



Tools for simulating humanoid robot dynamics: a survey based on user feedback

Serena Ivaldi, Jan Peters, Vincent Padois, Francesco Nori

► To cite this version:

Serena Ivaldi, Jan Peters, Vincent Padois, Francesco Nori. Tools for simulating humanoid robot dynamics: a survey based on user feedback. IEEE-RAS International Conference on Humanoid Robots (Humanoids), 2014, Madrid, Spain. hal-01116148

HAL Id: hal-01116148

<https://hal.science/hal-01116148>

Submitted on 12 Feb 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Tools for simulating humanoid robot dynamics: a survey based on user feedback

Serena Ivaldi[†], Jan Peters[‡], Vincent Padois^{*} and Francesco Nori[§]

Abstract—The number of tools for dynamics simulation has grown substantially in the last few years. Humanoid robots, in particular, make extensive use of such tools for a variety of applications, from simulating contacts to planning complex motions. It is necessary for the humanoid robotics community to have a systematic evaluation to assist in choosing which of the available tools is best for their research. This paper surveys the state of the art in dynamics simulation and reports on the analysis of an online survey about the use of dynamics simulation in the robotics research community. The major requirements for robotics researchers are better physics engines and open-source software. Despite the numerous tools, there is not a general-purpose simulator which dominates the others in terms of performance or application. However, for humanoid robotics, Gazebo emerges as the best choice among the open-source projects, while V-Rep is the preferred commercial simulator. The survey report has been instrumental for choosing Gazebo as the base for the new simulator for the iCub humanoid robot.

I. INTRODUCTION

Dynamics simulation is crucial for humanoid robotics, especially for motion optimization, rapid prototyping of controllers and verification in a simulated environment before execution on the real robot. Confronted with the problem of choosing a new simulator for the humanoid robot iCub, we have investigated the current status of dynamical simulation in robotics by means of an extensive online survey.¹ The purpose was to get feedback about the available tools and retrieve new elements to ponder which of the existing tools was the best for our research.

We realized that the number of software tools for dynamics simulation available to the robotics community has grown in the last few years. Nevertheless, to our knowledge a comparison of such tools does not exist. We reckon it is difficult to make helpful quantitative comparisons: different simulators usually have different features and requirements, and their performance cannot always be evaluated on the same machine. In addition, they are frequently tailored to solve a particular problem and are not general-purpose. Sometimes the underlying knowledge of the code or the numerical algorithms is not accessible: non-experienced users may not know all the tweaks to boost simulations, and we know

how “magic numbers” (e.g., integration stepsize, collision thresholds, just to name a few), tuning and experience can change the outcome of a simulation. In short, experimental comparisons are difficult to obtain.

In contrast, it is possible to find performance comparisons of middleware for robotics (ROS, YARP, OROCOS, Player, etc.) [1] and contact models [2], [3], because their corresponding software is open-source or cross-platform, well documented and customizable.

The same features (open-source and cross-platform compatibility, customization and documentation) should be supported by all tools for dynamics simulation to enable comparisons on test problems, but in practice it is not the case.

Here, we compensate for the lack of objective and quantitative comparisons by presenting the current status of dynamics simulation together with the results of our survey, and report on user feedback for the most diffused tools in the robotics community. User feedback can provide useful insights on the effective use of a tool based upon experience, and can help the researcher to pick the best tool for his needs.

We are aware that this comparison barely scratches the surface of the problem. Nevertheless, as no such a report on the state of the art is available to the community, it appears an important and needed assessment.

In this paper we present the state of the art of tools for simulating robot dynamics and discuss the results of our survey. We briefly introduce the most prominent tools in Section II. We report on the features that matter to the robotics community and highlight the results of the survey in Section III. In particular, we report on the diversity of existing tools and the fact that no single simulator dominates the others for research applications.

II. SIMULATION TECHNOLOGIES

The technologies used for simulating robot dynamics originated in the computer graphics community. Most of the currently used tools still rely on ODE [4] or Bullet [5] as underlying physics engines, which were originally designed for video-games, and hence do not address all robotics needs. In this section, we first give an overview on the currently available software tools for simulation of robot dynamics. Subsequently, we introduce the user experience based survey.

A. Challenges for robotics simulation

Dynamics simulators for robotics have stricter requirements than animation of virtual characters, where time, computational and physical accuracy can be less constraining. In entertainment (e.g. video-games), unfeasible forces may

E-mail: ivaldi@ias.tu-darmstadt.de

[†] S. Ivaldi and J. Peters are with the Intelligent Autonomous Systems Lab, TU Darmstadt, Germany.

[‡] J. Peters is also with the Max Planck Institute for Intelligent Systems.

^{*} V. Padois is with the Institut des Systèmes Intelligents et de Robotique (ISIR), CNRS, UMR 7222 and Sorbonne Universités, UPMC, France.

[§] F. Nori is with the Robotics, Brain and Cognitive Sciences Dept., Italian Institute of Technology.

¹Online survey: <http://goo.gl/Tmyf5A>

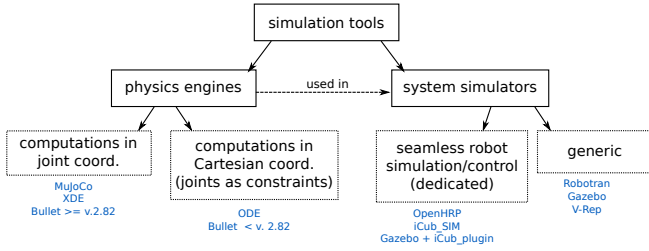


Fig. 1: A practical classification of the simulation tools (see Section II-B). *Physics engines* can be discriminated by the way they represent rigid-body structures: on the left we have software supporting natively the parametrized rigid-body dynamics representation that is common to the robotics community (e.g., MuJoCo, XDE); on the right, software inherited from the computer graphics community (e.g., ODE). *System simulators* build upon physics engines, but usually provide user-friendly interfaces, GUIs and additional features like sensor simulation. System simulators can be generic or platform-specific, providing interfaces for seamless control of simulated and real robot (e.g., HRP, iCub).

not be a problem since the laws of physics can be violated. In (bio-)mechanical studies, simulators can be used offline to analyze or synthesize behaviors. Despite that the field of dynamics modeling and simulation has substantially matured in the last decades [6], [7], [8], the control of whole-body movements of complex robots, such as humanoids, poses additional challenges to simulators.

A key issue is numerical stability, which poses strong limitations on the use of simulations in real-time control settings [2], [3]. To be useful as a predictive engine in real-time control loops [9], the simulator must be very fast at computing the dynamics and needs to guarantee convergence to physically feasible solutions within a limited time frame [10].

The simulation of rigid and soft bodies in contact with rigid and compliant environments [11], [12] has critical impacts on simulated interactions between the robot and the environment (for example, locomotion on various terrains such as concrete or grass). Supporting different types of contacts, for example with deformable materials, compliant and soft surfaces [13], is fundamental to optimize robot controllers to different environments and objects. Inaccurate computation of contact forces between bodies may result in unrealistic contacts or physically unfeasible contact forces and then in unrealistic behavior.

A further requirement is the capability to model and simulate new types of actuation systems, such as variable impedance or soft actuators [14]: this feature is fundamental for designing controllers for novel hardware and at the same time building transparent interfaces for writing the same code for the simulated and the real robot.

B. Physics engines and system simulators

Figure 1 presents a descriptive classification of the dynamics simulation tools. We can distinguish between *physics*

engines (e.g. ODE, Bullet) and more complex software, that we call here “*system simulators*” (e.g. Gazebo, V-Rep, iCub_SIM) that are based on a physics engine but also include simulation of sensors and robotic interfaces.

Physics engines can be discriminated by the way they represent rigid-body structures: on one hand we have software tools like ODE, which represents joints as constraints between bodies; on the other we have software like XDE, OpenHRP, which make use of parametrized rigid-body dynamics representations, where joints are simply part of the robotics structure. The second group benefits from the straightforward computation of quantities that recur in robot control, such as Jacobians, mass matrices etc. The critical difference between the two classes is in 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 general, finding the correct contact forces can be burdensome. Contact modelling and simulation is an area of research on its own, which is actively explored improving contact models [3] and solvers [9].

System simulators make use of physics engines of course, to simulate the dynamics of the bodies in the environment, but also provide additional features, such as sensor simulation, model editors or interaction with the operator. In some cases they also provide specific interfaces that facilitate seamless simulation and control of the robot and its virtual character. For example, they emulate the driver interfaces of the real robot in such a way that code running for the simulators can be switched to the robot at no cost. This makes them in practise the “official” tools for some platforms: for example, OpenHRP for the HRP robot series, iCub_SIM [15] and the more recent Gazebo plugin for iCub [16].

C. Assorted software tools

There is a great diversity of simulators, and while preparing the survey we discovered a considerable number of software projects (we easily found more than 40 tools, but we discovered through the survey and later a lot of “home developers” that are not sharing nor advertising their solutions, so the number of active software projects is probably higher). We hereby list some of the most prominent ones.

ODE (Open Dynamics Engine) [4] is an open-source library for simulating rigid body dynamics, with a built-in collision detector. It is one of the most known multi-purpose rigid-body physics engines, used in many computer games and embedded in several simulation tools. **PhysX** [17] is an engine developed by NVIDIA for enabling real-time physics in video-games. It is optimized for GPU computations. **Bullet** [5] is another open-source physics library, mostly used for computer graphics and animation. As ODE and NVidia PhysX, it was a game-oriented engine, that enforced joints constraints numerically. The latest release (v.2.82) also supports Featherstone’s Articulated Body Algorithm [6] and

a Mixed Linear Complementarity Problem. These make it more suitable for robotics applications, since dynamics is solved in joint coordinates and contacts resolution is more stable. **DART** [18] is another open-source physics engine, used for robotics and computer graphics. **Gazebo** [19] is a multi-robot simulator for outdoor environments, developed by the Open-Source Robotics Foundation. It is the official software tool for the Darpa Robotics Challenge. It supports multiple physics engines (ODE, Bullet, DART) and, thanks to its modular and plugin-based structure, can be easily extended with new features. The project and its community are very active. **ARGoS** [20] is a multi-robot, multi-engine simulator for swarm robotics, initially developed within the Swarmanoid project. **V-Rep** [21] is a robot simulator software with an integrated development environment, produced by Coppelia Robotics. Like Gazebo, it supports multiple physics engines (ODE, Bullet, Vortex). **Webots** [22] is a simulation and development environment developed by Cyberbotics Ltd. It uses ODE as physics engine. It is particularly dedicated to mobile robotics applications and also provides interfaces for controlling some robots (e.g., e-puck, DARwIn-OP). **OpenRave** [23] is an environment for simulating motion planning algorithms for robotics. It contains several models of industrial robots and targets robotics automation. **Robotran** [24] 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. **Vortex Dynamics** [25] is a software developed by CM Labs, specialized in simulating contact dynamics in different operating environments (e.g., terrain, water). It is coupled with a 3D editor for creating mechanisms, robots and scenes, and is particularly adapted to simulation of vehicles and cable systems. **OpenSim** [26] is a toolkit for musculoskeletal modeling and dynamic simulation of movement, developed at Stanford University and supported by the US NIH and by DARPA. It is freely available, open-source, and extensible through user plugins. The physics engine of this project is **SimBody** [27], an open-source C++ API implementing Featherstone's algorithms for rigid body mechanics, with support of different contact models. **MuJoCo** [28] is a dynamics engine mostly developed by E. Todorov and now property of Roboti LLC. It is one of the most recent physics engines, conceived for simulating robotics and biomechanical systems. It supports parallel computations, provides inverse dynamics with contacts and equality constraints, implements several contact dynamics. It is suitable for control optimization and can be used in real-time within a control loop [8]. **XDE** [29] is an interactive physics simulation software environment fully developed by CEA LIST, used for virtual reality applications in industrial contexts. **MOBY** [30] is an open-source physics simulation library mainly developed by E. Drumwright. It is a multi-body dynamics simulation library, with several features for accurate simulation of robot dynamics: multiple integrator types, two convex solvers, support for deformable bodies, several contact models. **OpenHRP** [31] is a system simulator

developed in Japan for the HRP robots. Interestingly, it consists of several modules (a dynamics simulator, a control interface with the robot, a collision detector, ...) and can be used for a seamless simulation/control of the robot.

III. THE USERS POINT OF VIEW

We hereby report on the results of the online survey about the use of tools for simulating robot dynamics. The survey was filled by 119 participants (92% male, 8% female; age 32 ± 6 , min 20, max 57) among whom 62% holds a PhD degree and 35% a BS or MS degree, mostly from USA (19%), France (17%), Italy (10%) and Germany (9%)².

The participants work mostly in University (70%) or do R&D in public (16%) or private (14%) institutes. They mostly work in GNU/Linux systems (66%)³, code in C++ (52%)⁴ on a powerful desktop (39%)⁵. The simulation tool they use is preferably open-source (67%)⁶ and only half of them is using a robotics middleware⁷.

They mostly work on control and locomotion (21% and 14%)⁸ for humanoid and mobile robots (26% and 20%)⁹. Notably, 8% of the participants (10 people) are competing in the Darpa Robotics Challenge (DRC).¹⁰

For their research, they mostly simulate the robot (physical) interaction with the environment, locomotion and navigation, in particular to test controllers and simulate behaviors before going on the real robot.¹¹

We 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 rating is reported in Table I. The stability of simulation is the only element that was evaluated as "very important", whereas speed, precision and accuracy of contact

²More information can be found in the survey report [32].

³**Primary OS:** 66% GNU/Linux, 30% Windows, 4% MAC OSX

⁴**Primary API language:** 52% C++, 18% python, 13% Matlab, 8% C, 3% LUA, 2% Java; 3% of participants do not use an API

⁵**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

⁶**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).

⁷**Middleware:** 52% is not using the tool with a middleware, the remainder is using ROS (25%), YARP (6%), OROCOS (4%).

⁸**Primary areas of research:** 21% control, 14% locomotion, 10% machine learning, 9% HRI, 8% planning, 6% mechanical design, 5% cognitive robotics, 5% mathematical modeling.

⁹**Primary application field:** 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.

¹⁰Interestingly, 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).

¹¹**Main purposes of dynamics simulation** (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.

resolution were marked important. Remarkably, the same API between real and simulated robot is also signed as important, which points to the need of seamless software interfaces for controlling virtual and real robot.

We asked participants to indicate **the most important criteria for choosing a simulator**. User selections are reported in Table II. The most important criteria are a realistic simulation (close to reality) and the open-source license of the software. This points to good physics engines and shared tools by the community. On a second level, fast simulations and seamless interfaces for writing the same code for simulated and real robot. This further confirm that simulators are used to prototype and optimize controllers: speed is important, as well as the fact of not having to rewrite code when switching from one system to the other.

#	Feature	Median rating
1	Stability of simulation	5 (4.50 $\pm$ 0.58)
2	Speed	4 (4.05 $\pm$ 0.75)
3	Precision of simulation	4 (4.02 $\pm$ 0.71)
4	Accuracy of contact resolution	4 (3.91 $\pm$ 0.92)
5	Same interface betw. real & simulated system	4 (3.67 $\pm$ 1.26)
6	Computational load (CPU)	3 (3.53 $\pm$ 0.85)
7	Computational load (memory)	3 (3.22 $\pm$ 0.90)
8	Visual rendering	3 (3.02 $\pm$ 1.02)

TABLE I: Most important features for a simulator.

#	Most important criteria	Users
1	Simulation very close to reality	32%
2	Open-source	24%
3	Same code for both real and simulated robot	19%
4	Light and fast	11%
5	Customization	6%
6	No interpenetration between bodies	3%

TABLE II: Most important criteria for choosing a simulator.

Tool	Currently used, and it's 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 III: Knowledge and past/present use of simulators.

A. A diversity of tools

As briefly discussed in Section II-C, there is an abundance of existing dynamics simulators, many inherited from computer graphics and virtual characters animation. In particular, many simulation software "clones", that basically wrap over existing physics engines (especially ODE and Bullet) providing a nicer IDE and GUI.

Are robotics researchers aware of the panorama of the available tools to the community? We asked the participants to indicate their familiarity with some existing simulation tools. We provided a list of more than 40 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 III.

We found that the **most currently used main tools** (i.e., tools that have more than 5% of positive answers to the fact that the software is currently used) are Gazebo (13%), 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. Interestingly, more than one simulation tool is being tested/used at the same time, which explains why many tools have been tested and abandoned or adopted. It has to be noticed that these results are biased by the "age" of the software tools, a parameter that we do not consider in our analysis as we are interested in getting a snapshot of the current status of the tools diffused among researchers. 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 III. 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%). This means that all these software tools have been used at some point, then probably replaced by a more suitable tool (or more than one). 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 software that could have been the "one among many" tools that have been used by researchers; 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" (i.e., their release date). The replacement tool is probably (but not necessarily) chosen among the pool of the "least unknown": ODE (10% of participants never heard about it), Gazebo (15%), Blender, Bullet (24%), Webots (27%), NVidia PhysX (32%), Stage (38%), V-Rep (39%), OpenSim (40%), ADAMS (45%). Interestingly, the first three are also open-source projects.

B. The main 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 have less than 4% of user share. We report in Table IV some essential information about the main software tools (i.e., the most diffused). Interestingly, the most diffused software tools are open-source (green colored cells).

As a complement to objective comparisons, we asked for **user ratings**, to avoid inappropriate choices of software and provide useful suggestions to the developers community. We 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 V reports the mean and standard deviation of the notes received by the users of each tool. Remarkably, the overall evaluation of their tool was indicated as positive,¹² but in free comments almost all participants asked for more documentation and better physics simulation.¹³

We asked therefore why did they adopt that particular tool over others. Overall, the main **reasons why they choose the tool** are because it was the best tool for their research upon evaluation (29%) or for "inheritance", because it was "the software" (already) used in their laboratory (23%). Interestingly there is quite a demarcation between the first reasons and the others.¹⁴ Only 3% of the participants chose the tool because of a robotics challenge. As we can see in Table VI, 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 Vortex (66%). In some cases, like for XDE, it was also a choice imposed by the project leader (40%).

C. A diversity of tools: too many?

User feedback can also be useful for pointing a researcher to a community that is actively using a particular tool and sharing the same concerns. Some area of research have specific needs, which are probably not addressed by all tools. For example, it is likely that people simulating flying robots have different needs than those simulating wheeled robots or humanoid bipeds.

A first observation is that there is not a real winner among the software tools, i.e. there is not a single simulator that

¹²We 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).

¹³Some of the survey participants indicated the main issues and the desirable features of their tools in the free comment space. The complete comments can be viewed in the survey extended report [32]. On average, the most desired features for their tools are better documentation and better physics simulation, especially contact models.

¹⁴**Main reason for choosing the tool:** 29% 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, 3% official tool of a robotics challenge.

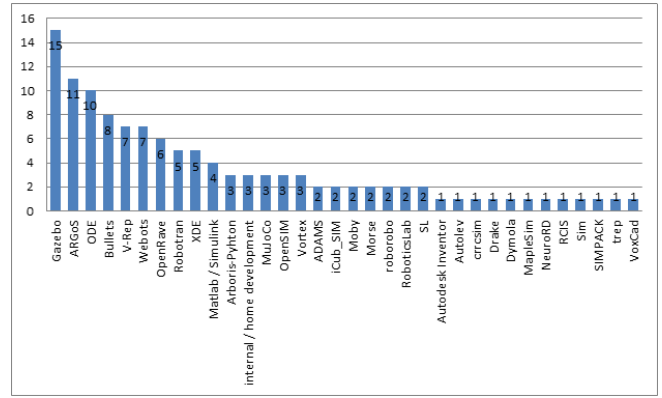


Fig. 2: The simulation tools currently in use among the participants to the survey. The vertical axis reports the number of users indicating the tool as the main one in use.

dominates the others in terms of user share, rating or features. Not even in terms of research application. Table VII reports the main applications and the main simulated robots for the most diffused tools. We highlighted the cells where there is a significant use of the tools for simulating humanoid robots: Gazebo, ODE, Bullet, Robotran, XDE and MuJoCo. We extracted the most used tools for a selection of research areas: results are shown in Table VIII. 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, which stand out of a variety of other simulators. It is interesting to notice that Gazebo supports ODE and Bullet as physical engines, so probably 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.

A second observation is that robots generally do not have "their" own simulator. The majority of participants to the survey is using the software tool to simulate robots (91%). Users could point out more than one simulated robot. The most simulated robot are wheeled vehicles (28), multi-legged robots (18), quadrotors (17), KUKA LWR (15), PR2 (14), iCub (13), Atlas (10), khepera/e-puck (10). We report in Table IX the simulation tools for a selection of robots. We highlight the cells corresponding to humanoid robots. Atlas is the only robot that has a major simulation software (Gazebo).

D. A use-case: the iCub simulator

The iCub community recently faced the problem of choosing a new tool for simulating whole-body dynamic movements with multiple contacts. The "official" simulator iCub.SIM [15] is based on ODE and, from our experience, unfit for simulating such tasks. An alternative simulator based on XDE was more convenient, but was not open-source. Our survey was instrumental to choose the new simulating platform for the iCub project: Gazebo. Besides the technical reasons, it was chosen because it is open-source, its community is active, and because it supports multiple physics

Tool	Ref.	Phys. Sim.	Sys. Sim.	License	OS (% upon survey participants)	API (% survey part.)	Middleware (% survey part.)
Gazebo	[19]		X	Apache 2	100% GNU/Linux	80% C++	93% ROS
ARGoS	[20]		X	GPLv3.0	91% GNU/Linux, 9% MAC OSX	73% C++	-
ODE	[4]	X		LGPL and BSD	100% GNU/Linux	80% C++	-
Bullet	[5]	X		ZLib license	50% Windows, 38% GNU/Linux, 12% MAC OSX	75% C++	-
V-Rep	[21]		X	Dual-licensed source code: commercial or GNU GPL	57% GNU/Linux, 43% Windows	57% C++, 29% LUA	43% ROS
Webots	[33], [22]		X	Commercial or limited features free academic license	57% GNU/Linux, 29% Windows, 14% MAC OSX	71% C++	-
OpenRave	[23], [34]		X	LGPL and Apache 2	100% GNU/Linux	83% python	-
Robotran	[24]	X		Commercial and free non commercial license	80% Windows, 20% GNU/Linux	60% C	-
Vortex	[25]	X	X	Commercial	66% GNU/Linux, 33% Windows	100% C++	33% ROS
OpenSIM	[26], [35]	X	X	Apache 2	66% Windows, 33% GNU/Linux	66% Matlab, 33% C++	33% ROS, 33% YARP
MuJoCo	[28], [9]	X		Free academic license	100% Windows	66% C, 33% Matlab	-
XDE	[29]	X	X	Commercial and free non commercial license	60% GNU/Linux, 40% Windows	100% python	100% OROCOS

TABLE IV: Information about the most diffused software tools - percentages refer to the use reported by the survey participants. The colored cells highlight whether the software is open-source or not.

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

Tool	Best tool upon evaluation	Software already used in the lab	Used happily by colleagues	Official tool for a challenge	Open-source	Boss choice	I'm developer	I used it before
Gazebo	47% (7)	20% (3)	-	20% (3)	13% (2)	-	-	-
ARGoS	54% (6)	-	-	9% (1)	9% (1)	9% (1)	9% (1)	9% (1)
ODE	50% (5)	10% (1)	-	-	10% (1)	10% (1)	-	20% (2)
Bullet	63% (5)	-	12% (1)	-	25% (2)	-	-	-
V-Rep	72% (5)	-	14% (1)	-	-	14% (1)	-	-
Webots	29% (2)	29% (2)	-	14% (1)	-	14% (1)	-	14% (1)
OpenRave	50% (3)	-	33% (2)	-	-	17% (1)	-	-
Robotran	20% (1)	40% (2)	-	-	20% (1) *	-	20% (1)	-
Vortex	-	66% (2)	-	-	-	33% (1)	-	-
OpenSIM	66% (2)	-	-	-	-	-	-	33% (1)
MuJoCo	-	33% (1)	-	-	33% (1) *	-	33% (1)	-
XDE	-	40% (2)	-	-	-	40% (2)	20% (1)	-

TABLE VI: Main reason for the adoption of a tool, as indicated by the survey participants. *free (not open-source)

Tool	Main applications		Main simulated robots	
Gazebo	33% (5) mobile robotics	(4) service robotics, (3) humanoid robotics	40% (6) Atlas , 33% (5) custom platform	(4) wheeled vehicle, quadrotor, turtlebot, (3) PR2
ARGoS	46% (5) mobile robotics, 36% (4) swarm robotics	(2) flying robots	64% (7) khepera/e-puck/thymio	(4) marXbot/footbot, (3) quadrotor
ODE	50% (5) humanoid robotics	(2) multi-legged robotics, snake robots, (1) numerical simulation of physical systems	40% (4) multi-legged robot	(2) iCub
Bullet	25% (2) humanoid robotics, numerical simulation of physical systems	(1) industrial manipulators, human motion analysis, mobile robotics, multi-legged robotics	25% (2) multi-legged robot	
V-Rep	29% (2) mobile robotics	(1) industrial manipulators, humanoid robotics, mechanical design, cognitive architectures, service robotics		29% (2) Nao , quadrotor, wheeled vehicle, Bioloid, khepera/ e-puck/ thymio
Webots	43% (3) mobile robotics, multi-legged robotics	(1) humanoid robotics		29% (2) KUKA LWR, Lego Mindstorm, wheeled vehicle
OpenRave	50% (3) humanoid robotics, service robotics		50% (3) PR2	
Robotran	60% (3) humanoid robotics	(1) human motion analysis, flying robots	60% (3) Coman	(2) iCub
Vortex	66% (2) mobile robotics	(1) humanoid robotics	66% (2) wheeled vehicle	(1) Barret arm
OpenSIM		33% (1) assistive robotics, numerical simulation of physical systems, humanoid robotics		33% (1) khepera/ e-puck/ thymio, iCub
MuJoCo		33% (1) mechanical design, humanoid robotics, numerical simulation of physical systems	66% (2) Atlas, Nao, Shadow hand, Barret arm, HRP2	
XDE	40% (2) humanoid robotics	20% (1) industrial manipulators, numerical simulation of physical systems, human motion analysis	40% (2) industrial robots, KUKA LWR	20% (1) iCub , wheeled vehicle

TABLE VII: Information about the application of the most diffused software tools, and the simulated robots.

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, crresim, 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 VIII: Most diffused tools for a selection of the research areas.

Robot	Users	Most used software	Other used software
Wheeled vehicle	28 (24%)	(4) Gazebo, V-Rep	(3) ARGoS, (2) Morse, Webots, Vortex, (1) Autodesk, Matlab/Simulink, Adams, trep, XDE, SIMPACK, Autolev, RCIS, Bullet, RoboticsLab, own code
Multi-legged robot	18 (15%)	(4) ODE	(2) SL, Bullet, Webots, (1) V-Rep, Adams, Drake, trep, MuJoCo, SIMPACK, Autolev, RoboticsLab
Quadrotor/quadcopter	17 (14%)	(4) Gazebo, ARGoS	(2) V-Rep, (1) Morse, Matlab/Simulink, Drake, ODE, trep, Webots, RoboticsLab
PR2	14 (12%)	(3) OpenRave	(2) Gazebo, MuJoCo, (1) Bullet, V-Rep, Drake, Morse, ODE, RoboticsLab, own code
iCub	13 (11%)	(3) Arboris-Python	(2) ODE, Robotran, iCub_SIM, (1) Bullet, Gazebo, OpenSim, XDE
Atlas	10 (8%)	(6) Gazebo	(2) MuJoCo, (1) Autolev, Drake
Nao	8 (7%)	(3) V-Rep	(2) MuJoCo, (1) ODE, OpenRave, Webots
HRP2/4	6 (5%)	(2) MuJoCo	(2) own code, (1) ODE, Drake
Hubo	3 (3%)		(1) RoboticsLab, ODE, Drake
Asimo	3 (3%)		(1) Arboris-Python, V-Rep, own code
Reem-C	1 (1%)	(1) Gazebo	

TABLE IX: Some of the most simulated robots - the colored cells indicate humanoid robots.

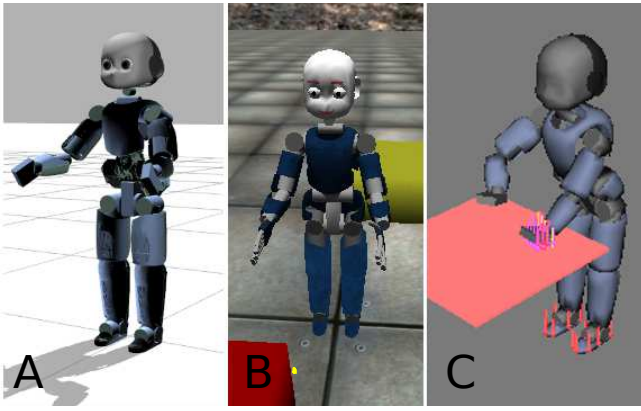


Fig. 3: Simulators of iCub: (A) Gazebo, (B) iCub_SIM, based on ODE, (C) XDE. See video attachment.

engines, which makes it the ideal candidate for dynamics simulations with contacts. Gazebo has a modular structure, and its architecture can be easily extended by adding plugins. Thanks to a plugin for interfacing Gazebo with YARP it is now possible to simulate the iCub and write the same code for the simulated and real robot [16]. The video attachment illustrates the behavior of the three simulators (see Figure 3) during multi-contact scenarios. The video illustrates that ODE is not suited for contact simulation, as there is co-penetration between the robot arm and the table during a contact. Conversely, XDE and Gazebo are able to simulate contacts in a proper way.

IV. CONCLUSIONS

In this paper we presented an overview of the tools for simulating the robot dynamics, together with the user evaluation of the most diffused tools extracted from an online survey. According to our survey, researchers stressed the importance of more realistic simulations, same code for both real and simulated robots, and open-source software, which was also indicated as a criteria for the adoption of a tool. Despite the great diversity of tools for dynamics simulation, these requirements are not generally met. In particular, most of the available tools are still based on physics engines classically used for virtual characters and computer graphics, whereas the robotics community demands physics engines with direct support of robotics descriptions of multi-body systems and optimized contact solvers. No single simulator dominate the others in terms of research application. However, for humanoid robotics, Gazebo emerges as the best choice among the open-source projects, while V-Rep is the preferred commercial simulator.

The plethora of existing software tools seems, in our view, a dispersion of efforts. Researchers should concentrate their efforts on common open-source projects so that all desirable features and improvements (e.g., contact models, solvers) can be shared by the robotics community. The benefit of open-source is not only in the community that can grow around the software, developing new tools and improving its quality, but especially in testing its efficiency and robustness with real

platforms, which is expensive. We believe Gazebo to be a good candidate and as such we chose it for the new iCub simulator.

ACKNOWLEDGMENT

The authors are supported by the EU Project CODYCO (FP7-ICT-2011-9, No. 600716). The authors would like to thank T. Hermans, E. Rückert and J.-B. Mouret for their comments.

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*, pages 2591–2598, 2012.
- [2] E. Drumwright and D.A. Shell. An evaluation of methods for modeling contact in multibody simulation. In *ICRA*, pages 1695–1701, 2011.
- [3] E. Drumwright and D.A. Shell. Extensive analysis of linear complementarity problem (lcp) solver performance on randomly generated rigid body contact problems. In *IROS*, pages 5034–5039, 2012.
- [4] ODE, <http://www.ode.org/>.
- [5] Bullet, <http://bulletphysics.org>.
- [6] R. Featherstone and D. E. Orin. *Handbook of Robotics*, chapter Dynamics, pages 35–65. B. Siciliano and O. Khatib Eds., 2008.
- [7] A. Jain. *Robot and Multibody dynamics: analysis and algorithms*. Springer, 2011.
- [8] E. Todorov. Analytically-invertible dynamics with contacts and constraints: theory and implementation in mujoco. In *ICRA*, 2014.
- [9] E. Todorov, T. Erez, and Y. Tassa. Mujoco: A physics engine for model-based control. In *IROS*, pages 5026–5033, 2012.
- [10] E. Todorov. A convex, smooth and invertible contact model for trajectory optimization. In *ICRA*, pages 1071–1076, 2011.
- [11] Numerical simulation of finite dimensional multibody nonsmooth mechanical systems. *Applied Mechanics Reviews*, 55:107–150, 2002.
- [12] Yan-Bin Jia. Three-dimensional impact: energy-based modeling of tangential compliance. *Int. J. Robotic Research*, 32(1):56–83, 2013.
- [13] C. Duriez, F. Dubois, A. Kheddar, and C. Andriot. Realistic haptic rendering of interacting deformable objects in virtual environments. *IEEE Trans. on Visualization and Comp. Graphics*, 12(1):36–47, 2006.
- [14] C. Duriez. Control of elastic soft robots based on real-time finite element method. In *ICRA*, pages 3982–3987. IEEE, 2013.
- [15] V. Tikhonoff, A. Cangelosi, P. Fitzpatrick, L. Metta, G. and Natale, and F. Nori. An open-source simulator for cognitive robotics research: the prototype of the icub humanoid robot simulator. In *Workshop on Perf. Metrics for Int. Systems*, pages 57–61, 2008.
- [16] E. Mingo Hoffman, S. Traversaro, A. Rocchi, M. Ferrati, A. Settini, F. Romano, L. Natale, A. Bicchì, F. Nori, and N. Tsagarakis. Yarp based plugins for gazebo simulator. In *Modelling and Simulation for Autonomous Systems Workshop (MESAS)*, 2014.
- [17] NVIDIA PhysX, <http://www.geforce.com/hardware/technology/physx>.
- [18] DART, <http://dartsim.github.io/>.
- [19] Gazebo, <http://gazebo-sim.org/>.
- [20] ARGoS, <http://iridia.ulb.ac.be/argos/>.
- [21] V-Rep, <http://www.coppeliarobotics.com/>.
- [22] O. Michel. Webots: Professional mobile robot simulation. *International Journal of Advanced Robotic Systems*, 1(1):39–42, 2004.
- [23] OpenRave, <http://openrave.org/>.
- [24] Robotran, <http://www.robotran.be/>.
- [25] Vortex Dynamics, <http://www.cm-labs.com/robotics>.
- [26] OpenSim, <http://opensim.stanford.edu>.
- [27] SimBody, <https://simtk.org/home/simbody/>.
- [28] MuJoCo, <http://www.mujoco.org/>.
- [29] XDE, <http://www.kaliteo.fr/lsi/en/aucune/a-propos-de-xde>.
- [30] MOBY, <http://physim.sourceforge.net/>.
- [31] OpenHRP, <http://www.openhrp.jp/openhrp3/en/>.
- [32] S. Ivaldi, V. Padois, and F. Nori. Tools for dynamics simulation of robots: a survey based on user feedback. *arXiv:1402.7050[cs.RO]*.
- [33] Webots, <http://www.cyberbotics.com>.
- [34] R. Diankov. *Automated Construction of Robotic Manipulation Programs*. PhD thesis, Carnegie Mellon University, August 2010.
- [35] S.L. Delp, F.C. Anderson, A.S. Arnold, P. Loan, A. Habib, C.T. John, E. Guendelman, and D.G. Thelen. Opensim: Open-source software to create and analyze dynamic simulations of movement. *IEEE Transactions on Biomedical Engineering*, 2007.