IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)
An International Open Access, Peer-reviewed, Refereed Journal

Line Tracking Mobile Robot Using Labview

[1]D.Prabhakaran, [2]T.Sathya

[1]Assistant Professor, [2]Assistant Professor
[1]Department of Mechatronics, [2]Department of Physics
[1]Mahendra Engineering College, Mallasamudram, Namakkal, India – 637 503
[2]Mahendra Engineering College for Women, Kumaramangalam, Namakkal, India – 637 503

Abstract: To control the movement of a robot, we need to understand and use some equations related to the position of the robot and its parts relative to both the world and to itself. This position is called the pose of the robot. The pose of a robot tells us the location of the robot in either two or three dimensions or also its orientation. That is, in what direction is it facing, the pose can be a simple three-element vector, but for arm-based robots, the robot portion of the project will have two distinct phases. First, we will work with embedded robotics controllers. This will mainly feature National Instruments controllers that are programmed using LabVIEW. LabVIEW is a graphical, commercial programming environment from National Instruments. It is quite often used in robotics, instrumentation testing and certain types of industrial control applications.

***Index Terms* - LabVIEW, Atmega328, Robotics**

I. Introduction

Robotics and Automation are becoming an essential component of engineering and scientific systems and consequently, they are very important topics for study by engineering and science students. Furthermore, robotics is built on fundamentals like transducer characterization, motor control, and data. Acquisition, mechanics of drive trains, network communication, computer vision, pattern recognition, kinematics, path planning, and others that is also fundamental to other fields, manufacturing, for instance. Learning these fundamentals can be challenging and fun by doing experiments with a capable mobile robot. The National Instruments (NI) LabVIEW Robotics Kit and LabVIEW provide an active-learning supplement to traditional robotics textbooks and curriculum by providing multiple capabilities in a compact and expandable kit.

The experiments described herein show how to communicate between a host computer and a robot, how robots communicate with sensors to obtain data from the robot's environment, how to implement algorithms for localization and planning in LabVIEW software, how the robot communicates with actuators to control sensor motion and driving motion, how to implement algorithms for controlling sensor and motion. National Instruments LabVIEW is a graphical programming environment used by millions of engineers and scientists to develop sophisticated measurement, test, and control systems using intuitive graphical icons and wires that resemble a flowchart. It facilitates integration with thousands of hardware devices and provides hundreds of built-in libraries for advanced analysis and data visualization – all for creating virtual instrumentation. The LabVIEW platform is scalable across multiple targets and operating systems.

The NI LabVIEW robotics module includes a robotic starter kit: an assembled robot with the frame, wheels, drive train, motors, transducers, computer, anmd wiring. The hardware can be studied, reverse engineered, and modified by the students.

2. Proposed Method

2.1 Kinematics

In this paper, Kinematics means the analytical study of the geometry of the motion of a machine, for a fixed reference coordinates, without regard to the focus that causes with the motion. The study of geometric and time-based quantities like position, velocity, and acceleration are considered. Inverse Kinematics is used to generate the trajectory angle of the robot motion. Robotics is keeping track of which frame of reference is being used. The global coordinate frame is relative to the robot's surroundings. We start robotics experiments with the robot at position (0, 0) and facing in the direction of the X axis ($\theta = 0$). The local coordinate frame is relative to the robot. We use the local coordinates when measuring the movement or setting the velocity of the wheels.

2.2 Robot Global Position

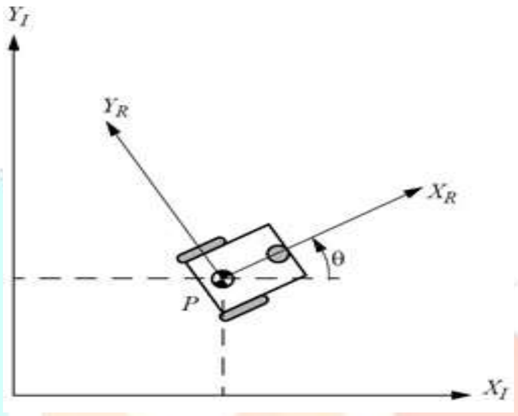

Fig.1: Axis position of mobile robot

The robot's location and orientation in the global coordinate frame the position, or pose, of the robot. The robot's position is represented as a matrix ($\mathcal{P}$) containing its x and y Cartesian coordinate pairs and its angle of orientation (θ).

$$\mathcal{P} = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix}$$

For robots that move in two dimensions, such as a mobile robot or arm type robots that move in a plane, the robot's pose is a three –element vector $\mathcal{P} = (x, y, \theta)$. An alternative expression of the pose is to define it as a coordinate frame. The coordinate frame consists of two perpendiculars (orthogonal) axis and (x, y) a point location of the origin of the coordinate frame. The coordinate frame pose is usually expressed by a matrix. This allows us to express the coordinate frame of connected parts and joints of the robot for other coordinate frames. Then by matrix multiplication and linear algebra, we can easily determine a coordinate frame to the world frame, or any other frame of interest.

2.3 Rotation of 2D Points

Rotation is a very important topic for both machine vision and robotics. Pixels in an image might be rotated to align objects with a model. After describing the rotation of a point, we can extend the concept of a rotation matrix to transformations consisting of rotation and translation. Then we consider transformations of coordinate frames that are used to describe the pose of robots and robotics moving parts. Rotation about other points is an extension of rotating about the origin. Point $P = (a, b)$ is rotated by an angle θ about the origin to point $P' = (a', b')$.

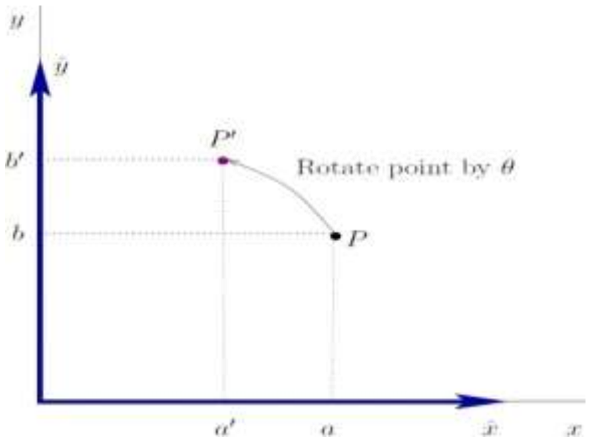

Fig.2: Rotate points of theta

To facilitate the discussion, the point P is defined in terms of unit vectors $\hat{x} = (1,0)$ and $\hat{y} = (0,1)$. The new location, P' is then defined by unit vectors $\hat{x'}$ and $\hat{y'}$ formed by rotating $\hat{x}$ and $\hat{y}$ by the angle θ.

$$P = a\,\hat{x} + b\,\hat{y}$$

$$P' = a\,\hat{x'} + b\,\hat{y'}$$

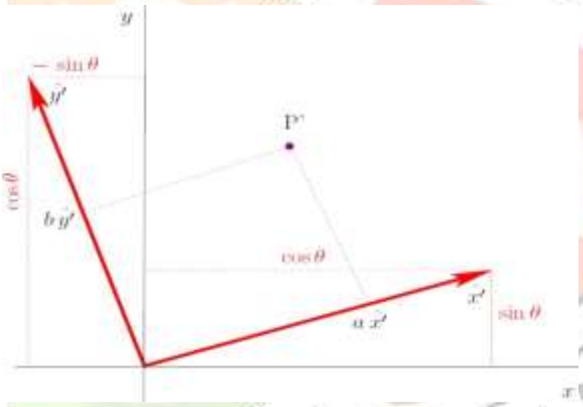

Fig.3: Point P of axis

$$\hat{x'} = \hat{x}\,\cos\theta + \hat{y}\,\sin\theta$$

$$\hat{y'} = -\hat{x}\,\sin\theta + \hat{y}\,\cos\theta$$

$$P' = \hat{x}(\,a\,\cos\theta - b\,\sin\theta\,) + \hat{y}(\,a\,\sin\theta + b\,\cos\theta\,)$$

Expressed in matrix notation:

$$\begin{aligned} P' &= \begin{bmatrix} \hat{x} & \hat{y} \end{bmatrix} \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} \\ &= \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} \end{aligned}$$

2.4 Rotation in 3D

Rotations are defined by one or more successive rotations about one of the three orthogonal axis of a coordinate frame. Rotation about the $z-$ axis is the same as 2D rotation. The rotation matrices can be directly gleaned from examining the geometry of rotated unit vector coordinate frames.

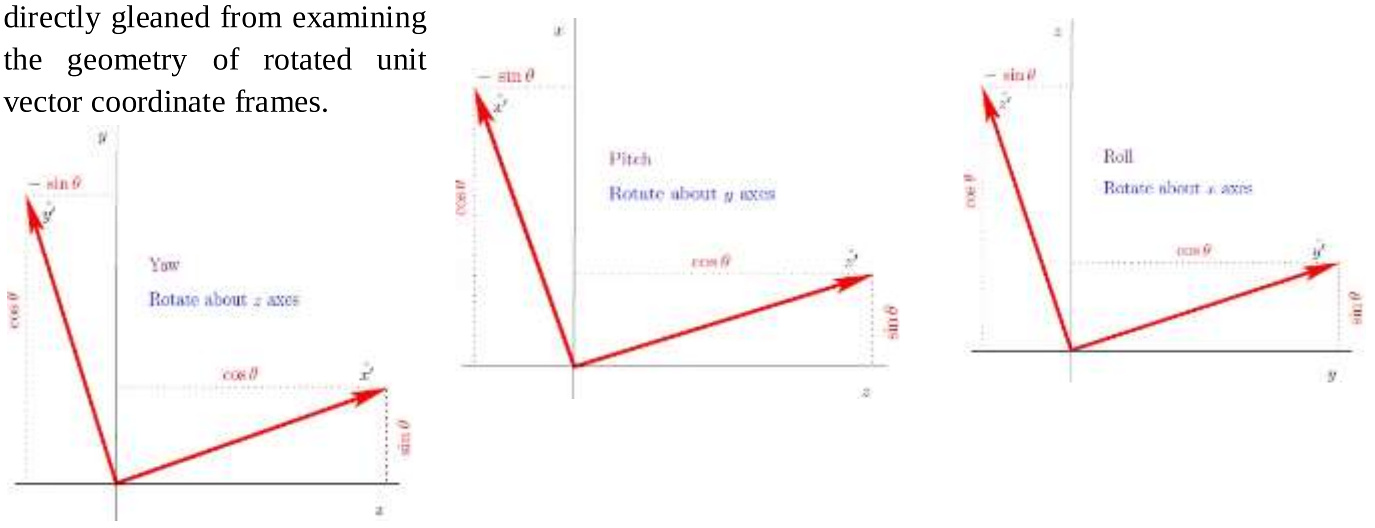

Fig.4: Rotation of 2D Axes

To the Front Panel, add a Horizontal Pointer Slider to enter the speed of the wheels in revolutions per minute (RPM) and a stop button. Note that the units of input to set the motor velocities are in units of radians/second, so we need conversion. The block diagram below used an equation node for this (Numeric function).

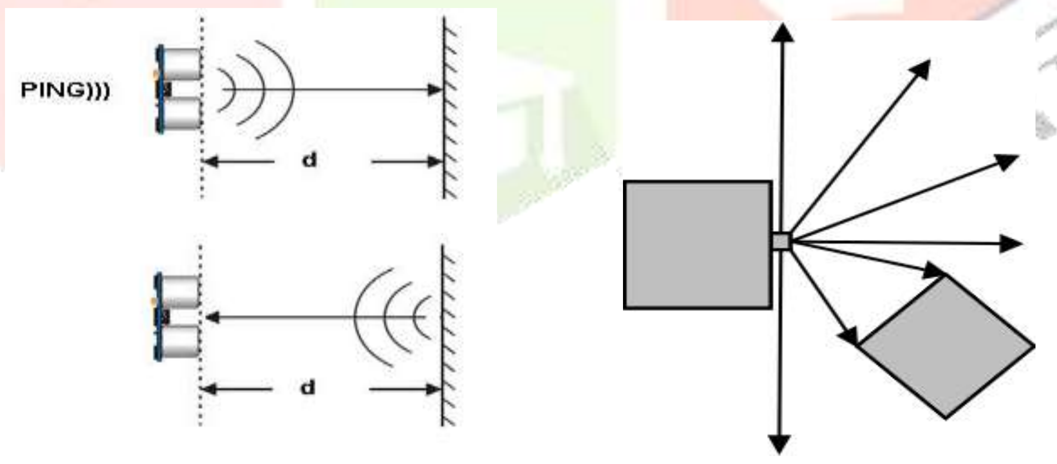

Fig.5: Ultrasonic sensor

2.5 Robotic Sensor

The sensors of a robot attempt to detect the robot’s environment. Thus, they are the potential inputs into autonomous algorithms. In a few specialized areas of computer vision, sensors can detect aspects of the environment better than humans, but in general, the sensors give the robot far less information about the environment than what a person is able to detect. This is one of the reasons why autonomous algorithms can be so difficult to master.

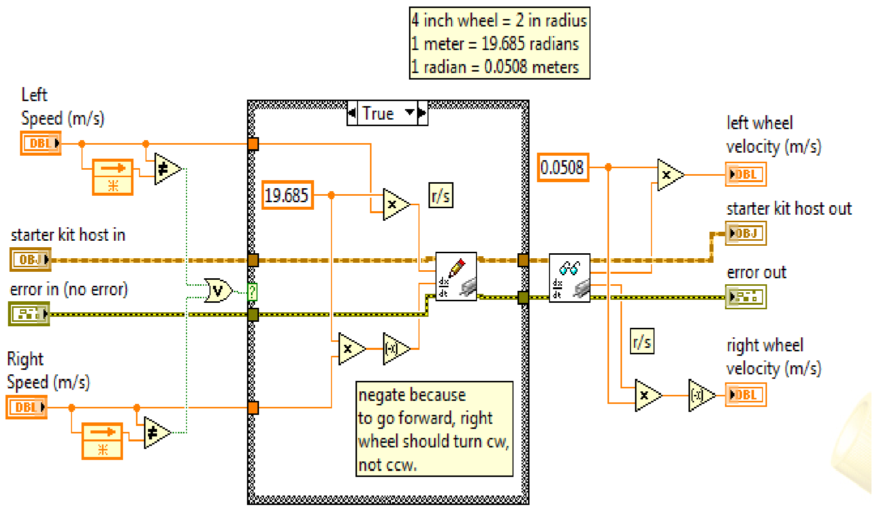

Fig.6: Path planning in LabVIEW

The basic idea is that ultrasonic acoustic pulses are transmitted (acoustic sound waves at a higher frequency than can be heard) and any reflected signals are received. To calculate an estimate for how far away an object is, the time difference between transmitting the pulse and receiving the reflection is measured. A common strategy in algorithms such as obstacle avoidance and wall fallowing is to treat a sequence of sonar measurements as vectors to obstacles. When we use the servo motor to rotate the transducer to a particular angle and take a measurement, we have an angle and a distance to either a found object or the maximum range of the transducer. Vectors to detected objects will be shorter than the vectors where no signal has reflected the transducer. We want to convert each vector to its (X, Y) coordinates relative to the robot and compute a composite sum of the vectors. Due to the symmetry of the measurements relative to the Y axis, when no object is detected, the Y values will cancel each other and the composite vector will point forward, but if some of the vectors are shorter due to detecting objects, then the composite vector will point away from the detected object.

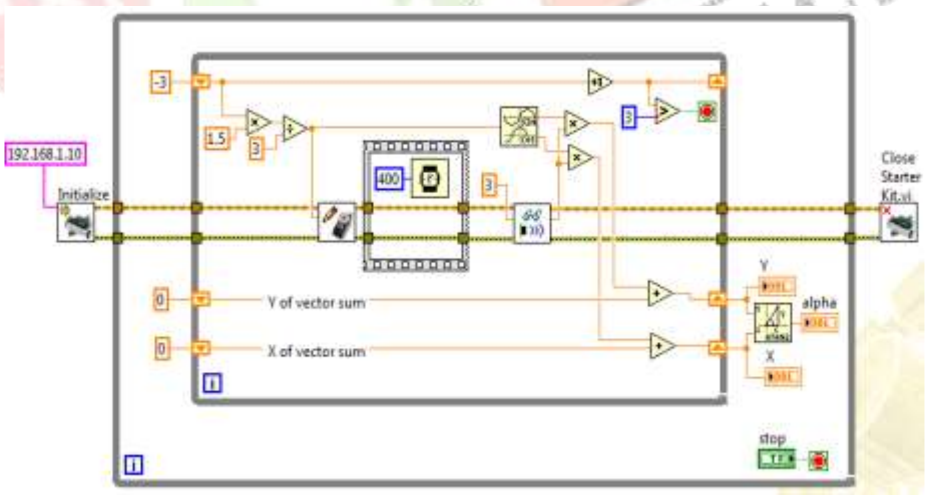

Fig.7: IP address assigning in LabVIEW

In the math variables used later, the vector in the robot’s coordinate frame indicating the angle to steer towards is given by the variable named. We measured the composite vector in the robot’s coordinate frame, but in the global coordinate frame, α it is the difference between the robot’s desired orientation and its current orientation $\alpha = \phi - \theta$. For obstacle avoidance *the* algorithm, we simply steer the robot towards the composite vector and sometimes we add a constant vector going in reverse to help the robot back out of tight places.

2.6 Infrared Sensor

Some robots use infrared (IR) instead of, or to supplement, ultrasound to detect objects. Like ultrasound, IR sensors use the principle of reflection to detect objects, except they use light (laser) instead of acoustic waves. Instead of being moved with a servo motor, IR sensors are usually mounted around the perimeter of the robot. The range of IR sensors is typically smaller than sonar and not all IR sensors have the same range.

2.7 Accelerometers

It measures the direction of forces, such as gravity and centrifugal force. In placed on a robotic arm, the output of the accelerometer is usually the sine of the angle between the arm and a line pointing directly down (vertically). Thus the accelerometer can be used to set the angle of the robot arm. A centrifugal force from rotation would also alter the reading from the accelerometer.

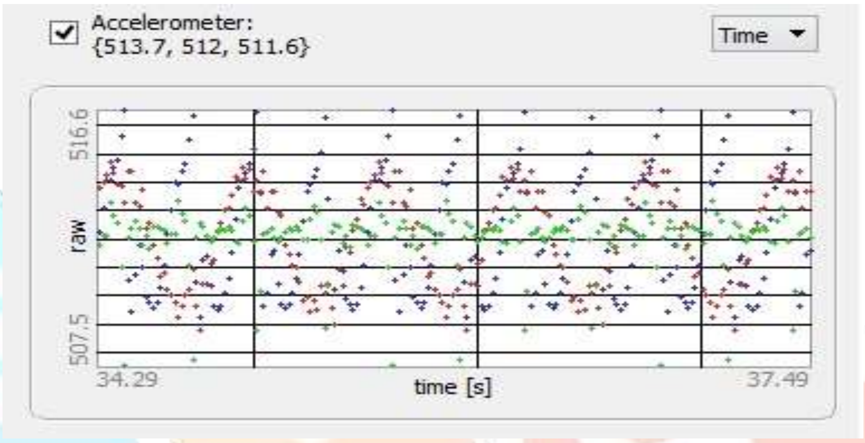

Fig.8: Accelerometer Graph

2.8 Gyro-scope

It was able to measure changes to the global orientation of the robot.

2.9 Video Camera

Cameras have a lot of applications for detecting objects. In some applications, lights are shined onto reflective tape so that a camera can easily detect an object and align itself to the object. The data from cameras is usually sent to an external computer because most robots do not have the processing speed to execute image processing algorithms in real-time.

1. Adding your code to the Finite State Machine robot control loop of the Drive Station block diagram, implement an algorithm
2. Test the robot’s ability to go to and stop at goals located at points: (1, 0), (1, 1), (0, 1), (-1, 1), (-1, 0), (-1, -1), (0, -1), and (1, -1).
3. Add the Hybrid avoid obstacle to the Drive.
4. Test the robot’s ability to go around flat or convex obstacles and stop at a goal location.
5. Earn extra points, as well as laurel’s of fame, prestige, and honor by demonstrating algorithm.

2.10 Hybrid FSM

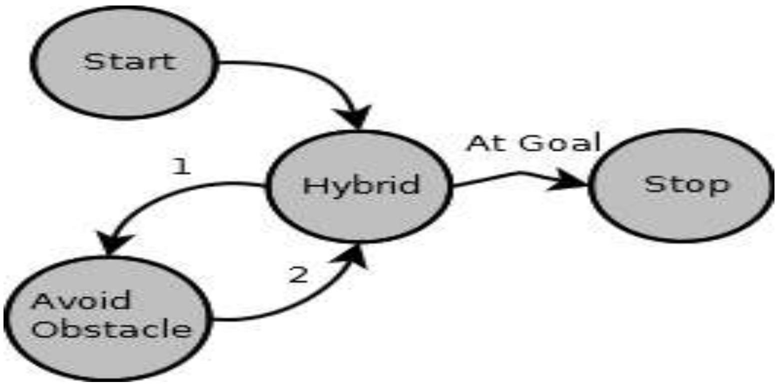

Fig.9: Hybrid FSM Model

Finite State Machine Diagram for the Hybrid Avoid Obstacles and Go-to-Goal Algorithm the conditions for the labeled transitions are as follows.

$$|\alpha_{ao}| \geq Max_{\alpha\_Hybrid}$$

$$|\alpha_{ao}| < Max_{\alpha\_Hybrid}$$

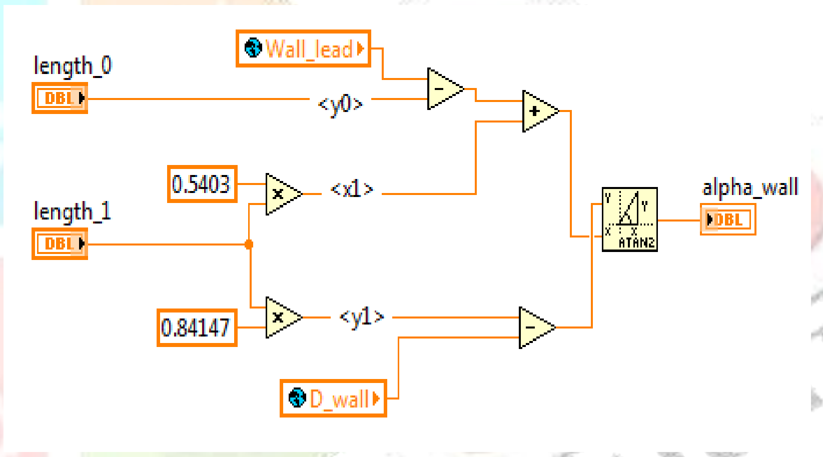

Fig.10: Length axis value

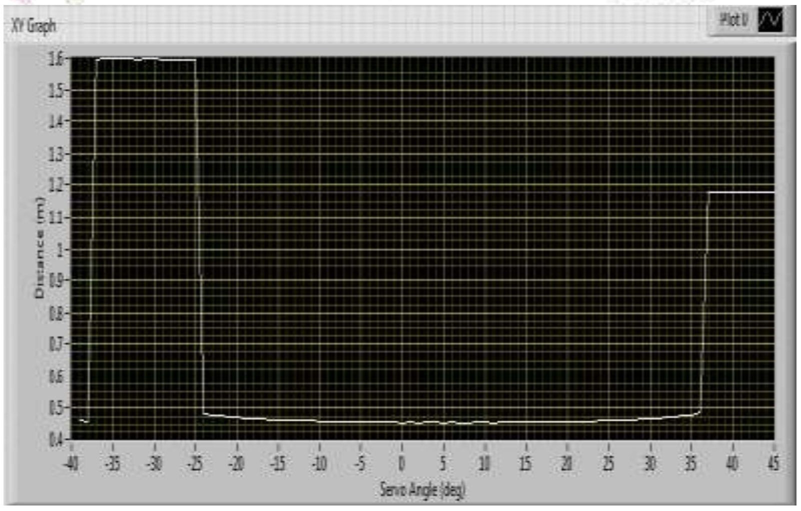

Fig.11: Movement of the mobile robot

3. Implementation & Conclusion

Dealing with the displacement and velocities of the two wheels of a differential drive robot is messy. A preferred model is that of a unicycle, where we can think of the robot as having one wheel that can move with the desired velocity (V) at a specified heading (ϕ). Having the equations to translate between the unicycle model and our wheel velocities is what allows us to simplify the robot with the unicycle model. We have seen how to take measured wheel displacements to calculate the new robot pose. Now we do the reverse and calculate the desired wheel velocities from the unicycle model.

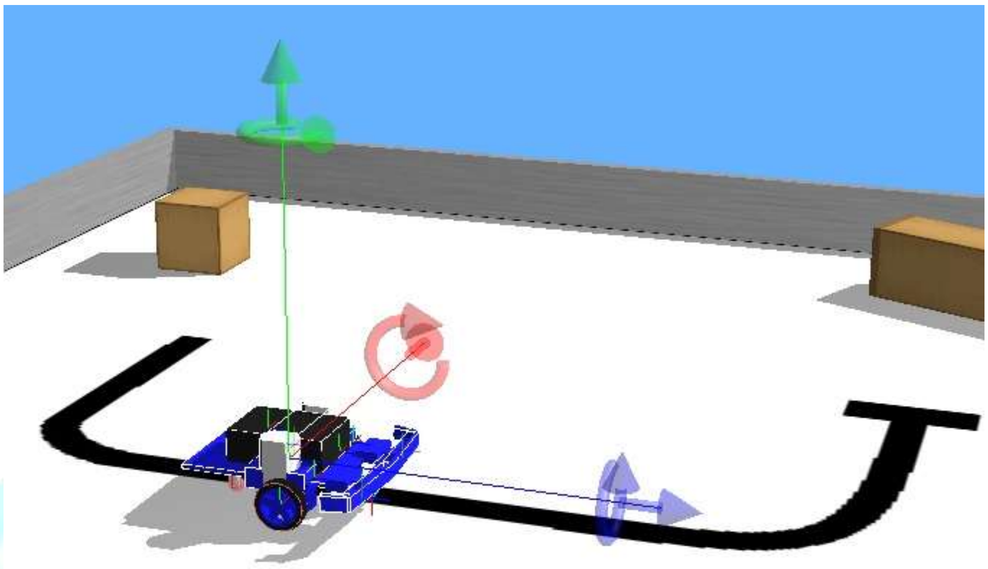

Fig.12: Line tracking mobile robot SCADA animation

References

[1] P. Sardain and G. Bessonnet, ``Forces acting on a biped robot. Centre of pressure-zero moment point,'' IEEE Trans. Syst., Man, Cybern. A, Syst., Humans, vol. 34, no. 5, pp. 630372, Sep. 2004.

[2] K. Erbatur and O. Kurt, ``Natural ZMP trajectories for biped robot reference generation, ''IEEE Trans. Ind. Electron., vol. 56, no. 3, pp. 835845, Mar. 2009.

[3] C. Fu and K. Chen, ``Gait synthesis and sensory control of stair climbing for a humanoid robot,'' IEEE Trans. Ind. Electron., vol. 55, no. 5, pp. 21112120, May 2008.

[4] S. Kajita, T. Nagasaki, K. Kaneko, and H. Hirukawa, ``ZMP-based biped running control,'' IEEE Robot. Autom. Mag., vol. 14, no. 2, pp. 6372, Jun. 2007.

[5] E. Ohashi, T. Aiko, T. Tsuji, H. Nishi, and K. Ohnis hi, ``Collision avoidance method of humanoid robot with arm force,'' IEEE Trans. Ind. Electron., vol. 54, no. 3, pp. 16321641, Jun. 2007.

[6] D.Prabhakaran, T.Jesudas, "A Novel Cognitive Intelligence Network with Architecture for Trajectory Recognition and Prediction of Destinations," Journal of Environmental Protection and Ecology, 24 (5), 1692–1700.

[7] T. S. Li, Y.-T. Su, S.-H. Liu, J.-J. Hu, and C.-C. Chen, ``Dynamic balance control for biped robot walking using sensor fusion, Kalman lter, and fuzzy logic,'' IEEE Trans. Ind. Electron., vol. 59, no. 11, pp. 43944408, Nov. 2012.

[8] K. Matsuoka, ``Sustained oscillations generated by mutually inhibiting neurons with adaptation,'' Biol. Cybern., vol. 52, no. 6, pp. 367376, 1985.

[9] K. Matsuoka, ``Mechanisms of frequency and pattern control in the neural rhythm generators,'' Biol. Cybern., vol. 56, nos. 56, pp. 345353, 1987.

[10] D.Prabhakaran, T.Jesudas, “Trajectory Pattern Recognition with Early Destination Prediction Algorithm for Intelligence Cognitive Networks and Architecture,” PERIODICO di MINERALOGIA, Volume 91, No. 5, 2022.

[11] C. Liu, D.Wang, and Q. Chen, ``Central pattern generator inspired control for adaptive walking of biped robots," IEEE Trans. Syst., Man, Cybern., Syst., vol. 43, no. 5, pp. 12061215, Sep. 2013.

[12]T. S. Li, Y.-T. Su, S.-W. Lai, and J.-J. Hu, ``Walking motion generation, synthesis, and control for biped robot by using PGRL, LPI, and fuzzy logic," IEEE Trans. Syst., Man, Cybern. B, Cybern., vol. 41, no. 3, pp. 736748, Jun. 2011.