Robotics and Computer-Integrated Manufacturing 28 (2012) 87–94

Contents lists available at SciVerse ScienceDirect

Robotics and Computer-Integrated Manufacturing

journal homepage: www.elsevier.com/locate/rcim

Review

# Recent progress on programming methods for industrial robots

Zengxi Pan*, Joseph Polden, Nathan Larkin, Stephen Van Duin, John Norrish

*Faculty of Engineering, University of Wollongong, Wollongong, NSW 2522, Australia*

ARTICLE INFO

*Article history:*
Received 18 May 2011
Received in revised form
6 August 2011
Accepted 9 August 2011
Available online 31 August 2011

*Keywords:*
Industrial robot
SMEs
Offline programming
Online programming
Augmented Reality

ABSTRACT

Although an automated flexible production cell is an intriguing prospect for small to median enterprises (SMEs) in current global market conditions, the complexity of programming remains one of the major hurdles preventing automation using industrial robots for SMEs. This paper provides a comprehensive review of the recent research progresses on the programming methods for industrial robots, including online programming, offline programming (OLP), and programming using Augmented Reality (AR). With the development of more powerful 3D CAD/PLM software, computer vision, sensor technology, etc. new programming methods suitable for SMEs are expected to grow in years to come.

© 2011 Elsevier Ltd. All rights reserved.

**Contents**



## 1. Introduction

In the era of globalisation, manufacturing industries are facing increasing dynamics of innovations, shortened product life cycles, and a continuing diversification of the product range. At the same time, they are under the pressure of the shortage and high cost of skilled workers. Industrial robots based automation represents

* Corresponding author.
*E-mail address:* zengxi@uow.edu.au (Z. Pan).

0736-5845/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.rcim.2011.08.004

the best solution for both productivity and flexibility. Nevertheless, the programming of industrial robotic system for a specific application is still very difficult, time-consuming, and expensive.

For example, manually programming a robotic arc welding system for the manufacture of a large vehicle hull takes more than eight months, while the cycle time of the welding process itself is only 16 h. In this case, the programming time is approximately 360 times the execution time. As a result, small to median sized enterprises (SMEs) are not able to be benefit from robotic automation due to this programming time overhead.

In practical industrial applications, today there are two main categories of robotic programming methods, which are, online programming (including lead-through and walk-through) and offline programming (OLP) [1]. Conventionally for online programming, the teach pendant is used to manually move the end-effector to the desired position and orientation at each stage of the robot task. Relevant robot configurations are recorded by the robot controller. And a robot programme is then written to command the robot to move through the recorded end-effector postures.

Although the concept is simple, it is only suitable for programming the application of an uncomplicated process onto a workpiece with a simple geometry. In addition, the quality of the programme is limited by the skills of the operator and once the programme is generated, it is very difficult to make further amendments. In spite of these drawbacks, it is widely used as its intuitiveness, low programming skill requirement, and low initial cost. A few new programming methods are proposed in this category to alleviate the burden of jogging assisted by implementing additional sensors and control technologies.

Nowadays, OLP method, which is based on the 3D model of the complete robot work cell, is becoming more popular. Without removing the tedious programming overhead, OLP shifts the burden of programming from the robot operator in the workshop to the software engineer in the office. OLP has its strength on programming complex systems and is proved to be more efficient and cost-effective for production with large volumes.

Compared to the online programming method, it is more reliable and provides certain flexibility to the changes of product design. Since it relies heavily on the modelling of the robot and the workpiece, additional calibration procedures are usually inevitable to meet process accuracy requirements. Although there are many different OLP software packages available on the market, employing an OLP system usually means great programming effort, large capital investment and long delivery time.

A number of researchers intend to combine the knowledge of real world and the CAD model together to enjoy the benefits of both methods. Robot Programming using augmented reality (RPAR) is a typical example recently developed aiming to improve the intuitiveness and flexibility of OLP task. With different levels of involvement of human–robot interaction, sensor technology and CAD, the boundary between online and offline programming becomes blurred. Recent examples of RPAR will be presented in a separate section although they have not been seen in practical industry yet.

This paper will provide a comprehensive review of research progresses on the robotic programming methods in the last 10 years. It can be seen that the majority of research efforts are focused on providing a suitable robotic programming method for SMEs, by improving online programming methods, OLP methods or combining these two methods together using new concepts such as AR. Welding (mainly arc welding) and machining (mainly deburring), are the most widely investigated processes due to the fact that welding is the most common task for an industrial robot and machining is considered as a potential challenging application for an industrial robot.

This paper is organised in six sections. Following this introduction, Section 2 presents the recent development on online programming aided by various sensors and control technologies. Section 3 introduces the structure of OLP methods and provides a survey of available OLP software. Section 4 describes the features of RPAR with some examples. Summary and research trend are presented in Section 5 followed by an acknowledgement.

## 2. Online programming

Online programming has conventionally been carried out by skilled robot operators by guiding the robot through the desired path using a teach pendant, namely the lead-through method. Typically, the lead-through method includes the steps of jogging the robot through the desired path, recording the specific points in robot controller, and utilizing the recorded points to create movement commands. The robot operator programming a robot using a lead-through method is responsible for guiding the robot and maintaining the desired position and orientation of the robot in six degree-of-freedoms (DOFs).

Although the conventional online programming method is simple and has been widely used, it has several drawbacks. First, jogging a robot using a teach pendant is not intuitive as many coordinate systems are usually defined in a robotic system. The operator must always track which coordinate frame the robot is set in when jogging. Guiding the robot through the desired motion accurately while never allowing a collision with an object in the workspace is usually a very difficult and time-consuming task, especially when the workpiece has a complex geometry or the process itself is very complicated.

In addition, when a programme is generated, a lot of testing work has to be done before the programme is satisfactory for reliability and safety reasons. Third, the robot programme generated using the lead-through method lacks the flexibility and reusability. The tedious programming process has to be repeated again for a workpiece with only a slight difference. Other drawbacks of lead-through method include; the robot cannot be used for production during the teaching period, the operator is exposed to a hostile environment, and the quality of motions taught rely on the skill level of the operator.

In spite of all the above-mentioned drawbacks, online programming is still the only programming choice for most SMEs.

Online programming methods using more intuitive human–machine interfaces (HMI) and sensory information have been proposed from several institutions. Table 1 lists the recent research efforts on assisted online programming. The assisted online programming can be categorised into operator assisted online programming and sensor guided online programming.

### 2.1. Operator assisted online programming

To make jogging a robot in 3D space more intuitive, a few assistant teaching devices have been developed for walk-through teaching. Sugita [2] presented a teaching method using teaching support devices developed for a deburring and finishing robot. Two teaching support devices were introduced to measure the position and direction vector of the dummy tool on the tip of the posture measuring unit, and used to generate robot programme in robot coordinate.

Choi [3] presented the development of a force/moment direction sensor named COSMO that can improve the teach pendant based robot teaching. An experiment teaching a six axis commercial robot using the sensor is described where operator holds the

**Table 1**
Online programming methods.

| Ref. and Year | Sensors | Features | Dependence on marks | Type of path |
|---|---|---|---|---|
| [1] and 2001 | Micro-switch | Assisted jogging | N/A | Any |
| [2] and 2003 | Mechanical | Assisted jogging | N/A | Any |
| [9] and 2005 | Vision touch | Simple and robust | Draw on screen | 3D plane |
| [4] and 2006 | Force | Voice command and PDA interface | No | 3D curve |
| [6] and 2006 | Force vision | Hybrid controller with visual servo | Draw on workpiece | 3D curve |
| [10] and 2007 | Vision laser dots | Closed 3D path | Laser matrix | 3D curve |
| [8] and 2007 | Vision virtual touch | Combined with CAD model | Draw on workpiece | 3D curve |
| [5] and 2007 | Force | Force controlle Automatic path learningr | No | 3D curve |
| [11] and 2007 | Vision laser | Complete 3D clouds-point is available | Laser line scan | 3D surface |
| [12] and 2008 | Stereo vision | Relies on geometric features | No | 2D curve |

sensor with a hand, and moves the robot by pushing, pulling, and twisting the sensor in the direction of the desired motion. No prior knowledge of the coordinate system is required. The sensor used in the device is a micro-switch, and this intuitive robot teaching can be implemented at a very low cost.

Schraft [4] proposed an intuitive teaching method to use a walk-through attempt to provide a tool for fast and effective teaching of industrial robots in this niche. The user guides the robot with a handle that is equipped with a force torque sensor and commands the robot using a speech dialog system. The acquired trajectory can be adapted by using a PDA and 3D graphical user interfaces.

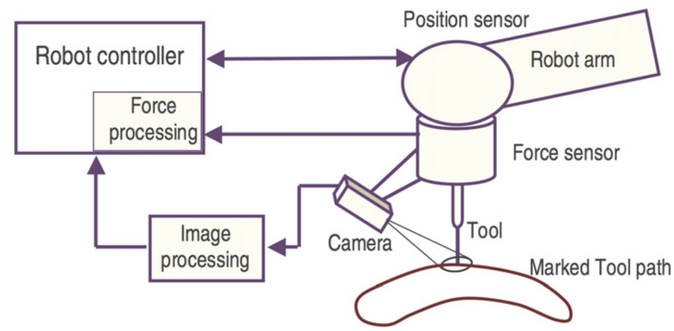


**Fig. 1.** The system configuration for robot path generation by controlling a robot tool to continuously follow the centre of the marked tool path based on the vision, force and position sensor fusion [6].

### 2.2. Sensor guided online programming

Using assistant teaching devices usually means introducing additional sensors and calibration procedures to an already very complex robotic system. Pan [5,7] developed a programming by guiding (PbG) method based on ABB's IRC5 controller with the optional force control feature. Two major functions provided by the commercial available force controller make the entire programming process collision free and automatic. The first function is walk-through, in which robot is compliant in selected directions (force control directions) and stiff in other directions (position control directions). To change the position or orientation of the robot, the robot operator could simply push or drag the robot with one hand. The second function is called path-learning, in which robot is compliant in the normal-to-path-direction to make the tool constantly contact the work piece.

As the accuracy of the final programme is determined by the robot force controller and does not rely on the skill of the robot operator, a 3D robot path with higher accuracy can be generated automatically. This is of extreme benefit for applications when process tools have contact with workpiece, such as in machining processes.

While Pan and Zhang's method still requires jogging during the first stage of programming to guide the robot motion, some other researchers have eliminated jogging from the entire programming process by involving other sensor technologies. Zhang [6] used the same controller platform and extended the concept by adding a visual servo. The system configuration of this system is shown in Fig. 1. A hybrid position/force/vision control platform was developed to control the robot motion in different directions using various sensor feedbacks. The system is able to generate a robot programme by automatically following a path marked with a standard marker pen. The position control is used to maintain the tool orientation; vision sensing is used to follow the curve; and force sensing is used to maintain the contact between the tool and the workpiece.

Solvang [8] also presented a vision based programming methodology by identifying a path drawn onto the workpiece. This line is captured by a single camera for the 2D ($x$ and $y$) coordinates. The depth coordinate ($z$) is achieved by a virtual "hit and withdrawal" procedure using a commercial available simulation programme that uses the industrial robot to map the surface of the workpiece. During the mapping process, the robot moves along the existing 2D path and at every point of the path contacts the work-piece surface. When contact occurs, the $z$ coordinate is stored establishing the position. Although CAD model of the workpiece is used, the major part of this method is still sensor-robot interaction rather than offline path planning.

Nicholson [9] developed a rapid robot programming method using image data for weld reclamation repair works. Instead of drawing marks on the workpiece, the user interacts with the image to define select/define the robot 2D path. This selection is done via a drawing module that allows the user to generate an area onto the picture of the workpiece. The $z$ coordinate is determined using the touch sensing built into the welding system. Unlike most vision based systems, which are reliable on calibration results and sensitive to lightening condition, this method provide robust results due to its simplicity.

In some situations, projecting structured light using a laser is more feasible than drawing marks on the workpiece. Gonzalex [10] presented aspects related to the generation and tracking of closed trajectories over a surface of unknown geometry using structured lighting in the form of a laser spot matrix. Simple image analysis algorithms can be used to detect the centre of laser spots in the images. After the process of surface characterization is complete, the user selects, in camera-space, a starting point and a direction of reference over the surface for the robot path. As the image plane information gathered from the projection of structured lighting is limited, a second order polynomial function is defined to approximate the 3D curve welding path considering the best fit to the surface. A closed 3D path is achieved by connecting the starting point and ending point of the neighbouring segment of the trajectory.

Hu [11] developed a strategy to automate a leather surface roughing process using structured light 3D machine vision for object profile perception. The structured light scanning system consists of an analogue camera, laser line generator and driven linear slide to provide scanning motion for the camera and laser. Non-Uniform Rational B-Spline (NURBS) interpolation is applied to reconstruct a smooth continuous trajectory from the discrete path coordinates.

Stereo vision was also used to acquire 3D coordinates for robot programming and distinct features such as corners and edges could be easily identified from a workpiece. Takarics [12] attempted to use the stereo vision technology to programme a weld trajectory based on the intelligent space concept using two fixed cameras. The weld seam is recognized in two images by edge detection algorithms and the path trajectory was generated by the 3D reconstruction from both images. The method is capable of generate a 2D planar curved path for arc welding processes.

Although dramatic progress has been carried out to make online programming more intuitive, less reliant on operator skill, and more automatic, most of the research outcomes are not commercial available aside from [6]. This is partially because most of these methods are limited to their specific setups and are yet to be applied to general applications. As cost-effective sensor assisted online programming solutions become commercially available, the installations of robotic automation cells will become more cost effective for SMEs.

## 3. Development of OLP

OLP methods, which utilise 3D CAD data of a workpiece to generate and simulate robot programs, are widely used for automation system with large product volumes. Herein the complete robot cell is modelled in 3D. The user can test the reachability, fine-tune properties of robot movements and handle process related information before generating a programme that can be downloaded to the robot.

OLP offers many advantages over the online method. First, the programming process does not require the actual robot, minimising the production robot down time. Robot programs can be developed earlier in the design/production cycle and programming can be carried out in parallel with production rather than in series with it. Second, programs generated offline are more flexible than jog-and-teach method. Programme changes can be incorporated quickly by only substituting the necessary part of the programme and previously developed routines can be easily included in new programs. Third, simulation is usually incorporated into the OLP method. As a result, programs can be pre-checked, thereby confirming the robots' movements, minimising the chance of error and therefore improving productivity and safety. There is also a greater possibility for optimization of the workspace layout and the planning of robot tasks.

Although OLP has the above-mentioned advantages, it is not popular for SMEs users due to its obvious drawbacks. It is difficult to economically justify an OLP for smaller product volumes due to the high cost of the OLP package and programming overhead required to customise the software for a specific application. Development of customised software for off-line programming is time-consuming and requires high level programming skills. Typically, these skills are not available from the process engineers and operators who often perform the robot programming in process today. As OLP methods rely accurate modelling of the robot and work cell, additional calibration procedures using extra sensors are in many cases inevitable to meet process requirements.

While OLP software providers emphasise on making the OLP package more powerful, modular, and flexible to reduce secondary development for specific applications, academic researchers have dedicated attention to improved process planning algorithms and have developed a few OLP software package using open source technology.

### 3.1. Steps of OLP

OLP is more complex than online programming as the programming method not only needs to acquire the 3D robot targets but also needs to plan the trajectory of robot motion and optimise the sequence of the process. The key steps of OLP are shown in Fig. 2.

#### 3.1.1. Generation of 3D CAD model

OLP starts from a 3D CAD model of the workpiece, while it is very common for a product to have a CAD model, for parts without a 3D model, or a product that has changed after its CAD model is finalized there are several methods available to generate the required 3D computer model.

In some situations, a 3D scanner can be used to capture the workpiece geometry [13]. The collected points-cloud is converted to the surface model of the workpiece and a smoothing/filtering procedure removes sensory noise before the model can be used for tag creation.

In other situations, when only a 2D CAD is available, the 3D model of the workpiece can be obtained from either multiple views of a 2D drawing [14], by additional sensors, or the robot is simply programmed in 2D [15].

Although there are various types of CAD files, most modern OLP software packages are capable of converting other types of CAD files to a compatible. Conversion between different types of CAD files is less a problem these days with the developments in the CAD/CAM industry.

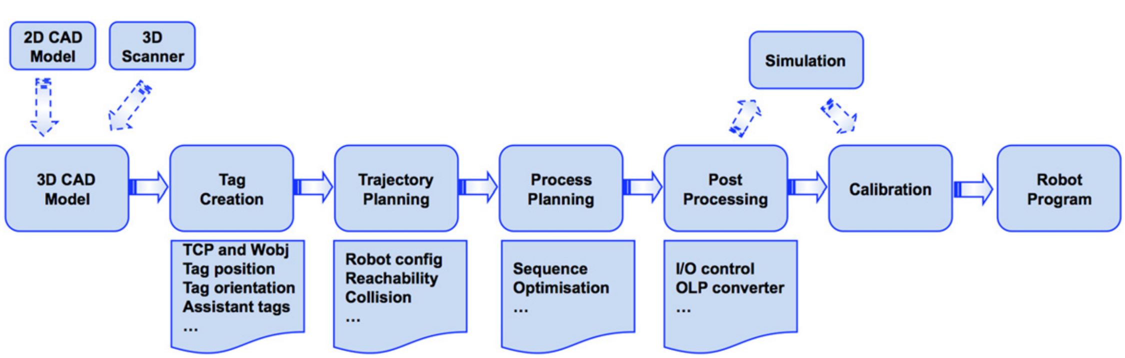


**Fig. 2.** Key steps of offline programming.

*3.1.2. Tag creation*

This step involves extracting robot position tags from 3D CAD data with a specific tool centre point (TCP). It is usually a time consuming process and can require secondary programming for automatic tag recognition. OLP software is available that provides built-in functions to generate tags from features, such as corners and edges, from CAD data. The position and orientation information of the tool must be generated from a combination of CAD model and process requirements. Assistant tags such as home points, approach points, and retreat points are also specified manually in a CAD environment. Attempts have been made to automatically extract robot motion information from the CAD data such as the system proposed by [16].

*3.1.3. Trajectory planning*

Since the inverse kinematics of industrial articulated robots usually have multiple solutions in Cartesian space, the robot configuration needs to be selected by considering issues such as reachability, minimising configuration transition, collision avoidance, etc. As most of the existing OLP software is not able to provide an optimal solution automatically, either manual assignment or secondary software development using APIs is necessary at this step.

*3.1.4. Process planning*

Planning a complex manufacturing process involves a higher level of optimisation for resource assignment, cooperation of multiple robots to minimise cycle time. As this step is more relevant to the requirement of a specific process, it is not available in commercial OLP software. For robotic weld large structures, the task sequencing of a large number of welds within limited cycle time can be treated as a general “travelling salesman problem” (TSP), solutions based on genetic algorithm have been proposed by a few researchers [17–19].

*3.1.5. Post-processing*

The post-processing stage includes adding necessary I/O control signals for equipment in the work cell, smoothing and fine tuning the path if necessary, and conversion to the programme language of the specific robot target. Post-processing is more of an issue for generic OLP software as it requires compatibility among different robot manufacturers [22].

*3.1.6. Simulation*

Robotic work cell simulation is considered as a significant tool that OLP software packages bring to the robotic programming. Simulation enables the programme to be verified without the use of an existing physical robot, which reduces the downtime of a robotic system [13,24].

*3.1.7. Calibration*

Ideally, a programme generated in an OLP system would be downloaded to the robot controller and put into action immediately [20]. In practise, however, the deviation between the actual geometry of elements in the work cell, such as the workpiece, and the nominal geometry makes calibration almost necessary for all OLP system.

*3.2. Existing robotics OLP software*

Robotic manipulators are highly complex systems. Consequently, the development of computational platforms that allow for their precise modelling, and close to real-life simulation of their behaviour, constitute a fundamental tool for robot designers, users, and students of the field. This reason has inspired the creation of numerous graphical software environments, from non robot manufacturers, academic researcher and also from the robot manufacturers themselves.

**Table 2**
Offline programming software package.

| Software/Ref. | Company/Feature |
|---|---|
| *Generic robotics software* | |
| Delmia (IGRIP, ENVISION): Kineo, CENIT; [13,21] | Dassault Systems; VR |
| RobCAD (Em-workplace); [22,23] | Technomatix; VR |
| Robomaster | Robomaster |
| Robsim; [24] | Camelot |
| Workspace 5 | Wat solutions |
| Cosimir | Festo |
| *Robotics software from robot manufactures* | |
| RobotStudio | ABB; Most popular |
| MotoSim | Motoman |
| KUKA-Sim, CAMrob; [25] | KUKA |
| Roboguide | Fanuc |
| Wincaps III | Denso |
| 3D STUDIO | Stäubli |
| MELFA WORKS | Mitsubushi |
| Pc-ROSET | Kawasaki |
| AX on Desk | Nachi |
| *Academic/open source robotics software* | |
| [26] | Various MATLAB based software |
| [27] | Aristoteles University of Thessaloniki, Greece; Based on Solidworks |
| [28] | Orebro University, Sweden; Based on standard CAD |
| [16,29] | Based on AutoCAD, Autolisp |
| PIN; [20] | European Centre for Mechatronics, OpenGL based macro programming |
| ROBOMO; [30] | OpenGL |
| [31] | Daegu University, Korea; VRML, Tribon |
| RoBott; [36] | University of Minho, Portugal; OOP Java |

*3.2.1. OLP software from robot manufacturers*

It can be seen from Table 2, that almost every robot manufacturer has its own OLP software. Since the OLP software is more compatible to the robot hardware, secondary development of the OLP system is relatively easier. The cost of this type of OLP package is generally lower than one using genetic OLP software as the hardware and software are packaged together. This explains why ABB RobotStudio is by far the most widely used OLP software.

*3.2.2. Generic OLP software*

This category includes two most powerful OLP software, Delmia (formally IGRIP, ENVISION with third party add-ons from Kineo, CENIT) from Dassault Systems and RobCAD (Em-Workplace) from Technomatix. The advantage of generic packages is that they are more flexible for hardware from different manufacturers and often link to product lifecycle management (PLM) packages to provide production line optimisation. Major automobile and airplane manufacturers use these packages to integrate the robotic systems into their general automated production line. Also, both software packages have the feature of Virtual Reality which allows the user to be fully immerged into the simulation environment.

Today, OLP systems are able to do more than just simulate robot trajectories and perform assembly simulation. Simulation technologies are also able to model the interaction of several manufacturing processes, manufacturing resources, and product maintenance issues.

*3.2.3. Open source or academic OLP software*

Due to the high cost and limited accessibility of commercial OLP software, a number of researchers have developed alternative

OLP software. While some researchers [27–29,16] have developed OLP packages based on the existing CAD software, such as AutoCAD and Solidworks, others [20,30,31] have started from scratch using OpenGL, VRML and Java technology. Fig. 3 shows the flow chart of offline programming developed in [31].

### 3.3. Gaps of OLP software and requirements

Due to the costs and the complexity, the advantages of OLP are not sufficient for the use of this technology in the manufacturing operations of SMEs.

There is no available OLP system in the market, which has implemented the complete OLP chain, although many links exist separately. For example, DELMIA V5 Robotics provides functions for tag creation and trajectory planning. However, they still need be created manually in OLP environment or coded using automation (VB/VBA) technology. For arc welding of a complex structure, steps 2, 3, and 4 of OLP are extremely tedious. It may take a few weeks to generate tags for the hundreds of seams inside a vehicle hull, including both accurate position and proper orientation. Although some components for path planning and process optimisation, such as collision detection, layout planning, time measurement, etc are available, a complete path planning function does not exist and is fully relied on the process knowledge of the programmer.

## 4. Programming using Augmented Reality

Burdea [32] provided a review of the synergy between virtual reality and robotics. AR is an emerging technology that has been derived from VR [33]. AR is an environment where computer-generated 3D objects are blended onto a real world scene, to enhance a user's interaction with the real world [34]. The use of Augmented Reality for robot programming represents a revolutionary concept.

Augmented Reality, i.e. interactively overlaying the real environment with virtual spatial information, can be used to make the advantages of graphically interactive simulation directly available in the real production environment and to provide an efficient and intuitive communication channel for spatial information [35]. As shown below in Fig. 4 a virtual model of an aeroplane washing robot can be superimposed over a scaled model of an aeroplane. The virtual model of the robot can be moved about the model airplane to generate a robot sequence that can later be calibrated and programmed for an actual airplane washing robot.

These Robot Programming using AR (RPAR) techniques allow a form of offline robot programming to take place without having to model the workpiece in the virtual environment. RPAR is also useful when an in-situ approach is required as the virtual robot can be augmented into the real-world workcell. This approach can eliminate a lot of technical difficulties that can relate to calibration issues between the virtual and real worlds.

RPAR carries through some of the inherent benefits of OLP, such as not having to take a physical robot out of production, and the safety and operational benefits are retained as well. Another advantage of the proposed RPAR environment with a virtual robot is the programming of large robots where the online method is unfeasible (such as airplane washing robots) as the proposed methodology for planning collision-free path and the RPAP approach are scalable [33].

A RPAR system was created by utilising the video-based tracking method in the ARToolkit and creating the necessary coding using the C programming language. This ARToolkit method utilises identification tags with unique patterns printed on them placed around critical elements of the workcell. The Head Mounted Display (HMD) of the RPAR system consists of a single IEEE Firefly camera and an i-glasses video display goggle. This setup utilises a tracking approach where markers attached to objects (static or moving) are tracked using a cameras attached to the HMD. The cameras not only provide the video images needed for processing, but also provide the user with a view of the real world [33].

The technologies AR offer a highly potential utility concerning the improved application of simulation technique during the planning and development of production systems. Fig. 5 shows in which ways the respective technologies can be applied during each stage of the simulation process [37].

## 5. Summary

Conventional online programming is a completely manual process. The robot operator has the freedom to move the robot,

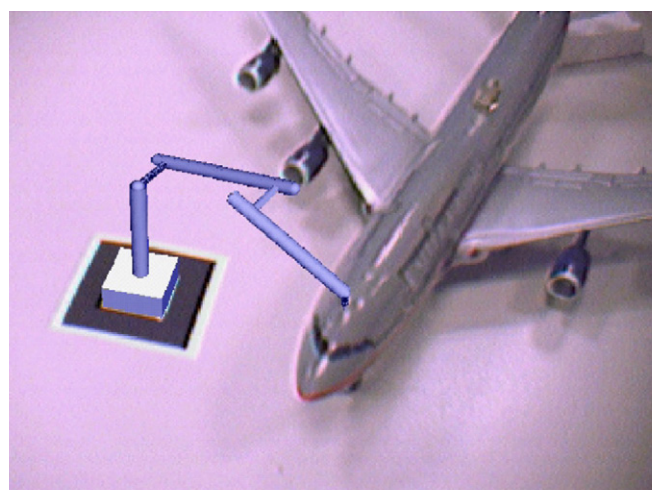

**Fig. 4.** Virtual robot in real environment, miniature airplane washing robot [33].

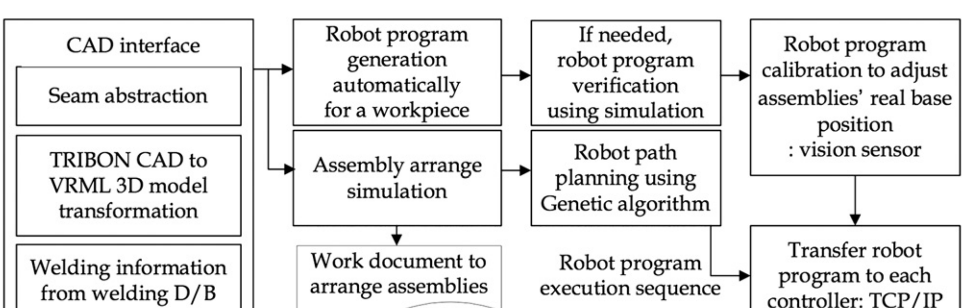


**Fig. 3.** Flow chart of offline programming from Ref. [31].

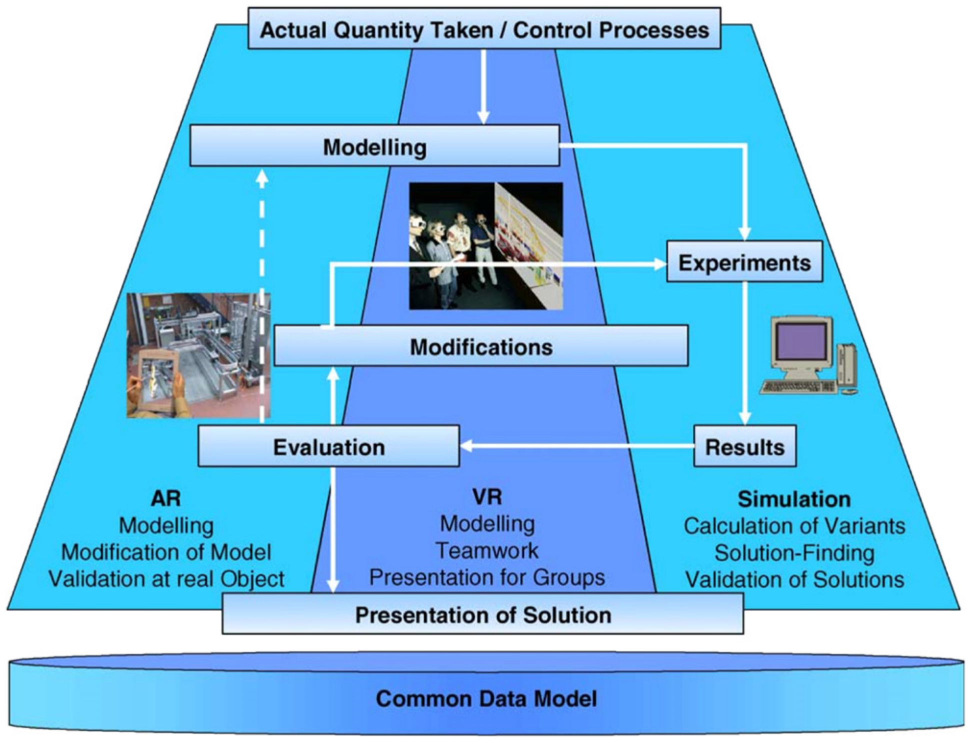


**Fig. 5.** Possible fields of applications of VR/AR-based tools during a material flow simulation [37].

**Table 3**
Comparison of various robot programming methods.

| Programming options | Pros | Cons |
|---|---|---|
| Online programming (jog and teach) | • Easiest for system integration<br>• Low cost and development time | • Least flexible, programming robot is done manually<br>• A robot programmer is necessary for any further programme change<br>• Cost and risk for maintaining the robot programmer are high |
| Assisted online programming | • Easy for system integration<br>• Does not need a robot programmer for system operation and maintenance | • Limited flexibility, only suitable for certain applications<br>• Some additional system integration cost<br>• May not be available from some robot manufacturers |
| Offline programming software from robot manufacturer | • More flexible<br>• Options for simulation<br>• Reduced system downtime for re-programming<br>• Easier to develop with dedicated software from robot manufacturer | • Additional development cost for system integration<br>• Software engineer required for system maintenance |
| Generic offline programming software | • Most flexible option for future system update<br>• Capable of integrating the robots from different manufacturers | • Most difficult for system integration<br>• There may be compatibility issues for some robot manufacturers |

select the configuration and plan the process. It is an efficient and cost effective solution for a simple robotic system. As the process becomes more complex, the suitability of online programming reduces.

On the other hand, OLP is a complete automatic programming process. Once the complete work cell is modelled in CAD and OLP code is developed for a specific application, the robot programme is generated automatically. As modelling and OLP coding creates a large cost overhead, it is only economically justified for production with large volumes, usually by large enterprises. Table 3 compares the pros and cons of various online and offline robot programming methods. Programming using VR/AR is not included as they have not been practically used by industry yet.

In the last 10 years, extensive research efforts have been carried out on the methodologies for programming industrial robots suitable for SMEs. The boundary between online and offline programming methods are becoming blurred as many new proposed methods includes components from both sides.

Progress in online programming is largely based around sensor and control technologies to assist the operator in creating complex robot motion more easily. Development in OLP bifurcates into different directions. While the commercial OLP providers are developing more powerful, modular and compatible OLP packages, academic researchers have not give up on low cost open source OLP solutions. RPAR is originated from the idea of making OLP more interactive and flexible. In fact, it combines the features of both online and offline programming. With the development of more powerful 3D CAD/PLM software, computer vision, sensor technology, etc, new programming methods suitable for SMEs are expected to grow in years to come.

## Acknowledgement

The authors acknowledge the support of the Defence Materials Technology Centre (DMTC), which was established and is supported by the Australian Government's Defence Future Capability Technology Centre (DFCTC) initiative.

## References

[1] Jr MH, Wei L, Yong LS. An industrial application of control of dynamic behaviour of robots—a walk-through programmed welding robot. In: Proceedings of the IEEE international conference on robotics and automation. San Francisco, CA; April 2000.
[2] SUGITA S, et al. Development of robot teaching support devices to automate deburring and finishing works in casting. The International Journal of Advanced Manufacturing Technology 2003. Springer-Verlag London, Dec.
[3] Choi MH, Lee WW. A force/moment sensor for intuitive robot teaching application. In: Proceedings of IEEE International Conference on Robotics and Automation, ICRA, vol. 4; 2001. p. 4011–6.
[4] Schraft RD, Meyer C. The need for an intuitive teaching method for small and medium enterprises. In: Porceedings of the ISR-Robotik. Munich, Germany; 2006.
[5] Pan Z, Zhang H. Robotic programming for manufacturing industry. In: Proceedings of ICMEM, international conference on mechanical engineering and mechanics. Wuxi, China; 2007.
[6] Zhang H, Chen, H et al. On-lien path generation for robotic deburring of cast aluminium wheels. In: Proceedings of the IEEE/RSJ international conference on intelligent robots and systems. Beijing, China; 2006.
[7] Pan Z, Zhang H. Robotic machining from programming to process control: a complete solution with force control. Industrial Robot: An International Journal 2008;35(5):400–9.
[8] Solvang B, Sziebig G, Korondi P. Robot programming in machining operations; chapter in book title. Robot Manipulators 2008. 978-953-7619-06-0, intechweb.
[9] Nicholson A. Rapid adaptive programming using image data. PhD dissertation. University of Wollongong, Australia; 2005.
[10] Gonzalex-Galvan EJ, et al. An algorithm for optimal closed-path generation over arbitrary surfaces using uncalibrated vision. In: Proceedings of the IEEE international conference on robotics and automation. Roma, Italy; 2007.
[11] Hu Z, Marshall C, Bicker R, Taylor P. Automatic surface roughing with 3D machine vision and cooperative robot control. Robotics and Autonomous Systems 2007;55(7):552–60.
[12] Takarics B, Szemes PT, Nemeth G, Korondi P. Welding trajectory reconstruction based on the Intelligent Space concept. In: Proceedings of the conference on human system interactions; 2008. p. 791–6.
[13] Bi ZM, Lang SYT. A framework for CAD- and sensor-based robotic coating automation. IEEE Transactions on Industrial Informatics 2007;3(1):84–91.
[14] Kim JY. CAD-based automated robot programming in adhesive spray systems for shoe outsoles and uppers. Journal of Robotic Systems 2004;21:625–34.
[15] Pulkkinen T, et al. 2D CAD based robot programming for processing metal profiles in short series manufacturing. In: Proceedings of the international conference on control, automation and systems. Seoul, Korea; 2008.
[16] Pries JN, Godinho T, Ferreira P. CAD interface for automatic robotic welding programming. Industrial Robot: An International Journal 2004;31(1):71–6.
[17] Kim KY, Kim DW, Nnaji BO. Robot arc welding task sequencing using genetic algorithms. IIE Transactions 2002;34:865–80.
[18] Kang HJ, Park JY. Work planning using genetic algorithm and 3D simulation at a subassembly line of shipyard. In: Proceedings of the OCEANS '04. MTTS/IEEE TECHNO-OCEAN '04, vol.1; 2004. p 218–22.
[19] Zacharia PTh, Aspragathos NA. Optimal robot task scheduling based on genetic algorithms. Robotics and Computer-Integrated Manufacturing 2005;21: 67–79.
[20] Dai W, Kampker M. PIN-a PC-based robot simulation and offline programming system using macro programming techniques. In: Proceedings of the 25th annual conference of the industrial electronics society, vol. 1; 1999. p. 442–6.
[21] Brown RG. Driving digital manufacturing to reality. In: Proceedings of 2000 Winter Simulatin Conference, vol. 1; 2000. p. 224–8.
[22] Bruccoleri M, D'Onofrio C, La Commare U. Off-line programming and simulation for automatic robot control software generation. In: Proceedings of the fifth international conference on industrial informatics, vol. 1; 2007. p. 491–6.
[23] Dong W, Li H, Teng X. Off-line programming of spot-weld robot for car-body in white based on Robcad. In: Proceedings of the international conference on mechatronics and automation, ICMA; 2007. p. 763–8.
[24] Lee DMA, Elmaraghy WH. ROBOSIM: a CAD-based off-line programming and analysis system for robotic manipulators. Computer-Aided Engineering Journal 1990. October.
[25] Vollmann K. A new approach to robot simulation tools with parametric components. In: Proceedings of the IEEE ICIT '02. IEEE International Conference on Industrial Technology, vol. 2; 2002. p. 881–5.
[26] Zlajpha L. Simulation in robotics. Mathematics and Computers in Simulation 2008;79:879–97.
[27] Mitsi S, et al. Off-line programming of an industrial robot for manufacturing. International Journal of Advanced Manufacturing Technology 2005;26: 262–7.
[28] Soron M, Kalaykov I. Generation of continuous tool paths based on CAD models for friction stir welding in 3D. In: Proceedings of the Mediterranean Conference on Control & Automation, MED '07; 2007. p. 1–5.
[29] Yang Y, Chen X, Ling C, Kang B, Robot A. Simulation system basing on AutoLisp. In: Proceedings of the second international conference on industrial electronics and applications, ICIEA; 2007. p. 2154–6.
[30] Jaramillo-Botero A, Matta-Gomez A, Correa-Caicedo JF, Perea-Castro W. ROBOMOSP. IEEE Robotics & Automation Magazine 2006;13(4):62–73.
[31] Kim Chang-Sei, Hong Keum-Shik, Han Hans Yong-Sub, Kim Soo-Ho, Kwon Soon-Chang. PC-based off-line programming using VRML for welding robots in shipbuilding. In: Proceedings of the IEEE conference on robotics, automation and mechatronics, vol. 2; 2004. p. 949–54.
[32] Burdea GC. Invited review: the synergy between virtual reality and robotics. IEEE Transactions on Robotics and Automation 1999;15(3):400–10.
[33] Chong JWS, Ong SK, Nee AYC, Youcef-Youmi K. Robot programming using augmented reality: An interactive method for planning collision-free paths. Robotics and Computer-Integrated Manufacturing 2009;25(3):689–701.
[34] Pettersen T, et al. Augmented reality for programming industrial robots. In: Proceedings of the second IEEE and ACM international symposium on mixed and augmented reality; 2003. p. 319–20.
[35] Reinhart G, Munzert U, Vogl W. A programming system for robot-based remote-laser-welding with conventional optics. CIRP Annals—Manufacturing Technology 2008;57(1):37–40.
[36] Bottazzi VS, Fonseca JFC. Off-line robot programming framework. In: Proceedings of the joint international conference on autonomic and autonomous systems and networking and services, ICAS-ICNS; 2005. p. 71.
[37] Dangelmaier W, et al. Virtual and augmented reality support for discrete manufacturing system simulation. Computers in Industry 2005;56:371–83.