Virtual engineering I

Intuitive offline programming for industrial robots

Liwei Qi, Xingguo Yin, Haipeng Wang, Li Tao

A new ABB robot simulation and offline programming tool has transformed a complex and time-consuming process into an intuitive and speedy exercise that can be performed during a sales visit to the customer. Designed for use by salespersons with limited engineering experience, the product has reduced robot simulation time from a typical eight hours to less than 30 minutes.

Conventional robot simulation and offline programming tools are usually designed for skilled engineers rather than for sales personnel whose engineering background may be limited. From a salesperson's point of view, existing simulation and offline programming tools are difficult to use. They require knowledge of geometric modeling, the ability to generate and adjust robot targets and paths, perform path configuration, position cell objects and so on.

ABB has identified the major bottlenecks that prevent robot simulation and offline programming from being intuitive and easy to use.

The usual solution to this predicament is for an engineer to create a 3-D simulation of a robotics cell on the salesperson's behalf. A simulation takes several hours to create, often a whole day. When the salesperson returns to the customer with the simulation, a certain amount of reengineering is usually necessary. All of this takes time and adds to the complexity of providing the customer with a quick and satisfactory solution.

To simplify this time-consuming process, ABB has identified the major bottlenecks that prevent robot simulation and offline programming from being intuitive and easy to use. The solution – RobotStudio Machine Tending PowerPac (MTPP) – enables salespersons and proposal engineers to quickly set up a simulation of a robotized machine tending cell which they can present to the customer on-site.

Removing the bottlenecks

The five bottlenecks that have prevented intuitive offline programming and simulation are geometric modeling, robot target and path generation, creating robot solution scenarios, configuring robot targets, and determining the optimal position of cell components.

Geometric modeling

The current method for performing geometric modeling is to use either a commercial CAD tool – like Pro/ENGINEER or SolidWorks – or the modeling module of a robot simulation program. There are several drawbacks to both alternatives:

- The competence threshold is high. At least several days' training is required in how to use the modeling tool, as is the necessary engineering experience
- Geometric modeling is a time-consuming process (even for a skilled engineer) that prolongs the initial stage of offline robot programming
- If the geometric model needs to be changed at the solution proposal stage, a new geometric model has to be created.

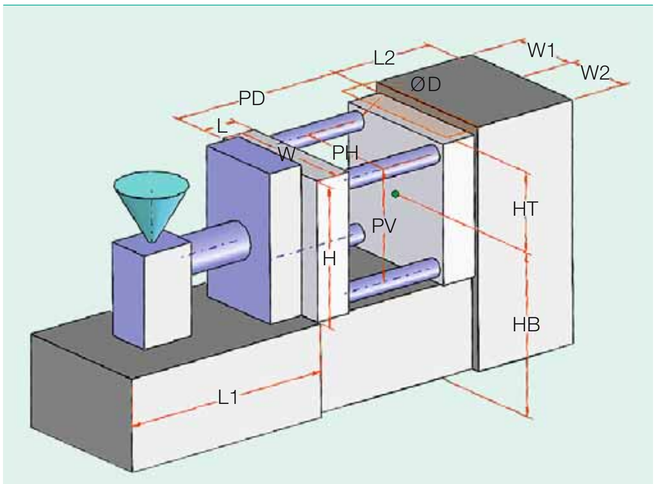
In order to lower the difficulty level and reduce the time spent on geometric modeling, ABB has introduced parametric-driven modeling. By analyzing various robotic applications, the devices for a robotized cell are categorized to form a device library. The geometric model of each device is standardized as one or several typical shapes.

1 shows a model of an injection molding machine with 13 driven parameters. Each parameter has a default value to ensure that models are created correctly. If the size of the model is not as desired, each parameter set can be changed separately or simultaneously to meet requirements. In this way a user can quickly and easily create a model of a complex injection molding machine without previous knowledge of conventional geometric modeling tools. And, by changing one or several driven parameters, a new model of the machine is immediately created. This can be used on any device in a robotics cell for which a 3-D geometrical model is required.

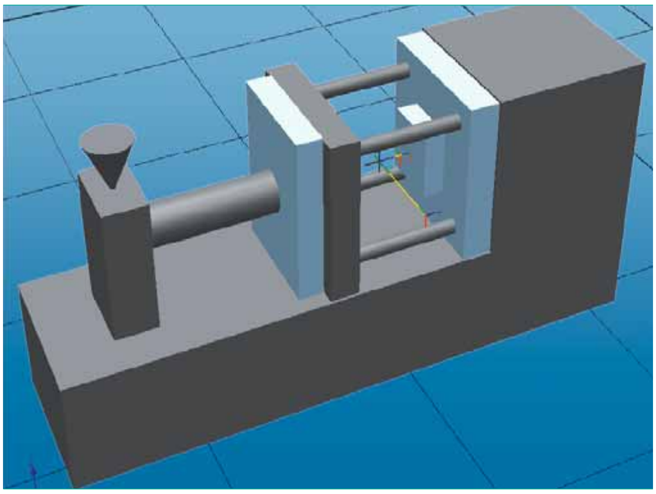
Target and path generation

The usual way to define the robot path is either to jog the robot to the target and record the posture (position and orientation) of the tool center point, or to input the value of the posture directly via the user interface. Either way is inconvenient and involves trial and error regarding the precision of the targets created. When

1 The parametric model of an injection molding machine



2 A geometric model and associated robot targets and paths



Robotics

all the targets have been defined, they then have to be organized into one or several paths. This involves defining the order and the properties of each move instruction – speed, tool data and whether, for instance, it is a linear or joint move. This is also a time-consuming process that demands extensive robotics knowledge and the skills to use a robot offline programming tool.

To reduce the complexity of target and path generation and integrate the procedure with process requirements, ABB has developed an automatic robot target and path generation method based on a parametric-driven model. The method is not limited to machine tending applications.

The machine tending application comprises various devices such as machines and conveyors. These devices can be defined as workstations that include a geometric model and two or more robot targets that form one or more paths. There is a connection between the geometric model and the target posture of the robot. For instance, the picking point for a molded part is always in line with the center point of the molding plate. By formulating algorithms of the relationship between a robot target and the geometric model, target generation becomes automatic. Furthermore, the specific application needs of robot targets and paths can be organized as templates. Each template defines the order of targets in a path, the motion properties of each robot move, and so on. The template enables a robot path that is well matched with application needs to be quickly generated.

Machine Tending PowerPac (MTPP) is fully integrated with ABB's offline programming and simulation platform RobotStudio.

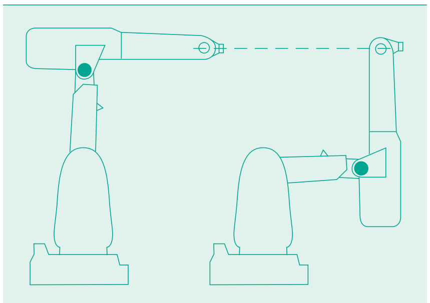
2 shows a geometric model and associated targets and paths. Robot targets are created from the same set of parametric-driven parameters as the model in figure 1. All robot targets are organized into two paths (one in-path and one out-path) in accordance with the process template. If the size of a workstation needs to be changed to accommodate application needs, a new geometric model can be easily created by changing its driven parameters. The targets, which belong to the workstation and are part of the geometric model, are automatically updated.

Scenario-based robot programming

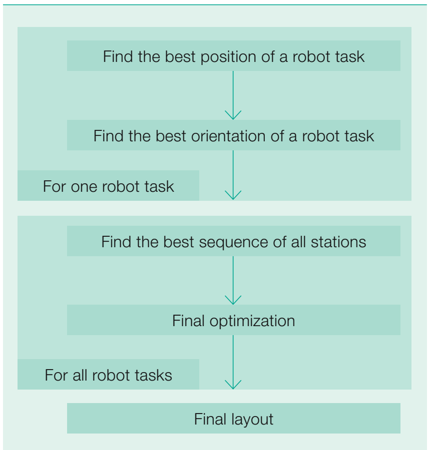
To provide an optimal solution for a customer it is necessary to prepare, compare and evaluate different scenarios. An example of a scenario for an injection molding machine is picking a part from the front of the machine; another example is releasing a part to the machine, then picking a new molded part from the top of the machine. A scenario is made up of robot motion, tool methods and station logic. Robot motion involves robot paths, tool methods involve tool data and tool signal logic, and station logic encapsulates station signal handling.

The conventional way to create different scenarios is to copy the station model and create another set of robot targets and paths manually, as well as the tool methods and station logic settings. Again, this is a time-consuming operation. The ABB solution is scenar-

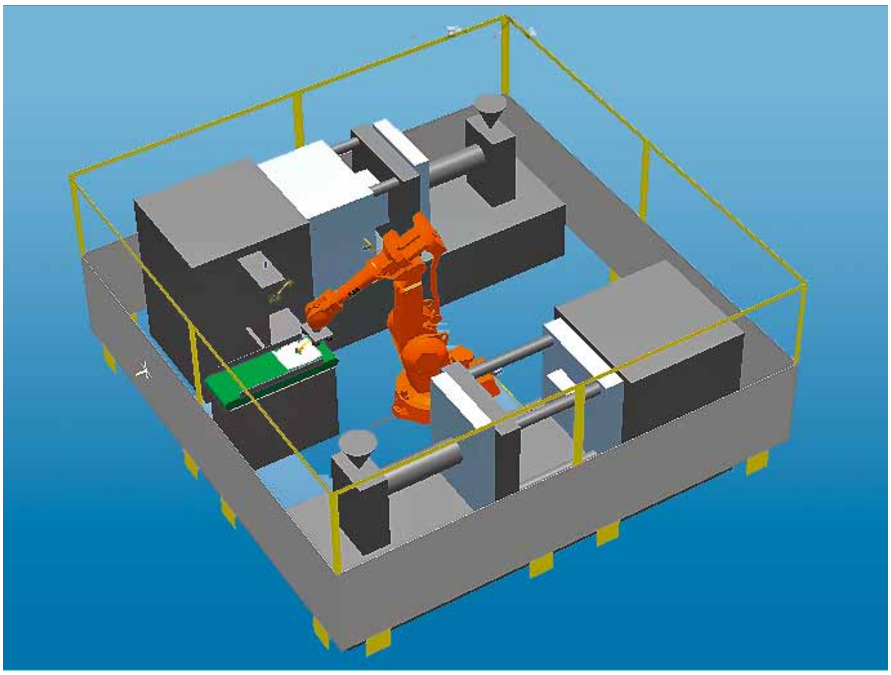
3 Two different arm configurations when a robot approaches a target



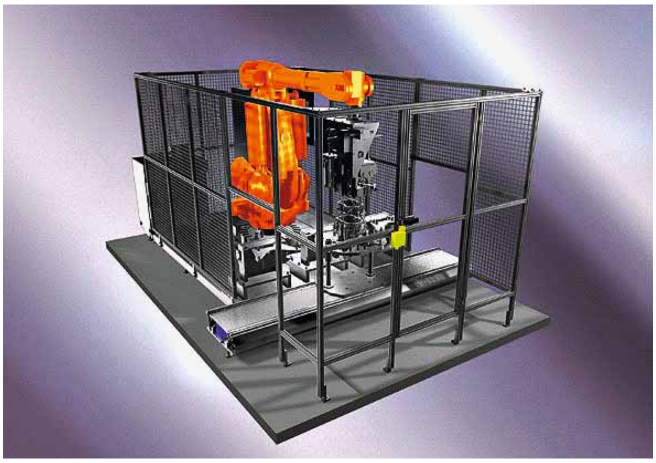
4 Flowchart of optimal cell placement



5 Optimized machine tending cell created in Machine Tending PowerPac



io-based robot programming. The method not only automatically generates the geometric model and robot targets and paths, it also enables a geometric model to be mapped with different sets of targets and paths from different scenarios. This makes it easier and faster to switch and compare different layouts (scenarios) for an application solution.



Target configuration planning

It is usually possible to attain the same robot target in different ways by using different sets of axis angles. These are known as robot configurations. 3 shows two different robot arm configurations that both attain the same robot target. The configuration on the right attains the target by rotating the robot arm backward and rotating axis-1 180 degrees.

The fact that the robot can reach two targets does not mean that it can traverse a path between the two targets. That depends on the rotation limits of the axis when the robot is moving. Since in most cases there is more than one configuration option for a robot target, this raises the issue of how to choose the appropriate configuration data so that the robot traverses the entire path to reach the target. Furthermore, the configuration setting for each robot target should ensure smooth robot motion throughout a cycle (a cycle is the collection of paths that a robot will perform in a robot sequence).

MTTP is already proving of immense benefit to ABB partners and customers in machine tending and other applications worldwide.

In most offline robot programming tools, the configuration data for each robot target is defined manually. This takes time and requires a high degree of knowledge and skill from the user. MTPP, on the other hand, incorporates a configuration planning algorithm for automatic target configuring to ensure smooth robot motion along a cycle.

Placement of cell components

Correct positioning of the robot cell components is essential to make good use of robot capacity and ensure maximum productivity from the cell. This requires a degree of knowledge and experience in robotics that is unusual in a salesperson. A solution for the optimal placement of cell components to achieve minimum robot cycle time is presented on page 90. A brief outline of the solution is given in 4.

Usually, more than one task is performed in a robot cell. Simply repositioning a robot task will not improve robot performance if it only involves a change in rotation of the first axis. Taking this into account, the positioning of several robot tasks can be divided into three stages: individual task positioning, task sorting, and cell optimization.

First, each individual robot task is considered separately to determine the best position and orientation relative to the robot. Second, the best order for the robot to perform the tasks is determined to achieve the minimum cycle time. Successful implementation of the first two stages provides the foundation of the third, cell optimization. At this stage, the position and orientation of the robot tasks can be adjusted simultaneously using a simulated annealing algorithm [1].

Simulations in minutes

Easy offline robot programming as described above is now possible with Machine Tending PowerPac (MTPP), which is fully integrated with ABB's offline programming and simulation platform RobotStudio 5. MTTP was launched in 2007 and is already proving of immense benefit to ABB partners and customers in machine tending and other applications worldwide. Designed primarily for salespersons and proposal engineers, its comprehensive scope of benefits includes the following:

- Quick virtual cell creation and modification
- Previous experience of RobotStudio not necessary
- Estimates cycle time and cell footprint
- Instantly identifies the bottleneck within a production cycle

Reduced cycle time is being achieved by optimizing the layout of the robot cell and is discussed in the article "Virtual engineering II" on page 90 of this issue of *ABB Review*.

Liwei Qi
Xingguo Yin
Haipeng Wang
Li Tao
ABB Corporate Research
Shanghai, China
levy-liwei.qi@cn.abb.com
xingguo.yin@cn.abb.com
jackson-haipeng.wang@cn.abb.com
matrix-li.tao@cn.abb.com

Reference

[1] ABB Research Ltd. (2007). Method for optimizing the performance of a robot. Patent number US 2007106421.