Journal of Engineering and Applied Sciences 3 (11): 835-847, 2008
ISSN: 1816-949X

Modeling a Deburring Process, Using DELMIA V5®

Mohammad Mohammad, Vipul Babriya, Tarek Sobh
School of Engineering, University of Bridgeport, Bridgeport CT 06604, USA

Abstract: This study investigates a Deburring process which is integrated with a robotics application. Conventional tumble deburring processes have been used for years in the surface finishing industries. The manual process is tedious, inconsistent and inaccurate. Automation in the finishing process has been proved very beneficial. Integration of robots in the process provides more flexible and convenient process. We investigate the off-line simulation of finishing edges using the Product Life Cycle tool DELMIA®. The process extracts the feature to generate a part finishing process plan. Collisions with the environment are detected and avoided through the simulation of cutting the path before the path is downloaded to the actual hardware. Either the tool or the part is being fixed on the robot's end effector and simulated with respect to the defined path. This study presents the complete off-line simulation of the Part in Hand as well as Tool in Hand approaches for the deburring process.

Key words: Dexterity, offline simulation, flexi burr, tool orientation, manufacturing, robotics

INTRODUCTION

As the number of robotic applications and installations continue to grow, the need for effective and efficient robot programming techniques grows proportionally. In the past, almost all robotic involvement was through teach-by-show techniques/manual deburring operations. This implied that the robot's paths are generated either by mounting a dowel pin that matches the tool's diameter or by mounting a pointed teaching tool on the robot. The pin or the teaching tool is moved manually to a point where they touch a finished part edges and the software records that point. After repeating this process along the part edges, the robot controller uses the recorded points to define the path. The more complex the part, the more programming time is required to achieve an acceptable path (Stouffer and Robert, 1995).

Off-line simulation: Off-line simulation refers to the technique of running models in a purely stand-alone format, not connected to any hardware and normally not run in Real-Time. Off-line simulation and lead-though programming promise acceleration and streamlining of the robot programming process (Shiakolas *et al*., 1999). They enable the user to program robot motions in a simulated "virtual computer environment".

Tumble deburring operation: Tumble Deburring removes sharp edges or burrs and smoothes surfaces of metal components (resulting from some manufacturing or casting process). It can be done as a wet or dry process. A 6-or 8-sided machine with either horizontal or vertical axis is used with abrasive "stones".

The deburring tool could be small and a part with many small/accurate details is hard to be reached via a CNC machine. A good solution would be to use an industrial robot (with 6 axis) to perform the deburring process. Pratt and Whitney has estimated that 12% of their total machining hours are devoted to manual deburring and chamfering of parts after they have been machined (Roberts *et al*., 1992).

BASIC CONFIGURATIONS OF DEBURRING PROCESS PERFORMED BY INDUSTRIAL ROBOTS

Tool in hand: In these applications a compliant tool is mounted on the robot end effectors and manipulated over the part to be finished. Tool in Hand configurations are used where the part to be finished is too large or unwieldy for a robot to carry. Belt media is rather rare in tool in Hand applications because it is difficult to build compact tools using belts. Figure 1 illustrates a tool in hand configuration.

Part in hand: The work piece is fixed on the robot end effector while the tool is fixed as an auxiliary device and the robot manipulate the work piece to be deburred as result of contact between the work piece and the tool. Part in hand applications are most often used when the part to

Corresponding Author: Vipul Babriya, 45 Park Terrace, Bridgeport CT 06604, USA

be finished is relatively small in size (comparing with deburring/grinding/polishing tool's size). Gripper tooling allows the robot to pick up the part and manipulate it against the abrasive finishing media (Godwin, 1996). Figure 2 illustrates a part in hand configuration.

Advantages of the part in hand approach:

- Often, robot load/unload operations can be combined with surface finishing operations on a single work station; i.e., a robot can remove a part from a serial line conveyor, finish the part and then place the part in final packaging. Doubling up these operations can provide a much greater return on investment
- The surface finishing apparatus, whether it is a belt, wheel or disk device, can be quite large using longer belts, large diameter wheels and higher horsepower, meaning that parts can be processed faster with longer intervals between media changes

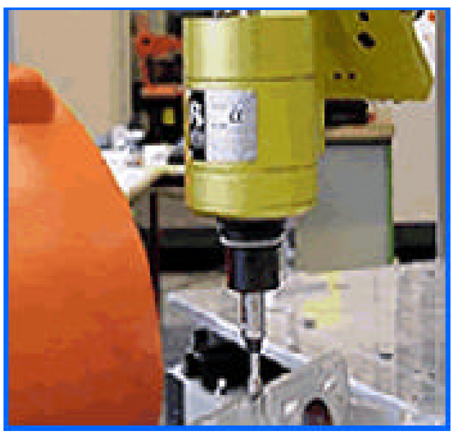


Fig. 1: Tool in hand configuration

Fig. 2: Part in hand configuration

Disadvantages: Sometimes it is impossible to finish the entire surface of the part. This can be due to interference with the robot gripper and insufficient robot dexterity to reach around the part in different positions.

EXAMPLES OF INDUSTRIAL ROBOTS USED IN DEBURRING PROCESSES

Motoman: Figure 3 shows the Motomon DX1350N robot. It is frequently used in deburring industrial application. Following is the configuration of the robot.

Nachi: Figure 4 illustrates the Nachi SC35F robot. The Nachi SC35F industrial robotic system is designed for heavy loads and long strokes, with high positioning repeatability and speed. The SC35F is a lightweight design allowing for easy installation and transport. Following is the configuration of this robot model.

ABB: The ABB IRB 140 is most commonly used in surface finishing industries. This robot is a compact and powerful 6-axes machine with a unique combination of

Fig. 3: Motoman DX1350N. Motoman DX1350 N Max Reach: 1355 mm, Max Payload: 35 kg, Weight: 27kg, Repeatability: 0.6 mm

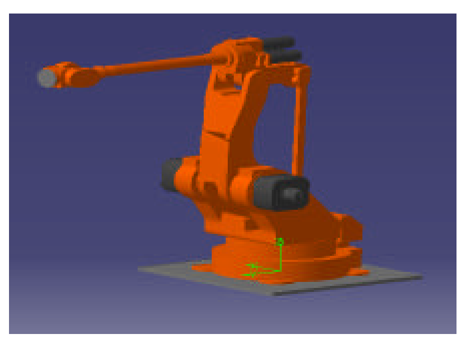


Fig. 4: Nachi SC35F. Nachi SC35 F, Max Reach: 2002 mm, Max Payload: 35 kg, Weight: 400 kg, Repeatability: 0.1 mm

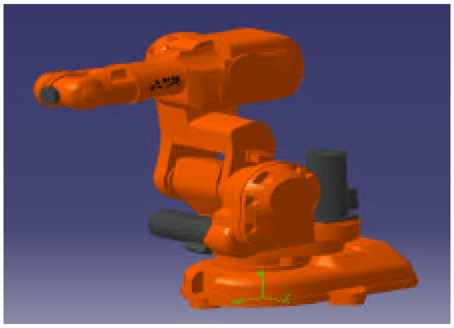


Fig. 5: ABB IRB 140. IRB 140, Axis: 6, Reach: 810 mm, Payload: 5 kg, Repeatability: 0.03 mm

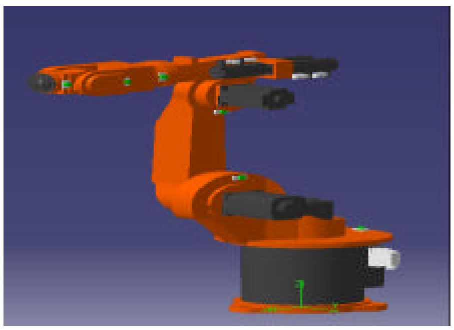


Fig. 6: Kuka KR 15 SL. Axis: 6, Reach: 1503 mm, Payload: 15 kg, Weight: 315 kg, Repeatability:. 01 mm

fast acceleration, large working area and high payload. Figure 5 shows the IRB 140. Following is the configuration of the robot.

Kuka: The exclusive use of stainless steel on all surfaces, together with high IP rating, makes the KR 15 SL suitable for fields of applications with stringent requirements as to hygiene, sterility and absence of particles-such as in the food handling industry or medicine as well as the surface finishing industry. Figure 6 illustrates the Kuka KR 15 SL model.

TOOL USED IN DEBURRING PROCESSES

The success of most automated deburring operations is dependent on the flexibility and consistency of the deburring tool. Problematic burrs, residual material on parting lines and flashing on die-cast parts must be removed. Systems designers typically program the robot to move its deburring tool along a path defined by discrete points. That path, however, may not coincide exactly with the shape or contour of the surface to be deburred because of variations in the part itself or differences between the part edge and the exact path the robot has interpolated. Otherwise, the system designer has to create thousands of very physically close tag points automatically to reduce the error. Figure 7 illustrates the interpolated path defined by off-line simulation and actual path/edge that the robot is supposed to follow.

Deburring tools with feedback force sensors [Flexdeburr]: Flexdeburr is a high-speed, air turbine driven tool for deburring aluminum, plastic and steel. While spinning at high speeds, the lightweight, rotary tool has radial compliance supported by air pressure applied to the shaft allowing the tool to perform consistently on irregular part patterns. Figure 8 illustrates, what occurs when a non-compliant tool is being linearly interpolated between taught points on a contoured surface. The surface contours in Fig. 8 are greatly exaggerated for clarity; however, this effect exists in parts with even a slight contour. As shown, even though path points are taught perfectly on the surface, as the robot interpolates between the points, the grinding media position relative to the part surface varies. On actual parts with slighter contours, the varying contact can be detected by listening for variations in the grinding/ deburring motor speed.

The bottom illustration in Figure 8 shows the same surface but using an auxiliary compliant tool. With this setup, it is possible to maintain consistent media contact with the surface with only a minimum number of taught points. In addition, if the path is linear or circular, it can be interpolated with a few number of discrete tag points. The problem always occurs with linear interpolation between tags on an irregular edge of the work piece. Thus, using compliant force tool design, one does not have to generate a large number of points connected by interpolating linear path, but the tool will correct the error in real time. Steps of the process on Delmia V5 will be clarified later in this study.

Figure 9 illustrates a deburring tool with feedback sensor. The deburring tool has a rigid outer housing and internal motor/spindle assembly that provides the compliance. The pneumatic motor/spindle assembly is mounted on a pivot bearing mounted to the tool housing. This allows the pneumatic motor to move with the pivot bearing independently of the housing. The radial “compliance field” is created by a ring of small pneumatic positions located near the front of the tool housing. The compliance force can be exerted on the spindle/tool in any direction (360°) radially from the tool.

Force/torque sensor: In the other types of deburring processes, when the robot holds the work piece and the

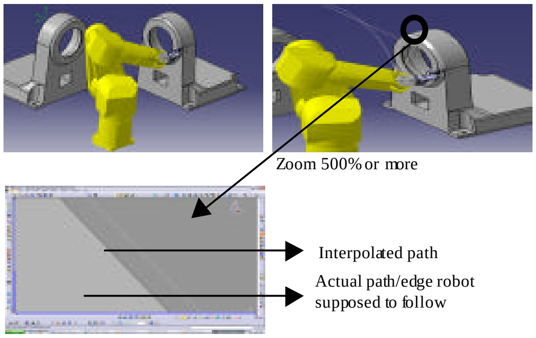


Fig. 7: Illustration of the interpolated and actual path of robot supposed to follow

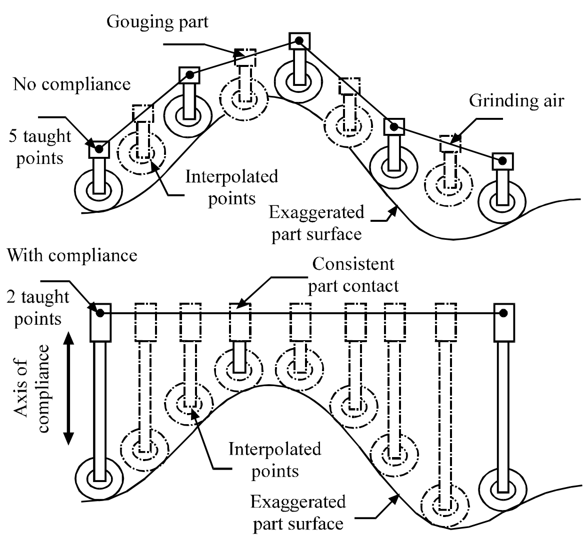


Fig. 8: Illustration of the compliance on surface contact

Fig. 9: Flexdeburr

tool/grinding belt is fixed as shown in Fig. 10. There should also be a fixed force/torque sensor between the

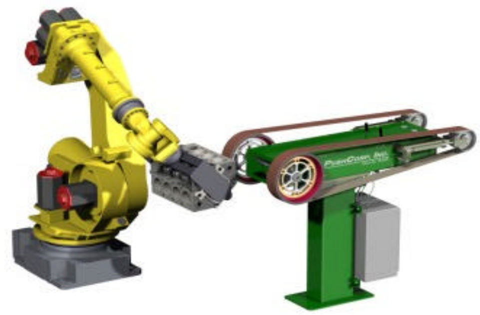


Fig. 10: Illustration of force/torque sensor in grinding Process

robot and the work piece to maintain constant contact force in surface finishing application.

Path profile: Robot motion for both part in hand and Tool in hand applications involves smooth sweeping movements. There are three important aspects of the motion which have to be carefully considered in order to have a good finish:

- Surface speed
- Part/Tool Orientation
- Approach Vector

Surface speed: There are four process variables which most greatly affect the Material Removal Rate (MRR) in finishing operations. These variables are:

- The aggressiveness of the media
- The force with which the media is applied
- Rate at which the media is fed (RPM)
- Speed at which the media is moved over the part surface

It should be noted, that all these variables can influence and offset each other when trying to achieve a particular surface finish while maintaining a desired process throughout. For example, a more aggressive media can be used with lighter applied force at lower RPM so that a higher surface speed can be used to increase production rates. However, this must be balanced with the fact that aggressive media does not produce extremely fine surface finishes. The one rule that should be remembered regarding surface speed is that the more aggressively a part being worked on a given pass (i.e., aggressive media at high force levels with high rpm) the faster the surface speeds. Thus, it can cause overheating the part surface and media.

Unfortunately, there is no theoretical approach to determine the optimal combination of the process variables. Trial and error combined with prior experience seem to be the best method to determine these variables.

Part/tool orientation: The relative orientation of the part and tool is very important in achieving a consistent, uniform surface finish. The overall surface appearance is directly affected by the consistency of location and shape of the area where the finishing media contacts the part. This area is called the contact patch. The contact path is "where the rubber hits the road". Therefore, the single most important path programming consideration is to maintain constant contact patch, no matter what configuration is used, Part in Hand or Tool in Hand and no matter what programming method is used. In most finishing operations, it is usually desirable to have the finishing media applied to the part surface at an angle varying from normal to the surface to three or five degrees off the normal direction depending on the type of media.

Approach vector: In the approach vector the part comes first with the finishing media. This transition point often is where many surface finishing inconsistencies are found in the final part. The first rule that applies to the approach vector is to approach gradually. If one looks at how a person finishes parts, one would find that person's movements are characterized by sweeping motions with gradual, controlled setting down and lifting of the media. These gradual and controlled motions serve to "feather" the finish at the edges.

Active tools, with closed-loop control, have the unique ability to eliminate these undesirable effects by giving robots a more human touch (Edwin, 2000). These tools are able to automatically compensate for stiction and inertia effects in the same way as human operators anticipate these effects.

Tool example: Figure 11 illustrates an active tool, with closed- loop control. The filing tool SWINGFILE 2000 is well suited to remove burrs particularly in narrow slots and grooves. It can be employed for metallic as well as nonmetallic parts. It is also well suited for the deflashing of aluminium die casting. It can be employed with any orientation. The file tip is compliable in two directions. The pressure against the part edge can be set from the robot program.

The filing tool can be mounted stationary either onto a tool stand or onto the robot arm. If these tools have to be changed automatically, a tool changer is available. Commercial filing insertion can be used. The air motor needs oiled compressed air for optimal life.

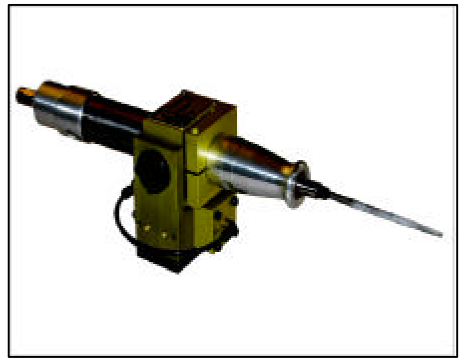


Fig. 11: Active control tool (SWINGFILE 2000)

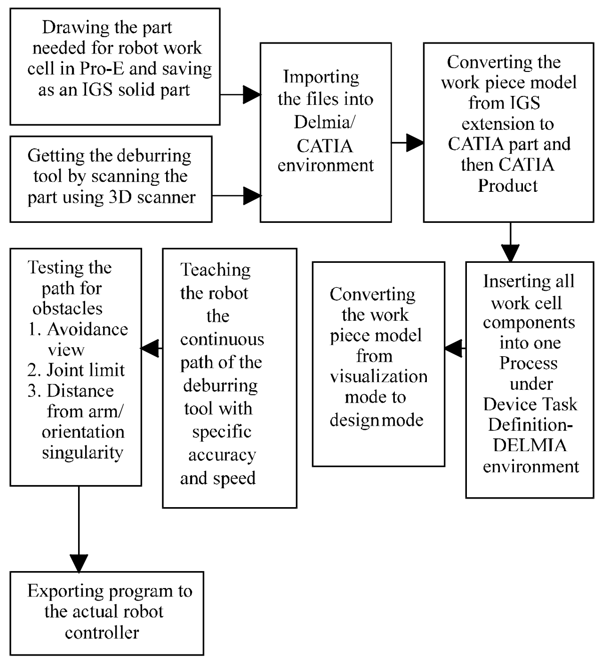


Fig. 12: Hierarchical block diagram of deburring process via DELMIA V5R17

STEPS/TERMINOLOGY FOR DEBURRING [TOOL IN HAND/PART IN HAND] VIA DELMIA V5 R17

Figure 12 illustrates the block diagram of a complete procedure for deburring (Tool in Hand/Part in Hand) via DELMIA V5 R17.

STEPS FOR TOOL IN HAND APPROACH USING DELMIA V5R17

- Inserting the all components of the process in Device task definition

- Attaching the tool to the robot
- Adjusting the positions of every component
- Adding new task to the robot program
- Automatic generating tag points
- Simulating for testing the scenario/process
- Adjusting and interpolating the tag points

Figure 13 illustrates a snapshot of the robot, tool and part in the DELMIA environment.

Procedure in details

Inserting all components of the process in device task definition: Figure 14 illustrates how to start with device building in DELMIA.

- File-open "basesolid.igs"
- Save as basesolid. cat part
- Start Resource-Device Building

Figure 15 illustrates procedure to generate a CATIA product.

- Right click on the P.P.R Tree on Application÷ Existing component÷Choose the Part name "basesolid.cat part"
- Save it as basesolid.cat product
- Start-resource-device task definition

Figure 16 illustrates the start up window of device task definition.

- Insert product
- Choose part name and tool name
- Insert resource

Path: C:\Program Files\Dessault/Systems\B17\intel_a\startup\Robotlib\V5DEVI-CES

Attaching the tool to the robot:

- Select Attach the tool icon from robot management toolbar.

A Robot Dressup window appears on the screen as depicted in Fig. 17.

- Choose the robot first
- Choose the tool second
- Change the snap reference to Base 1
- Change Tool Centre Point (TCP) to Tool 1

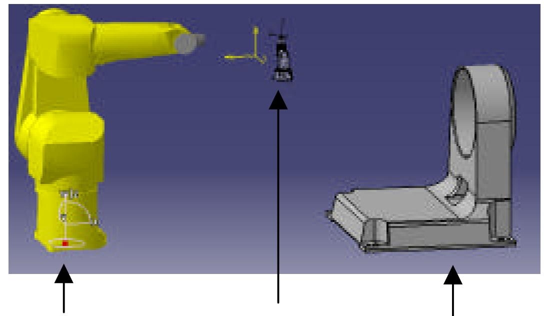


Fig. 13: Illustration of robot, tool and part in a DELMIA environment

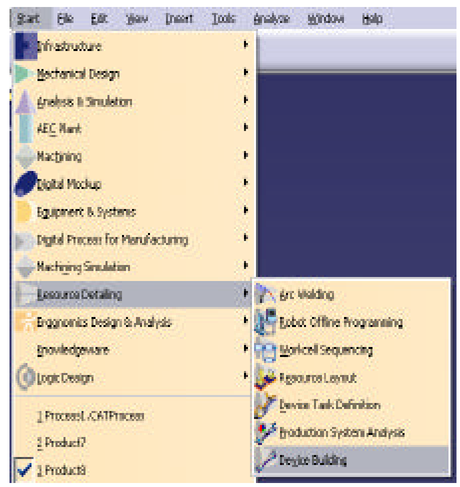


Fig. 14: Snapshot of device window of building

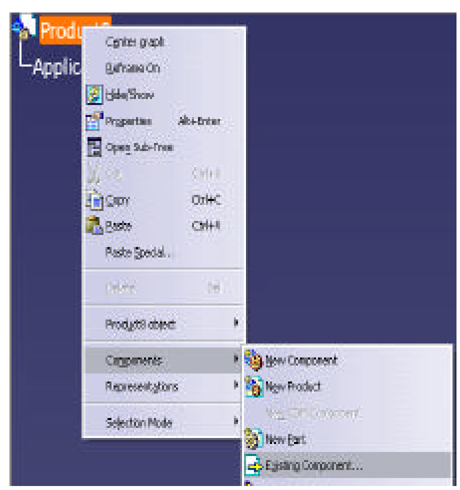


Fig. 15: Screenshot CATIA product

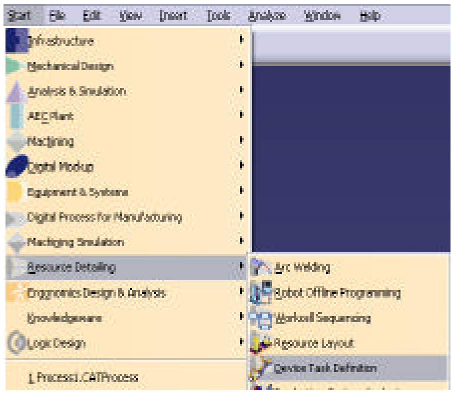


Fig. 16: Snapshot of start up device task definition in DELMIA

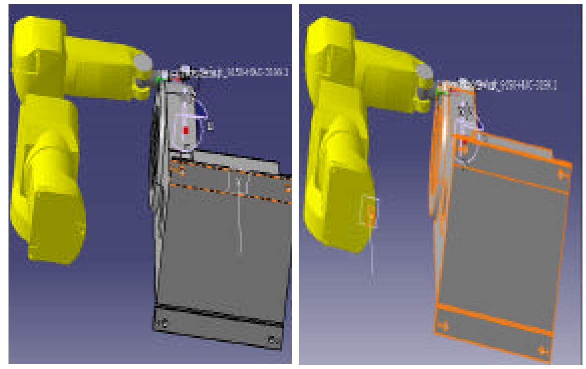


Fig. 18: Illustration of alignment of part and robot in device building

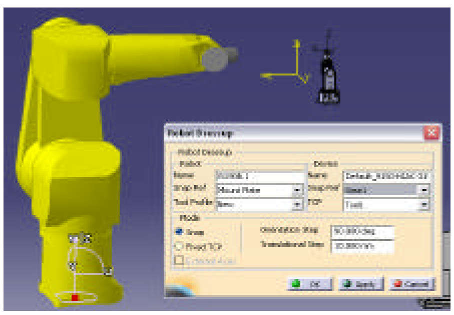


Fig. 17: Snapshots of robot and tool with dress up window in DELMIA

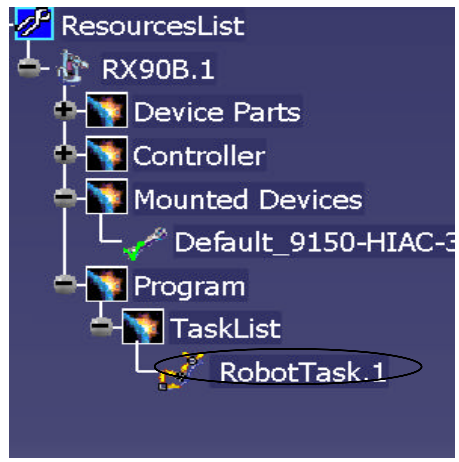


Fig. 19: Illustration of robot task selection under PPR tree

Adjusting the positions of every component:

- Align Side (optional procedure)
- Choose the down side of the part (as master)
- Choose the down side of the robot (slave)

The robot base will be in same plane with as the base as shown in Fig. 18.

Adding new task to the robot program:

- Add new task to the robot's program

Figure 19 illustrates the selection of a new robot task from TaskList under the PPR tree

- Choose the robot

Automatic generating tag points:

- Click Create follows Path activity
- Choose RobotTask.1 from the PPR tree
- Choose the edges/curves you want to generate tag points automatically
- Choose the part as product name
- Press Finish

Figure 20 illustrates the selection of follow path activity on the part. User can change chord length and can increase or decrease the number of tags generated on the edges.

Interpolate Tags after changing the orientation:

- Click Robot Teach, then choose robot
- Choose task "follow path activity"
- Double Click on the follow path activity or click modify

Figure 21 shows the robot teach window. Users are able to modify the tag points orientation as per requirement.

Adjusting and interpolating the tag points:

- Choose modify from the new window "Teach Continuous Path", as shown in Fig. 22. In the small window at the right choose first node then adjust the compass/coordinate tool orientation

Fig. 20: Snapshots windows for the selection of follow path activity under robot task

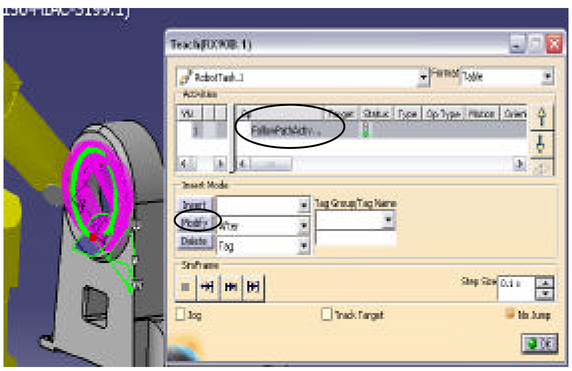


Fig. 21: Snapshots of robot tech window in DELMIA

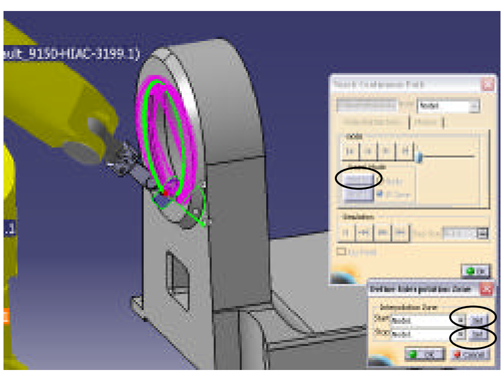


Fig. 22: Snapshots of Teach Continuous Path window

C Click set. Repeat same procedure for the end node.
C Press OK
C After interpolation. The user can find the desired interpolated path as shown in Fig. 23

Simulating for testing the scenario/process: Figure 24 illustrates simulation of the robot task. It is also possible to check for clash detection during simulation.

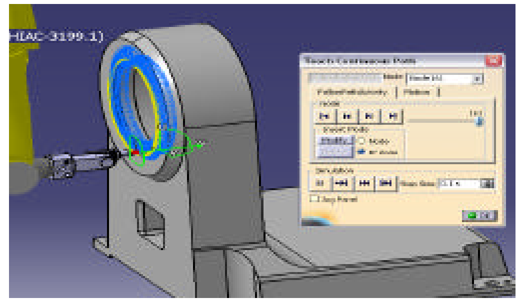


Fig. 23: Illustration of interpolated path after modification

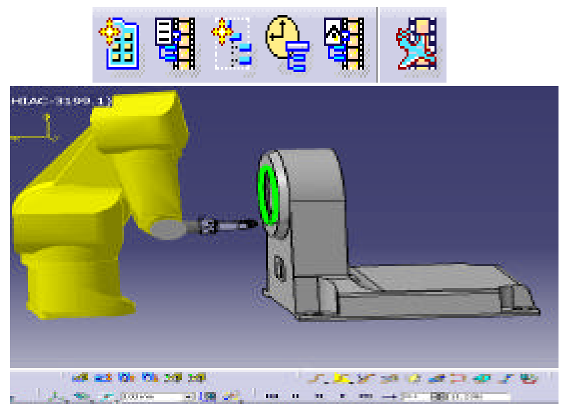


Fig. 24: Snapshot of simulation window for a predefined robot task

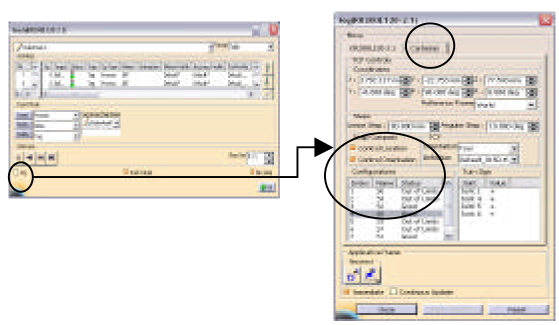


Fig. 25: Snapshots of Teach and Jog configuration windows for robot joint configuration

Check for Clash detection by 

C Change the "Singular/Out of Limit Configuration" To Configuration (Good /non Singular).

Figure 25 illustrates teach and jog configuration windows for the robot orientation in DELMIA.

- Click the “jog” Checkbox as shown below and choose Cartesian card
- Change the configuration to “Good”

Change the position of the tool if the user doesn’t find any configuration ‘Good’.

STEPS FOR PART IN HAND APPROACH USING DELMIA V5R17

- Inserting all the components of the process in Device task definition
- Adjusting the positions of every component
- Attaching the tool to the robot
- Adding new task to the robot program
- Generating automatic tag points
- Adjusting and interpolating the tag points
- Simulating for testing the scenario/process

Figure 26 illustrates some part in hand configuration screen shots in DELMIA V5 R17.

Procedures in details

Inserting all the components of the process in device task definition:

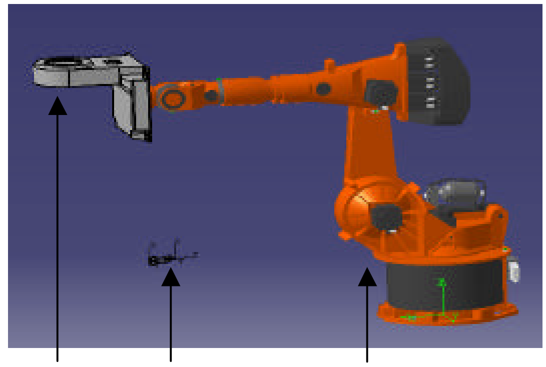
- Go to file-open “basesolid.igs”
- Save this part as basesolid.cat part
- Click on Start-Resource-Device Building as shown in Fig. 27
- Right click on the P.P.R Tree on Application÷ Existing component÷Choose the Part name “basesolid.cat part” as shown in Fig. 28
- Save it as basesolid.cat Product
- Start-Resource-Device Task Definition
- Insert Product
- Choose part name and tool name
- Insert Resource

Adjusting the positions of every component:

- Choose the last part of the robot as parent
- Choose the product/part as a child
- Press ok

Figure 29 illustrates the parent child relationship between robot and part.

- Snap Resource: Attach the part to the end of the robot
- Choose the part first
- Choose icon “Define origin at point or center of face” to set up the coordinate as depicted in Fig. 30a.



Part | Tool: ATI (9150-HIAC-3199) | Robot KR 200L120-2.1

Fig. 26: Screenshots of part in hand configuration mode in DELMIA

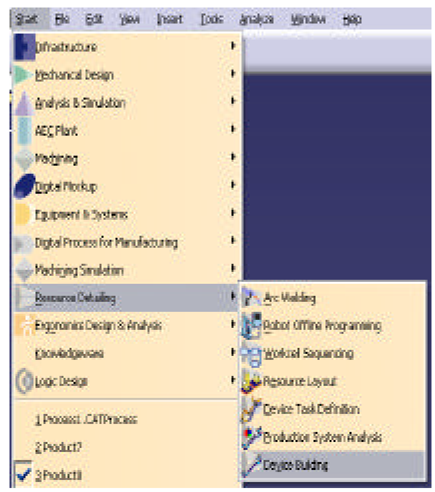


Fig. 27: Screenshots for start for up of device building

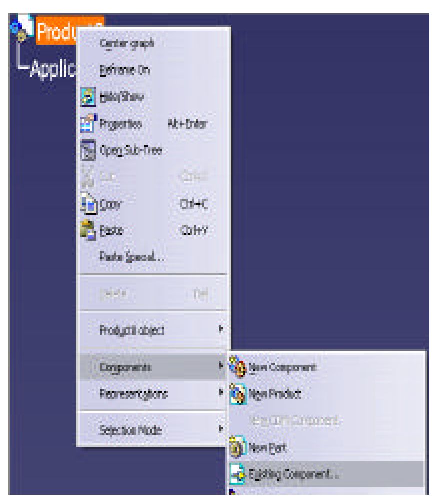


Fig. 28: Screenshots CATIA product

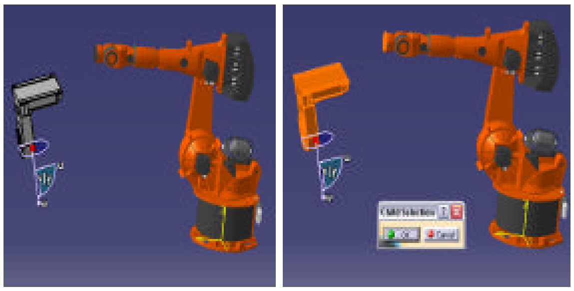


Fig. 29: Screenshot of the parent-child relationship between part and robot

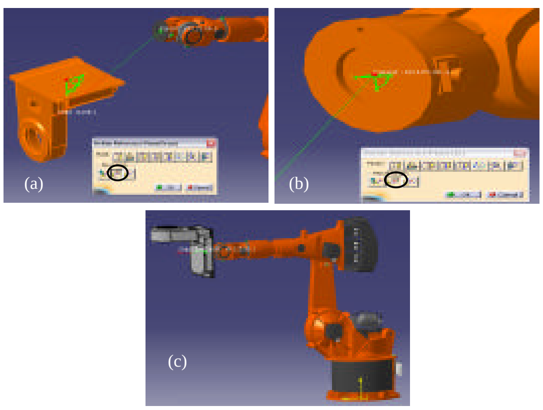


Fig. 30: Procedures for snapping the part at the end effector of the robot

- Choose the last tip of the robot [not whole robot body] as shown in Fig. 30b
- Choose icon “Define origin at point or center of face” to set up the coordinate again
- Check that the coordinates coincide as shown in Fig. 30c
- Save the initial State

Attaching the tool to the robot:

- Attach the tool as fixed

Figure 31 illustrates screenshots of the dress up window.

Adding new task to the robot program:

- Add New Robot Task

Figure 32 illustrates the robot task under TaskList on PPR tree.

- Change the mode for the part from “Visualization Mode” to “Design Mode”

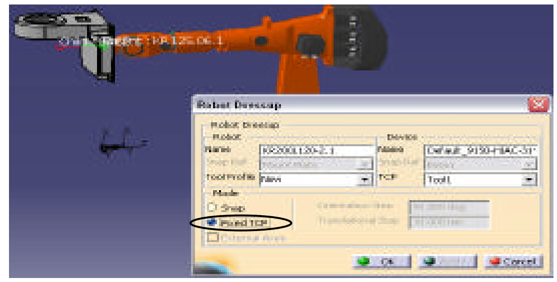


Fig. 31: Screenshots of the dress up window for the Part in Hand configuration

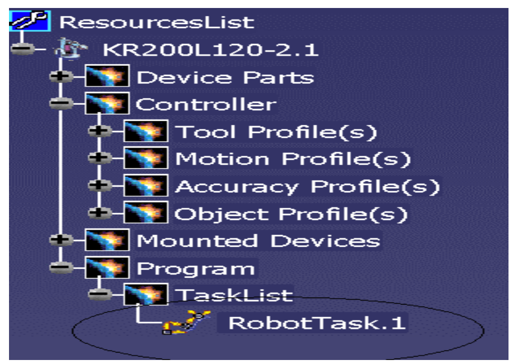


Fig. 32: Screenshots of robot task on PPR tree

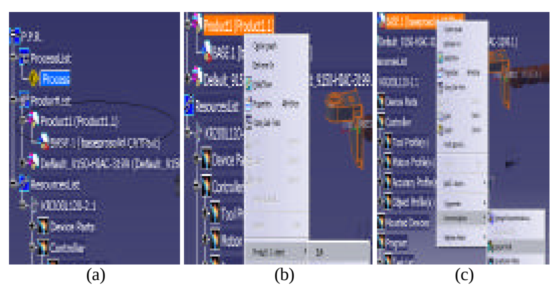


Fig. 33: Screenshots for changing the mode from Visualization to Design

Figure 33 shows the steps of mode change from Visualization to Design.

- Right click on Product1 (Product 1.1) in PPR tree as shown in Fig. 33b
- From Product1.obj click Edit
- Then Right click on Base.1 (basesolid.CATpart)-Representation-Design Mode

Find the final shape of the tree/branch of the part in “Design Mode” as shown in Fig. 34.

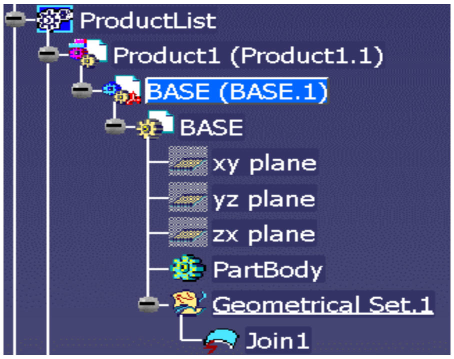


Fig. 34: Screenshots for final shape of the part in Design Mode

Fig. 35: Screenshots of process under PPR tree

- Return to the "Device Task Definition" Environment Double Click on Process in PPR tree as shown in Fig. 35.

Generating automatic tag points:

- Click Create follow Path activity
  - Choose the task from PPR tree under program as is illustrated in Fig. 36
  - Choose the Task from the PPR to add follow path activity to
- Choose the edge of part to create the path
  - Choose any edge for follow path activity as shown in Fig. 37.
- Generate new task from follow path activity
  - Right click on Follow Path Activity
  - Follow Path Activity.obj-choose Create New Robot Task as shown in Fig. 38
- Make Parent/child relation between the Tag Group and the part
  - Choose the part as parent
  - Then choose the tag group from PRR tree as child

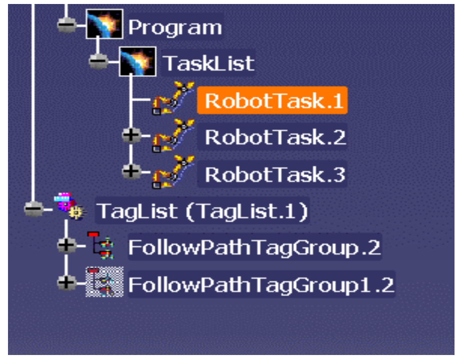


Fig. 36: Screenshots of robot task under PPR tree

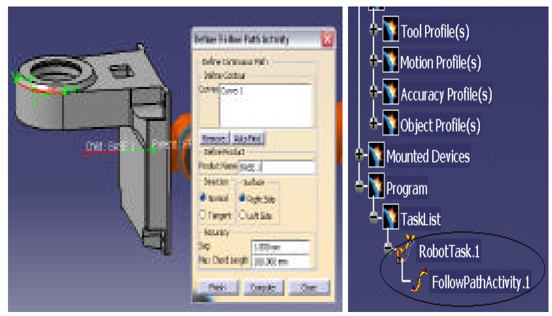


Fig. 37: Screenshots of follow path activity window and its branch under PPR tree

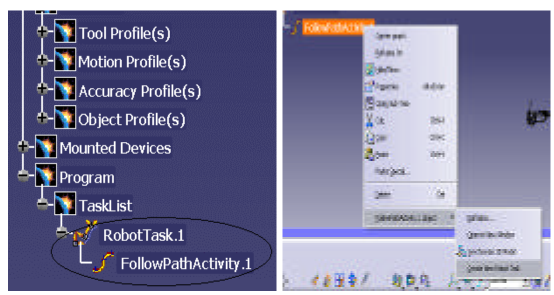


Fig. 38: Screenshots of creation of new robot task from follow path activity

Adjusting and interpolating the tag points:

- Adjust the tag coordinate whether by Tag Transformation or teach a device as per screenshots shown in Fig. 39
  - In case of discontinuous motion-choose all operation as shown in Fig. 40
  - Right click choose motion-Linear
- Interpolate the path
  - Choose the Tag Group
  - Interpolation of the tag points of predefined path
  - Select all tag points and click interpolate as shown in Fig. 41

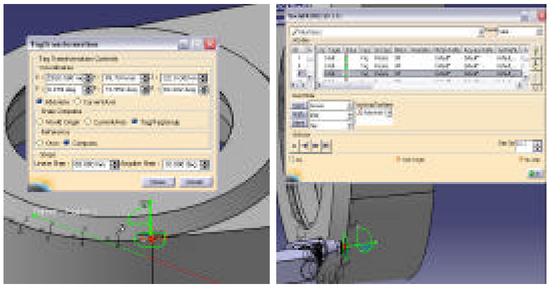


Fig. 39: Screenshots of Tag Transformation and Teach Robot Device

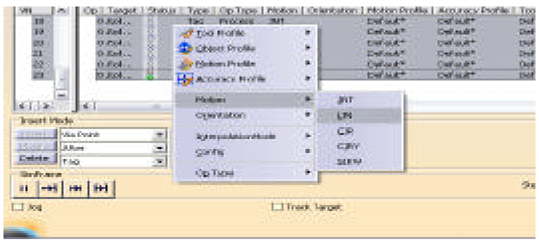


Fig. 40: Screenshots for selection of all tags in linear motion for predefined activity

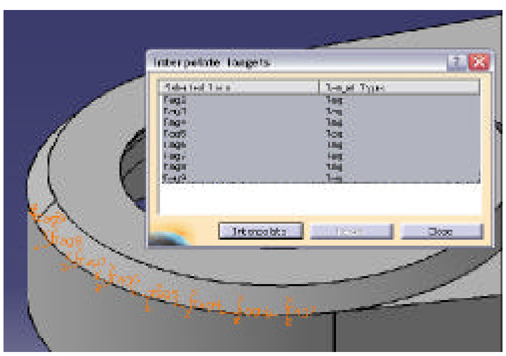


Fig. 41: Screenshots windows for interpolation of tag points

- Add one/two points for pure contact between part and the tool

Simulating for testing the scenario/process : Check the whole task/scenario with Teach Window or With Robot Task simulation as shown below in Fig. 42.

- User can also check for Clash detection by clicking the clash detection icon
- User might change the "Singular/Out of Limit Configuration" to "Configuration Good/non Singular".

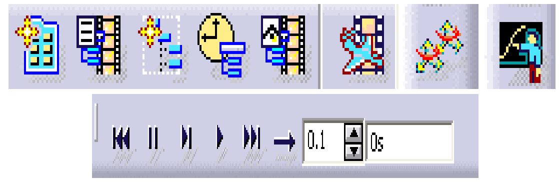


Fig. 42: Simulation toolbar

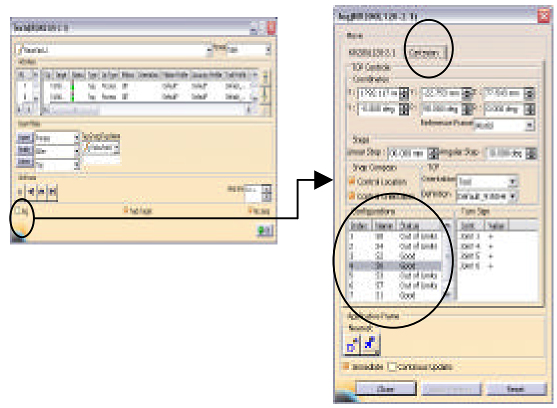


Fig. 43: Screenshots of Teach and Jog configuration window for robot joints configuration

  - Click "jog" Checkbox as shown below in Fig. 43 and choose Cartesian card
  - Change the configuration to "Good" as shown in Fig. 43.
- Change the position of the tool if don't find any configuration 'Good'.
- User might change the singularity*tolerance if the 4th joint is from 1-10 and the case indicates that current configuration is singular

Singularity: DELMIA as a software package is interested in orientation singularity when the 4th joint and 6th joint are on the same axis this is because the 5th joint (The middle joint between them) is at zero distant. It results in a loss of 1° of freedom, as the result of motion whether by the 4th or 6th joint is the same.

CONCLUSION

DELMIA® enables the offline simulation of tool as well as part in hand deburring approaches in the virtual environment cell. Automated deburring processes eliminate health and safety issues associated with hand held tools. The process is convenient for complex part finishing where it is very hard to reach by conventional machines. Robotics integration provides great flexibility and reliability for the finishing process, because robots allow for more degrees of freedom than other machines. The process has been successfully implemented in several industries.

Future research: Future research will address surface finishing on the part and generation of the patterns of the points on the surface using micros within DELMIA®. Anticipated work includes establishing prototypes of deburring cells which integrates with DELMIA® in any surface finishing application.

REFERENCES

Edwin, A.E., 2000. Robotic Grinding of Parts to Remove Machine tool Scallops and Mismatch. Industrial Robot: An Int. J., 27 (1): 20-29. DOI: 10. 1108/01439910010305130. http://www.emeraldinsight com/10.1108/01439910010305130.

Godwin, L.E., 1996. Programming with Force Control. The RIA Grinding, Deburring and Finishing workshop St. Paul, Minnesota. http://www.pushcorp.com/Tech% 20Papers/Programming-with-Force-Control.pdf.

Roberts, R.K., T.W. Engel and F. Proctor, 1992. Specification of an active force control tool for performing deburring and chamfering on a robot platform. Industrial Electronics, Control, Instrumentation and Automation. Power Electronics and Motion Control. Proceedings of the International Conference, 2 (9-13): 918-926. DOI: 10.1109/IECON. 1992.254482.

Stouffer, K.A. and R. Robert Jr., 1995. ADACS: An Advanced Deburring and Chamfering System. National Conference Publication-Institution of Engineers Australia NCP, pp: 623-628. http://www. isd.mel.nist.gov/documents/stouffer/icme95.pdf.

Shiakolas, P.S., D. Labalo and J.M. Fitzgerald, 1999. RobSurf: A Near Real OLP System for Robotic Surface Finishing. Proceedings of the 7th Mediterranean Conference on Control and Automation (MED). Haifa, Israel, June, 1999. http://med.ee.nd.edu/MED7/med99/papers/MED12 8.pdf.