Proceedings of the Annual Symposium of the Institute of Solid Mechanics
and Session of the Commission of Acoustics, SISOM 2016 Bucharest 12-13 May

3D MODELING OF MOBILE ROBOTS IN VIRTUAL REALITY ENVIRONMENT USING BLENDER AND UNITY APPLICATIONS, INTEGRATED IN VIPRO PLATFORM

Ionel-Alexandru GAL[1], Luige VLADAREANU[1], Radu A. MUNTEANU[2], Oana CHENARU[3],
Gelu FLOREA[3], Catalin-Eugen SIMION[3]

[1]Institute of Solid Mechanics, Romanian Academy, Bucharest, Romania,
[2]Technical University of Cluj Napoca
[3]SIS SA

The aim of this paper is to present a way to bring a real life robot into the virtual environment. The increasingly demanding modern industry and robotics scientists and engineers, require new design of robots, faster and faster. To design and test new robots, virtual tools are easier and cheaper to use, even by average skilled personnel.
By using the proposed method, one can build virtual robots using specialized tools for 3D modeling and tools for real time control of the designed virtual robot.

1. INTRODUCTION

This paper presents one of many modules of the VIPRO Platform [1, 11, 14] (Versatile, Intelligent and Portable Platform for Robots) which handles the virtual simulations of robots. The VIPRO Platform was conceived as a complete tool for 3D modeling and simulation of mobile robots, but can also help develop planning strategies for robots along with image processing and adaptive intelligent control laws [10-11].

The original idea behind the VIPRO Platform is to create an environment but also a process to easily create and develop new robots designed to help and replace human interactions in disaster areas which can be hazardous to human life [12, 15]. For this, one needs to have a way of creating mechanical 3D models which can be used in a virtual environment for testing. This two-step process is the main core for simulating new robots using virtual tools [8, 17], which is also the VIPRO Platform process of bringing new ideas to the virtual testing bench.

The two-step process of simulating 3D robots, can be thought as a simple one, but in fact is nothing but simple. The reason is that not every tool is right for every job. For example, not every 3D simulation tool can process 3D models made with certain modeling tools. This can be a problem when your favorite 3D modeling tool can’t save the model to a format supported by the required virtual reality software [4-7]. In this case, we need a third tool to convert the model to a supported format.

Currently on the market, there are different tools for dynamic simulation [8] of mechanical structures [6] or ones that specialize on mobile robots [5, 7]. Adding to these general simulation platforms we can find more specialized tools for grasping robots [18] or even model-based robots [19].

Similar to what we try to achieve there are other simulators for mobile robots [5] but these are targeted for specific robotic structures, or are based only on certain simulation tools which provide functions for simulation and control.

Taking these into account, we have chosen Blender for our 3D modeling tool and Unity3D for our 3D simulation environment. These tools have provided what was necessary for our work to be completed and more, because the simulation environment can be completely custom made fulfilling our every needs by allowing low level programming of the simulations.

2. 3D MODELING

3D modeling is the first step in bringing new ideas to the testing phase. At this point, we only have the first drawing ideas and the engineers try to build the 3D mechanical parts of the robot. This stage is very important because many structural problems can be found and fixed with a relative small cost to the manufacturing process. At this stage every change in the design will cost only a few work hours if the change is not significant, but even if the whole design fails to achieve the proposed target, it will still be much more cheaper that to actually build the robot.

To achieve the 3D model of the entire robot, the designer builds every component and then he assembles them intro a whole mechanical structure.

Figure 1 to 3 presents some of the components of the RABOT robot made in a 3D modeling environment that put together will build one of the legs of the robot. As one can see, just one leg is made out of several complex elements, meaning that building a robot in real life will be much more difficult to create, with a highest degree of complexity that leads to errors in the manufacturing process. Also, no one can predict every problem within a design but, if we start using 3D modeling, these problems can be adjusted while the robot is being built within the 3D environment.

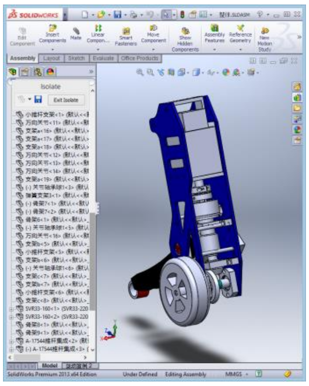

Figure 1 – Leg of RABOT robot

Figure 2 – Translation joint of RABOT robot

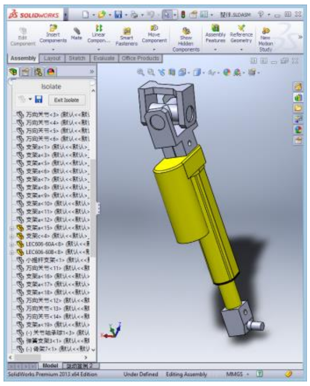

Figure 3 – Translation joint of RABOT robot

Using the components from figures 1, 2 and 3 and many others we can build the RABOT robot [4]. Figure 4 presents the components of the robot in an exploded view and figure 5 presents the assembled robot, while figure 6 presents the RABOT robot made with Blender 3D modeling software.

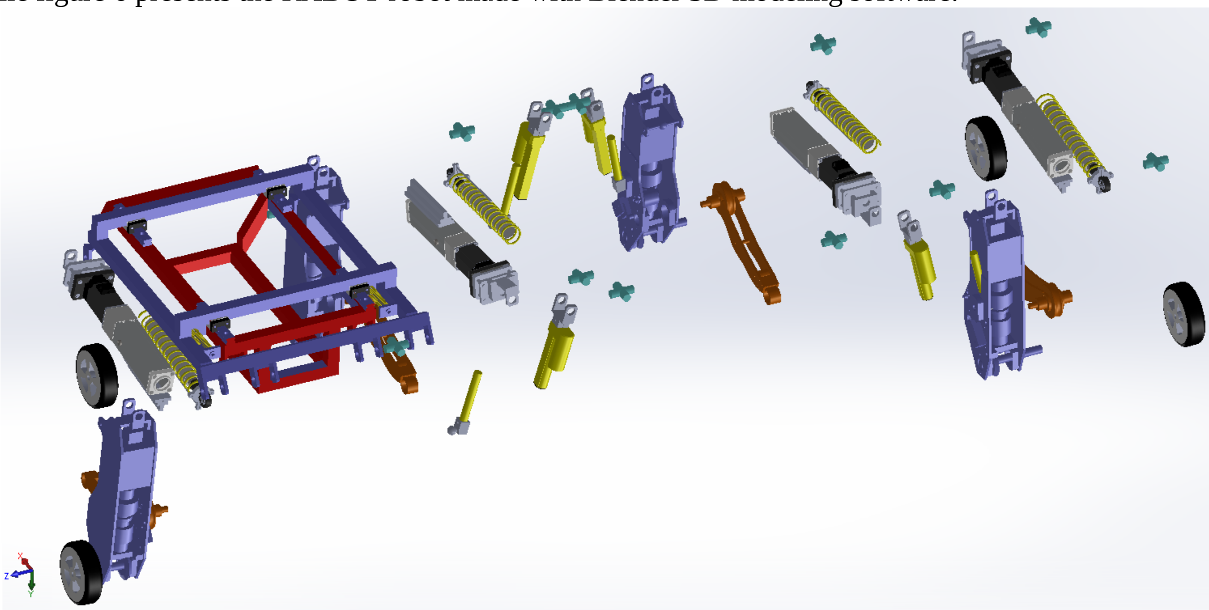

Figure 4 – Exploded view of the RABOT robot

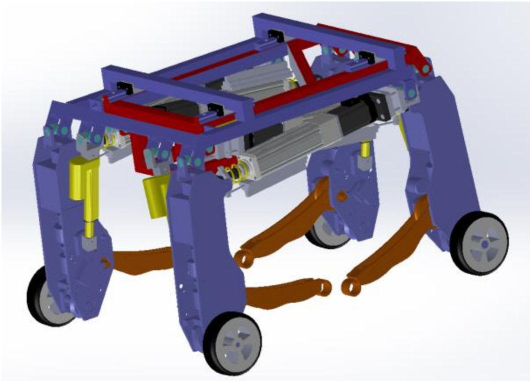

Figure 5 – The 3D model of the RABOT robot

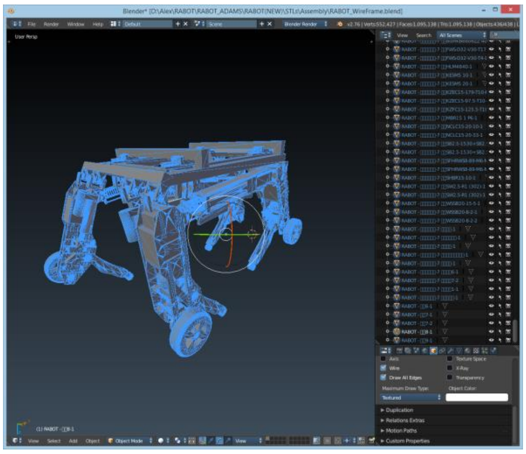

Figure 6 – The 3D model of the RABOT robot in Blender

While other 3D modeling tools have a paid license, Blender is released under the GNU General Public License. This means that the license grants people to use Blender for any purpose [2]. Also, because the software allows to import 3D models of standard file type which were made by other 3D modeling tools.

Using Blender we could adjust parameters or develop different 3D models of mechanical structure of NAO [3] and RABOT robots.

Figures 7, 8 and 9 present the 3D model of NAO robot which was created using Blender 3D modeling tool [3].

One concern about using Blender is that it is not so common in the world of 3D modeling, but it can use STL files saved from a different 3D modeling software. Another issue is that it is difficult to use at first, but it has a steep learning curve and a good number of users in the community.

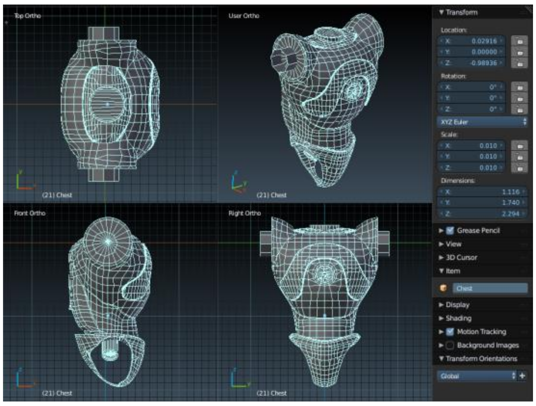

Figure 7 – Torso of NAO robot made in Blender [3]

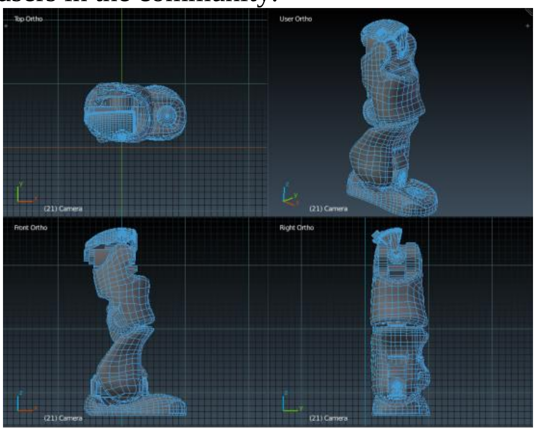

Figure 8 – Leg of NAO robot made in Blender [3]

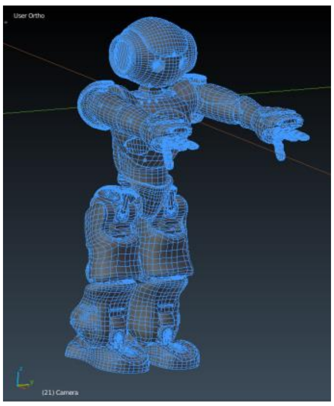

Figure 9 – NAO robot made in Blender [3]

Having the 3D model created now we can focus on implementing it into a virtual simulation using specialized tools which should be selected for the designed robots and application.

3. 3D TOOLS FOR VIRTUAL SIMULATION

So many 3D modeling tools have appeared, that now, every project should have a documentation phase in which the required 3D tools should be compared and analyzed according to the project needs.

Some of the requirements which should be taken into account by developers can be found in table 1, and table 2 presents the abbreviations used in table 1.

Table 1: Main features of the most common 3D simulation platforms

Platform: / **Feature:**	**MRD**	**U3D**	**Gz**	**Sb**	**MRS**	**Wb**	**Vr**	**ML**	**US**	**SS**
Free license	X	-	X	X	-	-	-	-	X	X
C++, C# programming	X	X	X	-	X	X	X	X	-	-
Java programming	-	X	-	X	X	X	X	-	X	-
Can use VPL	X	-	-	-	-	-	-	X	-	-
Physics	X	X	X	-	X	X	X	-	X	X
Can import 3D models	X	X	-	-	X	X	X	X	X	X
Can develop detailled 3D environments	X	X	X	-	X	X	X	X	X	-
Documentation	X	X	X	X	X	X	X	X	X	-
Extensive documentation	X	X	-	-	-	X	-	X	-	-
Examples and tutorials	X	X	X	-	X	X	X	X	X	-
User comunity	X	X	X	-	X	X	X	X	X	-
Big user comunity	X	X	-	-	-	-	-	X	-	-
Requires other licenses	X	-	-	-	-	-	-	-	-	-
Can be used on Windows	X	X	-	X	X	X	X	X	X	X

Table 2: 3D Simulation tools abbreviations

Simulation Tool	**Short Name**
Microsoft Robotics Developer Studio	MRD
Unity 3D	U3D
Gazebo	Gz
Simbad	Sb
Marilou Robotics Studio	MRS
Webots	Wb
V-rep	Vr
Matlab	ML
USARSim	US
SimSpark	SS

As one can see, there are several features that a 3D tool for Virtual Simulation should have, and others are useful when creating simulations that try to mimic the real life behavior of robots. Among these we have Unity 3D. This software has a free license version which the research community can use because they don’t always use software for economical purposes.

Unity 3D has allowed us to import the 3D model made in Blender and use it as is. By adding the mass property, it will behave almost as real robots do, due to its physics engine. In addition to its default behavior, one can add plugins that can increase the precision in motion of the mechatronic structures which will allow further development of robots by testing them and their control laws in situations most similar to real environments.

4. VIRTUAL SIMULATION

To simulate the virtual robots made with 3D modeling tools, we use the Unity 3D platform. This platform allows us to not only add 3D models of mechanical structures, but also create environments, as new objects which can be placed anywhere inside the simulation [11, 13, 16].

The 3D models which are added to the simulation exist only as floating objects which don't have mass, density nor tension surface. These properties have to be added by hand for each part of the assembly. If we have only a static object which is part of the environment, then we'll only need to add a mass property and a collider one. But if we have a body that is part of a mechanical assembly, then that object must have a rigid body property conferring mass and air drag. If the object is linked with another by a joint, then we need to add a joint property that allows the two objects to behave like a real joint would be put in place. In this fashion one can build a robot from parts, or import it from a complex model and then add the properties required for that specific structure.

Using this method we have used a 3D model of NAO robot and RABOT and we have introduced them in the virtual reality of Unity3D.

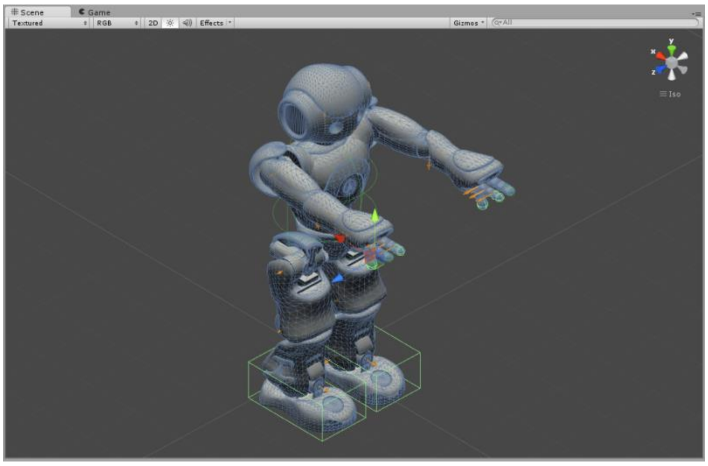

Figure 10 – 3D model of NAO in Unity with mass, joint and collider properties

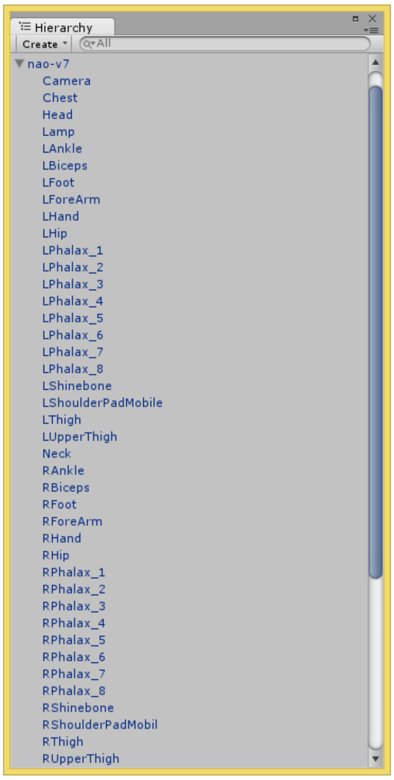

Figure 11 – List of NAO body components in Unity

Figure 10 present the NAO robot after it was imported in Unity3D and then add all the components we need to make it functional. These components include:

- "Rigidbody" property (figure 12) with the following options:
  - Mass of the object
  - Drag and Angular Drag which influence the object friction with air
  - Use Gravity
  - Is Kinematic which enables or not the dynamics of the robot
  - Constraints to freeze motion for 3 axes translation and 3axes rotation
- "Box Collider" property (figure 13) with limited options:
  - Material which we don't need for our purposes, but which can add different properties to the material of the object to which the property is added
  - Center being the center position of the collider within the object

  - Size is the dimension of the collider
- "Hinge Joint" property (figure 14) is necessary for the objects that are linked to other through joints. This component has the following properties:
  - Connected Body, required so that the current body will be attached to the connected one by the joint.
  - Anchor is the coordinates in the current object frame, to which the joint will be attached. This is the joint center position.
  - Axis is the rotation direction of the joint and is given in local space.
  - Spring is a force by which the object will try to reach a certain position.
  - Motor is the most used property for our application, and will add a driven motor in the joint center. Its properties (target velocity and force) will define the motor torque and speed.
  - Free Spin will disable the motor brake.
  - Use Limits will use the Min/Max properties so that the motor will act as physical constraints are present.
  - Break Force and Break Torque are the maximum values under which the joint will not break. If during the simulation these are exceeded the joint will break and the link between the two bodies will disappear.

Figure 11, present some of the components within the 3D model of NAO robot, which are all connected through Hinge Joints.

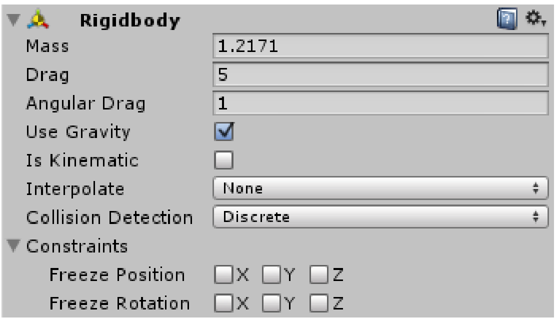

Figure 12 – Ridigbody property used to add physical properties to objects

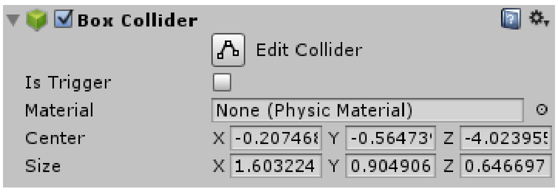

Figure 13 – Box Collider property, used to mark an object to interact with other colliders

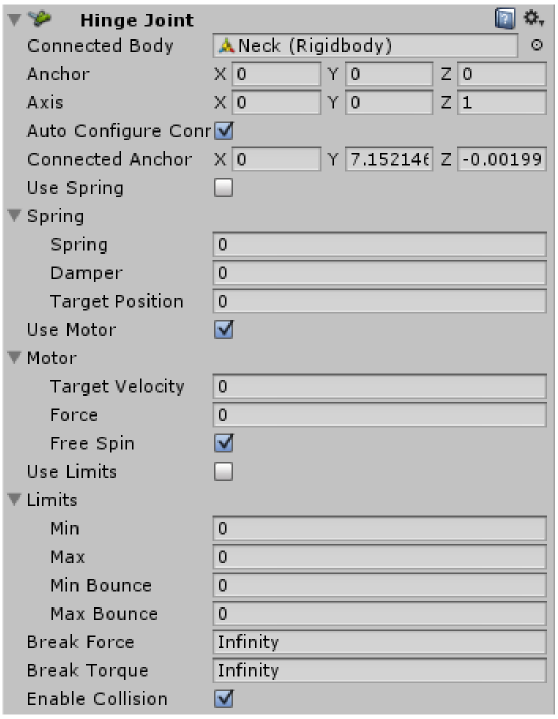

Figure 14 – Hinge Joint property, used to add single degree rotation joints as links between objects

To control the motor in each joint we have created a custom property called "NAOJoint" presented in figure 15. This feature allows us to control each joint using a PID controller. The PID controller will be different for each joint so the PID parameters will have different values.

This feature of Unity in which one can add custom properties to every object is a really good when we want to improve existing features, or create different ones required by each application, especially when the used tools don't have specialized featured for mechanical structures or real-time control.

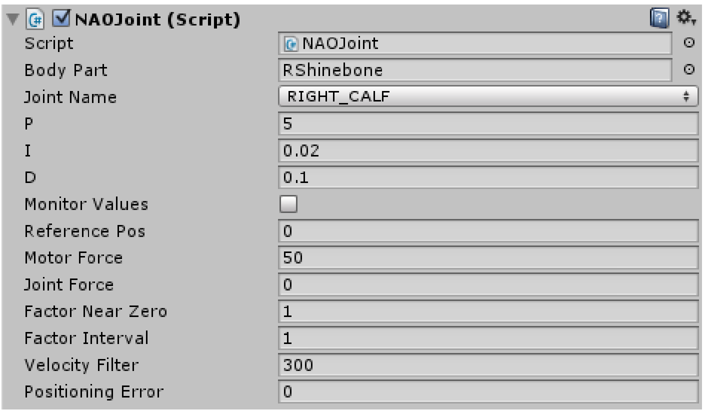

Figure 15 – The custom property for controlling the motor joint

Adding the PID controller to each robot joint, we now can use it for real time simulations. During the simulations, we compute the reference for each joint and then send it to the PID controllers. In this way we have used the Virtual Projection Method which integrates into VIPRO Platform.

5. CONCLUSIONS

We have designed a workflow which integrated into VIPRO platform helps reduce the development time on creating mobile robots, testing them using the Virtual Projection Method and 3D tools for modeling and simulation. Workflow:

Step 1 – Design the robot in a modeling environment (Blender, Maya, Solid-Works, AutoCAD Inventor, etc.). If the chosen tool was Blender, go to Step 4.
Step 2 – Save the files as STL files for each component or main part of the robot.
Step 3 – Import the STL files into Blender
Step 4 – Save the .blend file with the 3D model of the robot.
Step 5 – Import the .blend file into Unity3D
Step 6 – Create your custom environment and control code.
Step 7 – Run your Simulation and validate your results.

Using these steps, one can increase his productivity, by using the 3D modeling tool which he's accustomed to, thus decreasing modeling time. By decreasing modeling time, we can deliver multiple variations of the same mechanical structure, for the client to choose from. This will also mean cheaper research for new robots and a smaller investment for companies to develop new technology and robots. Some advantages and disadvantages for using 3D modeling and 3D simulation are presented in table 3.

Table 3: Advantages and disadvantages for using simulations in the development process of a mobile robot

Advantages	**Disadvantages**
• Reduce the costs of design/build of a robot. • Researcher can simulate several variations of the mobile robots with no adding costs. • The physics are already implemented in the 3D simulation tool. • The team can test the robot, its components and its environment individually or altogether. • The simulation can be conducted in stages if the project is too complex. • You can test and prove the system's reliability. • The team can run diagnosis tests for certain parts of the robot code or certain robot motions or behaviors. • Depending on the 3D simulation tool, it can be allowed multiple programming languages to interact. • Reduced time for developing experimental model. • Can use predefined software modules to test certain components or generate signals encountered by robots in common scenarios. • Can easily store robot data for off-line analysis. • Can visualize in real-time the robot parameters for on-line or off-line adjustments.	• The 3D simulation software can simulate only the events which were already modeled or programmed by someone. • The robot can encounter in real life a lot more scenarios than the ones that are programmed into the 3D virtual environment. • The virtual environment has to be created, and it will usually be predetermined. • The programming language offered by the software tool can sometimes be hard to control and use, and the users community can be very small to ensure capable

• Can use libraries for robot components (sensors, actuators, etc.) adding them to simulations. For the most common 3D simulation tools there are big communities where users help each other for technical support.	technical support.

By using the 3D modeling and simulation tools (Blender and Unity 3D) we have reduced the time for developing a mobile robot. By integrating this design process into the VIPRO platform, we now have the ability to test even further the capabilities of the designed mobile robots. This means that when sending the mobile robot design to be built as experimental model we now should have a working prototype. At this point any problems that will be encountered will be of small severity which will not need the rebuild or redesign of the entire mechanical structure.

ACKNOWLEDGEMENT. This work was accomplished through the Partnerships Program in priority fields - PN II, developed with the support of MEN-UEFISCDI, PN-II-PT-PCCA-2013-4, ID2009, VIPRO project no. 009/2014.

REFERENCES:

1. Vladareanu Luige. "Versatile Intelligent Portable Rescue Robot Platform through the Adaptive Networked Control."
2. www.blender.org
3. https://community.aldebaran-robotics.com
4. Xiaojie Wang, Xiaoyun Wang, Hongnian Yu, Hongbo Wang, Ling Lu, Luige Vladareanu and Daniel Octavian Melinte, "Dynamic analysis for the leg mechanism of a wheel-leg hybrid rescue robot," 2014 UKACC International Conference on Control (CONTROL), Loughborough, 2014, pp. 504-508. doi: 10.1109/CONTROL.2014.6915191
5. Zaratti Marco, Fratarcangeli Marco, Iocchi Luca, “A 3D simulator of multiple legged robots based on USARSim”, In: Robot Soccer World Cup. Springer Berlin Heidelberg, 2006. p. 13-24.
6. DELP, Scott L., et al. “OpenSim: open-source software to create and analyze dynamic simulations of movement”. IEEE transactions on biomedical engineering, 2007, 54.11: 1940-1950.
7. Gerkey, Brian; Vaughan, Richard T.; Howard, Andrew. “The player/stage project: Tools for multi-robot and distributed sensor systems”, In: Proceedings of the 11th international conference on advanced robotics. 2003. p. 317-323.
8. Ivaldi, Serena, Vincent Padois, and Francesco Nori. "Tools for dynamics simulation of robots: a survey based on user feedback." arXiv preprint arXiv:1402.7050 (2014).
9. Powell, Matthew J., Ayonga Hereid, and Aaron D. Ames. "Speed regulation in 3D robotic walking through motion transitions between human-inspired partial hybrid zero dynamics." Robotics and Automation (ICRA), 2013 IEEE International Conference on. IEEE, 2013.
10. M. Alexandru Moisescu, Ioan Dumitrache, S. I. Caramihai, A.M. Stanescu, I.S. Sacala, The Future of Knowledge in Manufacturing Systems in the Future Era of Internet of Things, IFAC- MCPL 2010, 8-10th September, Coimbra, Portugal, Published in: The Future of Knowledge in Manufacturing Systems in the Future Era of Internet of Things Management and Control of Production and Logistics, DOI 10.3182/ 20100908-3-PT-3007.00041
11. Vladareanu, V; Dumitrache, I; Vlădăreanu, L; Sacala, IS; Tont, G; Moisescu, MA, “Versatile Intelligent Portable Robot Control Platform Based on Cyber Physical Systems Principles”, Studies In Informatics And Control Volume: 24 Issue: 4 Pages: 409-418 Published: Dec 2015, WOS:000366543700005
12. Wang, Hongbo, Dong Zhang, Hao Lu, Yongfei Feng, Peng Xu, Razvan-Viorel Miha, Luige Vladareanu, „Active training research of a lower limb rehabilitation robot based on constrained trajectory”, Proceedings of the 2015 Advanced Mechatronic Systems (ICAMechS), 2015 International Conference on, pg. 24-2, IEEE, 2015.
13. S. Broumi, M. Talea, F. Smarandache and A. Bakali, Single Valued Neutrosophic Graphs: Degree, Order and Size. IEEE International Conference on Fuzzy Systems (FUZZ),2016, pp. 2444-2451.
14. Victor Vladareanua, Radu I. Munteanub, Ali Mumtazc, Florentin Smarandachedand Luige Vladareanua *, “The optimization of intelligent control interfaces using Versatile Intelligent Portable Robot Platform”, Procedia Computer Science 65 (2015): 225 – 232, ELSEVIER, DOI 10.1016/j.procs.2015.09.115, Publiched 2015
15. Wang, Hongbo, Liyu Xie, Xiong Zhao, Jianye Niu, Zhengcao Liu, Luige Vladareanu, and Razvan-Viorel Mihai. "Mechatronics system design and experiment research for a novel patient transfer apparatus." In Advanced Mechatronic Systems (ICAMechS), 2015 International Conference on, IEEE, 2015.
16. F.Smarandache, L.Vlădăreanu, Applications of Neutrosophic Logic to Robotics - An Introduction, The 2011 IEEE International Conference on Granular Computing Kaohsiung, Taiwan, Nov. 8-10, 2011, pp. 607-612, ISBN 978-1-4577-0370-6, IEEE
17. Lakkad, Shailesh. "Modeling and simulation of steering systems for autonomous vehicles." (2016).
18. Vahrenkamp, N., Kröhnert, M., Ulbrich, S., Asfour, T., Metta, G., Dillmann, R., & Sandini, G. (2013). Simox: A robotics toolbox for simulation, motion and grasp planning. In Intelligent Autonomous Systems 12 (pp. 585-594). Springer Berlin Heidelberg.
19. Erez, T., Tassa, Y., & Todorov, E. (2015, May). Simulation tools for model-based robotics: Comparison of Bullet, Havok, MuJoCo, ODE and PhysX. In Robotics and Automation (ICRA), 2015 IEEE International Conference on (pp. 4397-4404). IEEE.