

# *Virtualisation process of a sheet metal punching machine within the Industry 4.0 vision*

**Aitor Moreno, Gorka Velez, Aitor Ardanza, Iñigo Barandiaran, Álvaro Ruíz de Infante & Raúl Chopitea**

**International Journal on Interactive Design and Manufacturing (IJDeM)**

ISSN 1955-2513

Volume 11

Number 2

Int J Interact Des Manuf (2017)

11:365-373

DOI 10.1007/s12008-016-0319-2



**Your article is protected by copyright and all rights are held exclusively by Springer-Verlag France. This e-offprint is for personal use only and shall not be self-archived in electronic repositories. If you wish to self-archive your article, please use the accepted manuscript version for posting on your own website. You may further deposit the accepted manuscript version in any repository, provided it is only made publicly available 12 months after official publication or later and provided acknowledgement is given to the original source of publication and a link is inserted to the published article on Springer's website. The link must be accompanied by the following text: "The final publication is available at [link.springer.com](http://link.springer.com)".**

# Virtualisation process of a sheet metal punching machine within the Industry 4.0 vision

Aitor Moreno<sup>1</sup> · Gorka Velez<sup>1</sup> · Aitor Ardanza<sup>1</sup> · Iñigo Barandiaran<sup>1</sup> ·  
Álvaro Ruíz de Infante<sup>2</sup> · Raúl Chopitea<sup>2</sup>

Received: 16 May 2016 / Accepted: 26 May 2016 / Published online: 6 June 2016  
© Springer-Verlag France 2016

**Abstract** This paper presents the process to construct a digital twin for a sheet metal punching machine to support the interactive design of optimal NC machining programs. The results show that this entity manages to simulate interactively the basic behaviour of the actual sheet metal machine: movements, machining operations and connectivity with robotic arms. Using Ethernet/IP communication protocol, Digital Twins can be connected and cooperate to simulate virtual production lines. The paper concludes with the necessity of going deeper in the virtualisation of the sheet metal machining process, by adding more realistic physical behaviour (heat transfer, accelerations...) and connecting the Digital Twin with real mechanical parts under new simulation tools.

**Keywords** Industry 4.0 · Digital twin · Simulation · Visualisation

## 1 Introduction

It is a fact that the manufacturing landscape is changing by the introduction and adoption of the novel concepts presented under the Industry 4.0 tag. The technologies behind the Industry 4.0 concept will allow manufacturers to create new market opportunities regarding the products and services around the factories of the future.

The concepts addressed by the Industry 4.0 comprise several disciplines: from robotics, computer vision and simulation to knowledge management, big data analysis and interactive design. Heng [8] stated that all those technologies behind the Industry 4.0 conceptualisation are lacking of cohesion.

Visual computing is a conceptualisation of different isolated technologies working together to solve real industry problems. It can be considered as one of the key enabling technologies for the realisation of the Industry 4.0 vision [12] and it shows how this mixture of different and complementary technologies, such as computer graphics, computer vision, human machine interaction and multimodal and multimedia technologies, can emerge from the heterogeneous landscape of technologies [1, 2, 15].

Simulation techniques can be integrated perfectly with visual computing [12]. For example, it is quite common in the gaming industry the utilisation of real-time physical simulation engines to simulate the Newtonian physics [17]: gravity, collisions, forces and friction. The interactive design of custom and complex surfaces like the wings of an airplane also require the smart integration of the results from simulation engines (air turbulences, engine combustion patterns...) and

**Electronic supplementary material** The online version of this article (doi:10.1007/s12008-016-0319-2) contains supplementary material, which is available to authorized users.

✉ Aitor Moreno  
amoren@vicomtech.org  
  
Gorka Velez  
gvelez@vicomtech.org  
  
Aitor Ardanza  
aardanza@vicomtech.org  
  
Iñigo Barandiaran  
ibarandiaran@vicomtech.org  
  
Álvaro Ruíz de Infante  
a.ruizdeinfante@lantek.es  
  
Raúl Chopitea  
r.chopitea@lantek.es

<sup>1</sup> Mikeletegi Pasealekua 57, 20009 Donostia - San Sebastián, Spain

<sup>2</sup> Lantek Investigación y Desarrollo, Ferdinand Zeppelin 2, Miñano, Álava, Spain

the user decisions and criteria. This intertwined circle (simulation results, human and the interactive design) is key in the visual computing approach.

The industrial processes require harder requirements of what the simulations have to do [4], being one of the most important the accuracy of the simulation results. In the manufacturing scenarios, the simulation of the machining processes provides measurable benefits for the manufacturing industries, as lot of resources and energy can be saved and the workers safety can be increased if the machining processes are optimised.

The interactive design and programming of the machining processes can be simulated virtually. Once this stage fulfils the design objectives and tolerances, the produced NC program is transferred to the physical machine, expecting that the interactive simulation results will match the actual results of the machine. If this happens, the virtual simulated machine is said to be a Digital Twin of the physical machine [16], and it could receive the feedback from the physical machine and coexists during the manufacturing process.

This paper addresses the creation process of the Digital Twin of a material removal machine so that it can be used within a simulator to obtain valuable information for the designers and programmers without wasting resources.

The next section describes the simulation case study and its motivation. Section 3 describes the physical scenario, and Sect. 4 explains the steps followed for its virtualisation. Section 5 presents the results obtained after the virtualisation process. Section 6 presents some conclusions and some future work to address the further reduction of the distance between the virtual and real aspects of the Digital Twins.

## 2 Case study: sheet metal punching simulator

This case study is part of a joint project between Lantek, a company focused on developing software solutions for the sheet metal and steel fabrication industry and Vicomtech-UK4, a research institute specialised in Computer Graphics. One of Lantek's flagship products is Lantek Expert Punch [10], an advanced CAD/CAM nesting software system designed for automating the CNC programming of sheet metal punching machines.

Punching is a metal forming process that uses a punch press to force a tool, called a punch, through the sheet metal to create a hole or a certain deformation. Punching is often the cheapest method for creating holes in sheet metal in medium to high production volumes. The punching machine is controlled by a CNC, which automates the process executing a previously programmed machining.

Lantek Expert Punch is focused in the interactive design of optimal CNC programs for targeted punching machines. The produced nesting layout of the parts requires machining para-

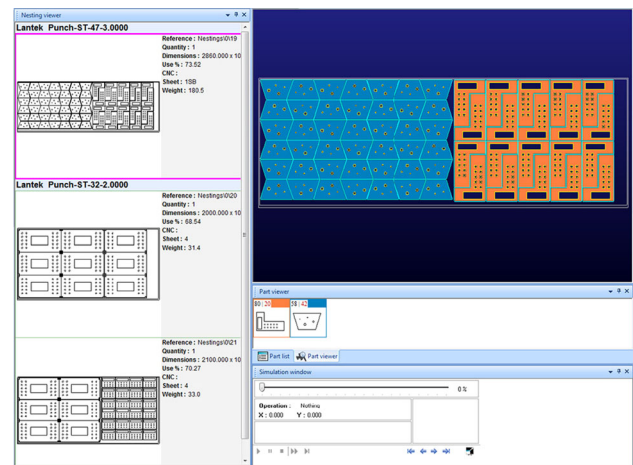


Fig. 1 Lantek Expert Punch CAD/CAM nesting software solution

meters like the physical dimensions of the sheet, the material properties of the sheet, the configuration of the turret's stations or the installed punching tools. All this information is taken into account in the nesting generation. The users can make interactive modifications to this produced nesting as their experience might recommend better strategies to obtain the results. This iterative design process of the nesting layout ends when the final CNC program is generated and send to the actual punching machine for testing. Interactive simulation tools can be used during this design phase to check the validity of a given nesting configuration.

Lantek Expert Punch, as the rest of the state-of-the-art solutions, is based on a 2-dimensional simulation (see Fig. 1). However, it is not possible to visualise the complete machining scenario using only a 2-dimensional view. The punching process is a complex process that includes many different methods and tools. This diversity leads to a variety of holes or deformations that the punching tool can do on the sheet metal. Some of these deformations can only be differentiated with a 3D representation of the sheet metal.

Non only the machining tools and their effects on their sheet metal are diverse, the turret is also one of the most important and variable parts of the punching machines. The turret is composed of stations, that can allocate multiple tools. Each station can also function as a turret, allocating other stations in a hierarchical way. The stations and the turret itself can rotate up to 360°, creating infinite combinations.

Lantek identified that existing CAD/CAM software tools are mainly based on 2-dimensional simulations and are not able to model some important features and events of the machining process. Their clients found it difficult to predict the final geometrical deformation using certain complex tools, or the potential collisions between different parts of the machine. It then became clear that it was necessary a more comprehensive modelling of the machining scene using a 3D representation. A major goal of the project is knowledge





**Fig. 2** Punching machine CHROMA, subject of the case study

transfer between Vicomtech-IK4 research team and the software engineers at Lantek, as to how the machining scene should be modelled and rendered.

The case study presented in this work is focused on the creation of a 3D simulator of a sheet metal punching machine. In other words, the main objective is to create a Digital Twin of the physical machine. The Digital Twin of the physical punching machine is composed of (1) the 3D virtual models, (2) the mechanical behaviour (axis movements) and (3) the sheet metal machining operations. Also, the modelling of the interoperation capabilities of the machine with other systems is required. In our case study, a robotic lift arm is used to extract the produced parts during or after the machining process. Also, a trap mechanism can be embedded in the punching table to allow part extraction. The next section describes the physical machine, and then, on the following section, the steps followed for its virtualisation are explained.

### 3 Description of physical scenario

The case study uses the CHROMA electric punching machine from the DANOBAT group [7] (see Fig. 2). It has a circular turret which can store up to 43 stations and 20 auto-index stations and it can machine a maximum material thickness of 6 mm. The punching machine can be configured with other additional options to fulfill the customers needs. For example, a robotic arm can be attached to load and unload the sheet metals and produced parts.

#### 3.1 Punching machines behaviour

The punching machine has three main movements: (1) the sheet movement, (2) the circular turret movement and (3) the punching action. The sheet is moved by clamps attached to the X and Y axis of the machine. The punching action is produced by a tool located in a fixed angle of the turret. The turret rotation allows the machine to change the current punching tool.

The movements of the punching machine are only one half of its behaviour. The other half corresponds with the effect of the punching action in the sheet metal. This behaviour encloses a variety of effects, such as forming, embossing, threading, rolling or stamping, but the material removal is the most used in the sheet metal machining.

There are other non primary actions with complex behaviour regarding the state of the punching machine. For example, the punching machine can modify the clamps position over the sheet. This movement called *Reposition* allows the machine to reach further areas of large sheet metals. First, the sheet is moved to the a position. Then, the sheet is released and the axis are moved to other position without the sheet. Finally, the sheet is clamped again. The translation offsets have to be taken into account for the following punching instructions.

#### 3.2 Relationships with other systems

The CHROMA punching machine can be used independently, but also it can be attached to other modules and cooperate. For example, the CHROMA can be used in combination with a shear module or a laser cutting module. Additionally, a robotic lift arm can be attached to the CHROMA machine to support the loading and unloading of the sheet metals and the produced parts during the machining process.

The next section presents the virtualisation methodology used to create a Digital Twin of the punching machine. In addition to the main functionality of the machine, the extraction mechanisms have been selected as use cases of how the relationships with other subsystems are included in the methodology.

### 4 Virtualisation method

The virtualisation process can be divided in several stages:

1. Model in 3D the individual parts of the machine, the parts of the robotic arm and any other necessary 3D elements.
2. Extract the behaviour of the machine and create a knowledge representation.
3. Model the interaction between the punching machine and other moving elements and integrate it to the knowledge representation.
4. Model the machining operations and the effects on the sheet metal.
5. Design and develop a simulator where all virtual elements cooperate to bring a virtual experience as close as possible to the physical machining behaviour.

This section explains how each of these stages was implemented in the current case study.

#### 4.1 3D modelling of the machinery

The 3D models provide the most visual impact regarding the exactness of the virtualisation. The closer to the physical machine, the more realistic immersion will be obtained.

Taking into account that the virtual model will be used in interactive simulations, the 3D models have to be light enough to be used with out-of-the-shelves computers and mobile devices in native applications or through HTML5 web interfaces.

In order to produce the appropriate 3D models, the CAD models of the machine were segmented in separated 3D parts. Only the necessary and visible elements should be modelled. For example, all the electronics embedded in the control boxes can be discarded by modelling only their containers. For the modelled objects, a polygon simplification and texturing process is carried out to reduce their number of polygons to match the interactive capabilities of the simulator and to provide a good looking appearance of the machine.

For a given punching machine representation, the modelled 3D objects are assembled into a scene graph, where the root node represents the whole machine and each internal node represents moving groups or axes of the punching machine. Using this scene graph representation, moving an internal transform node or axis will translate all its children elements.

Different machine configurations of the machine will have different scene graph representations and each machine configuration is saved as XML files. Therefore, the simulator will create a 3D virtual scene by reading one of the existing 3D machine configuration in XML. Loading this XML file loads and arranges all the 3D individual objects and prepares the required internal structures to access and interact with the moving axes of the machine.

#### 4.2 Extracting behaviour and knowledge representation

The behaviour of the punching machine and all of its functioning parameters are modelled and represented using carefully designed ontologies. Semantic reflects the meanings of objects, terms, features, code, message or any other form of representation. The most used formal approach to describe semantics is ontology [5].

The ontologies are defined in XML schemas, and then, the instances are stored as a set of XML files. The simulator loads these files and, in combination with the 3D scene graph, the semantically enriched information is loaded into the 3D virtual environment.

The set of instructions of the NC program are converted to a XML file, acting as an abstraction of the different G-code dialects. Using this XML representation of the NC instructions, the simulator becomes independent of the

actual G-code instructions which are specific to individual machines.

#### 4.3 Modelling the interaction between the punching machine and other moving elements

The punching machine and the sheet metal are the essential elements in a punching process. However there are more physical elements that can also interact with the machine or the sheet metal. This is the case of the robotic arm or the trap door. It is important to model the relationships and the interactions between these moving elements and the machine. At least, these questions need to be answered for each identified moving element:

1. What are the dynamics of the moving element?
2. How is the moving element controlled?
3. When is it activated?
4. Are its movements completely decoupled from the machine?
5. Can it collision with the machine?

We can take the trap door as an example, and answer the previous questions:

1. The trap door is situated in the machine, just under the sheet metal. It has a simple dynamic: when the trap door is activated, the cut parts are extracted from the sheet metal. When the extraction is finished, the trap door is closed. The opening angle and the opening/closing speed are retrieved from the semantic information attached to the trap configuration.
2. It is controlled by the same NC program than the punching turret. A special operation code will trigger the trap opening and other operation code will trigger the closing action. The speed and opening angle is taking into account in the virtual environment.
3. The opening and closing instructions are used to extract the sheet metal parts above the trap.
4. No, they are controlled by the same CNC.
5. No. But the physical simulation of the loose parts over the trap should be provided.

In order to extract the cut parts, it is necessary to be able to differentiate them from the rest of the sheet metal. A novel part detection algorithm [18] is used to obtain the cut parts in a virtually machined sheet metal.

#### 4.4 Modelling machining operations

Each machining operation has a different effect on the sheet metal. Some operations create a hole in the sheet metal with the profile of the punching tool, and others deform the sheet in

a certain way. It is then necessary to define a list of punching tools and assign to each of them, its effects on the sheet metal.

The simulation of machining operations was based on algorithms of computational geometry. These algorithms need to be efficient enough to allow the real-time modification of the sheet metal geometry. In order to simplify the problem these premises were taken:

- The sheet metal will be completely flat.
- The thickness will be constant in the same sheet metal, but it can change from one sheet to another.

While the tool is on the sheet metal, a geometric Boolean operation is performed between the tool and the sheet metals geometry. Although normally a Boolean operation between 3D solids is computationally expensive, a simplification is made by assuming that the Boolean operation can be performed between the 2D contour of the tool and the 2D representation of the sheet metal. Even with this optimisation, the geometry of the virtual sheet metal can become very complex and therefore, even the 2D Boolean operations can get very computationally expensive. We have introduced a spatial subdivision to address this issue successfully [11].

#### 4.5 Simulator application

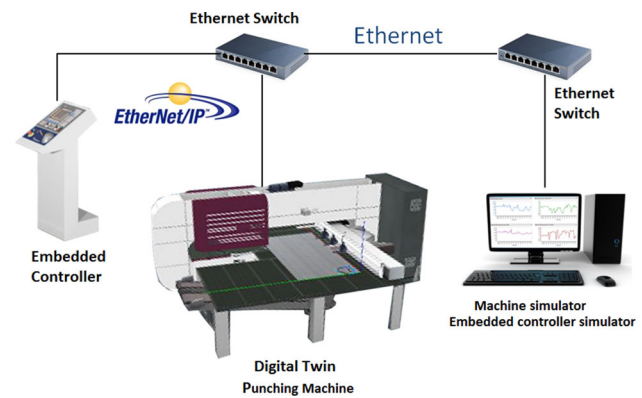
On the final stage, all the pieces are integrated and connected to each other. Following the traditional software coding good practices, making the software modular eases a lot the burden of integration. Not only a realistic and comprehensive modelling is necessary, also the user experience needs to be taken into account. As the final simulation tool needs to be usable, the user requirements need to be taken into account when designing the user interface. User centred methodologies that put the user in the center of the development cycle can help in this process [6].

#### 4.6 Digital twin communications

The ultimate goal of the Digital Twin vision is to produce an entity that can act as closely as possible to the real machine. There are three scenarios that can be approached:

- A virtual punching machine (the Digital Twin) connected to other physical devices and controllers
- A real punching machine connected to virtual devices (Digital Twins)
- Independent Digital Twins communicate and collaborate mimicking real production lines

All scenarios require to stablish physical connections among the entities (real or virtual). For this purpose, we



**Fig. 3** Any computer or controller can connect with the Digital Twin using Ethernet/IP

have added a communication layer to the Digital Twin of the punching machine which implements the low level communications using Ethernet/IP [9].

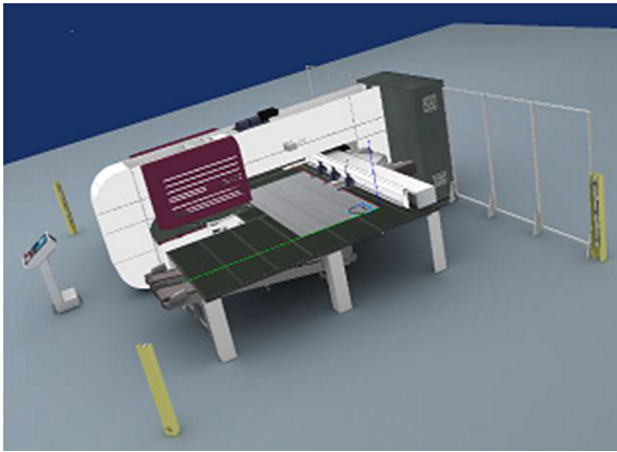
One of the main reasons advantages of using Ethernet/IP, developed by Rockwell Automation in 2001 and supported by ODVA [3, 14], is the compatibility of all the technology designed for Ethernet. This allows any PC that is connected to the Ethernet network to communicate with a machine that is using the Ethernet/IP protocol. Additionally, priority mechanisms (IEEE 802.1p) can also be implemented, which are important when a network is shared with other applications. As discussed in this analysis [13] about the performance of Ethernet based real time protocols, Ethernet/IP might be not the faster approach, but its other features give it an advantage over the alternatives.

A machine that uses the protocol Ethernet/IP communication can be controlled by a device consisting of an embedded with a touch screen that is installed a few meters from this, or could be monitored in real time from a PC in another factory hall. The architecture of the first scenario is depicted in Fig. 3: any computer or embedded controller on an Ethernet network can interact with the Digital Twin by connecting to the server via Ethernet/IP communication.

## 5 Results

An interactive simulator of the punching process has been developed following the steps described in the previous section. This simulator uses a combination of Computer Graphics techniques, simulation techniques and knowledge representation.

The simulator is currently part of Lantek's catalogue as a helping tool for the interactive design of CNC programs for punching machines, what demonstrates the success of the research project. This section summarises the main results obtained in the virtualisation of the punching process.



**Fig. 4** Digital Twin of the CHROMA punching machine

### 5.1 Virtual scene

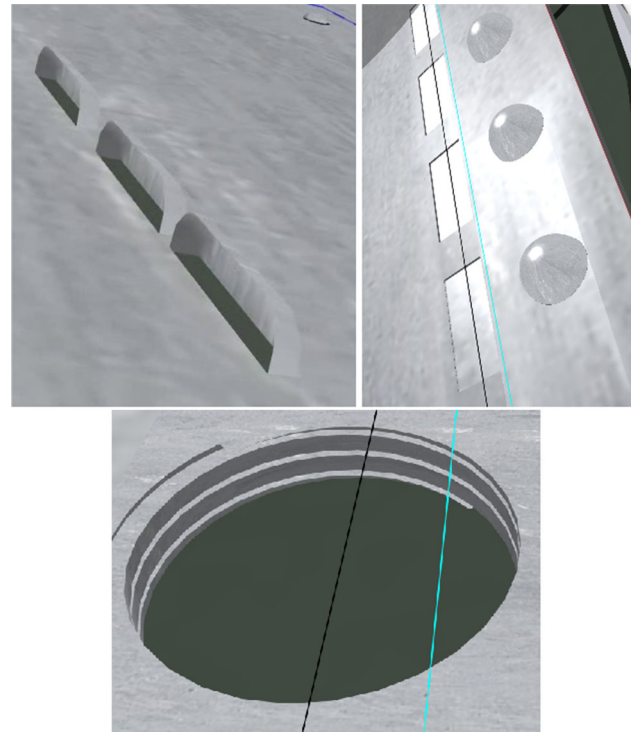
The virtual scene is created by loading the punching machine and the rest of the virtual elements (see Fig. 4). The machine representation is given in XML files and it basically arranges the 3D models in a graph representation. Additional linked XML files define the positions of the stations installed in the circular turret and the actual punching tools that are present in the turret stations.

Additional modules are also linked to the machine representation as XML files (the robotic arm or the trap). These modules interact and cooperate with the punching machine. They also share relevant information like the zero point or origin of the punching machine. The robotic arm will use this information as the reference point to pick elements from the sheet metal.

The rest of the elements may be just decoration 3D models (floor, building structure ...), static or with animations. Nevertheless, they are mapped as individual objects in the virtual scene and they can be selected to be hidden or to change its transparency.

### 5.2 Machining operations

Once the virtual scene is generated and the set of instruction is loaded, the simulation can begin. Each movement instruction defines precisely the targeted point of the sheet metal, its path (linear or arc), and its speed (linear speed or angular speed). Any other instruction is also timed. For example, the tool change instructions specify the amount of time that the simulator has to wait to complete the action. The simulator manages internally the punching machine state, which includes the current punching tool and the sheet position. Therefore, the punching machine can operate on the sheet metal appropriately when a punching instruction is executed.



**Fig. 5** Punching operations. *Top left* forming. *Top right* marking and semi-sphere deformation. *Bottom* threading

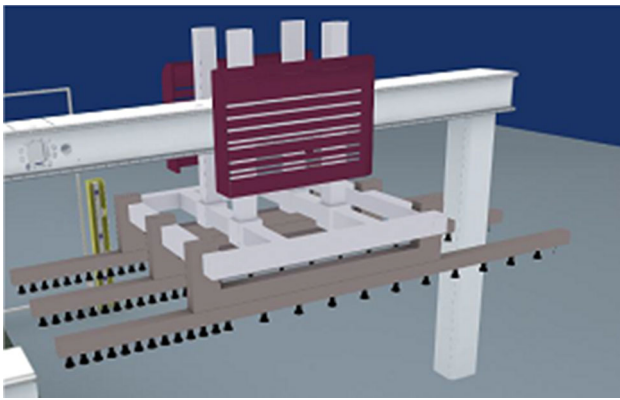
The punching machine main action is to perform the punching action on the sheet with a loaded tool in a turret station. At the exact punching time, the tool is projected very fast towards the sheet metal and then, it returns to the resting position.

The typical punching operations are oriented to cut the sheet metal (producing holes in it), but there are many other punching instructions which modify the sheet metals geometry in different ways. The virtualisation process of such instructions improves the realism and capabilities of the simulator (see Fig. 5). The current implementation of such advanced punching instructions allows the deformation of the sheet metals geometry to support several expected outputs: forming, embossing, threading, rolling and marking.

### 5.3 Part extraction with a robotic arm

An optional robotic arm can be attached to the punching machine to automate the unloading of the machined parts during the simulation (see Fig. 6). The lift is fully customizable. It can have any number of strips in X or Y directions. Each strip has a number of suction cups that will be the ultimate responsible for lifting the sheet metal parts. They can be activated individually or in predefined groups. The arrangement of the suction cups is also configured in separate XML files.





**Fig. 6** Robotic arm attached to the punching machine

A specific NC instruction is used to activate the lift, passing the target point as parameter of this instruction. The lift will use this reference point and its own configuration (regarding axis orientation and strip configuration) to move the 3D models towards the specific point over the sheet metal. This movement is not a direct line from one point to another. It is decomposed in X, Y and Z movements, and the movements are sent in that specific order. The instruction also specifies the speeds of the lift in each axis movement, and therefore, the total elapsed time can be calculated easily as the travelled distances are known.

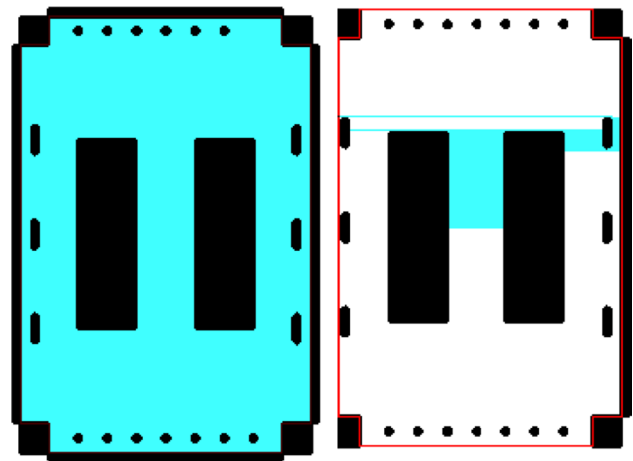
The behaviour model of the robotic arm is completed with the unloading action, i.e., when the arm returns to the resting position with the loaded parts. The deactivation instruction does not require additional parameters, but the movements of the axis have to be inverted: Z, Y and X. At the end of the movement sequence, the parts are unloaded. In order to optimize the hosting computer's memory resources, after a given time, the unloaded parts are removed from the scene.

From the sheet metals point of view, a geometric operation has to be done to detect if there are loose parts under the active suction cups of the lift. The spatial subdivision [11] breaks the sheet geometry into disjoint pieces, and therefore, any knowledge about the existing parts inside the sheet is lost. Using Computer Vision techniques like flood filling algorithms, the loose parts can be reconstructed, segmented, detected and finally, removed from the rest of the sheet metals geometric representation (see Fig. 7).

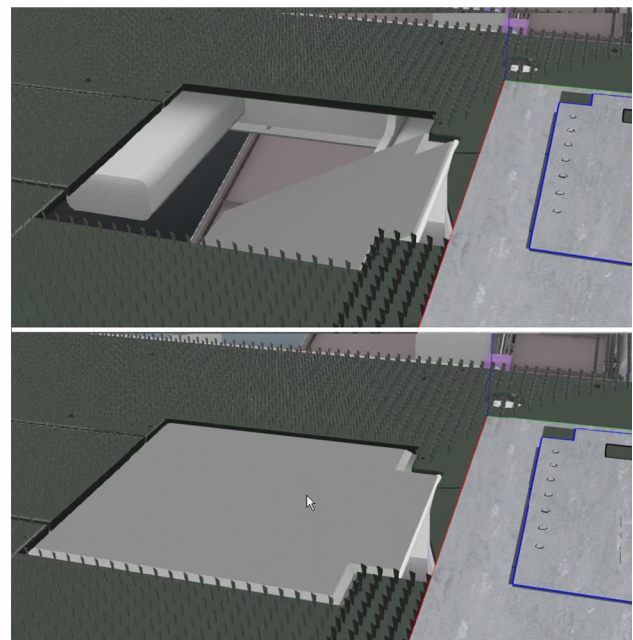
The part detection process is triggered only when the lift movements are finished and the activation of the suction cups is performed. At that precise moment, all the touching points of the active suction cups and the sheet metal are calculated and passed as seeds to the flood filling algorithm.

#### 5.4 Part extraction with trap doors

The part extraction process with a robotic arm can be complemented with trap doors installed in the punching table. A



**Fig. 7** The part detection was successful in the first image. In the *second image*, the part is discarded as it has not been completely cut



**Fig. 8** Trap door installed in the punching machine. *Top* opened. *Bottom* closed

punching machine may have a large trap with a door mechanism that can be opened and other auxiliary traps without any door. Basically, they are just holes in the table, normally not very big and they are intended to extract only small parts or scrap.

The inclusion of traps in the machine requires the modification of the 3D models of the punching table. Therefore, a given traps configuration should be accompanied with the compatible 3D model of the punching table (see Fig. 8).

Once the 3D models are prepared, the behaviour and dynamics of the moving traps has to be defined. In this case, a new pair of instructions is required to operate the traps (TRAPOPEN and TRAPCLOSE). The functions are para-

metrized with the opening angle of the trap and the angular speed. From these parameters, the elapsed time of the opening instruction is calculated. The closing action is just the reversed movement.

There are two possible utilisation of the trap mechanism. The first one positions the metal sheet over the trap and then, the trap is opened. At this point, the part detection algorithms are triggered and any loose parts in the affected region of the trap are removed from the metal sheet geometrical representation. The second utilisation method opens the trap before positioning the metal sheet over the trap. With this method, this movement cannot be done at full speed as potential collisions may occur. Using an appropriate speed, the metal sheet can be safely positioned over the trap and any detected loose part will be removed. This extraction method is also used with traps without doors, as they can be conceptualized like traps that are always in opened state.

The punching machine has access to the traps installed and their state. The moving speed of the metal sheet has to be controlled when it is intended to pass over an opened trap. Fast movements could lead to potential collisions and higher risk of malfunction. The simulator may warn about potential risky situations by comparing the program trajectories and their linear speeds with the trap's location and state.

### 5.5 User interface

The user interface of the simulator has been developed in Qt. The 3D virtual scene is rendered in a custom widget positioned in the center of the application. The basic functionality presented to the user is divided in three moving and dockable panels.

The control panel encloses the VCR functionality, i.e., the play, stop, fast-forward and rewind actions. Also, it includes some buttons to control the speed of the simulation.

The visualization panel presents the 3D objects of the scene in a hierarchical tree that reflects the scene graph of the loaded machine representation. The visibility and transparency of any object can be controlled with appropriate check boxes and sliders. The punching action occurs inside the circular turret and therefore, nothing is really seen if all the objects of the machine are rendered opaquely. Therefore, the elements near the punching actions (turret and supporting structure) should have a custom level of transparency. In our use case, a 40 % transparency gives a good tradeoff.

The CNC panel shows the loaded program showing the current instruction. Additionally, the user can select a instruction and the simulator will jump to that instruction by running a background simulation of the remaining instructions.

The Qt multilingual utilities are used to help the localization of the texts shown in the panels to any language. A custom function has been integrated in the simulator to change the shown language at any time.

### 5.6 Communications

The current tests show that the communication is feasible among different virtual entities: the punching machine and different modules of the machine (robotic arms, shearing module, laser cutting module and trap). The communication channel provided by Ethernet/IP works fine for TCP and UCP connections.

However, this communication mechanism has to be complemented with a high-level protocol for communicating the Digital Twins. This protocol should be defined to match the extracted knowledge of the punching machine behaviour (addressed in Sect. 4.2) and some automatic code generation tool should be used to convert the knowledge representation into actual C++ code in top of Ethernet/IP classes.

### 6 Conclusions and future work

This paper has shown how a Digital Twin of a commercial punching machine has been created and how it can be used as a helping tool in the interactive design of CNC programs for the punching machines. The creation process can be divided in several stages: (1) model in 3D the individual parts of the machine, the parts of the robotic arm and any other 3D elements, (2) extract the behaviour of the machine and create a knowledge representation, (3) establish the relationship between the machine and the robotic arm and integrate it to the knowledge representation, (4) model the machining operations and the effects on the sheet metal and (5) design and develop a functional interactive simulator where all pieces cooperate to bring a virtual experience to the users of the physical machining operations.

Even if the results show a clear resemblance with the output of the physical machining processes, there is room for improvement. For example, the physical parameters of the process are not taking into account. The current implementation can punch sheet metals with any thickness with the same punching force. The continuous addition of new physical behaviour in the simulator will bring the Digital Twin closer to the physical machine, closing the gap between reality and virtuality.

Another interesting future research line is related to the connectivity and synchronisation of the Digital Twin with the real punching machine. It is expected that the feedback coming from a working punching machine could be passed to its Digital Twin. Then, the simulator would replicate virtually the movements and reactions of the real punching machine.

### References

1. Acatech: Final report of the Industrie 4.0 Working Group. Tech. Rep. (2014). [http://www.gtai.de/GTAI/Content/EN/Invest/\\_Sha](http://www.gtai.de/GTAI/Content/EN/Invest/_Sha)

- [redDocs/Downloads/GTAI/Brochures/Industries/industrie4.0-smart-manufacturing-for-the-future-en.pdf](#)
2. Blanchet, M., Rinn, T., von Thaden, G., de Thieulloy, G.: Industry 4.0: the new industrial revolution. How Europe will succeed. In: Think Act, Roland Berger Strategy Consultants GmbH (2014)
  3. Brooks, P.: Ethernet/IP-industrial protocol. In: 8th IEEE International Conference on Emerging Technologies and Factory Automation, vol. 2, pp. 505–514 (2001)
  4. Drath, R., Horch, A.: Industrie 4.0: hit or hype? [industry forum]. IEEE Ind. Electron. Mag. **8**(2), 56–58 (2014). doi:[10.1109/MIE.2014.2312079](#)
  5. Folkert, K., Fojcik, M.: Ontology-based integrated monitoring of hadoop clusters in industrial environments with OPC UA and RESTful web services. In: Computer Networks, CN 2015, *Communications in Computer and Information Science*, vol. 522, pp. 162–171 (2015). doi:[10.1007/978-3-319-19419-6\\_15](#)
  6. Gardan, J.: Definition of users' requirements in the customized product design through a user-centered translation method. Int. J. Interact. Des. Manuf. (IJIDeM), 1–9 (2015). doi:[10.1007/s12008-015-0275-2](#)
  7. Group, D.: CHROMA, electric punching machine. Tech. Rep. (2016). <http://www.danobatgroup.com/en/punching-machines/electric-chroma>
  8. Heng, S.: Industry 4.0: upgrading of Germany's industrial capabilities on the Horizon (2014). Available at SSRN: <http://ssrn.com/abstract=2656608>
  9. Hu, T., Li, P., Zhang, C., Liu, R.: Design and application of a real-time industrial Ethernet protocol under Linux using RTAI. Int. J. Comput. Integr. Manuf. **26**(5), 429–439 (2013). doi:[10.1080/0951192X.2012.731609](#)
  10. Lantek: Lantek Expert CAD/CAM Nesting software. Tech. Rep. (2016). <http://www.lanteksms.com/uk/lantek-expert>
  11. Moreno, A., Segura, Á., Arregui, H., Ruiz de Infante, Á., Canto, N.: Real time 3d simulation tool for nc sheet metal cutting and punching processes. In: Industrial Simulation Conference 2012, pp. 55–62. Eurosis (2012)
  12. Posada, J., Toro, C., Barandiaran, I., Oyarzun, D., Stricker, D., De Amicis, R., Pinto, E., Eisert, P., Dollner, J., Vallarino, I.: Visual computing as a key enabling technology for industrie 4.0 and industrial internet. IEEE. Comput. Graph. **35**(2), 26–40 (2015). doi:[10.1109/MCG.2015.45](#)
  13. Robert, J., Georges, J.P., Rondeau, E., Divoux, T.: Minimum cycle time analysis of ethernet-based real-time protocols. Int. J. Comput. Commun. Control **7**(4), 744–758 (2014). doi:[10.15837/ijccc.2012.4.1372](#), <http://univagora.ro/jour/index.php/ijccc/article/view/1372>
  14. Schiffer, V.: The common industrial protocol (CIP) and the family of CIP networks. Tech. Rep. (2006). [https://www.odva.org/Portals/0/Library/Publications\\_Numbered/PUB00123R1\\_Common-Industrial\\_Protocol\\_and\\_Family\\_of\\_CIP\\_Networks.pdf](https://www.odva.org/Portals/0/Library/Publications_Numbered/PUB00123R1_Common-Industrial_Protocol_and_Family_of_CIP_Networks.pdf)
  15. Sha, L., Gopalakrishnan, S., Liu, X., Wang, Q.: Cyber-physical systems: a new frontier. In: SUTC '08. IEEE International Conference on Sensor Networks, Ubiquitous and Trustworthy Computing, pp. 1–9 (2008). doi:[10.1109/SUTC.2008.85](#)
  16. Stork, A.: Industry 4.0: cyber-physical equivalence. Tech. Rep. (2015). <https://www.igd.fraunhofer.de/en/Institut/Abteilungen/IE/T/Projekte/INDUSTRIE-40-CYBER-PHYSICAL-EQUIVALENCE>
  17. Tideman, M., van der Voort, M.C., van Houten, F.J.A.M.: A new product design method based on virtual reality, gaming and scenarios. Int. J. Interact. Des. Manuf. (IJIDeM) **2**(4), 195–205 (2008). doi:[10.1007/s12008-008-0049-1](#)
  18. Velez, G., Moreno, A., Ruíz De Infante, Á., Chopitea, R.: Real-time part detection in a virtually machined sheet metal defined as a set of disjoint regions. Int. J. Comput. Integr. Manuf. **0**(0), 1–16 (0). doi:[10.1080/0951192X.2015.1130263](#)