

PAPER • OPEN ACCESS

## A web application for remote control of ROS robot based on WebSocket protocol and Django development environment

To cite this article: Zinaid Kapi *et al* 2021 *IOP Conf. Ser.: Mater. Sci. Eng.* **1208** 012035

View the [article online](#) for updates and enhancements.

### You may also like

- [The Trigger and Data Acquisition System for the KM3NeT-Italy neutrino telescope](#)  
T Chiarusi, M Favaro, F Giacomini et al.
- [APFEL Web: a web-based application for the graphical visualization of parton distribution functions](#)  
Stefano Carrazza, Alfio Ferrara, Daniele Palazzo et al.
- [A Development of Lightweight Grid Interface](#)  
G Iwai, Y Kawai, T Sasaki et al.



The Electrochemical Society  
Advancing solid state & electrochemical science & technology

## 241st ECS Meeting

May 29 – June 2, 2022 Vancouver • BC • Canada

Extended abstract submission deadline: Dec 17, 2021

Connect. Engage. Champion. Empower. Accelerate.  
**Move science forward**



**Submit your abstract**



# A web application for remote control of ROS robot based on WebSocket protocol and Django development environment

**Zinaid Kapić, Aladin Crnkić, Edin Mujčić and Jasna Hamzabegović**

University of Bihać, Faculty of Technical Engineering, Bosnia and Herzegovina

E-mail: zinaid.kapic@unbi.ba

**Abstract.** The development of teleoperation systems, robots, or any physical part of the system can be costly and if something goes wrong it can lead to development overdue. Precisely for these reasons, engineers and scientists today resort to the development of simulated systems before the construction of a real system. Robot Operating System (ROS) is one of the most popular solutions for robot development, manipulation, and simulation. In this paper, we present a web application for remote control of a ROS robot. The robot is controlled via a web application that is used as a virtual Joystick. Also, in this paper, an experimental work analysis of the projected system is performed. Further research possibilities include upgrading the presented web interface, adding certain motion autonomy sensors, or integrating some path planning algorithms.

## 1. Introduction

Teleoperations are present in all spheres of human activity. Many endeavors of human civilization are inconceivable without the presence of teleoperations, ie remote control. Exploring impassable remote places, doing hard work, reducing the physical distance of people, or some everyday activities involve some form of teleoperation [1-3]. The choice of media for teleoperations is extremely important for the development of the teleoperating system, and the Internet is certainly one of the most popular ways of transmitting information [4]. Teleoperations over the Internet also depend on the choice of protocol for transmitting the information. The protocol used in this paper is the popular WebSocket protocol, which has recently been imposed as a standard in working with the Internet. The speed and possibility of full-duplex communication are especially emphasized [5]. The WebSocket protocol is mainly used in applications that include real-time chat applications, live broadcasting, and in recent works, it found its applicability in teleoperation systems and is used for ROS communication [6].

The development of teleoperation systems, robots, or any physical part of the system can be expensive and sometimes time-consuming. Precisely for these reasons, scientists today resort to the development of simulated systems before the construction of a real system. In this way, the system and its environment can be modeled sufficiently to examine the validity of the physical construction of that system. The necessity for this kind of development produced several robotics middleware platforms, such as OROCOS [7], Player [8], YARP [9], ROS [10], etc.

ROS (Robot Operating System) is a meta-operating system with a huge library and support for the development of robots and similar systems. It is this software that in recent years has become synonymous with the development of robots, and is increasingly advancing and gaining in importance



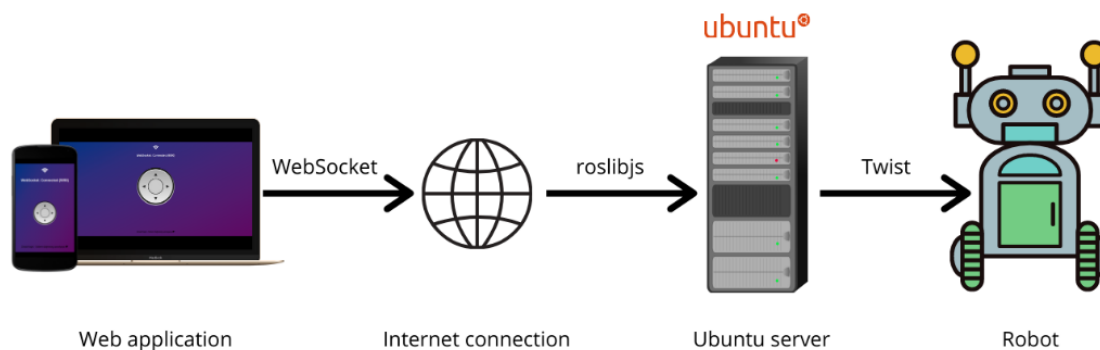
[11]. Since its beginning in 2007, ROS has found application in almost every robotic field, from mining, healthcare, home automation, and others [12-16].

It enables the creation of sufficiently good simulations of various systems, including teleoperations, before the realization of the system itself. It is based on the C++ and Python programming languages, and the code created within this system can be transferred directly to the physical robot [11]. Through this paper, we have developed an application for direct teleoperation based on the master-slave system. The master system is a joystick internet application that controls the movement of the ROS simulated robot. The communication between a web application and a ROS robot is based on the WebSocket protocol. The application is implemented using Python programming language and the Django development environment.

In the next chapters, we present an implementation process with a brief explanation of all parts used in developing an application and connecting it with a simulation. After that, we display an experimental analysis of our application. Finally, we present a conclusion and possible upgrades of this application.

## 2. Implementation of the proposed system

The scheme of the implemented system is presented in figure 1. As is seen in the figure, the web application is using the internet as a medium for data transfer. Protocol for establishing communication and sending data is a WebSocket protocol. After establishing a successful connection with the ROS server, the user can send commands consisted of Twist structure messages.



**Figure 1.** Scheme of the implemented system

Messages that are sent consist of two ROS structures, named `geometry_msgs/Vector3` that are consisted of linear and angular float64 data. Twist messages are sent from a web application with `roslibjs` library [17].

The web application is implemented with a Django development environment. Django is a popular framework that uses Python programming language [18]. We have used Django because of the power that Python provides. Python is also a language used by ROS for developing robot functionality. By using Django as a web development tool, we achieve consistency, clean code, and limit a need for knowledge of multiple programming languages.

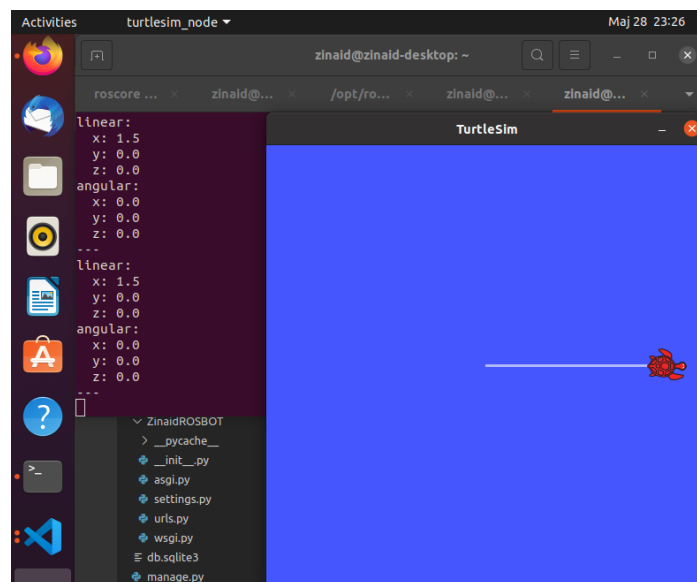
Besides that, Python exploits many libraries such as OpenCV, Numpy, Pandas, gmapping, PCL, and others [19]. These libraries are useful when developing simple or advanced web, mobile, or other applications integrated with ROS. There have been previous works that integrate Robot Operating System with Jupyter and Conda ecosystems [20], or even ReactJS [21]. Besides controlling mobile robots [22], ROS can be useful for working with 3D data, with applications that include rotations [23-24], and path planning algorithms [25].

The design of a web application is presented in figure 2. Application is built as a virtual joystick that is accessible through the browser and is responsive for multiple platforms, from desktop devices to mobile phones.



**Figure 2.** Application design

Virtual joystick sends commands for specific movements to a simulated ROS robot. In our case, we use the standard TurtleSim [26] robot for explaining purposes. Commands sent from a web application to TurtleSim simulation are Twist messages consisted of data for a specific movement. For example, forward movement is achievable with a message presented in figure 3.

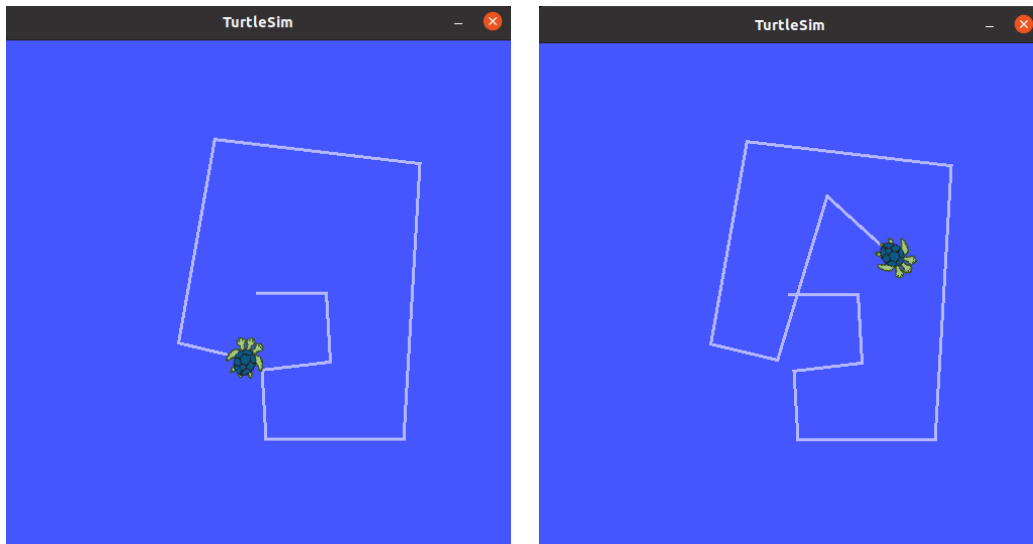


**Figure 3.** Forward movement

Forward movement, as is seen in Figure 3, can be achieved by sending a linear value of  $x = 1.5$  where 1.5 represents a movement's distance. The greater the value, the greater the movement. Reducing a value we reduce a distance of a movement and achieve finer-grained movements. Any other movement, linear or angular, is achieved in the same way.

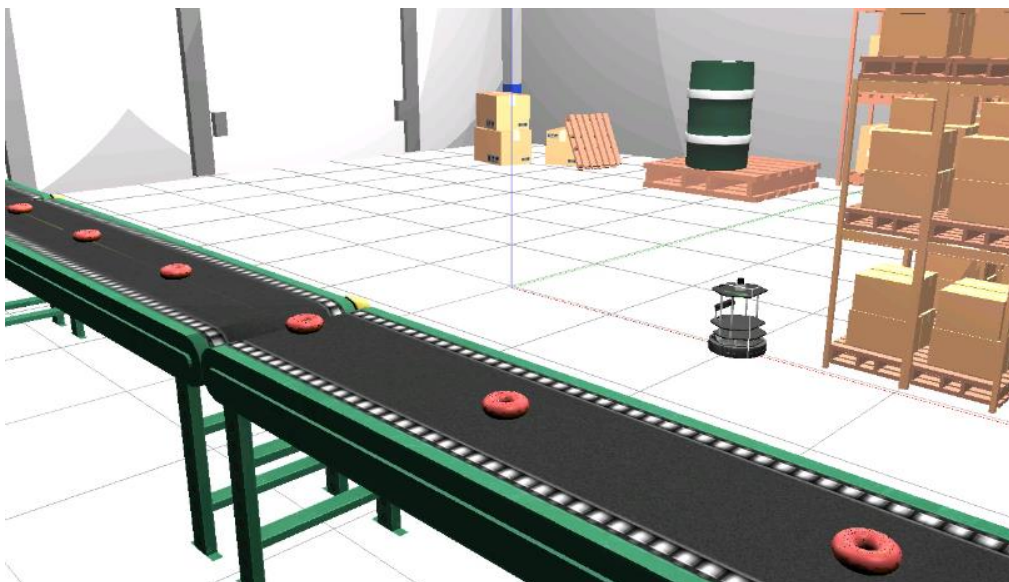
### 3. Simulations

The movement of a TurtleSim is presented in figure 4. Sending specific values of a Twist message to TurtleSim results in different movement paths.



**Figure 4.** TurtleSim movement

The strongest part of a Robot Operating System is its 3D visualizations. ROS visualization of a 3D TurtleSim is presented in figure 5. This simulation is created with a Gazebo library that is integrated into a ROS ecosystem [27].



**Figure 5.** Gazebo simulation

In the abovementioned figure, we present movements of a controlled TurtleSim in a 3D environment of a production factory. With these simulations, engineers can test the entire process of working with robots and their interactions with the environment.

#### 4. Conclusion

Teleoperations, and thus teleoperated robots, play an important role in the modern world. The possibility for simulations before the construction of a real system plays an important upgrade and provides beneficial time and money savings to engineers and scientists. Robot Operating System is currently the most popular tool for studying robotics, developing robot systems, and simulations.

In this paper, we have presented an application for internet communication with robots based on WebSocket protocol. Through experimental analysis of our application, we have concluded that WebSocket's exceptional speed and full-duplex communication is an advantage for teleoperation system development. A combination of web developing tools such as the Django framework, which exploits the strength of a Python programming language, with ROS software can provide fast testing and simulation development, before the construction of a real system.

Possible upgrades of this application are graphical interface improvements, integration of controllers for motion control, or adding sensors and path planning algorithms into a system for motion automation. In future work, we will also address the time delay problems caused by the internet connection.

#### References

- [1] Tomic T, Schmid K, Lutz P, Domel A, Kassecker M, Mair E, Grixia I, Ruess F, Suppa M and Burschka D 2012 Toward a fully Autonomous uav: Research platform for indoor and OUTDOOR Urban search and rescue *IEEE Robotics & Automation Magazine* **19** 46–56 doi: 10.1109/mra.2012.2206473
- [2] Su H, Yang C, Ferrigno G and De Momi E 2019 Improved Human–Robot collaborative control of Redundant robot for Teleoperated minimally invasive surgery *IEEE Robotics and Automation Letters* **4** 1447–53 doi: 10.1109/lra.2019.2897145
- [3] Drakulić U and Mujčić E 2019 Remote monitoring and control system for greenhouse based on IoT *Advanced Technologies, Systems, and Applications IV - Proceedings of the International Symposium on Innovative and Interdisciplinary Applications of Advanced Technologies (IAT 2019)* 481–95 doi: 10.1007/978-3-030-24986-1\_38
- [4] Pala M and Sincak P 2012 Towards the assisted teleoperation systems *2012 IV International Congress on Ultra Modern Telecommunications and Control Systems* doi: 10.1109/icumt.2012.6459715
- [5] Srinivasan L, Scharnagl J and Schilling K 2013 Analysis of websockets as the new age protocol for remote robot tele-operation *IFAC Proceedings Volumes* **46** 83–8 doi: 10.3182/20131111-3-kr-2043.00032
- [6] Sorrentino A, Cavallo F and Fiorini L 2020 A plug and play transparent communication layer for cloud robotics architectures *Robotics* **9** 17 doi: 10.3390/robotics9010017
- [7] Bruyninckx H 2001 Open robot control software: The OROCOS PROJECT *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No.01CH37164)* doi: 10.1109/robot.2001.933002
- [8] Vaughan R T and Gerkey B P 2007 Reusable robot software and the PLAYER/STAGE PROJECT *Springer Tracts in Advanced Robotics* 267–89
- [9] Fitzpatrick P, Ceseracciu E, Domenichelli D, Paikan A, Metta G and Natale L 2014 A middle way for robotics middleware *Journal of Software Engineering for Robotics* **5** 42–9
- [10] Quigley M, Conley K, Gerkey B, Faust J, Foote T, Leibs J, Wheeler R and Ng A Y 2009 ROS: An open-source robot operating system *ICRA workshop on open source software*
- [11] Koubaa A 2015 ROS As a Service: Web Services for Robot Operating System *Journal of Software Engineering for Robotics* **6**
- [12] Araújo A, Portugal D, Couceiro M S and Rocha R P 2014 Integrating arduino-based educational mobile robots in ros *Journal of Intelligent & Robotic Systems* **77** 281–98 doi: 10.1007/s10846-013-0007-4
- [13] Lleras N O, Brennan S, Murphy D, Klena M J, Garvey P and Sommer H 2016 Development of

- an open-source tractor driving simulator for tractor stability tests *Journal of Agricultural Safety and Health* **22** 227–46 doi: 10.13031/jash.22.11774
- [14] Bouchier P 2013 Embedded ros [ros topics] *IEEE Robotics & Automation Magazine* **20** 17–9 doi: 10.1109/mra.2013.2255491
- [15] Badger J, Gooding D, Ensley K, Hambuchen K and Thackston A 2016 Ros in space: A case study on robonaut 2 *Studies in Computational Intelligence* 343–73 doi: 10.1007/978-3-319-26054-9\_13
- [16] Kehoe B, Patil S, Abbeel P and Goldberg K 2015 A survey of research on cloud robotics and automation *IEEE Transactions on Automation Science and Engineering* **12** 398–409 doi: 10.1109/tase.2014.2376492
- [17] Tatarian K, Pereira S, Couceiro M S and Portugal D 2018 Tailoring a ros educational programming language architecture *Robotics in Education* 217–29 doi: 10.1007/978-3-319-97085-1\_22
- [18] Marani F 2018 Introduction to django *Practical Django 2 and Channels 2* 1–22 doi: 10.1007/978-1-4842-4099-1\_1
- [19] Millman K J and Aivazis M 2011 Python for scientists and engineers *Computing in Science & Engineering* **13** 9–12 doi: 10.1109/mcse.2011.36
- [20] Fischer T, Vollprecht W, Traversaro S, Yen S, Herrero C and Milford M 2021 RoboStack: Using the Robot operating system alongside the conda and Jupyter data Science Ecosystems *arXiv.org* arXiv:2104.12910
- [21] Ivanov A, Zakiev A, Tsoy T and Hsia K-H 2021 Online monitoring and visualization with ros and reactjs 2021 *International Siberian Conference on Control and Communications (SIBCON)* doi: 10.1109/sibcon50419.2021.9438890
- [22] Alisher K, Alexander K and Alexandr B 2015 Control of the mobile robots with Ros in Robotics Courses *Procedia Engineering* **100** 1475–84 doi: 10.1016/j.proeng.2015.01.519
- [23] Kapic Z, Crnkic A, Jacimovic V and Mijajlovic N 2021 A new dynamical model for solving rotation averaging problem 2021 *20th International Symposium INFOTEH-JAHORINA (INFOTEH)* doi: 10.1109/infoteh51037.2021.9400663
- [24] Kapic Z and Crnkic A 2021 Interpolating rotations with non-Abelian Kuramoto model on the 3-sphere *Advanced Technologies, Systems, and Applications VI - Proceedings of the International Symposium on Innovative and Interdisciplinary Applications of Advanced Technologies (IAT 2021)* doi: 10.1007/978-3-030-90055-7\_48
- [25] Galli M, Barber R, Garrido S and Moreno L 2017 Path planning USING Matlab-ROS integration applied to mobile robots 2017 *IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)* doi: 10.1109/icarsc.2017.7964059
- [26] Joseph L 2018 Programming with ros *Robot Operating System for Absolute Beginners* 171–236 doi: 10.1007/978-1-4842-3405-1\_5
- [27] Takaya K, Asai T, Kroumov V and Smarandache F 2016 Simulation environment for mobile robots testing using ros and gazebo 2016 *20th International Conference on System Theory, Control and Computing (ICSTCC)* doi: 10.1109/icstcc.2016.7790647