

# Matlab Virtual Reality Simulations for optimizations and rapid prototyping of flexible lines systems

VAMVU PETRE, BARBU CAMELIA, POP MARIA

Department of Automation, Computers, Electrical Engineering and Energetics

University of Petrosani

Str. Universitatii, Nr.20, Petrosani, Hunedoara

ROMANIA

petre\_vamvu@yahoo.com, tabacarucamelia@yahoo.com, mariapop@yahoo.com

*Abstract:* - In this paper a new approach for simulation of Computer Integrated Manufacturing Systems is considered and developed in order to deliver rapid prototyping and to optimize the process. This approach is based on the possibilities of the Matlab to simulate complex mathematical models and also to integrate simulation procedures into design using the virtual reality environment. In this way we can fully simulate mathematical and visual complex systems like Computer Integrated Manufacturing solutions. This approach will allow better and faster design of automation solutions.

*Key-Words:* - robot mathematical model, CIM modeling, virtual simulation, computer integrating manufacturing, flexible lines systems, rapid prototyping

## 1 Introduction

Due to increasing demand on fast delivery of components, technologies, manufacturing plants and facilities, new methods for prototyping and development have to be investigated and implemented in order to satisfy the time, precision, cost and profitability of the new systems and products. One solution would be the introduction of a phase of visual simulation between the design and implementation [5]. The proposed solution resulted from the design can be validated by a simulation in a visual environment, which will produce a refined design solution where all the components will interact perfect [2]. This simulation can verify the mathematic models of different components and also offer an accurate visual representation of the interaction between all of the components of the proposed installation or plant. This method of a visual representation of different elements and the entire process in a real life like environment can produce early modifications and changes in the structure of the installation to produce optimal results [4]. Also this method can validate the individual elements choose for the implementation from the point of view of dimensional parameters, speed parameters, matching sizes and others.

## 2 Problem Formulation

Let's consider a CIM automation solution having 3 robots, 2 conveyers, a visual inspection system and a machining unit used for flexible manufacturing.

This automation solution is intended to produce different products and have to comply with an increased degree of flexibility.

For decreasing the costs of the overall solution, the idea is to use components with the best price/performance ratio. That means utilization of components from different producers and the integration of them into a unified system.

The components of the automation solution are a 6 degree of mobility robot arm, a SCARA type robot arm, a Cartesian robot, 2 conveyers, programmable logic controllers, inverters, three phase squirrel cage AC motors and a video reconnaissance system.

The 6 degree of mobility robot arm will satisfy a dual role. The first task will be to provide raw pieces for the machining unit from the magazine and also to retrieve the processed pieces from the machining unit and transferring them on the processed pieces conveyer belt [1].

The secondary role, and also optional, will be to feed the first conveyer belt with pieces destined for an optional assembly phase with the pieces processed by the machining unit.

The SCARA robot will perform the task of transferring or/and assembly the pieces from the first conveyer belt to the machined pieces conveyer belt. The assembly operation will take place on a reinforced part of the second conveyer belt specially designed for that purpose. The SCARA robot is connected to a visual reconnaissance system which performs two functions. The first function is to determine the position on X and Y axes and also the

rotational vector of the piece on the conveyer belt so the robot arm can reach and retrieve the piece in the gripper in a manner that will facilitate assembly operation.

The second function is to verify the dimensional parameters of the pieces so mismatching events can not appear in the assembly phase.

The Cartesian robot will perform the function of palletizing the finalized pieces or assemblies, retrieving the processed pieces or finalized assemblies from the assembly station and disposing them to the receiving pallet.

The machining unit is custom made from three Flip X series axes coordinated by the motion controller through the special Flip X series controller RDX. The cutting tool is a PAC mill attached to the vertical axis of the machining unit.

The main problem of this heterogeneous system is the compatibility between the different parts of the systems. To solve those problems from the prototyping stage we can create the virtual environment models of each and every part and then implement those virtual models into a simulations environment.

Choosing the solution to use the Matlab Virtual Environment we can simulate not only the physical interactions between different parts but also the interactions between mathematical models that describes the function performed by each part.

The virtual model of the CIM installation can comprise the mathematical models for each of the individual components but is not mandatory.

If the mathematical models of the components are included in the virtual model it is possible to determine the minimum requirements for the components that have to be satisfied, in order to achieve the best performance of the entire system [3].

$$p_{10} = -\cos \theta_1 \cdot \sin(\theta_2 + \theta_3) \cos \theta_4 \sin \theta_5 \cdot (d_5 + d_6) - \sin \theta_1 \sin \theta_4 \sin \theta_5 (d_5 + d_6) + \cos \theta_1 \cos(\theta_2 + \theta_3) \cos \theta_5 (d_5 + d_6) + \cos \theta_1 \cos(\theta_2 + \theta_3)(d_3 + d_4) + \cos \theta_1 \sin \theta_2 \cdot d_2 \quad (2)$$

$$p_{20} = -\sin \theta_1 \cdot \sin(\theta_2 + \theta_3) \cos \theta_4 \sin \theta_5 (d_5 + d_6) + \cos \theta_1 \sin \theta_4 \sin \theta_5 (d_5 + d_6) + \sin \theta_1 \cos(\theta_2 + \theta_3) \cos \theta_5 (d_5 + d_6) + \sin \theta_1 \cos(\theta_2 + \theta_3)(d_3 + d_4) + \sin \theta_1 \sin \theta_2 \cdot d_2 \quad (3)$$

$$p_{30} = -\cos(\theta_2 + \theta_3) \cos \theta_4 \sin \theta_5 (d_5 + d_6) - \sin(\theta_2 + \theta_3) \cos \theta_5 (d_5 + d_6) + \sin(\theta_2 + \theta_3)(d_3 + d_4) + \cos \theta_2 \cdot d_2 + d_1 \quad (4)$$

Starting from this data we build the virtual model of the robot which presents the same movement possibilities and the same dimensions like the real version. The building of the virtual model is a for steps process. The first step is to build the individual

### 3 Problem Solution

The first step in the simulation process is to describe the structure and the models of the elements which will be simulated.

The following step is to realize the virtual model for each of the elements and afterwards to integrate all the discreet virtual models into a unified virtual reality model. This final model will comprise all the elements of the CIM solution and will be used to validate the positions of different equipments, to validate the workflow of the installation, to determine the speed requirements of different elements or even to study the solution for a new project implementation.

#### 3.1 Simulation of the robotic components

First was determined the kinematic diagram based on that the Hartenberg-Denavit method and the movement matrix. After the mathematical model was completed the virtual prototype was constructed and the mathematical models were integrated in the simulation environment.

First robot to be simulated with six degree of mobility is the articulated robotic arm.

Based on the procedures presented above the movement matrix was constructed.

$$T_{0,6} = \begin{bmatrix} n_{10} & o_{10} & a_{10} & p_{10} \\ n_{20} & o_{20} & a_{20} & p_{20} \\ n_{30} & o_{30} & a_{30} & p_{30} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

The elements of the matrix were computed and the positions components for X axis, Y axis and Z axis,  $p_{10}$ ,  $p_{20}$ ,  $p_{30}$ , are presented bellow.

components which comprise the final object. The construction of the elements can be done in a specialized dedicated environment that can be found in the Matlab package and named "v-realm" or can be done using general purpose 3D design software.

The second step in building the virtual model is to set the movement hierarchy structure of the elements. Multiple elements can be connected together in structure where the movement of the “n-th” element is determined by the position of the “(n-1)-th” element in the kinematic chain and his own movement parameter.

The third step in creating the virtual model is to create the scene of the modelled assembly. The scene has to be also constructed in the “v-realm” environment, and in order to assure the accuracy of the simulation, all the elements have to be placed to specific coordinates of the scene referenced to world system coordinates of the “v-realm” environment.

The result of this third step is a virtual prototype, which has the same parameters and movement possibilities like the real one but lacks the interface connections with Matlab Simulink and also lacks the mathematical model that interact with Matlab environment.

The fourth step in the simulation process is to link the Matlab environment to the simulation scene. The link is done through the “visual sink” element find in the Matlab Virtual Reality ToolBox. This element connects the Matlab environment and the virtual scene created by the “v-realm” software. Also this element selects the parameters of the scene that will undertake modifications from the Matlab variables.

The variables used to control the virtual model can have predefined value or can be real time editable.

Another optional step can be added to the simulation process. This step implies utilizing a script to run the simulation fully automated in order to observe the entire process of a complex installation during a complete functional cycle. The script can be constructed by using a multidimensional Matlab variable which contained the movement values for all the parameters to specific moments in time.

Other important component of the CIM system is the SCARA Yamaha robot. This type of robot is often utilize in the pick and place type of applications. The advantage of the SCARA type robots is due to their unique arrangement of their joints which allow complex movements and very high speeds.

Having the information for the SCARA robot we build the mathematical model and then the virtual reality model in the Matlab Simulink using the “v-realm” tool for virtual reality scene.

The simulation scene once constructed is linked to the Matlab Simulink environment with VR-Sink tool, where the parameters of the scene which will be modified are chosen.

Adding the corresponding rotational and translation movement vectors to the model will complete the structural part of the design.

Also, to the model have to be added elements that will permit user to control the model and to simulate manual control. The complete simulation scene for the SCARA robot is presented in Fig.1.

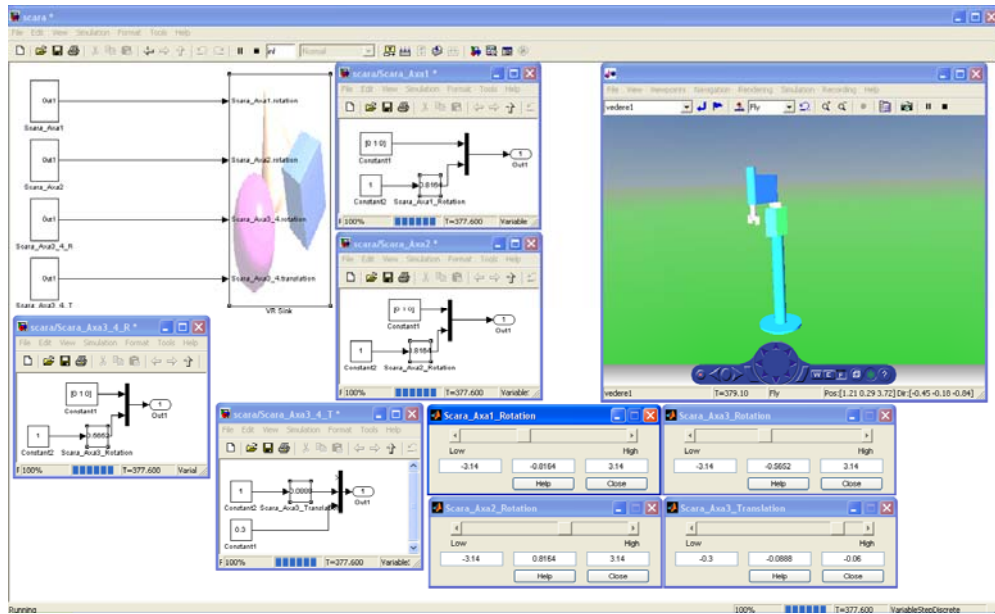


Fig.1 SCARA robot virtual reality real time simulation

The same process is used to build and integrate the virtual models of the Cartesian robot and

machining unit into the virtual simulation environment of Matlab Simulink.

### 3.2 System Simulation

After all the components of the CIM plant are individual simulated and the virtual reality model for each is constructed, the next step implies the connection process between all those components. First the 3D positions for each component have to be established. The coordinates have to be specified for each component of the virtual reality model. Due to hierarchical system that was used previously the coordinates have to be specified when the virtual reality scene is constructed just for the first element in the hierarchical chain.

The coordinates used are absolute coordinates of the virtual world. These coordinates play a role in setting the virtual reality scene by they do not influence the simulation process which is done using simulation vectors for each of the elements. The virtual reality simulation scene was constructed using the Matlab Simulink tool “v-realm”. The simulation scene is intended to be used in dual mode.

The first method is to manually manipulate the elements using altering parameter devices that Matlab Simulink offers.

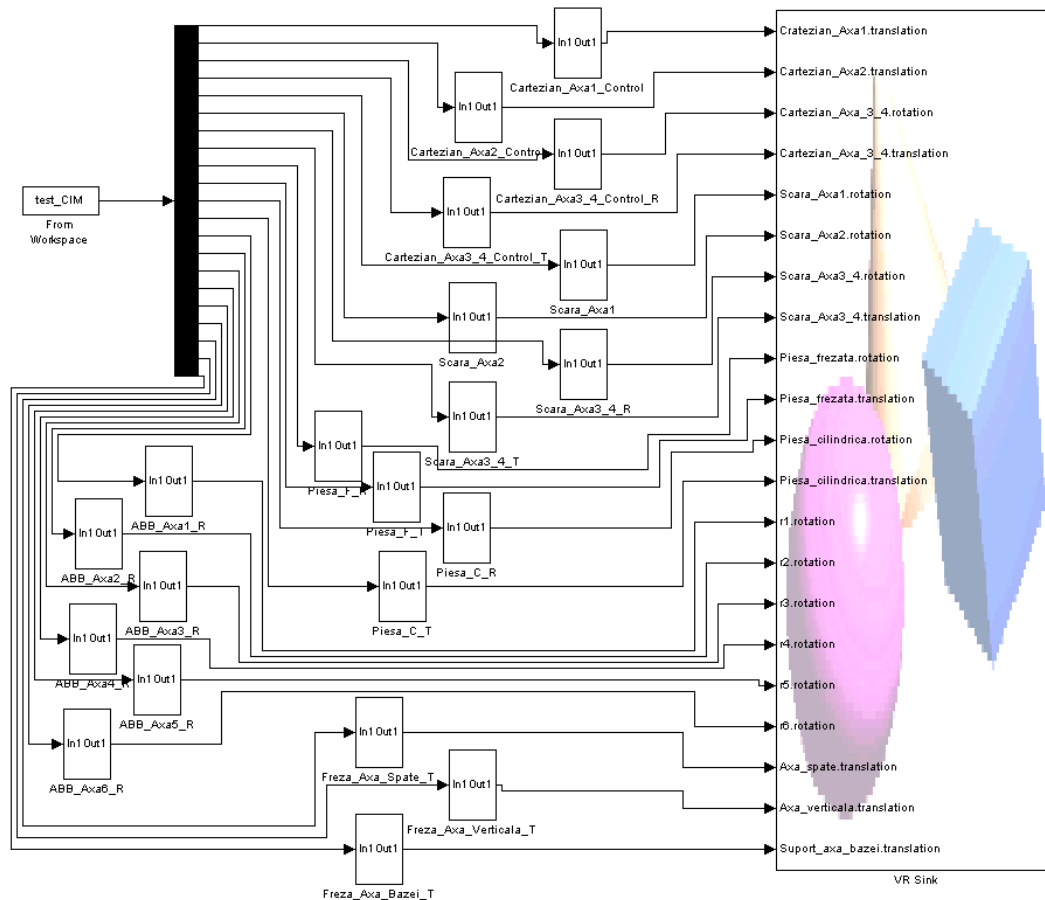


Fig.2 Matlab Simulink top level simulation design

This method give the user the possibility to modify the parameters so can verify the behavior of each component alone decoupled from the system.

This method will also provide the possibility for the user to study different solution for movement of different parts or components of the CIM installation.

The second way that the scene was intended to be used is to run following a script.

This second method applied to the virtual reality model offer the possibility to study very complex

operations that happened on the workflow of the system.

With the possibility to visually experience the complex workflow of the manufacturing system the designer can improve the design of different parts of the plant before the system is build.

Other benefit that occurs from this second method of virtual reality interaction is that the timing of different components and processes can be verified, corrected and optimized before the construction phase.

Because the manufacturing process is described like a chain of events that produce at the end a product, we can say that the manufacturing process is as strong as his weakest link

That means that if we want to verify if a manufacturing process can run at a certain speed, due to this method, the weakest link, slowest link, can be identified and replaced so the entire manufacturing system will produce better and faster from the first day.

In order to run the virtual reality simulation scene in an automated script we have to create a time driving component that will comprise the status of all the parameters of the plant at different moments in time. That component will provide for

the virtual reality simulation scene the events that produce the reaction of the environment at certain moments in time. The virtual reality model will react by modifying his parameters, calculating the mathematical expressions, and animating the scene accordingly.

This component will be a multidimensional Matlab workspace variable that will load the values from a linked Matlab file. The multidimensional variable will comprise on the first column the time vector.

For each position of the time vector a value it set or calculated for the same position in the following column vectors representing the parameters that will be modified in the simulation process.

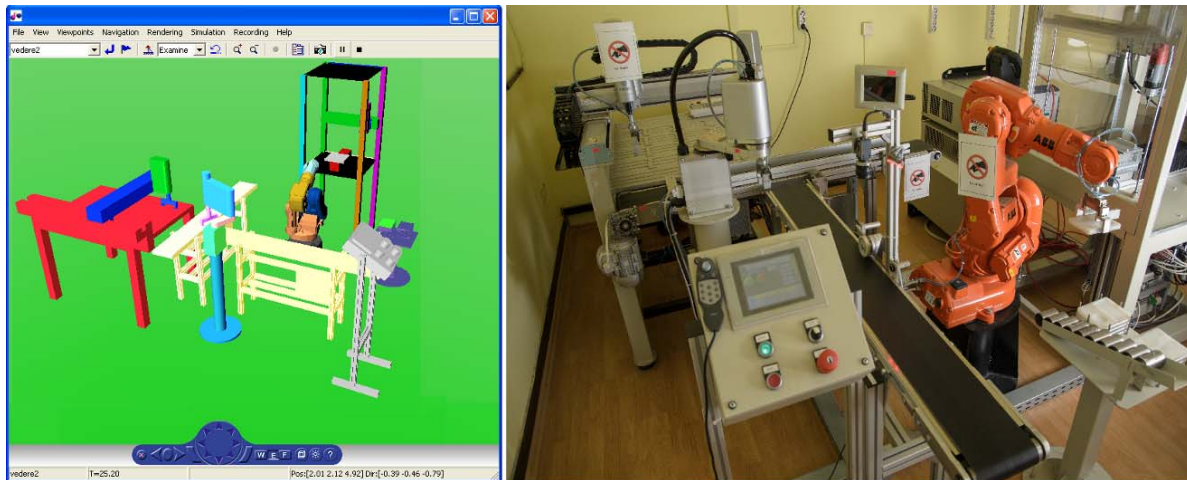


Fig.3 Complete CIM system autonomous model simulation vs. real installation

The simulation model links all the parameters of the virtual reality simulation scene with the multidimensional variable. The routing element is dispatching vectors values to each selected component. Each vector from the multidimensional variable is referenced to his unique simulation parameter.

The values from the multidimensional variable are transmitted by the routing element to the top level control blocks as can be seen in Fig.2.

Each of those blocks has inside a model which permits switching from manual to automated simulation mode for each of the component. These blocks deliver the control signals accordingly to the models inside to the Matlab Simulink linking tool which offer the connection between Matlab environment and the visual scene created with “v-realm” tool.

Also these blocks process the input value and if that values exceed the maximum allowed numbers the values are limited to maximal or minimal values. This is done in order to prevent wrong values to

perform actions on the virtual reality simulation scene.

In order to successfully animate the simulation model we have created a multidimensional Matlab workspace variable. This variable is linked to a .mat extension file which comprises all the data. After the multidimensional variable is created and saved, it will be necessary to be loaded into the workspace of the Matlab environment using the appropriate tools.

The simulation window in running mode for the entire CIM system is depicted below in Fig.3. The simulation is running completely autonomous based on the data provided by the multidimensional variable.

## 4 Conclusion

The Matlab virtual reality simulation used for the CIM installations can provide multiple advantages from the design stage to the secondary revision or refitting.

A main advantage is represented by the ability to test how different components match in terms of dimensional and speed parameters before the complete design process is finished and before the components are bought.

Another advantage provided by the virtual reality simulation using Matlab environment is the fact that mathematical models are not mandatory to develop the simulation scene, making the utilizations of the application much user friendly.

Using this system to model and simulate the function and the functioning of the CIM plant we can make verifications in terms of dimensional parameters, speed parameters, work flow, visual confirmation, before the building process had started.

This process of virtual reality simulation for a CIM plant can make the installation run at optimal parameters from the first time when the installation starts.

The future development implies using feedback information from the real installation and to monitor the functioning of the plant using the virtual reality model.

Further development is to combine the virtual reality model, Matlab environment and custom translator software programs in order to simulate and afterwards build the programs for different components in an automated fashion.

For further research the Matlab virtual reality model of the CIM is connected with the D-Space simulation and processing board in order to optimize the functionality and efficiency of the installation.

#### References:

- [1] Fawaz K., Merzouki R., Ould-Bouamama B., Model based real time monitoring for collision detection of an industrial robot, *Elsevier Mechatronics journal*, Vol. 19, 2009, pp.695-704
- [2] Bi Z.M, Wang L., Advances in 3D data acquisition and processing for industrial applications, *Robotics and Computer-Integrated Manufacturing*, Vol. 26, 2010, pp.403-413
- [3] Tomiyama T., Gu P., Jin Y., Lutters D., Kind Ch., Kimura F., Design methodologies: Industrial and educational applications, *CIRP Annals – Manufacturing Technology*, Vol 58, 2009, p543-565
- [4] Stark R., Israel J.H., Wöhler T., Towards hybrid modeling environments – Merging desktop-CAD and virtual reality-technologies, *CIRP Annals – Manufacturing Technology*, Vol 59, 2010, pp179-182
- [5] Ha S., Kim L., Park S., Jun C., Rho H., Virtual prototyping enhanced by a haptic interface, *CIRP Annals – Manufacturing Technology*, Vol 58, 2009, pp135-138
- [6] Leba M., Pop E., New Distributions Properties and Applications in Digital Control, Robotics and Automation, Proceedings of the 7<sup>th</sup> WSEAS International Conference on Signal Processing, Robotics and Automation (ISPRA '08), University of Cambridge, 2008, pp. 43-48