하계학술대회
논문86-3-3

A Practical Method to Adopt MAP in Industrial Robot Controller

Ikuo Nagamatsu

Yaskawa Electric
Manufacturing Co., Ltd.
Kitakyushu, Fukuoka,
JAPAN

Abstract

The ultimate goal of an industrial robot is to make full use of its real ability to communicate with any given intelligent device, and to do so independently of hardware, architecture and languages. This paper describes the necessary functions of a robot used in an integrated manufacturing system, and the basic philosophy of organization as applied to the robot controller.

An example of a machine vision system called MYVIS is reviewed in relation to MAP and LAN in a practical cell application.

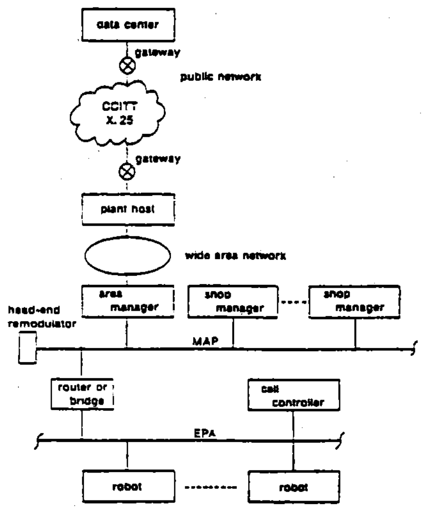

Figure 1 System topology

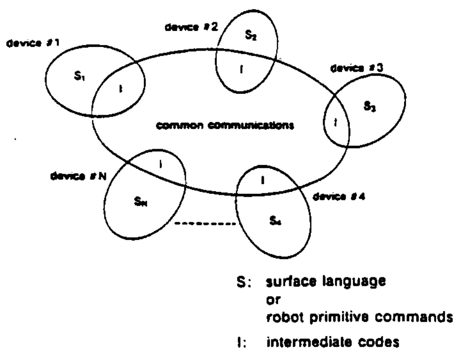

Figure 2 Conception of intermediate codes

INTRODUCTION

Recently the micro-electronic technology has made a remarkable progress in factory automation which has expanded to all levels of manufacturing. Because of a high degree of system integration several problems are encountered such as increased complexity, rise in cost or development time, interoperability, and maintenance.

To this end NBS (The National Bureau of Standards) is developing a five layered conceptual manufacturing system architecture much like the OSI model made by ISO (International Organization of Standards) to solve these problems by defining functional modules in each layer and also by standardizing communications using a horizontal or vertical concept.

Within this scheme the industrial robot (robot) is used at the lowest layer of a manufacturing system. A robot performs the basic functions of : material handling, welding, and inspection necessary for sequencing batch jobs in a total production schedule. It also is an element in a cell, which offers information on the progress of production and gives any status as is required by the host system. Therefore it becomes necessary for a robot to support functions that are powerful in their application for control of a cell ; and its factory wide standardized communications network.

The technological demands in manufacturing can be satisfied by considering MAP (Manufacturing Automation Protocol) and LAN (Local Area Network) which are now being vigorously developed; thus this article describes the required functions for a robot, which are used in an integrated manufacturing system, and describes the basic philosophy of organization as applied to the robot controller. Then an example of YASKAWA's machine vision system called MYVIS will be reviewed in relation to MAP and LAN in a practical cell application.

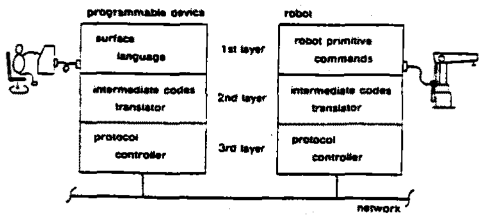

Figure 3 Diagram of communication through intermediate codes

THE INDUSTRIAL ROBOT IN AN INTEGRATED MANUFACTURING SYSTEM

At first an introduction on NBS architecture will be given. It has a hierachical structure composed of five layers. Each layer is given in order of size such as a facility, shop, cell, work station, and equipment.

One facility supports the administrative functions including estimating, accounting and invoicing, and manages long range schedules, and identifies production resource requirements and excess production capability.
The shop is responsible for the real time management of jobs and resources. The shop manages the scheduling of jobs, grouping of parts, allocation of tools, and marking. The cell controls the sequencing of batch jobs of similar parts or subassemblies and order of material handling or calibration. The work station directs and co-ordinates the grouping of equipment. The equipment performs the basic low level functions of materials processing, cleaning, and transportion.
A robot belongs to this lowest layer.

This concept is for small batch manufacturing systems. Probably there is a data center that manages all factories in one more higher layers. MAP and its subset EPA (Enchanced Performance Architecture) with the mames of the functions to each device along with the topology of all systems are shown in Figure 1. The data center manages all factories and the plant host manages production scheduling. The area manager manages communications with shops and distribution of production data; and the shop manager is responsible for task management. The cell controller controls the sequence of jobs. A robot in the lowest layer is controlled by a cell controller and its communication is almost within the cell. However it needs to communicate outside the cell to realize some functions such as the down-loading for robot programs from devices in the higher layers. This is the reason why a robot has to support dual architectures in its communication protocol.

Now we can point out the following three points as being the most important elements for a robot in a total system as mentioned above.

Communications with a shop manager or cell controller using a common language.

- Cell functions.
- Functions that offer information required by a shop manager.

In general, integrated systems are composed of many kinds of devices provided from various vendors. These devices have their own languages or interfaces, so that it is very difficult to connect each other and to maintain them.
In order to simplify these communication links it is mandatory that a common language be used.

It is proposed that from the view point of a robot maker the solution to this problem is in the concept of intermediate codes as shown in Figure 2. Intermediate codes are a group of codes that have a common format and semantics and they can connect any language within the programmable device that is used to control a robot having primitive commands to control an actuator.

In order to put into effect the concept of intermediate codes, it is convenient to select a three layered communication architecture. Figure 3 shows a type of communication using intermediate codes. The first layer is composed of languages that are dependent on the devices, such as FORTRAN, PASCAL, BASIC, or on their own language in any programmable device. Any language can be applied. In the case of a robot controller it is probably easier to interpret binary codes. The second layer is responsible for the interpretation of intermediate codes. A surface language is translated into intermediate codes and intermediate codes are translated into robot primitive commands in this layer. The lowest third layer controls the communication protocol and sends or receives intermediate codes through the network line.

Each layer is classified according to its type of function; therefore this architecture is independent from all kinds of devices and their languages, and can facilitate simple forms of communications with each other.

If we consider this third layer as layer 1-6 in the OSI model and in the same way we consider this second layer as layer 7, MMFS (Manufacturing Message Format Standards), we can say that this architecture is conceptionally conformable to ISO architecture. In addition, this architecture using a layered functional module can work no matter what changes take place in trends for standardization.

The work cell functions are also worthy of mention. In general these functions allow a cell controller to operate a robot. This robot is directory controlled by this cell controller because a robot is an actuator in the system. Consideration is given for functions that a cell controller requires.

At first, various models are assumed for cells in a manufacturing system and individual functions are selected in detail and classified into some large groups. Then functions that the cell controller has to execute are divided into the following requirements.

. Ability to lock the panel operation on a robot controller for safety. A robot generally has operational capability such as teaching of robot motions, calibration, and maintainance in the stand alone mode. When a robot is controlled as a part of the system, these operational functions on a robot control panel must be overridden.

. Ability to operate a robot using its own functions in the stand-alone mode. Operations can be executed using a remote command station instead of using a robot controller; However the robot controller is capable of independently controlling itself.

. Ability to acquire all robot data and to make changes. All informations and status of a robot in a cell must be obtainable. For example there is an error status, production data, current program, and position data of a robot. Some of these informations may be changed in case of necessity.

. Ability to adapt to the working environment. For example used tools are initially set by a shop manager in the upper layer, but some of them are changed during a job. In another case a small cell for material handling that consists of an AHC (Automatic Hand Changer) robot with the three hands requires the mechanical parameters to control the tip of a tool precisely. Used tools and their mechanical parameters are initially set while the time to use them is decided at the work cell level.

This decision is made according to the job sequences in a cell controller.

. Ability to control all robot motions. A robot is required to move to the desired absolute or relative position in a co-ordinate space and to control speed. Generally a robot motion is described as a program before operation. However it is often required to control each robot motion in real-time based on information given from sensors.

A robot has only to satisfy the functions mentioned above and to act in response to the controller's command.

A description of functions that accept information from the shop manager must be considered. A shop manager directs the order of jobs to a cell controller. Though this management is fairly wide, there are only two important items for a robot. The items are the following.

. Ability to set the working environment. (The same functions are required of a cell controller).

. Ability to transmit robot programs. Robot programs are required to be up-loaded or down-loaded from/to a robot controller. In general two different ways of programming are used to effectively control a robot. That is programming a robot program using a teach pendant or console terminal; and the other way is to use off-line teaching. In all cases a robot program is required by a shop manager to be up-loaded and managed, and to be down-loaded to a robot controller.

A robot has only to correspond to the demands by a shop manager.

It can be said that a robot works very well as an actuator in integrated manufacturing system, if it can satisfy all functions mentioned above.

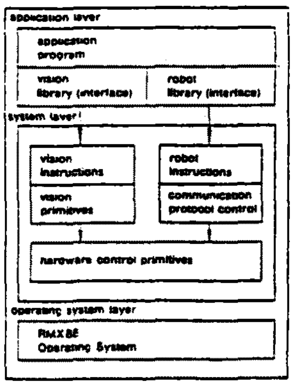

Figure 4 Block diagram of MYVIS

AN EXAMPLE OF A ROBOT CONTROL SYSTEM USING MY-PASCAL

An example of a robot control system that complies with the above requirements can be given here. The YASNAC RX (RX) controller manages the MOTOMAN robot and has a MYVIS machine vision system which uses PASCAL which is widely used in many computers. MYVIS is designed for PTP (Point To Point) control at the first step so that only two lower layers in the OSI model are supported at the third layer shown in Figure 3. They are RS232C and BSC (Binary Synchronous Communication) respectively and are the most popular protocol used in Japan. Of course the discussion below is still applicable even if the device, language, or communication protocol is changed.

Here an easy to understand explanation looking from within the cell controller is given. Figure 4 shows the block diagram of the MYVIS software. The MY-PASCAL language is used as a subroutine of PASCAL and is used to control the motion of the robot.

Robot commands are classified into five large groups :

1. System Control Function which are used to read and write the following system information :
   . Parameters for users to use a robot conveniently
   . Mechanical constants for tools
   . Internal toggle switch settings
   . Control panel interlocks

   This function uses the following commands :

   rget_param rset_param rget_mc
   rset_mc rget_tggl rset_tggl
   rlock_panel

2. Status Reading Function which are used to read the following status information:
   . Warning, error, and alarm codes
   . Relay's condition connections
   . H-counter contents
   . Position
   . Production information
   . Control panel display
   . Current program status

   This function uses the following commands :

   rget_status rget_relay_status rget_h_counter
   rget_pos rget_p_pos rget_pro_time
   rget_job_seq rget_panel_status

3. Operation Functions which are used to manage the following operating conditions:

. Mode/cycle

. Start/stop

. Reset/cancel

. Programs selecting

This function uses the following commands :

rset_mode rset_cycle rstart
rstart_job rhold rreset
rcancel rset_job_seq rset_job_to_prog
rsel_prog

4. Primitive Motion Function which are used to control the following robot motions :

. Movements to absolute or relative desired positions.

. Scan external I/O relays and time delays.

. Play back a robot program.

. Set welding functions.

This function uses the following commands :

rmov rmovf rmovj
rmovjf rmovs rmovfs
rpmov rpmovf rpmovj
rpmovjf rpmovs rpmovfs
rimov rimovf rtimer
rsignal_on rsignal_off rpulse
rsense rscan_on rscan_off
rplay_job rarc_on rarc_off
rwait rweld_voltage
rweld_current

5. Transmission Initialization Functions are used to manage transmission functions :

. Read/write transmission parameters.

. Initialize input_output devices.

This function uses the following commands :

rset_com_param rget_com_param rinit_com
rreset_com rconfig

Moreover MYVIS has a simple architecture that has only two lower layers, which it serves as a system host. These functions are divided as follows.

1. Robot program Transmission Functions which are used to transmit robot programs :

. Up-load/down-load robot programs in the transparent format.

. Up-load/down-load robot programs in the non-transparent format.

This function uses the following commands :

rsave_int_job rsave_lst_job rload_int_job
rload_int_job

2. Program Utility

. Delete robot programs.

. Look up robot program lists.

. Send messages.

This function uses the following commands :

rdelete_job rget_job_dir rsearch_job_dir rsend_message

These functions which are described in the intermediate codes are transmitted to the RX and executed there. RX supports an interpreter of intermediate codes that are directed by them.

The PASCAL language is used at the remote station. This language is translated into the language used by RX controller. There are two intermediate codes, one to control commands accepted by RX internal program; and the other to display commands in the RX that are to be manually carried out by the robot operator and other miscellaneous commands. A part of intermediate codes in transmission formats is shown in Figure 5. SOH, STX, and ETX are characters for communication control defined in BSC. The kind of command in text are indicated by a heading and intermediate codes are written in a text.

These intermediate codes are selected for the following reasons.

. Ease of understanding for users. Codes are the same as the formats displayed on the CRT of a robot controller.

. Quick operation. In any case codes are operated in only one algorithm used in a robot controller, and these codes are easily converted into internal formats.

All formats of these codes are shown to the public and are independent of the device's hardware or language. Therefore any programmable device can support them with no special functions required. We have also completed another system based on the same philosophy using a PC-9801 (NEC Corp.) personal computer and N88 BASIC language which is available and widely used in Japan.

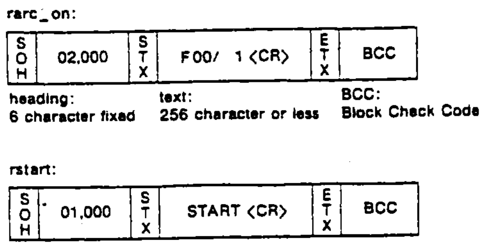

Figure 5 Formats of intermediate codes

AN EXAMPLE OF A SMALL CELL USING MYVIS

This cell consists of two robots, MYVIS, a sequence controller, a color graphic controller, a conveyor for transport, a part feeder, a press as a tool to assemble, a grease supplier, a heater, and a turning table for robot grippers. In this cell two different kinds of the links that drive the arms of Yaskawa's Motoman L10W robot (Carrying Weight 10 kg) are assembled. One is A-Link comprising of six parts. Another is B-Link comprising of seventeen parts. These parts are shown in Figure 6.

The following is an outline of the work done in a cell. The main parts are supplied at random on the conveyor. The MYVIS machine vision system recognize these parts using a camera and facilitates the sorting and positioning processes. If it is A-Link, robot #1 carries it on the conveyor to the press station and robot #2 carries small parts such as bearings, collars, and bearing covers to the press station. Then two robots assemble A-Link in co-operation with each other. In the case of B-Link, its work is a little more complicated, but similar to the one for A-Link.

The topology of the cell is conceptually shown in Figure 7. But the hardware consists of RS232C PTP controls and discrete input/output interfaces, so the mechanical concept appears different to the network shown in Figure 7. However it is believed that functions which each device divides can be discussed excluding the physical topology because of the foregoing layered functional modules.

Now, we describe the organization of this cell. PC-9801 as a system host sets the environment for the cell controller to execute job orders. The following is a list of this work :

- Management of parts and job orders.
- Down-loading of robot programs to the related robots.
- Setting of mechanical parameters on grippers to the related robots.
- Informing the MYVIS cell controller of a job order.
- Deleting robot programs when the job is completed.

On the other hand MYVIS as a cell controller controls a sequence of jobs. The following items reveals this work :

- Movement of a robot to absolute or relative desired position directly based on visual information.

- Starting of robot programs.
- Monitoring of the process of work.
- Acquisition of the current position of a robot.
- Display the process of work on a color graphic controller.
- Manage a sequence controller that controls auxiliary equipment such as a conveyor, a part feeder, and etc.

These functions are realized by the following robot commands in MY-PASCAL.

rload_int_job	rset_mc	rset_param
rstart	rget_status	rget_pro_time
rmov	rmovj	rimov
rget_pos	rsense	rsignal_on
rsignal_off	rpulse	

Each robot passively obeys commands given by the MYVIS cell controller.

This example of a small cell readily demonstrates the ease by which the foregoing selected functions may be put into effect and shows a clear ordered arrangement that enables a flexible system to be systematically defined. In this example a system host on the backbone network gives an initial working environment and a cell controller on the subnetwork controls time critical sequences of a job. When referring to Figure 1, one can see that the backbone network is probably equal to the full MAP in its work and the subnetwork also is equal to EPA within this network.

If manufacturing systems are to be more integrated then it is more important that a common media, a common language, and a hierachical architecture which defines the functions of devices in each layer are applied in spite of the method of realizing them.

YASKAWA will use this concept along with MAP which offers more flexibility to the public.

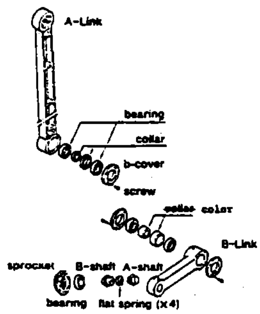

Figure 6 Parts of Motoman L10W robot

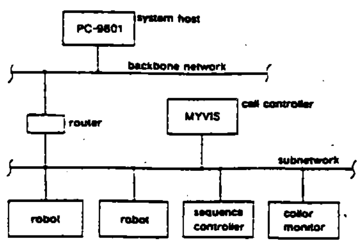

Figure 7 Conception of small cell for link assembly

CONCLUSION

Consideration was given to the objective of MAP as a basic building block for CIM (Computer Integrated Manufacturing) in providing suitable standardization. It is thought that the duty of a robot maker is to offer a system that meets the aim of MAP. The contents introduced here show an aspect of this effort. Here required robot functions were selected in an integrated manufacturing system and a description of the organizational philosophy. The MYVIS machine vision system was introduced in a practical example.

The main goal under study was the establishment of a robot that can make full use of its real ability and do so independently of different types of languages that might be used to effect its communication with any sort of intelligent device. The application of a hierachical architecture using intermediate codes as an intermediacy between common languages with definitions for functions was presented as a practical solution. A cell controller was based on standardized communication protocols. It is believed that widespread use of MAP will become eminent in the near future and this attempt is an effective step to adopt MAP.

At the comming Robots '86 Yaskawa will exhibit a small system based on the ideas of this paper in order to demonstrate integrated manufacturing capability.

ACKNOWLEDGEMENT

The authors of this paper would like to express their appreciation for the Robots '86 Conference Committee's, kind consideration; and also for Mr. Peter Brock Davis's help and assistance.

BIBLIOGRAPHY

(1) C. Mclean et al. "A computer architecture for small-batch manufacturing", IEEE Spectrum, vol.20, no.5, 59-64, (MAY 1983)

(2) I. Nagamatsu et al. "An Approach to Software Organization for Control of Industrial Robots", Journal of ROBOTIC SYSTEMS, JOHN WILEY & SONS, vol.2, no.3, 265-276, (FALL 1985)

(3) T. Arai, "Intermediate Language for Robot Controllers", 4th International Conference on Assembly Automation, 55-66, (1983)

(4) M. Inoue, "Robot and robot language", Robot Engineering and Application, IECE, Japan, 100-125, (1984)