

FUNCTIONAL BEHAVIOUR SIMULATION OF INDUSTRIAL PRODUCTS IN VIRTUAL REALITY

Fabio Bruno

Department of Mechanical Engineering
University of Calabria
Italy
f.bruno@unical.it

Agostino Angilica

Francesco Cosco

Maurizio Muzzupappa

Department of Mechanical Engineering
University of Calabria
Italy

{agostino.angilica, francesco.cosco, muzzupappa}@unical.it

ABSTRACT

VR (Virtual Reality) is a powerful tool for the simulation of virtual prototypes, because it allows engineers to enhance the analysis and validation of the digital product before manufacturing any physical mock-up. Unfortunately, VR software tools are not able to fully simulate the behaviour of a virtual product, because they are mainly conceived to reproduce the visual appearance of the product: the functional simulation is limited to basic behaviours related to the animation of the objects in the virtual world (e.g.: part movements). This paper describes an innovative approach to create functional behaviour simulations in VR using the same models and the same software employed by the engineers in the design phase. Our approach is based on the run-time connection between the VR software and the simulators used for product design and analysis. This means that there is no need to write code for describing the product behaviour, and any modification done on the behaviour models is immediately testable in VR. It is apparent that these advantages allow to reduce the time needed to implement the virtual prototyping, thus achieving a more efficient design process.

KEYWORDS

Product design evaluation, interactive virtual prototype, functional behaviour simulation, finite state machine

1. INTRODUCTION

Virtual Reality (VR) technologies are widely employed for the aesthetical validation of industrial products. VR applications, in fact, provide a high quality, immersive visual representation of virtual prototypes, so that designers can easily evaluate aesthetical qualities and/or discover any styling defect. Recently, VR has been used also for the simulation of virtual prototypes, because it allows engineers to enhance the analysis and validation of the digital product, before manufacturing any physical mock-up. Through VR it is also possible to conduct training sessions without any risk for the users (as it is in the case of particular products like vehicles, machine tools, etc.) or to carry out usability analyses [19]. In this second case, VR demonstrates to be a valid tool, because it is able to support and facilitate the participatory design of industrial products. VR allows designers to involve the final users of a product since the early design stages, without the need of a physical mock-up and with the advantage of being able to assess several design options [12].

Unfortunately, VR software tools are not able to fully simulate the behaviour of a virtual product, because they are mainly addressed to the aesthetical validation of the product. The functional simulation that one may obtain is limited to some basic behaviours like movements triggered by an event (e.g.: a door will open when clicking on a handle). Some of the

most recent tools provide built-in physics simulators based on videogame technologies, in order to simulate physical phenomena like collisions and rigid bodies dynamics. Such simulators can provide fast results, but they are not as accurate and flexible as the simulation software used by engineers. As a consequence we don't obtain a simulation, but an animation which cannot be employed as a robust validation tool. Furthermore, when the product behaviour has been simulated in VR [5, 3, 7, 21, 20], this has been done through the implementation of the code that replicates the behaviour models of the product inside the VR software. This means that each change in the digital mock-up has to be manually reported in the code of the VR application, thus requiring a big effort for the VR operators.

In previous works we tried to approach this problem proposing two solutions. The first [10] is a framework for numeric co-simulation and three-dimensional visualisation. This framework is addressed, above all, to the simulation of mechatronic products, in which different physical domains have to be simulated. It provides an approach to achieve a co-simulation using several heterogeneous solvers and, at the same time, visualise the results in a three-dimensional environment. The second [9] is a software tool that links a multi-body solver with a VR environment in order to allow the designer to simulate the dynamic behaviour of an excavator. The simulation software computes the displacements and the rotations of the mechanical parts, and sends the results to the visualisation environment, which moves the parts in accord with the simulation results.

This paper describes a research that extends this idea by experimenting another case study where not only multi-body, but also other kind of simulations (e.g.: electric, electronics, software) are needed to replicate the behaviour of the virtual prototype. The main goal of this research is to develop a VR environment in which the functional behaviour simulation is performed in VR using the same models and the same software employed by engineers in the design phase. This approach allows us to reach two goals. First of all, we can ensure the reliability of the simulation because it is performed by a solver with a level of precision acceptable for industries, and the behaviour models are exactly the same that engineers use to describe the product. Moreover, any modification done on the behaviour models is immediately testable in VR. Only in few cases the modification of the simulation model requires an adjustment of the digital

mock-up used in the VR environment (e.g.: when an element of the interface is added or deleted).

We implemented this idea extending the functionalities of an existing high level software library for the inter-process communication that had been previously developed to link a multi-body solver with a VR environment [9]. In particular, the library has been extended including the functionalities needed to transfer from the VR environment to the simulation software all the information related to the user generated event that may cause a feedback by the virtual product, e.g.: a button pressed, an action on a leverage, a part moved. On the other hand, the library allows also to send back (from simulation to VR) any modification that may occur in the virtual product, i.e.: not only part movements, as implemented before, but also the modification of a display, the change of a LED status, or other elements of the product interface.

We have chosen Matlab/Simulink to model and simulate the product behaviour. This environment is almost a standard for the general purpose simulation, thanks to its diffusion and versatility. Furthermore, a lot of optional packages (called toolboxes) are present. These packages provide further sets of high level operations for a specific task, that makes easier to implement the real-time simulation engine needed to fully reproduce each kind of behaviour. For the visualization, we have employed Virtools Dev because it is one of the most powerful authoring environment for interactive 3D graphics and VR. Moreover, Virtools Dev functionalities can be extended through its own C++ Application Program Interface (API).

2. RELATED WORK

Among the various utilisations of VR in the industrial field, Virtual Prototyping seems to be one of the most promising and challenging [19]. There are several examples where VR is used to simulate the behaviour of the product, a typical example is in training tasks. In [1] a training operator environment for a numerical control milling machine is presented. In [6] an innovative fork-lift simulator, suited for training in industrial environments is presented. The numerical model is modelled and simulated using the Open Dynamics Engine library, hence any modification to the model implies the modification of the application code. Other interesting examples about the use of VR and simulation for training tasks are present also in the medical field [17].

Augmented and mixed reality have been used as an

environment for the development of virtual prototyping applications. In [3] a mixed reality set-up has been realized in order to present a physical mock-up covered with a designed virtual mock-up that can be touched and used to evaluate the design. The set-up integrates ARToolkit with an electromagnetic tracker and uses rapid prototyping to realize the tangible dummy that is animated through the augmented reality visualization. The mixed prototyping approach is also adopted in an application for positioning information appliances within systems and for the evaluation of ergonomics aspects of interactive devices [7].

In [21] Park et al. present a study about the use of VR in product design evaluation. They focus their work on the simulation of the Human-Machine Interfaces (HMI) that are analyzed through a state transition methodology to capture the functional behaviour of the product, from which they construct a finite state machine for its functional simulation. The authors put in evidence that one of the limits of their approach is that “it is very meaningful and challenging to devise an efficient way to reduce the time and effort required to generate the HMI behaviour model of the product by reusing the information built in the embedded software”.

It is apparent that with our approach this problem can be easily overcome, because we propose to create a direct connection between VR and electric/electronics simulation environment. In a recent development, the same authors propose an extension of their approach adopting the use of tangible interfaces and augmented reality visualization [20]. They use rapid prototyping to create a physical mock-up, as also suggested in [3], and paste the markers needed for ARToolkit tracking on the physical model. The interaction with this augmented tangible prototype is done through a paper pen on which an ARToolkit marker is placed. Authors still adopt the same methodology proposed in their previous work to simulate the product behaviour [21].

The previously mentioned works put in evidence that there is a lack of software tools able to support designers in the development of interactive and functional virtual prototypes. This lack is probably one of the main obstacles in the diffusion of VR techniques for the product behaviour simulation. Engineers, in fact, use specialised simulation software to design the industrial product: at the moment it is not possible to evaluate the models developed in these simulation packages directly in VR. As pointed out in [12] it is necessary to work out a specific solution

for the several problems which occur during the integration between VR and the other applications of the product development process, including the simulation.

Some interesting works about the integration of simulation and VR have been realized in the Multi-Body Systems (MBS) field. Cuadrado et al. [14] present an application for the virtual prototype of a car, where manoeuvrability evaluation in a VE is carried out by integrating 3D visual and haptic feedback. Eberhard and Li [15] also present MBS simulation and control applications using VR as a user interface. In both these cases, the MBS real-time simulations represent the main limitation. Antonya and Talaba [2] present a novel approach to simplify MBS computations in the case of visual evaluation and modification of virtual mechanical systems. The applications developed by the authors incorporate common interactive facilities offered by a virtual environment, e.g. stereoscopic visualisation in a CAVE-like immersive system, walk-through, interactive object handling. These interesting works demonstrate the added value of integrating VR with simulation in order to optimise a design.

Another interesting approach to the integration of simulation packages and VR has been presented by Sanchez et al. [22], who developed the Easy Java Simulation (Ejs) a software tool designed to create interactive simulations in Java using models created with Simulink. Ejs can also be used with Java 3D to create interactive 3D virtual products, but it has been conceived mainly for educational purposes and it cannot be efficiently integrated into a design process, because Java 3D is not suitable for the visualisation of complex models.

In [23], an environment for the behavioural simulation of CAD assemblies is presented. The global model is formed by several component with a behaviour (simulation model) and a form (CAD model) connected through a port-based paradigm. The framework performs a numerical simulation to predict the behaviour, but no visualisation of the CAD model occurs.

The analysis of the state of the art puts in evidence that there is the need to create realistic product simulations in VR for several purpose like tests with the users, like training, functional validation, etc.. At the moment, previous researches have been focused on the devices that better allow the user to interact with the virtual or augmented mock-up. Some authors put

in evidence that, in order to make this approach more effective, it is necessary to reduce the time required to implement the behaviour simulation of the product. In our opinion, this problem can be overcome with the simultaneous use of VR and numerical simulation environments. In fact, the simulation models of the product are often present in manufacturing companies, but they cannot be used to simulate the product in VR. Our approach, instead, is based on the inter-process communication among different software modules, and relies on a middleware for the software communication. This approach allows to freely choose the software to be employed, both for simulation and visualisation. In this way engineers can use their favourite software during the product development phase.

3. INTERACTIVE VIRTUAL PROTOTYPING

The approach proposed in this paper for the creation of interactive virtual prototypes is based on the idea that the most effective way to replicate the behaviour of the product inside a virtual environment is by coupling a VR software with a simulation environment. This idea is mainly justified by the fact that, in many cases, the behaviour model of the product is already available since the first steps of the product development process. In fact, control engineers usually realize a behaviour model of the product using a Computer Aided Control Engineering (CACE) tool. This model is used not only to schematize the product behaviour but also to analyze how it works with a proper simulator. Figure 1 schematizes the proposed approach putting in evidence the role of the four actors involved in the process. The designer defines the geometries of the product and in particular the external shape, that is essential from the aesthetical point of view, and the appearance of the interface.

The VR operator, as explained in section 4.2, uses an authoring tool to prepare the virtual environment specifying which ‘input elements’ of the interface (button, knob, etc.) may be controlled by the user, and which ‘output elements’ (LED, LCD, etc.) may change their status during the simulation. On the other side the control engineer defines the behaviour of the product (see section 4.1 for more details), and connects the state transition of the model with the user generated events and the state variables with the ‘output elements’ of the User Interface (UI) defined in the virtual environment. This connection, that will be better explained later, allows a data exchange be-

tween the two environments, so that the simulation of the virtual prototype is managed by the same model that the control engineer uses for his/her design and analysis activities.

The three actors that we have identified on the left part of Figure 1 have to be intended as “roles played in the design process”. In some cases, e.g. in small enterprises, some of these roles may be played by the same person. For example, the VR environment may be prepared by the designer or by the control engineer. Extending this reasoning, we may assume that all the three roles may be played by only one actor with all the competences.

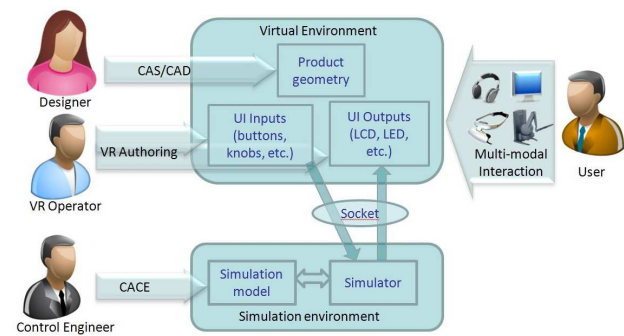


Figure 1 The schema of the proposed approach to virtual prototyping.

We assume that the final user is one of the actors of the process because, as clearly evidenced in literature [12, 5, 3, 7, 21, 20], VR tools are particularly effective in virtual prototyping when the user is involved in the testing phase. In other terms our approach allows the user to test what the engineers have designed by a direct interaction with the digital mock-up.

In this paper we present a case study to demonstrate how it is possible to realize an interactive virtual prototype connecting a simulation software with a VR development tool. In particular we use the Matlab/Simulink environment to model and simulate the product behaviour and Virtools Dev to create the virtual environment where the user interacts with the virtual product.

3.1. The link between simulation and VR

The link between Simulink and Virtools has been created by adapting an existing software library called SimLib that was previously developed for dynamic simulation in VR [9]. This library provides an easy to use Inter-Process Communication (IPC) channel using TCP socket, therefore it is possible to

run the simulator and the VR application on different machines. The SimLib library is quite easy to use. It has few functions, implementing the code for the TCP/IP communication and synchronization. Therefore, the developer must not take care of sockets and threads. The SimLib has been used to develop the customization for both Simulink and Virtools. To integrate the SimLib inside Simulink we have developed two custom S-Functions. These ones can be used within a Simulink model as a conventional Simulink block, with a user defined behaviour and set of actions. In this case, our S-Functions are responsible for the communication between Simulink and Virtools through the IPC channel provided by the SimLib. One S-Function receives the events generated by the user that interacts with the virtual product and switches different simulation parameters according to these events. The other S-Function sends the data that describe the state of the product and each change that occurs during the simulation. The data sending is based on an asynchronous channel: the S-Function sends data to Virtools, without stopping the simulation. In order to obtain a consistent visualization, in fact, it is necessary to simultaneously send all data at each time-step.

The Virtools socket connection for the IPC has been implemented through two user defined Building Blocks (BB). A BB is something like a black box with some parameters in input and some in output. By connecting several BBs, it is possible to create an interactive application in Virtools. The two BBs we have implemented to support the connection with Simulink are able to send and receive data through the IPC channel. The first BB sets the 3D model parameters in agreement with the data coming from Simulink. The second BB sends to Simulink the messages generated by the user interaction in the VR environment (e.g.: a button pressed, a knob rotated, etc.).

3.2. Creation of the virtual prototype

We assume that the Simulink model can be realized following a top-down process where the product behaviour is progressively refined until each component of the product has been defined. For example, the model we have realized for the test case can be considered as the first stage of the design process. In the subsequent steps of the process the model could be refined transforming the behaviour model (made of functions, formulas and equations) in a schema where each block represents a physical component

(e.g.: electric motor, actuators, processors, signal converters, etc.).

The geometric model of the product is usually defined following a bottom-up approach. Using a CAD system, the engineers model each single component or import them from previous projects or from the suppliers' catalogues. But in order to realize a virtual test with the product interface, only the external shape of the product is needed. All the components inside the object, that are not visible to the user, can be omitted in the virtual prototype. This allows engineers to start the test with the users as soon as the 'skin' of the product has been defined by the designers. When the shapes are ready, these are imported in the virtual environment and the properties of the interface elements are connected to the state variables of the simulation model. Then the events generated by the user are addressed to the simulator that considers them as triggers for the state transitions. In most cases the connection between the simulator and the VR software does not change during the process, because the exchanged data remain the same also when the behaviour model is updated and refined.

In the general case, at the end of the design process, several different models are defined for each engineering domain involved in the product. For example in our test-case we could have the Simulink model that simulates the electric and electronic behaviour and the virtual prototype with all the CAD geometries of each part assembled in a unique digital mock-up. The latter can also be used to realize multi-body and/or structural analyses using CAE software. Different models can also be used together to realize a mechatronic co-simulation coupling different simulators [8] or using an integrated simulation environment.

4. THE TEST CASE

The test-case we have analyzed to validate our virtual prototyping approach is a hot mixer (Figure 2). It is a quite complex appliance with several functions like weighing, blending, grinding, kneading, steaming and cooking.

The product presents a classic UI with a knob that allows the user to set the speed of the motor, eight buttons to set the temperature, a couple of buttons to set the timer, the stand-by button, and other three buttons for special functions: turbo, weighing, kneading and counter clockwise rotation. It has also a LCD that displays the state of the mixer (timer, weight, functions activated) and a set of eight LEDs, associated

to the temperature buttons, that show the current and the desired temperature.

The geometric modelling phase has been done following a reverse engineering approach. A photogrammetric software has been used to draw the profile curves that have been exported to a CAD where the surfaces have been created. The model has been completed with materials and textures acquired with a digital camera.



Figure 2 The hot mixer used as test case to validate the virtual prototyping approach proposed in this paper.

4.1. Design and simulation of the product behaviour

The realization of an interactive virtual prototype implies the formalization of the product behaviour in a simulator able to process the user inputs in order to determine how the status of the product changes.

The behaviour model of the product has been realized in Simulink using some of the several toolboxes available in this software package. For our purpose we did not need to schematize all the functions of the product, but only the ones that have a direct correlation to the product interface. The Simulink model we used in the test case can be considered as the first one that the engineers realize in a top-down design process.

The logics of the product behaviour, for which concerns the user interface, could be schematized us-

ing a Finite State Machine (FSM) model. Simulink supports the creation of FSM through the StateFlow toolbox that combines hierarchical state-machine diagrams (as the ones introduced by Statecharts [18]) with traditional flowchart diagrams.

In Figure 3 the complete Simulink model of the product is shown. The FSM, labelled as “Logic Unit”, is located in the left part of the picture. The FSM has two kinds of input: the first one, on its left, is related to the knob that controls the speed of the motor. It consists in two arrays that specify the speed of the motor for each angle of the knob rotation. The other input, on the upper part of the FSM block, is a bus that transmits all the data received from Virtools through the socket connection managed by the orange block.

The user actions are processed by Virtools that collects all the user generated events (button pressed, knob rotated, cover opened/closed) and sends them to Simulink. It receives all these data through the “Virtools socket connection” block. Also the clock signal is transmitted because it is used to synchronize the two applications. In Figure 3 the interaction between the user and the product has been explicated indicating where the user generated events are transmitted to the FMS.

The “Logic Unit” and the “Virtools socket connection” are linked in a loop, because they mutually exchange data at each iteration of the simulation. Since Simulink is not able to solve an algebraic loop, we have used a memory block that breaks this loop introducing a delay of one time step. The memory block, in fact, outputs its input from the previous time step, applying a one integration step sample-and-hold to its input signal.

The output of the FSM are six: the first value is the desired motor speed set by the user with the knob, four values (*md*, *mu*, *sd*, *su*) are the four digits on the LCD, *led_temp* is a vector that contains the state (on/off) of each temperature LED.

The desired motor speed is processed by a block that defines the behaviour of the electric motor and the relative control circuit using the SimScape and the SimElectronics toolboxes. The model contained in the motor simulation block is shown in Figure 4. The Universal Motor block has been used to simulate the voltage controlled AC motor. The control is realized through a variable resistor set by a closed loop feedback circuit with a PID (proportional-integral-derivative) controller block. The three constants in

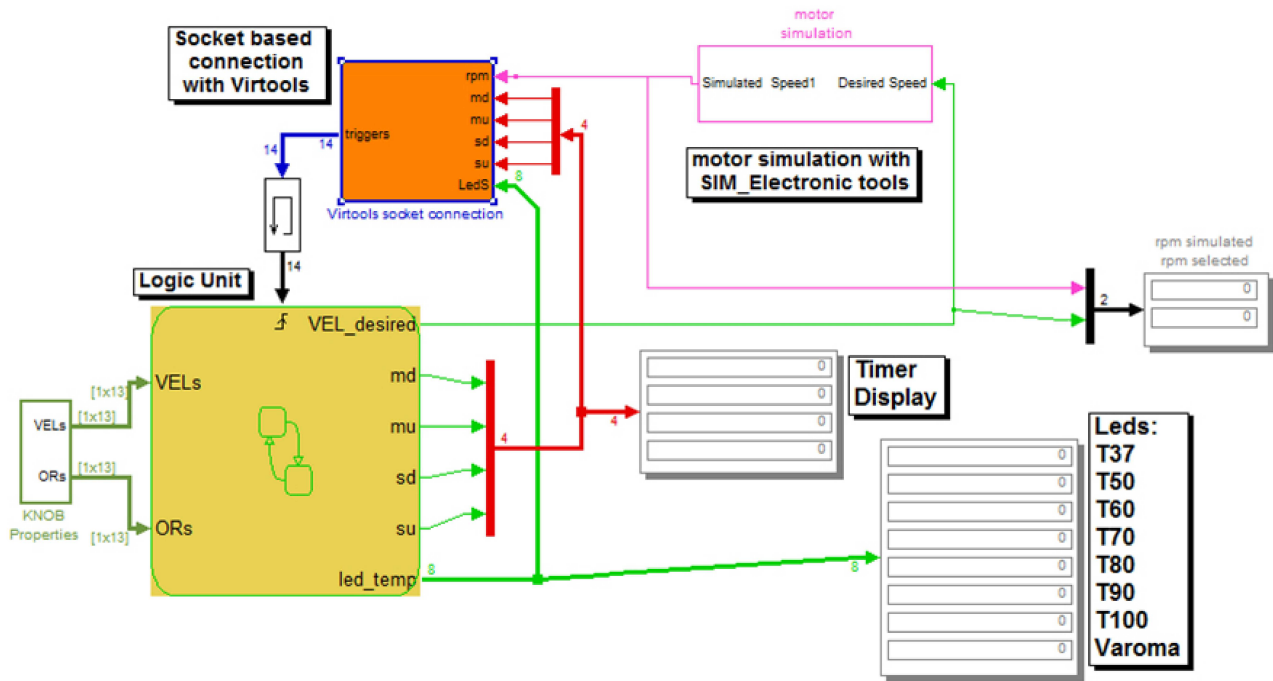


Figure 3 Simulink model of the product.

the PID controller (the gains of the proportional, integral and derivative terms) have been manually tuned to correct the error between the measured process variable (i.e.: the motor speed) and a desired setpoint that is, in this case, the speed set by the user through the knob.

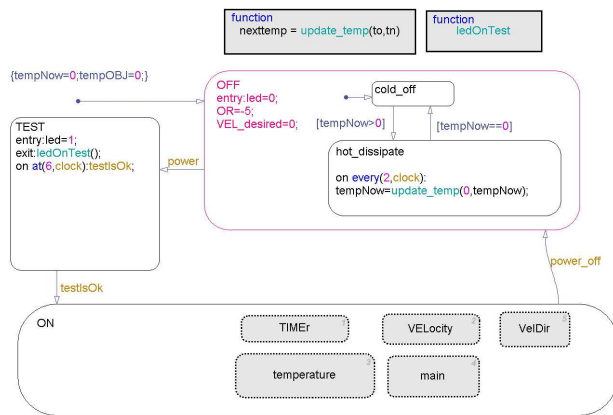


Figure 4 The Finite State Machine that reproduces the behaviour of the logic unit in the test case.

In Figure 5 the main FSM model of the logic unit is shown. This FSM has three states: *OFF*, *ON*, *TEST*. The default transition is characterized by the arrow present in the upper left corner of the image. It sets two environment variables to their initial states and activates the state *OFF*, that is the initial state of the

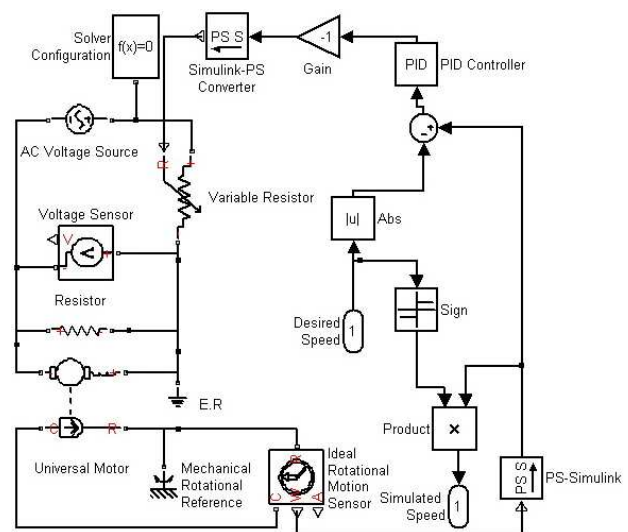


Figure 5 Simulink model of the electric motor and the relative control circuit.

product when the simulation starts. The *power* event, generated when the user presses the stand-by button, changes the state from *OFF* to *TEST* and from *ON* to *OFF*. The *TEST* state performs a test on the product. During the test, all the temperature LEDs are turned on and all the digits and the icons of the LCD appear, as shown in Figure 2. When the test is finished the current state switches to *ON*.

The *OFF* state is characterized by a subchart that

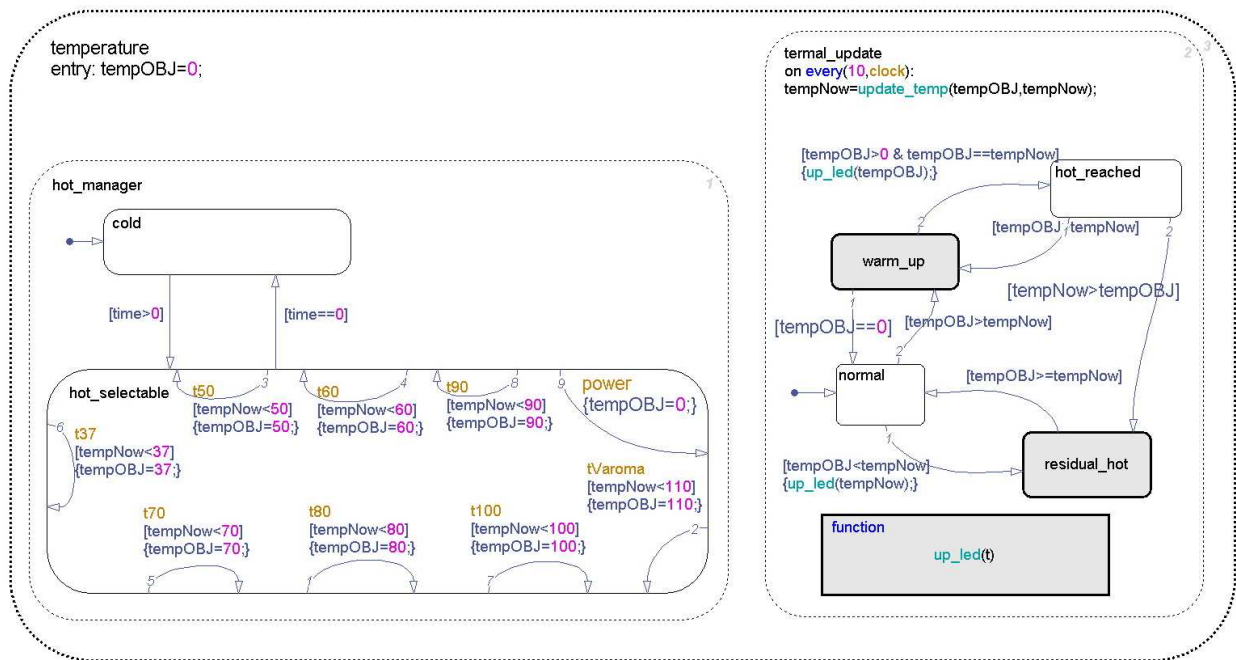


Figure 6 The temperature subchart.

contains two states: *cold_off* and *hot_dissipate*: the former is activated when the machine is cold, the latter when the machine has been switched off but it has not completely dissipated the heat. This second state manages the status of the temperature LEDs and manages the cooling of the machine. When the machine is cold it activates the *cold_off* state. The *ON* state is composed by four subcharts: *main*, *timer*, *velocity*, and *temperature*. The *main* subchart manages the completion of the job taking into account the time remaining before the job completion and any alteration of the motor speed or the timer.

The *timer* subchart catches and manages the user generated events related to the timer. In particular, when the user presses the “+” button associated to the clock icon, the timer is increased of a quantity that progressively changes: if the timer is between zero and one minute the increment is 1 second; if the timer is between one and ten minutes the increment is thirty seconds; after ten minutes the increment is one minute. The *timer* subchart controls also if the user reaches the maximum value for the timer (i.e.: sixty minutes) and calculates the value of each digit on the display. The *velocity* subchart manages the events generated by the user when he/she rotates the knob to set the rotational speed of the tool inside the bowl. This chart reads the values contained in the two arrays that for each angle of the knob rotation specify the speed of the motor.

The *temperature* subchart, shown in Figure 6, controls the part of the behaviour related to the hot working. In particular this chart defines how the temperature reaches the value set by the user, controls if the user has set the timer (otherwise the temperature cannot be changed) and manages the state of the LED. When the user sets a temperature the corresponding LED starts to flash indicating the user choice. While the temperature is increasing, the LEDs are progressively turned on, indicating the temperature reached inside the bowl.

4.2. Creation of the virtual environment

The design of the behaviour model of the product, described in the previous section, usually represents one of the first steps in the design process of the control system for an electric appliance. With our approach, even this preliminary model can be used to evaluate the functions and the usability of the product interface. In order to test with the user the previously described behaviour model, a digital mock-up of the product is needed. As previously mentioned, the geometries of the product have been reconstructed starting from an existing product. These geometries, completed with the materials and the textures, have been imported in Virtools. After that, the connection with the Simulink model has to be created. This activity consists of some tasks:

1. Define the state variables that have to be connected to the Simulink model. The present test-case has six state variables that are the output of the FSM described in the previous section.
2. Create a *socket receive* BB able to process the data stream sent by the Simulink model
3. Connect each output of the *socket receive* BB to its relative state variable.
4. Identify all the possible user generated events and catch them through a *Switch on Message* BB. This BB sends a message to the Simulink model each time that the user generates an event.

Figure 7 shows how the two BBs (*socket receive* and *BBSendMKey*) are used to create the connection between the Virtools model and the Simulink model.

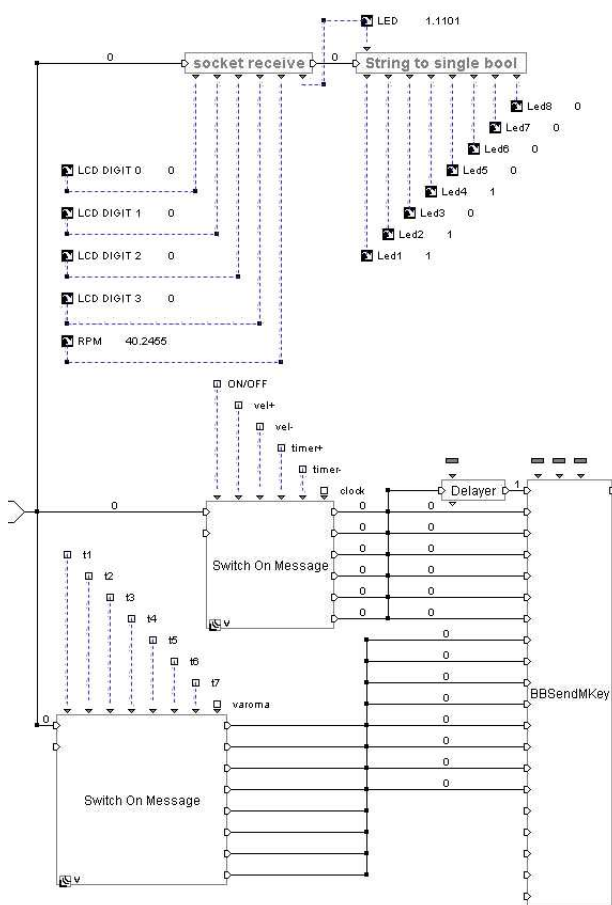


Figure 7 The Virtools schematic that manages the connection with Simulink.

The user generated events received by the *Switch On Message* BB may be *button pressed* and *knob rotated*. In the implementation of this test case the user interacts with the prototype using a mouse. Then, the *button press* event is easily retrieved as a pick event

on the geometries of the button. In a more complicated set-up based on a data-glove [12, 18] the button press may be identified as the collision between the geometry of the finger and the geometry of the button. The knob rotation is a bit more complicated. In [11] a technique to recognize the user gesture for knob rotation is described. The technique is related to a VR set-up based on a data-glove, but it has been easily adapted to the mouse interaction.



Figure 8 The display of the product is made up of nine rectangular patches with different textures that are changed at run time by the simulator.

The display is simulated by defining, on the virtual model of the product, a series of rectangular patches on which different textures are mapped; such textures are changed at run-time according to the values of the different variables that define the state of the product (Figure 8). The display is divided in nine parts: four are for the digits used to display time and temperature, one for the colon between the digits, four are used to show the status of some functions (timer, weighing, kneading, counter-clockwise rotation). The textures representing the ten digits are stored in a vector and are assigned to the four slots according to the messages received by the Simulink connection.

Figure 9 is a screenshot taken while the simulation is running. In the upper part of the image the Virtools window shows the interface with the timer at 00:28; the temperature is currently at 50 °C (yellow LED) and it has to reach 70 °C (orange flashing LED); the speed is set to 5. In the lower part of the image the simulation running in Simulink is illustrated. As it is apparent, the state variables reflect the value of temperature, time and speed seen on the Virtools model.

The Virtools model can be visualized using any virtual or Augmented Reality (AR) environment that the software supports. For example, in Figure 10, the product has been visualized in an augmented environment using a video see-through head mounted display and the ARToolkit Library for marker based tracking. This simple AR set-up allows us to evalu-

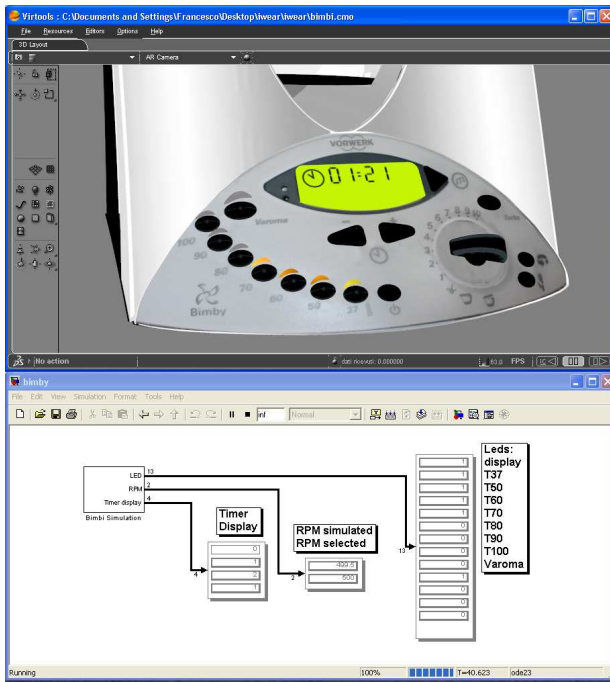


Figure 9 The simulation of the hot mixer running in Virtools (upper part of the image) and Simulink (lower part).

ate the visualization technique. From the first tests the augmented reality display seems to be better accepted by the user if compared with the immersive VR set-up. But more formal user studies are needed to establish the best visualization technique for our goals. As far as interaction is concerned, we are currently using the mouse to click on the buttons and rotate the knobs, but we are planning to test this approach with haptic equipment.

5. DISCUSSIONS

The implementation of the test case puts in evidence that the proposed approach is immediately utilizable in the design of several products where the user interface is made by electronics elements like buttons, knobs, LEDs and displays. In these cases it is sufficient to model a FSM that represents the logic of the user interface and to create connections between the simulation and the visualization environment. Even if the behaviour models of the product are more complex and detailed, this does not appear as a limitation for the proposed approach. The complexity of the model is only limited by the need to have an interactive simulation. In other words, the simulator has to be able to compute the state of the product at 25 Hz. This is usually not a problem if the behaviour is formalized through a lumped component model.

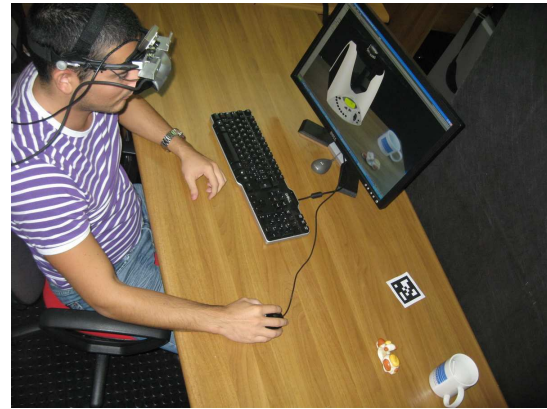


Figure 10 The augmented reality set-up, used to test the proposed approach, is based on a video see-through head mounted display and AR-Toolkit for marker based tracking.

Anyway the Simulink model can also be connected to other specific simulation software like FEM solvers for structural studies or MBS for dynamic analyses. The connection between Simulink and other simulators is a well known functionality, often used in the mechatronics field. An interesting extension of the present research could be the test of the proposed approach on a complex mechatronics case study using the Simulink co-simulation feature. In this case the performance could be a critical point, and it could be useful to evaluate the advantage offered by our approach that, through the socket connection, allows to use two different machines to run the simulation and the visualization, thus achieving an increment of the performances.

Moreover, the other aspect that needs to be better investigated is the interaction with the virtual product. This issue is particularly important for two reasons. The first is that in a virtual prototyping environment it could be useful to evaluate also ergonomics aspects as reported in [7]. The other is that the presence of a more complex human-product interface, that is not only composed by elements like buttons, levers and knobs, may create several interaction problems. Considering the test-case as an example we should approach the problem on how the user can operate to remove the cap and the bowl. At the moment it is sufficient to click with the mouse on the handle of the bowl to remove both the cap and the bowl, but it is apparent that this cannot be considered a valid technique to evaluate the usability and the ergonomics of the product.

In previous work [11] we noticed that the absence

of tactile feedback complicates the user's interaction with the interface of the virtual product. For this reason we are working on a new system, based on a haptic device and an AR display. Thanks to the haptic device, the user may feel the contact with interface widgets like buttons and knobs. Moreover, with AR the virtual product may be displayed in a real context and the user may see his/her own hand touching the digital objects. This idea presents several challenges related to the implementation of the visuo-haptic AR setup. The rendering pipeline has to be modified in order to delete the haptic device from the scene and, at the same time, to correctly display the real hand avoiding any wrong occlusion by the virtual objects. The key element of this mixed reality paradigm is a computational approach for the camouflage of the haptic device, using an image-based model of the context scene. We believe that this mixed reality paradigm can increase the versatility and outreach of visuo-haptic mixed reality [13].

6. CONCLUSIONS AND FUTURE WORK

In this paper we have presented an innovative approach to virtual prototyping based on the idea that the CACE software, usually employed by control, electrics or electronics engineers, can be also used to simulate the product behaviour in VR. This approach overcomes the classical problems related to the implementation of the code inside the VR software to replicate the product behaviour. The test-case implemented and described puts in evidence that the main advantages of our approach are:

- The time needed to implement an interactive virtual prototyping application is reduced because there is no need of code writing to model the product behaviour.
- Any modification done on the behaviour models in the simulation environment is immediately testable in VR.
- We can ensure the reliability of the simulation because it is performed by the same solver and with the same behaviour models that engineers use to design and manufacture the product.

The test-case presented in the paper describes a typical implementation of the proposed approach. Since the model is a quite simple one, we have planned to develop more complex cases in order to put in evidence limits and drawbacks that, at the moment, could be not apparent.

We have also planned to develop a formal methodology that can be used to implement the proposed

approach. We will describe all the steps and the tools needed to create an interactive virtual prototype starting from scratch. We have not still presented a methodology, because a stronger validation is needed before we will be able to formalize all the knowledge behind our approach. Other advancements of this research will concern the hardware set-up as described in the previous section.

REFERENCES

- [1] Acal, A.P., and Lobera, A.S., (2007), "Virtual reality simulation applied to a numerical control milling machine", *International Journal of Interactive Design and Manufacturing*, Vol. 1-3 pp. 143-154.
- [2] Antonya, C., and Talaba, D., (2007), "Design evaluation and modification of mechanical systems in virtual environments", *Virtual Reality*, Vol. 11-4, pp. 275-285.
- [3] Aoyama, H., and Kimishima, Y., (2009), "Mixed reality system for evaluating designability and operability of information appliances", *International Journal of Interactive Design and Manufacturing*, Vol.3, pp. 157-164.
- [4] Bao, J.S., Jin, Y., Gu, M.Q., Yan, J.Q., Ma, D.Z., (2002), "Immersive virtual product development", *Journal of Materials Processing Technology*, Volume 129, Number 1, 11 October 2002, pp. 592-596.
- [5] Barbieri, L., Bruno, F., Caruso, F., and Muzzupappa, M., (2008), "Innovative integration techniques between Virtual Reality systems and CAx tools", *The International Journal of Advanced Manufacturing Technology*, Vol. 38-11/12, pp. 1085-1097.
- [6] Bergamasco, M., Perotti, S., Avizzano, C.A., Angerilli, M., Carrozzino, M., Facenza, G., and Frisoli, A., (2005), "Fork-lift truck simulator for training in industrial environment", *Proceedings of Virtual Concept 2005*, DOI: 10.1109/ETFA.2005.1612593.
- [7] Bordegoni, M., Cugini, U., Caruso, G., and Polistena, S., (2009), "Mixed prototyping for product assessment: a reference framework", *International Journal on Interactive Design and Manufacturing*, Vol. 3-3, pp.177-187.
- [8] Bruno, F., Caruso, F., Falbo, L., and Muzzupappa, M., (2007), "A co-simulation based design methodology for mechatronic products", *Proceedings of ICED07 - International Conference on Engineering Design*, Paris.

- [9] Bruno, F., Caruso, F., Li, K., Milite, A., and Muzzupappa, M., (2009), "Dynamic simulation of virtual prototypes in immersive environment", *The International Journal of Advanced Manufacturing Technology*, Vol. 43-5/6, pp. 620-630.
- [10] Bruno, F., Caruso, F., Muzzupappa, M., and Stork, A., (2007), "An experimental environment for the runtime communication among different solvers and visualisation modules", *Proceedings of 19th European Modeling and Simulation Symposium*, Genova.
- [11] Bruno, F., Mattanò, R.M., Muzzupappa, M., and Pina, M., (2006), "A new approach to participatory design: usability tests in virtual environment", *Research in Interactive Design*. Vol. 1. Eds. Fischer X., Coutellier D., Cap. X, New York, NY: Springer-Verlag, pp. 80-90.
- [12] Bruno, F., Mattanò, R.M., Muzzupappa, M., and Pina, M., (2007), "Design for usability in virtual environment", In *Proceedings of ICED07 - International Conference on Engineering Design*, Paris.
- [13] Cosco, F.I., Garre, C., Bruno, F., Muzzupappa, M., and Otaduy, M.A., (2009), "Augmented Touch without Visual Obtrusion", *Proceedings of International Symposium on Mixed and Augmented Reality - ISMAR*, Orlando, Florida.
- [14] Cuadrado, J., Gonzalez, M., Gutierrez, R., and Naya, M.A., (2004), "Real time MBS formulations: towards virtual engineering", *Product engineering, eco-design technologies and green energies*, pp 253–272, Springer, Heidelberg.
- [15] Eberhard, P., and Li, Z., (2006), "Virtual reality simulation of multibody systems", In *Proceedings of EUROMECH colloquium 476*, Ferrol (Spain).
- [16] Gökdere, L.U., Benlyazid, K., Dougal, R.A., Santi, E., and Brice, C.W., (2002), "A virtual prototype for a hybrid electric vehicle", *Mechatronics*, Volume 12, Number 4, pp. 575–593.
- [17] Grantcharov, T.P., (2006), "Virtual reality simulation in training and assessment", *European Clinics in Obstetrics and Gynaecology*, Vol. 2-4, pp. 197-200.
- [18] Harel, D., (1987), "Statecharts: a visual formalism for complex system", *Science of Computer Programming*, Vol. 8, pp. 231-274.
- [19] Jimeno, A., and Puerta, A., (2007), "State of the art of the virtual reality applied to design and manufacturing processes", *The International Journal of Advanced Manufacturing Technology*, Vol. 33-9/10, pp. 866-874.
- [20] Park, H., Moon, H.C., Lee, J.Y., (2009), "Tangible augmented prototyping of digital handheld products", *Computers in Industry*, Vol. 60-2, pp. 114-125.
- [21] Park, H., Son, J.-S., and Lee, K.-H., (2008), "Design evaluation of digital consumer products using virtual reality-based functional behaviour simulation", *Journal of Engineering Design*, Vol. 19-4, pp. 359-375.
- [22] Sánchez, J., Esquembre, F., Martín, C., Dormido, S., Dormido-Canto, S., Canto, R.D., Pastor, R., and Urquía, A., (2005), "Easy java simulations: An open-source tool to develop interactive virtual laboratories using MATLAB/Simulink", *International Journal of Engineering Education*, 21(5):798-813.
- [23] Sinha, R., Paredis, C., and Koshla, P., (2000), "Integration of Mechanical CAD and Behavioural Modeling", *IEEE/ACM International Workshop on Behavioural Modeling and Simulation (BMAS'00)*.