Integrated Development Environment for Rover

S.Murali [#1], N. Jaisankar [*2]
School of Computing Sciences and Engineering, VIT University
Vellore-632014. Tamilnadu. India
[1] murali.s@vit.ac.in
[2] njaisankar@vit.ac.in

***Abstract*—Today, robotics is an auspicious and fast-growing branch of technology that involves the manufacturing, design, and maintenance of robot machines that can operate in an autonomous fashion and can be used in a wide variety of applications including space exploration, weaponry, household, and transportation. More particularly, in space applications, a common type of robots has been of widespread use in the recent years. It is called planetary rover which is a robot vehicle that moves across the surface of a planet and conducts detailed geological studies pertaining to the properties of the landing cosmic environment. IDE (Integrated Development Environment) for Rover is the development environment for the language Rova. We developed a language called rova, this language only built for the machine rover. Rover is a remotely controlled vehicle, which consists of camera to capture the images. The dynamic model of a Six-Wheeled Articulated lunar rover is researched in this paper.Refer to vehicle dynamics theory, the forces acting on the wheels are analyzed in terms of mechanical principles and configuration features on rough terrain. The language Rova consists of set of instruction and some control structures, which is used to control the Rover and to capture the videos and images. Using this IDE we can send the signal to the Rover and the Rover will respond according to the signal. And it will send images and videos to the system.**

Keyword- Rova, Rover, Circular Trajectory.

I. INTRODUCTION

The development environment summarizes the Editor, Interpreter, Compiler, button controls, Image capturing, Graphical view of signals and Execution of the program for the language Rova. The language Rova consists of set of instruction and some control structure, which is used to control the Rover and to capture the videos and images. We proposed model to control the Rover by writing the code or by clicking the buttons in the development environment.

Rover is remotely controlled vehicle which consists of six wheels, five DC motors and camera in that two DC motors are connected with two center wheels to move the Rover, one DC motor is in the center of the Rover which is connected with four corner wheels to position the Rover. Other two DC motors are used to rotate the camera in X-axis and Y-axis.

To achieve our goal, we selected the Rover as a model and then we identified all functionally of the Rover. Based on the functionality we created a IDE. In that IDE we can use the language Rova. Rova is a new high-level language to control the Rover. This language depends on our Rover model. Rover has the receiver, which accepts the signal from the transmitter, which is connected in the parallel port.

II. RELATED WORKS

There are a large variety of solutions already developed for robotic rover systems that are based on artificial neural networks. Several of them are examined in this section: Robotics technology is emerging at a rapid pace, offering new possibilities for automating tasks in many challenging applications, especially in space explorations, military operations, underwater missions, domestic services, and medical procedures[1]. Particularly, in space exploration, robotic devices are formally known as planetary rovers or simply rovers and they are aimed at conducting physical analysis of planetary terrains and astronomical bodies, and collecting data about air pressure, climate, temperature, wind, and other atmospheric phenomena surrounding the landing sites [2]. Fundamentally, a rover is a space exploration robotic vehicle used particularly in exploring the land of a planet. It has the capability to travel across the surface of a landscape and other cosmic bodies. A rover has many features: It can generate power from solar panels; capture high-resolution images; move in 360 degrees with the help of a navigation camera (Navcam); walk across obstacles such as bumps and rocks; conduct deep analysis and record measurements using multiple types of spectrometers; find properties of materials to identify their types and their composition; search for geological clues such as water to detect any presence of life on the landing environment; and inspect the mineralogy and texture of the local terrain using panoramic cameras (Pancam) [3][4].

Basically, rovers can be autonomous capable of operating with little or no assistance from ground control or they can be remotely controlled from earth ground stations called RCC short for Remote Collaboration Center [5]. In essence, the movement of autonomous rovers is not directed by human operators; instead, it is controlled by complex algorithms that allow the rover to traverse paths on multiple terrains while avoiding obstacles and path errors. This capability is more formally known as path-planning in which a rover or any robotic vehicle can perform terrain analysis and select the safest route to travel across [6]. The rover can then proceed towards the goal location over the selected trajectory while avoiding obstacles without previous knowledge of their existence.

The planetary rovers are measured as an addition to mobile robotics technology and autonomous mobile robots have become a key technology for unmanned planetary exploration missions. Because the cost to send a rover onto a distant planet is very expensive, small robotic rovers with low launch mass, high mobility and anti-overturn capability are highly desirable. Some innovative rover concepts in this field have been presented. JPL Rocky7 [7] and NASA Sojourner is small six-wheeled robotic rover with unique rocker-bogie suspension system [8]. JPL Nanorover is a small four-wheeled robotic rover and is capable of ability to actively turn over [9]. SMC is another small four-wheeled one with combinability, because a wheel of the rover can join another contiguous wheel shaft by a manipulation arm on its shaft [10]. In addition, NASDA Micro5 and Russian Marsokhod is also a small robotic rover [11][12].

The most terrestrial locomotion types are based on wheels, legs and caterpillars [13]. Caterpillar locomotion demonstrates good off-road ability on various ground geometries and soft soils due to low contact pressure, but their caterpillar mechanisms would suffer from high power dissipation and invasion of abrasive particles of planetary soil to cause critical failures. Legged locomotion has been intensively researched as a mobility option for the rough terrain. However, the legged mechanisms are more complex and may need autonomy for foot placement selection; in articular, its control problems are overwhelming. Wheeled vehicles won on the basis of simplicity, efficiency, lightweight and ease of control. Wheeled locomotion is optimal for planetary missions, proved by Apollo Planetary Roving Vehicle (LRV), Russian Rover (Lunakhod) and Spirit Rover etc.

III. PROPOSED FRAMEWORK

In this section, we present architecture for Rover. There are eight sections in the system. They are

- Development Environment.
- Editor.
- Interpreter.
- Compiler.
- Execution
- Graphical view of signals.
- Image Capturing.
- Help.

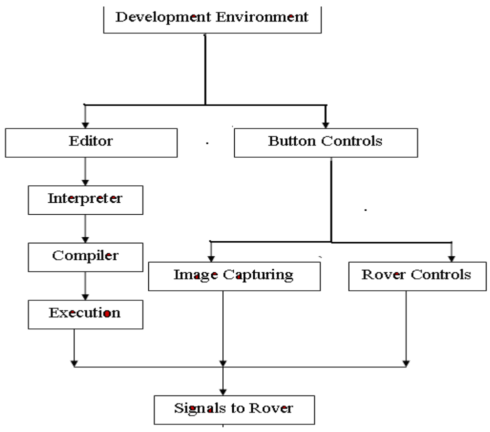

Fig. 1 Proposed framework for rover

A. *Development Environment*

The Development Environment summarizes the Editor, Interpreter, Compiler, Button controls, Image capturing, Graphical view of signals and Execution of the program for the language Rova. This Development Environment is powerful IDE for the language Rova. This provides the user with wide range of functionality. With this Development Environment we can directly Write, Compile and Execute the program. This Development Environment also shows the captured videos, images and graphical view of parallel port signals.

Using this Development Environment we can open and compile ‘rova’ extension file only. And we can run only ‘rovac’ extension file. The file saved from this Development Environment has the ‘rova’ extension.

B. Editor

Using this Editor, we can write the program for the language Rova. This Editor provides the pop-up help menu while writing the code. The pop-up menu contains all the instructions, which helps the user to write the code with correct syntax. This Editor automatically converts the non case sensitive instructions into cases sensitive instructions. To display the pop-up menu in Editor Window we used Help List table in IDE Database.

C. Interpreter

This interpreter checks the error line by line, if any error found then it shows the error message. This interpreter consists of two phases

- Syntax- This phase checks the syntax errors in the program and it will show the syntax error to the user.
- Semantics- This phase checks the semantic errors in the program and it will show the semantics error to the user.

D. Compiler

This compiler consists of three phases

- Syntax- This phase checks the syntax errors in the program and it will show the syntax error to the user.
- Semantics- This phase checks the semantic errors in the program and it will show the semantics error to the user.
- Intermediate code generator- After the compilation of the program, if there is no error then it will create intermediate file with the extension ‘rovac’.

E. Execution

The Execution of an instruction can be performed in two ways

1. Button Controls

2. Programs

For Execution we used Parallel Port table in IDE Database to calculate the value send to the parallel port.

1. Button Controls

In this Button controls we can directly send the signal to the Rover by clicking the buttons.

2. Programs

This section first it checks the intermediate file with extension ‘rovac’ created by the compiler. If such file found means it will start the execution otherwise it will shows the error message immediately. It will execute the instructions line by line. According to the instruction it will send the signals to the transmitter through parallel port. By writing the program we can perform the sequence of actions.

F. Graphical View Of Signal

This section shows that the graphical view of the output signals while executing the program or by clicking the buttons. The eight output pins in the parallel port is connected to the transmitter. Signals in the pins may be ‘0’ or ‘1’. This section draws the output signals in graphics by every one second.

Output signals may be ‘0’ or ‘1’

‘0’ indicates the value in the pin is 0.

‘1’ indicates the value in the pin is 1.

G. Image Capturing

In this section, we can capture the images and videos. And also we can store the captured images and videos in the specified location. Image capturing can be performed in two ways

1. Button Controls.

2. Programs.

By using this button controls we can directly capture the images and videos by clicking the buttons. And we can store it in a specified location.By using the instructions in the program we can capture the images and videos. And we can store it in a specified location.

H. Help

This section provides the guide about the Development Environment and also about the instructions and control structures in the language Rova.

IV. RESULTS AND DISCUSSION

The original impetus for Rova was to create a high level language, which should control the Rover. This Rova language is created entirely in Java. The output of Rova compiler is not executable code. Rather it is intermediate code. Intermediate code is optimized set of instructions. The intermediate file created by the Rova compiler has the rovac extension. Rova is new language, used to control the Rover. This language consists of keywords and control structures. Rover is remotely controlled vehicle, which consists of six wheels, five DC motors, camera and receiver circuit. Two DC Motors are connected with two center wheels. These two DC Motors are used to move the Rover in forward and reverse directions and also to rotate the Rover. Another two DC Motors are used to rotate the camera in X-axis and Y-axis. Remaining One DC Motor is connected with four corner wheels; it is used to position the four corner wheels in 45 and 90 degree. By aligning the four corner wheels in 45 degree and by moving the two center wheels in opposite directions we can able to rotate the Rover. Figure2 illustrates that Rover is in the shape of square and the DC Motor which is used to align the four corner wheels is fixed at the center of the Rover. To get the CIRCULAR TRAJECTORY from the square shape we have to align the four corner wheels to 45 degree (Figure 3) and to rotate the two center wheels in opposite direction. Our Rover is not like an ordinary vehicle, because our Rover does not use the concept of ordinary vehicle to turn right or to turn left.

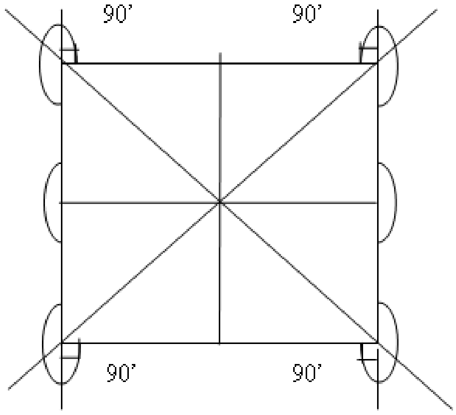

Fig.2 Normal position for Rover

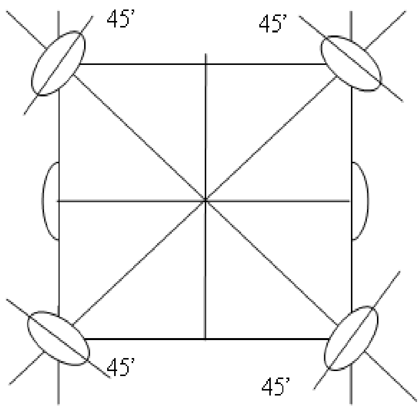

Fig. 3: After 45 degree rotated the Center Motor

To turn left or right

Figure 4 and figure 5 describe the process of how the rover will take turn left or right position. First step is to rotate the DC Motor which is located at the center of the Rover, so the four corner wheels will align to 45 degree.

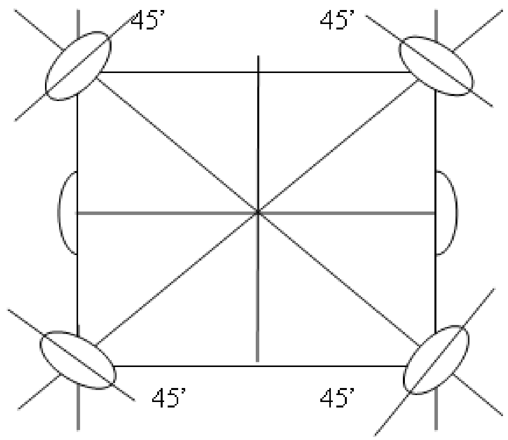

Fig. 4: After 45 degree rotated the Centre DC Motor

Second step is to move the center wheels in opposite directions, so the Rover will rotate.

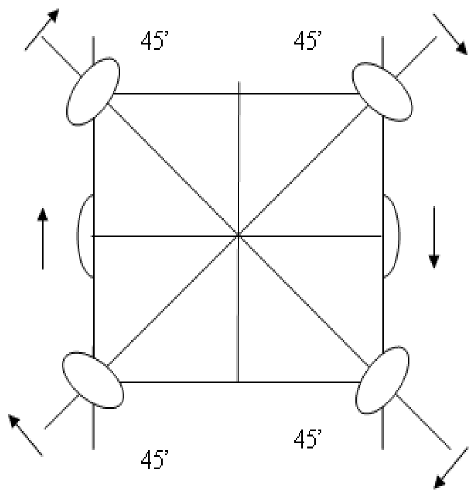

Fig. 5: Rover rotate in right direction

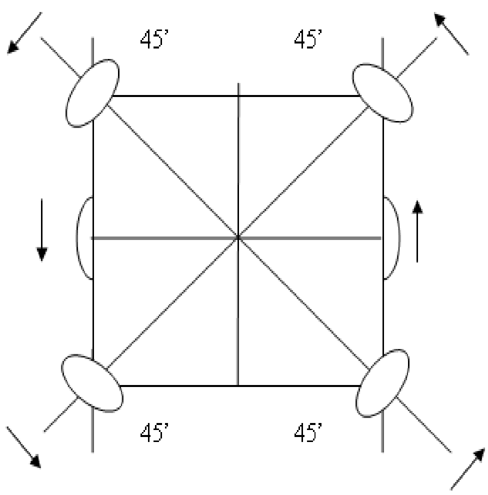

Fig. 6: Rover rotate in Left direction

After the Rover rotates up to 90 degree, the rover will turn left or right and then we have to align the four corner wheels to 90 degree. So the Rover will move front or move back. In ordinary vehicle, we can not able to turn left or right at the same place, i.e. where the vehicle is standing. But our Rover can do this by standing at the same place.

Difference between ordinary vehicle turn and rover turn

In ordinary vehicle we have to drive the vehicle to turn, so we can not able to get absolute turn, but our Rover can give the absolute turn by rotating to 90 degree (Figure 7).

Ordinary Vehicle Turn Right Rover Turn Right

Fig. 7.: Difference between ordinary vehicle and rover

Block Diagram & Circuit Design

In this section, we describes the over all block diagram (Figure 8) of Rover. In system side we have transmitter to transmit the signals and in Rover vehicle side we have receiver to receive the signals.

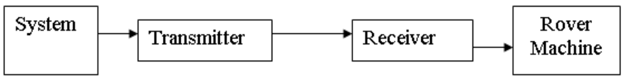

Fig. 8: Block diagram of Rover

When a program or control buttons executed the signal send to Transmitter through the Parallel Port and then Transmitter send the signal to Receiver. Based on the signal the Rover will respond.

PC Interface Circuit:

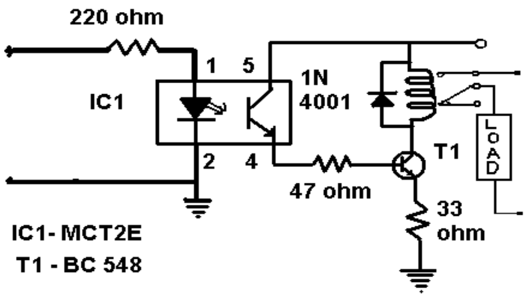

Fig. 9: PC Interface Circuit for transmitter and receiver

The circuit for interfacing the PC‘s parallel port with the load is very simple. It uses only one IC MCT2E, which isolates the PC and the relay driver circuits. The IC prevents the PC from any short circuit that may occur in the relay driver circuit. Transistor BC548 is used as a relay driver.

V. Conclusion

This paper proposes a model for rover helps the user, to control the Rover by writing the code or by clicking the buttons in the development environment. Thus, each section has been developed and tested individually to obtain the necessary required output in the desired form. This paper has been done as user-friendly software for easy handling of transactions. The entire system is documented and can be easily understood by the end users. The Development Environment is very user friendly and also easy to handle even by the beginners with very little effort and guidance.

Acknowledgment

This work has involved the efforts of many people whom we would like to thank: Rajavel, Karthik, Saaikumar, Vinothkumar, Karthikeyan and Gokul raj.

References

[1] Youssef Bassil , “Neural Network Model for Path-Planning Of Robotic Rover Systems” International Journal of Science and Technology (IJST), E-ISSN: 2224-3577, Vol. 2, No. 2, February, 2012.

[2] Amel Zerigui, Xiang Wu, Zong-Quan Deng, “A Survey of Rover Control Systems”, International Journal of Computer Sciences and Engineering Systems, Vol. 1, No. 4, pp. 105-109, 2007.

[3] Mars Exploration Rover, NASA Facts, National Aeronautics and Space Administration, Jet Propulsion Laboratory, California Institute of Technology Pasadena, 2004.
[4] Sarah Loff, NASA's Space Exploration Vehicle (SEV), NASA Official: Rocky Lind, 2011.
[5] Roland Siegwart, Illah R. Nourbakhsh, Davide Scaramuzza, Introduction to Autonomous Mobile Robots, 2nd ed, The MIT Press, 2011.
[6] Jean-Claude Latombe, Robot Motion Planning, Kluwer Academic Publishers, 1991.
[7] S. Hayati, R. Volpe, P. Backes, and J. Balaram, "The Rocky 7 rover :a Mars science craft prototype, in Proceedings of the 1997 IEEE International Conference on Robotics and Automation, Albuquerque, New Mexico, pp.2458-2460.
[8] Matijevic, and D. Shirley, ";The mission and operation of the Mars Pathfinder micro rover," Control Eng. Practice, 1997, pp. 827-833.
[9] E. Tunstel, "Evolution of autonomous self-righting behaviors for articulated nanorovers," in Proceedings of the 5th International Symposium on Artificial Intelligence, Robotics and Automation in Space, 1999, pp.341-346.
[10] Hirose S, "Super-Mechano-Clony and SMC rover with detachable wheel unite," in Proceedings ofCOE workshop99, pp.67-72.
[11] Takashi Kubota, and Yashuharu Kuniil, "Micro planetary rover 'Micro5'," in Proceedings of the 5th International Symposium on Artificial Intelligence, Robotics and Automation in Space, 1999,
pp.373-378.
[12] A. Kemurdjian, V. Gromov, and V. Mishkinnynk et al. "Small Marsokhod configuration," in Proceedings of the 1992 IEEE International Conference on Robotics andAutomation, pp.165-16.
[13] Hirose S, and Otsukasa N. et al, "Fundamental consideration for the design of a planetary rover," in Proceedings of the 1995 IEEE International Conference on Robotics andAutomation, pp.1939-1944.
[14] Z. F. BI, and Z. Q. DENG, "Distributed Vision System of Master-slaves Wireless Multiple Mobile Robots for Lunar Exploration," in Proceedings of the ICEMI '2005, pp. 352-355.
[15] Z. F. BI, and Z. Q. DENG, "Path planning for Master-slaves Wireless Multiple Mobile Robots," in Proceedings of the ISTM'2005, pp.7293-7296.