A Method for Accurate Simulation of Robotic Spray Application Using Empirical Parameterization

Erik D. Goodman
Leslie T. W. Hoppensteadt
A. H. Case Center for Computer-Aided Engineering and Manufacturing
Michigan State University
East Lansing, MI 48824

Abstract
As part of the development of an accurate simulator of robotically-applied spray coatings on sculptured surfaces, a tabular technique was developed for representing with arbitrary accuracy the spray pattern from a robot sprayer. The method allows parameterization of this spray distribution using a single spray test pattern applied by the robot, without symmetry assumptions. The thickness of the test pattern is measured at a user-specified grid of points. These data are used to calculate a least-squares solution to an overdetermined set of linear equations, with optional user-scalable smoothing (via regularization) to suppress effects of measurement noise. In closed-loop testing of the parameterization and simulation process using pseudorandom noise in the measurement data, the solution table entries typically contain less noise than the individual thickness measurements on which they are based, even when measured data points are not much more numerous than table entries.

1. Introduction
In some applications, it is desirable to program a spray robot to apply paint or other coatings to a variety of manufactured parts. Off-line programming of the robot is a complex task, particularly if the part is complex and/or the spray material is to be applied non-uniformly. The non-uniform distribution of the instantaneous spray pattern itself implies that overlapping passes must be made, and it may be very difficult to achieve the desired result in some situations, requiring many attempts to develop an acceptable program by trial and error. This is the motivation for the development of SPRAYTOOL, which allows the user to simulate the robotic application of spray coatings. Using this tool, the user may check the program during development to ascertain its correctness or partial correctness without the expense of repeatedly using the robot and measuring the results. The overall operation, algorithms, and features of SPRAYTOOL are described elsewhere [1], and are presented only in outline in this paper. A new empirical technique for accurate measurement of the spray distribution emerging from the robot and for use of these measurements to parameterize the simulator is presented here. The technique does not require special hardware or instrumentation, even to measure strongly asymmetrical spray patterns. Using this technique, complex coatings can be simulated with accuracy held to within a few percent.

1.1 Background
Robot simulation has been used for many purposes (see [2], [3], [4], [5]). Commercial robot simulation products such as Deneb, ROBCAD, and Soma are used for collision detection, workcell design and off-line programming, for example. While many of the mathematical and simulation techniques (e.g., [6]) used by other simulators are also employed in SPRAYTOOL, several other techniques were developed to deal with the problems peculiar to application of spray materials.

The three principal means of programming robots, as described for example by Lozano-Pérez [7], are guided programming (often done through the use of a teach pendant or direct manipulation of the robot, but also including off-line model-based guiding), explicit or robot-based programming (off-line development of the robot program, perhaps using a geometric model of the part to be sprayed, but employing a language for specifying explicitly the commands to access sensors and specify robot motions), and task-level programming, which allows programming by specification of the goals or targeted effects on the objects to be manipulated by the robot. Robotic spray paint-
ing frequently employs guided programming or teaching-in, in which an operator drives the robot manually through the pattern of motion which it will subsequently repeat (see, for example, [8], [9], [10] and discussion of difficulties with teach-in spray programming by Klein [11]). While this method is useful when coating thickness need not be precise, it is not appropriate for many problems such as the application of tightly controlled nonuniform coatings to complex sculptured surfaces. On the other hand, the general problem of developing task-level programming to generate variable thickness coatings on sculptured surfaces (i.e., automatically generating the explicit robot program given the desired thicknesses) is extremely difficult, and no system for doing it has been described in the literature. However, many simulators able to help with off-line preparation of robot programs have been written, and several commercial products exist (for example, Deneb, ROBCAD, and Soma). Klein [11], [12] describes a simulator specifically designed for spray applications, and the three commercial simulators mentioned all provide rudimentary spray simulation capabilities. ROBCAD has some capability for automatically generating a robot spray program given surfaces to be sprayed and a set of parameters describing the paint fan ([13]). However, since empirically-based parameterization of these spray simulators is not an integral part of their design, these simulators make strong simplifying assumptions about the shape of the spray distribution emerging from the tool tip. Klein [12], for example, assumes a conical pattern with uniform thickness within an inner cone and sinusoidal falloff to an outer cone. These simulators are generally useful for determining the areas covered by swaths, and may provide sufficient information for non-demanding applications, given adequate parameterization of their spray distributions. In contrast, the system described here provides a simple technique for experimentally measuring the steady-state output of a sprayer during actual operation, as a tabular function of 2-D location relative to the approach vector and distance from the tool tip. It requires spraying of only a single test pattern for each given set of spray conditions. After one-time least-squares solution of a large set of linear algebraic equations, the resulting measurements are used directly in the simulator to allow accurate simulation of the material deposition, with sufficient speed to be a very practical tool. It may be possible to revise other simulators to take advantage of this representation and parameterization technique.

2. Overview of the SPRAYTOOL Simulator

The SPRAYTOOL simulator is initialized by reading:

- a description of the geometry of the part to be sprayed,
- a post-processed robot program

Then, using commands entered via menus within an X-Windows-based user interface, the user controls generation and display of pseudocolor images which use various hues to encode the resulting spray thickness or the deviations of the resulting spray thicknesses from user-specified desired thicknesses and tolerances. These desired thicknesses and tolerances are defined as variable-offset surfaces relative to the part surfaces and the desired thicknesses, respectively. The user may view the pseudocolor output display from any angle, and may save intermediate stages for subsequent re-use with additional robot program segments, allowing the sequential development of building blocks which have been judged by the user to be acceptable.

2.1 Geometric Data Defining Surfaces to be Sprayed

The simulator reads geometry in either of two formats -- a proprietary internal format used by General Dynamics, or an industry-standard IGES Non-Uniform Rational B-Spline (NURBS) format [14]. The user provides the file name containing the geometry, and may also control the level of discretization used by the simulator in dividing the surfaces into triangular meshes. All surface evaluations, including points and their normals, are done at this time; the part geometry data structure for the simulation consists of linked lists of points and normals, linked lists of triangles, and pointers between them. The input to SPRAYTOOL may be, but need not be, a set of boundary representation (B-rep) solids. All that is required is a set of oriented surfaces -- that is, surfaces with normal vectors all facing away from the interior of the part bounded by the surface. If the surfaces read are not properly oriented, SPRAYTOOL provides a facility for interactively viewing and reversing the normals (by reparameterizing the surfaces with the order of the parametric variables reversed [15]) to obtain surfaces with outward-directed normal vectors. Problems such as self-intersecting surfaces are not considered, since the input data will normally be the output of either a surface modeler or a solid modeler, and such cases are not ordinarily used in descriptions of parts to be manufactured. At points on common boundaries between surfaces, there are potentially multiple distinct normal vectors. Because spray accumulation in SPRAYTOOL is measured along surface normals, and the part surface is actually moved along the normals as the spray accumulates, the user is given the option of asking SPRAYTOOL to average appropriate outward-facing normals at such points to create a single normal. This prevents SPRAYTOOL from creating gaps in the spray coverage at convex patch intersections, and produces a somewhat more realistic model for accumulation at concave patch intersections.

For each robot program path segment, the simulation would potentially need to consider in turn each discretized
point on the object to be sprayed to determine the amount of spray deposited. In order to avoid unnecessary calculations, it is important to use a volume decomposition technique on the objects to be sprayed to determine those volume elements which are intersected by a particular spray envelope, and to exclude from consideration any points not contained in those volume elements. SPRAYTOOL uses fixed-size, cubic volume elements (voxels) for representation of both the surfaces in the workspace and the envelope of spray produced by the robot. During the design of SPRAYTOOL, both arrays of fixed-sized voxels and recursively divided (e.g., binary space partitioning or octree) subdivision of three-dimensional space were considered. The technique which was selected bounds the entire object in a rectangular solid aligned with the robot coordinate system, translates that volume to be located in positive (x,y,z) space with a corner at the origin, and then divides that solid into fixed-size cubic voxels. The voxel indices of any point in space are thus calculated by applying the translation to its world-space (robot programming space) coordinates and simply truncating to obtain integer indices. All surface points in a voxel are stored in a linked list associated with the integer-addressed voxel. Additional pointers associate each point with the triangles to which it belongs, and each triangle with the points which define it.

An algorithm was developed for quickly calculating the indices of voxels which intersect a bounding volume containing the sprayed volume (to a user-specifiable depth) for any spray path segment, allowing the program to exclude all other voxels and the points they contain from any possible deposition during that robot path segment. The ease of enumerating the indices for these voxels led to the choice of uniform spatial subdivision over a tree-structured representation. Voxel space techniques are commonly applied in visualization and other domains (e.g., [16], [17], [18]) to represent in an approximate form the volumes occupied by either measured data or synthetic volumes, and are particularly useful for the sorts of approximate intersection testing used here ([19]).

Sprayed points are displaced along their normals as they accumulate spray, and points which accumulate sufficient spray to move into an adjacent voxel are changed appropriately in the list structures.

2.2 Robot Program Input

The robot program information required by SPRAYTOOL is the set of points and approach vectors defining the path segments, the interpolation method to be used between each pair (linear, circular, constant joint angle velocity, etc.), and spray gun parameter values for each path segment. This is now read in a specialized postprocessor format used by General Dynamics for programming the robot; however, any other format which includes the same information would be easily usable. Spraytool performs further discretization of this program file as it reads it in, approximating nonlinear or non-constant-approach-vector path segments by sequences of short linear segments of constant approach vector orientation. The user is allowed to specify the maximum change in orientation about any axis of the spray head between any two adjacent segments; similarly, for a circular segment, the user determines the maximum arc (in degrees) to be approximated by a linear segment. Thus the user can control the tradeoff of speed versus accuracy.

2.3 Internal Representation of Spray Distributions

In order to represent a particular spatial distribution of spray emerging from the robot, SPRAYTOOL uses one "fundamental" table and two "derived" tables. The fundamental table is an m x n matrix which represents instantaneous rates of deposition of spray material on a flat surface perpendicular to the approach vector (spray axis) at a fixed distance from the spray head (see Figure 1). The matrix is tall and wide enough to represent all of the spray (excepting "overspray"). For each particular spray pattern (determined by material delivery rate, material viscosity, several air pressures, etc.), the long axis of the spray pattern is aligned with the columns of the matrix, and the matrix center is on the axis of the spray head (odd numbers of rows and columns are always selected, to enable this). The en-

Figure 1. A fundamental table (13 x 7) superimposed on a spray distribution on a flat surface which is normal to the approach vector.
tries sum to the rate of material application, less any spray deflected by the surface being sprayed. This table is based on the parameterization measurements made from actual spray applications, as described in the following section. SPRAYTOOL uses straight-line-of-flight assumptions to extrapolate from this table to deposition on surfaces at distances other than that at which the distribution was measured; however, the user is counseled that if this assumption is not warranted, additional fundamental tables should be created for other distances from the sprayer, so that SPRAYTOOL can interpolate between tables, rather than extrapolating from a single one.

The fundamental table is used to derive two other tables, called the spherical table and the path table. The spherical table is similar to the fundamental table, except that its entries are scaled to represent the amount of spray which would be deposited on the interior of a sphere of radius equal to the offset of the fundamental table from the spray head (see Figure 2). Thus, the center entries of the spherical table and the fundamental table are the same. However, off-center entries in the spherical table are calculated assuming straight-line flight of particles measured in obtaining the entries in the fundamental table. The spherical table is used for all “non-normal” deposition calculations, i.e., those in which the acute angle between the line normal to the surface at the point being sprayed and the approach vector differ by more than n (typically 5) degrees. When the approach vector and normal vector are nearly aligned, either the fundamental table or the path table described below is used, depending on the user’s selection.

If the fundamental table is of size m x n, then the path table is a matrix of size mn x mn, and contains the integrals of paths from any entry in the fundamental table to any other entry in the fundamental table, sprayed at a constant reference speed. While in reality the spray distribution moves across a point being sprayed, it will be described below as if the point moved along a path through the fundamental table. The entry \((i1,j1)\) in the path table is the rate of deposition of spray on a point (with a normal aligned with the approach vector) on a path which starts at position \((i1,j1)\) in the fundamental table and ends at position \((i2,j2)\). The integrals are approximated using trapezoidal segments along paths through the fundamental table, using linear interpolation between adjacent entries to estimate the value at the intersection of the path with the lines joining each pair of entries (see Figure 3).

The path table is used only for paths along which the approach vector is nearly normal to the surface for the points receiving spray.

Details of the spray accumulation calculation are described elsewhere [1], and include tracking the changing shape of the surface being sprayed, several methods for display of accumulated spray depths, etc.

3. An Algorithm for Parameterizing the Spatial Distribution of the Spray

It is desired to parameterize the fundamental table described above using measurements from the robot in operation under the conditions to be simulated. It is assumed that the depth of deposition at a point (or small region) of a part is measurable by the user, or else there would be no way to perform quality assurance on the parts, with or without a simulator. The best technique for measuring the thickness of deposition clearly varies with the properties of the parts and of the materials being sprayed. Since sprayers are typically turned on or off only while the robot is pointed away from any part surface (in order to avoid depositing spray during the transients induced by start-up or shutdown of the sprayer), the only spray distribution usually required is the steady-state distribution. Clearly, that must be a relatively stable and repeatable distribution, or there is no way to program the robot to generate a desired spray deposition on a part. However, the task of measuring that spray distribution pattern for a particular set of conditions (materials, pressures, etc.) is complicated by several problems:

- Obtaining accurately measurable depths of the material usually requires several "coats," since to allow the material to accumulate to an accurately measurable depth on during one spray pass would cause the material to run. A sprayer is ordinarily not deliberately operated in this manner, and it would not yield useful information for normal
Turning the sprayer on and off at a fixed location does not provide a "snapshot" of the steady-state spatial distribution, since the deposition is strongly affected by the transients.

Using a slit or other orifice to "scan" the spray pattern in some fashion (after it has reached steady state) does not provide an accurate sampling of that pattern, since the changes in air flow, etc., induced by the orifice will strongly influence the final spray pattern.

These problems are overcome by using the robot to spray a circular arc (typically about 330 degrees) as portrayed in Figure 4. Multiple passes may be made, allowing drying of the deposit between passes, until an accurately measurable thickness is obtained. At fixed radial increments (typically 10 degrees), a radial line is defined, with a reference point at the intersection of the line and the circular arc. Then at fixed increments (typically 0.25") from this reference point along each line, the depth of the deposition is measured, typically using an indirect measurement technique appropriate to the material being sprayed. This set of depth measurements, represented as a column vector \( b \) below, is the data set from which the parameterization of the spray table is calculated. No measurements are made in the quadrant of the circle in which the sprayer is turned on and off, in order to avoid the effects of the transients. The radius of the circular arc (or "test circle") should be as large as practical (relative to the size of the spray pattern), as the fitting technique below will approximate short arc segments (usually 10 degrees) by their tangent lines.

Figure 4 shows a fundamental table superimposed on the test circle in several positions. The fundamental table is always sized by the program such that its height is larger than the width of the spray pattern at its widest point, and its width is larger than the width of the spray pattern at its narrowest point, using the information evident in the circle test pattern. This is to ensure that the outermost entries in the fundamental table always be zero. Prior to determination of this table size, the user is asked to specify a Minimum Significant Data Value ("MSDV") below which all measurements will be considered to be negligible, and will not cause enlargement of the table size. This MSDV is thus particularly important for non-circular spray distributions.

The refinement of the spray distribution is determined by 1) the dimensions \( m \) and \( n \) of the fundamental table, and 2) the user-selected size (in inches) represented by each cell in the table. There is a clear tradeoff between the spatial resolution of the table (determined by the number of cells selected, \( m \times n \)) and the noise-suppression capability of the

Figure 3. Integrals of spray accumulation along paths through the fundamental table are stored in path table, which has rows and columns indexed by the places where the path enters and leaves the fundamental table, respectively.
solution algorithm, based on the ratio of number of parameters to be determined to the number of measured data points available. The defaults in the data analysis program provide for a spatial resolution (cell size) which is 2.1 times the distance between measured data points. The program automatically sizes the fundamental table such that the product of the (user-specified) cell size and the number of elements in each direction slightly exceeds the size of the spray pattern. The solution procedure has been found to be more accurate when the cell size is not an exact integer multiple of the distance between the measured data points.

If the motion of the robot were linear (i.e., the circle had infinite radius), then each measured data point would represent the spray accumulated at its location as the sprayer (i.e., the fundamental table) passed over it on a straight line path. That is, it represents the integral along a path traversed at a reference speed through the fundamental table, where the path is determined by the position of the point and the radial line it is on, as shown in Figure 5. (The effect of the nonlinearity of the path is minimized by making the radius relatively large). Thus the data points represent a large set of path integrals through the table at various angles, and can be used to generate a set of linear equations in terms of the fundamental table entries. To allow solving for the entries in the fundamental matrix, it is first recast as a column vector of unknowns, \( x \) (see Figure 6). Then, just as paths through the spray distribution are approximated by trapezoidal integration along paths through the fundamental table to calculate spray deposition, these same relationships (the coefficients representing the weighting of each table entry in the path integral) form the \( A \) matrix which defines the system.

Eq. 1) \[ A x = b, \]
where \( b \) is the column vector of measured thicknesses.

![Diagram](image)

Figure 4. To generate parameterization data, the robot sprays a circular path with a fixed spray head orientation. This produces the elliptical interior and exterior boundaries shown. The fundamental matrix is superimposed in three locations to illustrate its meaning.
Of course, this system may be over- or under-determined, depending on the cardinality of the vector $b$. A least-squares solution $x$ for this system can be found by calculating:

$$A^T A x = A^T b,$$

then using LINPACK ([20]), for example, to perform singular value decomposition of $A^T A$, calculate its pseudo-inverse, and solve for $x$. However, the formulation of this system of equations clearly tends to yield ill-conditioned matrices, and the effects of errors in the measurements are amplified strongly in the solution vector $x$ (the fundamental table). Negative table entries, non-zero entries on the outer boundary of the table, and other undesirable phenomena typically result from this direct solution process. Steps taken to improve this solution algorithm are described below, after the description of the process used to assess the suitability of a solution algorithm.


A closed-loop procedure for assessing the quality of the parameterization process was developed. This procedure begins with an arbitrary fundamental table. Table 1 is the fundamental table used for the tests described below, and represents a reasonably shaped distribution with some "sharp" high-frequency edges to provide a difficult target for the regularization algorithm. This table is used to calculate a file of simulated "perfect" test data -- the depths which would theoretically be measured on the test circle if it were sprayed using this fundamental table. A separate program allows adding arbitrary amounts of uncorrelated additive and/or multiplicative Gaussian noise to the simulated measurements, producing "noisy" measurements.

```
\begin{align*}
\begin{bmatrix}
  x_{11} & x_{12} & x_{13} & x_{14} & x_{15} \\
  x_{21} & x_{22} & x_{23} & x_{24} & x_{25} \\
  x_{31} & x_{32} & x_{33} & x_{34} & x_{35} \\
  x_{41} & x_{42} & x_{43} & x_{44} & x_{45} \\
  x_{51} & x_{52} & x_{53} & x_{54} & x_{55} \\
  x_{61} & x_{62} & x_{63} & x_{64} & x_{65} \\
  x_{71} & x_{72} & x_{73} & x_{74} & x_{75}
\end{bmatrix}
\end{align*}
```

$m \times n$

```
\begin{align*}
\begin{bmatrix}
  x_{11} \\
  x_{12} \\
  x_{13} \\
  x_{14} \\
  x_{15} \\
  x_{21} \\
  x_{22} \\
  x_{23} \\
  x_{24} \\
  x_{25} \\
  \vdots \\
  x_{74} \\
  x_{75}
\end{bmatrix}
\end{align*}
```

$m n \times 1$

Figure 6. The two-dimensional fundamental table (unknowns) is recast as a column vector $x$, for an example $5 \times 7$ table.

Each $x$ represents a measured data point, which is the integral of the spray deposited on that point as the spray pattern moves over the point along one of the straight lines shown. It is used as an approximation of the integral as the spray pattern moves on the large-radius circular path.

Figure 5. The depth of deposition at each measurement ($X$) is treated as the integral along the corresponding path (here, a vertical line) through the fundamental table (shown as dots).
These noisy data may then be used in the parameterization process to solve for a fundamental table. A problem is introduced when additive noise is used: the normally-zero entries on the fringes of the simulated measurements are made non-zero, which would cause the analysis program to increase the size of the table it produces. This is because the additive noise, while perhaps a few percent of the maximum data values, is large relative to the near-zero measurements at the fringes of the measured data. To combat this effect, just as would be done with real measured data, reasonable values of the MSDV (i.e., the largest "noisy" value which will be treated as negligible) are used in the parameterization tests below to prevent the additive noise from increasing the table size beyond its desired value. For the cases described below, simulated test data were generated at 0.25 inch spacing, and the cell size in the fundamental table was set at its default of 0.525 inches.

The differences between the fundamental table resulting from this closed-loop analysis of noisy data and the original fundamental table reflect the effects of the noise introduced and the degree to which the solution process either amplifies or decreases the effect of that noise.

As a check on the straightforward least squares solution process, a variety of fundamental tables, ranging in size from 5 x 5 up to 17 x 9 were subjected to the closed-loop analysis procedure. For the 17 x 9 case (about an 8 inch x 4.5 inch fan at the measured offset), when no noise was added to the simulated "measurement" data, the maximum error found in the resulting table was 0.014% and the mean of the percent errors was 0.0022%. Given that single-precision floating point calculations are performed, this appears to be good agreement with the expected result. The largest absolute error was 2.5 x 10^-8 (entries in Tables 1-3 are times 1000).

A noisy measurement file was prepared with no additive noise, but with multiplicative noise having a standard deviation of 2% of the table values. This produced a maximum percent noise of 7.05% and a mean percent noise of 1.63% in the measured input file. Analysis of this file produced a fundamental table with errors up to 24.77% and mean error of 4.25% when compared with the original fundamental table. A second noisy measurement file was prepared with no multiplicative noise, but with additive noise with a standard deviation of 0.01. This produced a measurement file with additive noise up to 0.0329 and a mean of 6.98% of the corresponding entry. Analysis of this input yields a fundamental table with maximum error of 14.66% and mean error of 2.99%. The apparent sensitivity of this process to any noise in the input data makes clear the desirability of using additional noise-suppressing techniques to improve the solution process.

The problem was not that the solution algorithm could not handle sufficiently ill-conditioned matrices; in fact, the diagonal sequence of singular values σ_j typically had a range of less than 4 orders of magnitude, and contained no abrupt jumps. However, this problem formulation and solution technique did not include:

- compensating for the high sensitivity of the process to noise in those data, utilizing the intrinsic smoothness which must underlie any spray distribution,
- Forcing zeroes on the outer boundary of the fundamental table, and
- Barring negative entries from the fundamental table.

These issues are treated in the refinement of the parameterization process, as described below.

5. Improving the Parameterization Process

5.1 Enhancing the Smoothness of the Table Via Regularization.

The intrinsic smoothness expected in the fundamental table is attributable to the physical nature of an atomized spray process such as that employed in the spray heads to be modeled. Pressurized spray material might typically be atomized by air flow through a nozzle, then possibly spread by additional air jets to form a non-circular pattern. While the resulting pattern is not necessarily symmetrical, and need not be uniform in density before falling off near its edges, it does not contain any sharp discontinuities. Thus, it is reasonable to add some amount of smoothing influence.

$$
\begin{array}{cccccccccc}
0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.00 & 0.00 & 0.10 & 0.10 & 0.15 & 0.10 & 0.10 & 0.00 & 0.00 & 0.00 \\
0.00 & 0.10 & 0.15 & 0.25 & 0.30 & 0.25 & 0.15 & 0.10 & 0.00 & 0.00 \\
0.00 & 0.10 & 0.25 & 0.30 & 0.33 & 0.30 & 0.25 & 0.10 & 0.00 & 0.00 \\
0.00 & 0.10 & 0.25 & 0.30 & 0.33 & 0.30 & 0.25 & 0.10 & 0.00 & 0.00 \\
0.00 & 0.10 & 0.25 & 0.30 & 0.33 & 0.30 & 0.25 & 0.10 & 0.00 & 0.00 \\
0.00 & 0.10 & 0.25 & 0.30 & 0.33 & 0.30 & 0.25 & 0.10 & 0.00 & 0.00 \\
0.00 & 0.10 & 0.25 & 0.30 & 0.33 & 0.30 & 0.25 & 0.10 & 0.00 & 0.00 \\
0.00 & 0.10 & 0.25 & 0.30 & 0.33 & 0.30 & 0.25 & 0.10 & 0.00 & 0.00 \\
0.00 & 0.10 & 0.25 & 0.30 & 0.33 & 0.30 & 0.25 & 0.10 & 0.00 & 0.00 \\
0.00 & 0.10 & 0.25 & 0.30 & 0.33 & 0.30 & 0.25 & 0.10 & 0.00 & 0.00 \\
0.00 & 0.10 & 0.25 & 0.30 & 0.33 & 0.30 & 0.25 & 0.10 & 0.00 & 0.00 \\
0.00 & 0.10 & 0.25 & 0.30 & 0.33 & 0.30 & 0.25 & 0.10 & 0.00 & 0.00 \\
0.00 & 0.10 & 0.25 & 0.30 & 0.33 & 0.30 & 0.25 & 0.10 & 0.00 & 0.00 \\
0.00 & 0.10 & 0.25 & 0.30 & 0.33 & 0.30 & 0.25 & 0.10 & 0.00 & 0.00 \\
0.00 & 0.10 & 0.25 & 0.30 & 0.33 & 0.30 & 0.25 & 0.10 & 0.00 & 0.00 \\
0.00 & 0.10 & 0.25 & 0.30 & 0.33 & 0.30 & 0.25 & 0.10 & 0.00 & 0.00 \\
0.00 & 0.10 & 0.25 & 0.30 & 0.33 & 0.30 & 0.25 & 0.10 & 0.00 & 0.00 \\
0.00 & 0.10 & 0.25 & 0.30 & 0.33 & 0.30 & 0.25 & 0.10 & 0.00 & 0.00 \\
\end{array}
$$

Table 1. "Perfect" fundamental table showing the rate of accumulation of spray used for testing of the solution process. Values printed are times 1,000.
to the solution algorithm. Zeroth- and first-order regular- 
ization techniques were selected ([21]). The fundamental 
table was treated as a sequence of inscribed rectangular 
shells, each one element wide. The outermost entries (outer 
shell) in the fundamental table are known to be zero, since 
the table size (numbers of rows and columns) was selected 
by the program to assure it, using the measured spray in- 
formation and the user’s selection of table refinement (size of 
each square in the table). Thus, any nonzero entries in the 
solution for the outer shell can be attributed to the effects of 
noise in the measured data or to numerical problems in the 
solution process. In fact, since the outer elements of the ta-
ble enter into fewer of the equations than the center of the 
distribution, the least squares solution tends to 
concentrate error at these perimeter elements. Rather than 
truncating these outer entries to zero and losing the conser-
vation of the total deposition in the table, a large amount of 
zeroth-order regularization was used to drive the outermost 
shell entries to be essentially zero, in effect redistributing 
any non-zero quantities to the interior of the table. Zeroth-
order regularization is obtained by adding a user-selectable 
weight times each outer shell element to the minimization 
equations, rewarding solutions yielding zeroes for that 
shell. It is implemented by forming a matrix $H_p$ of size mn 
by mn (for a fundamental table of size m by n). All ele-
ments are 0 except that diagonal elements $((i-1)*m+j, 
(i-1)*m+j)$ are 1 whenever the (ij) element of the funda-
mental table is on its outer shell. 

Because the spray patterns to be described here tend to have 
roughly elliptical density contours, as does the test distribu-
tion, first-order regularization was applied based on differ-
encing adjacent elements within each interior shell. First-
order regularization adds to the system of equations being 
solved a term for each difference in values of adjacent en-
tries in any interior shell. This influences the least-squares 
minimization toward solutions in which adjacent values in 
a shell are identical, with the degree of the influence deter-
mined by a weighting coefficient. Clearly, this regularization 
must be used cautiously, because in extreme amounts, 
it could hide real differences between adjacent table entries. 
If regularities other than the rectangular shell pattern are 
present, the regularization can be modified to exploit them. 
Let $s = \min((m-1)/2, (n-1)/2)$ represent the number of 
shells present, excluding the center row or column (m and 
n are forced to be odd). The first-order regularization was 
implemented by calculating a sequence of matrices $H_{p,i}$ 
$1 \leq i \leq s$, each of which differences adjacent entries within 
shell $i$ of the fundamental table. ($H_{p,i}$ is made identically 0, 
because only zeroth-order regularization is used for shell 
$I$.) For each element $(ij)_{(i)}, 1 \leq i \leq m, 1 \leq j \leq n$, in shell $i$ of the 
fundamental table, the entry $((i-1)*m+j, (i-1)*m+j)$ in $H_{i}$ is 
set to -1. Then the $((i-1)*m+j, (k-1)*m+p)$ entry is set to 
+1, where $(k,p)$ is the clockwise neighbor of element $(ij)$ in 
the fundamental table. All other entries are 0. (See Figure 
7 for an example.) 

It was desired to be able to control the amount of regular-
ization separately for each concentric ellipse (or rectangu-
lar shell, after discretization). For each $H_{p, i} \ 0 \leq i \leq s$, the 
matrix $H_{i}^{T} H_{i}$ is scaled by a coefficient $\alpha_i$, then added to 
$A^{T} A$ and subjected to least-squares solution for $x$, accord-
ing to:

$$Eq. 3 \quad (A^{T} A + \sum \alpha_i H_{i}^{T} H_{i}) x = A^{T} b$$

The scaling coefficients $\alpha_i$ can be used not only to reward 
smoothness, but also to help enforce the constraint against 
negative entries which should be imposed on the table, as 
described below.

5.2 Barring Negative Table Entries.

Negative entries in the fundamental table represent a non-
realizable physical condition, and must be corrected if present. 
If negative entries occur even after the regularization 
employed above, they are removed by a conservative 
process which distributes the negative entries uniformly 
among the adjacent elements in the same shell and the next 
inner shell. This process is performed iteratively and does 
not allow any element to be driven below zero. At each 
step, each negative element passes up to one third of its val-
ue to each of the three neighbors, subject to the additional 
limitation that it does not pass more than one third of the 
value of that adjacent (receiving) element. The process 
stops when the magnitudes of all negative elements are be-
low a threshold, or if an element in the center column be-
comes negative (an indication of an unusable solution).

6. Alternative Techniques for Representing a Smooth Distribution

There are other, perhaps more direct, techniques for 
achieving a smooth table and ensuring that the edge con-
straints of zero elements are met. One could, for example, 
use an $n$-th order B-spline surface to represent the spray 
thickness distribution, with $n$ knots at each parametric ex-
treme and control points fixed at zero around the boundary. 
One could then select an appropriate number of interior 
knots and control points, and solve for the $(x,y,z)$ values of 
the control points using classical splining methods ([15]). 
However, for speed of evaluation during simulation, the 
authors would still have wanted to sample this distribution 
to create a fundamental table, and therefore preferred the 
direct parameterization of the table.
This file has noise with a maximum value of 0.091 and a ble 2 contains the output of the least-squares solution pro-

42.84% and the mean percent error (non-zero elements) is

fluenced by the noise. The maximum percent emr is

multiplicative noise of standard deviation 2% was created.

additive noise of standard deviation 0.01 and zero-mean

larization (with weight

ent. For example, a noisy measurement file with zero-mean

mean of 7.08% of the corresponding (non-zero) entry. Ta-

maximum error.

However, when noise is added, the results are quite differ-

erywhere and zeroes in the outer shell. Table

6.67%. Further, a fundamental table of this form does not

set to zero. It is clear that the solution has

small negative values, which sometimes persist on the out-

most shell, have been processed out conservatively as de-

cribed above. The superiority of Table 3 to Table 2 is
clear. Entries in Table 3 differ from the original table by

no more than an absolute amount of 0.0255 and no more

than 17.24% of the original (non-zero) entries. Mean error

in the table is 4.24% of the corresponding entry. This value

is well below the mean error in the noisy input measurements.

However, the measures reported for the errors in the funda-

mental table seriously overestimate the errors to be ex-

pected in the use of the SPRAYTOOL simulator, because

any point on a surface accumulates spray along some con-

ected path through the fundamental table, and the errors

on these paths are far less. The process used to solve for

the fundamental table does an excellent job of conserving

the total of the table entries -- in fact, for even the noisiest
data presented, the sum of the table entries was correct to

four significant figures. Thus, Table 4 also includes a com-

parison of the path table entries determined in the presence

of noise with the ideal path table entries. Since entries in

the path table represent integrals of spray accumulation

along paths through the fundamental table, their error is a

better estimate of the overall error in the simulation pro-

cess. For the fundamental table shown in Table 3, the max-

imum error in the path table was 33.50% and the mean

error was only 1.25%. Large maximum errors are still pos-

ible because of short diagonal paths through the corners of

the fundamental table, which contain very small values and

therefore higher percentage noise sensitivity. However,

the mean value is quite low, as expected. Table 4 shows

various error measures found for solutions based on the

sample "perfect" table (Table 1) using various combina-

tions of additive and multiplicative noise. It indicates that

both additive and multiplicative noise can be dealt with

reasonably well by the current parameterization procedure.

8. Conclusions

For many situations, it is reasonable to assume that the level

of error in measuring spray accumulations for the final

product is similar to that for the measurements done for the

parameterization process. Therefore, since the closed-loop

parameterization process appears to work without amplifying

those errors, it appears that SPRAYTOOL is an accu-

rate and useful tool for simulation of robotic spray

application. The initial practical application of SPRAY-

TOOL has borne this out.

Acknowledgment

The authors thank Professors James V. Beck, Department

of Mechanical Engineering, and Richard Hill, Department

of Mathematics, for their helpful suggestions regarding nu-

merical techniques for noise reduction in parameterization

Figure 7. Sample H2 matrix, for a 7 x 5 fundamental ta-

table. H2 is 35 x 35, and each non-zero row ultimately

does differing of an element of the second shell of

the fundamental table from a neighbor in the second

shell.

7. Results of the Refined Parameterization Procedure

The steps in a typical closed-loop parameterization proce-

dure are shown in Tables 1-3. Table 1 is the starting funda-

mental matrix. When no noise is added, the refined

analysis procedure again reproduces it to within 0.014%

maximum error.

However, when noise is added, the results are quite differ-

ent. For example, a noisy measurement file with zero-mean

additive noise of standard deviation 0.01 and zero-mean

multiplicative noise of standard deviation 2% was created.

This file has noise with a maximum value of 0.091 and a mean

of 7.08% of the corresponding (non-zero) entry. Ta-

ble 2 contains the output of the least-squares solution pro-

cedure for this input, with the \( \alpha_i \) smoothing coefficients all

set to zero. It is clear that the solution has been strongly in-

fluenced by the noise. The maximum percent error is

42.84% and the mean percent error (non-zero elements) is

6.67%. Further, a fundamental table of this form does not

satisfy the solution constraints of non-negative entries ev-

everywhere and zeroes in the outer shell. Table 3 represents

the output of the least-squares solution procedure when the

\( \alpha_i \) \( i = 1,4 \) are all set to 0.15, and with zero-th

order regu-

larization (with weight 1.0) of the outermost shell. Any
Table 2. Fundamental table showing the results of the solution process without regularization or processing of negative entries, given noisy input data. Values printed are times 1,000.

<table>
<thead>
<tr>
<th>Std. Dev.</th>
<th>Std. Dev.</th>
<th>Mean %</th>
<th>Max. %</th>
<th>Max.Abs.</th>
<th>Mean %</th>
<th>Max. %</th>
<th>Mean %</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add. of Add.</td>
<td>Noise</td>
<td>Error</td>
<td>Error</td>
<td>Error*</td>
<td>Error</td>
<td>Error</td>
<td>Error</td>
</tr>
<tr>
<td>0.015</td>
<td>0%</td>
<td>6.98%</td>
<td>8.85%</td>
<td>0.0104</td>
<td>1.69%</td>
<td>10.98%</td>
<td>0.52%</td>
</tr>
<tr>
<td>0.0</td>
<td>2.0%</td>
<td>1.59%</td>
<td>23.74%</td>
<td>0.0259</td>
<td>3.87%</td>
<td>23.74%</td>
<td>1.11%</td>
</tr>
<tr>
<td>0.0</td>
<td>5.0%</td>
<td>3.86%</td>
<td>44.31%</td>
<td>0.0632</td>
<td>9.14%</td>
<td>52.26%</td>
<td>2.82%</td>
</tr>
<tr>
<td>0.01</td>
<td>2.0%</td>
<td>7.08%</td>
<td>17.24</td>
<td>0.0255</td>
<td>4.24%</td>
<td>33.51%</td>
<td>1.25%</td>
</tr>
<tr>
<td>0.01</td>
<td>5.0%</td>
<td>10.53%</td>
<td>35.26%</td>
<td>0.0443</td>
<td>8.68%</td>
<td>59.77%</td>
<td>2.70%</td>
</tr>
</tbody>
</table>

Table 3. Fundamental table showing the results of the solution process employing regularization and processing of negative entries, given noisy input data. Values printed are times 1,000.

Table 4. Measures of error in input and output for solutions based on selected combinations of noisy input data. Values printed are times 1,000 to agree with Tables 1-3.
of the spray tables. The authors also thank the SPRAY-TOOL development team, whose members have contributed to both the conceptual design and implementation of the software. They include Guy Allen, Susanne Smith, Paul Haas, Bao Ming, Chang Ki-Yin, Chiu Chin Chuan, Brett Harper, Jonathan Courtney, Martin Correns, Brian Kingsbury, Thomas Lake, and Lovina Bhatnagar. This work was sponsored in part by General Dynamics, Convair Division. The authors thank Dan Popovich, Ed Dozier, and Bill Lindsay of General Dynamics for their significant contributions to the project.

References


