Omniwheel Chassis' Model and Plugin for Gazebo Simulator

Artem Apurin, Bulat Abbyasov, Alexandra Dobrokvashina
Intelligent Robotics Department, Kazan Federal University, 420111 Kazan, Russian Federation

Yang Bai, Mikhail Svinin
Information Science and Engineering Department, Ritsumeikan University, 1-1-1 Noji-higashi, Kusatsu, Shiga, 525-8577, Japan

Evgeni Magid
Intelligent Robotics Department, Kazan Federal University, 420111 Kazan, Russian Federation
E-mail: magid@it.kfu.ru
http://robot.kpfu.ru/eng

Abstract

Increasing a mobility of a robot in a limited space or in a presence of a large number of people is an important task. Mecanum wheels could provide the required high flexibility locomotion in any direction. This article presents a virtual model of an omniwheel robot in the Gazebo simulator. Freely rotating rollers were implemented to simulate the robot motion. We developed a four-wheel mecanum mobile robot plugin controls the robot by publishing linear velocity data along X and Y axes, and angular velocity data along Z-axis. The plugin could optionally publish a standard ground-truth odometry of the Gazebo or a calculated in real time wheel odometry. The open source code is extendable for similarly structured platforms and is available for a download via GitLab.

Keywords: Mecanum Wheels, Mobile Platform, Gazebo Simulator

1. Introduction

In the last decade, mobile robots with omnidirectional motion capabilities have been widely applied in various areas of a human life[1]. These mobile platforms are broadly employed in autonomous driving[2], manufacturing[3], industry[4], medical applications[5], social robotics[6], search and rescue[7], agriculture[8], education[9] and others. Omnidirectional movement is the ability to travel in any desired direction at any specified orientation[10]. To perform such locomotion an omnidirectional robot could use a special design of wheels called mecanum[11]. Mecanum wheels gain advantages in terms of maneuverability in narrow and crowded spaces[12]. A robot motion controller is decoupled and could perform translational and rotational motions.

Prior to a construction stage, it is important to model and simulate a new robot behavior in a virtual environment in order to decide on its efficiency and attempt to improve robot's mechanical and software features. Typically, roboticists prefer using standard simulators (e.g., ROS/Gazebo[13], Webots[14] or CoppeliaSim/V-REP[15]) while using simulators with a high-level of abstraction or developing in-house simulators are already considered to be an out-of-date approaches.

This paper presents a virtual model development of a four-wheeled omniwheel mobile robot LIRS-ArtBul[16] in the Gazebo simulator. Each mecanum wheel model has freely rotating rollers in order to perform omnidirectional motion. A motion control scheme was wrapped into a special four-wheel mecanum ROS plugin, which is fully configurable and can be easily reconfigured for researchers' tasks.

2. Modeling wheels and rollers

To achieve a high quality of a virtual model, it is important to take into account all the parameters and features of a corresponding real robot when creating its virtual representation for the simulator. The main task was to develop a model of an omniwheel robot with active freely rotating rollers. To create models Blender software was employed. A visual and collision parts of the wheel roller model are shown in Fig.1.

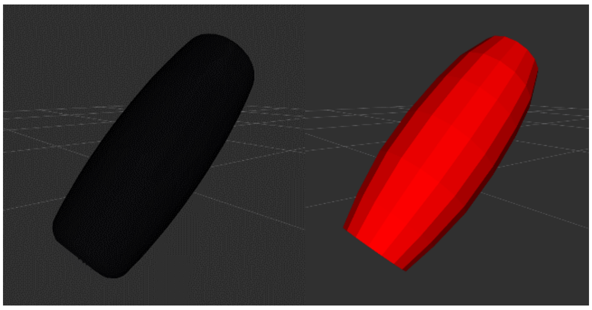


Fig.1.Wheel roller model in RViz: visual appearance (left) and collision (right) parts

To reduce complex collision calculations and increase real time factor (RTF), a low-poly model should be used. Roller's collision mesh with 220 polygons and a roller model for visual part with 7708 polygons allowed increasing RTF to 0.95-0.97. The wheel model in Gazebo is shown in Fig.2, each roller has its own joint and can be freely rotated along the Z-axis.

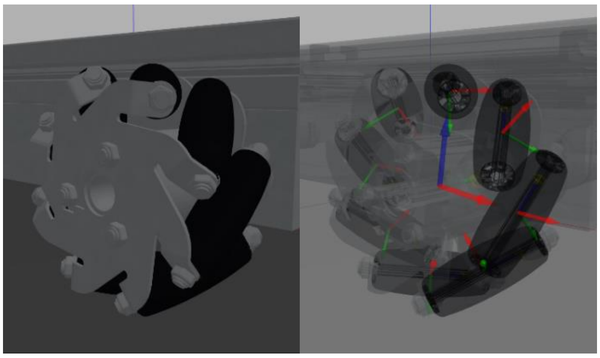


Fig.2.Wheel model in Gazebo

Four controllers of type *joint_velocity_controller* from *effort_controller* plugin were configured, since the real robot has a motor for each wheel. Package *joint_state_publisher* can be used to publish a position and a transformation of moving joints. *joint_state_publisher* connects to all non-fixed joints, but in this case *controller_manager* and *joint_state_publisher* packages publish positions of the duplicated joints to */joint_states* topic and multi-source publisher error occurs. Therefore, rollers' transmission parameters should be configured in URDF file, but controllers should not be created for these joints in YAML file. Then *controller_manager* publishes a position and a transformation of the roller joints and those joints could rotate freely. A mobile chassis with mecanum wheels in Gazebo is shown in Fig.3

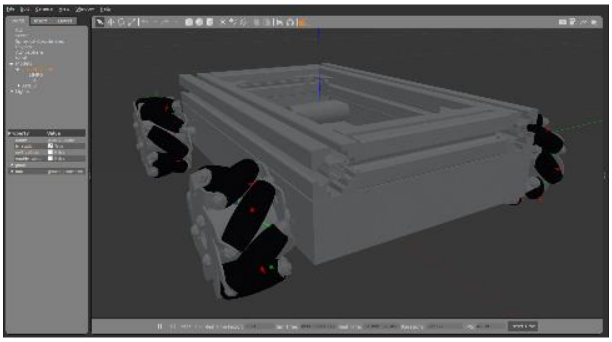


Fig.3.Mobile platform in Gazebo

3. Gazebo plugin for omniwheel robots

A plugin was developed to control the robot in Gazebo. It is suitable for all four-wheel mecanum mobile robots. The robot is controlled by publishing messages with linear velocities along the X and Y axes and an angular velocity along the Z axis to */cmd_vel* topic. From obtained data, the wheel velocities are calculated according to the equations (1) – (4)[17][18]:

$$v_{LF} = \frac{1}{R_{wheel}} \cdot \left(v_x - v_y - \left(\frac{s_{RL}}{2} + \frac{s_{FR}}{2}\right) \cdot v_z\right) \tag{1}$$

$$v_{RF} = \frac{1}{R_{wheel}} \cdot \left(v_x + v_y + \left(\frac{s_{RL}}{2} + \frac{s_{FR}}{2}\right) \cdot v_z\right) \tag{2}$$

$$v_{LR} = \frac{1}{R_{wheel}} \cdot \left(v_x + v_y - \left(\frac{s_{RL}}{2} + \frac{s_{FR}}{2}\right) \cdot v_z\right) \tag{3}$$

$$v_{RR} = \frac{1}{R_{wheel}} \cdot \left(v_x - v_y + \left(\frac{s_{RL}}{2} + \frac{s_{FR}}{2}\right) \cdot v_z\right) \tag{4}$$

where v_{LF}, v_{RF}, v_{LR}, v_{RR} – target rotation velocities of the left front, the right front, the left rear and the right rear wheels respectively in rad/s; R_{wheel} – the wheel radius; v_x is a target linear velocity along X-axis in m/s; v_y is a target linear velocity along Y-axis in m/s; v_z is a target angular velocity along Z-axis in rad/s; s_{RL} is the distance between the right and left wheels; s_{FR} is the distance between the front and rear wheels. Then the target wheel velocity is published to */command* topic of each controller. The plugin allows to select one of two types of published odometry:

1. Ground-truth odometry from Gazebo,
2. Wheel odometry.

In the first case, the position of the robot in the Gazebo is published. In the second case, the plugin searches for the necessary joints in the */joint_states* topic and their current rotation velocity, then calculates the robot velocity using the equations (5) – (7)[19]:

$$v_{xc} = (v_{LFC} + v_{RFC} + v_{LRC} + v_{RRC}) \cdot \frac{R_{wheel}}{4} \quad (5)$$

$$v_{yc} = (-v_{LFC} + v_{RFC} + v_{LRC} - v_{RRC}) \cdot \frac{R_{wheel}}{4} \quad (6)$$

$$v_{zc} = (-v_{LFC} + v_{RFC} - v_{LRC} + v_{RRC}) \cdot \frac{R_{wheel}}{2(s_{RL} + s_{FR})} \quad (7)$$

where v_{xc} is a current linear velocity along X-axis in m/s; v_{yc} is a current linear velocity along Y-axis in m/s; v_{zc} is an current angular velocity along Z-axis in rad/s; v_{LFC}, v_{RFC}, v_{LRC}, v_{RRC} are current rotation velocities of the left front, the right front, the left rear and the right rear wheels respectively; R_{wheel} is the wheel radius; s_{RL} is the distance between the right and left wheels; s_{FR} is the distance between front and rear wheels. Then the current robot position is calculated and the obtained data are published. This allows to choose between the accurate and the realistic odometry. Flowchart of received / sent messages by the plugin is shown in Fig.4

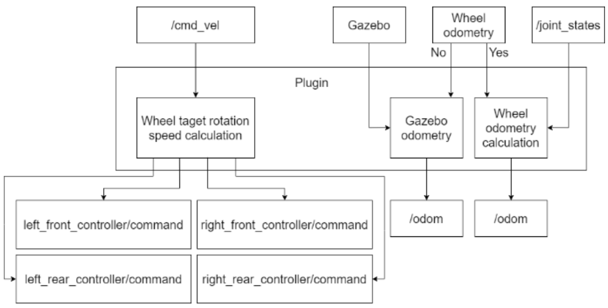


Fig.4.Flowchart of received/sent messages by the plugin

The plugin needs to set a command topic and an odometry topic, an odometry frame, the robot base frame, an update rate, to select the type of odometry, to specify names of the wheel controllers, half the length between the front and rear, right and left wheels and the wheel radius.

4. Conclusion

The article presented the virtual mecanum wheel mobile platform development. A high-poly and detailed model was developed for a visual part of the wheel, and a low-poly model was modeled for the collision part. This allowed to significantly increase the RTF of the Gazebo simulation. The developed plugin controls the model and is suitable for all four wheeled robots with mecanum wheels. The plugin publishes target speeds to wheel controller topics and can publish two types of odometry.

Acknowledgements

This paper has been supported by the Kazan Federal University Strategic Academic Leadership Program ("PRIORITY-2030").

References

1. Rubio F, Valero F, Llopis-Albert C. A review of mobile robots: Concepts, methods, theoretical framework, and applications[J]. Int. J. of Advanced Robotic Systems, 2019.
2. Watanabe K. et al. Autonomous Control for an Omnidirectional Mobile Robot with Feedback Control System[J]. Advances in Intelligent Autonomous Systems, 1999: 289-308.
3. Qian J. et al. The design and development of an omni-directional mobile robot oriented to an intelligent manufacturing system[J]. Sensors, 2017: 2073.
4. Paromtchik I E, Rembold U. A practical approach to motion generation and control for an omnidirectional mobile robot[C]//Int. Conf. on Robotics and Automation, 1994: 2790-2795.
5. Magid E. et al. Automating pandemic mitigation[J]. Advanced Robotics, 2021: 572-589.
6. Senft E, Satake S, Kanda T. Would You Mind Me if I Pass by You? : Socially-Appropriate Behaviour for an Omni-based Social Robot in Narrow Environment[C]//15th ACM/IEEE International Conference on Human-Robot Interaction (HRI). IEEE, 2020: 539-547.
7. Peng H. et al. Stability control for medical rescue robot in unconstructed environment[C]//5th International Conference on Advanced Robotics and Mechatronics (ICARM). IEEE, 2020: 184-188.
8. Iakovlev R, Saveliev A. Approach to implementation of local navigation of mobile robotic systems in agriculture with the aid of radio modules[J]. Telfor Journal, 2020: 92-97.
9. Wu X B. et al. Research on the design of educational robot with four-wheel omni-direction chassis[J]. J. Comput, 2018: 284-294.

10. Watanabe K. Control of an omnidirectional mobile robot[C]//Second Int. Conf. Knowledge-Based Intelligent Electronic Systems, 1998: 51-60.
11. Gfrerrer A. Geometry and kinematics of the Mecanum wheel[J]. Computer Aided Geometric Design, 2008: 784-791.
12. Shabalina K, Sagitov A, Magid E. Comparative analysis of mobile robot wheels design[C]//11th International Conference on Developments in esystems Engineering (dese). IEEE, 2018: 175-179.
13. Shabalina K. et al. Avrora unior car-like robot in gazebo environment[C]//International Conference on Artificial Life and Robotics, 2019: 116-119.
14. Dobrokvashina A. et al. Servosila engineer crawler robot modelling in webots simulator[J]. International Journal of Mechanical Engineering and Robotics Research (IJMERR), 2022: 417-421.
15. Zamfirescu I., Pascal C. Modelling and simulation of an omnidirectional mobile platform with robotic arm in CoppeliaSim[C]//24th International Conference on System Theory, Control and Computing (ICSTCC). IEEE, 2020: 667-672.
16. Apurin A. et al. LIRS-ArtBul: Design, Modelling and Construction of an Omnidirectional Chassis for a Modular Multipurpose Robotic Platform[C]//International Conference on Interactive Collaborative Robotics. Springer, Cham, 2022: 70-80.
17. Taheri H, Qiao B, Ghaeminezhad N. Kinematic model of a four mecanum wheeled mobile robot[J]. Int. J. of computer applications, 2015: 6-9.
18. Dickerson S L, Lapin B D. Control of an omni-directional robotic vehicle with Mecanum wheels[C]//National Telesystems Conf. Proc., 1991: 323-328.
19. Maulana E, Muslim M A, Hendrayawan V. Inverse kinematic implementation of four-wheels mecanum drive mobile robot using stepper motors[C]//Int. Seminar on Intelligent Technology and Its Applications, 2015: 51-56.

Authors Introduction

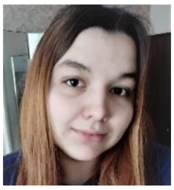
Mr. Artem Apurin

Artem Apurin received his B.Sc. degree at North-Caucasus Federal University in 2021. Now he is a 2-year student of the Master's program in Intelligent Robotics at Kazan Federal University. Currently he works as an engineer at the Laboratory of Intelligent Robotic Systems (LIRS) at Kazan Federal University, Russia

Mr. Bulat Abbyasov

Bulat Abbyasov received his master degree at Institute of Information Technology and Intelligent Systems (ITIS), Kazan Federal University in 2022. Currently, he is a first year PhD student in Robotics at ITIS, Kazan Federal University.

Ms. Alexandra Dobrokvashina



She received her master degree at Institute of Information Technology and Intelligent Systems, Kazan Federal University in 2021. Currently she works as a research assistant at the Laboratory of Intelligent Robotic Systems (LIRS) at Kazan Federal University, Russia.

Assistant Professor Yang Bai

Currently he works as an Assistant Professor at the College of Information Science and Engineering of the Ritsumeikan University. His interests are robotics, non-holonomic systems and motion planning.

Professor Mikhail Svinin

He works as a Professor at the College of Information Science and Engineering of the Ritsumeikan University. His current research interests include robotics, control theory, and analytical mechanics. He is a member of American Society of Mechanical Engineers (ASME), Robotics Society of Japan (RSJ), and the Society of Instrument and Control Engineers (SICE).

Professor Evgeni Magid

Professor, a Head of Intelligent Robotics Department and a Head of Laboratory of Intelligent Robotic Systems (LIRS) at Kazan Federal University, Russia. Professor at HSE University, Russia. Senior IEEE member. Previously he worked at University of Bristol; Carnegie Mellon University; University of Tsukuba, Japan; National Institute of Advanced Industrial Science and Technology, Japan. He earned his Ph.D. degree from University of Tsukuba. He authors over 200 publications