APPLICATION OF RIGID BODY DYNAMICS TO 3D PLANT SIMULATION

I.F. Ceruti[(a)], **G. Dal Maso**[(b)], **D. Rovere**[(c)]

[(a)] Technology Transfer System s.r.l. - TTS
[(b)] [(c)] University of Applied Science of Southern Switzerland - SUPSI

[(a)] ceruti@ttsnetwork.com; [(b)] giovanni.dalmaso@supsi.ch; [(c)] diego.rovere@supsi.ch

ABSTRACT

Nowadays the interest for simulation applied to both manufacturing systems and plants is rapidly growing. In this application the classic approach shows its limitation due to the inability to manage objects whose position is the result of interaction with other environment bodies.

The will to overcome these limitations without losing any of the benefits acquired with the introduction of simulation approach, leads to the introduction of rigid body dynamics into the simulation.

The limitations of the classic approach, a description of the novel approach and the results obtained on a real CNC system are the bricks composing this paper.

Keywords: manufacturing simulation, 3D simulation, physical simulation, CNC driven simulation

1. INTRODUCTION

The importance of computer animated simulations is ever increasing both in the design and in the production phase of manufacturing systems.

A comprehensive simulation, when applied to an entire manufacturing plant, must be capable to represent various aspects of the reality. Most common features, found in current 3D simulation engines, are direct-inverse kinematics solution and collision detection. More advanced features cover other process aspects like material removal and sensors simulation.

This kind of simulation is based on the assumption that relevant body positions are known as a result of defined motion paths, influenced only by kinematic relations (the object position is an input state). As mentioned, with this approach only the kinematics relationships are taken into account and no physical constraints like contacts or gravity influence the objects' position calculation.

This represents a limit especially in presence of free bodies whose motion is not governed by an analytic law but depends on the interaction with other geometries and on physical properties like friction. In kinematics simulation the problem is solved animating the part along theoretical paths (e.g. gravity based feeder).

It is to be noted that current vision in Digital Representation of manufacturing process moves towards an "adherent to reality" representation, as highlighted in the "Manu*future* Strategic Research Agenda" (European Commission 2004). The classic simulation clearly fails under this point of view.

This paper presents a new approach meant to integrate the rigid body dynamics aspects into the plant simulation. The purpose is a paradigm shift where the previous "animated" models in which the system state is known "a priori" is substituted by a real-time physics simulation model where interactions between geometrical entities and physical constraints influence the time evolution of their position.

The benefits of our approach have been verified developing the model of an automated CNC drilling and sawing system. More advanced topics, such as the integration of the simulation with the CNC logic has been explored.

The paper outline will be the follow: after a presentation of classic computer aided 3D simulation approach, the novel rigid body dynamics approach is presented. Section 5 presents a conclusive analysis of the application to a real industrial system.

2. PREVIOUS WORK

Virtual Reality (VR) and simulation in the manufacturing lifecycle as high value adding tools for cost-effective and rapid creation, management and use of the Next Generation Factory have been presented in (Pedrazzoli 2007).

These tools can be introduced as decoupled modules at both Product design and Factory design level of the product/process life cycle. The research done towards such tools has shown that these are a powerful way to gain flexibility in CNC machines as described in (Mancini 2004).

Virtual Reality simulation, both applied at machine and plant level, is a recent research area: despite this it is based on known fields such as real time collision detection (Kockara 2007) and physically based object behaviour (Baraff 2003).

Various applications can be found in literature describing the benefits of using collision detection techniques for simulation purposes. Besides application to generic 3D environments (Bergen 2003), there has been an extensive research on application of collision detection to robotics (Steinbach 2006; Kuffner 2002; Okada 2006). Recent studies show application of such engines to wider manufacturing applications (Ceruti 2008).

Attempts to take advantages from a physically based simulation can be found in (Loock 2001). Together with collision detection, physics simulation (in terms of gravity effects on objects and physical based cable simulation) is used to validate assembly tasks procedures.

In (Glencross 2001) a more integrated and complete framework has been proposed to simulate the interaction of the objects in a VR environment.

Other authors (Carpin 2007; Greggio 2006; Garber 2002) use physical based simulation for the motion planning of rigid and articulated robots.

In such cases benefits come from the ability to compute better results if compared to traditional methods, when there is an interaction in a complex environment with moving objects.

3. CLASSIC SIMULATION APPROACH

Computer aided simulation applied to manufacturing plants combines the typical approach of a three dimensional viewer with the peculiar information used to represent mechanical data of a manufacturing layout.

The simulated environment mainly deals with mechanical components, production items and production support structures.

3.1. Scene-graph

The classic simulation approach is based upon a hierarchical data structure called scene-graph. This structure is a directed acyclic graph (DAG) in which nodes represent simulation entities and edges represent their positional relationships. In fact, at a higher abstraction level, objects taking part to the simulation can be considered as pure reference systems (XYZ-O) with an associated collection of properties.

In modular software architectures, engines taking part to the simulation (Pedrazzoli 2001) populate this collection storing their own customized data structures for their purposes. Examples of properties are meshes used by the 3D visualization engine, collision structures used by the collision detection engine and *voxels* data used by the real-time material removal simulation engine.

Relationships between nodes of the DAG represent relative geometrical displacements between objects' reference systems and can be expressed in terms of homogeneous transform matrices. Therefore, given two nodes i and j representing two simulation entities, the edge connecting them is fully described by the matrix

$$H_j^i = \begin{bmatrix} R_j^j & \boldsymbol{t}_j^i \\ \mathbf{0} & 1 \end{bmatrix} \qquad (1)$$

where R_j^i represents the rotation part of the transform and $\boldsymbol{t}_j^i$ the translational part.

In plant simulation these transform matrices can be controlled either *directly* or *indirectly*. The control is *direct* when the absolute object's reference system 3D position and rotation is set and then relative transform are re-calculated.

An *indirect* control is obtained using mechanical joint models (rigid, rotational, translational) which constraint the motion of the children relatively to the parent position according to some setup parameters (like rotation/translation axes) and the runtime value of their joint variables (rotation angle or translation). In this case transform matrices $H_j^i(\boldsymbol{x})$ become functions of a set of joint parameters.

3.2. Simulation loop for CNC driven system

In the classic approach, running a simulation actually means animating the scene-graph nodes modifying the current absolute position, orientation and states of the objects composing the environment model.

This animation can be directly controlled by a source capable to generate inputs for the simulation model. In a CNC-driven simulation, the CNC computes the values of the joint variables that are fed into the simulation as state inputs through a proper communication interface.

These inputs activate the typical simulation loop (Figure 1): each input modifies some environment variables, triggers the simulation engines and brings the simulation into a new state.

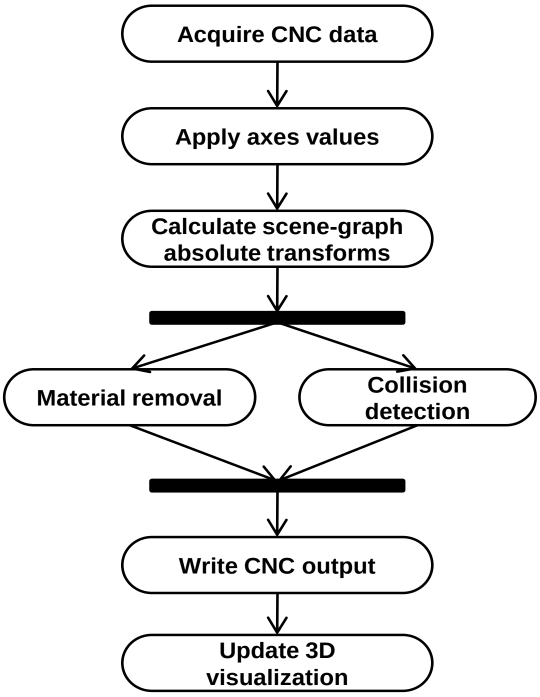

Figure 1 : Simulation loop for CNC driven system activity diagram.

Starting from the root node, walking through the scene-graph and taking into account the current type and value of the joints, the position and orientation of each node is updated.

Once the final state of each node is computed, simulation engines can perform their work accessing and modifying node's properties. As an example consider the real time material removal engine that checks for geometry intersections between tool nodes and modifiable objects, and the collision detection engine that checks for contact points between objects provided with collision structures (Ceruti 2008).

The results produced by simulation engines can be directed both internally (e.g. other engines) and to the CNC (as feedback data) closing the communication loop. The collision detection engine, for example, produces a list of colliding nodes that can be sent both to the 3D visualization engine which could highlight the objects for easy contact identification, and to the CNC software to produce a warning message for the user and possibly debug the CNC logic.

3.3. Benefits and limitations in plant simulation

The aforementioned simulation brings many benefits to the end user along the plant/machine lifecycle. In the design phase, testing CNC logics with a simulated machine instead of using a real prototype is cost saving since:

- the risk of costly machine breakings is totally eliminated
- motions can be tested at real feed rates (while normally low speeds are needed in order to prevent unforeseen collisions between parts of the equipments), thus reducing the testing time
- simulation can be run at accelerated time with batch processes enabling a “what if” analysis on the CNC logics and providing the means for optimization

Moreover the 3D visualization of the plant part motions is unparalleled when used during the debug phase of new complex sequences of operations, like tool change procedures can be.

3D simulation can be also used during the production phase of the plant. In fact the developed models can be exploited for remote monitoring applications, providing that most of the modern CNC devices support network communication protocols. In this case the cost saving is due to the fact that it is possible to perform remote diagnosis of breakdowns thus reducing (when not totally avoiding) the duration of technical personnel interventions.

This classic approach to manufacturing plants simulation has however some limits which are direct consequences of the fact that the motion animation of each part of the system is directly under the control of the input source. In fact, though it is easy to handle well known and predictable movements, like a mechanical axis positioning, it is almost impossible to manage motions resulting from objects interaction, like contact forces, friction, and so on.

A clear example of this limitation is represented by the handling of the piece to be manufactured: this is not constrained by any mechanical joint to the plant structures and its position in the system is not under the direct control of the CNC; instead it is the result of the interaction with other objects in the scene like handling devices (grippers) and transportation elements (roller ways, chains, magnets). Moreover, when pieces are queued on the feeders it becomes essential to model the interaction between free objects.

The fact that classic simulation fails to manage these common situations highlights the need for the new approach described in the following section.

4. RIGID BODY DYNAMICS APPROACH

The simulation of the motion of a rigid body is based on the motion of a particle. The rigid body can be seen as a system of particles where the distance between any two given entities remains constant in time regardless of external applied forces. A rigid body is thus non deformable but this limitation is not a problem as we are not interested to model material deformations.

4.1. Motion of a particle

If we denote $x(t)$ as the particle location in world space at time t, $v(t) = \dot{x}(t) = \frac{d}{dt}x(t)$ gives its velocity. We can thus define the state vector for the particle as:

$$Y(t) = \begin{pmatrix} x(t) \\ v(t) \end{pmatrix} \tag{2}$$

We want to know how the state of the particle changes over time, i.e. we want to calculate:

$$\frac{d}{dt}Y(t) = \frac{d}{dt}\begin{pmatrix} x(t) \\ v(t) \end{pmatrix} = \begin{pmatrix} v(t) \\ F/m \end{pmatrix} \tag{3}$$

where the relation $\frac{d}{dt}v(t) = F/m$ is derived from Newton’s second law. Thus we have to know the forces applied to the particle and its mass in order to solve the system.

4.2. Rigid body dynamics

Unlike mass particles, that have a finite mass but zero volume, rigid bodies have a considerable volume and have geometrical properties. At any instant the position of a point p_0, measured in body space, is given by:

$$p(t) = R(t)p_0 + x(t) \tag{4}$$

where $R(t)$ is the rotation and $x(t)$ is the translation of the body. To extend the definition of the state vector $Y(t)$ for a rigid body, we now need both $x(t)$ and $R(t)$. Similarly the motion of the body can be described by the linear momentum $P(t) = m\,v(t)$ and the angular momentum $L(t) = I\,\omega(t)$ as shown in (Shabana 2001).

To compute the evolution of the state $Y(t)$ it is no more sufficient to know the mass m, we also have to provide the values of the inertia tensor I. The solution is expressed in terms of the force $F(t)$ and of the torque $\tau(t)$ applied to the rigid body.

Let us consider a simple system composed of two rigid solid bodies. If we exclude the case of inter-penetration, when two bodies are in contact their motion is constrained and we can have two situations:

- *Colliding contact*: when the velocity relative to each other is non zero. To prevent inter-penetration we must have an instantaneous change of velocity vector.
- *Resting contact:* when the relative velocity is zero.

We enforce these non-penetration constraints by computing appropriate contact forces between contacting bodies and then applying them to each body.

The methods for resolving these constraints, as well as extending the computation to the case of n contacts, can be found in (Bender 2006). To model the response of real world object to contact we have to assign at least the following coefficients to the body:

- Restitution coefficient: $0 \leq \epsilon \leq 1$, takes into account how the velocity changes after a collision. $\epsilon = 0$ means no bouncing at all while $\epsilon = 1$ means completely elastic contact.
- Coulomb friction coefficient: $\mu \geq 0$. It is relative to the sliding motion of two surfaces in contact. If $\mu = 0$ there is no friction, while if $\mu = \infty$ the contact is considered to be sticky.

4.3. **Changes in the data model**

The physics based approach to simulation involves a significant change in the modelling phase. In fact data required in classic simulation is still needed and it must be integrated with the information related to the physics collisions geometry, mass properties and contact parameters. It is possible to summarize the main changes in the data modelling as follows.

4.3.1. Creating the collision geometries

All the objects taking part to a physics based simulation have to be provided with a collision geometry that is used by the physics engine in order to compute contacts details (e.g. contact points and normals).

These geometry representations are usually far more simplified than the ones used for visualization purposes. This simplification is a required phase to lower the computational complexity: drawing N models is $O(N)$ while collision detection is $O(N^2)$.

4.3.2. Assigning physical properties

Modelling a physical simulation entity requires the definition of material related properties like mass values and distribution, restitution and friction coefficients and joints max forces and torques.

This aspect of the physical modelling requires a particular care because it is not always easy to find the real values of these parameters. Whenever possible they are extracted directly from the documentation of the machines and equipments as maintenance manuals and CAD drawings. This usually happens for mass properties and joints forces, while friction and restitution coefficients are set to the values available in literature and then tuned according to the simulations results.

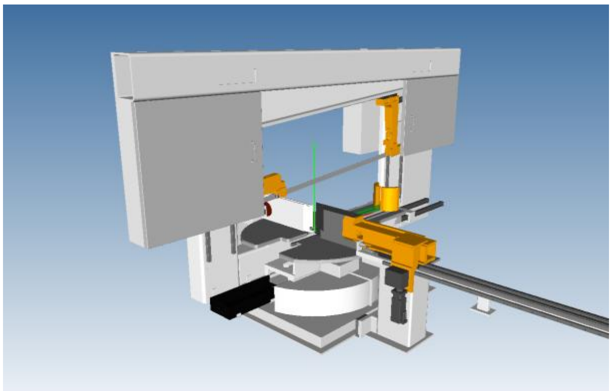

Figure 2 : A sawing machine visual model

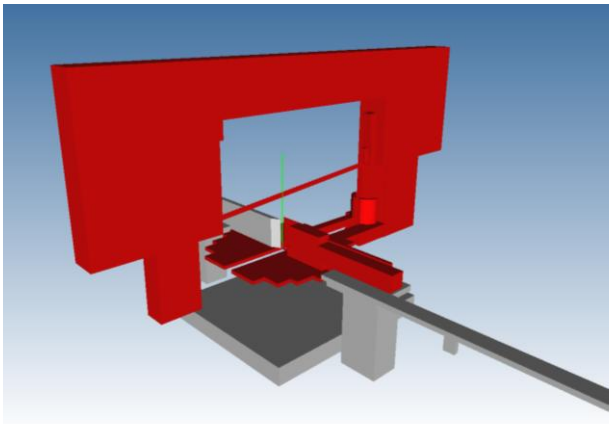

Figure 3 : Sawing machine physical model: a visual representation of its collision geometries

4.3.3. Tuning the simulation model

A big challenge in the modelling is represented by the simulation model tuning phase which is addressed to grant a high level of adherence to real system observed behaviour.

Since the model represents an approximation of the real system, it is often necessary to tweak the physical properties of the entities in relation to how they interact with each other in order to obtain meaningful results.

According to the target results it is always possible to increase the accuracy used to model some parts in order to locally improve the realism of the simulation while maintaining the quasi real-time responsiveness of the model.

4.4. **Changes in the simulation loop**

With the introduction of rigid body dynamics, the simulation loop described in section 3.2 has to be changed in order to take into account the physics engine presence.

The positional values coming from the CNC cannot be directly applied to joints; instead they are used to compute the velocity vectors of the bodies. These values are fed to the physics engine that computes the updated final state $Y(t)$ for all the objects in the model.

The translation and rotation part of the state is then applied to each node of the scene-graph and the simulation loop can continue with simulation engines triggering, like surface modification and collision detection.

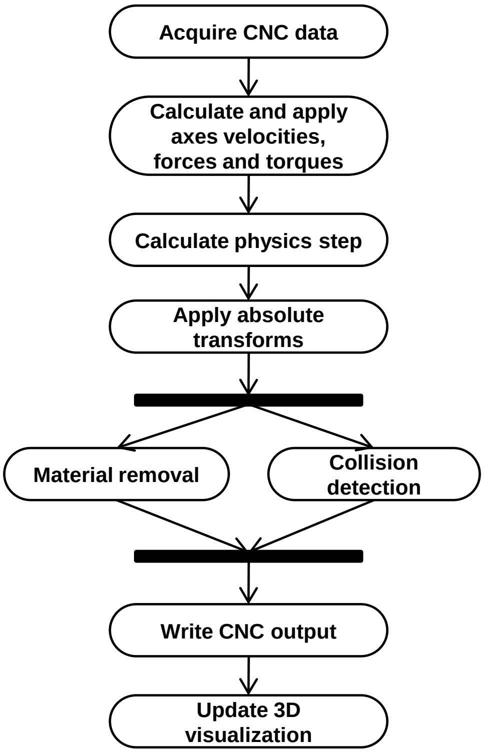

Figure 4 : Physic simulation loop for CNC driven system activity diagram

4.5. Benefits and limitations

The additional benefits of using physics based simulation are related to the possibility to manage the piece and to simulate a wide range of unpredictable behaviours. Moreover, the usage of a wrench based physics engine solver enables the simulation of complex closed links without the need to provide joints with closed form equations.

The limitations of the rigid body dynamics approach can be summarized as follows:

- *Increased modelling effort*: as section 4.3.1 shows, physics simulation needs simplified versions of the visual model geometries which are often replaced with a set of primitive volumes; this requires an extra modelling effort and time.
- *Risk of introducing model instabilities*: unbalanced physics properties (mass and forces) and constraints introduced while modelling the simulation environment can result in numerical instabilities of the physics step calculations.
- *Computational workload*: the introduction of the physics step (Figure 4) in the simulation loop requires a computational effort strictly related to the model complexity, which directly translates into higher memory and CPU requirements.

5. RESULTS

5.1. CNC drilling and sawing system

The case study layout is a heavy carpentry line for the manufacturing of steel beams and is composed by a drill unit and a saw unit connected by an automatic handling system.

The typical argument against having a drill and saw in tandem (side by side layout) has always been that if the drill is working, the saw is idle and vice versa. Even with this objection, the tandem approach has proven to be the ideal solution for many fabricators as the reduction in required plant space, the need for only one operator and the lower investment cost has proven to be the optimum solution for many facilities.

The CNC software engineers have developed a *multi tasking* solution that can be used in the majority of the applications to simultaneously drill and saw the profile. This multi tasking solution requires a more complex CNC logic to handle the sequence of operation needed to complete the tasks.

This layout is a suitable test case since the beam position is the result of complex geometrical interactions with both moving and steady elements of the handling system and not a CNC driven variable.

This is a typical case in which traditional simulation fails due to the high number of embedded logics required to model the behaviour of system. On the contrary, with the proposed approach only intuitive model parameters like friction coefficients, masses, forces and torques have been provided to the simulation. The obtained result is a simulation in which the CNC is reacting to unpredictable sensors signals, in the same way as it was connected to the real devices.

The interface between the application and CNC has been developed using a proprietary communication protocol based on TCP/IP.

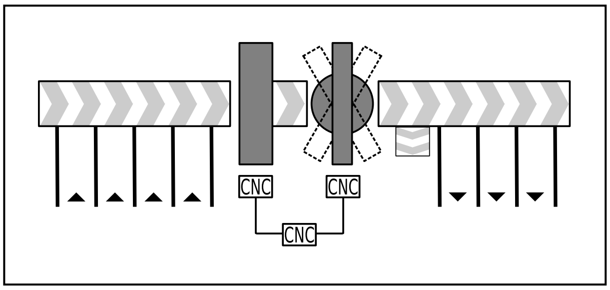

Figure 5 : The drilling and sawing system schema

5.2. Simulation Performances

The developed software has been based on the DDD Simulation Libraries (Ferrarini 2008) integrated with the ODE C++ physics library. Tests have been executed on a laptop equipped with an Intel® Core™ 2 Duo CPU T9300 - 2.50GHz, 3069 Mb RAM and a NVidia®

GeForce™ 8600M GT graphics card with 256 Mb of dedicated video memory.

The CNC used is a Mitrol® Minosse™ control unit for Ficep® lines integrating a PLC (IEC6-1131) at 2ms and 10ms and a machine axes control unit running both in asynchronous (Profile Position) and in synchronous mode (Interpolated Mode). Both 3D models of the line and CNC have been kindly provided by Ficep® S.p.A.

CPU, Memory and max FPS results have been collected using a beam and machines axes motion simulation. Model visual size is approximately 404K triangles. The physics calculations step has been set to 1ms, while the read/write interval from the CNC has been set to 10ms.

Table 1: CPU and Memory usage comparison between the two approaches (CPU usage is an estimated average value)

Simulation Results			
	CPU	RAM	FPS
Kinematics	15%	41 Mb	212
Physics	75%	53 Mb	33

The results show a significant impact of the computational workload of the physics engine on the CPU usage and a limited increment of occupied memory. Nevertheless, the resulting frame rate (FPS) value shows that the simulation remains interactive.

Figure 6 : Test system: laptop pc and CNC

REFERENCES

Baraff, D., Witkin, A., Anderson, J., Kass, M., 2003. Physical based modelling. *SIGGRAPH Course Notes*.

Bender, J., and Schmitt, A., 2006. Constraint-based collision and contact handling using impulses. *Proceedings of the 19th international conference on computer animation and social agents*, pp. 3-11, Geneva (Switzerland).

Bergen, G., 2003. Collision Detection. Interactive 3D Environments. *Interactive 3D Technology Series*, Morgan Kaufmann ed.

Carpin, S., Lewis, M, Wang, J., Balakirsky, S., Scrapper, C. 2007. USARSim: a Robot Simulator for Research and Education. *Proceedings of the IEEE 2007 International Conference on Robotics and Automation (ICRA)*, October.

Ceruti, I., Dal Maso, G., Rovere, D., Pedrazzoli, P., Boër, C.R., 2008. Real Time Collision Detection for Complex Simulations Based on Hybrid Multi Resolution Approximation of CAD Models. *The International Workshop on Modelling & Applied Simulation*, pp 225-232. September 17-19, Amantea, Italy

European Commission, MANUFUTURE – a vision for 2020, November 2004

Ferrarini, L., Veber, C., 2008. 3D graphic simulation of flexible manufacturing systems with Day Dream Daemon and 3DCreate. *Proceedings of 6th International IEEE Conference on Industrial Informatics (INDIN2008)*, pp. 1401-1406. July 13-16, Daejeon, Korea

Garber, M., 2002. Constraint-Based Motion Planning for Virtual Prototyping. *Proceedings of ACM Symposium on Solid Model and Applications*.

Glencross, M., and Howard, T., and Pettifer, S, 2001. Iota: An Approach to Physically-Based Modelling in Virtual Environment. *IEEE Virtual Reality Conference*; pp. 287-288.

Greggio, N., Silvestri, G., Antonello, S., Menegatti, E., Pagello, E., 2006. A 3D model of humanoid for usarsim simulator. *Proceedings of the First Workshop on Humanoid Soccer Robots*, Dec.

Kockara, S., Halic, T., Iqbal, K., Bayrak, C., Rowe, R., 2007. Collision detection: a survey. *IEEE International Conference on Systems, Man and Cybernetics*, pp 4046-4051

Kuffner, J., Nishiwaki, K., Kagami, S., Nishiwaki, K., Kagami, S., Kuniyoshi, Y., Inaba, M., Inoue, H., 2002. Self-Collision Detection and Prevention for Humanoid Robots. *Proceedings of the IEEE International Conference on Robotics and Automation*, pp 2265-2270

Loock, A. and Omer, E. S., and Stadtwald, I. 2001. A Virtual Environment for Interactive Assembly Simulation. Algorithms and Complexity Group.

Mancini, F., Lopez, M., Rinaldi, R., Boër, C.R., Pedrazzoli, P., Beccati, F., 2004. A new approach to programming flexibility for CNC machines. An example of application in woodworking machining. *Proceedings of the 37th CIRP Int. l Seminar on Manufacturing Systems (ISMS 2004)*, pp. 163-166, May 19-21, Budapest ,Hungary

Okada, K., Inaba, M., 2006. A Hybrid Approach to Practical Self Collision Detection System of Humanoid Robot. *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3952-3957. October 9-15, Beijing, China

Pedrazzoli, P., Bathelt, J., Chryssolouris, G., Rovere, D., Pappas, M., Boër, C.R., Constantinescu, C., Dépincé, P., Westkämper, E. 2007. High Value Adding Virtual Reality Tools for Networked Customer-Driven Factory. *Proceedings of the 4th International Conference on Digital Enterprise*

Technology, pp. 347-352. September 19-21, Bath, United Kingdom. Maropoulos, Paul (Ed.), CIRP u.a

Pedrazzoli, P., Sacco, M.; Jönsson, A., Boër, C.R. 2006. Virtual Factory Framework: key enabler for future manufacturing. *Digital enterprise technology: perspectives and future challenges*, pp 83-90. Springer, 1

Shabana, A., 2001. Computational Dynamics. 2nd ed. Wiley-Interscience.

Steinbach, K. , Kuffner, J., Asfour, T., Dillmann, R., 2006. Efficient Collision and Self-Collision Detection for Humanoids Based on Sphere Trees Hierarchies. *Proceedings of the 6th IEEE-RAS International Conference on Humanoid Robots*; pp. 560-566. December 4-6, Genova, Italy

AUTHORS BIOGRAPHY

Dr. Ing. **Ilario Francesco Ceruti** is a software development consultant for Technology Transfer System s.r.l. He received his degree in Computer Engineering from Politecnico di Milano in 2005. Since then and he has done software development for engineering research, participating to EU founded research projects in FP6 on advanced 3D simulation of manufacturing environments (KoBaS). Since 2008 he has followed EU founded research project in FP7 concerning the development of tools to support design of mass customization shoes (Dorothy). His interests include computational geometry, computer graphics, databases and software engineering, and he is currently consultant for the development of a NC processes result 3D visualization.

Dr. Ing. **Giovanni Dal Maso** is a software development consultant. His interests include simulation and computer graphics. He has done software development for engineering research in EU founded projects. He received his degree in Electronic Engineering from Politecnico di Milano. He is now a researcher at SUPSI-ICIMSI.

Diego Rovere graduated in Mechanical Engineering (Robotics and Automation) at Polytechnic of Milan in 2003 with a thesis on an Automated System for the model and cost analysis in foundry. Researcher at the National Research Council (CNR) of Italy where he was responsible of the Java development group for the 3D simulation of manufacturing environments in the EU KoBaS project. Consultant for the development and integration of 3D simulation systems with CNC based on CANOpen protocol and with CAM systems for the woodworking sector .Consultant for the 3D simulation of WaterJet cutting processes in collaboration with Water Jet Sweden and Polytechnic of Zurich. Consultant of the Polytechnic of Milan for design and development of a software system for the stability analysis of milling operations performed on magnetic clamping planes. Consultant for the development of a control software for magnetic working surfaces with embedded force sensors. Responsible of the Java development and integration with a CAN based Virtual Cockpit Hardware of a simulation system of construction machines and agriculture vehicles. Since June 2009 researcher at SUPSI-ICIMSI.