Filomat 34:15 (2020), 4953–4966
https://doi.org/10.2298/FIL2015953K

Published by Faculty of Sciences and Mathematics,
University of Niš, Serbia
Available at: http://www.pmf.ni.ac.rs/filomat

Open-Source Projects for Autonomous Robotics and Systems: A Survey

Ameer Tamoor Khan[a], Shuai Li[a], Dechao Chen[b], Yangming Li[c]

[a]*Department of Computing, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong, China*
[b]*School of Computer Science and Technology, Hangzhou Dianzi University, Hangzhou 310018, P.R. China*
[c]*Department of Electrical and Computer Engineering, Rochester Institute of Technology.*

Abstract. Open-Source has not only removed the monopoly of the few technological companies, but has also distributed the knowledge, at no cost. With knowledge moves on from person to person, and each person adds his/her contribution to the past work, a knowledge production chain keeps rolling, greatly reducing the effort to re-invent wheels. It allows the public availability of data and enables the addition, modification, and edition of data more efficiently at a faster pace. Robots, considered as a replacement of man-power are of meticulous interests for researchers in the past few decades. Their immunity to walk and talk more or less like a human is worth praising, but this radical change was not so obvious a decade or two ago before the wide propagation of open-source, the continuous spread of research work around the world allows the brilliant minds to add their pieces to incrementally growing joint efforts. It has revolutionized the robotics from the simple remote-control cars to the self-driven cars. This survey summarizes main stream open source projects emerging in recent years and expects to increase the exposure of existing open source projects and increase the popularity of them, with an intention to further reduce unnecessary effort to re-invent existing systems.

1. Introduction

In this section, we will talk about the open-source, its implications, why is it important? what changes had it brought to the technological advancement? we will also discuss in detail the innovative and significant work done in robotics through open-source channels [1].

1.1. Motivation

It is a license agreement that allows users to modify others' work, integrate it with their own projects or extend the original one. It promotes the exchange of innovative and disruptive idea among people of scientific and technological community [2]. Open-source is popular among the software and the business community. It has revolutionized the small industry because it allows them to build the complex, expensive systems at the economical price. It is increasingly popular among the educational masses, and forms a way to simplify the spreading of new ideas related to, electronics, IoT, machine learning, robotics, programming,

2010 *Mathematics Subject Classification*. 00-XX, 47N70

Keywords. Open-source, robotics, control, manipulator, autonomous robots.

Received: 11 September 2018; Accepted: 22 March 2019

Communicated by Bolin Liao

This work is supported by Hong Kong Research Grants Council Early Career Scheme (with number 25214015), by Department General Research Fund of Hong Kong Polytechnic University (with number G.61.37.UA7L), and also by PolyU Central Research Grant (with number G-YBMU), by National Natural Science Foundation of China (with number 61503413))

Email addresses: ameertamkhan1992@gmail.com (Ameer Tamoor Khan), shuaili@polyu.edu.hk (Shuai Li)

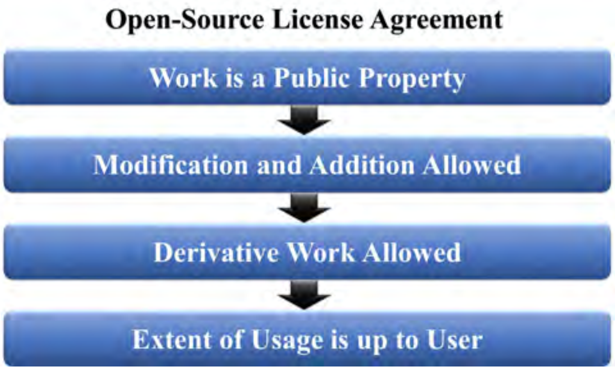
Open-Source License Agreement
Work is a Public Property
Modification and Addition Allowed
Derivative Work Allowed
Extent of Usage is up to User

Figure 1: Open-source license agreement. When a person or a company give up their right on work than they need to obey the above mentioned rules. Which mainly states, the work will become public and anyone has excess to it, one can modify and add things to that work.

biotechnology, *etc.* Many nerds are making use of this helpful platform to learn and practice new and useful concepts [3]. Open-source platforms even allow people to initiate start-ups and promote their own business. According to the agreement, the person or organization who license their work give up the following right on their work and agree to the following:

1. The entire code will be available to the public.
2. Everyone is entitled to the access of the data, the modification of it and the addition to the code.
3. It allows the making of any derivative work.
4. It allows the users to utilized the program to whatever extent they want.

The above-mentioned rules are shown in Fig. 1. This whole idea of open-source is originated from the technological community, for which different companies are working on collaborative projects have to share a pile of useful information with each other [4]. Imagine the scenario that motivates the emergence of open-source: a programmer in Hong Kong develops an application and there is another programmer from India may adopt the program of this application as one module of his/her project, or even modify the program better. It may happen that the Indian programmer will contact the originator to collaborate with each, supporting the community and thus allowing the open access to their works. Such integrated projects give rise to the open-source. Nowadays a person sitting in one corner of the world develops a piece of code, which can be amended and improved by another person sitting far away. The re-use of codes passes from one to one and this chain keeps rolling.

There is a number of open-source platforms available on the internet which serve the community with complex and expensive projects without paying nothing [5]. To name a few, GitHub, instructable, Arduino, raspberry pie, *etc.* There is a number of online hardware platforms available, which provides customers with customized hardware that a user can change, modify and add, without the fear of any liability [6].

1.2. Related Works

Robotics is one of the grooming filed of today's technological arena. It has an impact among the industrial, commerce, scientific and business community. Few years back, when there was no concept of open-source the robotic industry was monopolized by few technology experts. In contrast, benefiting from open-source, even a student can make high-tech robots including self-driving cars and quadcopters *etc.* There is a number of projects completed by utilizing the facility of open-source, we will discuss some of those briefly here and in greater detail in later sections.

Self-driving car is much popular these days among researchers, who are working on an intelligent and fully autonomous vehicle to work like a man-driving cars but in a human-free fashion [7]. Few years back, this concept was monopolized by few companies. Differently in recent years with the wide-spread of open-source platforms, this type of projects is becoming was available publicly and people from all over the world have made quite a number of semi or fully autonomous cars. Although, they may not be perfect for the public use, the open-source platform will keep rolling this concept from user to user, adding improvements to it [8].

There is a number of industrial robotic arms. e.g, like SCARA and PUMA robots, which are mainly used in industries in pick and drop activities. Though these robots look simple, bringing them to use is a hard nut to crack. Forward and reverse kinematics are the conventional methods to formulate them and utilize them for useful purposes. Thanks to open-source, which makes it available for firmware like marlin with predefined libraries of famous robots [9], leaving only basic information related to robots to further developers. In the coming sections, we will discuss in detail about the robots, open-source and its impact on the robotics field [10].

2. Robotic Systems

In this section, we will explore the architecture of robotic systems, their key elements, types of robots, their control and fundamental problems [11].

2.1. Key Elements of Robotic Systems

Robots are machines designed to execute one or more simple and complex tasks with precision, accuracy and speed. Robotic system is a combination of many small mechanical and electrical components integrated together, and with the aid of some control algorithms [12, 13]. The key elements of the robotic system are shown in Fig. 2 and explained below.

Mechanical Platform: It consists of a structure which could be, wheeled platform, a robotic arm or any other platform that is able to interact with the world out there.

Sensors: They are used to sense the environment and extract useful information out of it. The types of sensors include temperature, pressure, light, and sound sensors [14]. They sense the environment and then convert that information into analog or digital signals, and send it for further process. Robots then act according to the control signal. They are categorized into three types. Vision sensors observe the world with the things like the camera. We further apply the image processing techniques to extract the useful information out of it. Proximity sensors are distance measuring sensors which include; ultrasonic and IR sensors. Self-monitoring sensors are used self-monitoring and maintenance of the system itself. These sensors monitor the battery working, heat and pressure in the system.

Effectors: They are the parts of the robot that does some useful works. In conventional robots, they are made of metal or anything with hard texture. In soft robots, they are made of soft material like silicone elastomer [15]. End-effectors are located at the end of the robots, and their jobs are to hold, pick and drop an object.

Motors: They are used to move the robot itself or its parts. They are usually located at the connection between the effectors called joints. Generals, a single motor controls one effector [16–18]. There are a variety of motors available depending upon their use, e.g., dc-motor, ac-motor, servo-motor, brush-less dc-motor *etc.*

Power Mechanism: Power is supplied to the system by mainly using two sources non-rechargeable and rechargeable. Non-rechargeable power sources are the one-time used power sources, such as, alkaline and lithium primary batteries. Rechargeable power sources, e.g., lead-acid, nickel-metal hydride and lithium ion batteries, have a longer life. Once they are out of power we can charge them again.

Electronic Circuitry: This controls the electronic signals supplied to different parts and components of the system. It mostly consists of motor driving units, relays and other logical control [17–20]. It prevents the system from any mishap by limiting the voltage and current. Some of its prevention units are; switches, relays, transistors, H-Bridge. Switches are very basic prevention units that limit the current. If large spikes of current pass through them in case of a system failure, these switches will turn off and will open the circuit preventing it from burning. Relays are bit advance way to limit the high current, and they provide greater insulation than the normal switches. Transistors are made of semiconductor wafers, and are more desirable for circuit prevention from high voltage and current. H-Bridges are made from the transistors.

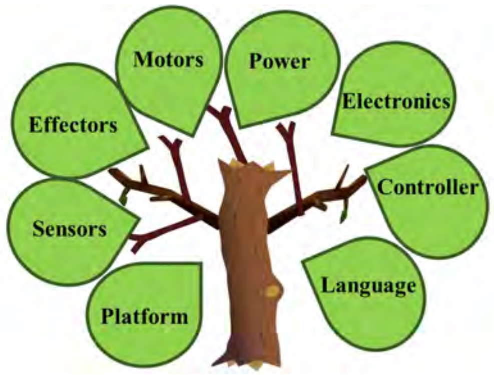

Figure 2: Key elements of robots.

Simple transistor allows the current in one direction. But in this configuration, current can flow in both the configuration, and allows the motor to rotate in both the directions.

Microcontrollers: They are the brain of the robots and are like the microprocessor (CPU) of the computer. All the logic of the system is defined in it and the robots work accordingly. PIC, arduino, raspberry pie are the examples of the microcontroller. The features of the controller are; speed, size, and memory. Speed means fast execution of the commands and efficient processing of data. Controllers offer speed in cycles per second, and they usually are of million cycles per second (MHz). Arduino Uno has the processing speed of 16 MHz. Size specifies the amount of information that it can execute in other words, the data in bits that it can process in one second, e.g., 4, 8, 16 and 32-bits. Memory specifies the amount of information that it can store. It usually has two types of storages. Read-only memory (ROM) is usually in kilobytes (kB) and random-access memory (RAM) is in bytes.

Languages: There are a number of languages that are designed for the robots such as; Robo ML (Robotic Markup Language), VAL (Variable Assembly Language), RoboLogics, XRCL (Extensible Robot Control Language), ROBOFORTH, *etc.*

2.2. *Types of Robots*

These days robotics is a growing industry and jobs in many fields are entrusted to the robots. Based on their jobs, they are divided into different types which are discussed below.

Industrial Robots: These robots are employed in both large and small industries and are usually articulated arms, specifically designed for tasks, such as, pick and drop objects, material handling, assembling objects, and painting *etc.*

Medical Robots: This is one of the fields where robots can contribute. There are several complex tasks in medical that may appear to be difficult for the human hands to reach. But in future we can trust the robots with that. We still have robots that assist the surgeons during operation, but researchers are expanding their scope of employment.

Medical robots are also required to perform other human-like tasks in the hospitals like carrying patients from one place to the other, handling and managing medicines, effective monitoring of the patients, *etc.* Other types of medical robots include micro-robots that operate inside the human body, in their initial phase they are used to remove the blocks inside the veins to continue the flow of blood. In coming years, they will be able to perform even more complex medical tasks even without performing major surgeries.

Military Robots: These robots are deployed in battlefields to carry weapons, information, medical aids, *etc.* They are also used as a military force to destroy terrorist's bunkers. Drone is an example of it. In which a drone locates the target in the terrorist premises and then target the location and return to its base. Military robots can also use to defuse, detonate or to locate the bomb in enemy areas. Similarly, transportation is one of the important tasks to perform during surgical operation. These robots can transport ammunition, information, intelligence and food supplies to other crew members in hard-to-reach areas.

Bioinspired Robots: This is another emerging field of the robots, different from the conventional robots. Nature has got inspiration and researchers made use of it. Different insects, animals, sea creatures have became their inspiration. By studying their anatomy and physical structure, they have been successfully limited in making a robot, whose touch is like any living organism touch. Despite this disruptive technology, researchers are still struggling with a number of barriers and limitations that they are facing in this field, like; the strength of the muscle and the ability of the system to power itself.

Swarm Robots: Swarm robots are the army of robots designed to perform complex tasks that are difficult for the single robot to perform. They can perform the number of the tasks which includes; espionage activities, hospital services, rescue operations, military operations, *etc.*

Robots in Astronomy: Robots have made their way to the high skies. Their jobs are not limited to the earthly matters but are also dealing with the challenges scientists facing in collecting data from space and other planets. There are space robots that are working in different space stations and there are other robots that are sent to different operations on different planets. Mars rover is an example of it. It was send to the mars to collect useful data about mars and its environment.

2.3. Control of the Robotic Systems

Controlling a system is a complex task to accomplish. Robots itself are complex machines so they even required more attention when it comes to their control. There is a number of control algorithms that are being used for this purpose. Here we will discuss some of those basic and advance controlling techniques [21–25]. Both the basic and advanced techniques of robot control are shown in Fig. 3.

2.3.1. Basic Control Techniques

Here are some of the basis techniques employed to control the robotics systems.

Open-Loop Control: This is the basic controlling technique for the simple loads, no feedback, no position or speed sensors required, no speed control required. It has fixed end-point known as "stop-to-stop" system. Its working is simple, with required change to parameters are calculated, with the amount of energy required to the actuators being determined and finally with the amount of energy required to bring that change being applied to the system[26, 27]. If the system is correctly modeled and all the calculations are right. Then no disturbance in the system will be observed and it will achieve the desired results.

Feedback Control Loop: This technique is more controlled as compare to the open-loop because it incorporates the feedback and minimizes the error. Contrary to open-loop it requires position and speed sensors, and based on that, we can move the robot to the required destination with more convenience. We can vary the speed of the wheels depending on our path, monitored from position and speed sensors[28, 29].

Feedforward Control: This is another controlling technique different from the feedback because it does not adjust the parameters on error-base instead it predicts the system condition based on it mathematical modal. Some prerequisite conditions, controlling signal, and the effect of the output of the system on load must be known[30, 31]. This controlling technique is useful when feedback signals are delayed, and prediction of actuator setting is more convenient.

Adaptive Control: This is an advanced type of control in a school of basic controlling techniques. It uses the feedback and updates the modal of the system based on that feedback. In this way, as the system progresses, it will also adopt the changes that it will face. This controlling technique is employed in a variable environment, where we do not want to compromise the working of our system despite the variability[32–34].

2.3.2. Advanced Control Techniques

Basic controlling techniques have some limitations to over come the advance problems. So some advance techniques are employed by the researchers, that are explained in [35–39].

Artificial Intelligence: Artificial intelligence has paved its way in the disruptive technological world. It enables the robots not only to control their motion but also allow them to learn new things and act accordingly, for example, a self-driving car where it observes the environment around and tries to learn from it so that it can provide even better service next time.

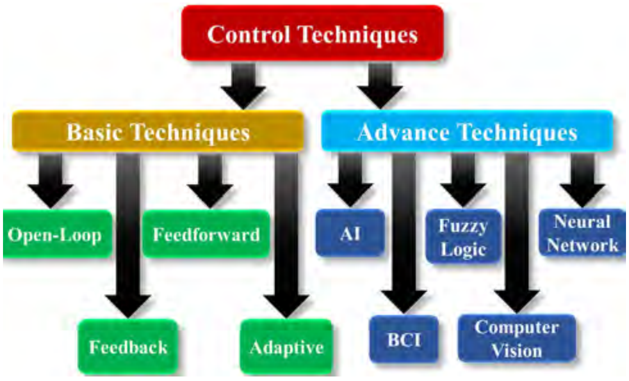

Figure 3: A interracial tree that explains the controlling techniques of the robots; basic and advance techniques. The adaption of these techniques are requirement based, robots for intelligent jobs required to be controlled by advance techniques which involves AI, computer-vision and brain-computer interaction, like techniques.

Brain-Computer Interfaces: It is a way to connect the human with the robots, signals are extracted from the human brain then processed in the computer and finally sends the control signals to the robot and robot act accordingly. Consider a robot that is controlled by the eye movement, if an eye moves to the right the robot will move right and vice versa. Similarly, we can use other controlling signals that may relate to our emotions and sensations.

Fuzzy Control: It is another control technique. Differing is different from defining 0 and 1 logics, it defines the degree of truth and false, and not the definite truth and false. Consider a fast-moving wall following robot. Using fuzzy logic, we will assure the certain distance from the wall and in case, if it deviates from the set limits, then it will not only come back to its given lane but its transition will also be smooth, i,e., speed and angle of the wheels while taking a turn.

Computer Vision: It involves camera and computer algorithms that process the real world and extract the useful information out of it and based on that information[40, 41]. It sends the control signal to the robot and the robot then act accordingly. For example, consider a robot whose job is to pick an apple and put it in a bucket, using computer vision we will observe the environment with the help of the camera and then will extract the apples out of it, based on its texture, its color. Once it is identified, we will calculate its coordinates and will send the robot to that position where it will pick the apple and will take it to the bucket.

Neural Network: It is an algorithm that is inspired by the millions of the neurons connected inside the humans. Since the neurons inside the body act as a signal transmitter and receiver of the brain. It learns the human body to learn different activities. Consider a baby trying to walk on his feet. At first it will be difficult for him in doing so but as soon as the neurons will find the effective path to communicate with the brain and the repetitive to and fro motion of neurons will help them to learn the particular tasks. A similar methodology is employed in the artificial neural network where we have neurons with certain weights, and their weight is decided based on the circuit or the metamathematical modal used.

2.4. *Fundamental Problems in Robotics*

In this section, we will discuss the fundamental problems that researchers still fill in robotics to some extent. The problems include; interaction with the real world, mobility and navigation, human-robot interaction, and new material and fabrication schemes [42–44]. The problems are also mentioned in Fig. 4.

2.4.1. *Interaction With the Real World*

Robots are not only designed to observe the world but also to effectively interact with it. Their interaction with their surrounding should be more humans like. They should be able to grasp, handle, and repair the objects. Their interaction with the men should be human like [45–47]. The robots like, kodomoroid, gemioid DK, junko chihira and sophia are the examples of such robots.

Interaction With Real world
Mobility and Navigation
Human Robot Interaction
New Material and Fabrication Scheme

Figure 4: Fundamental problems that robots face. These are only few otherwise more intelligent machines we think of, the more complexities and problems we will face and researchers are facing them.

2.4.2. *Mobility and Navigation*

This is one of the biggest problems that robots still face today, their ability to navigate around and reach their destination correctly. A robot either on a surface, in sear or in space must be able to locate its reaching point and also the shortest and effective path to it, and in case any hurdle occurs or the robot somehow dislocate from its target path. Then, it should be intelligent enough to relocate its path to the destination[48].

2.4.3. *Human-Robot Interaction*

Advancement in robots have reached to that point where they show can safely interact with the men, although much more research needed to be done to make them more human-like. We are living in an era where human-labor needs to perform life-threatening tasks and in doing so many of the people lost their lives so they need assistance. Soft robots are the kind of robots who have human-like touch because they are soft in nature still a lot of works needs to be done on hardware and software level. High-level artificial intelligence is also required to bring the robots nearer to human life.

2.4.4. *New Material and Fabrication Schemes*

The robot now days are moving away from motors, sensors, material effectors, gears and large batteries. Now they are transferring to artificial muscles, soft robots and new fabrication techniques to make it more compact and life-like. Researchers are working such that they can provide the sensing abilities in a single chip which is still a challenge for them [49]. It is a very delicate, intricate and a complex maneuver to perform, and it requires a lot of research work and professional expertise.

3. Open Source in Robotic Systems

In this section, we will discuss the open-source and how it paved its way in robotics. Open-source is an availability of something publicly which people can modify as required. There are number of open-source software and hardware available out there which includes, Linux, Android, blender, Arduino, raspberry, OSHW *etc.* Similarly, open-source is popular in robotics too, there are number of software, hardware, and simulators are available related to the development of robotics systems, which are open-source, for example, LeJOS, Rock, ROS, Sparki, TurtleBot, Gazebo, Morse, V-rep *etc.* Open-Source is divided into main three types; hardware, software, and simulator as shown in Fig. 5. The nature of these open-source platforms is shown in TABLE 1.

3.1. *Open-Source Hardwares*

A robot is a physical entity that is comprised of both hardware and software. A decade or more ago when all the robots hardware and software were under the hood of licensing and copyright, it was difficult

Table 1: This table shows some of the online available open-source platforms that are either categorize as software, hardware and simulator. It also states their online status, their visitations and popularity among the online folks.

Platform	Hardware	Software	Simulator	Page Views (Yearly)	Global Rank	Search Tarrif
ROS [40]	No	Yes	No	5,964,465	13,634	△ 6.000%
ROCK [41]	No	Yes	No	29,565	2,098,604	-
LeJos [45]	No	Yes	No	163,520	817,770	△ 9.000%
Gezebo [46]	No	No	Yes	89,060	119,064	▽ 22.00%
Morse [51]	No	No	Yes	74,825	1,075,294	△ 8.000%
V-rep [52]	No	No	Yes	74,825	120,629	▽ 22.00%
Sparki [63]	Yes	No	No	74,460	1,211,483	▽ 24.00%
Turtlebot [64]	Yes	No	No	434,350	377,227	△ 5.000%
Arcbot [65]	Yes	No	No	226,665	686,577	△ 131.0%

for the young developers to get their hands dirty with the robotic stuff. Fortunately, now there is a number of open-source hardware available to get started with robots. Sparki is a completely open-source platform available for the young nerds who have interest in robotics. It offers schematics and 3D design, all the source code is available with it. Their robotic kit includes; distance sensors, IMUs, light sensor, infrared communication, line follower, *etc.* OpenPilot is another open-source platforms that provide the users with the drones, quadcopter, copter, plane, rover, *etc.* Their aim is the making of unarmed, autonomous flying vehicles for the benefit of people. Sources like copter and ardupilot also serve the same purpose. TurtleBot is another open-source project initiated by Melonee Wise in 2010. It is a low-cost robot kit along with it an open-source software. The kit is sufficient enough to help you in making a good looking robot, with a good camera for monitoring and has enough horse-power to do an exciting job[50].

3.2. *Open-Source Softwares*

A robot is more than just a metallic body, and it also needs a spirit to walk around and perform other life-like duties. In the technological sense, that spirit for the robot is the software that allows them to perform the intelligent tasks. How intelligent a robot depends on its software although hardware does play its part too, but software plays a key role, from basic movement to the highly intelligent moves all are programmed through software. There is a number of open-source software available to facilitate the beginners as well as the professionals, such as LeJOS, Rock, ROC, *etc.*

LeJOS, is an open-source platform and a tool for the beginner who is new to the robotics fields, it is easy to use. It is an alternative for the firmware of the Mindstorms robots. A user can simply program the Mindstorms robots using Java programming language. Rock, is another open-source firmware for robotics, and it is based on Orocos RTT. It is expandable it accommodates a number of drivers and extensions of other applications. The key features of Rock are sustainability, scalability, reusability.

Sustainable System: It is designed keeping in view the long living systems those who have long durability.

Scalability: It is designed to handle big and complex systems without any fuss, and it does not want its users to learn the complexity behind those functions right away. Tools like oroGen guarantee the integration of different components using C++ and Ruby.

Reusable Codebase: The database of the code is independent of its firmware. It means that one can simply take the drivers, codes, localization algorithms and can integrate it with different firmware without any difficulty.

ROC (Robot Operating System), is an operating system for the robotics development. It is a firmware that includes different tools and libraries which simplifies the process. It is designed for the collaborative development. It allows the developers to design the robotics application, device drivers, message-passing, libraries, hardware abstraction, *etc.* It is licensed by BSD and sponsored by open source robotics foundation.

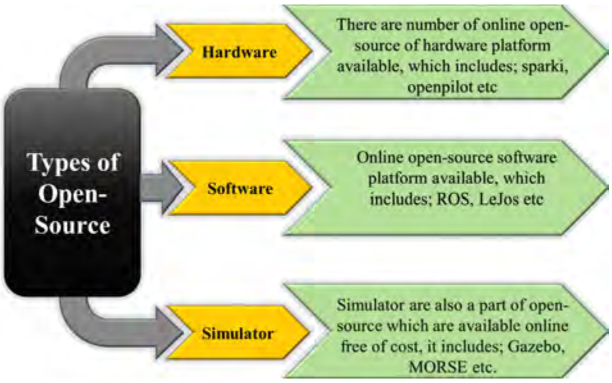
Types of Open-Source
Hardware
There are number of online open-source of hardware platform available, which includes; sparki, openpilot etc
Software
Online open-source software platform available, which includes; ROS, LeJos etc
Simulator
Simulator are also a part of open-source which are available online free of cost, it includes; Gazebo, MORSE etc.

Figure 5: Three basis types of open-sources that are available on the internet and those are, hardware, software and the simulators, and we can sub categorize them to further narrow down the things.

3.3. *Open-Source Simulators*

Robotics is all about planning and its implementation. Sometimes a user builds a good metallic body and breath some code in it too, but it fails the testing phase, that is mainly because of the lack of planning. To make sure that everything goes smoothly it is necessary to simulate the hardware and software before making any practical advancements. Here come the robotics simulators for this purpose, test your code before getting into real-world fuss. Just because everything looks good on paper does not mean that it will go straight in reality as well. There is a number of open-source simulators are available for this purpose that you can test your design according to the conditions that your robot will face in the real world and on the basis of the results, you can change the design as well. Some of those simulators are; Gazebo, MORSE, V-REP, *etc.* Gazebo is an Apache-licensed and offers a complete simulating environment with the 3D world, virtual sensors, broad command line and allows the developers to test their simulation in clouds as well. It allows the developers to test algorithms, build robots with an intelligent AI and test the robots under a realistic scenario and environment. It also allows the users to even simulate the robot army with each robot having its own functionality.

4. Open-Source Projects on Robotics

In this section, we will discuss the scope of open-source in robotics and will discuss in detail the types of robotics projects that are being done with the help of open-source, and the domain of the different project is shown in Fig. 6.

4.1. *Open-Source Projects on Robot Sensing*

Robot Sensing is a vital field of interest. It includes the involvement of sensors in the robot that includes, IMUs, pressure sensor, camera, etc. These data extracted from these sensors are then processed under intelligent algorithms to produce a state of the art of machine [54–56].

4.1.1. *Slam Projects*

SLAM (Simultaneous Localization and Mapping) is a robotic project with a purpose to map the unknown environment and keep the track of the robot in it. There is a number of algorithms developed to solve this problem which includes extended Kalman filter, particle filter and graph slam, *etc.* There are a number of open-source websites available where this problem is already solved, in a pretty advanced way and is available to the public. Github is one of the core sources for the robotics and other technological projects. It is an open-source platform where every user can upload his accomplishment so that other people can take advantage of it. Back to SLAM, GitHub offers a number of slam projects that are implemented in different ways with different algorithms we will briefly discuss them. Mono SLAM, is an open-source library developed under the hood of C++ for SLAM and it was designed by the Andrew Davison at the University of Oxford. PTAM (Parallel Tracking and Mapping), is an augmented based reality based open-source project that tracks the camera. It does not require any prerequisite data, map, templates and IMU-sensors information. This software is developed for the augmented reality, vision and SLAM. ORB SLAM is another open-source project implemented to give an accurate solution of SLAM. It is a versatile solution for the RGB-D camera, monocular and stereo problem. It has an advanced feature of constructing 3D modal of the scenes and environment around in a very detailed manner. It can construct within the environment of a car as well as the city blocks around the driven car. It ables the localization of the robot within the environment with high-precision. It offers a robust initialization from planer and non-planer scenes.

4.1.2. AprilTags Projects

AprilTags are the 2D bar-codes library designed to give robots a unique ID. Each bar-code has an identity associated with it which can be identified from the image. If the cameras are well calibrated and the relative position between the tag and the camera is known then one can easily identify the targeted robot. It was originally developed by Edwin Olson and was further modified by Michael Kaess, they introduced some new features to it which includes:

1. Made it a standalone library.
2. Using Open CV included stable homography recovery.
3. Principle point is essential for the homography so they added to it.
4. Debugging code was also added to it.
5. Included the approximation of arctan2 which was faster than Ed Olson's approximation.
6. For Quad computation homography is employed instead of the interpolation.

4.1.3. Computer Vision Projects

Computer Vision is actually sensing the world with the eye of the camera, analyzing it, extracting useful information from it, processing it and perform some useful task from it. It has made its way in robotics because a number of advanced projects in robots these days use computer vision and apply learning algorithms to it. Similarly, there are a number of open-source platforms out there on the Internet to facilitate the users and as mentioned earlier GitHub is one of them, other includes; robot shop, hackaday, introrobotics, *etc.* Sign Language Translator, is an open-source project made by Wick, J. (2018, April 03). Sign Language Translator. Retrieved from https://hackaday.io/project/116282-sign-language-translator, this project is about the transforming sign language into verbal communication. The purpose of this project is to facilitate the deaf people of our community, who feels difficulty in connecting with the world. This project aims to convert their signs into words so that a lay-man person who has no knowledge about the signs can understand them. Considering the practical aspect of this project, the core tool of this project is computer vision. One some sign is made in front of the camera that corresponding image is sent to the raspberry pie, the brain of this project, it will match the image with the pre-loaded database of images and after recognizing the sign it will send the corresponding text command to a text-to-speech application that converts the text into the sound.

Autonomous recharging and docking for multirotors, is another open-source project made by Giovanni, D. (2014, August 21). Autonomous recharging and docking for multirotors. Retrieved from https://hackaday.io/project/2759-autonomous-recharging-and-docking-for-multirotors, the purpose of this project is to guide the copters to autonomously fly from the initial point to the point where it requires charging and then again dock-off from that point to the targeted location. The problem with the normal GPS (Global Positioning System) is that it is accurate within +-5m which means that in case copter needs to get a recharge on its way to mission than with that tolerance level there is a chance that robot lands at wrong position, so one needs a human operator to guide the system. Recent advancement in low-power high-performance computing and computer vision, it will allow the copter to automatically dock-off from the initial point and in case of recharging it will land on the required spot autonomously. ImportWorld, is another project made by, File, K. (2015, January 22). ImportWorld. Retrieved from https://hackaday.io/project/4034-importworld, it is a 3D scanner project that uses laser scanners as well as

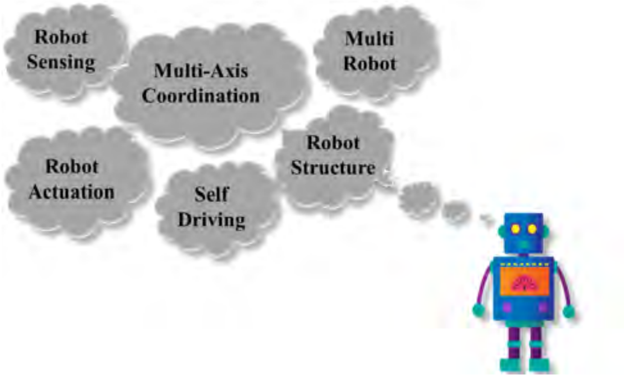

Figure 6: Core fields of robots that are being facilitated by the open-source, otherwise at every basic level open-source has given a lot to the educational institutes and the nerds who are investing themselves in the robotic filed.

cameras, computer vision helps to triangulate the object, the hardware employed in this project is teensy 3.1 that communicates with the computer and formulate the 3D image of the object there.

4.2. Open-Source Projects on Robot Actuation

There is a number of open-source projects available on motor controlling techniques and on self-driving, which shows that one can achieve useful results, using simple tools. Open-source allows the developers to connect different dots and build something innovative, project includes; 3D printer using stepper motors, motor PID speed controller, CNC machine using stepper motors, motor throttle controller, self-driving cars, *etc.* There are a number of open-source platforms out there for such projects, Instructables, Arduino, and electronicsforu are one of those facilitating platforms.

Motor Control Projects: There are open-source projects which include the wonders different users have done using simple motors and controllers. The prototype of Arduino Uno CNC Machine is made by the Sistem, K. (2018, March 21). The prototype of Arduino Uno CNC Machine. Retrieved from https://create.arduino.cc/projecthub/sistem-komputer, the purpose of the project is the availability of an alternative of CNC machine at a cheap price. CNC machine is used to for designing but even a small machine is very expensive. Through this project, they have made an alternative of CNC using Arduino and two stepper motors, and the speed of the motors was controlled very carefully for this matter.

Drive with PID control on an Arduino, is posted by, Hans, S. (2017, September 06). Drive with PID control on an Arduino Mega 2560. Retrieved from https://create.arduino.cc/projecthub/nothans, all the schematic, step by step procedure and source code is available at the mentioned link. The purpose of the project was the control of robot wheels using PID controller, if all the four wheels of the car or robot have it's on motor then its very hard for all four of them to get synchronized, because of there mechanical, electrical and manufactural differences. In this project the applied a feedback system, *i.e.*, PID controller, to control all the four wheels in a way that all of them remain synchronized.

Outboard Motor Throttle Controller, is another open-source project made by, the schematic, step-wise procedure and the code of the project is available at the mentioned source. It is an interesting project that is related to fishing. According to the developer, it is specially designed to catch the trout fish, since it has the ability to change is speed on the daily basis so it makes it difficult and cumbersome for the fisherman to catch them so this simple machine will them in catching the fish.

Self-Driving Projects: Self-driving is one of the disruptive technology that is inserting its claws into the modern field of robotics. It allows the machine to perform one of the most difficult tasks, *i.e.*, driving autonomously. There is a number of big companies, which are driving their way into this field and having somehow achieved many promising results. It makes our dream on a smart city without human drivers closer and closer. It not only looks fancy but extremely challenging, as it requires very delicate and intricate maneuver to perform great research expertise. Thanks to open-source platforms, which includes, Udacity, GitHub, instructable, *etc.*

4.3. Multi-Robot Communication and Cooperation Projects

Multi-robot or swarm robots are the armies of the robot that is another emerging field of the robots, and they are designed to perform special tasks that a single robot cannot perform. Their application scope includes; military, industries, rescue services, monitoring of an area affected by the earthquake or other natural disasters, espionage activities, *etc.* Nowadays, researchers are paying attention to this field as well, intelligent communication between the robots is one of the core problems that they are tackling, thanks to the open source because it not only broadens the community who can tackle this problem but it also allows them to discuss the problem openly. There are a number of platforms available which includes; Wikihow, instructables, Arduino are some of the platforms which explain in detail how to make the swarm robots.

5. Conclusion

In this paper, we have discussed the open-source platforms, their role in promoting technology, more specifically robotics. We have also discussed the robots their basic elements which turn them into intelligent machines, their basic and advance controlling algorithms. In addition, we have explored different families of robots. Then we have explored in detail about the open-source and its impact on the community, and have explored their different online available software and hardware platforms. At last, we have discussed the different projects done in robotics under the hood of the open-source.

References

[1] Bastian, Mathieu, Heymann, Sebastien, Jacomy, and Mathieu, "Gephi: an open source software for exploring and manipulating networks," *International AAAI Conference on Web and Social Media*, vol. 8, pp. 361–362, 2017.

[2] Giannozzi, Paolo, Baroni, Stefano, Bonini, Nicola, Calandra, Matteo, Car, Roberto, Cavazzoni, Carlo, Ceresoli, Davide, Chiarotti, L. Guido, Cococcioni, Matteo, Dabo, and Ismaila, "QUANTUM ESPRESSO: a modular and open-source software project for quantum simulations of materials," *Journal of physics: Condensed matter*, vol. 21, no. 39, pp. 395502, 2009.

[3] Feller, Joseph, Fitzgerald, and Brian, "Understanding open source software development," *Addison-Wesley London*, 2002.

[4] Oostenveld, Robert, Fries, Pascal and Maris, Eric and Schoffelen, and Jan-Mathijs, "FieldTrip: open source software for advanced analysis of MEG, EEG, and invasive electrophysiological data," *Computational intelligence and neuroscience*, vol. 2011, pp. 1, 2011.

[5] Hippel, V. Eric and Krogh, and Georg von, "Open source software and the "private-collective" innovation model: Issues for organization science," *Organization science*, vol. 14, n0. 2, pp. 209–223, 2003.

[6] Jaeger, Till and Metzger, and Axel, "Open Source Software," *Rechtliche Rahmenbedingungen der Freien Software*, vol.2, pp.51, 2011.

[7] Lakhani, R. Karim and V. Hippel, and Eric, "How open source software works:"free" user-to-user assistance," *Research policy*, vol. 32, no. 6, pp. 923–943, 2003.

[8] Lakhani, R. Karim and Wolf, and Robert G, "Why hackers do what they do: Understanding motivation and effort in free/open source software projects," *MIT press*, 2003.

[9] GitHub. (2018). Marlin. [online] Available at: https://github.com/MarlinFirmware.

[10] Delp, L. Scott, Anderson, C. Frank, Arnold, S. Allison, Loan, Peter, Habib, Ayman, John, T. Chand, Guendelman, Eran, Thelen, G. Darryl, "OpenSim: open-source software to create and analyze dynamic simulations of movement," *IEEE transactions on biomedical engineering*, vol. 54, no. 11, pp. 1940–1950, 2007.

[11] Cliff, Dave, Husbands, Phil, Harvey, and Inman, "Explorations in evolutionary robotics," *Adaptive behavior*, vol. 2, no. 1, pp. 73–110, 1993.

[12] Kamei, Koji, Nishio, Shuichi, Hagita, Norihiro, Sato, and Miki, "Cloud networked robotics," *IEEE Network*, vol. 26, no. 3, 2012.

[13] W. Durrant, and F. Hugh, "Uncertain geometry in robotics," *IEEE Journal on Robotics and Automation*, vol. 4, no. 1, pp. 23–32, 1988.

[14] McKee, and T. Gerard, "The development of Internet-based laboratory environments for teaching robotics and artificial intelligence," *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, vol 3, pp. 2695–2700, 2002.

[15] Hallstedt, I. Sophie, Thompson, W. Anthony, Lindahl, and Pia, "Key elements for implementing a strategic sustainability perspective in the product innovation process," *Journal of Cleaner Production*, vol. 51, pp. 277–288, 2013.

[16] Hutter, Marco, Gehring, Christian, Bloesch, Michael, Hoepflinger, A. Mark, Remy, C. David, Siegwart, and Roland, "StarlETH: A compliant quadrupedal robot for fast, efficient, and versatile locomotion," *Adaptive Mobile Robotics*, pp. 483–490, 2012.

[17] Chen, L. Qing, Qu, Q. Xiao, Hou, B. Huei, Sosso, Davide, Osorio, Sonia, Fernie, R. Alisdair, Frommer, B. Wolf, "Sucrose efflux mediated by SWEET proteins as a key step for phloem transport," *Science*, vol. 335, no. 6065, pp. 207–211, 2012.

[18] Larson, J. Wiley, Pranke, K. Linda, "Human spaceflight: mission analysis and design," *McGraw-Hill Companies*, 1999.

[19] Baeten, Johan, Donn, Kevin, Boedrij, Sven, Beckers, Wim, Claesen, and Eric, "Autonomous fruit picking machine: A robotic apple harvester," *Field and service robotics*, pp. 531–539, 2008.

[20] S. Ruiz, Javier, Avilés, and Roberto, "Robotics courses for children as a motivation tool: the Chilean experience," *IEEE Transactions on Education*, vol. 47, no. 4, pp. 474–480, 2004.

[21] Craig, and J. John, "Introduction to robotics: mechanics and control," *Pearson/Prentice Hall Upper Saddle River, NJ, USA*, vol. 3, 2005.

[22] Chaumette, Franois, Hutchinson, and Seth, "Visual servo control. I. Basic approaches," *IEEE Robotics & Automation Magazine*, vol. 13, no. 4, pp. 82–90, 2006.

[23] Fu, S. King, Gonzalez, Ralph, Lee, and C. George, "Robotics: Control Sensing. Vis," *Tata McGraw-Hill Education*, 1987.

[24] Jaulin, and Luc, "Applied interval analysis: with examples in parameter and state estimation, robust control and robotics," *Springer Science & Business Media*, vol. 1, 2001.

[25] Elci, Haluk, Longman, W. Richard, Phan, Q. Minh, Juang, N. Jer, Ugoletti, Roberto, "Simple learning control made practical by zero-phase filtering: Applications to robotics," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 49, no. 6, pp. 753–767, 2002.

[26] Hewit, and Burdess, "Fast dynamic decoupled control for robotics, using active force control," *Mechanism and Machine Theory*, vol. 16, no. 5, pp. 535–542, 1981.

[27] Spong, W. Mark, Hutchinson, Seth, Vidyasagar, and Mathukumalli, "Robot modeling and control,"

[28] Arkin, and C. Ronald, "Behavior-based robotics," *MIT press*, 1998.

[29] S. Olfati, and Reza, "Nonlinear control of underactuated mechanical systems with application to robotics and aerospace vehicles," *Massachusetts Institute of Technology*, 2001.

[30] Hirose, Shigeo, Morishima, and Akio, "Design and control of a mobile robot with an articulated body," *The International Journal of Robotics Research*, vol. 9, no. 2, pp. 99–114, 1990.

[31] Chevallereau, Christine, Abba, Gabriel, Aoustin, Yannick, Plestan, Franck, Westervelt, Eric, D. Wit, Carlos, Canudas, Grizzle, and Jessy, "Rabbit: A testbed for advanced control theory," *IEEE Control Systems Magazine*, vol. 23, no. 5, pp. 57–79, 2003.

[32] Chevallereau, Christine, Abba, Gabriel, Aoustin, Yannick, Plestan, Franck, Westervelt, Eric, D. Wit, C. Carlos, Grizzle, and Jessy, "Rabbit: A testbed for advanced control theory," *IEEE Control Systems Magazine*, vol. 23, no. 5, pp. 57–79, 2003.

[33] Chaumette, Franois, Hutchinson, and Seth, "Visual servo control. II. Advanced approaches [Tutorial]," *IEEE Robotics & Automation Magazine*, vol. 14, no. 1, pp. 109–118, 2007.

[34] Liegeois, and Alain, "Automatic supervisory control of the configuration and behaviour of multibody mechanisms," *IEEE Transactions on systems, man and cybernetics*, vol. 7, no. 12, pp. 868–871, 1997.

[35] Dounis, Anastasios, Caraiscos, and Christos, "Advanced control systems engineering for energy and comfort management in a building environment—A review," *Renewable and Sustainable Energy Reviews*, vol. 13, no. 6-7, pp. 1246–1261, 2009.

[36] Arimoto, and Suguru, "Control Theory of Non-Linear Mechanical Systems: A Passivity-Based and Circuit-Theoretic Approach (Oxford Engineering Science Series)," *Oxford University Press*, 1996.

[37] Kurfess, and R. Thomas, "Robotics and automation handbook," *CRC press*, 2004.

[38] Ohnishi, Kouhei, Murakami, and Toshiyuki, "Advanced motion control in robotics," *Industrial Electronics Society, 1989. IECON'89., 15th Annual Conference of IEEE*, pp. 356–359, 1989.

[39] J. Cannon, H. Robert, Schmitz, and Eric, "Initial experiments on the end-point control of a flexible one-link robot," *The International Journal of Robotics Research*, vol. 3, no. 3, pp. 62–75, 1984.

[40] Ros.org. (2018). ROS.org — *Powering the world's robots.* [online] Available at: http://www.ros.org/.

[41] team, T. (2018). [online] Rock-robotics.org. Available at: https://www.rock-robotics.org/.

[42] Baker, an R. Daniel, "Some topological problems in robotics," *The Mathematical Intelligencer*, vol. 12, no. 1, pp. 66–76, 1990.

[43] Mombaur, D. Katja, Bock, G. Hans, Schlder, P. Johannes, Longman, and W. Richard, "Open-loop stable solutions of periodic optimal control problems in robotics," *ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik*, vol. 85, no. 7, pp. 499–515, 2005.

[44] Dario, Paolo, Rucci, and Michele, "An approach to disassembly problems in robotics," *Intelligent Robots and Systems' 93, IROS'93. Proceedings of the 1993 IEEE/RSJ International Conference on*, vol. 1, pp. 460–467, 1993.

[45] Moral, J. (2018). LeJOS, Java for Lego Mindstorms. [online] Lejos.org. Available at: http://www.lejos.org/.

[46] Gazebosim.org. (2018). Gazebo. [online] Available at: http://gazebosim.org/.

[47] Houk, C. James, Davis, L. Joel, Beiser, G. David, "Models of information processing in the basal ganglia," *MIT press*, 1995.

[48] Siciliano, and Bruno, "Control in robotics: open problems and future directions," *Control Applications, 1998. Proceedings of the 1998 IEEE International Conference on*, vol. 1, pp. 81–85, 1998.

[49] Demange, Marc, Ekim, Tinaz, D. Werra, and Dominique, "A tutorial on the use of graph coloring for some problems in robotics", *European Journal of Operational Research*, vol. 192. no. 1, pp. 41–55, 2009.

[50] D. Sapio, Vincent, Khatib, Oussama, Delp, and Aott,"Least action principles and their application to constrained and task-level problems in robotics and biomechanics", *Multibody System Dynamics*, vol. 19, no. 3, pp. 303–322, 2008.

[51] Openrobots.org. (2018). OpenRobotsFrontPage - Openrobots. [online] Available at: http://www.openrobots.org/.

[52] Coppeliarobotics.com. (2018). Coppelia Robotics V-REP: Create. Compose. Simulate. Any Robot.. [online] Available at: http://coppeliarobotics.com/.

[53] Tikhanoff, Vadim, Cangelosi, Angelo, Fitzpatrick, Paul, Metta, Giorgio, Natale, Lorenzo, Nori, and Francesco, "An open-source simulator for cognitive robotics research: the prototype of the iCub humanoid robot simulator," pp. 57–61, 2008.

[54] Zykov, Victor, Phelps, Lassabe, Nicolas, Lipson, and Hod, "Molecubes extended: Diversifying capabilities of open-source modular robotics," *IROS-2008 Self-Reconfigurable Robotics Workshop*, pp. 22–26, 2008.

[55] Meier, Lorenz, Honegger, Dominik, Pollefeys, and Marc, "PX4: A node-based multithreaded open source robotics framework for deeply embedded platforms,"
emphRobotics and Automation (ICRA), 2015 IEEE International Conference on, pp. 6235–6240, 2015.

[56] Bruyninckx, and Herman, "Open robot control software: the OROCOS project," *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, vol. 3, pp. 2523–2528, 2001.

[57] Delp, L. Scott, Anderson, C. Frank, Arnold, S. Allison. Loan, Peter, Habib, Ayman, John, T. Chand, Guendelman, Eran, Thelen, G. Darryl, "OpenSim: open-source software to create and analyze dynamic simulations of movement", *IEEE transactions on biomedical engineering*, vol. 54, no. 11, pp. 1940–1950, 2007.

[58] Lim, Hyon, Park, Jaemann , Lee, Daewon, Kim, H. Jin, "Build your own quadrotor: Open-source projects on unmanned aerial vehicles," *IEEE Robotics & Automation Magazine*, vol. 19, no. 3, pp. 33–45, 2012.

[59] Nakadai, Kazuhiro, Takahashi, Toru Okuno, G. Hiroshi, Nakajima, Hirofumi, Hasegawa, Yuji, Tsujino, and Hiroshi, "Design and Implementation of Robot Audition System'HARK'—Open Source Software for Listening to Three Simultaneous Speakers," *Advanced Robotics*, vol. 24, no. 5-6, pp. 739–761, 2010.

[60] Craighead, Jeff, Murphy, Robin, Burke, Jenny, Goldiez, and Brian, "A survey of commercial & open source unmanned vehicle simulators," *Robotics and Automation, 2007 IEEE International Conference on*, pp. 852–857, 2007.

[61] Honegger, Dominik, Meier, Lorenz, Tanskanen, Petri, Pollefeys, and Marc, "An open source and open hardware embedded metric optical flow cmos camera for indoor and outdoor applications," *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pp. 1736–1741, 2013.

[62] O'Hara, J. Keith, Kay, and S. Jennifer, "Investigating open source software and educational robotics, *Journal of Computing Sciences in Colleges*, vol. 18, no. 3, pp. 8–16, 2003.

[63] Arcbotics.com. (2018). ArcBotics - Learn robotics and coding the easy way. [online] Available at: http://arcbotics.com/.

[64] Turtlebot.com. (2018). TurtleBot. [online] Available at: https://www.turtlebot.com.

[65] Arcbots.com. (2018). Home / xat bot - ARCbots [Advanced Really Cool Bots] - Number one used and trusted xat bots. [online] Available at: https://arcbots.com/.

[66] Sarabia, Miguel, Ros, Raquel, Demiris, and Yiannis, "Towards an open-source social middleware for humanoid robots," *Humanoid Robots (Humanoids), 2011 11th IEEE-RAS International Conference on, IEEE*, pp. 670–675, 2011.

[67] Metta, Giorgin, Sandini, Giulio, Vernon, David, Natale, Lorenzo, Nori, and Francesco, "The iCub humanoid robot: an open platform for research in embodied cognition," *Proceedings of the 8th workshop on performance metrics for intelligent systems*, pp. 50–56, 2008.

[68] Chen, Zihan, Deguet, Anton, Taylor, Russell, DiMaio, Simon, Fischer, Gregory, Kazanzides, and Peter, "An open-source hardware and software platform for telesurgical robotics research," *Proceedings of the MICCAI Workshop on Systems and Architecture for Computer Assisted Interventions, Nagoya, Japan*, pp. 22–26, 2013.

[69] Kozak, Kris, Zhou, Qian, Wang, and Jinsong, "Static analysis of cable-driven manipulators with non-negligible cable mass," *IEEE Transactions on Robotics*, vol. 22, no. 3, pp. 425–433, 2006.