A picture or a thousand words?

Graphic-based robot programming simplifies cell engineering

Jing Chen, Yongzhi Huang, Hui Zhang

Text-based robot programming for machine tending applications requires engineers to have a high level of robotics expertise. Alternative graphical-based programs require less skill but fall short in providing a clear graphical overview of the machine tending cell and robot movement logic. ABB is developing a solution to the problem with a fast and user-friendly graphical-based model for intuitive robot programming.

Machine tending cells are becoming increasingly complex. A typical work cell today consists of many stations – injection molding machine, conveyor, quality check station, scrap area, and so on. With so many stations it is not surprising that the 6-axis robot has become the robot of choice for operating these cells. However, programming a 6-axis robot is much harder than programming a linear robot. Application engineers are usually familiar with the machine tending process, but are less familiar with how best to integrate a robot into the process. A programming model that enables application engineers with minimal robotic skills to easily model the entire work cell and robot-machine interaction is clearly needed.

There are two main programming methods currently in use – manual and automatic. Manual programming involves text-based or graphical systems; automatic programming includes programming by demonstration (PBD). Text-based programming is especially common for industrial robots as it offers flexibility for modeling the robot working environment. But robotics companies use their own BASIC-like robot languages, which limit the usability of text-based programs. Engineers have to learn the robot language of each robot manufacturer to be able to meet the vendor preferences of their customers. Recent research has, therefore, focused on developing general programming languages based on existing languages like C++ and Java. Even so, text-based programming requires the application engineer to possess a high level of robotics expertise.

Graphical-based programming requires much less skill in robotics than the text-based approach, although flexibility is sacrificed for usability. Flowcharts or graphs are popular in the graphical-based programming approach, and many application products with graphical or icon-based programming models are offered by robot suppliers. But they all share the same deficiency in that they lack a clear graphical method that expresses robot movement logic and presents an overview of the work cell system.

With so many stations in the typical machine tending cell it is not surprising that the 6-axis robot has become the robot of choice in machine tending applications.

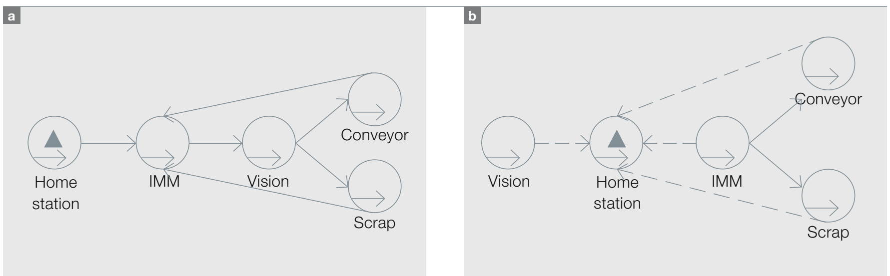
Bringing clarity to programming

A typical machine tending work cell always contains many work stations. The robot is responsible for loading/unloading the produced part and for holding the part while moving between stations in accordance with the I/O (input/output) status of those stations. Work station, robot, robot movement, I/O interaction between robot and machine – these are the physical elements in the machine tending work cell.

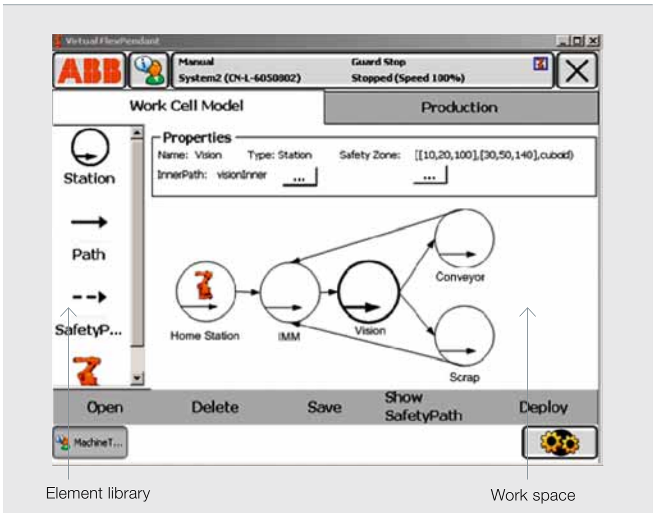
A typical example of robot movement and robot-machine interaction in a machine tending work cell is the following:

1. Robot moves to the position near the injection molding machine (IMM) and waits for the door to open
2. Robot enters the IMM when the door is fully open
3. Robot unloads a produced part from the IMM

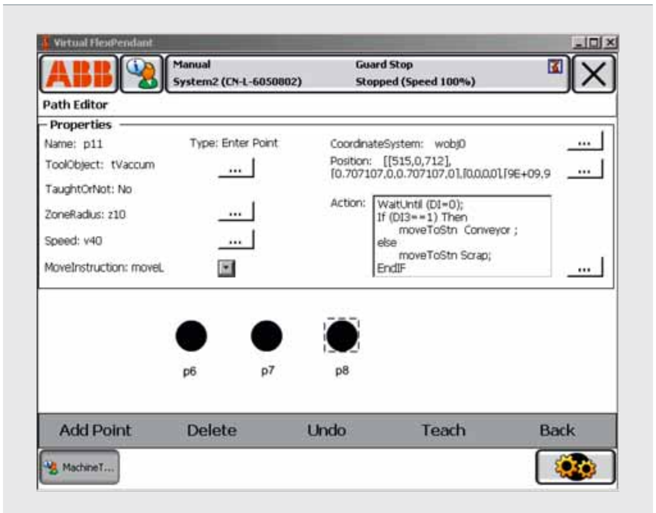
1 Production graphical model a and safety graphical model b



2 Work cell production model

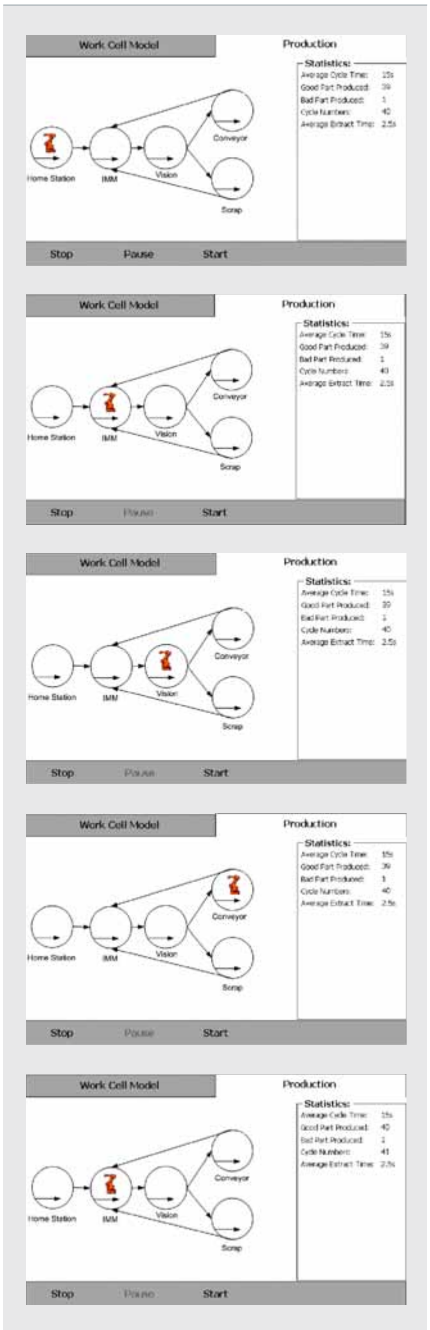


3 Path editor

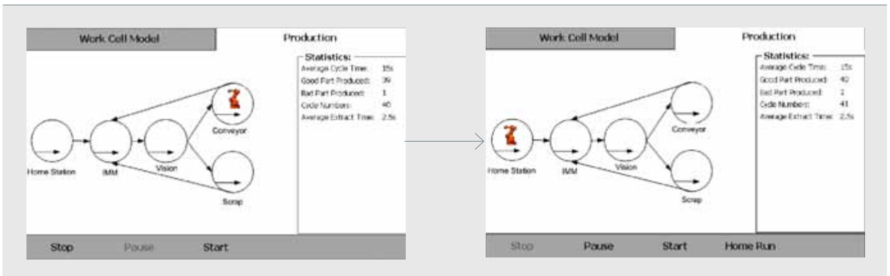


4. Robot exits the machine and signals the IMM to close
5. Robot moves to the vision station and unloads the part in the vision station
6. The vision station signals the robot to check the produced part. If the check suggests the part is a good one, the robot loads the part and moves to the conveyor and on to step 7. If the check suggests the part is a bad one, the robot unloads the part and moves to the scrap station and on to step 8

4 Production cycle



5 Homerun



7. Robot unloads the part onto the conveyor and returns to step 1
8. Robot unloads the part into the scrap station and returns to step 1

If a problem occurs during this sequence, everything in the work cell will stop. An operator then presses a button to move the robot to a safety position, or alternatively jogs the robot to the safety position. If all the continuous production cycles are finished – or if, for example, the robot requires maintenance – the robot will go to the safety position.

Existing graphical-based robot programming models lack a clear graphical method that expresses robot movement logic and presents an overview of the work cell system.

Graphical models of work cell production and the safety position are shown in 1. These models form the basis of a user-friendly graphical interface 2 which includes an element library from which programming elements – station, path, inner path, safety path, etc – are dragged into the work space and configured. Clicking a dragged programming element reveals its properties. Similarly, in the Path Editor function, clicking a point will reveal the point's properties 3. Actions bound with a robot target are triggered when the robot arrives at the robot target. The user can write rapid instructions to implement the branch motion logic. From here it is a short step to pressing the start button so that the robot will start production according to the cycle configured 4.

If the operator presses Stop, then the homerun button will be shown. If the operator presses Home Run, then the robot will return to the home station following the defined safety path, as shown in 5. If there is no safety path defined between the station at which the robot stopped and the home station, the homerun button will be disabled.

These and other aspects of the graphical model outlined above are to form the basis of a new ABB engineering product for machine tending that will enable users to model the entire work cell area easily and more intuitively than ABB's current products, RobotWare Plastics Mould and RobotWare DieCast.

Jing Chen
Yongzhi Huang
Hui Zhang
ABB Corporate Research
Shanghai, China
cathy-jing.chen@cn.abb.com
clement-yongzhi.huang@cn.abb.com
hui.zhang@cn.abb.com