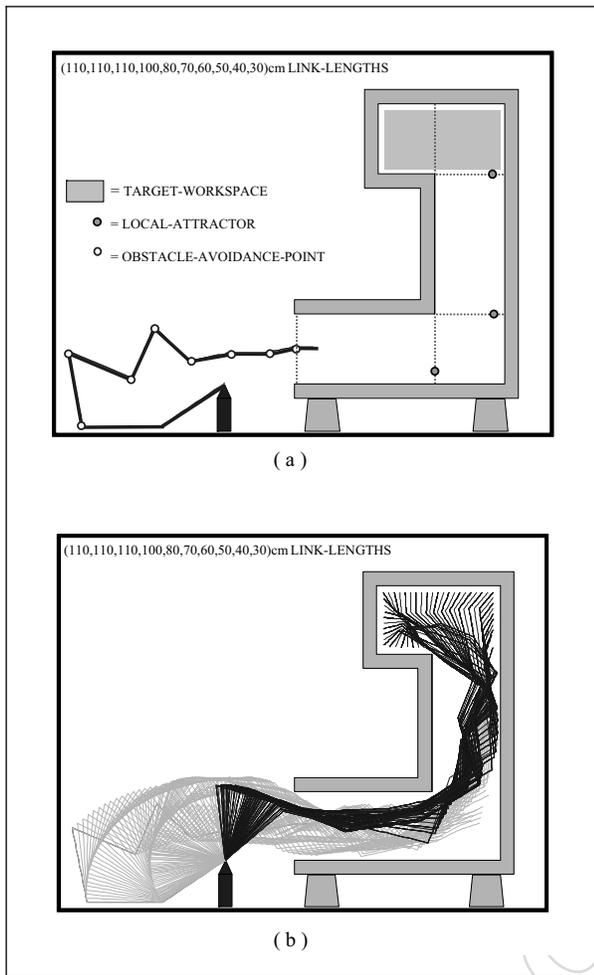
Figure 18. A 10 degree-of-freedom robotic arm

speed and dexterity and minimum joint-angle displacement, DOF, and consumption of available redundancy over test trajectories. The consumption of available redundancy is evaluated using a newly developed measure. Enclosures can be defined by assembling simulation primitives using simple rules, and a new path-planning technique is developed to allow maneuvering through complex enclosures. Also, since an arm is only repelled within close proximity of obstacles, it can maneuver relatively unconstrained throughout a significant part of the enclosure.

The use of pseudo-inverse robotic arm control relies on approximate least squares solutions, and there are always stability concerns when null space is used to satisfy secondary tasks. However, several measures have been taken here to allow path-planning trajectories a degree of robustness in this environment; more important, the resulting arm designs can be used with other control schemes since they have been proven capable of operating within the given

enclosure. This is especially true for most industrial arm implementations that are typically not programmed offline; they are simply moved through a trajectory by a technician while the robot controller is put into "teach mode."

10. Appendix

Terms

COAR	Consumption of available redundancy
DOF	Degree(s) of freedom
TCP	Tool center point
D_j	Distance of manipulator from obstacle
d_{ABORT}	d from obstacle to consider a crash
$\mathbf{J}$	Jacobian matrix
$\mathbf{J}^{-1}$	Inverse of Jacobian matrix
$\mathbf{J}^{\#}$	Pseudo-inverse of Jacobian matrix
K_j	Repelling velocity in field j
k_p	Weight of performance criterion p
L_i	ith link length
m	Dimension of workspace
n	Number of degrees of freedom
R	Total joint-angle displacement
S	Number of simulation steps
$t_{\#}$	Time index
t_j	Thickness of repelling-field j
w	Manipulability measure
$\overline{w}$	Average manipulability
$\hat{w}_{\max}$	Maximum possible manipulability
$\hat{\overline{w}}$	Normalized average manipulability
$\mathbf{x}$	Cartesian position vector
x	x component of Cartesian position
y	y component of Cartesian position
$\dot{\mathbf{x}}$	Cartesian velocity vector
$\dot{x}$	x component of Cartesian velocity
$\dot{y}$	y component of Cartesian velocity
u_j	Unit vector in field j
V_e	End-effector velocity
V_j	Fraction of end-effector velocity
∇	Gradient
η	Gradient step size
$\boldsymbol{\theta}$	Joint-angle vector
θ_i	ith joint angle
$\Delta\theta_i(t)$	Change in ith joint angle during time t
$\boldsymbol{\theta}_0$	Initial joint angles for gradient search
$\boldsymbol{\theta}_{\hat{w}_{\max}}$	Joint angles for optimal manipulability
$\dot{\boldsymbol{\theta}}$	Joint-angle velocity vector
$\dot{\theta}_i$	ith joint-angle velocity
$\dot{\boldsymbol{\Psi}}$	Secondary priority arbitrary vector
$\dot{\boldsymbol{\beta}}$	Tertiary priority arbitrary vector
$\dot{\boldsymbol{\Lambda}}$	Desired obstacle-avoidance point velocity
Γ	$\dot{\boldsymbol{\Psi}}$ to $\dot{\boldsymbol{\Lambda}}$ transformation matrix
$\|\,\|$	Euclidean norm
Ω	Angle between $\dot{\boldsymbol{\Psi}}$ and orthogonal projection
δ	Angle between exact and approximate solutions

11. References

[1] National Research Council. 1992. *Dispelling the manufacturing myth: American factories can compete in the global marketplace*. Washington, DC: National Academy Press.

[2] Nof, S. Y. 1999. *Handbook of industrial robots*. 2d ed. New York: John Wiley.

[3] TWI World Center for Materials Joining Technology. 2004. *TWI image library*. Retrieved from www.twi.co.uk/j32k/ unprotected/band_1/imglb013.html

[4] Greville, T. N. E. 1959. The pseudoinverse of a rectangular or singular matrix and its applications to the solutions of systems of linear equations. *SIAM Review* 1 (1): 38-43.

[5] Nakamura, Y., and H. Hanafusa. 1984. Singularity low-sensitive motion resolution of articulated robot arms. *Transactions of the Society of Instrument and Control Engineers* 20 (5): 453-59.

[6] Hanafusa, H., T. Yoshikawa, and Y. Nakamura. 1981. Analysis and control of articulated robot arms with redundancy. In *Proceedings of the IFAC Control Science and Technology 8th Triennial World Congress*.

[7] Nakamura, Y., H. Hanafusa, and T. Yoshikawa. 1987. Task-priority based redundancy control of robot manipulators. *International Journal of Robotics Research* 6 (2): 3-15.

[8] Maciejewski, A. A., and C. A. Klein. 1985. Obstacle avoidance for kinematically redundant manipulators in dynamic varying environments. *International Journal of Robotics Research* 4 (3): 109-17.

[9] Nemac, B., L. Zlajpah, and D. Omrcen. 2003. Stability of null space control algorithms. In *Proceedings of the RAAD'03 Int. Workshop on Robotics*, Alpe-Adria-Danube region, Cassino.

[10] English, J. D., and A. A. Maciejewski. 2000. On the implementation of velocity control for kinematically redundant manipulators. *IEEE Transactions on Systems, Man, and Cybernetics* 30 (3): 233-37.

[11] Wunderlich, J. T. 2004. Design of a welding arm for unibody automobile assembly. In *Proceedings of the IMG04 International Conference on Intelligent Manipulation and Grasping*, Genoa, Italy.

[12] Wunderlich, J. T., and C. G. Boncelet. 1996. Local optimization of redundant manipulator kinematics within constrained workspaces. In *Proceedings of IEEE International Conference on Robotics and Automation*, Minneapolis, MN.

[13] Wunderlich, J. T. 1996. Optimal kinematic design of redundant and hyper-redundant manipulators for constrained workspaces. Ph.D. diss., University of Delaware.

[14] Khatib, O. 1985. Real-time obstacle avoidance for manipulators and mobile robots. In *Proceedings of the IEEE International Conference on Robotics and Automation*.

[15] Chirikjian, G. S., and J. W. Burdick. 1990. An obstacle avoidance algorithm for hyper-redundant manipulators. In *Proceedings of the IEEE International Conference on Robotics and Automation*.

[16] Li, T. Y., and H. C. Chou. 2003. Motion planning for a crowd of robots. In *Proceedings of the IEEE International Conference on Robotics and Automation*.

[17] Barraquand, J., and J. Latombe. 1991. Robot motion planning: A distributed representation approach. *International Journal of Robotics Research* 10:628-49.

[18] Kavaki, J., P. Svestka, J. Latombe, and M. Overmars. 1996. Probabilistic roadmaps for fast path planning in high-dimensional configuration spaces. *IEEE Transactions on Robotics and Automation* 12:566-80.

[19] Nakamura, Y. 1991. *Advanced robotics: Redundancy and automation*. Reading, MA: Addison-Wesley.

[20] Hollerbach, J. M. 1988. Optimum kinematic design for a seven degree of freedom manipulator. In *Proceedings of the International Symposium of Robotics Research*.

[21] Shiller, Z., and S. Sundar. 1991. Design of robotic manipulators for optimal dynamic performance. In *Proceedings of the IEEE International Conference on Robotics and Automation*.

[22] Mayorga, R. V., B. Ressa, and A. K. C. Wong. 1991. A kinematic criterion for the design optimization of robot manipulators. In *Proceedings of the IEEE International Conference on Robotics and Automation*.

[23] Paredis, J. J., and P. K. Khosia. 1993. Kinematic design of serial link manipulators from task specifications. *International Journal of Robotics Research* 12 (3): 274-87.

[24] Chirikjian, G. S., and J. W. Burdick. 1991. Parallel formulation of the inverse kinematics of modular hyper-redundant manipulators. In *Proceedings of the IEEE International Conference on Robotics and Automation*.

[25] Yoshikawa, T. 1984. Manipulablity of robotic mechanisms. *International Journal of Robotics Research* 4 (2): 3-9.

[26] Park, F. C., and R. W. Brockett. 1994. Kinematic dexterity of robotic mechanisms. *International Journal of Robotics Research* 13 (1): 1-15.

[27] Doty, K. L., C. Melchiorri, E. M. Schwartz, and C. Bonivento. 1995. Robot manipulability. *IEEE Transactions on Robotics and Automation* 11 (3): 462-68.

[28] Kim, J., G. Marani, W. K. Chung, and J. Yuh. 2002. Kinematic singularity avoidance for autonomous manipulation under water. In *Proceedings of PACOMS-2002*.

[29] Theingi, I.-M. Chen, C. Li, and J. Angeles. 2003. Managing singularities of 3-DOF planar parallel manipulators using joint-coupling. In *Proceedings of the 11th World Congress in Mechanical and Machine Science*, Tianjin, China.

[30] Kim, D., and W. Chung. 1999. Analytic singularity equation and analysis of six-DOF parallel manipulators using local structurization method. *IEEE Transactions on Robotics and Automation* 15 (4): **000-000**.

[31] Wampler, C. W. 1986. Manipulator inverse kinematic solutions based on vector formulations and damped least-squares methods. *IEEE Transactions on Man-Machine Systems* 16 (1): 93-101.

Joseph T. Wunderlich *is the primary computer engineering program coordinator for Elizabethtown College. Previously, he worked for Purdue University as an assistant professor and for IBM as a researcher and hardware development engineer. He received his Ph.D. in electrical and computer engineering from the University of Delaware, his master's degree in engineering science/computer design from The Pennsylvania State University, and his B.S. in engineering from the University of Texas at Austin. He also has experience as an industrial automation consultant.*