IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received July 30, 2020, accepted August 15, 2020, date of publication August 18, 2020, date of current version August 31, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3017561

STEP-NC Architectures for Industrial Robotic Machining: Review, Implementation and Validation

ALBERTO J. ÁLVARES[1], EFRAIN RODRIGUEZ[1], CRISTHIAN I. RIAÑO JAIMES[1], JUAN S. TOQUICA[1], AND JOÃO C. E. FERREIRA[2]

[1]Department of Mechanical Engineering, University of Brasília, Brasília 70910-900, Brazil
[2]Department of Mechanical Engineering, Federal University of Santa Catarina, Florianópolis 88040-900, Brazil

Corresponding author: Efrain Rodriguez (efrainrg2009@gmail.com)

This work was supported in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), Brazil, under Grant 001, and in part by the Conselho Nacional de Desenvolvimento Científico e Tecnológico under Grant 310229/2013-8.

ABSTRACT In the manufacturing sector, industrial robots are being increasingly improved to execute machining tasks as they exhibit significant advantages in terms of flexibility, cost-effectiveness, affordability, and larger work-space when compared to traditional computer numeric control (CNC) machines. However, programming this kind of equipment for robotic machining is complex, due to closed architecture controller and proprietary programming languages limitations. For that reason, this work aims at contributing to the adoption of the STEP-NC standard (STandard for the Exchange of Product model data- Numerical Control (ISO 10303-238 and ISO 14649)), generating programs for robotic machining operations. The STEP-NC data model enables the integration of information from design, process planning, simulation, manufacturing, and even inspection in a single platform, which could create new alternatives for industrial robotic machining programming. In this context, several previous studies are described in this manuscript aiming to highlight the contribution of this work, in addition to the analysis, implementation, and validation of six different STEP-NC architectures describing the advantages that each architecture provides for achieving robotic machining capabilities. Each introduced architecture can successfully generate a STEP-NC robotic machining program, either as ISO 10303-238 or ISO 14649, which are validated in a simulation environment with both a virtual robot model and a real industrial robot equipped with a LinuxCNC controller. This approach can be implemented in different industrial robots.

INDEX TERMS Computer numerical control, data modelling, ISO 10303 AP238, linuxCNC controller, robotic machining, STEP-NC standard.

I. INTRODUCTION

Industrial robots have been generally applied in tasks such as painting, packaging, welding, and pick-and-place operations. In the last few years they have been increasingly used to perform machining tasks as well. Compared with traditional machine tools, industrial robots provide significant advantages in terms of flexibility, cost-effectiveness, affordability, and larger work-space. Furthermore, robots can allow new machining strategies for highly complex parts since they have more degrees of freedom than conventional machine tools. However, one of the substantial limitations that hinder the widespread adoption of industrial robots is their low stiffness, mainly in machining with high material removal rates [1]–[4]. In this context, a significant amount of research on robotic machining has been done over the past decades, and a review of some of those works was presented in [3].

An additional problem related to the use of robots in machining is the difficulty to integrate robotic systems present in the factory's numerical chain, since most robots tend to use closed architecture controllers and proprietary programming languages. This problem has motivated industry leaders to implement more efficient and standardized programming methods towards production approaches such as smart manufacturing [5]. In this context, STEP-NC emerges as a new CNC programming standard that provides means to integrate robotic systems with the upstream systems of the numeric chain.

The associate editor coordinating the review of this manuscript and approving it for publication was Ming Luo.

Ji and Wang [3] and Chen and Dong [2] reviewed many articles with applications in robotic machining. They identified the following research trends in the context of robotic machining: robotic machining dynamics, robotic machining configurations, tool path optimization, machining processes, machining quality, monitoring and compensation, calibration, STEP-NC standard, simulation, and energy efficiency.

First normalized as ISO 14649 [6] and after as ISO 10303-AP238 [7], STEP-NC provides a high-level object-oriented data model that extends the data models of the STEP ISO 10303 standard (STEP) to encompass NC process data. STEP-NC was developed due to the need of a new Computer Numerical Control (CNC) programming standard to replace the old G-code [8] and to enable information feedback from the manufacturing and inspection to upstream stages as product design and process planning [9]. The STEP-NC data models for milling and turning processes have already been developed as Application Reference Models (ARM) in ISO 14649 since 2004. These models were mapped to the ISO 10303 AP238 incorporating some modifications in a single Application Interpreted Model (AIM) to achieve integration with other resources of STEP [7]. Demonstrations of the integration of the CAD-CAM-Simulation-CNC chain provided by STEP-NC according to ISO 10303 AP238 are reported in [10]. Recently, the second edition of ISO 10303-238 has been announced with improvements that include new definitions for automated measurement, and new support for additive manufacturing, composite tape layup, and multi-robot drill and fill [11].

Some publications proposed the use of STEP-NC in robot machining [12], and a concept was also tested [13]. Rea Minango and Ferreira [14] combined STEP-NC and forward/inverse kinematics methods to generation tool paths for different types of robots. Zivanovic, Slavkovic, and Milutinovic [15], [16], Toquica *et al.* [17], [18] implemented a STEP-NC-compliant module for robotic machining.

Xiao *et al.* [19] developed a so-called STEP-compliant Industrial Robot Data Model (IRDM), which is used for data exchange between CAx systems and robot off-line programming systems. Geometric and mathematical aspects of IRs were defined in IRDM, which can be used to represent kinematic and dynamic behaviors of the robots.

In this context, this paper contains additional contributions to the development of STEP-NC architectures applied to robotic machining. Six different STEP-NC architectures are analyzed, implemented and validated. Three of presented architectures are novel contributions, while the other three are extensions of previous works by the authors [14], [17], [18]. These architectures promote the dissemination and application of STEP-NC in robotic machining by using some commonly used commercial CAD/CAM software such as Mastercam, Siemens NX and PTC Creo.

The new software tools developed in this work are available on the web, and they include: 1) a module that generates virtual robot models for simulation in the STEP-NC Machine software; 2) an adapter software that can convert data in APT CL file format to STEP-NC in AP238; and 3) a software that works as a plug-in for LinuxCNC [20] allowing to convert a STEP-NC file to G-code, which can be executed on the industrial ASEA robot. Complying with the requirements of the Industry 4.0 context, related to Cloud Robotics and Data Analytic concepts [21].

The remainder of this paper is organized as follows. Section II contains a literature review on robot machining, including some STEP-NC-based architectures for robot machining. Section III describes the six proposed architectures for STEP-NC-based industrial robot machining. Section IV presents a case study for each architecture. Finally, section V contains the conclusion and suggestions for future research works.

II. PREVIOUS RELATED WORKS

A. STEP-NC-BASED CAD/CAM/CNC SYSTEMS

The first prototype of the STEP-NC platform was developed in the European STEP-NC project Esprit [22]. A STEP-NC file was generated by a solution using Catia and Open Mind CAD software [23]. This file was then interpreted by an enhanced Siemens 840D controller.

A STEP-compliant interpreter, named Agent-Based Computer Aided Manufacture system (AB-CAM), was presented by Newman *et al.* [24]. Their system generated an ISO 14649 file, which was interpreted into G-code for machining. In other works reported by the Professor Newman's team, they proposed STEP-NC-compliant data models and different software tools for turning, wire electrical discharge machining, and inspection [25]–[27].

Other platform using interpreted STEP-NC was proposed by Suh *et al.* [28], [29]. The platform enables controlling the axes of milling and turning machine tools. However, that platform was not implemented in an industrial machine. Another solution of interpreted STEP-NC was developed by Xu [30], which allows the manufacture of parts without using G-code.

An implementation of STEP-NC on an industrial CNC controller was developed by Laguionie *et al.* [31] and Rauch *et al.* [23]. The platform enables indirect STEP-NC and can be implemented on many CNC controllers, using different commercial CAD/CAM systems. The platform was implemented and validated on a high speed machining machine tool, as well as a laser cladding machine [32].

Valilai and Houshmand [33] proposed a platform based on STEP data models for collaborative and interoperable product design and manufacture using different CAD/CAPP/CAM/CNC pieces of software. Their platform is comprised of different layers, which enable different pieces of software to work in an integrated manner using their internal data structures.

A prototype of a CAD/CAM/CNC platform based on ISO 14649 was developed by Xiao *et al.* [34], and it shows data flow between a STEP-compliant CAD/CAM system and a STEP-CNC system.

Seeking to develop an open CNC for a five-axis high-speed milling machine, Bensoussan *et al.* [35] proposed a method to generate the tool path directly on the surface to be machined in order to avoid approximations, leading to a smoother tool path. Their method considers the kinematic limitations of a high-speed milling machine. The experimental validation of their method requires a machine equipped with an open CNC.

Álvares *et al.* [36] presented an architecture of a CNC compliant with STEP-NC [6] by means of an adapter that converts a program in the STEP-NC format to G-code. The STEP-NC controller is based on the open source LinuxCNC controller, which is compliant with the ISO 6983 standard (G-code) [8]. In order to validate the architecture, a machine tool was built, being controlled by LinuxCNC. The controller is integrated with the STEP part 21 data model (ISO 10303 standard) using an adapter that reads a STEP-NC file. The LinuxCNC controller runs the STEP-NC file seamlessly, and an example part was machined using the system.

Another CAD/CAM/CNC platform was presented by Zivanovic and Puzovic [37] and Živanović and Vasilić [38]. They used the commercial CAD/CAM software Creo for design, process planning and simulation, as well as the STEP-NC machine software [39] to generate an AP238 program.

Srivastava and Komma [40] pointed out that AIM (Application Interpreted Model) format of STEP-NC is more sustainable than ARM (Application Reference Model) format and can be used throughout the life cycle of product. The AIM format of STEP-NC evolved as being effective for numeric control and intelligent machining of products. Srivastava and Komma [40] also discussed the development for interface for STEP-NC (AIM) for tool path generation to machine the outer contour of a part having linear and circular profiles. The interface is implemented using Java Standard Data Access Interface (JSDAI) [41], and the generated codes are verified on STEP-NC Machine software from STEP Tools [39].

Ferreira *et al.* [42] presented a CAD/CAPP/CAM system compliant with the ISO 14649 standard for the design, process planning, and manufacture of parts having general shapes, which can be executed by anyone through the Internet, referred to as STEP Modeler. The parts are built with the use of features, which can contain general surfaces, enabling the user to design complex parts. The process planning method considers the possibility of using cutting tools with different diameters for removing different amounts of material. The physical STEP-NC file is generated automatically by the system, and it is used as input to the module that simulates graphically the tool paths for machining the part in a three-axis CNC machine.

A set of tools for manufacturing based on ISO 10303 AP238 was proposed in STEPTools [39]. The main module of the platform is called STEP-NC Machine, which can generate ISO 10303 AP238 files. It also contains modules for simulation, inspection, real-time optimization and product design.

Faraz *et al.* [43] developed a STEP-compliant design and manufacturing framework for discrete sheet metal bend parts, which uses STEP (ISO 10303), AP203, AP207, and AP219 along with STEP-NC (ISO14649) to provide a basis of vendor neutral data modeling with capability of bidirectional communication between design and manufacturing.

The availability of STEP-NC software plug-ins for CAM software can contribute to the dissemination of STEP-NC. A STEP-NC plug-in converts a STEP-NC file into the data structures of an existing CAM system. Currently, plug-ins are available for GibbsCAM and Mastercam. With these plug-ins a shop can receive a customer's STEP-NC files and manufacture parts on existing CNC equipment.

B. ROBOT PROGRAMMING

According to the ISO 8373-2 standard [44], a robot is an automatically controlled, multipurpose manipulator, programmable in three or more axes, which may be either stationary or mobile for use in industrial automation applications.

The insertion of industrial robots in the global industrial scenario in 2016 was the most important in the robotics history, resulting in the sale of more than 290,000 units during that year, an increase of 14% in comparison with 2015 [45]. The most important sectors regarding industrial robots use is the automotive, electrical/electronics and metal and machinery industry, amounting to 70% [45]. Industrial robots have increased their presence in manufacturing significantly, reaching a record number of about 100,000 units in the automotive industry in 2015 [46].

Xu [47] highlights the key role of industrial robots for manufacturing in the concept of Industry 4.0, due to their ability to contribute to agile, advanced and intelligent manufacturing.

Industrial robots have more flexibility in machining than conventional machine tools, due to the larger amount of degrees of freedom (DOF) in the mechanical structure of manipulators. High percentage of industrial robots are used for handling and welding, since these processes have lower precision when compared with machining. However, several works have been developed in order to analyse and reduce machining error using robots, besides improving robot machining performance [48].

The generated cutter location file (CL-file) [49] needs to be converted into a kinematics machine language, but industrial robots do not have a standard programming language. This contributes significantly to CAD/CAM scenarios not to support post-processing for robots [12], consequently not allowing single-platform machining programming methods.

Ríbeíro textitet al. [50] developed a robot software that can carry out robot programming as well as simulation like a 5-axis machining center or 3D printer. It can convert G-code or CL-file programs to robot programs for different robots. Also, the paths generated by their software avoid singularities, axis limits and collisions [50].

Alvares *et al.* [51] presented a comparative study of numerical controllers integration using LinuxCNC controller and

MatLab/Mach3 in Windows solutions applied to a retrofitting process of an old industrial robot. The authors adopted different robot control platforms, and included robot kinematics. Their study considered performance in terms of computational cost, ease of implementation, integration with CAD/CAM, and usability, analyzing the advantages and disadvantages of each architecture. Finally, the most appropriate CNC was chosen, which was LinuxCNC.

Alvares *et al.* [20] presented a methodology that can be applied to CNC retrofitting of any industrial robot with serial or parallel kinematics, which guides the developer in steps associated with hardware and software specification for a desired robotic platform implementation as a custom solution based on the LinuxCNC system.

Zivanovic *et al.* [15] presented a methodology for programming, simulation, and execution of industrial robotic machining based on the STEP-NC ISO 10303-238 standard. The NC program, containing essentially machining tool paths, is generated by using commercial CAD/CAM software like PTC Creo, and then it is imported into the STEP-NC Machine software environment for simulation with the 3D kinematic model of the robot. STEP-NC Machine generates the STEP-NC AP238 robotic machining program in P21 format. They use a software module named RoboSTEP-NC, which converts the STEP-NC AP238 machining program to the robot's own programming language, whose control unit is based on the EMC2 software. In another work, Slavkovic *et al.* [16] applied a similar approach to generate and simulate STEP-NC AP238 machining programs for industrial robotic machining using a Mitsubishi Movemaster EX robot.

C. STEP-NC-BASED MANUFACTURING USING ROBOTS

According to Toquica *et al.* [17], the potential of CAD/CAM solutions available for off-line machining programming is not fully exploited because of the popular G-code, which enables the description of only elementary actions and cutting tool moves. This is a drawback in the context of Industry 4.0, which requires the presence of smart equipment at all levels of the production chain, ensuring the availability of high-level information from design to manufacturing.

Given this scenario, the STEP-NC standard provides opportunities for robot controllers to support high-level information, including design content [38]. STEP-NC can also be used to overcome programming limitations, as well as contribute to flexible and advanced manufacturing requirements of Industry 4.0 [47].

With regard to machining operations, machine tools have been used because of their higher stiffness and machining accuracy when compared with industrial robots [52]. However, according to Pan *et al.* [53], in an environment where product variety is high and life cycle is short, industrial robots can contribute to increase productivity and flexibility to meet these challenges.

In industrial robots applications such as part transfer and assembly, the paths are usually relatively simple [19]. However, machining operations are usually carried out along more complex tool paths [14], and its planning is normally difficult and time-consuming [53]. Thus, the development of robot off-line programming systems becomes necessary [54].

A unified manufacturing resource model was proposed by Vichare *et al.* [55] to provide the necessary information to specify various elements of a CNC machining system (including robotic arms), as well as support for automating decision making regarding process planning.

An open architecture of an intelligent CNC based on parameter-driven technology for multi-robot manipulation operations was presented by Song *et al.* [56], in which the CNC system was implemented in a H-beam machine for cutting steel.

Although these research works showed the importance of developing STEP or STEP-NC-based systems for manufacturing integration and execution, they did not provide some solutions for robotic machining.

Solvang *et al.* [12] implemented a STEP-NC-based CAM system for robot machining. The system reads the STEP-NC file and, then, it is parsed, translated, simulated in a virtual robot, and used by a real industrial robot, which did not perform machining. The system was developed in the DELMIA software environment.

Xiao *et al.* [19] proposed a STEP-compliant robot data model to exchange data between CAx systems and robot off-line programming systems. Their data model includes geometric and mathematic aspects of industrial robots. A robot off-line programming system with 3D virtual environment was presented in their work, but no real machining was carried out.

A STEP-NC-based method was proposed by Rea Minango and Ferreira [14] that uses forward and inverse kinematics procedures that can be applied to serial and hybrid robots, allowing the generation of robot paths for machining. Those paths were tested on virtual robots with different morphologies, and no real machining was performed.

Toquica *et al.* [17] and Toquica *et al.* [18] presented a high-level programming solution based on STEP-NC, using a CAD/CAM software for industrial robots machining. A complete platform has been developed to enable advanced and intelligent manufacturing with the possibility of integrating several modules for simulation, optimization, and visualization, as well as in-process manufacturing adaptation, cloud manufacturing, and machine learning with database analytics.

Given the above scenario, it is proposed to implement six architectures for STEP-NC-compliant industrial robot machining. STEP-NC was used because it has not machine or robot dependency, enabling the development of an intelligent and universal off-line programming solution. The ASEA IRB6 robot with 5 DOF was used in this work to validate STEP-NC-based architectures. This robot had its S2 controller updated, retrofitted and made compatible with the LinuxCNC Controller [20].

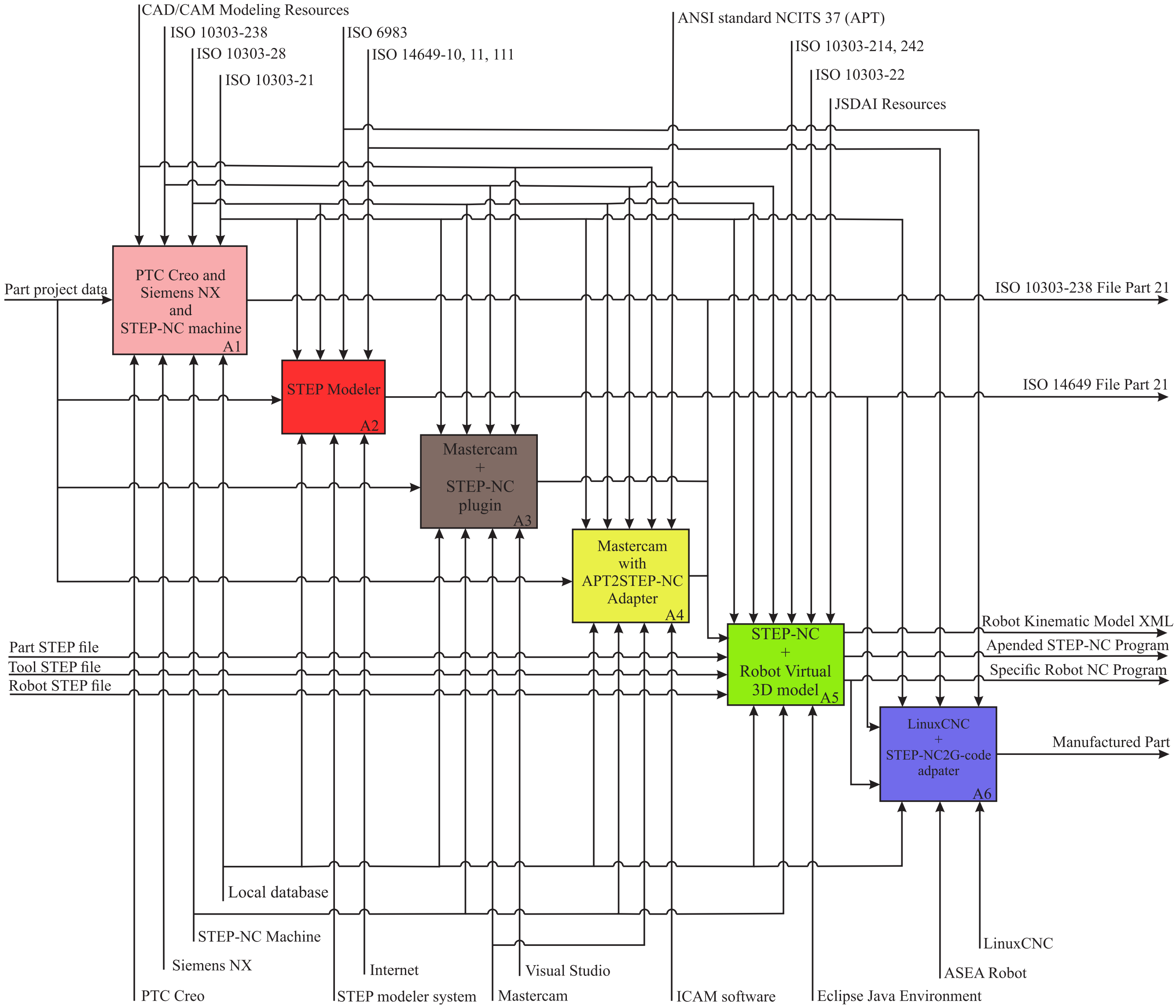


FIGURE 1. Proposed STEP-NC-based architectures for machining with industrial robots.

III. PROPOSED STEP-NC ARCHITECTURES FOR MACHINING USING INDUSTRIAL ROBOTS

This research is focused on implementing and validating different architectures to export STEP-NC files to be used in industrial robotic machining. These architectures use different software tools including some commercial CAD/CAM systems as well as new tools developed by authors. The functional model shown in Figure 1 presents the system requirements outlined as inputs, outputs, mechanisms, and controls. The model encompasses the proposed functional perspective, the flow of information, and the hierarchy of activities carried out for the development of the proposed architectures. The proposed architectures are listed below, and the associated functional model is represented by IDEF0 (Application Activity Model) in Figure 1:

1) ASEA virtual robot machining simulation using STEP-NC Machine (Step Tools technology and virtual robot model developed by the authors);
2) ASEA robot with LinuxCNC STEP-NC (ISO 14649) controller (developed by the authors);
3) STEP Modeler: CAD/CAPP/CAM/NC (developed by the authors);
4) PTC Creo or Siemens NX CAD/CAM with CLFile and STEP-NC Machine (Step Tools technology and virtual robot model developed by the authors);
5) Mastercam CAD/CAM with AP238 plug-in from STEP Tools (Step Tools technology);
6) PTC Creo, MasterCam or Siemens NX CAD/CAM with CLFile and adapter CLFile to AP238 (APT2STEP-NC, developed by the authors).

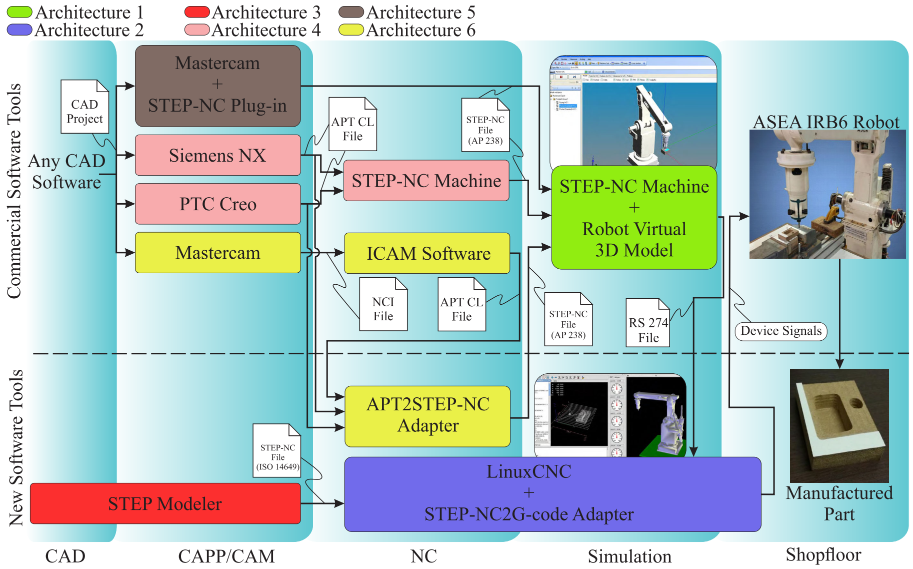


FIGURE 2. STEP-NC digital thread for machining with industrial robots.

Each architecture can generate STEP-NC files that are simulated in a digital scenario using a virtual model of the ASEA IRB6 industrial robot and, then, executed in a real (physical) ASEA IRB6 LinuxCNC robot with controller compatible with STEP-NC and G-code. STEP-NC Machine is used by all architectures, being presented as a simulation module, since it allows simulating STEP-NC files including the virtual model of the robot.

Fig. 2 presents the STEP-NC digital thread for industrial robotics machining using the six proposed architectures. The data flow is presented along the digital value chain, starting with CAD/CAPP/CAM systems, CNC compatible with RS-274, ISO 14649, and AP238 standards, and finally the Asea virtual robot and Asea real robot that manufacture the example parts. A description of these architectures is given in the next sections.

A. ASEA VIRTUAL ROBOT MACHINING SIMULATION USING STEP-NC MACHINE

This section describes the ASEA robot configuration and simulation using a virtual model within the STEP-NC Machine environment. Virtual simulation is performed to verify and validate the STEP-NC file obtained from each architecture proposed in this work. The architectures present different alternatives for the implementation of the STEP-NC standard (ISO 14649 and ISO 10303 AP238), and the simulation scenario allows verifying the integrity of each program generated within each architecture. The simulation environment provided by the STEP-NC Machine shows how a virtual ASEA robot configuration behaves during a machining task with information extracted from a STEP-NC file.

Toquica *et al.* [17] presented a manual method for generating the ASEA robot virtual model. Their method has been automated in this work by means of a software for automatic generation of virtual models of robots and machine tools with serial or parallel kinematics. This tool was developed in order to allow the generation of the virtual model of any compatible machine in the STEP-NC Machine platform. A free version of this software is available on the URL: https://tinyurl.com/y5tmw59v.

Figure 3 shows the data flow between the activities related to the configuration of STEP-NC Machine to use the ASEA robot as a machine tool in the simulation of machining parts. The diagram is composed of three functional blocks that describe the three main activities in this section.

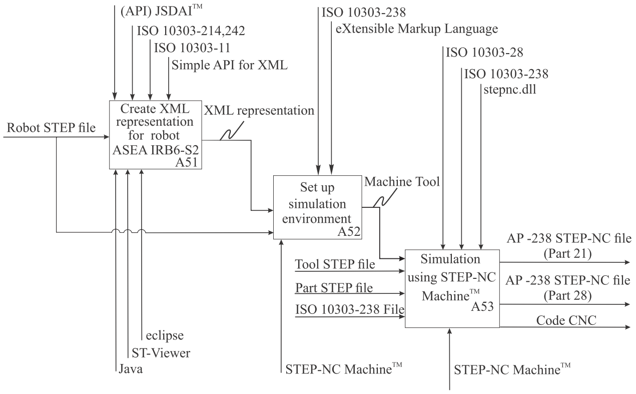
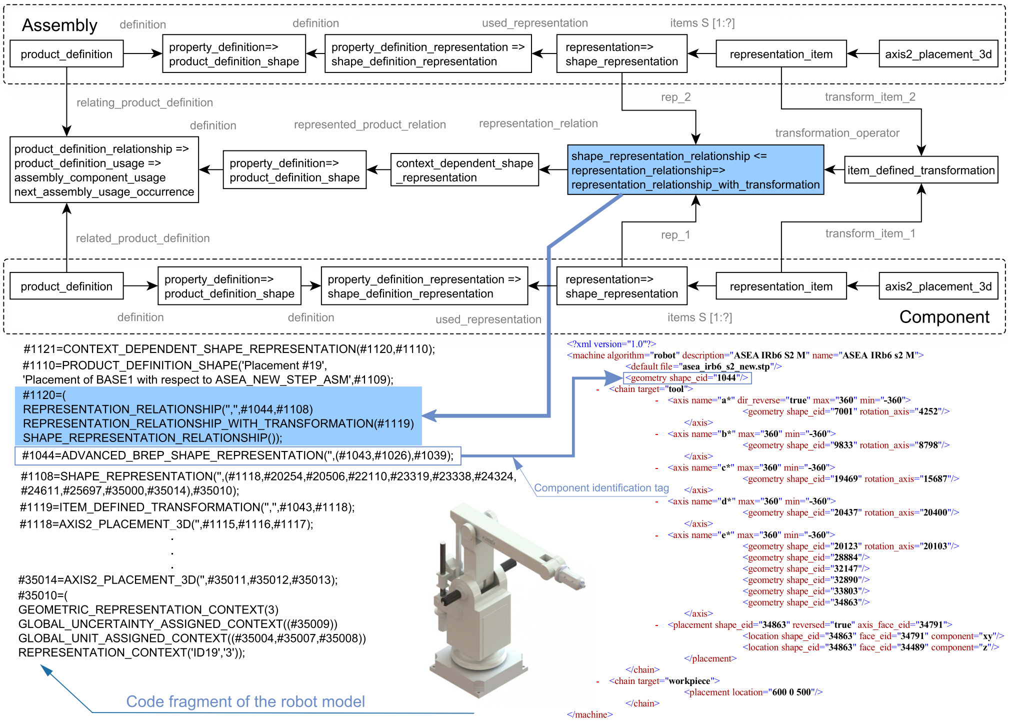


FIGURE 3. Data flow between activities necessary for simulation in STEP-NC Machine.

Block A51 in Figure 3 represents the activities to create an XML representation file for the ASEA IRB6-S2 robot, and this XML file is used to configure the virtual robot for simulation. Block A52 refers to the activities to prepare the machining simulation scenario. Block A53 contains the activities related to executing the simulation.


Assembly
definition
definition
used_representation
items S [1:?]
product_definition
property_definition=> product_definition_shape
property_definition_representation => shape_definition_representation
representation=> shape_representation
representation_item
axis2_placement_3d
relating_product_definition
rep_2
transform_item_2
definition
represented_product_relation
representation_relation
transformation_operator
product_definition_relationship => product_definition_usage => assembly_component_usage next_assembly_usage_occurrence
property_definition=> product_definition_shape
context_dependent_shape _representation
shape_representation_relationship <= representation_relationship=> representation_relationship_with_transformation
item_defined_transformation
related_product_definition
rep_1
transform_item_1
product_definition
property_definition=> product_definition_shape
property_definition_representation => shape_definition_representation
representation=> shape_representation
representation_item
axis2_placement_3d
definition
definition
used_representation
items S [1:?]
Component
Component identification tag
Code fragment of the robot model

```
#1121=CONTEXT_DEPENDENT_SHAPE_REPRESENTATION(#1120,#1110);
#1110=PRODUCT_DEFINITION_SHAPE('Placement #19',
'Placement of BASE1 with respect to ASEA_NEW_STEP_ASM',#1109);
#1120=(
REPRESENTATION_RELATIONSHIP('','',#1044,#1108)
REPRESENTATION_RELATIONSHIP_WITH_TRANSFORMATION(#1119)
SHAPE_REPRESENTATION_RELATIONSHIP());
#1044=ADVANCED_BREP_SHAPE_REPRESENTATION('',(#1043,#1026),#1039);
#1108=SHAPE_REPRESENTATION('',(#1118,#20254,#20506,#22110,#23319,#23338,#24324,
#24611,#25697,#35000,#35014),#35010);
#1119=ITEM_DEFINED_TRANSFORMATION('','',#1043,#1118);
#1118=AXIS2_PLACEMENT_3D('',#1115,#1116,#1117);
                              .
                              .
                              .
#35014=AXIS2_PLACEMENT_3D('',#35011,#35012,#35013);
#35010=(
GEOMETRIC_REPRESENTATION_CONTEXT(3)
GLOBAL_UNCERTAINTY_ASSIGNED_CONTEXT((#35009))
GLOBAL_UNIT_ASSIGNED_CONTEXT((#35004,#35007,#35008))
REPRESENTATION_CONTEXT('ID19','3'));
```

```
<?xml version="1.0"?>
<machine algorithm="robot" description="ASEA IRb6 S2 M" name="ASEA IRb6 s2 M">
          <default file="asea_irb6_s2_new.stp"/>
          <geometry shape_eid="1044"/>
  -    <chain target="tool">
            -   <axis name="a*" dir_reverse="true" max="360" min="-360">
                            <geometry shape_eid="7001" rotation_axis="4252"/>
                    </axis>
            -   <axis name="b*" max="360" min="-360">
                            <geometry shape_eid="9833" rotation_axis="8798"/>
                    </axis>
            -   <axis name="c*" max="360" min="-360">
                            <geometry shape_eid="19469" rotation_axis="15687"/>
                    </axis>
            -   <axis name="d*" max="360" min="-360">
                            <geometry shape_eid="20437" rotation_axis="20400"/>
                    </axis>
            -   <axis name="e*" max="360" min="-360">
                            <geometry shape_eid="20123" rotation_axis="20103"/>
                            <geometry shape_eid="28884"/>
                            <geometry shape_eid="32147"/>
                            <geometry shape_eid="32890"/>
                            <geometry shape_eid="33803"/>
                            <geometry shape_eid="34863"/>
                    </axis>
            -   <placement shape_eid="34863" reversed="true" axis_face_eid="34791">
                            <location shape_eid="34863" face_eid="34791" component="xy"/>
                            <location shape_eid="34863" face_eid="34489" component="z"/>
                    </placement>
          </chain>
  -    <chain target="workpiece">
                    <placement location="600 0 500"/>
          </chain>
</machine>
```

FIGURE 4. **Scheme of the reference paths processed to extract the attributes used in the XML file.**

The files used as system inputs include the virtual robot model in AP242 or AP214 formats, part model to be manufactured in AP238 STEP file format, and the cutting tool model. The portions of software used as mechanisms to enable simulation are: the development environment for Eclipse Java, JSDAI Application Programming Interface (API), ST-Viewer, and STEP-NC Machine. Outputs generated by the simulation are AP238 files enriched with alterations and parameter settings suitable for real manufacturing. It can also be exported as CNC compatible code for specific machine tools.

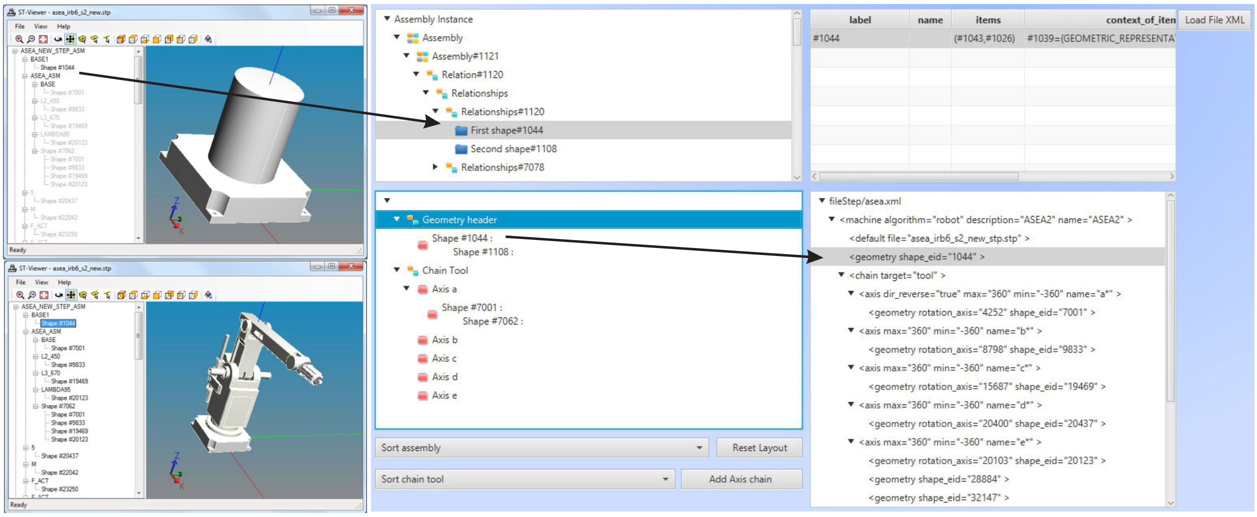
The simulation process begins with the generation of the virtual model: each robot component involved in the kinematic control is stored as an independent part. Each part is positioned individually in an assembly to build a functional kinematic arrangement of the robot, being necessary to define the position relationships and constraints. Finally, when the assembly is built, it is exported in STEP AP242 or AP214 formats. These two application protocols share the units of functionality necessary to describe models based on the definition.

The virtual robot model needs the XML file to be used as a machine tool within the simulation scenario. The XML file contains the specific data on the kinematic chain of the robot. The STEP file with the virtual robot model contains the robot assembly information, which can be exported from any CAD software.

The program was developed to simplify the identification of labels associated with each component involved in the kinematic model of the robot. The creation of the XML file starts with processing of STEP file (AP-242 or AP-214) in order to extract useful information to configure the parameters of the kinematic model of the robot within the XML file. The program uses Express schemes provided by the STEP standard to identify the explicit hierarchical structures that represent the assemblies as well as the components and their relationships. JSDAI Application Programming Interface (API) is used to read and manipulate object-oriented data defined in these EXPRESS data models.

Figure 4 presents a model with the reference paths followed in interpreting the requirements of the application used to generate the XML file. by means of these reference paths, the different integrated resources are explored and the attributes of the entities extracted, being presented briefly in the interface. Each element used in the ISO 10303 standard (STEP) has a specific reference route, which allows the creation of different applications for different purposes. In the case at hand, the developed application extracts only the information about the assembly and its components, and presents the necessary attributes to create the XML file.

The reference paths that structure the assembly information are presented at the top of Figure 4. An excerpt of the robot's virtual model code with the information of a component is shown in the lower left part of the figure. The application creates a list of the elements and attributes of each entity. This information is useful to identify each element of the robot's kinematic model and configure the label in the XML file. The XML file generated for the simulation of the



```
ST-Viewer - asea_irb6_s2_new.stp
File  View  Help
ASEA_NEW_STEP_ASM
  BASE1
    Shape #1044
  ASEA_ASM
    BASE
      Shape #7001
    L2_450
      Shape #9833
    L3_670
      Shape #19469
    LAMBDA95
      Shape #20123
    Shape #7062
      Shape #7001
      Shape #9833
      Shape #19469
      Shape #20123
  5
    Shape #20437
  M
    Shape #22042
  F_ACT
    Shape #23250
Ready
```

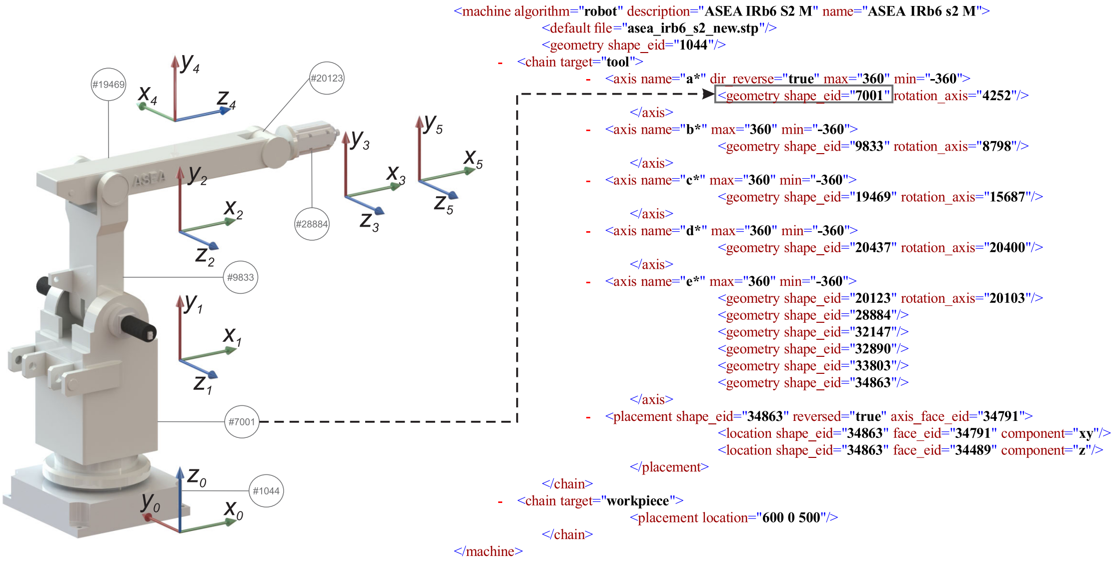
```
▼ Assembly Instance
  ▼ Assembly
    ▼ Assembly#1121
      ▼ Relation#1120
        ▼ Relationships
          ▼ Relationships#1120
              First shape#1044
              Second shape#1108
          ▶ Relationships#7078
```

label	name	items	context_of_iten
#1044		(#1043,#1026)	#1039=(GEOMETRIC_REPRESENTA

```
▼
  ▼ Geometry header
      Shape #1044 :
          Shape #1108 :
  ▼ Chain Tool
    ▼ Axis a
        Shape #7001 :
            Shape #7062 :
      Axis b
      Axis c
      Axis d
      Axis e
```

```
▼ fileStep/asea.xml
  ▼ <machine algorithm="robot" description="ASEA2" name="ASEA2" >
      <default file="asea_irb6_s2_new_stp.stp" >
      <geometry shape_eid="1044" >
    ▼ <chain target="tool" >
      ▼ <axis dir_reverse="true" max="360" min="-360" name="a*" >
          <geometry rotation_axis="4252" shape_eid="7001" >
      ▼ <axis max="360" min="-360" name="b*" >
          <geometry rotation_axis="8798" shape_eid="9833" >
      ▼ <axis max="360" min="-360" name="c*" >
          <geometry rotation_axis="15687" shape_eid="19469" >
      ▼ <axis max="360" min="-360" name="d*" >
          <geometry rotation_axis="20400" shape_eid="20437" >
      ▼ <axis max="360" min="-360" name="e*" >
          <geometry rotation_axis="20103" shape_eid="20123" >
          <geometry shape_eid="28884" >
          <geometry shape_eid="32147" >
```

FIGURE 5. XML file generation.



```
<machine algorithm="robot" description="ASEA IRb6 S2 M" name="ASEA IRb6 s2 M">
      <default file="asea_irb6_s2_new.stp"/>
      <geometry shape_eid="1044"/>
  - <chain target="tool">
      - <axis name="a*" dir_reverse="true" max="360" min="-360">
                <geometry shape_eid="7001" rotation_axis="4252"/>
          </axis>
      - <axis name="b*" max="360" min="-360">
                <geometry shape_eid="9833" rotation_axis="8798"/>
          </axis>
      - <axis name="c*" max="360" min="-360">
                <geometry shape_eid="19469" rotation_axis="15687"/>
          </axis>
      - <axis name="d*" max="360" min="-360">
                <geometry shape_eid="20437" rotation_axis="20400"/>
          </axis>
      - <axis name="e*" max="360" min="-360">
                <geometry shape_eid="20123" rotation_axis="20103"/>
                <geometry shape_eid="28884"/>
                <geometry shape_eid="32147"/>
                <geometry shape_eid="32890"/>
                <geometry shape_eid="33803"/>
                <geometry shape_eid="34863"/>
          </axis>
      - <placement shape_eid="34863" reversed="true" axis_face_eid="34791">
                <location shape_eid="34863" face_eid="34791" component="xy"/>
                <location shape_eid="34863" face_eid="34489" component="z"/>
          </placement>
      </chain>
  - <chain target="workpiece">
          <placement location="600 0 500"/>
      </chain>
</machine>
```

FIGURE 6. XML representation of the virtual robot ASEA IRB6.

ASEA robot is presented in the lower right part of Figure 4. It can be noticed that the label that identifies the component in the model file is finally part of the XML file.

An example of the process followed to generate the XML file is shown in Figure 5. In order to configure the different parameters correctly, it is necessary to identify the reference labels of each component in the kinematic chain within the STEP file of the robot model. Alternatively, the ST-Viewer software is used to assist this identification phase. ST-Viewer also presents the view of the virtual robot model, information of the reference paths useful to define precisely the geometry labels in the shape_eid attribute and the surface in the face_eid attribute of the XML file.

The positions of each rotational joint of the ASEA IRB6-S2 robot are also defined. In the file, five axes (a, b, c, d, e) are created and configured, which represent the robot's degrees of freedom. Figure 6 shows the result of the final configuration of the XML file, which enables virtual simulation within STEP-NC Machine.

With the XML file created and configured, it is stored next to the virtual model of the robot in the STEP-NC Machine installation folder together with the files of the available machine tools. The ASEA IRB 6-S2 robot is enabled to be used as a machine tool in the simulation. Figure 7 shows the scenario ready to run the simulation, and the programs generated by means of each architecture are verified.

B. ASEA ROBOT WITH LinuxCNC STEP-NC (ISO 14649) CONTROLLER

This section describes the architecture where two new functions are added into LinuxCNC controller: (a) run the STEP-NC file directly into the LinuxCNC controller,

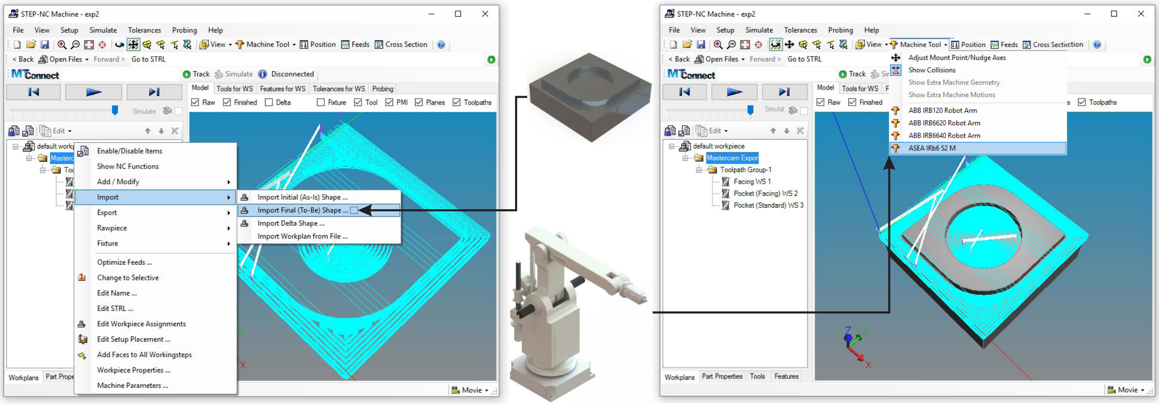

STEP-NC Machine - exp2
File View Setup Simulate Tolerances Probing Help
View Machine Tool Position Feeds Cross Section
< Back Open Files Forward > Go to STRL
MTConnect
Track Simulate Disconnected
Model Tools for WS Features for WS Tolerances for WS Probing
☑ Raw ☑ Finished □ Delta □ Fixture ☑ Tool ☑ PMI ☑ Planes ☑ Toolpaths
Simulate
Edit
default workpiece
Enable/Disable Items
Show NC Functions
Add / Modify
Import
Export
Rawpiece
Fixture
Optimize Feeds ...
Change to Selective
Edit Name ...
Edit STRL ...
Edit Workpiece Assignments
Edit Setup Placement ...
Add Faces to All Workingsteps
Workpiece Properties ...
Machine Parameters ...
Import Initial (As-Is) Shape ...
Import Final (To-Be) Shape ...
Import Delta Shape ...
Import Workplan from File ...
Workplans Part Properties
Movie
STEP-NC Machine - exp2
File View Setup Simulate Tolerances Probing Help
View Machine Tool Position Feeds Cross Secticction
Adjust Mount Point/Nudge Axes
Show Collisions
Show Extra Machine Geometry
Show Extra Machine Motions
ABB IRB120 Robot Arm
ABB IRB6620 Robot Arm
ABB IRB6640 Robot Arm
ASEA IRb6 S2 M
default workpiece
Mastercam Expor
Toolpath Group-1
Facing WS 1
Pocket (Facing) WS 2
Pocket (Standard) WS 3
Workplans Part Properties Tools Features
Z
Y
X
Movie

FIGURE 7. Configuration of the simulation scenario.

converting it to G-code automatically and transparently to the machine operator by means of the adapter described in the previous section; (b) visualize the STEP-NC file by means of an additional Graphical User Interface (GUI) to LinuxCNC.

In order to run the adapter in LinuxCNC, the startup configuration file ('.INI') between the machine and LinuxCNC should be modified, which corresponds to the GUI of the controller AXIS interface. AXIS is a GUI for LinuxCNC that displays both the G-code and the tool path, each in a different window.

In the INI file of the AXIS interface additional programs can be added by means of the [FILTER] section. These filters enable considering the file opened by the controller according to its needs. The filter can do tasks such as ensuring that the G-code file ends with a M2 or M30 command, or detecting if the input is a depth image, and generate the G-code to mill the shape defined by the image [51].

The method to implement a GUI that allows visualising the STEP-NC file is presented in Álvares *et al.* [36].

In order to add a new interface in AXIS, the Glade Interface Designer was used. Glade is an open source tool for visual GUI development, being independent of any programming language and does not generate source code, except an XML file. Using the widgets available in Glade, a GUI was created to visualise the STEP-NC file. Python was used, which works with Glade.

The file that belongs to the GUI of LinuxCNC can be recognised as ,axis.tcl,, and it is found in the controller,s folder. Tcl is mainly used for rapid development of software applications and GUIs.

In order to add the GUI to visualise the STEP-NC file in the AXIS GUI, the ,view, menu is chosen in order to add a new function. This function seeks to run the program in Python to visualise the STEP-NC file in that GUI. This is done by modifying the axis.tcl file. Figure 8 presents two windows: the ASEA virtual robot is shown on the right-hand side, whereas this menu is shown on the left-hand side containing the command that runs the program to display the STEP-NC file.

The STEP-NC controller is integrated with the STEP part 21 (ISO 10303 standard) data model using an adapter STEP-NC2GCode that reads a STEP-NC file generated by a software that models a 3D part via the internet using STEP Modeler [36]. The ASEA IRB6 LinuxCNC controller [20] runs the STEP-NC file transparently.

The implementation of the STEP-NC-compliant CNC is carried out in two stages: (a) implementation of the STEP-NC2GCode adapter; (b) adding the adapter to the LinuxCNC controller.

The architecture and the implementation of the adapter are presented in Alvares *et al.* [20], which shows the extraction of information from the STEP-NC file and the stages to generate the G-code from the extracted information.

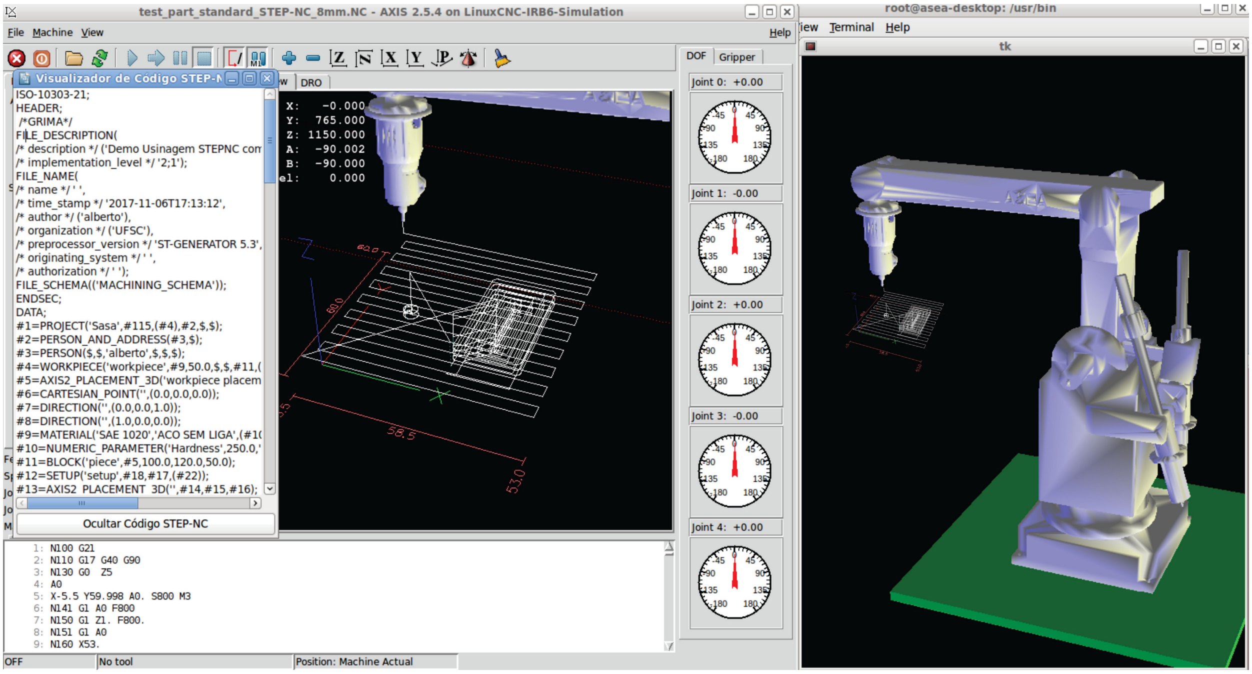
The forward and inverse kinematics developed by the authors for the ASEA IRB6 industrial robot in the LinuxCNC controller is available in https://tinyurl.com/rqfmbbu. The STEP-NC2GCode adapter integrated with the ASEA IRB6 robot LinuxCNC controller developed by the authors is available in https://tinyurl.com/y592fk5y.

Because LinuxCNC is an open source and highly-customizable application for CNC, it can be used as a starting point in the development of an open architecture controller fully compatible with STEP-NC. This objective can be achieved by adding the necessary functionalities so that LinuxCNC can directly interpret STEP-NC AP238 machining programs.

C. STEP MODELER: CAD/CAPP/CAM/NC

The STEP Modeler software [42] is an integrated CAD/CAPP/CAM system that is used in this work. It generates the STEP-NC file (ISO 14649 p21 or p28 formats) for machining a part, which is input to the CNC controller. Figure 9 shows the application activities that comprise the STEP Modeler.

By means of the CAD module of the STEP Modeler system, the user can design feature-based parts, which may contain general surfaces (e.g. Bèzièr). The CAPP module generates the process plan, which is available as output in



```
ISO-10303-21;
HEADER;
 /*GRIMA*/
FILE_DESCRIPTION(
/* description */ ('Demo Usinagem STEPNC com
/* implementation_level */ '2;1');
FILE_NAME(
/* name */ ' ',
/* time_stamp */ '2017-11-06T17:13:12',
/* author */ ('alberto'),
/* organization */ ('UFSC'),
/* preprocessor_version */ 'ST-GENERATOR 5.3',
/* originating_system */ ' ',
/* authorization */ ' ');
FILE_SCHEMA(('MACHINING_SCHEMA'));
ENDSEC;
DATA;
#1=PROJECT('Sasa',#115,(#4),#2,$,$);
#2=PERSON_AND_ADDRESS(#3,$);
#3=PERSON($,$,'alberto',$,$,$);
#4=WORKPIECE('workpiece',#9,50.0,$,$,#11,(
#5=AXIS2_PLACEMENT_3D('workpiece placem
#6=CARTESIAN_POINT('',(0.0,0.0,0.0));
#7=DIRECTION('',(0.0,0.0,1.0));
#8=DIRECTION('',(1.0,0.0,0.0));
#9=MATERIAL('SAE 1020','ACO SEM LIGA',(#1(
#10=NUMERIC_PARAMETER('Hardness',250.0,'
#11=BLOCK('piece',#5,100.0,120.0,50.0);
#12=SETUP('setup',#18,#17,(#22));
#13=AXIS2_PLACEMENT_3D('',#14,#15,#16);
```

```
X:    -0.000
Y:   765.000
Z:  1150.000
A:   -90.002
B:   -90.000
el:    0.000
```

```
N100 G21
N110 G17 G40 G90
N130 G0  Z5
A0
X-5.5 Y59.998 A0. S800 M3
N141 G1 A0 F800
N150 G1 Z1. F800.
N151 G1 A0
N160 X53.
```

OFF | No tool | Position: Machine Actual

FIGURE 8. Tool paths generated by the adapter STEP-NC2GCode visualised in the AXIS GUI of the LinuxCNC controller (left), and the GUI for visualising the ASEA virtual robot, LinuxCNC controller (right).

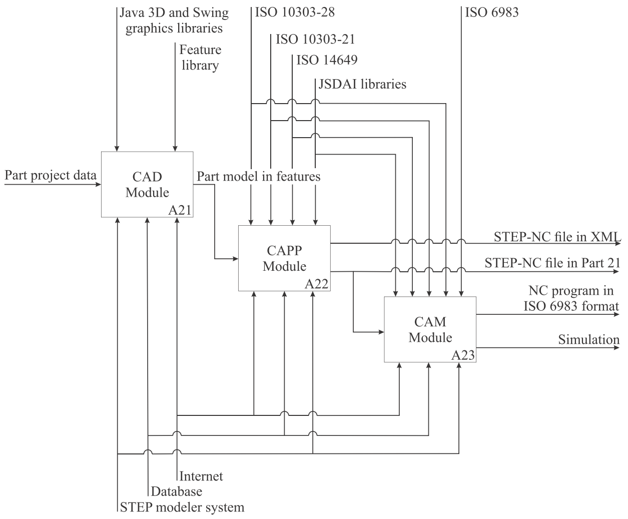


FIGURE 9. Functional model of the STEP Modeler software.

a STEP-NC file. Finally, the CAM module is responsible for converting the STEP-NC file (ISO 14649) to G-code (ISO 6983), in order to perform the machining operations, which can be carried out with different cutting tools. Further details about the STEP Modeler system are given in [36], [42].

D. PTC CREO OR SIEMENS NX CAD/CAM WITH CLFile AND STEP-NC MACHINE

In this architecture, Siemens NX and PTC Creo CAD/CAM software [57] are used to perform CAM modeling and analysis of the part to be manufactured by means of robotic machining. Both pieces of software allow exporting the manufacturing data in a CL file containing Automatically Programmed Tool (APT) language instructions, which is a high-level computer programming language used sometimes to generate instructions for CNC machine tools [58]. The APT CL file generated by both platforms can be directly imported by the STEP-NC Machine software, which converts the APT data to a STEP-NC program, which can then be exported as AP238 in part 21 file format. The approach to initiate this architecture has been proposed by Toquica *et al.* [17].

Figure 10 shows the data flow between application activities that comprise this architecture. It starts with activity A11, where modeling of the machining operations to be applied to manufacture a part is carried out using PTC Creo software [57].

CAD 3D part is modeled within PTC Creo by using its solid project module. Once the 3D model is created, it can be directly transferred as a reference model to the manufacturing module included in the PTC Creo suite, where different types of machining operations can be created for producing the geometric features of the part.

Facing and roughing operations were defined for a first example part as shown in Fig. 11. Both operations have been created by choosing the milling mode of the Creo manufacturing module, and then selecting the facing and roughing operations.

For each operation, one or more reference geometries can be defined from 3D part model. In the example part, the

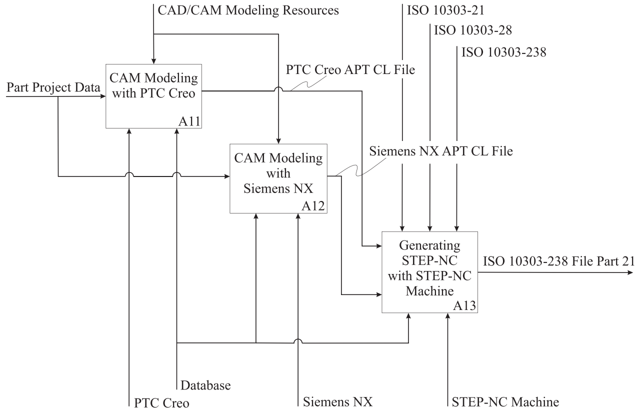


FIGURE 10. Data flow between activities involved in the generation of STEP-NC data by using Siemens NX, PTC Creo and STEP-NC Machine.

two internal geometries are related to a single roughing operation, while the facing operation is applied to the upper rectangular face of the part (Fig. 11). The reference geometries are considered as material to be removed from the workpiece, which is defined as a bounding box that encompasses the overall rectangular geometry of the part. Cutting tool configuration and process parameters such as feed rate, spindle speed and depth of cut are defined for each operation when it is created.

Tool path verification is carried out by means of virtual simulation of the machining operations as a previous step to exporting the NC code as an APT CL file. PTC Creo allows simulating: (a) only tool trajectory, (b) three-dimensional material removal, or (c) virtual machining including the machine tool 3D kinematic model. Thus, it enables incorporating a virtual 3D model of the robot ensuring all the relationships and movement constraints among its kinematic components in order to carry out robot machining simulation within PTC Creo.

Machining data are then exported to an independent APT CL file that will eventually be post-processed into the STEP-NC Machine software.

In activity A12 of Fig. 10, Siemens NX is also proposed for CAM modeling of the part, which provides a set of capabilities to facilitate part modeling, tool design, NC programming, inspection, and simulation, in a single platform. The section corresponding to Siemens NX in Fig. 11 illustrates some of the steps taken to generate machining data for the same example part. Firstly, a 3D model in STEP file format is opened in the manufacturing module of Siemens NX. The STEP 3D model was generated using other CAD software. Then, the workpiece size is defined according to the part dimensions, and a part reference zero is assigned to one of the workpiece corners (Fig. 11).

The operations are defined similarly to PTC Creo, where one operation can be applied to one or more reference geometries. Figure 11 shows the two operations being applied to the example part, listed in the operation panel on the workpiece view. Cutting tool type, feeds and speeds, depth of cut, and cutting strategy are some of the settings associated with each operation. Then, NX generates automatically the tool paths, which can immediately be verified with simulation.

Simulation with the 3D robot model can also be carried out using Siemens NX. For this, a 3D model of the ASEA robot has been built as a machine tool by using the machine tool builder module available in NX. Basically, it consists of assembling the elements that compose the robot, taking into account the logical order in the kinematic joints that connect the elements. The setup conditions for part, blank, tool and fixture can also be configured using the kinematic 3D model. Finally, the built robot kinematic model is stored into NX native format in the machine library directory of Siemens NX to be eventually used in virtual simulation. This is very important to verify the machining program before execution or its post-processing, in addition to analyzing the setup conditions of both the part and the robot.

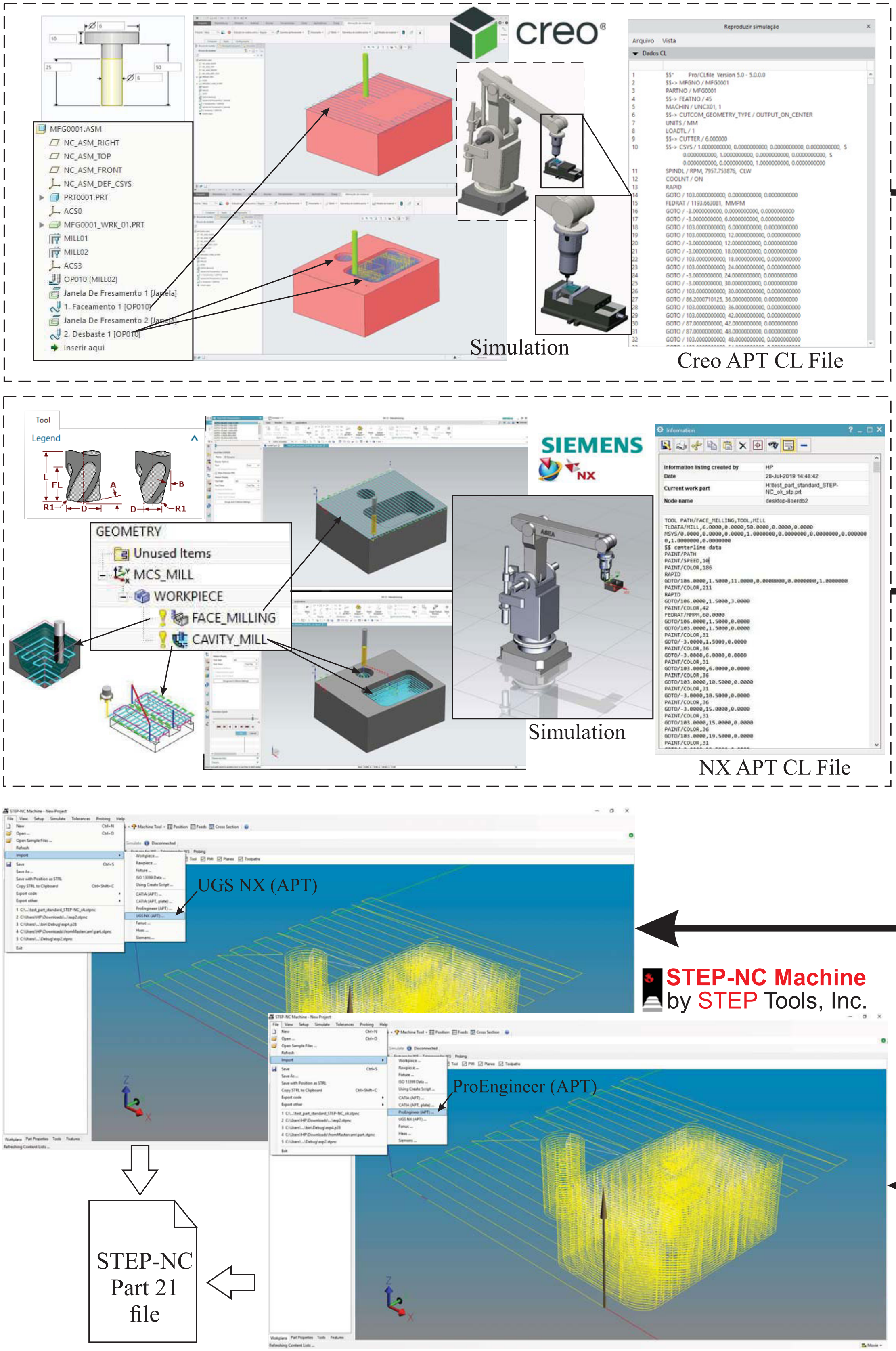
Siemens NX also allows exporting the machining program in an APT CL file format, which can be understood by the STEP-NC Machine software. Therefore, both APT CL files (generated by either PTC Creo or Siemens NX) are imported into STEP-NC Machine by means of the ''File'' tab, then ''Import'' and, finally, by choosing the option ''ProEngineer (APT)'' or ''UGS NX (APT)'' for Creo APT data and NX APT data, respectively.

STEP-NC Machine can interpret the APT data produced by Creo and NX to eventually generate a STEP-NC file in part 21 format according to ISO 10303 AP238. This architecture provides the advantage of using two recognized commercial CAD/CAM pieces of software to generate machine tool independent machining program files in the APT standard that can be directly imported into STEP-NC Machine for robotic machining simulation including the robot kinematic model. For the implementations carried out in this architecture, PTC Creo version 5, Siemens NX version 12 and STEP-NC Machine version 11 were used.

E. MASTERCAM CAD/CAM WITH AP238 PLUG-IN FROM STEP TOOLS

STEP Tools developed and made available a package that contains Visual Studio C++ project files to build a STEP-NC software plug-in for Mastercam [59]. Such a plug-in allows exporting a machining program as STEP-NC directly from Mastercam.

Using this plug-in to generate AP238 STEP-NC data for robot machining with Mastercam is another architecture addressed in this work. Data flow and activities involved in this architecture are outlined in Fig. 12. In activity A31, building the C++ project into the Visual Studio.NET framework creates a DLL (Dynamic Link Library) called ''ap238export.dll'' on Mastercam Chooks directory. Then, the DLL is used as control in activity A33, in which input machining data produced by Mastercam in activity A32 is used to generate the AP238 STEP-NC file in part 21 format.



```
$$*      Pro/CLfile  Version 5.0 - 5.0.0.0
$$-> MFGNO / MFG0001
PARTNO / MFG0001
$$-> FEATNO / 45
MACHIN / UNCX01, 1
$$-> CUTCOM_GEOMETRY_TYPE / OUTPUT_ON_CENTER
UNITS / MM
LOADTL / 1
$$-> CUTTER / 6.000000
$$-> CSYS / 1.0000000000, 0.0000000000, 0.0000000000, 0.0000000000,  $
     0.0000000000, 1.0000000000, 0.0000000000, 0.0000000000,  $
     0.0000000000, 0.0000000000, 1.0000000000, 0.0000000000
SPINDL / RPM, 7957.753876,  CLW
COOLNT / ON
RAPID
GOTO / 103.0000000000, 0.0000000000, 0.0000000000
FEDRAT / 1193.663081,  MMPM
GOTO / -3.0000000000, 0.0000000000, 0.0000000000
GOTO / -3.0000000000, 6.0000000000, 0.0000000000
GOTO / 103.0000000000, 6.0000000000, 0.0000000000
GOTO / 103.0000000000, 12.0000000000, 0.0000000000
GOTO / -3.0000000000, 12.0000000000, 0.0000000000
GOTO / -3.0000000000, 18.0000000000, 0.0000000000
GOTO / 103.0000000000, 18.0000000000, 0.0000000000
GOTO / 103.0000000000, 24.0000000000, 0.0000000000
GOTO / -3.0000000000, 24.0000000000, 0.0000000000
GOTO / -3.0000000000, 30.0000000000, 0.0000000000
GOTO / 103.0000000000, 30.0000000000, 0.0000000000
GOTO / 86.2000710125, 36.0000000000, 0.0000000000
GOTO / 103.0000000000, 36.0000000000, 0.0000000000
GOTO / 103.0000000000, 42.0000000000, 0.0000000000
GOTO / 87.0000000000, 42.0000000000, 0.0000000000
GOTO / 87.0000000000, 48.0000000000, 0.0000000000
GOTO / 103.0000000000, 48.0000000000, 0.0000000000
```

Information listing created by	HP
Date	28-Jul-2019 14:48:42
Current work part	H:\test_part_standard_STEP-NC_ok_stp.prt
Node name	desktop-8oerdb2

```
TOOL PATH/FACE_MILLING,TOOL,MILL
TLDATA/MILL,6.0000,0.0000,50.0000,0.0000,0.0000
MSYS/0.0000,0.0000,0.0000,1.0000000,0.0000000,0.0000000,0.000000
0,1.0000000,0.0000000
$$ centerline data
PAINT/PATH
PAINT/SPEED,10
PAINT/COLOR,186
RAPID
GOTO/106.0000,1.5000,11.0000,0.0000000,0.0000000,1.0000000
PAINT/COLOR,211
RAPID
GOTO/106.0000,1.5000,3.0000
PAINT/COLOR,42
FEDRAT/MMPM,60.0000
GOTO/106.0000,1.5000,0.0000
GOTO/103.0000,1.5000,0.0000
PAINT/COLOR,31
GOTO/-3.0000,1.5000,0.0000
PAINT/COLOR,36
GOTO/-3.0000,6.0000,0.0000
PAINT/COLOR,31
GOTO/103.0000,6.0000,0.0000
PAINT/COLOR,36
GOTO/103.0000,10.5000,0.0000
PAINT/COLOR,31
GOTO/-3.0000,10.5000,0.0000
PAINT/COLOR,36
GOTO/-3.0000,15.0000,0.0000
PAINT/COLOR,31
GOTO/103.0000,15.0000,0.0000
PAINT/COLOR,36
GOTO/103.0000,19.5000,0.0000
PAINT/COLOR,31
```

FIGURE 11. Generation of AP 238 STEP-NC data with STEP-NC Machine from APT CL file built in Siemens NX or PTC Creo.

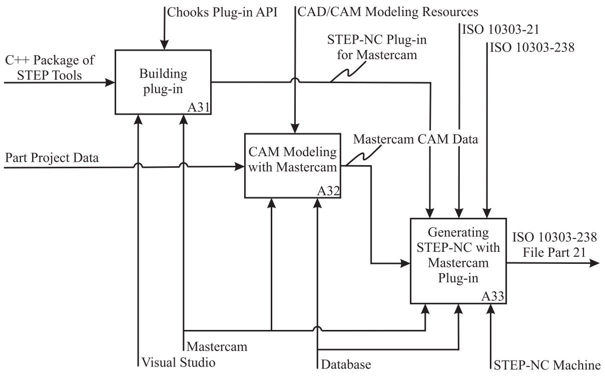
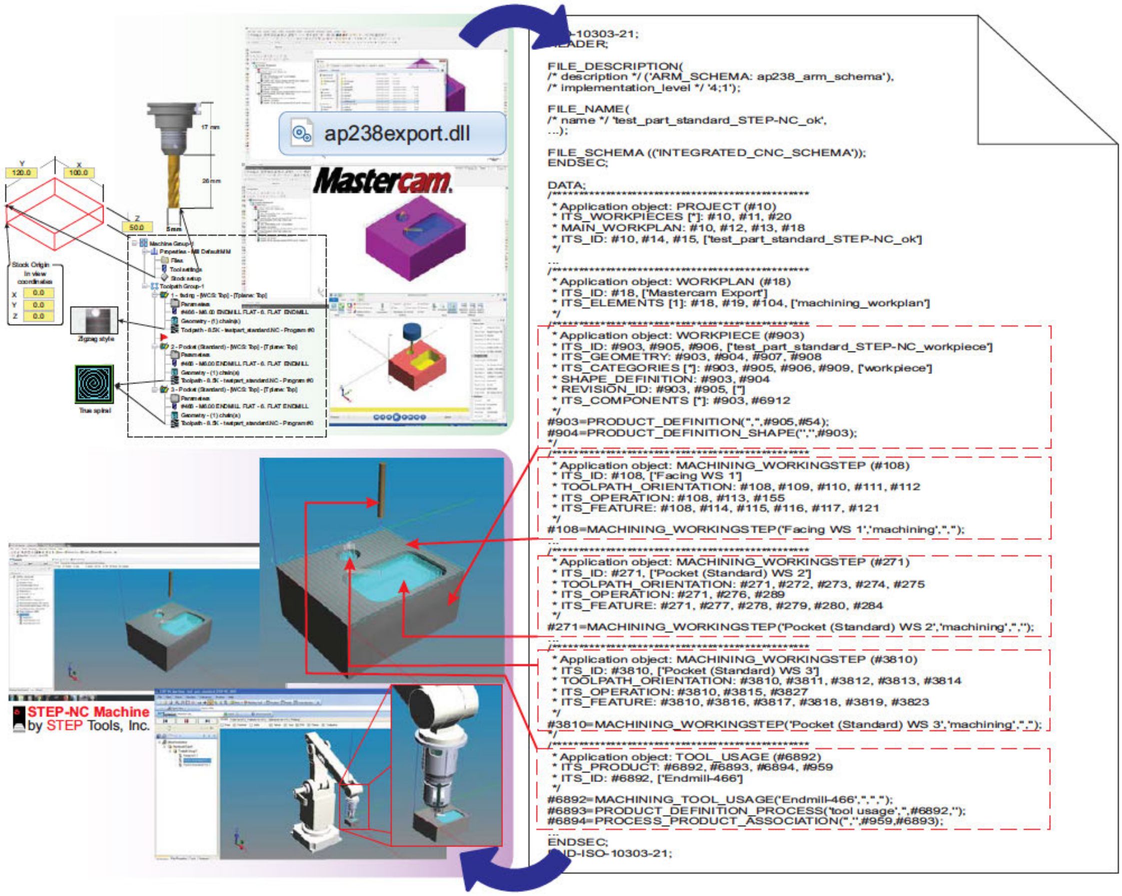

Chooks Plug-in API
CAD/CAM Modeling Resources
ISO 10303-21
ISO 10303-238
STEP-NC Plug-in for Mastercam
C++ Package of STEP Tools
Building plug-in
A31
Part Project Data
CAM Modeling with Mastercam
A32
Mastercam CAM Data
Generating STEP-NC with Mastercam Plug-in
A33
ISO 10303-238 File Part 21
Mastercam
Visual Studio
Database
STEP-NC Machine

FIGURE 12. Activities related to the generation of STEP-NC file in part 21 format using the Mastercam plug-in software.

The architecture for a test part is illustrated in Fig. 13. It starts with modeling in Mastercam the 3D model of the part to be manufactured. The machining operations, the machine, part setup, cutting tool and process parameters are defined using the Mastercam tools. For the test part three machining operations were defined: one facing and two pocketing operations, all of them performed using a flat bottom end mill tool with 6 mm diameter. Then, Mastercam automatically generates optimized tool paths for each of these machining operations.

After the machining operations are modeled, tool path verification can be performed in Mastercam by means of its simulation module. Then, STEP-NC data are exported thanks to the plug-in software. In order to use the plug-in, the user needs to click the "File" tab in Mastercam and, then, type Alt + C. This action opens the Windows file explorer located in the Chooks folder, from which the DLL "ap238export.dll" must be selected. The plug-in uses the DLL "stepnc.dll" of STEP-NC Machine to generate the STEP-NC file.

An excerpt of the STEP-NC file generated for the test part is shown on the right side of Fig. 13. It should be noticed that this file contains only tool path data that can be visualized when opened in the STEP-NC Machine environment, which can be appended with additional information including workpiece and cutting tool shape 3D-model data. Furthermore, simulation of the resulting STEP-NC file is carried out by incorporating the ASEA IRB6-S2 robot kinematic three-dimensional model into STEP-NC Machine. The robot machining simulation is described in more detail in section III-A.



```
-10303-21;
ADER;

FILE_DESCRIPTION(
/* description */ ('ARM_SCHEMA: ap238_arm_schema'),
/* implementation_level */ '4;1');

FILE_NAME(
/* name */ 'test_part_standard_STEP-NC_ok',
...);

FILE_SCHEMA (('INTEGRATED_CNC_SCHEMA'));
ENDSEC;

DATA;
/*********************************************
 * Application object: PROJECT (#10)
 * ITS_WORKPIECES [*]: #10, #11, #20
 * MAIN_WORKPLAN: #10, #12, #13, #18
 * ITS_ID: #10, #14, #15, ['test_part_standard_STEP-NC_ok']
 */
...
/*********************************************
 * Application object: WORKPLAN (#18)
 * ITS_ID: #18, ['Mastercam Export']
 * ITS_ELEMENTS [1]: #18, #19, #104, ['machining_workplan']
 */
/*********************************************
 * Application object: WORKPIECE (#903)
 * ITS_ID: #903, #905, #906, ['test_part_standard_STEP-NC_workpiece']
 * ITS_GEOMETRY: #903, #904, #907, #908
 * ITS_CATEGORIES [*]: #903, #905, #906, #909, ['workpiece']
 * SHAPE_DEFINITION: #903, #904
 * REVISION_ID: #903, #905, ['']
 * ITS_COMPONENTS [*]: #903, #6912
 */
#903=PRODUCT_DEFINITION('','',#905,#54);
#904=PRODUCT_DEFINITION_SHAPE('','',#903);
*/
/*********************************************
 * Application object: MACHINING_WORKINGSTEP (#108)
 * ITS_ID: #108, ['Facing WS 1']
 * TOOLPATH_ORIENTATION: #108, #109, #110, #111, #112
 * ITS_OPERATION: #108, #113, #155
 * ITS_FEATURE: #108, #114, #115, #116, #117, #121
 */
#108=MACHINING_WORKINGSTEP('Facing WS 1','machining','','');
...
/*********************************************
 * Application object: MACHINING_WORKINGSTEP (#271)
 * ITS_ID: #271, ['Pocket (Standard) WS 2']
 * TOOLPATH_ORIENTATION: #271, #272, #273, #274, #275
 * ITS_OPERATION: #271, #276, #289
 * ITS_FEATURE: #271, #277, #278, #279, #280, #284
 */
#271=MACHINING_WORKINGSTEP('Pocket (Standard) WS 2','machining','','');
...
/*********************************************
 * Application object: MACHINING_WORKINGSTEP (#3810)
 * ITS_ID: #3810, ['Pocket (Standard) WS 3']
 * TOOLPATH_ORIENTATION: #3810, #3811, #3812, #3813, #3814
 * ITS_OPERATION: #3810, #3815, #3827
 * ITS_FEATURE: #3810, #3816, #3817, #3818, #3819, #3823
 */
#3810=MACHINING_WORKINGSTEP('Pocket (Standard) WS 3','machining','','');
*/
/*********************************************
 * Application object: TOOL_USAGE (#6892)
 * ITS_PRODUCT: #6892, #6893, #6894, #959
 * ITS_ID: #6892, ['Endmill-466']
 */
#6892=MACHINING_TOOL_USAGE('Endmill-466','','','');
#6893=PRODUCT_DEFINITION_PROCESS('tool usage','',#6892,'');
#6894=PROCESS_PRODUCT_ASSOCIATION('','',#959,#6893);
...
ENDSEC;
D-ISO-10303-21;
```

FIGURE 13. Architecture describing the use of Mastercam with STEP-NC plug-in for robot machining.

This architecture contributes to encouraging the adoption of STEP-NC by means of its implementation on commercial CAD/CAM software such as Mastercam. However, a limitation of this architecture is that the available version of the STEP Tools project that builds the plug-in is compatible only with the Mastercam X8 SDK version, making it difficult to use it with earlier or newer versions. In this work, Mastercam X8 version, STEP-NC Machine 11 DLL, and Visual Studio 2013 were used.

F. PTC CREO, MasterCam OR SIEMENS NX CAD/CAM WITH CLFile AND ADAPTER CLFile TO AP238

This architecture is proposed to generate a STEP-NC program from the manufacturing information modeled in Mastercam. The functional model of the architecture comprises the three sub-activities shown in Fig 14. The architecture translates Mastercam machining data contained in a *.nci file to APT CLF data and, then, from APT to STEP-NC.

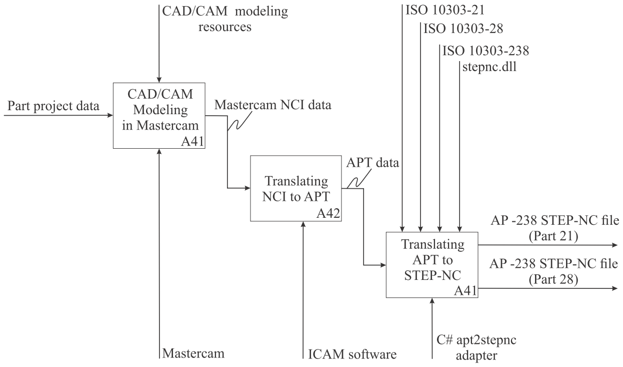


FIGURE 14. Deployment of activities for generating STEP-NC file from Mastercam machining data.

The workpiece model and machining operations in activity A41 are created using Mastercam. Besides, part setup, cutting tool, and process parameters are defined as per the available CNC machine characteristics. This information is condensed into a *.nci file type, which is primarily a Mastercam tool path file.

This file is input to activity A42, which converts it into APT data in a CL file by using the NCI2APT software from ICAM Technologies corporation. NCI2APT software is used because the current Mastercam versions do not directly generate an APT CL file. Subsequently, in activity A43, the developed adapter is used to parse the APT data to generate a STEP-NC file either in part 21 or Part 28 formats.

The APT2STEP-NC adapter was developed in C♯, which uses the DLL ''stepnc.dll'' from STEP-NC Machine to generate the STEP-NC data as ISO 10303 AP238. A snapshot of the adapter software is shown in Fig. 15. The APT data can be loaded into the adapter by means of the ''File'' tab, and then ''Import''. Several import options are provided to use APT data from PTC Creo, Siemens NX and Mastercam software.

Selection of the ''Mastercam APT'' option loads the APT CL file previously generated by means of the ICAM software, using the machining data built in Mastercam (NCI2APT converts Mastercam *.nci files into ISO 4343 Standard APT CL files). The APT data lines are displayed immediately in the text panel on the left-hand side of the APT2STEP-NC adapter software (Fig. 15). Then, the STEP-NC file is generated by clicking the button ''To STEP-NC'', and the STEP-NC machining data will be displayed in the text panel on the right-hand side of the adapter interface, as shown in Fig. 15.

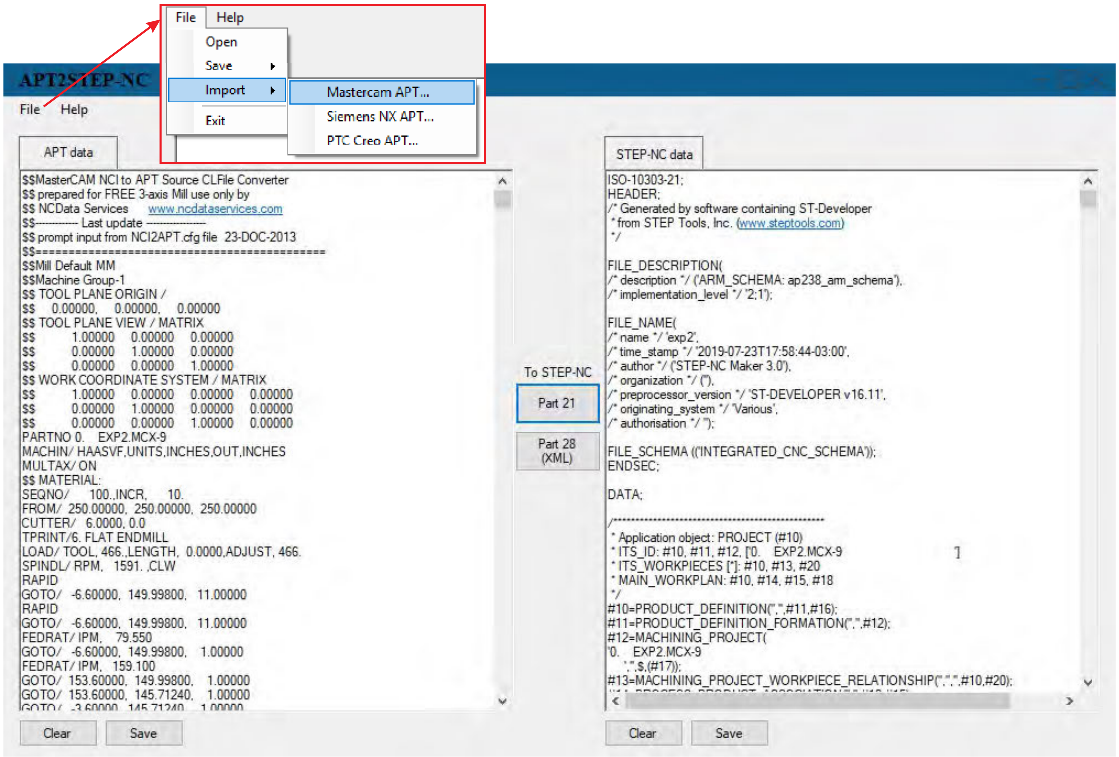
In the Mastercam APT CL file, machining information is grouped into sections called ''Machine Group'', in which several Machine Groups can coexist within the same file. The machining data associated with each machining operation belonging to the same Machine Group can be distinguished. The APT2STEP-NC software contains a class library from which instances are generated for each of the APT data entities in the APT CL file. Thus, instances of the *MachineGroup* class are created and stored in a list keeping the same sequence as in the APT file.

A machine group is composed of a list of instances of the *ToolpathGroup* class, which represents the machining operations associated with that machine group. Similarly, a tool path group comprises a list of instances of the *MachiningData* class, which is an abstract class that can take the form of APT instructions such as GOTO, RAPID, CIRCLE, FEDRAT, SPINDL, etc. represented by a C# class from the APT2STEP-NC APT library.

The STEP-NC machining program is generated by means of methods provided by the *STEPNCLib* class library from the STEP-NC Machine DLL. APT data entities are mapped into STEP-NC data entities. For example, machine groups are treated as nested Workplans within the main Workplan of a STEP-NC project, which can be created by using the *NestWorkplan* function.

Tool path groups are mapped as Workingsteps belonging to the current Workplan, being generated by means of the *Workingstep* function. On the other hand, machining data of a tool path group representing basic APT instructions such as GOTO, RAPID, CIRCLE, FEDRAT, SPINDL, CUTTER, LOADTOOL, COOLNT and MULTAX are directly mapped as its equivalent method from the *AptStepMaker* class: *GoToXYZ*, *Rapid*, *ArcXYPlane*, *Feedrate*, *SpindleSpeed*, *DefineTool*, *LoadTool*, *Coolant* and *MultaxOn/Off*, respectively. Then, the STEP-NC machining program can be generated either in part 21 or 28 format by using one of the *SaveAsP21* or *SaveFastAsModules* functions. A version of the source code of the APT2STEP-NC adapter software is available in the URL https://github.com/StepNcLadprer/apt2stepnc.

Siemens NX and PTC Creo work with ISO 4343 Standard and APT CL data structure, and can export CL data files directly without the need of complementary software, as in the case of MasterCam, and using the adapter to generate STEP AP238 files. Unlike the architecture presented in the previous section, this architecture has the advantage of using any version of software such as Mastercam.



```
$$MasterCAM NCI to APT Source CLFile Converter
$$ prepared for FREE 3-axis Mill use only by
$$ NCData Services     www.ncdataservices.com
$$----------- Last update ---------------
$$ prompt input from NCI2APT.cfg file  23-DOC-2013
$$============================================
$$Mill Default MM
$$Machine Group-1
$$ TOOL PLANE ORIGIN /
$$    0.00000,    0.00000,    0.00000
$$ TOOL PLANE VIEW / MATRIX
$$         1.00000    0.00000    0.00000
$$         0.00000    1.00000    0.00000
$$         0.00000    0.00000    1.00000
$$ WORK COORDINATE SYSTEM / MATRIX
$$         1.00000    0.00000    0.00000    0.00000
$$         0.00000    1.00000    0.00000    0.00000
$$         0.00000    0.00000    1.00000    0.00000
PARTNO 0.    EXP2.MCX-9
MACHIN/ HAASVF,UNITS,INCHES,OUT,INCHES
MULTAX/ ON
$$ MATERIAL:
SEQNO/     100.,INCR,     10.
FROM/  250.00000,  250.00000,  250.00000
CUTTER/   6.0000, 0.0
TPRINT/6. FLAT ENDMILL
LOAD/ TOOL, 466.,LENGTH,  0.0000,ADJUST, 466.
SPINDL/ RPM,   1591. ,CLW
RAPID
GOTO/   -6.60000,  149.99800,   11.00000
RAPID
GOTO/   -6.60000,  149.99800,   11.00000
FEDRAT/ IPM,    79.550
GOTO/   -6.60000,  149.99800,    1.00000
FEDRAT/ IPM,   159.100
GOTO/  153.60000,  149.99800,    1.00000
GOTO/  153.60000,  145.71240,    1.00000
GOTO/   -3.60000,  145.71240,    1.00000
```

```
ISO-10303-21;
HEADER;
/* Generated by software containing ST-Developer
 * from STEP Tools, Inc. (www.steptools.com)
 */

FILE_DESCRIPTION(
/* description */ ('ARM_SCHEMA: ap238_arm_schema'),
/* implementation_level */ '2;1');

FILE_NAME(
/* name */ 'exp2',
/* time_stamp */ '2019-07-23T17:58:44-03:00',
/* author */ ('STEP-NC Maker 3.0'),
/* organization */ (''),
/* preprocessor_version */ 'ST-DEVELOPER v16.11',
/* originating_system */ 'Various',
/* authorisation */ '');

FILE_SCHEMA (('INTEGRATED_CNC_SCHEMA'));
ENDSEC;

DATA;

/*************************************************
 * Application object: PROJECT (#10)
 * ITS_ID: #10, #11, #12, ['0.    EXP2.MCX-9                    ']
 * ITS_WORKPIECES [*]: #10, #13, #20
 * MAIN_WORKPLAN: #10, #14, #15, #18
 */
#10=PRODUCT_DEFINITION('','',#11,#16);
#11=PRODUCT_DEFINITION_FORMATION('','',#12);
#12=MACHINING_PROJECT(
'0.    EXP2.MCX-9
    ','',$,(#17));
#13=MACHINING_PROJECT_WORKPIECE_RELATIONSHIP('','','',#10,#20);
```

FIGURE 15. APT2STEP-NC adapter software.

TABLE 1. Comparison of presented STEP-NC-based architectures for robotic machining.

Architecture	CAD/CAM System	Tool Path	NC Program
1	Not applicable	Not applicable	RS-274 and AP238
2	Not applicable	Not applicable	ISO 14649 and RS-274
3	Creo PTC or Siemens NX	APT CLFile converted to STEP AP238	AP238 and RS-274
4	Mastercam	STEP-NC AP238	AP238 and RS-274
5	Creo PTC, Mastercam or Siemens NX	NCI File Mastercam, APT CLFile converted to STEP AP238	AP238 and RS-274
6	STEP Modeler CAD/CAPP/CAM	Not applicable	ISO 14649

Although STEP-NC Machine software has support to open some APT programs exported from PCT Creo or Siemens NX, these functionalities have also been added to the APT2STEP-NC adapter software. Unlike STEP-NC Machine, APT2STEP-NC software is an open-source software available for use by anyone. Moreover, older versions of the STEP-NC Machine software do not support some of the APT machining programs generated by newer versions of PTC Creo and Siemens NX, which are supported by APT2STEP-NC.

Finally, the STEP-NC AP238 file generated by the APT2STEP-NC adapter can be used in the STEP-NC Machine simulation scenario, allowing both simulating robotic machining and exporting the machining program in G-code, compatible with the ASEA IRB6 robot LinuxCNC controller for the real robotic machining.

IV. VALIDATION OF THE ARCHITECTURES BY MEANS OF CASE STUDIES

This section presents the validation and implementation results of the introduced architectures by means of case studies, including the simulation of robotic machining using LinuxCNC and ASEA robot as a real manufacturing scenario. The integration of LinuxCNC for robot control includes the Open Architecture Controllers (OAC) concept to the STEP-NC standard [17], [18], [32], contributing with Open STEP-NC solutions for industry and academics developments.

Aiming to highlight the contribution of this work related to the implementation and validation stages of the six presented architectures, the main characteristics of each one were centralized and the comparison between proposed parameters is detailed in the Table 1, summarizing the proposed correlation between the six introduced architectures. Second column (CAD/CAM System) describes the CAx compliant platform used in this work, while third and fourth columns relate to the standard file used for tool path programming in NC machining process.

Procedures for manufacturing four different example parts using the six architectures (Table 1) are presented below, including the validation on a real industrial robot. In this case, the LinuxCNC controller receives the four NC programs

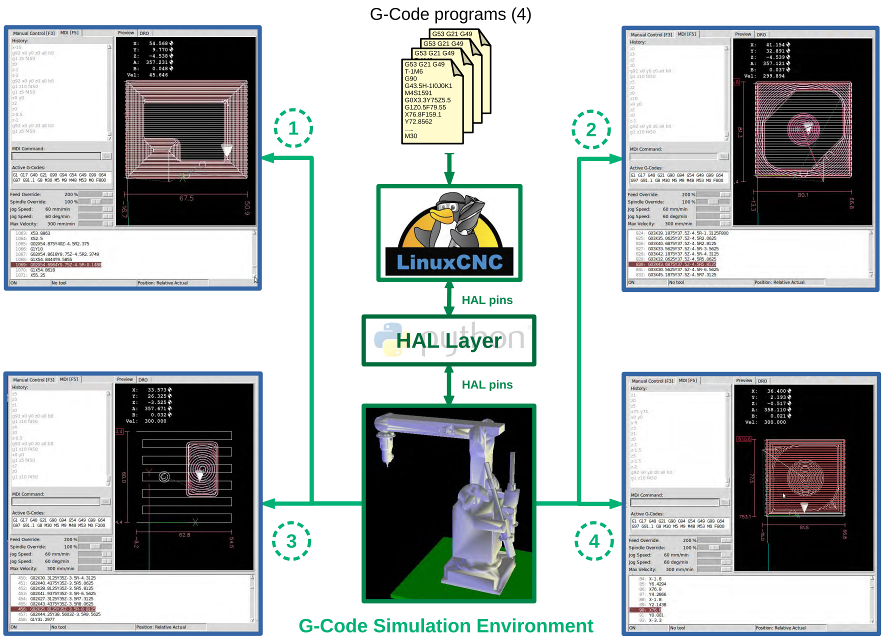


FIGURE 16. Machining simulation with LinuxCNC controller and virtual ASEA robot.

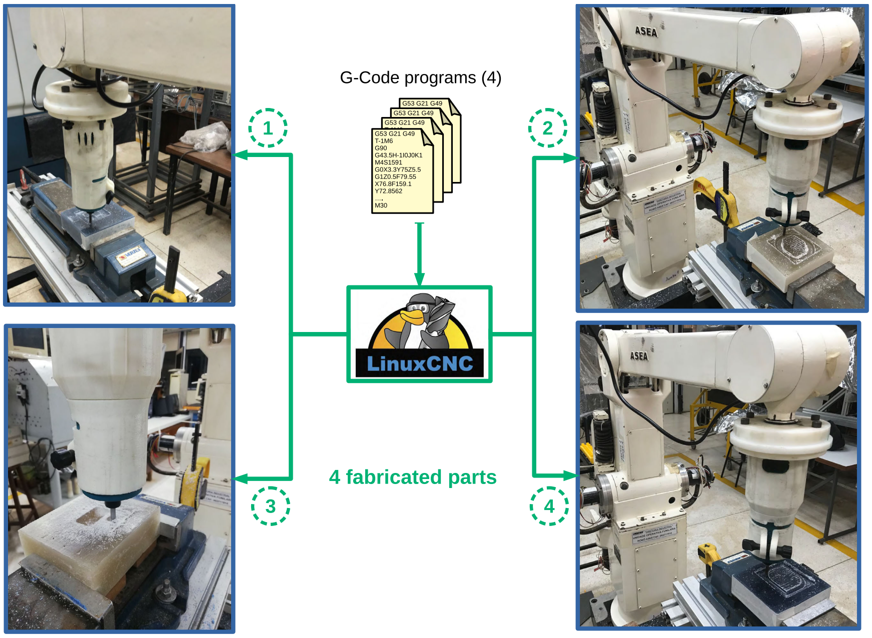


FIGURE 17. Robotic machining with LinuxCNC controller and ASEA robot.

(G-code) independently in order to generate the Asea robot movements.

Each NC program file was exported to the LinuxCNC simulator, ensuring the compatibility of the NC program generated by STEP-NC Machine with the RS274/NGC interpreter. In Fig. 16 the NC programs are simulated in the LinuxCNC controller simulator, associated with the four parts. HAL layer is an important middleware application developed in Python into LinuxCNC controller, responsible for simulating and processing input/outputs signals (HAL pins) in the machining process [18], [20].

After verifying the machining simulation, the four parts were effectively machined using the Asea robot. The end-effector was a Makita 3709 trimmer, whereas the material used to machine the parts was paraffin wax blocks, which reduces the effects of cutting forces on the robot structure [60] in comparison with harder materials.

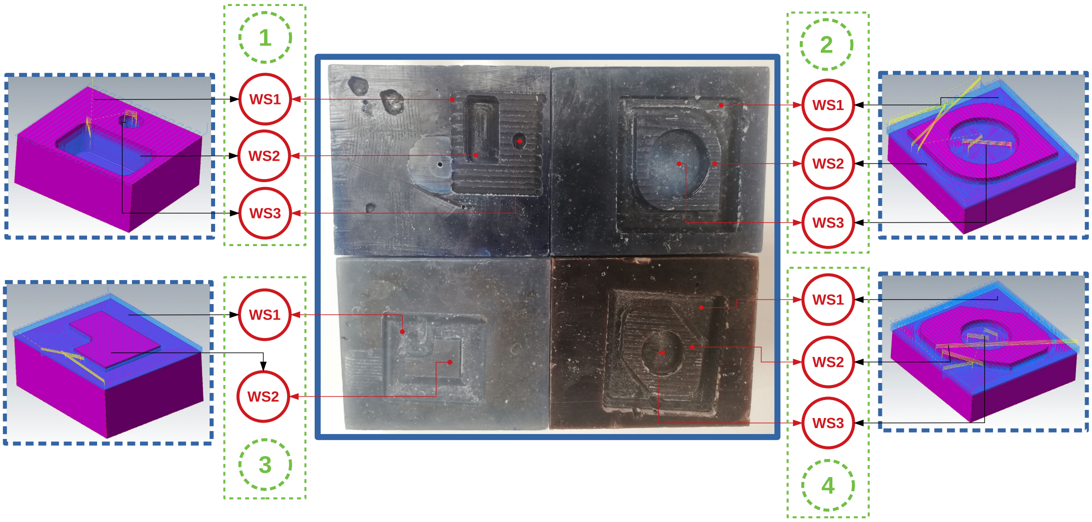


FIGURE 18. The four example parts and respective workingsteps machined using the ASEA robot and LinuxCNC controller.

Fig. 17 shows the machining of the four parts using the ASEA robot and LinuxCNC controller, and the Workingsteps in each machined part are shown in Fig. 18. It should be pointed out that the workpieces used in the experiments had the same dimensions, all of them being larger than the original CAD dimensions.

The the CAD/CAM systems generated circular tool strategies for machining the holes and pockets in the example parts, and this resulted in significant geometrical and dimensional errors in the parts machined with the ASEA robot. However, the objective was to validate the manufacture of parts using a real robot with a STEP-NC-compliant controller, and this validation was achieved.

The STEP-NC AP238, ISO 14649, and G-code (RS-274) files used in the validation of the proposed architectures are available from URL https://tinyurl.com/yyn49jow. Machining processing videos are available from https://tinyurl.com/wz4nyjd, https://tinyurl.com/rslfn6m, and https://tinyurl.com/wk7lv6p.

V. CONCLUSION

In order to contribute to the use of the STEP-NC standard in industrial robotic machining, some STEP-NC-compliant architectures were implemented and validated in this work. Each architecture is capable of generating a STEP-NC file, which is first simulated in a virtual environment and, then, the program is executed in a real physical ASEA IRB6-S2 robot with LinuxCNC controller compatible with STEP-NC and G-code. Different parts were modeled and machined.

The first proposed architecture allows generating virtual models of any robot to be simulated in the STEP-NC Machine software by reading assembly model data from a STEP (AP-203, AP-214, AP-242) file. This tool was used to generate the virtual model of the ASEA robot.

A new plug-in software for LinuxCNC was also developed, which enables reading a ISO 14649 STEP-NC file directly in the LinuxCNC environment and, then, translate it to G-code, which is read by the ASEA robot controller.

The STEP modeler software, presented in the third architecture, enables integrating CAD, CAPP, CAM, and CNC. It exports ISO 14649 STEP-NC files either as P21 or P28 formats.

The fourth architecture uses PTC Creo and Siemens NX software to generate a machine tool-independent machining program in APT CL file that can be directly imported and processed in the STEP-NC Machine software to create the STEP-NC data used for robotic machining. This architecture enables changing rapidly from native manufacturing data produced in the CAD/CAM platform to neutral STEP-NC data generated by STEP-NC Machine having APT data as intermediate.

The number of steps required to generate a STEP-NC file in the fifth architecture is significantly reduced because the manufacturing data modeled in Mastercam are written directly as AP238 STEP-NC in part 21 format. Thus, most Mastercam users are closer to implementing STEP-NC on their machine tools and industrial robots. However, in this architecture the plug-in functionality is compatible only with the Mastercam SDK version 8.

On the other hand, the sixth architecture provides the advantage of being implemented for any Mastercam version, which expands the possibilities for STEP-NC applications in industrial robotic machining with currently available commercial software.

By means of the application protocols of the STEP-NC standard, solutions and implementations can be generated based on a neutral, standardized, and open file that allows new functionalities to be included within the digital manufacturing chain. In the simulation of the architectures using STEP-NC Machine, it was necessary to have access to information about the kinematic chain of the ASEA robot, which was extracted from the STEP file containing the robot model. This article presents the method used to read, write, and modify STEP files, being a useful solution for developing future manufacturing functionalities.

A limitation to all presented architectures is the current lack of industrial robot controllers capable of interpreting a STEP-NC machining program directly. This will require significant effort in the future to develop open architecture controllers capable of processing STEP-NC programs effectively.

Future research will be focused on building an industrial robot controller based on the OAC concept, which could download from the Internet and execute programs in the STEP-NC AP238 format without third-party applications.

ACKNOWLEDGMENT

The authors would like to thank the University of Brasilia, MEC, CNPq, and FAPDF for supporting this research.

REFERENCES

[1] O. Ayari, A. Bouali, and P.-J. Méausoone, "Cutting forces and accuracy characterization during wood machining with serial robots," *Eur. J. Wood Wood Products*, vol. 78, no. 4, pp. 767–775, May 2020, doi: 10.1007/s00107-020-01539-4.

[2] Y. Chen and F. Dong, "Robot machining: Recent development and future research issues," *Int. J. Adv. Manuf. Technol.*, vol. 66, nos. 9–12, pp. 1489–1497, Jun. 2013, doi: 10.1007/s00170-012-4433-4.

[3] W. Ji and L. Wang, "Industrial robotic machining: A review," *Int. J. Adv. Manuf. Technol.*, vol. 103, nos. 1–4, pp. 1239–1255, Jul. 2019, doi: 10.1007/s00170-019-03403-z.

[4] I. Iglesias, J. E. Ares, C. Gonzalez-Gaya, F. Morales, and V. F. Rosales, "Predictive methodology for dimensional path precision in robotic machining operations," *IEEE Access*, vol. 6, pp. 49217–49223, 2018, doi: 10.1109/ACCESS.2018.2868549.

[5] P. Zheng, H. Wang, Z. Sang, R. Y. Zhong, Y. Liu, C. Liu, K. Mubarok, S. Yu, and X. Xu, "Smart manufacturing systems for industry 4.0: Conceptual framework, scenarios, and future perspectives," *Frontiers Mech. Eng.*, vol. 13, no. 2, pp. 137–150, Jun. 2018, doi: 10.1007/s11465-018-0499-5.

[6] *Industrial Automation Systems and Integration-Physical Device Control-Data Model for Computerized Numerical Controllers—Part 1: Overview and Fundamental Principles*, Standard ISO 14649-1, 2003.

[7] *Industrial Automation Systems and Integration-Product Data Representation and Exchange—Part 238: Application Protocol: Application Interpreted Model for Computerized Numerical Controllers*, Standard ISO 10303-238, 2007.

[8] *Numerical Control of Machines Program Format and Definition of Address Words Part1: Data Format for Positioning, Line Motion and Contouring Control Systems*, Standard ISO 6983-1, 1982.

[9] C. Danjou, J. Le Duigou, and B. Eynard, "Closed-loop manufacturing process based on STEP-NC," *Int. J. Interact. Design Manuf.*, vol. 11, no. 2, pp. 233–245, May 2017, doi: 10.1007/s12008-015-0268-1.

[10] M. Hardwick, Y. F. Zhao, F. M. Proctor, A. Nassehi, X. Xu, S. Venkatesh, D. Odendahl, L. Xu, M. Hedlind, M. Lundgren, L. Maggiano, D. Loffredo, J. Fritz, B. Olsson, J. Garrido, and A. Brail, "A roadmap for STEP-NC-enabled interoperable manufacturing," *Int. J. Adv. Manuf. Technol.*, vol. 68, nos. 5–8, pp. 1023–1037, Sep. 2013, doi: 10.1007/s00170-013-4894-0.

[11] *TC184/SC4/WG15 I. Working Draft for AP238 Edition 2*. Accessed: Mar. 9, 2020. [Online]. Available: https://githubcom/stepmfg/ap238e2

[12] B. Solvang, L. K. Refsahl, and G. Sziebig, "STEP-NC based industrial robot CAM system," *IFAC Proc. Vols.*, vol. 42, no. 16, pp. 245–250, 2009, doi: 10.3182/20090909-4-JP-2010.00043.

[13] L. J. Fang, L. Li, and G. X. Wang, "Integration of cutting robot with CAD/CAM system based on STEP-NC," *Appl. Mech. Mater.*, vol. 868, pp. 93–98, Jul. 2017, doi: 10.4028/www.scientific.net/AMM.868.93.

[14] S. N. Rea Minango and J. C. E. Ferreira, "Combining the STEP-NC standard and forward and inverse kinematics methods for generating manufacturing tool paths for serial and hybrid robots," *Int. J. Comput. Integr. Manuf.*, vol. 30, no. 11, pp. 1203–1223, Nov. 2017, doi: 10.1080/0951192X.2017.1305507.

[15] S. Zivanovic, N. Slavkovic, and D. Milutinovic, "An approach for applying STEP-NC in robot machining," *Robot. Comput.-Integr. Manuf.*, vol. 49, pp. 361–373, Feb. 2018, doi: 10.1016/j.rcim.2017.08.009.

[16] N. Slavkovic, S. Zivanovic, and D. Milutinovic, "An indirect method of industrial robot programming for machining tasks based on STEP-NC," *Int. J. Comput. Integr. Manuf.*, vol. 32, no. 1, pp. 43–57, Jan. 2019, doi: 10.1080/0951192X.2018.1543952.

[17] J. S. Toquica, S. Živanović, A. J. Alvares, and R. Bonnard, "A STEP-NC compliant robotic machining platform for advanced manufacturing," *Int. J. Adv. Manuf. Technol.*, vol. 95, nos. 9–12, pp. 3839–3854, Apr. 2018, doi: 10.1007/s00170-017-1466-8.

[18] J. S. Toquica, S. Zivanovic, R. Bonnard, E. Rodriguez, A. J. Alvares, and J. C. E. Ferreira, "STEP-NC-based machining architecture applied to industrial robots," *J. Brazilian Soc. Mech. Sci. Eng.*, vol. 41, no. 8, p. 314, Aug. 2019, doi: 10.1007/s40430-019-1811-y.

[19] W. Xiao, J. Huan, and S. Dong, "A STEP-compliant industrial robot data model for robot off-line programming systems," *Robot. Comput.-Integr. Manuf.*, vol. 30, no. 2, pp. 114–123, Apr. 2014, doi: 10.1016/j.rcim.2013.09.007.

[20] A. J. Alvares, J. S. Toquica, E. J. Lima, and M. H. S. Bomfim, "Retrofitting of the IRB6-S2 robotic manipulator using computer numerical control-based controllers," *J. Brazilian Soc. Mech. Sci. Eng.*, vol. 40, no. 3, p. 149, Mar. 2018, doi: 10.1007/s40430-018-1073-0.

[21] J. S. Toquica, D. Benavides, and J. M. S. T. Motta, "Web compliant open architecture for teleoperation of industrial robots," in *Proc. IEEE 15th Int. Conf. Autom. Sci. Eng. (CASE)*, Aug. 2019, pp. 1408–1414, doi: 10.1109/COASE.2019.8842847.

[22] M. Weck, J. Wolf, and D. Kiritsis, "STEP-NC-the STEP compliant NC programming interface," *Int. Intell. Manuf. Syst. Forum*, vol. 1, no. 1, pp. 1–4, 2001.

[23] M. Rauch, R. Laguionie, J.-Y. Hascoet, and S.-H. Suh, "An advanced STEP-NC controller for intelligent machining processes," *Robot. Comput.-Integr. Manuf.*, vol. 28, no. 3, pp. 375–384, Jun. 2012, doi: 10.1016/j.rcim.2011.11.001.

[24] S. Newman, R. Allen, and R. Rosso, "CAD/CAM solutions for STEP-compliant CNC manufacture," *Int. J. Comput. Integr. Manuf.*, vol. 16, nos. 7–8, pp. 590–597, Jan. 2003, doi: 10.1080/0951192031000115688.

[25] R. S. U. Rosso, S. T. Newman, and S. Rahimifard, "The adoption of STEP-NC for the manufacture of asymmetric rotational components," *Proc. Inst. Mech. Eng., B, J. Eng. Manuf.*, vol. 218, no. 11, pp. 1639–1644, Nov. 2004, doi: 10.1243/0954405042418554.

[26] K. H. Ho, S. T. Newman, and R. D. Allen, "STEP-NC compliant information modelling for wire electrical discharge machining component manufacture," *Proc. Inst. Mech. Eng., B, J. Eng. Manuf.*, vol. 219, no. 10, pp. 777–784, Oct. 2005, doi: 10.1243/095440505X32751.

[27] L. Ali, S. T. Newman, and J. Petzing, "Development of a STEP-compliant inspection framework for discrete components," *Proc. Inst. Mech. Eng., B, J. Eng. Manuf.*, vol. 219, no. 7, pp. 557–563, Jul. 2005, doi: 10.1243/095440505X32364.

[28] S. H. Suh, B. E. Lee, D. H. Chung, and S. U. Cheon, "Architecture and implementation of a shop-floor programming system for STEP-compliant CNC," *Comput. Des.*, vol. 35, no. 12, pp. 1069–1083, Oct. 2003, doi: 10.1016/S0010-4485(02)00179-3.

[29] S.-H. Suh, D.-H. Chung, B.-E. Lee, S. Shin, I. Choi, and K.-M. Kim, "STEP-compliant CNC system for turning: Data model, architecture, and implementation," *Comput.-Aided Design*, vol. 38, no. 6, pp. 677–688, Jun. 2006, doi: 10.1016/j.cad.2006.02.006.

[30] X. W. Xu, "Realization of STEP-NC enabled machining," *Robot. Comput.-Integr. Manuf.*, vol. 22, no. 2, pp. 144–153, Apr. 2006, doi: 10.1016/j.rcim.2005.02.009.

[31] R. Laguionie, M. Rauch, J.-Y. Hascoët, and S.-H. Suh, "An eXtended manufacturing integrated system for feature-based manufacturing with STEP-NC," *Int. J. Comput. Integr. Manuf.*, vol. 24, no. 9, pp. 785–799, Sep. 2011, doi: 10.1080/0951192X.2011.592992.

[32] M. Rauch, J.-Y. Hascoët, V. Simoes, and K. Hamilton, "Advanced programming of machine tools: Interests of an open CNC controller within a STEP-NC environment," *Int. J. Machining Machinability Mater.*, vol. 15, nos. 1–2, p. 2, 2014, doi: 10.1504/IJMMM.2014.059184.

[33] O. F. Valilai and M. Houshmand, "INFELT STEP: An integrated and interoperable platform for collaborative CAD/CAPP/CAM/CNC machining systems based on STEP standard," *Int. J. Comput. Integr. Manuf.*, vol. 23, no. 12, pp. 1095–1117, Dec. 2010, doi: 10.1080/0951192X.2010.527373.

[34] W. Xiao, L. Zheng, J. Huan, and P. Lei, "A complete CAD/CAM/CNC solution for STEP-compliant manufacturing," *Robot. Comput.-Integr. Manuf.*, vol. 31, pp. 1–10, Feb. 2015, doi: 10.1016/j.rcim.2014.06.003.

[35] A. Bensoussan and J. Bensoussan, ''Comparative handbook: Robotic technologies law,'' in *Lexing-Technologies Avancées & Droit*. Rennes, France: Éditions Larcier, 2016.

[36] A. J. Álvares, M. E. G. Paredes, J. C. E. Ferreira, and J. C. T. Benavente, ''A Web-based STEP-NC-compliant architecture for low cost 3D part manufacturing,'' *Int. J. Manuf. Res.*, vol. 11, no. 1, p. 1, 2016, doi: 10.1504/IJMR.2016.076975.

[37] S. Zivanovic and R. Puzovic, ''Wire EDM machining simulations based on STEP-NC program,'' *Tech. Gazette*, vol. 23, no. 6, pp. 1831–1838, Dec. 2016, doi: 10.17559/TV-20151122180547.

[38] S. Živanović and G. Vasilić, ''A new CNC programming method using STEP-NC protocol,'' *FME Trans.*, vol. 45, no. 1, pp. 149–158, 2017, doi: 10.5937/fmet1701149Z.

[39] *ST-Machine STEP-NC for CAM-CNC*, STEPTools, Troy, NY, USA, 2017.

[40] D. Srivastava and V. R. Komma, ''Development of STEP-NC for intelligent machining,'' in *Proc. Int. Conf. Adv. Mech., Ind., Autom. Manage. Syst. (AMIAMS)*, Feb. 2017, pp. 296–301, doi: 10.1109/AMIAMS.2017.8069228.

[41] C. Pan, S. S. Smith, and G. C. Smith, ''Automatic assembly sequence planning from STEP CAD files,'' *Int. J. Comput. Integr. Manuf.*, vol. 19, no. 8, pp. 775–783, Dec. 2006, doi: 10.1080/09511920500399425.

[42] J. C. E. Ferreira, J. C. T. Benavente, and P. H. S. Inoue, ''A Web-based CAD/CAPP/CAM system compliant with the STEP-NC standard to manufacture parts with general surfaces,'' *J. Brazilian Soc. Mech. Sci. Eng.*, vol. 39, no. 1, pp. 155–176, Jan. 2017, doi: 10.1007/s40430-016-0528-4.

[43] Z. Faraz, S. Waheed ul Haq, L. Ali, K. Mahmood, W. A. Tarar, A. A. Baqai, M. Khan, S. H. I. Jaffery, and R. S. Choudhry, ''Development of a STEP-compliant design and manufacturing framework for discrete sheet metal bend parts,'' *Proc. Inst. Mech. Eng., B, J. Eng. Manuf.*, vol. 232, no. 6, pp. 1090–1104, May 2018, doi: 10.1177/0954405416661007.

[44] *Robots and Robotic Devices–Vocabulary*, Standard ISO 8373-2 2012.

[45] Robotics IF. (2016). *World Robotics Report 2016*. [Online]. Available: https://tinyurl.com/y7y25obl

[46] Robotics IF. (2015). *World Robotics Survey: Industrial Robots are Conquering the World*. [Online]. Available: https://tinyurl.com/u8azuex

[47] X. Xu, ''Machine tool 4.0 for the new era of manufacturing,'' *Int. J. Adv. Manuf. Technol.*, vol. 92, nos. 5–8, pp. 1893–1900, Sep. 2017, doi: 10.1007/s00170-017-0300-7.

[48] S. H. Kim, E. Nam, T. I. Ha, S.-H. Hwang, J. H. Lee, S.-H. Park, and B.-K. Min, ''Robotic machining: A review of recent progress,'' *Int. J. Precis. Eng. Manuf.*, vol. 20, no. 9, pp. 1629–1642, Sep. 2019, doi: 10.1007/s12541-019-00187-w.

[49] *Industrial Automation Systems–Numerical Control of Machines–NC Processor Output–File Structure and Language Format*, Standard ISO 3592-2, 2000.

[50] F. M. Ribeiro, J. N. Pires, and A. S. Azar, ''Implementation of a robot control architecture for additive manufacturing applications,'' *Ind. Robot, Int. J. Robot. Res. Appl.*, vol. 46, no. 1, pp. 73–82, Jan. 2019, doi: 10.1108/IR-11-2018-0226.

[51] A. J. Alvares, J. S. Toquica, E. Jose Lima, and M. H. S. Bomfim, ''Retrofitting of ASEA IRB2-S6 industrial robot using numeric control technologies based on LinuxCNC and MACH3-MATLAB,'' in *Proc. IEEE Int. Conf. Robot. Biomimetics (ROBIO)*, Dec. 2017, pp. 2148–2153, doi: 10.1109/ROBIO.2017.8324737.

[52] U. Schneider, M. Drust, M. Ansaloni, C. Lehmann, M. Pellicciari, F. Leali, J. W. Gunnink, and A. Verl, ''Improving robotic machining accuracy through experimental error investigation and modular compensation,'' *Int. J. Adv. Manuf. Technol.*, vol. 85, nos. 1–4, pp. 3–15, Jul. 2016, doi: 10.1007/s00170-014-6021-2.

[53] Z. Pan, J. Polden, N. Larkin, S. Van Duin, and J. Norrish, ''Recent progress on programming methods for industrial robots,'' *Robot. Comput.-Integr. Manuf.*, vol. 28, no. 2, pp. 87–94, Apr. 2012, doi: 10.1016/j.rcim.2011.08.004.

[54] Y. Gan, X. Dai, and D. Li, ''Off-line programming techniques for multirobot cooperation system,'' *Int. J. Adv. Robotic Syst.*, vol. 10, no. 7, p. 282, Jul. 2013, doi: 10.5772/56506.

[55] P. Vichare, A. Nassehi, S. Kumar, and S. T. Newman, ''A unified manufacturing resource model for representing CNC machining systems,'' *Robot. Comput.-Integr. Manuf.*, vol. 25, no. 6, pp. 999–1007, Dec. 2009, doi: 10.1016/j.rcim.2009.04.014.

[56] W. Song, G. Wang, J. Xiao, G. Wang, and Y. Hong, ''Research on multirobot open architecture of an intelligent CNC system based on parameter-driven technology,'' *Robot. Comput.-Integr. Manuf.*, vol. 28, no. 3, pp. 326–333, Jun. 2012, doi: 10.1016/j.rcim.2011.10.002.

[57] Q. Qi, P. J. Scott, X. Jiang, and W. Lu, ''Design and implementation of an integrated surface texture information system for design, manufacture and measurement,'' *Comput.-Aided Design*, vol. 57, pp. 41–53, Dec. 2014, doi: 10.1016/j.cad.2014.06.013.

[58] O. I. Avram and P. Xirouchakis, ''Evaluating the use phase energy requirements of a machine tool system,'' *J. Cleaner Prod.*, vol. 19, nos. 6–7, pp. 699–711, Apr. 2011, doi: 10.1016/j.jclepro.2010.10.010.

[59] STEP Tools I. (2015). *Step-NC Export Chook for Mastercam*. [Online]. Available: https://githubcom/steptools/mastercam-stepnc

[60] L. Cen and S. N. Melkote, ''Effect of robot dynamics on the machining forces in robotic milling,'' *Procedia Manuf.*, vol. 10, pp. 486–496, Jan. 2017, doi: 10.1016/j.promfg.2017.07.034.

ALBERTO J. ÁLVARES graduated in mechanical engineering from the University of Brasília, in 1986, the master's and Ph.D. degrees in mechanical engineering from the Federal University of Santa Catarina, in 1990 and 2005, respectively, and the Ph.D. degree from Jaume University, Spain, in 2017. He is currently an Associate Professor in mechatronics engineering with the University of Brasília.

EFRAIN RODRIGUEZ received the B.S. degree in mechatronics engineering from the University of Pamplona, Colombia, and the M.S. degree in mechatronic systems from the University of Brasília, Brazil, where he is currently pursuing the Ph.D. degree in mechatronic systems with the Department of Mechanical Engineering. His research interests include parallel robots and additive manufacturing.

CRISTHIAN I. RIAÑO JAIMES received the M.S. degree in industrial control and automation from the University of Pamplona, Colombia. He is currently pursuing the Ph.D. degree in mechatronic systems with the Department of Mechanical Engineering, University of Brasília. His research interests include closed-loop inspection and advanced manufacturing systems.

JUAN S. TOQUICA received the M.Sc. degree in mechatronic systems from the University of Brasília, Brazil, where he is currently pursuing the Ph.D. degree. His research interests include precision robotics, industrial robots intelligence, and advanced manufacturing systems.

JOÃO C. E. FERREIRA received the Ph.D. degree in mechanical engineering from The University of Manchester Institute of Science and Technology (UMIST), England, in 1991. He was developed Postdoctoral activities in industrial engineering and manufacturing with Pennsylvania State University, USA, from 1999 to 2000. He is currently a Full Professor with the Department of Mechanical Engineering, Federal University of Santa Catarina.

• • •