Tools for dynamics simulation of robots: a survey based on user feedback

Serena Ivaldi[†,‡], Vincent Padois[†,‡] and Francesco Nori[§] [*†‡§]

February 27, 2014

Abstract

The number of tools for dynamics simulation has grown in the last years. It is necessary for the robotics community to have elements to ponder which of the available tools is the best for their research. As a complement to an objective and quantitative comparison, difficult to obtain since not all the tools are open-source, an element of evaluation is user feedback. With this goal in mind, we created an online survey about the use of dynamical simulation in robotics. This paper reports the analysis of the participants' answers and a descriptive information fiche for the most relevant tools. We believe this report will be helpful for roboticists to choose the best simulation tool for their researches.

1 Introduction

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.

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.

With this goal in mind, we created an online survey about the use of dynamical simulation in robotics.[1] The survey was 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 was 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.

This paper summarizes the analysis of the users' answers. We also report a descriptive fiche for the most relevant software tools, for the reader's interest. For the complete analysis of the simulators survey, we refer the user to the extended version of the report.[2]

[*]E-mail: serena.ivaldi@isir.upmc.fr

[††] Sorbonne Universités, UPMC Paris 06, UMR 7222, Institut des Systèmes Intelligents et de Robotique (ISIR), F-75005, Paris, France.

[‡‡] CNRS, UMR 7222, Institut des Systèmes Intelligents et de Robotique (ISIR), F-75005, Paris, France.

[§§] Robotics, Brain and Cognitive Sciences Dept., Italian Institute of Technology.

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

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

1.1 Why user feedback?

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 [1]. 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 [2, 3].

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? We believe 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.

1.2 Challenges in simulation

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 constraining. In entertainment (e.g. video-games), unfeasible forces may not be a problem since the law of physics can be violated. In bio/mechanical studies, simulators can be used offline to analyze or synthesize behaviors. Although the field of dynamics modeling and simulation has matured over the last decades [4, 5, 6], 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 [2, 3];

2) the capability to be used as predictive engines in real-time control loops [7], 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 [8];

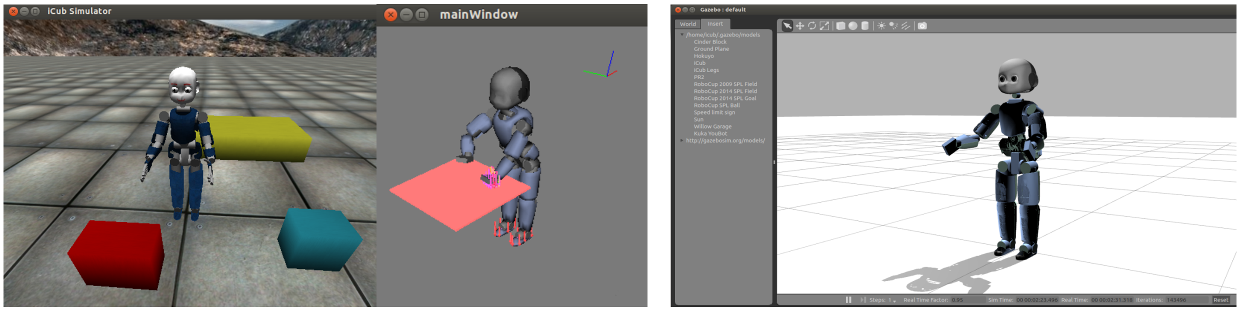
3) the simulation of rigid and soft bodies in contact with rigid and compliant environments [9, 10]: the inaccurate computation of contact forces between bodies may result in unrealistic contacts or physically unfeasible contact forces (this issue has been particularly evident in the virtual phase of the Darpa Robotics Challenge - DRC);

4) the capability to model and simulate new types of actuation systems, such as variable impedance or soft actuators [11], and different types of contacts, for example with deformable materials, compliant and soft surfaces [12].

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).

1.3 The iCub case

The iCub community recently faced the problem of choosing the correct tool for whole-body dynamics simulation. The existing simulator iCubSim [13], is based on ODE and is mostly used as a tool for testing behaviors before trying them on the real robot. It is provided with an interface that emulates the low-level control of iCub, so the same code can be used to control simulated and real robot. However, the dynamics engine makes it inadequate for research about control of contacts and compliant surfaces. At the moment two solutions are investigated: one based on XDE and the other based on Gazebo. The choice of these tools has been based on objective criteria (license, developing community, stability of the software simulation), previous experience and "subjective feedback" acquired orally discussing with colleagues, that provided



(a) iCubSim, based on ODE. (b) iCub in XDE. (c) iCub simulated in Gazebo.

Figure 1: Simulators of iCub. From left to right: iCubSim, based on ODE, XDE and Gazebo. (credits for Gazebo: Silvio Traversaro)

partial and unstructured information. A more structured information about user feedback would have been helpful. We believe this survey analysis could be a further element for choosing the best simulation tool in a research project.

1.4 Comparing simulators

It is certainly difficult to enumerate all the criteria that one can examine to choose a dynamics simulator, especially for a humanoid robot that is supposed to have physical interactions with rigid and compliant environments.

First, one can choose between physics engines (e.g. ODE, Bullet) and more complex softwares that include system simulation (e.g. Gazebo, V-Rep).

Second, facing the decision to adopt a simulator for a robot, a researcher should first decide between softwares that also include system simulation, and softwares which only simulate the dynamics of multi-body systems. This criterion allows us to consider under different perspectives two set of softwares: the first set, composed of software like Gazebo, OpenHRP, iCubSIM, which facilitate seamless simulation and control of the virtual characters and their corresponding physical system/robot; the second, like Humans, OpenSIM, Robotran, that are able to simulate the dynamics of complex systems but are not meant to provide seamless control of robotics platforms.

Another element of discrimination is the way the simulator represents rigid-body structures: on one hand we have software based on ODE and Bullet, such as Gazebo, iCubSim, MORSE, which represents joints as constraints between bodies; on the other we have softwares like XDE, OpenHRP, which make use of parameterized rigid-body dynamics representations, where joints are simply part of the robotics structure. These two classes determine not only the way forward/inverse dynamics are computed (and of course the second group also benefits from the straightforward computation of quantities useful in robotics, such as Jacobians, mass matrices etc.), but most importantly the way contact forces are computed. The first class considers contacts forces as bilateral/unilateral constraints, which are added to the list of constraints used to describe the joints; then the same solver is used to find the forces for the global system, including contacts and joints. In the second class, on the contrary, only constraints from the contacts are solved, which notably simplifies the problem. In generla, finding the correct contact forces can be burdensome. Current approaches to solve this problem are mostly based on the Linear Complementarity Problem (LCP) [3], and in some cases there are mixed approaches combining LCP with optimization techniques, such as in MuJoCo [7].

In short, there are several “objective” criteria that one can look at, on the basis essentially of what is advertised by the developers as a “supported feature”. However, it is very difficult to find practical comparison of different simulators on test problems, for many reasons: first, an extensive comparison would require access to the source code but not all software is released under open-source; second, even open-source

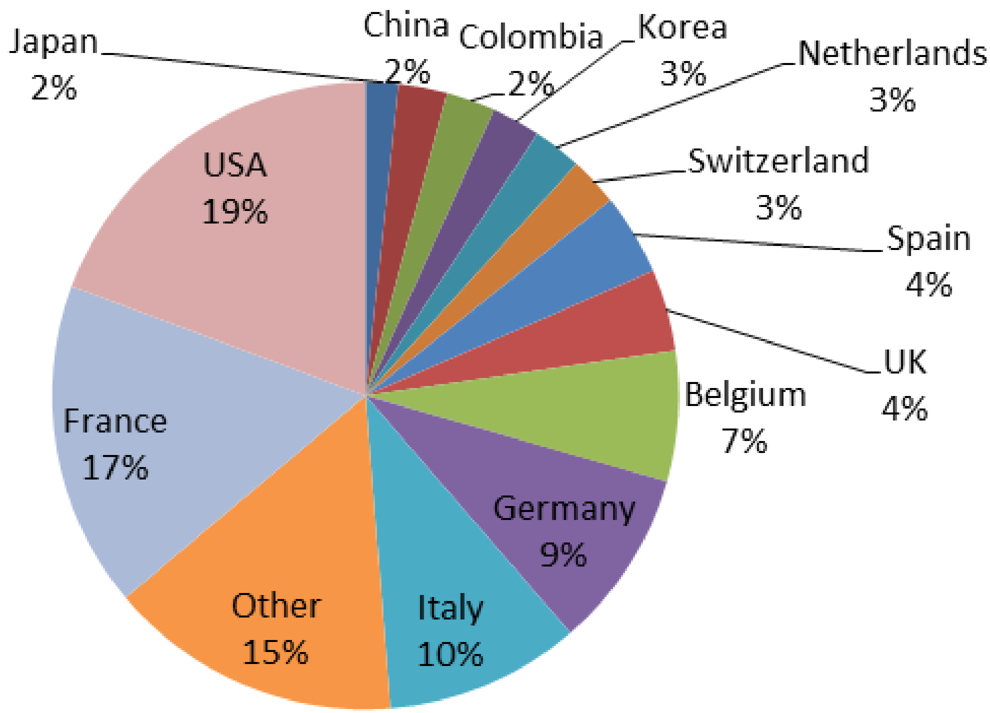


Figure 2: Country of provenience for the participants to the survey.

softwares can be difficult to compare, because their requirements in terms of architecture, dependencies etc. are different; finally, not all softwares are well-documented and easy to test in the same way, so non-experienced users may not know all the tweaks to boost simulations. We compensate the lack of objective experimental comparison with the user feedback provided by this survey.

2 Survey overview

The analysis of the survey is reported hereinafter.

2.1 About the participants

The survey was filled by 119 participants (92% male, 8% female; age 32 ± 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 2). 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]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 its 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	**13%**	7%	3%	18%	10%	34%	**15%**
ODE	**11%**	12%	5%	18%	**22%**	22%	**10%**
Bullet	5%	13%	7%	12%	10%	29%	24%
V-Rep	5%	3%	3%	18%	3%	29%	39%
Webots	4%	7%	1%	16%	**13%**	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%	**15%**
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%	**88%**
Moby	2%	1%	0%	0%	2%	14%	81%
Vortex	3%	2%	0%	5%	5%	17%	68%
RoboRobo	3%	1%	0%	0%	1%	4%	**91%**

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

2.2 General knowledge about simulating tools

We asked participants to indicate their familiarity with some of the most common existing simulation tools. We provided a list of existing software tools for simulations, used in different contexts. We asked the users to indicate whether the software was currently used or 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%), OpenRave (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".

[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.

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.

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 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)

2.5 Currently used tools

We asked participants to indicate the current simulation tool they are using. Results are shown in Figure 3. 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:

Rank	Most important criteria
1	Simulation very close to reality
2	Open-source
3	Same code for both real and simulated robot
4	Light and fast
5	Customization
6	No interpenetration between bodies

Table 3: Most important criteria for choosing a simulator.

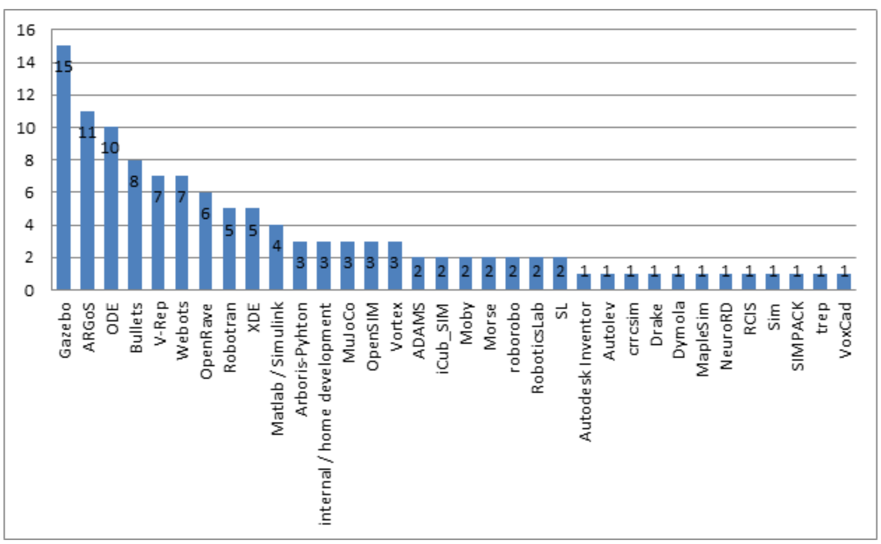


Figure 3: The simulation tools currently in use among the participants to the survey. The vertical axis reports the number of users that indicated the tool as their principal.

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.

- **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,

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	4.73 ± 0.45	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	4.75 ± 0.46	4.00 ± 0.76	3.75 ± 0.71	4.37 ± 0.74	3.87 ± 0.83	3.96 ± 0.78
V-Rep	4.28 ± 0.76	4.43 ± 0.79	4.71 ± 0.76	4.14 ± 0.90	4.28 ± 0.76	4.43 ± 0.53	4.14 ± 1.07	4.25 ± 0.80
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	2.67 ± 1.15	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	1.67 ± 0.58	4.33 ± 1.15	3.33 ± 1.15	4.67 ± 0.57	4.00 ± 0.00	5.00 ± 0.00	3.62 ± 0.66
XDE	1.40 ± 0.55	2.80 ± 1.09	3.60 ± 0.55	2.80 ± 1.09	3.40 ± 1.10	2.80 ± 0.84	3.00 ± 1.00	2.83 ± 1.07

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

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.

2.6 Tools for robots

The majority of participants to the survey is using the software tool to simulate robots (91%). We extracted the principal tools used for simulating the main robots:

- iCub: 25% Arboris-Python, 17% ODE, 17% Robotran, 17% iCub_SIM
- Atlas: 50% Gazebo, 25% MuJoCo, 12% Autolev, 12% Drake
- PR2: 21% OpenRave, 14% Gazebo, 14% MuJoCo, 7% Bullet, 7% V-Rep
- Multi-legged robot: 22% ODE, 11% SL, 11% Bullet, 11% Webots
- Wheeled vehicle: 14% Gazebo, 14% V-Rep, 11% ARGoS, 7% Morse, 7% Webots, 7% Vortex
- Quadrotor: 24% Gazebo, 24% ARGoS, 12% V-Rep

3 Software information fiches

We report in the following some essential information for the main software tools (the most diffused) that may be of help for the interested reader. Most of the information gathered here is extracted from the survey (each item is marked by a filled dot, •). When it is not the case, an empty dot ◦ is used. For the subjective user feedback we refer the reader to the full report of the survey. Data are reported with %, however to have a fair comparison we report in brackets the number of participants that selected the specified tool. Note that in the following "main simulated robots" refers to real robots that are simulated in the software.

3.1 Gazebo

Gazebo is a multi-robot simulator for outdoor environments, developed by Open-Source Robotics Foundation. It is the official software tool for the DRC. It supports multiple physics engines (ODE, Bullet).

- ◦ Web: `http://gazebosim.org/`
- ◦ License: Apache 2
- • Survey participants: 15
- • OS share: 100% GNU/Linux
- • Main API: 80% C++
- • Main reason for adoption: 53% best tool upon evaluation, 20% software already used in the lab, 20% official tool for a challenge, 7% open-source
- • Mostly used in USA (33%)
- • Mainly used for: 33% mobile robotics, 27% service robotics, 20% humanoid robotics
- • Main simulated robots: 40% Atlas, 33% custom platform, 27% wheeled vehicle, 27% quadrotor, 27% turtlebot, 20% PR2
- • Main middleware used with: 93% ROS
- • Main simulated robots: 40% Atlas, 33% custom platform, 27% wheeled vehicle, 27% quadrotor, 27% turtlebot, 20% PR2

3.2 ARGoS

ARGoS is a multi-robot, multi-engine simulator for swarm robotics, initially developed within the Swarmanoid project[5].

- ◦ Web: `http://iridia.ulb.ac.be/argos/`
- ◦ License: GPLv3.0
- • Survey participants: 11
- • OS share: 91% GNU/Linux, 9% MAC OSX
- • Main API: 73% C++
- • Main reason: 45% software already used in the lab, 27% colleagues using it
- • Mostly used in Belgium (36%) and Italy (27%)
- • Used for: 46% mobile robotics, 36% swarm robotics, 18% flying robots
- • Main simulated robots: 64% khepera/e-puck/thymio, 36% marXbot/footbot, 27% quadrotor

[5] `http://www.swarmanoid.org/`

3.3 ODE

ODE (Open Dynamics Engine) is an open-source library for simulating rigid body dynamics, used in many computer games and simulation tools. It is used as physics engines in several robotics simulators, such as Gazebo and V-Rep.

- Web: `http://www.ode.org/`
- License: GNU LGPL and BSD
- Survey participants: 10
- OS share: 100% GNU/Linux
- Main API: 80% C++
- Main reason: 50% best tool upon evaluation, 20% used before, 10% boss choice, 10% open-source, 10% software already used in the lab
- Mostly used in France (20%)
- Used for: 50% humanoid robotics, 20% multi-legged robotics, 20% snake robots, 10% numerical simulation of physical systems
- Main simulated robots: 40% multi-legged robot, 20% iCub

3.4 Bullet

Bullet is an open-source physics library, mostly used for computer graphics and animation. The latest release[6] also supports Featherstone's articulated body algorithm and a Mixed Linear Complementarity Problem solver, which makes it suitable for robotics applications.

- Web: `http://bulletphysics.org`
- License: ZLib license, free for commercial use
- Survey participants: 8
- OS share: 50% Windows, 38% GNU/Linux, 12% MAC OSX
- Main API: 75% C++
- Main reason: 63% best tool upon evaluation, 25% open-source, 12% colleagues using it
- Mostly used in France (25%), Italy (25%) and Belgium (25%)
- Used for: 25% humanoid robotics, 25% numerical simulation of physical systems, 12.5% industrial manipulators, 12.5% human motion analysis, 12.5% mobile robotics, 12.5% multi-legged robotics
- Main simulated robots: 25% multi-legged robot

[6] At the time we are submitting this paper, the latest version is 2.82, released at the end of october 2013 - after the survey.

3.5 V-Rep

V-Rep is a robot simulator software with an integrated development environment, produced by Coppelia Robotics. Like Gazebo, it supports multiple physics engines (ODE, Bullet, Vortex).

- ◦ Web: `http://www.coppeliarobotics.com/`
- ◦ License: Dual-licensed source code: commercial or GNU GPL
- Survey participants: 7
- OS share: 57% GNU/Linux, 43% Windows
- Main API: 57% C++, 29% LUA
- Middleware: 43% ROS, 57% None
- Main reason: 72% best tool upon evaluation, 14% colleagues using it, 14% boss choice
- Used for: 29% mobile robotics, 14% industrial manipulators, 14% humanoid robotics, 14% mechanical design, 14% cognitive architectures, 14% service robotics
- Main simulated robots: 29% Nao, 29% quadrotor, 29% wheeled vehicle, 29% Bioloid, 29% khepera/ e-puck/ thymio

3.6 Webots

Webots is a development environment used to model, program and simulate mobile robots developed by Cyberbotics Ltd.

- ◦ Web: `http://www.cyberbotics.com`
- ◦ License: Commercial or limited features free academic license
- Survey participants: 7
- OS share: 57% GNU/Linux, 29% Windows, 14% MAC OSX
- Main API: 71% C++
- Main reason: 29% best tool upon evaluation, 29% software already used in the lab, 14% boss choice, 14% official tool for a challenge, 14% used before
- Used for: 43% mobile robotics, 43% multi-legged robotics, 14% humanoid robotics
- Main simulated robots: 29% KUKA LWR, 29% Lego Mindstorm, 29% wheeled vehicle

3.7 OpenRave

OpenRave is an environment for simulating motion planning algorithms for robotics.

- ◦ Web: `http://openrave.org/`
- ◦ License: LGPL and Apache 2
- Survey participants: 6
- OS share: 100% GNU/Linux
- Main API: 83% python

- Main reason: 50% best tool upon evaluation, 33% colleagues using it, 17% boss choice
- Mostly used in USA (33%)
- Used for: 50% humanoid robotics, 50% service robotics
- Main simulated robots: 50% PR2

3.8 Robotran

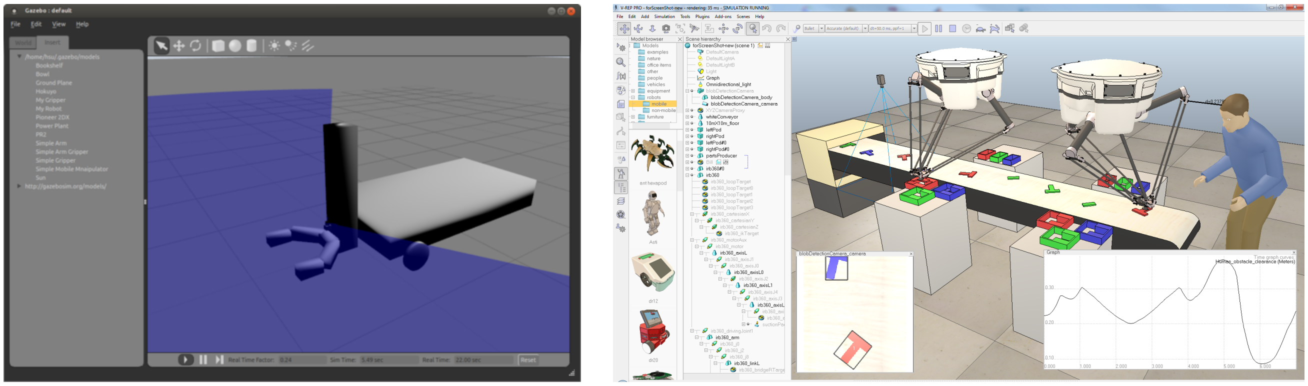
Robotran is a software that generates symbolic models of multi-body systems, which can be analysed and simulated in Matlab and Simulink. It is developed by the Center for Research in Mechatronics, Université Catholique de Louvain.

- ○ Web: `http://www.robotran.be/`
- ○ License: commercial and free non commercial license
- Survey participants: 5
- OS share: 80% Windows, 20% GNU/Linux
- Main API: 60% C
- Main reason: 40% software already used in the lab, 20% best tool upon evaluation, 20% developer, 20% open-source (free)
- Used only in Belgium (40%) and Italy (60%)
- Used for: 60% humanoid robotics, 20% human motion analysis, 20% flying robots
- Main simulated robots: 60% Coman, 40% iCub

3.9 XDE

XDE is an interactive physics simulation software environment fully developed by CEA LIST.

- ○ Web:
  `http://www.kalisteo.fr/lsi/en/aucune/a-propos-de-xde`
- ○ License: Commercial and free non commercial license
- Survey participants: 5
- OS share: 60% GNU/Linux, 40% Windows
- Main API: 100% python
- ○ Middleware: OROCOS
- Main reason: 40% boss choice, 40% software already used in the lab, 20% developer
- Used only in France (100%)
- Used for: 40% humanoid robotics, 20% industrial manipulators, 20% numerical simulation of physical systems, 20% human motion analysis
- Main simulated robots: 40% industrial robots, 40% KUKA LWR, 20% iCub, 20% wheeled vehicle



(a) Gazebo. (b) V-Rep.

Figure 4: The simulation environment of Gazebo and V-Rep (credits: `http://gazebosim.org` and `http://www.coppeliarobotics.com`).

4 Conclusions

With the growing interest of robotics for physical interaction, simulation is no longer a tool for offline computation and visualization, but is used in particular for rapidly prototyping controllers. That is why researchers stressed the importance of more realistic simulation, same code for both real and simulated robot, beside the availability of the source code.

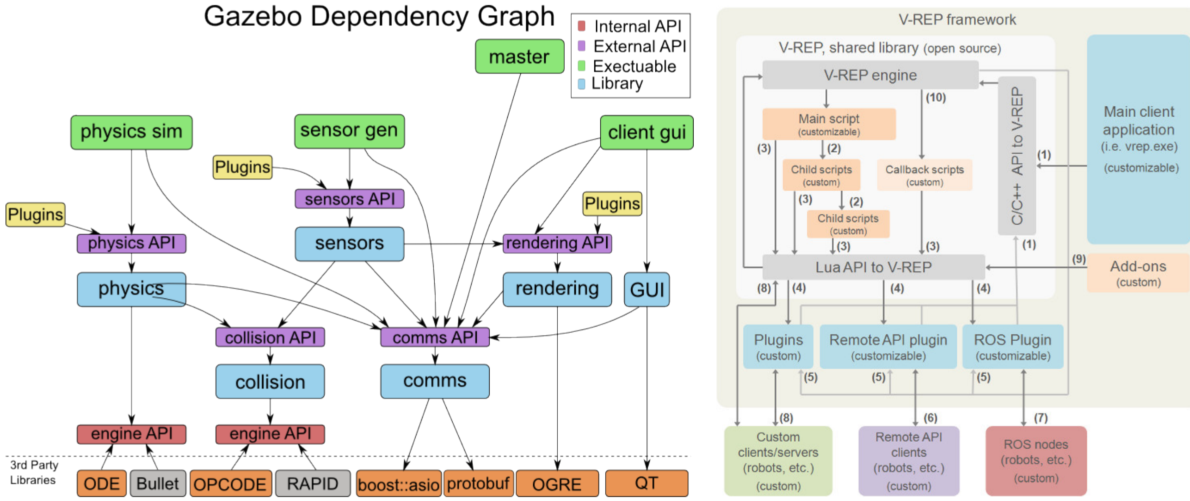
This shift in the expectations from simulation reflects in the migration from physics engines classically used for animation of virtual characters and computer graphics towards physics engines supporting robotics descriptions of bodies and more contact solvers. The users' knowledge of multiple simulation tools and their activity in testing and abandoning eventually a tool, suggest that users look for the right tool that meets their requirements and is fit for their problem. For instance, the robotics community demands physics engines with direct support of robotics descriptions of multi-body systems. This is the reason why Bullet is now supporting LCP solvers and Featherstone's ABA, and new physics engines like MuJoCo[7] or Vortex have been created.

A good compromise is a modular software that supports multiple physical engines, enabling a tradeoff between simulation accuracy and computational resources. Those features, together with the stability of the simulation, are of main concern for the users. This strategy, adopted with Gazebo by the research community and with V-Rep at industrial level, seems to pay off in terms of user feedback, because the first is the most diffused among the survey participants and the second the best rated. Subjective free-comments[8] reported that users of those tools, though acknowledging their current limitations, were confident in the announced developments that could sensibly improve the tools.

To conclude, we overviewed the panorama of simulation tools that are currently used in robotics. Each software inherits its specificities from the expected domains of application or the original application for which is was conceived, which results in a variety of tools with different features ranging from dynamic solver libraries to systems simulation software. More recent tools, like Gazebo and V-Rep, have the potential to be of general use thanks to their good support and community and the support of different physical engines. Notwithstanding, we remind that designing a perfect physics engine is impossible and there will always be a difference between simulation and reality, a gap that should be taken into account by the simulator and the robot controllers [14].

[7] MuJoCo is not merely a physics engine, it incorporates control and optimization modules.

[8] They can be read in the extended version of the survey report: `http://www.codyco.eu/survey-simulation`.



(a) Gazebo (b) V-Rep

Figure 5: A graphical representation of the software architectures of Gazebo and V-Rep (credits: `http://gazebosim.org` and `http://www.coppeliarobotics.com`).

Acknowledgment

The authors are supported by the EU Project CODYCO (FP7-ICT-2011-9, No. 600716).

References

[1] 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.

[2] 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.

[3] ——, "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.

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

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

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

[7] 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.

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

[9] 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.

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

[11] 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.

[12] 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.

[13] V. Tikhanoff, A. Cangelosi, P. Fitzpatrick, G. Metta, L. Natale, and F. Nori, “An open-source simulator for cognitive robotics research: the prototype of the icub humanoid robot simulator,” in *8th Workshop on Performance Metrics for Intelligent Systems*, 2008, pp. 57–61.

[14] J.-B. Mouret, S. Koos, and S. Doncieux, “Crossing the reality gap: a short introduction to the transferability approach,” in *"Evolution in Physical Systems" Workshop in ALIFE*, 2012.