Automated Generation of Weld Path Trajectories

Arlo L. Ames[†] Elaine M. Hinman-Sweeney, PhD[†] John M. Sizemore[*]

[†]Sandia National Laboratories
Dept 15233/MS1010
Albuquerque, New Mexico 87185-1010
USA

[*] Northrop Grumman Ship Systems
1000 Jerry St. Pe` Highway
Pascagoola, Mississippi 39567
USA

Abstract

AUTOGEN is software that automates the planning and compiling of control programs for robotic welding of ship structure. It works by evaluating computer representations of the ship design and the manufacturing plan, identifying and characterizing each weld, constructing the robot motions necessary to accomplish the welds, and determining the correct assignment of process control values. AUTOGEN generates robot control programs automatically, without manual intervention.

Most ship structure assemblies are unique or at best manufactured only a few times. Accordingly, the high cost inherent in all previous methods of preparing complex control programs has made robot welding of ship structures economically unattractive to the United States shipbuilding industry. AUTOGEN eliminates the cost of creating robot control programs. With programming costs eliminated, capitalization of robots to weld ship structure becomes economically viable. Robot welding of ship structures will result in reduced ship costs, uniform product quality and enhanced worker safety

Sandia National Laboratories and Northrop Grumman Ship Systems worked with the National Shipbuilding Research Program to develop a means of automated path and process generation for robotic welding. This effort resulted in the AUTOGEN program, which has successfully demonstrated automated path generation and robot control. Although the current implementation of AUTOGEN is optimized for welding applications, the path and process planning capability has applicability to a number of industrial applications, including painting, riveting, and adhesive delivery.

1 Introduction

AUTOGEN plans robotic welding using geometric part models and manufacturing intent. The required inputs for weld path generation are ship design information, robot data, and manufacturing information to determine what needs to be welded, what weld joint types and sizes to use, and the manufacturing intent for the workpiece. AUTOGEN attempts to capture a welder's knowledge of weld process parameters and best practices. AUTOGEN then directly provides paths and processes to a robot to enable completely automated welding. This is a significant step beyond manually teaching the robot or employing offline "macro-programming," as it uses data already available to automatically program the robot. Current teaching and macro-programming techniques result in an engineer taking nearly the same amount of time to program robot commands as it takes a welder to perform the welds. This is not cost-effective, especially for low-volume or one-off parts. AUTOGEN produces the entire weld path and process in a fraction of the time taken in offline macro-programming methods.

Weld planning proceeds according to local process conditions computed from the geometric contents of the work piece design information and the manufacturing plan. AUTOGEN completes construction of robot control programs by applying computed local conditions to weld process response surfaces and the kinematic and control language specifications of the available robot. All of the core issues of joint design, torch motion, and weld process and placement are directly coupled to each other through geometric reasoning.

Because of internal awareness of geometric context, AUTOGEN can plan welds on a work piece of arbitrary complexity presented before the robot in arbitrary orientations. By its definition, AUTOGEN can handle compound curved surfaces and compute collision-free and singularity-free trajectories for the torch manipulator. The architecture of the software is intentionally open to modification of most of the performance rules. AUTOGEN is independent of the process accomplished, and any particular robot or process hardware. This provides opportunity for a large domain of potential applications in a variety of industries.

The next sections will describe the elements that AUTOGEN brings together in producing automated process and path plans.

2 Ship Structure

AUTOGEN uses solid model representation of the objects being welded, along with any obstacles within the manufacturing space. AUTOGEN's internal data representation is ACIS (R), from Spatial (www.spatial.com).

The geometric information can be annotated with attributes describing part names, materials and weld definition. Name and material attributes are attached to BODY objects, while weld definitions are attached to EDGEs, with references to FACEs that are adjacent at each weld. Weld attributes additionally include weld symbol information.

Manufacturing pose of ship structure is provided by simple Laydown directives, which define transformation matrices to be applied to all ship geometry. Portions of ship geometry, obstacles, and welds can be selected for presence or absence from the current welding context, allowing planning over portions of large models. Leave loose directives are also available for weld attributes.

We use both “exact” (composed of higher-order curve and surface representations) and “faceted” (composed purely of lines and planes) geometric representations, depending on context. For problem definition purposes, we prefer the edge and surface definitions to be exact, as the exact representations provide means of interacting with the model in the same terms as it was designed in, preserving smoothness and level of continuity intended by the designer. Faceted representations are used in AUTOGEN for efficiency in distance computations, where speed is more important than exactness.

3 Weld Definition

AUTOGEN is process centered, based on core observations that welding occurs and responds in relation to the local physical geometry of the work piece and according to the actual direction of gravity. Corollary to these observations, workpiece edges locally dominate welding accomplishment. These edges conveniently serve as the basis for erecting local coordinate frames for process planning.

Within AUTOGEN the geometry of a weld is modeled by combining a geometric edge and one or more model faces. The edge provides a nominal location for the weld as well as a tangent direction along the weld, and the face provides a direction suggesting a normal to the weld. Welds can be specified to AUTOGEN within a data file, as an attribute attached to CAD geometry, or by user directive issued during run time.

The planning algorithm traverses every edge of each weld in an assembly. Typical CAD edge representations are sufficient for this task, as they can provide a constant velocity parameterization that can be queried to provide position and a variety of differential geometric properties (e.g. tangent, curvature, etc). Edges are traversed by sampling to produce an ordered list of discrete waypoints, which are located sufficiently close together to provide continuity in results. At their core, robot controllers provide joint angle control at discrete time intervals, so the sampling approach is warranted.

In order to provide orientation about the edge for placing the torch, we require information about the locally adjacent geometry involved. CAD surface representations are sufficient to define surface normal information at arbitrary locations along the weld. The normalized sum of two surface normals is sufficient to provide a local weld bisector that is both locally optimally far from immediate obstacles, and can be used in concert with the edge tangent to define a local coordinate system to use for orienting the torch.

The weld attribute attached to any edge includes information about the adjacent faces. Note that any design definition of the weld (e.g. diameter, number of passes, etc) can also be attached to the attribute, either directly as a part of the attribute, or indirectly through pointer reference.

Since welds are describable in terms of the geometry of an edge, welds are represented as attributes attached to edges. The required edge geometry may either be an edge of a part or a disconnected edge, as would be the case in representing a cladding operation.

Note that we only require the ability to determine a sequence of locations and orientations to define welds. A variety of possibilities exist, including directly providing the position and orientation information. We choose to label edges as a valuable convenience. Any extension that would provide the requisite waypoint information (e.g. specifying a random distribution of waypoints on a surface) can be added without any adverse effects.

Welds can be imported with CAD geometry as ACIS attributes, manually defined within AUTOGEN, or automatically recognized. Any combination of the above approaches is also valid.

Each weld has a number of attributes describing its classifications, including position (e.g. flat, horizontal, vertical, overhead), juxtaposition (e.g. tee, lap), type (groove, fillet). These attributes may be imported, manually assigned, or automatically recognized.

4 Process Information

Each weld has an attached weld symbol, which contains the bulk of the weld symbol information defined in [6]. The weld symbol represents information about preparation geometry (where the preparation geometry is implicitly defined), weld procedure, multipass stacking definition (for multi-pass welds), intermittent weld specification (for intermittent welds). The weld procedure contains all the welding parameters (power, feeds, speeds, etc) to permit welding to occur. Process information can be manually attached to welds, imported from external data sources, or automatically looked up in a library of weld procedures, as a function of the weld's position, juxtaposition, type, materials, and material thicknesses.

5 Robots

Robots are modeled as structured objects containing link geometries along with forward and inverse kinematics routines. The geometries of individual links are useful for both visualization and for computing collisions; the kinematics routines provide for correct robot motion.

Forward kinematics are computed directly. Inverse kinematic routines are provided in two forms: a closed form solution and an iterative solution. The iterative solver is available for any robot for which forward kinematics are implemented, but only provides a "closest" solution to a previously solved-for position. The closed form solution provides inverse kinematics solutions for all possible robot configurations, and so is much more useful in planning robot trajectories. Closed form solutions must be custom coded for each new robot provided, but the effort pays off in faster solutions and multiple configurations. Robot dynamics are provided by direct solution. Much of AUTOGEN's basic robot representation is provided by ROBOOP [3]; alternatives that avoid use of D-H parameterization are being explored.

Robot trajectories are represented as lists of waypoints, termed Robot Process States. Each waypoint is aware of geometric location, torch orientation, and robot orientation in joint terms. A trajectory is executed by simply visiting each waypoint.

The search algorithm must produce weld motions that are smooth, collision free, and well within the reach of the robot, avoiding singularities and joint limits. Additionally, the search must produce paths that are as process-optimal as possible in terms of torch pose relative to weld geometry. It is further desired that the planning algorithm be very fast when the planning problem is simple, but able to find solutions for difficult problems when they arise.

Weld motion planning is accomplished by searching in lead/lag, bias, and rotation space, these being variables that directly define torch pose. The lead/lag and bias terms are related to the optimal process, while the rotation variable has no bearing on the process, as the torch is locally rotationally symmetric about its tip (this would not be the case if our application were stapling).

The inverse kinematics solution is coupled with distance checks to obstacles. Should a collision (or near miss) occur for a given inverse kinematics solution, the solution is disallowed (and can be used to drive iterative inverse kinematics solvers).

The search algorithm works in three parts: a line search, a thorough search and a smoothing pass. At a given waypoint, if information exists about a previous point exists, we use it to initiate a line search that seeks toward the process optimal. At each step in the line search, we search rotation, as it is process independent and may provide a solution. If the line search finds a solution, it is remembered and we move forward. Failure of the line search invokes an exhaustive solution, starting at the previous solution. Exhaustive solution has not proven to be terribly expensive, given that the line search frequently finds most of the solutions, and our search space is only three-dimensional. After a solution is known (or known to not exist) for each waypoint along the path, the path is searched for discontinuities. Smoothing is attempted across the discontinuities, using an energy-minimizing smoother. Should smoothing fail, the weld line is broken in two pieces, indicating that some obstacle prevents the weld from being accomplished as a single motion.

Since multiple robot configurations (i.e. elbow up, elbow down) might be possible for any given reach, all robot configurations are tested until one is found that minimizes the number of breaks in the weld.

Free space motions between the end of one weld and the start of the next are computed using an RRT algorithm [8]. The algorithm computes random trajectories in joint space and attempts to connect the trajectories to seek a feasible path. The path is then pruned to reduce its overall length.

A fast distance function from the C-Space Toolkit [4] provides fundamental functionality for computing collision-free paths. Both weld motions and free-space motions compute distance to obstacles along the path, and search as necessary to avoid collision and maintain a reasonable distance between torch/robot and obstacles, including both the ship parts being welded, as well as any other geometric obstacles. The distance function relies on very simple geometry representation, providing the possibility of creating obstacle models from sensor data or simple operator input.

6 Implementation

The algorithms and data structures enumerated above have been implemented in C++. Figure 1 is an example collaboration graph indicating some of the complexity of interaction among the various kinds of objects required to support planning.

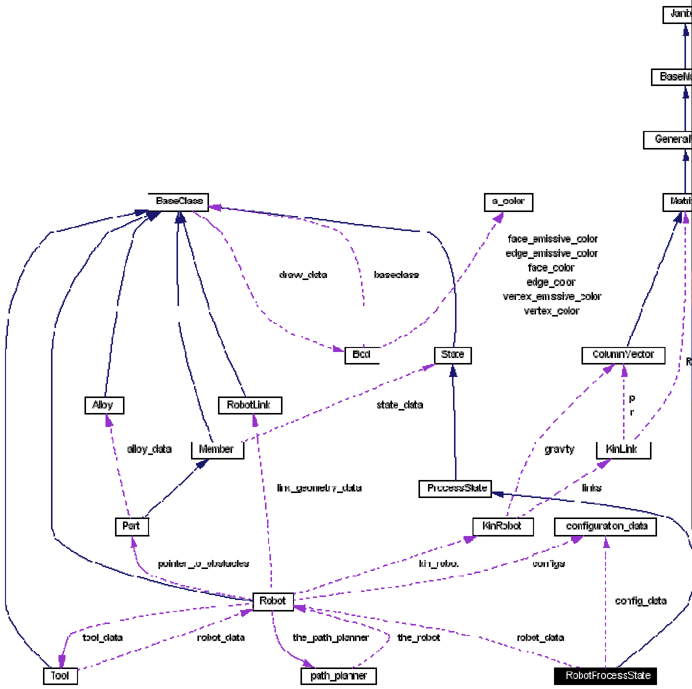

Figure 1: Collaboration Graph of Class Robot Process State

7 Results

Path generation was successfully demonstrated in December, 2001. The algorithms were tested on a variety of representative part geometries, and demonstrated with a physical robot against a plastic prototype of curved ship structure, as shown in Figure 2.

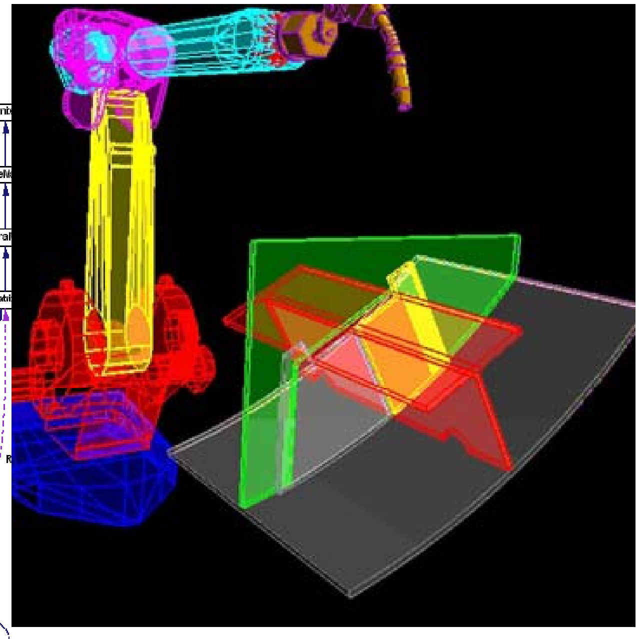

Figure 2: Graphic Representation of Demonstration Robot and Task

Fully automatic planning and execution of welding processes was demonstrated in December, 2003, on the ABB IRB1400 located at the SWRL facility in New Orleans.

The planning algorithms present in AUTOGEN have so far proven to be quite efficient. The path generation for the December 2001 demonstration has been computed, using 1.5 GHz processing hardware, in roughly 2 minutes. Execution of the welds for that assembly requires 20 minutes, yielding off-line planning that is significantly faster than required to keep ahead of production. Attempts to produce an equivalent plan using previous offline programming paradigms have proven to take between 30 minutes and 4 hours (depending on method and operator sophistication), costing more than the price of performing the weld by hand, and with more expensive labor.

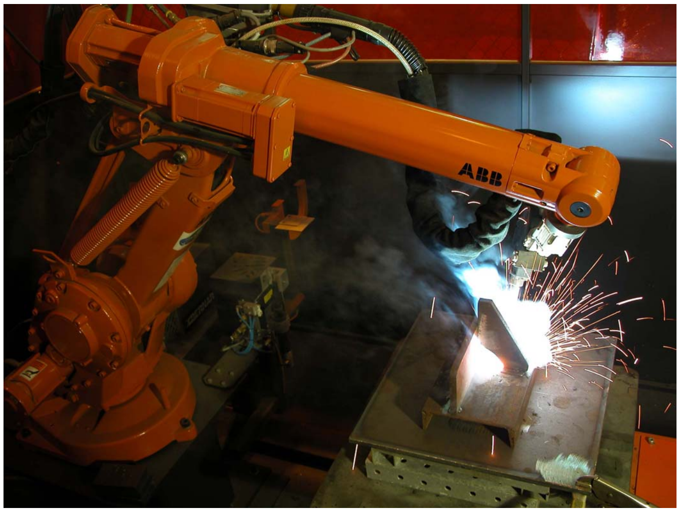

Figure 3: Robot executing welds at the SWRL laboratory

Current attempts to exercise the software by our welding engineer have met with good success; the software is, in its development state, powerful and easy to learn.

During early tests, each plan was meticulously simulated to verify the quality of weld paths before attempting to execute them on the robot. As confidence improved, use of the simulation capability declined, to the point where verification of programs occurred on-robot, with the torch on. While the planning algorithms continue to produce occasional poor results, the results are becoming sufficient to support production work, with high confidence in the quality and consistency of results.

The quality of welds produced with the automatically constructed programs was proportional to the effort expended to determine weld parameters. Roughly two days were spent experimenting with weld parameters (e.g. gas flow, wire speed, weave parameters) for flat, horizontal, and vertical welds. All of the welds executed automatically were of roughly equal quality to the test welds produced during parameter selection.

The parts for the 2003 demonstration were obtained by searching through scrap for reasonable flame-cut pieces, roughly 1 ft or .3 m on a side. Scrap parts were chosen both for economy and to ensure that typical part accuracy was represented.

Generally, the planning algorithms perform quite well. Long planning times occasionally occur in cases where the part is large relative to the reach of the robot, suggesting a move towards robots with greater reach, especially those with more degrees of freedom.

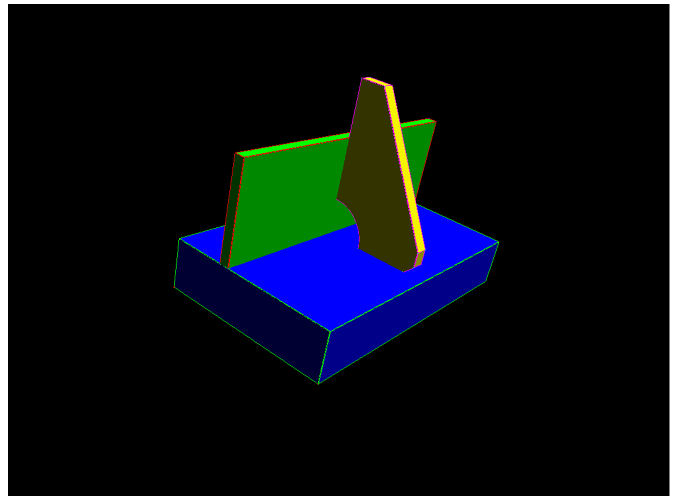

Figure 4: A test part with a snipe. Weld planning successfully accommodated access limitations imposed by the presence of the snipe.

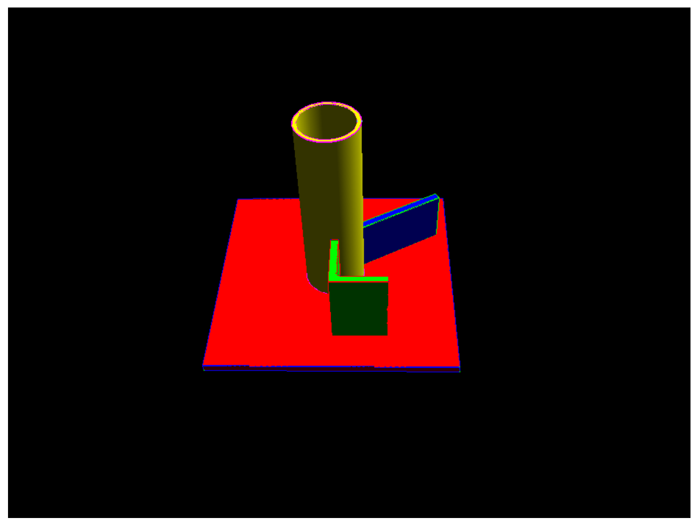

Figure 5: A test part containing round pipe. Corners, vertical and horizontal welds were correctly executed. Round pipe is rarely round; through-the-arc sensing mitigated roundness problems, but caused drift away from weld on one vertical weld.

8 Ongoing Development

Current development work focuses on expanding AUTOGEN's domain of discourse, both for correctness and to increase it's domain of applicability.

Correctness of results is being addressed by work to improve the smoothness of paths, improved singularity avoidance, improved path quality through corners, support for skew welds, better planning for weave and through-the-arc sensing for non-planar workpieces, and sensor strategy planning.

Increased domain of applicability is being addressed by developing planning algorithms that support robots with redundant degrees of freedom (greater than 6 DOF), gantries, part positioners, multiple non-interfering robots, mobile platforms, skew welds and implicit weld preparations.

Current plans call for testing using far larger test workpieces (roughly 10 ft or 3 m on a side). The test pieces will be selected from current production parts (or sufficiently close to satisfy shipyards) from at least two shipyards. Workpiece quality will be checked using typical production tests.

9 Conclusions

AUTOGEN provides an automatic means of producing control programs, much faster and more correctly than previous attempts at offline programming. The quality of welds produced is consistent, and commensurate with the amount of effort expended to determine good welding parameters. The approach is scalable to higher levels of completeness, both to improve correctness of results and to address a wider range of welding processes.

Acknowledgment

This work was performed at Sandia National Laboratories, a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

References

[1] Ames, A., Hensinger, D., Sizemore, J., "AUTOmated GENeration of Control Programs for Robotic Welding of Ship Structure (AUTOGEN)," *SNAME 2001 Ship Production Symposium*, June 13 - 15, 2001.

[2] Sizemore, J., "Technical Note 447", August 2002.

[3] Gourdeau, Richard, "ROBOOP – A Robotics Oriented Package in C++", http://www.cours.polymtl.ca/roboop.

[4] Xavier, Patrick, "A Generic Algorithm for Constructing Hierarchical Representations of Geometric Objects," *Proceedings 1996 IEEE Int'l Conf. On Robotics and Automation*, 1996.

[5] Ames, A, Hinman-Sweeney, E., Sizemore, J., "Automated Generation of Weld Path Trajectories", *Journal of Ship Production*, Volume 9, Number 3, August 2003.

[6] ANSI/AWS 2.4-98, Standard Symbols for Welding, Brazing, and Nondestructive Examination, American Welding Society, Miami, Florida, 1998.

[7] ANSI/AWS A3.0-94, Standard Welding Terms and Definitions, American Welding Society, Miami, Florida, 1994.

[8] LaValle, Steven M., "Rapidly-Exploring Random Trees: A New Tool for Path Planning", Technical Report No. 98-11, Dept. of Computer Science, Iowa State University, Oct. 1998.