SEVENTH FRAMEWORK PROGRAMME

CODYCO

FP7-600716

Whole-Body Compliant Dynamical Contacts in Cognitive Humanoids

D1.1
Enhanced iCub simulator for whole-body contact simulation

Editor(s)	Vincent Padois
Responsible Partner	UPMC
Affiliations	[1] First Institute, [2] Second Institute
Status-Version:	Draft-0.1
Date:	Feb. 28, 2014
EC Distribution:	Consortium
Project Number:	600716
Project Title:	Whole-Body Compliant Dynamical Contacts in Cognitive Humanoids

Title of Deliverable:	Enhanced iCub simulator for whole-body contact simulation
Date of delivery to the EC:	28/2/2014

Workpackage responsible for the Deliverable	WP1
Editor(s):	editors
Contributor(s):	contributors
Reviewer(s):	reviewers
Approved by:	All Partners

Abstract	The scope of the current deliverable is to present the results ...
Keyword List:	kw, list, etc,

Document Revision History

Version	Date	Description	Author
First draft	date	description	author

Table of Contents

Index of Figures

1 Introduction

The work described in this deliverable is part of WP1 and aims at providing the CoDyCo consortium with a shared framework for the simulation of humanoid robots and/or digital humans involved in whole-body and multi-contact activities. In order to reach this goal, several activities have been led in parallel. They are described in this deliverable as follows. In section 2, the definition of the requirements for the simulation framework is provided. In section 3, a survey of the existing simulators for robotics is presented. In section 4, iCubsim, the historical iCub simulator developed at IIT, is described and two alternative solutions, better suited for the simulation of whole-body motions in contact, are introduced. The results of the comparison between these two simulators and a real iCub performing a free-falling task are provided in section 5. The work achieved for this deliverable is summarized in the Conclusion and some perspectives are given. Appendix A contains the survey paper on simulators submitted to the IEEE Robotics and Automation Magazine. Appendix B contains the manual of the digital human URDF model generator developed at JSI.

2 Requirements for enhanced iCub simulator for whole-body contact simulation

In this section, the definition of the requirements for the simulation framework is provided.

2.1 Motivations

With the progress of powerful computers enabling fast computations, dynamics simulation in robotics is no longer expected to be an offline computational tool. It is used to rapidly prototype controllers, evaluate robots design, simulate virtual sensors, provide reduced model for model predictive controllers, supply with an architecture for real robot control, and so on.

This is especially true in the framework of the CoDyCo project where each technical work package can benefit from an efficient, modular dynamic simulator. Such a simulator is for example useful :

- in WP2 to evaluate the validity of dimensionally reduced models of human whole body motion in contact using simulated digital humans;
- in WP3 to rapidly prototype and evaluate the whole-body reactive controllers and potentially provide computationally efficient reduced model for model predictive controllers;
- in WP4 to bootstrap the learning algorithms without requiring the use of real robots in the first stage of the learning process;
- in WP5 to extensively test the various validation scenarii before running them on the real robots.

2.2 Critical features

Dynamics simulators for robotics have more strict requirements than the ones used for animating virtual characters, where time, computational burden and physical reality can be less con straining. In entertainment (e.g. video-games), infeasible forces may not be a problem since the law of physics can be violated. In bio/mechanical studies, simulators can be used offline to analyse or synthesize behaviours. Although the field of dynamics modelling and simulation has matured over the last decades [1, 2, 3], the growing need to control whole-body movements of complex structures, such as humanoids, poses additional challenges to simulators for robotics:

1. numerical stability, which poses strong limitations on the use of simulations in real-time control settings [4, 5];

2. the capability to be used as predictive engines in real-time control loops [6], which requires the ability to be extremely fast in computing the dynamics and the guarantee for the solvers to converge to physically feasible solutions upon a certain time [7];

3. the simulation of rigid and soft bodies in contact with rigid and compliant environments [8, 9]: the inaccurate computation of contact forces between bodies may result in unrealistic contacts or physically infeasible contact forces (this issue has been particularly evident in the virtual phase of the Darpa Robotics Challenge - DRC);

4. the capability to model and simulate new types of actuation systems, such as variable impedance or soft actuators [10], and different types of contacts, for example with deformable materials, compliant and soft surfaces [11].

Modularity is also a critical feature. Indeed, depending on the use made of the simulator all components such as the 3D graphics display, the graphical user interface, the interfaces with input devices (keyboard, space mouse,...), the physics core, the controller, etc. may not be required : a modular, component-based, software architecture can reduce the non-required computational load. It can also permit the use of different solutions for a given component such as the physics core, the 3D graphics display or the controller. Finally, if components are glued together using a middle-ware such as YARP [12], ROS [13] or OROCOS [14], the integration of a controller prototyped in simulation on the real robot can be largely simplified as the way to access (basically set control modes, send control inputs and get sensors feedback) to the simulated and real robot can be strictly similar.

Finally, the robotics community urges for standardized software tools and particularly open source software. The benefit of open-source is not only in the community that can grow around the software, developing new tools, improving its quality and avoiding to "re-invent the wheel" at each time, but also in checking its efficiency and robustness on real platforms (which is expensive).

2.3 Conclusions

Within the framework of the CoDyCo project, a modular, component-based dynamic simulation software providing numerically stable, computationally efficient and physically consistent simulations of whole-body virtual human(oid) systems in contact with rigid or soft environments is required.

3 Survey of existing robotics simulators

In this section, a review of existing simulators for Robotics is provided.

3.1 Objectives of the survey

There is a growing number of tools for dynamics simulation, ranging from dynamic solver libraries to systems simulation software, provided through either open or closed source code solutions, each more or less tailored to their expected domains of application.

The spectrum of robotics applications being large and in expansion, it is necessary for the developer community to have a feedback about the users' needs, and for the researchers to be aware of the available tools and have the elements to ponder which of the available tools is the best for their research. Most middleware for robotics (ROS, YARP, OROCOS, Player, etc.) are already open-source, some also cross-platforms. This makes it possible to produce interesting performance comparisons that can help the roboticists to pick the best middleware for their needs [15]. Similar ideas (open-source and cross-platform compatibility) should be used to compare dynamics models and simulators. For example, an interesting evaluation and performance comparison of contact modeling algorithms was presented in [4, 5].

As a complement to quantitative comparisons, a useful element of evaluation (often un-mentioned and neglected) is user feedback. What do users really think of the software they use for simulation? Would they suggest it? What is their experience in their particular use case? It is believed that user feedback may be useful to avoid time-consuming tuning and inappropriate choices of software to researchers. It could point a researcher to a community that is actively using the tool and that is sharing the same concern: for example, it is likely that people simulating flying robots have different needs than those simulating wheeled robots or those controlling bipeds. Furthermore, user feedback can provide useful suggestions to the developers community about the things that matter the most to users in simulation.

With this goal in mind, an online survey about the use of dynamical simulation in robotics has been created[1]. The survey is divided into four parts: general information about the user, user experience with dynamics simulation in general, user experience with one tool of his choice, technical questions and subjective evaluation about the selected tool. The survey has been advertised on the main robotics mailing lists (e.g., euron-dist, robotics-worldwide) as well as in other mailing lists of correlated disciplines (e.g. comp-neuro), and kept open for approximately one month.

The paper in Appendix A, submitted to the IEEE Robotics and Automation Magazine, summarizes the analysis of the users' answers. A descriptive fiche for the most relevant

[1] Online survey: `http://goo.gl/Tmyf5A`

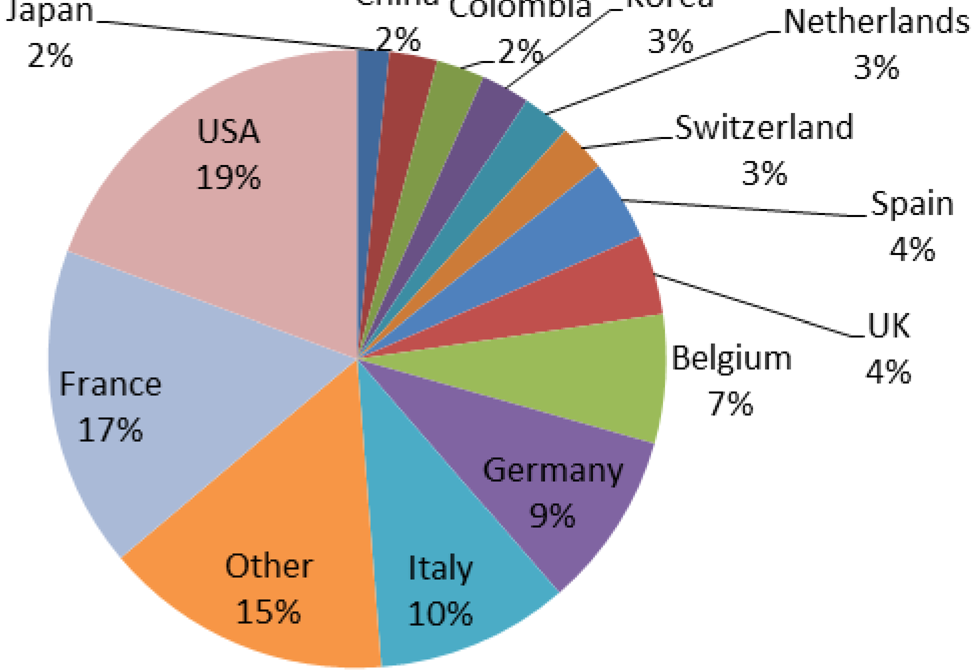

Figure 1: Country of provenience for the participants to the survey.

software tools, for the reader's interest, is also reported. For the complete analysis of the simulators survey, the reader is referred to the extended version of the report[2].

A summary of the findings of this survey is provided in the next subsection.

3.2 Findings of the survey

The analysis of the survey is reported hereinafter.

3.2.1 About the participants

The survey was filled by 119 participants (92% male, 8% female; age 32 $\pm$ 6, min 20, max 57), whose 62% holds a PhD degree and 35% a BS or MS degree, mostly from USA, France, Italy and Germany (see Figure 1). Participants work mostly in University (70%) or do R&D in public (16%) or private (14%) institutes. Their primary areas of research are: 21% control, 14% locomotion, 10% machine learning, 9% HRI, 8% planning, 6% mechanical design, 5% cognitive robotics, 5% mathematical modeling. Their primary application field is: 26% humanoid robotics, 20% mobile robotics, 11% multi-legged robotics, 8% service robotics, 7% industrial robotics, 7% numerical simulation of physical systems, 5% flying robots. Among the participants working in humanoid robotics, 16% is also competing in the Darpa Robotics Challenge (DRC), which makes 8% of the participants to the survey - 10 people.[3]

3.2.2 General knowledge about simulating tools

Participants were asked to indicate their familiarity with some of the most common existing simulation tools. A list of existing software tools for simulations, used in different contexts, was provided. The users were asked to indicate whether the software was currently used or

[2] `http://www.codyco.eu/survey-simulation`

[3] Interestingly, the software tool they indicated as the one currently used for their research (we can presume for the DRC as well) is Gazebo (3), MuJoCo (2), Robotran (2), Drake (1), Autolev (1) and ODE (1).

Tool	Currently used, and it is the main tool	Currently used, but not the main tool	Currently used, just to test it	Used once, just to test it	Used then abandoned	Known, but never used	Never heard of
Gazebo	[redacted]	7%	3%	18%	10%	34%	[redacted]
ODE	[redacted]	12%	5%	18%	[redacted]	22%	[redacted]
Bullet	5%	13%	7%	12%	10%	29%	24%
V-Rep	5%	3%	3%	18%	3%	29%	39%
Webots	4%	7%	1%	16%	[redacted]	32%	27%
OpenRave	5%	3%	2%	7%	5%	29%	49%
Robotran	4%	0%	1%	4%	2%	13%	76%
XDE	5%	3%	0%	3%	1%	14%	74%
Blender	5%	17%	7%	22%	6%	28%	[redacted]
MuJoCo	2%	0%	0%	4%	2%	21%	71%
iCub_SIM	4%	4%	2%	3%	3%	29%	55%
Nvidia PhysX	1%	1%	4%	12%	7%	43%	32%
OpenSIM	3%	4%	3%	8%	1%	41%	40%
HumanS	0%	0%	0%	1%	1%	10%	[redacted]
Moby	2%	1%	0%	0%	2%	14%	81%
Vortex	3%	2%	0%	5%	5%	17%	68%
RoboRobo	3%	1%	0%	0%	1%	4%	[redacted]

Table 1: Knowledge and past/present use of simulators.

not for their researches, if it had been used before or if it was unknown. A summary of the percentage of answers for the most relevant tools is shown in Table 1.

The **software tools that have more than 5% of user share** (i.e., positive answers to the fact that the software is currently used and it is the one or one of many main tools): the most used are Gazebo (15%) and ODE (11%), with a gap with respect to Bullet, OpenRave, V-Rep, XDE and Blender, all at 5%. These values provide an indicative dimension of the user community around each software tool.

The software tools that are less known (because maybe they were not sufficiently advertised or do not have a big community behind) and the ones that are most known (even if this does not necessarily means that they are used) can be retrieved from the column "Never heard of this software" from Table 1[4]. The **most known tools** are ODE (10%), Gazebo (15%), Blender (15%), Bullet (24%), Webots (27%), Nvidia PhysX (32%), Stage (38%), V-Rep (39%), OpenSIM (40%) and ADAMS (45%). Interestingly, the first three are also open-source projects.

An important information that we acquired through the survey is about the abandon of software for simulation: this can be found in the column "Used than abandoned" in Table 1. The **most abandoned software after use** are ODE (22%), Stage (16%), Webots (13%), Bullet (10%), Gazebo (10%), Nvidia PhysX (7%), OpenHRP (6%), Blender (6%), Open-

[4]Actually, Table 1 is only showing values for the most relevant software tools. To see the full data, we refer the reader to the full report of the survey.

Rank	Feature	Evaluation
1	Stability of simulation	Very important
2	Speed	Important
3	Precision of simulation	Important
4	Accuracy of contact resolution	Important
5	Same interface between real & simulated system	Important
6	Computational load (CPU)	Neutral
7	Computational load (memory)	Neutral
8	Visual rendering	Neutral

Table 2: Most important features for a simulator.

Rave (5%), Vortex (5%). Though this set may seem as a sort of "blacklist" of tools that disappointed users, it must be observed that most of them are open-source softwares that could have been the "one among many" tools that have been used then in one researcher's life; however, it can be equally presumed that the high percentage of abandon can be partly correlated to the difficulty that users have encountered in using these tools and partly by their "seniority".

3.2.3 Important features for simulation

We asked participants to indicate the main purposes for the use of dynamics simulation in their research (they could indicate more than one): 66% simulating the interaction of the robot with the environment, 60% simulating the robot locomotion, 59% simulating behaviors of the robot before doing them on the real robot, 49% simulating the robot navigation in the environment, 48% simulating collisions and interactions between bodies (not specifically robots), 41% testing low-level controllers for robots, 22% simulating multi-fingered grasp, 21% simulating human movements, 8% animating virtual characters.

We also asked participants to evaluate, upon their experience, what are the most important features for a good simulation (they could evaluate the importance of each element from "not important at all" - 1 to "very important, crucial" - 5). Their ranking of important features is reported in Table 2. The stability of simulation is the only element that was evaluated as "very important", whereas speed, precision and accuracy of contact resolution were marked important. Remarkably, the same API between real and simulated robot is also signed as important.

3.2.4 Criteria for choosing a simulator

We asked participants to indicate the most important criteria for choosing a simulator. The answer was broken in three parts, i.e. participants could point out the first, second, and third most important criteria. The first most important criteria: 32% simulation very close to reality, 24% open-source, 19% same code for real and simulated robot, 11% light and fast, 6% customization, 3% no inter-penetration between bodies, 5% other. The second and third

Rank	Most important criteria
1	Simulation very close to reality
2	Open-source
3	Same code for both real and simulated robot

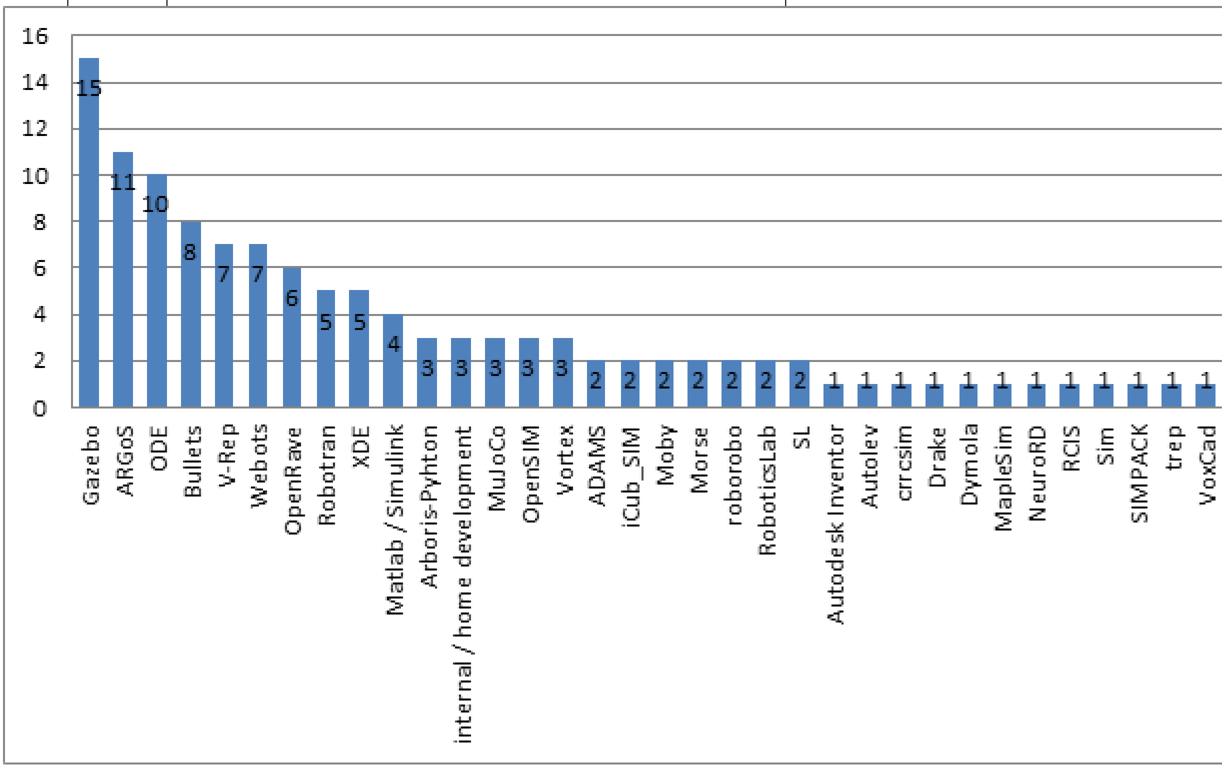

Figure 2: The simulation tools currently in use among the participants to the survey. The vertical axis reports the number of users that indicated the tool as their principal.

choice for the important criteria follow more or less accordingly. Considering the three criteria as a whole, i.e. grouping the three of them on the same level, the important criteria is 23% simulation very close to reality, 20% open-source, 18% light and fast, 16% same code for real and simulated robot, 14% customization, 4% no-inter-penetration between bodies, 1% ease to learn/use, 1% real time - based simulation, 2% other. If instead we consider the weight of each selection (most important=3, second important=2, third most important=1), then grouping the answers we have: 26% simulation very close to reality, 22% open-source, 17% same code for both real and simulated robot, 17% light and fast, 11% customization, 4% no inter-penetration between bodies (5% other)

3.2.5 Currently used tools

We asked participants to indicate the current simulation tool they are using. Results are shown in Figure 2. The most diffused software among the participants are: 13% Gazebo, 9% ARGoS, 8% ODE, 7% Bullet, 6% V-Rep, 6% Webots, 5% OpenRave, 4% Robotran, 4% XDE. All the other tools (see Figure 4) have less than 4% of user share. These tools are the ones we are focusing on in our following analysis.

Some technical information about the selected tools can be indicative of the user needs and use:

- **Primary OS**: 66% GNU/Linux, 30% Windows, 4% MAC OSX.
- **Primary API language**: 52% C++, 18% python, 13% Matlab, 8%C, 3% LUA, 2% Java; 3% of participants do not use an API
- **License**: 67% of the tools are open-source (GPL, Apache, BSD and analogous/derivatives licenses), only 17% of the tools have a commercial license, 16% have an academic license (i.e., they are free but not open-source).
- **Hardware**: 39% a powerful desktop (i.e., multi-core, 8/16GB RAM), 35% everyday laptop, 18% powerful desktop with powerful GPU card, 5% multi-core cluster.
- **Middleware**: 52% is not using the tool with a middleware, the remainder is using ROS (25%), YARP (6%), OROCOS (4%).

The research areas being different, we extracted the most used tools for a selection of research areas: results are shown in Table 4. The most relevant results are for humanoid robotics (31 users, that is 26% of the participants to the survey) and mobile robotics (25 users, that is 21% if the participants). For humanoid robotics, the most diffused tools are ODE and Gazebo, and there is a variety of several custom-made simulators. It is interesting to notice that Gazebo supports ODE and Bullet as physical engines, hence it is probable that the quota of ODE for humanoid robotics is higher. For mobile robotics, the most diffused tools among the survey participants are Gazebo, ARGoS and Webots.

The different concentration of tools for the different research areas reveals that some tools are more appropriate than others for simulating robotic systems in different contexts or applications. A researcher may therefore let his choice about the adoption of a simulator be guided by the custom in his field. With this in mind, we investigated what was the main reason for a researcher to pick up his current tool. Overall, the main reasons why they chose the current tool is: 29% the best tool for their research upon evaluation, 23% "inheritance", i.e. it was "the software" (already) used in their laboratory, 8% they are the developers, 8% it was chosen by their boss/project leader, 7% it is open-source, 7% it was happily used by colleagues. Only 3% of the participants chose the tool because of a robotic challenge. Interestingly there is quite a demarcation between the first reasons and the others. There are certainly some tools that distinguish for the fact that they have been chosen as best option for research, for example V-Rep (71%), Bullet (63%) and Gazebo (53%). Some tools have instead been adopted by "inheritance", i.e., they were already used in the lab: ARGoS (45%), Robotran (40%) and XDE (40%). For the latter, it is also a choice imposed by the project leader (40%).

We asked participants to evaluate their level of satisfaction of the use of their tool, in a global way, from Very negative (1) to Very Positive (5): all software tools were evaluated "positive", whereas only MuJoCo was "very positive" (subjective evaluation by 3 users). We also asked participants to indicate their level of satisfaction with respect to some specific aspects (documentation, support, installation, tutorials, advanced use, active project and community, API), and to rate each element on a scale from 1 to 5. Table 5 reports the mean and standard deviation of the notes received by the users of each tool.

Research area	Users	Most used software	Other used software
Humanoid Robotics	32	(4) ODE, (3) Gazebo, Robotran, OpenRave, Arboris-Python, (2) XDE, iCub_SIM	(1) Drake, MapleSim, MuJoCo, OpenSIM, RoboticsLab, SL, Vortex, V-Rep, Webots, own code
Mobile Robotics	25	(5) Gazebo, ARGoS, (3) Webots, (2) V-Rep, Vortex	(1) ADAMS, Autodesk Inventor, Bullet, ODE, Morse, roborobo, Sim, own code
Multi-legged robotics	13	(3) Webots, (2) ODE	(1) Gazebo, ADAMS, Autolev, Bullet, Moby, RoboticsLab, SIMPACK, VoxCad
Service robotics	12	(4) Gazebo, (3) OpenRave	(1) OpenSIM, V-Rep, Morse, RCIS, SL
Numerical simulation of physical systems	8	(2) Bullet	(1) MuJoCo, ODE, OpenSIM, Simulink, trep, XDE
Flying robots	6	(2) ARGoS	(1) Robotran, crrcsim, Gazebo, Simulink/Matlab
Swarm robotics	5	(4) ARGoS	(1) roborobo
Industrial manipulators	5		(1) Bullets, Dymola, Matlab, V-Rep, XDE
Mechanical design	4		(1) Moby, MuJoCo, V-Rep, own code
Human Motion analysis	3		(1) Robotran, Bullet, XDE
Snake robots	3	(2) ODE	(1) Matlab

Table 4: Most diffused tools for a selection of the research areas.

Tool	Documentation	Support	Installation	Tutorials	Advanced use	Active project & community	API	Global
Gazebo	3.47 ± 0.99	4.00 ± 1.07	3.93 ± 1.03	3.53 ± 1.12	3.80 ± 0.86		3.67 ± 0.82	3.88 ± 0.91
ARGoS	3.40 ± 0.70	3.90 ± 0.99	4.70 ± 0.48	4.20 ± 0.63	4.60 ± 0.70	4.10 ± 0.74	4.30 ± 0.67	4.17 ± 0.70
ODE	3.80 ± 0.63	3.40 ± 1.07	4.10 ± 1.28	3.20 ± 1.13	3.90 ± 1.37	3.30 ± 1.25	3.40 ± 1.26	3.59 ± 1.15
Bullets	3.37 ± 1.06	3.62 ± 0.91		4.00 ± 0.76	3.75 ± 0.71	4.37 ± 0.74	3.87 ± 0.83	3.96 ± 0.78
V-Rep			4.71 ± 0.76		4.28 ± 0.76	4.43 ± 0.53	4.14 ± 1.07	
Webots	3.86 ± 1.07	3.57 ± 1.13	4.43 ± 0.79	3.43 ± 1.51	4.42 ± 0.78	4.14 ± 0.69	4.57 ± 0.53	4.20 ± 0.96
OpenRave	3.50 ± 0.55	4.67 ± 0.52	4.17 ± 0.75	3.50 ± 1.22	4.33 ± 0.82	4.33 ± 0.52	4.33 ± 0.52	4.12 ± 0.70
Robotran	3.60 ± 0.55	3.80 ± 0.45	3.80 ± 0.45	3.20 ± 0.84	4.20 ± 0.84	3.20 ± 0.84	3.80 ± 0.45	3.66 ± 0.63
Vortex	3.33 ± 1.15	3.67 ± 1.53	5.00 ± 0.00	2.67 ± 0.58	3.67 ± 0.58		3.33 ± 0.58	3.48 ± 0.80
OpenSIM	4.33 ± 0.58	4.67 ± 0.58	3.67 ± 0.58	3.00 ± 1.00	4.00 ± 0.00	4.67 ± 0.58	3.67 ± 0.58	4.00 ± 0.55
MuJoCo	2.33 ± 1.15		4.33 ± 1.15	3.33 ± 1.15		4.00 ± 0.00		3.62 ± 0.66
XDE		2.80 ± 1.09				2.80 ± 0.84		

Table 5: Ratings for the level of user satisfaction of the most diffused tools.

3.3 Conclusions

4 Proposed technical solutions

4.1 Historical solution: iCubsim

4.2 iCubsim with Gazebo

4.3 iCub in XDE

5 Assessement of the proposed solutions

5.1 Assessement methodology

5.2 Comparison protocol

5.3 Results

6 Conclusion

A Robotics simulators survey paper

B Technical paper on the digital human URDF file generator

References

[1] R. Featherstone and D. E. Orin, *Handbook of Robotics*. B. Siciliano and O. Khatib Eds., Springer, 2008, ch. Dynamics, pp. 35–65.

[2] A. Jain, *Robot and Multibody dynamics: analysis and algorithms*. Springer, 2011.

[3] E. Todorov, "Analytically-invertible dynamics with contacts and constraints: theory and implementation in mujoco," in *IEEE Int. Conf. on Robotics and Automation*, 2014.

[4] E. Drumwright and D. Shell, "An evaluation of methods for modeling contact in multibody simulation," in *IEEE Int. Conf. on Robotics and Automation*, 2011, pp. 1695–1701.

[5] ——, "Extensive analysis of linear complementarity problem (lcp) solver performance on randomly generated rigid body contact problems," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2012, pp. 5034–5039.

[6] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2012, pp. 5026–5033.

[7] E. Todorov, “A convex, smooth and invertible contact model for trajectory optimization,” in *IEEE Int. Conf. on Robotics and Automation*, 2011, pp. 1071–1076.

[8] B. Brogliato, A. ten Dam, L. . Paoli, F. Gnot, and M. Abadie, “Numerical simulation of finite dimensional multibody nonsmooth mechanical systems,” *Applied Mechanics Reviews*, vol. 55, pp. 107–150, 2002.

[9] Y.-B. Jia, “Three-dimensional impact: energy-based modeling of tangential compliance,” *Int. J. Robotic Research*, vol. 32, no. 1, pp. 56–83, 2013.

[10] C. Duriez, “Control of elastic soft robots based on real-time finite element method,” in *IEEE International Conference on Robotics and Automation*. IEEE, 2013, pp. 3982–3987.

[11] C. Duriez, F. Dubois, A. Kheddar, and C. Andriot, “Realistic haptic rendering of interacting deformable objects in virtual environments,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 12, no. 1, pp. 36–47, 2006.

[12] G. Metta, P. Fitzpatrick, and L. Natale, “Yarp: Yet another robot platform.” *International Journal of Advanced Robotic Systems*, vol. 3, no. 1, 2006.

[13] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, “Ros: an open-source robot operating system,” *ICRA workshop on open source software*, vol. 3, no. 3.2, 2009.

[14] H. Bruyninckx, P. Soetens, and B. Koninckx, “The real-time motion control core of the Orocos project,” in *IEEE International Conference on Robotics and Automation*, 2003, pp. 2766–2771.

[15] E. Einhorn, T. Langner, R. Stricker, C. Martin, and H. Gross, “Mira - middleware for robotic applications,” in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2012, pp. 2591–2598.