# 16. The CAM System for the Arc-welding Robot for Hull Sub-assemblies

Tasuku YOSHITOMI*, *Member*, Takashi YOSHIMURA**, *Member*,
Kouji NISHIYAMA**, Nobuya AKIYAMA***,

(*From J.S.N.A. Japan Vol. 167. June 1990*)

**Summary**

In order to extend the applicable range from the parallel parts to the non-parallel parts of hull structures, the authors have developed a new CAM system for the arc-welding robot.

When automatic arc-welding is extended to all the non-parallel hull part workpieces of different shapes, the new CAM system is required to respond flexibly to the alterations of production order, shapes and the laying positions of related workpieces, without taking much time for programming.

Satisfying these needs, the following means has been studied for the new system.

(1) The new CAM system supplies a task level welding program to the robot controller, which simultaneously generates every robot motion. Consequently, when it becomes necessary to alter the laying positions of workpieces due to the change of production order, the robot can react freely to the alterations.

(2) For saving programming time and for simplifying the operation, the system of a human interface with an interactive mode is adopted using graphic displays. Operation guides of the system and those commands for selection are both shown on the graphic displays.

(3) For saving time and avoiding errors, the data-base of the hull structures is used for programming.

(4) A micro computer is provided at the control room in the subassembly shop for welding, altering and maintaining the welding program. The new function enables the system to react flexibly to the alterations of design and production order.

It is estimated that utilization of the new CAM system reduces programming time by 2/3.

Experimental calculations on the economy of the robot based on the actual record of programming and welding time has confirmed that it is possible to maintain an economical advantage even if applying the robot to workpieces of various shapes.

The arc-welding robot for hull sub-assemblies can be effectively used on non parallel part of hull structures by employing the new CAM system.

This new system offers full automation potential for every welding activity on the subassembly line of hull structures.

## 1. Introduction

Nowadays, keeping labor power is an important subject on manufacturing industries in Japan. Accordingly, the ship-building industry are aiming to innovate manufacturing processes. They are proceeding mechanization of building process with the development of qualified facilities. These may assure better working circumstances, higher productivity of welding works and improved quality of products.

There are simple shaped sub-assembly workpieces that have a lot of parallel welding lines. The multi-torched machines can weld such workpieces with high productivity because the torches of these machines run parallely and weld many lines simultaneously.

According to the nearest annual working record of a hull sub-assembly workshop, such the specialized machines weld 70% of total

* Faculty of Engineering, Kyushu Kyoritsu University
** Nagasaki Shipyard and Engine Works, Mitsubishi Heavy Industries, Ltd.
*** Nagoya Machinery Works, Mitsubishi Heavy Industries, Ltd.

welding length[1]. For the automation of remaining welding, the arc—welding robots should weld complex shaped workpieces. The authors have developed arc welding robots for hull sub—assemblies. The robots are executing the welding work on the parallel part.

On a non—parallel part of hull constructions, however, there are inferior conditions for automatic welding. Hence it is necessary to develop additional features in order to extend field of applications of robots toward that parts. That is, time for programming should be reduced more because a shape of every workpiece is different. Also, it is difficult to set levelled production schedule in advance. Then, it is necessary to add more flexible functions because the robots should allow change of working schedule and/or shape of workpieces.

By the research of solving methods for this problem, the authors developed a CAM system and put it into practice. The system calls for numerical data—base of hull constructions for reduction of programming time and prevention of programming errors. The system offers full automation potential for every welding activity on the sub—assembly lines of hull structures.

This paper describes advanced features of the system and its application records.

## 2. Structure of the CAM system

### 2.1 Total structure

The authors have developed the arc—welding robots for hull sub—assembly works. The robots can perform welding works with task level programs. Also, they provide flexible functions to weld stiffeners with complicated arrangement.

These robots bring forth the reduction of welding man—hour by 75%[1]. Still, the robots can perform economical gain only at a parallel part of hull structures.

For the full automation of the welding works on sub—assembly lines, the robots should extend an applicable range into non—parallel parts of hull structures. However, every workpiece on non—parallel parts has a different shape. Then, a task teaching is necessary for each workpiece. For the economical use of the robot on those parts, time required for task teaching should be reduced more.

In the ship—building industries, the revisions of construction schedules and design changes are inevitable. Thus, flexibility of production systems is necessary. Bringing a finished workpiece out the operational space, this place is vacant. Next workpiece occupies this vacant place. However, the vacant place cannot decide previously because an order of welding works changes so often. Then, the system should be so flexible that a workpiece occupy a free place.

(1) Generation of motions by the robot controllers

As shown in Fig. 1 (a) and (b), there are two levels of welding programs. One is motion level and the other is task level. Motion level programs give instructions of motions such as path, speed, attitude, settings of welding, to the robot controllers. Task level programs give statements that describe motions collectively. In this case, the robot controllers generate each motion.

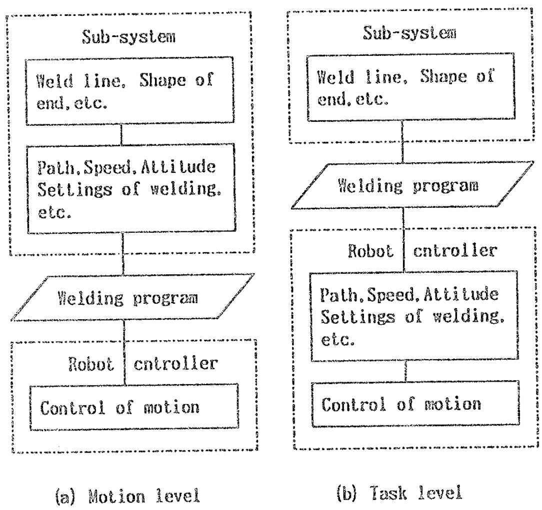


Fig. 1 Levels of welding programs

For ship—building works and steel fabricating works, CAM systems for arc—welding robots are reported [2)3)4)5)6)]. All the systems are motion level in order to reduce processing load of robot controllers. In case of these systems, each motion of robots is decided when programs are listed. Then, such robots cannot contain workpieces at a free position. Accordingly, it is difficult for the robots using motion level programs to be used economically for the welding works at the hull sub–assembly shop.

Contrarily, using task level programs, robot controllers produce each motion when the controllers execute welding programs. So, a workpiece can occupy a free position in the operational space of the robot.

Also, task level programs describe whole aspect about motions of welding works. Then, using task level ones, welding programs become shorter. Also, programmers can list program within short period of time.

Therefore, the authors have developed the CAM system of task level newly.

(2) Conversational mode and use of data–base

To reduce programming time more and to avoid program errors, the authors have developed a computer aided system with conversational mode. As operations are graphical works, human interfaces of the system are graphic display terminals. Also, for easy operations, the system shows guidance of operation and selection menu conversationally on the display planes.

Welding programs include large numbers of numerical data about shapes of hull constructions. Fortunately, the shipyard to equip robots is applying data–base about shapes of hull construction. A system named Ship's Hull Information Processing System (SHIP system) processes the data–base[7]. Also, at the preceded programming works for gas cutting operations, programmers check errors of the data. Then, the use of the data–base avoids numerical errors of the welding program.

(3) Distributed functions

In cases of no revision of works, on–line systems are most efficient because the systems can collectively operate many functions. If there are many revisions, an on–line system is not suitable because it requires complex operations for revisions. In the ship–building industry, revisions of works are inevitable. Then, the CAM system furnishes distributed functions.

The system comprises hardware as shown in Fig. 2. Also it comprises software as shown in Table 1. IBM 3081 is in wide use as a main frame computer for all the shipyard. This computer produces shapes of the hull parts and manages the data–base, super mini–computer of COSMO 800 is used exclusively for the SHIP

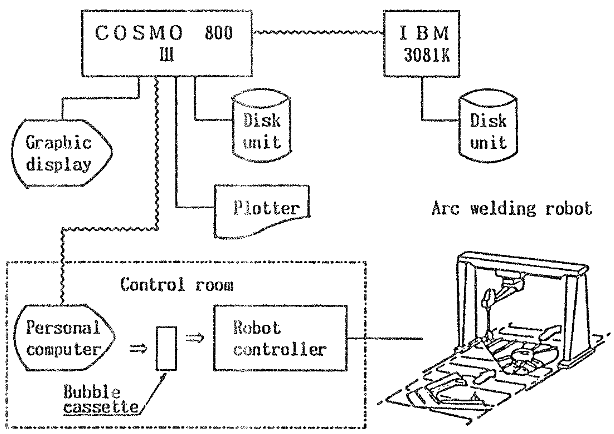


Fig. 2 Hardware configuration

Table 1 System configuration

| Function | Processing matter | Size | Hardware |
|---|---|---|---|
| Coding of Welding Program | Coding of task level program<br>Conversational use of G/D*<br>Output of parts drawing data | 1.1MB | COSMO 800<br>IBM 3081K |
| Parts Drawing | Plotting of parts drawing | 0.4MB | |
| Simulator of Robot Controller | Input, editing, checking syntax, calculating path, Outputting data with defined format | 0.5MB | Personal Computer |
| Path Display | Displaying calculated path | 0.5MB | |
| Robot Control | Generating motion level instructions with task level prog. and executing motion control | 0.6MB | Robot Controller |

* G/D : Graphic display

system. The graphic displays act as the terminals of this computer and are used for programming.

The system provides a personal computer at a sub-assembly shop. This computer can independently run and perform editings, revisions, maintenance and file management of welding programs. Receiving programs through a communication line, this computer also plays the role of a terminal of the super mini-computer. The personal computer receives welding programs with this mode. Also it writes them to bubble memory cassettes which are the media for the robot controller.

The following specific features were developed for the arc-welding robot for hull sub-assembly works[1].

1) The robot is furnished with task level robot language and produces each motion.
2) The robot is furnished with functions sensing both ends of welding lines and does boxing weld at each end.
3) Locations of workpieces do not have an effect on welding programs. So, the robots may easily run even if setting positions of works are altered.

Handling facilities at automated production lines are generally conveyers. However, systems using conveyers lack a flexibility for revisions of production order.

In the sub-assembly shop, workpieces are handled with cranes in order to keep flexibility for alterations of production order. Handling means bringing a finished workpiece out of the operational area of the robot and carrying a new one in there.

If a crane can handle a workpiece while a robot is welding another one, it is economical way. Therefore, the area of 20.0 m x 4.5 m which the robot covers is sufficiently enough to put two or more workpieces at the same time.

Compared with conveyer systems, the handling system with cranes demands greater labor-hours but give more flexibility for the revisions. Then, the system conforms to the hull sub-assembly works.

## 2.2 Welding program

(1) Levels of robot languages

On welding works at a hull sub-assembly shop, it is necessary to alter welding programs at the robot site because many revisions of designs often require immediate alterations of the programs. Therefore, programs of the robots should be so easy that operators can easily revise them at the site of robot.

Robot languages are the most suitable style of programs for the robots with higher level intelligence. Then a robot language is to be adopted to human interface for this robot.

There are various levels of description for robot languages. The lowest level corresponds to servo control level. Higher level languages instruct jobs inclusively. Here, the authors study levels about practical applications of robot languages.

Japan Industrial Robot Association issued a standard of a robot language in May 1989. This language is motion level to describe each motion of robots. It includes instructions for numerical calculations and sequential commands. Then, it may usually fulfill functions which are required now or in the near future[8].

Also, languages for practical applications of robots to ship-building industry and other industries are all motion level much the same as the standard of the Association[2)3)4)5)9)10)11)12)].

The controller of this robot should produce instructions of each motion to be flexible to revisions of works. Also, it is necessary to reduce amounts of programs for reduction of programming time. Motion level languages cannot fulfill such requirements. Then, a task level language which describes the whole aspects of motion should be put into the controller.

(2) Task level programs

By limitation of applications, task level language can be practical. The authors have developed task level robot language for the hull sub-assembly works. We take note of a similarity about form of workpieces and welding works as mentioned below.

1) Web-plates lie horizontally with stiffen-

ers etc. being put vertically on them.
2) There is open space above the workpieces.
3) There is a module of frame space.
4) Shapes of welding lines are combinations of straight lines and circular arcs on horizontal planes or vertical lines.
5) Type of weldings is fillet weld.

The task level languages are applicable to the non–parallel parts because these particulars are common. Welding programs given to the robots are described by the task level language. Welding programs comprises seven kinds of components. The components are[13]:

1) The definitions of the value on the coordinates concerning the both end points of welding lines,
2) The definitions of the three reference points, which are used to define the workpiece location,
3) The statement to quote the program called MACRO, which detects the workpiece position,
4) The instruction of the direct teaching of the path to detect a workpiece,
5) The order of coordinate transformation and its stop order,
6) The sub–program name appointing the type of welding, the point mark of the both ends and the index showing the type of terminal welding,
7) The name of welding program and the statements of start and end of the program.

(3) Instructions of motions

When the robots weld workpieces, they need instructions of motion. The instructions are:

1) Type of motions and their sequences,
2) Controls to keep each joint within its operational range,
3) Values on the coordinates of the path points,
4) Conditions on the motions, such as speed and attitude,
5) Welding conditions such as the welding current, the welding voltage, the weaving pattern and the arc–sensing parameters,
6) Control of programs to subdivide the type of welding works or to treat abnormal conditions.

As an example, the work to carry out single path of a horizontal fillet welding comprises 24 segments of line motions including sensing motion to detect a start point. Then, the robot needs 150 or more instructions concerning the positions, the attitudes, the speeds, the welding currents, etc..

The motion level program to produce such motions and their conditions comprises about 120 lines[1]. Also, the instructions to be used to the program comprise dozens of such kinds.

The welding program of task level produces only 6 items of instructions. Those items are the type of welding lines, the positions and type of welding at each end, also the side to be welded. Therefore, to convert the task level program to the instructions for the robot motions, over 140 items of instructions remaining should be additionally produced.

The welding works for the hull sub–assembly can be standardized into 4 types. These are the horizontal fillet weld of a straight line, the horizontal fillet weld including a circular arc, the vertical fillet weld, and the detection of the workpiece position.

(4) Generation of instructions

In order to operate robots easily even if setting positions of works are altered, the authors have developed following methods[13].

1) For each type of welding, it is possible to unify the motions and their sequences into one routine. The detailed paths at the ends of the horizontal fillet welding can be standardized into 4 types. According to these standardization, a program for each type welding becomes an identical arrangement of instructions.
2) All points on a path can be the same positions relative to the welding line as far as welding type is the same. So, in order to define each passing point with same description, the origin of the coordinate system puts on the start of the welding line or the end of it. Also, one coordinate axis is taken along the welding line while the other is taken perpendicularly.
3) In order to describe welding program independently to location of workpieces,

the system provides various means to express positions. The means comprise definitions of positions based on three kinds of coordinate system, geometrical calculation packages and calculation packages for coordinates transformation.

4) A worker can detect a laying position of a workpiece based on a customized method. The method comprises teaching four points directly on the detecting paths and determining the other passing points automatically from these four points.

5) In order to get higher operating speed, the authors executed many trial operations. These were trials on the level of moving speed, torch's attitude, welding condition, etc. The sub–programs include selected conditions which were determined by those trials.

6) To operate the robot continuously, it is necessary to hold every joint of the robot within its operating angle. If a program defines a position of a torch and its attitude, reverse kinematics for robots give solutions of angles at joints. For the mechanism of this robot, two solutions exist for one position of the torch. The adoptable solution to operate continuously is one of these two. Hence, the authors have developed algorithms to select the adoptable solution. The algorithms are written using calculation packages in the sub–programs.

(5) MACRO

The instructions to control and execute motions are organized into the subprograms called "MACRO". A unique name is given to each MACRO. The robot controller has the MACROs in its storage memory.

To reduce the total quantity, the MACROs are to be of hierarchy construction so as common portions to be used jointly. The total quantity of the MACROs consists of about 570 lines.

For an operator to deal easily with the welding programs in quoting the MACRO, the authors developed the linking function. When a name of the MACRO is given to a controller, the controller finds it from its storage memory. Then, the controller links the MACRO automatically to a main program.

The MACROs also call welding conditions together with their arguments. The conditions comprise the positions of the welding lines, the types of welding at their ends etc.

(6) Motion level language

When a programmer applies the task level language to a new type of workpieces, it is necessary to improve or modify the functions of the language. As operators of the robot should easily amend the language, the MACROs are described in the motion level language.

The motion level language includes 102 kinds of instructions. They are listed in Table 2. The language should describe the algorithms to produce each motion. It has the distinctive feature as follows.

1) The definitions of the positions are separated from the instructions of the motions. It is necessary to use the statements common to welding lines of same type even if there are differences in positions, directions or lengths. The conditions to be instructed to the robot are described in the form of system variables. Also, separate instructions give the values of these variables.

2) It is necessary to describe algorithms of motions independently of the laying positions of workpieces. The authors developed freer means to define positions. The means comprise six groups of instructions. These are one set of instruction to transform coordinates, one set of instruction to make mirror image, one set of calculating packages for geometrical calculations or coordinates' transformation and three types of instructions to define positions.

3) For the welding, instructions should define a trapezoid weaving and set sensitivity control of arc–sensor. Therefore, the authors have developed the instructions to define the weaving pattern and the conditions on arc–sensor.

4) In order to define standardized works

Table 2 Instructions of the robot language

| Function of instruction | | No | Explanation |
|---|---|---|---|
| (1)Definition of Data | | | |
| Position | Rectangular coord. | 6 | Based on work,welding line or torch; Absolute or relative |
| | Joint Angle Coord. | 1 | Definition with joint angle |
| | Position of Gantry | 2 | Absolute or relative running |
| Torch Attitude | Torch angle | 2 | Same manner to welding work |
| | Vector | 2 | Definition with unit vector |
| Condition of Motion | PTP | 1 | Speed |
| | Strait,Circle | 5 | Speed, weaving, arc-sensor |
| | Wrist | 1 | Executing period |
| Condition of welding | | 1 | Welding current, voltage |
| Data for Timer | Delay timer | 1 | Delaying period |
| | Waiting | 1 | Bit pattern of waiting signal |
| | Output | 1 | Bit pattern of output signal |
| (2)Motion Control | | | |
| Moving | PTP | 1 | Moving to defined point |
| | Strait | 1 | Strait moving to defined pt. |
| | Circle | 1 | Circle, passing thru mid. pt. |
| | Wrist | 1 | Rotating wrist, keeping original tip position |
| | Memo. Motion | 2 | Moving along memorized path |
| Welding | | 3 | Start,stop and changing cond. |
| Misc. | Output of timer | 5 | Waiting timer, input, etc. |
| | Complex motion | 2 | Touch sensing, adjust of wire |
| | Memorize | 2 | Start/stop of memorization |
| | Pause | 1 | Pause untill control signal |
| | Reading | 7 | Reading existing condition |
| (3)Calculation Package | | 15 | Geometric, coordinate trans. |
| (4)Variable, Scalar quantity | | | |
| Arithmetic Function | | 12 | Including trigonometric, square root, absolute value |
| Valiable | | 1 | Free scalar quantity |
| (5)Program Control | | | |
| Unconditional Jump | | 1 | Jump to Labeled name |
| Conditional Branch | | 1 | Branching with comparison of scalar quantity |
| Repeat | | 2 | |
| Macro | | 2 | Calling and returning |
| Subprogram | | 2 | Calling and returning |
| Stop | | 1 | Termination of automatic ope. |
| (6)Transformation of Coordinates | | | |
| Shifting and Scaling | | 3 | Shifting and scaling of coordinates' value |
| Mirror Image | | 2 | Generation of mirror image |
| (7)Specification Statement | | | |
| Program / Macro | | 4 | Definition of name and declation of start/end |
| Data / Sequence | | 4 | Declaration of start/end of data/execution |
| (8)Comment, Message | | 2 | |

shortly in a lump, the authors developed the function of sub–program named MACRO. In order to describe MACROs of task level, 77 kinds of instructions, which are about 3/4 of total 102 kinds of those, are used.

5) The MACROs are registered to the robot controllers with their own unique name. When the controller reads the program which include the list of MACROs, the controller quotes the sub–programs from its storage memory as it is instructed. It also links the MACROs to the main program.

The statements of MACROs can list arguments to define various conditions of motions. Therefore, the quoting MACROs is equivalent to programming the task level program.

However, robot languages for other robots present cannot list the generation of the motions. Because, they do not involve the functions mentioned before. For application of robots to non–parallel parts of hull structures at sub–assembly works, it is difficult to realize a flexible system with such languages.

### 2.3 Programming sub-system for welding works

In order to apply robots to non-parallel parts, it is necessary to reduce the time required for programming further. Making use of the characteristic of a task level language which has a simple syntax and few statements, the authors have developed a programming sub-system for those robots. The particular features are as follows.

(1) Conversational operations

Compared with the quantity of welding works by the robots, programming work is rather small. Hence full-time operators are not necessary. To promote efficiency of operator's jobs, an operator himself to define the form of a workpiece also does programming of welding the workpiece. Thus, all the mold-lofting workers may use the sub-system for programming welding procedure.

When there are many operator/workers but their use of the system is not frequent, it is difficult for all of them to memorize complicated operations of the system. Thus, simple and easy operations are necessary for practical use of such systems.

There are many graphic jobs at the programming works of welding programs. If a graphic language is used for a human interface, the system become difficult to be used because operators must learn many commands or instructions. On this account, the authors adopted graphic displays for human interface, which can be operated easily only by pointing appropriate items.

The authors employed menu system in order for operators to be able to program without memorizing operations. The system shows choices of procedures or instructions on displays.

Layout of a displayed plane is shown in Fig. 3. The system always displays choices of operating procedures and supreme choice of task level instructions at the customized commands area and customized instructions area respectively. When an operator hit a choice, the system displays necessary commands or instructions, which are below the hit choice, on sub-commands area.

The controller generates each motion of the robot using the registered MACRO. Thus the number of welding programs is reduced to 7

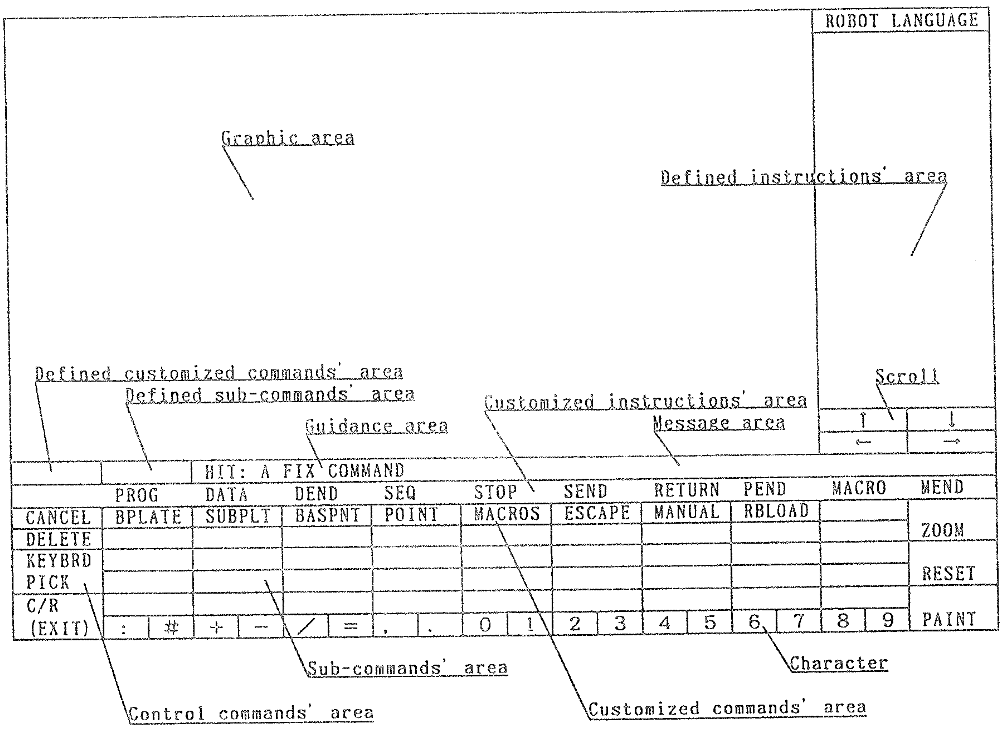


Fig. 3 Layout of a display face

kinds of statements previously described. Also, all statements have own customized format and can be described by 18 kinds of instructions.

Using the customized format and the limited instructions, the system displays guidance of operating sequence, instructions which should be selected next and a sample statement for guidance to format of the listing statement. Also, the system changes colors of indicated areas and highlights them to call the attention of operators.

The operators can easily program following the guidances and choosing the displayed instructions. This programming method prevents syntax and logical errors in the statements.

These improvements of human interfaces bring forth reduced time for programming and prevention of errors. Also, training on programming is virtually unnecessary.

(2) Utilization of data base

In the task level program, about 2/3 of the whole statements are the definitions of the ends of welding lines. It is necessary to prevent errors in these statements and to reduce time for programming. Hence, the authors have developed a programming system to define coordinates with the use of existing data base concerning hull structures.

When an operator points a base plate of hull sub-assembly, the system displays the base plate with all the welding lines on it. The system uses the parentage included in the data base for identifying the welding lines. The identifying function prevents missing of welding lines.

Next, the operator should point the displayed welding line to follow the order of the welding work. By these operations, the system puts sequential numbers on both the ends of the pointed welding line automatically. The system also generates statements to define these points simultaneously as instructed by the grammar of the robot language. At this time, the system convert coordinates from the coordinate system on the ship's hull to a local one on the base plate.

That is, the system contains automatic functions, such as generating points, converting coordinates and generating statements in the robot language. The functions can perfectly prevent errors in statements to define positions.

(3) Free positioning

The authors have also developed the following method of free positioning of a workpiece.

First, operators set a workpiece's coordinate system on a workpiece, having no relation to a setting position. The operators also define three reference points on this coordinate system. The robot can recognize the location where the workpiece is set by the distances measured from the reference points.

Secondly, the robot detects the location of the workpiece by the touch sensing motions. The system customizes sensing motions in a MACRO.

Finally, the controller executes the coordinate transformation from the workpiece's coordinates system to the robot's one following the instruction. The symbols of the reference points in the instruction transfer the data of the locations.

By this feature, programming operators need not pay attention to the positions of workpieces set at the robots. This feature has an advantage for the operators who is belonging in a mold lofting shop. Because the operators may strange to the operations of the robots in detail.

(4) Connections with SHIP system

Data base is built by the SHIP system. It contains numeral data attributive to all hull structure members such as form and name[7]. The data base acts as "a source to supply data" for the mechanization and the automation of the hull construction.

The sub-system for welding program and the SHIP system use the data base in common as shown in Fig.4.

Many functions are common to this programming system and the SHIP system. Common functions are transformation among coordinate systems, projecting workpieces, pointing workpieces, joining workpieces together, dividing a workpiece, geometrical calculations, basic routines for graphic displays, etc.

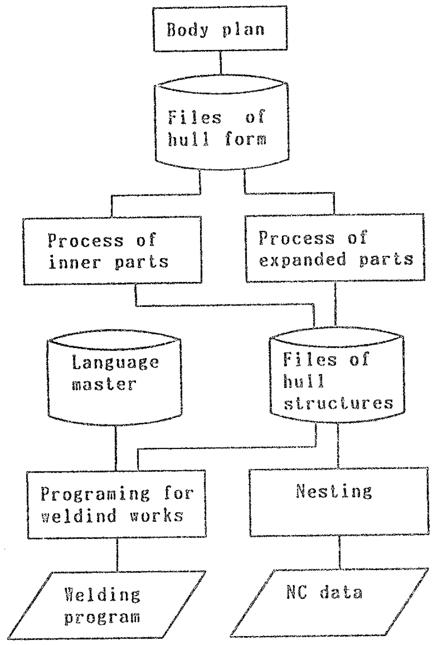


Fig. 4 SHIP system

This programming system is a part of the SHIP system. Both systems use computers, graphic displays, a data base and modules of programs in common. Thus, the costs for developments and operations are to be reduced. It has also an advantage in reduction of training time because programming operators usually have experience to operate the SHIP system.

(5) Improvement of operations

The authors have developed functions of human interface for easy programming. These are the utilization of the menu, the operating guidance and the changing colors and luminance also.

The improvements are illustrated on examples in Fig. 5 where a portion of menu section is shown by different scale in order to cut down the size in printing. Fig. 5 (a) shows the image plane at the start of program where base plates are called in. The system displays names of workpieces at the center of the display. When a workpiece is selected, the system highlights the selected name for easy confirmation. It also shows the form of selected workpiece at the upper part of the display. The automatic function to join workpieces with a butt joint has been developed.

Next, it is necessary to determine reference points of the coordinate system. Fig. 5 (b) shows the image plane where the origin of the coordinate system is given. The system displays necessary commands to determine reference points on the sub-command area and displays message to request the input of reference points. The operators can define reference points by the cursor of cross mark. It is not necessary for an operator to know the sequences of the operations owing to these features. The system automatically transforms the hull coordinate system to the workpiece coordinate system.

After the determination of the workpiece coordinate system, the procedure to define welding lines follows. Fig. 5 (c) shows a zoomed image plane for detailed observation. In this procedure, the system reads the coordinates at the ends of designated welding lines from the data base and generates statements to define such points. The commands used for definitions of welding lines are displayed on the sub-command area. Thus, it is unnecessary to memorize the operations.

Fig. 5 (d) shows an image plane at quoting the MACRO. The system displays listed statements at the output area of the robot language. At the procedure to list welding programs, first, the operators select an instruction from 10 kinds of instructions. The system displays instructions at the displaying area of customized robot language. At this time, the system selects next necessary commands and shows them at the sub-command area. Likewise, operators can carry programming forward by choosing a command from displayed ones. That is, it is unnecessary to memorize the operations.

The listed welding program can be output to the file for welding programs and the data file for plotting forms of workpiece at one stroke. The COSMO manages both the files.

(6) Parts drawing

In order to deliver information concerning the positions of welding lines etc. to the robot operators, the authors developed an automatic drawing system. An example drawing is shown

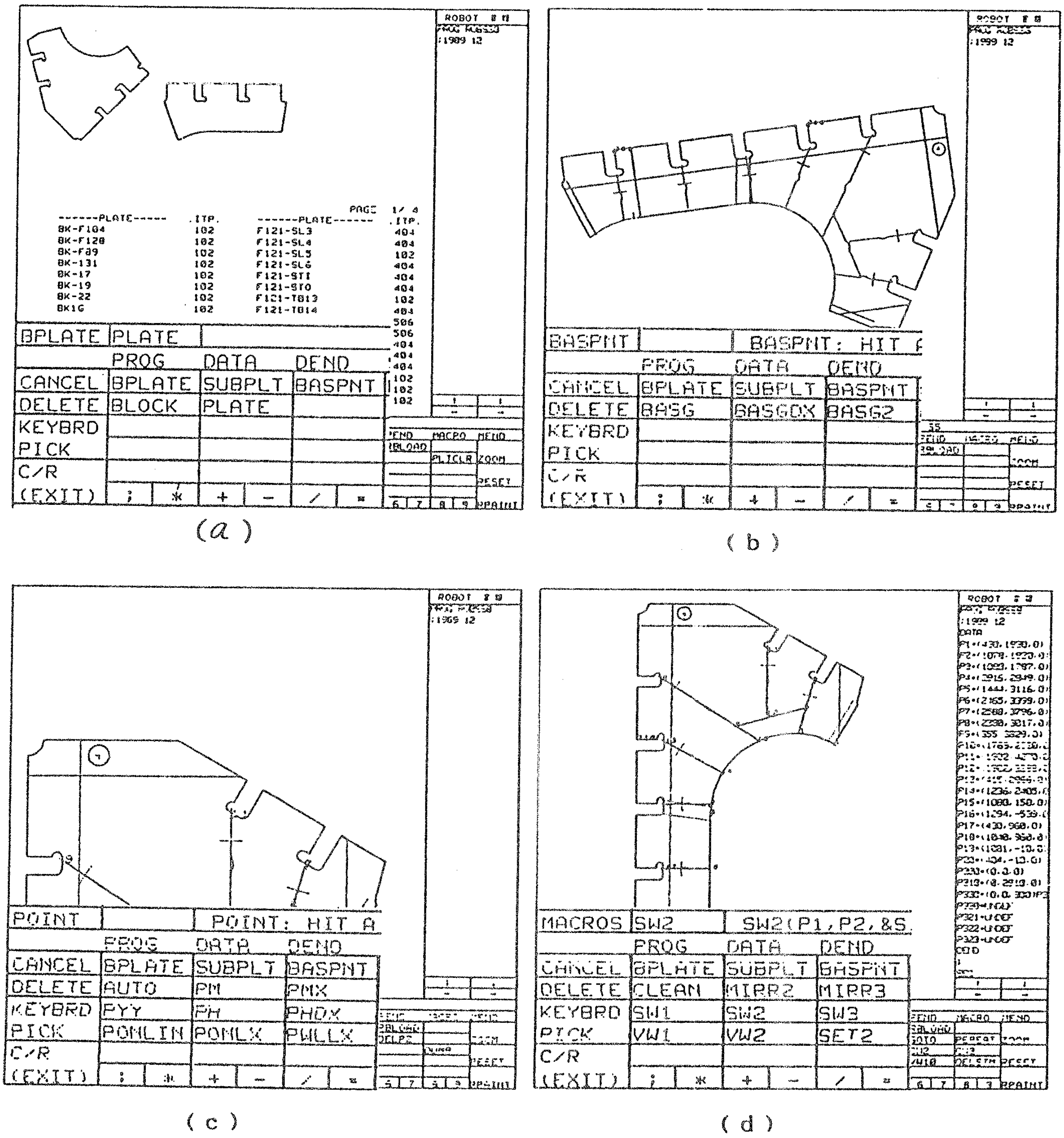


Fig. 5 Examples of displayed faces

in Fig. 6. The drawing shows a form of base plate, positions of welding lines, reference points of the coordinate system, moving directions of the welding torch, etc.. The drawings are very useful for the robot operators because these include all necessary information to weld workpieces.

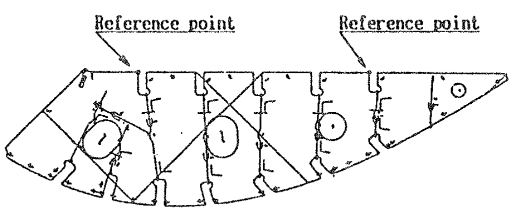


Fig. 6 An example of parts drawings

(7) System program of the sub—system

The system generates task level program by easy operations, that is, the choice of menu. The system calculates coordinates with utilization of the data base. Also, it generates welding statements with the use of stored formats and the sequential numbering feature properly.

Then, the system acts as an input generator for the robots for hull sub–assemblies.

The main jobs of this system are handling files, handling character strings and operations of graphic displays. The programming languages for the former two are FORTRAN. The language for the last one is the subroutine packages, which are supplied by the manufacture of the super mini–computer.

The task level programs comprise the character strings of instructions and numeric data. The system generates the characters with a choice of the menu. Also, it calculates the numerics with the data base.

Then, the system prevents both the syntax error and the numeric error on the program. Therefore, it is not necessary to check the syntax of the programs and the path of programmed motions with the compilation of the programs. To simplify the system program, the system does not include a compiler for robot languages.

## 2.4 Personal computer

The CAM system should react flexibly to the alterations of designs and the changes of producing order. Hence, the authors set a personal computer at the robot control room in a sub–assembly shop.

(1) Terminal mode

Some workers selected from welding workers of hull sub–assembly shops operate the robots. As the workers can freely manage the welding program on hand, the personal computer acts as a terminal of COSMO with the connection by a transmission line. This usage is a terminal mode. The software for the terminal mode is a general purpose terminal emulation program. The workers can start terminal mode when necessary. Next, they execute jobs of selecting the welding program from the files of COSMO. Also, they execute jobs of loading it to the files of the personal computer.

(2) Simulator mode

If a worker modifies a welding program at the shop, it is necessary to check syntax of a program, its compositions and path of motions. At this time, the personal computer acts as a simulation system of the robot controllers. This is simulator mode. This mode contains following three functions.

1) It can input welding programs, edit them, compile programs written by the robot language and simulate execution of the programs.
2) It can not only read programs from files of personal computer but also output to the bubble memory cassettes, which is the media for the robot controllers.
3) It can show the path of the torch on the display plane of the personal computer.

The system programs for the former two are a copy of some part of the system programs for the robots as original. The copied part is shown in Table 3. A general purpose emulation program takes over these programs.

When someone modifies a system program for the robot, the programs for the simulator must be changed accordingly as soon as

Table 3 Simulation system of robot controls

| Function | Execution |
|---|---|
| Management of Program | File management of welding program<br>Control of peripheral equipments |
| Compiler | Syntactic analysis of robot language<br>Data management of robot language<br>Interpreting language at execution |
| Execution | Analizing intermediate code<br>Calculation of position, attitude |
| Mathematical Analysis | Calculation of vector, matrix<br>Transforming coordinate system<br>Interporation of strait, circle<br>Kinematics and reverse kinematics |

possible. Here, the copied programs are the best. The language of these programs is PL/M, the same as ones for the robot controller. The authors have coded the program for displaying torch path by BASIC and compiled it for higher execution. The format of welding programs consists of 80 characters with ASCII code on both the data files and the bubble memory cassettes.

(3) Operations

For the arrangement of the next day's work, a worker starts simulator mode to select programs from the files. His next job is outputting them to the babble memory cassettes. Also, the worker loads program to robot controller through bubble memory cassettes.

When a designer altered design partially after programming by the CAM system had been finished, workers can modify the programs and confirm path of motions in this mode. Therefore, it is easy to take measures to meet alterations.

## 3. The applied results of the CAM system

### 3.1 Reductions of the programming time

The robots for hull sub–assembly are run by the task level programs. Then, the number of their programs is about 1/35 of not only that for playback robots but also that for robots with motion level programs[13].

When the CAM system was not available, average teaching time for a workpiece of ordinary size was as follows. Time required for programming was about 1 hour. That for checking and correcting data was about 1 hour. That for part drawing was about 1 hour. Total came to be 3 hours.

In order to evaluate the results of the CAM system, the authors measured working time consumed for 400 programming jobs. Fig. 7 shows statistics of these measures.

The statistics show followings. Working time for about 70% workpieces was 20 to 40 minutes. The average time was 39 minutes. No time for part drawing was necessary because the system drew them automatically after the programming had been finished.

Also, errors in programs became small owing to the effective use of the data base and so on. Accordingly, the time for checking program became as short as that for the programming.

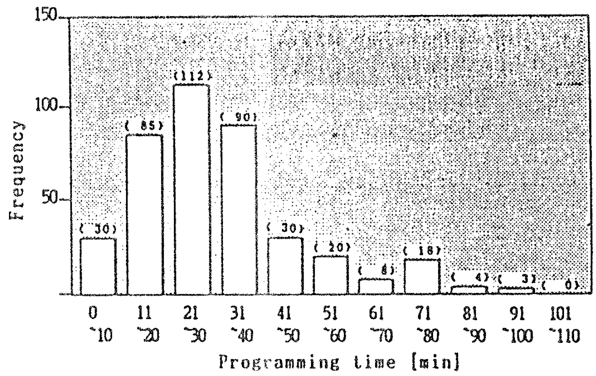


Fig. 7 Statistics of programming time

Using the CAM system, the average programming time for welding program became about 1 hour. As mentioned before, the total average teaching time was about 3 hours when the CAM system was not available. That is, due to the advantageous effect of the CAM system, the time required for teaching become 1/3.

Also, the CAM system prevents the programming errors on the coordinates or the arguments because the system generates welding program automatically with using the data base.

The record on the 20 abnormal stops of robots shows that 3 stops of them were caused by such programming errors[14]. Therefore, the system not only reduces programming time but also improves the net operating rate of the robots through the prevention of errors.

### 3.2 Expansion of the applicable range

Investments in automatic machine must be paid with the increased profits. In the application of robots, the inequality (1) gives an economical range of the amount of investment $P$.

$$P < \frac{F(G - r_D - r_P)\ E(K, i)}{(1 + r_D)\ \{1 + E(K, i) R\}} Y \qquad (1)$$

$Y$: Annual labor cost for a worker

$F$: The ratio of actual working time of a robot to that of a worker

$G$: The ratio of working ability of a robot to that of a worker

$R$ : Assumed ratio of repairing expence to investment $P$

$r_D$ : The ratio of direct teaching time to actual operating time

$r_P$ : The ratio of indirect teaching time to actual operating time

$$E(K,i)=[(1+i)^K-1]/[i(1+i)^K] \quad (2)$$

Uniform series present worth factor[15)]

$K$ : The term of depeciation in year

$i$ : Annual interest ratio

To compare on an equal footing, the length to be welded by manual welding is adjusted to be the same as that by automatic welding. As mentioned before, average programming time is 1 hour as shown in Table 4.

On the assumption of for $G$, $r_D$ and $r_P$ being the value in Table 4, conservative value of 1.0 for $F$, 8 years' depreciation, 8% of interest and the average value of 0.10 for $R$, the authors estimated the relation between $Y$ and $P$. These calculations shows that the upper limit of $P/Y$ is 2.85 for the case of tanker and 2.56 for that of LPG ship as shown in the Table 4.

According to the inquiry of the Ministry of Labor, average monthly pay for full–time workers of large factories is 412.3 thousands yen[16)]. Also, total welfare expense is 51,669 yen per month. Calculating from these amounts, the annual labor cost $Y$ is 5.568 millions yen.

If annual labor cost $Y$ is this amount and a safely assumed value of $P/Y$ is 2.5, the upper limit of the investment becomes 14 millions yen.

According to the inquiry of the Japan Industrial Robot Association, Japanese shipyards installed 39 sets of numerically controlled robots in total during 5 years from 1984 to 1988. The total investment was 507 million yen[17)]. Thus, the average amount is 13 million yen a set.

The upper limit of the investment is over this average amount. Considering additional indirect labor costs for manual welding and the extended operating time for automatic welding, robots have the advantage. Thus, the use of the robots becomes more profitable because the practical upper limit of the investment rise more.

The economic study above shows that the use of robots on non–parallel parts of hull are profitable because teaching time becomes small with the application of the CAM system. In other words, the CAM system has the effect that the applicable range of the robots expands to non–parallel parts of hull.

The average time for programming works becomes three times, i.e. 3 hours, unless the CAM system is available. Because the programming works consume equal or longer time than the welding time, robots lose profitability unless there are many identically shaped workpieces.

## 4. Conclusion

On the application of arc–welding robots to hull sub–assemblies, it is necessary to extend the applicable range toward the non–parallel parts. The authors have developed the CAM system for these robots.

Table 4 Effects of teaching job

| Type of ship<br>Name of workpiece | Tanker<br>Bottom transverse | LPG<br>Transverse at bilge |
|---|---|---|
| Welding length (m) | 67.8 | 39.8 |
| Playback time (min.)<br>Programming time (min.)<br>Direct teaching (min.) | 226<br>60<br>18 | 132<br>60<br>18 |
| Manual welding (min.) | 301 | 213 |
| $F \times G$<br>$r_P$<br>$r_D$<br>$P/Y$ | 1.33<br>0.265<br>0.080<br>2.85 | 1.61<br>0.454<br>0.136<br>2.56 |

Each workpiece for non–parallel parts has different shape. Also, form of work, working order and welding position are changed very often. Therefore, the CAM system must attain shorter programming time and flexible functions for various alterations also. The features of the system are as follows.

1) The CAM system gives task level programs to the robot controller. As the CAM system generates each motion, the robot can freely run even if an alteration of producing order causes the change of setting.
2) The system uses graphic displays conversationally. Also, it shows operational guide and choices of instructions on the displays. These functions are effective for the reduction of programming time and easy learning of operation.
3) The system is effective for the reduction of programming time and the prevention of errors because it uses the data base for the hull.
4) The system is equipped with a personal computer at a control room adjacent to a welding shop. It is useful for editing programs, their revisions and their management, which leads to the attainment of flexible features.

The CAM system reduces programming time to 1/3.

The authors estimated the economy of the robots based on the records of the actual welding time. This estimate shows that robots are surely applicable even if every workpiece is different. Therefore, the range of the arc–welding robot applicable to hull sub–assemblies can be extended to non–parallel parts of hull.

The authors have accomplished the means of automatic welding at the hull sub–assembly process as the result of this development.

## References


1) T. Yoshitomi, M. Yamamoto, et al.: On the Development and the Applications of the Arc Welding Robots for Hull Sub–assemblies (in Japanese); Journal of the Society of Naval Architects of Japan, Vol. 166, December 1989.
2) T. Ogawa: IHI's Offline Teaching System and its Application to the Joint Welding Robot (in Japanese); "Robot", Journal of Japan Industrial Robot Association. No. 66, January 1989.
3) Y. Ohtuka, et al.: Development of Robot Welding System for Steel Structure (in Japanese); Kawasaki Juko Giho, No. 94, December 1986.
4) T. Yamakawa et al.: Development of Robot Welding System for Steel Structure (Second Report) –Application to Panel Members in Bridge– (in Japanese); Kawasaki Juko Giho. No. 101, December 1988.
5) R. Adachi et al.: Application of Intelligent Robot Arc–Welding System to Large–sized Steel Construction (in Japanese); Hitachi Zosen Giho, Vol. 46, No. 6, December 1985.
6) Y. Higo, H.Yokouchi: Development and Applications of Arc Welding Robot on Shipbuilding Industries (in Japanese), Weld Engineering, Vol. 33, No. 12, December 1985.
7) K. Noguchi, E. Tokumaru, et al.: A Computer–Based Plate Generation System for Shipyard –"SHIP" system– (in Japanese), Mitsubishi Juko Giho, Vol. 8, No. 6. November 1971.
8) Japan Industrial Robot Association: Report of Research and Study on Standardization of Robot Language (in Japanese), Japan Industrial Robot Association, March 1989.
9) Japan Industrial Robot Association: Report of Research and Study on Standardization of Robot Language in 1982 (in Japanese); Japan Industrial Robot Association, March 1983.
10) Y. Kubota: Language System on PUMA Series (in Japanese); Automation, Vol. 27, No. 4, April 1982.
11) Y. Maruyama: Language System on RW Series (in Japanese), Automation, Vol. 27, No. 4, April 1982.
12) H. Ando, S. Mori, Y. Shiga: The

MELFA–RA2 Industrial Robot Controller and its Applications (in Japanese), Mitsubishi Denki Giho, Vol. 59, No. 10, October 1985.

13) T. Yoshitomi: Study on Macro Program on Arc Welding Robot with Robot Language for Hull Sub–assemblies (in Japanese); Transactions of the West–Japan Society of Naval Architects, No. 78, August 1989.

14) T. Yoshitomi, M. Yamamoto: The Unmanned Operation of the Arc Welding Robot with Robot Language for Hull Sub–assemblies (in Japanese); Transactions of the West–Japan Society of Naval Architects, No. 79, March 1990.

15) Industrial Engineering Research Committee: Text Book of Industrial Engineering, First Edition, Japan Science and Engineering Federation (in Japanese), 1970.

16) Toyo Keizai Inc.: Toyo Keizai Data Book 1989 (in Japanese), Toyo Keizai Inc., 1989

17) Japan Industrial Robot Association: Industrial Robot Hand Book [1990 Edition] (in Japanese); Japan Industrial Robot Association, 1989.