# A General Robot Application Platform for Machine Tending

Du Li, Yongzhi Huang
ABB Corporate Research Center
Robotics Department
Shanghai, China
Mike-Du.Li@cn.abb.com, Clement-Yongzhi.Huang@cn.abb.com

***Abstract*—a general robot application platform in machine tending area is presented in this paper. This new robot application platform is mainly about the common machine tending application architecture, which provides a general solution for easy robot programming and operation, for robot and machine integration and for the interaction between online and offline programming.**

***Keywords*—general platform, robot programming, operation, integration, machine tending**

## I. INTRODUCTION

With the increasing demand for industry robots from general industry such as plastics, foundry, machine tool[1], the requirements for robot application in machine tending area including easy robot programming and operation, robot and machine integration and interaction between online and offline programming are emerging and desired by machine builders, robot integrators and end customers. Although there is specific machine tending application software [2] in some industry area such as plastics, but a type of general application architecture that will meet the above requirements and be applicable for different industry area is still lacking. So it should be great significant to fulfill these requirements for robot application development in machine tending area. Under this situation this paper proposes a general robot application platform in machine tending area, which is mainly about the common application architecture that will be applicable for different types of machine tending application e.g. injection mould machine, die-casting machine, etc, and based on this architecture the general solution for easy robot programming and operation, robot and machine integration and the interaction between online and offline programming is provided.

## II. OVERALL ARCHITECTURE

### A. Overview

One of the largest uses of robotics is the machine tending [3]. Machine tending process mainly aims at unloading produced parts from a machine then putting it onto the output workstation. This process also covers loading insert into a machine and some post processing through the cooperation with the machine and surrounding workstations. The kernel for the general machine tending application platform is that it will not only provide the ability to easily program and operate a robot covering a wide variety of typical cases through a graphical user interface but also keep the flexibility for customization so as to meet the complex case requirements in machine tending area.

### B. Architecture

The overall architecture of this platform is outlined in Fig.1 as below. It is composed of the controller application, the teach pendant application for online programming and operation and the PC application for offline programming. The machine HMI (human machine interface) can communicate with the robot controller application through a specific interface defined in the controller application.

The controller application is the core of the whole machine tending application architecture. It is the base of robot programming and operation. It provides the interface for robot and machine integration and gives support to the interaction between online programming and offline programming.

The teach pendant application is the user interface for robot programming and operation. It provides the graphical user interface to create or modify robot program through the specific wizard and operate robot in production.

The PC application is an offline programming tool. It provides the simulation environment for offline robot programming and machine tending process simulation. It also supports the interaction between online and offline programming.

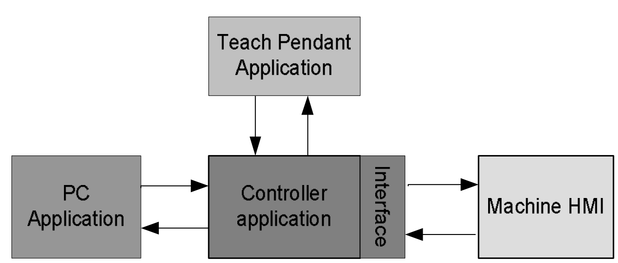


Figure 1.General Application Platform Architecture

## III. GENERAL APPLICATION CONCEPT

Machine tending process will be various depending on process specific requirements e.g. with insert or not, post-processing level. These various processes can be regarded as different work flows such as production flow with or without insert, warm-up flow, quality check flow, and definitely they will have the different requirements on robot scheduling logic and motion logic for the robot application.

To cover the different process requirements and provide a clear and effective description of the machine tending process, some general application concepts are abstracted from the machine tending process and applied into the process application description. These concepts are as follows.

- Cell: A cell represents the set of a robot and all the workstations in the real machine tending application environment. The minimum cell must consist of a robot and a machine.
- Station: A station represents a real workstation in the real machine tending application environment. It can be regarded as a work place where the robot picks or leaves a part, loads an insert or does some kind of processing for the part e.g. injection mould machine, conveyor, etc.
- Point: A point represents a position that robot will visit.
- Path: A path represents a robot visit track within a station or between different stations, which contains a series of points.
- Tool: A tool represents a physical tool held by a robot.
- Cycle: A cycle represents a repeated work flow during the machine tending process in which the robot will tend the stations sequentially according to the scheduling logic.
- Tool method: A tool method is a method used to operate one or more tools at a point during a robot cycle.
- Link procedure: A link procedure is a procedure triggered at a point and used to execute some tasks e.g. waiting for some specific signals.

## IV. ROBOT PROGRAMMING AND OPERATION

A general solution for easy robot programming and operation in machine tending area is the key of the whole application platform. For this solution the key factors include covering various machine tending process requirements, easy of use and flexibility. Based on the above factors a proposed solution is presented as follows.

### *A. Process requirement coverage*

First of all the general application platform should cover different machine tending process requirements. For this objective the structure of the controller application and the teach pendant application aims at providing the ability to meet various process requirements and it is outlined in Fig.2 as below.

The controller application is built in a module structured way. The common modules such as kernel, cell, station, tool, construct the main body of the controller application. The above general application concepts are implemented through these modules.

These modules of the controller application aim not at trying to provide detailed and specific implementation so as to meet different process requirements but focus on constructing a common framework to create specific robot program to cover different process requirements in machine tending area. So these modules only contain necessary frames for robot programming and the implementation details e.g. scheduling logic, motion logic will be dynamically implemented based on the common framework during robot programming.

The teach pendant application contains some programming views and operation views. Based on the controller application module structure the programming views in the teach pendant application provide the approach to create a robot program through the wizard on the graphical user interface so that robot programmer can configure the implementation details according to the process specific requirements.

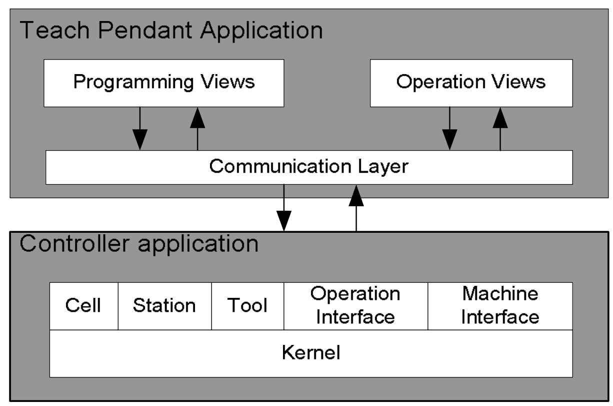


Figure 2.Controller and Teach Pendant Application Structure

### *B. Easy programming and operating*

How to shield the user from the complex details of a full robot application and implement easy robot programming and operation is significant for improving the robot usability in machine tending area.

*1) Easy robot programming:* Based on the above structure in Fig.2 the proposed solution for easy robot programming is described as follows.

For the solution mentioned here robot programming will aim at smart and effortless configuration of machine tending application without the immediate need for functional and conceptual robot programming. This application oriented configuration concept will solve a wide variety of application cases without the need of direct robot language programming.

As the base of robot programming the controller application is built in a module structured way and provides the necessary module skeletons which will be used to create machine tending process specific modules during robot programming. Fig.3 is the module static structure diagram. The basic modules are described as follows.

- Kernel: The kernel module is the control centre of the whole controller application. It manages the program execution, cycle scheduling, station scheduling, error handling, status monitoring of all the stations and tools through the calling of relevant routines inside it and the specific interface routines defined in the other modules such as station execution routine in a station module. So the kernel module has the dependency on all the other modules.
- Cell: The cell module contains the cell related information which will be used by the kernel module e.g. path between workstations.
- Station: A station module is used to describe a real workstation object. It consists of robot paths and path scheduling information in a workstation, execution interfaces called by the kernel module, interrupts and its response routines, tool methods, link procedures, etc. It has the dependency on the tool modules.
- Tool: A tool module is used to describe a physical tool object. It consists of tool data, tool operation interfaces called by a station module and the kernel module.
- Operation interface: The operation interface module provides the robot operation interfaces called by the teach pendant application in production.
- Machine Interface: The machine interface module defines related I/O and command interfaces for robot and machine integration.

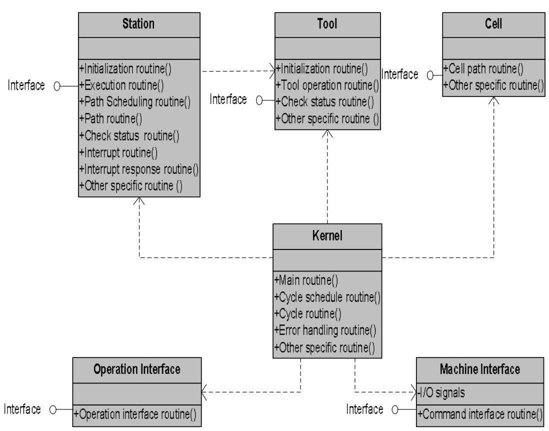


Figure 3.Controller Application Module Structure

Based on the application oriented configuration concept and the structured modules in the controller application, the teach pendant application provides a series of programming views as the graphical user interface for robot programming. The procedures of creating or modifying a program such as process work flow definition, motion path teaching and tool configuration are organized and implemented through a programming wizard that will guide the user through the process of creating a program. The steps of the wizard are as follows.

- Cell configuration: To define cell components and configure layout.
- Cycle configuration: To define cycle and configure cycle scheduling logic such as sequence logic, branch logic through a kind of tree view.
- Tool configuration: To set tool data and handle signal mapping.
- Station configuration: To define and teach robot path within a station or between different stations, configure motion logic e.g. attaching a tool method or link procedure to a point, configure path scheduling logic, test path, etc.

To implement the mapping between the controller application and the teach pendant application, the controller application not only provides robot program framework but also defines some simple naming convention for specific variables, constants, routines that will be parsed by the teach pendant application. So the teach pendant application can create or modify specific program information based on the module structure and naming convention. This will implement the mapping between the controller application and the teach pendant application.

*2) Easy robot operating:* Regarding easy robot operation the proposed solution is as follows.

- Controller application provides the operation interface shown in Fig.2, which will be called by the teach pendant application.
- Teach pendant application provides a series of operation views shown in Fig.2 on the teach pendant, which will be used for the robot operation.
- Comprehensive operation functionalities: Through the various operation views the following operation are available: robot program handling such as load/unload program, operation in production such as start/stop cycle, home run, monitoring in production such as production statistics information, signal status.

### *C. Flexibility*

On the one hand the module structured controller application with a common framework of robot program and the tech pendant application with graphical programming wizard provide the ability to easily program and operate robot covering a wide variety of typical cases through the graphical user interface, on the other hand they also keep the flexibility for customization so that the experienced programmer can directly modify the pre-defined modules to create robot program so as to meet the complex case requirements in machine tending area.

The flexibility will be reflected through the following aspects:

- All the modules in the controller application are scalable for customization.

- For complex schedule logic and motion logic requirements in machine tending process, which the teach pendant application can not support to implement it through the graphical user interface, the experienced programmer can do some customization based on the robot program created by the teach pendant application.

## V. Robot and Machine Integration

In machine tending area a kind of technical trend that the operator can control a machine and a robot from one point of control in production i.e. from machine side or robot side is emerging. More and more machine builders and end customers are looking forward to implement such kind of robot and machine integration. Here a proposed solution for robot and machine integration based on the general machine tending application architecture is presented, which aims at controlling a robot and a machine from the machine control panel.

### *A. Robot controller states and process states*

To implement the robot and machine integration so as to control a robot from the machine side, firstly the robot controller states such as motor on/off, auto/manual and process states such as robot program running, robot program stop must be available from the machine side, it is the base of robot and machine integration and as the precondition of sending control commands to a robot from the machine side. The following basic states are needed for the integration control.

*1) Robot controller states:* It means the state of robot controller itself.

- Robot controller mode: Controller operation mode such as auto, manual.
- Robot motor status: Motor power status such as motor on, motor off.
- Robot controller running: Controller running in production mode.
- Robot controller stopped: Controller stopped executing because of a commanded stop or an open safety chain.

*2) Robot process states:* It means the state related to machine tending process.

- Robot control mode: Robot is controlled by local operation or remote operation i.e. local control or remote control
- Robot program running: Robot program is in execution.
- Robot program stop: Robot program execution is stopped.
- Robot home running: Robot is returning to its initial position.
- Robot at home: Robot is at the initial position.
- Robot cycle active: Robot is performing a work flow through cooperating with machine.
- Robot engagement mode: Robot is engaged with a machine or not i.e. run with robot, run without robot.

### *B. Robobt program information*

Robot program information such as program numbers and names, specific program data are useful for robot and machine integration and they should be accessed from the machine side. For instance a machine operator can select corresponding robot program and load it by the program number or name from the machine side.

### *C. Robobt control command for remote control*

To control a robot from the machine side a series of control commands are necessary so that the relevant robot operation requests can be sent to a robot controller from the machine side based on the robot and machine states. The useful control commands are as follows.

- Remote control: To request the robot remote control authority
- Robot motor on/off: To switch the robot motor status
- Robot program load/unload: To load or unload a robot program in a controller
- Robot engagement or disengagement: To engage or disengage a robot with or from a machine
- Robot speed rate adjustment: To set the robot overall speed rate
- Robot cycle start: To start a cycle defined in a robot program
- Robot cycle stop: To stop the current active cycle
- Robot cycle continue: To continue the stopped robot cycle
- Robot home run: To return a robot from the stopped position to its initial position

### *D. Machine interface*

To implement the integration between robot and machine a specific machine interface as shown in Fig.1 is defined in the controller application. It contains two types of interface i.e. I/O interface and command interface. The I/O interface is mainly used to access the robot controller states and process states from the machine side and it also can be used to send robot control command from the machine side. The command interface is used to send robot control command from the machine side and it also can be used to ask for the robot program information. The details are as follows.

*1) State I/O interface:* The robot controller state and process state are reflected through the output signal bits defined in the *I/O* interface. So the machine control system can access the real-time states information from a robot through monitoring these output signals.

*2) Command interface:* The controller application defines a series of parameterized commands as the interface through which the machine can send control commands to a robot controller via ethernet. The command structure are as follows.

- Command number: An unique identity of the command

- Command name: As the description of command functionality e.g. Cycle Start
- Input parameters: A command with one or more input parameters e.g. robot program number
- Output parameters: A command with one or more output parameters e.g. command return code, robot program data.

*3) Command I/O interface:* To expose control commands on I/O signals so that the commands can be executed through I/O signals instead of ethernet, a kind of I/O protocol is defined to implement this approach. This protocol is briefly described as follows.

- Input signals: The command name is represented by the signal name, the command input parameters are represented by the combination of some input signal values.
- Output signals: The command output parameters can be represented by the combination of some output signal values.

## VI. INTERACTION BETWEEN ONLINE AND OFFLINE PROGRAMMING

As shown in Fig.1, the controller application and the teach pendant application build the online programming environment. The PC application is an offline programming tool and it builds the simulation environment for offline robot programming and machine tending process simulation. How to implement the interaction between online programming and offline programming is significant for easy robot programming. The proposed solution is as follows.

### *A. Robot program round load*

The controller application and the PC application are based on the same module architecture as shown in Fig.3 and the corresponding robot programs created online and offline based on this architecture have the same structure. So they can be round loaded between online programming and offline programming and this makes it easy to implement the interaction between online and offline programming.

### *B. Interaction implementation approach*

The interaction between online and offline programming will focus on the following aspects: how to improve online programming efficiency through offline programming, how to test a robot program in the offline environment, how to simulate errors and test error handling.

*1) To improve online programming efficiency:* The PC application can build the simulation environment of machine tending process so that the programmer can quickly offline create a robot program then download it to a real controller and complete it through online tuning. The procedure is below.

- Set up simulation environment in the PC application
- Offline create a program through the PC application
- Download the program to a real controller
- Load the program through the teach pendant application
- Online tune the program through the teach pendant application

*2) To test program easily:* The online robot program testing is time consuming and dependent on the machine and the other peripheral equipments. The PC application provides the simulation environment so that the programmer can upload the program in a controller and easily test it in the offline environment. The procedure is below.

- Online create a robot program through the teach pendant application
- Upload the robot program in a real controller to a PC
- Set up simulation environment in the PC application
- Load the program through the PC application
- Offline test t robot program through the PC application
- Online tune the program based on the offline test results

*3) To simulate errors and test error handling:* For online programming, errors simulation and error handling testing is difficult and limited. But for offline programming through the PC application this is easy . So a programmer can download the program from a real controller to a PC then test error handling through simulating various errors in the offline environment. The procedure is below.

- Online create a robot program through the teach pendant
- Upload the robot program in a real controller to a PC
- Set up simulation environment in the PC application
- Load the program through the PC application
- Offline run the program and simulate errors
- Test error handling and modify program through the PC application

## VII. CONCLUSION

The purpose of this paper is to present a general robot application platform in machine tending area. It builds the common application platform architecture. Based on this architecture, the general solution for easy robot programming and operation, for robot and machine integration and for the interaction between online and offline programming is described in details.

## REFERENCES


[1] M.Tolinski., "Lean Machine-Tending Robotics," Manufacturing Engineering Magazine, Vol. 139, No. 3, September, 2007

[2] A. Liberg, "Instant and Consistent Programming," Plastics Magazine, October, 2006

[3] J. Portelli, "Agile Assembly With Robots," Manufacturing Engineering Magazine, March, 2003