Reprinted from February 1988, Vol. 110, Journal of Engineering for Industry

A Hierarchically Structured Knowledge-Based System for Welding Automation and Control

D. R. Thompson[1]

A. Ray
Mechanical Engineering Department
Mem. ASME

S. Kumara
Industrial Engineering Department

The Pennsylvania State University,
University Park, PA 16802

The paper presents the concept and prototype development of a hierarchically structured knowledge-based system (KBS) for coordinated control of a welding robot and a positioning table. The KBS is designed for continuous seam welding in an autonomous manufacturing environment. The structure of the KBS allows updating and installation of individual function and database modules such as those used for determining welding parameters and identifying constraints and parameters for robot and table controllers. The KBS is designed to plan the feasible table configurations and robot trajectories as well as to coordinate the movements of the robot and positioning table. The KBS has a modular structure which can accommodate multiple configurations of different types of robots and positioning tables within a flexible manufacturing cell. Initial testing has demonstrated the feasibility of integrated heuristic and algorithmic procedures for real-time applications. A fully implemented version of this KBS will enable a user to perform high quality welds in a relatively short period of time with (possibly) incomplete knowledge of the welding cell equipment. The KBS software is written in the C language and has been tested for real-time operations within a simulated welding cell environment.

1 Introduction

The process of continuous seam welding requires a considerable amount of experience and skill to perform manually. Often it takes several hours of work to complete a single work piece if it has many seams. To simplify this task robots have been employed to follow taught weld paths repeatedly. However, until recently, welding equipment has been considered as an ancillary device of the welding robot assembly. Often the integration of this equipment with commercially available robots proved difficult. This problem lead to the introduction of Integrated Robotic Welding Systems such as the Puma Arc Welding System (PAWS) [1]. Integrated systems are designed with the welding gear as part of the robot. Control languages, such as VAL by Unimation and RAIL by Automatix, are used with these systems to provide a more intuitive means of programming through the use of English-like commands. Integrated systems do improve the interactions between the robot and the welding equipment. However, an experienced user is still required to identify the welding parameters and teach the "best" weld path for the robot to follow.

In a flexible manufacturing environment there may be multiple configurations of welding robots and positioning tables which could be used to perform a given job. The dynamic environment of a factory may demand the use of different configurations from time to time. It is rather unlikely that a human operator will be proficient in the use of all the different robots, positioning tables, and their configurations. Under these circumstances if appropriate assistance is not available from an expert, the job is likely to be delayed, added with the risks of degraded welding quality, equipment damage, and increased down-time. To increase the productivity of the operators and to enhance quality, a knowledge-based system (KBS) needs to be developed that would determine, with little or no human intervention, the feasible weld path orientations, i.e., the table position and the corresponding robot trajectory, and to control the interactions between them [2,3,4]. The KBS should also be able to determine the appropriate weld parameters for a given job based on its description.

The concept of using Artificial Intelligence (AI) for robotic control stems from the work done in the 1970s [5,6]. Winograd's work [5] on the Blocks World concept concentrated on a natural language interface to AI problems. However, control was not the main feature of his work. Grossman and Taylor [6] proposed the interactive use of generic object models for path planning of robot manipulators in three dimensions. A significant amount of research work from an analytical point of view has been published in the areas of robot trajectory planning, adaptive control, and machine vision [7–11]. Use of AI techniques for robot control has been concerned with the development of programming languages (such as AUTOPASS [12] and AML [13]), and generic, high level nonspecific expert systems for path planning (such as STRIPS [14] and ROPES [15]).

The application of knowledge-based systems to control of robots in welding operations has been rather limited [3]. Kerth [16] introduced the concept of knowledge-based control for regulating welding process parameters in real time. Inputs on

[1]Currently with E. I. Du Pont De Nemours & Company, Wilmington, DE 19898.

Contributed by the Production Engineering Division for publication in the JOURNAL OF ENGINEERING FOR INDUSTRY. Manuscript received at ASME Headquarters, October 15, 1987.

seam profile were obtained from a laser vision system to update the welding process parameters while following a taught path. However, Kerth's work did not address the interactions between the robot and the part position prior to the welding process. Apparently no KBS has been developed to deal with the coordination of the robot and the positioning table when performing the path planning function.

To address this problem both the factual and heuristic information for all robots and positioning tables that could be used for the job should be assimilated into a KBS. This system should then be able to coordinate the physical interaction between a robot and a table on the basis of an expert's knowledge. The following methodology is adopted to accomplish this goal.

- Identify the robot and the positioning table to be used.
- Obtain all the factual information pertaining to the welding robot and the positioning table such as robot geometry and table motion specifications.
- Determine appropriate welding parameters from the information about the part being welded, including the weld path and orientation.
- Consult analytical rules to determine if the specified path is feasible with the current robot geometry and table orientation.
- Use a set of experience-based rules to determine if the weld quality can be improved by reorienting the table position or if an originally infeasible path can be made feasible by altering the table orientation.

Once the final table orientation and the corresponding robot trajectory have been determined, the information is converted into commands such as position, velocity, or motor control voltages. These commands must be compatible with the controllers in the welding cell.

From the proposed methodology it can be seen that there are many different types of knowledge and reasoning processes required for the complete coordination of a robot and a positioning table for a given welding process. The knowledge base builds upon the facts like robot geometry and table orientation and comprises both rule-based and algorithmic procedures. Rules could be based on analytical principles as well as heuristics. For example, simple analytical rules are used to assess the robot's limitations, and rather complex heuristics for determining the best orientation for the positioning table. The knowledge-based system also includes algorithmic procedures such as the inverse kinematics for the computation of joint angular movements on the basis of endpoint displacement of the robot.

To date, flexible welding systems have been largely a collection of products from different manufacturers put together to form a welding cell with little preplanned coordination between various cell components [3,4]. An integrated welding system offers well-defined interactions between the robot and its welding accessories but are not coordinated with the positioning table. The research work reported in this paper demonstrates the capabilities of a prototypical knowledge-based system (KBS) to provide for the real-time coordinated control of continuous seam welding robots and positioning tables along with weld parameter determination for a selected class of jobs. (The development of an extensive knowledge base and the associated search procedure for determination of weld parameters on the basis of the job description is not the focus of this research.) The KBS has a modular structure which allows expansion of its capabilities by incorporating larger, more powerful modules in place of current ones without affecting the operation of the rest of the system. The KBS is designed to operate in a decision-making and supervisory role within the control structure of a work cell. Coordinated and time-sequenced commands are issued by the KBS to the individual robot and table controllers.

The objectives of the paper are (*1*) to provide coordinated control for a continuous seam welding robot and a welding positioning table to perform high quality welds in a relatively short period of time with (possibly) incomplete knowledge of the welding cell, and (*2*) to demonstrate a real-time implementation of the knowledge base.

The paper is organized in six sections starting with the Introduction. Section 2 provides a brief outline of knowledge-based systems and their relevance to welding processes. The structure of the reported KBS is presented in Section 3. Section 4 describes how the KBS is implemented in a microcomputer environment. Section 5 explains the details of the experimental facility and some of the experimental results. Section 6 addresses the summary of the reported research, its conclusions, and the potential for future research.

2 Features of the Knowledge-Based System for Seam Welding

The literature in Artifical Intelligence (AI) abounds with information concerning knowledge-based systems. Several prototype KBS have been built in the areas of manufacturing, robotics, and fault diagnosis [17]. For example, trajectory generation and path finding in the presence of obstacles are important research problems that have been formulated and investigated in many ways. Programs that autonomously solve collision avoidance problems and generate paths for the robot require a very thorough understanding of the specific problem at an abstract level to obtain the validation of the algorithm and at the computation coding level to generate practical and efficient programs based on the above-formulated algorithms [18,19].

The essence of KBS has not been captured in the area of seam welding in an autonomous manufacturing environment. The task of continuous seam welding requires complex computations, and the solution to the robot path planning problem may not be unique, i.e., a number of possible table positions and the corresponding robot paths could be viable. Furthermore the adopted solution depends on the expertise of the problem solver.

The design of an intelligent robot for continuous seam welding is particularly difficult because of the complexity of the representation of the robot's working environment. Apparently the most important aspect of the automatic path planning problem is a detailed analysis of the 3-D or 2-D environment based on geometric models. This design of an intelligent robot requires the integration of the knowledge of information processing and rule-oriented programming. Like other robots, the software programming of welding robots must support sensor-based real-time control features and is broadly classified in two categories [20]: (*1*) explicit programming where the user states every instruction that the robot must follow, and (*2*) world modelling where the robot must make its own decisions about the path to be followed on the basis of the knowledge of the work space. World modelling is specially suited for real-time control and consists of the following three major tasks [21].

- Acquire the world description.
- Map the world description into the search space.
- Search for a path via heuristic or algorithmic strategies.

We have emphasized the world modelling approach in the development of a KBS for continuous seam welding processes which are characterized as follows:

- The entire process, including selecting the welding parameters, determining the table positions, finding the robot trajectories, and issuing appropriate command sequences to the table and robot controllers in real-time, is difficult to formulate and solve analytically.
- The knowledge related to welding technology is vast and needs to be used selectively.
- Problems such as determining table positions and finding

robot trajectories may have a number of alternative solutions and is computationally explosive if only an analytical solution strategy is sought.

These aspects make the problem of seam welding path control a prime candidate for the KBS approach.

3 Structure of the Knowledge-Based System

The concepts involved for coordinated control of the welding processes are divided into four areas: *weld planning, movement planning, robot control, and positioning table control*. Each of these areas requires its own type of knowledge in diverse disciplines. Figure 1 is a representation of the basic KBS structure along with the above four types of knowledge required.

The actual program for controlling the welding cell is structured in two levels of hierarchy, as shown in Fig. 2, with the third level being the databases required to drive the working modules. There are three databases used in the program: (*1*) the parametric information pertinent to the types of seam welds and certain feasible welding paths, (*2*) device-oriented information for the given types of robot manipulators, and (*3*) device-oriented information for the given types of positioning tables. Each of these databases are designed based on the principles of relational schema.

The top level in Fig. 2 is the supervisory program which serves as the user interface for receiving the necessary information as well as for transmitting the status reports. All of the main control and setup menus are located at this level. It is also responsible for processng the information in a format that is acceptable to the intermediate level which comprises the aforementioned four modules, each of which has a specific function. These functions are described below.

Weld Planning. The weld planning module receives the user-supplied information on weld type, material, thickness, etc. On this basis it determines the appropriate weld parameters by consulting the welding database. The selected parameters are displayed to the user with an option to modify.

Conceptually the weld planner has a table-look-up structure; however, the amount of knowledge associated with the welding domain is vast. The prime consideration is to design the database with an efficient search technique that will function in real time. In the prototypical knowledge-base, the information is contained in a relational schema to be captured in the form of rules. The query retrieval schema is defined on the basis of the logical selection operations where appropriate tuples are selected with the help of relational algebraic operators.

New knowledge could be incorporated into the current database with the aid of an update utility. The welding database used in the reported research is limited to certain selected weld types. The modular structure of the weld planner allows for the modification and augmentation of the current database into an extensive knowledge base for the determination of weld parameters on the basis of the job description.

Movement Planning. The movement planning module makes use of a combination of rules and algorithms to determine the feasible orientations for the positioning table. Information specific to the robot and the positioning table being used is retrieved from their respective databases. A weld path is defined on the basis of this information and the rules imbedded in the knowledge base. This constitutes the first pass of the movement planner. All the restrictions of the robot and table geometry are taken into account at this stage. If the user wishes, a second phase of the movement planner can be invoked which will decide if the part can be re-oriented to produce alternative paths. The path selection is accomplished using several process-oriented rules that take into account weld quality, time requirement for job completion, preferred robot orientations, table configuration, etc. Upon determination of the weld path, the details of planned, time-sequenced movements for the robot arm and the positioning table are calculated and stored for repeated execution of batch jobs.

Robot Control. The robot control module receives the weld path information from the movement planner and processes it for the specific robot in operation. The global positioning information is converted into appropriate motor control commands by using an inverse kinematic relationship that solves for individual joint angles of the robot manipulator. The commands may represent position or velocity information, motor control voltages, or at a higher level, weave and interpolation commands. It is important to note that two robot controllers may have different capabilities and command protocols even though they may be kinematically similar.

Table Control. The robot control module receives the table positioning information from the movement planner and processes it in the form of command sequences for the chosen table. The transformation algorithms that are necessary to convert a weld path from one table orientation to another is located in this module.

Structuring the KBS into functional modules and defining the interactions between them simplify the implementation process. However, as the system must operate in real-time, there is a trade-off between modularity and program execution time.

In the following pages, we summarize the functions of the system program outlined in Fig. 2.

Knowledge of the seam welding process is imbedded in the form of rules in the supervisory program which functions as a meta-level controller. Depending on the user-supplied information, the supervisory program invokes the query retrieval schema in the weld planner. Having gathered appropriate in-

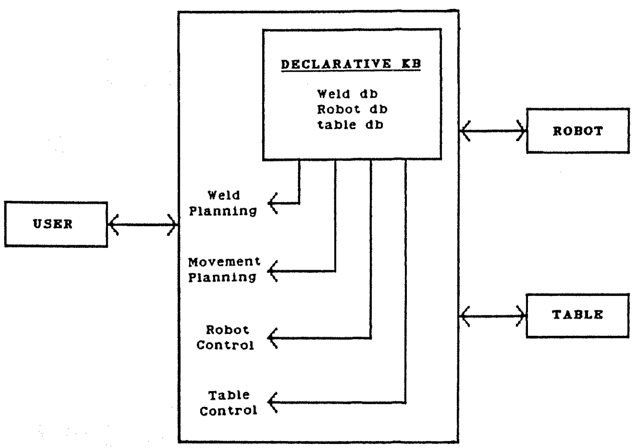

Fig. 1 Knowledge-base structure

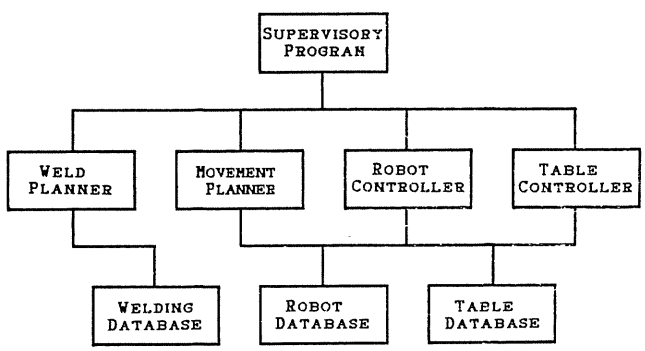

Fig. 2 Control program hierarchy

formation for a given seam, the movement planner is invoked and it interacts with the robot controller and the table controller. Both these controllers have generic rules and algorithms, which interface with their respective databases. The meta-rules in the supervisory program feed the retrieved information from the Robot and Table database to the movement planner which generates the path. Although the four functions at the intermediate level are modular, there are ample interactions between them via the supervisory program.

4 Implementation of the Knowledge-Based System

The knowledge base, in its current configuration, is partially rule-based and relational-schema-oriented. To implement this KBS on microcomputers, such as a personal computer, a language is needed that can handle both conventional and symbolic programming problems and be able to interface with peripheral equipment using device specific functions. Conventional programming languages like FORTRAN tend to rely on algorithms to provide their overall structure, whereas AI languages such as PROLOG [22] and LISP [23] are more flexible in terms of the language syntax. They rely on heuristics for the program construction, and have the capability of directly associating information with symbols. However, the commercially available versions of AI languages that are suited for microcomputers have several disadvantages in view of real-time control applications, such as:

- Less efficient (relative to languages like C) for numerical computations such as those required in robot inverse kinematics.
- Interpretive as opposed to compiled. (Compilable versions of LISP are recently available but they require non-standard operating systems).

The requirements for the KBS include a structured and easily accessible database, and several subprogram modules that represent sets of domain-specific heuristics, rules for drawing inferences, analytical models of table and robot kinematics, algorithms for position and velocity control, and data formatting for information display. All these functions of the KBS are highly interactive, and have to be carried out in real time. For this reason the C programming language was selected [24]. It enjoys the benefits of both high-level and low-level languages, promotes a modular programming style, facilitates program development, and has a large assortment of practical control flow statements and data types. It also consists of powerful memory addressing data structures and operators which are closely tied to the device hardware. C is very powerful in its string manipulation ability and list handling features as well. These aspects, along with its compact compiled size and speed of execution, make C a very versatile language that is suited for the KBS of a continuous seam welding process.

Preassigned data structures are employed to store the information that is generated for each programmed weld path. One structure is dedicated to each active weld path for a given job. The weld parameters are dynamically assigned to their appropriate structure member by the weld planner module. As an example of the data structure used in the KBS, the WELD structure in the C format is illustrated in Fig. 3.

The "path" member of the structure is matrix containing up to a maximum of 50 points for a five degree-of-freedom robot. The "step" member contains the necessary motor control commands for the specific robot chosen that correspond with the above path points. The "phi" array contains the two table angles for the chosen orientation. The remainder of the structure contains the necessary weld parameters. This data structure is available to each module so that the required information for each weld path can be easily accessed and updated.

The user-supplied information on welding joint type, material, thickness, etc., are passed onto the weld planner module which provides the interface between the welding database and the user. The development of a welding database is an evolutionary process and is beyond the scope of the research work reported in this paper. For illustrative purposes a conceptual schema for a welding database is given in Fig. 4. This schema is then translated into a relational database and implemented. The welding database can be expanded to accommodate a wide class of welding jobs.

The weld planning module operates by prompting the user to enter each attribute one at a time. At each step a list of all possible responses is displayed. For example the first prompt asks for the MATERIAL to be welded. A list of all the materials in the database is displayed and the user is allowed to choose one of them for the current application. This process is repeated for each attribute. Once all the questions have been answered, the weld parameters are uniquely defined in the domain of the current database. The appropriate parameters corresponding to each query are found by following the associated file pointer and are displayed on the CRT screen before being placed in the current WELD structure (see Fig. 3). These parameters may then be examined and altered by the user if desired. Also the user has the option of entering new data into the database to keep it up to date.

The first goal of the movement planner is to determine whether the desired weld path is feasible with the current part orientation and work cell geometry. If it is not possible to achieve this objective, the next goal is to find out if the part can be moved, via the positioning table, to generate a feasible weld path. The final goal for all paths is to find out the best orientation for the job to achieve the highest quality weld without exceeding a specified time limit. Finding the best orientation depends on the extent of available knowledge and thus may not always be possible. In that case a feasible path could be an acceptable solution. The inference mechanism handles the problem by processing device specific rules for robot motion, which are available from the robot controller module. Inferences are drawn from the application of each of these rules along with weld quality heuristics. These heuristics are based on the orientation of the weld path which is defined as one of the four types: *flat*, *horizontal*, *vertical*, *and*

```
struct WELD{
   double     path[50][5];       /*   global X,Y,Z,PITCH,ROLL      */
   int        step[50][7];       /*   joint control values         */
   double     phi[2];            /*   table angles                 */
   int        steps;             /*   No. of steps in path         */
   int        voltage;           /*   welding voltage              */
   int        current;           /*   welding current              */
   int        speed;             /*   torch endpoint speed         */
   int        feed;              /*   wire feed rate               */
                :
                :
};
```

Fig. 3 Data structure for welding parameters and trajectory

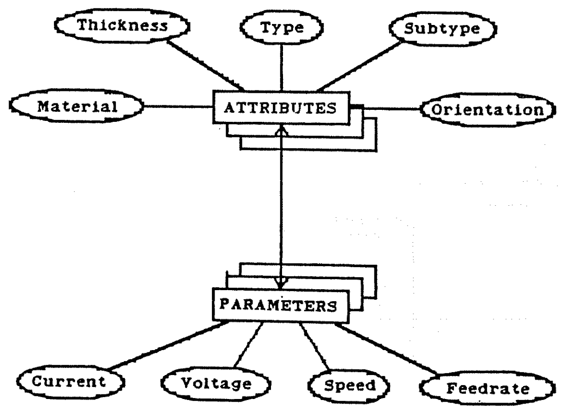

Fig. 4 Schema for weld planner database

overhead. The preferred orientation is *flat* which is the desired goal of the movement planner. Each of the other types have a decreasing preference and are not used unless necessary. The user is constantly informed of the current status of the inference mechanism's goal state. Each of these states can be overridden by the user if desired.

Both the robot controller and table controller are device specific. It would be inappropriate to adopt a generalized knowledge base for manipulating the robot and table. This necessitates retrieval of the appropriate information related to the specific type of robot and table from the respective controller modules. The next step for the inference mechanism is to plan a sequence of events for the robot and table as delineated below.

Once the final position of the table has been determined the resulting robot trajectory information is stored in the current WELD data structure. The robot and table control modules are then used to convert the weld path information into the appropriate device commands. This information is then entered into the completed WELD data structure and is stored on the disk. Each weld path is programmed and stored in a weld structure database (see Fig. 4) in this way. These generated paths may then be used separately or in conjunction to complete the welding operation.

5 Experimental Facility and Results

As a proof-of-concept, a work cell consisting of a five degree-or-freedom robot (MICROBAT TEACHMOVER) and a positioning table has been developed as shown in Fig. 5. The robot has an anthropomorphic geometry similar to many welding robots. The positioning table has a rotating surface and a rotation toward the robot, which simulates two degree-of-freedom welding tables. Both the robot and the table have their own controllers that are interfaced with a personal computer (PC) via RS232-C ports. The PC hosts the software for the knowledge-based system and acts as the supervisory controller and decision maker for the cell environment. The following example will demonstrate how the system operates in the manufacturing environment.

When a new job enters the system the first available welding cell is assigned to perform the task. To set up the KBS the operator must choose the assigned robot and table from the initial menu. Next the robot and table are aligned to ensure accuracy. The part is then mounted on the table in its initial position. The system is now ready for operation in the current cell. The next step is to enter the path programming section of the KBS. The first prompts are for the basic weld attributes which are needed by the weld planner. Weld path coordinates are then entered by putting the robot into teach mode or possibly down-loaded from a CAD database. (Data can also be entered manually by the user.) With this information the appropriate weld parameters are assigned to the current path. These are displayed for the user to inspect and modify if desired. Next the movement planner undertakes path feasibility analysis. For example, if the part to be welded is as shown in Fig. 6, there are two unique paths to be programmed. One is horizontal along the parts base and the other is at a sixty degree angle up an away from the robot. The first path is easily accessible by the robot and is in the optimum down hand orientation. The second path is considered vertical and extends beyond the reach of the robot. The first pass of the Movement Planner, through the application of the robot limitation rules, will identify this vertical path as infeasible with the current table position and return an error of *Reach Out of Range*. If instructed by the user to try and reorient the part, the movement planner would then invoke the second phase and examine the path orientation and coordinates.

The movement planner would determine that if the table surface is rotated by 180 degrees and the base is tilted by 30 degrees toward the robot, the path would be feasible and follow the otimum down hand position. If the user wishes to have more control over the system, he may enter any position for the table to be oriented and execute the weld as long as the weld planner determines that it is feasible. For illustrative purposes, the structure of typical rules used in the KBS is presented below:

```
if
  (weldxmax <= XMAX && weldxmin >= XMIN
  && weldymax <= YMAX && weldymin >=
  YMIN)
  ROT180 = TRUE;
else
if
  (weldlengthxy <= MAXLENGTHXY && weldymax
  <= YMAX && weldymin >= YMIN)
  ROT90 = TRUE;

if
  (ROT180 == TRUE && weldslopexz != 0.0)
  TIPSLOPE = TRUE;
else
if
  (ROT90 == TRUE && weldpitch < -90.0)
  TIPPITCH = TRUE;
```

The next planned phase for this research project is to build an actual welding environment where the cell is scheduled to consist of a Unimate 760 Series six-axis articulated arm, using a VAL II controller and a full scale welding positioning table. Since several robot control functions are incorporated in the controller of a welding robot, the KBS can be freed from performing the computations that are required for the instructional robot. A modified version of this knowledge-based

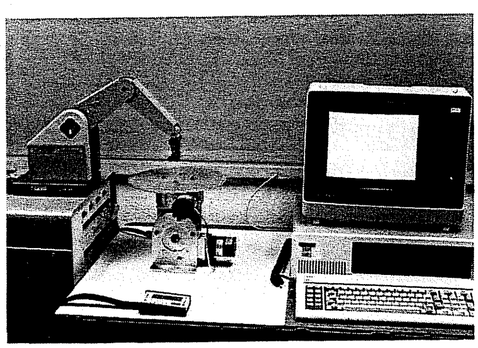

Fig. 5 Experimental test cell

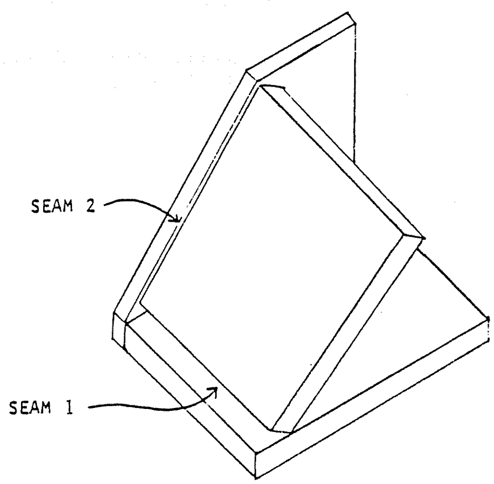

Fig. 6 Sample part

system would be combined with a totally integrated robotic welding system similar to the Puma Arc Welding System (PAWS).

6 Summary, Conclusions, and Future Scope of Research

A knowledge-based system (KBS) has been developed for continuous seam welding processes such that high quality welding can be autonomously performed in a relatively short period of time with (possibly) incomplete knowledge of the welding cell environment. The KBS is designed to coordinate the robot and table movements independently by examining the initial programmed path and determining the feasible table orientations and robot trajectories that could improve the weld quality. It is also capable of determining appropriate weld parameters (such as voltage, speed, and feedrate) for certain types of seams on the basis of the job description.

With the advent of flexible manufacturing, the emphasis has recently been on the development of systems that are easily adaptable to the changing demands of the factory. Dedicated welding cells for one specific part type are expensive and costly to modify. The KBS described in this paper is designed to faciliate the flexible use of combinations of different types of robots and positioning tables.

The major accomplishments of the research reported in this paper are: (*1*) The developed software is modular, which allows for the addition of more detailed function modules or databases without altering the rest of the program. Different function modules may be added for new types of robots and positioning tables being used. Similarly, the obsolete types can be deleted. (*2*) The advantages of both algorithmic and heuristic procedures have been combined to provide complete control of the welding cell in real time. Coordinated control of the robot and table is achieved along with weld parameter determination. (*3*) An experimental test cell has been developed as a proof-of-concept in which the prototypical KBS software has been successfully implemented.

The presented work, though preliminary, should provide a strong basis for actual implementation in the manufacturing environment. The results of the initial testing are encouraging and suggest several directions for future research to be carried out.

An area for future work is to improve the robot control module by including enhanced capabilities for seam tracking. For example, Nayak, Ray, and Vavreck [25] have reported the development of an adaptive real-time intelligent seam tracker using a 3-D laser vision sensor; Khosla et al. [18] developed an iterative algorithm using the inverse Jacobian to track a weld seam in real time. Various algorithmic and heuristic procedures could be incorporated into the robot control module to augment the system capabilities for real-time seam tracking.

The KBS structure would also be well-suited to coordinated control of two robots [26,27]. The control of the two degree-of-freedom positioning table in the current configuration could be augmented for the motion control of the second robot.

Acknowledgment

The authors are thankful to Mr. Henry E. Watson of the Applied Research Laboratory of the Pennsylvania State University for providing the financial support for this research work.

References

1 Pavone, V. J., "User Friendly Welding Robots," *13th International Symposium On Industrial Robots and Robots 7, Conference Proceedings*, Vol. 1, April 17-21, 1983, pp. 6.89-6.102.

2 Lane, J. D., ed., *Robotic Welding*, Springer Verlag/ IFS Publications, U.K., 1987.

3 Warnecke, H. J., Gzik, H., and Scholz, H-J., "Computer-aided Welding Operations—A Comparison of Future and Current Potential," *Welding and Cutting*, September 1984, pp. E141-E144.

4 Taylor, W. A., "Expert Systems to Generate Arc Welding Procedures," *Metal Construction*, July 1986, pp. 426-431.

5 Winograd, T., *Understanding Natural Language*, Academic Press, New York, 1972.

6 Grossman, D. D., and Taylor, R. H., "Interactive Generation of Object Models with a Manipulator," *IEEE Trans. on Systems, Man and Cybernetics*, Vol. SMC-8, No. 9, September 1978, pp. 667-679.

7 Taylor, R. H., "Planning and Execution of Straight Line Manipulator Trajectories," *IBM Journal of Research and Development*, Vol. 23, No. 4, July 1979, pp. 424-436.

8 Luh, J. Y. S., and Lin, C. S., "Optimum Path Planning for Mechanical Manipulators," *ASME Journal of Dynamic Systems, Measurement, and Control*, Vol. 102, June 1981, pp. 142-151.

9 Luh, J. Y. S., Walker, W., and Paul, R. C., "Resolved-Acceleration Control of Mechanical Manipulators," *IEEE Trans. Aut. Control*, Vol. AC-25, No. 3, June 1980, pp. 468-474.

10 Hsia,T. C., "Adaptive Control of Robot Manipulators—A Review," *Proc. IEEE Intl. Conf. Robotics and Automation*, San Fransisco, April 1986, pp. 183-189.

11 Horn, B. K. P., *Robot Vision*, MIT Press, 1986.

12 Lieberman, L. I., and Wesley, M. A., "AUTOPASS: An Automatic Programming System for Computer Controlled Mechanical Assembly," *IBM Journal of Research and Development*, Vol. 21, No. 4, July 1977, pp. 321-333.

13 Taylor, R. H., Summers, P. D., and Meyer, J. M., "AML: A Manufacturing Language," *The International Journal of Robotic Research*, Vol. 1, No. 3, Fall 1982, pp. 19-41.

14 Fikes, R. E., and Nilsson, N. J., "STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving," *Proc. Second Intl. Joint Conf. AI*, September 1971, pp. 608-620.

15 Cai, Z., and Fu, K-S., "Robot Planning Expert Systems," *Proc. IEEE Intl. Conf. Robotics and Automation*, San Fransisco, April 1986, pp. 1973-1978.

16 Kerth, W. J., "Knowledge-based Expert Welding," *Robots 9 Conference Proceedings*, Detroit, MI, June 2-6, 1985, Advanced Applications, Vol. 1, pp. 5-98 to 5-110.

17 Kumara, S. R. T., Joshi, S., Kashyap, R. L., Moodie, C., and Chang, T. S., "Expert Systems in Industrial Engineering," *Intl. Journal of Production Research*, September-October, 1987.

18 Khosla, P. K., Neuman, C. P., and Prinz, F. B., "An Algorithm for Seam Tracking Applications," *The International Journal of Robotics Reseach*, Vol. 4, No. 1, Spring 1985, pp. 27-41.

19 Kambhampati, S. K., and Davis, L. S., "Multiresolution Path Planning for Multiple Robots," *IEEE Jour. Robotics and Autom.*, Vol. RA-2, No. 3, September 1986, pp. 135-145.

20 Herman, M., "Fast Three Dimensional, Collision Free Motion Planning," *Proc. Joint Conf. on AI*, August 1986.

21 Chaudhuri, B. B., "Application of Quadtree, Octtree and Binary Tree Decomposition Techniques to Shape Analysis and Pattern Recognition," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. PAMI-7, No. 6, November 1985, pp. 652-661.

22 Clocksin, W. F., and Melhish, C. S., *Programming In Prolog*, Springer-Verlag, 1984.

23 Winston, P. H., and Horn, D. M., *LISP*, 2nd ed., Addison Wesley, Reading MA, 1984.

24 Kernighan, B. W., and Ritchie, D. M., *The C Programming Language*, Prentice-Hall, Englewood Cliffs, NJ, 1978.

25 Nayak, N., Ray, A., and Vavreck, A. N., "An Adaptive Real-Time Intelligent Seam Tracking System," *Jour. Manufac. Systems*, September 1987.

26 Hemani, A., "Kinematics of Two-Arm Robots," *IEEE Jour. Robotics and Autom.*, Vol. RA-2, No. 4, December 1986, pp. 225-228.

27 Tarn, T. J., Bejczy, A. K., and Yun, X., "Coordinated Control of Two Robot Arms," *IEEE Intl. Conf. Robotics and Automation*, San Fransisco, CA, April 1986, pp. 1193-1201.