Trends in modern robot programming

Robot cell offline programming and calibration compliment each other perfectly, setting an example of harmony.

The increasing use of robots for industrial applications (spot welding of car bodies, arc welding, painting, material handling, cutting, deburring, assembly, etc) combined with a shortage of trained robot programmers means that robots need to be integrated and programmed quickly and efficiently using more effective programming tools.

The traditional and most common way to program robots is by using the robot’s programming pendant using the ‘teach’ method. While this robot programming method works quite well for a large percentage of applications (typically simple stand-alone robot cells, simple pick and place applications, standard arc-welding cells, etc), a number of factors as outlined below make the case to use effective ‘off-line’ programming tools and concepts:

- multiple products to be produced by the same robot cell/line – for example, the Tata World Truck Cab weld shop produces more than 20 product variants on the same line!
- new products being introduced more frequently – the traditional teach

Ajay Gopalswamy
CEO
Diffacto Robotics and Automation
gopalswamy@yulatu.com
method means that the robot is unproductive while the new part is being programmed
- shorter time to market for products – customers want their production equipment to be commissioned much faster than traditional methods allow
- precision applications – such as cutting, grinding, roller hemming, and similar applications - need a large number of part coordinates to be programmed, which are cumbersome, and – in some cases – impossible to program using traditional teach methods

**Offline programming**

Quite simply put, off-line programming of robots is the process of generating robot programs without using the robot teach pendant itself (consequently, without using the robot itself, thereby freeing up valuable robot cell productivity). To computer/CNC machine programmers, this may seem an obvious and easy method of programming, but robot users know that it is tricky to program a part which is not visible/physically available. The first step in off-line programming is to simulate the cycle of operations of the robot using 'robot simulation' software.

**Robot simulation**

Simulation is the process of creating the virtual robot cell and checking the robot's sequence of operations to ensure that the robot cell's objectives are met prior to actual implementation. This enables robot users to plan their manufacturing process thoroughly before committing to expensive investments or costly reworks at the time of implementation.

There are several robot simulation software available as tools for robot simulation - the most popular simulation tools are CimStation Robotics, Delmia and RobCAD. These ‘generic’ simulation software enable users to select any robot make from the library, and is generally not tied-in with a particular robot manufacturer. In addition, robot manufacturers themselves have developed the capability to simulate and program their robots – for example, ABB's RobotStudio, Fanuc’s Roboguide, Motoman’s MotoSIM EG, Kuka’s KukaSim, etc.

Simulation of a robotic work-cell involves the following process:
- import the CAD data of the part, tool (gripper, weld gun, glue gun, etc.), fixture, and all other peripheral equipment (conveyors, safety fencing, robot risers/peDESTALs, robot tracks/gANtries, etc)
- select the robot model from the software's library (some simulation softwares such as CimStation Robotics allow users to define and create robot models even if they are not available in the robot library by importing the CAD model of the robot and defining the kinematic parameters of individual robot links/joints)
- attach the tools to the robot tool mounting flange
- place the various peripheral equipment as per the process requirement in the robot cell (it must be noted that the results of a robot simulation study is optimum robot placement, pedestal design, and validation of the tooling on the robot,
hence the tool mounting arrangement and robot placement would initially be tentative and needs to be modified during the course of the robot simulation process.

- Check the entire robot operation for cycle time, tooling collisions, interference zones among robots (when multiple robots work together), manual loading/unloading issues, etc.

After the robot cell is simulated, the 3D cell created in the simulation can be used to generate the robot path programs. It may be noted that typically, robot off-line programming is used to generate the robot motion instructions only – i.e. 'path programs'. However, a robot program usually consists of several other instructions – for example, input/output instructions, welding schedules, logic instructions, etc., which need to be inserted specifically. It must also be remembered that robot programming languages differ from manufacturer to manufacturer. Therefore, it is necessary to have the post-processor specific to the robot being used – for example, ABB IRC5, Fanuc R30iA, Kuka KRC2, Motoman DX100, and so on.

**Need for robot calibration**

After a robot has been programmed using an off-line programming software, the robot program is ready for execution. However, we need to consider the following questions about the direct applicability of off-line programs:

- Is the actual robot on which the program being executed of exactly the same dimensions as the theoretical robot model used in the simulation?
- Is the actual tool (spot welding gun, gripper, etc.) manufactured exactly as per the design dimensions of the CAD model?
- Is the fixture, part locator/conveyor installed exactly as per the layout dimensions on the actual shop floor?

The answer to all the above 3 questions is “No”. The typical errors for the robot, tool and fixture installation are 5-10 mm, 2-5 mm, 25-50 mm respectively, resulting in the robot TCP (Tool Centre Point) being off from the desired position by 10-100 mm. This error in the TCP results in the need for manual touch-up. In typical automotive weld lines using 100s of robots, manual touch-up of programs is a cause for slower product launch.

Robot cell calibration eliminates the need for such touch-up by identifying the cell parameters (including the robot kinematic parameters, the tool XYZ and orientation errors, and the fixture XYZ and orientation errors) and then applying a correction to the off-line programs. A measurement hardware is used to identify the cell errors using randomly programmed robot positions, and using mathematical calculations, the kinematic parameters of the cell are estimated and a calculated correction factor applied to each of the target positions in the robot program, so that the corrected programs reach their intended positions without needing touch-up.

The measurement / identification phase is a one-time activity for a particular robot cell – until such time that a kinematic parameter is changed (for example, a motor replacement in the robot, change in the gun shank, relocation of the fixture's relative position with the robot, as might happen in relocation of a production cell/line).

---

**SMALL, MIGHTY AND AFFORDABLE I/O CONTROLLERS**

**THE RIO POCKET PLC**—packed with analog and digital I/O in a compact package. Its powerful RISC processor provides fast I/O handling and logic control. Unlike most PLCs, the RIO includes Ethernet 10/100Base-T and analog I/O at no extra cost. Network multiple RIOs to handle the I/O in your machine. Easy-to-program with Galil’s Ladder Interface Software or intuitive command language.

**RIO Pocket PLC**

**All the Bells and Whistles are Standard:**

- 32 digital and 16 analog I/O
- PID Process loops, Timers, counters and more
- Web interface & email capability
- Encoder feedback option
- Thermocouple and RTD options
- Ethernet 10/100Base-T & RS232 ports
- Power-over-Ethernet (PoE) or 18-36 VDC
- Measures just 99 x 109 x 3.3 cm
- D-type or screw terminals

View free web tutorials and specs at www.galilmc.com/pocketplc_AD or email India@galilmc.com to contact an Application Engineer.
Additional benefits of robot calibration

Dynacal, the world’s leading robot calibration system, developed and supplied by Dynalog, Inc., USA, has been used for many other robot applications apart from off-line programming:

- **Relocating robot cells** – calibration of the cell prior to or after shifting
- **Cloning and mirroring of robot cells** – when customers use 2 or more different robot cells to produce the identical same parts
- **Performance measurement** – robot calibration has been used in major automotive customers to determine the performance of each robot
- **Maintenance** – after a major repair on a robot’s mechanical manipulator (for example motor replacement, encoder replacement, and others), it is required to re-program the robot. If the robot is producing many different component variants, and is part of a line, re-programming can be an expensive and time-consuming activity. By carrying out a re-calibration of the robot, this time consuming re-programming can be avoided.
- **Robot crash recovery** – in line calibration of the robot TCP is possible and compensates for bent welding torches, water-jet cutting nozzles, due to crashes that occur due to abnormal situations during production
- **Robot replacement** – without re-programming the parts on the new robot using robot calibration
- **Reinstallation** – It is common for line builders/system integrators to carry out integration of the robot cells on their shop floor and trouble-shoot them prior to shipment to the end customer. By calibrating the work-cells in the line-builder’s shop-floor and re-calibrating the cells after installation at the end-customer’s facility, time-consuming re-programming of the robots can be avoided.
- **Precision robot applications** – like laser cutting, grinding/polishing, it is necessary for the robot to be very accurate. Calibration of the robot mechanical unit and TCP results in improving the accuracy of the robot and makes it suitable to perform precision jobs.
- **Temperature compensation calibration** – Some applications, such as robots being used for in-line inspection of car bodies, require the robot to be calibrated continuously in order to ensure that inaccuracies developing in the robot during the production process (for instance, expansion/contraction of the robot links due to self-heating or ambient temperature differences during the day) do not impact the accuracy of the robots over time. Such robots use automated calibration techniques, typically called ‘Temperature Compensation’, on a regular basis during production, in order to maintain the high level of accuracy required for such precision applications.

As seen above, robot calibration has wide applicability in different production environments, and its use can greatly benefit automation users by improving robot precision, reducing downtime and faster start-up of robotic production lines.

However, it should be remembered that robot cell calibration corrects for errors in the production equipment (robot, tool, fixture), but not in the customer part. It is still absolutely imperative for the customer to ensure good and consistent quality for a successful robot installation. In order for robot calibration to be effective, it is necessary that the quality of off-line programming is good.

Conclusion

As usage of robots becomes more common, and their applications and components being produced become more sophisticated, it is essential to look beyond traditional robot programming methods. It is inevitable that robot cells are simulated and programmed using sophisticated robot simulation software. Calibration of robot cells ensures that robots programmed off-line can be put into production without need for time-consuming ‘touch-up’ resulting in faster project start-ups. Indian industry needs to be geared up to adopt such technologies in order to be at the forefront of manufacturing competitiveness.

*Courtesy: Dynalog Inc., USA*