ISSN (Online) 2394-6849

International Journal of Engineering Research in Electronics and Communication Engineering (IJERECE)

Volume 11 Issue 7 July 2024

Robot Dance Choreography

[1] Gaddaaley Ghannesh, [2] Tsuyoshi Nakajima, [3] Dr. Nakandhrakumar R S, [4] Ms Seenu N

[1] [2] [3] Hindustan Institute of Technology & Science
Corresponding Author Email: [1] saiganesh220602@gmail.com

Abstract*— Our project will join tech together with dance robots through the robot choreography interface by using the TurtleBot3 mobile robot platform along with the Robot Operating Systems (ROS-Noetic) and then improve the robotics choreography and programming capabilities. Additionally, the idea is to choreograph a dynamic dance sequence that will require different skills to solve the technical issues concerning TurtleBot3 and ROS in this segment. The beginnings of this project will be based on that; TurtleBot3 research and will subsequently range into actuators as well as programming interface. Specifically, ROS Noetic projects refer to the software development where the software written, acts as if robot is thinking and planning according to its physical constraints and capabilities. The algorithm is now shipped with TurtleBot3 and then we can run the dance repetition by the algorithm and the loved and hated score are totaled up by the technical score and art score where technical score includes stuff like precision, straightness and creativity while art score also contains the elements like choreography, emotional depth and the quality of expression. Similarly to other ordinary robot projects where the efficiency becomes the most important aspect, to these devices the creative options are also added. Hence, it can be established that such fusion has an influence on this discipline that can be investigated for development in the future.

Keywords*— TurtleBot3, Robot choreography, Dance programming, Virtual space, Motion capture technology, Synchronization, Real-time feedback.

I. INTRODUCTION

The notion of the creative and expressive side of progress in the robotic realm is turning into the expressive ways of art and development. This study applies a unique strategy that concentrates on advanced choreography techniques for the TurtleBot3 robotic platform – an adapted, versatile, and creative robot. Assembling a robot motion implies the set of special challenges that is direct synchronization of the similar motion sequences and of the aesthetic geometric pattern. To resolve these design issues, we use our software together with TurtleBot3 and a ROS simulation environment. Another contribution offered by robot creative potential which involves integrating advanced softwares and simulation systems is the starting of an iterative development that needs no physical robot movement.

Movements may be either being conducted by the artist or arranged by him or maybe the robotics, performance, communication, entertainment may involve the usage of the movements. If you are looking for a cheap and flexible hardware platform to build and research about robotic development, popular for its low cost and adaptability, the open source TurtleBot3 might be just the right option for you. The project is purposed to make the robot TurtleBot3 to wiggle new styles and consequently to complete the complex pattern in the ROS initiated virtual environment.

Choreography program is the basis of our program that can be easily used by the interface to control and carry out the dance. This piece of software is based on ROS, which is a powerful and open-source platform for solving task robot software development. ROS helps us in creating an interface between the choreographic software and the digital environment that makes the flow of data be real time and therefore we make the robot to exhibit the right behavior and to respond well.

Ubuntu 20.04, with its nickname of "Focal Fossa", underlines the project's commitment to stable, secure, and user-friendly design. This long-term support version (LTS), that was introduced in April 2020, has become a favorite of open-source admirers quickly as a consequence of its reliability and very feature-interesting UI.

Firstly, the main purpose of Ubuntu 20.04 is to provide users with the latest kernel of 5.4, which comes with advanced compatibility, speed and security features. GNOME is the default desktop environment with the clutter-free, clear, and simple design that makes the system attractive to both novice and advanced users.

Snap package integration is an aspect, which is greatly accentuated in Ubuntu 20.04. Snap packages are easy and secure in terms of software distribution across Linux distributions and give end users access to an immense software library. It simplifies the installation and update procedures giving rise to trouble-free service for the customer.

ROS (Robotic Operating System) Noetic is the leap forward in the expansion of the open-source robotics framework, empowering developers and researchers to assemble the complicated systems of robots. ROS Noetic Ninjemys, a Long-Term Support (LTS) version released in May 2020, comes with a bunch of improvements and features to address different needs of the robotics community.

ROS Noetic which is based on the Python 3 programming language, follows advanced programming techniques so as to be compatible with the latest libraries and software tools. The migration from Python 2 to Python 3 proves the commitment to stay with the latest market standards and security practices.

ISSN (Online) 2394-6849

International Journal of Engineering Research in Electronics and Communication Engineering (IJERECE)

Volume 11 Issue 7 July 2024

II. SYSTEM SETUP AND ENVIRONMENT

Ubuntu 20.04 Installation:

- Install Ubuntu 20.04 LTS (Long-Term Support) on a dedicated computer or your development workstation.
- Make sure your machine has the minimal hardware needed to run Gazebo and ROS.

ROS Noetic Installation:

- Ubuntu 20.04 is compatible with the most recent version of ROS, which is ROS Noetic Ninjemys.
- Refer to the ROS Noetic Installation Guide for authoritative information on how to install Noetic.

TurtleBot3 Setup:

- One well-known open-source robot platform is TurtleBot3. There are several models to select from, such as Burger, Waffle, and Waffle Pi.
- Assemble the robot's parts to configure the TurtleBot3 hardware (wheels, sensors, etc.).
- Use Wi-Fi or USB to connect the TurtleBot3 to your computer.

Simulation Environment (Gazebo):

- You may test and see how your robot behaves in a simulated world with Gazebo, a potent robot simulation tool.

Install Gazebo using ROS packages:

- sudo apt-get install ros-noetic-gazebo-ros-pkgs ros-noetic-gazebo-ros-control

Launch the TurtleBot3 simulation in Gazebo:

- roslaunch turtlebot3_gazebo turtlebot3_empty_world.launch

TurtleBot3 ROS Packages:

Install the necessary TurtleBot3 ROS packages:

sudo apt-get install ros-noetic-turtlebot3*

- Configure the TurtleBot3 model (Burger, Waffle, or Waffle Pi) in your ROS environment.

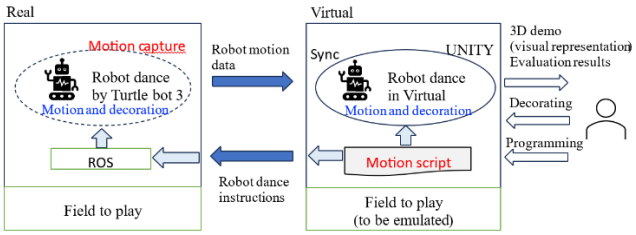
Testing and Verification:

- Check that the TurtleBot3 simulation in Gazebo is operating without a hitch.
- Use teleoperation instructions to test basic movements (forward, backward, and turning).

Physical Robot Setup:

- Connect the LiDAR, camera, and IMU sensors if you're using a physical TurtleBot3, and make sure your computer is communicating with it properly.
- Adjust sensor calibration if required.
- Development Environment:To write and debug ROS code, set up your favorite Integrated Development Environment (IDE) (e.g., Visual Studio Code, PyCharm, etc.).

III. METHODOLOGY



Choreography Program Development:

- Make choreography software or motion script to produce dance motions for the TurtleBot3 robot.
- Make your system ROS compatible for quick inclusion into virtual and physical environments.

Virtual Space Simulation:

- Implement a virtual environment within ROS so that the choreography software may imitate the robot's motions.
- Allow the software to create visual representations of the robot's dancing routines from various perspectives in the virtual space.

Artistic Enhancement and Quantitative Evaluation:

- Integrate elements into the virtual simulation environment that allow for aesthetic enrichment and decorating of the virtual robot.
- Implement methods for quantitative evaluation of artistic aspects, allowing researchers to examine and improve choreography.

Real Space Interface with ROS:

- Create a communication link between the choreographic software and the TurtleBot3 robot in real time using ROS.
- Create protocols for communicating dancing instructions from the choreography program to the robot, assuring immediate reaction.

Motion Capture Integration:

- Use motion capture technology to monitor the robot's motions in actual space, then synchronize the data with the virtual simulation to automatically correct for variations between the two models.

IV. ROBOT KINEMATICS AND MOTION CONTROL

Kinematic Model:

- The relationship between robot's joint locations (wheel's rotations) and the motion that is occurred (linear and angular velocities) is represented by the kinematical model with the use of mathematics. Here, instead of a elaborate draft of a 2 DOF robot with a differential drive propulsion system, we will focus on a simplified model.
- Conceptualize a case of a robot that has two wheels and one central point which are apart from each other at length L.Let:

ISSN (Online) 2394-6849

International Journal of Engineering Research in Electronics and Communication Engineering (IJERECE)

Volume 11 Issue 7 July 2024

- W_r and W_l denote the rotational velocities of the right and left wheels in positive direction for counter-clockwise convention.
- Manipulator speed (assume forward has positive sign) be the robot linear's relative velocity.
- ω be the angular velocity of robot (counter-clockwise is their a positive value or is it prevelant?
- The linear velocity (v) of the robot is simply the average of the linear velocities of each wheel:The linear velocity (v) of the robot is simply the average of the linear velocities of each wheel:
- v = (ω_r + ω_l)/(m_r + m_l) L / 2.
- The angular velocity (ω) of the robot relates to the difference in wheel velocities:The angular velocity (ω) of the robot relates to the difference in wheel velocities:
- ω = (ω_r -ω_l)/L.

V. LITERATURE REVIEW

This paper drives at dance production for robots, while courses of motion synthesis and planning are done using chance movement primitives among others. It shows that the ability of generating several boring and exciting robot dance moves can be accomplished taking advantage of motion primitives from that are learnt through machine learning. The framework is not limited to this only, it also provides a ground for changing human dance moves to the TurtleBot3 platform, that can then be optimized specifically for the robot and its hardware.

In interactive manner study report focuses on robotized dance performances powered by online reinforcement learning. It brings attention to the more prompt and direct audience interaction thus allowing the development of more dynamic or reactive TurtelBot3's routines accordingly to the kind of audience. This approach does not limit the shows to typical controlled reaction of the spectators, leaving space for making the overall perception the most compelling and captivating.

Whereas not the concern of this paper, these valuable insights however, do offer information on dynamic footprints for mobile navigation by robots in human crowds. Through the discussed ideas, the safe robot navigation in human environments can be achieved. Then, in the purpose of smooth and interruption-free movement of Turtlebot3 during dancing in common spaces, the mentioned in this paper principles can be applied in order to decrease the chances of collisions or other disruptions.

The purpose of this paper is the concentrated signal processing for robot movement generating by the music analysis. This paper will provide not only how to create expressive and synchronized movements, but also the method on how make the robots danced with music. Through treatment of musical cues, TurtleBot3 artificially increases harmonization of movement with the rhytm, atmosphere and other features of the chosen track improving among the other things and the emotional impact and artistic quality of its performance.

The presentation of the TurtleBot3 platform in its entirety (the article's goal is to enrich readers with all they need to know about the TurtleBot3 platform and ultimately what it is made of and the extent it can be maximized). Choreographers can learn the hardware and software restrictions to reach answers and create dance movements that could be operating under TurtleBot3. Such movement can be performed perfectly without interruptions and errors during the dance performance, that is, with high performance and functionality.

Though not limiting to the kinematic limitations of TurtleBot3, an open source choreography authoring tool was presented in the paper particularly those who don't have any programming experience. Even though sometimes its orientation may not be in perfect harmony with the functionality of TurtleBot3, the app provides an inspiration to create user-friendly interfaces and tools that make it possible to test and run different dance routines for robots. It is the line of the approach, which is oriented on democratization of the movement design process allowing everybody who is interested in this sort of art to join the process of creation.

VI. DANCE ROUTINE EXECUTION

Testing in Gazebo Simulation

Overview

- With the help of Gazebo, a potent simulation tool, we can see and verify our robot's actions in a safe virtual setting.
- We carefully tested our dancing routine in Gazebo before implementing it on the actual robot.

Steps Taken

Launch Simulation:

With Gazebo, we used the following command to start the TurtleBot3 simulation:

- turtlebot3_gazebo turtlebot3_empty_world.launch (roslaunch)

Observations and Debugging:

- We watched the robot to make sure it moved in unison with the prearranged dance routine.
- Any anomalies or strange conduct were thoroughly investigated.

Transitioning to the Real TurtleBot3

Overview

- Making the switch from simulation to the real robot is a crucial stage.
- Our goal was to get the dancing routine performed flawlessly on the actual TurtleBot3.

Steps Taken

Hardware Setup:

- We linked the TurtleBot3 to the sensors.
- Made sure our development machine and the robot

ISSN (Online) 2394-6849

International Journal of Engineering Research in Electronics and Communication Engineering (IJERECE)

Volume 11 Issue 7 July 2024

were properly communicating.

Calibration and Verification:

- If necessary, calibrated sensors to increase accuracy.
- Confirmed that the robot complied with velocity orders.

VII. RESULT

Installation of Ubuntu 20.04 and ROS Noetic:

- Initially installed Ubuntu 20.04 on Raspberry Pi and set up ROS Noetic. However, encountered errors during the process due to the absence of TurtleBot3 packages.

Error in Teleoperation Setup:

- Attempted to teleoperate the TurtleBot3 in RViz but faced issues, later identified the missing TurtleBot3 packages as the cause of errors.

Resolution of Package and Model Errors:

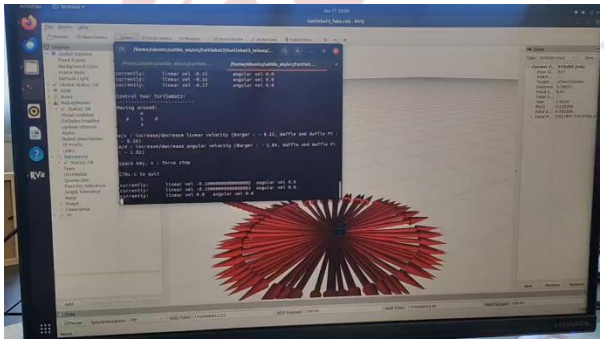
- Researched and identified model and map errors in RViz. Discovered that the missing packages issue could be resolved by installing the necessary TurtleBot3 packages. Successfully resolved the errors by downloading and installing these packages.

Understanding the Importance of Sourcing:

- Encountered persistent errors even after package installation, realized the significance of sourcing the required files. Learned that without sourcing the file, running the program was not possible. Solved the issues by incorporating this step.

Current Status:

- After addressing the errors and sourcing the necessary files, the system is now successfully configured for running TurtleBot3 in RViz. The teleoperation setup in the virtual environment is now functional, marking progress in the project.



- The actual TurtleBot3 was controlled by teleoperation.
- Created a virtual environment for TurtleBot3 in Gazebo
- Created a Python script to make the TurtleBot3 dance in the virtual environment.
- Real TurtleBot3 Dancing:
- The TurtleBot3 has been successfully programmed to execute a dance routine.
- The robot danced in both the virtual Gazebo and the actual environment.

ROS Integration:

- Used the Robot Operating System (ROS) for control and communication.
- Demonstrated a smooth transition from the virtual robot to the real one.

Ubuntu 20.04 and ROS Noetic:

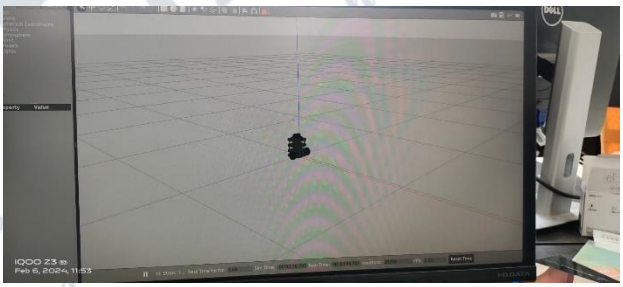
- Utilizing ROS Noetic, the code was developed and tested on Ubuntu 20.04.
- Made certain that the newest software versions are compatible.

Choreography Details:

- Defined particular motions, including spin, diagonal, forward, and backward.
- A certain amount of time and repetitions for every exercise.

Report Considerations:

- Add the objectives, technical information, and project summary.
- Emphasize the hardware's and the simulation's successful execution.



VIII. CONCLUSION

On the whole, employing the TurtleBot3 choreography framework, which caters for the virtual simulation using ROS and motion capture with wonderful combination, forms a significant step in the development of robotic dance. Reiteration process reduces the need of person involved and says implicitly that the more efficient and creative will be the invention. This new approach to robotic choreography not only is a pioneer in the robotics field but also spotlights the possibility of 3D motion capture, simulations, robotics interdisciplinary cooperation, that lays the foundation for a platform of interlinked technologies that can open new possibilities in expressive robotics.

REFERENCES

[1] Levine S, Maturana F, Goyal R, Kersting K, Kaelbling LP. Choreographic Exploration for Humanoid Robots: Motion Synthesis and Planning Using Probabilistic Movement Primitives. Proceedings of the Robotics: Science and Systems Conference. 2015.

[2] Yang F, Wang X, Zhou X, Zhao J, Xu R, Huang H. Interactive

Robot Dance Performance Using Online Reinforcement Learning. IEEE Transactions on Cybernetics. 2018

[3] Bouraine S, Sisbot EA, Alami H, Chatila R. Dynamic Footprints for Mobile Robot Navigation in Human Crowds. Autonomous Robots. 2013

[4] Park JH, Kim JH, Park KS, Bang SW. Musical Signal Processing for Robotic Movement Generation. Proceedings of the 2011 IEEE International Conference on Robotics and Automation. 2011.

[5] Park J, Hong T, Kim Y, Hwang JH, Kim S, You I. TurtleBot3: A Platform for Mobile Robotics Research and Education. International Journal of Control, Automation and Systems. 2016

[6] LaValle SM, Mallek JM, Kavanagh KT, DeRose LP. Open-Source Humanoid Choreography Authoring Tool for Non-Programmers. ACM Transactions on Graphics. 2013