JOURNAL OF ALGEBRAIC STATISTICS
Volume 13, No. 1, 2022, p. 900 - 909
https://publishoa.com
ISSN: 1309-3452

Quadruped Robot Navigation Using ROS

R. Merlin[1], S. Ameena[2]

[1]Assistant Professor, Dept of ECE, Ashoka Women's Engineering College, Affiliated to JNTUA, Kurnool, A.P, India.
[2] Assistant Professor, Dept of ECE, Ashoka Women's Engineering College, Affiliated to JNTUA, Kurnool, A.P, India.

ABSTRACT

The development of quadruped robot platforms is a relatively active area of scientific resource. Taking into account the intuitiveness and high efficiency of robot development process, the robot development method based on ROS (robot operating system) is adopted. First, the kinematics modeling of the quadruped robot is carried out, and the robot kinematics is solved. Secondly, the basic principles of gait planning for quadruped. Finally, the quadruped robot model file is imported into ROS and the gait planning control program of the quadruped robot is written uses through the user's API for motion planning simulation. At the same time, RViz can visualize the robot movement process, and obtain other relevant data during the movement process, indicating the correctness of the trajectory planning. The results shot that the development method of quadruped robot based on ROS is feasible. Autonomous quadruped robots require localization and mapping for navigation. Different from mobile robots, quadruped robots need local dense maps with more detailed information for motion planning. We use lidar sensor in this quadruped robot.

Keywords: Lidar sensor,ROS,GAZEBO

1. Introduction

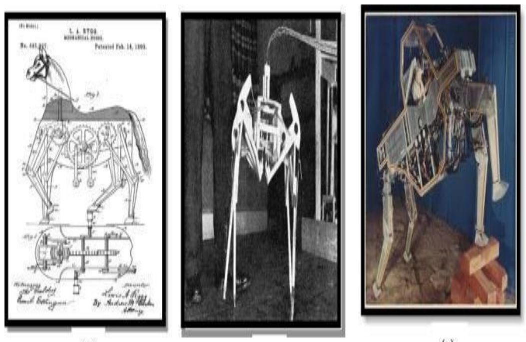
Robotics is a branch of engineering and science that includes electronics engineering, mechanical engineering and computer science and so on. This branch deals with the design, construction, use to control robots, sensory feedback and information processing.It was invented by Issac and he stated that:Robots will never harm human beings.Robots will follow instructions given by humans with breaking law one.Robots will protect themselves without breaking other rules. Types of Robots: Articulated: The feature of this robot is its rotary joints and range of these are from 2 to 10 or more joints. The arm is connected to the rotary joint and each joint is known as the axis which provides a range of movements.Cartesian: These are also known as gantry robots. These have three joints which use the Cartesian coordinate system i.e x, y, z. These robots are provided with attached wrists to provide rotatory motion.Cylindrical: These types of robots have at least one rotatory joints and one prismatic joint which are used to connect the links. The use of rotatory joints is to rotate along the axis and prismatic joint used to provide linear motion.Polar: These are also known as spherical robots. The arm is connected to base with a twisting joint and have a combination of 2 rotatory joint and one linear joint.

Scara: These robots are mainly used in assembly applications. Its arm is in cylindrical in design. It has two parallel joints which are used to provide compliance in one selected plane.Delta: The structure of these robots are like spider-shaped. They are built by joint parallelograms that are connected to the common base. The parallelogram moves in a dome-shaped work area. These are mainly used in food and electrical industries. Scope of robots:The advance version of machines are robots which are used to do advanced tasks and are programmed to make decisions on their own. In general basic robots, their complexity is decided by the number of limbs, actuators and the sensors that are used while for advanced robots the complexity is decided by the number of microprocessors and microcontroller used. As increasing any component in the robot, it is increasing the scope of the robot and with every joint added, the degree of the robot is enhanced.

JOURNAL OF ALGEBRAIC STATISTICS
Volume 13, No. 1, 2022, p. 900 - 909
https://publishoa.com
ISSN: 1309-3452

2. Existing Methods

In the early 1900s, many scientists and researchers were devoting their time to research the leg mechanism of the four-footed robot. Chebyshev .The Chebyshev's mechanism looks like the Greek letter Lambda; therefore, the linkage is also named a lambda mechanism. This device could walk dynamically on flat terrain only and does not have an independent leg motion. Later, this mechanism got incorporated into two machines MELWALK and DANTE Chebyshev Mechanism .



(a) (b) (c)

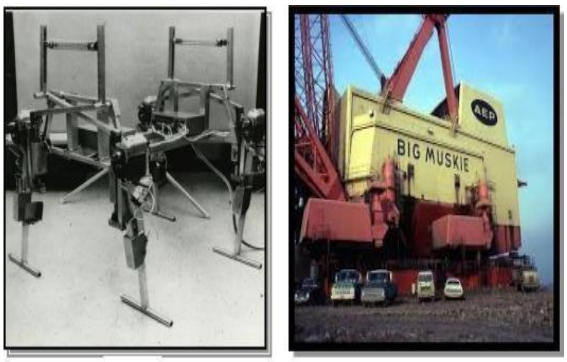
Fig 2.1: Early 1900's Quadruped Robot

(a)Mechanical Horse (b) Hutchinson Four legged

Mid 1900s quadruped Robot:

The first autonomous quadruped robot in the United States was constructed in the 1960s at the University of Southern California. It was named "phony pony"

a) Phoney Pony (b) Big Muskie.



(a) (b)

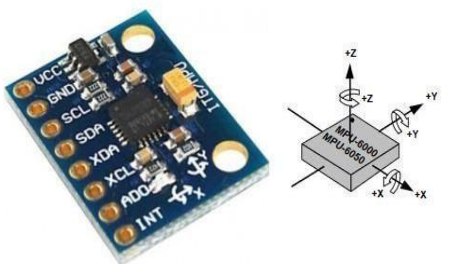


Fig 2.2: Mid 1900's Quadruped Robot

Early 2000s quadruped Robot:

JOURNAL OF ALGEBRAIC STATISTICS
Volume 13, No. 1, 2022, p. 900 - 909
https://publishoa.com
ISSN: 1309-3452

In the early 20th century, Patrush, and then followed by the biomorphic quadruped TEKKEN series have developed by Kimura et al. In Patrush, each leg has three joints, and the approximate weight of the robot is 5.2 kg. It can walk with speed 0.6 m/s on the flat surface and controlled and written by RT-Linux, C-language respectively

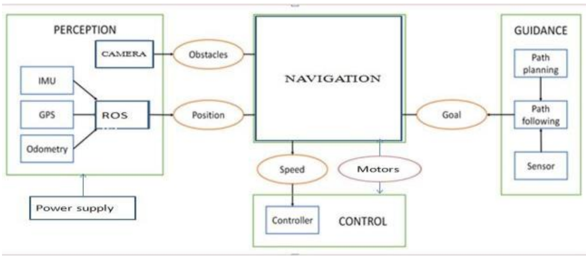


Fig 3.1.1: Block diagram for Quadruped Robot Navigation Using ROS

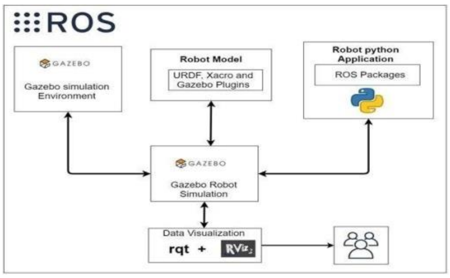


Fig 3.1.2: Block diagram for Robotic Operating System

IMU:

Navigating a robot can be pretty easy while using human assistance. In certain cases, full control will be needed. As part of a helping hand project, we need to use IMU (inertial measurement unit) sensor. This type of sensor can measure and report robot's specific force, angular rate, and the magnetic field surrounding the robot in all three directions (X, Y and Z) and for sure for the helping hand.

From IMU (specifically MPU6050 IMU) through Arduino board and send it directly to ROS using rosserial. As reading the IMU raw sensors' data will be a cornerstone part for any project that uses IMU with ROS.

MPU6050 features from the datasheet:

I2C Digital-output

Input Voltage: 2.3 – 3.4V.

Tri-Axis angular rate sensor (gyro) with a sensitivity up to 131 LSBs/dps and a full-scale range of ±250, ±500, ±1000, and ±2000dps.Tri-Axis accelerometer with a programmable full-scale range of ±2g, ±4g, ±8g and ±16g.Digital Motion Processing™ (DMP™) engine offloads complex Motion Fusion, sensor timing synchronization and gesture detection.

Quadruped robot needs

GPS:

The GPS system is used for the robot to position and navigate in a long distance, and the robot compare the robot's GPS data and the target's GPS data to get the relative position between both, which guides the direction of motion for the robot later.

JOURNAL OF ALGEBRAIC STATISTICS
Volume 13, No. 1, 2022, p. 900 - 909
https://publishoa.com
ISSN: 1309-3452

ODOMETRY SCALAR :

You can use this parameter as a multiplier to the calculated velocities for dead reckoning. This can be useful to compensate odometry errors on open-loop systems. Normally this value ranges from 1.0 to 1.20.

CONTROLLERS:

A list of available controller plugins, contained in ros_controllers, as of this writing. You can of course create your own and are not limited to the below list.

1.joint_state_controller-Publishes the state of all resources

2.hardware_interface::Joint State Interface to a topic of type sensor_msgs/JointState.

3.velocity_controllers –Command a desired velocity to the Hardware Interface.

4.effort_controllers-Command desired effort(force/torque) to the Hardware Interface.

5.joint_trajectory_controllers –Extra functionality for splining an entire trajectory.

SERVO MOTORS USED IN ROBOTICS:

They are small, powerful, easily programmable, and accurate. Most importantly, though, they allow for near perfect repeatability of motion. They are used in robotic applications such as: robotic welding , robotic vehicles

NAVIGATION IN ROBOTICS :

Robot navigation means the robot's ability to determine its own position in its frame of reference and then to plan a path towards some goal location.Navigation can be defined as the combination of the three fundamental competences:Self-localisation. Path planning. Map-building and map interpretation.

SELF LOCALIZATION:

. In robotics, self-localization on the basis of monomodal perceptual information has been investigated intensively.

PATH PLANNING:

It is a computational problem to find a sequence of valid configurations that moves the object from the source to destination.

MAP BUILDING AND MAP INTERPRETATION:

An autonomous robot is to be able to construct (or use) a map (outdoor use) or floor plan (indoor use) and to localize itself and its recharging bases or beacons in it. Robotic mapping is that branch which deals with the study and application of ability to localize itself in a map / plan and sometimes to construct the map or floor plan by the autonomous robot.

ROBOT OPERATING SYSTEM(ROS):

GAZEBO SIMULATION ENVIRONMENT:

Gazebo is an open-source 3D robotics simulator. It integrated the ODE physics engine, OpenGL rendering, and support code for sensor simulation and actuator control.

Gazebo can use multiple high-performance physics engines, such as ODE, Bullet etc. (the default is ODE). It provides realistic rendering of environments including high-quality lighting, shadows, and textures. It can model sensors that "see" the simulated environment, such as laser range finders. cameras (including wide- angle), Kinect style sensors, etc.

JOURNAL OF ALGEBRAIC STATISTICS
Volume 13, No. 1, 2022, p. 900 - 909
https://publishoa.com
ISSN: 1309-3452

GAZEBO ROBOT SIMULATION :

Gazebo is an open-source 3D robotics simulator It integrated the ODE physics engine, OpenGL rendering, and support code for sensor simulation and actuator control.Gazebo can use multiple high-performance physics engines, such as ODE Bullet, etc. (the default is ODE).

It provides realistic rendering of environments including high-quality lighting, shadows, and textures. It can model sensors that "see" the simulated environment, such as laser range finders, cameras (including wide-angle), Kinect style sensors, etc.

DATA VISUALIZATION:

rqt:rqt is a Qt-based framework for GUI

Development for ROS. It consists of three parts/meta packages

rqt (you're here)

rqt_common_plugins - ROS backend tools suite that can be used on/off of robot runtime.

rqt_robot_plugins - Tools for interacting with robots during their runtime.

rqt metapackage provides a widget

rqt_gui that enables multiple `rqt` widgets

to be docked in a single window.

RVIZ:

3D visualization tool for ROS. RVIZ is a ROS graphical interface that allows you to visualize a lot of information, using plugins for many kinds of available topics.

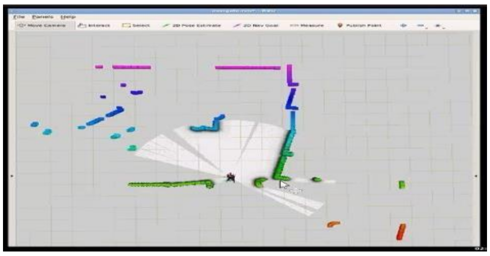


Fig : 3.RVIZ

3. Robot Operating System

The Robot Operating System (ROS) is a flexible framework for writing robot software. It is a collection of tools, libraries, and conventions that aim to simplify the task of creating complex and robust robot behavior across a wide variety of robotic platforms. As a result, ROS was built from the ground up to encourage collaborative robotics software development.

Robot Operating System (ROS or ros) is an open source robotics middleware suite. Although ROS is not an operating system but a collection of software frameworks for robot software development, it provides services designed for a heterogeneous computer cluster such as hardware abstraction, low-level device control, implementation of commonly used functionality, message-passing between processes, and package management. Running sets of ROS-based processes are represented in a graph architecture where processing takes place in nodes that may receive, post and multiplex sensor data, control, state, planning, actuator, and other messages. It is possible, however, to integrate ROS with real-time code. The lack of support for real-time systems has been addressed in the creation of ROS 2.0, a major revision of the ROS API which will take advantage of modern libraries and technologies for core ROS functionality and add support for real- time code and embedded hardware.

JOURNAL OF ALGEBRAIC STATISTICS
Volume 13, No. 1, 2022, p. 900 - 909
https://publishoa.com
ISSN: 1309-3452

Fig : 6.1 Robot Operating System

WORKING MODEL OF QUADRUPED ROBOT NAVIGATION

An open source framework for building new quadrupedal robots, CHAMP is based on a hierarchical controller design for highly dynamic locomotion first implemented on the MIT Cheetah robot. The framework, also developed for implementing new control algorithms, includes a setup assistant to configure newly-built robots and features full autonomy using the ROS navigation stack.

Running your own robot:

There are two ways to run CHAMP on a real robot:

Linux Machine

Use this ROS package to calculate the joint angles and send it to a hardware interface to control your actuators. You can follow these guidelines to create your actuators' interface.

Lightweight Version

Run CHAMP's lightweight version on Teensy series microcontrollers and use it to directly control our actuators.

Generate robot configuration:

First generate a configuration package using champ_setup_assistant Follow the instructions in the README to configure your own robot. The generated package contains:

URDF path to your robot.

Joints and Links map to help the controller know the semantics of the robot.

Gait parameters.

Hardware Drivers.

Navigation parameters (move_base, amcl and gmapping).

Microcontroller header files for gait and lightweight robot description. This only applies to robot builds that use microcontroller to run the quadruped controller.

Next, build your workspace

This will run the quadruped controller and all sensor/hardware drivers:

roslaunch <myrobot_config> bringup.launch

Available Parameters:

rviz - Launch together with RVIZ. Default: false

lite - Always set this to true if you're using a microcontroller to run the algorithms. Default false

JOURNAL OF ALGEBRAIC STATISTICS
Volume 13, No. 1, 2022, p. 900 - 909
https://publishoa.com
ISSN: 1309-3452

Creating a map:

The base driver must be running to run gmapping and move_base.

Run gmapping package and mov

roslaunch <myrobot_config> slam.launch

To open RVIZ and view the map:

roscd champ_navigation/rviz

rviz -d navigate.rviz

To start mapping:

Click '2D Nav Goal'

Click and drag at the position you want the robot to go

Save the map by running:

roscd <myrobot_config>/maps

rosrun map_server map_saver

Running your robot in Gazebo:

Run Gazebo and the base driver in simulation mode

roslaunch <myrobot_config> gazebo.launch

Tuning gait parameters:

The gait configuration for your robot can be found in <my_robot_config>/gait/gait.yaml.

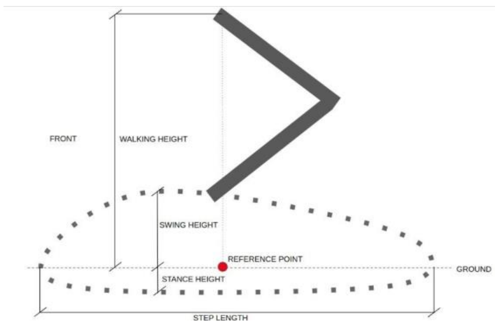


Fig 7.Tunning Gait Parameters

Knee Orientation - How the knees should be bent. You can can configure the robot to follow the following orientation .>> .>< .<< .<> where dot is the front side of the robot.

Max Linear Velocity X (meters/second) - Robot's maximum forward/reverse speed.

Max Linear Velocity Y (meteres/second) - Robot's maximum speed when moving sideways.

Max Angular Velocity Z (radians/second)- Robot's maximum rotational speed.

Stance Duration (seconds)- How long should each leg spend on the ground while walking. You can set this to default(0.25) if you're not sure. The higher the stance duration the further the displacement is from the reference point.

Leg Swing Height (meters)- Trajectory height during swing phase.

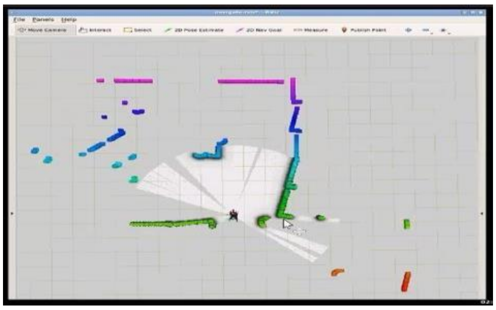
JOURNAL OF ALGEBRAIC STATISTICS
Volume 13, No. 1, 2022, p. 900 - 909
https://publishoa.com
ISSN: 1309-3452

Leg Stance Height (meters)- Trajectory depth

during stance phase.

Robot Walking Height (meters) - Distance from hip to the ground while walking. Take note that setting this parameter too high can get your robot unstable.

CoM X Translation (meters) - You can use this parameter to move the reference point in the X axis. This is useful when you want to compensate for the weight if the center of mass is not in the middle of the robot (from front hip to rear hip). For instance, if you find that the robot is heavier at the back, you'll set a negative value to shift the reference point to the back.



Odometry Scaler - You can use this parameter as a multiplier to the calculated velocities for dead reckoning. This can be useful to compensate odometry errors on open-loop systems. Normally this value ranges from 1.0 to 1.20

4. Experimental Results

The aim of this project is to design and develop Quadruped robot navigation using ros.

Run the gazebo environment:

source devel/setup.bash

roslaunch champ_config gazebo.launch

Run gmapping package and move_base:

cd catkin_ws

source devel/setup.bash

roslaunch champ_config slam.launch rviz:=true

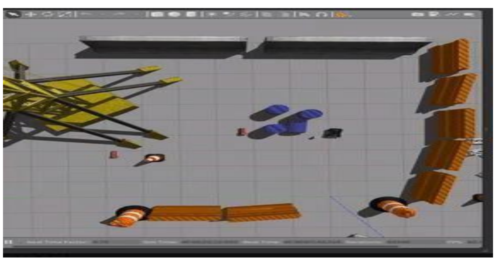


Fig 8: Gazebo Simulator

JOURNAL OF ALGEBRAIC STATISTICS
Volume 13, No. 1, 2022, p. 900 - 909
https://publishoa.com
ISSN: 1309-3452

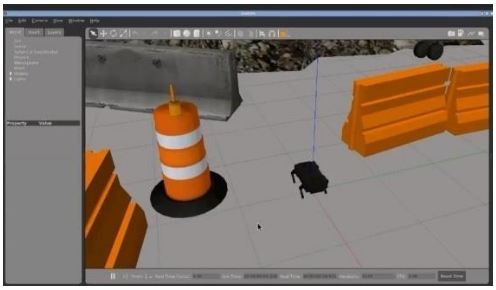


Fig 9:RVIZ

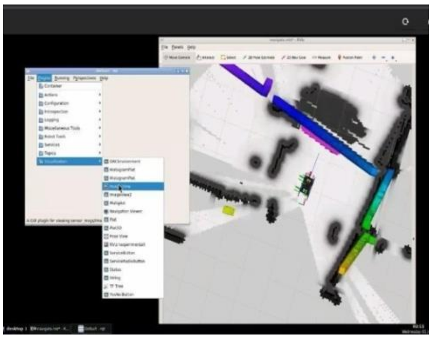
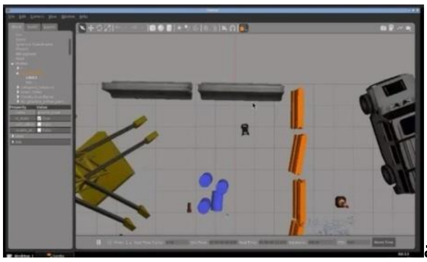


Fig .10:2D NAV GOAL



a

Fig 8.5 RVIZ+rqt

5. Conclusion

Quadrupedal robots (Tetrapod robots) are statically stable, especially when they are not in motion, they have 4 legs and their walking pattern similar to that of animals, they are well balanced in posture, At slow speeds, They can move either by moving one leg at a time, ensuring a stable tripod or moving the alternate pair of legs to walk.There are clear stability advantages to quadrupeds over other approaches (bipeds, wheels andtreads/track plates) across multiple of terrains and elevations. At Ted last year, Raibert demonstrated how his robo-pups, instead of drones and rovers, could be used for package delivery by easily ascending and descending stairs or other vertical obstacles. AnyMAL is able to perform dynamic maneuvers with its four legs to find footholes blindly without the need for vision sensors. While wheeled systems literally get stuck in the mud, Hunter's mechanical beast can work continuously: above ground, underneath the surface, falling, spinning and bouncing upright to perform a mission with precise accuracy. Quadruped robots, it has been concluded that machines, that are likeanimals have servedmankind only due to the advanced technology and the quality of design. The quadruped robots are made to perform in practical applications like mine inspection, space exploration, fire-fighting or where navigation is required in an unconstrained environment.An autonomous navigation scheme for quadruped robot, including constructing global maps in advance, real-time position,

JOURNAL OF ALGEBRAIC STATISTICS
Volume 13, No. 1, 2022, p. 900 - 909
https://publishoa.com
ISSN: 1309-3452

building a dense elevation map and finally performing path planning in the global map and local map. After several tests in complex outdoor environments, the robot was able to reach the artificially set target successfully.

References

[1]. Meng X, Wang S, Cao Z, Zhang L. A review of quadruped robots and environment perception. In: Control Conference (CCC), 2016 35th Chinese. IEEE; 2016. p. 6350– 6356.Y. Zhong, R. Wang, H. Feng, Y. Chen

[2]. A review of heavy-duty legged robots Sci China Technol Sci, 57 (2) (2014), pp. 298-314CrossRefD.J. Todd

[3]. Walking machines: an introduction to legged robots Springer Science and BusinessMedia (2013) J.A. Tenreiro Machado, M. Silva

[4]. Legged robots Commun ACM, 29 (6) (1986), pp. 499-514View Record in Scopu P.G. DeSantos, E. Garcia, J. Estremera

[5]. Study on quadruped walking robot in Tokyo institute of technology Proceedings of the 2000 IEEE international conference on robotics and automation (2000),pp. 414-419

[6]. Hirose S, Yoneda K, Tsukagoshi H. TITAN VII: Quadruped walking and manipulating robot on a steep slope. In: 1997 IEEE international conference on robotics and automation, 1997. Proceedings, vol. 1. IEEE; 1997. p. 494–500.