УДК 004.2; 004.896
МРНТИ 50.41;55.30

A REVIEW OF ROBOTICS AND HAND TOOLBOX: A JOURNEY ON THE OPEN-SOURCE ROBOTICS WORLD WITH SIX TOOLS

NURMUKAN K.[1], KUANDYKOV A.[2], AMANTAYEVA A.[1], TOLEUGAZY P.[1]

[1]al-Farabi Kazakh National University
[2]International Information Technology University

Abstract: *In robotics development processes start from high level abstraction to demonstrating the particular case or simply prove an algorithm compared to another algorithm works better most of the time researchers might be seeking a straightforward way. It does work if you already set up your own ready research environment but it is not an easy way if you don't have a proper lab environment. In this paper authors motivated by those research environment problems, beginning to discover from authors own perspective attempt to classify the robotics tools with following criteria: by open source, by frequently improving updated versions activity, by whether or not widely supported by researchers and industry or worldwide user groups etc. Finally, author come out with a pace of recommendation for new robotics research with authors research experience to the six open source robotics tools. During the past two years author used the six tools, some of them used not much, some used more. Whatever now is the time to tell people which one has what features compared to each other. In addition, because of the author's own research interest mainly from robotic hand grasping, naturally two related toolboxes (GraspIT! and Syngrasp) are also covered with the same criteria. Author hopes the viewpoint of "the journey" will help others save a pace of time for robotics research.*

Key words: *Open Source Robotics, Robotics Toolbox, Robotics Library,Robotics Simulator, Syngrasp, GraspIT!, Robot Hand, ROS*

АЛТЫ АШЫҚ РОБОТОТЕХНИКАСЫНА ШОЛУ

Аңдатпа: *Робот техникасын дамыту процесінде жоғары деңгейдегі абстракциядан нақты жағдайды көрсетуге дейін немесе жай ғана дәлелденген алгоритммен немесе басқа алгоритммен салыстырғанда, алға қарай жылжуға ыңғайлы болады. Егер сіз дайын зерттеу ортасын орнатқан болсаңыз, онда жақсы жұмыс істейді, бірақ сізде тиісті зертханалық орта болмаса, оңай емес. Бұл жұмыста автор робототехника құралдарын бірнеше критерийлер бойынша жіктеуді өз тұрғысынан ашуға тырысып, зерттеу ортасының проблемаларын қозғаған.*

Түйінді сөздер: *ашық робототехника, Robotics Toolbox, Robotics Library, Robotics Simulator, Syngrasp, GraspIT!, Robot Hand, ROS*

ОБЗОР ШЕСТИ ОТКРЫТЫХ ПЛАТФОРМ РОБОТОТЕХНИКИ

Аннотация: *В процессе разработки робототехники начните с высокого уровня абстракции до демонстрации конкретного случая или просто проверенный алгоритм или по сравнению с другим алгоритмом работает лучше, в этом случае может быть поиск прямого пути. Это работает лучше, если вы уже настроили свою собственную готовую исследовательскую среду, но это не простой способ, если у вас нет подходящей лабораторной среды. В этой статье авторы руководствуются этими проблемами исследовательской среды, пытаясь обнаружить с собственной точки зрения классификацию инструментов робототехники по нескольким критериям.*

Ключевые слова: *Открытая робототехника, Robotics Toolbox, Robotics Library, Robotics Simulator, Syngrasp, GraspIT!, Robot Hand, ROS*

Introduction

Any researcher who believes an idea is worth trying, it leads that eager to success. Assume that you are not a kind of "lucky enough" new researcher, does not have enough lab resources either limited budget to purchase robotics even so who believe that the "Algorithm" has unique capabilities, turn out that beyond the eagerness to success of the feeling drive you continue to prepare the R&D environment. Probably, the expectation happens all the time on your side. Another assumption that you are a member of an interesting project which follows up an unexpected plan, or that project needs cross-disciplinary researchers, to each researcher some aspect of the project seems to be black box at first. Furthermore, It makes communication difficult in the research process. Either of case, In order to continue the research activity or effectively assist each other, the obstacles and the unfamiliar part needs a highly abstract layer to wrap it up by service, which mostly adapt with software Application Programming Interface (API) by specific library or a tool box maybe possible in a simulation platform to increasingly simplify the communication each individual researcher also helps out the productivity of research result. The main idea of each aspect in the project has to be connected with loss coupled independent abstract layers. This helps each researcher stand up in a proper working environment able to effort their own skills and with the best knowledge. Especially, one typical case that some researcher who coming from computer engineering background doing robotics research, generally, robotics include cross four science field (those are mechanics plus mathematics and electronics engineering and software engineering field) basically software engineer it is not possible to understand at first place all of the tools or methodology of robotics science, but if want to learn to apply it, have to has a way to gradually possible. That is the reason, I went through the personal research experience, understood how difficult and what is the most difficult for as a software engineer doing robotics research if not has proper guides. The following content will describe six tools specific for robotics, reusable level differently, some of them can be applicable libraries and some of them platform simulation level. All of them are open source. For each of them exposing the main feature starts from evaluating building from source file to installation, configuration, using in the test or demonstrating examples to compare each other how those features are provided, how long been versions updated, compared each robotics tool does the same feature or differently provided. Also because of my own interest mainly from robotic hand grasping, naturally many related toolboxes are also covered with the same criteria. Finally I made a little bit of a recommendation as a conclusion.

Robotics Library (RL)

Robotics Library (RL) [14] is a library for mechanical kinematic, motion planning and robotics control. It includes math function, kinematics, dynamics and hardware abstraction, collision detection, and visualization. It was not only used research project [5] but also used an industry robotics project [13]. In Figure 1, Figure 2 and Figure 3 with few examples to show basic robotics functionality of what RL can do.

Figure 1: Using RL Demonstrating the basic of rigid body kinematics behavior

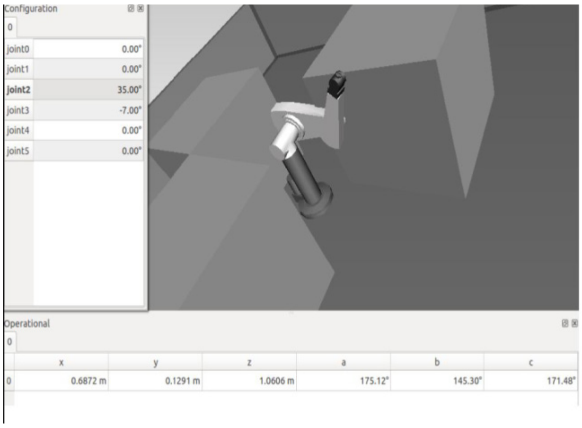

Figure 2: Demonstrating more standard robotics arm with kinematics joint constraints

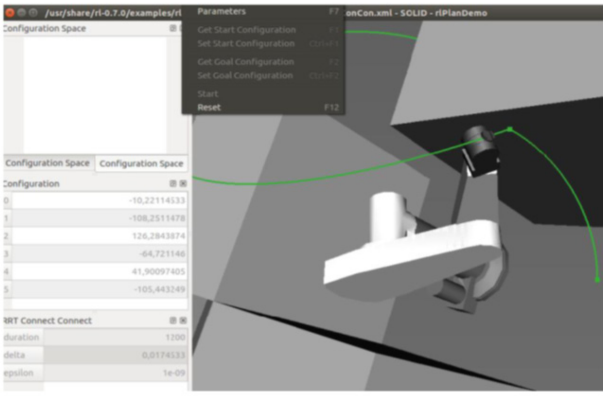

Figure 3: Demonstrating the path planning algorithm with robotics arm

DART

DART (Dynamic Animation and Robotics Toolkit) [8], official website: https://dartsim.github.io, DART more importantly focused on accuracy of for all aspect of robotics kinematic dynamic computation, more likely it can contrast with other physics engine ODE, Bullet etc. The following picture (Figure 4, Figure 5, Figure 6, Figure 7, Figure 8, Figure 9) visually shows few aspects of the capability of DART does.

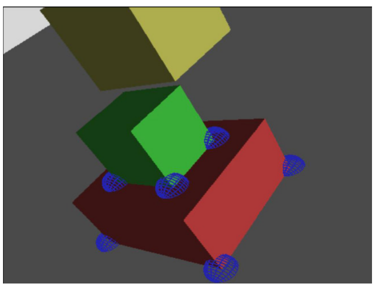

Figure 4: DART demonstrating basic rigid body falling down process with physics nature visualizing, as the demo they didn't cross penetrating each other while boxes hit each other

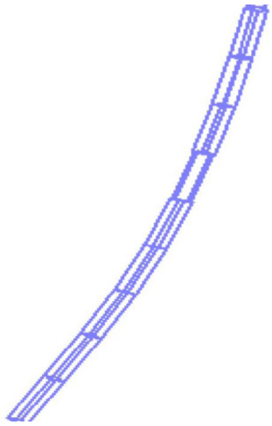

Figure 5: DART demonstrating a multi jointed part pendulum process, which shows that with nature of physics behavioral process

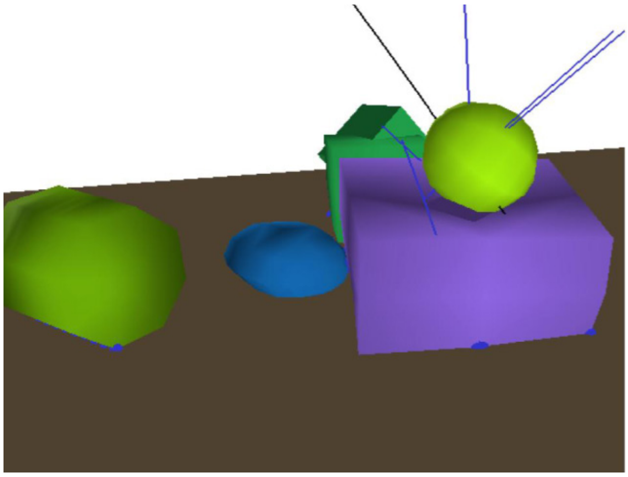

Figure 6: DART demonstrating basic soft rigid body falling down process with collision effect physics nature visualizing , as the demo they didn't cross penetrating each other but they deforming or squeezing while boxes are hit each other

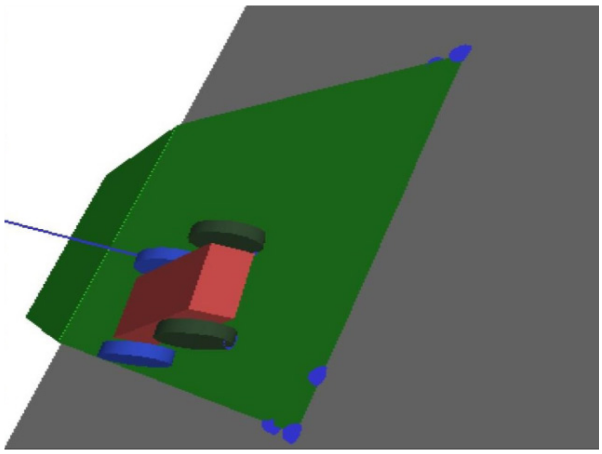

Figure 7: DART demonstrating simple modules of vehicle moving path in the particular environment, also be able to control the moving process

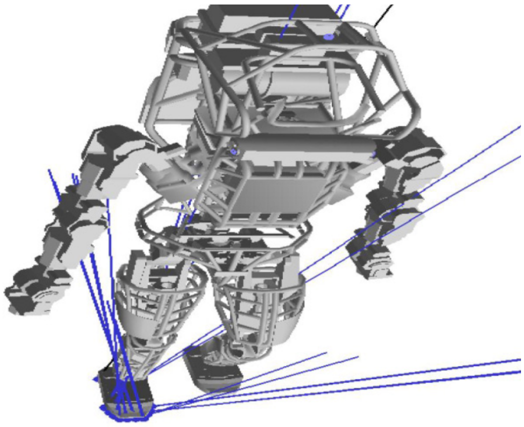

Figure 8: DART demonstrating the step moving, keep balancing during the moments etc. with the famous ATLAS whole body humanoid robotics

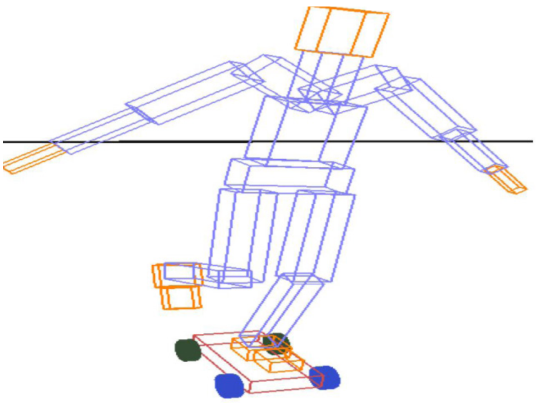

Figure 9: DART demonstrating another simpler whole humanoid rigid body skeleton module balanced during the moving

SynGrasp

SynGrasp Toolbox, it is a matlab toolbox[10], focus on hand dexterity and manipulation analize. Active open source tool. Last updated in December 2019, till now a new updated 2.3 version, found on the official site: syngrasp.dii.unisi.it. The tools used many eropone project as following :Hands.dvi ECHORD Project [15], THE – The Hand Embodied, WEARHAP – Wearable haptics for humans and robots [12], SOMA – Soft Manipulation[16] [17]. Main feature of SynGrasp Toolbox: 1. Hand Grasp Modeling: Four robot hand models are provided. Humanoid hand, three-finger hand, modular hand, DLR / HIT second-generation hand. In Figure 10 -Figure 11 shows that grasping object with three fingered hands, but in Figure 12 - Figure 13 shows with five fingered hands. 2. Grab modeling and planning: Grab gripper configuration, contact point, grab Jacobian matrix calculation, controllable internal force subspace, rigid object operation. 3. Grasp analysis: Provides different grasp quality evaluation indicators. Including optimized force distribution, and taking into account the under-actuated hand joint coupling (hand synergy) see in Figure 14 and Figure 15. allowing the execution of motion and force operation capability analysis. 4. Graphical interface: visualization of objects and grippers.

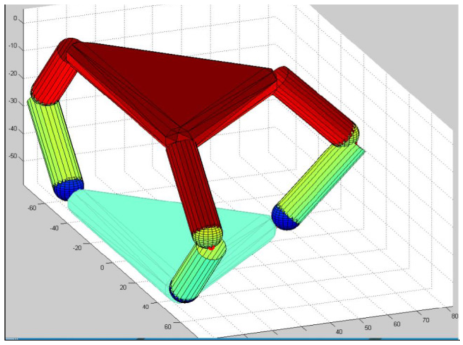

Figure 10: SynGrasp showing the persigin grep with three fingered hand modules

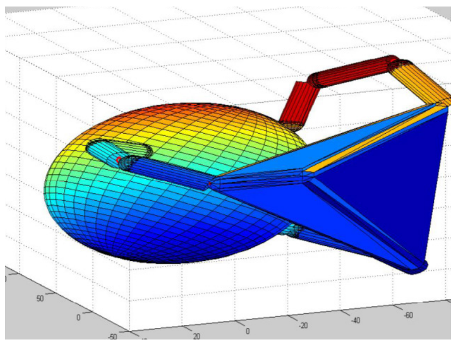

Figure 11: SynGrasp showing the finger grasp the sphere with three fingered hand modules

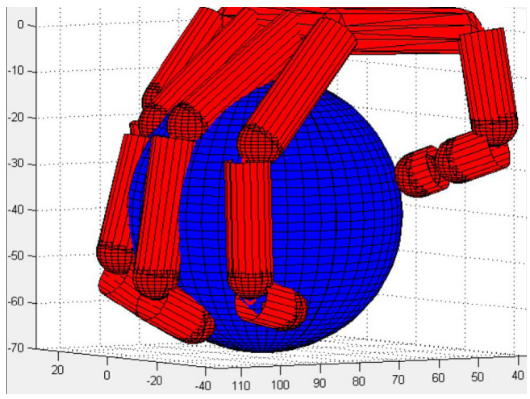

Figure 12: SynGrasp showing articulate the spare object with Five finger hand modules

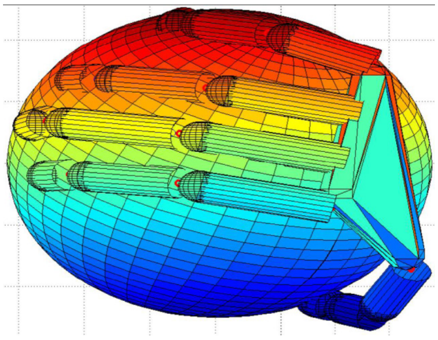

Figure 13: SynGrasp showing the power grasp the spare object with five finger hand modules

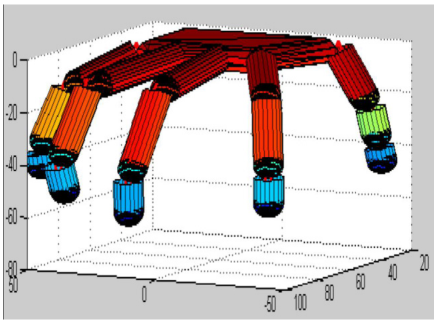

Figure 14: SynGrasp demonstrating the hand synergy, hand extension

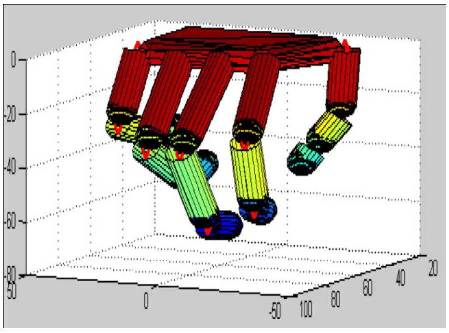

Figure 15: SynGrasp demonstrating the hand synergy, hand power grasp planning

OpenRAVE

OpenRAVE (Open Robotics Automation Virtual Environment), the offical web site http://openrave.org last version 0.9, was last updated in 2013[4]. OpenRAVE is an open source robot simulation software. OpenRAVE provides a testing environment for robots, and its main function is the simulation and analysis of kinematics and geometric information for motion planning. In terms of applications, OpenRAVE is mainly used to develop and deploy robot motion planning algorithms, and these algorithms can be applied to actual robots. Due to the nature of OpenRAVE's independent operation, these algorithms can be easily integrated into existing robotic systems. It provides many command-line tools for robot developers and robots with a core runtime small enough to be used for internal controllers and larger frameworks. OpenRAVE is an open source crossplatform software architecture, namely an open robot and animation virtual environment. OpenRAVE targets real-world robotic applications, including seamless integration of 3-D simulation, visualization, planning, scripting, and control. Its plug-in architecture allows users to easily write custom controllers and extend functionality. By using the OpenRAVE plug-in, any designed algorithm, robot controller, or sensing subsystem can be distributed and dynamically loaded at runtime, thereby freeing developers from using a monolithic code base. In this way, OpenRAVE users can focus on problem planning and script development without having to explicitly manage the details of robot kinematics and dynamics, collision detection (e.g Figure 16), world updates, and robot control. The OpenRAVE architecture also provides a exible interface that can be used in conjunction with other popular robot software packages such as Player and ROS, as it focuses on automatic motion planning (e.g Figure 17) and high-level scripting rather than low-level control and information protocols. OpenRAVE also supports a powerful scripting environment, making it easier to control and monitor robots and change execution processes at runtime.

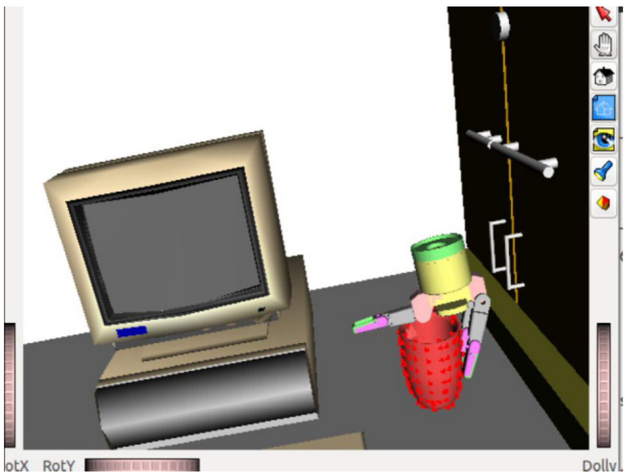

Figure 16: OpenRAVE demonstrating the grasping a cup

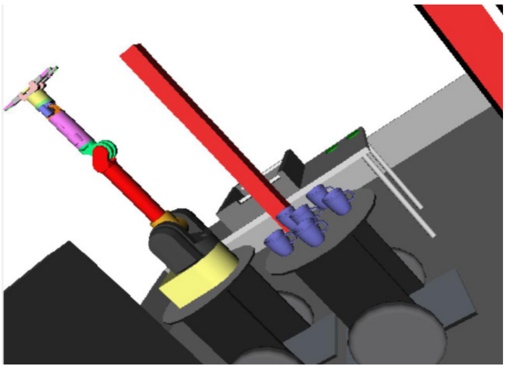

Figure 17: OpenRAVE demonstrating path planning trajectory

GraspIt!

GraspIt! [11] was designed for hand grasping open source systems, not for all general purposes functionality of robotics simulation. Grasping control has a problem of high performance computation. Especially multi- fingered grasp has a more complex DoF design complexity problem [7]. The complexity of computation is very time consuming in terms of grasp planning including collision detection. That is the reason some researchers doing pre-grasp computation as offline optimisation result save it for "best grasp" result to database [6], get load while need feasible grasp without collision, the grasp plan available without math modeling computation instead of getting it searching in the database, it would show outstanding performance. If grap planning counts as part of the motion planning, turnout Improving motion path planning eventually improves the whole robotics performance. GraspIT! one of the solutions and reasons why researchers like to use it. Following two picture (see 3-17.Figure 3-17 and 3-18.Figure) show visual grasping results with a multi-fingered hand.

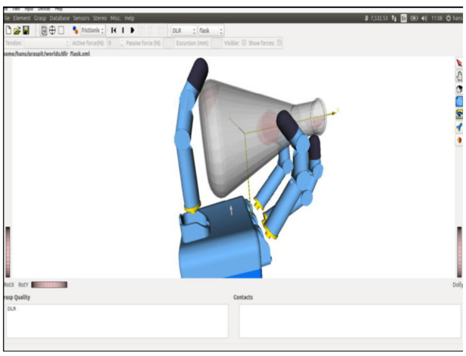

Figure 18: GraspIT! Demonstrating the auto grasping algorithm

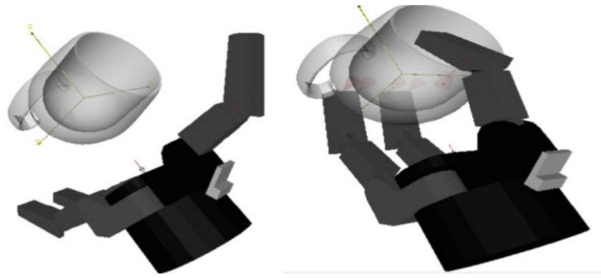

Figure 19: GraspIT! Showing that the two separate stage of the grasping

ROS (Robotics Operating System)

The primary design goal of ROS is to increase code reuse in the field of robotics research and development [2]. ROS is a distributed processing framework (as Nodes). This allows executables to be designed individually (nodes) and loosely coupled at runtime. These processes can be encapsulated into packages and stacks for easy sharing and distribution. ROS also supports a federated system of code bases. Enable collaboration to be distributed. This design from the file system level to the community level makes it possible to independently determine development and implementation. learning or experiencing robotics multi agent system, ROS where are the right places to go[1]. ROS makes it possible to share open source robotics to the public [3].

Disscustion

In this paper compared the six robotics libraries/framework/ or platform (ROS and OpenRAVE can be called platform) from seven different aspects. See more details in Table-1. Basically, RL and DART are forced on more fundamental robotics application purposes. But DART is especially more care about calculation accuracy and applying physics law as a core of usages in the library. This is the reason DART has been used as a physics engine in some platforms (i.e OpenRAVE). As you can see from a programming language perspective C ++ is a very important language in robotics research. SynGrasp and Graspit! Spacillay only for robotics hand analysing purposes has been provided. Noticeably, one of the robotics applications interestingly covered investigation specific

Table 1: The six robotics platforms compared with seven different aspects

	Program Language	Single Robot/ Multi Agent	Motion Planning	Collision Detection	Robotics Research Purpose	Developmentt Environmen	Robotics Geometry Format
RL	c++	single	Yes	Yes	General	Liberary	URDF/VRML (.wrl)
DART	C++, Python	Single	Yes	Yes	General	Library	URDF/SDF
OpenRAVE	C++, Python, Octave, Lisp	Single	Third part integrable	Third part integrable	General + Robot Hand	Plugin + Simulation	COLLADA (*.dae/*.zae)
SynGrasp	MatLAB only	Single	Only for Grasp Planning	No	Robot Hand	Library + Analysing tool	No
GraspIT!	C++, MatLAB interface	Single	Only for Grasp Planning	Yes	Robot Hand	Simulation + Analysing tool	VRML (.wrl) and Inventor (.iv) files
ROS	C++, Python, Octave, Lisp	Multi Robot(Agent)	Third part integrable	Third part integrable	All	All	URDF

for robotics hand grasping aspects, which tool called "OpenGrasp"[9], which integrated many usefully lib and tools like OpenRAVE, FISICAS etc, the main focus was also only for robotics hand, but personally I did not recommend at all for two reason, first, currently OpenGrasp didn't work either with source in svn or other pre-build packages distribution installation, same time I found it some users notice that it is a long time didn't update the release(last updated time 2011). Second, personally spending a lot of time trying it out didn't succeed. Finally, I would like to recommend the ROS(Robotics Operating System) compared with other robotics tools. ROS might be a better choice if you like to investigate multi robotics(multi agent) fields.

Conclusion

In order to continue research activity or effectively assist each other, the obstacles and the unfamiliar part needs a highly abstract layer to wrap it up by service, which mostly adapt with software Application Programming Interface(API) by specific library or a tool box maybe possible in a simulation platform to increasingly simplify the communication each individual researcher also helps out the productivity of research result. The main idea of each aspect in the project has to be connected with loss coupled independent abstract layers. This helps each researcher stand up in a proper working environment able to effort their own skills and with the best knowledge.

REFERENCES

1. M. Alberri, S. Hegazy, M. Badra, M. Nasr, O. M. Shehata, and E. I. Morgan. Generic ros-based architecture for heterogeneous multi-autonomous systems development. In 2018 IEEE International Conference on Vehicular Electronics and Safety (ICVES), pages 1–6, Sep. 2018.
2. S. Cousins. Welcome to ros topics [ros topics]. IEEE Robotics Automation Magazine, 17(1):13-14, March 2010.
3. S. Cousins, B. Gerkey, K. Conley, and W. Garage. Sharing software with ros [ros topics]. IEEE Robotics Automation Magazine, 17(2):12–14, June 2010.
4. Rosen Diankov. Welcome to open robotics automation virtual environment-openrave, 2013.

5. Mary Ellen Foster, Markus Rickert, and Michael Braun. The JAST collaborative human-robot dialogue system. In Proceedings of CogSys II (Poster session), Nijmegen, Netherlands, 2006.
6. C. Goldfeder, M. Ciocarlie, Hao Dang, and P. K. Allen. The columbia grasp database. In 2009 IEEE International Conference on Robotics and Automation, pages 1710–1716, May 2009.
7. John Lin, Ying Wu, and T. S. Huang. Modeling the constraints of human hand motion. In Proceedings Workshop on Human Motion, pages 121–126, Dec 2000.
8. Jeongseok Lee, Michael X. Grey, Sehoon Ha, Tobias Kunz, Sumit Jain, Yuting Ye, Siddhartha S. Srinivasa, Mike Stilman, and C. Karen Liu. DART: Dynamic animation and robotics toolkit. The Journal of Open Source Software, 3(22):500, Feb 2018.
9. Beatriz León, Stefan Ulbrich, Rosen Diankov, Gustavo Puche, Markus Przybylski, Antonio Morales, Tamim Asfour, Sami Moisio, Jeannette Bohg, James Kuffner, and Rudiger Dillmann. Opengrasp: A toolkit for robot grasping simulation. In Noriaki Ando, Stephen Balakirsky, Thomas Hemker, Monica Reggiani, and Oskar von Stryk, editors, Simulation, Modeling, and Programming for Autonomous Robots, pages 109–120, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.
10. M. Malvezzi, G. Gioioso, G. Salvietti, and D. Prattichizzo. Syngrasp: A matlab toolbox for underactuated and compliant hands. IEEE Robotics Automation Magazine, 22(4):52–68, Dec 2015.
11. A. T. Miller and P. K. Allen. Graspit! a versatile simulator for robotic grasping. IEEE Robotics Automation Magazine, 11(4):110–122, Dec 2004.
12. Domenico Prattichizzo and Giovanni Spagnoletti. Wearhap - wearable haptics for humans and robots, 2017.
13. Gunther Reinhart, Wolfgang Vogel, Wolfgang Rösel, Frank Wallhoff, and Claus Lenz. Jahir - joint action for humans and industrial robots. In Fachforum: Intelligente Sensorik - Robotik und Automation. Bayern Innovativ - Gesellschaft four Innovation und Wissenstransfer mbH, 2007.
14. M. Rickert and A. Gaschler. Robotics library: An object-oriented approach to robot applications. In 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 733–740, Sep. 2017.
15. Gionata Salvietti, Guido Gioioso, Monica Malvezzi, Domenico Prattichizzo, Alessandro Serio, Edoardo Farnioli, Marco Gabiccini, Antonio Bicchi, Ioannis Sarakoglou, Nikos Tsagarakis, and Darwin Caldwell. Hands.dvi: A device-independent programming and control framework for robotic hands. In Florian Rohrbein, Germano Veiga, and Ciro Natale, editors, Gearing up and accelerating cross-fertilization between academic and industrial robotics research in Europe:, pages 197{215, Cham, 2014. Springer International Publishing.
16. C.D. Santina, V. Arapi, G. Averta, F. Damiani, G. Fiore, A. Settimi, M. G. Catalano, D. Bacciu, A. Bicchi, and M. Bianchi. Learning from humans how to grasp: A data-driven architecture for autonomous grasping with anthropomorphic soft hands. IEEE Robotics and Automation Letters, 4(2):1533–1540, April 2019.
17. C.D. Santina, C. Piazza, G. Grioli, M. G. Catalano, and A. Bicchi. Toward dexterous manipulation with augmented adaptive synergies: The pisa/iit softhand 2. IEEE Transactions on Robotics, 34(5):1141–1156, Oct 2018.